



HAL
open science

Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service

Ignacio Herrera-Aguilar

► To cite this version:

Ignacio Herrera-Aguilar. Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 2007. Français. NNT: . tel-00199179

HAL Id: tel-00199179

<https://theses.hal.science/tel-00199179>

Submitted on 18 Dec 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service

THÈSE

présentée et soutenue publiquement le 28 septembre de 2007

pour l'obtention du

Doctorat de l'Université Toulouse III
(spécialité robotique)

par

Ignacio Herrera Aguilar

Composition du jury

<i>Président :</i>	Wisama Khalil	Professeur École Centrale de Nantes
<i>Rapporteurs :</i>	Philippe Fraisse Nicolas Andreff	MCF HDR, LIRMM MCF HDR, LASMEA IFMA
<i>Examineurs :</i>	Rachid Alami Walter Rubio	Directeur de recherche CNRS, LAAS-CNRS Professeur UPS
<i>Directeur de thèse :</i>	Daniel Sidobre	MCF UPS

Remerciements

Les travaux présentés dans ce mémoire ont été effectués dans l'équipe de Robotique et d'Intelligence Artificielle du Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS à Toulouse. J'adresse mes remerciements à Monsieur Malik Ghallab et Monsieur Raja Chatila, Directeurs successifs du Laboratoire, pour m'avoir accueilli dans cet établissement. J'exprime également mes remerciements à Monsieur Gerard Authie et Madame Caroline Berard, Directeurs successifs de l'École doctorale Systèmes.

Mes remerciements vont également à mes rapporteurs Monsieur Philippe Fraisse et Monsieur Nicolas Andreff. Je les remercie pour le soin avec lequel ils ont lu ce manuscrit, ainsi que pour la qualité de leurs critiques. Je remercie Monsieur Wisama Khalil et Monsieur Walter Rubio d'avoir accepté de participer à ce jury de thèse.

Je souhaite remercier mon encadrant Daniel Sidobre, maître de Conférence à l'Université Paul Sabatier, pour l'attention et la patience qu'il a portées au bon déroulement de ma thèse, pour les conseils qu'il m'a donnés, pour le temps qu'il m'a consacré et pour tous les problèmes qu'il a surmontés pour que je puisse arriver à ce niveau.

J'exprime également ma reconnaissance à Rachid Alami, responsable du groupe RIS et mon compagnon de bureau, pour tous ses enseignements, ses conseils et son amitié.

Je voudrais également remercier Sarah Fleury et Matthieu Herrb pour leur aide technique irremplaçable qui m'a permis de faire bouger le bras. À Jérôme, Patrick, Xavier et Christian pour le support technique.

Oublier ceux qui m'ont inspiré ou ceux qui m'ont servi de modèles serait inqualifiable, un grand merci pour tous les chercheurs et mes professeurs en France et au Mexique.

Je tiens à adresser des remerciements plus décontractés, mais néanmoins chaleureux et sincères, aux autres doctorants, anciens doctorants et stagiaires du laboratoire. Un remerciement spécial à Xavier B. pour son aide.

Enfin, en prenant le risque d'oublier des personnes qui ont largement contribué à rendre mon séjour en France plus facile et plus amical, un grand *MERCI* à Aurelie (Jefa), Julien et Thierry (Francesito). À Paulo, (An)Akin et Wael (Wallito) pour les bons moments passés au Portugal.

Me siento obligado a agradecer a toda la comunidad *castellano*-parlante por todo lo pasado juntos. Citar tan sólo los nombres sería una pena por no describir los recuerdos, otra vez gracias a todos. Sin embargo, Joan gracias por tu amistad y por toda la ayuda en el desarrollo de este trabajo. Claudia, gracias por todo, hermanita. Leo, espero que nos veamos en Colombia. Jib y Antonin, los primeros mexicanitos, muchas gracias por toda la ayuda. A Beto, Memo y familia, Gabriel y familia, Sergio y familia, Joel y familia, Gerardo y familia, Toño y familia, gracias por permitirme sentirme como en casa. Gustavo, Maggie, Efraín, Gibran y América, gracias. Edoard, cuñado, y Luis, hermano, gracias totales.

Este trabajo no pudo haberse realizado sin el apoyo del COSNET, DGEST y del Tecnológico de Orizaba.

Finalmente, a mi familia y seres queridos, gracias por todo el apoyo.

*Al nano*¹

¹Mi hermano QEPD

Table des matières

Liste des figures ix

Introduction xiii

Chapitre 1

Contexte de l'étude

1.1	Introduction	1
1.2	Planification de chemin	2
1.3	Commande des bras manipulateurs	2
1.3.1	Commande dans l'espace libre	3
1.3.2	Commande dans l'espace contraint	5
1.4	Génération de trajectoires	5
1.4.1	Génération de trajectoires en temps minimal par optimisation	6
1.4.2	Génération de trajectoires par satisfaction des contraintes cinématiques	6
1.4.3	Génération de trajectoire en ligne	7
1.5	Asservissement Visuel	8
1.5.1	Asservissement en boucle ouverte	9
1.5.2	Asservissement en boucle fermée	9
1.6	La commande d'un bras manipulateur pour la robotique de service	10

Chapitre 2

Commande des bras manipulateurs

2.1	Introduction	13
2.2	Généralités sur la sélection de la loi de commande	14
2.2.1	Définition de la Situation de l'organe terminal	14
2.2.2	Modèles Directs du robot manipulateur	15
2.3	Asservissement des articulations	16
2.3.1	La commande classique	16
2.3.2	La commande par ordinateur	16

2.3.3	Exemple 1 : Changement de type d'asservissement	18
2.3.4	Exemple 2 : Changement du type d'asservissement avec consigne calculée de position	19
2.3.5	Exemple 3 : Schéma avec signal d'anticipation	20
2.4	Asservissement dans l'espace cartésien	21
2.4.1	Commande à résolution de vitesse par retour de quaternion	21
2.4.2	Commande au voisinage des positions singulières	23
2.4.3	Résultats de simulation	24
2.5	Conclusions	27

Chapitre 3

Génération des mouvements souples

3.1	Introduction	29
3.2	Chemins et Trajectoires	30
3.3	Méthode de base pour les mouvements souples	30
3.3.1	Mouvement élémentaire à jerk maximal	31
3.3.2	Mouvement élémentaire à accélération maximale	31
3.3.3	Mouvement élémentaire à vitesse maximale	32
3.3.4	Mouvement souple à sept segments	32
3.3.5	Système d'équations du mouvement souple à sept segments	33
3.4	Résolution du système d'équation	35
3.4.1	Premier seuil D_{S1}	37
3.4.2	Mouvement point à point à cinq segments	37
3.4.3	Deuxième seuil D_{S2}	37
3.4.4	Mouvement point à point à trois segments	37
3.4.5	Algorithme pour le mouvement point à point	38
3.5	Mouvements en temps imposé	39
3.6	Mouvements à conditions initiales et finales quelconques	39
3.6.1	Mouvements élémentaires	40
3.6.2	Les trois types de mouvements élémentaires optimaux	40
3.6.3	Déplacements optimaux dans le plan accélération-vitesse	42
3.7	Mouvement multidimensionnel	45
3.7.1	Mouvement point à point	45
3.7.2	Trajectoire continue en dimension N sur un chemin défini	47
3.7.3	Considérations sur la génération du mouvement	51
3.7.4	Contraintes non homothétiques	57
3.8	Considérations sur les applications	60

3.8.1	Cas du bras manipulateur	60
3.8.2	Cas du manipulateur mobile	64
3.9	Conclusions	66

Chapitre 4

Retour Visuel

4.1	Introduction	67
4.2	Modèle d'une caméra	68
4.2.1	Matrice de projection perspective	69
4.2.2	Transformation affine caméra/image	70
4.2.3	Paramètres intrinsèques	71
4.2.4	Paramètres extrinsèques	71
4.3	Graphe du système avec caméra embarquée	71
4.4	Asservissement Visuel 2D partagé	73
4.4.1	Introduction	73
4.4.2	Schéma de commande	73
4.4.3	Loi de commande partagée	74
4.4.4	Avantages et inconvénients	76
4.5	Asservissement Visuel 3D	76
4.5.1	Introduction	76
4.5.2	Reconstruction 3D stéréo	76
4.5.3	Schéma de commande	78
4.5.4	Avantages et inconvénients	79
4.6	Localisation d'un objet et auto-calibration main/caméra ou « hand-eye »	79
4.6.1	Introduction	79
4.6.2	Schéma de commande	80
4.6.3	Avantages et inconvénients	84
4.7	Conclusions	84

Chapitre 5

Résultats expérimentaux

5.1	Introduction	85
5.2	Architecture logicielle	85
5.3	Mouvement point à point	86
5.3.1	Translation	86
5.3.2	Rotation	86
5.4	Mouvement continu	89

5.5	Asservissement visuel partagé	92
5.6	Guidage visuel	94
5.7	« Auto-calibration Hand-Eye »	94
5.8	Conclusions	96
Conclusions		97
Annexes		99
Annexe A Coordonnées Opérationnelles		99
A.1	Position	99
A.1.1	Coordonnées cartésiennes	99
A.1.2	Coordonnées cylindriques	100
A.1.3	Coordonnées sphériques	100
A.2	Orientation	100
A.2.1	Angles de Bryant	100
A.2.2	Angles d'Euler	100
A.2.3	Matrice de rotation	101
A.2.4	Angle plus vecteur	102
A.2.5	Quaternion	103
A.3	Représentation de la situation	105
A.3.1	Vecteur de coordonnées opérationnelles	105
A.3.2	Matrice de passage homogène	106
A.4	Vecteur vitesse instantanée de rotation	106
A.4.1	Le quaternion et le vecteur de vitesse instantanée de rotation	107
Annexe B Modèles pour un robot 6R		109
B.1	Modèle Géométrique Direct	109
B.2	Modèle Cinématique Direct	110
B.2.1	Calcul Indirect de la matrice Jacobienne	111
B.2.2	Calcul direct de la matrice Jacobienne	111
Annexe C Stabilité par Lyapunov		113
C.1	Limite de Stabilité pour la position	113
C.2	Limite de Stabilité pour l'Orientation	114
Références bibliographiques		117
Résumé		123

Liste des figures

1	Chemin pour la saisie d'une banane.	xiv
2	Schéma hiérarchique de commande pour la manipulation.	xv
3	Le robot de service au milieu humaine.	xv
1.1	Schéma hiérarchique de commande pour la manipulation avec un robot de service.	1
2.1	Schéma d'asservissement pour le servomoteur d'une articulation.	17
2.2	Évolution de la position, de la vitesse et de l'accélération au changement de type d'asservissement : passage de vitesse à position. La figure (d) montre l'intégrale de l'erreur quadratique de position.	18
2.3	Évolution de la position et zoom sur la position au changement de type d'asservissement : passage de l'asservissement en vitesse à l'asservissement en position.	19
2.4	Intégrale de l'erreur quadratique de position et évolution de la vitesse au changement de type d'asservissement : passage de l'asservissement en vitesse à l'asservissement en position.	20
2.5	Schéma d'asservissement avec terme d'anticipation pour un servomoteur.	20
2.6	Évolution de la position, de la vitesse et de l'accélération en utilisant le schéma de commande saturé plus signal d'anticipation. La figure (d) montre l'intégrale de l'erreur quadratique de position	21
2.7	Schéma de commande pour le bras manipulateur utilisant des quaternions pour le retour d'orientation.	22
2.8	Évolution de l'orientation	25
2.9	Évolution de la position et de l'intégrale de l'erreur quadratique en x, y et z	26
3.1	Définitions des sept segments pour la méthode de base avec des conditions cinématiques initiales et finales nulles	32
3.2	Seuils de Position pour choisir entre les trois cas	39
3.3	Espace vitesse - accélération : courbes des mouvements optimaux à sept segments.	40
3.4	Espace vitesse - accélération : à gauche courbes à jerk positif et à droite courbes à jerk négatif	41
3.5	Espace vitesse - accélération : évolutions pour différentes valeurs de J_{max}	42
3.6	Espace vitesse - Accélération : évolutions optimale entre deux situations quelconques P1 et P2	44
3.7	Espace vitesse - Accélération : de l'arrêt à une situation quelconque	45
3.8	Espace vitesse - Accélération : contraintes non homothétiques représentées après une homothétie égalant les V_{max}	46
3.9	Mouvement point à point en deux dimensions (X, Y) : Première étape de la stratégie, calcul de la trajectoire dans chacune des directions	48
3.10	Mouvement point à point en deux dimensions (X, Y) : homothétie dans le mouvement par correction du jerk	49
3.11	Virage entre segments de droites avec des points de contrôle.	50
3.12	Virage entre segments de droites parcourus à vitesse constantes	51

3.13	Exemple de trajectoires obtenues pour différents temps de commutation. Les premières courbes montrent les vitesses obtenues pour $T_c = [0.25 \ 0.50 \ 0.75 \ 1.00 \ 1.25]$, la dernière présente de façon comparative les trajectoires.	53
3.14	Effet de « coude » produit par les différentes destinations finales pour une même trajectoire initiale.	54
3.15	Corrections en durée et en jerk pour les quatre types de mouvements	57
3.16	Exemple de mouvement à deux dimensions pour une trajectoire définie par trois points. Les courbes de jerk, d'accélération, de vitesse de position indépendante et de position pour le mouvement composé sont présentées. Les courbes pointillées représentent la trajectoire vers la position intermédiaire.	59
3.17	Rotation de 90° et Rotation de 180° autour de Z , pour comparer l'interpolation linéaire et le slerp. Les quatre premières images correspondent aux composants du quaternion n et k ou les sigles i correspond à interpolation, n à normalisé et s à slerp. Les deux dernières graphiques correspondent à la rotation dans l'espace 3D.	62
3.18	Variation entre interpolations successives et rotation relative pour une rotation de 90° pour le slerp et le nlerp.	63
3.19	Erreur d'interpolation. L'image (a) montre l'erreur pour 5 quaternions interpolées entre Q_a et Q_b . L'image (b) montre les mesures pour calculer l'erreur maximale	64
3.20	Erreur de Interpolation. L'image (a) montre l'erreur de interpolations pour des rotations successives de 180° à -180° avec un pas de 10° entre courbes. L'image (b) montre l'erreur en pourcentage pour des rotations de 180° , 135° , 90° et 45°	65
4.1	Modèle Sténopé d'une caméra, avec le plan image réel $U'V'$ et le plan image virtuel UV	69
4.2	Image dans le repère UV	70
4.3	Graphe des repères du système de caméra embarquée et repères associés à une tâche de manipulation	72
4.4	Schéma de commande pour le bras manipulateur en utilisant un schéma d'asservissement visuel 2D	73
4.5	Reconstruction 3D par information stéréo	77
4.6	Schéma de commande pour le bras manipulateur utilisant un schéma d'asservissement visuel 3D	79
4.7	Schéma de commande pour le bras manipulateur en utilisant un filtre de Kalman étendu (EKF) pour estimer la position de la cible	80
5.1	Architecture logicielle pour la manipulation	85
5.2	Translation entre les points A et B sans changement d'orientation. Cette vue présente sous forme filaire les trois premiers corps du bras manipulateur.	87
5.3	Résultats obtenus après un mouvement de translation pure. À gauche, les courbes d'accélération, de vitesse et de position pour les directions X et Z . À droite, des courbes comparatives entre la position de consigne et la position mesurée.	87
5.4	Rotation entre les quaternions Q_o et Q_f sans changement de position de l'organe terminal.	88
5.5	Résultats obtenus après un mouvement de rotation pure. A gauche, les courbes d'accélération, de vitesse et de position pour les composants du quaternion n et j . A droite, des comparaisons entre la position de consigne et la position mesurée, et les erreurs de position.	89
5.6	Résultats obtenus pendant un mouvement de rotation pure. La position est fixe, donc nous présentons les directions X et Z , avec du côté gauche la direction X et du côté droit la direction Z . En haut la comparaison entre la consigne et la valeur mesurée, en bas l'erreur obtenue.	89
5.7	Séquence de mouvement réalisée par le bras manipulateur pour saisir la boîte bleue.	90
5.8	Comparaison entre la trajectoire planifiée (noir) et la trajectoire réalisée (bleu). Les points noirs définissent la trajectoire réalisée par la pointe de l'organe terminal.	91

5.9	Courbes planifiées pour la direction X du côté gauche et erreurs obtenues du côté droit.	91
5.10	À gauche les courbes planifiées pour la composante n et à droite les comparaisons de performance.	92
5.11	Montage « hand-eye » de la caméra et la ligne à suivre	92
5.12	Asservissement visuel partagé pour le suivi ascendant d'un ligne droite. À gauche le mouvement planifié et à droite le mouvement réalisé avec l'asservissement visuel activé.	93
5.13	Mouvement dans la direction Z et erreur de déplacement en Z à gauche. À droite, comparaison entre la consigne planifiée et la position mesurée dans la direction Y , et l'erreur entre P_p et P_y .	93
5.14	Évolution de deux points dans l'image après 40 « photos ».	94
5.15	L'estimation de la position de la cible converge vers la position réelle de la cible (en noir) à partir d'une position initiale estimée (en rouge). À gauche, l'évolution de l'estimation de la position de la cible pour la calibration de l'orientation et à droite pour la calibration de la position.	95
5.16	Évolution de l'estimation de l'orientation des caméras. À gauche la composante n du quaternion et à droite le composante i . Les lignes centrales correspondent aux valeurs simulées.	95
5.17	Évolution de l'estimation de la position des caméras. À gauche la position en X et à droite la position en Z . Les lignes centrales correspondent aux valeurs simulées.	96
A.1	Représentation de la position du point \mathbf{P} dans les divers systèmes de coordonnées.	99
A.2	Angles de Bryant, les trois rotations successives (λ, μ, ν)	100
A.3	Angles d'Euler, les trois rotations successives (ψ, θ, ϕ)	101
A.4	Composantes des vecteurs unitaires du repère \mathcal{R}_{i-1} exprimées dans le repère \mathcal{R}_i .	101
A.5	Rotations autour des axes x , y et z	102
A.6	Vitesse de Rotation autour d'un axe	107
B.1	Définition des Coordonnées Généralisées et des repères d'un robot manipulateur de type 6R.	110

Introduction

Quand on parle de robotique, plusieurs idées viennent à l'esprit de chacun de nous. Historiquement, nous pourrions nous référer aux premiers concepts et automates de l'antiquité ou aux premiers robots comme à des personnages de la mythologie. Même le mot *robot* a sa propre histoire. Séparer la science de la science fiction n'est pas une chose aisée, surtout lorsque en robotique nous cherchons parfois à faire réalité la fiction. Un exemple de l'influence des fictions nous est donné par les lois de la robotique.

C'est au siècle dernier que le « boom » de la robotique industrielle a amorcé l'explosion des thèmes de recherche. A cette époque les robots étaient conçus en respectant les contraintes imposées par le milieu industriel, comme la répétabilité, la précision dans la réalisation des tâches, le respect des cadences de production, etc. Prenons l'exemple des bras manipulateurs industriels des années 80 très semblables aux bras industriels actuels, ils se déplacent à très grandes vitesses sur des trajectoires calculées *hors ligne* dans des environnements protégés et structurés, sans interaction avec les opérateurs humains.

C'est avec les développements scientifiques, spécifiquement de l'électronique et de l'informatique mais aussi automatique, mathématique, mécanique, matériaux, que la technologie robotique a progressé. Les robots actuels sont dotés d'une « intelligence » qui leur donne une certaine autonomie qui va leur permettre de se diffuser dans de nouveaux domaines. Nous parlons maintenant de robots d'exploration, de robots d'intervention, de robots ludiques, de robots de service, etc.

Dans ce travail de thèse, nous nous sommes intéressés aux robots manipulateurs pour des applications en robotique de service. Les contraintes de la robotique de service sont très différentes des contraintes de la robotique industrielle. Le robot manipulateur ne sera plus un outil qui effectue une tâche répétitive dans un espace sécurisé, mais un partenaire des activités quotidiennes. Les contraintes de la robotique de service sont liées à la sécurité et au confort de l'homme.

Reprenons l'exemple du bras manipulateur en milieu industriel, il réalise des tâches de façon répétitive, à hautes vitesses, pour satisfaire les temps d'exécution. Les lois de commande utilisées utilisent des capteurs proprioceptifs pour mesurer l'état du robot et des capteurs extéroceptifs pour mesurer des variables liés à la tâche. Lorsque l'espace de travail est sécurisé et la tâche connue *a priori*, les calculs de la trajectoire sont réalisés *hors ligne* pour minimiser les temps d'exécution.

Considérons maintenant le cas d'un robot de service effectuant des tâches de manipulation. L'utilisation des capteurs proprioceptifs couvre les mêmes besoins mais seule l'utilisation de capteurs extéroceptifs permet l'obtention d'informations sur la tâche et son environnement. Lorsque l'espace de travail d'un robot de service est partagé avec l'homme, la connaissance de l'environnement prend une importance considérable pour la réalisation de la tâche. La trajectoire à réaliser est alors fonction de l'environnement, ce qui nécessite que tous les calculs nécessaires à la réalisation de la tâche soient réalisés *en ligne*. En effet les trajectoires doivent évoluer en fonction des incertitudes sur la connaissance de l'environnement. Les capteurs extéroceptifs utilisés le plus souvent pour ces applications sont des systèmes de vision embarqués.

Décrivons maintenant une tâche typique de la robotique de service, la saisie d'un objet pour l'apporter à un homme. Les travaux de López-Damian portent sur la planification de la saisie par un robot autonome [Damian, 2006]. La figure 1 présente le chemin calculé par le robot Jido du LAAS - CNRS pour la saisie d'une banane en utilisant les algorithmes développés par López-Damian.

Pour simplifier la tâche nous allons faire certaines hypothèses :

- L'objet à saisir est connu (le modèle de l'objet est connu) et les algorithmes sont capables de calculer une position de saisie.

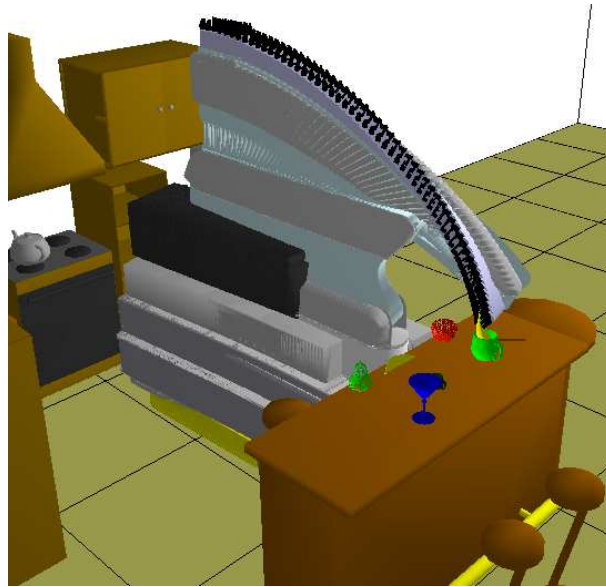


FIG. 1 – Chemin pour la saisie d'une banane.

- L'objet se trouve dans l'espace de travail du robot manipulateur. Tel que si le robot est monté sur une plate-forme mobile, il n'y a pas de déplacement de la plate-forme.
- L'homme se trouve dans le même espace et ne bouge pas pendant la réalisation de la tâche, de telle sorte que le système n'a pas besoin de réagir au comportement humain.

Alors, une stratégie pour la saisie d'un objet est définie par :

1. Le système de vision détermine la position de l'objet et la position de l'homme par rapport au robot.
2. Le système de planification de saisie choisit la position de saisie de l'objet et planifie le chemin à suivre.
3. Un système de génération de trajectoire calcule la loi de mouvement pour réaliser la tâche.
4. Une boucle d'asservissement réalise le suivi de cette trajectoire.
5. Le système de planification détermine le chemin pour apporter l'objet à l'homme.
6. Le système de génération de mouvement calcule cette nouvelle trajectoire.
7. Le boucle d'asservissement exécute cette trajectoire.

Si les incertitudes sont faibles, la réussite de la réalisation de la tâche est très élevée. Considérons maintenant des incertitudes plus importantes dans la mesure de la position de l'objet. La source de ces incertitudes est très variée : un modèle imparfait, des erreurs de calibration du système de vision, etc. Le planificateur de saisie va calculer un chemin qui ne garantit pas la saisie même si on améliore la qualité de l'asservissement sur la trajectoire. En l'absence de capteurs extéroceptifs pour corriger les incertitudes, le robot a une probabilité de réussir qui est très faible.

Dans cette thèse nous considérons le schéma de commande hiérarchique pour la manipulation présenté sur la figure 2 , il correspond à la stratégie décrite précédemment. L'information fournie par les capteurs proprioceptifs est utilisée comme mesure de retour à deux niveaux différents : au plus bas niveau pour l'asservissement des articulations qui sont le plus souvent commandées en vitesse angulaire ; au deuxième niveau qui correspond à l'asservissement de la trajectoire en position. L'information fournie par les capteurs extéroceptifs est utilisée aux deux niveaux : comme mesure de retour dans le premier cas pour corriger les positions et/ou vitesses dans les boucles d'asservissement de trajectoire et sous la forme de positions de consigne pour la génération de trajectoire dans le second.

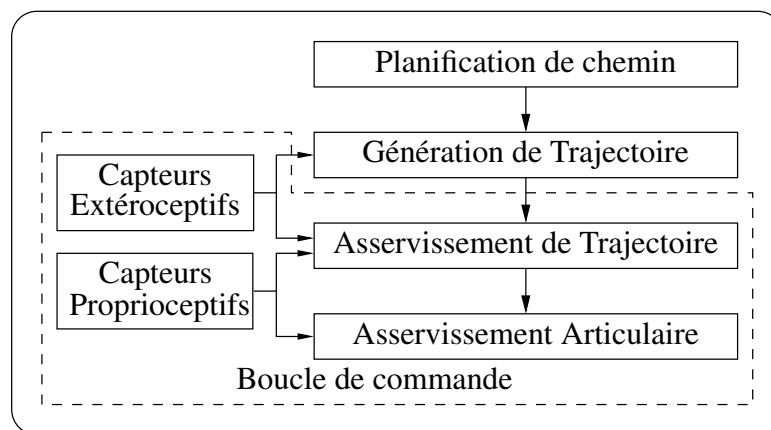


FIG. 2 – Schéma hiérarchique de commande pour la manipulation.

Pour les expérimentations, nous utilisons comme capteur extéroceptif une paire de caméras stéréoscopiques embarquée sur l'organe terminal du bras manipulateur. Nous pouvons ainsi réaliser des contrôleurs qui intègrent l'asservissement articulaire, l'asservissement de trajectoire et la génération de trajectoire.

La sécurité des hommes autour des robots de service a guidé nos choix pour concevoir une commande capable de réagir à toutes informations nécessitant de dévier ou d'arrêter le robot. Nous avons aussi essayé d'aller plus loin en proposant une commande qui assure le confort des humains en évitant les mouvements brusques. À titre d'illustration, la figure 3 présente le robot jido en train d'apporter une petite bouteille.

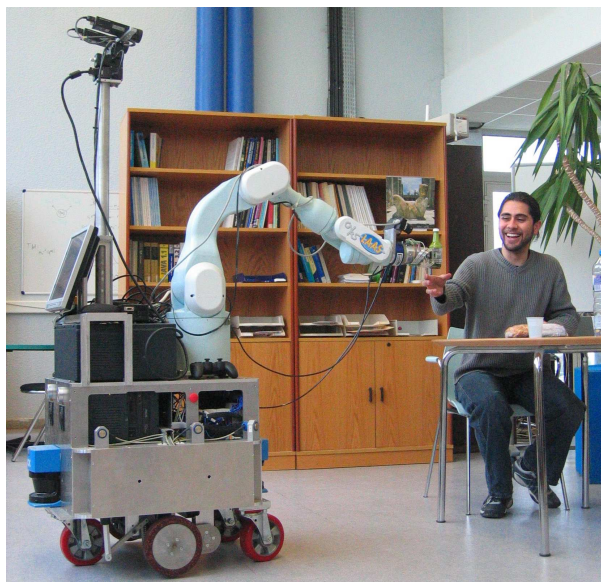


FIG. 3 – Le robot de service au milieu humaine.

Dans le domaine des neurosciences, les travaux de Hogan et Flash ([Hogan, 1984], [Flash et al., 1985]) sur la modélisation des mouvements des bras, d'abord de primates puis d'humains, établissent que la souplesse des mouvements est obtenue en minimisant une fonction objectif qui porte sur la variation de l'accélération appelée le *jerk*.

Sur le modèle des mouvements humains, nous proposons de commander les bras manipulateurs au

niveau du jerk de l'organe terminal. Pour cela, notre commande borne le jerk, l'accélération et la vitesse dès l'étape de génération de mouvement. Dans la suite du document, nous qualifierons ces mouvements de mouvements souples.

Objectifs

Le problème de la commande du bras manipulateur pour des applications à la robotique de service sert de guide à ce mémoire. L'objectif est de présenter une stratégie de commande utilisant des données proprioceptives et extéroceptives et prenant en compte les contraintes propres de la robotique de service telles que la sécurité et le confort des humains.

Contributions

Notre contribution se situe dans le domaine de la commande des bras manipulateurs pour des applications en robotique de service. Pour assurer le confort des utilisateurs humains lorsque les applications nécessitent que le système effectue le calcul de la trajectoire *en ligne*, nous introduisons les mouvements souples. L'utilisation de retour visuel est considérée au milieu des stratégies différentes sur le schéma de base.

Organisation de la thèse

Ce document est organisé de la façon suivante :

- Le chapitre 1 est dédié au contexte de l'étude. La robotique de service impose ses propres contraintes, dans ce mémoire nous reprenons les travaux menés sur la robotique industrielle, et les intégrons à la robotique de service.
- Le chapitre 2 présente la commande des bras manipulateurs. Lorsque les contraintes sont différentes de celles de la robotique industrielle, nous reprenons les schémas classiques et proposons des améliorations pour satisfaire les contraintes de la robotique de service.
- Le chapitre 3 traite la génération des mouvements souples. Nous présentons la méthode en partant du cas le plus simple qui correspond à un mouvement point à point à une dimension jusqu'au le mouvement continu sur un chemin défini par plusieurs points dans l'espace.
- Le chapitre 4 est dédié au retour visuel. L'utilisation d'information visuelle est présentée sur trois schémas différents, le premier est un schéma d'asservissement visuel, le deuxième correspond à un schéma de guidage visuel et le troisième utilise un filtre de Kalman pour gérer les incertitudes de la modélisation.
- Le chapitre 5 présente l'architecture robotique permettant de mettre en oeuvre nos schémas de commande pour différentes expériences, et les résultats expérimentaux obtenus.
- Dans les conclusions générales nous portons un regard critique sur l'ensemble des travaux réalisés et nous finissons cet exposé par quelques propositions de travaux.
- Trois annexes sont incluses : la première traite les coordonnées opérationnelles, la deuxième présente les modèles pour un robot 6R et la dernière présente l'analyse de stabilité par Lyapunov pour les boucles de commande.

Chapitre 1

Contexte de l'étude

1.1 Introduction

Reprenons la figure du schéma hiérarchique de commande présentée en l'introduction de ce document. Pour illustrer l'utilisation de l'information provenant des capteurs extéroceptifs, nous avons choisi un système de vision qui dotera le robot d'une certaine autonomie pour la réalisation de tâches différentes. Nous utiliserons aussi les capteurs proprioceptifs de position angulaire des articulations du bras manipulateur. Le schéma de la figure 1.1 montre le schéma hiérarchique que nous utiliserons au cours de ce travail.

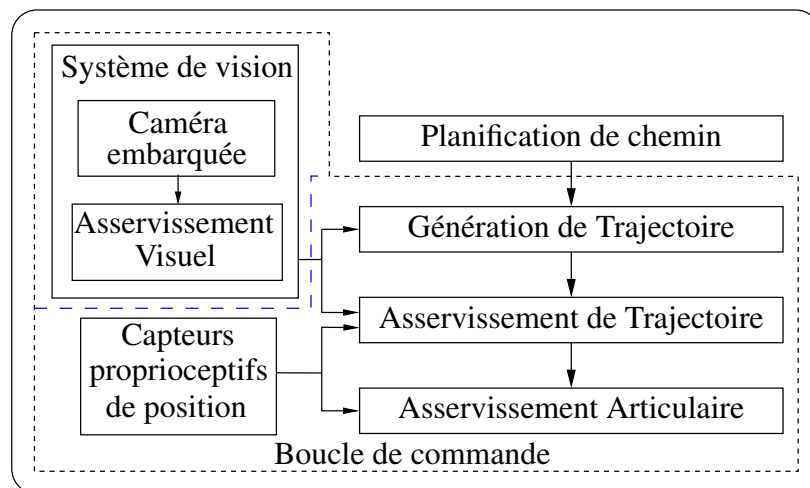


FIG. 1.1 – Schéma hiérarchique de commande pour la manipulation avec un robot de service.

Sur ce schéma, nous pouvons distinguer quatre thèmes : la planification de chemin, la commande des bras manipulateurs, la génération du mouvement et le retour visuel. Nous avons établi que les contraintes propres de la robotique de service ne sont pas les mêmes que celles de la robotique industrielle. En gardant cette idée en tête, nous allons présenter différents développements de ces quatre thèmes de la robotique et les relier au contexte de notre étude. La dernière partie du chapitre est consacrée à la présentation de notre approche.

1.2 Planification de chemin

Une tâche de robotique de service peut être décomposée en une suite de mouvements. La complexité de ces mouvements augmente avec la nature de des déplacements mais aussi avec la dynamique de l'environnement. La planification de chemin a un rôle prépondérant pour la robotique de service, elle est liée aux chemins que les robots doivent suivre pour l'exécution d'une tâche. Les techniques de planification de chemin sont largement étudiés dans [Latombe, 1991] et [Laumond, 1998].

La manipulation d'objets en environnements humains est une des aptitudes qu'un robot de service doit être capable de maîtriser. Les travaux de López-Damian [Damian, 2006] portent sur la planification de la saisie par un robot autonome. La saisie est planifiée en deux étapes : la première étape recherche des positions de saisie en fonction de l'organe préhenseur ou pince et des caractéristiques de l'objet comme ses axes d'inertie. La deuxième étape consiste à trouver le chemin que le robot suivra.

Par ailleurs, certains problèmes liés au confort de l'homme sont traités par Sisbot. En utilisant un planificateur de chemins, le mouvement du manipulateur est généré en fonction de la position et du champ de vision de l'utilisateur humain [Sisbot et al., 2006].

1.3 Commande des bras manipulateurs

Dans [Sciavicco et al., 2000], le problème de la commande d'un robot manipulateur peut être formulé comme la détermination de l'évolution des forces généralisées (forces ou couples) que les actionneurs doivent exercer pour garantir l'exécution de la tâche tout en satisfaisant certains critères de performance.

Différentes techniques sont utilisées pour la commande des bras manipulateurs. La conception mécanique du bras manipulateur a une influence sur le choix du schéma de commande. Un robot manipulateur est une structure mécanique complexe dont les inerties par rapport aux axes des articulations varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations.

La plupart des robots utilisent des servomoteurs électriques comme actionneurs. Dans le cas de servomoteurs ayant de faibles rapports de réduction, ce sont les servomoteurs qui doivent compenser les effets des variations des forces d'inertie et de gravité. Dans le cas de servomoteurs avec de forts rapports de réduction, l'inertie vue par les moteurs varie beaucoup moins et il est alors possible de modéliser le robot par un système linéaire qui permet de découpler les articulations.

Dans le contexte de ce document nous considérons uniquement l'utilisation de servomoteurs avec de forts rapports de réduction comme actionneurs, ce qui produit des robots à articulations rigides. Le problème de la rigidité des articulations est évident lorsqu'on parle d'interaction avec l'environnement ou des collisions. Des imprécisions dans la modélisation de l'environnement peuvent se traduire par des efforts de contact importants qui peuvent endommager les mécanismes internes du robot ou son environnement.

En robotique de service, le risque de collision avec les humains est élevé. Lorsque l'articulation est rigide, les blessures qu'un robot peut produire sont fonction de l'inertie du mouvement. Pour des applications en robotique de service, et pour réduire les dommages produits par une collision, il est recommandé d'utiliser des robots ayant « une élasticité », on parle alors de robots avec des articulations flexibles. On peut considérer aussi des actionneurs élastiques [Pratt et al., 1995] ou une action parallèle-distribué [Zinn et al., 2002] pour la conception d'un robot de service, sujet qui est en dehors de ce travail. Un chapitre du livre [Khalil, 2002] est dédié à la commande de robots à articulations flexibles que nous recommandons au lecteur intéressé. La littérature présente au robot léger DLR qui a été développé pour évoluer au milieu des espaces humains, les travaux de [Albu-Schäffer et al., 2007] utilisent la commande en couple pour chacune des joints pendant que les travaux de [De Luca et al., 2006] considèrent la détection des collisions et la façon de réagir après la collision.

Deux types de mouvements apparaissent quand on parle de commande du bras manipulateur. Un premier type considère que les mouvements nécessaires pour la réalisation de la tâche sont exécutés dans l'espace libre. Le deuxième type considère des mouvements spécifiques avec des forces de contact pour l'organe terminal qui se déplace dans un espace contraint. Toute tâche de robotique de service est réalisée

par une combinaison de ces deux types de mouvement. Pour simplifier la commande, les deux types de mouvement sont abordés séparément.

1.3.1 Commande dans l'espace libre

Nous ne pouvons pas dans le cadre de ce mémoire traiter en détail l'ensemble des techniques de commande des bras manipulateurs. Pour une étude plus détaillée, le lecteur pourra se référer à [Paul, 1979], [Gorla et al., 1984], [Khalil et al., 1999], [Sciavicco et al., 2000] et [Canudas et al., 1997]. Nous présentons juste une vue générale de différentes techniques rapportées dans la littérature.

Dans les paragraphes suivants nous nous limiterons à une description simple des différentes techniques. Nous présentons tout d'abord le schéma de commande par articulation pour progresser jusqu'à des techniques plus performantes comme la commande optimale. À la fin de cette section, nous décrivons notre approche pour la commande.

Commande par articulation

Cette technique est utilisée par des robots manipulateurs qui utilisent des servomoteurs avec de forts rapports de réduction. Lorsque le système présente un comportement linéaire, l'asservissement du mouvement peut être réalisée par des techniques classiques de commande. Nous parlons alors d'une commande décentralisée de type PID. Dans [Gorla et al., 1984] et [Canudas et al., 1997], le schéma classique est amélioré avec des signaux d'anticipation pour corriger les effets de la force de gravité et de couplage.

Commande jacobienne

Cette technique est utilisée depuis les travaux de Whitney [Whitney, 1969] et elle est appelée de cette façon lorsqu'elle utilise la matrice jacobienne inverse du bras manipulateur pour calculer les vitesses de consigne aux articulations. Elle est aussi connu sous le nom de *commande à mouvement résolu*. Les approches les plus courantes sont de type : mouvement à vitesse résolu, mouvement à accélération résolu [Luh et al., 1980] et mouvement à force résolu [Wu et al., 1982]. La technique de mouvement résolu commande la position de l'organe terminal du manipulateur dans l'espace cartésien, par combinaison des mouvements de plusieurs articulations.

Ce type de schémas peut se présenter sous la forme d'un schéma cinématique quand les vitesses calculées aux articulations sont utilisées directement comme des consignes pour les boucles de commande de chaque articulation, ou sous la forme d'un schéma dynamique quand le schéma utilise le modèle dynamique pour découpler les articulations.

Son principal inconvénient est l'utilisation de l'inverse de la matrice jacobienne, qui peut devenir singulière dans certaines conditions.

Commande par découplage non linéaire

Cette technique est aussi connu sous les noms de *commande dynamique* ou *commande par couple calculé*. Lorsque l'application exige des évolutions rapides avec des contraintes dynamiques, la commande doit prendre en compte les forces d'interaction. Ce type de technique considère l'ensemble des articulations et, pour les découpler, utilise la théorie du découplage non linéaire. Cette théorie utilise le modèle dynamique du robot pour le calcul de la lois de commande, ce qui conduit à des lois de commande centralisées non linéaires. Des signaux d'anticipation peuvent être utilisées pour minimiser des effets non linéaires [Er, 1993].

Ce type de technique permet la commande dans l'espace des articulations ou dans l'espace cartésien, avec l'avantage que les articulations sont découplées et peuvent évoluer a grandes vitesses avec de fortes inerties. Cette méthode dépend fortement du modèle du système, elle est très sensible aux imprécisions du modèle qui entraînent un découplage imparfait. Ceci constitue son principal inconvénient.

Commande fondée sur une fonction de Lyapunov

Des méthodes basées sur une fonction de Lyapunov ont été utilisées pour la commande des bras manipulateurs de façon satisfaisante pour des tâches de suivi. Particulièrement lorsqu'on cherche à garantir la convergence asymptotique et non à linéariser le système ou à obtenir le découplage [Canudas et al., 1997].

Commande passive

Cette technique considère le robot comme un système passif, c'est à dire un système qui dissipe de l'énergie. De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il réalise la tâche. En utilisant le formalisme de Hamilton, la commande cherche à minimiser l'énergie du système en utilisant un bloc non linéaire passif dans la boucle de retour. La commande passive tend à être plus robuste que le découplage non linéaire, lorsque la technique ne recherche pas à annuler les non linéarités [Sciavicco et al., 2000].

Commande adaptative

Ce type de techniques vise à corriger les déficiences de la commande par découplage non linéaire, comme la connaissance approximative des paramètres du modèle du robot ou pour s'adapter aux différentes conditions opératoires. Ce type de schémas cherche à estimer ou ajuster en-ligne les valeurs des paramètres utilisées dans le calcul de la loi de commande. Un des travaux les plus intéressants sur ce sujet est la commande proposé par [Slotine et al., 1987] appelée commande de Slotine-Li ou commande adaptative passive. Plusieurs travaux sur la commande adaptative sont présentés dans [Lozano et al., 2001]. Les avantages de ce type de techniques sont évidents, malheureusement la puissance de calcul demandée au système constitue un inconvénient important.

Commande prédictive

Ce type de schémas, en utilisant le modèle du système et les consignes, est capable de prédire son évolution, de telle manière qu'il est capable d'agir en fonction de l'erreur de prédiction. Trois schémas différents sont proposés dans [Hedjar et al., 2002] : point final fixe, horizon fini et une combinaison des deux précédents. Un grand avantage de ce type d'approche est lié au fait que l'erreur de prédiction n'est pas contaminé par les bruits de mesure mais la dépendance au modèle reste forte.

Commande robuste

Dans le cas de paramètres fixes, il est connu que la technique de découplage non linéaire peut devenir instable en présence d'incertitudes. Si les paramètres du modèle ne sont pas connus de façon précise et si l'incertitude sur les paramètres admet des bornes connus, alors les techniques de commande robuste peuvent être utilisées. Par exemple, les travaux de Slotine [Slotine, 1985] considèrent la technique de modes glissants appelé aussi commande de structure variable. Cette technique utilise une surface de glissement où la stabilité du système est garantie.

Commande optimale

Pour réaliser une tâche, il peut exister un grand nombre de solutions. Dans ce cas, il peut être souhaitable de choisir une solution qui satisfasse un certain critère.

La littérature présente différents types de critères pour la commande optimale : la commande en temps minimal ([Kahn et al., 1971], [Bobrow et al., 1985], [Chen et al., 1989]), du domaine des neurosciences la minimisation du jerk pour maximiser la souplesse du mouvement ([Flash et al., 1985], [Kyriakopoulos et al., 1988], [Seki et al., 2004], [Amirabdollahian et al., 2002]) et la minimisation du couple ([Egeland et al., 1988], [Martin et al., 1999]), entre autres.

La complexité du problème de commande optimale a motivé les chercheurs pour diviser la tâche en deux étapes : la première étape est la planification de trajectoire suivi d'une étape d'asservissement de la trajectoire [Lin et al., 1983], [Kim et al., 1985], [Shin et al., 1985].

Approche sélectionnée

La division en deux étapes de la commande proposée par les schémas de commande optimale a été conservée pour la suite de nos travaux. Une étape de planification de trajectoire est nécessaire pour satisfaire les conditions de mouvement et une étape d'asservissement de trajectoire devra être utilisée pour suivre cette trajectoire. À la différence des méthodes optimales, nous cherchons ici un schéma qui soit assez robuste pour être utilisé dans des environnements humains et assez flexible pour l'utiliser avec des robots différents.

Les robots que nous avons utilisé ont de forts rapports de réduction, nous pouvons utiliser une commande par articulation, qui est assez robuste pour les applications que nous envisageons. Et, pour la commande dans l'espace cartésien, une commande jacobienne est envisagée. Souhaitant tester notre approche sur des robots différents en utilisant leurs modèles cinématiques respectifs sans avoir une forte dépendance par rapport à leur modèle dynamique.

1.3.2 Commande dans l'espace contraint

Lorsque l'organe terminal entre en contact avec une surface, la chaîne cinématique du robot est fermée sur l'environnement, et du fait de la raideur de l'ensemble, de faibles variations de position du point de contact peuvent induire des efforts importants. Lorsque le robot est contraint par l'environnement suivant toutes les directions, aucun déplacement de l'organe terminal n'est possible et il ne peut qu'exercer des efforts sur l'environnement ; en revanche, lorsque le robot est dans l'espace libre, l'absence de tout contact empêche la génération d'efforts. Ainsi, les notions de commande en position et de commande en effort s'excluent mutuellement : on ne peut contrôler simultanément une force et un déplacement suivant une même direction.

Dans la pratique, on associe un repère de compliancance à l'organe terminal, de façon à ce que pour chacun des 6 degrés de liberté on associe soit une « position » soit un « effort ».

La littérature propose cinq schémas de commande, les trois premiers n'utilisent pas de consigne d'effort : compliancance passive, commande par raideur active et commande en impédance. Les deux suivantes utilisent la consigne d'effort : la commande hybride et la commande hybride externe.

En considérant que la sécurité est prépondérante dans le domaine de la robotique de service, la compliancance ne doit pas être associée uniquement à l'organe terminal mais à toute la chaîne cinématique du manipulateur. Pour réduire les effets d'une collision avec l'homme, la littérature propose plusieurs résultats de recherche qui portent sur l'utilisation d'une peau artificielle pour les robots dans le but d'éloigner le bras manipulateur des zones de collision. Par ailleurs, les travaux de Brock [Brock et al., 1997] considèrent des efforts artificiels dans l'espace libre pour éloigner le robot d'une région de l'espace, en utilisant le concept de bande élastique.

1.4 Génération de trajectoires

La génération de trajectoires est la partie du système de commande qui accepte des instructions de mouvement (déplacement dans un sens vers une position finale sur un certain chemin) et qui produit une série de consignes. De telles consignes peuvent être envoyées directement aux actionneurs ou aux boucles de commande. Les instructions de mouvement typiques contiennent la définition du chemin, les contraintes cinématiques que le robot doit respecter, les temps d'exécution, etc. Donc la génération du mouvement doit produire une trajectoire qui suit le chemin de façon uniforme, c'est à dire sans discontinuités de vitesse, et si possible des dérivés d'ordre supérieur.

Les méthodes de génération de trajectoires peuvent être classées de différentes façons : selon l'espace utilisé, articulaire ou cartésien ; selon que le calcul est effectué *en ligne* ou *hors ligne* ; selon le type de

données d'entrée et les contraintes imposées : trajectoire rectiligne, suivi de chemin ou point à point ; selon le type de mobilité : bras manipulateur, plate-forme mobile, bras mobiles.

Dans les paragraphes suivants nous ne traiterons que de la planification de trajectoire des bras manipulateurs, le lecteur intéressé par la planification de trajectoires pour des plates-formes mobiles peut se référer aux travaux de [Reeds et al., 1990], [Fleury et al., 1995] et [Lamiroux et al., 1997] entre autres.

Le problème de la génération de trajectoire pour des bras manipulateurs est largement traité dans la littérature, en particulier deux références attirent fortement notre attention [Brady et al., 1982] et [Khalil et al., 1999]. Le lecteur pourra se reporter à ces ouvrages pour plus de détails.

1.4.1 Génération de trajectoires en temps minimal par optimisation

Le problème de la génération de trajectoire est couramment présenté comme un problème de satisfaction de contraintes : une description polynomiale du chemin à parcourir est utilisée pour minimiser un certain critère de performance. Un polynôme d'ordre 1 peut être utilisé pour des déplacements à vitesses constantes, un polynôme d'ordre trois peut être utilisé pour satisfaire des contraintes de vitesse et un polynôme d'ordre cinq peut être utilisé pour satisfaire des contraintes de position, de vitesse et d'accélération. Le critère le plus couramment utilisé cherche à minimiser le temps de parcours de la trajectoire, ceci transforme le processus de génération de trajectoire en un problème de commande optimale. Nous avons choisi de diviser le problème de commande optimale en deux étapes : la génération de trajectoire et l'asservissement. De cette façon, le calcul de la trajectoire optimale se réduit à optimiser la trajectoire pour satisfaire des contraintes cinématiques de la tâche et des contraintes dynamiques du robot. Suivant cette approche, nous pouvons citer les travaux de [Sahar et al., 1986], [Geering et al., 1986], [Shin et al., 1986], [Pfeiffer et al., 1987], [Bobrow, 1988], entre autres. Dans [Slotine et al., 1989], la trajectoire en temps minimal est trouvée en choisissant les instants de commutation du couple appliqué à chaque articulation, en considérant qu'au moins un actionneur a le couple saturé. Won propose une méthode pour le mouvement point à point qui minimise l'énergie du mouvement en utilisant l'analyse de Fourier pour supprimer les composantes de haute fréquence et par conséquent améliorer la souplesse du mouvement [Won et al., 1991]. Piazzi présente une méthode pour minimiser le jerk pendant le processus de génération de trajectoire dans l'espace articulaire [Piazzi et al., 1997].

1.4.2 Génération de trajectoires par satisfaction des contraintes cinématiques

D'un point de vue strictement cinématique, les travaux de Paul [Paul, 1979], Taylor [Taylor, 1979] et Lin [Lin et al., 1983] sont des références classiques sur la planification de trajectoire.

Paul en utilisant la représentation par matrice de passage homogène définit « l'équation fondamentale de la manipulation » qui relie l'objet à saisir et l'organe terminal avec la configuration du bras manipulateur. En faisant des interpolations en ligne droite pour la position et des rotations autour d'un axe fixe, le mouvement entre deux positions est réalisé, par une séquence de rotation - translation - rotation. Le mouvement est effectué avec des accélérations constantes pour des variations linéaires de vitesse. Pour la transition entre deux segments rectilignes, il propose des changements symétriques d'accélération pendant une période 2τ , la méthode garantit la transition entre vitesses de façon continue. La configuration des articulations du bras manipulateur est obtenue à partir des matrices interpolées et une interpolation dans l'espace des configurations est réalisée pour garantir la continuité du mouvement dans l'espace articulaire.

Taylor propose deux méthodes pour les mouvements en ligne droite, la première est une amélioration de la méthode de Paul et utilise des quaternions pour représenter les rotations. Il traite les rotations comme des positions pour obtenir un mouvement uniforme. La transition entre deux segments rectilignes est présentée selon cette nouvelle formulation du problème. La deuxième méthode cherche à corriger au moyen d'une deuxième étape de planification la trajectoire au niveau des articulations.

Lin par contre ne considère pas le mouvement dans l'espace cartésien, sa méthode considère la transformation du chemin de l'espace cartésien vers l'espace articulaire. En utilisant des *spline* cubiques et

des contraintes en vitesse, accélération et jerk, il cherche la trajectoire qui relie les points. Cette trajectoire est soumise à une étape d'optimisation pour déterminer la trajectoire qui satisfait les contraintes en temps minimal.

Les méthodes de Paul et de Taylor proposent des lois d'accélération de type bang-bang. Si nous souhaitons limiter le jerk, les lois bang-bang, ne pouvant satisfaire cette contrainte, ne peuvent pas être retenues. Dans un communiqué Castain [Castain et al., 1984] propose de changer la loi bang-bang par une loi trapèze. Pour résoudre le problème de génération de trajectoire, Castain définit un point intermédiaire entre deux points consécutifs du chemin, les trois splines cubiques qui définissent la loi trapèze d'accélération sont utilisés pour atteindre le point intermédiaire depuis le point initial et trois autres splines sont utilisées pour relier le point intermédiaire et le point final. En utilisant cette méthode, six splines cubiques sont utilisées pour relier les points du chemin. Dans la conclusion de ce communiqué, Castain propose l'utilisation des différents points intermédiaires pour réduire les phénomènes de promenade. Un autre avantage apparaît, l'utilisation des lois d'accélération trapèze implique l'utilisation de jerks bornés. Cette méthode est utilisée par les travaux de [Bazaz et al., 1997] pour des applications dans l'espace cartésien. Khalil présente une loi d'accélération du deuxième ordre qui simplifie le calcul de la trajectoire, cependant ce type de loi ne satisfait pas la contrainte de jerk. En suivant cette démarche, les travaux de [Chand et al., 1985] proposent l'utilisation des splines cubiques par estimation de la vitesse de passage de chaque point intermédiaire du chemin en fonction des points suivants. Ce type de technique peut être utilisé soit dans l'espace articulaire soit dans l'espace opérationnel.

