N° d'ordre : 2070
EDSPIC : 497

Université Blaise Pascal - Clermont II

École Doctorale
Sciences Pour L'Ingénieur de Clermont-Ferrand

THÈSE

Présentée par :

Redwan Dahmouche

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

Spécialité: Vision pour la Robotique

Titre de la thèse :

Contributions à l'estimation de mouvement 3D et à la commande par vision rapide :

Application aux robots parallèles

Soutenue publiquement le 18 Novembre 2010 devant le jury :

M.	Wisama	Khalil	Président
M.	François	Pierrot	Rapporteur
M.	Éric	Marchand	Rapporteur
M.	Youcef	Mezouar	Co-encadrant
M.	Philippe	Martinet	Co-encadrant
M.	Nicolas	Andreff	Directeur de thèse

A mes parents

Remerciements

Je tiens d'abord à remercier les membres du jury :

- Wisama Khalil, Professeur à l'École Centrale de Nantes, d'avoir bien voulu présider le jury.
- François Pierrot, Directeur de Recherche au LIRMM (CNRS) et Éric Marchand, Professeur à l'Université de Rennes 1 pour m'avoir fait l'honneur d'être les rapporteurs de cette thèse. Leurs critiques et remarques, pertinentes, m'ont permis de prendre plus de recul par rapport aux travaux effectués et d'affiner la qualité du document.
- Philippe Martinet, Professeur à l'IFMA et Youcef Mezouar Maître de Conférences à l'Université Blaise Pascal et HDR pour m'avoir encadré tout au long de cette thèse. Leurs conseils, critiques et suggestions ont été déterminants aussi bien dans l'avancement des travaux que dans ma propre formation.
- Nicolas Andreff, Professeur à l'Université de Franche-Comté, pour son implication, sa disponibilité et sa rigueur. Les nombreuses discussions scientifiques, riches et contradictoires, ainsi que la confiance et la marge de manœuvre dont j'ai pu jouir m'ont été très profitables.

Je tiens aussi à remercier les membres de l'IRCCyN et plus particulièrement Damien Chablat, Philippe Lemoine et Wisama Khalil d'avoir bien voulu m'ouvrir les portes du laboratoire et d'avoir mis à ma disposition leur excellent robot "Orthoglide" sans lequel ces travaux n'auraient pas pu être menés jusqu'au bout.

Enfin, un grand merci à mon épouse qui m'a soutenu, encouragé, et supporté tout au long de la thèse.

Table des matières

Remerciements **5**

Introduction **15**

1 État de l'art **19**

1.1 Vision pour la robotique 19

1.1.1 Composants d'un système de vision 19

1.1.2 Modèle de projection d'une caméra perspective 20

1.1.3 Configuration de la caméra 21

1.2 La commande par vision 22

1.2.1 Le formalisme des fonctions de tâche 22

1.2.2 Architectures de commande 24

1.2.3 Type de primitives 26

1.3 Asservissement visuel cinématique 26

1.3.1 Asservissement visuel 3D 26

1.3.2 Asservissement visuel 2D 29

1.3.3 Asservissement visuel hybride 31

1.3.4 Dualité entre l'asservissement visuel et l'estimation de pose 32

1.4 Asservissement visuel dynamique 33

1.4.1 Commande pour la vision 35

1.4.2 Vision pour la commande 38

1.5 Vision pour la commande des robots parallèles 40

1.6 Conclusion 46

2 Estimation de pose et de vitesse par vision rapide **49**

2.1 Introduction 49

2.2 Modélisation 52

2.2.1 Acquisition séquentielle de fenêtres d'intérêt 52

2.2.2 Modèle de projection 53

2.2.3 Modèle du mouvement 54

2.2.3.1 Intégration du mouvement d'un corps solide dans l'espace 54

2.2.3.2 Modèle de vitesse constante dans le repère mobile 55
2.2.3.3 Modèle de vitesse constante dans le repère absolu 56
2.2.3.4 Modèle de mouvement à accélération constante 57
2.2.3.5 Intégration du mouvement en fonction du modèle cinématique du robot 58
2.2.3.6 Modèle de projection 59
2.3 Estimation de pose et de vitesse par optimisation non-linéaire 61
2.3.1 Calcul de la matrice Jacobienne 62
2.3.2 Algorithme d'estimation 65
2.3.2.1 Choix du temps de référence 65
2.3.2.2 Suivi de points dans l'image 65
2.3.2.3 Ordonnancement parallèle 66
2.4 Estimation d'état par filtrage de Kalman étendu 66
2.4.1 Estimation de pose et de vitesse 69
2.4.2 Coût calculatoire 72
2.5 Conclusion 75

3 Asservissement visuel cinématique par vision rapide 77
3.1 Introduction 77
3.2 Synthèse de lois de commande cinématiques 78
3.2.1 La fonction de tâche 78
3.2.2 Asservissement visuel à partir de primitives spatio-temporelles . . . 83
3.2.2.1 Commande à partir du torseur cinématique 83
3.2.2.2 Commande à partir de l'accélération 85
3.2.2.3 Commande à partir du torseur cinématique et de l'accélération 86
3.2.3 Asservissement visuel virtuel 88
3.2.3.1 Version explicite 88
3.2.3.2 Version simplifiée 90
3.2.4 Asservissement visuel combiné : Réel-Virtuel 91
3.3 Calcul de la matrice d'interaction 92
3.3.1 Matrice d'interaction du point 2D 94
3.3.2 Calcul de la matrice d'interaction complète 94
3.4 Estimation approchée de la matrice d'interaction 95
3.4.1 Approximation du modèle de mouvement 96
3.4.2 Approximation de la matrice d'interaction 98
3.5 Conclusion 99

4 Commande dynamique par vision rapide **101**
4.1 Introduction . . . 101
4.2 Utilisation de l'estimation par vision pour la commande dynamique . . . 102
4.3 Stabilité de la commande dynamique face aux retards sur l'estimation . . . 104
4.3.1 Modélisation des différents éléments de la commande . . . 105
4.3.1.1 Modélisation de la boucle de commande . . . 105
4.3.1.2 Modèle du correcteur . . . 105
4.3.1.3 Modélisation des processus d'acquisition et d'estimation . 107
4.3.1.4 Modélisation du robot + compensation de la dynamique . 109
4.3.2 Analyse de stabilité de la commande avec un correcteur PID . . . 111
4.3.3 Analyse de stabilité de la commande avec un correcteur PD . . . 115
4.3.3.1 Discussion sur le réglage des paramètres . . . 117
4.4 Exploitation du vecteur de commande dans l'estimation d'état . . . 117
4.5 Asservissement visuel dynamique des robots parallèles par vision rapide . . 119
4.6 Comparaison entre la commande dynamique 3D et l'asservissement visuel dynamique 2D . . . 124
4.7 Conclusion . . . 125

5 Résultats expérimentaux **127**
5.1 Dispositif expérimental . . . 127
5.2 Caractérisation du système de vision et de la rigidité du robot . . . 129
5.3 Estimation de pose et de vitesse . . . 132
5.3.1 Choix de la matrice d'interaction . . . 132
5.3.2 Suivi d'un objet rapide avec estimation de la pose et de la vitesse . 133
5.3.3 Estimation par filtrage de Kalman étendu . . . 138
5.4 Asservissement visuel 3D dynamique . . . 140
5.5 Asservissement visuel 2D dynamique . . . 146
5.6 Conclusion . . . 153

Conclusion générale et perspectives **157**

A Notations **163**

B Filtrage de Kalman **167**
B.1 Le Filtre de Kalman linéaire . . . 167
B.2 Le filtre de Kalman étendu . . . 169

Table des figures

1.1 Chaine d'acquisition simple pour un système de vision. 20
1.2 Modèle de projection classique de points appartenant à un objet rigide. . . 21
1.3 Configuration de la caméra : (a) Embarquée ; (b) Déportée. 22
1.4 Architecture de la commande cinématique "look-and-move". 25
1.5 Architecture de la commande par retour direct. 25
1.6 Dualité entre les asservissements visuels réel et virtuel. (a) Asservissement visuel réel où l'image acquise correspond à la pose courante et l'image de référence à la pose désirée. (b) Asservissement visuel virtuel où l'image capturée correspond à la pose désirée et les points dans l'image obtenue par projection correspondent à la pose estimée (pose virtuelle courante). . 33
1.7 Exemples de robots à structure cinématique parallèles. (a) Plate-forme de Gough-Stewart ; (b) Le robot Quattro. 40
1.8 Exemples de robots à chaine ouverte simple. (a) Le Motoman ; (b) Le robot Scara. 41
1.9 Exemple d'utilisation d'un robot parallèle (plate-forme de Gough-Stewart) dans un simulateur de vol. 42
1.10 Schéma de commande en couple calculé de robots parallèles dans l'espace articulaire avec mesure articulaire. 44
1.11 Schéma de commande en couple calculé de robots parallèles dans l'espace cartésien avec mesure de position et de vitesse cartésienne [PAM09]. 44

2.1 Acquisition de primitives visuelles dans la scène par sélection de régions d'intérêt. (a) Image entière acquise par une caméra standard. (b) Fenêtres d'intérêt acquises et replacées sur un fond gris. 50
2.2 Distorsions dans l'image induites par une acquisition séquentielle de fenêtres d'intérêt d'un objet rigide en mouvement. 52
2.3 Projection séquentielle de points appartenant à un objet rigide en mouvement. 53
2.4 Intégration du mouvement d'un repère mobile dont la vitesse est constante dans son propre repère 55

2.5 Intégration du mouvement d'un repère mobile dont la vitesse est constante dans le repère absolu. 57
2.6 Ordonnancement série et parallèle pour un système d'acquisition séquentielle de fenêtres d'intérêt. 67
2.7 Structure de l'algorithme d'estimation de pose et de vitesse par minimisation des moindres carrés à partir d'une acquisition séquentielle de fenêtres d'intéret. 68
2.8 Nombre d'opérations nécessaires pour l'estimation de pose et de vitesse dans une acquisition globale et dans une acquisition séquentielle à partir de la même quantité de données. 74

3.1 Écart entre les positions des points dans l'image pour une acquisition classique et pour une acquistion séquentielle.(a) Deux positions successives de la cible en acquisition classique. (b) Positions des primitives dans une acquisition globale et dans une acquisition séquentielle de fenêtres d'intérêts. 80
3.2 Asservissement visuel à partir du torseur cinématique 84
3.3 Asservissement visuel cinématique à partir de l'accélération 86
3.4 Asservissement visuel à partir du torseur cinématique et de l'accélération . 87
3.5 Asservissement visuel virtuel pour l'estimation de pose et de vitesse. 90
3.6 Asservissement visuel virtuel simplifié. 91
3.7 Asservissement visuel combiné réel-virtuel par acqsuisition non séquentielle. 92
3.8 Intégration des mouvements des points dans l'espace 3D 96

4.1 Commande en couple calculé par vision rapide. 103
4.2 Commande dynamique équivalente 105
4.3 Schéma de commande dynamique par vision avec une régulation PID . . . 111
4.4 Limite de stabilité de la commande dynamique par vision avec régulation PID en fonction des facteurs d'amortissement équivalents de la boucle d'estimation. 114
4.5 Limite de stabilité de la commande dynamique par vision avec régulation PD en fonction des facteurs d'amortissement équivalents de la boucle d'estimation. 116
4.6 Commande dynamique par vision exploitant l'accélération de commande. . 118
4.7 Asservissement visuel 2D en couple calculé 123
4.8 Asservissement visuel 2D en couple calculé avec estimation de pose et de vitesse 124

5.1 Dispositif Experimental montrant le robot parallèle "Orthoglide" sur lequel est monté une mire 3D dont les primitives sont capturées par une caméra rapide permettant l'acquisition séquentielle de fenêtres d'intérêt. 129

5.2 Chronogramme des tâches d'acquisition et de traitement (PC) synchronisées avec le processus de commande (carte de commande du robot). . . . 130

5.3 Déplacements des primitives dans l'image du fait des efforts manuels appliqués sur la plate-forme lorsque le robots est à l'arrêt (freins engagés). . . 131

5.4 Estimation par vision des déplacements de la plate-forme dans l'espace en raison des efforts manuels appliqués sur l'effecteur lorsque le robots est à l'arrêt (freins engagés). 131

5.5 Trajectoires de la plate-forme obtenues par vision et par le modèle géométrique direct du robot. (a) Trajectoires projetées sur le plan YZ. (b) Trajectoires projetées sur le plan XZ. 132

5.6 Position cartésienne de la plateforme (a) obtenue à partir des mesures articulaires et (b) à partir de l'estimation par vision et (c) l'erreur de position entre les deux estimations. 135

5.7 Vitesse cartésienne de la plateforme (a) obtenue à partir des mesures articulaires et (b) à partir de l'estimation par vision et (c) l'erreur de position entre les deux estimations. 136

5.8 Comparaison entre les erreurs de reprojection dans l'estimation de la pose et de la vitesse par asservissement visuel virtuel avec et sans terme d'anticipation. 137

5.9 Position de la plate-forme obtenue par (a) le modèle géométrique direct du robot et par (b) le filtre de Kalman étendu ainsi que (c) l'erreur entre les deux mesures. 139

5.10 Vitesse de la plate-forme obtenue par (a) le modèle géométrique direct du robot et par (b) le filtre de Kalman étendu ainsi que (c) l'erreur entre les deux mesures. 140

5.11 Schéma synoptique de l'architecture du système. 141

5.12 Comparaison entre les trajectoires obtenues par la commande référencée vision et par la commande référencée modèle projetées sur les plan (a) YZ et (b) XY. 142

5.13 (a) Trajectoire de consigne et les erreurs correspondant à (b) la commande référencée modèle et à (c) la commande référencée vision en fonction du temps. 143

5.14 (a) Vitesse de consigne et les erreurs correspondant (b) à la commande référencée modèle et (c) à la commande référencée vision. 144

5.15 Erreurs de suivi dans l'image en fonction du temps. 144

5.16 Positions des 16 primitives dans l'image dans la trajectoire acquises (points blancs) et leurs positions initiale (points rouges). 147

5.17 Positions des primitives dans l'image pour pulsation propre de 25 rad/s selon (a) l'axe u et (b) l'axe v ainsi que les erreurs entre les positions réelle et désirée sur (c) u et (d) v. 148
5.18 Trajectoire cartésienne de consigne et celle obtenue par l'asservissement visuel 2D dynamique (a) vues par la caméra et (b) représentée dans l'espace.150
5.19 (a) Positions cartésiennes de consigne et celles obtenue par l'asservissement visuel dynamique 2D et (b) l'erreur entre les deux. 151
5.20 (a) Positions cartésiennes de consigne et celles obtenue par l'asservissement visuel dynamique 2D et (b) l'erreur entre les deux. 152
5.21 Comparaison entre les positions cartésiennes. (a) vue normale et (b) vue agrandie. 153
5.22 Positions des primitives dans l'image pour pulsation propre de 50 rad/s selon (a) l'axe u et (b) l'axe v ainsi que les erreurs entre les positions réelle et désirée sur (c) u et (d) v. 154

B.1 Représentation d'un système dans l'espace d'état. 167
Table des figures

Introduction

L'utilisation de la vision en robotique se développe depuis environ trois décennies afin d'améliorer l'interaction des robots, initialement aveugles, avec leur environnement. Le nombre de travaux menés par la communauté scientifique témoigne de l'importance de ce domaine de recherche. Cependant, même si les capteurs de vision offrent aujourd'hui un rendu très fidèle de la scène observée à des fréquences supérieures à la persistance rétinienne, l'interprétation de l'information visuelle reste encore très complexe. Par conséquent, malgré la richesse de l'information visuelle fournie par les capteurs qui alimente la recherche en vision cognitive, le problème de la commande basée sur une mesure par vision est souvent restreint à un niveau d'abstraction inférieur. Les tâches à accomplir par le robot (positionnement, navigation, etc.) sont ainsi décrites à partir de primitives visuelles simples à interpréter (position de points, configuration de droites, etc).

Même si la commande par vision a déjà été utilisée pour quelques chaines de montages, cette solution technologique n'a cependant pas encore réussi à s'imposer dans le milieu industriel. L'un des défis que les chercheurs doivent relever pour que la commande référencée vision devienne une solution plus intéressante est d'améliorer ses performances dynamiques (précision, vitesse, répétabilité, etc.). En effet, l'un des inconvénients de l'asservissement visuel classique est que ses performances dynamiques sont relativement faibles par rapport aux commandes basées sur les mesures articulaires. Typiquement, les performances de cette commande sont limitées par la faible fréquence d'acquisition et le temps de latence induits par les systèmes de vision. La communauté scientifique s'est par conséquent plus intéressée à la commande cinématique, moins sensible à ces paramètres. Cependant, ce type de commande néglige la dynamique du système ce qui entraine une dégradation des performances de la commande.

Le problème de la fréquence d'acquisition des systèmes de vision classiques pose inévitablement la question de l'adéquation du mode d'acquisition dans les applications hautes performances. En effet, les systèmes de vision ont été initialement conçus à l'image de la vision humaine. L'objectif est alors de fournir un rendu fidèle de la scène à des fréquences proches de la persistance rétinienne. Toujours dans un esprit de biomimétisme, la commande référencée vision s'est ensuite intéressée à l'utilisation de cette mesure en commande sans forcément remettre en question le mode d'acquisition utilisé jusque-là. Le développement de systèmes plus performants (par rapports aux performances humaines,

entre autre) exige cependant une mesure plus adaptée à l'application envisagée. En effet, étant donné le potentiel intrinsèque de la lumière en termes de précision et de vitesse, le problème posé est celui du mode d'acquisition qui permet de l'exploiter au mieux.

Par ailleurs, les robots qui ont aujourd'hui les meilleures performances dynamiques sont les robots parallèles. La vision représente justement un outil de mesure pertinent pour la commande de ce type de robots. En effet, l'une des particularités des robots parallèles est qu'une configuration articulaire unique peut correspondre à plusieurs configurations de la plate-forme alors que l'inverse est souvent faux. Par conséquent, la pose et la vitesse de l'effecteur définissent d'une manière plus pertinente l'état du robot que la configuration articulaire. La commande des robots parallèles par observation des éléments du robot en général et de sa plate-forme en particulier représente ainsi une voie de recherche très prometteuse.

