

Spécialité Ingénieur Robotique Systèmes Robotiques et Interactifs



Modélisation robotique II

Michel TAÏX

Contexte

Nous étudierons principalement les robots manipulateurs à base fixe (**robotique industrielle**) avec une introduction à la modélisation des robots mobiles à roue (**robotique de service**). Cette classification est de moins en moins rigoureuse avec l'introduction des manipulateurs mobiles ou de la robotique humanoïde. Cet enseignement de modélisation des robots est la continuité du cours en SRI1. Cet enseignement est utilisé dans les deux matières suivantes :

- Commande de systèmes robotiques (S7)
- Contrôle de robot mobile (S8)

Objectifs

A l'issue de ces enseignements vous devez être capable de :

- modéliser n'importe quel type de robot série industriels que ce soit au niveau géométrique, cinématique ou dynamique (la modélisation dynamique sera complétée au niveau SRI3),
- générer des mouvements dans l'espace articulaire et opérationnel,
- modéliser des robots mobiles à roues et savoir appliquer des contrôles de mouvements classiques,
- connaître les constituants des principaux robots.

Les enseignements sont complétés par différents TP.

Note à l'attention des étudiants

Les photocopiés des cours de robotique comporte des chapitres écrits sous forme de notes ou sous forme de transparents. Ils ne sont qu'un support de cours, ils ne sont pas écrits comme un ouvrage, ils comprennent seulement le **minimum d'information** et doivent être complétés (et parfois corrigés!) lors des séances par une prise de notes personnelles.

Les chapitres 1 et 2 sont principalement un rappel des notions de SRI1 avec quelques compléments. Il est conseillé de prendre du temps à les étudier si ces notions ne sont pas acquises.

Bibliographie

Pour ceux qui veulent aller plus loin (références/images aussi utilisées sur les transparents) :

- Modélisation, identification et commande des robots. W. Khalil, E. Dombre. Hermès. 1999.
- Robotique. Aspects fondamentaux. J-P. Lallemand et S. Zeghloul. Masson, 1994.
- Robot Modeling and Control. M.W. Spong, S. Hutchinson, M. Vidyasagar. Wiley. 2005.
- Introduction to robotics : mechanics and control. J.J. Craig. Pearson Prentice Hall. 2009.
- Robotics : Modelling, Planning and Control, Bruno Siciliano and all. Springer. 2009.
- Robot Manipulators, E. Dombre, W. Khalil. Wiley. 2007.

Vous trouverez de multiples liens sur des cours de robotique sur le web, en voici deux à titre d'exemple :

www.gdr-robotique.org/cours_de_robotique/

www.diag.uniroma1.it/deluca

Attention à toujours vérifier la source dans ce cas...et ne pas oublier :

L'information n'est pas la connaissance. La seule source de connaissances est l'expérience. Vous avez besoin d'expérience pour acquérir la sagesse (citation d'Albert Einstein).

Table des matières

1 Généralités sur les Bras Manipulateurs	2
1.1 Définitions	4
1.2 Morphologie des BM	6
1.3 Caractéristiques des BM	7
2 Outils fondamentaux en robotique	8
2.1 Matrices de transformation	8
2.2 Situation d'un corps rigide	11
2.3 Formules utiles	16
3 Modèle Géométrique Direct (MGD)	17
3.1 Calcul de $T_{0,n}$	18
3.2 Calcul de X	21
3.3 Aspect pratique	21
3.4 Représentation URDF pour la modélisation des robots	22
4 Modèle Géométrique Inverse (MGI)	27
4.1 Méthodes analytiques	27
4.2 Méthodes de calcul	27
4.3 Principaux types d'équations associées au MGI	27
5 Modèle Différentiel Direct (MDD)	33
5.1 Définition de la Jacobienne	36
5.2 Méthodes de calcul du MDD	36
5.3 Jacobienne (géométrique)	36
5.4 Calcul de la Jacobienne analytique	38
6 Modèle Différentiel Inverse (MDI)	39
6.1 Méthodes de calcul du MDI	39
6.2 Résolution dans le cas de robot non-redondant	39
6.3 Singularités	39
6.4 Utilisation de la Jacobienne	39
7 Les robots séries redondants	47
7.1 Redondance	47
7.2 Méthodes de résolution basées sur la jacobienne	47
8 Génération de trajectoires (mouvements)	54
8.1 Définitions	54
8.2 GT dans l'espace articulaire	54
8.3 GT dans l'espace de la tâche	54
9 Modèle dynamique	68
9.1 Equations d'Euler-Lagrange	68
9.2 Equations d'Euler-Lagrange dans le cas d'un robot	68
9.3 Exemples	68
10 Modélisation de robot mobile à roues	74
10.1 Contrainte non holonôme	74
10.2 Modèle cinématique pour un RM de type unicycle	74
10.3 Modèle cinématique pour un RM de type voiture	74
10.4 Commandabilité	74

Chapitre 1

Généralités sur les Bras Manipulateurs

Tâche robotique

Une tâche robotique nécessite de **définir le placement** des objets dans l'environnement et aussi de **manipuler les repères** qui sont attachés à ces objets. Il nous faut donc définir un formalisme pour faciliter toutes les opérations de changement de repères et savoir définir un repère attaché à un objet.

Conventions :

- on ne considère que des repères directs (règle de la main droite, $\mathbf{z} = \mathbf{y} \wedge \mathbf{x}$).
- le sens positif pour une rotation est le sens trigonométrique (sens inverse des aiguilles d'une montre).

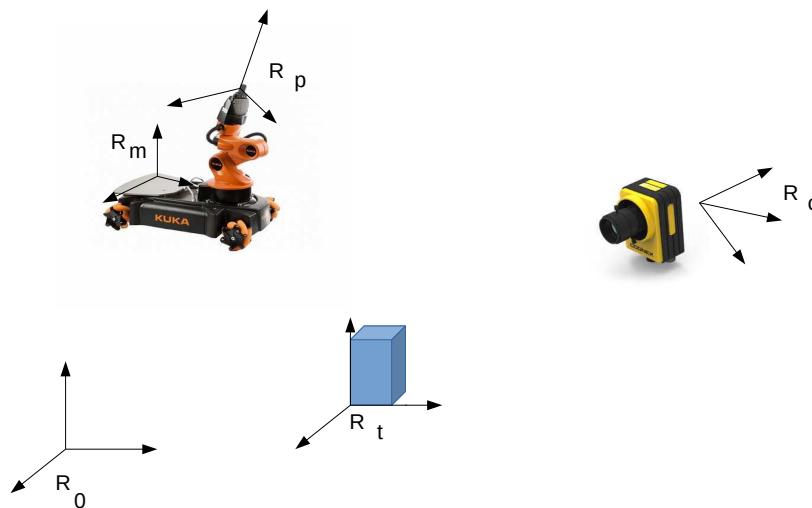


FIGURE 1.1 – Exemples de définition des repères lié à une application

Pourquoi construire des modèles?

L'utilisateur décrit généralement la tâche robotique sous la forme de mouvements et d'actions que doit effectuer l'outil porté par le robot. Le robot doit alors coordonner le mouvement de l'ensemble de ses liaisons pour réaliser le mouvement désiré tout en respectant les contraintes liées à sa mécanique.

La tâche est décrite dans l'espace opérationnel, \mathbf{X} , alors que le robot est contrôlé dans l'espace articulaire, \mathbf{Q} . Il faut donc construire des modèles entre ces espaces.

Modèle géométrique direct : $\dot{\mathbf{X}} = f(\mathbf{q})$ ou inverse $\dot{\mathbf{q}} = g(\mathbf{X})$ (p. ex. tâche de pick & place)

Modèle différentiel direct : $\dot{\mathbf{X}} = J(\mathbf{q}) \cdot \dot{\mathbf{q}}$ ou inverse $\dot{\mathbf{q}} = J(\mathbf{q})^{-1} \cdot \dot{\mathbf{X}}$ (p. ex. tâche de découpage à vitesse imposée)

Modèle dynamique direct : $\ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, \tau, \mathbf{F}_e)$ ou inverse $\tau = \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$ (p. ex. tâche d'interaction en effort).

Constitution d'un Bras Manipulateur

Un bras manipulateur, BM, peut se décrire par deux sous-ensembles :

- une Structure Mécanique Articulée (SMA) : assemblage de n corps rigides (*link*) reliés par des liaisons (*joint*). Chaque liaison est caractérisée par une coordonnée généralisé q_i (espace généralisé Q).

Son but est de pouvoir déplacer le dernier corps du robot (extrémité) dans une position et orientation désirée.

- un Organe Terminal (OT) : c'est le dernier corps du robot qui est destiné à manipuler l'outil (par exemple préhenseur de type pince ou effecteur de type torche de soudage,...). C'est l'interface qui permet au robot d'agir sur son environnement. Par abus de langage on parle parfois d'OT pour parler de l'outil puisque l'outil est fixé au dernier corps du robot. Bien souvent la position de l'OT ne correspond pas avec l'origine du dernier repère R_n , dans ce cas on introduit un point supplémentaire noté O_{n+1} .

Pour pouvoir réaliser une tâche robotique il faudra mettre en correspondance le repère attaché à l'OT, ou à l'outil, avec le ou les repères associés à la tâche.

Différents types d'assemblage entre corps/liaisons sont possibles (voir figure 1.2) :

- chaînes série (simple) : 1 antécédent et 1 successeur
- chaînes arborescente : 1 antécédent et 1 ou plusieurs successeurs
- chaînes complex (parallèle) : 1 ou plusieurs antécédent et 1 ou plusieurs successeurs

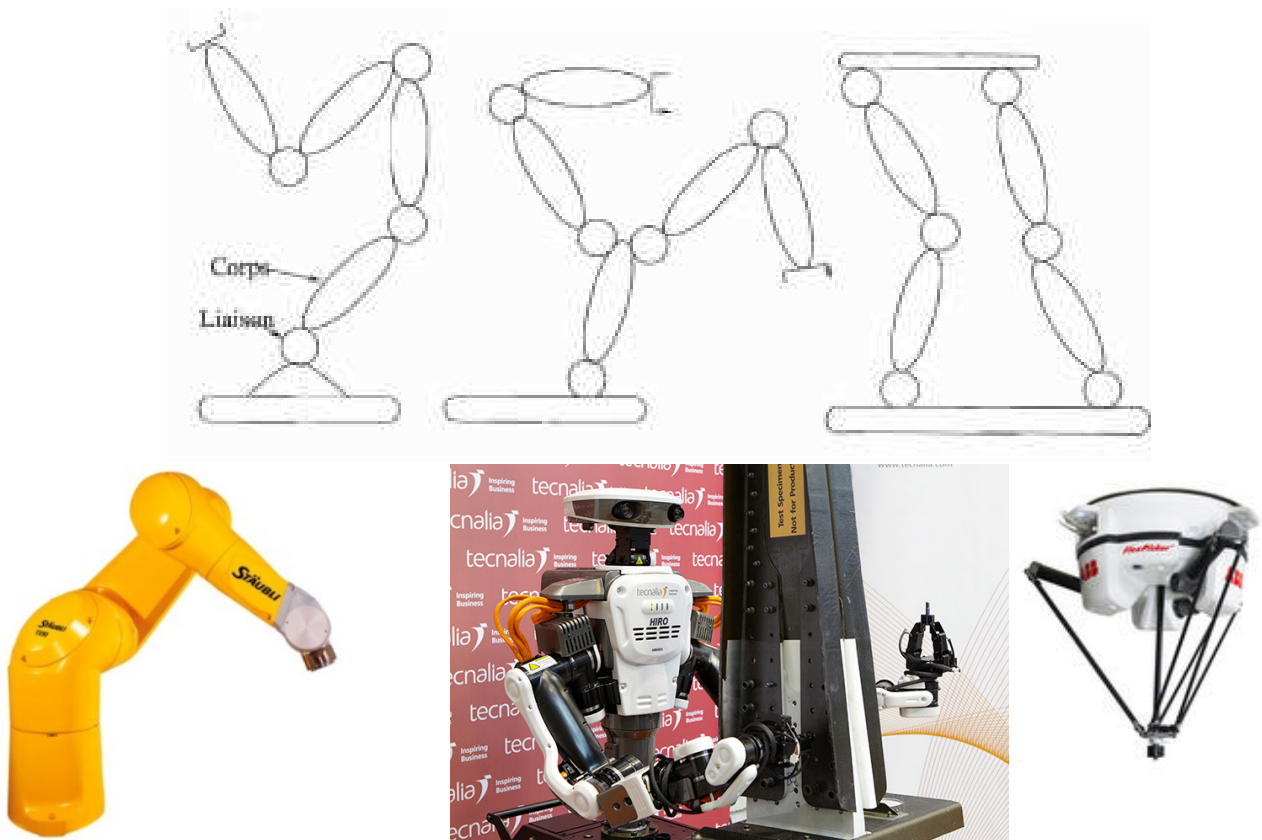


FIGURE 1.2 – Structures série, arborescente et complexe

Dans le cadre de ce cours, nous nous limitons aux structures séries.

1.1 Définitions

1.1.1 Situation

La **situation** d'un corps dans l'espace est définie par la donnée de la position et de l'orientation de ce corps par rapport à un repère de référence.

Il faut au minimum 6 paramètres pour décrire la situation de l'OT dans l'espace : 3 pour la position et 3 pour l'orientation. Si le nombre de paramètres est supérieur à 6 on a des paramètres non indépendants (voir section 2).

On appelle donc situation de l'OT, l'orientation et la position de l'outil attaché au dernier corps du robot.

Situation \triangleq placement \triangleq position + orientation (attitude) \triangleq *pose* en anglais.

1.1.2 Liaison

Liaison \triangleq articulation

Une **liaison** lie deux corps successifs en limitant le nombre de degré de liberté (d.d.l.) de l'un par rapport à l'autre.

Le nombre de d.d.l. d'une liaison est le degré de mobilité, m , de la liaison. On a $0 \leq m \leq 6$

Sur les bras manipulateurs industriels on trouve deux types de liaison entre deux corps C_0 et C_1 qui possèdent un seul d.d.l. ($m = 1$) :

- Liaison rotoïde (R) : rotation autour d'un axe commun (pivot). La situation du corps C_1 par rapport au corps C_0 est donnée par un angle.
- Liaison prismatique (P) : translation suivant un axe (glissière). La situation du corps C_1 par rapport au corps C_0 est donnée par une distance.



FIGURE 1.3 – Liaison rotoïde et prismatique

Les différentes combinaisons de R et P permettent de créer tous types de robot.

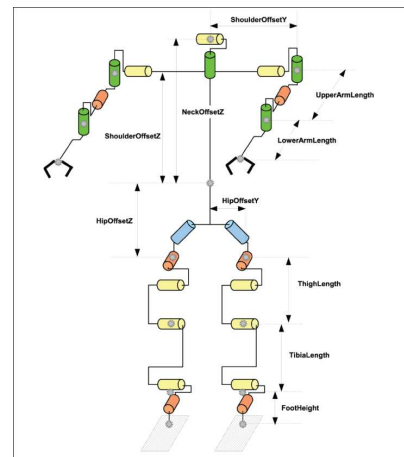
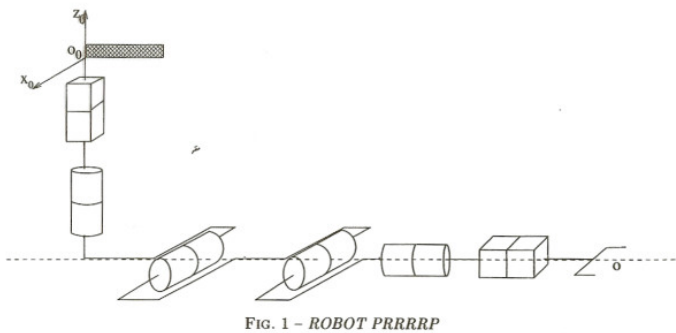


FIGURE 1.4 – 0

1.1.3 Espace généralisé

Pour un bras manipulateur possédant N liaisons (R ou P), on appelle indice de mobilité, M , le nombre de paramètres indépendants déterminant la géométrie du BM, c.a.d la situation de tous ses corps. Ce nombre de paramètres détermine la **configuration** du BM.

Pour un BM série on a $M = N$.

Dans le cas général la formule de Grübler permet de calculer le degré de mobilité d'un système mécanique quelconque. Soient n le nombre de corps (le sol est considéré aussi comme un corps), J le nombre de liaison, m le nombre de ddl d'un corps rigide ($m=3$ ou 6 pour mécanisme en 2D ou en 3D), f_j le nombre de ddl de l'articulation j et $c_j = 6 - f_j$ la classe de la liaison alors le nombre de ddl d'une chaîne cinématique est :

$$M = m.(n-1) - \sum_{j=1}^J c_j = m.(n-1) - \sum_{j=1}^J (6 - f_j) = m.(n-1 - J) + \sum_{j=1}^J f_j$$

\triangle : formule qui ne donne qu'une borne inférieure pour certaines structures complexes avec des boucles redondantes, contraintes des liaisons non indépendantes.

On note $\mathbf{q} = (q_1, q_2, \dots, q_M)^t$ le vecteur généralisé des M composantes des coordonnées généralisées. L'ensemble des vecteurs \mathbf{q} définit l'espace généralisé \mathbf{Q} . On parlera de vecteur généralisé ou de vecteur de configuration.

Les coordonnées généralisées (configurations, articulaires) sont les **variables que l'on va commander** pour faire bouger l'OT (c'est l'action des moteurs qui fait changer les variables q_i en bougeant les corps entre eux).

1.1.4 Espace Opérationnel

Le degré de liberté (*d.d.l.*) d'un BM, noté D , est égal au nombre de paramètres indépendants qui fixent la situation de l'OT.

On a $0 \leq D \leq 6$

On appelle vecteur opérationnel $\mathbf{X} = (x_1, x_2, \dots, x_D)^t$ le vecteur constitué des D coordonnées opérationnelles.

L'ensemble des \mathbf{X} définit l'espace opérationnel \mathbb{X} . La représentation de \mathbf{X} n'est pas unique et en pratique on prendra souvent un nombre de paramètres supérieur à six.

1.1.5 Redondance

Un BM est redondant si $D < M$

Si $M = D$ (non redondant) alors à chaque \mathbf{X} correspond un nombre fini de \mathbf{q}

Si $M > D$ (redondant) alors à chaque \mathbf{X} peut correspond nombre fini ou infini de \mathbf{q}

Intérêt de la redondance : augmenter le volume de travail de l'OT, avoir des d.d.l. libres au niveau des liaisons par rapport au d.d.l. de l'OT.

1.1.6 Configuration singulière

Dans certaines configurations, dites singulières, le nombre de d.d.l. de l'OT peut être inférieur à D , localement on a le nombre de d.d.l. $d < D$ et on a une configuration singulière d'ordre $(D - d)$.

Ces configurations correspondent au fait que, pour ces configurations, la structure mécanique induit des contraintes sur la situation de l'OT.

Le BM **perd des capacités de mouvement de son OT** pour ces configurations singulières.

1.1.7 Espace de travail d'un robot

L'espace de travail est l'ensemble des positions et orientations accessibles par un repère particulier lié à l'organe terminal du robot lorsque les paramètres articulaires prennent toutes les valeurs permises.

Généralement il est défini en position (*reachable workspace, workspace*) et on parle d'espace de travail primaire l'ensemble des positions de l'espace de travail qui peuvent être atteintes avec toutes les orientations possibles (*dexterous workspace*).

L'espace de travail du robot dépend du choix du robot et il doit être cohérent avec l'espace de la tâche (*task space*).

1.2 Morphologie des BM

On peut définir un grand nombre d'enchaînement de liaisons R ou P .

En général dans les structures industrielles, l'angle entre deux liaisons consécutives est fixé à 0 ou 90 degrés, les axes consécutifs sont donc parallèles ou perpendiculaires. Ce qui induit un nombre possible de structures par nombre de liaisons ($2l = 8, 3l = 36, 4l = 168, \dots$).

Dans le but de simplifier la relation non linéaire $\mathbf{X} = F(\mathbf{q})$ on découple souvent le **poignet** du **porteur** (voir figure 1.5).

Si le poignet possède trois axes concourants alors on peut avoir un découplage partiel de la situation de l'OT :

- la position de O_n ne dépend que de q_1, q_2, \dots, q_{n-3}
- l'orientation de l'OT dépend de toutes les liaisons
- le passage de O_n à O_{n+1} est simple

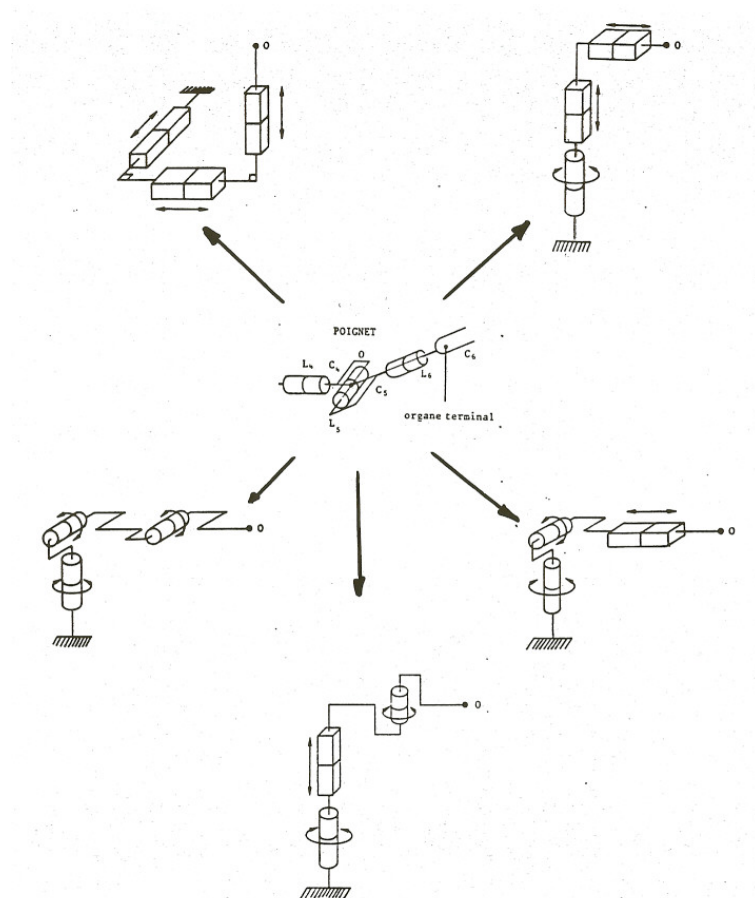


FIGURE 1.5 – Principales structures porteuses de poignet

1.3 Caractéristiques des BM

Volume de travail

Charge utile

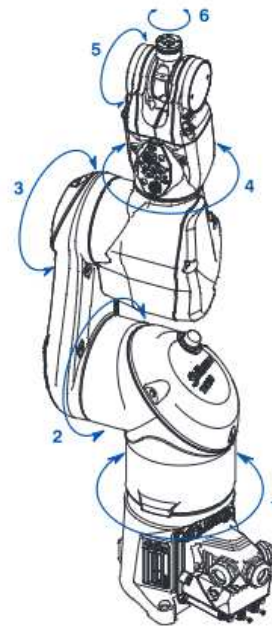
Vitesse et accélérations maximales (temps de cycle)

Précision (écart moyen)

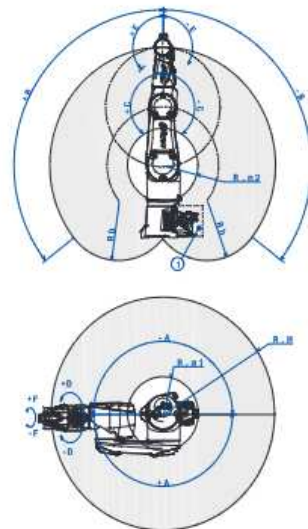
Répétabilité (dispersion)

MODÈLE	TX2-60	TX2-60L
Charge maximale	4,5 kg (9 kg sous conditions)	3,7 kg (5 kg sous conditions)
Charge nominale	3,5 kg	2 kg
Rayon (entre les axes 1 et 6)	670 mm	920 mm
Nombre de degré de liberté	6	6
Répétabilité suivant la norme ISO 9283	± 0,02 mm	± 0,03 mm
Contrôleur Staubli	CS9	CS9
Poids	52.2 kg	52,9 kg
VITESSE MAXIMALE		
Axe 1	435°/s	435°/s
Axe 2	410°/s	385°/s
Axe 3	540°/s	500°/s
Axe 4	995°/s	995°/s
Axe 5	1065°/s	1065°/s
Axe 6	1445°/s	1445°/s
Vitesse maximale au centre de gravité de la charge	8,4 m/s	11,1 m/s
Inertie maximale axe 5	0,325 kg.m²	0,12 5 kg.m²
Inertie maximale axe 6	0,1 kg.m²	0,032 kg.m²
Freins	Tous les axes	
ENVELOPPE DE TRAVAIL		
Rayon maxi entre axes 1 et 5 (R. M)	600 mm	850 mm
Rayon mini entre axes 1 et 5 (R.m1)	190 mm	209 mm
Rayon mini entre axes 2 et 5 (R.m2)	189 mm	208 mm
Rayon entre axes 3 et 5 (R.b)	310 mm	450 mm
PLAGE DE MOUVEMENT		
Axe 1 (A)	± 180°	± 180°
Axe 2 (B)	± 127,5°	± 127,5°
Axe 3 (C)	± 142,5°	± 152,5°
Axe 4 (D)	± 270°	± 270°
Axe 5 (E)	+132,5°/-121°	+132,5°/-121°
Axe 6 (F)	± 270° (1)	± 270° (1)
ENVIRONNEMENT D'INSTALLATION		
Température de fonctionnement suivant NF EN 60 204-1	+5°C à +40°C	
Humidité suivant NF EN 60 204-1	30% à 95% max. sans condensation	
Modes de fixation	Sol/mur/plafond	
Version câble sous le pied	•	
Version pressurisée	•	
CONNEXIONS AVANT-BRAS		
Pneumatique	2 lignes pneumatiques directes ou 2 électrodistributeurs en option : pression 5/2 bi-stable ou pression 5/3 mono-stable « centre fermé » ou vide 3/2 mono-stable	
Electrique	10 lignes électriques E/S et multibus à l'intérieur du bras + 3 lignes électriques + 4 câbles cat5e ; DIN M12	
Classe de propreté selon ISO 14644-1	5	
Classe de protection selon EN 60529	IP65 / IP67	
VERSIONS MARCHÉS SPÉCIFIQUES		
Version ElectroStatic Discharge (ESD)	•	
Version environnement humide (HE)	•	
Salle blanche classe de propreté 2 (Supercleanroom) - ISO 14644-1	•	
Version stérile (Stericlean)	•	