Pour remplacer les splines, la littérature propose d'autres approches : Thompson propose l'utilisation des *B-splines* à la place des splines et présente un algorithme pour la solution du problème avec des contraintes en accélération et en vitesse [Thompson et al., 1987], Ozaki cherche la solution optimale pour les points de contrôle de la B-spline [Ozaki et al., 1996], Zalzal propose une stratégie distribuée pour le calcul de trajectoires en utilisant des splines cubiques [Zalzal et al., 1991], Yang propose l'utilisation des *NURBS* dans une trajectoire calculée hors-ligne pour des courbes complexes [Yang et al., 1997] et Aleotti utilise les *NURBS* pour reconstruire les trajectoires exécutées par l'homme dans un processus de commande par apprentissage [Aleotti et al., 2005]. Froissart propose l'utilisation des courbes de Bézier pour reconstruire un polynôme d'ordre cinq [Froissart et al., 1993], cette méthode demande la connaissance de la vitesse et de l'accélération de sortie du point de départ et la position des deux points suivants. Müller propose l'utilisation des trajectoires hyperboliques pour des tâches de « saisie-et-dépose (Pick-and-Place en anglais) » par définition des trois points : un pour l'origine, un autre pour la destination et un troisième qui permet de déformer la trajectoire dans le cas d'évitement des obstacles [Müller-Karger et al., 2000]. Van Aken impose un profil d'accélération pour deux points du chemin pour des mouvements sans contraintes [Aken et al., 1988]. Volpe propose l'utilisation de trois types de fonctions pour fusionner les courbes de vitesse de segments successifs [Volpe, 1993].

1.4.3 Génération de trajectoire en ligne

Bestaoui propose une méthode qui utilise des polynômes du quatrième degré avec des contraintes en vitesse et en accélération pour des applications en ligne par actualisation de la trajectoire à chaque temps d'échantillonnage [Bestaoui, 1989]. Cette méthode est reprise par les travaux de [Bazaz et al., 1997].

Andersson propose l'utilisation des polynômes d'ordre 5 (quintiques) pour résoudre le problème de génération de trajectoire avec des conditions initiales et finales d'accélération, de vitesse et de position imposées et des temps de déplacement fixes [Andersson, 1989].

Lloyd considère l'utilisation des polynômes quintiques pour des mouvements en ligne droite entre deux points. Les deux polynômes qui définissent les deux segments consécutifs sont mélangés en utilisant une fonction polynomiale [Lloyd et al., 1993].

Tondu propose une amélioration du schéma proposé par Paul avec des mouvements en ligne droite entre points successifs du chemin et des virages pour les transitions. La méthode peut aussi bien être appliquée dans l'espace cartésien que dans l'espace opérationnel. Une contrainte nécessaire pour la réussite de cette méthode porte sur la durée de déplacement sur un segment qui devra être plus grande que la somme des temps de transition [Tondu et al., 1994]. Une question reste ouverte par rapport à cette méthode, est-ce que les vitesses angulaires peuvent être intégrées pour obtenir l'orientation ?

Bazaz et Tondu introduisent une méthode à trois splines cubiques dans [Bazaz et al., 1998], une description plus détaillée est présentée dans [Tondu et al., 1999]. En utilisant quatre splines cubiques en quatre combinaisons possibles, le mouvement est réalisé entre deux points avec des vitesses (initiale et finale) imposées et des accélérations (initiale et finale) nulles. Dans ses travaux, Tondu aborde le phénomène de promenade autour du chemin par l'augmentation de la dérivé de l'accélération. cette solution ne peut être retenue dans notre contexte où le jerk est borné.

Liu propose une méthode à sept segments cubiques pour relier deux points avec des contraintes en jerk, accélération et vitesse [Liu, 2002]. Cette méthode est traitée uniquement en une dimension. La trajectoire est calculée pour un mouvement point à point. Si la destination change pendant le déplacement, la trajectoire est actualisée en définissant une vitesse de transition avec une accélération nulle. Cette vitesse est calculée à partir des conditions de mouvement courantes à l'instant où la nouvelle destination est définie. Mcfarlane propose l'utilisation de 7 segments quintiques à la place des segments cubiques [Macfarlane et al., 2003]. Cette extension est utilisée pour des mouvements en point à point en ligne droite dans l'espace cartésien. Pour un chemin, la méthode proposée considère des mouvements en ligne droite entre points successifs et propose des fonctions polynomiales qui respectent les contraintes pour les relier.

Nous proposons une méthode à sept segments avec des contraintes en jerk, accélération et vitesse dans les cas à plusieurs dimensions pour des applications en temps réel comme les systèmes de guidage visuel.

1.5 Asservissement Visuel

L'objectif visé, par l'introduction de la vision dans la boucle de commande, est d'augmenter l'autonomie des robots par rapport aux variations de l'environnement. La vision est un processus de traitement de l'information qui utilise des stratégies bien connues afin d'atteindre ces objectifs. L'entrée d'un système de vision est constituée par une séquence d'images et la sortie est une description de l'entrée en termes d'objets et de relations entre ces objets. Le système de vision doit fournir les connaissances nécessaires afin de permettre une interprétation non ambiguë [Horaud et al., 1993]. Nous parlons alors d'asservissement en situation, ou encore de commande référencée capteurs [Chaumette, 1990].

Nous devons considérer que nous ne sommes pas capables de faire un système de vision « Universel ». Au lieu de cela, nous nous intéressons aux systèmes qui accomplissent une tâche particulière dans un environnement contrôlé ou aux modules qui pourraient par la suite devenir une partie d'un système d'usage universel [Klaus et al., 1986].

L'utilisation du retour visuel pour commander un robot se nomme généralement asservissement visuel (Visual Servoing en anglais). L'asservissement visuel est étudié sous diverses formes depuis plus de deux décennies, en commençant avec des tâches simples de saisie-et-dépose pour arriver actuellement à la manipulation d'objets en temps réel. Une excellente introduction à l'asservissement visuel sous la forme d'un « tutorial » est présentée par Hager *et al.* en 1996 [Hager et al., 1996] suivi du travail de Kragic et Christensen qui proposent en 2001 une vue d'ensemble sur l'asservissement visuel pour la manipulation [Kragic et al., 2002]. Plus récemment un tutorial en deux parties sur l'asservissement visuel est présenté par Chaumette et Hutchinson [Chaumette et al., 2006] et [Chaumette et al., 2007].

En 1979, Hill et Park ont introduit l'expression asservissement visuel, pour distinguer leur travaux des travaux précédant. Les systèmes d'asservissement visuel sont classés selon deux approches, soit en fonction de la configuration du système, soit en fonction du mode dans lequel l'asservissement est réalisé

Les configurations caméra-robot suivantes peuvent être rencontrées :

- configuration de caméra embarquée ou « Eye-in-Hand ». La caméra est placée sur l'organe terminal du bras manipulateur ;
- configuration de caméra déportée ou « Stand-Alone ». La caméra est placée ailleurs et regarde la scène. En opposition à Eye-in-hand, cette configuration est aussi appelé « Hand-to-eye » ;
- Eye-in-Hand stéréo. Les caméras embarquées forment une paire stéréo ;
- Stand-Alone stéréo. Une paire stéréo regarde la scène ;
- système multi-caméra. Plusieurs caméras combinant les configurations précédentes.

Les modes d'asservissement classiques sont : l'asservissement en boucle ouverte et l'asservissement en boucle fermée que nous décrivons dans les paragraphes suivants.

1.5.1 Asservissement en boucle ouverte

Cette approche est aussi appelée « Static Look & Move ». L'extraction de l'information de l'image et la commande d'un robot sont deux tâches séparées. Tout d'abord le traitement d'image est exécuté, et ensuite, une commande devant conduire à la réalisation du déplacement est générée. Prenons par exemple la tâche de saisie d'un objet, pour déterminer la position et l'orientation de l'objet à saisir relativement par rapport au robot, nous avons besoin de :

- un modèle de l'objet pour l'identifier sur l'image ;
- une caméra calibrée pour déterminer la position et l'orientation de l'objet par rapport à la caméra ;
- toutes les relations géométriques connues. Par exemple, dans le cas d'une configuration caméra embarquée, les modèles géométriques du robot et la situation de la caméra par rapport à l'organe terminal ;
- de scènes « statiques » pendant le déplacement du robot, car les mouvements sont réalisées « en aveugle ».

Cette approche présente rapidement plusieurs insuffisances [Huynh, 1998] :

- l'incapacité de compenser les incertitudes internes (sur le modèle géométrique du robot, sur le traitement d'image) ;
- l'impossibilité de réaliser une tâche avec un objet mobile ;
- la commande en « aveugle » du robot.

1.5.2 Asservissement en boucle fermée

Le principe de l'asservissement visuel consiste à prendre en compte des informations visuelles issues d'une ou plusieurs caméras dans la boucle de commande d'un robot afin d'en contrôler le mouvement. Les techniques d'asservissement visuel utilisent la perception afin d'améliorer l'action et augmentent donc la flexibilité et la précision des systèmes robotiques [Malis, 1998].

Une tâche d'asservissement visuel typique inclut habituellement une certaine forme de positionnement ou de suivi. Un exemple de positionnement consiste à aligner l'organe terminal du robot avec la cible. Un exemple de suivi consiste à maintenir une distance constante entre l'organe terminal du robot et une cible mobile, ou entre le mouvement de l'organe terminal et la cible (suivi d'une trajectoire).

Dans les deux cas, l'information visuelle est employée pour mesurer l'erreur entre la position courante du robot et sa référence ou position désirée. L'asservissement visuel est ainsi un processus de régulation en boucle fermée. Deux stratégies sont présentées :

Systèmes « Dynamic Look & Move ». Ces systèmes effectuent la commande du robot en deux étapes : le système de vision fournit les consignes au contrôleur du robot qui utilise alors le retour des positions articulaires pour définir la boucle de commande interne du robot.

Systèmes « Direct Visual Servo ». Ici, le système de commande visuel calcule directement les positions des liaisons du robot et le contrôleur interne du robot est éliminé.

Dans les années 80, une taxinomie transversale a été présentée dans [Weiss et al., 1987] : asservissement 3D et asservissement 2D, des méthodes hybrides ont ensuite été introduites dans les années 90.

Systèmes d'asservissement 3D

Aussi appelé système d'asservissement visuel « basé position ». Puisque les mesures obtenues dans l'image sont employées pour déterminer la position et l'orientation de la cible par rapport à la caméra ou à un repère commun du monde. L'erreur entre la position courante et la position désirée de la cible est définie dans l'espace cartésien 3D de la tâche du robot. Par conséquent, l'erreur est une fonction des paramètres de position.

En général, l'avantage principal de cette approche est que la trajectoire du système caméra/robot est commandée directement dans un repère cartésien. Ceci permet une planification de trajectoires plus facile. Cependant, particulièrement dans le cas de la configuration caméra sur bras, les repères visuels utilisés pour l'évaluation de la position peuvent sortir de l'image. Si la caméra est grossièrement calibré, les positions courantes et désirées de la caméra ne seront pas estimées exactement, ce qui mènera à de faibles performances.

Systèmes d'asservissement 2D

Aussi appelé système d'asservissement visuel « basé image ». Ici, le système utilise l'information 2D de l'image pour estimer le mouvement désiré du robot. Des tâches typiques comme le suivi et le positionnement sont accomplies en réduisant l'erreur entre un ensemble de primitives but et un ensemble de primitives courantes du plan image.

L'asservissement visuel 2D implique le calcul de la matrice d'interaction qui relie les coordonnées du repère de la scène à celles du repère de la caméra (soit le repère de la caméra, soit le repère de la scène est attaché au robot). Le calcul de la matrice d'interaction exige la connaissance des paramètres intrinsèques et extrinsèques de la caméra. Si une seule caméra est employée pendant le processus d'asservissement, l'information de profondeur requise pour mettre à jour la matrice d'interaction de l'image est perdue. La plupart des systèmes existants utilisent une matrice d'interaction constante qui est calculée pour la position désirée de la caméra. Ceci constitue un des inconvénients de cette approche, car la convergence n'est assurée qu'autour de la position désirée.

L'asservissement visuel 2D est considérée comme une commande très robuste par rapport aux erreurs de calibration de la caméra et du robot. La calibration faible affecte seulement le taux de convergence de la loi de commande dans le sens où un temps plus long est nécessaire pour atteindre la position désirée.

Méthodes hybrides

Malis et al. ont présenté l'asservissement visuel 2D 1/2 en 1998 [Malis et al., 1998]. La méthode a été proposée à l'origine pour la configuration caméra embarquée. Cette approche est à mi-chemin entre les approches 2D et 3D. Elle évite leurs inconvénients respectifs : contrairement à l'approche 3D, elle ne nécessite pas de modèle géométrique 3D de l'objet. Par rapport à l'approche 2D, elle assure la convergence de la loi de commande dans tout l'espace de la tâche. La méthode est basée sur l'évaluation du déplacement de la caméra entre la vue courante et la vue désirée de l'objet. A chaque itération, la rotation entre ces deux vues est estimée pour que les boucles de translation et de rotation soient découplées.

Crétual et al. ont présenté une méthode qui utilise des informations visuelles dynamiques à la place de motifs visuels géométriques. Les mouvements de la caméra sont alors contrôlés pour que le mouvement 2D mesuré atteigne un champ de vitesse désiré. Cette technique est appelé asservissement visuel $d2/dt$ dans [Chaumette, 1998]

1.6 La commande d'un bras manipulateur pour la robotique de service

Reprenons maintenant le schéma de la figure 1.1, à partir des thèmes que nous avons abordé : la planification du chemin, la commande des bras manipulateurs, la génération du mouvement et l'asservissement visuel.

Nous pourrions dire que nous avons présenté un aperçu de *l'état de l'art* pour quatre des thèmes de la robotique. Là n'était pas notre principal objectif, et le résultat est loin d'être complet. Notre objectif était plutôt de montrer que le problème auquel nous nous intéressons a une très grande variété de solutions. En établissant le contexte général des différentes approches, nous pouvons mettre en évidence notre approche et focaliser l'attention sur les différentes hypothèses que nous avons faites au cours du

développement de notre schéma de commande. Tous les choix ont été faits en fonction du cadre applicatif et seront détaillés dans les chapitres suivants.

Le premier bloc de notre schéma correspond à l'étape de génération du chemin. Notre intérêt pour ce thème se limite à l'utilisation des chemins obtenus par ce type de techniques. L'évitement de collision avec l'environnement et le robot lui même, la satisfaction des contraintes de manipulabilité et la sélection du chemin dans le champ de vision de l'utilisateur humain pour augmenter son confort, sont des problèmes qui doivent être résolus à ce niveau.

Le deuxième bloc concerne l'étape de la commande du bras manipulateur. Nous traiterons de façon séparée chacune des boucles de commande, mais l'objectif que nous poursuivons est de les présenter comme un ensemble cohérent.

Avant de continuer, nous devons présenter les plates-formes expérimentales. Nos travaux ont commencé sur le robot « H2 bis » qui utilise le système d'exploitation *VxWorks* et ils ont été poursuivis sur le robot *Jido* qui utilise le système d'exploitation *Linux*. Les deux sont constitués d'une plate-forme mobile avec un bras manipulateur doté d'un préhenseur et d'un système de vision embarqué. Les deux plates-formes ne fournissent que des informations sur la position angulaire de chacune des articulations des bras manipulateurs et utilisent comme actionneurs des servomoteurs avec de forts rapports de réduction.

Au niveau de l'asservissement des articulations, seule les boucles de commande du robot H2 bis sont accessibles en programmation. Notre première approche de la commande du bras est dédiée à l'amélioration de la boucle de commande des articulations, nous cherchons à augmenter ses performances en respectant les contraintes d'accélération et de vitesse de chacune des articulations.

Pour l'asservissement sur la trajectoire, nous avons considéré un schéma de commande indépendant du modèle dynamique du robot pour pouvoir l'adapter plus facilement à des plates-formes différentes. Comme les contraintes de la robotique de service sont différentes de celles de la robotique industrielle et les servomoteurs ont de forts rapports de réduction, les schémas de commande dynamique ne sont pas indispensables. Par ailleurs, nous désirons un schéma de commande qui opère dans l'espace cartésien pour pouvoir intégrer des capteurs extéroceptifs plus facilement. Donc, les schémas de type commande jacobienne sont des schémas simples qui satisfont nos besoins.

Afin d'améliorer le confort des utilisateurs humains qui sont au centre des applications en robotique de service, nous proposons d'effectuer des mouvements souples. Notre objectif est d'intégrer l'étape de génération du mouvement à l'intérieur de la boucle de commande afin de respecter des contraintes de jerk, d'accélération et de vitesse qui assurent la souplesse du mouvement.

Finalement, pour doter un robot d'une certaine autonomie et l'aider à gérer les imprécisions, notre schéma devra être capable d'utiliser les informations provenant du système de vision. Trois schémas différents seront présentés : les deux premiers considèrent des approches classiques de l'asservissement visuel alors que le troisième propose l'intégration d'un filtre de Kalman pour gérer les incertitudes dans la modélisation du système de vision.

Chapitre 2

Commande des bras manipulateurs

2.1 Introduction

Ce chapitre se propose de traiter de la commande des robots manipulateurs pour des applications en robotique de service. Tout d'abord, nous devons considérer qu'un robot de service, bien qu'ayant beaucoup de similarités avec les robots industriels rencontre des conditions opérationnelles très différentes. Un robot industriel réalise des tâches répétitives à l'intérieur d'espaces sécurisés pour ses opérations. Les contraintes de performance retenues sont généralement liées aux temps d'exécution. En opposition, pour développer un robot de service capable d'évoluer dans un milieu humain, qui la plupart du temps n'est pas statique, nous devons considérer des contraintes différentes de celles de la robotique industrielle. Par exemple, l'interaction avec l'environnement a plus d'importance que le temps mis pour accomplir la tâche. Donc, les vitesses opérationnelles restent plus faibles que les vitesses développées par les robots industriels. La sécurité et le confort des utilisateurs humains sont alors des thèmes de première importance pour la robotique de service.

Pour contribuer à augmenter la sécurité opératoire des robots de service, les schémas de commande doivent prendre en compte l'information provenant des capteurs proprioceptifs des boucles de commande de plus bas niveau et celle des capteurs extéroceptifs pour les boucles supérieures. Le schéma de commande que nous proposons pour le plus bas niveau n'utilise que des capteurs proprioceptifs et reste ouvert pour l'inclusion de capteurs extéroceptifs pour les boucles de commande de plus haut niveau. Le schéma de la figure 1.1 présente cette idée de façon simplifiée.

Plus généralement, la fonction d'un robot manipulateur est de situer dans l'espace son *organe terminal OT*. La *Situation* de l'organe terminal est définie par la position et l'orientation d'un repère lié à son but. Une tâche simple peut être définie comme le mouvement libre réalisé par le robot manipulateur pour aller d'une situation initiale à une situation finale. Une tâche complexe peut être définie de la même façon, en précisant que le robot doit suivre un chemin prédéfini tout en respectant des contraintes cinématiques liées au mouvement. Dans les deux cas, le mouvement que le robot réalise devra être défini par des consignes qui correspondent à des trajectoires temporelles. Pour contribuer à augmenter le confort de l'utilisateur humain, nous proposons l'utilisation d'un générateur de mouvements souples pour le calcul des trajectoires, le chapitre 3 est dédié à ce sujet. Les mouvements souples sont caractérisés par des limites en jerk, en accélération et en vitesse. Ces contraintes qui devront être respectées au niveau des schémas de commande, servent de guide pour les schémas que nous développons.

Dans ce chapitre, nous allons étudier des boucles de commande utilisant uniquement des capteurs proprioceptifs pour les signaux de retour. L'utilisation des signaux extéroceptifs de retour sera abordée au chapitre 4.

Le schéma de commande que nous proposons découpe le problème de commande en deux : la première étape correspond à la génération de consignes, cette étape est développée dans le chapitre 3. La deuxième étape correspond à la loi de commande pour l'asservissement du bras manipulateur, elle prend en compte les contraintes cinématiques de la trajectoire.

Ce chapitre est organisé de la façon suivante, tout d'abord nous détaillons quelques généralités pour

la sélection de la loi de commande. Ensuite nous présentons des améliorations réalisées pour l'asservissement des articulations. La section suivante présente la boucle de commande du bras manipulateur pour l'asservissement dans l'espace cartésien. Finalement, des conclusions sont présentées à la dernière section.

2.2 Généralités sur la sélection de la loi de commande

Tout d'abord, un robot manipulateur est une machine constituée par :

- une structure mécanique de support ;
- des actionneurs ;
- des capteurs ;
- un système de commande.

Donc, nous pouvons définir un robot manipulateur comme une structure mécanique complexe dont les caractéristiques dynamiques varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations.

Dans ce travail, nous n'abordons pas la construction d'un robot manipulateur. Nos travaux portent sur l'utilisation d'un bras manipulateur à six degrés de liberté du type $6R$.

La plupart des robots utilisent des servomoteurs électriques comme actionneurs. Les caractéristiques des servomoteurs ont un rôle important pour la sélection du système de commande. Dans le cas où le robot utilise des servomoteurs avec de faibles rapports de réduction, les boucles de commande doivent compenser les effets des variations des forces d'inertie et de gravité. Les lois de commande basées sur les modèles dynamiques des robots donnent de très bon résultats dans ce cas. Ils permettent de maintenir la réponse dynamique du système dans certains critères de performance. En utilisant ce type de techniques les robots peuvent évoluer à grandes vitesses. L'inconvénient est lié à la forte dépendance au modèle dynamique. En opposition, quand les articulations sont actionnées par l'intermédiaire de réducteurs à forts rapports de réduction, l'inertie vue par les moteurs varie peu. Dans ce cas, les asservissements peuvent être assurés axe par axe par des boucles de commande classiques.

Pour les robots manipulateurs de type $6R$, les capteurs les plus couramment utilisés sont les capteurs de position et de vitesse angulaire. Nous n'utilisons que des capteurs de position absolue qui ne demandent pas de processus d'initialisation.

La boucle de commande que nous proposons pour le bras manipulateur profite des forts rapports de réduction des servomoteurs et des conditions opératoires de la robotique de service (faibles vitesses, accélérations et jerks).

2.2.1 Définition de la Situation de l'organe terminal

Pour définir la *situation* de l'organe terminal du robot dans l'espace, il faut préciser sa *position* et son *orientation*.

Pour définir la *position* de l'organe terminal dans l'espace, trois représentations sont couramment utilisées dans la littérature :

- Coordonnées cartésiennes ;
- Coordonnées cylindriques ;
- Coordonnées sphériques.

Nous utilisons la représentation en coordonnées cartésiennes pour sa simplicité, c'est aussi la plus couramment utilisée.

Pour définir l'*orientation* de l'organe terminal dans l'espace, plusieurs représentations sont présentées dans la littérature [Renaud, 1996]. Par exemple :

- Angles de Bryant ;
- Angles d'Euler ;
- Matrice de rotation ou cosinus directeurs ;
- Quaternion ;
- Angle plus vecteur.

La plus courante des représentations est la matrice de rotation qui a neuf composantes. Elle associe une représentation unique à chaque rotation. Les représentations par quaternion ou par angle plus vecteur sont des représentations à quatre composantes, une de plus que les angles d'Euler ou de Bryant sans les problèmes de singularités de ces représentations. Pour ce travail, nous avons utilisé les quaternions et dans une moindre mesure les matrices de rotation qui constituent la partie rotation des matrices homogènes. Pour plus de détails, par exemple sur les conversions entre représentations, consulter l'annexe A. Les avantages de l'utilisation des quaternions sont largement exposés dans [Funda et al., 1991].

2.2.2 Modèles Directs du robot manipulateur

L'obtention des modèles géométriques et cinématiques est uniquement basée sur des considérations géométriques.

Modèle géométrique direct

Le modèle géométrique direct d'un robot manipulateur est la fonction \mathbf{F} qui permet d'exprimer la situation de l'organe terminal du robot manipulateur en fonction de sa configuration.

La situation \mathcal{S} de l'organe terminal est définie par m coordonnées que nous appelons *opérationnelles* et que nous notons :

$$x_1, x_2, \dots, x_m$$

La configuration du robot manipulateur est définie par n coordonnées que nous appelons *généralisées* et que nous notons :

$$\theta_1, \theta_2, \dots, \theta_n$$

Si \mathbf{X} désigne le m -uplet des coordonnées opérationnelles et $\boldsymbol{\theta}$ le n -uplet des coordonnées généralisées, le modèle géométrique du robot manipulateur est :

$$\mathbf{X} = \mathbf{F}(\boldsymbol{\theta}) \quad (2.1)$$

Le détail du calcul de la fonction \mathbf{F} est présenté en annexe B.

Modèle cinématique direct

Le modèle cinématique direct permet de calculer le torseur cinématique

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

à partir des vitesses généralisées $\dot{\boldsymbol{\theta}}$ en utilisant la matrice jacobienne $\mathbf{J}(\boldsymbol{\theta})$:

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} \quad (2.2)$$

Pour des informations plus détaillées sur la construction de la matrice jacobienne, se référer à l'annexe B.

2.3 Asservissement des articulations

En introduction, nous pouvons considérer la phrase suivante : *La commande dynamique n'est pas dans tous les cas le type de commande nécessaire pour obtenir une bonne précision et une bonne stabilité. En effet une commande classique suffit lorsque le robot manipulateur évolue sans contraintes de performance, de rapidité et de précision car dans ce cas, les inerties ont une influence moins importante.* [Gorla et al., 1984]

2.3.1 La commande classique

Notre approche pour l'asservissement des articulations est fondé sur une commande classique. Dans [Paul, 1981], [Gorla et al., 1984], [Canudas et al., 1997] et [Sciavicco et al., 2000] nous pouvons trouver une série de schémas pour l'asservissement des articulations. Nous pouvons observer différentes façons de calculer le couple pour obtenir un suivi correct de la trajectoire par compensation de l'inertie et/ou des forces de gravité. Les schémas présentés par [Gorla et al., 1984] utilisent la vitesse comme signal d'anticipation alors que ceux présentés par [Paul, 1981] utilisent la vitesse de deux façons : soit comme signal d'anticipation, soit dans une deuxième boucle de commande.

En définissant les variables θ pour la position, $\dot{\theta}$ pour la vitesse, et les consignes θ_d pour la position désirée et $\dot{\theta}_d$ pour la vitesse désirée. Les lois de commande u ont la forme d'une commande du type proportionnel plus dérivée (P + D) :

$$u(t) = K_p(\theta_d - \theta) - K_v\dot{\theta} \quad (2.3)$$

$$u(t) = K_p(\theta_d - \theta) + K_v\dot{\theta}_d \quad (2.4)$$

Des extensions sont présentés dans [Canudas et al., 1997] et [Sciavicco et al., 2000], que nous simplifions en :

$$u(t) = K_p(\theta_d - \theta) + K_i \int (\theta_d - \theta)dt - K_v\dot{\theta} \quad (2.5)$$

où K_p , K_i et K_v sont les gains du correcteur. Une méthode pour calculer les gains est proposé par [Canudas et al., 1997].

Nous pouvons considérer qu'il y a deux types d'asservissement, *l'asservissement de position* et *l'asservissement de vitesse*. Dans le premier cas, la grandeur de consigne à la même nature que la sortie à asservir. Dans le deuxième cas, elles sont de nature différente. L'utilisation de l'un ou de l'autre dépend de l'application.

Nous avons établi que les conditions opératoires en robotique de service sont faibles par rapport aux conditions opératoires en robotique industrielle pour des raisons de sécurité et de confort entre autres. Notre approche est basée sur un contrôleur de type(P + I) de la forme :

$$u(s) = C(1 + \frac{1}{T_i s}) \quad (2.6)$$

où s est la variable de Laplace, C est le gain du contrôleur et T_i est le temps intégral. Les algorithmes de commande basés sur un schéma P + I présentent un gros inconvénient : la saturation de la partie intégrale. Cet inconvénient doit être corrigé pour obtenir de bonnes performances.

2.3.2 La commande par ordinateur

Pour la commande par ordinateur deux algorithmes ont été proposés [Mahmoud, 1991]. Le premier algorithme connu sous le nom de l'algorithme de position, calcule la nouvelle loi de commande à chaque instant d'échantillonnage. Le deuxième algorithme, appelé algorithme de vitesse, calcule l'incrément ou *delta* de la loi de commande. Nous avons retenu l'algorithme de vitesse, il prend la forme :

$$u[kT_s] = u[(k - 1)T_s] + \Delta u[kT_s] \quad (2.7)$$

avec

$$\Delta u[k\mathbf{T}_s] = C \left((e[k\mathbf{T}_s] - e[(k-1)\mathbf{T}_s]) + \frac{\mathbf{T}_s}{T_i} e[k\mathbf{T}_s] \right) \quad (2.8)$$

où \mathbf{T}_s est le temps d'échantillonnage. Nous utiliserons la notation $[k]$ pour $[k\mathbf{T}_s]$ afin d'alléger les expressions dans la suite du document.

L'algorithme de vitesse permet de corriger le problème de saturation de la partie intégrale de la loi de commande en utilisant une fonction de saturation en sortie du calcul de $u[k]$.

La fonction de saturation

Définissons la fonction de saturation $S(v)$ telle que :

$$S(v) = \begin{cases} v_{max} & \text{si } v \geq v_{max} \\ v & \text{si } v_{min} < v < v_{max} \\ v_{min} & \text{si } v \leq v_{min} \end{cases} \quad (2.9)$$

La stabilité du système en utilisant ce type de fonctions est traitée dans [Glattfelder et al., 1983].

Le schéma complet

Nous définissons l'erreur de position angulaire e_P par :

$$e_P = \theta_d - \theta$$

et l'erreur de vitesse angulaire e_V par :

$$e_V = \dot{\theta}_d - \dot{\theta}$$

où l'indice P désigne la position et V la vitesse.

Nous pouvons calculer les incréments des lois de commande pour l'asservissement en vitesse $\Delta u_V[k]$ et $\Delta u_P[k]$ pour l'asservissement en position. Nous utilisons une fonction de saturation pour limiter chacun des deltas. Pour donner au système la capacité d'être asservi en vitesse ou en position, un sélecteur permet de choisir entre les deux types d'asservissement. La loi de commande $u[k]$ peut être calculée après la sélection. Nous utilisons une fonction de saturation pour limiter la loi de commande. Nous pouvons construire le schéma d'asservissement de la figure 2.1.

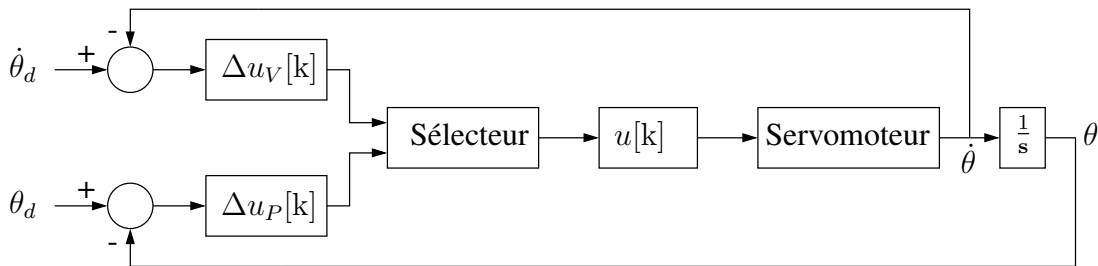


FIG. 2.1 – Schéma d'asservissement pour le servomoteur d'une articulation.

Comme la loi de commande ($u[k]$) n'est pas fonction de la grandeur d'entrée, nous pouvons faire la sélection du type d'asservissement (en position ou en vitesse) avant de la calculer. De cette façon, nous pouvons assurer un transfert sans sauts dans la loi de commande car c'est le Δu qui est sélectionné, et la loi de commande peut varier au maximum comme deux fois la valeur maximale de Δu .

En limitant la valeur de $u[k]$, le problème de saturation de la partie intégrale est corrigé. Un autre avantage de l'utilisation d'une fonction de limitation pour $u[k]$ est que les bornes correspondent aux limites en vitesse du servomoteur. On observe que la limitation de $\Delta u_V[k]$ et $\Delta u_P[k]$ correspond à une limitation de l'accélération du servomoteur. L'utilisation des deux fonctions de limitation pour $u[k]$ et pour $\Delta u_V[k]$ permet de respecter les contraintes cinématiques de chacune des articulations.

2.3.3 Exemple 1 : Changement de type d'asservissement

Considérons la boucle de commande de la Figure 2.1, nous envisageons de tester la réponse du système à une entrée échelon en position et le comportement du système quand il y a un changement de type d'asservissement.

Pendant les quatre premières secondes de simulation de l'asservissement en vitesse, pour un signal d'entrée de type échelon de position désirée ($\theta_d = 2$) et une trajectoire de vitesse ($\dot{\theta}_d$) calculée pour arriver à cette position, *mouvement imposé*. La fonction de saturation pour $\Delta u_V[k]$ et $\Delta u_P[k]$ a été fixée à $\mathbf{T}_s A_{max}$ avec $A_{max} = 2$ unités, celle de la fonction de saturation pour $u[k]$ a été fixée à $V_{max} = 1$ unité. Pendant la deuxième partie de la simulation, nous considérons un asservissement en position, avec une nouvelle position désirée ($\theta_d = 4$). La figure 2.2 montre les résultats obtenus en simulation.

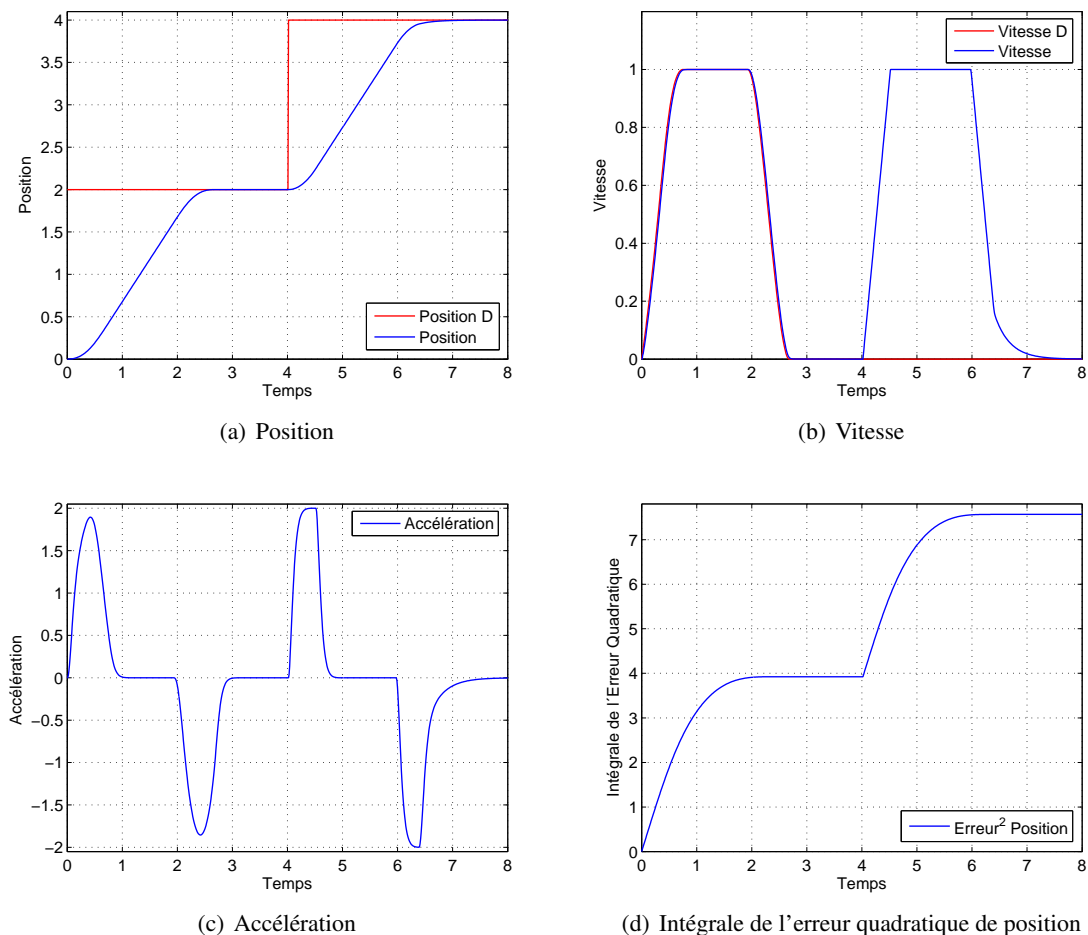


FIG. 2.2 – Évolution de la position, de la vitesse et de l'accélération au changement de type d'asservissement : passage de vitesse à position. La figure (d) montre l'intégrale de l'erreur quadratique de position.

Nous pouvons observer que :

- il n’y a pas de sauts au changement d’asservissement (Cf. figure 2.2.a) ;
- nous observons un bon suivi de la trajectoire de vitesse (Cf. figure 2.2.b) ;
- la vitesse maximale et l’accélération maximale ne sont pas dépassées (Cf. figure 2.2.b et 2.2.c) ;
- l’intégrale de l’erreur quadratique de position reste constante une fois que la position désirée est atteinte ($e_P \simeq 0$), tant en asservissement de vitesse qu’en asservissement de position. (Cf. figure 2.2.d) ;
- en utilisant l’asservissement en vitesse, l’erreur de position zéro ne dépend que de la consigne de vitesse, car il n’y a pas de retour de position pour la correction ;
- la souplesse des accélérations observées est plus grande dans l’asservissement en vitesse que dans l’asservissement en position, comme nous pouvons le vérifier sur la figure 2.2.d.

2.3.4 Exemple 2 : Changement du type d’asservissement avec consigne calculée de position

Dans l’exemple précédent, le changement de type d’asservissement a été effectué quand l’erreur de position était nulle. La position désirée est définie par un échelon. Dans cet exemple, la consigne de position ou position désirée θ_d est calculée comme l’intégrale de la vitesse désirée $\dot{\theta}_d$

$$\theta_d = \int \dot{\theta}_d$$

Il est préférable d’utiliser l’intégration de la vitesse plutôt que la dérivée de la position, car la dérivation se comporte comme un amplificateur des hautes fréquences alors qu’une intégration agit comme un filtre passe bas.

Le but de cet exemple est de tester un changement de type d’asservissement à un instant quelconque. La figure 2.3 et la figure 2.4 montrent les résultats obtenus.

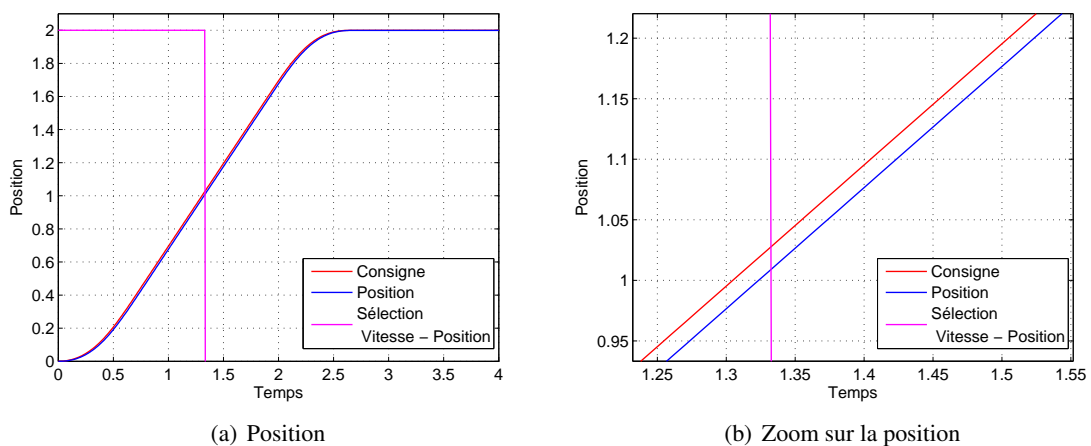


FIG. 2.3 – Évolution de la position et zoom sur la position au changement de type d’asservissement : passage de l’ asservissement en vitesse à l’asservissement en position.

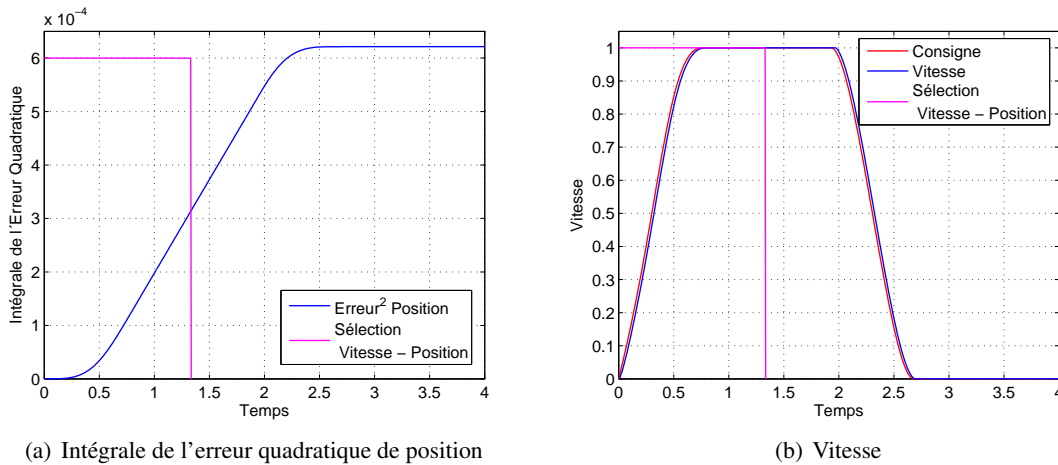


FIG. 2.4 – Intégrale de l'erreur quadratique de position et évolution de la vitesse au changement de type d'asservissement : passage de l'asservissement en vitesse à l'asservissement en position.

À l'instant où le changement du type d'asservissement est réalisé, nous n'observons pas de sauts dans la position du servomoteur. Nous pouvons définir la stratégie suivante :

1. pour le transfert d'une position désirée à une autre, une vitesse de consigne $\dot{\theta}_d$ doit être calculée et appliquée en utilisant l'asservissement en vitesse ;
2. lorsque la position du servomoteur est proche de la position finale désirée ou la trajectoire est finie, l'asservissement en position est utilisé pour corriger les erreurs de position.

2.3.5 Exemple 3 : Schéma avec signal d'anticipation

Les schémas précédents nécessitent deux signaux de retour, la position θ et la vitesse $\dot{\theta}$. Nous avons établi que les robots manipulateurs que nous utilisons ne possèdent que des capteurs de position, nous devons prendre en compte cette contrainte dans nos schémas.

Dans [Lambrechts et al., 2004] en utilisant quatre termes d'anticipation (dérivée du jerk, jerk, accélération et vitesse) avec une loi de type P + D et une étape de planification de trajectoire, la performance du système est améliorée.

En utilisant un seul signal d'anticipation, nous proposons le schéma de la figure 2.5 :

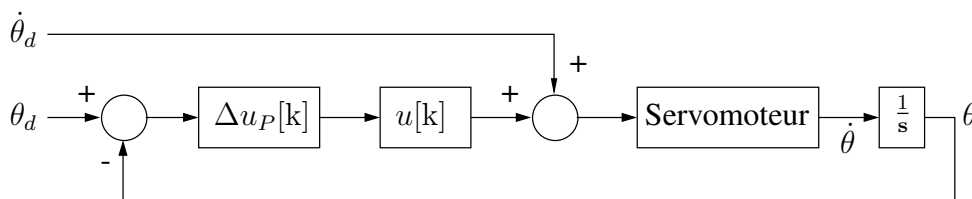


FIG. 2.5 – Schéma d'asservissement avec terme d'anticipation pour un servomoteur.

Le schéma de la figure 2.5 est équivalent à une loi de commande de type P + I avec signal d'anticipation. La loi de commande P + I est construite à partir des équations 2.7 et 2.8 avec les fonctions de saturation de l'équation 2.9. Comme signal d'anticipation nous utilisons la vitesse de consigne. Pour l'étape de planification de trajectoire, ou calcul des consignes, se référer au chapitre 3. Les résultats de l'utilisation de ce schéma sont montrés sur la figure 2.6.

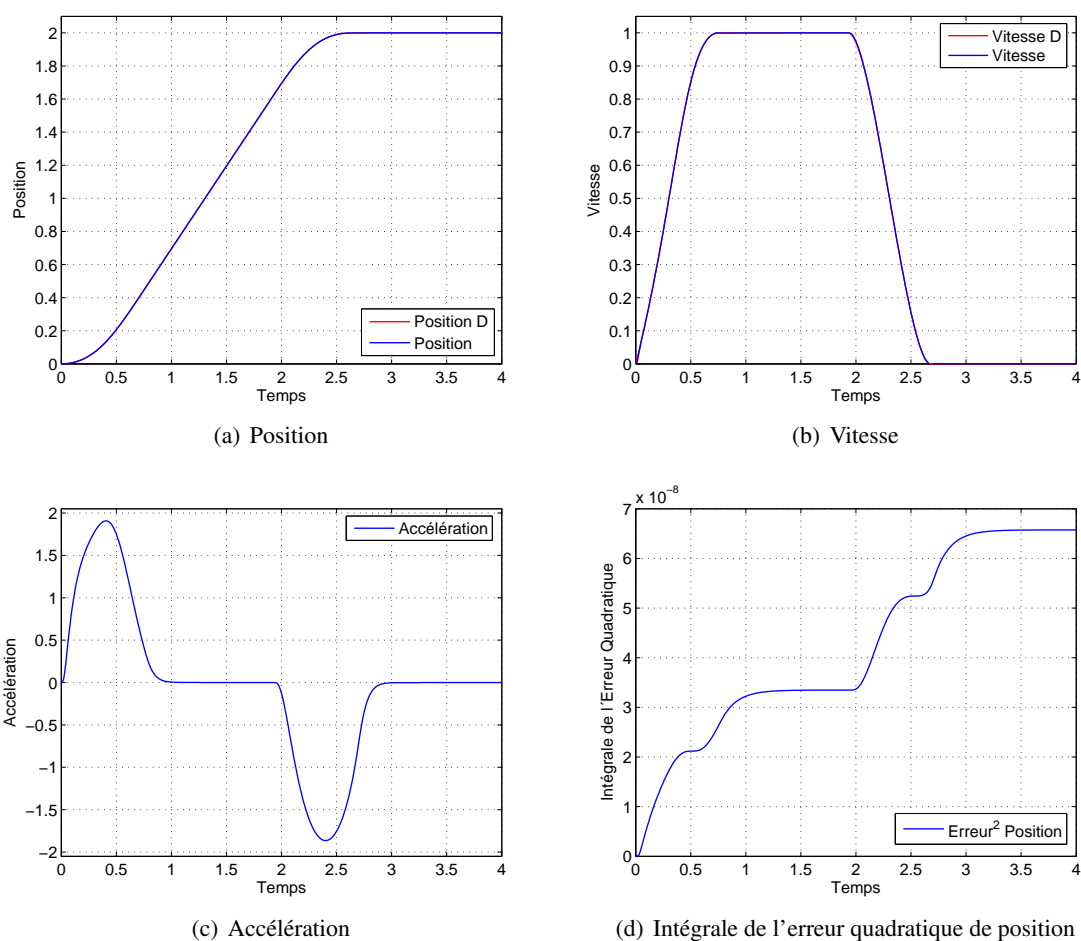


FIG. 2.6 – Évolution de la position, de la vitesse et de l'accélération en utilisant le schéma de commande saturé plus signal d'anticipation. La figure (d) montre l'intégrale de l'erreur quadratique de position

Ce schéma qui ne nécessite qu'un signal de retour, la position θ , et deux consignes θ_d et $\dot{\theta}_d$ présente des erreurs de position négligeables tout en respectant les contraintes d'accélération et de vitesse au niveaux des articulations.

2.4 Asservissement dans l'espace cartésien

Le problème de l'asservissement dans l'espace cartésien pour un bras manipulateur, consiste à produire les consignes de vitesses généralisées capables de réaliser le mouvement opérationnel.

2.4.1 Commande à résolution de vitesse par retour de quaternion

Considérons le modèle cinématique direct (*MCD*), par simplicité la matrice Jacobienne $\mathbf{J}(\theta)$ sera notée \mathbf{J} .

$$\begin{bmatrix} \mathbf{V} \\ \omega \end{bmatrix} = \mathbf{J}\dot{\theta}$$

Ce modèle définit les vitesses opérationnelles en fonction des vitesses généralisées. Whitney introduit la commande à résolution de vitesse [Whitney, 1969], les vitesses généralisées sont calculées en boucle ouverte en fonction des vitesses opérationnelles :

$$\dot{\theta} = \mathbf{J}^{-1} \begin{bmatrix} \mathbf{V}_d \\ \boldsymbol{\omega}_d \end{bmatrix} \quad (2.10)$$

De cette façon, pour une commande en boucle ouverte uniquement les vitesses opérationnelles $(\mathbf{V}_d, \boldsymbol{\omega}_d)$ doivent être calculées pour obtenir le mouvement opérationnel désiré.

Pour une commande en boucle fermée [Wu et al., 1982], la loi de commande est remplacée par

$$\dot{\theta} = \mathbf{J}^{-1} \begin{bmatrix} \mathbf{V}_d - \mathbf{K}_P \mathbf{e}_P \\ \boldsymbol{\omega}_d - \mathbf{K}_O \mathbf{e}_O \end{bmatrix} \quad (2.11)$$

où \mathbf{K}_P et \mathbf{K}_O sont des matrices de gain, et \mathbf{e}_P et \mathbf{e}_O représentent respectivement les vecteurs d'erreur de position et d'orientation.

Dans [Yuan, 1988], un schéma de commande qui utilise comme retour des quaternions est présenté. Nous proposons une version modifiée de ce schéma sur la figure 2.7

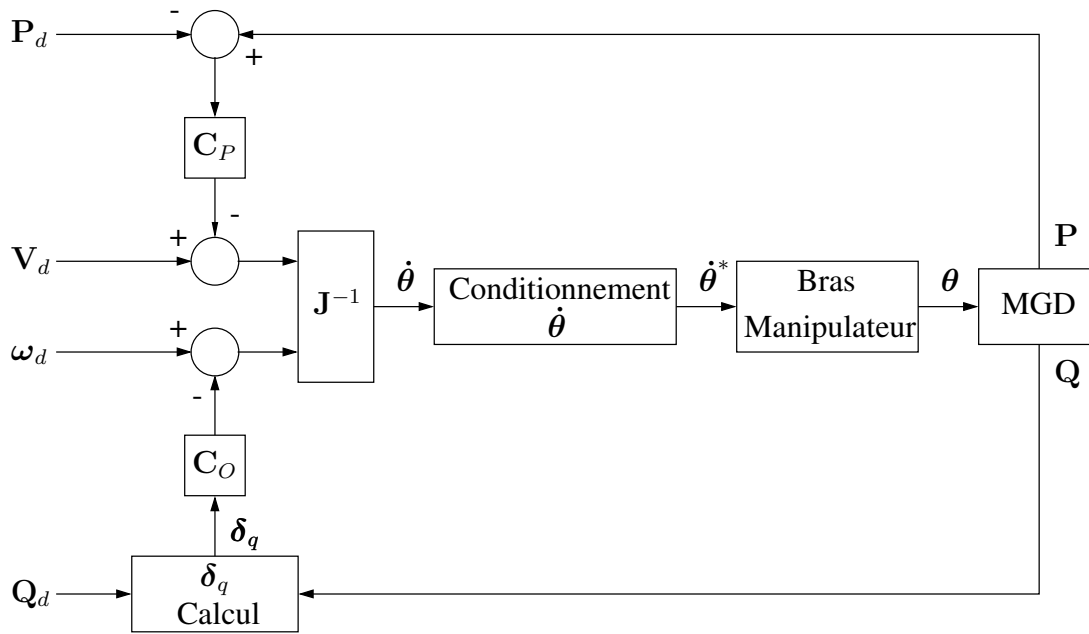


FIG. 2.7 – Schéma de commande pour le bras manipulateur utilisant des quaternions pour le retour d'orientation.

Les applications que nous envisageons permettent d'apporter des simplifications au schéma proposé par [Yuan, 1988]. Comme les contraintes cinématiques pour la robotique de service sont différentes, nous abandonnons le modèle dynamique au profit d'une fonction de conditionnement des vitesses.

Analyse de stabilité

Yuan propose des lois de commande du type P :

$$\mathbf{C}_P = \mathbf{K}_P \quad \mathbf{C}_O = \mathbf{K}_O \quad (2.12)$$

Pour l'analyse de stabilité, nous définissons la fonction de \mathcal{V}_P pour la position par :

$$\mathcal{V}_P = \frac{1}{2}(\mathbf{P} - \mathbf{P}_d)^T(\mathbf{P} - \mathbf{P}_d) \quad (2.13)$$

où l'indice d indique que ce sont des valeurs de consigne.

L'erreur de position \mathbf{e}_P est défini par :

$$\mathbf{e}_P = \mathbf{P} - \mathbf{P}_d \quad (2.14)$$

La stabilité du système est assurée pour une matrice \mathbf{K}_P définie positive. Pour plus de détails se référer à l'annexe C.1.

Nous définissons la fonction de \mathcal{V}_O pour l'orientation par :

$$\mathcal{V}_O = (n - n_d)^2 + (\mathbf{q} - \mathbf{q}_d)^T(\mathbf{q} - \mathbf{q}_d) \quad (2.15)$$

L'erreur d'orientation \mathbf{e}_O est défini par :

$$\mathbf{e}_O = n_d \mathbf{q} - n \mathbf{q}_d - \mathbf{q}_d \times \mathbf{q} \quad (2.16)$$

la stabilité du système est assurée pour une matrice \mathbf{K}_O définie positive. Pour plus de détails se référer à l'annexe C.2.

Amélioration des performances

Pour améliorer le suivi des trajectoires, nous remplaçons la loi de commande de type P par une loi de type P + I en utilisant les équations 2.7, 2.8 et 2.9. De la même façon que dans la section précédente, les limites des fonctions de saturation sont définis par les vitesses et les accélérations linéaires et angulaires que nous souhaitons pour l'organe terminal du bras manipulateur.

