

Université de Bourgogne Franche-Comté – UFC – ENSMM

Robotique industrielle

Guillaume Laurent et Nicolas Chaillet

2017

Table des matières

1	Introduction	5
1.1	Définitions	5
1.2	Composition d'un robot industriel	6
1.3	Configurations usuelles des robots industriels	8
1.4	Les marchés de la robotique	12
1.5	Conclusion	15
2	Repérage d'un solide dans l'espace	17
2.1	Représentation d'un point	17
2.2	Position	17
2.3	Orientation déterminée par une matrice	18
2.4	Orientation déterminée par trois angles	21
2.5	Orientation déterminée par un vecteur et un angle	23
2.6	Orientation déterminée par les quaternions	24
2.7	Transformation homogène	24
2.8	Conclusion	26
3	Modèles géométriques direct et inverse	27
3.1	Modélisation géométrique directe d'un robot	27
3.2	Domaine atteignable	34
3.3	Modèle géométrique inverse d'un robot	35
3.4	Conclusion	39
4	Modèles cinématiques direct et inverse	41
4.1	Définitions	41
4.2	Jacobienne analytique	42
4.3	Jacobienne cinématique	42
4.4	Passage d'une jacobienne à l'autre	46
4.5	Vitesses atteignables et manipulabilité	46
4.6	Aspects et configurations singulières	47
4.7	Conclusion	51
5	Efforts et modélisation dynamique	53
5.1	Définition	53
5.2	Modèle dynamique inverse	53
5.3	Efforts appliqués par le robot sur son environnement	56
5.4	Efforts dus à la pesanteur et équilibrage statique	56
5.5	Conclusion	58

6	Génération de mouvement	59
6.1	Programmation d'un robot	59
6.2	Génération de mouvement dans l'espace articulaire	60
6.3	Génération de mouvement dans l'espace opérationnel	65
6.4	Conclusion	69
7	Commande des axes (dans l'espace articulaire)	71
7.1	Chaîne de commande d'un robot industriel	71
7.2	Commande individuelle des axes	72
7.3	Commande à couples calculés	73
7.4	Conclusion	74
8	Choix d'un robot	75
8.1	Choix du robot	75
8.2	Choix du préhenseur ou de l'outil	78
8.3	Placement du robot	78
8.4	Choix des trajectoires	79
9	Ce qui n'a pas été vu ...	81
9.1	Planification de trajectoires	81
9.2	Étude des robots parallèles	81
9.3	Identification des paramètres géométriques des robots	81
9.4	Identification des paramètres dynamiques des robots	81
9.5	Compliance	81
9.6	Commande référencée capteur : commande en effort - commande hybride - as- servissements visuels	81
9.7	Parcourabilité - Dextérité - Manipulabilité	81
	Bibliographie	83
A	Sites web, blogs, réseaux d'informations et salons industriels	85
B	Logiciels et bibliothèques libres	87
B.1	Logiciels	87
B.2	Bibliothèques	87

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 manufacturière, c'est-à-dire les robots manipulateurs destinés à des tâches de soudure, de peinture, de manutention, d'emballage-conditionnement, etc. L'objectif est de fournir un ensemble minimal de connaissances pour comprendre le fonctionnement d'un robot industriel, son comportement et son implantation.

1.1 Définitions

1.1.1 Définition d'un robot

Étymologiquement, le mot « robot » tire sa racine du bulgare « robu », qui signifie « serviteur » et qui a donné naissance, entre autres, au russe « rabota » qui signifie « travail » ou « corvée » et au tchèque « robota » qui se traduit par « travail forcé ». C'est d'ailleurs l'écrivain tchèque Karel Capek qui 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 scientifiques et les ingénieurs programmeraient la « 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 une machine qui bouge commandée par un ordinateur, pouvant se substituer à l'homme pour effectuer certaines opérations, et capable d'en modifier de elle-même le cycle en appréhendant son environnement.

Cependant, il n'est pas toujours aisé d'établir une frontière claire entre robots et certains 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.

Définition 1 (Robot industriel pour manipulation). *Système de manipulation programmable de trois axes ou plus, à usage multiple, contrôlé automatiquement, reprogrammable, qui peut être mobile ou fixe lors de son utilisation automatique (définition ISO 8373).*

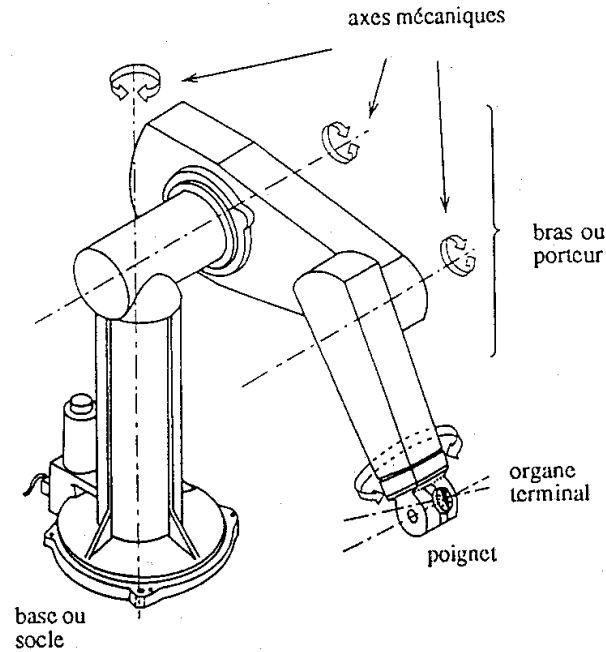


FIGURE 1.1 – Éléments d'un robot industriel (illustration extraite de [3]).

Un robot industriel doit en principe comporter au moins trois axes (ce qui n'est pas forcément le cas dans la classification japonaise). Le terme « reprogrammable » signifie que les mouvements programmés ou fonctions auxiliaires peuvent être changés sans entraîner d'altérations physiques. Enfin, un robot industriel peut être adapté pour une utilisation différente avec altérations physiques et est donc « à usage multiple ».

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 depuis quelques années de nombreux autres domaines sont explorés, en vue notamment de développer une *robotique de service*.

Définition 2 (Robot de service). *Robot mi ou totalement autonome accomplissant un travail directement utile à un usager ou sur un équipement, à l'exception des tâches de production industrielle.*

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 (aide aux handicapés, 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 située l'origine du premier élément de la structure articulée constituant le bras,

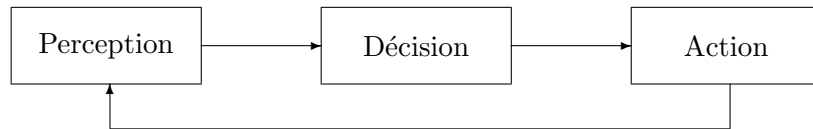


FIGURE 1.2 – Boucle perception-décision-action.

- porteur (ou bras) : chaîne constituée d'éléments de translation et/ou de rotation animés les uns par rapport aux autres et qui permet de déplacer et d'orienter un organe terminal (ou effecteur), comme par exemple un préhenseur ou un outil (selon les définitions, le porteur inclut ou non le poignet, qui est alors dans ce dernier cas considéré comme faisant partie de l'organe terminal),
- poignet : ensemble mécanique de l'extrémité du bras d'un robot, constitué d'articulations et permettant l'orientation de l'organe terminal.
- liaison active : liaison motorisée entre deux corps successifs d'un robot. La pratique industrielle fait que l'on parle ici d'*axe* (parfois d'« articulation », voire de « degré de liberté articulaire ») pour désigner un tel degré de mobilité. On dira donc communément qu'un robot a n axes.
- liaison passive : liaison non motorisée entre deux corps successifs d'un robot,
- coordonnée articulaire : variable représentant un degré de mobilité d'un robot, en rotation ou en translation. L'ensemble des coordonnées articulaires correspondant à une certaine posture du robot définit le vecteur des coordonnées articulaires ou généralisées. L'ensemble des vecteurs des coordonnées articulaires est appelé espace de configuration.
- coordonnée opérationnelle : 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, l'ensemble représentant le vecteur des coordonnées opérationnelles. Les coordonnées opérationnelles et articulaires sont reliées par la géométrie du robot ; la tâche que le robot doit effectuer se définit dans l'espace opérationnel, qui correspond à l'espace de la tâche.

En fonction du nombre d'articulations 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 à n axes avec $n < 6$ ne peut pas générer 6 ddl indépendants au niveau de son effecteur. Pour qu'il puisse le faire, un robot doit posséder au minimum 6 axes.
- Cette dernière condition est nécessaire mais pas suffisante car on peut trouver des robots redondants possédant plus de 6 axes et générant moins de 6 ddl.

D'autre part, l'espace 3D comportant six degrés de liberté, un robot, quel que soit le nombre de ses axes, ne pourra générer au maximum que six degrés de liberté de son organe terminal.

1.2.2 Boucle perception-décision-action

Un robot est système bouclé dont le fonctionnement peut être décrit par le schéma très général « perception-décision-action » représenté par la figure 1.2.

Le robot peut agir sur ses mouvements et son environnement à l'aide de ses actionneurs. 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 renseignent sur son état interne (positions, forces, intensités, etc.),
- les capteurs extéroceptifs qui l'informent 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.).

La robotique est donc un domaine d'intégration où interviennent plusieurs disciplines différentes : la mécanique, les actionneurs (électromagnétiques, piézoélectriques, hydrauliques, pneumatiques, etc.), les capteurs (position, vitesse, force, vision, etc.), l'électronique, l'automatique, l'informatique, la robotique « pure » (génération de trajectoires, planification, placement d'un robot, manipulabilité, etc.).

1.3 Configurations usuelles des robots industriels

1.3.1 Porteurs à chaîne cinématique ouverte (ou porteurs série)

La majorité des robots manipulateurs existant à l'heure actuelle présente un caractère anthropomorphe, en l'occurrence une forte ressemblance avec le bras humain. Ce type de porteur est constitué d'une succession de corps en série, chacun étant relié à son prédécesseur et à son successeur par une articulation à une mobilité, générant ainsi une chaîne cinématique ouverte. On parle alors de robots *série*.

Les articulations sont généralement soit des liaisons pivots (ou rotoïde, symbolisée par la lettre R) soit des liaisons glissières (ou prismatique, symbolisée par la lettre P).

Les robots industriels peuvent être classifiés en fonction de leur structure mécanique et du type des axes. Les principales configurations des porteurs industriels à chaîne cinématique ouverte sont :

- structure cartésienne (ou robot portique) = PPP
- structure cylindrique = RPP,
- structure SCARA (bi-cylindrique) = RRP,
- structure sphérique = RRP,
- structure anthropomorphique (bi-sphérique) = RRR.

Ces configurations 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 que l'on peut trouver des robots hyper-redondants, souvent nommés « trompes d'éléphant » qui peuvent être intéressants dans des cas où l'accès est rendu difficile par la présence d'obstacles (ex : soudure ou peinture à l'intérieur d'une carrosserie d'automobile). D'une manière générale, la redondance augmente la *dextérité* du robot. En outre, on peut envisager que si une articulation se bloque, la redondance fait que le robot peut néanmoins accomplir sa tâche.

NB: 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 Configurations du poignet

Suivant la tâche envisagée, on trouve des poignets à un, deux ou trois axes. De nombreux robots ont un poignet à deux degrés de liberté, car plus on s'éloigne de la base, plus le degré de liberté a tendance à coûter cher, de point de vue de la réalisation et de la commande (la partie terminale du robot s'alourdit, surtout si les moteurs du poignet ne sont pas ramenés sur

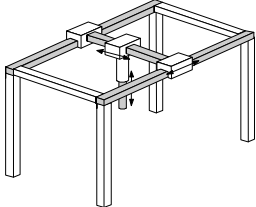
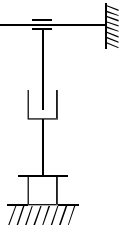
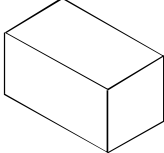

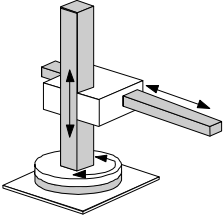
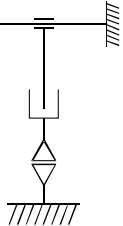
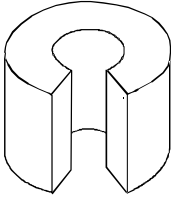

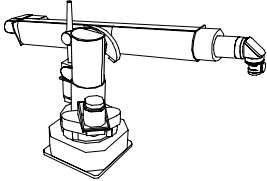
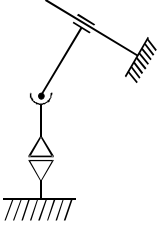
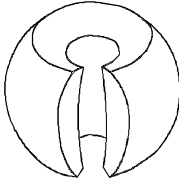
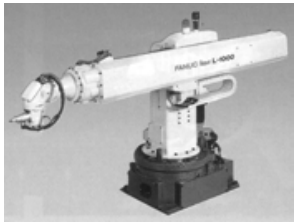
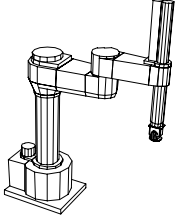
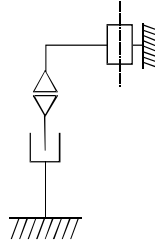
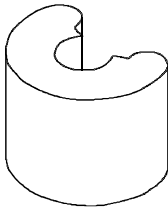

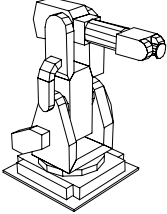
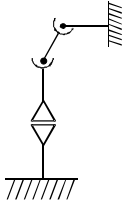
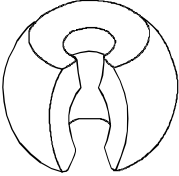

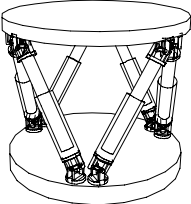
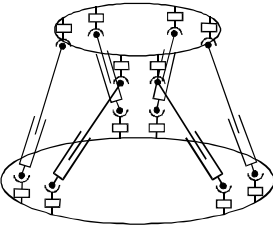
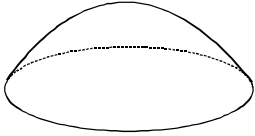

Robot	Axes		Examples
	Kinematic Structure	Workspace	Photo
 Cartesian Robot			
 Cylindrical Robot			
 Spherical Robot			
 SCARA Robot			
 Articulated Robot			
 Parallel Robot			

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

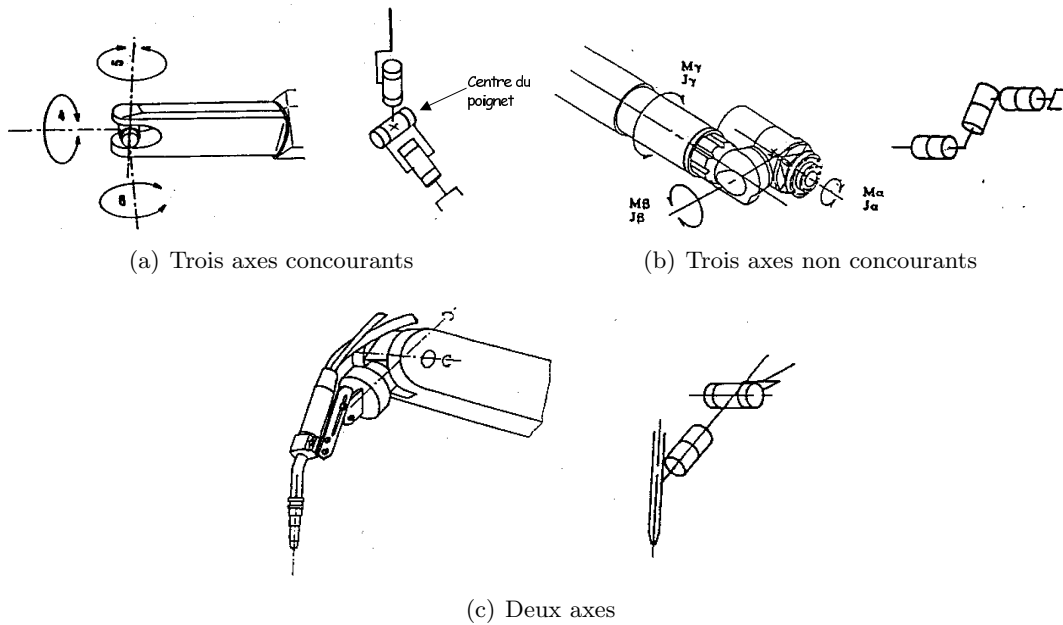


FIGURE 1.3 – Configurations du poignet (illustrations extraites de [3]).

la base, la commande s'en trouve ainsi compliquée et la masse embarquable diminuée) et de nombreuses tâches industrielles peuvent être correctement réalisées avec un robot à cinq axes (3+2), si elles ne nécessitent que cinq degrés de liberté (dans l'espace opérationnel) .