Plage de mouvement



Enveloppe de travail



(1) Configurable par logiciel jusqu'à ±11 250°

FIGURE 1.6 – Caractéristiques techniques Staubli TX-60

Chapitre 2

Outils fondamentaux en robotique

2.1 Matrices de transformation

2.1.1 Changement de repère vectoriel orthonormé

Soient deux bases vectorielles orthonormées : $B_i = (\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ et $B_j = (\mathbf{x}_j, \mathbf{y}_j, \mathbf{z}_j)$

On note $\mathbf{v}_i(\mathbf{v}_j)$, vecteur des composantes du vecteur \mathbf{v} dans $B_i(B_j)$.

$$\mathbf{v}_i = a^i \cdot \mathbf{x}_i + b^i \cdot \mathbf{y}_i + c^i \cdot \mathbf{z}_i = \begin{pmatrix} a^i \\ b^i \\ c^i \end{pmatrix} = \begin{pmatrix} \mathbf{v}_i \cdot \mathbf{x}_i \\ \mathbf{v}_i \cdot \mathbf{y}_i \\ \mathbf{v}_i \cdot \mathbf{z}_i \end{pmatrix} \text{ et } \mathbf{v}_j = a^j \cdot \mathbf{x}_j + b^j \cdot \mathbf{y}_j + c^j \cdot \mathbf{z}_j$$

Si on veut exprimer le \mathbf{v} dans la base B_i connaissant le \mathbf{v} dans la base B_j : $\mathbf{v}_i = \mathbf{v}_{j(i)} = a^j \cdot \mathbf{x}_{j(i)} + b^j \cdot \mathbf{y}_{j(i)} + c^j \cdot \mathbf{z}_{j(i)}$

$$= [\mathbf{x}_{j(i)} \quad \mathbf{y}_{j(i)} \quad \mathbf{z}_{j(i)}] \cdot \begin{pmatrix} a^j \\ b^j \\ c^j \end{pmatrix} = \mathbf{v}_i. \text{ On a : } \mathbf{v}_i = R_{i,j} \cdot \mathbf{v}_j \text{ avec } R_{i,j} \text{ matrice de rotation (changement de base).}$$

$$R_{i,j} = (\mathbf{x}_{j(i)} \quad \mathbf{y}_{j(i)} \quad \mathbf{z}_{j(i)}) = \begin{matrix} & \begin{matrix} x_j & y_j & z_j \end{matrix} \\ \begin{matrix} x_i \\ y_i \\ z_i \end{matrix} & \begin{pmatrix} \mathbf{x}_j \cdot \mathbf{x}_i & \mathbf{y}_j \cdot \mathbf{x}_i & \mathbf{z}_j \cdot \mathbf{x}_i \\ \mathbf{x}_j \cdot \mathbf{y}_i & \mathbf{y}_j \cdot \mathbf{y}_i & \mathbf{z}_j \cdot \mathbf{y}_i \\ \mathbf{x}_j \cdot \mathbf{z}_i & \mathbf{y}_j \cdot \mathbf{z}_i & \mathbf{z}_j \cdot \mathbf{z}_i \end{pmatrix} \end{matrix} = \begin{pmatrix} \mathbf{x}_{i(j)} \\ \mathbf{y}_{i(j)} \\ \mathbf{z}_{i(j)} \end{pmatrix}$$

$R_{i,j}$ représente le changement d'orientation entre les vecteurs de la base B_i et B_j .

On a donc : $R_{j,i} = R_{i,j}^T$ et $R_{i,j} = R_{j,i}^T = R_{j,i}^{-1}$ (matrice orthogonale). $\Rightarrow R_{i,j} \in SO(3)$.

$SO(3)$ (*special orthogonal group*) est l'ensemble des matrices R qui vérifient $R^T \cdot R = I$ et $\det(R) = 1$.

Propriétés : $R^{-1} = R^T \in SO(3)$, $R_1 \cdot R_2 \in SO(3)$, $(R_1 \cdot R_2) \cdot R_3 = R_1 \cdot (R_2 \cdot R_3)$, $R_1 \cdot R_2 \neq R_2 \cdot R_1$

Utilisation des matrices de rotation :

1. représenter une orientation
2. changer le repère de référence dans lequel on représente un vecteur ou un repère
3. appliquer une rotation à un vecteur ou une base, $R = Rot(\omega, \theta)$

Exemple : tableau

Changement entre n repères : produit des n changements de repères successifs.

$$\mathbf{v}_{0,n} = R_{0,1} \cdot R_{1,2} \cdot \dots \cdot R_{n-1,n} \cdot \mathbf{v}_n$$

Exemple : définition d'un repère de prise d'un objet localisé par une caméra au plafond, avec un manipulateur mobile équipé d'une caméra en bout de bras (figure 2.1)

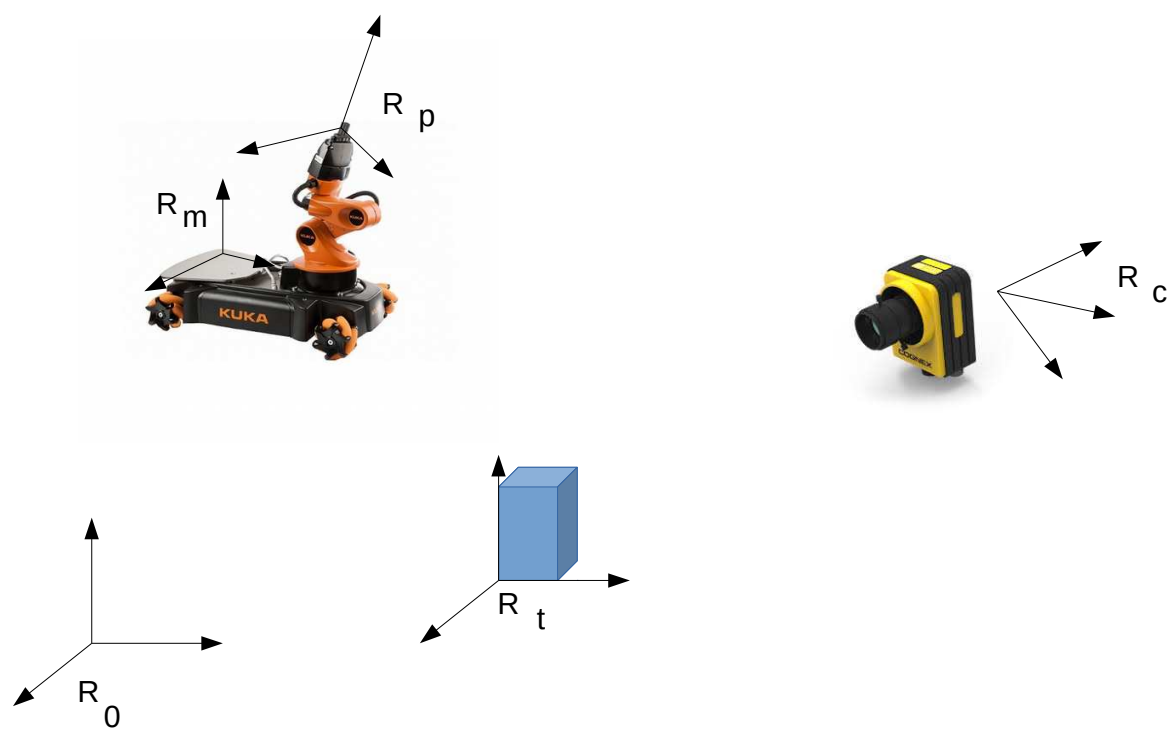


FIGURE 2.1 –

2.1.2 Changement de repère affine orthonormé

Soit deux repères affines (Origine + Base) orthonormés $R_0 = (O_0, B_0)$ et $R_1 = (O_1, B_1)$.

Soit $\underline{O_1 M_1} \triangleq \underline{X_1}$ coordonnées de M dans R_1 et $\underline{O_0 O_{10}} \triangleq \underline{P_{0,1}}$ coordonnées de O_1 dans R_0 **connues**.

$\underline{O_0 M_0} \triangleq \underline{X_0}$ coordonnées de M dans R_0 ?.

On sait que : $\underline{O_0 M} = \underline{O_0 O_1} + \underline{O_1 M}$. Dans la base R_0 : $\underline{O_0 M_0} = \underline{X_0} = \underline{O_0 O_{10}} + \underline{O_1 M_0} = \underline{P_{0,1}} + R_{0,1} \cdot \underline{X_1}$

Changement entre n repères : $\underline{X_0} = \underline{P_{0,1}} + R_{0,1} \cdot \underline{P_{1,2}} + R_{0,2} \cdot \underline{P_{2,3}} + R_{0,3} \cdot \underline{P_{3,4}} + \dots + R_{0,n} \cdot \underline{X_n}$

Relation beaucoup plus complexe pour des points que pour des vecteurs!

Uniformisation des changements de base en utilisant le formalisme des matrices homogènes.

2.1.3 Matrice de passage homogène

Composantes homogènes d'un vecteur $\underline{v} = \begin{pmatrix} a^i \\ b^i \\ c^i \end{pmatrix}$ dans la base $B_i : \underline{v_i} = \begin{pmatrix} \underline{v_i} \\ 0 \end{pmatrix} = \begin{pmatrix} a^i \\ b^i \\ c^i \\ 0 \end{pmatrix}$

Composantes homogènes d'un point $\underline{M} = \begin{pmatrix} x_M \\ y_M \\ z_M \end{pmatrix}$ dans le repère $R_i : \underline{X_i} = \begin{pmatrix} \underline{X_i} \\ 1 \end{pmatrix} = \begin{pmatrix} x_{M(i)} \\ y_{M(i)} \\ z_{M(i)} \\ 1 \end{pmatrix}$

La matrice de passage homogène, ou de transformation homogène (de changement de repère affine orthonormé direct) entre le repère R_i et R_j est définie par :

$$T_{i,j} \triangleq \begin{pmatrix} \underline{x_j \cdot x_i} & \underline{y_j \cdot x_i} & \underline{z_j \cdot x_i} & \underline{O_i O_j \cdot x_i} \\ \underline{x_j \cdot y_i} & \underline{y_j \cdot y_i} & \underline{z_j \cdot y_i} & \underline{O_i O_j \cdot y_i} \\ \underline{x_j \cdot z_i} & \underline{y_j \cdot z_i} & \underline{z_j \cdot z_i} & \underline{O_i O_j \cdot z_i} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R_{i,j} & \underline{P_{i,j}} \\ \underline{0_{1 \times 3}} & 1 \end{pmatrix}$$

$T_{i,j}$ est donc une matrice 4x4 qui contient à la fois la composante rotation et la composante translation entre les deux repères.

$$\underline{X_i} = T_{i,j} \cdot \underline{X_j},$$

Composition de n changements de repères successifs : $R_0 \longrightarrow R_1 \cdots \longrightarrow R_n$

$T_{0,n} = T_{0,1} \cdot T_{1,2} \cdot T_{2,3} \cdots T_{n-1,n}$ (produit de matrices par bloc, attention aux dimensions des blocs!)

Matrice homogène inverse : $T_{j,i} = T_{i,j}^{-1}$

$$T_{j,i} = \begin{pmatrix} R_{i,j}^T & -R_{i,j}^T \cdot \underline{P_{i,j}} \\ \underline{0_{1 \times 3}} & 1 \end{pmatrix} \quad \triangleq : T_{i,j} \text{ n'est pas une matrice orthogonale.}$$

$T_{i,j}$ appartient à $SE(3)$, groupe spécial euclidien.

$SE(3)$ (*special Euclidian group*) est l'ensemble des matrice T qui vérifient $T = \begin{pmatrix} R & \underline{P} \\ \underline{0_{1 \times 3}} & 1 \end{pmatrix}$ avec $R \in$

$SO(3)$ et $\underline{P} \in \mathbb{R}^3$.

Propriétés :

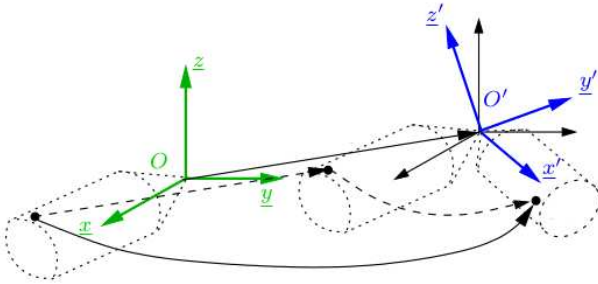
$$T^{-1} = \begin{pmatrix} R^T & -R^T \cdot \underline{P} \\ \underline{0} & 1 \end{pmatrix} \in SE(3), \quad T_1 \cdot T_2 \in SE(3), \quad (T_1 \cdot T_2) \cdot T_3 = T_1 \cdot (T_2 \cdot T_3), \quad T_1 \cdot T_2 \neq T_2 \cdot T_1$$

Utilisation des matrices homogènes :

1. représenter une situation d'un corps rigide,
2. changer le repère de référence dans lequel on représente un vecteur ou un point,
3. appliquer un mouvement à un vecteur ou un repère.

La matrice T qui décrit le passage du repère i au repère j définit la transformation rigide $g = (P, R)$

Exemple :



2.2 Situation d'un corps rigide

La situation d'un corps rigide P est décrite par rapport à un repère de base qu'on note $R_0 = (O_0, \mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$. Elle est définie par :

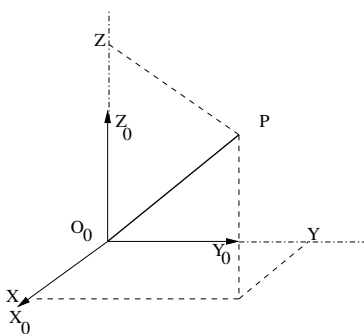
- la position d'un point du corps P noté O_P . En général O_P est idéalement situé au centre du corps P (ou centre de gravité)
- l'orientation d'un repère vectoriel **lié** au corps P qu'on note R_P .

On sépare les paramètres de situation en deux ensembles : $\mathbf{X} = \begin{pmatrix} \mathbf{X}_P \\ \mathbf{X}_R \end{pmatrix} = \begin{pmatrix} \text{coordonnées de position} \\ \text{coordonnées d'orientation} \end{pmatrix}$

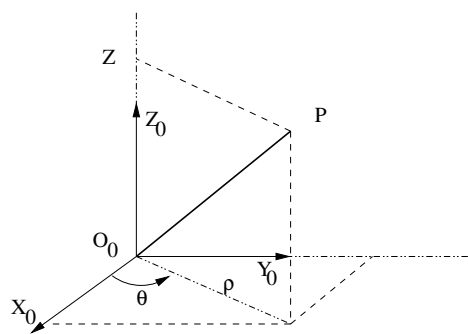
En pratique on prendra souvent plus de six paramètres pour décrire \mathbf{x} (paramètres liés).

2.2.1 Coordonnées de position de l'OT

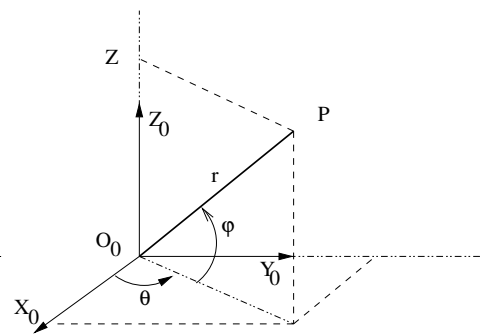
Paramètres décrivant la position de O_{n+1} dans $R_0, (O_0, X_0, Y_0, Z_0)$: 3 paramètres indépendants notés X_P .



(a) Cartésiennes



(b) Cylindriques



(c) Sphériques

2.2.1.1 Coordonnées cartésiennes : $X_P = (X, Y, Z)^t$

Projection de $\underline{O_0P}$ dans R_0 .

$$\mathbf{P}_0 = \begin{pmatrix} \underline{O_0O_P} \cdot \mathbf{X}_0 \\ \underline{O_0O_P} \cdot \mathbf{Y}_0 \\ \underline{O_0O_P} \cdot \mathbf{Z}_0 \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{X}_P$$

2.2.1.2 Coordonnées cylindriques : $\mathbf{X_P} = (\rho, \theta, Z)^t$

$$\rho \in [0, +\infty) \quad \theta \in [-\Pi, \Pi] \quad Z \in]-\infty, +\infty[$$

$$\mathbf{P_0} = \begin{pmatrix} \rho \cdot \cos(\theta) \\ \rho \cdot \sin(\theta) \\ Z \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \iff \mathbf{X_P} = \begin{pmatrix} \rho \\ \theta \\ Z \end{pmatrix} = \begin{pmatrix} \sqrt{X^2 + Y^2} \\ \text{Atan2}(Y, X) \\ Z \end{pmatrix}$$

Singularité de représentation si $X = Y = 0$

2.2.1.3 Coordonnées sphériques : $\mathbf{X_P} = (r, \theta, \phi)^t$

$$r \in [0, +\infty) \quad \theta \in [-\Pi, +\Pi] \quad \phi \in [-\frac{\Pi}{2}, +\frac{\Pi}{2}]$$

$$\mathbf{P_0} = \begin{pmatrix} r \cdot \cos(\phi) \cdot \cos(\theta) \\ r \cdot \cos(\phi) \cdot \sin(\theta) \\ r \cdot \sin(\phi) \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \iff \mathbf{X_P} = \begin{pmatrix} r \\ \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \sqrt{X^2 + Y^2 + Z^2} \\ \text{Atan2}(Y, X) \\ \arcsin(\frac{Z}{r}) \end{pmatrix}^1$$

Singularité de représentation si $X = Y = 0$ ou $X = Y = Z = 0$

2.2.2 Coordonnées d'orientation de l'OT

Paramètres décrivant l'orientation de la base $(\mathbf{x_n}, \mathbf{y_n}, \mathbf{z_n})$ lié au corps C_n dans la base $(\mathbf{x_0}, \mathbf{y_0}, \mathbf{z_0})$: 3 paramètres indépendants ou plus si les paramètres sont liés.

2.2.2.1 Systèmes d'angles indépendants

Succession de 3 rotations indépendantes suivant des axes différents ($X \longrightarrow Y \longrightarrow Z, Y \longrightarrow X \longrightarrow Y \dots$).

- On définit les trois matrices de rotation élémentaire
- On fait le produit des trois rotations dans le bon ordre
- Problème de singularité

Angles de Bryant : λ, μ, ν

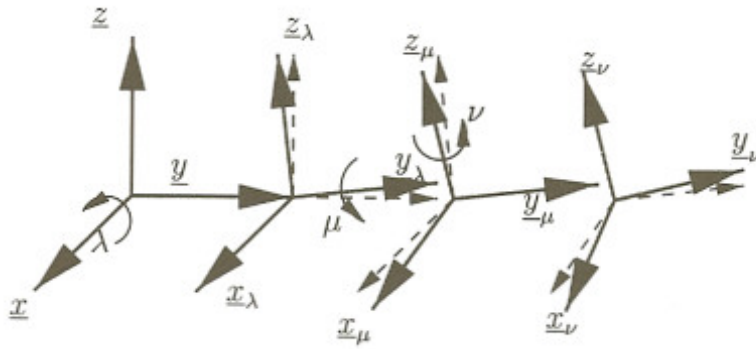


FIGURE 2.2 – Angles de Bryant

Les *angles de Bryant* (ou angles de Cardan) correspondent aux trois rotations successives (Cf. figure 2.2) :

- * λ autour du vecteur unitaire \underline{x} ,
- * μ autour du nouveau vecteur unitaire $\underline{y_\lambda}$ déduit du vecteur unitaire \underline{y} ,
- * ν autour du nouveau vecteur unitaire $\underline{z_{\mu}}$ déduit du vecteur unitaire $\underline{z_\lambda}$.

Les angles de Bryant sont choisis de telle manière que :

$$-\pi < \lambda \leq \pi, \quad -\pi/2 \leq \mu \leq \pi/2, \quad -\pi < \nu \leq \pi$$

1. la fonction $\text{Atan2}(y,x)$ est une extension de la fonction *arctangente*. Pour tous arguments réels x et y non nuls, elle définit un angle appartenant à l'intervalle $] -\pi, \pi]$ avec y et x respectivement proportionnels au sinus et au cosinus de l'angle.

Calcul de R en fonction de λ, μ, ν R correspond à la composition de 3 rotations élémentaires :

$$R_{0,\lambda} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \lambda & -\sin \lambda \\ 0 & \sin \lambda & \cos \lambda \end{pmatrix} \quad R_{\lambda,\mu} = \begin{pmatrix} \cos \mu & 0 & \sin \mu \\ 0 & 1 & 0 \\ -\sin \mu & 0 & \cos \mu \end{pmatrix} \quad R_{\mu,\nu} = \begin{pmatrix} \cos \nu & -\sin \nu & 0 \\ \sin \nu & \cos \nu & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} \cos \mu \cos \nu & -\cos \mu \sin \nu & \sin \mu \\ \sin \lambda \sin \mu \cos \nu + \cos \lambda \sin \nu & -\sin \lambda \sin \mu \sin \nu + \cos \lambda \cos \nu & -\sin \lambda \cos \mu \\ -\cos \lambda \sin \mu \cos \nu + \sin \lambda \sin \nu & \cos \lambda \sin \mu \sin \nu + \sin \lambda \cos \nu & \cos \lambda \cos \mu \end{pmatrix}$$

Calcul de λ, μ, ν en fonction de la matrice des cosinus directeurs R

Si $t_{13} \neq \pm 1$:	Si $t_{13} = \pm 1$:
$\lambda = \text{Atan2}(-t_{23}, t_{33})^2$ $\mu = \text{Arc sin}(t_{13})^3$ $\nu = \text{Atan2}(-t_{12}, t_{11})$	$\mu = \pi \cdot t_{13} / 2$ $t_{13} \cdot \lambda + \nu = \text{Atan2}(t_{21}, t_{22})$ (les valeurs de λ et ν sont indéterminées)

Angles d'Euler classiques : ψ, θ, φ

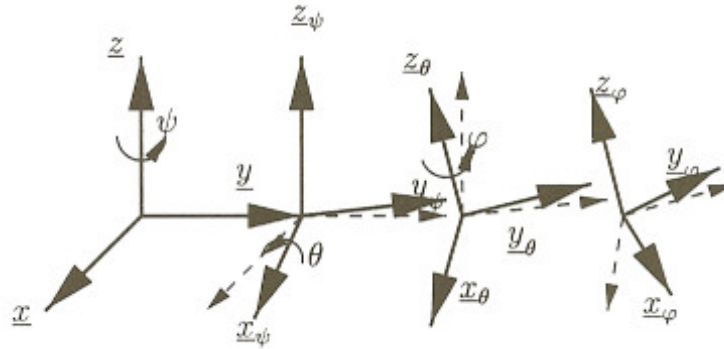


FIGURE 2.3 – Angles d'Euler

Les *angles d'Euler classiques* correspondent aux trois rotations successives (Cf. figure 2.3) :

- * ψ autour du vecteur unitaire \underline{z} ,
- * θ autour du nouveau vecteur unitaire \underline{x}_{ψ} déduit du vecteur unitaire \underline{x} ,
- * φ autour du nouveau vecteur unitaire \underline{z}_{θ} déduit du vecteur unitaire \underline{z} .

Ces angles sont appelés respectivement *précession* (ψ) , *nutaton* (θ) et *rotation propre* (φ).

Les angles d'Euler sont choisis de telle manière que :

$$-\pi < \psi \leq \pi, \quad 0 \leq \theta \leq \pi, \quad -\pi < \varphi \leq \pi,$$

Calcul de R en fonction de ψ, θ, φ R correspond à la composition de 3 rotations élémentaires :

$$R_{0,\psi} = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad R_{\psi,\theta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \quad R_{\theta,\varphi} = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} \cos \psi \cos \varphi - \sin \psi \cos \theta \sin \varphi & -\cos \psi \sin \varphi - \sin \psi \cos \theta \cos \varphi & \sin \psi \sin \theta \\ \sin \psi \cos \varphi + \cos \psi \cos \theta \sin \varphi & -\sin \psi \sin \varphi + \cos \psi \cos \theta \cos \varphi & -\cos \psi \sin \theta \\ \sin \theta \sin \varphi & \sin \theta \cos \varphi & \cos \theta \end{pmatrix}$$

Calcul de ψ, θ, φ en fonction de la matrice des cosinus directeurs R

Si $t_{33} \neq \pm 1$:	Si $t_{33} = \pm 1$:
$\psi = \text{Atan2}(t_{13}, -t_{23})$ $\theta = \text{Arc cos}(t_{33})$ $\varphi = \text{Atan2}(t_{31}, t_{32})$	$\theta = \pi(1 - t_{33})/2$ $\psi + t_{33} \cdot \varphi = \text{Atan2}(t_{21}, t_{11})$ (les valeurs de ψ et φ sont indéterminées).