Les correcteurs de position et d'orientation ont alors la forme :

$$C_P = \begin{bmatrix} u_{Px}[k] \\ u_{Py}[k] \\ u_{Pz}[k] \end{bmatrix} \quad C_O = \begin{bmatrix} u_{Ox}[k] \\ u_{Oy}[k] \\ u_{Oz}[k] \end{bmatrix} \quad (2.17)$$

où

$$\begin{aligned} u_{Px}[k] &= u_{Px}[k-1] + \Delta u_{Px}[k] & u_{Ox}[k] &= u_{Ox}[k-1] + \Delta u_{Ox}[k] \\ u_{Py}[k] &= u_{Py}[k-1] + \Delta u_{Py}[k] & u_{Oy}[k] &= u_{Oy}[k-1] + \Delta u_{Oy}[k] \\ u_{Pz}[k] &= u_{Pz}[k-1] + \Delta u_{Pz}[k] & u_{Oz}[k] &= u_{Oz}[k-1] + \Delta u_{Oz}[k] \end{aligned} \quad (2.18)$$

avec des fonctions de saturation pour $u[k]$ et $\Delta u[k]$.

2.4.2 Commande au voisinage des positions singulières

Le calcul du MCD proprement dit, c'est à dire de la matrice jacobienne a été largement étudié. Nous pouvons trouver une étude approfondie pour le calcul numérique dans [Orin et al., 1983]. Cependant, le MCD du bras robotique à 6 liaisons pivot est bien connu et nous avons choisi de travailler avec le MCD calculé de façon analytique. Pour plus de détails se référer à l'annexe B.2.

Pour la commande, nous utilisons l'inverse de cette matrice. Le plus grave inconvénient de ce type de loi de commande est dû au fait que la matrice jacobienne peut devenir singulière. Les configurations du bras manipulateur qui produisent une matrice jacobienne singulière sont appelées *configurations singulières*. Deux problèmes apparaissent :

- Au niveau d'une configuration singulière, l'inverse de la matrice jacobienne n'existe plus. Ce qui produit des vitesses de consigne articulaires infinies. L'utilisation de l'inverse généralisée résout ce problème.
- Au voisinage d'une configuration singulière, les vitesses de consigne des articulations augmentent. Une fonction de conditionnement est utilisé pour garantir le suivi de la trajectoire.

Inverse généralisée de la matrice jacobienne

Pour le calcul de l'inverse de la matrice jacobienne d'un bras manipulateur de type $6R$, plusieurs alternatives sont présentées en détails dans [Buss, 2004] et une solution analytique est proposée dans [Gorla et al., 1984]. La solution analytique est la moins coûteuse en temps de calcul, cependant elle ne résout pas le problème de la commande pour les positions singulières. Certaines méthodes utilisent le calcul des valeurs singulières ou le déterminant de la matrice jacobienne pour déterminer le voisinage d'une position singulière et utiliser des stratégies complémentaires, ce qui augmente le temps de calcul.

Nakamura propose une méthode robuste pour le calcul de la matrice inverse généralisée au voisinage des singularités. La matrice est définie par :

$$\mathbf{J}^* = (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})^{-1} \mathbf{J}^T \quad (2.19)$$

où λ est utilisé comme paramètre d'amortissement de l'inverse de la matrice jacobienne. En choisissant $\lambda > 0$, nous assurons l'existence de l'inverse de la matrice jacobienne. Ceci permet de traverser les positions singulières. Une valeur de $\lambda = 0.001$ donne des résultats satisfaisants [Nakamura, 1991].

Fonction de conditionnement des vitesses articulaires

Des vitesses articulaires assez importantes sont observées au voisinage d'une position singulière. En utilisant, la matrice inverse généralisée, le problème est partiellement résolu, car nous assurons l'existence de la matrice inverse mais les valeurs de vitesses articulaires peuvent dépasser les limites physiques.

Une première approche pour résoudre ce problème est de limiter les vitesses de consigne des servomoteurs. Ce type de solution n'est pas satisfaisante, car une fois qu'une vitesse est saturée le suivi d'une trajectoire n'est plus possible. Nous proposons une limitation de la vitesse proportionnelle à la plus grande vitesse, en utilisant une fonction de pondération.

Soit $\dot{\boldsymbol{\theta}}_L$ le vecteur des vitesses articulaires maximales. Nous déterminons le facteur de vitesse f_v par :

$$f_v = \max \left(\left| \frac{\dot{\theta}_1}{\dot{\theta}_{L1}} \right|, \dots, \left| \frac{\dot{\theta}_6}{\dot{\theta}_{L6}} \right| \right) \quad (2.20)$$

Nous définissons la fonction de conditionnement $cmd(f_v)$ par :

$$cmd(f_v) = \begin{cases} 1 & \text{si } f_v \leq 1 \\ \frac{1}{f_v} & \text{si } f_v > 1 \end{cases} \quad (2.21)$$

Les vitesses articulaires ($\dot{\boldsymbol{\theta}}^*$) qui sont utilisées comme consignes pour les boucles de commande de chaque articulation sont données par :

$$\dot{\boldsymbol{\theta}}^* = cmd(f_v) \dot{\boldsymbol{\theta}} \quad (2.22)$$

2.4.3 Résultats de simulation

Nous souhaitons déplacer l'organe terminal du bras manipulateur d'une situation origine \mathcal{S}_0

$$\mathcal{S}_0 = \begin{bmatrix} 0.840 \\ 0.000 \\ 0.860 \\ 0.000 \\ 0.380 \\ 0.000 \\ 0.923 \end{bmatrix}$$

à une situation finale \mathcal{S}_F

$$\mathcal{S}_F = \begin{bmatrix} 0.510 \\ -0.046 \\ 1.180 \\ 0.146 \\ 0.010 \\ 0.880 \\ 0.985 \end{bmatrix}$$

Le but est de montrer le suivi d'une trajectoire quelconque dans toutes les directions de la situation. Pour le calcul des trajectoires de consigne se référer au chapitre 3. La figure 2.8 montre l'évolution de l'orientation, en présentant le quaternion de consigne et le quaternion mesuré.

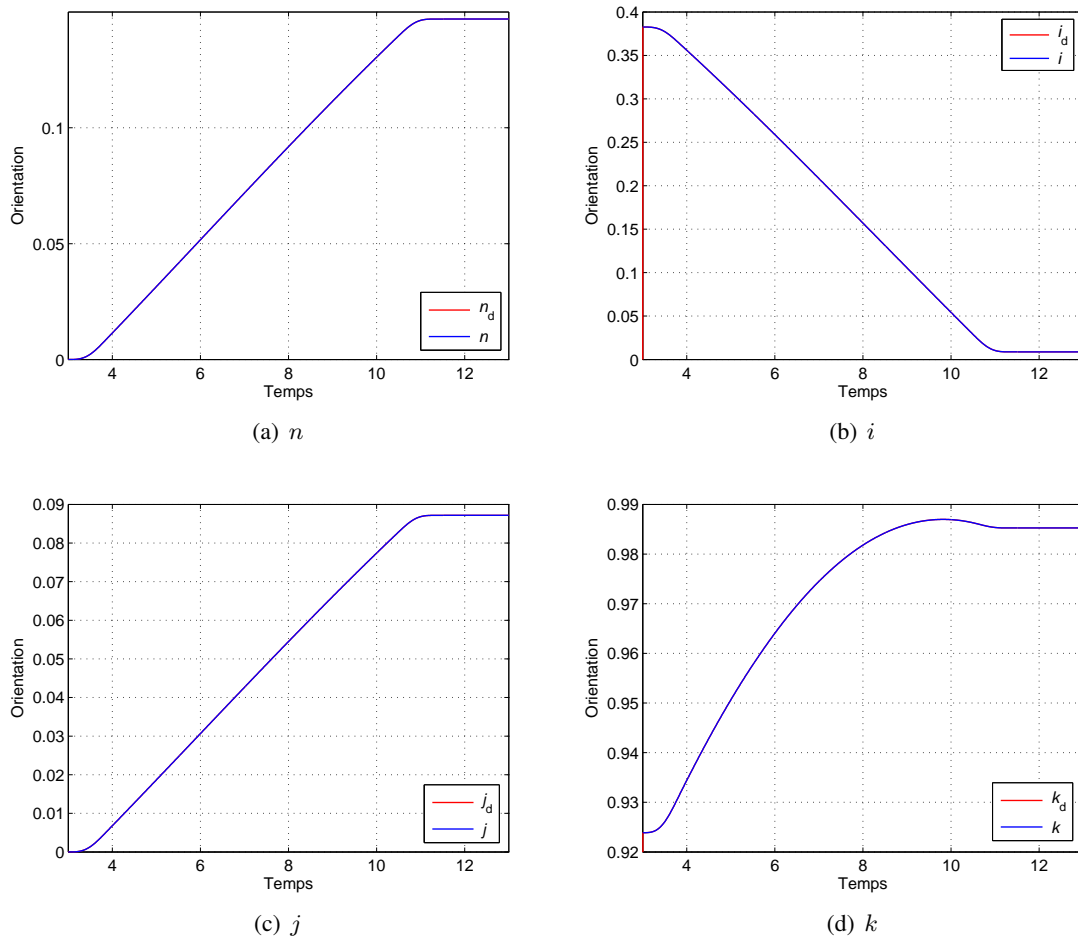


FIG. 2.8 – Évolution de l'orientation

La figure 2.9 montre l'évolution de x , y et z avec leur intégrale de l'erreur quadratique.

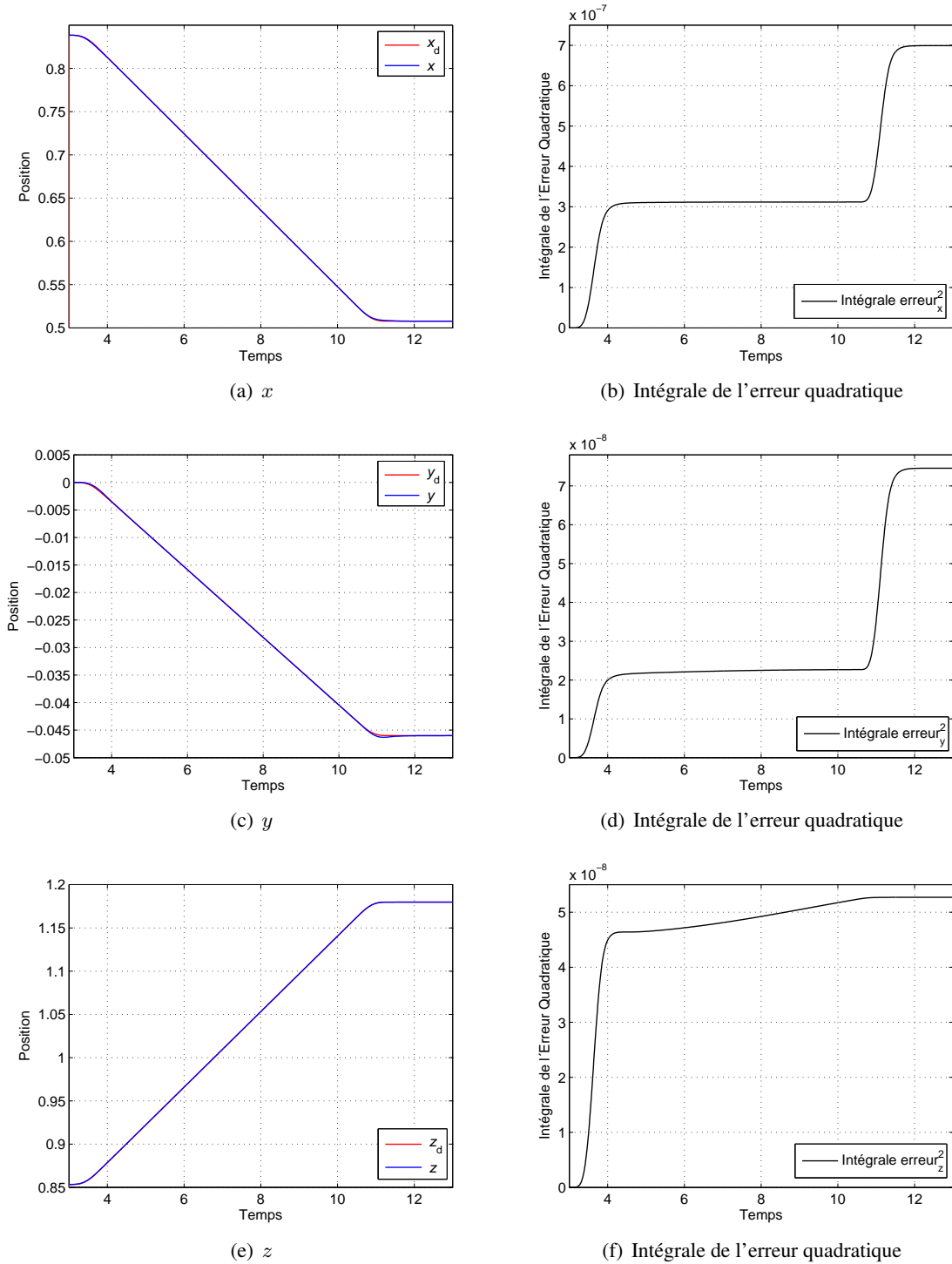


FIG. 2.9 – Évolution de la position et de l'intégrale de l'erreur quadratique en x , y et z

2.5 Conclusions

Nous avons présenté dans ce chapitre les schémas de commande pour un bras manipulateur de type $6R$ pour des applications en robotique de service. Les contraintes propres de la robotique de service permettent des simplifications sur les boucles de commande pour des robots équipés de servomoteurs avec forts rapports de réduction, les lois de commande restent « strictement cinématiques ».

Les schémas de commande pour les articulations présentés sous la forme d'exemples ont été validés sur la plate-forme expérimentale *H2-Bis* du *LAAS-CNRS*. La transition entre l'asservissement en position et l'asservissement en vitesse a été réalisée sans sauts et sans « ré-initialisation » de la partie intégrale de la loi de commande.

La fonction de saturation utilisée à différentes étapes permet de limiter les contraintes cinématiques d'accélération et de vitesse d'une façon naturelle. Cette stratégie développée ici est utilisée dans la suite du travail, par exemple pour limiter les contraintes cinématiques pour l'asservissement dans l'espace cartésien.

C'est dans l'étape de validation des différents schémas de commande que nous avons constaté l'importance de la génération de trajectoires en vitesse et en position pour calculer les consignes. Si nous observons le schéma d'asservissement dans l'espace cartésien, il constitue un cas particulier du schéma multidimensionnel avec signal d'anticipation où l'inverse de la matrice jacobienne est utilisée comme fonction de transformation. Cette information nous a permis d'améliorer la performance en ajoutant une partie intégrale à la loi de commande.

Nous obtenons un haut niveau de performance de la boucle d'asservissement dans l'espace cartésien, même en présence de configurations singulières. L'utilisation de l'inverse généralisée associée aux fonctions de pondération ont permis de traverser les configurations singulières sans problèmes.

Finalement, après toutes ces expériences, nous obtenons un schéma de commande qui satisfait les contraintes de la robotique de service. Cependant, notre solution est strictement cinématique dans l'espace libre. Pour des applications dans l'espace contraint, applications avec des contacts ou d'interaction en effort, nous considérons que la commande des articulations devra être réalisé en *couple*. Ce type de commande n'est pas accessible sur les robots dont nous disposons.

Chapitre 3

Génération des mouvements souples

3.1 Introduction

L'approche en quatre étapes que nous avons proposée au premier chapitre (figure 1.1) nécessite une représentation des chemins pour passer de l'étape de planification à celle de génération du mouvement. Les planificateurs actuels fournissent généralement ces chemins que le robot doit suivre sous la forme d'une liste de points de passage. A ce niveau, les caractéristiques cinématiques et dynamiques du mouvement n'ont pas été spécifiées, il revient donc à l'étape de génération des mouvements de calculer ces trajectoires.

Au chapitre 2, nous avons vu que les consignes peuvent être appliquées directement à chacune des articulations sous la forme de consignes de vitesse $\dot{\theta}_d$ et de position θ_d . De même, les vitesses opérationnelles \mathbf{V}_d et ω_d associées aux consignes de position \mathbf{P}_d et d'orientation \mathbf{Q}_d constituent les consignes d'une commande à résolution de vitesse lorsque nous utilisons une boucle de retour. Ces consignes sont définies par la trajectoire calculée à l'étape de génération de mouvement.

Wisama Khalil dans [Khalil, 2002] propose une description de différentes approches du problème de la génération de mouvements. L'utilisation des équations *polynomiales* ou des *splines* y est traitée en détail. L'idée de base de l'approche que nous présentons ici a été introduite par Castain et Paul dans [Castain et al., 1984]. Ils ont utilisé six segments de cubiques et une borne pour l'accélération, Liu dans [Liu, 2002], par contre, introduit la méthode de base avec sept segments cubiques pour la génération des mouvements d'un servomoteur dans un seul sens en utilisant des contraintes sur le jerk, l'accélération et la vitesse. Macfarlane et Croft dans [Macfarlane et al., 2003] proposent une méthode plus générale utilisant 7 segments quintiques.

La méthode de base que nous présentons définit une spline de sept segments cubiques en dimension un, elle s'étend facilement à plusieurs dimensions et à différentes conditions de mouvement. Les mouvements obtenus sont optimaux en dimension un. Toutefois, son principal intérêt provient de la simplicité qu'elle induit pour donner un sens physique aux coefficients des polynômes. Ces coefficients décrivent chacun des segments de la trajectoire et sont reliés aux propriétés de l'organe terminal du bras manipulateur pour un mouvement *en ligne* donné.

Pour les applications en robotique de service, les mouvements réalisés par un bras manipulateur doivent respecter des contraintes assez différentes des contraintes qui s'imposent aux robots évoluant en milieu industriel. Hogan [Hogan, 1984] a montré qu'en minimisant le jerk, la trajectoire obtenue satisfait les contraintes de continuité. Il est important de noter que les travaux de Hogan ont été réalisés pour modéliser les mouvements des bras humains. Ces résultats nous ont incité à poursuivre nos travaux dans cette voie, en minimisant le jerk nous obtenons une trajectoire lisse. En limitant aussi l'accélération et la vitesse nous obtenons une *souplesse* dans le mouvement. Pour les applications que nous envisageons, ces limitations sont nécessaires afin d'assurer la sécurité et le confort des utilisateurs humains. Par exemple, l'énergie cinétique que pourrait transmettre le robot à un humain lors d'une collision augmente avec le carré de la vitesse.

Nous définissons les *mouvements souples* comme les mouvements ayant un jerk, une accélération

et une vitesse limités en translation et en rotation. Ces conditions interviennent aussi au démarrage et à l'arrêt du mouvement afin de réaliser des mouvements avec un démarrage souple et un arrêt souple. L'intérêt des mouvements souples apparaît aussi pour assurer le déplacement d'un verre contenant un liquide.

Pourquoi s'arrêter à la limitation du jerk et ne pas continuer à limiter la première dérivée du jerk ou les dérivées suivantes ? Les travaux de [Lambrechts et al., 2004] proposent l'utilisation de la dérivée du jerk et nous verrons que la stratégie utilisée dans ce chapitre peut être étendue à de nouvelles contraintes, malheureusement avec un coût prohibitif en terme de complexité des calculs. À la différence des travaux de Lambrechts, notre approche peut s'appliquer à une articulation ou à un bras manipulateur entier.

Nous avons privilégié, tout au long de ce chapitre, des solutions analytiques aux problèmes posés par rapport à des processus d'optimisation plus gourmand en calcul. En effet, notre objectif est de privilégier l'application de cette méthode à la génération du mouvement en ligne.

Ce chapitre est organisé de la façon suivante. Tout d'abord nous définissons les concepts de *chemin* et de *trajectoire*, et décrivons la méthode de base pour les mouvements souples. Pour respecter la numération la quatrième section correspond à la résolution du système d'équations et son application à la génération de mouvement point à point. La cinquième section se focalise sur le mouvement en temps imposé. La sixième section aborde le problème de mouvement à conditions initiales et finales quelconques. Ensuite, à la septième section, nous abordons le cas multidimensionnel. La huitième section recense des particularités relatives à différentes applications ou extensions de ces méthodes. Et pour finir, nous donnons quelques conclusions.

3.2 Chemins et Trajectoires

Soit $X \in \mathbb{R}^n$ la position d'un point dans l'espace à n dimensions.

Un *chemin* Φ de \mathbb{R}^n , $\Phi \subset \mathbb{R}^n$, est un ensemble de positions successives de \mathbb{R}^n . Il relie une position initiale X_0 à une position finale X_f .

Certains chemins peuvent être représentés sous la forme d'une liste de $m + 1$ positions successives, telles que $X_i \in \Phi$ avec $0 \leq i \leq m$. Lorsque le chemin est défini par les seules positions initiale X_0 et finale X_f on parle de mouvement point à point.

Définissons la *trajectoire* TR d'un point comme l'ensemble des positions successives de ce point quand le temps t varie de T_0 à T_f pour un déplacement de X_0 à X_f :

$$\begin{aligned} \text{TR} : [T_0, T_f] &\longrightarrow \Phi \subset \mathbb{R}^n \\ t &\longmapsto \text{TR}(t) \end{aligned} \quad (3.1)$$

Un chemin Φ est un ensemble de positions de l'espace sans relation temporelle. Une trajectoire TR est une fonction du temps qui définit une loi d'évolution sur un chemin. La trajectoire définit les caractéristiques cinématiques du mouvement.

3.3 Méthode de base pour les mouvements souples

La méthode de base définit la loi d'évolution pour un mouvement point à point à une dimension. Définissons les fonctions du temps $J(t)$ pour le jerk, $A(t)$ pour l'accélération, $V(t)$ pour la vitesse et $X(t)$ pour la position. De la mécanique classique, nous savons que la vitesse correspond à la première dérivée de la position, l'accélération à la première dérivée de la vitesse et le jerk à la première dérivée de l'accélération :

$$V(t) = \frac{dX(t)}{dt} \quad A(t) = \frac{dV(t)}{dt} \quad J(t) = \frac{dA(t)}{dt} \quad (3.2)$$

Les conditions initiales du mouvement sont définies par A_0 pour l'accélération, V_0 pour la vitesse et X_0 pour la position. Nous pouvons alors écrire :

$$\begin{aligned} A(t) &= A_0 + \int_0^t J(u)du \\ V(t) &= V_0 + \int_0^t A(u)du \\ X(t) &= X_0 + \int_0^t V(u)du \end{aligned} \quad (3.3)$$

Les contraintes cinématiques sont définies par J_{max} pour le jerk maximal ($|J(t)| \leq J_{max}$), A_{max} pour l'accélération maximale et V_{max} pour la vitesse maximale.

propriété 1 *Un mouvement soumis à des contraintes cinématiques a une seule contrainte saturée sur un intervalle.*

Soit une fonction $f(t)$ saturée, alors $f(t) = f_{Max} = Cte$. Sa dérivé est :

$$\frac{df(t)}{dt} = 0 \quad \text{et donc} \quad \frac{d^2f(t)}{dt^2} = \frac{d^3f(t)}{dt^3} = 0$$

Donc, si la vitesse $V(t)$ est saturée, l'accélération $A(t)$ et le jerk $J(t)$ ne peuvent pas être saturés. De même, si $A(t)$ est saturé, $J(t)$ et $V(t)$ ne peuvent pas être saturés.

Nous obtenons trois types de mouvements que nous appellerons *élémentaires* ayant chacun une contrainte saturée sur un intervalle $]T_0, T_f[$:

1. jerk saturé ;
2. accélération saturée : $J(t) = 0$;
3. vitesse saturée : $J(t) = 0$ et $A(t) = 0$;

La durée de l'intervalle $]T_0, T_f[$ est $\mathcal{T} = T_f - T_0$.

3.3.1 Mouvement élémentaire à jerk maximal

Avec le jerk fixé à J_{max} , les équations du mouvement sont :

$$\begin{aligned} J(t) &= J_{max} \\ A(t) &= A_0 + J_{max}(t - T_0) \\ V(t) &= V_0 + A_0(t - T_0) + \frac{1}{2}J_{max}(t - T_0)^2 \\ X(t) &= X_0 + V_0(t - T_0) + \frac{1}{2}A_0(t - T_0)^2 + \frac{1}{6}J_{max}(t - T_0)^3 \end{aligned} \quad (3.4)$$

Au cours de ce type de mouvement, l'accélération varie linéairement par rapport au temps, alors que la vitesse et la position varient de façon polynomiale. Les conditions finales en T_f pour le mouvement sont données par :

$$\begin{aligned} A(T_f) &= A_0 + J_{max}\mathcal{T} \\ V(T_f) &= V_0 + A_0\mathcal{T} + \frac{1}{2}J_{max}\mathcal{T}^2 \\ X(T_f) &= X_0 + V_0\mathcal{T} + \frac{1}{2}A_0\mathcal{T}^2 + \frac{1}{6}J_{max}\mathcal{T}^3 \end{aligned} \quad (3.5)$$

3.3.2 Mouvement élémentaire à accélération maximale

Les équations du mouvement à accélération donnée A_{max} sont :

$$\begin{aligned} J(t) &= 0 \\ A(t) &= A_{max} \\ V(t) &= V_0 + A_{max}(t - T_0) \\ X(t) &= X_0 + V_0(t - T_0) + \frac{1}{2}A_{max}(t - T_0)^2 \end{aligned} \quad (3.6)$$

Durant ce type de mouvement, la vitesse varie linéairement par rapport au temps et la position de façon polynomiale. Les conditions finales en T_f pour le mouvement sont données par :

$$\begin{aligned} A(T_f) &= A_{max} \\ V(T_f) &= V_0 + A_{max}T \\ X(T_f) &= X_0 + V_0T + \frac{1}{2}A_{max}T^2 \end{aligned} \quad (3.7)$$

3.3.3 Mouvement élémentaire à vitesse maximale

Les équations d'un mouvement ayant une vitesse fixée à V_{max} sont :

$$\begin{aligned} J(t) &= 0 \\ A(t) &= 0 \\ V(t) &= V_{max} \\ X(t) &= X_0 + V_{max}(t - T_0) \end{aligned} \quad (3.8)$$

Pendant ce type de mouvement, la position varie linéairement avec le temps. Les conditions finales en T_f pour le mouvement sont données par :

$$\begin{aligned} A(T_f) &= 0 \\ V(T_f) &= V_{max} \\ X(T_f) &= X_0 + V_{max}T \end{aligned} \quad (3.9)$$

Au niveau de la régularité, chacun des mouvements élémentaires est de classe C^∞ à l'intérieur de l'intervalle $]T_0, T_f[$ car ce sont des fonctions polynomiales.

3.3.4 Mouvement souple à sept segments

La figure 3.1 montre une série de sept segments, chacun lié à un mouvement élémentaire.

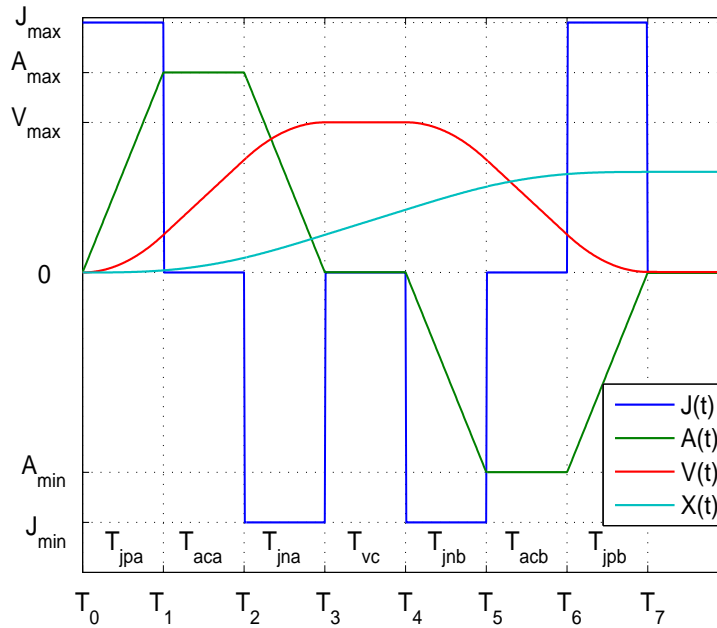


FIG. 3.1 – Définitions des sept segments pour la méthode de base avec des conditions cinématiques initiales et finales nulles

Nous souhaitons que $X(t)$ soit de classe \mathcal{C}^2 sur $[T_0, T_f]$, et donc $V(t)$ de classe \mathcal{C}^1 et $A(t)$ de classe \mathcal{C}^0 sur le même intervalle. Ceci afin de garantir un lissage minimum de la courbe de position aux instants de commutation. En chacun de ces points de commutation, ceci se traduit par des contraintes égalités entre les limites à gauche et à droite des positions, vitesses et accélérations.

La méthode de base considère des conditions cinématiques initiales et finales nulles. Ces conditions peuvent s'écrire : $A_0 = 0, V_0 = 0, A_f = 0$ et $V_f = 0$.

En utilisant les lettres a et b pour différencier les intervalles de la première moitié des courbes de ceux de la deuxième moitié, la durée de chaque intervalle est donné par :

$\mathcal{T}_{jpa} = T_1 - T_0$	Mouvement à jerk maximal sur $[T_0, T_1]$
$\mathcal{T}_{aca} = T_2 - T_1$	Mouvement à accélération maximale sur $[T_1, T_2]$
$\mathcal{T}_{jna} = T_3 - T_2$	Mouvement à jerk minimal sur $[T_2, T_3]$
$\mathcal{T}_{vc} = T_4 - T_3$	Mouvement à vitesse maximale sur $[T_3, T_4]$
$\mathcal{T}_{jnb} = T_5 - T_4$	Mouvement à jerk minimal sur $[T_4, T_5]$
$\mathcal{T}_{acb} = T_6 - T_5$	Mouvement à accélération minimale sur $[T_5, T_6]$
$\mathcal{T}_{jpb} = T_7 - T_6$	Mouvement à jerk maximal sur $[T_6, T_7]$

Nous pouvons observer que la courbe de vitesse est symétrique sur l'intervalle total $[T_0, T_7]$ par rapport au milieu de l'intervalle. De même, l'accélération est antisymétrique, elle se compose d'une période d'accélération positive suivi d'une période d'accélération négative ou décélération.

A l'intérieur de l'intervalle $[T_0, T_3]$, la courbe d'accélération est symétrique alors que la courbe de jerk est antisymétrique. Le même comportement est observable dans l'intervalle $[T_4, T_7]$ qui correspond à la période de décélération.

Ces propriétés de symétrie n'apparaissent que si les contraintes du mouvement respectent les conditions suivantes que nous supposons vérifiées dans la suite de ce manuscrit :

$$\begin{aligned} J_{min} &= -J_{max} \\ A_{min} &= -A_{max} \\ V_{min} &= -V_{max} \end{aligned} \quad (3.10)$$

Optimalité

Lorsque les sept mouvements élémentaires définissant la méthode de base sont réalisés en saturant une contrainte, soit $J(t) = J_{max}$, $A(t) = A_{max}$ ou $V(t) = V_{max}$, le mouvement complet réalisé par les sept segments produit un mouvement en temps minimal sous contraintes.

Pour la séquence définie par la méthode de base, le temps minimal \mathcal{T}_M est donné par :

$$\mathcal{T}_M = \mathcal{T}_{jpa} + \mathcal{T}_{aca} + \mathcal{T}_{jna} + \mathcal{T}_{vc} + \mathcal{T}_{jnb} + \mathcal{T}_{acb} + \mathcal{T}_{jpb}$$

3.3.5 Système d'équations du mouvement souple à sept segments

Considérons les sept intervalles suivants : $[T_0, T_1]$, $[T_1, T_2]$, $[T_2, T_3]$, $[T_3, T_4]$, $[T_4, T_5]$, $[T_5, T_6]$ et $[T_6, T_7]$

Les conditions initiale et finale s'expriment :

$$\begin{aligned} A(T_0) &= 0 & A(T_f) &= 0 \\ V(T_0) &= 0 & V(T_f) &= 0 \\ X(T_0) &= X_0 & X(T_f) &= X_f \end{aligned} \quad (3.11)$$

A partir des conditions de continuité \mathcal{C}^2 , nous pouvons construire les fonctions suivantes :

$$\begin{aligned}
 t \in [T_0, T_1] \quad & \begin{aligned} A(t) &= J_{max}(t-T_0) \\ V(t) &= \frac{1}{2}J_{max}(t-T_0)^2 \\ X(t) &= X_0 + \frac{1}{6}J_{max}(t-T_0)^3 \end{aligned} \\
 t \in [T_1, T_2] \quad & \begin{aligned} A(t) &= A_{max} \\ V(t) &= V(T_1) + A_{max}(t-T_1) \\ X(t) &= X(T_1) + V(T_1)(t-T_1) + \frac{1}{2}A_{max}(t-T_1)^2 \end{aligned} \\
 t \in [T_2, T_3] \quad & \begin{aligned} A(t) &= A(T_2) - J_{max}(t-T_2) \\ V(t) &= V(T_2) + A(T_2)(t-T_2) - \frac{1}{2}J_{max}(t-T_2)^2 \\ X(t) &= X(T_2) + V(T_2)(t-T_2) + \frac{1}{2}A(T_2)(t-T_2)^2 - \frac{1}{6}J_{max}(t-T_2)^3 \end{aligned} \\
 t \in [T_3, T_4] \quad & \begin{aligned} A(t) &= 0 \\ V(t) &= V_{max} \\ X(t) &= X(T_3) + V_{max}(t-T_3) \end{aligned} \\
 t \in [T_4, T_5] \quad & \begin{aligned} A(t) &= A(T_4) - J_{max}(t-T_4) \\ V(t) &= V(T_4) + A(T_4)(t-T_4) - \frac{1}{2}J_{max}(t-T_4)^2 \\ X(t) &= X(T_4) + V(T_4)(t-T_4) + \frac{1}{2}A(T_4)(t-T_4)^2 - \frac{1}{6}J_{max}(t-T_4)^3 \end{aligned} \\
 t \in [T_5, T_6] \quad & \begin{aligned} A(t) &= -A_{max} \\ V(t) &= V(T_5) - A_{max}(t-T_5) \\ X(t) &= X(T_5) + V(T_5)(t-T_5) - \frac{1}{2}A_{max}(t-T_5)^2 \end{aligned} \\
 t \in [T_6, T_7] \quad & \begin{aligned} A(t) &= A(T_6) + J_{max}(t-T_6) \\ V(t) &= V(T_6) + A(T_6)(t-T_6) + \frac{1}{2}J_{max}(t-T_6)^2 \\ X(t) &= X(T_6) + V(T_6)(t-T_6) + \frac{1}{2}A(T_6)(t-T_6)^2 + \frac{1}{6}J_{max}(t-T_6)^3 \end{aligned}
 \end{aligned} \tag{3.12}$$

L'expression de ces 21 fonctions fait apparaître 21 inconnus : les sept instants de commutation ($T_1, T_2, T_3, T_4, T_5, T_6$ et T_7) et les constantes suivantes : $V(T_1), X(T_1), A(T_2), V(T_2), X(T_2), X(T_3), A(T_4), V(T_4), X(T_4), V(T_5), X(T_5), A(T_6), V(T_6)$ et $X(T_6)$.

Les contraintes de continuité en position, vitesse et accélération aux instants de commutations fournissent $6 \times 3 = 18$ équations auxquelles s'ajoutent les position, vitesse et accélération finales ; soit aussi 21 équations.

Ces équations ne sont pas linéaires et nécessitent donc quelques précautions pour la résolution. Les hypothèses de symétrie et de conditions initiales et finales nulles en accélération et vitesse simplifient la résolution.

3.4 Résolution du système d'équation

Écrivons le système d'équations à résoudre :

$$\begin{aligned}
& A_{max} = J_{max} \mathcal{T}_{jpa} \\
T_1 \quad & V(T_1) = \frac{1}{2} J_{max} \mathcal{T}_{jpa}^2 \\
& X(T_1) = X_0 + \frac{1}{6} J_{max} \mathcal{T}_{jpa}^3 \\
& A(T_2) = A_{max} \\
T_2 \quad & V(T_2) = V(T_1) + A_{max} \mathcal{T}_{aca} \\
& X(T_2) = X(T_1) + V(T_1) \mathcal{T}_{aca} + \frac{1}{2} A_{max} \mathcal{T}_{aca}^2 \\
& 0 = A_{max} - J_{max} \mathcal{T}_{jna} \\
T_3 \quad & V_{max} = V(T_2) + A_{max} \mathcal{T}_{jna} - \frac{1}{2} J_{max} \mathcal{T}_{jna}^2 \\
& X(T_3) = X(T_2) + V(T_2) \mathcal{T}_{jna} + \frac{1}{2} A_{max} \mathcal{T}_{jna}^2 - \frac{1}{6} J_{max} \mathcal{T}_{jna}^3 \\
& A(T_4) = 0 \\
T_4 \quad & V(T_4) = V_{max} \\
& X(T_4) = X(T_3) + V_{max} \mathcal{T}_{vc} \\
& -A_{max} = -J_{max} \mathcal{T}_{jnb} \\
T_5 \quad & V(T_5) = V_{max} - \frac{1}{2} J_{max} \mathcal{T}_{jnb}^2 \\
& X(T_5) = X(T_4) + V_{max} \mathcal{T}_{jnb} - \frac{1}{6} J_{max} \mathcal{T}_{jnb}^3 \\
& A(T_6) = -A_{max} \\
T_6 \quad & V(T_6) = V(T_5) - A_{max} \mathcal{T}_{acb} \\
& X(T_6) = X(T_5) + V(T_5) \mathcal{T}_{acb} - \frac{1}{2} A_{max} \mathcal{T}_{acb}^2 \\
& 0 = -A_{max} + J_{max} \mathcal{T}_{jpb} \\
T_7 \quad & 0 = V(T_6) - A_{max} \mathcal{T}_{jpb} + \frac{1}{2} J_{max} \mathcal{T}_{jpb}^2 \\
& X_f = X(T_6) + V(T_6) \mathcal{T}_{jpb} - \frac{1}{2} A_{max} \mathcal{T}_{jpb}^2 + \frac{1}{6} J_{max} \mathcal{T}_{jpb}^3
\end{aligned} \tag{3.13}$$

A partir de ces équations, nous pouvons directement calculer la durée \mathcal{T}_{jmax} des segments effectués à jerk maximal ou minimal :

$$\mathcal{T}_{jmax} = \mathcal{T}_{jpa} = \mathcal{T}_{jna} = \mathcal{T}_{jnb} = \mathcal{T}_{jpb} = \frac{A_{max}}{J_{max}} \tag{3.14}$$

L'égalité de ces quatre durées est en accord avec les propriétés de symétrie.

En substituant \mathcal{T}_{jmax} dans les équations de $V(T_1)$ et $X(T_1)$, nous déterminons la vitesse atteinte et le déplacement effectué à l'instant T_1 .

$$\begin{aligned}
V(T_1) = V_{jpa} &= \frac{A_{max}^2}{2J_{max}} \\
X(T_1) = X_{jpa} &= X_0 + \frac{A_{max}^3}{6J_{max}^2}
\end{aligned} \tag{3.15}$$

En reportant ces résultats dans les équations relatives à l'instant T_2 nous obtenons :

$$\begin{aligned}
A(T_2) &= A_{max} \\
V(T_2) &= \frac{A_{max}^2}{2J_{max}} + A_{max} \mathcal{T}_{aca} \\
X(T_2) &= X_0 + \frac{A_{max}^3}{6J_{max}^2} + \frac{A_{max}^2}{2J_{max}} \mathcal{T}_{aca} + \frac{A_{max}}{2} \mathcal{T}_{aca}^2
\end{aligned} \tag{3.16}$$

Par substitution dans les équations relatives à l'instant T_3 on aboutit à :

$$\begin{aligned} V_{max} &= \frac{A_{max}^2}{J_{max}} + A_{max} \mathcal{T}_{aca} \\ X(T_3) &= X_0 + \frac{A_{max}^3}{J_{max}^2} + \frac{3A_{max}^2}{2J_{max}} \mathcal{T}_{aca} + \frac{1}{2} A_{max} \mathcal{T}_{aca}^2 \end{aligned} \quad (3.17)$$

En considérant les conditions de symétrie, le temps à accélération maximale est $\mathcal{T}_{aca} = \mathcal{T}_{acb}$ que nous appellerons T_{amax} . A partir de l'équation 3.17 en V_{max} nous obtenons ce temps d'accélération :

$$T_{amax} = \frac{V_{max}}{A_{max}} - \frac{A_{max}}{J_{max}} \quad (3.18)$$

Comme $T_{amax} \geq 0$ car c'est une durée, nous obtenons la condition :

$$\frac{V_{max}}{A_{max}} \geq \frac{A_{max}}{J_{max}} \quad (3.19)$$

Dans le cas contraire, la vitesse atteint sa limite V_{max} avant que l'accélération atteigne sa limite A_{max} .

En substituant T_{amax} dans l'équation 3.16 nous obtenons la vitesse limite après une période d'accélération maximale $V_{aca} = V(T_2)$, sans dépasser les contraintes de vitesse maximale.

$$V_{aca} = V(T_2) = V_{max} - \frac{A_{max}^2}{2J_{max}} \quad (3.20)$$

Comme \mathcal{T}_{jmax} est connu on peut calculer la vitesse et la position atteinte en T_3 :

$$V(T_3) = \frac{1}{2} \frac{A_{max}^2}{J_{max}} + A_{max} \left(\sqrt{\frac{A_{max}^2}{4J_{max}^2} + \frac{D}{A_{max}}} - \frac{3A_{max}}{2J_{max}} \right) + A_{max} \left(\frac{A_{max}}{J_{max}} \right) - \frac{1}{2} \left(\frac{A_{max}^2}{J_{max}} \right) \quad (3.21)$$

A partir des propriétés de symétrie nous obtenons l'expression de la position finale X_f où seul le temps à vitesse maximale \mathcal{T}_{vc} est inconnu :

$$X_f = X_0 + \frac{V_{max} A_{max}}{J_{max}} + \frac{V_{max}^2}{A_{max}} + V_{max} \mathcal{T}_{vc} \quad (3.22)$$

En définissant, la distance parcourue D comme

$$D = X_f - X_0 \quad (3.23)$$

Le temps \mathcal{T}_{vc} qui permet d'atteindre la position finale désirée est donné par

$$\mathcal{T}_{vc} = \frac{D - \frac{V_{max} A_{max}}{J_{max}} - \frac{V_{max}^2}{A_{max}}}{V_{max}} \quad (3.24)$$

\mathcal{T}_{vc} est une durée, elle doit donc être positive : $\mathcal{T}_{vc} \geq 0$. Comme J_{max} , A_{max} et V_{max} sont positif, le numérateur de l'équation 3.24 doit donc satisfaire la condition suivante :

$$D \geq \frac{V_{max} A_{max}}{J_{max}} + \frac{V_{max}^2}{A_{max}}$$

Notons D_{S1} la distance limite :

$$D_{S1} = \frac{V_{max} A_{max}}{J_{max}} + \frac{V_{max}^2}{A_{max}} \quad (3.25)$$

3.4.1 Premier seuil D_{S1}

Le seuil D_{S1} constitue une limite qui sépare les mouvements qui atteignent la vitesse maximale V_{max} et comportent sept segments de ceux qui ne l'atteignent pas et comportent seulement 5 segments.

Si la distance à parcourir $D > D_{S1}$, le mouvement atteint la vitesse maximale V_{max} , nous pouvons calculer sa durée T_{vc} en utilisant l'équation 3.24. Si $D = D_{S1}$, le mouvement atteint V_{max} mais $T_{vc} = 0$.

3.4.2 Mouvement point à point à cinq segments

Nous considérons ici les mouvements de distance $D < D_{S1}$ qui n'atteignent pas la vitesse maximale. Nous pouvons alors considérer les segments T_{jna} et T_{jnb} comme un seul segment de durée T_{jn} .

Les cinq segments sont : T_{jpa} , T_{aca} , T_{jn} , T_{acb} et T_{jpb} . Considérons le système d'équations 3.13, avec $T_{jpa} = T_{jna} = T_{jnb} = T_{jpb} = T_{jmax}$, $T_{aca} = T_{acb} = T_a$ et $T_{vc} = 0$. Nous obtenons l'expression de X_f :

$$X_f = X_0 + \frac{2A_{max}^3}{J_{max}^2} + \frac{3A_{max}^2}{J_{max}} T_a + A_{max} T_a^2 \quad (3.26)$$

Nous obtenons alors la valeur de T_a :

$$T_a = \sqrt{\frac{A_{max}^2}{4J_{max}^2} + \frac{D}{A_{max}}} - \frac{3A_{max}}{2J_{max}} \quad (3.27)$$

3.4.3 Deuxième seuil D_{S2}

La durée T_a doit aussi être positive $T_a \geq 0$, ce qui donne une condition de validité de l'équation 3.27 :

$$D \geq 2 \frac{A_{max}^3}{J_{max}^2}$$

Nous notons D_{S2} ce deuxième seuil de distance :

$$D_{S2} = 2 \frac{A_{max}^3}{J_{max}^2} \quad (3.28)$$

La vitesse maximale V_{S2} atteinte est :

$$V_{S2} = \frac{A_{max}^2}{J_{max}} \quad (3.29)$$

3.4.4 Mouvement point à point à trois segments

Dans ce troisième cas, le mouvement est réalisé sans atteindre ni l'accélération maximale ni la vitesse maximale. Nous avons donc :

$$T_{aca} = 0 \quad T_{vc} = 0 \quad T_{acb} = 0$$

En considérant T_{jna} et T_{jpb} comme un seul segment T_{jn} , les trois segments sont : T_{jpa} , T_{jn} et T_{jpb} . Reprenons le système d'équations 3.13, avec $T_{jpa} = T_{jna} = T_{jnb} = T_{jpb} = T_j$. L'expression de la position finale X_f en T_j devient :

$$X_f = X_0 + 2J_{max} T_j^3 \quad (3.30)$$

Le valeur de T_j qui permet de réaliser le déplacement désirée est donné par :

$$T_j = \sqrt[3]{\frac{D}{2J_{max}}} \quad (3.31)$$

Le déplacement minimal D_{min} qui peut être réalisé par le système est fonction du temps d'échantillonnage du système T_s . Le temps de jerk minimal T_j est $T_j = T_s$ et le déplacement minimal obtenu est :

$$D_{min} = 2J_{max}T_s^3 \quad (3.32)$$

3.4.5 Algorithme pour le mouvement point à point

A partir des équations précédentes, nous pouvons proposer l'algorithme suivant pour le mouvement point à point.

Algorithme 1 : Algorithme pour le mouvement point à point

début

Calcul de la distance $D = X_f - X_0$
 Calcul des seuils D_{S1} et D_{S2}

$$D_{S1} = \frac{V_{max}A_{max}}{J_{max}} + \frac{V_{max}^2}{A_{max}}$$

$$D_{S2} = 2\frac{A_{max}^3}{J_{max}^2}$$

si $D \geq D_{s1}$ alors

$$T_j = \frac{A_{max}}{J_{max}} \quad T_a = \frac{V_{max}}{A_{max}} - \frac{A_{max}}{J_{max}}$$

$$T_{vc} = \frac{D - D_{S1}}{V_{max}}$$

sinon

si $D \geq D_{s2}$ alors

$$T_{vc} = 0 \quad T_j = \frac{A_{max}}{J_{max}}$$

$$T_a = \sqrt{\frac{A_{max}^2}{4J_{max}^2} + \frac{D}{A_{max}}} - \frac{3A_{max}}{2J_{max}}$$

sinon

$$T_{vc} = 0 \quad T_a = 0$$

$$T_j = \sqrt[3]{\frac{D}{2J_{max}}}$$

fin

fin

fin

Trois, cinq ou sept segments

Nous avons donc établi qu'un mouvement peut être réalisé en trois, cinq ou sept segments pour un mouvement, à conditions cinématiques initiales et finales nulles, et à condition de considérer les segments correspondant à T_{jna} et T_{jnb} comme un segment unique.

Ces trois types de déplacements qui ont leurs propres domaines de validité sont présentés sur la figure 3.2.

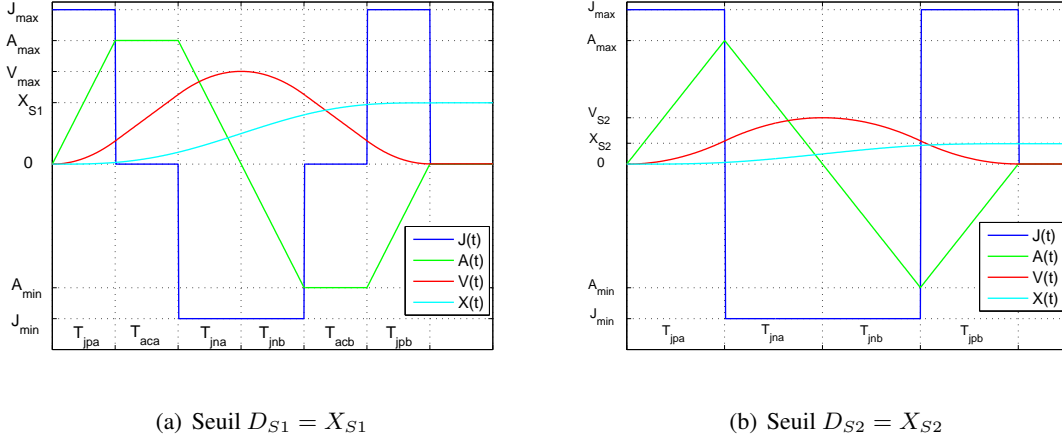


FIG. 3.2 – Seuils de Position pour choisir entre les trois cas

3.5 Mouvements en temps imposé

Nous avons établi un algorithme pour le calcul des déplacements en temps minimal en dimension un. Chacun des segments composant le mouvement point à point correspond à un mouvement élémentaire et le mouvement total obtenu est en temps minimal sous contraintes. Le temps minimal est $\mathcal{T}_f = \mathcal{T}_7 - \mathcal{T}_0$, il est donné par :

$$\mathcal{T}_f = 4\mathcal{T}_j + 2\mathcal{T}_a + \mathcal{T}_{vc} \quad (3.33)$$

Comme nous relâchons la contrainte de temps minimal pour effectuer un déplacement en temps imposé, il existe une infinité de solutions. Remarquons que le temps de parcours ne peut pas être inférieur au temps minimal.

Si on impose les temps \mathcal{T}_j , \mathcal{T}_a et \mathcal{T}_{vc} , il est possible de calculer α , $0 < \alpha < 1$, tel que $J_c = \alpha \times J_{max}$. Il est alors immédiat de calculer les accélérations, vitesses et positions atteintes à chaque instants.

On peut aussi remarquer que si l'on ne modifie pas les instants, on obtient un mouvement homothétique. Choisir $\alpha = 0.5$ conduit à diviser par 2 l'accélération, la vitesse et la distance parcourue.

3.6 Mouvements à conditions initiales et finales quelconques

Dans les paragraphes précédents, nous avons considéré des vitesses et accélérations nulles au point de départ et au point d'arrivée. Nous allons maintenant étudier ce qui se passe dans le cas général.

L'état initial et l'état final sont définies par :

$$\begin{aligned} A(T_0) &= A_0 & A(T_f) &= A_f \\ V(T_0) &= V_0 & V(T_f) &= V_f \\ X(T_0) &= X_0 & X(T_f) &= X_f \end{aligned} \quad (3.34)$$

Nous reprenons la même hypothèse de mouvements optimaux qui impose au système d'évoluer suivant une suite de mouvements élémentaires. Chaque mouvement élémentaire sature soit le jerk, soit l'accélération ou soit la vitesse.

La première question qui se pose concerne le nombre et le type de ces mouvements élémentaires. Nous allons dans les paragraphes suivants étudier ces mouvements élémentaires, puis nous en déduisons les propriétés intéressantes pour le calcul des mouvements.

3.6.1 Mouvements élémentaires

L'état du système est définie par trois grandeurs : la position, la vitesse et l'accélération. Comme il n'est pas aisé de représenter un système à trois dimensions, nous allons le projeter dans le plan vitesse - accélération.

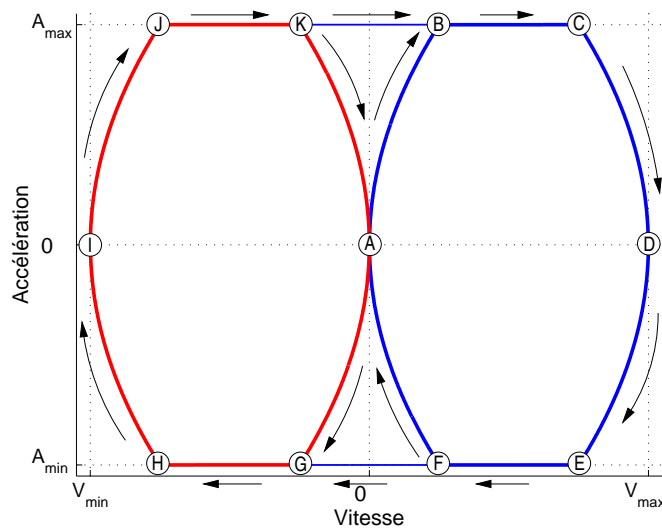


FIG. 3.3 – Espace vitesse - accélération : courbes des mouvements optimaux à sept segments.

Au centre de la figure 3.3, le point A représente les états stationnaires où la vitesse et l'accélération sont nulles et pour lesquels la position reste constante. En haut et en bas se trouvent respectivement les accélérations maximale A_{max} et minimale $-A_{max}$. De même le point gauche extrême, I, correspond à la vitesse minimale $-V_{max}$ et son symétrique de droite, D, à la vitesse maximale V_{max} .

La ligne courbe de droite CDE correspond à une évolution à jerk minimal qui fait évoluer le système de l'accélération maximale à l'accélération minimale. Cette courbe est donc orientée. De même la courbe de gauche HIJ correspond à une évolution à jerk maximal.

