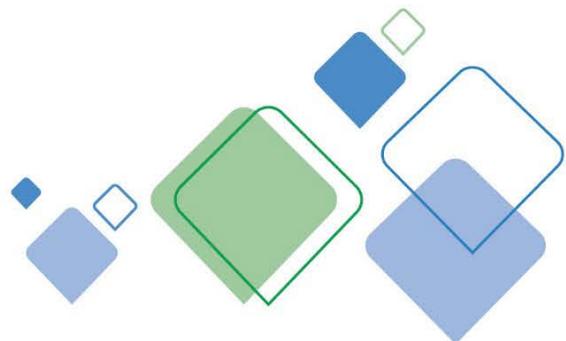


# Robotique industrielle

---

Guillaume Laurent & Nicolas Chaillet  
Option SMR  
2024





# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Définitions . . . . .	5
1.2	Composition d'un robot industriel . . . . .	6
1.3	Structures usuelles des robots industriels . . . . .	8
1.4	Conclusion . . . . .	11
<b>2</b>	<b>Repérage et transformations géométriques</b>	<b>13</b>
2.1	Repérage d'un solide . . . . .	13
2.2	Orientation déterminée par une matrice de rotation . . . . .	14
2.3	Orientation déterminée par trois angles . . . . .	16
2.4	Orientation déterminée par un vecteur et un angle . . . . .	18
2.5	Orientation déterminée par un quaternion . . . . .	19
2.6	Transformations homogènes . . . . .	20
2.7	Conclusion . . . . .	21
<b>3</b>	<b>Modèle géométrique direct</b>	<b>23</b>
3.1	Définitions et hypothèses . . . . .	23
3.2	Modélisation d'une articulation . . . . .	24
3.3	Pose de l'organe terminal . . . . .	26
3.4	Convention de Denavit-Hartenberg . . . . .	27
3.5	Domaine atteignable . . . . .	28
3.6	Conclusion . . . . .	29
<b>4</b>	<b>Modèle géométrique inverse</b>	<b>31</b>
4.1	Définition . . . . .	31
4.2	Méthode trigonométrique . . . . .	32
4.3	Méthode de Paul . . . . .	33
4.4	Méthode d'inversion pour les robots à 6 ddl à poignet sphérique . . . . .	33
4.5	Conclusion . . . . .	35
<b>5</b>	<b>Modèles cinématiques direct et inverse</b>	<b>37</b>
5.1	Définitions . . . . .	37
5.2	Jacobienne analytique . . . . .	38
5.3	Jacobienne cinématique . . . . .	38
5.4	Vitesses atteignables et manipulabilité . . . . .	42
5.5	Aspects et configurations singulières . . . . .	42
5.6	Efforts appliqués par le robot sur son environnement . . . . .	44
5.7	Conclusion . . . . .	45

<b>6</b>	<b>Modèles géométriques et cinématiques des robots parallèles</b>	<b>47</b>
6.1	Modèle géométrique inverse . . . . .	47
6.2	Modèle géométrique direct . . . . .	48
6.3	Modèle cinématique et singularités . . . . .	48
<b>7</b>	<b>Génération de mouvement</b>	<b>51</b>
7.1	Programmation d'un robot . . . . .	51
7.2	Génération de mouvement dans l'espace articulaire . . . . .	52
7.3	Génération de mouvement dans l'espace opérationnel . . . . .	56
7.4	Inversion numérique du modèle géométrique direct . . . . .	57
7.5	Conclusion . . . . .	59
<b>8</b>	<b>Modélisation dynamique et commande des axes</b>	<b>61</b>
8.1	Définition . . . . .	61
8.2	Modèle dynamique inverse . . . . .	61
8.3	Commande découplée . . . . .	64
8.4	Commande avec compensation de la gravité . . . . .	65
8.5	Commande à couples calculés . . . . .	65
8.6	Conclusion . . . . .	66
<b>9</b>	<b>Conception d'une cellule robotisée</b>	<b>67</b>
9.1	Choix du robot . . . . .	67
9.2	Choix du préhenseur ou de l'outil . . . . .	70
9.3	Placement du robot . . . . .	70
9.4	Choix des trajectoires . . . . .	71
	<b>Bibliographie</b>	<b>73</b>
<b>A</b>	<b>Sites web, blogs, réseaux d'informations et salons industriels</b>	<b>75</b>
<b>B</b>	<b>Logiciels et bibliothèques libres</b>	<b>77</b>
B.1	Logiciels . . . . .	77
B.2	Bibliothèques . . . . .	77
<b>C</b>	<b>Travaux dirigés</b>	<b>79</b>

# Chapitre 1

## Introduction

Un système robotique est une machine mécatronique programmable disposant de moyens de perception, d'interprétation et d'action qui lui permettent d'agir sur un environnement physique. Ce cours concerne plus particulièrement la robotique industrielle, c'est-à-dire les robots manipulateurs destinés à des tâches manufacturières comme la soudure, la peinture, la manutention, le conditionnement, etc. L'objectif est de fournir un ensemble de connaissances et de méthodes sur le fonctionnement d'un robot industriel, sa modélisation, sa commande et son implantation.

### 1.1 Définitions

#### 1.1.1 Définition d'un robot

Étymologiquement, le mot *robot* tire sa racine des langues slaves et renvoie au travail (*rabota* en russe, *robot* en polonais). L'écrivain tchèque Karel Capek a popularisé le terme dans les années 1920, à travers une pièce de théâtre intitulée « *Rossum's Universal Robots* » qui a connu un certain succès, en particulier en France. Elle met en scène des êtres artificiels à l'apparence humaine, qui finissent par se révolter et anéantir l'humanité...

Isaac Asimov, dans les années 1940, pensait lui au contraire, que les robots seraient bénéfiques et que les ingénieurs programmeraient une certaine bienveillance dans tous les robots. Asimov nomma l'industrie des robots la *robotique* et légua aux roboticiens une « éthique » de la robotique au travers de ses célèbres lois de la robotique.

De nos jours, on peut définir un robot comme un système mécatronique pouvant se substituer à l'homme (ou à l'animal) pour effectuer certaines opérations, et capable d'en modifier de lui-même le cycle en appréhendant son environnement.

Pour réagir de façon adaptée, le robot doit percevoir des informations sur lui-même et sur son environnement par l'intermédiaire de capteurs. On distingue généralement les capteurs proprioceptifs qui le renseigne sur son état interne (positions, forces, intensités, etc.) et les capteurs extéroceptifs qui l'informe sur son environnement (attente d'un convoyeur, qu'un opérateur appuie sur un bouton, utilisation de la vision pour localiser des objets, etc.). En fonction des informations qu'il reçoit, le robot doit prendre une décision pour accomplir sa tâche (atteindre une destination, maintenir une force, éviter les obstacles, etc.). Le robot agit sur ses mouvements et son environnement à l'aide de ses actionneurs.

Un robot est donc un système bouclé dont le fonctionnement peut être décrit par le schéma général « perception-décision-action ». Cependant, il n'est pas toujours aisé d'établir une frontière claire entre robots et les autres systèmes automatisés tant les applications sont maintenant diversifiées.

### 1.1.2 Robotique industrielle

Le concept de *robot industriel* n'a vraiment pris forme que lorsque les ingénieurs se sont rendus compte de la difficulté qu'il y avait à donner à une machine le comportement et l'apparence d'un être humain. Les efforts ont donc été concentrés sur la conception de bras mécaniques en vue d'exécuter des tâches industrielles. La norme ISO 8373:2021 fournit un glossaire de termes et de définitions relatifs à la robotique [8].

**Définition 1** (Robot industriel). *Mécanisme à commande automatique, reprogrammable, multiapplication, pouvant être programmé suivant trois axes ou plus, pouvant être fixe ou fixé sur une plateforme mobile et destiné à être utilisé dans les applications d'automatisation dans un environnement industriel (norme ISO 8373:2021).*

Un *axe* est une liaison motorisée entre deux segments successifs d'un robot. Le terme *reprogrammable* signifie que les mouvements et fonctions auxiliaires peuvent être changés sans nécessiter d'altérations physiques. Enfin, un robot industriel peut être utilisé pour différentes tâches industrielles et est donc « multiapplication ».

### 1.1.3 Robotique de service

Si la robotique est restée pendant longtemps relativement confinée à des applications industrielles manufacturières, les avancées technologiques font que de nombreux autres domaines sont explorés, en vue notamment de développer une *robotique de service*.

**Définition 2** (Robot de service). *Robot à usage personnel ou professionnel qui exécute des tâches utiles pour des humains ou des appareillages (norme ISO 8373:2021).*

On peut citer ici quelques domaines d'utilisation des robots de service : robotique médicale (chirurgie téléopérée), robotique de manutention, robotique d'exploration et de surveillance (terrestre, sous-marine, aérienne, spatiale), robotique pour la sécurité civile, robotique agricole, robotique minière, robotique militaire (déminage, soldat robotisé, drone), robotique de chantier (ou du bâtiment), robotique de nettoyage et d'entretien, robotique de divertissements (jouets, robots de compagnie), robotique d'aide à la personne (assistances, prothèses), etc.

## 1.2 Composition d'un robot industriel

### 1.2.1 Éléments d'un robot

Les différents éléments d'un robot industriel sont (voir figure 1.1) :

- *base* : support sur lequel est situé l'origine du premier élément de la structure articulée constituant le bras,
- *bras* ou *porteur* : chaîne articulée qui permet de déplacer le poignet dans l'espace,
- *poignet* : chaîne articulée fixé à l'extrémité du bras d'un robot permettant l'orientation de l'organe terminal,
- *axe* : liaison motorisée en translation (glissière) ou en rotation (pivot) entre deux segments successifs d'un robot,
- *organe terminal* : outil ou préhenseur destiné à effectuer une tâche.

En fonction du nombre d'axes et de la cinématique du robot, celui-ci n'est pas forcément en mesure de générer les 6 degrés de liberté de l'organe terminal. Notamment, un robot avec moins de 6 axes ne peut pas générer six degrés de liberté (DDL) indépendants au niveau de son effecteur.

Posséder au minimum six axes est une condition nécessaire mais non suffisante car on peut trouver des robots *redondants* possédant plus de 6 axes et générant moins de 6 degrés de liberté

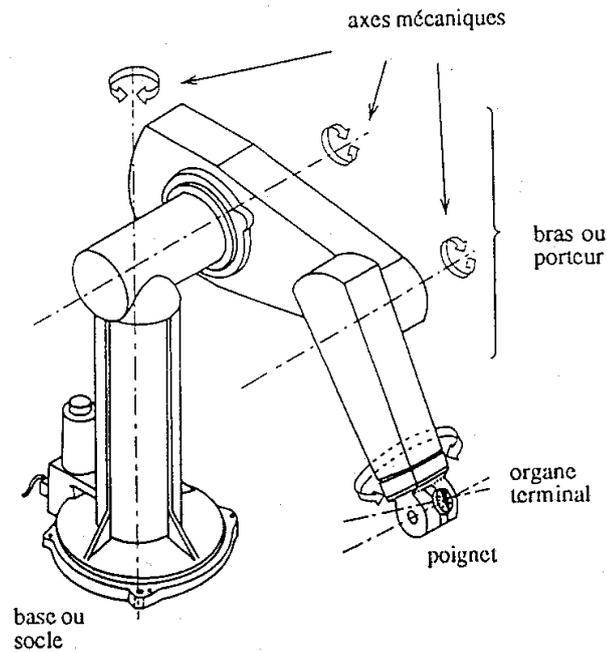


FIGURE 1.1 – Éléments d'un robot industriel (illustration extraite de [2]).

au niveau de son effecteur. D'autre part, l'espace cartésien comportant 6 degrés de liberté, un robot, quel que soit le nombre de ses axes, ne pourra générer au maximum que 6 degrés de liberté au niveau de son organe terminal.

### 1.2.2 Chaîne de commande d'un robot industriel

La chaîne de commande d'un robot industriel est composée des éléments suivants :

- le *calculateur* est chargé de calculer les consignes de couple ou de vitesse des actionneurs à partir du programme défini par l'utilisateur. Il est constitué d'une unité centrale, d'une unité de stockage, d'interfaces de communications (série, ethernet, etc.), et de ports d'entrées/sorties (CAN, CNA, E/S numériques, compteurs, etc.).
- les *actionneurs* mettent la structure en mouvement. Il s'agit le plus souvent d'actionneurs électromagnétiques mais des actionneurs hydrauliques peuvent être utilisés pour des robots portant de très fortes charges. Les éléments périphériques du robot (préhenseurs mécaniques, préhenseurs par dépression, etc.) utilisent principalement des actionneurs pneumatiques.
- les *variateurs* (ou *drivers*) assurent la régulation du couple ou de la vitesse des moteurs électromagnétiques (généralement par une régulation du courant). Les variateurs gèrent aussi la sécurité « bas niveau » du robot (protection contre les surcharges, protection contre une défaillance interne du variateur, gestions des fins de course, gestions des arrêts d'urgences).
- les *transmissions* transmettent le mouvement des actionneurs à la structure (avec généralement un facteur de réduction important).
- la *structure mécanique* réalise le mouvement.
- les *capteurs proprioceptifs* mesurent des grandeurs internes au robot (positions, vitesses, couples, forces).
- des *capteurs extéroceptifs* peuvent renseigner le calculateur sur son environnement (ouverture d'un porte, etc.) ou permettre de réaliser une commande référencée capteur (commande en force, asservissement visuel, etc.).

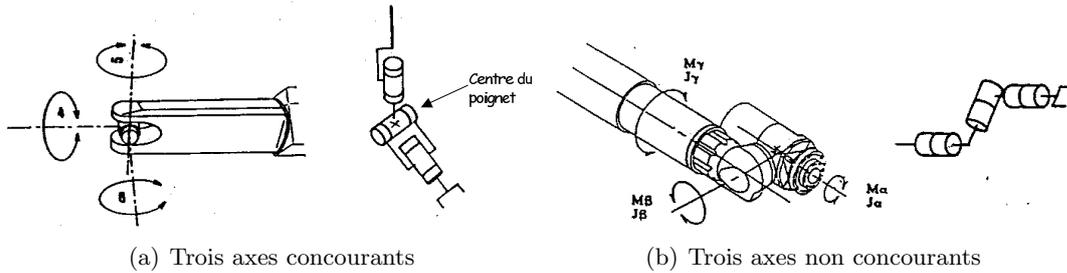


FIGURE 1.2 – Structures mécaniques du poignet (illustrations extraites de [2]).

## 1.3 Structures usuelles des robots industriels

### 1.3.1 Robots séries

La majorité des robots manipulateurs existant à l'heure actuelle présente un caractère anthropomorphe, en l'occurrence une forte ressemblance avec le bras humain.

**Définition 3** (Robot série). *Un robot série est constitué d'une succession de segments en série, chacun étant relié à son prédécesseur et à son successeur par une articulation à une mobilité, générant ainsi une chaîne cinématique ouverte.*

Les articulations des robots séries sont généralement soit des liaisons pivots (en anglais *revolute joint*, symbolisée par la lettre R) soit des liaisons glissières (*prismatic joint*, symbolisée par la lettre P). Ainsi, ils peuvent être classifiés en fonction de leur structure mécanique et du type des axes :

- PPP : robot cartésien/portique (en anglais *cartesian/gantry robot*),
- RPP : robot cylindrique,
- RRPR : robot SCARA (Selective Compliance Articulated Robot Arm),
- RRRRRR : robot anthropomorphe à poignet sphérique (*articulated robot*).

Ces structures sont illustrées dans le tableau 1.1.

Parmi toutes les structures possibles de porteur, on cherche surtout celles qui, à l'instar du bras humain, cherchent à découpler le mouvement de l'effecteur en un mouvement l'amenant à une position donnée (bras à 3 ddl) et un mouvement l'orientant convenablement à cette position (poignet à 3 ddl). Notons également qu'il existe des robots redondants à 7 ddl voire hyper-redondants (en « trompes d'éléphant ») qui peuvent être intéressants dans des cas où l'accès est rendu difficile par la présence d'obstacles (par exemple, l'inspection à l'intérieur d'un moteur).

**Définition 4** (Robot redondant). *Un robot est dit redondant lorsque le nombre de degrés de liberté de l'organe terminal (dans l'espace opérationnel) est inférieur au nombre d'articulations motorisées.*

### 1.3.2 Poignets

Suivant la tâche envisagée, on trouve des poignets à un, deux ou trois axes. De nombreuses tâches industrielles, comme la prise-dépose, peuvent être correctement réalisées avec un robot à 4 axes (3+1). Néanmoins, la majorité des robots industriels ont 6 axes (3+3). La structure de poignet à trois axes la plus courante est constituée de trois articulations pivot en série et à axes concourants et orthogonaux deux à deux (voir figure 1.2). Cette structure est quasiment équivalente à une liaison rotule (singularité mise à part). Elle permet donc d'obtenir une orientation quelconque de l'outil et un découplage par rapport au porteur, d'où une simplification

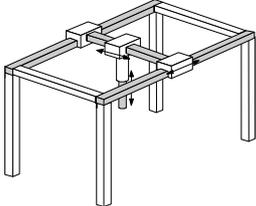
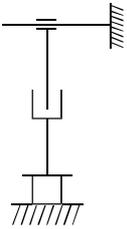
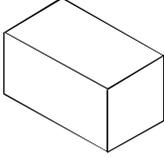
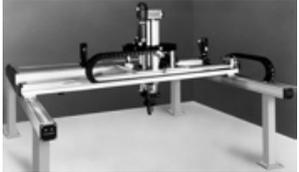
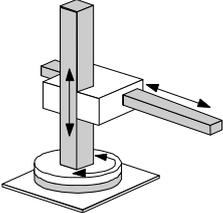
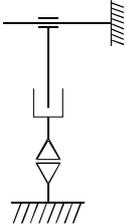
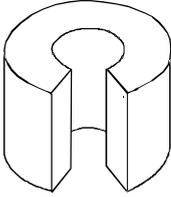
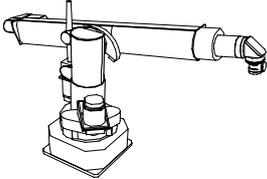
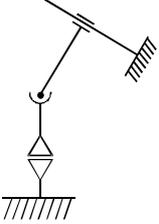
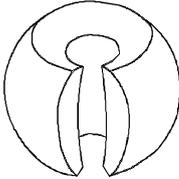
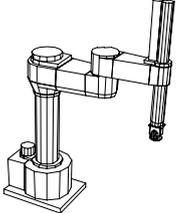
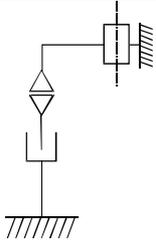
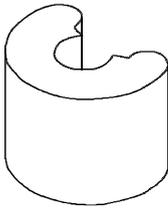
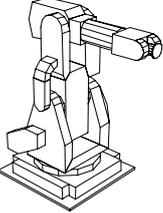
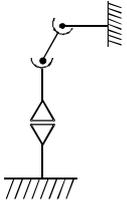
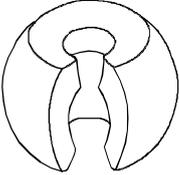
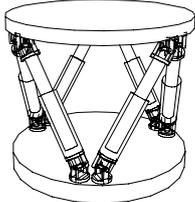
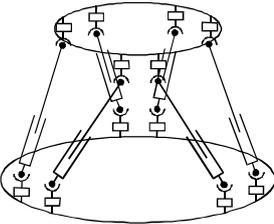
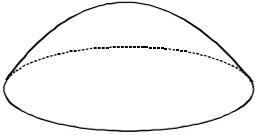
Robot	Axes		Examples
	Kinematic Structure	Workspace	
 Cartesian Robot			
 Cylindrical Robot			
 Spherical Robot			
 SCARA Robot			
 Articulated Robot			
 Parallel Robot			

TABLE 1.1 – Classification des robots industriels par structure mécanique (norme ISO 8373).

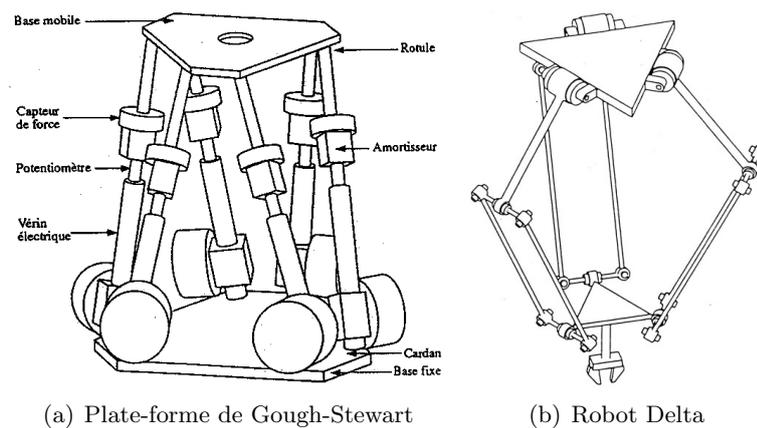


FIGURE 1.3 – Deux exemples de porteurs parallèles (illustrations extraites de [2]).

de la commande : le centre du poignet est fixe par rapport au bras et sa position ne dépend que des trois premiers axes (ceux du porteur). Un poignet à axes non concourants a l'avantage de permettre des débattements angulaires plus importants par un montage en « porte-à-faux » du deuxième axe.

### 1.3.3 Robots parallèles

**Définition 5** (Robot parallèle). *Un robot parallèle est caractérisé par le fait que l'organe terminal est relié à la base du robot par au moins deux chaînes cinématiques distinctes formant des boucles cinématiques fermées. Il faut noter que dans de telles structures cinématiques, toutes les articulations ne sont pas actives : certaines sont nécessairement passives.*

Le robot parallèle le plus célèbre est la plateforme de Gough-Stewart (1965) illustré par la figure 1.3. Ce type de structure était notamment utilisé à l'origine dans des simulateurs de vol pour l'entraînement des pilotes. On qualifie la cinématique plateforme de Gough-Stewart de structure  $6(\underline{U}S)$ , où U désigne une liaison cardan (*universal joint*) et S une liaison rotule (*spherical joint*). La lettre soulignée désigne les articulations actives.

Développé dans les années 1980 par Clavel, le robot Delta (EPFL/ABB), dédié à la prise-dépose rapide, connaît le succès industriel à partir des années 1990. La plateforme du robot Delta a 3 degrés de liberté en translation et est actionné par 3 axes rotatifs. Sa structure cinématique est  $3(\underline{R}2(SS))$ .

En dépit du fait que les débattements de ces robots sont plus faibles que ceux des robots séries, générant donc un volume de travail plus faible, les robots parallèles offrent d'autres avantages :

- les erreurs de position (ainsi que les imprécisions des capteurs, les défauts divers : jeux, frottements, flexion des segments) se produisent en parallèle au lieu de se cumuler en série. Ces robots ont donc par nature une meilleure répétabilité.
- une telle architecture permet une meilleure répartition des efforts et un montage des actionneurs sur la base, autorisant une structure mécanique plus légère et donc des mouvements rapides, de grandes accélérations et un rapport charge utile sur masse totale mouvement plus favorable (dans le cas des robots série ce rapport est en général très largement inférieur à 0,3) ;
- cela permet également d'augmenter la rigidité de la structure (beaucoup moins de sollicitation en flexion des segments, peu de déformation) afin de porter des charges importantes comme la cabine d'un simulateur de vol ;

- l'utilisation de chaînes parallèles permet d'utiliser des composants identiques (moteurs, constituants mécaniques), d'où une facilitation de la maintenance et une réduction des coûts.

Les applications industrielles des structures parallèles sont nombreuses, aussi bien en robotique (notamment pour la prise-dépose) qu'en machines outils. Elles sont également utilisées pour les applications nécessitant des précisions importantes. Il est également possible d'utiliser une structure parallèle comme poignet de robot afin d'augmenter sa rigidité ou de déporter les actionneurs.

### 1.3.4 Robots arborescents

**Définition 6** (Robot arborescent). *Un robot à chaîne cinématique arborescente est constitué de plusieurs chaînes cinématiques ouvertes reliées à une base commune.*

Ce type de structure est davantage utilisé pour les robots de service (robots marcheurs, robot humanoïde) mais depuis les années 2010 des robots manipulateurs industriels à deux bras sont proposés pour réaliser des tâches de manipulations complexes.

En fonctionnement, un robot arborescent peut devenir un robot à chaîne cinématique fermée par la saisie d'une pièce avec deux bras ou le contact avec le sol de deux ou plusieurs pattes.

## 1.4 Conclusion

Les améliorations techniques des robots industriels ont été considérables, et ce pour un prix globalement divisé par plus de 4 en 20 ans. La durée de vie moyenne d'un robot se situe entre 12 et 15 ans. Le retour sur investissement est très court, souvent d'un à trois ans. Un grand constructeur japonais d'équipements de production en général et de robots en particulier, indique que le robot est l'équipement de production qui présente la plus grande fiabilité ! Il est aujourd'hui souvent intéressant et efficace de penser « cellule robotisée » avant d'investir dans une machine spéciale.

La tendance actuelle est le travail collaboratif entre les hommes et les robots, avec les fonctions de sécurité associées (ce qui permet d'éliminer les barrières de sécurité sur les lignes de production). Un point essentiel est de conférer aux robots plus d'autonomie décisionnelle s'accompagnant de l'exploitation de capteurs extéroceptifs, notamment la vision par ordinateur et les lidars pour la perception de l'environnement. Cette autonomie permettra aux fabricants de robots de conquérir de nouveaux marchés notamment dans la robotique de service.

Il existe également une tendance qui consiste à aller vers les extrêmes, c'est-à-dire vers le développement de robots de plus en plus petits (microrobots permettant la manipulation à grande précision d'objets très petits [12]) et les robots de plus en plus grands pour le génie civil ou naval.

Une ressource intéressante pour suivre l'évolution des marchés de la robotique est le rapport annuel « World Robotics » de l'IFR (International Federation of Robotics) qui fournit des données sur la vente et l'installation de robots dans le monde. Une synthèse de ce rapport est disponible sur <https://ifr.org/>.



## Chapitre 2

# Repérage et transformations géométriques

Ce chapitre présente les outils mathématiques permettant de décrire la position et l'orientation d'un repère dans un espace cartésien ainsi que les transformations entre repères. Comme dans le domaine de la vision 3D, il est courant en robotique d'utiliser le terme *pose* pour désigner à la fois la position et l'orientation d'un repère. La position d'un repère est définie par les coordonnées de son origine tandis que son orientation peut être définie par une matrice de rotation, par trois angles, par un vecteur et un angle ou encore par un quaternion. En robotique, on utilise généralement les transformations homogènes qui permettent de décrire la pose d'un repère à l'aide d'une seule matrice de dimension 4.

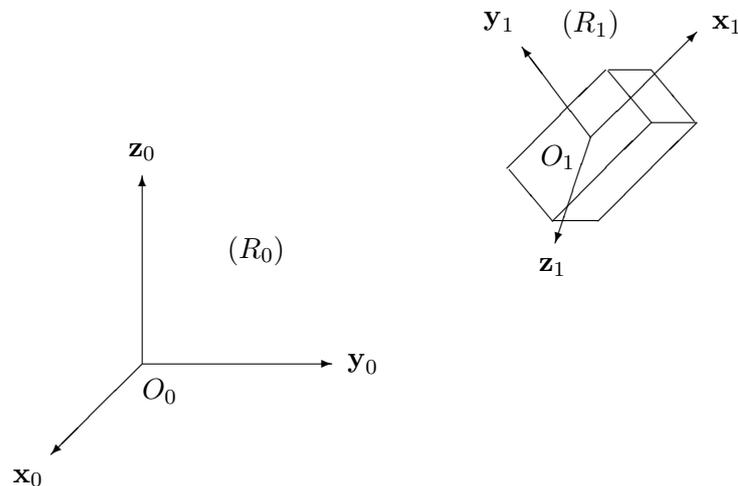
Il est également d'usage de respecter les conventions de notations suivantes :

- les scalaires en caractères minuscules italiques, par exemple  $x$  et  $t$ ,
- les points en caractères majuscules italiques, par exemple  $P$  et  $A$ ,
- les vecteurs en caractères minuscules gras, par exemple  $\mathbf{x}$  et  $\mathbf{u}$ ,
- les matrices en caractères majuscules gras, par exemple  $\mathbf{M}$  et  $\mathbf{R}$ .

L'exposant qui précède un point ou un vecteur désigne le repère dans lequel ses coordonnées sont exprimées, comme par exemple  ${}^1P$  et  ${}^1\mathbf{x}$ .