Angles de roulis, tangage, lacet (roll, pich, yaw) : γ, β, α

Ce sont des angles très utilisés par les anglo-saxons. Il s'agit des angles d'Euler non classiques où on effectue trois rotations successives autour des axes fixes du repère initial : autour de x puis de y et enfin de z .

$$R = R(\mathbf{z}, \alpha) \cdot R(\mathbf{y}, \beta) \cdot R(\mathbf{x}, \gamma)$$

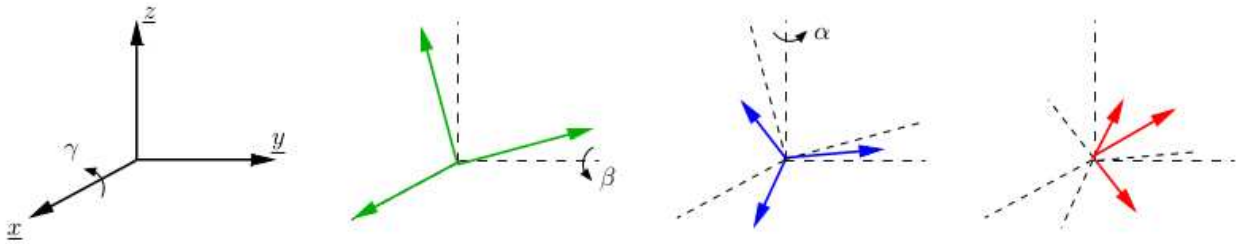


FIGURE 2.4 – Angles de roulis, tangage, lacet

$$R = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}$$

$$R = \begin{pmatrix} \cos \alpha \cos \beta & -\sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma & \sin \alpha \sin \gamma + \cos \alpha \sin \beta \cos \gamma \\ \sin \alpha \cos \beta & \cos \alpha \cos \gamma + \sin \alpha \sin \beta \sin \gamma & -\cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{pmatrix}$$

A titre d'exercice vous calculerez la transformation inverse.

2.2.2.2 Systèmes d'angles redondants

Il est parfois utile d'exprimer les rotations à l'aide de paramètres liés.

Cosinus directeurs complets : $X_R = (x_x, x_y, x_z, y_x, y_y, y_z, z_x, z_y, z_z)^t$

Les *cosinus directeurs complets* sont les composantes, dans la base \mathcal{B}_0 , des vecteurs $\underline{x}_n, \underline{y}_n$ et \underline{z}_n de la base \mathcal{B}_n lié au corps n . Il s'agit, par conséquent, des éléments de la matrice R considérés en colonne.

Ces paramètres ont tous un module inférieur ou égal à un, et ont le grand inconvénient d'être au nombre prohibitif de 9. Il existe de très nombreuses relations de dépendance entre ces neuf paramètres, dues au fait que la matrice de ces paramètres est orthogonale (6 relations indépendantes entre les 9 paramètres).

$$R_{0,n} = \begin{pmatrix} \underline{x}_0 \cdot \underline{x}_n & \underline{x}_0 \cdot \underline{y}_n & \underline{x}_0 \cdot \underline{z}_n \\ \underline{y}_0 \cdot \underline{x}_n & \underline{y}_0 \cdot \underline{y}_n & \underline{y}_0 \cdot \underline{z}_n \\ \underline{z}_0 \cdot \underline{x}_n & \underline{z}_0 \cdot \underline{y}_n & \underline{z}_0 \cdot \underline{z}_n \end{pmatrix} \triangleq \begin{pmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{pmatrix}$$

Pour réduire la redondance existant entre ces paramètres on peut utiliser les *cosinus directeurs partiels*.

Cosinus directeurs partiels : $X_R = (x_x, x_y, x_z, z_x, z_y, z_z)^t$

Nous avons choisi ici ceux qui représentent les coordonnées, dans la base \mathcal{B}_O , des vecteurs \underline{x}_n et \underline{z}_n de la nouvelle base \mathcal{B}_n (bien entendu d'autres choix sont possibles). On peut remarquer que les cosinus directeurs manquants s'expriment de manière très simple en fonction de ceux qui ont été retenus : $y_n = z_n \times x_n$. Ces paramètres ont tous un module inférieur à un. Ils ont encore le grand inconvénient d'être au nombre surabondant de six. Il existe toujours de nombreuses relations de dépendance entre ces six paramètres, dues au fait que la matrice R est orthogonale.

Rotation autour d'un vecteur \mathbf{r} passant par l'origine et d'angle θ : $R(\theta, \mathbf{r})$

Considérons une rotation d'angle θ autour d'un axe portant un vecteur unitaire \mathbf{r} passant par l'origine du repère R_i .

On définit un repère R_k de telle sorte que \mathbf{z}_k soit porté par l'axe de \mathbf{r} et que \mathbf{x}_k soit selon la perpendiculaire commune à \mathbf{z}_k et \mathbf{z}_i . On a : $R_{i,k} = R(\alpha, \mathbf{z}).R(\beta, \mathbf{x})$

Pour calculer $R(\theta, \mathbf{r})$ exprimant la rotation d'axe \mathbf{r} et d'angle θ on peut raisonner en se ramenant à des rotations élémentaires dans le repère de base.

Tourner autour de \mathbf{r} dans R_i est équivalent à tourner autour de \mathbf{z} dans R_k .

On a donc $R(\theta, \mathbf{r}).R_{i,k} = R_{i,k}.Rot(\mathbf{z}, \theta)$ avec $R_{i,k} = R(\alpha, \mathbf{z}).R(\beta, \mathbf{x})$

(voir figure 2.6 avec $T_{i,j} = T_{i',j'}$)

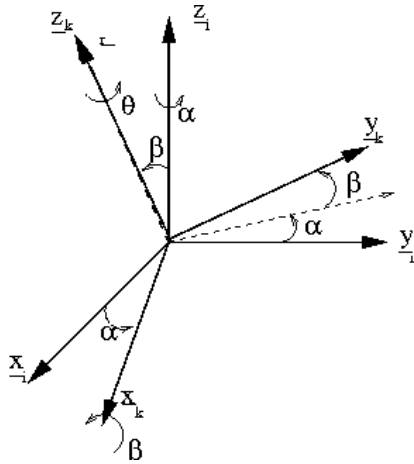


FIGURE 2.5 – Rotation pure de θ autour de \mathbf{r}

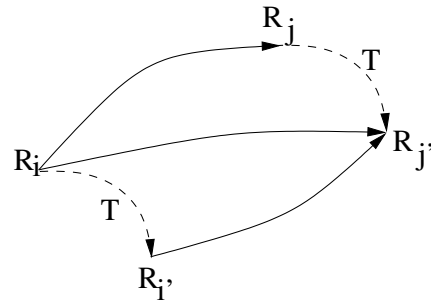


FIGURE 2.6 – Matrice $T_{i,j'} = T * T_{i',j'} = T_{i,j} * T$

donc $R(\theta, \mathbf{r}) = R(\alpha, \mathbf{z}).R(\beta, \mathbf{x}).R_z(\theta).R(-\beta, \mathbf{x}).R(-\alpha, \mathbf{z})$

En développant la formule on obtient :

$$R(\theta, \mathbf{r}) \triangleq \begin{pmatrix} r_x^2(1 - c\theta) + c\theta & r_x.r_y.(1 - c\theta) - r_z.s\theta & r_x.r_z.(1 - c\theta) + r_y.s\theta \\ r_x.r_y.(1 - c\theta) + r_z.s\theta & r_y^2(1 - c\theta) + c\theta & r_y.r_z.(1 - c\theta) - r_x.s\theta \\ r_x.r_z.(1 - c\theta) - r_y.s\theta & r_y.r_z.(1 - c\theta) + r_x.s\theta & r_z^2(1 - c\theta) + c\theta \end{pmatrix} \triangleq R(-\theta, -\mathbf{r})$$

On peut résoudre le problème inverse qui consiste à calculer l'angle et le vecteur unitaire de l'axe de rotation pour une matrice de rotation donnée.

$$\text{Soit } R = \begin{pmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{pmatrix} \text{ alors } \theta = \cos^{-1}\left(\frac{x_x + y_y + z_z - 1}{2}\right) \quad \text{Si } \sin\theta \neq 0 \text{ alors : } \mathbf{r} = \frac{1}{2\sin(\theta)} \cdot \begin{pmatrix} y_z - z_y \\ z_x - x_z \\ x_y - y_x \end{pmatrix}$$

sinon il faut écrire la relation particulière à cette valeur.

Il faut noter que pour $\theta = 0$ (rotation nulle), le vecteur unitaire \mathbf{r} est arbitraire (singularité).

Remarque : on exprime la rotation à l'aide de 4 paramètres qui sont liées par la relation $r_x^2 + r_y^2 + r_z^2 = 1$

Quaternion : H

Un quaternion $H \in \mathbb{H}$ peut se décrire comme un quadruplet de réels, le premier élément étant un scalaire et les trois autres formant un vecteur. $H = a + b.i + c.j + d.k$ avec $i^2 = j^2 = k^2 = i.j.k = -1$.

Dans cette représentation, l'orientation est exprimée par quatre paramètres qui décrivent une rotation unique équivalente de valeur θ ($0 \leq \theta \leq \frac{\pi}{2}$) autour d'un axe de vecteur unitaire \mathbf{r} .

On définit le quaternion par $H = (\eta, \epsilon)$ tel que :

$$\eta = \cos\left(\frac{\theta}{2}\right)$$

$$\epsilon = \sin\left(\frac{\theta}{2}\right) \cdot \mathbf{r}$$

On considère ici les quaternions unitaires encore appelés paramètres d'Euler ou paramètres d'Olinde-Rodrigues. La contrainte est : $\eta^2 + \epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2 = 1$

On obtient alors la matrice d'orientation :

$$R(\eta, \epsilon) \triangleq \begin{pmatrix} 2.(\eta^2 + \epsilon_x^2) - 1 & 2.(\epsilon_x \epsilon_y - \eta \epsilon_z) & 2.(\epsilon_x \epsilon_z + \eta \epsilon_y) \\ 2.(\epsilon_x \epsilon_y + \eta \epsilon_z) & 2.(\eta^2 + \epsilon_y^2) - 1 & 2.(\epsilon_y \epsilon_z - \eta \epsilon_x) \\ 2.(\epsilon_x \epsilon_z - \eta \epsilon_y) & 2.(\epsilon_y \epsilon_z + \eta \epsilon_x) & 2.(\eta^2 + \epsilon_z^2) - 1 \end{pmatrix}$$

On peut résoudre le problème inverse qui consiste à calculer la valeur scalaire et le vecteur du quaternion à partir de R .

$$\text{Soit } R = \begin{pmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{pmatrix}. \text{ Il existe des relations avec } z_x \pm x_z = \dots, x_y \pm y_x = \dots, y_z \pm z_y = \dots :$$

$$\begin{aligned} \frac{1}{4} \cdot (z_x + x_z) &= 4 \cdot \epsilon_x \cdot \epsilon_z & \frac{1}{4} \cdot (z_x - x_z) &= 4 \cdot \eta \cdot \epsilon_y \\ \frac{1}{4} \cdot (x_y + y_x) &= 4 \cdot \epsilon_x \cdot \epsilon_y & \frac{1}{4} \cdot (x_y - y_x) &= 4 \cdot \eta \cdot \epsilon_z \\ \frac{1}{4} \cdot (y_z + z_y) &= 4 \cdot \epsilon_y \cdot \epsilon_z & \frac{1}{4} \cdot (y_z - z_y) &= 4 \cdot \eta \cdot \epsilon_x \end{aligned}$$

Si on connaît une inconnue on peut résoudre le système.

Par combinaison des éléments de la diagonale on obtient :

$$\begin{aligned} \frac{1}{4} \cdot (x_x + y_y + z_z + 1) &= \eta^2 & \frac{1}{4} \cdot (x_x - y_y - z_z + 1) &= \epsilon_x^2 \\ \frac{1}{4} \cdot (-x_x + y_y - z_z + 1) &= \epsilon_y^2 & \frac{1}{4} \cdot (-x_x - y_y + z_z + 1) &= \epsilon_z^2 \end{aligned}$$

Implicitement $\eta \geq 0$, ce qui correspond à une valeur de rotation $\theta \in [-\pi, \pi]$.

En pratique on calcule la plus grande inconnue ($\eta, \epsilon_x, \epsilon_y, \epsilon_z$) puis on en déduit les autres.

Discussion : intérêt des quaternions?

2.3 Formules utiles

$$\cos(a + b) = \cos a \cdot \cos b - \sin a \cdot \sin b$$

$$\cos(a - b) = \cos a \cdot \cos b + \sin a \cdot \sin b$$

$$\sin(a + b) = \sin a \cdot \cos b + \cos a \cdot \sin b$$

$$\sin(a - b) = \sin a \cdot \cos b - \cos a \cdot \sin b$$

Nous représenterons souvent le produit vectoriel comme un produit d'une matrice (matrice de pré-produit vectoriel) par un vecteur :

$$\mathbf{X} \wedge \mathbf{Z} = \hat{\mathbf{X}} \cdot \mathbf{Z} = \begin{pmatrix} 0 & -x_z & x_y \\ x_z & 0 & -x_x \\ -x_y & x_x & 0 \end{pmatrix} \cdot \mathbf{Z} \quad \text{avec } \mathbf{X} = (x_x, x_y, x_z)^t$$

Chapitre 3

Modèle Géométrique Direct (MGD)

Rappel : Constitution d'un Bras Manipulateur

Un bras manipulateur, BM, peut se décrire par deux sous-ensembles :

- une Structure Mécanique Articulée (SMA) : assemblage de n corps rigides (*link*) reliés par des liaisons (*joint*). Chaque liaison est caractérisée par une coordonnée généralisée q_i (espace généralisé Q).
Son but est de pouvoir déplacer le dernier corps du robot (extrémité) dans une position et orientation désirée.
- un Organe Terminal (OT) : c'est le dernier corps du robot qui est destiné à manipuler l'outil (par exemple préhenseur de type pince ou effecteur de type torche de soudage,...). C'est l'interface qui permet au robot d'agir sur son environnement. Par abus de langage on parle parfois d'OT pour parler de l'outil puisque l'outil est fixé au dernier corps du robot. Bien souvent la position de l'OT ne correspond pas avec l'origine du dernier repère R_n , dans ce cas on introduit un point supplémentaire noté O_{n+1} .

Réaliser une tâche robotique consiste à décrire le mouvement de l'OT. Pour cela, il faut mettre en correspondance le repère attaché à l'OT ou à l'outil, avec le ou les repères associés à la tâche.

Pour faire bouger l'OT il faudra faire bouger les liaisons.



Objectif du MGD

On cherche le modèle permettant de passer de l'espace généralisé Q à l'espace opérationnel X . X peut décrire la situation de l'OT ou bien de l'outil.

On cherche à calculer

$$\mathbf{X} = F(\mathbf{q})$$

.

On va décomposer le calcul du MGD en deux étapes :

- Calcul de $T_{0,n}(\mathbf{q})$: ce calcul ne dépend que du robot
- Calcul de \mathbf{X} en fonction de $T_{0,n}(\mathbf{q})$: ne dépend que du choix des coordonnées opérationnelles

3.1 Calcul de $T_{0,n}$

Il faut écrire la matrice de passage entre le repère de base R_0 et le repère R_n .

Pour cela on va utiliser les changements successifs de repères attachés à chaque corps du robot.

Les principales étapes seront :

- Mettre en place les repères : on va placer les repères sur les différents corps du robot de manière à minimiser les calculs permettant de caractériser la situation d'un corps par rapport à son prédécesseur. Les paramètres de Denavit-Hartenberg modifiés (DHM) permettent d'avoir un paramétrage minimal.
- Ecrire les matrices de passage élémentaires : on va écrire les paramètres de DHM dans un tableau afin de rendre systématique (et de simplifier) le calcul des différentes matrices.
- Faire le produit des matrices de changement élémentaire de repère pour calculer $T_{0,n}$.

3.1.1 Introduction des repères

Il faut affecter à chaque corps du robot un repère affine orthonormé si possible de manière itérative pour rendre cette étape systématique.

Soit un robot manipulateur constitué de n corps rigides reliés par n liaisons.

3.1.1.1 Repère R_0

- Soit il est imposé
- sinon on choisit \mathbf{x}_0 perpendiculaire à la liaison L_1 et \mathbf{z}_0 porté par l'axe de la liaison.

3.1.1.2 Repère R_{i-1} pour $i = 2, 3, \dots, n$

- 0_{i-1} est le pied de la perpendiculaire commune Δ à L_{i-1} et L_i et $\in L_{i-1}$.
- \mathbf{x}_{i-1} est le vecteur unitaire de Δ orienté de L_{i-1} vers L_i .
- \mathbf{z}_{i-1} est porté par l'axe de la liaison arbitrairement orienté.
- $\mathbf{y}_{i-1} = \mathbf{z}_{i-1} \otimes \mathbf{x}_{i-1}$

3.1.1.3 Repère R_n

- O_n est choisi arbitrairement sur L_n .
- \mathbf{z}_n est porté par L_n
- \mathbf{x}_n est choisi, si possible, dans le plan défini par l'axe de la liaison L_n et par le point O_{n+1} (point particulier de l'OT).

Cas particuliers : voir tableau

La méthode consiste à placer tous les 0_{i-1} , tous les \mathbf{x}_{i-1} puis tous les \mathbf{z}_{i-1} .

Grâce à cette convention de placement des repères, la situation du corps C_i par rapport au corps C_{i-1} est définie à l'aide de 4 paramètres uniquement (au lieu de 6 dans le cas général).

Ces paramètres sont appelés les paramètres de DENAVIT-HARTENBERG modifiés (DHM). On a donc un paramétrage minimal et systématique.

3.1.2 Paramètres de DHM

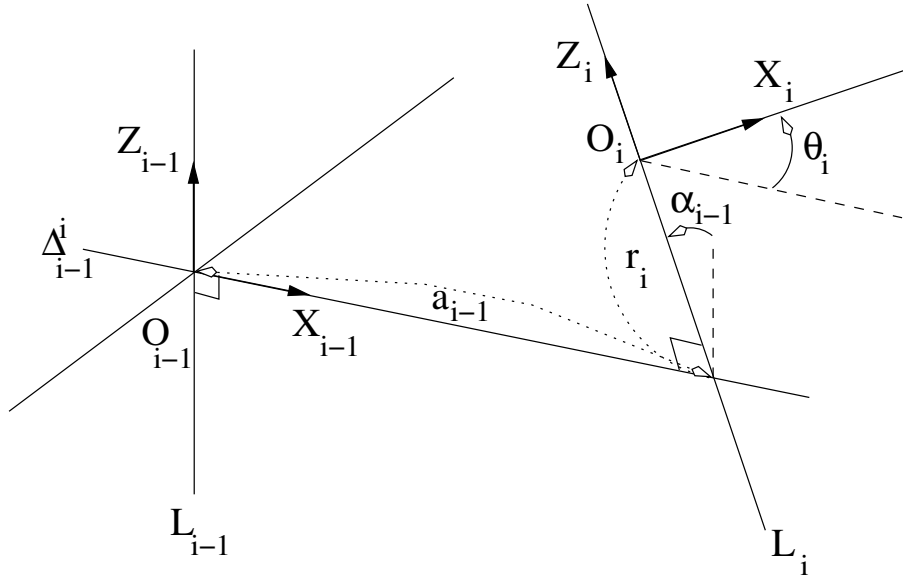


FIGURE 3.1 –

$$a_{i-1} \triangleq \overrightarrow{O_{i-1}O_i} \cdot X_{i-1} \quad \alpha_{i-1} \triangleq \widehat{Z_{i-1}, Z_i} / X_{i-1} \quad r_i \triangleq \overrightarrow{O_{i-1}O_i} \cdot Z_i \quad \theta_i \triangleq \widehat{X_{i-1}, X_i} / Z_i$$

$$T_{i-1, a_{i-1}} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{a_{i-1}, \alpha_{i-1}} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{\alpha_{i-1}, r_i} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{r_i, i} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{i-1, i} = T_{i-1, a_{i-1}} \times T_{a_{i-1}, \alpha_{i-1}} \times T_{\alpha_{i-1}, r_i} \times T_{r_i, i}$$

Nous obtenons alors la matrice générique suivante (changement de repère entre deux repères consécutifs) :

$$T_{i-1, i} = \left(\begin{array}{ccc|c} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \cos(\alpha_{i-1}) \cdot \sin(\theta_i) & \cos(\alpha_{i-1}) \cdot \cos(\theta_i) & -\sin(\alpha_{i-1}) & -r_i \cdot \sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1}) \cdot \sin(\theta_i) & \sin(\alpha_{i-1}) \cdot \cos(\theta_i) & \cos(\alpha_{i-1}) & r_i \cdot \cos(\alpha_{i-1}) \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

On représente l'ensemble des paramètres dans un tableau :

Corps i	1	...	i	...	n
σ_i					
a_{i-1}			$d(L_{i-1}, L_i) = \mathbf{O}_{i-1} \mathbf{O}_i \cdot \mathbf{x}_{i-1}$		
α_{i-1}			$\text{angle}(\mathbf{z}_{i-1}, \mathbf{z}_i) / \mathbf{x}_{i-1}$		
r_i			$\mathbf{z}_i \cdot \mathbf{O}_{i-1} \mathbf{O}_i$		
θ_i			$\text{angle}(\mathbf{x}_{i-1}, \mathbf{x}_i) / \mathbf{z}_i$		
Valeur q_i figure					

- a_{i-1} et α_{i-1} sont les paramètres de forme (constantes)
- r_i et θ_i sont les paramètres de mouvement (un seul est constant) :
 - si $L_i = P$ alors $q_i = r_i$ et $\theta_i = \text{constante}$
 - si $L_i = R$ alors $q_i = \theta_i$ et $r_i = \text{constante}$

A l'aide du symbole binaire σ_i ($\sigma_i = 1/0$ pour une liaison P / R) on peut écrire : $q_i = \sigma_i r_i + \bar{\sigma}_i \theta_i$

3.1.3 Matrice $T_{i-1,i}$

A l'aide de la matrice générale de passage entre repères associées aux paramètres de DHM écrire les n matrices $T_{i-1,i}$.

Pour la configuration de la figure vérifier chaque matrice. Cette étape permet de trouver la quasi totalité des erreurs (placement de repères, écriture du tableau ou de la matrice).

3.1.4 Calculer $T_{0,n}$

Par définition $T_{0,n}(\mathbf{q}) = T_{0,1}(q_1) \cdot T_{1,2}(q_2) \cdot \dots \cdot T_{n-1,n}(q_n)$

$$T_{0,n}(\mathbf{q}) = \left(\begin{array}{cccc} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ \hline 0 & 0 & 0 & 1 \end{array} \right) = \left(\begin{array}{ccc|c} R_{0,n} & & & P_{0,n} \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

- On ne calcule pas la deuxième colonne de $T_{0,n}$
- On calcule les produits de la droite vers la gauche
- On regroupe si possible les rotations consécutives
- Dès qu'un coefficient apparaît on le mémorise dans une variable intermédiaire D_i

Vérifier le matrice $T_{0,n}$ pour la configuration de la figure.

A ce stade, on a calculé la situation du repère R_n en fonction de la valeur de **toutes** les coordonnées généralisées q_i . Il ne reste plus qu'à calculer la transformation rigide permettant de passer du repère R_n au repère de l'OT.

3.2 Calcul de X

$\mathbf{X} = \begin{pmatrix} \mathbf{X}_p \\ \mathbf{X}_r \end{pmatrix}$ On va calculer \mathbf{X} (situation de l'OT) en fonction de $T_{0,n}$

3.2.1 Calcul de \mathbf{X}_p

Position du point O_{n+1} dans le repère R_0 .

Projection de $\underline{\mathbf{O}_0\mathbf{O}_{n+1}}$ dans R_0 .

$$\mathbf{P}_{0,n+1} = \begin{pmatrix} \underline{\mathbf{O}_0\mathbf{O}_{n+1}} \cdot \mathbf{X}_0 \\ \underline{\mathbf{O}_0\mathbf{O}_{n+1}} \cdot \mathbf{Y}_0 \\ \underline{\mathbf{O}_0\mathbf{O}_{n+1}} \cdot \mathbf{Z}_0 \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{X}_p$$

$$\underline{\mathbf{O}_0\mathbf{O}_{n+1}} = \underline{\mathbf{O}_0\mathbf{O}_n} + \underline{\mathbf{O}_n\mathbf{O}_{n+1}}$$

Si on projette cette relation dans la base initiale on obtient :

$$\underline{\mathbf{O}_0\mathbf{O}_{n+1}}_{(0)} = \underline{\mathbf{O}_0\mathbf{O}_n}_{(0)} + \underline{\mathbf{O}_n\mathbf{O}_{n+1}}_{(0)} = \underline{\mathbf{O}_0\mathbf{O}_n}_{(0)} + R_{0,n} \cdot \underline{\mathbf{O}_n\mathbf{O}_{n+1}}_{(n)}$$

$$\underline{\mathbf{O}_0\mathbf{O}_{n+1}}_{(0)} \triangleq \mathbf{P}_{0,n+1} = \mathbf{P}_{0,n} + \underline{\mathbf{O}_n\mathbf{O}_{n+1}}_{(0)} = \mathbf{P}_{0,n} + R_{0,n} \cdot \underline{\mathbf{O}_n\mathbf{O}_{n+1}}_{(n)}$$

Si on a choisi le point O_{n+1} dans la plan $(\mathbf{x}_n, \mathbf{z}_n)$ on a : $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} t_{14} \\ t_{24} \\ t_{34} \end{pmatrix} + \begin{pmatrix} t_{11} & . & t_{13} \\ t_{21} & . & t_{23} \\ t_{31} & . & t_{33} \end{pmatrix} \cdot \begin{pmatrix} a_n \\ 0 \\ r_{n+1} \end{pmatrix}$

sinon il faut calculer entièrement la matrice de rotation.

$\mathbf{O}_{0,n+1}$ est le vecteur des coordonnées cartésiennes de position. On peut exprimer le vecteur \mathbf{X}_p sous une autre forme de coordonnée par un changement de coordonnées.

