N° d'ordre : 4409 ANNÉE 2011

UNIVERSITÉ DE RENNES 1

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du signal et télécommunications

École doctorale Matisse

présentée par

Olivier KERMORGANT

préparée à l'unité de recherche INRIA Rennes - Bretagne Atlantique
Institut de Recherche en Informatique et Système Aléatoires

Fusion d'informations multi-capteurs en asservissement visuel

Thèse soutenue à Rennes le 28 novembre 2011

devant le jury composé de :

Michèle BASSEVILLE
Directeur de recherche CNRS-IRISA
Présidente

Philippe MARTINET
Professeur à l'École Centrale de Nantes
Rapporteur

Ezio MALIS
PDG Robocortex
Rapporteur

Pierre-Brice WIEBER
Chargé de recherche INRIA
Examinateur

François CHAUMETTE
Directeur de recherche INRIA-IRISA
Directeur de thèse

Remerciements

Je remercie tout d'abord les membres du jury de m'avoir fait l'honneur de s'être intéressés à mon travail. Merci à Michèle Basseville d'avoir présidé avec enthousiasme ce jury. Je remercie particulièrement Philippe Martinet et Ezio Malis d'avoir accepté la charge de rapporteur. Leurs remarques très complémentaires m'ont permis de bien mieux cerner les contours de mes travaux. Merci à Pierre-Brice Wieber pour ses critiques avisées et cette intéressante boîte à outils.

Je remercie chaleureusement François Chaumette pour avoir dirigé mes travaux. Merci notamment pour l'énergie investie dans la relecture de ce mémoire, qui m'a amené à grandement le détailler et l'améliorer. Tu as su me transmettre curiosité et rigueur scientifique et je suis heureux de dire que cette thèse est loin d'être uniquement la mienne.

La convivialité et l'échange qui règnent dans l'équipe Lagadic sont pour beaucoup dans le plaisir de venir chaque matin. Merci tout d'abord à Fabien pour son travail de maintien et d'amélioration des plates-formes robotiques. Contrôler de vrais robots aussi simplement qu'on jouerait avec un simulateur permet de ne jamais remettre une expérience au lendemain et d'oser en faire toujours davantage. Je remercie Céline, wonder-woman-assistante, pour son efficacité redoutable. Merci à Éric et Alexandre pour leur bonne humeur et pour contribuer à cette ambiance.

Les thèses à Lagadic se succèdent et les personnes qu'on y croise en trois ans reflètent cette roue qui tourne. Merci donc à mes coboureaux : Ryuta, Andrea, Pierre puis Tao et finalement François et Caroline. Je suis ravi de clôturer cette année 2011 en compagnie de ma co-thésarde avec qui nous avons partagé les étapes de cette aventure. Un merci particulier aux grandes petites mains de ViSP : Nicolas, Romain et Filip. Merci également à Fabien, Jean, Mohammed, Guillaume F, Deukhee, Guillaume C, Céline, Amaury, Laurent, Hideaki, Antoine, El Mustapha, Bertrand, Mani et Clément.

Un merci particulier à Marie-Pierre, Bertrand et Alain, ma vieille équipe d'Arcelor pour m'avoir donné le goût de la recherche et soutenu dans ma démarche. Merci Gab' pour avoir fait le déplacement pour la soutenance.

Merci à tous ceux dans mon entourage qui ne savent pas la différence exacte entre un jacobien et une matrice d'interaction. Les voileux des Glénans, les jazzeux du big band, et tous les autres de Rennes et d'ailleurs.

Merci enfin à ma famille et à Clémence.

Table des matières

Remerciements i

Table des matières v

Notations vii

Introduction générale 1

1 Notions fondamentales et organisation du manuscrit 3
1.1 Robotique . . . 3
1.1.1 Modélisation . . . 3
1.1.2 Commande proprioceptive . . . 4
1.2 Commande référencée capteurs . . . 5
1.2.1 Généralités . . . 6
1.2.2 Méthodes locales et convergence globale . . . 7
1.2.3 Fusion de capteurs . . . 7
1.3 Vision par ordinateur . . . 8
1.3.1 Projection perspective . . . 9
1.3.2 Modélisation de caméra . . . 10
1.3.3 Géométrie et transformations de l'espace . . . 10
1.3.4 Asservissement visuel . . . 11
1.4 Organisation du mémoire . . . 14

2 Calibration de capteurs 15
2.1 État de l'art . . . 15
2.1.1 Calibrage et étalonnage . . . 16
2.1.2 Calibration de capteur . . . 17
2.1.3 Auto-calibration . . . 18
2.2 Approche proposée . . . 19
2.2.1 Modèle cinématique et expression de l'erreur . . . 20
2.2.2 Asservissement visuel virtuel . . . 21
2.2.3 Paramètres intrinsèques . . . 22
2.2.4 Paramètres extrinsèques . . . 22
2.3 Application à une caméra . . . 25
2.3.1 Paramètres intrinsèques . . . 25
2.3.2 Comparaison avec des approches existantes . . . 26
2.3.3 Calibration pendant un asservissement visuel . . . 28
2.3.4 Simulation . . . 28
2.3.5 Expérience . . . 32
2.3.6 Conclusion sur la calibration de caméra . . . 35
2.4 Observabilité et vision active . . . 36

2.4.1 Calibration partielle d'une caméra . . . 36
2.4.2 Observabilité . . . 37
2.4.3 Optimisation par vision active . . . 37
2.4.4 Résultats de simulation . . . 40
2.4.5 Conclusion sur l'observabilité . . . 42
2.5 Conclusion . . . 44

3 Commande référencée multi-capteurs 45
3.1 État de l'art . . . 45
3.1.1 Modélisation multi-capteurs . . . 46
3.1.2 Commande partitionnée . . . 47
3.1.3 Approches hiérarchiques . . . 48
3.1.4 Commande commutative . . . 49
3.1.5 Commande hybride . . . 50
3.1.6 Tâches à dimension variable . . . 52
3.2 Approche proposée . . . 53
3.2.1 Loi de commande . . . 54
3.2.2 Propriétés de la commande multi-capteurs . . . 59
3.2.3 Pondération adaptative pour l'évitement de butées . . . 62
3.2.4 Problèmes soulevés . . . 72
3.3 Conclusion . . . 77

4 Applications de la fusion 79
4.1 Coopération multi-caméras . . . 80
4.1.1 Fusionner deux caméras . . . 80
4.1.2 Système expérimental . . . 80
4.1.3 Comportement mono-caméra . . . 81
4.1.4 Fusion par moyenne des vitesses . . . 84
4.1.5 Autre configuration . . . 88
4.1.6 Conclusion . . . 91
4.2 Asservissement visuel hybride . . . 91
4.2.1 Loi de commande hybride . . . 92
4.2.2 Simulations exhaustives . . . 93
4.2.3 Comparaison avec d'autres approches . . . 95
4.2.4 Expérience sur un robot réel . . . 102
4.2.5 Conclusion . . . 105
4.3 Évitement de butées articulaires . . . 106
4.3.1 Poids optimaux . . . 107
4.3.2 Simulations pour une position désirée . . . 108
4.3.3 Rotation pure sur robot à 6 axes . . . 108
4.3.4 Comparaison avec un gradient projeté . . . 111
4.3.5 Conclusion . . . 114
4.4 Combinaison de contraintes . . . 115
4.4.1 Dispositif expérimental . . . 115

4.4.2 Application 117
4.4.3 Conclusion 123
4.5 Synthèse 124

Conclusion générale 125

A Calibration entre une caméra et une centrale inertielle 131
A.1 Modèle de centrale inertielle 131
A.2 Formulation de l'approche 133
A.3 Expression des jacobiens 133

Bibliographie 135

Notations

Règles générales

x minuscule	:	scalaire
$\mathbf{v}$ minuscule gras	:	vecteur
$\mathbf{M}$ majuscule gras	:	matrice
$\hat{s}$	:	valeur estimée de s
s^*	:	grandeur de référence pour s
$\mathrm{d}\mathbf{x}$	:	variation infinitésimale de $\mathbf{x}$ entre deux itérations
$\propto$	:	égal à un facteur d'échelle positif près

Cas particulier pour les entités 3D : les coordonnées d'un objet 3D (point, plan, droite) sont en majuscules. Le vecteur des coordonnées est en majuscules gras.

Algèbre

$\mathbb{P}^n$	:	espace projectif de dimension n
$SO(3)$	:	espace orthogonal des matrices de rotation
$SE(3)$	:	espace euclidien des matrices homogènes
$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$	:	concaténation verticale de matrices, vecteurs ou scalaires
$\mathbf{x} = [\mathbf{x}_1 \ldots \mathbf{x}_n]$	:	concaténation horizontale de matrices, vecteurs ou scalaires
$\mathbf{M}_{ij}$	:	élément de la i-ème ligne et j-ème colonne de la matrice $\mathbf{M}$
$\mathbf{M}^\top$	:	matrice transposée de $\mathbf{M}$
$\mathbf{M}^{-1}$	:	matrice inverse de $\mathbf{M}$
$\mathbf{M}^+$	:	matrice pseudo-inverse de $\mathbf{M}$
$\det(\mathbf{M})$	:	déterminant de $\mathbf{M}$
$\mathrm{Ker}(\mathbf{M})$	:	noyau de $\mathbf{M}$
$\mathrm{diag}(\mathbf{d})$	:	matrice de diagonale $\mathbf{d}$
$\exp(\mathbf{M})$	:	matrice exponentielle de $\mathbf{M}$
$[\mathbf{v}]_\times$	:	matrice de préproduit vectoriel associée à $\mathbf{v}$
$\mathbf{0}_{n\times m}$	:	matrice nulle de dimension $n \times m$
$\mathbb{I}_n$	:	matrice identité de dimension $n \times n$

Géométrie euclidienne et projective

$\mathcal{F}_o$	:	repère cartésien orthonormé associé à o
$\mathcal{X}({}^oX, {}^oY, {}^oZ)$	:	point 3D et coordonnées dans le repère $\mathcal{F}_o$
${}^i\mathbf{t}_j = (t_x, t_y, t_z)$	:	vecteur de translation
$\theta\mathbf{u} = (\theta u_x, \theta u_y, \theta u_z)$	:	vecteur de rotation (d'axe $\mathbf{u}$ et d'angle θ)
${}^i\mathbf{R}_j = \exp[\theta\mathbf{u}]_\times$	:	matrice de rotation
${}^i\mathbf{M}_j$	:	matrice homogène de changement de repère
$\mathbf{x}(x, y)$	:	point image en coordonnées normalisées ($x = X/Z$, $y = Y/Z$)
$\mathbf{x}_p(x_p, y_p)$	:	point image en coordonnées pixelliques
$\widetilde{\mathbf{x}} = (\mathbf{x}, 1) = (\lambda\mathbf{x}, \lambda)$	:	coordonnées homogènes de $\mathbf{x}$

Robotique et commande

$\mathbf{q}$	:	variables articulaires
${}^e\mathbf{p}_f = ({}^e\mathbf{t}_f, \theta\mathbf{u})$	:	pose de l'effecteur
$\mathbf{v}_e = (\boldsymbol{v}_e, \boldsymbol{\omega}_e)$	:	torseur cinématique du repère $\mathcal{F}_e$ en son origine $\mathcal{O}_e$
${}^i\mathbf{W}_j$	:	matrice de changement de repère pour le torseur cinématique
$\mathbf{e}$	:	fonction de tâche
$\mathbf{s}$	:	informations capteur
$\mathbf{L}_s$	:	matrice d'interaction associée à $\mathbf{s}$ ($\dot{\mathbf{s}} = \mathbf{L}_s\mathbf{v}_s$)
$\mathbf{J}_s$	:	jacobien associé à $\mathbf{s}$ ($\dot{\mathbf{s}} = \mathbf{J}_s\dot{\mathbf{q}}$)

Introduction générale

Un robot est avant tout défini par sa capacité à se déplacer. La complexité d'un tel système est liée à l'autonomie dont il peut faire preuve dans la réalisation de la mission qui lui a été confiée. Loin des premiers bras manipulateurs répétant en boucle des mouvements gravés dans leur mémoire, le mouvement autonome repose sur l'utilisation de capteurs permettant d'appréhender l'environnement.

Dès lors, une façon naturelle de spécifier une tâche est de la définir par rapport aux capteurs. Ce n'est plus le robot en lui-même qui doit se déplacer ici ou là : c'est son capteur ultrason qui doit mesurer telle distance, sa caméra qui doit acquérir telle image, son altimètre qui doit veiller à rester à 30m au dessus du niveau de la mer. Avec cette approche, le robot peut être vu comme le simple moyen de locomotion des capteurs embarqués. La commande dite référencée capteur offre ainsi une grande souplesse dans la définition de tâches.

Après une phase d'apprentissage, les propriétés connues de la commande référencée capteur sont une excellente précision de positionnement et une robustesse aux erreurs de calibration. La commande référencée capteur trouve aussi son utilité sans apprentissage en définissant les consignes d'après ce qu'on sait - ou pense savoir - de l'environnement. Une bonne calibration du robot est alors nécessaire pour obtenir un comportement satisfaisant.

Un moyen d'augmenter la complexité des tâches réalisables est de combiner plusieurs capteurs. La conception de la commande devient alors plus complexe : l'équilibre entre les capteurs rentre en jeu. D'éventuelles contraintes peuvent s'ajouter à la tâche de positionnement : évitement d'obstacles pour les robots mobiles, maintien d'un objet dans l'image dès qu'une caméra est utilisée.

Nos travaux portent sur la définition de tâches par rapport à plusieurs capteurs. L'objectif est de concevoir des lois de commande prenant en compte des mesures variées. Un aspect important est l'existence de contraintes éventuelles dans le fonctionnement du robot. Leur prise en compte est effectuée directement au niveau de la commande. Le schéma générique proposé permet de définir simplement des lois de commande performantes.

Notre approche repose sur le compromis entre les différents capteurs. La trajectoire du robot est définie comme un optimum globlal sur l'ensemble des mesures. Des critères simples liés aux contraintes permettent alors de faire évoluer ce compromis afin de déterminer de façon réactive le prochain mouvement à opérer.

Publications

Articles de journaux

[R1] O. Kermorgant & F. Chaumette. Multi-sensor-based control. Prepared for IEEE Transactions on Robotics, 2011

Conférences internationales

[C1] O. Kermorgant, D. Folio & F. Chaumette. A new sensor self-calibration framework from velocity measurements. IEEE Int. Conf. on Robotics and Automation, Anchorage, USA, 2010.

[C2] O. Kermorgant & F. Chaumette. Multi-sensor data fusion in sensor-based control : application to multi-camera visual servoing. IEEE Int. Conf. on Robotics and Automation, Shanghai, China, 2011.

[C3] O. Kermorgant & F. Chaumette. Avoiding joint limits with a low-level fusion scheme. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, San Francisco, USA, 2011.

[C4] O. Kermorgant & F. Chaumette. Combining IBVS and PBVS to ensure the visibility constraint. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, San Francisco, USA, 2011.

[C5] T. Li, O. Kermorgant & A. Krupa. Maintaining visibility constraints during tele-echography with ultrasound visual servoing. Submitted in IEEE Int. Conf. on Robotics and Automation, St. Paul, USA, 2012.

Chapitre 1

Notions fondamentales et organisation du manuscrit

Ce premier chapitre introduit le contexte de l'étude et les notions sur lesquelles notre travail s'appuie. Dans un premier temps nous rappelons des éléments de robotique, avant de nous focaliser sur la commande référencée capteurs. Nous nous attardons sur le cas particulier des capteurs de vision, en exposant le fonctionnement des caméras perspectives. Nous présentons ensuite le principe de l'asservissement visuel et l'intérêt de contrôler les robots par la vision. Pour finir, l'organisation générale de ce mémoire est exposée.

1.1 Robotique

Un robot est un système mécatronique dont la particularité est de pouvoir se déplacer en partie (bras manipulateurs) ou entièrement (robots mobiles). Un système robotisé peut être défini par la complexité des tâches qu'il peut réaliser, et le degré d'autonomie dont il est capable. Pour ce faire, il dispose de capteurs qui peuvent être proprioceptifs, centrés sur l'état du système, ou extéroceptifs, donnant des informations sur son environnement. Dans la suite nous ne traitons pas les aspects mécaniques et électroniques, et nous concentrons sur la modélisation géométrique et la commande du système.

1.1.1 Modélisation

Les notions essentielles sur la modélisation de robots sont détaillées dans [Khalil 04]. La pose (position et orientation) d'un robot peut être décrite dans deux espaces :

- L'espace articulaire décrit les positions $\mathbf{q}$ de chaque axe. La dimension n de cet espace est appelé le nombre de degrés de liberté (ddl) du robot.
- L'espace opérationnel décrit la pose $\mathbf{p} = (\mathbf{t}, \theta\mathbf{u})$ [1] du robot par rapport à un repère fixe. Dans ce cas, le repère du robot est attaché à l'effecteur (bras manipulateur) ou au corps du véhicule (robot mobile). Sa dimension N peut aller de 1 (robot se déplaçant suivant un axe unique) à 6 (robot se déplaçant et s'orientant dans l'espace).

S'il est difficile d'évaluer l'ensemble des valeurs possibles pour $\mathbf{p}$, l'espace articulaire est généralement bien identifié pour les bras manipulateurs. Chaque axe est mécaniquement limité par ses butées, la valeur articulaire correspondante ne pouvant sortir d'un intervale

[1] Plusieurs paramétrisations existent pour les poses 3D. La convention utilisée dans ce mémoire est détaillée en Section 1.3.3

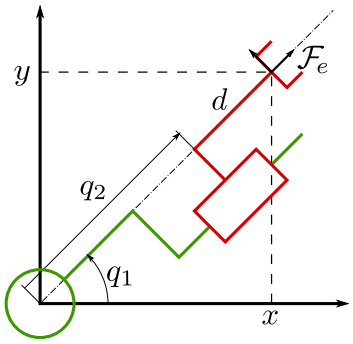


FIG. 1.1 – Manipulateur à 2 degrés de liberté : rotation (q_1) et translation (q_2)

$q_i \in [q_i^-, q_i^+]$.
$\mathbf{q}$ et $\mathbf{p}$ sont reliés par le modèle géométrique : $\mathbf{p} = MG(\mathbf{q})$. Ce modèle dépend des caractéristiques du robot (longueur des bras, diamètre des roues, etc.). Un exemple simple de manipulateur à 2 axes (pivot q_1 et glissière q_2) est représenté Figure 1.1. Évoluant dans le plan, la pose de son effecteur est $\mathbf{p} = (x, y)$. Le modèle géométrique direct s'exprime simplement :

$$MG : \begin{cases} x &= (q_2 + d)\cos q_1 \\ y &= (q_2 + d)\sin q_1 \end{cases} \tag{1.1}$$

La longueur du bras d apparaît explicitement dans le modèle : en l'absence d'autres capteurs, une approximation sur la géométrie du robot entraîne donc une erreur dans la connaissance de sa position opérationnelle, même en mesurant parfaitement sa position articulaire $\mathbf{q}$. Nous nous intéressons maintenant à la commande qui découle de cette modélisation.

1.1.2 Commande proprioceptive

La commande d'un robot peut s'exprimer de trois façons différentes. Le plus bas niveau consiste à agir directement sur la force ou le couple des actionneurs. La loi de commande correspondante est généralement un PID prenant en compte le comportement dynamique du système. Une telle commande est adaptée pour les systèmes rapides où les effets dynamiques ne peuvent être négligés. Plus souvent, on souhaite commander la pose $\mathbf{p}$ de l'effecteur. On utilise alors le modèle géométrique inverse pour déterminer la position articulaire correspondante $\mathbf{q}^* = MGINV(\mathbf{p}^*)$. Une fonction bas niveau du robot prend ensuite le relai afin d'assurer cette tâche.

Enfin, dans ce mémoire les consignes sont exprimées directement en vitesse. Cette commande correspond à une valeur désirée pour le torseur cinématique $\mathbf{v}_e \approx \dot{\mathbf{p}}$, ou pour les vitesses articulaires $\dot{\mathbf{q}}$. Ces grandeurs sont reliées par le jacobien du robot qui correspond à la matrice jacobienne du modèle géométrique et définie par $\mathbf{v}_e = {}^e\mathbf{J}_q\dot{\mathbf{q}}$ avec :

$${}^e\mathbf{J}_q = \frac{\partial MG(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{N \times n} \tag{1.2}$$

Dans l'exemple représenté Figure 1.1 on a :

$${}^e\mathbf{J}_q = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -(q_2 + d)\sin q_1 & \cos q_1 \\ (q_2 + d)\cos q_1 & \sin q_1 \end{bmatrix} \tag{1.3}$$

La non-linéarité de la commande apparaît, ainsi que la dépendence aux grandeurs géométriques du robot. La commande en vitesse de l'effecteur implique de recalculer à chaque pas d'itération le jacobien inverse ${}^{e}\mathbf{J}_{q}^{-1}$ afin de déterminer la vitesse articulaire désirée, qui sera à son tour traitée par la commande bas niveau. Dans la suite de ce mémoire, nous supposons que le robot est capable de suivre une consigne de vitesse et nous intéressons à la production de cette consigne. Dans certaines configurations, le jacobien ne peut pas être inversé : on parle alors de singularité. Avec l'expression de (1.3), c'est par exemple le cas si $q_2 = -d$: la première colonne du jacobien est alors nulle. Cela correspond à une configuration où toutes les combinaisons de vitesses de l'effecteur ne sont pas possibles à appliquer. Des méthodes existent pour détecter et éviter les singularités [Paul 83, Baillieul 84, Nelson 95]. Elles consistent généralement à prendre en compte la proximité d'une singularité à travers une fonction de coût, par exemple $c = \frac{1}{|\det {}^{e}\mathbf{J}_{q}|}$ [Paul 83]. La minimisation de cette fonction de coût est alors intégrée dans la commande du système.

Comme on vient de le voir, l'utilisation des seuls capteurs proprioceptifs implique une grande sensibilité de la commande aux imprécisions géométriques. Cela correspond à une configuration où le robot considère que la tâche est réalisée (les moteurs ont tourné à la bonne vitesse), alors qu'elle ne l'est pas (le robot n'est pas au bon endroit). Ce phénomène est encore amplifié pour les robots mobiles, qui peuvent être soumis à des déplacements non mesurés (glissement sur le sol, dérive due au vent ou au courant). En pratique, une telle commande est envisageable pour les tâches répétitives des robots industriels, après une phase d'apprentissage qui assure que la position finale de l'effecteur est bien celle désirée lors de la tâche. La prochaine section présente la commande référencée capteur, qui donne un premier degré d'autonomie aux robots.

1.2 Commande référencée capteurs

La commande proprioceptive limite le robot aux tâches de positionnement et de suivi de trajectoire prédéfinies. Le formalisme de la fonction de tâche [Samson 91] définit une erreur $\mathbf{e}$ comme l'écart entre la valeur courante et la valeur désirée pour les variables définissant la tâche de positionnement. $\mathbf{e}$ dépend de la position du robot, du temps et éventuellement d'autres grandeurs liées à l'environnement. Outre les problèmes de précision déjà évoqués, une contrainte importante est que l'erreur elle-même doit être définie dans les espaces opérationnel ($\mathbf{e} = \mathbf{p} - \mathbf{p}^*$) ou articulaire ($\mathbf{e} = \mathbf{q} - \mathbf{q}^*$). Le recours à des capteurs extéroceptifs, qui informent sur l'environnement, peut se faire selon deux approches. La première est de venir en complément à la commande proprioceptive afin d'en vérifier le bon fonctionnement : on s'assure que le robot est bien à la position désirée, ou par exemple qu'il faut prendre la décision de modifier la consigne. La deuxième utilisation est d'intégrer directement les informations des capteurs dans la boucle de commande. Cela permet davantage de souplesse, en grande partie car les tâches peuvent être définies directement dans l'espace des mesures capteur. Avec la première approche, un capteur de distance à l'avant d'un robot mobile est cantonné à un rôle de détection d'obstacles permettant de stopper la marche du robot qui

parcourt un chemin prédéfini. Avec la commande référencée capteur, le même robot peut suivre un véhicule en régulant directement la distance mesurée.

Nous détaillons maintenant le formalisme de la commande référencée capteur, avant de dresser un éventail des méthodes de fusion de données.

1.2.1 Généralités

Comme nous l'avons vu, le propre d'un robot est de se déplacer. Une utilisation pertinente des capteurs est de les fixer sur les parties mobiles (typiquement l'effecteur) afin de donner un maximum d'informations sur l'environnement tel que perçu par le robot. L'approche complémentaire est d'utiliser des capteurs fixes pour obtenir des informations sur la position des parties mobiles.

Nous notons $\mathbf{s}$ le vecteur des informations fournies par le capteur, de dimension m. Cela correspond à la grandeur mesurée pour un capteur scalaire (distance, température), ou bien à un vecteur dans le cas de capteurs plus complexes comme les caméras. En supposant que nous disposons d'une consigne $\mathbf{s}^*$, l'erreur correspondante s'exprime simplement par $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$: elle ne s'exprime ni dans l'espace opérationnel, ni dans l'espace articulaire mais directement dans l'espace des mesures du capteur.

La modélisation suit la même logique que pour la commande proprioceptive. Dans un environnement inconnu, le modèle géométrique qui relie la mesure à la position du robot est généralement complexe. On a alors recours au modèle cinématique, qui relie la variation $\dot{\mathbf{s}}$ de la mesure au torseur cinématique $\mathbf{v}_s$ du capteur :

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_s \tag{1.4}$$

où $\mathbf{L}_s \in \mathbb{R}^{m\times 6}$ est la matrice d'interaction du capteur [Samson 91]. Son expression analytique peut être déterminée pour de nombreux types d'informations provenant de capteurs extéroceptifs. Dans [Espiau 92], les calculs analytiques sont explicités pour plusieurs informations visuelles, provenant d'une caméra.

La commance référencée capteur met également en jeu la position du capteur par rapport à l'effecteur. Les torseurs cinématiques $\mathbf{v}_s$ et $\mathbf{v}_e$ sont liés par la relation :

$$\mathbf{v}_s = {}^s\mathbf{W}_e \mathbf{v}_e \tag{1.5}$$

où ${}^s\mathbf{W}_e \in \mathbb{R}^{6\times 6}$ exprime le changement de repère du torseur, et est donné par [Paul 81] :

$${}^s\mathbf{W}_e = \begin{bmatrix} {}^s\mathbf{R}_e & [{}^s\mathbf{t}_e]_\times \, {}^s\mathbf{R}_e \\ \mathbf{0}_{3\times 3} & {}^s\mathbf{R}_e \end{bmatrix} \tag{1.6}$$

où ${}^s\mathbf{R}_e \in \mathcal{SO}(3)$ est la matrice de rotation et ${}^s\mathbf{t}_e \in \mathbb{R}^3$ le vecteur de translation entre $\mathcal{F}_e$ et $\mathcal{F}_s$. D'après (1.2), (1.4) et (1.5) le jacobien capteur $\mathbf{J}_s = \frac{\partial \mathbf{s}}{\partial \mathbf{q}}$ s'exprime par :

$$\mathbf{J}_s = \mathbf{L}_s \, {}^s\mathbf{W}_e \, {}^e\mathbf{J}_q \in \mathbb{R}^{m\times n} \tag{1.7}$$

On peut alors déterminer la loi de commande assurant au mieux une décroissance exponentielle des composantes de $\mathbf{s}$:

$$\dot{\mathbf{q}} = -\lambda \widehat{\mathbf{J}}_s^{\,+} (\mathbf{s} - \mathbf{s}^*) \tag{1.8}$$

où $\lambda > 0$ est un gain et où $\widehat{\mathbf{J}}_s^+$ est une estimation de la pseudo-inverse de $\mathbf{J}_s$. Ainsi, ce formalisme repose sur la connaissance de la matrice d'interaction et de la position du capteur par rapport à l'effecteur : ces aspects sont l'objet du chapitre 2 de ce mémoire. Dans la prochaine section, nous rappelons l'existence de méthodes de planification qui ne reposent pas sur la seule réactivité par rapport aux mesures.

1.2.2 Méthodes locales et convergence globale

La commande référencée capteur fait partie des méthodes locales : elle tente d'assurer qu'à chaque pas d'itération l'erreur décroît dans l'espace des mesures. Comme nous le verrons, cette approche réactive possède de bonnes propriétés de robustesse et d'adaptation à un environnement inconnu. En revanche, la convergence globale du système est potentiellement difficile à garantir : des minima locaux peuvent notamment exister dans certaines configurations.

Pour assurer une convergence globale, d'autres méthodes de contrôle ont recours à la notion de planification. Dans cette approche, un modèle de l'environnement est requis afin de prévoir à l'avance la trajectoire à suivre pour atteindre la position désirée. La commande consiste alors en une série de positions à atteindre [Lozano-Pérez 79]. La méthode des grilles d'occupation [Moravec 85] est populaire en robotique mobile et consiste à discrétiser l'environnement afin d'évaluer facilement les trajectoires possibles. Le calcul de la meilleure trajectoire se fait alors sous la forme d'une minimisation de fonction prenant en compte les contraintes. Elles peuvent provenir de l'environnement [Zhu 91], ou bien de la dynamique du robot, typiquement la planification est adaptée pour la gestion des contraintes non-holonomes [Laumond 94]. Des techniques additionnelles ont été imaginées pour adapter la planification aux changements d'environnement, comme la planification dynamique où la trajectoire peut être recalculée [Erdmann 87], ou déformée [Khatib 97]. En pratique, la convergence globale reste difficile à prouver dans le cas général, et les méthodes de planification reposent en grande partie sur la connaissance de l'environnement et des calculs intensifs permettant de recalculer la trajectoire.

Ce mémoire ne traite pas des aspects de planification, mais tente de répondre aux problèmes des méthodes locales. Notamment, le chapitre 3 présente une solution pour détecter un minimum local et s'en échapper. Pour clôre ces fondamentaux de robotique, nous présentons maintenant les aspects de fusion de capteurs.

1.2.3 Fusion de capteurs

La fusion de capteurs regroupe les techniques permettant d'utiliser conjointement les différents capteurs dont dispose un robot. Comme nous l'avons vu, un capteur peut superviser la commande proprioceptive, ou bien être intégré dans la loi de commande. Cette séparation se retrouve dans le cas de plusieurs capteurs.

Dans le premier cas, l'objectif est de regrouper les données fournies afin d'évaluer au mieux la pose du robot. Il s'agit de la fusion de données au sens classique [Durrant-Whyte 87], dont un large éventail des techniques et des applications est exposé dans [Luo 02]. Ces méthodes sont communes aux systèmes d'information à grande échelle

et reposent sur des outils probabilistes permettant de fournir une estimation de l'état du système, voire de détecter des défauts de fonctionnement. En robotique, une application classique est la fusion entre une caméra et une centrale inertielle [Corke 07] : les mesures inertielles assistent le suivi visuel d'une image à l'autre, tandis que l'estimation de pose permet de corriger les biais inertiels. Dans [Lippiello 07], deux caméras coopèrent pour estimer la position 3D de l'objet observé. Ces méthodes sont pertinentes quand un contrôleur performant existe dans l'espace opérationnel. Le problème est alors de reconstruire l'état du système. Les travaux de [Mellinger 10] illustrent les performances d'un système récent de fusion de données. Un drone léger réalise des manœuvres agressives et très rapides. Ici l'état du système est estimé de façon très précise par fusion des informations fournies par plusieurs caméras dans la salle.

De la même façon, le paradigme de la commande référencée capteur peut être étendu au cas multi-capteurs. L'avantage est que la définition de la tâche se fait dans l'espace des capteurs : aucun modèle particulier du robot n'est requis. Dans cette configuration, on associe généralement une tâche à chaque capteur et la question est de savoir comment combiner ces tâches. Les approches se divisent en deux grandes catégories. La première est de fixer une hiérarchie entre les tâches et de concevoir une loi de commande empêchant les tâches à faible priorité de perturber le déroulement des tâches à haute priorité [Hosoda 98]. C'est une stratégie qui se prête naturellement à la gestion des contraintes [Yoshikawa 96]. L'autre approche est de réaliser l'ensemble des tâches à la fois. L'étude se porte alors sur la façon de réaliser le compromis entre des tâches qui peuvent être contradictoires [Malis 01]. Le chapitre 3 détaille les différentes méthodes employées en commande référencée multi-capteurs, et présente notre contribution à cette approche.

De par leur richesse, les capteurs de vision ont une place importante dans la commande de robots et dans la fusion de capteurs. La section suivante présente les principes de fonctionnement des caméras perspectives et les problématiques de modélisation correspondantes.

1.3 Vision par ordinateur

Contrairement à de nombreux capteurs qui délivrent un signal correspondant à une grandeur physique unique (distance, température, pression...), les caméras fournissent un signal vidéo extrêmement riche, prenant la forme d'un tableau numérique de pixels définis par leur position et leur couleur. L'exploitation de ce signal est donc lié à l'informatique, et peut se situer à différents niveaux. Le plus bas niveau est de considérer directement les informations pixelliques [Collewet 08], qui sont les plus riches mais n'ont pas de sens géométrique à proprement dit. Au niveau du dessus, l'extraction de primitives géométriques (points, lignes [Espiau 87], contours et surfaces [Weiss 85]) est une approche populaire par sa facilité d'implantation et la richesse des informations délivrées quant à l'interaction avec l'environnement. Enfin, il est possible de suivre un objet dans l'image tout en estimant sa pose par rapport à la caméra, à partir d'un modèle 3D de l'objet [Dementhon 95, Drummond 02, Comport 05] ou de sa texture [Pressigout 07]. Dans ce mémoire nous exploitons les résultats d'extraction de primitives géométriques et de suivi 3D pour construire des lois de commande. Aussi, cette section présente les fondamentaux de

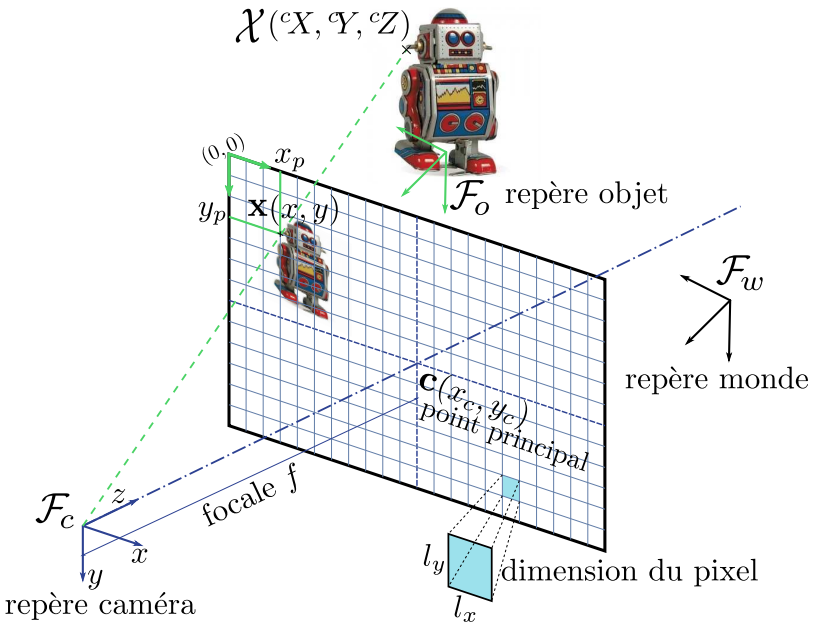


FIG. 1.2 – Projection perspective sur un plan image caméra

géométrie et de modélisation des caméras perspectives, sans aborder le traitement d'image proprement dit. En pratique, nous utilisons la librairie ViSP [Marchand 05] pour suivre et extraire les informations visuelles en temps réel, et pour la mise en oeuvre des méthodes proposées.

1.3.1 Projection perspective

Plusieurs modèles décrivent la projection opérée par une caméra depuis l'espace 3D observé jusqu'à l'image 2D. Le plus populaire est le modèle sténopé (pinhole), de par sa simplicité et son excellente approximation des caméras perspectives. Ce modèle repose sur une projection centrale [Faugeras 93] où l'image d'un point $\mathcal{X}$ de l'espace est la projection de centre $\mathcal{C}$ sur le plan image de la caméra. Par convention, l'axe optique est porté par l'axe z_c, définissant ainsi le plan image par $\mathcal{P}(Z = f)$, où f est la distance focale de la caméra. Le point principal $\mathbf{c}$ est l'intersection de l'axe optique $(\mathcal{C}, z_c)$ avec le plan image. À partir des coordonnées de $\mathcal{X}({}^cX, {}^cY, {}^cZ)$ dans le repère caméra $\mathcal{F}_c$, les coordonnées de la projection $\mathbf{x}(x, y)$ s'expriment par :

$$x = f\frac{{}^cX}{{}^cZ} \ , \ y = f\frac{{}^cY}{{}^cZ} \tag{1.9}$$

ou sous forme matricielle, en coordonnées homogènes :

$$\widetilde{\mathbf{x}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^c\widetilde{\mathbf{X}} = \mathbf{A}\,{}^c\widetilde{\mathbf{X}} \tag{1.10}$$

La Figure 1.2 représente la projection d'un point d'un objet observé. Du point de vue de l'optique géométrique, ce modèle suppose que les conditions de Gauss sont respectées, c'est à dire que les angles d'incidence sont faibles (objets lointains) et que le point principal est proche du centre de l'image.

1.3.2 Modélisation de caméra

La caméra délivre une mesure sous forme d'un tableau de pixels. La position d'un pixel s'exprime par les coordonnées pixelliques $\mathbf{x}_p(x_p, y_p)$ dont l'origine est, par convention, le coin supérieur gauche de l'image. Le modèle de caméra représenté Figure 1.2 relie les coordonnées pixelliques aux coordonnées métriques : il dépend des coordonnées pixelliques (x_c, y_c) du point principal et des dimensions (l_x, l_y) d'un pixel. Le modèle s'exprime par :

$$\left\{ \begin{array}{ll} x_p & = x_c + x/l_x \\ y_p & = y_c + y/l_y \end{array} \right. \tag{1.11}$$

ou sous forme matricielle :

$$\widetilde{\mathbf{x}}_p = \begin{bmatrix} \frac{1}{l_x} & 0 & x_c \\ 0 & \frac{1}{l_y} & y_c \\ 0 & 0 & 1 \end{bmatrix} \widetilde{\mathbf{x}} = \mathbf{K}_l \widetilde{\mathbf{x}} = \mathbf{K}_l \mathbf{A} {}^c\widetilde{\mathbf{X}} \tag{1.12}$$

On regroupe généralement les matrices $\mathbf{K}_l$ et $\mathbf{A}$ en une matrice de calibration $\mathbf{K}$ regroupant l'ensemble des paramètres de la caméra :

$$\mathbf{K} = \mathbf{K}_l \mathbf{A} = \begin{bmatrix} \alpha_x & 0 & x_c & 0 \\ 0 & \alpha_y & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ avec } \alpha_x = \frac{f}{l_x} \text{ et } \alpha_y = \frac{f}{l_y} \tag{1.13}$$

Les coordonnées métriques utilisées sont alors les coordonnées normalisées, indépendantes de la focale :

$$x = \frac{{}^cX}{{}^cZ} \ , \ y = \frac{{}^cY}{{}^cZ} \tag{1.14}$$

La connaissance des paramètres intrinsèques $\boldsymbol{\xi}_{\text{in}} = (\alpha_x, \alpha_y, x_c, y_c)$ est nécessaire pour pouvoir interpréter les mesures pixelliques de la caméra en termes géométriques, et ainsi faire le lien entre l'image et l'environnement. L'opération consistant à identifier ces paramètres est la calibration de caméra et fait l'objet du chapitre 2 de ce mémoire. Des modèles plus évolués existent dans le cas des caméras perspectives, notamment pour la distortion qui est une déformation progressive de la projection à mesure que l'on s'éloigne de l'axe optique [Faugeras 93].

Nous nous attachons maintenant aux transformations dans l'espace qui permettent de décrire la pose de la caméra par rapport à l'objet observé et par rapport à l'effecteur.

1.3.3 Géométrie et transformations de l'espace

La projection dans l'image d'un point 3D $\mathcal{X}$ fait intervenir ses coordonnées ${}^c\mathbf{X} = ({}^cX, {}^cY, {}^cZ)$ exprimées dans le repère caméra. Ces coordonnées changent naturellement avec le temps dès que la caméra ou l'objet bougent. En pratique, le point étant rattaché à l'objet observé, ses coordonnées dans le repère caméra s'expriment à partir de ses coordonnées dans le repère objet ${}^o\mathbf{X} = ({}^oX, {}^oY, {}^oZ)$ avec la relation bien connue : ${}^c\mathbf{X} = {}^c\mathbf{t}_o + {}^c\mathbf{R}_o {}^o\mathbf{X}$. Comme pour les torseurs cinématiques (1.6), cette relation s'écrit sous forme matricielle :

$${}^c\widetilde{\mathbf{X}} = {}^c\mathbf{M}_o {}^o\widetilde{\mathbf{X}} \tag{1.15}$$

où ${}^{c}\mathbf{M}_o$ est la matrice homogène de transformation définie par :

$$ {}^{c}\mathbf{M}_o = \begin{bmatrix} {}^{c}\mathbf{R}_o & {}^{c}\mathbf{t}_o \\ 0 & 1 \end{bmatrix} \tag{1.16} $$

où ${}^{c}\mathbf{t}_o$ et ${}^{c}\mathbf{R}_o$ sont le vecteur de translation et la matrice de rotation pour passer de $\mathcal{F}_o$ à $\mathcal{F}_c$. Comme déjà évoqué, la pose entre la caméra et l'objet observé (qu'on résume en pose de la caméra) peut être estimée à partir d'un modèle de l'objet.

Dans ce mémoire nous utilisons la représentation exponentielle canonique pour les rotations [Grassia 98]. D'autres représentations sont possibles, notamment les angles d'Euler sont populaires pour leur lien avec les axes traditionnels utilisés pour les véhicules (roulis, tangage, lacet).
Dans notre cas, une rotation dans l'espace est paramétrée par le vecteur $\theta\mathbf{u} \in \mathbb{R}^3$ traduisant une rotation d'angle θ autour de l'axe unitaire $\mathbf{u}$. La matrice de rotation correspondante, qui est la matrice de passage du repère d'origine au repère d'arrivée, se construit par une exponentielle de matrice dont la formule de Rodrigues donne une expression analytique :

$$ \mathbf{R} = \exp\left[\theta\mathbf{u}\right]_\times = \mathbb{I}_3 + (1 - \cos\theta)\mathbf{u}^\top\mathbf{u} + \sin\theta\left[\mathbf{u}\right]_\times \tag{1.17} $$

La correspondance étant faite entre un vecteur $\theta\mathbf{u}$ et une matrice de rotation $\mathbf{R}$, un élément du groupe spécial euclidien $SE(3)$ des transformations rigides dans l'espace peut s'écrire indifféremment comme un vecteur à 6 dimensions paramétrant la translation et la rotation $\mathbf{p}(t_x, t_y, t_z, \theta u_x, \theta u_y, \theta u_z)$ ou comme une matrice homogène de changement de repère ${}^{c}\mathbf{M}_o$ (1.16). Notamment le modèle géométrique d'un robot vu en Section 1.1.1 peut s'exprimer sous la forme d'une matrice homogène ${}^{e}\mathbf{M}_f(\mathbf{q})$ traduisant le passage du repère fixe $\mathcal{F}_f$ au repère effecteur $\mathcal{F}_e$.

En particulier, l'écriture paramétrique de la pose entre la caméra et l'effecteur du robot constitue les 6 paramètres extrinsèques de la caméra $\boldsymbol{\xi}_{\text{ex}} = {}^{c}\mathbf{p}_e$. Leur estimation fait aussi partie du processus de calibration. En effet ce changement de repère apparaît dans le jacobien du capteur (1.7), et une bonne approximation de $\boldsymbol{\xi}_{\text{ex}}$ est nécessaire pour obtenir une commande précise. Dans la prochaine section, nous présentons la façon dont une caméra peut être intégrée au sein d'une commande référencée capteur.

1.3.4 Asservissement visuel

L'asservissement visuel consiste à utiliser directement dans la boucle de commande les informations fournies par une ou plusieurs caméras [Chaumette 06, Chaumette 07]. Une étape fondamentale dans cette approche est le choix des primitives géométriques qui seront utilisées. Ces informations visuelles sont l'équivalent vectoriel du signal délivré par un capteur classique.

Les premiers schémas se limitaient à un contrôleur adaptatif sur des informations visuelles choisies pour être découplées sur des robots à faibles degrés de liberté [Sanderson 80, Weiss 85] : typiquement un point peut être centré dans l'image en asservissant les rotations de la caméra sur les coordonnées cartésiennes du point observé. De la même façon, l'aire d'une surface peut être régulée dans l'image avec la seule translation suivant l'axe optique. Dans [Espiau 92], une méthode générale est présentée pour calculer la matrice

d'interaction $\mathbf{L}_s$ de nombreuses primitives géométriques. Héritée de la commande référencée capteur, cette matrice définie en (1.4) relie la variation du vecteur des informations visuelles $\dot{\mathbf{s}}$ au torseur cinématique de la caméra $\mathbf{v}_s$. Son utilisation a ouvert de nombreuses perspectives, notamment par la prise en compte du couplage entre les informations visuelles et les mouvements de la caméra. À l'heure actuelle les propriétés des lois de commande d'asservissement visuel sont bien connues pour la plupart des primitives géométriques. Cette section offre un aperçu des propriétés et des problématiques d'asservissement visuel mises en jeu dans ce mémoire.

Les schémas d'asservissement visuel peuvent être divisés en deux principales approches selon le choix des informations visuelles utilisées [Weiss 87, Chaumette 06]. La première est l'asservissement visuel en position (3D, ou PBVS pour position-based visual servoing) [Sanderson 83]. Dans ce cadre, la vision est utilisée pour estimer la pose de la caméra par rapport à l'objet observé. À l'inverse, l'approche image (2D, ou IBVS pour image-based visual servoing) utilise des primitives géométriques dans l'image. L'étape d'extraction et de suivi est généralement simplifié et le comportement du robot dépend grandement des primitives utilisées. Chaque approche ayant ses défauts et ses qualités, des méthodes hybrides ont été envisagées.

1.3.4.1 Asservissement visuel 3D

Nous avons présenté la commande référencée capteur comme l'intégration du signal du capteur dans la boucle de commande, à l'opposé de son utilisation pour reconstruire l'état du robot. Toutefois, en asservissement visuel on considère que l'estimation de la pose de la caméra est de l'ordre de la commande référencée capteur. Avec cette approche, la caméra fournit les informations visuelles 3D paramétrant la pose ou les coordonnées de points 3D dans le repère caméra [Martinet 97, Wilson 02, Schramm 05]. On parle d'asservissement visuel 3D (PBVS pour Position-Based Visual Servo). Pour ce faire, la caméra doit être calibrée et un modèle de l'objet est généralement nécessaire. Dans [Malis 00], une méthode est présentée pour estimer l'erreur entre la pose courante et la pose désirée à un facteur d'échelle près, à partir de points mis en correspondance. Une paramétrisation populaire est celle représentant la pose entre le repère courant $\mathcal{F}_c$ et le repère désiré $\mathcal{F}_{c*}$: $\mathbf{s} = ({}^{c*}\mathbf{t}_c, {}^{c*}\theta\mathbf{u}_c)$. Cette approche conduit à une trajectoire rectiligne de l'effecteur dans l'espace et est connue pour être globalement asymptotiquement stable (GAS) si la pose est parfaitement estimée. Le principal défaut est qu'aucun contrôle n'est effectué dans l'image. Ainsi, l'objet observé peut sortir du champ de vision, empêchant l'estimation de pose et donc la poursuite de la tâche. Une autre paramétrisation de l'erreur assure que l'origine du repère objet $\mathcal{F}_o$ suit une trajectoire rectiligne dans l'image. Ici le vecteur de translation décrit la pose entre la caméra et l'objet observé : $\mathbf{s} = ({}^{c}\mathbf{t}_o, {}^{c*}\theta\mathbf{u}_c)$. La trajectoire 3D induite n'est en revanche plus une droite dans le cas général, et d'autres parties de l'objet observé peuvent toujours sortir de l'image.

1.3.4.2 Asservissement visuel 2D

Les informations visuelles directement extraites de l'image sont les primitives géométriques classiques : coordonnées d'un point, paramètres d'un segment ou d'un

contour, moments d'une surface [Chaumette 04]. Dernièrement, la valeur brute des intensité des pixels a été utilisée directement dans une boucle de commande [Collewet 08]. En asservissement visuel 2D, la convergence se fait dans l'image : ainsi des points auront une trajectoire rectiligne si leurs coordonnées cartésiennes sont utilisées comme informations visuelles. Conséquence directe, les motifs observés restent visibles au cours de l'asservissement. On parle ici d'asservissement visuel 2D (ou IBVS pour Image-Based Visual Servo). En revanche, aucun contrôle n'est fait dans l'espace 3D. La trajectoire de l'effecteur peut donc être peu satisfaisante, notamment en termes de limites de butées articulaires. Des cas problématiques de singularités ou de minimums locaux ont été identifiés dans [Chaumette 98] et sont régulièrement utilisés pour évaluer la pertinence des nouvelles approches.

1.3.4.3 Approches hybrides

Plusieurs approches ont été proposées pour obtenir les avantages des deux approches précédentes tout en évitant leurs défauts. Une des plus populaires est de regrouper des informations 2D et 3D au sein d'un vecteur de dimension 6 [Malis 99, Cervera 03]. Ces approches sont généralement GAS, conduisent à de bonnes trajectoires 3D tout en garantissant un certain contrôle dans l'image : typiquement le centre de l'objet observé suit une trajectoire rectiligne. D'autres approches ont été spécifiquement imaginées pour la contrainte de visibilité en asservissement visuel 3D. Elles sont détaillées dans le chapitre 4 de ce mémoire, où nous proposons une nouvelle solution à ce problème.

1.4 Organisation du mémoire

La suite du mémoire se structure en trois chapitres. L'objectif est de contrôler un robot de façon réactive en combinant les informations fournies par ses différents capteurs. Les problématiques évoquées dans cette section sont traitées de la façon suivante :

Chapitre 2 - Calibration de capteurs : Tout d'abord nous fournissons un formalisme original pour la calibration de capteur. Cette méthode est adaptée pour la calibration intrinsèque mais aussi extrinsèque des capteurs extéroceptifs. L'approche fait partie des techniques d'auto-calibration : elle peut être réalisée à n'importe quel moment en utilisant les mesures du capteur et les informations proprioceptives. Le formalisme est explicité dans le cas d'une caméra perspective, avant d'être validé expérimentalement sur un robot cartésien. À la suite de ce chapitre, nous faisons l'hypothèse que les capteurs utilisés sont raisonnablement calibrés.

Chapitre 3 - Fusion de capteurs : Le deuxième chapitre présente une approche pour la fusion de capteurs. Après un état de l'art des différentes méthodes de fusion en commande référencée capteurs, nous formalisons l'approche proposée et en exposons les propriétés. Nous identifions ensuite les limitations de la méthode, en proposant des stratégies additionnelles répondant notamment au problème des minima locaux.

Chapitre 4 - Applications de la fusion : Le dernier chapitre de ce mémoire est dédié à différentes mises en œuvre de la fusion bas niveau. Le formalisme présenté dans le chapitre 3 est tout d'abord illustré à travers une tâche de coopération multi-caméras. Nous proposons ensuite une nouvelle solution pour la contrainte de visibilité en asservissement visuel 3D. Nous nous intéressons également au problème de l'évitement de butées articulaires, en intégrant les mesures proprioceptives à la loi de commande. Ces applications sont validées séparément puis regroupées au sein d'une unique expérimentation.

Chapitre 2

Calibration de capteurs

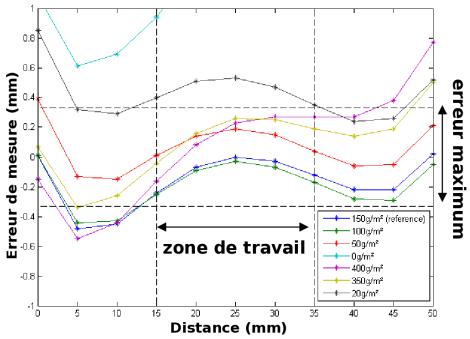
En tant qu'interface entre le robot et son environnement, un capteur extéroceptif fournit un signal fonction de la configuration du robot dans son environnement. La calibration de capteur consiste à identifier les paramètres de cette fonction. Une certaine disparité de fabrication est toujours présente même entre des capteurs identiques. La calibration est donc une étape primordiale pour obtenir des informations précises sur l'environnement, a fortiori pour les utiliser dans une loi de commande. Elle se fait sur deux plans : la calibration intrinsèque identifie les paramètres du modèle du capteur. Les méthodes dépendent généralement du type de capteur utilisé. La calibration extrinsèque consiste quant à elle à identifier la pose du capteur par rapport à un référentiel donné. Cela peut être un référentiel fixe, typiquement on peut estimer la pose d'une caméra dans le repère de l'objet observé [Marchand 01] ou un référentiel mobile comme l'organe terminal du robot considéré. Le référentiel choisi peut également être lié au processus de calibration : toujours en calibration de caméra, il est usuel d'utiliser comme référence le repère de la caméra de la première image d'une séquence [Triggs 98]. Dans le cas où deux capteurs sont utilisés conjointement, le but peut être de déterminer la matrice de passage entre les deux. C'est une étape courante dans l'utilisation d'une caméra et d'un capteur de distance [Zhang 04], ou pour fusionner une caméra et une centrale inertielle [Kelly 11]. Dans ce mémoire nous considérons la calibration pour la commande référencée capteur, par conséquent les paramètres extrinsèques représentent la pose entre le capteur et l'effecteur du robot [Tsai 87]. Ce chapitre présente tout d'abord un état de l'art des méthodes de calibration. Nous exposons l'approche d'auto-calibration proposée, avant de l'expliciter dans le cas d'une caméra. Notre formalisme est ensuite abordé sous l'angle de la vision active pour la calibration d'une caméra.

2.1 État de l'art

L'identification de la réponse d'un capteur peut se faire suivant deux grandes approches. La première consiste à définir une procédure à suivre avant d'utiliser le capteur. L'opération associée requiert souvent une intervention humaine. On parle de calibrage, d'étalonnage ou de calibration[1] selon la façon de procéder. Dans la deuxième approche, c'est l'environnement de travail qui est utilisé pour déterminer les caractéristiques du capteur. Cette configuration ne faisant pas intervenir d'opérateur, elle est particulièrement adaptée à la robotique. On parle alors d'auto-calibration.

[1] Cet anglicisme étant communément admis pour les capteurs, nous proposons d'en limiter l'usage à un certain type d'identification.

(a) Vis de calibrage (3 grossiers et 3 fins) sur un capteur de distance



(b) Courbes d'étalonnage d'un capteur de distance inductif sur différents matériaux

(c) Banc d'étalonnage pour capteur de force

FIG. 2.1 – Étalonnage traditionnel : calibrage par action sur le capteur (a), courbes d'étalonnage d'un capteur face à différents matériaux (b) et banc de mesures (c).

2.1.1 Calibrage et étalonnage

Les procédures traditionnelles d'identification de capteur ne consistent pas à estimer les paramètres de son modèle, mais à établir directement une carte de la réponse du capteur face à des grandeurs connues (étalons). Dans le cas des capteurs analogiques, une première étape dite de calibrage est d'obtenir une réponse aussi fidèle que possible en jouant sur les réglages du capteur. Un capteur de distance peut ainsi présenter jusqu'à six vis de calibrage permettant d'affiner sa réponse (Figure 2.1a). L'objectif du réglage est d'obtenir une relation aussi linéaire que possible entre la grandeur mesurée et la tension ou l'intensité de sortie du capteur. L'étalonnage quant à lui concerne à la fois les capteurs analogiques et numériques. Dans cette procédure on compare la mesure du capteur à la grandeur connue pour en tirer des courbes d'étalonnage. Dans le cas le plus simple, on détermine ainsi l'erreur systématique (ordonnée à l'origine) et l'erreur de sensibilité (pente), ce qui revient à identifier une fonction affine reliant la réponse du capteur à la grandeur mesurée. Dans d'autres cas, la courbe d'étalonnage est loin d'être une droite, elle est alors souvent définie par une fonction linéaire par morceaux (Figure 2.1b). Ces courbes peuvent être utilisées pour valider le capteur si l'estimation d'une réponse affine est d'une précision suffisante. Dans le cas contraire, la courbe d'étalonnage peut être prise en compte dans la commande sous forme d'abaques. En pratique, les procédures de calibrage et d'étalonnage sont longues et fastidieuses. Il est souvent nécessaire de procéder à un grand nombre de mesures et de réglages successifs pour valider un capteur. Ces opérations ont lieu sur des bancs de mesure certifiés. La Figure 2.1c représente un banc de mesure pour capteur de force : cet appareil permet d'appliquer au capteur une force ou un couple avec une grande précision, et ainsi d'établir la courbe d'étalonnage correspondante [Watson 75].

Pour les capteurs dont le modèle analytique est connu, tels que les caméras, l'identification de la réponse du capteur revient à estimer les paramètres de son modèle. Les mesures utilisées servent alors de points de régression pour une estimation linéaire ou non-linéaire. Les procédures associées, dites de calibration de capteur, rejoignent alors le domaine de l'identification paramétrique.

2.1.2 Calibration de capteur

La calibration de capteur consiste à réaliser, avant l'utilisation, une série de mesures permettant d'estimer avec le maximum de précision les paramètres du capteur. En robotique cette technique s'adresse principalement à deux problématiques. La première est la calibration du capteur lui-même, pour en estimer les paramètres intrinsèques. La deuxième est la calibration extrinsèque où l'on cherche à estimer la pose relative de deux capteurs, ou la pose d'un capteur par rapport à l'effecteur du robot (calibration main-œil ou hand-eye calibration). La calibration peut porter sur d'autres types de problèmes. Par exemple, dans [Borenstein 96] on estime l'erreur systématique et la dérive d'un capteur odométrique pour robots mobiles. Dans [Voyles 97], le modèle linéaire d'un capteur d'effort est estimé en confrontant les mesures du capteur dans différentes positions.

2.1.2.1 Calibration de caméra

La calibration de caméra est une étape indispensable pour faire le lien entre les informations pixelliques et l'espace métrique projeté. Cette procédure requiert une mire de calibration, équivalent de l'étalon pour les mesures physiques. Une telle mire est généralement constituée de points ou d'un damier dont la disposition 3D est connue et non ambigue. Selon la mire, une ou plusieurs images sont nécessaires pour procéder à la calibration. La problématique est d'identifier les paramètres de la relation entre la mire 3D et l'image 2D observée. Les premières techniques de calibration se contentent d'une estimation linéaire [Faugeras 87] ou non-linéaire [Brown 71]. Des approches partitionnées ont également été proposées pour estimer certains paramètres de façon linéaire, avant d'estimer les autres en non-linéaire [Tsai 87].

Plus récemment, la calibration de caméra a été formulée comme un problème d'asservissement visuel virtuel [Marchand 01]. Comme dans les techniques d'estimation non linéaire, l'approche est de minimiser l'erreur de reprojection des informations visuelles de la mire sur une séquence de plusieurs images. L'originalité est de formuler cette minimisation comme un asservissement visuel, ce qui la rend générique du point de vue des primitives géométriques choisies et reprend la simplicité des fonctions de tâche exposée en Section 1.2.1. D'un point de vue algébrique, la minimisation se fait dans $SE(3)$ pour la partie extrinsèque, ce qui permet de calculer plus facilement le jacobien utilisé dans la minimisation.

Dans notre approche nous exprimons également le problème sous la forme d'un asservissement visuel virtuel. En revanche nous n'utilisons pas de mire connue, par conséquent l'erreur à minimiser ne s'exprime pas sous la forme d'une erreur de reprojection.

2.1.2.2 Calibration main-œil

La pose relative entre l'effecteur et un capteur peut souvent être grossièrement estimée à la main avec une simple règle graduée. Toutefois, il est parfois nécessaire de connaître avec précision ce changement de repère. Dans [Tsai 89], une première solution est donnée en comparant les poses de l'effecteur et de la caméra dans plusieurs configurations spatiales. La pose de l'effecteur par rapport au repère de base du robot ${}^{f}\mathbf{M}_{e}$ est connue avec une grande précision à travers le modèle géométrique direct du robot, tandis que celle de la caméra par rapport à l'objet ${}^{c}\mathbf{M}_{o}$ peut être estimée à partir des images si la caméra est calibrée et la mire connue. L'objet observé étant fixe, la matrice ${}^{f}\mathbf{M}_{o}$ est constante pour toutes les configurations :

$$ {}^{f}\mathbf{M}_{o} = {}^{f}\mathbf{M}_{e}\,{}^{e}\mathbf{M}_{c}\,{}^{c}\mathbf{M}_{o} = {}^{f}\mathbf{M}_{e'}\,{}^{e}\mathbf{M}_{c}\,{}^{c'}\mathbf{M}_{o} \tag{2.1} $$

ce qui revient à :

$$ {}^{e'}\mathbf{M}_{e}\,{}^{e}\mathbf{M}_{c} = {}^{e}\mathbf{M}_{c}\,{}^{c'}\mathbf{M}_{c} \tag{2.2} $$

où ${}^{e}\mathbf{M}_{c}$ est la pose main-œil à estimer. Dans [Tsai 89] la résolution se fait en deux étapes permettant une formulation linéaire. D'autres approches ont été proposées [Horaud 95, Remy 97, Andreff 98] pour résoudre (2.2) de façon globale avec une minimisation non-linéaire.
Un cas particulier de la calibration main-œil est la calibration du changement de repère entre une caméra et une centrale inertielle. Ces deux capteurs étant très complémentaires, leur utilisation conjointe requiert en effet de connaître la pose relative entre les deux capteurs. Dans [Lobo 07], la caméra observe une cible composée de lignes verticales. Les mouvements de rotation sont isolés des mouvements de translation afin d'identifier les paramètres de calibration. Dans [Mirzaei 08] c'est un filtre de Kalman qui est utilisé pour estimer à la fois la pose relative, et les biais d'accélération et de vitesse angulaire de la centrale inertielle.

Comme nous l'avons vu, les techniques classiques de calibration requièrent la connaissance de la géométrie d'une mire pour exprimer explicitement une erreur à minimiser. Ces approches sont adaptées à des robots évoluant dans des environnements contrôlés, où il est possible de réaliser régulièrement la procédure de calibration. À l'inverse, les robots mobiles ou évoluant dans des milieux très contraints (sous-marin, espace) ont rarement une mire de calibration à leur disposition. On fait alors appel aux techniques dites d'auto-calibration, où ce sont les données naturelles de l'environnement qui sont utilisées pour estimer les paramètres du capteur.

2.1.3 Auto-calibration

L'auto-calibration a été proposée dans [Maybank 92] pour estimer les paramètres intrinsèques d'une caméra uniquement à partir d'une séquence d'images de la même scène dont la structure euclidienne est inconnue. Dans [Heyden 99], il est montré qu'une hypothèse minimale pour l'auto-calibration est qu'un des paramètres doit être connu ou supposé constant. Les différentes approches d'auto-calibration sont présentées et détaillées dans

[Skordas 95]. Deux grandes classes de méthodes sont recensées. Tout d'abord, les méthodes reposant sur la géométrie épipolaire qui relie les images d'une même scène, généralement fondées sur les équations de Kruppa [Kruppa 13]. Un inventaire très complet en a été dressé dans [Hemayed 03]. Comme en calibration classique, des méthodes non-linéaires furent d'abord proposées [Maybank 92, Faugeras 92]. Par la suite, des approches stratifiées furent envisagées pour résoudre une première partie de façon linéaire [Pollefeys 99]. D'autres travaux simplifient les équations de la géométrie épipolaire en imposant des contraintes sur la scène [Sturm 99, Zhang 00] ou sur le mouvement de la caméra [Moons 96, Hartley 94]. Enfin, la notion de mouvements critiques (critical motion sequences) a été introduite dans [Sturm 97]. Ces mouvements critiques représentent les séquences d'images qui aboutissent à des systèmes dégénérés qui ne permettent pas de retrouver les paramètres de la caméra.