### 2.1 Repérage d'un solide

Soit un solide  $S_1$  quelconque. On associe au solide un repère  $R_1$ , formé d'une base orthonormée directe  $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1)$  et d'une origine  $O_1$ . On définit également un repère fixe dans l'espace  $R_0 = (O_0, \mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$ , tel qu'indiqué sur le schéma suivant :



Soit  $P$  un point du solide  $S_1$ . Les coordonnées de  $P$  dans le repère  $R_1$  sont représentées par le vecteur  $\overrightarrow{O_1P}$  tel que :

$${}^1P = \overrightarrow{O_1P}_{/R_1} = \begin{bmatrix} \overrightarrow{O_1P} \cdot \mathbf{x}_1 \\ \overrightarrow{O_1P} \cdot \mathbf{y}_1 \\ \overrightarrow{O_1P} \cdot \mathbf{z}_1 \end{bmatrix}$$

La position du repère  $R_1$  dans l'espace peut alors s'exprimer par l'intermédiaire de la position du point  $O_1$  dans  $R_0$ , c'est-à-dire par les coordonnées du vecteur  $\overrightarrow{O_0O_1}$  dans  $R_0$  :

$${}^0O_1 = \overrightarrow{O_0O_1}_{/R_0} = \begin{bmatrix} \overrightarrow{O_0O_1} \cdot \mathbf{x}_0 \\ \overrightarrow{O_0O_1} \cdot \mathbf{y}_0 \\ \overrightarrow{O_0O_1} \cdot \mathbf{z}_0 \end{bmatrix}$$

## 2.2 Orientation déterminée par une matrice de rotation

### 2.2.1 Définition

Soit  $P$  un point du solide  $S_1$ . On note  $a$ ,  $b$  et  $c$  ses coordonnées exprimées dans le repère  $R_1$ . On a alors :

$${}^1P = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = a\mathbf{x}_1 + b\mathbf{y}_1 + c\mathbf{z}_1$$

Si les origines des deux repères  $R_0$  et  $R_1$  sont confondues, on peut substituer le vecteur  $\overrightarrow{O_0P}$  par  $\overrightarrow{O_1P}$ . Puis en exprimant les coordonnées du vecteur  $\overrightarrow{O_0P}$  dans le repère  $R_0$ , il vient :

$${}^0P = \overrightarrow{O_0P}_{/R_0} = \overrightarrow{O_1P}_{/R_0} = \begin{bmatrix} \overrightarrow{O_1P} \cdot \mathbf{x}_0 \\ \overrightarrow{O_1P} \cdot \mathbf{y}_0 \\ \overrightarrow{O_1P} \cdot \mathbf{z}_0 \end{bmatrix} = \begin{bmatrix} (a\mathbf{x}_1 + b\mathbf{y}_1 + c\mathbf{z}_1) \cdot \mathbf{x}_0 \\ (a\mathbf{x}_1 + b\mathbf{y}_1 + c\mathbf{z}_1) \cdot \mathbf{y}_0 \\ (a\mathbf{x}_1 + b\mathbf{y}_1 + c\mathbf{z}_1) \cdot \mathbf{z}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \cdot \mathbf{x}_0 & \mathbf{y}_1 \cdot \mathbf{x}_0 & \mathbf{z}_1 \cdot \mathbf{x}_0 \\ \mathbf{x}_1 \cdot \mathbf{y}_0 & \mathbf{y}_1 \cdot \mathbf{y}_0 & \mathbf{z}_1 \cdot \mathbf{y}_0 \\ \mathbf{x}_1 \cdot \mathbf{z}_0 & \mathbf{y}_1 \cdot \mathbf{z}_0 & \mathbf{z}_1 \cdot \mathbf{z}_0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

soit :

$${}^0P = {}^0\mathbf{M}_1 \cdot {}^1P$$

La matrice  ${}^0\mathbf{M}_1$  est la matrice de rotation (également appelée matrice des cosinus directeurs) correspondant à la matrice de passage de  $R_0$  à  $R_1$ . Les colonnes de cette matrice sont les composantes des vecteurs de la base de  $R_1$  exprimés dans  $R_0$  :

$${}^0\mathbf{M}_1 = \begin{bmatrix} \mathbf{x}_1 \cdot \mathbf{x}_0 & \mathbf{y}_1 \cdot \mathbf{x}_0 & \mathbf{z}_1 \cdot \mathbf{x}_0 \\ \mathbf{x}_1 \cdot \mathbf{y}_0 & \mathbf{y}_1 \cdot \mathbf{y}_0 & \mathbf{z}_1 \cdot \mathbf{y}_0 \\ \mathbf{x}_1 \cdot \mathbf{z}_0 & \mathbf{y}_1 \cdot \mathbf{z}_0 & \mathbf{z}_1 \cdot \mathbf{z}_0 \end{bmatrix} = [{}^0\mathbf{x}_1 \quad {}^0\mathbf{y}_1 \quad {}^0\mathbf{z}_1]$$

Notons que les 3 lignes sont les composantes des vecteurs de la base de  $R_0$  exprimés dans  $R_1$  soit respectivement  ${}^0\mathbf{x}_1$ ,  ${}^0\mathbf{y}_1$  et  ${}^0\mathbf{z}_1$ . Chaque élément de la matrice est nommé *cosinus directeur*.

### 2.2.2 Propriétés

Par définition, les lignes et les colonnes sont des vecteurs unitaires. De plus, les lignes (resp. colonnes) représentent des vecteurs perpendiculaires 2 à 2, donc de produit scalaire nul. Une ligne (resp. colonne) est le produit vectoriel des 2 autres (en respectant le sens direct). Les coefficients de  ${}^0\mathbf{M}_1$  sont donc liés par des équations de contrainte. Ils ne sont donc pas indépendants et l'information fournie par  ${}^0\mathbf{M}_1$  sur l'orientation de  $R_1$  est ainsi exprimée de façon redondante et implicite (les angles donnant l'orientation n'apparaissent pas directement).

Exprimons à présent  ${}^1\mathbf{M}_0$ , la matrice de passage de  $R_1$  à  $R_0$  :

$${}^1\mathbf{M}_0 = \begin{bmatrix} \mathbf{x}_0 \cdot \mathbf{x}_1 & \mathbf{y}_0 \cdot \mathbf{x}_1 & \mathbf{z}_0 \cdot \mathbf{x}_1 \\ \mathbf{x}_0 \cdot \mathbf{y}_1 & \mathbf{y}_0 \cdot \mathbf{y}_1 & \mathbf{z}_0 \cdot \mathbf{y}_1 \\ \mathbf{x}_0 \cdot \mathbf{z}_1 & \mathbf{y}_0 \cdot \mathbf{z}_1 & \mathbf{z}_0 \cdot \mathbf{z}_1 \end{bmatrix} = [{}^1\mathbf{x}_0 \quad {}^1\mathbf{y}_0 \quad {}^1\mathbf{z}_0]$$

Il apparaît alors que :

$${}^1\mathbf{M}_0 = ({}^0\mathbf{M}_1)^T$$

En outre, comme les vecteurs de base de chaque repère sont perpendiculaires 2 à 2, si l'on calcule  ${}^1\mathbf{M}_0 \cdot {}^0\mathbf{M}_1$  et  ${}^0\mathbf{M}_1 \cdot {}^1\mathbf{M}_0$ , il vient :

$${}^1\mathbf{M}_0 \cdot {}^0\mathbf{M}_1 = {}^0\mathbf{M}_1 \cdot {}^1\mathbf{M}_0 = \mathbf{I}_3 \quad (\text{matrice identité})$$

D'où :

$$\boxed{{}^1\mathbf{M}_0 = ({}^0\mathbf{M}_1)^T = ({}^0\mathbf{M}_1)^{-1}}$$

On peut également noter que :

$$\det {}^0\mathbf{M}_1 = \det {}^1\mathbf{M}_0 = 1$$

Outre le fait d'exprimer l'orientation d'un repère par rapport à un autre, la matrice  ${}^0\mathbf{M}_1$  de passage de  $R_0$  à  $R_1$  permet d'exprimer les coordonnées d'un vecteur  $\mathbf{v}$  dans le repère  $R_0$  à partir de ses coordonnées connues dans le repère  $R_1$  :

$$\boxed{{}^0\mathbf{v} = {}^0\mathbf{M}_1 \cdot {}^1\mathbf{v}}$$

A l'inverse, on peut bien entendu écrire :

$${}^1\mathbf{v} = {}^1\mathbf{M}_0 \cdot {}^0\mathbf{v}$$

Enfin, on démontre la relation de composition. Soit un vecteur quelconque  $\mathbf{v}$  dans l'espace.  ${}^i\mathbf{M}_j$  est telle que  ${}^i\mathbf{v} = {}^i\mathbf{M}_j \cdot {}^j\mathbf{v}$ . En utilisant deux fois cette relation, on peut écrire que :

$${}^0\mathbf{v} = {}^0\mathbf{M}_1 \cdot {}^1\mathbf{v} = {}^0\mathbf{M}_1 \cdot {}^1\mathbf{M}_2 \cdot {}^2\mathbf{v}$$

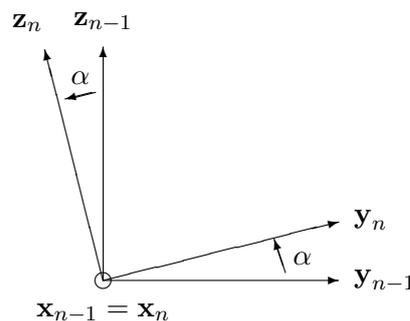
On en déduit aisément la relation de composition suivant :

$$\boxed{{}^0\mathbf{M}_2 = {}^0\mathbf{M}_1 \cdot {}^1\mathbf{M}_2}$$

### 2.2.3 Matrices de rotation usuelles

Si la rotation d'un repère  $R_n$  par rapport à un repère  $R_{n-1}$  s'effectue autour d'un des axes de base de  $R_{n-1}$ , il est facile d'écrire la matrice de passage entre  $R_{n-1}$  et  $R_n$ .

#### Rotation autour de l'axe $\mathbf{x}_{n-1}$ (quand $\mathbf{x}_n = \mathbf{x}_{n-1}$ )



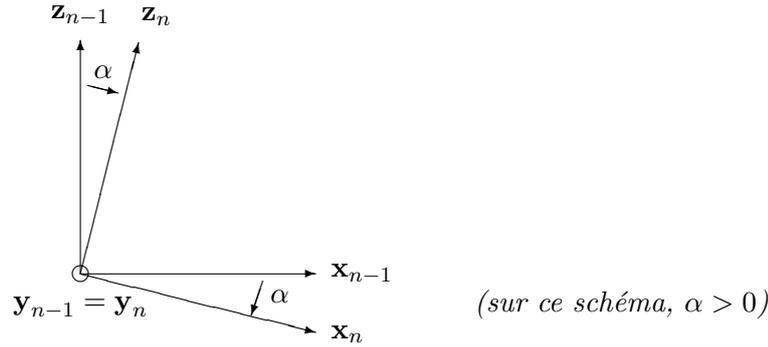
(sur ce schéma,  $\alpha > 0$ )

$${}^{n-1}\mathbf{M}_n = \text{rot}(\mathbf{x}_{n-1}, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha \\ 0 & s_\alpha & c_\alpha \end{bmatrix}$$

**NB:**  $c_\alpha$  désigne  $\cos(\alpha)$  et  $s_\alpha$  désigne  $\sin(\alpha)$ .

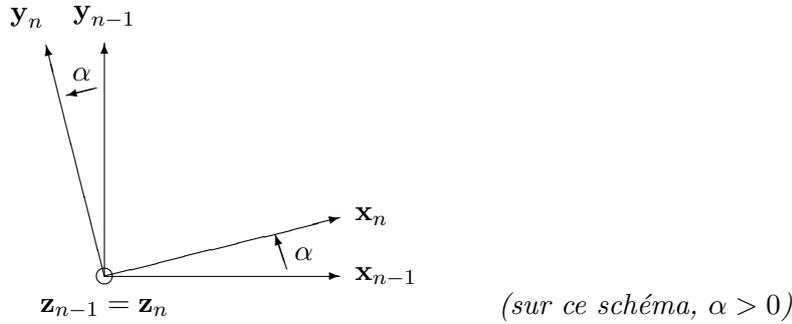
**NB:** Les rotations sont définies dans le sens direct.

**Rotation autour de l'axe  $\mathbf{y}_{n-1}$  (quand  $\mathbf{y}_n = \mathbf{y}_{n-1}$ )**



$${}^{n-1}\mathbf{M}_n = \text{rot}(\mathbf{y}_{n-1}, \alpha) = \begin{bmatrix} c_\alpha & 0 & s_\alpha \\ 0 & 1 & 0 \\ -s_\alpha & 0 & c_\alpha \end{bmatrix}$$

**Rotation autour de l'axe  $\mathbf{z}_{n-1}$  (quand  $\mathbf{z}_n = \mathbf{z}_{n-1}$ )**



$${}^{n-1}\mathbf{M}_n = \text{rot}(\mathbf{z}_{n-1}, \alpha) = \begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

### 2.3 Orientation déterminée par trois angles

On peut également exprimer l'orientation de  $R_1$  par rapport à  $R_0$  de manière explicite et non redondante, par trois angles. Les rotations peuvent être définies de différentes manières (à condition que deux rotations consécutives ne soient pas autour du même axe). Pour ce faire, on définit  $R_1 = (O_1, \mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1)$  ayant même origine que  $R_0$ .  $R_1$  peut se déduire de  $R_0$  par trois rotations successives, en définissant deux repères intermédiaires  $R'$  et  $R''$ . La première rotation fait passer de  $R_0$  à  $R'$ , la seconde de  $R'$  à  $R''$  et la troisième de  $R''$  à  $R_1$ . Les rotations autour d'axes fixes par rapport à  $R_0$  sont appelées *rotations extrinsèques*. Les rotations autour d'axes obtenus par une précédente rotation sont appelées *rotations intrinsèques* ou *rotations propres*.

Ainsi, pour exprimer de cette manière l'orientation d'un solide, on peut choisir parmi les diverses possibilités celle qui permettra l'expression de l'orientation la plus simple selon l'application considérée. En robotique industrielle, les angles de Tait-Bryan sont très courants car ils présentent l'avantage d'être plus intuitifs pour un opérateur se tenant à côté d'un robot.

### 2.3.1 Angles de Tait-Bryan (lacet, tangage, roulis)

Les angles de Tait-Bryan, aussi appelés angles nautiques définissent habituellement les trois rotations intrinsèques suivantes :

- le lacet (yaw)  $\alpha$ , autour de l'axe  $\mathbf{z}_0$ , fait passer de  $R_0$  à  $R'$ ,
- le tangage (pitch)  $\beta$ , autour de l'axe  $\mathbf{y}'$ , fait passer de  $R'$  à  $R''$ ,
- le roulis (roll)  $\gamma$ , autour de l'axe  $\mathbf{x}''$ , fait passer de  $R''$  à  $R_1$ .

$$\begin{array}{ccccccc} \text{rot}(\mathbf{z}_0, \alpha) & & \text{rot}(\mathbf{y}', \beta) & & \text{rot}(\mathbf{x}'', \gamma) & & \\ R_0 & \longrightarrow & R' & \longrightarrow & R'' & \longrightarrow & R_1 \end{array}$$

On choisit généralement, les angles nautiques dans les intervalles suivants :

$$-\pi \leq \alpha < \pi \quad ; \quad -\frac{\pi}{2} \leq \beta < \frac{\pi}{2} \quad ; \quad -\pi \leq \gamma < \pi.$$

Malgré cela, la représentation n'est pas toujours unique. En effet, si  $\beta = \pm \frac{\pi}{2}$ , les angles  $\alpha$  et  $\gamma$  jouent le même rôle et la configuration comporte une singularité appelée blocage de cardan (*gimbal lock*).

La matrice de rotation équivalente à une orientation donnée par les angles de Tait-Bryan est :

$$\begin{aligned} \mathbf{M}(\alpha, \beta, \gamma) &= \text{rot}(\mathbf{z}_0, \alpha) \cdot \text{rot}(\mathbf{y}', \beta) \cdot \text{rot}(\mathbf{x}'', \gamma) = \begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{bmatrix} \\ &= \begin{bmatrix} c_\alpha c_\beta & -s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma & s_\alpha s_\gamma + c_\alpha s_\beta c_\gamma \\ s_\alpha c_\beta & c_\alpha c_\gamma + s_\alpha s_\beta s_\gamma & -c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \end{aligned}$$

Inversement, connaissant une matrice de rotation  $\mathbf{M}$ , il est possible de calculer les angles de Tait-Bryan :

- si  $m_{11} = 0$  et  $m_{21} = 0$ ,

$$\begin{cases} \alpha &= 0 \\ \beta &= \text{atan2}(-m_{31}, 0) \\ \gamma &= \text{atan2}(-m_{23}, m_{22}) \end{cases}$$

- sinon,

$$\begin{cases} \alpha &= \text{atan2}(m_{21}, m_{11}) \\ \beta &= \text{atan2}(-m_{31}, c_\alpha m_{11} + s_\alpha m_{21}) \\ \gamma &= \text{atan2}(s_\alpha m_{13} - c_\alpha m_{23}, c_\alpha m_{22} - s_\alpha m_{12}) \end{cases}$$

**NB:** La fonction  $\text{atan2}$  est une fonction réciproque de la tangente à deux arguments définie telle que :

$$\begin{aligned} \text{atan2} : \quad \mathbb{R}^2 &\rightarrow ]-\pi; \pi] \\ (y, x) &\mapsto \text{atan2}(y, x) \end{aligned}$$

Dans certains ouvrages ou interfaces de programmation, les angles nautiques sont définis dans l'ordre RTL et non LTR comme il vient d'être présenté. Les rotations sont alors les suivantes :

$$\begin{array}{ccccccc} \text{rot}(\mathbf{x}_0, \gamma) & & \text{rot}(\mathbf{y}', \beta) & & \text{rot}(\mathbf{z}'', \alpha) & & \\ R_0 & \longrightarrow & R' & \longrightarrow & R'' & \longrightarrow & R_1 \end{array}$$

### 2.3.2 Angles d'Euler (précession, nutation, rotation propre)

Les angles d'Euler définissent habituellement les trois rotations intrinsèques suivantes :

- la précession  $\psi$ , autour de l'axe  $\mathbf{z}_0$ , fait passer de  $R_0$  à  $R'$ ,
- la nutation  $\theta$ , autour de l'axe  $\mathbf{x}'$ , fait passer de  $R'$  à  $R''$ ,
- la rotation propre  $\phi$ , autour de l'axe  $\mathbf{z}''$ , fait passer de  $R''$  à  $R_1$ .

$$\begin{array}{ccccccc} \text{rot}(\mathbf{z}_0, \psi) & & \text{rot}(\mathbf{x}', \theta) & & \text{rot}(\mathbf{z}'', \phi) & & \\ R_0 & \longrightarrow & R' & \longrightarrow & R'' & \longrightarrow & R_1 \end{array}$$

La matrice de rotation équivalente à une orientation donnée par les angles d'Euler est :

$$\mathbf{M}(\psi, \theta, \phi) = \text{rot}(\mathbf{z}_0, \psi) \cdot \text{rot}(\mathbf{x}', \theta) \cdot \text{rot}(\mathbf{z}'', \phi) = \begin{bmatrix} c_\phi c_\psi - s_\phi c_\theta s_\psi & -s_\phi c_\psi - c_\phi c_\theta s_\psi & s_\theta s_\psi \\ c_\phi s_\psi + s_\phi c_\theta c_\psi & -s_\phi s_\psi + c_\phi c_\theta c_\psi & -s_\theta c_\psi \\ s_\phi s_\theta & c_\phi s_\theta & c_\theta \end{bmatrix}$$

On choisit généralement, les angles d'Euler dans les intervalles suivants :

$$-\pi \leq \psi < \pi \quad ; \quad -\frac{\pi}{2} \leq \theta < \frac{\pi}{2} \quad ; \quad -\pi \leq \phi < \pi.$$

Si  $\theta = 0$  ou  $\theta = \pi$ , la représentation n'est plus unique et comporte une singularité.

Comme pour les angles nautiques, il existe des formules permettant de retrouver les angles d'Euler à partir d'une matrice de rotation [11].

## 2.4 Orientation déterminée par un vecteur et un angle

L'orientation d'un repère par rapport à un autre peut se décrire par une unique rotation (théorème d'Euler). En effet, une matrice de rotation  $\mathbf{M}$  a toujours une valeur propre égale à un et deux valeurs propres conjuguées  $\cos(\theta) \pm i \sin(\theta)$  où  $\theta$  est l'angle de rotation. Le vecteur propre  $\mathbf{u}$  (unitaire) associé à la valeur propre unitaire est invariant par la rotation puisque par définition,  $\mathbf{u}$  est solution de  $\mathbf{M} \cdot \mathbf{u} = \mathbf{u}$ .

Connaissant une matrice de rotation  $\mathbf{M}$ , l'angle  $\theta$  et le vecteur  $\mathbf{u}$  se calculent facilement à l'aide des formules :

$$\theta = \arccos \left( \frac{\text{Tr}(\mathbf{M}) - 1}{2} \right)$$

$$\mathbf{u} = \frac{1}{2 \sin \theta} \begin{bmatrix} m_{32} - m_{23} \\ m_{13} - m_{31} \\ m_{21} - m_{12} \end{bmatrix}$$

L'orientation de  $R_1$  par rapport à  $R_0$  est donc complètement définie par quatre paramètres qui décrivent une rotation unique d'angle  $\theta$  (avec  $0 \leq \theta \leq \pi$ ) autour de l'axe de vecteur directeur  $\mathbf{u}$ .

Inversement, connaissant un vecteur  $\mathbf{u}$  et un angle de rotation  $\theta$ , il est possible de calculer la matrice de rotation en utilisant la formule de Rodrigues [11] :

$$\mathbf{M}(\mathbf{u}, \theta) = \text{rot}(\mathbf{u}, \theta) = \mathbf{I}_3 + \sin(\theta)\mathbf{S}(\mathbf{u}) + (1 - \cos(\theta)) \mathbf{u}\mathbf{u}^T$$

avec  $\mathbf{S}(\mathbf{u})$  la matrice antisymétrique associée au vecteur  $\mathbf{u}$ . Tous les calculs faits, il vient :

$$\mathbf{M} = \begin{bmatrix} u_x^2 + (1 - u_x^2)c_\theta & u_x u_y (1 - c_\theta) - u_z s_\theta & u_x u_z (1 - c_\theta) + u_y s_\theta \\ u_x u_y (1 - c_\theta) + u_z s_\theta & u_y^2 + (1 - u_y^2)c_\theta & u_y u_z (1 - c_\theta) - u_x s_\theta \\ u_x u_z (1 - c_\theta) + u_y s_\theta & u_y u_z (1 - c_\theta) + u_x s_\theta & u_z^2 + (1 - u_z^2)c_\theta \end{bmatrix}$$

**NB:** Une rotation peut être représentée par le vecteur  $\theta\mathbf{u}$  et donc seulement 3 paramètres. Bien que cette représentation soit minimale, elle est à éviter car elle dégénère au voisinage de  $\theta = 0$ .

## 2.5 Orientation déterminée par un quaternion

Les quaternions sont également appelés paramètres d'Euler ou paramètres de Rodrigues. On trouve cette représentation sur certains robots industriels pour déterminer l'orientation de l'organe terminal. Par rapport aux matrices de rotation, les calculs utilisant les quaternions sont plus efficaces (en terme de complexité) et plus stables numériquement. Ils sont notamment utilisés en synthèse d'images 3D.

Dans cette représentation, l'orientation est également exprimée de manière redondante par quatre scalaires  $a$ ,  $b$ ,  $c$  et  $d$  qui décrivent une rotation unique d'angle  $\theta$  ( $0 \leq \theta \leq \pi$ ) autour d'un axe de vecteur unitaire  $\mathbf{u}$ .

Les paramètres  $a$ ,  $b$ ,  $c$  et  $d$  sont définis comme suit :

$$\begin{cases} a = \cos(\theta/2) \\ b = u_x \sin(\theta/2) \\ c = u_y \sin(\theta/2) \\ d = u_z \sin(\theta/2) \end{cases}$$

Les paramètres  $a$ ,  $b$ ,  $c$  et  $d$  définissent un quaternion unitaire  $z = a + bi + cj + dk$  où  $i$ ,  $j$  et  $k$  dont des nombres imaginaires vérifiant  $i^2 = j^2 = k^2 = ijk = -1$ .

L'avantage des quaternions est que la composition de deux rotations se conforme aux règles usuelles de l'algèbre et est particulièrement rapide à calculer. En effet, étant donné deux rotations décrites par les paramètres  $a_1, b_1, c_1, d_1$  et  $a_2, b_2, c_2, d_2$ , les paramètres de la rotation composée s'écrivent :

$$\begin{aligned} a &= a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2 \\ b &= a_1 b_2 + b_1 a_2 - c_1 d_2 + d_1 c_2 \\ c &= a_1 c_2 + c_1 a_2 - d_1 b_2 + b_1 d_2 \\ d &= a_1 d_2 + d_1 a_2 - b_1 c_2 + c_1 b_2 \end{aligned}$$

En utilisant la formule de Rodrigues, on peut calculer la matrice de rotation équivalente à un quaternion :

$$\mathbf{M} = \begin{bmatrix} 2(a^2 + b^2) - 1 & 2(bc - ad) & 2(bd + ac) \\ 2(bc + ad) & 2(a^2 + c^2) - 1 & 2(cd - ab) \\ 2(bd - ac) & 2(cd + ab) & 2(a^2 + d^2) - 1 \end{bmatrix}$$

Il existe également des formules pour calculer directement un quaternion à partir d'une matrice de rotation sans passer par la représentation angle-vecteur.

## 2.6 Transformations homogènes

Les transformations géométriques en coordonnées homogènes sont beaucoup utilisées en robotique en vision 3D car elles permettent de caractériser les transformations affines dans l'espace, comme les translations et les rotations à l'aide d'une seule matrice.

### 2.6.1 Définition

Soit  $P$  un point du solide. Le but est ici d'exprimer les coordonnées du vecteur  $\overrightarrow{O_0P}$  dans  $R_0$  en fonction des coordonnées de  $\overrightarrow{O_1P}$  dans  $R_1$ , c'est-à-dire d'exprimer  $\overrightarrow{O_0P}/_{R_0}$  en fonction de  $\overrightarrow{O_1P}/_{R_1}$ .

La relation de Chasles permet d'écrire :

$$\overrightarrow{O_0P} = \overrightarrow{O_0O_1} + \overrightarrow{O_1P}$$

Notons que  $\overrightarrow{O_0O_1}$  est un vecteur qui peut s'exprimer aisément dans  $R_0$ . De même,  $\overrightarrow{O_1P}$  est un vecteur qui peut s'exprimer aisément dans  $R_1$ .

La relation précédente, exprimée dans  $R_0$  s'écrit :

$${}^0P = \overrightarrow{O_0P}/_{R_0} = \overrightarrow{O_0O_1}/_{R_0} + \overrightarrow{O_1P}/_{R_0}$$

Les coordonnées du vecteur  $\overrightarrow{O_1P}$  doivent être exprimées dans le repère  $R_0$ , calcul qui dépend de l'orientation du repère  $R_1$  par rapport au repère  $R_0$ . On fait alors appel aux matrices de rotations. En désignant  ${}^0\mathbf{M}_1$  la matrice de passage de  $R_0$  à  $R_1$ , on peut écrire :

$$\overrightarrow{O_1P}/_{R_0} = {}^0\mathbf{M}_1 \cdot \overrightarrow{O_1P}/_{R_1}$$

Ainsi, il vient la relation utile :

$$\boxed{\overrightarrow{O_0P}/_{R_0} = \overrightarrow{O_0O_1}/_{R_0} + {}^0\mathbf{M}_1 \cdot \overrightarrow{O_1P}/_{R_1}}$$

Écrivons cette expression sous forme matricielle :

$$\begin{bmatrix} \overrightarrow{O_0P}/_{R_0} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{M}_1 & \overrightarrow{O_0O_1}/_{R_0} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \overrightarrow{O_1P}/_{R_1} \\ 1 \end{bmatrix}$$

On pose :

$$\boxed{{}^0\mathbf{T}_1 = \begin{bmatrix} {}^0\mathbf{M}_1 & \overrightarrow{O_0O_1}/_{R_0} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}}$$

La matrice  ${}^0\mathbf{T}_1$  est appelée *matrice de transformation homogène*. C'est une matrice de dimension 4 qui permet de représenter des applications affines en dimension 3.

La matrice de transformation homogène  ${}^0\mathbf{T}_1$  définit le repère  $R_1$  dans le repère  $R_0$ . Autrement dit c'est une matrice de passage de  $R_0$  à  $R_1$  qui permet d'exprimer dans le repère  $R_0$  les coordonnées d'un point  $P$  à partir de ses coordonnées données dans le repère  $R_1$  :

$$\boxed{\begin{bmatrix} \overrightarrow{O_0P}/_{R_0} \\ 1 \end{bmatrix} = {}^0\mathbf{T}_1 \cdot \begin{bmatrix} \overrightarrow{O_1P}/_{R_1} \\ 1 \end{bmatrix}}$$

qu'on peut également écrire :

$$\boxed{{}^0\hat{P} = {}^0\mathbf{T}_1 \cdot {}^1\hat{P}}$$

Dans ce cas, le vecteur  $\hat{P}$  représente les *coordonnées homogènes* (de dimension 4) du point  $P$ .

### 2.6.2 Propriétés

On a, par définition :

$${}^1\mathbf{T}_0 = \begin{bmatrix} {}^1\mathbf{M}_0 & \overrightarrow{O_1O_{0/R_1}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{M}_0 & -\overrightarrow{O_0O_{1/R_1}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{M}_0 & -{}^1\mathbf{M}_0 \cdot \overrightarrow{O_0O_{1/R_0}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

Soit :

$$\boxed{{}^1\mathbf{T}_0 = \begin{bmatrix} ({}^0\mathbf{M}_1)^T & -({}^0\mathbf{M}_1)^T \cdot \overrightarrow{O_0O_{1/R_0}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}}$$

On montre alors aisément que  ${}^1\mathbf{T}_0 \cdot {}^0\mathbf{T}_1 = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_0 = \mathbf{I}_4$  et donc que :

$$\boxed{{}^1\mathbf{T}_0 = ({}^0\mathbf{T}_1)^{-1} \neq ({}^0\mathbf{T}_1)^T}$$

et :

$$\det {}^0\mathbf{T}_1 = \det {}^0\mathbf{M}_1 = 1$$

Comme pour les matrices de rotation, on a naturellement aussi :

$$\boxed{{}^0\mathbf{T}_2 = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_2}$$

## 2.7 Conclusion

En résumé, la pose d'un solide dans l'espace peut être représenté sous différentes formes :

- la position d'un point et trois angles,
- la position d'un point, un angle et un axe de rotation,
- la position d'un point et un quaternion,
- la position d'un point et une matrice de rotation,
- une matrice de transformation homogène.

En robotique, les trois premières représentations sont utilisées pour décrire les tâches qu'un robot doit effectuer, notamment dans les langages de programmation et les interfaces homme-machine des différents constructeurs. Les deux dernières représentations sont utilisées pour calculer les différents modèles (géométrique, cinématique et dynamique) d'un robot et pour implanter les lois de commandes.



## Chapitre 3

# Modèle géométrique direct

L'étude d'un robot manipulateur commence par décrire la pose de son effecteur en fonction de la configuration de ses articulations. Ce chapitre présente la modélisation géométrique des robots séries.

### 3.1 Définitions et hypothèses

La pose de l'effecteur dans l'espace est totalement décrit par l'évolution d'au minimum 6 paramètres (3 pour la position et 3 pour l'orientation).