3.2.2 Calcul de \mathbf{X}_r

Sauf cas particulier, l'orientation du repère de l'OT est l'orientation du repère R_n . Les Cosinus directeurs partiels sont contenus dans la matrice $R_{0,n} : (t_{11}, t_{21}, t_{31}, t_{13}, t_{23}, t_{33})$. On peut exprimer le vecteur \mathbf{X}_r sous une autre forme de coordonnée par un changement de coordonnées (angles d'Euler, quaternion,...).

Résumé

- Mettre en place les repères (en respectant les règles!)
- Ecrire le tableau des paramètres de DHM
- Ecrire les matrices élémentaires $T_{i-1,i}$. Vérifier sur la figure.
- Calculer $T_{0,n}$ par le produit de gauche à droite des matrices élémentaires. Vérifier sur la figure.
- Calculer \mathbf{X}_p vecteur des coordonnées cartésiennes de position.
- Calculer \mathbf{X} en fonction des coordonnées opérationnelles demandées.

3.3 Aspect pratique

Il faut savoir que pour certaines tâches il est nécessaire de définir un repère outil particulier. Il faut alors rajouter une transformation rigide (liaison fixe, sans ddl) entre le repère de l'OT et le repère de l'outil.

La situation de l'outil étant connue dans le repère R_{n+1} il suffit d'effectuer un calcul similaire à 3.2 en prenant en compte la situation du repère outil par rapport à R_n . Ce calcul est obligatoire lorsque le robot doit changer automatiquement d'outil lors de la réalisation de sa tâche.

Si les contraintes de temps de calcul ne sont pas fortes, on peut calculer numériquement $T_{0,n}$ par le produit des matrices élémentaires.

3.4 Représentation URDF pour la modélisation des robots

URDF : Universal Robotic Description Format. C'est le format qui s'est imposé avec l'environnement ROS (Robotic Operating System).

Dans ce format on ne définit pas de règle de placement des repères. Le robot est représenté pour la configuration nulle ($\forall i, q_i = 0$) et les repères sont placés pour cette configuration

Un fichier URDF est de type XML et il décrit la chaîne cinématique du robot. Il contient :

- la description des corps (*link*)
 - le nom du corps
 - paramètres inertiels (masse, centre de gravité, inertie...)
 - paramètres de visualisation (fichier 3D...)
 - paramètres de collision (pour le calcul de distance)
 - la description des articulations (*joint*)
 - le nom de l'articulation
 - le corps précédent et courant (*parent* et *child*)
 - la transformation entre les corps *parent* et *child*
 - le type de liaison (rotation, translation)
 - l'axe (ou les axes) de la liaison
 - les limites articulaires (position, vitesse, couple)
- et plusieurs autres informations

Les transformations sont représentées par 6 paramètres :

- translation en x,y,z (xyz)
- rotation par trois angles Roll, Pitch, Yaw (rpy)

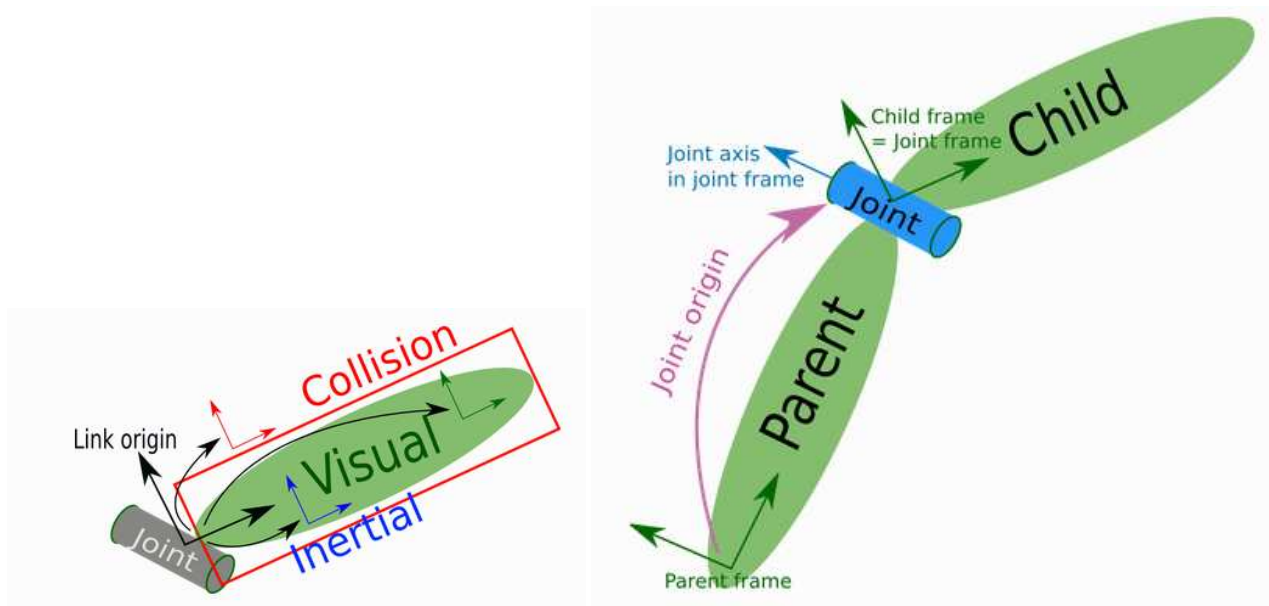


FIGURE 3.2 – ©Copyright 2018, Abed

Détails sur : https://abedgnu.github.io/Notes-ROS/chapters/ROS/10_robot_modeling/urdf.html

Pour les *joint*, la transformation qui correspond à *origin* est la transformée du corps *parent* vers le corps *child*. Le référentiel du corps *child* est localisé à l'origine du référentiel *joint*.

Exemple : robot RRP

```

<robot name="RRP">
  <link name="base_link">
    ....
  </link>

  <link name="link_1">
    <inertial>
      <origin xyz="0 0 0.5" rpy="0 0 0"/>
      <mass value="1"/>
      <inertia ixx="100" ixy="0" ixz="0" iyy="100" iyz="0" izz="100" />
    </inertial>

    <visual>
      <origin xyz="0 0 0" rpy="0 0 0" />
      <geometry>
        <box size="1 1.2 1" />
      </geometry>
      <material name="Cyan">
        <color rgba="0 1.0 1.0 1.0"/>
      </material>
    </visual>

    <collision>
      <origin xyz="0 0 0" rpy="0 0 0"/>
      <geometry>
        <cylinder radius="1" length="0.5"/>
      </geometry>
    </collision>
  </link>

  <link name="link_2">
    .....
  </link>

  <link name="link_3">
    .....
  </link>

  <joint name="link1_to_base" type="revolute">
    <parent link="base_link"/>
    <child link="link_1"/>
    <origin xyz="0 0 0.129" rpy="0 0 0"/>
    <axis xyz="0 0 1" />
    <limit effort="300" velocity="0.1" lower="-3.14" upper="3.14"/>
  </joint>

  <joint name="link2_to_link1" type="revolute">
    <parent link="link_1"/>
    <child link="link_2"/>
    <origin xyz="0.120 0 0" rpy="0 0 0"/>
    <axis xyz="0 0 1" />
    <limit effort="300" velocity="0.1" lower="-3.14" upper="3.14"/>
  </joint>

  <joint name="link3_to_link2" type="prismatic">
    <parent link="link_2"/>
    <child link="link_3"/>
    <origin xyz="0.130 0 0"/>
    <axis xyz="0 0 1" />
    <limit effort="300" velocity="0.1" lower="-0.150" upper="0"/>
  </joint>
</robot>

```

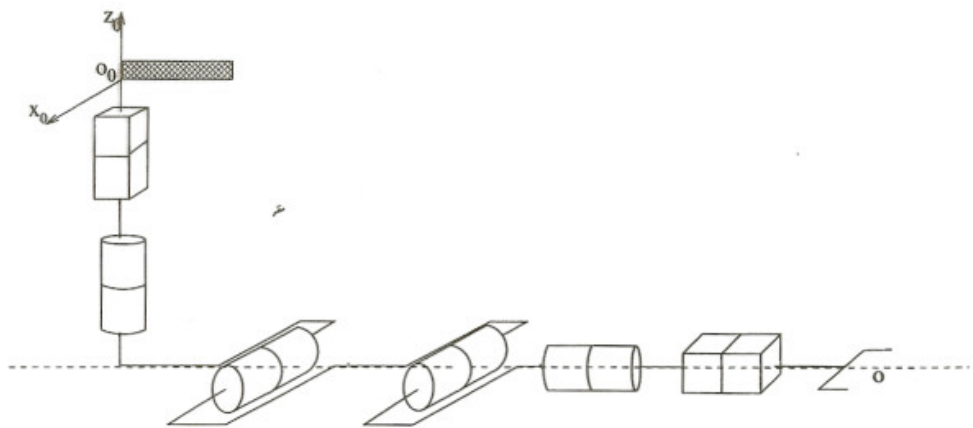


FIG. 1 – ROBOT PRRRRP

FIG. 1 – ROBOT PRRRRP

Paramètres de Denavit-Hartenberg :

	1	2	3	4	5	6
σ_i						
α_{i-1}						
a_{i-1}						
θ_i						
r_i						
$q_i(\text{figure})$						

$$T_{01} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{12} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{23} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{34} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{45} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{56} = \left(\begin{array}{ccc|c} & & & \\ & & & \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

Calcul de $T_{0,6}$ pour le robot PRRRRP.

$$T_{46} = \left(\begin{array}{ccc|c} c5 & -s5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s5 & c5 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \cdot \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_6 \\ \hline 0 & 0 & 0 & 1 \end{array} \right) = \left(\begin{array}{ccc|c} c5 & . & 0 & 0 \\ 0 & . & -1 & -q_6 \\ s5 & . & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{36} = \left(\begin{array}{ccc|c} c4 & -s4 & 0 & a_3 \\ s4 & c4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \cdot T_{46} = \left(\begin{array}{ccc|c} c4.c5 = D_1 & . & s4 & s4.q_6 + a_3 = D_3 \\ s4.c5 = D_2 & . & -c4 & -c4.q_6 = D_4 \\ s5 & . & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{26} = \left(\begin{array}{ccc|c} c3 & -s3 & 0 & a_2 \\ 0 & 0 & -1 & 0 \\ s3 & c3 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \cdot T_{36} = \left(\begin{array}{ccc|c} c3.D_1 - s3.D_2 = D_5 & . & s_{34} = s3.c4 + s4.c3 & D_7 = c3.D_3 - s3.D_4 + a_2 \\ -s5 & . & 0 & 0 \\ s3.D_1 + c3.D_2 = D_6 & . & -c_{34} = -c3.c4 + s3.s4 & D_8 = s3.D_3 + c3.D_4 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{16} = \left(\begin{array}{ccc|c} c2 & -s2 & 0 & 0 \\ s2 & c2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \cdot T_{26} = \left(\begin{array}{ccc|c} c2.D_5 + s2.s_5 & x & c2.s_{34} & c2.D_7 \\ s2.D_5 - c2.s_5 & x & s2.s_{34} & s2.D_7 \\ D_6 & x & -c_{34} & D_8 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$T_{06} = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_1 \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \cdot T_{16} = \left(\begin{array}{ccc|c} c2.D_5 + s2.s_5 & x & c2.s_{34} & c2.D_7 \\ s2.D_5 - c2.s_5 & x & s2.s_{34} & s2.D_7 \\ D_6 & x & -c_{34} & D_8 + q_1 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

Calcul de $\underline{X_p}$, et $\underline{X_R}$ pour le robot PRRRRP

$\underline{X_R}$: la matrice T_{06} contient les cosinus directeurs partiels.

$$\underline{X_p} : \underline{O_0 O_7} = \underline{O_0 O_6} + \underline{O_6 O_7}$$

$$\underline{O_0 O_{7(O)}} = \underline{O_0 O_{6(O)}} + R_{0,6} \cdot \underline{O_6 O_{7(6)}}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} c2.D_7 \\ s2.D_7 \\ D_8 + q_1 \end{pmatrix} + \begin{pmatrix} t_{11} & . & t_{13} \\ t_{21} & . & t_{23} \\ t_{31} & . & t_{33} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ r_7 \end{pmatrix} = \begin{pmatrix} c2.D_7 + c2.s_{34}.r_7 \\ s2.D_7 + c2.s_{34}.r_7 \\ D_8 + q_1 - c_{34}.r_7 \end{pmatrix}$$

Algorithme de calcul du MGD pour le robot PRRRRP

Data: lecture des 6 valeurs des codeurs q_i

Result: Calcul du MGD : $\underline{X_p}^t = (x, y, z)$, $\underline{X_R}^t = (x_x, x_y, x_z, z_x, z_y, z_z)$

for $i=2$ à 5 **do**

$ci = \cos(q_i)$;

$si = \sin(q_i)$;

end

$$D_1 = c4.c5;$$

$$D_2 = s4.c5;$$

$$D_3 = s4.q_6 + a_3;$$

$$D_4 = -c4.q_6;$$

$$D_5 = c3.D_1 - s3.D_2;$$

$$D_6 = s3.D_1 + c3.D_2;$$

$$D_7 = c3.D_3 - s3.D_4 + a_2;$$

$$D_8 = s3.D_3 + c3.D_4;$$

$$s_{34} = s3.c4 + s4.c3;$$

$$c_{34} = c3.c4 - s3.s4;$$

Position :

$$x = c2.(D_7 + s_{34}.r_7);$$

$$y = s2.(D_7 + s_{34}.r_7);$$

$$z = D_8 + q_1 - c_{34}.r_7$$

Orientation :

$$x_x = c2.D_5 + s2.s5;$$

$$x_y = s2.D_5 - c2.s5;$$

$$x_z = D_6;$$

$$z_x = c2.s_{34};$$

$$z_y = s2.s_{34};$$

$$z_z = -c_{34}$$

Chapitre 4

Modèle Géométrique Inverse (MGI)

4.1 Méthodes analytiques

4.2 Méthodes de calcul

4.3 Principaux types d'équations associées au MGI

Introduction au MGI

La définition d'une tâche robotique en situation se fait dans l'espace de la tâche (opérationnel) en décrivant les $\mathbf{X} = (X_p, X_r)^t$ à atteindre. Il faut pouvoir calculer la(s) configuration(s) correspondante(s)

Résolution du MGI : trouver \mathbf{q} solution de $\mathbf{X} = f(\mathbf{q})$ pour \mathbf{X} donnée.

Il faut résoudre

$$\mathbf{q} = g(\mathbf{X})$$

Problème : $f(\mathbf{q})$ est une fonction non-linéaire

Problème non-linéaire

- * existence ou non de solution (workspace)
- * une ou plusieurs solutions pour une même situation \mathbf{X}
- * méthodes de résolution analytique ou numérique

Exemples : tableau

Résolution analytique

- * Toutes les solutions sont calculées
- * Implémentation temps réel
- * Facile à résoudre pour des robots simples (industriels, poignets axes concourants)
- * Complexe voire impossible, pour robots quelconques.

Résolution numérique

- * Méthode générale qui s'applique à tous les robots
- * Une seule solution est calculée (fonction des conditions initiales)
- * Problème de convergence de l'algorithme d'optimisation
- * Temps de calcul.

Pas de résultat général sur le nombre de solution ou la possibilité de trouver une solution analytique

Un bras manipulateur est **résoluble** s'il est possible de calculer toutes les configurations q permettant d'atteindre une situation X.

Tous les BM série avec $n = 6$, non redondants, sont considérés comme résolubles.

Le nombre de solution dépend de l'architecture du BM :

- * pour un robot 6R ou 5R1P, le nombre maximum de MGI est égal à 16,
- * pour un robot avec 3 liaisons rotoïdes d'axes concourants, ce nombre maximum est égal à 8,
- * pour un robot 3P3R, ce nombre maximum est égal à 2

Remarques : Si $n < 6$, on ne peut pas décrire librement la situation de l'OT sinon il peut ne pas y avoir de solution au calcul du MGI. (exemple : robot SCARA).

Si $\dim(X) < 6$, on a des contraintes sur la situation de l'OT, il faudra penser à introduire ces contraintes dans le choix de la situation à atteindre X.

Dans le calcul du MGI, trois cas peuvent se présenter :

- * Absence de solution : lorsque la situation est inaccessible pour le bras manipulateur (en dehors du volume de travail, butées articulaires,...)
- * Infinité de solution : le robot est redondant (globalement ou par rapport à la tâche robotique) ou le robot se trouve dans une configuration singulière
- * Nombre fini de solutions

Méthode analytique

On veut calculer \mathbf{q} solution de $\mathbf{X}^* = f(\mathbf{q})$ pour \mathbf{X}^* donnée.

A une situation \mathbf{X}^* de l'OT donnée correspond une matrice $T_{0,n}(\mathbf{X}^*)$

On va calculer \mathbf{q} tel que :

$$T_{0,n}(\mathbf{X}^*) = T_{0,n}(\mathbf{q})$$

Il existe plusieurs approches différentes (méthode de Paul, Pieper, Rahavan et Roth,.....) qui vont toutes se ramener à écrire un système d'équations non linéaires pour ensuite le résoudre.

Le calcul se déroule en trois étapes :

- Ecriture de $T_{0,n}(\mathbf{X}^*)$ en fonction de la situation désirée \mathbf{X}^* . Cette étape est totalement indépendante du BM considéré.
- Ecriture du système d'équations non linéaires liant \mathbf{X}^* à \mathbf{q} . Il est toujours possible d'écrire le système pour un bras manipulateur donné.
- Résolution du système d'équations non linéaires. Cette résolution peut échouer car elle est liée au SMA constituant le BM

Simplification : $T_{0,n}(\mathbf{X}^*) = T_{0,n}^*$

a - Expression de $T_{0,n}^*$

Ecrire $T_{0,n}^*$ en fonction de $\mathbf{X}^* \implies T_{0,n}^* = \left(\begin{array}{ccc|c} & R_{0,n} & & P_{0,n} \\ 0 & 0 & 0 & 1 \end{array} \right)$

Calcul de $R_{0,n}$

En fonction de la paramétrisation de l'orientation choisie (angles de bryant,), on obtient $R_{0,n}$ en appliquant directement les formules de changement de paramétrisation pour obtenir les cosinus directeurs constituant $R_{0,n}$.

Calcul de $P_{0,n}$

On connaît la position désirée du point O_{n+1} représentant l'OT mais on ne connaît pas les coordonnées du point O_n . On a vu (MGD) que : $\underline{O_0 O_{n+1}} = \underline{O_0 O_n} + \underline{O_n O_{n+1}}$
 $P_{0,n+1} = \underline{O_0 O_{n+1}}_{(0)} = \underline{O_0 O_n}_{(0)} + R_{0,n} \cdot \underline{O_n O_{n+1}}_{(n)} = \underline{P_{o,n}} + R_{o,n} \cdot \underline{O_n O_{n+1}}_{(n)}$

On a donc $\underline{P_{o,n}} = \underline{P_{0,n+1}} - R_{o,n} \cdot \underline{O_n O_{n+1}}_{(n)}$

On connaît donc $P_{0,n}$ et $R_{0,n}$ ($T_{0,n}^*$) indépendamment du robot.

b - Ecriture du système d'équations

Il faut donc résoudre : $T_{0,n}^* = T_{0,n}(\mathbf{q}) = T_{0,1}(q_1) \cdot T_{1,2}(q_2) \dots T_{(n-1),n}(q_n)$
 $T_{0,n}(\mathbf{q})$ a des coefficients non-linéaires d'une grande complexité et non explicites
difficile à résoudre car toutes les inconnues q_i sont liées \Rightarrow répartition des
inconnues q_i dans l'équation matricielle afin de simplifier les équations :

$$T_{p,p-1}(q_p) \dots T_{1,0}(q_1) \cdot T_{0,n}^* = T_{p,p-1}(q_p) \dots T_{1,0}(q_1) \cdot T_{0,n}(\mathbf{q})$$

p inconnues / (n - p) inconnues

et $T_{p,p-1}(q_p) \dots T_{1,0}(q_1) \cdot T_{0,n}(\mathbf{q}) = T_{p,p+1}(q_{p+1}) \dots T_{n-1,n}(q_n)$ (calculé par MGD)

Système d'équations à résoudre

$$T_{p,p-1}(q_p) \dots T_{1,0}(q_1) \cdot T_{0,n}^* = T_{p,p+1}(q_{p+1}) \dots T_{n-1,n}(q_n)$$

On a simplifier les équations en répartissant les inconnues q_i .

c - Résolution

On écrit un système de 9 équations à n inconnues à résoudre.

Le système est souvent plus simple si $p = \text{Int}(\frac{n}{2})$

On ne peut pas résoudre le système en une fois :

- * on cherche une variable q_i , « la plus simple », qui va devenir une constante dans les autres équations, et ainsi de suite.
- * on combine les équations pour isoler un q_i ou un \cos ou $\sin(q_i)$
- * toutes les équations doivent être utilisées/vérifiées.

Une équation du type $\alpha \cos(q) + \beta \sin(q) = \gamma$ admet une solution si et seulement si $\alpha^2 + \beta^2 \geq \gamma^2$

$$\sin(q) = \frac{\beta \cdot \gamma + \epsilon \alpha \cdot \sqrt{\alpha^2 + \beta^2 - \gamma^2}}{\alpha^2 + \beta^2} \quad \cos(q) = \frac{\alpha \cdot \gamma - \epsilon \beta \cdot \sqrt{\alpha^2 + \beta^2 - \gamma^2}}{\alpha^2 + \beta^2}$$

avec $\epsilon = \pm 1$

Principaux types d'équations associées au MGI

$$X \cdot \cos(q_i) + Y \cdot \sin(q_i) = Z \quad (1)$$

Quatre cas se présentent :

Si $X = 0$ et $Y \neq 0$ alors $\sin(q_i) = s_i = \frac{Z}{Y}$. On obtient $q_i = \text{Atan2}(s_i, \pm \sqrt{1 - (s_i)^2})$

Si $Y = 0$ et $X \neq 0$ alors $\cos(q_i) = c_i = \frac{Z}{X}$. On obtient $q_i = \text{Atan2}(\pm \sqrt{1 - (c_i)^2}, c_i)$

Si $X \neq 0$ et $Y \neq 0$ et $Z = 0$ alors $q_i = \text{Atan2}(-X, Y)$ et $q'_i = q_i + \Pi$

Si $X \neq 0$ et $Y \neq 0$ et $Z \neq 0$, l'équation $X \cdot \cos(q_i) + Y \cdot \sin(q_i) = Z$ admet une solution \Leftrightarrow

$X^2 + Y^2 \geq Z^2$. Alors $\sin(q_i) = \frac{Y \cdot Z + \epsilon \cdot X \cdot \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2}$ et $\cos(q_i) = \frac{X \cdot Z - \epsilon \cdot Y \cdot \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2}$

avec $\epsilon = \pm 1$

$q_i = \text{Atan2}(s_i, c_i)$ (liaison rotoïde)

$$X_1 \cdot \sin(q_i) + Y_1 \cdot \cos(q_i) = Z_1 \text{ et } X_2 \cdot \sin(q_i) + Y_2 \cdot \cos(q_i) = Z_2 \quad (2)$$

Si $X_1 \cdot Y_2 - X_2 \cdot Y_1 \neq 0$ alors $\sin(q_i) = \frac{Z_1 \cdot Y_2 - Z_2 \cdot Y_1}{X_1 \cdot Y_2 - X_2 \cdot Y_1}$ et $\cos(q_i) = \frac{Z_2 \cdot X_1 - Z_1 \cdot X_2}{X_1 \cdot Y_2 - X_2 \cdot Y_1}$

$q_i = \text{Atan2}(s_i, c_i)$ (liaison rotoïde)

Sinon les deux équations sont dépendantes. On choisit une équation qui est du type (1)

$$X_1 \cdot q_j \cdot \sin(q_i) = Y_1 \text{ et } X_2 \cdot q_j \cdot \sin(q_i) = Y_2 \quad (3)$$

$q_j = \pm \sqrt{(Y_1/X_1)^2 + (Y_2/X_2)^2}$ (liaison prismatique)

$q_i = \text{Atan2}((Y_1/(X_1 \cdot q_j)), (Y_2/(X_2 \cdot q_j)))$ (liaison rotoïde)

$$X_1 \cdot \sin(q_i) = Y_1 + Z_1 \cdot q_j \text{ et } X_2 \cdot \cos(q_i) = Y_2 + Z_2 \cdot q_j$$

On normalise les équations

On élève au carré les 2 équations puis on les additionne

On obtient un système du second degré en q_j (prismatique) que l'on cherche à résoudre

On calcule ensuite q_i (rotation) à l'aide du type (2).

$$W \cdot \sin(q_j) = X \cdot \cos(q_i) + Y \cdot \sin(q_i) + Z_1 \text{ et } W \cdot \cos(q_j) = X \cdot \sin(q_i) - Y \cdot \cos(q_i) + Z_2 \text{ (avec } Z_1 \neq 0 \text{ et/ou } Z_2 \neq 0)$$

On élève au carré et on fait la somme des 2 équations

On se ramène à une équation du type ?? dont la résolution donne q_j

Connaissant q_j , on résout le système en q_i par un système de type (2)

$$X \cdot \cos(q_i) + Y \cdot \cos(q_i + q_j) = Z_1 \text{ et}$$

$$X \cdot \sin(q_i) + Y \cdot \sin(q_i + q_j) = Z_2$$

Equations au carré et addition \Rightarrow élimine q_i .