Nous présentons dans ce document une nouvelle approche pour l'asservissement visuel haute performance. En effet, nous verrons que l'utilisation d'un mode d'acquisition innovant permet de développer des méthodes d'estimation de pose et de vitesse ainsi que des schémas d'asservissements visuels cinématiques et dynamiques performants. La présentation de ces travaux sera ainsi organisée de la manière suivante.

Après un bref descriptif des systèmes de vision classiques et des bases théoriques de la vision artificielle, le chapitre I dresse un état de l'art de l'asservissement visuel. Les principales architecture de régulation ainsi que les schémas de commande les plus répandus seront alors présentés. On s'intéressera ensuite aux performances de ces derniers ainsi qu'aux différentes approches proposées dans la littérature pour les améliorer tant au niveau de la perception (la mesure par vision) qu'au niveau de la commande. Après avoir identifié ce qui limite les performances de la commande par vision, une démarche pour l'élaboration d'une commande haute performance par vision sera présentée. Enfin, nous verrons comment la vision peut être utilisée pour la commande dynamique des robots parallèles.

Dans le deuxième chapitre nous développerons des méthodes d'estimation rapide de la pose et de la vitesse nécessaires à l'implémentation de la commande dynamique. Nous présenterons tout d'abord un système de vision permettant d'augmenter la fréquence d'acquisition et de réduire le temps de latence. Cependant, le signal observé ne permettant pas l'utilisation des méthodes d'estimation classiques, deux méthodes d'estimation simultanée de la pose et de la vitesse serons proposées. Enfin, une comparaison entre les deux méthodes proposées sera présentée.

Dans le chapitre III, on s'intéressera plus particulièrement à la commande cinématique hautes performances. En effet, certains robots ont la particularité d'être très légers. Pour ce type de robots, il n'est pas toujours nécessaire d'effectuer une compensation de la dynamique. En effet, la commande cinématique peut s'avérer dans ce cas suffisante pour assurer de bonnes performances. Cependant, les vitesses atteintes par ces robots étant

élevées, il est tout de même nécessaire de disposer d'une acquisition haute fréquence. De plus, l'exécution de mouvements rapides nécessite souvent une phase préalable de génération de trajectoire. Or, l'asservissement visuel classique ne permettant pas de corriger les erreurs de vitesse, la précision du suivi peut être médiocre. Nous proposerons dans ce chapitre plusieurs schémas de commande permettant le suivi précis dans l'image à partir de mesures obtenues via le mode d'acquisition proposé. De plus, la dualité qui existe entre l'asservissement visuel et l'estimation sera exploitée pour la synthèse d'un schéma d'asservissement visuel virtuel permettant l'estimation de la pose et de la vitesse.

Dans le chapitre IV, on s'intéressera à la commande dynamique. Des schémas d'asservissements visuels 3D et 2D seront ainsi présentés. L'approche 3D utilise l'estimation de pose et de vitesse par vision rapide présentée pour la synthèse d'une commande en couple basée vision. Dans ce contexte, une étude de stabilité de la commande proposée par rapport aux paramètres d'estimation est présentée. Dans le second schéma de commande, la régulation est assurée directement dans l'image en vue d'améliorer sa robustesse face aux éventuelles erreurs de modélisation. On verra aussi que la réponse imposée au système n'est pas une décroissance exponentielle du premier ordre comme dans le cas classique mais une décroissance plus rapide de l'erreur afin d'assurer une meilleure convergence vers la solution désirée.

Le dernier chapitre montre les résultats expérimentaux obtenus à partir du mode d'acquisition proposé en estimation de pose et de vitesse, en commande en couple dans l'espace cartésien et en asservissement visuel dynamique d'un robot parallèle rapide.

Enfin, la conclusion générale tire un bilan sur les travaux effectués et propose quelques perspectives de recherche.

Les principales contributions de ces travaux sont :

- L'utilisation d'un nouveau mode d'acquisition rapide pour l'estimation et la commande ;
- La modélisation du mode d'acquisition proposé ;
- le calcul de la matrice jacobienne correspondant au modèle de projection ;
- L'estimation haute fréquence de pose et de vitesse à partir d'acquisition séquentielle par deux méthodes différentes : minimisation aux moindres carrés non linéaires et filtrage de Kalman étendu ;
- L'asservissement visuel cinématique avec correction de la vitesse ;
- La commande en couple référencée vision dans l'espace cartésien ;
- L'étude de stabilité de la commande dynamique proposée vis-à-vis des paramètres d'estimation ;
- L'asservissement visuel 2D dynamique.

Ces contributions ont été partiellement publiées dans :

- R. Dahmouche, N. Andreff, Y. Mezouar, and P. Martinet. Efficient High-speed Vision-based Computed Torque Control of the Orthoglide Parallel Robot. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'10)*, pages 644-649, Anchorage, AK, USA, May 2010.
- R. Dahmouche, N. Andreff, Y. Mezouar, and P. Martinet. 3D pose and velocity visual tracking based on sequential region of interest acquisition. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09)*, pages 5426-5431, Piscataway, NJ, USA, October 2009.
- R. Dahmouche, O. Ait-Aider, N. Andreff, and Y. Mezouar. High-speed pose and velocity measurement from vision. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, pages 107-112, Pasadena, CA, USA, May 2008.

Chapitre 1

État de l'art

Nous allons aborder dans ce chapitre les bases de la commande par vision vue sous l'angle des performances et de la robustesse. Ainsi, après un bref descriptif des composants qui constituent un système de vision, nous nous intéresserons à la commande référencée capteur en présentant les méthodes d'asservissement visuel les plus répandues. Nous aborderons par la même occasion la dualité entre l'asservissement visuel et l'estimation de pose et de vitesse. Après avoir passé en revue les bases théoriques concernant l'asservissement visuel, nous adresserons la problématique de l'amélioration des performances de la commande par vision aussi bien sur les aspects théoriques que matériels. Finalement, nous verrons pourquoi et comment l'utilisation de la vision peut être une bonne solution pour la commande dynamique des robots parallèles.

1.1 Vision pour la robotique

1.1.1 Composants d'un système de vision

La chaine d'acquisition dans les applications de vision artificielle temps-réel classique est constituée : d'une caméra intégrant un capteur de vision (CCD ou CMOS) et un objectif, d'une interface de communication (unidirectionnelle ou bidirectionnelle) comportant éventuellement une carte d'acquisition et d'une unité de traitement (PC par exemple). Lors du processus d'acquisition, la caméra, synchronisée en interne (horloge propre à la caméra) ou en externe, intègre l'information visuelle grâce au capteur de vision puis transmet l'image à l'unité de traitement à travers l'interface de communication. Après réception de l'image, la phase de traitement consiste à extraire et à exploiter l'information visuelle (Fig. 1.1).

Deux des propriétés les plus importantes qui caractérisent les performances d'un système de vision sont, d'une part, la résolution du capteur qui détermine la largeur du spectre spatial des images acquises, et donc les détails qui s'y trouvent, et d'autre part, la période d'échantillonnage de la caméra qui détermine la fréquence temporelle maximale

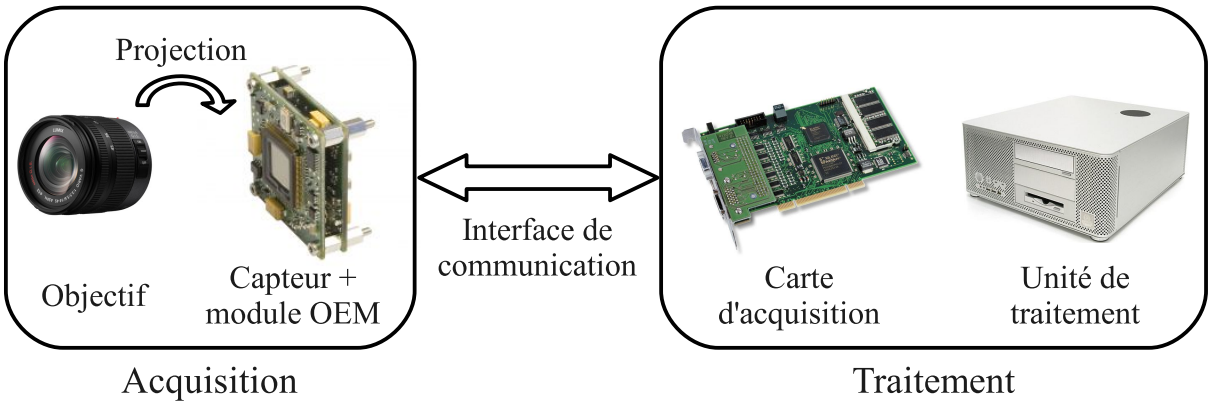


FIGURE 1.1 – Chaine d'acquisition simple pour un système de vision.

du signal acquis défini par le critère de Shannon [1]. Le choix de l'objectif permet ensuite de faire un compromis entre l'espace observé et la définition des images acquises.

1.1.2 Modèle de projection d'une caméra perspective

Soit ${}^{o}\mathbf{P}$ un point appartenant à un objet rigide et ${}^{o}\tilde{\mathbf{P}}$ sa représentation homogène (voir Fig. 1.2). Les coordonnées du point dans le repère caméra s'écrivent comme suit :

$$ {}^{c}\tilde{\mathbf{P}} = {}^{c}\mathbf{T}_{o}\ {}^{o}\tilde{\mathbf{P}} \tag{1.1} $$

où ${}^{c}\mathbf{T}_{o}$, la matrice de transformation homogène entre l'objet et la caméra est donnée par :

$$ {}^{c}\mathbf{T}_{o} = \begin{bmatrix} {}^{c}\mathbf{R}_{o} & {}^{c}\mathbf{t}_{o} \\ \mathbf{0} & 1 \end{bmatrix} \tag{1.2} $$

${}^{c}\mathbf{R}_{o}$ et ${}^{c}\mathbf{t}_{o}$ étant respectivement la matrice d'orientation et le vecteur de position entre l'objet et la caméra.

La projection du point 3D ${}^{c}\mathbf{P} = (x, y, z)^{T}$ dans le plan normalisé de l'image est donnée par :

$$ \mathbf{m} = \begin{bmatrix} X \\ \\ Y \end{bmatrix} = \begin{bmatrix} \frac{x}{z} \\ \\ \frac{y}{z} \end{bmatrix} \tag{1.3} $$

1. La fréquence d'acquisition d'une caméra doit être double de la fréquence temporelle contenue dans le signal acquis. Par exemple, la fréquence d'acquisition d'un système de vision utilisé pour la mesure des modes propres (vibrations) d'un robot doit obligatoirement vérifier la condition de Shannon par rapport à la fréquence des vibrations observées. Comme nous allons le voir par la suite, cette condition est d'autant plus forte dans des applications de commande.

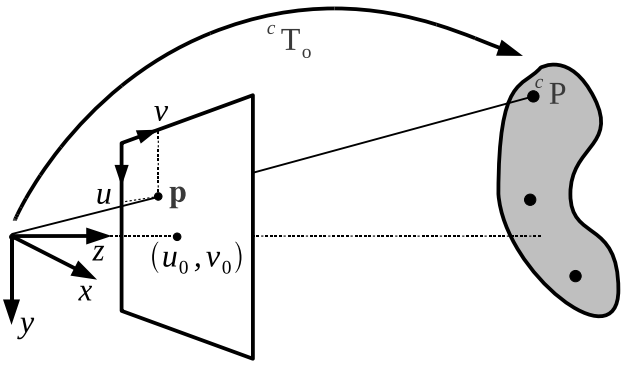


FIGURE 1.2 – Modèle de projection classique de points appartenant à un objet rigide.

Enfin, les coordonnées homogènes des points images en pixels $\tilde{\mathbf{p}}$ sont obtenues en utilisant la matrice des paramètres intrinsèques de la caméra [2] :

$$\tilde{\mathbf{p}} = \mathbf{K}\ \tilde{\mathbf{m}} \tag{1.4}$$

avec

$$\mathbf{K} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.5}$$

où α_u, α_v représentent les distances focales selon les directions respectives u et v, u_0 et v_0 correspondent à la position du centre optique dans l'image. Les vecteurs $\tilde{\mathbf{p}}$ et $\tilde{\mathbf{m}}$ sont les représentations homogènes des points $\mathbf{p}$ et $\mathbf{m}$.

1.1.3 Configuration de la caméra

Indépendamment des primitives choisies pour assurer la régulation de la commande, il existe deux configurations possibles de la caméra : caméra embarquée ou caméra déportée.

Dans la configuration caméra embarquée, la caméra est fixée sur l'organe terminal du robot (Fig. 1.3(a)). Dans ce cas, la transformation entre la caméra et l'outil est constante ce qui garantit le maintien de l'outil dans le champ de vue de la caméra. Cette transformation, souvent appelée base/caméra ou bras/œil doit être estimée soit par des méthodes spécifiques [HM95, ZR95, AHE01] soit conjointement à l'étalonnage du robot [KD99] afin de pouvoir traduire le mouvement désiré de la caméra par le mouvement correspondant du robot. La transformation entre la caméra et la cible est, quant à elle, variable. C'est pourquoi, lors de l'exécution de la commande, il faut veiller à ce que la cible soit toujours visible.

2. En pratique, certains objectifs peuvent introduire des distorsions dans l'images. Néanmoins, il est possible de modéliser ces distorsions, de les identifier et de les compenser [HM95, LVD98].

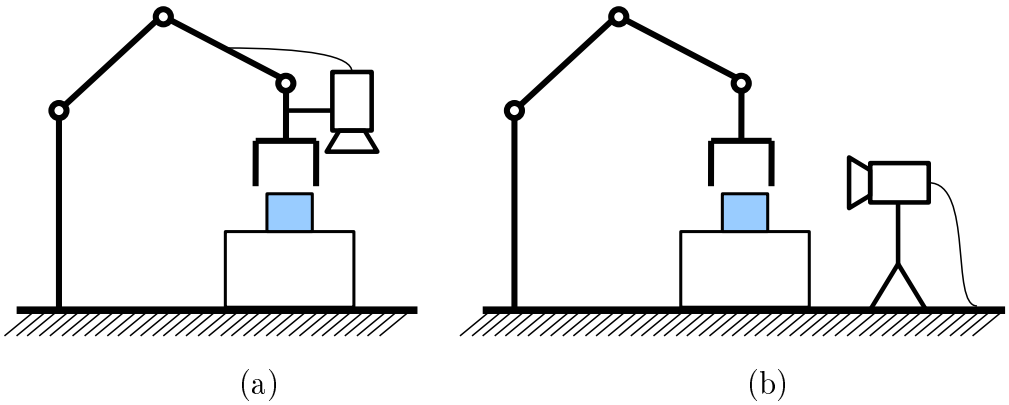


FIGURE 1.3 – Configuration de la caméra : (a) Embarquée ; (b) Déportée.

La seconde configuration possible est que la caméra soit fixée dans l'espace de travail (Fig. 1.3(b)). Dans ce cas, la transformation entre la caméra et la base du robot est fixe et doit être étalonnée pour la raison évoquée ci-dessus. Notons que du point de vue de l'étalonnage, il n'y a guère de différence entre les deux configurations.

Il est important de noter que les lois de commande qui seront développées dans ce chapitre sont valables quelle que soit la configuration choisie. La seule différence est le repère dans lequel la commande est exprimée. Ainsi, le passage d'une configuration à une autre consiste à effectuer un changement de repère du vecteur de commande à partir de la transformation entre la base du robot et le repère de la caméra. Pour des raisons de simplicité, nous allons considérer dans ce qui suit que la caméra est embarquée sur le robot.

1.2 La commande par vision

La commande référencée vision désigne toute commande dont le signal de retour assurant la régulation est obtenu à partir d'images acquises par un capteur optique et, par extension, tout autre capteur susceptible de fournir une image (images par ultra-son, imagerie par résonance magnétique, microscope à balayage électronique, etc.). Ainsi, il existe différentes manières de construire une commande référencée vision en fonction de l'architecture de commande, de l'espace dans lequel les primitives sont définies et de la configuration de la caméra [WSN87, ATYM93].

1.2.1 Le formalisme des fonctions de tâche

Le formalisme privilégié par "l'école française" en asservissement visuel est celui de la commande par fonction de tâche. L'objectif de ce type de commande est de minimiser un critère d'erreur, appelé la fonction de tâche [SELB91, ECR92], qui dépend du signal de

régulation $\mathbf{s}(\mathbf{m}(\mathbf{r}), \xi)$ et de la configuration désirée $\mathbf{s}^*$ qui doit décrire un difféomorphisme dans l'ensemble des poses dans l'espace. La fonction de tâche est généralement définie comme suit :

$$\mathbf{e} = \mathbf{C}\ (\mathbf{s}(\mathbf{r}, \xi) - \mathbf{s}^*(t)) \tag{1.6}$$

où $\mathbf{C}$ est appelée la matrice de combinaison et $\mathbf{s}(\mathbf{r}, \xi)$ représente le signal à réguler.

La matrice de combinaison $\mathbf{C}$ permet de considérer un ensemble de primitives dont la dimension est supérieure au nombre d'éléments du vecteur de commande. Le signal de régulation $\mathbf{s}(\mathbf{r}, \xi)$ dépend de la localisation de la caméra par rapport à la scène observée et des paramètres de la caméra ξ. Pour des raisons de lisibilité nous noterons le signal de régulation $\mathbf{s} = \mathbf{s}(\mathbf{r}, \xi)$ et le signal désiré $\mathbf{s}^*(t) = \mathbf{s}^*$.

L'approche consiste ensuite à calculer le vecteur de commande permettant la minimisation de cette fonction au cours du temps. Dans un asservissement visuel cinématique, le vecteur de commande représente la vitesse que le robot doit exécuter. Par conséquent, afin de calculer le vecteur de commande qui permet d'assurer la décroissance de la fonction de tâche, il suffit de calculer la matrice qui exprime la variation de la fonction de tâche par rapport au vecteur de commande. On obtient ainsi :

$$\begin{aligned} \dot{\mathbf{e}} &= \mathbf{C}\ \frac{d(\mathbf{s} - \mathbf{s}^*)}{dt} + \frac{d\mathbf{C}}{dt}\ (\mathbf{s} - \mathbf{s}^*) \\ &= \mathbf{C}\ \mathbf{L}_s \boldsymbol{\tau} - \mathbf{C}\ \dot{\mathbf{s}}^* + \dot{\mathbf{C}}\ (\mathbf{s} - \mathbf{s}^*) \end{aligned} \tag{1.7}$$

où $\mathbf{L}_s$ est communément appelée "matrice d'interaction". Cette matrice relie la vitesse dans l'espace ($\boldsymbol{\tau}$) aux variations du signal $\mathbf{s}$ extrait de l'image.

