



HAL
open science

Asservissement visuel coordonné de deux bras manipulateurs

Renliw Fleurmond

► **To cite this version:**

Renliw Fleurmond. Asservissement visuel coordonné de deux bras manipulateurs. Robotique [cs.RO]. Université Toulouse III Paul Sabatier, 2015. Français. NNT: . tel-01280180

HAL Id: tel-01280180

<https://theses.hal.science/tel-01280180v1>

Submitted on 29 Feb 2016

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.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue par :

Renliw Fleurmond

le jeudi 17 décembre 2015

Titre :

Asservissement visuel coordonné de deux bras manipulateurs

École doctorale et discipline ou spécialité :

EDSYS : Automatique, Signal, Productique, Robotique 4200046

Unité de recherche :

Laboratoire d'Analyse et d'Architecture des Systèmes

Directeur/trice(s) de Thèse :

Viviane Cadenat

Daniel Sidobre

Jury :

Viviane Cadenat directrice de thèse

Daniel Sidobre codirecteur de thèse

Youcef Mezouar rapporteur

Philippe Fraise rapporteur

Patrick Danès examinateur

Nestor Arana Arexolaleiba examinateur

Asservissement visuel coordonné de deux bras manipulateurs

Renliw Fleurmond

10 février 2016

Remerciements

J'aime à imaginer le travail de thèse comme une expédition dans des terres lointaines, pour dresser une carte d'une région inconnue ou plutôt méconnue. Je serais le géographe amateur rêvant de découvrir de nouvelles sources, des paysages fabuleux, des animaux étranges, me lançant témérairement dans l'aventure sans grand idée du parcours à emprunter, des difficultés à surmonter, et de ce qu'il faut emporter.

Heureusement que deux personnes aguerries ont préparé avec soin cette expédition et ont accepté de superviser l'étudiant que j'étais. Elles m'ont indiqué les jalons, les points de repères que je devais localiser sur ma carte, m'ont prévenu des dangers que je pourrais rencontrer, et ont été tout le temps disponibles, comme un téléphone satellite toujours chargé, toujours en ligne dans la jungle étouffante ou dans le désert brûlant. Je veux ici remercier mes directeurs de thèse, Mme Viviane Cadenat et M. Daniel Sidobre, pour m'avoir choisi pour cette bourse de thèse, pour leur temps, leurs critiques constructives, leur souci du détail et leur patience. Leur concours m'a été précieux, il ont bien voulu lire toutes les versions de mon manuscrit de thèse et de mes autres publications.

Je suis reconnaissant à M. Youcef Mezouar et M. Philippe Fraisse d'avoir accepté de rapporter sur mes travaux, et leur lecture de mon manuscrit. Je remercie également M. Patrick Danès pour m'avoir fait l'honneur de présider mon jury de thèse, et M. Nestor Arana Arexolaleiba pour avoir gracieusement accepté d'en faire partie. Je suis leur obligé.

Ma reconnaissance va au directeur du laboratoire Jean Arlat qui m'a accueilli au sein de son institution. Je remercie une nouvelle fois, Patrick Danès pour m'avoir accepté dans le groupe Robotique, Action, Perception, et pour l'atmosphère agréable qu'il contribue à y créer.

Cette atmosphère, et la camaraderie qui règne dans le groupe, m'ont beaucoup aidé. J'ai aimé participer aux soirées que nous organisons, les discussions improvisées autour d'un café, du déjeuner ou même au bureau. Je sais gré particulièrement à Marcus Futterlieb, Gabriel Bustamante et Ariel Poldubne d'avoir supporté mes bavardages, mes siestes, et mes fredonnements. Je remercie toutes les autres personnes que j'ai pu côtoyer pendant ces trois ans : Alban Portello, Alexandre Izaute, Alhayat Ali Mekonnen, Ali Alhamwi, Aurélie Clodic, Bruno Celaries, Daniel Tortei, Elie Moussy, Francisco Rodolfo Barbosa Anda, François Brenot, Guido Manfredi, Guilhem Marion, Jessica Combiér, Jorge Othon Esparza Jimenez, Laurent Fitte-Duval, Lucas Marti, Stanislas Larnier, Samuel Petrocelli, Thomas Forgeue. Cette liste n'est pas exhaustive. Il se peut que j'ai fait des oublis, si c'est le cas, veuillez me pardonner, et y ajouter votre nom. J'ai également une pensée pour les membres du laboratoire qui par leur travail, ont permis que je fasse le mien. Je vous remercie pour tous les sourires échangés qui ont souvent apporté des notes de gaieté et des pinceaux de lumière dans mes journées.

Je ne pourrais pas oublier les amis que j'ai fait à Toulouse ou en Haïti, qui m'ont encouragé, soutenu, égayé, aidé pendant ces études. Leur chaleur, leur affection et leur amitié m'ont donné la force d'arriver jusqu'ici. Certains ont pu assister à ma soutenance, mais tous étaient de coeur avec moi, et leur soutien m'a fortifié, affermi et réjoui. Je veux vous signifier ici mon infinie gratitude. Je remercie particulièrement Marie Carmel Lucien, Rosemond Saint-Paulin et Stevenson Pierre pour avoir gracieusement accepté de relire mon manuscrit de thèse.

Enfin, sans ma famille, ma soeur Cassandra qui était avec moi à Toulouse, mes autres parents en Haïti, je ne sais pas ce que je serais devenu et je leur suis infiniment reconnaissant. Sans eux, rien de cela n'aurait été possible. J'espère faire toujours leur joie, parfois les rendre fier, et chaque jour essayer de leur renvoyer une part de l'amour dont ils m'enveloppent. Mille mercis !

Notations

Règles générales

- Les lettres latines minuscules a, b, \dots, z sont des scalaires et seront assez souvent des inconnues.
- Des exceptions à la règle précédente seront les lettres e, q et s qui seront des vecteurs de dimension quelconques.
- Les lettres latines x, y, z ou leurs majuscules X, Y, Z sont des scalaires et seront les coordonnées d'un vecteur ou d'un point dans un repère ou dans un plan.
- Les lettres majuscules H, I, J, L, M, R et W sont des matrices.
- Dans certains chapitres la lettre majuscule P représentera une matrice, alors que dans les autres, ce sera un vecteur. L'on fera cette précision dans chaque chapitre à la première utilisation.
- Les autres lettres latines majuscules A, B, C, \dots, U, V seront des vecteurs, sauf contre indication, de dimension 3×1 .
- Les lettres minuscules grecques suivantes représentent, sauf contre indication, des scalaires réels, et sont des valeurs connues ou fonctions de valeurs connues ou mesurées :

$$\alpha \quad \epsilon \quad \theta \quad \lambda \quad \mu \quad \rho \quad \sigma \quad \tau$$

- La lettre grecque ω représente le vecteur vitesse angulaire instantanée et ses versions indicées $\omega_x, \omega_y, \omega_z$ sont ses coordonnées dans un repère.
- Les lettres minuscules grecques suivantes, sont des vecteurs de dimensions 3×1 et représenteront des mesures ou combinaison de mesures :

$$\beta \quad \gamma \quad \delta \quad \zeta \quad \xi$$

- Enfin les lettres grecques suivantes, en respectant la casse, sont des matrices et représentent des mesures ou des fonctions de mesures :

$$\Delta \quad \Theta \quad \Pi \quad \phi \quad \Phi \quad \chi \quad \psi \quad \Psi \quad \Omega$$

- Signification de certains accents et indices

- \dot{a} : dérivée de a par rapport au temps.
- \ddot{a} : dérivée seconde de a par rapport au temps.
- \hat{a} : valeur estimée ou estimateur de a
- a^* : grandeur de référence pour a

Algèbre

\mathbb{R}	: ensemble des nombres réels
\mathbb{R}^n	: ensemble des n -uplets d'éléments de \mathbb{R}
$SO(3)$: groupe orthogonal des rotations dans l'espace euclidien
$SE(3)$: espace euclidien des matrices homogènes $SE(3) = SO(3) \times \mathbb{R}^3$
M^T	: transposée de la matrice M
M^{-1}	: matrice inverse de M
M^+	: pseudo-inverse de Moore-Penrose de M
$\det(M)$: déterminant de M
$Ker(M)$: noyau de M
$[V]_{\wedge}$: matrice de pré-produit vectoriel associé au vecteur V
$\ V\ $: 2-norme du vecteur V
I_n	: matrice identité de dimensions $n \times n$
$0_{n \times m}$: matrice nulle de dimensions $n \times m$

Géométrie

Dans tout ce document la convention du trièdre direct sera utilisée.

F_a	: repère cartésien orthonormé associé à a
F_w	: repère monde
aV	: expression du vecteur V dans le repère F_a
$U \wedge V$: produit vectoriel de U par V
iR_j	: matrice de rotation qui exprime la base j en fonction de la base i
T_j	: torseur cinématique de F_j par rapport au repère monde F_w en l'origine du repère F_j et exprimé dans ce même repère F_j
iT_j	: torseur cinématique de F_j par rapport à F_w , en l'origine du repère F_i et exprimé dans la base du repère F_i .
iW_j	: matrice de changement de repère pour les torseurs cinématiques ${}^iT_j = {}^iW_j \times T_j$

Table des matières

1	Introduction générale	13
2	La coordination entre bras manipulateurs	17
2.1	Problématique et définitions	17
2.2	Cadre théorique de la coordination	19
2.2.1	Commande en position ou en vitesse	19
2.2.2	Commande hybride position/force	19
2.2.3	Commande par impédance	22
2.2.4	Commande basée sur la vision	22
2.3	Conclusion	23
3	La coordination par asservissement visuel	25
3.1	Introduction	25
3.2	Asservissement visuel 2D	26
3.3	La coordination par l'asservissement visuel	28
3.3.1	Caméra déportée	30
3.3.1.1	Comanipulation	30
3.3.1.2	Coordination asymétrique	31
3.3.2	Caméra embarquée	33
3.3.3	Caméra embarquée indépendante	34
3.3.3.1	Comanipulation	35
3.3.3.2	Coordination asymétrique	36
3.3.4	Liaisons virtuelles entre objets	36
3.3.4.1	Liaison rigide	37
3.3.4.2	Liaison prismatique	37
3.3.4.3	Liaison pivot glissant	37
3.3.5	Conclusion	38
3.4	Application : refermer un stylo	38
3.4.1	Présentation du système robotique considéré et des caméras utilisées	39
3.4.1.1	Système robotique	39
3.4.1.2	Modèle de la caméra	40
3.4.2	Choix des indices visuels	41
3.4.3	Modélisation de la tâche et stratégie de commande	42
3.4.3.1	Décomposition en trois sous-tâches	42
3.4.3.2	Synthèse de la commande	43
3.4.3.3	Enchaînement des correcteurs	44
3.4.4	Résultats de simulation	45
3.5	Conclusion	50

4	La gestion de la redondance	53
4.1	Introduction	53
4.2	Redondance dans le formalisme des fonctions de tâche	54
4.2.1	Etat de l'art	54
4.2.2	Discussion générale	56
4.2.3	Analyse des approches retenues et proposition d'une nouvelle stratégie	57
4.2.4	Évaluation de la stratégie proposée au travers d'un exemple	62
4.2.5	Conclusion	64
4.3	Prise en compte des butées et des contraintes de visibilité	64
4.3.1	Évitement des butées	65
4.3.2	Contraintes sur la visibilité	66
4.3.3	Résultats de simulation	68
4.4	Conclusion	71
5	Estimation et reconstruction d'indices visuels	73
5.1	Introduction	73
5.2	État de l'art	74
5.3	Discussion	77
5.4	Estimation du point	78
5.4.1	Méthode du point milieu	78
5.4.2	Résultats de simulation	80
5.5	Extension de la méthode du point milieu à d'autres primitives.	85
5.5.1	Cas des droites	85
5.5.1.1	Méthode développée	85
5.5.1.2	Résultats de simulation	88
5.5.2	Estimation du cylindre	94
5.5.2.1	Cas du cylindre de longueur infinie	94
5.5.2.2	Cas du cylindre de longueur finie	98
5.5.2.3	Résultats de simulation	102
5.6	Conclusion	105
6	Résultats	107
6.1	Introduction	107
6.2	Premier scénario : gestion d'une panne temporaire des caméras	107
6.3	Second scénario : extension de la coordination à la tête et aux bras du robot	112
6.4	Conclusion	121
7	Conclusion et perspectives	123
A	Détermination des différentes matrices d'interaction	127
A.1	Matrice d'interaction du point	127
A.2	Matrice d'interaction de la droite	128
A.3	Matrice d'interaction du segment	129
A.4	Indices visuels et matrice d'interaction pour le cylindre	129
A.5	Extension au cas du cylindre de longueur finie	131

Table des figures

1.1	Le robot PR2 (Courtoisie de Willow Garage).	14
2.1	Manipulation d'un corps rigide par deux bras	18
2.2	Coordination pour une tâche d'assemblage	18
2.3	Coopération entre les organes terminaux	20
2.4	Coordination et contrôle par impédance.	22
3.1	Deux configurations : caméra embarquée et caméra déportée	25
3.2	Coopération entre caméra embarquée et caméra déportée.	28
3.3	Repère liés aux organes terminaux et à l'objet	29
3.4	Caméra déportée par rapport aux deux bras	30
3.5	Assemblage de deux parallélépipèdes.	32
3.6	Caméra solidaire de l'organe terminal d'un des bras.	33
3.7	Caméra active, quand le champ de vue de la caméra varie	35
3.8	Liaison virtuelle rigide entre deux parallélépipèdes	37
3.9	Liaison virtuelle entre deux cylindres	38
3.10	Géométrie des objets et de la tâche à réaliser	39
3.11	Placement des repères sur le PR2.	40
3.12	Modèle sténopé de la caméra.	40
3.13	Indices visuels retenus : ρ , θ et k .	41
3.14	Évolution de la tâche de fermeture du stylo	46
3.15	Commande du bras droit du robot PR2	47
3.16	Commande du bras gauche du robot PR2	47
3.17	Vues du robot PR2 effectuant la tâche de fermeture du stylo	48
3.18	Évolution de la tâche bruitée en fonction du temps.	49
3.19	Commande en vitesse du bras gauche du PR2 en présence de bruit.	49
3.20	Une articulation arrive en butée.	50
4.1	Espace des vitesses articulaires	57
4.2	Les lois de commande de e_1 et e_2 paraissent compatibles.	58
4.3	Les lois de commande de e_1 et e_2 vont dans des directions opposées.	58
4.4	Singularité algorithmique de la redondance optimale.	59
4.5	Transitions entre la redondance classique et la redondance optimale.	60
4.6	Risque de discontinuité dans notre approche	61
4.7	Trajectoires du robot plan	63
4.8	Évolution des coordonnées du robot avec la redondance optimale	63
4.9	Vitesses du robot plan	64
4.10	Structure des tâches avec la redondance.	65
4.11	Fonction répulsive en dehors de l'intervalle de sécurité.	65
4.12	Limites des indices visuels choisis.	67
4.13	Évolution de la fonction de tâche d'évitement des butées F_b .	69
4.14	Une articulation s'arrête sur sa butée.	69
4.15	L'articulation est maintenue à distance de sa butée.	69

4.16	Évolution de la tâche d’asservissement visuel.	70
4.17	Évolution de la fonction de tâche d’évitement des occultations F_v	70
4.18	Commande du bras droit du robot PR2	71
4.19	Commande du bras gauche du robot PR2	71
5.1	Méthodes de triangulation.	75
5.2	Généralisation de la méthode du point milieu à n vues.	79
5.3	Conditions de simulations	81
5.4	Définitions des erreurs pour la triangulation du point	81
5.5	Erreur médiane de reconstruction pour deux vues.	82
5.6	Erreur médiane de reprojection pour deux vues.	82
5.7	Erreur médiane de reconstruction en fonction du nombre de vues.	83
5.8	Erreur médiane de reprojection en fonction du nombre de vues.	84
5.9	Modèle de l’erreur de reprojection pour deux vues.	84
5.10	Modèle de l’erreur de reprojection pour n vues.	85
5.11	Erreur géométrique dans le cas de la droite dans l’espace.	86
5.12	Conditions de simulations pour la droite	89
5.13	Estimation du paramètre P de la droite en fonction du temps.	89
5.14	Estimation du paramètre U de la droite en fonction du temps.	90
5.15	Coordonnées du centre optique de la caméra en fonction du temps.	90
5.16	Évolution des composantes de la tâche référencée vision sur trois droites.	91
5.17	Reconstruction des indices visuels pendant une tâche d’asservissement	91
5.18	Reconstruction des indices visuels de la droite	92
5.19	Position de la caméra au cours du temps.	92
5.20	Orientation de la caméra au cours du temps.	93
5.21	Estimation du paramètre P de la droite en fonction du temps.	93
5.22	Estimation du paramètre U de la droite en fonction du temps.	94
5.23	Plans tangents au cylindre passant par le centre optique.	95
5.24	Paramètres et variables pour un cylindre court.	99
5.25	Indices visuels pour un cylindre de longueur finie	102
5.26	Conditions de simulations pour le cylindre	102
5.27	Évolution des composantes de la tâche en fonction du temps.	103
5.28	Estimation du rayon du cylindre en fonction du temps.	103
5.29	Estimation de la base du cylindre court	104
5.30	Estimation de la face supérieure du cylindre court	104
6.1	Évolution de la tâche coordonnée référencée vision.	109
6.2	Évolution de la fonction de tâche d’évitement des butées F_b	109
6.3	Commande articulaire du bras droit du PR2.	110
6.4	Commande articulaire du bras gauche du PR2.	110
6.5	Estimation du paramètre P du stylo.	111
6.6	Estimation du paramètre U du stylo.	111
6.7	Estimation du paramètre K du stylo.	112
6.8	Indices visuels pour un cylindre de longueur finie	113
6.9	Évolution de la tâche coordonnée référencée vision.	114
6.10	Commande en vitesse sur le bras droit du PR2.	114
6.11	Commande en vitesse sur le bras gauche du PR2.	115
6.12	Évolution de la tâche d’évitement des occultations	115
6.13	Commande en vitesse des articulations de la tête.	116
6.14	Évolution des coordonnées articulaires de la tête.	116
6.15	Estimation du rayon du stylo.	117
6.16	Estimation de la base du stylo.	117

6.17	Estimation du dessus du stylo.	118
6.18	Estimation du rayon du capuchon.	118
6.19	Estimation de la base du capuchon.	119
6.20	Estimation du dessus du capuchon.	119
6.21	Coordination des mouvements de la tête et des bras du PR2.	120
A.1	Vue de haut du cylindre et de la caméra	130

Liste des tableaux

3.1	Valeurs initiales des coordonnées articulaires des deux bras du PR2.	45
3.2	Paramètres intrinsèques des caméras utilisés en simulation.	45
3.3	Significations des composantes des sous-tâches d'asservissement visuel. . .	46
4.1	Limites des articulations des deux bras du PR2.	68
6.1	Paramètres intrinsèques des caméras utilisés en simulation.	108
6.2	Valeurs initiales des coordonnées articulaires des deux bras du PR2.	108
6.3	Significations des composantes des sous-tâches référencées vision.	113
6.4	Limites des articulations de la tête du PR2.	113
6.5	Valeur initiale des articulations de la tête du PR2.	114

Chapitre 1

Introduction générale

Tim, tim ? Bwa chèch ! - pour annoncer une devinette, ou un conte.

Expression haïtienne

Les robots font maintenant partie intégrante de notre monde. Ils ont également conquis notre imaginaire, hantant nos univers de science-fiction, jusqu'à se matérialiser sur nos écrans, dotés de capacités extraordinaires. Mais comme d'habitude, la réalité est plus ordinaire. La plupart d'entre eux restent confinés dans des usines, réalisant des tâches dangereuses pour l'homme ou répétitives. Ils soudent, peignent, manipulent, et participent à la construction des objets de tous les jours, des jouets ou des automobiles. Pourtant certains nous côtoient chaque jour, quelques uns même tentent d'évincer de notre quotidien le balai et l'aspirateur, nous dispensant de quelques travaux domestiques.

L'ATILF (Analyse et traitement automatique de la langue française) définit un robot comme un "appareil effectuant, grâce à un système de commande automatique à base de micro-processeur, une tâche précise pour laquelle il a été conçu dans le domaine industriel, scientifique ou domestique." Cette définition décrit le principe de fonctionnement de tous les robots existants, mais ne donne aucune information sur les tâches effectuées ou sur leur aspect visuel. Les robots se présentent sous des formes diverses et variées, les morphologies choisies, étant *a priori* adaptées aux tâches qu'ils devront effectuer. Les robots industriels se résument le plus souvent à des bras mécaniques commandés par ordinateur, munis d'outils fixés sur l'organe terminal. Ils travaillent généralement dans un environnement très structuré et peu changeant. Les premiers robots étaient aveugles à leur environnement extérieur, et ne contrôlaient leurs mouvements qu'en utilisant la *proprioception*. Mais les robots industriels ont dû évoluer et s'adapter pour l'usine du futur. Ils doivent collaborer avec des opérateurs humains, mettant leur force et leur répétabilité au service de l'expérience et de l'intuition humaine. D'autres s'aventurent dans nos maisons ou dans la nature, mais tous doivent s'adapter à un environnement dynamique.

Ils ont donc été dotés de capteurs *extéroceptifs*, leur permettant de recueillir des informations sur le monde environnant. Ces capteurs sont de natures diverses et sont employés à des fins très variées. Les capteurs de force peuvent renseigner sur les forces exercées par le robot ou par des agents extérieurs. Les capteurs à ultrasons et les lasers offrent la possibilité de détecter la proximité immédiate d'un obstacle ou de scruter une partie de la structure de l'environnement du robot. Les caméras et les capteurs de profondeur type "Kinect" offrent une vue de cet environnement, et peuvent fournir de riches informations, comme la couleur, la texture, la présence, la localisation et l'identification de personnes ou d'objets. Les micros sont eux aussi utilisés pour permettre au robot d'exécuter des

ordres oraux, ouvrant ainsi la voie à l'audition en robotique. Ce domaine est d'ailleurs actuellement en expansion, avec le développement de travaux notamment orientés vers la localisation de sources sonores. Pour exploiter efficacement les mesures extéroceptives, il est nécessaire de pouvoir les traiter automatiquement. Pour cela, il faut un ou des processeurs suffisamment puissants et des algorithmes permettant d'extraire les informations utiles pour les tâches envisagées. La capacité brute de calcul des processeurs croît quasiment de manière exponentielle vérifiant les conjectures de Moore, et la recherche est très active dans beaucoup de domaines scientifiques tels le traitement du signal, la vision par ordinateur ou l'intelligence artificielle.

La morphologie des robots s'adapte aussi pour leur permettre d'évoluer dans différents environnements. Certains comme TurtleBot [Turtlebot, 2015], Curiosity [Curiosity, 2015], Roomba [Roomba, 2015] ou même les voitures autonomes de Google, sont mobiles et munis de roues, ils parcourent nos parquets, nos routes ou même les lointaines contrées de la Lune, de Mars (le robot Philae [Philae, 2015] a même récemment roulé sur une comète). Les voiliers autonomes tels le Vaimos [Vaimos, 2015] sillonnent nos mers, les robots sous-marins [OpenROV, 2015] auscultent les profondeurs des océans et les drones aériens civils ou militaires quadrillent le ciel. D'autres tentent de reproduire des capacités intéressantes que la nature a accordées aux animaux. Il en est de quadrupèdes qui s'adaptent aux terrains accidentés ou glissants comme Big Dog ou Cheetah de Boston Dynamics [Dynamics, 2015]. Il existe des robots inspirés des araignées, des insectes pour la marche et le vol, des poissons et des serpents pour la reptation et la nage, des oiseaux pour le vol [Bird, 2015]. Et ils se diversifient d'année en année, au fil des recherches, des découvertes et des inventions. D'autres robots ont été conçus pour être morphologiquement proches de l'homme, ils possèdent une tête, un tronc, deux bras. Beaucoup sont mobiles, quelques uns sont bipèdes alors que d'autres se meuvent à l'aide de roues. Les robots Asimo [Asimo, 2015] de Honda, Robonaut [Robonaut, 2015] de la NASA, les robots HRP [HRP, 2015] de Kawada Industries, Nao [Nao, 2015] et Roméo [Romeo, 2015] de Aldebaran Robotics, PR2 [PR2, 2015] de Willow Garage, Justin [Justin, 2015] et les robots ARMAR [Armar, 2015] en sont des exemples. Ces structures anthropomorphiques devraient permettre au robot d'effectuer des tâches réalisées par les humains et encore inaccessibles aux machines. Elles devraient aussi faciliter son insertion dans des environnements conçus par et pour les humains. Les robots anthropomorphes seraient ainsi appelés à remplacer les humains pour réaliser des tâches dangereuses, comme par exemple lors d'une catastrophe naturelle ou industrielle¹. Ils pourraient aussi les aider dans leur vie quotidienne.

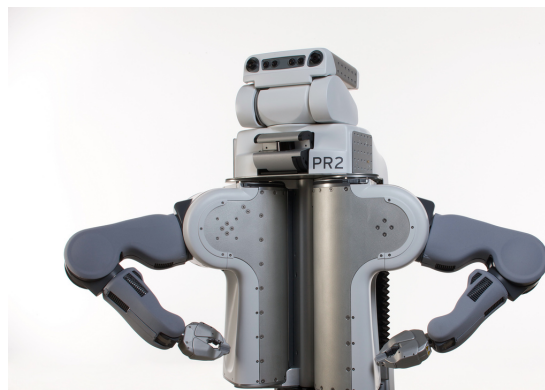


FIGURE 1.1 – Le robot PR2 (Courtoisie de Willow Garage).

1. Cela a été le thème d'une compétition internationale de robotique : DARPA Robotics Challenge [DRP2015, 2015]

Ces robots étant appelés à nous côtoyer, l'aspect interaction et coopération avec l'homme fait l'objet d'une recherche intense. Il faut en effet s'assurer qu'ils seront sans danger pour l'homme, et qu'ils ne menaceront pas son confort par leur présence. Il nous faut des machines, qui puissent se faire oublier quand c'est nécessaire, et être néanmoins présentes pour nous aider. Pour répondre à nos attentes, plusieurs robots doivent être capables de manipuler des objets usuels, au moins aussi bien que les humains, et d'utiliser leurs deux bras pour l'accomplissement de leur tâche, comme nous le faisons inconsciemment et sans effort tous les jours. En effet, beaucoup d'actions quotidiennes requièrent chez l'homme l'emploi des deux mains. Visser/dévisser le bouchon d'une bouteille, en verser le contenu dans un verre, porter un plateau ou d'autres objets lourds, plier/déplier une pièce de tissu, manœuvrer un balai, tourner une vanne ou un volant, sont autant d'exemples d'actions que nous accomplissons tous les jours. Elle semblent intuitives et aisées aux jeunes en bonne santé, mais les difficultés sont bien réelles pour ceux qui souffrent de déficiences motrices ou sensitives et sont un défi certain pour un robot qui prétend apporter une réelle aide domestique.

Ainsi, un aspect important de la manipulation à deux bras, et présent dans les exemples de manipulation cités plus haut est la *coordination* entre les deux bras. Comment peut-on caractériser cette coordination, et quels sont les moyens dont on dispose pour la mettre en œuvre sur un robot à deux bras ? C'est le problème auquel s'intéressera cette thèse, en étudiant une tâche simple : la fermeture d'un stylo. Il s'agit un exemple domestique d'une tâche d'assemblage, et tous les résultats tirés de cette étude sont susceptibles de pouvoir être étendus à de nombreuses tâches d'assemblage dans un contexte plus industriel. De nombreux aspects entrent en compte dans la réalisation d'une manipulation coordonnée. La perception des objets à manipuler, leur préhension par les organes terminaux du robot, la planification automatique du mouvement à effectuer, la commande pour la réalisation de ces mouvements, la gestion des collisions entre les bras eux-mêmes et les obstacles extérieurs éventuels, sont des problèmes ouverts et possèdent pour beaucoup une littérature scientifique spécialisée. Dans le cadre de cette thèse, le choix a été fait de se focaliser sur l'aspect commande du robot, en négligeant l'étape de la préhension des objets. Pour être le moins tributaire possible de la manière dont les objets ont été saisis, l'utilisation de caméras a été privilégiée. Le but premier de cette thèse a donc été d'élaborer une commande pour la coordination basée sur la vision. En d'autres termes, l'objectif est de calculer à chaque instant la direction et l'amplitude du mouvement à exécuter par chaque articulation des deux bras du robot en utilisant les informations fournies par des caméras sur les positions des objets à manipuler pour mettre en œuvre cette coordination recherchée.

Le *deuxième* chapitre de cette thèse présentera donc succinctement la manipulation à deux bras et la manière dont est abordé le problème de la coordination. Il tentera également de préciser le cadre théorique de la coordination telle qu'elle est comprise dans ce travail. Il décrira plusieurs commandes, et témoignera du petit nombre de travaux sur la coordination par la vision.

Le *troisième* chapitre s'intéressera à la commande basée sur la vision, décrira la stratégie de commande développée pour coordonner deux bras. La méthode proposée sera détaillée pour plusieurs cas, mettant en lumière notre contribution sur l'asservissement visuel avec plusieurs caméras, cibles et bras. Cette méthode est aussi conçue pour contraindre au minimum les bras du robot, lui permettant de réaliser plus facilement d'autres tâches simultanément.

Le *quatrième* chapitre s'intéressera ainsi à la réalisation simultanée de plusieurs tâches

par un robot, ou dit autrement la gestion de la redondance lorsque le robot dispose de plus de degrés de liberté que nécessaire pour effectuer une tâche particulière. Pour la réalisation de la tâche d'assemblage considérée, plusieurs conditions doivent être satisfaites dont le respect des butées du robot et le maintien dans le champ de vue des caméras des objets manipulés. Ces conditions seront décrites comme des tâches à réaliser conjointement avec la première évoquée. Malgré ces précautions, une panne de la caméra ou du traitement d'images associé peut survenir.

Le *cinquième* chapitre développera donc une méthode pour localiser différents objets géométriques, en vue de reconstruire les données visuelles les décrivant sur les images fournies par les caméras. Grâce à cette contribution, notre système sera robuste aux occultations (quelles qu'en soient leurs origines), et les informations estimées pourront même permettre d'améliorer la réalisation de la commande décrite au chapitre trois.

Le *sixième* chapitre enfin sera dédié aux résultats obtenus pour valider nos contributions. Nous terminerons ce manuscrit avec un chapitre dédié aux conclusions et perspectives ouvertes par nos travaux.

Chapitre 2

La coordination entre bras manipulateurs

"Men anpil, chay pa lou"

littéralement, plus il y a de bras,
moins lourde est la charge.

Proverbe haïtien

2.1 Problématique et définitions

La problématique de la manipulation à deux bras a été abordée dès les années 70 [Caccavale and Uchiyama, 2008]. Plusieurs facteurs peuvent expliquer cet engouement de la communauté robotique pour l'utilisation de plateformes à deux bras. Utiliser deux bras dans une chaîne cinématique fermée permet de bénéficier d'une force accrue, d'une certaine rigidité, tout en gardant la flexibilité d'un bras série [Bonitz and Hsia, 1996]. La manipulation conjointe d'objets flexibles requiert souvent l'emploi de deux bras [Zheng and Chen, 1993, Al-Jarrah et al., 1995, Al-Yahmadi and Hsia, 2007] comme par exemple plier un mouchoir ou une serviette [Maytin-Shepard et al., 2010]. De nombreuses tâches d'assemblages [Yamada et al., 1995, Caccavale et al., 2001, Park et al., 2006, Takamatsu et al., 2007, Liza Ahmad Shauri and Nonami, 2011] nécessitent le contrôle de la *pose*, c'est-à-dire de l'orientation et de la position, de chacun des objets à assembler, comme par exemple usiner une pièce, serrer une vis. Toutes ces actions sont rendues plus faciles avec l'utilisation de plusieurs bras. En outre, grâce à cette redondance pour la manipulation, les plateformes à plusieurs bras peuvent mieux coopérer avec des humains. Elles sont aussi davantage susceptibles de reproduire leurs mouvements et leur savoir-faire.

La manipulation d'un objet est l'application de forces sur cet objet, dans le but de le déplacer et/ou de le déformer. La manipulation peut être rigide, dextre ou souple, dépendant de la rigidité de l'objet et des outils utilisés (pince ou doigts par exemple). Dans ce document, nous entendons par manipulation à deux bras, toutes les tâches où tous les deux bras doivent manipuler des objets, incluant le cas où ils manipulent le même objet.

Les travaux sur la manipulation à deux bras peuvent être répartis en deux catégories, selon la présence ou l'absence d'une coordination entre les deux bras [Zollner et al., 2004, Smith et al., 2012]. Encore faut-il que la notion de coordination soit précisément définie. Certains auteurs tels que Smith et al. [Smith et al., 2012] utilisent une définition assez large. Ainsi la coordination est-elle vue comme un ensemble d'actions effectuées par

les deux bras pour exécuter une tâche. Si un robot utilise ses deux bras pour prendre des objets sur un plan de travail, et les ranger dans une boîte, si l'on considère la tâche de haut niveau qui consiste à ranger les objets dans la boîte, cet exemple satisfait la définition [Smith et al., 2012], même dans le cas où les deux bras sont activés alternativement et jamais simultanément.

Un cas de manipulation considéré à l'unanimité comme coordonné [Caccavale and Uchiyama, 2008, Zollner et al., 2004, Adorno et al., 2010, Smith et al., 2012], est l'utilisation de plusieurs bras pour manipuler un seul objet (Figure 2.1). Les poses, les vitesses des organes terminaux, et les forces et moments qu'ils exercent sont liés à cause de la chaîne cinématique fermée.

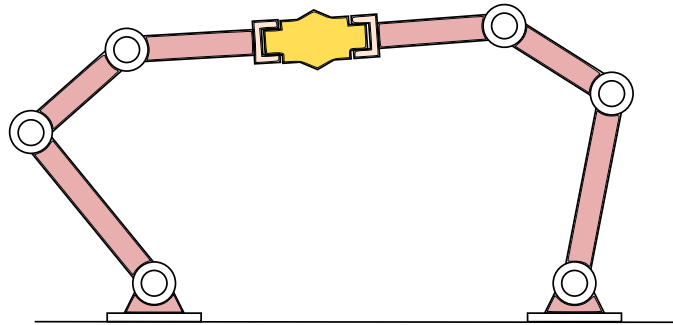


FIGURE 2.1 – Manipulation d'un corps rigide par deux bras

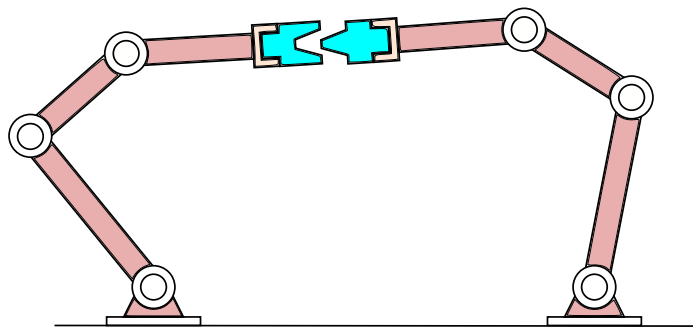


FIGURE 2.2 – Coordination pour une tâche d'assemblage

De même les tâches d'assemblage (Figure 2.2), comme refermer un stylo ou verser du liquide dans un verre apparaissent également comme des tâches coordonnées [Dauchez et al., 2005, Adorno et al., 2010]. En effet, d'une part, les deux bras *peuvent* être contrôlés ensemble, et d'autre part le mouvement d'un bras influence celui de l'autre bras. Dans la suite de ce document, la *coordination de plusieurs bras* sera comprise comme la *mise en place ou la prise en compte de contraintes entre les organes terminaux des bras*, contraintes portant sur leurs poses, leurs vitesses et les forces qu'ils exercent. Ainsi, le mouvement alterné de deux bras n'est pas inclus dans cette définition, parce que les bras ne bougent pas en même temps. Est exclu également le cas où les bras effectuent en même temps des actions indépendantes.

Se distinguent donc deux types de tâches coordonnées de manipulation, que Zollner et al. [Zollner et al., 2004] ont qualifiés de *symétrique* et d'*antisymétrique*. Le premier type est représenté par la manipulation d'un seul objet par plusieurs bras que nous appellerons dans la suite du document la *comanipulation*. Le deuxième type regroupe les tâches d'assemblage où les pièces sont positionnées chacune par un bras manipulateur.

2.2 Cadre théorique de la coordination

Après avoir défini ce que nous entendons par coordination, nous présentons dans cette section un bref survol de la littérature portant sur ce sujet. De nombreuses solutions issues de domaines scientifiques variés permettent en effet de répondre à ce problème. Certains travaux ont ainsi abordé la manipulation à deux bras et la comanipulation par le biais de la planification de trajectoire avec ou sans obstacles [Zheng and Chen, 1993, Al-Jarrah et al., 1995, Gharbi et al., 2008, Vahrenkamp et al., 2009a, Cohen et al., 2012]. D'autres auteurs ont proposé d'utiliser des techniques d'apprentissage relevant davantage du domaine de l'intelligence artificielle [Zollner et al., 2004, Takamatsu et al., 2007]. Pour notre part, nous avons choisi dans cette thèse de nous focaliser sur les techniques de commande. Nous décrirons ainsi différentes structures ayant été utilisées pour coordonner les mouvements de deux bras manipulateurs. Nous aborderons notamment la commande en position/vitesse, la commande hybride position/force, la commande par impédance, et enfin la commande référencée vision.

2.2.1 Commande en position ou en vitesse

Peu de travaux se sont intéressés à la commande en position ou en vitesse seule. L'une des raisons est qu'une petite erreur de position peut induire des forces de contact relativement grandes [Caccavale et al., 2001]. Adorno, après avoir recommandé l'utilisation du quaternion dual dans le formalisme des tâches coopératives [Adorno et al., 2010], propose un contrôleur en vitesse, utilisant ce même objet mathématique [Adorno, 2011]. Il réalise alors une tâche classique de comanipulation avec un objet déformable [Adorno et al., 2010], la déformation étant ici un effet non contrôlé et compensant les éventuelles erreurs de position. Il effectue également des tâches d'assemblage où il régule la position relative d'un objet par rapport à un autre, pour verser du liquide d'un récipient à un autre ou même coordonner le mouvement du robot à celui d'un humain [Adorno, 2011, Adorno et al., 2011]. Il est à noter qu'il décrit plusieurs tâches coordonnées domestiques et la manière dont on peut les décrire dans ce formalisme. En voici quelques exemples :

- Manipuler une vanne avec les deux mains, un plateau repas ou un balai : la pose relative entre les mains reste constante, et l'on précise le comportement voulu pour la pose absolue de l'objet.
- Manipuler un Rubik's cube ou ouvrir une bouteille : une partie de la pose relative doit rester constante, comme la position, tandis que l'autre est libre pour le mouvement.
- Transvaser un liquide entre deux récipients ou assembler deux objets : l'on régule la pose relative et l'on peut également imposer que l'orientation absolue d'un objet soit constante pour éviter de perdre du liquide par exemple.

2.2.2 Commande hybride position/force

Certains travaux ont appliqué, au cas de la comanipulation [Hayati, 1986, Hu et al., 1995, Kraus and McCarragher, 1997, Liu and Arimoto, 1998], le contrôle hybride en position et en force d'un bras manipulateur étudié par Craig et Robert [Craig and Raibert, 1979] en 1979. Notamment, Uchiyama et Dauchez [Uchiyama and Dauchez, 1988, Dauchez et al., 1992] ont proposé ce qu'ils ont appelé *le schéma symétrique*. Ils ont exprimé la résultante des forces et des moments extérieurs exercés sur un solide rigide manipulé, en fonction des forces et moments exercés par les organes terminaux des manipulateurs. Dans ce formalisme, comme souvent en mécanique [Pérez, 1989], la force et le moment

sont regroupés dans le vecteur généralisé force/moment :

$${}^a\Gamma = [t_x \ t_y \ t_z \ f_x \ f_y \ f_z]^T$$

avec t_x , t_y et t_z les composantes du moment résultant et f_x , f_y et f_z les composantes de la force résultante à l'origine du repère F_a et projetés dans ce repère.

Considérons la figure 2.3. Soient :

- les repères F_1 et F_2 associés aux organes terminaux des manipulateurs.
- F_e un repère lié au solide.
- Γ_1 et Γ_2 les forces et moments exercés sur le solide par les manipulateurs respectifs.

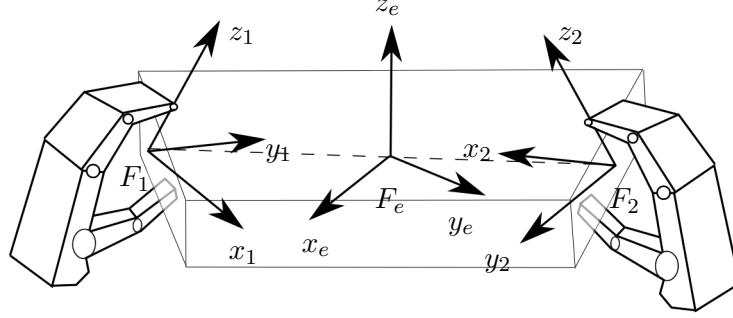


FIGURE 2.3 – Coopération entre les organes terminaux

La résultante à l'origine du repère F_e lié à l'objet, en ignorant la gravité, est la somme des vecteurs généralisés force/moment exercés sur le solide par les deux objets. La résultante des forces extérieures a pour expression [Uchiyama and Dauchez, 1988, Caccavale and Uchiyama, 2008] :

$$\begin{aligned} \Gamma_e &= {}^e\Gamma_1 + {}^e\Gamma_2 \\ \Gamma_e &= {}^eW_1 \Gamma_1 + {}^eW_2 \Gamma_2 \\ \Gamma_e &= \begin{bmatrix} {}^eW_1 & {}^eW_2 \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} \end{aligned} \quad (2.1)$$

avec eW_1 et eW_2 des matrices inversibles 6×6 de transformation de torseurs. Ces matrices dépendent des repères F_1 , F_2 et F_e . Rappelons en effet que la matrice aW_b , qui transforme le vecteur force/moment d'un repère F_b vers un repère F_a est inversible et définie par :

$${}^aW_b = \begin{bmatrix} {}^aR_b & [{}^aP_b]_{\wedge} \times {}^aR_b \\ 0_{3 \times 3} & {}^aR_b \end{bmatrix} \quad (2.2)$$

avec aR_b , la matrice de rotation 3×3 entre les repères F_a et F_b , le vecteur translation aP_b entre les repères F_a et F_b , et $[{}^aP_b]_{\wedge}$ sa matrice de pré-produit vectoriel.

La matrice ${}^eW_{12} = [{}^eW_1 \ {}^eW_2]$ est communément appelée la matrice de préhension. Cette matrice est de dimensions 6×12 et de rang 6, et possède donc un noyau de dimension 6. Les forces généralisées $[\Gamma_1^T \ \Gamma_2^T]^T$ appliquées par les manipulateurs qui se trouvent dans ce noyau, ne contribuent pas à la résultante des couples et des moments extérieurs sur le solide, mais causent des contraintes sur ce solide. La structure de cette matrice peut donc être exploitée pour contrôler la résultante des efforts extérieurs ainsi que les contraintes exercées [Watanabe et al., 2005, Hu et al., 1995, Kawasaki et al., 2005]. Certains ont pu minimiser ces contraintes [Hayati, 1986, Hu et al., 1995, Caccavale et al., 1999] tandis que d'autres les utilisent pour contrôler la déformation du corps [Kraus and McCarragher, 1997, Al-Yahmadi and Hsia, 2007], ou mieux répartir la charge sur les articulations [Liu and Arimoto, 1998].

La relation similaire entre le torseur cinématique résultant de l'objet manipulé et les torseurs cinématiques des différents organes terminaux a été également explicitée [Dauchez et al., 1992], dans le cadre d'une *saisie rigide* de l'objet, c'est-à-dire que les pinces peuvent être considérées comme solidaires de l'objet. Le torseur cinématique est représenté par un vecteur colonne de six paramètres réunissant la vitesse linéaire et la vitesse angulaire

$${}^aT = [v_x \ v_y \ v_z \ \omega_x \ \omega_y \ \omega_z]^T$$

avec v_x , v_y et v_z les composantes de la vitesse linéaire à l'origine d'un repère F_a , et ω_x , ω_y et ω_z les composantes de la vitesse angulaire, tous projetés dans le repère F_a . Soient T_1 et T_2 les torseurs cinématiques respectifs des organes terminaux 1 et 2, projetés respectivement dans les repères F_1 et F_2 . Avec une saisie rigide, et un corps non déformable, il vient [Dauchez et al., 1992] :

$${}^eT = {}^eW_1T_1 \qquad {}^eT = {}^eW_2T_2 \qquad T_1 = {}^1W_2T_2 \qquad (2.3)$$

Chiacchio et al. [Chiacchio et al., 1996] ont introduit *l'espace des tâches coopératives*. Dans cet espace, les tâches *coordonnées* symétriques et asymétriques ne sont pas définies par les poses absolues des organes terminaux. Elles sont définies en utilisant deux variables : la pose relative entre les deux organes terminaux et une *pose intermédiaire* entre celles des deux organes terminaux. Si l'on représente les poses des repères F_1 et F_2 par rapport au repère monde F_w , par la position respective de leur origine wP_1 et wP_2 et leurs matrices de rotation respectives wR_1 et wR_2 . La pose relative, représentée par les variables 1P_2 et 1R_2 , est donnée par :

$${}^1P_2 = {}^1R_w({}^wP_2 - {}^wP_1) \qquad (2.4)$$

$${}^1R_2 = {}^1R_w {}^wR_2 \qquad (2.5)$$

La pose intermédiaire absolue, saisie dans les variables wP_e et wR_e , a été choisie comme le *milieu* des poses des repères F_1 et F_2 [Chiacchio et al., 1996] :

$${}^wP_e = \frac{1}{2}({}^wP_1 + {}^wP_2) \qquad (2.6)$$

La matrice de rotation intermédiaire wR_e est plus complexe à calculer. Si la matrice de rotation 1R_2 entre F_1 et F_2 correspond au vecteur U_{12} directeur de l'axe de rotation et à l'angle de rotation θ_{12} , wR_e a pour expression :

$${}^wR_e = {}^wR_1 \cdot R(U_{12}, \theta_{12}/2) \qquad (2.7)$$

De là, l'on a pu en déduire les équations, liant les torseurs cinématiques, données par le schéma symétrique. Le torseur cinématique de F_e est donné par :

$${}^eT_e = \frac{1}{2}({}^eW_1T_1 + {}^eW_2T_2) \qquad (2.8)$$

tandis que le torseur cinématique relatif entre les deux organes terminaux a pour expression :

$${}^eT_r = {}^eW_2T_2 - {}^eW_1T_1 \qquad (2.9)$$

Dans les travaux de Uchiyama et Dauchez [Uchiyama and Dauchez, 1988, Dauchez et al., 1992], l'orientation avait été représentée par les angles d'Euler qui souffrent de singularités, notamment du *blocage de Cardan* tandis que la position était représentée classiquement par un vecteur de \mathbb{R}^3 . D'ailleurs, les calculs avaient été faits dans l'hypothèse

d'une faible différence d'orientation entre les organes terminaux, ce qui restreignait le domaine de validité. Caccavale et al. [Caccavale et al., 2000] ont donc proposé l'utilisation du *quaternion* unitaire pour représenter l'orientation dans ce formalisme. Adorno [Adorno et al., 2010, Adorno, 2011] a suggéré récemment l'utilisation d'un seul objet mathématique, le quaternion *dual* pour représenter les poses, les torseurs cinématiques et les efforts (force et moment)¹. Il a aussi fourni les calculs pour la transformations des torseurs, le modèle géométrique direct, ainsi que le jacobien du robot. La coordination entre un bras de robot et un bras humain a été aussi étudiée.

2.2.3 Commande par impédance

La commande par "*impédance*" a aussi été appliquée au problème de la coordination et plus particulièrement à la comanipulation [Caccavale et al., 2008]. Cette approche de la commande permet d'éviter les fortes contraintes sur l'objet manipulé. La commande par impédance prend quelque peu le contrepied du contrôle hybride force/position [Craig and Raibert, 1979].

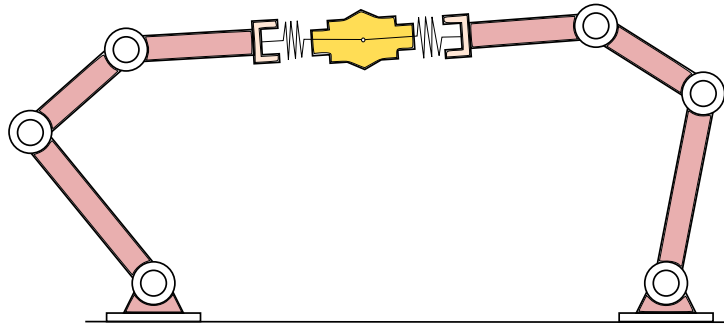


FIGURE 2.4 – Coordination et contrôle par impédance.

Elle ne découple pas le contrôle de la position de celui des forces mais plutôt renforce et explicite le lien entre le mouvement et les forces exercées. Schneider et Cannon [Schneider and Cannon, 1992] ont modélisé le problème en fixant des "*ressorts*" virtuels entre les organes terminaux et le centre de masse de l'objet à manipuler pour chaque degré de liberté à contrôler. Les forces à appliquer aux extrémités du ressort sont calculées en fonction du comportement de l'objet souhaité et de l'impédance. Ensuite un contrôleur dynamique est utilisé pour commander les bras pour obtenir ces forces aux organes terminaux. L'inertie de l'objet, la raideur (*stiffness*) et l'amortissement (*damping*) des ressorts doivent être judicieusement choisis pour obtenir le comportement dynamique désiré. La figure 2.4 très simplifiée tente d'illustrer l'approche.

D'autres travaux ont tenté ensuite d'améliorer le contrôle de la dynamique des bras [Bonitz and Hsia, 1993, Bonitz and Hsia, 1996], se sont intéressés aux objets déformables [Albrichsfeld, 1996, Albrichsfeld and Tolle, 2002] ou ont pris en compte la flexibilité des articulations des robots manipulateurs [Wimböck and Ott, 2012].

2.2.4 Commande basée sur la vision

La vision a déjà été utilisée avec succès pour contrôler des robots et les amener à une situation désirée [Espiau et al., 1992, Hutchinson et al., 1996]. La commande basée sur la vision est souvent appliquée au cas d'un seul bras manipulateur [Chaumette and Hutchinson, 2006, Chaumette and Hutchinson, 2007]. Et à notre connaissance peu de

1. Précisons que ce quaternion dual n'est pas un objet mathématique récent

travaux l'ont utilisée pour des manipulations à deux bras, voire réaliser une coordination. Nous proposons ci-après un bref survol de la littérature.

Miyabe et al. [Miyabe et al., 2003] amènent un manipulateur à deux bras flexibles, conçu pour opérer dans l'espace, à saisir un objet. L'asservissement visuel 2D est utilisé pour commander chaque bras séparément en vue d'atteindre une situation propice à la saisie de l'objet. Pour la saisie et la comanipulation, ils mettent en place une vraie coordination à l'aide d'un contrôle hybride position/force. Cependant, l'étape d'asservissement visuel, où les deux bras sont amenés à une situation relative par rapport à l'objet adéquate pour la prise, ne satisfait pas la définition que nous avons retenue pour la coordination. En effet bien que les bras soient commandés simultanément, aucun lien explicite n'a été défini entre leurs mouvements.

Dauchez et al. [Dauchez et al., 2005] s'intéressent à l'assemblage de deux objets similaires à un tenon et une mortaise par un robot humanoïde. Celui-ci possède deux caméras reliées à la tête. Lorsque les deux objets sont en contact, un contrôle hybride position/force est mis en œuvre. Pour la phase d'approche, ils proposent une idée *intéressante* que nous adapterons dans le prochain chapitre : il s'agit de considérer les deux bras comme un unique robot série, dont la base serait un des organes terminaux, et l'autre bout le deuxième organe terminal. Ils commandent ce nouveau système robotique en position pour atteindre la pose relative souhaitée entre les deux organes terminaux. De cette manière, il mettent en œuvre une *réelle coordination asymétrique*. La vision n'est utilisée ici que pour contrôler la tête afin de garder les deux organes terminaux, dans les champs de vue des caméras, contribuant indirectement à la coordination.

Hynes et al. [Hynes et al., 2005, Hynes et al., 2006] quant à eux, s'intéressent à l'automatisation de la chirurgie mini-invasive. Rappelons que dans ce type de chirurgie, des outils chirurgicaux longs et fins sont introduits par des incisions étroites dans le corps du patient. Leur objectif est d'évaluer l'intérêt d'utiliser l'asservissement visuel pour réaliser des tâches de manipulation à deux bras dans ce contexte particulier [Hynes et al., 2005]. Ainsi dans [Hynes et al., 2006] la tâche considérée consiste à effectuer des nœuds chirurgicaux à l'aide de deux bras manipulateurs. Ceux-ci sont commandés par asservissement visuel selon une séquence établie. Ils sont actionnés parfois simultanément, le plus souvent alternativement, mais de manière indépendante.

Vahrenkamp et al. [Vahrenkamp et al., 2009b] utilisent l'asservissement visuel pour réaliser deux tâches de coordination différentes. Dans l'une, le robot saisit un wok avec les deux mains en utilisant un asservissement visuel 3D. Il manipule ensuite l'objet en continuant à commander les bras de manière indépendante, mais en adaptant légèrement les références pour réduire les efforts internes. Dans l'autre, le robot transvase un liquide mais une pose absolue est spécifiée pour chaque bras, en sorte que les bras sont toujours commandés indépendamment. La réalisation de la coordination au sens où nous l'entendons n'est donc pas convaincante. Nous mentionnons enfin les travaux de Maytin-Shepard et al [Maytin-Shepard et al., 2010] qui ont permis au robot PR2, de plier des serviettes en utilisant les deux bras. Un asservissement visuel a été utilisé mais n'a pas été décrit, l'accent ayant été mis sur la détection des coins de tissu.

2.3 Conclusion

La manipulation à deux bras présente un grand intérêt pour les robots industriels, humanoïdes ou anthropomorphes. La coordination des bras est la prise en compte de contraintes imposées physiquement par une chaîne cinématique fermée ou la mise en place de contraintes "artificielles" par une technique de commande entre les bras manipulateurs. Un formalisme symétrique a été proposé pour le cas de la manipulation d'un objet tenu rigide par plusieurs bras manipulateurs. Ce formalisme permet un contrôle hybride

force/position sans une architecture maître/esclave entre les bras. Une approche duale, le formalisme des tâches coopératives a permis d'exprimer un ensemble plus large de tâches coordonnées à partir de variables décrivant des poses relatives ou absolues des organes terminaux. Plusieurs autres techniques de commande ont été employées pour résoudre le problème de la coordination, se focalisant le plus souvent sur la comanipulation. Parmi ces approches, à notre connaissance, peu de travaux ont utilisé la vision pour mettre en place une coordination entre des bras manipulateurs. Pourtant, l'asservissement visuel offre une bonne stabilité et une grande robustesse aux erreurs de modélisation [Chaumette and Hutchinson, 2006]. Il permettrait également de s'adapter à la manière dont les objets auraient été saisis. Au prochain chapitre, nous rappelons les fondements de l'asservissement visuel avant de proposer un formalisme de coordination basé sur cette technique de commande. Nous utilisons ensuite ce formalisme pour réaliser la tâche de manipulation coordonnée souhaitée : refermer automatiquement un stylo.

Chapitre 3

La coordination par asservissement visuel

"*Yon sèl dwèt pa manje kalalou*" -
peut être compris ainsi : il faut plus
d'un doigt pour maintenir un objet.
L'union fait la force.

Proverbe haïtien

3.1 Introduction

La commande basée sur la vision ou l'asservissement visuel consiste à utiliser des données visuelles pour contrôler le mouvement d'un robot. Ces données sont fournies par une caméra qui peut être embarquée (configuration "*eye-in-hand*") ou fixe dans l'environnement (on parle alors de caméra déportée ou configuration "*eye-to-hand*"). Ces deux configurations sont illustrées sur la figure 3.1.

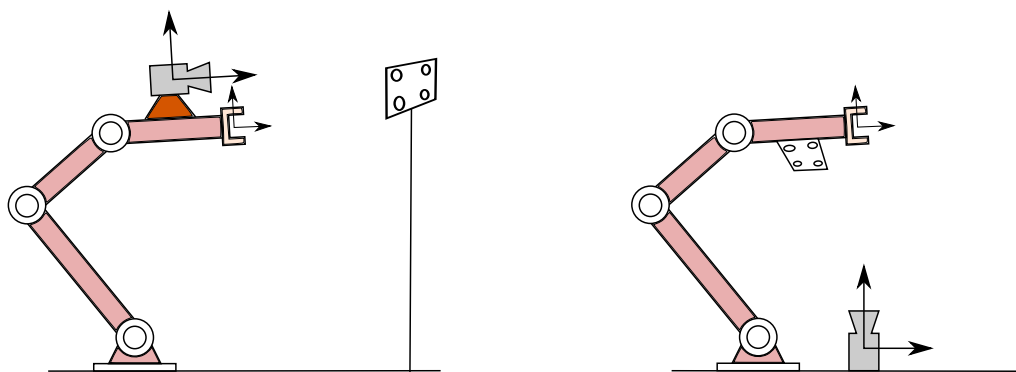


FIGURE 3.1 – Deux configurations : caméra embarquée et caméra déportée

L'asservissement visuel s'inscrit dans la commande référencée capteur qui décrit la tâche, non en fonction de l'état du robot, mais en fonction des informations fournies par le capteur [Chaumette, 1990, Espiau et al., 1992, Hutchinson et al., 1996, Chaumette and Hutchinson, 2006] mesurant l'interaction du robot avec l'environnement. Une tâche est en général définie par l'erreur relative entre la valeur désirée et celle fournie par le capteur. Le robot est ensuite commandé de manière à réguler à zéro cette erreur. Dans le cadre de l'asservissement visuel, le capteur est une caméra. La commande basée vision se divise grossièrement en deux approches [Hutchinson et al., 1996, Chaumette and Hutchinson, 2006] : l'asservissement visuel 3D ou PBVS (*Position Based Visual Servoing*) où la caméra

est utilisée comme un capteur 3D qui fournit la pose d'une cible par rapport à la caméra, et l'asservissement visuel 2D ou IBVS (*Image Based Visual Servoing*) où les informations 2D extraites des images sont utilisées directement dans la boucle de commande. Le PBVS nécessite de disposer a priori d'un modèle 3D de la cible, et permet de bien maîtriser les trajectoires effectuées par le robot, bien que la commande puisse amener la cible à sortir du champ de vue. L'IBVS ne nécessite pas ce modèle 3D, est robuste aux erreurs de modélisation, mais le mouvement de l'image imposé peut se traduire par des trajectoires "étranges" dans l'espace [Chaumette and Hutchinson, 2006, Chaumette and Hutchinson, 2007]. Des approches hybrides telles que l'asservissement visuel 2D1/2 [Malis et al., 1999] existent également et tentent de réunir les avantages des deux techniques précédemment mentionnées. Cependant, il est important de noter que toutes les approches proposées cherchent à atteindre une pose souhaitée entre la caméra et la cible. En d'autres termes, elles permettent de rejoindre une pose souhaitée de l'organe terminal d'un bras par rapport à un repère monde. Il suffit d'amener, une première fois, l'organe terminal à la pose souhaitée par rapport à la cible, et enregistrer le vecteur d'indices visuels obtenus comme référence dans un vecteur s^* . C'est un avantage certain pour l'automatisation des tâches de manipulation. Elles peuvent être réalisées tout en permettant une variabilité dans la manière de saisir les objets. C'est donc pour nous affranchir de la nécessité d'un modèle 3D des objets et bénéficier de sa robustesse au bruit et aux erreurs de modélisation que nous choisissons l'asservissement visuel 2D pour mettre en place la coordination entre les bras manipulateurs.

Nous avons vu dans le chapitre précédent que peu de travaux utilisent l'asservissement visuel pour la manipulation à deux bras. Dans les lignes qui suivent, nous rappelons brièvement les concepts et les principales équations de l'asservissement visuel 2D dans le cas d'une tâche impliquant un seul robot. Ensuite nous proposons notre approche pour la coordination de deux bras manipulateurs, pour enfin l'appliquer à notre cas : la fermeture d'un stylo.