**Définition 7** (Coordonnée opérationnelle). *Une coordonnée opérationnelle est une variable représentant un degré de liberté (ddl) de l'organe terminal du robot dans l'espace (en position ou en orientation). Il y a donc six coordonnées opérationnelles, souvent regroupées en un vecteur colonne ( $6 \times 1$ ) noté  $\mathbf{x}$ . L'ensemble des vecteurs des coordonnées opérationnelles est appelé espace opérationnel ou plus simplement espace cartésien.*

Le mouvement de l'effecteur est engendré par l'actionnement des articulations actives du robot.

**Définition 8** (Coordonnée articulaire). *Une coordonnée articulaire (appelée aussi coordonnée généralisée) est une variable représentant un degré de mobilité d'un robot, en rotation ou en translation. Les coordonnées articulaires sont regroupées en un vecteur colonne ( $n \times 1$ ) noté  $\mathbf{q}$ . Le vecteur  $\mathbf{q}$  définit la configuration (appelée aussi la posture) du robot. L'ensemble des vecteurs des coordonnées articulaires est appelé espace articulaire ou espace des configurations.*

Les coordonnées opérationnelles et articulaires sont reliées par la cinématique du robot ; la tâche que le robot doit effectuer se définit dans l'espace opérationnel, qui correspond à l'espace de la tâche.

**Définition 9** (modèle géométrique direct). *On appelle modèle géométrique direct (MGD) d'un robot, l'application  $\mathbf{f}$  de  $\mathbb{R}^n$  dans  $\mathbb{R}^6$  (où  $n$  est le nombre d'axes du robot) exprimant les coordonnées opérationnelles  $\mathbf{x}$  en fonction des coordonnées articulaires  $\mathbf{q}$ .*

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} \text{position de l'effecteur dans } R_0 \\ \text{orientation de l'effecteur par rapport à } R_0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{bmatrix} \quad (\text{ici avec les angles nautiques})$$

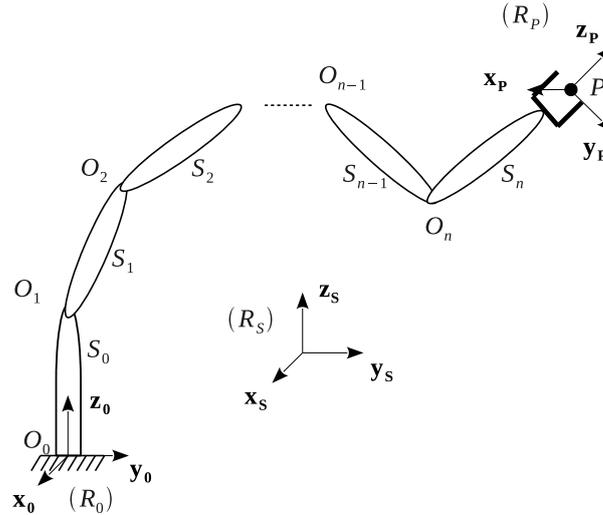


FIGURE 3.1 – Conventions de repérage.

Le modèle géométrique direct peut également être décrit par une matrice de transformation homogène.

Le MGD permet donc de connaître la pose de l'effecteur dans l'espace de la tâche, en fonction des positions des articulations du robot. Il permet notamment de déterminer l'espace de travail du robot (domaine atteignable) en prenant en compte les butées articulaires. L'espace de travail est évidemment un critère majeur du choix d'un robot.

Les structures de robot étudiées ici sont à chaîne cinématique ouverte (robot série), le dernier corps supportant l'effecteur. Les segments du robot sont supposés parfaitement rigides et sont connectés entre eux par des articulations supposées idéales (pas de jeu mécanique ni d'élasticité).

Dans la suite  $n$  désigne le nombre total d'axes du robot. On choisira les repères associés aux segments du robot étudié de la manière suivante :

- à chaque segment  $S_i$  du robot est associé un repère  $R_i$ , du repère  $R_0$  associé à la base, jusqu'au repère  $R_n$  associé au dernier segment,
- un repère outil  $R_P$  est associé à l'organe terminal avec le vecteur  $\mathbf{z}_P$  dans l'axe de l'effecteur et sortant du robot (dirigé vers l'extérieur),
- les coordonnées opérationnelles de l'effecteur s'exprime en un point privilégié  $P$  de cet effecteur ( $P$  est donc l'origine du repère outil  $R_P$ ).
- le repère monde  $R_S$  désigne un repère de référence (distinct de  $R_0$ ), la transformation de  $R_S$  à  $R_0$  est appelé *placement du robot*,
- dans la configuration dite de *référence* toutes les coordonnées articulaires sont nulles.

## 3.2 Modélisation d'une articulation

Soient deux solides  $S_{i-1}$  et  $S_i$ , liés par une articulation à une mobilité, de type pivot ou glissière.

### 3.2.1 Articulation pivot

La rotation de  $S_i$  par rapport à  $S_{i-1}$  est défini par l'angle  $q_i$  autour de l'axe de rotation de vecteur unitaire  $\mathbf{u}_i$ , avec  $q_i = 0$  dans la configuration de référence.

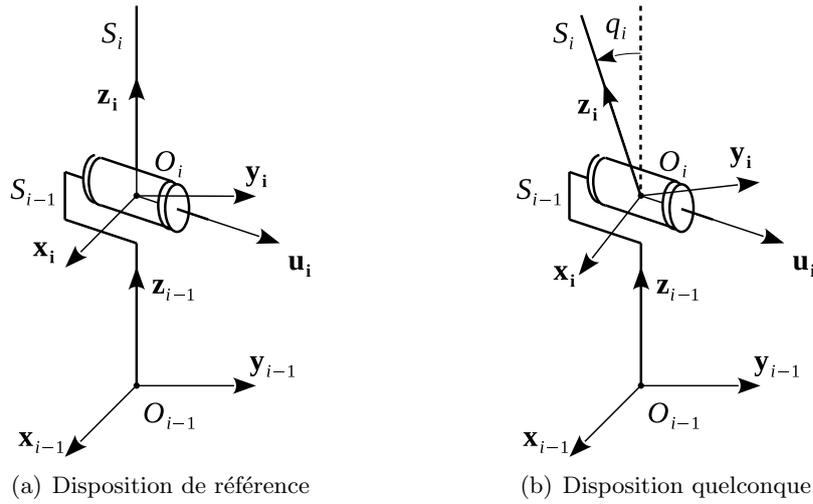


FIGURE 3.2 – Convention de repérage d'une articulation pivot.

A chaque solide est lié un repère ( $R_i$  pour  $S_i$  et  $R_{i-1}$  pour  $S_{i-1}$ ). On choisit ces repères de manière à ce que les transformations soient les plus simples possible, par exemple en faisant coïncider l'un des axes de  $R_{i-1}$  avec l'axe de rotation  $\mathbf{u}_i$ . Enfin, on choisit l'origine  $O_i$  de  $R_i$  sur l'axe de rotation.

Ainsi, la matrice de passage de  $R_{i-1}$  à  $R_i$  s'écrit :

$${}^{i-1}\mathbf{M}_i = \text{rot}({}^{i-1}\mathbf{u}_i, q_i)$$

Soit  $P$  un point de  $S_i$ . On exprime alors les coordonnées du vecteur  $\overrightarrow{O_{i-1}P}$  dans  $R_{i-1}$  :

$$\overrightarrow{O_{i-1}P}/R_{i-1} = \overrightarrow{O_{i-1}O_i}/R_{i-1} + {}^{i-1}\mathbf{M}_i \cdot \overrightarrow{O_iP}/R_i$$

D'où la matrice de transformation homogène :

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} {}^{i-1}\mathbf{M}_i & \overrightarrow{O_{i-1}O_i}/R_{i-1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

Notons que  $\overrightarrow{O_{i-1}O_i}$  est un vecteur de longueur constante qui peut s'exprimer aisément dans  $R_{i-1}$ . Par exemple, si on choisit  $O_i$  sur l'axe  $\mathbf{z}_{i-1}$ , et que  $l_{i-}$  désigne la distance  $O_{i-1}O_i$ , on peut alors écrire :

$$\overrightarrow{O_{i-1}O_i}/R_{i-1} = \begin{bmatrix} 0 \\ 0 \\ l_{i-} \end{bmatrix}$$

### 3.2.2 Articulation glissière

Dans ce cas, le mouvement relatif des deux solides est une translation d'axe représenté par son vecteur unitaire  $\mathbf{u}_i$ . La variable  $q_i$  représente cette fois-ci non plus un angle, mais une longueur correspondant à la translation effectuée.

Comme dans le cas précédent, on choisit l'origine  $O_i$  de  $R_i$  sur l'axe de translation. La matrice de transformation homogène s'écrit aussi :

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} {}^{i-1}\mathbf{M}_i & \overrightarrow{O_{i-1}O_i}/R_{i-1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

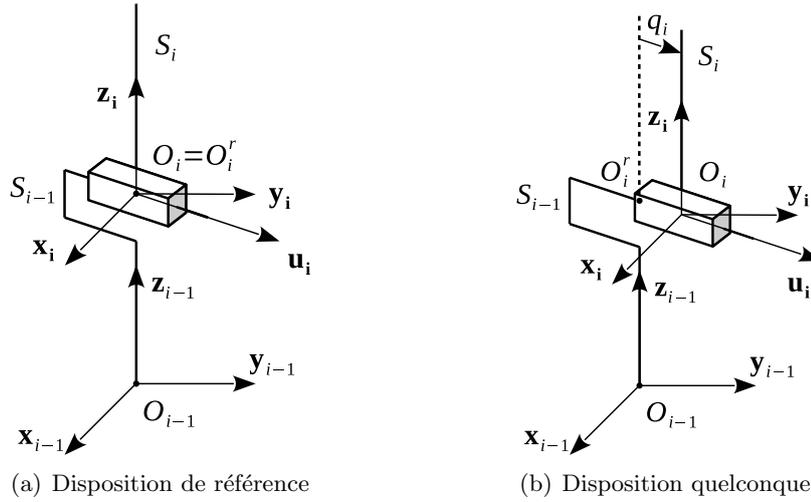


FIGURE 3.3 – Convention de repérage d'une articulation glissière.

Contrairement à une liaison pivot, la matrice de rotation  ${}^{i-1}\mathbf{M}_i$  est ici constante. Si de plus on choisit  $R_{i-1}$  et  $R_i$  équipollents, la matrice est l'identité ( ${}^{i-1}\mathbf{M}_i = \mathbf{I}_3$ ).

La distance  $O_{i-1}O_i$  dépend de  $q_i$ . On définit  $O_i^r$  l'origine  $O_i$  du repère  $R_i$  dans la disposition de référence. On peut alors écrire :

$$\overrightarrow{O_{i-1}O_i/R_{i-1}} = \overrightarrow{O_{i-1}O_i^r} + \overrightarrow{O_i^rO_i} = \overrightarrow{O_{i-1}O_i^r} + q_i \cdot {}^{i-1}\mathbf{u}_i$$

Au final, on obtient :

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} \mathbf{I}_3 & \overrightarrow{O_{i-1}O_i/R_{i-1}} + q_i \mathbf{u}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

### 3.3 Pose de l'organe terminal

#### 3.3.1 Orientation de l'organe terminal

L'orientation de l'effecteur par rapport à la base, c'est à dire de  $R_P$  par rapport à  $R_0$  peut s'exprimer de manière implicite par la matrice de passage  ${}^0\mathbf{M}_P$  de  $R_0$  à  $R_P$ .

D'après les paragraphes précédents, on peut aisément exprimer  ${}^{i-1}\mathbf{M}_i$  correspondant à un axe de rotation ou à un axe de translation. On en déduit donc aisément que :

$${}^0\mathbf{M}_n = \prod_{i=1}^n {}^{i-1}\mathbf{M}_i$$

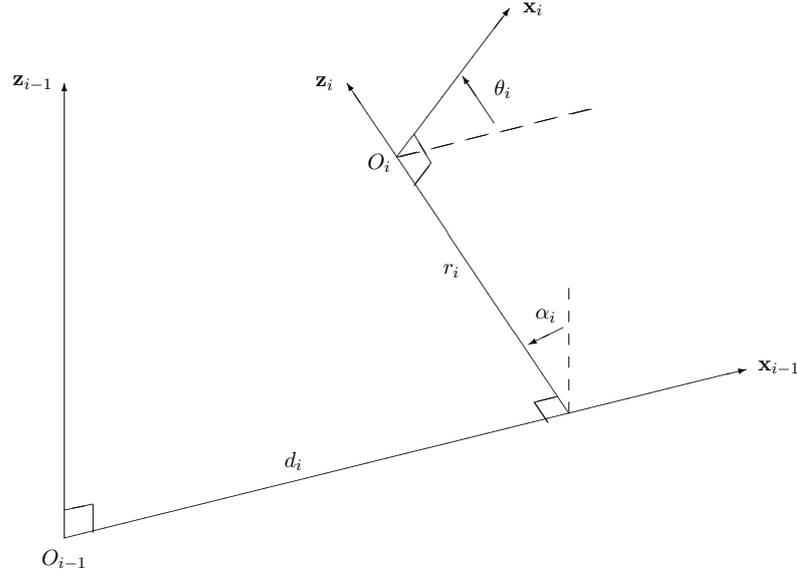
et :

$${}^0\mathbf{M}_P = {}^0\mathbf{M}_n \cdot {}^n\mathbf{M}_P$$

Si on souhaite connaître les coordonnées d'un vecteur particulier, par exemple l'orientation de l'organe terminal (axe  $\mathbf{z}$  du repère outil), on écrit :

$${}^0\mathbf{z}_P = {}^0\mathbf{M}_P \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Si on souhaite exprimer l'orientation du repère outil à travers trois angles (angles nautiques, d'Euler, etc.), on utilise les formules de transformations inverses (cf. chapitre 2).

FIGURE 3.4 – Convention de Denavit-Hartenberg (sur ce schéma  $\alpha_i < 0$  et  $\theta_i < 0$ ).

### 3.3.2 Position de l'organe terminal

Pour calculer les coordonnées du point  $P$  de l'organe terminal dans le repère  $R_0$ , soit  $\overrightarrow{O_0P}/R_0$ , on utilise la relation de Chasles :

$$\overrightarrow{O_0P} = \overrightarrow{O_0O_1} + \overrightarrow{O_1O_2} + \overrightarrow{O_2O_3} + \dots + \overrightarrow{O_{n-1}O_n} + \overrightarrow{O_nP}$$

Quel que soit l'axe  $i$  du robot, on peut aisément exprimer les coordonnées des vecteurs  $\overrightarrow{O_{i-1}O_i}$  dans les repères  $R_{i-1}$ . S'il y a des articulations glissières, on décompose  $\overrightarrow{O_{i-1}O_i}$  en  $\overrightarrow{O_{i-1}O_i^r} + \overrightarrow{O_i^rO_i}$ .

En utilisant, les matrices de passage  ${}^0\mathbf{M}_i$ , on a :

$$\overrightarrow{O_0P}/R_0 = \overrightarrow{O_0O_1}/R_0 + {}^0\mathbf{M}_1 \overrightarrow{O_1O_2}/R_1 + \dots + {}^0\mathbf{M}_n \overrightarrow{O_nP}/R_n$$

### 3.3.3 Pose de l'organe terminal

En multipliant les matrices de transformations homogènes, on a

$${}^0\mathbf{T}_n = \prod_{i=1}^n {}^{i-1}\mathbf{T}_i$$

et :

$${}^0\mathbf{T}_P = {}^0\mathbf{T}_n \cdot {}^n\mathbf{T}_P$$

Cette matrice décrit à la fois l'orientation et la position de l'organe terminal car par définition :

$${}^0\mathbf{T}_P = \begin{bmatrix} {}^0\mathbf{M}_P & \overrightarrow{O_0P}/R_0 \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

## 3.4 Convention de Denavit-Hartenberg

En choisissant les repères  $R_i$  d'une manière bien précise, on peut exprimer les relations aux différents axes en utilisant les paramètres de Denavit-Hartenberg<sup>1</sup>. Cette convention fournit

1. On utilise dans ce cours la convention de Denavit-Hartenberg modifiée

pour chaque axe la matrice de transformation homogène correspondante par la connaissance de quatre paramètres. Cette méthode est intéressante pour systématiser les calculs du modèle géométrique.

Le choix des repères est le suivant :

- l'axe  $\mathbf{z}_i$  correspond à l'axe de l'articulation  $i$ ,
- l'axe  $\mathbf{x}_i$  est porté par la perpendiculaire commune aux axes  $\mathbf{z}_i$  et  $\mathbf{z}_{i+1}$ ,
- si les axes des articulations  $i$  et  $i + 1$  sont concourants, on choisit  $O_i$  et  $O_{i+1}$  au point d'intersection.

Le passage de  $R_{i-1}$  à  $R_i$  s'exprime en fonction des quatre paramètres de Denavit-Hartenberg modifié suivants :

- $\alpha_i$  : angle entre les axes  $\mathbf{z}_{i-1}$  et  $\mathbf{z}_i$ , correspondant à une rotation autour de  $\mathbf{x}_{i-1}$ ,
- $d_i$  : distance entre  $\mathbf{z}_{i-1}$  et  $\mathbf{z}_i$  le long de  $\mathbf{x}_{i-1}$ ,
- $\theta_i$  : angle entre les axes  $\mathbf{x}_{i-1}$  et  $\mathbf{x}_i$ , correspondant à une rotation autour de  $\mathbf{z}_i$ ,
- $r_i$  : distance entre  $\mathbf{x}_{i-1}$  et  $\mathbf{x}_i$  le long de  $\mathbf{z}_i$ .

La variable articulaire  $q_i$  est soit  $\theta_i$ , soit  $r_i$ , selon que l'articulation  $i$  est un pivot ou une glissière.

En utilisant ces paramètres, la matrice de transformation homogène  ${}^{i-1}\mathbf{T}_i$  s'écrit :

$${}^{i-1}\mathbf{T}_i = \text{rot}(\mathbf{z}_i, \theta_i) \cdot \text{trans}(\mathbf{z}_i, r_i) \cdot \text{tans}(\mathbf{x}_{i-1}, d_i) \cdot \text{rot}(\mathbf{x}_{i-1}, \alpha_i)$$

soit :

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

La méthode consiste tout d'abord à placer les repères sur le robot puis à remplir un tableau avec ses paramètres géométriques.

Articulation	$\sigma_i$	$\alpha_i$	$d_j$	$\theta_i$	$r_i$
1					
2					
...					
$n$					

(avec  $\sigma_i = 0$  si l'articulation est un pivot,  
 $\sigma_i = 1$  si l'articulation est une glissière)

A partir du tableau, il est alors aisé de calculer  ${}^0\mathbf{T}_1, {}^1\mathbf{T}_2, \dots, {}^{n-1}\mathbf{T}_n$  puis  ${}^0\mathbf{T}_n$  sachant que :

$${}^0\mathbf{T}_n = \prod_{i=1}^n {}^{i-1}\mathbf{T}_i$$

### 3.5 Domaine atteignable

**Définition 10.** Soit  $\mathbf{q}$  les coordonnées articulaires d'un robot à  $n$  axes et tel que chaque  $q_i$  ait un débattement limité par les butées articulaires :

$$q_i \in [q_{i_{min}}, q_{i_{max}}] \Rightarrow \mathbf{q} \in \mathbb{Q} \subset \mathbb{R}^n \quad (\mathbb{Q} \text{ est un pavé de } \mathbb{R}^n)$$

$\mathbb{Q}$  est nommé espace des configurations du robot.

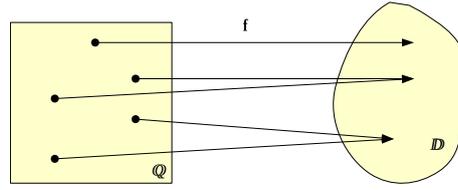


FIGURE 3.5 – Illustration du lien entre espace des configurations et domaine atteignable.

L'un des critères fondamentaux pour effectuer le choix d'un robot en fonction de la tâche à réaliser est naturellement les mouvements possibles de l'effecteur dans l'espace opérationnel. Cette donnée s'exprime par l'intermédiaire de l'espace de travail ou domaine atteignable  $\mathbb{D} = \mathbf{f}(\mathbb{Q})$  comme l'illustre la figure 3.5 (notons qu'il peut y avoir éventuellement des contraintes d'auto-collision du porteur qui réduisent le domaine  $\mathbb{D}$ ).

**Définition 11.** On appelle domaine atteignable d'un robot, la variété  $\mathbb{D}$  de  $\mathbb{R}^6$  définie par :

$$\mathbb{D} = \{\mathbf{f}(\mathbf{q}) \mid \mathbf{q} \in \mathbb{Q}\}$$

La dimension  $d$  de  $\mathbb{D}$  est « presque partout » (en excluant les configurations singulières) égale au nombre de degrés de liberté du robot :  $d = \dim(\mathbb{D}) \leq \min(n, 6)$  (le cas  $n > d$  correspond à une structure redondante de robot).

Il est difficile de se représenter complètement  $\mathbb{D}$ , qui est a priori de dimension 6. Pour le représenter, on le scinde en deux :

- $\mathbb{D}_P = \mathbf{f}_P(\mathbb{Q})$  : domaine des positions atteignables, chacune pour au moins une orientation de l'effecteur,
- $\mathbb{D}_O = \mathbf{f}_O(\mathbb{Q})$  : domaine des orientations atteignables, chacune pour au moins une position de  $P$ .

Le cas échéant, si la tâche implique que les trajectoires de l'effecteur doivent s'effectuer sur un plan, on peut être amené à déterminer le domaine atteignable pour une orientation constante de l'effecteur, correspondant à celle du plan. On peut ainsi, selon la tâche à réaliser, être amené à s'intéresser à :

- $\mathbb{D}_P^*(\alpha, \beta, \gamma)$  : domaine des positions atteignables, chacune **pour une orientation donnée de l'effecteur**,
- $\mathbb{D}_O^*(x, y, z)$  : domaine des orientations atteignables, chacune **pour une position donnée de  $P$** .

Ces domaines atteignables sont plus restrictifs que les précédents :

$$\mathbb{D}_O^* \subset \mathbb{D}_O \quad \text{et} \quad \mathbb{D}_P^* \subset \mathbb{D}_P$$

### 3.6 Conclusion

On est donc à présent capable d'exprimer le modèle géométrique d'un robot. Quelle que soit la méthode, on peut exprimer ce modèle à l'aide d'une unique matrice de transformation homogène  ${}^0\mathbf{T}_P$ .



## Chapitre 4

# Modèle géométrique inverse

La commande d'un robot requiert, à partir des trajectoires désirées de l'effecteur dans l'espace opérationnel, de connaître les coordonnées articulaires correspondantes. Ceci peut donc être réalisée à partir du modèle géométrique inverse.

### 4.1 Définition

**Définition 12.** On appelle modèle géométrique inverse (MGI), l'application de  $\mathbb{R}^6$  dans  $\mathbb{R}^n$  exprimant l'ensemble des  $\mathbf{q}$  en fonction de  $\mathbf{x}$  telle que :

$$\{\mathbf{q} \mid \mathbf{f}(\mathbf{q}) = \mathbf{x}\}$$

avec  $\mathbf{f}(\mathbf{q})$  le modèle géométrique direct.

Le MGI est destiné à réaliser un *transformateur de coordonnées*, à partir des coordonnées opérationnelles souhaitées vers les consignes articulaires correspondantes. Le MGI est donc la forme explicite qui donne toutes les solutions possibles pour  $\mathbf{q}$ , correspondantes à un  $\mathbf{x}$  fixé. Un robot est dit *résoluble* lorsqu'il est possible de calculer toutes les configurations des axes (tous les différents  $\mathbf{q}$  possibles) permettant d'atteindre une pose opérationnelle  $\mathbf{x}$  donnée.

Le problème majeur est que dans la majorité des cas (sauf robots cartésiens), le MGI est non linéaire (il contient des fonctions trigonométriques) et qu'il n'est pas toujours possible de l'inverser explicitement. Plusieurs méthodes existent pour obtenir le MGI :

- les méthodes trigonométriques, permettant d'obtenir chaque variable articulaire à partir de relations entre les distances et les angles du robot,
- les méthodes systématiques, comme la méthode de Paul, utilisant le MGD et s'adaptant à la plupart des robots,
- les méthodes spécifiques à des cinématiques particulières, comme les robots à six degrés de liberté à poignet sphérique ou les robots redondants,
- les méthodes numériques type Newton-Raphson lorsqu'il n'est pas possible de trouver une forme explicite du MGI (voir chapitre 7).

Quand on calcule le MGI d'un robot, trois cas peuvent se présenter :

- il n'y a aucune solution permettant de satisfaire au  $\mathbf{x}$  souhaité, car celui-ci est en dehors du domaine atteignable du robot,
- il y a une infinité de solutions : si le robot est redondant vis-à-vis de la tâche à réaliser, ou si le robot se trouve en configuration singulière (voir chapitre suivant),
- il y a un nombre fini de solutions. Ce nombre dépend de la structure cinématique du robot. Par exemple pour les robots anthropomorphe à poignet sphérique, il y a au maximum 8 solutions du MGI pour une pose donnée illustrées par la figure 4.1.

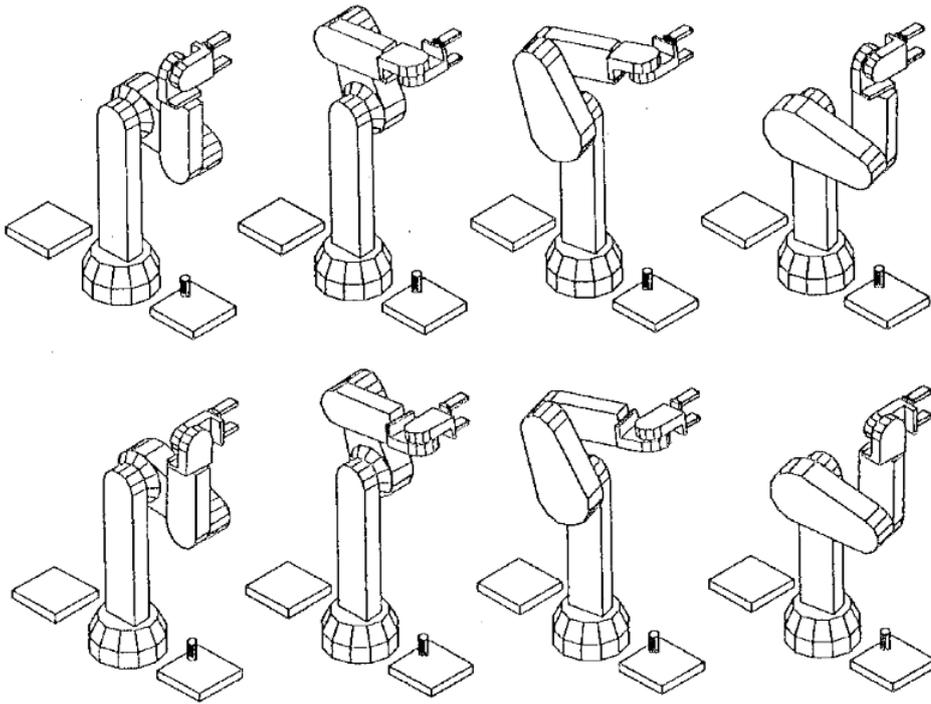


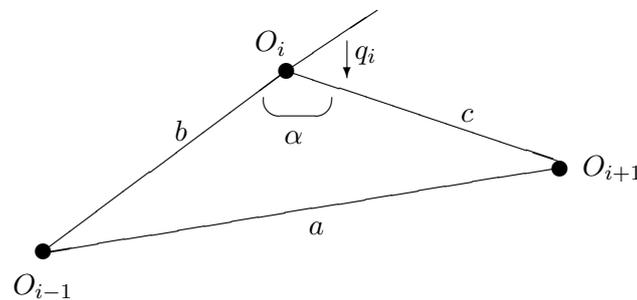
FIGURE 4.1 – Exemple de solutions du MGI pour un robot 6R.

## 4.2 Méthode trigonométrique

Cette méthode est intéressante pour des robots à structure simple et ne nécessite pas forcément le calcul préalable du modèle géométrique direct. Elle consiste à utiliser les relations trigonométriques d'un triangle quelconque. On essaie alors de ramener le problème à la résolution de triangles, en partant souvent de l'effecteur, pour remonter jusqu'à la base du robot.

Prenons par exemple une articulation pivot d'un robot, comme indiquée sur le schéma suivant. En supposant connues les longueurs  $a$ ,  $b$  et  $c$ , on peut calculer l'angle  $\alpha$  par la loi des cosinus (théorème d'Al-Kashi) :

$$a^2 = b^2 + c^2 - 2bc \cos \alpha$$



Ceci permet alors de calculer  $q_i$ . Il peut naturellement exister plusieurs solutions qui seront exprimées à l'aide des fonctions trigonométriques réciproques. On rappelle que :

$$\begin{aligned} \text{acos} : [-1; 1] &\rightarrow [0; \pi] \\ x &\mapsto \text{acos}(x) \end{aligned}$$

et :

$$\begin{aligned} \text{asin} : [-1; 1] &\rightarrow [-\pi/2; \pi/2] \\ x &\mapsto \text{asin}(x) \end{aligned}$$

On utilise également la fonction *arctangente* à deux arguments :

$$\begin{aligned} \text{atan2} : \quad \mathbb{R}^2 &\rightarrow ]-\pi; \pi] \\ (y, x) &\mapsto \text{atan2}(y, x) \end{aligned}$$

### 4.3 Méthode de Paul

Cette méthode a été proposée par Paul en 1981 [11].

Soit un robot à  $n$  degrés de liberté dont la structure est représentée par la matrice de transformation homogène  ${}^0\mathbf{T}_P = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_2 \dots {}^n\mathbf{T}_P$ . Soit  $\mathbf{U}$  la matrice de transformation décrivant l'orientation et la position désirées. On souhaite résoudre l'équation générale suivante dont les  $q_i$  sont solutions :

$${}^0\mathbf{T}_P = \mathbf{U}$$

Pour calculer les  $q_i$ , on procède en plusieurs étapes. On écrit d'abord :

$${}^1\mathbf{T}_2 \dots {}^n\mathbf{T}_P = {}^1\mathbf{T}_0 \cdot \mathbf{U}$$

Le terme de droite n'est fonction que de  $q_1$  et le terme de gauche est fonction de  $q_2 \dots q_n$ . On peut alors obtenir  $q_1$  par identification d'un ou deux coefficients judicieusement choisis parmi les 12 disponibles.