Néanmoins, la tendance actuelle consiste à utiliser des robots « complets » à six degrés de liberté (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 1.3). Cet ensemble est équivalent à une liaison sphérique. Elle permet donc d'obtenir une orientation quelconque de la pince et un découplage par rapport au porteur, d'où une simplification de la commande : le centre du poignet est fixe par rapport au bras, mais aussi par rapport à l'effecteur 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 Porteurs à chaîne cinématique fermée (ou porteurs parallèle)

Un robot parallèle est caractérisé par le fait que l'organe terminal est relié à la base du robot par des chaînes cinématiques distinctes et indépendantes, d'où des boucles cinématiques. Le robot parallèle le plus célèbre est la plate-forme de Gough-Stewart (1965) illustré par la figure 1.4. Ce type de structure était notamment utilisé à l'origine dans des simulateurs de vol pour l'entraînement des pilotes. Plus récemment, développé dans les années 1980, le robot suisse Delta (EPFL/ABB) connaît le succès industriel à partir des années 1990.

Il faut noter que dans de telles structures cinématiques, toutes les articulations ne sont pas actives. Certaines sont nécessairement passives.

Les avantages de ce type de structure sont les suivants :

- en dépit du fait que les débattements de ces porteurs sont plus faibles que ceux des porteurs séries, générant donc un volume de travail plus faible, leur structure articulaire met les erreurs de position (ainsi que les imprécisions des capteurs, les défauts divers : jeux, frottements, flexion des corps) en parallèle au lieu de les mettre en série. Ces robots ont donc par nature une meilleure précision. Ainsi ils sont intéressants pour

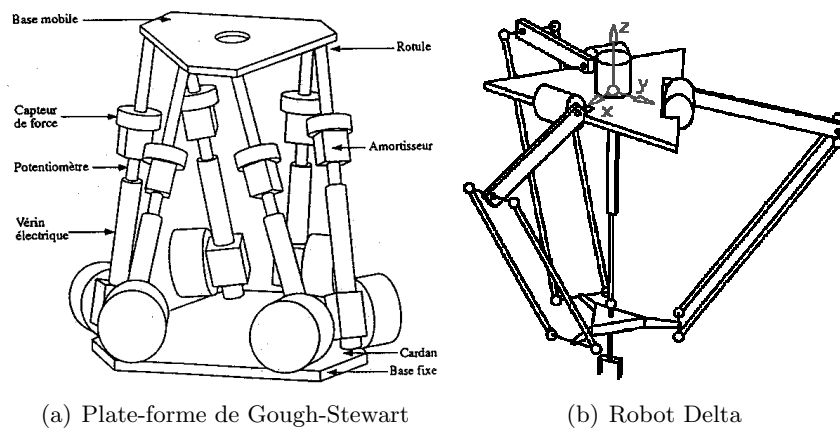


FIGURE 1.4 – Deux exemples de porteurs parallèle (illustrations extraites de [3] et de [5]).

les applications de type « petits mouvements et grande précision » (notamment en microrobotique) ;

- 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 corps, peu de déformation) ;
- l'utilisation de chaînes parallèles permet d'utiliser des composants identiques (moteurs, constituants mécaniques), d'où une facilitation de la maintenance.

Les applications industrielles des architectures parallèles sont en plein développement (il a fallu du temps pour résoudre certains problèmes techniques et de commande), aussi bien en robotique (pick and place agro-alimentaire, recherche pharmaceutique, assemblage, ...) qu'en machines outils (les centres d'usinage parallèles permettent de multiplier les performances par rapport aux centres d'usinage classiques).

NB: Il est également possible d'utiliser une structure parallèle comme poignet de robot afin d'augmenter la dextérité (et obtenir une compliance active ou passive du poignet).

Les types de transmission en chaîne fermée de type *parallélogramme*, tels que celui représenté en figure 1.5 permettant de réaliser une transmission de mouvement (mécanisme à un degré de liberté) à partir de quatre corps formant un parallélogramme, n'entrent pas dans la catégorie des robots parallèles. Les relations géométriques entre les différents angles restent simples et similaires à celles d'un robot série. Le parallélogramme articulé permet de ramener les actionneurs vers la base du robot, limitant la masse du robot en bout de porteur et augmentant ainsi les possibilités en terme de masse embarquée. En outre, une structure en parallélogramme peut être utilisée pour l'équilibrage d'un robot.

1.3.4 Robots à chaîne cinématique arborescente

Un robot à chaîne cinématique arborescente est constitué de plusieurs chaînes cinématiques ouvertes reliées à un corps central.

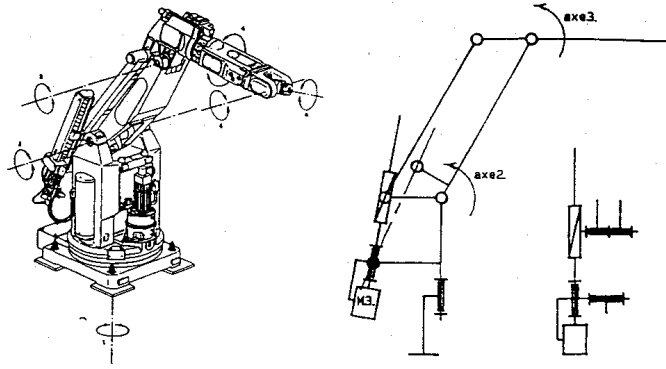


FIGURE 1.5 – Porteurs à parallélogramme (illustrations extraites de [5]).

En fonctionnement, un robot à chaîne cinématique arborescente 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.

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 particulièrement complexes.

1.4 Les marchés de la robotique¹

Fin 2012, le nombre de robots industriels opérationnels installés dans le monde dépassait le million d'unités (les estimations vont de 1 235 000 si l'on considère une durée de vie de 12 ans, à 1 500 000 si la durée de vie retenue est de 15 ans ce qui d'après les dernières études semble plus réaliste).

En 2011, la Corée du Sud a atteint la densité de robot la plus élevée dans le monde, environ 347 robots industriels en fonctionnement pour 10 000 employés dans l'industrie manufacturière. La même année, la densité de robot au Japon était de 339, celle de l'Allemagne de 261. 10 pays avaient une densité de robot entre 104 (pour l'Autriche) et 149 (pour l'Italie), 5 pays de 50 à 100 (dont la France), 11 pays de 20 à 49 et tous les autres avaient moins de 20 robots en fonctionnement pour 10 000 employés. La densité de robot moyenne estimée dans le monde était d'environ 55 robots industriels pour 10 000 employés.

C'est un marché dynamique qui connaît toujours une forte croissance notamment en Asie (cf. figure 1.6). 2011 a été l'année où il s'est vendu le plus de robots industriels depuis 1961 (environ 166 000 unités).

Malgré l'affaiblissement du contexte économique mondial, les ventes de robots industriels devraient continuer à progresser. L'IFR estime qu'entre 2014 et 2016 les ventes de robots dans le monde devraient augmenter d'environ 6% en moyenne par an (4% dans les Amériques et en Europe, 8% en Asie et Australie). Les plus grosses progressions sont prévues en Amérique du Nord, en Chine, en Corée du Sud, au Brésil et en Europe de l'est.

Le secteur automobile demeure l'un des premiers marchés consommateurs de robots industriels : environ un tiers des robots en service dans le monde (deux tiers en France). Au Japon, en Allemagne, en Italie et en France, il y a plus d'un robot pour 10 employés dans l'industrie automobile. La soudure par points représente le plus grand domaine d'application.

1. L'essentiel des chiffres présentés ici est issu de l'UE, de l'ONU, de l'IFR (International Federation of Robotics) et des associations nationales. Une synthèse du rapport « World Robotics » qui fournit chaque année un ensemble de données sur la robotique mondiale, est disponible sur www.ifrstat.org.

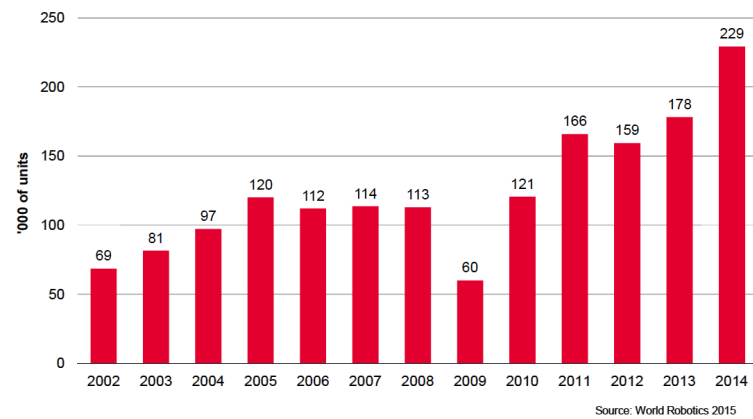


FIGURE 1.6 – Ventes annuelles de robots industriels (source : IFR World Robotics).

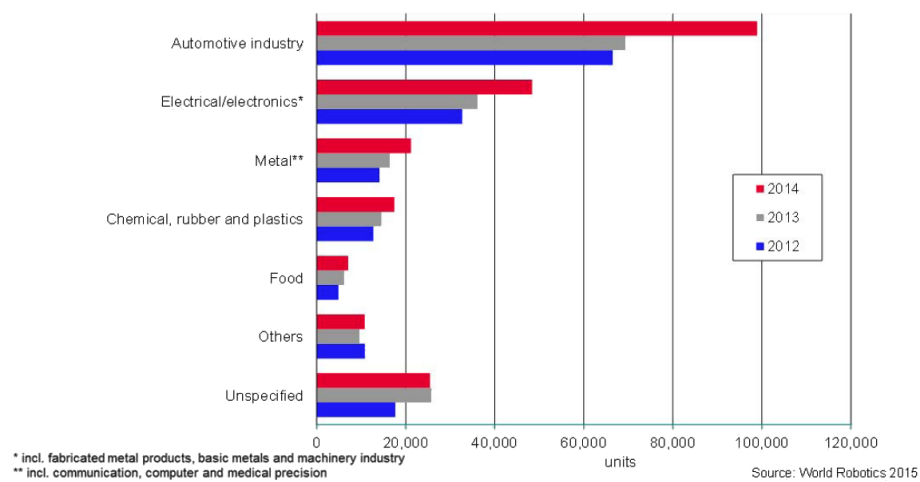


FIGURE 1.7 – Ventes annuelles de robots par secteurs industriels (source : IFR World Robotics).

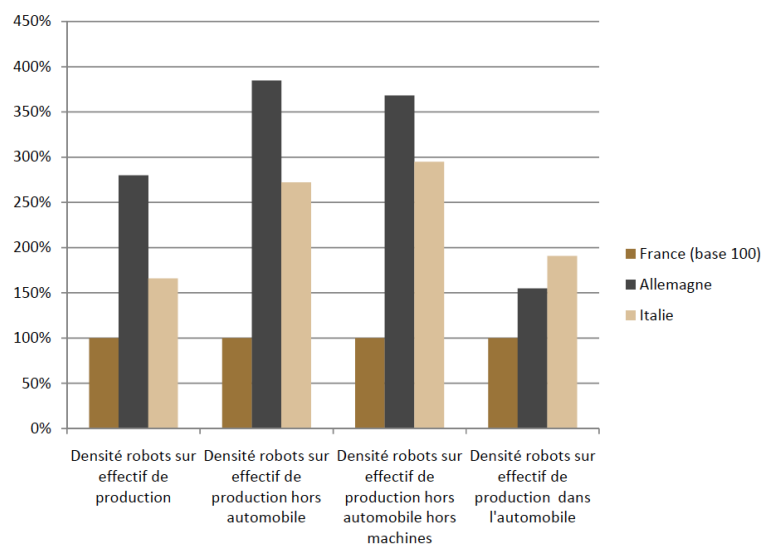


FIGURE 1.8 – Comparaison de la densité de robot entre l'Allemagne, l'Italie et la France (source : analyse Nodal d'après données IFR World Robotics 2008).

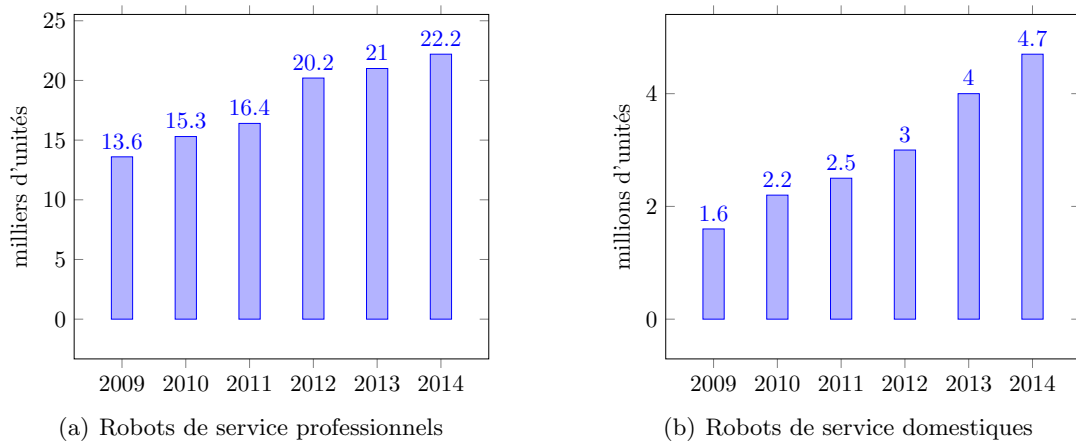


FIGURE 1.9 – Ventes annuelles de robots de service (source : IFR World Robotics).

Au delà du secteur automobile, la demande de robots industriels augmente également dans d'autres secteurs comme l'électronique, la plasturgie, l'agro-alimentaire et la métallurgie (cf. figure 1.7).

La France accuse un certain retard en terme d'équipement robotique dans les secteurs autre que l'automobile et notamment dans les PME/PMI (cf. figure 1.8). Dans ce contexte, des industriels français ont pris en 2008 une initiative visant à promouvoir l'intégration de robots à la production, dans le but d'éviter de délocaliser². En 2013, le gouvernement français a lancé un plan d'aide pour la robotisation des PME baptisé ROBOT Start PME³.

Le secteur de la robotique de service est également en plein essor (cf. figure 1.9). Depuis 1998, environ 150,000 robots de service professionnels ont été vendus dans le monde. Il s'agit essentiellement de robots militaires, de robots agricoles (machines de traite automatisée), de robots médicaux et de robots de manutention.

Les ventes de robots de services domestiques ont atteint 4,000 000 en 2013 (cf. figure 1.9). Il s'agit pour les deux tiers de robots de nettoyage (aspirateurs) et pour le reste de robots de divertissements (jouets, robots éducatifs).

Les ventes de robots d'assistance pour personnes handicapées commencent à décoller. Un grand nombre de projets de recherche nationaux dans de nombreux pays se concentrent sur ce marché énorme pour l'avenir robots de service. Pour l'anecdote⁴, en 2006, le gouvernement sud-coréen avait demandé à des scientifiques et des industriels de développer des robots totalement intégrés à la société. *Mon objectif personnel est qu'il y ait un robot dans chaque foyer à l'horizon 2010* expliquait un cadre du ministère. Il était ainsi prévu que des robots domestiques soient produits en série. Ils devaient, par exemple, transmettre des messages aux parents, enseigner l'anglais aux enfants ou jouer avec eux lorsque ceux-ci s'ennuient. En dehors du foyer, ils devaient également être capables de guider des clients dans les bureaux de poste ou même de patrouiller dans des espaces publics en localisant des intrus puis en transmettant des images à des centres de surveillance.

2. Voir www.robocaliser.com. Robotcaliser est une initiative du Symop, Syndicat des Entreprises de Technologies de Production

3. Voir www.robotstartpme.fr

4. Tirée d'un bulletin d'information de l'Ambassade de France en Corée, 2006.

1.5 Conclusion

Au cours des dernières années, les améliorations techniques des robots industriels ont été considérables, et ce pour un prix globalement divisé par plus de 3 entre 1990 et 2006. La durée de vie moyenne d'un robot se situe entre 12 et 15 ans (avec un retour sur investissement très court, souvent d'un à deux ans).

Le robot industriel est l'équipement de production flexible par excellence, qui est désormais performant, relativement peu coûteux et fiable. Un grand constructeur japonais d'équipements de production en général et de robots en particulier, indique que le robot est actuellement 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.

Malgré l'état actuel de sophistication des robots, d'énormes progrès peuvent encore être faits, ne serait-ce que pour l'amélioration des robots industriels existants (sans parler du développement de robots totalement nouveaux, fondés sur des technologies nouvelles et dotés de fonctions nouvelles).

Les robots industriels doivent travailler plus vite, sans perdre et même si possible en améliorant leur précision :

- au niveau de la commande, mise en œuvre de commandes toujours plus performantes (ex : précision, prise en compte de la charge transportée, prise en compte des imperfections, prise en compte de l'environnement),
- allègement de la structure (le rapport entre la charge utile et la masse des robots reste faible),
- on peut également étudier des structures de robots flexibles ou souples, plus légers, plus rapides, permettant d'emmagasiner de l'énergie potentielle et offrant une compliance passive parfois utile.

En robotique industrielle, une tendance actuelle est le travail collaboratif entre plusieurs robots et, plus récemment, 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). Par ailleurs, les moyens de programmation des robots sont également en évolution (interfaces homme-machine, programmation hors ligne, CAO).