3.2 Asservissement visuel 2D

L'asservissement visuel 2D utilise des informations visuelles géométriques extraites des images fournies par une ou plusieurs caméras. Ces informations visuelles sont rangées dans un vecteur noté s , le vecteur des indices visuels. Les indices visuels peuvent être les coordonnées des points dans l'image, correspondant à des points dans l'espace, ou des caractéristiques géométriques représentant des lignes, des ellipses, des sphères ou des cylindres, sans s'y limiter. Le but de l'asservissement visuel est de minimiser l'erreur entre les vecteurs d'indices visuels s et désirés s^* . Cette erreur a pour expression :

$$e(t) = s(q(t), t) - s^*(t) \quad (3.1)$$

q étant un vecteur décrivant l'état du robot. Le vecteur $s^*(t)$ des indices visuels désirés est généralement pris constant. Mais dans certaines applications, comme la planification de chemin dans l'espace d'image [Mezouar and Chaumette, 2002a], ce vecteur est variable. Le formalisme des fonctions de tâches, développé dans [Samson et al., 1991] pour fournir un cadre général et rigoureux à la commande des robots manipulateurs, s'applique à l'asservissement visuel. La fonction d'erreur e est donc définie comme une fonction de tâche. L'on construit généralement une commande en vitesse pour le robot pour réguler l'erreur à zéro. Dans la plupart des cas, on commande le robot de sorte que l'erreur décroisse exponentiellement, alors la vitesse désirée \dot{e}^* s'écrit, avec un gain λ défini par un réel positif ou une matrice définie positive :

$$\dot{e}^* = -\lambda e \quad (3.2)$$

Nous pouvons calculer la dérivée par rapport au temps de la fonction de tâche e . Elle a pour expression :

$$\dot{e}(t) = \dot{s}(t) - \dot{s}^*(t) \quad (3.3)$$

La dérivée $\dot{s}^*(t)$ est nulle dans le cas général. Dans le cas contraire elle est imposée, s'il s'agit de planification dans l'image [Mezouar and Chaumette, 2002b]. La dérivée \dot{s} par rapport au temps du vecteur des indices visuels, est liée au mouvement de la caméra ou de la cible. Cette relation peut s'exprimer ainsi [Chaumette, 2002] :

$$\dot{s} = L_s \cdot ({}^cT_c - {}^cT_o) \quad (3.4)$$

où

- L_s est la *matrice d'interaction* qui relie la variation des indices visuels dans l'image de la caméra au *torseur cinématique* de la caméra.
- cT_c ou T_c est le *torseur cinématique* de la caméra par rapport à un repère fixe, exprimé dans le repère F_c lié à la caméra, en son origine C .
- cT_o est le torseur cinématique qu'aurait un point solidaire de la cible, qui coïnciderait à l'instant avec l'origine du repère F_c . Ce torseur est exprimé dans ce même repère. Généralement, la cible est fixe et le torseur est nul, dans la configuration "*eye-in-hand*", la plus courante. Dans l'autre configuration "*eye-to-hand*", la cible est mue par un robot et la caméra fixe. Lorsque la caméra et la cible sont toutes deux mobiles, il s'agit d'un problème de poursuite de cible et son torseur doit être estimé [Chaumette et al., 1991, Papanikolopoulos et al., 1993, Allen et al., 1993].

Si la caméra est *embarquée*, cT_c est lié à nT_n , le torseur cinématique de l'organe terminal dans son repère F_n , par cW_n , la matrice de transformation cinétique de torseurs cinématiques du repère F_n vers le repère F_c .

$${}^cT_c = {}^cT_n = {}^cW_n {}^nT_n \quad (3.5)$$

Si la caméra est *déportée*, c'est le torseur cinématique de l'objet qui est lié à celui de l'organe terminal.

$${}^cT_o = {}^cT_n = {}^cW_n {}^nT_n \quad (3.6)$$

Le torseur cinématique nT_n de l'organe terminal est classiquement donné par le modèle cinématique du robot. On a donc :

$${}^nT_n = J_n \dot{q} \quad (3.7)$$

avec

- \dot{q} , la dérivée temporelle des coordonnées articulaires du robot.
- J_n le jacobien du robot, exprimé dans le repère F_n de l'organe terminal du robot.

En utilisant les équations précédentes : (3.3), (3.4), (3.5) et (3.7), dans le cas d'une cible fixe et d'une caméra embarquée, la dérivée de l'erreur peut s'exprimer en fonction de la vitesse des articulations du robot :

$$\dot{e} = L_s {}^cW_n J_n \dot{q} \quad (3.8)$$

En remplaçant la dérivée de la fonction de tâche par son expression dans (3.2) nous obtenons :

$$L_s {}^cW_n J_n \dot{q} = -\lambda e \quad (3.9)$$

La commande en vitesse du robot permettant d'obtenir la dynamique souhaitée, est donc donnée par la pseudo inverse de $L_s^c W_n J_n$:

$$\dot{q} = -[L_s^c W_n J_n]^+ \lambda e \quad (3.10)$$

Dans la configuration (*eye-to-hand*), la commande s'exprime de manière similaire :

$$\dot{q} = [L_s^c W_n J_n]^+ \lambda e \quad (3.11)$$

Ces commandes permettent de contrôler un bras robotique unique pour positionner une caméra embarquée par rapport à une cible fixe, ou pour positionner une cible mobile, par rapport à une caméra fixe déportée. Il est également possible d'utiliser les informations visuelles fournies par plusieurs caméras pour commander le robot. On parle alors d'asservissement visuel multi-caméras. Mentionnons les travaux de Malis et al. [Malis et al., 2000, Malis et al., 2001], qui ont mis en œuvre un asservissement visuel multi-caméras. Celles-ci étaient toutes fixées sur le même bras manipulateur pour atteindre une pose souhaitée par rapport à une porte de grande dimension. Pour contrôler ce robot, ils ont choisi de définir une fonction de tâche classique pour chaque caméra et d'utiliser une pondération de leurs lois de commande respectives. D'autres travaux [Flandin et al., 2000, Lippiello et al., 2005, Kermorgant and Chaumette, 2011c] ont fait coopérer des caméras embarquées et des caméras déportées comme illustré sur la figure 3.2 pour encore positionner l'organe terminal du robot par rapport à un objet fixe.

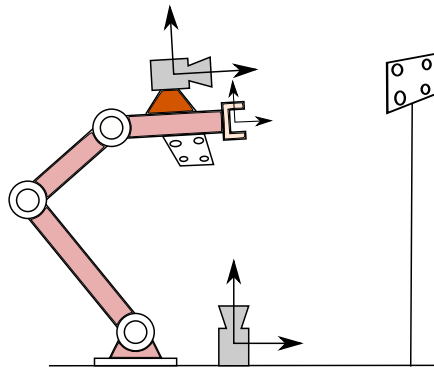


FIGURE 3.2 – Coopération entre caméra embarquée et caméra déportée.

Flandin et al. [Flandin et al., 2000] ont utilisé cette configuration pour découpler la position et l'orientation de la caméra. Kermorgant et Chaumette [Kermorgant and Chaumette, 2011c] ont regroupé de préférence les indices visuels des caméras dans un seul vecteur et calculé la loi de commande correspondant à cette tâche.

Il y a donc un grand nombre de configurations possibles pour l'emplacement des caméras et des objets d'intérêt, et plusieurs approches pour la commande dans ces situations. Mais une caractéristique est commune à toutes ces réalisations : le but poursuivi est d'atteindre une pose de l'organe terminal par rapport à un repère monde fixe inertiel. Dans les sections qui suivent, nous utiliserons l'asservissement visuel basé 2D pour atteindre des poses absolues et des poses relatives entre des caméras, bras manipulateurs ou des objets d'intérêt.

3.3 La coordination par l'asservissement visuel

Les travaux de Chiacchio et d'Adorno [Chiacchio et al., 1996, Adorno, 2011] sur la coordination entre deux bras, ont montré que ces tâches de manipulation, peuvent être

définies avec deux variables : la pose relative entre les organes terminaux et la pose intermédiaire entre ces deux organes. Nous étudions le moyen de transposer ce formalisme des tâches coopératives au contexte de l'asservissement visuel 2D. Nous considérons un système de deux bras fixés sur une base immobile.

Nous distinguons le bras gauche et le bras droit. Les coordonnées articulaires des bras gauche et droit seront notés respectivement q_g et q_d . Nous représentons sur la figure 3.3 les différents repères nécessaires à notre étude. F_w désigne le repère du monde, tandis que F_g et F_d caractérisent ceux liés aux organes terminaux gauche et droit. Les torseurs cinématiques T_g et T_d respectifs de F_g et F_d par rapport au repère monde, sont exprimés respectivement dans les repères F_g et F_d . Les modèles géométriques et cinématiques direct des deux bras sont connus. Nous avons donc :

$$T_g = J_g \dot{q}_g \qquad T_d = J_d \dot{q}_d \qquad (3.12)$$

où J_g et J_d représentent les jacobiniennes des bras gauche et droit exprimées respectivement dans F_g et F_d . Dans le cas d'un solide manipulé par les deux bras, le repère lié au solide sera noté F_o et son torseur cinématique par rapport au monde est T_o .

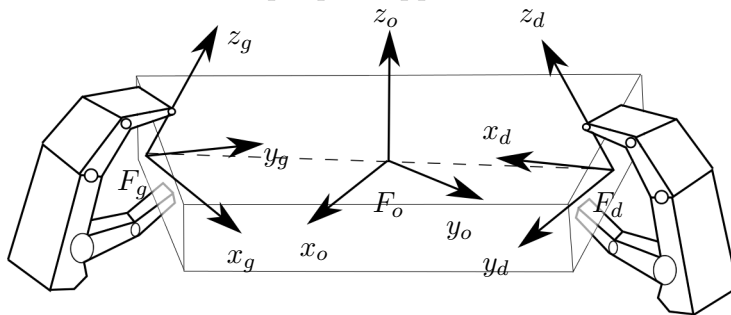


FIGURE 3.3 – Repère liés aux organes terminaux et à l'objet

Sans perte de généralité, nous étudierons le problème de la coordination de deux bras en utilisant les informations visuelles d'une seule caméra. Nous verrons ensuite plus loin comment prendre en compte les cas où plusieurs capteurs visuels sont disponibles. Le repère lié à la caméra est noté F_c et nous considérerons trois cas de figures :

- la caméra est fixe par rapport au repère monde, elle est dite *déportée* (Figure 3.4).
- la caméra est *mobile* et solidaire d'un des deux bras manipulateurs (Figure 3.6).
- la caméra est *mobile* mais solidaire d'un troisième bras (Figure 3.7)

Notre premier choix est de considérer les deux bras comme un unique système robotique, comme l'ont proposé Uchiyama et Dauchez [Uchiyama and Dauchez, 1988, Dauchez et al., 2005]. Nous supposons que les bras du robot sont contrôlés en vitesse. Notre vecteur de commande est donc constitué des vitesses articulaires des deux bras, soit :

$$\dot{q} = \begin{bmatrix} \dot{q}_g \\ \dot{q}_d \end{bmatrix} \qquad (3.13)$$

La loi de commande sera calculée dans l'espace des vitesses articulaires des deux bras. De cette manière, nous jouissons de la redondance offerte par les deux bras. Si une articulation d'un bras est en singularité ou non disponible pour une tâche, une autre articulation même dans l'autre bras peut compenser cette perte momentanée de degrés de liberté.

Notre objectif étant de réaliser la coordination au moyen de l'asservissement visuel, nous définissons dans les prochaines sections des fonctions d'erreur à partir des indices visuels. Nous construisons les commandes appropriées pour le système formé par les deux bras, en considérant les deux configurations de caméras et les deux types de tâches coordonnées. Nous considérons d'abord le cas de la caméra déportée.

3.3.1 Caméra déportée

Dans cette configuration, illustrée sur la figure 3.4, la caméra est fixe par rapport à la base du robot. Nous supposons, pour nos calculs, que tous les objets manipulés restent toujours visibles.

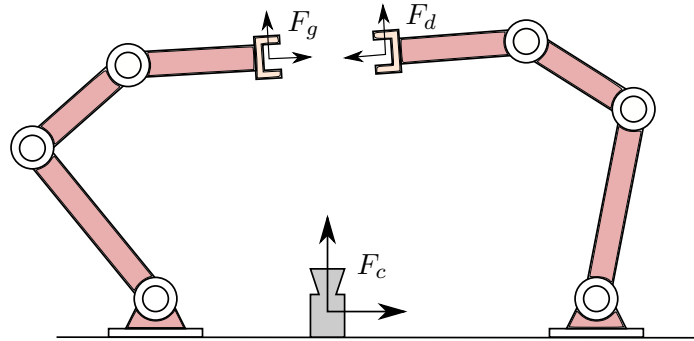


FIGURE 3.4 – Caméra déportée par rapport aux deux bras

Nous nous intéressons d'abord au cas de la comanipulation.

3.3.1.1 Comanipulation

Dans le cas de la comanipulation, nous notons s , le vecteur d'indices visuels décrivant l'image de l'objet dans la caméra. Le but est d'atteindre une position absolue de l'objet [Chiacchio et al., 1996, Adorno et al., 2010] par rapport au repère monde. Dans cette pose désirée, le vecteur d'indices visuels est constant et est noté s^* . Nous pouvons donc définir notre fonction de tâche de manière classique, par l'erreur e :

$$e = s - s^* \quad (3.14)$$

La loi de commande classique, pour la caméra déportée, dans le cas d'un seul bras, est donnée par l'équation (3.11). Comme nous en avons deux, il est nécessaire de les coordonner. Pour cela, nous suivons une démarche similaire à l'approche traditionnellement utilisée et nous calculons d'abord le jacobien de cette fonction de tâche. Nous pouvons exprimer la dérivée du vecteur d'indices visuels en utilisant l'équation (3.4) :

$$\dot{s} = -L_s^c T_o \quad (3.15)$$

$$\dot{s} = -L_s^c W_o T_o \quad (3.16)$$

Le solide et la saisie étant supposés rigides, les organes terminaux et le solide sont solidaires entre eux. Le repère F_o peut donc être choisi comme le repère intermédiaire F_e défini dans le formalisme des tâches coopératives à la section 2.2.2. Nous pouvons alors écrire :

$$T_o = {}^oW_g T_g \quad (3.17)$$

$$T_o = {}^oW_d T_d \quad (3.18)$$

et en utilisant l'équation (2.8) :

$$T_o = \frac{1}{2} {}^oW_g T_g + \frac{1}{2} {}^oW_d T_d \quad (3.19)$$

$$T_o = \frac{1}{2} \begin{bmatrix} {}^oW_g & {}^oW_d \end{bmatrix} \begin{bmatrix} T_g \\ T_d \end{bmatrix}$$

Nous pouvons ainsi relier le mouvement des indices visuels \dot{s} dans l'image aux vitesses articulaires :

$$\dot{s} = -\frac{1}{2}L_s {}^cW_o \begin{bmatrix} {}^oW_g & {}^oW_d \end{bmatrix} \begin{bmatrix} T_g \\ T_d \end{bmatrix} \quad (3.20)$$

Et en utilisant les modèles cinématiques des bras (3.12) :

$$\dot{s} = -\frac{1}{2}L_s {}^cW_o \begin{bmatrix} {}^oW_g & {}^oW_d \end{bmatrix} \begin{bmatrix} J_g & 0 \\ 0 & J_d \end{bmatrix} \begin{bmatrix} \dot{q}_g \\ \dot{q}_d \end{bmatrix} \quad (3.21)$$

Nous définissons ces deux matrices :

$$W_e = -\frac{1}{2} {}^cW_o \begin{bmatrix} {}^oW_g & {}^oW_d \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} {}^cW_g & {}^cW_d \end{bmatrix} \quad (3.22)$$

$$J_{gd} = \begin{bmatrix} J_g & 0 \\ 0 & J_d \end{bmatrix} \quad (3.23)$$

W_e rappelle donc la matrice de préhension introduite dans le schéma symétrique [Uchiyama and Dauchez, 1988, Dauchez et al., 1992]. Elle est d'ailleurs de rang 6 comme cette dernière. Notons de plus qu'elle n'est pas fonction du repère F_o choisi et qu'elle ne dépend que des repères F_g , F_d et F_c . L'expression de la dérivée des indices visuels s'écrit donc, en utilisant l'égalité (5.49) :

$$\dot{s} = L_s W_e J_{gd} \cdot \dot{q} \quad (3.24)$$

Alors la dérivée de la fonction d'erreur est :

$$\dot{e} = L_s W_e J_{gd} \cdot \dot{q} \quad (3.25)$$

Par conséquent, la loi de commande, pour une décroissance exponentielle, est donnée par :

$$\dot{q} = -[L_s W_e J_{gd}]^+ \lambda e \quad (3.26)$$

avec λ un gain ou une matrice définie positive, comme dans l'asservissement visuel classique. Le terme $L_s W_e J_{gd}$ est le jacobien de la fonction d'erreur e . Son rang maximum est de 6, donc il contraint au plus 6 degrés de liberté du robot.

3.3.1.2 Coordination asymétrique

Nous considérons maintenant les tâches d'assemblage. Nous supposons deux objets saisis fermement par les deux organes terminaux et décrits par la caméra déportée par les vecteurs d'indices visuels s_g et s_d avec leur matrice d'interaction correspondante L_g et L_d . Les vecteurs s_g et s_d correspondent aux objets manipulés par les bras gauche et droit respectivement. Notre objectif est d'atteindre la pose relative souhaitée entre les deux organes terminaux [Chiacchio et al., 1996, Adorno et al., 2010]. Celle-ci est obtenue en comparant leur pose absolue. Il est donc raisonnable de vouloir comparer les indices visuels s_g et s_d , ce qui conduirait à la fonction de tâche suivante :

$$e(t) = s_g(t) - s_d(t) \quad (3.27)$$

Pour que la fonction de tâche ainsi définie s'annule, lorsque la pose relative souhaitée est atteinte, il faut que cette erreur ait un sens. Les vecteurs d'indices visuels s_g et s_d doivent être choisis dans ce but. Nous considérons dans le cadre de ce document le modèle sténopé de la caméra [Ma et al., 2001]. Pour comparer les objets manipulés, nous devons donc nous intéresser aux invariants de la géométrie projective. Cette dernière ne

conserve ni les angles, ni les longueurs, ni les rapports. Par contre elle conserve le *rapport anharmonique* et la colinéarité [Eiden, 2009, Ingraio, 2011, Hartley and Zisserman, 2003]. Il faut se rappeler de plus, dans le modèle sténopé de la caméra, la projection d'un point dans l'image est un point, et celle d'une droite est une droite sauf dans un cas dégénéré [Ma et al., 2001]. Cela signifie que si deux points sont confondus dans l'espace, leurs projections seront aussi confondues dans l'image. Il en est de même pour les droites. Une conséquence est que si deux segments de droite sont alignés dans l'espace, leurs projections seront aussi alignées dans le plan image. Il nous faut donc identifier les points ou coins visibles des objets qui doivent entrer en contact. L'on peut également recenser les droites ou segments de droites comme les arêtes des deux objets qui sont alignées dans la pose relative désirée. Donc le jeu d'indices visuels que nous devons retenir, doit décrire la projection dans l'image de ces points, droites ou segments de droite.

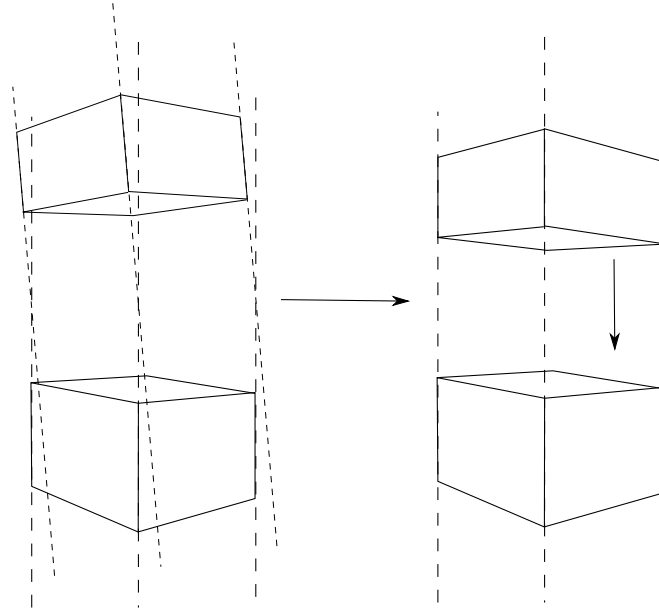


FIGURE 3.5 – Assemblage de deux parallélépipèdes.

La figure 3.5 illustre la manière dont nous pouvons nous assurer d'avoir la pose désirée entre deux parallélépipèdes. À gauche, les arêtes visibles des objets ne sont pas alignées. Aligner ces segments, revient à réaliser une liaison virtuelle de type prismatique entre les deux objets [Chaumette, 1990]. Pour contraindre totalement la pose relative entre les deux cubes, l'on peut exiger que certains sommets soient en contact. On peut donc facilement mettre en œuvre la plupart des liaisons virtuelles décrites par Chaumette dans sa thèse [Chaumette, 1990]. Nous détaillerons certaines de ces liaisons dans la section 3.3.4. Revenons à notre fonction d'erreur en supposant que les indices visuels sont bien choisis. La dérivée de la fonction d'erreur est donnée par :

$$\dot{e} = \dot{s}_g - \dot{s}_d \quad (3.28)$$

En utilisant (3.4) et le modèle cinématique des bras du robot (3.12), en se rappelant que la caméra est déportée, on obtient :

$$\dot{e} = -L_g {}^c T_g + L_d {}^c T_d \quad (3.29)$$

$$\dot{e} = \begin{bmatrix} -L_g & L_d \end{bmatrix} \begin{bmatrix} {}^c W_g & 0 \\ 0 & {}^c W_d \end{bmatrix} \begin{bmatrix} T_g \\ T_d \end{bmatrix} \quad (3.30)$$

$$\dot{e} = \begin{bmatrix} -L_g & L_d \end{bmatrix} \begin{bmatrix} {}^c W_g & 0 \\ 0 & {}^c W_d \end{bmatrix} \begin{bmatrix} J_g & 0 \\ 0 & J_d \end{bmatrix} \begin{bmatrix} \dot{q}_g \\ \dot{q}_d \end{bmatrix} \quad (3.31)$$

Nous définissons les matrices suivantes :

$$L_r = \begin{bmatrix} -L_g & L_d \end{bmatrix} \quad (3.32)$$

$${}^cW_{gd} = \begin{bmatrix} {}^cW_g & 0 \\ 0 & {}^cW_d \end{bmatrix} \quad (3.33)$$

Par conséquent, la dérivée de la fonction d'erreur a pour expression :

$$\dot{e} = L_r {}^cW_{gd} J_{gd} \cdot \dot{q} \quad (3.34)$$

La commande pour une décroissance exponentielle de l'erreur peut donc être déduite :

$$\dot{q} = -[L_r {}^cW_{gd} J_{gd}]^+ \lambda e \quad (3.35)$$

Le jacobien de la fonction de tâche est ici de rang maximal 6. En définissant ainsi notre fonction de tâche pour la coordination asymétrique, la pose absolue des objets au moment du *rendez-vous*, est inconnue et volontairement non spécifiée. Cela a pour effet de réduire le nombre de degrés de liberté requis pour notre fonction de tâche et donc de nous permettre de bénéficier au mieux de la redondance pour réaliser des objectifs complémentaires ou prendre en compte des contraintes nécessaires (cf. chapitre 4).

3.3.2 Caméra embarquée

La caméra est maintenant solidaire de l'organe terminal du robot. Il s'agit de l'une des configurations les plus fréquentes dans la littérature de l'asservissement visuel. Néanmoins, le cas de la comanipulation est difficile à aborder dans cette configuration. En effet la caméra est solidaire des deux organes terminaux, tout comme l'objet manipulé. L'image de l'objet est fixe et les indices visuels sont constants. Il n'y a alors aucune information sur la pose absolue de l'objet par rapport au repère monde. Nous étudions donc la mise en œuvre d'une **coordination asymétrique**. Dans un tel cas, il est préférable de contrôler la pose relative entre les organes terminaux, en utilisant une boucle d'asservissement classique [Dauchez et al., 2005].

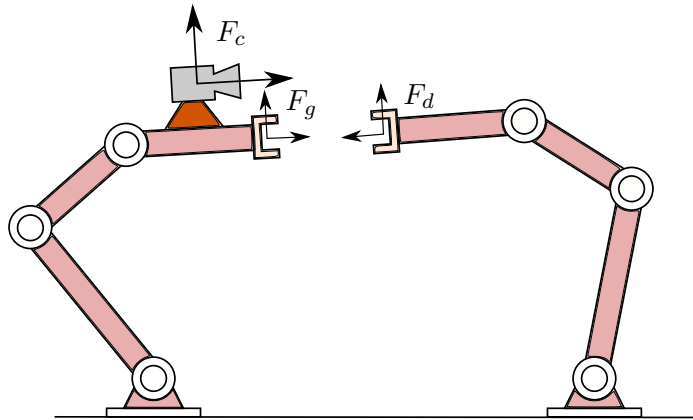


FIGURE 3.6 – Caméra solidaire de l'organe terminal d'un des bras.

Supposons sans perte de généralité que la caméra est fixée sur le bras gauche comme sur la figure 3.6. Si la caméra perçoit les objets manipulés par les deux bras, seul l'objet porté par le bras droit est animé d'un mouvement relatif, tandis que l'autre est immobile. Quelle que soit la pose absolue des deux bras, si la pose relative souhaitée est atteinte, les indices visuels décrivant l'objet du bras droit sont constants et peuvent être notés s_d^* . Il n'est donc pas nécessaire pour la tâche que l'objet manipulé par le bras gauche soit

visible par la caméra. Il n'est pas non plus obligatoire de choisir les indices visuels comme préconisé pour la caméra déportée. Nous retrouvons donc un cas d'asservissement visuel classique avec néanmoins une cible mobile, puisque l'objet qui est perçu est mû par le bras droit. Nous pouvons donc définir notre fonction d'erreur comme suit :

$$e = s_d - s_d^* \quad (3.36)$$

La dérivée temporelle de la fonction d'erreur donne, en utilisant l'équation (3.4) :

$$\dot{e} = \dot{s}_d = L_d \cdot ({}^cT_c - {}^cT_d) \quad (3.37)$$

Comme nous l'avons montré, la caméra et la cible sont mobiles. Cette dernière étant constituée par l'objet manipulé par le bras droit, son torseur et donc son mouvement est connu. On a alors :

$$\dot{s}_d = L_d \cdot ({}^cW_g T_g - {}^cW_d T_d) \quad (3.38)$$

$$\dot{s}_d = L_d \cdot \begin{bmatrix} {}^cW_g & -{}^cW_d \end{bmatrix} \begin{bmatrix} T_g \\ T_d \end{bmatrix} \quad (3.39)$$

$$\dot{s}_d = L_d \cdot \begin{bmatrix} {}^cW_g & -{}^cW_d \end{bmatrix} \begin{bmatrix} J_g & 0 \\ 0 & J_d \end{bmatrix} \begin{bmatrix} \dot{q}_g \\ \dot{q}_d \end{bmatrix} \quad (3.40)$$

En définissant la matrice :

$$W_r = \begin{bmatrix} {}^cW_g & -{}^cW_d \end{bmatrix} \quad (3.41)$$

il est possible d'écrire :

$$\dot{e} = L_d \cdot W_r \cdot J_{gd} \cdot \dot{q} \quad (3.42)$$

La commande sur les deux bras permettant d'annuler cette fonction d'erreur a pour expression :

$$\dot{q} = -[L_d W_r J_{gd}]^+ \lambda e \quad (3.43)$$

3.3.3 Caméra embarquée indépendante

Une troisième configuration peut se présenter sur des plateformes robotiques anthropomorphes. La caméra est solidaire de la tête du robot qui possède quelques degrés de liberté. Nous modélisons cette situation par l'ajout d'un troisième bras robotique. Dans ce cas, la caméra est fixée sur l'organe terminal de ce troisième bras, comme sur la figure 3.7. Cette configuration est un hybride des deux premières, la caméra est fixée sur un bras, mais pas sur ceux concernés directement par la tâche de manipulation. Dans cette configuration, il est possible de contrôler la pose de la caméra, pour garder les objets dans son champ de vue [Dauchez et al., 2005, Agravante et al., 2013]. Grâce à notre approche, le mouvement de la caméra peut être coordonné avec celui des deux autres bras manipulateurs.

Nous notons F_t le repère lié à l'organe terminal de ce troisième bras, T_t son torseur cinématique par rapport au repère monde et exprimé dans F_t et q_t le vecteur des coordonnées articulaires associées. Les modèles géométrique et cinématique de ce bras supplémentaire sont supposés connus et son jacobien exprimé dans F_t est noté J_t :

$$T_t = J_t \dot{q}_t \quad (3.44)$$

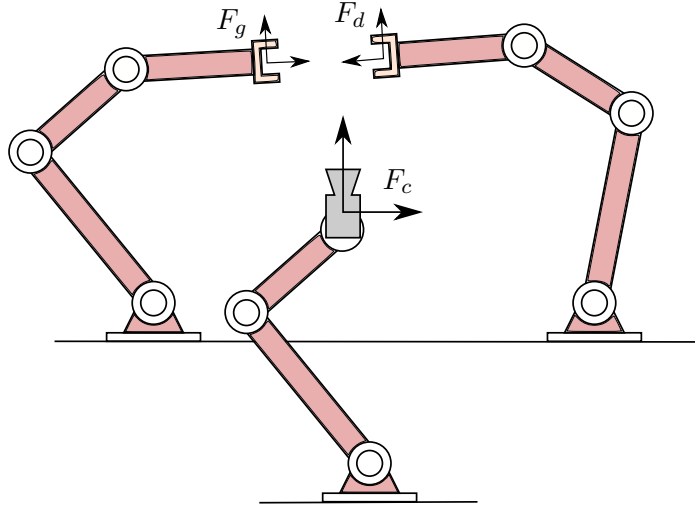


FIGURE 3.7 – Caméra active, quand le champ de vue de la caméra varie

3.3.3.1 Comanipulation

Par rapport aux deux bras manipulateurs, la caméra peut être considérée comme déportée. Mais pour cette configuration, la comanipulation présente certaines difficultés. Puisque la caméra est mobile, il est difficile de garantir que l'objet manipulé atteigne une pose absolue souhaitée. Pour cela, l'on peut comparer l'objet à un autre objet fixe dans la scène, ou à un vecteur d'indices visuels constant s^* et imposer que l'objet soit vu d'une certaine manière. La fonction d'erreur est définie de manière classique :

$$e = s - s^* \quad \dot{e} = \dot{s} \quad (3.45)$$

La dérivée du vecteur d'indices visuels est cette fois-ci différente. L'équation (3.4) appliquée à notre cas, nous permet d'écrire :

$$\dot{s} = L_s \cdot ({}^cT_c - {}^cT_o) \quad (3.46)$$

$$\dot{s} = L_s \cdot ({}^cW_t T_t - {}^cW_o T_o) \quad (3.47)$$

En remplaçant le torseur T_o par son expression (3.19), nous obtenons :

$$\dot{s} = L_s \cdot ({}^cW_t T_t - {}^cW_o \frac{1}{2} ({}^oW_g T_g + {}^oW_d T_d)) \quad (3.48)$$

$$\dot{s} = L_s \begin{bmatrix} {}^cW_t & -\frac{1}{2} {}^cW_g & -\frac{1}{2} {}^cW_d \end{bmatrix} \begin{bmatrix} T_t \\ T_g \\ T_d \end{bmatrix} \quad (3.49)$$

En utilisant les modèles cinématiques des différents bras (3.12), et (3.44), l'expression devient :

$$\dot{s} = L_s \begin{bmatrix} {}^cW_t & -\frac{1}{2} {}^cW_g & -\frac{1}{2} {}^cW_d \end{bmatrix} \begin{bmatrix} J_t & 0 & 0 \\ 0 & J_g & 0 \\ 0 & 0 & J_d \end{bmatrix} \begin{bmatrix} \dot{q}_t \\ \dot{q}_g \\ \dot{q}_d \end{bmatrix} \quad (3.50)$$

$$\dot{s} = L_s \begin{bmatrix} {}^cW_t & W_e \end{bmatrix} \begin{bmatrix} J_t & 0 \\ 0 & J_{gd} \end{bmatrix} \begin{bmatrix} \dot{q}_t \\ \dot{q} \end{bmatrix} \quad (3.51)$$

La loi de commande pour les trois bras a donc pour expression :

$$\begin{pmatrix} \dot{q}_t \\ \dot{q} \end{pmatrix} = - \left[L_s \begin{pmatrix} {}^cW_t & W_e \end{pmatrix} \begin{pmatrix} J_t & 0 \\ 0 & J_{gd} \end{pmatrix} \right]^+ \lambda e \quad (3.52)$$

3.3.3.2 Coordination asymétrique

La tâche de coordination asymétrique n'offre pas plus de difficulté dans cette configuration que lorsque la caméra est fixe. La fonction d'erreur compare, comme pour la caméra déportée, les deux vecteurs d'indices visuels, et a donc pour expression :

$$e = s_g - s_d \quad (3.53)$$

Les indices visuels doivent être choisis comme pour la caméra déportée pour pouvoir atteindre la pose relative souhaitée entre les deux objets. La dérivée de la fonction d'erreur dépend aussi du torseur de la caméra. En utilisant (3.4) et le modèle cinématique des bras du robot (3.12) (3.44), on obtient cette fois-ci :

$$\dot{e} = L_g({}^cW_tT_t - {}^cW_gT_g) - L_d({}^cW_tT_t - {}^cW_dT_d) \quad (3.54)$$

$$\dot{e} = \begin{bmatrix} L_g & -L_d \end{bmatrix} \begin{bmatrix} {}^cW_tT_t - {}^cW_gT_g \\ {}^cW_tT_t - {}^cW_dT_d \end{bmatrix} \quad (3.55)$$

$$\dot{e} = \begin{bmatrix} L_g & -L_d \end{bmatrix} \begin{bmatrix} {}^cW_t & -{}^cW_g & 0 \\ {}^cW_t & 0 & {}^cW_d \end{bmatrix} \begin{bmatrix} T_t \\ T_g \\ T_d \end{bmatrix} \quad (3.56)$$

$$\dot{e} = \begin{bmatrix} L_g & -L_d \end{bmatrix} \begin{bmatrix} {}^cW_t & -{}^cW_g & 0 \\ {}^cW_t & 0 & -{}^cW_d \end{bmatrix} \begin{bmatrix} J_t & 0 & 0 \\ 0 & J_g & 0 \\ 0 & 0 & J_d \end{bmatrix} \begin{bmatrix} \dot{q}_t \\ \dot{q}_g \\ \dot{q}_d \end{bmatrix} \quad (3.57)$$

$$\dot{e} = -L_r \begin{bmatrix} {}^cW_t & -{}^cW_g & 0 \\ {}^cW_t & 0 & -{}^cW_d \end{bmatrix} \begin{bmatrix} J_t & 0 \\ 0 & J_{gd} \end{bmatrix} \begin{bmatrix} \dot{q}_t \\ \dot{q} \end{bmatrix} \quad (3.58)$$

L'expression pour la commande des trois bras est donnée par :

$$\begin{pmatrix} \dot{q}_t \\ \dot{q} \end{pmatrix} = \left[L_r \begin{pmatrix} {}^cW_t & -{}^cW_g & 0 \\ {}^cW_t & 0 & -{}^cW_d \end{pmatrix} \begin{pmatrix} J_t & 0 \\ 0 & J_{gd} \end{pmatrix} \right]^+ \lambda e \quad (3.59)$$

3.3.4 Liaisons virtuelles entre objets

Nous avons comparé, pour réaliser une coordination asymétrique, les indices visuels décrivant l'image des objets manipulés par les deux bras. Nous avons trouvé que le rang des fonctions de tâches définies était limité par le rang de la matrice L_r , inférieur ou égal à 6. Cette dernière matrice a pour expression $\begin{bmatrix} L_g & -L_d \end{bmatrix}$. Son rang dépend donc principalement du choix des indices visuels. Nous avons précisé, dans la sous-section 3.3.1.2, que pour atteindre une pose relative entre deux objets, les indices visuels doivent être choisis judicieusement. Les indices décrivant des points, des droites ou des segments de droite sont les candidats les plus prometteurs.

Mais la pose relative recherchée peut ne pas être complètement spécifiée ou contrainte. Par exemple pour ouvrir une bouteille, la pose relative entre les deux bras est partiellement contrainte, mais une rotation doit être possible autour du bon axe. Dans cet exemple, une liaison rotoïde *virtuelle* doit donc être mise en place entre les deux bras. Elle est dite *virtuelle* car non imposée par un système mécanique, mais mise en œuvre par une loi de commande. Les liaisons virtuelles que l'on peut réaliser entre une caméra et un objet de la scène ont été étudiées par Chaumette et Espiau [Chaumette, 1990, Espiau et al., 1992]. Ils ont présenté des exemples d'indices visuels ou de motifs pour réaliser différentes liaisons [Chaumette, 1990]. Ils les ont caractérisées par la structure du noyau des matrices d'interaction correspondant aux choix d'indices visuels. Nous décrivons ici quelques choix d'indices visuels pour mettre en œuvre différentes liaisons virtuelles, dans le cadre de notre

coordination asymétrique, grâce à notre approche en asservissement visuel 2D. Dans notre cas, les liaisons virtuelles sont mises en place entre des objets alors qu'elles sont définies entre la caméra et un objet de la scène dans les cas proposés par Chaumette [Chaumette, 1990].

3.3.4.1 Liaison rigide

Une liaison rigide peut être mise en place, en choisissant un ensemble d'indices visuels qui donnera une matrice d'interaction de rang plein.

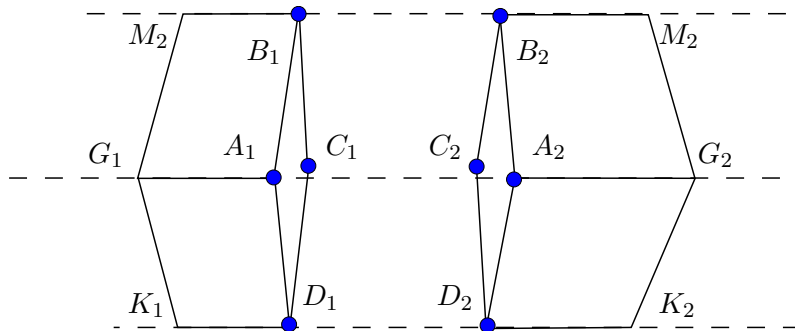


FIGURE 3.8 – Liaison virtuelle rigide entre deux parallélépipèdes

Sur la figure 3.8 apparaissent deux parallélépipèdes qui peuvent être mis en contact et qui ont deux dimensions en commun, comme la largeur et la hauteur. Pour assembler ces objets, une première possibilité est de choisir quatre points non colinéaires appartenant aux deux objets. Une fonction de tâche peut comparer les vecteurs d'indices visuels composés des images des sommets A_1 , B_1 , C_1 et D_1 pour l'objet de gauche et les images des sommets A_2 , B_2 , C_2 et D_2 pour l'objet de droite. Si cette fonction est nulle, les deux cubes sont en contact, et parfaitement alignés.

Mais les points C_1 et C_2 risquent de ne pas être visibles pendant l'exécution de la tâche. Un autre choix peut être d'aligner deux groupes de quatre segments de droites non parallèles, entre les deux objets. Aligner les arêtes A_1B_1 et A_2B_2 , A_1D_1 et A_2D_2 , A_1G_1 et A_2G_2 , D_1K_1 et D_2K_2 . De nombreux autres choix de motifs sont envisageables. Pour ces choix d'indices visuels, un seul point de vue, c'est-à-dire l'utilisation d'une seule caméra, suffit pour réaliser la liaison rigide souhaitée.

3.3.4.2 Liaison prismatique

Pour réaliser une liaison prismatique, il suffit d'aligner dans l'image trois paires de segments de droite qui sont parallèles dans l'espace. Si nous voulons mettre en œuvre une liaison prismatique entre les deux objets de la figure 3.8, il suffit d'aligner les arêtes qui sont sur les lignes en tiret. Trois segments parallèles sur chaque objet suffisent à réaliser la liaison, en utilisant une seule caméra : les segments B_1M_1 et B_2M_2 , A_1G_1 et A_2G_2 , D_1K_1 et D_2K_2 . Si l'on dispose de deux points de vue sur les deux objets, c'est-à-dire de deux caméras, deux segments sont suffisants.

3.3.4.3 Liaison pivot glissant

Une liaison virtuelle pivot glissant peut être mise en place entre deux cylindres de même diamètres comme sur la figure 3.9, pour réaliser cette liaison. Si les cylindres sont décrits chacun, dans l'image d'une caméra, par les deux segments de leur contour, il faut aligner ces segments comme sur la figure 3.9. Cette liaison peut être transformée en une liaison rotoïde, en imposant que les extrémités des segments se touchent. Si nous

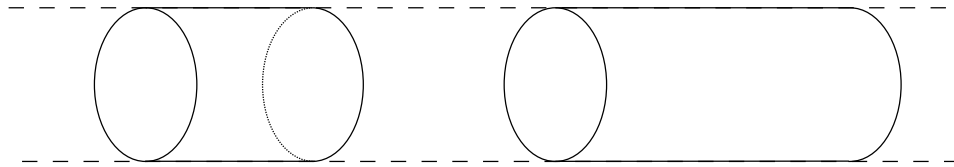


FIGURE 3.9 – Liaison virtuelle entre deux cylindres

considérons les parallélépipèdes de la figure 3.8, une liaison virtuelle rotoïde peut être aussi réalisée, en utilisant deux caméras et en alignant un segment de chacun des objets qui devront se toucher par une extrémité.

3.3.5 Conclusion

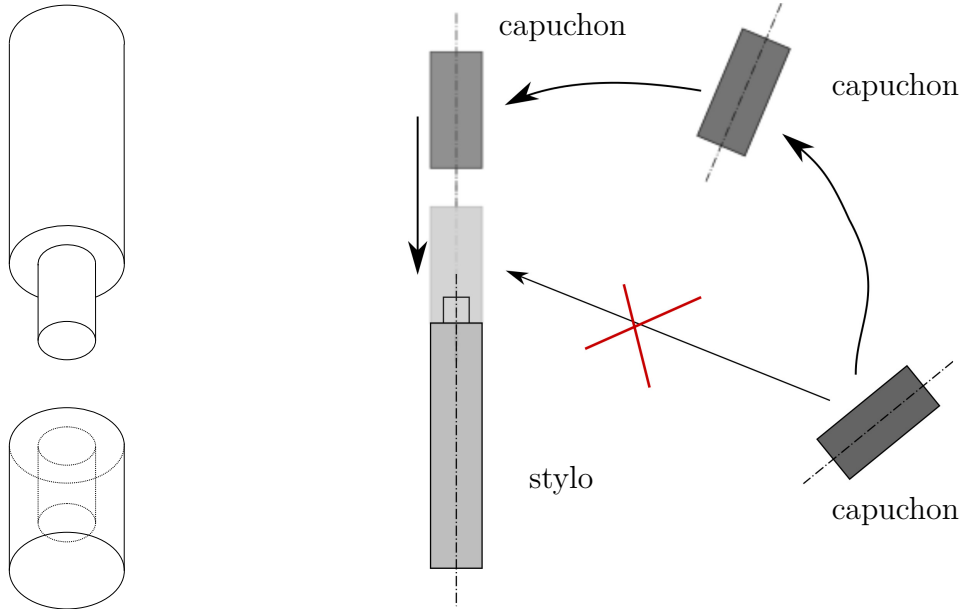
Nous avons proposé un ensemble d'outils (*framework*) pour mettre en œuvre la coordination entre deux bras, en couplant l'asservissement visuel 2D et le formalisme des tâches coopératives. Nous avons étudié différentes configurations définies par l'emplacement de la caméra. Pour ces configurations, nous avons défini les fonctions d'erreur appropriées et élaboré les commandes correspondantes pour les deux types de tâches coordonnées existantes. Les commandes proposées ne suivent pas le schéma maître/esclave ou suiveur/suivant, tout comme le formalisme symétrique [Uchiyama and Dauchez, 1988]. Pour l'asservissement visuel 2D, nous avons précisé les types d'indices visuels qui peuvent être utilisés pour positionner relativement un objet par rapport à un autre. Dans l'élan, nous avons proposé des choix d'indices visuels pour réaliser des liaisons virtuelles entre deux objets. Nous avons également montré que tous les jacobiens utilisés pour le calcul des commandes sont au plus de rang 6, ce qui veut dire que nous n'utilisons que 6 degrés de liberté pour réaliser une tâche coordonnée. Les systèmes robotiques multi-bras disposant généralement d'un grand nombre de degrés de liberté, notre approche nous permet de bénéficier de la redondance naturelle du système robotique pour réaliser des objectifs secondaires, en sus de la tâche de manipulation coordonnée. Ce n'est pas le cas d'autres travaux qui exploitent l'asservissement visuel dans le contexte de la coordination [Hynes et al., 2006, Vahrenkamp et al., 2009b, Boissinot, 2008, Khalfaoui, 2009]. En effet, ils calculent une loi de commande indépendante pour chaque bras, ce qui signifie qu'ils utilisent 6 degrés de liberté sur chacun des bras et qu'ils ne peuvent exploiter par là-même la redondance offerte par le système. Il s'agit d'un aspect important de notre contribution : les tâches et les lois de commande développées conservent la redondance apportée par le système. Dans les lignes qui suivent, nous appliquons l'approche que nous venons d'exposer à une tâche d'assemblage témoin : la fermeture d'un stylo. Nous aborderons l'utilisation de la redondance dans le chapitre suivant.

3.4 Application : refermer un stylo

Nous nous proposons de refermer un stylo en utilisant l'approche précédente. Nous supposons que le stylo est saisi par le bras gauche tandis que le capuchon est manipulé par le bras droit. Ces objets sont modélisés comme des cylindres de même diamètre représentés sur la figure 3.10a.

Cet assemblage ne peut pas se faire en un seul mouvement. Nous avons donc décomposé la tâche à réaliser en une séquence de trois sous-tâches illustrées sur la figure 3.10b :

- Rendre coplanaires les axes des cylindres.
- Aligner les deux cylindres tout en maintenant une certaine distance entre eux.
- Garder le stylo et le capuchon alignés tout en les rapprochant jusqu'au contact.



(a) Modèles du stylo et du capuchon

(b) Enchaînement de mouvements.

FIGURE 3.10 – Sur la figure, le stylo reste fixe juste par souci de lisibilité, mais rien ne l’empêche de se déplacer.

3.4.1 Présentation du système robotique considéré et des caméras utilisées

3.4.1.1 Système robotique

Nous utilisons le robot PR2 de la plateforme robotique du LAAS-CNRS. Ce robot comporte deux bras de 7 degrés de liberté montés sur une base mobile (cf. figure 3.11). Nous considérons ici que la base et la tête sont fixes et que le robot est commandé en vitesse. Le robot possède deux paires de caméras sur la tête et une caméra fixée à chaque avant-bras¹. Nous avons utilisé deux caméras. La première C_t fixée sur la tête joue le rôle d’une caméra déportée et offre un large champ de vue. La deuxième de centre optique C_m , positionnée sur l’avant-bras droit du robot, est mobile et permet de bénéficier d’un second point de vue sur la scène.

Nous avons conservé des notations similaires à celles que nous avons utilisées pour définir le formalisme de coordination. Nous introduisons les différents repères représentés sur la figure 3.11. F_w est le repère du monde. F_g et F_d sont les repères des organes terminaux respectivement gauche et droit du robot. F_t et F_m sont les repères liés aux caméras fixe et mobile susmentionnées, leurs origines respectives sont C_t et C_m . Nous définissons par q_d et q_g (respectivement \dot{q}_d et \dot{q}_g) les configurations (respectivement les vitesses articulaires) des bras droit et gauche.

À l’aide du modèle cinématique direct, il est possible d’exprimer les torseurs cinématiques de la caméra mobile et des organes terminaux gauche et droit par rapport à F_w :

$$T_d = J_d \dot{q}_d, \quad T_g = J_g \dot{q}_g, \quad T_m = J_{5d} \dot{q}_r \quad (3.60)$$

où J_r, J_l et J_{5d} sont des matrices de dimension 6×7 fournies par le modèle cinématique. Ces torseurs sont exprimés respectivement dans F_r, F_l et F_m .

1. La caméra n’est pas fixée sur l’organe terminal, mais sur le corps qui le précède dans la chaîne cinématique.

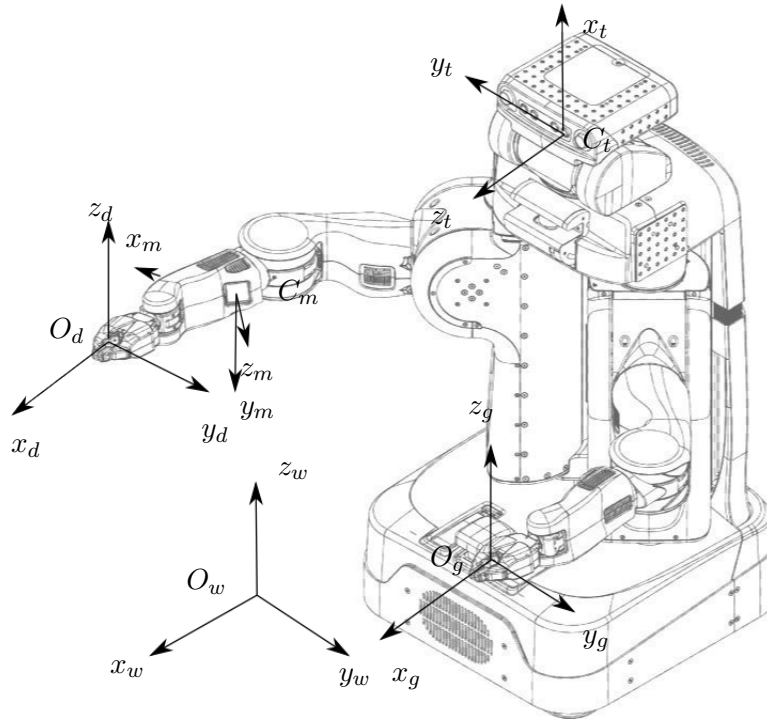


FIGURE 3.11 – Placement des repères sur le PR2.

3.4.1.2 Modèle de la caméra

Puisque nous exploitons l'asservissement visuel 2D, nous devons préciser le modèle de caméra utilisé. Les caméras du robot peuvent être décrites avec le modèle sténopé de la caméra et illustré par la figure 3.12.

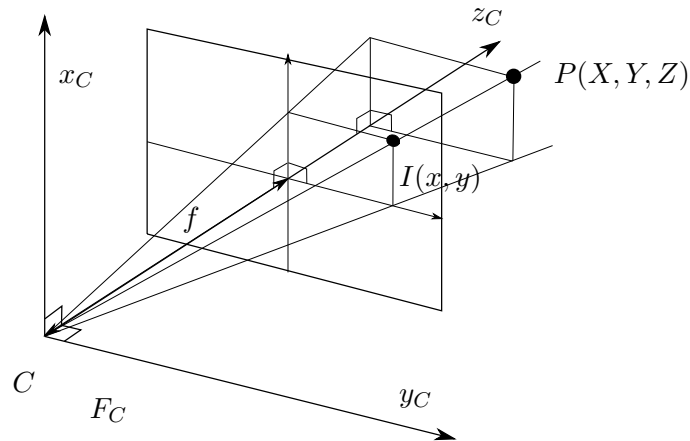


FIGURE 3.12 – Modèle sténopé de la caméra.

Dans ce modèle l'image d'un point M de coordonnées (X, Y, Z) dans le repère de la caméra F_c est un point I de coordonnées (x, y) dans le plan image. x et y sont donnés par :

$$\begin{cases} x = fX/Z = (u - c_u)\alpha_u \\ y = fY/Z = (v - c_v)\alpha_v \end{cases}$$

Avec

- (u, v) les coordonnées en pixels du point image.
- $c_u, c_v, f, \alpha_u, \alpha_v$ sont les paramètres intrinsèques de la caméra où
 - f est la focale,
 - (c_u, c_v) étant les coordonnées en pixels du centre optique dans l'image

- α_u et α_v les dimensions du capteur correspondant au pixel sur la rétine de la caméra.

Nous souhaitons réaliser une tâche de coordination asymétrique en utilisant une caméra embarquée et une caméra déportée. Notre stratégie est donc multi-caméras.

3.4.2 Choix des indices visuels

Notre objectif ici est de choisir un ensemble d'indices visuels modélisant le stylo et le capuchon. Nous avons vu en 3.3.1.2 que pour atteindre une pose relative entre les deux objets, il est nécessaire de chercher des droites ou segments de droites qui sont alignés dans la pose souhaitée. Nous supposons que les cylindres sont vus de côté, tout au long du déroulement de la tâche. Si les caméras perçoivent leurs faces, les images obtenues s'apparentent à des ellipses. Alors les centres optiques des caméras sont proches de l'axe du cylindre, ce qui réduit l'information que l'on peut obtenir sur leur orientation.

Inspirés par Berry [Berry, 1999, Berry et al., 2000], nous avons choisi d'approximer la projection de l'axe du cylindre par l'axe d'inertie de la région de l'image correspondant au cylindre. Nous obtenons ainsi la droite représentée sur la figure 3.13. Pour la paramétrer,

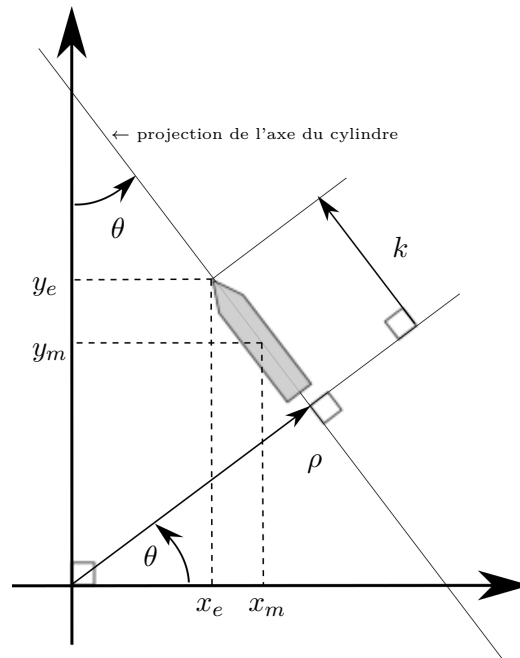


FIGURE 3.13 – Indices visuels retenus : ρ , θ et k .

nous avons utilisé la représentation polaire (ρ, θ) présentée dans [Chaumette, 2002]. ρ définit la distance entre la droite et l'origine du repère, et θ l'orientation de la droite par rapport à la verticale (cf. fig 3.13). Pour déterminer les valeurs de ρ et θ , nous avons adapté à notre cas les formules données dans [Berry et al., 2000].

$$\theta = \frac{1}{2} \operatorname{atan} \left(\frac{2I_{xy}}{I_x - I_y} \right)$$

$$\rho = x_m \cdot \cos(\theta) + y_m \cdot \sin(\theta)$$

où (x_m, y_m) est le centre de masse de la région en gris sur la figure 3.13. I_{xy}, I_x, I_y sont les

moments centrés d'ordre deux de la région grisée, et ont pour expression :

$$\begin{aligned}
 A &= \iint_{\text{aire}} dx \cdot dy & x_m &= \frac{1}{A} \iint_{\text{aire}} x \, dx \cdot dy & y_m &= \frac{1}{A} \iint_{\text{aire}} y \, dx \cdot dy \\
 I_x &= \iint_{\text{aire}} (x - x_m)^2 \, dx \cdot dy & I_y &= \iint_{\text{aire}} (y - y_m)^2 \, dx \cdot dy & I_{xy} &= \iint_{\text{aire}} (x - x_m)(y - y_m) \, dx \cdot dy
 \end{aligned}$$

Ces informations sont relativement faciles à obtenir avec des traitements d'images. Les indices visuels ρ et θ permettent de contrôler l'orientation et partiellement la position de chaque objet. Cependant il est aussi nécessaire de contrôler la translation du cylindre sur son axe. Pour cela, nous introduisons un troisième indice nous permettant d'exprimer la position d'une extrémité du cylindre sur la ligne (cf. fig-3.13). En notant E (x_e, y_e) cette extrémité, k s'écrit :

$$k = y_e \cdot \cos(\theta) - x_e \cdot \sin(\theta) \quad (3.61)$$

Nous avons ainsi représenté le stylo et le capuchon par trois paramètres (ρ , θ et k).

Remarque : Cette définition des indices visuels souffre d'une singularité si l'image du cylindre s'approche d'un disque. Dans ce cas, $I_x \simeq I_y$, $I_{xy} \simeq 0$, et l'orientation θ n'est plus définie.

La tâche finale doit réaliser une liaison virtuelle de type rotoïde. Avec notre choix des indices visuels, il nous faut deux points de vue. Nous choisissons donc d'utiliser deux caméras, une embarquée et une déportée. Nous aurons donc quatre vecteurs d'indices visuels définis à partir de ces trois paramètres :

	Capuchon	Stylo
Caméra fixe C_t	${}^t S_c = [{}^t \rho_c \quad {}^t \theta_c \quad {}^t k_c]^T$	${}^t S_p = [{}^t \rho_p \quad {}^t \theta_p \quad {}^t k_p]^T$
Caméra fixe C_m	${}^m S_c = [{}^m \rho_c \quad {}^m \theta_c \quad {}^m k_c]^T$	${}^m S_p = [{}^m \rho_p \quad {}^m \theta_p \quad {}^m k_p]^T$

3.4.3 Modélisation de la tâche et stratégie de commande

Afin de réaliser notre tâche, nous nous proposons d'utiliser la commande référencée vision. Il est donc nécessaire de définir les fonctions de tâche permettant de réaliser les trois sous-tâches mentionnées plus haut.

3.4.3.1 Décomposition en trois sous-tâches

Nous rappelons que la première sous-tâche doit rendre coplanaires les axes des cylindres, tout en gardant une certaine distance entre eux. Pour cela, il suffit d'aligner la projection des axes dans l'image donnée par la caméra embarquée. Nous faisons l'hypothèse que cette caméra est bien placée de manière à voir le capuchon et le stylo. Notre première sous-tâche s'écrit :

$$e_1 = \begin{bmatrix} {}^m \rho_c - {}^m \rho_p \\ {}^m \theta_c - {}^m \theta_p \\ {}^m k_c - {}^m k_p - d \end{bmatrix} \quad (3.62)$$

où d est une constante permettant de conserver un écart entre le capuchon et le stylo.

La deuxième sous-tâche doit aligner les axes des cylindres tout en conservant une distance entre les deux. Avec un seul point de vue, il est impossible de garantir l'alignement de deux droites. Nous devons donc utiliser conjointement les informations issues des deux

caméras. Nous réalisons donc ici un asservissement visuel multi-caméras. Notre deuxième sous-tâche a pour expression :

$$e_2 = \begin{bmatrix} e_1 \\ {}^t\rho_c - {}^t\rho_p \\ {}^t\theta_c - {}^t\theta_p \end{bmatrix} \quad (3.63)$$

Notre dernière sous-tâche doit rapprocher les deux cylindres tout en les gardant alignés. Elle s'écrit :

$$e_3 = \begin{bmatrix} {}^mS_c - {}^mS_p \\ {}^tS_c - {}^tS_p \end{bmatrix} \quad (3.64)$$

e_1 , e_2 , et e_3 s'expriment donc en fonction de tout ou partie des indices visuels représentant le capuchon et le stylo. Il est possible de montrer que ces trois fonctions de tâche peuvent être mises sous la forme générale suivante :

$$e_i = H_i \cdot \begin{bmatrix} {}^mS_c - {}^mS_p - A_i \\ {}^tS_c - {}^tS_p \end{bmatrix} \quad (3.65)$$

où H_i est la matrice d'activation qui permet de considérer uniquement les indices visuels nécessaires à chaque sous-tâche. H_i et A_i sont définis par :

$$\begin{aligned} H_1 &= \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} & \text{et } A_1 &= [0, 0, d]^T \\ H_2 &= \begin{bmatrix} I_{5 \times 5} & 0_{5 \times 1} \end{bmatrix} & \text{et } A_2 &= [0, 0, d]^T \\ H_3 &= I_{6 \times 6} & \text{et } A_3 &= 0_{3 \times 1} \end{aligned}$$

Nous cherchons maintenant à synthétiser la loi de commande permettant d'annuler ces fonctions de tâche.

3.4.3.2 Synthèse de la commande

Pour synthétiser nos lois de commande, nous avons exploité les équations développées dans la section précédente 3.3. Déterminons le jacobien de nos fonctions de tâche. D'après la relation (3.65), la dérivée de e_i par rapport au temps s'écrit :

$$\dot{e}_i = J_i \dot{q}, \quad \dot{e}_i = H_i \cdot \begin{bmatrix} {}^m\dot{S}_c - {}^m\dot{S}_p \\ {}^t\dot{S}_c - {}^t\dot{S}_p \end{bmatrix} \quad (3.66)$$

La caméra mobile n'est pas solidaire de l'organe terminal, même si elle est fixée sur un des bras manipulateurs. Nous ne pouvons donc pas appliquer immédiatement les équations de la section 3.3 pour une partie de la fonction de tâche, mais nous pouvons les adapter. Les équations (3.4) et (3.60) nous permettent d'exprimer la dérivée des indices visuels :

$${}^m\dot{S}_c = {}^mL_c(J_{5d}\dot{q}_d - {}^mW_dJ_d\dot{q}_d) \quad (3.67)$$

$${}^m\dot{S}_p = {}^mL_p(J_{5d}\dot{q}_d - {}^mW_gJ_g\dot{q}_g) \quad (3.68)$$

$${}^m\dot{S}_c - {}^m\dot{S}_p = \begin{bmatrix} {}^mL_c & -{}^mL_p \end{bmatrix} \begin{bmatrix} 0 & -{}^mW_d & I_6 \\ -{}^mW_g & 0 & I_6 \end{bmatrix} \begin{bmatrix} J_g & 0 \\ 0 & J_d \\ 0 & J_{5d} \end{bmatrix} \begin{bmatrix} \dot{q}_g \\ \dot{q}_d \end{bmatrix} \quad (3.69)$$

où mL_c et mL_p sont les matrices d'interaction associées à mS_c et mS_p . Le calcul de ces matrices d'interaction est relégué à l'annexe A.