Une fois  $q_1$  calculé, on écrit :

$${}^2\mathbf{T}_3 \dots {}^n\mathbf{T}_P = {}^2\mathbf{T}_1 \cdot {}^1\mathbf{T}_0 \cdot \mathbf{U}$$

Cette fois,  $q_1$  étant calculé, le terme de droite ne dépend que de  $q_2$  et celui de gauche de  $q_3 \dots q_n$ . De même que précédemment, on choisit d'identifier le ou les coefficients des deux matrices permettant le calcul de  $q_2$ .

On réitère ainsi le processus, jusqu'à obtenir la dernière coordonnée articulaire  $q_n$  en écrivant :

$${}^n\mathbf{T}_P = {}^n\mathbf{T}_{n-1} \dots {}^2\mathbf{T}_1 \cdot {}^1\mathbf{T}_0 \cdot \mathbf{U}$$

Cette méthode est systématique mais demande de connaître la solution d'équations trigonométriques à une ou deux inconnues. Heureusement pour les robots ayant une cinématique classique, on retombe sur un petit nombre de cas de figures qui sont répertoriés dans les ouvrages de références (voir le tableau 4.1). A la main, elle est souvent plus complexe à mettre en œuvre pour les robots étudiés que la méthode trigonométrique.

### 4.4 Méthode d'inversion pour les robots à 6 ddl à poignet sphérique

On a vu dans le premier chapitre que les poignets à 3 articulations pivots d'axes concourants correspondent quasiment à une liaison rotule. Pour cette raison, un tel poignet est qualifié de *sphérique*. De tels robots sont résolubles et on peut obtenir le MGI de manière explicite.

La figure 4.2 représente un porteur quelconque à 3 axes ( $q_1, q_2, q_3$ ) suivi des 3 articulations du poignet ( $q_4, q_5, q_6$ ). Ces 3 articulations sont représentées par une liaison rotule de centre  $C$ .

Les données du problème sont le vecteur  $\overrightarrow{O_0P}/R_0$  et l'orientation du repère  $R_P$  dans  $R_0$ , représentée par une matrice de passage  $\mathbf{M}(\alpha, \beta, \gamma)$  définie par exemple par les angles nautiques ou par un quaternion.

Type 1	$X r_i = Y$
Type 2	$X S\theta_i + Y C\theta_i = Z$
Type 3	$X_1 S\theta_i + Y_1 C\theta_i = Z_1$ $X_2 S\theta_i + Y_2 C\theta_i = Z_2$
Type 4	$X_1 r_j S\theta_i = Y_1$ $X_2 r_j C\theta_i = Y_2$
Type 5	$X_1 S\theta_i = Y_1 + Z_1 r_j$ $X_2 C\theta_i = Y_2 + Z_2 r_j$
Type 6	$W S\theta_j = X C\theta_i + Y S\theta_i + Z_1$ $W C\theta_j = X S\theta_i - Y C\theta_i + Z_2$
Type 7	$W_1 C\theta_j + W_2 S\theta_j = X C\theta_i + Y S\theta_i + Z_1$ $W_1 S\theta_j - W_2 C\theta_j = X S\theta_i - Y C\theta_i + Z_2$
Type 8	$X C\theta_i + Y C(\theta_i + \theta_j) = Z_1$ $X S\theta_i + Y S(\theta_i + \theta_j) = Z_2$

TABLE 4.1 – Classification des équations trigonométriques usuelles extraite de [3] (se reporter à cet ouvrage pour les solutions).

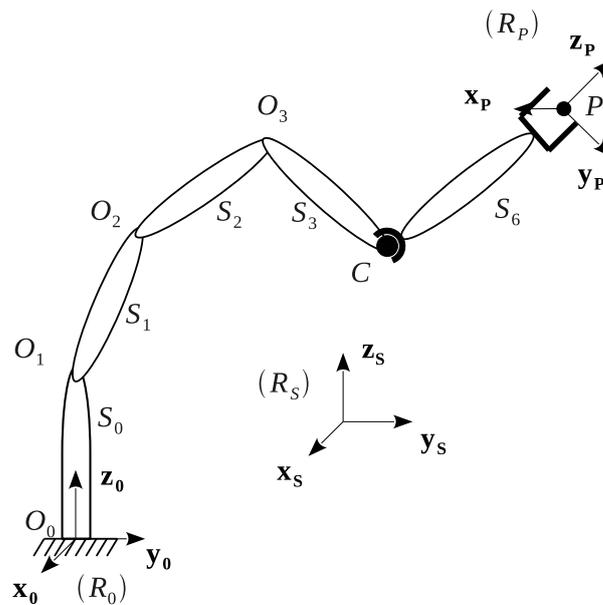


FIGURE 4.2 – Conventions de repérage pour l'inversion des robots à 6 ddl à poignet sphérique.

#### 4.4.1 Calcul de $\overrightarrow{O_0C}/_{R_0}$

La distance  $PC$  est constante et  $C$  est un point fixe de  $R_P$ , donc on peut connaître aisément  $\overrightarrow{PC}/_{R_P}$ . On peut ensuite écrire :

$$\overrightarrow{O_0C}/_{R_0} = \overrightarrow{O_0P}/_{R_0} + \overrightarrow{PC}/_{R_0} = \overrightarrow{O_0P}/_{R_0} + \mathbf{M}(\alpha, \beta, \gamma) \overrightarrow{PC}/_{R_P}$$

$\overrightarrow{O_0P}/_{R_0}$ ,  $\mathbf{M}(\alpha, \beta, \gamma)$  et  $\overrightarrow{PC}/_{R_P}$  sont connus, donc on peut calculer  $\overrightarrow{O_0C}/_{R_0}$ .

#### 4.4.2 Résolution du porteur : calcul de $q_1$ , $q_2$ et $q_3$

Connaissant  $\overrightarrow{O_0C}/_{R_0}$ , on peut résoudre le porteur en prenant soin d'examiner toutes les solutions possibles. Si on ne tient pas compte des butées articulaires, il y a au plus quatre solutions, c'est-à-dire quatre postures du porteur permettant d'atteindre le même point  $C$ .

#### 4.4.3 Calcul de ${}^3\mathbf{M}_P$ (MGD du poignet)

On calcule  ${}^3\mathbf{M}_P$  à partir des trois variables articulaires  $q_4$ ,  $q_5$  et  $q_6$ .

#### 4.4.4 Résolution du poignet : calcul de $q_4$ , $q_5$ et $q_6$

Les trois variables articulaires  $q_1$ ,  $q_2$  et  $q_3$  étant connues, on peut calculer la matrice de passage  ${}^0\mathbf{M}_3 = {}^0\mathbf{M}_1 \cdot {}^1\mathbf{M}_2 \cdot {}^2\mathbf{M}_3$ . Connaissant l'orientation désirée de  $R_P$  définie par  $\mathbf{M}(\alpha, \beta, \gamma)$ , on peut écrire :

$${}^0\mathbf{M}_P = {}^0\mathbf{M}_3 \cdot {}^3\mathbf{M}_P = \mathbf{M}(\alpha, \beta, \gamma)$$

On peut alors isoler  ${}^3\mathbf{M}_P$  :

$${}^3\mathbf{M}_P = {}^3\mathbf{M}_0 \cdot \mathbf{M}(\alpha, \beta, \gamma) = ({}^0\mathbf{M}_3)^T \cdot \mathbf{M}(\alpha, \beta, \gamma)$$

On peut alors en déduire  $q_4$ ,  $q_5$  et  $q_6$  en identifiant les coefficients des matrices (méthode de Paul).

Si  $q_5 = 0$  (ou  $q_5 = \pi$ ), il y a une infinité de solutions. Il s'agit de la singularité du poignet (voir chapitre suivant). Dans les autres cas, il y a deux solutions (sans tenir compte des butées articulaires).

## 4.5 Conclusion

Du fait de la non linéarité du MGD, le MGI peut s'avérer complexe à calculer de manière explicite. Il n'existe pas d'algorithme universel permettant de résoudre toutes les structures de robot. Néanmoins, la quasi-totalité des robots est résoluble. Pour les autres, on peut utiliser une méthode numérique utilisant le modèle cinématique défini au chapitre suivant.

Par rapport aux méthodes numériques, le MGI explicite a l'avantage de donner toutes les solutions possibles. Certaines sont d'emblée éliminées car elles ne satisfont pas les contraintes de butées articulaires. L'utilisateur a alors le choix des solutions restantes.



# Chapitre 5

## Modèles cinématiques direct et inverse

On s'intéresse ici aux relations liant les vitesses opérationnelles (c'est-à-dire la vitesse instantanée du point  $P$  de l'effecteur et la vitesse de rotation du repère  $R_P$  lié à l'organe terminal) en fonction des vitesses articulaires  $\dot{q}_i$ .

### 5.1 Définitions

#### 5.1.1 Modèle cinématique direct

**Définition 13.** On appelle modèle cinématique direct (MCD) d'un robot, l'application donnant les vitesses opérationnelles  $\dot{\mathbf{x}}$  en fonction des positions  $\mathbf{q}$  et des vitesses  $\dot{\mathbf{q}}$  :

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})$$

Au contraire du modèle géométrique, qui correspond à une relation non linéaire entre  $\mathbf{x}$  et  $\mathbf{q}$ , le modèle cinématique correspond à une relation linéaire entre  $\dot{\mathbf{x}}$  et  $\dot{\mathbf{q}}$ . Le modèle cinématique peut s'écrire sous la forme :

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

La matrice  $\mathbf{J}(\mathbf{q})$  est nommée *matrice jacobienne* ou simplement jacobienne du robot. Cette matrice est de dimension  $(d \times n)$ , où  $d$  est le nombre de degrés de liberté du robot ( $d = 6$  au maximum) et  $n$  le nombre d'axes. La matrice jacobienne dépend de  $\mathbf{q}$ . Elle varie donc dans le temps en fonction des mouvements du robot.

Par convention, les matrices jacobiennes présentées dans ce cours seront toujours exprimées dans le repère  $R_0$ .

#### 5.1.2 Modèle cinématique inverse

On appelle *modèle cinématique inverse* (MCI), l'application inverse de la précédente, donnant  $\dot{\mathbf{q}}$  en fonction de  $\dot{\mathbf{x}}$ , soit :

$$\dot{\mathbf{q}} = [\mathbf{J}(\mathbf{q})]^{-1} \cdot \dot{\mathbf{x}}$$

Ce modèle est calculable à condition que  $\mathbf{J}(\mathbf{q})$  soit inversible. En fonction des valeurs de  $\mathbf{q}$ , cette condition peut être toujours vérifiée, ou jamais vérifiée, ou encore non vérifiée sur quelques points seulement, correspondant à des configurations singulières.

## 5.2 Jacobienne analytique

La vitesse instantanée de l'organe terminal représenté par le point  $P$  dans le repère  $R_0$  est définie par la dérivée du vecteur position :

$$\boxed{{}^0\mathbf{v}(P) = \frac{d}{dt}\overrightarrow{O_0P}/R_0}$$

Notons que l'opération de dérivation ne dépend pas du référentiel. C'est bien le vecteur qui s'exprime différemment selon le repère.

En généralisant, la matrice jacobienne peut s'obtenir en dérivant le modèle géométrique représentée par la fonction vectorielle  $\mathbf{f}(\mathbf{q})$  par rapport au temps.

Soit le modèle géométrique direct d'un robot exprimé dans le repère  $R_0$  :

$$\mathbf{x} = \mathbf{f}(\mathbf{q})$$

En dérivant cette équation par rapport au temps, on obtient :

$$\dot{\mathbf{x}} = \mathbf{J}^a(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

avec  $\mathbf{J}^a(\mathbf{q})$  la matrice jacobienne :

$$\mathbf{J}^a(\mathbf{q}) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_d}{\partial q_1} & \dots & \frac{\partial f_d}{\partial q_n} \end{bmatrix}$$

Dans ce cours, cette matrice est nommée matrice jacobienne *analytique* pour la distinguer de matrice jacobienne *cinématique* présentée plus loin.

Les  $f_i$  sont dérivables car formés de fonctions affines et trigonométriques. Le calcul de  $\mathbf{J}^a(\mathbf{q})$  est donc aisée. La jacobienne analytique est généralement calculée en utilisant une représentation explicite de l'orientation comme les angles d'Euler ou les angles nautiques. Par exemple, pour un robot 6 axes, 6 ddl et une orientation définie par les angles nautiques, la jacobienne analytique s'écrit :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_6}{\partial q_1} & \dots & \frac{\partial f_6}{\partial q_6} \end{pmatrix} \cdot \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{pmatrix}$$

Néanmoins, les représentations de l'orientation par trois angles sont mal adaptées pour calculer les efforts articulaires. Pour ce faire, on préférera utiliser la jacobienne cinématique fondée sur les torseurs cinématiques.

## 5.3 Jacobienne cinématique

Le mouvement de l'organe terminal peut également être défini par sa vitesse instantanée et sa vitesse de rotation instantanée sous la forme d'un torseur cinématique. Cette partie propose d'établir la relation entre ce torseur et les positions et vitesses articulaires  $\mathbf{q}$  et  $\dot{\mathbf{q}}$ .

**NB:** L'opération de dérivation ne dépend pas du référentiel. C'est bien le vecteur qui s'exprime différemment selon le repère.

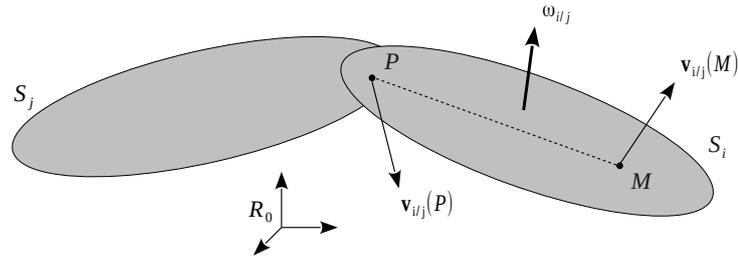


FIGURE 5.1 – Illustration du transport des vitesses (formule de Varignon).

### 5.3.1 Cinématique du solide

#### Torseur cinématique

Le champ de vitesse d'un solide indéformable  $S_i$  en mouvement par rapport à un solide  $S_j$  est complètement défini par un torseur cinématique, écrit en un point  $P$  :

$$\mathcal{V}_{i/j}(P) = \begin{bmatrix} \mathbf{v}_{i/j}(P) \\ \boldsymbol{\omega}_{i/j} \end{bmatrix}$$

où :

- $\mathbf{v}_{i/j}(P)$  est la vitesse instantanée du point  $P$  dans le mouvement de  $S_i$  par rapport à  $S_j$ ,
- $\boldsymbol{\omega}_{i/j}$  est la vitesse de rotation instantanée du solide  $S_i$  par rapport à  $S_j$ .

Notons que le torseur cinématique  $\mathcal{V}_{i/j}(P)$  peut être exprimé dans n'importe quel repère.

#### Formule de Varignon

Soit  $M$  et  $P$  deux points. La formule de Varignon (ou formule transport des vitesses) permet d'écrire la vitesse d'un point  $P$  quelconque en fonction de  $\mathbf{v}_{i/j}(M)$  et de  $\boldsymbol{\omega}_{i/j}$  :

$$\boxed{\mathbf{v}_{i/j}(P) = \mathbf{v}_{i/j}(M) + \overrightarrow{PM} \wedge \boldsymbol{\omega}_{i/j}}$$

On rappelle qu'en dimension trois le produit vectoriel de deux vecteurs  $\mathbf{u}$  et  $\mathbf{v}$  est :

$$\mathbf{u} \wedge \mathbf{v} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix}$$

La formule de Varignon peut également s'écrire sous forme matricielle :

$$\boxed{\mathcal{V}_{i/j}(P) = \begin{bmatrix} \mathbf{I}_3 & [\overrightarrow{PM}]_{\times} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix} \cdot \mathcal{V}_{i/j}(M)}$$

avec  $[\overrightarrow{PM}]_{\times}$  la matrice antisymétrique de pré-produit vectoriel du vecteur  $\overrightarrow{PM}$ . En effet pour deux vecteurs  $\mathbf{u}$  et  $\mathbf{v}$  on peut écrire :

$$\mathbf{u} \wedge \mathbf{v} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = [\mathbf{u}]_{\times} \cdot \mathbf{v}$$

### Composition des vitesses

Soit  $i, j$  et  $k$  trois solides et  $P$  un point, la vitesse  $\mathbf{v}_{i/j}(P)$  de  $P$  dans le mouvement de  $S_i$  par rapport à  $S_j$  peut s'écrire en fonction du mouvement de  $S_i$  par rapport à  $S_k$  et du mouvement de  $S_k$  par rapport à  $S_j$ , soit :

$$\boxed{\mathbf{v}_{i/j}(P) = \mathbf{v}_{i/k}(P) + \mathbf{v}_{k/j}(P)}$$

La loi de composition de vitesse de rotation instantanée s'écrit également :

$$\boxed{\boldsymbol{\omega}_{i/j} = \boldsymbol{\omega}_{i/k} + \boldsymbol{\omega}_{k/j}}$$

Ces deux équations peuvent s'écrire sous la forme de torseurs :

$$\boxed{\mathcal{V}_{i/j}(P) = \mathcal{V}_{i/k}(P) + \mathcal{V}_{k/j}(P)}$$

**Attention, il est impératif que les trois torseurs soient exprimés dans le même repère !**

On remarquera que comme un solide  $i$  est fixe par rapport à lui-même, on a  $\mathcal{V}_{i/i} = 0$  et donc :

$$\mathcal{V}_{i/j}(P) = -\mathcal{V}_{j/i}(P)$$

### Changement de repère

On peut exprimer un torseur  $\mathcal{V}_{i/j}(P)$  dans n'importe quel repère. Pour exprimer un torseur connu dans le repère  $l$  dans un autre repère  $k$ , on écrit :

$$\boxed{{}^k\mathcal{V}_{i/j}(P) = {}^k\mathcal{M}_l \cdot {}^l\mathcal{V}_{i/j}(P)}$$

La matrice  ${}^k\mathcal{M}_l$  est telle que :

$${}^k\mathcal{M}_l = \begin{bmatrix} {}^k\mathbf{M}_l & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^k\mathbf{M}_l \end{bmatrix}$$

où  ${}^k\mathbf{M}_l$  est la matrice de passage entre les repères  $R_k$  et  $R_l$ .

### 5.3.2 Modélisation des articulations

On s'intéresse maintenant aux équations cinématiques liant deux solides  $S_{i-1}$  et  $S_i$  reliés par une articulation. L'axe de l'articulation est noté  $\mathbf{u}_i$ , son centre  $O_i$  et la valeur de l'angle ou de la translation est notée  $q_i$  (cf. figures 3.2 et 3.3).

#### Articulation pivot

Pour une articulation pivot, la vitesse de rotation instantanée de  $S_i$  par rapport à  $S_{i-1}$  est par définition :

$$\boldsymbol{\omega}_{i/i-1} = \mathbf{u}_i \cdot \dot{q}_i$$

De plus, la vitesse du centre de rotation  $O_i$  par rapport à  $S_{i-1}$  est nulle :

$$\mathbf{v}_{i/i-1}(O_i) = 0$$

Donc le torseur cinématique d'une articulation pivot en  $O_i$  exprimé dans le repère  $R_i$  est :

$${}^i\mathcal{V}_{i/i-1}(O_i) = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ {}^i\mathbf{u}_i \cdot \dot{q}_i \end{bmatrix}$$

Transporté en  $P$ , le torseur cinématique d'une articulation pivot est au final :

$${}^i\mathcal{V}_{i/i-1}(P) = \begin{bmatrix} {}^i\mathbf{u}_i \wedge (\overrightarrow{O_i P}_{/R_i}) \\ {}^i\mathbf{u}_i \end{bmatrix} \cdot \dot{q}_i$$

### Articulation glissière

Pour une articulation glissière, la vitesse de rotation instantanée de  $S_i$  par rapport à  $S_{i-1}$  est nulle par définition :

$$\boldsymbol{\omega}_{i/i-1} = 0$$

La vitesse du point  $O_i$  est par définition :

$$\mathbf{v}_{i/i-1}(O_i) = \mathbf{u}_i \cdot \dot{q}_i$$

Donc le torseur cinématique d'une articulation glissière en  $O_i$  exprimé dans le repère  $R_i$  est :

$${}^i\mathcal{V}_{i/i-1}(O_i) = \begin{bmatrix} {}^i\mathbf{u}_i \cdot \dot{q}_i \\ \mathbf{0}_{3 \times 1} \end{bmatrix}$$

Transporté en  $P$ , le torseur cinématique d'une articulation glissière est au final :

$${}^i\mathcal{V}_{i/i-1}(P) = \begin{bmatrix} {}^i\mathbf{u}_i \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \cdot \dot{q}_i$$

### 5.3.3 Calcul de la matrice jacobienne cinématique

Le mouvement de l'organe terminal est complètement défini par le torseur cinématique  ${}^0\mathcal{V}_{n/0}(P)$ . Pour calculer  ${}^0\mathcal{V}_{n/0}(P)$ , on peut utiliser les lois de composition :

$${}^0\mathcal{V}_{n/0}(P) = {}^0\mathcal{V}_{n/n-1}(P) + {}^0\mathcal{V}_{n-1/n-2}(P) + \dots + {}^0\mathcal{V}_{1/0}(P)$$

Or, les torseurs  $\mathcal{V}_{i/i-1}(P)$  relatifs à chaque articulations sont connus dans les repères  $R_i$ . On effectue alors un changement de repère pour de  $R_0$  à  $R_i$  et on obtient :

$${}^0\mathcal{V}_{n/0}(P) = {}^0\mathcal{M}_n \cdot {}^n\mathcal{V}_{n/n-1}(P) + {}^0\mathcal{M}_{n-1} \cdot {}^{n-1}\mathcal{V}_{n-1/n-2}(P) + \dots + {}^0\mathcal{M}_1 \cdot {}^1\mathcal{V}_{1/0}(P)$$

Soit :

$${}^0\mathcal{V}_{n/0}(P) = \sum_{i=1}^n {}^0\mathcal{M}_i \cdot {}^i\mathcal{V}_{i/i-1}(P)$$

Chaque terme de la somme étant linéairement dépendant d'un  $\dot{q}_i$ , il est aisé d'exprimer  ${}^0\mathcal{V}_{n/0}(P)$  sous la forme d'un produit matrice vecteur :

$${}^0\mathcal{V}_{n/0}(P) = \mathbf{J}^c(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

$\mathbf{J}^c(\mathbf{q})$  est la matrice jacobienne dite *cinématique* (ou encore jacobienne de base) pour ne pas la confondre avec la matrice jacobienne analytique.

Les matrices jacobiennes analytique et cinématique sont identiques sur les trois premières lignes et diffèrent sur les trois dernières lignes. Il existe néanmoins des formules pour passer directement d'une forme à l'autre (voir notamment [13]).

## 5.4 Vitesses atteignables et manipulabilité

Les matrices jacobiennes (analytique ou cinématique) permettent de caractériser les vitesses atteignables d'un robot dans une configuration  $\mathbf{q}$  donnée. En effet, si l'on connaît les vitesses articulaires maximales  $\dot{\mathbf{q}}_{\max}$ , on a :

$$-\dot{\mathbf{q}}_{\max} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{\max}$$

et donc :

$$\min \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \leq \dot{\mathbf{x}} \leq \max \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

Pour représenter graphiquement les vitesses atteignables du robot dans une certaine configuration, on peut utiliser les *ellipsoïdes de manipulabilité*. Dans l'espace des vitesses articulaires, on considère l'ensemble des vecteurs  $\dot{\mathbf{q}}$  unitaires, c'est-à-dire l'ensemble des vecteurs vitesses vérifiant :

$$\dot{\mathbf{q}}^T \dot{\mathbf{q}} = 1$$

Cette équation définit une sphère dans l'espace des vitesses articulaires. L'image de cet ensemble dans l'espace opérationnel donne l'équation :

$$\dot{\mathbf{x}}^T \left( \mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T \right)^{-1} \dot{\mathbf{x}} = 1$$

Cette équation définit une ellipsoïde qui est représentative des vitesses potentielles du robot dans chaque direction de l'espace opérationnel. Les axes principaux de l'ellipsoïde correspondent aux vecteurs propres de la matrice  $\mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T$  et les longueurs des demi-axes correspondent aux valeurs propres associées.

Le volume intérieur  $m(\mathbf{q})$  de l'ellipsoïde donne une mesure de la capacité du robot à générer des vitesses homogènes dans l'espace opérationnel. Ce volume est défini par :

$$m(\mathbf{q}) = \sqrt{\det \mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T}$$

Cette mesure  $m(\mathbf{q})$  est appelée *manipulabilité en vitesse* du robot. Elle a été introduite par Yoshikawa en 1984. Néanmoins, elle n'a pas de sens physique quand la matrice jacobienne mélange des vitesses ayant des unités différentes (par exemple des m/s et des rad/s).

## 5.5 Aspects et configurations singulières

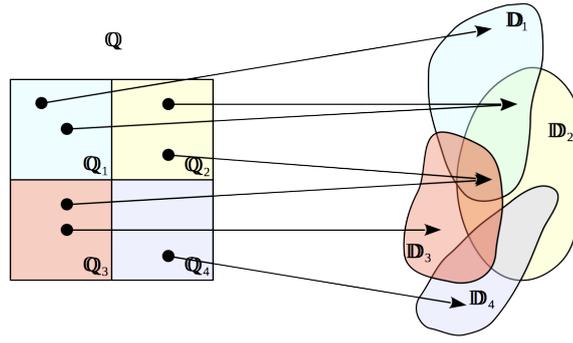
Les matrices jacobiennes permettent également de partitionner l'espace articulaire d'un robot en sous-espaces appelés *aspects* et de détecter ses configurations singulières.

### 5.5.1 Aspects et domaine atteignable

Rappelons les notations relatives au domaine atteignable définies dans le chapitre précédent : en prenant en compte les butées articulaires d'un robot ( $q_i \in [q_{i_{\min}}, q_{i_{\max}}]$ ), on a  $\mathbf{q} \in \mathbb{Q} \subset \mathbb{R}^6$ . Le domaine atteignable  $\mathbb{D}$  est alors une variété de  $\mathbb{R}^6$ , dont la dimension  $d$  est « presque partout » (en excluant les configurations singulières) égale au nombre de degrés de liberté du robot.

Dans ces conditions, le modèle géométrique direct du robot est une application de  $\mathbb{Q}$  vers  $\mathbb{D}$ , c'est-à-dire qu'un élément de  $\mathbb{Q}$  **n'a qu'une seule image dans  $\mathbb{D}$** . L'inverse n'est pas vrai, c'est-à-dire qu'un élément de  $\mathbb{D}$  **pourra être l'image de plusieurs éléments de  $\mathbb{Q}$** .

On cherche alors à trouver les conditions dans lesquelles l'application est bijective, c'est-à-dire les conditions dans lesquelles un élément de  $\mathbb{D}$  n'est l'image que d'un seul élément de  $\mathbb{Q}$ . Pour cela, on utilise le théorème d'inversion local, qui donne le résultat suivant :

FIGURE 5.2 – Illustration des aspects  $Q_i$  et de leur image  $D_i$ .

**Si dans un sous-domaine de  $Q$ , noté  $Q'$ , le déterminant de  $J(\mathbf{q})$  ( $\mathbf{q} \in Q'$ ) garde un signe constant, alors l'application est bijective dans ce sous-domaine.**

Ceci permet de donner une description complète du domaine atteignable. En effet, la relation  $\det \mathbf{J}(\mathbf{q}) = 0$  définit une partition de  $Q$  en  $m$  domaines disjoints notés  $Q_i$  ( $i = 1, \dots, m$ ) et appelés *aspects*. Un aspect  $Q_i$  est donc défini par :

$$\forall \mathbf{q} \in Q_i, \text{ signe } [\det \mathbf{J}(\mathbf{q})] = \text{constant}$$

Soit  $D_i$ , l'image de  $Q_i$ . Le domaine atteignable  $D$  peut être alors décrit comme la réunion de toutes les images des aspects  $Q_i$  (cf. figure 5.2). On peut écrire les propriétés suivantes :

$$Q = \bigcup_{i=1}^m Q_i \quad ; \quad Q_i \cap Q_j = \emptyset \quad ; \quad D = \bigcup_{i=1}^m D_i \quad ; \quad D_i \cap D_j \neq \emptyset$$

Pour  $\mathbf{q} \in Q_i$ , le robot est dit dans l'aspect  $n^\circ i$ .

La description précédente permet alors de connaître les solutions multiples, c'est-à-dire toutes les configurations possibles pour un  $\mathbf{x}$  donné. En effet, si par exemple le vecteur opérationnel  $\mathbf{x} \in D_i \cap D_j \cap D_k$ , il est l'image de trois vecteurs articulaires différents, appartenant à  $Q_i$ ,  $Q_j$  et  $Q_k$ .

### 5.5.2 Configurations singulières

L'analyse des configurations singulières repose, sur le plan mathématique, sur le calcul du MGI, c'est-à-dire sur la résolution d'un système de  $n$  équations à  $n$  inconnues ( $n$  étant le nombre d'axes du robot). On va alors plutôt examiner les singularités sur le modèle cinématique, qui est linéaire, contrairement au modèle géométrique.

Une configuration singulière correspond à la perte de contrôle d'un ou plusieurs mouvements de l'effecteur et corrélativement à l'apparition d'indéterminées ou de redondances sur les mouvements encore contrôlables, ce qui augmente la vitesse du robot sur ces mouvements. Ceci se produit quand la matrice jacobienne est de rang inférieur au nombre d'axes du robot, c'est-à-dire quand  $\det \mathbf{J}(\mathbf{q}) = 0$ . Dans ce cas, le MCD n'est pas inversible.

Lorsqu'un robot s'approche d'une configuration singulière,  $\det \mathbf{J}(\mathbf{q})$  devient de plus en plus faible, donc les termes de  $\mathbf{J}^{-1}(\mathbf{q})$  risquent de devenir de plus en plus grands, ce qui entraîne, à  $\dot{\mathbf{x}}$  fixé, des vitesses articulaires très importantes, parfois incompatibles avec les possibilités des actionneurs. D'une manière générale, les configurations singulières sont plutôt à éviter, mais pour certaines tâches nécessitant de la vitesse dans une direction particulière, elles peuvent être intéressantes.