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 [9]) et les robots de plus en plus grands (manipulation d'ensembles plus complets, plus lourds, voire remplacement d'éléments de convoyage, c'est le robot qui amène la pièce à l'outil et plus le robot qui amène l'outil à la pièce).

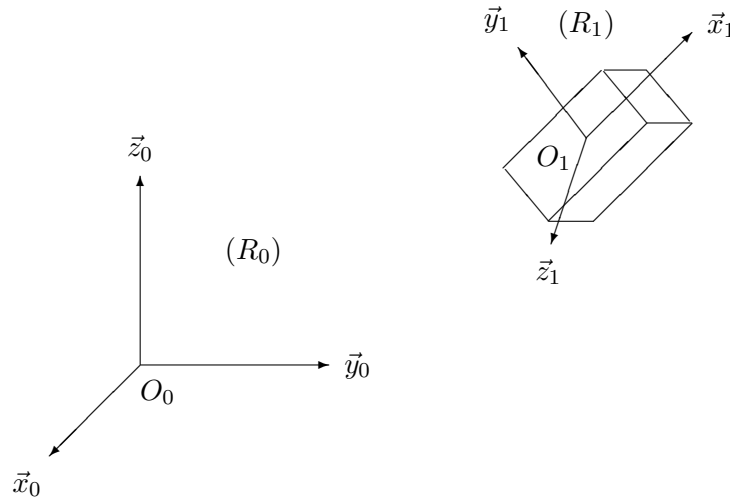
Un point essentiel pour le futur est celui de l'« intelligence » des robots. Il faut en effet conférer aux robots de plus en plus d'autonomie décisionnelle (génération de plans d'actions), de mobilité, s'accompagnant de la mise en œuvre de capteurs extéroceptifs, notamment la vision artificielle, pour la perception de l'environnement. Cette autonomie permettra aux fabricants de robots de conquérir de nouveaux marchés dans le secteur industriel et surtout dans les services.

Chapitre 2

Repérage d'un solide dans l'espace

Soit un solide S_1 quelconque, de centre O_1 (le centre étant un point du solide choisi de façon arbitraire). On souhaite caractériser sa situation dans l'espace.

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



2.1 Représentation d'un point

Soit P un point du solide S_1 . Les coordonnées de P dans R_1 sont représentées par le vecteur ${}^1\mathbf{p}$ tel que :

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

NB: L'exposant 1 précédant \mathbf{p} désigne le repère dans lequel les coordonnées de P sont exprimées.

2.2 Position

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 noté

${}^0\mathbf{o}_1$.

$${}^0\mathbf{o}_1 = \overrightarrow{O_0O_1}_{/R_0} = \begin{bmatrix} \overrightarrow{O_0O_1} \cdot \vec{x}_0 \\ \overrightarrow{O_0O_1} \cdot \vec{y}_0 \\ \overrightarrow{O_0O_1} \cdot \vec{z}_0 \end{bmatrix}$$

Si les repères sont équipollents (translation pure entre les repères), le vecteur ${}^0\mathbf{o}_1$ permet d'exprimer dans le repère R_0 les coordonnées d'un point données dans le repère R_1 . En effet, dans ce cas, on a $\overrightarrow{O_1P}_{/R_0} = \overrightarrow{O_1P}_{/R_1}$ et donc on peut écrire :

$$\overrightarrow{O_0P}_{/R_0} = \overrightarrow{O_0O_1}_{/R_0} + \overrightarrow{O_1P}_{/R_0} = \overrightarrow{O_0O_1}_{/R_0} + \overrightarrow{O_1P}_{/R_1}$$

soit :

$${}^0\mathbf{p} = {}^0\mathbf{o}_1 + {}^1\mathbf{p}$$

En ce qui concerne l'orientation du repère R_1 par rapport au repère de référence R_0 , elle peut s'exprimer de différentes manières. L'orientation peut se définir par une matrice de rotation, par trois angles, par un vecteur et un angle ou encore par un quaternion.

2.3 Orientation déterminée par une matrice

2.3.1 Définition

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

$${}^1\mathbf{p} = \overrightarrow{O_1P}_{/R_1} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

On a alors :

$$\overrightarrow{O_1P} = a\vec{x}_1 + b\vec{y}_1 + c\vec{z}_1$$

Si les origines des deux repères R_0 et R_1 sont confondues (rotation pure entre les repères), 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 :

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

Soit :

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

La matrice ${}^0\mathbf{M}_1$ est la matrice de rotation (également appelée matrice des cosinus directeurs) correspondant à la matrice dite « 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 :

$${}^0\mathbf{M}_1 = \begin{bmatrix} \vec{x}_1/R_0 & \vec{y}_1/R_0 & \vec{z}_1/R_0 \end{bmatrix} = \begin{bmatrix} \vec{x}_1 \cdot \vec{x}_0 & \vec{y}_1 \cdot \vec{x}_0 & \vec{z}_1 \cdot \vec{x}_0 \\ \vec{x}_1 \cdot \vec{y}_0 & \vec{y}_1 \cdot \vec{y}_0 & \vec{z}_1 \cdot \vec{y}_0 \\ \vec{x}_1 \cdot \vec{z}_0 & \vec{y}_1 \cdot \vec{z}_0 & \vec{z}_1 \cdot \vec{z}_0 \end{bmatrix}$$

Notons que les 3 lignes sont les composantes des vecteurs de la base de R_0 exprimés dans R_1 soit respectivement \vec{x}_0/R_1 , \vec{y}_0/R_1 et \vec{z}_0/R_1 .

Chaque élément de la matrice est nommé *cosinus directeur*.

Outre le fait d'exprimer l'orientation d'un repère par rapport à un autre, la matrice ${}^0\mathbf{M}_1$ dite « 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 données dans le repère R_1 :

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

NB: A l'inverse, on peut bien entendu écrire : ${}^1\mathbf{v} = {}^1\mathbf{M}_0 \cdot {}^0\mathbf{v}$.

2.3.2 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 M_0^1 sont donc liés par des équations de contrainte. Ils ne sont donc pas indépendants et l'information fournie par M_0^1 sur l'orientation de S_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 :

$${}^1\mathbf{M}_0 = \begin{bmatrix} \vec{x}_{0/R_1} & \vec{y}_{0/R_1} & \vec{z}_{0/R_1} \end{bmatrix} = \begin{bmatrix} \vec{x}_0 \cdot \vec{x}_1 & \vec{y}_0 \cdot \vec{x}_1 & \vec{z}_0 \cdot \vec{x}_1 \\ \vec{x}_0 \cdot \vec{y}_1 & \vec{y}_0 \cdot \vec{y}_1 & \vec{z}_0 \cdot \vec{y}_1 \\ \vec{x}_0 \cdot \vec{z}_1 & \vec{y}_0 \cdot \vec{z}_1 & \vec{z}_0 \cdot \vec{z}_1 \end{bmatrix}$$

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 :

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

Les matrices de passage sont des matrices unitaires et orthogonales.

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 :

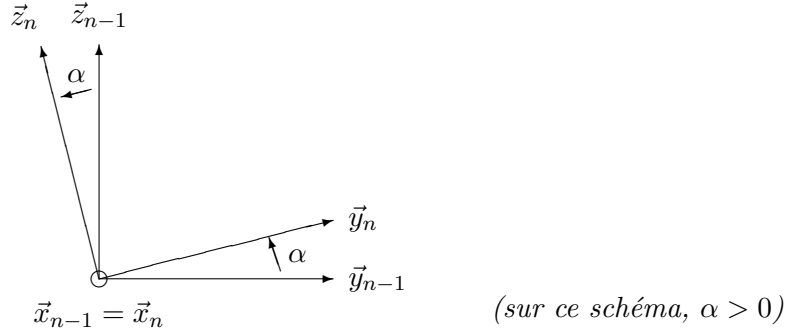
$$\boxed{{}^0\mathbf{M}_2 = {}^0\mathbf{M}_1 \cdot {}^1\mathbf{M}_2}$$

2.3.3 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 .

NB: Les rotations sont définies dans le sens direct.

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

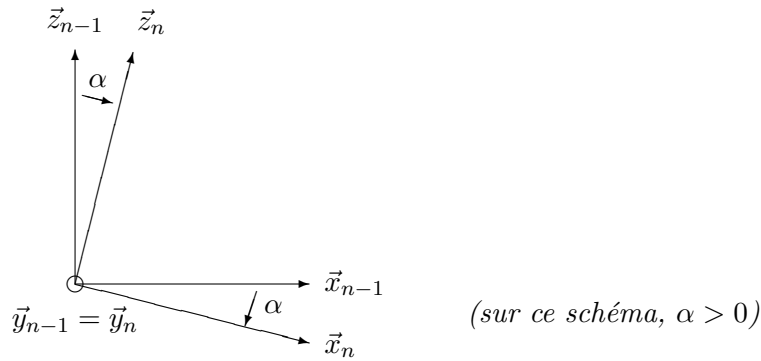


En calculant les coefficients de la matrice de passage entre R_{n-1} et R_n dans cette situation, il vient :

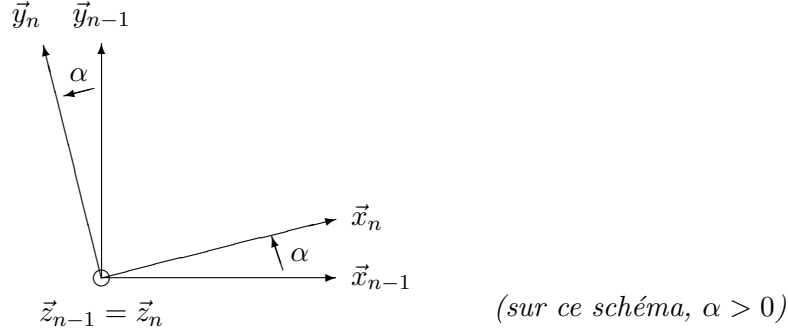
$${}^{n-1}\mathbf{M}_n = \text{rot}(\vec{x}_{n-1}, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha \\ 0 & s_\alpha & c_\alpha \end{bmatrix}$$

NB: c_α désigne $\cos(\alpha)$ et s_α désigne $\sin(\alpha)$.

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



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

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

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

2.4 Orientation déterminée par trois angles

Outre l'utilisation de la matrice de rotation décrite précédemment, on peut également exprimer les trois ddl de rotation de R_1 par rapport à R_0 de manière explicite et non redondante, par trois angles.

Pour ce faire, on définit $R_1 = (O_O, \vec{x}_1, \vec{y}_1, \vec{z}_1)$ 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 . Ces trois rotations peuvent alors être définies de différentes manières (à condition que deux rotations consécutives ne soient pas autour du même axe).

Ainsi, pour exprimer de cette manière l'orientation d'un solide, ou de l'organe terminal d'un robot, on peut choisir, parmi les diverses possibilités, celle qui, selon le robot considéré, permettra l'expression de l'orientation la plus simple en fonction des q_i (coordonnées articulaires du robot). Notons que les angles nautiques ont peut-être l'avantage d'être plus « parlants », donc plus ergonomiques pour un opérateur qui essaie de se représenter l'orientation de l'effecteur par rapport à la base du robot. On trouve donc ce type de représentation sur la plupart des robots industriels.

2.4.1 Angles nautiques

Les angles nautiques, également appelés angles de cardan, angles de Tait-Bryan ou encore lacet-tangage-roulis (LTR), définissent les trois rotations suivantes :

- le lacet (yaw) α , autour de l'axe \vec{z}_0 , fait passer de R_0 à R' ,
- le tangage (pitch) β , autour de l'axe \vec{y}' , fait passer de R' à R'' ,
- le roulis (roll) γ , autour de l'axe \vec{x}'' , fait passer de R'' à R_1 .

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

NB: Pour garantir l'unicité de la représentation de l'orientation par les trois angles nautiques, il faut choisir ces angles dans les intervalles suivants :

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

De plus, si $\beta = -\frac{\pi}{2}$, la représentation n'est plus unique.

La matrice de rotation équivalente à une orientation donnée par les angles nautiques est :

$$\begin{aligned} {}^0\mathbf{M}_1 &= \text{rot}(\vec{z}_0, \alpha) \cdot \text{rot}(\vec{y}', \beta) \cdot \text{rot}(\vec{x}'', \gamma) = \begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{bmatrix} \\ &= \begin{bmatrix} c_\alpha c_\beta & -s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma & s_\alpha s_\gamma + c_\alpha s_\beta c_\gamma \\ s_\alpha c_\beta & c_\alpha c_\gamma + s_\alpha s_\beta s_\gamma & -c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \end{aligned}$$

Inversement, connaissant une matrice de rotation ${}^0\mathbf{M}_1$, il est possible de calculer les angles nautiques :

— si $m_{11} = 0$ et $m_{21} = 0$,

$$\begin{cases} \alpha &= 0 \\ \beta &= \text{atan2}(-m_{31}, m_{11}) \\ \gamma &= \text{atan2}(-m_{23}, m_{22}) \end{cases}$$

— 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 atan2 est une fonction *arctangente* 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 :

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

2.4.2 Angles d'Euler

Les angles d'Euler définissent les trois rotations suivantes :

- la précession ψ , autour de l'axe \vec{z}_0 , fait passer de R_0 à R' ,
- la nutation θ , autour de l'axe \vec{x}' , fait passer de R' à R'' ,
- la rotation propre ϕ , autour de l'axe \vec{z}'' , fait passer de R'' à R_1 .

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

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

$${}^0\mathbf{M}_1 = \text{rot}(\vec{z}_0, \psi) \cdot \text{rot}(\vec{x}', \theta) \cdot \text{rot}(\vec{z}'', \phi) = \begin{bmatrix} c_\phi c_\psi - s_\phi c_\theta s_\psi & -s_\phi c_\psi - c_\phi c_\theta s_\psi & s_\theta s_\psi \\ c_\phi s_\psi + s_\phi c_\theta s_\psi & -s_\phi s_\psi + c_\phi c_\theta s_\psi & -s_\theta c_\psi \\ s_\phi s_\theta & c_\phi s_\theta & c_\theta \end{bmatrix}$$

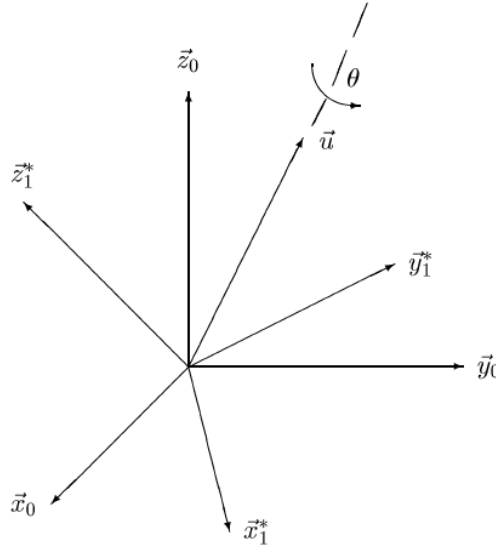
NB: Pour garantir l'unicité de la représentation de l'orientation par les trois angles d'Euler, il faut choisir ces angles dans les intervalles suivants :

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

De plus, si $\theta = 0$, la représentation n'est plus unique.

Comme pour les angles nautiques, il existe des formules permettant de retrouver les angles d'Euler à partir d'une matrice de rotation [8].

2.5 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. En effet, une matrice de rotation a toujours une valeur propre égale à un et deux valeurs propres conjuguées $\cos(\theta) \pm i \sin(\theta)$ où θ 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 ${}^0\mathbf{M}_1 \cdot \mathbf{u} = \mathbf{u}$. L'orientation de R_1 par rapport à R_0 est donc complètement définie par quatre paramètres qui décrivent une rotation unique d'angle θ (avec $0 \leq \theta \leq \pi$) autour de l'axe de vecteur unitaire \mathbf{u} .

Inversement, connaissant un vecteur de rotation unitaire \mathbf{u} et un angle de rotation θ , il est possible de calculer la matrice de rotation en utilisant la formule de Rodrigues [8] :

$${}^0\mathbf{M}_1 = \mathbf{I}_3 + \sin(\theta)\mathbf{S}(\mathbf{u}) + (1 - \cos(\theta))\mathbf{u}\mathbf{u}^T$$

avec $\mathbf{S}(\mathbf{u})$ la matrice antisymétrique associée au vecteur \mathbf{u} .

De manière plus compacte, la rotation peut être également représentée par le vecteur $\theta\mathbf{u}$ et donc seulement 3 paramètres. Bien que cette représentation soit minimale, elle n'est pas utilisée car elle pose des problèmes de conditionnement au voisinage de $\theta = 0$.

2.6 Orientation déterminée par les quaternions

Les quaternions sont également appelés paramètres Euler ou paramètres d'Olinde-Rodrigues. Dans cette représentation, l'orientation est également exprimée de manière redondante par quatre paramètres qui décrivent une rotation unique d'angle θ ($0 \leq \theta \leq \pi$) autour d'un axe de vecteur unitaire \mathbf{u} .