En suivant un raisonnement similaire, la dérivée de la partie de la fonction de tâche dépendant de la caméra fixe a pour expression :

$${}^t\dot{S}_c - {}^t\dot{S}_p = \begin{bmatrix} {}^tL_c & -{}^tL_p \end{bmatrix} \begin{bmatrix} 0 & -{}^tW_d \\ -{}^tW_g & 0 \end{bmatrix} \begin{bmatrix} J_g & 0 \\ 0 & J_d \end{bmatrix} \begin{bmatrix} \dot{q}_g \\ \dot{q}_d \end{bmatrix}$$

En exploitant ces résultats, le jacobien de chaque fonction de tâche e_i s'écrit :

$$J_i = H_i \begin{bmatrix} {}^mL_c & -{}^mL_p & 0 & 0 \\ 0 & 0 & {}^tL_c & -{}^tL_p \end{bmatrix} \begin{bmatrix} 0 & -{}^mW_d & I_6 \\ -{}^mW_g & 0 & I_6 \\ 0 & -{}^tW_d & 0 \\ -{}^tW_g & 0 & 0 \end{bmatrix} \begin{bmatrix} J_g & 0 \\ 0 & J_d \\ 0 & J_{5d} \end{bmatrix} \quad (3.70)$$

La commande à appliquer pour chaque sous-tâche est donnée par :

$$\dot{q}_i = -J_i^+ \lambda_i e_i \quad (3.71)$$

3.4.3.3 Enchaînement des correcteurs

Nous avons donc défini trois fonctions de tâche différentes que nous réglons à zéro, l'une après l'autre. Dès que la norme d'une tâche descend en dessous d'une valeur choisie au préalable, un superviseur supprime le contrôleur de la tâche courante, et active le suivant. Ce changement de tâche étant instantané, la commande envoyée au robot risque de changer brusquement conduisant les actionneurs des articulations à saturer et induisant par là-même des à-coups dans le mouvement du robot. Il est donc nécessaire de lisser la commande. Plusieurs solutions ont été développées pour gérer la transition entre deux tâches.

Pissard-Gibollet et Rives [Pissard-Gibollet and Rives, 1995] ont proposé d'utiliser une fonction $a(t)$ qui varie continûment entre 0 et 1 en fonction du temps.

$$\dot{q} = (1 - a(t))\dot{q}_A + a(t)\dot{q}_B \quad (3.72)$$

où \dot{q}_A et \dot{q}_B sont les commandes à enchaîner. Mais pendant cette période de transition, l'évolution de la dernière fonction de tâche est inconnue. Une autre solution apportée par Souères et al. en 2003 [Soueres et al., 2003] consistait à définir la dynamique de la fonction de tâche à l'aide d'une équation différentielle du second degré de la forme :

$$\ddot{e} + \alpha\dot{e} + \beta e = 0 \quad (3.73)$$

Les paramètres α et β sont alors à ajuster afin d'obtenir la dynamique souhaitée pour la fonction de tâche. Cette solution présente l'avantage de garantir qu'à l'instant de basculement la loi de commande calculée par les correcteurs précédent et suivant est la même. On parle alors d'enchaînement dynamique car la continuité au premier ordre de la loi de commande est garantie.

Dans le prolongement de ces résultats, Mansard a proposé de définir la dynamique de la fonction de tâche par une équation différentielle du premier ordre, à second membre variable. L'expression retenue est la suivante [Mansard and Chaumette, 2004, Mansard, 2006] :

$$\dot{e}_2^*(t) = -\lambda e_2(t_i) + \exp(-\mu(t - t_i))(\dot{e}_1^*(t_i) + \lambda e_2(t_i)) \quad (3.74)$$

où e_2 suit e_1 et t_i est l'instant de transition. Cette solution est analogue à la précédente car il s'agit encore d'une équation différentielle du second degré

$$\ddot{e} + (\lambda + \mu)\dot{e} + \lambda\mu e = 0 \quad (3.75)$$

Mais elle a le mérite de faire apparaître deux paramètres qui peuvent être ajustés facilement. La période de transition est fonction de $1/\mu$ et doit être plus courte que la période

d'accomplissement de la tâche qui est de l'ordre de $1/\lambda$. Donc μ doit être plus grand que λ . Mansard utilise généralement $\mu = 10\lambda$. Lorsque μ est égal à zéro, nous retrouvons la décroissance exponentielle classique.

Pour notre part, nous avons choisi d'utiliser cette solution, mais en remplaçant le terme $\dot{e}_1^*(t_i)$ dans (3.74) par le terme $\dot{e}_2(t_i)$ qui est plus approprié. En effet, les fonctions de tâche e_1 et e_2 sont différentes, et nous voulons assurer la continuité de la loi de commande. Le terme $\dot{e}_2(t_i)$ est la vitesse de la tâche e_2 à l'instant t_i de la transition. Soit e_i la tâche qui a été activée à l'instant t_i avec le gain λ_i pour remplacer la tâche e_{i-1} . Nous avons alors :

$$\dot{e}_i^*(t) = -\lambda_i e_i(t) + \exp(-\mu(t - t_i))[\dot{e}_i(t_i) + \lambda_i e_i(t_i)] \quad (3.76)$$

La commande devient pour $t \geq t_i$:

$$\dot{q}(t) = -J_i^+ \lambda_i e_i(t) + \exp(-\mu(t - t_i))[\dot{q}(t_i) + J_i^+ \lambda_i e_i(t_i)] \quad (3.77)$$

avec $\dot{q}(t_i)$ la vitesse articulaire due à e_{i-1} à l'instant t_i .

3.4.4 Résultats de simulation

Nous avons testé notre approche pour la fermeture du stylo en simulation. Le diamètre du stylo a été fixé à 2 cm, sa longueur à 7.5 cm tandis que celle du capuchon à 5 cm. Ils ont été placés à 17 cm suivant x dans les repères F_g et F_d liés aux poignets. Nous les avons supposés déjà saisis par les pinces. Nous avons implémenté dans notre programme, les modèles géométrique et cinématique des bras du robot PR2². Les valeurs initiales des coordonnées articulaires des bras du robot sont regroupées dans le tableau 3.1.

Articulation	Bras gauche	Bras droit
shoulder_pan_joint	$\pi/4$	$-\pi/4$
shoulder_lift_joint	0.1	-0.1
upper_arm_roll_joint	$\pi/2$	$-\pi/2$
elbow_flex_joint	$-\pi/2$	$-\pi/2$
forearm_roll_joint	$-\pi/2$	$\pi/2$
wrist_flex_joint	-0.05	-0.05
wrist_roll_joint	0.0	0.0

TABLE 3.1 – Valeurs initiales des coordonnées articulaires des deux bras du PR2 (en radians).

Nous avons considéré un débit de 15 images par seconde provenant des deux caméras. La loi de commande a donc été mise à jour à une fréquence de 15 Hertz. Dans la simulation, nous avons choisi de conserver des paramètres intrinsèques proches de ceux des caméras présentes sur le robot PR2, les paramètres utilisés sont présentés dans le tableau 3.2.

	Caméra sur l'avant bras	Caméra sur la tête
$M = \begin{bmatrix} f/\alpha_u & 0.0 & c_u \\ 0.0 & f/\alpha_v & c_v \end{bmatrix}$	$M = \begin{bmatrix} 320.0 & 0.0 & 320.5 \\ 0.0 & 320.0 & 240.5 \end{bmatrix}$	$M = \begin{bmatrix} 418.68 & 0.0 & 309.05 \\ 0.0 & 419.16 & 224.18 \end{bmatrix}$

TABLE 3.2 – Paramètres intrinsèques des caméras utilisés en simulation.

2. <https://www.willowgarage.com/pages/pr2/overview>

Nous avons séquencé les trois sous-tâches définies e_1 , e_2 et e_3 , en se basant sur la norme. Dès que la norme d'une tâche devient inférieure à 1, la tâche est désactivée, pour laisser place à la suivante. Les distances ρ et k ont été exprimées en pixels, tandis que les angles θ ont été exprimés en moitié de degrés pour pouvoir être représentés sur les figures. Dans ce cas, la constante d de l'équation (3.65) a pour valeur 50 pixels. Les gains λ_i ont été fixés pour les trois sous-tâches à 1 tandis que la constante μ , qui pilote la transition, a été prise égale à 3. Aucun bruit n'a été ajouté pour cette phase de simulation sur les coordonnées articulaires ou sur les indices visuels.

La figure 3.14 présente, dans le même graphe, l'évolution des composantes de trois sous-tâches décrites dans la section précédente. Ces composantes sont ordonnées dans le tableau 3.3.

Composante v_i	e_1	e_2	e_3
v_1	${}^m\rho_c - {}^m\rho_p$	${}^m\rho_c - {}^m\rho_p$	${}^m\rho_c - {}^m\rho_p$
v_2	${}^m\theta_c - {}^m\theta_p$	${}^m\theta_c - {}^m\theta_p$	${}^m\theta_c - {}^m\theta_p$
v_3	${}^mk_c - {}^mk_p - d$	${}^mk_c - {}^mk_p - d$	${}^mk_c - {}^mk_p$
v_4		${}^t\rho_c - {}^t\rho_p$	${}^t\rho_c - {}^t\rho_p$
v_5		${}^t\theta_c - {}^t\theta_p$	${}^t\theta_c - {}^t\theta_p$
v_6			${}^tk_c - {}^tk_p$

TABLE 3.3 – Significations des composantes des sous-tâches d'asservissement visuel.

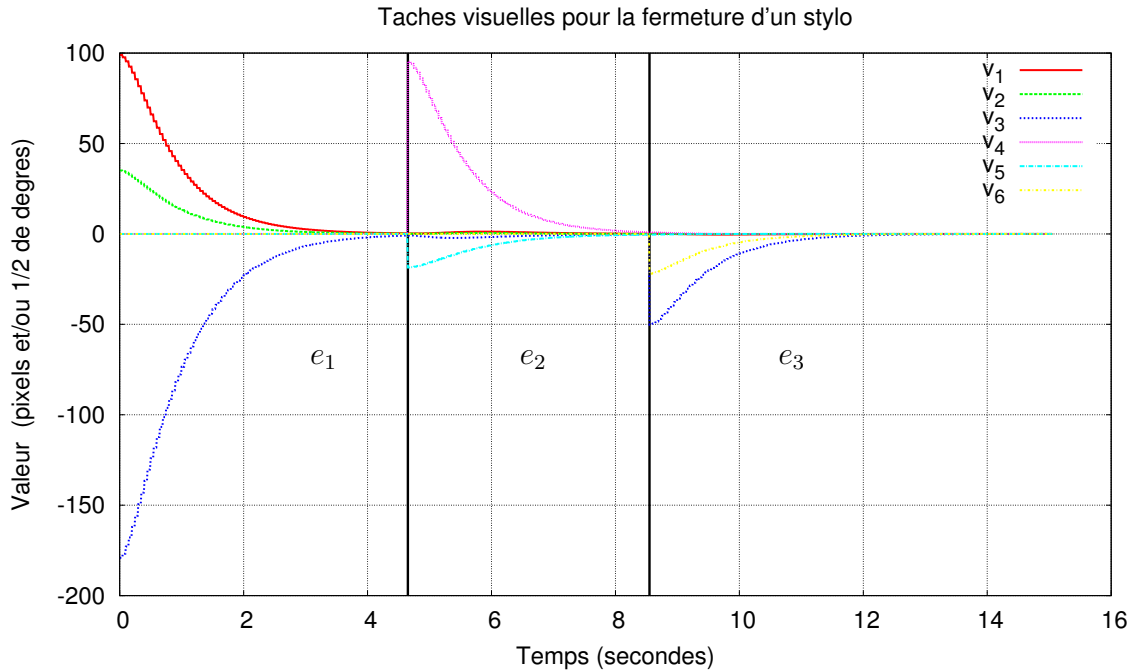


FIGURE 3.14 – Évolution des composantes des trois sous-tâches e_1 , e_2 et e_3 .

La figure 3.14 montre l'évolution des trois sous-tâches et l'ensemble des composantes v_i convergent vers zéro, ce qui montre que les tâches correspondantes sont réalisées. Les instants de transitions à $t = 4.65s$ et $t = 8.55s$, sont matérialisés par deux lignes épaisses noires verticales sur les figures 3.14, 3.15 et 3.16. Les figures 3.15 et 3.16, représentent l'évolution des vitesses envoyées aux deux bras pendant les trois sous-tâches. Elles sont similaires, signe de la coordination efficace réalisée par la stratégie de commande. Elles

sont continues aux instants de transition, démontrant la pertinence du dispositif de lissage (3.77).

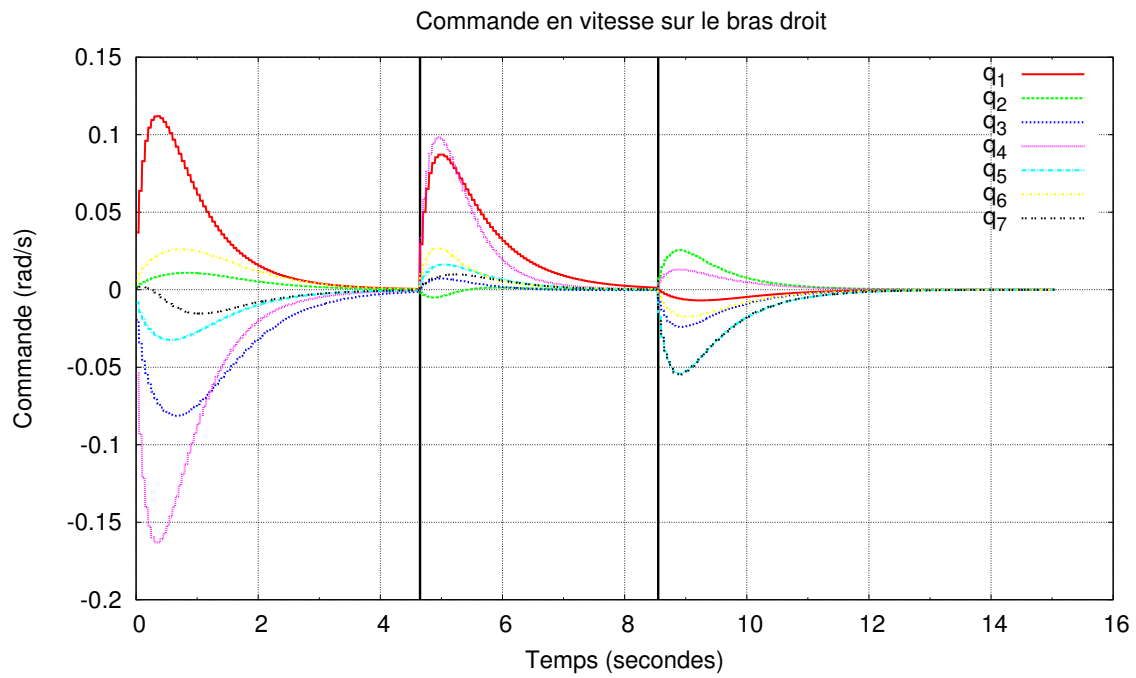


FIGURE 3.15 – Commande en vitesse des 7 articulations du bras droit du PR2.

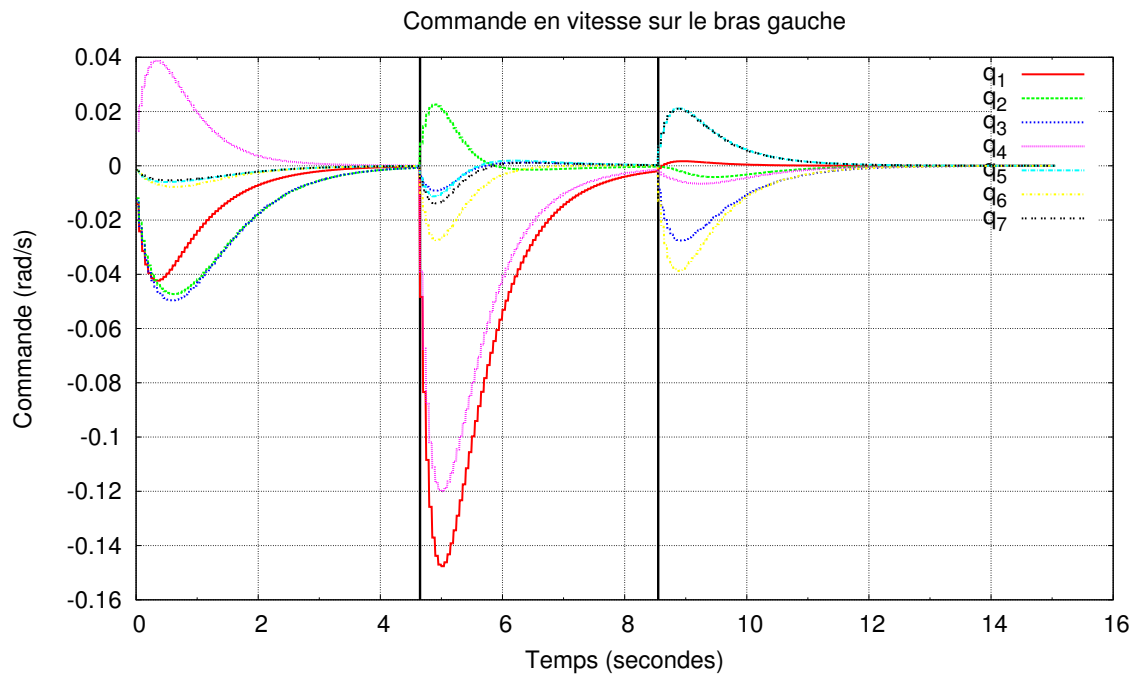


FIGURE 3.16 – Commande en vitesse des 7 articulations du bras gauche du PR2.

Les trajectoires obtenues pour les bras ont été enregistrées puis rejouées sur un modèle du PR2 avec ROS³ et le simulateur Gazebo⁴. La figure 3.17 montre des vues du robot

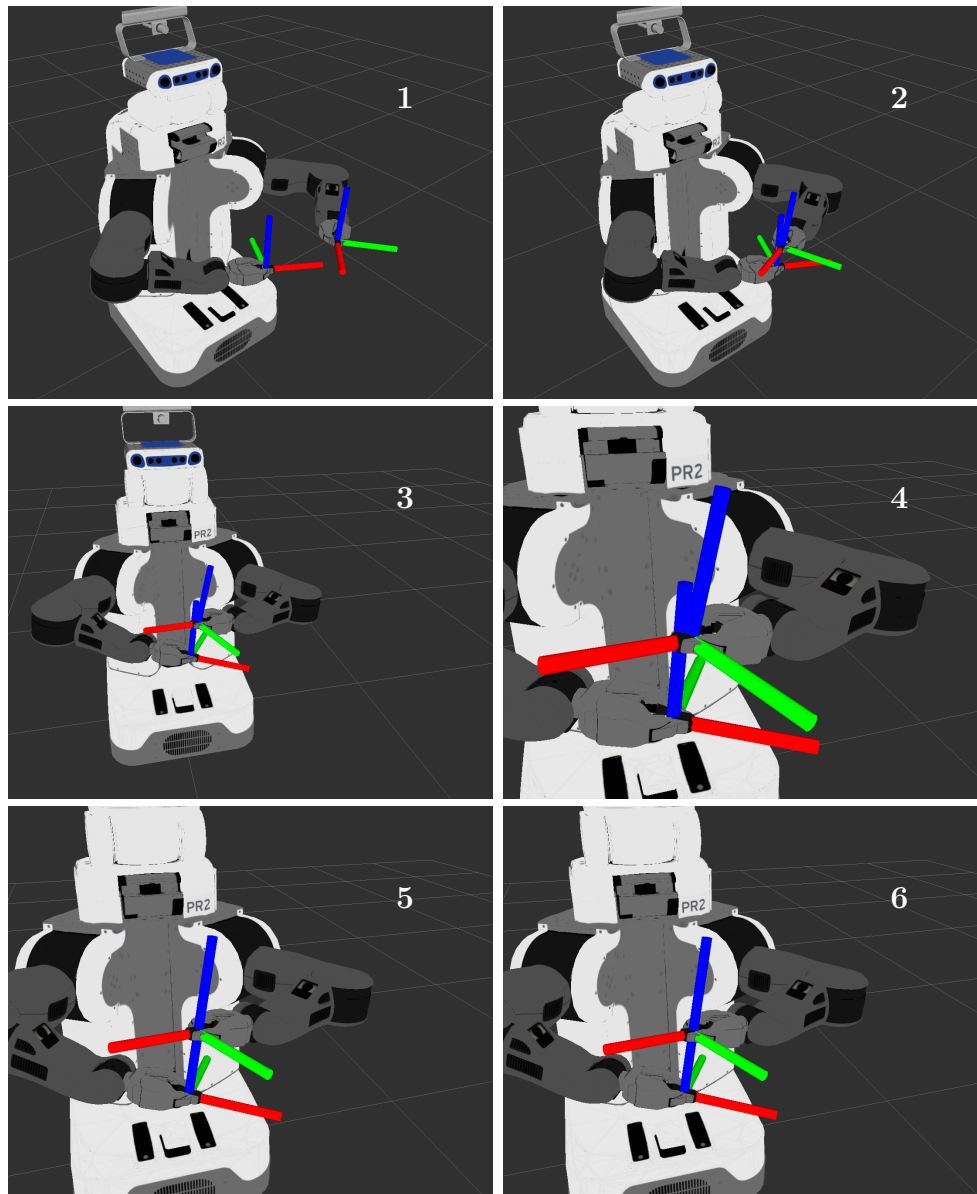


FIGURE 3.17 – Vues du robot PR2 effectuant la tâche de fermeture du stylo

PR2 sur le visualiseur de ROS, réalisant les tâches. Les repères liés au stylo et au capuchon sont aussi affichés avec l'axe X en rouge, l'axe Y en vert et l'axe Z en bleu. Le stylo et le capuchon sont orientés selon Z . Les photos sont dans l'ordre chronologique de gauche à droite, et du haut vers le bas. L'on peut voir que les deux bras bougent, et qu'à la fin de la tâche, le stylo et le capuchon sont correctement alignés (les axes Z en bleu).

Nous avons ajouté ensuite un bruit d'écart-type 1 pixel sur les mesures dans l'image, et un bruit de covariance 8×10^{-9} radians² sur les mesures des codeurs des articulations et sur la loi de commande calculée. Le seuil sur la norme des tâches pour le basculement automatique vers la suivante a été relevé à 1.5. Les gains λ_1 , λ_2 ont été abaissés à 0.5 pour e_1 et e_2 et $\lambda_3 = 0.2$ pour e_3 , tandis que μ est égal à $3\lambda_i$. Nous affichons sur les figures 3.18 et 3.19, l'évolution obtenue des fonctions de tâches et de la loi de commande pour un des bras, dans ces conditions.

3. <http://www.ros.org/>

4. <http://gazebosim.org/>

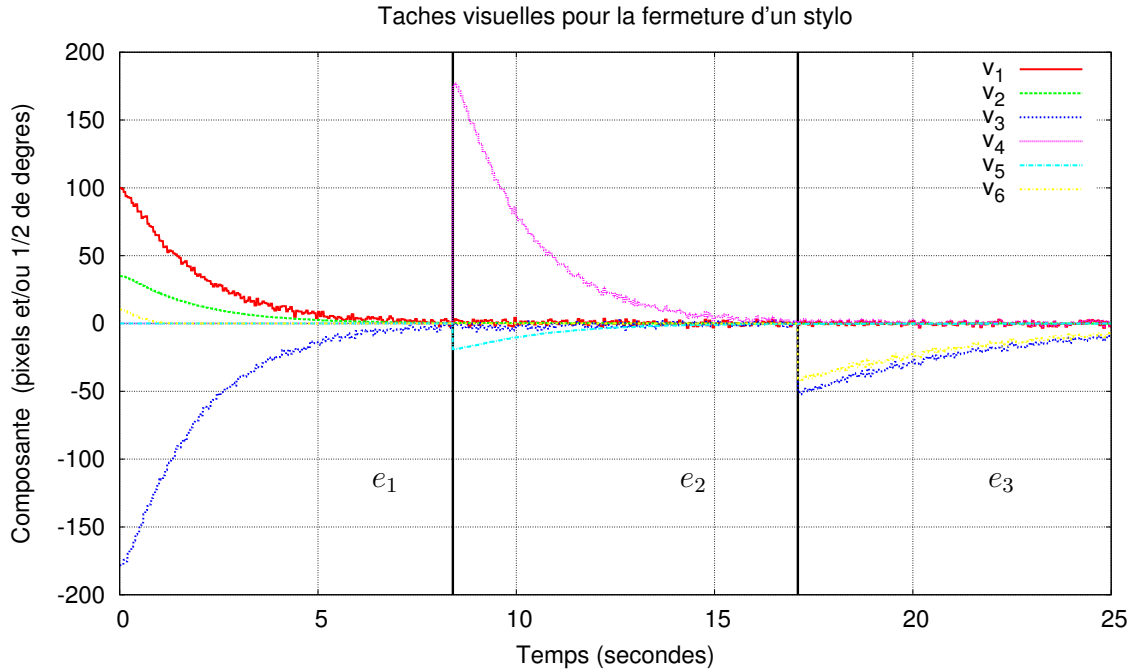


FIGURE 3.18 – Évolution de la tâche bruitée en fonction du temps.

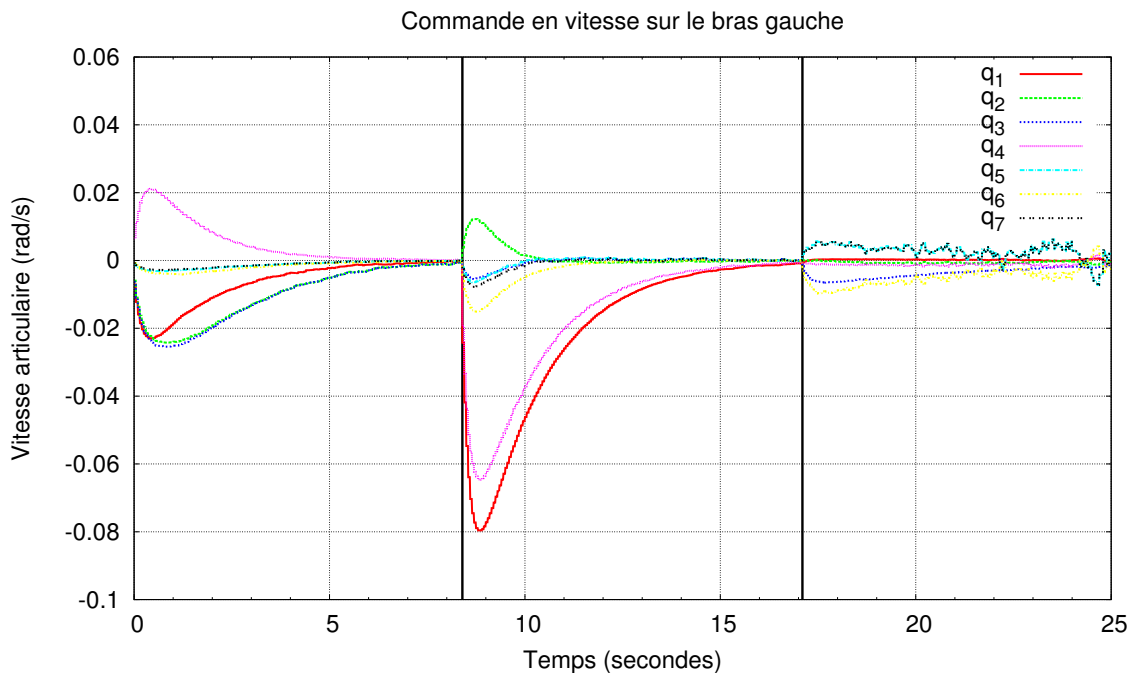


FIGURE 3.19 – Commande en vitesse des articulations du bras gauche du PR2 en présence de bruit.

Nous pouvons constater que la tâche de fermeture est encore réalisée malgré le niveau de bruit injecté. Une vidéo, réalisée dans des conditions similaires, est d'ailleurs disponible sur le web⁵. Mais nous avons constaté que certaines articulations restaient souvent blo-

5. <https://homepages.laas.fr/rfleurmo/>

quées près de leur butées. Par exemple, sur la figure 3.20, obtenue dans la simulation sans bruit, une des articulations du poignet droit du robot arrive en butée. Ce comportement

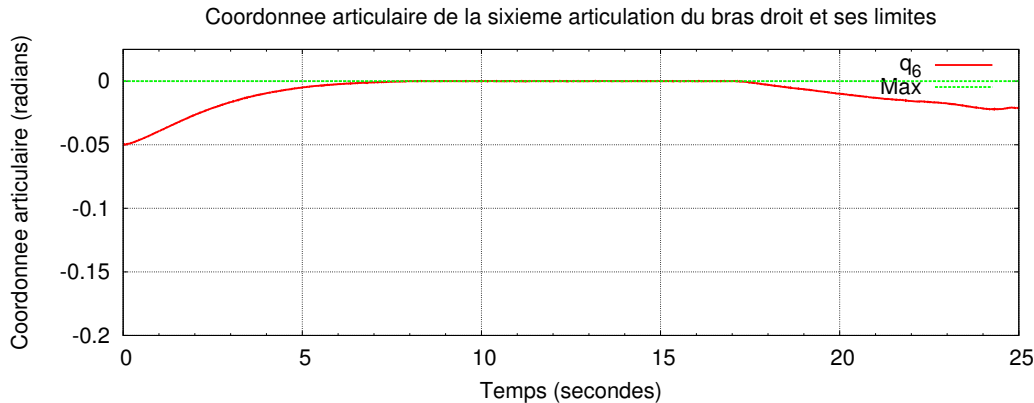


FIGURE 3.20 – Une articulation arrive en butée.

est indésirable et dangereux pour le robot. Avant donc de passer aux expérimentations, nous devons mettre en œuvre une tâche additionnelle pour éviter les butées. En exploitant la redondance, cette tâche devra être exécutée simultanément avec la séquence de tâches d’asservissement visuel définies précédemment. Cet aspect sera traité dans le chapitre suivant.

3.5 Conclusion

Nous avons proposé un cadre général permettant de réaliser la coordination entre plusieurs bras, en couplant l’asservissement visuel 2D avec le formalisme des tâches coopératives. Ce cadre permet d’utiliser une ou plusieurs caméras, embarquées sur un bras ou fixes par rapport à la scène. Il est ainsi possible d’effectuer les deux types de tâches coordonnées existantes en développant des stratégies de commande basées sur l’asservissement visuel multi-caméras. De plus nous avons montré que notre approche permet une vraie coopération des bras. En d’autres termes, les deux manipulateurs contribuent à la réalisation de la tâche, sans imposer de schéma maître/esclave. Ce résultat s’avère intéressant car il permet de n’utiliser que les degrés de liberté nécessaires pour la tâche. La redondance offerte par le système robotique peut donc être préservée, contrairement aux approches exploitant l’asservissement visuel dans le contexte de la coordination que nous avons pu trouver dans la littérature [Hynes et al., 2006, Vahrenkamp et al., 2009b].

Les outils développés ont été employés pour exécuter une tâche particulière : refermer un stylo. Nous avons décrit le modèle géométrique du stylo et du capuchon, et conséquemment, choisi les indices visuels qui permettaient de décrire l’image des objets et de construire les fonctions de tâches nécessaires à l’obtention de la pose relative désirée. Nous avons divisé la tâche à accomplir en une séquence de trois sous-tâches qui se succèdent automatiquement. Toutes ces sous-tâches ont été définies par la comparaison entre les indices visuels du stylo et du capuchon fournies par deux caméras, l’une mobile et l’autre fixe. Par conséquent, en plus de la coordination entre les deux bras, nous avons aussi réalisé un asservissement visuel sur plusieurs cibles et avec plusieurs caméras dans les configurations "eye-in-hand" et "eye-to-hand". Cette contribution a été présentée à une conférence internationale et publiée dans cet article [Fleurmond and Cadenat, 2014].

Mais les tâches définies peuvent, selon les conditions initiales, amener les articulations à frôler ou même tenter de dépasser les butées. De plus, les références utilisées pour notre asservissement étant dans certains cas mobiles, rien ne contraint les objets manipulés à rester dans le champ de vue de nos caméras. Il est donc nécessaire de pouvoir éviter les butées et maintenir les objets d'intérêt dans le champ de vision des caméras. C'est pour cela que, dans le chapitre suivant, nous nous intéresserons à la redondance pour prendre en compte l'ensemble de ces objectifs.

Chapitre 4

La gestion de la redondance

Kay piti, ou pran nat ou anba bra ou
Même si on est un peu juste, on peut
te faire de la place.

Proverbe haïtien

4.1 Introduction

La redondance est généralement comprise en robotique, comme le fait d'avoir plus d'actionneurs que de degrés de liberté pour l'organe terminal. Un bras robotisé ayant plus de six actionneurs est généralement décrit comme redondant [Jazar, 2007, Corke, 2013]. Cela sous-entend que la tâche à exécuter est définie dans l'espace cartésien, où l'organe terminal doit atteindre une pose. Mais la redondance d'un robot se définit surtout par rapport à la tâche courante à effectuer [Samson et al., 1991]. Si le nombre d'articulations contrôlables est supérieur aux degrés de liberté requis par la tâche, le robot peut être dit redondant vis-à-vis de cette tâche.

Les tâches que nous avons décrites dans le chapitre précédent avaient pour but de contrôler, soit la pose absolue d'un objet par rapport à un repère monde, soit la pose relative d'un objet par rapport à un autre. La pose peut être complètement caractérisée par six variables indépendantes. De plus, nous avons montré que le rang du jacobien des fonctions de tâches coordonnées était inférieur ou égal à 6, ce qui veut dire qu'elle contraignent au maximum 6 degrés de liberté de notre système multi-bras. Dans le cadre de la manipulation à deux bras, nous supposons que chaque bras dispose d'au moins six liaisons. Donc, par rapport aux tâches définies à la section 3.3, notre robot est redondant.

Cette redondance a été étudiée dans l'approche des fonctions de tâches proposée par Samson, Le Borgne et Espiau [Samson et al., 1991]. Ce formalisme a été développé pour fournir un cadre général à la commande des robots manipulateurs. Dans ce formalisme, une action à réaliser par le robot est définie par une fonction $e(q, t)$ de classe C^2 et de dimension n . La commande doit être élaborée pour amener cette fonction à zéro dans un horizon $[0, T]$. Cette fonction peut être définie à partir d'informations proprioceptives ou extéroceptives, mais elle doit respecter certaines conditions pour être une fonction de tâche admissible. La tâche doit être *réalisable* c'est-à-dire qu'il existe, à partir d'une condition initiale, une trajectoire *idéale* unique pour la réalisation de la tâche et qu'au voisinage de cette trajectoire, son jacobien doit être de rang plein [Samson et al., 1991, Cadenat, 1999]. Lorsque le rang du jacobien de la tâche est différent du nombre de colonnes, le nombre de variables articulaires du robot, la tâche n'est pas *admissible*. Le robot est alors redondant par rapport à la tâche. La redondance a donc été étudiée [Samson et al., 1991] pour construire des tâches admissibles à partir d'autres redondantes. Des stratégies

de commande permettant de réaliser simultanément plusieurs objectifs ont alors été synthétisées. Nous nous proposons dans ce chapitre d'analyser quelques approches extraites de la littérature avant de décrire notre proposition.

4.2 Redondance dans le formalisme des fonctions de tâche

Notre but est de profiter de la redondance pour permettre au robot d'accomplir plusieurs tâches simultanément. Pour cela, il est d'abord nécessaire de construire des fonctions de tâche pour tous les objectifs à satisfaire avant de concevoir une structure de contrôle pour l'ensemble de ces tâches. Nous supposons que les tâches à réaliser sont définies par une ensemble de m fonctions de tâches redondantes $e_1, e_2, \dots, e_i, \dots, e_m$. L'indice i ne représente pas ici un ordre temporel, car les fonctions de tâche e_i sont génériques. Ces fonctions e_i sont des fonctions vectorielles de dimension inférieure à n , le nombre de variables articulaires du robot. La loi de commande permettant de réguler e_i à zéro indépendamment des autres tâches est notée \dot{q}_i et est donnée classiquement par :

$$\dot{q}_i = J_i^+ \dot{e}_i^* \quad (4.1)$$

où J_i est le jacobien de e_i et \dot{e}_i^* la dynamique souhaitée.

4.2.1 Etat de l'art

Plusieurs stratégies de commande ont été proposées pour réguler à zéro plusieurs tâches simultanément. Dans certains cas, les tâches définies agissent séparément sur des liaisons ou des degrés de liberté différents. Il est alors possible de mettre en place des approches dites "partitionnées" comme dans [Deguchi, 1998, Malis, 1998, Cadenat, 1999, Corke and Hutchinson, 2001, Folio, 2007, Durand Petiteville, 2012]. Cependant, ces approches dépendent de chaque application et il est souvent difficile de partitionner ainsi les fonctions de tâches à réaliser.

Il est préférable d'établir une stratégie de commande qui prenne en compte les dynamiques souhaitées de toutes les tâches à réaliser. Une première solution envisageable consiste à pondérer les lois de commande garantissant la réalisation indépendante de chacune des tâches à réaliser. Il vient alors [Malis et al., 2001] :

$$\dot{q} = h_1 \dot{q}_1 + h_2 \dot{q}_2 + \dots + h_m \dot{q}_m \quad \sum_{i=1}^m h_i = 1 \quad (4.2)$$

Les termes de pondérations h_i peuvent être utilisés pour imposer une priorité, les tâches les plus importantes se voyant alors attribuer les coefficients les plus grands. Mais cette loi de commande ne garantit pas le respect des dynamiques imposées séparément à chaque fonction de tâche, ni la parfaite exécution de chacune d'entre elles.

Une meilleure solution consiste à définir une fonction de tâche globale constituée de l'ensemble des m fonctions de tâche définies précédemment. La loi de commande tenant compte de toutes les tâches a pour expression [Kermorgant, 2011, Kermorgant and Chaudette, 2011c] :

$$\dot{q} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_m \end{bmatrix}^+ \cdot \begin{bmatrix} \dot{e}_1^* \\ \dot{e}_2^* \\ \vdots \\ \dot{e}_m^* \end{bmatrix} \quad (4.3)$$

Rappelons que les e_i sont des vecteurs et non des scalaires. Ce choix permet d'obtenir les vitesses permettant de respecter *au mieux* les contraintes et les dynamiques imposées par toutes les tâches. Mais ce système d'équations peut ne pas avoir de solution ou ce jacobien étendu peut être une matrice mal conditionnée. La commande peut alors être aberrante.

Pour répondre à ce problème, des approches hiérarchiques permettant d'ordonner les tâches selon un ordre de priorité ont été proposées. Ces approches utilisent la méthode du gradient projeté [Nakamura et al., 1987, Siciliano and Slotine, 1991, Samson et al., 1991]. Elles calculent l'espace des mouvements disponibles pour les tâches secondaires de sorte qu'ils soient compatibles ou n'aient aucun effet sur la dynamique de la tâche principale. Les vitesses calculées pour exécuter les objectifs non prioritaires sont donc projetées dans cet espace avant d'être ajoutées à la loi de commande de la tâche principale. La compatibilité des tâches est donc garantie par construction, ce qui présente un avantage non négligeable lorsque le nombre de degrés de liberté ou de contraintes à satisfaire est important. Le formalisme des tâches redondantes [Samson et al., 1991] s'inscrit dans ce cadre et cherche à exécuter les tâches secondaires sans modifier la dynamique de la tâche principale. Dans les cas d'une tâche principale e_1 et d'une tâche secondaire e_2 , la loi de commande permettant de réaliser cet objectif est donnée par [Siciliano and Slotine, 1991, Samson et al., 1991] :

$$\dot{q} = \dot{q}_1 + P_1 \dot{q}_2 \quad (4.4)$$

où P_1 est un projecteur dans le noyau de J_1 . L'expression la plus usitée de ce projecteur est [Siciliano and Slotine, 1991] :

$$P_i = I - J_i^+ J_i \quad (4.5)$$

qui est un cas particulier de l'opérateur de projection préconisé dans [Samson et al., 1991]. Cette loi de commande garantit donc la convergence de la tâche e_1 , tandis que la tâche e_2 est réalisée au mieux en utilisant les degrés de liberté restés disponibles [Cadenat et al., 2001].

Dans le but d'améliorer l'exécution de la tâche secondaire e_2 , il est possible de prendre en compte l'impact de la loi de commande pour la tâche prioritaire sur la dynamique de la tâche secondaire. La loi de commande suivante permet d'atteindre cet objectif [Siciliano and Slotine, 1991] :

$$\dot{q} = \dot{q}_1 + (J_2 P_1)^+ (\dot{e}_2^* - J_2 \dot{q}_1) \quad (4.6)$$

Cette loi ne change pas la dynamique de la tâche prioritaire et calcule le mouvement qui respecte le plus la dynamique souhaitée de la tâche secondaire. Les travaux de Chiaverini [Chiaverini, 1997] ont mis en évidence les singularités algorithmiques de cette loi de commande, qui paradoxalement, donne le même résultat que celle du jacobien étendu (4.3). En effet, tout comme ce dernier, elle cherche à respecter les deux dynamiques. Elle souffre donc des mêmes limitations et présente les mêmes avantages.

Une autre solution consiste à utiliser le concept de redondance directionnelle introduit par Mansard et Chaumette [Mansard and Chaumette, 2009]. La commande calculée pour la tâche prioritaire seule ayant pour but de la faire converger vers zéro, ils ont eu l'idée d'autoriser toute partie de la commande de la tâche secondaire qui prise séparément fait décroître la fonction d'erreur principale ou n'impacte pas sa dynamique. Ils ont donc proposé un projecteur particulier plus permissif que celui donné par l'équation (4.5). Ce projecteur $\tilde{P}(J_1, \dot{q}_1, \dot{q}_2)$ dépend donc également de la loi de commande de la tâche secondaire. La loi de commande ainsi obtenue est donnée par :

$$\dot{q} = \dot{q}_1 + \tilde{P}(J_1, \dot{q}_1, \dot{q}_2) \cdot \dot{q}_2 \quad (4.7)$$

L'expression quelque peu complexe du projecteur peut être retrouvée dans [Mansard and Chaumette, 2009]. Le revers de la médaille est que le système réagit comme si le correcteur

avait un fort gain, ce qui peut dégrader la stabilité du système. Pour pallier ce problème, il a été nécessaire, à l'intérieur du projecteur, de limiter l'apport de la seconde tâche en fonction de la commande synthétisée pour la première.

Enfin, nous pouvons également mentionner les travaux de Marey [Marey and Chaudette, 2010]. Dans ses travaux, il définit un projecteur adapté de (4.5) où la tâche prioritaire est réduite à une tâche de dimension 1. De cette manière, un plus grand nombre de degrés de liberté est disponible pour la tâche secondaire, facilitant sa réalisation. Mais lorsque la norme de la tâche prioritaire devient faible (i.e., au voisinage du but), il doit basculer vers le projecteur classique (4.5).

4.2.2 Discussion générale

Les approches hiérarchiques sont généralisables à n tâches par récurrence [Mansard, 2006, Siciliano and Slotine, 1991, Baerlocher and Boulic, 2004]. Elles semblent bien s'adapter à un nombre quelconque de tâches sur un robot ayant de nombreuses liaisons. Donc, les stratégies susceptibles de nous intéresser dans notre problème de redondance et de coordination sont données par les équations (4.4), (4.6) et (4.7). Considérons maintenant deux tâches : une première tâche d'évitement de butées prioritaire e_1 et une deuxième tâche référencée vision coordonnée e_2 . Dans le cas où une articulation est près de sa butée, la loi de commande de la tâche e_1 doit amener cette articulation à s'éloigner de cette butée.

En utilisant la redondance classique (4.4) ou la redondance directionnelle (4.7), la vitesse de l'articulation en butée est imposée par la tâche prioritaire. Or, le calcul de la loi de commande de la tâche référence vision ne tient pas compte de cette contrainte et Les vitesses articulaires de l'autre bras ne peuvent donc s'adapter. La vitesse relative d'un organe terminal par rapport à l'autre est ainsi impactée, ce qui peut induire des efforts indésirables sur l'objet manipulé ou sa chute. Si les priorités des tâches sont inversées, la tâche de coordination s'exécute, mais l'articulation peut arriver en butée, jusqu'à mettre en péril l'exécution de la tâche prioritaire.

Avec la redondance optimale (4.6), la tâche e_2 est informée de la contrainte apportée sur les articulations d'un bras par e_1 , la loi de commande s'adapte en répercutant le changement sur l'autre bras. La vitesse relative entre les organes terminaux n'est pas affectée. La communication entre les tâches redondantes est donc intéressante pour la coordination entre les bras manipulateurs. Si les priorités des tâches sont inversées, la tâche d'évitement des butées peut être en contradiction avec la direction imposée par la tâche de coordination. Il vaut donc mieux que la tâche d'évitement des butées, qui n'intervient significativement que localement sur une ou quelques articulations, soit prioritaire. Alors, la tâche de coordination, qui a une vue plus globale du système, peut alors compenser avec des articulations de l'autre bras et être réalisée sans perturbations.

La redondance optimale semble donc être la meilleure candidate pour la coordination, mais elle est connue pour présenter des *singularités algorithmiques*. En effet, lorsque $Im(J_1^+) \simeq Im(J_2^+)$, la loi de commande produit des vitesses articulaires aberrantes dont la norme peut tendre vers l'infini [Mansard, 2006, Chiaverini, 1997, Lee et al., 2012]. Chiaverini [Chiaverini, 1997], en éliminant ces singularités, retrouve la redondance classique de l'équation (4.4). Nous allons à présent étudier ces singularités et tenter d'améliorer la redondance optimale.

4.2.3 Analyse des approches retenues et proposition d'une nouvelle stratégie

Illustrons le comportement de la redondance classique (4.4), de la redondance optimale (4.6) et de la redondance directionnelle (4.7) sur le cas simple d'un robot plan constitué de deux liaisons prismatiques perpendiculaires afin de faciliter l'analyse. Soient deux tâches e_1 et e_2 de rang 1.

Considérons d'abord la tâche e_1 de jacobien J_1 . La loi de commande permettant de la réaliser seule est donnée par $\dot{q}_1 = J_1^+ \dot{e}_1^*$. La matrice J_1 exprime une application linéaire de l'espace des vitesses articulaires vers les vitesses de e_1 . Les vitesses articulaires de l'autre bras ne peuvent donc s'adapter. Le noyau $Ker(J_1)$ de J_1 est un sous-espace vectoriel de l'espace des vitesses articulaires de dimension 1, perpendiculaire à $Im(J_1^+)$. La figure 4.1 illustre ces deux sous-espaces dans l'espace des vitesses articulaires du robot. La tâche e_1 étant redondante, le vecteur \dot{q} est la solution de norme minimale au système linéaire $J_1 \dot{q} = \dot{e}_1^*$. Naturellement une infinité de solutions à ce système existent, les trois vecteurs en traits fins en sont trois exemples.

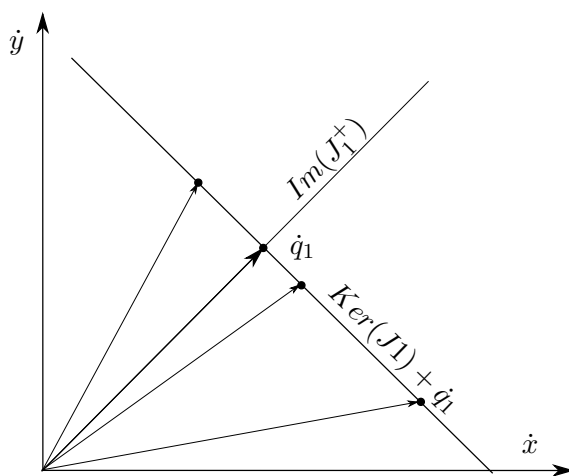


FIGURE 4.1 – Espace des vitesses articulaires

Dans les figures suivantes, la loi de commande \dot{q}_2 permettant de réguler à zéro la fonction de tâche e_2 , les espaces $Im(J_2^+)$ et $Ker(J_2)$ seront illustrés en rouge, tandis que tout ce qui concerne la tâche prioritaire e_1 restera en noir. Nous présenterons trois situations différentes possibles lorsque la redondance est utilisée pour réaliser ces deux tâches e_1 et e_2 .

La figure 4.2 présente une situation où les lois de commande des tâches sont compatibles. Les vecteurs noir et rouge sont respectivement les vitesses articulaires \dot{q}_1 et \dot{q}_2 qui permettent de réaliser indépendamment ces deux tâches. La loi de commande donnée par la redondance classique (4.4) est illustrée par le vecteur vert. Ce vecteur est la somme de \dot{q}_1 en noir, et de la projection perpendiculaire (ligne en pointillés) sur $Ker(J_1)$ de \dot{q}_2 en rouge. La loi de commande donnée par la redondance optimale (4.6) est illustrée par le vecteur bleu. Ce vecteur est à la rencontre des deux noyaux $Ker(J_1)$ et $Ker(J_2)$. La loi de commande donnée par la redondance directionnelle (4.7) est illustrée par le vecteur magenta, qui est la somme des vecteurs \dot{q}_1 et \dot{q}_2 . Dans cette situation, la redondance optimale 4.6 donne les meilleurs résultats, puisqu'elle respecte des deux dynamiques. Elle donne également le vecteur des vitesses articulaires de norme minimale. Dans les prochaines figures, nous conservons la même convention de couleur.

La figure 4.3 représente le cas où les lois de commande \dot{q}_1 et \dot{q}_2 poussent le robot dans deux directions contraires. La redondance classique et la redondance directionnelle

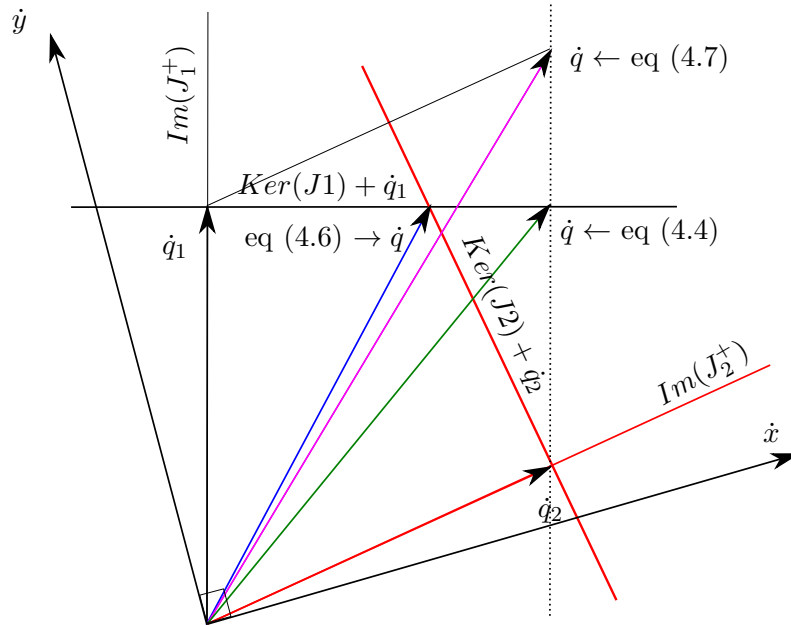


FIGURE 4.2 – Espace des vitesses articulaires. Illustration des noyaux des jacobiens des tâches e_1 et e_2 . Les lois de commande de e_1 et e_2 sont compatibles. Illustration des vitesses articulaires données dans ce cas, par la redondance classique (4.4), la redondance directionnelle (4.7) et la redondance optimale (4.6).

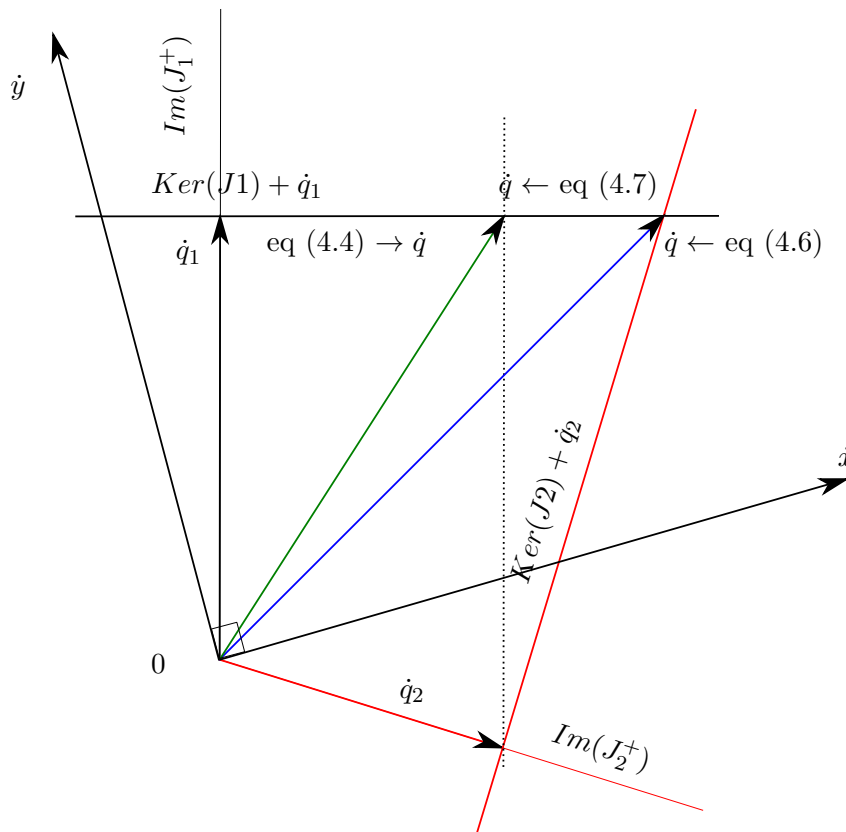


FIGURE 4.3 – Espace des vitesses articulaires. Illustration des noyaux des jacobiens des tâches e_1 et e_2 . Les lois de commande de e_1 et e_2 vont dans des directions opposées. Illustration des vitesses articulaires données dans ce cas, par la redondance classique (4.4), la redondance directionnelle (4.7) et la redondance optimale (4.6).

donnent ici les mêmes vitesses articulaires. Par sécurité, \dot{q}_2 est projeté dans l'espace nul de J_1 . Ce vecteur est de norme minimale. La redondance optimale (4.6) tend encore à

respecter les dynamiques des deux tâches, ce qui fait que la commande peut facilement tendre vers l'infini et mener à l'instabilité. En effet, si l'on tourne le vecteur et les droites rouges \dot{q}_2 , $Ker(J_2)$ et $Im(J_2^+)$ autour de 0 dans le sens des aiguilles d'une montre, l'on se rend compte que l'intersection de la droite rouge et de la droite horizontale noire tend vers l'infini.

Nous venons de mettre en évidence que la redondance optimale a un problème avec des commandes de direction contraires, lorsque $\dot{q}_1^T \cdot \dot{q}_2 < 0$. Cependant, une singularité algorithmique peut aussi se produire lorsque les deux lois de commande ont la même direction. La figure 4.4 illustre ce problème. Maintenant les deux lois de commandes \dot{q}_1 et \dot{q}_2 ont une direction, un sens et une grandeur similaires. La redondance classique ne fait que projeter sur l'espace nul $Ker(J_1)$ tandis que la redondance directionnelle laisse passer une plus grande partie de \dot{q}_2 qui va dans la même direction que \dot{q}_1 . Dans cette situation, la redondance classique donne le vecteur des vitesses articulaires de norme minimale qui respecte la dynamique de la tâche prioritaire et satisfait au mieux la tâche secondaire. La redondance optimale (4.6) donne une vitesse qui tend vers l'infini parce qu'elle essaie encore de satisfaire les deux dynamiques. Le problème ne dépend pas seulement des jacobiens J_1 et J_2 et de leurs noyaux. Si par exemple la valeur de \dot{q}_2 diminue, mais ne change pas de direction, ce qui se traduit visuellement sur la figure 4.4 par une homothétie de rapport inférieur à 1 et de centre 0 des droites et vecteur rouges, les tâches peuvent devenir compatibles comme sur la figure 4.2. Il faut donc tenir compte des valeurs des vitesses articulaires pour détecter la singularité. Dans ce cas, le vecteur des vitesses articulaires choisi est celui de norme minimale.

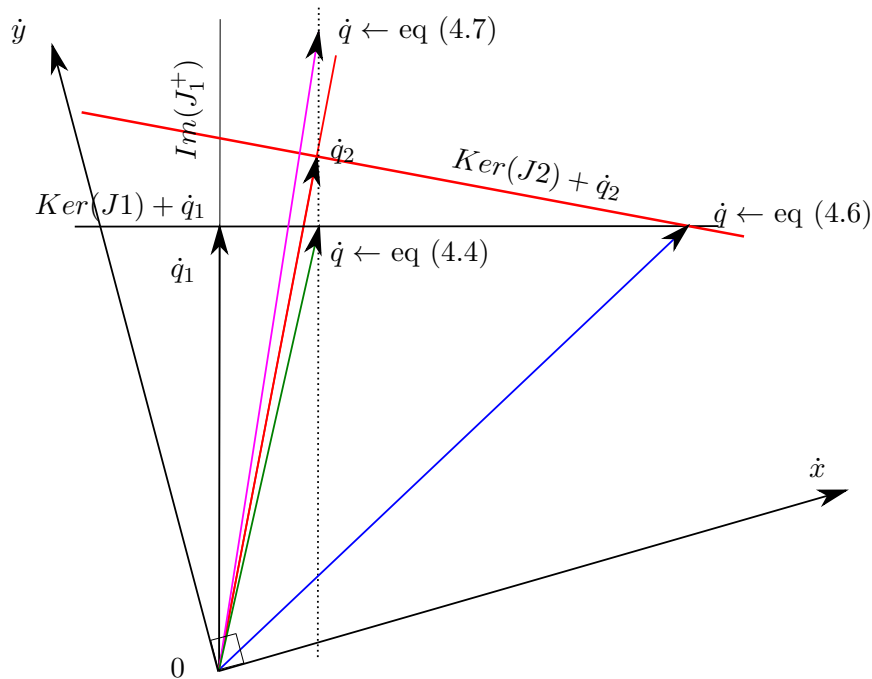


FIGURE 4.4 – Espace des vitesses articulaires. Illustration des noyaux des jacobiens des tâches e_1 et e_2 . Les lois de commande de e_1 et e_2 sont similaires. Illustration des vitesses articulaires données dans ce cas, par la redondance classique (4.4), la redondance directionnelle (4.7) et la redondance optimale (4.6). Illustration d'une singularité algorithmique de la redondance optimale.

La redondance optimale est la stratégie qui semble mieux convenir pour la coordination, comme pour le cas général des fonctions de tâche, mais lorsqu'elle rencontre ces singularités (cf. figure 4.3 et 4.4), il vaut mieux revenir à la redondance classique (4.4) qui donne une solution plus sûre quelle que soit la situation. Dans chacune des trois situa-

tions illustrées par les figures 4.2, 4.3 et 4.4, la redondance optimale donnait le vecteur des vitesses articulaires de norme minimale quand \dot{q}_2 et \dot{q}_1 étaient compatibles. En cas de singularité, le vecteur calculé par la redondance classique était celui de norme minimale. Notre stratégie de commande est donc donnée par :

$$\begin{aligned} \dot{q}_p &= \dot{q}_1 + P_1 \dot{q}_2 & \dot{q}_o &= \dot{q}_1 + (J_2 P_1)^+ J_2 (\dot{q}_2 - \dot{q}_1) \\ \dot{q} &= \begin{cases} \dot{q}_o & \text{si } \|\dot{q}_o\| < \|\dot{q}_p\| \\ \dot{q}_p & \text{sinon} \end{cases} \end{aligned} \quad (4.8)$$

Elle peut aussi être présentée sous forme de l'algorithme suivant :

Algorithme 1 : Loi de commande pour deux tâches en simultanée.

Data : \dot{e}_1^* , J_1 , \dot{e}_2^* et J_2

Result : \dot{q}

$\dot{q}_1 \leftarrow J_1^+ \dot{e}_1^*$;

$\dot{q}_2 \leftarrow J_2^+ \dot{e}_2^*$;

$P_1 \leftarrow I - J_1^+ \times J_1$;

$\dot{q} \leftarrow \dot{q}_1 + P_1 \times \dot{q}_2$;

if $\dot{q}_1^T \cdot \dot{q}_2 > 0$ **then**

$\dot{q}_x \leftarrow \dot{q}_1 + (J_2 P_1)^+ \times (\dot{e}_2^* - J_2 \times \dot{q}_1)$;

if $\|\dot{q}_x\| < \|\dot{q}\|$ **then**

$\dot{q} \leftarrow \dot{q}_x$;

end

end

En utilisant cet algorithme, la transition entre les deux redondances se fait de manière continue, lorsque les normes des commandes calculées correspondantes sont égales, c'est-à-dire lorsque \dot{q}_1 et \dot{q}_2 sont perpendiculaires $\dot{q}_1^T \cdot \dot{q}_2 = 0$ (Figure 4.5a) ou forment un triangle rectangle $J_1 \dot{q}_2 = \dot{e}_1^*$ (Figure 4.5b).