Par ailleurs, la matrice de combinaison est généralement choisie de façon à ce que sa variation dans le temps soit nulle (constante) ou négligeable devant la variation du signal de régulation $\mathbf{s}$. La dérivée de la fonction de tâche (1.7) peut dans ce cas être approximée par :

$$\dot{\mathbf{e}} = \mathbf{C}\ \mathbf{L}_s \boldsymbol{\tau} - \mathbf{C}\ \dot{\mathbf{s}}^* \tag{1.8}$$

Comme le système commandé est du premier ordre (commande en vitesse), la réponse généralement souhaitée est une décroissance exponentielle de l'erreur. Ce type de réponse peut être obtenu en introduisant une équation de contrainte sur la décroissance de la fonction de tâche :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \tag{1.9}$$

où λ est un facteur d'amortissement, homogène au temps de réponse, qui permet de régler la vitesse de convergence qui caractérise la réactivité du système.

À partir de (1.8) et (1.9), on obtient l'expression du vecteur de commande :

$$\boldsymbol{\tau} = -\lambda (\mathbf{C}\ \mathbf{L}_s)^{-1}\ \mathbf{e} + (\mathbf{C}\ \mathbf{L}_s)^{-1} \mathbf{C}\ \dot{\mathbf{s}}^* \tag{1.10}$$

Un choix classique de la matrice de combinaison est $\mathbf{C} = \mathbf{L}_s^+$, où $\mathbf{L}_s^+$ représente la pseudo-inverse de la matrice d'interaction $\mathbf{L}_s$. Ce choix permet de linéariser et de découpler le système commandé. Cependant, il n'est pas possible en pratique d'obtenir la valeur exacte de la matrice d'interaction, c'est pourquoi le vecteur de commande est calculé sur la base d'une estimation de cette matrice ($\widehat{\mathbf{L}}_s^+$). On obtient ainsi :

$$\boldsymbol{\tau} = -\lambda \widehat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*) + \widehat{\mathbf{L}}_s^+ \ \dot{\mathbf{s}}^* \tag{1.11}$$

L'équation de fermeture de boucle s'obtient en substituant le vecteur de commande (1.11) dans l'expression de la dérivée temporelle de la fonction de tâche (1.8). On obtient ainsi :

$$\dot{\mathbf{e}} = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{L}_s \mathbf{e} + \widehat{\mathbf{L}}_s^+ \mathbf{L}_s \widehat{\mathbf{L}}_s^+ \ \dot{\mathbf{s}}^* - \widehat{\mathbf{L}}_s \ \dot{\mathbf{s}}^* \tag{1.12}$$

Cette expression représente un système d'équations différentielles du premier ordre avec second membre. La décroissance de la fonction $\mathbf{e}$ (régime transitoire) ne dépend que de la solution de l'équation sans second terme :

$$\dot{\mathbf{e}} + \lambda \widehat{\mathbf{L}}_s^+ \mathbf{L}_s \ \mathbf{e} = 0 \tag{1.13}$$

Comme le facteur d'amortissement est choisi positif, la condition de stabilité locale de la commande est remplie si le produit de la matrice d'interaction réelle par la matrice estimée est défini positif :

$$\widehat{\mathbf{L}}_s^+ \mathbf{L}_s > 0 \tag{1.14}$$

Cette expression peut être utilisée pour l'étude théorique formelle ou numérique (simulations) de la stabilité de ce type de commande. Par ailleurs, l'équation (1.12) montre que l'erreur de suivi n'est nulle que si la dérivée partielle de la fonction de tâche par rapport au temps ainsi que la matrice d'interaction sont parfaitement connues.

1.2.2 Architectures de commande

L'architecture de commande par vision la plus simple est une commande en boucle ouverte sur la mesure par vision qui consiste à comparer les positions estimée et désirée, à estimer le déplacement nécessaire, puis à exécuter le déplacement d'où le nom "look-then-move". Même si cette structure de contrôle est destinée à converger en un pas, il peut être nécessaire en pratique de réexécuter la commande afin de réduire les éventuelles erreurs de positionnement. Cependant, la nature de la commande fait que la période de répétition de la commande est assez importante ($T_c > 0.1$s). Les performances de ce type de commande assez médiocres font que cette approche est devenue archaïque depuis plusieurs années.

Dans l'architecture de commande cinématique ("look-and-move"), qui est la plus largement répandue, le schéma de commande est constitué de deux boucles : haut-niveau et

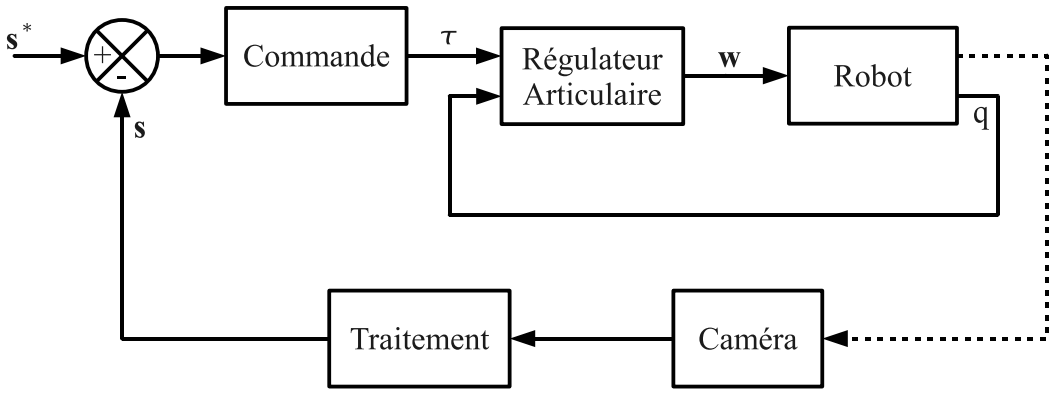


FIGURE 1.4 – Architecture de la commande cinématique "look-and-move".

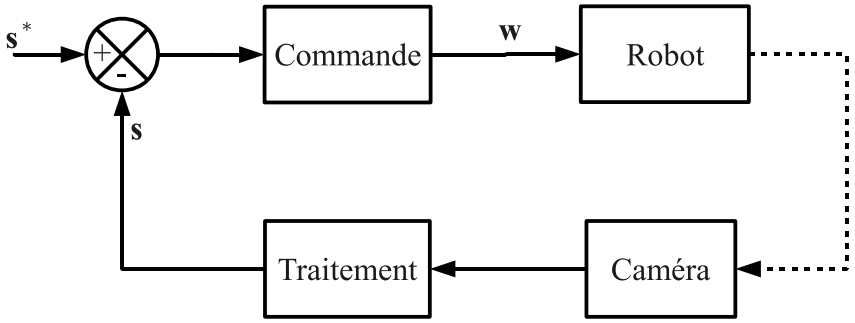


FIGURE 1.5 – Architecture de la commande par retour direct.

bas-niveau (Fig. 1.4). La boucle haut niveau, référencée vision, consiste à calculer un vecteur de commande à partir des mesures par vision. Elle est généralement réalisée à basse fréquence et le vecteur de commande correspond à des vitesses articulaires ou cartésiennes. Le vecteur de commande calculé représente le vecteur d'entrée (consigne) de la boucle bas-niveau. Son rôle est d'asservir le robot sur la vitesse de commande désirée à partir de la mesure articulaire. La fréquence d'échantillonnage de la boucle bas niveau doit ainsi être élevée afin de garantir la stabilité du système.

L'architecture de commande dynamique ("retour direct") est, quant à elle, constituée d'une seule boucle qui permet de calculer directement les couples de commande à partir des mesures effectuées par le capteur de vision et de la configuration désirée (Fig. 1.5). Dans ce schéma de commande les capteurs articulaires sont facultatifs. Cependant, la fréquence d'échantillonnage de la commande dans cette architecture doit être importante ce qui peut poser des problèmes pratiques. En effet, les mesures par vision ne permettent souvent pas d'atteindre les fréquences d'acquisition nécessaires à ce type de commandes.

1.2.3 Type de primitives

La commande par vision est dans la plupart des cas une commande non linéaire. La démarche généralement adoptée dans la synthèse de commande par vision consiste à trouver les primitives ainsi que le jeu de paramètres qui les décrivent en vue de linéariser et de découpler au mieux la boucle de commande. L'objectif de cette démarche est d'obtenir un système linéaire qui permetterait d'utiliser les outils mathématiques de l'automatique linéaire aussi bien pour la synthèse des lois de commande que pour l'analyse de stabilité et de robustesse. Les primitives qui sont le plus souvent utilisées sont classiquement des primitives géométriques, points [ECR92], droites [ECR92, AEH02], cercles [ECR92, MC02b], sphères [TFC09], etc. Par ailleurs, les paramètres qui permettent de les décrire peuvent être des paramètres de position définis dans l'image ou dans l'espace en utilisant une représentations cartésienne mais aussi polaire [IO05], cylindrique [IO05]. Il est même possible de combiner plusieurs jeux de paramètres où chacun est utilisé pour la commande du degré de liberté le plus intéressant en termes de découplage et de linéarité [CH01]. Des paramètres plus particuliers tels que le moment [Cha04, TC05, KI07b] peuvent également être utilisés.

Les primitives visuels peuvent aussi ne pas être purement géométriques. En effet, des paramètres cinématiques obtenus à partir de la dérivée des paramètres géométriques peuvent être utilisés pour la commande [MBG96, CC01b, LD08, MC10]. Dans [MBG96] les paramètres cinématiques sont associés à des paramètres géométriques afin d'améliorer les performances de la commande par vision alors que dans [CC01a, CC01b] seuls les paramètres cinématiques sont utilisés afin d'améliorer la robustesse de la commande dans le cas où les images sont trop variables et bruités pour suivre des primitives tout au long de la commande. Nous reviendrons sur cette approche au chapitre III. De façon plus globale, ces primitives peuvent être considérées comme étant des primitives spatio-temporelles [SJ07]. Nous découvrirons plus en détail au chapitre suivant ce type de primitives.

1.3 Asservissement visuel cinématique

1.3.1 Asservissement visuel 3D

L'asservissement visuel 3D est basé sur l'estimation de la pose (position et orientation) d'une caméra embarquée par rapport à un repère attaché à un objet dans la scène (cible qui peut être statique ou dynamique). Connaissant la position de l'effecteur par rapport au repère lié à l'objet et la jacobienne du robot, les variables articulaires sont commandées de façon à minimiser l'erreur entre la position désirée et la position actuelle de l'effecteur. Cette erreur est ensuite ramenée au repère lié au robot, puis régulée dans l'espace articulaire en utilisant la jacobienne cinématique inverse. La configuration du robot doit dans

ce cas être non singulière.

Dans l'asservissement visuel 3D le signal de référence peut représenter soit la pose relative de la caméra par rapport à la cible (asservissement visuel 3D pose) soit les coordonnées 3D des primitives observées (asservissement visuel 3D point par exemple). Dans ce dernier cas, si la primitive choisie est un point, alors la matrice d'interaction n'est autre que la matrice $\mathbf{L}_{3D}$ donnée par l'équation (1.17). Cependant, quel que soit le signal choisi, définir le signal dans l'espace cartésien à partir de mesures dans l'espace 2D nécessite obligatoirement une reconstruction 3D. En effet, une projection dans un plan implique une perte d'informations liée à la profondeur. Une reconstruction consiste à retrouver l'information perdue soit à partir d'une mesure redondante des primitives en question (grâce à plusieurs caméras où une caméra qui se déplace [HZ00]), soit à partir d'une connaissance à priori de la cible (modèle 3D de l'objet). Notons tout de même que dans l'asservissement visuel 3D, on s'intéresse plus aux choix du signal de régulation et à la façon de l'exploiter qu'à la manière dont le signal est reconstruit. Ainsi, dans l'asservissement visuel 3D pose, différents choix de paramétrages peuvent être utilisés pour décrire la pose de la caméra aussi bien en ce qui concerne la position (coordonnées cartésiennes, sphériques, cylindriques, etc.), qu'en ce qui concerne de l'orientation (angles d'Euler, quaternions, etc.).

Rappelons que le choix du signal se fait, entre autres, par rapport aux propriétés de découplage de la commande. Ainsi, les propriétés du signal dépendent du vecteur de commande qui, lui même, dépend de la structure cinématique du robot. Ainsi, en toute rigueur, le choix du signal de référence devrait tenir compte de la structure du robot. Cependant, pour un robot à 6 degrés de liberté, le vecteur de commande est généralement considéré comme étant le torseur cinématique, qui peut être ensuite traduit en vitesses articulaires en utilisant la jacobienne cinématique inverse du robot. La synthèse d'asservissement visuel 3D pose consiste donc à choisir les paramètres de pose qui présentent les meilleures propriétés de découplage par rapport au torseur cinématique et de calculer la matrice d'interaction qui lui correspond. Par ailleurs, la seconde contrainte liée au choix des paramètres de la pose est que la matrice d'interaction obtenue soit définie et inversible dans tout l'espace de travail.

Si l'on prend comme vecteur de commande le torseur cinématique, alors les paramètres de position les plus intéressants par rapport à ce vecteur de commande sont les coordonnées cartésiennes ${}^{o}\mathbf{t}_{c}$. La dérivée temporelle de ce vecteur n'est autre que la vitesse de translation du vecteur de commande. La matrice d'interaction correspondant à ces paramètres peut alors être obtenue en exprimant la vitesse de l'origine de l'objet par rapport à la vitesse de la caméra exprimée dans le repère de la caméra. On obtient ainsi :

$$ {}^{c}\dot{\mathbf{P}}(t) = -{}^{c}\mathbf{v} - {}^{c}\omega \times {}^{c}\mathbf{OP} \tag{1.15} $$

Cette équation peut s'écrire sous forme matricielle comme suit :

$$ {}^c\dot{\mathbf{P}}(t) = \begin{bmatrix} -\mathbf{I}, & [{}^c\mathbf{OP}]_\times \end{bmatrix} \begin{bmatrix} {}^c\mathbf{v} \\ {}^c\omega \end{bmatrix} = \mathbf{L}_{3D}\ \boldsymbol{\tau} \tag{1.16} $$

avec :

$$ \mathbf{L}_{3D} = \begin{bmatrix} -\mathbf{I}\ , & [{}^c\mathbf{OP}]_\times \end{bmatrix} \tag{1.17} $$

où $\mathbf{L}_{3D}$ représente la matrice d'interaction du point 3D.

Concernant l'orientation, le vecteur de paramètres qui semble être le plus approprié est le vecteur $\theta\mathbf{u}$ [SELB91, Mal98], où θ et $\mathbf{u} = [\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z]^T$ représentent respectivement l'angle et l'axe de rotation. La particularité la plus intéressante de ce vecteur de paramètres, en plus du fait qu'il soit à représentation minimale (3 éléments correspondant à 3 *d.d.l*), et que sa jacobienne par rapport à la vitesse de rotation est définie et inversible dans tout l'espace d'orientation. De plus, comme ce vecteur est le vecteur propre réel de la matrice d'orientation homogène qui lui correspond, il présente des propriétés intéressantes par rapport à cette dernière. Ainsi, la variation du vecteur $\theta\mathbf{u}$ par rapport à la vitesse de rotation ω est donnée par :

$$ \frac{d\theta\mathbf{u}}{dt} = \mathbf{L}_\omega \omega \tag{1.18} $$

avec [Mal98] :

$$ \mathbf{L}_\omega(\mathbf{u}, \theta) = \mathbf{L}_\omega = \mathbf{I} - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\mathrm{sinc}(\theta)}{\mathrm{sinc}^2\left(\frac{\theta}{2}\right)} \right) [\mathbf{u}]_\times{}^2 \tag{1.19} $$

Ainsi, la matrice d'interaction liée à la primitive $\mathbf{s} = \mathbf{r} = \begin{bmatrix} \mathbf{t} \\ \theta\mathbf{u} \end{bmatrix}$ est donnée par :

$$ \mathbf{L}_\mathbf{r} = \begin{bmatrix} -\mathbf{I} & [\mathbf{t}]_\times \\ 0 & \mathbf{L}_\omega \end{bmatrix} \tag{1.20} $$

L'erreur $\mathbf{T}$ entre les configurations courante et désirée de la caméra est calculée à partir des matrices de transformations correspondant aux deux configurations. Par ailleurs, comme la connaissance du modèle de l'objet et des paramètres de la caméra ne peut être parfaite, le calcul de la pose à partir de mesures par vision ne représente qu'une estimation qui est forcément entachée d'erreurs :

$$ \mathbf{T} = {}^c\mathbf{T}_o(\mathbf{s}, \widehat{\xi}, \widehat{\mathbf{M}})\ \left({}^d\mathbf{T}_o(\mathbf{s}^*, \widehat{\xi}, \widehat{\mathbf{M}}) \right)^{-1} = \begin{bmatrix} {}^c\mathbf{R}_d(\mathbf{s}^*, \widehat{\xi}, \widehat{\mathbf{M}}) & {}^c\mathbf{t}_d(\mathbf{s}^*, \widehat{\xi}, \widehat{\mathbf{M}}) \\ 0 & 1 \end{bmatrix} \tag{1.21} $$

Ainsi, l'erreur en translation n'est autre que le vecteur ${}^c\mathbf{t}_d(\mathbf{s}^*, \widehat{\xi}, \widehat{\mathbf{M}})$ alors que le vecteur de rotation correspondant à l'erreur est calculé à partir de la matrice de rotation.

Toujours en choisissant une décroissance exponentielle de l'erreur, le vecteur de commande est donné par :

$$\boldsymbol{\tau} = -\lambda \, \widehat{\mathbf{L}}_{\mathbf{r}}^{+} \, \delta\hat{\mathbf{r}} \tag{1.22}$$

où $\delta\hat{\mathbf{r}}$ est une estimation du vecteur d'erreur entre la configuration courante et désirée calculée à partir de $\mathbf{T}$.

Le problème qui se pose en pratique est la reconstruction euclidienne (indispensable à l'implémentation de la commande). Dans les applications de robotique manipulatrice, il est souvent possible de connaitre le modèle de la cible par rapport à laquelle le robot est positionné et d'exploiter ce modèle pour effectuer une reconstruction euclidienne.