L'autre classe est à rapprocher de la robotique car elle repose sur la connaissance du mouvement de la caméra. Une première approche est de contrôler la caméra spécialement pour la calibration : ainsi dans [Viéville 94, Basu 93, Du 93], on opère des rotations connues pour déterminer avec précision les paramètres intrinsèques. Une autre approche est de déterminer les mouvements optimaux à appliquer. L'estimation des informations 3D de l'environnement à partir de la vitesse a été introduite dans [Chaumette 96] sous le nom de structure from controlled motion. L'objectif initial est d'identifier les informations 3D des primitives géométriques observées dans l'image (points, lignes, cylindres, etc.). Dans [De Luca 08], l'approche est combinée à l'auto-calibration à travers un observateur permettant d'estimer à la fois la profondeur des points observés et la focale variable de la caméra.

L'auto-calibration a aussi été appliquée aux paramètres extrinsèques : dans [Andreff 01] la calibration main-œil est résolue sans mire, et dans [Wei 98b] c'est la pose d'un capteur de distance laser qui est estimée. Dans [Wei 98a], une méthode est proposée pour déterminer les paramètres intrinsèques et extrinsèques d'une caméra observant des points. Toutefois la résolution est purement numérique et oblige à estimer également la position 3D absolue des points observés.

Notre approche fait partie des techniques d'auto-calibration où le mouvement de l'effecteur est connu. La méthode repose sur la mesure de la vitesse de l'effecteur. Elle est à rapprocher de [De Luca 08], car nous utilisons le modèle cinématique pour exprimer l'erreur à minimiser en fonction des paramètres de calibration. Elle est voisine de [Folio 08] où le modèle cinématique est utilisé pour estimer la profondeur des points observés. Pour l'application à une caméra, ces deux approches seront explicitées dans un formalisme identique au nôtre. Le modèle cinématique est également utilisé dans [Mirzaei 08] pour la calibration entre une caméra et une centrale inertielle.

2.2 Approche proposée

Notre approche exploite le modèle cinématique exposé en Section 1.2.1. L'idée est de confronter les variations des mesures capteur avec la vitesse de l'effecteur. La même formulation permet d'estimer les paramètres intrinsèques et extrinsèques. Dans cette section nous exposons les principes généraux valables pour tout capteur extéroceptif dont la matrice

d'interaction peut être définie.

2.2.1 Modèle cinématique et expression de l'erreur

Dans ce chapitre, nous considérons un système dont la vitesse de l'effecteur est connue et dont le capteur fournit un vecteur de mesures $\mathbf{s}$. On rappelle que les paramètres intrinsèques relient les données du capteur à sa vitesse dans l'espace :

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_s \tag{2.3}$$

où $\mathbf{L}_s$ est la matrice d'interaction fonction des mesures et des paramètres intrinsèques, et où le torseur cinématique du capteur $\mathbf{v}_s$ est lié à celui de l'effecteur $\mathbf{v}_e$:

$$\mathbf{v}_s = {}^s\mathbf{W}_e \mathbf{v}_e \tag{2.4}$$

où ${}^s\mathbf{W}_e$ est fonction des paramètres extrinsèques :

$${}^s\mathbf{W}_e = \begin{bmatrix} {}^s\mathbf{R}_e & [{}^s\mathbf{t}_e]_\times \, {}^s\mathbf{R}_e \\ \mathbf{0}_{3\times 3} & {}^s\mathbf{R}_e \end{bmatrix} \tag{2.5}$$

Les équations (1.4) et (1.5) permettent de regrouper ces considérations en reliant directement la variation des mesures capteur $\dot{\mathbf{s}}$ à la vitesse de l'effecteur $\mathbf{v}_e$:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_s = \mathbf{L}_s \, {}^s\mathbf{W}_e \mathbf{v}_e \tag{2.6}$$

Le modèle cinématique ainsi exprimé fait intervenir l'ensemble des paramètres du capteur :

- La matrice d'interaction $\mathbf{L}_s$ dépend des paramètres intrinsèques $\boldsymbol{\xi}_{\text{in}}$ et des mesures $\mathbf{s}$. Elle peut aussi dépendre d'autres grandeurs : par exemple la matrice d'interaction d'un point observé par une caméra dépend de la profondeur Z qui n'est pas mesurée dans l'image.
- La matrice de changement de torseur cinématique ${}^s\mathbf{W}_e$ dépend elle des paramètres extrinsèques $\boldsymbol{\xi}_{\text{ex}}$.

En supposant une approximation sur la valeur des paramètres de calibration, (2.6) permet d'exprimer l'erreur de calibration :

$$\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})} = \mathbf{L}_{s(\widehat{\boldsymbol{\xi}_{\text{in}}})} \, {}^s\mathbf{W}_{e(\widehat{\boldsymbol{\xi}_{\text{ex}}})} \mathbf{v}_e - \dot{\mathbf{s}} \tag{2.7}$$

où $\mathbf{L}_{s(\widehat{\boldsymbol{\xi}_{\text{in}}})}$ et ${}^s\mathbf{W}_{e(\widehat{\boldsymbol{\xi}_{\text{ex}}})}$ sont les matrices calculées à partir des valeurs estimées $\widehat{\boldsymbol{\xi}_{\text{in}}}$ et $\widehat{\boldsymbol{\xi}_{\text{ex}}}$. On note $\widehat{\mathbf{v}_s} = {}^s\mathbf{W}_{e(\widehat{\boldsymbol{\xi}_{\text{ex}}})} \mathbf{v}_e$ la vitesse estimée du capteur. L'erreur définie dans (2.7) étant non-linéaire suivant les paramètres, nous proposons de l'annuler par une descente de gradient réalisée à chaque nouvelle mesure. Nous présentons tout d'abord le schéma global d'asservissement visuel virtuel similaire à celui proposé dans [Marchand 01], avant d'expliciter le calcul du jacobien de (2.7) par rapport aux paramètres intrinsèques et extrinsèques.

2.2.2 Asservissement visuel virtuel

L'évolution de l'erreur (2.7) en fonction des paramètres de calibration est donnée par :

$$\mathrm{d}\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})} = \frac{\partial \boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})}}{\partial \boldsymbol{\xi}_{\text{in}}} \mathrm{d}\boldsymbol{\xi}_{\text{in}} + \frac{\partial \boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})}}{\partial \boldsymbol{\xi}_{\text{ex}}} \mathrm{d}\boldsymbol{\xi}_{\text{ex}} \tag{2.8}$$

où d$\mathbf{x}$ indique la variation infinitésimale de $\mathbf{x}$ entre deux itérations. Les paramètres extrinsèques dénotant la pose entre le capteur et l'effecteur du robot, leur évolution d$\boldsymbol{\xi}_{\text{ex}}$ peut être assimilée à un déplacement infinitésimal. Ce déplacement infinitésimal entre deux itérations est noté $\bar{\mathbf{v}} = (\bar{\boldsymbol{v}}, \bar{\boldsymbol{\omega}})$ par analogie avec un torseur cinématique. (2.8) peut alors s'écrire de la façon suivante :

$$\mathrm{d}\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})} = \mathbf{J}_{\text{in}} \mathrm{d}\boldsymbol{\xi}_{\text{in}} + \mathbf{L}_{\text{ex}} \bar{\mathbf{v}} \tag{2.9}$$

où $\mathbf{J}_{\text{in}}$ est le jacobien associé aux paramètres intrinsèques, et $\mathbf{L}_{\text{ex}}$ la matrice d'interaction de l'erreur associée aux paramètres extrinsèques. En posant $\mathrm{d}\mathbf{m} = (\mathrm{d}\boldsymbol{\xi}_{\text{in}}, \bar{\mathbf{v}})$, (2.9) s'écrit :

$$\mathrm{d}\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})} = \begin{bmatrix} \mathbf{J}_{\text{in}} & \mathbf{L}_{\text{ex}} \end{bmatrix} \mathrm{d}\mathbf{m} \tag{2.10}$$

Imposer une décroissance exponentielle des éléments de l'erreur (2.7) revient à déterminer d$\mathbf{m}$ selon la loi suivante :

$$\mathrm{d}\mathbf{m} = -\lambda \begin{bmatrix} \mathbf{J}_{\text{in}} & \mathbf{L}_{\text{ex}} \end{bmatrix}^{+} \boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})} \tag{2.11}$$

Pour plus de robustesse et pour avoir suffisamment de mesures, l'optimisation peut se faire sur une fenêtre glissante de profondeur h. On a alors :

$$\mathrm{d}\mathbf{m} = -\lambda \begin{bmatrix} \mathbf{J}_{\text{in}1} & \mathbf{L}_{\text{ex}1} \\ \vdots & \vdots \\ \mathbf{J}_{\text{in}h} & \mathbf{L}_{\text{ex}h} \end{bmatrix}^{+} \begin{bmatrix} \boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})^1} \\ \vdots \\ \boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}},\widehat{\boldsymbol{\xi}_{\text{ex}}})^h} \end{bmatrix} \tag{2.12}$$

En outre nous verrons que la calibration extrinsèque requiert une fenêtre de dimension $h \geq 2$ pour être réalisée.

Entre deux itérations i et $(i+1)$ de la minimisation, la mise à jour des paramètres intrinsèques est une somme classique. Les paramètres extrinsèques étant assimilés à une matrice homogène ${}^{s}\mathbf{M}_{e(i)}$, leur mise à jour est réalisée dans $SE(3)$ par carte exponentielle :

$${}^{s}\mathbf{M}_{e(i+1)} = \exp\left(-\left[\bar{\mathbf{v}}\right]_{\times}\right) {}^{s}\mathbf{M}_{e(i)} \tag{2.13}$$

Entre deux acquisitions k et $(k+1)$ pour les mesures, les paramètres de calibration sont mis à jour par moyenne pondérée afin de lisser le processus. Le paramètre de pondération est $\alpha \in [0, 1]$ où aucune mise à jour n'est faite quand $\alpha = 0$ et où aucun lissage n'est fait quand $\alpha = 1$. En notant $(\widehat{\boldsymbol{\xi}_{\text{in}}}^{\text{new}}, {}^{s}\mathbf{M}_{e}^{\text{new}})$ le résultat de la minimisation, la mise à jour des paramètres intrinsèques se fait par :

$$\widehat{\boldsymbol{\xi}_{\text{in}(k+1)}} = (1-\alpha)\widehat{\boldsymbol{\xi}_{\text{in}(k)}} + \alpha \widehat{\boldsymbol{\xi}_{\text{in}}}^{\text{new}} \tag{2.14}$$

La mise à jour des paramètres extrinsèques se fait dans $SE(3)$ de façon similaire à (2.13) par carte exponentielle avec le terme de pondération α :

$$ {}^{s}\mathbf{M}_{e(k+1)} = {}^{s}\mathbf{M}_{e(k)} \exp\left(\alpha \ln\left({}^{s}\mathbf{M}_{e(k)}^{-1} {}^{s}\mathbf{M}_{e}^{\text{new}}\right)\right) \tag{2.15} $$

Avec cette formulation, on a bien ${}^{s}\mathbf{M}_{e(k+1)} = {}^{s}\mathbf{M}_{e(k)}$ quand $\alpha = 0$ et ${}^{s}\mathbf{M}_{e(k+1)} = {}^{s}\mathbf{M}_{e}^{\text{new}}$ quand $\alpha = 1$. D'autres approches existent pour réaliser des moyennes dans $SE(3)$: par exemple dans [Moakher 02] la moyenne arithmétique des matrices homogènes est projetée sur $SE(3)$. Nous explicitons maintenant le calcul de $\mathbf{J}_{\text{in}}$ et $\mathbf{L}_{\text{ex}}$.

2.2.3 Paramètres intrinsèques

On note $m = \dim(\mathbf{s})$ et $k = \dim(\boldsymbol{\xi}_{\text{in}})$. $\mathbf{J}_{\text{in}}$ est la matrice de dimension $(m \times k)$ telle que :

$$ \mathrm{d}\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}})} = \mathbf{J}_{\text{in}} \mathrm{d}\boldsymbol{\xi}_{\text{in}} \tag{2.16} $$

En notant $\boldsymbol{\xi}_{\text{in}} = (\boldsymbol{\xi}_{\text{in}1}, \ldots, \boldsymbol{\xi}_{\text{in}k})$, $\mathbf{J}_{\text{in}}$ s'exprime directement à partir de (2.7) par :

$$ \mathbf{J}_{\text{in}} = \frac{\partial \boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}})}}{\partial \boldsymbol{\xi}_{\text{in}}} = \frac{\partial \mathbf{L}_{s(\widehat{\boldsymbol{\xi}_{\text{in}}})} \widehat{\mathbf{v}_s}}{\partial \boldsymbol{\xi}_{\text{in}}} = \begin{bmatrix} \frac{\partial \mathbf{L}_{s1}}{\partial \boldsymbol{\xi}_{\text{in}}} \widehat{\mathbf{v}_s} \\ \vdots \\ \frac{\partial \mathbf{L}_{sm}}{\partial \boldsymbol{\xi}_{\text{in}}} \widehat{\mathbf{v}_s} \end{bmatrix} \tag{2.17} $$

où $\mathbf{L}_{si}$ est la $i^{\text{ème}}$ ligne de $\mathbf{L}_{s(\widehat{\boldsymbol{\xi}_{\text{in}}})}$. Le jacobien par rapport aux paramètres intrinsèques dépend des dérivées de la matrice d'interaction. Par conséquent, il peut être déterminé analytiquement pour tout capteur dont la matrice d'interaction est connue. En Section 2.3.1, le calcul est explicité dans le cas d'une caméra observant un point.

2.2.4 Paramètres extrinsèques

Dans cette section nous exprimons tout d'abord la matrice d'interaction $\mathbf{L}_{\text{ex}}$ définie en Section 2.2.2. Nous établissons ensuite un lien entre notre approche et les techniques usuelles de calibration extrinsèques.

2.2.4.1 Matrice d'interaction

La matrice d'interaction $\mathbf{L}_{\text{ex}}$ de l'erreur (2.7) est la matrice de dimension $(m \times 6)$ telle que :

$$ \mathrm{d}\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{ex}}})} = \mathbf{L}_s \; \mathrm{d}^{s}\mathbf{W}_{e(\widehat{\boldsymbol{\xi}_{\text{ex}}})} \; \mathbf{v}_e = \mathbf{L}_{\text{ex}} \bar{\mathbf{v}} \tag{2.18} $$

En notant $\mathbf{L}_{\text{ex}} = \mathbf{L}_s \mathbf{K}$, déterminer $\mathbf{L}_{\text{ex}}$ revient à exprimer la matrice $\mathbf{K} \in \mathbb{R}^{6\times6}$ qui vérifie :

$$ \mathrm{d}^{s}\mathbf{W}_{e(\widehat{\boldsymbol{\xi}_{\text{ex}}})} \; \mathbf{v}_e = \mathbf{K} \bar{\mathbf{v}} \tag{2.19} $$

D'après (1.6), $\mathrm{d}^{s}\mathbf{W}_{e(\widehat{\boldsymbol{\xi}_{\text{ex}}})}$ s'exprime par :

$$ \mathrm{d}^{s}\mathbf{W}_{e} = \left[\begin{array}{c|c} \mathrm{d}^{s}\mathbf{R}_e & [\mathrm{d}^{s}\mathbf{t}_e]_\times \, {}^{s}\mathbf{R}_e + [{}^{s}\mathbf{t}_e]_\times \, \mathrm{d}^{s}\mathbf{R}_e \\ \hline \mathbf{0}_{3\times3} & \mathrm{d}^{s}\mathbf{R}_e \end{array}\right] \tag{2.20} $$

avec :

$$\begin{cases} \mathrm{d}^s\mathbf{t}_e = & -\bar{\boldsymbol{v}} + [{}^s\mathbf{t}_e]_\times \bar{\boldsymbol{\omega}} \\ \mathrm{d}^s\mathbf{R}_e = & {}^s\mathbf{R}_e [\bar{\boldsymbol{\omega}}]_\times \end{cases} \tag{2.21}$$

D'après (2.20) et (2.21), $\mathrm{d}^s\mathbf{W}_e$ se décompose en trois matrices 3×3 fonctions de $\bar{\mathbf{v}}$, dénotées $\mathbf{A}(\bar{\boldsymbol{v}})$ fonction de la partie linéaire $\bar{\boldsymbol{v}}$, et $\mathbf{B}(\bar{\boldsymbol{\omega}})$ et $\mathbf{C}(\bar{\boldsymbol{\omega}})$ fonctions de la partie angulaire $\bar{\boldsymbol{\omega}}$ telles que :

$$\mathrm{d}^s\mathbf{W}_e = \left[\begin{array}{c|c} \mathbf{B}(\bar{\boldsymbol{\omega}}) & \mathbf{A}(\bar{\boldsymbol{v}}) + \mathbf{C}(\bar{\boldsymbol{\omega}}) \\ \hline \mathbf{0}_{3\times3} & \mathbf{B}(\bar{\boldsymbol{\omega}}) \end{array}\right] \tag{2.22}$$

En injectant (2.21) dans (2.20) on obtient :

$$\begin{aligned} \mathbf{B} = \mathrm{d}^s\mathbf{R}_e &= {}^s\mathbf{R}_e [\bar{\boldsymbol{\omega}}]_\times \\ \mathbf{A} + \mathbf{C} = \mathrm{d}[{}^s\mathbf{t}_e]_\times {}^s\mathbf{R}_e &+ [{}^s\mathbf{t}_e]_\times \mathrm{d}^s\mathbf{R}_e \\ &= \left[-\bar{\boldsymbol{v}} + [{}^s\mathbf{t}_e]_\times \bar{\boldsymbol{\omega}}\right]_\times {}^s\mathbf{R}_e + [{}^s\mathbf{t}_e]_\times {}^s\mathbf{R}_e [\bar{\boldsymbol{\omega}}]_\times \end{aligned}$$

d'où on déduit :

$$\mathbf{A}(\bar{\boldsymbol{v}}) = -[\bar{\boldsymbol{v}}]_\times {}^s\mathbf{R}_e \tag{2.23}$$

$$\mathbf{B}(\bar{\boldsymbol{\omega}}) = {}^s\mathbf{R}_e [\bar{\boldsymbol{\omega}}]_\times \tag{2.24}$$

$$\mathbf{C}(\bar{\boldsymbol{\omega}}) = \left[[{}^s\mathbf{t}_e]_\times \bar{\boldsymbol{\omega}}\right]_\times {}^s\mathbf{R}_e + [{}^s\mathbf{t}_e]_\times {}^s\mathbf{R}_e [\bar{\boldsymbol{\omega}}]_\times \tag{2.25}$$

On voit que $\mathbf{A}$ ne dépend que de $\bar{\boldsymbol{v}}$, et $\mathbf{B}$ et $\mathbf{C}$ que de $\boldsymbol{\omega}$. On décompose de la même façon $\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}$. Injecter (2.22) dans (2.19) conduit à :

$$\begin{aligned} \left[\begin{array}{c|c} \mathbf{B} & \mathbf{A}+\mathbf{C} \\ \hline \mathbf{0}_{3\times3} & \mathbf{B} \end{array}\right] \begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} &= \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{v}} \\ \bar{\boldsymbol{\omega}} \end{bmatrix} \\ \begin{bmatrix} \mathbf{B}\boldsymbol{v}_e + (\mathbf{A}+\mathbf{C})\,\boldsymbol{\omega}_e \\ \mathbf{B}\boldsymbol{\omega}_e \end{bmatrix} &= \begin{bmatrix} \mathbf{K}_{11}\bar{\boldsymbol{v}} + \mathbf{K}_{12}\bar{\boldsymbol{\omega}} \\ \mathbf{K}_{21}\bar{\boldsymbol{v}} + \mathbf{K}_{22}\bar{\boldsymbol{\omega}} \end{bmatrix} \end{aligned} \tag{2.26}$$

La première ligne de (2.26) conduit à :

$$\begin{aligned} \mathbf{K}_{11}\bar{\boldsymbol{v}} + \mathbf{K}_{12}\bar{\boldsymbol{\omega}} &= \mathbf{B}(\bar{\boldsymbol{\omega}})\boldsymbol{v}_e + (\mathbf{A}(\bar{\boldsymbol{v}}) + \mathbf{C}(\bar{\boldsymbol{\omega}}))\,\boldsymbol{\omega}_e \\ &= \mathbf{A}(\bar{\boldsymbol{v}})\boldsymbol{\omega}_e + (\mathbf{B}(\bar{\boldsymbol{\omega}})\boldsymbol{v}_e + \mathbf{C}(\bar{\boldsymbol{\omega}})\boldsymbol{\omega}_e) \end{aligned} \tag{2.27}$$

Par identification, on a donc :

$$\mathbf{K}_{11}\bar{\boldsymbol{v}} = \mathbf{A}\boldsymbol{\omega}_e \tag{2.28}$$

$$\mathbf{K}_{12}\bar{\boldsymbol{\omega}} = \mathbf{B}\boldsymbol{v}_e + \mathbf{C}\boldsymbol{\omega}_e \tag{2.29}$$

De la même façon, par identification de la deuxième ligne de (2.26) on a :

$$\mathbf{K}_{21}\bar{\boldsymbol{v}} = 0 \tag{2.30}$$

$$\mathbf{K}_{22}\bar{\boldsymbol{\omega}} = \mathbf{B}\boldsymbol{\omega}_e \tag{2.31}$$

On a donc $\mathbf{K}_{21} = 0$. Nous déterminons maintenant les matrices $\mathbf{K}_{11}$, $\mathbf{K}_{12}$ et $\mathbf{K}_{22}$. Injecter (2.23) dans (2.28) revient à :

$$\begin{aligned} \mathbf{K}_{11}\bar{\boldsymbol{v}} &= -\left[\bar{\boldsymbol{v}}\right]_\times {}^s\mathbf{R}_e\boldsymbol{\omega}_e = \left[{}^s\mathbf{R}_e\boldsymbol{\omega}_e\right]_\times \bar{\boldsymbol{v}} \\ \mathbf{K}_{11} &= \left[{}^s\mathbf{R}_e\boldsymbol{\omega}_e\right]_\times \end{aligned} \tag{2.32}$$

De la même façon, (2.24), (2.25) et (2.29) conduisent à :

$$\begin{aligned} \mathbf{K}_{12}\bar{\boldsymbol{\omega}} &= {}^s\mathbf{R}_e\left[\bar{\boldsymbol{\omega}}\right]_\times \boldsymbol{v}_e + \left(\left[\left[{}^s\mathbf{t}_e\right]_\times \bar{\boldsymbol{\omega}}\right]_\times {}^s\mathbf{R}_e + \left[{}^s\mathbf{t}_e\right]_\times {}^s\mathbf{R}_e\left[\bar{\boldsymbol{\omega}}\right]_\times\right)\boldsymbol{\omega}_e \\ &= -{}^s\mathbf{R}_e\left[\boldsymbol{v}_e\right]_\times \bar{\boldsymbol{\omega}} + \left[\left[{}^s\mathbf{t}_e\right]_\times \bar{\boldsymbol{\omega}}\right]_\times {}^s\mathbf{R}_e\boldsymbol{\omega}_e - \left[{}^s\mathbf{t}_e\right]_\times {}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times \bar{\boldsymbol{\omega}} \\ &= -{}^s\mathbf{R}_e\left[\boldsymbol{v}_e\right]_\times \bar{\boldsymbol{\omega}} - \left[{}^s\mathbf{t}_e\right]_\times {}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times \bar{\boldsymbol{\omega}} - \left({}^s\mathbf{t}_e{}^\top {}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)\bar{\boldsymbol{\omega}} + \left(\bar{\boldsymbol{\omega}}^\top {}^s\mathbf{R}_e\boldsymbol{\omega}_e\right){}^s\mathbf{t}_e \\ &= -{}^s\mathbf{R}_e\left[\boldsymbol{v}_e\right]_\times \bar{\boldsymbol{\omega}} - \left[{}^s\mathbf{t}_e\right]_\times {}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times \bar{\boldsymbol{\omega}} - \left({}^s\mathbf{t}_e{}^\top {}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)\bar{\boldsymbol{\omega}} + {}^s\mathbf{t}_e\left(\left({}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)^\top\bar{\boldsymbol{\omega}}\right) \\ &= -{}^s\mathbf{R}_e\left[\boldsymbol{v}_e\right]_\times \bar{\boldsymbol{\omega}} - \left[{}^s\mathbf{t}_e\right]_\times {}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times \bar{\boldsymbol{\omega}} - \left({}^s\mathbf{t}_e{}^\top {}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)\bar{\boldsymbol{\omega}} + \left({}^s\mathbf{t}_e\left({}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)^\top\right)\bar{\boldsymbol{\omega}} \\ &= -\left({}^s\mathbf{R}_e\left[\boldsymbol{v}_e\right]_\times + \left[{}^s\mathbf{t}_e\right]_\times {}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times - \left({}^s\mathbf{t}_e{}^\top {}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)\mathbb{I}_3 + {}^s\mathbf{t}_e\left({}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)^\top\right)\bar{\boldsymbol{\omega}} \\ \mathbf{K}_{12} &= -\left({}^s\mathbf{R}_e\left[\boldsymbol{v}_e\right]_\times + \left[{}^s\mathbf{t}_e\right]_\times {}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times - \left({}^s\mathbf{t}_e{}^\top {}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)\mathbb{I}_3 + {}^s\mathbf{t}_e\left({}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)^\top\right) \end{aligned} \tag{2.33}$$

Enfin, injecter (2.24) dans (2.31) permet d'exprimer $\mathbf{K}_{22}$:

$$\begin{aligned} \mathbf{K}_{22}\bar{\boldsymbol{\omega}} &= {}^s\mathbf{R}_e\left[\bar{\boldsymbol{\omega}}\right]_\times \boldsymbol{\omega}_e = -{}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times \bar{\boldsymbol{\omega}} \\ \mathbf{K}_{22} &= -{}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times \end{aligned} \tag{2.34}$$

La matrice $\mathbf{K}$ est ainsi déterminée et donnée par :

$$\mathbf{K} = \begin{bmatrix} \left[{}^s\mathbf{R}_e\boldsymbol{\omega}_e\right]_\times & -\left({}^s\mathbf{R}_e\left[\boldsymbol{v}_e\right]_\times + \left[{}^s\mathbf{t}_e\right]_\times {}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times - \left({}^s\mathbf{t}_e{}^\top {}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)\mathbb{I}_3 + {}^s\mathbf{t}_e\left({}^s\mathbf{R}_e\boldsymbol{\omega}_e\right)^\top\right) \\ \mathbf{0}_{3\times 3} & -{}^s\mathbf{R}_e\left[\boldsymbol{\omega}_e\right]_\times \end{bmatrix} \tag{2.35}$$

et $\mathbf{L}_{\text{ex}} = \mathbf{L}_s\mathbf{K}$ vérifie la relation (2.18).

Ainsi, la formulation du problème de calibration extrinsèque est générique pour tout capteur dont la matrice d'interaction $\mathbf{L}_s$ est connue. Toutefois, on peut remarquer que :

- $\mathbf{K}_{11}$ et $\mathbf{K}_{22}$ sont de rang 2, sauf si $\boldsymbol{\omega}_e = 0$ où elles deviennent nulles
- $\mathbf{K}_{12}$ est au plus de rang 3

$\mathbf{K}$ est donc au plus de rang 5. En supposant que $\mathbf{L}_s$ est choisie pour être de rang plein 6 (au moins 6 informations capteur indépendantes sont utilisées), $\mathbf{L}_{\text{ex}}$ est au plus de rang 5. Pour la calibration extrinsèque, on doit donc utiliser une fenêtre d'au moins deux acquisitions dans (2.12).

2.2.4.2 Calibration main/œil

L'approche classique de la calibration main/œil, présentée en Section 2.1.2.2, est d'estimer les paramètres extrinsèques en mesurant les poses de l'effecteur et du capteur dans différentes configurations. La pose de la caméra est ainsi utilisée dans [Tsai 89]. Dans [Andreff 01], c'est le déplacement entre deux poses qui est estimé par structure from motion.

Notre approche peut s'appliquer directement à la calibration main/œil en utilisant une forme modifiée de l'erreur de calibration (2.7). En supposant que les torseurs de l'effecteur $\mathbf{v}_e$ et du capteur $\mathbf{v}_s$ sont connus, l'erreur de calibration extrinsèque est définie par :

$$\tilde{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{ex}}})} = {}^{s}\mathbf{W}_{e(\widehat{\boldsymbol{\xi}_{\text{ex}}})}\mathbf{v}_e - \mathbf{v}_s \tag{2.36}$$

Minimiser cette erreur par asservissement visuel virtuel revient, par similitude avec (2.18), à déterminer la matrice d'interaction $\mathbf{L}_{\tilde{\varepsilon}}$ telle que :

$$\mathrm{d}\tilde{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{ex}}})} = \mathrm{d}{}^{s}\mathbf{W}_{e(\widehat{\boldsymbol{\xi}_{\text{ex}}})}\ \mathbf{v}_e = \mathbf{L}_{\tilde{\varepsilon}}\bar{\mathbf{v}} \tag{2.37}$$

D'après (2.19), $\mathbf{L}_{\tilde{\varepsilon}}$ est de fait la matrice $\mathbf{K}$ définie dans la section précédente.

Dans [Tsai 89], il est démontré que deux déplacements sont nécessaires pour réaliser la calibration extrinsèque : il en est de même avec notre approche du fait du rang insuffisant de $\mathbf{K}$.

Ce formalisme réduit par rapport au cas général exposé précédemment pourrait être appliqué à la calibration entre une caméra et une centrale inertielle. En effet, dans ce cadre on suppose généralement que la vitesse de la caméra peut être estimée par la vision. Ces travaux n'ont malheureusement pas été validé expérimentalement et l'approche est détaillée en Annexe A.

2.3 Application à une caméra

Dans la suite de cette section nous appliquons l'approche proposée à la calibration d'une caméra. Nous explicitons tout d'abord le calcul analytique du jacobien $\mathbf{J}_{\text{in}}$ pour une caméra observant des points. Enfin, la méthode est testée sur un robot réel.

2.3.1 Paramètres intrinsèques

L'expression du jacobien donnée en Section 2.2.3 fait intervenir la dérivée de la matrice d'interaction $\mathbf{L}_s$ du capteur par rapport aux paramètres $\boldsymbol{\xi}_{\text{in}} = (\alpha_x, \alpha_y, x_c, y_c)$. Dans le cas d'une caméra observant un point de coordonnées pixelliques $\mathbf{s} = (x_p, y_p)$ tel que décrit en Section 1.3.1, la matrice d'interaction a la forme bien connue [Chaumette 02] :

$$\mathbf{L}_s = \begin{bmatrix} \mathbf{L}_{x_p} \\ \mathbf{L}_{y_p} \end{bmatrix} = \begin{bmatrix} -\frac{\alpha_x}{Z} & 0 & \frac{x_p - x_c}{Z} & \frac{(x_p-x_c)(y_p-y_c)}{\alpha_y} & -\left(\alpha_x + \frac{(x_p-x_c)^2}{\alpha_x}\right) & \frac{\alpha_x}{\alpha_y}(y_p - y_c) \\ 0 & -\frac{\alpha_y}{Z} & \frac{y_p-y_c}{Z} & \left(\alpha_y + \frac{(y_p-y_c)^2}{\alpha_y}\right) & -\frac{(x_p-x_c)(y_p-y_c)}{\alpha_x} & -\frac{\alpha_y}{\alpha_x}(x_p - x_c) \end{bmatrix} \tag{2.38}$$

$\mathbf{L}_s$ dépend de la profondeur Z du point observé, qui n'est pas mesurée dans l'image. Nous considérons donc les profondeur comme des paramètres additionnels à estimer en même temps que les paramètres intrinsèques.s

Avec $\boldsymbol{\xi}_{\text{in}} = (\alpha_x, \alpha_y, x_c, y_c, Z)$, (2.17) devient :

$$\mathbf{J}_{\text{in}} = \begin{bmatrix} \frac{\partial \mathbf{L}_{x_p}}{\partial \alpha_x}\widehat{\mathbf{v}}_s & \frac{\partial \mathbf{L}_{x_p}}{\partial \alpha_y}\widehat{\mathbf{v}}_s & \frac{\partial \mathbf{L}_{x_p}}{\partial x_c}\widehat{\mathbf{v}}_s & \frac{\partial \mathbf{L}_{x_p}}{\partial y_c}\widehat{\mathbf{v}}_s & \frac{\partial \mathbf{L}_{x_p}}{\partial Z}\widehat{\mathbf{v}}_s \\ \frac{\partial \mathbf{L}_{y_p}}{\partial \alpha_x}\widehat{\mathbf{v}}_s & \frac{\partial \mathbf{L}_{y_p}}{\partial \alpha_y}\widehat{\mathbf{v}}_s & \frac{\partial \mathbf{L}_{y_p}}{\partial x_c}\widehat{\mathbf{v}}_s & \frac{\partial \mathbf{L}_{y_p}}{\partial y_c}\widehat{\mathbf{v}}_s & \frac{\partial \mathbf{L}_{y_p}}{\partial Z}\widehat{\mathbf{v}}_s \end{bmatrix} \tag{2.39}$$

avec, d'après (2.38) :

$$\begin{cases} \frac{\partial \mathbf{L}_{x_p}}{\partial \alpha_x} &= \begin{bmatrix} -\frac{1}{Z} & 0 & 0 & 0 & -\left(1-x^2\right) & y \end{bmatrix} \\ \frac{\partial \mathbf{L}_{x_p}}{\partial \alpha_y} &= \begin{bmatrix} 0 & 0 & 0 & -\frac{\alpha_x}{\alpha_y} xy & 0 & -\frac{\alpha_x}{\alpha_y} y \end{bmatrix} \\ \frac{\partial \mathbf{L}_{x_p}}{\partial x_c} &= \begin{bmatrix} 0 & 0 & -\frac{1}{Z} & -y & 2x & 0 \end{bmatrix} \\ \frac{\partial \mathbf{L}_{x_p}}{\partial y_c} &= \begin{bmatrix} 0 & 0 & 0 & -\frac{\alpha_x}{\alpha_y} x & 0 & -\frac{\alpha_x}{\alpha_y} \end{bmatrix} \\ \frac{\partial \mathbf{L}_{x_p}}{\partial Z} &= \begin{bmatrix} \alpha_x \frac{1}{Z^2} & 0 & -\alpha_x \frac{x}{Z^2} & 0 & 0 & 0 \end{bmatrix} \end{cases}, \begin{cases} \frac{\partial \mathbf{L}_{y_p}}{\partial \alpha_x} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{\alpha_y}{\alpha_x} xy & \frac{\alpha_y}{\alpha_x} x \end{bmatrix} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial \alpha_y} &= \begin{bmatrix} 0 & -\frac{1}{Z} & 0 & \left(1-y^2\right) & 0 & -x \end{bmatrix} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial x_c} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{\alpha_y}{\alpha_x} y & \frac{\alpha_y}{\alpha_x} \end{bmatrix} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial y_c} &= \begin{bmatrix} 0 & 0 & -\frac{1}{Z} & -2y & x & 0 \end{bmatrix} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial Z} &= \begin{bmatrix} 0 & \alpha_y \frac{1}{Z^2} & -\alpha_y \frac{y}{Z^2} & 0 & 0 & 0 \end{bmatrix} \end{cases}$$

où $x = \frac{x_p - x_c}{\alpha_x}$ et $y = \frac{y_p - y_c}{\alpha_y}$. On a donc l'expression du jacobien $\mathbf{J}_{\text{in}}$:

$$\mathbf{J}_{\text{in}} = \begin{bmatrix} -\frac{v_x}{Z} - \left(1-x^2\right)\omega_y + y\omega_z & \frac{\alpha_y}{\alpha_x} x\left(y\omega_y + \omega_z\right) \\ -\frac{\alpha_x}{\alpha_y} y\left(x\omega_x + \omega_z\right) & -\frac{v_y}{Z} + \left(1-y^2\right)\omega_x - x\omega_z \\ -\frac{v_z}{Z} - y\omega_x + 2x\omega_y & \frac{\alpha_y}{\alpha_x}\left(y\omega_y + \omega_z\right) \\ -\frac{\alpha_x}{\alpha_y}\left(x\omega_x + \omega_z\right) & -\frac{v_z}{Z} - 2y\omega_x + x\omega_y \\ \frac{\alpha_x}{Z^2}\left(v_x - xv_z\right) & \frac{\alpha_y}{Z^2}\left(v_y - yv_z\right) \end{bmatrix}^\top \tag{2.40}$$

En supposant que $\mathbf{J}_{\text{in}}$ est de rang 2, un minimum de 4 points est nécessaire pour obtenir un jacobien de rang plein dans la minimisation (2.12). Les 8 informations visuelles permettent alors d'estimer les 4 paramètres intrinsèques et les 4 profondeurs inconnues.

2.3.2 Comparaison avec des approches existantes

L'application à une caméra étant maintenant détaillée, nous comparons la formulation du problème avec celles de [Folio 08] et [De Luca 08].

2.3.2.1 Estimation de la profondeur par équation différentielle

Dans [Folio 08], l'objectif est de pouvoir reconstruire l'évolution des informations visuelles en cas d'occultation. Les paramètres intrinsèques sont supposés connus, et on s'intéresse à la résolution de l'équation reliant la dynamique de l'état d'un point $\mathbf{x} = (x, y, Z)$ au torseur cinématique $\mathbf{v}_s$. Dans le cas de la seule estimation de la profondeur Z, cela revient à exprimer la matrice d'interaction $\mathbf{L}_Z$ telle que :

$$\dot{Z} = \mathbf{L}_Z \mathbf{v}_s \tag{2.41}$$

avec :

$$\mathbf{L}_Z = \begin{bmatrix} 0 & 0 & -1 & -yZ & xZ & 0 \end{bmatrix} \tag{2.42}$$

Le modèle cinématique ainsi exprimé, il peut être vu comme une équation différentielle en Z :

$$\dot{Z} = -v_z - yZ\omega_x + xZ\omega_y \tag{2.43}$$

L'équation est résolue avec comme condition initiale, la dernière estimation connue pour la profondeur. Comme dans notre approche, la vitesse de la caméra est supposée connue. Dans le cas de la seule estimation de Z cette formulation est assez simple. Toutefois dans [Folio 08] les auteurs estiment également les coordonnées (x, y) du point dans l'image.

Dans notre approche, on suppose les points toujours visibles et le modèle cinématique est utilisé pour estimer à la fois les profondeurs et les paramètres intrinsèques.

2.3.2.2 Estimation de la focale et de la profondeur par observateur

Dans [De Luca 08] deux observateurs sont construits à partir du modèle cinématique. La caméra est partiellement calibrée : on suppose que x_c et y_c sont connus, et que $\alpha_x = \alpha_y$. L'objectif est alors d'estimer la focale $f_p = \alpha_x$ exprimée en pixels et la profondeur inverse $1/Z$. La vitesse de la caméra $\mathbf{v}_s$ étant la commande du système, elle est supposée connue.

Profondeur : Pour l'estimation de la profondeur, on définit le système d'état :

$$\mathbf{x} = \begin{bmatrix} x_p - x_c \\ y_p - y_c \\ 1/Z \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{2.44}$$

la mesure associée est $\mathbf{y} = (x, y)$. Un observateur est ensuite construit sur le modèle cinématique et une analyse d'observabilité est effectuée sur base d'un critère d'excitation persistante [Marino 96]. L'observateur pour la profondeur correspond à la loi de mise à jour de l'estimation $\hat{\mathbf{x}}$:

$$\dot{\hat{\mathbf{x}}} = \alpha(\hat{\mathbf{x}}, \mathbf{y})\mathbf{v}_s + \beta(\hat{\mathbf{x}}, \mathbf{y}, \mathbf{v}_s) \tag{2.45}$$

avec :

$$\alpha(\hat{\mathbf{x}}, \mathbf{y}) = \begin{bmatrix} -f_p\hat{x}_3 & 0 & y_1\hat{x}_3 & \frac{y_1 y_2}{f_p} & -(f_p + \frac{y_1^2}{f_p}) & y_2 \\ 0 & -f_p\hat{x}_3 & y_2\hat{x}_3 & f_p + \frac{y_2^2}{f_p} & -\frac{y_1 y_2}{f_p} & -y_1 \\ 0 & 0 & \hat{x}_3^2 & \frac{y_2\hat{x}_3}{f_p} & -\frac{y_1\hat{x}_3}{f_p} & 0 \end{bmatrix} \tag{2.46}$$

et

$$\beta(\hat{\mathbf{x}}, \mathbf{y}, \mathbf{v}_s) = \begin{bmatrix} k_1(x_1 - \hat{x}_1) \\ k_2(x_2 - \hat{x}_2) \\ k_3\left((-f_p v_x + y_1 v_z)(x_1 - \hat{x}_1) + (-f_p v_y + y_2 v_z)(x_2 - \hat{x}_2)\right) \end{bmatrix} \tag{2.47}$$

où k_1, k_2 et k_3 sont des gains. La dernière ligne du paramètre de mise à jour $\beta(\hat{\mathbf{x}}, \mathbf{y}, \mathbf{v}_s)$ contient des termes similaires à la dernière ligne de (2.40), qui correspond à l'estimation de la profondeur : $\alpha_x\,(v_x - x v_z)$ et $\alpha_y\,(v_y - y v_z)$ pour notre méthode par jacobien, et $(-f_p v_x + y_1 v_z)$ et $(-f_p v_x + y_1 v_z)$ pour l'observateur. On a en effet :

$$\begin{cases} -f_p v_x + y_1 v_z = -f_p v_x + (x_p - x_c)v_z = -f_p(v_x + x v_z) = -(\alpha_x\,(v_x - x v_z)) \\ -f_p v_y + y_2 v_z = -f_p v_y + (y_p - y_c)v_z = -f_p(v_y + y v_z) = -(\alpha_y\,(v_y - y v_z)) \end{cases} \tag{2.48}$$

Ces termes régissent dans les deux approches l'évolution de l'estimation de la profondeur.

Focale : Pour l'observation de la focale supposée variable, l'état est défini par :

$$\mathbf{x} = \begin{bmatrix} x_p - x_c \\ y_p - y_c \\ f_p \\ 1/f_p \end{bmatrix} \tag{2.49}$$

La vitesse $\mathbf{v}_s$ est limitée à sa seule partie angulaire $\boldsymbol{\omega}_s$. Cela permet d'estimer la focale sans être perturbé par la non-connaissance de la profondeur. Un parallèle similaire peut être dressé avec notre approche.

Dans la Section 2.4 nous réduirons le problème de calibration à l'estimation de la profondeur et de la focale : les critères d'observabilité déterminés sont alors les mêmes que dans [De Luca 08]. Cette approche d'observabilité permet d'estimer tout d'abord la focale par des mouvements ne faisant par intervenir la profondeur Z, puis d'estimer les profondeurs lors du fonctionnement du robot. Cela rentre toutefois en contradiction avec l'idée d'une focale variable pendant l'utilisation.

Dans notre approche les paramètres de calibration sont supposés constants. Dans la section suivante nous présentons les résultats obtenus avec la méthode proposée.

2.3.3 Calibration pendant un asservissement visuel

Dans cette section, l'approche proposée est appliquée à un robot réalisant un asservissement visuel [Chaumette 06]. La cible est composée de 4 points formant un carré, les informations visuelles utilisées sont les coordonnées pixelliques des points. La loi de commande utilisée est la loi classique exposée en Section 1.2.1 : elle fait intervenir les paramètres de calibration qui sont injectés au fur et à mesure de l'estimation.

La calibration est réalisée en parallèle de la tâche, à partir des images acquises et de la mesure de la vitesse de l'effecteur. Dans un premier temps des résultats de simulation sont présentés afin de valider l'approche. La calibration est ensuite réalisée sur un robot réel.

2.3.4 Simulation

En simulation, nous validons tout d'abord la minimisation proposée en Section 2.2.2. Les résultats de calibration au long de l'asservissement visuel sont ensuite exposés. Les valeurs réelles des paramètres sont les suivantes :

$$\begin{cases} \boldsymbol{\xi}_{\text{in}} & = (595, 607, 192, 144) \\ \boldsymbol{\xi}_{\text{ex}} & = (0.5m, -0.3m, 0.1m, -30°, 45°, 60°) \\ Z(t=0) & = (3.03m, 2.86m, 2.84m, 3.02m) \end{cases} \tag{2.50}$$

L'estimation initiale étant :

$$\begin{cases} \widehat{\boldsymbol{\xi}_{\text{in}}} & = (500, 500, 200, 250) \\ \widehat{\boldsymbol{\xi}_{\text{ex}}} & = (0.45m, -0.2m, -0.1m, -22°, 50°, 50°) \\ \widehat{Z}(t=0) & = 3m \end{cases} \tag{2.51}$$

Les simulations sont réalisées en C++ avec la librairie ViSP [Marchand 05].

2.3.4.1 Minimisation

La Figure 2.2 représente la première descente de gradient pour la calibration intrinsèque, après acquisition de la deuxième image. Les paramètres extrinsèques sont supposés connus. On constate la décroissance exponentielle de l'erreur de calibration $\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}})}$. Les estimations des paramètres convergent vers les valeurs réelles sans toutefois y parvenir. Les profondeurs

quant à elles ne sont pas parfaitement estimées après cette première optimisation. Cet écart vient de l'absence d'estimation des profondeurs de la 1ère image, seules les profondeurs de la 2ème image étant intégrées dans le vecteur des paramètres. Avec la mesure de vitesse entre les deux images, une estimation des profondeurs initiales pourrait toutefois être déduite de l'estimation courante des profondeurs de la 2ème image.

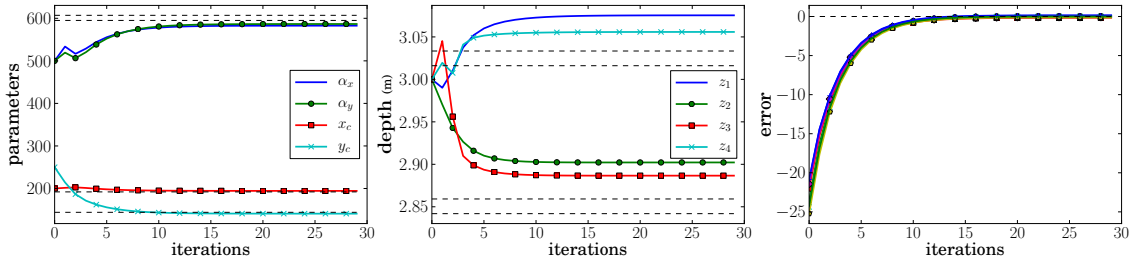


FIG. 2.2 – Descente de gradient pour la calibration intrinsèque. Évolution de $\widehat{\boldsymbol{\xi}_{\text{in}}}$, des profondeurs estimées et des composantes de $\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{in}}})}$.

La Figure 2.3 représente la première descente de gradient pour la calibration extrinsèque, après acquisition de la troisième image (deux mouvements étant nécessaires pour estimer les paramètres extrinsèques). Ici les paramètres intrinsèques et les profondeurs sont supposés connus. Comme précédemment, l'erreur de calibration $\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{ex}}})}$ décroît de façon exponentielle. Cette fois-ci les paramètres extrinsèques $\boldsymbol{\xi}_{\text{ex}}$ convergent tous vers la valeur réelle dès la première minimisation.

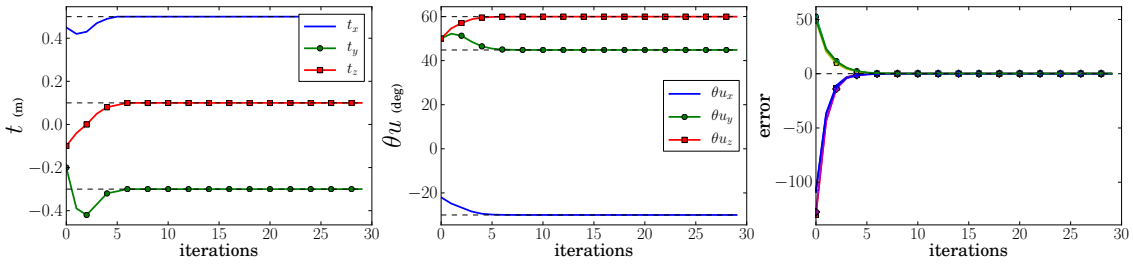


FIG. 2.3 – Descente de gradient pour la calibration extrinsèque. Évolution de $\widehat{\boldsymbol{\xi}_{\text{ex}}}$ et des composantes de $\boldsymbol{\varepsilon}_{(\widehat{\boldsymbol{\xi}_{\text{ex}}})}$.

L'approche exposée en Section 2.2.2 est ainsi validée. Nous nous intéressons maintenant à l'évolution des paramètres durant la tâche d'asservissement visuel.

2.3.4.2 Évolution pendant l'asservissement visuel

La minimisation exposée est répétée à chaque itération de l'asservissement visuel. Pour une meilleure robustesse, nous utilisons une fenêtre de taille $h=4$ pour l'optimisation (2.12).

Les résultats de la calibration intrinsèque sont exposés Figure 2.4. Les valeurs estimées après les deux premières acquisitions correspondent aux valeurs finales de la minimisation en Figure 2.2. Ici l'estimation n'est pas lissée : $\alpha = 0$ dans (2.14) et (2.15). Les paramètres

intrinsèques convergent rapidement vers les valeurs réelles. Un écart de quelques centimètres (soit quelques % des valeurs réelles) reste présent pour l'estimation des profondeurs des points. Ceci est dû au fait que les profondeurs n'étant pas constantes, les valeurs passées sont conservées et induisent un biais dans l'estimation courante qui utilise une fenêtre glissante de taille 4. Comme déjà évoqué, les estimations passées pourraient être corrigées avec la mesure de vitesse.

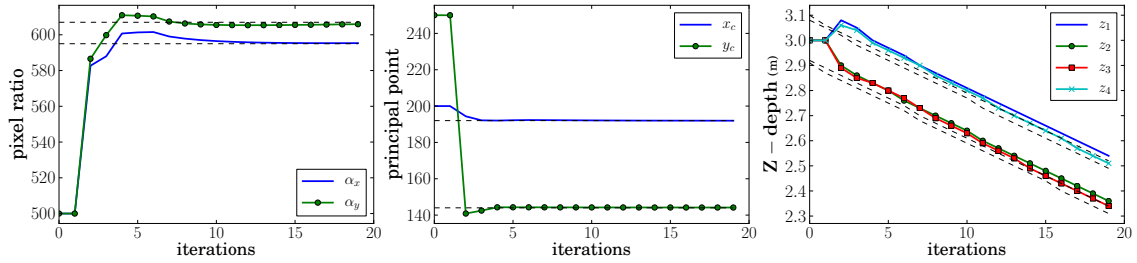


FIG. 2.4 – Calibration intrinsèque pendant l'asservissement visuel. Les paramètres $\boldsymbol{\xi}_{\text{in}}$ convergent rapidement vers leur valeur réelle. Un écart reste présent pour les profondeurs.

La Figure 2.5 représente les résultats d'une calibration simultanée des paramètres intrinsèques et extrinsèques, où on suppose les profondeurs des points connues. Comme précédemment, les paramètres convergent rapidement vers les valeurs réelles. Finalement, en cas d'estimation simultanée des paramètres intrinsèques, extrinsèques et des profondeurs, on se heurte au problème d'observabilité à un facteur d'échelle près bien connu en structure from motion.

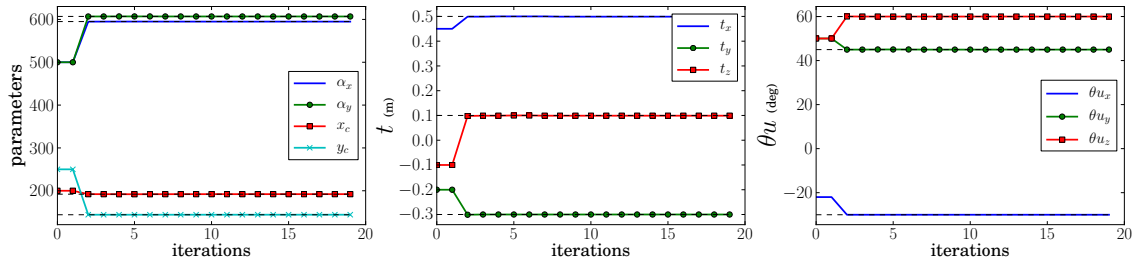


FIG. 2.5 – Calibration simultanée pendant l'asservissement visuel. Tous les paramètres convergent rapidement vers leur valeur réelle.

En l'absence de bruit de mesure, la calibration est donc parfaite même si un biais reste présent sur les estimations des profondeurs. Dans la section suivante nous étudions l'influence du bruit de mesure sur la précision de la calibration intrinsèque.

2.3.4.3 Influence du bruit de mesure

L'approche proposée repose sur la mesure de la vitesse de l'effecteur. Dans cette section nous ajoutons du bruit sur les composantes de $\mathbf{v}_e$. La vitesse bruitée $\widetilde{\mathbf{v}}_e$ est définie par :

$$\widetilde{\mathbf{v}}_e = \begin{cases} \widetilde{\boldsymbol{v}_e} & = \boldsymbol{v}_e + \nu||\boldsymbol{v}_e||\mathbf{n} \\ \widetilde{\boldsymbol{\omega}_e} & = \boldsymbol{\omega}_e + \nu||\boldsymbol{\omega}_e||\mathbf{n} \end{cases} \tag{2.52}$$

où **n** est un vecteur de dimension 3 de bruit blanc gaussien de moyenne nulle et de variance 1. $\nu \in [0, 1]$ est le rapport signal-bruit.

La Figure 2.6 illustre l'erreur de calibration intrinsèque pour des valeurs de ν comprises entre 0 et 5%. On constate que l'erreur augmente rapidement dès que la mesure de vitesse est bruitée.

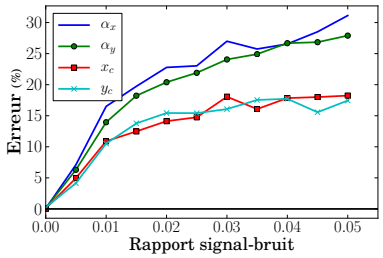


FIG. 2.6 – Calibration intrinsèque. Erreur d'estimation en % pour différentes valeurs de ν.

Pour $\nu = 3\%$, la Figure 2.7 représente l'évolution des paramètres intrinsèques pendant la calibration : si les estimées sont globalement centrées sur les valeurs réelles, des écarts importants existent.Un lissage drastique ($\alpha = 0.1$) permet de diminuer les oscillations comme représenté en Figure 2.8.

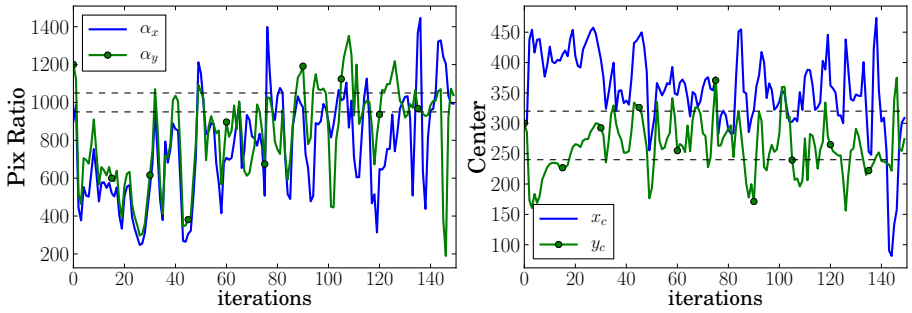


FIG. 2.7 – Calibration intrinsèque pendant l'asservissement visuel pour $\nu = 3\%$. Sans lissage, les paramètres oscillent de façon importante autour des valeurs réelles.

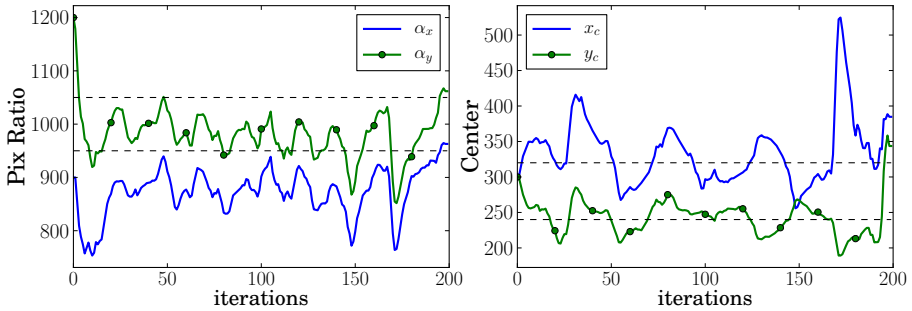


FIG. 2.8 – Calibration intrinsèque pendant l'asservissement visuel pour $\nu = 3\%$. Le lissage important permet de réduire les oscillations dans l'estimation.

La Figure 2.9 présente l'erreur de calibration pour les paramètres extrinsèques, en supposant les paramètres intrinsèques et les profondeurs connus. Là encore l'erreur augmente rapidement pour les composantes de translation : jusqu'à 50% d'erreur pour t_z quand $\nu = 5\%$. L'erreur d'estimation augmente également pour les composantes de rotation. Les résultats sont équivalents en simulation dans le cas main-œil.

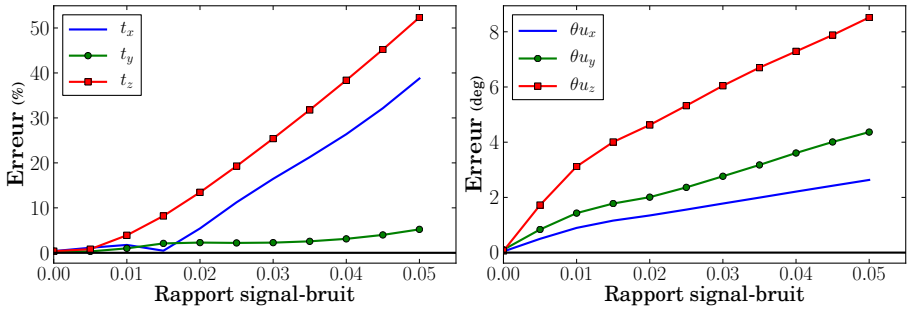


FIG. 2.9 – Calibration extrinsèque. Erreur d'estimation en % pour la translation et en degrés pour la rotation.

Le bruit de mesure de la vitesse de l'effecteur a donc une grande influence sur la qualité de la calibration. Dans la section suivante, nous appliquons l'approche proposée sur un robot réel. Comme nous le verrons, les mesures sont faiblement bruitées et les paramètres de calibration sont estimés avec une bonne précision.

2.3.5 Expérience

Les expériences sont réalisées sur le robot cartésien Afma6 de l'IRISA. Comme en simulation, le robot effectue une tâche d'asservissement visuel sur 4 points. Le suivi visuel est réalisé avec la librairie ViSP.

Afin d'avoir une vérité terrain, la caméra est tout d'abord calibrée avec une mire 3D en utilisant la méthode proposée dans [Marchand 01]. Les paramètres extrinsèques sont eux déterminés avec l'approche de [Tsai 89]. Les valeurs réelles sont ainsi :

$$\begin{cases} \boldsymbol{\xi}_{\text{in}} & = (1129, 1127, 313, 270) \\ \boldsymbol{\xi}_{\text{ex}} & = (0.009m, -0.23m, -0.01m, -70.5°, -68.8°, -69.9°) \\ Z(t=0) & = (0.49m, 0.41m, 0.41m, 0.49m) \end{cases} \tag{2.53}$$

L'estimation initiale étant :

$$\begin{cases} \widehat{\boldsymbol{\xi}_{\text{in}}} & = (1000, 1200, 300, 200) \\ \widehat{\boldsymbol{\xi}_{\text{ex}}} & = (0m, 0m, 0m, -70°, -75°, -78°) \\ \widehat{Z}(t=0) & = 0.45m \end{cases} \tag{2.54}$$

2.3.5.1 Résultats de calibration

Contrairement à la simulation, on utilise ici le paramètre $\alpha = 0.5$. Ainsi à chaque acquisition la nouvelle valeur estimée des paramètres est la moyenne entre la valeur précédemment estimée et le résultat de la dernière minimisation.

La Figure 2.10 présente les résultats de la calibration intrinsèque avec estimation des profondeurs et en supposant les paramètres extrinsèques connus. Comme en simulation, les paramètres convergent rapidement autour de leur valeur réelle. Dans le même temps les profondeurs sont estimées avec une bonne précision.

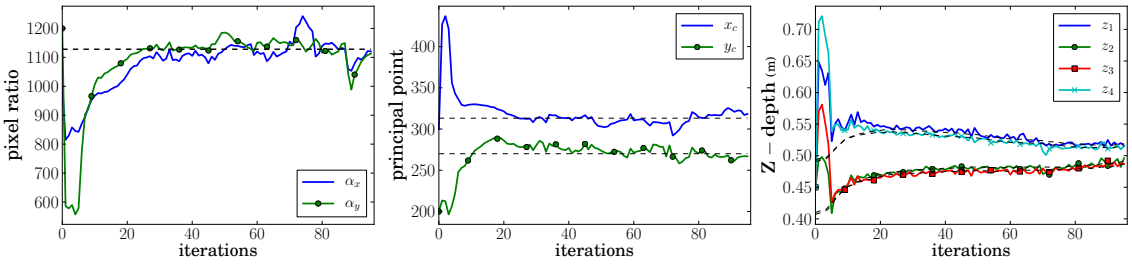


FIG. 2.10 – Calibration intrinsèque et estimation des profondeurs pendant l'asservissement visuel.

Nous procédons ensuite à la calibration extrinsèque, en supposant la caméra calibrée et les profondeurs connues. Les résultats sont reportés sur la Figure 2.11. Les paramètres de translation et de rotation convergent rapidement vers leur valeur réelle.

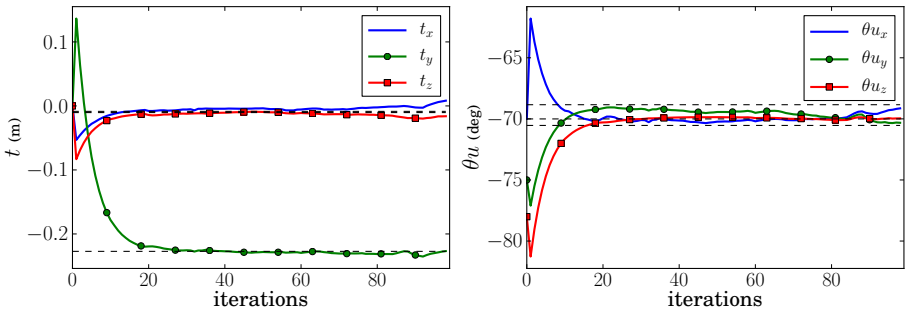


FIG. 2.11 – Calibration extrinsèque pendant l'asservissement visuel.

À l'itération 100 de l'asservissement visuel, la vitesse de l'effecteur devient trop faible pour continuer la calibration. L'asservissement visuel se poursuit alors avec les dernières valeurs estimées des paramètres.

2.3.5.2 Comportement de l'asservissement visuel

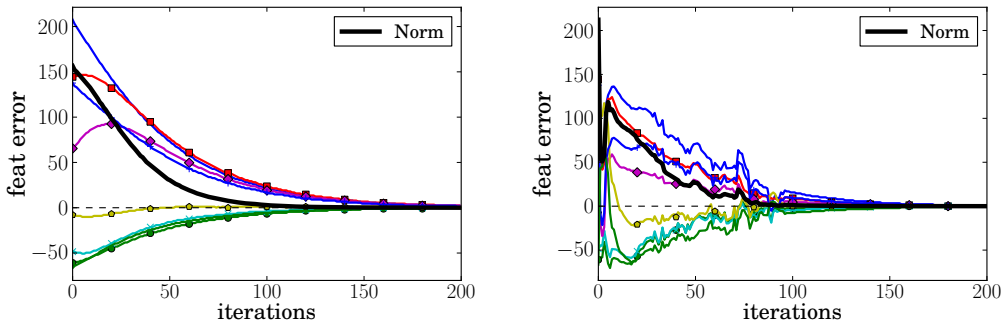
Nous étudions maintenant le déroulement de l'asservissement visuel accompagnant la calibration. Le schéma est comparé à un asservissement visuel classique sur le système déjà calibré. L'asservissement visuel 2D est connu pour être robuste aux erreurs de calibration. Aussi, les deux schémas convergent vers une erreur visuelle nulle. Nous avons vu que la matrice d'interaction dépend des profondeurs des points. C'est un problème classique en asservissement visuel 2D, dont une solution est d'utiliser dans la loi de commande la matrice $\mathbf{L}_s^*$ calculée à la position désirée. Le schéma sur système calibré est ainsi :

$$\mathbf{v}_s = -\lambda \mathbf{L}_s^{*+}(\mathbf{s} - \mathbf{s}^*) \tag{2.55}$$

Avec notre approche, les profondeurs sont estimées et le schéma suivant est utilisé :

$$\mathbf{v}_s = -\lambda \widehat{\mathbf{L}_s}^{+} (\mathbf{s} - \mathbf{s}^*) \tag{2.56}$$

La Figure 2.12 présente l'évolution de l'erreur visuelle lors de l'asservissement. Comme présenté en Section 2.3.1, les coordonnées pixelliques sont utilisées de façon à prendre en compte explicitement les paramètres de la caméra. La Figure 2.12a présente une décroissance classique d'un asservissement utilisant la matrice d'interaction désirée $\mathbf{L}_s^*$. La Figure 2.12b illustre la convergence du système pendant l'auto-calibration. La décroissance est visible malgré l'estimation des paramètres. À l'itération 100, on voit clairement la fin de l'auto-calibration, les dernières itérations ayant un comportement beaucoup plus régulier.