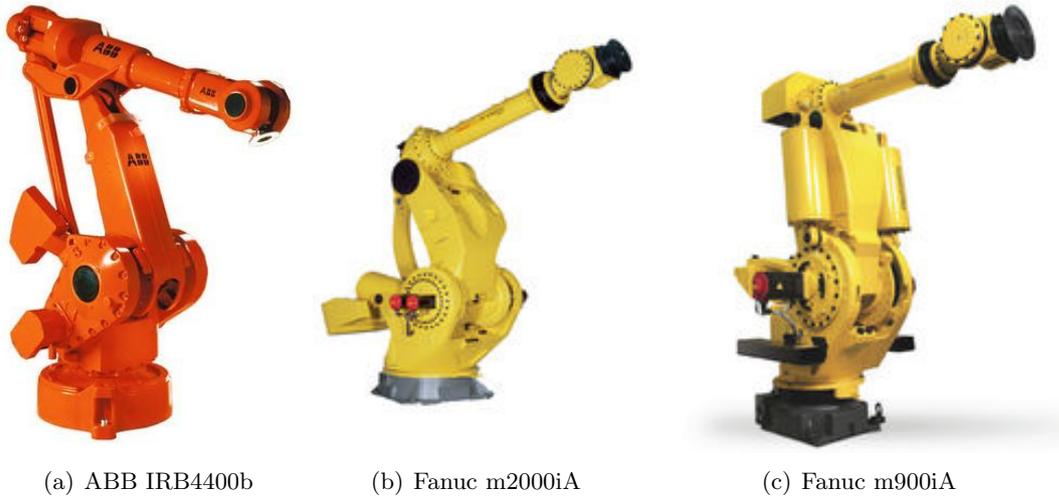


FIGURE 5.3 – Illustration de l'équilibrage statique.

## 5.6 Efforts appliqués par le robot sur son environnement

Les efforts d'interaction  $\Gamma_e$  entre le robot et son environnement, peuvent être calculés directement en utilisant la matrice jacobienne cinématique.

Soit  ${}^0\mathcal{T}_e(P)$  le torseur des efforts appliqués par l'organe terminal du robot sur l'environnement exprimé dans le repère  $R_0$  :

$${}^0\mathcal{T}_e(P) = \begin{bmatrix} \mathbf{f}_e \\ \mathbf{m}_e \end{bmatrix}$$

Le principe des puissances virtuelles permet de calculer les couples moteurs supplémentaires  $\Gamma_e$  pour contrebalancer ces efforts extérieurs en statique :

$$\Gamma_e^T \cdot \dot{\mathbf{q}} = [{}^0\mathcal{T}_e(P)]^T \cdot {}^0\mathcal{V}_{n/0}(P)$$

or :

$${}^0\mathcal{V}_{n/0}(P) = \mathbf{J}^c(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

avec  $\mathbf{J}_n^c(P, \mathbf{q})$  la matrice jacobienne cinématique au point d'application  $P$  de la force extérieure, et donc :

$$\boxed{\Gamma_e = [\mathbf{J}^c(\mathbf{q})]^T \cdot {}^0\mathcal{T}_e(P)}$$

### 5.6.1 Équilibrage statique d'une charge utile

La connaissance des couples nécessaires pour contrebalancer la force de pesanteur peut être utilisée pour équilibrer le robot afin de réduire sa consommation énergétique dans sa posture la plus courante. C'est notamment le cas pour les robots pouvant déplacer une charge utile importante. Cet *équilibrage statique* peut être réalisé par une masse (contrepois) ou par des ressorts comme l'illustre la figure 5.3.

Le torseur des efforts d'une charge utile de masse  $m$  et de centre de masse  $G$  est :

$${}^0\mathcal{T}_e(G) = \begin{bmatrix} m\mathbf{g} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}$$

En transportant ce torseur au point  $P$ , on bitent les efforts et moments que le robot doit exercer pour soulever la charge :

$${}^0\mathcal{T}_e(P) = \begin{bmatrix} m\mathbf{g} \\ \overrightarrow{PG} \wedge m\mathbf{g} \end{bmatrix}$$

Puis, en appliquant le principe des puissances virtuelles, on obtient les couples moteurs supplémentaires nécessaires pour soulever la charge :

$$\mathbf{\Gamma}_e = [\mathbf{J}^c(\mathbf{q})]^T \cdot {}^0\mathcal{T}_e(P) = [\mathbf{J}^c(\mathbf{q})]^T \cdot \begin{bmatrix} m\mathbf{g} \\ \overrightarrow{PG} \wedge m\mathbf{g} \end{bmatrix}$$

### 5.6.2 Évaluation automatique de la masse et de la position du centre de masse d'une charge utile

Tout d'abord, on place le robot immobile sans charge dans une configuration  $\mathbf{q}_0$ . On mesure alors les couples moteurs  $\mathbf{\Gamma}_m$ .

Le robot va ensuite prendre la charge et revient dans la configuration  $\mathbf{q}_0$ . On mesure à nouveau les couples moteurs  $\mathbf{\Gamma}'_m$ .

La différence entre  $\mathbf{\Gamma}'_m$  et  $\mathbf{\Gamma}_m$  ne dépend que de la masse et de la position du centre de masse de la charge. Cette différence se calcule aisément et on obtient :

$$\mathbf{\Gamma}_e = \mathbf{\Gamma}'_m - \mathbf{\Gamma}_m = [\mathbf{J}^c(\mathbf{q})]^T \cdot \begin{bmatrix} m\mathbf{g} \\ \overrightarrow{PG} \wedge m\mathbf{g} \end{bmatrix}$$

Cette relation fournit 6 équations pour 4 inconnues, on en déduit alors  $m$  et les trois coordonnées de  $\overrightarrow{PG}$ .

## 5.7 Conclusion

Le modèle cinématique direct donne les performances en vitesse du robot, ses positions singulières et la partition de son espace articulaire en aspects.

Sous sa forme inverse, le modèle cinématique permet de calculer les consignes de vitesses articulaires en fonction des consignes de vitesses opérationnelles et donc de générer des mouvements standards. De plus, lorsqu'il est impossible d'inverser analytiquement le modèle géométrique direct, on peut mettre en œuvre des méthodes numériques à l'aide de la matrice jacobienne inverse (voir chapitre 7).

La connaissance du modèle cinématique permet également de calculer les efforts articulaires correspondant à un effort statique exercé sur un robot. Il est également nécessaire pour la commande de robot en effort, en vision (et d'une manière plus générale en commande référencée capteur).



## Chapitre 6

# Modèles géométriques et cinématiques des robots parallèles

Un robot parallèle est caractérisé par le fait que son organe terminal est relié à sa base par au moins deux chaînes cinématiques distinctes formant des boucles cinématiques fermées. Ces boucles cinématiques fermées contraignent l'établissement des modèles géométriques et cinématiques.

Ce chapitre présente la modélisation des robots parallèles dans les cas simples. Pour les cas plus complexes, nous renvoyons le lecteur aux ouvrages de références [1, 3, 7].

### 6.1 Modèle géométrique inverse

Le modèle géométrique inverse (MGI) a pour objectif de trouver les coordonnées articulaires permettant à son organe terminal, souvent appelé *plateforme* dans ce cas, d'atteindre une pose opérationnelle donnée. Comme pour les robots séries, le MGI admet plusieurs solutions.

Il est possible de calculer le MGI en résolvant indépendamment chaque bras du robot parallèle comme autant de robots séries. Néanmoins, comme les bras d'un robot parallèles comportent des liaisons passives, la détermination de la valeur de toutes les articulations n'est pas toujours nécessaire.

La méthode présentée ci-dessous permet de trouver directement les coordonnées articulaires sans paramétrer les liaisons passives. Elle s'applique aux robots parallèles usuels (Gough-Stewart, Delta, etc.). Elle est illustrée par la figure 6.1.

On note tout d'abord  $A_i$  la position du centre de l'articulation passive reliant la plateforme au bras  $i$ . On exprime alors les coordonnées de  $A_i$  dans le repère de base  $R_0$  en fonction de la pose désirée de la plateforme  $\mathbf{x}$  dont le centre est représentée par le point  $P$  sur la figure 6.1.

On note ensuite  $B_i$  la position du centre de l'articulation reliant le bras  $i$  à la base du robot. On exprime les coordonnées de  $B_i$  dans le repère de base  $R_0$  en fonction de la position de l'axe  $i$  (ou des axes si plusieurs axes actionnent le bras  $i$ ).

On écrit enfin que, pour chaque bras  $i$ , la distance entre  $A_i$  et  $B_i$  est constante et donc que  $A_i B_i = d_i$ . On obtient au final un système de  $n$  équations non linéaires ( $n$  étant le nombre de bras du robot parallèle).

$$\begin{cases} A_1(\mathbf{x})B_1(q_1) & = d_1 \\ \vdots & \vdots \\ A_n(\mathbf{x})B_n(q_n) & = d_n \end{cases}$$

Ces équations sont appelées *équations de fermeture*.

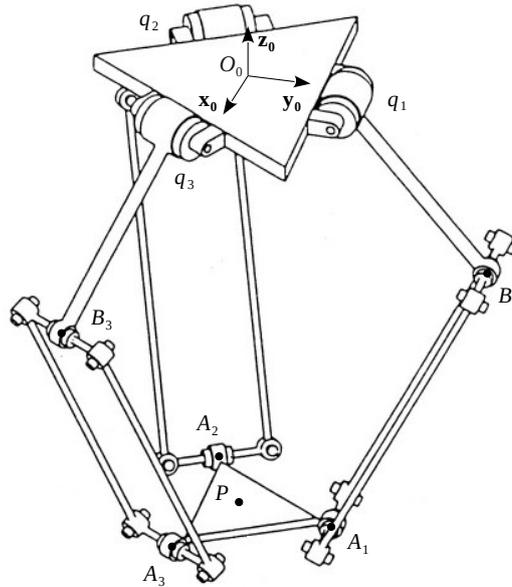


FIGURE 6.1 – Exemple de paramétrage d'un robot Delta.

Pour déterminer le MGI, il suffit alors de résoudre indépendamment chaque équation,  $\mathbf{x}$  étant connu. Cette résolution est aisée pour les robots parallèles industriels et fait appel aux formules trigonométriques répertoriés dans le tableau 4.1. Chaque équation donne généralement plusieurs solutions.

Notons que dans le cas d'une plateforme de Gough-Stewart, la distance  $d_i$  est exprimée en fonction de  $q_i$  et donc les équations de fermetures sont de la forme :

$$\begin{cases} A_1(\mathbf{x})B_1 = d_1(q_1) \\ \vdots \\ A_n(\mathbf{x})B_n = d_n(q_n) \end{cases}$$

## 6.2 Modèle géométrique direct

Le modèle géométrique direct (MGD) consiste à déterminer la pose de la plateforme en fonction des coordonnées articulaires. Pour cela, il faut résoudre le système des équations de fermetures précédent connaissant les  $q_i$ . Dans le cas général, c'est un problème très difficile à résoudre. Contrairement à un robot série, le MGD d'un robot parallèle admet plusieurs solutions appelées *modes d'assemblage*. Par exemple, la plateforme de Gough-Stewart peut avoir jusqu'à 40 modes d'assemblages.

Cependant, pour certaines architectures à trois bras comme le Delta ou le 3(PPSR), le problème est équivalent à l'intersection de trois sphères et peut donc être facilement résolu.

## 6.3 Modèle cinématique et singularités

Le modèle cinématique d'un robot parallèle peut être obtenu en différentiant les équations de fermeture ou à l'aide des torseurs cinématiques dans le cas où l'on a également déterminé la configurations des articulations passives. Nous ne traitons ici que le premier cas.

En dérivant les équations de fermeture par rapport au temps, puis en regroupant les termes, on obtient :

$$\mathbf{J}_x(\mathbf{x}, \mathbf{q}) \cdot \dot{\mathbf{x}} = \mathbf{J}_q(\mathbf{x}, \mathbf{q}) \cdot \dot{\mathbf{q}}$$

$\mathbf{J}_x(\mathbf{x}, \mathbf{q})$  est appelée matrice jacobienne série et  $\mathbf{J}_q(\mathbf{x}, \mathbf{q})$  est appelée matrice jacobienne parallèle.

Dans le cas où le nombre d'articulations correspond à la dimension du vecteur  $\dot{\mathbf{x}}$  et que les matrices jacobiennes sont de rang plein, le modèle cinématique du robot est donnée par :

$$\dot{\mathbf{x}} = \mathbf{J}_x^{-1}(\mathbf{x}, \mathbf{q}) \cdot \mathbf{J}_q(\mathbf{x}, \mathbf{q}) \cdot \dot{\mathbf{q}}$$

Si l'une ou l'autre des matrices jacobiennes ne sont pas de rang plein, le robot est dans une configuration singulière. On distinguera les singularités de type 1 (singularités séries) qui apparaissent quand :

$$\det \mathbf{J}_q(\mathbf{x}, \mathbf{q}) = 0$$

Dans ces configurations, le robot perd un ou plusieurs degrés de liberté au niveau de son organe terminal.

Les singularités de type 2 (singularités parallèles) surviennent quand :

$$\det \mathbf{J}_x(\mathbf{x}, \mathbf{q}) = 0$$

Dans ces configurations, un ou plusieurs degrés de liberté du robot ne sont plus contraints et des mouvements libres sont possibles.

Enfin, quand les deux matrices jacobiennes sont déficientes simultanément, on parle de singularité de type 3.

$$\det \mathbf{J}_q(\mathbf{x}, \mathbf{q}) = \det \mathbf{J}_x(\mathbf{x}, \mathbf{q}) = 0$$



# Chapitre 7

## Génération de mouvement

La génération de mouvement a pour objectif de calculer les consignes en position, vitesse et accélération des axes du robot permettant de réaliser une tâche. Cette tâche est décrite à l'aide d'un programme constitué d'une séquence d'instructions élémentaires. En fonction des instructions, la génération de mouvement peut s'effectuer dans l'espace articulaire ou dans l'espace opérationnel.

### 7.1 Programmation d'un robot

On trouve principalement deux types de programmation : l'*apprentissage* et la *programmation hors ligne*.

#### 7.1.1 Programmation par apprentissage

La programmation par apprentissage consiste à mémoriser, directement sur le site, les points et/ou les trajectoires à suivre. Pour cela, l'opérateur peut piloter le robot en mode manuel à l'aide du *teach pendant* dans différents repères (repère atelier, repère outil, repère lié à une tâche, etc.).

Il est également possible d'utiliser un dispositif d'apprentissage direct fixé au poignet du robot. L'opérateur guide alors directement l'organe terminal du robot à l'aide d'une poignée spéciale dotée d'un capteur de force. Une commande en force déplace le robot afin de maintenir un effort nul sur la poignée.

#### 7.1.2 Programmation hors ligne

Une autre manière de programmer un robot est d'utiliser la programmation hors ligne (PHL). Cela consiste à réaliser, tester et programmer les mouvements du robot en simulation (CAO robotique) et, une fois cette étape terminée, à charger le programme directement dans l'armoire de commande du robot réel. Le temps passé sur le site s'en trouve en principe réduit, au profit du bureau d'études. Cela permet donc de programmer un robot pour une nouvelle tâche sans interrompre son fonctionnement en production, mais également de réaliser des tâches très difficiles à programmer par apprentissage (ex : suivi de trajectoires gauches ou choix d'une configuration pour atteindre un point « délicat »). La programmation hors ligne impose que le robot soit bien étalonné et que sa position vis-à-vis de la tâche soit précisément connue.

#### 7.1.3 Langages et programmes

Quelque soit la méthode utilisée, les trajectoires sont représentées sous forme d'une séquence de mouvements élémentaires formant un programme. Le langage de programmation est propre à chaque fabricant. Il n'existe pas de langage robotique normalisé.

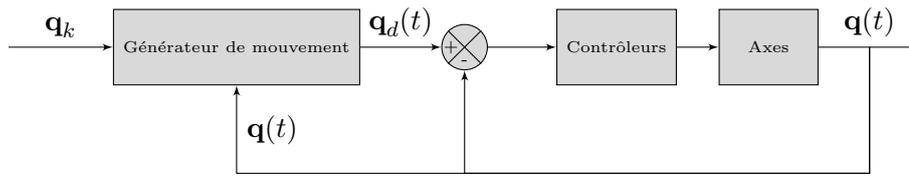


FIGURE 7.1 – Génération de mouvement dans l'espace articulaire ( $\mathbf{q}_k$  désigne la position finale à atteindre,  $\mathbf{q}_d(t)$  la position désirée et  $\mathbf{q}(t)$  la position courante des axes).

Cependant, on retrouvera dans tous les langages, des instructions permettant de déplacer l'organe terminal d'un point à un autre avec une trajectoire libre ou avec une trajectoire contrainte.

Ces instructions génériques permettent donc de réaliser :

- des mouvements de type « point-à-point », où l'on souhaite que le robot passe d'une configuration à une autre, sans se préoccuper de la trajectoire, si ce n'est pour aller le plus vite possible ; ces mouvements sont générés dans l'espace articulaire et sont adaptés aux phases d'approches, aux tâches de prise-dépose, de soudure par point, etc.
- des mouvements de type « interpolés », où l'on souhaite au contraire maîtriser toute la trajectoire de l'effecteur, avec une loi horaire donnée (rectiligne, circulaire ou autre) ; ces mouvements sont générés dans l'espace opérationnel et sont adaptés à la soudure en continu, à la dépose de colle, à la peinture et aux tâches d'usinage, de découpe, d'ébavurage, etc.
- des mouvements de type « référencés capteurs », c'est-à-dire que le robot se déplace à partir d'une information provenant d'un capteur mesurant l'environnement (système de vision pour repérage et suivi de cible, capteur d'effort pour le contrôle des efforts de contact avec l'environnement). Ceci peut, le cas échéant, correspondre à une boucle de compliance active pour des tâches d'insertion, de polissage, etc.

## 7.2 Génération de mouvement dans l'espace articulaire

La commande point-à-point, probablement la majorité des mouvements demandés aux robots, correspond au cas où la tâche est définie comme une succession de situations à atteindre par l'effecteur du robot. On cherche donc à générer les trajectoires pour que le déplacement entre deux configurations consécutives  $\mathbf{q}_k$  à  $\mathbf{q}_{k+1}$  soit le plus rapide possible. Si le robot doit s'arrêter à  $\mathbf{q}_{k+1}$  pour y exécuter une tâche, c'est-à-dire avoir une vitesse nulle dans cette situation, il s'agit d'un *point d'arrêt*. Si le robot doit simplement passer par ce point sans s'y arrêter, pour éviter un obstacle par exemple, il s'agit d'un *point de passage*.

La commande du robot s'effectue donc dans l'espace articulaire comme l'illustre le schéma de la figure 7.1. La commande dans l'espace articulaire ne nécessite pas l'utilisation du MGI et n'est donc pas affectée par les configurations singulières.

En revanche, le déplacement de l'effecteur en ligne droite dans l'espace opérationnel, qui est la trajectoire la plus courte de celui-ci entre deux points, **ne correspond donc pas nécessairement à la trajectoire la plus rapide**, qui consiste à amener les axes du robot le plus rapidement possible de  $\mathbf{q}_k$  à  $\mathbf{q}_{k+1}$ , ce qui dans le cas général ne conduit pas à un déplacement rectiligne de l'effecteur dans l'espace opérationnel.

Plusieurs méthodes existent pour générer une trajectoire point-à-point en temps minimal. Nous ne développons ici que la loi bang-bang avec palier de vitesse pour des raisons didactiques mais la plupart des contrôleurs industriels génèrent des trajectoires plus douces, à jerk constant.

### 7.2.1 Loi bang-bang avec palier de vitesse

On s'intéresse ici à la génération de mouvement entre deux points d'arrêts, de  $\mathbf{q}_k$  à  $\mathbf{q}_{k+1}$ . On pose  $\Delta\mathbf{q} = \mathbf{q}_{k+1} - \mathbf{q}_k$ .

Sur chaque axe  $i$ , la vitesse maximale est notée  $v_i$  et l'accélération maximale est notée  $a_i$ . Ces valeurs peuvent se déduire directement des capacités physiques du robot. On note également  $T_i$  le temps nécessaire à l'axe  $i$  pour effectuer le mouvement. Enfin, pour simplifier l'exposé, on supposera  $\Delta q_i > 0$  pour tous les axes  $i$ .

En fonction, de  $\Delta q_i$ ,  $v_i$  et  $a_i$ , deux situations sont possibles : une loi *bang-bang* ou une loi *bang-bang avec palier de vitesse*.

Dans une loi bang-bang, l'accélération maximale est maintenue sur la première moitié du mouvement, puis l'opposée de l'accélération maximale est appliquée sur le reste du mouvement (cf. figure 7.2). Ceci n'est réalisable que si la vitesse reste inférieure à la vitesse maximale admissible par l'axe  $i$ .

L'accélération étant constante et la vitesse nulle au départ, la distance parcourue par l'articulation  $i$  pendant  $T_i/2$  est :

$$\frac{\Delta q_i}{2} = \frac{1}{2} a_i \left( \frac{T_i}{2} \right)^2$$

d'où :

$$\boxed{\frac{T_i}{2} = \sqrt{\frac{\Delta q_i}{a_i}}}$$

La vitesse en cet instant est alors :

$$a_i \cdot \frac{T_i}{2} = a_i \cdot \sqrt{\frac{\Delta q_i}{a_i}} = \sqrt{a_i \cdot \Delta q_i}$$

Cette vitesse doit être inférieure à  $v_i$  pour pouvoir appliquer la loi bang-bang, c'est-à-dire que :

$$\sqrt{a_i \cdot \Delta q_i} \leq v_i$$

soit :

$$\boxed{\Delta q_i \leq \frac{v_i^2}{a_i}}$$

Dans le cas contraire, la vitesse maximale est atteinte avant  $T_i/2$ . Dans ce cas, on utilise une loi bang-bang avec palier de vitesse (cf. figure 7.2). La vitesse maximale est atteinte à l'instant  $t_i = v_i/a_i$ , puis l'accélération est nulle jusqu'à  $T_i - t_i$  où le freinage commence. On a finalement :

$$\boxed{T_i = \frac{\Delta q_i}{v_i} + t_i}$$

Selon que l'on atteint ou non les limites de vitesse et d'accélération du robot, on réalisera pour chaque axe un déplacement en temps minimal selon la loi bang-bang avec ou sans palier de vitesse.

### 7.2.2 Synchronisation des axes du robot

La synchronisation des axes, non indispensable, permet de régler l'ensemble des axes sur l'axe le plus lent et ainsi de moins solliciter les actionneurs sans dégrader le temps de déplacement de l'effecteur d'un point au suivant.

La méthode consiste à ralentir les axes les plus rapides pour qu'ils terminent leur mouvement en même temps que l'axe le plus lent. Il en effet inutile de donner leur vitesse maximale aux

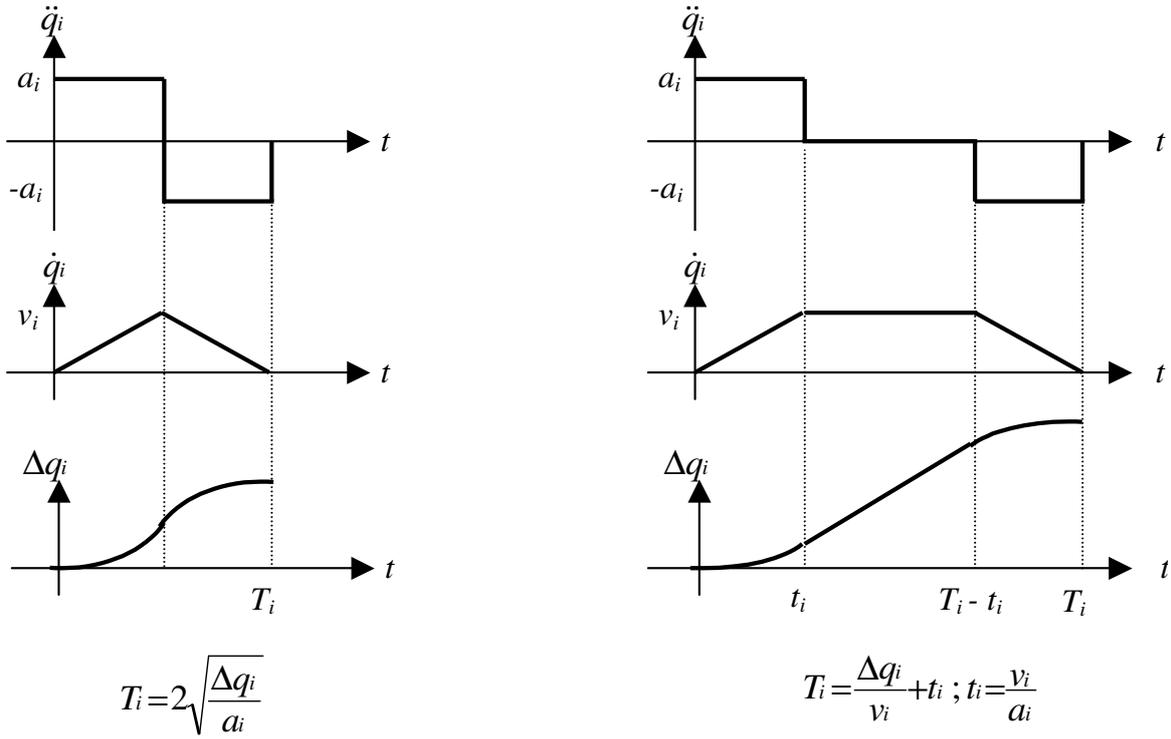


FIGURE 7.2 – Loi bang-bang sans (à gauche) ou avec palier de vitesse (à droite).

autres axes, ce qui les sollicite inutilement, mécaniquement et énergétiquement, car c'est de toute façon l'axe le plus lent qui fixe le temps mis pour effectuer le mouvement d'ensemble.

Il est également intéressant que les phases d'accélération et de décélération soient complètement synchrones, c'est-à-dire que  $t_i = t_j$  quels que soient  $i$  et  $\mathbf{J}$ . Ceci permet notamment d'obtenir des asservissements plus précis.

On souhaite donc avoir  $T_i = T_j = T$  et  $t_i = t_j = t$  quels que soient  $i$  et  $\mathbf{J}$ . Pour cela, on associe à chaque articulation des coefficients  $\nu_i$  et  $\lambda_i$  inférieurs 1. L'accélération de l'axe  $i$  est ainsi  $\nu_i a_i$  et sa vitesse  $\lambda_i v_i$ .

Chaque articulation suivant une loi bang-bang avec palier de vitesse (cf. figure 7.2). On peut écrire pour tout  $i$  :

$$T = \frac{\Delta q_1}{\lambda_1 v_1} + \frac{\lambda_1 v_1}{\nu_1 a_1} = \frac{\Delta q_i}{\lambda_i v_i} + \frac{\lambda_i v_i}{\nu_i a_i}$$

ainsi que :

$$t = \frac{\lambda_1 v_1}{\nu_1 a_1} = \frac{\lambda_i v_i}{\nu_i a_i}$$

On en déduit que :

$$\lambda_i = \lambda_1 \frac{v_1 \Delta q_i}{v_i \Delta q_1}$$

Or  $0 \leq \lambda_i \leq 1$ , donc :

$$0 \leq \lambda_1 \leq \frac{v_i \Delta q_1}{v_1 \Delta q_i}$$

On montre de même que :

$$0 \leq \nu_1 \leq \frac{a_i \Delta q_1}{a_1 \Delta q_i}$$

---

**Algorithme 1 :** Algorithme de génération de loi bang-bang avec palier de vitesse et synchronisation homothétique des axes.

---

**Données :**  $q_{i,k}$  les coordonnées articulaires initiales de chaque axe  
 $q_{i,k+1}$  les coordonnées articulaires à atteindre par chaque axe  
 $v_i$  les vitesses maximales de chaque axe  
 $a_i$  les accélérations maximales de chaque axe

**début**

$$\lambda_1 \leftarrow \min_{i=2\dots n} \left[ 1, \frac{v_i \Delta q_1}{v_1 \Delta q_i} \right];$$

$$\nu_1 \leftarrow \min_{i=2\dots n} \left[ 1, \frac{a_i \Delta q_1}{a_1 \Delta q_i} \right];$$

**pour** chaque articulation  $i$  **faire**

$$\lambda_i \leftarrow \lambda_1 \frac{v_1 \Delta q_i}{v_i \Delta q_1};$$

$$\nu_i \leftarrow \nu_1 \frac{a_1 \Delta q_i}{a_i \Delta q_1};$$

**fin**

$$T_{max} \leftarrow \frac{\Delta q_1}{\lambda_1 \nu_1} + \frac{\lambda_1 v_1}{\nu_1 a_1};$$

**si**  $\frac{\lambda_1 v_1}{\nu_1 a_1} < \frac{T_{max}}{2}$  **alors**

$$t \leftarrow \frac{\lambda_1 v_1}{\nu_1 a_1};$$

**sinon**

$$t \leftarrow \frac{T_{max}}{2};$$

**fin**

**fin**

**Résultat :**  $T_{max}$ ,  $t$ ,  $\nu_i$  et  $\lambda_i$  pour chaque axe  $i$

---

Le temps de déplacement  $T$  est minimal quand  $\nu_1$  et  $\lambda_1$  sont maximales. On choisit alors :

$$\begin{cases} \lambda_1 = \min_{i=2\dots n} \left[ 1, \frac{v_i \Delta q_1}{v_1 \Delta q_i} \right] \\ \nu_1 = \min_{i=2\dots n} \left[ 1, \frac{a_i \Delta q_1}{a_1 \Delta q_i} \right] \end{cases}$$

Si le déplacement est très court, le palier peut être absent et on retombe alors sur une loi bang-bang simple pour tous les axes. Cette situation arrive quand  $\frac{\lambda_1 v_1}{\nu_1 a_1} \geq \frac{T_{max}}{2}$ , on fixera alors  $t$  à  $\frac{T_{max}}{2}$ .

L'algorithme 1 résume la génération de loi bang-bang avec palier de vitesse et synchronisation homothétique des axes. C'est ce type de loi qui est généralement implantée sur les robots industriels.