Les quaternions q_1 à q_4 sont définis comme suit :

$$\begin{cases} q_1 = \cos(\theta/2) \\ q_2 = u_x \sin(\theta/2) \\ q_3 = u_y \sin(\theta/2) \\ q_4 = u_z \sin(\theta/2) \end{cases} \Rightarrow \begin{cases} q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \\ \cos \theta = 2q_1^2 - 1 \end{cases}$$

Dans ces conditions et tous calculs faits, la matrice de rotation s'exprime de la façon suivante :

$${}^0\mathbf{M}_1 = \begin{bmatrix} 2(q_1^2 + q_2^2) - 1 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & 2(q_1^2 + q_3^2) - 1 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & 2(q_1^2 + q_4^2) - 1 \end{bmatrix}$$

On trouve cette représentation de l'orientation de l'organe terminal du robot par rapport à sa base sur certains robots industriels. Par rapport aux matrices de rotation, les calculs utilisant les quaternions sont plus efficaces (en terme de complexité) et numériquement stable. Ils sont notamment utilisés en infographie 3D.

2.7 Transformation homogène

2.7.1 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 $\overrightarrow{O_1P}_{/R_1}$. Pour ce faire, on peut 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 :

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

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 :

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

qu'on peut également écrire :

$$\boxed{{}^0\mathbf{p} = {}^0\mathbf{o}_1 + {}^0\mathbf{M}_1 \cdot {}^1\mathbf{p}}$$

Écrivons cette expression sous la forme matricielle suivante :

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

On pose :

$${}^0\mathbf{T}_1 = \begin{bmatrix} {}^0\mathbf{M}_1 & \overrightarrow{O_0O_{1/R_0}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{M}_1 & {}^0\mathbf{o}_1 \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

La matrice ${}^0\mathbf{T}_1$ est appelée *matrice de transformation homogène*. Elle contient les mêmes informations que les équations écrites précédemment, mais sous une forme matricielle plus « compacte ».

La matrice de transformation homogène ${}^0\mathbf{T}_1$ dite « de passage de R_0 à R_1 » 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 :

$$\begin{bmatrix} \overrightarrow{O_0P_{/R_0}} \\ 1 \end{bmatrix} = {}^0\mathbf{T}_1 \cdot \begin{bmatrix} \overrightarrow{O_1P_{/R_1}} \\ 1 \end{bmatrix}$$

soit :

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

Le vecteur \mathbf{p} (de dimension 4) représente les *coordonnées homogènes* du point P .

On dit également que la transformation ${}^0\mathbf{T}_1$ définit le repère R_1 dans le repère R_0 .

Outre les changements de repère, les matrices de transformations permettent de représenter les homothéties ainsi que les modèles de projections utilisés en vision comme la projection perspective et la projection orthographique (cf. chapitre 9.6).

2.7.2 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 :

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

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 :

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

et :

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

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

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

2.8 Conclusion

La situation d'un solide S_1 dans l'espace peut être représenté sous différentes formes :

- le vecteur $\overrightarrow{O_0O_1}$ et les trois angles nautiques α , β et γ ,
- le vecteur $\overrightarrow{O_0O_1}$ et les trois angles d'Euler ψ , θ et ϕ ,
- le vecteur $\overrightarrow{O_0O_1}$, l'angle θ et un vecteur unitaire \mathbf{u} ,
- le vecteur $\overrightarrow{O_0O_1}$ et le quaternion q_1 , q_2 , q_3 et q_4 ,
- le vecteur $\overrightarrow{O_0O_1}$ et la matrice de rotation ${}^0\mathbf{M}_1$,
- la matrice de transformation homogène ${}^0\mathbf{T}_1$.

En robotique, les quatre premières représentations sont utilisées pour décrire les tâches qu'un robot doit effectuer, notamment dans les langages de programmation et les interfaces homme-machine. 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

Modèles géométriques direct et inverse

Rappelons qu'un robot est constitué d'un mécanisme polyarticulé, le porteur, et d'un outil déplacé et orienté par le porteur, l'effecteur ou organe terminal.

Dans la majorité des cas, on s'intéresse exclusivement aux mouvements de l'effecteur dans l'espace opérationnel, sans d'ailleurs s'intéresser nécessairement à celui dans le même espace des différents corps constituant le robot (sauf en cas de risque de collision du porteur avec des objets de son environnement).

Le mouvement de l'effecteur dans l'espace est totalement décrit par l'évolution de 6 paramètres (3 pour la position et 3 pour l'orientation), chacun représentant alors une coordonnée opérationnelle, regroupés en un vecteur colonne (6×1) noté \mathbf{x} .

Ce mouvement est engendré par l'actionnement de n articulations actives (motorisées) du porteur, dont les variables (coordonnées articulaires) peuvent être regroupées en un vecteur colonne ($n \times 1$) noté \mathbf{q} (\mathbf{q} définit la « posture » du robot).

NB: La $i^{\text{ème}}$ coordonnée de \mathbf{x} (resp. \mathbf{q}) est noté x_i (resp. q_i).

3.1 Modélisation géométrique directe d'un robot

3.1.1 Définitions et hypothèses

Définition 3. On appelle « modèle géométrique direct » (MGD) d'un robot, l'application \mathbf{f} de \mathbb{R}^n dans \mathbb{R}^6 (si n est le nombre d'axes du robot) exprimant \mathbf{x} en fonction de \mathbf{q} :

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

Le MGD permet donc de connaître les mouvements de l'effecteur dans l'espace de la tâche, en fonction des mouvements 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.

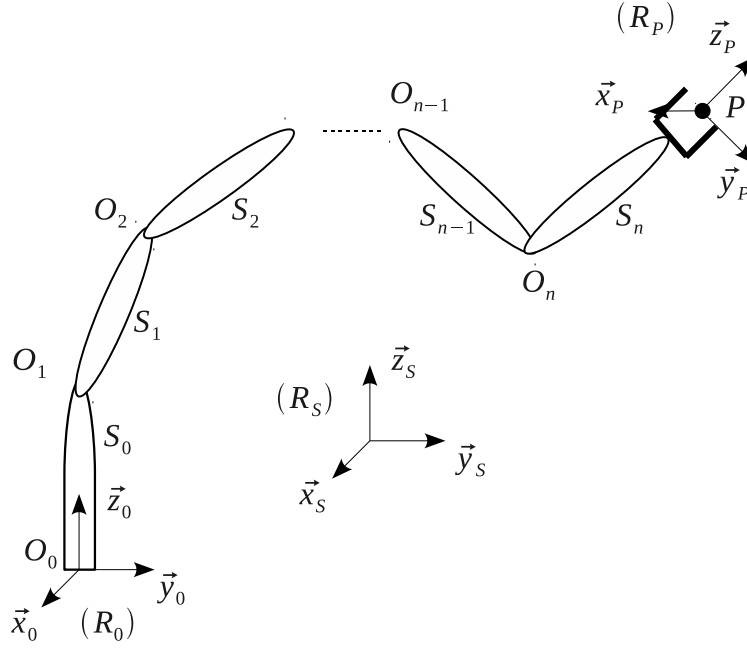


FIGURE 3.1 – Conventions de repérage.

NB: Le modèle géométrique est donc un système de 6 équations permettant de calculer les 6 inconnues x_i en fonction des n valeurs q_i .

Les structures mécaniques de robot étudiées ici ont les caractéristiques suivantes :

- les robots sont à chaîne cinématique ouverte, le dernier corps supportant l'effecteur (ou organe terminal) ;
- les corps sont parfaitement rigides et sont connectés entre eux par des articulations « 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 corps du robot étudié de la manière suivante :

- à chaque corps 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 corps ;
- le repère R_0 est situé à la base du robot, l'axe \vec{z}_0 vers le haut
- 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* ;
- le repère outil R_P associé à l'organe terminal a son vecteur \vec{z}_P dans l'axe de l'effecteur et « sortant » du robot (dirigé vers l'extérieur) ; R_P peut être distinct de R_n ;
- 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).
- les coordonnées articulaires sont notés q_i (q_i désignera un angle pour une articulation rotoïde et une longueur pour une articulation prismatique) ;
- dans la posture dite de *référence* toutes les coordonnées articulaires sont nulles,

NB: Les hypothèses concernant la structure des robots étudiées sont quelque peu restrictives, mais sont valables pour la grande majorité des robots industriels. Les hypothèses sur le choix des repères ne génèrent en revanche aucune restriction. Elles sont destinées uniquement à simplifier le calcul des modèles.

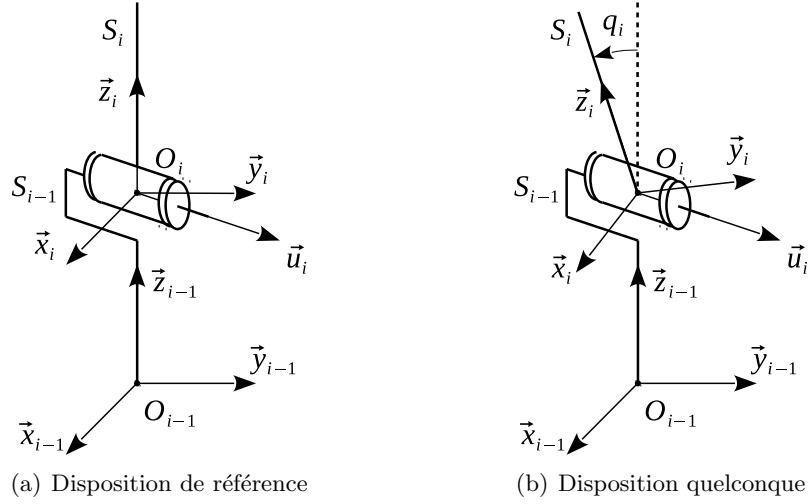


FIGURE 3.2 – Convention de repérage d'une articulation rotoïde.

Il s'agit maintenant d'exprimer le modèle géométrique direct d'un robot, $\mathbf{x} = \mathbf{f}(\mathbf{q})$, qui exprime la position et l'orientation de l'organe terminal en fonction des coordonnées articulaires du robot.

3.1.2 Modélisation d'une articulation

Soient deux solides S_{i-1} et S_i , liés par une articulation de mobilité 1, de type rotoïde ou prismatique.

Articulation rotoïde (liaison 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 telle manière qu'ils soient équipollents (c'est-à-dire qu'ils se déduisent par une simple translation) dans la disposition de référence.

D'autre part, la rotation de S_i par rapport à S_{i-1} est d'angle q_i autour de l'axe de rotation de vecteur unitaire \vec{u}_i , avec $q_i = 0$ dans la disposition de référence.

Enfin, on choisit l'origine O_i de R_i sur l'axe de rotation et on fait coïncider l'un des axes de R_i avec l'axe de rotation \vec{u}_i . Ainsi, la matrice de passage ${}^{i-1}\mathbf{M}_i$ s'écrit aisément.

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

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

ou sous forme d'une 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} . De même, $\overrightarrow{O_iP}$ est un vecteur de longueur constante qui peut s'exprimer aisément dans R_i .

Par exemple, si on choisit P sur \vec{z}_i , et que l_i désigne la longueur O_iP , on peut alors écrire :

$$\overrightarrow{O_iP}_{/R_i} = \begin{bmatrix} 0 \\ 0 \\ l_i \end{bmatrix}$$

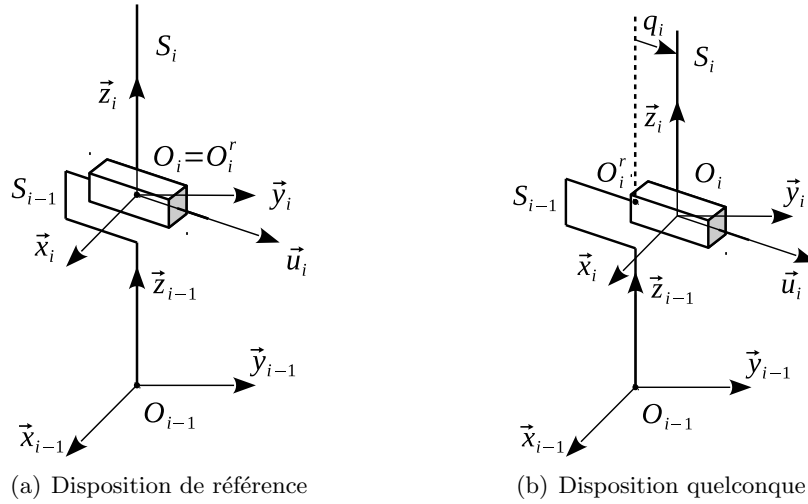


FIGURE 3.3 – Convention de repérage d'une articulation prismatique.

Articulation prismatique (liaison glissière)

Dans ce cas, le mouvement relatif des deux solides est une translation d'axe représenté par son vecteur unitaire \vec{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 R_{i-1} et R_i équipollents dans la disposition de référence, c'est-à-dire correspondant à une translation pure. R_{i-1} et R_i resteront ici constamment équipollents.

Enfin, on choisit l'origine O_i de R_i sur l'axe de translation.

Le point O_i^r désigne l'origine O_i du repère R_i dans la disposition de référence. La longueur $O_{i-1}O_i^r$ est donc constante. Elle s'exprime aisément dans R_{i-1} . Lorsqu'un mouvement est effectué, O_i « s'éloigne » de O_i^r . On peut écrire :

$$\overrightarrow{O_{i-1}P}_{/R_{i-1}} = \overrightarrow{O_{i-1}O_i^r} + \overrightarrow{O_i^r O_i} + \overrightarrow{O_i P}$$

C'est à dire :

$$\boxed{\overrightarrow{O_{i-1}P}_{/R_{i-1}} = \overrightarrow{O_{i-1}O_i^r} + q_i \vec{u}_i + \overrightarrow{O_i P}}$$

Ou encore sous forme d'une matrice de transformation homogène :

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

Quand cela est possible, on peut choisir l'un des axes de R_{i-1} (c'est à dire également l'un des axes de R_i), par exemple \vec{x}_{i-1} , identique à l'axe de translation et les points O_{i-1} , O_i et P le long de cet axe tel qu'indiqué sur la figure précédente. Dans ces conditions, on peut écrire :

$$\overrightarrow{O_{i-1}P}_{/R_{i-1}} = (l_{i-1} + q_i + l_i) \vec{x}_{i-1}$$

où l_{i-1} et l_i désigne respectivement les longueurs $O_{i-1}O_i^r$ et $O_i P$.

3.1.3 Orientation de l'organe terminal

De manière implicite

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 (par l'une des trois formes explicitée au paragraphe 2.3) ou à un axe de translation (dans le cas de la translation, cette matrice est la matrice identité). On en déduit donc aisément que :

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

et :

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

De manière explicite

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

$$\vec{z}_{n/R_0} = {}^0\mathbf{M}_n \cdot {}^n\mathbf{M}_P \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

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

3.1.4 Position de l'organe terminal

On cherche à calculer les coordonnées du point P de l'organe terminal dans le repère R_0 , soit $\overrightarrow{O_0P}_{/R_0}$.

Avec les matrices d'orientation

On écrit :

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

S'il y a des translations, une par exemple entre le corps $i-1$ et le corps i , on décompose $\overrightarrow{O_{i-1}O_i}$ en $\overrightarrow{O_{i-1}O_i^r} + \overrightarrow{O_i^rO_i}$.

Quel que soit l'axe i du robot ($1 \leq i \leq n$), on peut aisément exprimer les coordonnées des vecteurs $\overrightarrow{O_{i-1}O_i}_{/R_{i-1}}$. D'autre part, on sait également exprimer les matrices de passage ${}^{i-1}\mathbf{M}_i$ en fonction des q_i .

On écrit alors :

$$\begin{aligned} \overrightarrow{O_0 P}_{/R_0} &= \overrightarrow{O_0 O_1}_{/R_0} + \overrightarrow{O_1 O_2}_{/R_0} + \overrightarrow{O_2 O_3}_{/R_0} + \dots + \overrightarrow{O_{n-1} O_n}_{/R_0} + \overrightarrow{O_n P}_{/R_0} \\ &= \overrightarrow{O_0 O_1}_{/R_0} + {}^0\mathbf{M}_1 \overrightarrow{O_1 O_2}_{/R_1} + {}^0\mathbf{M}_1 {}^1\mathbf{M}_2 \overrightarrow{O_2 O_3}_{/R_2} + \dots + {}^0\mathbf{M}_1 {}^1\mathbf{M}_2 \dots {}^{n-2}\mathbf{M}_{n-1} \overrightarrow{O_{n-1} O_n}_{/R_{n-1}} + {}^0\mathbf{M}_1 \dots {}^{n-1}\mathbf{M}_n \overrightarrow{O_n P}_{/R_n} \end{aligned}$$

Soit :

$$\boxed{\overrightarrow{O_0 P}_{/R_0} = \overrightarrow{O_0 O_1}_{/R_0} + \dots + {}^0\mathbf{M}_{i-1} \overrightarrow{O_{i-1} O_i}_{/R_{i-1}} + \dots + {}^0\mathbf{M}_{n-1} \overrightarrow{O_{n-1} O_n}_{/R_{n-1}} + {}^0\mathbf{M}_n \overrightarrow{O_n P}_{/R_n}}$$

Avec les matrices de transformation homogène

En calculant ${}^0\mathbf{T}_n = \prod_{i=1}^n {}^{i-1}\mathbf{T}_i$, on obtient :

$${}^0\mathbf{T}_n = \begin{bmatrix} {}^0\mathbf{M}_n & \overrightarrow{O_0 O_n}_{/R_0} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

On peut alors obtenir directement les coordonnées du point P dans le repère R_0 :

$$\boxed{\begin{bmatrix} \overrightarrow{O_0 P}_{/R_0} \\ 1 \end{bmatrix} = {}^0\mathbf{T}_n \cdot \begin{bmatrix} \overrightarrow{O_n P}_{/R_n} \\ 1 \end{bmatrix}}$$

Cette méthode est intéressante de par sa « compacité » : il ne s'agit que de multiplier des matrices.