L'axe horizontal IAD correspond aux évolutions à accélération nulle, c'est à dire à vitesse constante. Lorsque le système est sur ce segment il peut évoluer indéfiniment pour effectuer un déplacement de longueur quelconque. Les points qui ne sont pas sur ce segment correspondent à des états évolutifs. Par exemple, à partir du point C la seule évolution possible est un mouvement à jerk minimal pour faire diminuer l'accélération et faire évoluer la vitesse vers la vitesse maximale.

3.6.2 Les trois types de mouvements élémentaires optimaux

Nous avons vu un premier état du système qui correspond au point A de la figure 3.3 et pour lequel le système n'évolue pas.

Évolution à vitesse maximale

Le système peut évoluer à la vitesse maximale correspondant au point D de la figure 3.3 pour effectuer un mouvement dans le sens positif de longueur quelconque. Symétriquement, il peut évoluer à la

vitesse minimale correspondant au point I pour effectuer un mouvement dans le sens négatif.

Évolution à accélération maximale

Le système peut évoluer à accélération maximale le long du segment JC ou à accélération minimale le long du segment EH.

Évolution à jerk maximal

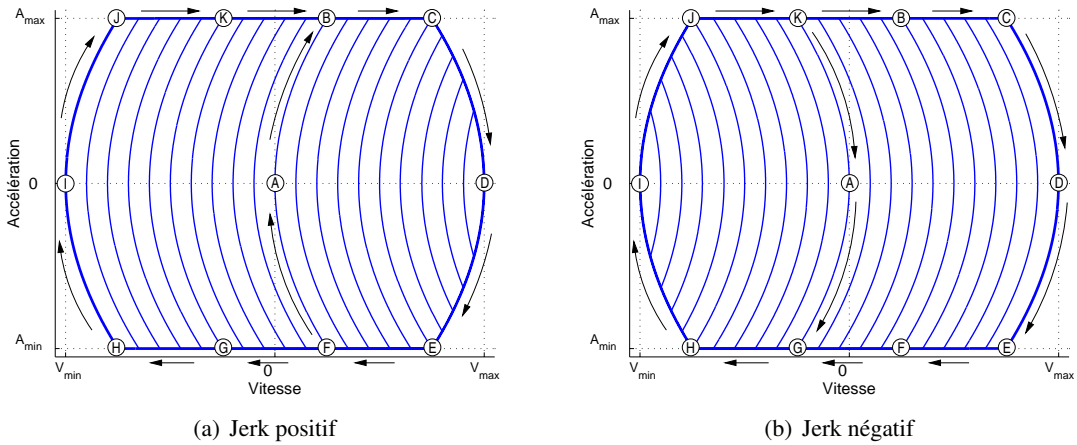


FIG. 3.4 – Espace vitesse - accélération : à gauche courbes à jerk positif et à droite courbes à jerk négatif

La figure 3.4 montre à gauche des évolutions à jerk maximal et à droite des évolutions à jerk minimal.

Ces courbes à jerk maximal ou minimal permettent d'atteindre n'importe quel état du système. Par exemple, la figure 3.6 montre un chemin qui permet d'atteindre l'état P2.

Forme des courbes à jerk maximal

L'expression de la vitesse et de l'accélération, pour un mouvement à jerk maximal, sont :

$$\begin{aligned} A &= A_0 + J_{max}(t) \\ V &= V_0 + A_0 t + \frac{1}{2} J_{max} t^2 \end{aligned} \quad (3.35)$$

Exprimons t à partir de l'accélération dans la première égalité et reportons le dans la seconde :

$$\begin{aligned} t &= (A - A_0) \frac{1}{J_{max}} \\ V &= V_0 + A_0 (A - A_0) \frac{1}{J_{max}} + \frac{1}{2} J_{max} \left((A - A_0) \frac{1}{J_{max}} \right)^2 \\ &= V_0 + \frac{1}{2 J_{max}} (2A_0 A - 2A_0^2 + A^2 - 2A_0 A + A_0^2) \\ &= V_0 + \frac{1}{2 J_{max}} (A^2 - A_0^2) \end{aligned} \quad (3.36)$$

Nous pouvons tracer les courbes à jerk maximal à partir de l'axe horizontal des accélérations nulles, c'est à dire en choisissant $A_0 = 0$. Les courbes à jerk maximal sont donc des paraboles horizontales passant par le point d'accélération 0 et de vitesse V_0 . Elles ont pour équation :

$$V = V_0 + \frac{1}{2 J_{max}} A^2 \quad (3.37)$$

Elles sont décalées horizontalement les unes par rapport aux autres. Les courbes à jerk minimal ($-J_{max}$) sont leurs symétriques par rapport à l'axe des vitesses nulles.

Relation entre J_{max} et A_{max}

La figure 3.5 présente différentes courbes qui correspondent à des valeurs de jerk différentes.

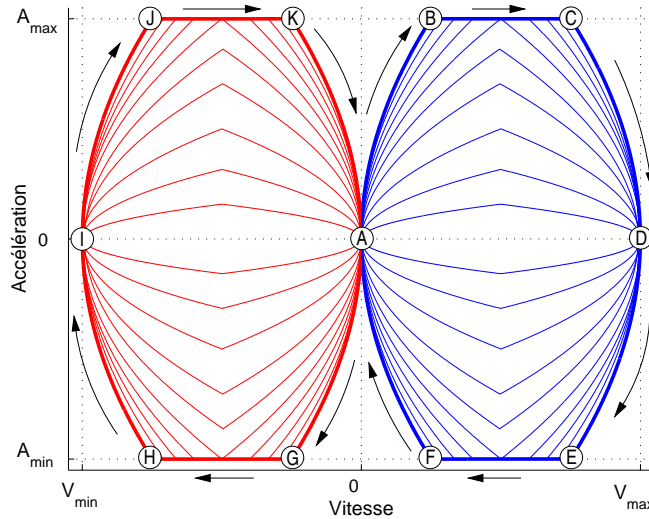


FIG. 3.5 – Espace vitesse - accélération : évolutions pour différentes valeurs de J_{max} .

Pour de faibles valeurs de J_{max} , le système n'atteint pas l'accélération maximale A_{max} et les déplacements optimaux comportent moins de sept segments. Il existe des déplacements à accélération maximale si la condition suivante est vérifiée :

$$\frac{V_{max}}{A_{max}} \geq \frac{A_{max}}{J_{max}} \tag{3.38}$$

Déplacements non optimaux

Il est possible d'évoluer dans toutes les directions du plan accélération-vitesse si l'on ne contraint pas le jerk. Pour ce travail, en dehors des mouvements optimaux, nous utiliserons aussi :

- des déplacements à jerk constant. Chaque valeur de jerk permet de définir une famille de paraboles.
- des déplacements à accélération constante. Ce sont des déplacements suivant des segments horizontaux.
- des déplacements à vitesse constante. Ces déplacements sont représentés par les points de l'axe IAD.

3.6.3 Déplacements optimaux dans le plan accélération-vitesse

Les déplacements de la méthode de base que nous avons étudiés au début du chapitre sont constitués de sept déplacements élémentaires que nous retrouvons sur la figure 3.3 :

1. un mouvement à jerk maximal le long de AB ;
2. un mouvement à accélération maximale le long de BC ;

3. un mouvement à jerk minimal le long de CD ;
4. un mouvement à vitesse maximale en D ;
5. un mouvement à jerk minimal le long de DE ;
6. un mouvement à accélération minimale le long de EF ;
7. un mouvement à jerk maximal le long de FA ;

Un mouvement optimal comporte au plus 7 mouvements élémentaires

Nous allons montrer que les mouvements optimaux comportent au plus sept déplacements élémentaires. Tout d'abord, montrons qu'un mouvement optimal ne peut pas comporter à la fois un déplacement à vitesse maximale et un déplacement à vitesse minimale.

Considérons un mouvement comportant un déplacement élémentaire à vitesse maximale pendant un temps supérieur ou égal à dt et un autre déplacement élémentaire à vitesse minimale pendant un temps supérieur ou égal à dt . Le déplacement effectué à vitesse maximale pendant le temps dt compense le mouvement effectué à vitesse minimale pendant le temps dt . Donc il existe un mouvement plus court amenant le système dans le même état de position, vitesse et accélération : le même auquel on a enlevé le déplacement à vitesse maximale pendant le temps dt et le déplacement à vitesse minimale pendant le temps dt .

Dans le plan vitesse - accélération, un déplacement optimal ne peut pas comporter à la fois les mouvements correspondants aux points I et D. De même il ne peut pas comporter le point A. Les courbes à jerk maximal ne passant pas par I ou D ne peuvent correspondre qu'au premier ou au dernier élément. En respectant ces contraintes, il n'est pas possible de définir de chemin comportant plus de 7 déplacements optimaux élémentaires dans le plan vitesse - accélération.

Types de déplacements optimaux à sept segments

A partir d'un état quelconque de vitesse et d'accélération, deux stratégies sont possibles :

- commencer un mouvement à jerk maximal pour rejoindre le point D. Le point D est atteint par au plus 3 déplacements élémentaires : celui à jerk maximal, suivi éventuellement d'un mouvement à accélération maximale puis d'un mouvement à jerk minimal.
- commencer un mouvement à jerk minimal pour rejoindre le point I. Le point I est atteint par au plus 3 déplacements élémentaires : celui à jerk minimal, suivi éventuellement d'un mouvement à accélération minimale puis d'un mouvement à jerk maximal.

De même, il est possible d'atteindre n'importe quel état par au plus trois déplacements élémentaires à partir du point D ou du point I.

On obtient les deux types de parcours possibles à sept segments :

départ à jerk maximal

1. jerk maximal
2. accélération maximale
3. jerk minimal
4. vitesse maximale
5. jerk minimal
6. accélération minimale
7. jerk maximal

départ à jerk minimal

1. jerk minimal
2. accélération minimal
3. jerk maximal
4. vitesse minimale
5. jerk maximal
6. accélération maximale
7. jerk minimal

Mouvements à moins de sept segments

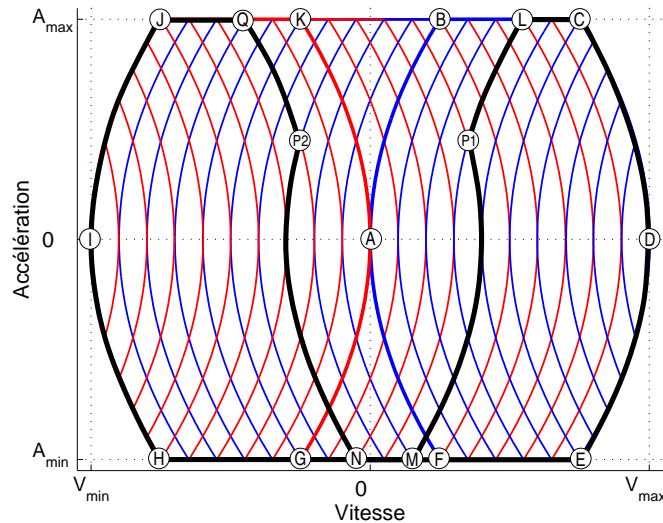


FIG. 3.6 – Espace vitesse - Accélération : évolutions optimale entre deux situations quelconques P1 et P2

Il existe deux types de mouvements à moins de sept segments. Le premier consiste à couper plus tôt que le point I ou D. Par exemple nous avons vu que pour effectuer un petit déplacement positif il n'est pas nécessaire d'atteindre la vitesse maximale. Le deuxième type de mouvement comportant moins de sept segments est le type hybride correspondant à la commutation d'un type de chemin sur l'autre. Par exemple, sur la figure 3.6 pour aller de la situation P1 à la situation P2 les deux mouvements à 7 segments sont :

- P1L LC CD D DE EN NP2
- P1M MH HI I IJ JQ QP2

Il est aussi possible de commencer sur le deuxième chemin par P1M puis de commuter sur le premier par MN suivi de NP2. Le chemin hybride est donc P1M MN NP2.

Influence de la position

Le choix du chemin optimal se fait en fonction de la position, en terme de distance, à atteindre.

Dans toute la première partie où les déplacements que nous avons considérés étaient de distance positive, nous n'avons rencontré que le premier type. Si nous avons effectué des déplacements vers les positions négatives, nous aurions utilisé le deuxième type. Le troisième type n'existe pas dans ce cas, car les deux chemins ne se coupent pas.

Considérons maintenant l'exemple de la figure 3.7, qui est relatif à un déplacement de la position zéro avec une vitesse et une accélération nulle à un point avec une vitesse de $V_{max}/2$ et une accélération donnée correspondant au point P1 de la figure 3.7.

Considérons un très grand déplacement positif, le chemin suivi sera AB BC CD D DE EM MP1. La plus grande partie du déplacement s'effectue sur le segment D à vitesse maximale.

Par contre pour un déplacement négatif, le chemin suivi sera AG GH HI I IJ JL LP1. La trajectoire dépasse le point final afin que le système puisse y revenir avec la vitesse positive $V_{max}/2$.

Il est aussi possible de suivre le chemin hybride AB BL LP1, mais uniquement si la distance à parcourir correspond exactement à une longueur particulière que nous noterons l_c pour longueur critique. Ce chemin peut aussi être analysé comme un chemin du premier type qui n'atteint pas la vitesse maximale

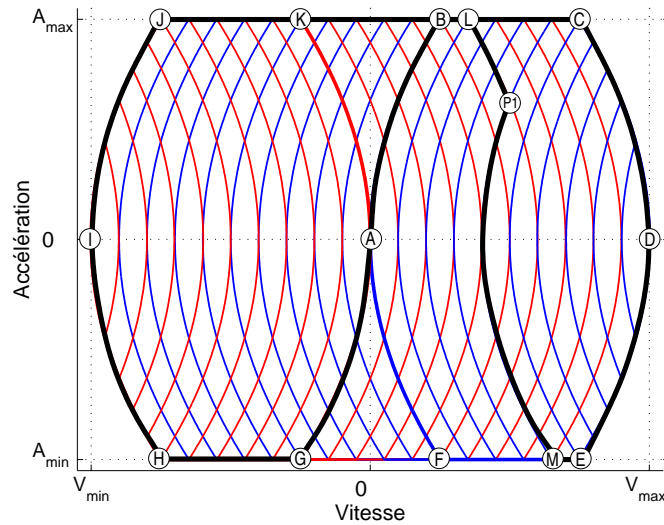


FIG. 3.7 – Espace vitesse - Accélération : de l'arrêt à une situation quelconque

et ne comporte pas de mouvements de décélération ; ou comme un chemin de type 2 qui ne comporte pas la première partie et ne décélère pas. Son principal intérêt est de délimiter les déplacements optimaux : pour un déplacement supérieur à l_c c'est le premier type qui est optimum ; pour un déplacement inférieur à l_c c'est le deuxième type.

3.7 Mouvement multidimensionnel

Les déplacements dans des espaces à plusieurs dimensions réalisent des trajectoires. Celles-ci peuvent être imposées comme dans les déplacements en ligne droite ou partiellement définies, par exemple, par une suite points. Ces trajectoires introduisent des relations entre les déplacements mono-dimensionnels que nous allons étudier ici. Le cas des rotations sera abordé dans l'espace des quaternions.

Nous reprenons la méthode de base et les algorithmes présentés dans les paragraphes précédents pour les étendre aux cas multidimensionnels. Nous présenterons d'abord le cas bidimensionnel.

3.7.1 Mouvement point à point

Considérons deux points \mathbf{X}_1 et \mathbf{X}_2 définis par leurs coordonnées dans un espace à deux dimensions :

$$\mathbf{X}_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad \mathbf{X}_f = \begin{bmatrix} x_f \\ y_f \end{bmatrix} \quad (3.39)$$

Ces deux points définissent un segment de droite qui correspond au chemin le plus court entre ces deux points. Comme pour le cas mono-dimensionnel, nous considérons dans un premier temps que les vitesses et accélérations initiales et finales sont nulles. Nous proposons une stratégie en deux étapes :

1. Calcul du mouvement sur chacune des directions selon ses propres contraintes, en utilisant l'algorithme 1.
2. Ajustement du mouvement en fonction du déplacement de plus longue durée, pour garantir la forme de la trajectoire en ligne droite.

Trajectoire en ligne droite et temps minimal sous contraintes

Considérons deux mouvement élémentaires à jerk maximal $X_1(t)$ et $X_2(t)$, chacun avec sa propre contrainte de jerk, J_{max1} et J_{max2} . Avec $t \in [T_0, T_f]$, tel que les deux mouvements ont la même durée, la trajectoire en ligne droite est défini par :

$$\begin{aligned} X_2(t) &= K_H X_1(t) + X_0 \\ V_2(t) &= K_H V_1(t) \\ A_2(t) &= K_H A_1(t) \\ J_{max1} &= K_H J_{max2} \end{aligned} \quad (3.40)$$

Où X_0 est une constante et K_H est le rapport d'homothétie. Nous trouvons ce même rapport pour les mouvements élémentaires à accélération maximale et à vitesse maximale. Tel que :

$$\begin{aligned} A_{max1} &= K_H A_{max2} \\ V_{max1} &= K_H V_{max2} \end{aligned} \quad (3.41)$$

La première étape consiste à calculer l'évolution temporelle optimale indépendamment dans les deux directions en utilisant les résultats de la méthode de base 3.12.

Dans le cas où les contraintes cinématiques des deux directions sont homothétiques, on sélectionne la durée maximale qui définit les temps T_j , T_a et T_{vc} . On applique ensuite les résultats vue en 3.5 pour recalculer l'évolution dans l'autre direction.

Contraintes non homothétiques

McCarthy dans [McCarthy et al., 1992] établi que pour un bras manipulateur, le mouvement est optimal quand au moins une articulation se déplace avec le couple saturé. Pour le mouvement multidimensionnel limité en jerk, accélération et vitesse, le mouvement est optimal quand à tout instant, au moins un mouvement unidirectionnel sature une de ses contraintes.

Reportons-nous dans l'espace vitesse-accélération, figure 3.8, les contraintes cinématiques relatives à deux directions après avoir fait subir une homothétie aux contraintes de l'une des deux afin que les vitesses maximales soient représentées par les mêmes points A et D.

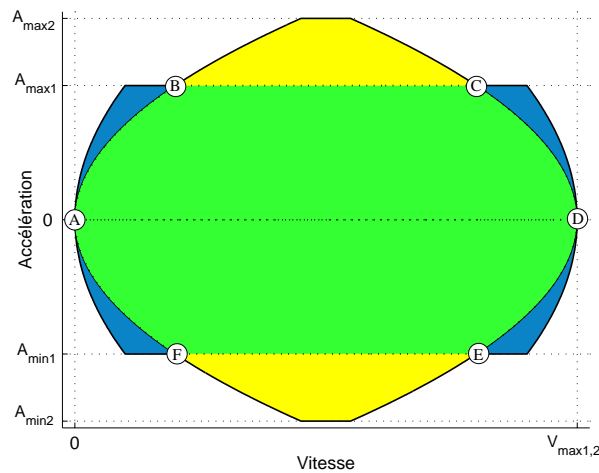


FIG. 3.8 – Espace vitesse - Accélération : contraintes non homothétiques représentées après une homothétie égalant les V_{max} .

La zone exploitable est représentée en vert, elle correspond à l'intersection des deux zones intérieures aux limites. Le plus petit jerk des deux définit les courbes à jerk maximal FA, AB, CD et DE.

Dans la suite de ce mémoire, afin de simplifier la présentation, nous considérerons essentiellement le cas où les contraintes sont homothétiques, il existe $k > 0$ tel que :

$$\begin{aligned} J_{max1} &= k J_{max2} \\ A_{max1} &= k A_{max2} \\ V_{max1} &= k V_{max2} \end{aligned} \quad (3.42)$$

Exemple de mouvement point à point

Soient \mathbf{X}_0 et \mathbf{X}_f les points initial et final d'un déplacement dans un plan :

$$\mathbf{X}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{X}_f = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

Considérons des contraintes cinématiques de jerk, d'accélération et de vitesse identiques dans les deux directions :

$$\begin{aligned} J_{max} &= 6 \text{ m/s}^3 \\ A_{max} &= 2 \text{ m/s}^2 \\ V_{max} &= 1 \text{ m/s.} \end{aligned}$$

La figure 3.9 présente les résultats obtenus avec l'algorithme 1, premier pas de la stratégie. La figure 3.10 présente les résultats obtenus après la correction du jerk, deuxième étape de la stratégie.

L'objectif de la correction est de synchroniser les instants de commutation entre les différents segments dans les deux directions. Dans l'exemple, seulement le déplacement dans la direction X atteint ses contraintes. Le mouvement dans la direction Y a été ralenti en réduisant le jerk (Figure 3.10.a). Nous pouvons constater l'homothétie du mouvement en regardant les jerks sur la figure 3.10.a, les accélérations sur la figure 3.10.b et les vitesses sur la figure 3.10.c. La trajectoire obtenue est bien une ligne droite (Figure 3.10.e).

Le mouvement reste optimal car à tout instant au moins une contrainte reste saturée.

3.7.2 Trajectoire continue en dimension N sur un chemin défini

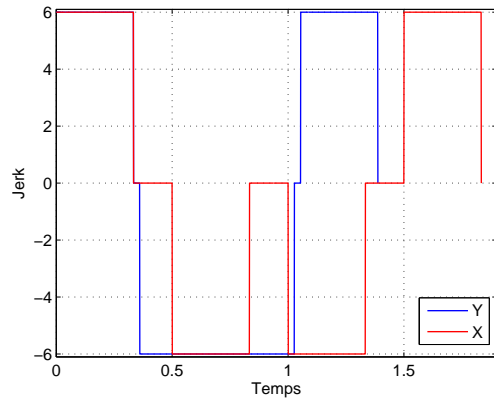
Nous avons défini que pour un chemin Φ défini par $m + 1$ positions, il existe une trajectoire TR qui définit l'évolution temporelle sur le chemin dans l'intervalle de temps $[T_0, T_f]$. À l'instant courant $T_c \in [T_0, T_f]$, la position courante est $X_c = X(T_c)$;

A partir des résultats précédents, lorsque la trajectoire est définie par une suite de $N + 1$ situations S_i ($0 \leq i \leq N$) définies chacune par sa position X_i , sa vitesse V_i et son accélération A_i , nous avons vu comment calculer la trajectoire.

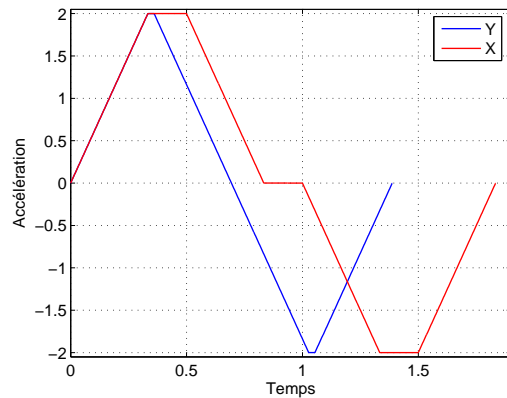
Toutefois la connaissance des vitesses et des accélérations n'est pas systématique. Se pose alors le problème du calcul des vitesses et accélérations intermédiaires.

Arrêt en chaque point

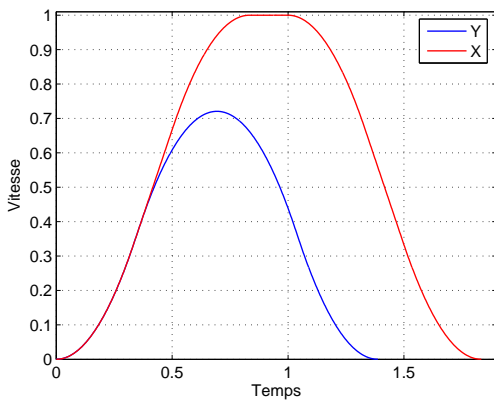
Une solution simple consiste à arrêter le mouvement en chaque point. Cette solution est la seule qui permette au repère contrôlé de se déplacer sur la ligne polygonale définie par les points. En effet le passage d'un sommet d'une ligne brisée avec une vitesse non nulle entraîne des accélérations infinies. Son calcul ne pose pas de problème.



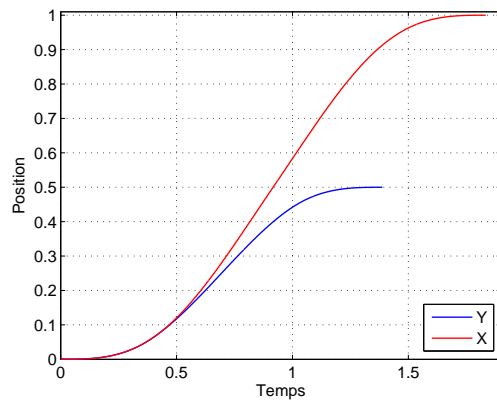
(a) Jerk



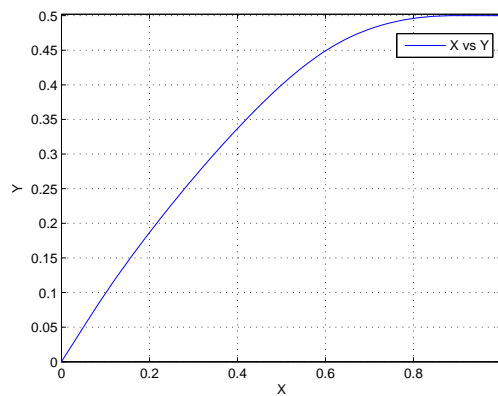
(b) Accélération



(c) Vitesse



(d) Position



(e) Position XY

FIG. 3.9 – Mouvement point à point en deux dimensions (X,Y) : Première étape de la stratégie, calcul de la trajectoire dans chacune des directions

Déplacement ne suivant pas la ligne polygonale

Il est aussi possible de définir une tangente en chaque point, par exemple la perpendiculaire à la bissectrice, et de calculer la vitesse maximale de passage sur ce point avec une accélération nulle. Cette

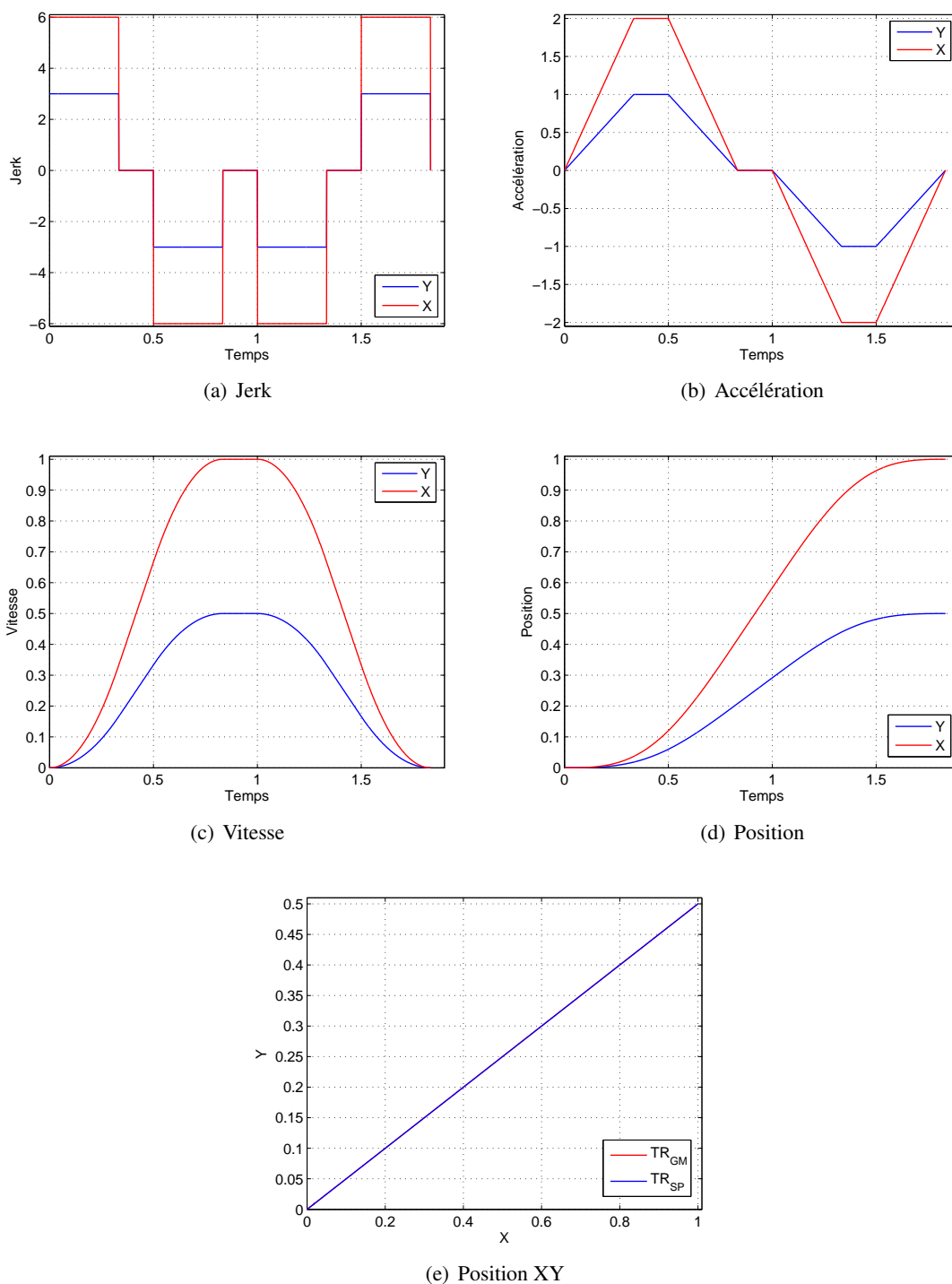


FIG. 3.10 – Mouvement point à point en deux dimensions (X,Y) : homothétie dans le mouvement par correction du jerk

approche conduit à des trajectoires qui passent par les points mais qui sont éloignées du polygone.

Une autre possibilité plus adaptée à la commande où l'on découvre les points au cours du temps,

consiste à aller vers le point en supposant que l'on va s'y arrêter et à commuter pour aller vers le nouveau point dès qu'il apparaît.

Déplacement sur la ligne polygonale avec virages

Une autre approche consiste à suivre le polygone en définissant des virages qui permette de contrôler la position du chemin effectivement suivi par rapport au polygone.

On peut distinguer plusieurs stratégies suivant que l'on accepte ou non les dépassements. Nous allons en présenter deux. La première coupe les virages en restant sur le polygone le plus possible et en saturant les contraintes dans les virages. La deuxième parcourt tout le polygone en saturant les contraintes, mais effectue les virages à l'extérieur.

Les travaux de Lloyd [Lloyd et al., 1993] présentent une méthode pour la transition entre différents segments utilisant une fonction polynomiale pour le lissage de la transition. Des paramètres d'ajustement permettent de modifier la forme de la transition.

Mouvement avec virages coupés

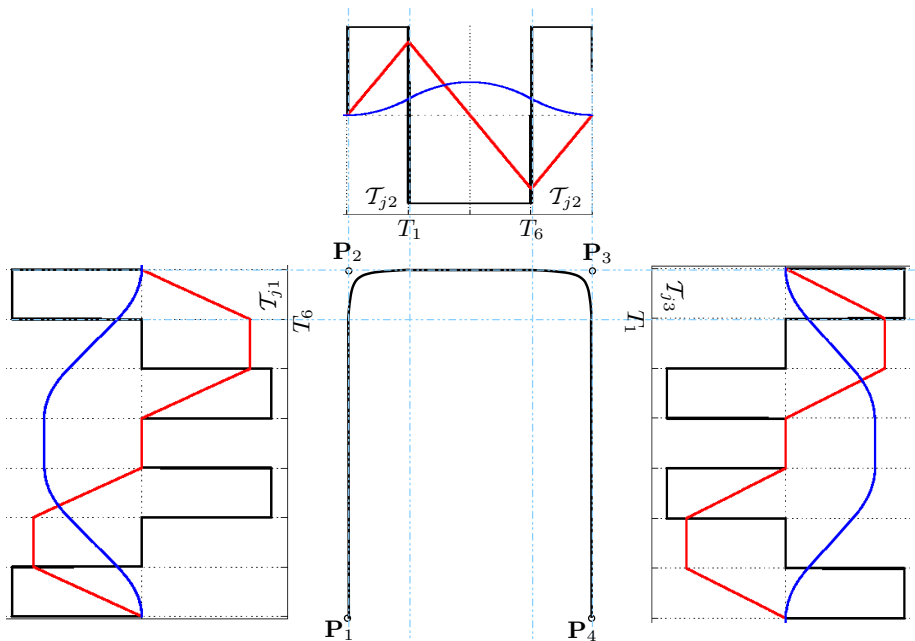


FIG. 3.11 – Virage entre segments de droites avec des points de contrôle.

La figure 3.11 présente trois segments ($[P_1P_2], [P_2P_3]$ et $[P_3P_4]$). A partir des résultats obtenus en 3.7, nous pouvons calculer la trajectoire qui suit le chemin en s'arrêtant en P_2 et P_3 . A partir de cette première trajectoire, il est possible de choisir de commencer le segment suivant à un instant de la phase de décélération du premier. Suivant l'instant, le chemin comportera des virages plus ou moins importants. Pour la figure 3.11, l'instant de commutation choisi est T_6 .

Nous avons établi que le mouvement, avec des conditions cinématiques initiales et finales nulles, peut être réalisé en trois, quatre ou cinq segments. Ce qui produit un, trois ou cinq segments intermédiaires entre le segment de démarrage et le segment d'arrêt. En considérant la méthode de base, le segment de démarrage T_{jpa} termine à l'instant T_1 et le segment d'arrêt T_{jpb} démarre à l'instant T_6 .

Mouvement avec boucle aux points de passage

Considérons un chemin Φ que nous souhaitons parcourir à la vitesse maximale, chaque sommet de la ligne définit deux vitesses de directions différentes. Pour ces configurations, l'accélération est nulle. Il est alors possible d'effectuer un déplacement de transition minimal en temps et reliant ces deux configurations en restant à l'extérieur de la ligne. La figure 3.12 présente une telle boucle.

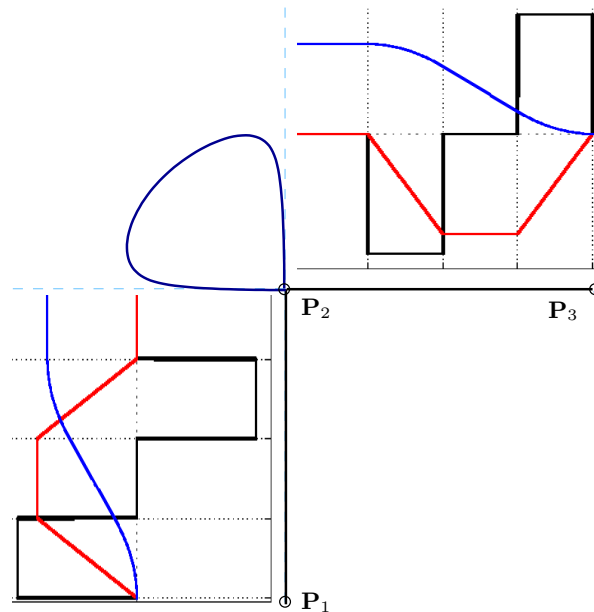


FIG. 3.12 – Virage entre segments de droites parcourus à vitesse constantes

3.7.3 Considérations sur la génération du mouvement

La stratégie utilisée pour le mouvement continu à une dimension s'étend donc au mouvement à plusieurs dimensions. Cette extension introduit des points supplémentaires : l'ajustement de la durée, la sélection du temps de commutation, la correction du jerk, les contraintes non homothétiques.

Nous allons, dans les paragraphes suivants, revenir sur ces points importants puis nous présenterons un exemple de trajectoire continue à deux dimensions.

Sélection du temps de commutation

Comme nous l'avons défini, ce problème apparaît quand le chemin Φ est connu à l'avance. Nous avons établi que pour garder la condition de temps minimal sous contraintes, le mouvement dans le même sens à une dimension doit démarrer vers la position suivante après le temps \mathcal{T}_{aca} .

Pour visualiser l'effet de la sélection du temps de commutation, considérons l'exemple suivant. Soit un chemin Φ défini par trois points en deux dimensions, X et Y ,

$$\Phi = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0.5 & 2 \end{bmatrix}$$

avec des contraintes cinématiques identiques pour les deux directions :

$$\begin{aligned} J_{max} &= 6m/s^3 \\ A_{max} &= 2m/s^2 \\ V_{max} &= 1m/s \end{aligned}$$

Nous avons choisi 5 temps de commutation différents :

$$T_c = [0.25 \quad 0.50 \quad 0.75 \quad 1.00 \quad 1.25]$$

Avec ces temps de commutation, nous obtenons les courbes de vitesse de la figure 3.13. La correction de la durée à été réalisé, nous reviendrons sur ce sujet plus tard.

La sixième image de la figure 3.13 présente les trajectoires obtenues pour chaque instant de commutation. Nous vérifions sur les courbes de vitesse que le temps le plus petit correspond au temps de commutation le plus petit, cependant l'écart à la trajectoire de consigne Φ est plus grand. Il y a un compromis à trouver entre la durée du mouvement pour arriver à la position finale et l'écart avec la trajectoire.

La stratégie que nous utilisons pour réaliser un mouvement sur le chemin Φ , des trajectoires en ligne droite, consiste à effectuer une série de mouvements point à point en considérant que les vitesses et les accélérations sont nulles en chaque point de Φ .

Reprenons l'exemple, le mouvement point à point pour la première destination ($X = 1, Y = 0.5$) est réalisé avec les temps $T_{jpa} = 0.3333$, $T_{aca} = 0.1667$, $T_{jna} = 0.33333$, $T_{vc} = 0.1667$, $T_{jnb} = 0.3333$, $T_{acb} = 0.1667$ et $T_{acb} = 0.3333$. Analysons chacune des courbes :

$T_c = 0.25$: La première courbe de vitesse est obtenu avec $T_c < T_{jpa}$, le mouvement vers la première position n'atteint pas son accélération maximale quand il va vers la deuxième position. C'est la durée la plus courte mais l'écart le plus grand.

$T_c = 0.50$: Pour le deuxième temps de commutation, $T_c = T_{jpa} + T_{aca}$, le mouvement vers la destination finale démarre quand le mouvement est dans une période d'accélération maximale, équivalant au temps minimal dans le mouvement à une dimension.

$T_c = 0.75$: La troisième courbe est calculée avec $T_c < T_{jpa} + T_{aca} + T_{jna}$, le mouvement vers le premier point n'atteint pas sa vitesse maximale, quand il va vers la deuxième position.

$T_c = 1.00$: Pour la quatrième courbe avec $T_c = T_{jpa} + T_{aca} + T_{jna} + T_{vc}$, l'accélération est nulle et le mouvement vers la destination intermédiaire atteint et se déplace à sa vitesse maximale, le mouvement vers la destination finale démarre avant la période de décélération.

$T_c = 1.25$: La cinquième courbe est obtenu pendant la période de décélération, lorsque l'accélération est négative le mouvement vers la destination finales doit changer cette tendance. La durée de déplacement va vers la durée des deux mouvements indépendants.

En considérant la durée de mouvement et l'écart de la trajectoire, nous avons choisi un temps de commutation

$$T_c = T_{jpa} + T_{aca} + T_{jna} + T_{vc} \quad (3.43)$$

qui correspond à la période d'accélération entre points successifs.

La figure 3.14 présente une série de chemins Φ de trois points avec les deux premiers points égaux et les destinations finales (troisième point) différentes par la coordonnée Y .

$$\Phi = \begin{bmatrix} 0 & 1 & 1.5 \\ 0 & 1.5 & Y_2 \end{bmatrix}$$

Les coordonnées des destinations finales sont $X_2 = 1.5$ et $Y_2 = [1.5 \quad 2.0 \quad 2.5 \quad 3.0 \quad 3.5]$.

En utilisant le temps de commutation défini en 3.43, nous obtenons la série de trajectoires présentées sur la figure. Nous pouvons observer la formation d'un « coude » avec les différentes courbes qui restent dans une région au voisinage de la destination intermédiaire.

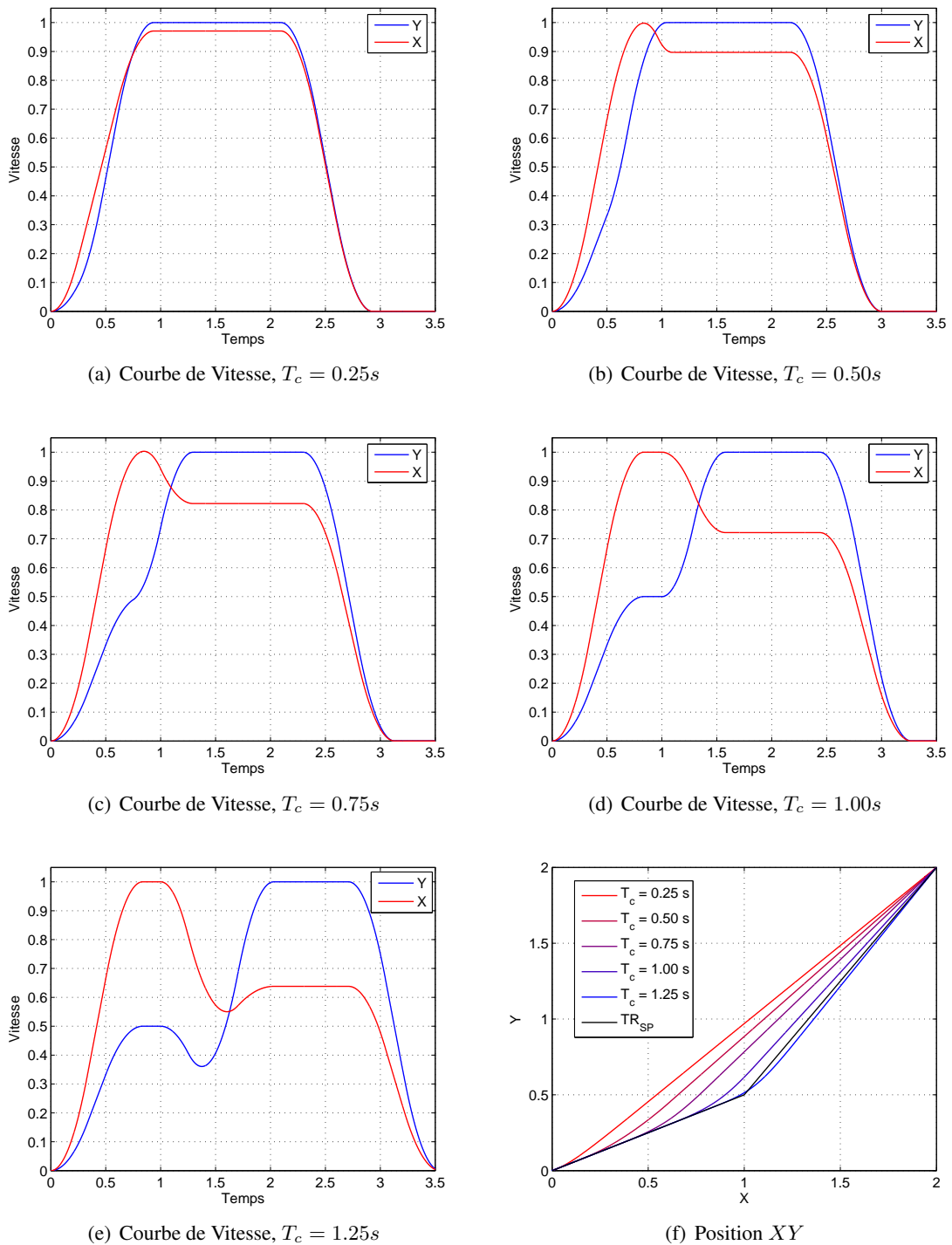


FIG. 3.13 – Exemple de trajectoires obtenues pour différents temps de commutation. Les premières courbes montrent les vitesses obtenues pour $T_c = [0.25 \ 0.50 \ 0.75 \ 1.00 \ 1.25]$, la dernière présente de façon comparative les trajectoires.

Correction de la durée du mouvement

De la même façon que pour le mouvement point à point, la durée du mouvement devra être ajustée pour garantir le mouvement en ligne droite. Lorsque les mouvements dans chacune des dimensions sont

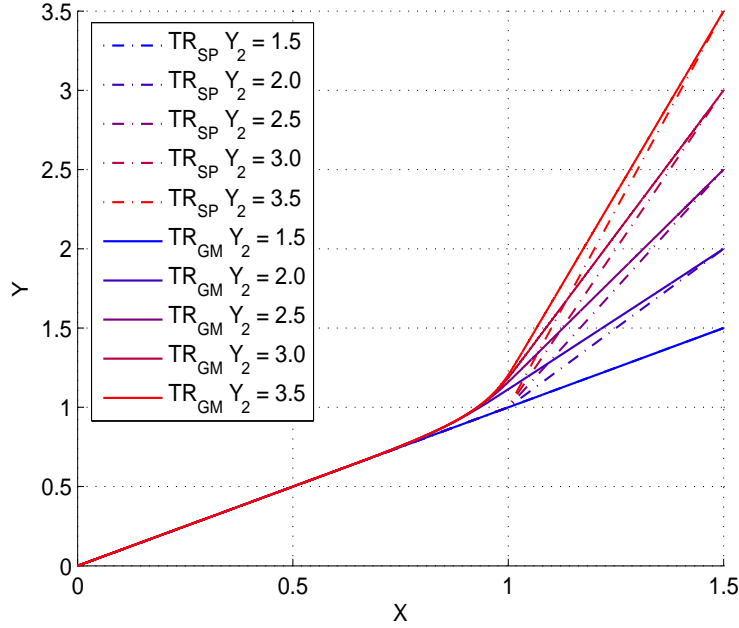


FIG. 3.14 – Effet de « coude » produit par les différentes destinations finales pour une même trajectoire initiale.

indépendants, les différents types de mouvement peuvent être présents pour un mouvement de \mathbf{X}_i à \mathbf{X}_{i+1} . Avec cette caractéristique, nous ne pouvons pas parler d'une homothétie sinon des mouvements de durée identique.

Certaines règles heuristiques doivent être considérées pour garantir le mouvement en ligne droite quand il est possible de le réaliser ou pour garantir les conditions de mouvement continu.

- R1 : À l'intérieur d'un espace n -dimensionnel le mouvement peut être réalisé sur toutes les dimensions ou sur m dimensions, où $m \in [0, n]$.
- R2 : Pour les caractéristiques de mouvement sous-optimal, au moins le mouvement dans une direction doit être réalisé en contraintes saturées.
- R3 : Le mouvement vers la destination suivante sera découpé en deux parties.
- R4 : Le mouvement de décélération vers la destination finale doit commencer au même instant pour garantir le déplacement rectiligne dans cette partie du mouvement.

Pour les n -directions calculer les temps indépendants. Considérons les différents cas :

- C1 : Pour les $(n - m)$ directions où il n'y pas de déplacement la vitesse reste nulle.
- C2 : Pour les directions où le mouvement est de démarrage. Déterminer le plus long temps de démarrage pour la période d'accélération et la période de décélération, qui correspondent aux deux parties de mouvement. Considérons

$$\begin{aligned}
 T_{DA} &= [\mathcal{T}_{jpa} \quad \mathcal{T}_{aca} \quad \mathcal{T}_{jna} \quad \mathcal{T}_{vc}] \\
 T_{DF} &= [\mathcal{T}_{jnb} \quad \mathcal{T}_{acb} \quad \mathcal{T}_{jpb}] \\
 T_{sDA} &= \mathcal{T}_{jpa} + \mathcal{T}_{aca} + \mathcal{T}_{jna} + \mathcal{T}_{vc} \\
 T_{sDF} &= \mathcal{T}_{jnb} + \mathcal{T}_{acb} + \mathcal{T}_{jpb}
 \end{aligned} \tag{3.44}$$

où les indices D correspondent au démarrage, A à la période d'accélération, F à la période de décélération, et s à la somme.

C3 : Pour les mouvements dans le même sens, le temps le plus long est considéré, tel que nous avons une période de ré-accélération et une période de décélération.

$$\begin{aligned} T_{MR} &= [\mathcal{T}_{jpa} \quad \mathcal{T}_{aca} \quad \mathcal{T}_{jna} \quad \mathcal{T}_{vc}] \\ T_{MF} &= [\mathcal{T}_{jnb} \quad \mathcal{T}_{acb} \quad \mathcal{T}_{jpb}] \\ T_{sMR} &= \mathcal{T}_{jpa} + \mathcal{T}_{aca} + \mathcal{T}_{jna} + \mathcal{T}_{vc} \\ T_{sMF} &= \mathcal{T}_{jnb} + \mathcal{T}_{acb} + \mathcal{T}_{jpb} \end{aligned} \quad (3.45)$$

où les indices M correspondent à un même sens et R à la période de ré-accélération.

C4 : Pour le mouvement d'arrêt, nous considérons le temps le plus long, le mouvement d'arrêt a une seule partie, la partie de décélération, telle que :

$$\begin{aligned} T_{SF} &= [\mathcal{T}_{jnb} \quad \mathcal{T}_{acb} \quad \mathcal{T}_{jpb}] \\ T_{sSF} &= \mathcal{T}_{jnb} + \mathcal{T}_{acb} + \mathcal{T}_{jpb} \end{aligned} \quad (3.46)$$

où l'indice S correspond à l'arrêt.

C5 : Pour le changement de sens, nous avons établi que le mouvement est réalisé en deux parties, un mouvement d'arrêt suivi d'un mouvement de démarrage dans le sens contraire. Lorsqu'il y a un double mouvement, la longueur totale de la période de ré-accélération considère le mouvement d'arrêt dans un sens et l'accélération dans le sens contraire. La période de décélération correspond à la décélération du deuxième mouvement.

$$\begin{aligned} T_{CR} &= [\mathcal{T}_{jnS} \quad \mathcal{T}_{acS} \quad \mathcal{T}_{jpS} \quad \mathcal{T}_{jpa} \quad \mathcal{T}_{aca} \quad \mathcal{T}_{jna} \quad \mathcal{T}_{vc}] \\ T_{CF} &= [\mathcal{T}_{jnb} \quad \mathcal{T}_{acb} \quad \mathcal{T}_{jpb}] \\ T_{sSC} &= \mathcal{T}_{jnS} + \mathcal{T}_{acS} + \mathcal{T}_{jpS} \\ T_{sCA} &= \mathcal{T}_{jpa} + \mathcal{T}_{aca} + \mathcal{T}_{jna} + \mathcal{T}_{vc} \\ T_{sCR} &= T_{sSC} + T_{sCA} \\ T_{sCF} &= \mathcal{T}_{jnb} + \mathcal{T}_{acb} + \mathcal{T}_{jpb} \end{aligned} \quad (3.47)$$

où l'indice C correspond à un changement de sens.

Avant de continuer, reprenons certaines considérations pour les mouvement souples :

- La méthode de base considère des conditions nulles de vitesse et d'accélération au début et à la fin de mouvement, et dans le cas à plusieurs dimensions, nécessite ces conditions pour toutes les directions.
- Pour garantir le mouvement rectiligne entre deux points à plusieurs dimensions, les mouvements devront être homothétiques.
- Pour le mouvement continu à une dimension, nous avons relaxé les conditions d'optimalité en découpant le changement de sens en deux.
- Le mouvement de démarrage dans une ou plusieurs directions au milieu d'une trajectoire permet l'intégration d'une nouvelle direction de façon souple.
- Le mouvement d'arrêt d'une ou plusieurs directions doit considérer que l'arrêt soit réalisé avant de considérer le mouvement vers la position suivante.

Définissons deux vecteurs, l'un pour la première partie du mouvement et l'autre pour la deuxième, tel que :

$$\begin{aligned} T_{m1} &= [T_{sDA} \quad T_{sMR} \quad T_{sSF} \quad T_{sCR}] \\ T_{m2} &= [T_{sDF} \quad T_{sMF} \quad T_{sCF}] \end{aligned} \quad (3.48)$$

Calculons les temps maximaux pour chacune des parties :

$$T_{max1} = \max(T_{m1}) \quad T_{max2} = \max(T_{m2}) \quad (3.49)$$

La durée du mouvement pour atteindre la position suivante est donné par

$$T_{PS} = T_{max1} + T_{max2} \quad (3.50)$$

Les vecteurs de temps pour chaque partie du mouvement sont ajustés selon le type de mouvement. Pour le mouvement de démarrage :

$$T_{DAN} = \frac{T_{sDA}}{T_{max1}} T_{DA} \quad T_{DFN} = \frac{T_{sDF}}{T_{max2}} T_{DF} \quad (3.51)$$

Pour le mouvement dans le même sens l'ajustement de la durée est donné par :

$$T_{MRN} = \frac{T_{sMR}}{T_{max1}} T_{MR} \quad T_{MFN} = \frac{T_{sMF}}{T_{max2}} T_{MF} \quad (3.52)$$

Le mouvement d'arrêt reste sans changement, car il a été calculé pour l'utilisation de J_{max} , un changement de la durée en fonction de T_{max1} demanderait un jerk plus grand. Nous avons établi que le mouvement d'arrêt doit finir dans la première partie du mouvement vers la position suivante, mais sa longueur ne peut pas être modifiée sans relaxer les contraintes, donc il affecte T_{max1} mais il n'est pas modifié, en d'autres mots, T_{max1} peut être égal à T_{sSF} mais le vecteur T_{SF} n'est pas corrigé par un rapport T_{sSF}/T_{max1} .