Par ailleurs, comme la transformation entre l'espace projectif et l'espace cartésien n'est pas linéaire, une trajectoire rectiligne dans l'espace cartésien ne correspond pas forcément à une trajectoire rectiligne dans l'image. Il en résulte que les primitives observées peuvent sortir du champ de vue de la caméra ce qui nécessite de prendre des mesures particulières [TMCG02]. De plus, comme les erreurs dans les paramètres du modèle de la cible ou dans ceux de la caméra induisent des erreurs dans la pose estimée, la tâche de positionnement se trouve directement affectée. Afin d'améliorer la précision et la robustesse de la mesure un filtre de Kalman étendu [WHB96] ou des méthodes d'estimation s'inspirant de commande robuste telle que H_∞ [TO00] peuvent être implémentés. Cependant, ces solutions ne permettent pas d'éliminer les erreurs statiques ou dynamiques. En effet, comme la fonction de tâche n'est pas définie dans l'espace de mesure, une erreur nulle de la fonction de tâche ne signifie pas forcément que le robot a atteint la configuration désirée.

1.3.2 Asservissement visuel 2D

L'asservissement visuel 2-D [ECR92] consiste à contrôler les mouvements du robot de façon à déplacer les primitives observées dans l'image, de la position courante vers la position désirée. Dans ce cas, le signal choisi représente la configuration des primitives dans l'image notée $\mathbf{m}$. La fonction de tâche est ainsi donnée par :

$$\mathbf{e} = \mathbf{C} \, (\mathbf{m} - \mathbf{m}^*) + \mathbf{L}_{2D}^{+} \dot{\mathbf{m}}^* \tag{1.23}$$

La matrice d'interaction correspondant à cette primitive peut s'exprimer comme suit :

$$\frac{d\mathbf{m}}{dt} = \frac{\partial \mathbf{m}}{\partial \mathbf{P}} \frac{d\mathbf{P}}{dt} \tag{1.24}$$

où $\dfrac{\partial \mathbf{m}}{\partial \mathbf{P}}$ représente la jacobienne de la position du point dans l'image par rapport à celle

du point dans l'espace et $\dfrac{d\mathbf{P}}{dt}$ représente la matrice d'interaction du point 3D.

Notons qu'en pratique, l'asservissement visuel 2D avec une caméra classique peut aussi bien s'effectuer à partir des coordonnées des points exprimés en pixels qu'à partir des coordonnées des points exprimés dans le plan normalisé. Dans ce dernier cas, les coordonnées des points sont calculées à partir des coordonnées des points dans l'image par :

$$\mathbf{m} = \begin{bmatrix} \frac{1}{\alpha_u} & 0 & -\frac{u_0}{\alpha_u} \\ 0 & \frac{1}{\alpha_v} & -\frac{v_0}{\alpha_v} \end{bmatrix} \tilde{\mathbf{p}} \tag{1.25}$$

À partir de l'équation (1.3), la dérivée temporelle de $\mathbf{m}$ est donnée par :

$$\begin{aligned} \dot{\mathbf{m}} &= \begin{bmatrix} \dfrac{\dot{x}z - x\dot{z}}{z^2} \\ \\ \dfrac{\dot{y}z - y\dot{z}}{z^2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{z} & 0 & -\frac{x}{z^2} \\ 0 & \frac{1}{z} & -\frac{y}{z^2} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{z} & 0 & -\frac{X}{z} \\ 0 & \frac{1}{z} & -\frac{Y}{z} \end{bmatrix} \dot{\mathbf{P}} \end{aligned} \tag{1.26}$$

Puis, en substituant (1.16) dans (1.26) on aboutit à :

$$\dot{\mathbf{m}} = \begin{bmatrix} \frac{1}{z} & 0 & -\dfrac{X}{z} \\ \\ 0 & \frac{1}{z} & -\dfrac{Y}{z} \end{bmatrix} \left[-\mathbf{I}, \quad [{}^c\mathbf{OP}]_\times \right] \dot{\mathbf{P}} \tag{1.27}$$

Enfin, la matrice d'interaction du point 2D est donnée par :

$$\mathbf{L}_{2D} = \begin{bmatrix} -\dfrac{1}{z} & 0 & \dfrac{X}{z} & XY & -(1+X^2) & Y \\ \\ 0 & -\dfrac{1}{z} & \dfrac{Y}{z} & 1+Y^2 & -XY & -X \end{bmatrix} \tag{1.28}$$

À partir de l'équation (1.11) on obtient l'expression de la loi de commande qui permet une décroissance exponentielle de l'erreur :

$$\boldsymbol{\tau} = -\lambda \, \widehat{\mathbf{L}}_{2D}^{+}(\mathbf{m} - \mathbf{m}^*) + \widehat{\frac{\partial \mathbf{e}}{\partial t}} \tag{1.29}$$

Cette commande a pour avantage que l'erreur est calculée dans le même espace que la mesure (le plan de l'image). Ainsi, dans le cas où la consigne a été apprise, des erreurs sur les paramètres intrinsèques ont une influence plus faible sur la convergence du système que dans la commande 3D. Pour cette raison, la commande 2D a la réputation d'être robuste vis-à-vis d'erreurs sur les paramètres de la caméra. En effet, on peut noter à partir de l'équation (1.28) que si $\widehat{\mathbf{L}}_{2D}^{+}$ est de rang plein et pour une configuration non dégénérée du vecteur $\mathbf{m}-\mathbf{m}^{*}$ (le vecteur d'erreur n'étant pas contenu dans le noyau de la matrice $\widehat{\mathbf{L}}_{2D}^{+}$) alors $\boldsymbol{\tau}$ ne peut être nulle que si l'erreur $\mathbf{m}-\mathbf{m}^{*}$ est nulle. La condition de convergence étant définie par l'équation 1.14.

Cependant, le problème qui peut se poser pour l'implémentation d'une telle commande est que la relation entre le signal de régulation et le vecteur de commande n'est pas linéaire et dépend de paramètres à priori inconnus tels que ceux qui décrivent la configuration de la caméra par rapport à la cible observée ou toute autre information non mesurée. Différentes approches ont été proposées afin de pallier à ce problème. L'une d'entre elles consiste à définir la matrice d'interaction comme étant une estimation de la matrice d'interaction à la configuration désirée à partir de laquelle le vecteur $\mathbf{m}^{*}$ a été défini. Cependant, comme la linéarisation de la fonction de tâche n'est que locale, une configuration trop éloignée de la configuration désirée ne permet pas de linéariser de façon exacte le système la commande et donc d'obtenir une décroissance exponentielle de l'erreur. Ceci se traduit par des mouvements non rectilignes des points dans l'image, ce qui peut être nuisible aussi bien au niveau des trajectoires des primitives observées qui peuvent sortir du champ de vue de la caméra qu'au niveau de la stabilité de la commande (condition de convergence non respectée).

1.3.3 Asservissement visuel hybride

L'idée principale de l'asservissement visuel hybride est de s'affranchir de la contrainte auto-imposée de définir tout le signal de commande dans un seul espace. Ainsi, la commande proposée par Malis et *al.* [MCB99], qui se situe entre la commande 2D et 3D, tire profit des avantages de chaque approche. Ainsi, une partie du signal de référence de la commande est définie dans l'espace projectif et l'autre partie dans l'espace cartésien. Cette approche ayant pour objectif d'améliorer les propriétés de linéarisation et de découplage par rapport à un asservissement visuel 2D sans requérir une connaissance précise sur le modèle de la cible comme pour la commande 3D.

En effet, l'une des contraintes principales dans une commande 3D est la connaissance du modèle de la cible observée. Cependant, même sans connaissance à priori de la cible, il est possible de calculer l'orientation relative de la caméra ainsi que le rapport de profondeur à partir de points coplanaires en effectuant un calcul d'homographie [HZ00]. Cette mesure permet donc de contrôler quatre degrés de liberté : l'orientation et la profondeur.

Les deux autres degrés de libertés sont asservis sur la position d'un point dans l'image. Comme l'orientation est engendrée par le calcul d'homographie, elle est indépendante de la translation. L'inverse n'étant pas vrai.

En dehors de la particularité de robustesse et de découplage, cette loi de commande a l'avantage d'être stable. En effet, contrairement à la commande 2D, la stabilité globale de cette loi de commande est démontrée dans tout l'espace de travail pour une caméra parfaitement calibrée [MC02a]. Cependant, cette loi de commande est moins robuste face aux erreurs d'étalonnage que la commande 2D car elle comporte une reconstruction, même partielle (calcul d'homographie). De plus, la commande est plus sensible aux bruits de mesure dans l'image puisque la loi de commande utilise un seul point de référence pour commander la translation. Par ailleurs, tout comme l'asservissement visuel 3D, cette commande ne garantit pas la visibilité de tous les points dans l'image car les rotations sont estimées et contrôlées dans l'espace cartésien.

1.3.4 Dualité entre l'asservissement visuel et l'estimation de pose

L'estimation de pose consiste à retrouver les paramètres extrinsèques de la caméra par rapport à la cible observée à partir des mesures par vision. Pour ce faire, différentes méthodes ont été proposées. L'une des plus simples est sans doute la méthode de Dementhon [DD95]. Celle-ci est parmi les méthodes les moins coûteuses car elle ne nécessite pas d'inversion matricielle. Cependant, cette méthode reste assez limitée en termes de précision et de robustesse. En réalité, il a été démontré que l'asservissement visuel et l'estimation de pose partagent fondamentalement les mêmes formalismes [MC02b, MB05]. En effet, du point de vue mathématique, les deux applications consistent à minimiser un critère lié à des fonctions non linéaires (fonction de tâche ou bien critère d'optimisation). Ainsi, un asservissement visuel peut être vu comme une optimisation non linéaire [PML99] où le vecteur de commande et l'erreur entre la consigne et la mesure représentent respectivement le pas d'avance et le critère à minimiser.

Ainsi, l'approche "Asservissement Visuel Virtuel" [MC02b] consiste à utiliser le formalisme de l'asservissement visuel pour l'estimation de pose. Dans le cas de l'asservissement visuel réel la position courante des primitives dans l'image est mesurée par le capteur de vision à chaque acquisition alors que la configuration désirée des primitives est définie par une ou plusieurs images de référence (voir Fig. 1.6). À l'inverse, dans l'asservissement visuel virtuel (qui est considéré comme étant dual à l'asservissement visuel réel) la position désirée des primitives est mesurée par le capteur dans l'image alors que la position des primitives dans l'image est calculée à partir du modèle de projection. Pour ce faire, les paramètres intrinsèques de la caméra et le modèle de l'objet observé doivent être connus. L'objectif est de calculer le torseur cinématique qui permet de corriger les paramètres de pose. Des expérimentations [MC02b] montrent que les résultats obtenus

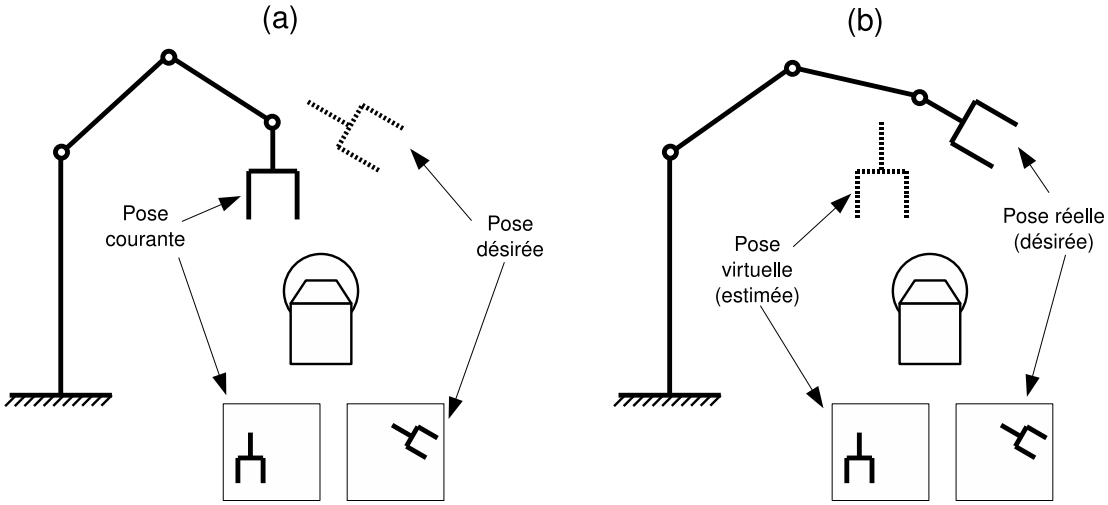


FIGURE 1.6 – Dualité entre les asservissements visuels réel et virtuel. (a) Asservissement visuel réel où l'image acquise correspond à la pose courante et l'image de référence à la pose désirée. (b) Asservissement visuel virtuel où l'image capturée correspond à la pose désirée et les points dans l'image obtenue par projection correspondent à la pose estimée (pose virtuelle courante).

par cette méthode sont plus précis que ceux obtenus par l'algorithme de Dementhon, et comparables à ceux obtenus par calcul de pose en utilisant une optimisation non linéaire classique (Levenberg-Marquardt).

1.4 Asservissement visuel dynamique

Nous avons vu que dans les schémas de commandes cinématiques la fréquence d'échantillonnage de la boucle haut-niveau (mesure par vision) est relativement faible (25-60 Hz) par rapport à la fréquence de la boucle bas-niveau ($\sim>$ 100 Hz). De plus, les temps d'exposition, de transmission et de traitement induisent une certaine latence sur la mesure effectuée par vision [LHW04]. Cependant, si la boucle de vision (haut-niveau) souffre d'un temps de latence, la boucle bas-niveau, en revanche, n'est pas affectée par ce phénomène. Par ailleurs, la boucle bas-niveau interprète la vitesse de commande comme étant une erreur dans la configuration qu'elle tente de corriger à partir des mesures articulaires. On se rend ainsi compte que le déphasage entre la boucle haut-niveau (mesure par vision) et la boucle bas-niveau (mesure articulaire) peut rendre la commande instable pour des vitesses élevées ou des temps de latences trop longs [ZLWS03]. Ainsi, la bande passante de la commande cinématique se trouve finalement limitée à cause du temps de latence [VAC+02]. Par ailleurs, la décroissance exponentielle généralement choisie dans les commandes cinématiques n'est pas la réponse la plus rapide que l'on puisse espérer (convergence à l'infini).

Ajouter à cela l'imbrication des boucles qui se traduit par l'accumulation des erreurs de suivi fait que cette structure de commande n'est pas la plus appropriée pour améliorer les performances de la commande par vision.

Cette réflexion nous conduit à la commande dynamique qui consiste à calculer directement les couples de commande à partir des mesures par vision. Cependant, cette commandes est encore plus sensible au temps de latence que la commande cinématique [WSN87]. En effet, dans [CG96] une étude comparative entre les différentes lois de commandes : commande en couples, vitesses, positions et position+intégrateur, face à une entrée échelon sur une caméra fixée à un système ayant deux *d.d.l* (tangage et lacet) a été présentée. La conclusion de cette étude montre que la commande la plus performante en termes de temps de réponse est la commande en position+intégrateur, contrairement à la commande en couples qui possède le temps de réponse le plus important. Cela est dû à la fréquence d'acquisition qui est trop faible (50Hz) pour une commande en couples ce qui impose de réduire les gains de la commande afin de conserver la stabilité du système. En contre-partie, des gains faibles ne permettent pas d'assurer une réponse rapide ni de rejeter les perturbations dues à la dynamique non linéaire du système. Afin de réduire les effets du temps de latence en asservissement visuel, un schéma de commande avec un bloc prédictif ajouté à la boucle de retour est proposé dans [CG96]. Cependant, si cette approche permet, en effet, de réduire le temps de latence, elle a l'inconvénient d'injecter du bruit dû aux erreurs de prédiction dans la boucle de commande. Ainsi, pour améliorer les performances de la commande par vision il est nécessaire :

- de connaître aussi bien la dynamique du robot que le modèle de la caméra ;
- d'augmenter la fréquence d'échantillonnage qui doit être comprise au moins entre 4 et 20 fois la fréquence du système en boucle fermée ;
- de réduire le temps de latence qui est principalement dû aux temps d'acquisition, de transmission et de traitement des données.

La démarche proposée dans [CG96] est d'abord de réduire le temps de latence autant que possible puis, sur la base de la modélisation du robot et de la caméra, d'élaborer le compensateur qui permet d'obtenir la dynamique désirée.

Une première synthèse de la problématique des performances de la commande par vision se résume à un dilemme. D'une part, la commande cinématique ne requiert pas une fréquence d'échantillonnage importante mais possède des performances limitées. D'autres part, la commande dynamique permet, en principe, d'obtenir de meilleures performances mais à condition que la mesure par vision ait une période d'acquisition et un temps de latence suffisamment faibles. Ainsi, la première approche consiste à adapter la commande par vision (cinématique et dynamique) à la fréquence d'acquisition faible de la vision, alors que la seconde tente plutôt d'adapter les systèmes de vision aux besoins de la commande dynamique. Nous allons aborder dans ce qui suit les deux solutions.

1.4.1 Commande pour la vision

Un des points les plus importants dans l'amélioration des performances de la commande cinématique et de pouvoir réduire ou même éliminer le déphasage qui existe entre la boucle haut-niveau (vision) et la boucle bas-niveau (mesures articulaires). Pour ce faire, il faut soit réduire le retard dans la boucle de vision, tâche qui n'est pas simple, soit introduire du retard dans la boucle de régulation bas-niveau. Dans [ZLWS03] c'est la seconde solution qui a été retenue. En effet, dans la commande proposée dans cet article, le retard de la mesure par vision est estimé puis injecté dans la boucle bas-niveau. Les mesures articulaires sont ainsi mémorisées et décalées d'un temps équivalent à la latence de la boucle de vision afin de synchroniser les deux boucles haut-niveau et bas-niveau. Une telle opération a pour effet de déplacer le retard de la boucle de régulation vers l'entrée du système. Le résultat est que la stabilité de la commande est théoriquement garantie quels que soient les gains. Cependant, l'inconvénient majeur est que la consigne est exécutée avec un retard équivalent au temps de latence de la boucle de commande ce qui peut introduire des erreurs considérables dans des tâches de suivi.