3.1.5 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 (ou de Denavit-Hartenberg modifié, qui permet également la prise en compte des robots à chaîne cinématique complexe). Cette convention fournit pour chaque axe la matrice de transformation homogène correspondante par la connaissance de quatre paramètres. Cette méthode peut être intéressante pour « systématiser » les calculs du modèle géométrique.

Le choix des repères est le suivant :

- l'axe \vec{z}_i correspond à l'axe de l'articulation i ,
- l'axe \vec{x}_i est porté par la perpendiculaire commune aux axes \vec{z}_i et \vec{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.

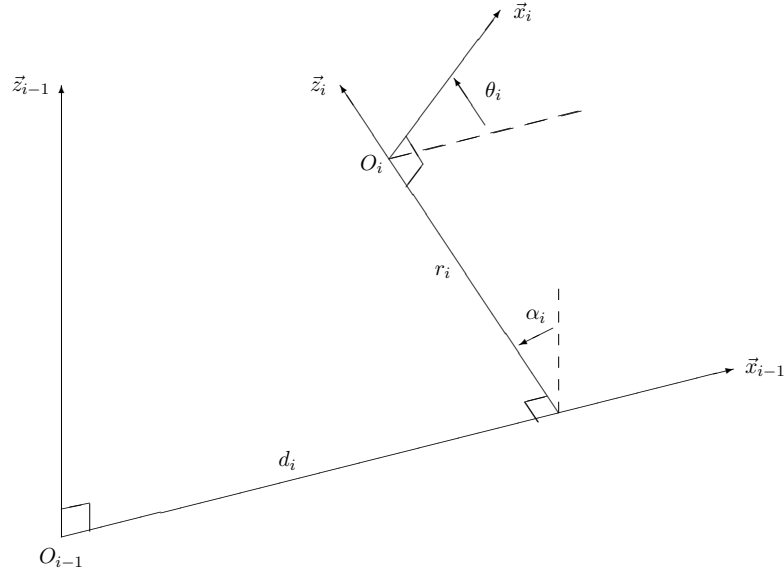
Le passage de R_{i-1} à R_i s'exprime en fonction des quatre paramètres de Denavit-Hartenberg modifié suivants :

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

La variable articulaire q_i est soit θ_i , soit r_i , selon que l'articulation i est rotoïde ou prismatique.

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

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

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

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 tout en remplissant un tableau avec les paramètres utiles.

Articulation	σ_i	α_i	d_j	θ_i	r_i
1					
2					
...					
n					

(avec $\sigma_i = 0$ si l'articulation est rotoïde,
 $\sigma_i = 1$ si l'articulation est prismatique)

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

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

Ces calculs peuvent être automatisés.

Ce type de représentation permet de traiter tous les cas de figures, y compris ceux des robots « bicornus ». Cela n'est pourtant apparemment pas le cas des robots industriels, mais, lorsque l'on veut obtenir par identification le modèle géométrique très précis d'un robot (notamment pour en améliorer sa commande), c'est à dire prendre en compte les défauts de fabrication (décalages divers, non-coaxialités, défauts angulaires), une représentation générique de type Denavit-Hartenberg s'avère très utile.

3.1.6 Synthèse

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 ${}^0\mathbf{T}_P$.

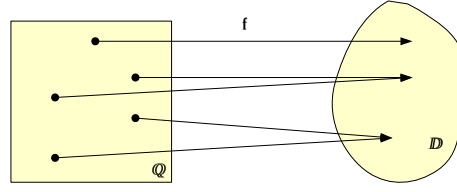


FIGURE 3.5 – Illustration du lien entre espace des configurations et domaine atteignable.

En multipliant la matrice ${}^0\mathbf{T}_n$ par la matrice de transformation homogène associée à l'effecteur, notée ${}^n\mathbf{T}_P$, dans laquelle la matrice de rotation est la matrice identité. On obtient alors :

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

Cette matrice ${}^0\mathbf{T}_P$ exprime d'un seul coup la rotation de l'effecteur par ${}^0\mathbf{M}_P$ et sa position par $\overrightarrow{O_0P}/_{R_0}$.

3.2 Domaine atteignable

Définition 4. 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.

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 5. 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 postures 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 :

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

3.3 Modèle géométrique inverse d'un robot

Comme indiqué dans le chapitre précédent, 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, en calculant $\mathbf{q}_d = \mathbf{f}^{-1}(\mathbf{x}_d)$ (à condition que l'on puisse inverser le modèle géométrique).

3.3.1 Définition

Définition 6. On appelle modèle géométrique inverse (MGI), l'application \mathbf{f}^{-1} de \mathbb{R}^6 dans \mathbb{R}^n exprimant \mathbf{q} en fonction de \mathbf{x} :

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x})$$

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.

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 est donc souvent difficile, voire parfois quasi-impossible, de l'inverser directement. Néanmoins, pour la plupart des robots industriels, cette opération reste faisable.

Plusieurs méthodes existent pour obtenir le MGI. Parmi celles-ci, on peut citer les cinq classes de méthodes suivantes :

- les méthodes géométriques, permettant d'obtenir chaque variable articulaire à partir de considérations géométriques,
- les méthodes générales, utilisant le MGD et s'adaptant à la plupart des robots,
- les méthodes spécifiques aux robots à six degrés de liberté à poignet sphérique (trois articulations rotoïdes d'axes concourants),
- les méthodes spécifiques pour des robots à « cinématique particulière ».

3.3.2 Nombre de solutions possibles

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 situation opérationnelle donnée (\mathbf{x} donné). La quasi-totalité des robots industriels est résoluble.

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

- il n'y a aucune solution permettant de satisfaire au \mathbf{x} souhaité, car celui-ci est en dehors du domaine atteignable du robot : dans ce cas, le robot n'est pas adapté à la tâche ou mal positionné dans la cellule ;
- 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), générant ainsi une redondance locale au prix de la perte d'un ou plusieurs ddl. Cette situation n'est pas directement imputable à la structure du robot, mais aux valeurs numériques particulières prises par les coordonnées de \mathbf{x} ;

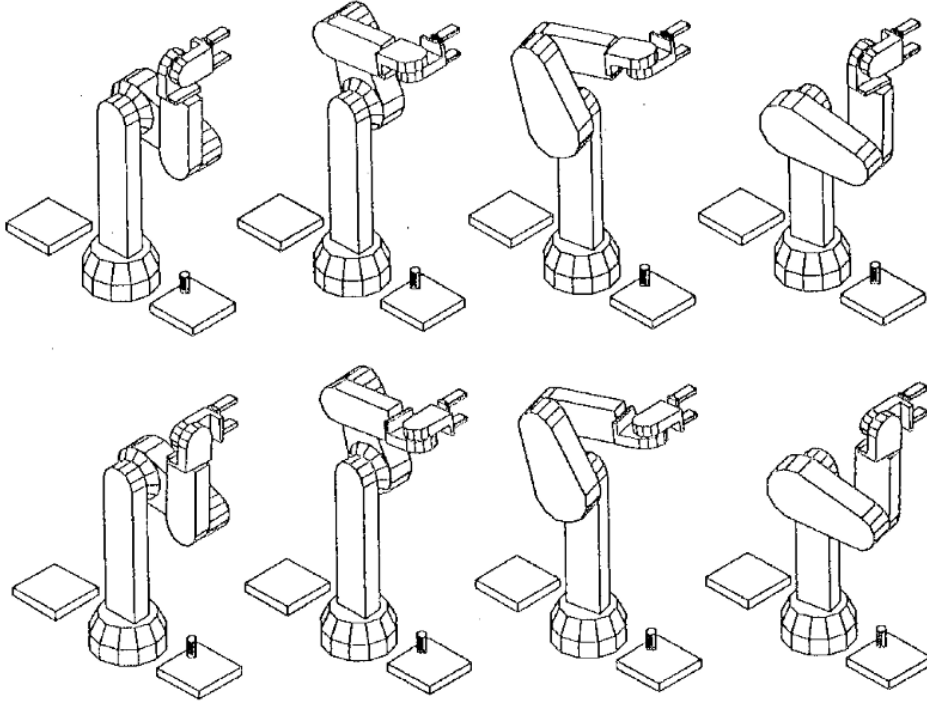


FIGURE 3.6 – Exemple de solutions du MGI pour un robot 6R.

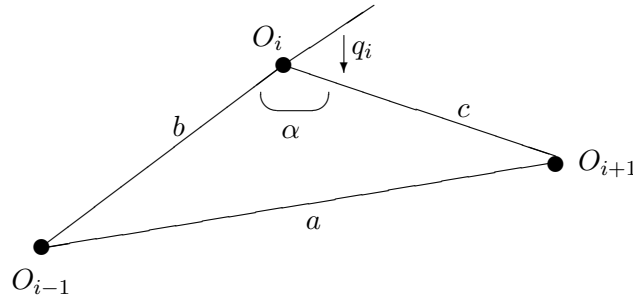
- il y a un nombre fini de solutions. Ce nombre dépend de la structure cinématique du robot. Par exemple pour les robots à 6 ddl et à poignet sphérique, il y a au maximum 32 solutions. La figure 3.6 donne les 8 solutions du MGI d'un robot 6R (6 articulations rotoïdes).

3.3.3 Méthode géométrique

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 rotoïde d'un robot, comme indiquée sur le schéma suivant. En supposant connues les longueurs a , b et c , on peut calculer l'angle α par la relation :

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



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

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

3.3.4 Méthode générale d'inversion

La méthode générale d'inversion a été proposée par Paul en 1981 [8].

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 :

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

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

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

Le terme de gauche, noté matrice \mathbf{M}_G , n'est fonction que de q_1 et le terme de droite, noté matrice \mathbf{M}_D , est fonction de $q_2 \dots q_n$. On peut alors obtenir q_1 par identification d'un ou deux éléments judicieusement choisis des deux matrices \mathbf{M}_G et \mathbf{M}_D .

Une fois q_1 calculé, on écrit :

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

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

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

Cette méthode est générale mais demande un peu de pratique. 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 par exemple [4]). A la main, elle est souvent plus complexe à mettre en œuvre pour les robots étudiés que la méthode géométrique.

3.3.5 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 rotoïdes d'axes concourants correspondent à une liaison rotule. Un tel poignet est donc équivalent à un poignet sphérique.

Pour avoir un robot à 6 ddl, le porteur (en excluant le poignet) doit donc posséder également 3 articulations.

Pour de tels robots, on peut alors obtenir de manière explicite le MGI. Le schéma suivant 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.

Les données du problème sont le vecteur $\overrightarrow{O_0P}/_{R_0}$ et l'orientation du repère R_P dans R_0 , représentée par la matrice de passage ${}^0\mathbf{M}_P$ (définie par exemple par les angles nautiques).

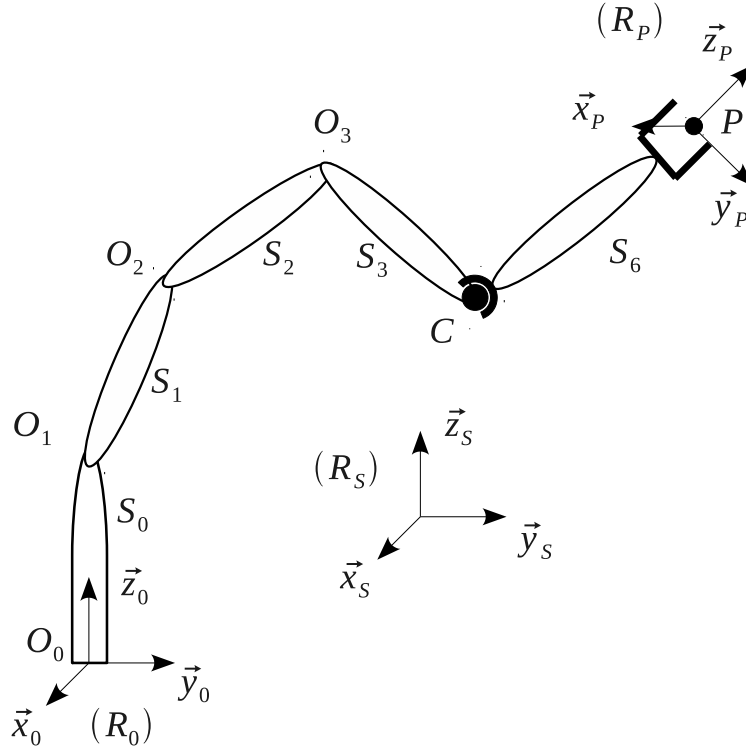


FIGURE 3.7 – Conventions de repérage pour l'inversion des robots à 6 ddl à poignet sphérique.

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 :

$$\overrightarrow{O_0C}/_{R_0} = \overrightarrow{O_0P}/_{R_0} + \overrightarrow{PC}/_{R_0} = \overrightarrow{O_0P}/_{R_0} + {}^0\mathbf{M}_P \overrightarrow{PC}/_{R_P}$$

$\overrightarrow{O_0P}/_{R_0}$, ${}^0\mathbf{M}_P$ et $\overrightarrow{PC}/_{R_P}$ sont connus, donc on peut calculer $\overrightarrow{O_0C}/_{R_0}$.

Résolution du porteur : calcul de q_1 , q_2 et q_3

Connaissant $\overrightarrow{O_0C}/_{R_0}$, on peut résoudre le porteur (par exemple par une méthode géométrique) 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).

NB: $\overrightarrow{O_0C}/_{R_0} = \mathbf{g}(q_1, q_2, q_3)$ correspond à un système de trois équations à trois inconnues.

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 .

Résolution du poignet : calcul de q_4 , q_5 et q_6

Les angles q_1 , q_2 et q_3 étant connus, 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 ${}^0\mathbf{M}_P$, on peut alors calculer ${}^3\mathbf{M}_P$ par la relation :

$${}^3\mathbf{M}_P = {}^3\mathbf{M}_0. {}^0\mathbf{M}_P = ({}^0\mathbf{M}_3)^T. {}^0\mathbf{M}_P$$

En égalisant les deux matrices ${}^3\mathbf{M}_P$, on peut alors en déduire q_4 , q_5 et q_6 .

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).

3.3.6 Méthodes d'inversion pour des robots à « cinématique particulière »

Rappelons que le MGI peut être vu comme la résolution de 6 équations à n inconnues (q_1 à q_n).

Robots redondants

Lorsque $n > 6$, c'est à dire lorsque le robot est redondant, le système à résoudre contient plus d'inconnues que d'équations. Il y a donc une infinité de solutions possibles.

On peut alors fixer arbitrairement $(n-6)$ variables articulaires ou introduire $(n-6)$ relations supplémentaires. Ces choix peuvent être dictés par la tâche à effectuer ou par la morphologie du robot.

Robots à nombre d'articulations inférieur à 6

Dans le cas où $n < 6$, le robot ne peut pas générer les 6 ddl de l'espace. Le MGD fait donc apparaître $(6 - n)$ équations de contraintes entre les 6 coordonnées de \mathbf{x} , interdisant ainsi de les choisir indépendamment.

3.4 Conclusion

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

Le MGI, par rapport aux méthodes numériques, 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 4

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 .

4.1 Définitions

4.1.1 Modèle cinématique direct

Définition 7. 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.

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

NB: La matrice jacobienne dépend de \mathbf{q} . Elle varie donc dans le temps en fonction des mouvements du robot.

4.1.2 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.

4.2 Jacobienne analytique

La matrice jacobienne peut s'obtenir en différentiant le modèle géométrique représentée par la fonction vectorielle $\mathbf{f}(\mathbf{q})$.

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

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

En différentiant cette expression, on obtient :

$$\dot{\mathbf{x}} = {}^0\mathbf{J}^a(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

avec ${}^0\mathbf{J}^a(\mathbf{q})$ la matrice jacobienne :

$${}^0\mathbf{J}^a(\mathbf{q}) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_d}{\partial q_1} & \cdots & \frac{\partial f_d}{\partial q_n} \end{bmatrix}$$

Cette matrice est appelée matrice jacobienne *analytique* (ou différentielle) 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 ${}^0\mathbf{J}^a(\mathbf{q})$ est donc aisée mais peut être coûteux en temps de calcul. Il faut également faire attention à l'instabilité numérique des algorithmes mis en œuvre.

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 :

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

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.

4.3 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 $\dot{\mathbf{q}}$.