Pour le changement de sens, les temps du premier mouvement ne sont pas modifiés seulement les temps du deuxième mouvement, tel que les vecteurs de temps deviennent :

$$\begin{aligned} T_{CRN} &= \left[T_{jnS} \quad T_{acS} \quad T_{jpS} \quad \frac{T_{sCA}}{T_{max1}} T_{jpa} \quad \frac{T_{sCA}}{T_{max1}} T_{aca} \quad \frac{T_{sCA}}{T_{max1}} T_{jna} \quad \frac{T_{sCA}}{T_{max1}} T_{vc} \right] \\ T_{CFN} &= \frac{T_{sCF}}{T_{max2}} T_{CF} \end{aligned} \quad (3.53)$$

Ajustement du jerk

Après la correction de la durée, les jerks utilisés durant chaque intervalle pendant les périodes de T_{jpa} , T_{jna} , T_{jnb} et T_{jpb} doivent être ajustés. Certaines considérations sont prises selon le type de mouvement.

J1 : Considérons le mouvement d'arrêt, nous avons établi que les temps n'ont pas été modifiés, donc les jerks restent sans modification.

J2 : Pour le mouvement de démarrage, les simplifications faites pour le mouvement point à point ne sont plus applicables lorsque $T_{max1} \neq T_{max2}$, il n'y a pas de condition de symétrie par rapport à la vitesse. Les simplifications applicables dans ce type de mouvement sont $J_{Tjna} = -J_{Tjpa}$ et $J_{Tjnb} = -J_{Tjpb}$.

En utilisant les temps de T_{DAN} et T_{DFN} , les jerks J_{Tjpa} et J_{Tjpb} sont exprimés en fonction d'une vitesse auxiliaire V_{rm} , qui peut être ponctuelle ou pas selon l'existence de T_{vc} . L'objectif est de calculer la vitesse auxiliaire V_{rm} qui résout le système d'équations pour la distance désirée.

J3 : Pour le mouvement dans le même sens, en utilisant les temps de T_{MRN} et T_{MFN} , les jerks sont calculés en utilisant une vitesse auxiliaire V_{rm} . La seule simplification que nous pouvons faire est $J_{Tjnb} = -J_{Tjpb}$.

Pour le calcul de V_{rm} considérons la figure 3.6, où le point **M** représente les conditions cinématiques initiales et les points **P1** et **P2** les différentes valeurs de V_{rm} . Pour aller au point **A** depuis le point **M** en passant par le point **P1** ou **P2**, la séquence de segments a un ordre différent. Pour $\overline{MP1}$ le segment initial est T_{jpa} et pour $\overline{MP2}$ est T_{jna} . Donc, l'ordre des segments peut être altéré, lorsque la valeur de la vitesse auxiliaire V_{rm} est plus grande ou plus petite que la vitesse initiale V_0 , par exemple sur la figure 3.13.d. Dans la direction *Y*, l'ordre du deuxième mouvement est T_{jpa} suivi de T_{jna} du deuxième mouvement, alors que pour la direction *X* l'ordre est inversé.

Un cas spécial apparaît quand la durée de la première partie du mouvement est « relativement beaucoup » plus grande que la durée de la période de ré-accélération propre, $T_{max1} \gg T_{MR}$, ce qui demande une période à vitesse constante qui permettra l'égalité des durées. Pour résoudre ce type de problème l'ordre des premiers segments est inversé T_{jna} suivi de T_{jpa} et le système doit être résolu pour T_{jpa} , T_{jna} , T_{aca} et $T_{vc} = T_{max1} - T_{jpa} - T_{aca} - T_{jna}$ définis en fonction de la vitesse auxiliaire V_{rm} qui sera constant en T_{vc} , avec $V_{rm} < \max$.

J4 : Pour le mouvement avec changement de sens, nous avons établi que les temps d'arrêt du premier mouvement ne changent pas, pourtant les jerks concernés par ces temps restent sans changement. Les jerks du deuxième mouvement sont modifiés selon les considérations faites pour le mouvement de démarrage.

La figure 3.15 présente du côté gauche les temps calculés pour chaque type de mouvement, et du côté droit les résultats obtenus après les corrections en durée et en jerk.

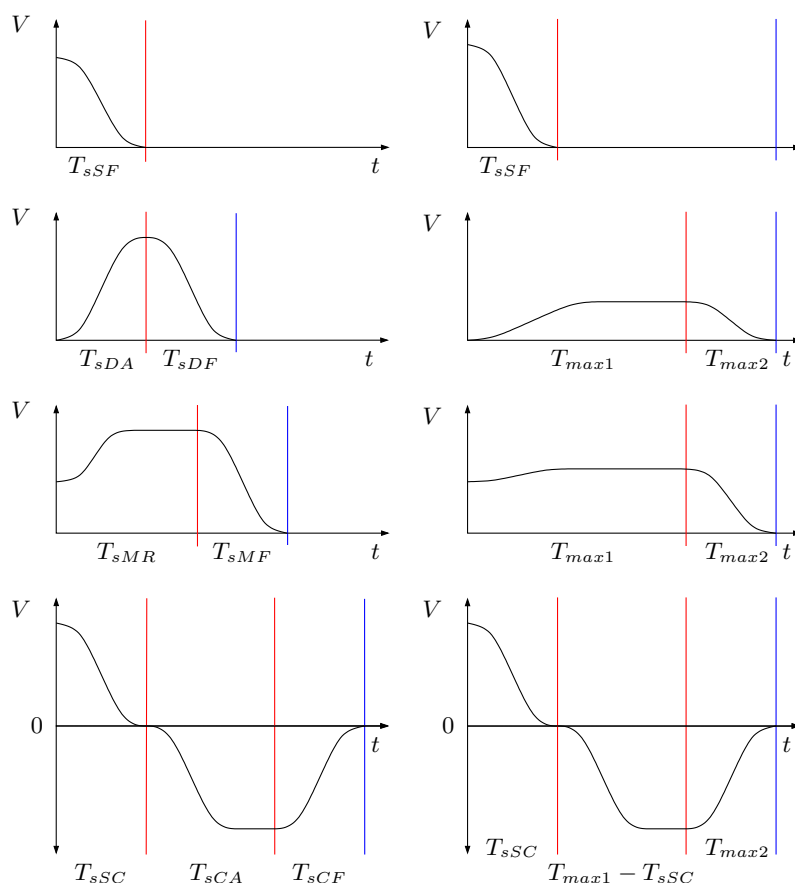


FIG. 3.15 – Corrections en durée et en jerk pour les quatre types de mouvements

Sur le côté droit de la figure 3.15 on constate que pour la première partie du mouvement il n'y a pas d'homothétie tandis que pour la deuxième partie, la décélération, les mouvements sont homothétiques et permettent d'avoir une trajectoire en ligne droite vers la destination finale.

3.7.4 Contraintes non homothétiques

Pour les mouvements satisfaisant la condition de sept segments 3.38, la condition d'homothétie est définie par :

$$\frac{J_{max1}}{J_{max2}} = \frac{A_{max1}}{A_{max2}} = \frac{V_{max1}}{V_{max2}} \quad (3.54)$$

Si la condition n'est pas satisfaite, le mouvement en ligne droite optimal n'est plus garanti. Nous proposons deux types de solutions pour résoudre ce type de contraintes.

Mouvement optimal non homothétique

Considérons un mouvement sur N dimensions, les contraintes des n_1 premières directions satisfassent la condition de homothétie, avec $n_1 < N$, pendant que les $N - n_1$ directions restantes ne les satisfassent pas.

Supposons que le mouvement a une durée maximale T_{max} sur une des premières n_1 directions.

Pour la direction i avec $i \in]n_1, N]$ avec des temps optimaux T_{ji} pour les intervalles à jerk maximal, T_{ai} pour les intervalles à accélération maximale et T_{vi} pour l'intervalle à vitesse maximale. La durée T_{ni} est calculée selon l'équation 3.33. La durée est corrigée à la durée maximale en faisant :

$$T_{ji}^* = \frac{T_{max}}{T_{ni}} T_{ji} \quad T_{ai}^* = \frac{T_{max}}{T_{ni}} T_{ai} \quad T_{vi}^* = \frac{T_{max}}{T_{ni}} T_{vi} \quad (3.55)$$

En utilisant les temps corrigés, le jerk devra être recalculé.

Le mouvement obtenu est optimal lorsqu'il reste optimal dans au moins une direction, le mouvement obtenu n'est plus une ligne droite mais une courbe dans l'espace à N directions.

Exemple en deux dimensions

Considérons la trajectoire dans l'espace à deux dimensions, X et Y :

$$\Phi = \begin{bmatrix} 0.0 & 1.0 & 1.2 \\ 0.0 & 0.5 & 1.5 \end{bmatrix} \quad (3.56)$$

Les contraintes cinématiques pour chacune des directions sont les suivantes

Direction	X	Y
J_{max}	6.0 m/s^3	3.0 m/s^3
A_{max}	2.0 m/s^2	1.0 m/s^2
V_{max}	1.0 m/s	0.5 m/s

La figure 3.16 montre les courbes de jerk, d'accélération, de vitesse et de position dans l'espace pour cette trajectoire.

Pendant la première seconde, la trajectoire correspond à la première partie du mouvement, c'est dans cette période que le mouvement atteint l'accélération maximale dans chaque dimension, suivi d'une période à vitesse maximale dans les deux directions. Si nous considérons une seule position, c'est après de la première seconde que le mouvement de décélération devrait commencer (en pointillés sur les graphiques). Comme il y a un mouvement vers une deuxième position, c'est après la première seconde qu'il démarre.

Dans le cas particulier de la trajectoire vers le troisième point, la vitesse maximale est conservée dans la direction Y , pendant que la direction X doit ralentir. Sur les figures 3.16.a et 3.16.b, il est possible de vérifier un temps d'accélération constant plus petit que le nécessaire pour arrêter le mouvement dans la direction X , qui produit une vitesse constante qui va durer jusqu'à la troisième seconde où le mouvement d'arrêt démarre. Lorsque la vitesse dans la direction Y est maximale, la période de décélération est maximale pour Y , pendant que pour la direction X , les jerks ont été corrigées pour garantir le mouvement homothétique et le segment de trajectoire final en ligne droite. Dans ce cas particulier, le mouvement homothétique démarre à l'instant où les deux directions ont une vitesse constante.

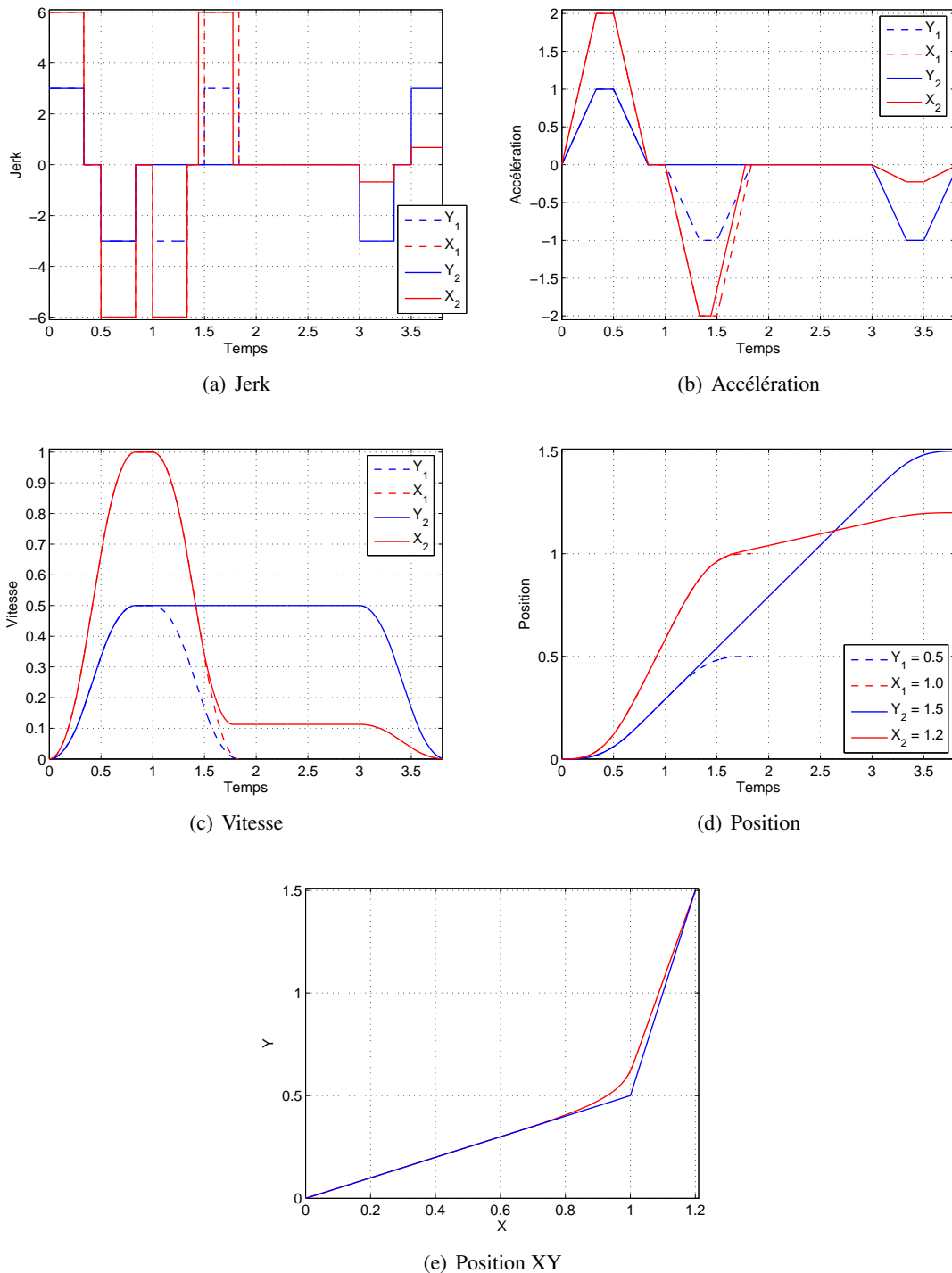


FIG. 3.16 – Exemple de mouvement à deux dimensions pour une trajectoire définie par trois points. Les courbes de jerk, d'accélération, de vitesse de position indépendante et de position pour le mouvement composé sont présentées. Les courbes pointillés représentent la trajectoire vers la position intermédiaire.

3.8 Considérations sur les applications

3.8.1 Cas du bras manipulateur

Dans ce paragraphe, nous nous intéressons à l'application des résultats précédents à la génération des mouvements d'un bras manipulateur. Nous avons établi dans le chapitre précédent que la commande des bras manipulateurs peut être réalisée dans l'espace articulaire ou dans l'espace opérationnel. Nous allons considérer les deux approches successivement dans les paragraphes suivants.

Mouvement dans l'espace articulaire

Une configuration est définie par un vecteur des n coordonnées généralisées,

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \quad (3.57)$$

Une situation est aussi définie par les vitesses $\dot{\boldsymbol{\theta}}$ et les accélérations $\ddot{\boldsymbol{\theta}}$.

Une trajectoire Φ est définie dans l'espace articulaire comme une série de m configurations

$$\Phi = [\boldsymbol{\theta}_1 \quad \boldsymbol{\theta}_2 \quad \cdots \quad \boldsymbol{\theta}_m] = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,m} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,m} \\ \vdots & \vdots & \cdots & \vdots \\ \theta_{n,1} & \theta_{n,2} & \cdots & \theta_{n,m} \end{bmatrix} \quad (3.58)$$

Chaque ligne correspond à l'évolution d'une articulation. Pour un robot à 6 articulations, la trajectoire est une matrice de dimension $6 \times m$.

Chaque articulation est traitée comme une direction indépendante avec ses propres contraintes cinématiques. La stratégie que nous avons présentée pour le mouvement continue en dimension N s'applique.

Mouvement dans l'espace opérationnel

L'objectif est ici de définir le mouvement de l'organe terminal du bras manipulateur dans l'espace opérationnel. La situation de l'organe terminal est décrite par sa position et son orientation que nous représentons par un quaternion unitaire.

$$\mathbf{X} = \begin{bmatrix} \mathbf{P} \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ n \\ i \\ j \\ k \end{bmatrix} \quad (3.59)$$

Une trajectoire Φ est définie dans l'espace opérationnel par une suite de m situations

$$\Phi = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \cdots \quad \mathbf{X}_m] = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ y_1 & y_2 & \cdots & y_m \\ z_1 & z_2 & \cdots & z_m \\ n_1 & n_2 & \cdots & n_m \\ i_1 & i_2 & \cdots & i_m \\ j_1 & j_2 & \cdots & j_m \\ k_1 & k_2 & \cdots & k_m \end{bmatrix} \quad (3.60)$$

Nous obtenons une matrice Φ de taille $7 \times m$. Les trois premières lignes qui correspondent aux coordonnées cartésiennes de position sont traitées selon la stratégie de mouvement multidimensionnel. L'orientation, utilisant les quaternions, est traitée de façon similaire, ce qui est équivalent à une interpolation linéaire dans l'espace des quaternions.

Shoemake dans [Shoemake, 1985] introduit le *slerp* (de l'anglais **spherical linear interpolation**) pour corriger les problèmes de l'interpolation linéaire. La forme de l'espace des quaternions unitaires est une hyper-sphère de dimension 4 et de rayon égal à 1. Le *slerp* produit un arc sur la surface de l'hyper-sphère où ses échantillons sont uniformément distribués. L'interpolation linéaire correspond à une ligne droite située à l'intérieur de cette hyper-sphère. Lorsque l'interpolation sur la ligne droite ne correspond pas à un quaternion unitaire, chaque quaternion produit par interpolation linéaire devra être normalisé. Par simplicité nous appellerons cette méthode comme *nlerp* (de l'anglais **normalized linear interpolation**). Le *nlerp* produit une distribution non uniforme sur la surface de l'hyper-sphère.

La figure 3.17 présente la comparaison entre le *slerp* et le *nlerp* pour une rotation 90° et une rotation de 180° , les deux autour de l'axe Z .

Afin de comparer les deux méthodes, la figure 3.18 présente la mesure de la variation entre deux interpolations successives pour une rotation de 90° en utilisant le *slerp* $\Delta \mathbf{Q}_s$ et le *nlerp* $\Delta \mathbf{Q}_n$. Le deuxième critère de comparaison est la rotation relative entre le quaternion généré par le *slerp* \mathbf{Q}_s et celui généré par le *nlerp* \mathbf{Q}_n .

Un troisième paramètre de comparaison est le nombre d'échantillons. Sur la figure 3.18 les résultats obtenus sont présentés pour 100 et 1000 échantillons.

Les courbes de la figure 3.18 montrent la variation la différence entre le *slerp* et le *nlerp*. Une variation non uniforme mais symétrique est mesurée. Cette symétrie conduit à une erreur d'interpolation nulle à mi-chemin. L'erreur d'interpolation la plus importante (figures (b) et (d)) se trouve au quart du déplacement total qui correspond à la même variation entre le *slerp* et le *nlerp*, images (a) et (c). Il faut noter que l'erreur d'interpolation maximale entre les deux méthodes est indépendante du nombre d'échantillons.

Considérons deux quaternions \mathbf{Q}_a et \mathbf{Q}_b , le quaternion relatif entre eux est donné par

$$\mathbf{Q}_r = \mathbf{Q}_b^{-1} \mathbf{Q}_a$$

voir l'annexe A.2 pour plus de détails.

Dans l'espace des quaternions l'angle entre deux quaternions est défini par le produit scalaire :

$$\alpha_Q = \cos^{-1}(n_a n_b + \mathbf{q}_a^T \mathbf{q}_b) = \cos^{-1} n_r \quad (3.61)$$

où n_r est la composante n du quaternion relatif \mathbf{Q}_r .

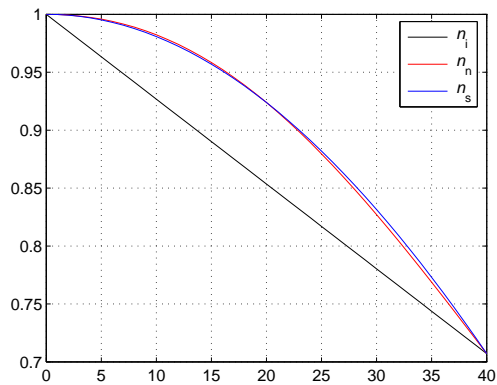
En utilisant la représentation en angle α et axe équivalent \mathbf{e}_{eq} du quaternion relatif, l'angle relatif α_r entre deux rotations dans l'espace cartésien est donné par :

$$\alpha_r = 2 \cos^{-1} n_r \quad (3.62)$$

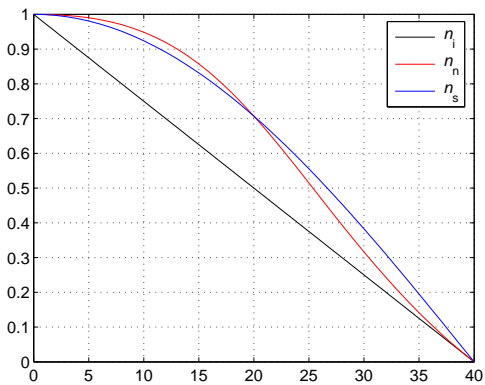
Nous pouvons faire deux remarques :

R1 : Une rotation d'angle α dans l'espace cartésien correspond à une rotation d'angle $\alpha/2$ dans l'espace des quaternions. Par exemple, considérons $\mathbf{Q}_a = [1 \ 0 \ 0 \ 0]^T$ qui correspond au quaternion identité, et $\mathbf{Q}_b = [0 \ 0 \ 0 \ 1]^T$ qui correspond à une rotation de 180° autour de Z . L'angle de rotation relatif est $\alpha_r = 180^\circ$, cependant le produit scalaire entre les deux quaternions est nul, ce qui est équivalent à deux vecteur orthogonaux entre eux.

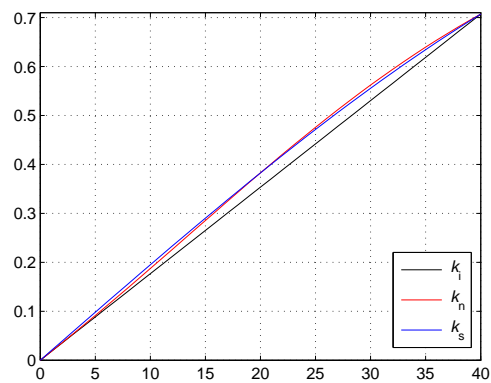
R2 : L'angle α_r est uniquement fonction de l'angle de rotation relatif entre les quaternions \mathbf{Q}_a et \mathbf{Q}_b . Ce qui permet l'analyse de l'erreur entre les quaternions dans le plan formé par ces quaternions \mathbf{Q}_a et \mathbf{Q}_b . Ce plan inclut la ligne droite du *nlerp* et l'arc du *slerp*.



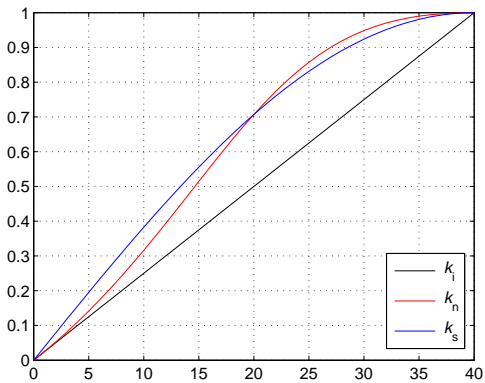
(a) n pour 90°



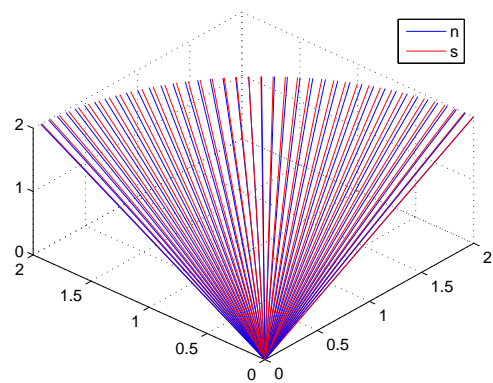
(b) n pour 180°



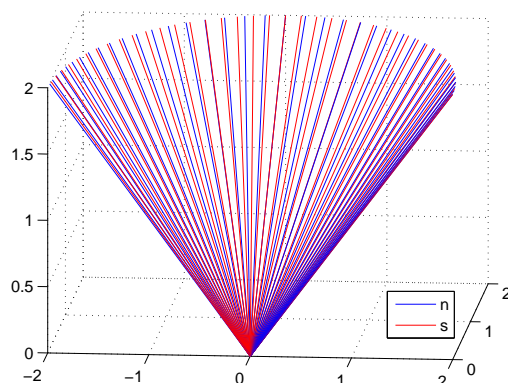
(c) k pour 90°



(d) k pour 180°



(e) Mouvement en 3D pour 90°



(f) Mouvement en 3D pour 180°

FIG. 3.17 – Rotation de 90° et Rotation de 180° autour de Z , pour comparer l'interpolation linéaire et le slerp. Les quatre premières images correspondent aux composants du quaternion n et k ou les sigles i correspond à interpolation, n à normalisé et s à slerp. Les deux dernières graphiques correspondent à la rotation dans l'espace 3D.

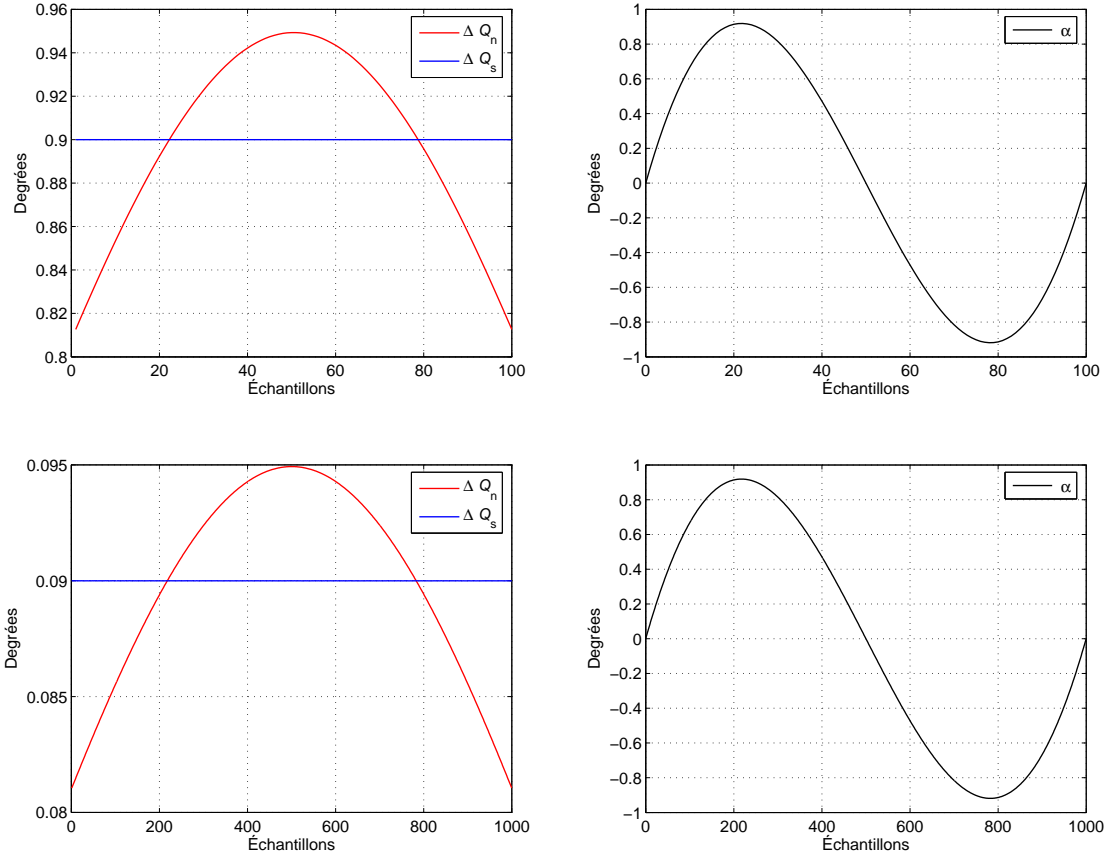


FIG. 3.18 – Variation entre interpolations successives et rotation relative pour une rotation de 90° pour le slerp et le nlerp.

Considérons la figure 3.19 qui présente le plan obtenu pour deux quaternions \mathbf{Q}_a et \mathbf{Q}_b . La figure 3.19.(a) considère 5 quaternions interpolées entre \mathbf{Q}_a et \mathbf{Q}_b . On observe une erreur d'interpolation nulle en $\mathbf{Q}_{s3} = \mathbf{Q}_{i3}$, qui correspond à $\alpha_Q/2$.

Pour calculer l'erreur d'interpolation entre le quaternion \mathbf{Q}_s obtenu par le slerp et le quaternion \mathbf{Q}_i obtenu par le nlerp correspondant, nous avons deux approches.

La première approche considère le calcul de la distance D_Q entre les quaternions \mathbf{Q}_a et \mathbf{Q}_b .

$$D_Q = \|\mathbf{Q}_a - \mathbf{Q}_b\|$$

suivi du calcul de la distance du centre de l'hyper-sphère O à la ligne droite.

$$D_L = \sqrt{1 - \left(\frac{D_Q}{2}\right)^2}$$

L'angle α_I est donné par

$$\alpha_I = \tan^{-1} \frac{\rho D_Q}{2D_L}$$

où ρ est le facteur de proportionnalité qui relie la distance $D_Q/2$ et $\alpha_Q/2$.

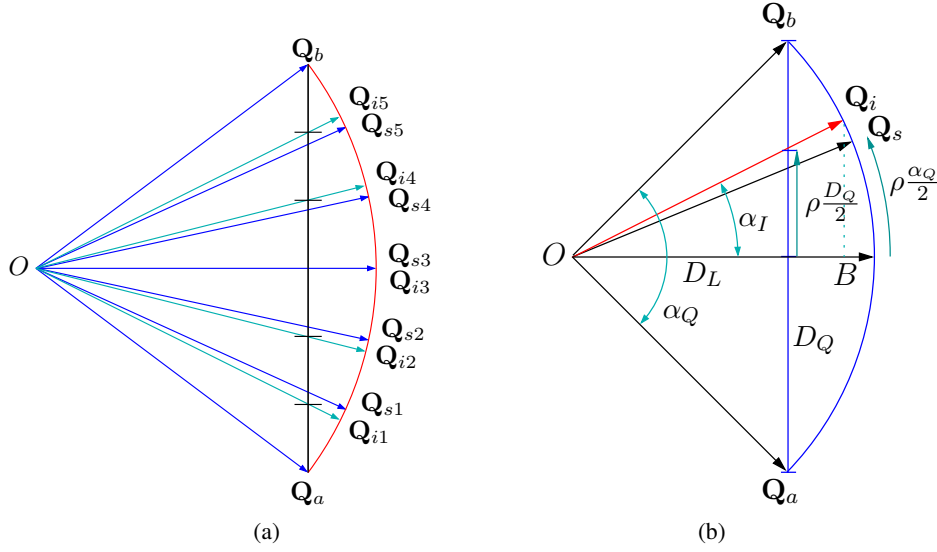


FIG. 3.19 – Erreur d’interpolation. L’image (a) montre l’erreur pour 5 quaternions interpolées entre Q_a et Q_b . L’image (b) montre les mesures pour calculer l’erreur maximale

La deuxième approche calcule le triangle formé par O, B et Q_I , tel que

$$\alpha_I = \tan^{-1} \frac{\rho \sin(\alpha_Q/2)}{\cos(\alpha_Q/2)} = \tan^{-1} \left(\rho \tan\left(\frac{\alpha_Q}{2}\right) \right)$$

L’erreur d’interpolation α_e dans l’espace de quaternions est défini par

$$\alpha_e = \rho \frac{\alpha_Q}{2} - \alpha_I \quad (3.63)$$

et l’erreur dans l’espace cartésien

$$\alpha_c = 2\alpha_e = \rho \frac{\alpha_r}{2} - 2 \tan^{-1} \left(\rho \tan\left(\frac{\alpha_r}{4}\right) \right) \quad (3.64)$$

où α_r est l’angle relatif entre les deux rotations.

Avec $\rho = 1/2$, on obtient l’erreur d’interpolation maximale. La figure 3.20 montre les erreurs d’interpolation pour différentes rotations.

Le schéma de commande avec retour de quaternion, présenté sur la figure 2.7 requiert des consignes de position P_d et d’orientation Q_d . La vitesse linéaire V_d est obtenue directement par la stratégie des mouvements souples. Cependant, la vitesse angulaire ω_d devra être calculée en utilisant

$$\begin{bmatrix} \omega_d \\ 0 \end{bmatrix} = -2Q_d^T \dot{Q}_d = 2\dot{Q}_d Q_d^{-1} \quad (3.65)$$

où Q_d est une représentation matricielle du quaternion, pour plus de détails se référer à l’annexe A.2.

3.8.2 Cas du manipulateur mobile

Dans la section précédente, la génération du mouvement pour le bras manipulateur a été présenté. Soit dans l’espace articulaire soit dans l’espace opérationnel, la stratégie de mouvement souple considère un espace n_1 dimensionnel. Selon le nombre m de configurations ou de situations, la trajectoire Φ est de taille $n_1 \times m$.

Le mouvement des plates-formes mobiles est traité dans deux cas différents, selon les conditions d’holonomie.

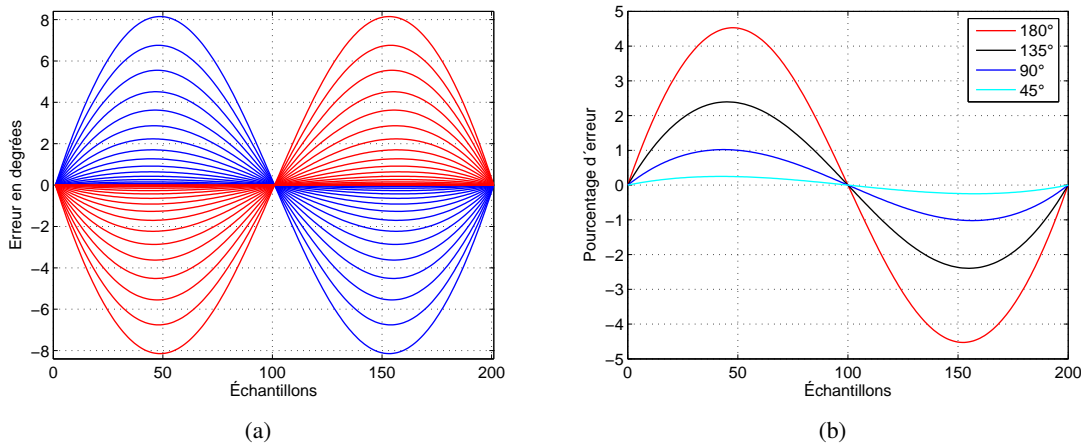


FIG. 3.20 – Erreur de Interpolation. L'image (a) montre l'erreur de interpolations pour des rotations successives de 180° a -180° avec un pas de 10° entre courbes. L'image (b) montre l'erreur en pourcentage pour des rotations de 180° , 135° , 90° et 45°

Plates-formes mobiles holonomes

Dans le cas particulier des manipulateurs sur plates-formes mobiles holonomes, les n_2 directions de la plate-forme sont ajoutés au vecteur de coordonnées du bras manipulateur et la stratégie de mouvements souples est appliquée au système augmenté. La trajectoire Φ devient de taille $(n_1 + n_2) \times m$.

Plates-formes mobiles non holonomes

Les plates-formes mobiles non holonomes présentent des propriétés différentes, la méthode du mouvement souples ne peut pas être appliquée directement car la plate-forme ne peut pas se déplacer dans la direction Y. Nous proposons deux approches pour aborder ce type de problème.

Le première approche considère un générateur de mouvement indépendant pour la plate-forme mobile [Laumond, 1998]. C'est la plate-forme qui détermine les temps de passage sur chacune des configurations ou situations. Donc chacune des lignes de la matrice Φ est liée à une composante T_p d'un vecteur de temps \mathbf{T}_{SP} qui regroupe les temps de passage. Dans la section précédente les temps T_{max1} et T_{max2} ont été calculés, lorsque la durée du mouvement est imposé par la plate-forme les temps deviennent

$$T_{max1} = T_p - T_{max2}$$

avec la condition

$$T_p \geq T_{max1} + T_{max2}$$

Si cette condition n'est pas satisfaite, le bras ne sera pas capable d'atteindre la situation ou configuration désirée sans dépasser les contraintes cinématiques.

La deuxième approche est celle de Lamiroux dans [Lamiroux et al., 1997], la vitesse linéaire $V(t)$ et la vitesse angulaire $\omega(t)$ de la plate-forme sont exprimées en fonction de $s(t)$. C'est le s qui devra être ajouté pour construire le système augmenté et pouvoir appliquer la méthode des mouvements souples.

Considérations sur les contraintes cinématiques

Dans le cas des manipulateurs mobiles le système est généralement redondant dans certaines directions. Les caractéristiques cinématiques de l'organe terminal dépendent du mouvement du bras manipulateur en plus du mouvement de la plate-forme. L'approche de génération des mouvements souples présentée au cours de ce chapitre peut être appliquée aux mouvements indépendants, la satisfaction des contraintes pour le mouvement total n'est pas encore traitée.

3.9 Conclusions

Nous avons présenté dans ce chapitre les mouvements souples pour les applications de robotique de service dans le but de satisfaire les contraintes de sécurité et de confort.

La méthode présentée permet des applications en ligne, pour des opérations dans l'espace opérationnel ou articulaire.

Nous avons montré que tout mouvement sous contraintes peut être réalisé au plus en sept segments cubiques sans perdre la continuité en C^2 .

Un logiciel d'application a été écrit pour la génération des mouvements souples pour un bras manipulateur à six degrés de liberté dans l'espace opérationnel.

Chapitre 4

Retour Visuel

4.1 Introduction

Qu'est-ce que le retour visuel ? Une définition simple est : l'utilisation de l'information provenant d'un capteur visuel, appareil photo ou caméra, dans la boucle de retour d'un système d'asservissement. Ce type de définition nous apporte plus de questions que de réponses. Par exemple, une des premières questions qu'on peut se poser est : quel type d'information visuelle utilise-t-on ? Ou plutôt, que doit on comprendre par information visuelle ? Ensuite, des questions par rapport aux capteurs visuels apparaissent : quelle est la différence entre un appareil photo et une caméra ? Est-il possible d'utiliser plusieurs capteurs visuels ? Finalement, on va aussi se poser des questions par rapport à la boucle de commande et à ses performances. Lorsque on parle de retour visuel, on est dans le domaine de la vision par ordinateur et de l'automatique, où la recherche est très active. En conséquence, si on essaie de répondre à toutes les questions que l'on peut se poser par rapport au sujet, on va atteindre un domaine où les réponses ne sont pas encore connus. Comme de plus on doit traiter le domaine de la robotique, le chemin serait long et dépasserait largement le but de ce chapitre.

Nous allons donc préciser la portée de notre travail. Nous avons besoin du retour visuel car notre objectif est de présenter une stratégie de commande pour les applications de la robotique de service. Le schéma hiérarchique présenté sur la figure 1.1 est une vision générale de notre approche. Dans ce contexte particulier, le bloc d'asservissement visuel fournit des entrées à deux blocs différents qui sont la génération de mouvement et l'asservissement de trajectoire. Mais revenons à la question initiale, pourquoi ? La réponse est simple, parce que nous voulons doter le robot de la capacité à agir de façon autonome et robuste dans des environnements dynamiques. Cette réponse, logique en robotique de service, n'est pas facile à mettre en oeuvre et demande des architectures logicielles plus complexes dont notre schéma hiérarchique de commande peut faire partie. Pour illustrer notre travail nous proposons une tâche de saisie d'un objet suivi de celle de l'apporter à un utilisateur humain en faisant l'hypothèse que tous les modèles sont connus. Supposons la stratégie suivante :

1. Le système de vision détermine la position de l'objet et la position de l'homme par rapport au robot.
2. Le système de planification de saisie choisit la position de saisi de l'objet et planifie le chemin à suivre.
3. Un système de génération de trajectoire calcule la loi de mouvement pour réaliser la tâche.
4. Une boucle d'asservissement réalise le suivi de cette trajectoire.
5. Le système de planification détermine le chemin pour apporter l'objet à l'homme.
6. Le système de génération de mouvement calcule cette nouvelle trajectoire.
7. Le boucle d'asservissement exécute cette trajectoire et donne l'objet à l'homme.

Selon la stratégie précédente, la vision est seulement utilisée dans la première étape. Par la suite, le robot exécute des mouvements en aveugle. Pour atteindre ces objectifs, plusieurs hypothèses sont nécessaires :

- i La scène est statique.
- ii Le modèle du bras manipulateur est parfait.
- iii Le modèle de l'objet est parfait.
- iv La calibration du système de vision est parfaite.

L'hypothèse i est « impossible » à satisfaire lorsque l'homme est en scène et le pas 7 de la stratégie ne peut être accompli que dans le cas où c'est l'homme qui collabore pour aller chercher l'objet, dans les autres cas c'est un système de vision qui doit guider le robot. Pour satisfaire les autres hypothèses (ii, iii et iv) le système devra être capable de modifier la trajectoire pour corriger les erreurs de modélisation et/ou de calibration en utilisant des informations visuelles, donc l'approche nécessite soit une déformation de la trajectoire pendant son exécution, soit une nouvelle étape de planification du chemin.

L'objectif de nos travaux est de fournir des schémas de commande capables de gérer l'information visuelle pour accomplir des tâches de robotique de service. Lorsque le problème est complexe, les schémas de commande que nous proposons visent à intégrer l'information visuelle dans les schémas en respectant les contraintes cinématiques de la robotique de service.

Concernant l'utilisation du retour visuel pour commander des bras manipulateurs, la littérature présente plusieurs configurations. La configuration que nous avons retenue utilise une paire stéréo embarquée sur l'organe terminal du bras manipulateur, les anglophones parlent de « hand-eye stéréo ».

La littérature présente une taxinomie croisée avec d'un côté la manière dont la commande des articulations est réalisée, soit avec un schéma qui commande directement les actionneurs de chaque joint (*Direct servo control*) soit avec un schéma qui génère des consignes pour les boucles de commande du robot (*Look and move*). De l'autre côté, la façon de gérer l'information visuelle, asservissement 3D, asservissement 2D et asservissement hybride.

Pour satisfaire les contraintes cinématiques nous utiliserons le schéma de commande présenté sur la figure 2.7. Nous avons montré que l'utilisation de ce type de schéma conduit à un erreur de position négligeable.

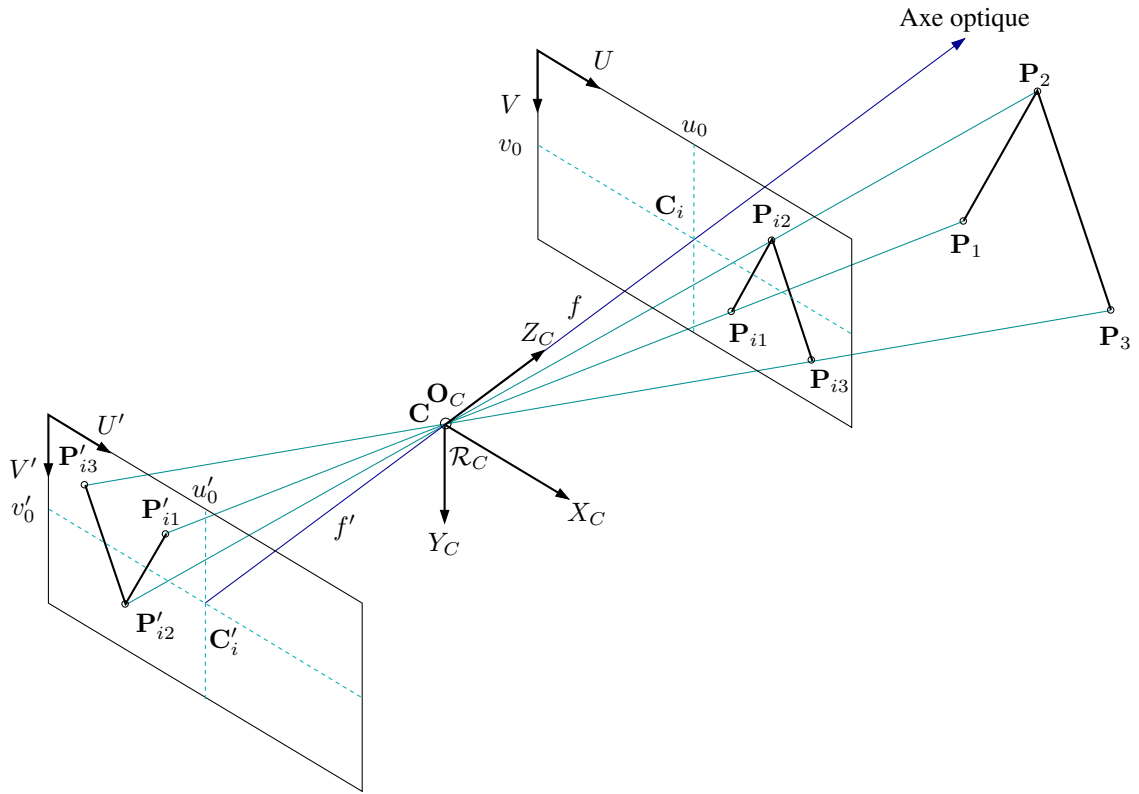
Ce chapitre est organisé en six parties : la première partie est dédiée au modèle du capteur visuel, suivi de la définition des différents repères. La troisième partie correspond à un schéma d'asservissement 2D et la quatrième à un schéma d'asservissement 3D. La cinquième partie propose un schéma d'asservissement 3D pour gérer les incertitudes du système. La dernière partie présente nos conclusions.

4.2 Modèle d'une caméra

La situation d'une caméra est définie par la position de son centre optique C et l'orientation de son axe optique. Sur la figure 4.1, le centre optique de la caméra C est lié à l'origine O_C du repère caméra \mathcal{R}_C et l'axe optique est confondu avec l'axe Z_C de ce repère. Physiquement, le plan image $U'V'$ est un plan perpendiculaire à l'axe optique localisé à une distance f' du centre optique, ou f' est la distance focale. Le point C'_i est appelé le *point principal* et correspond au point où l'axe optique traverse le plan image.

Le modèle *sténopé* ou *pin-hole* suppose que tous les rayons passent par le centre optique C . Considérons un objet défini dans le repère caméra par trois points P_1 , P_2 et P_3 . Leurs projections dans le plan image produit les points P'_{i1} , P'_{i2} et P'_{i3} . Ce qui produit une image inversée de l'objet.

Pour faciliter l'analyse des images, nous utilisons un plan image « virtuel » UV situé à une distance focale f en avant du centre optique C . La projection des points de l'objet sur ce plan virtuel produit les points P_{i1} , P_{i2} et P_{i3} qui génèrent une image équivalente ayant la propriété d'être dans le même sens que l'objet.

FIG. 4.1 – Modèle Sténopé d'une caméra, avec le plan image réel $U'V'$ et le plan image virtuel UV .

4.2.1 Matrice de projection perspective

Nous nous intéressons à la projection et aux propriétés géométriques du système. Le point P_1 est défini par les coordonnées

$$\mathbf{P}_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_C$$

Les coordonnées du point P_{i1} correspondent à la projection sur le plan image UV du point P_1 , elles s'expriment dans le repère caméra par :

$$\mathbf{P}_{i1} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_C$$

où x_i et y_i représentent les coordonnées dans le plan image :

$$x_i = f \frac{x}{z} \quad y_i = f \frac{y}{z} \quad z_i = f \quad (4.1)$$

En utilisant la représentation par des coordonnées homogènes, le système s'écrit :

$$s \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}_C = \begin{bmatrix} sx_i \\ sy_i \\ s \end{bmatrix}_C = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_C \quad (4.2)$$

où s représente la profondeur du point dans l'image.

La matrice de projection \mathcal{P} est alors définie par :

$$\mathcal{P} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.3)$$

Cette matrice peut être décomposée en un produit

$$\mathcal{P} = \mathcal{K}_P \mathcal{P}_0 \quad (4.4)$$

où

$$\mathcal{K}_P \triangleq \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathcal{P}_0 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.5)$$

4.2.2 Transformation affine caméra/image

Considérons la figure 4.2. Les points image sont mesurés en pixels (de l'anglais *picture element*) dans le repère UV associé au plan image. Le point principal C_i exprimé en coordonnées images est donné par (u_0, v_0) . Les pixels d'une caméra sont rarement carrés, soit λ_u la largeur et λ_v la hauteur d'un pixel.

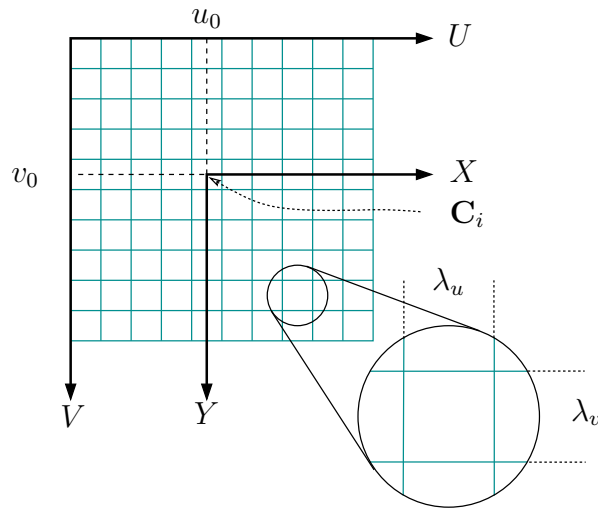


FIG. 4.2 – Image dans le repère UV.

Les coordonnées du point de l'image mesurées en pixel sont données par :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} K_u & 0 & u_0 \\ 0 & K_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (4.6)$$

où :

$$K_u = \frac{1}{\lambda_u} \quad K_v = \frac{1}{\lambda_v}$$

La matrice \mathcal{K}_A définit une transformation affine :

$$\mathcal{K}_A \triangleq \begin{bmatrix} K_u & 0 & u_0 \\ 0 & K_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

4.2.3 Paramètres intrinsèques

En faisant une projection perspective suivi d'une transformation affine, on abouti à :

$$u = K_u f \frac{x}{z} + u_0 \quad v = K_v f \frac{y}{z} + v_0 \quad (4.8)$$

en multipliant les matrices $\mathcal{K}_A \mathcal{K}_P \mathcal{P}_0$, on abouti à une représentation par matrice homogène :

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_C \quad (4.9)$$

où :

$$\alpha_u = fK_u \quad \alpha_v = fK_v$$

Ce modèle de la caméra sténopé ne comporte que des paramètres intrinsèques (α_u , α_v , u_0 et v_0) et suppose :

- un montage mécanique parfait, tel que la matrice CCD du capteur soit orthogonale à l'axe optique des lentilles et les axes X_C et Y_C soient alignées aux lignes et colonnes de la matrice CCD.
- les pixels ont une forme rectangulaire.
- les lentilles ne présentent pas de distorsion.

Pour estimer les paramètres intrinsèques plusieurs techniques sont proposées dans la littérature, pour plus de détails le lecteur peut se référer à [Horaud et al., 1993].

4.2.4 Paramètres extrinsèques

Lorsque on utilise la caméra comme capteur, les mesures que l'on peut obtenir sont exprimées dans le repère caméra. Pour avoir un modèle complet de la caméra, il faut prendre en compte un repère de référence, par exemple le repère monde \mathcal{R}_W . Une transformation qui exprime le changement de repère de la caméra au monde ${}^C\mathbf{T}_W$ définit les paramètres extrinsèques. En utilisant la transformation inverse ${}^W\mathbf{T}_C$, le modèle complet de la caméra devient :

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^W\mathbf{R}_C & {}^W\mathbf{T}_C \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_W \quad (4.10)$$

4.3 Graphe du système avec caméra embarquée

L'utilisation d'une caméra embarquée nécessite la définition de plusieurs repères pour exploiter pleinement l'information provenant du système de vision. Observons la figure 4.3 qui présente les repères qui permettent de modéliser le système.

Définissons les repères :

\mathcal{R}_W : le repère monde.

\mathcal{R}_B : repère lié à la base du bras manipulateur. Dans la suite, sauf indication contraire, le repère base sera considéré fixe et confondu avec le repère monde.

\mathcal{R}_P : repère lié au poignet du bras manipulateur.

\mathcal{R}_C : repère lié à la caméra, dans le cas d'une paire stéréo, ce repère est dédoublé en un repère \mathcal{R}_{CG} pour la caméra gauche et un repère \mathcal{R}_{CD} pour la caméra droite.

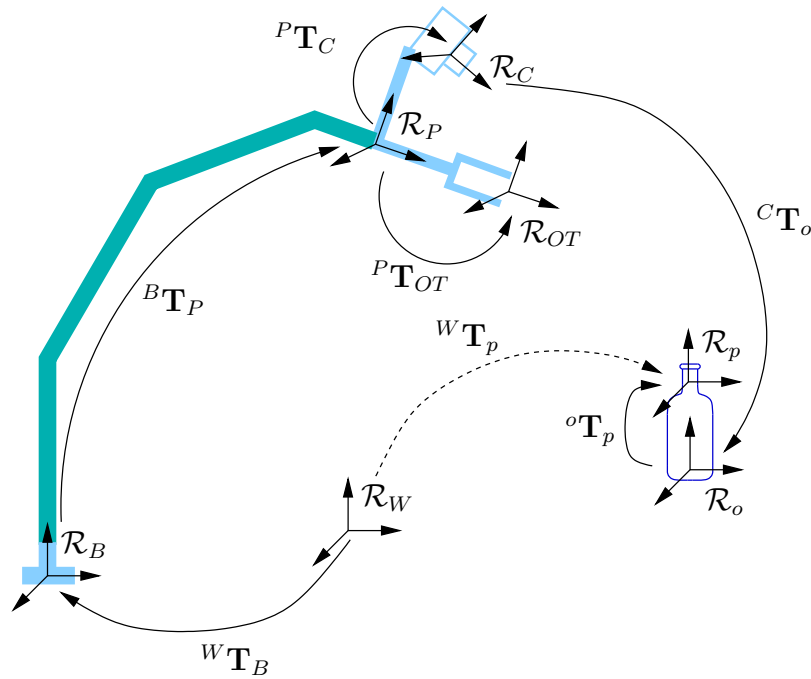


FIG. 4.3 – Graphe des repères du système de caméra embarquée et repères associés à une tâche de manipulation

\mathcal{R}_{OT} : repère lié à l'organe terminal (préhenseur ou outil), dans notre cas particulier il est aligné avec le repère \mathcal{R}_P .