Dans [GdM03] une commande prédictive (General Predictive Control GPC) est proposée pour la commande d'un robot sériel à 6 *d.d.l* sur la base d'une structure de commande "look-and-move" dans laquelle la boucle bas-niveau reçoit des consignes en vitesse. Dans ce schéma de commande l'acquisition (images de 640×240 pixels à 120 Hz) et le traitement prennent un temps équivalent à deux périodes d'échantillonnage. Ce retard est modélisé et compensé via une commande prédictive. Par ailleurs, afin d'améliorer les performances de la commande, un modèle dynamique simplifié du robot, représenté par une matrice diagonale (système découplé) qui fait office de matrice d'inertie calculée à l'état courant à partir des mesures articulaires et qui néglige de ce fait les effets des forces de Coriolis et centrifuge ainsi que la force de gravité, a été proposé. Les résultats montrent que le temps de réponse pour cette commande est amélioré par rapport à une commande ayant un PID comme régulateur bas-niveau.

Cette approche a été utilisée dans une application médicale pour la compensation du mouvement cardiaque [GGdM$^+$05]. Ainsi, avec une mesure par vision de spot laser et une anticipation du mouvement obtenue grâce à l'électrocardiogramme (ECG) qui permet d'anticiper le mouvement du cœur, le mouvement cyclique du cœur a pu être en grande partie compensé afin d'améliorer le positionnement relatif du robot par rapport à ce dernier. Dans cette application, la dimension des images acquises a été réduite (256×256) afin d'augmenter la fréquence d'échantillonnage qui atteint 500Hz. Dans [CGdMF06], une comparaison entre cette approche et une commande H_∞ avec une prise en compte des flexibilités montre que la commande robuste (H_∞) permet d'obtenir des performances comparables à la commande prédictive avec un vecteur de commande plus lisse qui permet de moins exciter les actionneurs et les modes de vibrations du robot.

Une commande dynamique écrite dans l'espace de l'image appliquée à un dirigeable a été proposée dans [ZP99]. L'asservissement proposé dans cet article est basé sur un seul point pour la commande de deux degrés de liberté ce qui permet d'écrire les équations de la dynamique dans l'espace de l'image en fonction de la position de la primitive visuelle (un point). La dynamique est ensuite compensée dans un schéma de régulation PID. Cependant, cette méthode n'est valide que si un seul point est observé et ne permet pas d'utiliser la redondance de mesure ou d'étendre la méthode à d'autres degrés de liberté. Par ailleurs, étant donné que le système est très peu réactif (dirigeable) le temps de latence dû à la mesure par vision n'est pas très nuisible contrairement à une application à un système plus réactif (robot manipulateur par exemple).

Une approche de commande par backstepping pour la commande dynamique d'un système sous-actionné du type hélicoptère dans l'image a été proposée dans [HM02] où la dynamique complète du système commandé est considérée. Ainsi, une commande virtuelle a pour objectif de réduire l'erreur dans l'image en calculant une vitesse de commande virtuelle que la commande réelle par backstepping doit réguler. Cependant, le système commandé a besoin d'une mesure des vitesses de translation et de rotation. En l'absence d'une telle mesure avec une précision suffisante pour le type de système considéré, seules des simulations ont pu être présentées. Cependant, les auteurs attirent l'attention sur le fait que la fréquence et la latence des systèmes de vision n'a pas été prise en compte, chose qui doit l'être dans un système réel.

Une des solutions les plus intéressantes de commande dynamique a été présentée dans [KCN+00]. La commande proposée permet de compenser la dynamique du robot à partir des mesures par vision et des mesures articulaires avec une connaissance à priori de la profondeur des primitives. Même si la commande est à retour direct, le schéma de commande est tout de même multifréquence en raison des deux retours articulaires et visuels. Ainsi, la correction de position est obtenue par vision à une fréquence de 50Hz alors que la vitesse est corrigée à partir des mesures articulaires actualisées toutes les 2,5ms. La stabilité locale de la commande a été démontrée et les expérimentations faites sur un robot plan à 2 *d.d.l.* asservi sur un point à distance fixe de la caméra montrent la validité de la commande.

Cependant, étant donné que la fréquence d'échantillonnage de la mesure par vision est faible, la bande passante de la commande reste limitée ce qui impose de faibles gains qui donnent lieu à des erreurs statiques importantes (~ 4 pixels). On peut aussi observer des oscillations autour de la position désirée. L'explication suggérée par les auteurs mais non argumentée est que ces oscillations sont dues aux frottements secs ou aux distorsions radiales. L'hypothèse que ces oscillations soient dues aux corrections de la position et de la vitesse, qui se font à des fréquences différentes et à partir de mesures différentes, n'est pas évoquée même si elle semble être plus plausible. Cette approche a inspiré quelques travaux qui ont repris la même structure de commande dont l'une qui utilise la méthode de "energy

shaping plus dumping injection". La méthode est démontrée comme étant stable sous l'hypothèse que les erreurs de modèle de l'objet et de la caméra soient négligeables. Par ailleurs, le temps de latence dû au système de vision qui représente l'effet potentiellement le plus déstabilisant n'a pas été pris en compte dans cette étude. Une version de commande adaptative par rapport à la charge portée par le manipulateur a été proposée dans [NC03].

Du point de vue de l'automatique, la caméra a aussi une certaine "dynamique" qui n'est pas due à sa masse ou à son inertie mais au processus d'acquisition lui-même. En effet, dans les faibles vitesses de mouvement, on se contente de considérer l'aspect géométrique du modèle de projection de la caméra faisant intervenir les principes de l'optique géométrique. Cela suppose que le mouvement de l'objet observé est négligeable devant le temps d'exposition, ce qui est souvent le cas. Cependant, les capteurs de vision ont, pour la plupart, besoin d'un certain temps pour la capture d'une image. Ce temps d'intégration peut introduire des déformations dans l'image acquise dans le cas où la vitesse relative de l'objet observé par rapport à la caméra est importante par rapport au temps d'exposition [LD08].

Dans [RCGVS07] il est proposé de prendre en compte la forme des primitives observées dans l'image ainsi que le temps d'exposition du capteur afin de disposer d'un modèle plus réaliste de la caméra. Ainsi, l'instant d'acquisition considéré ne correspond pas à la fin d'exposition mais au moment de demi-période d'exposition du capteur. La fonction de transfert obtenue à partir de ce modèle est prise en compte dans la synthèse du régulateur (de type RST). Les expérimentations sur la commande d'un moteur ultrasonique (piézo-électrique) ont montré que l'implémentation de ce modèle permet d'améliorer les performances de la commande aussi bien en termes de stabilité qu'en termes de temps de réponse.

Un autre problème lié à la vision rapide est d'ordre photogrammétrique. En effet, la quantité de lumière accumulée durant la période d'exposition du capteur est proportionnelle à la sensibilité du capteur, à l'intensité de la lumière et au temps d'accumulation. S'il n'y a pas moyen de jouer sur la sensibilité du capteur et l'intensité de la lumière, la réduction du temps d'exposition dans un système de vision rapide a pour effet d'assombrir l'image. Comme le rapport signal/bruit dans les images sombres est assez faible, il est assez difficile d'extraire des points caractéristiques tels que les points de Harris [HS88]. Dans [KI07b] il est proposé une méthode de suivi d'objets 3D connus à partir du calcul du moment de la forme de la projection de l'objet dans l'image. Après initialisation de la pose, le suivi se fait de façon itérative grâce au calcul de la matrice d'interaction qui représente la variation du moment par rapport aux paramètres de pose. On peut, cependant, observer un effet de dérive pendant le suivi. Dans [OOY+03] afin d'augmenter la visibilité des images (1280×512) acquises par un système de vision à 1kHz, une deuxième image de plus faible résolution est reconstruite à partir de l'originale, où chaque pixel de l'image résultante représente la somme d'une matrice de pixels 4×4 de l'image initiale.

La résolution de l'image obtenue est ainsi de 512×128 mais la valeur de chaque pixel a été en moyenne multipliée par 16. Cette approche permet, entre autre, d'agir comme un filtre moyenneur contre le bruit de mesure.

Afin d'éviter la complexité de ce type d'approche (compensation du retard, approche multifréquences), d'autres travaux ont porté sur le développement de systèmes d'acquisitions plus rapides.

1.4.2 Vision pour la commande

La démarche proposée dans [CG96] afin d'adapter les systèmes de vision aux systèmes dynamiques consiste à augmenter la fréquence d'acquisition des images et de réduire le temps de latence dû aux temps d'exposition, de transmission et de traitement des données. En réalité, les capteurs existants dans le marché sont capables d'acquérir des images à des fréquences très élevées [EPK+03]. Cependant, ce qui limite l'exploitation de telles vitesses d'acquisitions est la bande passante limitée qui représente un goulot d'étranglement entre le capteur et l'unité de traitement (Fig. 1.1). Afin d'éviter ce phénomène, plusieurs approches ont été proposées dans la littérature. L'une d'entre elles, qui est assez originale, consiste à utiliser des dizaines de caméras miniatures dont la fréquence d'acquisition est assez faible (30 Hz) pour simuler l'acquisition d'une caméra rapide [WJV+05]. Ainsi, la fréquence équivalente de l'acquisition séquentielle d'une scène par 52 caméras est équivalente à une fréquence d'acquisition de $52 \times 30 = 2500$ Hz. Cependant, même si l'approche est intéressante, l'entrelacement temporel entre les images acquises par les différentes caméras dû au temps d'exposition qui est finalement supérieur à la fréquence d'acquisition équivalente ainsi que la variation de l'angle de vue de la scène observée par les différentes caméras, dont les paramètres intrinsèques sont à priori différents, font que l'émulation d'une caméra rapide n'est certainement pas parfaite, ce qui conduit à d'autres problématiques.

Une approche plus classique consiste à faire un compromis entre la fréquence d'acquisition et la résolution de l'image. Dans [ATS+09], un système de vision à 120 Hz avec une résolution de 640×220 a été utilisé dans une commande cinématique de robot parallèle. La même configuration du système de vision avait été adoptée dans l'implémentation d'une commande prédictive d'un robot manipulateur [GdM03]. Dans [CGdMF06], comme l'application de robotique chirurgicale ne nécessitait pas un large champ de vue, la résolution a encore été baissée par la même équipe à (256×256) afin d'augmenter la fréquence d'acquisition qui atteint 500Hz. Cependant, dans des applications où l'espace de travail est plus important, une faible résolution se traduirait par une per te de précision sur la mesure.

Une autre approche pour éviter le goulot d'étranglement consiste à rapprocher l'unité de traitement du capteur grâce à de l'électronique embarquée [IMT92, KIIY03]. Le

système est basée sur une architecture parallèle ce qui permet d'atteindre des fréquences d'acquisition élevées 1kHz mais avec une résolution assez faible (16×16 pixels [INI96] et 128×128 pixels [NITM00, CRVG+04]). L'inconvénient majeur de ce système est que la résolution de l'image est trop faible. De plus, à cause de la méthode de transmission et de traitement des images, la résolution du capteur ne peut être augmentée sans procéder à des modifications matérielles importantes. Ce mode d'acquisition a été utilisé dans des applications de suivi dans l'image [NITM00] et de saisie d'objets en vol par une main artificielle [FNTI06] et de commande d'un robot poly articulé grâce à un système stéréo [SNI04]. Cependant, la résolution faible des images acquises limitent considérablement le rapport entre la précision et la largeur du champ de vue.

Dans [MDP07] des algorithmes de compression ont été utilisés pour la transmission d'images de hautes résolutions (1280×1024) à une fréquence de 500 images seconde à travers une interface de communication USB. Cependant, l'utilisation de compression avec perte afin d'atteindre le taux de compression souhaité représente une source de dégradation de l'image d'autant plus importante que le taux de compression est élevé, ce qui peut détériorer, par la même occasion, la précision sur la détection des primitives dans l'image. Dans un contexte de reconstruction de forme par lumière structurée où il n'y a pas besoin d'avoir des images en niveau de gris, les pixels des images acquises ont été codées sur 1 bit afin d'augmenter la fréquence d'acquisition [TKH07].

Nous pouvons constater que les méthodes qui ont été proposées s'inscrivent dans deux approches globales. La première consiste à réduire la quantité d'informations à transmettre, ce qui se traduit par la détérioration du signal utile, alors que la deuxième consiste à effectuer les traitements le plus près possible de l'image. Cependant, dans l'une comme dans l'autre méthode, l'information circule dans le même sens *c.-à-d.* du capteur vers l'unité de traitement. Une autre manière plus efficace de faire mieux collaborer le traitement haut et bas-niveau serait de faire circuler l'information dans les deux sens ce qui permet de voir le problème sous un autre angle.

En effet, la solution de vision rapide qui semble être la plus intéressante dans une application de commande haute performance à été proposée dans [URLP04]. Ce système de vision intègre un FPGA auprès du capteur CMOS (1024×1024) pour permettre l'accès à une partie des images acquises. Ainsi, grâce à une communication avec une unité de traitement, seules les régions d'intérêt de l'image sont transmises ce qui permet de réduire la quantité de données à transmettre sans pour autant perdre les informations utiles. Cette optimisation de la bande passante de l'interface de communication (USB) permet d'augmenter la fréquence d'acquisition qui atteint 2500 Hz pour des dimensions de fenêtres de 16×16 pixels. Ce mode d'acquisition par fenêtre d'intérêt sera développé plus en détail dans le chapitre suivant.



(a) (b)

FIGURE 1.7 – Exemples de robots à structure cinématique parallèles. (a) Plate-forme de Gough-Stewart ; (b) Le robot Quattro.

1.5 Vision pour la commande des robots parallèles

Les robots parallèles sont des mécanismes à structure fermée où la plate-forme, qui représente le support de l'élément opérationnel (outil-final), est reliée à la base du robot par plusieurs chaines cinématiques (Fig. 1.7). Cette structure mécanique particulière leur confère des particularités intéressantes par rapport aux robots à chaine cinématique ouverte (robots sériels Fig. 1.8). En effet, pour le même nombre de degrés de libertés, le nombre d'articulations dans les robots parallèles est plus important que pour un robot sériel. Ceci permet d'une part d'avoir un plus grand nombre de solutions technologiques, et d'autre part, d'obtenir une plus forte interaction entre les éléments du robot et la plate-forme grâce auxquels des couplages géométriques, cinématiques et dynamiques peuvent être élaborés afin d'obtenir le comportement souhaité.

Une première conséquence à ces particularités est que, contrairement aux robots sériels, un robot parallèle à plusieurs degrés de liberté peut être conçu en gardant les actionneurs solidaires au bâti, chose qui ne peut se faire dans un robot sériel sans perdre considérablement en rigidité [3]. Ainsi, libérer les actionneurs et les jambes de la charge que représentent les actionneurs permet de réduire considérablement la masse transportée et donc d'améliorer le rapport entre la charge utile et la masse du robot ainsi que les performances dynamiques du système. De fortes accélérations, non égalées à ce jour par des robots sériels, peuvent ainsi être atteintes [NdlORC+05].

Par ailleurs, dans les robots à chaine ouverte, l'outil n'est attaché qu'à travers une seule articulation. Ainsi, les efforts appliqués sur l'outil se traduisent par des moments de

3. Les actionneurs peuvent aussi être fixés dans le cas sériel mais cela nécessite de concevoir des éléments de transmission (câbles, chaines, arbre, etc.) ce qui peut considérablement détériorer la rigidité du robot.



(a) (b)

FIGURE 1.8 – Exemples de robots à chaine ouverte simple. (a) Le Motoman ; (b) Le robot Scara.

torsions importants qui peuvent se traduire par d'importantes flexions. En revanche, le fait que la plate-forme dans les robots parallèles soit supportée par plusieurs jambes permet de répartir les efforts appliqués par les différents éléments ce qui lui confère une meilleure rigidité par rapport aux robots sériels. De plus, les robots parallèles permettent de générer des mouvements en rotation exclusivement à partir d'articulations prismatiques. Ceci permet de concevoir des systèmes où les efforts supportés par les jambes sont principalement des efforts de traction et de compression permettant ainsi d'améliorer d'autant plus la rigidité (Fig. 1.9).

Cependant, les robots parallèles ont aussi des défauts. En effet, sur le plan géométrique, l'espace de travail des robots parallèles est généralement plus faible par rapport à l'encombrement du robot que pour les robots sériels. De plus, une configuration des articulations actives correspond souvent à plusieurs configurations possibles de la plate-forme correspondant aux différents modes d'assemblage du robot [Mer90, Hus96]. Dans ce cas, il n'est pas possible de connaitre avec certitude la configuration de la plate-forme exclusivement à partir des mesures articulaires. Le robot peut même changer de mode d'assemblage sans que cela ne puisse être observé [CW98]. Par ailleurs, nous savons que le passage d'un mode d'assemblage à un autre implique le passage par une singularité. Ainsi, l'existence de plusieurs modes d'assemblages implique qu'il existe autant de configurations singulières que de frontières entre les différents modes d'assemblage. Par ailleurs, il est bien connu que la commandabilité des robots de manière générale décroit à proximité des singularités



FIGURE 1.9 – Exemple d'utilisation d'un robot parallèle (plate-forme de Gough-Stewart) dans un simulateur de vol.

ce qui peut amener à réduire encore plus l'espace de travail

L'existence d'un ensemble important de singularités signifie aussi qu'il peut exister d'importantes variations d'efforts sur les jambes dans l'espace de travail en fonction de la configuration du robot. Ainsi, les efforts n'étant pas forcément appliqués dans le sens de déplacement de la plate-forme [TZR99], le déplacement où l'application d'efforts de cette dernière sur certaines directions peut nécessiter des efforts beaucoup plus importants au niveau des actionneurs et des jambes, ce qui peut être néfaste du point de vue de la rigidité [Pri02] et contraignant pour le dimensionnement des actionneurs et des jambes. Un dernier point, qui n'est pas le moindre, est que les contraintes de fermetures de boucles peuvent nécessiter dans certaines structures de robot un usinage et un montage très précis des pièces au risque de ne pas respecter les contraintes géométriques de fermeture de boucle, ce qui se traduirait pas des efforts internes très nuisibles à la structure mécanique du robot et à la commande.