On déduit $\cos(q_j) = c_j = \frac{Z_1^2 + Z_2^2 - X^2 - Y^2}{2 \cdot X \cdot Y}$ $q_j = \text{Atan2}(\pm \sqrt{1 - (c_j)^2}, c_j)$

On pose $B_1 = X + Y \cdot c_j$ et $B_2 = Y \cdot s_j$ et on a :

$$\sin(q_i) = s_i = \frac{B_1 \cdot Z_2 - B_2 \cdot Z_1}{B_1^2 + B_2^2} \text{ et } \cos(q_i) = c_i = \frac{B_1 \cdot Z_1 + B_2 \cdot Z_2}{B_1^2 + B_2^2}.$$

$$q_i = \text{Atan2}(s_i, c_i) \text{ (liaison rotoïde)}$$

$$W_1 \cdot \cos(q_j) + W_2 \cdot \sin(q_j) = X \cdot \cos(q_i) + Y \cdot \sin(q_i) + Z_1 \text{ et}$$

$$W_1 \cdot \sin(q_j) - W_2 \cdot \cos(q_j) = X \cdot \sin(q_i) - Y \cdot \cos(q_i) + Z_2$$

On élève au carré et on fait la somme des 2 équations

On se ramène à une équation du type (1) dont la résolution donne q_i

Connaissant q_i , on résout le système en q_j par un système de type (2)

Exemple : robot industriel RRPRRR

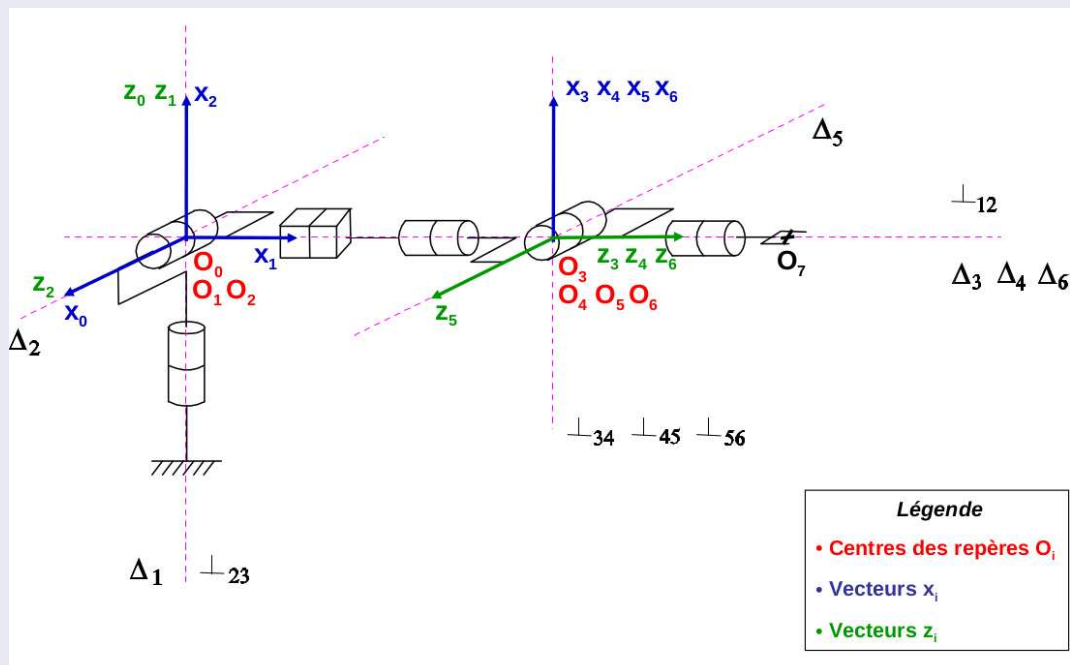


FIGURE – Placement des repères du RRPRRR

Chapitre 5

Modèle Différentiel Direct (MDD)

Le Modèle géométrique permet de passer de l'espace généralisé Q à l'espace opérationnel X et inversement. Ce sont des modèles indépendants du temps, le robot est statique.

On cherche maintenant à établir des modèles, cinématiques, qui vont faire intervenir le temps sans prendre en compte les effets de la dynamique (masses, frottements, inerties,...).

On peut distinguer :

- Modèle différentiel : $\underset{MDI}{\underset{\text{MDD}}{d\mathbf{q}}} \equiv d\mathbf{x}$
Différentielle des coord. configuration \Rightarrow Différentielle de coord. opérationnelles.
- Modèle cinématique : $\underset{MCI}{\underset{MCD}{\dot{\mathbf{q}}}} \equiv \dot{\mathbf{x}}$
vitesses généralisées \Rightarrow vitesse opérationnelles
- Modèle variationnel (approché) : $\underset{MVI}{\underset{MVD}{\delta\mathbf{q}}} \equiv \delta\mathbf{x}$
Variation infinitésimale $\delta\mathbf{q}$ autour de $\mathbf{q} \Rightarrow$ var. infinitésimale de $\delta\mathbf{x}$ autour de \mathbf{x} .

Utilité du MDD

- Connaître la vitesse du repère R_n ou de l'OT en fonction de $\dot{\mathbf{q}}$.
- Analyser le comportement du robot (singularités, domaine de travail,...).
- Commande (commande référencée tâche,...).
- Relation statique entre actions extérieures et les couples moteurs.

Exemple robot 2R

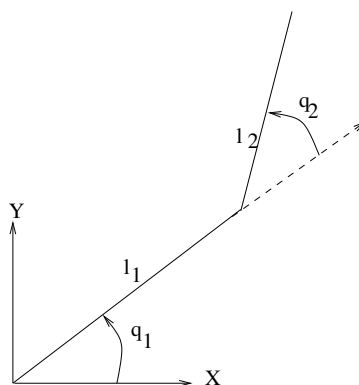


FIGURE 5.1 – Robot 2R dans le plan

Rappels/Compléments

Formule de la base mobile

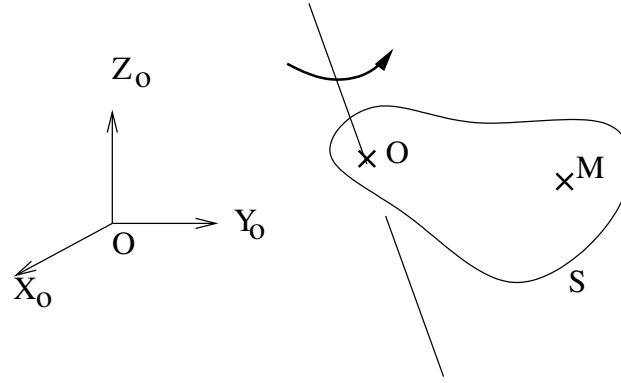


FIGURE 5.2 – Solide S en rotation autour d'un axe

Soit un solide S indéformable en mouvement par rapport à $R_0 = (O_0, \mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$. Soit deux points du solide O et M . Soit une rotation de S d'angle Ψ laissant O invariant ($O \in$ axe de rotation), alors la rotation transforme M en M' .

\mathbf{OM} est perpendiculaire à $\frac{d(\mathbf{OM})}{dt} \Rightarrow \exists \quad \Omega \quad | \quad \frac{d(\mathbf{OM})}{dt} = \Omega \wedge \mathbf{OM} = \hat{\Omega} \cdot \mathbf{OM}$ avec $\hat{\Omega}$ matrice anti-symétrique.
 Ω est le vecteur de rotation de S

Remarque : $\mathbf{OM} = \mathbf{OO}_0 + \mathbf{O}_0\mathbf{M} = \mathbf{O}_0\mathbf{M} - \mathbf{O}_0\mathbf{O} \Rightarrow \frac{d(\mathbf{OM})}{dt} = \Omega \wedge \mathbf{OM} = V(M) - V(O) \Rightarrow V(M) = V(O) + \Omega \wedge \mathbf{OM}$

Application

Soit un repère R' attaché à S : $(O', \mathbf{x}', \mathbf{y}', \mathbf{z}')$ alors nous avons : $\frac{d(\mathbf{x}')}{dt} = \Omega \wedge \mathbf{x}'$, $\frac{d(\mathbf{y}')}{dt} = \Omega \wedge \mathbf{y}'$, $\frac{d(\mathbf{z}')}{dt} = \Omega \wedge \mathbf{z}'$

Définition de Ω

Soit un repère R fixe lié à la scène et un repère R' attaché à S avec les origines et les axes z et z' confondus.

On fait tourner S d'un angle ψ autour de z (Δ axe de rotation) : le vecteur \mathbf{x}'_R devient $\begin{pmatrix} \cos \psi \\ \sin \psi \end{pmatrix} z'$ ne bouge pas lorsque S tourne

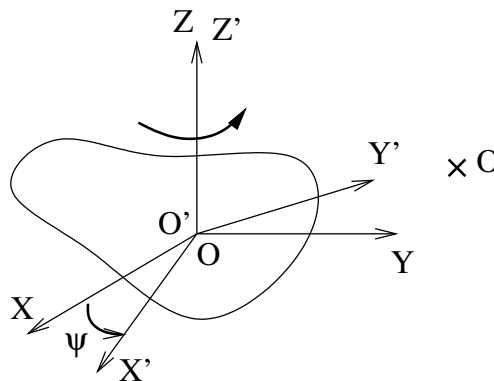


FIGURE 5.3 – Solide S en rotation autour d'un axe

autour de Δ : $\frac{d(\mathbf{z}')}{dt} = \Omega \wedge \mathbf{z}' = 0 \Rightarrow$ vecteurs équipollents : $\exists \quad w \quad | \quad \Omega = w \cdot \mathbf{z}'$

$\Rightarrow \frac{d(\mathbf{x}')}{dt} = \Omega \wedge \mathbf{x}' = w \cdot \mathbf{z}' \wedge \mathbf{x}' = w \cdot \mathbf{y}'$

or $\frac{d(\mathbf{x}')}{dt} = \begin{pmatrix} -\dot{\psi} \sin \psi \\ \dot{\psi} \cos \psi \end{pmatrix} = \dot{\psi} \cdot \mathbf{y}'$. On a donc $\dot{\psi} = w$ et $\Omega = \dot{\psi} \cdot \mathbf{z}'$

Rotation infinitésimale :

$\frac{d(\mathbf{OM})}{dt} = \Omega \wedge \mathbf{OM} = (\dot{\psi} \cdot \mathbf{z}') \wedge \mathbf{OM} = \frac{d\psi}{dt} \Rightarrow d\mathbf{OM} = d\psi \wedge \mathbf{OM}$

$d\psi$ est le vecteur de rotation infinitésimal identique pour tous les points du solide.

Nous pouvons combiner les rotations infinitésimales : $d\psi = d\psi_1 \cdot \mathbf{u}_1 + d\psi_2 \cdot \mathbf{u}_2$

Torseur cinématique

Soit un point O_i du corps B_i . Si \mathbf{v}_i représente la vitesse absolue de O_i par rapport au repère de base R_0 et si ω_i est le vecteur de rotation instantanée de B_i par rapport à R_0 alors on peut calculer la vitesse en tout point : $\mathbf{v}_p = \mathbf{v}_i + \omega_i \times \vec{O_i P}$

La concaténation de \mathbf{v}_i (moment résultant en O_i) et de ω_i (résultante) définit le torseur cinématique en O_i : $C_i = (\mathbf{v}_i^\top \ \omega_i^\top)^\top$

Matrice de transformation entre torseurs

Soit le torseur cinématique en O_i exprimés dans le repère R_i : $C_i = (\mathbf{v}_i^\top \ \omega_i^\top)^\top$

On veut calculer le torseur cinématique en O_j exprimés dans le repère R_j : $C_j = (\mathbf{v}_j^\top \ \omega_j^\top)^\top$

On a $\omega_i = \omega_j$ et $\mathbf{v}_j = \mathbf{v}_i + \omega_i \times \mathbf{P}_{ij}$ avec $\mathbf{P}_{ij} = \mathbf{O}_i \mathbf{O}_j$.

On peut écrire ces 2 équations sous forme : $\begin{bmatrix} v_j \\ \omega_j \end{bmatrix} = \begin{bmatrix} I_3 & -\hat{\mathbf{P}}_{ij} \\ 0_3 & I_3 \end{bmatrix} \cdot \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}$

Si on projette cette relation dans le repère i on obtient :

$$\begin{bmatrix} v_j \\ \omega_j \end{bmatrix}_i = \begin{bmatrix} I_3 & -\hat{\mathbf{P}}_{ij(i)} \\ 0_3 & I_3 \end{bmatrix} \cdot \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}_i$$

On obtient alors le torseur cinématique en O_j exprimés dans le repère R_j :

$$\begin{bmatrix} v_j \\ \omega_j \end{bmatrix}_j = \begin{bmatrix} R_{i,j} & 0_3 \\ 0_3 & R_{i,j} \end{bmatrix} \cdot \begin{bmatrix} I_3 & -\hat{\mathbf{P}}_{ij(i)} \\ 0_3 & I_3 \end{bmatrix} \cdot \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}_i = \begin{bmatrix} R_{i,j} & -R_{i,j} \cdot \hat{\mathbf{P}}_{ij(i)} \\ 0_3 & R_{i,j} \end{bmatrix} \cdot \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}_i$$

Dérivée d'une matrice de rotation

Soit $\mathbf{R} = \mathbf{R}(t)$. On sait que $\mathbf{R}(t) \cdot \mathbf{R}(t)^\top = \mathbf{I}$. En dérivant on obtient : $\dot{\mathbf{R}}(t) \cdot \mathbf{R}(t)^\top + \mathbf{R}(t) \cdot \dot{\mathbf{R}}^\top(t) = \mathbf{0}$

Posons $\mathbf{S}(t) = \dot{\mathbf{R}}(t) \cdot \mathbf{R}(t)^\top \implies \mathbf{S}(t) + \mathbf{S}(t)^\top = \mathbf{0}$

On obtient donc : $\dot{\mathbf{R}}(t) = \mathbf{S}(t) \cdot \mathbf{R}(t)$ (la dérivée de la matrice de rotation est fonction d'elle-même).

Interprétation : soit \mathbf{p}_a un vecteur fixe dans un repère a . Soit une matrice de rotation $\mathbf{R}(t)$.

$$\mathbf{p} = \mathbf{R}(t) \cdot \mathbf{p}_a \implies \dot{\mathbf{p}} = \dot{\mathbf{R}}(t) \cdot \mathbf{p}_a = \mathbf{S}(t) \cdot \mathbf{R}(t) \cdot \mathbf{p}_a$$

Soit $\omega(t)$ la vitesse angulaire de rotation du repère $\mathbf{R}(t)$ par rapport au repère de base alors :

$$\dot{\mathbf{p}} = \omega(t) \times \mathbf{p} \implies \dot{\mathbf{p}} = \omega(t) \times \mathbf{R}(t) \cdot \mathbf{p}_a$$

La matrice $\mathbf{S}(t)$ est la matrice de pré-produit vectoriel à partir de ω .

$$\text{Si } \omega = (\omega_x, \omega_y, \omega_z)^\top \text{ alors } \mathbf{S}(t) = \mathbf{S}(\omega(t)) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

On peut montrer que pour toute matrice de rotation \mathbf{R} on a : $\mathbf{R} \cdot \mathbf{S}(\omega) \cdot \mathbf{R}^\top = \mathbf{S}(\mathbf{R}\omega)$

5.1 Définition de la Jacobienne

Le MGD traduit la fonction : $\mathbf{q} = (q_1, \dots, q_n)^t \longrightarrow \mathbf{X} = (x_1, \dots, x_m)^t = f(\mathbf{q})$.

$$x_1 = f_1(q_1, \dots, q_n)$$

\vdots

$$x_m = f_m(q_1, \dots, q_n)$$

avec les f_i qui sont des fonctions non-linéaires complexes.

Le MDD apparaît alors comme la différentielle du MGD : $d\mathbf{q} \longrightarrow d\mathbf{x}$

$$dx_1 = \frac{\partial f_1}{\partial q_1} dq_1 + \dots + \frac{\partial f_1}{\partial q_n} dq_n$$

\vdots

$$dx_m = \frac{\partial f_m}{\partial q_1} dq_1 + \dots + \frac{\partial f_m}{\partial q_n} dq_n$$

Nous pouvons écrire sous forme matricielle les équations précédentes :

$$d\mathbf{X} = \begin{pmatrix} dx_1 \\ \vdots \\ dx_m \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial q_1} & \dots & \frac{\partial f_m}{\partial q_n} \end{pmatrix} \cdot \begin{pmatrix} dq_1 \\ \vdots \\ dq_n \end{pmatrix} = J(\mathbf{q}) \cdot d\mathbf{q} \text{ avec } J \text{ matrice jacobienne de } f \text{ en } \mathbf{q}$$

ou

$$\dot{\mathbf{X}} = J(\mathbf{q}) \cdot \dot{\mathbf{q}} = \begin{pmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial q_1} & \dots & \frac{\partial f_m}{\partial q_n} \end{pmatrix} \cdot \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{pmatrix}$$

Le calcul de $J(\mathbf{q})$ dépend explicitement de \mathbf{q} . Le calcul du MDD/MCD nécessite le calcul de $J(\mathbf{q})$ qui n'est pas toujours carrée, ni inversible.

- Si $\dot{\mathbf{X}}$ représente la vitesse de l'OT, on appelle la matrice $J(\mathbf{q})$ la jacobienne analytique car elle est la dérivée analytique du MGD. Elle est notée $J_a(\mathbf{q})$. Ce calcul est donc fonction du choix des coordonnées pour le MGD.
- Si $\dot{\mathbf{X}}$ représente la vitesse linéaire du point O_n et la vitesse de rotation du repère R_n par rapport au repère fixe R_0 on appelle la matrice $J(\mathbf{q})$ la jacobienne géométrique ou jacobienne.

5.2 Méthodes de calcul du MDD

- o Différentiation du MGD : très facile pour manipulateur simple (n petit), trop complexe si n grand
- o Méthode directe : il y a plusieurs manière de calculer $J(\mathbf{q})$

On va utiliser le même schéma que pour le MGD : on va séparer la partie robot du choix des coordonnées de l'OT On va calculer :

- la vitesse linéaire du point O_n et la vitesse de rotation du repère R_n dans le repère de base à partir de la jacobienne géométrique, J_g .
- la vitesse de l'OT en fonction de J_g et du choix des coordonnées de situation utilisé.

5.3 Jacobienne (géométrique)

On va calculer la vitesse linéaire du point O_n et la vitesse de rotation du repère R_n par rapport au repère fixe R_0 .

On veut donc calculer la vitesse opérationnelle $\dot{\mathbf{X}} = \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix}$ en fonction de $\dot{\mathbf{q}}$.

5.3.1 Relations cinématique pour une liaison

On considère que seule la liaison i bouge, les autres liaisons étant fixes. Quel est l'effet du mouvement de la liaison i sur les autres liaisons? .

La vitesse \dot{q}_i de la liaison i engendre une vitesse linéaire $\dot{\mathbf{p}}_i$ du point O_n et une vitesse de rotation $\boldsymbol{\omega}_i$ du repère R_n par rapport au repère fixe R_0

5.3.1.1 Liaison prismatique

On a : $\dot{\mathbf{p}}_i = \dot{q}_i \cdot \mathbf{z}_i$ et $\boldsymbol{\omega}_i = 0$

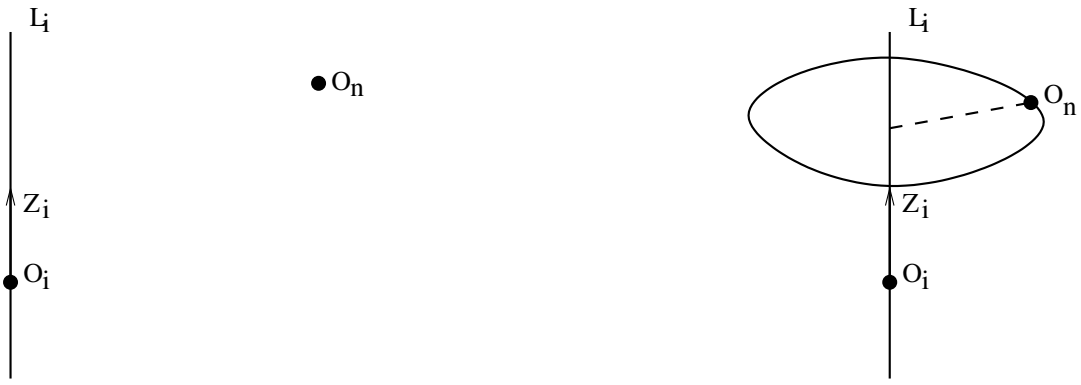


FIGURE 5.4 – Effet d'un \dot{q}_i sur une liaison

5.3.1.2 Liaison rotoïde

On a : $\dot{\mathbf{p}}_i = \dot{q}_i \cdot \mathbf{z}_i \times \mathbf{p}_{i,n}$ et $\boldsymbol{\omega}_i = \dot{q}_i \cdot \mathbf{z}_i$

Exercice : Illustrer les relations précédentes sur la figure 5.4.

A l'aide du symbole σ_i et de son complémentaire ($\sigma_i = 1/0$ pour une liaison P / R) on peut écrire :

$$\dot{\mathbf{p}}_i = (\sigma_i \mathbf{z}_i + \bar{\sigma}_i \mathbf{z}_i \times \mathbf{p}_{i,n}) \dot{q}_i \quad \text{et} \quad \boldsymbol{\omega}_i = \bar{\sigma}_i \cdot \mathbf{z}_i \cdot \dot{q}_i$$

5.3.2 Calcul de la Jacobienne (géométrique)

On va calculer séparément la vitesse linéaire du point O_n et le vitesse de rotation du repère R_n en appliquant le principe de superposition des vitesses :

$$\dot{\mathbf{p}} = \sum_{i=1}^n (\sigma_i \mathbf{z}_i + \bar{\sigma}_i \mathbf{z}_i \times \mathbf{p}_{i,n}) \dot{q}_i \quad \boldsymbol{\omega} = \sum_{i=1}^n (\bar{\sigma}_i \cdot \mathbf{z}_i) \cdot \dot{q}_i$$

On obtient alors :

$$\begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix} = J_g(\mathbf{q}) \cdot \dot{\mathbf{q}} = \begin{pmatrix} J_P \\ J_O \end{pmatrix} \cdot \dot{\mathbf{q}} = \begin{pmatrix} J_{P1} & J_{P2} & \dots & J_{Pn} \\ J_{O1} & J_{O2} & \dots & J_{On} \end{pmatrix} \cdot \dot{\mathbf{q}} \quad \text{avec} \quad \begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} \mathbf{z}_i \\ \mathbf{0} \end{bmatrix} & \text{prismatique} \\ \begin{bmatrix} \mathbf{z}_i \times (\mathbf{p}_n - \mathbf{p}_i) \\ \mathbf{z}_i \end{bmatrix} & \text{rotoïde} \end{cases}$$

$$J(\mathbf{q}) = \begin{pmatrix} \sigma_1 \mathbf{z}_1 + \bar{\sigma}_1 \mathbf{z}_1 \times \mathbf{p}_{1,n} & \sigma_2 \mathbf{z}_2 + \bar{\sigma}_2 \mathbf{z}_2 \times \mathbf{p}_{2,n} & \dots & \sigma_n \mathbf{z}_n + \bar{\sigma}_n \mathbf{z}_n \times \mathbf{p}_{n,n} \\ \bar{\sigma}_1 \cdot \mathbf{z}_1 & \bar{\sigma}_2 \cdot \mathbf{z}_2 & \dots & \bar{\sigma}_n \cdot \mathbf{z}_n \end{pmatrix}$$

Il suffit de projeter ces relations dans le repère de base pour avoir les coordonnées de $\dot{\mathbf{p}}$ et $\boldsymbol{\omega}$.

Pour calculer les vecteurs \mathbf{z}_i et $(\mathbf{p}_n - \mathbf{p}_i)$, on se sert des matrices de passages $T_{i,j}$ calculées pour le MGD.

On a calculé la jacobienne par rapport au repère de base. Nous pouvons exprimer la jacobienne dans n'importe quel repère du robot à l'aide de la matrice de rotation. Ainsi si on veut exprimer les vitesses linéaire et angulaire dans un repère b il suffit d'utiliser la matrice de changement de repère \mathbf{R}_b .

$$\begin{bmatrix} \dot{\mathbf{p}}_b \\ \boldsymbol{\omega}_b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{R}_b \end{bmatrix} \cdot \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{R}_b \end{bmatrix} \cdot J_g \cdot \dot{\mathbf{q}} = J_g^b \cdot \dot{\mathbf{q}} \quad \text{avec} \quad J_g^b \text{ la jacobienne dans le repère } b$$

Utilisation de la jacobienne préférentielle :

Lorsqu'on fait le calcul analytique de la jacobienne on peut utiliser la jacobienne préférentielle afin de simplifier le calcul analytique. On peut démontrer que :

$$J = \begin{pmatrix} R_{0,1} & 0 \\ 0 & R_{0,1} \end{pmatrix} \dots \begin{pmatrix} R_{j-1,j} & 0 \\ 0 & R_{j-1,j} \end{pmatrix} \begin{pmatrix} I & -\hat{\mathbf{P}}_{j+1,n(j)} \\ 0 & I \end{pmatrix} J_{j+1(n(j))}$$

avec :

- $j = \text{Ent}(\frac{n}{2})$, $\text{Ent}()$ est la fonction partie entière et j est l'indice préférentiel.
- $\hat{\mathbf{P}}_{j+1,n(j)}$ est la matrice de pré-produit vectoriel associé à la projection de $\mathbf{P}_{j+1,n}$ dans R_j .

— $J_{j+1(j)}$ la projection dans R_j de :

$$J_{j+1} = \begin{pmatrix} \sigma_1 \mathbf{z}_1 + \bar{\sigma}_1 \mathbf{z}_1 \times \mathbf{p}_{1,j+1} & \sigma_2 \mathbf{z}_2 + \bar{\sigma}_2 \mathbf{z}_2 \times \mathbf{p}_{2,j+1} & \dots & \sigma_n \mathbf{z}_n + \bar{\sigma}_n \mathbf{z}_n \times \mathbf{p}_{n,j+1} \\ \bar{\sigma}_1 \cdot \mathbf{z}_1 & \bar{\sigma}_2 \cdot \mathbf{z}_2 & \dots & \bar{\sigma}_n \cdot \mathbf{z}_n \end{pmatrix}$$

— $\text{rang}(J_{j+1}) = \text{rang}(J)$

Pour des robots industriels, $J_{j+1(j)}$ a généralement une majorité de termes nuls.