\mathcal{R}_o : repère lié à l'origine de l'objet. Ce repère est défini par le modèle de l'objet.

\mathcal{R}_p : repère lié à la position de prise. Pour une tâche de saisie, il est défini par l'algorithme de planification de saisie.

Les différentes transformations qui interviennent :

${}^W\mathbf{T}_B$: transformation qui lie la base au monde. Lorsque le repère de base et le repère monde sont confondus, ${}^W\mathbf{T}_B = I_{4 \times 4}$.

${}^B\mathbf{T}_P$: transformation qui lie la base du robot à son poignet, elle est donnée par le modèle géométrique direct du robot.

${}^P\mathbf{T}_{OT}$: transformation qui définit la position de l'organe terminal. Voir l'annexe B pour plus de détails.

${}^P\mathbf{T}_C$: transformation correspondant aux paramètres extrinsèques de la caméra, elle lie la caméra au repère poignet. Lorsque on parle d'une configuration caméra embarquée, la détermination de cette transformation est appelée calibration hand-eye dans la littérature. Pour une paire stéréo, nous préférons parler de deux transformations indépendantes, une pour chaque caméra, ${}^P\mathbf{T}_{CG}$ pour la caméra gauche et ${}^P\mathbf{T}_{CD}$ pour la caméra droite.

${}^C\mathbf{T}_o$: transformation calculée par le système de vision, elle fournit la position de l'objet à partir des informations visuelles. Dans un espace structuré, la position de l'objet dans le monde est connue, dans ce cas ${}^C\mathbf{T}_o$ peut être calculée à partir de ${}^W\mathbf{T}_o$ et l'information visuelle peut être utilisée pour calibrer le système.

${}^o\mathbf{T}_p$: transformation qui lie le repère de l'objet et la position de prise.

Par exemple, pour un objet connu (modèle de l'objet connu) avec une position de prise définie *a priori*, le système de vision doit être capable de déterminer la position relative objet/caméra. Dans le cas de transformations parfaites, la position de prise dans le repère monde est donnée par la composition suivante :

$${}^W\mathbf{T}_p = {}^W\mathbf{T}_B {}^B\mathbf{T}_P {}^P\mathbf{T}_C {}^C\mathbf{T}_o {}^o\mathbf{T}_p \quad (4.11)$$

Un planificateur de mouvement devra produire le chemin que le robot devra réaliser pour déplacer l'organe terminal depuis sa situation courante vers la situation de prise. Le calcul du chemin est hors de la portée de ce travail. Les travaux de [Lopez-Damian et al., 2005] présentent une solution à ce problème du calcul de chemin. Nous utilisons les chemins produits par les algorithmes de López-Damian comme entrée de notre système.

4.4 Asservissement Visuel 2D partagé

4.4.1 Introduction

L'asservissement visuel 2D correspond à l'utilisation des informations visuelles directement par la boucle de commande. Par exemple, dans [Chaumette et al., 1991], le positionnement de la caméra par rapport à la cible est réalisé en utilisant le formalisme de la fonction de tâche [Samson et al., 1990]. Ce type d'approche est considéré comme la commande référencée capteur, le lecteur intéressé par le sujet peut se référer aux travaux de [Bellot, 2002] pour plus de détails. Notre approche ne suit pas cette voie. Une description simple de notre approche est la suivante : un processus de planification établit la trajectoire à suivre et une caméra fournit des informations visuelles pour corriger les imprécisions dans la modélisation de la tâche.

4.4.2 Schéma de commande

Dans [Nelson et al., 1996], trois schémas de commande sont présentés pour le couplage vision - effort.

- **Commande par Compromis (Traded Control)**, dans ce type de schéma la commande est réalisée alternativement par la boucle de vision et par la boucle d'effort, selon les conditions opératoires.
- **Commande Hybride**, dans ce type de schéma chacune des directions est commandée de façon séparée, soit par le boucle d'effort soit par le boucle de vision.
- **Commande partagée**, dans ce type de schéma la commande dans chacune des directions est réalisée simultanément par les boucles de vision et d'effort.

Nous proposons le schéma de commande de la figure 4.4.

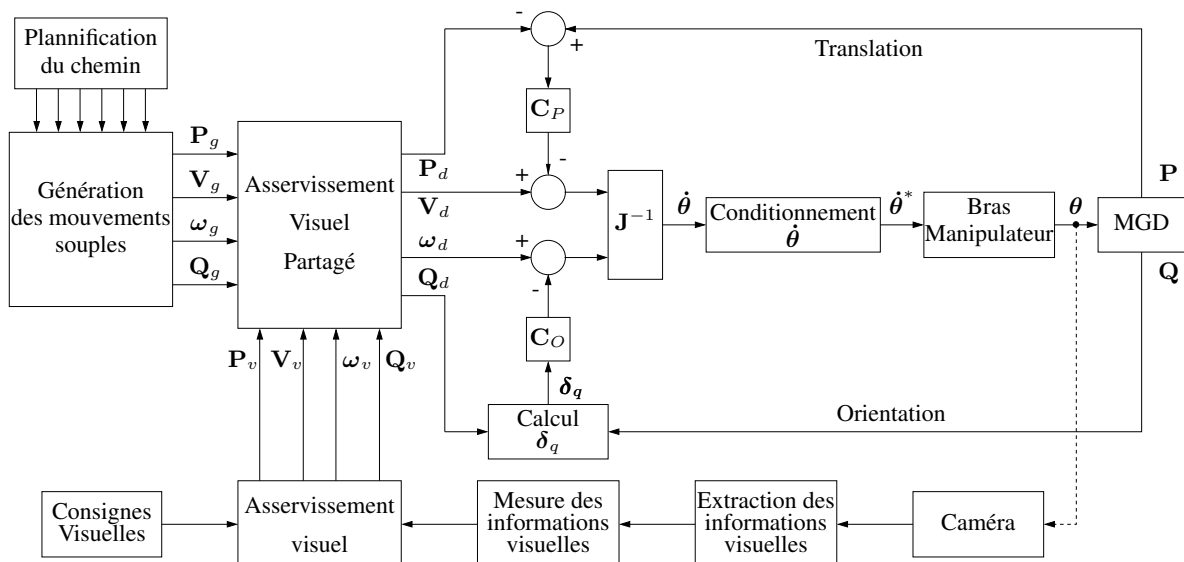


FIG. 4.4 – Schéma de commande pour le bras manipulateur en utilisant un schéma d'asservissement visuel 2D

Comme la tâche est planifiée *à priori*, l'objectif de la boucle de commande est de positionner et d'aligner l'organe terminal en fonction des indices visuels. Dans [Hager et al., 1996], il est établi que la tâche peut être réalisée en utilisant des lois de commande classiques.

4.4.3 Loi de commande partagée

La boucle de commande interne constituée par un schéma de commande à résolution de vitesse par retour de quaternion pour l'orientation utilise comme signaux de consigne la position \mathbf{P} , la vitesse linéaire \mathbf{V} , l'orientation \mathbf{Q} et la vitesse angulaire $\boldsymbol{\omega}$ de l'organe terminal du bras manipulateur :

$$\begin{aligned}\mathbf{P}_d &= \lambda \mathbf{P}_g + (1 - \lambda) \mathbf{P}_v \\ \mathbf{V}_d &= \lambda \mathbf{V}_g + (1 - \lambda) \mathbf{V}_v \\ \boldsymbol{\omega}_d &= \lambda \boldsymbol{\omega}_g + (1 - \lambda) \boldsymbol{\omega}_v \\ \mathbf{Q}_d &= \lambda \mathbf{Q}_g + (1 - \lambda) \mathbf{Q}_v\end{aligned}\tag{4.12}$$

où les indices d correspondent aux consignes de la boucle de commande interne, les indices g aux consignes calculées par le générateur de mouvements souples et les indices v aux lois de commande par asservissement visuel. Le coefficient λ est utilisé comme paramètre d'ajustement, couramment $\lambda = 1/2$.

Le formalisme de la fonction de tâche définit la fonction suivante pour une tâche de positionnement :

$$\mathbf{e}(\mathbf{X}(t)) = M(\xi(\mathbf{X}(t)) - \xi^*)\tag{4.13}$$

où $\mathbf{X}(t)$ représente la situation de la caméra par rapport au monde, ξ^* désigne l'ensemble des indices visuels de consigne, $\xi(\mathbf{X}(t))$ désigne la valeurs des indices visuels pour la situation courante et la matrice M transforme les composantes du vecteur $\xi(\mathbf{X}(t)) - \xi^*$ pour obtenir $\mathbf{e}(\mathbf{X}(t))$.

En posant le problème de commande comme un problème de régulation de la sortie, la tâche est réalisée quand $\mathbf{e}(\mathbf{X}(t)) = \mathbf{0}$. Nous envisageons deux types de solutions : une solution statique qui considère uniquement des informations géométriques obtenues directement de l'image et une solution dynamique qui considère le flux optique.

Solution statique

Cette approche 2D ne considère pas la correction de l'information 3D de la scène. La tâche consiste uniquement au positionnement par rapport aux coordonnées XY et à l'alignement en rotations autour de l'axe Z du repère caméra. L'erreur est fonction de la tâche et peut être mesurée directement dans l'image. Cette solution calcule \mathbf{P}_v et \mathbf{Q}_v et considère par définition $\mathbf{V}_v = \mathbf{0}$ et $\boldsymbol{\omega}_v = \mathbf{0}$.

Considérons les exemples suivants :

- pour une tâche de pointage, la mesure d'un seul point est nécessaire. L'erreur de position de la cible peut être mesurée par rapport à un point quelconque du plan image, par exemple, la projection du centre optique pour centrer la cible dans le plan image.
- le deuxième exemple correspond à l'orientation de la caméra par rapport à une cible. Le système requiert la mesure d'au moins deux points, tel qu'une rotation autour de l'axe optique soit capable de réaliser l'alignement entre cet axe et un axe quelconque défini dans le plan image.

Pour nos applications, les points de l'objet qui correspondent aux indices visuels peuvent être projetés dans l'image pour obtenir l'erreur dans le plan image. Les indices visuels définis par un point image \mathbf{P}_i peuvent être transformés en points dans le repère caméra \mathcal{R}_C par :

$$\mathbf{P}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} \quad \mathbf{P}_{iC} = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix}$$

et en utilisant l'équation 4.8, on obtient

$$x_c = z_C \frac{u_i - u_0}{\alpha_u} \quad y_c = z_C \frac{v_i - v_0}{\alpha_v}$$

où α_u, α_v, u_0 et v_0 sont les paramètres de la caméra.

L'erreur image $\mathbf{e}_i = \xi(\mathbf{X}(t)) - \xi^*$ devra être reconstruite dans \mathcal{R}_C en utilisant les équations précédentes. L'erreur calculée par la reconstruction des points est $\mathbf{e}_C = \mathbf{X}_{iC} - \mathbf{X}^*$. Dans les deux cas, l'erreur doit être exprimée dans le repère de base \mathcal{R}_B .

La loi de commande utilisée pour calculer \mathbf{P}_v et \mathbf{Q}_v est définie par l'équation 2.7 où les limites de chaque loi de commande correspondent aux contraintes d'accélération et de vitesse de chacune des composantes.

Solution dynamique

La solution est de la forme :

$$\begin{bmatrix} \mathbf{V}_v \\ \boldsymbol{\omega}_v \end{bmatrix} = -\lambda \mathbf{J}_i^+ (\xi(\mathbf{X}(t)) - \xi^*) \quad (4.14)$$

où \mathbf{J}_i correspond à la matrice jacobienne de l'image appelé aussi *matrice d'interaction* avec \mathbf{J}_i^+ défini comme sa pseudo-inverse.

Pour calculer la matrice d'interaction, de l'équation 4.8 pour un point défini dans le repère caméra \mathbf{P}_C et un point image \mathbf{P}_i ,

$$\mathbf{P}_C = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} \quad \mathbf{P}_i = \begin{bmatrix} u \\ v \end{bmatrix}$$

nous avons

$$u_s = u - u_0 = \alpha_u \frac{x_C}{z_C} \quad v_s = v - v_0 = \alpha_v \frac{y_C}{z_C} \quad (4.15)$$

Le mouvement dans le plan image est défini par

$$\dot{u}_s = \alpha_u \frac{\dot{x}_C}{z_C} - u_s \frac{\dot{z}_C}{z_C} \quad \dot{v}_s = \alpha_v \frac{\dot{y}_C}{z_C} - v_s \frac{\dot{z}_C}{z_C} \quad (4.16)$$

Lorsque l'objet est fixe et la caméra se déplace à une vitesse linéaire \mathbf{V}_C et angulaire $\boldsymbol{\omega}_C$, le mouvement du point dans \mathcal{R}_C est donné par

$$\dot{\mathbf{P}}_C = -\mathbf{V}_C - \boldsymbol{\omega}_C \times \mathbf{P}_C \quad (4.17)$$

avec :

$$\begin{aligned} \dot{x}_C &= -V_x - z_C \omega_y + y_C \omega_z \\ \dot{y}_C &= -V_y + z_C \omega_x - x_C \omega_z \\ \dot{z}_C &= -V_z - y_C \omega_x + x_C \omega_y \end{aligned} \quad (4.18)$$

En combinant les équations 4.16 et 4.18 :

$$\begin{bmatrix} \dot{u}_s \\ \dot{v}_s \end{bmatrix} = \mathbf{J}_i \begin{bmatrix} \mathbf{V}_C \\ \boldsymbol{\omega}_C \end{bmatrix} \quad (4.19)$$

La matrice d'interaction pour chacun des m indices visuels avec $j = 1 \dots m$ est :

$$\mathbf{J}_{ij} = \begin{bmatrix} -\frac{\alpha_u}{z_C} & 0 & \frac{u_s}{z_C} & \frac{u_s y_C}{z_C} & -\alpha_u - \frac{u_s x_C}{z_C} & \frac{\alpha_u y_C}{z_C} \\ 0 & -\frac{\alpha_v}{z_C} & \frac{v_s}{z_C} & \alpha_v + \frac{v_s y_C}{z_C} & -\frac{v_s x_C}{z_C} & -\frac{\alpha_v x_C}{z_C} \end{bmatrix} \quad (4.20)$$

Pour pouvoir inverser la matrice jacobienne sans perdre de degrés de liberté, on a besoin d'au moins trois indices visuels. L'utilisation de plusieurs indices visuels augmente la robustesse de la méthode. La jacobienne devient alors :

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{J}_{i1} \\ \mathbf{J}_{i2} \\ \vdots \\ \mathbf{J}_{im} \end{bmatrix} \quad (4.21)$$

Le torseur cinématique est exprimé dans le repère caméra \mathcal{R}_C , il doit être transporté dans le repère de base \mathcal{R}_B pour pouvoir être utilisé dans la boucle de commande.

$$\begin{bmatrix} \mathbf{V}_B \\ \boldsymbol{\omega}_B \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{P}^\times \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{V}_C \\ \boldsymbol{\omega}_C \end{bmatrix} \quad (4.22)$$

La solution statique permet d'utiliser des consignes de vitesse nulles ($\mathbf{V}_v = 0$ et $\boldsymbol{\omega}_v = 0$). Pour la solution dynamique il faut calculer les consignes de position.

4.4.4 Avantages et inconvénients

L'avantage principal de ce type de méthodes est la robustesse vis à vis des imprécisions des paramètres du modèle de la caméra, lorsqu'ils agissent comme des gains par rapport à l'erreur visuelle.

La méthode statique requiert la connaissance ou une estimation de la profondeur de la scène alors que la méthode dynamique requiert le calcul de la matrice d'interaction. Ces deux contraintes peuvent être satisfaites en utilisant l'information provenant du modèle de la scène. Cependant, le principal inconvénient, pour les applications que nous envisageons, est qu'elle nécessite le calcul des consignes visuelles qui devront correspondre aux scènes perçues par le caméra pendant l'exécution de la trajectoire, ce qui peut demander beaucoup de temps de calcul.

4.5 Asservissement Visuel 3D

4.5.1 Introduction

L'asservissement visuel 3D requiert la reconstruction 3D de la scène. L'avantage de ce type de schémas provient de la possibilité de programmer la trajectoire directement dans l'espace cartésien de la tâche.

Hager, pour résoudre le problème de commande de systèmes par vision 3D propose la décomposition de la tâche en « capacités primitives » que le robot doit être capable d'exécuter, telles que le positionnement ou alignement [Hager, 1997]. Les travaux de Feddema portent sur la génération des trajectoires dans le plan image par une approche 2D [Feddema et al., 1989] et Mezouar intègre des contraintes mécaniques dans la génération des trajectoires pour une approche 2D 1/2 [Mezouar et al., 2001].

Notre approche considère la reconstruction 3D à l'aide d'une paire stéréo montée sur l'organe terminal du bras manipulateur. Pour un objet qui se déplace, chaque fois qu'une nouvelle position de la cible a été calculée, cette position est utilisée comme consigne pour la génération de trajectoires en utilisant le générateur de mouvements souples.

4.5.2 Reconstruction 3D stéréo

Reprenons le modèle sténopé d'une caméra, éq.4.9 et considérons que les points sont définis dans le repère caméra \mathcal{R}_C . À partir d'un point défini dans l'espace 3D nous calculons son point image correspondant en 2D. Lors de l'acquisition d'une photo de la scène, l'information sur la profondeur est perdue. Donc, une caméra est un capteur de mesures angulaires, tel que chaque point image de coordonnées image (u,v) par rapport au centre optique correspond à la projection de tous les points d'un « rayon » infini. En ajoutant des informations qui apportent la connaissance de la scène, certaines informations 3D peuvent être reconstruites. Par exemple, la connaissance *à priori* des dimensions d'un objet peut fournir des informations sur la taille et la position des objets voisins.

Pour pouvoir obtenir des informations 3D, nous avons besoin d'au moins deux images de la même scène depuis deux situations différentes. Les méthodes d'asservissement 2D $\frac{1}{2}$ proposent la reconstruction 3D à partir de la connaissance du mouvement réalisé par la caméra. La solution la plus simple utilise une paire stéréo pour reconstruire la scène par triangulation. La littérature propose aussi des méthodes plus robustes qui considèrent la géométrie épipolaire [Luong et al., 1996]. Par la suite, nous considérons

seulement la reconstruction 3D par triangulation, qui demande la connaissance des paramètres intrinsèques des caméras qui constituent la paire stéréo.

Considérons la configuration stéréo hand-eye, avec les transformations ${}^P\mathbf{T}_{CD}$, ${}^P\mathbf{T}_{CG}$ et des paramètres intrinsèques des deux caméras connus. Notre objectif est de trouver les coordonnées du point \mathbf{P}_P dans le repère poignet \mathcal{R}_P . La figure 4.5 présente le schéma utilisé pour la reconstruction 3D.

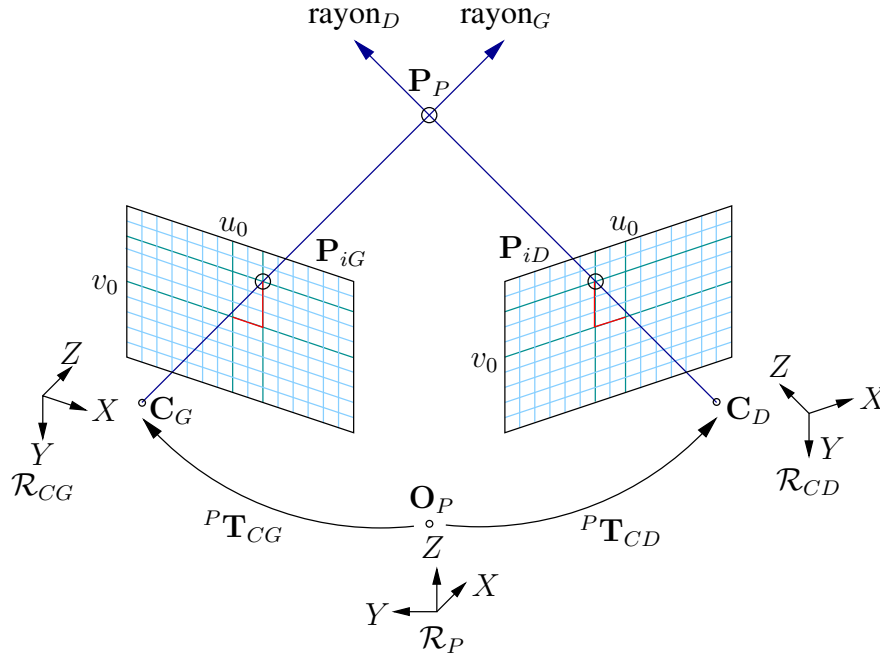


FIG. 4.5 – Reconstruction 3D par information stéréo

Soient les coordonnées image du point :

$$\mathbf{P}_{iD} = \begin{bmatrix} u_D \\ v_D \end{bmatrix} \quad \mathbf{P}_{iG} = \begin{bmatrix} u_G \\ v_G \end{bmatrix}$$

et $(\alpha_{uD}, \alpha_{vD}, u_{0D}, v_{0D})$ les paramètres intrinsèques de la caméra droite et $(\alpha_{uG}, \alpha_{vG}, u_{0G}, v_{0G})$ ceux de la caméra gauche.

Les rayons droit et gauche sont donnés, dans leurs repères respectifs par les équations :

$$\text{rayon}_D = \begin{bmatrix} \frac{u_D - u_{0D}}{\alpha_{uD}} \\ \frac{v_D - v_{0D}}{\alpha_{vD}} \\ 1 \end{bmatrix} s_D \quad \text{rayon}_G = \begin{bmatrix} \frac{u_G - u_{0G}}{\alpha_{uG}} \\ \frac{v_G - v_{0G}}{\alpha_{vG}} \\ 1 \end{bmatrix} s_G \quad (4.23)$$

où s_D est la profondeur du point dans l'image droite et s_G est la profondeur dans l'image gauche.

Étant donné que les transformations ${}^P\mathbf{T}_{CD}$ et ${}^P\mathbf{T}_{CG}$ sont connues, leur inverse est facilement calculée par :

$${}^{CD}\mathbf{T}_P = \begin{bmatrix} {}^{CD}\mathbf{R}_P & {}^{CD}\mathbf{T}_P \\ 0 & 1 \end{bmatrix} \quad {}^{CG}\mathbf{T}_P = \begin{bmatrix} {}^{CG}\mathbf{R}_P & {}^{CG}\mathbf{T}_P \\ 0 & 1 \end{bmatrix}$$

Les deux rayons exprimés dans le repère du poignet deviennent :

$$\text{rayon}_D = {}^{CD}\mathbf{R}_P \begin{bmatrix} \frac{u_D - u_{0D}}{\alpha_{uD}} \\ \frac{v_D - v_{0D}}{\alpha_{vD}} \\ 1 \end{bmatrix} s_D + {}^{CD}\mathbf{T}_P \quad (4.24)$$

$$\text{rayon}_G = {}^{CG}\mathbf{R}_P \begin{bmatrix} \frac{u_G - u_{0G}}{\alpha_{uG}} \\ \frac{v_G - v_{0G}}{\alpha_{vG}} \\ 1 \end{bmatrix} s_G + {}^{CG}\mathbf{T}_P \quad (4.25)$$

Le point \mathbf{P}_P se localise à l'intersection des deux rayons exprimés dans le repère poignet. Nous obtenons alors l'équation suivante :

$${}^{CD}\mathbf{R}_P \begin{bmatrix} \frac{u_D - u_{0D}}{\alpha_{uD}} \\ \frac{v_D - v_{0D}}{\alpha_{vD}} \\ 1 \end{bmatrix} s_D + {}^{CD}\mathbf{T}_P = {}^{CG}\mathbf{R}_P \begin{bmatrix} \frac{u_G - u_{0G}}{\alpha_{uG}} \\ \frac{v_G - v_{0G}}{\alpha_{vG}} \\ 1 \end{bmatrix} s_G + {}^{CG}\mathbf{T}_P \quad (4.26)$$

Que nous pouvons écrire :

$$\begin{bmatrix} r_{xD} \\ r_{yD} \\ r_{zD} \end{bmatrix} s_D + \begin{bmatrix} t_{xD} \\ t_{yD} \\ t_{zD} \end{bmatrix} = \begin{bmatrix} r_{xG} \\ r_{yG} \\ r_{zG} \end{bmatrix} s_G + \begin{bmatrix} t_{xG} \\ t_{yG} \\ t_{zG} \end{bmatrix} \quad (4.27)$$

ou sous forme matricielle :

$$\begin{bmatrix} r_{xD} & -r_{xG} \\ r_{yD} & -r_{yG} \\ r_{zD} & -r_{zG} \end{bmatrix} \begin{bmatrix} s_D \\ s_G \end{bmatrix} = \begin{bmatrix} t_{xG} - t_{xD} \\ t_{yG} - t_{yD} \\ t_{zG} - t_{zD} \end{bmatrix} \quad (4.28)$$

C'est un système de la forme $\mathbf{Ax} = \mathbf{b}$. Une solution pour ce type de système est donnée par :

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

Les coordonnées du point \mathbf{P}_P sont ensuite obtenues par substitution de s_D dans l'équation 4.24.

4.5.3 Schéma de commande

Le schéma de commande utilise la position de la cible fournie par le système de vision 3D. Nous proposons l'utilisation d'un planificateur de chemin et du générateur de mouvements souples dans la boucle de commande. La boucle de commande interne est le schéma de commande à résolution de vitesse par retour de quaternion.

Deux stratégies en utilisant ce schéma sont présentées : le schéma avec planificateur de trajectoire et le schéma avec guidage visuel. Le schéma de commande proposé est présenté sur la figure 4.6.

Schéma complet avec planificateur

Cette approche s'appuie sur l'utilisation d'un planificateur pour définir le chemin que le robot doit suivre pour atteindre la situation finale. Cette stratégie est intéressante lorsque la situation courante du robot et la situation finale sont éloignées ou lorsque il y a des obstacles entre la situation courante et la situation finale.

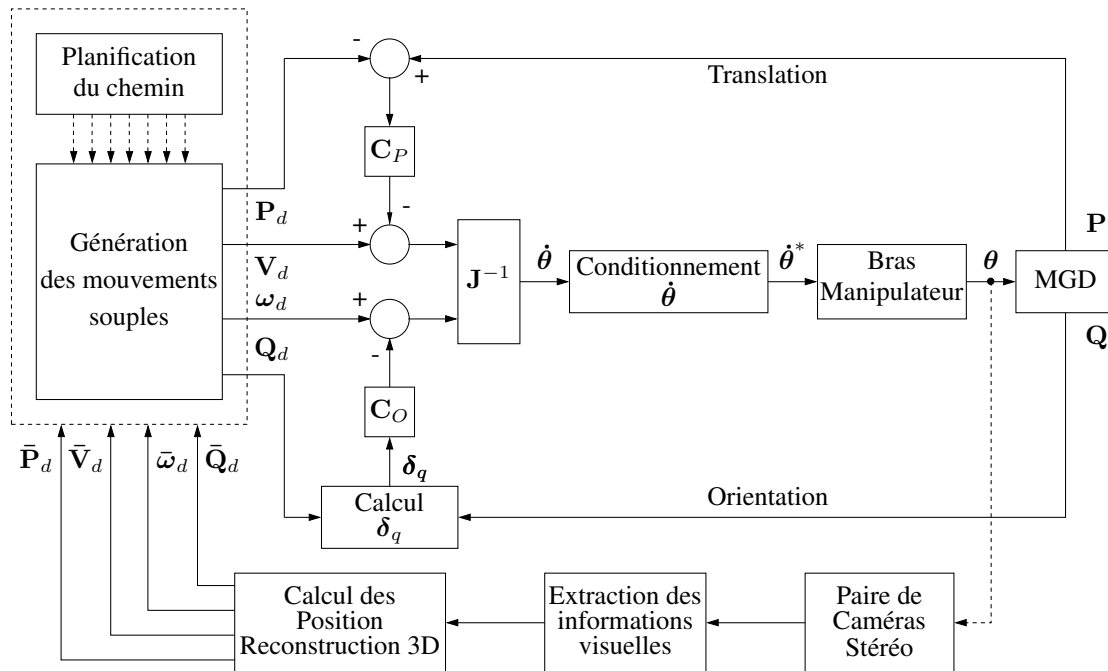


FIG. 4.6 – Schéma de commande pour le bras manipulateur utilisant un schéma d'asservissement visuel 3D

Schéma de guidage visuel

Cette stratégie considère que la situation suivante est déterminée par le système de vision, le générateur de mouvements souples utilise la nouvelle situation comme consigne. Le planificateur de chemin n'est plus nécessaire. Cette stratégie est intéressante pour des tâches de suivi ou quand la situation finale est proche, nous considérons alors que l'espace entre les deux situations est libre d'obstacles.

4.5.4 Avantages et inconvénients

L'avantage principal de notre schéma est que la commande s'exprime dans le repère de la tâche et non dans le repère image et que, par conséquent, les contraintes cinématiques peuvent être respectées.

En considérant une tâche de guidage visuel, l'utilisation du générateur de mouvements souples permet le passage d'un segment de trajectoire à un autre de façon lisse, sans sauts dans les transitions qui pourraient réduire le confort des utilisateurs.

Il demeure un inconvénient lié à la forte dépendance par rapport aux modèles des méthodes 3D que nous allons essayer de diminuer dans le paragraphe suivant.

4.6 Localisation d'un objet et auto-calibration main/caméra ou « hand-eye »

4.6.1 Introduction

La forte dépendance aux modèles doit être compensée d'une certaine façon. Alors que corriger un modèle complet de l'environnement est une tâche lourde en temps d'exécution, nous cherchons à corriger seulement les modèles qui lient le capteur au robot et les informations sur la position de la cible. Ce qui conduit à un processus de calibration main/caméra ou « Hand-eye » et à un processus de localisation.

Pour des systèmes où les paramètres intrinsèques de la caméra peuvent être connus en faisant au moins trois mouvements, la littérature montre que la calibration hand-eye est équivalente à résoudre

une équation de la forme $\mathbf{AX} = \mathbf{XB}$, où \mathbf{X} est la transformation inconnu qui lie la caméra à la main du robot, \mathbf{A} est une matrice qui est calculée en fonction du mouvement de la main (mouvement du robot) et \mathbf{B} est une matrice qui est calculée par calibration extrinsèque de la caméra [Tsai et al., 1989], [Shiu et al., 1989].

Par ailleurs, Westmore propose l'utilisation d'un filtre de Kalman étendu (connu par les sigles EKF en anglais) pour déterminer la position 3D de la cible caractérisée par au moins trois points en utilisant un système de vision mono-caméra [Westmore et al., 1991].

Les travaux de [Wei et al., 1998] et [Andreff et al., 1999] portent sur l'auto-calibration hand-eye en considérant les effets des rotations et translations sur les amers dans l'image. Wei propose un processus de minimisation non linéaire pendant que Andreff propose une analyse algébrique inspirée de l'équation matricielle de Sylvester.

Dans [Andreff, 1997] à l'intérieur d'une boucle d'asservissement visuel, un filtre de Kalman est utilisé pour estimer *en ligne* la calibration hand-eye.

Depuis 1986 la littérature présente des techniques de SLAM (de l'anglais *Simultaneous localisation and Mapping*, cartographie et localisation simultanée) [Durrant-White et al., 2006]. Foxlin a présenté une architecture « décentralisée » pour le SLAM, elle considère aussi l'auto-calibration [Foxlin, 2002]. Solà considère l'utilisation des caméras comme des capteurs angulaires et présente une approche bi-caméra qui considère l'auto-calibration de la direction d'une caméra par rapport à l'autre [Solà et al., 2007].

En utilisant les développements présentés par Solà, nous proposons une stratégie « simplifiée » du SLAM pour une tâche de saisie d'un objet. Notre objectif est de déterminer la localisation de la cible et les paramètres de calibration hand-eye dans un processus d'auto-calibration.

4.6.2 Schéma de commande

Le schéma de commande de la figure 4.7 correspond à l'utilisation d'un système bi-caméra pour déterminer la situation suivante que l'organe terminal doit atteindre pour la saisie d'un objet.

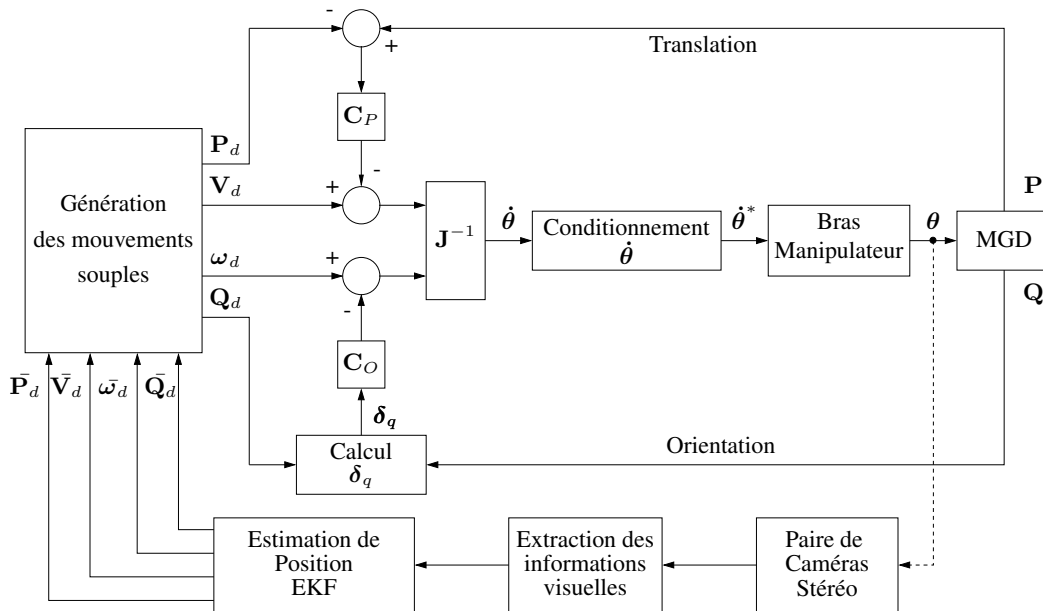


FIG. 4.7 – Schéma de commande pour le bras manipulateur en utilisant un filtre de Kalman étendu (EKF) pour estimer la position de la cible

Nous proposons une stratégie du type *Look and Move* pour supprimer le bruit produit par le mouvement dans la position du robot et dans l'image. Notre stratégie est définie en 5 étapes :

1. Acquisition des images.
2. Obtention des indices visuels à partir des images.
3. Application du filtre de Kalman étendu.
4. Définition de la situation suivante.
5. Déplacement vers la nouvelle situation.

Définissons m situations successives entre la situation initiale \mathbf{X}_i correspondant à la situation courante au moment d'initier le processus et la situation finale \mathbf{X}_f définie en fonction de la cible. Pour chaque situation le système réalise les 5 étapes de la stratégie précédente. Nous faisons les hypothèses que l'espace opérationnel entre les deux situations est libre d'obstacles et que la situation finale est dans cet espace.

L'étape 1 ne requiert plus d'explication, l'étape 2 est fonction des capacités du système de vision par ordinateur et l'étape 5 est réalisé par un mouvement point à point. Dans les paragraphes suivants nous allons décrire les étapes 3 et 4.

Filtre de Kalman étendu

Les équations du filtre de Kalman étendu sont largement traitées dans la littérature. En définissant les fonctions non linéaires d'observation \mathcal{F} et de mesure \mathcal{H} pour un système non linéaire :

$$\mathbf{X}_{k+1} = \mathcal{F}(\mathbf{X}_k, \mathbf{w}_k) \quad \mathbf{Y}_k = \mathcal{H}(\mathbf{X}_k) + \mathbf{v}_k \quad (4.29)$$

où \mathbf{X} est le vecteur d'état et \mathbf{Y} le vecteur de sortie ou des variables mesurées. \mathbf{w}_k représente les entrées du système (loi de commande et perturbations), et \mathbf{v}_k est le bruit de mesure. En considérant des variables gaussiennes définies par une moyenne et une variance :

$$\mathbf{w}_k \sim \mathcal{N}(\bar{\mathbf{w}}, \mathbf{Q}) \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$$

L'état est aussi gaussien avec une moyenne $\bar{\mathbf{X}}$ et une covariance \mathbf{P} .

La fonction d'observation est linéarisé autour de l'état estimé $\hat{\mathbf{X}}$ et de la loi de commande $\bar{\mathbf{w}}$ tel que on obtient les matrices jacobiniennes :

$$\mathbf{F}_X = \left. \frac{\partial \mathcal{F}}{\partial \mathbf{X}^T} \right|_{\hat{\mathbf{X}}, \bar{\mathbf{w}}} \quad \mathbf{F}_w = \left. \frac{\partial \mathcal{F}}{\partial \mathbf{w}^T} \right|_{\hat{\mathbf{X}}, \bar{\mathbf{w}}} \quad (4.30)$$

L'étape de prédiction est donnée par :

$$\begin{aligned} \hat{\mathbf{X}}^+ &= \mathcal{F}(\hat{\mathbf{X}}, 0) \\ \mathbf{P}^+ &= \mathbf{F}_X \mathbf{P} \mathbf{F}_X^T + \mathbf{F}_w \mathbf{Q} \mathbf{F}_w^T \end{aligned} \quad (4.31)$$

La fonction de mesure est linéarisée autour de l'état estimé, tel que sa matrice jacobienne soit :

$$\mathbf{H} = \left. \frac{\partial \mathcal{H}}{\partial \mathbf{X}^T} \right|_{\hat{\mathbf{X}}} \quad (4.32)$$

L'étape de correction est donnée par :

$$\begin{aligned} \mathbf{Z} &= \mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R} \\ \mathbf{K} &= \mathbf{P} \mathbf{H}^T \mathbf{Z}^{-1} \\ \hat{\mathbf{X}}^+ &= \hat{\mathbf{X}} + \mathbf{K} (\mathbf{Y} - \mathcal{H}(\hat{\mathbf{X}})) \\ \mathbf{P}^+ &= \mathbf{P} - \mathbf{K} \mathbf{Z} \mathbf{K}^T \end{aligned} \quad (4.33)$$

Auto-calibration et localisation de l'objet

À la différence d'une approche strictement SLAM, notre méthode ne cherche pas à se localiser par rapport à la scène et ne construit pas la cartographie. L'effet de supprimer la localisation, simplifie l'algorithme et réduit l'incertitude. Physiquement, le système utilise comme des capteurs proprioceptifs des capteurs angulaires très précis pour chacune des liaisons. Lorsque, l'approche est « look and move », les mesures de position aux arrêts sont libres de bruit de mesure. Par ailleurs, les amers visuels utilisés appartiennent à l'objet et sont en nombre fixe $m2$; pour pouvoir définir une situation dans l'espace, le système a besoin d'au moins trois amers non co-linéaires.

Le vecteur d'état possède deux types de composantes, la situation de la caméra C et la position des amers M . Chaque situation est définie par sept composantes (trois de position plus quatre d'orientation) et chaque amer par trois composantes (position de l'amer). Pour simplifier la tâche, les situations de la caméra et des amers sont définis dans le repère poignet \mathcal{R}_P par :

$$C = \begin{bmatrix} \mathbf{P} \\ \mathbf{Q} \end{bmatrix} \quad M = \begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_{m2} \end{bmatrix}$$

Le vecteur d'état X et la matrice des covariances P sont définis par :

$$X = \begin{bmatrix} C_G \\ C_D \\ M \end{bmatrix} \quad P = \begin{bmatrix} P_{CCG} & \mathbf{0} & P_{CPG} \\ \mathbf{0} & P_{CCD} & P_{CPD} \\ P_{PCG} & P_{PCD} & P_{PP} \end{bmatrix} \quad (4.34)$$

où l'indice D est utilisé pour la caméra droite et G pour celle de gauche.

Prédiction

Pour chacun des amers, on utilise le formalisme du filtre de Kalman étendu de telle sorte que dans l'étape de prédiction, la position de l'amer estimé dans le plan image est obtenue en utilisant le modèle de la caméra. Considérons la caméra droite, tel que y_{Dj} soit la mesure du j -ième amer dans le plan image

$$\hat{y}_{Dj} = \mathcal{H}_D(\hat{\mathbf{P}}_j)$$

pour $j = 1 \dots m2$. Pour l'amer j , l'innovation ou erreur de prédiction e_j est donnée par

$$e_{Dj} = y_{Dj} - \mathcal{H}_D(\hat{\mathbf{P}}_j) \quad (4.35)$$

Cette étape de prédiction n'affecte pas la matrice de covariances lorsque nous ne considérons pas d'imprécisions dans la mesure de la position du poignet.

Correction

Pour l'étape de correction, les équations sont :

$$\begin{aligned} \mathbf{Z}_{Dj} &= \mathbf{H}_{Dj} \mathbf{P} \mathbf{H}_{Dj}^T + \mathbf{R}_D \\ \mathbf{K}_{Dj} &= \mathbf{P} \mathbf{H}_{Dj}^T \mathbf{Z}_{Dj}^{-1} \\ \hat{\mathbf{X}}^+ &= \hat{\mathbf{X}} + \mathbf{K}_{Dj} e_{Dj} \\ \mathbf{P}^+ &= \mathbf{P} - \mathbf{K}_{Dj} \mathbf{Z}_{Dj} \mathbf{K}_{Dj}^T \end{aligned} \quad (4.36)$$

où la matrice \mathbf{H}_{Dj} est de la forme

$$\mathbf{H}_{Dj} = \begin{bmatrix} \frac{\partial \mathcal{H}_D}{\partial \mathbf{C}_D^T} \Big|_{C_D, \hat{\mathbf{P}}_j} & \mathbf{0} & \mathbf{0} & \dots & \frac{\partial \mathcal{H}_D}{\partial \mathbf{P}_j^T} \Big|_{C_D, \hat{\mathbf{P}}_j} & \dots & \mathbf{0} \end{bmatrix} \quad (4.37)$$

lorsque :

$$\left. \frac{\partial \mathcal{H}_D}{\partial \mathbf{P}_i^T} \right|_{C_D, \hat{\mathbf{P}}_i} = \mathbf{0}$$

pour $i \neq j$.

Les étapes de prédiction et correction sont réalisées, de manière similaire pour la caméra gauche, avec :

$$\mathbf{H}_{Gj} = \begin{bmatrix} \mathbf{0} & \left. \frac{\partial \mathcal{H}_G}{\partial \mathbf{C}_G^T} \right|_{C_G, \hat{\mathbf{P}}_j} & \mathbf{0} & \dots & \left. \frac{\partial \mathcal{H}_G}{\partial \mathbf{P}_j^T} \right|_{C_G, \hat{\mathbf{P}}_j} & \dots & \mathbf{0} \end{bmatrix} \quad (4.38)$$

Initialisation

Le processus d'initialisation consiste à introduire les différents amers dans le filtre. Ceci correspond à calculer la fonction de densité de probabilité qui lie l'état du filtre \mathbf{X} à la mesure \mathbf{Y} . En définissant la fonction $\mathcal{G} = \mathcal{H}^{-1}$.

L'initialisation de chacun des amers du vecteur d'état du filtre consiste à déterminer ses positions initiales. Nous considérons deux méthodes, la première méthode considère la reconstruction 3D de l'amer en utilisant le point image (\mathbf{P}_i) dans les deux caméras et les valeurs initiales des paramètres extrinsèques en calculant la profondeur s de l'amer par rapport aux caméras, équation 4.28. La deuxième méthode utilise une estimation initiale de la profondeur définie par l'utilisateur (reconstruction heuristique).

Soit \mathbf{P} la position d'un amer :

$$\mathbf{P} = \mathcal{G}(\mathbf{P}_i, s) \quad (4.39)$$

Nous définissons alors les fonctions

$$\mathbf{G}_{P_i} = \left. \frac{\partial \mathcal{G}}{\partial \mathbf{P}_i^T} \right|_{C, \mathbf{P}_i} \quad \mathbf{G}_C = \left. \frac{\partial \mathcal{G}}{\partial \mathbf{C}^T} \right|_{C, \mathbf{P}_i} \quad \mathbf{G}_s = \left. \frac{\partial \mathcal{G}}{\partial s} \right|_{C, \mathbf{P}_i} \quad (4.40)$$

Pour chacun des amers, la matrice de covariance est initialisée par :

$$\begin{aligned} \mathbf{P}_{PC} &= \mathbf{G}_{CG} \mathbf{P}_{PCG} + \mathbf{G}_{CD} \mathbf{P}_{PCD} \\ \mathbf{P}_{PP} &= \mathbf{G}_{P_iG} \mathbf{R}_{P_iG} \mathbf{G}_{P_iG}^T + \mathbf{G}_{sG} \mathbf{R}_{sG} \mathbf{G}_{sG}^T + \mathbf{G}_{CG} \mathbf{P}_{CCG} \mathbf{G}_{CG}^T + \\ &\quad \mathbf{G}_{P_iD} \mathbf{R}_{P_iD} \mathbf{G}_{P_iD}^T + \mathbf{G}_{sD} \mathbf{R}_{sD} \mathbf{G}_{sD}^T + \mathbf{G}_{CD} \mathbf{P}_{CCD} \mathbf{G}_{CD}^T \end{aligned} \quad (4.41)$$

où :

- \mathbf{P}_{PC} : Covariance entre l'amer et les paramètres de la caméra.
- \mathbf{P}_{CC} : Covariance entre les paramètres de la caméra.
- \mathbf{R}_{P_i} : Variance du bruit de mesure du point image.
- \mathbf{R}_s : Variance dans l'estimation de la profondeur.

Il faut noter que l'initialisation des matrices de covariance des amers est réalisée en considérant la caméra droite et la caméra gauche simultanément.

Définition de la situation suivante

En considérant que la situation initiale \mathbf{X}_i et la situation finale \mathbf{X}_f se trouvent à l'intérieur de l'espace des configurations du bras manipulateur et qu'il existe une trajectoire sans collision entre les deux.

Soit m le nombre d'étapes entre les situations initiale et finale. La situation finale est définie par l'utilisateur dans le repère objet. Des informations sur les amers et l'objet permettent de calculer la transformation entre le repère objet et le repère robot. Sinon, c'est l'information sur les amers qui est utilisée pour reconstruire la situation finale. Lorsque la cible est définie par plusieurs amers, nous utilisons leur moyenne ou « centre de masse » pour calculer la situation finale \mathbf{X}_f .

La situation suivante, à l' i -ième itération, est calculée par

$$\begin{aligned} \mathbf{P}_s &= \frac{i}{m-2} (\mathbf{P}_f - \mathbf{P}_0) + \mathbf{P}_0 \\ \mathbf{Q}_s &= \frac{i}{m-2} (\mathbf{Q}_f - \mathbf{Q}_0) + \mathbf{Q}_0 \end{aligned} \quad (4.42)$$

L'orientation définie par cette méthode doit être normalisée.

4.6.3 Avantages et inconvénients

L'avantage principal de cette méthode est sa capacité à gérer l'incertitude dans la modélisation du système de vision en utilisant un filtre de Kalman étendu. Cette méthode peut facilement être étendue à une stratégie plus dynamique en améliorant les capacités de traitement des images et/ou en considérant le mouvement de la plate-forme à condition d'augmenter les états du filtre.

Pour réduire le nombre d'itérations, on peut augmenter le nombre d'amers, ce qui améliore la convergence du filtre car on dispose de plus d'information. Malheureusement, le temps de calcul augmente avec le nombre d'amers.

4.7 Conclusions

Dans ce chapitre, nous avons abordé le problème de l'intégration de l'information provenant des caméras à partir de trois schémas différents qui utilisent tous le retour visuel dans la boucle de commande du bras manipulateur.

Dans le premier schéma en utilisant une approche d'asservissement visuel 2D, l'information visuelle est utilisée pour corriger les incertitudes dans la modélisation de la scène par une stratégie de commande partagée. Pour des environnements structurés, cette approche présente beaucoup d'avantages.

Le deuxième schéma considère la reconstruction 3D de la scène en utilisant une paire stéréo. La position 3D de la cible est utilisée pour définir la situation suivante dans un schéma dynamique qui considère des mouvements souples entre situations. Cette approche impose le mouvement souple à l'organe terminal pour les applications de suivi, ce qui est une caractéristique souhaitable pour des applications en robotique de service.

Le troisième schéma utilise un filtre de Kalman étendu dans la boucle de retour. Dans une stratégie « look and move » et en utilisant une paire stéréo comme deux capteurs angulaires indépendants, le filtre calcule la position de la cible en même temps qu'il corrige la calibration « hand-eye ». À notre connaissance, c'est la première utilisation du formalisme de SLAM pour résoudre le problème de calibration Hand-Eye.

Le chapitre suivant présente les résultats expérimentaux des schémas de commande introduits. Pour l'asservissement 2D, nous considérons le suivi d'une ligne droite. Pour l'asservissement 3D nous présentons un exemple de guidage visuel. Finalement, nous présentons les résultats en simulation obtenus pour l'auto-calibration.

Chapitre 5

Résultats expérimentaux

5.1 Introduction

Dans ce chapitre nous présentons quelques résultats expérimentaux obtenus sur la plate-forme expérimentale *Jido*. Elle est constituée d'une base mobile Neobotix MP_L655 et est équipée d'un bras manipulateur Mitsubishi PA-10 à six degrés de liberté, d'une paire de caméras (banc de stéréovision) « *videre* » et d'un PC pentium IV pour contrôler le bras et la paire stéréo. L'objectif de l'expérimentation est de valider les différents schémas présentés au cours de ce document et de montrer leurs fonctionnalités.

Tout d'abord nous présentons l'architecture logicielle qui implémente les différentes fonctionnalités. Ensuite nous donnons quelques résultats de commande du mouvement du bras manipulateur, pour des mouvements point à point puis pour une trajectoire continue. Enfin nous présentons les résultats obtenus après l'intégration des données visuelles dans la boucle de commande.

5.2 Architecture logicielle

Nous utilisons « l'architecture LAAS » [Alami et al., 1998], nous présentons brièvement ici cette architecture logicielle qui implémente les différentes fonctionnalités. Nous avons contribué au développement des modules représentés en gras sur la figure 5.1.

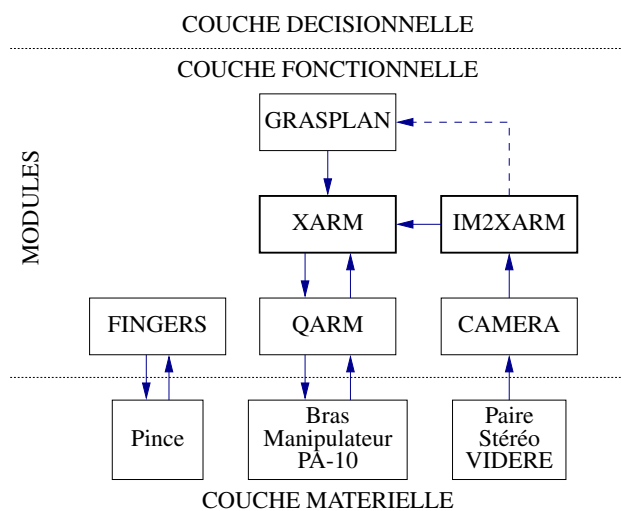


FIG. 5.1 – Architecture logicielle pour la manipulation

Cette architecture logicielle s'abstrait de la couche matérielle. Cela est rendu possible par l'utilisation de l'environnement de programmation $G^{en}bM$ (Générateur de Modules [Fleury et al., 1997]). Chaque module est écrit en langage de programmation C et implémente une fonctionnalité. $G^{en}bM$ gère alors les éléments suivants :

- La compilation spécifique pour le système informatique embarqué sur le robot.
- Les communications et le partage de ressources entre les différents modules.
- La synchronisation temporelle des différents modules.

Une description de chaque module est la suivante :

- QARM : Ce module gère le bras manipulateur. Il réalise la mesure des positions angulaires de chacune des liaisons, vérifie l'état du bras et envoie les consignes en position ou en vitesse à chacune des articulations.
- XARM : Ce module calcule les consignes de vitesse pour QARM à partir des positions angulaires des articulations. Il réalise le calcul des consignes pour des mouvements point à point et pour des trajectoires continues dans l'espace cartésien en même temps qu'il limite la vitesse, l'accélération et le jerk du mouvement de l'organe terminal du bras manipulateur.
- CAMERA gère l'acquisition des images par la paire stéréo videre.
- IM2XARM réalise le traitement des images et produit les signaux utilisés par le retour visuel dans les différents schémas de commande visuel.
- FINGERS commande la pince, il permet son ouverture ou sa fermeture.
- GRASPLAN planifie des trajectoires pour réaliser la saisie des objets en évitant les collisions avec des obstacles. Ces trajectoires sont envoyées au module XARM pour être exécutées.

5.3 Mouvement point à point

Dans cette section nous traiterons le cas du mouvement point à point, avec l'intention d'explicitier la performance de l'étape de génération des mouvements souples et de la boucle de commande. Nous considérons tout d'abord le cas d'une translation pure puis celui d'une rotation pure.

5.3.1 Translation

Dans ce cas, le robot réalise un mouvement point à point en trois dimensions. Pour ce mouvement, l'orientation de l'organe terminal est fixe pendant le déplacement.