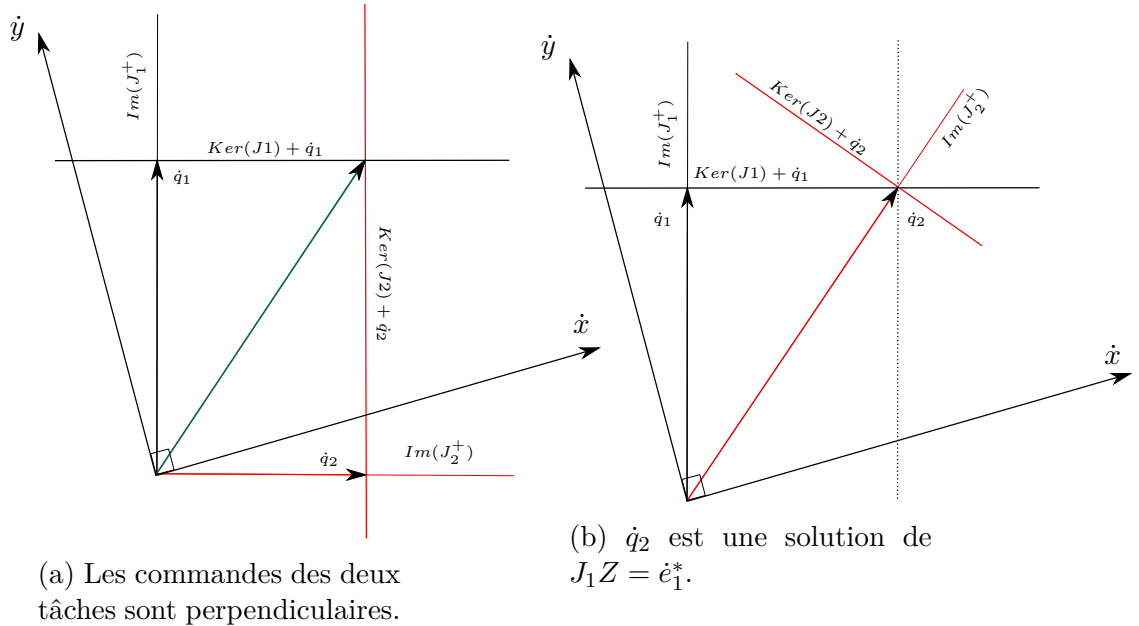


FIGURE 4.5 – Transitions entre la redondance classique et la redondance optimale.

Mais il y a un cas illustré par la figure 4.6 où la transition entre les redondances classique et optimale est discontinue. Sur cette figure, la redondance optimale (4.6) donne

le vecteur bleu de norme légèrement inférieure au vecteur vert de l'autre côté de \dot{q}_1 . Dans cette situation, nous avons $(\dot{q} - \dot{q}_1)^T(\dot{q} - \dot{q}_1) < 0$. Si nous faisons tourner légèrement autour de 0 dans le sens contraire des aiguilles d'une montre, les droites et le vecteur rouges, la norme du vecteur vert deviendra égale à celle du vecteur bleu. À ce moment, les deux vecteurs des vitesses articulaires sont de directions différentes. La transition risque donc d'être discontinue. Pour éviter cette discontinuité, nous proposons de prendre

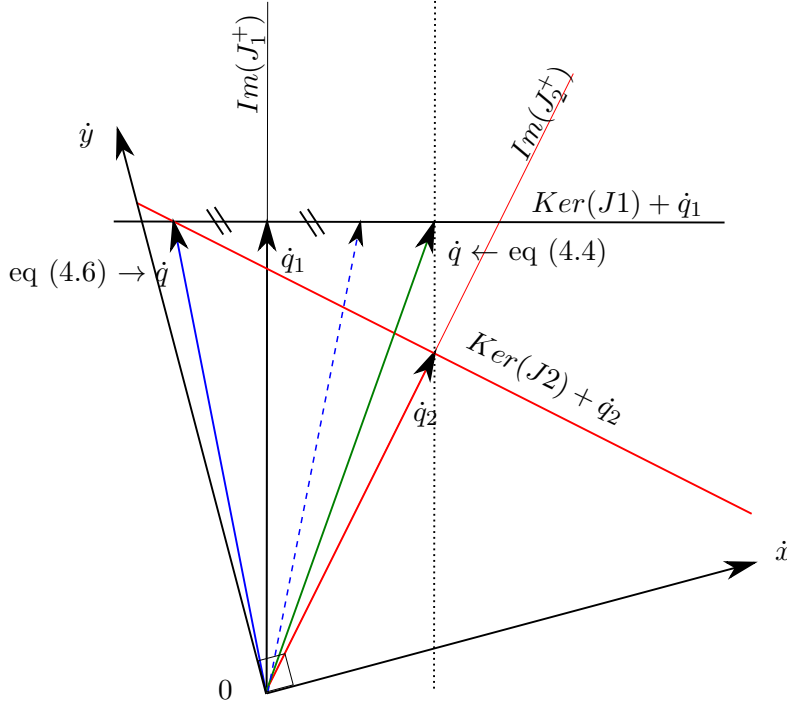


FIGURE 4.6 – Espace des vitesses articulaires. Illustration des noyaux des jacobiens des tâches e_1 et e_2 . Les lois de commande de e_1 et e_2 sont compatibles. Illustration des vitesses articulaires données dans ce cas, par la redondance classique (4.4) et la redondance optimale (4.6). Les vitesses articulaires données par ces deux approches sont des deux côtés de \dot{q}_1 . Risque de discontinuité dans notre approche.

le symétrique (bleu pointillé sur la figure 4.6) du vecteur bleu plein (4.6) par rapport à \dot{q}_1 dans ce cas précis. Notre gestion de la redondance devient alors :

Algorithme 2 : Gestion de la redondance pour deux tâches.

Data : e_1^* , J_1 , e_2^* et J_2
Result : \dot{q}
 $\dot{q}_1 \leftarrow J_1^+ e_1^*$;
 $\dot{q}_2 \leftarrow J_2^+ e_2^*$;
 $P_1 \leftarrow I - J_1^+ \times J_1$;
 $\Delta \dot{q}_p \leftarrow P_1 \times \dot{q}_2$;
 $\dot{q} \leftarrow \dot{q}_1 + \Delta \dot{q}_p$;
if $\dot{q}_1^T \dot{q}_2 > 0$ **then**
 $\Delta \dot{q}_o \leftarrow (J_2 P_1)^+ \times (e_2^* - J_2 \times \dot{q}_1)$;
 $\dot{q}_o \leftarrow \dot{q}_1 + \Delta \dot{q}_o$;
 if $\|\dot{q}_o\| < \|\dot{q}\|$ **then**
 $\dot{q} \leftarrow \dot{q}_o$;
 if $\Delta \dot{q}_o^T \Delta \dot{q}_p < 0$ **then**
 $\dot{q} \leftarrow \dot{q}_1 - \Delta \dot{q}_o$;
 end
 end
end
end

La stratégie de commande peut être étendue à m tâches ordonnées en utilisant la récurrence [Siciliano and Slotine, 1991, Baerlocher and Boulic, 2004, Mansard, 2006]. En choisissant initialement $P_1 = I$ et $\dot{q} = 0$, il vient :

$$\begin{aligned}
\dot{q}_i &= J_i^+ \dot{e}_i^* \\
{}^p\dot{q}_i &= \dot{q} + P_i \cdot \dot{q}_i \\
{}^o\dot{q}_i &= \dot{q} + (J_i P_i)^+ \cdot J_i \cdot (\dot{q}_i - \dot{q}) \\
\dot{q} &= \begin{cases} {}^o\dot{q}_i & \text{si } \|{}^o\dot{q}_i\| < \|{}^p\dot{q}_i\| \text{ et } ({}^o\dot{q}_i - \dot{q})^T ({}^p\dot{q}_i - \dot{q}) > 0 \\ 2\dot{q} - {}^o\dot{q}_i & \text{si } \|{}^o\dot{q}_i\| < \|{}^p\dot{q}_i\| \text{ et } ({}^o\dot{q}_i - \dot{q})^T ({}^p\dot{q}_i - \dot{q}) < 0 \\ {}^p\dot{q}_i & \text{sinon} \end{cases} \quad (4.9) \\
P_{i+1} &= P_i - (J_i P_i)^+ (J_i P_i)
\end{aligned}$$

Nous rappelons que les tâches sont listées par ordre décroissant de priorité. Grâce à notre approche, la dernière tâche est en quelque sorte informée de la direction prise par les tâches précédentes. Si elle peut s'en accommoder, elle adapte la loi de commande, sinon seule la partie qui n'influence pas les tâches précédentes est retenue.

4.2.4 Évaluation de la stratégie proposée au travers d'un exemple

Nous comparons en simulation les quatre approches évoquées précédemment pour gérer la redondance. Nous considérons le cas d'un robot mobile évoluant dans un plan. Nous considérons ici que le vecteur q est défini par le couple (x, y) . Le robot est initialement à l'origine $(0, 0)$ du repère. Le robot doit réaliser deux tâches e_1 et e_2 , la première étant prioritaire sur la seconde. Ces tâches l'attirent vers les points respectifs $M_1(6, 2)$ et $M_2(6, -2)$ qui sont à égale distance du robot. Il s'agit donc de deux tâches incompatibles. Les tâches e_i sont définies comme le carré de la distance entre le robot et $M_i(x_i, y_i)$:

$$\begin{aligned}
e_i &= (x - x_i)^2 + (y - y_i)^2 & \dot{e}_i^* &= -\lambda e_i \\
J_i &= \begin{bmatrix} 2(x - x_i) & 2(y - y_i) \end{bmatrix} & \dot{q}_i &= J_i^+ \dot{e}_i^*
\end{aligned}$$

Le gain λ a été fixé à 1 pour les deux tâches. Quatre approches ont été simulées :

- la pondération des commandes des deux tâches (4.2) avec $h_1 = 0.75$ pour e_1 et $h_2 = 0.25$ pour e_2 .
- la redondance classique (4.4).
- la redondance optimale (4.6).
- la stratégie que nous avons proposée dans l'algorithme 2.

La figure 4.7 présente les trajectoires obtenues pour ces différentes approches. Soumis seul à la tâche e_1 , le robot part de $(0, 0)$ et parcourt la trajectoire rectiligne matérialisée par la ligne mauve #5 et atteint la position cible $M_1(6, 2)$. Mais soumis à la tâche e_2 , le robot parcourt la trajectoire rectiligne matérialisée par la ligne orange #6 et atteint la position cible $M_2(6, -2)$. En essayant de satisfaire les deux tâches e_1 et e_2 avec la pondération choisie, le robot parcourt la trajectoire rectiligne magenta #1 pour atteindre une position intermédiaire $0.75(6, 2) + 0.25(6, -2) = (6, 1)$. Aucune des deux tâches n'est donc réalisée. Seul un compromis est réalisé. En utilisant la redondance classique, le robot parcourt la trajectoire courbe verte #2 et atteint la cible prioritaire $M_1(6, 2)$. Notre approche donne la trajectoire curviligne rouge #4. À l'extérieur du cercle, c'est la redondance optimale qui est appliquée. Mais à l'intérieur du cercle, les commandes respectives des deux tâches sont détectées difficilement compatibles, et c'est la redondance classique qui est exploitée pour sa sûreté. Le robot atteint la cible prioritaire $M_1(6, 2)$.

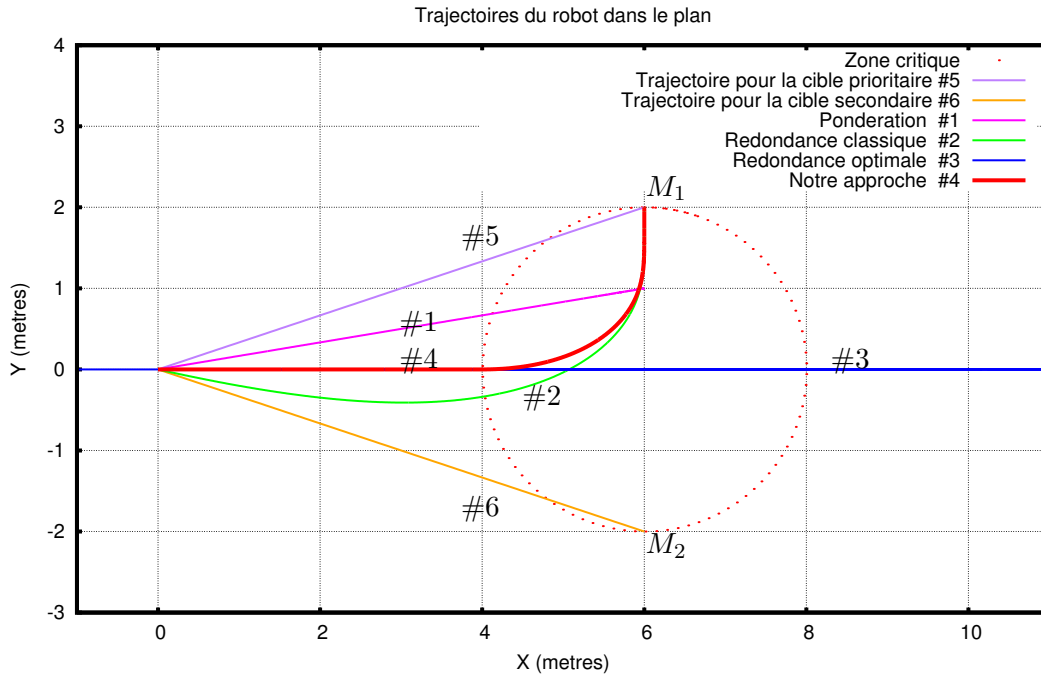


FIGURE 4.7 – Trajectoires du robot plan avec plusieurs stratégies de redondance.

Lorsque la redondance optimale est utilisée, le robot est attiré également par les deux cibles et suit donc la médiatrice de $[M_1M_2]$, ce qui donne la trajectoire bleue rectiligne #3. Mais dès que le robot pénètre dans le cercle rouge de diamètre $[M_1M_2]$, les lois de commande respectives des deux tâches deviennent incompatibles comme sur la figure 4.3, parce que $\dot{q}_1^T \dot{q}_2 < 0$, le système devient instable, et le robot oscille très fortement sur cette médiatrice, comme le montre la figure 4.8. Le robot n’atteint donc aucun des deux objectifs. Plusieurs positions initiales ont été testées qui ont toutes menées aux mêmes résultats qualitatifs.

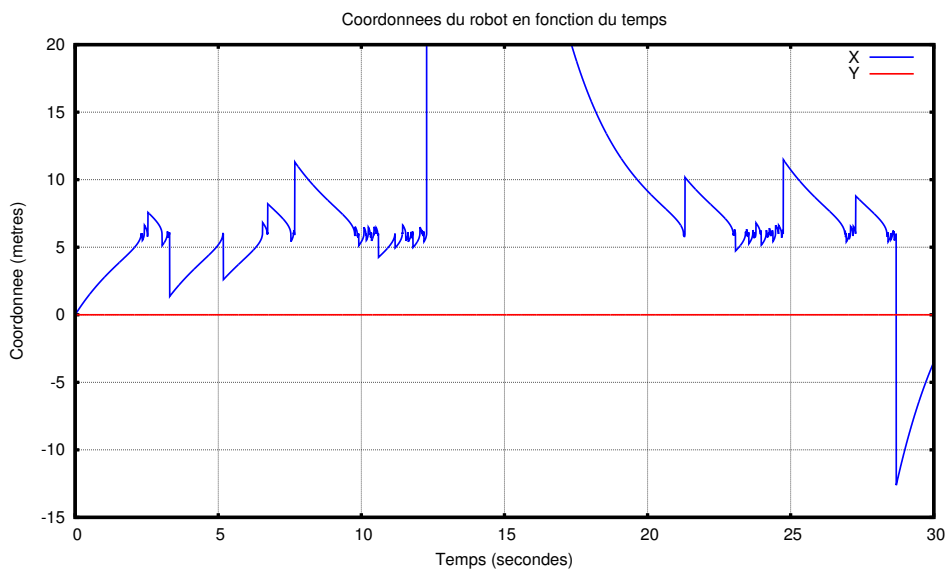


FIGURE 4.8 – Évolution des coordonnées du robot avec la redondance optimale. Le robot oscille et s’éloigne des deux cibles.

La figure 4.9 représente les vecteurs vitesses résultants obtenus pour les différentes approches. Les vitesses obtenues pour la pondération ont la même direction, ce qui correspond à la trajectoire rectiligne obtenue. Les commandes calculées avec la redondance classique tracent la courbe verte, tandis que le vecteur vitesse obtenu avec la redondance optimale parcourt la droite bleue et sort du cadre. Les vitesses obtenues avec notre approche évoluent de manière continue, même si le comportement change, ceci grâce au choix judicieux des transitions. Toutes les vitesses convergent vers zéro sauf lorsque la redondance optimale est utilisée, le robot devenant instable lors de cette simulation.

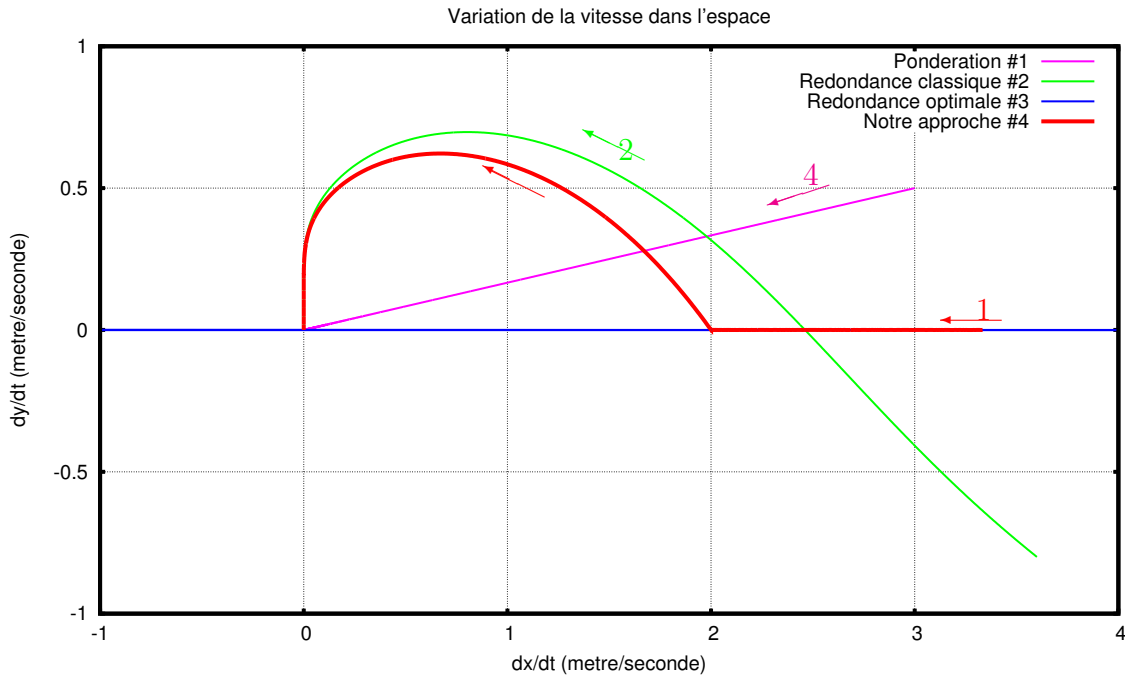


FIGURE 4.9 – Vitesses du robot plan

4.2.5 Conclusion

Notre principale contribution dans cette section est la sécurisation de la redondance optimale. Nous avons mis en lumière les situations où elle provoque un comportement instable du système. Dans ces situations nous avons choisi de la remplacer par la redondance classique, pour bénéficier de la sécurité qu'elle offre. De cette manière nous pouvons utiliser presque tout le temps la redondance optimale qui est particulièrement utile pour les tâches de manipulation coordonnée. Nous allons maintenant utiliser cette approche pour gérer les problèmes de butées et d'occultations qui peuvent se produire durant l'exécution de la tâche de coordination.

4.3 Prise en compte des butées et des contraintes de visibilité

Nous devons prendre en compte l'évitement des butées et des occultations. Nous allons donc définir deux tâches, l'une pour l'évitement des butées et l'autre pour le maintien des objets dans le champ de vue des caméras. Comme dans l'exemple précédent, nous avons choisi d'imposer une hiérarchie entre les tâches :

- La tâche prioritaire est la tâche d'évitement des butées parce qu'elle permet de protéger les articulations du robot.
- La deuxième tâche est la tâche d'asservissement visuel coordonné pour refermer le stylo (cf . section 3.4). Notons qu'elle est en réalité composée d'une séquence de trois sous-tâches.
- Enfin la dernière tâche contraint les objets à rester le plus près possible du centre de l'image.

La figure 4.10 montre la manière dont nous organisons l'ensemble des tâches. Les commandes et les jacobiens de la tâche d'évitement des butées, de la sous-tâche active d'asservissement visuel e_i et de la tâche d'évitement des occultations sont traitées selon notre stratégie de redondance (4.9) pour calculer la commande finale à envoyer au robot.

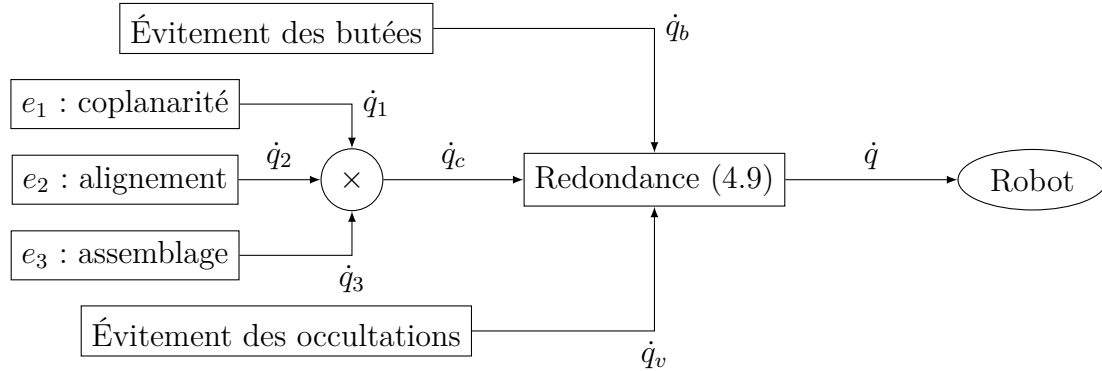


FIGURE 4.10 – Structure des tâches avec la redondance.

Nous allons, dans les lignes qui suivent, définir les tâches d'évitement de butées et d'occultations, donner l'expression de leur jacobien et de leurs lois de commande individuelles \dot{q}_b et \dot{q}_v .

4.3.1 Évitement des butées

La tâche d'évitement des butées doit être active seulement quand une articulation arrive près de ses limites. Si nous notons $[q_{min}^i, q_{max}^i]$ l'intervalle des limites physiques d'une articulation i , un intervalle de sécurité $[q_{s-}^i, q_{s+}^i]$ peut être défini par :

$$\begin{cases} q_{s-}^i &= q_{min}^i + (1 - \tau_b) \cdot (q_{max}^i - q_{min}^i) \\ q_{s+}^i &= q_{max}^i - (1 - \tau_b) \cdot (q_{max}^i - q_{min}^i) \end{cases}$$

avec $\tau_b \in]0, 0.5[$ un paramètre ajustable. Pour contenir l'articulation q^i dans l'intervalle de sécurité, il est courant de définir une fonction de tâche F_{bi} qui agit comme une force répulsive [Mansard et al., 2009] sur l'articulation. Inspiré par [Kermorgant and Chaumette, 2011a, Mansard and Chaumette, 2005], nous avons choisi de construire une fonction de la coordonnée articulaire q^i évoluant comme le montre la figure 4.11.



FIGURE 4.11 – Fonction répulsive en dehors de l'intervalle de sécurité. Elle tend vers $+\infty$ quand q^i tend vers q_{min}^i ou q_{max}^i .

Cette fonction croît rapidement et tend vers l'infini quand la coordonnée articulaire s'approche de l'une des limites physiques, et s'annule quand elle entre dans l'intervalle de sécurité. La fonction choisie a pour expression :

$$F_{bi} = \begin{cases} (q^i - q_{s+}^i)^2 / (q_{max}^i - q^i) & \text{si } q^i \geq q_{s+}^i \\ (q_{s-}^i - q^i)^2 / (q^i - q_{min}^i) & \text{si } q^i \leq q_{s-}^i \\ 0 & \text{sinon} \end{cases} \quad (4.10)$$

Ensuite nous définissons la fonction de tâche pour l'évitement des butées :

$$F_b = [F_{b1} \dots F_{bn}]^T$$

La loi de commande pour cette fonction de tâche est définie classiquement :

$$\dot{q}_b = -J_b^+ \lambda_b F_b(q) \quad (4.11)$$

λ_b est un gain positif ou une matrice définie positive et J_b le jacobien de la fonction F_b est donné par :

$$J_b = \begin{bmatrix} J_{b1} & 0 & \dots & 0 \\ 0 & J_{b2} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & J_{bn} \end{bmatrix}$$

où J_{bi} a pour expression :

$$J_{bi} = \begin{cases} (q^i - q_{s+}^i)(2q_{max}^i - q_{s+}^i - q^i) / (q_{max}^i - q^i)^2 & \text{si } q^i \geq q_{s+}^i \\ (q_{s-}^i - q^i)(q^i - 2q_{min}^i - q_{s-}^i) / (q^i - q_{min}^i)^2 & \text{si } q^i \leq q_{s-}^i \\ 0 & \text{sinon} \end{cases} \quad (4.12)$$

Grâce à la définition de notre fonction de tâche, la loi de commande évolue continûment en fonction des coordonnées articulaires.

4.3.2 Contraintes sur la visibilité

L'approche décrite pour l'évitement des butées peut aussi être appliquée pour gérer le problème des occultations, en remplaçant les coordonnées articulaires par les indices visuels. Cependant, il est nécessaire de pouvoir exprimer les intervalles de sécurité. Si cela est relativement simple pour des points [Remazeilles et al., 2006], cela devient plus compliqué pour les indices visuels choisis : ρ , θ et k . Rappelons que ρ et k sont des distances mesurées sur le plan image, tandis que θ représente l'orientation de la ligne. Le choix est fait d'imposer à l'axe du cylindre de rester dans un voisinage du centre de l'image, matérialisé par un cercle de rayon $\min(h, l)/2$ où $l \times h$ sont les dimensions de l'image (cf. figure 4.12).

Nous spécifions donc un intervalle $[\rho_{min}, \rho_{max}] \times [k_{min}, k_{max}]$ pour les indices visuels ρ et k , l'angle θ n'étant pas contraint. Cet intervalle définit un rectangle qui doit être inscrit dans ce cercle, comme sur la figure 4.12. Le rectangle de plus grande aire inscrit dans un cercle de rayon r est le carré de côté $r\sqrt{2}$. Alors nous avons choisi les valeurs limites ci-dessous

$$r = \min(h, l)/2 \quad (4.13)$$

$$\rho_{max} = \sqrt{2}r/2 \quad k_{max} = \sqrt{2}r/2 \quad (4.14)$$

$$\rho_{min} = -\sqrt{2}r/2 \quad k_{min} = -\sqrt{2}r/2 \quad (4.15)$$

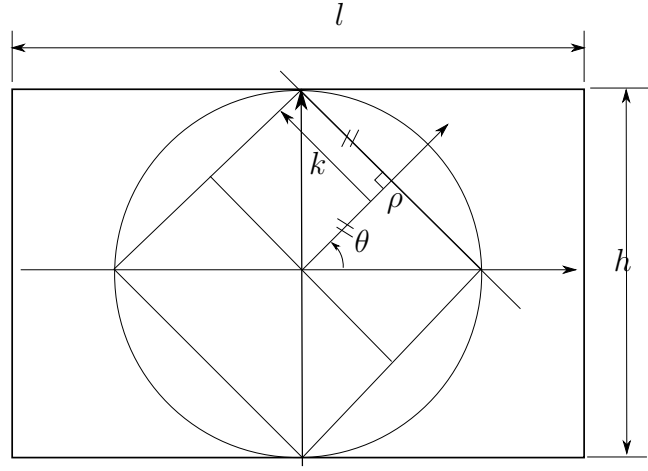


FIGURE 4.12 – Limites des indices visuels choisis.

Les images fournies par les caméras de notre robot ont une dimension de 640×480 pixels. Si nous notons S_a le vecteur de dimensions 12×1 contenant tous les vecteurs d'indices visuels, et S_v , le vecteur de dimensions 8×1 contenant uniquement les valeurs de k et ρ , la relation entre ces deux vecteurs a pour expression :

$$S_a = \begin{bmatrix} {}^m S_c \\ {}^m S_p \\ {}^t S_c \\ {}^t S_p \end{bmatrix} \quad S_v = H_v S_a = \begin{bmatrix} \rho_{mc} \\ k_{mc} \\ \vdots \\ \rho_{fp} \\ k_{fp} \end{bmatrix} \quad (4.16)$$

H_v est une matrice constante de dimensions 8×12 permettant de sélectionner les indices visuels adéquats. Elle est donc donnée ici par :

$$I_{3v} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad H_v = \begin{bmatrix} I_{3v} & 0 & 0 \\ 0 & I_{3v} & 0 \\ 0 & 0 & I_{3v} \end{bmatrix}$$

Pour assurer le maintien dans le champ de vue des objets, nous définissons comme précédemment l'intervalle de sécurité $[s_{si-}, s_{si+}]$:

$$s_{si-} = s_{imin} + (1 - \tau_s) \cdot (s_{imax} - s_{imin}) \quad (4.17)$$

$$s_{si+} = s_{imax} - (1 - \tau_s) \cdot (s_{imax} - s_{imin}) \quad (4.18)$$

avec $[s_{imin}, s_{imax}]$ les valeurs extrêmes de s_i et $\tau_s \in]0, 0.5[$ un paramètre ajustable. Ainsi, s_{si-} et s_{si+} varient entre s_{imin} (respectivement s_{imax}) et la moyenne du débattement admissible. A partir de ces définitions, il devient possible d'exprimer la fonction de tâche $F_v^T = [F_{v1} \dots F_{v8}]$ qui préserve la visibilité des objets. Pour cela, nous suivons un raisonnement similaire à celui que nous avons développé pour traiter le problème des butées articulaires. Nous définissons chaque composante F_{vi} de F_v comme suit :

$$F_{vi} = \begin{cases} (s_i - s_{si+})^2 / (s_{imax} - s_i) & \text{si } s_i \geq s_{si+} \\ (s_{si-} - s_i)^2 / (s_i - s_{imin}) & \text{si } s_i \leq s_{si-} \\ 0 & \text{sinon} \end{cases} \quad (4.19)$$

Le jacobien J_v de cette fonction de tâche par rapport au vecteur d'indices visuels S_v a pour expression :

$$J_v = \begin{bmatrix} J_{v1} & 0 & \dots & 0 \\ 0 & J_{v2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & J_{vn} \end{bmatrix}$$

où J_{vi} est donné par :

$$J_{vi} = \begin{cases} (s_i - s_{si+})(2s_{imax} - s_{si+} - s_i)/(s_{imax} - s_i)^2 & \text{si } s_i \geq s_{si+} \\ (s_{si-} - s_i)(s_i - 2s_{imin} - s_{si-})/(s_i - s_{imin})^2 & \text{si } s_i \leq s_{si-} \\ 0 & \text{sinon} \end{cases} \quad (4.20)$$

Le jacobien de S_a par rapport aux coordonnées articulaires peut être déduit des jacobiens de ses composantes étudiées à la section 3.4, que nous rappelons ci-après :

$$S_a = J_a \begin{bmatrix} \dot{q}_g \\ \dot{q}_d \end{bmatrix} \quad (4.21)$$

où

$$J_a = \begin{bmatrix} {}^mL_c & 0 & 0 & 0 \\ 0 & {}^mL_p & 0 & 0 \\ 0 & 0 & {}^tL_c & 0 \\ 0 & 0 & 0 & {}^tL_p \end{bmatrix} \begin{bmatrix} 0 & -{}^mW_d & I_6 \\ -{}^mW_g & 0 & I_6 \\ 0 & -{}^tW_d & 0 \\ -{}^tW_g & 0 & 0 \end{bmatrix} \begin{bmatrix} J_g & 0 \\ 0 & J_d \\ 0 & J_{5d} \end{bmatrix} \quad (4.22)$$

La loi de commande permettant de préserver la visibilité des indices visuels peut donc s'exprimer ainsi :

$$\dot{q}_v = -(J_v H_v J_a)^+ \lambda_v F_v \quad (4.23)$$

avec λ_v un gain positif ou une matrice définie positive.

4.3.3 Résultats de simulation

Nous avons testé notre approche pour la redondance et les tâches d'évitement des butées et des occultations dans le cas de la fermeture du stylo. Nous avons effectué la même simulation que celle décrite dans le chapitre précédent. Les conditions initiales sont également identiques à celles recensées dans le tableau 3.1. Les tâches sont ordonnées comme sur la figure 4.10.

Le gain λ_1 et λ_2 sont égaux à 0.5 pour e_1 et e_2 et $\lambda_3 = 0.2$ pour e_3 . μ est égal à $3\lambda_i$. Les gains λ_b et λ_v ont été fixés à 2 pour la tâche d'évitement des butées et pour la tâche d'évitement des occultations. Nous avons gardé le bruit d'écart-type 1 pixel sur les mesures dans l'image, et le bruit de covariance 8×10^{-9} radians² sur les mesures des codeurs des articulations et sur la loi de commande calculée. Les limites des articulations utilisées pour cette simulation sont tirées du manuel du PR2 et listées dans le tableau 4.1. Les paramètres τ_b et τ_s réglant les intervalles de sécurité ont été fixés respectivement à 0.04 et 0.1.

Articulation	Bras gauche		Bras droit	
	-	+	-	+
shoulder_pan_joint	-40°	130°	-130°	40°
shoulder_lift_joint	-30°	80°	-30°	80°
upper_arm_roll_joint	-44°	224°	-224°	44°
elbow_flex_joint	0°	133°	0°	133°
wrist_flex_joint	0°	130°	0°	130°

TABLE 4.1 – Limites des articulations des deux bras du PR2.

La figure 4.13 montre l'évolution de la tâche d'évitement des butées. Toutes les composantes F_{bi} non nulles convergent rapidement vers zéro. Seules deux composantes F_{b6} et

F_{b13} sont non nulles au début. Ces courbes correspondent à deux liaisons qui sont près de leur limites. Pour toutes les autres articulations, l'évitement des butées est inactif.

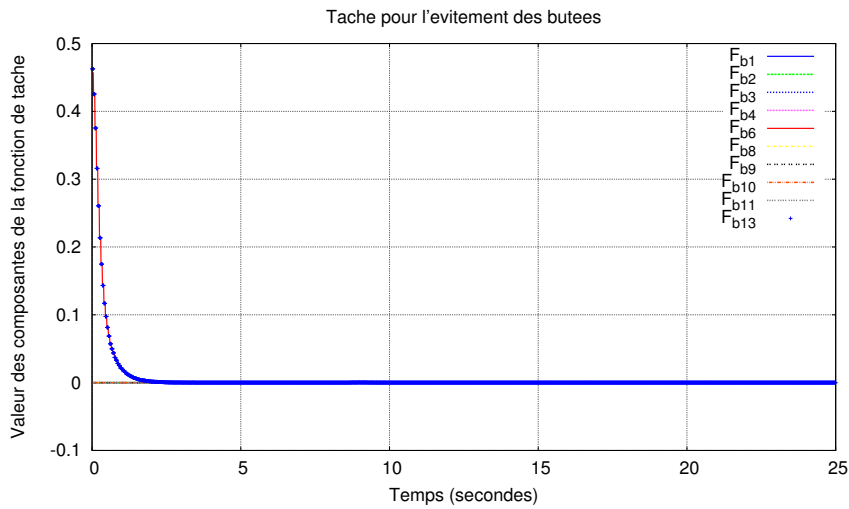


FIGURE 4.13 – Évolution de la fonction de tâche d'évitement des butées F_b .

Les figures 4.14 et 4.15 montrent l'évolution de la même articulation dans deux situations selon que la tâche d'évitement des butées est prise en compte ou non. Comme nous pouvons le voir (cf. figure 4.15), lorsque cette tâche est activée, la contrainte sur les limites articulaires n'est jamais saturée. La tâche peut ainsi être correctement exécutée.

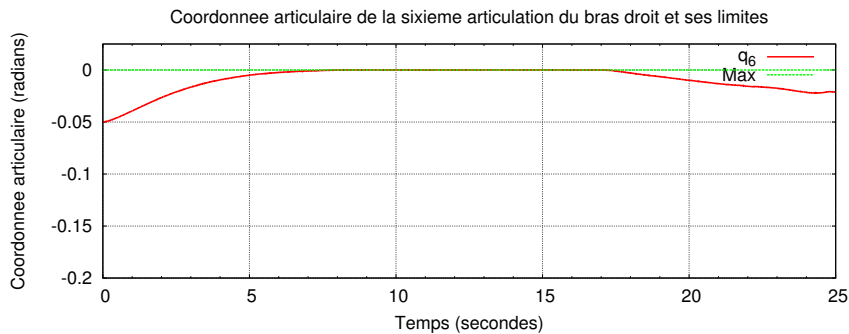


FIGURE 4.14 – Une articulation s'arrête sur sa butée.

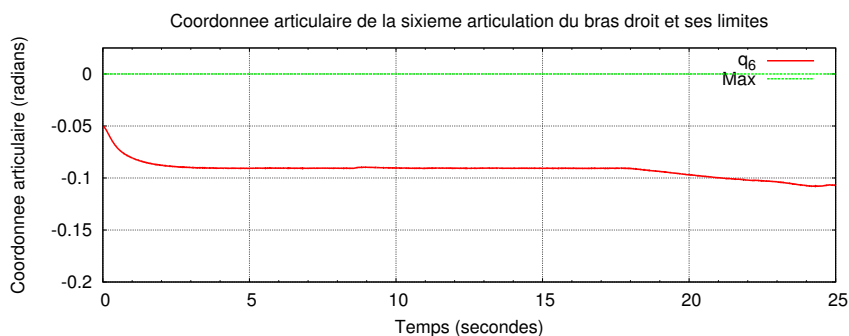


FIGURE 4.15 – L'articulation est maintenue à distance de sa butée.

La figure 4.16 permet de visualiser l'évolution des trois sous-tâches d'asservissement visuel coordonné définies pour la fermeture du stylo dans le chapitre précédent (cf. section 3.4). Toutes les composantes des sous-tâches, définies dans le tableau 3.3, convergent vers zéro. La fermeture du stylo est donc réalisée. Enfin la figure 4.17 présente l'évolution de la

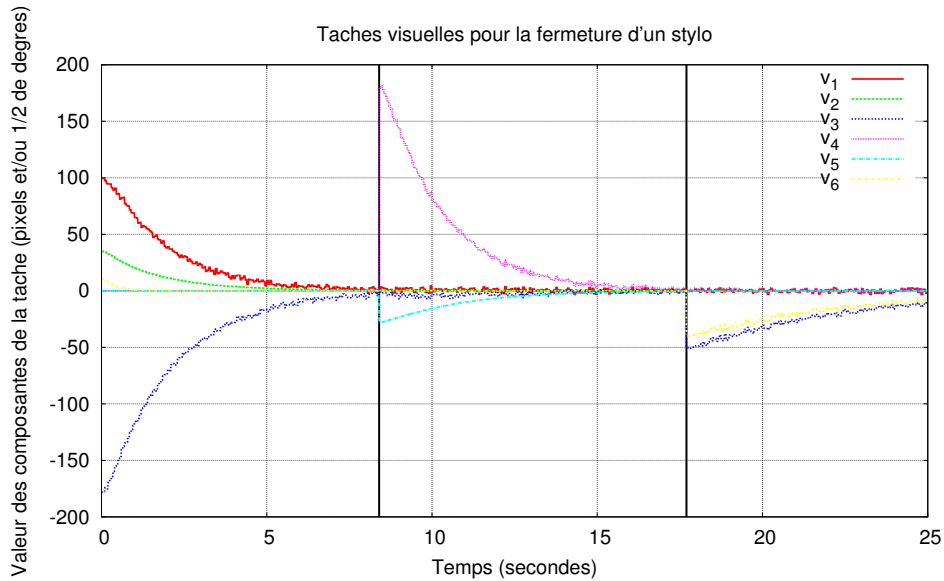


FIGURE 4.16 – Évolution de la séquence des tâches d'asservissement visuel pour la fermeture du stylo.

tâche F_v d'évitement des occultations. Toutes les composantes de cette fonction de tâche convergent rapidement vers zéro, ce qui garantit que les indices visuels restent toujours visibles.

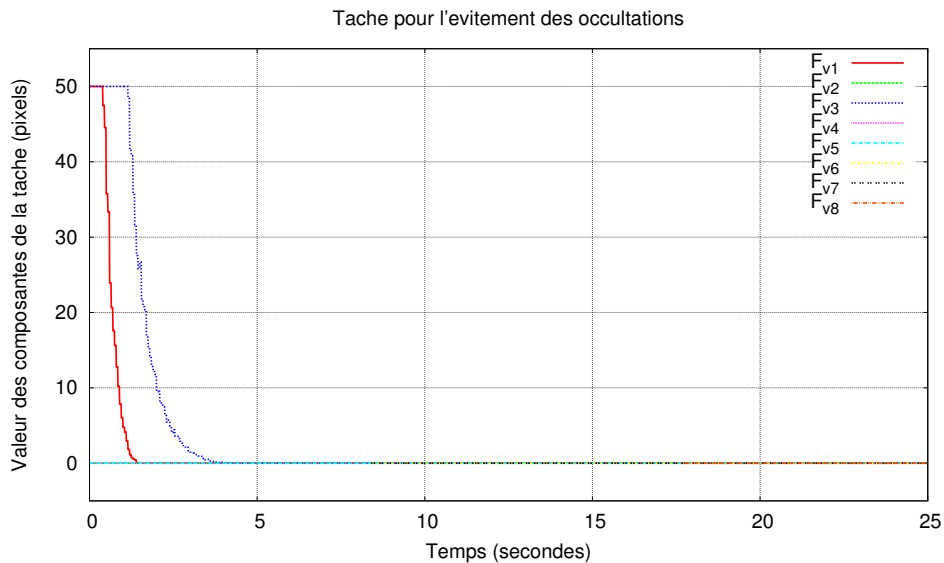


FIGURE 4.17 – Évolution de la fonction de tâche d'évitement des occultations F_v .

Nous présentons également l'évolution des lois de commandes envoyées aux deux bras manipulateurs avec ces tâches dans les figures 4.18 et 4.19. Elles sont similaires, signe de la coordination et de la répartition des déplacements entre les deux bras. Elles sont

continues grâce à notre approche de la redondance et notre gestion de la transition entre les sous-tâches.

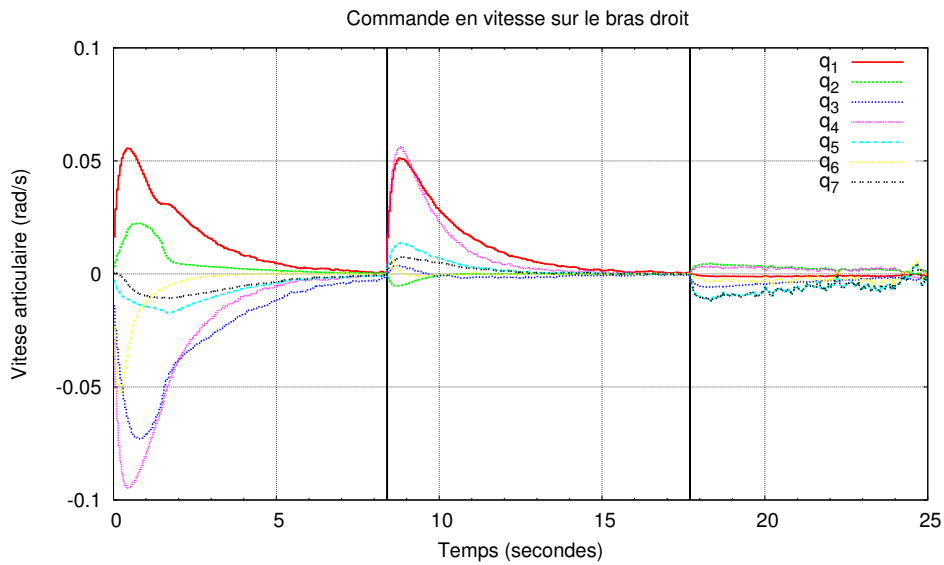


FIGURE 4.18 – Commande en vitesse des 7 articulations du bras droit du PR2.

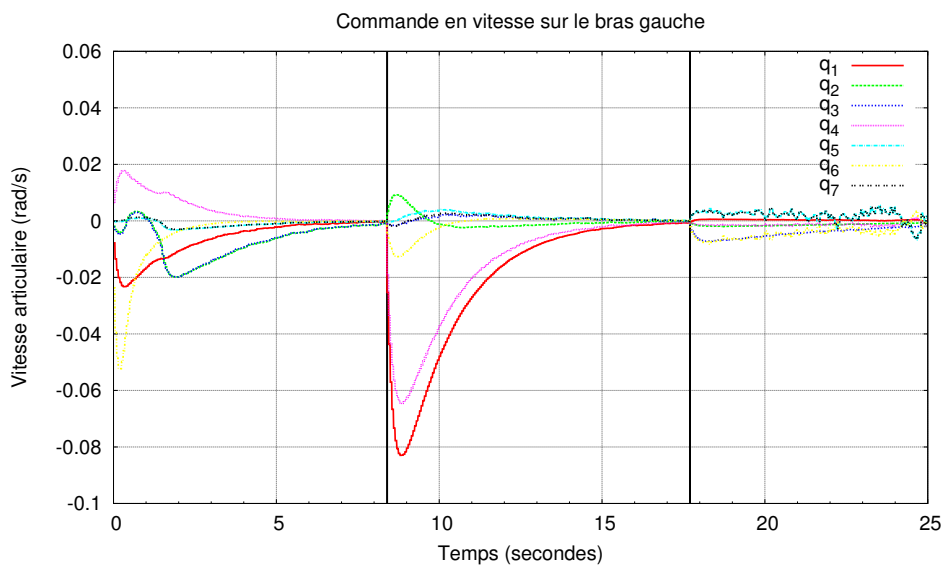


FIGURE 4.19 – Commande en vitesse des 7 articulations du bras gauche du PR2.

Grâce à notre approche de la redondance, toutes les tâches sont réalisées, et les articulations sont maintenues éloignées de leurs butées et les objets manipulés conservés dans le champ de vue de la caméra.

4.4 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la redondance d'un robot. Nous avons rappelé qu'un robot est redondant par rapport à une tâche lorsqu'il compte plus

d'articulations contrôlables que de degrés de libertés contraints par cette tâche. Cette redondance peut être utilisée pour faire réaliser simultanément plusieurs tâches par le robot. Nous avons présenté les principales méthodes qui se basent sur le formalisme de la redondance. Ces approches diffèrent sur la manière d'intégrer les lois de commande individuelles des tâches dans une loi de commande globale. Notre choix s'est porté sur deux approches hiérarchiques qui nous ont paru particulièrement intéressantes. La première, classique, garantit un découplage des différentes tâches ; la seconde, optimale, assure en plus un transfert d'information des tâches prioritaires vers les tâches secondaires. Cette deuxième approche, optimale pour les tâches secondaires sous contrainte de non perturbation des tâches prioritaires, souffre de certaines singularités algorithmiques. Nos efforts se sont portés sur l'identification des situations menant à ces singularités. Nous avons pu les détecter et proposer un basculement entre ces deux approches en fonction de la situation. Nous avons porté un soin particulier à la continuité de la loi de commande. Nous avons donc proposé une gestion sécurisée et efficace de la redondance. Nous avons validé notre approche pour le cas d'un système simple à deux degrés de liberté. Cette simulation a montré que cette approche réunissait les avantages de deux méthodes sans leurs inconvénients.

Nous avons ensuite appliqué notre approche à la tâche d'asservissement visuel coordonné consistant à refermer un stylo. Nous avons défini une tâche prioritaire pour l'évitement des butées et une tâche pour le maintien dans le champ de vue des caméras des objets manipulés. Nous avons pu valider en simulation, la fermeture du stylo tout en évitant les butées et en assurant le maintien des objets manipulés dans le champ de vue des caméras. Une version de notre travail a été publiée dans la conférence internationale ECMSM [Fleurmond and Cadenat, 2015].

La redondance nous offre donc des solutions permettant d'éviter des problèmes tels que les limites articulaires ou les occultations. Cependant, la perte du signal visuel ne peut pas toujours être évitée. Ainsi, par exemple, une panne de la caméra ou un ralentissement de la cadence d'obtention des images peut survenir. De même, le traitement d'images permettant d'extraire les indices visuels peut faillir, suite à des variations d'éclairage ou bien lorsque les objets manipulés deviennent trop proches. Notons que ce problème se produit à la fin de notre tâche référencée vision coordonnée, lorsque le capuchon est quasiment posé sur le stylo. Dans ces situations, les indices visuels ne peuvent plus être extraits et notre asservissement n'est plus alimenté. Pour pallier ce problème, nous nous proposons de localiser les objets pour reconstruire les indices visuels. Le chapitre suivant sera centré sur cette problématique.

Chapitre 5

Estimation et reconstruction d'indices visuels

Dèyè mòn, gen mòn - littéralement
par-delà un morne (colline ou
montagne), il y en a d'autres.

Proverbe haïtien

5.1 Introduction

Les chapitres précédents ont traité de la manipulation coordonnée d'objets et de la gestion de la redondance. Au chapitre 3, un formalisme de la coordination par l'asservissement visuel 2D a été proposé. Les tâches proposées dépendent de l'extraction des indices visuels pour calculer la loi de commande. Ces indices visuels peuvent faire défaut, en cas de panne de caméra, de délais dans le traitement des images, d'occultations des objets manipulés. Dans les tâches d'assemblage, différents cas peuvent se produire : les objets peuvent se masquer mutuellement ou bien être occultés par des portions du robot. Ce problème n'est pas résoluble par la tâche d'évitement d'occultations définie au chapitre précédent (cf. section 4.3.2). Plusieurs travaux ont tenté de rendre les tâches référencées vision plus robustes aux occultations partielles à défaut de pouvoir les éviter [Mezouar and Chaumette, 2002a, Garcia-Aracil et al., 2005, Remazeilles et al., 2006, Kazemi et al., 2010, Kermorgant and Chaumette, 2011b]. Mais lorsque l'occultation est totale, ces techniques ne suffisent pas. Pour continuer à exécuter l'asservissement visuel, une solution envisageable est de virtuellement projeter l'objet d'intérêt dans l'image [Folio and Cadenat, 2008, Durand Petiteville et al., 2013]. Nous souhaitons suivre ici une démarche similaire. Mais, pour cela, il faut connaître la structure de cet objet dans le repère de la caméra.

De plus, la connaissance de cette structure va s'avérer nécessaire pour le calcul des lois de commande référencées vision développées. En effet, ces lois de commande reposent sur la matrice d'interaction qui relie le mouvement des indices visuels dans l'image à celui de la caméra dans l'espace (équations (3.4) et (3.10)). Or, cette matrice dépend de paramètres 3D caractérisant l'objet, a priori inconnus [Chaumette, 1990]. Il faut alors l'estimer. Différentes solutions ont été proposées dans la littérature. La plus courante consiste à l'approximer par sa valeur à la pose souhaitée [Chaumette and Hutchinson, 2007]. Cependant, cette approche ne peut pas fonctionner pour les tâches référencées vision coordonnées considérées ici puisque la pose finale à atteindre est inconnue¹. La

1. Rappelons en effet que l'attention a été portée sur la pose relative entre les objets manipulés, leur pose absolue au moment du rendez-vous étant inconnue et volontairement non spécifiée.

reconstruction de la structure 3D est donc une étape nécessaire pour le calcul de nos lois de commande.

Dans ce chapitre, notre objectif est donc de proposer des méthodes d'estimation de la structure 3D des objets manipulés. Les paramètres ainsi estimés nous permettront de calculer les matrices d'interaction nécessaires au calcul des lois de commande référencées vision et de reconstruire les indices visuels en cas d'occultation. Les méthodes développées devront être adaptées au contexte de la manipulation coordonnée. En particulier, nous rappelons que nous utilisons une caméra fixe et une caméra mobile. Nous supposons que ces caméras sont calibrées. En outre, il est important de noter que nous travaillons dans une scène dynamique puisqu'autant les capteurs de vision que les objets sont mobiles. Cependant, compte tenu de notre contexte, il est à noter que tous les mouvements sont connus. En effet, les objets et les caméras étant portés par les bras, il est possible de déterminer leur mouvements à partir des modèles géométriques et cinématiques directs du robot.

Notre contribution dans ce chapitre sera donc de proposer des solutions au problème de l'estimation en ligne de paramètres 3D décrivant un objet mobile en exploitant les images fournies par une seule caméra mobile également. Cette approche peut ensuite être facilement étendue au cas de plusieurs caméras, fixes ou mobiles. Nous décrivons un estimateur pour le point, et proposons une approche originale pour l'estimation de la droite et du cylindre. À partir de ces données, nous serons en mesure de calculer la structure 3D du stylo et de son capuchon (cf. section 3.4).

5.2 État de l'art

L'estimation de la profondeur ou le calcul des coordonnées 3D des structures est un champ de recherche très actif en vision. Ces méthodes sont connues sous le nom de "structure from motion" ou SFM en abrégé. En général le mouvement des caméras est supposé inconnu ou imparfaitement connu, et le but est d'estimer à la fois les poses des caméras et la structure 3D de la scène.

Une technique répandue dans le domaine de la vision est l'ajustement de faisceaux ("*bundle adjustment*") [Triggs et al., 2000]. Cette technique se base sur la minimisation des erreurs de projection sur toutes les images en fonction de la structure 3D et de la calibration intrinsèque et extrinsèque des caméras. Elle est très utilisée pour un grand nombre de vues et généralement dans un contexte hors ligne (*algorithmes "batch"*). Le critère à minimiser est non linéaire par rapport aux variables, et souvent posé sous la forme d'un problème des moindres carrés ou d'une maximisation de vraisemblance. Les algorithmes d'optimisation utilisés sont généralement itératifs comme celui de Levenberg-Marquardt et nécessitent une solution initiale assez proche de la solution recherchée [Taylor and Kriegman, 1995, Beardsley et al., 1997]. Cependant, cette solution initiale doit être trouvée par d'autres moyens.

Différentes possibilités existent dans la littérature. De nombreuses approches décomposent souvent le problème en deux étapes : l'estimation des poses et des paramètres intrinsèques des caméras, puis la triangulation de la structure 3D à partir des poses estimées [Bartoli and Sturm, 2005]. Si les poses des caméras sont inconnues, elles sont généralement calculées en utilisant la géométrie épipolaire et les homographies, en déterminant notamment la matrice fondamentale [Ma et al., 2001]. L'algorithme des 8 points est couramment utilisé dans ce but [Ma et al., 2001, Hartley and Zisserman, 2003, Forsyth

and Ponce, 2002]. La calibration des caméras étant connue, reste le calcul de la structure 3D qui est un problème de triangulation. Avec les images d'un point 3D dans deux poses de caméras, il est possible de calculer les coordonnées de ce point avec une simple équation linéaire [Ma et al., 2001]. A causes des différentes erreurs, de mesure dans l'image, ou dans la calibration des caméras, ces rayons ne sont généralement pas sécants. Plusieurs méthodes ont donc été proposées pour déterminer la structure 3D en présence d'erreurs. Ces approches se divisent en général en deux écoles selon que l'on aborde le problème sous l'angle de la vision ou bien sous celui de l'automatique.

Dans les travaux de la communauté **vision**, des méthodes linéaires ont été proposées pour estimer la position d'un point en utilisant deux images, avec néanmoins une sensibilité au bruit [Taylor and Kriegman, 1995, Ma et al., 2001, Bartoli and Sturm, 2005]. Des approches *optimales* [Hartley and Sturm, 1997, Kanatani et al., 2008] ont été développées pour la triangulation en présence de bruit. Elles calculent les coordonnées de nouveaux points sur l'image qui respectent la contrainte épipolaire et qui minimisent l'erreur de re-projection, comme le montre la figure 5.1.

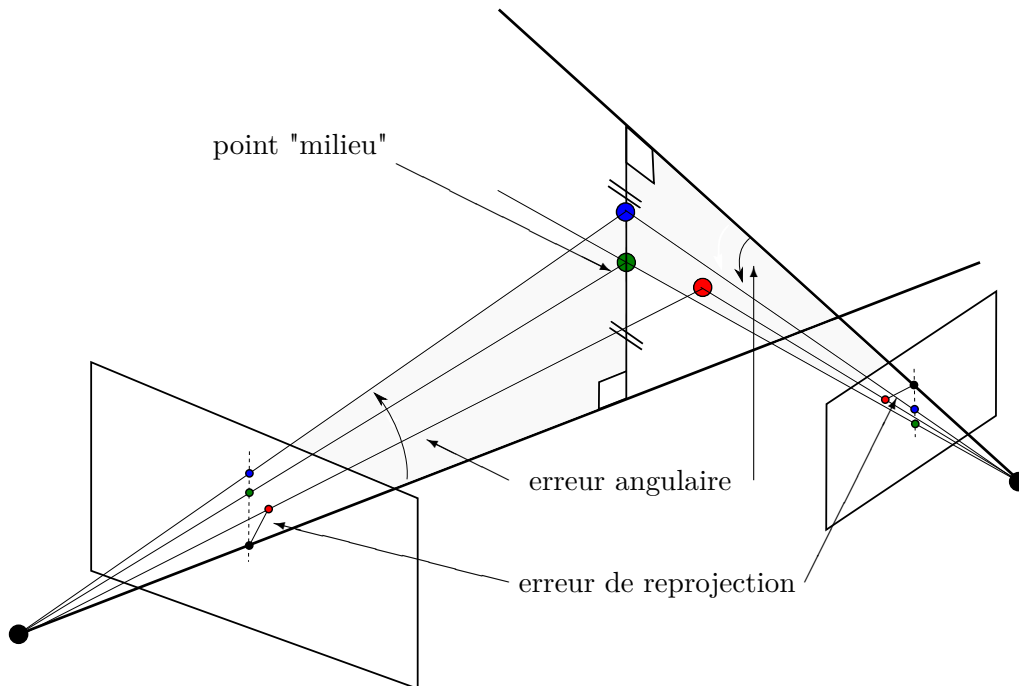


FIGURE 5.1 – Illustrations de différentes méthodes de triangulation. Les deux droites noires épaisses correspondent aux mesures dans les images. Le point vert est donné par la méthode du "midpoint", à égale distance des deux faisceaux. Le point bleu minimise l'erreur "angulaire". Le point rouge minimise l'erreur de re-projection dans les deux images.

En assurant que les rayons correspondant aux nouveaux points images se rencontrent, les coordonnées du point 3D peuvent être calculées avec une équation linéaire. Ces approches se disent optimales parce qu'elles minimisent une somme de carrés de distances sur les images, en supposant généralement un modèle de bruit gaussien centré sur l'image. La fonction de coût obtenue est un polynôme de degré 6 dans le cas de deux images, et son degré augmente avec le cube du nombre d'images utilisées [Recker et al., 2013]. Des solutions analytiques ont été exhibées pour trouver l'optimum global de cette fonction pour trois images. Ces approches sont coûteuses en calcul et semblent être difficilement généralisables pour un plus grand nombre d'images [Recker et al., 2013]. Récemment, Recker et al. [Recker et al., 2013] ont proposé une méthode de triangulation pour n images avec un coût de calcul au mieux linéaire en fonction de n et au pire quadratique $O(n^2)$.

Cette approche se base sur la minimisation d'une erreur angulaire, s'écartant du modèle classique d'un bruit centré et gaussien sur l'image [Hartley and Sturm, 1997, Ramalingam et al., 2006]. La figure 5.1 met en évidence cette erreur angulaire et montre les résultats obtenus dans le cas de deux vues. Il est aussi possible de minimiser une distance euclidienne comme la méthode dite du point milieu ("*midpoint*"). Cette méthode détermine le point 3D qui est à égale distance des deux faisceaux donnés par les deux points images. Elle est également illustrée sur la figure 5.1. Cette approche n'est valide que dans un repère euclidien. Elle n'a pas de signification dans les transformations projectives ou affines, parce que les perpendiculaires et/ou les distances ne sont conservées. Elle est rapide, et est souvent utilisée pour fournir des solutions initiales à des problèmes de minimisation [Beardsley et al., 1997, Ramalingam et al., 2006, Recker et al., 2013]. Elle fournit une des meilleures reconstructions dans l'espace euclidien [Hartley and Sturm, 1997] mais est pourtant décrite comme imprécise ou non optimale (au sens de la minimisation des erreurs de reprojection) [Hartley and Sturm, 1997, Hartley and Zisserman, 2003] dans le cas de deux images. Elle est aisément généralisable au cas de n images [Beardsley et al., 1997, Ramalingam et al., 2006].