4.3.1 Cinématique du point

La vitesse instantanée d'un point P dans le repère R_j est définie par la dérivée du vecteur position :

$${}^j\mathbf{v}(P) = \frac{d}{dt} \overrightarrow{O_j P}_{/R_j}$$

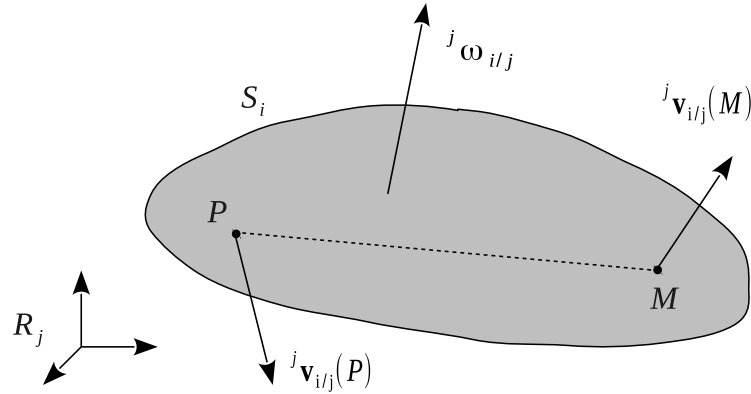


FIGURE 4.1 – Illustration de la formule de Varignon.

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

4.3.2 Cinématique du solide

Champ de vitesse

Le champ de vitesse d'un solide S_i par rapport à un repère R_j est l'application qui associe, à tout point P de S_i , la vitesse de ce point exprimée dans le repère R_j et telle que :

$$j\mathbf{v}_{i/j}(P) = \frac{d}{dt} \overrightarrow{O_j P}_{/R_j}$$

On dit que $j\mathbf{v}_{i/j}(P)$ est la vitesse instantanée du point P dans le mouvement de S_i par rapport à R_j et exprimée dans le repère R_j .

Formule de Varignon

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

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

où $j\boldsymbol{\omega}_i$ est la vitesse de rotation instantanée du solide S_i par rapport à R_j .

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

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

Torseur cinématique

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

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

où :

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

Composition des vitesses

Soit S_i , S_j et S_k trois solides et P un point, la vitesse absolue est égale à la somme de la vitesse relative et de la vitesse d'entraînement :

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

La vitesse relative ${}^j\mathbf{v}_{i/k}(P)$ est la vitesse de P dans le mouvement de S_i par rapport à S_k exprimée dans le repère R_j .

La vitesse d'entraînement ${}^j\mathbf{v}_{k/j}(P)$ est la vitesse de P dans le mouvement de S_k par rapport à S_j (c'est-à-dire s'il était fixe dans le repère R_k) exprimée dans le repère R_j .

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

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

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

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

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

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

4.3.3 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 et la valeur de l'angle ou de la translation est noté q_i (cf. figures 3.2 et 3.3).

Articulation rotoïde (liaison pivot)

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

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

Comme la vitesse du centre de rotation O_i dans le repère R_{i-1} est nulle, la vitesse d'un point P lié au solide S_i dans le repère R_{i-1} s'écrit :

$$\begin{aligned} {}^{i-1}\mathbf{v}_{i/i-1}(P) &= {}^{i-1}\mathbf{v}_{i/i-1}(O_i) + \overrightarrow{PO_i}_{/R_{i-1}} \wedge {}^{i-1}\boldsymbol{\omega}_{i/i-1} \\ &= {}^{i-1}\boldsymbol{\omega}_{i/i-1} \wedge \overrightarrow{O_iP}_{/R_{i-1}} \\ &= ({}^{i-1}\mathbf{u}_i \cdot \dot{\mathbf{q}}_i) \wedge \overrightarrow{O_iP}_{/R_{i-1}} \end{aligned}$$

Le vecteur $\overrightarrow{O_iP}$ étant plus facile à exprimer dans le repère R_i , on peut réécrire l'équation en utilisant la matrice de rotation entre R_{i-1} et R_i . On obtient :

$$\boxed{{}^{i-1}\mathbf{v}_{i/i-1}(P) = ({}^{i-1}\mathbf{u}_i \cdot \dot{\mathbf{q}}_i) \wedge ({}^{i-1}\mathbf{M}_i \cdot \overrightarrow{O_iP}_{/R_i})}$$

Ainsi le torseur cinématique pour une articulation rotoïde s'écrit :

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

Articulation prismatique (liaison glissière)

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

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

La vitesse du point O_i dans le repère R_{i-1} est par définition :

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

La vitesse d'un point P lié au solide S_i dans le repère R_{i-1} s'écrit également :

$${}^{i-1}\mathbf{v}_{i/i-1}(P) = {}^{i-1}\mathbf{v}_{i/i-1}(O_i) + \overrightarrow{PO_i}_{/R_{i-1}} \wedge {}^{i-1}\boldsymbol{\omega}_{i/i-1}$$

Donc :

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

Soit sous forme d'un torseur :

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

4.3.4 Calcul de la matrice jacobienne cinématique

Le mouvement de l'organe terminal est complètement défini par le torseur cinématique ${}^0\mathcal{V}_{n/0}(P)$. On cherche donc à établir la relation entre la vitesse de l'organe terminal et les positions et vitesses articulaires \mathbf{q} et $\dot{\mathbf{q}}$ sous la forme suivante :

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

${}^0\mathbf{J}^c(\mathbf{q})$ est la matrice jacobienne dite *cinématique* (ou encore jacobienne de base) pour ne pas la confondre avec la matrice jacobienne analytique.

Pour calculer le torseur ${}^0\mathcal{V}_{n/0}(P)$, on peut utiliser les lois de composition. On écrit :

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

On exprime alors chaque torseur ${}^0\mathcal{V}_{i/i-1}(P)$ à l'aide de la matrice de changement de base ${}^0\mathcal{M}_{i-1}$ de R_0 à R_{i-1} telle que :

$${}^0\mathcal{M}_{i-1} = \begin{bmatrix} {}^0\mathbf{M}_{i-1} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^0\mathbf{M}_{i-1} \end{bmatrix}$$

qui permet d'écrire :

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

On obtient enfin :

$$\begin{aligned} {}^0\mathcal{V}_{n/0}(P) = {}^0\mathcal{M}_{n-1} \cdot {}^{n-1}\mathcal{V}_{n/n-1}(P) + {}^0\mathcal{M}_{n-2} \cdot {}^{n-2}\mathcal{V}_{n-1/n-2}(P) + \cdots \\ + {}^0\mathcal{M}_1 \cdot {}^1\mathcal{V}_{2/1}(P) + {}^0\mathcal{M}_0 \cdot {}^0\mathcal{V}_{1/0}(P) \end{aligned}$$

Soit :

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

4.4 Passage d'une jacobienne à l'autre

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 [10]).

4.5 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 :

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

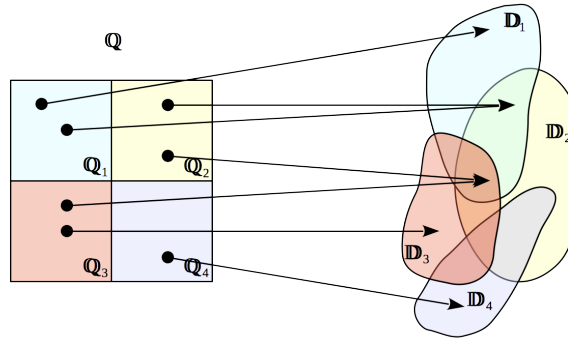
Pour représenter graphiquement les vitesses atteignables du robot dans une certaine configuration, on utilise les *ellipsoïdes de manipulabilité*. Dans l'espace des vitesses articulaires, on considère l'ensemble des vecteurs $\dot{\mathbf{q}}$ unitaires, c'est-à-dire l'ensemble des vecteurs vitesses 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 un ellipsoïde qui est représentative des vitesses potentielles du robot dans chaque direction de l'espace opérationnel. Les axes principaux de l'ellipsoïde correspondent

FIGURE 4.2 – Illustration des aspects Q_i et de leur image D_i .

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 une vitesse. 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.

4.6 Aspects et configurations singulières

Les matrices jacobiennes permettent également de connaître les différentes de postures d'un robot (aspects) et de détecter ses configurations singulières.

4.6.1 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 postures 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 :

Si dans un sous-domaine de \mathbb{Q} , noté \mathbb{Q}' , le déterminant de $\mathbf{J}(\mathbf{q})$ ($\mathbf{q} \in \mathbb{Q}'$) 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 \mathbb{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 \mathbb{D}_i , l'image de \mathbb{Q}_i . Le domaine atteignable \mathbb{D} peut être alors décrit comme la réunion de toutes les images des aspects \mathbb{Q}_i (cf. figure 4.2). On peut écrire les propriétés suivantes :

$$\mathbb{Q} = \bigcup_{i=1}^m \mathbb{Q}_i \quad ; \quad \mathbb{Q}_i \cap \mathbb{Q}_j = \emptyset \quad ; \quad \mathbb{D} = \bigcup_{i=1}^m \mathbb{D}_i \quad ; \quad \mathbb{D}_i \cap \mathbb{D}_j \neq \emptyset$$

Pour $\mathbf{q} \in \mathbb{Q}_i$, le robot est dit dans la posture n° i . La description précédente permet alors de connaître les solutions multiples, c'est-à-dire toutes les postures possibles pour un \mathbf{x} donné. En effet, si par exemple le vecteur opérationnel $\mathbf{x} \in \mathbb{D}_i \cap \mathbb{D}_j \cap \mathbb{D}_k$, il est l'image de trois vecteurs articulaires différents, appartenant à \mathbb{Q}_i , \mathbb{Q}_j et \mathbb{Q}_k .

4.6.2 Configurations singulières

D'après le chapitre précédent, on rappelle que le calcul du MGI peut conduire à une infinité de solutions dans deux cas :

- soit le robot est redondant vis-à-vis de la tâche à réaliser,
- soit le robot se trouve dans une certaine configuration, générant une indéterminée ou une redondance locale au prix de la perte d'un ou plusieurs ddl. Cette situation n'est pas directement imputable à la structure du robot, mais aux valeurs numériques particulières prises par les coordonnées de \mathbf{x} .

Dans ce dernier cas, la configuration correspondante est dite *singulière*.

L'analyse des c singulières repose donc, 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.

Le MCD n'est pas inversible si $\det \mathbf{J}(\mathbf{q}) = 0$. On se place alors dans ce cas et on définit r , le rang de $\mathbf{J}(\mathbf{q})$.

NB: On rappelle que le rang d'une matrice ($n \times p$) (n non nécessairement égal à p) est l'ordre r de la plus grande sous-matrice carrée ($r \times r$) extraite de la précédente et dont le déterminant est non nul. Pour qu'une matrice soit de rang r , il suffit donc qu'un seul des déterminants d'ordre r soit différent de zéro.

Dans le cas où $\det \mathbf{J}(\mathbf{q}) = 0$, on peut réarranger l'ordre des équations et des inconnues du MCD de telle sorte que Δ_r , le déterminant non nul d'ordre r , apparaisse « en haut à gauche » dans la nouvelle matrice \mathbf{J} :

$$\begin{matrix} & \overbrace{\quad}^r & \overbrace{\quad}^{n-r} \\ \begin{matrix} r \{ \\ n-r \{ \end{matrix} & \begin{bmatrix} \Delta_r & \times \\ \times & \times \end{bmatrix} & \cdot \dot{\mathbf{q}} = \dot{\mathbf{x}} \end{matrix}$$

On en déduit alors qu'il y a :

- $(n-r)$ indéterminées \dot{q}_i dans les r équations principales (celle « du haut »), d'où une redondance de certains q_i pour engendrer certains mouvements de l'effecteur ;
- $(n-r)$ relations de contraintes entre les \dot{x}_i dues aux $(n-r)$ équations non principales (celles « du bas ») qui doivent être vérifiées, d'où la perte de $(n-r)$ degrés de liberté de l'effecteur.

On peut donc dire qu'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 dextérité du robot sur ces mouvements. D'une manière générale, les configurations singulières sont

plutôt à éviter, mais pour certaines tâches nécessitant de la dextérité (certains assemblages par exemple), elles peuvent être intéressantes.

On peut également noter qu'il est généralement préférable, dans le cadre de l'utilisation d'algorithmes de type « commande incrémentale » présentée plus loin, de ne pas générer de trajectoires passant « trop près » des configurations singulières, afin d'éviter les problèmes de stabilité numérique, voire de divergence.

D'autre part, lorsqu'un robot s'approche d'une configuration singulière, $\det \mathbf{J}$ devient de plus en plus faible, donc les termes de \mathbf{J}^{-1} 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.

Si l'on souhaite que le robot passe quand même près ou par des configurations singulières, il faut « sophistiquer » le transformateur de coordonnées.

4.6.3 Configurations singulières d'un robot anthropomorphe à poignet sphérique

Singularité du poignet

Soit la configuration du poignet sphérique représentée par la figure 4.3

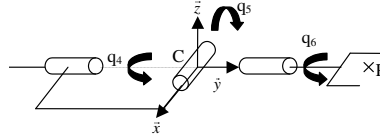


FIGURE 4.3 – Singularité du poignet sphérique

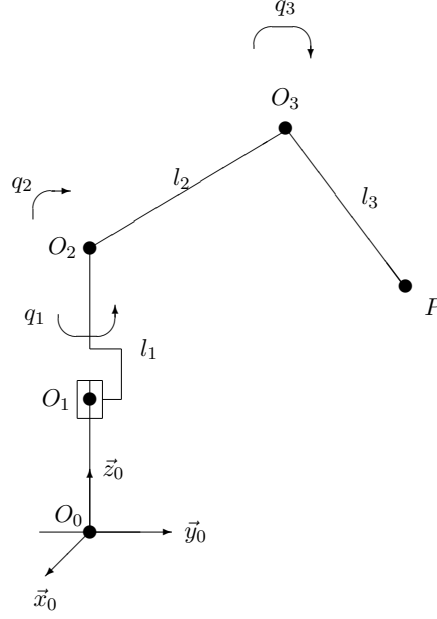
Dans cette situation, il apparaît clairement que q_4 et q_6 sont redondants pour engendrer la rotation autour de l'axe \vec{y} et que la rotation autour de l'axe \vec{z} ne peut être engendrée. On retrouve bien entendu ce résultat par calcul de la jacobienne (analytique) du poignet, qui peut alors s'écrire :

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}$$

Dans cette configuration, la jacobienne est de rang 2.

Singularité du porteur

Soit le porteur à trois axes suivants :



Le modèle géométrique est :

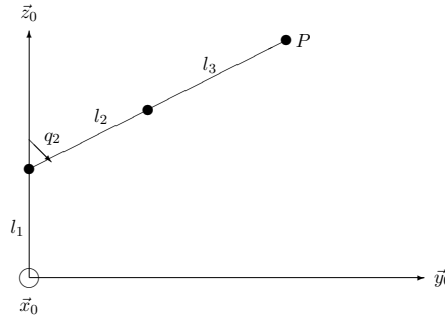
$$\begin{cases} x &= s_1(l_2 s_2 + l_3 s_{23}) \\ y &= -c_1(l_2 s_2 + l_3 s_{23}) \\ z &= l_1 + l_2 c_2 + l_3 c_{23} \end{cases}$$

La jacobienne s'écrit donc :

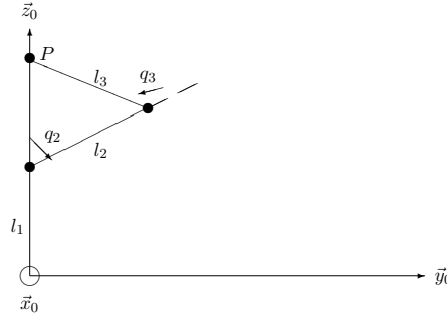
$$\mathbf{J} = \begin{bmatrix} c_1(l_2 s_2 + l_3 s_{23}) & s_1(l_2 c_2 + l_3 c_{23}) & l_3 s_1 c_{23} \\ s_1(l_2 s_2 + l_3 s_{23}) & -c_1(l_2 c_2 + l_3 c_{23}) & -l_3 c_1 c_{23} \\ 0 & -(l_2 s_2 + l_3 s_{23}) & -l_3 s_{23} \end{bmatrix}$$

$$\implies \det \mathbf{J} = l_2 l_3 s_3 (l_2 s_2 + l_3 s_{23})$$

Configurations singulières de type 1 : $q_3 = 0$ ou π : $\text{rang}(\mathbf{J})=2$, d'où la perte d'un ddl. On se ramène ici au cas des configurations singulières du robot plan à deux axes étudié précédemment.



Configurations singulières de type 2 : $l_2 c_2 + l_3 c_{23} = 0$: $\text{rang}(\mathbf{J})=2$, d'où la perte d'un ddl. Dans ce cas, le point P est situé sur l'axe \vec{z}_0 et l'angle q_1 devient inopérant pour modifier la position de P .



4.7 Conclusion

Le modèle cinématique direct donne les performances en vitesse du robot, ses positions singulières et le nombre de postures différentes (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 très difficile, voire impossible d'inverser analytiquement le modèle géométrique direct, on peut mettre en œuvre des méthodes numériques à l'aide de la matrice jacobienne inverse.