Si les avantages des structures parallèles sont théoriquement indiscutables, la pratique montre que leurs performances dynamiques en terme de précision ne sont pas toujours meilleures que pour des robots sériels [WM93, TZR99, Pri02, BWY06]. En réalité, si la commande des robots sériels a atteint une certaine maturité, la commande des robots parallèles n'est qu'au stade de développement. Dans cette phase transitoire, les commandes qui sont aujourd'hui implémentées sur les robots parallèles ont été initialement développées pour les robots sériels [PAM09]. Aussi, et en considération des différences qui existent entre ces deux types d'architectures, il parait évident que ces commandes ne sont pas très bien adaptées aux spécificités des robots parallèles.

Ainsi, la raison principale pour laquelle ces commandes inadaptées aux robots parallèles est l'absence de modèle géométrique direct formel, sauf pour certains robots particuliers [Gog04, CW03]. Il devient par conséquent difficile de calculer les modèles cinéma-

tique et dynamique à partir des mesures articulaires. Se pose ainsi la question de l'espace de régulation de la commande des robots parallèles. En effet, la mesure dans l'espace articulaire ne semble pas pertinente car la configuration du robot est mieux définie dans l'espace cartésien [PAM09, DAMM06].

Une commande articulaire ne nécessite pas d'estimation de la pose de la plate-forme dès lors que la dynamique n'est pas prise en compte. Cependant, négliger la dynamique ou considérer un modèle simplifié [LSCH03] limite les performances de la commande surtout pour les robots parallèles, qui présentent un fort couplage dynamique [BWY06, DH06, OZW02]. Ainsi, les performances dynamiques ne sont pas uniformes dans l'espace de travail [BG05].

Différents travaux montrent que l'espace de commande le plus approprié pour les robots parallèles est l'espace cartésien [DC99, KI07a, CPP06, PAM09]. L'implémentation d'une telle commande à partir de mesures articulaires nécessiterait l'estimation numérique de la pose de la plate-forme à la fréquence de la commande. Cependant, les éventuelles erreurs de suivi et de convergence sont plus nuisibles dans cette commande car, contrairement à la commande dynamique articulaire où la pose n'est utilisée que pour la linéarisation du système, cette commande utilise la pose comme signal de régulation. Ainsi, une erreur d'estimation se traduit directement par une erreur de positionnement. De plus, l'estimation nécessite d'autant plus d'itérations que la précision recherchée est importante, ce qui peut être encore plus coûteux en termes de temps de calcul. Quel que soit l'espace de régulation, l'utilisation du modèle géométrique direct introduit inévitablement des erreurs aussi bien en raison d'erreurs de modèles qui, dans tous les cas, admettent des hypothèses simplificatrices qui ne sont pas nécessairement vérifiées (rigidité parfaite, jeu inexistant dans les liaisons ou les réducteurs des moteurs, etc.) [WM93] que d'erreurs de paramètres estimés [Pri02, ZR95]. Tout cela fait que la mesure articulaire n'est pas pertinente pour les robots parallèles. Ainsi, la dualité qui peut exister entre les robots sériels et parallèles [WH91, Bru99] suggère que l'espace de mesure devrait être l'espace d'état du robot qui est plus lié à la configuration de la plateforme qu'à la configuration articulaire.

La figure 1.11 montre ainsi la commande dynamique en couple calculé avec mesure cartésienne. On peut voir dans ce schéma qu'il n'y a nul besoin de passer vers la configuration articulaire, contrairement à la commande référencée mesure articulaire (Fig. 1.10). Ainsi, le couple qui permet de compenser la dynamique du système est donné par le modèle dynamique inverse [KI07a] exprimé dans l'espace cartésien [PAM09] :

$$\Gamma = \mathbb{D}^T \left(\mathbb{M}\mathbf{w} + \mathbb{C}(\mathbf{r}, \boldsymbol{\tau}) + \mathbb{G}(\mathbf{r}) + \sum_{j=1}^{k} \mathbb{J}_{jp}^T \mathbb{J}_j^{-1} \mathbb{H}_j \right) \tag{1.30}$$

où :

– $\mathbf{w}$: le vecteur de commande, la pose et la vitesse de la plateforme du robot ;

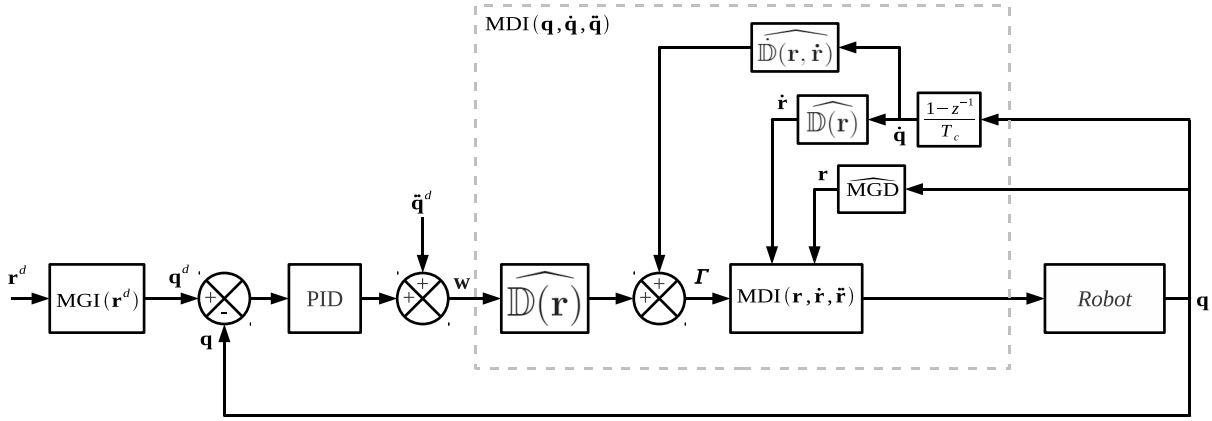


FIGURE 1.10 – Schéma de commande en couple calculé de robots parallèles dans l'espace articulaire avec mesure articulaire.

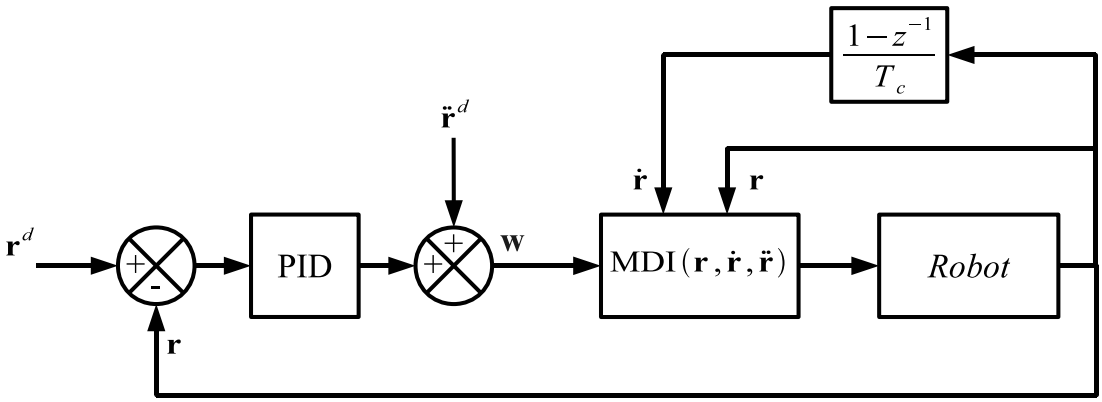


FIGURE 1.11 – Schéma de commande en couple calculé de robots parallèles dans l'espace cartésien avec mesure de position et de vitesse cartésienne [PAM09].

- $\mathbb{D}$: le modèle cinématique direct du robot ;
- $\mathbb{M}$: la matrice d'inertie du robot ;
- $\mathbb{C}(\mathbf{r}, \boldsymbol{\tau})$: les forces de Coriolis et centrifuges et $\mathbb{G}(\mathbf{r})$ l'attraction gravitationnelle ;
- $\mathbb{H}_j$: le modèle dynamique inverse de la $j^{\text{ème}}$ jambe ;
- $\mathbb{J}_{jp}$: la matrice jacobienne entre la dernière liaison de la jambe et le repère de la plateforme et $\mathbb{J}_j^{-1}$ la jacobienne inverse de la jambe $j^{\text{ème}}$.

L'implémentation de ce type de commande pose le problème de la mesure de la configuration de la plate-forme dans l'espace cartésien. Ceci peut être fait en utilisant des capteurs extéroceptifs. Les solutions technologiques existantes sont les capteurs de position linéaire, les interféromètres laser et les capteurs de vision. L'avantage des capteurs de position linéaire c'est leur précision. Ils sont cependant encombrants, ce qui peut réduire encore plus l'espace de travail ou provoquer des auto-collisions si leur position n'est pas prise en compte dans la génération de trajectoire. Les interféromètres laser représentent aussi un moyen de mesure de distance très précis. Cependant, s'ils sont capables de mesurer la position d'un point avec une précision micrométrique, ils ne permettent pas de mesurer directement l'orientation. Cette dernière ne peut s'obtenir qu'à partir de la position de trois points ce qui nécessite trois appareils de mesures. Étant donné le coût de ce moyen de mesure, cette solution n'est pas viable du point de vue économique. La dernière solution, qui semble être la plus intéressante pour les applications de commande est la mesure par vision. En effet, ce moyen de mesure est largement utilisé dans la commande des robots manipulateurs ainsi que dans celle des robots mobiles.

Dans [PLA+08] une commande dynamique d'un robot parallèle par vision a été implémentée. Cette commande utilise une mesure extéroceptive de la pose et de la vitesse de l'outil finale pour la compensation de la dynamique et de la régulation du système. Ainsi, le schéma de commande devient plus simple à partir du moment où la tâche et le modèle dynamique sont définis dans l'espace cartésien (Fig. 1.11). Cependant, afin d'assurer les calculs à la fréquence de commande requise (400 Hz [4]), l'estimation de la pose de l'effecteur basée sur l'algorithme de Dementhon [DD95] est très bruitée [5], ce qui perturbe considérablement la commande. De plus, la vitesse de l'effecteur est d'autant plus bruitée qu'elle est obtenue par dérivation numérique de la pose à une fréquence relativement importante. Une solution pour atténuer le bruit de mesure est de filtrer la vitesse estimée, mais le déphasage introduit par le filtre entre les vitesses réelle et estimée introduit un retard sur la vitesse estimée ce qui perturbe la commande au point de la rendre instable.

Aussi, les principaux critères à respecter dans la synthèse de commande dynamique par vision sont :

4. La commande initialement implémentée est une commande dynamique référencée modèle qui tourne à 400Hz.

5. Rappelons que l'algorithme de Dementhon tourne généralement plus rapidement que les méthodes basées sur une optimisation non linéaire car il ne nécessite pas d'inversion matricielle. Cependant, il est moins performant car il minimise une erreur algébrique plutôt qu'une erreur géométrique.

- une latence et un déphasage minimaux ;
- des fréquences d'acquisition et de commande élevées ;
- une estimation faiblement bruitée de la pose et de la vitesse (en évitant la dérivation numérique).

En ce qui concerne ce dernier point, une méthode de calcul de pose et de vitesse à partir d'une seule image et sans dérivation numérique a été présentée dans [AAALM06]. Cette méthode est basée sur une acquisition d'images par une caméra "Rolling Shutter" dont les pixels ne sont pas exposés au même instant. Cela signifie que pour un objet (ou caméra) en mouvement, la projection des primitives visuelles se fait à des poses différentes de l'objet. Dans ce cas, le modèle de projection d'un ensemble de primitives lié à un objet rigide dépend de la pose de l'objet par rapport à la caméra au moment d'exposition des pixels coïncidant avec la projection des primitives observées. En faisant des hypothèses sur le mouvement de l'objet (vitesses constantes dans ce cas) les paramètres précédents peuvent être déterminés de manière assez précise grâce à une minimisation non linéaire des erreurs de reprojections. Ainsi, cette approche permet de satisfaire le dernier point des critères cités ci-dessus. Cependant, l'acquisition à fréquence faible (30Hz) et l'implémentation hors ligne de l'algorithme d'estimation font que le système présenté dans cette publication ne permet pas de satisfaire les deux premiers critères.

1.6 Conclusion

Les principaux aspects de la commande par vision ont été abordés dans ce chapitre. Ainsi, du point de vue de la structure de contrôle, nous avons vu qu'il existe essentiellement deux types de commande qui sont la commande cinématique et la commande dynamique. Le schéma de commande cinématique, même s'il est le plus répandu, présente des déficiences en termes de performances qui sont dues : au temps de latence, à la fréquence d'acquisition qui est faible, à la vitesse de convergence plutôt lente et à la dynamique du système qui n'est généralement pas compensée. Par ailleurs, la commande dynamique est potentiellement la plus intéressante en termes de performance, mais bien plus exigeante en termes de période d'acquisition et de retard qui doivent être encore plus faibles que pour une commande cinématique. Néanmoins, nous avons pu voir que la mesure par vision pouvait être très rapide ce qui laisse à penser qu'il est possible de réaliser une commande dynamique performante par vision.

Par ailleurs, nous avons vu que l'une des applications les plus intéressantes de l'asservissement visuel dynamique est la commande des robots parallèles pour lesquels la mesure de la plate-forme dans l'espace cartésien représente une mesure plus pertinente que la mesure articulaire. Cette mesure permettrait, d'une part, de simplifier le schéma de commande et, d'autre part, de rendre la commande robuste face aux erreurs de modèle du robot. Cependant, les résultats publiés dans [PLA+08] montrent que la commande dynamique

par vision peut être très bruitée à cause, entre autre, de l'estimation de vitesse. Néanmoins, les travaux de [AAALM06] montrent qu'il est possible d'avoir une estimation assez précise de la pose et de la vitesse sans avoir recours à une dérivation numérique, à partir de mesures non instantanées de primitives visuelles. Si l'on associe à cela la méthode d'acquisition par fenêtres d'intérêts proposée dans [URLP04], que l'on peut adapter afin d'obtenir une acquisition non simultanée de ces fenêtres, alors nous obtenons le noyau à partir duquel des méthodes d'estimation de mouvements rapides ainsi que d'asservissements visuels, cinématiques et dynamiques 2D et 3D peuvent être développées. Du moins, c'est ce que l'on se proposera de démontrer dans la suite de ce document.

Chapitre 2

Estimation de pose et de vitesse par vision rapide

Ce chapitre présente des méthodes d'estimation de pose et de vitesse par vision rapide basées sur l'acquisition séquentielle de fenêtres d'intérêt. Cette approche permet d'augmenter la vitesse d'acquisition en réduisant la quantité de données à transmettre tout en améliorant, par la même occasion, la densité d'information dans les données transmises. Cependant, comme l'acquisition des différentes primitives dans l'image n'est pas simultanée, ce mode d'acquisition introduit des déformations dans l'image reconstruite. Ces distorsions sont modélisées pour être exploitées dans l'estimation de pose et de vitesse.

2.1 Introduction

Dans la plupart des systèmes de vision temps-réel, la contrainte de temps ne permet pas d'effectuer le traitement de tous les pixels de l'image acquis afin d'extraire les informations utiles. Cela impose de prédire les régions de l'image où l'information est supposée se trouver. La sélection des fenêtres d'intérêt se fait sur la base des dernières images acquises. Par ailleurs, dans le cas où une ou plusieurs primitives sortent de la zone de recherche (à cause d'un déplacement rapide de la cible par exemple), le suivi de la primitive dans l'image est perdu car la contrainte temps-réel du système ne permet pas de rechercher la primitive dans toute l'image (en raison du temps de calcul nécessaire à l'accomplissement de cette tâche). Néanmoins, une solution [SIT07] consiste à enregistrer toutes les images acquises dès que l'objet observé est perdu. Durant l'enregistrement, un algorithme de recherche est lancé dans l'image où le suivi à été perdu. Dès que l'objet est retrouvé, le suivi reprend sur les images enregistrées jusqu'à atteindre la dernière image acquise. Cependant, même si cette méthode est intéressante du point de vue de la vision car elle permet d'utiliser les parties de l'image autres que les régions d'intérêt, le traitement non temps-réel (images enregistrées) la rend inappropriée pour des applications de commande. Au



(a) (b)

FIGURE 2.1 – Acquisition de primitives visuelles dans la scène par sélection de régions d'intérêt. (a) Image entière acquise par une caméra standard. (b) Fenêtres d'intérêt acquises et replacées sur un fond gris.

final, on peut retenir que dans la plupart des commandes par vision, la plus grande partie des données images transmise par la caméra n'est pas traitée et donc finalement inutile. Par ailleurs, sachant que la bande passante de l'interface de communication est limitée, la transmission des données inutiles ralentit considérablement la fréquence d'acquisition du système et augmente le temps de latence.