Des travaux issus de la communauté **automatique** ont été menés sur l'estimation de la profondeur ou de la structure 3D. Dans leur majorité, les paramètres à estimer sont regroupés dans un vecteur d'état qui est mis à jour à chaque nouvelle mesure. De nombreuses approches sont basées sur des variantes du filtre de Kalman étendu (**E**xtended **K**alman **F**ilter) et sont utilisées pour faire de la cartographie et de la localisation simultanée (*SLAM pour "Simultaneous Localisation And Mapping"*) [Matthies et al., 1989, Klein and Murray, 2007, Jones and Soatto, 2010, Martinelli, 2012]. Ces méthodes reconstruisent la pose de la caméra et la structure 3D, en prenant en compte explicitement le bruit, mais linéarisent localement un modèle de la dynamique de la caméra.

Chaumette et al. [Chaumette et al., 1996] ont proposé une méthodologie pour estimer les paramètres 3D des structures connaissant le mouvement de la caméra. Cette approche continue peut être utilisée pour différents types de structures. Les paramètres 3D décrivant le contour de l'objet vu sont déterminés en comparant le mouvement des indices visuels mesuré dans l'image et celui calculé à partir du torseur cinématique connu de la caméra. Mais pour minimiser l'effet du bruit et des erreurs, ils doivent restreindre les mouvements possibles de la caméra. Ils déterminent également le mouvement de la caméra à réaliser qui améliore l'estimation de la structure.

Récemment, De Luca [De Luca et al., 2007] a construit un observateur non linéaire déterministe de la profondeur d'un point fixe à partir de la relation analytique entre le torseur cinématique de la caméra et le mouvement de l'image comme [Chaumette et al., 1996]. La différence majeure réside dans le fait que le mouvement de la caméra n'est plus contraint, et que la mesure du mouvement des indices visuels n'est plus strictement nécessaire. Mais, une condition nécessaire pour garantir la convergence de cet observateur est une vitesse non nulle de la caméra, qui ne doit pas se déplacer dans la direction du point à estimer. Il utilise ensuite cet observateur dans une boucle d'asservissement visuel [De Luca et al., 2008] pour le calcul de la loi de matrice d'interaction. Un observateur de la focale est aussi proposé en utilisant le mouvement des indices visuels dû seulement à la rotation de cette caméra. Il vaut la peine de mentionner qu'il a choisi comme état du système les coordonnées de l'image du point et l'inverse de sa profondeur.

Spica [Spica and Robuffo Giordano, 2013] a porté son attention à la vitesse de convergence d'une classe d'observateurs non-linéaires qui peuvent décrire la majorité des problèmes de SFM. Cette classe comprend l'observateur proposé par De Luca [De Luca et al., 2007] qu'il réutilise. Il peut ainsi calculer le mouvement de la caméra qui optimise la convergence de l'observateur dans un schéma d'asservissement visuel [Spica et al.,

2014a, Spica et al., 2014b, Spica et al., 2014c] et proposer de nouveaux observateurs pour la sphère et le cylindre, en réduisant l'état à un seul paramètre à estimer. Grave et Tang [Grave and Tang, 2015] utilisent la théorie de la contraction pour améliorer les performances des observateurs non linéaires en présence de bruit.

Enfin, Durand Petiteville et al. [Durand Petiteville et al., 2010, Durand Petiteville et al., 2013] adaptent un algorithme développé par Folio [Folio, 2007, Folio and Cadenat, 2008] et proposent un "prédicteur/correcteur" pour estimer la profondeur d'un point à partir d'un nombre n_c d'images. Cette approche présente certains traits des travaux de la communauté vision, même si elle met à jour un état. Il s'agit ici de trouver la profondeur du point qui minimise une fonction de coût quadratique définie à partir de l'erreur de reprojection, calculée sur n_c images. En cas d'occultation [Durand Petiteville et al., 2013], il est possible de mettre à jour la profondeur estimée et l'image du point en intégrant le mouvement de la caméra.

5.3 Discussion

Nous cherchons une méthode d'estimation qui puisse être utilisée en ligne pendant l'exécution de notre asservissement visuel. Il nous paraît donc raisonnable, d'écarter pour l'instant, malgré les nombreuses améliorations déjà apportées, les méthodes d'optimisation itératives comme l'ajustement de faisceaux [Triggs et al., 2000], et même les méthodes dites "*optimales*" [Hartley and Sturm, 1997, Kanatani et al., 2008]. Nous souhaitons une méthode d'estimation itérative qui puisse intégrer les nouvelles mesures et mettre à jour l'estimation avec un coût de calcul constant. Les méthodes proposées dans [Recker et al., 2013, Hartley and Sturm, 1997] ne garantissent pas ce coût de calcul.

Les méthodes s'appuyant sur les variantes du filtre de Kalman [Matthies et al., 1989] linéarisent une fonction d'erreur fortement non linéaire. Elles effectuent, à chaque nouvelle mesure, l'équivalent de la première itération d'un algorithme d'optimisation alors que la fonction d'erreur et l'état changent entre-temps. Il leur faut aussi une initialisation correcte pour assurer une bonne convergence. Enfin les mouvements possibles de la caméra sont en outre restreints, ce qui n'est pas adaptable à notre contexte où les cibles et les caméras vont se déplacer de manière à ce qu'une tâche coordonnée soit réalisée. Les observateurs non linéaires [De Luca et al., 2007, Spica and Robuffo Giordano, 2013] ont été proposés pour des scènes fixes (ce qui ne correspond pas à notre cas) et leur convergence dépend de la vitesse instantanée de la caméra. Ils peuvent de plus être sujet à la dérive. Nous ne les retiendrons donc pas ici.

La méthode proposée par Durand Petiteville [Durand Petiteville et al., 2013] possède un coût de calcul constant parce que le nombre d'images retenues n_c est constant. La qualité de l'estimation croît avec le nombre d'images retenues, car le rapport signal/ bruit augmente. Mais la précision de l'estimation dépend du déplacement relatif de la caméra pendant l'acquisition des n_c images. Elle se dégrade si la caméra ralentit ou s'immobilise. Cette méthode reste pour l'instant encore restreinte au cas des scènes statiques.

Inspiré par [Durand Petiteville et al., 2013], nous voulons construire un estimateur de la structure 3D, à partir de toutes les n images disponibles. Cet estimateur minimiserait un critère sur les n mesures avec un coût de calcul linéaire en fonction de n , et sa version récursive présenterait un coût de calcul constant. La méthode dite du point milieu ("*mid-point*"), illustrée sur la figure 5.1, remplit ces conditions. Elle est généralement utilisée pour fournir des solutions initiales à d'autres méthodes de SFM pour n vues [Beardsley et al., 1997, Ramalingam et al., 2006, Recker et al., 2013]. Hartley et Sturm [Hartley and Sturm, 1997] ont comparé cette méthode avec les méthodes de triangulations courantes en vision, dans le cas de **deux** images. Ils reconnaissent sa rapidité et la précision de la reconstruction 3D dans l'espace euclidien, mais ne la recommandent pas. Ils lui pré-

fèrent les méthodes linéaires ou "*optimales*". Ces dernières sont conçues pour minimiser l'erreur de re-projection et sont donc optimales selon ce critère. Ce critère d'optimalité est généralement retenu sous l'hypothèse d'un bruit gaussien centré sur l'image. Cette hypothèse, bien que satisfaisante dans la plupart des cas, n'étant pas prouvée, le choix d'autres critères reste possible, comme la minimisation de l'erreur angulaire ou de la distance euclidienne [Hartley and Sturm, 1997, Recker et al., 2013, Triggs et al., 2000]. Ce dernier critère, la minimisation de la distance euclidienne est celui utilisé dans la méthode du point milieu. Nous retenons cette méthode et nous montrerons que ses performances augmentent avec le nombre d'images, et sont suffisantes pour une utilisation dans une tâche référencée vision.

Dans les tâches considérées, les objets se déplacent, de même que l'une des deux caméras utilisées. Nous sommes donc dans le contexte d'une scène dynamique. Cependant, il est important de noter que tous les mouvements sont connus. En outre, les cibles sont vues simultanément par plusieurs caméras. N'ayant pas de repère caméra à privilégier, nous préférons estimer la structure 3D des cibles dans les repères qui leur sont liés. De cette manière, notre problème se ramène systématiquement à l'estimation de paramètres *constants*, indépendants du mouvement de l'objet, ce qui simplifie grandement notre problème. Ensuite, ayant à disposition les différents modèles géométriques des bras robotiques, il sera possible de déduire les informations utiles pour le calcul de la matrice d'interaction et des indices visuels.

5.4 Estimation du point

Dans les lignes qui suivent nous décrivons la méthode du point milieu, pour ensuite appliquer le même raisonnement à d'autres primitives géométriques : la ligne droite, la sphère et le cylindre.

5.4.1 Méthode du point milieu

Nous décrivons ici la méthode du point milieu dans le cas général de n images. Le point estimé minimise la somme du carré des distances du point estimé aux rayons issus du centre des caméras (cf. figure 5.1). Cette méthode n'est valide que dans les cas où l'on considère un repère euclidien, et où le point à estimer n'est pas situé à l'infini. Dans notre cas, ces deux conditions sont satisfaites puisque les points que nous chercherons à estimer sont exprimés dans le repère euclidien lié à l'objet et se situent donc à une distance bornée de l'origine. De plus, cette méthode a un coût de calcul linéaire en fonction du nombre d'images $O(n)$, et sa version récursive que nous décrivons a un coût de calcul constant $O(1)$. Le ratio signal/bruit augmente avec le nombre d'images, améliorant la précision de l'estimation, tout en conservant la rapidité et le faible coût de calcul. La formule que nous proposons peut être retrouvée dans [Beardsley et al., 1997, Ramalingam et al., 2006]. Nous la reprenons ici avec notre propre formulation car elle sera ensuite étendue pour la reconstruction d'autres types d'indices visuels.

Nous notons F_o le repère lié à l'objet dans cette section et les suivantes. Par souci de lisibilité, les composantes d'un vecteur V ou les coordonnées d'un point M dans le repère F_o seront respectivement notés \bar{V} et \bar{M} . Nous notons C_i la $i^{\text{ème}}$ position du centre optique d'une caméra.

Considérons la figure 5.2. Une caméra se déplace dans un environnement et observe un point P appartenant à un objet en mouvement à partir de différentes positions C_i ($i = 1, 2, \dots, n$). Pour chacune de ces positions, on perçoit dans l'image un point de coordonnées

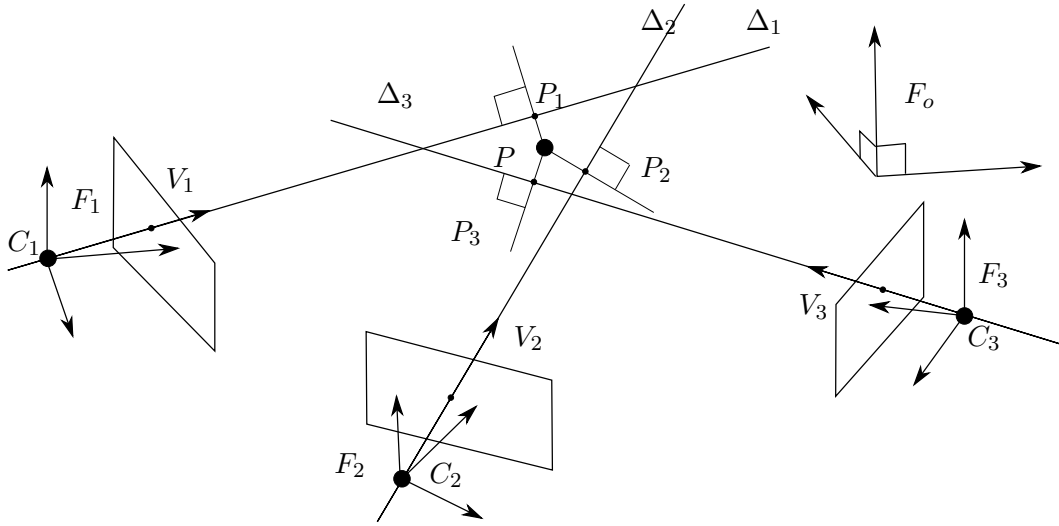


FIGURE 5.2 – Généralisation de la méthode du point milieu à n vues.

(x_i, y_i) . Il est ainsi possible de définir la droite Δ_i qui représente le rayon reliant le point C_i au point (x_i, y_i) mesuré dans l'image. Nous cherchons à déterminer les coordonnées de P dans le repère F_o lié à l'objet. Pour atteindre notre objectif, calculons les composantes de V_i , le vecteur directeur de Δ_i dans le repère lié à la caméra F_i :

$$V_i = \frac{1}{\sqrt{x_i^2 + y_i^2 + f^2}} [x_i \quad y_i \quad f]^T \quad (5.1)$$

avec f la focale de la caméra, supposée connue, puisque cette dernière a été préalablement calibrée. La transformation entre les repères F_i et F_o étant connue également d'après les hypothèses formulées dans la section 5.1, nous pouvons calculer \bar{V}_i , la projection de V_i dans le repère F_o . Afin de pouvoir poser notre problème d'optimisation, nous introduisons maintenant le point P_i , le point de la droite Δ_i le plus proche de P (cf. figure 5.2). Ses coordonnées dans le repère F_o peuvent être déduites en projetant orthogonalement le vecteur $\overrightarrow{C_i P}$ sur la droite Δ_i . Il vient donc :

$$\bar{P}_i = \bar{V}_i \bar{V}_i^T (\bar{P} - \bar{C}_i) + \bar{C}_i \quad (5.2)$$

A partir de cette relation, il est possible de déterminer le vecteur $\overrightarrow{P_i P}$ dans le repère F_o . Nous avons :

$$\bar{P} - \bar{P}_i = \bar{P} - \bar{V}_i \bar{V}_i^T (\bar{P} - \bar{C}_i) - \bar{C}_i \quad (5.3)$$

$$\bar{P} - \bar{P}_i = (I_3 - \bar{V}_i \bar{V}_i^T) (\bar{P} - \bar{C}_i) \quad (5.4)$$

où $\Pi_i = I_3 - \bar{V}_i \bar{V}_i^T$ est un projecteur orthogonal de rang 2. Nous pouvons maintenant déduire le carré de la distance séparant les point P et P_i comme suit :

$$e_i^p = (\bar{P} - \bar{P}_i)^T (\bar{P} - \bar{P}_i) \quad (5.5)$$

$$e_i^p = (\bar{P} - \bar{C}_i)^T \Pi_i (\bar{P} - \bar{C}_i) \quad (5.6)$$

En développant, l'expression du carré de la distance entre P et P_i devient :

$$e_i^p = \bar{P}^T \Pi_i \bar{P} - 2\bar{P}^T \Pi_i \bar{C}_i + \bar{C}_i^T \Pi_i \bar{C}_i \quad (5.7)$$

Notre problème d'optimisation consiste alors à trouver \bar{P} qui minimise le critère :

$$E_n^p = \sum_{i=1}^n e_i^p \quad (5.8)$$

avec n le nombre d'images utilisées. Ce critère peut s'écrire sous la forme quadratique suivante :

$$E_n^p = \bar{P}^T \Phi_n \bar{P} - 2\bar{P}^T \beta_n + \sigma_n \quad (5.9)$$

avec

$$\Phi_n = \sum_{i=1}^n \Pi_i \quad \beta_n = \sum_{i=1}^n [\Pi_i \bar{C}_i] \quad \sigma_n = \sum_{i=1}^n [\bar{C}_i^T \Pi_i \bar{C}_i] \quad (5.10)$$

Φ_n est une matrice de dimensions 3×3 , β_n un vecteur de dimensions 3×1 et σ_n est un scalaire. L'estimateur \hat{P}_n de \bar{P} qui minimise le critère est donné par :

$$\hat{P}_n = [\Phi_n]^+ \beta_n \quad (5.11)$$

où Φ_n^+ est la pseudo-inverse de la matrice Φ_n . Etudions maintenant l'inversibilité de cette matrice. Φ_n est une matrice hermitienne de dimension 3×3 . Deux cas peuvent se produire :

- Si tous les vecteurs V_i sont linéairement dépendants, c'est-à-dire, si le point P est aligné avec toutes les positions C_i de la caméra (cas dégénéré), Φ_n est semi-définie positive et de rang 2. Il existe alors une infinité de solutions données par tous les points de la droite Δ_i . La pseudo inverse nous donne dans ce cas la solution de norme minimale qui correspond ici au point solution le plus proche de l'origine de F_o .
- Dans tous les autres cas, avec $n > 1$, Φ_n est définie positive et inversible. Il y a donc une unique solution au problème qui est le minimum global de notre critère, puisqu'il s'agit d'une fonction convexe.

La méthode d'estimation que nous proposons peut être mise en œuvre aisément sous la forme itérative suivante :

$$\Pi_{n+1} = I_3 - \bar{V}_{n+1} \bar{V}_{n+1}^T \quad (5.12)$$

$$\Phi_{n+1} = \Phi_n + \Pi_{n+1} \quad (5.13)$$

$$\beta_{n+1} = \beta_n + \Pi_{n+1} \bar{C}_{n+1} \quad (5.14)$$

$$\hat{P}_{n+1} = [\Phi_{n+1}]^+ \beta_{n+1} \quad (5.15)$$

Ainsi, toute l'information des n dernières images concernant P est conservée dans Φ_n et β_n . Le calcul de l'estimation revient à quelques additions et multiplications de matrices, et à une inversion d'une matrice symétrique positive de dimensions 3×3 . Cette méthode est donc très peu coûteuse en temps de calcul et s'avère très simple en termes d'implantation. La reconstruction des indices visuels correspondant au point se fait immédiatement en utilisant le modèle sténopé de la caméra et la projection perspective (cf. section 3.4.1.2). Le calcul de la matrice d'interaction est rappelé en annexe A.

5.4.2 Résultats de simulation

Nous avons testé l'estimateur du point en simulation. Cette simulation est inspirée de celle proposée dans [Hartley and Sturm, 1997], où l'utilisation de la méthode du point milieu pour deux images est déconseillée. La figure 5.3 illustre les conditions de simulation. Une cible de quatre points, classique en asservissement visuel, est placée en face d'une caméra. Les positions extrêmes de la caméra sont espacées d'une unité. La position la plus proche de la caméra est à 0.15 unité de la cible. Les paramètres intrinsèques de la caméra ont été fixés à ces valeurs :

$$M = \begin{bmatrix} f/\alpha_u & 0.0 & c_u \\ 0.0 & f/\alpha_v & c_v \end{bmatrix} = \begin{bmatrix} 700 & 0.0 & 350 \\ 0.0 & 700 & 350 \end{bmatrix} \quad (5.16)$$

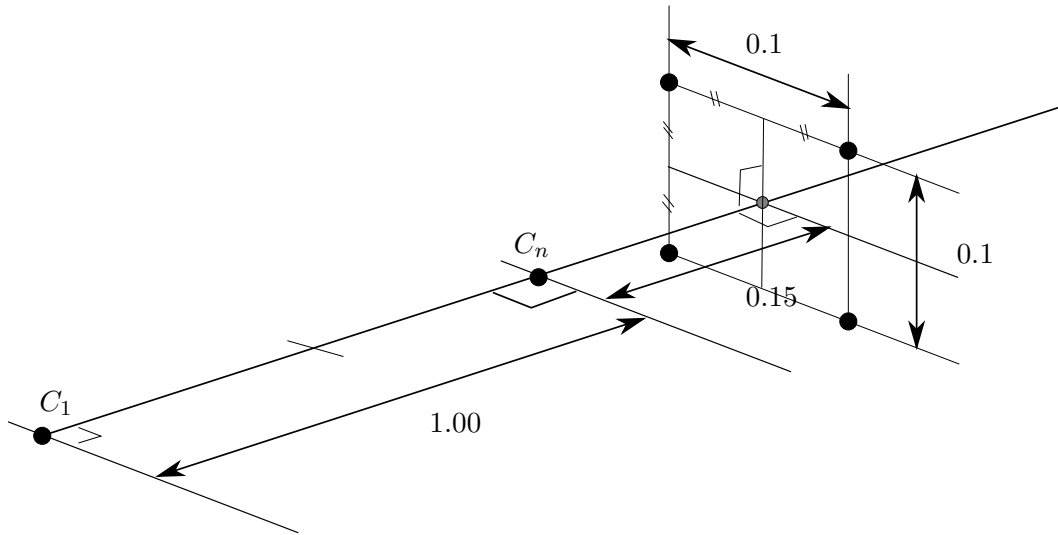


FIGURE 5.3 – Conditions de simulations

Pour étudier l'impact du nombre n de vues sur la précision de l'estimation, nous avons divisé le segment de longueur 1 unité en $n - 1$ segments de longueur égale. Par exemple, pour trois vues, les trois positions de la caméra sont espacées de 0.5 unité, et pour cinq, elles sont espacées de 0.25 unité.

Nous avons ajouté à la mesure dans l'image de la projection des points, un bruit gaussien centré isotrope de variance variable. Nous avons fait varier l'écart-type du bruit de 0.1 à 10 pixels. Pour chaque niveau de bruit, nous avons effectué 5000 réalisations. Nous avons étudié l'erreur médiane² de reconstruction dans l'espace euclidien et l'erreur médiane de reprojection des points, en fonction du niveau de bruit dans l'image et du nombre de vues. L'erreur médiane de reconstruction est la médiane des distances euclidiennes entre les points réels et les points estimés pour toutes les réalisations pour un niveau de bruit donné. L'erreur médiane de reprojection est la médiane des distances sur l'image entre la projection sans bruit des vrais points, et celle des points estimés, pour toutes les positions de la caméra utilisées pour l'estimation, sur toutes les réalisations et pour un niveau de bruit donné. Nous illustrons la définition de nos erreurs de reconstruction et de reprojection dans la figure 5.4.

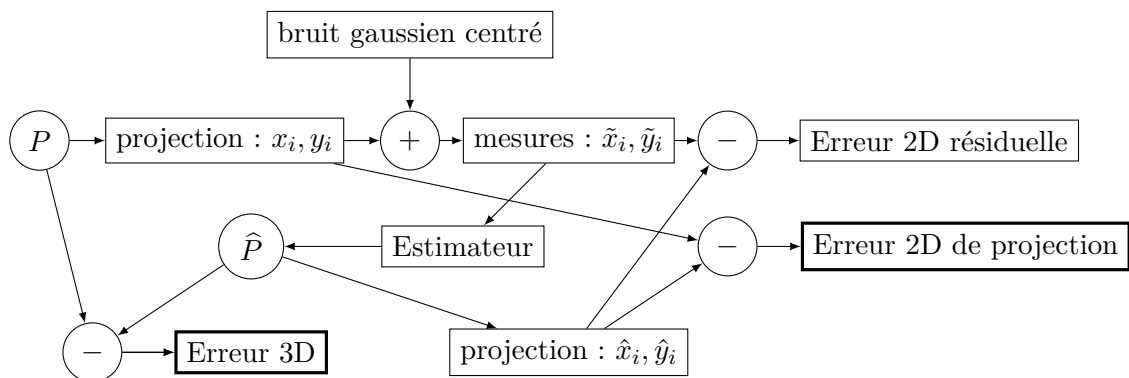


FIGURE 5.4 – Définitions des erreurs pour la triangulation du point. Dans les prochaines figures, nous étudions l'erreur 2D de projection, et l'erreur 3D de reconstruction qui sont matérialisées dans les cases en gras.

Comme on peut le voir, il y a deux erreurs 2D possibles, nous choisissons ici de com-

2. La médiane statistique, nous le rappelons, est la valeur centrale d'une liste ordonnée de valeurs. Elle est principalement utilisée comme indicateur pour les distributions asymétriques et pour sa robustesse aux valeurs aberrantes.

parer la projection des points réels avec la projection des points estimés.

La figure 5.5 présente l'évolution de l'erreur médiane de reconstruction pour deux vues en fonction de l'écart-type du bruit. Comme dans [Hartley and Sturm, 1997], l'erreur pour la méthode du point "milieu" croît quasi-linéairement en fonction de l'écart-type. Les erreurs obtenues dans notre simulation sont raisonnables.

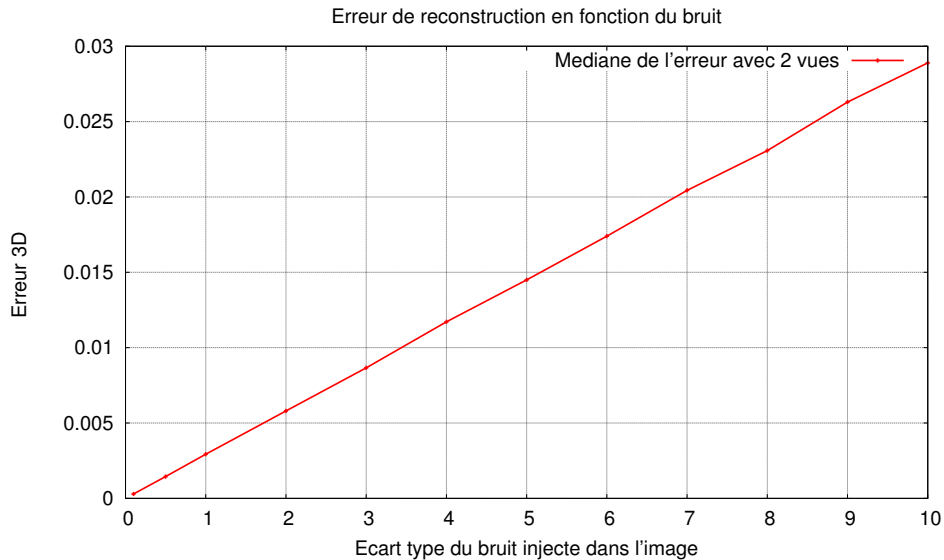


FIGURE 5.5 – Erreur médiane de reconstruction pour deux vues.

La figure 5.6 présente l'évolution de l'erreur médiane de reprojection en fonction de l'écart-type du bruit. Ici encore, l'erreur est une fonction linéaire du niveau de bruit et les valeurs atteintes sont acceptables.

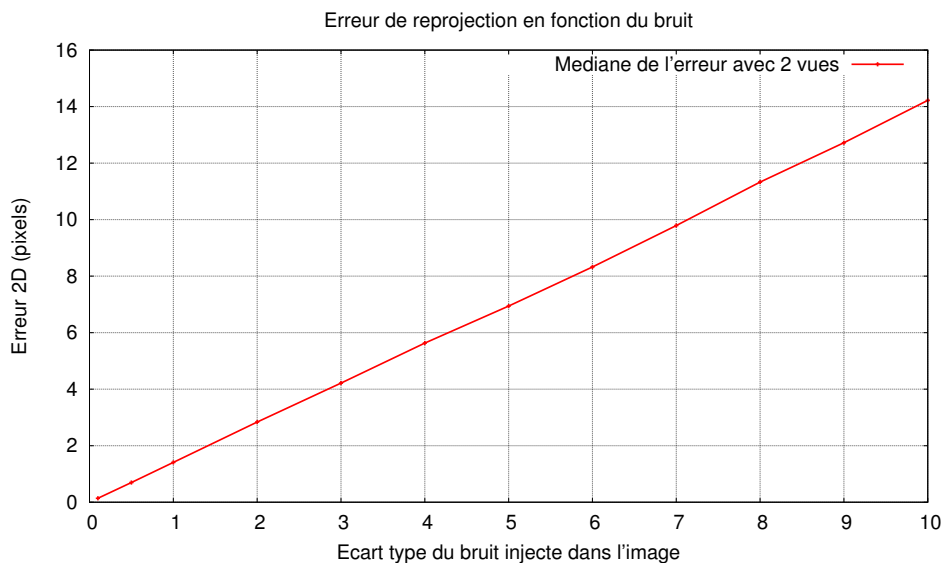


FIGURE 5.6 – Erreur médiane de reprojection pour deux vues.

Nous avons étudié l'évolution des erreurs médianes de reprojection et de reconstruction en fonction du nombre d'images. La figure 5.7 présente l'évolution de l'erreur médiane de reconstruction pour plusieurs niveaux de bruit en fonction du nombre d'images utilisées.

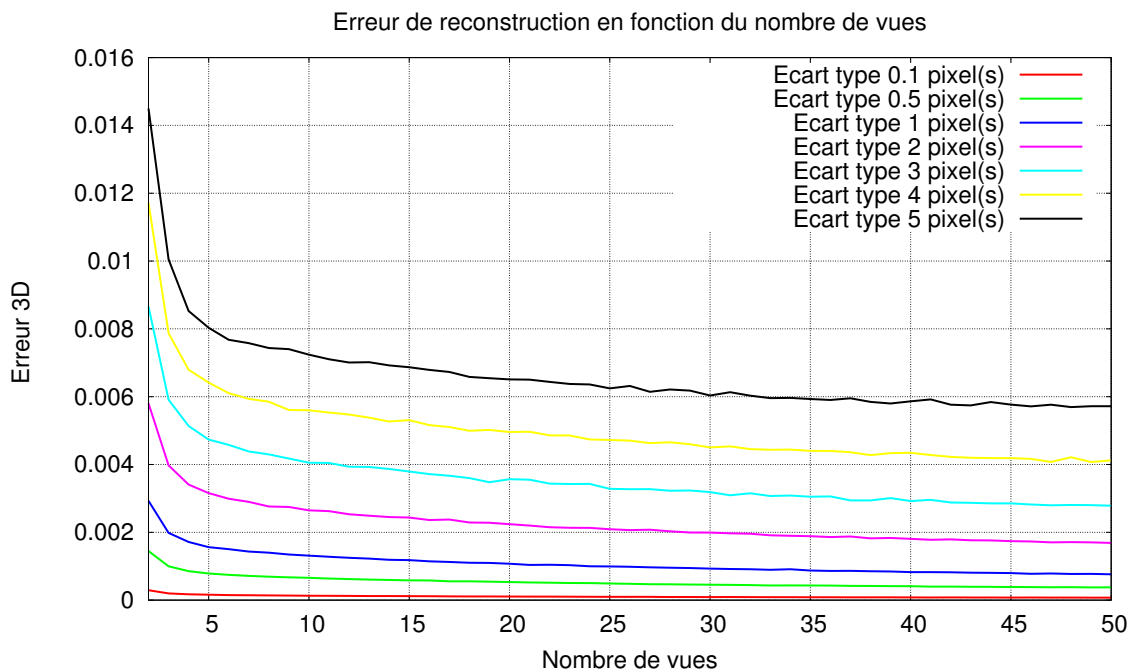


FIGURE 5.7 – Erreur médiane de reconstruction en fonction du nombre de vues.

Cette erreur décroît en fonction du nombre de vues, et tend vers une valeur inférieure à la moitié de l’erreur obtenue pour deux vues. Mais pour les niveaux de bruit très grands, d’écart type supérieur à 7 pixels, non affichés sur la figure 5.7, l’erreur de reconstruction peut même augmenter avec le nombre d’images alors que l’erreur de reprojection continue à diminuer. Une raison qui pourrait expliquer ce comportement, c’est que plus le nombre de vues est grand, plus l’écart entre deux positions de la caméra est faible et donc le rapport signal/bruit diminue et devient insuffisant lorsque le niveau d’erreur est élevé. Nous constatons, comme l’ont fait remarquer Hartley et Sturm [Hartley and Sturm, 1997], que les erreurs 2D et 3D ne sont pas forcément corrélées.

La figure 5.8 présente l’évolution de l’erreur médiane de reprojection pour plusieurs niveaux de bruit en fonction du nombre d’images utilisées. Comme pour la reconstruction en 3D, l’erreur 2D diminue en fonction du nombre d’images. Nous avons pu vérifier que pour cette simulation, cette erreur E_{2d} se comporte en fonction de b , l’écart type du bruit ajouté sur les mesures et de n le nombre d’images, suivant l’équation :

$$E_{2d} \approx b \sqrt{\frac{2}{n-1}} \quad (5.17)$$

Nous avons fait varier les positions des caméras ou les dimensions de la cible (cf. figure 5.3), mais nous obtenons sensiblement les mêmes résultats. Lorsque $n = 2$, pour deux vues, en faisant varier le bruit, ce modèle se superpose à l’erreur médiane de reconstruction obtenue comme le montre la figure 5.9.

Dans la figure 5.10, nous comparons les erreurs médianes de reprojection obtenues en fonction du nombre d’images pour différents niveaux de bruit, et le modèle de l’équation (5.17). Ce modèle prédit correctement les erreurs obtenues.

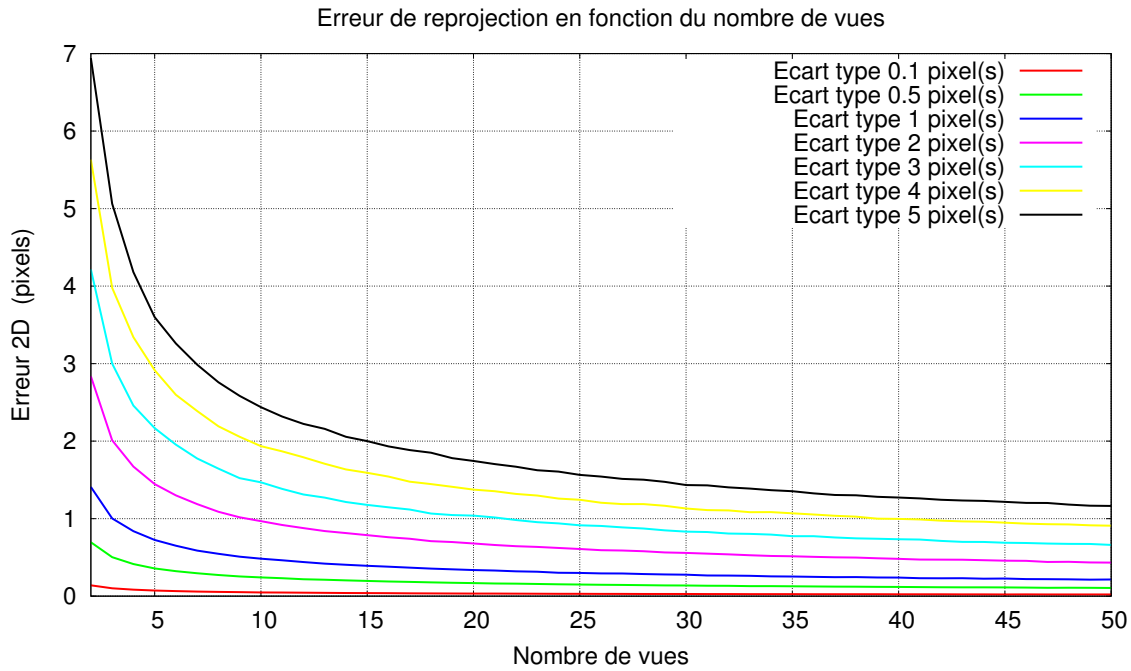


FIGURE 5.8 – Erreur médiane de reprojection en fonction du nombre de vues.

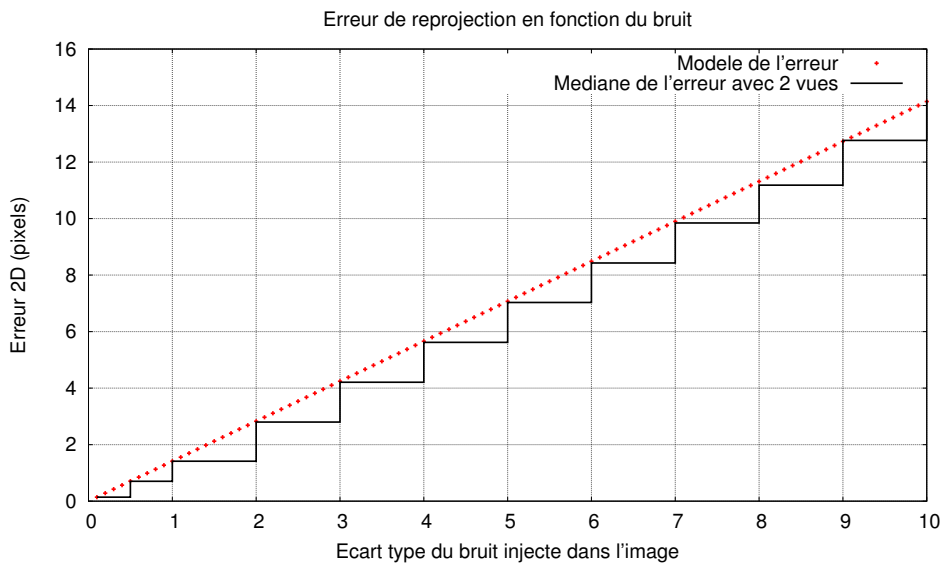


FIGURE 5.9 – Erreur médiane de reprojection pour deux vues. Comparaison avec le modèle $E_{2d} = b\sqrt{2}$.

L'erreur médiane de reprojection converge donc vers zéro en fonction du nombre d'images. Nous avons obtenu des résultats équivalents en utilisant la moyenne au lieu de la médiane. La méthode du "point milieu" généralisée à n vues s'avère donc une méthode précise, simple et rapide. Au vu de ces performances, cette méthode convient pour la reconstruction des indices visuels dans le cadre de notre tâche référencée vision.

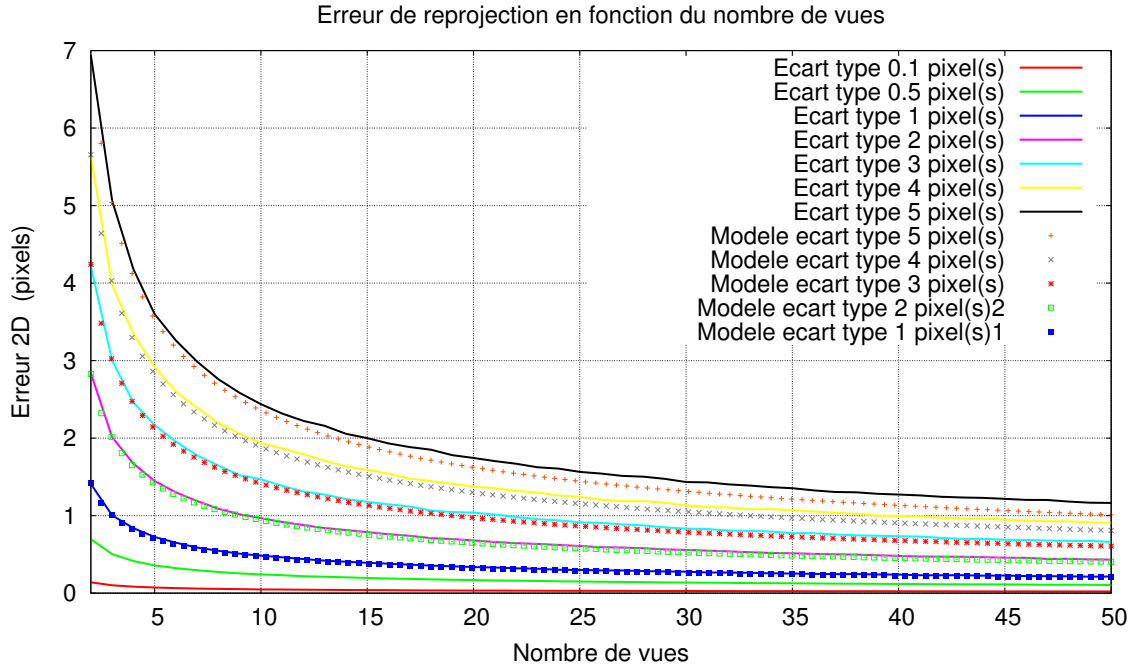


FIGURE 5.10 – Erreur médiane de reprojection en fonction du nombre de vues. Comparaison avec notre modèle de l’erreur.

5.5 Extension de la méthode du point milieu à d’autres primitives.

Nous considérons maintenant le cas des primitives de type droite et cylindre qui sont au cœur de la réalisation de notre tâche de manipulation coordonnée.

5.5.1 Cas des droites

Nous cherchons ici à estimer les paramètres d’une droite 3D à partir d’images fournies par une caméra. Nous avons choisi d’adapter le raisonnement du point milieu décrit plus haut. Pour cela il est nécessaire de choisir une représentation de la droite. Différentes solutions sont proposées dans la littérature : intersection de deux plans, paire de points, coordonnées de Plucker, couple constitué du vecteur directeur de la droite et d’un point lui appartenant [Chaumette, 1990, Chaumette and Rives, 1990, Espiau et al., 1992]. Nous avons retenu ce dernier paramétrage parce qu’il nous permet d’exprimer plus facilement nos critères.

5.5.1.1 Méthode développée

Considérons maintenant la figure 5.11. Comme précédemment, une caméra en mouvement observe une droite Δ . On note C_i les positions successives de son centre optique. Pour chacune de ces positions, on perçoit dans l’image la droite projetée Δ_{im} . Celle-ci est définie en utilisant la représentation polaire (ρ, θ) déjà rencontrée dans la section 3.4 (cf. figure 3.13). Comme indiqué plus haut, la droite 3D Δ est représentée par le couple (P, U) où U est un vecteur directeur de la droite et P un point lui appartenant. Cette représentation étant non minimale, deux contraintes sont imposées aux paramètres :

- le vecteur U est pris unitaire

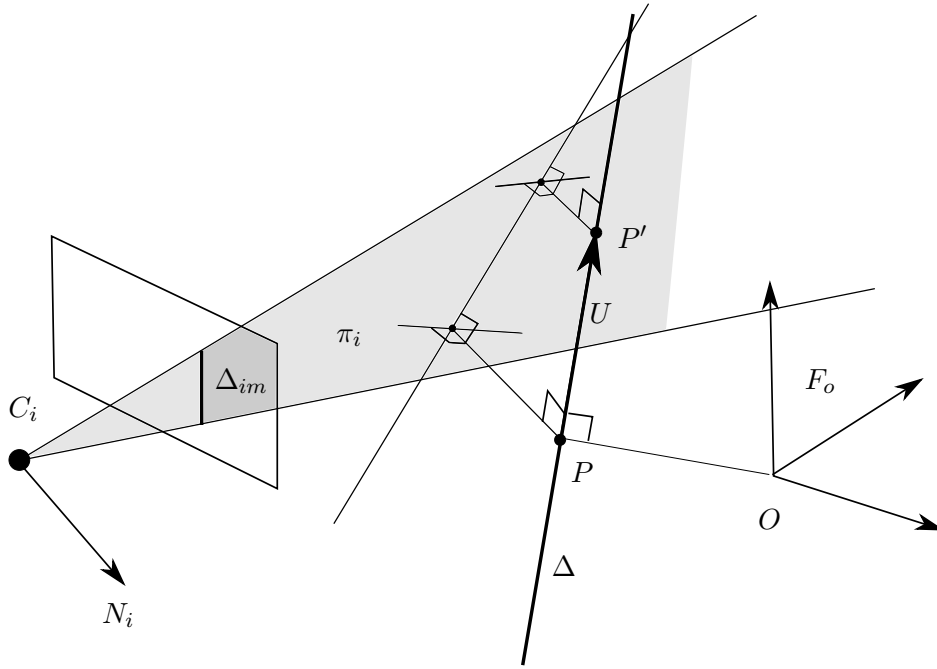


FIGURE 5.11 – Erreur géométrique dans le cas de la droite dans l'espace.

- Le point P est défini par le point le plus proche du centre du repère F_o , F_o désignant toujours le repère lié à l'objet en mouvement.

Il vient donc :

$$\bar{P}^T \cdot \bar{U} = 0 \text{ et } \bar{U}^T \cdot \bar{U} = 1$$

où \bar{P} et \bar{U} sont respectivement les coordonnées de P et les composantes de U dans F_o . Notre objectif sera donc de reconstruire ces grandeurs à partir des indices visuels fournis par la caméra pour chacune de ses positions C_i ($i = 1, 2, \dots, n$). Les paramètres à estimer sont donc encore une fois constants, indépendants du mouvement de l'objet. Ensuite, comme précédemment, ayant à disposition les différents modèles et matrices de passage, il sera possible de déduire les informations utiles pour le calcul de la matrice d'interaction et des indices visuels.

Nous cherchons d'abord à exprimer le critère à minimiser. Considérons la $i^{\text{ème}}$ position de la caméra C_i et notons N_i le vecteur unitaire normal au plan π_i formé par Δ_{im} et le point C_i (plan grisé sur la figure 5.11). Les composantes de N_i dans le repère F_i , lié à la caméra en la position C_i peuvent être déduits des indices visuels mesurés (ρ_i, θ_i) comme suit :

$${}^i N_i = \frac{1}{\sqrt{f^2 + \rho_i^2}} \begin{bmatrix} f \cos \theta_i & f \sin \theta_i & -\rho_i \end{bmatrix}^T \quad (5.18)$$

où f est la focale de la caméra.

Nous voulons, comme pour le point, exprimer une erreur géométrique dans l'espace euclidien en utilisant notamment les orthogonalités. Comme l'on peut voir sur la figure 5.11, la droite Δ appartient au plan π_i , si deux points distincts de cette droite lui appartiennent. Choisissons deux points de Δ : P et $P' = \bar{P} + \bar{U}$. Ces points peuvent être exprimés aisément en fonction des paramètres de la droite³. Nous définissons notre erreur géométrique e_i^d comme la somme des carrés des distances des points P et P' au plan π_i , soit :

$$e_i^d = (\bar{P}' - \bar{C}_i)^T \bar{N}_i \bar{N}_i^T (\bar{P}' - \bar{C}_i) + (\bar{P} - \bar{C}_i)^T \bar{N}_i \bar{N}_i^T (\bar{P} - \bar{C}_i) \quad (5.19)$$

3. Le point P' peut être aussi défini comme $\bar{P}' = \bar{P} + \alpha \bar{U}$ avec $\alpha \in \mathbb{R}^*$

avec \bar{N}_i la valeur de N_i exprimée dans le repère F_o . Cette expression s'obtient en additionnant la distance entre P et sa projection sur le plan π_i et la distance entre P' et sa projection sur le même plan. En remplaçant P' par son expression, nous obtenons :

$$e_i^d = (\bar{P}' + \bar{U} - \bar{C}_i)^T \bar{N}_i \bar{N}_i^T (\bar{P}' + \bar{U} - \bar{C}_i) + (\bar{P} - \bar{C}_i)^T \bar{N}_i \bar{N}_i^T (\bar{P} - \bar{C}_i) \quad (5.20)$$

En définissant ces variables :

$$\Theta_i = \bar{N}_i \bar{N}_i^T \quad \xi_i = \Theta_i \bar{C}_i \quad \epsilon_i = \bar{C}_i^T \Theta_i \bar{C}_i \quad (5.21)$$

et en développant, e_i^d s'écrit :

$$e_i^d = 2\bar{P}^T \Theta_i \bar{P} + 2\bar{P}^T \Theta_i \bar{U} - 4\bar{P}^T \xi_i + \bar{U}^T \Theta_i \bar{U} - 2\bar{U}^T \xi_i + 2\epsilon_i \quad (5.22)$$

Le critère à minimiser sur n images a pour expression :

$$E_n^d = \sum_{i=1}^n e_i^d$$

$$E_n^d = 2\bar{P}^T \left[\sum_{i=1}^n \Theta_i \right] \bar{P} + 2\bar{P}^T \left[\sum_{i=1}^n \Theta_i \right] \bar{U} - 4\bar{P}^T \left[\sum_{i=1}^n \xi_i \right] + \bar{U}^T \left[\sum_{i=1}^n \Theta_i \right] \bar{U} - 2\bar{U}^T \left[\sum_{i=1}^n \xi_i \right] + 2 \sum_{i=1}^n \epsilon_i \quad (5.23)$$

Pour simplifier l'écriture nous définissons ces nouvelles variables, qui sont dans l'ordre une matrice de dimensions 3×3 , un vecteur 3×1 et un scalaire :

$$\Psi_n = \sum_{i=1}^n \Theta_i, \quad \delta_n = \sum_{i=1}^n [\xi_i], \quad \mu_n = \sum_{i=1}^n [\epsilon_i], \quad (5.24)$$

Le critère à minimiser s'écrit alors :

$$E_n^d = 2\bar{P}^T \Psi_n \bar{P} + \bar{U}^T \Psi_n \bar{U} + 2\bar{P}^T \Psi_n \bar{U} - 4\bar{P}^T \delta_n - 2\bar{U}^T \delta_n + 2\mu_n \quad (5.25)$$

Le critère obtenu est donc quadratique et dépend de \bar{P} et de \bar{U} . Dérivons ce critère par rapport aux paramètres et plaçons-nous au minimum. Il vient :

$$\Psi_n \bar{P} - \delta_n = 0 \quad (5.26)$$

$$\Psi_n \bar{U} = 0 \quad (5.27)$$

Analysons maintenant ces deux relations. De l'équation (5.26) nous déduisons l'estimateur \hat{P}_n de \bar{P} :

$$\hat{P}_n = [\Psi_n]^+ \delta_n \quad (5.28)$$

Étudions le rang de la matrice Ψ_n . De dimensions 3×3 , elle est obtenue en additionnant des matrices $\Theta_i = \bar{N}_i \bar{N}_i^T$. Ces matrices sont symétriques, semi-définies positives et de rang 1⁴. Cela implique que Ψ_n est de rang supérieur ou égal à 1 de par sa construction. Différents cas peuvent se produire :

- Dans le cas dégénéré où tous les plans π_i sont confondus, en absence de bruit, le rang de la matrice Ψ_n est égal à 1 et il n'y a aucun moyen de déterminer la droite [Ma et al., 2001].
- Dans tous les autres cas, le rang est égal à 2, et la droite peut être complètement déterminée.

4. La multiplication à gauche par Θ_i de tout vecteur orthogonal à N_i donne le vecteur nul.

\hat{P}_n , calculé avec la pseudo-inverse est la solution de norme minimale. Nous obtenons donc le point de la droite le plus proche de l'origine du repère F_o . L'équation (5.27) quant à elle montre que \bar{U} appartient au noyau de Ψ_n . \bar{U} est donc colinéaire au vecteur propre correspondant à la seule valeur propre nulle de Ψ_n . Dans la pratique, aucune valeur propre ne sera strictement nulle, du fait des bruits et des erreurs. Toutefois, nous choisirons la valeur propre la plus faible, qui devrait être négligeable par rapport aux deux autres. Ainsi l'estimateur \hat{U} de \bar{U} est défini comme le *vecteur propre associé à la plus petite valeur propre* de Ψ_n .

Remarques :

- Nous pouvons, comme dans le cas du point, mettre à jour notre estimation avec l'image $n + 1$ à un coût constant. Notre estimateur récursif fonctionne donc ainsi :

$$\Theta_{n+1} = \bar{N}_{n+1}\bar{N}_{n+1}^T \quad (5.29)$$

$$\Psi_{n+1} = \Psi_n + \Theta_{n+1} \quad (5.30)$$

$$\delta_{n+1} = \delta_n + \Theta_{n+1}\bar{C}_{n+1} \quad (5.31)$$

$$\hat{P}_{n+1} = [\Psi_{n+1}]^+ \delta_{n+1} \quad (5.32)$$

et calcul de \hat{U}_{n+1} en fonction de Ψ_{n+1} .

- Connaissant les paramètres de la droite, nous pouvons reconstruire aisément les indices visuels dans l'image. Nous donnons ci-après un bref aperçu de la méthode. La normale du plan formé par le centre optique C_i et la droite Δ estimée est donnée par : $\widehat{N}_i = \overrightarrow{C_i\hat{P}} \wedge U$. De là, nous pouvons écrire l'équation du plan dans le repère de la caméra :

$$a_i x + b_i y + c_i z = 0$$

avec (a_i, b_i, c_i) les composantes de \widehat{N}_i dans le repère F_i de la caméra. La droite sur l'image est l'intersection de ce plan avec le plan image $z = f$ (Figure 3.12), on peut donc écrire

$$a_i x + b_i y + c_i f = 0$$

Enfin les indices visuels $(\hat{\rho}_i, \hat{\theta}_i)$ décrivant l'image de la droite peuvent être déduits [Espiau et al., 1992] :

$$\hat{\rho}_i = -\frac{c_i f}{\sqrt{a_i^2 + b_i^2}}$$

$$\hat{\theta}_i = \arctan(b_i, a_i)$$

L'expression de la matrice d'interaction est fournie dans l'annexe A.

- Cas des segments de droite : Il est à noter que, pour estimer la structure d'un segment de droite, il suffit d'estimer séparément les coordonnées de ses extrémités à l'aide de la méthode décrite dans la section 5.4.1.

5.5.1.2 Résultats de simulation

Nous avons testé notre estimateur de la droite en simulation. Nous avons défini une tâche référencée vision en utilisant les indices visuels de trois droites coplanaires et perpendiculaires pouvant former un carré de 0.2 m de côté, semblable à celui de la figure 5.3. Les composantes de la tâche sont dans l'ordre $(\rho_1, \theta_1, \rho_2, \theta_2, \rho_3, \theta_3)$ correspondant aux indices visuels des trois droites. À l'instant initial, la caméra fait face au carré, et est placée à 1.6 m de ce carré (le repère F_1 sur la figure 5.12). Une rotation de 0.5 radians

a été réalisée autour de l'axe optique de la caméra. Dans la pose désirée, la caméra doit se placer à 0.6 m de la cible et compenser la rotation (le repère F_f sur la figure 5.12). Le mouvement que doit effectuer la caméra, pendant la tâche, s'apparente donc à un vissage. Les paramètres intrinsèques sont les mêmes que pour le cas du point.

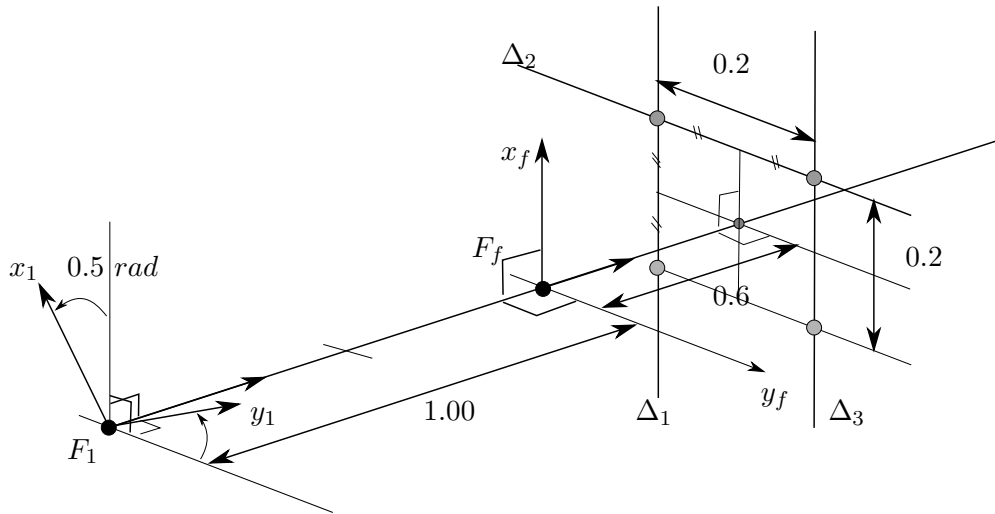


FIGURE 5.12 – Conditions de simulations pour la droite

Nous avons ajouté un bruit gaussien d'écart-type de 2 pixels aux mesures de ρ dans l'image et un bruit gaussien d'écart-type de 1 degré sur les mesures de θ dans l'image. Un gain $\lambda = 0.1$ a été imposé au correcteur proportionnel.

Pendant la réalisation de la tâche d'asservissement visuel, nous avons utilisé les indices visuels mesurés (bruités) pour alimenter notre estimateur de manière récursive. Nous présentons sur les figures 5.13 et 5.14, l'évolution des paramètres estimés de l'une des trois droites. Les coordonnées réelles de P dans le repère F_o de la cible sont $[0\ 0.1\ 0]^T$, tandis que les composantes réelles de U sont $[0\ 0\ 1]^T$.

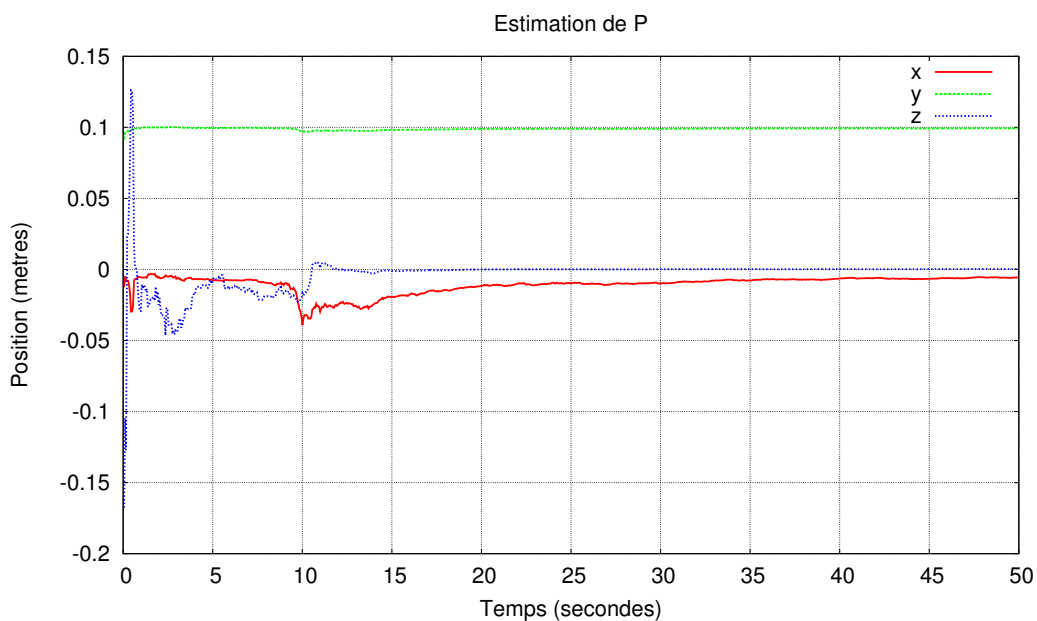


FIGURE 5.13 – Estimation du paramètre P de la droite en fonction du temps. Les valeurs réelles sont $x = 0$, $y = 0.1$ et $z = 0$.

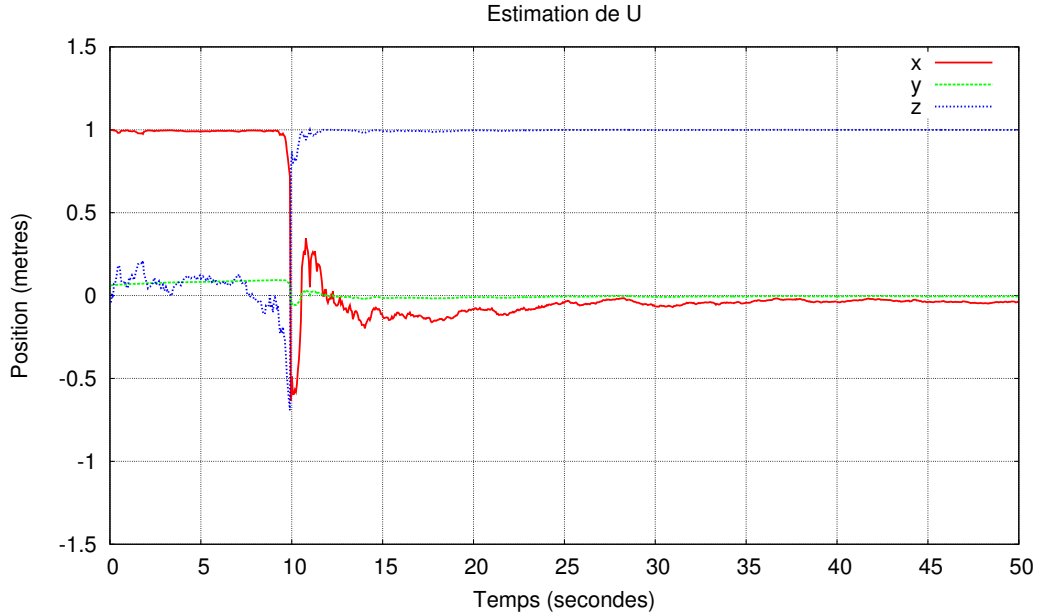


FIGURE 5.14 – Estimation du paramètre U de la droite en fonction du temps. Les valeurs réelles sont $x = 0$, $y = 0$ et $z = 1$.

Nous constatons que les estimateurs de P et de U convergent vers des valeurs constantes proches de celles désirée. Pendant les 10 premières secondes, le système n'a pas assez d'information sur l'orientation de la droite, mais il converge déjà vers la valeur réelle de P . Dès qu'il reçoit assez d'information sur l'orientation, l'estimateur redirige rapidement le vecteur U dans la bonne direction. D'où la discontinuité observée sur l'évolution de \hat{U} .

Nous reproduisons sur la figure 5.15, l'évolution des coordonnées de la caméra dans le repère F_0 de la cible pour illustrer le déplacement effectué par la caméra pendant cette estimation.

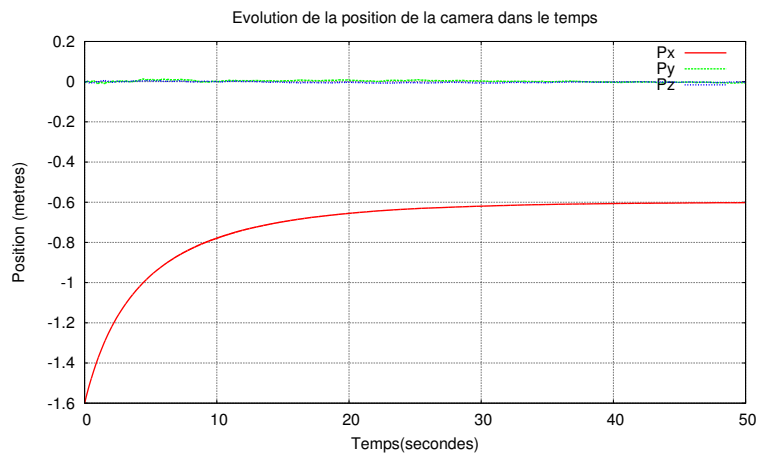


FIGURE 5.15 – Évolution des coordonnées du centre optique de la caméra en fonction du temps.