(a) Erreur visuelle sur système calibré

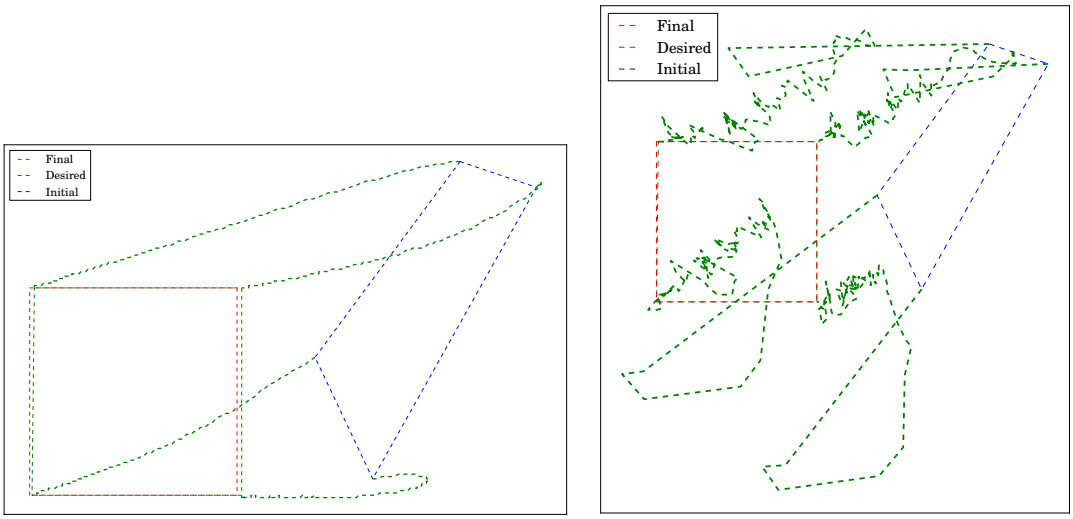
(b) Erreur visuelle avec auto-calibration

FIG. 2.12 – Erreur dans l'image. Comportement sur système calibré (gauche) et avec auto-calibration jusqu'à l'itération 100 (droite).

La trajectoire des points est représentée sur la Figure 2.13. Un comportement d'asservissement visuel classique est visible sur la Figure 2.13b. À l'inverse, sur la Figure 2.13b les premiers mouvements correspondent aux itérations où la caméra est encore très mal calibrée et où les estimations des profondeurs sont encore mauvaises. On observe qu'après quelques itérations erratiques, les points se dirigent vers leur position désirée malgré des oscillations dues à l'auto-calibration toujours en action. L'erreur visuelle étant déjà très faible à l'itération 100, les itérations correspondantes à la partie sans calibration ne sont pas perceptibles sur la Figure 2.13b.

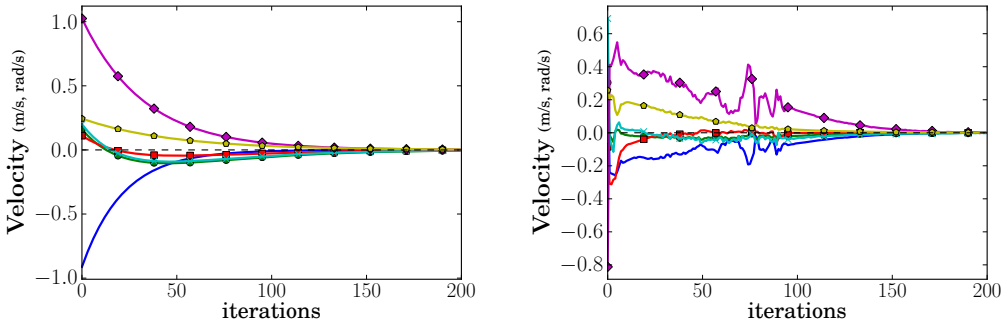
La Figure 2.14 présente la vitesse de l'effecteur pendant l'asservissement. Le comportement classique d'un système calibré est visible sur la Figure 2.14a. L'influence de l'auto-calibration est clairement visible sur la Figure 2.14b. La consigne de vitesse étant calculée en fonction des paramètres de calibration et des profondeurs estimées dans (2.56), elle n'est pas régulière comme sur un système calibré. Les vitesses générées lors de l'auto-calibration sont plus faibles que celles sur le système calibré utilisant la matrice d'interaction désirée. L'arrêt de l'auto-calibration à l'itération 100 apparaît nettement : le comportement devient alors plus régulier.

L'approche d'auto-calibration proposée permet ainsi de calibrer un capteur lors d'une tâche utilisant le capteur. La vitesse de l'effecteur est calculée par la tâche d'asservissement visuel. La mesure de vitesse est ainsi une simple entrée de l'algorithme de calibration.



(a) Trajectoire sur système calibré (b) Trajectoire avec auto-calibration

FIG. 2.13 – Trajectoire des points dans l'image. Comportement système calibré (gauche) et avec auto-calibration jusqu'à l'itération 100 (droite). Image initiale (bleue), image désirée (rouge) et trajectoires des points (vert).



(a) Vitesse sur système calibré (b) Vitesse avec auto-calibration

FIG. 2.14 – Vitesse de l'effecteur. Comportement système calibré (gauche) et avec auto-calibration jusqu'à l'itération 100 (droite).

2.3.6 Conclusion sur la calibration de caméra

Nous avons montré que le modèle cinématique était exploitable pour la calibration de caméra. Une même formulation permet d'exprimer la calibration intrinsèque et la calibration extrinsèque. Dans le cas d'une caméra observant des points, l'approche nécessite d'estimer également les profondeurs. Cela limite les possibilités d'estimation simultanée : en effet il serait nécessaire d'ajouter des contraintes pour estimer à la fois les profondeurs et les paramètres extrinsèques.

L'étude de la sensibilité au bruit montre une grande influence du bruit de mesure de la vitesse de l'effecteur. Les méthodes usuelles de calibration reposent plus souvent sur la

mesure de la position de l'effecteur qui est notoirement moins bruitée que la vitesse.

Les résultats sur un robot réel montrent toutefois qu'il est possible de calibrer une caméra et d'estimer les profondeurs des points tout en réalisant un asservissement visuel. Le schéma permet notamment d'utiliser la matrice d'interaction courante au lieu de sa valeur à la position désirée.

L'approche proposée suppose que les paramètres sont constants. Si elle ne se substitue pas aux approches classiques de pré-calibration, un cadre d'utilisation serait les capteurs à dérive lente. Les paramètres peuvent être supposés constants le long d'une séquence d'auto-calibration. Un filtre passe-bas (2.14),(2.15) a été proposé pour la mise à jour des paramètres entre deux itérations. Une modélisation de l'approche par un filtre de Kalman ou par observateur de type [De Luca 08] pourrait peut-être améliorer les performances. Notamment, l'utilisation d'une fenêtre glissante est actuellement nécessaire pour un minimum de robustesse : une estimation récursive permettrait de lever cette obligation et d'étendre l'approche aux paramètres variables.

Dans la section suivante, nous étudions l'observabilité du problème de calibration intrinsèque. La formulation permet d'appliquer ensuite le paradigme de la vision active à l'approche proposée.

2.4 Observabilité et vision active

Dans cette section nous simplifions le problème de calibration pour en étudier l'observabilité. Dans un premier temps nous présentons la formulation de la calibration partielle de caméra qui sert de support à cette étude. Les critères d'observabilité en découlent directement. La formulation obtenue permet ensuite d'appliquer le paradigme de la vision active afin d'améliorer la précision de notre approche.

2.4.1 Calibration partielle d'une caméra

Nous considérons ici que la caméra est partiellement calibrée : les coordonnées du centre optique (x_c, y_c) sont connues et on suppose les pixels carrés : $l_x = l_y$ dans (1.11). Le modèle de la caméra s'écrit alors sous la forme :

$$\begin{cases} x_p = x_c + f_p x \\ y_p = y_c + f_p y \end{cases} \tag{2.57}$$

où $f_p = \alpha_x = \alpha_y$ est la focale de la caméra exprimée en pixels. La calibration partielle consiste à estimer cette valeur. Comme vu précédemment, l'utilisation de points oblige à estimer les profondeurs correspondantes. Avec le changement de coordonnées $d = \frac{1}{Z}$ [Conticelli 01], la matrice d'interaction (2.38) devient :

$$\mathbf{L}_s = \begin{bmatrix} -f_p d & 0 & d(x_p - x_c) & \frac{(x_p-x_c)(y_p-y_c)}{f_p} & -\left(f_p + \frac{(x_p-x_c)^2}{f_p}\right) & y_p - y_c \\ 0 & -f_p d & d(y_p - y_c) & \left(f_p + \frac{(y_p-y_c)^2}{f_p}\right) & -\frac{(x_p-x_c)(y_p-y_c)}{f_p} & x_c - x_p \end{bmatrix} \tag{2.58}$$

Nous considérons les paramètres extrinsèques connus. En notant $\mathbf{v}_s$ la vitesse de la caméra, le jacobien du problème de calibration associé à (2.58) est :

$$\begin{aligned}\mathbf{J}_{\text{in}} &= \begin{bmatrix} \frac{\partial \mathbf{L}_{x_p}}{\partial f_p}\mathbf{v}_s & \frac{\partial \mathbf{L}_{x_p}}{\partial d}\mathbf{v}_s \\ \frac{\partial \mathbf{L}_{y_p}}{\partial f_p}\mathbf{v}_s & \frac{\partial \mathbf{L}_{y_p}}{\partial d}\mathbf{v}_s \end{bmatrix} \\ &= \begin{bmatrix} -dv_x - xy\omega_x - (1-x^2)\omega_y & -f_p(v_x - xv_z) \\ -dv_y + (1-y^2)\omega_x + xy\omega_y & -f_p(v_y - yv_z) \end{bmatrix} \end{aligned} \tag{2.59}$$

La forme analytique ainsi obtenue permet d'étudier l'observabilité du problème de calibration partielle. Dans la suite, nous isolons l'estimation des profondeurs d'une part, et de la focale d'autre part.

2.4.2 Observabilité

Pour un point, la focale est observable si la première colonne de $\mathbf{J}_{\text{in}}$ est non nulle. Dans le cas général, cela se traduit simplement par la présence d'une translation ou d'une rotation sur un des axes X ou Y. En d'autres termes, une valeur doit être non nulle parmi $(v_x, v_y, \omega_x, \omega_y)$.

De la même façon, la profondeur d'un point est observable si la deuxième colonne de $\mathbf{J}_{\text{in}}$ est non nulle. Le critère est donc que $v_x - xv_z$ ou $v_y - yv_z$ doit être non nul. On retrouve un résultat bien connu en structure from motion : la caméra doit avoir un mouvement de translation, et ce mouvement doit être en dehors de la ligne de vue portée par l'axe (Cz). En particulier, dans le cas d'une translation pure cela revient à avoir un mouvement du point dans l'image : $(\dot{x}_p, \dot{y}_p) \neq 0$. Là encore, on retrouve le critère d'observabilité de la profondeur exprimé dans [De Luca 08] dans le cas d'un observateur.

Les mouvements critiques étant déterminés pour la calibration partielle, nous appliquons maintenant la vision active afin d'optimiser l'estimation.

2.4.3 Optimisation par vision active

Une stratégie usuelle en calibration est de contrôler les mouvements du capteur dans le but d'améliorer l'estimation des paramètres. Ainsi des translations pures sont utilisées dans [Moons 96], et des rotations pures dans [Hartley 94]. Dans [Wei 98a], 5 translations et 2 rotations sont nécessaires pour déterminer les paramètres de calibration d'une caméra. Ces considérations sont fondées sur la structure algébrique du problème de calibration (changement de repère, projection perspective).

Dans le cas plus général, le principe de la vision active [Aloimonos 88] peut s'appliquer de façon générique à notre approche. La calibration étant exprimée comme un problème de minimisation, l'objectif est de déterminer la vitesse de caméra correspondant à l'estimateur optimal au sens de Fisher [Fisher 25]. Cela revient à maximiser le déterminant δ de la matrice d'information de Fisher définie par :

$$\mathbf{M}_F = \mathbf{J}_{\text{in}}^\top \mathbf{J}_{\text{in}} \tag{2.60}$$

avec $\delta = \det \mathbf{M}_F$. En notant Ω le domaine des vitesses de la caméra, la vitesse optimale $\mathbf{v}_s^*$

est définie par :

$$\mathbf{v}_s^* = \arg\max_{\mathbf{v}_s \in \Omega} (\delta) \tag{2.61}$$

Contrairement à la Section 2.3.3 où la calibration est réalisée pendant un asservissement visuel, le mouvement de la caméra est ici dédié à la calibration. L'objectif revient à déterminer la vitesse $\mathbf{v}_s^*$ correspondant au critère (2.61).

Une première indication provient de la forme de $\mathbf{J}_{\text{in}}$ dans (2.59). $\mathbf{v}_s$ apparaît explicitement dans le jacobien et on déduit directement que toutes choses égales par ailleurs, δ augmente avec $||\mathbf{v}_s|$. À norme égale, la vitesse optimale (2.61) doit donc permettre une meilleure estimation.

Une variation infinitésimale du déterminant $\mathrm{d}\delta$ est reliée à une variation de la consigne par :

$$\mathrm{d}\delta = \mathbf{J}_{\mathbf{v}} \mathrm{d}\mathbf{v}_s \tag{2.62}$$

où $\mathbf{J}_{\mathbf{v}}$ est le jacobien de δ par rapport à $\mathbf{v}_s$. À partir d'une valeur initiale $\mathbf{v}_{s(0)}$, un schéma itératif permet de maximiser le déterminant δ et d'obtenir la vitesse optimale de la tâche de calibration :

$$\mathbf{v}_{s(k+1)} = \pi_\Omega(\mathbf{v}_{s(k)} + \lambda \mathbf{J}_{\mathbf{v}}^\top) \tag{2.63}$$

où π_Ω est la projection sur Ω et où l'itération prend fin quand la vitesse n'évolue plus : $\|\lambda \mathbf{J}_{\mathbf{v}}^\top\| < \varepsilon$. Dans la suite, nous considérons que la caméra observe n points.

2.4.3.1 Estimation des profondeurs

Pour le point i de coordonnées (x_i, y_i) dans l'image, le jacobien $\mathbf{J}_{\text{in}(i)}$ réduit à sa deuxième colonne est :

$$\mathbf{J}_{\text{in}(i)} = \begin{bmatrix} -f_p(v_x - x_i v_z) \\ -f_p(v_y - y_i v_z) \end{bmatrix} \tag{2.64}$$

Dans le cas de n points, le jacobien total $\mathbf{J}_{\text{in}}$ est défini par :

$$\mathbf{J}_{\text{in}} = \begin{bmatrix} \mathbf{J}_{\text{in}(1)} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{J}_{\text{in}(n)} \end{bmatrix} \tag{2.65}$$

La matrice d'information (2.60) s'exprime ainsi par :

$$\mathbf{M}_F = f_p^2 \begin{bmatrix} (v_x - x_1 v_z)^2 + (v_y - y_1 v_z)^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & (v_x - x_n v_z)^2 + (v_y - y_n v_z)^2 \end{bmatrix} \tag{2.66}$$

En notant δ_i la contribution du point i, le déterminant de $\mathbf{M}_F$ est défini par :

$$\delta = f_p^{2n} \prod_{i=1}^{n} \delta_i \tag{2.67}$$

avec :

$$\delta_i = (v_x - x_i v_z)^2 + (v_y - y_i v_z)^2 \tag{2.68}$$

En notant $\bar{\delta}_i = \frac{\delta}{f_p^{2n}\delta_i} = \prod_{j\neq i}^{n} \delta_j$, le jacobien du déterminant s'exprime par :

$$\mathbf{J}_{\mathbf{v}} = f_p^{2n} \sum_{i=1}^{n} \frac{\partial \delta_i}{\partial \mathbf{v}_s} \bar{\delta}_i \tag{2.69}$$

avec, d'après (2.68) :

$$\frac{\partial \delta_i}{\partial v_x} = 2\left(v_x - x_i v_z\right) \tag{2.70}$$

$$\frac{\partial \delta_i}{\partial v_y} = 2\left(v_y - y_i v_z\right) \tag{2.71}$$

$$\frac{\partial \delta_i}{\partial v_z} = -2\left(x_i\left(v_x - x_i v_z\right) + y_i\left(v_y - y_i v_z\right)\right) \tag{2.72}$$

Le jacobien $\mathbf{J}_{\mathbf{v}}$ est ainsi déterminé et l'estimation des profondeurs peut être réalisée de façon optimale. Nous nous intéressons maintenant à l'estimation de la focale.

2.4.3.2 Estimation de la focale

Pour le point i de coordonnées (x_i, y_i) dans l'image et de profondeur $Z_i = \frac{1}{d_i}$, le jacobien $\mathbf{J}_{\mathrm{in}(i)}$ réduit à sa première colonne est :

$$\mathbf{J}_{\mathrm{in}(i)} = \begin{bmatrix} -d_i v_x - x_i y_i \omega_x - (1 - x_i^2)\omega_y \\ -d_i v_y + (1 - y_i^2)\omega_x + x_i y_i \omega_y \end{bmatrix} \tag{2.73}$$

Le jacobien global s'exprime par :

$$\mathbf{J}_{\mathrm{in}} = \sum_{i=1}^{n} \mathbf{J}_{\mathrm{in}(i)} \tag{2.74}$$

La matrice d'information $\mathbf{M}_F$ est donc une matrice de dimension 1×1, dont le déterminant δ peut s'écrire :

$$\delta = \sum_{i=1}^{n} \delta_i \tag{2.75}$$

où δ_i est la contribution du point i et s'exprime par :

$$\delta_i = \left(d_i v_x + x_i y_i \omega_x + \left(1 - x_i^2\right)\omega_y\right)^2 + \left(-d_i v_y + \left(1 - y_i^2\right)\omega_x + x_i y_i \omega_y\right)^2 \tag{2.76}$$

Le jacobien du déterminant s'exprime alors simplement par :

$$\mathbf{J}_{\mathbf{v}} = \sum_{i=1}^{n} \frac{\partial \delta_i}{\partial \mathbf{v}_s} \tag{2.77}$$

avec, d'après (2.76) :

$$\frac{\partial \delta_i}{\partial v_x} = 2d_i \left(d_i v_x + x_i y_i \omega_x + \left(1 - x_i^2\right) \omega_y \right) \tag{2.78}$$

$$\frac{\partial \delta_i}{\partial v_y} = -2d_i \left(-d_i v_y + \left(1 - y_i^2\right) \omega_x + x_i y_i \omega_y \right) \tag{2.79}$$

$$\frac{\partial \delta_i}{\partial \omega_x} = 2x_i y_i \left(d_i v_x + x_i y_i \omega_x + \left(1 - x_i^2\right) \omega_y \right) + 2\left(1 - y_i^2\right) \left(-d_i v_y + \left(1 - y_i^2\right) \omega_x + x_i y_i \omega_y \right) \tag{2.80}$$

$$\frac{\partial \delta_i}{\partial \omega_y} = 2\left(1 - x_i^2\right) \left(d_i v_x + x_i y_i \omega_x + \left(1 - x_i^2\right) \omega_y \right) + 2x_i y_i \left(-d_i v_y + \left(1 - y_i^2\right) \omega_x + x_i y_i \omega_y \right) \tag{2.81}$$

L'estimation de la focale peut alors se faire de façon optimale.

Ainsi il possible d'exprimer analytiquement la formulation optimale de l'estimation de la focale d'une part, et des profondeurs d'autre part. Toutefois, la forme analytique du jacobien global (2.59) rend difficile l'expression d'un mouvement optimal pour l'estimation simultanée de la focale et des profondeurs de n points. Nous présentons maintenant les résultats de simulation visant à améliorer les estimations.

2.4.4 Résultats de simulation

Comme exposé en Section 2.4.3, l'application de la vitesse de caméra obtenue par optimisation doit permettre une meilleure estimation de la profondeur et de la focale qu'une autre vitesse de même norme.

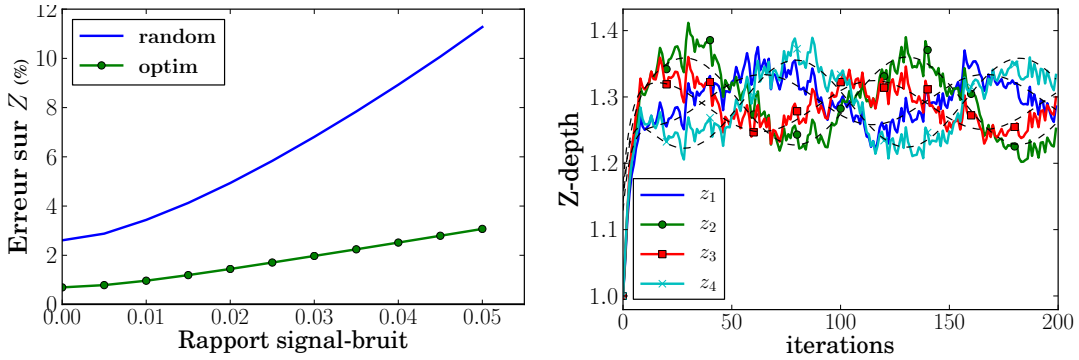
Pour ce faire, nous comparons deux schémas d'auto-calibration : le premier utilise des vitesses aléatoires de norme donnée, et le second une vitesse de même norme générée par l'algorithme exposé. Dans le schéma de vision active, les vitesses sont lissées afin d'obtenir une consigne continue. Comme on le voit en (2.70) et (2.78), l'algorithme déterminant une vitesse optimale nécessite une vitesse d'initialisation non nulle sans quoi le jacobien $\mathbf{J_v}$ serait lui aussi nul. Les vitesses sont alors générées comme suit :

- À la première itération, une vitesse aléatoire sert d'initialisation à l'algorithme qui aboutit à la vitesse de consigne $\mathbf{v}_s$.
- Aux itérations suivantes, on initialise l'algorithme avec $\beta\mathbf{v}_s$, où $\beta \in]0, 1[$, déterminant ainsi une nouvelle vitesse de consigne $\mathbf{v}_s$.
  - Si $\beta \approx 0$ alors la nouvelle initialisation se fait à une vitesse très différente de $\mathbf{v}_s$, et l'algorithme peut converger vers une nouvelle consigne très différente de la précédente.
  - Si $\beta \approx 1$ la nouvelle initialisation se fait dans le voisinnage de $\mathbf{v}_s$. Notamment, sa norme est donc proche de la norme maximale et seules quelques itérations seront effectuées avant d'aboutir à une nouvelle consigne $\mathbf{v}_s$.

Comme dans tout algorithme de gradient fortement non linéaire, une possibilité serait aussi de procéder à plusieurs tentatives à partir de différentes valeurs initiales. Ces deux schémas sont comparés pour des niveaux de bruit croissant sur une cible composée de 4 points. Le lissage est paramétré par $\beta = 0.5$.

2.4.4.1 Estimation des profondeurs

Pour l'estimation des profondeurs, la forme du jacobien (2.70) conduit naturellement à des vitesses linéaires importantes et à des vitesses angulaires nulles. Malheureusement, ce mouvement éloigne très rapidement la caméra de l'objet observé et aboutit à un système mal conditionné. Les composantes de vitesse angulaire n'entrant pas en compte dans l'estimation des profondeurs, elles sont artificiellement utilisées pour garder l'objet visible.



(a) Erreur d'estimation des profondeurs

(b) Évolution des estimations pour $\nu = 3\%$ avec la vision active.

FIG. 2.15 – Estimation des profondeurs. Le schéma optimisé (vert) est plus précis que le schéma aléatoire (bleu). Les profondeurs sont estimées avec un bruit résiduel.

La Figure 2.15 présente les erreurs d'estimation des profondeurs et compare les valeurs estimées aux vraies profondeurs pour un niveau de bruit défini par $\nu = 3\%$. Les résultats obtenus montrent une nette amélioration entre l'estimation à partir de vitesses optimales, et à partir de vitesses aléatoires.

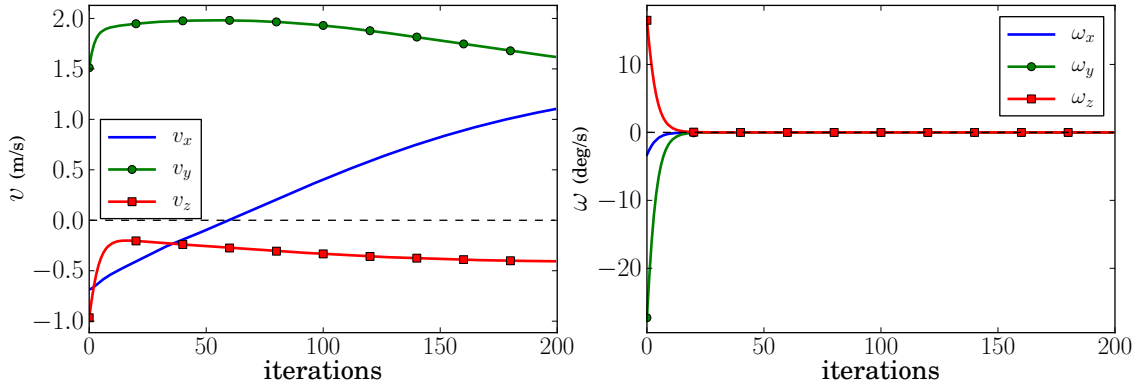


FIG. 2.16 – Vitesse optimale pour l'estimation des profondeurs. Les composantes linéaires sont favorisées au détriment des composantes angulaires.

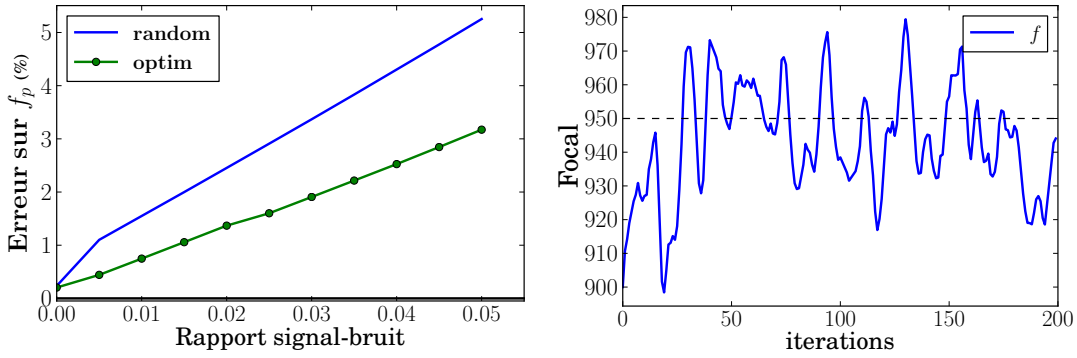
La vitesse optimale déterminée par l'algrithme est représentée en Figure 2.16. Nous rappelons que la vitesse réellement appliquée a des composantes en (ω_x, ω_y) pour centrer l'objet. Le comportement fait écho aux considérations de la Section 2.4.2 sur l'observabilité

des profondeurs. Comme dit précédemment, seules les vitesses $(v_x, v_y, v_z, \omega_z)$ sont issues de l'optimisation. Les deux autres composantes angulaires sont définies par $\omega_x = v_y/\bar{Z}$ et $\omega_y = -v_x/\bar{Z}$ où $\bar{Z}$ est la profondeur moyenne. L'objet reste donc dans le voisinage du centre de l'image. Le lissage rend les vitesses de consigne très régulières, un régime stable s'installe où les vitesses linéaires sont favorisées et où la composante ω_z est logiquement annulée. Avec la contrainte de visibilité, le mouvement induit est que la caméra tourne autour de l'objet observé.

2.4.4.2 Estimation de la focale

Dans le cas de l'estimation de la focale, les vitesses générées éloignent également la caméra de l'objet observé. Comme précédemment, nous utilisons artificiellement la composante libre v_z pour éviter un mauvais conditionnement du système.

La Figure 2.17a représente la sensibilité au bruit. La calibration partielle améliore grandement la précision par rapport à la calibration globale étudiée en Section 2.3.4.3. L'estimation par mouvement optimal donne de meilleurs résultats que par mouvement aléatoire de même norme.



(a) Erreur d'estimation de la focale

(b) Évolution de l'estimation de la focale pour $\nu = 3\%$.

FIG. 2.17 – Erreur d'estimation lors de la calibration partielle. Le schéma optimisé (vert) est plus précis que le schéma aléatoire (bleu).

Les vitesses issues de l'optimisation sont représentées sur la Figure 2.18. La composante v_z rajoutée artificiellement n'est pas indiquée. Comme précédemment, les composantes (v_z, ω_z) inutiles à la calibration sont rapidement annulées. Il est intéressant de remarquer que les deux autres composantes linéaires sont elles aussi rapidement annulées. La focale semble être estimée au mieux avec de purs mouvements de rotation. Aucun régime stable ne s'installe en revanche pour ces derniers : ω_x et ω_y oscillent tout au long de l'estimation.

2.4.5 Conclusion sur l'observabilité

L'étude d'observabilité effectuée dans cette section permet de définir des critères simples pour les mouvements critiques. La calibration partielle a également été abordée sous l'angle de la vision active. Cette approche nécessite toutefois de séparer l'estimation des

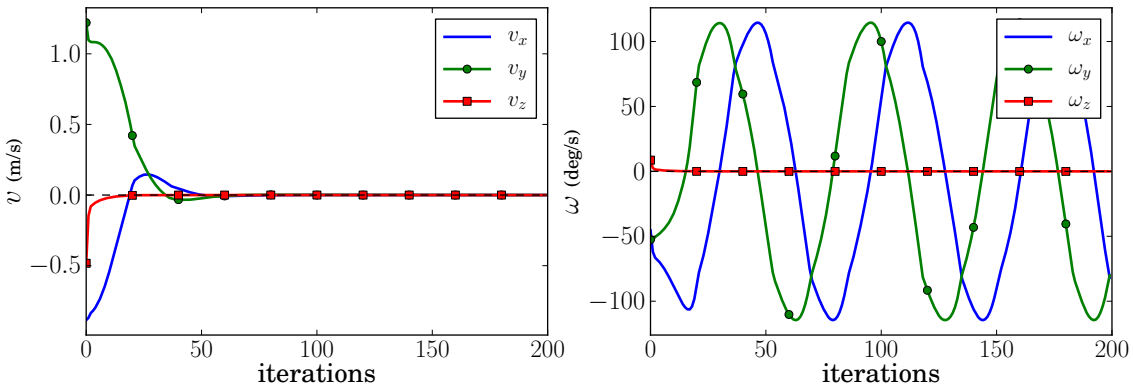


FIG. 2.18 – Vitesse optimale pour l'estimation de la focale. Seules ω_x et ω_y sont favorisées.

profondeurs et celle de la focale de la caméra pour être résolue analytiquement. Les simulations révèlent des comportements en accord avec les critères d'observabilité.

Un problème constaté dans les simulations est celui de la visibilité pendant les mouvements optimaux. En pratique, le conditionnement du système est également mis en danger si la caméra s'éloigne trop de l'objet observé. Une réponse pourrait être de limiter la vision active à certains degrés de liberté [Dune 09], et d'utiliser la redondance ou la fusion de consigne pour appliquer des mouvements optimaux sous la contrainte d'un asservissement visuel qualitatif [Remazeilles 06, Cheah 07] visant à conserver l'objet dans le champ de vue. Dans le chapitre 4 nous proposons une approche pour la visibilité, qui pourrait également être utilisée ici.

Par ailleurs, les critères d'observabilité dégagés peuvent se révéler utiles dans le cadre d'un processus d'auto-calibration où la vitesse est subie. Dans ce cadre, le schéma de calibration ou d'estimation des profondeurs ne serait appliqué qu'en cas de bonnes conditions d'observabilité.

Enfin, la résolution analytique de l'observabilité ou de la vision active pour la calibration extrinsèque n'est à ce jour pas effectuée. Au vu de la forme de la matrice d'interaction correspondante (2.35), il paraît très difficile de transposer l'approche proposée à ce problème.

2.5 Conclusion

Dans ce chapitre nous avons proposé une approche originale pour la calibration de capteurs. Exploiter le modèle cinématique rend la formulation générique pour tout capteur dont la matrice d'interaction peut être exprimée analytiquement.

En particulier, la matrice d'interaction du capteur apparaît directement dans la calibration extrinsèque. La formulation proposée est ainsi explicite dans ce cas. La calibration intrinsèque nécessite en revanche de dériver la matrice d'interaction par rapport aux paramètres intrinsèques. La formulation correspondante est donc différente en fonction du capteur utilisé.

Si la matrice d'interaction fait intervenir des grandeurs non mesurées, comme la profondeur d'un point pour une caméra, ces grandeurs doivent être également estimées. Cette estimation peut toutefois se faire avec le formalisme proposé, comme cela a été montré pour les profondeurs.

Le formalisme a été détaillé dans le cas d'une caméra observant des points. L'approche a ainsi été validée en simulation, même si une grande sensibilité au bruit de mesure peut rendre l'application difficile sur un robot peu précis. Il a été montré qu'on pouvait utiliser notre approche tout en réalisant une tâche qui utilise le capteur. Avec les mesures du capteur et les mesures de vitesse, l'auto-calibration se fait en parallèle de la tâche.

Cette application a été expérimentée sur un robot réel réalisant une tâche d'asservissement visuel. L'auto-calibration nécessitant une certaine vitesse de l'effecteur, elle doit toutefois être stoppée avant la fin de la tâche de positionnement. L'estimation des profondeurs des points est un problème classique de l'asservissement visuel. Nous avons montré que la matrice d'interaction courante pouvait être estimée avec notre approche.

En réduisant la calibration à la seule estimation de la focale et des profondeurs, il est possible d'étudier l'observabilité de la méthode proposée. Les critères dégagés sont en accord avec la littérature. Dans le cas où la vitesse de la caméra est subie, typiquement pendant un asservissement visuel, il est alors possible de déterminer les situations où l'auto-calibration ne doit pas se faire.

Simplifier le problème a aussi permis d'appliquer une stratégie de vision active, où le mouvement de la caméra est déterminé pour augmenter la précision de l'estimation. Là encore, les vitesses dégagées sont en accord avec la littérature en structure from motion. Des simulations ont montré que les estimations sont plus précises. Il serait intéressant d'utiliser la même approche pour la calibration main-œil et en déterminer les mouvements critiques.

Chapitre 3

Commande référencée multi-capteurs

Dans le chapitre précédent, un formalisme générique a été proposé pour calibrer des capteurs extéroceptifs. En confrontant les mesures des capteurs et la vitesse de l'effecteur, nous pouvons déterminer à la fois les paramètres intrinsèques et extrinsèques. Les capteurs calibrés peuvent alors être intégrés au sein d'une boucle de commande.

Ce chapitre porte sur la commande référencée multi-capteurs. Nous nous intéressons à la conception de lois de commande permettant à un robot équipé d'un ou plusieurs capteurs de réaliser des tâches variées. Ces tâches peuvent éventuellement être soumises à des contraintes, l'objectif est alors de déterminer le mouvement qui permet d'effectuer la tâche tout en respectant les contraintes. Comme exposé dans le chapitre 1, le formalisme utilisé repose sur l'approche de la fonction de tâche [Samson 91]. Les approches considérées sont uniquement réactives, et nulle planification n'est considérée.

Dans un premier temps, nous rappelons l'expression de la fonction de tâche dans le cas multi-capteurs. Ceci permet de dresser un état de l'art des différentes approches existantes, avec une notation homogène des lois de commande. Nous exposons ensuite l'approche proposée dans ce mémoire. Elle repose sur la régulation à zéro d'une erreur pondérée. Les poids sont directement appliqués aux informations capteur, permettant une grande souplesse dans la combinaison de plusieurs capteurs ainsi que la prise en compte de contraintes multiples. Les principales propriétés de la loi de commande sont données. Un accent est mis sur les possibilités données par l'utilisation de poids variables.

3.1 État de l'art

Comme exposé en Section 1.2.3, deux approches s'opposent dans l'utilisation des capteurs en robotique. La première est d'utiliser les mesures pour reconstruire l'état du robot. La tâche est définie dans l'espace des états du système. C'est donc l'état estimé par fusion qui est utilisé comme grandeur de référence dans la boucle de commande. Cette approche est liée à la commande classique et son intérêt est d'estimer l'état du système avec la meilleure précision possible.

Dans le paradigme de la commande référencée capteurs, nous cherchons à utiliser directement les informations capteur dans la commande. La tâche s'exprime dans l'espace des mesures capteurs. Cette formulation donne une grande souplesse dans la définition de la tâche et montre généralement de bonnes propriétés de robustesse et de stabilité. La fusion consiste alors à sélectionner ou combiner les mesures pour définir une consigne réalisant la tâche désirée.

Cette section dresse un état de l'art des méthodes de commande référencée multi-capteurs. Afin de pouvoir les comparer formellement, nous introduisons en premier lieu la modélisation qui sera utilisée dans le chapitre.

3.1.1 Modélisation multi-capteurs

Comme précédemment exposé en Section 1.2.1, la commande référencée capteur repose sur la régulation à zéro d'une erreur définie comme la différence $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ entre les mesures $\mathbf{s}$ du capteur et une valeur de référence $\mathbf{s}^*$. Le jacobien $\mathbf{J}_s$ du capteur relie l'évolution temporelle de $\mathbf{s}$ aux vitesses articulaires $\dot{\mathbf{q}}$ du robot :

$$\dot{\mathbf{s}} = \mathbf{J}_s \dot{\mathbf{q}} \tag{3.1}$$

En particulier, quand la matrice d'interaction $\mathbf{L}_s$ du capteur peut être exprimée, le jacobien s'écrit $\mathbf{J}_s = \mathbf{L}_s {}^s\mathbf{W}_e {}^e\mathbf{J}_q$ (voir (1.7)) où ${}^s\mathbf{W}_e$ est la matrice de changement de repère du torseur cinématique et où ${}^e\mathbf{J}_q$ est le jacobien du robot.

Dans le suite de ce mémoire, nous considérons un système composé de k capteurs. Les variables articulaires sont notées $\mathbf{q}$ avec $n = \dim(\mathbf{q})$. Chaque capteur dénoté S_i délivre un signal $\mathbf{s}_i$ de dimension m_i. On note $m = \sum_{i=1}^{k} m_i$ la dimension de l'ensemble des mesures capteurs. Pour chaque capteur un signal de référence $\mathbf{s}_i^*$ est défini, l'erreur dans l'espace du capteur est donc $\mathbf{e}_i = \mathbf{s}_i - \mathbf{s}_i^*$. Le jacobien associée est noté $\mathbf{J}_i$.

Plusieurs approches considèrent l'espace global des mesures capteurs. On note $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_k)$ le vecteur de dimension m regroupant l'ensemble des mesures. Sa variation s'exprime par :

$$\dot{\mathbf{s}} = \mathbf{J}_s \dot{\mathbf{q}} \tag{3.2}$$

où $\mathbf{J}_s$ est le jacobien étendu du système et défini par :

$$\mathbf{J}_s = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_k \end{bmatrix} \tag{3.3}$$

En particulier, quand les matrices d'interaction $\mathbf{L}_i$ des capteurs peuvent être exprimées, le jacobien étendu $\mathbf{J}_s$ peut se décomposer de la façon suivante :

$$\mathbf{J}_s = \mathbf{L}\mathbf{W}_e {}^e\mathbf{J}_q = \begin{bmatrix} \mathbf{L}_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \mathbf{L}_k \end{bmatrix} \begin{bmatrix} {}^1\mathbf{W}_e \\ \vdots \\ {}^k\mathbf{W}_e \end{bmatrix} {}^e\mathbf{J}_q \tag{3.4}$$

où $\mathbf{L} \in \mathbb{R}^{m \times 6k}$ regroupe l'ensemble des matrices d'interaction et où $\mathbf{W}_e \in \mathbb{R}^{6k \times 6}$ regroupe les matrices de transformation.

L'objectif de la commande référencée multi-capteurs est de réguler l'ensemble des erreurs $(\mathbf{e}_i)_{i \in [1,k]}$ à zéro. On note $\dot{\mathbf{e}}_i^*$ la vitesse de variation désirée pour l'erreur $\mathbf{e}_i$.

Par abus de langage[1] on assimile $\dot{\mathbf{e}}_i^*$ à la tâche associée au capteur S_i. Classiquement, la régulation à zéro de l'erreur est mise en œuvre par un correcteur proportionnel garantissant dans le cas idéal une décroissance exponentielle et découplée des composantes de $\mathbf{e}$. La vitesse de variation désirée est donc dans ce cas $\dot{\mathbf{e}}^* = -\lambda\mathbf{e}$ où λ est le gain réglant la vitesse de convergence.

On peut distinguer quatre grandes classes de méthodes pour la commande référencée multi-capteurs. La première consiste à assigner à chaque capteur la commande de certains degrés de liberté du système. On parle alors de commande partitionnée. La deuxième méthode suppose une hiérarchie entre les tâches. Dans ce cas, certains capteurs sont prioritaires sur d'autres. La difficulté est alors de permettre aux tâches secondaires de pouvoir se réaliser au mieux sous la contrainte de la tâche prioritaire. Une troisième approche est la commande commutative. Dans ce cadre, une seule tâche est appliquée à la fois et des critères sont définis pour passer d'une tâche à une autre. Ce mémoire s'inscrit dans une quatrième approche, qui consiste à réaliser un compromis entre les différentes tâches. L'enjeu est alors de définir où placer le compromis.

3.1.2 Commande partitionnée

La commande partitionnée illustre un cas particulier de fusion de capteur. Dans cette approche on suppose que chaque capteur peut contrôler certains degrés de liberté du robot. Pour le cas de deux capteurs, la commande s'écrit simplement :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+\dot{\mathbf{e}}_1^* + \mathbf{J}_2^+\dot{\mathbf{e}}_2^* \tag{3.5}$$

avec $\mathbf{J}_1^+\mathbf{J}_2 = \mathbf{J}_2^+\mathbf{J}_1 = 0$: les tâches sont indépendantes. Cette méthode est notamment utilisée pour fusionner des mesures d'effort et de vision dans [Hosoda 98]. Dans [Nadeau 11], une sonde échographique réalise une tâche de suivi ultrason. La sonde ultrason contrôle 5 degrés de liberté du système, tandis qu'un capteur de force contrôle la translation dans la direction du patient afin de réguler l'effort appliqué.

Dans [Flandin 00] deux caméras sont utilisées pour contrôler un robot. La première, déportée, contrôle les degrés de translation, et la seconde, embarquée, les degrés de rotation. Cette fois les tâches ne sont pas vraiment indépendantes : l'image acquise par la caméra embarquée est dépendante des mouvements de translation contrôlés par la caméra déportée. La commande prend alors la forme suivante :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+\dot{\mathbf{e}}_1^* + \mathbf{J}_2^+\dot{\mathbf{e}}_2^* - \mathbf{J}_2^+\widehat{\frac{\partial \mathbf{e}_2}{\partial t}} \tag{3.6}$$

où le terme additionel $-\mathbf{J}_2^+\widehat{\frac{\partial \mathbf{e}_2}{\partial t}}$ permet de compenser l'erreur de traînage.

Les méthodes partitionnées sont dépendantes de chaque application, puisqu'il faut dès la conception savoir quel capteur est à même de contrôler quels articulations. Les éventuels couplages sont directement pris en compte dans les approches hiérarchiques.

[1] Dans le formalisme de la fonction de tâche, une tâche ne peut normalement pas être de dimension supérieure à celle de l'espace articulaire.

3.1.3 Approches hiérarchiques

Imposer une hiérarchie entre les tâches revient à déterminer les mouvements disponibles qui laissent la tâche principale invariante. Cette approche repose sur la méthode du gradient projeté (GPM pour Gradient Projection Method) [Liégeois 77, Nakamura 91, Siciliano 91, Samson 91]. Selon cette méthode, le sous-espace des mouvements disponibles est calculé en même temps que la tâche principale. On parle de redondance, car la méthode nécessite que la tâche principale ne contraint pas tous les degrés de liberté du robot ($n > m_1$). Les degrés de liberté restants peuvent alors servir à réaliser les tâches secondaires. Un large éventail des méthodes de redondance est dressé dans [Marey 10a]. L'expression la plus simple est de projeter la tâche secondaire $\mathbf{e}_2$ sur le noyau de la tâche principale $\mathbf{e}_1$. Dans ce cas, la loi de commande est :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+ \dot{\mathbf{e}}_1^* + (\mathbb{I}_n - \mathbf{J}_1^+ \mathbf{J}_1)\mathbf{J}_2^+ \dot{\mathbf{e}}_2^* \tag{3.7}$$

La matrice $\mathbf{P}_1 = \mathbb{I}_n - \mathbf{J}_1^+ \mathbf{J}_1$ est l'opérateur de projection sur le noyau de la tâche principale. En effet, l'évolution de $\mathbf{e}_1$ s'écrit, d'après (3.7) :

$$\dot{\mathbf{e}}_1 = \mathbf{J}_1 \dot{\mathbf{q}} = \mathbf{J}_1 \mathbf{J}_1^+ \dot{\mathbf{e}}_1^* + \mathbf{J}_1(\mathbb{I}_n - \mathbf{J}_1^+ \mathbf{J}_1)\mathbf{J}_2^+ \dot{\mathbf{e}}_2^* = \dot{\mathbf{e}}_1^* \tag{3.8}$$

De cette façon, l'évolution de $\mathbf{e}_1$ est exactement l'évolution désirée $\dot{\mathbf{e}}_1^*$, quelle que soit le vecteur $\dot{\mathbf{e}}_2^*$. Une amélioration de cette approche est de prendre en compte les composantes de $\dot{\mathbf{e}}_2^*$ déjà réalisées par $\dot{\mathbf{e}}_1^*$. La formulation est alors :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+ \dot{\mathbf{e}}_1^* + \mathbf{P}_1(\mathbf{J}_2\mathbf{P}_1)^+(\dot{\mathbf{e}}_2^* - \mathbf{J}_2\mathbf{J}_1^+\dot{\mathbf{e}}_1^*) \tag{3.9}$$

Sur bras manipulateur, le GPM a ainsi été utilisé pour éviter les limites de butées articulaires [Liégeois 77], des obstacles ou des singularités [Nakamura 91]. En asservissement visuel ce formalisme a également été utilisé pour éviter les occultations [Marchand 98]. En robotique mobile, le GPM a été utilisé dans [Folio 05] pour éviter des obstacles dans une tâche de navigation visuelle.

On peut noter une façon originale de réaliser des tâches hiérarchisées dans [Park 01]. La consigne est définie comme une pseudo-inverse pondérée du jacobien $\mathbf{J}_1$ de la tâche principale :

$$\dot{\mathbf{q}} = \mathbf{J}_1^{\mathbf{W}+} \dot{\mathbf{e}}_1^* + \left(\mathbb{I}_n - \mathbf{J}_1^{\mathbf{W}+}\mathbf{J}_1\right) \mathbf{W}^+ \mathbf{J}_2^\top \dot{\mathbf{e}}_2^* \tag{3.10}$$

où $\mathbf{J}_1^{\mathbf{W}+}$ est la pseudo-inverse de $\mathbf{J}_1$ pondérée par $\mathbf{W}$:

$$\mathbf{J}_1^{\mathbf{W}+} = \mathbf{W}\mathbf{J}_1^+ \left(\mathbf{J}_1 \mathbf{W}^+ \mathbf{J}_1^\top\right)^+ \tag{3.11}$$

$\mathbf{W} \in \mathbb{R}^{n \times n}$ est la matrice de pondération calculée pour que la tâche principale soit réalisée sans perturbation, et la tâche secondaire réalisée au mieux :

$$\mathbf{W} = \mathbf{J}_1^\top \mathbf{J}_1 + \mathbf{J}_2^\top \mathbf{J}_2 \tag{3.12}$$

Sous un formalisme différent, (3.10) réalise la même loi de commande que (3.7).

Dans le cas de plus de deux tâches hiérarchisées, un problème apparaît car les opérateurs de projection ne sont pas commutatifs. Ainsi étendre (3.7) à 3 tâches conduit à la loi de commande suivante :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+\dot{\mathbf{e}}_1^* + \mathbf{P}_1(\mathbf{J}_2^+\dot{\mathbf{e}}_2^* + \mathbf{P}_2\mathbf{J}_3^+\dot{\mathbf{e}}_3^*) = \mathbf{J}_1^+\dot{\mathbf{e}}_1^* + \mathbf{P}_1\mathbf{J}_2^+\dot{\mathbf{e}}_2^* + \mathbf{P}_1\mathbf{P}_2\mathbf{J}_3^+\dot{\mathbf{e}}_3^* \tag{3.13}$$

Avec cette formulation, la tâche $\mathbf{e}_1$ n'est pas perturbée mais la tâche $\mathbf{e}_2$ peut être perturbée par la tâche $\mathbf{e}_3$. Des approches ont été proposées [Nakamura 87, Baerlocher 04, Mansard 07, Antonelli 09] pour garantir que la hiérarchie des tâches est bien respectée.

Les approches hiérarchiques sont particulièrement adaptées pour les robots disposant de nombreux degrés de liberté comme les humanoïdes. Ainsi dans [Baerlocher 04] un humanoïde virtuel effectue des mouvements avec un grand nombre de contraintes hiérarchisées. Dans [Sentis 05], la redondance est utilisée pour hiérarchiser des comportements et des contraintes. Des contraintes d'inégalités peuvent également être exprimées par une approche hiérarchique [Mansard 09b, Kanoun 11].

Dans le cas des robots n'ayant pas beaucoup de degrés de liberté, la contrainte de ne pas perturber la tâche principale peut devenir telle que la tâche secondaire n'est pas du tout appliquée. En effet, si $\mathbf{J}_1$ est de rang plein alors $\mathbf{P}_1 = \mathbb{I}_n - \mathbf{J}_1^+\mathbf{J}_1 = 0$, et donc la projection de toutes les tâches non prioritaires est nulle. Typiquement, un asservissement visuel utilisant 4 points est une tâche de dimension 8 : les 6 degrés de liberté du robot sont donc utilisés par la tâche principale. Des méthodes dérivées du GPM ont été proposées pour construire des opérateurs de projection moins stricts. Dans [Mansard 09a], certaines composantes de la tâche secondaire s'appliquent si elles vont dans le même sens que la tâche principale. Dans [Marey 10b] un opérateur de projection est construit pour ne pas modifier la norme de la tâche principale. Dans ce cas, la tâche secondaire dispose systématiquement de $(n-1)$ degrés de liberté. L'opérateur ainsi défini montre toutefois un problème de singularité à convergence. On doit alors commuter vers le projecteur classique au voisinage de la convergence de la tâche principale.

La plupart du temps les contraintes sont paradoxalement considérées comme des tâches secondaires. Si la réalisation de la tâche principale nécessite de violer une contrainte (i.e. si aucune autre solution n'est trouvée), alors le schéma de commande ne l'en empêchera pas. Quand le nombre de degrés de liberté n'est toujours pas suffisant pour respecter une contrainte, une solution compatible avec les méthodes GPM est de modifier l'ordre de priorité sous certains critères [Mansard 07]. Les lois de commande correspondantes rejoignent alors le champ de la commande commutative.

3.1.4 Commande commutative

Contrairement aux approches GPM, la commande commutative [Branicky 98] applique les tâches une à une. La loi de commande passe d'une tâche à une autre selon des critères prédéfinis. Dans le cas de deux tâches, la loi de commande correspondante s'écrit :

$$\dot{\mathbf{q}} = \begin{cases} \mathbf{J}_1^+\dot{\mathbf{e}}_1^* & \text{si C1} \\ \mathbf{J}_2^+\dot{\mathbf{e}}_2^* & \text{si C2} \end{cases} \tag{3.14}$$

où C1 et C2 sont les conditions qui autorisent à passer d'un schéma à un autre, classiquement avec un comportement d'hystérésis. Dans [Gans 07], un asservissement visuel

est ainsi défini avec des informations 2D pour la première tâche et des informations 3D pour la seconde. Pour conserver une erreur minimale à la fois dans l'espace opérationnel et dans l'image, le robot applique tour à tour un PBVS et un IBVS selon que l'erreur des informations 2D ou 3D est trop importante. Un asservissement visuel commutatif a également été proposé dans [Chesi 04]. Un schéma PBVS est réalisé, mais la consigne n'est appliquée que partiellement quand l'objet observé risque de sortir du champ de vue. Dans un autre registre, [Folio 08] propose un asservissement visuel pour robot mobile prenant en compte la perte de visibilité : si l'objet observé n'est plus visible, la commande passe en mode dégradé, en essayant de prévoir l'évolution des informations visuelles jusqu'à récupération dans l'image. Dans [Cherubini 11], un robot mobile est contrôlé par vision pour la navigation et répond au problème précédent. La tâche secondaire, appliquée par GPM, est une tâche d'évitement d'obstacle si un obstacle est présent, ou de centrage de la caméra dans le cas contraire. Si les méthodes commutatives sont séduisantes par leur simplicité, les conditions définissant le passage d'une tâche à l'autre ne sont pas toujours simples à déterminer. Dans [Nelson 96], le concept de résolvabilité est développé pour un capteur d'effort et une caméra. L'objectif est de déterminer quel capteur est le plus à même de contrôler le mouvement du robot dans sa position courante. Dans de nombreux cas, on souhaite simplement réaliser l'ensemble des tâches en même temps. Les commandes associées sont alors du type hybride.

3.1.5 Commande hybride

Dans ce mémoire on appelle commande hybride les schémas qui régulent l'ensemble des informations capteur à la fois, sans hiérarchie stricte dans la structure de la loi de commande. Les méthodes hybrides diffèrent par la façon de combiner les mesures ou de réaliser le compromis entre elles.

Une première classe de commande hybride consiste à fusionner les capteurs dans l'espace des vitesses articulaires. C'est l'approche choisie dans [Malis 01] pour fusionner deux caméras, puis une caméra et un capteur d'effort. Avec cette approche, la loi de commande s'écrit de la façon suivante :

$$\dot{\mathbf{q}} = h_1 \mathbf{J}_1^+ \dot{\mathbf{e}}_1^* + \ldots + h_k \mathbf{J}_k^+ \dot{\mathbf{e}}_k^* \tag{3.15}$$

où $\sum_{i=1}^{k} h_i = 1$. Les termes de pondérations apparaissent pour équilibrer les contributions des différents capteurs. Si ces poids sont souvent constants voire même égaux, le respect de certaines contraintes peut nécessiter de les rendre dépendant de la tâche. Ainsi dans [Hafez 07a, Hafez 08] les auteurs utilisent une somme pondérée pour fusionner un asservissement visuel 2D et 3D. Plus l'objet observé se rapproche de l'image, plus le poids de l'asservissement 2D augmente, ce qui permet de trouver un compromis entre la bonne trajectoire du PBVS et la contrainte de visibilité. Cette approche est séduisante pour les possibilités d'analyse qu'elle permet. Notamment, le système considéré est stable dès que tous les sous-systèmes sont stables. En revanche, le découplage n'est pas optimal.

Une deuxième approche est fondée sur le jacobien étendu [Nakamura 91, Hager 97, Seraji 98, De Luca 06]. Comme défini en (3.3), le jacobien étendu regroupe l'ensemble des

informations capteur. Toutes les composantes de $\dot{\mathbf{e}}^*$ convergent ainsi au même rythme avec la loi de commande suivante :

$$\dot{\mathbf{q}} = \mathbf{J}_s^+ \dot{\mathbf{e}}^* \tag{3.16}$$

Le jacobien étendu est aussi utilisé dans des schémas d'asservissement visuel hybride. C'est le cas dans [Malis 99] et [Morel 00], où des informations 2D et 3D sont utilisées au sein d'une même loi de commande. Cette formulation est toutefois insuffisante pour prendre explicitement en compte les contraintes. Par ailleurs, le regroupement de grandeurs de natures diverses (distances, forces, angles, pixels...) au sein d'un même vecteur d'erreur peut poser des problèmes de conditionnement. Une forme d'inverse généralisée est définie par l'introduction d'une matrice définie positive (et habituellement diagonale) $\mathbf{H}$ définissant la loi de commande :

$$\dot{\mathbf{q}} = (\mathbf{H}\mathbf{J}_s)^+ \mathbf{H}\dot{\mathbf{e}}^* \tag{3.17}$$

La matrice $\mathbf{H}$ peut être utilisée selon différentes considérations. Dans [Khatib 87] la pondération est dynamique et fonction de la matrice d'inertie du système. La loi de commande ainsi déterminée est celle qui minimise l'énergie cinétique du bras manipulateur. Cette approche rejoint également la commande optimale. Dans le paradigme de la commande linéaire quadratique (LQ) [Wittenmark 87, Papanikolopoulos 92], l'objectif de la loi de commande est de minimiser la fonction de coût :

$$F = \mathbf{e}^\top \mathbf{Q}\mathbf{e} + \dot{\mathbf{q}}^\top \mathbf{R}\dot{\mathbf{q}} \tag{3.18}$$

où $\mathbf{Q}$ et $\mathbf{R}$ sont des matrices de pondération constantes utilisées pour obtenir un comportement optimal du robot en boucle fermée. Pour le cas particulier $\mathbf{R} = 0$ et $\mathbf{Q} = \mathbf{H}^2$, minimiser (3.18) revient à appliquer la loi de commande (3.17). En commande LQ, la pondération se fait classiquement par des essais sur le domaine de fonctionnement du robot afin de déterminer le réglage optimal. Dans [Shen 07], un formalisme comparable est proposé pour équilibrer les contributions des différentes informations capteur. Les poids étant ici constants, des contraintes peuvent éventuellement être violées si les poids associés ne sont pas assez élevés.

Par analogie avec (3.15) on peut également définir la matrice $\mathbf{H}$ en isolant la contribution de chaque tâche :

$$\mathbf{H} = \begin{bmatrix} h_1 \mathbb{I}_{m_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & h_k \mathbb{I}_{m_k} \end{bmatrix} \tag{3.19}$$

où $\sum_{i=1}^{k} h_i = 1$. Des poids variables permettent alors d'équilibrer les contributions des différentes tâches. La stratégie de pondération proposée dans [Hafez 07a] pour l'asservissement visuel hybride par moyenne des vitesses (3.15) est ainsi appliquée également dans [Hafez 06, Hafez 07b] pour la même contrainte de visibilité. Toutefois les auteurs n'utilisent pas l'inverse généralisée mais la loi de commande suivante :

$$\dot{\mathbf{q}} = \mathbf{J}_s^+ \mathbf{H}\dot{\mathbf{e}}^* \tag{3.20}$$

La matrice de pondération n'étant pas prise en compte dans le jacobien, les performances sont moins satisfaisantes comme nous le verrons en Section 3.2.2.3.

Dans [Chan 95] une pseudo-inverse pondérée (WLN pour weighted least norm) est définie pour l'évitement des butées articulaires. La pondération $\mathbf{W}$ ne se fait pas sur les informations capteur mais sur les variables articulaires :

$$\dot{\mathbf{q}} = \mathbf{W}(\mathbf{J}_s\mathbf{W})^+\dot{\mathbf{e}}^* \tag{3.21}$$

où $\mathbf{W} \in \mathbb{R}^{n\times n}$ est une matrice de pondération diagonale. Une approche originale a récemment été proposée pour réaliser une tâche par jacobien étendu tout en respectant un nombre arbitraire de contraintes [Xiang 10]. Ce formalisme repose sur la pseudo-inverse pondérée (3.21), en considérant n'importe quel type de contrainte sous la forme de butées virtuelles. Cette approche est adaptée pour les systèmes redondants. Des problèmes de discontinuité de la commande sont toutefois soulevés, car le rang de la matrice à inverser peut être variable.

Ce problème d'inversion est commun à de nombreux systèmes qui intègrent une matrice de pondération dont les poids peuvent être nuls. Ces systèmes sont regroupés dans [Mansard 09c] sous le nom de tâches à dimension variable. Nous en exposons maintenant les principales propriétés.

3.1.6 Tâches à dimension variable

Les tâches d'asservissement visuel utilisent classiquement le jacobien étendu (3.16) sans pondérer l'équilibre entre les différentes informations visuelles. Exploitant au mieux la richesse des informations fournies, la loi de commande d'inverse généralisée (3.17) s'est révélée adaptée pour gérer différentes problématiques. L'idée générale est d'utiliser des poids $h_i \in [0,1]$ pour traduire l'intérêt d'utiliser une information visuelle. On parle alors d'information désactivée ou activée selon que le poids est nul ou non. Dans ce cadre, la matrice $\mathbf{H}$ est dénommée la matrice d'activation. Dans [García-Aracil 05], les points approchant du bord de l'image sont progressivement ignorés dans la loi de commande. La consigne est alors continue même en cas de perte partielle de visibilité. Dans [Remazeilles 06], l'objectif n'est pas d'atteindre une consigne visuelle exacte mais un intervalle donné pour chaque information visuelle. Au cours de l'asservissement, les informations visuelles ayant atteint leur intervalle sont progressivement ignorées, laissant au robot davantage de degrés de liberté pour faire converger les autres. La pondération se fait sur des critères statistiques dans [Comport 06]. L'indice de confiance du traitement d'image est calculé pour chaque information visuelle. Les informations sures ont alors un poids de 1, alors que les informations aberrantes ont un poids nul. Le poids de chaque composante de $\mathbf{e}$ varie ainsi entre 0 et 1 en fonction de l'indice de confiance associé. Ces travaux font l'hypothèse qu'à chaque instant, $\mathbf{HJ}_s$ reste de rang plein malgré les poids nuls. La méthode est donc appropriée pour les tâches à grande dimension, comme peuvent l'être les schémas d'asservissement visuel. Dans [Cheah 07], l'objectif est de positionner un robot en définissant une région au lieu d'une position désirée. La technique est similaire à [Remazeilles 06], toutefois la tâche est définie dans l'espace opérationnel. Le rang de $\mathbf{HJ}_s$ diminue donc dès qu'un poids est nul, et c'est la loi de commande $\dot{\mathbf{q}} = \mathbf{J}_s^\top\mathbf{H}\dot{\mathbf{e}}^*$, moins découplée mais toujours continue, qui est utilisée.

Ces lois de commande sont étudiées dans [Mansard 09c], notamment les critères de continuité sont définis. La loi de commande (3.17) est ainsi continue si :

- Le jacobien pondéré $\mathbf{HJ}_s$ est continu, ce qui est généralement le cas en définissant bien la matrice d'activation $\mathbf{H}$.
- L'erreur pondérée $\mathbf{He}$ est également continue, ce qui revient au même critère sur la définition de la matrice d'activation.
- La pseudo-inverse $(\mathbf{HJ}_s)^+$ est continue. Une condition suffisante est que $\mathbf{HJ}_s$ soit de rang plein, ce qui est le cas si $\mathbf{J}_s$ est de rang plein, et si suffisamment d'informations restent activées.

Dans la suite de ce chapitre, nous utilisons également des tâches à dimension variable. Les schémas illustrant notre approche respectent les critères de continuité de la loi de commande.

Toutefois, dans un cadre plus général, une intéressante formulation de pseudo-inverse continue est proposée dans [Mansard 09c] et peut être appliquée à la fusion multi-capteurs. En notant p le nombre de contraintes, on calcule les 2^p pseudo-inverses correspondant à toutes les combinaisons d'activation des contraintes. La pseudo-inverse continue $\mathbf{J}_s^{\oplus \mathbf{H}}$ est alors une moyenne pondérée de ces 2^p matrices. Cette approche est développée dans [Mansard 09b] pour la réalisation de tâches référencées capteur tout en respectant des contraintes d'inégalité. Il s'agit probablement là du formalisme le plus proche du nôtre. Il est également très proche de [Kanoun 11], qui considère explicitement des contraintes d'inégalité dans une tâche référencée capteurs. Ces travaux font aussi partie des approches hiérarchiques exposées en Section 3.1.3, avec moins de rigidité dans le respect des priorités.