La figure 5.2 présente le mouvement réalisé entre les points A et B par le bras manipulateur. Les contraintes de jerk, d'accélération et de vitesse sont identiques pour chacune des directions X , Y et Z . Avec :

$$A = \begin{bmatrix} 0.695 \\ 0.000 \\ 0.600 \end{bmatrix} \quad B = \begin{bmatrix} 0.600 \\ 0.100 \\ 0.700 \end{bmatrix} \quad \begin{aligned} J_{max} &= 0.6 \text{ m/s}^3 \\ A_{max} &= 0.2 \text{ m/s}^2 \\ V_{max} &= 0.1 \text{ m/s} \end{aligned}$$

La figure 5.3 présente quelques résultats obtenus. Les courbes de gauche montrent l'accélération, la vitesse et la position pour les directions X et Z . Le mouvement est négatif dans la direction X et positif dans les directions Y et Z . Les déplacements sont d'une longueur différente, ce qui permet d'évaluer la performance de l'ajustement de temps pour obtenir un mouvement en ligne droite. Les courbes de droite permettent de réaliser une comparaison des performances pendant l'exécution du mouvement en présentant la position désirée et la position mesurée. On peut voir que ces courbes semblent pratiquement confondues, nous observons des erreurs de l'ordre du dixièmes de millimètre.

5.3.2 Rotation

Dans ce deuxième cas, nous considérons un mouvement de rotation pur autour de l'axe Y . L'organe terminal va d'une orientation horizontale Q_o vers une orientation verticale Q_f , pendant que la position

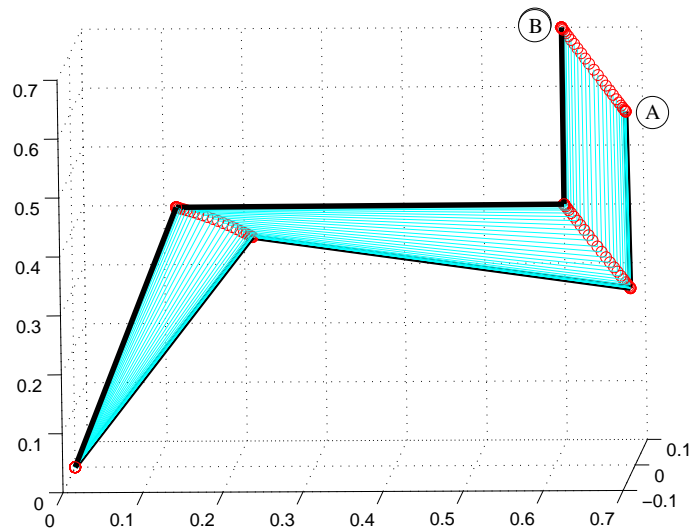


FIG. 5.2 – Translation entre les points A et B sans changement d'orientation. Cette vue présente sous forme filaire les trois premiers corps du bras manipulateur.

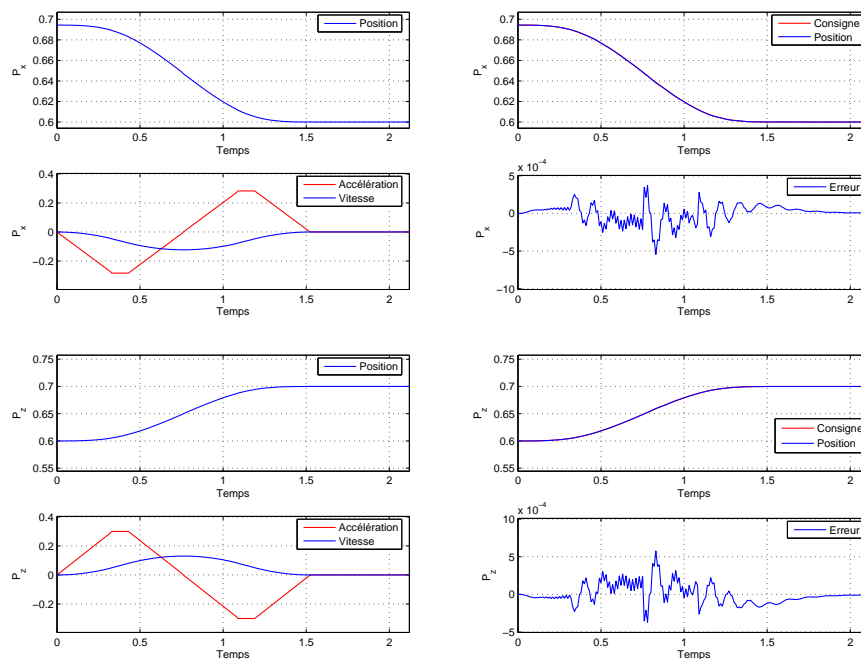


FIG. 5.3 – Résultats obtenus après un mouvement de translation pure. À gauche, les courbes d'accélération, de vitesse et de position pour les directions X et Z . À droite, des courbes comparatives entre la position de consigne et la position mesurée.

de l'organe terminal ne varie pas. La figure 5.4 présente le mouvement réalisé par le bras manipulateur. L'orientation horizontale Q_o et l'orientation verticale sont définies par les quaternions :

$$Q_o = \begin{bmatrix} 1.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix} \quad Q_f = \begin{bmatrix} 0.7071 \\ 0.0000 \\ 0.7071 \\ 0.0000 \end{bmatrix}$$

Les contraintes de jerk, d'accélération et de vitesse pour chacune des composantes du quaternion sont données par :

$$J_{max} = 0.6 \quad m/s^3 \quad A_{max} = 0.2 \quad m/s^2 \quad V_{max} = 0.1 \quad m/s$$

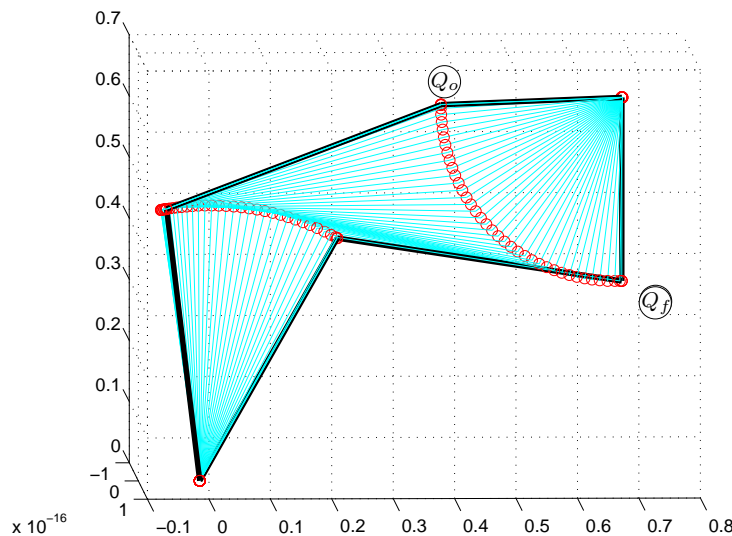


FIG. 5.4 – Rotation entre les quaternions Q_o et Q_f sans changement de position de l'organe terminal.

La figure 5.5 présente les résultats obtenus durant un déplacement en rotation. Du côté gauche, nous avons les courbes produites par le générateur de mouvements souples. Nous avons établi que les mouvements doivent avoir une même durée, lorsque les déplacements sont de longueur différente une adaptation des contraintes est réalisée pour corriger la durée. La vitesse atteinte par la composante n est inférieure à la vitesse atteinte par la composante j qui est maximale. Du côté droit, nous montrons la comparaison entre la consigne d'orientation et l'orientation mesurée puis l'erreur d'orientation pour les composantes n et j .

La figure 5.6 présente les résultats obtenus pour la position de l'organe terminal. Lorsque cette position est fixe, les valeurs de consigne de la boucle de commande de position sont nulles, cependant la boucle de commande doit corriger les erreurs de position. Nous observons que les erreurs sont plus importantes (de l'ordre d'un millimètre) au début et à la fin du mouvement de rotation, ce qui nous permet de dire que le système de commande a une réponse satisfaisante pour les tâches de régulation. Il est important de noter que le mouvement de rotation passe par une singularité dans les différentes configurations du bras. La robustesse de la boucle de commande obtenu par l'utilisation de l'inverse généralisée de la matrice jacobienne est montrée dans cette expérience. Lorsque le mouvement de rotation traverse une singularité, aucun effet n'est visible.

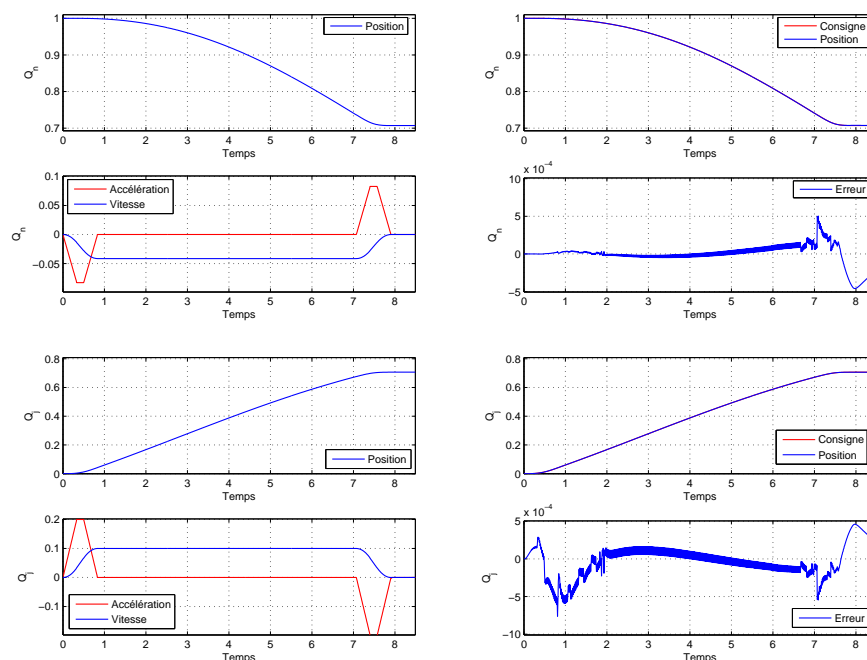


FIG. 5.5 – Résultats obtenus après un mouvement de rotation pure. A gauche, les courbes d'accélération, de vitesse et de position pour les composantes du quaternion n et j . A droite, des comparaisons entre la position de consigne et la position mesurée, et les erreurs de position.

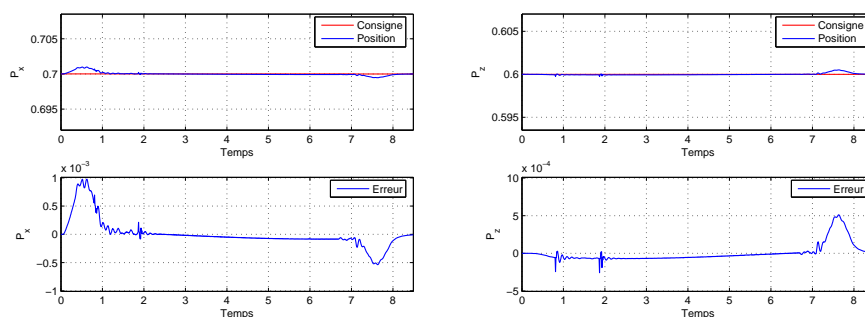


FIG. 5.6 – Résultats obtenus pendant un mouvement de rotation pure. La position est fixe, donc nous présentons les directions X et Z , avec du côté gauche la direction X et du côté droit la direction Z . En haut la comparaison entre la consigne et la valeur mesurée, en bas l'erreur obtenue.

5.4 Mouvement continu

Pour tester le générateur de mouvements souples sur une trajectoire continue, nous considérons une trajectoire construite par le planificateur de saisie développé par López [Lopez-Damian et al., 2005]. La trajectoire est définie par 10 configurations successives que le bras manipulateur doit parcourir depuis une configuration quelconque jusqu'à la configuration de saisie de la boîte bleue située sur la table. La figure 5.7 présente la séquence de mouvements réalisée par le bras manipulateur telle qu'elle a été produite par le planificateur.

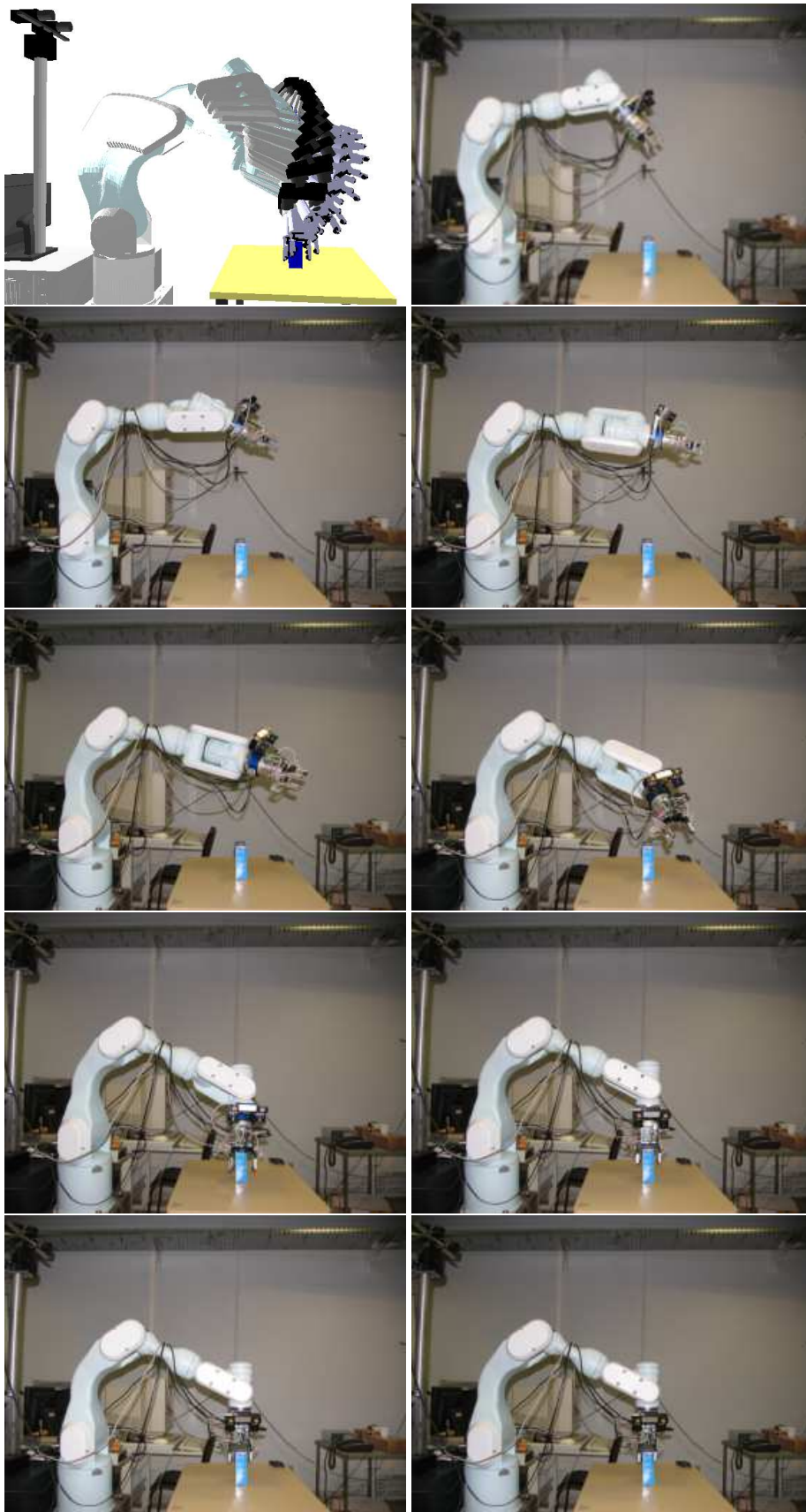


FIG. 5.7 – Séquence de mouvement réalisée par le bras manipulateur pour saisir la boîte bleue.

Pour comparer la trajectoire réalisée par le bras manipulateur et la trajectoire originale définie par le planificateur de saisie, la figure 5.8 présente les deux trajectoires. Les configurations du bras manipulateur présentées en noir sur la figure correspondent à la trajectoire définie par le planificateur. Le tracé du mouvement réalisé par Jido lors de l'exécution est présenté en bleu. L'évolution de la pointe de l'organe terminal est représentée par un point noir pour chaque configuration. Nous pouvons observer que le mouvement pour la saisie est bien réalisé en passant par les configurations planifiées.

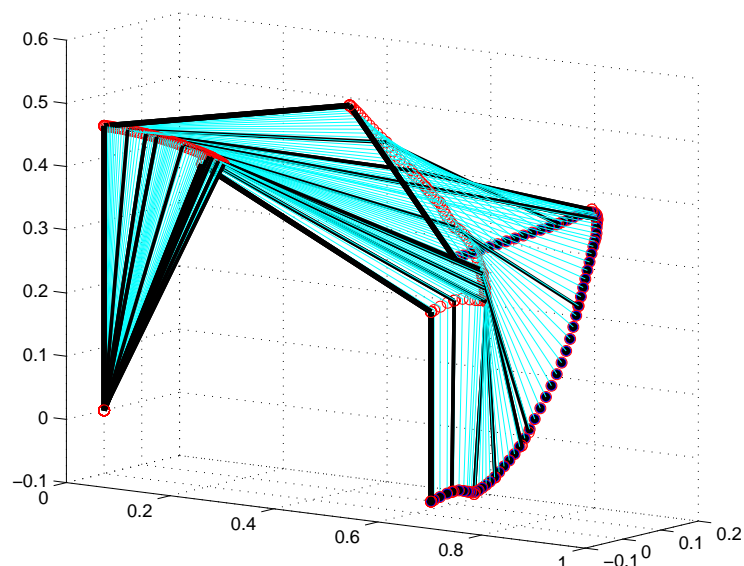


FIG. 5.8 – Comparaison entre la trajectoire planifiée (noir) et la trajectoire réalisée (bleu). Les points noirs définissent la trajectoire réalisée par la pointe de l'organe terminal.

Les figures 5.9 et 5.10 présentent à leur gauche les courbes d'accélération, de vitesse et de position produites par le générateur de mouvements souples pour les dix configurations de la trajectoire de saisie.

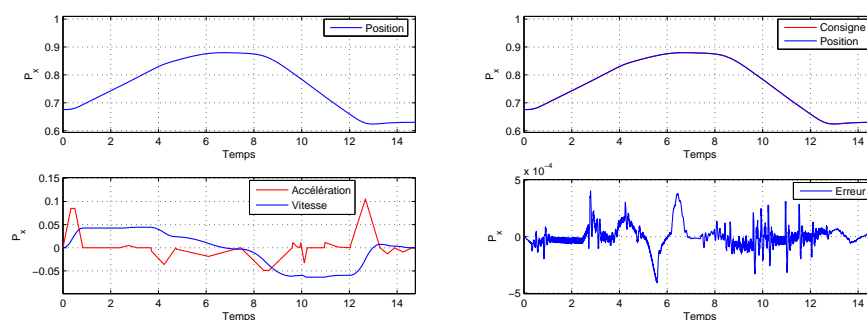


FIG. 5.9 – Courbes planifiées pour la direction X du côté gauche et erreurs obtenues du côté droit.

Nous avons choisi les courbes représentant la direction X (Figure 5.9) et la composante n du quaternion (Figure 5.10) comme les plus représentatives, car elles réalisent des déplacements longs et montrent plus d'effets quand la configuration du bras est proche d'une singularité. À droite des figures, on peut voir les courbes comparatives entre la consigne de position et la position mesurée de ces variables et les erreurs obtenues lorsque Jido réalise le mouvement de saisie. L'effet de la singularité peut être apprécié autour de la sixième seconde sur les courbes d'erreur. C'est l'instant où les erreurs atteignent leurs valeurs maximales. Pour expliquer cet événement, nous devons considérer les courbes de vitesse planifiées,

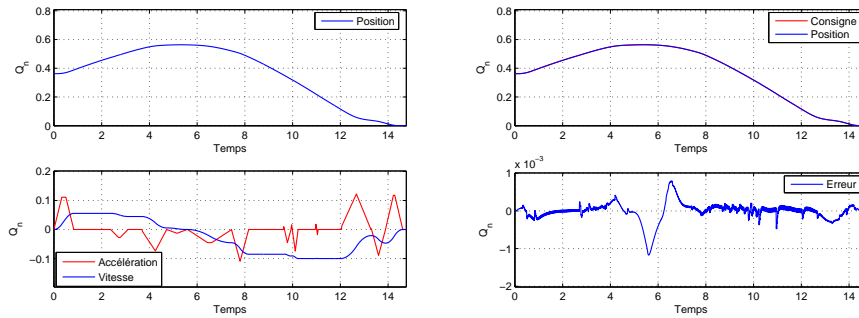


FIG. 5.10 – À gauche les courbes planifiées pour la composante n et à droite les comparaisons de performance.

elles sont présentées sur les images de gauche. Nous pouvons apprécier que c'est autour de la sixième seconde que la direction X et la composante n changent de sens. Ainsi, les vitesses sont faibles, ce qui produit un temps relativement long au voisinage de la singularité.

Nous pouvons remarquer que le calcul des consignes pour une trajectoire quelconque est réalisé en temps réel. Pour une trajectoire de 100 configurations, proches consécutivement, le calcul est réalisé en moins de 30 ms. Pour la trajectoire de la saisie de l'objet, le calcul de consignes est réalisé en moins de 10 ms sur un PC pentium IV.

Le suivi de la trajectoire a été réalisé correctement avec des erreurs de l'ordre de quelques dixièmes de millimètre, ce qui permet bien la saisie d'un objet.

5.5 Asservissement visuel partagé

Dans cette section nous abordons l'utilisation des informations visuelles dans la boucle de commande. Cette première approche est fondée sur le schéma de commande partagé introduit à la section 4.4. Le but de cette expérience est de montrer la flexibilité du schéma dans un exemple simple, en n'utilisant que les informations 2D provenant de la caméra gauche du banc stéréo « videre ».

Considérons un mouvement de suivi d'une ligne verticale. La longueur et la direction du déplacement (Z) sont connues *a priori*, tels que le générateur de mouvements souples a calculé les consignes de vitesse et de position. La caméra est montée en bout du bras manipulateur de Jido tel qu'illustré sur la figure 5.11.

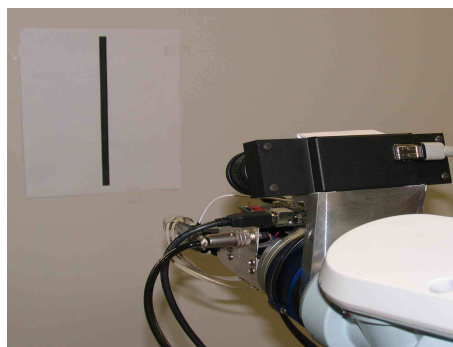


FIG. 5.11 – Montage « hand-eye » de la caméra et la ligne à suivre

Deux expériences ont été réalisées. La figure 5.12 présente les deux mouvements réalisés. Le côté gauche présente le mouvement dans la direction Z réalisé tel qu'il a été planifié, ce qui est équivalent à un modèle parfait de l'environnement et à une parfaite localisation du robot. Comme nous pouvons

le vérifier sur la figure, les erreurs de modélisation ou de localisation ne sont pas corrigées et l'organe terminal du manipulateur réalise le déplacement vertical sans considérer la position de la ligne droite. Le côté droit de la figure présente le mouvement quand la boucle d'asservissement visuel est activée. Pendant que l'extrémité du bras se déplace, vers le haut, dans la direction Z , une correction dans la direction Y est réalisée, pour corriger les erreurs de modélisation ou de localisation.

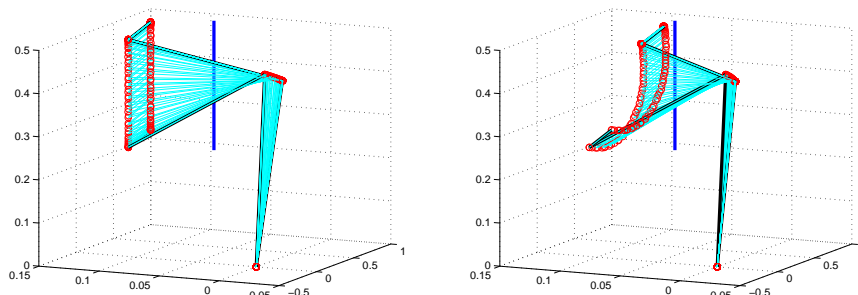


FIG. 5.12 – Asservissement visuel partagé pour le suivi ascendant d'une ligne droite. À gauche le mouvement planifié et à droite le mouvement réalisé avec l'asservissement visuel activé.

La figure 5.13 présente les résultats obtenus concernant les variables du système. Le côté gauche présente le mouvement dans la direction Z . Comme nous l'attendions, le mouvement dans cette direction est indépendant du retour visuel. Par contre, le côté droit montre l'effet du retour visuel en présentant les résultats obtenus dans la direction Y . La partie supérieure du côté droit présente la consigne planifiée P_y , lorsque le mouvement est vertical la consigne planifiée est constante, cependant la position mesurée de l'organe terminal P_y n'est pas sur la ligne de consigne. Nous pouvons observer que la position mesurée est différente de la consigne planifiée. Cependant, la loi de commande que nous utilisons est partagée :

$$P_p = P_g + P_v$$

où P_v est la consigne visuelle produite par la boucle d'asservissement visuel et P_p est la consigne partagée. Dans la partie inférieure du côté droit, nous présentons l'erreur entre la position mesurée et la consigne partagée. Nous observons que l'erreur converge vers zéro.

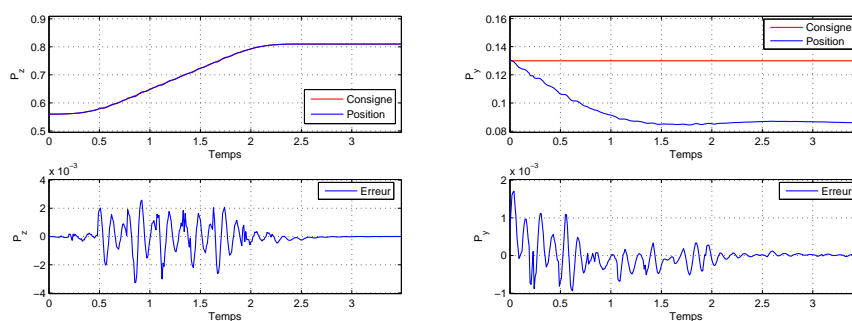


FIG. 5.13 – Mouvement dans la direction Z et erreur de déplacement en Z à gauche. À droite, comparaison entre la consigne planifiée et la position mesurée dans la direction Y , et l'erreur entre P_p et P_y .

5.6 Guidage visuel

Ici nous présentons les résultats expérimentaux du guidage visuel qui illustrent les développements de la section 4.5.

Le but de l'expérience est de demander au robot de suivre une cible mobile. La position de la cible dans le repère du bras est calculée à partir des informations de stéréovision de la paire videre. L'objectif est de montrer la possibilité et surtout les avantages de l'utilisation du générateur de mouvements souples.

La tâche de suivi peut être décrite simplement à partir de la position relative de la cible par rapport au bras. Elle consiste à effectuer le mouvement souple entre la position courante et la position but. Ce mouvement est actualisé à chaque nouvelle acquisition d'une position de la cible par les caméras.

Les capacités du système de génération de mouvements souples permettent une actualisation en temps réel de la trajectoire. Le système de vision utilisé ne permet pas d'atteindre une cadence importante, ni de suivre une cible compliquée. C'est pourquoi nous utilisons un point noir sur un tableau blanc comme cible.

Même si un système de vision moderne permettrait de mieux mettre en valeur ces résultats, nous obtenons déjà des résultats intéressants comme l'illustrent nos résultats.

5.7 « Auto-calibration Hand-Eye »

Cette section montre les résultats de la simulation faite pour valider la localisation de la cible et les paramètres de calibration hand-eye tel que présenté dans la section 4.6. Pour simuler le système, nous considérons que nous disposons des paramètres de la caméra. En utilisant le modèle sténopé de la caméra nous construisons les images. Nous supposons que la position de la caméra dans le repère monde est connue de façon imprécise à travers deux transformations. La première transformation est connue de façon précise et correspond au mouvement du bras manipulateur. La deuxième transformation est connue grossièrement et correspond à la calibration hand-eye.

La cible est modélisée par 8 points définis dans le repère monde. La « photo » est prise en utilisant les modèles simulés et en ajoutant du bruit. La figure 5.14 présente l'évolution des deux points différents dans l'image (i.e. variation de la position des points lorsque la caméra s'approche). La courbe lisse correspond à la position image du point sans bruit et l'autre courbe est la position mesurée qui contient du bruit.

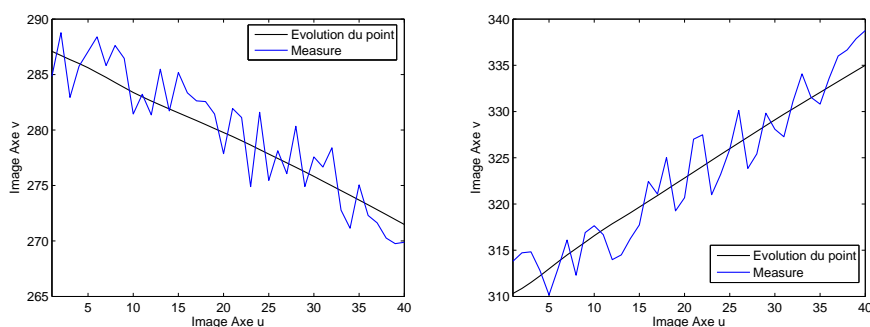


FIG. 5.14 – Évolution de deux points dans l'image après 40 « photos ».

La calibration hand-eye est exprimée sous la forme d'un vecteur de position et d'un quaternion. Le modèle étant non linéaire, la calibration a été réalisée en deux parties. La première partie cherche à améliorer la connaissance sur l'orientation des caméras en considérant des erreurs fixes pour la position des caméras. La deuxième partie considère des erreurs fixes pour l'orientation des caméras et le système cherche à améliorer la connaissance de la position des caméras. La figure 5.15 présente l'évolution de l'estimation de la position de la cible.

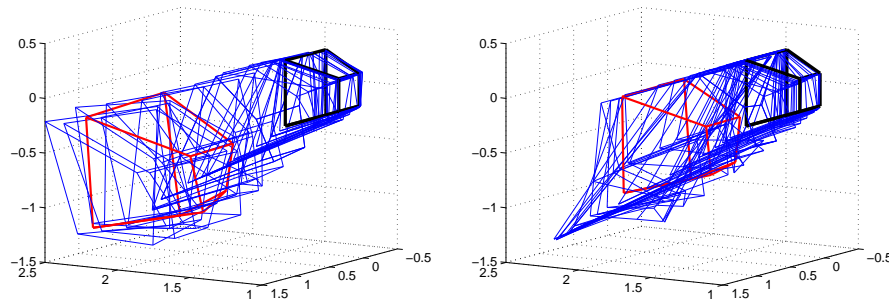


FIG. 5.15 – L'estimation de la position de la cible converge vers la position réelle de la cible (en noir) à partir d'une position initiale estimée (en rouge). À gauche, l'évolution de l'estimation de la position de la cible pour la calibration de l'orientation et à droite pour la calibration de la position.

Les données provenant des caméras contiennent plus d'information sur l'orientation des amers que sur leur position. Nous choisissons donc pour la première étape de la calibration de corriger l'orientation en utilisant une position de la caméra grossièrement estimée. La deuxième étape corrige la position à partir de la calibration de l'orientation obtenue à la première étape.

La figure 5.16 présente l'évolution de l'estimation des composantes n et i du quaternion d'orientation des caméras. Pour réaliser une simulation plus réaliste, nous considérons la distribution physique de la paire Videre. Comme nous l'observons sur la figure 5.11, les caméras sont montées dans un seul boîtier, leur position relative est définie par le fabricant. Lors du montage sur l'extrémité du bras manipulateur de Jido, l'orientation est la même pour les deux caméras. Nous avons initialisé leurs orientations avec des erreurs symétriques autour de l'orientation des caméras simulées. Comme la connaissance de l'orientation de la caméra est grossière, nous initialisons les matrices de covariance avec des valeurs qui visent à étendre la zone de recherche.

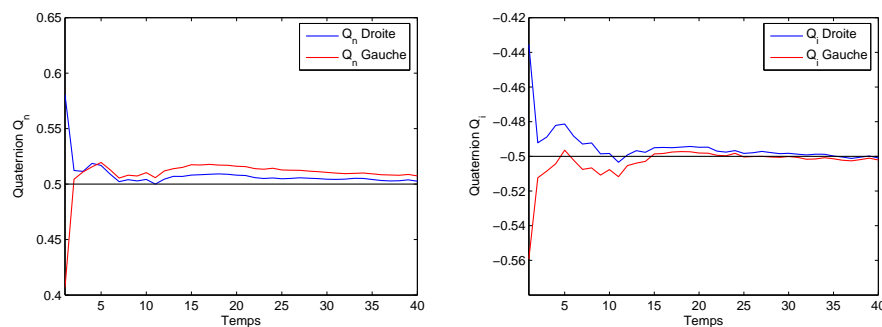


FIG. 5.16 – Évolution de l'estimation de l'orientation des caméras. À gauche la composante n du quaternion et à droite le composante i . Les lignes centrales correspondent aux valeurs simulées.

La figure 5.17 présente l'évolution de l'estimation de la position de la caméra dans les directions X et Z . Ces directions correspondent à la profondeur et à la hauteur de la caméra. Nous considérons des erreurs d'orientation faibles et les matrices de covariance ont été initialisées à des valeurs « faibles » (10^{-6}) pour des erreurs de l'ordre de quelques millimètres. Ces conditions produisent des estimations de la position des caméras qui convergent lentement.

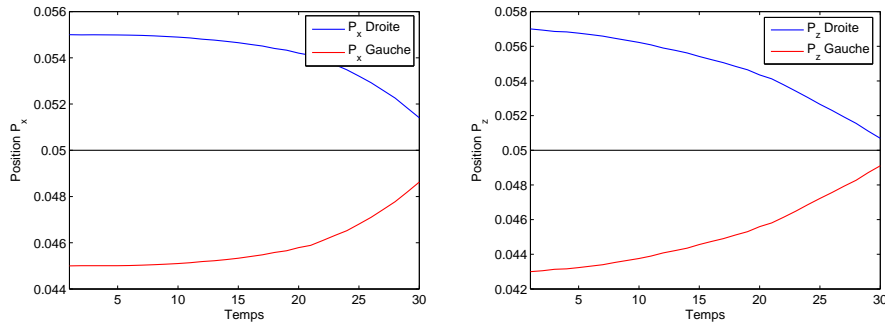


FIG. 5.17 – Évolution de l’estimation de la position des caméras. À gauche la position en X et à droite la position en Z . Les lignes centrales correspondent aux valeurs simulées.

5.8 Conclusions

À l’exception de l’auto-calibration restée au niveau de la simulation par manque de temps, toutes les expériences ont été conduites sur la plate-forme expérimentale Jido. Les résultats obtenus sont satisfaisants et ont permis une intégration dans d’autres travaux.

Pour la commande de bras manipulateur, les expériences ont montré une robustesse au voisinage des singularités. Le cas le plus critique se présente quand la position et l’orientation finales correspondent à une configuration singulière, mais les protections que nous utilisons garantissent une réponse satisfaisante.

Concernant la précision du mouvement, la réponse est satisfaisante. Les erreurs sont négligeables pour les applications que nous envisageons. Les expériences réalisées pour la saisie d’un objet l’ont été avec succès. Nous avons réalisé des test pour le transport de verres contenant des liquides. Les résultats ont été très satisfaisants, pour des trajectoires simples comme pour des trajectoires complexes.

L’utilisation du générateur de mouvements souples pour des applications en temps réel a été une réussite.

La simplicité du schéma de commande a permis d’incorporer très facilement les boucles de vision pour corriger les erreurs de modélisation ou de localisation. Le schéma reste ouvert pour des applications utilisant d’autres capteurs.

Dans le cadre du projet européen *Cogniron*, les algorithmes développés ont été utilisés quotidiennement pour réaliser les trajectoires calculées par les différents planificateurs de trajectoires.

Le confort produit par la souplesse du mouvement a pu être validé par les autres participants au projet qui évoluent en face du robot pour des tâches en interaction avec celui-ci.

Conclusions

Nous avons abordé dans cette thèse plusieurs aspects de la commande des robots manipulateurs en robotique de service et notamment l'utilisation du retour visuel pour faire évoluer les robots dans les environnements non structurés propres à l'homme.

Nous avons établi que les conditions opératoires de la robotique de service sont différentes de celles de la robotique industrielle. Principalement des contraintes temporelles et de précision pour la robotique industrielle et des contraintes portant sur la sécurité et le confort de l'utilisateur pour la robotique de service. Ce travail ne réalise pas une recherche exhaustive sur tous les domaines pour satisfaire ces contraintes, mais essaie de proposer un schéma de commande qui permet la réalisation de différentes tâches en respectant des contraintes cinématiques lorsque l'information provient de capteurs extéroceptifs, dans le cas particulier de ce travail seule la vision a été utilisée.

Une première partie de ce travail est dédiée à la commande des bras manipulateurs. À partir des schémas classiques, nous reprenons le schéma de commande cartésienne utilisant les quaternions comme signal de retour pour l'orientation et nous présentons un schéma qui permet au robot d'évoluer dans les environnements humains. Nous pouvons formuler trois remarques. Tout d'abord la stratégie de commande reste ouverte pour l'inclusion de capteurs extéroceptifs en respectant les contraintes d'accélération et de vitesse de façon intrinsèque. Ensuite, la commande est robuste au passage des configurations singulières pendant le suivi d'un chemin réalisé en respectant les contraintes cinématiques. Finalement, la précision obtenue par le schéma due à l'utilisation du terme intégral permet la réalisation des tâches « en aveugle ».

Une deuxième partie est consacrée à l'étude des mouvements souples. Mouvements en temps minimal qui, par la limitation du jerk, de l'accélération et de la vitesse, permettent la réalisation de tâches propres à la robotique de service. La « souplesse » des mouvements obtenus augmente le confort des utilisateurs humains.

La troisième partie est relative à l'utilisation du retour visuel dans la boucle de commande afin de doter le robot d'une certaine autonomie pour la réalisation de tâches. Trois schémas différents ont été proposé pour montrer la flexibilité de la stratégie de commande du bras manipulateur.

Des résultats expérimentaux et de simulation sont présentés pour valider les différentes approches. L'application logicielle développée au cours de ce travail est utilisée quotidiennement pour commander le robot Jido dans le cadre du projet Cogniron.

Perspectives

Deux axes de recherches apparaissent à l'issue de ce travail : la génération de mouvements et la commande par retour extéroceptifs.

Pour la génération du mouvement, nous n'avons considéré que le cas des points de passage produit par un planificateur de trajectoires. Peut-on satisfaire les contraintes géométriques et cinématiques à ce niveau ? Peut-on étendre cette approche pour les contraintes dynamiques ? L'introduction des trajectoires souples au niveau d'un planificateur de chemin apporterait-elle des améliorations ?

Les mouvements souples à sept segments apportent-ils une solution pour la commande des systèmes redondant ou non-holonomes ?

Nous avons présenté trois schémas différents pour traiter l'utilisation du retour visuel. Ces schémas utilisent la vision d'une façon simple, mais tout le potentiel de cette approche n'a pas été exploité au cours de ce travail. De futurs travaux sont nécessaires pour proposer des schémas opérationnels. La

convergence du filtre de Kalman étendu n'est pas toujours garantie pour la localisation de l'objet et pour l'auto-calibration. C'est notamment le cas lorsque la plate-forme est mobile car dans ce cas la mesure de la position de la base du bras manipulateur par rapport au repère du monde est fortement bruitée. Finalement, toutes les manipulations ont été réalisées dans l'espace libre du bras manipulateur, que se passe-t-il si on travaille dans un espace contraint en utilisant des capteurs d'effort ? Est-il possible de définir une stratégie hybride de la même façon que nous avons défini la stratégie partagée en respectant la souplesse du mouvement ?

Annexe A

Coordonnées Opérationnelles

Nous précisons ici quelques éléments nécessaires à notre étude. Pour plus des détails le lecteur pourra, par exemple, se reporter aux ouvrages de [Renaud, 1996] et [Bruyninckx et al., 2001].

Pour définir la *situation* d'un objet dans un espace de dimension 3 (\mathbf{E}^3) nous devons définir sa position et son orientation.

A.1 Position

Nous repérons la position d'un point dans un espace (\mathbf{E}^3) par ses coordonnées. Parmi les systèmes de coordonnées nous mentionnerons : les coordonnées cartésiennes, les coordonnées cylindriques et les coordonnées sphériques (Cf. figure A.1)

La position du repère \mathcal{R}_{n+1} par rapport au repère \mathcal{R}_n est définie par les coordonnées de l'*Origine* \mathbf{O}_{n+1} par rapport à l'origine \mathbf{O}_n .

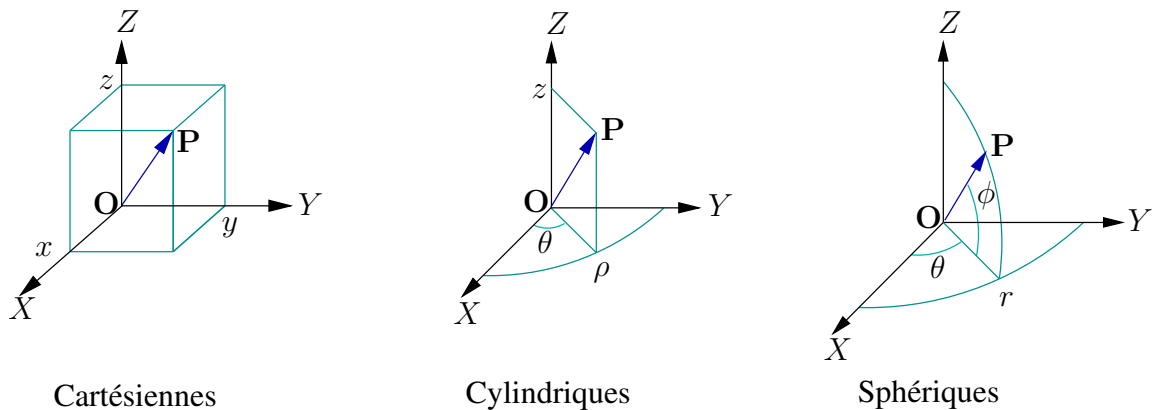


FIG. A.1 – Représentation de la position du point **P** dans les divers systèmes de coordonnées.

A.1.1 Coordonnées cartésiennes

Un point **P** est défini par ses coordonnées cartésiennes par rapport à un point origine **O**.

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

A.1.2 Coordonnées cylindriques

Un point P est défini par ses coordonnées cylindriques par rapport à un point origine O .

$$P = \begin{bmatrix} \rho \\ \theta \\ z \end{bmatrix}$$

A.1.3 Coordonnées sphériques

Un point P est défini par ses coordonnées sphériques par rapport à un point origine O .

$$P = \begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix}$$

A.2 Orientation

L'orientation du repère \mathcal{R}_1 par rapport au repère \mathcal{R}_0 peut être définie par une *rotation* et représentée de diverses manières. il existe des représentations minimales constituées de trois composantes et des représentations non minimales. Parmi les représentations minimales nous pouvons citer les angles de Bryant et les angles de Euler. Parmi les représentations non minimales nous pouvons citer la matrice de rotation, la représentation angle plus vecteur et les quaternions.

A.2.1 Angles de Bryant

Les angles de Bryant correspondent aux trois rotations successives (Cf. figure A.2.1) :

- λ autour de x_{i-1}
- μ autour de y_λ
- ν autour de z_μ

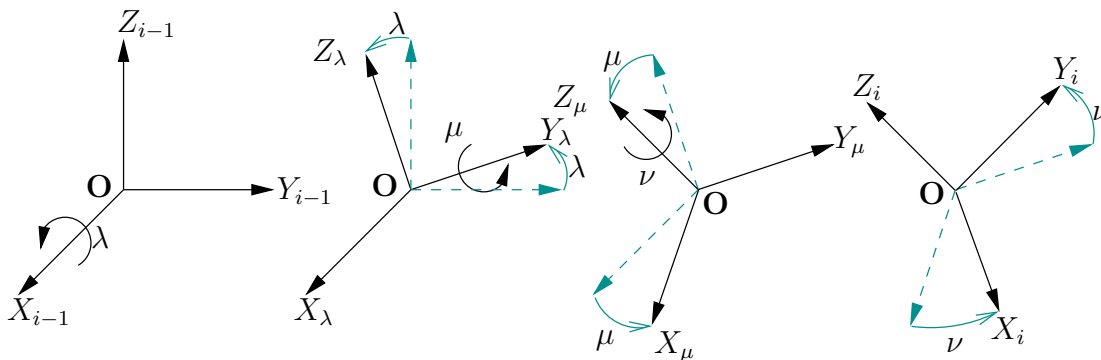
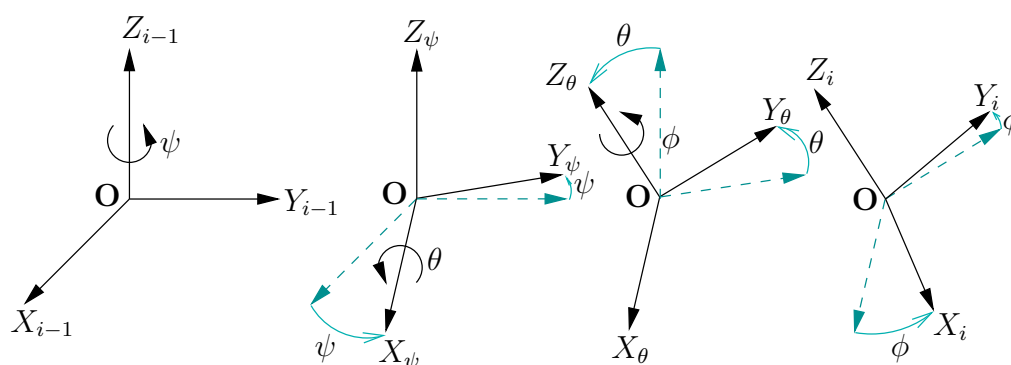


FIG. A.2 – Angles de Bryant, les trois rotations successives (λ, μ, ν)

A.2.2 Angles d'Euler

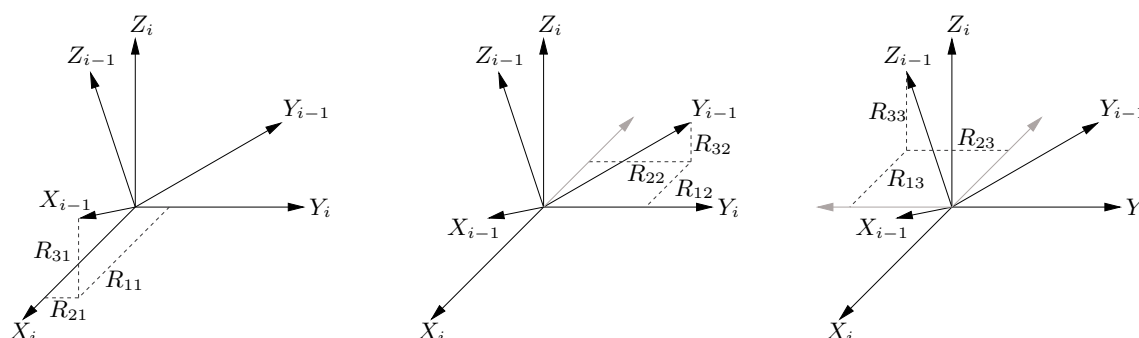
Les angles d'Euler correspondent aux trois rotations successives (Cf. figure A.2.2) :

- ψ autour de z_{i-1}
- θ autour de x_ψ
- ϕ autour de z_θ


FIG. A.3 – Angles d’Euler, les trois rotations successives (ψ , θ , ϕ)

A.2.3 Matrice de rotation

La matrice de rotation \mathbf{R} appelé aussi *matrice d’orientation* ou *matrice des cosinus directeurs*, représente la rotation entre deux repères (\mathcal{R}_{i-1} et \mathcal{R}_i). C’est une matrice carrée d’ordre 3. Les colonnes de la matrice ${}^{i-1}R_i$ représentent les composantes des vecteurs unitaires du repère \mathcal{R}_{i-1} dans le repère \mathcal{R}_i . (Cf. figure A.2.3).


FIG. A.4 – Composantes des vecteurs unitaires du repère \mathcal{R}_{i-1} exprimées dans le repère \mathcal{R}_i .

A partir de la figure A.2.3, nous pouvons écrire les matrices de rotations qui correspondent aux rotations autour des axes x, y et z :

- α autour de x
- ϕ autour de y
- θ autour de z

$$\mathbf{R}(X, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (\text{A.1})$$

$$\mathbf{R}(Y, \phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \quad (\text{A.2})$$

$$\mathbf{R}(Z, \theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.3})$$

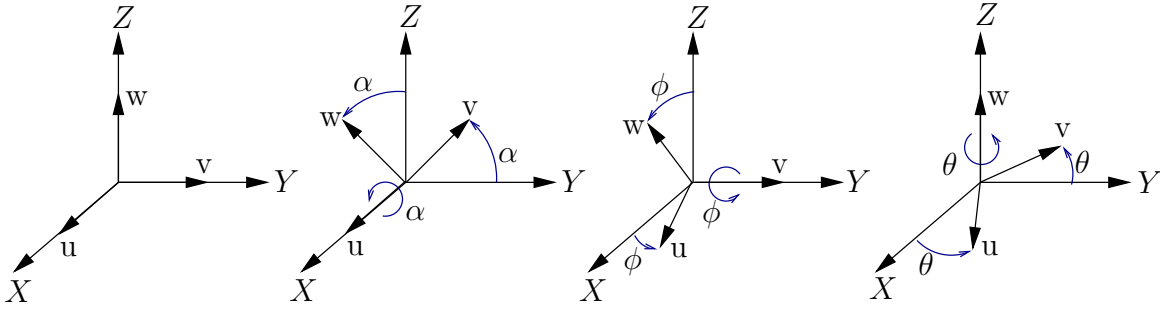


FIG. A.5 – Rotations autour des axes x, y et z

La représentation d'une rotation par matrice de rotation *n'est pas minimale* car elle utilise neuf paramètres pour représenter les trois degrés de liberté.

Inverse d'une matrice de rotation

L'inverse de la matrice de rotation ${}^O\mathbf{R}_i$ est définie par sa transposée :

$${}^O\mathbf{R}_i^{-1} = {}^O\mathbf{R}_i^T = {}^i\mathbf{R}_O \quad (\text{A.4})$$

Rotation relative entre matrices de rotation

Soient ${}^O\mathbf{R}_i$ et ${}^O\mathbf{R}_j$ deux rotations définies par rapport au repère \mathcal{R}_O . La rotation relative ${}^i\mathbf{R}_j$ entre les deux rotations est défini par

$${}^i\mathbf{R}_j = {}^O\mathbf{R}_i^T {}^O\mathbf{R}_j = {}^i\mathbf{R}_O {}^O\mathbf{R}_j \quad (\text{A.5})$$

Rotation d'un point

Considérons \mathbf{P}_a les coordonnées d'un point exprimées dans le repère \mathcal{R}_a . Soit ${}^a\mathbf{R}_b$ la matrice de rotation qui lie le repère \mathcal{R}_a au repère \mathcal{R}_b . Les coordonnées du point \mathbf{P} dans le repère \mathcal{R}_b sont données par l'équation :

$$\mathbf{P}_b = {}^a\mathbf{R}_b \mathbf{P}_a \quad (\text{A.6})$$

A.2.4 Angle plus vecteur

Chaque matrice de rotation \mathbf{R} peut être obtenue par une seule rotation d'angle α autour d'un axe unitaire équivalent \mathbf{e}_{eq} .

Si nous connaissons l'angle α et le vecteur \mathbf{e}_{eq} , nous pouvons construire la matrice de rotation équivalent $\mathbf{R}(\mathbf{e}_{eq}, \alpha)$ par la composition de cinq rotations suivante :

$$\mathbf{R}(\mathbf{e}_{eq}, \alpha) = \mathbf{R}(Z, -\theta)\mathbf{R}(Y, -\phi)\mathbf{R}(X, \alpha)\mathbf{R}(Y, \phi)\mathbf{R}(Z, \theta) \quad (\text{A.7})$$

Soit

$$\mathbf{e}_{eq} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} \quad (\text{A.8})$$

et

$$\theta = \arctan\left(\frac{e_y}{e_x}\right) \quad \phi = \arctan\left(\frac{e_z}{e_x^2 + e_y^2}\right)$$

La matrice $\mathbf{R}(\mathbf{e}_{eq}, \alpha)$ est définie par :

$$\mathbf{R}(\mathbf{e}_{eq}, \alpha) = \begin{bmatrix} e_x^2 v(\alpha) + \cos(\alpha) & e_x e_y v(\alpha) - e_z \sin(\alpha) & e_x e_z v(\alpha) + e_y \sin(\alpha) \\ e_x e_y v(\alpha) + e_z \sin(\alpha) & e_y^2 v(\alpha) + \cos(\alpha) & e_y e_z v(\alpha) - e_x \sin(\alpha) \\ e_x e_z v(\alpha) - e_y \sin(\alpha) & e_y e_z v(\alpha) + e_x \sin(\alpha) & e_z^2 v(\alpha) + \cos(\alpha) \end{bmatrix} \quad (\text{A.9})$$

avec $v(\alpha) = 1 - \cos(\alpha)$.