La figure 5.16 montre l'évolution des composantes de la fonction de tâche définie pour réaliser l'asservissement visuel sur les trois droites. Elle permet également de visualiser le niveau de bruit ajouté aux indices visuels. En dépit du bruit ajouté, la tâche est réalisée.

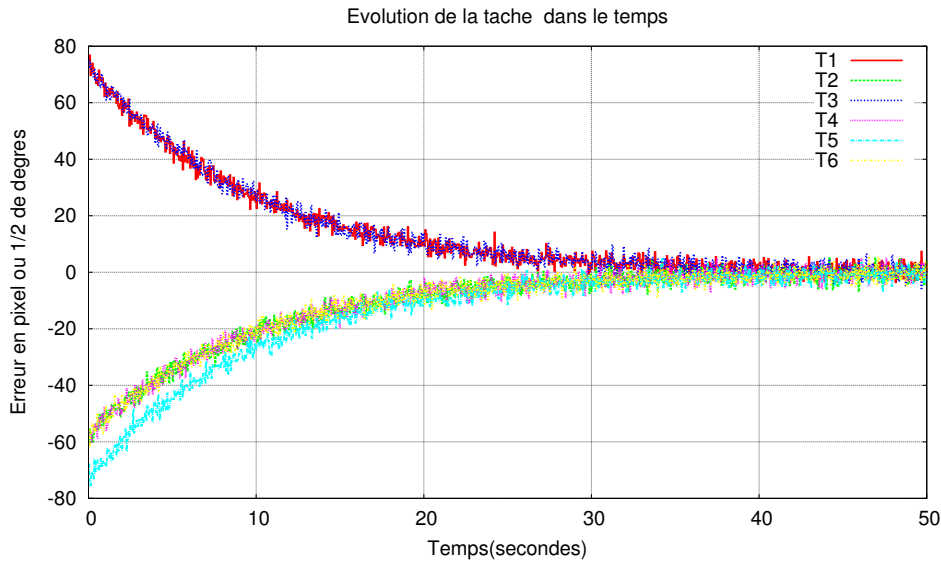


FIGURE 5.16 – Évolution des composantes de la tâche référencée vision sur trois droites.

Pour montrer la qualité obtenue de la reconstruction des indices visuels dans l'image, nous avons rejoué la simulation. Cette fois-ci après 15 secondes, nous utilisons dans le calcul de la tâche et de la commande, non pas les indices visuels bruités, mais leur reconstruction à partir de nos estimateurs. La figure 5.17 montre l'évolution des composantes de la tâche d'asservissement visuel dans ces conditions.

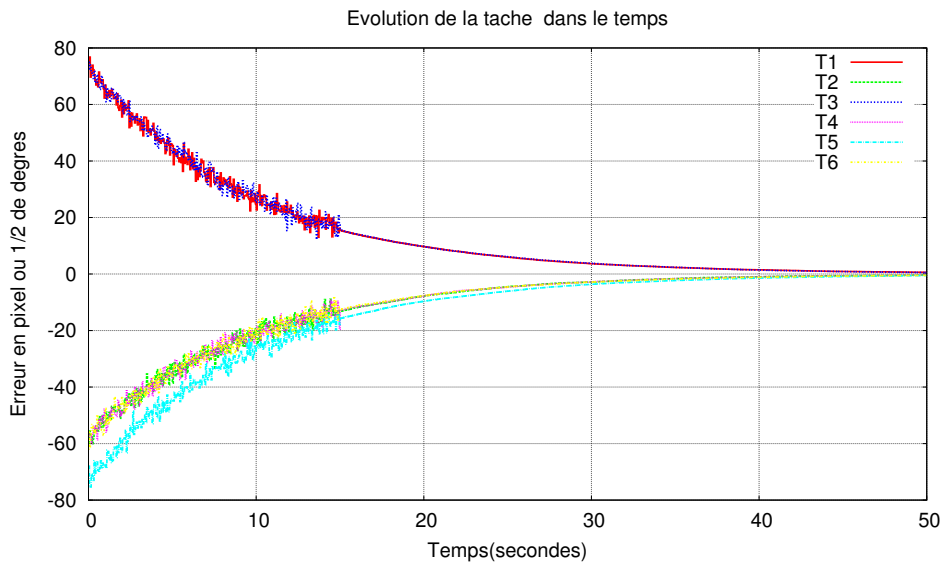


FIGURE 5.17 – Évolution des composantes de la tâche référencée vision sur trois droites. Reconstruction des indices visuels après les 15 premières secondes.

La transition est continue entre les deux phases, et un effet lissant est observé dans les composantes de la fonction de tâche. Ce qui montre que l'erreur de reprojection est inférieure à l'écart-type du bruit ajouté. La tâche référencée vision est réalisée de la même manière, et le changement dans la trajectoire de la caméra est difficilement décelable.

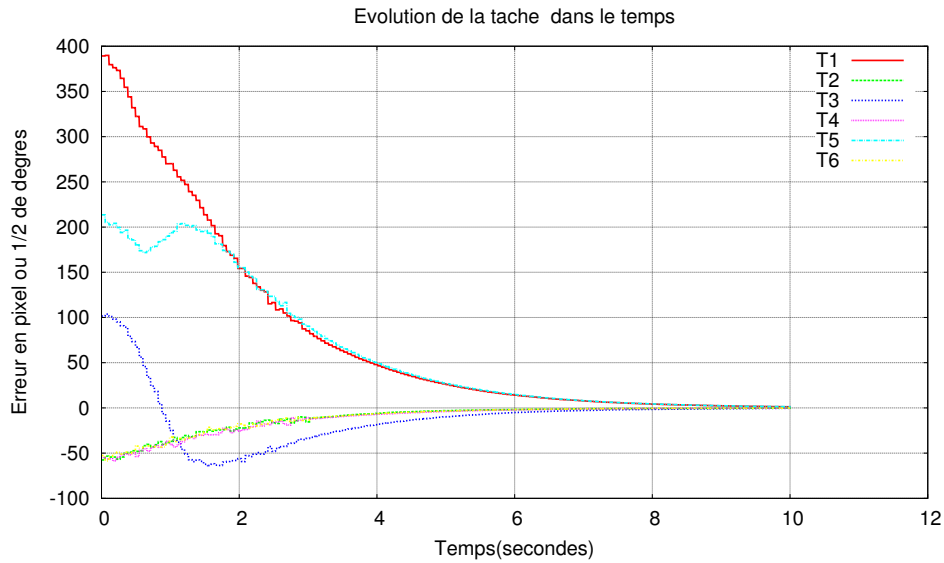


FIGURE 5.18 – Évolution des composantes de la tâche référencée vision sur trois droites. Reconstruction des indices visuels après les 3 premières secondes.

Nous avons modifié la pose initiale de la caméra en ajoutant des rotations selon les trois axes, ainsi qu'en modifiant la position. Nous avons augmenté le gain de la loi de commande $\lambda = 0.8$ et gardé le même niveau de bruit. Cette fois-ci, nous utilisons les indices visuels reconstruits pour nourrir la tâche après seulement 3 secondes. La figure 5.18 montre l'évolution des composantes de la fonction de tâche. Elles convergent toutes vers zéro.

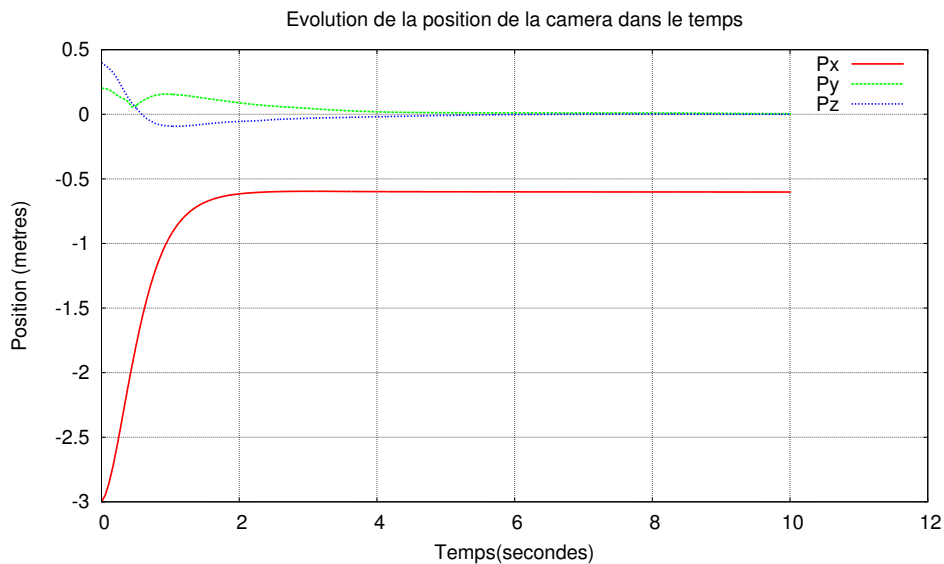


FIGURE 5.19 – Évolution de la position du centre optique de la caméra en fonction du temps. Valeurs désirées $[0 \ 0 \ -0.6]$

Les figures 5.19 et 5.20 présentent l'évolution de la position et de l'orientation de la caméra pendant la réalisation de la tâche. Elles montrent que la caméra atteint la pose souhaitée en utilisant les indices visuels reconstruits. Ce qui montre la qualité de notre

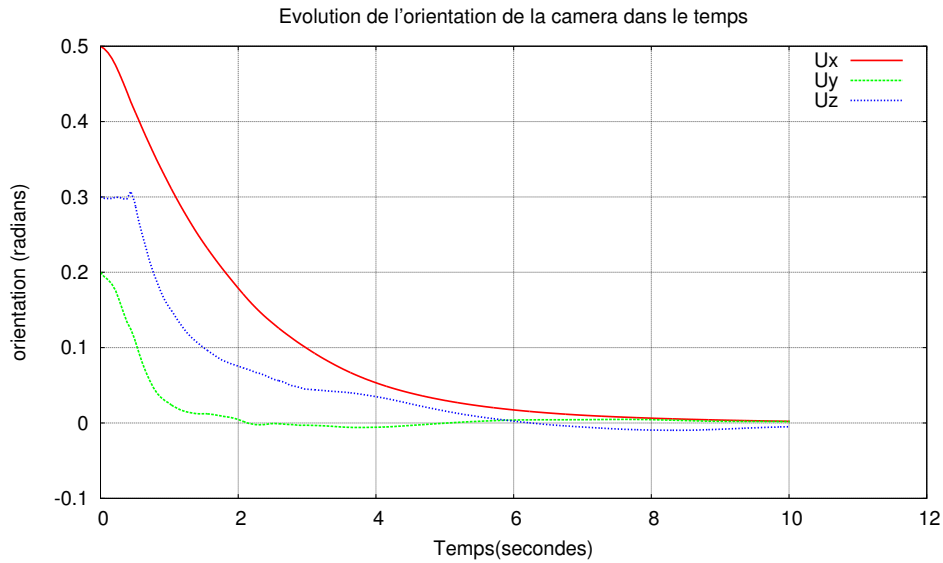


FIGURE 5.20 – Évolution de l'orientation de la caméra en fonction du temps. Valeurs désirées $[0\ 0\ 0]$

reprojection sur l'image.

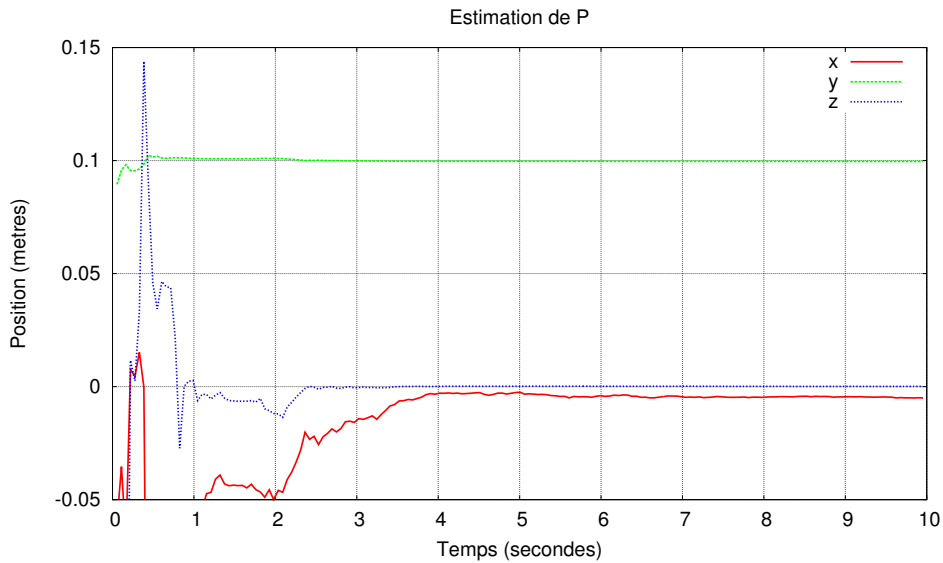


FIGURE 5.21 – Estimation du paramètre P de la droite en fonction du temps. Les valeurs réelles sont $x = 0$, $y = 0.1$ et $z = 0$.

Enfin les figures 5.21 et 5.22 illustrent la convergence rapide de nos estimateurs de \hat{P} et de \hat{U} suite au mouvement effectué par la caméra. Ils convergent vers les valeurs réelles de P et de U . Les valeurs estimées ont été utilisées pour le calcul de la matrice d'interaction et de la loi de commande. Grâce à la précision de notre estimation et à la robustesse de l'asservissement visuel, la tâche a été réalisée. Notre estimateur peut donc être utilisé, pendant une tâche d'asservissement visuel pour reconstruire les indices visuels en cas d'occultation partielle ou totale.

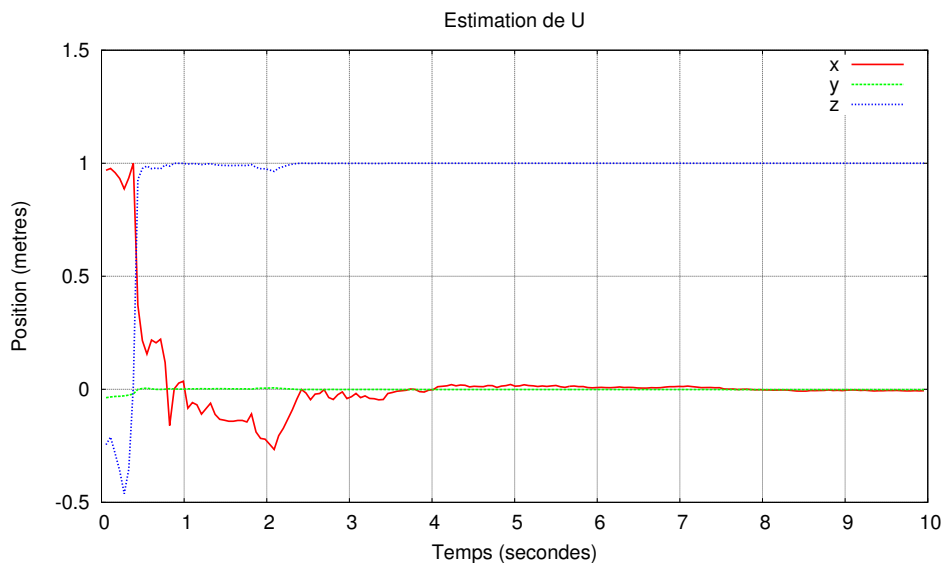


FIGURE 5.22 – Estimation du paramètre U de la droite en fonction du temps. Les valeurs réelles sont $x = 0$, $y = 0$ et $z = 1$.

5.5.2 Estimation du cylindre

Nous considérons ici le cas du cylindre. Dans un premier temps, nous traiterons le cas d'un cylindre de longueur infinie. Notre problème d'estimation se réduira alors à la détermination de son axe et de son rayon. Il s'agit de l'un des cas les plus couramment traités dans la littérature [Chaumette, 1990, Spica et al., 2014c]. Mais nous nous intéresserons également à la reconstruction d'un cylindre de longueur finie comme dans [Miglani et al., 2013]. Il s'agira alors de déterminer complètement la longueur et la position de ce cylindre. Cette dernière reconstruction nous sera utile pour la tâche de fermeture de stylo décrite dans la section 3.4.

Remarque : Dans cette section traitant des cylindres, nous introduirons de nombreuses variables, mais quel que soit le symbole latin ou grec utilisé, elles ne feront référence à aucune variable définie dans les sections précédentes.

5.5.2.1 Cas du cylindre de longueur infinie

Nous considérons un objet cylindrique en mouvement auquel est lié le repère F_0 . Cet objet est supposé de longueur infinie. Dans l'image, il sera donc caractérisé par deux lignes droites [Chaumette, 1990, Chaumette, 1994]. Les indices visuels seront définis de la même manière que dans la section 5.5.1 par les paramètres ρ et θ comme suit :

$$s = [\rho_1 \quad \theta_1 \quad \rho_2 \quad \theta_2]^T$$

Pour représenter le cylindre dans l'espace, nous avons choisi une représentation classique, utilisée notamment dans [Chaumette, 1994, Spica et al., 2014a]. Celle-ci consiste à décrire l'objet à l'aide du point P de son axe le plus proche de l'origine de F_0 , du vecteur directeur U de son axe, et du rayon r (cf. figure 5.23). Nous aurons donc, outre le rayon, à reconstruire les mêmes variables que dans le cas de la ligne droite. Comme précédemment, nous estimerons les paramètres dans le repère F_0 à partir des indices visuels fournis par

la caméra. Rappelons que la caméra se déplace et perçoit l'objet à partir de différentes positions successives de son centre optique notées C_i .

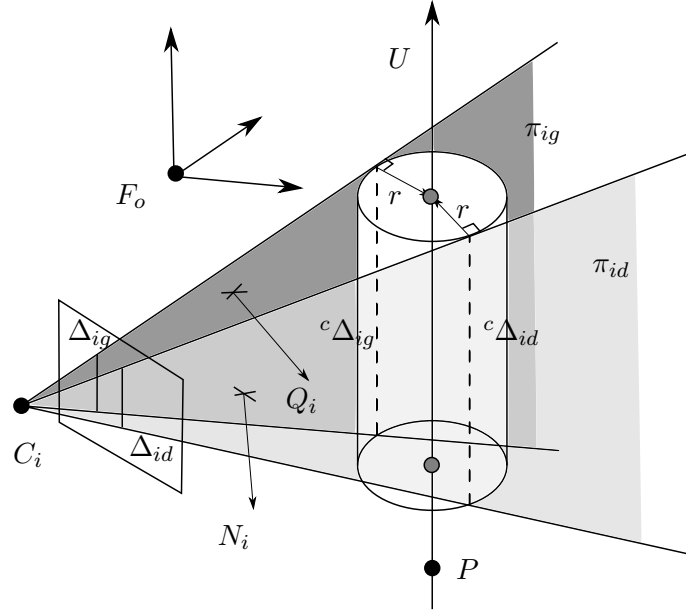


FIGURE 5.23 – Plans tangents au cylindre passant par le centre optique.

Considérons la figure 5.23. Les deux droites $\Delta_{ig}(\rho_1, \theta_1)$ et $\Delta_{id}(\rho_2, \theta_2)$ qui caractérisent l'image du cylindre définissent avec le centre optique de la caméra C_i deux plans π_{ig} et π_{id} de normale unitaire Q_i et N_i . Les composantes des normales à ces deux plans dans le repère F_i lié à la caméra peuvent être déduites des indices visuels en utilisant l'équation (5.18). Ces deux plans sont tangents au cylindre en deux lignes de contact ${}^c\Delta_{ig}$ et ${}^c\Delta_{id}$ illustrées en pointillés sur la figure 5.23. Les deux plans sont perpendiculaires, sur les lignes de contact, aux rayons issus de l'axe du cylindre et passant par ces lignes. Ces rayons sont donc colinéaires sur les lignes de contact aux normales des plans considérés. Par conséquent les points d'une ligne de contact sont la projection orthogonale au plan tangent de l'axe du cylindre. La distance entre une ligne de contact et l'axe est égale au rayon r du cylindre. Nous pouvons donc écrire que le vecteur défini entre un point P sur l'axe du cylindre et sa projection sur un des plans est égale à la normale de ce plan multipliée par le rayon (au sens près). En d'autres termes, pour le point P et le plan π_{id} de normale N_i , nous avons, exprimé dans le repère F_o lié à l'objet, cette relation :

$$\bar{N}_i \bar{N}_i^T (\bar{P} - \bar{C}_i) = a_i r \bar{N}_i$$

où a_i est un scalaire égal à -1 ou 1 permettant de tenir compte du sens de la normale N_i . Nous écrivons la même relation entre deux points P et $\bar{P}' = \bar{P} + \bar{U}$ de l'axe du cylindre et les deux plans π_{ig} et π_{id} , ce qui nous donne :

$$\bar{N}_i \bar{N}_i^T (\bar{P} - \bar{C}_i) = a_i r \bar{N}_i \quad (5.33)$$

$$\bar{N}_i \bar{N}_i^T (\bar{P} + \bar{U} - \bar{C}_i) = a_i r \bar{N}_i \quad (5.34)$$

$$Q_i Q_i^T (\bar{P} - \bar{C}_i) = b_i r Q_i \quad (5.35)$$

$$Q_i Q_i^T (\bar{P} + \bar{U} - \bar{C}_i) = b_i r Q_i \quad (5.36)$$

où b_i est aussi un scalaire valant -1 ou 1 selon le sens de la normale Q_i ⁵.

5. Sur la figure 5.23, nous aurions $a_i = -1$, $b_i = 1$.

Les quatre égalités précédentes (5.33) à (5.36) sont vraies en absence de bruit ou d'erreurs. Dans le cas contraire, les termes de gauche et de droite seront différents. Nous choisissons donc de définir notre erreur géométrique comme la somme du carré des erreurs entre les termes de gauche et de droite de ces quatre égalités. Pour simplifier l'écriture nous définissons :

$$\phi_i = \bar{N}_i \bar{N}_i^T \quad (5.37)$$

$$\psi_i = \bar{Q}_i \bar{Q}_i^T \quad (5.38)$$

En remarquant que $\bar{N}_i \bar{N}_i^T \bar{N}_i = \bar{N}_i$ et que $\bar{Q}_i \bar{Q}_i^T \bar{Q}_i = \bar{Q}_i$, notre erreur géométrique peut s'écrire comme suit :

$$\begin{aligned} e_i^c &= (\bar{P} - \bar{C}_i - a_i r \bar{N}_i)^T \phi_i (\bar{P} - \bar{C}_i - a_i r \bar{N}_i) + (\bar{P} - \bar{C}_i - b_i r \bar{Q}_i)^T \psi_i (\bar{P} - \bar{C}_i - b_i r \bar{Q}_i) \\ &+ (\bar{P} + \bar{U} - \bar{C}_i - a_i r \bar{N}_i)^T \phi_i (\bar{P} + \bar{U} - \bar{C}_i - a_i r \bar{N}_i) \\ &+ (\bar{P} + \bar{U} - \bar{C}_i - b_i r \bar{Q}_i)^T \psi_i (\bar{P} + \bar{U} - \bar{C}_i - b_i r \bar{Q}_i) \end{aligned} \quad (5.39)$$

En développant, il vient :

$$\begin{aligned} e_i^c &= 2\bar{P}^T [\phi_i + \psi_i] \bar{P} + U^T [\phi_i + \psi_i] \bar{U} + 2\bar{P}^T [\phi_i + \psi_i] \bar{U} - 2(2\bar{P} + U)^T [\phi_i + \psi_i] \bar{C}_i \\ &- 2r(2\bar{P} + U)^T (a_i \bar{N}_i + b_i \bar{Q}_i) + 4r C_i^T (a_i \bar{N}_i + b_i \bar{Q}_i) \\ &+ 2C_i^T [\phi_i + \psi_i] C_i + 4r^2 \end{aligned} \quad (5.40)$$

Le critère à minimiser choisi est la somme des erreurs e_i^c sur n images. Par souci de lisibilité, nous définissons les variables suivantes :

$$\Pi_n = \sum_{i=1}^n [\phi_i + \psi_i] \quad \beta_n = \sum_{i=1}^n [\phi_i + \psi_i] \bar{C}_i \quad \sigma_n = \sum_{i=1}^n \bar{C}_i^T [\phi_i + \psi_i] \bar{C}_i \quad (5.41)$$

$$\gamma_n = \sum_{i=1}^n a_i \bar{N}_i + b_i \bar{Q}_i \quad \mu_n = \sum_{i=1}^n C_i^T (a_i \bar{N}_i + b_i \bar{Q}_i) \quad (5.42)$$

La variable Π_n est une matrice de dimension 3×3 , les variables β_n et γ_n sont des vecteurs 3×1 , σ_n et μ_n sont des scalaires. L'expression du critère devient :

$$E_n^c = \sum_{i=1}^n e_i^c \quad (5.43)$$

$$\begin{aligned} E_n^c &= 2\bar{P}^T \Pi_n \bar{P} + \bar{U}^T \Pi_n \bar{U} + 2\bar{P}^T \Pi_n \bar{U} - 2(2\bar{P} + \bar{U})^T \beta_n - 2r(2\bar{P} + \bar{U})^T \gamma_n \\ &+ 4nr^2 + 4r\mu_n + 2\sigma_n \end{aligned} \quad (5.44)$$

Le critère obtenu est quadratique et dépend de \bar{P} , de \bar{U} et de r . Dérivons ce critère par rapport aux paramètres et plaçons-nous au minimum. Il vient :

$$4\Pi_n \bar{P} + 2\Pi_n \bar{U} - 4r\gamma_n - 4\beta_n = 0 \quad (5.45)$$

$$2\Pi_n \bar{P} + 2\Pi_n \bar{U} - 2r\gamma_n - 2\beta_n = 0 \quad (5.46)$$

$$-4\gamma_n^T \bar{P} - 2\gamma_n^T \bar{U} + 8nr + 4\mu_n = 0 \quad (5.47)$$

En soustrayant à l'équation (5.45), le double de l'équation (5.46), nous obtenons :

$$\Pi_n \bar{U} = 0 \quad (5.48)$$

Nous avons rencontré un cas similaire pour la droite. La matrice Π_n de dimensions 3×3 est ici une matrice hermitienne au moins de rang 2 quel que soit n le nombre d'images. En l'absence de bruit, elle est strictement de rang 2, et possède donc une seule valeur propre nulle. Dans la pratique, aucune valeur propre ne sera identiquement nulle, mais une devrait être négligeable par rapport aux deux autres. L'estimateur \hat{U}_n du vecteur \bar{U} est défini comme le vecteur propre correspondant à la plus petite valeur propre de Π_n . \bar{U} peut donc être déterminé quel que soit le nombre d'images.

En utilisant l'égalité (5.48), les deux équations (5.45) et (5.46) deviennent linéairement dépendantes et peuvent être résumées ainsi :

$$4\Pi_n\bar{P} - 4r\gamma_n - 4\beta_n = 0 \quad (5.49)$$

A partir des équations (5.47) le rayon r peut être exprimé en fonction des autres paramètres :

$$4r = \frac{1}{n}(2\gamma_n^T\bar{P} + \gamma_n^T\bar{U} - 2\mu_n) \quad (5.50)$$

En remplaçant r dans l'équation (5.49), nous obtenons :

$$\begin{aligned} 4\Pi_n\bar{P} - \frac{1}{n}\gamma_n(2\gamma_n^T\bar{P} + \gamma_n^T\bar{U} - 2\mu_n) &= 4\beta_n \\ (4\Pi_n - \frac{2}{n}\gamma_n\gamma_n^T)\bar{P} &= 4\beta_n + \frac{1}{n}\gamma_n\gamma_n^T\bar{U} - \frac{2}{n}\gamma_n\mu_n \end{aligned}$$

Nous en déduisons l'expression de notre estimateur de \bar{P} :

$$\hat{P}_n = (\Pi_n - \frac{1}{2n}\gamma_n\gamma_n^T)^+(\beta_n + \frac{1}{n}\gamma_n\gamma_n^T\hat{U}_n - \frac{\mu_n}{2n}\gamma_n) \quad (5.51)$$

Étudions le comportement de cet estimateur. En l'absence d'erreur ou de bruit, quel que soit n , la matrice $(\Pi_n - \frac{1}{2n}\gamma_n\gamma_n^T)$ de dimensions 3×3 est une matrice hermitienne de rang 2 ayant le même noyau que Π_n . Mais deux cas peuvent se présenter :

- Lorsque nous disposons d'une seule image $n = 1$ ou d'un seul point de vue, nous ne disposons d'aucune information permettant de situer l'axe du cylindre. Et nous avons :

$$\beta_n = \Pi_n\bar{C}_n \quad \mu_n = \gamma_n^T\bar{C}_n \quad \gamma_n^T\bar{U} = 0$$

Rappelons que μ_n est un scalaire. En utilisant ces égalités dans notre estimateur nous obtenons :

$$\begin{aligned} \hat{P}_n &= (\Pi_n - \frac{1}{2n}\gamma_n\gamma_n^T)^+(\Pi_n\bar{C}_n - \frac{1}{2n}\gamma_n\mu_n) \\ \hat{P}_n &= (\Pi_n - \frac{1}{2n}\gamma_n\gamma_n^T)^+(\Pi_n\bar{C}_n - \frac{1}{2n}\gamma_n\gamma_n^T\bar{C}_n) \end{aligned}$$

En factorisant :

$$\hat{P}_n = (\Pi_n - \frac{1}{2n}\gamma_n\gamma_n^T)^+(\Pi_n - \frac{1}{2n}\gamma_n\gamma_n^T)\bar{C}_n$$

\bar{C}_n est donc projeté orthogonalement (selon \bar{U}) dans l'image de $(\Pi_n - \frac{1}{2n}\gamma_n\gamma_n^T)$. Notre estimateur nous donne donc le point de la droite passant par C_n et de vecteur directeur U le plus proche de l'origine du repère F_o .

- Avec plusieurs points de vue, le point P est correctement déterminé et l'estimateur fournit cette fois-ci le point de l'axe du cylindre le plus proche de l'origine du repère F_o .

Un estimateur du rayon r peut être déduit de l'équation (5.50) :

$$\hat{r}_n = \frac{1}{4n}(2\gamma_n^T \hat{P} + \gamma_n^T \hat{U}_n - 2\mu_n) \quad (5.52)$$

Cet estimateur du rayon dépend de \hat{P} . Avec un seul point de vue, il est impossible de reconstruire le rayon puisque nous avons :

$$\hat{r}_n = \frac{1}{4n}(2\gamma_n^T \bar{C}_n + 0 - 2\gamma_n^T \bar{C}_n) = 0$$

Mais avec plusieurs points de vue le rayon peut être complètement déterminé.

Dans certains cas, le rayon du cylindre peut être connu a priori. Nous tirons alors de l'équation 5.49, une expression plus simple de l'estimateur de \bar{P} :

$$\hat{P}_n = \Pi_n^+(\beta_n + r\gamma_n) \quad (5.53)$$

Si le rayon est connu, un seul point de vue suffit pour déterminer l'axe du cylindre. Nous constatons que les estimateurs \hat{P}_n et \hat{r}_n sont étroitement liés et que leurs valeurs dépendent l'une de l'autre comme l'ont montré Spica et al. [Spica et al., 2014a].

Comme pour le cas du point et de la droite, nous pouvons mettre à jour notre estimation avec l'image $n + 1$ avec un coût constant. Notre estimateur récursif fonctionne donc ainsi :

$$\phi_{n+1} = \bar{N}_{n+1} \bar{N}_{n+1}^T \quad (5.54)$$

$$\psi_{n+1} = \bar{Q}_{n+1} \bar{Q}_{n+1}^T \quad (5.55)$$

$$\Pi_{n+1} = \Pi_n + \phi_{n+1} + \psi_{n+1} \quad (5.56)$$

$$\gamma_{n+1} = \gamma_n + a_{n+1} \bar{N}_{n+1} + b_{n+1} \bar{Q}_{n+1} \quad (5.57)$$

$$\beta_{n+1} = [\phi_{n+1} + \psi_{n+1}] \bar{C}_{n+1} \quad (5.58)$$

$$\mu_{n+1} = \bar{C}_{n+1}^T (a_{n+1} \bar{N}_{n+1} + b_{n+1} \bar{Q}_{n+1}) \quad (5.59)$$

$$\hat{U}_{n+1} = \text{Vecteur propre associé à la plus petite valeur propre de } \Pi_{n+1} \quad (5.60)$$

$$\hat{P}_{n+1} = [\Pi_{n+1} - \frac{1}{2n} \gamma_{n+1} \gamma_{n+1}^T]^+ (\beta_{n+1} + \frac{1}{n} \gamma_{n+1} \gamma_{n+1}^T \hat{U}_{n+1} - \frac{\mu_{n+1}}{2n} \gamma_{n+1}) \quad (5.61)$$

$$\hat{r}_{n+1} = \frac{1}{4n} (2\gamma_{n+1}^T \hat{P}_{n+1} + \gamma_{n+1}^T \hat{U}_{n+1} - 2\mu_n) \quad (5.62)$$

Le calcul des indices visuels et de la matrice d'interaction correspondante au cylindre est donné en annexe A.

5.5.2.2 Cas du cylindre de longueur finie

Considérons maintenant un cylindre de longueur finie. Le cylindre est complètement déterminé, si en plus de l'axe de symétrie, et du rayon, nous pouvons calculer les centres des faces circulaires. Nous définissons deux nouveaux paramètres :

- B le centre de la face inférieure du cylindre.
- T le centre de la face supérieure.

Ces deux points sont situés sur l'axe du cylindre et s'expriment comme suit en fonction de P et U .

$$\bar{B} = \bar{P} + k_b \bar{U} \quad \bar{T} = \bar{P} + k_t \bar{U} \quad (5.63)$$

avec k_b et k_t deux réels. $|k_b - k_t|$ est la longueur de notre cylindre. Nous pouvons donc remplacer P et P' dans les équations (5.33) à (5.36) par B et T , il vient :

$$\bar{N}_i \bar{N}_i^T (\bar{B} - \bar{C}_i) = a_i r \bar{N}_i \quad (5.64)$$

$$\bar{N}_i \bar{N}_i^T (\bar{T} - \bar{C}_i) = a_i r \bar{N}_i \quad (5.65)$$

$$\bar{Q}_i \bar{Q}_i^T (\bar{B} - \bar{C}_i) = b_i r \bar{Q}_i \quad (5.66)$$

$$\bar{Q}_i \bar{Q}_i^T (\bar{T} - \bar{C}_i) = b_i r \bar{Q}_i \quad (5.67)$$

où a_i, b_i, N_i, Q_i conservent les mêmes définitions que celles données pour le cylindre de longueur infinie. Si l'image de ce dernier est décrite par deux droites, l'image d'un cylindre de longueur finie peut être caractérisée par deux segments de droite. Les extrémités de ces segments sont des points. De ces quatre points, et de la position C_i de la caméra, l'on peut former quatre droites, quatre faisceaux. Les composantes des vecteurs directeurs de ces droites, dans le repère F_i lié à la caméra, peuvent être calculées en utilisant l'équation (5.1). Ces vecteurs seront notés A_i, G_i, D_i et M_i .

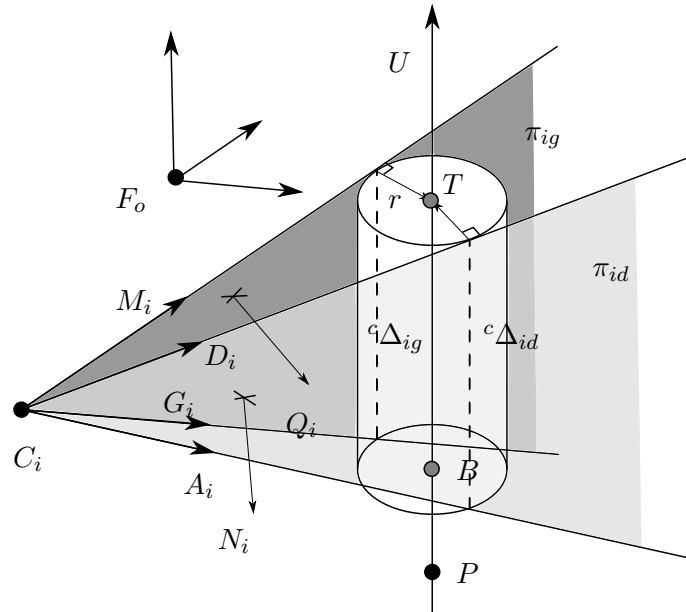


FIGURE 5.24 – Paramètres et variables pour un cylindre court.

Considérons maintenant la figure 5.24 où nous avons représenté ces quatre vecteurs. Les droites qu'ils portent, appartiennent au plans tangents π_{ig} et π_{id} et sont sécantes aux lignes de contact ${}^c\Delta_{ig}$ et ${}^c\Delta_{id}$. Les projections de B et de T sur ces droites se confondent avec la projection de B et de T sur les plans tangents. Nous pouvons donc encore adapter les équations (5.33) à (5.36) et écrire :

$$(I_3 - \bar{A}_i \bar{A}_i^T) (\bar{B} - \bar{C}_i) = a_i r \bar{N}_i \quad (5.68)$$

$$(I_3 - \bar{D}_i \bar{D}_i^T) (\bar{T} - \bar{C}_i) = a_i r \bar{N}_i \quad (5.69)$$

$$(I_3 - \bar{G}_i \bar{G}_i^T) (\bar{B} - \bar{C}_i) = b_i r \bar{Q}_i \quad (5.70)$$

$$(I_3 - \bar{M}_i \bar{M}_i^T) (\bar{T} - \bar{C}_i) = b_i r \bar{Q}_i \quad (5.71)$$

Nous considérons la possibilité que dans certaines images, toutes les extrémités des cylindres ne soient pas visibles. Nous introduisons donc des variables d'activations f_i, g_i, h_i et j_i qui sont égales à zéro si les vecteurs A_i, D_i, G_i et M_i dans l'ordre respectif ne peuvent pas être extraits de l'image de la caméra à la position C_i . Si leur vecteur correspondant peuvent être extraits, ces variables valent 1. Opérons également un changement

de variables pour la lisibilité des prochaines équations (ϕ_i et ψ_i représentent les mêmes variables que dans le cas du cylindre de longueur infinie) :

$$\phi_i = \bar{N}_i \bar{N}_i^T \quad \Delta_i = I_3 - \bar{A}_i \bar{A}_i^T \quad \Theta_i = I_3 - \bar{D}_i \bar{D}_i^T \quad (5.72)$$

$$\psi_i = \bar{Q}_i \bar{Q}_i^T \quad \chi_i = I_3 - \bar{G}_i \bar{G}_i^T \quad \Omega_i = I_3 - \bar{M}_i \bar{M}_i^T \quad (5.73)$$

Comme pour le cas du cylindre infini, nous définissons une erreur géométrique comme la somme du carré de la différence des termes de gauche et de droites des égalités (5.64) à (5.71). En tenant compte du fait que les vecteurs A_i , G_i , D_i et M_i , peuvent être inconnus pour certaines images, cette erreur a pour expression :

$$\begin{aligned} e_i^s = & (\bar{B} - \bar{C}_i - a_i r \bar{N}_i)^T \phi_i (\bar{B} - \bar{C}_i - a_i r \bar{N}_i) + (\bar{B} - \bar{C}_i - b_i r \bar{Q}_i)^T \psi_i (\bar{B} - \bar{C}_i - b_i r \bar{Q}_i) + \\ & (\bar{T} - \bar{C}_i - a_i r N_i)^T \phi_i (\bar{T} - \bar{C}_i - a_i r N_i) + (\bar{T} - \bar{C}_i - b_i r \bar{Q}_i)^T \psi_i (\bar{T} - \bar{C}_i - b_i r \bar{Q}_i) + \\ & f_i \cdot (\Delta_i (\bar{B} - \bar{C}_i) - a_i r N_i)^T (\Delta_i (\bar{B} - \bar{C}_i) - a_i r N_i) + \\ & g_i \cdot (\chi_i (\bar{B} - \bar{C}_i) - b_i r \bar{Q}_i)^T (\chi_i (\bar{B} - \bar{C}_i) - b_i r \bar{Q}_i) + \\ & h_i \cdot (\Theta_i (\bar{T} - \bar{C}_i) - a_i r N_i)^T (\Theta_i (\bar{T} - \bar{C}_i) - a_i r N_i) + \\ & j_i \cdot (\Omega_i (\bar{T} - \bar{C}_i) - b_i r \bar{Q}_i)^T (\Omega_i (\bar{T} - \bar{C}_i) - b_i r \bar{Q}_i) \end{aligned} \quad (5.74)$$

Notre critère sur n images s'écrit ainsi :

$$E_n^s = \sum_{i=1}^n e_i^s \quad (5.75)$$

Pour la lisibilité de notre critère sur n images, nous définissons ces variables :

$$\begin{aligned} \Pi_n &= \sum_{i=1}^n [\phi_i + \psi_i] & \Psi_b &= \sum_{i=1}^n [f_i \Delta_i + g_i \chi_i] & \Psi_t &= \sum_{i=1}^n [h_i \Theta_i + j_i \Omega_i] \\ \beta_n &= \sum_{i=1}^n [\phi_i + \psi_i] \bar{C}_i & \lambda_b &= \sum_{i=1}^n [f_i \Delta_i + g_i \chi_i] \bar{C}_i & \lambda_t &= \sum_{i=1}^n [h_i \Theta_i + j_i \Omega_i] \bar{C}_i \\ \gamma_n &= \sum_{i=1}^n [a_i \bar{N}_i + b_i \bar{Q}_i] & \delta_b &= \sum_{i=1}^n [f_i a_i \Delta_i \bar{N}_i + g_i b_i \chi_i \bar{Q}_i] & \delta_t &= \sum_{i=1}^n [h_i a_i \Theta_i \bar{N}_i + j_i b_i \Omega_i \bar{Q}_i] \\ \mu_n &= \sum_{i=1}^n \bar{C}_i^T [a_i \bar{N}_i + b_i \bar{Q}_i] \end{aligned}$$

$$\begin{aligned} \epsilon_n &= \sum_{i=1}^n \bar{C}_i^T [a_i [f_i \Delta_i + h_i \Theta_i] \bar{N}_i + b_i [g_i \chi_i + j_i \Omega_i] \bar{Q}_i] \\ \sigma_n &= \sum_{i=1}^n \bar{C}_i^T [2\phi_i + 2\psi_i + f_i \Delta_i + h_i \Theta_i + g_i \chi_i + j_i \Omega_i] \bar{C}_i \end{aligned}$$

$$\omega_b = \sum_{i=1}^n [f_i + g_i] \quad \omega_t = \sum_{i=1}^n [h_i + j_i] \quad \omega_n = \omega_t + \omega_b$$

Les variables Π_n , β_n , γ_n et μ_n (première colonne de gauche ci-dessus) sont exactement les mêmes que celles définies dans le cas du cylindre de longueur infinie. En développant et simplifiant l'équation (5.74), en la réinjectant dans (5.75), puis en utilisant les variables que nous venons de définir, nous obtenons cette expression de notre critère :

$$\begin{aligned} E_n^s = & \bar{B}^T [\Pi_n + \Psi_b] \bar{B} + \bar{T}^T [\Pi_n + \Psi_t] \bar{T} - 2\bar{B}^T (\beta_n + \lambda_b) - 2\bar{T}^T (\beta_n + \lambda_t) \\ & - 2r \bar{B}^T (\gamma_n + \delta_b) - 2r \bar{T}^T (\gamma_n + \delta_t) + \sigma_n + 2r(2\mu_n + \epsilon_n) + (4n + \omega_n)r^2 \end{aligned} \quad (5.76)$$

Ce critère est quadratique en fonction des paramètres \bar{B} , \bar{T} et r . Nous nous plaçons au minimum et nous obtenons ce système d'équations :

$$\begin{aligned} (\Pi_n + \Psi_b)\bar{B} - r(\gamma_n + \delta_b) &= \beta_n + \lambda_b \\ (\Pi_n + \Psi_t)\bar{T} - r(\gamma_n + \delta_t) &= \beta_n + \lambda_t \\ (\gamma_n + \delta_b)^T \bar{B} + (\gamma_n + \delta_t)^T \bar{T} - (4n + \omega_n)r &= 2\mu_n + \epsilon_n \end{aligned} \quad (5.77)$$

Remarquons que toutes les variables (Π_n , β_n , γ_n et μ_n) permettant de déterminer les paramètres \bar{P} , \bar{U} et r dans le cas du cylindre de longueur infinie ont été gardées. Nous déterminons donc ces paramètres en utilisant les équations proposées précédemment. Ces paramètres vérifient alors les égalités suivantes :

$$\Pi_n \bar{U} = 0 \quad \Pi_n \bar{P} = \beta_n + r\gamma_n \quad \gamma_n^T (2\bar{P} + \bar{U}) = 4nr + 2\mu_n$$

En exploitant ces égalités et en remplaçant \bar{B} et \bar{T} par leur expression (5.63), le système d'équations devient après quelques simplifications :

$$\begin{aligned} k_b &= \lambda_b + r\delta_b - \Psi_b \bar{P} \\ [\Psi_t \bar{U}] k_t &= \lambda_t + r\delta_t - \Psi_t \bar{P} \\ [(\delta_b + \gamma_n)^T \bar{U}] k_b + [(\delta_t + \gamma_n)^T \bar{U}] k_t &= \epsilon_n + r\omega_n - (\delta_b + \delta_t)^T \bar{P} \end{aligned} \quad (5.78)$$

ou présenté sous forme matricielle :

$$\begin{bmatrix} \Psi_b \bar{U} & 0_{3 \times 1} \\ 0_{3 \times 1} & \Psi_t \bar{U} \\ (\delta_b + \gamma_n)^T \bar{U} & (\delta_t + \gamma_n)^T \bar{U} \end{bmatrix} \begin{bmatrix} k_b \\ k_t \end{bmatrix} = \begin{bmatrix} \lambda_b + r\delta_b - \Psi_b \bar{P} \\ \lambda_t + r\delta_t - \Psi_t \bar{P} \\ \epsilon_n + r\omega_n - (\delta_b + \delta_t)^T \bar{P} \end{bmatrix} \quad (5.79)$$

Étudions ce système d'équations. Les variables ω_b , ω_t et ω_n sont fonction de la somme des variables d'activation f_i , g_i , h_i et j_i . Lorsque $\omega_b = 0$ (respectivement $\omega_t = 0$) aucune information sur la face inférieure (respectivement supérieure) du cylindre n'est disponible pour le calcul. Si les deux sont nuls, $\omega_n = 0$, nous retombons dans le cas du cylindre sans longueur précise. Par construction on a :

$$\omega_b = 0 \implies \begin{cases} \Psi_b = 0 \\ \delta_b = 0 \\ \lambda_b = 0 \end{cases} \quad \omega_t = 0 \implies \begin{cases} \Psi_t = 0 \\ \delta_t = 0 \\ \lambda_t = 0 \end{cases}$$

Si $\omega_b \neq 0$, k_b peut être déterminé. Et si $\omega_t \neq 0$, k_t peut être déterminé. Ces paramètres sont calculés ainsi :

$$\begin{bmatrix} \hat{k}_b \\ \hat{k}_t \end{bmatrix} = \begin{bmatrix} \Psi_b \bar{U} & 0_{3 \times 1} \\ 0_{3 \times 1} & \Psi_t \bar{U} \\ (\delta_b + \gamma_n)^T \bar{U} & (\delta_t + \gamma_n)^T \bar{U} \end{bmatrix}^+ \begin{bmatrix} \lambda_b + r\delta_b - \Psi_b \bar{P} \\ \lambda_t + r\delta_t - \Psi_t \bar{P} \\ \epsilon_n + r\omega_n - (\delta_b + \delta_t)^T \bar{P} \end{bmatrix} \quad (5.80)$$

Les estimateurs de \bar{B} et de \bar{T} sont donnés par :

$$\hat{B} = \hat{P} + \hat{k}_b \hat{U} \quad (5.81)$$

$$\hat{T} = \hat{P} + \hat{k}_t \hat{U} \quad (5.82)$$

Remarquons que si $\omega_b \neq 0$ et $\omega_t \neq 0$, les paramètres \bar{B} , \bar{T} et r peuvent être déterminés directement à partir du système d'équations (5.77) :

$$\begin{bmatrix} \hat{B} \\ \hat{T} \\ \hat{r} \end{bmatrix} = \begin{bmatrix} \Pi_n + \Psi_b & 0_{3 \times 3} & -\gamma_n - \delta_b \\ 0_{3 \times 3} & \Pi_n + \Psi_t & -\gamma_n - \delta_t \\ (\gamma_n + \delta_b)^T & (\gamma_n + \delta_t)^T & -4n - \omega_n \end{bmatrix}^+ \begin{bmatrix} \beta_n + \lambda_b \\ \beta_n + \lambda_t \\ 2\mu_n + \epsilon_n \end{bmatrix} \quad (5.83)$$

Le calcul pour la reconstruction des indices visuels et de la matrice d'interaction correspondante au cylindre est disponible en annexe A.

5.5.2.3 Résultats de simulation

Nous avons testé l'estimation du cylindre de longueur finie en simulation. La longueur du cylindre a été fixée à 0.1 m et son rayon à 0.025 m . Nous avons réalisé une tâche référencée vision classique en utilisant les indices visuels décrivant l'image du cylindre. Le vecteur des indices visuels est donné par :

$$s_{sc} = [\rho_1 \quad \theta_1 \quad k_1^1 \quad k_2^1 \quad \rho_2 \quad \theta_2 \quad k_1^2 \quad k_2^2]^T$$

Ces indices visuels sont décrits dans la figure 5.25.

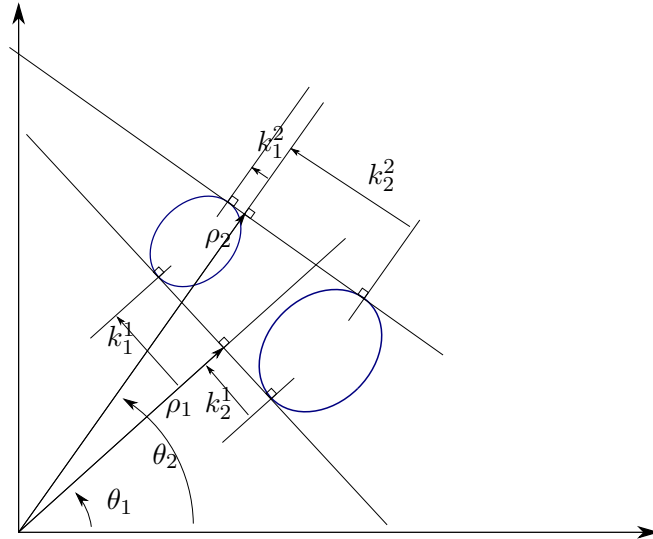


FIGURE 5.25 – Indices visuels pour un cylindre de longueur finie

La position finale C_n souhaitée de la caméra est à 0.5 m du cylindre, et la position initiale C_0 à 1.5 m . Dans la position désirée, la caméra fait face au cylindre, et nous avons changé l'orientation de la caméra à la pose initiale ($\theta U = [0.3 \ 0 \ 0.2]\text{ rad}$). Le cylindre et les positions extrêmes de la caméra sont représentées sur la figure 5.26. Les paramètres intrinsèques de la caméra sont les mêmes que pour les simulations avec le point et la droite.

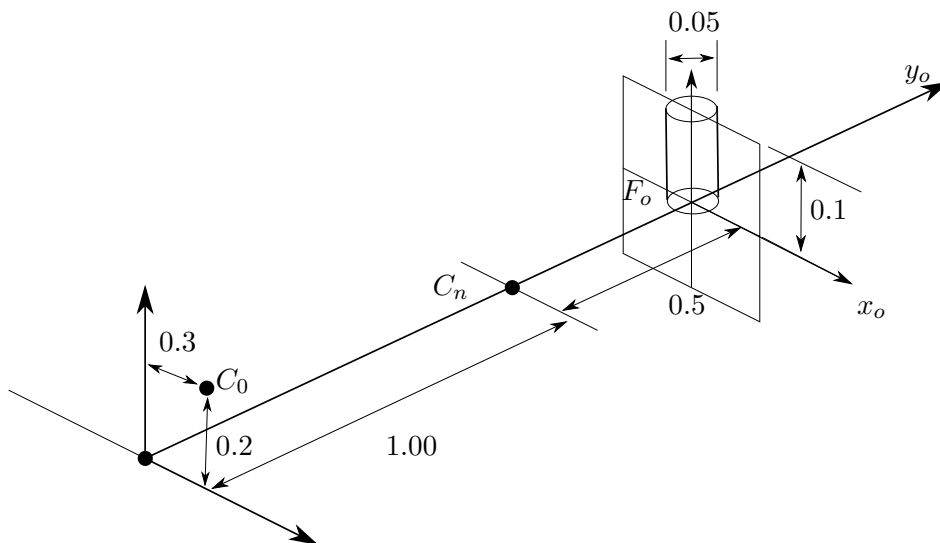


FIGURE 5.26 – Conditions de simulations pour le cylindre

Nous avons ajouté un bruit centré gaussien d'écart-type de 2 pixels sur les indices exprimant une distance ρ et θ et d'écart-type d'un degré sur les mesures d'angles θ . Nos

estimateurs ont utilisé ces mesures bruitées pour reconstruire la structure 3D du cylindre. Après seulement 1.5 secondes, nous avons alimenté la tâche référencée vision avec les indices visuels reconstruits.

La figure 5.27 présente l'évolution des composantes de la fonction de tâche. Comme pour le cas de la droite, la transition entre les mesures bruitées et les indices visuels reconstruits est imperceptible, si ce n'est l'effet lissant. La tâche de positionnement par rapport au cylindre est réalisée.

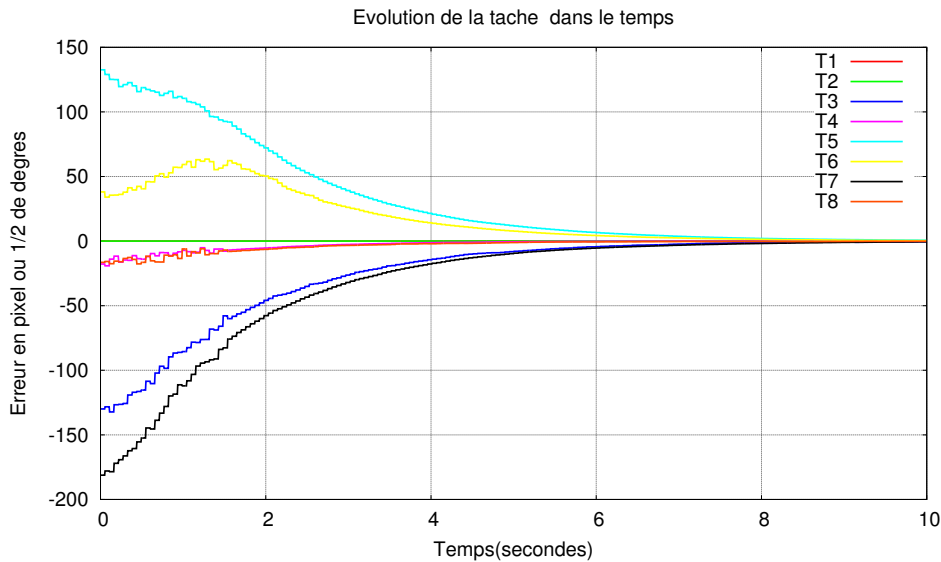


FIGURE 5.27 – Évolution des composantes de la tâche en fonction du temps.

La figure 5.28 illustre l'évolution de la valeur estimée du rayon du cylindre. Malgré le niveau de bruit injecté, et le faible rayon du cylindre par rapport à la distance à la caméra, notre estimateur converge vers la valeur réelle du rayon soit 0.025 m.

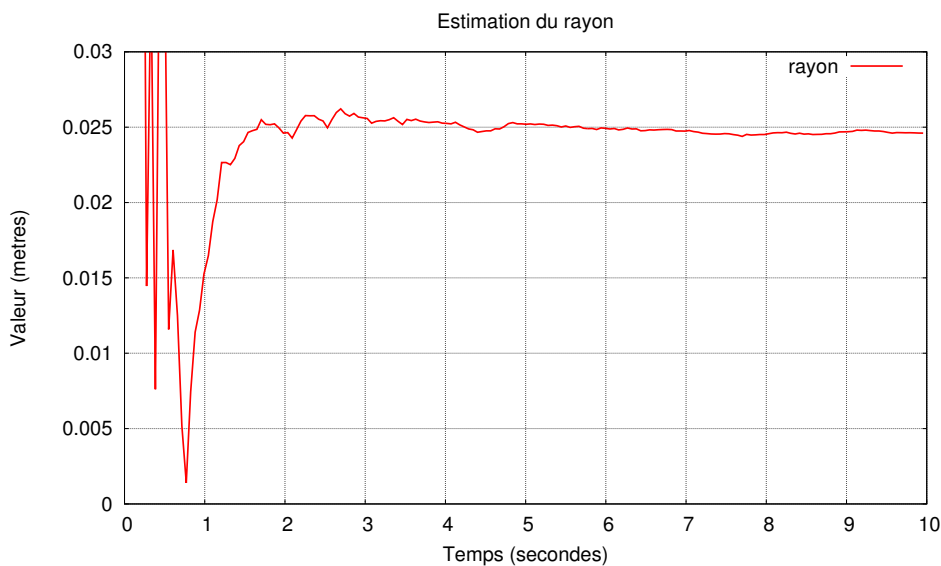


FIGURE 5.28 – Estimation du rayon du cylindre en fonction du temps.

Les figures 5.29 et 5.30 présentent l'évolution des coordonnées dans F_o , des points B et T , centres des faces inférieures et supérieures du cylindre. Nos estimateurs convergent

vers les valeurs réelles :

$$\bar{B} = [0 \ 0 \ 0]^T$$

$$\bar{T} = [0 \ 0 \ 0.1]^T$$

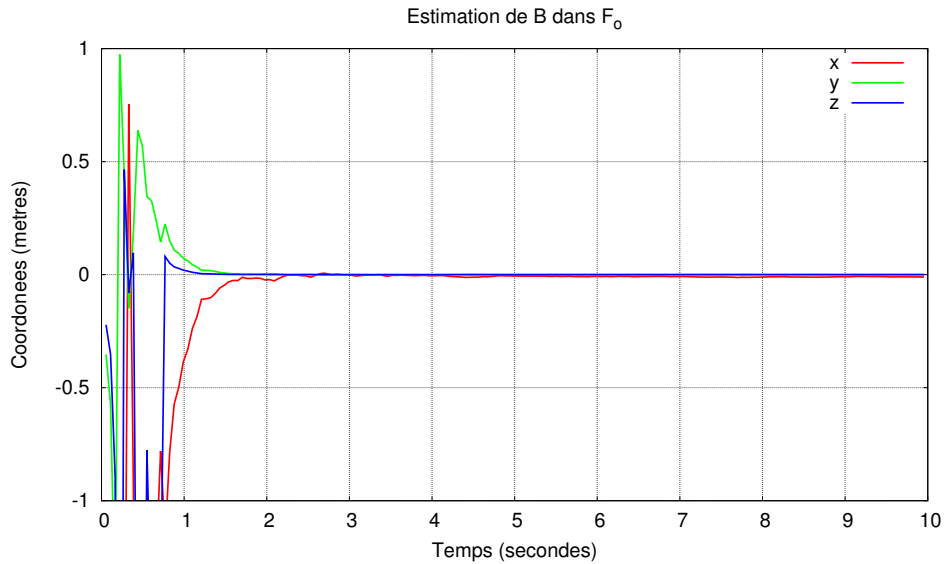


FIGURE 5.29 – Évolution des coordonnées du centre de la base du cylindre en fonction du temps.

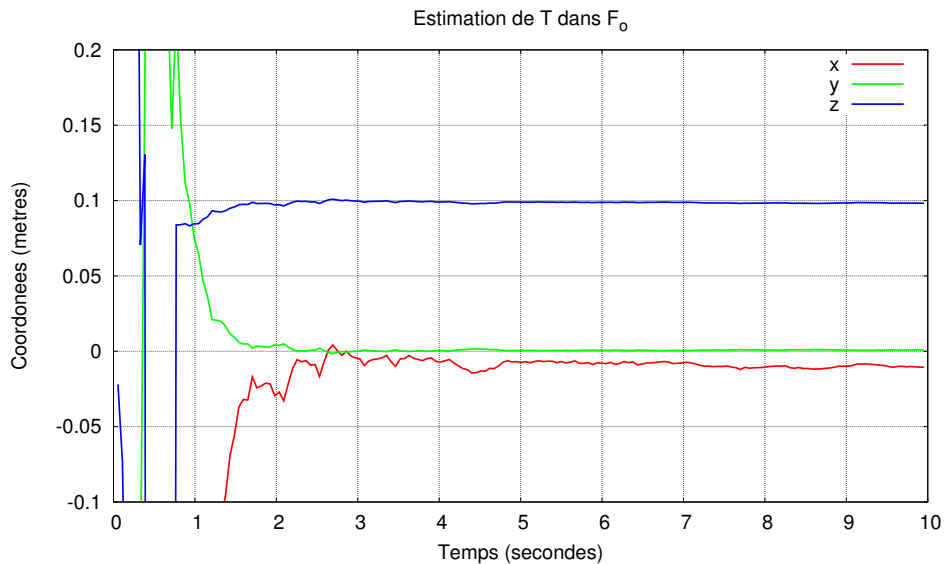


FIGURE 5.30 – Évolution des coordonnées du centre de la face supérieure du cylindre en fonction du temps.