3.2 Approche proposée

Nos travaux font partie des approches de type hybride. L'objectif est de concevoir un schéma générique réalisant un compromis entre les différents capteurs. Le formalisme de l'inverse généralisée (3.17) est utilisé.

Même si une hiérarchie n'est pas définie entre les capteurs, des contraintes doivent toutefois pouvoir être prises en compte. Bien souvent les contraintes portent sur les valeurs des informations capteur utilisées. C'est typiquement le cas en asservissement visuel pour la visibilité. Nous ne modélisons pas les contraintes comme des butées virtuelles comme les approches de type pseudo-inverse pondérée [Xiang 10]. Au contraire, toute contrainte est vue comme une information capteur et intégrée à la loi de commande à travers le jacobien étendu (3.3).

Les tâches à dimension variable ont pour l'instant été utilisées pour évincer d'une loi de commande les informations inutiles ou jugées peu sures. Nous utilisons ce formalisme pour gérer les contraintes au niveau des informations capteur.

Dans cette section, nous détaillons l'approche proposée. Nous proposons une forme générique pour la pondération de l'inverse généralisée. Plusieurs formes de lois de commande sont proposées et comparées. Les propriétés du schéma général sont ensuite exposées. Une pondération optimale est proposée pour le cas de l'évitement des butées articulaires. Nous étudions enfin les principaux problèmes inhérents à cette approche.

3.2.1 Loi de commande

Nous reprenons ici le formalisme présenté en état de l'art. Dans le cas de k capteurs, la matrice de pondération peut s'écrire :

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{H}_k \end{bmatrix} = \text{Diag}\left(h_1, \dots, h_m\right) \tag{3.22}$$

où $\mathbf{H}_i$ est la matrice diagonale de pondération du capteur S_i et où h_i est un poids générique. L'objectif de la loi de commande est de réguler à zéro l'erreur pondérée $\mathbf{e_H} = \mathbf{He}$. Nous proposons tout d'abord une forme générique pour la pondération, avant de détailler l'expression de la loi de commande.

3.2.1.1 Pondération générique

Nous avons vu que deux motifs justifient d'équilibrer les contributions des capteurs. Tout d'abord, la commande LQ (3.18) ou la fusion au niveau articulaire (3.15) tentent d'optimiser le comportement général du système en déterminant des gains spécifiques pour chaque capteur. À l'inverse, les schémas prenant en compte les contraintes utilisent des poids variables permettant de favoriser le respect des contraintes quand elles sont mises en danger.

Une pondération générique est donc d'écrire chaque poids sous la forme suivante :

$$h_i = h_i^t + h_i^c \tag{3.23}$$

où h_i^t traduit la pondération générale du capteur et où h_i^c permet de prendre en compte les contraintes. Les lois de commande basiques utilisent généralement la forme correspondant à $\mathbf{H} = \mathbb{I}_m$, c'est à dire :

$$\forall i, h_i^t = 1 \;\text{ et }\; h_i^c = 0 \tag{3.24}$$

Chaque poids $(h_i^t)_i$ peut également être réglé de façon indépendante, de façon similaire à la matrice $\mathbf{Q}$ en commande LQ. En pratique, plusieurs essais sont souvent nécessaires pour obtenir le comportement optimal attendu [Wittenmark 87]. Les lois de commande à dimension variable [García-Aracil 05, Remazeilles 06, Comport 06] utilisent une pondération entre 0 et 1 en fonction de l'intérêt de l'information capteur correspondante. L'activation et la désactivation d'une information peut également se traduire dans la partie h^t. Le réglage de la pondération générale se fait sur des critères ad hoc, en pratique, nous utilisons dans la suite le réglage suivant :

$$\forall i, h_i^t = \begin{cases} 1 & \text{si l'information } s_i \text{ est utilisée à chaque instant dans la commande} \\ 0 & \text{si l'information } s_i \text{ traduit uniquement une contrainte à respecter} \end{cases} \tag{3.25}$$

Nous proposons maintenant une forme générale pour les poids $(h_i^c)_i$ gérant les contraintes. Une contrainte s'exprime généralement par une inégalité sur une information capteur. De fait, si la contrainte ne fait pas partie des informations capteur de la tâche considérée (position d'une variable articulaire, distance entre l'objet observé et un objet

occultant...), elle peut toujours être rajoutée au vecteur des informations et modélisée comme telle avec le jacobien étendu. Sans perte de généralité, nous pouvons donc considérer qu'une contrainte générique est de maintenir la valeur d'une information capteur s dans un intervalle $\mathcal{I}^l = [s^-, s^+]$. Un intervalle sûr $\mathcal{I}^s = [s^{s-}, s^{s+}]$ peut alors être défini par :

$$\begin{cases} s^{s-} = s^- + \rho(s^+ - s^-) \\ s^{s+} = s^+ - \rho(s^+ - s^-) \end{cases} \tag{3.26}$$

où $\rho \in [0, 0.5]$ est un paramètre de réglage. Une telle définition est usuelle pour modéliser une zone de prise en compte des contraintes [Marchand 96, Mansard 09a, Marey 10c]. La fonction de pondération associée est classiquement nulle dans l'intervalle $\mathcal{I}^s$ et augmente progressivement à mesure que l'on approche des limites de $\mathcal{I}^l$. Toutefois, comme nous le verrons dans la Section 3.2.2.4 il est difficile de déterminer une fonction de poids bornée sur $[s^-, s^+]$ garantissant le respect de la contrainte correspondante. La fonction de poids est donc ici définie par :

$$h^c(s) = \begin{cases} \frac{s - s^{s+}}{s^+ - s} & \text{si } s > s^{s+} \\ \frac{s^{s-} - s}{s - s^-} & \text{si } s < s^{s-} \\ 0 & \text{sinon} \end{cases} \tag{3.27}$$

h^c est représentée sur la Figure 3.1. Le poids associé à s est nul dans l'intervalle $\mathcal{I}^s$, et augmente de façon continue jusqu'à l'infini aux limites acceptables s^- et s^+. Soulignons que $h^c = 1$ quand s est exactement au milieu de la zone d'activation. Dans le cadre du

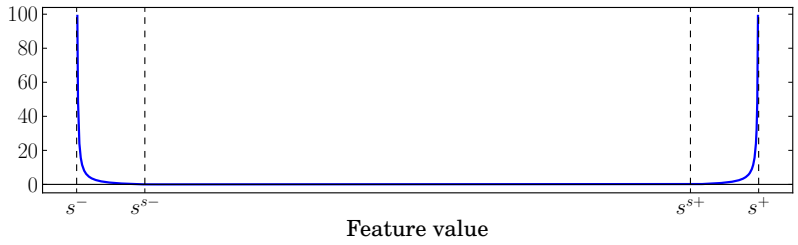


FIG. 3.1 – Pondération h^c pour la gestion des contraintes

jacobien étendu, donc d'une approche par compromis, une telle formulation permet de garantir le respect des contraintes. En effet, la pondération d'une contrainte augmente dans son voisinage. Le compromis entre la contrainte et les autres informations évolue donc en faveur du respect de la contrainte la plus proche. Cet aspect sera détaillé en Section 3.2.2.4, où nous mettons en évidence le respect d'une contrainte à partir d'un poids minimal. Dans le cas des butées articulaires, nous verrons en Section 3.2.3 que ce poids minimal peut être déterminé analytiquement. Cela permet alors une pondération adaptative optimale dans le cas de l'évitement des butées articulaires.

Enfin, les tâches à dimension variable utilisent classiquement des poids inférieurs à 1. De fait, la matrice $\mathbf{H}$ est définie à un facteur d'échelle près :

$$\forall \alpha \neq 0, \quad \mathbf{C}(\alpha) = (\alpha \mathbf{H} \mathbf{J}_s)^+ (\alpha \mathbf{H}) = (\mathbf{H} \mathbf{J}_s)^+ \mathbf{H} = \mathbf{C}(1) \tag{3.28}$$

Les poids peuvent donc être ramenés à $[0, 1]$ si cela s'avère nécessaire. Une relation d'équivalence entre les deux formalismes reste toutefois à définir. En effet, les poids peuvent

être égaux à 1 pour les tâches à dimension variable alors qu'ils ne sont évidemment jamais égaux à $+\infty$ dans notre approche. Dans le formalisme des tâches à dimension variable, la pondération importante d'une contrainte correspond donc à une faible pondération pour toutes les autres informations capteur. Des problèmes numériques peuvent toutefois apparaître quand les poids diffèrent de plusieurs ordres de grandeur. Nous verrons qu'en pratique cette situation n'a jamais été rencontrée. La forme de la matrice de pondération étant définie, nous étudions plus en détail la formulation de la loi de commande.

3.2.1.2 Choix de la loi de commande

Dans l'approche fonction de tâche [Samson 91], on définit une matrice de combinaison $\mathbf{C} \in \mathbb{R}^{n \times m}$ permettant de prendre en compte la redondance entre les informations capteur. Un contrôleur associé est alors le schéma proportionnel :

$$\dot{\mathbf{q}} = -\lambda \mathbf{C} \mathbf{e} \tag{3.29}$$

La loi de commande par inverse généralisée (3.17) correspond à cette approche avec la matrice de combinaison suivante :

$$\mathbf{C} = (\mathbf{H}\widehat{\mathbf{J}}_s)^+ \mathbf{H} \tag{3.30}$$

où $\widehat{\mathbf{J}}_s$ est une valeur estimée de $\mathbf{J}_s$. Comme nous l'avons vu dans les premiers chapitres, $\mathbf{J}_s$ dépend des paramètres de calibration du robot et des capteurs, et peut également dépendre de grandeurs non mesurées directement. Aux approximations de calibration près, plusieurs stratégies peuvent être choisies pour estimer le jacobien. Comme exposé en Section 3.1.1, on rappelle que le jacobien multi-capteurs peut généralement s'exprimer sous la forme $\mathbf{J}_s = \mathbf{L}\mathbf{W}_e {}^e\mathbf{J}_q$, où $\mathbf{L}$ regroupe les matrices d'interaction, $\mathbf{W}_e$ regroupe les matrices de transformation et où ${}^e\mathbf{J}_q$ est le jacobien du robot. Nous transposons à la commande multi-capteurs les trois approches les plus populaires issues de l'asservissement visuel [Chaumette 06]. Il s'agit du jacobien courant, du jacobien désiré et de la moyenne des deux.

Jacobien courant

Utiliser le jacobien courant revient à disposer d'une estimation en temps réel de $\mathbf{L}$ et $\mathbf{W}_e$. C'est généralement le cas pour la dernière. En effet, les matrices de transformation des capteurs sont :

- constantes pour les capteurs fixés à l'effecteur, et déterminées par calibration ;
- variables pour les capteurs déportés, mais pouvant se calculer en temps réel avec le modèle géométrique direct du robot. En effet, en supposant connue la matrice de passage (constante) ${}^s\mathbf{M}_f$ entre le capteur et le repère fixe du robot, le modèle géométrique ${}^e\mathbf{M}_f(\mathbf{q})$ permet de déterminer à tout instant la matrice ${}^s\mathbf{M}_e(\mathbf{q}) = {}^s\mathbf{M}_f {}^e\mathbf{M}_f^{-1}(\mathbf{q})$ et donc $\mathbf{W}_e$.

L'estimation en temps réel des matrices d'interaction peut en revanche poser des difficultés. Comme nous l'avons vu au chapitre 2, tous les paramètres d'une matrice d'interaction ne sont pas contenus dans les informations capteur. C'est typiquement le cas pour une caméra observant un point, mais aussi pour un capteur de distance pointant

une surface inconnue. En pratique, on estimera les paramètres inconnus avec la méthode proposée au chapitre 2, par calcul de pose ou bien encore en leur donnant une valeur constante.

Dans la suite on note $\mathbf{C}_{\text{cur}}$ la matrice de combinaison utilisant l'estimation la plus proche du jacobien courant :

$$\mathbf{C}_{\text{cur}} = (\mathbf{H}\widehat{\mathbf{L}}\mathbf{W}_e{}^e\mathbf{J}_q)^+\mathbf{H} \tag{3.31}$$

Jacobien à la position désirée

En asservissement visuel l'utilisation de la matrice d'interaction à la position désirée est une stratégie classique [Chaumette 06]. Elle permet de s'affranchir de l'estimation en temps réel des paramètres inconnus comme les profondeurs des points. Certains schémas utilisent des informations visuelles complexes dont la matrice peut être fastidieuse à calculer [Dame 11] : cette stratégie permet alors d'utiliser une matrice constante calculée une fois pour toutes à partir des informations désirées. Par ailleurs, dans certaines situations on observe que le comportement 3D de l'effecteur est plus satisfaisant en utilisant cette méthode [Chaumette 06]. La matrice d'interaction étant calculée en fonction des informations capteur $\mathbf{s}$, il est naturellement possible de savoir à l'avance sa valeur à la position désirée en utilisant les informations correspondantes $\mathbf{s}^*$.

Il en va autrement pour la matrice des transformations $\mathbf{W}_e$:

- Dans le cas embarqué, la matrice de transformation ${}^s\mathbf{W}_e$ est constante, sa valeur à la position désirée est donc connue.
- Dans le cas déporté en revanche, la connaissance de ${}^s\mathbf{W}_e^*$ revient à connaître la pose désirée. C'est parfois le cas en commande déportée [Marchand 02], mais une loi de commande utilisant la pose désirée est généralement construite sur le jacobien courant. Dans le cas général, on considère que ${}^s\mathbf{W}_e^*$ n'est pas connue pour un capteur déporté.

Au final, la matrice $\mathbf{W}_e^*$ étant égale à $\mathbf{W}_e$ si tous les capteurs sont embarquée, ou non connue dans le cas contraire, on définit la matrice de combinaison $\mathbf{C}_*$ utilisant la pose désirée par :

$$\mathbf{C}_* = (\mathbf{H}\mathbf{L}^*\mathbf{W}_e{}^e\mathbf{J}_q)^+\mathbf{H} \tag{3.32}$$

Nous mettons maintenant en avant le lien entre le jacobien moyen et la minimisation au second ordre.

Jacobien moyen et minimisation au second ordre

Dans [Malis 04a], un schéma d'asservissement visuel est proposé avec la moyenne des matrices d'interaction :

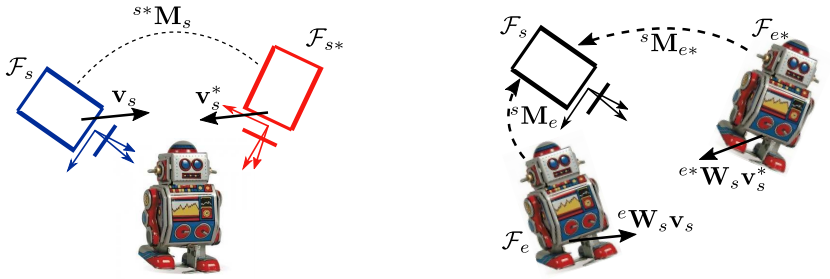
$$\mathbf{v}_s = 2(\mathbf{L}_s + \mathbf{L}_s^*)^+\dot{\mathbf{e}}^* \tag{3.33}$$

Cette approche est assimilée à une minimisation au second ordre de l'erreur visuelle.
Une approche voisine également proposée est d'utiliser la moyenne des pseudo-inverse : $\mathbf{v}_s = 1/2(\mathbf{L}_s^+ + \mathbf{L}_s^{*+})\dot{\mathbf{e}}^*$. Les observations de ce paragraphe sont transposables à cette formulation. La loi de commande (3.33) est discutée dans [Tahri 09], mettant en avant l'approximation réalisée en négligeant la transformation ${}^{s*}\mathbf{W}_s$ entre le repère courant $\mathcal{F}_s$ et le repère désiré

$\mathcal{F}_{s*}$ de la caméra. Une nouvelle minimisation au second ordre est alors proposée :

$$\mathbf{v}_s = 2(\mathbf{L}_s + \mathbf{L}_s^{*s*}\mathbf{W}_s)^{+}\dot{\mathbf{e}}^* \tag{3.34}$$

Nous exposons maintenant la forme générale d'une minimisation au second ordre dans le cas multi-capteurs. L'approche de [Malis 04a] et [Tahri 09] repose sur l'expression de $\dot{\mathbf{s}}$ en fonction de la vitesse $\mathbf{v}_s$ à appliquer à la caméra courante pour aller à la pose désirée d'une part, et et la vitesse $\mathbf{v}_s^*$ à appliquer à la caméra désirée pour aller à la pose courante d'autre part. (3.34) est alors obtenu en écrivant toutes les vitesses dans le même repère.



(a) Cas embarqué (b) Cas déporté

FIG. 3.2 – Minimisation au second ordre dans le cas embarqué (a) et déporté (b).

Dans le cas embarqué (Figure 3.2a), les vitesses sont exprimées dans $\mathcal{F}_s$:

$$\begin{cases} \dot{\mathbf{s}} &= \mathbf{L}_s\mathbf{v}_s \\ \dot{\mathbf{s}} &= -\mathbf{L}_s^*\mathbf{v}_s^* = \mathbf{L}_s^{*s*}\mathbf{W}_s\mathbf{v}_s \end{cases} \tag{3.35}$$

d'où on déduit la variation estimée :

$$\dot{\mathbf{s}} = \frac{1}{2}(\mathbf{L}_s + \mathbf{L}_s^{*s*}\mathbf{W}_s)\mathbf{v}_s \tag{3.36}$$

ce qui conduit à la loi de commande (3.34).

Dans le cas d'un capteur déporté, le repère capteur étant fixe on a bien sûr $\mathcal{F}_s = \mathcal{F}_{s*}$. Exprimer les vitesses dans le repère capteur $\mathcal{F}_s$ conduit donc à :

$$\begin{cases} \dot{\mathbf{s}} &= \mathbf{L}_s\mathbf{v}_s \\ \dot{\mathbf{s}} &= -\mathbf{L}_s^*\mathbf{v}_s \end{cases} \tag{3.37}$$

Et l'estimation au second ordre revient à utiliser la moyenne de $\mathbf{L}_s$ et $\mathbf{L}_s^*$. En exprimant chaque vitesse dans le repère effecteur associé (voir Figure 3.2b), $\dot{\mathbf{s}}$ s'écrit en revanche :

$$\begin{cases} \dot{\mathbf{s}} &= \mathbf{L}_s\mathbf{v}_s = \mathbf{L}_s\,{}^s\mathbf{W}_e\mathbf{v}_e \\ \dot{\mathbf{s}} &= -\mathbf{L}_s^*\mathbf{v}_s = -\mathbf{L}_s^{*s}\mathbf{W}_{e*}\mathbf{v}_e^* = -\mathbf{L}_s^{*s}\mathbf{W}_{e*}\mathbf{v}_e^* \end{cases} \tag{3.38}$$

Cette fois c'est en négligeant le changement de repère ${}^{e*}\mathbf{W}_e$, donc en considérant $\mathbf{v}_e \approx -\mathbf{v}_e^*$, que l'on retrouve une formulation similaire à (3.35) :

$$\dot{\mathbf{s}} = \frac{1}{2}(\mathbf{L}_s\,{}^s\mathbf{W}_e + \mathbf{L}_s^{*s}\mathbf{W}_{e*})\mathbf{v}_e \tag{3.39}$$

la loi de commande correspondante s'exprime par :

$$\mathbf{v}_s = 2{}^s\mathbf{W}_e(\mathbf{L}_s{}^s\mathbf{W}_e + \mathbf{L}_s^*{}^s\mathbf{W}_{e*})^+\dot{\mathbf{e}}^* \tag{3.40}$$

L'asservissement visuel ainsi défini a exactement le même comportement que celui étudié dans [Tahri 09] dans le cas embarqué. Soulignons que les approximations ${}^{s*}\mathbf{W}_s \approx \mathbb{I}_6$ du cas embarqué et ${}^{e*}\mathbf{W}_e \approx \mathbb{I}_6$ du cas déporté conduisent à des conclusions exactement opposées.

En pratique, nous avons vu qu'il était rare de connaître la matrice de transformation entre la pose courante et la pose désirée. La matrice de combinaison $\mathbf{C}_{\frac{1}{2}}$ correspondant au mieux à une minimisation au second ordre s'exprime donc par :

$$\mathbf{C}_{\frac{1}{2}} = 2(\mathbf{H}(\mathbf{L} + \mathbf{L}^*)\mathbf{W}_e{}^e\mathbf{J}_q)^+\mathbf{H} \tag{3.41}$$

Les comportements des trois matrices de combinaison proposées dans cette partie sont comparés en Section 4.1

Dans la suite, nous considérons que la matrice de combinaison est construite sur une estimation $\widehat{\mathbf{J}}_s$ raisonnable du jacobien. Les propriétés de la loi de commande proposée sont maintenant exposées.

3.2.2 Propriétés de la commande multi-capteurs

Dans cette section nous exposons les principales propriétés de la loi de commande (3.29) avec la matrice de pondération définie en Section 3.2.1.1. Tout d'abord, la continuité et la stabilité générale sont étudiées. Nous nous intéressons ensuite à l'influence des poids dans la commande.

3.2.2.1 Continuité

En Section 3.1.6 nous avons rappelé les conditions de continuité des tâches à dimension variable. La formulation de la matrice de pondération est continue d'après (3.23) et (3.27). En faisant l'hypothèse que les capteurs fournissent des mesures continues, $\mathbf{HJ}_s$ et $\mathbf{He}$ sont donc continus.

La pseudo-inverse est continue dès que $\mathbf{HJ}_s$ est de rang plein. En particulier cela suppose qu'au moins n poids soient activés à chaque instant car $\mathbf{J}_s \in \mathbb{R}^{m\times n}$. Dans ce mémoire nous faisons également cette hypothèse, qui est vérifiée sur les applications de notre formalisme. Cette hypothèse est relativement naturelle en fusion de capteurs, car le nombre d'informations capteur est facilement élevé notamment en utilisant la vision. Toutefois dans le cas général, la pseudo-inverse continue proposée dans [Mansard 09c] offrirait une solution adéquate.

3.2.2.2 Stabilité

Le formalisme des tâches à dimension variable néglige classiquement la variation de la matrice de pondération $\mathbf{H}$. Les hypothèses sont qu'elle varie lentement [Comport 06] ou qu'elle est nulle à la pose désirée, comme dans l'asservissement visuel qualitatif

[Remazeilles 06] ou l'asservissement sur une région [Cheah 07]. Le fait d'intégrer $\mathbf{H}$ à la matrice de combinaison comme en (3.30) ramène l'étude de stabilité à la stabilité générale des fonctions de tâche [Chaumette 06]. En effet dans ce cas, on définit la fonction de tâche par $\mathbf{e}_c = \mathbf{C}\mathbf{e}$ et sa variation s'exprime par :

$$\begin{aligned}\dot{\mathbf{e}}_c &= \mathbf{C}\dot{\mathbf{e}} + \dot{\mathbf{C}}\mathbf{e} = (\mathbf{C}\mathbf{J}_s + \mathbf{O})\dot{\mathbf{q}} \\ &= -\lambda(\mathbf{C}\mathbf{J}_s + \mathbf{O})\mathbf{e}_c\end{aligned} \tag{3.42}$$

où $\mathbf{O} \in \mathbb{R}^{n\times n} = 0$ quand $\mathbf{e}_c = 0$ [Malis 04b]. Avec la matrice de combinaison définie en (3.30), une telle loi est localement asymptoticallement stable (LAS) dans un voisinage de $\mathbf{e}_c = 0$ si [Isidori 95] :

$$\mathbf{C}\mathbf{J}_s = (\mathbf{H}\widehat{\mathbf{J}}_s)^+\mathbf{H}\mathbf{J}_s > 0 \tag{3.43}$$

Le système est donc LAS dès que $\mathbf{H}\mathbf{J}_s$ et $\mathbf{H}\widehat{\mathbf{J}}_s$ sont de rang plein et que le jacobien $\mathbf{J}_s$ est suffisamment bien estimé.

Dans le cas où $m > n$ (avec $m = \dim(\mathbf{s})$ et $n = \dim(\mathbf{q})$), des minima locaux peuvent exister. Ils correspondent aux configurations où $\mathbf{H}^2(\mathbf{s}-\mathbf{s}^*) \in \text{Ker } \widehat{\mathbf{J}}_s^\top$. En effet, dans ce cas on a :

$$\dot{\mathbf{q}} = -\lambda(\mathbf{H}\widehat{\mathbf{J}}_s)^+\mathbf{e} = -\lambda(\widehat{\mathbf{J}}_s^\top\mathbf{H}^2\widehat{\mathbf{J}}_s)^{-1}\widehat{\mathbf{J}}_s^\top\mathbf{H}^2(\mathbf{s}-\mathbf{s}^*) \tag{3.44}$$

$(\widehat{\mathbf{J}}_s^\top\mathbf{H}^2\widehat{\mathbf{J}}_s)$ étant de rang plein par hypothèse, $\dot{\mathbf{q}} = 0$ est équivalent à $\widehat{\mathbf{J}}_s^\top\mathbf{H}^2(\mathbf{s}-\mathbf{s}^*) = 0$.

Certaines approches permettent de déterminer analytiquement les minima locaux potentiels : ainsi dans [Schramm 05] les informations visuelles utilisées sont les coordonnées d'au moins 3 points dans le repère caméra. Le système est donc redondant, mais il est prouvé qu'aucun minimum local attractif n'existe. Toutefois dans le cas général il est très difficile de déterminer les configurations 3D qui correspondent à un minimum local attractif. Nous proposons en Section 3.2.4.1 une méthode pour détecter et s'échapper de ces configurations. Dans la section suivante, nous considérons le cas de poids nuls.

3.2.2.3 Comportement en cas de poids nuls

Nous rappelons ici une propriété des tâches à dimension variable lorsque certains poids sont nuls. Dans la pondération générique exposée en Section 3.2.1.1, certaines informations capteur ont un poids nul si elles correspondent à des contraintes se trouvant dans la zone de confiance (3.27). On décompose le vecteur des informations capteur en $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_0)$ où les informations $\mathbf{s}_0$ ont un poids nul $\mathbf{H}_0 = 0$. La commande (3.29) s'écrit alors :

$$\begin{aligned}\dot{\mathbf{q}} &= -\lambda\begin{bmatrix}\mathbf{H}_1\widehat{\mathbf{J}}_1 \\ \mathbf{H}_0\widehat{\mathbf{J}}_0\end{bmatrix}^+\begin{bmatrix}\mathbf{H}_1\mathbf{e}_1 \\ \mathbf{H}_0\mathbf{e}_0\end{bmatrix} = -\lambda\begin{bmatrix}\mathbf{H}_1\widehat{\mathbf{J}}_1 \\ \mathbf{0}\end{bmatrix}^+\begin{bmatrix}\mathbf{H}_1\mathbf{e}_1 \\ \mathbf{0}\end{bmatrix} \\ &= -\lambda\begin{bmatrix}(\mathbf{H}_1\widehat{\mathbf{J}}_1)^+ & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{H}_1\mathbf{e}_1 \\ \mathbf{0}\end{bmatrix} = -\lambda(\mathbf{H}_1\widehat{\mathbf{J}}_1)^+\mathbf{H}_1\mathbf{e}_1\end{aligned} \tag{3.45}$$

Comme il se doit, la commande est ainsi équivalente à une commande construite sur les seules informations activées.

Le schéma utilisé est notamment différent de l'approche de [Hafez 07b] où la loi de commande s'écrit :

$$\dot{\mathbf{q}} = -\lambda \widehat{\mathbf{J}}_s^{+} \mathbf{H} \mathbf{e}_s \tag{3.46}$$

$$= -\lambda \begin{bmatrix} \widehat{\mathbf{J}}_1 \\ \widehat{\mathbf{J}}_0 \end{bmatrix}^{+} \begin{bmatrix} \mathbf{H}_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \tag{3.47}$$

Dans ce cas les composantes annulées par pondération continuent d'être prises en compte dans la commande. La consigne correspondante est exactement la même que si la tâche $\mathbf{e}_0$ était réalisée ($\mathbf{e}_0 = 0$). Le phénomène induit est un comportement conservatif qui tend à ralentir le système quand des poids sont annulés. Ce comportement n'est pas présent avec la loi de commande que nous considérons.

Nous nous intéressons maintenant au problème inverse, c'est-à-dire la pondération minimale assurant le respect des contraintes.

3.2.2.4 Respect d'une contrainte par un poids minimal

Le problème usuel des tâches contraintes est de déterminer une loi de commande perturbant la tâche le moins possible tout en respectant les contraintes du système. Dans notre cas, l'objectif est de définir le poids minimum pour qu'une contrainte donnée soir respectée. Une fonction de Lyapunov associée à l'erreur pondérée $\mathbf{e_H} = \mathbf{He}$ est $V(\mathbf{e_H}) = \frac{1}{2}\mathbf{e_H}^{\top}\mathbf{e_H}$. Dans le domaine de stabilité locale, la dérivée de V vérifie :

$$\dot{V} = \frac{\partial V}{\partial \mathbf{e}} \dot{\mathbf{e}} = \sum_{i=1}^{m} h_i^2 e_i \dot{e}_i < 0 \tag{3.48}$$

Nous considérons l'information capteur d'indice $i \in [1, m]$ dont la valeur est dans l'intervalle autorisé $s_i \in \mathcal{I}^l$. L'erreur correspondante e_i décroît si et seulement si $e_i \dot{e}_i < 0$. D'après (3.48), cette condition revient à :

$$h_i^2 > -\frac{1}{e_i \dot{e}_i} \sum_{j \neq i} h_j^2 e_j \dot{e}_j \tag{3.49}$$

Ainsi il existe un poids minimal assurant la décroissance de l'erreur de l'information correspondante. Si la valeur désirée s_i^* est dans $\mathcal{I}^l$, c'est une condition suffisante pour que la contrainte soit respectée.

Comme récemment relevé dans [Keith 11], cette existence d'un poids suffisant est à mettre en correspondance avec les travaux de [Van Loan 85] dans le cadre des méthodes d'optimisation. En effet, il y est prouvé que faire tendre un poids vers l'infini ramène le problème à une minimisation sous contrainte. Dans notre cas, cette contrainte est l'égalité $\dot{s}_i = -\lambda(s_i - s_i^*)$: la vérifier assure naturellement que s_i converge vers s_i^*, et donc que la contrainte $s_i \in \mathcal{I}^l$ est respectée. Si s_i^* est strictement à l'intérieur de l'intervalle $\mathcal{I}^l$ alors une pondération finie suffit à respecter cette contrainte.

Quand plusieurs contraintes sont considérées simultanément, les poids minimaux sont difficiles à déterminer analytiquement. Dans (3.49) la valeur minimale de h_i est fonction

des autres poids $(h_j)_{j\neq i}$. C'est pourquoi la fonction de pondération (3.27) proposée pour les contraintes est autorisée à croître indéfiniement. En pratique, comme nous le verrons au chapitre 4 les poids observés sont relativement faibles même en cas de contraintes multiples. La concurrence entre les différentes contraintes apparaît toutefois : augmenter le nombre de contraintes a naturellement tendance à augmenter les valeurs des poids permettant de les respecter.

Dans la prochaine section, nous considérons l'évitement de butées articulaires. Pour cette contrainte particulière, il est possible de déterminer anlytiquement des poids optimaux.

3.2.3 Pondération adaptative pour l'évitement de butées

Comme nous venons de le voir, il est difficile de déterminer le jeu de poids minimaux permettant le respect des contraintes. Dans cette section nous détaillons cet aspect en décomposant le vecteur des informations capteur $\mathbf{s} = (\mathbf{s}_t, \mathbf{s}_c)$ où $\mathbf{s}_t$ regroupe les informations capteur correspondant à la tâche référencée du robot, et où $\mathbf{s}_c$ regroupe les informations capteur représentant uniquement des contraintes à respecter. Par analogie nous décomposons le jacobien $\mathbf{J}_s = (\mathbf{J}_t, \mathbf{J}_c)$ et la matrice de pondération $\mathbf{H} = \text{Diag}\,(\mathbf{H}_t, \mathbf{H}_c)$. L'idée générale est que la pondération arbitraire des composantes de $\mathbf{s}_t$ est tolérable, alors que la pondération de $\mathbf{s}_c$ gagne à être minimale pour perturber le moins possible la tâche du robot.

Pour davantage de lisibilité, on note les poids $\mathbf{H}_c = \text{Diag}\,(h_1, \ldots, h_n)$. On assigne la forme suivante à ces poids :

$$\forall i, h_i = \mu_i(s_{ci}) h_i^{\text{opt}} \tag{3.50}$$

où $\mu_i(s_{ci}) \in [0, 1]$ est une fonction d'activation assurant la continuité des poids et où $h_i^{\text{opt}} \geq 0$ est le poids minimal assurant que la contrainte est respectée. Une telle formulation est proposée dans [Marey 10c] pour l'évitement optimal des butées articulaires par GPM. Nous verrons qu'elle souffre des mêmes limitations que notre approche, toutefois une réponse sera apportée dans notre cas. Nous définissons maintenant ces deux termes.

3.2.3.1 Poids optimal

Avec la loi de commande (3.29), la variation des informations contraintes $\mathbf{s}_c$ s'écrit :

$$\dot{\mathbf{s}}_c = \mathbf{J}_c \dot{\mathbf{q}} = -\lambda \mathbf{J}_c (\mathbf{H}\mathbf{J}_s)^{+} \mathbf{H}\mathbf{e} \tag{3.51}$$

Dans le cas général, la forme analytique de $\mathbf{J}_c$ est complexe et il est difficile d'exploiter (3.51). Nous nous limitons par la suite aux butées articulaires : $\mathbf{s}_c = \mathbf{q}$. Le jacobien associé est naturellement $\mathbf{J}_c = \mathbb{I}_n$. L'objectif étant d'éviter les butées, une valeur usuelle pour la position désirée q_i^* de chaque articulation est le centre de l'intervalle $\mathcal{I}^l$. Cette valeur ne correspond évidemment pas à la pose désirée du robot, mais c'est la valeur prise en compte quand la contrainte est activée.

Nous montrons maintenant que (3.51) conduit à une expression simple pour la pondération

minimale. Pour $\mathbf{s}_c = \mathbf{q}$, (3.51) s'écrit :

$$\dot{\mathbf{q}} = -\lambda(\mathbf{H}\mathbf{J}_s)^{+}\mathbf{H}\mathbf{e} = -\lambda \begin{bmatrix} \mathbf{H}_t\mathbf{J}_t \\ \mathbf{H}_c \end{bmatrix}^{+} \mathbf{H}\mathbf{e} \tag{3.52}$$

$\mathbf{H}\mathbf{J}_s$ étant supposée de rang plein, sa pseudo-inverse s'écrit $(\mathbf{H}\mathbf{J}_s)^{+} = (\mathbf{J}_s^{\top}\mathbf{H}^2\mathbf{J}_s)^{-1}\mathbf{H}\mathbf{J}_s^{\top}$. En particulier, la partie inversée s'écrit :

$$\begin{aligned}(\mathbf{J}_s^{\top}\mathbf{H}^{\top}\mathbf{H}\mathbf{J}_s)^{-1} &= (\mathbf{J}_t^{\top}\mathbf{H}_t^2\mathbf{J}_t + \mathbf{H}_{\mathbf{q}}^2)^{-1} = \frac{\mathrm{com}(\mathbf{J}_t^{\top}\mathbf{H}_t^2\mathbf{J}_t + \mathbf{H}_{\mathbf{q}}^2)}{\det(\mathbf{J}_t^{\top}\mathbf{H}_t^2\mathbf{J}_t + \mathbf{H}_{\mathbf{q}}^2)} \\ &= \frac{1}{D(\mathbf{h}_c)}\mathbf{N}(\mathbf{h}_c)\end{aligned} \tag{3.53}$$

où $D(\mathbf{h}_c)$ est un polynôme des $(h_i)_i$, strictement positif en tant que déterminant d'une matrice définie positive. $\mathbf{N}(\mathbf{h}_c)$ est la comatrice de $\mathbf{J}_t^{\top}\mathbf{H}_t^2\mathbf{J}_t + \mathbf{H}_{\mathbf{q}}^2$, soit la matrice des cofacteurs. L'élément N_{jk} ne dépend donc ni de h_j, ni de h_k. $\mathbf{N}$ est symétrique et de rang plein. D'après le critère de Sylvester sur les matrices définies positives, les éléments diagonaux de $\mathbf{N}$ sont positifs. D'après (3.51) et (3.52), la variation des variables articulaires peut donc s'écrire :

$$\dot{\mathbf{q}} = -\frac{\lambda}{D}\mathbf{N}\left(\mathbf{J}_t^{\top}\mathbf{H}_t^2\mathbf{e}_t + \mathbf{H}_{\mathbf{q}}^2(\mathbf{q} - \mathbf{q}^*)\right) \tag{3.54}$$

Au facteur d'échelle positif $\frac{\lambda}{D}$ près, la variation de chaque variable articulaire s'écrit :

$$\forall i \in [1, n],\ \dot{q}_i \propto -\mathbf{N}_i\mathbf{c} - \mathbf{N}_i\mathbf{H}_{\mathbf{q}}^2(\mathbf{q} - \mathbf{q}^*) \tag{3.55}$$

où $\mathbf{c} = \mathbf{J}_t^{\top}\mathbf{H}_t^2\mathbf{e}_t$ est un vecteur indépendant de $\mathbf{H}_{\mathbf{q}}$ et où $\mathbf{N}_i$ est la i$^{\text{ème}}$ ligne de $\mathbf{N}$, donc indépendante de h_i. Le cas de plusieurs articulations activées simultanément sera traité en Section 3.2.3.3. Nous faisons maintenant l'hypothèse qu'une seule articulation q_i est activée. (3.55) devient alors linéaire en h_i^2 :

$$\dot{q}_i \propto -\mathbf{N}_i\mathbf{c} - N_{ii}h_i^2(q_i - q_i^*) \tag{3.56}$$

On peut alors distinguer deux configurations C1 et C2, représentées en Figure 3.3 :

– évitement actif (C1) : si $-\mathbf{N}_i\mathbf{c}$ et $(q_i - q_i^*)$ sont du même signe, le robot se rapproche de la butée la plus proche. Comme $N_{ii} > 0$ il existe un h_i^2 positif annulant la vitesse $\dot{q}_i$.

– évitement passif (C2) : si $-\mathbf{N}_i\mathbf{c}$ et $(q_i - q_i^*)$ sont de signes différents, le robot s'éloigne de lui-même de la limite la plus proche. L'évitement de butée est alors superflu, et cela correspond à l'absence de solution en h_i à l'équation $\dot{q}_i = 0$.

(a) Configuration C1 : l'articulation se rapproche de la butée la plus proche

(b) Configuration C2 : l'articulation s'éloigne de la butée la plus proche

FIG. 3.3 – Configurations d'évitement : évitements actif (a) et passif (b).

En notant q_i^{a} la position à laquelle on souhaite garantir l'évitement de butée ($\dot{q}_i = 0$), la forme linéaire (3.56) permet donc d'exprimer un poids minimal :

$$h_i^{\mathrm{opt}} = \begin{cases} \sqrt{\frac{-\mathbf{N}_i\mathbf{c}}{N_{ii}(q_i^{\mathrm{a}}-q_i^*)}} & \text{si } \frac{-\mathbf{N}_i\mathbf{c}}{(q_i^{\mathrm{a}}-q_i^*)} > 0 \quad \text{(C1)} \\ 0 & \text{sinon} \quad \text{(C2)} \end{cases} \tag{3.57}$$

Ce poids minimal peut être calculé à chaque itération et garantit que l'articulation i s'arrête au moins quand $q_i = q_i^{\mathrm{a}}$. On a alors en effet $\dot{q}_i = -\mathbf{N}_i\mathbf{c} - N_{ii}\left(\frac{-\mathbf{N}_i\mathbf{c}}{N_{ii}(q_i^{\mathrm{a}}-q_i^*)}\right)(q_i^{\mathrm{a}} - q_i^*) = 0$. Toutefois pour obtenir une commande continue il est nécessaire d'injecter progressivement le poids minimal : c'est le rôle de la fonction d'activation.

3.2.3.2 Fonction d'activation

Afin d'obtenir une commande continue, les poids doivent être continus lors de l'activation et de la désactivation des informations capteur. Pour la mesure s_c, un intervalle d'activation $\mathcal{I}^a = [s_c^{\mathrm{a}-}, s_c^{\mathrm{a}+}]$ est défini à l'image de l'intervalle sûr $\mathcal{I}^s$ dans (3.26) par :

$$\begin{cases} s_c^{\mathrm{a}-} &= s_c^- + \rho^{\mathrm{a}}(s_c^+ - s_c^-) \\ s_c^{\mathrm{a}+} &= s_c^+ - \rho^{\mathrm{a}}(s_c^+ - s_c^-) \end{cases} \tag{3.58}$$

où $\rho^{\mathrm{a}} < \rho$ est un paramètre de réglage. Une sigmoïde permet alors de définir la fonction d'activation classique [Remazeilles 06] :

$$\mu = \begin{cases} 0 & \text{si } s_c^{\mathrm{s}-} \leq s_c \leq s_c^{\mathrm{s}+} \\ \frac{1}{2}\left(1 + \tanh\left(\frac{s_c^{\mathrm{a}+}-s_c^{\mathrm{s}+}}{s_c^{\mathrm{a}+}-s_c} - \frac{s_c^{\mathrm{a}+}-s_c^{\mathrm{s}+}}{s_c-s_c^{\mathrm{s}+}}\right)\right) & \text{si } s_c^{\mathrm{s}+} < s_c < s_c^{\mathrm{a}+} \\ \frac{1}{2}\left(1 - \tanh\left(\frac{s_c^{\mathrm{a}-}-s_c^{\mathrm{s}-}}{s_c^{\mathrm{a}-}-s_c} - \frac{s_c^{\mathrm{a}-}-s_c^{\mathrm{s}-}}{s_c-s_c^{\mathrm{s}-}}\right)\right) & \text{si } s_c^{\mathrm{a}-} < s_c < s_c^{\mathrm{s}-} \\ 1 & \text{sinon} \end{cases} \tag{3.59}$$

La fonction ainsi définie est $\mathcal{C}^\infty$ sur $\mathcal{I}^l$ et permet d'injecter progressivement la valeur h^{opt} dans la loi de commande. La fonction d'activation est représentée en Figure 3.4.

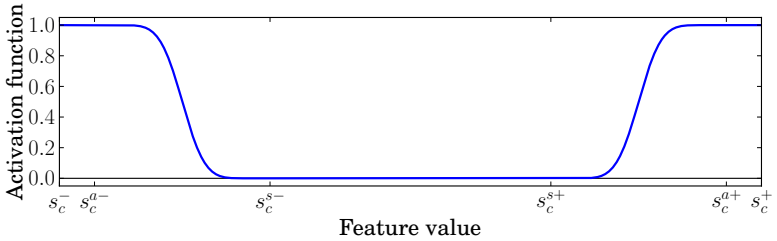


FIG. 3.4 – Fonction d'activation μ

Avec cette fonction d'activation, le poids optimal est injecté de façon progressive jusqu'à la limite de l'intervalle d'activation. Le poids minimal étant alors totalement activé, l'articulation s'arrête nécessairement. Nous résumons maintenant l'approche proposée dans cette section.

3.2.3.3 Stratégie de pondération pour les butées

La pondération optimale proposée paraît difficile à généraliser à d'autres types de contraintes. En effet le critère (3.56) permettant de déterminer un poids optimal n'est linéaire que dans le cas d'une seule butée articulaire. Dans les autres cas, le couplage entre les évitements des différentes butées complique le calcul analytique. On retrouve une limitation semblable dans l'approche de [Marey 10c] qui s'adresse à la même problématique de l'évitement optimal des butées articulaires. La méthode adaptative proposée ne garantit rien si plusieurs butées sont évitées en même temps. L'évitement de plusieurs butées est réalisé de façon itérative dans [Chaumette 01]. Si l'expression analytique d'un poids optimal est direct pour une seule butée (3.57), la résolution devrait également se faire de façon itérative - et numérique - pour résoudre le système (3.55) avec plusieurs contraintes activées.

Le calcul du poids minimum met en évidence le concept d'évitement passif. Un critère simple (3.57) permet d'identifier les configurations où l'activation n'est pas nécessaire. Dans les approches de pseudo-inverse pondérée [Chan 95, Xiang 10], le sens de variation de chaque articulation est mesuré et la pondération de l'évitement est réduite si l'articulation s'éloigne de la butée. Toutefois, aucune différence n'est faite entre les configurations d'évitement actif (dû à la pondération courante) et d'évitement passif, où la pondération ne joue aucun rôle. Le poids optimal (3.57) est donc séduisant dans le cas où une seule butée est considérée.

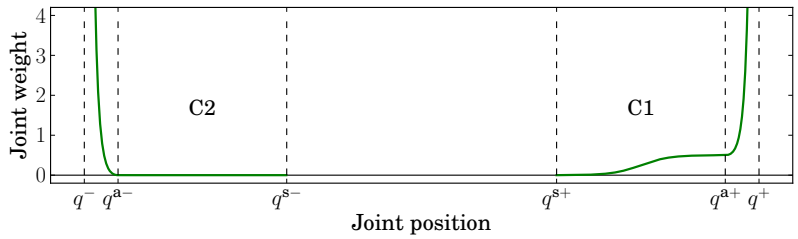


FIG. 3.5 – Stratégie de pondération pour les butées articulaires pour les évitements actif (C1) et passif (C2). La limite supérieure est en évitement actif ($h^{\text{opt}} = 0.5$), la limite inférieure en évitement passif. La pondération est minimale sur l'intervalle d'activation $[q^{a-}, q^{a+}]$. En dehors de cet intervalle, la pondération générique assure que les butées sont évitées.

Dans un cadre plus général, une stratégie peut être définie pour les butées articulaires en utilisant le poids adaptatif (3.50) sur son intervalle d'activation $\mathcal{I}^a - \mathcal{I}^s$ et le poids générique (3.27) si l'articulation sort malgré tout de $\mathcal{I}^a$. Cette stratégie est représentée en Figure 3.5 pour un évitement passif sur la limite inférieure, et actif sur la limite supérieure. Dans l'intervalle $\mathcal{I}^a$ la pondération est optimale si une seule butée est considérée : l'articulation s'arrête au pire des cas quand $q_i = q_i^{\text{a}}$. Elle peut également s'arrêter avant, si l'évolution de la tâche fait qu'un évitement passif est réalisé. Nous illustrons maintenant l'approche proposée par des simulations sur un cas simple.

3.2.3.4 Comportement d'un évitement de butées

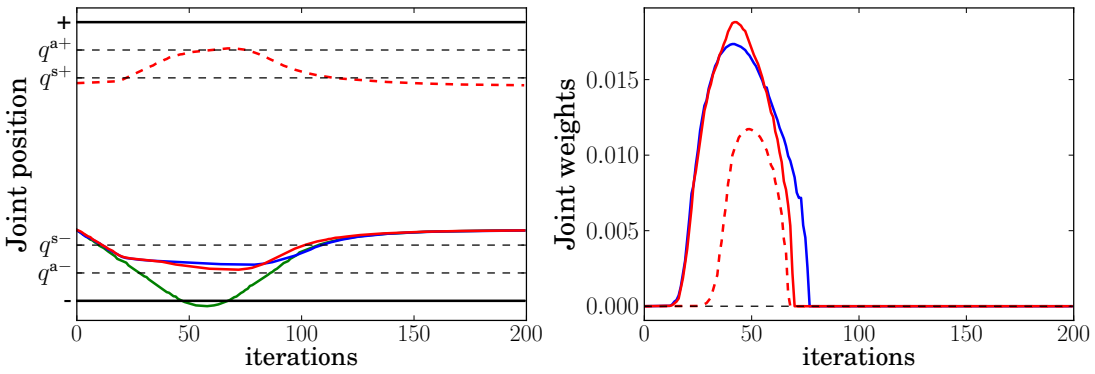
Dans cette section nous réalisons un évitement de butées en simulation sur un robot cartésien à 6 degrés de liberté. La tâche est un asservissement visuel 2D. Les informations $\mathbf{s}_{2d}$ utilisées sont les coordonnées cartésiennes dans l'image de 4 points formant un carré. La pose initiale diffère de la pose désirée par une rotation de 110° autour de l'axe optique. Il est notoire que dans cette configuration, la loi de commande génère un important mouvement en arrière [Chaumette 98]. Le robot peut donc potentiellement atteindre ses butées articulaires. La tâche est donc définie par :

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{2d} \\ \mathbf{q} \end{bmatrix} \quad \begin{array}{l} \cdots \text{ IBVS (dim. } 2\times 4) \\ \cdots \text{ Articulations (dim. 6)} \end{array} \tag{3.60}$$

La matrice de pondération est constante pour les informations visuelles et utilise l'approche proposée pour les variables articulaires :

$$\mathbf{H} = \text{Diag}\left(\mathbb{I}, \mathbf{H}_c\right) \tag{3.61}$$

Le comportement de l'évitement de butées est représenté en Figure 3.6. Les positions articulaires sont représentées en échelle normalisée sur la Figure 3.6a. Les limites des intervalles $\mathcal{I}^s$ et $\mathcal{I}^a$ sont en pointillés. Les limites physiques sont en traits gras.



(a) Positions articulaires

(b) Poids associés aux articulations

FIG. 3.6 – Positions articulaires (a) et poids associés (b) pour une tâche sans évitement (vert), évitant une butée (bleu) et évitant deux butées (rouge).

L'environnement de simulation permet de définir les butées de façon à ce que seules 1 ou 2 articulations approchent de la limite. La Figure 3.6b représente l'évolution des poids. Trois simulations sont comparées :

Pas d'évitement (courbe verte) La Figure 3.6a montre que l'articulation 1 franchit la limite q^-.

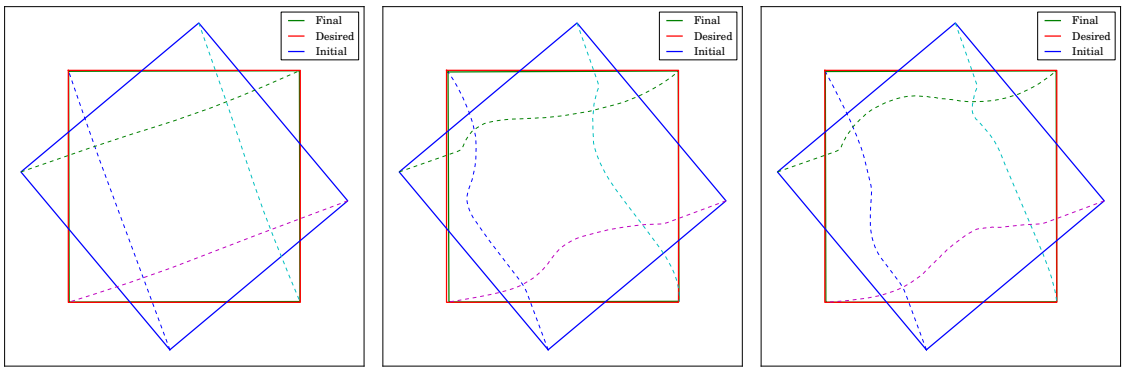
Une limite à éviter (courbes bleues) Cette fois l'articulation 1 n'atteint pas la limite et s'arrête avant q^{a-}. On remarque qu'après l'itération 75 l'articulation est toujours hors de

l'intervalle de confiance $\mathcal{I}^s$. La Figure 3.6b indique que le poids correspondant est pourtant nul : il s'agit là d'une situation d'évitement passif, où l'asservissement visuel s'éloigne seul de la butée articulaire.

Deux limites à éviter (courbes rouges) Dans cette simulation les butées sont définies telles que l'articulation 2 sorte de l'intervalle de confiance. Comme précédemment, l'articulation 1 s'arrête avant q^{a-}. En revanche l'articulation 2 franchit la valeur q^{a+} pendant plusieurs itérations. Ceci est dû au couplage entre les deux évitements empêchant une pondération optimale pendant quelques itérations. Les deux articulations restent toutefois éloignées de leurs butées réelles. Comme pour la $2^{\text{ème}}$ simulation, les deux poids sont nuls à partir de l'itération 75. Un évitement passif est ainsi réalisé pour les deux articulations.

Enfin, il est intéressant de remarquer que les poids liés aux contraintes sont très faibles (inférieurs à 0.02). Ces valeurs sont suffisantes pour éviter les butées. Des expériences d'évitement de butées sur robots réels sont exposées en Section 4.3.

La faible perturbation de la tâche visuelle peut être constatée en Figure 3.7. Le cas sans butées (Figure 3.7a) montre un asservissement visuel parfait avec des trajectoires rectilignes pour les 4 points. Ces droites sont très peu modifiées pour les cas avec évitement.



(a) Cas 0, sans butées (b) Cas 1, butée sur l'axe Z (c) Cas 2, butées sur Z et X

FIG. 3.7 – Trajectoires dans l'image pour les différents cas. L'évitement perturbe très peu la tâche visuelle.

Les différentes trajectoires sont représentées sur la Figure 3.8. Le cas 0 (bleu) montre une trajectoire parfaitement rectiligne : la caméra s'éloigne de l'objet en pivotant, puis revient vers la pose désirée. Le cas 1 (vert) possède une butée le long de l'axe optique. Contrairement à d'autres approches qui suivraient toujours une trajectoire rectiligne, mais plus courte, l'évitement proposé utilise au mieux l'espace articulaire disponible pour assurer la tâche visuelle. Pour le cas 2 (rouge), une butée est rajoutée dans la direction favorisée par le cas 1. La trajectoire initiale est semblable, mais la caméra ne peut pas dessiner la même boucle et exploite les autres axes pour réaliser la tâche visuelle. Des trajectoires 3D très différentes correspondent ainsi à des trajectoires très proches dans l'image.

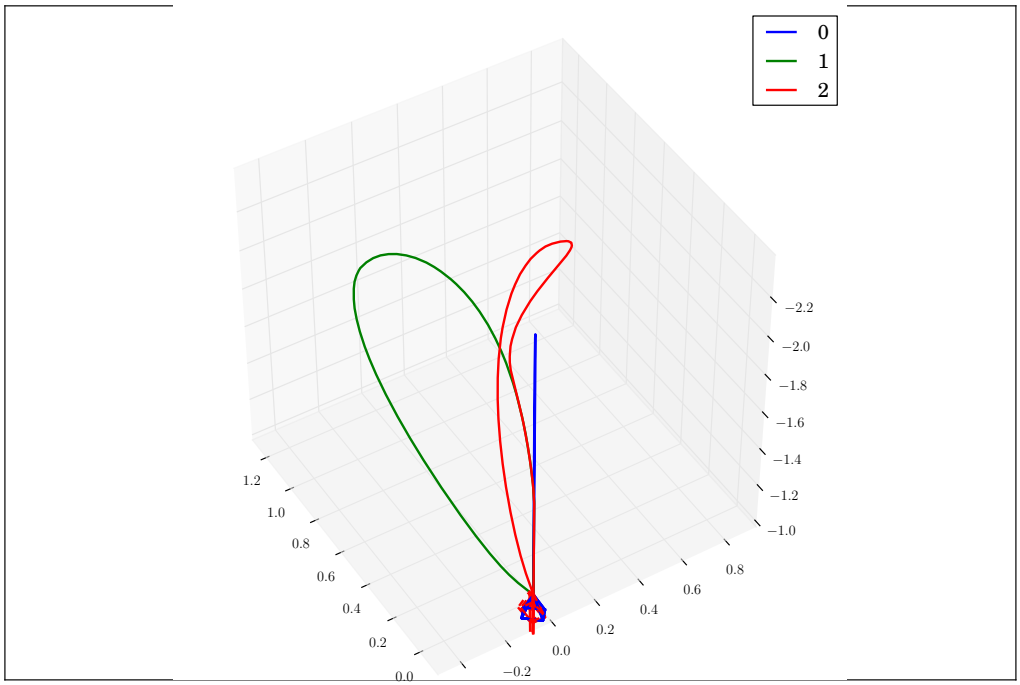


FIG. 3.8 – Trajectoires des différents cas. Trajectoire rectiligne sans butées (bleu), très large boucle avec une butée (vert) et plus limitée avec deux butées (rouge).

3.2.3.5 Comparaison avec les autres approches

L'inventaire des principales propriétés que nous venons d'établir permet de dresser un comparatif avec les autres types de commandes référencées capteurs.

Par rapport aux techniques hiérarchiques, l'intérêt de notre approche est que la pondération générique permet à la fois de respecter les contraintes, et de réaliser au mieux la tâche liée au déplacement du robot. Avec le GPM, les contraintes peuvent être violées si elles sont considérées comme des tâches secondaires. À l'inverse, la tâche référencée capteurs n'est que partiellement réalisée si les contraintes sont considérées comme la tâche principale.

En cas de nombreux capteurs, la commande commutative peut se révéler difficile à concevoir, notamment pour les critères permettant de passer à l'un ou l'autre des capteurs. Ce problème n'est pas présent avec notre approche, les poids continus déplacent simplement le compromis en fonction des différentes contraintes.

Par rapport aux autres commandes hybrides, l'intérêt principal est la pondération générique appliquée sur la technique du jacobien étendu. Cela permet une conception très simple des lois de commande, à l'image des commandes hybrides basées sur une moyenne pondérée des vitesses. Toutefois les schémas obtenus par le jacobien étendu sont davantage découplés que par moyenne des vitesses. Comme nous l'avons vu, le formalisme de la commande LQ est intégré à notre approche à travers la partie h^t de la pondération générique. Les techniques associées à la commande LQ peuvent être utilisées pour régler les poids en cas de critères particuliers sur le comportement du système. Au vu de la formulation des schémas basés sur la pseudo-inverse pondérée (WLN) il nous semble

que considérer les contraintes comme des informations capteur plutôt que comme des articulations virtuelles offre davantage de souplesse. Enfin, le fait d'autoriser les poids à croître indéfiniement donne une formulation absente des schémas existants de tâches à dimension variable. La concurrence entre les différentes contraintes se fait naturellement via la comparaison des poids. Un aspect intéressant - et rassurant - à ce sujet est que de multiples contraintes peuvent être respectées avec des poids restant relativement faibles, comme nous le verrons au chapitre 4 lors des applications.

Dans les travaux liés à la commande hiérarchique [Mansard 09b], les informations capteur correspondants aux contraintes sont considérées comme la tâche principale. Chaque composante est pondérée à 0 dans la zone de confiance et à 1 au voisinage de la contrainte. La tâche de navigation est considérée comme secondaire, mais peut toutefois perturber les composantes de la tâche principale dont les poids sont inférieurs à 1 (partiellement activés).

Comparaison en simulation En l'absence de comparaison formelle avec [Mansard 09b], une piste de réflexion est donnée à travers une simulation numérique sur un cas simple.

Nous considérons une tâche $\mathbf{e} = (\mathbf{e}_{12}, \mathbf{e}_3)$ de dimension $m = 3$ pour un système à $n = 2$ degrés de liberté. On suppose que seule la $3^{\text{ème}}$ composante $\mathbf{e}_3$ est liée à une contrainte. Le jacobien est noté $\mathbf{J} = (\mathbf{J}_{12}, \mathbf{J}_3) \in \mathbb{R}^{3\times 2}$. Nous comparons trois lois de commande sur cette configuration : l'approche hiérarchique pondérée proposée dans [Mansard 09b], notre proposition et son expression avec la pseudo-inverse continue [Mansard 09c].

– L'approche hiérarchique pondérée définit la loi de commande suivante :

$$\dot{\mathbf{q}}^p = h_3 \mathbf{J}_3^+ \dot{\mathbf{e}}_3^* + (\mathbb{I}_2 - h_3 \mathbf{J}_3^+ \mathbf{J}_3)\mathbf{J}_{12}^+ (\dot{\mathbf{e}}_{12}^* - h_3 \mathbf{J}_{12}\mathbf{J}_3^+ \dot{\mathbf{e}}_3^*) \tag{3.62}$$

où $h_3 \in [0, 1]$ est le poids associé à la contrainte. On remarque que (3.62) revient à :
  – $\dot{\mathbf{q}}^p = \mathbf{J}_{12}^+ \dot{\mathbf{e}}_{12}^*$ si $h_3 = 0$, soit la loi minimisant $\mathbf{e}_{12}$ en l'absence de contrainte
  – $\dot{\mathbf{q}}^p = \mathbf{J}_3^+ \dot{\mathbf{e}}_3^* + (\mathbb{I}_2 - \mathbf{J}_3^+ \mathbf{J}_3)\mathbf{J}_{12}^+ (\dot{\mathbf{e}}_{12}^* - \mathbf{J}_{12}\mathbf{J}_3^+ \dot{\mathbf{e}}_3^*)$ si $h_3 = 1$, soit la formulation de l'approche hiérarchique par opérateur de projection si la contrainte est totalement activée. Cette formulation prend en compte la partie de $\dot{\mathbf{e}}_{12}^*$ réalisée par $\dot{\mathbf{e}}_3^*$.

– Notre approche utilise quant à elle la loi de commande suivante :

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_{12} \\ \tilde{h}_3 \mathbf{J}_3 \end{bmatrix}^+ \begin{bmatrix} \dot{\mathbf{e}}_{12}^* \\ \tilde{h}_3 \dot{\mathbf{e}}_3^* \end{bmatrix} = (\mathbf{HJ})^+ \mathbf{He} \tag{3.63}$$

avec $\mathbf{H} = \text{Diag}\,(1, 1, \tilde{h}_3)$, où $\tilde{h}_3 \in [0, +\infty[$ est le poids associé à la contrainte. Les deux pondérations peuvent être mises en correspondance avec la relation $\tilde{h}_3 = \frac{h_3}{1-h_3}$.

– Avec la pseudo-inverse continue, notre approche s'écrit :

$$\dot{\mathbf{q}}^c = \mathbf{J}^{\oplus \bar{\mathbf{H}}} \dot{\mathbf{e}}^* \tag{3.64}$$

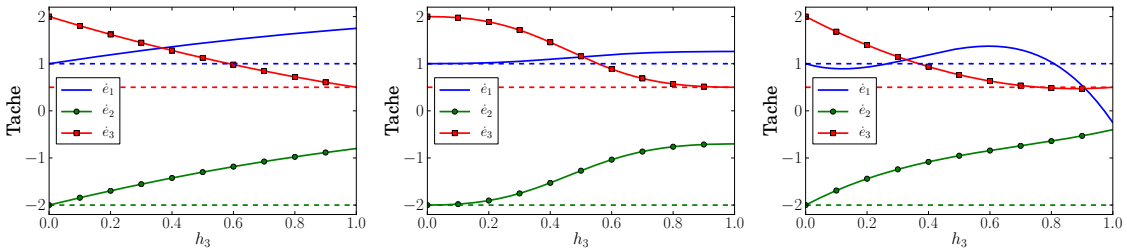
où l'opérateur continu $\mathbf{J}^{\oplus \bar{\mathbf{H}}}$ est défini dans [Mansard 09c] pour être équivalent à la seule tâche $\mathbf{e}_{12}$ quand $h_3 = 0$ et à la seule tâche $\mathbf{e}_3$ quand $h_3 = 1$. Entre les deux, une interpolation est réalisée. $\bar{\mathbf{H}}$ est une expression normalisée de $\mathbf{H}$:

$$\bar{\mathbf{H}} = (1 - h_3)\mathbf{H} = \text{Diag}\,(1 - h_3, 1 - h_3, h_3) \tag{3.65}$$

La comparaison est effectuée pour les valeurs suivantes :

$$\mathbf{J} = \begin{bmatrix} 1 & 3 \\ -2 & 3 \\ 2 & -4 \end{bmatrix} \quad \text{et} \quad \dot{\mathbf{e}}^* = \begin{bmatrix} 1 \\ -2 \\ 0.5 \end{bmatrix} \tag{3.66}$$

La Figure 3.9 représente l'évolution de la tâche globale $\dot{\mathbf{e}} = \mathbf{J}\dot{\mathbf{q}}$ en fonction de l'injection de la contrainte $\dot{\mathbf{e}}_3^*$ selon les trois stratégies.



(a) Hiérarchie pondérée $\mathbf{J}\dot{\mathbf{q}}^p$ (b) Jacobien étendu $\mathbf{J}\dot{\mathbf{q}}$ (c) Inverse continue $\mathbf{J}\dot{\mathbf{q}}^c$

FIG. 3.9 – Évolution de la réalisation de la tâche globale lors de l'injection de la contrainte. Les valeurs désirées sont représentées en pointillés. Comportement par hiérarchie pondérée (3.62) (a), avec notre approche (3.63) (b) et pseudo-inverse continue (3.64) (c). La tâche $\dot{\mathbf{e}}_{12}^*$ (bleu+vert) est toujours parfaitement réalisée quand $h_3 = 0$. La tâche $\dot{\mathbf{e}}_3^*$ (rouge) l'est quand $h_3 = 1$.