Dans le cas où nous connaissons la matrice $\mathbf{R}(\mathbf{e}_{eq}, \alpha)$, nous pouvons trouver α et \mathbf{e}_{eq} en utilisant :

$$\alpha = \arccos \left(\frac{\text{trace}(\mathbf{R}(\mathbf{e}_{eq}, \alpha)) - 1}{2} \right) \quad (\text{A.10})$$

$$\mathbf{e}_{eq} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = \frac{1}{\sin \alpha} \begin{bmatrix} R_{32}(\mathbf{e}_{eq}, \alpha) - R_{23}(\mathbf{e}_{eq}, \alpha) \\ R_{13}(\mathbf{e}_{eq}, \alpha) - R_{31}(\mathbf{e}_{eq}, \alpha) \\ R_{21}(\mathbf{e}_{eq}, \alpha) - R_{12}(\mathbf{e}_{eq}, \alpha) \end{bmatrix} \quad (\text{A.11})$$

Nous pouvons faire deux remarques importantes : la première est fondée sur le fait que la rotation d'angle α autour d'une axe \mathbf{e}_{eq} peut aussi être exprimée comme la rotation d'angle $-\alpha$ autour de l'axe $-\mathbf{e}_{eq}$. La deuxième concerne le problème du conditionnement de l'équation A.11 pour $\alpha = 0$ et $\alpha = \pi$.

Il est aussi possible de définir une rotation par trois éléments en considérant le vecteur :

$$\mathbf{e}_a = \alpha \mathbf{e}_{eq}$$

A.2.5 Quaternion

Alors que la représentation d'une rotation par un ensemble de trois angles présente l'inconvénient de posséder des *singularités*, la représentation par une matrice de rotation n'est pas minimale. Un quaternion unitaire utilise quatre paramètres pour représenter une rotation sans les problèmes de singularités, ce qui constitue un choix intéressant.

Nous considérons les trois variables complexes \mathbf{i} , \mathbf{j} et \mathbf{k} , telles que :

$$\begin{aligned} \mathbf{i}^2 &= \mathbf{j}^2 = \mathbf{k}^2 = -1 \\ \mathbf{ij} &= \mathbf{k} \quad \mathbf{jk} = \mathbf{i} \quad \mathbf{ki} = \mathbf{j} \end{aligned}$$

Nous définissons le quaternion \mathbf{Q} :

$$\mathbf{Q} = n + \mathbf{ii} + \mathbf{jj} + \mathbf{kk} \quad (\text{A.12})$$

Sous forme vectorielle :

$$\mathbf{Q} = \begin{bmatrix} n \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} n \\ i \\ j \\ k \end{bmatrix} \quad (\text{A.13})$$

Nous associons au quaternion une matrice carré d'ordre quatre, telle que

$$\mathcal{Q} = \begin{bmatrix} n & -i & -j & -k \\ i & n & -k & j \\ j & k & n & -i \\ k & -j & i & n \end{bmatrix} \quad (\text{A.14})$$

Pour représenter les rotations, nous ne considérons que les quaternions de module 1.

Historiquement, les quaternions unitaires utilisés pour représenter une rotation ont été appelés *paramètres d'Euler*, *paramètres de Rodrigues* ou *paramètres de Rodrigues-Hamilton*.

Relation avec la représentation angle plus vecteur

Le quaternion \mathbf{Q} correspondant à la rotation d'angle α autour de l'axe \mathbf{e}_{eq} est définie par :

$$\mathbf{Q} = \begin{bmatrix} n \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} n \\ i \\ j \\ k \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ e_x \sin\left(\frac{\alpha}{2}\right) \\ e_y \sin\left(\frac{\alpha}{2}\right) \\ e_z \sin\left(\frac{\alpha}{2}\right) \end{bmatrix} \quad (\text{A.15})$$

De la même façon, l'angle de rotation α et l'axe équivalent \mathbf{e}_{eq} sont obtenus à partir d'un quaternion par :

$$\eta = \|\mathbf{q}\| \quad \alpha = 2 \cos^{-1} n = 2 \tan^{-1} \frac{\eta}{n} \quad \mathbf{e}_{eq} = \frac{1}{\eta} \mathbf{q} \quad (\text{A.16})$$

Comme pour la représentation d'une rotation par un angle et un vecteur, il existe deux quaternions qui représentent la même rotation : $[n \ i \ j \ k]^T$ et $[-n \ -i \ -j \ -k]^T$.

Inverse du quaternion

On appelle quaternion conjugué du quaternion $\mathbf{Q} = [n \ i \ j \ k]^T$, le quaternion $[n \ -i \ -j \ -k]^T$. Il correspond à la rotation inverse :

$$\mathbf{Q}^{-1} = \begin{bmatrix} n \\ -\mathbf{q} \end{bmatrix} = \begin{bmatrix} n \\ -i \\ -j \\ -k \end{bmatrix} \quad (\text{A.17})$$

Composition de rotations par quaternions

La composition de rotations en utilisant les quaternions est définie par le produit de quaternions. Pour deux quaternions \mathbf{Q}_1 et \mathbf{Q}_2 , le produit est défini par :

$$\mathbf{Q}_1 \mathbf{Q}_2 = \begin{bmatrix} n_1 n_2 - \mathbf{q}_1^T \mathbf{q}_2 \\ n_1 \mathbf{q}_2 + n_2 \mathbf{q}_1 + \mathbf{q}_1 \times \mathbf{q}_2 \end{bmatrix} = \begin{bmatrix} n_1 n_2 - i_1 i_2 - j_1 j_2 - k_1 k_2 \\ n_1 i_2 + n_2 i_1 + j_1 k_2 - k_1 j_2 \\ n_1 j_2 + n_2 j_1 + k_1 i_2 - i_1 k_2 \\ n_1 k_2 + n_2 k_1 + i_1 j_2 - j_1 i_2 \end{bmatrix} \quad (\text{A.18})$$

Rotation relative entre quaternions

Considérons deux quaternions ${}^O\mathbf{Q}_1$ et ${}^O\mathbf{Q}_2$, la rotation relative ${}^1\mathbf{Q}_2$ en considérant A.5 et A.17 est définie par :

$${}^1\mathbf{Q}_2 = \mathbf{Q}_1^{-1} \mathbf{Q}_2 = \begin{bmatrix} n_1 n_2 + \mathbf{q}_1^T \mathbf{q}_2 \\ n_1 \mathbf{q}_2 - n_2 \mathbf{q}_1 - \mathbf{q}_1 \times \mathbf{q}_2 \end{bmatrix} = \begin{bmatrix} n_1 n_2 + i_1 i_2 + j_1 j_2 + k_1 k_2 \\ n_1 i_2 - n_2 i_1 - j_1 k_2 + k_1 j_2 \\ n_1 j_2 - n_2 j_1 - k_1 i_2 + i_1 k_2 \\ n_1 k_2 - n_2 k_1 - i_1 j_2 + j_1 i_2 \end{bmatrix} \quad (\text{A.19})$$

Conversions entre matrice de rotation et quaternions

La matrice \mathbf{R} est obtenue à partir du quaternion \mathbf{Q} par :

$$\mathbf{R} = \begin{bmatrix} 1 - 2j^2 - 2k^2 & 2ij + 2nk & 2ik - 2nj \\ 2ij - 2nk & 1 - 2i^2 - 2k^2 & 2jk + 2ni \\ 2ik + 2nj & 2jk - 2ni & 1 - 2i^2 - 2j^2 \end{bmatrix} \quad (\text{A.20})$$

Une méthode indirecte pour obtenir le quaternion à partir d'une matrice de rotation, consiste à obtenir l'angle α et l'axe équivalent e_{eq} à partir de la matrice de rotation puis à construire le quaternion correspondant.

Shoemake [Shoemake, 1985] propose l'algorithme 2

Algorithme 2 : Calcul du quaternion à partir d'une matrice de rotation

```

début
  Définition de  $\varepsilon$ 
  Calcul  $n^2 = \frac{1+R_{11}+R_{22}+R_{33}}{4}$ 
  si  $n^2 > \varepsilon$  alors
     $n = \sqrt{n^2}$ 
     $i = \frac{R_{23}-R_{32}}{4n}$ 
     $j = \frac{R_{31}-R_{13}}{4n}$ 
     $k = \frac{R_{12}-R_{21}}{4n}$ 
  sinon
     $n = 0$ 
     $i^2 = -\frac{R_{22}-R_{33}}{2}$ 
    si  $i^2 > \varepsilon$  alors
       $i = \sqrt{i^2}$ 
       $j = \frac{R_{12}}{2i}$ 
       $k = \frac{R_{13}}{2i}$ 
    sinon
       $i = 0$ 
       $j^2 = \frac{1-R_{33}}{2}$ 
      si  $j^2 > \varepsilon$  alors
         $j = \sqrt{j^2}$ 
         $k = \frac{R_{23}}{2j}$ 
      sinon
         $j = 0$ 
         $k = 1$ 
      fin
    fin
  fin
fin

```

A.3 Représentation de la situation

Pour définir la situation d'un objet, nous avons besoin d'*au moins* six paramètres indépendants :

- trois paramètres définissent la position d'un point de l'objet ;
- au moins trois paramètres qui définissent l'orientation de l'objet autour de ce point.

Nous utiliserons deux types de représentations de la situation, un vecteur de coordonnées opérationnelles et la matrice de passage homogène.

A.3.1 Vecteur de coordonnées opérationnelles

Pour éviter le problème des singularités dans la représentation de l'orientation, nous utiliserons une représentation non minimale à sept composantes. Le vecteur de coordonnées opérationnelles \mathbf{X} est com-

posé des trois composantes du vecteur de position et des quatre composantes du quaternion :

$$\mathbf{X} = \begin{bmatrix} \mathbf{P} \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ n \\ i \\ j \\ k \end{bmatrix} \quad (\text{A.21})$$

A.3.2 Matrice de passage homogène

La situation d'un repère \mathcal{R}_{n+i} par rapport au repère \mathcal{R}_n peut être représentée par

1. le vecteur de position \mathbf{P} de l'origine du repère \mathcal{R}_{n+i} par rapport à l'origine du repère \mathcal{R}_n ;
2. la matrice de rotation \mathbf{R} du repère \mathcal{R}_{n+i} par rapport au repère \mathcal{R}_n .

Les deux sont pris en compte simultanément en utilisant une matrice d'ordre quatre appelée *Matrice de Passage Homogène*, ou très couramment *Matrice Homogène*, de la forme :

$${}^n\mathbf{T}_{n+1} = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ 0 & 1 \end{bmatrix} \quad (\text{A.22})$$

L'inverse de la matrice homogène ${}^n\mathbf{T}_{n+1}$ est défini par :

$$({}^n\mathbf{T}_{n+1})^{-1} = {}^{n+1}\mathbf{T}_n = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{P} \\ 0 & 1 \end{bmatrix} \quad (\text{A.23})$$

A.4 Vecteur vitesse instantanée de rotation

Nous considérons la rotation d'un point défini par un *vecteur de position* \mathbf{P} autour d'un axe fixe.

Considérons un repère fixe \mathcal{R}_a et un repère mobile \mathcal{R}_b . Les coordonnées d'un point fixe par rapport au repère au \mathcal{R}_b , s'expriment dans le repère \mathcal{R}_a par :

$$\mathbf{P}_a = {}^b\mathbf{R}_a\mathbf{P}_b$$

La dérivée par rapport au temps est

$$\dot{\mathbf{P}}_a = {}^b\dot{\mathbf{R}}_a\mathbf{P}_b$$

car \mathbf{P}_b est fixe par rapport à \mathcal{R}_b .

Par définition, la *Vitesse de translation* est la dérivée de la position par rapport au temps :

$$\mathbf{V} \triangleq \dot{\mathbf{P}}$$

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \triangleq \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

et

$$\mathbf{V}_a = {}^b\dot{\mathbf{R}}_a({}^a\mathbf{R}_b\mathbf{P}_a) \quad (\text{A.24})$$

Considérons la figure A.4.

La vitesse de translation \mathbf{V} du point \mathbf{P} est définie par

$$\mathbf{V} = \boldsymbol{\omega} \times \mathbf{P} \quad (\text{A.25})$$

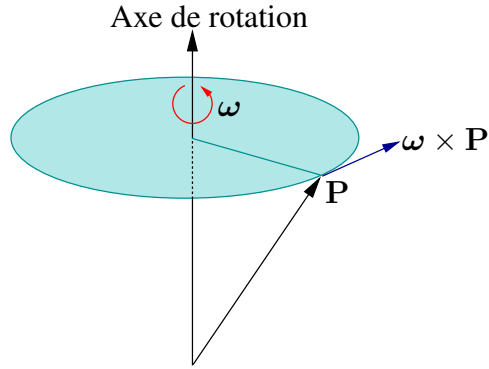


FIG. A.6 – Vitesse de Rotation autour d'un axe

où ω représente le *Vecteur instantané de rotation* qui a la forme

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

de façon matricielle nous pouvons écrire :

$$\mathbf{V} = \omega^\times \mathbf{P}$$

où ω^\times est appelée *matrice de pré-produit vectoriel*

$$\omega^\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (\text{A.26})$$

En considérant A.24 et A.25, nous obtenons :

$$\omega^\times = {}^b \dot{\mathbf{R}}_a {}^b \mathbf{R}_a^{-1} \quad (\text{A.27})$$

d'où

$${}^b \dot{\mathbf{R}}_a = \omega^\times {}^b \mathbf{R}_a \quad (\text{A.28})$$

A.4.1 Le quaternion et le vecteur de vitesse instantanée de rotation

La dérivée temporelle d'un quaternion est lié au vecteur instantané de rotation de la façon suivante :

$$\dot{\mathbf{Q}} = \frac{1}{2} \Omega \mathbf{Q} \quad (\text{A.29})$$

où Ω :

$$\Omega = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega^T \\ \omega & -\omega^\times \end{bmatrix} \quad (\text{A.30})$$

Et l'inverse de cette fonction est donné par

$$\begin{bmatrix} \omega \\ 0 \end{bmatrix} = -2\mathbf{Q}^T \dot{\mathbf{Q}} = 2\dot{\mathbf{Q}}\mathbf{Q}^{-1} \quad (\text{A.31})$$

Annexe B

Modèles pour un robot 6R

B.1 Modèle Géométrique Direct

Le modèle géométrique direct d'un robot manipulateur est la fonction \mathbf{F} qui permet d'exprimer la situation de l'organe terminal du robot manipulateur en fonction de sa configuration. Le modèle géométrique du robot manipulateur est :

$$\mathbf{X} = \mathbf{F}(\theta) \quad (\text{B.1})$$

La technique la plus répandue pour modéliser un robot manipulateur de type 6R consiste à utiliser les paramètres de Denavit-Hartenberg.

Soit un bras manipulateur constitué par n corps mobiles supposés parfaitement rigides suivant une structure de chaîne cinématique simple ouverte reliés entre eux par n liaisons rotoïdes.

Deux corps extrêmes ont des rôles particuliers :

- la *base* qui est fixée au sol ou sur un véhicule. On associe un repère à la base \mathcal{R}_O .
- l'*organe terminal* qui porte l'outil. On associe un repère à l'organe terminal \mathcal{R}_{OT} .

La figure B.1 indique les coordonnées et les repères d'un robot manipulateur de type 6R.

On associe au i -ème corps le repère \mathcal{R}_i , pour $i = 1, \dots, n + 1$. Le repère \mathcal{R}_{n+1} correspond au centre de la poignet (\mathcal{R}_p) qui est lié par une transformation rigide au repère de l'organe terminal du bras manipulateur \mathcal{R}_{OT} .

On note de plus ${}^{i-1}\mathbf{T}_i$ la matrice homogène qui permet d'exprimer les coordonnées d'un point ou d'un vecteur du repère \mathcal{R}_i dans le repère \mathcal{R}_{i-1} . En utilisant les paramètres de Denavit-Hartenberg, la matrice homogène est définie par :

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \cos(\alpha_{i-1}) \sin(\theta_i) & \cos(\alpha_{i-1}) \cos(\theta_i) & -\sin(\alpha_{i-1}) & -r_i \sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1}) \sin(\theta_i) & \sin(\alpha_{i-1}) \cos(\theta_i) & \cos(\alpha_{i-1}) & -r_i \cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.2})$$

En réalisant la composition des transformations du repère de la base \mathcal{R}_O jusqu'au repère du poignet \mathcal{R}_p .

$${}^0\mathbf{T}_P = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_4 {}^4\mathbf{T}_5 {}^5\mathbf{T}_P \quad (\text{B.3})$$

Et pour l'organe terminal

$${}^0\mathbf{T}_{OT} = {}^0\mathbf{T}_P {}^P\mathbf{T}_{OT} \quad (\text{B.4})$$

La Situation de l'organe terminal S_{OT} est défini par la position obtenu directement de ${}^0\mathbf{T}_{OT}$, pendant

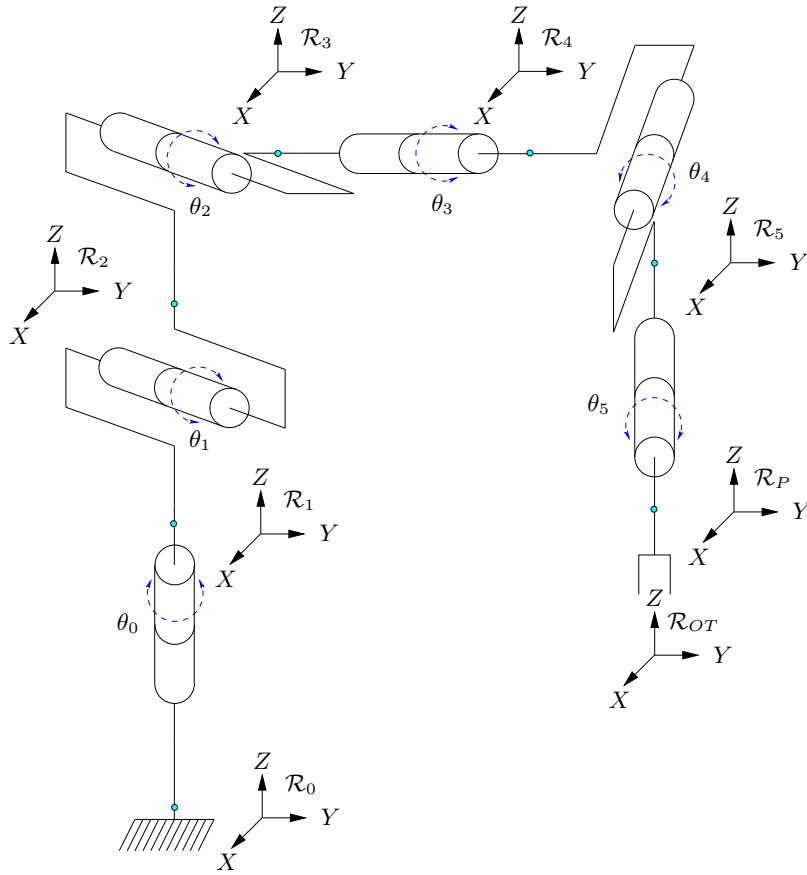


FIG. B.1 – Définition des Coordonnées Généralisées et des repères d'un robot manipulateur de type 6R.

que l'orientation est obtenu de la conversion de la matrice de rotation \mathbf{R} de ${}^0\mathbf{T}_{OT}$ à quaternion. Tel que

$$\mathcal{S}_{OT} = \begin{bmatrix} \mathbf{P} \\ \mathbf{Q} \end{bmatrix}_{OT} = \begin{bmatrix} x \\ y \\ z \\ n \\ i \\ j \\ k \end{bmatrix}_{OT} \quad (\text{B.5})$$

B.2 Modèle Cinématique Direct

Le *Modèle Cinématique Direct* permet de calculer les composantes du torseur cinématique $\dot{\mathbf{X}}$ à partir de la donnée des vitesses articulaires dites généralisées $\dot{\theta}$, dérivées par rapport au temps des coordonnées généralisées θ . Le torseur cinématique est défini par

$$\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} \quad (\text{B.6})$$

Le Modèle Cinématique Direct fait intervenir la matrice *jacobienne*, fonction de la configuration du

robot manipulateur, tel que :

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad (\text{B.7})$$

B.2.1 Calcul Indirect de la matrice Jacobienne

Le calcul indirect de la matrice jacobienne consiste à utiliser le modèle géométrique du robot manipulateur.

$$\mathbf{X} = \mathbf{F}(\boldsymbol{\theta}) \quad (\text{B.8})$$

Et par définition, la matrice jacobienne est la matrice des dérivées partielles de la fonction \mathbf{F} par rapport aux coordonnées généralisées, ainsi

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) \quad (\text{B.9})$$

B.2.2 Calcul direct de la matrice Jacobienne

En considérant que le torseur cinématique pour une liaison rotoïde est défini par

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} \times \mathbf{P} \\ \boldsymbol{\omega} \end{bmatrix} \quad (\text{B.10})$$

Si la rotation est définie autour de l'axe unitaire z , en utilisant la dérivée de la variable articulaire

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \dot{\theta}z \times \mathbf{P} \\ \dot{\theta}z \end{bmatrix} \quad (\text{B.11})$$

Le torseur cinématique de l'organe terminal est obtenu par composition des torseurs des corps constituant la chaîne cinématique. Le théorème de superposition permet d'exprimer le torseur cinématique de l'organe terminal comme la somme de ceux engendrés aux différentes liaisons :

$$\begin{bmatrix} \mathbf{V}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{i-1} + \dot{\theta}_i z_i \times \mathbf{P}_i \\ \boldsymbol{\omega}_{i-1} + \dot{\theta}_i z_i \end{bmatrix} \quad (\text{B.12})$$

Finalement, le torseur cinématique pour l'organe terminal est donné par

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix}_n = \begin{bmatrix} \sum_{i=1}^n \dot{\theta}_i z_i \times \mathbf{P}_i \\ \sum_{i=1}^n \dot{\theta}_i z_i \end{bmatrix} \quad (\text{B.13})$$

et de façon matricielle

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad (\text{B.14})$$

Dans [Orin et al., 1983], différentes méthodes sont traitées pour le calcul de la matrice jacobienne.

Annexe C

Stabilité par Lyapunov

Le Modèle Cinématique Direct est défini par :

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}\dot{\boldsymbol{\theta}} \quad (\text{C.1})$$

Le Modèle Cinématique Inverse est défini par

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1} \begin{bmatrix} \mathbf{V} \\ \boldsymbol{\omega} \end{bmatrix} \quad (\text{C.2})$$

Dans une commande en boucle fermée [Wu et al., 1982], la loi de commande est remplacée par

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1} \begin{bmatrix} \mathbf{V}_d - \mathbf{K}_P \mathbf{e}_P \\ \boldsymbol{\omega}_d - \mathbf{K}_O \mathbf{e}_O \end{bmatrix} \quad (\text{C.3})$$

où \mathbf{K}_P et \mathbf{K}_O sont des matrices de gain, et \mathbf{e}_P et \mathbf{e}_O représentent respectivement les vecteurs d'erreur de position et d'orientation.

C.1 Limite de Stabilité pour la position

La fonction de Lyapunov \mathcal{V}_P est défini comme

$$\mathcal{V}_P = \frac{1}{2}(\mathbf{P} - \mathbf{P}_d)^T(\mathbf{P} - \mathbf{P}_d)$$

En dérivant par rapport au temps

$$\dot{\mathcal{V}}_P = \frac{1}{2} \left((\mathbf{P} - \mathbf{P}_d)^T(\dot{\mathbf{P}} - \dot{\mathbf{P}}_d) + (\dot{\mathbf{P}} - \dot{\mathbf{P}}_d)^T(\mathbf{P} - \mathbf{P}_d) \right)$$

en utilisant

$$\mathbf{V} \triangleq \dot{\mathbf{P}}$$

Nous obtenons

$$\dot{\mathcal{V}}_P = \frac{1}{2} \left((\mathbf{P} - \mathbf{P}_d)^T(\mathbf{V} - \mathbf{V}_d) + (\mathbf{V} - \mathbf{V}_d)^T(\mathbf{P} - \mathbf{P}_d) \right)$$

$$\dot{\mathcal{V}}_P = \frac{1}{2} (2\mathbf{P}^T\mathbf{V} - 2\mathbf{P}_d^T\mathbf{V} - 2\mathbf{P}^T\mathbf{V}_d + 2\mathbf{P}_d^T\mathbf{V}_d)$$

$$\dot{\mathcal{V}}_P = (\mathbf{P} - \mathbf{P}_d)^T(\mathbf{V} - \mathbf{V}_d)$$

Des équations C.2 et C.3 nous obtenons

$$\mathbf{V} = \mathbf{V}_d - \mathbf{K}_P \mathbf{e}_P$$

tel que

$$\mathbf{V} - \mathbf{V}_d = -\mathbf{K}_P \mathbf{e}_P$$

donc

$$\dot{\mathbf{V}}_P = -(\mathbf{P} - \mathbf{P}_d)^T \mathbf{K}_P \mathbf{e}_P$$

En faisant,

$$\mathbf{e}_P = \mathbf{P} - \mathbf{P}_d$$

La stabilité du système est assurée en choisissant \mathbf{K}_P comme une matrice positive définie.

C.2 Limite de Stabilité pour l'Orientation

Étant le quaternion défini par

$$\mathbf{Q} = \begin{bmatrix} n \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} n \\ i \\ j \\ k \end{bmatrix}$$

et la dérivé du quaternion défini par

$$\dot{\mathbf{Q}} = \frac{1}{2} \begin{bmatrix} \dot{n} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -\boldsymbol{\omega}^\times \end{bmatrix} \begin{bmatrix} n \\ \mathbf{q} \end{bmatrix} \quad (\text{C.4})$$

tel que pour le quaternion de consigne \mathbf{Q}_d

$$\dot{\mathbf{Q}}_d = \frac{1}{2} \begin{bmatrix} \dot{n}_d \\ \dot{\mathbf{q}}_d \end{bmatrix} = \begin{bmatrix} 0 & -\boldsymbol{\omega}_d^T \\ \boldsymbol{\omega}_d & -\boldsymbol{\omega}_d^\times \end{bmatrix} \begin{bmatrix} n_d \\ \mathbf{q}_d \end{bmatrix} \quad (\text{C.5})$$

La fonction de Lyapunov \mathcal{V}_O est défini comme

$$\mathcal{V}_O = (n - n_d)^2 + (\mathbf{q} - \mathbf{q}_d)^T (\mathbf{q} - \mathbf{q}_d)$$

En dérivant par rapport au temps

$$\dot{\mathcal{V}}_O = 2(n - n_d)(\dot{n} - \dot{n}_d) + 2(\mathbf{q} - \mathbf{q}_d)^T (\dot{\mathbf{q}} - \dot{\mathbf{q}}_d)$$

$$\dot{\mathcal{V}}_O = 2(n\dot{n} - n\dot{n}_d - n_d\dot{n} + n_d\dot{n}_d + \mathbf{q}^T \dot{\mathbf{q}} - \mathbf{q}^T \dot{\mathbf{q}}_d - \mathbf{q}_d^T \dot{\mathbf{q}} + \mathbf{q}_d^T \dot{\mathbf{q}}_d)$$

en utilisant les équations C.4 et C.5

$$\dot{\mathcal{V}}_O = n\mathbf{q}_d^T \boldsymbol{\omega}_d - n\mathbf{q}_d^T \boldsymbol{\omega} + n_d \mathbf{q}^T \boldsymbol{\omega} - n_d \mathbf{q}^T \boldsymbol{\omega}_d - \mathbf{q}^T \boldsymbol{\omega} \times \mathbf{q} + \mathbf{q}^T \boldsymbol{\omega}_d \times \mathbf{q}_d + \mathbf{q}_d^T \boldsymbol{\omega} \times \mathbf{q} - \mathbf{q}_d^T \boldsymbol{\omega}_d \times \mathbf{q}_d$$

$$\dot{\mathcal{V}}_O = n_d \mathbf{q}^T (\boldsymbol{\omega} - \boldsymbol{\omega}_d) - n \mathbf{q}_d^T (\boldsymbol{\omega} - \boldsymbol{\omega}_d) - (\mathbf{q}_d \times \mathbf{q})^T (\boldsymbol{\omega} - \boldsymbol{\omega}_d)$$

$$\dot{\mathcal{V}}_O = (n_d \mathbf{q} - n \mathbf{q}_d - \mathbf{q}_d \times \mathbf{q})^T (\boldsymbol{\omega} - \boldsymbol{\omega}_d)$$

Des équations C.2 et C.3 nous obtenons

$$\boldsymbol{\omega} = \boldsymbol{\omega}_d - \mathbf{K}_O \mathbf{e}_O$$

tel que

$$\boldsymbol{\omega} - \boldsymbol{\omega}_d = -\mathbf{K}_O \mathbf{e}_O$$

donc

$$\dot{\mathcal{V}}_O = -(n_d \mathbf{q} - n \mathbf{q}_d - \mathbf{q}_d \times \mathbf{q})^T \mathbf{K}_O \mathbf{e}_O$$

En faisant,

$$\mathbf{e}_O = n_d \mathbf{q} - n \mathbf{q}_d - \mathbf{q}_d \times \mathbf{q}$$

La stabilité du système est assurée en choisissant \mathbf{K}_O comme une matrice positive définie.

En utilisant la même stratégie, c'est facile à montrer que si la rotation est exprimée par le quaternion conjugué, l'erreur d'orientation est défini par

$$\mathbf{e}_O = n_d \mathbf{q} - n \mathbf{q}_d + \mathbf{q}_d \times \mathbf{q}$$

Références bibliographiques

- [Aken et al., 1988] I. L. V. Aken et H. V. Brussel (1988). On-line robot trajectory control in joint coordinates by means of imposed accélération profiles. *Robotica*, 6 :185–195.
- [Alami et al., 1998] R. Alami, R. Chatila, S. Fleury, M. Ghallab, et F. Ingrand (1998). An architecture for autonomy. *International Journal of Robotics Research*, 17 :315–337.
- [Albu-Schäffer et al., 2007] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, et G. Hirzinger (2007). The DLR lightweight robot - design and control concepts for robots in human environments. *Industrial Robot : An International Journal*, 34 :376–385.
- [Aleotti et al., 2005] J. Aleotti, S. Caselli, et G. Mecerrozzi (2005). Trajectory reconstruction with nurbs curves for robot programming by demonstration. Dans *IEEE Proc. Int. Symp. on Computational Intelligence in Robotics and Automation*, pages 73–78, Finland.
- [Amirabdollahian et al., 2002] F. Amirabdollahian, R.Loureiro, et W. Harwin (2002). Minimum jerk trajectory control for rehabilitation and haptic applications. Dans *IEEE International Conference on Robotics and Automation*, pages 3380–3385, May.
- [Andersson, 1989] R. L. Andersson (1989). Aggressive trajectory generator for a robot ping-pong player. *IEEE Control Systems Magazine*, 9 :15–20.
- [Andreff, 1997] N. Andreff (1997). Towards the embedding of on-line hand-eye calibration into visual servoing. Dans *IEEE/RSJ/INRIA Workshop On New Trends in Image based Robot Servoing*.
- [Andreff et al., 1999] N. Andreff, R. Horaud, et B. Espiau (1999). On-line hand-eye calibration. Dans *Second Int. Conf. on 3D Digital Imaging and Modeling*, Ottawa, Canada.
- [Bazaz et al., 1997] S. A. Bazaz et B. Tondu (1997). Optimal on-line robot trajectory generation in cartesian space. Dans *SPIE Proc. Architectures, Networks and Intelligent Systems for Manufacturing Integration*, volume 3203, pages 2–10.
- [Bazaz et al., 1998] S. A. Bazaz et B. Tondu (1998). 3-cubic spline for on-line cartesian space trajectory planning of an industrial manipulator. Dans *Int. Workshop on Advanced Motion Control*, pages 493–498, Coimbra.
- [Bellot, 2002] D. Bellot (2002). *Contribution à l'analyse et à la synthèse de schémas de commande référencée vision*. Thèse de doctorat, Université Toulouse III, LAAS - CNRS, France.
- [Bestaoui, 1989] Y. Bestaoui (1989). On line motion generation with velocity and acceleration constraints. *Robotics and Autonomous Systems*, 5 :279–288.
- [Bobrow, 1988] J. E. Bobrow (1988). Optimal robot path planning using the minimum-time criterion. *IEEE J. of Robotics and Automation*, 4 :443–450.
- [Bobrow et al., 1985] J. E. Bobrow, S. Dubowsky, et J. S. Gibson (1985). Time optimal control of robotic manipulators along specified paths. *The Int. J. of Robotics Research*, 4 :3–17.
- [Brady et al., 1982] M. Brady, J. Hollerbach, T. Johnson, et T. Lozano-Perez (1982). *Robot Motion, Planning and Control*. The MIT Press, Cambridge, Massachusetts.
- [Brock et al., 1997] O. Brock et O. Khatib (1997). Elastic strip : Real-time path modification for mobile manipulation. Dans *The eight International Symposium of Robotics Research*, Hayama, Japan.

- [Bruyninckx et al., 2001] H. Bruyninckx et J. Shutter (2001). Introduction to intelligent robotics. Rapport technique, Katholieke Universiteit de Leuven.
- [Buss, 2004] S. R. Buss (2004). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Rapport technique, University of California.
- [Canudas et al., 1997] C. Canudas, B. Siciliano, et G. Bastin (1997). *Theory of Robot Control*. Springer.
- [Castain et al., 1984] R. H. Castain et R. P. Paul (1984). An on-line dynamic trajectory generator. *The International Journal of Robotics Research*, 3 :68–72.
- [Chand et al., 1985] S. Chand et K. L. Doty (1985). On-line polynomial trajectories for robot manipulators. *The Int. J. of Robotics Research*, 4 :38–48.
- [Chaumette, 1990] F. Chaumette (1990). *La relation vision-commande : Théorie et applications à des tâches robotiques*. Thèse de doctorat, Université de Rennes I, IRISA, France.
- [Chaumette, 1998] F. Chaumette (1998). De la perception à l'action : l'asservissement visuel, de l'action à la perception : la vision active. Rapport technique, Habilitation à diriger des Recherches, Université de Rennes I, France.
- [Chaumette et al., 2006] F. Chaumette et S. Hutchinson (2006). Visual servo control. part i : Basic approaches. *IEEE Robotics and Automation Magazine*, pages 82–90.
- [Chaumette et al., 2007] F. Chaumette et S. Hutchinson (2007). Visual servo control. part ii : Advanced approaches. *IEEE Robotics and Automation Magazine*, pages 109–118.
- [Chaumette et al., 1991] F. Chaumette, P. Rives, et B. Espiau (1991). Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. Dans *IEEE Proc. Int. Conf. on Robotics and Automation*, pages 2248–2253, Sacramento.
- [Chen et al., 1989] Y. Chen et A. Desrochers (1989). Minimum-time control laws for robotic manipulators. Dans *Proc. of the 28th Conference on Decision and Control*, pages 2494–2499, Tampa, Fl.
- [Damian, 2006] E. L. Damian (2006). *Grasp planning for object manipulation by an autonomous robot*. Thèse de doctorat, INSA Toulouse - LAAS CNRS, France.
- [De Luca et al., 2006] A. De Luca, A. Albu-Schäffer, S. Haddadin, et G. Hirzinger (2006). Collision detection and safe reaction with the DLR-III lightweight manipulator arm.
- [Durrant-White et al., 2006] H. Durrant-White et T. Bailey (2006). Tutorial : Simultaneous localization and mapping : Part i. *IEEE Robotics & Automation Magazine*, pages 99–108.
- [Egeland et al., 1988] O. Egeland et E. Lunde (1988). Trajectory generation for manipulators using linear quadratic optimal tracking. Dans *IEEE*, pages 376–381.
- [Er, 1993] M. J. Er (1993). Recent developments and futuristic trends in robot manipulator control. Dans *Asia-Pacific Workshop on Advances in Motion Control Proceedings*, pages 106–111.
- [Feddema et al., 1989] J. T. Feddema et O. R. Mithcell (1989). Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5 :691–700.
- [Flash et al., 1985] T. Flash et N. Hogan (1985). The coordination of arm movements : An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5 :1688–1703.
- [Fleury et al., 1997] S. Fleury, M. Herrb, et R. Chatila (1997). Genom : A tool for the specification and the implementation of operating modules in a distributed robot architecture. Dans *IEEE Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 842–848, Grenoble, France.
- [Fleury et al., 1995] S. Fleury, P. Souères, J. P. Laumond, et R. Chatila (1995). Primitives for smoothing mobile robot trajectories. *IEEE. Trans. on Robotics and Automation*, 11(3) :441–448.
- [Foxlin, 2002] E. M. Foxlin (2002). Generalized architecture for simultaneous localization, auto-calibration and map-building. Dans *IEEE Proc. Int. Conf. on Robots and Systems*, Lausanne, Switzerland.
- [Froissart et al., 1993] C. Froissart et P. Mechler (1993). On-line polynomial path planning in cartesian space for robot manipulators. *Robotica*, 11 :245–251.

-
- [Funda et al., 1991] J. Funda, R. H. Taylor, et R. P. Paul (1991). On homogeneous transforms, quaternions and computational efficiency. *IEEE Trans. on Robotics and Automation*, 6 :382–388.
- [Geering et al., 1986] H. P. Geering, L. Guzzella, S. A. R. Hepner, et C. H. Onder (1986). Time-optimal motions of robots in assembly tasks. *IEEE Transactions on Automatic Control*, AC-31 :512–518.
- [Glattfelder et al., 1983] A. Glattfelder et W. Schaufelberger (1983). Stability analysis of single loop control systems with saturation and antireset-windup circuits. *IEEE Trans. on Automatic Control*, 28 :1074–1081.
- [Gorla et al., 1984] B. Gorla et M. Renaud (1984). *Modèles des Robots Manipulateurs application à leur commande*. Cepadues Editions.
- [Hager, 1997] G. D. Hager (1997). A modular system for robust positioning using feedback from stereo vision. *IEEE Transactions on Robotics and Automation*, 13 :582–595.
- [Hager et al., 1996] G. D. Hager, S. Hutchinson, et P. Corke (1996). Tutorial tt3 : A tutorial on visual servo control. Dans *IEEE Int. Conf. Robotics and Automation*, Minneapolis.
- [Hedjar et al., 2002] R. Hedjar, R. Toumi, P. Boucher, et D. Dumur (2002). Feedback nonlinear predictive control of rigid link robot manipulators. Dans *Proceedings of the American Control Conference*, pages 3594–3599, Anchorage.
- [Hogan, 1984] N. Hogan (1984). An organizing principle for a class of voluntary movements. *The Journal of Neuroscience*, 4 :2745–2754.
- [Horaud et al., 1993] R. Horaud et O. Monga (1993). *Vision par ordinateur : outils fondamentaux*. Editions Hermes.
- [Huynh, 1998] F. Huynh (1998). *Manipulation assistée par la vision pour des tâches de poursuite d'un objet mobile et de saisie d'un objet fixe*. Thèse de doctorat, Université Toulouse III, LAAS CNRS, France.
- [Kahn et al., 1971] M. E. Kahn et B. E. Roth (1971). The near-minimum-time control of open-loop articulated kinematic chains. *Trans. ASME J. Dynamic, Systems, Meas., Cont.*, 93 :164–172.
- [Khalil, 2002] W. Khalil (2002). *Commande des robots manipulateurs*. Hermes Science - Lavoisier.
- [Khalil et al., 1999] W. Khalil et E. Dombre (1999). *Modélisation, identification et commande des robots*. Hermes Science Publications, Paris.
- [Kim et al., 1985] B. K. Kim et K. G. Shin (1985). Minimum-time path planning for robot arms and their dynamics. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-15 :213–223.
- [Klaus et al., 1986] B. Klaus et P. Horn (1986). *Robot vision*. The MIT Press.
- [Kragic et al., 2002] D. Kragic et H. Christensen (2002). Survey on visual servoing for manipulation. Rapport Technique ISRN KTHA/NA/P-02/01-SE, Computational Vision and Active Perception Laboratory, Fiskartorpsv 15 A 10044 Stockholm, Sweden.
- [Kyriakopoulos et al., 1988] K. J. Kyriakopoulos et G. N. Saridis (1988). Minimum jerk path generation. Dans *Proc. IEEE International Conference on Robotics and Automation*, pages 364–369.
- [Lambrechts et al., 2004] P. Lambrechts, M. Boerlage, et M. Steinbuch (2004). Trajectory planning and feedforward design for high performance motion systems. Dans *American Control Conference*, Boston.
- [Lamiroux et al., 1997] F. Lamiroux et J. P. Laumond (1997). From paths to trajectories for multi-body mobile robots. Dans *The fifth International Symposium on Experimental Robotics V*, pages 301–309.
- [Latombe, 1991] J. C. Latombe (1991). *Robot Motion Planning*. Kluwer Academic Press.
- [Laumond, 1998] J. P. Laumond (1998). *Lectures Notes in Control and Information Sciences 229. Robot Motion. Planning and Control*. Springer.
- [Lin et al., 1983] C. S. Lin, P. R. Chang, et J. Y. S. Luh (1983). Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Transactions on Automatic Control*, AC-28 :1066–1074.

- [Liu, 2002] S. Liu (2002). An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators. Dans *7th International Workshop on Advanced Motion Control*, pages 365–370.
- [Lloyd et al., 1993] J. Lloyd et V. Hayward (1993). Trajectory generation for sensor-driven and time-varying tasks. *International Journal Robotics Research*, 12 :380–393.
- [Lopez-Damian et al., 2005] E. Lopez-Damian, D. Sidobre, et R. Alami (2005). A grasp planner based on inertial properties. Dans *Proc. IEEE International Conference on Robotics and Automation*, pages 766–771, Barcelona, Spain.
- [Lozano et al., 2001] R. Lozano et D. Taoutaou (2001). *Commande adaptative et applications*. Hermes Science Publications, Paris.
- [Luh et al., 1980] J. Y. S. Luh, M. W. Walker, et R. P. Paul (1980). Resolved-acceleration control of mechanical manipulators. *IEEE Trans. Automatic Control*, AC-25 :468–474.
- [Luong et al., 1996] Q. T. Luong et O. D. Faugeras (1996). The fundamental matrix : theory, algorithmes and stability analysis. *Int. Journal of Computer Vision*, pages 43–76.
- [Macfarlane et al., 2003] S. Macfarlane et E. Croft (2003). Jerk-bounded manipulator trajectory planning : Design for real-time applications. *IEEE Transactions on Robotics and Automation*, 19 :42–52.
- [Mahmoud, 1991] M. Mahmoud (1991). *Computer Operated Systems Control*. Marcel Dekker, Inc.
- [Malis, 1998] E. Malis (1998). *Contributions à la modelisation et la commande en asservissement visuel*. Thèse de doctorat, Université de Rennes, IRISA, France.
- [Malis et al., 1998] E. Malis, F. Chaumette, et S. Boudet (1998). Positioning a coarse-calibrated camera with respect to an unknown object by 2d 1/2 visual servoing. Dans *IEEE Int. Conf. on Robotics and Automation*, pages 1352–1359, Leuven, Belgium.
- [Martin et al., 1999] B. J. Martin et J. E. Bobrow (1999). Minimum-effort motions for open-chain manipulators with task-dependant end-effector constraints. *The International Journal of Robotics Research*, 18 :213–224.
- [McCarthy et al., 1992] J. M. McCarthy et J. E. Bobrow (1992). The number of saturated actuators and constraints forces during time-optimal movement of a general robotic system. *IEEE Transactions Robotics and Automatization*, 8 :407–409.
- [Mezouar et al., 2001] Y. Mezouar et F. Chaumette (2001). Design and tracking of desirable trajectories in the image space by integrating mechanical and visibility constraints. Dans *IEEE Proc. Int. Conf. on Robotics and Automation*, pages 731–736, Seoul, Korea.
- [Müller-Karger et al., 2000] C. M. Müller-Karger, A. L. G. Mirena, et J. T. S. López (2000). Hyperbolic trajectories for pick-and-place operations to elude obstacles. *IEEE Trans. on Robotics and Automation*, 16 :294–300.
- [Nakamura, 1991] Y. Nakamura (1991). *Advanced Robotics. Robotics and Optimization*. Addison-Wesley Publishing Company.
- [Nelson et al., 1996] B. J. Nelson, J. Morrow, et P. H. Kohsla (1996). Robotic manipulation using high bandwidth force and vision feedback. *Mathl. Comput. Modelling*, 24 :11–29.
- [Orin et al., 1983] D. E. Orin et W. W. Schrader (1983). Efficient jacobien determination for robot manipulators. Dans *Proc. First Int. Symp. on Robotics Research*, pages 727–734, Bretton Woods.
- [Ozaki et al., 1996] H. Ozaki et C. J. Lin (1996). Optimal b-spline joint trajectory generation for collision free movements of a manipulator under dynamic constraints. Dans *IEEE Proc. Int. Conf. on Robotics and Automation*, pages 3592–3597, Minneapolis.
- [Paul, 1979] R. P. Paul (1979). Manipulator cartesian path control. *IEEE Trans. Syst., Man Cybern.*, 9 :702–711.
- [Paul, 1981] R. P. Paul (1981). *Robot Manipulators. Mathematics, Programming and Control*. The MIT Press, The Massachusetts Institute of Technology.

-
- [Pfeiffer et al., 1987] F. Pfeiffer et R. Johanni (1987). A concept for manipulator trajectory planning. *IEEE J. of Robotics and Automation*, RA-3 :115–123.
- [Piazzi et al., 1997] A. Piazzi et A. Visioli (1997). An interval algorithm for minimum-jerk trajectory planning of robot manipulators. Dans *IEEE Conf. on Decision and Control*, pages 1924–1927.
- [Pratt et al., 1995] G. A. Pratt et M. M. Williamson (1995). Series elastic actuators. Dans *Proc.Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 399–406.
- [Reeds et al., 1990] J. A. Reeds et R. A. Shepp (1990). Optimal paths for a car that goes both forward and backwards. Dans *Pacific Journal of Mathematics*, volume 145(2), pages –.
- [Renaud, 1996] M. Renaud (1996). Comment définir l'orientation d'un corps ? quelles relations unissent les dérivées des paramètres d'orientation aux composantes de la vitesse angulaire du corps. Rapport Technique 96078, LAAS - CNRS, 7, Avenue du Colonel Roche, Toulouse, FRANCE.
- [Sahar et al., 1986] G. Sahar et J. M. Hollerbach (1986). Planning of minimum-time trajectories for robots arms. *International Journal of Robotics Research*, 5 :90–100.
- [Samson et al., 1990] C. Samson et B. Espiau (1990). Application of the task-function approach to sensor-based control of robot manipulators. Dans *11th IFAC World Congress*, volume 9, pages 286–291, Tallinn, Estonia.
- [Sciavicco et al., 2000] L. Sciavicco et B. Siciliano (2000). *Modelling and Control of Robot Manipulators*. Springer.
- [Seki et al., 2004] K. Seki et S. Tadakuma (2004). Minimum jerk control of power assisting robot based on human arm behavior characteristic. Dans *International Conference on Systems, Man and Cybernetics*, pages 722–721.
- [Shin et al., 1985] K. G. Shin et N. D. McKay (1985). Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, AC-30 :531–541.
- [Shin et al., 1986] K. G. Shin et N. D. McKay (1986). Selection of near-minimum time geometric paths for robotic manipulators. *IEEE Transactions on Automatic Control*, AC-31 :501–511.
- [Shiu et al., 1989] Y. C. Shiu et S. Ahmad (1989). Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $ax=xb$. *IEEE Trans. on Robotics and Automation*, 5 :16–29.
- [Shoemake, 1985] K. Shoemake (1985). Animating rotation with quaternion curves. *Proc. on Computer graphics and interactive techniques*, 19 :245–254.
- [Sisbot et al., 2006] E. A. Sisbot, L. F. Marin-Urias, R. Alami, et T. Simeon (2006). A mobile robot that performs human acceptable motion. Dans *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China.
- [Slotine, 1985] J. J. E. Slotine (1985). Robust control of robot manipulators. *The International Journal of Robotics Research*, 4 :49–64.
- [Slotine et al., 1987] J. J. E. Slotine et W. Li (1987). Adaptive strategies in constrained manipulation. Dans *Proc. IEEE Int. Conf. on Robotics and Automation*, volume Raleigh, N.C., pages 595–601.
- [Slotine et al., 1989] J. J. E. Slotine et H. S. Yang (1989). Improving the efficiency of time-optimal path-following algorithms. *IEEE Trans. on Robotics and Automation*, 5 :118–124.
- [Solà et al., 2007] J. Solà, A. Monin, et M. Devy (2007). Bicamslam : Two times mono is more than stereo. Dans *IEEE Proc. Int. Conf. on Robotics and Automation*, Rome, Italy.
- [Taylor, 1979] R. H. Taylor (1979). Planning and execution of straight line manipulator trajectories. *IBM Journal of Research and Development*, 23 :424–436.
- [Thompson et al., 1987] S. E. Thompson et R. V. Patel (1987). Formulation of joint trajectories for industrial robots using b-splines. *IEEE Trans. on Industrial Electronics*, IE-34 :192–199.
- [Toussaint et al., 1999] B. Toussaint et S. A. Bazaz (1999). The three-cubic method : An optimal online robot joint trajectory generator under velocity, acceleration and wandering constraints. *Journal of Robotics Systems*, 18(9) :893–901.

- [Toudu et al., 1994] B. Toudu et H. El-Zorkany (1994). Identification of a trajectory generator model for the puma-560 robot. *Journal of Robotics Systems*, 11 :77–90.
- [Tsai et al., 1989] R. Y. Tsai et R. K. Lenz (1989). A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Trans. on Robotics and Automation*, 5 :345–358.
- [Volpe, 1993] R. Volpe (1993). Task space velocity blending for real-time trajectory generation. Dans *IEEE Proc. Int. Conf. on Robotics and Automation*, Atlanta.
- [Wei et al., 1998] G. Q. Wei, K. Arbter, et G. Hirzinger (1998). Active self-calibration of robotic eyes and hand-eye relationships with model identification. *IEEE Trans. on Robotics and Automation*, 14 :158–166.
- [Weiss et al., 1987] L. E. Weiss, A. C. Sanderson, et C. P. Neumann (1987). Dynamic sensor-based control of robots with visual feedback. *IEEE J. of Robotics and Automation*, RA-3 :404–417.
- [Westmore et al., 1991] D. B. Westmore et W. J. Wilson (1991). Direct dynamic control of a robot using an end-point mounted camera and kalman filter position estimation. Dans *IEEE Proc. Int. Conf. on Robotics and Automation*, pages 2376–2384, Sacramento, USA.
- [Whitney, 1969] D. E. Whitney (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Trans. Man-Machine Syst.*, 10 :47–53.
- [Won et al., 1991] J. H. Won, B. W. Choi, et M. J. Chung (1991). Smooth joint trajectory planning for a point to point task. Dans *IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1299–1303, Osaka.
- [Wu et al., 1982] C. H. Wu et R. P. Paul (1982). Resolved motion force control of robot manipulator. *IEEE Trans. Syst., Man Cybern.*, pages 266–275.
- [Yang et al., 1997] Z. Yang et E. Red (1997). On-line cartesian trajectory control of mechanisms along complex curves. *Robotica*, 15 :263–274.
- [Yuan, 1988] J. S. C. Yuan (1988). Closed-loop manipulator control using quaternion feedback. *IEEE J. Robotics and Automotion*, 4 :434–440.
- [Zalzala et al., 1991] A. M. S. Zalzala et A. S. Morris (1991). A distributed on-line trajectory generator for intelligent sensory-based manipulator. *Robotica*, 9 :145–155.
- [Zinn et al., 2002] M. Zinn, O. Khatib, B. Roth, et J. K. Salisbury (2002). Toward a human-centered intrinsically-safe robotic manipulateur. Dans *Second IARP-IEEE/RAS Joint Workshop on technical challenges for dependable robots in human environments*, LAAS-CNRS, Toulouse, France.

Abstract

Arm manipulator control and visual servoing for service robotic applications

In the context of service robotic, robots have to interact with environment and human. Human presence create a particular context where security and comfort take precedent over system dynamic. As environment is not fixed in this context, the use of exteroceptive sensors is necessary. A first subject developed in the report is the trajectory planning for which we propose to limit velocity, acceleration and jerk. Trajectories are then defined as a sequence of cubic, seven in most to go from a initial position to any final position. The method is extended to space and to the tracking of trajectory defined by points. The rotations are represented by quaternions. We then propose a visual servoing scheme based on the tracking of planned soft movements. Using an extended Kalman filter to estimate the target position permit a self calibration of the system. Simulations and experimentations with the Jido robot permit to validate this approach. The trajectory planner is used daily on Jido for the european Cogniron project.

Keywords: service robotic, manipulation arm, trajectory generation, visual servoing.

Résumé

Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service

Dans le contexte de la robotique de service, les robots doivent interagir avec l'environnement et les humains qui s'y trouvent. La présence d'humain crée un contexte particulier où la sécurité et le confort des utilisateurs prime sur la dynamique du système. L'environnement n'étant pas figé dans ce contexte, l'utilisation de capteurs extéroceptifs est indispensable. Un premier thème développé dans le mémoire est la planification de trajectoires pour laquelle nous proposons de borner la vitesse, l'accélération et le jerk. Les trajectoires sont alors définies par une succession de cubiques, sept pour passer d'une situation initiale à une situation finale quelconque. La méthode est étendu à l'espace et au suivi d'une trajectoire définie par des points, les rotations y sont représentées par des quaternions. Nous proposons ensuite une commande par retour visuel fondée sur le suivi de mouvements souples planifiés. L'utilisation d'un filtre de Kalman étendu pour estimer la position de la cible autorise une auto-calibration du système. Des simulations et des expérimentations sur le robot Jido permettent de valider cette approche. Le planificateur de trajectoire est utilisé quotidiennement sur le robot Jido dans le cadre du projet européen Cogniron.

Mots-clés: robotique de service, bras manipulateurs, génération des trajectoires, retour visuel.