La connaissance du modèle cinématique permet également de calculer les efforts articulaires correspondant à un effort statique exercé sur un robot et de calculer les matrices de compliance d'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 5

Efforts et modélisation dynamique

Lorsqu'un robot se met en mouvement des effets dynamiques non linéaires et dépendant de la posture surviennent. Il est nécessaire de les connaître pour dimensionner les actionneurs et calculer les lois de commandes des axes.

5.1 Définition

Définition 8. 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 posture 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 ces méthodes, nous renvoyons le lecteur aux nombreux ouvrages de références en mécanique (voir par exemple [7]).

5.2 Modèle dynamique inverse

5.2.1 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 Γ_i est l'effort au niveau de l'articulation i .

5.2.2 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 utilisant le modèle cinématique, on a :

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

Donc :

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

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

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

Soit sous une forme plus compacte :

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

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

5.2.3 Calcul de l'énergie potentielle

L'énergie potentielle du segment i est définie par :

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

avec :

- m_i la masse du segment i ,
- g la constante universelle de gravitation,
- G_i le centre d'inertie du segment i .

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

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

5.2.4 Calcul du modèle dynamique

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

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

On calcule alors les différents termes des équations d'Euler-Lagrange. On a tout d'abord :

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

et donc :

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} = \left[\mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{d\mathbf{D}(\mathbf{q})}{dt} \dot{\mathbf{q}} \right]_i = \left[\mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}} + \left(\sum_{k=1}^n \frac{\partial \mathbf{D}(\mathbf{q})}{\partial q_k} \dot{q}_k \right) \dot{\mathbf{q}} \right]_i$$

Pour le second terme, on a :

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

Au final, les équations d'Euler-Lagrange peuvent s'écrire sous la forme suivante :

$$\mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Q}(\mathbf{q}) = \mathbf{\Gamma}$$

Où :

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

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

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

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

— le terme $\mathbf{Q}(\mathbf{q})$ désigne les contributions de la gravité,

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} \frac{\partial \mathcal{P}(\mathbf{q})}{\partial q_1} \\ \vdots \\ \frac{\partial \mathcal{P}(\mathbf{q})}{\partial q_n} \end{bmatrix} = \sum_{i=1}^n m_i g \begin{bmatrix} \frac{\partial z_{G_i/R_0}}{\partial q_1} \\ \vdots \\ \frac{\partial z_{G_i/R_0}}{\partial q_n} \end{bmatrix}$$

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

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

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

$$\boxed{\mathbf{\Gamma}_m = \mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Q}(\mathbf{q}) + \mathbf{\Gamma}_f(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{\Gamma}_e}$$

5.3 Efforts appliqués par le robot sur son environnement

Les efforts d'interaction $\mathbf{\Gamma}_e$ 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 :

$${}^0\mathcal{T}_e(P) = [f_x \ f_y \ f_z \ m_x \ m_y \ m_z]^T$$

Le principe des travaux virtuels implique que :

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

or :

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

avec ${}^0\mathbf{J}_n^c(P, \mathbf{q})$ la matrice jacobienne cinématique du segment S_n au point d'application P de la force extérieure.

Donc :

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

5.4 Efforts dus à la pesanteur et équilibrage statique

Le modèle dynamique permet de calculer les couples moteurs nécessaires pour contrecarrer la force de pesanteur.

En statique et sans interaction avec l'environnement, le modèle dynamique se réduit à :

$$\mathbf{\Gamma}_m = \mathbf{Q}(\mathbf{q})$$

Le vecteur $\mathbf{Q}(\mathbf{q})$ peut être calculé par la méthode générale de Lagrange ou plus simplement en écrivant le principe des travaux virtuels pour chaque segment.

Soit ${}^0\mathcal{T}_i(P)$ le torseur des efforts par le robot pour contrecarrer le poids du solide S_i exprimé dans le repère R_0 :

$${}^0\mathcal{T}_i(P) = [0 \ 0 \ m_i g \ 0 \ 0 \ 0]^T$$

Comme précédemment, en appliquant le principe des travaux virtuels, il vient :

$$\mathbf{\Gamma}_{m,i} = [{}^0\mathbf{J}_i^c(G_i, \mathbf{q})]^T \cdot {}^0\mathcal{T}_i(P)$$

avec ${}^0\mathbf{J}_i^c(G_i, \mathbf{q})$ la matrice jacobienne cinématique du segment S_i au point G_i .

Comme seule la troisième ligne du torseur ${}^0\mathcal{T}_i(P)$ est non nulle, on a :

$$\mathbf{\Gamma}_{m,i} = \begin{bmatrix} m_i g \frac{\partial z_{G_i/R_0}}{\partial q_1} \\ \vdots \\ m_i g \frac{\partial z_{G_i/R_0}}{\partial q_i} \\ 0 \\ \vdots \end{bmatrix}$$

avec $z_{G_i/R_0} = \vec{z}_0 \cdot \overrightarrow{O_0 G_i}(\mathbf{q})_{/R_0}$.

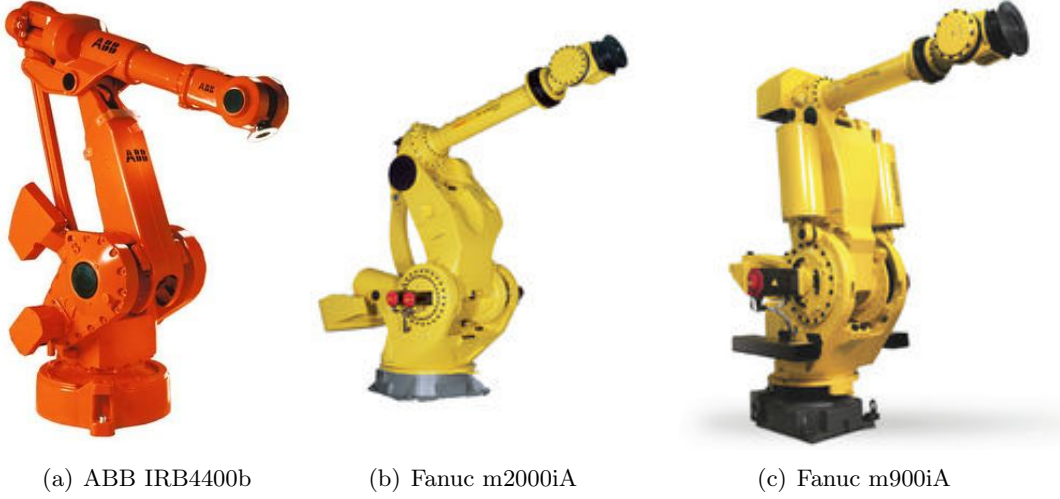


FIGURE 5.1 – Illustration de l'équilibrage statique.

En effectuant cette démarche sur tous les corps constitutifs du robot, il vient :

$$\mathbf{Q}(\mathbf{q}) = \mathbf{\Gamma}_m = \begin{bmatrix} m_1 g \frac{\partial z_{G_1/R_0}}{\partial q_1} + m_2 g \frac{\partial z_{G_2/R_0}}{\partial q_1} + \dots + m_n g \frac{\partial z_{G_n/R_0}}{\partial q_1} \\ m_2 g \frac{\partial z_{G_2/R_0}}{\partial q_2} + \dots + m_n g \frac{\partial z_{G_n/R_0}}{\partial q_2} \\ \vdots \\ m_n g \frac{\partial z_{G_n/R_0}}{\partial q_n} \end{bmatrix}$$

La connaissance des couples nécessaires pour contrecarrer 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 (contrepoids) ou par des ressorts comme l'illustre la figure 5.1.

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

La méthode de calcul de $\mathbf{Q}(\mathbf{q})$ peut également être employée pour mesurer la masse et de la position du centre de masse d'une charge utile. Cette mesure permet notamment de réajuster les asservissements des axes en fonction de la charge.

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 étant à l'équilibre nous avons :

$$\mathbf{\Gamma}_m = \mathbf{Q}(\mathbf{q}_0)$$

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

$$\mathbf{\Gamma}'_m = \mathbf{Q}'(\mathbf{q}_0)$$

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 :

$$\mathbf{\Gamma}'_m - \mathbf{\Gamma}_m = \begin{bmatrix} Mg \frac{\partial z_{G_M/R_0}}{\partial q_1} \\ \vdots \\ Mg \frac{\partial z_{G_M/R_0}}{\partial q_n} \end{bmatrix}$$

où M est la masse de la charge et G_M son centre de masse.

Cette relation fournit n équations pour 4 inconnues, on en déduit alors M et les trois coordonnées de G_M .

5.5 Conclusion

Le modèle dynamique d'un robot constitue un outil de dimensionnement afin de choisir les actionneurs adéquats, permettant de générer les bons couples et forces moteurs pour permettre au robot d'atteindre les performances spécifiées par un cahier des charges.

Il permet également de réaliser la commande à couple calculé présentée au chapitre 7.

Chapitre 6

Génération de mouvement

La génération de mouvement a pour objectif de calculer les consignes en position, vitesse ou 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.

6.1 Programmation d'un robot

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

6.1.1 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 *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.

6.1.2 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.

6.1.3 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.

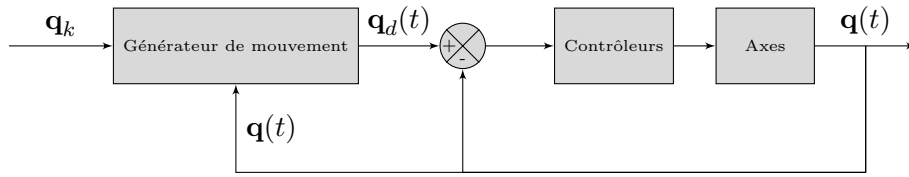


FIGURE 6.1 – Génération de mouvement dans l'espace articulaire (\mathbf{q}_k désigne la position finale à atteindre, $\mathbf{q}_d(t)$ la position désirée et $\mathbf{q}(t)$ la position courante des axes).

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 (trajectoire rectiligne, circulaire, ajout de points de passage, etc.).

Ces instructions « classiques » 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 et en évitant les obstacles (prise-dépose, soudure par point, etc.), ces mouvements sont générés dans l'espace articulaire,
- des mouvements de type « interpolés », où l'on souhaite au contraire maîtriser toute la trajectoire de l'effecteur, avec une loi horaire donnée (suivi de contour, dépôt de joint, dépôt de colle, soudure en continu, découpe, ébavurage, usinage, etc.), ces mouvements sont générés dans l'espace opérationnel,
- 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 d'asservissement externe (compliance active pour l'insertion, asservissement d'effort pour le suivi de contour, etc.)

6.2 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, notées \mathbf{x}_k (position et orientation) à atteindre par l'effecteur du robot. A chaque situation opérationnelle \mathbf{x}_k correspond la situation articulaire \mathbf{q}_k . 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 6.1. 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. On peut notamment citer l'interpolation polynomiale (de degré trois ou cinq), la loi *bang-bang*

avec ou sans palier de vitesse. Dans certains cas, on peut également envisager de générer des trajectoires plus douces, à jerk constant, voire d'ordre plus élevé.

Nous ne développerons ici que la loi bang-bang avec palier de vitesse qui est utilisé dans la plupart des contrôleurs industriels.

6.2.1 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 Δq_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 6.2). 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 6.2). 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.

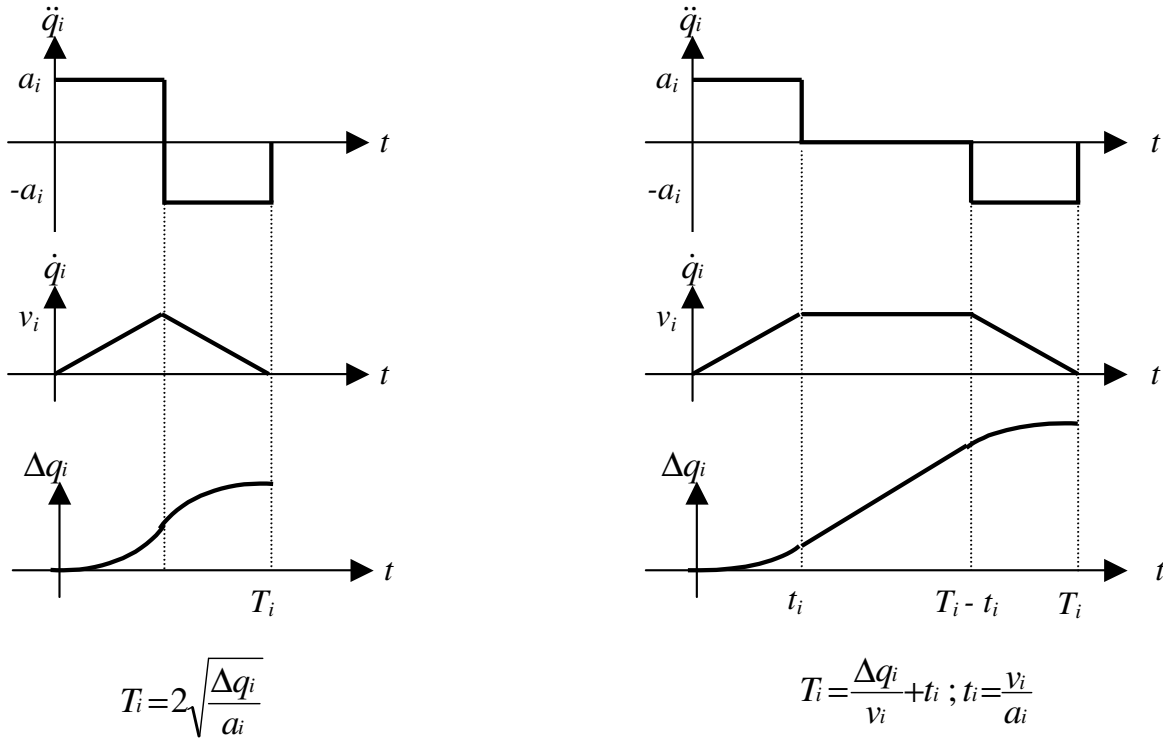


FIGURE 6.2 – Loi bang-bang sans (à gauche) ou avec palier de vitesse (à droite).

6.2.2 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.

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

On associe à chaque articulation un coefficient de réduction ν_i inférieur 1. L'accélération de l'axe i est ainsi $\nu_i a_i$. Ainsi, si l'axe i suit une loi bang-bang, son accélération $\nu_i a_i$ est telle que :

$$\nu_i a_i = \frac{4\Delta q_i}{T_{max}^2}$$

et donc :

$$\nu_i = \frac{4\Delta q_i}{a_i T_{max}^2}$$

avec T_{max} le temps de déplacement de l'axe le plus lent.

De même, si l'axe i suit une loi bang-bang avec palier de vitesse, sa nouvelle accélération est :

$$\nu_i a_i = \frac{v_i^2}{v_i T_{max} - \Delta q_i}$$

et donc :

$$\nu_i = \frac{v_i^2}{(v_i T_{max} - \Delta q_i) a_i}$$

L'algorithme 1 résume la génération de loi bang-bang avec palier de vitesse et synchronisation des axes.

Algorithme 1 : Algorithme de génération de loi bang-bang avec palier de vitesse et synchronisation 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

pour *chaque articulation i faire*

$\Delta q_i \leftarrow q_{k+1,i} - q_{k,i}$

si $\Delta q_i \leq \frac{v_i^2}{a_i}$ **alors**

$T_i \leftarrow 2\sqrt{\frac{\Delta q_i}{a_i}}$

sinon

$T_i \leftarrow \frac{\Delta q_i}{v_i} + \frac{v_i}{a_i}$

fin

fin

$T_{max} \leftarrow \max_{1 \leq i \leq n} T_i$

pour *chaque articulation i faire*

si $\Delta q_i \leq \frac{v_i^2}{a_i}$ **alors**

$\nu_i \leftarrow \frac{4\Delta q_i}{a_i T_{max}^2}$;

$t_i \leftarrow \frac{T_{max}}{2}$

sinon

$\nu_i \leftarrow \frac{v_i^2}{(v_i T_{max} - \Delta q_i) a_i}$;

$t_i \leftarrow \frac{v_i}{\nu_i a_i}$

fin

fin

Résultat : T_{max} , ν_i et t_i pour chaque axe i

6.2.3 Synchronisation des axes du robot avec lois homothétiques

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. En revanche, dans la plupart des cas, le déplacement alors obtenu n'est pas minimal.