L'une des solutions pour améliorer les performances des systèmes de vision est d'utiliser les ressources en bande passante de manière plus efficace. Pour cela, il suffit d'inverser certaines étapes du processus d'acquisition. En effet, au lieu d'envoyer les données et ensuite sélectionner une zone de recherche dans l'image, la position des fenêtres d'intérêt (FI) à acquérir est d'abord sélectionnée, et ensuite, le contenu des imagettes est transmis vers l'unité de traitement (Fig. 2.1). En fait, ce mode d'acquisition peut très bien être réalisé en utilisant des caméras du marché (comme, par exemple, la série TrackCam de la société Photon focus). Du point de vue du traitement, n'aquérir que des parties de l'image contenant l'information ne représente pas de différence particulière par rapport à l'acquisition classique. Par contre, du point de vue de la fréquence d'acquisition, la période d'échantillonnage peut être divisée par le rapport entre le nombre de pixels envoyés (contenant l'information utile) et la taille de l'image entière. Considérons par exemple le cas de l'acquisition d'images contenant 10 points avec une caméra de résolution totale 1024×1024 pixels. En codant les pixels sur 1 Octet, la taille de chaque image est de $1024 \times 1024 = 1$ MOctets. Si les informations extraites des images sont les coordonnées des points dans l'image représentées par deux nombres à virgule flottante codés sur 2×4 Octets (ce qui est plus que suffisant étant donné la précision de détection des points dans l'image), alors le rapport entre l'information utile et la quantité de données transmise ne

représente que :

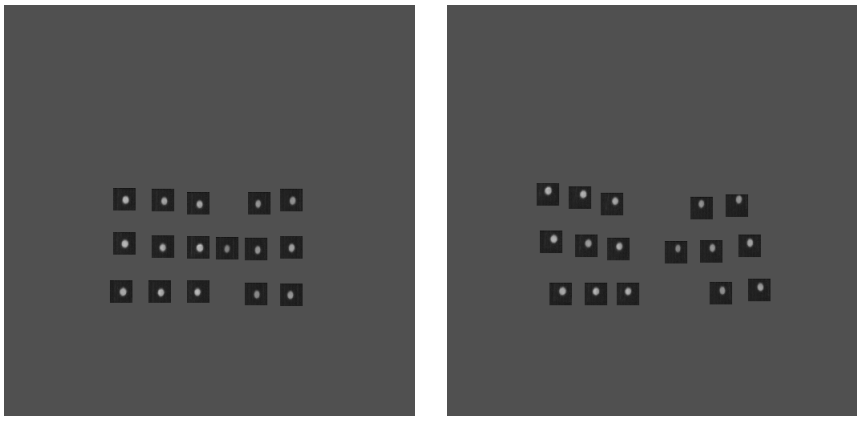
$$\eta = \frac{2 \times 4 \times 10}{1024 \times 1024} = 7,6\ 10^{-6} \tag{2.1}$$

En revanche, en ne transmettant que des fenêtres d'intérêt de 10 pixels × 10 pixels contenant les points, la quantité de données transmises pour la même information utile est de $(10 \times 10) \times 10 = 1000$ Octets. Ainsi, la quantité de données à transmettre est réduite d'un facteur de plus de 1000. Le rapport entre l'information utile et l'information transmise η' est alors multiplié par le même facteur :

$$\eta' = \frac{2 \times 4 \times 10}{(10 \times 10) \times 10} = 0.08 \tag{2.2}$$

Avec une liaison CameraLink qui peut atteindre un débit de $2,28$ Gbits/s les fréquences d'acquisition maximales que peuvent théoriquement atteindre les deux systèmes sont respectivement ~ 0.3 kHz et 300 kHz. En pratique les fréquences réelles sont plus faibles en raison des bits de contrôle du bus de données et des performances temps-réel de l'unité de traitement. Par ailleurs, il faut noter que dans le cas classique, toutes les primitives sont contenues dans une seule image alors que dans le second cas une fenêtre d'intérêt (FI) ne contient qu'une seule primitive. Ainsi, la fréquence d'acquisition des données (F_e) dans ce dernier cas dépend de la fréquence d'acquisition (F_a) des FI et du nombre de points $F_e = \frac{F_a}{n}$.

Nous verrons dans ce chapitre comment ces primitives "spatio-temporelles", qui permettent théoriquement d'augmenter la fréquence d'acquisition, peuvent être utilisées pour l'estimation de la pose et de la vitesse d'objets rapides. Pour cela, nous allons tout d'abord présenter le modèle de projection associé à ce mode d'acquisition. Ensuite, deux approches d'estimation de pose et de vitesse seront proposées. La première utilise une optimisation non linéaire de type Newton-Raphson associée au modèle de projection de ces primitives visuelles où le calcul de la matrice d'interaction nécessaire à la mise en œuvre temps-réel de cet algorithme sera présenté. La seconde est basée sur l'approche par filtrage de Kalman étendu associé au modèle de projection classique où nous verrons comment le mode d'acquisition proposé peut améliorer le temps d'exécution par rapport au mode d'acquisition classique (image globale) en réduisant le nombre d'opérations induites par le processus d'estimation. Enfin, une comparaison entre les deux méthodes d'estimation proposées sera présentée.



(a) (b)

FIGURE 2.2 – Distorsions dans l'image induites par une acquisition séquentielle de fenêtres d'intérêt d'un objet rigide en mouvement.

2.2 Modélisation

2.2.1 Acquisition séquentielle de fenêtres d'intérêt

Contrairement à la méthode d'acquisition classique où tous les pixels de l'image sont acquis au même moment, l'acquisition par régions d'intérêt se fait de manière séquentielle [1], avec une seule FI à la fois, ce qui introduit un délai entre les différentes FI. Dans le cas où la scène est statique, ce mode d'acquisition n'a pas d'effet particulier sur l'image obtenue. Par contre, du fait que les primitives ne sont pas capturées au même instant, l'acquisition de primitives sur une scène dynamique (objets où caméra en mouvement) introduit des distorsions dans l'image (Fig. 2.2). Du fait de ces distorsions, le modèle de projection perspective devient invalide car l'image obtenue ne correspond (sauf dans des cas singuliers) à aucune pose valide de l'objet rigide. Les méthodes d'estimation classiques du type moindre carrés deviennent inutilisables à moins de prendre en compte ces distorsions [AAALM06]. L'introduction de ces distorsions peut être vue comme un effet indésirable dû à ce mode d'acquisition. Cependant, en plus du fait que l'acquisition sélective de FI permet d'augmenter la fréquence d'acquisition, les distorsions dans l'image apportent une information supplémentaire par rapport à une acquisition classique.

En effet, dans [AAALM06] les distorsions qui apparaissent dans des images prises par une caméra "Rolling Shutter" (où l'image est acquise ligne par ligne) sont exploitées pour le calcul de la pose et de la vitesse d'un objet en mouvement. Bien que l'acquisition ne soit pas rapide (60Hz) et le traitement hors ligne, la méthode proposée présente l'avantage d'être précise car le modèle de projection proposé, qui respecte la rigidité de l'objet et

1. Les caméras disponibles sur le marché ne permettent d'acquérir qu'une seule FI à la fois. Cependant, des caméras destinées à la recherche ont été développées pour permettre l'acquisition de plusieurs régions d'intérêt simultanément [URLP04].

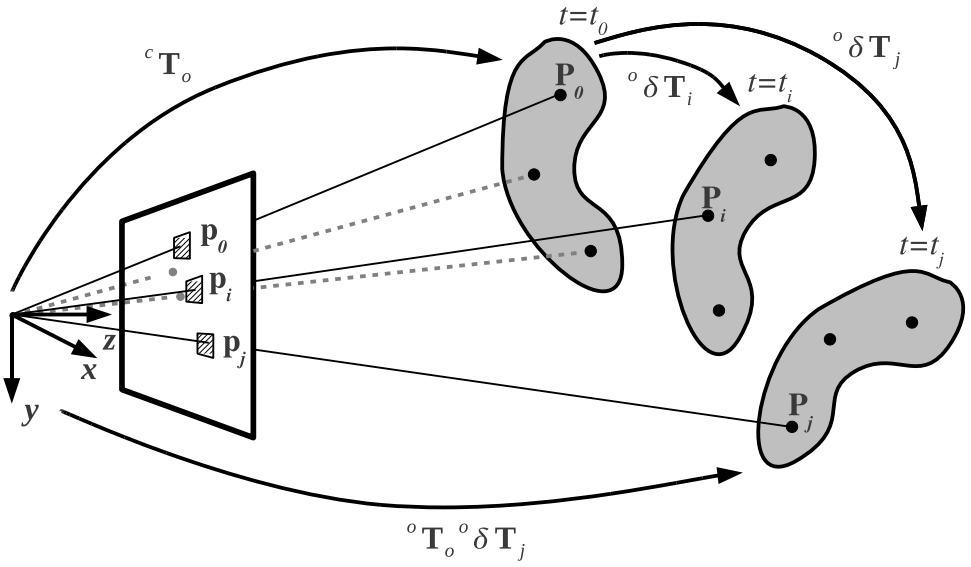


FIGURE 2.3 – Projection séquentielle de points appartenant à un objet rigide en mouvement.

l'estimation de la vitesse ne fait pas intervenir de dérivation numérique.

2.2.2 Modèle de projection

Dans le cas où la projection des points n'est pas simultanée (voir Fig. 2.3), la configuration de l'objet par rapport à la caméra lors de l'acquisition des différents points n'est pas la même si la cible ou la caméra est en mouvement (Eq. 2.3). Ainsi, en relâchant la contrainte d'acquisition simultanée, l'équation (1.1) peut s'écrire de façon plus générique :

$$\forall i = 1..n, \quad {}^{c}\tilde{\mathbf{P}}_i = {}^{c}\mathbf{T}_o(t_i)\ {}^{o}\tilde{\mathbf{P}}_i \tag{2.3}$$

où ${}^{c}\mathbf{T}_o(t_i)$ représente la matrice de passage entre l'objet et la caméra à l'instant d'acquisition de la primitive ${}^{o}\tilde{\mathbf{P}}_i$.

L'équation (2.3) peut être réécrite en décomposant la matrice ${}^{c}\mathbf{T}_o(t_i)$ en une partie constante qui dépend de la pose de l'objet à l'instant de référence t_0 et une partie qui dépend de l'instant d'acquisition du i^e point. Cette décomposition s'écrit différemment selon que le mouvement concerne la caméra ou l'objet.

Dans une configuration caméra déportée où l'objet est en mouvement, les coordonnées des points dans le repère de la caméra sont données par :

$$\forall i = 1..n, \ {}^{c}\tilde{\mathbf{P}}_i = {}^{c}\mathbf{T}_o\,{}^{o}\delta\mathbf{T}_i\ {}^{o}\tilde{\mathbf{P}}_i \tag{2.4}$$

où ${}^{o}\delta\mathbf{T}_i$ est la matrice de transformation décrivant le déplacement de l'objet entre les

instants t_0 et t_i exprimée dans le repère de l'objet à $t = t_0$.

Dans le second cas, où la caméra est en mouvement, le point ${}^c\tilde{\mathbf{P}}_i$ est obtenu par :

$$\forall i = 1..n,\ {}^c\tilde{\mathbf{P}}_i = {}^c\delta\mathbf{T}_i\ {}^c\mathbf{T}_o\ {}^o\tilde{\mathbf{P}}_i \tag{2.5}$$

où ${}^c\delta\mathbf{T}_i$ est la matrice représentant le déplacement dans le repère caméra.

D'après les équations (2.4) et (2.5), dans une acquisition non simultanée, les coordonnées du point 3D dans le repère de la caméra dépendent de la configuration de l'objet à un instant de référence (t_0) et du déplacement de l'objet à l'instant d'acquisition t_i. La forme de cette matrice est donc déterminée par le modèle de mouvement adopté.

2.2.3 Modèle du mouvement

Il existe plusieurs approches à la modélisation du mouvement apparent d'un objet rigide observé par une caméra. Dans le cas général où le modèle de l'objet est inconnu, le mouvement des primitives est modélisé dans le plan image. Cependant, la rigidité de l'objet impose des contraintes supplémentaires au modèle de mouvement. En effet, lorsque le nombre de points sur un objet rigide est supérieur à 3, ce qui est souvent le cas, le nombre de degrés de libertés (*ddl*) des primitives dans l'image devient plus important que le nombre de *ddl* maximal de l'objet rigide (6 *ddl*). Afin de respecter cette contrainte, des modèles de mouvements qui conservent les distances entre les primitives dans l'espace ont été proposés [Sub89, Ber92]. Cependant, vu que les modèles proposés sont des approximations polynomiales des mouvements apparents, la contrainte de rigidité n'est pas complètement respectée mais juste approximée. Cette approximation est suffisante quand elle est utilisée pour la détermination des directions de mouvement dans l'image. Mais dans notre cas, c'est le modèle de projection lui même qui dépend du mouvement et une approximation de la rigidité introduirait des erreurs de reprojection qui pourraient être nuisibles à l'estimation et à la commande.

2.2.3.1 Intégration du mouvement d'un corps solide dans l'espace

Afin de respecter la contrainte de rigidité de l'objet, l'intégration du mouvement des points dans l'espace doit respecter la structure spécifique de SE(3). Pour cela, on intègre le mouvement du repère mobile décrit par son torseur cinématique. Cependant, dans l'hypothèse de vitesse constante sur une courte durée, le torseur cinématique du repère mobile s'intègre différemment selon que l'on considère le torseur cinématique constant dans le repère absolu (fixe) ou dans le repère mobile. Supposer que le torseur cinématique est constant dans le repère absolu signifie que la direction aussi bien que les modules des vecteurs vitesses de translation et de rotations sont constants dans ce même repère. Ce mouvement correspondrait par exemple au déplacement d'un robot cartésien dans lequel

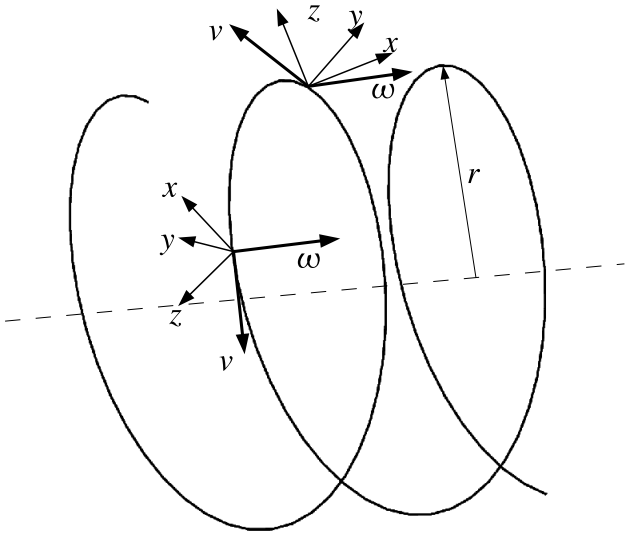


FIGURE 2.4 – Intégration du mouvement d'un repère mobile dont la vitesse est constante dans son propre repère

les axes de rotations de l'effecteur sont concourants. Étant donné que la vitesse de translation n'influe pas sur la vitesse de rotation et inversement, les vitesses de translation et de rotation sont considérées comme étant découplées. Ainsi, le mouvement de translation et de rotation du repère mobile s'intègrent indépendamment l'un de l'autre ce qui simplifie considérablement les calculs. Dans le second cas où le torseur cinématique est supposé constant dans le repère mobile, seul le module de la vitesse de translation est constant (sauf quand la vitesse de rotation est nulle). En effet, comme la vitesse de translation est constante dans le repère de l'objet, alors la direction de celle-ci évolue avec la rotation de l'objet. Les vitesses dans ce cas sont couplées. Même si ce modèle est plus complexe, il peut se révéler plus approprié pour la description de certains types de mouvements [PC05].

2.2.3.2 Modèle de vitesse constante dans le repère mobile

Soit $\mathbf{F}(t)$ un repère en mouvement par rapport au repère absolu. Dans le premier cas où l'on considère que la vitesse de translation est portée par le repère mobile (Fig. 2.4), les équations du mouvement de l'objet peuvent être déterminées en intégrant la vitesse instantanée de l'objet. Pour ce faire, on utilise les formalismes de l'algèbre de Lie.

Soit $\mathbf{T}$ un élément du groupe spéciale euclidien $SE(3)$. Ce groupe représente l'ensemble des matrices de transformations homogènes. Il est acquis que l'ensemble $SE(3)$ forme un groupe de Lie différentiable de dimension 6. L'espace tangent à $SE(3)$ définit l'espace des vecteurs de translation et de rotation instantanés $\boldsymbol{\tau} = \left(\mathbf{v}^T, \omega^T\right)^T$ représentés sous forme

matricielle par :

$$\lceil \boldsymbol{\tau} \rceil = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{v} \\ 0 & 0 \end{bmatrix} \tag{2.6}$$

où la notation $\lceil \bullet \rceil$ représente les crochets de Lie et $[\bullet]_\times$ la matrice antisymétrique associée au vecteur $\bullet$.

La relation entre la vitesse de déplacement et la dérivée de la matrice $\mathbf{T}$ est donnée par :

$$\lceil \boldsymbol{\tau} \rceil = \mathbf{T}^{-1}\dot{\mathbf{T}} \tag{2.7}$$

Le déplacement de l'objet pendant un intervalle de temps $\Delta t = t - t_0$ est obtenu en intégrant la vitesse instantanée sur l'espace tangent à $SE(3)$. Cette intégrale s'exprime en fonction de l'exponentielle matricielle définie par :

$$\delta\mathbf{T} = e^{\lceil \Delta t \ \boldsymbol{\tau} \rceil} = \sum_{k=0}^{\infty} \frac{1}{k!} \lceil \Delta t \ \boldsymbol{\tau} \rceil^k = \begin{bmatrix} \delta\mathbf{R} & \delta\mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{2.8}$$

où $\delta\mathbf{R} = \mathbf{e}^{\left(\Delta t [\boldsymbol{\omega}]_\times\right)} \in SO(3)$ est la matrice de rotation donnée par la formule de Rodrigues :

$$\delta\mathbf{R} = \mathbf{I} + \frac{\sin(\Delta t \, \|\omega\|)}{\|\omega\|} [\boldsymbol{\omega}]_\times + \frac{\sin^2\left(\dfrac{\Delta t \, \|\omega\|}{2}\right)}{\dfrac{\|\omega\|^2}{2}} [\boldsymbol{\omega}]_\times^2 \tag{2.9}$$

et $\delta\mathbf{t}$ le déplacement en translation :

$$\delta\mathbf{t} = \frac{1}{\|\omega\|^2} \left((\mathbf{I} - \delta\mathbf{R})(\omega \times \mathbf{v}) + \Delta t \ \omega\omega^T \mathbf{v} \right) \tag{2.10}$$

2.2.3.3 Modèle de vitesse constante dans le repère absolu

Comme expliqué précédemment, considérer la vitesse de translation constante dans le repère absolu revient à découpler la translation de la rotation (Fig. 2.5). Étant donné que toutes les composantes des deux vecteurs sont constantes sur la période d'intégration Δt, le déplacement se calcule de façon triviale :

$$\delta\mathbf{t} = \int_0^{\Delta t} \mathbf{v} \ dt = \mathbf{v} \ \Delta t \tag{2.11}$$

$$\delta\mathbf{R} = e^{\lceil \Delta t \ \omega \rceil} \tag{2.12}$$

Le choix du modèle de mouvement utilisé dépend des propriétés du mouvement réellement observé. Cependant, dans le cas général où l'on n'a pas de connaissance à priori sur le mouvement réel nous adopterons le modèle de mouvement le plus simple *c.-à-d.* le modèle

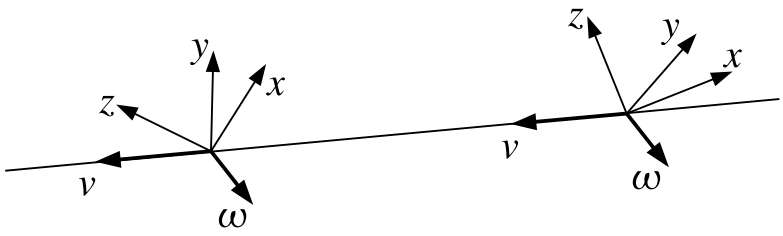


FIGURE 2.5 – Intégration du mouvement d'un repère mobile dont la vitesse est constante dans le repère absolu.