Une fois calculés les coefficients  $\nu_i$  et les temps  $T_{max}$  et  $t_i$  par l'une des deux méthodes, la loi temporelle envoyée au contrôleur de chaque axe  $i$  est définie par :

$$q_{d,i}(t) = \begin{cases} q_{k,i} + \frac{1}{2} \nu_i a_i t^2 & \text{pour } 0 \leq t < t_i \\ q_{k,i} - \frac{1}{2} \nu_i a_i t_i^2 + \nu_i a_i t_i t & \text{pour } t_i \leq t < T_{max} - t_i \\ q_{k,i} + \nu_i a_i t_i (T_{max} - t_i) - \frac{1}{2} \nu_i a_i (t - T_{max})^2 & \text{pour } T_{max} - t_i \leq t < T_{max} \end{cases}$$

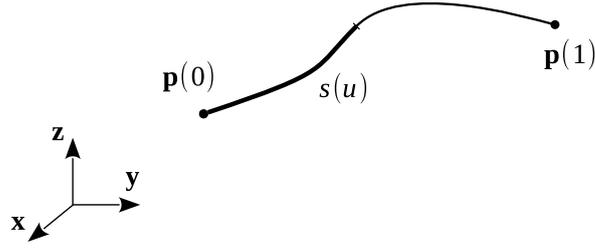


FIGURE 7.3 – Exemple d’arc paramétré définissant la trajectoire de l’organe terminal dans l’espace opérationnel.

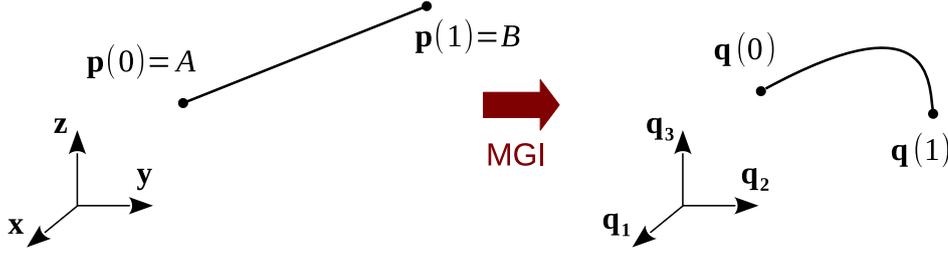


FIGURE 7.4 – Illustration de la génération d’une trajectoire rectiligne de l’organe terminal.

### 7.3 Génération de mouvement dans l’espace opérationnel

Pour certaines applications, il est nécessaire de contrôler la trajectoire de l’organe terminal à cours du temps par une loi horaire.

La trajectoire de l’organe terminal (position et orientation) peut être décrite à l’aide d’un arc paramétré  $\mathbf{p}(u)$  dans l’espace opérationnel,  $\mathbf{p}(u)$  étant de classe  $\mathcal{C}^2$  et défini de  $[0; 1]$  dans  $\mathbb{R}^3$ .

Afin de déterminer la loi horaire de l’organe terminal, on utilise l’abscisse curviligne et un profil de vitesse. L’abscisse curviligne de  $\mathbf{p}(u)$  est définie par :

$$s(u) = \int_0^u \left\| \frac{d\mathbf{p}}{du} \right\| dx$$

On souhaite généralement imposer une vitesse tangentielle constante  $v$  sur cette trajectoire, soit :

$$\frac{ds}{dt}(u) = v(u) = \frac{d}{dt} \int_0^u \left\| \frac{d\mathbf{p}}{du} \right\| dx$$

Cette équation admet des solutions analytiques pour des trajectoires simples comme une ligne droite ou un arc de cercle. Pour des trajectoires plus complexes, on utilisera un algorithme d’intégration numérique.

Pour une trajectoire rectiligne entre une position initiale  $A$  et une position finale  $B$ , on a :

$$\mathbf{p}(u) = \overrightarrow{O_0A} + \overrightarrow{AB}.u$$

Si on souhaite imposer une vitesses  $v$  constante sur cette trajectoire, le temps de parcours est alors  $T_{max} = \frac{\|\overrightarrow{AB}\|}{v}$  et donc :

$$u(t) = \frac{t}{T_{max}} = \frac{v}{\|\overrightarrow{AB}\|}.t$$

La loi horaire dans l’espace opérationnel s’écrit au final :

$$\mathbf{x}_d(t) = \mathbf{p}(u(t)) = \overrightarrow{O_0A} + \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|}.v.t$$

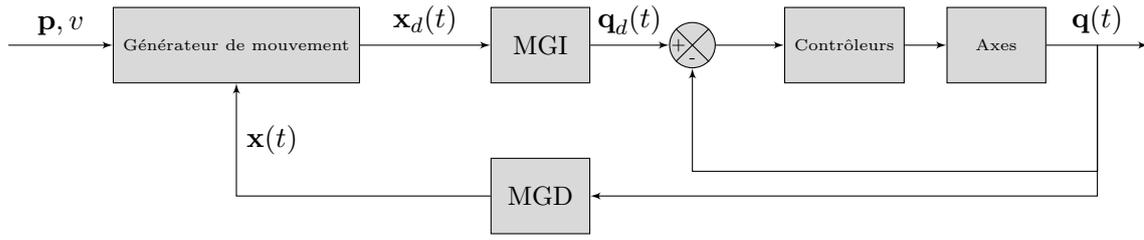


FIGURE 7.5 – Génération de mouvement dans l'espace opérationnel ( $\mathbf{x}_k$  désigne la position finale à atteindre,  $\mathbf{x}_d(t)$  la position désirée et  $\mathbf{x}(t)$  la position courante de l'organe terminale.

La poursuite de la trajectoire  $\mathbf{x}_d(t)$  par l'organe terminal du robot consiste à fournir à sa commande numérique une succession de valeurs  $\mathbf{x}_{d_k}$  où l'indice  $k$  correspond à l'instant d'échantillonnage  $kT$ .

Un transformateur de coordonnées doit donc fournir aux asservissements des axes la série de valeurs  $\mathbf{q}_{d_k}$ , correspondant aux  $\mathbf{x}_{d_k}$ . La commande du robot dans l'espace opérationnel nécessite donc l'utilisation du modèle géométrique direct et inverse pour transformer les coordonnées d'un espace à l'autre comme l'illustre la figure 7.5.

L'utilisation du MGI pose plusieurs problèmes. D'une part, il faut s'assurer que la position opérationnelle désirée est atteignable pour que le MGI admette au moins une solution. Il faut également vérifier que la position désirée appartienne au même aspect que la position courante.

D'autre part, le franchissement d'une configuration singulière peut poser problème, puisque toutes les trajectoires ne sont pas admissibles en raison de la perte de degrés de liberté. Nous verrons à la fin du chapitre une technique pour réaliser ce franchissement.

Enfin, il faut s'assurer, par exemple à l'aide du modèle cinématique, que la trajectoire est réalisable par le robot du point de vue des vitesses et des accélérations articulaires.

## 7.4 Inversion numérique du modèle géométrique direct

Dans le cas où l'inversion analytique du MGD s'avère complexe ou impossible, on peut calculer  $\mathbf{q}$  en fonction de  $\mathbf{x}$  par inversion numérique, en utilisant l'inverse de la matrice jacobienne. Le problème est donc ici de calculer le vecteur articulaire  $\mathbf{q}^*$  correspondant à un vecteur opérationnel désiré  $\mathbf{x}^*$ . Le vecteur  $\mathbf{q}^*$  cherché est donc une solution de l'équation :

$$\mathbf{x}^* = \mathbf{f}(\mathbf{q})$$

A partir de l'équation précédente, on introduit la fonction d'erreur suivante :

$$\mathbf{e}(\mathbf{q}) = \mathbf{f}(\mathbf{q}) - \mathbf{x}^*$$

Le problème de l'inversion revient alors à trouver un zéro de cette fonction d'erreur. On peut pour ce faire utiliser l'algorithme de Newton-Raphson (cf. algorithme 2).

Partant d'un point  $\mathbf{q}_0$  (de préférence proche du zéro à trouver), la méthode consiste à une approximation affine  $\mathbf{p}$  de  $\mathbf{e}$  autour du point  $\mathbf{q}_0$  à l'aide de son développement de Taylor au premier ordre :

$$\mathbf{p}(\mathbf{q}) = \mathbf{e}(\mathbf{q}_0) + \mathbf{J}(\mathbf{q}_0) \cdot (\mathbf{q} - \mathbf{q}_0)$$

avec  $\mathbf{J}(\mathbf{q}_0)$  la matrice jacobienne de la fonction  $\mathbf{e}$ . La jacobienne de  $\mathbf{e}$  est identique à celle de  $\mathbf{f}$  puisque  $\mathbf{x}^*$  est constant.

On calcule alors un nouveau point  $\mathbf{q}_1$  solution de l'équation :

$$\mathbf{e}(\mathbf{q}_0) + \mathbf{J}(\mathbf{q}_0) \cdot (\mathbf{q} - \mathbf{q}_0) = 0$$

---

**Algorithme 2 :** Algorithme de Newton-Raphson pour l'inversion numérique du MGD.

---

**Données :**  $\mathbf{x}^*$  les coordonnées opérationnelles à atteindre ;  
 $\mathbf{J}(\mathbf{q})$  la matrice jacobienne ;  
 $\mathbf{q}_0$ , tel que  $\mathbf{x}_0 = \mathbf{f}(\mathbf{q}_0)$  ne soit « pas trop éloigné » de  $\mathbf{x}^*$  ;  
**début**  
  **tant que** *critère d'arrêt non satisfait* **faire**  
  |  $\mathbf{q}_{k+1} \leftarrow \mathbf{q}_k + [\mathbf{J}(\mathbf{q}_k)]^{-1} \cdot (\mathbf{x}^* - \mathbf{f}(\mathbf{q}_k))$  ;  
  **fin**  
**fin**  
**Résultat :**  $\mathbf{q}_{k+1}$

---

Soit :

$$\mathbf{q}_1 = \mathbf{q}_0 - [\mathbf{J}(\mathbf{q}_0)]^{-1} \cdot \mathbf{e}(\mathbf{q}_0)$$

Sous certaines hypothèses, le point  $\mathbf{q}_1$  sera plus proche du vrai zéro de  $\mathbf{e}$  que le point  $\mathbf{q}_0$  précédent.

On peut ensuite réitérer le processus, c'est-à-dire mettre définir  $\mathbf{q}_{k+1}$  comme la solution de l'équation :

$$\mathbf{e}(\mathbf{q}_k) + \mathbf{J}(\mathbf{q}_k) \cdot (\mathbf{q} - \mathbf{q}_k) = 0$$

soit en remplaçant la fonction  $\mathbf{e}$  :

$$\mathbf{x}^* - \mathbf{f}(\mathbf{q}_k) + \mathbf{J}(\mathbf{q}_k) \cdot (\mathbf{q} - \mathbf{q}_k) = 0$$

Ce qui correspond à définir la récurrence suivante :

$$\boxed{\mathbf{q}_{k+1} = \mathbf{q}_k - [\mathbf{J}(\mathbf{q}_k)]^{-1} \cdot (\mathbf{f}(\mathbf{q}_k) - \mathbf{x}^*)}$$

La convergence de cette suite est quadratique à condition que  $\mathbf{x}^*$  soit atteignable, que le vecteur initial  $\mathbf{q}_0$  ne soit « pas trop éloigné » de  $\mathbf{q}^*$  et que tous les  $\mathbf{J}(\mathbf{q}_k)$  calculés soient inversibles. Elle ne permet par contre que le calcul d'une seule solution au problème, alors qu'il pourrait éventuellement y avoir plusieurs configurations possibles pour une même disposition opérationnelle  $\mathbf{x}^*$ .

En pratique, on arrête le calcul quand un critère d'arrêt se vérifie ou quand on a effectué un certain nombre d'itération. Les critères d'arrêt possibles sont :

$$\|\mathbf{f}(\mathbf{q}_k) - \mathbf{x}^*\| < \varepsilon_1 \quad \text{et} \quad \|\mathbf{q}_{k+1} - \mathbf{q}_k\| < \varepsilon_2$$

où  $\varepsilon_1, \varepsilon_2 \in \mathbb{R}^+$  représentent des erreurs d'approximations caractérisant la qualité de la solution numérique.

#### 7.4.1 Commande incrémentale d'un robot

Si, à chaque instant  $kT$ , le transformateur de coordonnée ne peut calculer  $\mathbf{q}_{d_k}$  à partir du MGI, il est possible d'utiliser l'algorithme précédent sous la forme de la « commande incrémentale » suivante :

$$\mathbf{q}_{d_{k+1}} = \mathbf{q}_{d_k} + [\mathbf{J}(\mathbf{q}_{d_k})]^{-1} \cdot (\mathbf{x}_{d_{k+1}} - \mathbf{f}(\mathbf{q}_{d_k}))$$

Cette commande incrémentale sera d'autant plus précise que les  $\delta \mathbf{q}_k = \mathbf{q}_{d_{k+1}} - \mathbf{q}_{d_k}$  et  $\delta \mathbf{x}_k = \mathbf{x}_{d_{k+1}} - \mathbf{x}_{d_k}$  seront petits. Pour que cette méthode puisse être mise en œuvre, il faut :

— la connaissance de valeurs initiales  $\mathbf{x}_0$  et  $\mathbf{q}_0$ ,

- une période d'échantillonnage suffisamment faible vis-à-vis des vitesses de déplacement mises en jeu,
- une remise à jour à chaque pas d'échantillonnage de la jacobienne  $\mathbf{J}(\mathbf{q}_{d_k})$ ,
- la résolution d'un système d'équations linéaires (dans la mesure où la jacobienne est inversible et bien conditionnée).

On peut noter qu'il est généralement préférable de ne pas générer de trajectoires passant « trop près » des configurations singulières, afin d'éviter les problèmes de stabilité numérique, voire de divergence de l'algorithme d'inversion.

#### 7.4.2 Inversion de la matrice jacobienne

Jusqu'à présent, nous avons considéré que la jacobienne  $\mathbf{J}$  était inversible. Pourtant, il y a de nombreux cas où  $\mathbf{J}$  n'est pas inversible.

D'une part, lorsque le robot est redondant (nombre de degrés de liberté  $d$  inférieur au nombre d'axes  $n$ ) la matrice  $\mathbf{J}$ , de dimension  $(d \times n)$ , n'est pas carrée. Dans ce cas, on peut par exemple fixer  $(n-d)$  variations articulaires  $\delta q_i$  et calculer les autres en fonctions des  $\delta x_i$  en réduisant la matrice  $\mathbf{J}$ .

Le problème est alors que l'on n'a pas forcément de critère clair pour savoir quels  $\delta q_i$  on fixe et à quelle valeur. On peut alors faire appel au calcul de la *pseudo-inverse* de  $\mathbf{J}$ , notée  $\mathbf{J}^+$ , qui s'obtient mathématiquement par minimisation d'un critère quadratique (matrice qui vérifie  $\mathbf{J}\mathbf{J}^+ = \mathbf{I}$ )

En effet, si l'on souhaite minimiser l'incrément de déplacement articulaire, autrement dit dans le cas d'une commande incrémentale minimiser les vitesses articulaires, on cherche à minimiser le critère quadratique  $\frac{1}{2}\mathbf{q}^T \cdot \mathbf{q}$  et on obtient la pseudo-inverse suivante :

$$\mathbf{J}^+ = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1}$$

**NB:** Du point de vue mathématique, la solution générale de l'inversion du modèle cinématique est :  $\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^+ \mathbf{J})\mathbf{z}$ , où  $\mathbf{z}$  désigne un vecteur arbitraire de  $\mathbb{R}^n$ . Le premier terme donne, comme vu précédemment, une solution minimisant la norme euclidienne  $\|\dot{\mathbf{q}}\|^2$ . Le second terme peut être utilisé pour satisfaire des contraintes d'optimisation supplémentaires, comme par exemple l'éloignement des butées articulaires, l'éloignement des configurations singulières, les efforts moteurs des actionneurs ou encore la manipulabilité du robot, c'est-à-dire son aptitude à générer les déplacements demandés à partir d'une configuration donnée.

D'autre part, il a été vu précédemment que dans le cas de configurations singulières ou de redondances, le déterminant de la matrice jacobienne devient nul, ce qui interdit l'inversion de cette matrice.

On peut alors, comme dans le cas précédent, utiliser la pseudo-inverse de  $\mathbf{J}$ , qui permet de calculer des déplacements articulaires minimisant un critère. Le déplacement engendré du robot ne sera peut être pas exactement le même que celui  $\delta \mathbf{x}$  souhaité, mais peut être acceptable jusqu'à ce que la singularité soit franchie.

## 7.5 Conclusion

Réalisée dans l'espace articulaire ou dans l'espace opérationnel, la génération de mouvement fournit une loi horaire  $q_{d,i}(t)$  pour chacun des axes  $i$ . Il reste ensuite que chaque position articulaire  $q_i$  (rotation ou translation) soit asservie à la consigne articulaire  $q_{d,i}(t)$  correspondante.



## Chapitre 8

# Modélisation dynamique et commande des axes

Le comportement d'un robot est fortement non linéaire, variable dans le temps (dépendant de la configuration) et à couplage dynamique (le mouvement d'un axe influe sur celui des autres axes). Pour dimensionner les actionneurs et calculer les lois de commandes des axes, il est nécessaire de connaître et modéliser ces effets dynamiques.

### 8.1 Définition

**Définition 14.** On appelle modèle dynamique inverse, l'application donnant les efforts moteurs articulaires  $\mathbf{\Gamma}_m$  (forces ou couples générés par les actionneurs) en fonction des positions  $\mathbf{q}$ , des vitesses  $\dot{\mathbf{q}}$ , des accélérations  $\ddot{\mathbf{q}}$ , des efforts de frottements  $\mathbf{\Gamma}_f$  (visqueux et/ou secs) et des efforts d'interaction  $\mathbf{\Gamma}_e$  entre le robot et son environnement :

$$\mathbf{\Gamma}_m = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{\Gamma}_f, \mathbf{\Gamma}_e)$$

Le modèle dynamique inverse est non linéaire, couplé (le mouvement d'un axe influe sur celui des autres axes) et dépendant de la configuration et des vitesses.

Le modèle dynamique peut être obtenu en utilisant le formalisme d'Euler-Lagrange ou le principe fondamental de la dynamique (formalisme de Newton-Euler). Dans la suite, nous développerons succinctement la méthode utilisant le formalisme d'Euler-Lagrange. Pour plus de détails sur cette méthode, nous renvoyons le lecteur aux nombreux ouvrages de références de mécanique (par exemple [10]).

### 8.2 Modèle dynamique inverse

#### 8.2.1 Formalisme de Lagrange

Le Lagrangien est généralement défini par la différence entre l'énergie cinétique totale  $\mathcal{K}$  du système (ici le robot) et l'énergie potentielle totale  $\mathcal{P}$  du système :

$$\mathcal{L} = \mathcal{K} - \mathcal{P}$$

Les équations d'Euler-Lagrange s'écrivent alors :

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

où  $\Gamma_i$  est l'effort au niveau de l'articulation  $i$ .

### 8.2.2 Calcul de l'énergie cinétique

L'énergie cinétique du segment  $i$  s'écrit :

$$\mathcal{K}_i = \frac{1}{2} m_i {}^0\mathbf{v}_{i/0}^T(G_i) \cdot {}^0\mathbf{v}_{i/0}(G_i) + \frac{1}{2} {}^0\boldsymbol{\omega}_{i/0}^T \cdot \bar{\mathbf{I}}_i \cdot {}^0\boldsymbol{\omega}_{i/0}$$

avec :

- $m_i$  la masse du segment  $i$ ,
- $G_i$  le centre d'inertie du segment  $i$ ,
- $\bar{\mathbf{I}}_i$  le tenseur d'inertie en  $G_i$ .

En regroupant les termes, on fait apparaître le torseur cinématique :

$$\mathcal{K}_i = \frac{1}{2} [{}^0\mathbf{v}_{i/0}(G_i) \quad {}^0\boldsymbol{\omega}_{i/0}] \cdot \begin{bmatrix} m_i & 0 & 0 & \mathbf{0}_{1 \times 3} \\ 0 & m_i & 0 & \mathbf{0}_{1 \times 3} \\ 0 & 0 & m_i & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \bar{\mathbf{I}}_i \end{bmatrix} \cdot \begin{bmatrix} {}^0\mathbf{v}_{i/0}(G_i) \\ {}^0\boldsymbol{\omega}_{i/0} \end{bmatrix}$$

Or, d'après le modèle cinématique, on a :

$${}^0\mathcal{V}_{i/0}(G) = \begin{bmatrix} {}^0\mathbf{v}_{i/0}(G_i) \\ {}^0\boldsymbol{\omega}_{i/0} \end{bmatrix} = \mathbf{J}_i^c(G_i, \mathbf{q}) \cdot \dot{\mathbf{q}}$$

Donc :

$$\mathcal{K}_i = \frac{1}{2} \dot{\mathbf{q}}^T \underbrace{[\mathbf{J}_i^c(G_i, \mathbf{q})]^T \cdot \begin{bmatrix} m_i & 0 & 0 & \mathbf{0}_{1 \times 3} \\ 0 & m_i & 0 & \mathbf{0}_{1 \times 3} \\ 0 & 0 & m_i & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \bar{\mathbf{I}}_i \end{bmatrix}}_{\mathbf{M}_i(\mathbf{q})} \cdot \mathbf{J}_i^c(G_i, \mathbf{q}) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}_i(\mathbf{q}) \dot{\mathbf{q}}$$

L'énergie cinétique totale du robot s'écrit donc :

$$\mathcal{K} = \sum_{i=1}^n \mathcal{K}_i = \sum_{i=1}^n \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}_i(\mathbf{q}) \dot{\mathbf{q}}$$

Soit sous une forme plus compacte :

$$\mathcal{K} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}$$

avec  $\mathbf{M}(\mathbf{q}) = \sum_{i=1}^n \mathbf{M}_i(\mathbf{q})$ .

$\mathbf{M}_i(\mathbf{q})$  est la matrice d'inertie du robot. Elle est symétrique et définie positive.

### 8.2.3 Calcul de l'énergie potentielle

L'énergie potentielle de pesanteur du segment  $i$  est définie par :

$$\mathcal{P}_i = m_i g \mathbf{z}_0 \cdot \overrightarrow{O_0 G_i}(\mathbf{q})_{/R_0} = m_i g z_{G_i}$$

avec :

- $m_i$  la masse du segment  $i$ ,
- $g$  la constante universelle de gravitation,
- $z_{G_i}$  le centre d'inertie du segment  $i$  dans le repère  $R_0$ .

L'énergie potentielle totale du robot s'écrit donc :

$$\mathcal{P} = \sum_{i=1}^n \mathcal{P}_i(\mathbf{q}) = \sum_{i=1}^n m_i g z_{G_i}$$

### 8.2.4 Calcul du modèle dynamique

Connaissant les énergies cinétiques et potentielles du robot, nous pouvons maintenant définir son Lagrangien, soit :

$$\mathcal{L} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} - \mathcal{P}(\mathbf{q})$$

On calcule alors les différents termes de l'équation d'Euler-Lagrange pour l'articulation  $i$ . On a tout d'abord :

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_i} = [\mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}]_i$$

où  $[\bullet]_i$  désigne la  $i$ ème ligne de la matrice, et donc :

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} = \left[ \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{d\mathbf{M}(\mathbf{q})}{dt} \dot{\mathbf{q}} \right]_i$$

Pour le second terme, on a :

$$\frac{\partial \mathcal{L}}{\partial q_i} = \frac{1}{2} \dot{\mathbf{q}}^T \frac{\partial \mathbf{M}(\mathbf{q})}{\partial q_i} \dot{\mathbf{q}} - \frac{\partial \mathcal{P}(\mathbf{q})}{\partial q_i}$$

Donc, l'équation d'Euler-Lagrange pour l'articulation  $i$  s'écrit :

$$[\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}}]_i + \left[ \frac{d\mathbf{M}(\mathbf{q})}{dt} \dot{\mathbf{q}} \right]_i + \frac{1}{2} \dot{\mathbf{q}}^T \frac{\partial \mathbf{M}(\mathbf{q})}{\partial q_i} \dot{\mathbf{q}} - \frac{\partial \mathcal{P}(\mathbf{q})}{\partial q_i} = \Gamma_i$$

En regroupant toutes les équations sous forme vectorielle, on obtient au final :

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{W}(\mathbf{q}) = \mathbf{\Gamma}$$

Où :

— le terme  $\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}}$  désigne les contributions inertielles, avec,

$$\mathbf{M}(\mathbf{q}) = \sum_{i=1}^n [\mathbf{J}_i^c(G_i, \mathbf{q})]^T \cdot \begin{bmatrix} m_i & 0 & 0 & \mathbf{0}_{1 \times 3} \\ 0 & m_i & 0 & \mathbf{0}_{1 \times 3} \\ 0 & 0 & m_i & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \bar{\mathbf{I}}_i \end{bmatrix} \cdot \mathbf{J}_i^c(G_i, \mathbf{q})$$

— le terme  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$  désigne les contributions centrifuges et de Coriolis<sup>1</sup>,

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{d\mathbf{M}(\mathbf{q})}{dt} - \frac{1}{2} \begin{bmatrix} \dot{\mathbf{q}}^T \cdot \frac{\partial \mathbf{M}(\mathbf{q})}{\partial q_1} \\ \vdots \\ \dot{\mathbf{q}}^T \cdot \frac{\partial \mathbf{M}(\mathbf{q})}{\partial q_n} \end{bmatrix} = \sum_{k=1}^n \frac{\partial \mathbf{M}(\mathbf{q})}{\partial q_k} \dot{q}_k - \frac{1}{2} \begin{bmatrix} \dot{\mathbf{q}}^T \cdot \frac{\partial \mathbf{M}(\mathbf{q})}{\partial q_1} \\ \vdots \\ \dot{\mathbf{q}}^T \cdot \frac{\partial \mathbf{M}(\mathbf{q})}{\partial q_n} \end{bmatrix}$$

— le terme  $\mathbf{W}(\mathbf{q})$  désigne les contributions de la gravité<sup>2</sup>,

$$\mathbf{W}(\mathbf{q}) = \begin{bmatrix} -\frac{\partial \mathcal{P}(\mathbf{q})}{\partial q_1} \\ \vdots \\ -\frac{\partial \mathcal{P}(\mathbf{q})}{\partial q_n} \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} -m_i g \frac{\partial z_{G_i}}{\partial q_1} \\ \dots \\ -m_i g \frac{\partial z_{G_i}}{\partial q_n} \end{bmatrix} = \sum_{i=1}^n [\mathbf{J}^c(G_i, \mathbf{q})]^T \cdot \begin{bmatrix} 0 \\ 0 \\ -m_i g \\ 0 \\ 0 \end{bmatrix}$$

1. La matrice  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  peut se s'écrire sous d'autres formes notamment avec les symboles de Christoffel.

2. Le vecteur  $\mathbf{W}(\mathbf{q})$  peut être également calculé en écrivant le principe des puissances virtuelles pour chaque segment (voir chapitre précédent).

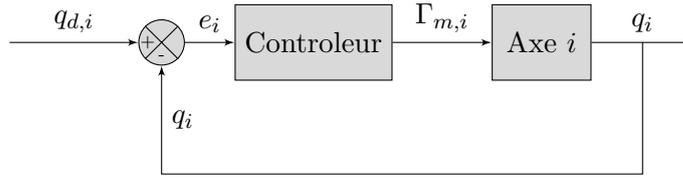


FIGURE 8.1 – Commande découplée

Les efforts articulaires  $\mathbf{\Gamma}$  correspondent aux efforts des moteurs moins les efforts de frottements  $\mathbf{\Gamma}_f$  (secs et visqueux) et les efforts d'interaction  $\mathbf{\Gamma}_e$  entre le robot et son environnement, soit :

$$\mathbf{\Gamma} = \mathbf{\Gamma}_m - \mathbf{\Gamma}_e - \mathbf{\Gamma}_f$$

En prenant en compte les frottements et les interactions avec l'environnement, le modèle dynamique inverse du robot est finalement défini par :

$$\mathbf{\Gamma}_m = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{W}(\mathbf{q}) + \mathbf{\Gamma}_f + \mathbf{\Gamma}_e$$

### 8.3 Commande découplée

La commande des axes a pour objectif d'asservir la position articulaire  $\mathbf{q}(t)$  sur la trajectoire  $\mathbf{q}_d(t)$  calculée par la génération de mouvement. A vitesse réduite, les axes peuvent être asservis séparément en négligeant les effets dynamiques. Pour des applications à grande vitesse, il est nécessaire de les prendre en compte pour ne pas dégrader les performances du robot.

La commande découplée (également appelée commande individuelle) consiste à asservir indépendamment chaque axe sans tenir compte des couplages dynamiques et de l'influence de la configuration sur l'inertie. Le couple moteur de chaque axe est calculé par un contrôleur dédié (généralement de type PID) comme le montre la figure 8.1.

On se place dans la situation où l'on néglige les effets non linéaires, les variations des coefficients en fonction de la configuration et les couplages dynamiques. Par ailleurs, on suppose les effets de la gravité majoritairement compensés par un système mécanique (cf. chapitre 8). Dans ces conditions et en l'absence d'efforts d'interaction avec l'environnement, le comportement dynamique de chaque axe  $i$  du robot peut s'écrire comme suit :

$$a_{ii}\ddot{q}_i = \Gamma_{m_i} - f_i\dot{q}_i - \Gamma_{p_i}$$

où  $f_i$  représente le coefficient de frottement visqueux de l'axe et  $\Gamma_{p_i}$  représente l'effort de perturbation correspondant à tous les effets dynamiques négligés.