5.4 Calcul de la Jacobienne analytique

Il suffit d'exprimer la vitesse opérationnelle $\dot{\mathbf{x}} = (\dot{\mathbf{x}}_p^T, \dot{\mathbf{x}}_o^T)^T$ en fonction de $\dot{\mathbf{p}}$ et ω .

5.4.1 Dérivées de position

En repartant de l'équation de MGD : $\mathbf{x}_p = P_{0,n} + R_{0,n} \cdot \underline{O_n O_{n+1}_n}$ on obtient :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{pmatrix} + \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \times \left(a_n \cdot \begin{pmatrix} x_x \\ x_y \\ x_z \end{pmatrix} + r_{n+1} \cdot \begin{pmatrix} z_x \\ z_y \\ z_z \end{pmatrix} \right)$$

En utilisant la matrice de pré-produit vectoriel on obtient :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{pmatrix} + D \cdot \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad \text{avec} \quad D = \begin{pmatrix} 0 & a_n x_z + r_{n+1} z_z & -a_n x_y - r_{n+1} z_y \\ -a_n x_z - r_{n+1} z_z & 0 & a_n x_x + r_{n+1} z_x \\ a_n x_y + r_{n+1} z_y & -a_n x_x - r_{n+1} z_x & 0 \end{pmatrix}$$

5.4.2 Dérivées d'orientation

Soit l'orientation définie par le vecteur $(x_4, x_5, \dots, x_m)^T$. On note C la matrice définissant le passage entre les vitesses de rotation de l'OT et les dérivées des paramètres angulaires choisis¹ :

$$\begin{pmatrix} \dot{x}_4 \\ \dot{x}_5 \\ \dots \\ \dot{x}_m \end{pmatrix} = C \cdot \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

On peut écrire la Jacobienne analytique en regroupant les résultats précédents :

$$\dot{\mathbf{x}} = \begin{pmatrix} I & D \\ 0 & C \end{pmatrix} \cdot \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} I & D \\ 0 & C \end{pmatrix} \cdot J(\mathbf{q}) \cdot \dot{\mathbf{q}} = J_a(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

1. voir exemple en TD

Chapitre 6

Modèle Différentiel Inverse (MDI)

6.1 Méthodes de calcul du MDI

6.2 Résolution dans le cas de robot non-redondant

6.3 Singularités

6.4 Utilisation de la Jacobienne

Introduction au Modèle différentiel inverse

Position du problème

Le *MCD/MDD* permet de calculer $\dot{\mathbf{X}}$ pour un $\dot{\mathbf{q}}$ donné.

Maintenant on désire faire une variation de la situation de l'OT, pour une situation connue, \mathbf{X} , de l'OT.

Comment faire varier les variables de configuration ou de commande q_i en vitesse ? Quel $\dot{\mathbf{q}}$ donner en consigne ?

Hypothèses

- $\dot{\mathbf{X}}$ ou $d\mathbf{X}$ connu
- \mathbf{X} connue
- \mathbf{q} est calculé à partir de $\mathbf{X} \implies$ plusieurs solutions (tableau)

But

Trouver $\dot{\mathbf{q}}$ pour \mathbf{X} et $\dot{\mathbf{X}}$ données tel que $\dot{\mathbf{X}} = J(\mathbf{q}) \cdot \dot{\mathbf{q}}$
Il y a autant de solutions au MDI/MCI qu'au MGI.

Méthodes de calcul du MDI

On a : $\dot{\mathbf{X}}_{OT} = J_A(\mathbf{q}).\dot{\mathbf{q}}$ ou $\delta = J_{p+1(p)}(\mathbf{q}).\dot{\mathbf{q}}$ ou $\dot{\mathbf{X}}_n = \begin{pmatrix} \dot{\mathbf{p}} \\ \omega \end{pmatrix} = J(\mathbf{q}).\dot{\mathbf{q}}$

On peut montrer que $\text{rang}(J(\mathbf{q})) = \text{rang}(J_{p+1(p)}(\mathbf{q})) = \text{rang}(J_A(\mathbf{q}))$
(hors singularité du choix de coordonnées pour J_A)

On a donc à résoudre une équation du type $\dot{\mathbf{X}} = J(\mathbf{q}).\dot{\mathbf{q}}$ avec $J(\mathbf{q})$ la jacobienne correspondant à la valeur de $\dot{\mathbf{X}}$ considérée.

- Dériver le MGI : trop complexe.
- Résoudre un système linéaire de type $Y = A.X$.
 - inversion numérique de J
 - résolution analytique.

Problèmes d'inversion de J

- J n'est pas toujours inversible/n'est pas toujours de rang plein.
- J n'est pas toujours carrée.

2 / 14

Résolution du MDI pour des robots non-redondants

Robot non-redondant

Dans ce cas la, la matrice J est carrée.

On a donc $\dot{\mathbf{X}} = J(\mathbf{q}).\dot{\mathbf{q}}$ à résoudre :

- Hors singularité ($\det(J) \neq 0$) on sait que $\dot{\mathbf{q}} = J^{-1}.\dot{\mathbf{X}}$.
- Pour les configurations singulières ($\det(J) = 0$), il faut vérifier si le système est compatible ou non. J n'est pas inversible mais si le système est compatible alors il existe au moins une solution $\dot{\mathbf{q}}$.

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

3 / 14

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 carrées ($\min \|Y - A.X\|^2$)

4 / 14

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)

5 / 14

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 Σ 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 $J = U \cdot \Sigma \cdot V^t$
on a J^+ , matrice $(n \times m)$:

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

avec

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

Le jacobien inverse est indéfini si le robot est dans une configuration singulière.

Il existe deux types de singularités :

- Hors de l'espace de travail du robot.
- A l'intérieur de l'espace de travail du robot

Les configurations singulières du MGI sont aussi des configurations singulières que pour le MDI

Cette notion de singularité traduit se par une division par zéro ou par des vitesses articulaires qui tendent vers l'infinie.

Détection des singularités

- Tester les configurations singulières.
- Tester le facteur de conditionnement $\kappa = \frac{\sigma_{max}}{\sigma_{min}}$.

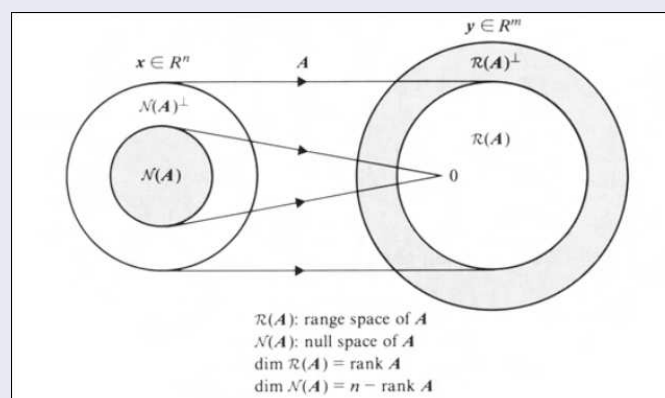
8 / 14

Utilisation de la Jacobienne

a- Transmission des vitesses entre espaces articulaire et tâche

- 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 $[\sigma_i \cdot u_i \cdot v_i^t]$.



Du livre de Y. Nakamura : Redundancy and Optimization, 1991

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

b- Efforts statiques

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

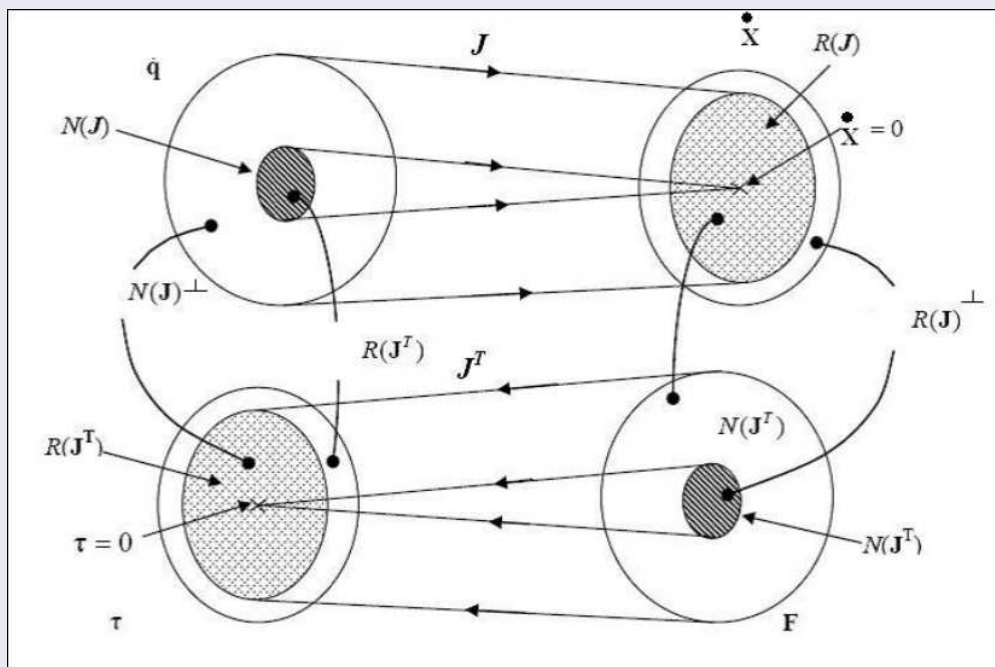
Puissance instantanée qui est transmise par l'outil (lié à l'O.T) sur l'environnement = puissance instantanée fournie par les actionneurs.
L'application du principe des travaux virtuels donne :

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

avec F = vecteur 6×1 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 .

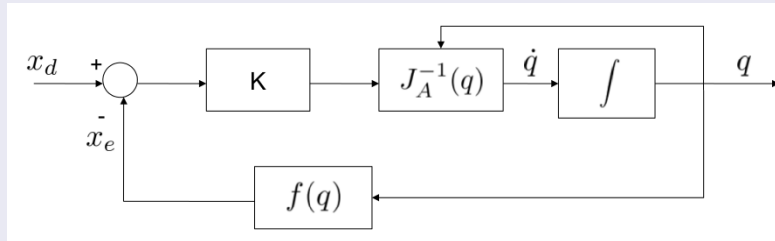
Dualité vitesse effort



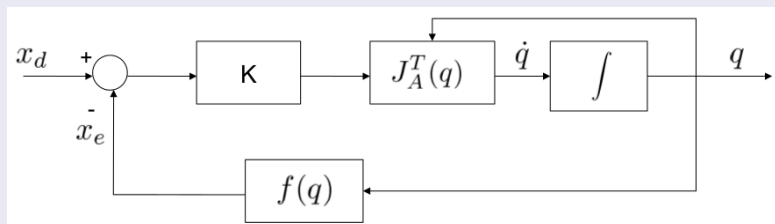
Nakamura-Redundancy and Optimization

c- Calcul du MGI numérique

Résoudre numériquement le MGI revient à appliquer une méthode d'optimisation non linéaire. Les méthodes d'optimisation non linéaire utilise au minimum la dérivée d'ordre un de la fonction à optimiser ce qui amène à utiliser la jacobienne J dans le cas du MGI.



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



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

- Exemple RRPRRR

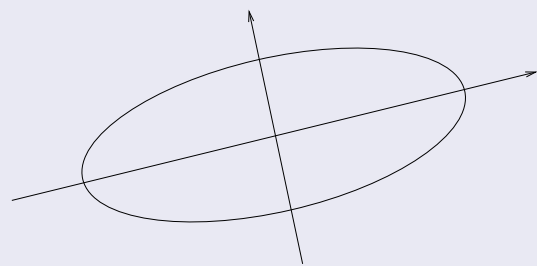
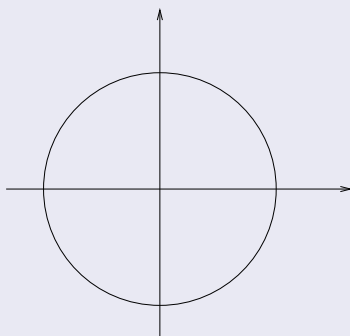
12 / 14

d- Indice de manipulabilité

- 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, hypersphère



Ellipsoïde de vitesse

Ellipsoïde de vitesse

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

- Exemple RRPRRR

13 / 14

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

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

e- Modèle cinématique d'ordre 2

On a : $\dot{\mathbf{X}} = J(\mathbf{q}).\dot{\mathbf{q}}$

$$\ddot{\mathbf{X}} = J(\mathbf{q}).\ddot{\mathbf{q}} + \dot{J}(\mathbf{q}).\dot{\mathbf{q}}$$

$$\ddot{\mathbf{q}} = J^{-1}(\mathbf{q}).[\ddot{\mathbf{X}} - \dot{J}(\mathbf{q}).\dot{\mathbf{q}}]$$

Chapitre 7

Les robots séries redondants

7.1 Redondance

7.2 Méthodes de résolution basées sur la jacobienne

Les robots series redondants

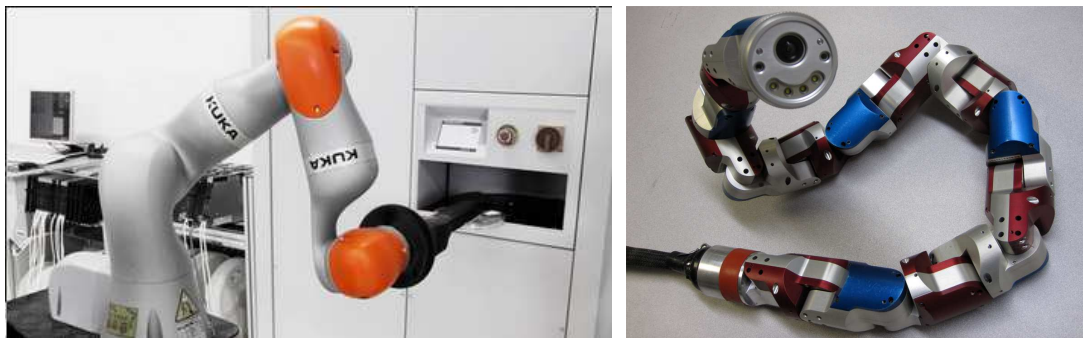


FIGURE – Exemples de robot redondant



FIGURE –

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 $(n - m)$.

Intérêt 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

2 / 12

Robot redondant par rapport à la tâche

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'OT

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 $\longrightarrow m = ?$

3 / 12

Robots séries redondants

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

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 :

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

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

4 / 12

Robots séries redondants

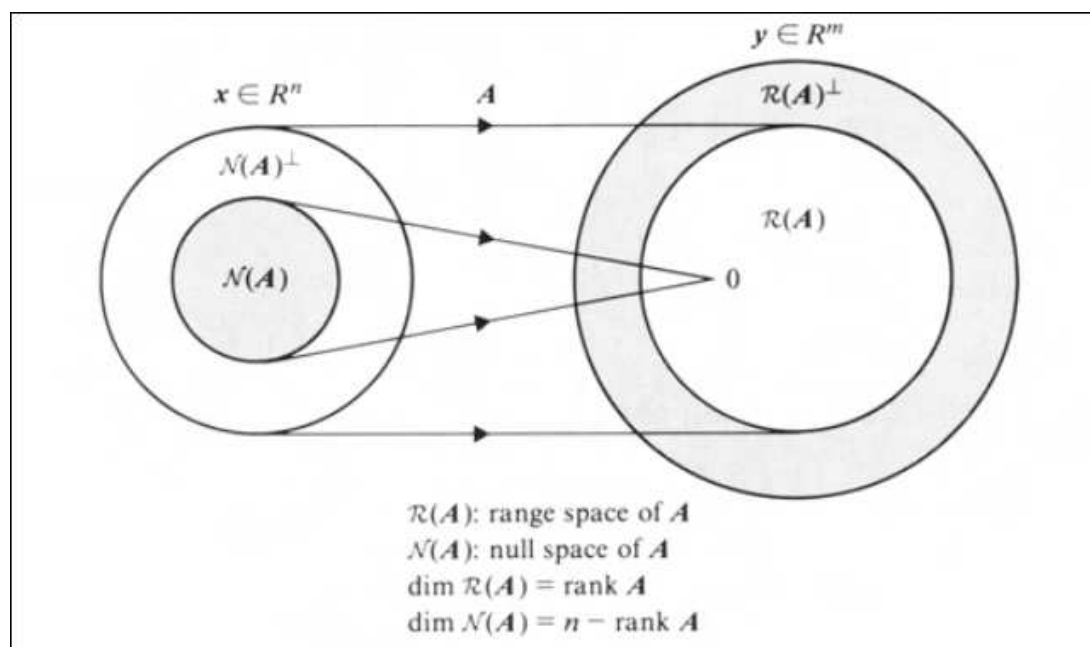


FIGURE – Nakamura-Redundancy and Optimization

Signification de ces espaces ?

5 / 12

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

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

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

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

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

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

6 / 12

Méthodes de résolution

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 (minimisation 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).

7 / 12

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 la 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 ?

8 / 12

Méthodes de résolution basées sur la Jacobienne

2 Pseudo-inverse pondérée :

On cherche à optimiser

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

sous la contrainte

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

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

J_W^+ appelé pseudo-inverse pondérée de J n'est pas une pseudo-inverse.

Intérêt ?

9 / 12

- ③ Moindres carrés amortis : Damped Least Square (DLS)
On cherche à optimiser

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

On obtient :

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

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

Intérêt : robuste proche des singularités.

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

10 / 12

Interprétation géométrique de l'effet de μ

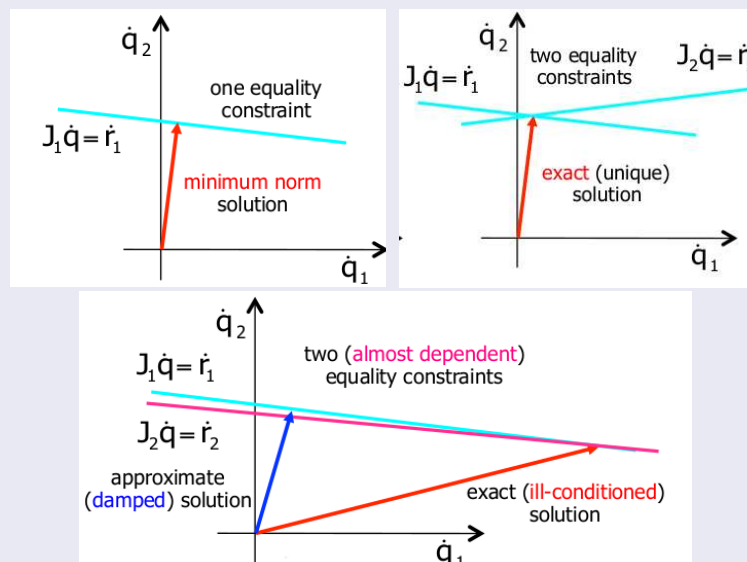


FIGURE – Alessandro De Luca- Sapienza

11 / 12

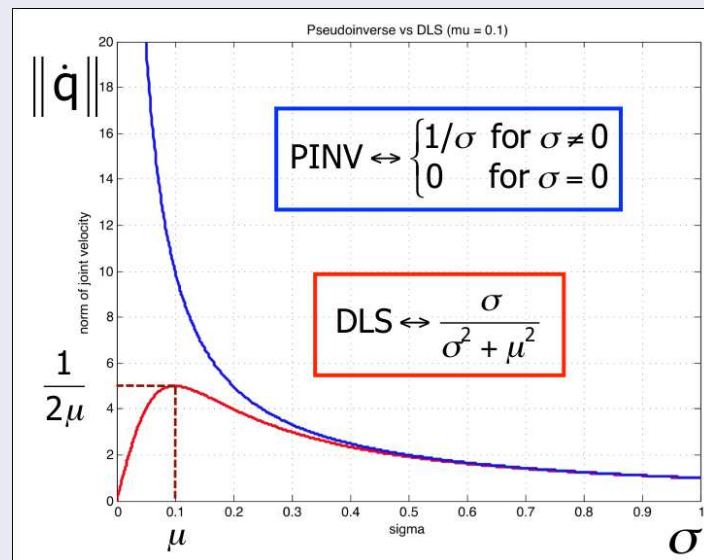


FIGURE – Alessandro De Luca- Sapienza

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}

Chapitre 8

Génération de trajectoires (mouvements)

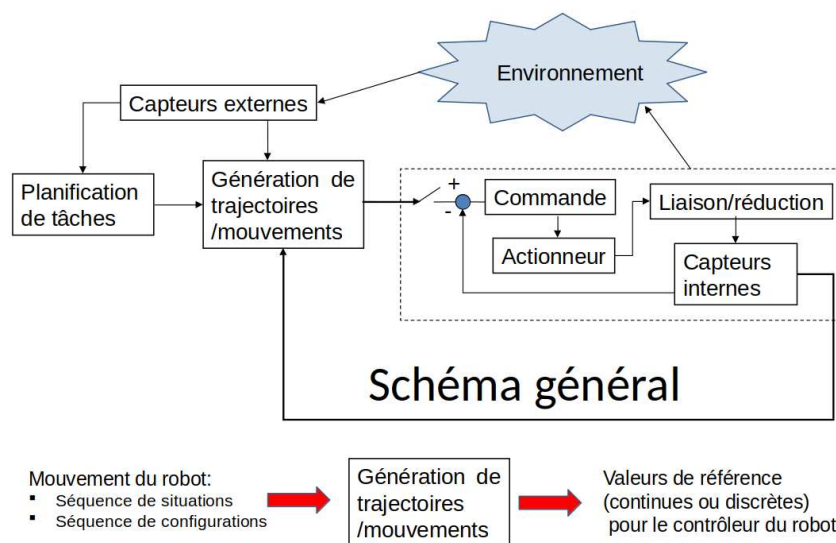
8.1 Définitions

8.2 GT dans l'espace articulaire

8.2.1 Loi polynomiale

8.2.2 Profil de vitesse en trapèze

8.3 GT dans l'espace de la tâche



La génération de trajectoires (ou de mouvement) consiste à définir la consigne en fonction du temps à envoyer à la partie commande (consigne des asservissements).

La génération de trajectoires doit :

- * respecter la trajectoire/mouvement de la tâche
- * avoir des trajectoires articulaires lisses pour préserver la mécanique
- * respecter les limites articulaires de vitesses et d'accélérations maximales
- * tenir compte de contraintes de début et de fin des mouvements
- * ...

Procédure standard sur les robots industriels

Définir les « points » cartésiens intéressants (situations) en utilisant le pendant (BaB). Ici pas de problème de MGI !

Programmer une vitesse (moyenne) entre ces points comme un pourcentage de la vitesse maximum (différence entre mouvements articulaires et cartésiens)

Interpolation linéaire dans l'espace articulaire entre les « points »

Options

Trajectoire circulaire entre 3 « points »

Points de passage/points d'arrêt

...

Inconvénients :

Programmation semi-manuelle (langage robotique de « first generation »)

Visualisation du mouvement du robot limitée

Il faut définir la notion de trajectoire/mouvement \Rightarrow formalisme mathématique

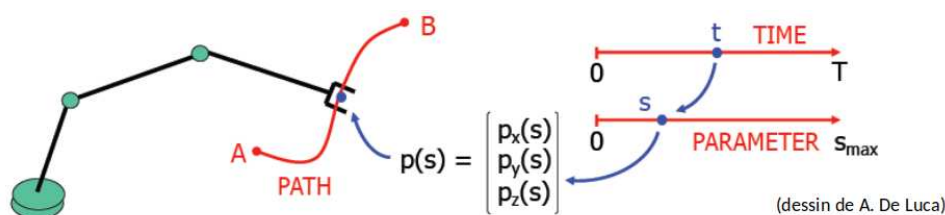
1- Définitions

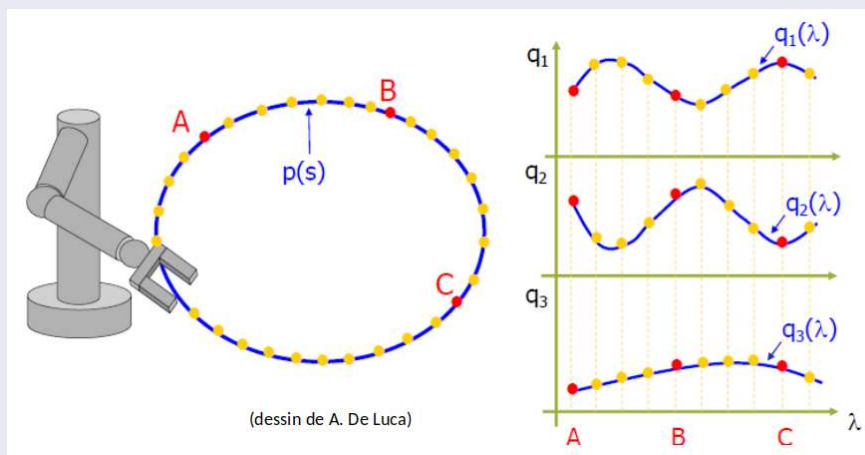
Trajectoire = courbe décrite par un objet en mouvement.

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.





Trajectoire géométrique (*path*) : $p = p(s)$
 +
 Loi de mouvement (*timing law*) : $s = s(t)$
 ↓
 Génération de mouvement (*trajectory*) $p(t) = p(s(t))$
 ↓
 Génération de coordonnées articulaires $q_i(t)$

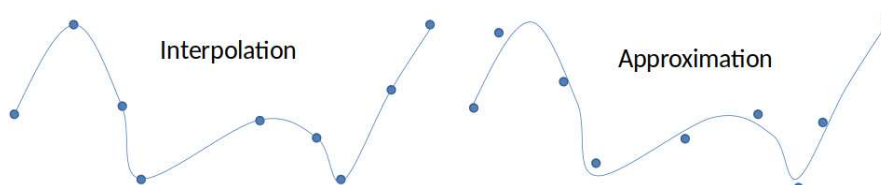
Contraintes

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.