Nous avons présenté les résultats obtenus en simulation pour le cylindre de longueur finie, qui est, nous l'avons vu, pour notre méthode d'estimation, un cas plus général que celui du cylindre de longueur infinie. Nous pouvons donc reconstruire pendant une tâche référencée vision, la structure 3D de cylindres en tenant compte ou non de leur longueur.

5.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'estimation de la structure 3D d'un objet en mouvement. Nous avons d'abord fait un survol des différentes méthodes traitant de la reconstruction de la structure d'objets à partir du mouvement. Nous avons cherché des techniques permettant de reconstruire des objets mobiles à partir de mesures issues d'une ou plusieurs caméras mobiles. La plupart des travaux de la communauté automatique, à notre connaissance, s'intéressent principalement au cas du point, et reconstruisent la profondeur du point fixe à partir d'une caméra mobile. Les méthodes de la communauté vision, permettent de déterminer les coordonnées d'un point fixe dans un repère fixe, à partir de deux ou plusieurs caméras.

Dans le contexte de notre problème, les objets manipulés sont mobiles ainsi que l'une de nos caméras. Nous avons cherché une approche qui puisse, à coût constant, intégrer toutes les mesures d'indices visuels disponibles pendant la réalisation de la tâche référencée vision pour déterminer la structure des objets mobiles. Nous avons choisi d'utiliser la méthode dite du "point milieu" généralisée à n vues pour estimer les coordonnées d'un point fixe dans un repère mobile à partir d'images fournies par une ou plusieurs caméras mobiles. Nous avons montré que l'erreur de reconstruction 3D diminue en fonction du nombre de vues, comme l'erreur de reconstruction des indices visuels qui évolue en fonction de l'inverse de la racine carrée du nombre n de vues. Nous avons jugé que ses performances étaient convenables pour nos tâches référencées vision.

Les objets manipulés, pour notre tâche de fermeture du stylo, étant modélisés comme des cylindres, nous avons étendu cette méthode du "point milieu" au cas de la droite et du cylindre. Dans ce but, nous avons exploité les orthogonalités présentes dans la géométrie de la projection de la droite et du cylindre dans l'image des caméras. Nous avons effectué de nombreuses simulations où nous réalisons des tâches référencées vision sur des points, des droites et des cylindres. Les estimations des paramètres décrivant les structures 3D ont toutes convergé vers les valeurs réelles constantes attendues. La connaissance de la structure 3D des objets nous a permis de reconstruire les indices visuels. Lorsque nous utilisons ces indices visuels reconstruits, les tâches sont correctement réalisées. Aucun biais appréciable dans la reconstruction des images n'a été détecté et un effet lissant est constaté. L'estimation de la structure 3D nous a également permis de calculer en ligne la valeur de certaines matrices nécessaires au calcul des lois de commande. Nous pensons que cette méthode peut être également appliquée à la sphère. Dans le chapitre suivant, nous présenterons des simulations couplant ces résultats avec les lois de commande établies dans les chapitres précédents.

Chapitre 6

Résultats

*Tande ak wè se de - Il faut le voir
pour le croire.*

Proverbe haïtien

6.1 Introduction

Tout au long de ce document, nous avons traité de la coordination des bras manipulateurs par asservissement visuel. Nous avons développé des lois de commande répondant à ce problème tout en prenant en compte les butées et la visibilité du motif visuel. Nous avons aussi proposé des solutions de reconstruction de ce dernier lorsqu'il ne peut être extrait de l'image. Nous présentons ici des résultats de simulation où toutes nos contributions sont intégrées pour réaliser une tâche de manipulation coordonnée particulière consistant à refermer un stylo. Il s'agit d'une tâche domestique d'assemblage, mais les résultats de cette étude peuvent être adaptés à des contextes plus industriels.

Nous rappelons que la tâche pour la fermeture du stylo a été décomposée en une séquence de trois sous-tâches :

- la première pour rendre coplanaires les axes du stylo et du capuchon.
- la deuxième pour aligner dans les espaces les axes des deux objets.
- enfin, la dernière, pour rapprocher les deux objets en les gardant alignés.

Nous rappelons également que nous avons défini deux autres tâches permettant respectivement d'éviter les butées et les occultations. Nous avons introduit une hiérarchie entre les tâches que nous avons classées dans l'ordre de priorité décroissant suivant :

- la tâche d'évitement des butées.
- la séquence des trois sous-tâches coordonnées.
- la tâche d'évitement des occultations.

Dans les lignes qui suivent, nous illustrerons deux scénarios : dans un premier cas nous simulerons une panne de la caméra, et dans le deuxième nous intégrerons le contrôle de la direction d'une des caméras fixées sur la tête.

6.2 Premier scénario : gestion d'une panne temporaire des caméras

Dans cette simulation, nous avons choisi de conserver les paramètres intrinsèques utilisés dans le chapitre 3, proches de ceux des caméras présentes sur le robot PR2. Ils sont

rappelés dans le tableau 6.1.

	Caméra sur l'avant bras	Caméra sur la tête
$M = \begin{bmatrix} f/\alpha_u & 0.0 & c_u \\ 0.0 & f/\alpha_v & c_v \end{bmatrix}$	$M = \begin{bmatrix} 320.0 & 0.0 & 320.5 \\ 0.0 & 320.0 & 240.5 \end{bmatrix}$	$M = \begin{bmatrix} 418.68 & 0.0 & 309.05 \\ 0.0 & 419.16 & 224.18 \end{bmatrix}$

TABLE 6.1 – Paramètres intrinsèques des caméras utilisés en simulation.

Pour la simulation présentée ici, nous avons gardé les conditions initiales du chapitre 3. Nous les reproduisons dans le tableau 6.2 :

Articulation	Bras gauche	Bras droit
shoulder_pan_joint	$\pi/4$	$-\pi/4$
shoulder_lift_joint	0.1	-0.1
upper_arm_roll_joint	$\pi/2$	$-\pi/2$
elbow_flex_joint	$-\pi/2$	$-\pi/2$
forearm_roll_joint	$-\pi/2$	$\pi/2$
wrist_flex_joint	-0.05	-0.05
wrist_roll_joint	0.0	0.0

TABLE 6.2 – Valeurs initiales des coordonnées articulaires des deux bras du PR2.

Les limites des articulations ont été données au chapitre 4. Nous avons considéré un débit de 15 images par seconde provenant des deux caméras. La loi de commande a donc été mise à jour à une fréquence de 15 Hertz. Le gain λ de toutes les tâches référencées vision a été fixé à 1. Nous avons ajouté un bruit d'écart-type de 2 pixels sur les indices visuels mesurant des distances sur l'image ρ et θ . Un bruit d'écart-type de 1 degré a été ajouté sur les mesures d'angles. Nous avons simulé une panne de toutes les caméras 3 secondes après le lancement des tâches. Pendant cette panne, aucune mesure des indices visuels n'est possible. Nous fournissons à la tâche, les indices visuels reconstruits, pendant que nos estimations ne sont plus mises à jour. 8 secondes après le début, nous activons à nouveau les mesures des indices visuels. Le début de la panne simulée est matérialisée sur les figures par une flèche rouge verticale vers le bas, tandis qu'une flèche bleue verticale signale la fin de la panne.

La figure 6.1 présente l'évolution des composantes de la séquence des trois sous-tâches. Les instants de transition entre les sous-tâches sont matérialisées par des lignes noires épaisses et verticales. Nous constatons que toutes les composantes convergent vers zéro, la fermeture du stylo est donc correctement effectuée. La période de panne entre $t = 3 \text{ secs}$ et $t = 8 \text{ secs}$ n'a aucun impact sur la réalisation de la tâche. Nous pouvons en effet noter que la transition entre indices visuels réels et reconstruits est continue. La différence notable est l'absence de bruit apparent dû à l'exploitation des indices visuels reconstruits. Nous pouvons donc conclure à l'efficacité de la méthode de reconstruction choisie en dépit des bruits rajoutés sur les indices visuels et les mesures des angles des liaisons.

La figure 6.2 montre l'évolution de la tâche d'évitement des butées. Toutes les composantes F_{bi} non nulles convergent rapidement vers zéro. Seules deux composantes F_{b6} et F_{b13} sont non nulles au début. Ces courbes correspondent à deux liaisons qui sont près de leur limites¹. Pour toutes les autres articulations, l'évitement des butées est inactif.

1. Ce sont ici les articulations des poignets "wrist_flex_joint" des deux bras.

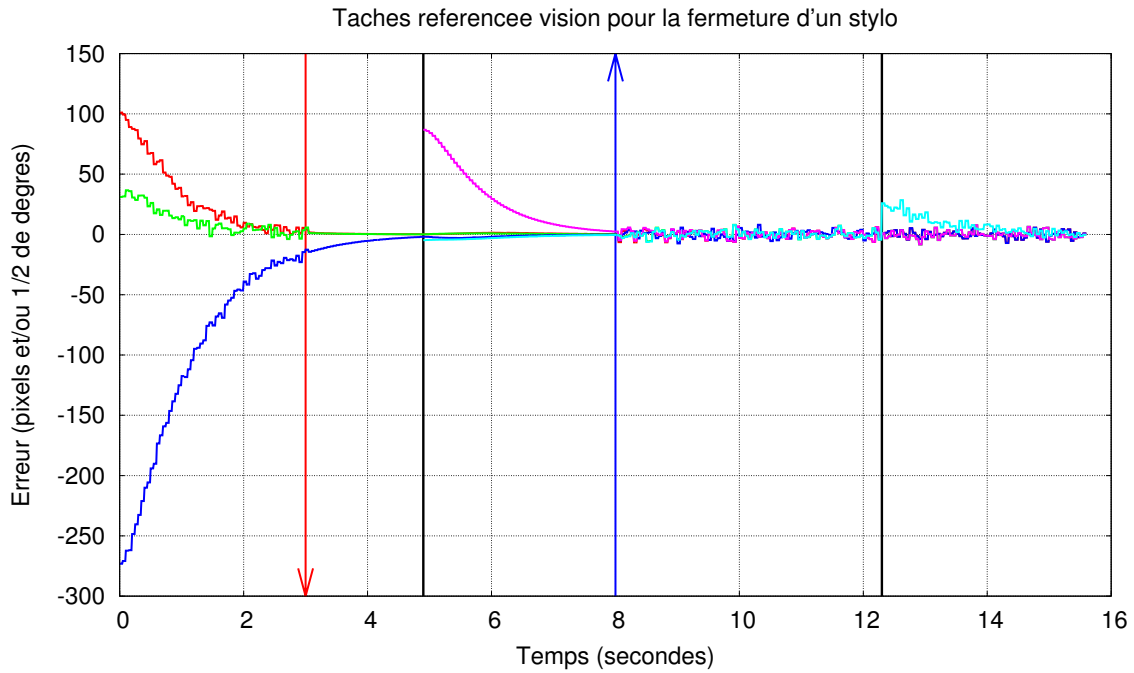


FIGURE 6.1 – Évolution de la tâche coordonnée référencée vision. Simulation d'une panne dans le traitement d'images.

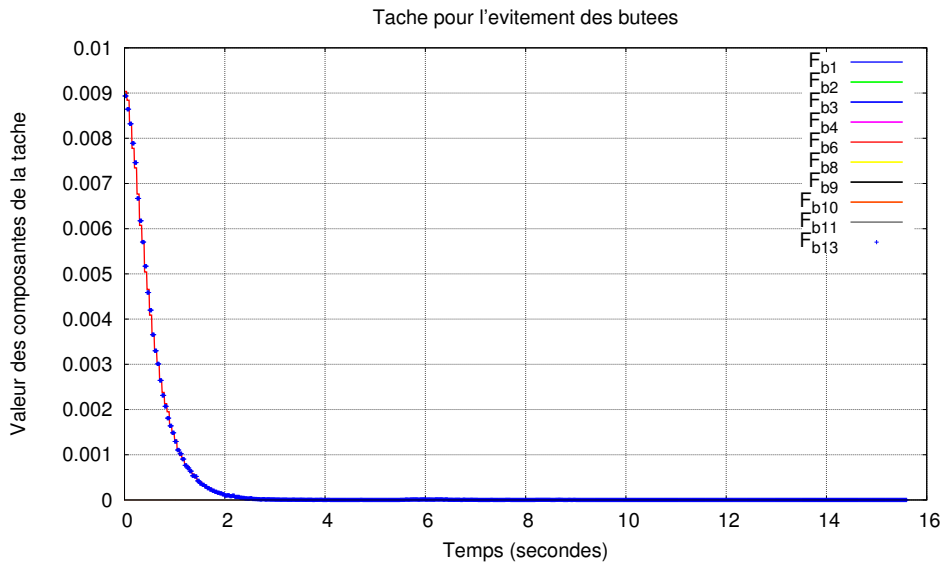


FIGURE 6.2 – Évolution de la fonction de tâche d'évitement des butées F_b .

Les figures 6.3 et 6.4 présentent l'évolution des vitesses articulaires envoyées au robot. Les commandes sont similaires, indiquant une bonne répartition des efforts entre les deux bras, qui se coordonnent. La panne n'a pas affecté la commande des bras du robot, et aucun à-coup dû à ce problème temporaire n'a été constaté. Cette continuité des vitesses témoigne de l'efficacité de notre stratégie de commande et par là même des performances des différentes techniques de gestion des occultations et de la redondance qui sont à sa base.

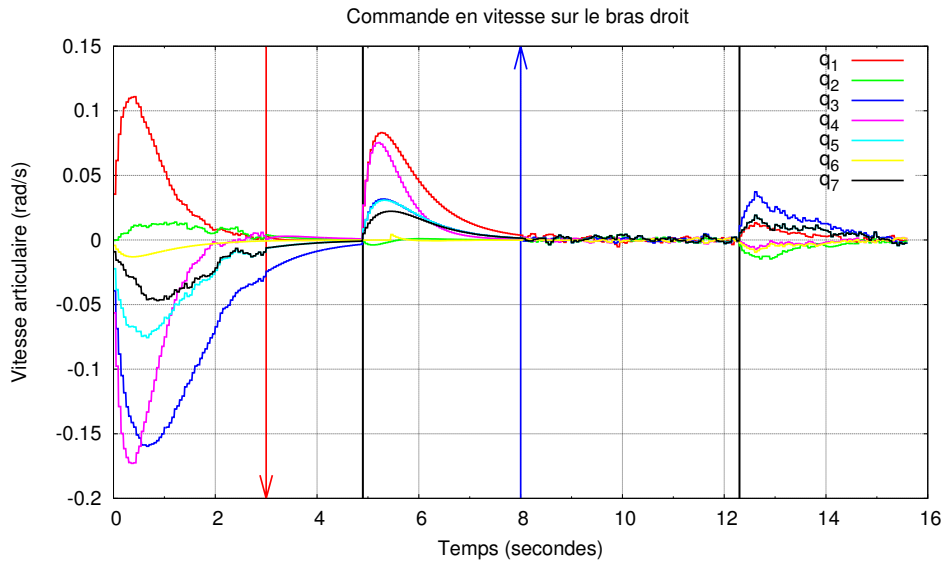


FIGURE 6.3 – Commande articulaire du bras droit du PR2.

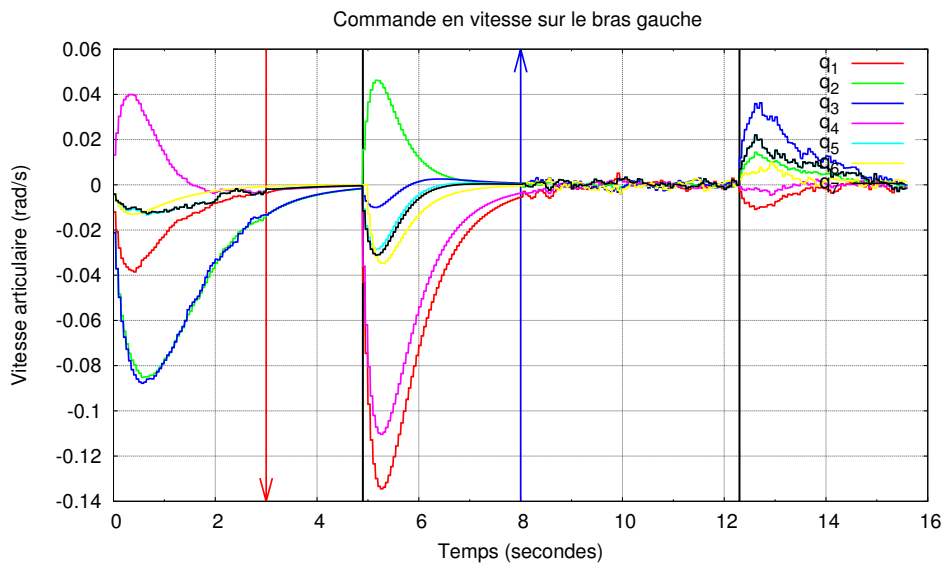


FIGURE 6.4 – Commande articulaire du bras gauche du PR2.

Nous nous focalisons maintenant sur la reconstruction des indices visuels utilisés pour notre tâche. Ils ont été décrits dans la section 3.4. Ils décrivent l'image d'une droite, l'axe de symétrie de l'objet, et la position sur cette droite d'un point-extrémité du même objet (cf. figure 3.13). Pour reconstruire les indices visuels nécessaires à notre tâche, nous devons donc exploiter les algorithmes décrits précédemment pour déterminer la structure 3D d'un point et d'une droite.

Nous présentons les résultats obtenus pour les paramètres 3D du stylo dans les figures suivantes. L'axe de symétrie est défini ici par un point P et son vecteur directeur U . L'extrémité du stylo est caractérisée par le point K . Rappelons que ces paramètres sont exprimés dans le repère lié au stylo. Dans ce repère, ils ont pour valeur :

$$P = [0 \ 0 \ 0]^T \quad U = [0 \ 0 \ 1]^T \quad K = [0 \ 0 \ 0.07]^T$$

La longueur du stylo a été prise ici égale à 7 cm. Les figures 6.5, 6.6 et 6.7 présentent l'évolution de l'estimation de P , U et K dans le repère lié au stylo. Les estimations

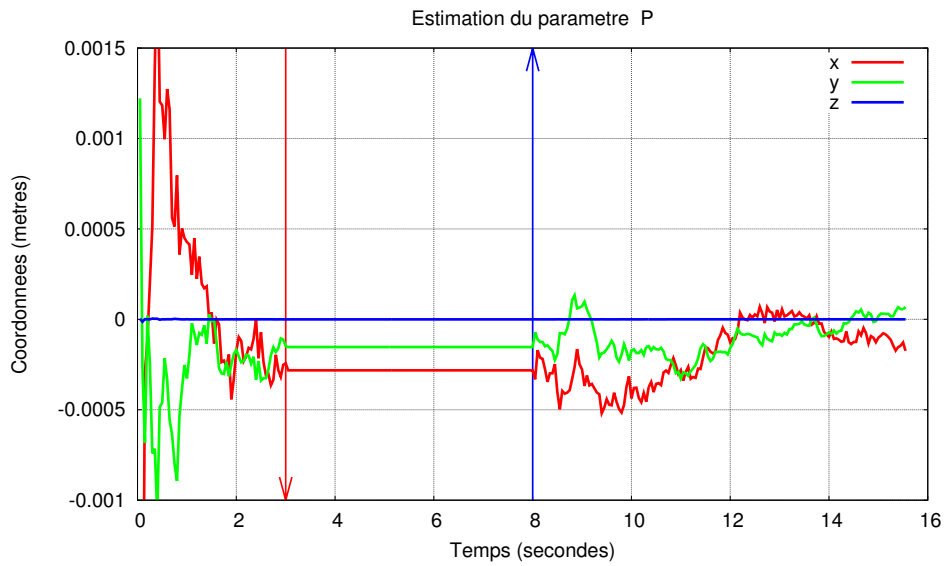


FIGURE 6.5 – Estimation du paramètre P du stylo.

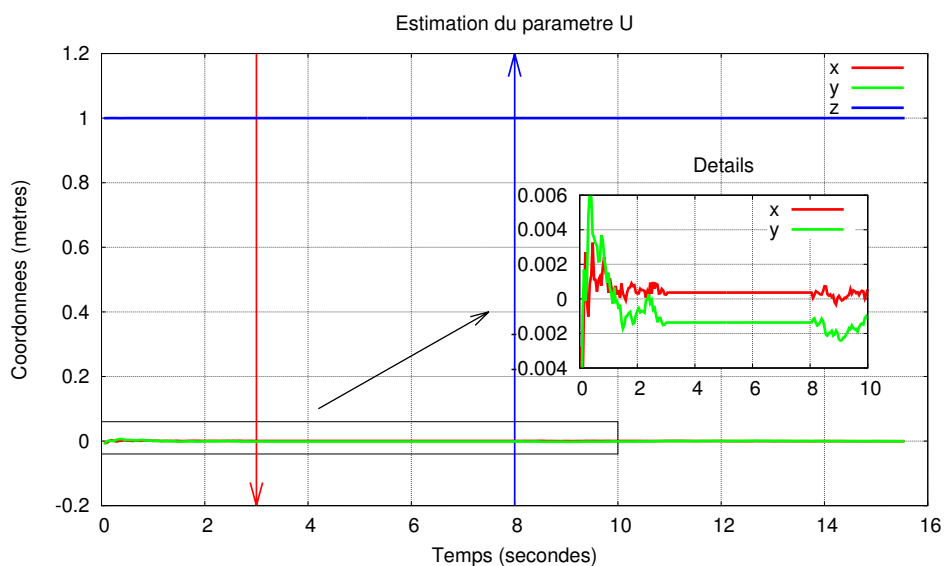


FIGURE 6.6 – Estimation du paramètre U du stylo.

convergent rapidement vers les valeurs réelles malgré l'importance du bruit ajouté sur les mesures (écart-type 2 pixels). Nous pouvons remarquer qu'entre $t = 3$ secs et $t = 8$ secs, toutes les estimations sont constantes. Cette période correspond à la panne simulée des caméras. Pendant ce laps de temps, aucun estimateur n'est mis à jour. Les indices visuels sont reconstruits fidèlement, malgré le mouvement des caméras et des objets.

La très bonne précision obtenue pour l'estimation (avoisinant le millimètre), est due ici à l'utilisation des deux caméras qui permettent de bénéficier de deux points de vue complémentaires de la scène, augmentant ainsi l'information disponible pour retrouver

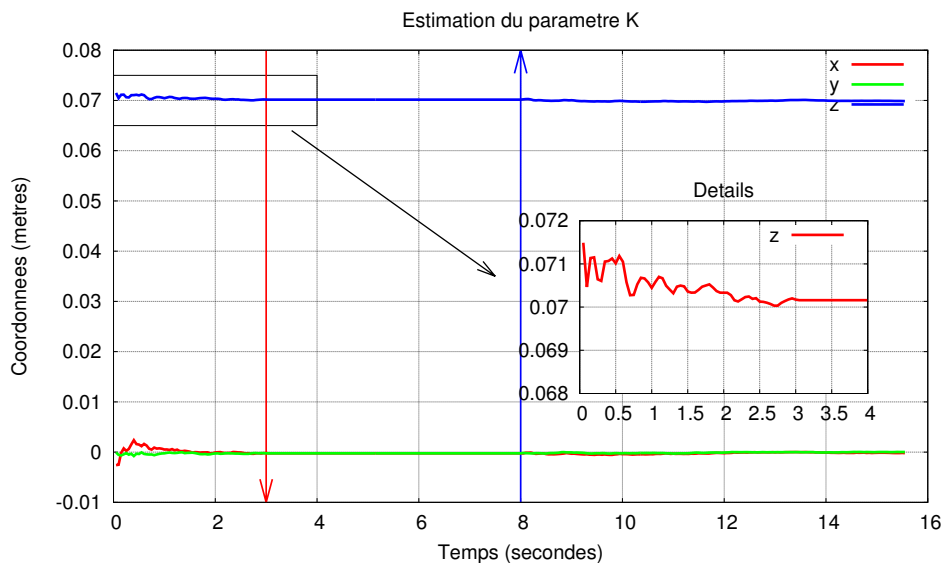


FIGURE 6.7 – Estimation du paramètre K du stylo.

la structure 3D. Notons de plus que les estimateurs que nous avons développés continuent d'intégrer les informations apportées par le déplacement des objets et des caméras, permettant d'affiner encore l'estimation.

6.3 Second scénario : extension de la coordination à la tête et aux bras du robot

Nous nous intéressons maintenant à la seconde tâche consistant à refermer le stylo en contrôlant la direction de la caméra positionnée sur la tête du robot. En effet, jusqu'ici cette caméra restait fixe. Notre idée est donc maintenant d'utiliser ses degrés de liberté en site et azimuth pour maintenir les objets manipulés dans son champ de vue. Nous rajoutons donc deux degrés de liberté supplémentaires qu'il s'agira de coordonner avec ceux des bras manipulateurs. De plus, nous avons également modifié les indices visuels utilisés pour considérer ceux définissant le cylindre de longueur finie. Ces indices ont déjà été présentés sur la figure 5.25 que nous avons reproduite ici (cf. figure 6.8) afin de faciliter la lecture.

Le vecteur des indices visuels décrivant l'image d'un cylindre est donc défini par :

$$s = [\rho_1 \quad \theta_1 \quad k_{1a} \quad k_{1b} \quad \rho_2 \quad \theta_2 \quad k_{2a} \quad k_{2b}]^T$$

Nous noterons ici $^s s$ et $^c s$ les vecteurs des indices visuels correspondant respectivement au stylo et au capuchon fournis par la caméra fixée sur le bras. Les indices visuels donnés par la caméra de la tête sont notés $^{ts} s$ et $^{tc} s$.

Avant de présenter nos résultats, nous indiquons d'abord les adaptations que nous avons dû apporter suite aux deux modifications mentionnées plus haut (utilisation de la caméra de la tête et changement des indices visuels). Tout d'abord, concernant la tâche de coordination, la séquence des trois sous-tâches visuelles a été définie de la même manière en comparant les vecteurs des indices visuels décrivant le stylo et le capuchon. Les composantes de ces trois sous-tâches sont données dans le tableau 6.3.

Nous avons adapté la tâche d'évitement d'occultation pour tenir compte du nouveau jeu d'indices visuels. Nous avons fixé la longueur du stylo à 10 cm, celle du capuchon à

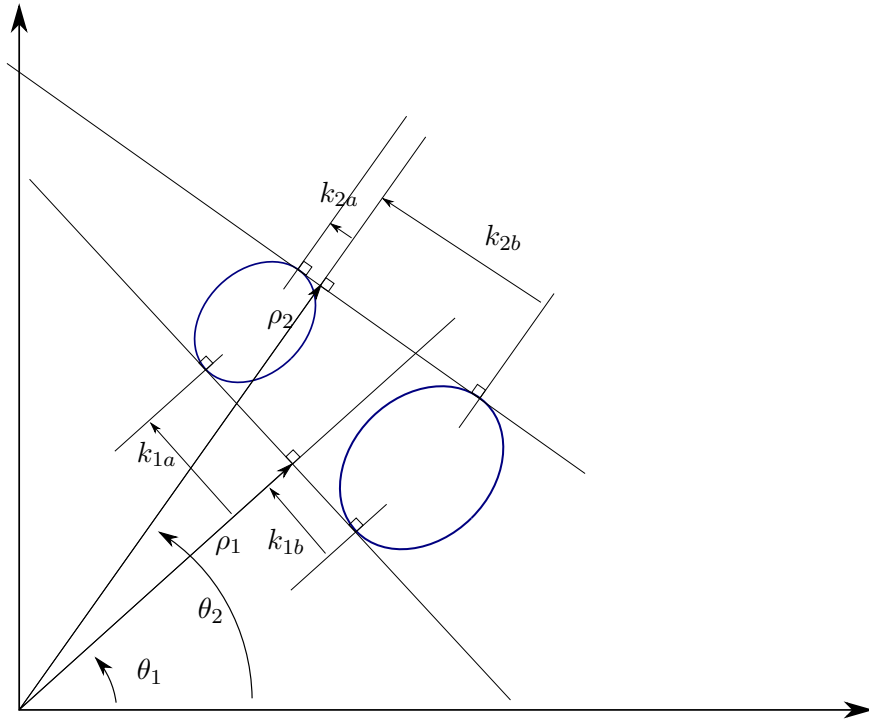


FIGURE 6.8 – Indices visuels pour un cylindre de longueur finie

Composante v_i	e_1	e_2	e_3
v_1	${}^s k_{1b} - {}^c k_{1b} - d$	${}^s \rho_1 - {}^c \rho_1$	${}^s \rho_1 - {}^c \rho_1$
v_2		${}^s \theta_1 - {}^c \theta_1$	${}^s \theta_1 - {}^c \theta_1$
v_3		${}^s \rho_2 - {}^c \rho_2$	${}^s \rho_2 - {}^c \rho_2$
v_4		${}^s \theta_2 - {}^c \theta_2$	${}^s \theta_2 - {}^c \theta_2$
v_5		${}^s k_{1b} - {}^c k_{1b} - d$	${}^s k_{1b} - {}^c k_{1b}$

TABLE 6.3 – Significations des composantes des sous-tâches référencées vision. d est une constante fixée ici à 30.

7.5 *cm*. Leur rayon est égal à 1 *cm*.

En outre, comme nous contrôlons maintenant l'orientation de la tête du PR2 et par conséquent l'orientation de la caméra fixe que nous utilisons, nous sommes dans la configuration décrite à la section 3.3.3 où la caméra déportée est dotée de degrés de liberté. Nous avons donc modifié la tâche d'évitement des butées pour y inclure les articulations de la tête. Les limites physiques de ces articulations, tirées du manuel du PR2, sont données dans le tableau 6.4.

Concernant maintenant la simulation elle-même, nous avons placé les bras dans la même

Articulation	-	+
head_pan_joint	-168°	168°
head_tilt_joint	-30°	60°

TABLE 6.4 – Limites des articulations de la tête du PR2.

configuration initiale que précédemment. Les articulations de la tête sont quant à elles initialement fixées aux valeurs indiquées dans la table 6.5. Enfin, les niveaux de bruit et les valeurs des paramètres intrinsèques des caméras restent inchangés.

La figure 6.9 présente l'évolution des tâches référencées vision. Nous pouvons observer que toutes les composantes des trois fonctions convergent vers zéro. Le stylo est donc

Articulation	Valeur
head_pan_joint	0.0 radians
head_tilt_joint	0.7 radians

TABLE 6.5 – Valeur initiale des articulations de la tête du PR2.

correctement refermé. Les courbes ne présentent ici aucun bruit apparent car nous avons choisi d'utiliser les indices visuels reconstruits après la première seconde.

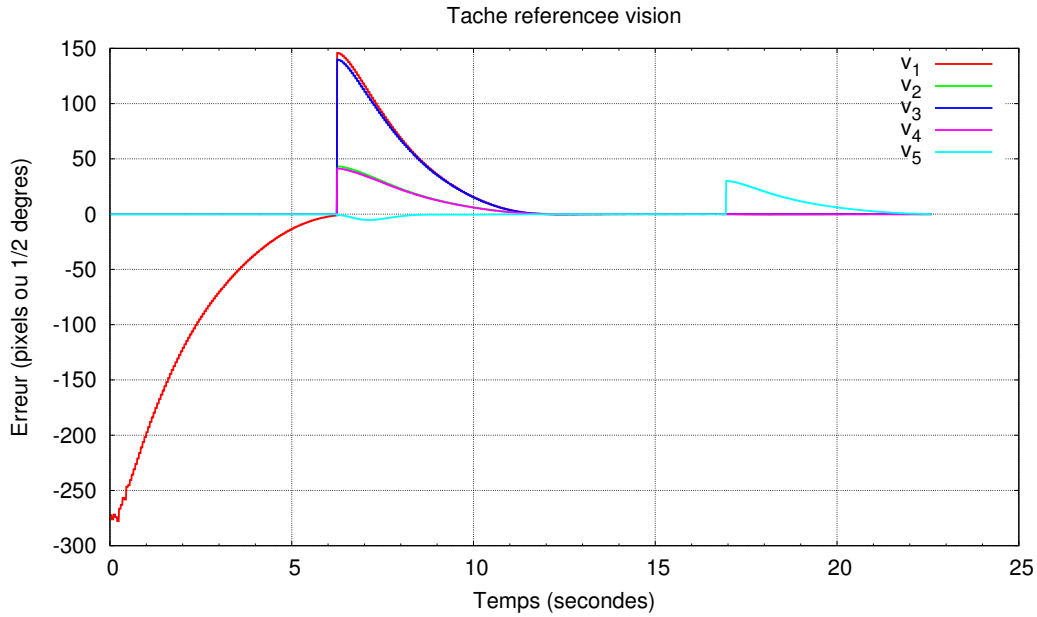


FIGURE 6.9 – Évolution de la tâche coordonnée référencée vision. Les composantes sont décrites dans le tableau 6.3.

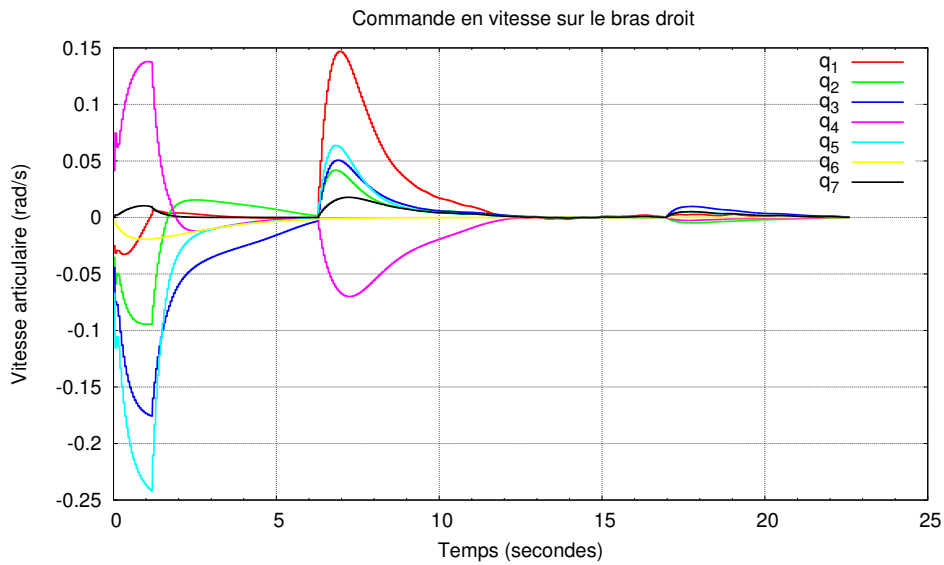


FIGURE 6.10 – Commande en vitesse sur le bras droit du PR2.

L'évolution de la fonction de tâche pour l'évitement des butées est similaire à celle présentée dans la figure 6.2. Nous ne la présentons donc pas ici pour ne pas alourdir une présentation des résultats déjà chargée. Concernant maintenant les commandes envoyées

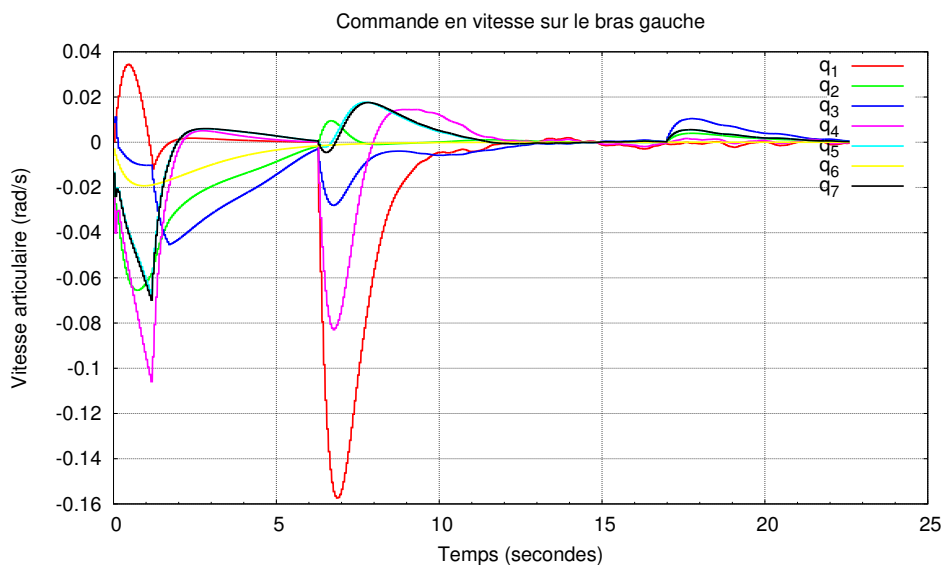


FIGURE 6.11 – Commande en vitesse sur le bras gauche du PR2.

aux deux bras, nous obtenons des résultats semblables à ceux que nous avons montrés précédemment. En effet, les vitesses des articulations sont similaires, signe de la coordination réalisée entre eux (cf. figures 6.10 et 6.11). Cependant, dans ce test, nous commandons en plus les degrés de liberté de la tête du PR2 pour mieux assurer la visibilité des indices visuels.

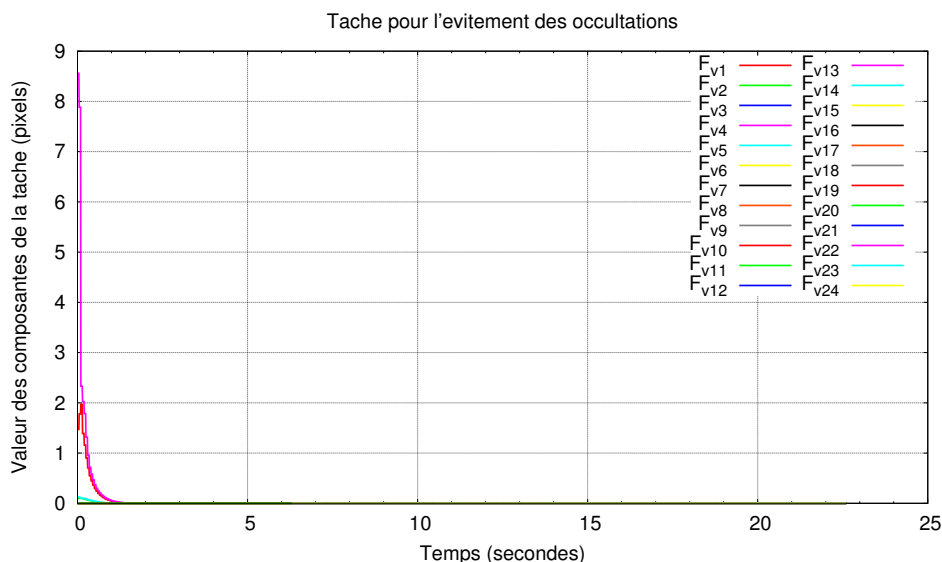


FIGURE 6.12 – Évolution de la tâche d'évitement des occultations

La figure 6.12 montre d'ailleurs l'évolution des composantes de la fonction de tâche définie dans ce but. C'est la tâche la moins prioritaire, ce qui fait que dans les précédentes simulations, son impact avait été faible. Ici, elle contraint le robot PR2 à tourner la tête

pour centrer les objets dans le champ de vue de la caméra. Les vitesses et les positions articulaires obtenues pour la tête sont représentées respectivement sur les figures 6.13 et 6.14. Lorsqu'elle est désactivée, la tête du robot reste immobile. Donc, grâce à notre formalisme pour la coordination par l'asservissement visuel, nous pouvons contrôler le "regard" du robot, et coordonner le mouvement de la tête avec celui des deux bras.

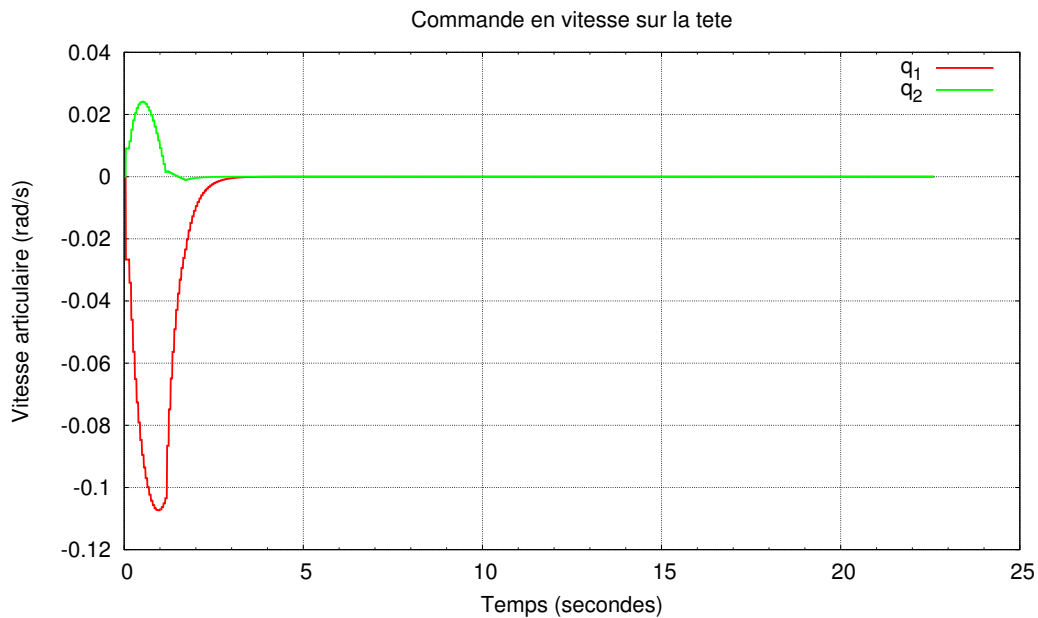


FIGURE 6.13 – Commande en vitesse des articulations de la tête.

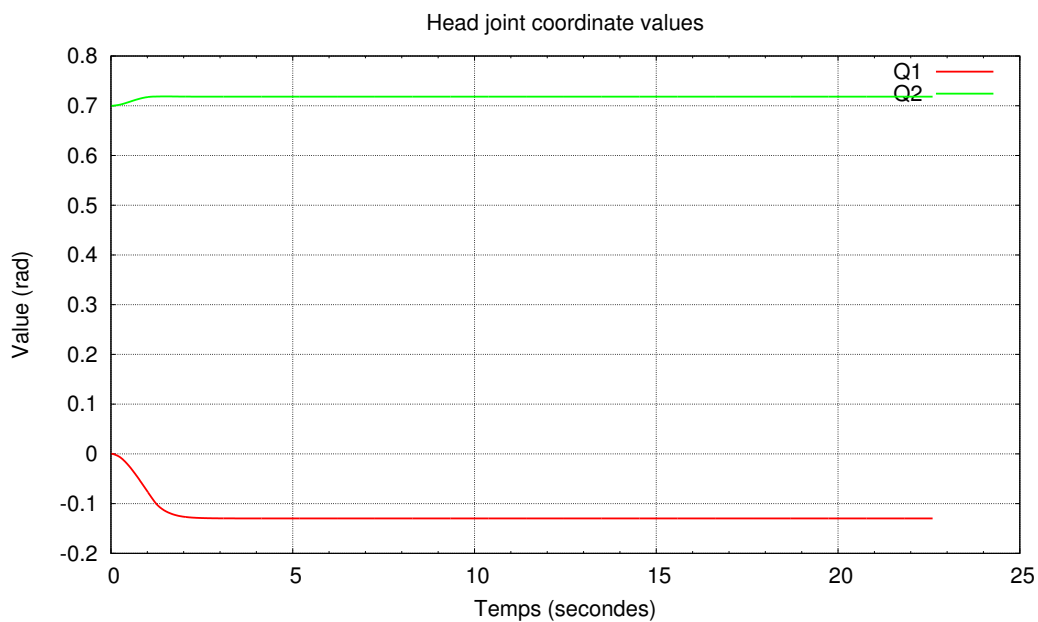


FIGURE 6.14 – Évolution des coordonnées articulaires de la tête.

La tâche référencée vision a de plus été réalisée en utilisant les indices reconstruits à partir de l'estimation de la structure 3D des cylindres. Nous rappelons que les deux cylindres et les deux caméras sont mobiles. Pour chaque cylindre, il nous faut estimer

trois paramètres : le rayon r , les coordonnées des centres des faces inférieure et supérieure B et T . Nous rappelons que le rayon des deux objets est de 1 cm . Les figures 6.15 à 6.20 matérialisent l'évolution de l'estimation de ces paramètres pendant la réalisation de la tâche de coordination. Grâce à l'utilisation des deux caméras, la précision des estimations est satisfaisante, elles convergent toutes vers les valeurs réelles constantes attendues, bien que tous les objets et les caméras bougent simultanément et malgré le niveau de bruit ajouté.

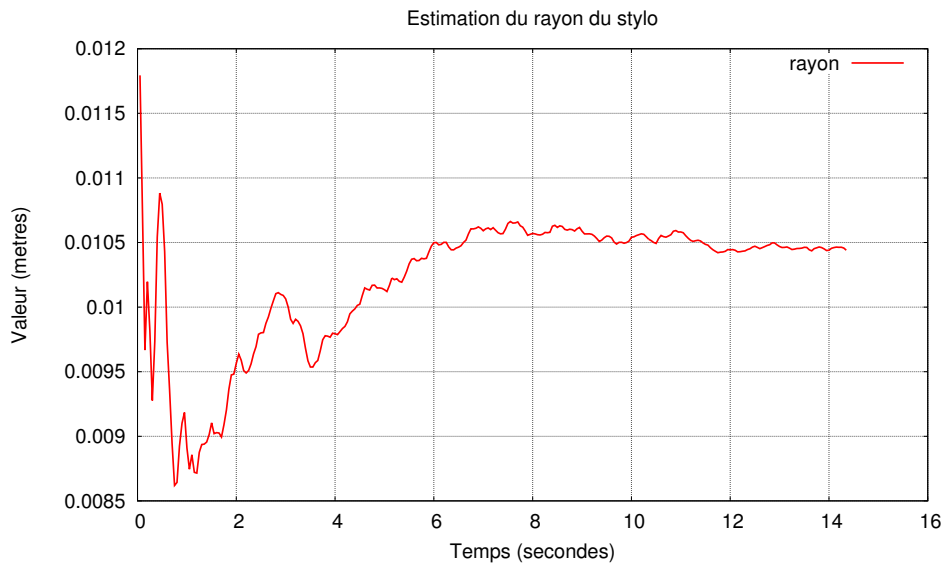


FIGURE 6.15 – Estimation du rayon du stylo, valeur réelle : 0.01 m .

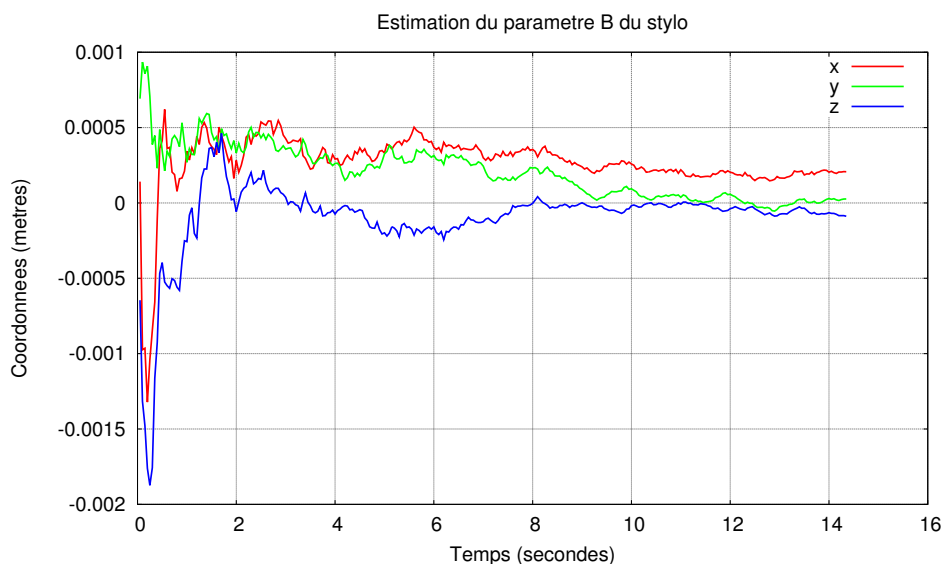


FIGURE 6.16 – Estimation de la base du stylo, valeurs réelles $x = 0\text{ m}$, $y = 0\text{ m}$, $z = 0\text{ m}$.

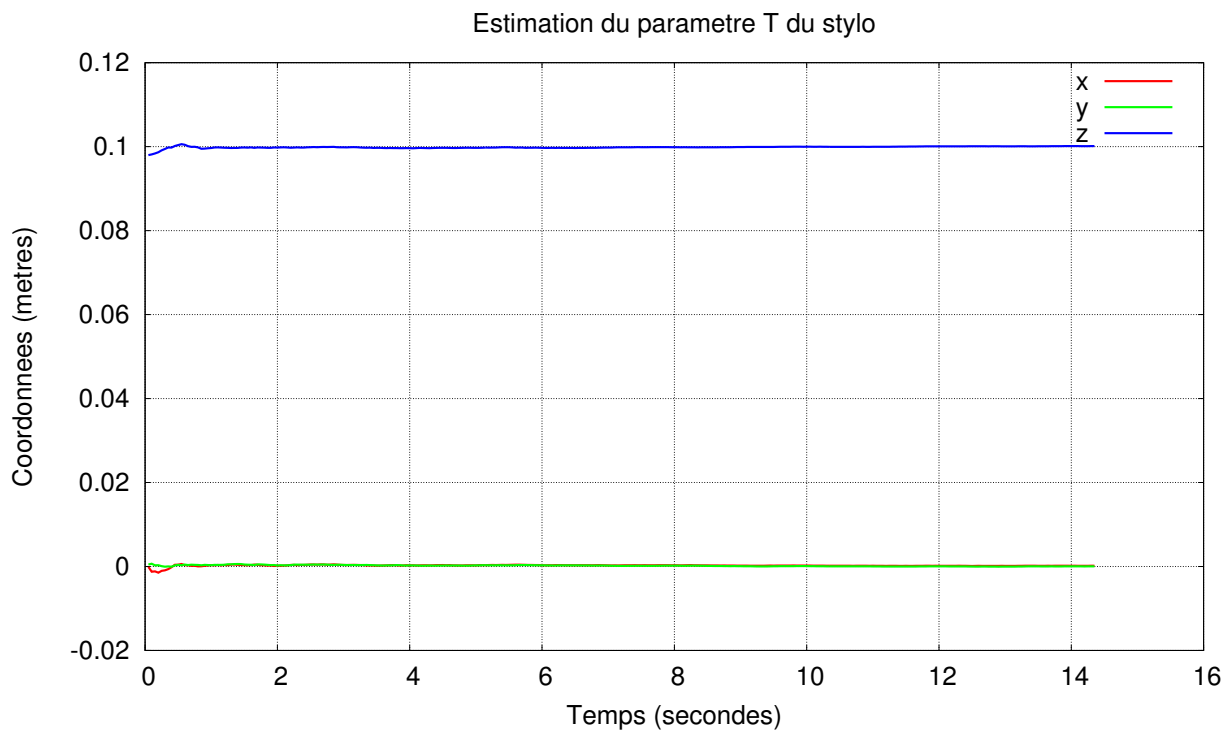


FIGURE 6.17 – Estimation du dessus du stylo, valeurs réelles $x = 0\text{ m}$, $y = 0\text{ m}$, $z = 0.1\text{ m}$.

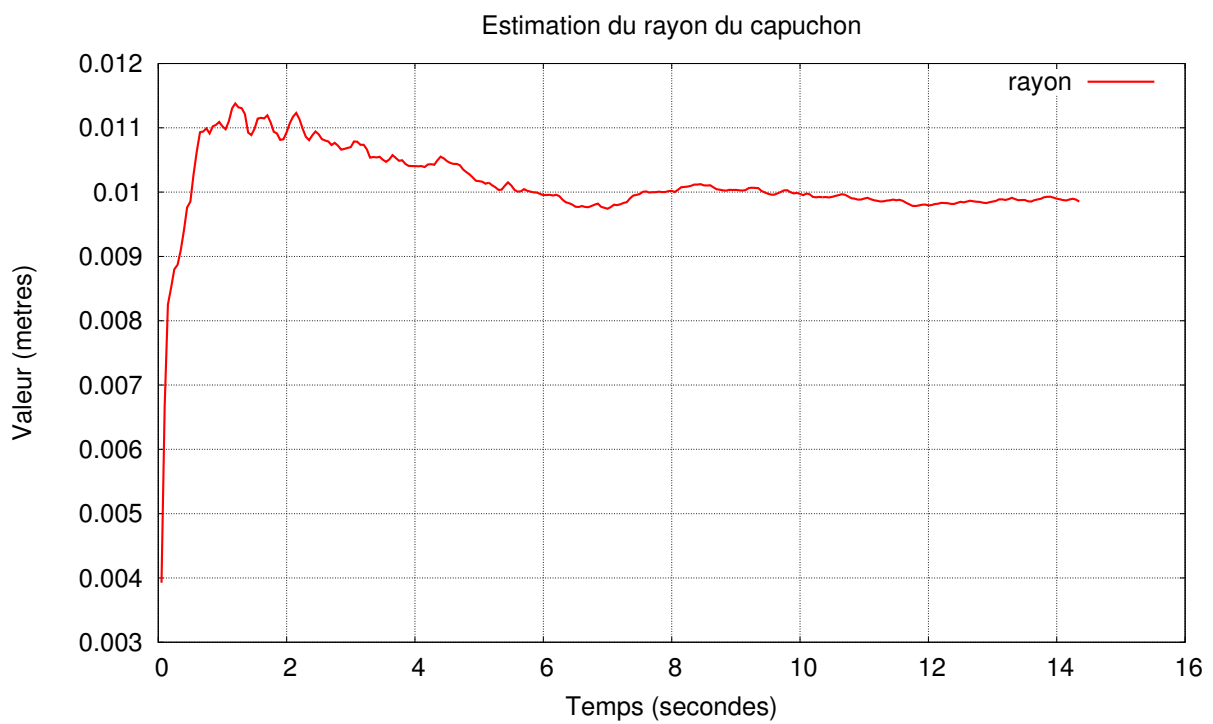


FIGURE 6.18 – Estimation du rayon du capuchon, valeur réelle 0.01 m .

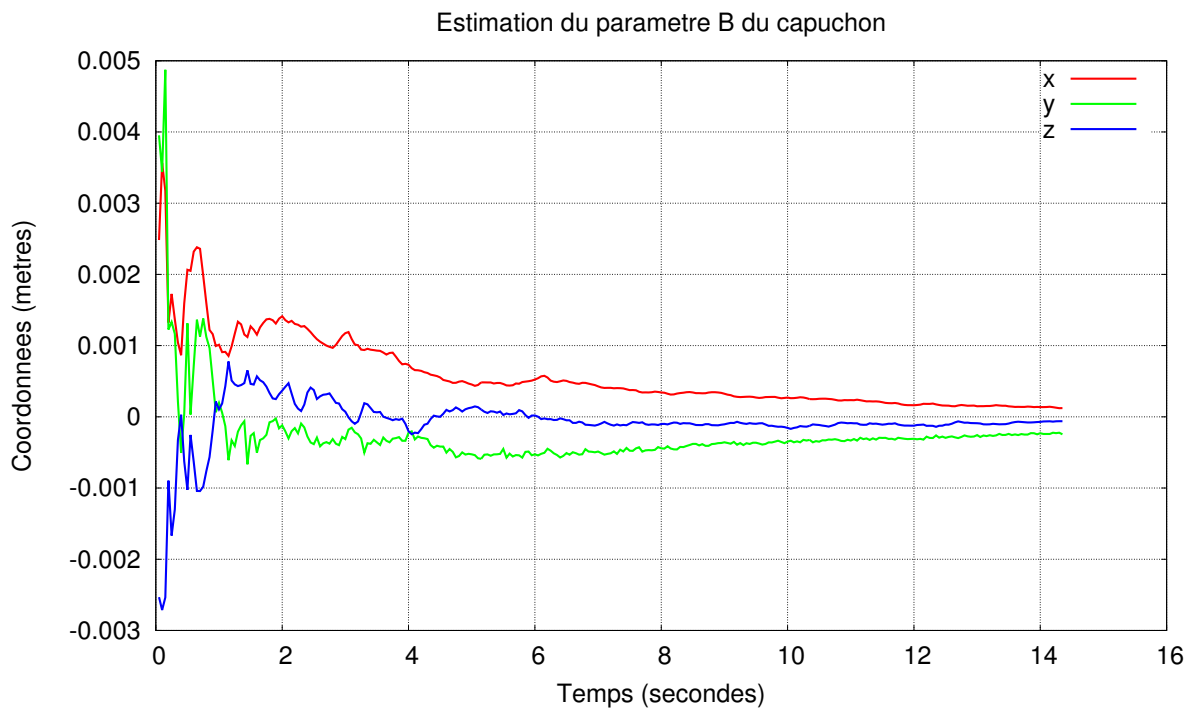


FIGURE 6.19 – Estimation de la base du capuchon, valeurs réelles $x = 0\text{ m}$, $y = 0\text{ m}$, $z = 0\text{ m}$.

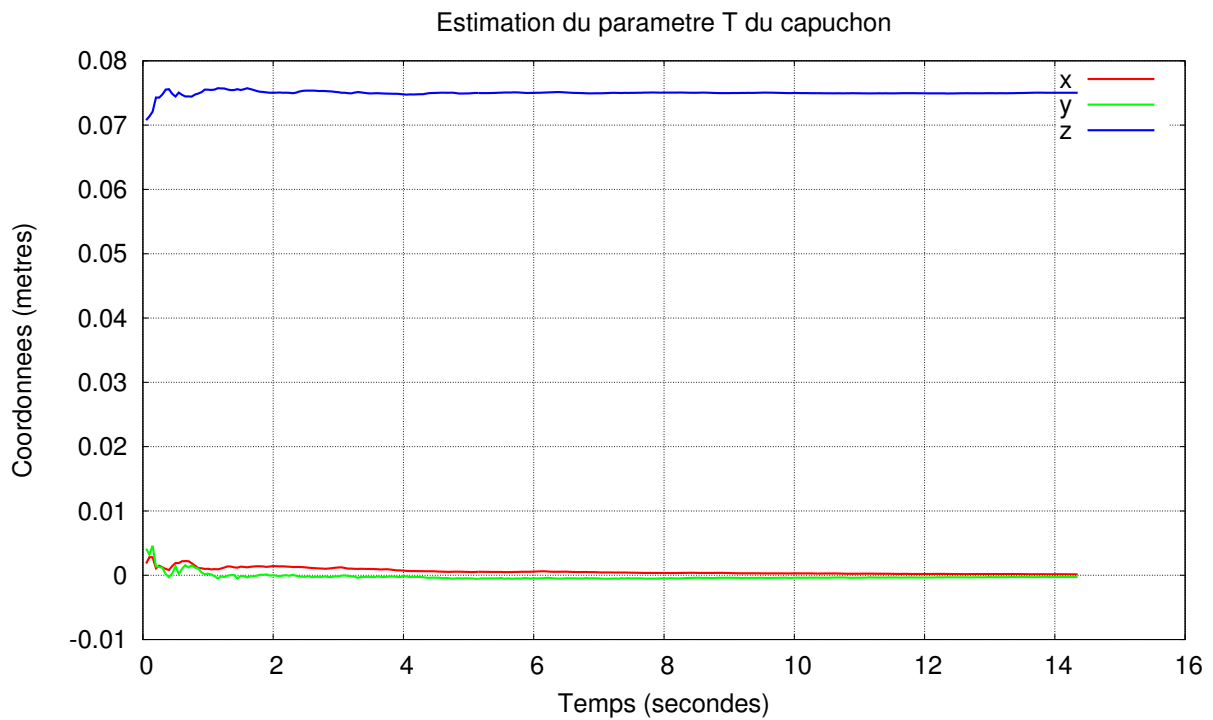


FIGURE 6.20 – Estimation du dessus du capuchon, valeurs réelles $x = 0\text{ m}$, $y = 0\text{ m}$, $z = 0.075\text{ m}$.