On constate que les valeurs extrêmes correspondent bien aux contraintes :
- en $h_3 = 0$, $\dot{\mathbf{e}}_{12}^*$ est réalisée parfaitement : la contrainte est ignorée.
- en $h_3 = 1$, c'est $\dot{\mathbf{e}}_3^*$ qui est réalisée parfaitement : la contrainte est respectée.

La façon d'injecter la contrainte n'est cependant pas la même suivant les trois approches :
- La pseudo-inverse pondérée (3.64) (Figure 3.9c) réalise une interpolation entre les valeurs extrêmes sans la tâche secondaire. La consigne atteinte pour $h_3 = 1$ correspond donc à la seule tâche $\dot{\mathbf{e}}_3^*$, ce qui n'est pas satisfaisant dans ce cas.
- La hiérarchie pondérée (3.62) (Figure 3.9a) réalise une interpolation entre les valeurs extrêmes de h_3, en prenant en compte la projection de la tâche secondaire sur l'espace disponible.
- Notre approche (3.63) (Figure 3.9b) suit globalement la même direction que la hiérarchie pondérée. Elle met toutefois en avant la non-linéarité du couplage entre les deux tâches. Par ailleurs, elle conduit à une meilleure réalisation de $\dot{\mathbf{e}}_1^*$ (bleu), et à une réalisation semblable pour $\dot{\mathbf{e}}_2^*$ (vert).

Enfin, il est remarquable que notre loi de commande ne converge pas vers la consigne de la seule tâche $\mathbf{e}_3$ même quand $\tilde{h}_3$ converge vers $+\infty$.

Ce paragraphe illustre brièvement les similitudes entre notre méthode et le schéma considéré comme le plus proche. Cette étude n'a pas été développée davantage faute de temps, toutefois il serait intéressant d'analyser plus en détail ces différences de comportement. Une différence fondamentale reste que la pondération entre 0 et 1 laisse l'idée d'une contrainte totalement activée, alors qu'une pondération entre 0 et $+\infty$ ne

conçoit que des poids relatifs. Nous avons toutefois vu en Section 3.2.2.4 que le respect d'une contrainte était précisément une question de poids relatif. Nous apportons maintenant des stratégies additionnelles pour des problèmes particuliers liés à notre approche.

3.2.4 Problèmes soulevés

Dans cette section nous apportons des réponses aux principaux problèmes soulevés par notre approche. Comme annoncé en Section 3.2.2.2, une stratégie est d'abord proposée pour détecter et s'échapper des minima locaux. Nous nous intéressons ensuite aux situations où la position désirée est en dehors de la région de confiance des contraintes. Enfin, nous apportons une solution simple pour prévenir les oscillations dans le cas de contraintes multiples.

3.2.4.1 Sortie d'un minimum local

Un problème commun à de nombreux schémas de commande référencée capteurs est l'existence de minima locaux potentiels. En particulier, dès que $n < m$ le noyau de $\widehat{\mathbf{J}}_s^\top$ n'est pas nul. Toutefois, les éléments de Ker $\widehat{\mathbf{J}}_s^\top$ ne correspondent pas forcément à des configurations physiquement réalisables. Dans cette section nous proposons une méthode pour détecter qu'un minimum local a été atteint. Une modification de la loi de commande permet alors d'essayer d'en sortir. Une technique alternative serait d'avoir recours à de la planification.

Un minimum local correspond nécessairement à une situation où la vitesse de l'effecteur est nulle bien que des éléments de l'erreur pondérée $\mathbf{He}$ ne sont pas nuls. En optimisation numérique, le recuit simulé [Brooks 95, Fang 06] est une méthode populaire pour éviter les minima locaux. Avec cette technique, un modèle probabiliste permet au critère minimisé d'augmenter entre deux itérations.

Données : $\alpha = 1$, $\alpha^+ > 1$, $\alpha^- < 1$
tant que *la convergence n'est pas atteinte* **faire**
  calculer la matrice de pondération $\mathbf{H} = \text{Diag}(\mathbf{H}_1, \mathbf{H}_2)$;
  calculer $\mathbf{v}_e$ avec (3.29);
  si $\|\mathbf{v}_e\| < v_0$ *et* $\mathbf{H}(\mathbf{s} - \mathbf{s}^*) > e_0$ **alors**
    $\alpha \leftarrow \alpha^+ \alpha$;
  sinon
    $\alpha \leftarrow 1 + \alpha^-(\alpha - 1)$;
  fin
  appliquer la loi de commande avec la matrice de pondération $\text{Diag}(\alpha\mathbf{H}_1, \mathbf{H}_2)$;
fin

Algorithme 1: Sortie d'un minimum local.

Bien que notre approche repose sur le compromis, sortir d'un minimum local nécessite de définir une hiérarchie entre deux sous-ensembles des informations capteur. Sans perte de généralité, on note $\mathbf{e}_1$ la tâche favorisée et $\mathbf{e}_2$ la tâche non favorisée. $\mathbf{e}_1$ et $\mathbf{e}_2$ peuvent être des combinaisons de tâches : par exemple on peut favoriser le respect de l'ensemble des contraintes le temps de sortir d'un minimum local. La tâche $\mathbf{e}_1$ regroupera alors les contraintes, et $\mathbf{e}_2$ regroupera les informations capteur utilisées pour la navigation. Le recuit simulé est appliqué sur la matrice de pondération $\mathbf{H}$. La stratégie proposée est décrite en

Algorithme 1. Elle consiste à augmenter petit à petit les poids de la tâche favorisée quand un minimum local est détecté. Comme précédemment exposé, on suppose qu'un minimum local est atteint quand la vitesse de l'effecteur devient très faible alors que $\mathbf{H}(\mathbf{s}-\mathbf{s}^*) \neq 0$. À chaque itération, les poids $\mathbf{H}_1$ sont multipliés par un facteur $\alpha > 1$, favorisant ainsi la tâche $\mathbf{e}_1$. Le compromis ainsi modifié peut potentiellement sortir du minimum local. Cette sortie correspond à une configuration telle que :

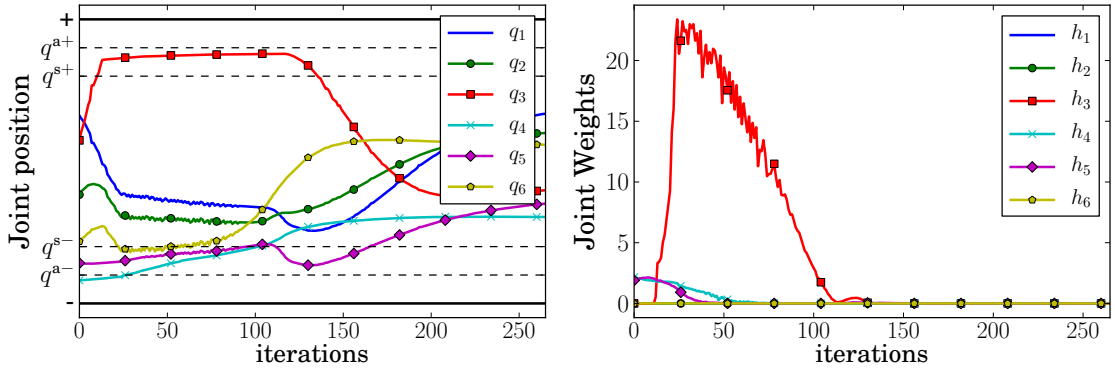
$$\begin{cases} \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix}^2 (\mathbf{s}-\mathbf{s}^*) \in \text{Ker } \widehat{\mathbf{J}}_s{}^+ \\ \begin{bmatrix} \alpha\mathbf{H}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix}^2 (\mathbf{s}-\mathbf{s}^*) \notin \text{Ker } \widehat{\mathbf{J}}_s{}^+ \end{cases} \tag{3.67}$$

Dans ce cas, le mouvement est nul si $\alpha = 1$ mais non nul si $\alpha > 1$.

α est toujours égal à 1 tant qu'un minimum local n'a pas été détecté. Une fois la sortie effectuée, α est lentement réajusté à 1. Le réglage de α^+ et α^- peut se révéler délicat. Toutefois, pour que la sortie d'un minimum local soit nette et définitive, il est nécessaire que α augmente plus rapidement qu'il ne reconverge vers 1 : $\alpha^+\alpha^- > 1$. Nous avons choisi de fixer ces paramètres à $\alpha^+ = 1.05$ et $\alpha^- = 0.99$.

La question des minima locaux a été envisagée de par l'utilisation d'un grand nombre d'informations capteur. Toutefois, en pratique de tels minima surviennent très rarement.

Le comportement typique d'une sortie de minimum local est représenté en Figure 3.10. Dans cette simulation, un asservissement visuel 3D est réalisé avec un évitement des butées articulaires. Les positions relatives des butées sont représentées sur la Figure 3.10a. Le minimum local apparaît vers l'itération 10. Dans cette configuration, l'asservissement visuel et l'évitement de butée s'équilibrent presque exactement. La pondération de l'évitement de butée est alors artificiellement augmentée. Comme on le voit en Figure 3.10b, le poids de l'articulation 3 augmente ainsi jusqu'à 20 pour permettre de sortir du minimum local.



(a) Positions articulaires

(b) Poids des articulations

FIG. 3.10 – Position et pondération des articulations lors d'une sortie de minimum local. On observe de faibles oscillations dans le calcul du poids h_3, entraînant un mouvement légèrement oscillant.

Sur les bras manipulateurs utilisés lors des expériences, les minima locaux surviennent très rarement et cette stratégie ne s'est pas révélée nécessaire malgré le grand nombre

d'informations capteur utilisées. Elle se révélerait sans doute indispensable pour des robots mobiles en environnement encombré. Dans la prochaine section nous présentons le cas beaucoup plus courant de la position désirée en dehors de la zone de confiance.

3.2.4.2 Atteindre une position où les contraintes sont activées

L'approche proposée minimise la norme de l'erreur pondérée $\mathbf{He}$. Dans le cas général d'une tâche de navigation soumise à certaines contraintes, l'erreur se décompose en $\mathbf{e} = (\mathbf{e}_t, \mathbf{e}_c)$ où $\mathbf{e}_t$ correspond aux informations capteur commandant la navigation, et où $\mathbf{e}_c$ exprime les contraintes potentielles. La position désirée correspond alors généralement au minimum de la seule erreur de navigation $\mathbf{H}_t\mathbf{e}_t$. Si certaines contraintes sont activées à la position désirée, alors le minimum de l'erreur de navigation ne correspond généralement pas au minimum global :

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} \left(\mathbf{e}_t^\top \mathbf{H}_t^2 \mathbf{e}_t\right) \neq \arg\min_{\mathbf{q}} \left(\mathbf{e}^\top \mathbf{H}^2 \mathbf{e}\right) \tag{3.68}$$

où $\mathbf{e}^\top\mathbf{H}^2\mathbf{e} = \mathbf{e}_t^\top\mathbf{H}_t^2\mathbf{e}_t + \mathbf{e}_c^\top\mathbf{H}_c^2\mathbf{e}_c$. Parfois le compromis entre la position désirée et le respect des contraintes peut être un effet recherché. C'est typiquement le cas si la position désirée est proche d'une butée articulaire, ou même si elle a été définie au delà des limites du robot. Dans ce cas, le respect des contraintes empêche d'atteindre la consigne donnée dans l'espace des mesures capteur.

Une condition suffisante pour que l'inégalité (3.68) soit surmontée est $\mathbf{e}_c^\top\mathbf{H}_c^2\mathbf{e}_c = 0$. Au niveau des informations capteur $\mathbf{s}_c$ liées aux contraintes, cette condition se vérifie dans deux configurations :

$$\forall s_{ci} \in \mathbf{s}_c, \quad \begin{cases} h_i = 0 & \text{(Z1)} \\ \qquad \text{ou} & \\ s_{ci} = s_{ci}^* & \text{(Z2)} \end{cases} \tag{3.69}$$

Une position à proximité des contraintes peut donc être atteinte si chaque contrainte est :

- soit associée à un poids nul (Z1)
- soit associée à une erreur nulle (Z2) à la position désirée

Dans le cas général, la condition Z2 n'est pas réalisée : la valeur désirée $\mathbf{s}_c^*$ pour les informations liées aux contraintes est en effet souvent arbitraire et définie uniquement pour ramener la valeur vers son intervalle de confiance. Nous verrons toutefois que pour la contrainte de visibilité, $\mathbf{s}_c^*$ est définie exactement à la position désirée.

Par définition, la condition Z1 n'est pas réalisée pour toutes les contraintes, car cela signifierait que la position désirée est dans la zone de confiance ($\mathbf{H}_c = 0$). Si on souhaite atteindre systématiquement la position désirée dans l'espace des mesures, alors une solution est de désactiver progressivement les contraintes à mesure que la tâche de navigation converge. On définit le paramètre de progression $\xi(\|\mathbf{e}_t\|)$ par :

$$\xi(\|\mathbf{e}_t\|) = \begin{cases} 0 & \text{si } \|\mathbf{e}_t\| \leq e_0 \\ 1 & \text{si } \|\mathbf{e}_t\| \geq e_1 \\ \frac{1}{2}\left(1 + \tanh\left(\frac{e_1 - e_0}{e_1 - \|\mathbf{e}_t\|} - \frac{e_1 - e_0}{\|\mathbf{e}_t\| - e_0}\right)\right) & \text{sinon} \end{cases} \tag{3.70}$$

où e_0 et e_1 sont des paramètres tels que les contraintes sont normalement activées quand la tâche de navigation est loin de son objectif ($\|\mathbf{e}_t\| > e_1$) et totalement désactivées quand la

convergence est proche ($\|\mathbf{e}_t\| < e_0$). Un exemple de paramètre de progression est représenté en Figure 3.11.

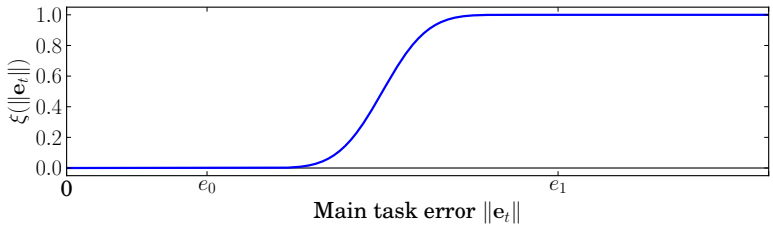


FIG. 3.11 – Paramètre de progression. Au fur et à mesure que l'erreur de la tâche $\mathbf{e}_t$ diminue, les poids des contraintes sont diminués pour permettre la convergence.

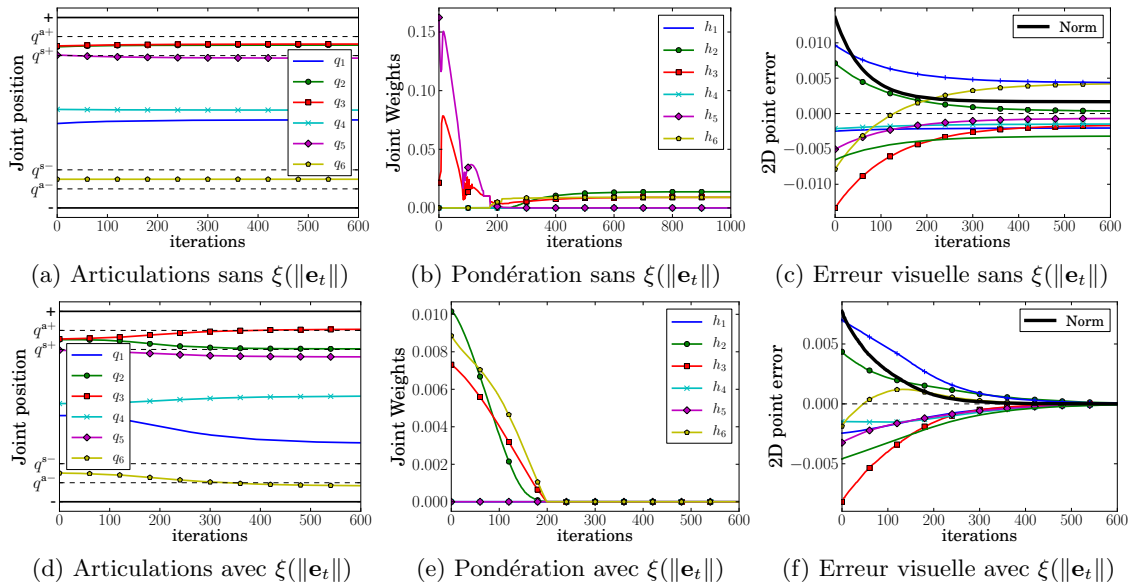
Dans ce cas, la matrice de pondération s'écrit $\mathbf{H} = \text{Diag}\left(\mathbf{H}_t, \xi(\|\mathbf{e}_t\|)\mathbf{H}_c\right)$ et est égale à $\mathbf{H}^* = \text{Diag}\left(\mathbf{H}_t, 0\right)$ dans un voisinage de la position désirée. Dans la suite nous mettrons en œuvre cette méthode pour toutes les situations d'évitement de butées. En effet il est difficile de savoir à l'avance si la position correspondant aux valeurs désirées dans l'espace des mesures est dans la zone de confiance pour toutes les articulations. Pour atteindre la position désirée, on désactive donc progressivement les contraintes de butées articulaires. Nous proposons de régler les seuils e_0 et e_1 au début de la tâche en fonction de l'erreur initiale :

$$\begin{cases} e_0 = \frac{1}{10}\|\mathbf{e}_t(t=0)\| \\ e_1 = \frac{1}{5}\|\mathbf{e}_t(t=0)\| \end{cases} \tag{3.71}$$

Dans le cas de l'évitement de butées, un exemple de minimum global non désiré est représenté en Figure 3.12.

Certaines articulations sont hors de l'intervalle de confiance pour la position désirée. Si les poids ne sont pas annulés artificiellement, le schéma converge vers un minimum global qui ne correspond pas à une erreur visuelle nulle. En annulant les poids, le minimum global est bien la pose désirée. En pratique, la pose atteinte sans annuler les poids reste cependant très proche de la pose désirée.

Le choix d'ignorer les contraintes au voisinage de la pose désirée ne s'impose pas toujours. En effet, il est légitime de vouloir respecter les contraintes même si cela empêche de rejoindre exactement la pose désirée. En commande référencée capteur, il est également possible de définir une pose désirée en dehors des limites de l'espace atteignable. Dans ce cas il est évidemment préférable de rejoindre une position intermédiaire et de continuer à prendre en compte les contraintes jusqu'à convergence.



(a) Articulations sans $\xi(\|\mathbf{e}_t\|)$ (b) Pondération sans $\xi(\|\mathbf{e}_t\|)$ (c) Erreur visuelle sans $\xi(\|\mathbf{e}_t\|)$

(d) Articulations avec $\xi(\|\mathbf{e}_t\|)$ (e) Pondération avec $\xi(\|\mathbf{e}_t\|)$ (f) Erreur visuelle avec $\xi(\|\mathbf{e}_t\|)$

FIG. 3.12 – Minimim global non désiré (en haut). Les poids ne sont pas nuls en fin de convergence (b), ce qui empêche l'erreur visuelle d'être annulée (c). En utilisant la méthode proposée (en bas), les poids sont annulés (e) et la tâche visuelle converge (f).

3.2.4.3 Respect de contraintes multiples et oscillations

La pondération générique proposée pour le respect des contraintes permet à certains poids d'atteindre des valeurs arbitrairement élevées. Comme nous l'avons vu, cette stratégie est justifiée par l'absence de critère analytique donnant un jeu de poids optimaux. Toutefois, quand plusieurs contraintes sont prises en compte, des oscillations peuvent être causées par d'importantes variations de poids. De plus, par effet de discrétisation certaines contraintes peuvent potentiellement être violées. En effet, en cas de mouvements rapides la valeur d'une information capteur peut potentiellement sortir de l'intervalle $\mathcal{I}^l$ avant que le poids associé ne soit suffisant pour l'y maintenir.

Ces problématiques étant liées à la vitesse du système, une solution simple est de la diminuer dans le voisinage des contraintes. Cette approche est similaire à celle proposée dans [Gans 07]. Dans ces travaux, une commande commutative (3.14) est définie entre des asservissements visuels 2D et 3D. Afin de limiter les oscillations, le système est ralenti dans le voisinage des conditions permettant de passer d'un schéma à l'autre.

Pour notre part, nous définissons le gain adaptatif suivant pour la loi de commande (3.29) :

$$\lambda(\|\mathbf{H}\|) = (\lambda_0 - \lambda_\infty) e^{-\frac{\lambda'_0}{\lambda_0 - \lambda_\infty}\|\mathbf{H}\|} + \lambda_\infty \tag{3.72}$$

où :

- $\lambda_0 = \lambda(0)$ est le gain en 0, soit pour une faible pondération[2].
- $\lambda_\infty = \lim_{\|\mathbf{H}\| \to \infty} \lambda(\|\mathbf{H}\|)$ est le gain à l'infini, correspondant à des poids importants.

[2]la norme de $\mathbf{H}$ n'est en pratique bien sûr jamais nulle

– λ'_0 est la pente de λ en $\|\mathbf{H}\| = 0$.

Ce gain permet de ralentir le système au voisinage des contraintes, et notamment des contraintes multiples qui entrent en concurrence. Dans le chapitre 4, toutes les expériences n'utilisent pas un tel gain adaptatif. Son utilité sera en revanche mise en avant dans le cas de contraintes multiples.

3.3 Conclusion

Dans ce chapitre, une approche a été proposée pour construire des lois de commande référencées multi-capteurs. La commande obtenue est de type hybride, et les contraintes sont prises en compte avec le formalisme des tâches à dimension variable. L'idée principale est que n'importe quelle contrainte est considérée comme une information capteur. C'est le poids associé à chaque information qui permet de régler le comportement général du système et les contraintes éventuelles à respecter. Dans ce cadre, une pondération générique a été proposée. Cela permet une grande souplesse d'utilisation, en bénéficiant des méthodes de réglage développées notamment pour la commande LQ. La pondération générique assure le respect des contraintes tout en gardant une approche reposant sur le compromis. Ce compromis dynamique peut être interprété comme une mise en concurrence des différentes contraintes. Le cas des butées articulaires a été détaillé, et une pondération optimale a été proposée. Il paraît difficile de déterminer analytiquement la pondération optimale d'une contrainte quelconque. La pondération optimale met en évidence le phénomène d'évitement passif. Des simulations ont montré comment ce phénomène se traduit lors de l'évitement d'une ou plusieurs butées.

Dans un cadre redondant ($n > m$), il serait intéressant d'utiliser la pseudo-inverse continue [Mansard 09c] avec l'approche de pondération proposée. Les résultats sur la pondération optimale faisant explicitement appel à la pseudo-inverse de Moore-Penrose (3.53), ils ne se transposent pas directement à la formulation de pseudo-inverse continue.

Les principales difficultés de la loi de commande proposée ont été identifiées. Une réponse a été proposée pour éviter les oscillations au voisinage des contraintes. Il s'agit de ralentir la vitesse du système au moyen d'un gain adaptatif reposant sur la norme de la matrice de pondération.

Deux cas de minima non désirés ont été différenciés. Les premiers sont les minima locaux classiques inhérents aux systèmes surdéterminés. Une approche a été proposée pour sortir d'un tel minimum local. Le second type de minimum non désiré correspond aux positions désirées se trouvant dans les zones de contrainte. Il est alors possible de désactiver les contraintes lorsque la tâche de navigation est proche de son point de convergence.

Soulignons que les deux cas de minima s'opposent : pour sortir du premier nous proposons d'augmenter les poids des contraintes, alors que la solution du deuxième est de diminuer les poids des contraintes pour rejoindre la position désirée. Ces stratégies dépendent donc fortement de la tâche envisagée. En pratique les minima locaux sont très rares, et nous avons fait le choix de favoriser la convergence vers la position désirée au détriment de la prise en compte des contraintes à son voisinage. Notons que ce problème est commun à l'opérateur de projection sur la norme utilisé dans [Marey 10c]. Dans ce cas, les contraintes exprimées sous la forme d'une tâche secondaire sont également ignorées

lorsque la tâche principale est proche de la convergence.

L'idée même d'une commande hybride assurant le respect des contraintes pose la question de la pertinence de la position désirée. La construction d'une loi de commande par jacobien étendu semble adaptée pour un robot mobile évoluant dans un environnement fortement contraint. Dans ce cas, il se pourrait que la position désirée ne soit pas atteignable, parce que mal définie ou parce que des contraintes imprévues en empêchent l'accès. Dans cette optique il serait intéressant de pouvoir identifier le type de minimum dans lequel le robot est stoppé : minimum local, ou bien position désirée dans la zone d'activation.

Le prochain chapitre regroupe les expériences mettant en œuvre la loi de commande présentée dans ce chapitre.

Chapitre 4

Applications de la fusion

Un formalisme a été proposé au chapitre précédent pour la commande référencée multi-capteurs. En tant que commande hybride, le comportement du système dépend de l'équilibre entre les capteurs.

Dans ce chapitre nous illustrons l'approche proposée à travers différentes applications. Dans un premier temps, le choix de la loi de commande est étudié à travers la coopération multi-caméras. Dans ce cadre, les comportements des différentes commandes sont comparés. La notion de contribution d'un capteur est également mise en avant. Pour cette première illustration, la pondération n'est pas utilisée.

La gestion des contraintes est ensuite prise en compte à travers la contrainte de visibilité en asservissement visuel. La pondération générique est utilisée à partir d'une formulation simple de la contrainte. L'approche proposée est également comparée à des schémas similaires.

Nous nous intéressons par la suite à l'évitement de butées en asservissement visuel. Dans ce cadre, la pondération optimale proposée au chapitre 3 est détaillée.

Les trois problématiques sont ensuite regroupées. L'objectif est alors de faire coopérer deux caméras en respectant les contraintes d'évitement de butées articulaires et de visibilité dans les deux images. Nous présentons les résultats obtenus sur ce système à contraintes multiples. Différentes combinaisons de contraintes sont présentées afin d'identifier le comportement induit.

4.1 Coopération multi-caméras

Dans cette section nous illustrons les aspects fondamentaux de la commande référencée multi-capteurs. La gestion des contraintes n'est pas considérée, et nous mettons l'accent sur l'équilibrage des contributions.

4.1.1 Fusionner deux caméras

L'utilisation conjointe de deux caméras est une méthode classique en vision. Une première approche est inspirée de la stéréovision, où les deux caméra sont rigidement liées et observent le même objet. L'application classique consiste alors à utiliser les deux images pour estimer la pose avec davantage de précision [Martin 02, Lippiello 07, Dionnet 09]. Cette méthode est généralisable aux caméras omnidirectionnelles [Boutteau 08, Eynard 10]. Dans ce cadre, on réalise une fusion au sens classique : les informations des deux images sont utilisées au sein d'un algorithme d'estimation de pose et la commande se fait dans l'espace opérationnel.

Le terme de coopération suppose généralement qu'on utilise une caméra fixe et une caméra mobile (eye-in-hand/eye-to-hand cooperation). C'est par exemple le cas dans [Horaud 06] où la caméra fixe permet de détecter la zone d'intérêt de l'espace que doit observer la caméra mobile. Dans [Dune 07], les dimensions d'un objet de forme inconnu sont estimées. La caméra mobile explore l'espace à partir d'une ligne visuelle définie dans le repère de la caméra fixe.

La commande référencée multi-capteurs s'applique à la fois à la stéréovision et à la coopération. Dans [Malis 01] deux caméras rigidement liées observent chacune un marqueur. L'approche proposée est d'établir une moyenne des vitesses (3.15). La coopération entre caméra embarquée et caméra déportée est effectuée de façon partitionnée dans [Flandin 00] pour un robot cartésien. La caméra déportée observe un marqueur fixé sur la partie linéaire du robot et contrôle les degrés de translation. La caméra embarquée observe un objet fixe et contrôle les degrés de rotation.

Nous proposons d'utiliser le schéma de coopération pour discuter du choix de la loi de commande. Nous présentons maintenant le système considéré.

4.1.2 Système expérimental

La coopération multi-caméras est réalisée sur un robot cartésien Afma6. La tâche est définie par deux asservissements visuels 2D. À cet effet, la caméra embarquée $\mathcal{F}_c$ observe une cible $\mathcal{F}_o$ fixe composée de 4 points. Les informations visuelles correspondantes, constituées des coordonnées cartésiennes des 4 points de l'image, sont notées $\mathbf{s}_{\text{int}}$. La caméra déportée $\mathcal{F}_{c2}$ observe quant à elle une cible $\mathcal{F}_{o2}$ liée à l'effecteur, également composée de 4 points. Les informations visuelles correspondantes sont notées $\mathbf{s}_{\text{ext}}$ et sont de nouveau constituées des coordonnées cartésiennes de ces 4 points dans l'image. Par analogie on parlera d'image embarquée et d'image déportée pour désigner les images acquises par chacune des caméras. Aucune pondération n'est utilisée ($\mathbf{H} = \mathbb{I}_{16}$). La disposition générale est représentée en Figure 4.1.

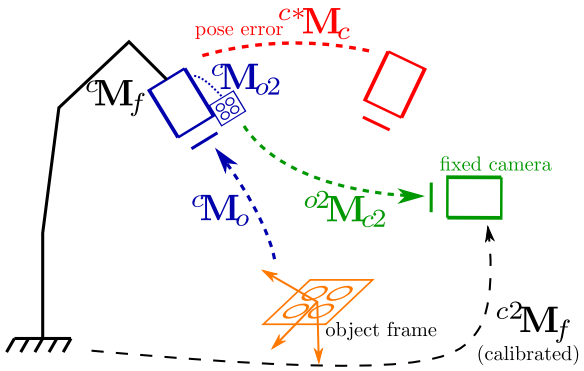


FIG. 4.1 – Coopération multi-caméras. Caméra embarquée (bleu), déportée (vert) et position désirée (rouge).

Les caméras sont supposées calibrées, et le repère de commande est assimilé au repère $\mathcal{F}_c$. Toutes les vitesses sont exprimées dans ce repère. La matrice de passage entre $\mathcal{F}_{c2}$ et $\mathcal{F}_c$ s'exprime par :

$$^{c2}\mathbf{M}_c = {}^{c2}\mathbf{M}_f \, {}^{f}\mathbf{M}_c \tag{4.1}$$

où $^{f}\mathbf{M}_c$ est connue en temps réel à travers le modèle géométrique du robot. La matrice de passage $^{c2}\mathbf{M}_f$ entre la caméra déportée et le repère fixe du robot est constante et grossièrement estimée hors-ligne. Pour ce faire, on se place dans une configuration où les deux caméras observent la cible $\mathcal{F}_o$. Par estimation de pose, on détermine $^{c}\mathbf{M}_o$ et $^{c2}\mathbf{M}_o$. La matrice constante $^{c2}\mathbf{M}_f$ est alors estimée par :

$$^{c2}\mathbf{M}_f = {}^{c2}\mathbf{M}_o \, {}^{c}\mathbf{M}_o^{-1} \, {}^{c}\mathbf{M}_f \tag{4.2}$$

La matrice de passage $^{c}\mathbf{M}_{o2}$ entre la caméra embarquée et la cible embarquée est constante mais n'est pas connue. Sa connaissance n'est pas nécessaire, car la tâche est définie par apprentissage. En début d'expérience, le robot est positionné dans la configuration désirée. Les deux images sont alors acquises et le vecteur des informations désirées $\mathbf{s}^*$ est construit. Nous bénéficions ainsi de la robustesse aux erreurs de calibration bien connue en asservissement visuel 2D [Espiau 92]. La configuration désirée est définie par $^{c}\mathbf{M}_o^* = (0, 0, 0.5, 0, 0, 0)$ (m,deg). La pose initiale est $^{c}\mathbf{M}_o = (0.02, 0.02, 0.52, 47, 12, 28)$ (m,deg). Les profondeurs sont de l'ordre de 0.5m pour l'image embarquée, et de 1.2m pour l'image déportée.

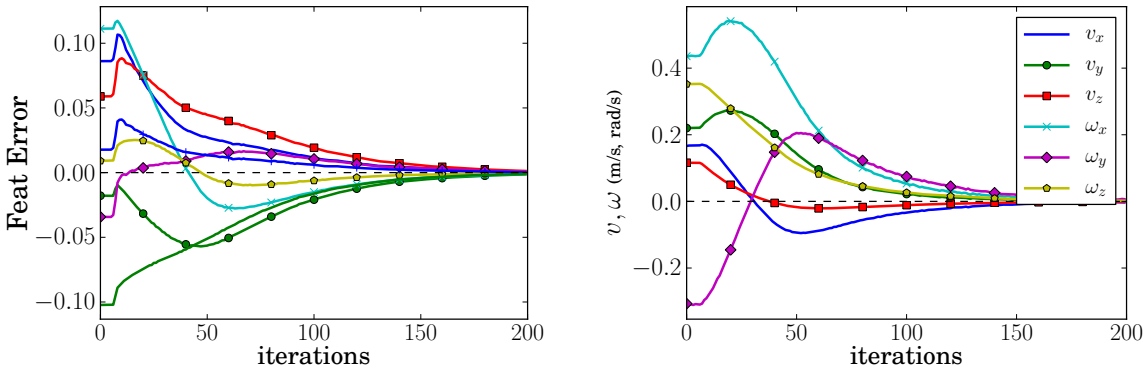
Nous comparons maintenant les comportements obtenus en n'utilisant qu'une seule caméra à la fois.

4.1.3 Comportement mono-caméra

Dans cette section nous présentons les résultats des asservissements visuels n'utilisant qu'une des deux caméras. Cela met en évidence la notion de contribution d'un capteur dans le cadre de la fusion.

4.1.3.1 Caméra embarquée

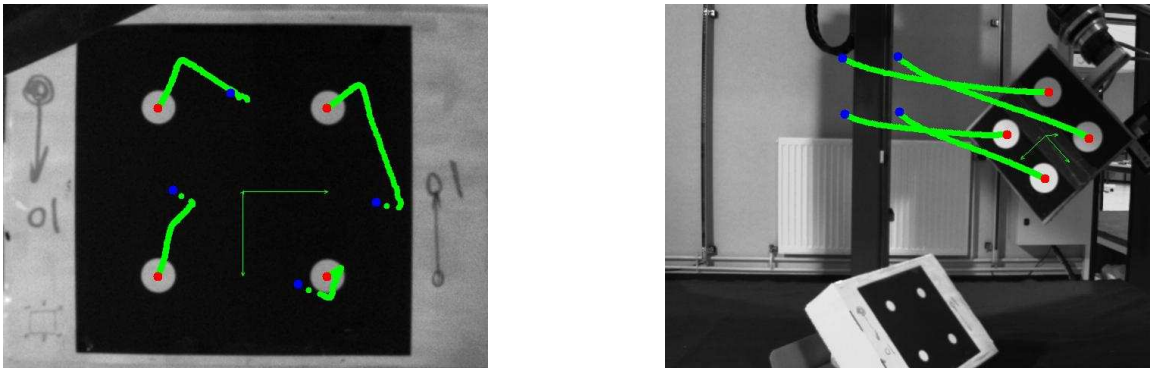
Nous utilisons tout d'abord la caméra embarquée : $\mathbf{s} = \mathbf{s}_{\text{int}}$. La matrice d'interaction utilisée est celle calculée à la position désirée $\mathbf{L}_s^*$. Les résultats sont représentés sur la Figure 4.2. Le comportement est classique d'un asservissement visuel utilisant la matrice d'interaction à la position désirée. La décroissance de l'erreur visuelle est tout d'abord approximative ($\mathbf{L}_s^* \neq \mathbf{L}_s$) puis devient plus régulière ($\mathbf{L}_s^* \approx \mathbf{L}_s$) (Figure 4.2a).



(a) Erreur visuelle (b) Vitesse de la caméra

FIG. 4.2 – Comportement avec caméra embarquée.

Le comportement dans les deux images est représenté en Figure 4.3. On retrouve les trajectoires classiques d'un asservissement visuel 2D avec $\mathbf{L}_s^*$ en Figure 4.3a. Les trajectoires des points ne sont pas rectilignes, la matrice d'interaction étant grossièrement estimée. Les trajectoires dans l'image déportée sont surprenantes : elles sont très rectilignes alors que ces mesures ne sont pas utilisées dans la commande.



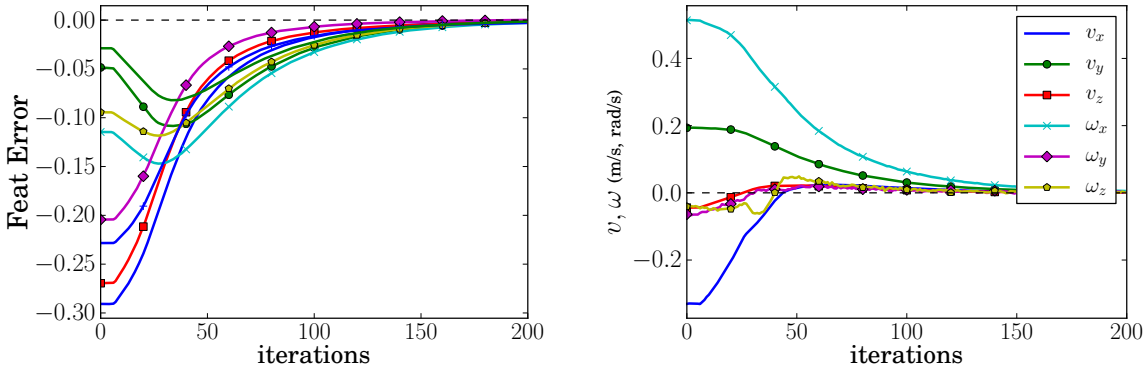
(a) Trajectoire dans l'image embarquée (b) Trajectoire dans l'image déportée

FIG. 4.3 – Comportement avec caméra embarquée. La trajectoire dans l'image embarquée (a) est classique d'un asservissement visuel 2D utilisant $\mathbf{L}_s^*$. La trajectoire dans l'image déportée (b) est presque rectiligne.

Nous comparons maintenant ce comportement avec celui induit par la caméra déportée.

4.1.3.2 Caméra déportée

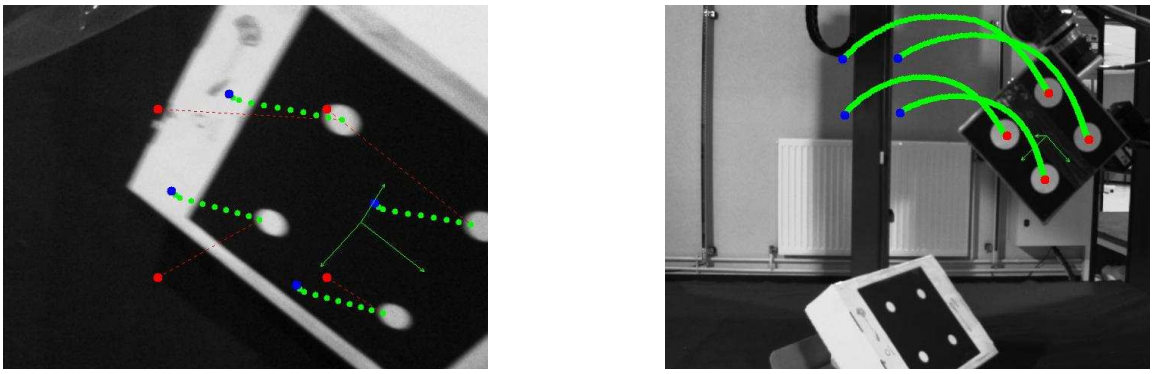
Nous utilisons maintenant la caméra déportée : $\mathbf{s} = \mathbf{s}_{\text{ext}}$. La matrice d'interaction utilisée est celle calculée à la position désirée $\mathbf{L}_s^*$. Le comportement général représenté sur la Figure 4.4 est toujours caractéristique d'un asservissement visuel utilisant une matrice d'interaction grossièrement estimée. Sur la Figure 4.4b on constate que certaines composantes cinématiques sont similaires au schéma avec caméra embarqué : v_y et ω_x notamment. On peut donc s'attendre à un comportement similaire dans le sens vertical du plan image. La composante v_x en revanche est opposée par rapport à la Figure 4.2b : le comportement horizontal est donc différent en utilisant la caméra déportée.



(a) Erreur visuelle (b) Vitesse de la caméra

FIG. 4.4 – Comportement avec caméra déportée.

Ces considérations sont confirmées par la Figure 4.5. Contrairement au cas embarqué, le comportement est loin d'être satisfaisant dans les deux images. En particulier, utiliser la caméra déportée conduit à une perte rapide de la cible dans l'image embarquée (Figure 4.5a).



(a) Trajectoire dans l'image embarquée (b) Trajectoire dans l'image déportée

FIG. 4.5 – Comportement avec caméra déportée. La cible sort rapidement de l'image embarquée (a). Les trajectoires observées dans l'image déportée (b) sont classiques d'un asservissement visuel 2D utilisant $\mathbf{L}_s^*$.

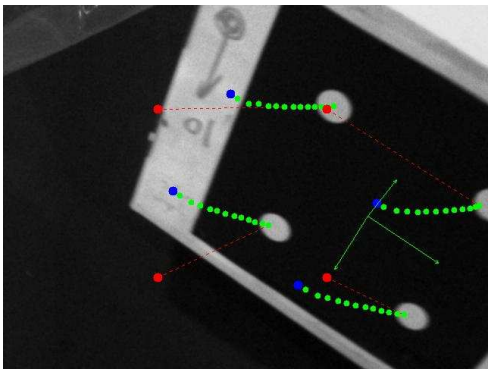
Les trajectoires de la Figure 4.5b sont encore une fois classiques d'un asservissement visuel 2D utilisant une matrice d'interaction approximative. Dans la prochaine section, nous réalisons une fusion dans l'espace opérationnel.

4.1.4 Fusion par moyenne des vitesses

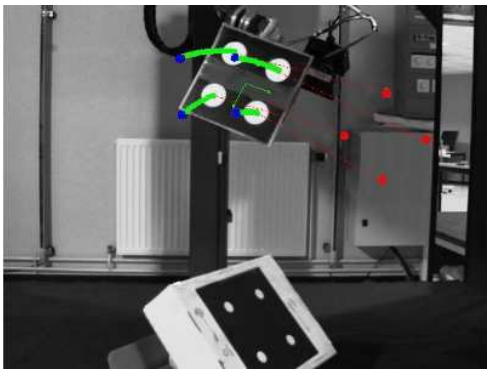
La commande multi-capteurs proposée dans [Malis 01] consiste à définir la consigne comme une moyenne pondérée des consignes des différents capteurs (3.15). Dans le cas de la coopération multi-caméras, la commande est donc :

$$\mathbf{v}_c = h_{\text{int}}\mathbf{v}_{\text{int}} + h_{\text{ext}}\mathbf{v}_{\text{ext}} = -\lambda \left(h_{\text{int}}\mathbf{L}_{\text{int}}^{*+}(\mathbf{s}_{\text{int}} - \mathbf{s}_{\text{int}}^{*}) + h_{\text{ext}}(\mathbf{L}_{\text{ext}}^{*}\,{}^{c2}\mathbf{W}_c)^{+}(\mathbf{s}_{\text{ext}} - \mathbf{s}_{\text{ext}}^{*}) \right) \quad (4.3)$$

Nous avons vu dans la section précédente que les comportements étaient très différents suivant la caméra utilisée. Sans pondération ($h_{\text{int}} = h_{\text{ext}} = 0.5$), on retrouve le comportement non satisfaisant lié à la caméra déportée. La Figure 4.6 représente les deux images : comme dans le cas purement déporté, la cible quitte rapidement l'image embarquée. Ces informations étant cette fois utilisées dans la commande, le robot ne peut pas continuer.



(a) Trajectoire dans l'image embarquée

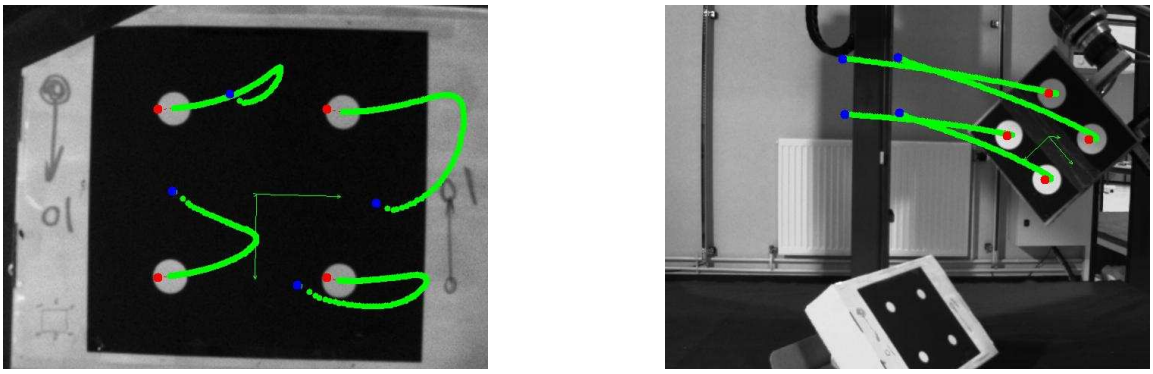


(b) Trajectoire dans l'image déportée

FIG. 4.6 – Comportement avec moyenne non pondérée des vitesses. La cible quitte rapidement l'image embarquée (a), se comportant de façon similaire au cas purement déporté.

Au delà de la contrainte de visibilité, qui n'est pas prise en compte explicitement ici, le comportement général de la fusion par moyenne non pondérée est proche de la commande par la seule caméra déportée. Cela vient probablement de la différence entre les profondeurs : autour de 0.5m pour l'image embarquée, et autour de 1.2m pour l'image déportée. Les mouvements dans l'image embarquée sont donc naturellement plus rapides, illustrant l'absence de découplage dans la convergence. Nous réduisons la contribution de la caméra déportée en définissant $h_{\text{int}} = 0.7$ et $h_{\text{ext}} = 0.3$. Le comportement dans l'image est représenté en Figure 4.7.

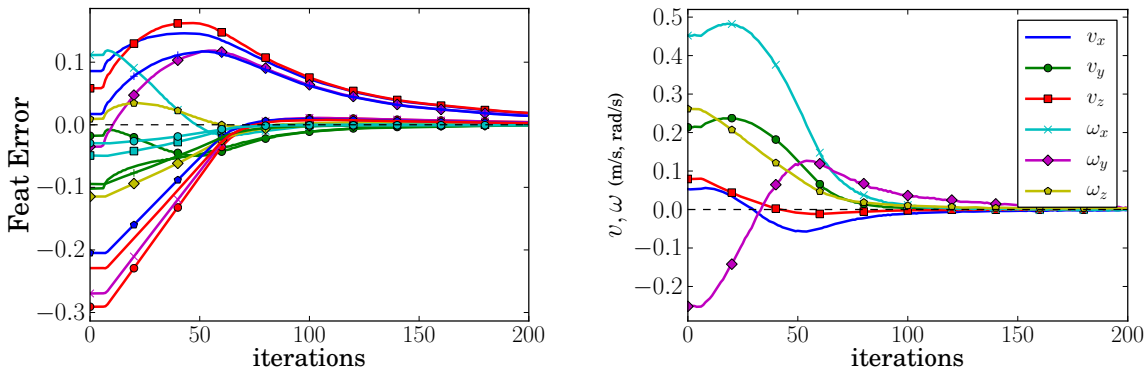
Le large mouvement horizontal est toujours visible sur la Figure 4.7a pour l'image embarquée. Toutefois le fait de favoriser cette caméra empêche la cible de sortir de l'image. Les trajectoires sont cependant beaucoup plus courbes dans l'image embarquée (Figure 4.7a) que dans l'image déportée (Figure 4.7b).



(a) Trajectoire dans l'image embarquée (b) Trajectoire dans l'image déportée

FIG. 4.7 – Comportement avec moyenne pondérée des vitesses. Les trajectoires sont très rectilignes dans l'image déportée (b) et plus courbes dans l'image embarquée (a).

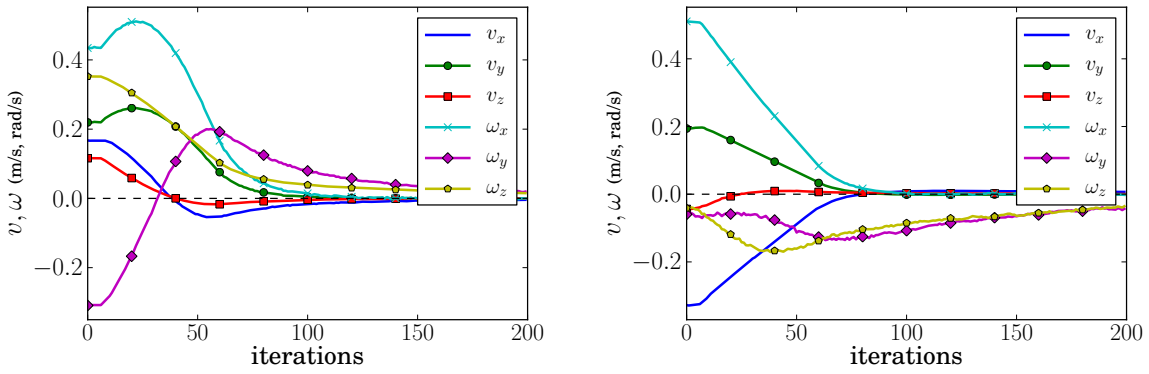
L'erreur globale est représentée sur la Figure 4.8a. Le schéma (4.3) ne réalisant pas un découplage optimal, on observe clairement deux groupes d'informations visuelles. Certaines convergent très rapidement vers la valeur désirées, alors que d'autres montrent un comportement transitoire moins satisfaisant. La vitesse de la caméra est représentée sur la Figure 4.8b. Les contributions des deux capteurs sont représentées sur la Figure 4.9. De part la moyenne pondérée, la vitesse finale est très proche de la vitesse émanant de la seule caméra embarquée (Figure 4.9a). De fait, le comportement général est proche du comportement mono-caméra en embarqué.



(a) Erreur visuelle (b) Vitesse $\mathbf{v}_c$ de la caméra

FIG. 4.8 – Comportement avec moyenne pondérée des vitesses. L'absence de découplage apparaît clairement dans l'évolution de l'erreur (a). Les vitesses sont similaires au cas seulement embarqué (b).

Cette observation est confirmée sur la Figure 4.10 qui représente les trajectoires 3D des trois commandes considérées jusqu'ici : caméra embarquée (bleue), caméra déportée (verte) et moyenne pondérée (rouge). La pondération est nécessaire pour obtenir un comportement acceptable pour le système. Toutefois il est difficile de déterminer une pondération a priori



(a) Contribution $\mathbf{v}_{\text{int}}$ de la caméra embarquée

(b) Contribution $\mathbf{v}_{\text{ext}}$ de la caméra déportée

FIG. 4.9 – Contributions des caméras dans la moyenne pondérée.

entre les deux caméras. Au delà de la contrainte de visibilité, les points ont des trajectoires très différentes dans les deux images, traduisant un découplage très approximatif.

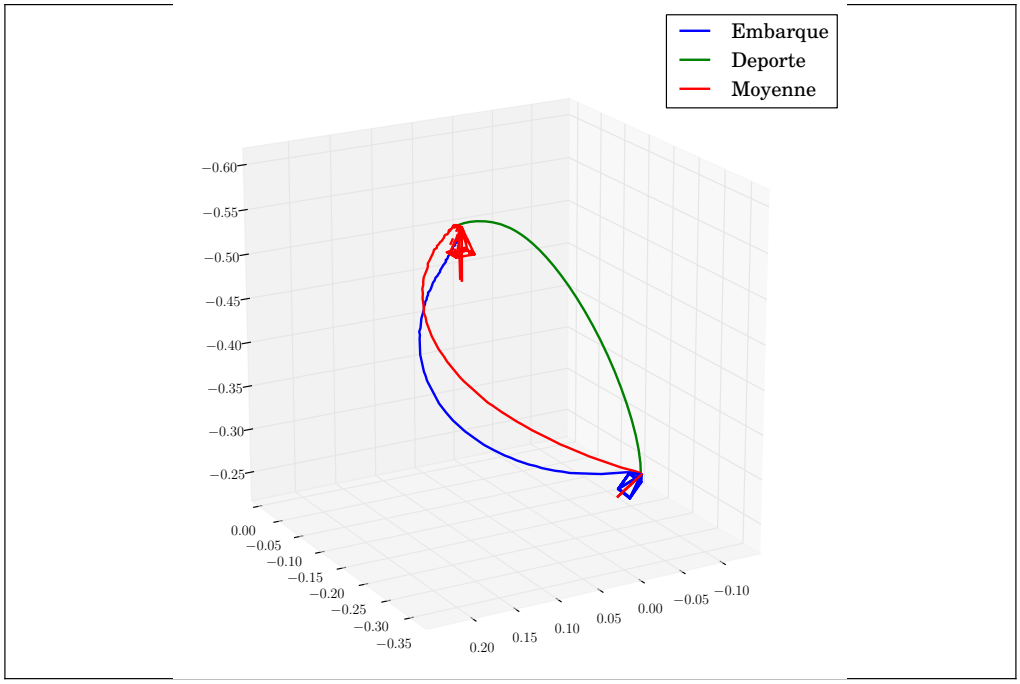


FIG. 4.10 – Moyenne pondérée des vitesses. La trajectoire résultante de la moyenne (rouge) est voisine de la trajectoire par caméra embarquée (bleue) et très différente de la trajectoire par caméra déportée (verte). La pose initiale est en bleue, la pose finale est en rouge.

Nous comparons maintenant ces observations avec notre approche.

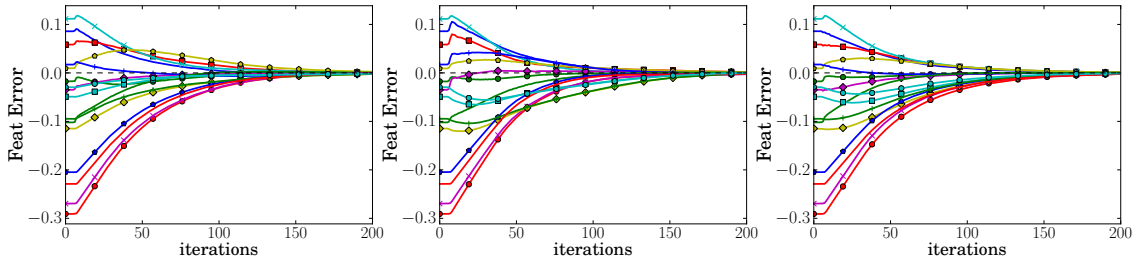
4.1.4.1 Jacobien étendu

La coopération multi-caméras est ici appliquée par l'approche du jacobien étendu (3.16). Le vecteur des informations capteur est ici défini par :

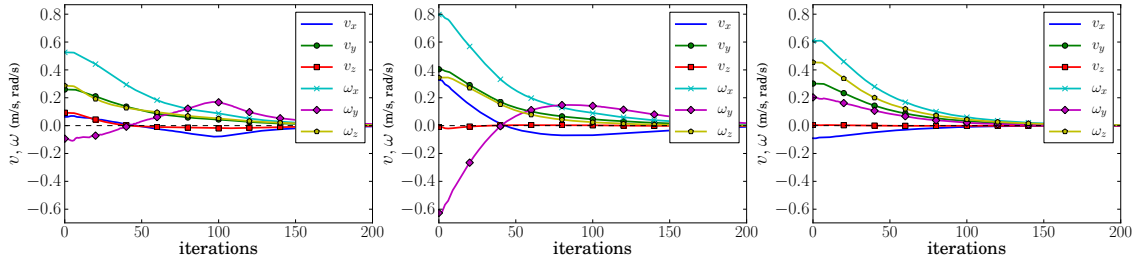
$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{\text{int}} \\ \mathbf{s}_{\text{ext}} \end{bmatrix} \quad \begin{matrix} \cdots \text{ Embarqué (dim. } 2\times 4) \\ \cdots \text{ Déporté (dim. } 2\times 4) \end{matrix} \tag{4.4}$$

Nous comparons les trois matrices de combinaison proposées en Section 3.2.1.2 :
- $\mathbf{C}_{\text{cur}}$ pour l'estimation courante des matrices d'interaction (3.31)
- $\mathbf{C}_*$ pour la valeur à la position désirée (3.32)
- $\mathbf{C}_{\frac{1}{2}}$ pour la moyenne des deux (3.41)

Les composantes de l'erreur globale $\mathbf{s}-\mathbf{s}^*$ sont représentées sur la Figure 4.11a pour les trois schémas. Le comportement est très similaire d'un schéma à l'autre. Le découplage apparaît clairement par rapport à l'approche par moyenne pondérée. Les vitesses correspondantes aux trois schémas sont représentées en Figure 4.11b. La décroissance la plus régulière est obtenue pour la matrice de combinaison moyenne $\mathbf{C}_{\frac{1}{2}}$. Les vitesses diffèrent principalement entre elles par leurs composantes v_x et ω_y, soit celles qui séparaient également les schémas mono-caméra (Figure 4.2b et Figure 4.4b).



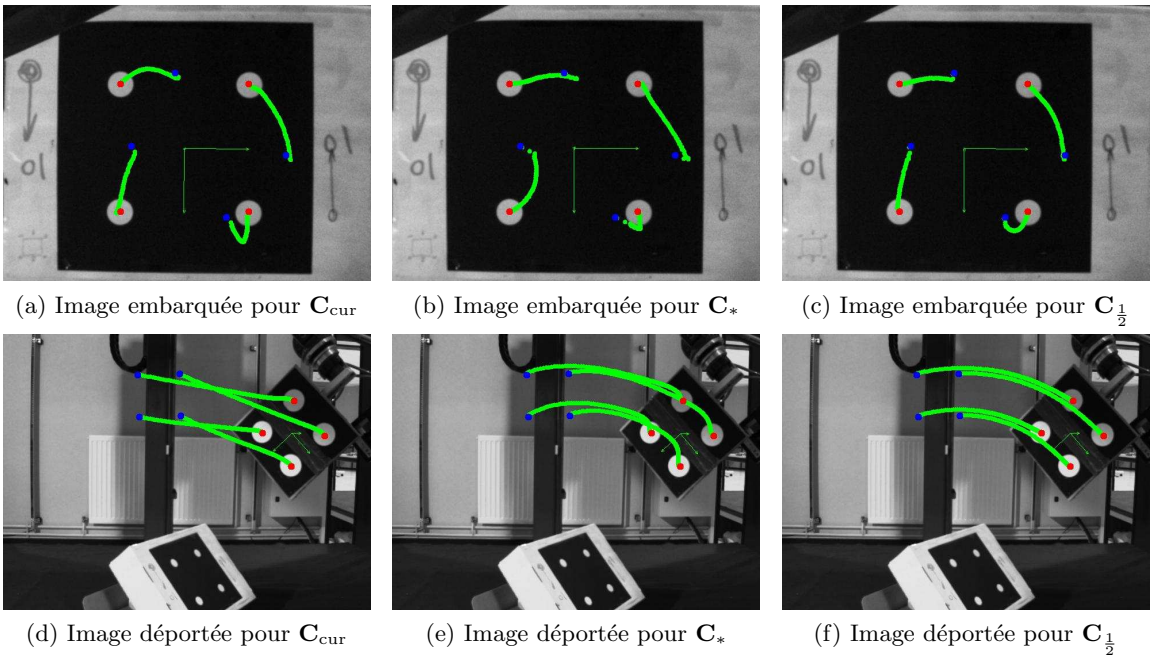
(a) Erreur globale pour l'estimation courante $\mathbf{C}_{\text{cur}}$ (gauche), désirée $\mathbf{C}_*$ (milieu) et moyenne $\mathbf{C}_{\frac{1}{2}}$ (droite). Les convergences sont très similaires.



(b) Vitesses pour l'estimation courante $\mathbf{C}_{\text{cur}}$ (gauche), désirée $\mathbf{C}_*$ (milieu) et moyenne $\mathbf{C}_{\frac{1}{2}}$ (droite).

FIG. 4.11 – Comportement (erreur et vitesse) selon les trois schémas : estimation courante $\mathbf{C}_{\text{cur}}$ (gauche), désirée $\mathbf{C}_*$ (milieu) et moyenne $\mathbf{C}_{\frac{1}{2}}$ (droite)

Les résultats restent très proches malgré la différence dans le choix de la matrice de combinaison. La Figure 4.12 confirme la similarité. Les trajectoires dans les images embarquée et déportée sont semblables quel que soit le choix de $\mathbf{C}$. Ceci fait preuve d'un bon découplage dans la loi de commande, sans avoir besoin d'utiliser la matrice de pondération.



(a) Image embarquée pour $\mathbf{C}_{\text{cur}}$ (b) Image embarquée pour $\mathbf{C}_*$ (c) Image embarquée pour $\mathbf{C}_{\frac{1}{2}}$

(d) Image déportée pour $\mathbf{C}_{\text{cur}}$ (e) Image déportée pour $\mathbf{C}_*$ (f) Image déportée pour $\mathbf{C}_{\frac{1}{2}}$

FIG. 4.12 – Trajectoires dans les images embarquée (haut) et déportée (bas), selon les trois schémas : estimation courante $\mathbf{C}_{\text{cur}}$ (gauche), désirée $\mathbf{C}_*$ (milieu) et moyenne $\mathbf{C}_{\frac{1}{2}}$ (droite)

Les trajectoires 3D correspondant aux lois de commande exposées sont représentées sur la Figure 4.13. À titre de comparaison, on y retrouve les trajectoires mono-caméra précédemment exposées. Les observations sont similaires à ce qu'on peut observer dans le cas du choix de la commande en asservissement visuel. Dans le cas multi-capteurs également, le choix entre $\mathbf{C}_{\text{cur}}$, $\mathbf{C}_*$ et $\mathbf{C}_{\frac{1}{2}}$ conduit à des trajectoires voisines dans l'espace 3D. On peut noter que dans ce cas la trajectoire obtenue avec $\mathbf{C}_{\frac{1}{2}}$ est proche de la ligne droite. Sans aucune pondération, les trajectoires obtenues avec $\mathbf{C}_{\text{cur}}$ et $\mathbf{C}_*$ sont plus proches du schéma purement embarqué que du schéma purement déporté.

Selon le découplage qu'il induit, le jacobien étendu semble donc plus à même de réaliser des tâches multi-capteurs que la moyenne pondérée des vitesses. Dans la prochaine section nous modifions les poses initiale et désirée, et mettons en relief les limitations du seul jacobien étendu.

4.1.5 Autre configuration

Dans cette section nous éloignons la caméra déportée de la zone de travail du robot. La profondeur correspondante est de l'ordre de 1.5m. La pose initiale est également plus éloignée de la position désirée : ${}^{\mathcal{C}}\mathbf{M}_o = (-0.12, 0.08, 0.8, -34, -63, -30)$(m, deg).