Dans le domaine de Laplace, on a donc :

$$\Gamma_{m_i}(p) = (a_{ii}p^2 + f_i p)q_i(p) + \Gamma_{p_i}(p)$$

D'où l'expression de la fonction de transfert de l'axe :

$$q_i(p) = \frac{\Gamma_{m_i}(p) - \Gamma_{p_i}(p)}{(a_{ii}p + f_i)p}$$

A partir de là, il est possible de synthétiser un PID permettant un asservissement convenable de l'axe. A titre d'exemple aisément calculable, si on prend la loi de commande simple suivante, correspondant à un régulateur PD (pour alléger l'écriture, on n'indique plus l'indice  $i$  correspondant à l'axe  $i$  considéré) :

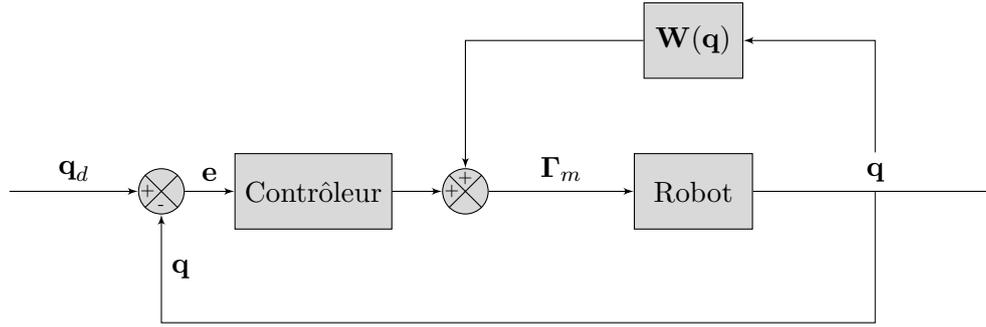


FIGURE 8.2 – Commande avec compensation de la gravité

$$\Gamma_m(t) = K_p e(t) - K_d \dot{q}(t)$$

où  $e(t) = q_d(t) - q(t)$ .

D'où le comportement en boucle fermée :

$$\Gamma_m(p) = K_p q_d(p) - (K_p + K_d p) q(p) = (ap^2 + fp) q(p) + \Gamma_p(p)$$

soit :

$$q(p) = F(p) q_d(p) - \frac{1}{K_p} F(p) \Gamma_p(p)$$

avec :

$$F(p) = \frac{1}{\frac{a}{K_p} p^2 + \frac{K_d + f}{K_p} p + 1}$$

Ceci correspond donc à un système du deuxième ordre dont on peut aisément régler la rapidité et la stabilité (pulsation propre et amortissement) grâce à  $K_p$  et  $K_d$ .

## 8.4 Commande avec compensation de la gravité

Un compromis intéressant entre la commande découplée et la commande prenant en compte tous les effets dynamique présentée dans la section suivante est de compenser uniquement la gravité. Cette commande avec compensation de la gravité néglige les termes inertiels, centrifuges et de Coriolis et n'utilise que le terme  $\mathbf{W}(\mathbf{q})$  du modèle dynamique.

En statique et sans interaction avec l'environnement, les moteurs doivent compenser le poids des segments du robot (et éventuellement de la charge si sa masse est connue) et donc :

$$\Gamma_m = \mathbf{W}(\mathbf{q})$$

On ajoute donc ces couples à la commande calculée par le contrôleur, ce qui donne avec un régulateur PD :

$$\Gamma_{m,i}(t) = K_p e_i(t) - K_d \dot{q}_i(t) + W_i(\mathbf{q})$$

La commande avec compensation de la gravité est représentée par le schéma de la figure 8.2.

## 8.5 Commande à couples calculés

La commande découplée peut générer des différences de comportement importantes en boucle fermée (temps de réponse, stabilité, précision) selon le domaine de l'espace atteignable où le robot évolue et quand le robot se déplace à grande vitesse.

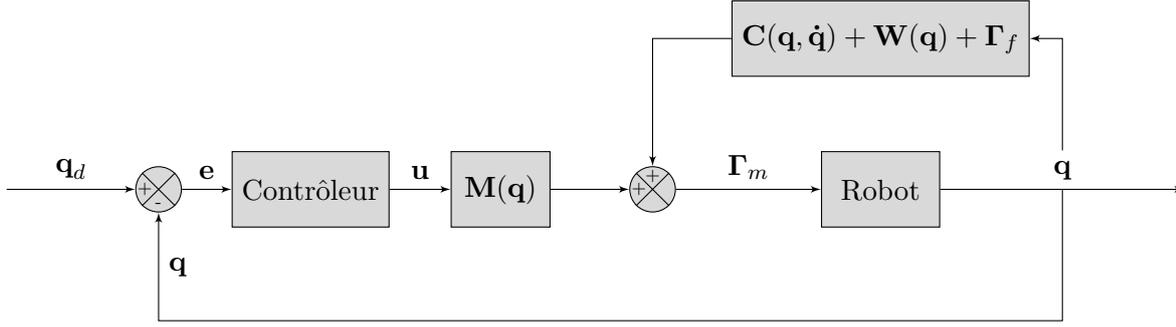


FIGURE 8.3 – Commande à couples calculés

La commande à couples calculés (également appelée commande dynamique) utilise le modèle dynamique inverse du robot afin de compenser tous ces effets.

La commande à couples calculés consiste à imposer aux axes les couples ou forces  $\Gamma_m$  tels que :

$$\Gamma_m = \mathbf{M}(\mathbf{q})\mathbf{u} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{W}(\mathbf{q}) + \Gamma_f$$

avec  $\mathbf{u}$  un vecteur de contrôle. La figure 8.3 illustre cette équation.

Si le modèle correspond « suffisamment » à la réalité du comportement dynamique du robot, alors le robot associé à son modèle dynamique inverse présente le comportement dynamique d'un double intégrateur :

$$\mathbf{u} = \ddot{\mathbf{q}}$$

Ainsi, chaque axe est découplé des autres et linéarisé. Chaque contrôleur d'axe pilote directement un axe en accélération  $u_i = \ddot{q}_i$ .

On peut alors choisir ici un régulateur linéaire classique, comme par exemple ici un régulateur PD :

$$u_i(t) = K_{p_i}e_i(t) - K_{d_i}\dot{q}_i(t)$$

Comme  $u_i = \ddot{q}_i$ , on obtient alors dans le domaine de Laplace :

$$p^2 q_i(p) = K_{p_i} q_{d_i}(p) - (K_{p_i} + p K_{d_i}) q_i(p)$$

soit :

$$q_i(p) = \frac{1}{\frac{1}{K_{p_i}} p^2 + \frac{K_{d_i}}{K_{p_i}} p + 1} q_{d_i}(p)$$

On peut alors régler la dynamique de l'axe  $i$  en jouant sur  $K_{p_i}$  et  $K_{d_i}$ .

## 8.6 Conclusion

Le modèle dynamique d'un robot constitue un outil de dimensionnement afin de choisir les actionneurs adéquats, c'est-à-dire permettant de générer des couples/forces moteurs suffisants pour atteindre les performances en accélération spécifiées par le cahier des charges.

Il permet également de réaliser la commande à couples calculés qui compense efficacement et sans approximation la dynamique totale du robot. Il existe néanmoins d'autres approches de commande (commande prédictive, commande passive, commande adaptative, etc.). Pour plus d'informations, nous renvoyons le lecteur aux ouvrages de référence [3, 13].

## Chapitre 9

# Conception d'une cellule robotisée

Au départ, la décision de mettre en œuvre un système robotisé relève de choix sur les flux de production et leur gestion et de choix économiques, donc correspond plus à un problème de productique qu'à un problème de robotique. Ceci n'est pas évoqué dans ce chapitre, qui résume très succinctement quelques aspects techniques de la conception d'une cellule robotisée.

Du point de vue du robot, la synthèse d'une telle cellule pose principalement quatre types de problèmes :

- le choix du robot,
- le choix du préhenseur,
- le placement du robot dans la cellule,
- le choix et la génération des trajectoires entre les points à atteindre.

### 9.1 Choix du robot

#### 9.1.1 Critères géométriques

Deux critères principaux sont à prendre en compte : le nombre de degrés de liberté de l'effecteur requis par la tâche à réaliser, et le domaine atteignable nécessaire à l'exécution de cette tâche.

D'une manière générale, des théories des mécanismes existent pour obtenir une cinématique de robot permettant de générer les ddl nécessaires et le domaine atteignable requis d'une manière optimisée. Mais ces théories sont souvent très complexes et l'on se tournera en général plutôt vers l'offre des constructeurs.

#### Nombre de degrés de liberté

Il s'agit ici de caractériser le nombre de degrés de liberté dans l'espace dont l'effecteur a besoin pour réaliser convenablement sa ou ses tâches.

Par exemple, un robot prenant des biscuits sur un tapis roulant n'aura en général besoin que de travailler dans le plan du tapis à condition que les biscuits soient à ranger dans des boîtes de même orientation (coplanaire) que le tapis. Ainsi, il n'aura dans ce cas besoin que de 4 ddl pour permettre la prise-dépose : les 3 ddl du plan (a priori horizontal) considéré et un ddl en  $z$  (altitude).

Il s'agit donc bien ici de déterminer les degrés de liberté **de la tâche**, pour adapter le nombre d'axes du robot au besoin (attention à ne pas le sous-estimer ni le sur-estimer).

**NB:** Dans certains cas, où l'environnement est envahi d'obstacles, une redondance peut être intéressante car elle offre une infinité de choix de configurations possibles pour atteindre la situation considérée.

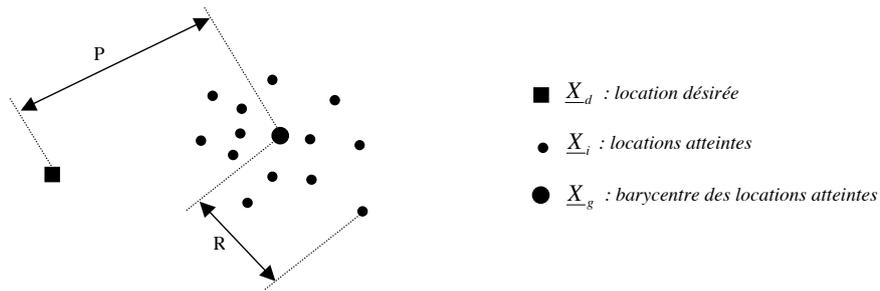


FIGURE 9.1 – Illustration géométrique de la justesse  $P$  et de la répétabilité  $R$ .

### Domaine atteignable

Le domaine atteignable en position est défini par sa forme géométrique et son volume. Il faut donc vérifier que l'ensemble des points à atteindre et/ou des trajectoires à effectuer soit inclus dans un volume pouvant lui-même être inclus dans le domaine atteignable.

Les documentations des constructeurs donnent le domaine atteignable en position, mais en général pour une orientation quelconque. Il ne faut pas oublier que ce domaine se réduit lorsque l'on souhaite par exemple travailler à orientation fixe. On peut donc être amené à représenter soi-même le domaine atteignable. Ceci peut-être très fastidieux « à la main » et se réalise plus aisément à l'aide d'un outil de CAO robotique.

**NB:** En général, le domaine atteignable est donné au centre de la bride sur laquelle on fixera l'organe terminal ou, dans le cas d'un poignet à axes concourants, au point de concours des axes.

Quand l'outil est plus grand que la pièce à travailler, il peut être intéressant de réaliser l'opération avec la pièce portée par le robot. C'est le robot qui amène la pièce à l'outil et plus le robot qui amène l'outil à la pièce.

### Précision

Un robot est dit *précis* (ou exact) si il est à la fois juste et fidèle (bonne répétabilité). Les conditions d'expériences permettant de mesurer la justesse et la répétabilité d'un robot sont définies par la norme ISO9283 [9]. La démarche est la suivante : on définit  $\mathbf{x}_d$  une « situation » cible (position et orientation programmées que l'on souhaite atteindre) :

$$\mathbf{x}_d = [x_d \ y_d \ z_d \ \alpha_d \ \beta_d \ \gamma_d]^T$$

On demande au robot de se positionner  $n$  fois en  $\mathbf{x}_d$ , en suivant toujours la même trajectoire (30 cycles dans la norme). On nomme  $\mathbf{x}_g$  le barycentre de l'ensemble des  $n$  locations effectivement atteintes, notées  $\mathbf{x}_i$  et mesurés par un système de mesure externe.

On définit alors la justesse et la répétabilité comme suit (voir également la figure 9.1).

**Définition 15** (Justesse). *La justesse statique en position d'un robot est définie par l'écart entre la position moyenne obtenue et la position désirée (valeur vraie), c'est-à-dire par la distance  $P$  entre  $\mathbf{x}_d$  et  $\mathbf{x}_g$ , telle que :*

$$P = \sqrt{(x_g - x_d)^2 + (y_g - y_d)^2 + (z_g - z_d)^2}$$

avec :

$$x_g = \frac{1}{n} \sum_{i=1}^n x_i \ ; \ y_g = \frac{1}{n} \sum_{i=1}^n y_i \ ; \ z_g = \frac{1}{n} \sum_{i=1}^n z_i.$$

**Définition 16** (répétabilité). *La répétabilité statique d'un robot (appelée aussi répétabilité) est caractérisée par la dispersion spatiale des positions atteintes :*

$$R = \bar{l} + 3S_l$$

Où :

—  $\bar{l}$  est la distance moyenne au barycentre :

$$\bar{l} = \frac{1}{n} \sum_{i=1}^n l_i$$

—  $l_i$  est la distance entre  $\mathbf{x}_i$  et le barycentre  $\mathbf{x}_g$  :

$$l_i = \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2 + (z_i - z_g)^2} \quad (9.1)$$

—  $S_{n-1}$  est l'estimateur de l'écart-type :

$$S_l = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (l_i - \bar{l})^2}$$

Des définitions similaires existent définissant la justesse et la répétabilité statique en orientation [9].

**NB:** Attention aux faux amis ! En anglais, le terme *precision* désigne souvent la répétabilité. La justesse est traduite par le terme *accuracy*.

Les erreurs de justesse et de répétabilité sont dues aux performances des asservissements, à la résolution des capteurs, aux défauts des organes mécaniques (frottements secs, jeux). En ce qui concerne la justesse, une grande partie de sa dégradation peut venir des erreurs de modèle pour le transformateur de coordonnées  $\mathbf{x} \rightarrow \mathbf{q}$  (en particulier, les valeurs des paramètres géométriques du modèle ne correspondent pas exactement à celles du robot).

La répétabilité est beaucoup plus importante que la justesse, notamment lorsque le robot est programmé par apprentissage. Des problèmes liés à la précision peuvent cependant apparaître lorsqu'on utilise la programmation hors ligne (la CAO considère les asservissements parfaits : consigne = mesure).

**NB:** La répétabilité est toujours indiquée dans les documentations constructeurs (souvent sous le terme répétabilité), mais rarement la justesse. Pour exemple, la répétabilité du STAÜBLI RX90 est 20  $\mu\text{m}$ .

### 9.1.2 Critères cinématiques

Comme pour toute machine, le temps de cycle permis pour la tâche robotisée est un critère fondamental du cahier des charges. Il est donc important de prévoir si le robot choisi pourra respecter ce temps de cycle. Pour cela, il est nécessaire de connaître les vitesses maximales (instantanées et de rotation), voire les accélérations maximales que l'effecteur peut atteindre. Les constructeurs fournissent en général quelques informations sur ce point.

Il ne faut cependant pas oublier qu'un robot n'est pas un système linéaire et qu'à vitesses articulaires constantes, les vitesses opérationnelles varient en fonction de la configuration ( $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}}$ ). Là encore un logiciel de CAO robotique peut être d'un grand secours.

### 9.1.3 Critères dynamiques

Les efforts extérieurs appliqués sur un robot sont en général issus du poids du préhenseur choisi et de celui de la charge transportée (il est d'ailleurs souvent préférable de choisir ou concevoir le préhenseur avant de choisir le robot, afin de prendre correctement en compte le critère de poids du préhenseur dans le choix du robot). Dans les cas de commande en effort, il faut également prendre en compte l'effort que le robot doit appliquer sur l'environnement. Les documentations constructeur précisent systématiquement la charge maximale admissible — ne pas oublier de compter l'effecteur dans cette charge! — sans toutefois donner les six composantes du torseur des efforts (forces et couples).

On peut également trouver dans ces documentations les positions, vitesses et accélérations à ne pas dépasser en fonction de la charge déplacée par le robot, de manière à ne pas endommager la mécanique et la motorisation.

## 9.2 Choix du préhenseur ou de l'outil

Il existe une large gamme de préhenseurs et d'outils génériques adaptés à la plupart des tâches (doigts de serrage, ventouses à vide, pinces de soudage, broches, etc.). Mais pour certaines applications, il pourra être nécessaire de faire réaliser un préhenseur ou un outil spécifique.

Dans tous les cas, il faut toujours essayer de choisir une solution la plus simple possible. Pour ce faire, on peut envisager des positions et orientations judicieuses de pièces à prendre, des posages et des aménagements appropriés (à retournement automatique des pièces par exemples).

**NB:** Les fabricants de préhenseur ne sont en général pas les mêmes que les fabricants de robots.

## 9.3 Placement du robot

Une fois le robot choisi, il s'agit à présent de le placer sur le site. Le placement d'un robot consiste à déterminer la position de sa base par rapport à la tâche à réaliser dans la cellule. D'une manière générale, trouver un placement « optimal » du robot est un problème délicat. Une CAO robotique peut également rendre de grands services sur ce point.

### 9.3.1 Problème géométrique posé

Soient  $p$  situations (positions et orientations) que le centre de l'effecteur du robot devra atteindre, données dans un repère lié au site robotisé,  $R_{site}$ .

On décrit les  $p$  situations  $S_k$  par  $p$  matrices  ${}^{site}\mathbf{T}_{S_k}$ . Il s'agit là du cahier des charges minimal correspondant au problème du placement du robot.

Par ailleurs, le robot étant préalablement choisi, son modèle géométrique direct  ${}^0\mathbf{T}_P$  donne les positions et orientations du centre de l'effecteur par rapport à un repère lié à sa base,  $R_0$ .

Le problème est alors de positionner  $R_0$  dans  $R_{site}$  de manière judicieuse, c'est-à-dire en optimisant un (ou des) critère(s) choisi(s).

### 9.3.2 Critères

Les critères à optimiser sont variés et dépendent de la tâche à réaliser. On peut par exemple chercher à faire en sorte que le placement proposé soit tel que pour les  $p$  situations à atteindre  $S_k$ , les débattements soient les plus faibles possibles (gain de temps de cycle) et les variables articulaires correspondantes soient les plus éloignées possibles de leurs valeurs limites (lorsqu'un axe s'approche d'une de ses butées, il peut générer des problèmes de retournement pour atteindre le point suivant).

On peut également souhaiter s'éloigner ou se rapprocher au maximum, sur certains points de la trajectoire, des configurations singulières du robot.

### 9.3.3 Contraintes

Le robot doit naturellement être placé de manière à ce que le centre de l'effecteur puisse atteindre les  $p$  situations  $\mathbf{S}_k$  qu'il doit atteindre pour réaliser la tâche. En d'autres termes, il doit exister pour tout  $k$  une solution à l'équation :

$${}^{site}\mathbf{T}_{S_k} = {}^{site}\mathbf{T}_0 {}^0\mathbf{T}_{S_k}$$

La base du robot ne peut être déplacée que dans un domaine limité, compatible avec l'espace disponible au sol mais surtout prenant en compte les recommandations éventuelles du constructeur quant aux orientations permises de la base du robot. Ces considérations contraignent la matrice  ${}^{site}\mathbf{T}_0$  qui doit appartenir au domaine d'installation.

En résumé, on cherche à déterminer la transformation  ${}^{site}\mathbf{T}_0$  qui optimise un critère défini sur les situations  $\mathbf{S}_k$  et respectant les contraintes énoncées ci-dessus. Ceci ne peut quasiment se résoudre qu'à l'aide d'une CAO robotique appropriée.

## 9.4 Choix des trajectoires

Un autre type de problème courant est un problème combinatoire. Lorsque par exemple le robot doit atteindre  $p$  situations  $\mathbf{S}_k$  sans ordre préférentiel (par exemple pour le soudage par points ou la prise-dépose), il s'agit de trouver l'ordre des trajectoires générant un temps de cycle minimum.

On est donc ramené à un problème de voyageur de commerce qui ne peut se résoudre que par heuristique : il y a en effet une explosion combinatoire, le nombre de circuits possibles est égal à  $p!$ . Ici encore, l'utilisation de l'informatique est indispensable.

Enfin, pour certaines tâches de manipulation complexes dans des environnements encombrés, il n'est pas toujours facile de trouver une trajectoire évitant les collisions. On fait alors appel à des logiciels de planification qui recherchent une solution possible à l'aide de méthodes probabilistes (la résolution analytique étant ici aussi trop combinatoire).



# Bibliographie

- [1] Olivier Company and Sébastien Briot. Les robots parallèles. *Techniques de l'ingénieur*, S7768 :1–29, 2015. 47
- [2] Max Giordano and Jacques Lottin. *Cours de robotique : Description et fonctionnement des robots industriels*. Armand Colin, 1997. 7, 8, 10
- [3] Wisama Khalil and Etienne Dombre. *Modélisation, identification et commande des robots*. Hermes Science, 1999. 34, 47, 66
- [4] Wisama Khalil and Etienne Dombre. *Modeling, identification & control of robots*. Kogan Page Science, 2004.
- [5] J.P. Lallemand and S. Zeghloul. *Robotique : Aspects Fondamentaux*. Masson, 1994.
- [6] Alain Liégeois. Modélisation et commande des robots manipulateurs. *Techniques de l'ingénieur*, S7730 :1–16, 2000.
- [7] Jean-Pierre Merlet. *Parallel Robots*. Springer, 2006. 47
- [8] International Standard Organization. *ISO 8373 Robotique — Vocabulaire*. 6
- [9] International Standard Organization, editor. *ISO 9283 Manipulating industrial robots - Performance criteria and related test methods*. 1998. 68, 69
- [10] Morvan Ouisse and Sylvaine Mallet. *Mécanique générale : Cinématique et dynamique des mécanismes*. Hermes Science, 2010. 61
- [11] Richard P. Paul. *Robot Manipulators : Mathematics, Programming, and Control*. The MIT press, 1981. 18, 19, 33
- [12] Stéphane Régnier and Nicolas Chaillet, editors. *La microrobotique : Applications à la micromanipulation*. Hermes Science, 2008. 11
- [13] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2005. 41, 66



## Annexe A

# Sites web, blogs, réseaux d'informations et salons industriels

Dans le but de vous permettre de compléter votre formation et de vous aider dans la recherche d'un stage ou d'un emploi, cette annexe propose une liste de sites web, blogs, réseaux d'informations et salons industriels portant sur la robotique.

### **Automatica**

Adresse : <https://www.automatica-munich.com/>

Description : salon industriel à Munich

### **Collège de France**

Adresse : <https://www.college-de-france.fr/fr/chaire/jean-paul-laumond-innovation-techno>

Description : vidéos des cours donnés au collège de France par J.-P. Laumond et de nombreuses personnalités invitées

### **Cours de troisième année de l'ENSPS et du Master M2 ISTI**

Adresse : <https://www.youtube.com/watch?v=2YH6KDJAuN8>

Description : Cours de robotique de J. Gangloff (vidéos, dispos, exercices, sujets d'examen)

### **Robotics worldwide**

Adresse : <https://www.robotics-worldwide.org/>

Description : liste de diffusion internationale relayant notamment des offres de thèses et d'emplois en robotique dans le monde

### **Groupe de Recherche (GdR) en Robotique**

Adresse : <https://www.gdr-robotique.org/>

Description : réseau français d'acteurs académiques (financé par le CNRS), liste des entreprises et des laboratoires français de robotique (inscription gratuite obligatoire pour y accéder), liste de diffusion, offres de stages et d'emplois

### **IEEE Robotics & Automation Society**

Adresse : <https://www.ieee-ras.org/>

Description : réseau international d'acteurs académiques, liste de diffusion (inscription payante)

### **International Federation of Robotics**

Adresse : <https://ifr.org/>

Description : fédération internationale d'acteurs industriels et académiques

**euRobotics**

Adresse : <https://eu-robotics.net/>

Description : réseau européen d'acteurs industriels et académiques (inscription payante)

**Robohub**

Adresse : <https://robohub.org/>

Description : réseau international d'acteurs industriels et académiques, liste de diffusion

**ROBOT Start PME**

Adresse : <http://www.robotstartpme.fr/>

Description : programme de soutien aux PME primo-accédantes à la robotisation mis en place en 2013

**The Robot Report**

Adresse : <http://www.therobotreport.com/>

Description : site d'actualités sur la robotique industrielle et la robotique de service

## Annexe B

# Logiciels et bibliothèques libres

Cette annexe propose une sélection de quelques logiciels et bibliothèques libres permettant de simuler ou de commander des robots.

### B.1 Logiciels

#### CoppeliaSim (ex V-REP)

Adresse : <https://www.coppeliarobotics.com/>

Description : simulateur de robots articulés sériels/parallèles et de robots mobiles

Distributions : Windows, Linux, Mac OS

#### Gazebo

Adresse : <https://gazebo.org/>

Description : simulateur de robots articulés sériels/parallèles et de robots mobiles

Distribution : Linux

#### OpenSYMORO

Adresse : <https://github.com/symoro/symoro>

Description : logiciel de calcul symbolique des différents modèles (géométriques, cinématiques et dynamiques) de robots articulés sériels/parallèles

Distribution : Windows, Linux, Mac OS (développé en Python)

### B.2 Bibliothèques

#### Robotics Toolbox (de Peter Corke)

Adresse : <https://petercorke.com/toolboxes/robotics-toolbox/>

Description : bibliothèque permettant de modéliser des robots articulés sériels

Contenu : transformations homogènes, quaternions, DH, modélisation géométrique, modélisation cinématique, modélisation dynamique, génération de trajectoire et planification

Langage : Matlab

**Robotics Library (RL)**

Adresse : <https://www.roboticslibrary.org/>

Description : bibliothèque permettant de modéliser des robots articulés sériels

Contenu : transformations homogènes, quaternions, DH, modélisation géométrique, modélisation cinématique, modélisation dynamique, génération de trajectoire, planification, détection de collision, affichage 3D via la bibliothèque COIN.

Langage : C++

Distributions : Windows, Linux, Mac OS

**Robot Operating System (ROS)**

Adresse : <https://www.ros.org/>,

Description : ROS permet d'interfacer un grand nombre de robots et de capteurs en vue de leur commande. ROS utilise la bibliothèque de traitement d'image OpenCV pour la vision.

Langage : C++

Distribution : Linux

**SOFA**

Adresse : <https://www.sofa-framework.org/>,

Description : SOFA est une bibliothèque permettant la simulation et la commande de robots déformables

Langages : C++, python

Distribution : Linux, Windows, Mac OS

**Visual Servoing Platform (ViSP)**

Adresse : <https://visp.inria.fr/>,

Description : ViSP est une bibliothèque dédiée à l'asservissement visuel.

Langage : C++

Distributions : Windows, Linux, Mac OS

**Annexe C**

**Travaux dirigés**

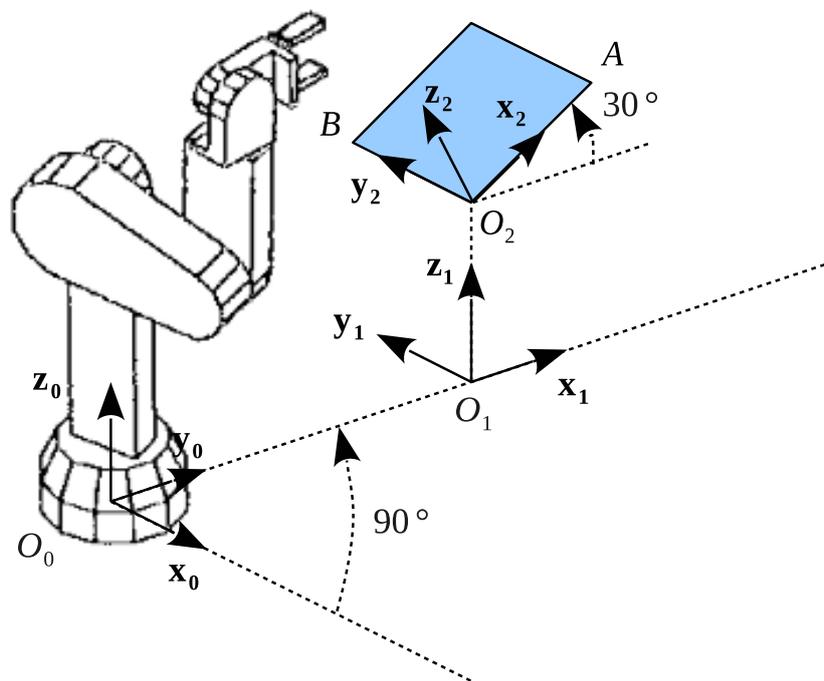
---

## Robotique industrielle

### TD n°1 : Repérage et transformations géométriques

---

Un robot doit aller saisir des pièces sur un plateau de stockage incliné. Le plateau est un carré de 0.2 mètre de côté. Le coin du plateau est situé à 1 mètre de la base du robot, à 0.5 mètre de hauteur. Le plateau est tourné de  $90^\circ$  par rapport à l'axe  $z_0$  et de  $30^\circ$  par rapport à l'horizontal.