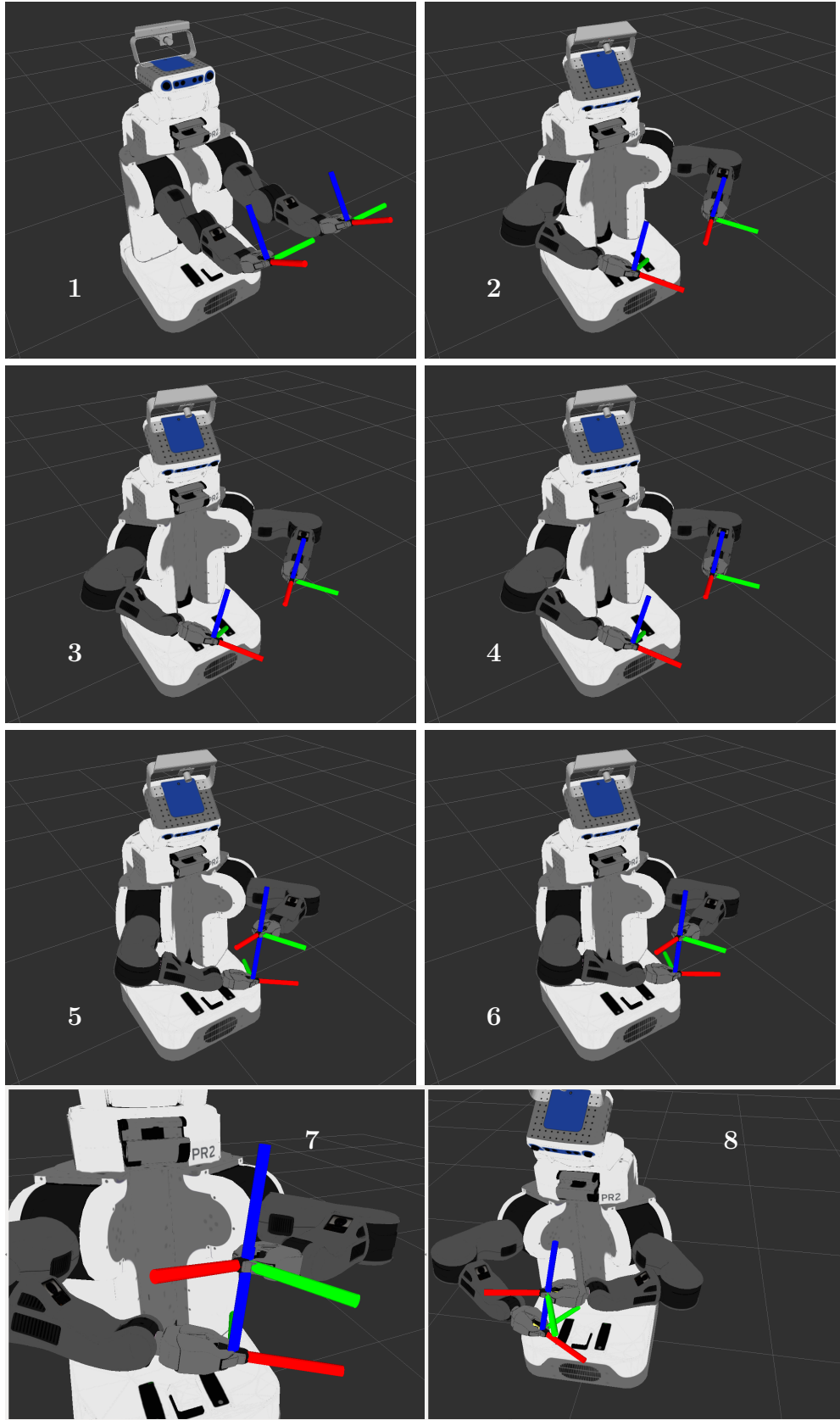


FIGURE 6.21 – Vues du robot PR2 effectuant la tâche de fermeture du stylo avec coordination des mouvements de la tête et des bras. La tâche démarre à la vignette 2.

Enfin, la figure 6.21 montre des vues du robot PR2 obtenues avec **rviz**, un outil de visualisation de ROS. Ces vues, rangées dans l'ordre chronologique, illustrent le comportement du robot PR2 réalisant la fermeture du stylo. Les repères liés au stylo et au capuchon sont aussi affichés avec l'axe X en rouge, l'axe Y en vert et l'axe Z en bleu. Le stylo et le capuchon sont orientés selon Z . L'on peut voir que les deux bras et la tête bougent, et qu'à la fin de la tâche, le stylo et le capuchon sont correctement alignés (les axes Z en bleu). Notre tâche référencée vision coordonnée est donc correctement réalisée.

6.4 Conclusion

Les résultats obtenus ont montré la pertinence des différentes approches que nous avons proposées : le formalisme pour la coordination par l'asservissement visuel, une gestion de la redondance préservée par la coordination, et l'estimation des structures 3D pour rendre robuste aux occultations la tâche référencée vision. Les résultats obtenus dans les simulations complexes que nous avons mises en œuvre, dans des conditions les plus proches possibles de conditions expérimentales, nous permettent d'espérer des résultats satisfaisants dans les expériences que nous sommes en train de mener. Nous pensons donc être en mesure de pouvoir réaliser la fermeture du stylo ainsi que d'autres tâches coordonnées.

Chapitre 7

Conclusion et perspectives

Veux-tu que je ferme la fenêtre ? Non,
laisse la fenêtre ouverte. Et demain,
sur la route, c'est moi qui parlerai...

La belle amour humaine, 2011
Lyonel Trouillot.

Tout au long de ce document, nous avons traité de la coordination de bras manipulateurs. Au chapitre 2 nous avons présenté un succinct état de l'art de la commande pour la coordination. Au chapitre 3, nous avons proposé un formalisme pour la coordination par la commande référencée vision. Notre approche préservant la redondance du système formé par les deux bras, nous nous sommes intéressés à la gestion de la redondance au chapitre 4. Nous avons défini une tâche d'évitement des butées et une autre tâche pour le maintien dans le champ de vue des caméras des objets manipulés. Pour rendre la tâche référencée vision robuste aux occultations, nous avons, dans le chapitre 5, estimé la structure 3D des objets manipulés pour reconstruire les indices visuels en cas de perte totale de l'image. Nous revisitons rapidement ces différents aspects dans les lignes qui suivent.

Nous avons d'abord précisé ce que nous entendions par coordination entre des bras manipulateurs dans ce document. Elle est, à notre sens, la prise en compte, dans la commande, de contraintes entre les bras. Cet aspect apparaît naturellement dans le cas de la comanipulation où les mouvements doivent être coordonnés pour réaliser la tâche et éviter des efforts inutiles ou dangereux. Mais la comanipulation n'est pas le seul cas où les bras doivent coopérer. En effet, la réalisation d'une tâche d'assemblage peut nécessiter la coordination des robots manipulateurs. On parle alors de tâche coordonnée asymétrique et il faut atteindre une pose relative donnée entre les objets pour la réaliser. Nous avons souhaité aborder le problème de coordination sous l'angle de la commande, en particulier celui de la commande référencée vision.

En analysant ce domaine, nous avons constaté que peu de travaux utilisant l'asservissement visuel ont abordé la problématique de la manipulation coordonnée. De plus, les solutions proposées dans la littérature pour répondre à ce problème ne nous ont pas convaincu car la coopération entre les bras restait restreinte. Nous avons alors voulu apporter notre contribution à cette problématique. Nous avons donc proposé un formalisme de la coordination basé sur la commande référencée vision. Ce formalisme offre un cadre permettant de synthétiser des asservissements visuels 2D pour l'exécution de tâches de manipulation coordonnée aussi bien symétriques qu'asymétriques. Différentes configurations de la caméra ont été envisagées et les lois de commande correspondantes synthétisées. Nous avons exploité ce formalisme pour réaliser la fermeture d'un stylo par un robot à

deux bras. Nous avons alors montré qu'il est possible, selon les besoins, d'exploiter les informations fournies par plusieurs capteurs visuels (embarqués ou déportés) pour définir les lois de commande, conduisant à des asservissements visuels multi-caméras. En outre, en considérant un système robotique unique, seuls les degrés de liberté nécessaires à l'exécution de la tâche ont été contraints. Nous avons donc laissé une certaine latitude aux bras du robot.

Cette latitude reflète la redondance du robot par rapport à notre tâche et se traduit par des degrés de liberté libres. Cette redondance permet au robot d'effectuer simultanément plusieurs tâches. De nombreuses approches ont été proposées pour la gestion de cette redondance. Elles varient de la pondération des vitesses articulaires qu'auraient imposé chacune des tâches séparément aux méthodes hiérarchiques. Parmi ces méthodes hiérarchiques couramment utilisées sur les robots ayant un grand nombre de liaisons, nous en avons sélectionné deux. La première, la plus utilisée, assure la réalisation de la tâche prioritaire, tandis que l'autre tente de réaliser au mieux les tâches secondaires. Cette dernière méthode permet d'utiliser l'information sur la vitesse résultante de la tâche prioritaire pour calculer la commande qui satisferait le mieux les tâches secondaires. Ce transfert d'information convient particulièrement aux tâches coordonnées que nous avons définies, parce qu'elles peuvent compenser les contraintes locales sur un bras, en adaptant les vitesses des autres bras. Mais cette méthode optimale présente des singularités gênantes. Nous avons identifié les situations où elle provoque des vitesses dangereuses, et avons décidé de basculer dans ces cas précis, sur la méthode classique. Notre gestion de la redondance réunit donc les avantages des deux méthodes. Nous avons pu valider notre approche sur l'exemple de la fermeture du stylo en imposant le respect des butées des articulations et le maintien des objets manipulés dans le champ de vue des caméras.

Or, cette dernière tâche, l'évitement des occultations, ne garantit pas que les objets seront toujours visibles ou distingués dans la scène. En effet, ces objets peuvent se cacher mutuellement ou être masqués par des éléments de la structure du robot. Les algorithmes de traitement d'images dédiés à leur détection et leur suivi peuvent ne pas fonctionner correctement. Dans ces conditions, la poursuite de la tâche référencée vision se trouve menacée. La solution que nous avons adoptée consiste à calculer l'image que devraient projeter les objets dans les caméras concernées. Pour ce faire, il est nécessaire de connaître la structure tridimensionnelle des objets manipulés. Nous avons donc développé des méthodes pour déterminer la structure des droites et des cylindres en nous inspirant d'une méthode existante pour les points. Nous avons choisi de nous placer dans les repères liés aux objets mobiles, de sorte que les paramètres à estimer soient constants. Les estimateurs que nous avons proposés intègrent les informations obtenues sur les objets à partir de toutes les images qui ont été fournies par les caméras pour améliorer l'estimation de la structure avec un faible coût de calcul. Lorsqu'une occultation survient, la position de l'objet étant connue, son image est calculée et utilisée pour réaliser la tâche référencée vision. Nous avons montré que l'erreur entre l'image réelle de l'objet et celle reconstruite diminuait avec le nombre de prises de vues et était suffisamment faible pour assurer la réalisation de la tâche référencée vision. Nous avons donc intégré cette solution pour la réalisation de notre tâche pour la fermeture du stylo. L'ensemble de nos contributions a été validé dans de nombreuses simulations dans des conditions proches des conditions réelles.

Cette thèse nous a permis d'entrevoir l'étendue du travail qui reste à accomplir. Les travaux que nous avons pu réaliser sont du reste perfectibles et les contributions proposées ne résolvent pas tous les problèmes que nous avons abordés. Nous pensons notamment

que notre formalisme pour la coordination par la commande référencée vision ne traite qu'un aspect du problème. Une gestion des forces devrait y être ajoutée [Uchiyama and Dauchez, 1988, Chiacchio et al., 1996]. L'utilisation d'une technique de commande hybride vision/force telle que [Malis et al., 2001, Dauchez et al., 2005, Mezouar et al., 2007] pourrait répondre à ce problème. La commande par impédance pourrait aussi être une bonne candidate car elle permet de modéliser les contacts entre les organes terminaux et les objets manipulés.

En outre, nous souhaitons établir une preuve de la stabilité du système lorsque la commande est synthétisée à partir de notre algorithme de gestion de la redondance. Il nous semble aussi important de mieux analyser la continuité de la commande lors de la transition entre les tâches [Lee et al., 2012, Mansard and Khatib, 2008]. Nous pensons aussi qu'une intégration de notre approche avec les systèmes de haut niveau tels que la pile de tâches [Mansard and Chaumette, 2007] est souhaitable. Dans la même idée, rapprocher nos travaux de ceux concernant la problématique du mouvement corps complet serait aussi intéressant [Mansard et al., 2007]. Outre cela, la modélisation des différentes contraintes, notamment les butées et les occultations devraient être approfondies ; à travers entre autres la notion d'évitement passif et actif [Kermorgant, 2011, Kermorgant and Chaumette, 2011a, Remazeilles et al., 2006]. Enfin, la problématique des collisions avec l'environnement doit être abordée afin de garantir pleinement la sécurité du robot.

La reconstruction des structures à partir du mouvement est un problème ouvert. Nous avons proposé des méthodes pour la reconstruction des droites et des cylindres que nous sommes en train d'adapter pour traiter le cas de la sphère. Nous suivons de plus quelques pistes pour améliorer nos techniques. Tout d'abord, nous pensons pouvoir, dans leur version récursive, adapter nos estimateurs pour minimiser l'erreur angulaire [Recker et al., 2013]. Il nous semble également que les méthodes de vision active [Chaumette et al., 1996, De Luca et al., 2007, Spica and Robuffo Giordano, 2013, Spica et al., 2014c] devraient pouvoir être appliquées à nos estimateurs. Dans ce but, nous étudions la structure des matrices que nous inversons pour déterminer le mouvement à appliquer pour améliorer la reconstruction. Ce mouvement peut être défini comme une tâche secondaire à la tâche de coordination [Spica et al., 2014b]. Enfin, nous avons pris comme hypothèse que les caméras étaient calibrées et que leur mouvement ainsi que ceux des objets étaient connus. Cependant, pour aller plus loin, la robustesse aux erreurs de calibration et aux erreurs de pose causés par les mesures des codeurs et la flexibilité des articulations serait intéressante à étudier. La détection des pannes et des mesures aberrantes sur l'image serait aussi utile.

Enfin, il nous semble fondamental de valider expérimentalement ces approches sur le robot PR2 du LAAS. Nous y travaillons actuellement. Ces résultats expérimentaux viendront compléter les solutions théoriques développées, avec le souhait d'avoir fait œuvre qui vaille dans le domaine de la coordination de bras manipulateurs par asservissement visuel.

Annexe A

Détermination des différentes matrices d'interaction

Dans cette annexe, nous présentons le calcul des indices visuels utilisés tout au long du document et des matrices d'interaction correspondantes. L'expression de la plupart de ces matrices peuvent être retrouvées dans [Chaumette, 1990, Espiau et al., 1992, Chaumette, 1994]. Elles sont données ici pour le confort du lecteur.

A.1 Matrice d'interaction du point

Le motif le plus simple que l'on utilise en asservissement visuel 2D est le point. L'image d'un point P fixe de coordonnées (X, Y, Z) dans le repère de la caméra F_c est un point M de coordonnées (x, y) dans le plan image. x et y sont donnés par :

$$\begin{cases} x = fX/Z \\ y = fY/Z \end{cases}$$

où f est la focale de la caméra. Si l'on dérive par rapport au temps, on obtient :

$$\begin{cases} \dot{x} = f\dot{X}/Z - fX\dot{Z}/Z^2 = (f\dot{X} - x\dot{Z})/Z \\ \dot{y} = f\dot{Y}/Z - fY\dot{Z}/Z^2 = (f\dot{Y} - y\dot{Z})/Z. \end{cases}$$

Le point étant fixe, c'est la caméra qui bouge et l'on doit calculer la vitesse relative du point par rapport à la caméra dans le repère F_c :

$$\overrightarrow{V_{F_c}}(M) = -\overrightarrow{V_{F_w}}(C) - \overrightarrow{\Omega}(F_c/F_w) \wedge \overrightarrow{CM}$$

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = - \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} - \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \wedge \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\dot{X} = -v_x - \omega_y Z + \omega_z Y$$

$$\dot{Y} = -v_y - \omega_z X + \omega_x Z$$

$$\dot{Z} = -v_z - \omega_x Y + \omega_y X$$

En remplaçant \dot{X} , \dot{Y} , et \dot{Z} par leur expression dans le calcul de \dot{x} et de \dot{y} on obtient :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -f/Z & 0 & x/Z & xy/f & -(1+x^2)/f & y \\ 0 & -f/Z & y/Z & (1+y^2)/f & -xy/f & -x \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Le vecteur $[v_x \dots \omega_z]^T$ représente le torseur cinématique de la caméra par rapport à un repère fixe et exprimé dans F_c . La matrice d'interaction du point est donnée par [Chaumette, 1990] :

$$L_{xy} = \begin{bmatrix} -f/Z & 0 & x/Z & xy/f & -(1+x^2)/f & y \\ 0 & -f/Z & y/Z & (1+y^2)/f & -xy/f & -x \end{bmatrix} \quad (\text{A.1})$$

Dans la plupart des documents, la focale est prise égale à 1 sans perte de généralité [Chaumette, 1990].

A.2 Matrice d'interaction de la droite

Nous avons utilisé la paramétrisation polaire pour représenter les droites dans l'image. Les indices visuels sont donc ρ et θ (cf. figure 3.13). Les points dans l'image appartenant à la droite dans l'image respectent l'équation :

$$x \cos \theta + y \sin \theta - \rho = 0$$

Calculons la dérivée par rapport au temps de cette équation :

$$\dot{x} \cos \theta - x \dot{\theta} \sin \theta + \dot{y} \sin \theta + y \dot{\theta} \cos \theta - \dot{\rho} = 0$$

Soient deux points distincts dans l'image $A(x_a, y_a)$ et $B(x_b, y_b)$ appartenant à cette droite. Notons L_a et L_b leur matrices d'interaction. Il vient :

$$\begin{aligned} \dot{x}_a \cos \theta + \dot{y}_a \sin \theta - (x_a \sin \theta - y_b \cos \theta) \dot{\theta} - \dot{\rho} &= 0 \\ \dot{x}_b \cos \theta + \dot{y}_b \sin \theta - (x_b \sin \theta - y_b \cos \theta) \dot{\theta} - \dot{\rho} &= 0 \end{aligned}$$

$$\begin{bmatrix} 1 & (x_a \sin \theta - y_b \cos \theta) \\ 1 & (x_b \sin \theta - y_b \cos \theta) \end{bmatrix} \begin{bmatrix} \dot{\rho} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \end{bmatrix} \begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{x}_b \\ \dot{y}_b \end{bmatrix}$$

La matrice d'interaction $L_{\rho\theta}$ correspondant au vecteur des indices visuels $[\rho \ \theta]$ peut alors s'exprimer ainsi [Chaumette, 2002] :

$$L_{\rho\theta} = \begin{bmatrix} 1 & (x_a \sin \theta - y_b \cos \theta) \\ 1 & (x_b \sin \theta - y_b \cos \theta) \end{bmatrix}^{-1} \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \end{bmatrix} \begin{bmatrix} L_a \\ L_b \end{bmatrix} \quad (\text{A.2})$$

A.3 Matrice d'interaction du segment

Nous avons introduit un indice visuel k , pour exprimer la position d'un point sur une droite de l'image. En notant E (x_e, y_e) ce point, k s'écrit :

$$k = y_e \cos \theta - x_e \sin \theta$$

En dérivant par rapport au temps, l'on obtient :

$$\dot{k} = \dot{y}_e \cos \theta - \dot{x}_e \sin \theta - y_e \dot{\theta} \sin \theta - x_e \dot{\theta} \cos \theta$$

$$\dot{k} = \begin{bmatrix} -\sin \theta & \cos \theta & 0 & -(x_e \cos \theta + y_e \sin \theta) \end{bmatrix} \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\rho} \\ \dot{\theta} \end{bmatrix}$$

La matrice d'interaction L_k pour l'indice visuel k , s'exprime grâce aux formules précédentes, avec L_e la matrice d'interaction du point E :

$$L_k = \begin{bmatrix} -\sin \theta & \cos \theta & 0 & -(x_e \cos \theta + y_e \sin \theta) \end{bmatrix} \begin{bmatrix} L_e \\ L_{\rho\theta} \end{bmatrix} \quad (\text{A.3})$$

Les segments de droite peuvent être décrits avec le vecteur d'indices visuels suivant :

$$s = [\rho \quad \theta \quad k_1 \quad k_2]^T$$

où k_1 et k_2 correspondent aux extrémités du segment.

A.4 Indices visuels et matrice d'interaction pour le cylindre

Considérons un cylindre de longueur infinie dans le repère de la caméra F_c . Les coordonnées dans F_c de P , le point le plus proche de l'origine de ce repère, sont notées cP . Dans ce même repère, les composantes du vecteur directeur de l'axe du cylindre sont désignées par cU , tandis que le rayon est représenté par r . On a :

$${}^cU^T {}^cP = 0 \quad \|{}^cU\| = 1$$

Le cylindre peut être représenté dans l'image par deux droites.

$$S_{cyl} = [\rho_1 \quad \theta_1 \quad \rho_2 \quad \theta_2]^T$$

Calculons les indices visuels à partir des variables cP , cU et r . Nous définissons ces variables :

$$d = \|{}^cP\| \quad s_\alpha = \frac{r}{d} = \sin \alpha \quad c_\alpha = \frac{1}{d} \sqrt{d^2 - r^2} = \cos \alpha$$

$${}^cN = c_\alpha \frac{1}{d^2} ({}^cU \wedge {}^cP) - s_\alpha \frac{1}{d} {}^cP$$

$${}^cQ = c_\alpha \frac{1}{d^2} ({}^cU \wedge {}^cP) + s_\alpha \frac{1}{d} {}^cP$$

d est la distance entre l'origine du repère F_c et l'axe du cylindre. α est la moitié de l'angle sous lequel le cylindre est perçu par la caméra. N et Q sont des normales au cylindre au tangentes passant par l'origine de F_c . Toutes ces variables sont illustrées sur la figure A.1.

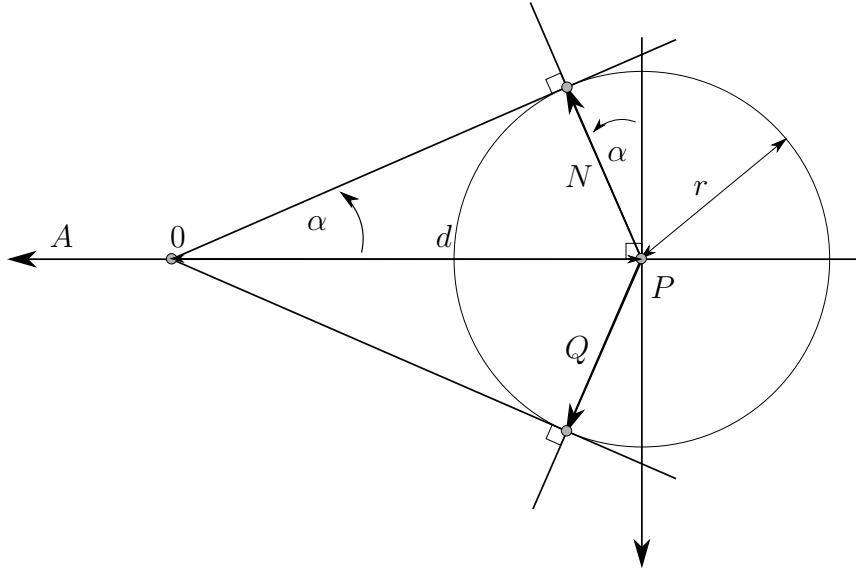


FIGURE A.1 – Vue de haut du cylindre et de la caméra

Les droites de l'image sont définies par l'intersection des plans de normale N et Q passant par l'origine du repère F_c et du plan de l'image. Si (a, b, c) et (g, h, i) sont les composantes de cN et cQ , l'on peut déduire les indices visuels du cylindre :

$$\begin{aligned} \rho_1 &= -\frac{cf}{\sqrt{a^2 + b^2}} & \theta_1 &= \arctan(b, a) \\ \rho_2 &= -\frac{if}{\sqrt{g^2 + h^2}} & \theta_2 &= \arctan(g, i) \end{aligned}$$

avec f la focale de la caméra. La matrice d'interaction du cylindre pourrait être donnée par :

$$L_{cyl} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}$$

où L_1 et L_2 sont les matrices d'interaction correspondant aux indices visuels ρ_1 , θ_1 et ρ_2 , θ_2 . Mais ce serait oublier la géométrie du cylindre. Les deux droites changent avec le point de vue de la caméra sur le cylindre. Nous allons modifier les matrices d'interaction pour prendre en compte deux aspects : l'angle α varie en fonction de la vitesse radiale par rapport au cylindre, et l'image perçue du cylindre est constante si le cylindre tourne autour de son axe. Définissons ces variables :

$$\beta = (r^2 - d^2)/d^2 \quad \lambda = (1/r - r/d^2)\sqrt{d^2 - r^2}$$

β et λ sont des paramètres que l'on retrouve lorsque qu'on calcule la dérivée des normales par rapport à la distance d entre l'axe du cylindre et l'origine du repère F_c . Nous déterminons un repère F_{cyl} dont l'origine est P et dont un des axes est le vecteur U . Dans ce repère, la symétrie du cylindre est plus facile à exprimer.

$$\begin{aligned} A &= -\frac{1}{d} {}^cP & B &= {}^cU \wedge A \\ {}^cR_{cyl} &= \begin{bmatrix} A & B & {}^cU \end{bmatrix} & {}^cW_{cyl} &= \begin{bmatrix} {}^cR_{cyl} & [{}^cP]_{\wedge} \times {}^cR_{cyl} \\ 0_{3 \times 3} & {}^cR_{cyl} \end{bmatrix} \end{aligned}$$

Nous définissons ces matrices pour rendre compte de la symétrie du cylindre, et du changement de l'angle α .

$$D_1 = \begin{bmatrix} 1 + \beta & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad D_2 = \begin{bmatrix} 1 + \beta & 0 & \\ 0 & -\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$I_{cyl} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

L'expression de la matrice d'interaction du cylindre devient :

$$\tilde{L}_1 = L_1 \times {}^cW_{cyl}^{-1} \times \begin{bmatrix} D_1 & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 \end{bmatrix} \begin{bmatrix} I_{cyl} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{cyl} \end{bmatrix} \times {}^cW_{cyl} \quad (\text{A.4})$$

$$\tilde{L}_2 = L_2 \times {}^cW_{cyl}^{-1} \times \begin{bmatrix} D_2 & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 \end{bmatrix} \begin{bmatrix} I_{cyl} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{cyl} \end{bmatrix} \times {}^cW_{cyl} \quad (\text{A.5})$$

$$L_{cyl} = \begin{bmatrix} \tilde{L}_1 \\ \tilde{L}_2 \end{bmatrix} \quad (\text{A.6})$$

A.5 Extension au cas du cylindre de longueur finie

Pour le cas du cylindre de longueur infinie, l'image est décrite par deux segments de droite. Nous notons P_1^a , P_1^b , P_2^a et P_2^b les points sur le cylindre correspondant à leurs extrémités respectives. Le vecteur des indices visuels retenu a été :

$$S_{cyl} = [\rho_1 \quad \theta_1 \quad k_1^a \quad k_2^b \quad \rho_2 \quad \theta_2 \quad k_1^a \quad k_2^b]^T$$

k_1^a , k_1^b , k_2^a et k_2^b correspondent aux extrémités des deux segments. Retenons les paramètres B , et T comme le centre des faces supérieures et inférieure du cylindre 5. Nous avons alors :

$${}^cP_1^a = {}^cT + r \times {}^cN \quad {}^cP_1^b = {}^cB + r \times {}^cN \quad {}^cP_2^a = {}^cT + r \times {}^cQ \quad {}^cP_2^b = {}^cB + r \times {}^cQ$$

Les matrices d'interaction correspondantes à ces points sont données par :

$$\tilde{L}_1^a = L_1^a \times {}^cW_{cyl}^{-1} \times \begin{bmatrix} D_1 & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 \end{bmatrix} \begin{bmatrix} I_{cyl} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{cyl} \end{bmatrix} \times {}^cW_{cyl} \quad (\text{A.7})$$

$$\tilde{L}_1^b = L_1^b \times {}^cW_{cyl}^{-1} \times \begin{bmatrix} D_1 & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 \end{bmatrix} \begin{bmatrix} I_{cyl} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{cyl} \end{bmatrix} \times {}^cW_{cyl} \quad (\text{A.8})$$

$$\tilde{L}_2^a = L_2^a \times {}^cW_{cyl}^{-1} \times \begin{bmatrix} D_2 & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 \end{bmatrix} \begin{bmatrix} I_{cyl} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{cyl} \end{bmatrix} \times {}^cW_{cyl} \quad (\text{A.9})$$

$$\tilde{L}_2^b = L_2^b \times {}^cW_{cyl}^{-1} \times \begin{bmatrix} D_2 & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 \end{bmatrix} \begin{bmatrix} I_{cyl} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{cyl} \end{bmatrix} \times {}^cW_{cyl} \quad (\text{A.10})$$

Les matrices L_1^a , L_1^b , L_2^a et L_2^b sont données par l'équation (A.1). En utilisant ensuite les équations (A.3) et (A.2) avec ces matrices modifiées, la matrice d'interaction complète du cylindre de longueur finie peut être calculée.

Bibliographie

- [Adorno, 2011] Adorno, B. (2011). *Two-arm Manipulation : From Manipulators to Enhanced Human-Robot Collaboration*. PhD thesis, Université Montpellier II. 2.2.1, 2.2.2, 3.3
- [Adorno et al., 2011] Adorno, B., Bo, A., Fraitse, P., and Poignet, P. (2011). Towards a cooperative framework for interactive manipulation involving a human and a humanoid. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3777–3783. 2.2.1
- [Adorno et al., 2010] Adorno, B., Fraitse, P., and Druon, S. (2010). Dual position control strategies using the cooperative dual task-space framework. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3955–3960. 2.1, 2.1, 2.2.1, 2.2.2, 3.3.1.1, 3.3.1.2
- [Agravante et al., 2013] Agravante, D., Pages, J., and Chaumette, F. (2013). Visual servoing for the reem humanoid robot’s upper body. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5253–5258. 3.3.3
- [Al-Jarrah et al., 1995] Al-Jarrah, O., Zheng, Y., and Yi, K.-Y. (1995). Efficient trajectory planning for two manipulators to deform flexible materials with experiments. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 312–317 vol.1. 2.1, 2.2
- [Al-Yahmadi and Hsia, 2007] Al-Yahmadi, A. S. and Hsia, T. (2007). Modeling and control of two manipulators handling a flexible beam. *International Journal of Electrical, Electronic Science and Engineering*, 1(6) :122 – 125. 2.1, 2.2.2
- [Albrichsfeld, 1996] Albrichsfeld, C. V. (1996). Self-adjusting active compliance controller for two cooperating robots handling a flexible object. 2.2.3
- [Albrichsfeld and Tolle, 2002] Albrichsfeld, C. V. and Tolle, H. (2002). A self-adjusting active compliance controller for multiple robots handling an object. *Control Engineering Practice*, 10(2) :165 – 173. 2.2.3
- [Allen et al., 1993] Allen, P., Timcenko, A., Yoshimi, B., and Michelman, P. (1993). Automated tracking and grasping of a moving object with a robotic hand-eye system. *Robotics and Automation, IEEE Transactions on*, 9(2) :152–165. 3.2
- [Armar, 2015] Armar (2015). Robots armar. <http://his.anthropomatik.kit.edu/english/241.php>. Accessed 9 September 2015. 1
- [Asimo, 2015] Asimo (2015). Asimo honda. <http://asimo.honda.com/>. 1
- [Baerlocher and Boulic, 2004] Baerlocher, P. and Boulic, R. (2004). An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20 :402–417. 4.2.2, 4.2.3
- [Bartoli and Sturm, 2005] Bartoli, A. and Sturm, P. (2005). Structure-from-motion using lines : Representation, triangulation and bundle adjustment. *Computer Vision and Image Understanding*, 100 :2005. 5.2

- [Beardsley et al., 1997] Beardsley, P. A., Zisserman, A., and Murray, D. W. (1997). Sequential updating of projective and affine structure from motion. *Int. J. Comput. Vision*, 23(3) :235–259. 5.2, 5.2, 5.3, 5.4.1
- [Berry, 1999] Berry, F. (1999). *Contournement d’objet par asservissement visuel*. PhD thesis, Université Blaise Pascal - Clermont II. 3.4.2
- [Berry et al., 2000] Berry, F., Martinet, P., and Gallice, J. (2000). Turning around an unknown object using visual servoing. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 1, pages 257–262 vol.1. 3.4.2, 3.4.2
- [Bird, 2015] Bird, S. (2015). Smart bird festo. http://www.festo.com/cms/en_corp/11369.htm. Accessed 9 September 2015. 1
- [Boissinot, 2008] Boissinot, D. (2008). Asservissement visuel coordonné pour un manipulateur mobile à deux bras. Mémoire de stage. 3.3.5
- [Bonitz and Hsia, 1993] Bonitz, R. and Hsia, T. (1993). Internal force-based impedance control for cooperating manipulators. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 944–949 vol.3. 2.2.3
- [Bonitz and Hsia, 1996] Bonitz, R. and Hsia, T. (1996). Robust internal-force based impedance control for coordinating manipulators-theory and experiments. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 622–628 vol.1. 2.1, 2.2.3
- [Caccavale et al., 1999] Caccavale, F., Chiacchio, P., and Chiaverini, S. (1999). Analysis of a task-space regulator for cooperative manipulators. In *American Control Conference, 1999. Proceedings of the 1999*, volume 3, pages 1945–1949 vol.3. 2.2.2
- [Caccavale et al., 2000] Caccavale, F., Chiacchio, P., and Chiaverini, S. (2000). Task-space regulation of cooperative manipulators. *Automatica*, 36(6) :879 – 887. 2.2.2
- [Caccavale et al., 2008] Caccavale, F., Chiacchio, P., Marino, A., and Villani, L. (2008). Six-dof impedance control of dual-arm cooperative manipulators. volume 13, pages 576–586. 2.2.3
- [Caccavale et al., 2001] Caccavale, F., Ciro, N., Siciliano, B., and Villani, L. (2001). Achieving a cooperative behavior in a dual-arm robot system via a modular control structure. *Journal of Robotic Systems*, 18(12) :691–699. 2.1, 2.2.1
- [Caccavale and Uchiyama, 2008] Caccavale, F. and Uchiyama, M. (2008). Cooperative manipulators. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, pages 701–718. Springer Berlin Heidelberg. 2.1, 2.2.2
- [Cadenat, 1999] Cadenat, V. (1999). *Commande référencée multi-capteurs pour la navigation d’un robot mobile*. PhD thesis, Université Paul Sabatier. 4.1, 4.2.1
- [Cadenat et al., 2001] Cadenat, V., Soueres, P., and Courdesses, M. (2001). Using system redundancy to perform a sensor-based navigation task amidst obstacles. *International Journal of Robotics & Automation*, 16(2) :61–73. 4.2.1
- [Chaumette, 1990] Chaumette, F. (1990). *La relation vision-commande : théorie et application à des tâches robotiques*. PhD thesis, Université de Rennes 1. 3.1, 3.3.1.2, 3.3.4, 5.1, 5.5.1, 5.5.2, 5.5.2.1, A, A.1, A.1
- [Chaumette, 1994] Chaumette, F. (1994). Visual servoing using image features defined upon geometrical primitives. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 4, pages 3782–3787 vol.4. 5.5.2.1, A
- [Chaumette, 2002] Chaumette, F. (2002). *La commande des robots manipulateurs : Asservissement visuel*, chapter 3, pages 101 – 151. 3.2, 3.4.2, A.2

- [Chaumette et al., 1996] Chaumette, F., Boukir, S., Bouthemy, P., and Juvin, D. (1996). Structure from controlled motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(5) :492–504. 5.2, 7
- [Chaumette and Hutchinson, 2006] Chaumette, F. and Hutchinson, S. (2006). Visual servo control part 1 : Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4) :82–90. 2.2.4, 2.3, 3.1
- [Chaumette and Hutchinson, 2007] Chaumette, F. and Hutchinson, S. (2007). Visual servo control part ii : Advanced approaches. *IEEE Robotics and Automation Magazine*, pages 109 –118. 2.2.4, 3.1, 5.1
- [Chaumette and Rives, 1990] Chaumette, F. and Rives, P. (1990). Vision-based-control for robotic tasks. In *Intelligent Motion Control, 1990. Proceedings of the IEEE International Workshop on*, volume 2, pages 395–400. 5.5.1
- [Chaumette et al., 1991] Chaumette, F., Rives, P., and Espiau, B. (1991). Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *Robotics and Automation. Proceedings., 1991 IEEE International Conference on*, pages 2248–2253 vol.3. 3.2
- [Chiacchio et al., 1996] Chiacchio, P., Chiaverini, S., and Siciliano, B. (1996). Direct and inverse kinematics for coordinated mot tasks of a two-manipulator system. *Journal of Dynamic Systems, Measurement, and Control*, 118 :691–697. 2.2.2, 2.2.2, 3.3, 3.3.1.1, 3.3.1.2, 7
- [Chiaverini, 1997] Chiaverini, S. (1997). Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *Robotics and Automation, IEEE Transactions on*, 13(3) :398–410. 4.2.1, 4.2.2
- [Cohen et al., 2012] Cohen, B., Chitta, S., and Likhachev, M. (2012). Search-based planning for dual-arm manipulation with upright orientation constraints. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3784–3790. 2.2
- [Corke, 2013] Corke, P. (2013). *Robotics, Vision and Control*. Springer Tracts in Advanced Robotics. 4.1
- [Corke and Hutchinson, 2001] Corke, P. and Hutchinson, S. (2001). A new partitioned approach to image-based visual servo control. *Robotics and Automation, IEEE Transactions on*, 17(4) :507–515. 4.2.1
- [Craig and Raibert, 1979] Craig, J. and Raibert, M. (1979). A systematic method of hybrid position/force control of a manipulator. In *Computer Software and Applications Conference, 1979. Proceedings. COMPSAC 79. The IEEE Computer Society's Third International*, pages 446–451. 2.2.2, 2.2.3
- [Curiosity, 2015] Curiosity (2015). Mars science laboratory. <http://mars.nasa.gov/msl/>. Accessed 8 September 2015. 1
- [Dauchez et al., 1992] Dauchez, P., Delebarre, X., Bouffard-Vercelli, Y., and Dégoulange, E. (1992). Force control experiments with a two-arm robot. *Journal of Intelligent and Robotic Systems*, 5(3) :253–269. 2.2.2, 2.2.2, 2.2.2, 3.3.1.1
- [Dauchez et al., 2005] Dauchez, P., Fraithe, P., and Pierrot, F. (2005). A vision/position/force control approach for performing assembly tasks with a humanoid robot. In *5th IEEE-RAS International Conference on Humanoid Robots*, pages 277–282. 2.1, 2.2.4, 3.3, 3.3.2, 3.3.3, 7
- [De Luca et al., 2007] De Luca, A., Oriolo, G., and Giordano, P. (2007). On-line estimation of feature depth for image-based visual servoing schemes. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2823–2828. 5.2, 5.3, 7

- [De Luca et al., 2008] De Luca, A., Oriolo, G., and Robuffo Giordano, P. (2008). Feature depth observation for image-based visual servoing : Theory and experiments. *The International Journal of Robotics Research*, 27(10) :1093–1116. 5.2
- [Deguchi, 1998] Deguchi, K. (1998). Optimal motion control for image-based visual servoing by decoupling translation and rotation. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 2, pages 705–711 vol.2. 4.2.1
- [DRP2015, 2015] DRP2015 (2015). Darpa robotics challenge. <http://www.theroboticschallenge.org/>. Accessed 9 September 2015. 1
- [Durand Petiteville, 2012] Durand Petiteville, A. (2012). *Navigation référencée multi-capteurs d'un robot mobile en environnement encombré*. PhD thesis, Université Paul Sabatier. 4.2.1
- [Durand Petiteville et al., 2010] Durand Petiteville, A., Courdresses, M., Cadenat, V., and Baillion, P. (2010). On-line estimation of the reference visual features application to a vision based long range navigation task. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3925–3930. 5.2
- [Durand Petiteville et al., 2013] Durand Petiteville, A., Durola, S., Cadenat, V., and Courdresses, M. (2013). Management of visual signal loss during image based visual servoing. In *Control Conference (ECC), 2013 European*, pages 2305–2310. 5.1, 5.2, 5.3
- [Dynamics, 2015] Dynamics, B. (2015). Boston dynamics. <http://www.bostondynamics.com/>. 1
- [Eiden, 2009] Eiden, J.-D. (2009). *Géométrie analytique classique*. 3.3.1.2
- [Espiau et al., 1992] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3) :313–326. 2.2.4, 3.1, 3.3.4, 5.5.1, 5.5.1.1, A
- [Flandin et al., 2000] Flandin, G., Chaumette, F., and Marchand, E. (2000). Eye-in-hand/eye-to-hand cooperation for visual servoing. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2741–2746 vol.3. 3.2, 3.2
- [Fleurmond and Cadenat, 2014] Fleurmond, R. and Cadenat, V. (2014). Multi-cameras visual servoing to perform a coordinated task using a dual arm robot. In *International Conference on Informatics in Control, Automation and Robotics*. 3.5
- [Fleurmond and Cadenat, 2015] Fleurmond, R. and Cadenat, V. (2015). A redundancy based control law for executing a coordinated vision-based task using a dual arm robotic system. In *IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics*. 4.4
- [Folio, 2007] Folio, D. (2007). *Stratégies de commande référencées multi-capteurs et gestion de la perte du signal visuel pour la navigation d'un robot mobile*. PhD thesis, Université Paul Sabatier. 4.2.1, 5.2
- [Folio and Cadenat, 2008] Folio, D. and Cadenat, V. (2008). *Computer Vision - Treating Image Loss by using the Vision/Motion Link : A Generic Framework*, chapter 4. IN-TECH. 5.1, 5.2
- [Forsyth and Ponce, 2002] Forsyth, D. A. and Ponce, J. (2002). *Computer Vision : A Modern Approach*. Prentice Hall Professional Technical Reference. 5.2
- [Garcia-Aracil et al., 2005] Garcia-Aracil, N., Malis, E., Aracil-Santonja, R., and Perez-Vidal, C. (2005). Continuous visual servoing despite the changes of visibility in image features. *IEEE Transactions on Robotics*, 21(6) :1214–1220. 5.1

- [Gharbi et al., 2008] Gharbi, M., Cortes, J., and Simeon, T. (2008). A sampling-based path planner for dual-arm manipulation. In *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pages 383–388. 2.2
- [Grave and Tang, 2015] Grave, I. and Tang, Y. (2015). A new observer for perspective vision systems under noisy measurements. *Automatic Control, IEEE Transactions on*, 60(2) :503–508. 5.2
- [Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition. 3.3.1.2, 5.2, 5.2
- [Hartley and Sturm, 1997] Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer vision and image understanding*, 68(2) :146–157. 5.2, 5.2, 5.3, 5.4.2, 5.4.2
- [Hayati, 1986] Hayati, S. (1986). Hybrid position/force control of multi-arm cooperating robots. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 82–89. 2.2.2, 2.2.2
- [HRP, 2015] HRP (2015). Hrp robots. <http://global.kawada.jp/mechatronics/index.html>. Accessed 9 September 2015. 1
- [Hu et al., 1995] Hu, Y.-R., Goldenberg, A. A., and Zhou, C. (1995). Motion and force control of coordinated robots during constrained motion tasks. *The International Journal of Robotics Research*, 14(4) :351–365. 2.2.2, 2.2.2
- [Hutchinson et al., 1996] Hutchinson, S., Hager, G. D., and Corke, P. I. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5) :651–670. 2.2.4, 3.1
- [Hynes et al., 2005] Hynes, P., Dodds, G., and Wilkinson, A. J. (2005). Uncalibrated visual-servoing of a dual-arm robot for surgical tasks. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation Proceedings*, pages 151–156. 2.2.4
- [Hynes et al., 2006] Hynes, P., Dodds, G., and Wilkinson, A. J. (2006). Uncalibrated visual-servoing of a dual-arm robot for mis suturing. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 420–425. 2.2.4, 3.3.5, 3.5
- [Ingrao, 2011] Ingrao, B. (2011). *Coniques projectives, affines et métriques*. 3.3.1.2
- [Jazar, 2007] Jazar, R. N. (2007). *Theory of Applied Robotics : Kinematics, Dynamics, and Control*. Springer Publishing Company, Incorporated. 4.1
- [Jones and Soatto, 2010] Jones, E. S. and Soatto, S. (2010). Visual-inertial navigation, mapping and localization : A scalable real-time causal approach. *International Journal of Robotics Research*. 5.2
- [Justin, 2015] Justin, D. (2015). Justin dlr. <http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-5471/>. Accessed 9 September 2015. 1
- [Kanatani et al., 2008] Kanatani, K., Sugaya, Y., and Niitsuma, H. (2008). Triangulation from Two Views Revisited : Hartley-Sturm vs. Optimal Correction. In *British Machine Vision Conference*. 5.2, 5.3
- [Kawasaki et al., 2005] Kawasaki, H., Ramli, R., and Ito, S. (2005). Decentralized adaptive coordinated control of multiple robot arms handling one object. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 48(4) :632–639. 2.2.2
- [Kazemi et al., 2010] Kazemi, M., Gupta, K., and Mehrandezh, M. (2010). Path-planning for visual servoing : A review and issues. *G. Chesi et K. Hashimoto, editors, Visual Servoing via Advanced Numerical Methods, Springer Verlag*. 5.1

- [Kermorgant, 2011] Kermorgant, O. (2011). *Fusion d'informations multi-capteurs en asservissement visuel*. PhD thesis, Université de Rennes. 4.2.1, 7
- [Kermorgant and Chaumette, 2011a] Kermorgant, O. and Chaumette, F. (2011a). Avoiding joint limits with a low-level fusion scheme. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 768–773. 4.3.1, 7
- [Kermorgant and Chaumette, 2011b] Kermorgant, O. and Chaumette, F. (2011b). Combining ibvs and pbvs to ensure the visibility constraint. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2849–2854. 5.1
- [Kermorgant and Chaumette, 2011c] Kermorgant, O. and Chaumette, F. (2011c). Multi-sensor data fusion in sensor-based control application to multi-camera visual servoing. In *IEEE International Conference on Robotics and Automation*, pages 4518–4523. 3.2, 3.2, 4.2.1
- [Khalfaoui, 2009] Khalfaoui, S. (2009). Asservissement visuel coordonné pour un manipulateur à deux bras. Mémoire de stage. 3.3.5
- [Klein and Murray, 2007] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*. 5.2
- [Kraus and McCarragher, 1997] Kraus, W., J. and McCarragher, B. J. (1997). Hybrid position/force coordination for dual-arm manipulation of flexible materials. In *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 1, pages 202–207 vol.1. 2.2.2, 2.2.2
- [Lee et al., 2012] Lee, J., Mansard, N., and Park, J. (2012). Intermediate desired value approach for task transition of robots in kinematic control. *IEEE Transactions on Robotics*, 28(6) :1260–1277. 4.2.2, 7
- [Lippiello et al., 2005] Lippiello, V., Siciliano, B., and Villani, L. (2005). Eye-in-hand/eye-to-hand multi-camera visual servoing. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 5354–5359. 3.2
- [Liu and Arimoto, 1998] Liu, Y.-H. and Arimoto, S. (1998). Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations. *The International Journal of Robotics Research*, 17(3) :232–247. 2.2.2, 2.2.2
- [Liza Ahmad Shauri and Nonami, 2011] Liza Ahmad Shauri, R. and Nonami, K. (2011). Assembly manipulation of small objects by dual arm manipulator. *Assembly Automation*, 31(3) :263–274. 2.1
- [Ma et al., 2001] Ma, Y., Kosecka, J., Soatto, S., and Sastry, S. (2001). *An Invitation to 3D Vision :From Images to Models*. Springer. 3.3.1.2, 5.2, 5.5.1.1
- [Malis, 1998] Malis, E. (1998). *Contributions à la modélisation et à la commande en asservissement visuel*. PhD thesis, Université Rennes 1. 4.2.1
- [Malis et al., 1999] Malis, E., Chaumette, F., and Boudet, S. (1999). 2-1/2-d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2) :238–250. 3.1
- [Malis et al., 2000] Malis, E., Chaumette, F., and Boudet, S. (2000). Multi-cameras visual servoing. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3183–3188 vol.4. 3.2
- [Malis et al., 2001] Malis, E., Morel, G., and Chaumette, F. (2001). Robot control using disparate multiple sensors. *The International Journal of Robotics Research*, 20(5) :364 – 377. 3.2, 4.2.1, 7
- [Mansard, 2006] Mansard, N. (2006). *Enchaînement de tâches robotiques*. PhD thesis, Université de Rennes. 3.4.3.3, 4.2.2, 4.2.3

- [Mansard and Chaumette, 2004] Mansard, N. and Chaumette, F. (2004). Tasks sequencing for visual servoing. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 992–997 vol.1. 3.4.3.3
- [Mansard and Chaumette, 2005] Mansard, N. and Chaumette, F. (2005). Visual servoing sequencing able to avoid obstacles. In *IEEE Int. Conf. on Robotics and Automation, ICRA '05*, pages 3154–3159, Barcelona, Spain, France. 4.3.1
- [Mansard and Chaumette, 2007] Mansard, N. and Chaumette, F. (2007). Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics*, 23(1) :60–72. 7
- [Mansard and Chaumette, 2009] Mansard, N. and Chaumette, F. (2009). Directional redundancy for robot control. *Automatic Control, IEEE Transactions on*, 54(6) :1179–1192. 4.2.1, 4.2.1
- [Mansard and Khatib, 2008] Mansard, N. and Khatib, O. (2008). Continuous control law from unilateral constraints. In *IEEE Int. Conf. on Robotics and Automation, ICRA '08*, Las Passadenas, USA. 7
- [Mansard et al., 2009] Mansard, N., Khatib, O., and Kheddar, A. (2009). A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transactions on Robotics*, 25(3) :670–685. 4.3.1
- [Mansard et al., 2007] Mansard, N., Stasse, O., Chaumette, F., and Yokoi, K. (2007). Visually-guided grasping while walking on a humanoid robot. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3041–3047, Rome, Italy. 7
- [Marey and Chaumette, 2010] Marey, M. and Chaumette, F. (2010). A new large projection operator for the redundancy framework. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3727–3732. 4.2.1
- [Martinelli, 2012] Martinelli, A. (2012). Vision and imu data fusion : Closed-form solutions for attitude, speed, absolute scale and bias determination. *IEEE TRANSACTIONS ON ROBOTICS*, 28. 5.2
- [Matthies et al., 1989] Matthies, L. H., Szeliski, R., and Kanade, T. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *Int. J. Computer Vision*, 3(3) :209–236. 5.2, 5.3
- [Maytin-Shepard et al., 2010] Maytin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *IEEE International Conference on Robotics and Automation*, pages 2308–2315. 2.1, 2.2.4
- [Mezouar and Chaumette, 2002a] Mezouar, Y. and Chaumette, F. (2002a). Avoiding self-occlusions and preserving visibility by path planning in the image. *Robotics and Autonomous Systems*, 41(2) :77–87. 3.2, 5.1
- [Mezouar and Chaumette, 2002b] Mezouar, Y. and Chaumette, F. (2002b). Path planning for robust image-based control. *Robotics and Automation, IEEE Transactions on*, 18(4) :534–549. 3.2
- [Mezouar et al., 2007] Mezouar, Y., Prats, M., and Martinet, P. (2007). External hybrid vision/force control. In *13th International Conference on Advanced Robotics, ICAR '07*, pages 170–175, Jeju, Korea. 7
- [Miglani et al., 2013] Miglani, A., Roy, S., Chaudhury, S., and Srivastava, J. (2013). Symmetry based 3d reconstruction of repeated cylinders. In *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on*, pages 1–4. 5.5.2

- [Miyabe et al., 2003] Miyabe, T., Konno, A., and Uchiyama, M. (2003). Automated object capturing with a two-arm flexible manipulator. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 2529–2534 vol.2. 2.2.4
- [Nakamura et al., 1987] Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *Int. J. Rob. Res.*, 6(2) :3–15. 4.2.1
- [Nao, 2015] Nao (2015). Nao aldebaran. <https://www.aldebaran.com/fr/qui-est-nao>. Accessed 9 September 2015. 1
- [OpenROV, 2015] OpenROV (2015). Openrov. <http://www.openrov.com/>. Accessed 9 September 2015. 1
- [Papanikolopoulos et al., 1993] Papanikolopoulos, N., Khosla, P., and Kanade, T. (1993). Visual tracking of a moving target by a camera mounted on a robot : a combination of control and vision. *Robotics and Automation, IEEE Transactions on*, 9(1) :14–35. 3.2
- [Park et al., 2006] Park, C.-H., Park, K.-T., and Gweon, D.-G. (2006). Development of industrial dual arm robot for precision assembly of mechanical parts for automobiles. In *SICE-ICASE, 2006. International Joint Conference*, pages 3059–3062. 2.1
- [Pérez, 1989] Pérez, J. (1989). *Mécanique : points matériels, solides, fluides : avec exercices et problèmes résolus*. Enseignement de la physique. Masson. 2.2.2
- [Philae, 2015] Philae (2015). Philae. [https://en.wikipedia.org/wiki/Philae_\(spacecraft\)](https://en.wikipedia.org/wiki/Philae_(spacecraft)). Accessed 8 September 2015. 1
- [Pissard-Gibollet and Rives, 1995] Pissard-Gibollet, R. and Rives, P. (1995). Applying visual servoing techniques to control a mobile hand-eye system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA' 95)*, pages 166–171, Nagoya, Japan. 3.4.3.3
- [PR2, 2015] PR2 (2015). Pr2 willow garage. <https://www.willowgarage.com/pages/pr2/overview>. Accessed 9 September 2015. 1
- [Ramalingam et al., 2006] Ramalingam, S., Lodha, S. K., and Sturm, P. (2006). A generic structure-from-motion framework. *Comput. Vis. Image Underst.*, 103(3) :218–228. 5.2, 5.3, 5.4.1
- [Recker et al., 2013] Recker, S., Hess-Flores, M., and Joy, K. (2013). Fury of the swarm : Efficient and very accurate triangulation for multi-view scene reconstruction. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 652–659. 5.2, 5.3, 7
- [Remazeilles et al., 2006] Remazeilles, A., Mansard, N., and Chaumette, F. (2006). A qualitative visual servoing to ensure the visibility constraint. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4297–4303. 4.3.2, 5.1, 7
- [Robonaut, 2015] Robonaut (2015). Robonaut. <http://robonaut.jsc.nasa.gov/default.asp>. 1
- [Romeo, 2015] Romeo (2015). Romeo aldebaran. <https://www.aldebaran.com/fr/entreprise-robotique/projet-robotique>. Accessed 9 September 2015. 1
- [Roomba, 2015] Roomba (2015). Roomba. <http://www.irobot.fr/robots-domestiques/aspirateur>. 1
- [Samson et al., 1991] Samson, C., Borgne, M. L., and Espiau, B. (1991). *Robot control : the task function approach*. Oxford engineering science series. Clarendon Press. 3.2, 4.1, 4.2.1, 4.2.1

- [Schneider and Cannon, 1992] Schneider, S. and Cannon, R.H., J. (1992). Object impedance control for cooperative manipulation : theory and experimental results. *Robotics and Automation, IEEE Transactions on*, 8(3) :383–394. 2.2.3
- [Siciliano and Slotine, 1991] Siciliano, B. and Slotine, J.-J. E. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1211 –1216 vol.2. 4.2.1, 4.2.1, 4.2.1, 4.2.2, 4.2.3
- [Smith et al., 2012] Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D., and Kragic, D. (2012). Dual arm manipulation - a survey. *Robotics and Autonomous Systems*, 60(10) :1340 – 1353. 2.1
- [Soueres et al., 2003] Soueres, P., Cadenat, V., and Djeddou, M. (2003). Dynamical sequence of multi-sensor based tasks for mobile robots navigation. *7th Symposium on Robot Control (SYROCO'03)*, 2 :423–428. 3.4.3.3
- [Spica et al., 2014a] Spica, R., Giordano, P., and Chaumette, F. (2014a). Active structure from motion for spherical and cylindrical targets. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5434–5440. 5.2, 5.5.2.1, 5.5.2.1
- [Spica et al., 2014b] Spica, R., Giordano, P., and Chaumette, F. (2014b). Coupling visual servoing with active structure from motion. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3090–3095. 5.2, 7
- [Spica and Robuffo Giordano, 2013] Spica, R. and Robuffo Giordano, P. (2013). A Framework for Active Estimation : Application to Structure from Motion. In *IEEE Conference on Decision and Control, CDC'13*, Firenze, Italy. 5.2, 5.3, 7
- [Spica et al., 2014c] Spica, R., Robuffo Giordano, P., and Chaumette, F. (2014c). Active structure from motion : Application to point, sphere and cylinder. *IEEE Trans. on Robotics*, 30(6) :1499–1513. 5.2, 5.5.2, 7
- [Takamatsu et al., 2007] Takamatsu, J., Ogawara, K., Kimura, H., and Ikeuchi, K. (2007). Recognizing assembly tasks through human demonstration. *The International Journal of Robotics Research*, 26(7) :641–659. 2.1, 2.2
- [Taylor and Kriegman, 1995] Taylor, C. J. and Kriegman, D. J. (1995). Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11) :1021–1032. 5.2
- [Triggs et al., 2000] Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A. (2000). Bundle adjustment – a modern synthesis. In *VISION ALGORITHMS : THEORY AND PRACTICE, LNCS*, pages 298–375. Springer Verlag. 5.2, 5.3
- [Turtlebot, 2015] Turtlebot (2015). Turtlebot. <http://www.turtlebot.com/>. Accessed 8 September 2015. 1
- [Uchiyama and Dauchez, 1988] Uchiyama, M. and Dauchez, P. (1988). A symmetric hybrid position/force control scheme for the coordination of two robots. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 350–356 vol.1. 2.2.2, 2.2.2, 2.2.2, 3.3, 3.3.1.1, 3.3.5, 7
- [Vahrenkamp et al., 2009a] Vahrenkamp, N., Berenson, D., Asfour, T., Kuffner, J., and Dillmann, R. (2009a). Humanoid motion planning for dual-arm manipulation and re-grasping tasks. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09*, pages 2464–2470, Piscataway, NJ, USA. IEEE Press. 2.2
- [Vahrenkamp et al., 2009b] Vahrenkamp, N., Boge, C., Welke, K., Asfour, T., Walter, J., and Dillmann, R. (2009b). Visual servoing for dual arm motions on a humanoid robot. In *9th IEEE-RAS International Conference on Humanoid Robots*, pages 208–214. 2.2.4, 3.3.5, 3.5

- [Vaimos, 2015] Vaimos (2015). Vaimos. <https://www.ensta-bretagne.fr/jaulin/vaimos.html>. Accessed 8 September 2015. 1
- [Watanabe et al., 2005] Watanabe, T., Harada, K., Jiang, Z., and Yoshikawa, T. (2005). Object manipulation under hybrid active/passive closure. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1013–1020. 2.2.2
- [Wimböck and Ott, 2012] Wimböck, T. and Ott, C. (2012). Dual-arm manipulation. In Prassler, E., Zöllner, M., Bischoff, R., Burgard, W., Haschke, R., Hägele, M., Lawitzky, G., Nebel, B., Plöger, P., and Reiser, U., editors, *Towards Service Robots for Everyday Environments*, volume 76 of *Springer Tracts in Advanced Robotics*, pages 353–366. Springer Berlin Heidelberg. 2.2.3
- [Yamada et al., 1995] Yamada, Y., Nagamatsu, S., and Sato, Y. (1995). Development of multi-arm robots for automobile assembly. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 3, pages 2224–2229 vol.3. 2.1
- [Zheng and Chen, 1993] Zheng, Y. and Chen, M. (1993). Trajectory planning for two manipulators to deform flexible beams. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 1019–1024 vol.1. 2.1, 2.2
- [Zollner et al., 2004] Zollner, R., Asfour, T., and Dillmann, R. (2004). Programming by demonstration : Dual arm manipulation tasks for humanoids robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2.1, 2.1, 2.2

Abstract

We address the problem of coordinating a dual arm robot using one or several cameras. After proposing an overview of the control techniques dedicated to this problem, we develop a formalism allowing to coordinate the motions of several arms thanks to multi-cameras image based visual servoing. Our approach allows to benefit from the natural redundancy provided by the robotic system to take into account useful constraints such as joint limits and occlusions avoidance. We propose a strategy to deal with these tasks simultaneously. Finally, to make our control more robust with respect to image losses, we reconstruct the structure of the manipulated objects and the corresponding visual features. To validate our approach, we use the formalism to make the dual arm PR2 robot recap a pen. Simulations and experimental results are provided.

Auteur : Renliw FLEURMOND

Titre : Asservissement visuel coordonné de deux bras manipulateurs

Directeurs de thèse : Viviane CADENAT et Daniel SIDOBRE

Lieu et date de soutenance : LAAS, Toulouse, le 25 Novembre 2015

Résumé

Nous nous intéressons ici au problème de la coordination de plusieurs bras manipulateurs au moyen de la vision. Après avoir étudié les approches de commande dédiées à ce problème, notre première contribution a consisté à établir un formalisme basé sur l'asservissement visuel 2D. Ce formalisme permet de bénéficier des images fournies par une ou plusieurs caméras, embarquées ou déportées, pour coordonner les mouvements d'un système robotique multi-bras. Il permet de plus d'exploiter la redondance de ce type de système pour prendre en compte des contraintes supplémentaires. Nous avons ainsi développé une stratégie de commande pour réaliser une tâche de manipulation coordonnée tout en évitant les butées articulaires et la perte des indices visuels. Afin d'aller plus loin et de tolérer les occultations, nous avons proposé des approches permettant de reconstruire la structure des objets manipulés et donc les indices visuels qui les caractérisent. Enfin, nous avons validé nos travaux en simulation et expérimentalement sur le robot PR2.

Mot clés : Asservissement visuel, coordination, coopération, bras manipulateurs, robot à deux bras, redondance, occultation, structure from motion.