La Figure 4.14 représente les trajectoires dans les images. Dans le cas de $\mathbf{C}_*$, on constate que l'approximation consistant à utiliser $\mathbf{L}_s^* \approx \mathbf{L}_s$ est trop grossière. Le schéma $\mathbf{C}_*$ s'éloigne en effet rapidement de la pose désirée et la cible finit par sortir de l'image embarquée (Figure 4.14b). Les schémas utilisant l'estimée courante $\mathbf{C}_{\text{cur}}$ et moyenne $\mathbf{C}_{\frac{1}{2}}$ ont encore

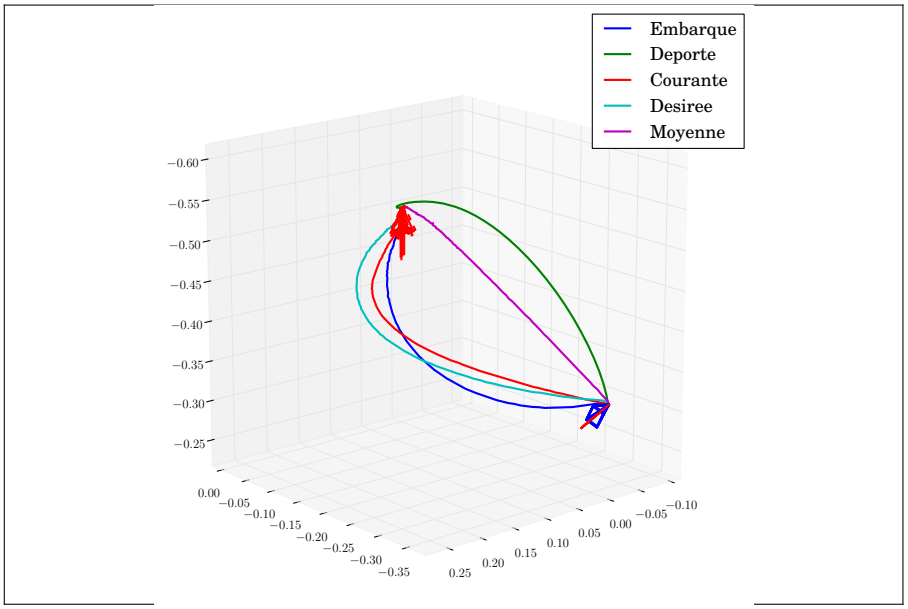
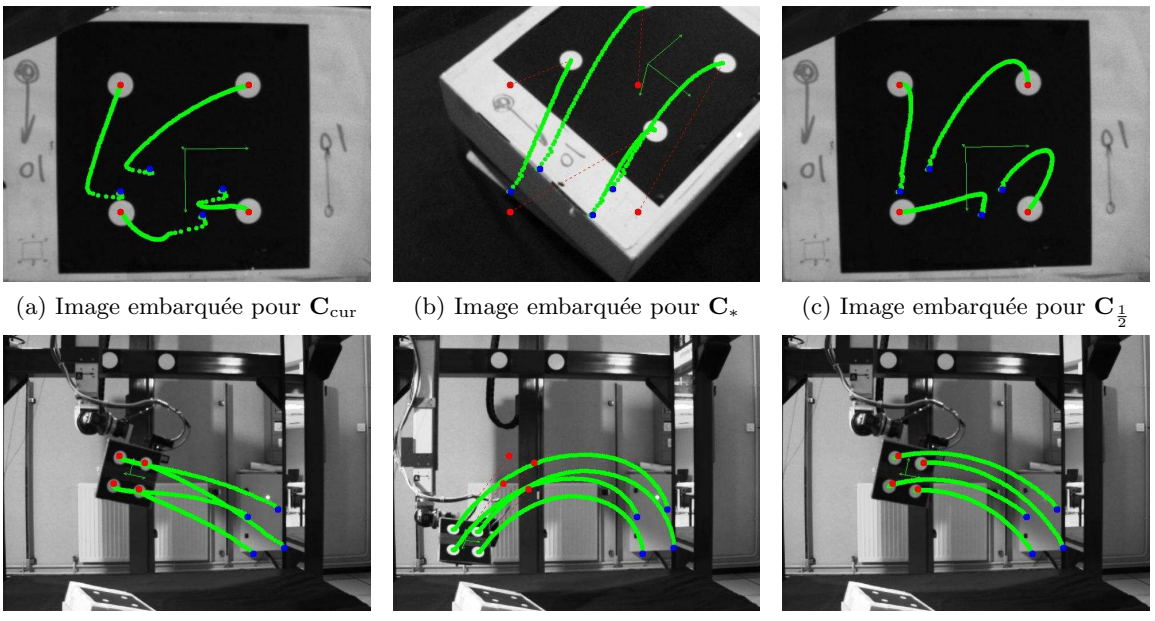


FIG. 4.13 – Choix de la matrice de combinaison. Trajectoires pour la caméra embarquée (bleu), la caméra déportée (vert), $\mathbf{C}_{\text{cur}}$ (rouge), $\mathbf{C}_*$ (cyan) et $\mathbf{C}_{\frac{1}{2}}$ (magenta)

une fois des comportements similaires.



(a) Image embarquée pour $\mathbf{C}_{\text{cur}}$ (b) Image embarquée pour $\mathbf{C}_*$ (c) Image embarquée pour $\mathbf{C}_{\frac{1}{2}}$

(d) Image déportée pour $\mathbf{C}_{\text{cur}}$ (e) Image déportée pour $\mathbf{C}_*$ (f) Image déportée pour $\mathbf{C}_{\frac{1}{2}}$

FIG. 4.14 – Trajectoires dans les images embarquée (haut) et déportée (bas), selon les trois schémas : estimation courante $\mathbf{C}_{\text{cur}}$ (gauche), désirée $\mathbf{C}_*$ (milieu) et moyenne $\mathbf{C}_{\frac{1}{2}}$ (droite)

Les trajectoires 3D (Figure 4.15) indiquent que c'est le schéma avec la matrice courante $\mathbf{C}_{\text{cur}}$ (bleu) qui est le plus près d'une trajectoire rectiligne. La trajectoire induite par la matrice moyenne $\mathbf{C}_{\frac{1}{2}}$ reste très similaire.

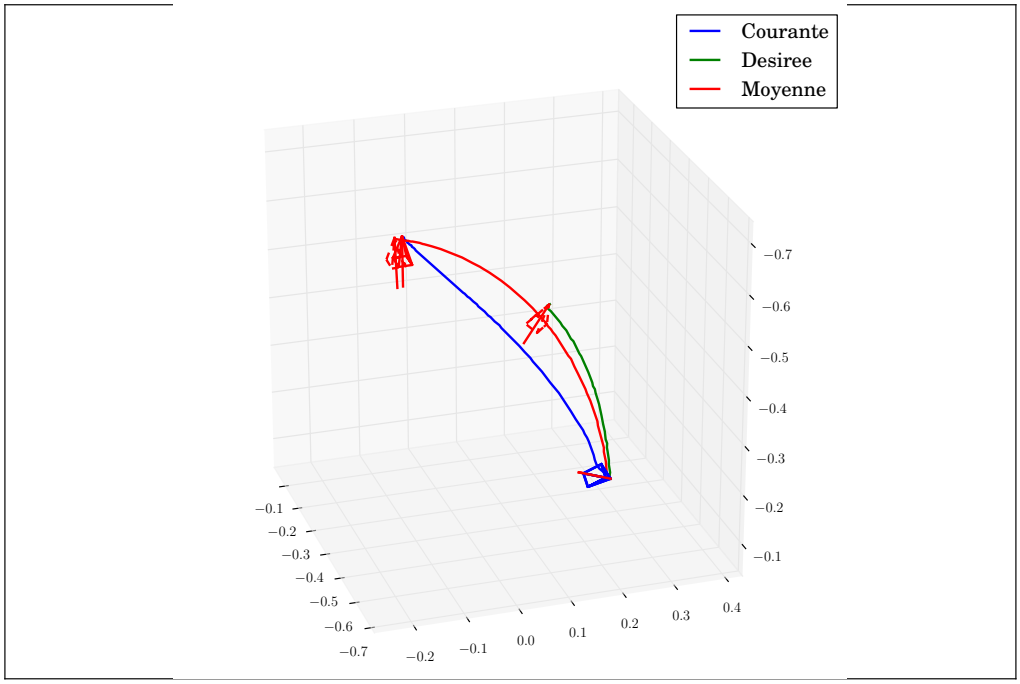


FIG. 4.15 – Choix de la matrice de combinaison. Trajectoires pour $\mathbf{C}_{\text{cur}}$ (rouge), $\mathbf{C}_*$ (cyan) et $\mathbf{C}_{\frac{1}{2}}$ (magenta). La loi de commande utilisant $\mathbf{C}_*$ perd le signal visuel et ne peut aller jusqu'à la position désirée.

4.1.6 Conclusion

À travers la coopération multi-caméras nous avons tenté d'illustrer le comportement bas niveau de l'approche proposée. Les schémas mono-capteur ont été comparés d'une part à l'approche par moyenne des vitesses, et d'autre part à celle utilisant le jacobien étendu. Pour cette dernière, les différents choix de loi de commande exposés en Section 3.2.1.2 ont été comparés.

Il apparaît nettement que le jacobien étendu montre un meilleur comportement que la moyenne des vitesses. Les expériences exposées mettent en jeu un large déséquilibre des contributions des capteurs. Sans pondération, le jacobien étendu assure un découplage des informations visuelles et montre de bonnes propriétés. Les différentes matrices de combinaison montrent des comportements très proches et il paraît difficile de statuer sur la supériorité de l'une d'entre elles dans le cas général. Dans la suite on utilisera simplement celle utilisant l'estimation courante du jacobien, soit $\mathbf{C}_{\text{cur}}$.

Outre la contrainte de visibilité, une pondération optimale pour le comportement du système utilisant le jacobien étendu relèverait des techniques de réglage des gains des différentes informations capteurs, comme en commande LQ. Pour un grand nombre d'informations visuelles, cela s'avérerait très délicat à mettre en œuvre en pratique. La partie constante de la pondération générique (3.23) est donc réglée par défaut à 1 dans la suite de ce chapitre.

Nous présentons maintenant notre approche pour la contrainte de visibilité.

4.2 Asservissement visuel hybride

Comme exposé en Section 1.3.4, deux principales approches existent en asservissement visuel. Utiliser des informations 2D assure de bonnes propriétés dans l'image, mais peut générer des trajectoires peu satisfaisantes dans l'espace [Chaumette 98]. À l'inverse, les informations 3D induisent un mouvement satisfaisant dans l'espace mais la cible observée peut sortir de l'image. De nombreux schémas ont été conçus pour respecter la contrainte de visibilité tout en essayant d'obtenir la meilleure trajectoire 3D. [Morel 00] développe l'asservissement visuel 2D1/2 et utilise comme informations visuelles le centre et le rayon du cercle contenant tous les points. En pratique, quand l'objet est grand dans l'image certains points peuvent être perdus et une stratégie commutative est mise en place. Dans [Corke 01] une stratégie partitionnée isole l'axe optique dans la loi de commande. Si un point s'approche du bord de l'image, la caméra recule automatiquement pour le conserver dans le champ de vue. Cela conduit malheureusement à de nombreux mouvements arrière, qui ne sont pas optimaux en terme de trajectoire 3D. Dans [Chesi 04], une approche commutative permet de désactiver les composantes linéaires ou angulaires du contrôleur PBVS. Si la cible approche du bord de l'image, la loi de commande n'est appliquée que partiellement afin de garantir la visibilité. Une autre approche commutative est proposée dans [Gans 07]. Les deux schémas IBVS et PBVS sont utilisés, et la commande passe de l'un à l'autre dès que l'erreur correspondante devient trop importante. Le critère de commutation est fonction de l'erreur dans l'espace des mesures 2D et 3D. Cette erreur n'est cependant pas directement liée à la contrainte de visibilité (elle ne dépend pas des dimensions de l'image).

Le niveau acceptable pour l'erreur 2D doit donc être réglé en fonction du cas étudié : si il est trop élevé, un point pourrait sortir de l'image. Si il est trop faible, la commande pourrait rester en IBVS. Dans [Allibert 10] la contrainte de visibilité est abordée sous l'angle de la commande prédictive. À chaque acquisition une trajectoire est planifiée et garantit le respect de la contrainte sur un horizon de temps.

Une loi de commande hybride a été proposée dans [Hafez 07b], avec une fusion par jacobien étendu (3.20) qui ne prend pas en compte la pondération dans le jacobien. De façon similaire à la coopération multi-caméras exposée en Section 4.1.4, la fusion par moyenne des vitesses (3.15) est utilisée dans [Hafez 08] pour la visibilité. Ces travaux sont les plus proches des nôtres. La première différence est la prise en compte de $\mathbf{H}$ dans l'inversion dans notre cas. Notre approche établit également un poids différent pour chaque information visuelle, là où [Hafez 07b] ne définit que le compromis général entre les schémas 2D et 3D.

Nous présentons maintenant notre approche pour l'asservissement visuel hybride. Des simulations exhaustives sont ensuite exposées. Nous comparons également en simulation notre approche avec celles de [Malis 99], [Gans 07] et [Hafez 08]. Enfin, des résultats sont présentés pour un asservissement visuel hybride sur un robot cartésien.

4.2.1 Loi de commande hybride

Nous considérons un schéma classique d'asservissement visuel 3D. La pose utilisée est celle qui amène à une ligne droite dans l'espace : $\mathbf{s} = ({}^{c*}\mathbf{t}_c, {}^{c*}\theta\mathbf{u}_c)$. La caméra est supposée calibrée et on note ${}^c\mathbf{M}_o$ la pose de la caméra par rapport à l'objet observé.

Nous considérons un ensemble de p points 3D de coordonnées $({}^o\mathbf{x}_1, \ldots, {}^o\mathbf{x}_p)$ rattachés au repère $\mathcal{F}_o$ de l'objet. De nombreux modèles CAO sont définis par des points 3D : les p points choisis pour la visibilité peuvent typiquement décrire l'enveloppe de l'objet. En connaissant ${}^c\mathbf{M}_o$, un point 3D de coordonnées ${}^o\mathbf{x}_i$ se projette en un point image (x_i, y_i) par la relation exposée en Section 1.3.3 :

$$\left\{ \begin{array}{ll} x_i & = X_i/Z_i \\ y_i & = Y_i/Z_i \end{array} \right. \tag{4.5}$$

avec $(X_i, Y_i, Z_i, 1) = {}^c\mathbf{M}_o {}^o\tilde{\mathbf{x}}_i$. Un vecteur d'informations 2D $\mathbf{s}_{2d}$ peut ainsi être défini à partie de ${}^c\mathbf{M}_o$ et des points choisis. De la même façon, les informations désirées $\mathbf{s}^*_{2d}$ sont définies à partir de ${}^c\mathbf{M}^*_o$. Les informations capteur du schéma hybride s'expriment donc par :

$$\mathbf{s} = \left[\begin{array}{l} \mathbf{s}_{3d} \\ \mathbf{s}_{2d} \end{array} \right] \begin{array}{l} \cdots \text{ PBVS (dim. 6)} \\ \cdots \text{ Visibilité (dim. } 2\times p) \end{array} \tag{4.6}$$

La pondération générique (3.23) est appliquée. L'asservissement visuel repose sur le schéma PBVS, les poids associés sont donc toujours égaux à 1 : $\mathbf{H}_{3d} = \mathbb{I}_6$.

Comme exposé dans (3.26), une région de confiance peut être paramétrée dans l'image par $\rho \in [0, 0.5]$. La pondération correspondante (3.27) est utilisée pour les informations 2D :

$$\left\{ \begin{array}{l} h^t_x = 0\,,\ h^c_x = h^c(x) \\ h^t_y = 0\,,\ h^c_y = h^c(y) \end{array} \right. \tag{4.7}$$

La pondération associée est schématisée sur la Figure 4.16.

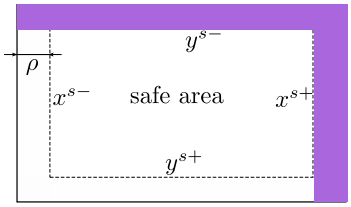


FIG. 4.16 – Pondération dans l'image. Les poids sont nuls dans la région centrale, et augmentent indéfiniment au voisinage des bords.

La pose étant estimée, les profondeurs sont connues en temps réel et la matrice d'interaction courante peut être utilisée. Il est intéressant de relever que le minimum global du schéma hybride correspond nécessairement à la pose désirée quelle que soit la pondération $\mathbf{H}_{2d}$. En effet, si $\mathbf{s}_{3d} = \mathbf{s}^*_{3d}$ alors ${}^c\mathbf{M}_o = {}^c\mathbf{M}^*_o$ et donc $\mathbf{s}_{2d} = \mathbf{s}^*_{2d}$. L'asservissement hybride se situe donc dans la configuration Z2 exposée en Section 3.2.4.2. Le schéma pouvant converger vers la pose désirée même si des contraintes sont activées, le paramètre d'évitement d'un minimum global non désiré (Section 3.2.4.2) n'est pas utilisé ici. Nous n'utilisons pas non plus la stratégie de sortie d'un minimum local (Section 3.2.4.1). Nous verrons dans la prochaine section que ce cas n'a jamais été rencontré. Le gain adaptatif décrit en Section 3.2.4.3 est en revanche mis en place pour ralentir le système quand des points approchent du bord de l'image.
Nous présentons maintenant les résultats de simulations exhaustives.

4.2.2 Simulations exhaustives

Dans cette section, la cible est composée de 4 points formant un carré. Les bords de l'image correspondent à $x^{\pm} = \pm 0.16$ et $y^{\pm} = \pm 0.12$, en coordonnées métriques normalisées. La zone de visibilité est paramétrée par $\rho = 5\%$, ce qui correspond à une distance de 0.016 pour les abscisses et 0.012 pour les ordonnées.

700 poses aléatoires ont été générées telles que la cible est visible dans l'image. Un minimum est spécifié pour l'aire de l'objet projeté, afin de ne pas générer de poses trop éloignées, ou trop inclinées par rapport à la cible. Toutes les combinaisons de poses initiale et désirée sont simulées, soit 244650 asservissements. Nous séparons les résultats en 3 catégories :

- Dans 1739 cas, soit 0.7%, un des points sort de l'image. Ces cas correspondent à des situations où l'objet sort partiellement dès la première itération. La contrainte n'a donc pas le temps d'être activée.
- Dans 2791 des cas restant, soit 1.1%, un des poids dépasse la valeur de 10. Pour des raisons de lisibilité, nous les isolons des cas suivants.
- Dans les 242334 autres cas, soit 98.2%, le schéma hybride converge vers la pose désirée sans perte de visibilité.

En pratique, les cas non satisfaisants convergent avec un gain plus faible. Nous étudions maintenant les cas satisfaisants.

La distance minimum atteinte par rapport aux bords de l'image lors des simulations est représentée en Figure 4.17. On constate que le pic est atteint autour de 0.005, soit environ

la moitié de la bande des 5%. Imposer des poses relativement proches de l'objet observé fait que seuls 2% des cas restent entièrement dans la zone de confiance ($d_{\min} > 0.016$), ce qui correspond à un PBVS pur.

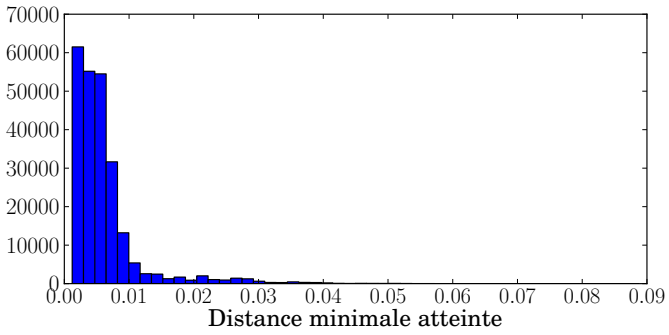


FIG. 4.17 – Distance minimale atteinte par rapport aux bords de l'image

La Figure 4.18 représente la distribution des valeurs maximums atteintes par les poids. On rappelle que le schéma PBVS a un poids de 1 pendant tout l'asservissement. Comme indiqué précédemment, par souci de lisibilité nous nous limitons ici aux cas où le poids maximum est inférieur à 10. Le pic en 0 inclut les 2% de cas où la contrainte n'a pas été activée du tout ($h = 0$). Comme annoncé en Section 3.2.2.4, le schéma proposé permet de respecter des contraintes avec une pondération relativement faible. Le pic se situe autour de 1.5, ce qui ne met pas en danger le conditionnement du système. La pondération maximale décroît ensuite rapidement pour devenir anecdotique au dessus de 10. Ces valeurs sont compatibles avec la méthode proposée.

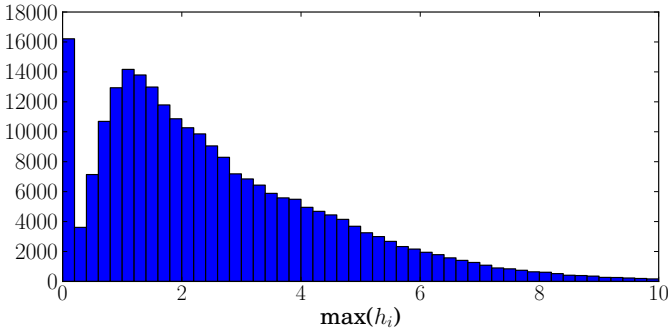


FIG. 4.18 – Valeur maximale atteinte par les poids durant l'asservissement

Le gain adaptatif (3.72) est représenté en Figure 4.19. Encore une fois, le pic à 1 inclut les cas où les contraintes n'ont pas été activées. On voit que dans le cas général, le schéma ralentit très légèrement ($\lambda > 0.9$) dans le voisinage des contraintes. Rappelons que λ dépend de la norme de $\mathbf{H}$, donc diminue peu a priori si une seule contrainte est activée, ce qui est le cas la plupart du temps.

De précédentes simulations exhaustives, n'utilisant pas le gain adaptatif, aboutissaient à des résultats nettement moins satisfaisants avec des pertes plus fréquentes de la visibilité.

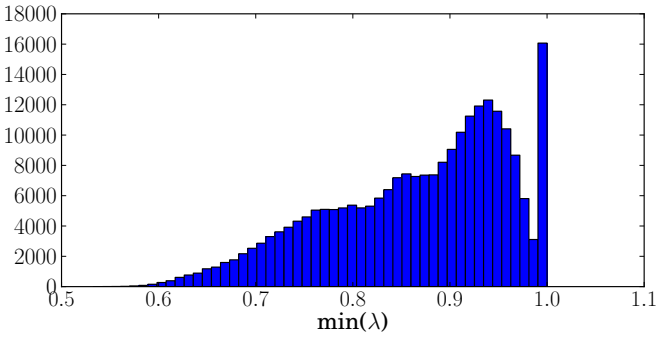


FIG. 4.19 – Valeur minimale atteinte par le gain adaptatif λ durant l'asservissement

La pondération proposée semble donc répondre aux contraintes. Aucun minimum local n'a été rencontré, et seuls les cas extrêmes correspondent à des violations de contraintes. La fonction de poids non bornée atteint en pratique des valeurs relativement faibles. Ces simulations exhaustives sont toutes réalisées sur une cible composée de 4 points formant un carré. Les résultats seraient sans doute légèrement différents pour d'autres cibles visuelles, toutefois la très forte proportion de cas satisfaisants laisse penser que l'approche proposée possède d'excellentes propriétés de convergence globale. Nous comparons maintenant notre approche avec celles de la littérature.

4.2.3 Comparaison avec d'autres approches

Dans cette section l'approche proposée est comparée avec celles de [Malis 99] (2D1/2, schéma implicite), [Gans 07] (asservissement commutatif) et [Hafez 08] (AV 5D, jacobien étendu avec pondération par tâche). La stratégie de pondération proposée dans [Hafez 08] est modifiée pour ce comparatif, car elle ne permet pas de respecter parfaitement la contrainte de visibilité. En effet, la pondération proposée dans [Hafez 08] est :

$$\begin{cases} h_{2d} = \frac{1}{1+\exp(-\exp(-D_{\min}^2))} \\ h_{3d} = 1 - h_{2d} \end{cases} \tag{4.8}$$

où h_{2d} est la pondération du schéma 2D, h_{3d} celle du schéma 3D et où $D_{\min}$ est la distance minimale entre un point et le bord de l'image. On remarque que le poids maximum du schéma 2D est $h_{2d}(D_{\min} = 0) = 0.73$ pour $h_{3d}(D_{\min} = 0) = 0.27$. Ramené à l'échelle de notre approche ($h_{3d} = 1$), cela correspond à un poids maximum de 2.7 pour le schéma 2D. Cette valeur est souvent insuffisante pour garantir la contrainte de visibilité. Pour comparer tout de même la pondération individuelle et la pondération par tâche (3.19), nous appliquons la pondération exposée dans ce mémoire. Le poids du schéma 2D est alors le poids du point le plus près du bord :

$$\begin{cases} h_{2d} = \max(h_i) \\ h_{3d} = 1 \end{cases} \tag{4.9}$$

Deux configurations sont présentées, montrant les limites de chaque approche. Ces configurations sont issues des cas satisfaisants des simulations exhaustives, elles ne correspondent donc pas aux cas d'école en asservissement visuel.

4.2.3.1 Première configuration

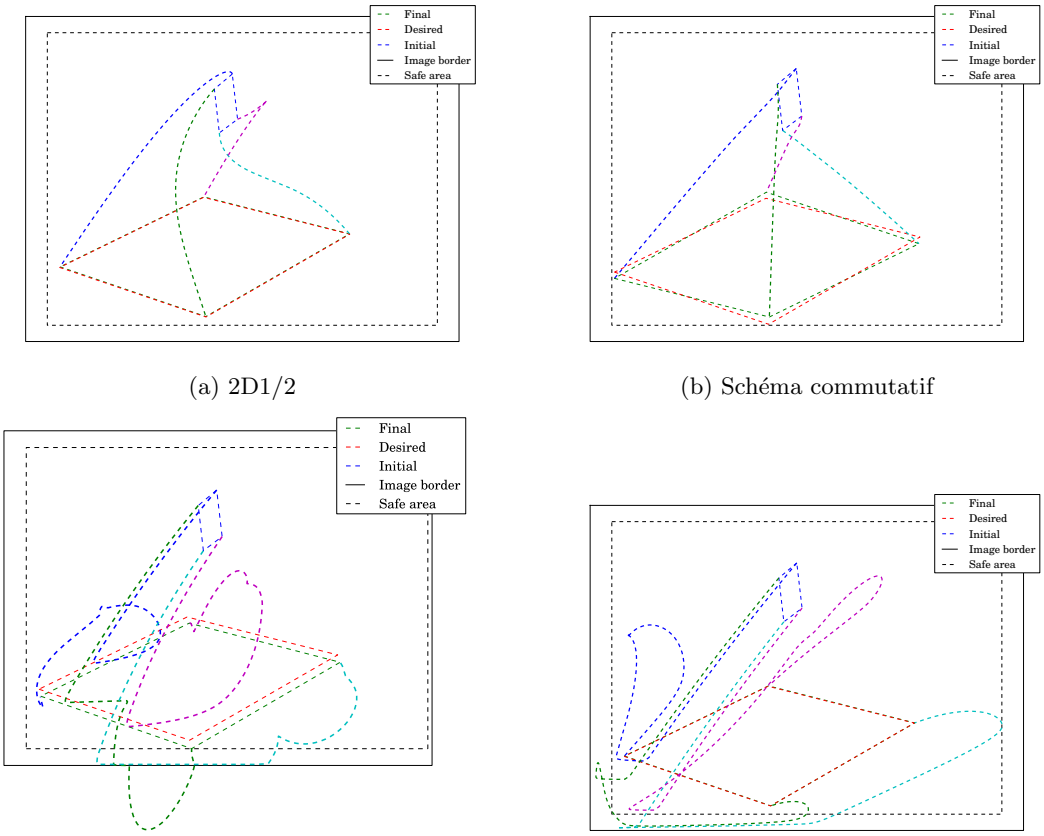
Une première simulation est faite pour la configuration suivante :

$$\begin{cases} {}^{c}\mathbf{M}_o = (-0.04, -0.16, 2.76, -0.26, -1.98, 0.16) \text{ (m,rad)} \\ {}^{c}\mathbf{M}_o^* = (-0.02, 0.03, 0.62, 1.87, 0.66, -0.59) \text{ (m,rad)} \end{cases} \tag{4.10}$$

Les comportements dans l'image sont représentés en Figure 4.20. Le schéma 2D1/2 (Figure 4.20a) converge vers la pose désirée en gardant l'objet dans l'image. L'asservissement visuel commutatif (Figure 4.20b) démarre en PBVS. Il passe rapidement en IBVS, mais converge alors vers un minimum local. Les critères de commutation n'étant pas directement liés à la visibilité, ils sont relativement difficiles à régler. Si le schéma passe trop tardivement en IBVS, il sort de l'image aux premières itérations. À l'inverse, adoucir le critère de passage en IBVS fait que la commande est en réalité un IBVS tout du long, et qui converge vers un minimum local. L'asservissement 5D (Figure 4.20c) souffre de sa pondération globale. Lors de l'activation de la contrainte ($h_{2d} > 0$), la configuration est telle que le point le plus près du bord ne converge pas vers l'intérieur. Il sort alors de l'image. De fait, en asservissement visuel 2D avec plus de 3 points, rien ne garantit que tous les points suivent une ligne droite. L'absence de prise en compte de la pondération dans le jacobien fait que le schéma converge vers un minimum local, alors que tous les points sont dans la zone de confiance. On rappelle que dans cette configuration, le schéma proposé dans [Hafez 08] est alors : $\mathbf{v}_c = -\lambda \mathbf{L}_s^+(\mathbf{e}_{3d}, 0_{2d})$. Il se comporte comme si les informations 2D avaient convergé, induisant un comportement conservatif.

Enfin, notre approche (Figure 4.20d) respecte la contrainte de visibilité et converge vers la pose désirée.

Les trajectoires dans l'espace des différentes approches sont représentées en Figure 4.21. Le schéma 2D1/2 (bleu) montre une trajectoire très éloignée de la ligne droite. Notre approche (cyan) est plus rectiligne et très proche de la trajectoire de l'asservissement visuel 5D (rouge). Le schéma commutatif (vert) étant rapidement en IBVS, la trajectoire est logiquement très différente.



(a) 2D1/2

(b) Schéma commutatif

(c) 5D VS

(d) Hybride

FIG. 4.20 – Comportement dans l'image pour les 4 schémas. Image initiale (bleu), finale (vert) et désirée (rouge). Le 2D1/2 (a) converge rapidement vers la pose désirée (le carré vert n'apparaît pas). L'approche commutative (b) converge vers un minimum local. La méthode AV 5D (c) sort également de l'image avant de converger vers un minimum local. L'approche proposée (d) respecte la contrainte de visibilité et converge vers la pose désirée.

Les vitesses de la caméra obtenues pour les deux schémas satisfaisants sont représentés sur la Figure 4.22. L'asservissement visuel 2D1/2 (Figure 4.22a) montre une décroissance classique des vitesses. Notre approche (Figure 4.22b) montre une forte modification de la commande après les premières itérations. La vitesse est ensuite très régulière jusqu'à convergence.

Notre approche est détaillée en Figure 4.23. La Figure 4.23b indique que seules 4 informations 2D sont utilisées pendant l'asservissement, dont une très brièvement (magenta lors des premières itérations). On constate en effet sur la Figure 4.20d que :

- un point fait un aller-retour rapide dans le bas de l'image
- un autre point passe dans le bas de l'image
- un dernier point passe dans le bas de l'image, puis dans la bande gauche

Par ailleurs les poids restent relativement faibles ($h_i < 5$), suffisamment pour ne pas mettre

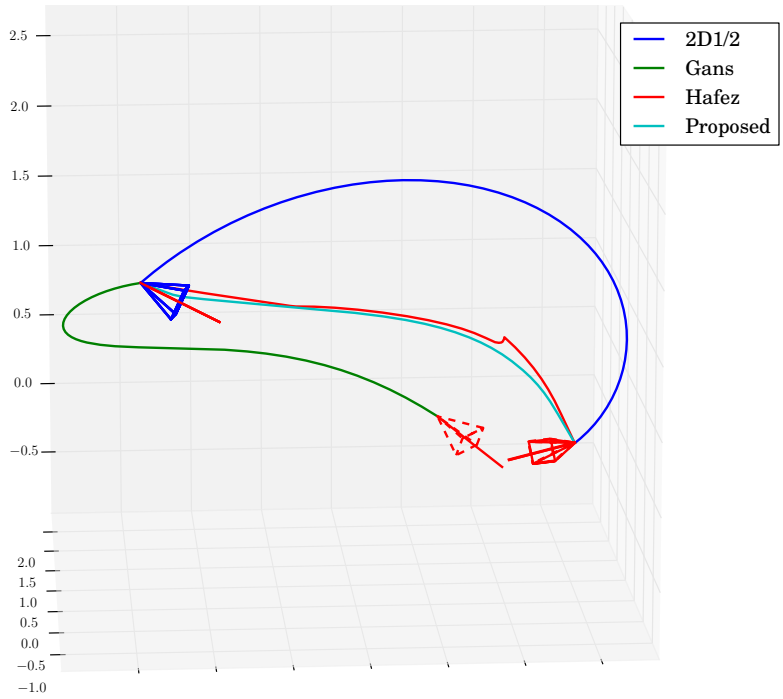
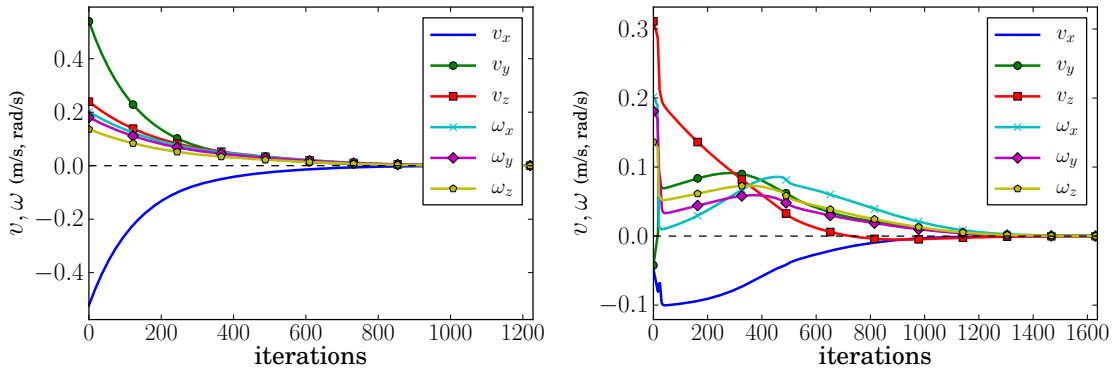


FIG. 4.21 – Trajectoires des différents schémas. L'approche 2D1/2 (bleue) est éloignée de la ligne droite. L'asservissement visuel 5D (rouge) et notre approche (cyan) ont des trajectoires très proches et relativement rectilignes. L'asservissement visuel commutatif (vert) est en mode IBVS et suit donc une trajectoire très différente. Il converge vers un minimum local.



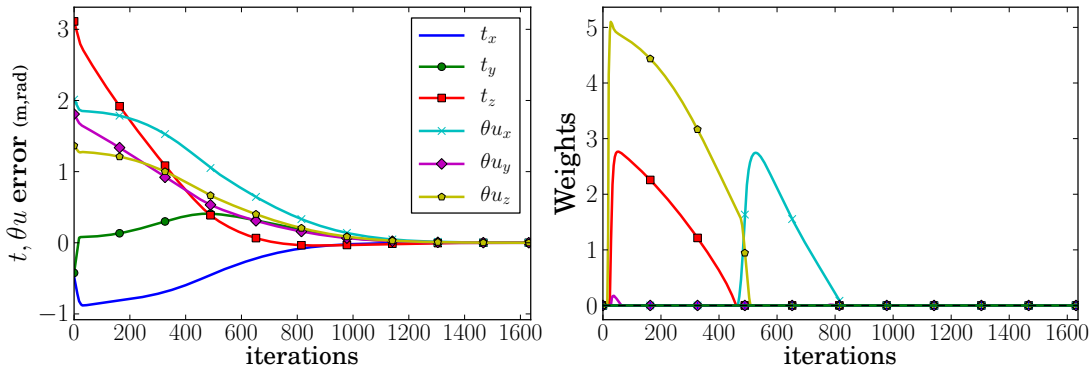
(a) $\mathbf{v}_s$ en asservissement 2D1/2

(b) $\mathbf{v}_s$ avec notre approche

FIG. 4.22 – Vitesses des schémas satisfaisants.

en danger le conditionnement du système.

On constate donc qu'une pondération à l'échelle des informations capteur permet de respecter des contraintes. Un avantage notable est la facilité de mise en œuvre, par rapport à d'autres approches dont les gains peuvent être difficiles à régler.



(a) Erreur dans l'espace 3D (b) Pondération des points

FIG. 4.23 – Détail de l'approche proposée. L'erreur de la pose décroît de façon découplée à partir de l'itération 500 (a). Seuls 4 informations 2D sont utilisées pendant l'asservissement (b).

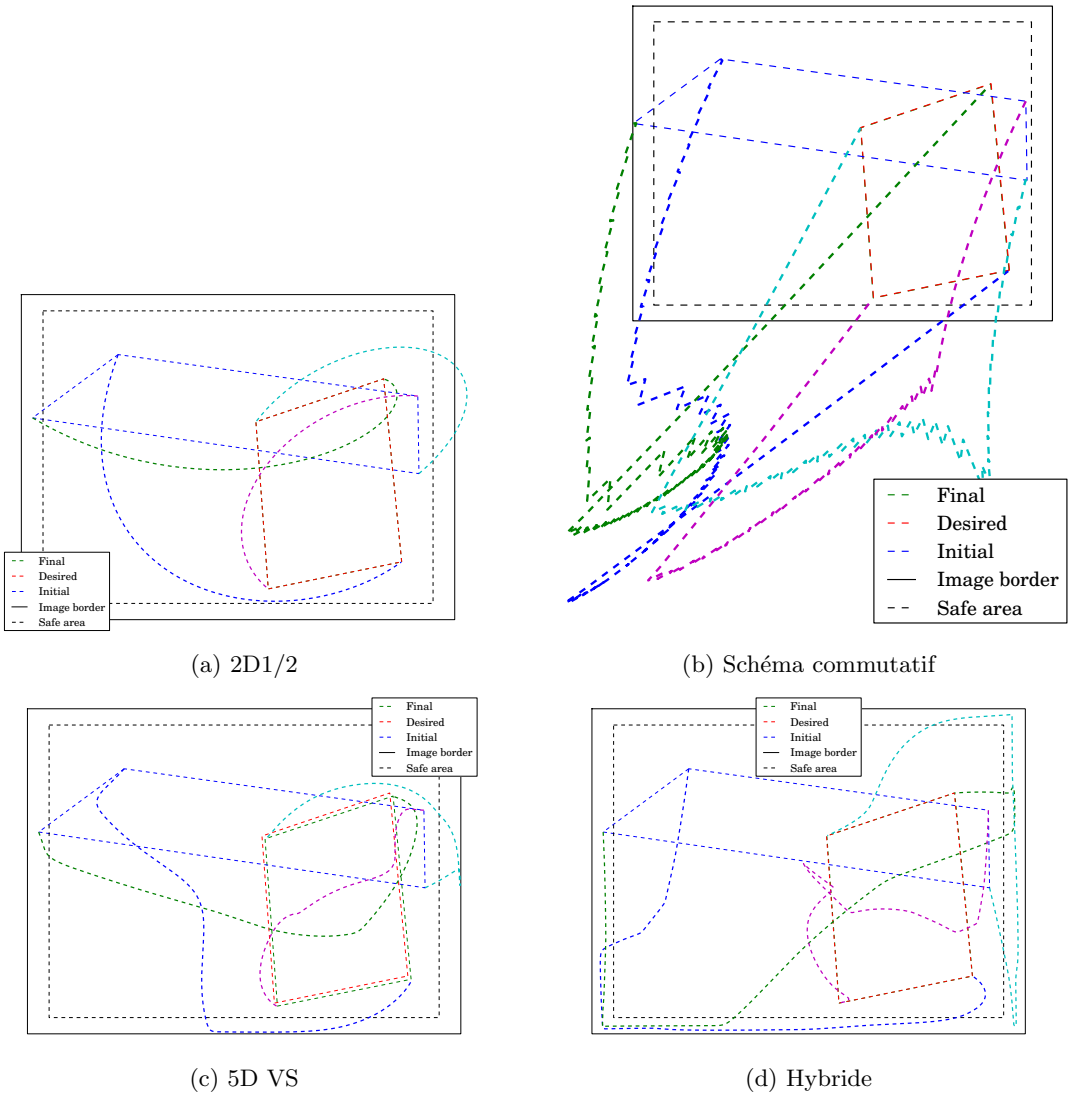
4.2.3.2 Seconde configuration

Les poses initiale et désirée sont ici définies par :

$$\begin{cases} {}^{c}\mathbf{M}_o = (0., -0.02, 0.38, 1.23, 1.54, -1.07) \text{ (m,rad)} \\ {}^{c}\mathbf{M}_o^* = (0.05, 0.02, 0.72, 1.54, -1.82, -0.55) \text{ (m,rad)} \end{cases} \tag{4.11}$$

Les trajectoires dans l'image sont représentées en Figure 4.24. Cette fois, l'approche 2D1/2 laisse sortir un point de l'image. Étant globalement asymptotiquement stable, elle convergerait toutefois vers la pose désirée. Le schéma commutatif (Figure 4.24b) s'est encore une fois révélé difficile à régler. Nous avons favorisé le PBVS par rapport à la configuration précédente. L'objet sort rapidement de l'image (PBVS), puis commute fréquemment entre les deux schémas. Sur la fin, en mode IBVS il convergerait cette fois vers la pose désirée. L'asservissement visuel 5D (Figure 4.24c) reste cette fois dans l'image, mais converge vers un minimum local alors que tous les points sont dans la zone de confiance. Ne pas prendre en compte la matrice de pondération dans le jacobien ne semble pas vraiment procurer d'avantage. Notre approche converge vers la pose désirée.

Les trajectoires 3D sont comparées en Figure 4.25. La commande commutative, passant d'un mode à l'autre, a une trajectoire particulière. Les trois autres schémas partagent des trajectoires voisines. Notre approche reste la plus rectiligne.



(a) 2D1/2

(b) Schéma commutatif

(c) 5D VS

(d) Hybride

FIG. 4.24 – Comportement dans l'image pour les 4 schémas. Le 2D1/2 (a) converge vers la pose désirée mais un point sort de l'image. L'approche commutative (b) converge vers un minimum local. La méthode 5D VS (c) reste dans l'image mais converge vers un minimum local. L'approche proposée (d) respecte la contrainte de visibilité et converge vers la pose désirée.

Les pondérations des deux schémas satisfaisants sont représentées sur la Figure 4.26. L'asservissement visuel 5D (Figure 4.26a) active l'ensemble des informations 2D en un seul bloc. Le poids h_{2d} doit atteindre une valeur assez élevée pour permettre de respecter la contrainte de visibilité. À l'inverse, en pondérant à l'échelle des informations capteur (Figure 4.26b), de faibles poids suffisent à respecter la contrainte. Par rapport à la simulation précédente, de nombreuses informations 2D sont ici utilisées au long de l'asservissement.

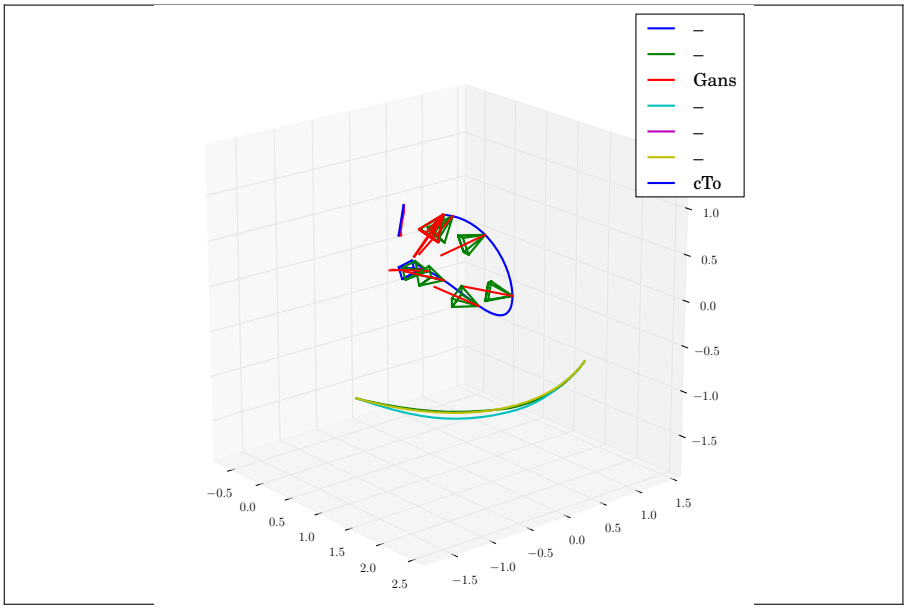
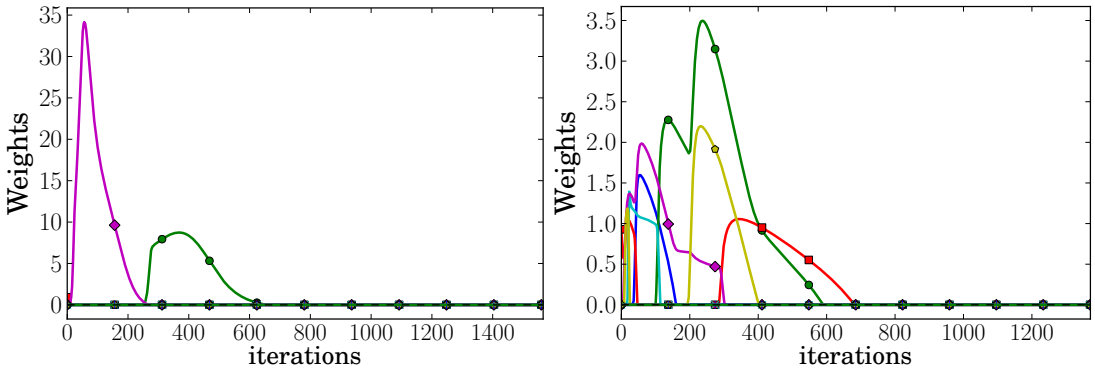


FIG. 4.25 – Trajectoires des différents schémas. L'approche 2D1/2 (bleue) est proche du schéma 5D (rouge). Tous deux sont éloignés d'une droite dans l'espace. L'asservissement commutatif (vert) montre une trajectoire très différente faisant apparaître les commutations sur la droite de l'image. Notre approche (cyan) est la plus rectiligne des 4.



(a) Pondération du schéma 2D en 5D VS (b) Pondération pour l'approche hybride proposée

FIG. 4.26 – Évolution des poids pour les approches 5D (a) et proposée (b).

4.2.3.3 Conclusion

Notre approche a été comparée avec celles de l'état de l'art dans deux configurations. On rappelle que ces configurations sont issues des simulations exhaustives et sont particulièrement difficiles. Elles mettent en œuvre un large déplacement à la fois linéaire et angulaire, avec un objet relativement grand dans les images initiale et désirée. et une pose désirée qui rend possible les minimums locaux en asservissement visuel 2D.

L'asservissement visuel 2D1/2 [Malis 99] donne des résultats satisfaisants sur le premier

cas, où l'objet est petit dans l'image initiale. En revanche, ne pas prendre en compte de façon explicite la contrainte de visibilité laisse sortir un des points sur la deuxième simulation.

Le schéma commutatif [Gans 07] s'est révélé particulièrement difficile à régler. Dans le cas général, il n'y a effectivement pas de lien direct entre l'erreur 2D $\mathbf{e}_{2d}$ maximum acceptable et la contrainte de visibilité. En effet, si l'objet désiré est près d'un bord, l'erreur visuelle peut être faible même avec un objet partiellement hors de l'image. L'asservissement commutatif réussit sur des cas plus simples, notamment sans minima locaux, et là où les autres approches réussissent également.

L'approche 5D de [Hafez 08], avec une modification de la stratégie de pondération, s'est révélée handicapée par la non prise en compte de la pondération dans le jacobien étendu. Par ailleurs, pondérer au niveau des tâches est limitant par rapport à une pondération au niveau des informations.

Notre approche confirme les conclusions des simulations exhaustives. Les poids permettant de respecter la visibilité restent relativement faibles. Les trajectoires des points dans l'image montrent des points qui ont tendance à longer les bords. Les trajectoires 3D correspondantes sont en revanche satisfaisantes, et susceptibles de respecter facilement d'autres contraintes comme les butées articulaires.

4.2.4 Expérience sur un robot réel

L'expérience est réalisée sur le robot cartésien Afma6. La caméra est calibrée et observe une pièce mécanique dont le modèle 3D est connu. Les arêtes de l'objet sont suivies [Comport 05] et la pose ${}^{c}\mathbf{M}_{o}$ est estimée en temps réel. Les points 3D utilisés pour la visibilité sont simplement les 24 nœud du modèle. Notons qu'on pourrait réduire le nombre de points 3D en n'utilisant que ceux correspondant à l'enveloppe de l'objet.

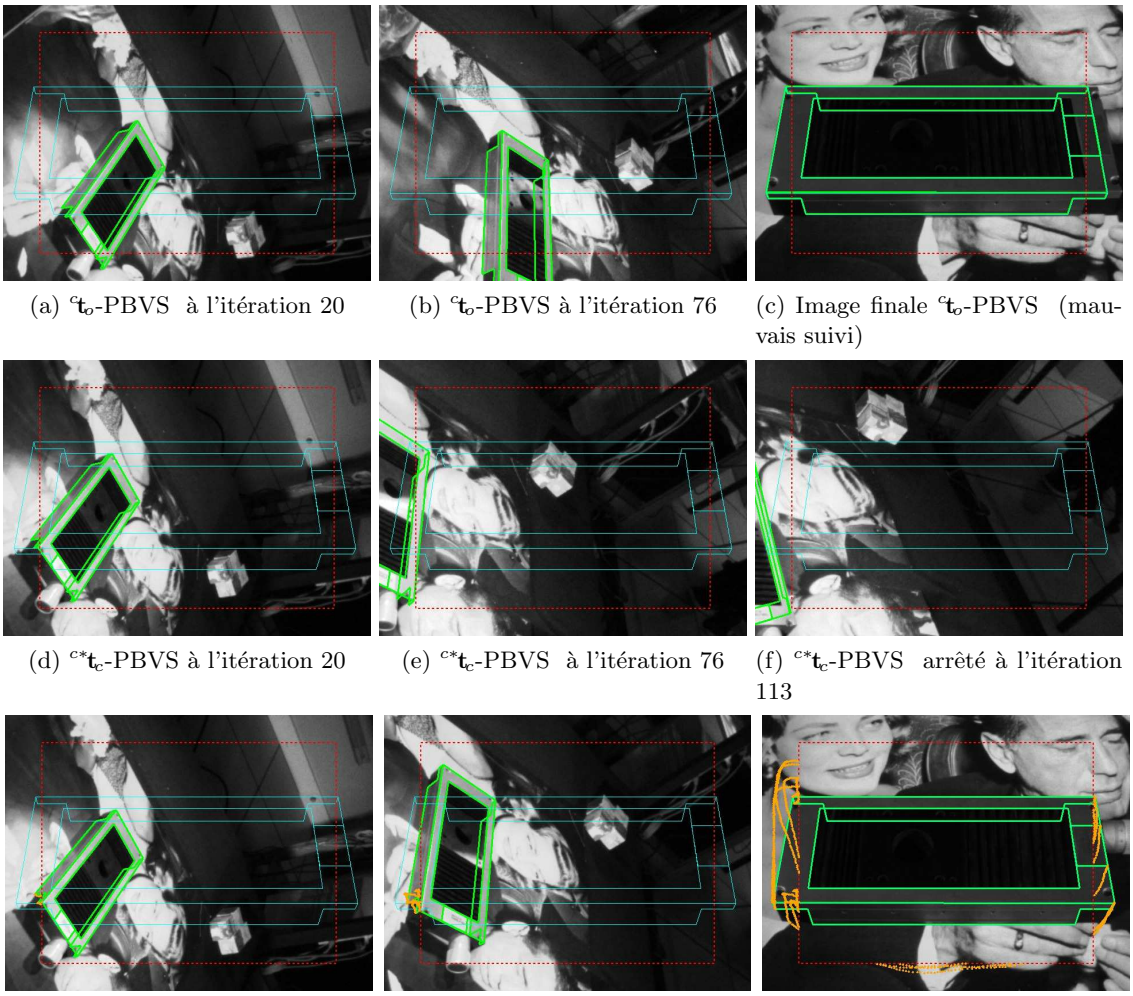
Les deux asservissements PBVS classiques sont également réalisés :

- ${}^{c}\mathbf{t}_{o}$-PBVS utilise la paramétrisation $\mathbf{s} = ({}^{c}\mathbf{t}_{o}, {}^{c*}\theta\mathbf{u}_{c})$ et assure que l'origine du repère objet $\mathcal{F}_o$ suit une ligne droite dans l'image.
- ${}^{c*}\mathbf{t}_{c}$-PBVS utilise la paramétrisation $\mathbf{s} = ({}^{c*}\mathbf{t}_{c}, {}^{c*}\theta\mathbf{u}_{c})$, ce qui correspond à une trajectoire rectiligne dans l'espace.

Notre asservissement hybride repose sur la paramétrisation $\mathbf{s}_{3d} = ({}^{c*}\mathbf{t}_{c}, {}^{c*}\theta\mathbf{u}_{c})$. L'objectif est donc la trajectoire la plus rectiligne possible dans l'espace, tout en respectant la visibilité.

La Figure 4.27 montre le comportement dans l'image pour les différents schémas.

Comme dans la seconde simulation, l'approche ${}^{c}\mathbf{t}_{o}$-PBVS perd une partie de l'objet ce qui conduit à un mauvais suivi visuel et un calcul erroné de la pose. Pour de grandes rotations ou quand l'objet est grand dans l'image, cette approche ne garantit donc pas que tout l'objet reste visible. Le schéma ${}^{c*}\mathbf{t}_{c}$-PBVS ne réalisant aucun contrôle dans l'image, l'objet est rapidement perdu. Pour notre approche, la zone de confiance est représentée en pointillés rouges. Les points 2D utilisés sont affichés en orange. L'objet suivi se dirige, comme en ${}^{c*}\mathbf{t}_{c}$-PBVS, vers la gauche de l'image. L'activation des points 2D permet de conserver l'objet dans le champ de vue jusqu'à la pose désirée. Notons que cette fois, certains points sont près du bord dans l'image désirée. Comme évoqué en Section 4.2.1, la valeur désirée $\mathbf{s}^{*}_{2d}$ étant calculée à partir de $\mathbf{s}^{*}_{3d}$, la condition Z2 exposée en Section 3.2.4.2 est vérifiée et la configuration finale ne nécessite pas d'annuler artificiellement les poids.



(a) ${}^{c}\mathbf{t}_{o}$-PBVS à l'itération 20 (b) ${}^{c}\mathbf{t}_{o}$-PBVS à l'itération 76 (c) Image finale ${}^{c}\mathbf{t}_{o}$-PBVS (mauvais suivi)

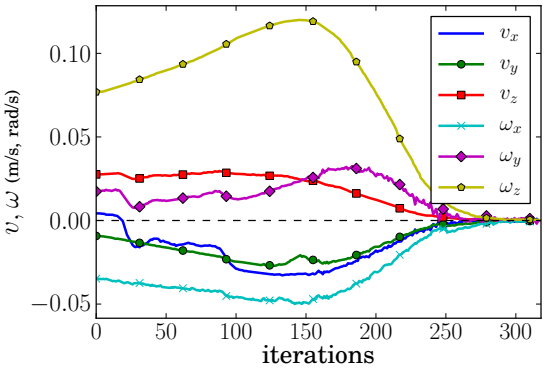
(d) ${}^{c*}\mathbf{t}_{c}$-PBVS à l'itération 20 (e) ${}^{c*}\mathbf{t}_{c}$-PBVS à l'itération 76 (f) ${}^{c*}\mathbf{t}_{c}$-PBVS arrêté à l'itération 113

(g) Approche hybride à l'itération 20 (h) Approche hybride à l'itération 76 (i) Image finale pour l'approche hybride (convergence)

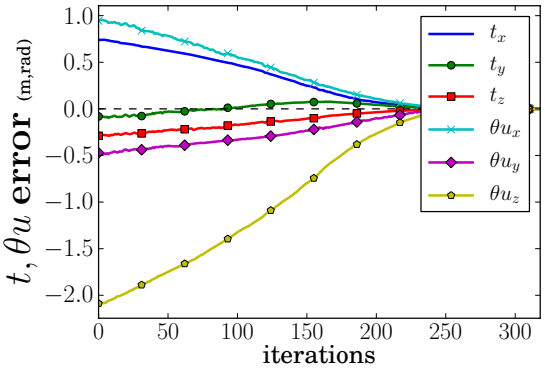
FIG. 4.27 – Évolution de l'image pour ${}^{c}\mathbf{t}_{o}$-PBVS (haut), ${}^{c*}\mathbf{t}_{c}$-PBVS (milieu) et pour l'approche proposée (bas). L'objet sort partiellement de l'image avec l'approche ${}^{c}\mathbf{t}_{o}$-PBVS, puis l'estimation de pose devient erronée, ce qui entraîne que la pose finale n'est pas celle désirée. Avec ${}^{c*}\mathbf{t}_{c}$-PBVS l'objet sort rapidement du champ de vision et le robot doit s'arrêter. Notre approche converge en respectant la contrainte de visibilité. Certains points 2D sont activés jusqu'à la position désirée.

Le comportement dans l'espace pour notre approche est représenté en Figure 4.28. On constate un large mouvement de rotation autour de l'axe optique (Figure 4.28a). L'erreur correspondante (Figure 4.28b) est en effet très importante. À l'itération 20, deux vitesses montrent une rapide variation : il s'agit des composantes v_x et ω_y qui régissent le mouvement horizontal dans l'image. Nous avons en effet vu en Figure 4.27h que c'est par la gauche que l'objet approche du bord de l'image. La convergence vers la pose désirée est très régulière. Cette observation est confirmée sur la Figure 4.28c : la trajectoire suivie dans l'espace est

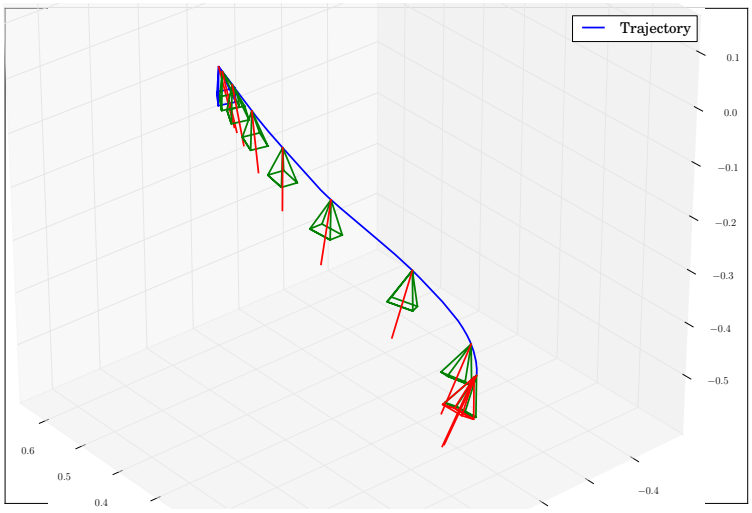
presque rectiligne. Les premières itérations sont très regroupées dans l'espace (nombreuses caméras vertes près de la pose initiale). Cela correspond aux mouvements indiqués qui sont essentiellement des rotations. Dans cette configuration, le fait d'imposer la contrainte de visibilité fait que la caméra commence par tourner avant de se déplacer vers la pose désirée.



(a) Vitesse de la caméra $\mathbf{v}_s$



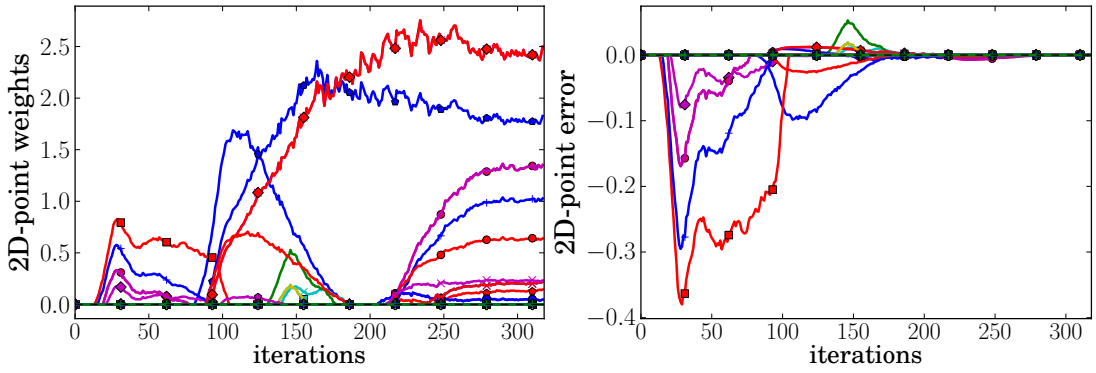
(b) Erreur 3D $\mathbf{e}_{3d}$



(c) Trajectoire dans l'espace

FIG. 4.28 – Comportement dans l'espace. La trajectoire suivie (c) est presque rectiligne.

Le comportement lié aux points 2D est représenté en Figure 4.29. Comme suggéré sur le tracé des vitesses, les premières informations 2D sont activées autour de l'itération 20. Ce premier groupe (itérations 20-80) permet d'éviter la sortie par la gauche. Les poids diminuent ensuite et d'autres informations 2D sont activées entre les itérations 100 et 170. Elles correspondent sur la Figure 4.27i aux points entrés dans la partie inférieure de l'image. Sur la fin, de nombreux poids convergent vers une valeur non-nulle : cela correspond aux points près des bords à la position désirée. L'erreur pondérée est représentée en Figure 4.29b. Si les deux phases de pondération (20-80 et 100-170) y sont visibles, l'erreur 2D est nulle à la fin malgré les poids non-nuls. Comme précédemment indiqué, cela vient du fait que la pose désirée correspond à $\mathbf{s}_{2d} = \mathbf{s}^*_{2d}$ donc $\mathbf{e}_{2d} = 0$ quelle que soit la pondération associée. On constate encore une fois que des poids faibles ($h_i < 2.5$) suffisent à respecter la contrainte de visibilité. Notamment après l'itération 100, l'erreur 2D est très faible et suffit à garder l'objet visible.



(a) Poids des informations 2D $\mathbf{H}_{2d}$

(b) Erreur 2D pondérée $\mathbf{H}_{2d}\mathbf{e}_{2d}$

FIG. 4.29 – Comportement dans l'image. Évolution des poids (a) et de l'erreur 2D pondérée (b). Les poids finaux sont non-nuls mais l'erreur correspondante est nulle car les points convergent dans l'image.

4.2.5 Conclusion

Notre approche a été appliquée au problème de la visibilité en asservissement visuel. La pondération générique a été utilisée, définissant en pratique une zone d'activation le long des bords de l'image. Le seul paramètre à régler nécessaire pour l'asservissement visuel hybride est ρ qui indique la largeur de la zone d'activation.

Les simulations exhaustives montrent que moins de 1% des cas ne respectent pas la contrainte de visibilité. En pratique, augmenter ρ améliore le taux de succès mais le comportement devient alors moins proche d'un asservissement visuel 3D. Diminuer le gain global λ permet également d'augmenter le taux de succès, mais il paraît difficile de déterminer un gain minimal assurant une visibilité systématique. Il est intéressant de relever qu'aucun minimum local n'a été rencontré, malgré un système surdéterminé ($m = 14$). Une explication est qu'il est très rare que de nombreux points soient activés en même temps. Par ailleurs, le minimum global correspond nécessairement à la pose désirée quels que soient les

points activés.

Notre approche se montre très efficace, tout en restant simple à mettre en œuvre. Le schéma commutatif est séduisant mais n'a pas pu être réglé convenablement. L'asservissement visuel 5D quant à lui souffre de la pondération au niveau des tâches. Un asservissement visuel 2D sur plus de 3 points ne garantit pas que chacun des points suit une ligne droite. Activer d'un bloc toutes les informations 2D ne permet donc pas de garantir que c'est le point le plus proche du bord qui va immédiatement faire demi-tour.

L'approche 2D1/2, réalisant une visibilité implicite, ne montre ses limites que si l'objet est grand dans l'image. À ce moment l'objet peut très bien sortir partiellement. Ceci peut conduire à des problèmes de suivi.

Il serait intéressant de réaliser une étude sur la stabilité de l'asservissement visuel hybride proposé. Le schéma étant surdéterminé par nature puisque 6 informations 3D sont complétées par $2p$ informations 2D, le noyau de $\mathbf{HL}_s$ n'est pas nul mais peut-être pourrait-on montrer que ses éléments ne correspondent pas à des poses réalisables.

Nous exposons maintenant les résultats en évitement de butées articulaires.

4.3 Évitement de butées articulaires

L'évitement de butées est un problème classique en robotique. Il se pose particulièrement en commande référencée capteur. En effet, une tâche définie dans l'espace des mesures capteurs peut générer des mouvements impossibles à réaliser par le robot. Des approches par projection de gradient ont été appliquées avec succès [Siciliano 91, Marchand 96, Yoshikawa 96, Zghal 02]. Dans ce cadre, l'évitement est modélisé par une fonction de coût dont le gradient est projeté dans le noyau de la tâche principale. Comme évoqué en Section 3.1.3, cette approche n'est valide que pour les systèmes redondants. La redondance directionnelle [Mansard 09a] permet d'assouplir cette limitation. Imposant une hiérarchie, elle ne garantit pas que les butées sont évitées si la tâche principale persiste à aller dans cette direction.