Types de contraintes

- * spatiale
- * temporelle
- * saturation(K_v , K_a)
- * continuité
- * ...



Types de problème

On peut définir la GT dans :

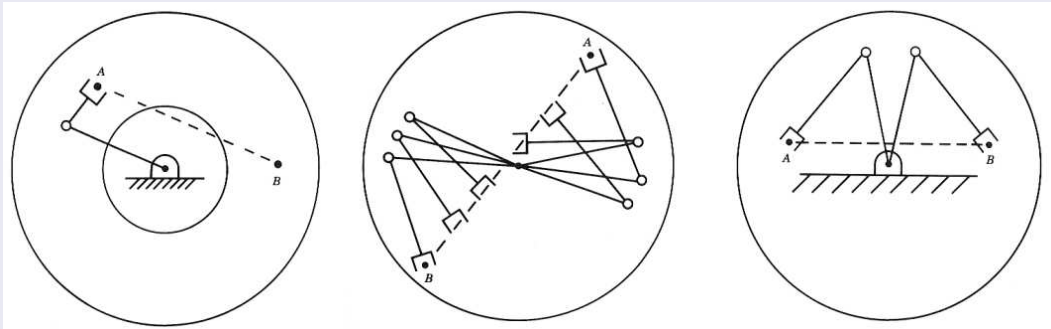
- * l'espace articulaire (Q)
 - Facilité pour « points » de passage
 - Pas de problème de singularité
 - Peu de calcul
 - Impossible de suivre une trajectoire définie dans X
- * l'espace de la tâche
 - Suivi de forme
 - "Couteux" en temps de calcul (calcul des modèles inverses)
 - Problème de discontinuité

Plusieurs types de problème : tableau

- * X_i et X_f (ou Q_i et Q_f) imposées (trajectoire libre)
- * X_i et X_f (ou Q_i et Q_f) imposées + "points" de passage
- * X_i et X_f (ou Q_i et Q_f) + trajectoire $X(u)$ (ou $Q(u)$) imposées
- * X_i et X_f (ou Q_i et Q_f) + $X(u)$ (ou $Q(u)$) + $u(t)$ imposées (loi de mvt)

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 différent type de solutions : limites articulaires \Rightarrow différentes solutions du MGI
Exemple : tous les points atteignables mais dans des configurations de bras différentes



2- GM dans l'espace articulaire : Loi polynomiale

Loi polynomiale

Comment définir un polynôme qui convienne ?

$$q(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + \dots + a_p \cdot t^p$$

- * degré ?
- * coefficients a_i ?

Prendre en compte les contraintes sur le mouvement des variables articulaires

Exemple sur un axe q

- * q_0 et q_f fixées
- * \dot{q}_0 et \dot{q}_f fixées (on prend $\dot{q}_0 = \dot{q}_f = 0$)
- * K_v et K_a : vitesse et accélération maximale

Calculer le polynôme correspondant

2-1 Calcul du polynôme

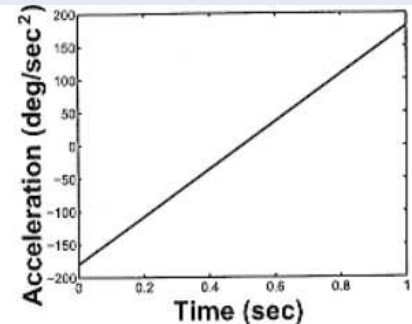
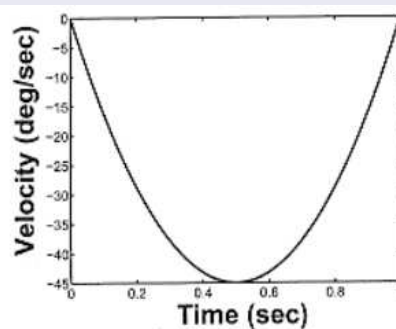
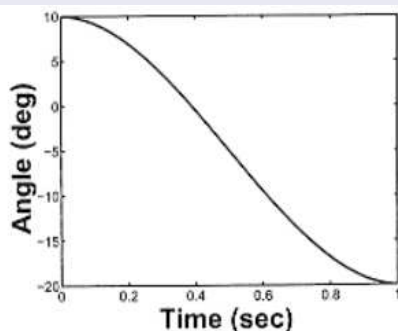
$$q(t) =$$

$$\dot{q}(t) =$$

$$\ddot{q}(t) =$$

Profil de vitesse parabolique

Profil d'accélération linéaire



Calcul des coefficients en fonction de t_f

2-2 Calcul du temps final t_f

Arbitraire ?

Temps minimum ?

Prendre en compte K_v et K_a

La vitesse est maximale pour $t = \frac{t_f}{2}$ et ne doit pas dépasser K_v

$$\Rightarrow t_{K_v} \text{ minimal} = \frac{2}{3} \cdot \frac{\Delta q}{K_v}$$

L'accélération est maximale pour $t = 0$ et $t = t_f$ et ne doit pas dépasser $\pm K_a$

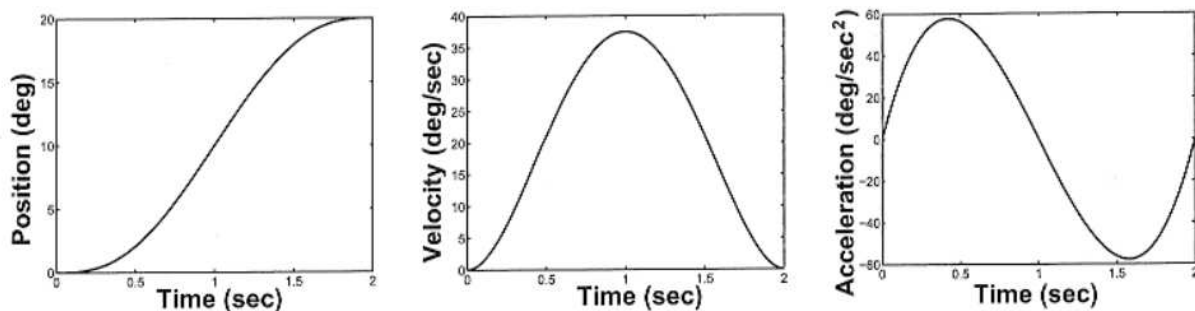
$$\Rightarrow t_{K_a} \text{ minimal} = \sqrt{\frac{6 \cdot \Delta q}{K_a}}$$

$$t_f = \max(t_{K_v}, t_{K_a})$$

Sur un robot il faut faire ce calcul pour tous les axes i et prendre $t_f = \max(t_{f_i})$

Discontinuité en accélération : dépend de la rigidité de la structure mécanique.
Solution : ajouter deux contraintes en accélération, \ddot{q}_0 et \ddot{q}_f .

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



On peut imaginer des contraintes sur le *jerk* (dérivée de l'accélération), ...

3- GM dans l'espace articulaire : Profil de vitesse trapézoïdal

Objectif : minimiser le temps (Bang-bang)

- * Accélération/décélération constantes
- * trajectoire = segment linéaire et deux segments paraboliques

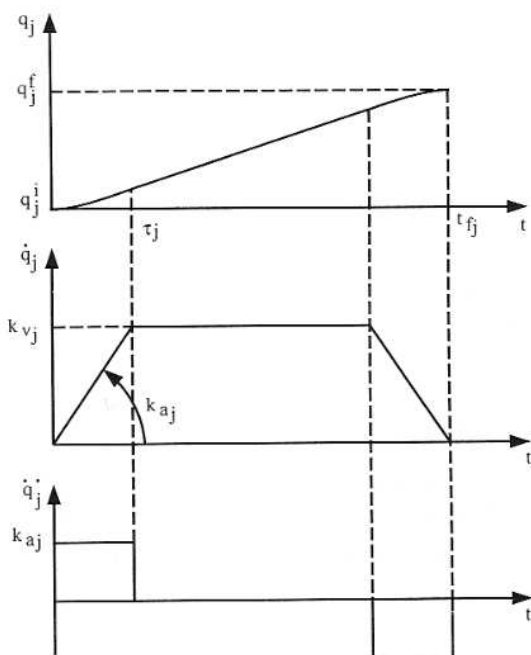
Problème

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

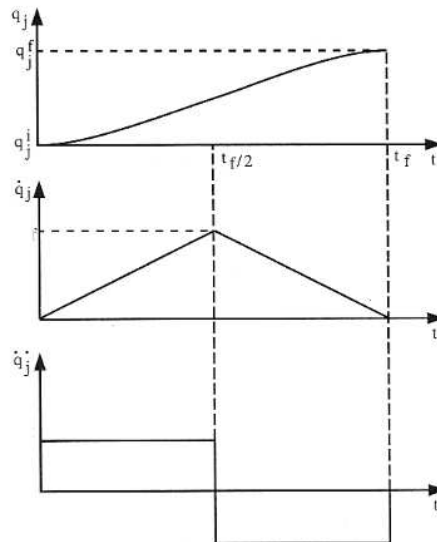
Input = q_0, q_f, K_v, K_a

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

3-1 Calcul de $\tau, t_f, q(t), \dot{q}(t), \ddot{q}(t)$



Cas vitesse max $< K_v$?



Discontinuité en accélération \Rightarrow jerk

3-2 Synchronisation entre les axes

Tous les axes bougent en même temps

Même temps de début et de fin

\Rightarrow l'axe qui prend le plus de temps impose le temps final.

Intérêt :

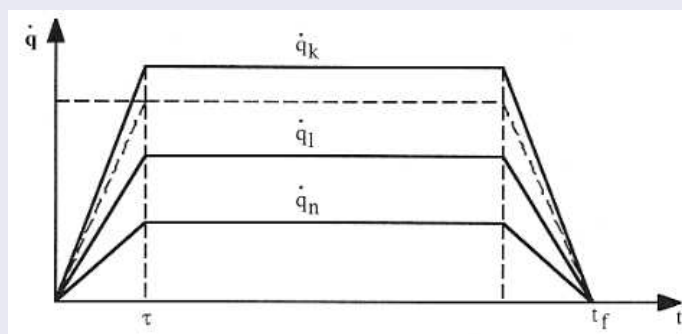
Trajectoire identique : A \rightarrow B and B \rightarrow A

minimise le nombre d'axes à vitesse/accélération maximale

Cas simple : tous les temps d'accélération sont égaux

On suppose que c'est l'axe k qui impose t_f .

Pour tout $i \neq k$ calculer la nouvelle vitesse max.



Cas général : les temps d'accélération sont différents

On synchronise les temps τ_i et τ_j de tous les axes

⇒ pour tous les axes on va calculer :

- * vitesse maximum = $\lambda_j \cdot K_{v_j}$ avec $0 \leq \lambda_j \leq 1$
- * accélération maximum = $\gamma_j \cdot K_{a_j}$ avec $0 \leq \gamma_j \leq 1$

Il faut donc calculer les variables λ_j et γ_j .

Commencer par calculer tous les temps finaux pour chaque axe pour connaître t_f minimum pour la trajectoire

Exemple avec deux axes

$$t_{f_1} = \tau_1 + \frac{\Delta_1}{K_{v_1}} = \frac{K_{v_1}}{K_{a_1}} + \frac{\Delta_1}{K_{v_1}} \text{ et } t_{f_2} = \tau_2 + \frac{\Delta_2}{K_{v_2}} = \frac{K_{v_2}}{K_{a_2}} + \frac{\Delta_2}{K_{v_2}} \text{ et } t_f = \max(t_{f_1}, t_{f_2})$$

Exemple avec deux axes

$$t_f = \frac{\lambda_1 \cdot K_{v_1}}{\gamma_1 K_{a_1}} + \frac{\Delta_1}{\lambda_1 \cdot K_{v_1}} = \frac{\lambda_2 \cdot K_{v_2}}{\gamma_2 \cdot K_{a_2}} + \frac{\Delta_2}{\lambda_2 \cdot K_{v_2}}$$

↓

$$\tau = \frac{\lambda_1 \cdot K_{v_1}}{\gamma_1 K_{a_1}} = \frac{\lambda_2 \cdot K_{v_2}}{\gamma_2 \cdot K_{a_2}} \quad \lambda_2 = \lambda_1 \cdot \frac{K_{v_1}}{K_{v_2}} \cdot \frac{\Delta_2}{\Delta_1}$$

$$\text{On en déduit : } \gamma_2 = \gamma_1 \cdot \frac{K_{a_1}}{K_{a_2}} \cdot \frac{\Delta_2}{\Delta_1}$$

$$\text{Or } 0 \leq \lambda_i \leq 1 \text{ et } 0 \leq \gamma_i \leq 1 \implies 0 \leq \lambda_1 \leq \frac{K_{v_2}}{K_{v_1}} \cdot \frac{\Delta_1}{\Delta_2} \text{ et } 0 \leq \gamma_1 \leq \frac{K_{a_2}}{K_{a_1}} \cdot \frac{\Delta_1}{\Delta_2}$$

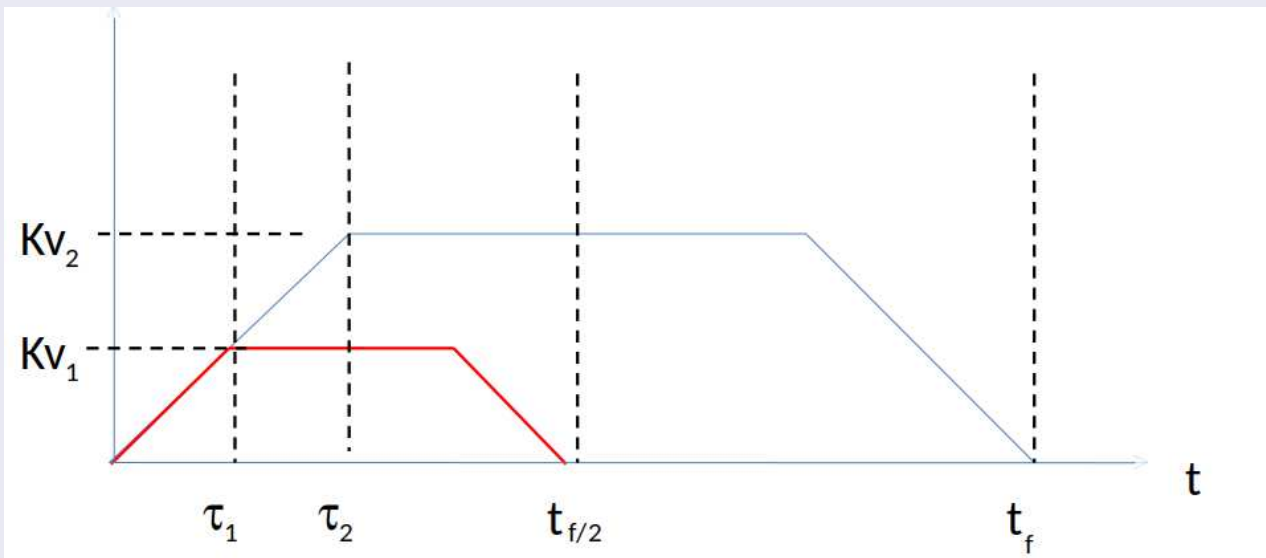
$$\lambda_1 = \min\left(1, \frac{K_{v_2}}{K_{v_1}} \cdot \frac{\Delta_1}{\Delta_2}\right) \text{ et } \gamma_1 = \min\left(1, \frac{K_{a_2}}{K_{a_1}} \cdot \frac{\Delta_1}{\Delta_2}\right) \implies \tau = \frac{\lambda_1 \cdot K_{v_1}}{\gamma_1 K_{a_1}}$$

Généralisation pour les n axes

Exemple avec deux axes

Exemple : $Ka_1 = Ka_2$, $2.Kv_1 = Kv_2$, $\Delta_2 = 4.\Delta_1$

Nouveau profil de vitesse ?



3- Mouvement dans l'espace opérationnel (de la tâche)

En général les méthodes proposées dans l'espace généralisé peuvent s'appliquer dans l'espace cartésien. Il suffit de considérer de manière indépendante chaque composante de la tâche (une variable de position ou angulaire) comme une variable q_i dans l'espace généralisé.

Principe général

- * calculer une trajectoire géométrique (*path*) : $X = X(s)$
- * calculer une loi de mouvement (*timing law*) : $s = s(t)$

Génération de mouvement (*trajectory*) $X(t) = X(s(t))$

Quand c'est possible, on préfère générer séparément le mouvement en position et en rotation dans l'espace opérationnel.

En général le nombre de situations à interpoler est faible dans l'espace opérationnel (2 ou 3 X_i).

Méthode d'interpolation simple : ligne droite, arc de cercle...

Représentation des rotations

- * Représentation minimale : 3 angles indépendants. On peut générer le mouvement pour chaque composante angulaire définissant l'orientation de l'O.T
Exemple : une loi linéaire dans l'espace (θ, β, ϕ) avec une loi de temps cubique.
Difficulté : prédire/comprendre les orientations intermédiaires
- * Représentation non minimale : quaternion ou de vecteur de rotation r et d'angle θ
 - Connaissant l'orientation initiale R_i et finale R_f on calcule la matrice de changement d'orientation $R = R_i^t \cdot R_f$
 - Calculer le vecteur de rotation r et l'angle de rotation $\theta_{i,f}$
 - Générer une loi de mouvement $\theta(t)$ pour $\theta \in [0, \theta_{i,f}]$
 $\forall t, R_i \cdot R(r, \theta(t))$ donne l'orientation de l'O.T au temps tOn peut facilement rajouter des contraintes (exemple sur $\dot{\theta}(t)$)

Exemple 1 : mouvement linéaire en position

Inputs : P_i (point initial), P_f (point final), V_M (vitesse maximale), A_M (accélération maximale).

On prend les vitesses $V_i = 0$ et $V_f = 0$ afin de simplifier les calculs.

Information à connaître

$$L = \|P_f - P_i\|$$

$\frac{P_f - P_i}{\|P_f - P_i\|}$ vecteur unitaire de direction P_i vers P_f .

Soit u l'abscisse curviligne, $u \in [0, L]$ (longueur parcourue le long de la trajectoire).

En général on norme u : $s = \frac{u}{L} \implies s \in [0, 1]$.

$$P(s) = p_i + s \cdot (P_f - P_i)$$

$$\frac{dP}{ds} = ?$$

$$\dot{P}(s) = \dot{P}(s(t)) = \frac{dP}{dt} = ?$$

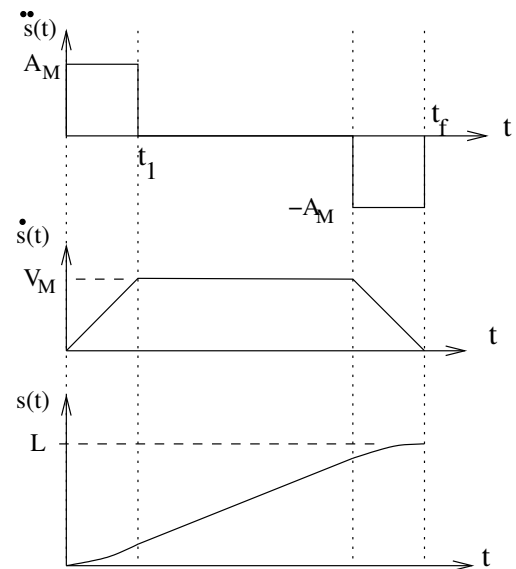
$$\ddot{P}(s) = ?$$

Profil de vitesse en trapèze de $\dot{s}(t)$

\Rightarrow même raisonnement que dans l'espace généralisé mais ici le mouvement est en 3D :

$$P(s) = \begin{pmatrix} x(s) \\ y(s) \\ z(s) \end{pmatrix} \quad t_1 = \frac{V_M}{A_M}$$

$$t_f = \frac{L}{V_M} + \frac{V_M}{A_M}$$



$$0 \leq t \leq t_1 : s(t) = \frac{A_M}{2} \cdot t^2$$

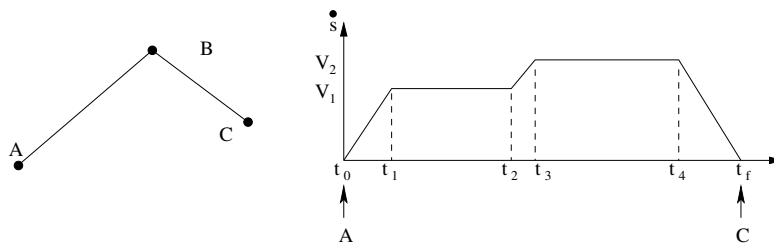
$$t_1 \leq t \leq (t_f - t_1) : s(t) = V_M \cdot t - \frac{V_M^2}{2 \cdot A_M}$$

$$(t_f - t_1) \leq t \leq t_f : s(t) = -\frac{A_M}{2} \cdot (t - t_f)^2 + V_M \cdot t - \frac{V_M^2}{A_M}$$

Exemple 2 : trajectoire en ligne brisée avec point de passage

On désire la trajectoire suivante : on part de A, on veut atteindre une vitesse V_1 en allant vers B, on passe au voisinage de B sans s'arrêter (point de passage) et on se dirige vers C avec une vitesse V_2 .

Hypothèse : la transition entre les segments se fait à accélération constante.



Problème : calculer la transition entre les segments

Point de départ A' entre A et B ?

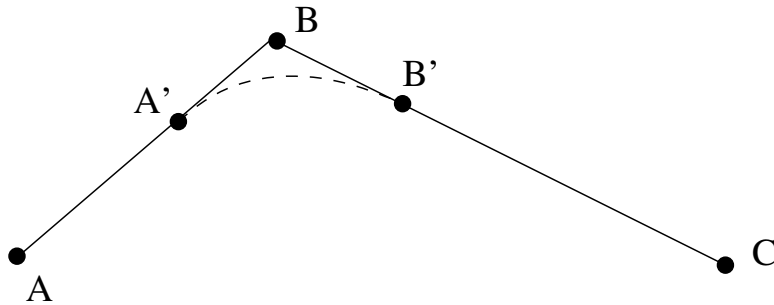
Point d'arrivée B' entre B et C ?

Durée ΔT de la transition ?

$K_{AB} = \frac{B-A}{\|B-A\|}$ vecteur unitaire de direction A vers B

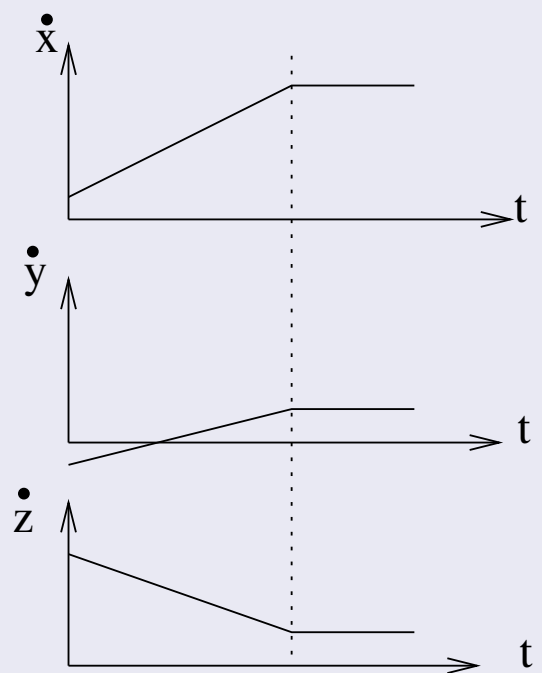
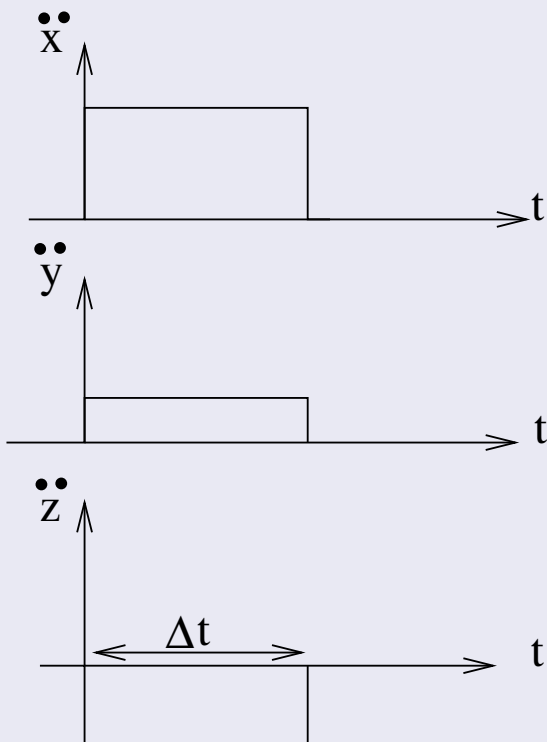
$K_{BC} = \frac{C-B}{\|C-B\|}$ vecteur unitaire de direction B vers C

De A' à B' la trajectoire reste dans le plan défini par les lignes AB et BC , c'est un problème dans le plan.



Loi temporelle durant la transition

On désire des accélérations constantes pendant le temps de transition ΔT .



Calcul du mouvement du point P le long de la trajectoire

$$P(s) = \begin{pmatrix} x(s) \\ y(s) \\ z(s) \end{pmatrix} =$$

$$\dot{P}(s) = \begin{pmatrix} \dot{x}(s) \\ \dot{y}(s) \\ \dot{z}(s) \end{pmatrix} =$$

$$\ddot{P}(s) = \begin{pmatrix} \ddot{x}(s) \\ \ddot{y}(s) \\ \ddot{z}(s) \end{pmatrix} =$$

Comment calculer A' et B' ?