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 ν_i et λ_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 6.2). 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 :

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

Algorithme 2 : 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
 $T_{max} \leftarrow \frac{\Delta q_1}{\lambda_1 \nu_1} + \frac{\lambda_1 \nu_1}{\nu_1 a_1};$ 
si  $\frac{\lambda_1 \nu_1}{\nu_1 a_1} < \frac{T_{max}}{2}$  alors
     $t \leftarrow \frac{\lambda_1 \nu_1}{\nu_1 a_1};$ 
sinon
     $t \leftarrow \frac{T_{max}}{2};$ 
fin

```

fin

Résultat : T_{max} , t , ν_i et λ_i pour chaque axe i

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 :

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

Le temps de déplacement T est minimal quand ν_1 et λ_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 $\frac{\lambda_1 \nu_1}{\nu_1 a_1} \geq \frac{T_{max}}{2}$, on fixera alors t à $\frac{T_{max}}{2}$.

L'algorithme 2 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.

6.2.4 Synthèse

Une fois calculés les coefficients ν_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 :

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

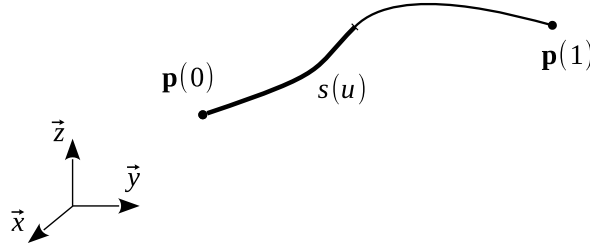


FIGURE 6.3 – Exemple d'arc paramétré définissant la trajectoire de l'organe terminal dans l'espace opérationnel.

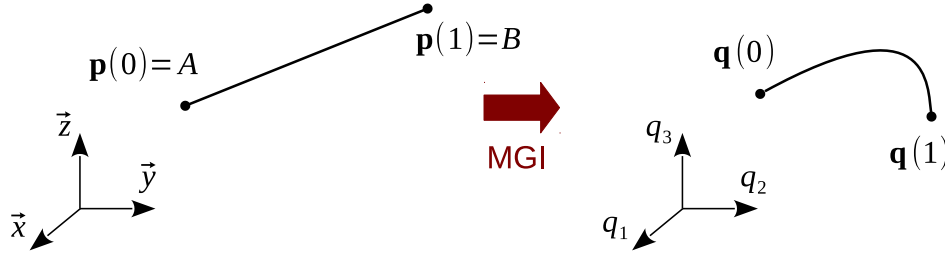


FIGURE 6.4 – Illustration de la génération d'une trajectoire rectiligne de l'organe terminal.

6.3 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.

6.3.1 Définition de la trajectoire

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 :

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

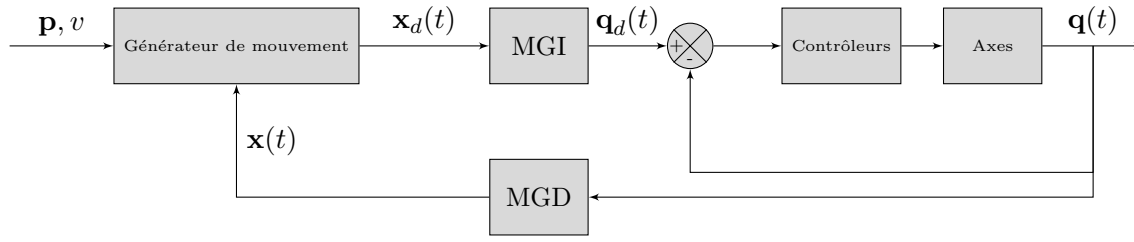


FIGURE 6.5 – Génération de mouvement dans l'espace opérationnel (\mathbf{x}_k désigne la position finale à atteindre, $\mathbf{x}_d(t)$ la position désirée et $\mathbf{x}(t)$ la position courante de l'organe terminale.

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

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

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 6.5.

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.

6.3.2 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 3).

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 :

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

Algorithme 3 : 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**

 Déterminer \mathbf{q}_{k+1} comme solution de l'équation :

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

 C'est-à-dire : $\mathbf{q}_{k+1} \leftarrow \mathbf{q}_k + [\mathbf{J}(\mathbf{q}_k)]^{-1} \cdot (\mathbf{x}^* - \mathbf{f}(\mathbf{q}_k))$;

fin

fin

Résultat : \mathbf{q}_{k+1}

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 :

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

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 postures 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.

6.3.3 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).

6.3.4 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 δq_i et calculer les autres en fonctions des δx_i en réduisant la matrice \mathbf{J} .

Le problème est alors que l'on n'a pas forcément de critère clair pour savoir quels δq_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 posture 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.

6.4 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 7

Commande des axes (dans l'espace articulaire)

La commande des axes dans l'espace articulaire a pour objectif d'asservir la position articulaire $\mathbf{q}(t)$ sur la trajectoire $\mathbf{q}_d(t)$ précédemment calculée.

Sur la plupart des robots industriels, les axes sont asservis séparément ainsi que le présente la section suivante, mais les effets négligés peuvent intervenir pour dégrader les performances des asservissements, notamment aux vitesses importantes (l'augmentation de la vitesse des axes a notamment pour effet l'augmentation de la contribution des efforts centrifuges). Il existe des lois de commande qui permettent de prendre en compte efficacement et sans approximation la dynamique totale du robot, mais celles-ci sont plus complexes et gourmandes en temps de calcul.

7.1 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. 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. Dans le cas des actionneurs électromagnétiques, des variateurs assurent la régulation du couple ou de la vitesse du moteur (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).
- 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.).

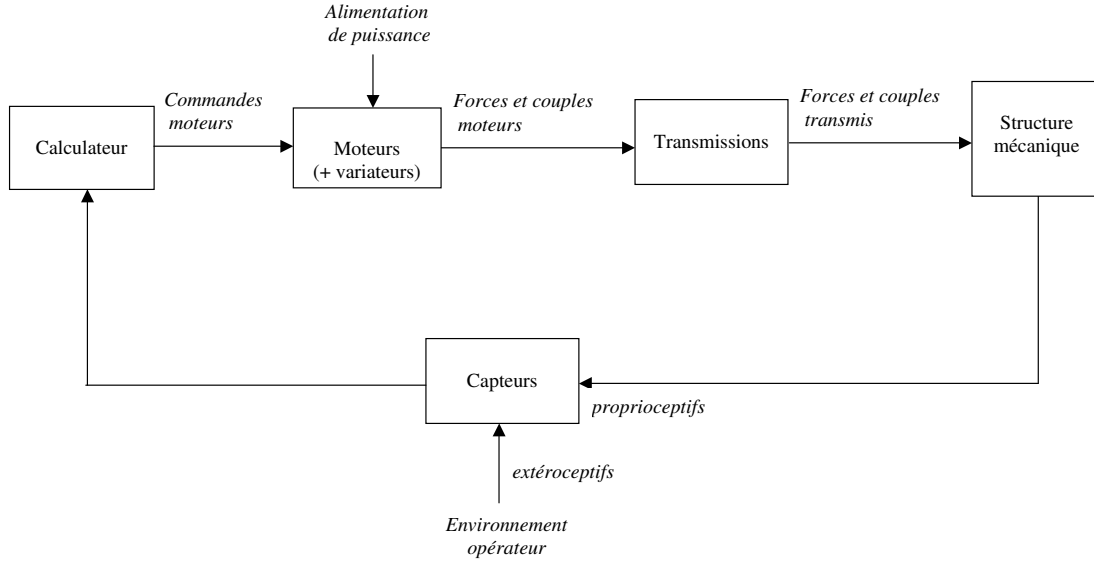


FIGURE 7.1 – Chaîne de commande d'un robot industriel.

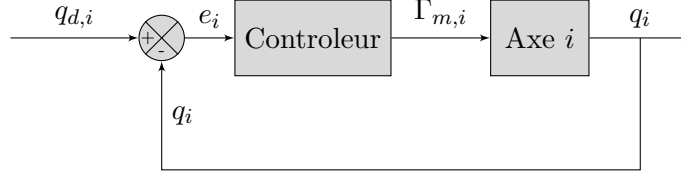


FIGURE 7.2 – Commande individuelle des axes

7.2 Commande individuelle des axes

La commande individuelle (également appelée commande découplée) consiste à asservir indépendamment chaque axe sans tenir compte des couplages dynamiques et de l'influence de la posture sur l'inertie. Le couple moteur de chaque axe est calculé par un contrôleur dédié (généralement de type PID) comme le montre la figure 7.2.

On se place dans la situation où l'on néglige les effets non linéaires, les variations des coefficients en fonction de la posture et les couplages dynamiques. Par ailleurs, on suppose les effets de la gravité majoritairement compensés par un système mécanique (cf. chapitre 5). 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 Γ_{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) :

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

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 .

7.3 Commande à couples calculés

Le comportement d'un robot est en réalité fortement non linéaire, variable dans le temps (dépendant de la posture) et à couplage dynamique (le mouvement d'un axe influe sur celui des autres axes). La commande individuelle des axes 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.

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

On rappelle que le modèle dynamique de la structure mécanique d'un robot (sans prendre en compte ni la contribution électromagnétique des moteurs, ni les interactions avec l'environnement), peut s'exprimer de la manière générale suivante :

$$\mathbf{\Gamma}_m = \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Q}(\mathbf{q}) + \mathbf{\Gamma}_f(\mathbf{q}, \dot{\mathbf{q}})$$

La commande à couples calculés consiste à imposer aux axes les couples ou forces $\mathbf{\Gamma}_m$ tels que :

$$\mathbf{\Gamma}_m = \hat{\mathbf{D}}(\mathbf{q})\mathbf{u} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) + \hat{\mathbf{Q}}(\mathbf{q}) + \hat{\mathbf{\Gamma}}_f(\mathbf{q}, \dot{\mathbf{q}})$$

avec \mathbf{u} un vecteur de contrôle. Le symbole $\hat{}$ désigne ici les grandeurs modélisées. La figure 7.3 illustre cette équation.

Si le modèle correspond « suffisamment convenablement » à la réalité du comportement dynamique du robot, alors le robot associé à sa commande présente le comportement dynamique suivant :

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

Ainsi, chaque axe est découplé des autres et linéarisé. Il devient un double intégrateur : $u_i = \ddot{q}_i$!

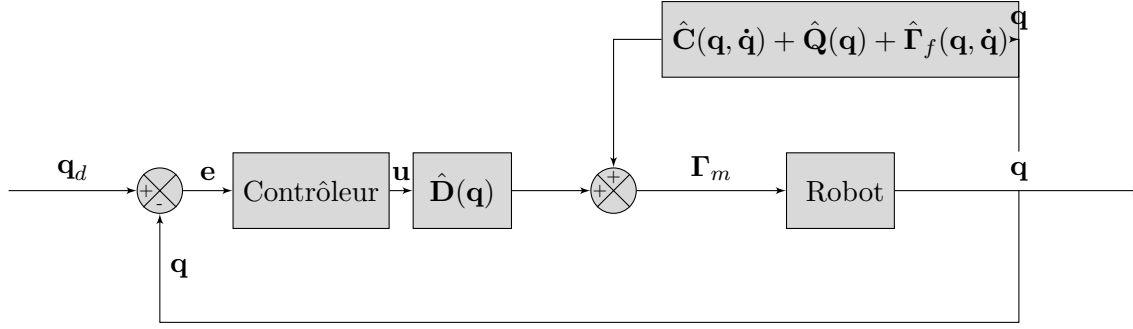


FIGURE 7.3 – Commande à couples calculés

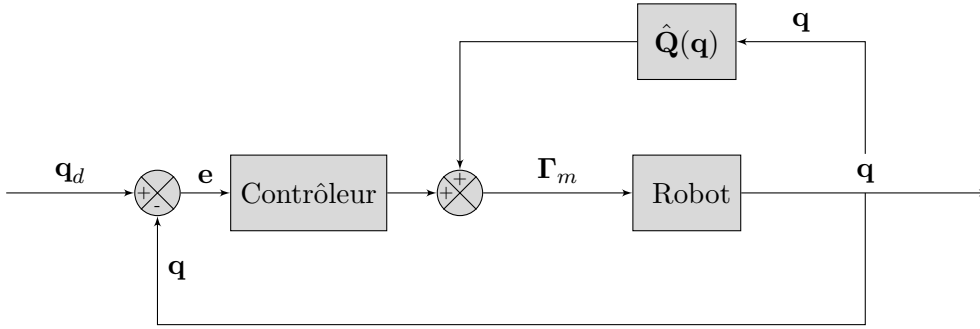


FIGURE 7.4 – Commande avec compensation de la gravité

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

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

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} .

NB: Un compromis intéressant entre la commande individuelle des axes et la commande à couples calculés est de n'utiliser que le terme $\mathbf{Q}(\mathbf{q})$ pour compenser la gravité. Cette version « dégradée », qui néglige les termes inertiels, centrifuges et de Coriolis, est représentée par le schéma de la figure 7.4.

7.4 Conclusion

Dans ce chapitre, nous avons présenté deux méthodes classiques de commande des axes mais il existe de nombreuses autres approches (commande prédictive, commande passive, commande adaptative, etc.). Pour plus d'informations, nous renvoyons le lecteur aux ouvrages de référence [4, 10].

Chapitre 8

Choix d'un robot

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 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.

8.1 Choix du robot

8.1.1 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.

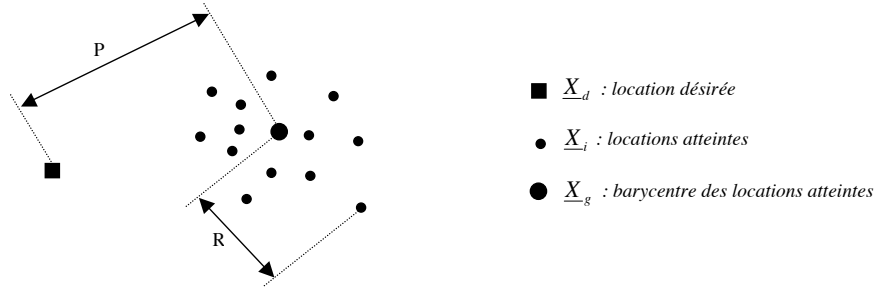
Nombre de degrés de liberté

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 8.1 – Illustration géométrique de la justesse P et de la fidélité 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.

Précision

Un robot est dit *précis* (ou exact) si il est à la fois juste et fidèle. Les conditions d'expériences permettant de mesurer la justesse et la fidélité d'un robot sont définies par la norme ISO9283 [1]. 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 fidélité comme suit (voir également la figure 8.1).

Définition 9 (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 10 (Fidélité/répétabilité). *La fidélité 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 :

$$l_i = \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2 + (z_i - z_g)^2} \quad (8.1)$$

— 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 fidélité statique en orientation [1].

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

Les erreurs de justesse et de fidélité 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 fidélité 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 fidélité 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 μm .

8.1.2 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 posture ($\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}}$). Là encore un logiciel de CAO robotique peut être d'un grand secours.

8.1.3 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.

8.2 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.

8.3 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.

8.3.1 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).

8.3.2 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 débâtements 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 postures singulières du robot.

8.3.3 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.

8.4 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).

Chapitre 9

Ce qui n'a pas été vu ...

9.1 Planification de trajectoires

9.2 Étude des robots parallèles

9.3 Identification des paramètres géométriques des robots

9.4 Identification des paramètres dynamiques des robots

9.5 Compliance

9.6 Commande référencée capteur : commande en effort - commande hybride - asservissements visuels

9.7 Parcourabilité - Dextérité - Manipulabilité

Bibliographie

- [1] Manipulating industrial robots - performance criteria and related test methods. International Standard ISO 9283, 1998. 76, 77
- [2] Philippe Coiffet. *La robotique : principes et applications*. Hermes Science, 1992.
- [3] Max Giordano and Jacques Lottin. *Cours de robotique : Description et fonctionnement des robots industriels*. Armand Colin, 1997. 6, 10, 11
- [4] Wisama Khalil and Etienne Dombre. *Modélisation, identification et commande des robots*. Hermes Science, 1999. 37, 74
- [5] J.P. Lallemand and S. Zeghloul. *Robotique : Aspects Fondamentaux*. Masson, 1994. 11, 12
- [6] Alain Liégeois. Modélisation et commande des robots manipulateurs. *Techniques de l'ingénieur*, S7730 :1–16, 2000.
- [7] Morvan Ouisse and Sylvaine Mallet. *Mécanique générale : Cinématique et dynamique des mécanismes*. Hermes Science, 2010. 53
- [8] Richard P. Paul. *Robot Manipulators : Mathematics, Programming, and Control*. The MIT press, 1981. 23, 37
- [9] Stéphane Régnier and Nicolas Chaillet, editors. *La microrobotique : Applications à la micromanipulation*. Hermes Science, 2008. 15
- [10] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2005. 46, 74

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 : <http://www.automatica-munich.com/>

Description : salon industriel à Munich

Collège de France

Adresse : <http://www.college-de-france.fr/site/jean-paul-laumond/>

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 : http://icube-avr.unistra.fr/index.php/Robotique_de_manipulation

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

European Robotics Research Network

Adresse : <http://www.euron.org/>

Description : réseau européen d'acteurs industriels et académiques, liste de diffusion, offres de stages et d'emplois

Everything-Robotic

Adresse : <http://www.everything-robotic.com/>

Description : blog

Groupement de Recherche (GdR) en Robotique

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

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

Innorobo

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

Description : salon industriel à Lyon

International Federation of Robotics

Adresse : <http://www.ifr.org/>

Description : fédération internationale d'acteurs industriels et académiques

Robohub

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

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

Robocaliser

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

Description : site de promotion de la robotique à destination des industriels français

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 : blog

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

V-REP

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

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

Distributions : Windows, Linux, Mac OS

Gazebo

Adresse : <http://www.gazebosim.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 : http://petercorke.com/Robotics_Toolbox.html

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 : <http://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 : <http://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

Visual Servoing Platform (ViSP)

Adresse : <http://www.irisa.fr/lagadic/visp/visp.html>,

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

Langage : C++

Distributions : Windows, Linux, Mac OS