Dans [Marey 10c], le GPM est utilisé avec un opérateur de projection qui ne modifie pas la norme de la tâche principale. Les tâches de rang plein peuvent alors être considérées. L'évitement est adaptatif et permet d'assurer une modification minimale de la tâche principale. Le calcul du gain adaptatif de l'évitement ne prend toutefois pas en compte les situations où plusieurs butées sont évitées en même temps.

Des approches par commande hybride et compromis ont également été formulées pour l'évitement de butées. Dans [Nelson 95] une commande LQ est appliquée. Toutefois l'évitement n'est pas pris en compte dans le jacobien. En effet, avec des notations similaires ($\mathbf{Q} = \mathbf{H}^2$ et $\mathbf{R} = 0$) la commande correspondante s'exprime par :

$$\dot{\mathbf{q}} = -\lambda(\widehat{\mathbf{J}}_1\mathbf{H}^2\widehat{\mathbf{J}}_1^\top)^{-1}\left(\widehat{\mathbf{J}}_1^\top\mathbf{H}_1^2\mathbf{e}_1 + \frac{1}{2}\left(\frac{\partial c}{\partial \mathbf{q}}\right)^\top\right) \tag{4.12}$$

où $\mathbf{e}_1$ est la tâche référencée capteur (en l'occurence un asservissement visuel) et où $c(\mathbf{q})$ est la fonction de coût pour l'évitement. On voit que cette formulation ne prend pas en compte l'évitement dans la pseudo-inverse. Le schéma est donc moins bien découplé que les

approches par jacobien étendu. Par ailleurs la fonction de coût proposée est bornée, rien ne garantit donc que les butées sont évitées. Dans [Chan 95] une pseudo-inverse pondérée (3.21) est utilisée. Si cette approche est bien adaptée à l'évitement de butées, elle ne fait toutefois pas la différence entre les butées évitées naturellement (évitement passif) et celles qui ne le sont que grâce à la pondération (évitement actif). Dans sa mise en œuvre, le sens de variation des articulations est mesuré afin de n'appliquer l'évitement que si l'articulation se rapproche de la butée.

La difficulté des approches par compromis est de trouver la pondération qui permet d'éviter les butées en modifiant le moins possible la tâche principale. Nous détaillons tout d'abord l'approche considérée en Section 3.2.3.3 pour un évitement de butées pendant un asservissement visuel 2D sur p points.

4.3.1 Poids optimaux

Les informations visuelles sont notées $\mathbf{s}_{2d}$. Le schéma global s'exprime par :

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{2d} \\ \mathbf{q} \end{bmatrix} \quad \begin{matrix} \cdots \text{ IBVS (dim. } 2\times p) \\ \cdots \text{ Articulations (dim. 6)} \end{matrix} \tag{4.13}$$

Nous associons une pondération constante aux informations visuelles : $\mathbf{H}_{2d} = \mathbb{I}_{2p}$. La matrice de pondération des articulations est notée $\mathbf{H}_q$. La pondération associée est décrite en Section 3.2.3. Elle est paramétrée par ρ et ρ^{a} qui définissent deux intervalles :

- $\mathcal{I}^s = [q^{\mathrm{s}-}, q^{\mathrm{s}+}]$ est l'intervalle de confiance, où l'articulation n'est pas prise en compte.
- $\mathcal{I}^a = [q^{\mathrm{a}-}, q^{\mathrm{a}+}]$ définit la zone de pondération optimale. Si une seule articulation est proche de sa butée alors l'évitement arrive avant q^{a}. L'évitement peut être actif (C1, $h(q) > 0$) ou passif (C2, $h(q) = 0$) si l'asservissement visuel revient de lui-même dans la bonne direction.

En dehors de $\mathcal{I}^a$, la pondération générique s'applique : $h(q_i) = h^c(q_i)$. Le schéma de pondération est rappelé en Figure 4.30.

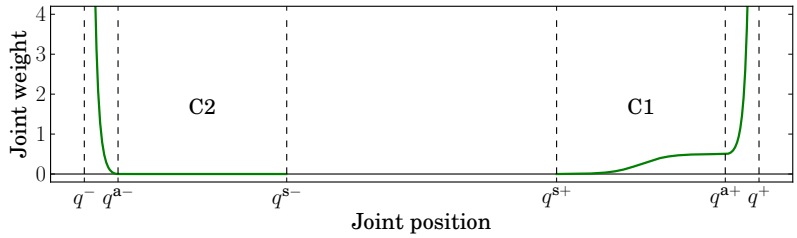


FIG. 4.30 – Stratégie de pondération pour les butées articulaires pour les évitements actif (C1) et passif (C2). La limite supérieure est en évitement actif ($h^{\mathrm{opt}} = 0.5$), la limite inférieure en évitement passif. La pondération est minimale sur l'intervalle d'activation $[q^{\mathrm{a}-}, q^{\mathrm{a}+}]$. En dehors de cet intervalle, la pondération générique assure que les butées sont évitées.

Le choix d'un asservissement visuel sur les coordonnée cartésiennes de points dans l'image est paradoxalement motivé par les mauvaises propriétés 3D de ce schéma. Il est en effet bien connu que quand une large rotation est nécessaire pour rejoindre la pose désirée,

ces informations génèrent un important mouvement en arrière [Chaumette 98]. Les butées sont alors susceptibles d'être atteintes.

4.3.2 Simulations pour une position désirée

Les simulations sont réalisées sur un robot cartésien à 6 degrés de liberté. La pose désirée est paramétrée par ${}^{c}\mathbf{M}_{o}^{*} = (0, 0, 0.5, 0, 0, 0)$ quand chaque articulation est au milieu de son intervalle : $\mathbf{q}^{*} = \frac{1}{2}(\mathbf{q}^{+} + \mathbf{q}^{-})$.
Un algorithme de remplissage d'espace (space-filling design) [Fang 06] explore tout l'espace articulaire. On a ainsi déterminé 1000 poses initiales où l'objet est devant la caméra. Ces simulations ne prennent pas en compte la contrainte de visibilité : le plan image est considéré infini et l'objet est considéré visible s'il est devant la caméra. Les résultats se classent en 3 catégories :

- 636 cas convergent directement vers la pose désirée en évitant les butées.
- Dans 54 cas, un minimum local est atteint. L'algorithme proposé permet d'en sortir puis de converger vers la pose désirée.
- Les 310 cas restants correspondent également à des configurations où un minimum local est atteint. Toutefois, les tentatives de sortie de minimum font passer l'objet derrière la caméra. Ces cas correspondent à des situations de reconfiguration du robot. Depuis le minimum local, la caméra doit réaliser un grand mouvement de rotation et perd ainsi l'objet observé. N'ayant pas d'algorithme de planification à notre disposition, il ne nous a pas été possible de dire si l'ensemble de ces cas n'ont pas de solution ou si, dans certains cas, une trajectoire respectant la contrainte existe.

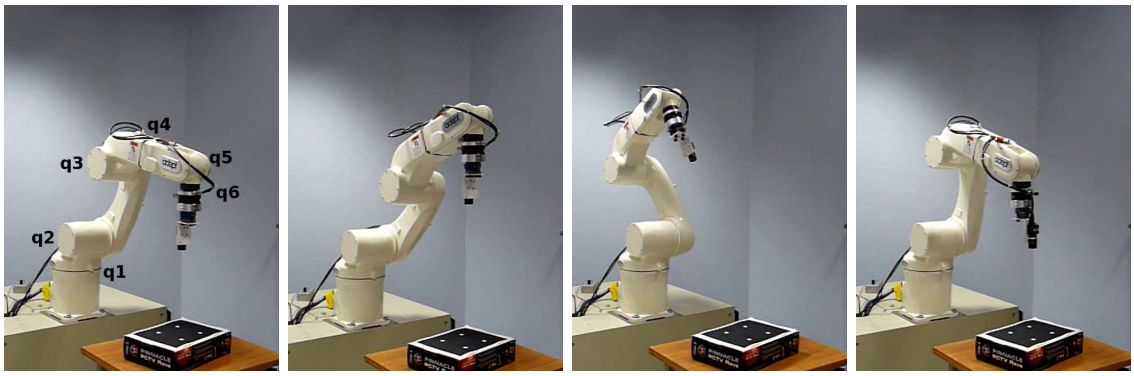
Contrairement aux simulations exhaustives proposées en asservissement visuel hybride (Section 4.2.2) qui pouvaient explorer un large espace, ces résultats ne sont valides que pour le couple $({}^{c}\mathbf{M}_{o}^{*}, \mathbf{q}^{*})$ défini préalablement. Ils ne sont également valides que pour le bras utilisé, l'évitement de butées pouvant avoir un comportement très différent d'un robot à un autre.

Les cas de reconfiguration du robot touchent aux limites de l'approche réactive proposée. Une phase de planification permettrait de gérer la perte de visibilité en retrouvant l'objet une fois le robot reconfiguré. La problématique de sortie d'un minimum local relève en robotique de la planification de trajectoire, il serait intéressant d'explorer plus en profondeur cette ébauche de stratégie réactive.

Nous étudions maintenant le comportement de l'évitement de butées pour une rotation pure.

4.3.3 Rotation pure sur robot à 6 axes

L'expérience est réalisée sur un robot à 6 axes Viper850 (voir Figure 4.31). Les paramètres pour l'évitement sont $\rho = 20\%$ et $\rho^{a} = 10\%$. Sur le robot réel, la position désirée ne correspond pas nécessairement à la zone de confiance de toutes les butées. Nous utilisons le paramètre de progression exposé en Section 3.2.4.2. Les poids des articulations sont donc artificiellement annulés au fur et à mesure que la tâche visuelle converge. Nous utilisons également la stratégie de sortie d'un minimum local exposée en Section 3.2.4.1.



(a) Position initiale (b) Position en butée (c) Position en évitement (d) Position finale

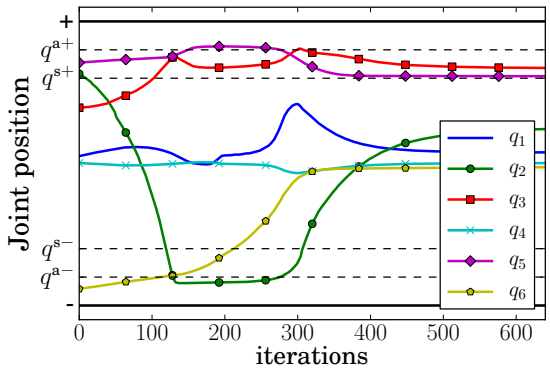
FIG. 4.31 – Robot Viper en position initiale (a), en butée sur l'axe 5 (b), en évitement de butées (c) puis en position finale (d).

Sans évitement de butées, la caméra réalise une translation en arrière le long de l'axe optique (astucieusement symbolisé par l'angle du mur). L'axe 5 arrive rapidement en butée articulaire (Figure 4.31b). Avec le schéma d'évitement de butées (Figure 4.31c), le robot continue de s'éloigner de l'objet mais en s'orientant afin de réaliser au mieux la tâche visuelle. La position finale (Figure 4.31d) diffère de la position initiale par une rotation de 110°.

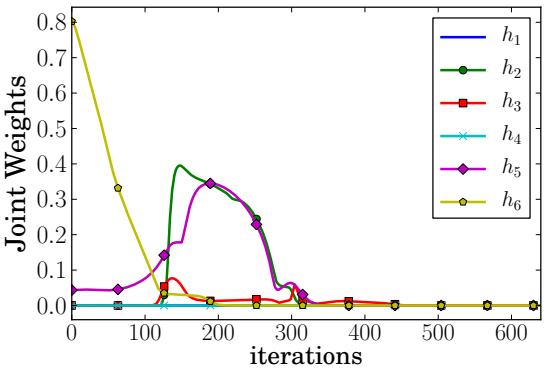
Le comportement d'évitement est représenté sur la Figure 4.32. En position initiale, l'axe 6 (jaune) est très proche d'une butée. Étant hors de la zone de pondération optimale $\mathcal{I}^a$, la pondération générique est utilisée. Le poids chute rapidement jusqu'à l'itération 120, quand la butée passe en zone de pondération optimale. Le poids alors associé est minimal, toutefois on note que cette articulation est activée jusqu'à ce qu'elle se trouve en zone de confiance vers l'itération 200 : l'évitement est donc actif jusqu'au bout même si le poids optimal est très faible.

Comme on pouvait s'y attendre, l'axe 5 (magenta) est rapidement activé. Il sort de la zone d'activation optimale vers l'itération 120. Comme exposé en Section 3.2.3.3, ceci vient du fait que la pondération minimale n'est potentiellement pas suffisante quand plusieurs axes sont activés. La butée supérieure de l'axe 5 est toutefois évitée avec un poids très faible ($h_5 = 0.3$ à l'itération 200). Le très large mouvement de l'axe 2 (vert) se retrouve en comparant la Figure 4.31a et la Figure 4.31c. L'axe est rapidement activé et encore une fois un poids très faible suffit à éviter la butée : $h_2 = 0.4$ à l'itération 150. Notons que la pose désirée correspond à la zone d'activation des axes 3 et 5. Le paramètre de progression (Section 3.2.4.2) étant utilisé, les poids associés à ces articulations sont annulés et le robot peut converger vers la pose désirée. Les valeurs des poids sont très faibles pendant tout l'asservissement, cela suffit pourtant à éviter plusieurs butées à la fois.

La pondération optimale ne ressort pas beaucoup sur cette expérience. Seul l'axe 5 (mauve), entre les itérations 350 et 400, est en zone d'activation avec un poids nul (évitement passif). Rappelons que la tâche visuelle induit une trajectoire 3D particulièrement singulière. Elle est peu propice à éloigner les articulations de leurs butées.

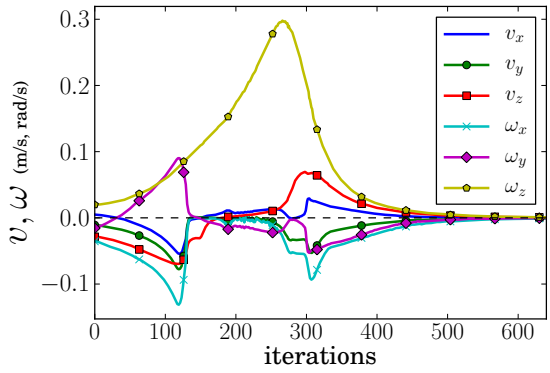


(a) Positions articulaires

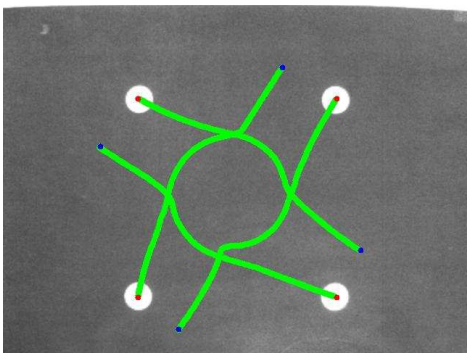


(b) Pondération des articulations

FIG. 4.32 – Positions articulaire (a) et pondération associée (b). L'axe 6 est proche d'une butée en position initiale, il s'en éloigne rapidement. La pondération de l'axe 5 permet d'éviter la butée supérieure. La butée de l'axe 2 est également évitée, puis le robot converge vers la position finale.



(a) Vitesse de la caméra



(b) Trajectoires dans l'image

FIG. 4.33 – Comportement de l'asservissement visuel. La caméra réalise une grande rotation autour de l'axe optique (a). Trois phases sont visibles pour les trajectoires des points (b) : une partie linéaire, une courbe associée à la rotation de la caméra, puis une convergence en ligne droite.

Les vitesses de la caméra sont représentées en Figure 4.33a. La rotation autour de l'axe optique (jaune) est prépondérante. La translation en arrière (rouge) est stoppée vers l'itération 120, quand l'axe 5 se rapproche de la butée. Entre les itérations 120 et 350, le couplage entre la tâche visuelle et l'évitement de butée entraîne un mouvement complexe. À partir de l'itération 400 les vitesses décroissent de façon découplée. Seule la tâche visuelle est active à ce moment-là. La décomposition en trois phases est clairement visible sur les trajectoires des points en Figure 4.33b. Les mouvements découplés de la caméra dessinent tout d'abord des lignes dans l'image (itérations 0-100). Entre les itérations 100 et 350, la forte rotation autour de l'axe optique fait que les points suivent un arc de cercle presque parfait. Les dernières itérations correspondent à la seule tâche visuelle : les points suivent

alors les trajectoires rectilignes caractéristiques d'un asservissement visuel sur des points.

Par rapport aux cas à 0, 1 puis 2 butées étudiés en simulation en Section 3.2.3.4, la tâche visuelle est davantage modifiée. Elle se réalise toutefois sans problème particulier. La trajectoire 3D correspondante, représentée en Figure 4.34, confirme que l'approche hybride exploite une grande partie de l'espace articulaire pour réaliser au mieux la tâche visuelle. La position la plus éloignée de la pose initiale correspond à la Figure 4.31c et à l'itération 200.

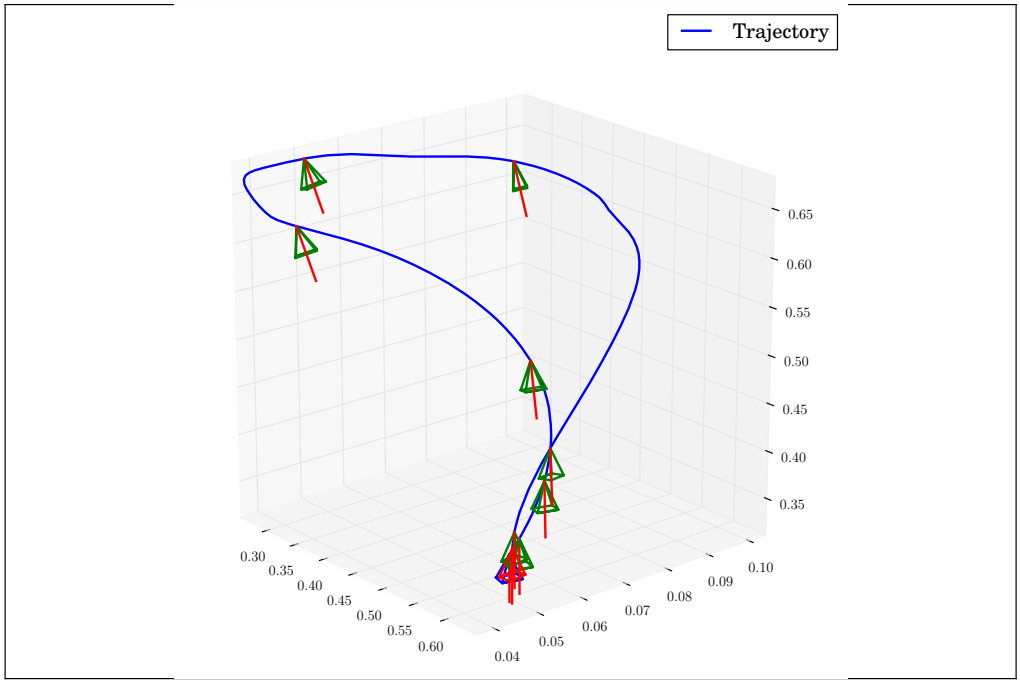


FIG. 4.34 – Trajectoire pour une rotation pure. Pour réaliser au mieux la tâche visuelle, la caméra décrit une grande boucle dans l'espace avant de converger vers la pose désirée.

L'étude de la rotation pure est un cas d'école en asservissement visuel. Notre approche ne se contente pas de limiter le mouvement arrière de la caméra. Le mouvement induit est une large boucle dans l'espace et correspond à une minimisation au mieux de l'erreur visuelle. À ce titre, la pondération générique montre que de très faibles poids suffisent à éviter les butées. La tâche visuelle en est d'autant moins perturbée.

4.3.4 Comparaison avec un gradient projeté

Comme précédemment énoncé, dans [Marey 10c] un opérateur de projection $\mathbf{P}_n$ permet d'éviter les butées tout en considérant une tâche de rang plein. Cet opérateur est construit sur le noyau non pas de la tâche référencée capteur, mais sur le noyau de sa norme. La tâche secondaire dispose donc alors de 5 degrés de liberté, mais la tâche visuelle peut être perturbée même si sa norme ne l'est pas. De façon similaire à la pondération adaptative proposée en Section 3.2.3, un gain adaptatif permet d'éviter au mieux une butée en perturbant le moins possible la tâche principale. Ce gain ne prend toutefois pas en compte

les cas où plusieurs butées sont à éviter en même temps. Comme montré en Section 3.2.3, le couplage entre les différents évitements rend en effet difficile l'expression analytique d'un évitement optimal pour un nombre indéfini d'articulations. Dans [Marey 10c], la solution apportée à ce problème est d'augmenter artificiellement le gain optimal noté λ_{sec}. Ce gain est multiplié par $(1 + \lambda_i)$ où $\lambda_i \geq 0$ est défini tel que :

- $\lambda_i = 0$ correspond au gain adaptatif exact. La tâche visuelle est perturbée le moins possible, mais l'évitement n'est garanti que si une seule articulation est activée à la fois.
- $\lambda_i > 0$ dégrade la tâche visuelle mais augmente la probabilité d'éviter plusieurs butées à la fois.

En pratique λ_i est relativement difficile à régler sans quelques essais. Pour la configuration considérée, la valeur minimale a été déterminée à 2.5.
Au fur et à mesure que la tâche visuelle progresse, l'opérateur $\mathbf{P}_n$ évolue vers l'opérateur de projection classique $\mathbf{P}$. Dans le cas d'une tâche de rang plein, $\mathbf{P}$ est une matrice nulle et seule la tâche visuelle est appliquée. On retrouve un comportement similaire dans notre approche quand la matrice de pondération $\mathbf{H}_q$ de l'évitement est artificiellement annulée (configuration Z1 à convergence exposée en Section 3.2.4.2). La raison commune est que les deux méthodes nécessitent de perturber la tâche visuelle car elle est de rang plein. Au voisinage de la convergence, la tâche visuelle ne doit en revanche pas être perturbée si l'on souhaite qu'elle atteigne son minimum global. Le schéma d'évitement doit donc être progressivement désactivé.

Pour les expérimentations qui suivent, les deux schémas partagent les mêmes intervalles d'activation, paramétrés cette fois par $\rho = 30\%$ et $\rho^{\mathrm{a}} = 15\%$. Les poses initiale et désirée sont définies par :

$$\begin{cases} {}^{c}\mathbf{M}_o = (0, 0, 0.6, 0.36, 0.54, -2.05) \text{ (m,rad)} \\ {}^{c}\mathbf{M}_o^* = (0, 0, 0.3, 0, 0, 0) \text{ (m,rad)} \end{cases} \tag{4.14}$$

Une large rotation autour de l'axe optique est nécessaire pour passer de la pose initiale à la pose désirée.

Les trajectoires 3D correspondant aux deux approches sont représentées en Figure 4.35. Le projecteur conduit à une trajectoire beaucoup plus courte que celle de notre approche. On retrouve en effet la large boucle déjà constatée pour la rotation pure. Cette observation est également faite sur la Figure 4.36 qui indique la position relative des articulations. Comme pour la rotation pure, l'axe 2 (vert) passe d'un extrême à l'autre avec notre approche. Par GPM, le même axe évolue dans la direction opposée.

Deux cas d'évitement passif apparaissent nettement sur la Figure 4.36c. À partir de l'itération 300 le poids de l'axe 3 (rouge) est nul alors qu'il est toujours dans la zone d'activation. Dans ce cas, c'est donc la tâche visuelle qui le ramène vers la zone de confiance. De la même façon, l'axe 5 (magenta) n'est jamais dans la zone de confiance, pourtant son poids est nul à partir de l'itération 400.

L'annulation progressive des poids se fait à partir de l'itération 600 : le poids de l'axe 3 est annulé alors que l'axe est hors de la zone d'activation, et se rapproche en fait de sa butée. Cela permet d'atteindre la position désirée alors qu'elle est hors de la zone de confiance.

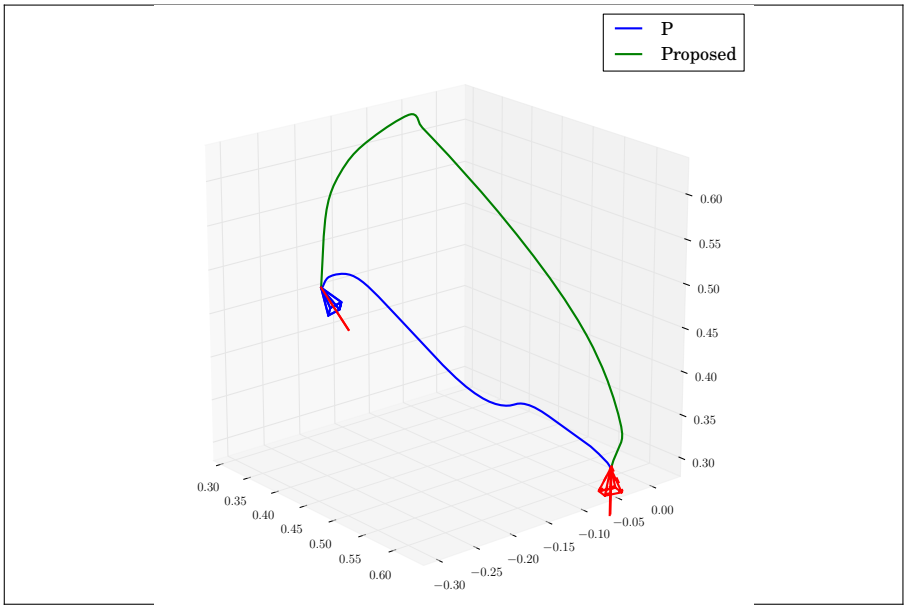
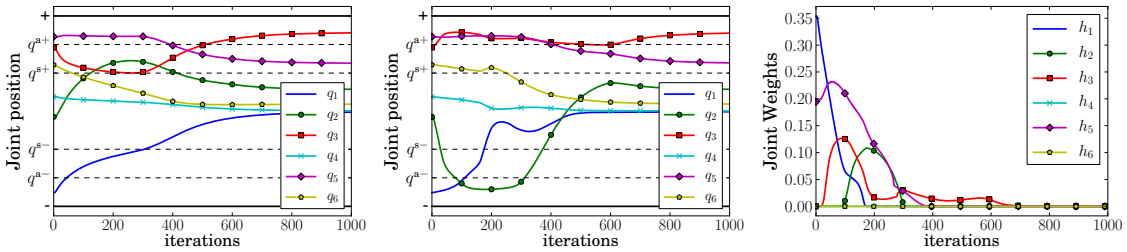


FIG. 4.35 – Trajectoires dans l'espace pour l'approche par GPM (bleu) et hybride (vert). La trajectoire est beaucoup plus rectiligne avec le projecteur.

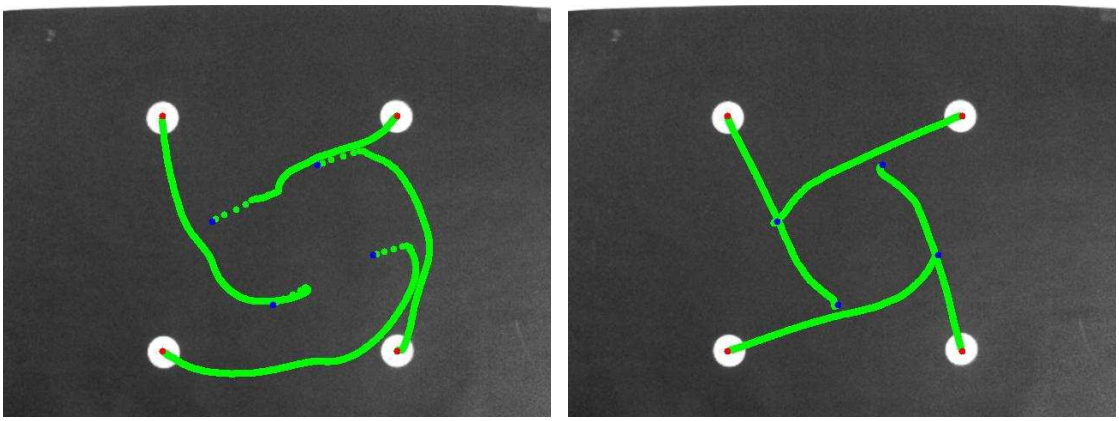


(a) Positions articulaires (GPM) (b) Positions articulaires (c) Pondération des axes

FIG. 4.36 – Positions articulaires pour l'évitement de butées par GPM (a) et avec la commande hybride (b). L'axe 2 a un comportement très différent : la commande hybride s'éloigne beaucoup plus de l'objet que la commande par GPM.

Les trajectoires des points dans l'image sont représentées sur la Figure 4.37. La méthode GPM montre une grande vitesse dans l'image en début d'asservissement : on distingue en effet des points successifs à partir de la position initiale. Cela vient probablement de la proximité des butées dans la pose initiale. Comme précédemment exposé, il a été nécessaire de dégrader l'évitement optimal afin de permettre d'éviter plusieurs butées à la fois. La tâche visuelle s'en trouve impactée. Les trajectoires sont très rectilignes pour l'évitement par commande hybride. Exploiter davantage d'espace permet de favoriser la tâche visuelle.

En conclusion de ce comparatif, l'approche GPM, en mode dégradé pour tolérer les contraintes multiples, semble favoriser l'espace articulaire à l'espace des informations visuelles. La commande hybride quant à elle permet une convergence très satisfaisante dans l'image, au détriment d'un large mouvement en 3D. On rappelle que les informations



(a) Trajectoires dans l'image (GPM) (b) Trajectoires dans l'image (Hybride)

FIG. 4.37 – Trajectoires dans l'image pour l'évitement de butées par GPM (a) et avec la commande hybride (b). Par GPM, la vitesse dans l'image est très rapide au début (points isolés). Les trajectoires sont moins rectilignes que par approche hybride.

visuelles utilisées ne sont pas du tout optimales pour la trajectoire 3D. L'objectif de l'évitement de butées n'est pas d'optimiser le comportement dans l'espace mais bien de réaliser la tâche référencée capteurs tout en respectant les contraintes, ce que réalise notre approche.

4.3.5 Conclusion

Dans cette section, notre approche a été appliquée à la contrainte classique de l'évitement des butées articulaires. Le schéma hybride permet de réaliser au mieux la tâche référencée capteur en respectant cette contrainte. La pondération optimale a été mise en œuvre afin de n'activer les articulations que si nécessaire. Plusieurs cas d'évitements passifs ont ainsi pu être mis en évidence. Comme en asservissement visuel hybride, on remarque que les poids liés aux contraintes sont généralement très faibles.

Des simulations pseudo-exhaustives indiquent que le schéma est performant dans un grand nombre de configurations. La sortie d'un minimum local est toutefois souvent mise en échec par la contrainte de visibilité. Pour y remédier, l'utilisation de caméras omnidirectionnelles pourrait permettre un plus large domaine de convergence. De façon générale, la sortie de minimum local par un schéma purement réactif reste une problématique en soit pour la commande référencée capteurs.

Le cas de la rotation pure permet d'illustrer le principe du schéma proposé. La tâche visuelle est modifiée le moins possible, mais les mouvements dans l'espace peuvent être très grands si la tâche référencée capteurs l'exige.

Notre approche a été comparée avec une autre méthode permettant l'évitement malgré une tâche de rang plein. Quand plusieurs butées sont à éviter en même temps, l'évitement optimal est difficile à déterminer et la tâche visuelle est donc perturbée. Elle est cependant moins perturbée par la commande hybride qu'avec le projecteur considéré.

Ainsi, dans la commande hybride l'évitement de butées se modélise comme une

tâche multi-capteurs classique. La pondération générique de chaque information suffit à réaliser des tâches complexes en respectant des contraintes. La prochaine section regroupe l'ensemble des applications exposées.

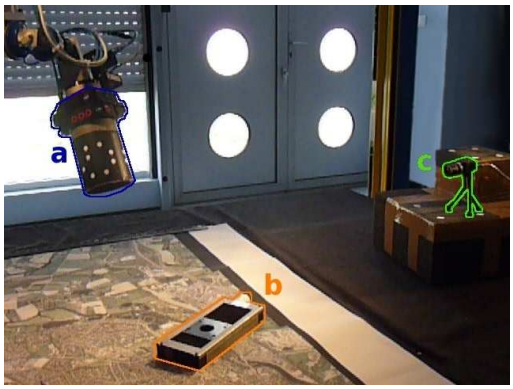
4.4 Combinaison de contraintes

Un point fort de l'approche proposée est sa versatilité. Nous avons vu que l'état de l'art apportait des réponses parfois différentes pour des problématiques voisines. La commande hybride permet au contraire de modéliser toutes les contraintes de la même façon, comme cela a été montré pour la visibilité et l'évitement de butées. La pondération générique proposée offre ainsi une grande souplesse. Si elle permet aux poids d'atteindre des valeurs arbitrairement élevées, nous avons vu en pratique que des cas difficiles étaient résolus avec une pondération relativement faible.

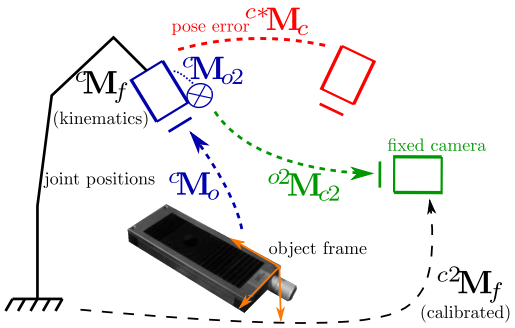
Dans cette section, nous combinons au sein d'une même expérience les applications déjà exposées. Nous présentons tout d'abord le système expérimental. Les résultats sont ensuite présentés pour plusieurs combinaisons de contraintes.

4.4.1 Dispositif expérimental

Les expériences sont réalisées sur le robot cartésien Afma6. Le dispositif général est représenté en Figure 4.38. L'objectif est un positionnement par rapport à un objet fixe. Une caméra embarquée réalise un asservissement visuel 3D. Une caméra déportée observe une mire liée à l'effecteur. Plusieurs contraintes sont à respecter : la visibilité dans les images embarquée et déportée, et l'évitement des butées articulaires.



(a) Disposition générale. Caméra et mire embarquées (a), objet fixe (b) et caméra déportée (c).



(b) Repères utilisés.

FIG. 4.38 – Dispositif expérimental. La mire 3D permet de à la caméra déportée de suivre l'effecteur dans de nombreuses configurations.

La caméra embarquée observe un objet dont le modèle 3D est connu. La pose $^{c}\mathbf{M}_o$ correspondante est estimée en temps réel et on note $\mathbf{s}_{3d} = (^{c*}\mathbf{t}_c, \theta\mathbf{u})$ les informations visuelles associées. Comme en asservissement visuel hybride, les nœuds du modèle 3D sont utilisés

pour la contrainte de visibilité. Les informations visuelles 2D associées sont notées $\mathbf{s}_{2d}$.

La caméra déportée observe une mire composée de 30 points disposés sur un cylindre de repère $\mathcal{F}_{o2}$, rigidement lié à l'effecteur. Les informations visuelles 2D associées sont notées $\mathbf{s}_{\text{ext}}$. Cette mire provient des travaux de [Marchand 02] pour un asservissement PBVS déporté. La disposition des points sur le cylindre fait qu'à tout instant il est possible de déterminer la pose ${}^{o2}\mathbf{M}_{c2}$ entre la caméra déportée et la mire mobile. Dans notre cas, cette pose n'intervient pas dans la commande [1] mais sert uniquement pour le suivi des points de la mire. On peut ainsi prévoir l'apparition ou la disparition de points lors des rotations. On souhaite également assurer la contrainte de visibilité sur les points observés.

La tâche est donc définie par :

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{3d} \\ \mathbf{s}_{2d} \\ \mathbf{s}_{\text{ext}} \\ \mathbf{q} \end{bmatrix} \quad \begin{array}{l} \cdots \text{ PBVS (dim. 6)} \\ \cdots \text{ Visibilité embarquée (dim. } 2\times 28) \\ \cdots \text{ Visibilité déportée (dim. } 2\times 30) \\ \cdots \text{ Articulations (dim. 6)} \end{array} \tag{4.15}$$

La pondération correspondante est :

$$\mathbf{H} = \text{Diag}\left(\mathbf{H}_{3d}, \mathbf{H}_{2d}, \mathbf{H}_{\text{ext}}, \mathbf{H}_{\mathbf{q}}\right) \tag{4.16}$$

où $\mathbf{H}_{3d} = \mathbb{I}$, $\mathbf{H}_{2d}$ et $\mathbf{H}_{\text{ext}}$ sont définies pour la contrainte de visibilité paramétrée par $\rho = 5\%$, et où $\mathbf{H}_{\mathbf{q}}$ est définie pour l'évitement de butées avec la zone de pondération optimale définie par $\rho = 10\%$ et $\rho^{\text{a}} = 5\%$.

Calibration

Les caméras sont calibrées. La géométrie du système quant à elle est grossièrement calibrée. La matrice de passage ${}^{c}\mathbf{M}_{o2}$ constante entre la caméra et la mire embarquées est estimée à la main. La matrice de passage ${}^{c2}\mathbf{M}_{f}$ constante entre la caméra déportée et le repère de base du robot est calibrée hors-ligne en exploitant la relation suivante :

$${}^{c2}\mathbf{M}_{f} = {}^{c2}\mathbf{M}_{o2}\,{}^{c}\mathbf{M}_{o2}^{-1}\,{}^{c}\mathbf{M}_{f} \tag{4.17}$$

où ${}^{c2}\mathbf{M}_{o2}$ est le résultat du calcul de pose entre la caméra déportée et la mire 3D, ${}^{c}\mathbf{M}_{o2}$ est l'estimation grossière précédemment évoquée et ${}^{c}\mathbf{M}_{f}$ est obtenue par le modèle géométrique du robot.

Informations visuelles désirées dans l'image déportée

Contrairement à la coopération multi-caméras, les informations visuelles désirées $\mathbf{s}^{*}_{\text{ext}}$ de l'image déportée ne sont pas déterminées par apprentissage. Elles sont calculées par projection de la mire 3D dans l'image déportée, à partir de sa pose désirée ${}^{c2}\mathbf{M}^{*}_{o2}$. Cette matrice de passage, constante, vérifie à chaque instant t la relation suivante :

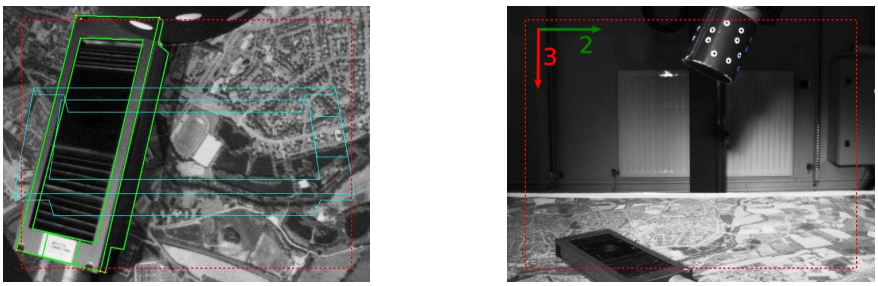
$${}^{c2}\mathbf{M}^{*}_{o2} = {}^{c2}\mathbf{M}_{o2}(t)\,{}^{c}\mathbf{M}_{o2}^{-1}\,{}^{c}\mathbf{M}_{o}(t)\,{}^{c*}\mathbf{M}_{o}^{-1}\,{}^{c}\mathbf{M}_{o2} \tag{4.18}$$

où ${}^{c2}\mathbf{M}_{o2}$ et ${}^{c}\mathbf{M}_{o}$ proviennent des estimations de pose, ${}^{c*}\mathbf{M}_{o}$ est la pose désirée de la caméra embarquée et ${}^{c}\mathbf{M}_{o2}$ est calibrée à la main. Par cette méthode, les informations visuelles désirées $\mathbf{s}^{*}_{\text{ext}}$ de l'image déportée ne correspondent que grossièrement à la pose désirée ${}^{c}\mathbf{M}^{*}_{o}$. Nous verrons que cela n'empêche la convergence.

[1] L'utiliser dans la commande relèverait de la fusion classique pour reconstruire l'état du système.

4.4.2 Application

Les poses initiales et désirées sont définies telles que l'objet soit grand dans l'image embarquée. Les images initiales sont représentées en Figure 4.39. Les axes de translation 2 et 3 du robot sont représentés sur l'image déportée. L'axe 2 correspond ainsi à un mouvement horizontal dans l'image, et l'axe 3 à un mouvement vertical.



(a) Image embarquée (b) Image déportée

FIG. 4.39 – Images initiales. L'objet est grand dans l'image embarquée (a) et la mire 3D est proche du haut de l'image déportée (b).

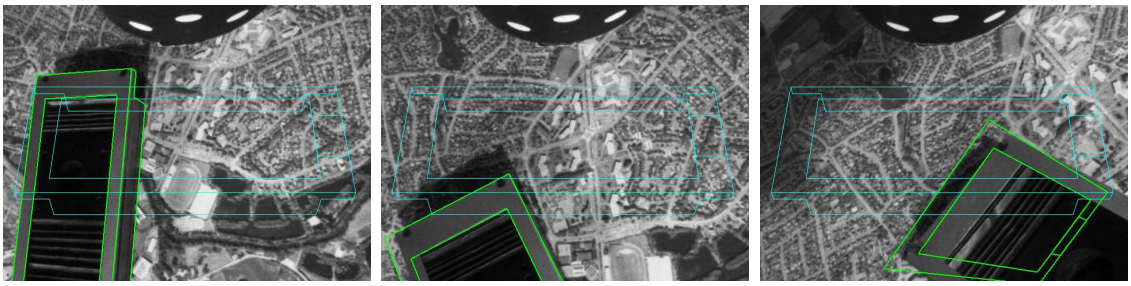
La caméra doit donc s'éloigner de l'objet pour respecter la contrainte de visibilité. Ce mouvement en arrière entre en conflit avec les contraintes de visibilité dans l'image déportée, mais aussi avec une butée articulaire[2]. Comme nous le verrons, la tâche globale ne peut être réalisée que dans un couloir très limité.

Nous étudions maintenant le comportement du système à mesure que les contraintes sont rajoutées. Dans un premier temps, seules les informations visuelles 3D $\mathbf{s}_{3d}$ sont utilisées. Les contraintes imposées successivement sont ensuite la visibilité d'abord embarquée puis déportée, puis les butées articulaires. Les schémas intermédiaires ne sont pas étudiés dans le détail mais permettent de mettre en évidence le respect des contraintes activées et le non-respect éventuel de contraintes non prises en compte.

4.4.2.1 PBVS pur (cas 0)

Un asservissement visuel 3D est d'abord réalisé : $\mathbf{s} = \mathbf{s}_{3d}$. On constate sur la Figure 4.40 que l'objet observé sort rapidement de l'image. Soulignons la performance de l'algorithme de suivi qui continue à estimer la pose avec une précision honnête alors que les deux tiers de l'objet sont déjà sortis de l'image (Figure 4.40b). Ce n'est toutefois pas suffisant, et le suivi ne parvient pas à se recaler sur l'objet quand il réintègre l'image (Figure 4.40c).

[2]La butée rencontrée n'est pas une butée réelle du robot mais une butée définie virtuellement.



(a) Image embarquée (iter. 280) (b) Image embarquée (iter. 640) (c) Image embarquée finale

FIG. 4.40 – Schéma PBVS. L'objet observé sort rapidement de l'image et le suivi visuel ne fonctionne plus.

4.4.2.2 Ajout de la contrainte de visibilité (cas 1)

La contrainte de visibilité est rajoutée par rapport au schéma précédent : $\mathbf{s} = (\mathbf{s}_{3d}, \mathbf{s}_{2d})$. L'asservissement visuel hybride respecte facilement la contrainte de visibilité, comme étudié en Section 4.2. On constate sur la Figure 4.41c que la pose désirée est atteinte avec un comportement dans l'image caractéristique du schéma proposé.

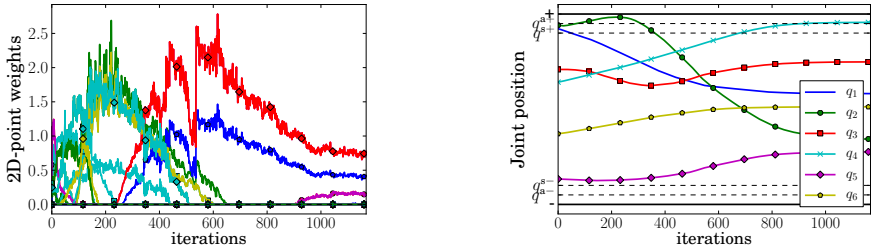
La Figure 4.41a représente l'image embarquée au moment où l'objet est le plus grand dans l'image. On constate en effet qu'il approche à la fois les bords supérieur et inférieur, pour des nœuds diagonalement opposés. C'est dans cette configuration que la caméra est le plus loin de l'objet. On constate sur la Figure 4.41b que cette position fait partiellement sortir la mire 3D de l'image déportée. Le mouvement réalisé pour s'éloigner de l'objet est ainsi un mouvement vers le haut, non compatible avec la visibilité dans l'image déportée.

(a) Image embarquée (iter. 240) (b) Image déportée (iter. 240) (c) Image embarquée finale

FIG. 4.41 – Asservissement visuel hybride. La caméra s'éloigne pour pouvoir observer tout l'objet (a). La position correspondante dans l'image embarquée fait que la mire sort partiellement (b). L'asservissement converge en respectant la contrainte de visibilité (c).

Sur la Figure 4.42a la pondération de la contrainte de visibilité est représentée. Elle correspond aux résultats déjà exposés, avec un poids faible ($h < 3$) qui permet de respecter la contrainte. On constate plusieurs poids supérieurs à 2 vers l'itération 240, qui correspond aux images présentées. La contrainte est maximale à l'itération 600, mais pour une seule information visuelle. Les articulations (Figure 4.42b) ne sortent pas de leurs limites alors

qu'aucun évitement de butée n'est réalisé. La butée virtuelle de l'axe 2 (vert) est définie lors de cette expérience. Elle correspond à une butée tout juste évitée à l'itération 240, quand la contrainte de visibilité est assurée. La valeur de l'articulation 3 (rouge) est à ce moment-là à son minimum, indiquant une position élevée dans l'espace.



(a) Pondération de la contrainte de visibilité $\mathbf{H}_{2d}$

(b) Positions articulaires

FIG. 4.42 – Cas 1. Une pondération faible permet de respecter la contrainte de visibilité (a). Les articulations restent dans l'intervalle autorisé mais l'axe 2 (vert) s'approche de la butée supérieure (b).

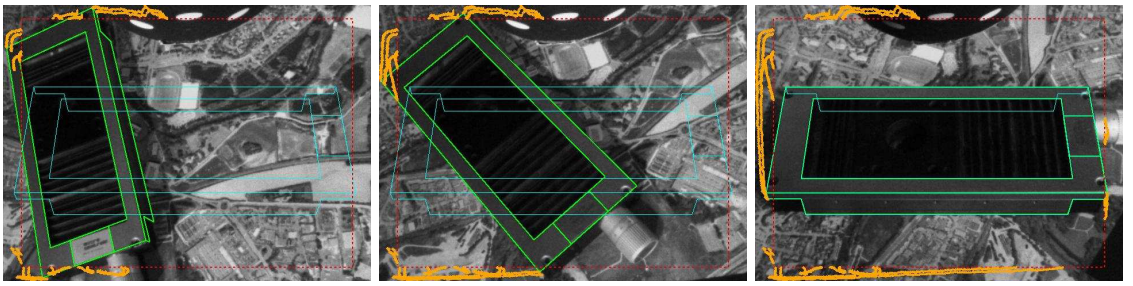
4.4.2.3 Ajout de la contrainte de visibilité déportée (cas 2)

Cette expérience prend en compte la contrainte de visibilité dans l'image déportée : $\mathbf{s} = (\mathbf{s}_{3d}, \mathbf{s}_{2d}, \mathbf{s}_{\text{ext}})$. Les images-clés sont représentées en Figure 4.43.

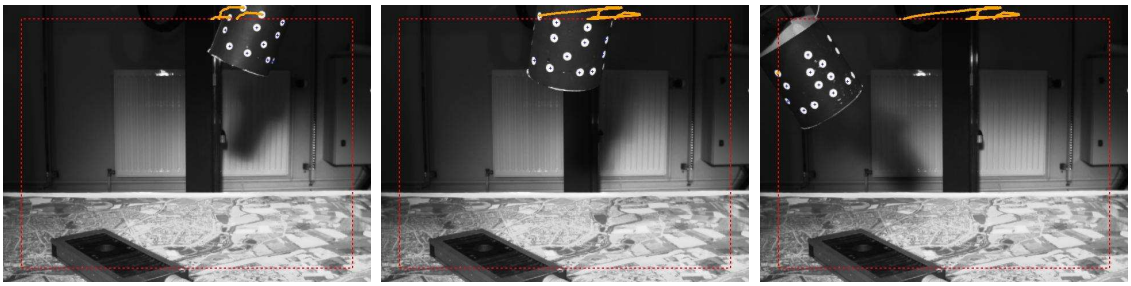
À l'itération 240 cette fois, la mire est bien visible dans l'image déportée. Certains points sont alors proches du bord dans les deux images. Par rapport à l'expérience précédente, l'objet observé est plus incliné à ce moment-là (Figure 4.41a contre Figure 4.43a). Ne pouvant s'éloigner vers le haut pour respecter la contrainte de visibilité, la caméra s'éloigne vers le côté et n'observe plus l'objet sous le même angle. Comme indiqué sur la Figure 4.44c, l'axe 2 (vert) dépasse à ce moment sa butée supérieure. Par rapport à la configuration précédente, l'axe 3 (rouge) n'atteint pas de valeurs aussi faibles que précédemment.

Comme pour le cas 1, ce n'est pas au moment où le maximum de contraintes sont activées que les poids sont le plus élevés. En effet, à l'itération 240 plusieurs poids de la visibilité embarquée dépassent la valeur de 10 (Figure 4.44a) et un poids atteint 5 pour la visibilité déportée (Figure 4.44b). C'est lors de la rotation autour de l'axe optique, à l'itération 600 que le poids maximal est atteint avec la valeur de 25 pour l'abscisse d'un point approchant le bord gauche de l'image sur la Figure 4.43b. Cette configuration correspond à des contraintes largement respectées dans l'image déportée (Figure 4.43e).

L'asservissement converge ensuite vers la pose finale. Soulignons que cette pose correspond à une contrainte toujours activée dans l'image déportée (point le plus à gauche sur la Figure 4.43f). La condition Z1 exposée en Section 3.2.4.2 n'est donc pas réalisée ($\mathbf{H}_{\text{ext}} \neq 0$). Les valeurs désirées $\mathbf{s}^*_{\text{ext}}$ ne correspondant qu'approximativement à la pose désirée, la condition Z2 n'est pas non plus réalisée ($\mathbf{e}_{\text{ext}} \neq 0$). La pose finale est donc en réalité un compromis entre la pose désirée et l'infime pondération restant pour le schéma déporté. Cet écart s'exprime par l'inégalité entre le minimum global de la tâche contrainte,

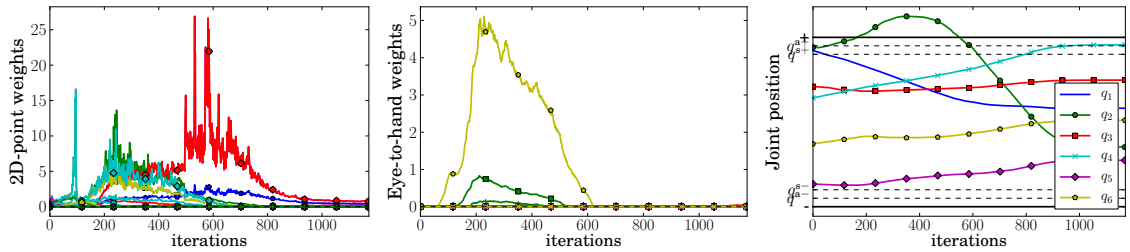


(a) Image embarquée (iter. 240) (b) Image embarquée (iter. 600) (c) Image embarquée finale



(d) Image déportée (iter. 240) (e) Image déportée (iter. 600) (f) Image déportée finale

FIG. 4.43 – Cas 2. Pour s'éloigner de l'objet la caméra se dirige vers la droite de l'image déportée (a,d). La rotation autour de l'axe optique (b) est effectuée en se rapprochant de la caméra déportée (e). Le schéma converge vers la pose finale (c) qui laisse une contrainte activée dans l'image déportée (f).



(a) Pondération de la contrainte de visibilité embarquée $\mathbf{H}_{2d}$ (b) Pondération de la contrainte de visibilité déportée $\mathbf{H}_{\text{ext}}$ (c) Positions articulaires

FIG. 4.44 – Cas 2. Les contraintes de visibilité embarquée (a) et déportée (b) sont concurrentes vers l'itération 240. Par la suite seule la visibilité embarquée est activée.

et celui de la seule tâche de positionnement :

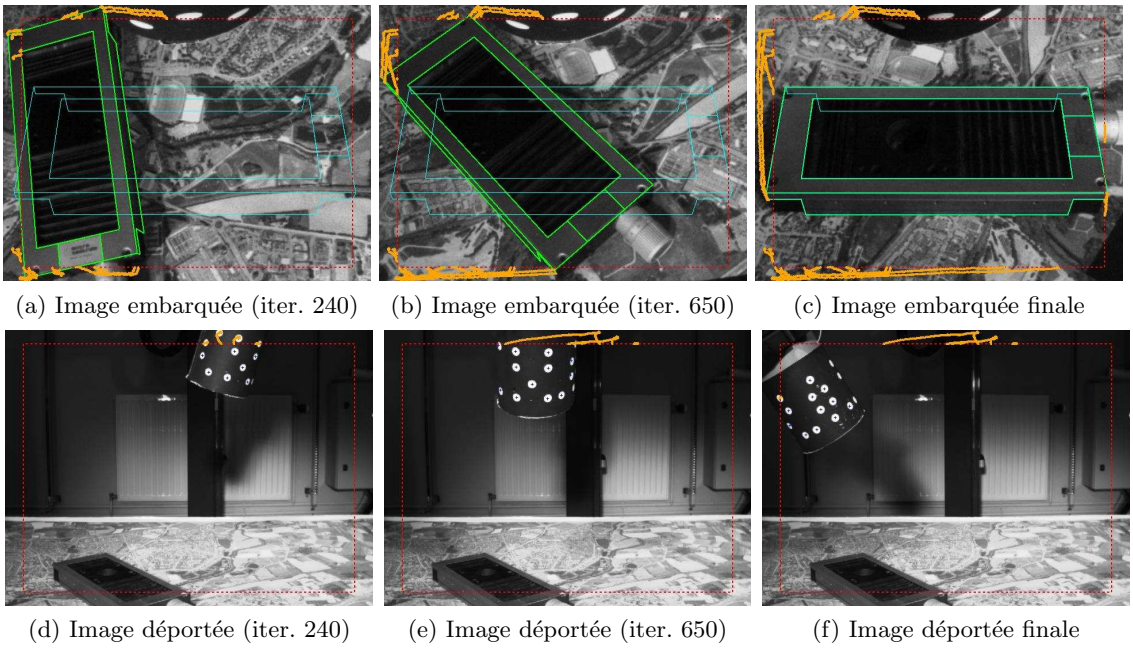
$$\mathbf{q}^* = \arg\min_{\mathbf{q}} \left(\mathbf{e}_{3d}^\top \mathbf{e}_{3d} + h_i^2 e_i \right) \neq \arg\min_{\mathbf{q}} \left(\mathbf{e}_{3d}^\top \mathbf{e}_{3d} \right) \tag{4.19}$$

où e_i est l'erreur visuelle associée au point activé dans l'image déportée, avec un poids $h_i = 0.07$. Nous illustrons ainsi un schéma où le compromis est préféré au fait de rejoindre la pose exacte désirée. En pratique, on constate que la pose est extrêmement proche, l'erreur étant inférieure à 1mm en translation et à 10^{-3} radians en rotation.

Nous ajoutons maintenant la dernière contrainte.

4.4.2.4 Ajout de l'évitement de butées (cas 3)

Pour cette dernière expérience, toutes les contraintes sont utilisées : $\mathbf{s} = (\mathbf{s}_{3d}, \mathbf{s}_{2d}, \mathbf{s}_{\text{ext}}, \mathbf{q})$. Le comportement dans l'image est représenté en Figure 4.45.



(a) Image embarquée (iter. 240) (b) Image embarquée (iter. 650) (c) Image embarquée finale

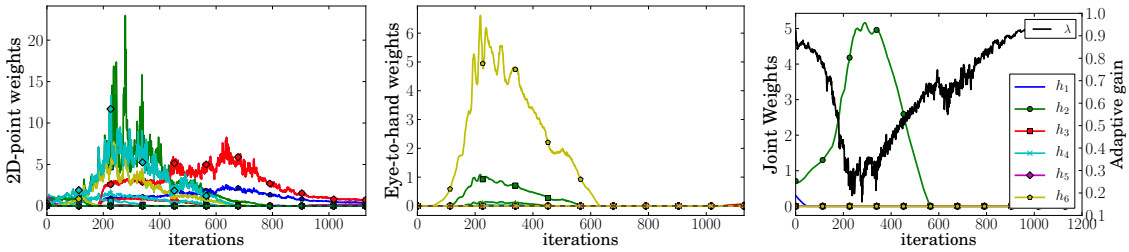
(d) Image déportée (iter. 240) (e) Image déportée (iter. 650) (f) Image déportée finale

FIG. 4.45 – Cas 3. Le mouvement vers la droite de l'image déportée n'est plus possible au moment où l'objet est grand dans l'image embarquée (a,d). Comme précédemment, la mire se rapproche de la caméra déportée au moment de la rotation autour de l'axe optique (b,e). La même pose est atteinte que dans le cas 2 (c,f).

La principale différence avec le cas précédent est visible pour l'itération 240. Avec la limite de butée de l'axe 2, la caméra ne peut se déplacer vers la droite de l'image déportée (Figure 4.45d). On constate sur la Figure 4.46 que les trois contraintes sont alors concurrentes : les poids sont activés pour la visibilité dans les deux images et pour l'évitement de butée de l'axe 2.

Les poids atteignent tous leur valeur maximale pendant l'activation simultanée de toutes les contraintes. Cette valeur dépasse 20 lors d'une itération pour la visibilité embarquée. Elle est du même ordre de grandeur que précédemment. Rappelons que la zone d'activation est paramétrée par $\rho = 5\%$. Pour une image 640x480, un poids de 20 signifie que le point se trouve entre 1 et 2 pixels du bord. Les oscillations dans les poids proviennent donc de l'algorithme d'estimation de pose. La pondération reste relativement faible pour la visibilité dans l'image déportée ($h_{\text{ext}} < 7$). Elle est toutefois plus élevée que dans le cas 2. L'évitement de butée rend donc plus difficile le respect de la contrainte de visibilité. Le poids de l'axe 2 atteint la valeur de 5. Dans cette configuration, l'effecteur n'a en réalité qu'un très faible couloir disponible pour réaliser la tâche visuelle en respectant l'ensemble des contraintes.

Le gain adaptatif λ défini en (3.72) est représenté sur la Figure 4.46c. Sa valeur est divisée par 10 au moment de respecter toutes les contraintes. Le système est donc fortement

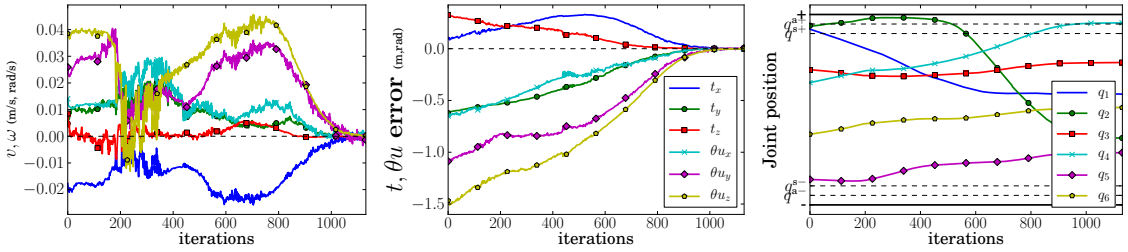


(a) Pondération de la contrainte de visibilité embarquée $\mathbf{H}_{2d}$ (b) Pondération de la contrainte de visibilité déportée $\mathbf{H}_{\text{ext}}$ (c) Pondération de l'évitement de butée $\mathbf{H}_{\mathbf{q}}$ et gain adaptatif λ

FIG. 4.46 – Cas 3. Toutes les contraintes sont activées autour de l'itération 240 : visibilité embarquée (a), déportée (b) et évitement de butées (c). Le gain adaptatif (c) est alors minimal.

ralenti, ce qui empêche aux points 2D de sortir de l'image entre deux itérations.

Comme dans les cas précédents, la visibilité dans l'image embarquée est activée une deuxième fois autour de l'itération 650. La problématique est la même : un coin de l'objet se rapproche du bord gauche de l'image lors de la rotation autour de l'axe optique. La pondération associée dans le cas 3 ne dépasse pas la valeur de 10. Au même moment, la visibilité dans l'image déportée est largement respectée, et aucune butée n'est hors de la zone de confiance (Figure 4.47c). La caméra converge ensuite vers la même pose que précédemment, définie par un compromis entre la pose désirée et l'activation infime ($h = 0.07$) d'une contrainte dans l'image déportée.



(a) Vitesse de la caméra $\mathbf{v}_c$ (b) Erreur du schéma PBVS $\mathbf{e}_{3d}$ (c) Positions articulaires $\mathbf{q}$

FIG. 4.47 – Cas 3. La vitesse de consigne n'est pas lisse au moment de la réalisation de toutes les contraintes (a). L'erreur de position est annulée sans décroissance exponentielle (b). Les oscillations de la vitesse restent peu visibles sur les positions articulaires, qui respectent les butées (c).

Le schéma général est réalisé à faible vitesse comme indiqué en Figure 4.47a. L'effet du gain adaptatif apparaît nettement entre les itérations 200 et 400, où la vitesse générale est plus faible. Des oscillations sont toutefois observées, du fait des poids très variables. La pondération pourrait probablement être lissée pour éviter de tels phénomènes. L'erreur en terme de pose 3D est représentée sur la Figure 4.47b et montre un mouvement lisse de l'effecteur à l'échelle du déplacement global. Les vitesses oscillent quand elles ont les valeurs les plus faibles, ce qui ne transparaît pas sur les positions. Les nombreuses contraintes font

que le schéma PBVS est loin de montrer une décroissance exponentielle de l'erreur. La composante t_x notamment est fortement impactée par la prise en compte des contraintes. Cela correspond au fait que l'objet observé est tout au long de l'asservissement très près du bord gauche de l'image. Le maintenir dans l'image ne peut se faire qu'en perturbant la vitesse de translation associée. La composante θu_y, qui aurait pu jouer un rôle similaire, est moins perturbée par la visibilité.

La Figure 4.47c montre que l'axe 2 évite la butée supérieure. L'évitement de butées articulaires est moins sollicité avec un asservissement visuel 3D qu'avec les points 2D utilisés dans la Section 4.3. Ce schéma a en effet un comportement beaucoup plus satisfaisant dans l'espace et ne génère pas les très larges trajectoires observées dans la section dédiée à l'évitement de butées.

4.4.3 Conclusion

Quatre asservissements ont été réalisés avec des contraintes croissantes. Les trajectoires correspondantes sont comparées sur la Figure 4.48. Le PBVS pur (cyan) montre une trajectoire parfaitement rectiligne, qui correspond comme nous l'avons vu à une perte de la visibilité. L'asservissement doit alors être stoppé. L'asservissement visuel hybride (cas 1 en bleu) assure la visiblité en amenant la caméra au dessus de l'image déportée. La trajectoire correspondante décrit bien une boucle surplombant les autres. En activant la contrainte de visiblité dans l'image déportée (cas 2 en vert), le comportement est comme nous l'avons vu de se décaler sur le côté. Ce phénomène apparaît nettement sur la trajectoire correspondante. Ce mouvement sur le côté est rendu impossible par l'activation de l'évitement de butées (cas 3, rouge). La trajectoire suivie est alors légèrement plus proche de la caméra déportée.

Seul le cas 1 (asservissement visuel hybride) rejoint la pose désirée (caméra rouge en traits pleins). La pose finale des cas 2 et 3 (traits pointillés) est en effet légèrement différente, du fait de la contrainte de visibilité dans l'image déportée.