Plusieurs solutions

Contrainte sur la distance au point B

On veut que $B - A' = d_1 \cdot K_{AB}$ et $C' - B = d_2 \cdot K_{BC}$

La trajectoire du point P en partant de A' est :

$$P(t) = A' + V_1 \cdot K_{AB} \cdot t + \frac{(V_2 \cdot K_{BC} - V_1 \cdot K_{AB})}{2 \cdot \Delta T} \cdot t^2$$

$$P(\Delta T) = A' + V_1 \cdot K_{AB} \cdot t + \frac{\Delta T}{2} \cdot (V_2 \cdot K_{BC} + V_1 \cdot K_{AB}) = C' \\ -B + A' + V_1 \cdot K_{AB} \cdot t + \frac{\Delta T}{2} \cdot (V_2 \cdot K_{BC} + V_1 \cdot K_{AB}) = C' - B$$

$$d_1 \cdot K_{AB} + d_2 \cdot K_{BC} = \frac{\Delta T}{2} \cdot (V_2 \cdot K_{BC} + V_1 \cdot K_{AB})$$

Relation vraie $\forall K_{AB}, K_{BC}$: $d_1 = V_1 \cdot \frac{\Delta T}{2}$ et $d_2 = V_2 \cdot \frac{\Delta T}{2}$

On choisit par exemple $d_1 \implies \Delta T = \frac{2 \cdot d_1}{V_1} \implies d_2 = d_1 \cdot \frac{V_2}{V_1}$

Chapitre 9

Modèle dynamique

9.1 Equations d'Euler-Lagrange

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

9.3 Exemples

Introduction aux équations de la dynamique en robotique

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

Modèle dynamique

Système de n équations différentielles d'ordre 2 reliant les vitesses et accélérations (dérivées premières et secondes par rapport au temps) des n coordonnées articulaires aux n efforts articulaires.

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, dimensionnement des actionneurs, identification des inerties,...

Rappels : efforts statiques

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

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

F = vecteur 6×1 des efforts (forces et couples) agissant sur l'outil.

Calcul des équations de la dynamique

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

Équation dynamique d'un robot

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

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

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

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,

τ_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).

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

avec :

I_i : tenseur (matrice) d'inertie du corps i ,

m_i : masse du corps i ,

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

ω_i : vitesse angulaire du corps i .

Tenseur d'inertie d'un solide S

$$I = \begin{pmatrix} \int_S (y^2 + z^2).dm & -\int_S xy.dm & -\int_S xz.dm \\ -\int_S xy.dm & \int_S (x^2 + z^2).dm & -\int_S yz.dm \\ -\int_S xz.dm & -\int_S yz.dm & \int_S (x^2 + y^2).dm \end{pmatrix}$$

- Matrice constante si elle est calculée dans un repère lié au solide
- Si le solide est symétrique par rapport aux axes de son repère alors les éléments hors diagonale sont nuls
- Calcul effectué par logiciel de CAO (*onshape*, ...)

Energie cinétique dans le cas d'un robot

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

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

- $J_{v_i}(q)$: 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.

Attention : il faut que cette matrice soit exprimée dans le même repère que la matrice d'inertie I_i : $J_{\omega_i}(q) = {}^i J_{\omega_i}(q) = R_{i,0}^T J_{w_i}(q) = R_{0,i}^T J_{w_i}(q)$

Matrice d'inertie du robot

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

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

La matrice $A(q)$ est appelée matrice d'inertie du robot.

$A(q)$ 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

Sur certains robots, l'effet de la gravité est compensé par des ressorts (on peut négliger alors V).

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

Lagrangien dans le cas d'un robot

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

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

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.

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

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

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

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

Écriture 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 $(k, j)^{\text{ème}}$ élément de \mathbf{C} est défini par $r_{kj} = \sum_j c_{ijk}(q)\dot{q}_i$

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

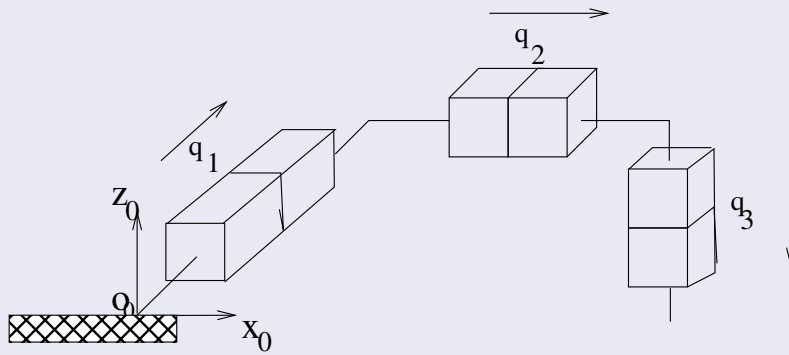
Les termes impliquant un produit $\dot{q}_i \dot{q}_j$ ($i \neq j$) sont appelés Coriolis

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

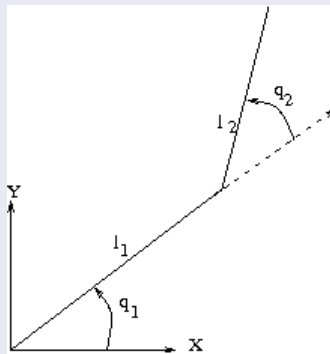
- $\boldsymbol{\tau}_m$ couples des actionneurs,
- F_c forces de contact
- $\gamma_f(\mathbf{q}, \dot{\mathbf{q}})$ de frottement

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

Exemple 1 : robot cartésien PPP



Exemple 2 : robot plan RR



Chapitre 10

Modélisation de robot mobile à roues

10.1 Contrainte non holonôme

10.2 Modèle cinématique pour un RM de type unicycle

10.3 Modèle cinématique pour un RM de type voiture

10.4 Commandabilité

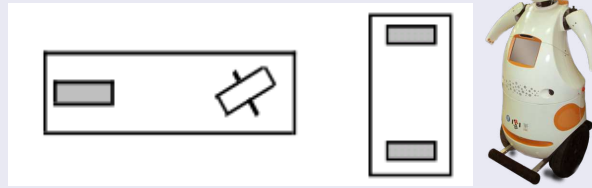
Introduction à la modélisation de robot mobile à roues

Ce chapitre aborde les thèmes suivants :

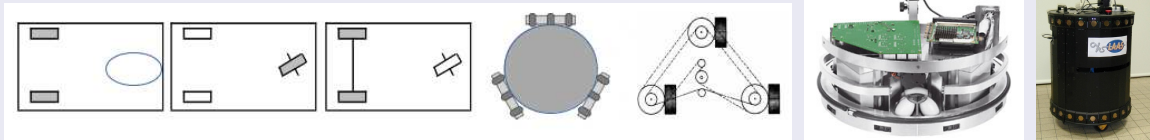
- Généralités sur les modèles de robots à roues
 - Modèle cinématique pour RM à roues
 - Roulement sans glissement
 - Contrainte non holonome
 - Modèle cinématique pour un RM de type "differential drive"
 - Système commandable ou non ?
-
- Le modèle cinématique fait correspondre la vitesse du robot avec la vitesse de ses roues.
 - Le modèle dynamique fait correspondre le couple des moteurs au niveau des roues avec l'accélération du robot. Nous ignorerons la dynamique dans ce chapitre.
 - Intérêt d'un modèle géométrique ?

Différents Arrangements de roues

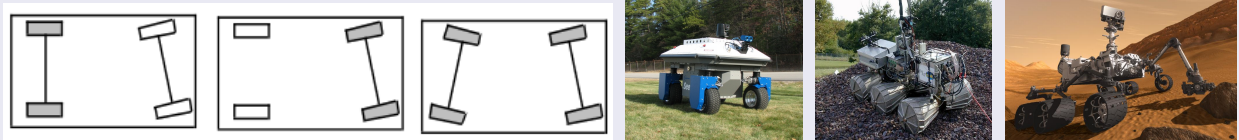
- Deux roues



- Trois roues

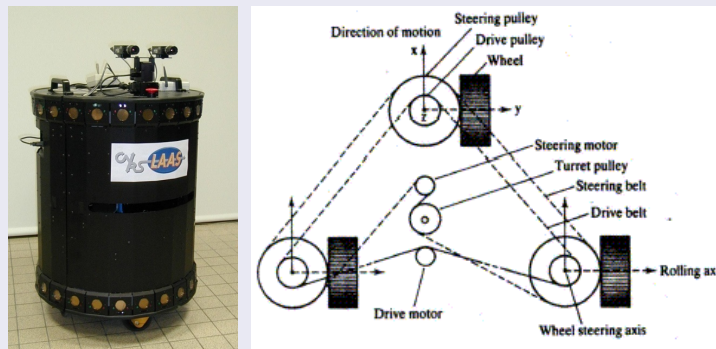


- Quatre roues... ou plus



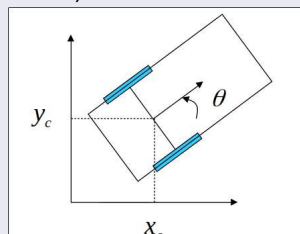
Exemple de robot mobile à roues de type Synchro Drive

- Plateforme omnidirectionnelle :
- Découplage de la commande entre orientation et translation



Exemple de robot mobile de type *Differential-Drive*, unicycle

- 2 roues motrices indépendantes, actionnées chacune par un moteur
- 1 roue folle, libre



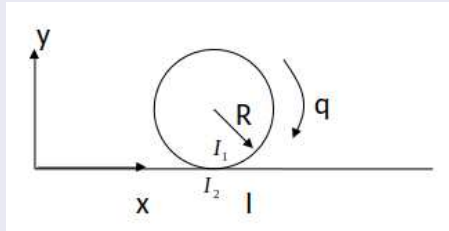
Hypothèses

- Robot est constitué d'un corps rigide
- Contact roue/sol ponctuel sur un sol plat
- Roues indéformables et de rayon constant
- **Roulement sans glissement**

Roulement sans glissement

Contrainte de roulement sans glissement :

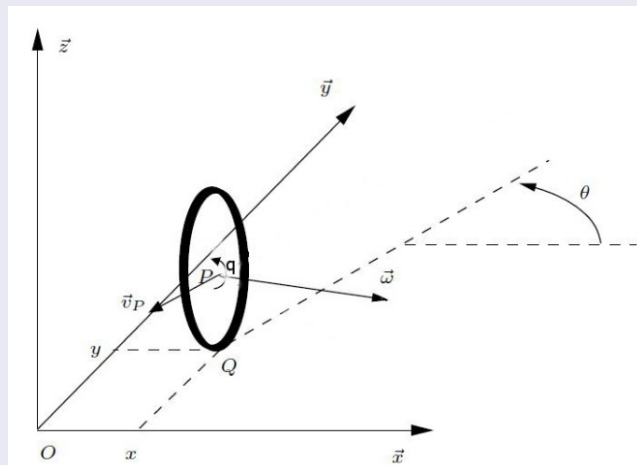
- Un solide S1 roule sans glisser sur un solide S2 si la vitesse de glissement du premier par rapport au second est nulle.
- Considérons une roue avec I point fixe en contact avec le sol



- $V_{1/2} = V_{I_1/R} - V_{I_2/R} = 0 \implies$ mouvement rectiligne de la roue : $\dot{x} + r.\dot{q} = 0$

Contrainte de roulement sans glissement : roue sur un plan

- Configuration de la roue : $= (x, y, \theta, q)$
- Vitesse relative de la roue au point de contact $V_Q = V_P + \omega \times \vec{PQ} = 0$
- $P = (x, y, r)^T$, $Q = (x, y, 0)^T$ et $\omega = (\sin(\theta).\dot{q}, \cos(\theta).\dot{q}, 0)^T$ donne :
 $\dot{x} + r.\cos(\theta).\dot{q} = 0$
 $\dot{y} + r.\sin(\theta).\dot{q} = 0$



- Composantes de la vitesse dans le plan de la roue et dans le plan :
 $-\dot{x}.\sin(\theta) + \dot{y}.\cos(\theta) = 0$ et $\dot{x}.\cos(\theta) + \dot{y}.\sin(\theta) = -r.\dot{q}$

Contrainte non holonôme

Le système est soumis à une contrainte holonome si les coordonnées $x_j, j = 1, \dots, n$, vérifient une équation de liaison de la forme $f(x_1, \dots, x_n, t) = 0$. Le système peut être soumis simultanément à $c \leq n$ contraintes holonomes indépendantes.

Exemple : bras manipulateurs

Contrainte non-holonome

Une contrainte est dite non holonome si elle ne peut pas s'écrire sous la forme $f(x_1, \dots, x_n, t) = 0$.

- soit la forme différentielle est non intégrable,
- soit elle s'exprime par une inégalité,
- soit elle fait intervenir les vitesses.

Question :

La contrainte non holonome $F(q, \dot{q}, t) = 0$ est-elle non intégrable ?

Théorème de Frobenius

Contrainte non holonome

Soit un système de configuration q , de dimension n , soumis à un ensemble de contraintes indépendantes s'écrivant sous la forme : $A^T(q) \cdot \dot{q} = 0$

Soit $B(q)$ matrice de rang plein m orthogonale à $A(q)$. ($A^T(q) \cdot B(q) = 0$)

Soit l'algèbre de Lie de dimension p ($m \leq p \leq n$) engendrée par les colonnes de $B(q)$ et leurs crochets de Lie successifs à condition qu'ils augmentent la dimension de l'algèbre alors $(n - p)$ contraintes sont intégrables.

Exemple de la roue

Ecrire la matrice $A(q)$

Vérifier que $B(q)$ est bien une matrice orthogonale à $A(q)$

$$[b_1(q), b_2(q)] = (\sin(\theta), -\cos(\theta), 0, 0)^T$$

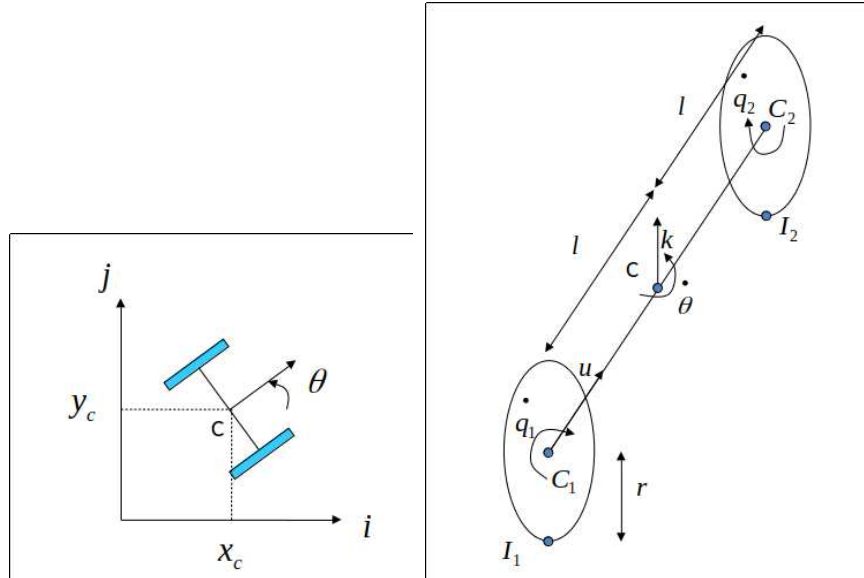
$$[b_1, [b_1(q), b_2(q)]] = 0$$

$$[b_2(q), b_2(q)] = (\cos(\theta), \sin(\theta), 0, 0)^T$$

$b_1(q), b_2(q), [b_1(q), b_2(q)]$ et $[b_2(q), [b_1(q), b_2(q)]]$ engendrent une algèbre de Lie de dimension 4, égale à la dimension de $q \implies$ aucune contrainte est intégrable \implies toutes les contraintes sont non intégrables.

Modèle cinématique pour un unicycle (*differential-drive*)

Unicycle : robot actionné par deux roues motorisées indépendantes et un certain nombre de roues folles pour assurer sa stabilité.



- $V_{I_1} = V_C + \omega \times \vec{CI_1}$ et $V_{I_2} = V_C + \omega \times \vec{CI_2}$
- $\omega_1 = \dot{\theta} \cdot \vec{k} + \dot{q}_1 \cdot \vec{u}$ et $\omega_2 = \dot{\theta} \cdot \vec{k} + \dot{q}_2 \cdot \vec{u}$ avec $\vec{u} = (-\sin(\theta) \cos(\theta) 0)^T$

$$\begin{cases} \dot{x}_c - r \cdot \cos(\theta) \cdot \dot{q}_1 + l \cdot \cos(\theta) \cdot \dot{\theta} & = 0 \\ \dot{y}_c - r \cdot \sin(\theta) \cdot \dot{q}_1 + l \cdot \sin(\theta) \cdot \dot{\theta} & = 0 \\ l \cdot \sin(\theta) \cdot \cos(\theta) \cdot \dot{q}_1 - l \cdot \sin(\theta) \cdot \cos(\theta) \cdot \dot{q}_1 & = 0 \\ \dot{x}_c - r \cdot \cos(\theta) \cdot \dot{q}_2 - l \cdot \cos(\theta) \cdot \dot{\theta} & = 0 \\ \dot{y}_c - r \cdot \sin(\theta) \cdot \dot{q}_2 - l \cdot \sin(\theta) \cdot \dot{\theta} & = 0 \\ -l \cdot \sin(\theta) \cdot \cos(\theta) \cdot \dot{q}_2 + l \cdot \sin(\theta) \cdot \cos(\theta) \cdot \dot{q}_2 & = 0 \end{cases}$$

Modèle cinématique pour un unicycle

$$\begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{pmatrix} = \frac{r}{2} \cdot \begin{pmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ \frac{1}{l} & \frac{1}{l} \end{pmatrix} \cdot \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

$$\begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_C \\ \omega \end{pmatrix} = B(X_C) \cdot U$$

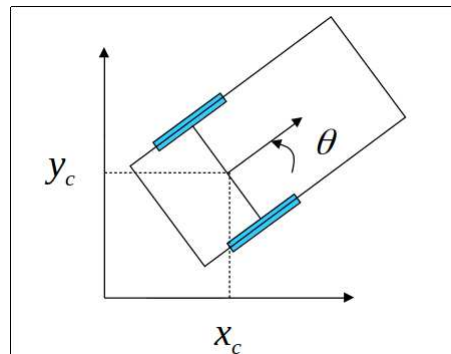
Contrainte non-holonome ?

Calculer la vitesse d'un point quelconque P du robot avec $P = (p, q)^T$ dans le repère robot. Montrer que :

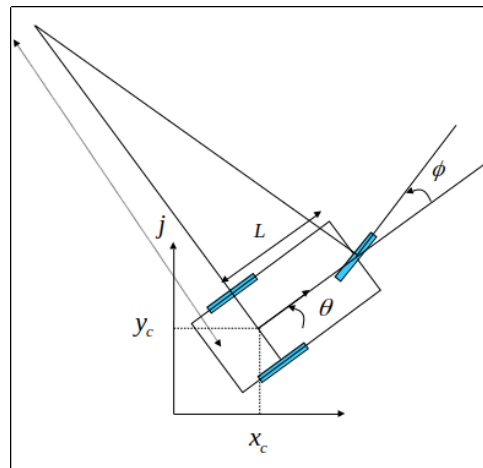
$$\begin{pmatrix} \dot{x}_p \\ \dot{y}_p \end{pmatrix} = \begin{pmatrix} 1 & 0 & (-p \sin(\theta) - q \cos(\theta)) \\ 0 & 1 & (p \cos(\theta) - q \sin(\theta)) \end{pmatrix} \cdot \begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{pmatrix} =$$

$$\begin{pmatrix} \cos(\theta) & -(p \sin(\theta) + q \cos(\theta)) \\ \sin(\theta) & (p \cos(\theta) - q \sin(\theta)) \end{pmatrix} \cdot \begin{pmatrix} v_c \\ \omega \end{pmatrix} = H(X) \cdot U$$

A quelle condition $H(X)$ est inversible ?



Modèle cinématique pour RM de type voiture



$$\begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ \frac{1}{L} \cdot \tan(\phi) & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = B(X_c) \cdot U$$

avec $-\phi_{max} \leq \phi \leq \phi_{max}$

$$r_{min} = \frac{L}{\tan(\phi_{max})}$$

Commande de RM à roues

Si on suppose qu'on connaît la trajectoire à suivre, ou la configuration à atteindre, comment faire réaliser le mouvement par le RM à l'aide de la commande des actionneurs qui l'équipe ?

Dépend de la dynamique du système.

- Un RM à roue est-il commandable ?
- Commande en boucle ouverte : peu robuste,
- Commande en boucle fermée
 - Commande stabilisante par retour d'état ?
 - Existence d'un procédé de reconstruction de l'état ?
 - incertitudes (modèle et mesure) ?
 - ...

Commande de RM à roues : commandabilité

Soit un robot mobile de type $\dot{X} = B(X).U = \sum_{i=1}^m g_i(X).u_i$ avec $X \in \mathbb{R}^n$, $U \in \mathbb{R}^m$, les g_i formant la matrice cinématique $B(X)$ de dimension $n \times m$.

Ce robot mobile est commandable si les colonnes g_i de $B(X)$ et leurs crochets de Lie successifs forment un ensemble de n colonnes indépendantes.

Dimension $[Lie(g_i)] = n$.

$$[g(q), h(q)] = \frac{\partial h}{\partial q} \cdot g(q) - \frac{\partial g}{\partial q} \cdot h(q)$$

$$\dot{X} = \begin{pmatrix} c\theta \\ s\theta \\ 0 \end{pmatrix} \cdot v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \omega = g(X) \cdot v + h(X) \cdot \omega$$

$$[g, h] = (s\theta - c\theta \ 0)^T$$

$$\text{rang}(g, h, [g, h]) = \text{rang} \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ \sin(\theta) & 0 & -\cos(\theta) \\ 0 & 1 & 0 \end{pmatrix} = 3$$

\Rightarrow système commandable (importance de cette notion)



Commande en BF : un début de piste...

Linéarisation tangente : linéarisation autour d'un point

Si le système linéarisé tangent d'un système non-linéaire est commandable alors le système non linéaire original est localement commandable et stabilisable par un retour d'état continu. La réciproque est fausse.

Differential-drive : système linéarisé tangent n'est pas commandable !

Commande par retour d'état continu

Peut-on déterminer une fonction continue de type $U = K(X)$ telle que le système $\dot{X} = B(X).K(X)$ soit asymptotiquement stable ?

($X(t) \rightarrow 0$ pour $X(0)$ dans le voisinage de 0).

Condition de Brockett

Pour un système de type $\dot{X} = \sum_{i=1}^m g_i(X).u_i$ avec $X \in \mathbb{R}^n$, $U \in \mathbb{R}^m$ cette fonction continue existe $\iff \text{rang}(g_1, \dots, g_m) = n$

Condition non vérifiée pour le differential drive et plus généralement pour les systèmes non-holonomes \implies il n'existe pas de retour d'état continu permettant de stabiliser le RM en une configuration finale fixe quelconque.

Exemples de contrôle

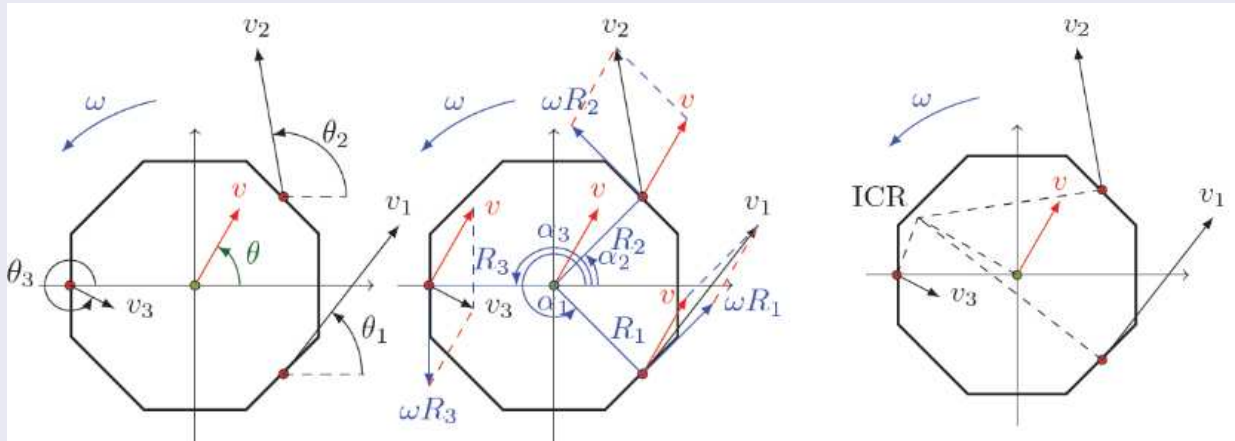
- Suivi de chemin (*path following*) : étant donné une courbe C du plan, une vitesse v , on souhaite qu'un point du robot suive C lorsque le robot roule à la vitesse v .
Exemple : conduite sur route en cherchant à maintenir une distance constante par rapport à la route
- Suivi de trajectoire (*tracking*) : la vitesse d'avancement du véhicule n'est plus fixée à l'avance car on veut aussi réguler la distance parcourue le long de C . On souhaite réguler à zéro une erreur entre un véhicule de référence (virtuel) et le RM.
Exemple : asservir le RM sur un véhicule de référence dont la trajectoire est donnée ainsi que son mouvement $(X_r(t), Y_r(t))$.
- Stabilisation en une configuration fixe : on veut se positionner et s'orienter en une configuration fixe.
Exemple : on veut garer le robot en une configuration finale sans donner la trajectoire.

Exemple de robot pour la Biennale de Venise (2015)



*Chaque roue peut s'orienter et tourner indépendamment.
Comment garantir roulement sans glissement ?*

Exemple pour le robot de la Biennale de Venise (2015)



Variable de contrôle ?

$\theta \in [0, 2\pi[$: direction du centre du robot

$v \in [0, 1]$: vitesse linéaire dans la direction θ

$\omega \in [-1, 1]$: vitesse angulaire

Pour $(\theta, v, \omega) \implies$ calculer (v_i, θ_i)