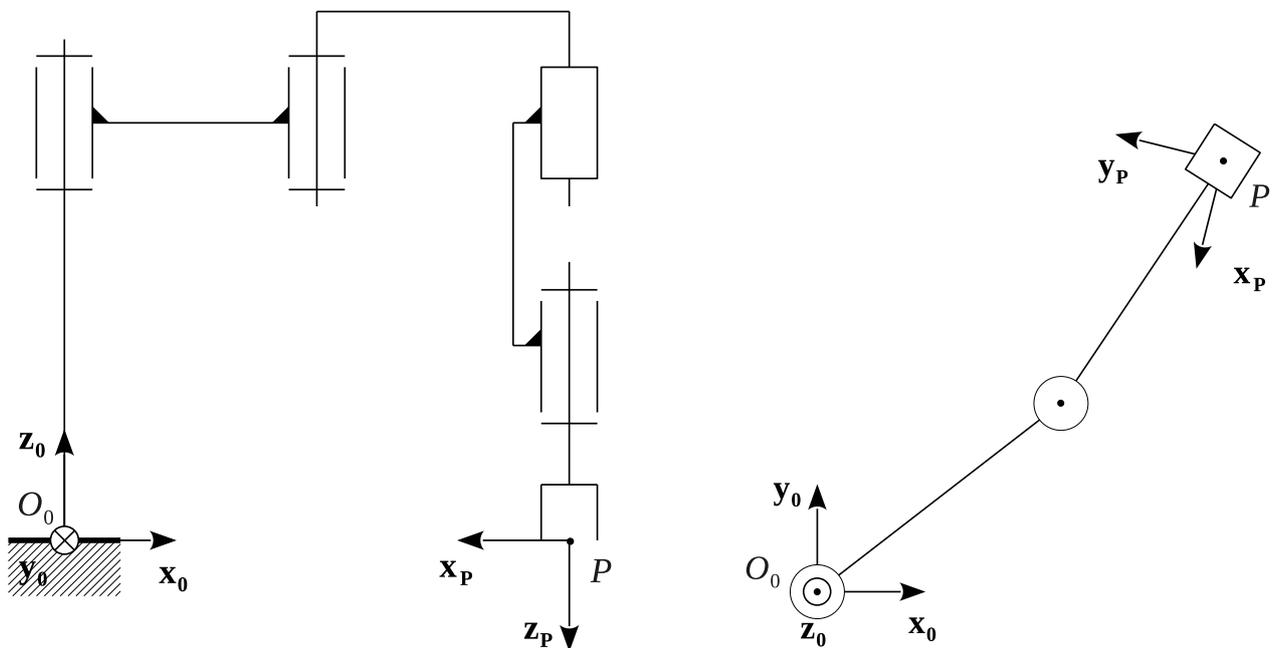
1. Calculer la matrice de rotation permettant de passer du repère  $R_0$  au repère  $R_2$  en utilisant la composition des rotations.
2. Recalculer la matrice de rotation permettant de passer du repère  $R_0$  au repère  $R_2$  en utilisant les angles nautiques.
3. Soit  $P$  un point du plateau de coordonnées  $(0.2; 0.1; 0)$  dans le repère  $R_2$ . Calculer les coordonnées de  $P$  dans le repère  $R_0$ .
4. Refaire les calculs en utilisant les matrices de transformation homogène.

## Robotique industrielle

### TD n°2 : Modélisation géométrique directe

**Exercice 1. Modèle géométrique direct d'un robot SCARA**

Soit le robot série 4 axes RRPR décrit par la figure ci-dessous. Ce robot est représenté à gauche dans la configuration où les coordonnées articulaires  $q_1, q_2, q_3$  et  $q_4$  sont nulles et à droite dans une configuration quelconque (et vu du haut).



1.a. Placer les repères  $R_1, R_2, R_3$  et  $R_4$  sur le schéma.

1.b. Calculer  ${}^0\mathbf{T}_P$

1.c. Vérifier la validité de vos calculs sur quelques configurations particulières.

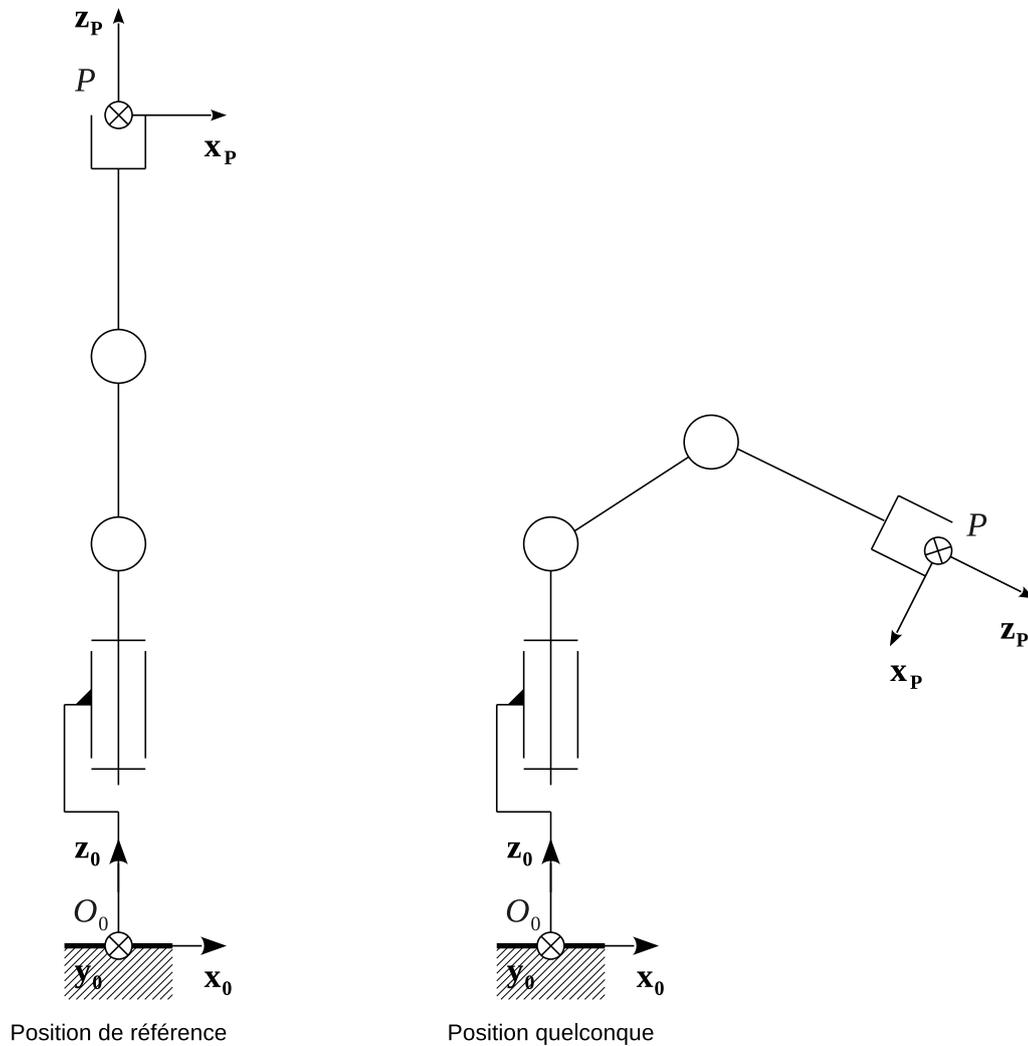
1.d. Exprimer l'orientation de l'organe terminal à l'aide des angles nautiques.

1.e. Représenter la projection du domaine atteignable  $D_P$  de ce robot dans le plan  $O_0\vec{x}_0\vec{y}_0$  sachant que les butées articulaires sont les suivantes :

$$q_1 \in \left[-\frac{2\pi}{3}, \frac{2\pi}{3}\right] ; \quad q_2 \in \left[-\frac{2\pi}{3}, \frac{2\pi}{3}\right]$$

**Exercice 2. Modèle géométrique direct d'un porteur anthropomorphe**

Soit le robot série 3 axes RRR décrit par la figure ci-dessous. Ce robot est complètement tendu à la verticale quand les coordonnées articulaires  $q_1, q_2$  et  $q_3$  sont nulles.



2.a. Placer les repères  $R_1, R_2$  et  $R_3$  sur le schéma.

2.b. Calculer la matrice de transformation permettant de passer du repère  $R_0$  au repère  $R_P$ .

2.c. Vérifier la validité de vos calculs sur quelques configurations particulières.

2.d. Exprimer l'orientation de l'organe terminal à l'aide des angles nautiques.

2.e. Représenter la coupe du domaine atteignable  $D_P$  de ce robot dans le plan  $O_0\vec{x}_0\vec{z}_0$  sachant que les butées articulaires sont les suivantes :

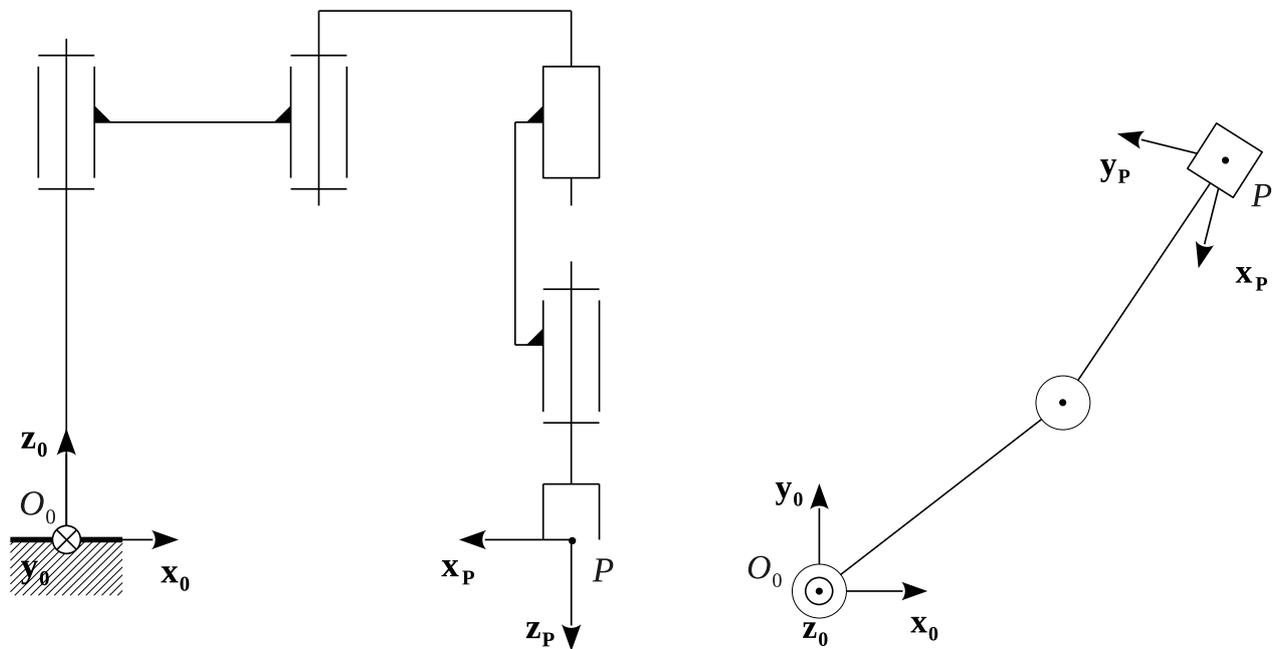
$$q_1 \in [-\pi, \pi] ; q_2 \in [-\pi/2, \pi/2] ; q_3 \in [-\pi/2, \pi/2]$$

## Robotique industrielle

### TD n°3 : Modélisation géométrique inverse

#### Exercice 1. Modèle géométrique inverse d'un robot SCARA

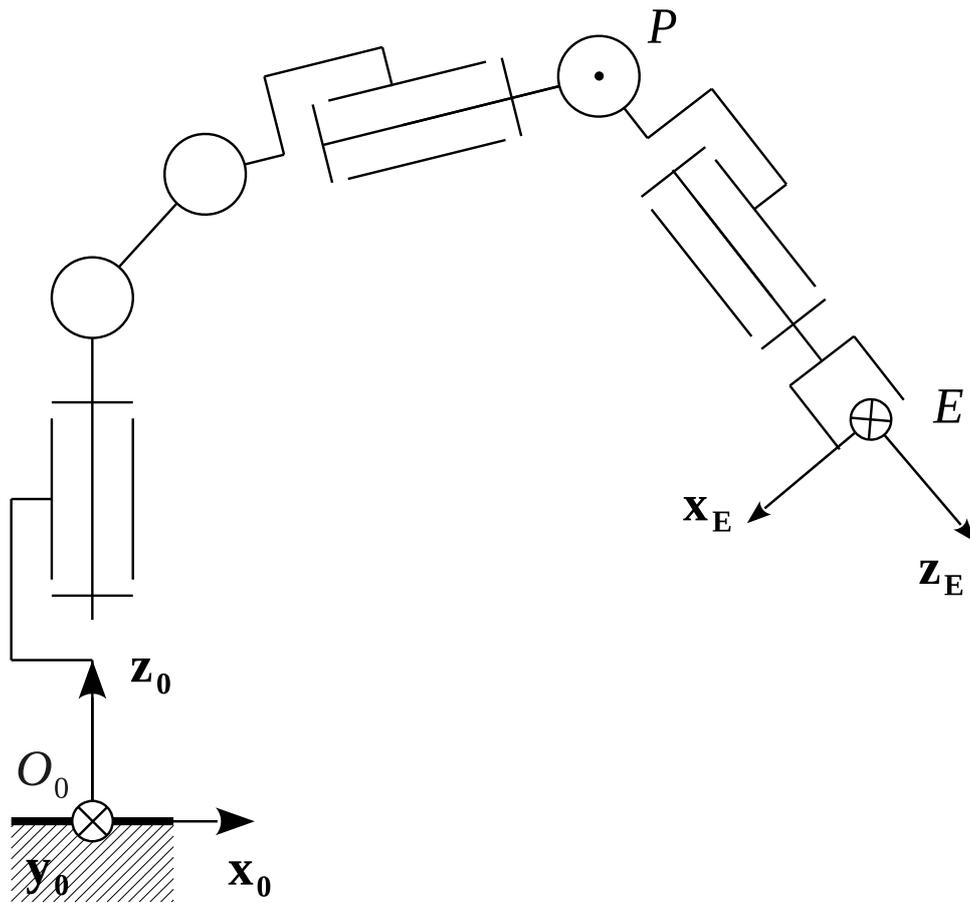
Le but de l'exercice est de calculer le modèle géométrique inverse du robot série 4 axes RRPR décrit dans le TD précédent. On cherche donc à exprimer  $q_1, q_2, q_3$  et  $q_4$  en fonction d'une position et d'une orientation désirées. La position désirée est définie par les coordonnées  $P_x, P_y$  et  $P_z$  du point  $P$ . L'orientation désirée est définie par l'angle de lacet  $\alpha$ .



1.a. Calculer  $q_1, q_2, q_3$  et  $q_4$  en fonction de  $P_x, P_y, P_z$  et de l'angle  $\alpha$  à l'aide de la méthode trigonométrique.

**Exercice 2. Modèle géométrique inverse d'un robot anthropomorphe**

Soit le robot série 6 axes RRRRRR décrit par la figure ci-dessous. Ce robot est complètement tendu à la verticale quand les coordonnées articulaires sont nulles. Le but de l'exercice est de calculer le modèle géométrique inverse du robot. On cherche donc à exprimer  $q_1, q_2, q_3, q_4, q_5$  et  $q_6$  en fonction d'une position et d'une orientation désirées. La position désirée est définie par les coordonnées  $E_x, E_y$  et  $E_z$  du point  $E$ . L'orientation désirée est définie par les angles nautiques  $\alpha, \beta$  et  $\gamma$ . Le point  $P$  désigne le centre du poignet sphérique.



On s'intéresse pour commencer au porteur.

2.a. Calculez  $q_1, q_2$  et  $q_3$  en fonction de  $P_x, P_y, P_z$  à l'aide de la méthode trigonométrique ou de la méthode de Paul.

2.b. Calculez  $P_x, P_y$  et  $P_z$  en fonction de  $P_x, P_y, P_z, \alpha, \beta$  et  $\gamma$ .

On s'intéresse maintenant au poignet.

2.c. Calculez  ${}^3M_6$  en fonction de  $q_4, q_5$  et  $q_6$  (MGD du poignet).

2.d. Calculez  $q_4, q_5$  et  $q_6$  en fonction de  $q_1, q_2, q_3, \alpha, \beta, \gamma$  par identification des termes des matrices.

---

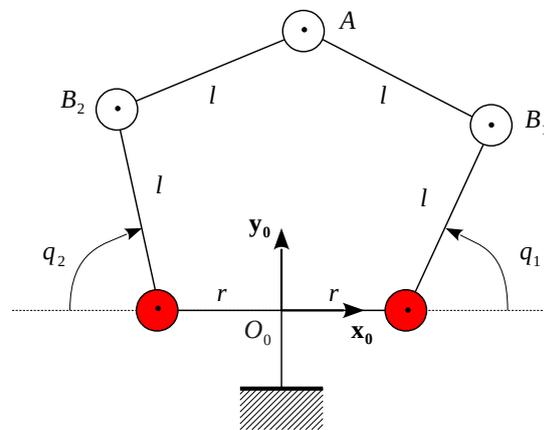
## Robotique industrielle

### TD n°4 : Modélisation des robots parallèles

---

#### Exercice 1. Modélisation du robot Mitsubishi RP 3AH

On souhaite calculer les modèles géométriques et cinématique du robot parallèle de prise-dépose Mitsubishi RP 3AH présenté ci-dessous.



On note  $[x; y]^T$  les coordonnées du point  $A$  dans le repère  $R_0$ .

1.a. Exprimer les coordonnées des points  $B_1$  et  $B_2$  dans le repère  $R_0$  en fonction de  $q_1$ ,  $q_2$ ,  $r$  et  $l$ .

1.b. Écrire les équations de fermeture.

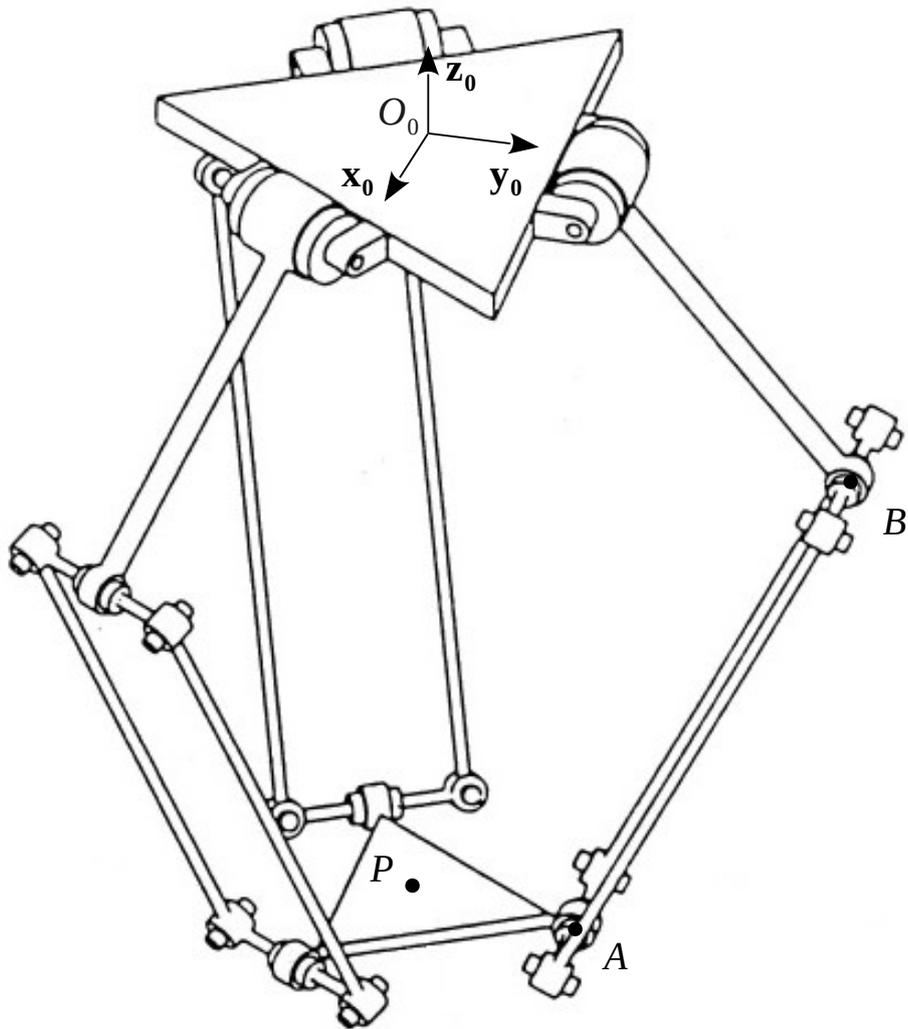
1.c. En déduire les modèles géométriques directe et inverse.

1.d. Dériver les équations de fermeture par rapport au temps et en déduire le modèle cinématique.

1.e. Identifier les configurations singulières du robot.

**Exercice 2. Modélisation d'un robot delta**

Soit le robot Delta 3-RUU décrit par la figure ci-dessous.



2.a. Exprimer les coordonnées du point A dans le repère  $R_0$  en fonction des coordonnées  $[x; y; z]^T$  du point P dans le repère  $R_0$ .

2.b. Exprimer les coordonnées du point B dans le repère  $R_0$  en fonction de  $q_1$ .

2.c. Écrire les équations de fermeture.

2.d. En déduire les systèmes d'équations à résoudre pour obtenir les modèles géométriques directe et inverse.

---

---

## Robotique industrielle

### TD n°5 : Modélisation cinématique

---

#### Exercice 1. Modèle cinématique analytique d'un robot SCARA

Soit le robot série 4 axes RRPR décrit dans l'exercice 1 du TD n°2. Son modèle géométrique direct exprimé avec les angles nautiques est :

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \alpha \end{bmatrix} = \begin{bmatrix} l_2 c_{1+2} + l_1 c_1 \\ l_2 s_{1+2} + l_1 s_1 \\ l_0 - l_3 - l_4 + q_3 \\ q_1 + q_2 + q_4 + \pi \end{bmatrix}$$

Les dimensions du robot sont :  $l_0 = 1$  m,  $l_1 = 0.5$  m,  $l_2 = 0.5$  m,  $l_3 = 0$  m,  $l_4 = 0.3$  m.

1.a. Calculer la matrice jacobienne analytique  ${}^0\mathbf{J}^a(\mathbf{q})$  du robot liant les vitesses opérationnelles  $\dot{x}$ ,  $\dot{y}$ ,  $\dot{z}$ ,  $\dot{\alpha}$  du point  $P$  aux vitesses articulaires  $\dot{q}_1$ ,  $\dot{q}_2$ ,  $\dot{q}_3$ ,  $\dot{q}_4$  et telle que :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \end{bmatrix} = {}^0\mathbf{J}^a(\mathbf{q}) \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix}$$

1.b. Calculer les vitesses minimales et maximales du point  $P$  dans l'espace opérationnel pour la configuration  $\mathbf{q} = [0 \ 2\pi/3 \ 0 \ 0]^T$  sachant que les vitesses maximales des axes dans les deux sens sont les suivantes :

$$\begin{cases} \dot{q}_{1,max} = 2\pi \text{ rad/s} \\ \dot{q}_{2,max} = 2\pi \text{ rad/s} \\ \dot{q}_{3,max} = 0.5 \text{ m/s} \\ \dot{q}_{4,max} = 4\pi \text{ rad/s} \end{cases}$$

1.c. Calculer la manipulabilité en ce point.

1.d. Établir les différentes postures et configurations singulières du robot.

**Exercice 2. Modèle cinématique en utilisant les torseurs**

On rappelle les matrices de rotations entre deux repères calculées au TD n°2 :

$${}^0\mathbf{M}_1 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^1\mathbf{M}_2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^2\mathbf{M}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^3\mathbf{M}_4 = \begin{bmatrix} c_4 & -s_4 & 0 \\ s_4 & c_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^4\mathbf{M}_P = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

On rappelle également :

$${}^0\mathbf{M}_2 = \begin{bmatrix} c_{1+2} & -s_{1+2} & 0 \\ s_{1+2} & c_{1+2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^0\mathbf{M}_3 = \begin{bmatrix} c_{1+2} & -s_{1+2} & 0 \\ s_{1+2} & c_{1+2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^0\mathbf{M}_4 = \begin{bmatrix} c_{1+2+4} & -s_{1+2+4} & 0 \\ s_{1+2+4} & c_{1+2+4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^0\mathbf{M}_P = \begin{bmatrix} -c_{1+2+4} & -s_{1+2+4} & 0 \\ -s_{1+2+4} & c_{1+2+4} & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Enfin, pour gagner du temps, on a aussi calculé :

$$\overrightarrow{O_4P}_{/R_4} = \begin{bmatrix} 0 \\ 0 \\ -l_4 \end{bmatrix} \quad \overrightarrow{O_3P}_{/R_3} = \begin{bmatrix} 0 \\ 0 \\ -l_3 - l_4 \end{bmatrix} \quad \overrightarrow{O_2P}_{/R_2} = \begin{bmatrix} l_2 \\ 0 \\ q_3 - l_3 - l_4 \end{bmatrix} \quad \overrightarrow{O_1P}_{/R_1} = \begin{bmatrix} l_1 + l_2 c_2 \\ l_2 s_2 \\ q_3 - l_3 - l_4 \end{bmatrix}$$

2.a. Écrire les torseurs cinématiques  ${}^{i-1}\mathcal{V}_{i/i-1}(P)$  de chaque articulation.

2.b. Calculer le torseur cinématique  ${}^0\mathcal{V}_{P/0}(P)$  décrivant la vitesse de  $P$  dans  $R_0$ .

2.c. En déduire la matrice jacobienne cinématique. Que remarque-t'on quand on la compare à la matrice jacobienne analytique obtenue dans l'exercice 1 ?

On souhaite que le robot exerce une force de 50 N dans la direction de  $\vec{x}_0$  et sachant que  $q_1 = 0$  et  $q_2 = \pi/2$ .

2.d. Calculer les couples moteurs nécessaires pour exercer cet effort.

## Robotique industrielle

### TD n°6 : Génération de mouvement

Soit le robot série 4 axes RRPR décrit dans le TD n°2. Son modèle géométrique direct exprimé avec les angles nautiques est :

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \alpha \end{bmatrix} = \begin{bmatrix} l_2 c_{1+2} + l_1 c_1 \\ l_2 s_{1+2} + l_1 s_1 \\ l_0 - l_3 - l_4 + q_3 \\ q_1 + q_2 + q_4 \end{bmatrix}$$

Les dimensions du robot sont :  $l_0 = 1$  m,  $l_1 = 0.5$  m,  $l_2 = 0.5$  m,  $l_3 = 0$  m,  $l_4 = 0.3$  m.

Son modèle géométrique inverse est (coude à droite) :

$$\begin{cases} q_1 = \text{atan2}(y, x) - \arccos\left(\frac{x^2+y^2+l_1^2-l_2^2}{2l_1\sqrt{x^2+y^2}}\right) \\ q_2 = \arccos\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right) \\ q_3 = z - l_0 + l_3 + l_4 \\ q_4 = \alpha - q_1 - q_2 \end{cases}$$

Les vitesses maximales des axes dans les deux sens sont les suivantes :

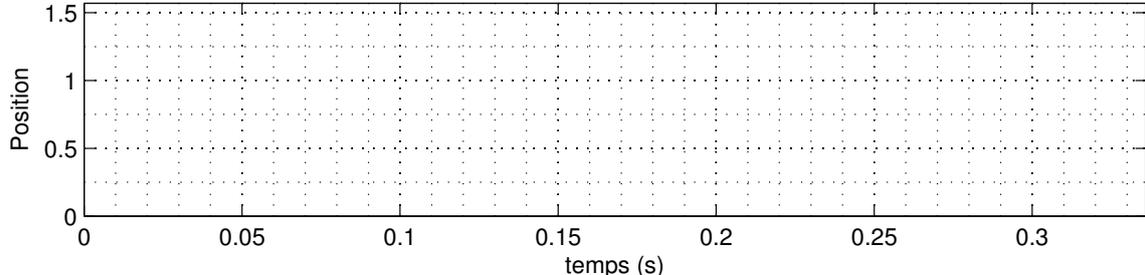
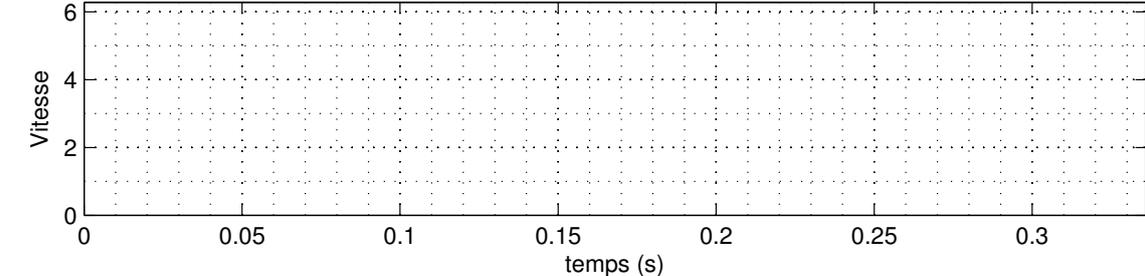
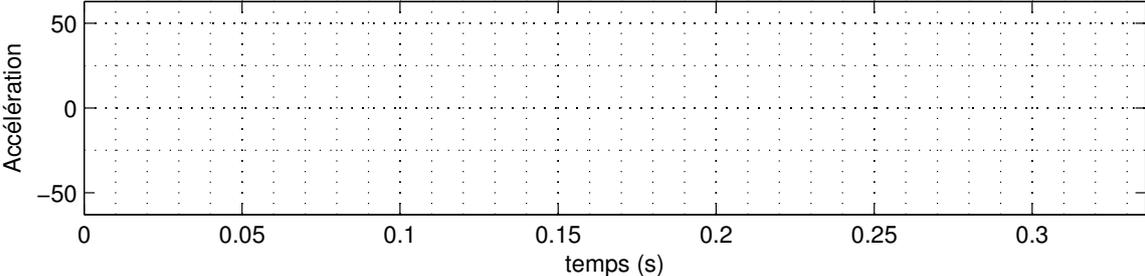
$$\begin{cases} \dot{q}_{1,\max} = 2\pi \text{ rad/s} \\ \dot{q}_{2,\max} = 2\pi \text{ rad/s} \\ \dot{q}_{3,\max} = 0.5 \text{ m/s} \\ \dot{q}_{4,\max} = 4\pi \text{ rad/s} \end{cases}$$

Les vitesses maximales des axes dans les deux sens sont les suivantes :

$$\begin{cases} \ddot{q}_{1,\max} = 20\pi \text{ rad/s}^2 \\ \ddot{q}_{2,\max} = 20\pi \text{ rad/s}^2 \\ \ddot{q}_{3,\max} = 5 \text{ m/s}^2 \\ \ddot{q}_{4,\max} = 40\pi \text{ rad/s}^2 \end{cases}$$

1. Calculer et tracer les profils position/vitesse/accélération des axes 1 et 2 permettant de passer du point d'arrêt  $A = (0.7; 0.5; 0.7; 0)$  au point d'arrêt  $B = (-0.5; 0.5; 0.7; 0)$  dans l'espace articulaire en synchronisant les axes avec des lois homothétiques (les coordonnées sont données en mètres).

2. Établir la loi horaire d'une trajectoire rectiligne dans l'espace opérationnel à vitesse constante  $v = 0,1$  m/s allant de A à B.



---