Le choix de paramétrer l'intervalle de confiance par $\rho = 5\%$ fait que les contraintes peuvent être approchées d'assez près pour réaliser au mieux la tâche de positionnement (toujours pondérée à 1). Une conséquence est que les poids atteignent des valeurs relativement élevées. Toutefois nous avons pu voir que le comportement restait satisfaisant. Augmenter ρ permettrait de prendre plus tôt en compte les contraintes mais dégraderait la tâche référencée capteurs.

Activer plusieurs types de contraintes met également en avant la notion de concurrence entre les capteurs. À l'itération 240, l'ordre de grandeur de la pondération pour la visibilité est de 2.5 pour le cas 1, puis passe à 15 puis 20 pour les cas 2 et 3. De la même façon, le poids maximum de la visibilité déportée est de 5 dans le cas 2 et 6 dans le cas 3. Si de multiples contraintes conduisent à une augmentation générale des poids de $\mathbf{H}$, cela permet de toutes les respecter en l'absence d'un critère de pondération optimale a priori.

Le réglage de la butée de l'axe 2 permet de mettre le robot dans une situation où le couloir permettant de réaliser la tâche en respectant les contraintes est très mince. La régler plus durement rend la tâche infaisable en respectant les contraintes.

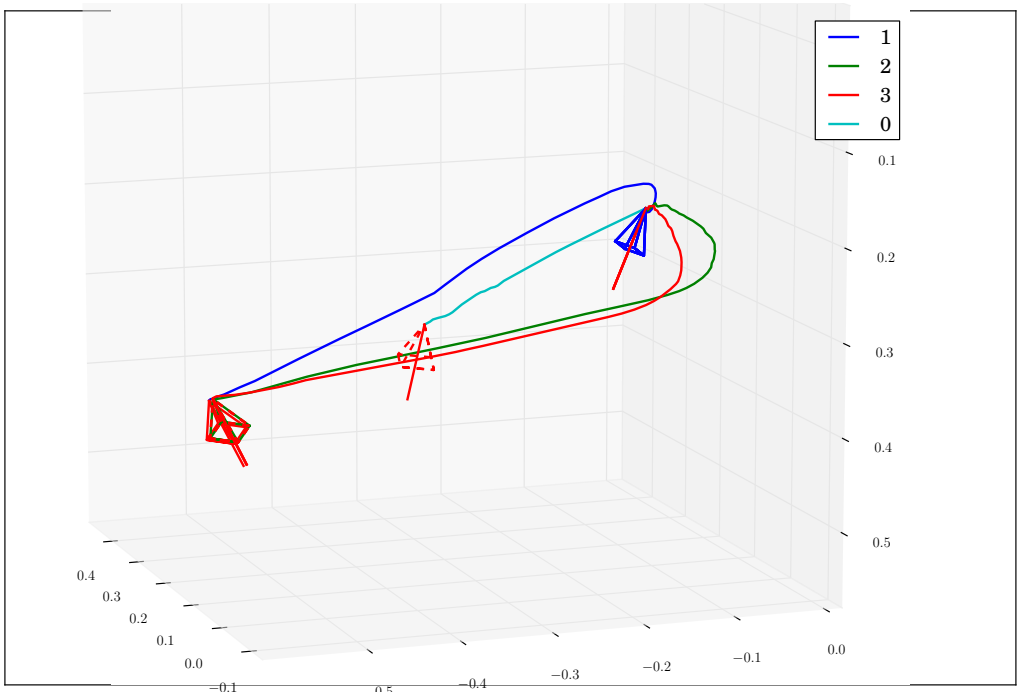


FIG. 4.48 – Trajectoires dans l'espace pour l'ensemble des cas étudiés, vues depuis la caméra déportée. La trajectoire en PBVS (cyan) est parfaitement rectiligne mais la perte du suivi visuel empêche la convergence. La trajectoire du cas 1 (bleu) est celle qui passe le plus haut. Le cas 2 (vert) fait que la caméra se décale sur le côté droit. La trajectoire est plus proche de la caméra déportée quand toutes les contraintes sont activées (rouge).

4.5 Synthèse

Dans ce chapitre nous avons illustré les différents aspects de l'approche proposée à travers plusieurs configurations.

L'utilisation du jacobien étendu rend très versatile la loi de commande utilisée. Les contraintes étant définies dans l'espace de chaque information capteur, la matrice de pondération a une forme générique et est facilement déterminée pour un nouveau capteur.

Pour chaque expérience, des comparaisons ont été effectuées avec d'autres approches, montrant les performances du schéma hybride proposé. L'application en asservissement visuel hybride, à travers les simulations exhaustives, est particulièrement séduisante car la visibilité est un problème récurrent en asservissement visuel 3D. Il trouve ici une solution très adaptée dès qu'un modèle CAO est présent, puisqu'il suffit d'injecter dans la commande les points 2D de l'enveloppe du modèle. Le paramètre ρ s'exprimant dans l'espace de l'image métrique, il s'est révélé plus facile à régler que dans l'espace des erreurs visuelles. La dernière expérience illustre comment le comportement d'un même système évolue à mesure que des contraintes supplémentaires sont activées.

Conclusion générale

Dans ce mémoire nous avons étudié la commande référencée multi-capteurs de façon aussi générique que possible. L'objectif était de proposer un formalisme pour la fusion de données permettant la conception de lois de commande robustes et performantes. Aucune planification n'a été envisagée et nous avons proposé des approches pour pallier aux lacunes de la commande réactive. Dans un premier temps, un formalisme général a été établi pour la calibration de capteurs extéroceptifs à partir de mesures de vitesse. La méthode est adaptée pour la calibration intrinsèque et extrinsèque. Nous avons suite étendu la commande référencée capteur aux tâches à plusieurs capteurs avec des contraintes génériques.

Contributions

Calibration : La méthode proposée pour la calibration repose sur l'expression analytique du modèle cinématique. Elle permet de déterminer les paramètres de calibration mais aussi d'éventuelles grandeurs inconnues, comme les profondeurs des points observés. Reposant sur la mesure de vitesse, elle n'égale pas les méthodes statiques mais permet de réaliser la calibration lors du fonctionnement du robot. L'expérience d'une calibration pendant un asservissement visuel montre qu'un même capteur peut être calibré alors qu'il est utilisé dans la boucle de commande. En exprimant le problème de calibration comme une minimisation, nous bénéficions des techniques liées au domaine de l'optimisation. Notamment, l'observabilité du système s'étudie analytiquement : en fonctionnement normal, il est donc possible de filtrer les séquences de mesures afin de conserver un système bien conditionné. À l'inverse, s'il est possible de contrôler l'effecteur spécifiquement pour la calibration, des techniques de vision active montrent une nette amélioration des performances.

Fusion de capteurs : L'approche proposée pour la fusion de capteurs partait du jacobien étendu. Nous avons conservé la modélisation consistant à associer une tâche à chaque capteur. Elle est héritée à la fois pour les approches hiérarchiques et pour les fusions par moyenne de vitesse ou par commande commutative. Cette modélisation au niveau des capteurs apporte beaucoup de souplesse pour définir des lois de commande référencées multi-capteurs. Toutefois elle ne structure par la loi de commande par jacobien étendu : la pondération s'adresse en effet au niveau des informations capteur. Le socle de notre approche consiste à définir pour chaque information une pondération générique, en partant d'une observation : certaines informations capteurs peuvent atteindre des valeurs a priori arbitraires lors de la tâche de positionnement. D'autres sont limitées à un intervalle de fonctionnement. Cela peut être pour des raisons mécaniques (articulations, mesure de distance) auquel cas atteindre les limites peut endommager le matériel, ou bien des raisons liées à la nature des informations (vision), qui n'assurent plus le bon fonctionnement du

capteur si les limites sont atteintes (l'objet sort de l'image). Le poids générique suit cette logique et se définit par :

- Si l'information considérée n'a pas de contrainte, un poids fixe peut lui être attribué. Un parallèle immédiat étant dressé entre l'approche fonction de tâche et la commande LQ, les méthodes de réglage des poids sont donc les mêmes. En pratique, sans objectifs particuliers de comportement dynamique, un réglage à 1 consiste à se rapprocher d'une décroissance simultanée de l'ensemble des erreurs.
- Si l'information considérée est contrainte, alors un poids variable est utilisé. Le poids est paramétré par ρ, défini dans l'espace de la mesure capteur, qui traduit la prise en compte de l'information approchant trop près de sa limite.

L'approche ainsi envisagée rejoint le domaine des tâches à dimension variable. La principale différence est que la pondération n'est pas limitée à des valeurs entre 0 et 1. De multiples contraintes peuvent alors rentrer en concurrence et être toutes respectées, même pour une tâche de navigation de rang plein.

Commande référencée capteur : Les solutions apportées pour pallier les défauts de la commande proposée peuvent s'appliquer à l'ensemble des schémas référencés capteur.

La définition de tâches dans l'espace du capteur pose la question de la cohérence des informations à la position désirée. En l'absence d'une phase d'apprentissage, un modèle parfait est requis pour être certain que le vecteur $\mathbf{s}^*$ est effectivement atteignable : typiquement une cible carrée ne se projette jamais sous la forme d'un rectangle si les pixels sont carrés. En cas d'incertitudes sur la cohérence de la tâche, le paramètre de progression $\xi(\|\mathbf{e}_t\|)$ permet de favoriser certaines composantes par rapport à d'autres et de converger vers la pose désirée. Ainsi certaines informations sont utilisées dans la loi de commande uniquement jusqu'à ce que les informations principales soient suffisamment proches de leur valeur finale. L'algorithme de sortie d'un minimum local repose sur le fait que la pondération $\mathbf{H}$ de la tâche modifie la trajectoire mais pas le minimum global s'il correspond à $\mathbf{e} = 0$. En simulation, des minima ont ainsi pu être évités avant de converger vers la pose désirée.

Asservissement visuel : Au-delà de la fusion générique proposée, l'application à l'asservissement visuel hybride se révèle particulièrement intéressante. La situation où un PBVS est réalisé à partir d'un modèle 3D est très courante. De là, injecter dans la loi de commande les points 2D correspondant à l'enveloppe du modèle se fait directement. Aucun traitement d'image supplémentaire n'est requis pour utiliser ces points virtuels. La position désirée des points se calcule elle aussi directement à partir de la pose désirée du PBVS. Le schéma correspondant montre de très bonnes performances, à la fois dans l'image et dans l'espace.

Évitement de butées : L'application à l'évitement de butée donne une nouvelle solution pour ce problème en cas de tâche de rang plein. La formulation analytique aboutit à l'expression simple d'une pondération minimale pour l'évitement, permettant de perturber le moins possible la tâche principale. Cette expression met en avant la notion d'évitement passif, où la tâche elle-même évite la butée. Dans le cas de plusieurs butées à éviter la pondération optimale reste difficile à déterminer, et la pondération générique est utilisée.

Expérimentations : Des simulations ont été réalisées pour illustrer les aspects spécifiques des méthodes proposées. Lorsque cela se révélait représentatif, nous avons procédé à des simulations exhaustives. Dans le cas de l'asservissement visuel hybride, cette étape permet de montrer expérimentalement - mais pas analytiquement, hélas - qu'aucun minimum local n'est atteint pour un schéma pourtant largement surdéterminé.

Les expériences ont été l'occasion de confronter les schémas proposés à des configurations particulièrement difficiles. La calibration est ainsi réalisée avec des mouvements arbitrairements décidés par le schéma d'asservissement visuel. L'évitement des butées se fait à partir d'informations visuelles entraînant une mauvaise trajectoire dans l'espace. L'asservissement visuel hybride utilise un objet grand dans l'image et nécessite de larges rotations. La combinaison de contraintes, en réalisant une tâche au plus près des limites acceptables, soulève la question de la limite de la méthode proposée et surtout de la détection d'une telle limite.

Perspectives

Ce mémoire laisse plusieurs pistes ouvertes. Les premières sont de l'ordre des applications et des capteurs utilisés. Le formalisme en lui-même mériterait d'être plus abouti afin de donner davantage de valeur aux résultats expérimentaux.

Calibration : Si nous avons évalué les performances brutes de l'approche par erreur du modèle cinématique, il serait intéressant de lui adjoindre un filtrage plus robuste que la simple moyenne proposée. En posant le modèle cinématique sous forme d'état, nous pourrions prendre en compte la possibilité de bruit de mesure (vitesse et/ou capteur) afin de ne pas uniquement les subir. Par ailleurs, la vision active pour la calibration main-œil serait intéressante à développer même s'il est peu probable de pouvoir la réaliser de façon analytique.

Capteurs et applications : Une extension directe aux travaux de cette thèse porte sur la nature des capteurs et des robots utilisés. Nous avons évoqué à plusieurs reprises les capteurs de distance : il serait intéressant de porter ces travaux pour des robots mobiles dotés de tels capteurs. Si des capteurs scalaires (infrarouge, ultrasons) conduiraient simplement à des tâches de navigation avec suivi de mur, des capteurs plus complexes comme le LiDAR (Light Detection And Ranging) permettraient de prendre en compte l'évitement d'obstacles et la navigation en environnement inconnu. Quelques simulations simples ont été réalisées dans cette direction, mais le cœur de l'étude reste à faire.

Formalisme de la fusion : Si le formalisme général bénéficie des avantages de la commande référencée capteurs, il subit également ses défauts. L'absence de preuve de stabilité globale est toujours délicate au moment d'utiliser une loi de commande . Une hypothèse forte pour le respect de contraintes génériques est que la tâche doit rester de rang plein. Cette hypothèse pourrait être relâchée en utilisant l'opérateur d'inverse continu, en le généralisant pour des poids supérieurs à 1. Le schéma purement réactif

pourrait également bénéficier d'une approche prédictive sur un horizon limité. Des travaux [Allibert 10] montrent en effet que le respect des contraintes peut être anticipé avec cette classe de lois de commande. Il nous paraît aussi intéressant d'étudier formellement la différence entre l'approche proposée dans ce mémoire et celle utilisant l'inverse continue avec la pile de tâche [Mansard 09b]. Les objectifs de ces deux méthodes sont en effet très proches : réaliser des tâches référencées capteur tout en respectant un nombre arbitraire de contraintes. Alors que nous développons le compromis offert par le jacobien étendu, la hiérarchie de la pile de tâche est assouplie par l'introduction de la pondération. L'étude formelle et expérimentale des différences entre ces deux approches paraît donc séduisante pour identifier les équivalences et les principales différences. Un autre formalisme, proposé dans [Kanoun 11], pose le problème de commande sous contraintes comme un problème d'optimisation et soulève certaines équivalences entre les différentes approches par pseudo-inverse. Une approche unifiée pourrait éventuellement être déterminée pour différents degrés de redondance.

Faisabilité d'une tâche sous contraintes : Les simulations exhaustives ont révélé des situations de minima locaux dûs à une nécessité de regonfigurer le robot. Nous avons proposé une stratégie pour sortir d'un minimum local, et une autre pour atteindre une position désirée qui se trouverait hors de la zone de confiance. Dans le cadre d'un environnement inconnu, il paraît nécessaire de pouvoir identifier les situations où les contraintes peuvent être ignorées - car le robot est presque à destination - et celles où le minimum en passe d'être atteint ne correspond vraiment pas à la pose désirée. Face à un minimum local, la stratégie de sortie a tendance à faire rebrousser chemin. L'autre stratégie a tendance à faire continuer dans la même direction, en s'autorisant temporairement à se rapprocher des contraintes pour passer dans un couloir étroit. Plus généralement, si la méthode proposée permet de respecter les contraintes habituelles en commande référencée capteurs, le défaut inhérent de ce type de commande ressort. Hors des situations d'apprentissage, comment s'assurer que des tâches définies dans l'espace des mesures correspondent à des poses atteignables dans l'espace opérationnel ? Cette problématique se retrouve par exemple dans le cas du suivi d'un objet mobile. La question de la stratégie à adopter se pose alors pour les positions qui ne sont pas atteignables.

Annexes

Annexe A

Calibration entre une caméra et une centrale inertielle

Dans la Section 2.2.4.2 nous avons présenté une approche réduite de la calibration extrinsèque, où la caméra observe une cible connue permettant d'estimer sa vitesse $\mathbf{v}_s$. Dans cette section, nous appliquons cette approche à la calibration entre une caméra et une centrale inertielle.

Sur les robots manipulateurs la vitesse de l'effecteur est facilement mesurée avec le modèle cinématique direct. Sur d'autres types de robots, notamment les robots mobiles, cette méthode n'est pas applicable et la vitesse est souvent estimée à l'aide d'une centrale inertielle. Le problème est alors d'estimer le changement de repère entre la caméra et la centrale inertielle.

Nous définissons tout d'abord les notations du modèle de centrale inertielle, avant d'appliquer notre approche sur ce problème.

A.1 Modèle de centrale inertielle

Une centrale inertielle classique est composée d'un accéléromètre 3 axes et d'un gyroscope 3 axes. Ces capteurs fournissent les mesures de l'accélération linéaire $\mathbf{a}_e$ et de la vitesse angulaire $\boldsymbol{\omega}_e$ dans le repère de la centrale que nous assimilons au repère de l'effecteur $\mathcal{F}_e$. Le vecteur des mesures inertielles $(\mathbf{a}_m, \boldsymbol{\omega}_m)$ étant souvent bruité et surtout biaisé, nous définissons les biais d'accélération $\mathbf{b}_a$ et de vitesse angulaire $\mathbf{b}_\omega$. La mesure fournie par la centrale inertielle est ainsi :

$$\begin{cases} \mathbf{a}_m &= \mathbf{a}_e + \mathbf{b}_a \\ \boldsymbol{\omega}_m &= \boldsymbol{\omega}_e + \mathbf{b}_\omega \end{cases} \tag{A.1}$$

Dans la suite, nous estimons le biais $\mathbf{b} = (\mathbf{b}_a, \mathbf{b}_\omega)$ tout en réalisant la calibration.
Nous introduisons les notations suivantes :

- $\mathcal{F}_w$ est le repère fixe de convention NED (North - East - Down) dans lequel on exprime la pose absolue du robot ${}^e\mathbf{M}_w$. Le vecteur gravité est donc ${}^w\mathbf{g} = g\vec{k}_w$.
- $\mathcal{F}_o$ est le repère fixe associé à la cible observée par la caméra, permettant d'estimer la pose ${}^s\mathbf{R}_o$.
- $\boldsymbol{v}_w$ est la vitesse linéaire de la centrale inertielle, exprimée dans le repère $\mathcal{F}_w$.

On suppose que la direction de la gravité dans le repère objet est connue et notée ${}^o\mathbf{g}$.
Par soucis d'homogénéité avec l'approche générique proposée dans le chapitre 2, le repère de la caméra est dénoté $\mathcal{F}_s$ en lieu et place du $\mathcal{F}_c$ habituel en vision par ordinateur. La pose de la caméra est notée ${}^s\mathbf{M}_o$. Le système est représenté en Figure A.1.

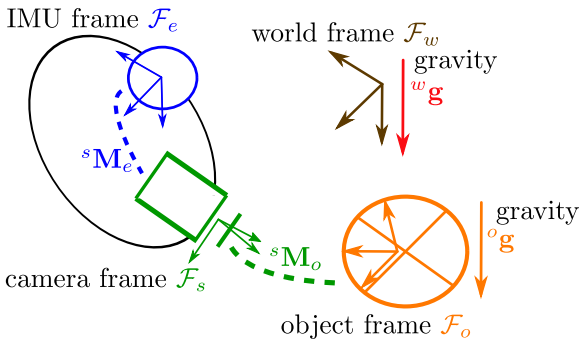


FIG. A.1 – Système caméra-centrale inertielle. La gravité est connue dans le repère objet.

Les accéléromètres mesurant la force de gravité, l'estimation de l'accélération linéaire nécessite d'estimer la rotation entre $\mathcal{F}_e$ et $\mathcal{F}_w$. Cette valeur peut être exprimée en fonction de ${}^s\mathbf{R}_e$ par :

$$ {}^e\mathbf{R}_w\, {}^w\mathbf{g} = {}^s\mathbf{R}_e^\top\, {}^s\mathbf{R}_o\, {}^o\mathbf{R}_w\, {}^w\mathbf{g} = {}^s\mathbf{R}_e^\top\, {}^s\mathbf{R}_o\, {}^o\mathbf{g} \tag{A.2} $$

où ${}^s\mathbf{R}_o$ est obtenue par l'estimation de pose de la caméra et ${}^o\mathbf{g}$ est supposé connu.

Ainsi, le torseur cinématique $\mathbf{v}_e$ de la centrale inertielle n'est pas mesuré directement mais dépend de l'estimation courante de ${}^s\mathbf{R}_e$, $\mathbf{b}_a$ et $\mathbf{b}_\omega$. Dans le repère monde $\mathcal{F}_w$, la dérivée de la vitesse linéaire de la centrale inertielle s'exprime par :

$$ \dot{\boldsymbol{v}}_w = {}^w\mathbf{R}_e \mathbf{a}_e + {}^w\mathbf{g} \tag{A.3} $$

Dans le repère de la centrale $\mathcal{F}_e$, on a donc :

$$ \begin{aligned} \dot{\boldsymbol{v}}_e &= {}^e\mathbf{R}_w \dot{\boldsymbol{v}}_w = \mathbf{a}_e + {}^e\mathbf{R}_w\, {}^w\mathbf{g} = \mathbf{a}_e + {}^s\mathbf{R}_e^\top\, {}^s\mathbf{R}_o\, {}^o\mathbf{g} \\ &= \mathbf{a}_m - \mathbf{b}_a + {}^s\mathbf{R}_e^\top\, {}^s\mathbf{g} \end{aligned} \tag{A.4} $$

où ${}^s\mathbf{g}$ est l'expression de la gravité dans le repère caméra, obtenue par l'estimation de pose. En intégrant (A.4) on obtient l'estimation de la vitesse linéaire entre deux images :

$$ \begin{aligned} \boldsymbol{v}_e &= \int \mathbf{a}_m dt - \int \mathbf{b}_a dt + {}^s\mathbf{R}_e^\top \int {}^s\mathbf{g} dt \\ &= \boldsymbol{v}_m - \mathbf{b}_a T + {}^s\mathbf{R}_e^\top \boldsymbol{v}_g \end{aligned} \tag{A.5} $$

où :

- $T = \int dt$ est le temps d'acquisition
- $\boldsymbol{v}_m = \int \mathbf{a}_m dt$ est une vitesse reconstruite par intégration de la mesure $\mathbf{a}_m$
- $\boldsymbol{v}_g = \int {}^s\mathbf{g} dt$ est une vitesse reconstruite par intégration de la gravité ${}^s\mathbf{g}$ exprimée dans le repère caméra

Par ailleurs, d'après (A.1), l'estimation de la vitesse angulaire est :

$$ \boldsymbol{\omega}_e = \boldsymbol{\omega}_m - \mathbf{b}_\omega \tag{A.6} $$

Le torseur cinématique lié à la centrale inertielle étant exprimé en fonction des mesures et des valeurs estimées, nous appliquons maintenant le formalisme proposé.

A.2 Formulation de l'approche

Par analogie avec (2.37), une variation infinitésimale de l'erreur (2.36) s'exprime par :

$$\mathrm{d}\tilde{\boldsymbol{\varepsilon}}_{(\widehat{\boldsymbol{\xi}_{\mathrm{ex}}},\widehat{\mathbf{b}})} = \mathrm{d}^{s}\mathbf{W}_e\ \mathbf{v}_e + {}^{s}\mathbf{W}_e\ \mathrm{d}\mathbf{v}_e \tag{A.7}$$

où $\mathrm{d}^{s}\mathbf{W}_e$ est défini en (2.20) et exprime la dépendance de ${}^{s}\mathbf{W}_e$ envers l'estimation des paramètres extrinsèques $\boldsymbol{\xi}_{\mathrm{ex}}$. Comme exposé en (2.37), on a :

$$\mathrm{d}^{s}\mathbf{W}_e\ \mathbf{v}_e\ = \mathbf{L}_{\tilde{\boldsymbol{\varepsilon}}}\bar{\mathbf{v}} \tag{A.8}$$

où $\mathbf{L}_{\tilde{\boldsymbol{\varepsilon}}}$ a été déterminée en Section 2.2.4.

$\mathbf{v}_e$ étant fonction à la fois de ${}^{s}\mathbf{R}_e$ et de $\mathbf{b}$, sa variation $\mathrm{d}\mathbf{v}_e$ fait intervenir à la fois $\bar{\mathbf{v}}$ et $\mathrm{d}\mathbf{b}$. On peut donc écrire :

$${}^{s}\mathbf{W}_e\ \mathrm{d}\mathbf{v}_e = \mathbf{L}_{\mathbf{g}}\bar{\mathbf{v}} + \mathbf{J}_{\mathbf{b}}\mathrm{d}\mathbf{b} \tag{A.9}$$

où :

- $\mathbf{L}_{\mathbf{g}}$ est une matrice d'interaction additionnelle exprimant la dépendance de $\mathbf{v}_e$ envers l'estimation de ${}^{s}\mathbf{R}_e$.
- $\mathbf{J}_{\mathbf{b}}$ est le jacobien de l'erreur par rapport aux biais $\mathbf{b}$.

D'après (A.8) et (A.9), la variation infinitésimale de l'erreur (A.7) peut ainsi s'écrire :

$$\mathrm{d}\tilde{\boldsymbol{\varepsilon}}_{(\widehat{\boldsymbol{\xi}_{\mathrm{ex}}},\widehat{\mathbf{b}})} = (\mathbf{L}_{\tilde{\boldsymbol{\varepsilon}}} + \mathbf{L}_{\mathbf{g}})\bar{\mathbf{v}} + \mathbf{J}_{\mathbf{b}}\mathrm{d}\mathbf{b} \tag{A.10}$$

Le minimisation se fait alors avec la loi de mise à jour :

$$\begin{bmatrix} \bar{\mathbf{v}} \\ \mathrm{d}\mathbf{b} \end{bmatrix} = -\lambda \begin{bmatrix} \mathbf{L}_{\tilde{\boldsymbol{\varepsilon}}}+\mathbf{L}_{\mathbf{g}} & \mathbf{J}_{\mathbf{b}} \end{bmatrix}^{+} ({}^{s}\mathbf{W}_e\mathbf{v}_e - \mathbf{v}_s) \tag{A.11}$$

Nous déterminons maintenant les expressions de $\mathbf{L}_{\mathbf{g}}$ et $\mathbf{J}_{\mathbf{b}}$.

A.3 Expression des jacobiens

D'après (A.5), l'évolution de $\boldsymbol{v}_e$ pendant la minimisation est :

$$\begin{aligned}
\mathrm{d}\boldsymbol{v}_e &= -T\mathrm{d}\mathbf{b}_a + T\mathrm{d}{}^{s}\mathbf{R}_e^{\top}{}^{s}\mathbf{g} = -T\mathrm{d}\mathbf{b}_a + \left({}^{s}\mathbf{R}_e\left[\bar{\boldsymbol{\omega}}\right]_{\times}\right)^{\top}\boldsymbol{v}_g \\
&= -T\mathrm{d}\mathbf{b}_a + \left[\bar{\boldsymbol{\omega}}\right]_{\times}^{\top}{}^{s}\mathbf{R}_e^{\top}\boldsymbol{v}_g \\
&= -T\mathrm{d}\mathbf{b}_a + \left[{}^{s}\mathbf{R}_e^{\top}\boldsymbol{v}_g\right]_{\times}\bar{\boldsymbol{\omega}}
\end{aligned} \tag{A.12}$$

et par ailleurs, d'après (A.6) :

$$\mathrm{d}\boldsymbol{\omega}_e = -\mathrm{d}\mathbf{b}_{\omega} \tag{A.13}$$

Par identification avec (A.7), les matrices $\mathbf{L}_{\mathbf{g}}$ et $\mathbf{J}_{\mathbf{b}}$ du problème de calibration s'expriment ainsi par :

$$\mathbf{L}_{\mathbf{g}} = {}^{s}\mathbf{W}_e\begin{bmatrix} \mathbf{0}_{3\times3} & \left[{}^{s}\mathbf{R}_e^{\top}\boldsymbol{v}_g\right]_{\times} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times3} & {}^{s}\mathbf{R}_e\left[{}^{s}\mathbf{R}_e^{\top}\boldsymbol{v}_g\right]_{\times} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{A.14}$$

$$\mathbf{J}_{\mathbf{b}} = -{}^{s}\mathbf{W}_e\begin{bmatrix} T\mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix} = -\begin{bmatrix} T{}^{s}\mathbf{R}_e & \left[{}^{s}\mathbf{t}_e\right]_{\times}{}^{s}\mathbf{R}_e \\ \mathbf{0}_{3\times3} & {}^{s}\mathbf{R}_e \end{bmatrix} \tag{A.15}$$

Les jacobiens sont ainsi déterminés et la calibration caméra-centrale inertielle peut être effectuée. Par rapport aux approches classiques de calibration caméra / centrale inertielle, notre formulation ne fait aucune hypothèse sur les mouvements appliqués.

Bibliographie

[Allibert 10] G. Allibert, E. Courtial & F. Chaumette. Predictive control for constrained image-based visual servoing. IEEE Transactions on Robotics, vol. 26, no. 5, pages 933–939, 2010. (Cité en pages 92 et 128)

[Aloimonos 88] J. Aloimonos, I. Weiss & A. Bandyopadhyay. Active vision. International Journal of Computer Vision, vol. 1, no. 4, pages 333–356, 1988. (Cité en page 37)

[Andreff 98] N. Andreff, Bernard Espiau & Radu Horaud. Une méthode d'auto-étalonnage pince-caméra. Rapport de recherche RR-3507, INRIA, 1998. (Cité en page 18)

[Andreff 01] N. Andreff, R. Horaud & B. Espiau. Robot hand-eye calibration using structure-from-motion. The International Journal of Robotics Research, vol. 20, no. 3, page 228, 2001. (Cité en pages 19 et 24)

[Antonelli 09] G. Antonelli. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. IEEE Transactions on Robotics, vol. 25, no. 5, pages 985–994, 2009. (Cité en page 49)

[Baerlocher 04] P. Baerlocher & R. Boulic. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. The visual computer, vol. 20, no. 6, pages 402–417, 2004. (Cité en page 49)

[Baillieul 84] J. Baillieul, J. Hollerbach & R. Brockett. Programming and control of kinematically redundant manipulators. In IEEE Conference on Decision and Control, volume 23, pages 768–774. IEEE, 1984. (Cité en page 5)

[Basu 93] A. Basu. Active calibration : Alternative strategy and analysis. In IEEE Conference on Computer Vision and Pattern Recognition, pages 495–500. IEEE, 1993. (Cité en page 19)

[Borenstein 96] J. Borenstein & Liqiang Feng. Measurement and correction of systematic odometry errors in mobile robots. IEEE Transactions on Robotics and Automation, vol. 12, no. 6, pages 869–880, December 1996. (Cité en page 17)

[Boutteau 08] R. Boutteau, X. Savatier, JY Ertaud & B. Mazari. An omnidirectional stereoscopic system for mobile robot navigation. In Robotic and Sensors Environments, 2008. ROSE 2008. International Workshop on, pages 138–143. IEEE, 2008. (Cité en page 80)

[Branicky 98] M.S. Branicky, V.S. Borkar & S.K. Mitter. A unified framework for hybrid control : Model and optimal control theory. IEEE Transactions on Automatic Control, vol. 43, no. 1, pages 31–45, 1998. (Cité en page 49)

[Brooks 95] SP Brooks & BJT Morgan. Optimization using simulated annealing. The Statistician, vol. 44, no. 2, pages 241–257, 1995. (Cité en page 72)

[Brown 71] D.C. Brown. Close-Range Camera Calibration. Photogrammetric Engineering, vol. 37, no. 8, pages 855–866, August 1971. (Cité en page 17)

[Cervera 03] E. Cervera, AP Del Pobil, F. Berry & P. Martinet. Improving image-based visual servoing with three-dimensional features. The International Journal of Robotics Research, vol. 22, no. 10-11, page 821, 2003. (Cité en page 13)

[Chan 95] T.F. Chan & R.V. Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. IEEE Transactions on Robotics and Automation, vol. 11, no. 2, pages 286–292, 1995. (Cité en pages 52, 65 et 107)

[Chaumette 96] F. Chaumette, S. Boukir, P. Bouthemy & D. Juvin. Structure from Controlled Motion. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 5, pages 492–504, May 1996. (Cité en page 19)

[Chaumette 98] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In The Confluence of Vision and Control, pages 66–78. LNCIS Series, No 237, 1998. (Cité en pages 13, 66, 91 et 108)

[Chaumette 01] F. Chaumette & E. Marchand. A redundancy-based iterative approach for avoiding joint limits : Application to visual servoing. IEEE Trans. on Robotics and Automation, vol. 17, no. 5, pages 719–730, October 2001. (Cité en page 65)

[Chaumette 02] F. Chaumette. Asservissement visuel. In W. Khalil, editeur, La commande des robots manipulateurs, Traité IC2, chapitre 3, pages 105–150. Hermès, 2002. (Cité en page 25)

[Chaumette 04] F. Chaumette. Image moments : a general and useful set of features for visual servoing. IEEE Transactions on Robotics, vol. 20, no. 4, pages 713–723, August 2004. (Cité en page 13)

[Chaumette 06] F. Chaumette & S. Hutchinson. Visual servo control. I. Basic approaches. IEEE Robotics and Automation Magazine, vol. 13, no. 4, pages 82–90, 2006. (Cité en pages 11, 12, 28, 56, 57 et 60)

[Chaumette 07] F. Chaumette & S. Hutchinson. Visual servo control, Part II : Advanced approaches. IEEE Robotics and Automation Magazine, vol. 14, no. 1, pages 109–118, March 2007. (Cité en page 11)

[Cheah 07] CC Cheah, DQ Wang & YC Sun. Region-reaching control of robots. IEEE Transactions on Robotics, vol. 23, no. 6, pages 1260–1264, 2007. (Cité en pages 43, 52 et 60)

[Cherubini 11] A. Cherubini, F. Spindler & F. Chaumette. A Redundancy-Based Approach for Visual Navigation with Collision Avoidance. In IEEE Symp. on Computational Intelligence in Vehicles and Transportation Systems, Paris, April 2011. (Cité en page 50)

[Chesi 04] G. Chesi, K. Hashimoto, D. Prattichizzo & A. Vicino. Keeping features in the field of view in eye-in-hand visual servoing : A switching approach. IEEE Transactions on Robotics, vol. 20, no. 5, pages 908–914, 2004. (Cité en pages 50 et 91)

[Collewet 08] C. Collewet, E. Marchand & F. Chaumette. Visual servoing set free from image processing. In IEEE Int. Conf. on Robotics and Automation, pages 81–86, Pasadena, California, May 2008. (Cité en pages 8 et 13)

[Comport 05] A.I. Comport, E. Marchand & F. Chaumette. Efficient model-based tracking for robot vision. Advanced Robotics, vol. 19, no. 10, pages 1097–1113, October 2005. (Cité en pages 8 et 102)

[Comport 06] A.I. Comport, E. Marchand & F. Chaumette. Statistically robust 2-D visual servoing. IEEE Transactions on Robotics, vol. 22, no. 2, pages 415–420, 2006. (Cité en pages 52, 54 et 59)

[Conticelli 01] F. Conticelli & B. Allotta. Nonlinear controllability and stability analysis of adaptive image-based systems. IEEE Transactions on Robotics and Automation, vol. 17, no. 2, pages 208–214, 2001. (Cité en page 36)

[Corke 01] P.I. Corke & S.A. Hutchinson. A new partitioned approach to image-based visual servo control. IEEE Transactions on Robotics and Automation, vol. 17, no. 4, pages 507–515, 2001. (Cité en page 91)

[Corke 07] P. Corke, J. Lobo & J. Dias. An Introduction to Inertial and Visual Sensing. Int. J. of Robotics Research, vol. 26, no. 6, pages 519–535, June 2007. (Cité en page 8)

[Dame 11] A. Dame & E. Marchand. Mutual information-based visual servoing. IEEE Trans. on Robotics, vol. 27, no. 5, October 2011. (Cité en page 57)

[De Luca 06] A. De Luca, G. Oriolo & P.R. Giordano. Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators. In IEEE Int. Conf. on Robotics and Automation, pages 1867–1873. IEEE, 2006. (Cité en page 50)

[De Luca 08] Alessandro De Luca, Guiseppe Oriolo & Paolo Robuffo Giordano. Feature Depths Observation for Image-based Visual Servoing : Theory and Experiments. Int. J. of Robotics Research, vol. 27, no. 10, pages 1093–1116, October 2008. (Cité en pages 19, 26, 27, 28, 36 et 37)

[Dementhon 95] D.F. Dementhon & L.S. Davis. Model-based object pose in 25 lines of code. Int. J. of Computer Vision, vol. 15, no. 1, pages 123–141, 1995. (Cité en page 8)

[Dionnet 09] F. Dionnet & E. Marchand. Stereo tracking and servoing for space applications. Advanced Robotics, vol. 23, no. 5, pages 579–599, April 2009. (Cité en page 80)

[Drummond 02] T. Drummond & R. Cipolla. Real-time visual tracking of complex structures. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pages 932–946, 2002. (Cité en page 8)

[Du 93] F. Du & M. Brady. Self-calibration of the intrinsic parameters of cameras for active vision systems. In IEEE on Computer Vision and Pattern Recognition, pages 477–482. IEEE, 1993. (Cité en page 19)

[Dune 07] C. Dune, E. Marchand & C. Leroux. One click focus with eye-in-hand/eye-to-hand cooperation. In IEEE Int. Conf. on Robotics and Automation, Roma, Italy, April 2007. (Cité en page 80)

[Dune 09] C. Dune. Localisation et caractérisation d'objets inconnus à partir d'informations visuelles : vers une saisie intuitive pour les personnes en situation de handicap. PhD thesis, Université de Rennes 1, Mention informatique, April 2009. (Cité en page 43)

[Durrant-Whyte 87] H.F. Durrant-Whyte. Integration, coordination and control of multi-sensor robot systems. Kluwer Academic Publishers, 1987. (Cité en page 7)

[Erdmann 87] M. Erdmann & T. Lozano-Perez. On multiple moving robots. Algorithmica, vol. 2, no. 4, pages 477–521, 1987. (Cité en page 7)

[Espiau 87] B. Espiau & P. Rives. Closed-loop recursive estimation of 3D features for a mobile vision system. In IEEE Int. Conf. on Robotics and Automation, volume 4, pages 1436–1443. IEEE, 1987. (Cité en page 8)

[Espiau 92] B. Espiau, F. Chaumette & P. Rives. A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation, vol. 8, no. 3, pages 313–326, June 1992. (Cité en pages 6, 11 et 81)

[Eynard 10] D. Eynard, P. Vasseur, C. Demonceaux & V. Frémont. UAV altitude estimation by mixed stereoscopic vision. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 646–651. IEEE, 2010. (Cité en page 80)

[Fang 06] K. Fang, R. Li & A. Sudjianto. Design and modeling for computer experiments. CRC Press, 2006. (Cité en pages 72 et 108)

[Faugeras 87] O.D Faugeras & G. Toscani. Camera Calibration for 3D Computer Vision. In Int. Workshop on Machine Vision and Machine Intelligence, pages 240–247, Tokyo, February 1987. (Cité en page 17)

[Faugeras 92] O. Faugeras, Q. T. Luong & S.-J. Maybank. Camera Self-Calibration : Theory and Experiments. In 2nd European Conf. on Computer Vision, pages 321–334, London, UK, 1992. Springer-Verlag. (Cité en page 19)

[Faugeras 93] O. Faugeras. Three-dimensional computer vision : a geometric viewpoint. MIT Press, Cambridge, MA, 1993. (Cité en pages 9 et 10)

[Fisher 25] R.A. Fisher. Theory of statistical estimation. In Mathematical Proceedings of the Cambridge Philosophical Society, volume 22, pages 700–725. Cambridge Univ Press, 1925. (Cité en page 37)

[Flandin 00] G. Flandin, F. Chaumette & E. Marchand. Eye-in-hand / eye-to-hand cooperation for visual servoing. In IEEE Int. Conf. on Robotics and Automation, San Francisco, California, April 2000. (Cité en pages 47 et 80)

[Folio 05] D. Folio & V. Cadenat. Using redundancy to avoid simultaneously occlusions and collisions while performing a vision-based task amidst obstacles. In Proceedings of European Conference on Mobile Robots, pages 92–97, 2005. (Cité en page 48)

[Folio 08] D. Folio & V. Cadenat. Dealing with visual features loss during a vision-based task for a mobile robot. Int. J. of Optomechatronics, vol. 2, pages 185–204, 2008. (Cité en pages 19, 26 et 50)

[Gans 07] N.R. Gans & S.A. Hutchinson. Stable visual servoing through hybrid switched-system control. IEEE Transactions on Robotics, vol. 23, no. 3, pages 530–540, 2007. (Cité en pages 49, 76, 91, 92, 95 et 102)

[García-Aracil 05] N. García-Aracil, E. Malis, R. Aracil-Santonja & C. Pérez-Vidal. Continuous visual servoing despite the changes of visibility in image features. IEEE Transactions on Robotics, vol. 21, no. 6, pages 1214–1220, 2005. (Cité en pages 52 et 54)

[Grassia 98] F.S. Grassia. Practical parameterization of rotations using the exponential map. Journal of graphics tools, vol. 3, pages 29–48, 1998. (Cité en page 11)

[Hafez 06] A.H.A. Hafez & CV Jawahar. Improvement to the Minimization of Hybrid Error Functions for Pose Alignment. In International Conference on Control, Automation, Robotics and Vision, pages 1–6. IEEE, 2006. (Cité en page 51)

[Hafez 07a] A.H.A. Hafez & CV Jawahar. Probabilistic Integration of 2D and 3D Cues for Visual Servoing. In 9th IEEE Int. Conf. on Control, Automation, Robotics and Vision, pages 1–6, 2007. (Cité en pages 50 et 51)

[Hafez 07b] A.H.A. Hafez & CV Jawahar. Visual servoing by optimization of a 2D/3D hybrid objective function. In IEEE Int. Conf. on Robotics and Automation, Roma, Italy, 2007. (Cité en pages 51, 61 et 92)

[Hafez 08] A.H.A. Hafez, E. Cervera & CV Jawahar. Hybrid Visual Servoing by Boosting IBVS and PBVS. In 3rd Int. Conf. on Information and Communication Technologies : From Theory to Applications, pages 1–6. IEEE, 2008. (Cité en pages 50, 92, 95, 96 et 102)

[Hager 97] G.D. Hager. A modular system for robust positioning using feedback from stereo vision. IEEE Transactions on Robotics and Automation, vol. 13, no. 4, pages 582–595, 1997. (Cité en page 50)

[Hartley 94] R. Hartley. Self-calibration from multiple views with a rotating camera. Computer Vision—ECCV'94, pages 471–478, 1994. (Cité en pages 19 et 37)

[Hemayed 03] E.E. Hemayed. A survey of camera self-calibration. In IEEE Conf. on Advanced Video and Signal Based Surveillance., pages 351–357, July 2003. (Cité en page 19)

[Heyden 99] A. Heyden & K. Astrom. Flexible calibration : Minimal cases for auto-calibration. In IEEE International Conference on Computer Vision, volume 1, pages 350–355. IEEE, 1999. (Cité en page 18)

[Horaud 95] R. Horaud & F. Dornaika. Hand-eye calibration. The international journal of robotics research, vol. 14, no. 3, page 195, 1995. (Cité en page 18)

[Horaud 06] R. Horaud, D. Knossow & M. Michaelis. Camera cooperation for achieving visual attention. Machine Vision and Applications, vol. 16, no. 6, pages 1–2, 2006. (Cité en page 80)

[Hosoda 98] K. Hosoda, K. Igarashi & M. Asada. Adaptive hybrid control for visual and force servoing in an unknown environment. IEEE Robotics and Automation Magazine, vol. 5, no. 4, pages 39–43, 1998. (Cité en pages 8 et 47)

[Isidori 95] A. Isidori. Nonlinear control systems. Springer Verlag, 1995. (Cité en page 60)

[Kanoun 11] Oussama Kanoun, Florent Lamiraux & Pierre-Brice Wieber. Kinematic Control of Redundant Manipulators : Generalizing the Task-Priority Framework to Inequality Task. IEEE Transactions on Robotics, vol. 27, no. 4, pages 785–792, 2011. (Cité en pages 49, 53 et 128)

[Keith 11] F. Keith, P.B. Wieber, N. Mansard & A. Kheddar. Analysis of the Discontinuities in Prioritized Tasks-Space Control Under Discreet Task Scheduling Operations. September 2011. (Cité en page 61)

[Kelly 11] J. Kelly & G.S. Sukhatme. Visual-Inertial Sensor Fusion : Localization, Mapping and Sensor-to-Sensor Self-calibration. The International Journal of Robotics Research, vol. 30, no. 1, page 56, 2011. (Cité en page 15)

[Khalil 04] W. Khalil & E. Dombre. Modeling, identification & control of robots. Butterworth-Heinemann, 2004. (Cité en page 3)

[Khatib 87] O. Khatib. A unified approach for motion and force control of robot manipulators : The operational space formulation. IEEE Transactions on Robotics and Automation, vol. 3, no. 1, pages 43–53, 1987. (Cité en page 51)

[Khatib 97] O. Khatib, S. Quinlan & D. Williams. Robot planning and control. Robotics and Autonomous Systems, vol. 21, no. 3, pages 249–261, 1997. (Cité en page 7)

[Kruppa 13] E. Kruppa. Zur Ermittlung eines Objektes aus zwei Perspektiven mit Innerer Orientierung, Sitz. Ber. Akad. Wiss., Wien, Math. Naturw. Kl., Abt. IIa, vol. 122, pages 1939–1948, 1913. (Cité en page 19)

[Laumond 94] J.P. Laumond, P.E. Jacobs, M. Taix & R.M. Murray. A motion planner for nonholonomic mobile robots. IEEE Transactions on Robotics and Automation, vol. 10, no. 5, pages 577–593, 1994. (Cité en page 7)

[Liégeois 77] A. Liégeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Transactions on Systems, Man, and Cybernetics, vol. 7, no. 12, pages 868–871, 1977. (Cité en page 48)

[Lippiello 07] V. Lippiello, B. Siciliano & L. Villani. Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration. IEEE Trans. on Robotics, vol. 23, no. 1, pages 73–86, 2007. (Cité en pages 8 et 80)

[Lobo 07] J. Lobo & J. Dias. Relative pose calibration between visual and inertial sensors. The International Journal of Robotics Research, vol. 26, no. 6, page 561, 2007. (Cité en page 18)

[Lozano-Pérez 79] T. Lozano-Pérez & M.A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. Communications of the ACM, vol. 22, no. 10, pages 560–570, 1979. (Cité en page 7)

[Luo 02] R.C. Luo, C.C. Yih & K.L. Su. Multisensor fusion and integration : approaches, applications, and future research directions. IEEE Sensors J., vol. 2, no. 2, pages 107–119, 2002. (Cité en page 7)

[Malis 99] E. Malis, F. Chaumette & S. Boudet. 2-1/2D visual servoing. IEEE Transactions on Robotics and Automation, vol. 15, no. 2, pages 238–250, April 1999. (Cité en pages 13, 51, 92, 95 et 101)

[Malis 00] E. Malis & F. Chaumette. 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. Int. J. of Computer Vision, vol. 37, no. 1, pages 79–97, 2000. (Cité en page 12)

[Malis 01] E. Malis, G. Morel & F. Chaumette. Robot Control Using Disparate Multiple Sensors. Int. J. of Robotics Research, vol. 20, no. 5, pages 364–377, May 2001. (Cité en pages 8, 50, 80 et 84)

[Malis 04a] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In IEEE Int. Conf. on Robotics and Automation, volume 2, pages 1843–1848. Citeseer, 2004. (Cité en pages 57 et 58)

[Malis 04b] E. Malis. Visual servoing invariant to changes in camera-intrinsic parameters. IEEE Trans. on Robotics and Automation, 2004. (Cité en page 60)

[Mansard 07] N. Mansard & F. Chaumette. Task sequencing for sensor-based control. IEEE Transactions on Robotics, vol. 23, no. 1, pages 60–72, February 2007. (Cité en page 49)

[Mansard 09a] N. Mansard & F. Chaumette. Directional redundancy for robot control. IEEE Transactions onAutomatic Control, vol. 54, no. 6, pages 1179–1192, 2009. (Cité en pages 49, 55 et 106)

[Mansard 09b] N. Mansard, O. Khatib & A. Kheddar. A unified approach to integrate unilateral constraints in the stack of tasks. Robotics, IEEE Transactions on, vol. 25, no. 3, pages 670–685, 2009. (Cité en pages 49, 53, 69 et 128)

[Mansard 09c] N. Mansard, A. Remazeilles & F. Chaumette. Continuity of varying-feature-set control laws. IEEE Transactions on Automatic Control, vol. 54, no. 11, pages 2493–2505, November 2009. (Cité en pages 52, 53, 59, 69 et 77)

[Marchand 96] E. Marchand, F. Chaumette & A. Rizzo. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'96, volume 3, pages 1083–1090, Osaka, Japan, November 1996. (Cité en pages 55 et 106)

[Marchand 98] E. Marchand & G.D. Hager. Dynamic sensor planning in visual servoing. In IEEE Int. Conf. on Robotics and Automation, ICRA'98, volume 3, pages 1988–1993, Leuven, Belgium, May 1998. (Cité en page 48)

[Marchand 01] E. Marchand & F. Chaumette. A new formulation for non-linear camera calibration using virtual visual servoing. Rapport technique 1366, IRISA, January 2001. (Cité en pages 15, 17, 20 et 32)

[Marchand 02] E. Marchand, F. Chaumette, F. Spindler & M. Perrier. Controlling an uninstrumented manipulator by visual servoing. The International Journal of Robotics Research, vol. 21, no. 7, page 635, 2002. (Cité en pages 57 et 116)

[Marchand 05] E. Marchand, F. Spindler & F. Chaumette. ViSP for visual servoing : a generic software platform with a wide class of robot control skills. IEEE Robot. Autom. Mag., vol. 12, no. 4, December 2005. (Cité en pages 9 et 28)

[Marey 10a] M. Marey. Contributions to control modeling in visual servoing, task redundancy, and joint limits avoidance. PhD thesis, Université de Rennes 1, Mention informatique, December 2010. (Cité en page 48)

[Marey 10b] M. Marey & F. Chaumette. A new large projection operator for the redundancy framework. In IEEE Int. Conf. on Robotics and Automation, Anchorage, Alaska, May 2010. (Cité en page 49)

[Marey 10c] M. Marey & F. Chaumette. New strategies for avoiding robot joint limits : Application to visual servoing using a large projection operator. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pages 6222–6227, Taipei, Taiwan, October 2010. (Cité en pages 55, 62, 65, 77, 106, 111 et 112)

[Marino 96] R. Marino & P. Tomei. Nonlinear control design : geometric, adaptive and robust. 1996. (Cité en page 27)

[Martin 02] F. Martin & R. Horaud. Multiple-camera tracking of rigid objects. The International Journal of Robotics Research, vol. 21, no. 2, page 97, 2002. (Cité en page 80)

[Martinet 97] P. Martinet, N. Daucher, J. Gallice & M. Dhome. Robot control using monocular pose estimation. Worshop on New Trends in Image-based Robot Servoing, IROS'97, vol. 97, pages 1–12, Sept 1997. (Cité en page 12)

[Maybank 92] S.-J. Maybank & O. Faugeras. A theory of self calibration of a moving camera. Int. J. of Computer Vision, vol. 8, no. 1, pages 123–152, 1992. (Cité en pages 18 et 19)

[Mellinger 10] D. Mellinger, N. Michael & V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. In International Symposium on Experimental Robotics, Delhi, India, 2010. (Cité en page 8)

[Mirzaei 08] F.M. Mirzaei & S.I. Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration : Observability analysis and performance evaluation. IEEE Transactions on Robotics, vol. 24, no. 5, pages 1143–1156, 2008. (Cité en pages 18 et 19)

[Moakher 02] M. Moakher. Means and averaging in the group of rotations. SIAM journal on matrix analysis and applications, vol. 24, pages 1–16, 2002. (Cité en page 22)

[Moons 96] T. Moons, L. Van Gool, M. Proesmans & E. Pauwels. Affine reconstruction from perspective image pairs with a relative object-camera translation in between. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 1, pages 77–83, 1996. (Cité en pages 19 et 37)

[Moravec 85] H. Moravec & A. Elfes. High resolution maps from wide angle sonar. In IEEE Int. Conf. on Robotics and Automation, volume 2, pages 116–121. IEEE, 1985. (Cité en page 7)

[Morel 00] G. Morel, T. Liebezeit, J. Szewczyk, S. Boudet & J. Pot. Explicit incorporation of 2d constraints in vision based control of robot manipulators. Experimental Robotics VI, pages 99–108, 2000. (Cité en pages 51 et 91)

[Nadeau 11] C. Nadeau & A. Krupa. Intensity-based direct visual servoing of an ultrasound probe. In IEEE Int. Conf. on Robotics and Automation, ICRA'11, pages 5677–5682, Shanghai, China, May 2011. (Cité en page 47)

[Nakamura 87] Y. Nakamura, H. Hanafusa & T. Yoshikawa. Task-priority based redundancy control of robot manipulators. The International Journal of Robotics Research, vol. 6, no. 2, page 3, 1987. (Cité en page 49)

[Nakamura 91] Y. Nakamura. Advanced robotics : redundancy and optimization. Addison-Wesley Pub. Co., 1991. (Cité en pages 48 et 50)

[Nelson 95] B.J. Nelson & P.K. Khosla. Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance. The Int. J. of Robotics Research, vol. 14, no. 3, page 255, 1995. (Cité en pages 5 et 106)

[Nelson 96] B.J. Nelson & P.K. Khosla. Force and vision resolvability for assimilating disparate sensory feedback. IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pages 714–731, 1996. (Cité en page 50)

[Papanikolopoulos 92] NP Papanikolopoulos, B. Nelson & PK Khosla. Full 3-d tracking using the controlled active vision paradigm. In IEEE Int. Symposium on Intelligent Control, pages 267–274. IEEE, 1992. (Cité en page 51)

[Park 01] J. Park, Y. Choi, W.K. Chung & Y. Youm. Multiple tasks kinematics using weighted pseudo-inverse for kinematically redundant manipulators. In IEEE International Conference on Robotics and Automation, volume 4, pages 4041–4047. IEEE, 2001. (Cité en page 48)

[Paul 81] R.P. Paul. Robot manipulators : mathematics, programming, and control : the computer control of robot manipulators. The MIT Press, 1981. (Cité en page 6)

[Paul 83] R.P. Paul & C.N. Stevenson. Kinematics of robot wrists. The International journal of robotics research, vol. 2, no. 1, page 31, 1983. (Cité en page 5)

[Pollefeys 99] M. Pollefeys & L. Van Gool. Stratified self-calibration with the modulus constraint. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 8, pages 707–724, 1999. (Cité en page 19)

[Pressigout 07] M. Pressigout & E. Marchand. Real-time hybrid tracking using edge and texture information. Int. Journal of Robotics Research, vol. 26, no. 7, pages 689–713, July 2007. (Cité en page 8)

[Remazeilles 06] A. Remazeilles, N. Mansard & F. Chaumette. Qualitative visual servoing : Application to the visibility constraint. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pages 4297–4303, Beijing, China, October 2006. (Cité en pages 43, 52, 54, 60 et 64)

[Remy 97] S. Remy, M. Dhome, JM Lavest & N. Daucher. Hand-eye calibration. In IEEE/RSJ International Conference on Intelligent Robots and Systems, volume 2, pages 1057–1065. IEEE, 1997. (Cité en page 18)

[Samson 91] C. Samson, M. Le Borgne & B. Espiau. Robot control : The task function approach. Clarendon Press, 1991. (Cité en pages 5, 6, 45, 48 et 56)

[Sanderson 80] AC Sanderson & LE Weiss. Image-based visual servo control using relational graph error signals. IEEE Int. Conf. on Cybernetics and Society, vol. 68, pages 1074–1077, October 1980. (Cité en page 11)

[Sanderson 83] A. C. Sanderson & L. E. Weiss. Adaptive Visual Servo Control of Robots. In A. Pugh, editeur, Robot Vision, pages 107–116. Springer-Verlag, New York, 1983. (Cité en page 12)

[Schramm 05] F. Schramm, G. Morel, A. Micaelli & A. Lottin. Extended-2d visual servoing. In IEEE Int. Conf. on Robotics and Automation, 2005. (Cité en pages 12 et 60)

[Sentis 05] L. Sentis & O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. International Journal of Humanoid Robotics, vol. 2, no. 4, pages 505–518, 2005. (Cité en page 49)

[Seraji 98] H. Seraji. A unified approach to motion control of mobile manipulators. The International Journal of Robotics Research, vol. 17, no. 2, page 107, 1998. (Cité en page 50)

[Shen 07] W. Shen & J. Gu. Multi-criteria kinematics control for the pa10-7c robot arm with robust singularities. In IEEE International Conference on Robotics and Biomimetics, pages 1242–1248. IEEE, 2007. (Cité en page 51)

[Siciliano 91] B. Siciliano & J.J.E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In IEEE Fifth Int. Conf. on Advanced Robotics, 1991. (Cité en pages 48 et 106)

[Skordas 95] T. Skordas. Une revue des derniers progrès en autocalibration de caméras CCD. Traitement du Signal, vol. 12, no. 1, 1995. (Cité en page 19)

[Sturm 97] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 1100–1105. IEEE, 1997. (Cité en page 19)

[Sturm 99] P.F. Sturm & S.J. Maybank. On plane-based camera calibration : A general algorithm, singularities, applications. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1. IEEE, 1999. (Cité en page 19)

[Tahri 09] O. Tahri & Y. Mezouar. On visual servoing based on efficient second order minimization. Robotics and Autonomous Systems, 2009. (Cité en pages 57, 58 et 59)

[Triggs 98] B. Triggs. Autocalibration from planar scenes. Computer Vision—ECCV'98, pages 89–105, 1998. (Cité en page 15)

[Tsai 87] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal of Robotics and Automation, vol. 3, no. 4, pages 323–344, August 1987. (Cité en pages 15 et 17)

[Tsai 89] R.Y. Tsai & R.K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. IEEE Transactions on Robotics and Automation, vol. 5, no. 3, pages 345–358, 1989. (Cité en pages 18, 24, 25 et 32)

[Van Loan 85] C. Van Loan. On the method of weighting for equality-constrained least-squares problems. SIAM Journal on Numerical Analysis, pages 851–864, 1985. (Cité en page 61)

[Viéville 94] T. Viéville. Auto-calibration of visual sensor parameters on a robotic head. Image and Vision Computing, vol. 12, no. 4, pages 227–237, 1994. (Cité en page 19)

[Voyles 97] R.M. Voyles, J.D. Morrow & P.K. Khosla. The shape from motion approach to rapid and precise force/torque sensor calibration. Tran. ASME-G-Journal of Dynamic Systems Measurement and Control, vol. 119, no. 2, pages 229–235, 1997. (Cité en page 17)

[Watson 75] PC Watson & SH Drake. Pedestal and wrist force sensors for automatic assembly. In Proceedings of the 5th International Symposium on Industrial Robots, pages 501–511, 1975. (Cité en page 16)

[Wei 98a] G.Q. Wei, K. Arbter & G. Hirzinger. Active self-calibration of robotic eyes and hand-eye relationships with model identification. IEEE Transactions on Robotics and Automation, vol. 14, no. 1, pages 158–166, 1998. (Cité en pages 19 et 37)

[Wei 98b] G.Q. Wei & G. Hirzinger. Active self-calibration of hand-mounted laser range finders. IEEE Transactions on Robotics and Automation, vol. 14, no. 3, pages 493–497, 1998. (Cité en page 19)

[Weiss 85] L.E. Weiss, A.C. Sanderson & CP Neuman. Dynamic visual servo control of robots : an adaptive image-based approach. In IEEE Int. Conf. on Robotics and Automation, volume 2, pages 662–668. IEEE, 1985. (Cité en pages 8 et 11)

[Weiss 87] L. Weiss, A. Sanderson & C. Neuman. Dynamic sensor-based control of robots with visual feedback. IEEE Transactions on Robotics and Automation, vol. 3, no. 5, pages 404–417, 1987. (Cité en page 12)

[Wilson 02] WJ Wilson, W. Hulls & GS Bell. Relative end-effector control using cartesian position based visual servoing. IEEE Trans. on Robotics and Automation, vol. 12, no. 5, pages 684–696, 2002. (Cité en page 12)

[Wittenmark 87] B. Wittenmark, R.J. Evans & Y.C. Soh. Constrained pole-placement using transformation and LQ-design* 1. Automatica, vol. 23, no. 6, pages 767–769, 1987. (Cité en pages 51 et 54)

[Xiang 10] J. Xiang, C. Zhong & W. Wei. General-weighted least-norm control for redundant manipulators. IEEE Transactions on Robotics, no. 99, pages 1–10, 2010. (Cité en pages 52, 53 et 65)

[Yoshikawa 96] T. Yoshikawa. Basic optimization methods of redundant manipulators. Laboratory Robotics and Automation, vol. 8, no. 1, pages 49–60, 1996. (Cité en pages 8 et 106)

[Zghal 02] H. Zghal, RV Dubey & JA Euler. Efficient gradient projection optimization for manipulators with multiple degrees of redundancy. In IEEE Int. Conf. on Robotics and Automation, 2002. (Cité en page 106)

[Zhang 00] Z. Zhang. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pages 1330–1334, 2000. (Cité en page 19)

[Zhang 04] Q. Zhang & R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, volume 3, pages 2301–2306. IEEE, 2004. (Cité en page 15)

[Zhu 91] DJ Zhu & J.C. Latombe. New heuristic algorithms for efficient hierarchical path planning. IEEE Transactions on Robotics and Automation, vol. 7, no. 1, pages 9–20, 1991. (Cité en page 7)

Résumé

Cette thèse s'inscrit dans le cadre de la commande référencée capteurs. Pour un robot mobile ou un bras manipulateur doté d'un ou plusieurs capteurs, cette approche consiste à définir une tâche de déplacement en spécifiant uniquement les mesures à atteindre. Ce paradigme est connu pour permettre une très bonne précision de positionnement. En revanche, le robot peut présenter un comportement non désiré en cas de mouvement important ou de contraintes à respecter (évitement des butées articulaires, visibilité, obstacles, etc.). Les solutions nécessitent généralement des degrés de liberté supplémentaires, permettant de réaliser la tâche tout en respectant les contraintes.

Nous proposons tout d'abord un formalisme original pour la calibration d'un capteur extéroceptif à partir de la mesure de vitesse. Il est ainsi possible d'estimer à la fois les paramètres de calibration intrinsèques du capteur et la position du capteur par rapport au repère du robot. Des résultats de simulation et d'expérience sont exposés pour la calibration d'une caméra. Nous établissons ensuite un formalisme complet pour la réalisation d'une tâche référencée multi-capteurs soumise à un nombre arbitraire de contraintes. Nous proposons une façon générique d'introduire les contraintes dans la loi de commande. Le comportement induit est une perturbation minimale de la tâche référencée capteurs tout en respectant plusieurs contraintes. Ce formalisme est appliqué à des problèmes classiques en robotique : évitement de butées et contrainte de visibilité. De nombreuses contraintes sont enfin combinées au sein d'un même système.

Mots clés : Robotique, commande référencée capteurs, fusion de capteurs, asservissement visuel, calibration de capteurs, contraintes

Abstract

This thesis deals with sensor-based control of robots. For a robot arm or a mobile robot, this approach consists in defining robotic tasks directly in the sensor space. This scheme is known for its good properties of accuracy and stability. The robot may still have an undesired behavior when the motion to perform is large or when constraints have to be respected. Classical schemes uses supplementary degrees of freedom to perform the main task while respecting the constraints.

First we propose a new framework for sensor calibration from velocity measurements. The same formulation allows to estimate both intrinsic and extrinsic parameters of an exteroceptive sensor. Simulations and experiments illustrate this approach in the case of camera calibration. A generic framework is then exposed for multi-sensor-based control under several constraints. All constraints are introduced in the control law in the same way. The induced behavior is the respect of all constraints, while perturbing the main task as little as possible. Our formalism is applied to classical issues in robotics, that are joint limits avoidance and the visibility constraint in visual servoing. Numerous constraints are then combined into a single system.

Keywords : Robotics, sensor-based control, sensor fusion, visual servoing, sensor calibration, constraints