à vitesse constante dans le repère immobile.

2.2.3.4 Modèle de mouvement à accélération constante

Nous avons proposé ci-dessus deux modèles de mouvements, tous deux représentent des approximations au premier ordre du mouvement réel. Afin d'améliorer la précision du modèle, deux approches sont concurrentes. La première consiste à réduire le temps d'acquisition et de traitement en sélectionnant un modèle simple qui permettrait de réduire les coûts en calculs alors que la seconde consiste à augmenter la précision du modèle pour un intervalle de temps donné. La question dans ce cas est "est-ce qu'on gagne en précision plus que ce l'on perd en temps ?" et inversement. Il n'existe probablement pas de réponse absolue à cette question. Mais on peut considérer que dans certaines applications il est plus pertinent de considérer une approximation au second ordre (accélération constante) du mouvement observé. En effet, dans le contexte qui nous intéresse, la commande dynamique entre autre, la sortie du régulateur est une accélération qui est traduite en couple par le modèle dynamique du robot. Ce couple de commande est appliqué au robot durant la période d'échantillonnage de la commande, ce qui laisse penser que l'accélération est plus ou moins constante durant cet intervalle de temps. La fréquence d'acquisition des FI étant plus importante que celle de la commande, l'hypothèse d'accélération constante serait plus réaliste. Mais comme l'intégration de l'accélération en rotation induit des calculs lourds, un bon compromis entre la simplicité et la précision du modèle de mouvement serait de considérer l'accélération en translation constante avec un mouvement en rotation accéléré [2].

Pour les raisons évoquées au §2.2.3.3, nous exprimons l'accélération en translation dans le repère absolu. Les expressions des vitesses en fonction des accélérations sont alors

2. Axe de rotation constant avec une vitesse de rotation uniformément variée.

données par :

$$\mathbf{v}(t) = \mathbf{a}\Delta t + \mathbf{v}(t_0) \tag{2.13}$$

$$\omega(t) = \gamma\Delta t\mathbf{u}_\omega + \omega(t_0) \tag{2.14}$$

où $\mathbf{a} = [\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z]^T$ est le vecteur d'accélération en translation et γ la norme de l'accélération angulaire dont le vecteur porteur est défini par l'axe de la vitesse de rotation $\mathbf{u}_\omega = \dfrac{\omega}{\|\omega\|}$.

Les déplacements en translation et en rotation du repère mobile sur le même intervalle de temps Δt, s'écrivent dans le repère absolu de la manière suivante :

$$\delta\mathbf{t} = \frac{1}{2}\mathbf{a}\Delta t^2 + \mathbf{v}\Delta t \tag{2.15}$$

$$\delta\mathbf{R} = \exp(\frac{1}{2}\gamma\mathbf{u}_\omega\Delta t^2 + \omega\Delta t) \tag{2.16}$$

Une propriété intéressante de ce modèle de mouvement, comme nous allons le voir par la suite, est que sa dérivée par rapport aux paramètres de la pose et de la vitesse se calcule de la même manière que dans le modèle de mouvement à vitesse constante.

2.2.3.5 Intégration du mouvement en fonction du modèle cinématique du robot

Dans les modèles de mouvement précédents nous avons considéré le cas général où les mouvements sont libres. En pratique l'objet ou la caméra serait porté par un système robotisé. Dans ce cas, il serait intéressant de tenir compte de la structure cinématique du système porteur dans l'intégration du mouvement.

En particulier, pour les robots dont le nombre de *ddl* est inférieur à 6 il serait, à priori, plus adéquat de réduire le nombre de paramètres décrivant le mouvement. Ces paramètres peuvent éventuellement correspondre aux variables articulaires du robot. Dans une approche de commande dynamique, les consignes envoyées au robot sont des consignes d'accélérations articulaires appliquées au robot pendant l'intervalle de temps T_c défini par la période d'échantillonnage de la commande. Ainsi, le robot agit comme un double intégrateur dans l'espace articulaire. Par conséquent, il serait théoriquement plus pertinent de proposer un modèle de mouvement basé sur l'intégration des accélérations articulaires :

$$q_i(t) = \iint_{t_0}^{t} \ddot{q}_i \; dt = \frac{1}{2}\ddot{q}_i\Delta t^2 + \dot{q}_i(t_0)\Delta t + q_i(t_0) \tag{2.17}$$

où $q_i(t)$ représente la configuration de la i^e articulation à l'instant t.

La configuration de l'objet ou de la caméra porté par le robot exprimée dans le repère

de base à l'instant t est ainsi calculée à partir du modèle géométrique direct :

$$^{b}\mathbf{T}_{m} = MGD(q(t))\ \mathbf{W} = MGD(\frac{1}{2}\ddot{q}_i\Delta t^2 + \dot{q}_i(t_0)\Delta t + q_i(t_0))\ \mathbf{W} \tag{2.18}$$

où $\mathbf{W}$ est la matrice de transformation entre l'objet ou la caméra et le repère lié à l'effecteur.

L'avantage de ce modèle, qui peut aussi être vu commme un inconvénient, est que le déplacement et sa dérivée par rapports aux paramètres de mouvement se calculent respectivement par le MGD et le MCD.

Cependant, pour les robots parallèles le calcul du MGD n'est souvent pas trivial [Mer09]. En réalité ce problème ne se pose pas vraiment. En effet, le choix des paramètres articulaires n'est pas limité aux articulations actives. De ce fait, un ensemble de paramètres adéquat peut être choisi de façon à pouvoir décrire sans ambiguïté la configuration de l'effecteur et à pouvoir en dériver le modèle cinématique. Le second problème posé par la réduction du nombre de degrés de libertés est la validité des hypothèses relatives au modèle. En effet, des erreurs d'estimation sur les paramètres géométriques ou des imperfections liées à la mécanique du robot introduirait systématiquement des erreurs statiques (espace des configurations de l'outil) et dynamiques dans le modèle de mouvement qui se répercuteraient directement sur le modèle de projection. C'est pourquoi l'utilisation de cette approche nécessiterait une bonne connaissance du modèle du robot.

2.2.3.6 Modèle de projection

Nous avons vu dans la section 2.2.2 que le modèle de projection diffère selon la configuration du système de vision par rapport au robot.

Ainsi, d'après l'équation (2.4), dans une configuration de caméra déportée, la matrice de passage entre l'objet et la caméra à l'instant $t_i = t_0 + \Delta t_i$ est donnée par :

$$\begin{aligned} ^{c}\mathbf{T}_{o}(t_i) &= {}^{c}\mathbf{T}_{o}(t_0)\,{}^{o}\delta\mathbf{T}_{i} && (2.19)\\ &= \begin{bmatrix} {}^{c}\mathbf{R}_{o} & {}^{c}\mathbf{t}_{o} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{o}\delta\mathbf{R}_{i} & {}^{o}\delta\mathbf{t}_{i} \\ 0 & 1 \end{bmatrix} && (2.20)\\ &= \begin{bmatrix} {}^{c}\mathbf{R}_{o}\,{}^{o}\delta\mathbf{R}_{i} & {}^{c}\mathbf{R}_{o}\,{}^{o}\delta\mathbf{t}_{i} + {}^{c}\mathbf{t}_{o} \\ 0 & 1 \end{bmatrix} && (2.21) \end{aligned}$$

Noter que le fait de représenter les déplacements $({}^{o}\delta\mathbf{R}_{i}, {}^{o}\delta\mathbf{t}_{i})$ dans le repère de l'objet (2.21) laisse penser qu'il existe un couplage entre la translation et la rotation. C'est pourquoi il est plus adéquat de représenter les déplacements dans le repère où sont représentés les

paramètres décrivant le mouvement (*c.-à-d.* le repère caméra) :

$$
\begin{aligned}
{}^{c}\mathbf{R}_{o}\,{}^{o}\delta\mathbf{t}_{i} &= {}^{c}\delta\mathbf{t}_{i} && (2.22)\\
{}^{c}\mathbf{R}_{o}\,{}^{o}\delta\mathbf{R}_{i} &= {}^{c}\mathbf{R}_{o}\,{}^{c}\mathbf{R}_{o}^{T}\,{}^{c}\delta\mathbf{R}_{i}\,{}^{c}\mathbf{R}_{o} = {}^{c}\delta\mathbf{R}_{i}\,{}^{c}\mathbf{R}_{o} && (2.23)
\end{aligned}
$$

En substituant ces deux dernières équations dans (2.21) on obtient :

$$
{}^{c}\mathbf{T}_{o}(t_i) = \begin{bmatrix} {}^{c}\delta\mathbf{R}_{i}\,{}^{c}\mathbf{R}_{o} & {}^{c}\delta\mathbf{t}_{i} + {}^{c}\mathbf{t}_{o} \\ 0 & 1 \end{bmatrix} \tag{2.24}
$$

En effet, cette forme de la matrice de passage révèle bien le découplage entre la translation et la rotation. Cela était prévisible car une rotation de l'objet par rapport à l'origine de son repère ne modifie en rien les coordonnées de son origine.

Les coordonnées du point 3D sont ainsi données par :

$$
{}^{c}\mathbf{P}_{i} = {}^{c}\delta\mathbf{R}_{i}\ {}^{c}\mathbf{R}_{o}\mathbf{P}_{o} + {}^{c}\delta\mathbf{t}_{i} + {}^{c}\mathbf{t}_{o} \tag{2.25}
$$

avec :

$$
\begin{aligned}
{}^{c}\delta\mathbf{t}_{i} &= \frac{1}{2}\ {}^{c}\mathbf{a}\ \Delta t_{i}^{2} + {}^{c}\mathbf{v}\ \Delta t_{i} && (2.26)\\
{}^{c}\delta\mathbf{R}_{i} &= \exp(\frac{1}{2}\ \gamma\ \mathbf{u}_{\omega}\ \Delta t_{i}^{2} + [{}^{c}\boldsymbol{\omega}]_{\times}\,\Delta t_{i}) && (2.27)
\end{aligned}
$$

Dans une configuration de caméra embarquée, la matrice de passage entre l'objet et la caméra est donnée par (2.5) :

$$
\begin{aligned}
{}^{c}\mathbf{T}_{o}(t_i) &= {}^{c}\delta\mathbf{T}_{i}\ {}^{c}\mathbf{T}_{o} && (2.28)\\
&= \begin{bmatrix} {}^{c}\delta\mathbf{R}_{i} & {}^{c}\delta\mathbf{t}_{i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{c}\mathbf{R}_{o} & {}^{c}\mathbf{t}_{o} \\ 0 & 1 \end{bmatrix} && (2.29)
\end{aligned}
$$

En représentant comme précédemment les déplacements dans le repère fixe (repère de l'objet cette fois) :

$$
\begin{aligned}
{}^{o}\delta\mathbf{t}_{i} &= {}^{c}\mathbf{R}_{o}^{T}\ {}^{c}\delta\mathbf{t}_{i} && (2.30)\\
{}^{o}\delta\mathbf{R}_{i} &= {}^{c}\mathbf{R}_{o}\ {}^{o}\delta\mathbf{R}_{i}\ {}^{c}\mathbf{R}_{o}^{T} && (2.31)
\end{aligned}
$$

En substituant ces deux équations dans (2.29) on obtient :

$$
{}^{c}\mathbf{T}_{o}(t_i) = \begin{bmatrix} {}^{c}\mathbf{R}\ {}^{o}\delta\mathbf{R} & {}^{c}\mathbf{R}_{o}({}^{o}\delta\mathbf{R}_{i}\,{}^{o}\mathbf{t} + {}^{o}\delta\mathbf{t}_{i}) \\ 0 & 1 \end{bmatrix} \tag{2.32}
$$

Noter que dans cette configuration les vitesses ne sont pas découplées, ce qui est normal car

contrairement à la configuration caméra déportée où c'est l'objet qui est en mouvement, les coordonnées en translation de l'objet dépendent de la rotation de la caméra.

Le point 3D est ainsi donné par :

$$ {}^{c}\mathbf{P}_i = {}^{c}\mathbf{R}\ {}^{o}\delta\mathbf{R}_i\ {}^{o}\mathbf{P} + {}^{c}\mathbf{R}_o({}^{o}\delta\mathbf{R}_i\,{}^{o}\mathbf{t} + {}^{o}\delta\mathbf{t}_i) \tag{2.33} $$

avec :

$$ {}^{o}\delta\mathbf{t}_i = \frac{1}{2}\ {}^{o}\mathbf{a}\ \Delta t_i^2 + {}^{o}\mathbf{v}\ \Delta t_i \tag{2.34} $$

$$ {}^{o}\delta\mathbf{R}_i = \exp(\frac{1}{2}\ \gamma\ \mathbf{u}_\omega\ \Delta t_i^2 + [{}^{o}\boldsymbol{\omega}]_\times\ \Delta t_i) \tag{2.35} $$

Enfin, quelle que soit la configuration de la caméra, le passage entre le point 3D et sa projection dans le plan normalisé de l'image est obtenu par l'équation (1.3).

2.3 Estimation de pose et de vitesse par optimisation non-linéaire

Le modèle de projection développé ci-dessus nous permet de calculer les positions des primitives dans l'image à partir de la pose à l'instant initial et des déplacements sur un intervalle de temps donné. Cette section présente la démarche inverse qui consiste à calculer la pose et la vitesse d'un objet en mouvement dont les primitives n'ont pas été acquises simultanément. Pour cela nous utiliserons un algorithme d'optimisation du type Newton-Raphson.

Soient $\mathbf{m}_i^*, \forall i = 1..n$ les positions des primitives dans l'image acquises aux instants $t_i = t_0 + \Delta t_i$ et soient $\mathbf{m}_i(\mathbf{x}), \forall i = 1..n$ les positions des primitives obtenues à partir du modèle de projection (Eq. 2.3). L'objectif de la méthode d'optimisation est de minimiser les erreurs entre les positions mesurées des primitives et les positions obtenues par le modèle de projection (erreurs de reprojection) données par :

$$ \epsilon_i(\mathbf{x}) = \mathbf{m}_i(\mathbf{x}) - \mathbf{m}_i^* \tag{2.36} $$

Le vecteur d'erreur incluant toutes les primitives est obtenu par la concaténation verticale des erreurs correspondant à chaque primitive :

$$ \epsilon(\mathbf{x}) = \begin{bmatrix} \mathbf{m}_1(\mathbf{x}) - \mathbf{m}_1^* \\ \vdots \\ \mathbf{m}_n(\mathbf{x}) - \mathbf{m}_n^* \end{bmatrix} \tag{2.37} $$

La fonction d'erreur peut ainsi être définie comme les moindres carrées des erreurs de

reprojection :

$$\mathbf{E} = \frac{1}{2}\epsilon(\mathbf{x})^T\epsilon(\mathbf{x}) \tag{2.38}$$

La minimisation de l'erreur de reprojection par la méthode de Newton-Raphson repose sur la récurrence suivante :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\ \mathbf{J}^+\ \epsilon \tag{2.39}$$

où $\mathbf{J}$ est la matrice Jacobienne de l'erreur $\mathbf{E}$.

L'implémentation de cet algorithme nécessite ainsi une initialisation de l'état (pose et vitesse) pour la première itération ainsi qu'une estimation de la Jacobienne. Il existe deux approches pour l'estimation de la Jacobienne : numérique ou formelle. Cependant, dans un contexte de vision rapide il est plus pertinent en termes de temps de calcul de privilégier l'approche formelle ce qui nécessite de déterminer l'expression de cette matrice en fonction de l'état.

2.3.1 Calcul de la matrice Jacobienne

Pour rester dans le cas le plus général nous adopterons le modèle de mouvement à accélération constante. Ainsi, dans le cas où l'accélération est connue, il sera possible de l'exploiter directement. A l'inverse, comme il a été signalé dans la section §2.2.3.4, dans le cas où l'accélération n'est pas connue, il suffit de négliger le terme d'accélération ce qui revient à considérer un mouvement à vitesse constante. Concernant le choix des paramètres de mouvement, nous opterons pour une représentation minimale où les paramètres de rotation qui semblent décrire le plus simplement le modèle de mouvement choisi (Éq. 2.16) est la représentation $\theta\mathbf{u}$ où l'on rappelle que $\mathbf{u}$ et θ désignent respectivement l'axe et l'angle de rotation. Le vecteur d'état 12×1 est ainsi défini par :

$$\mathbf{x} = [\mathbf{t}^T, \theta\mathbf{u}^T, \mathbf{v}^T, \omega^T]^T \tag{2.40}$$

À partir de la position du point 3D dans l'espace (2.25), la Jacobienne du point 3D $\mathbf{P}_i$ est définie par :

$$\mathbf{J}_{P_i} = \frac{\partial {}^c\mathbf{P}_i}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}}\left({}^c\delta\mathbf{R}_i\ {}^c\mathbf{R}_o{}^o\mathbf{P}_i + {}^c\delta\mathbf{t}_i + {}^c\mathbf{t}_o\right) \tag{2.41}$$

où $\delta\mathbf{t}_i$ et $\delta\mathbf{R}_i$ sont définis par (2.26) et (2.27).

Dans cette matrice de dimensions 3×12, chaque colonne j de $\mathbf{J}_{P_i}$ représente la dérivée du point 3D par rapport au paramètre $\mathbf{x}_j$. Afin de simplifier les calculs, cette matrice peut être décomposée en quatre blocs de matrices carrées où chaque bloc représente la dérivée

du point par rapport à un vecteur de paramètres 3×1 :

$$\mathbf{J}_{P_i} = \left[\frac{\partial^c \mathbf{P}_i(t)}{\partial^c \mathbf{t}_o}, \frac{\partial^c \mathbf{P}_i(t)}{\partial \theta \mathbf{u}}, \frac{\partial^c \mathbf{P}_i(t)}{\partial^c \mathbf{v}}, \frac{\partial^c \mathbf{P}_i(t)}{\partial^c \omega} \right] \tag{2.42}$$

Tout d'abord, la dérivée du point dans l'espace par rapport à la position du repère est donnée par :

$$\frac{\partial^c \mathbf{P}_i}{\partial^c \mathbf{t}_o} = \frac{\partial}{\partial^c \mathbf{t}_o} \left({}^c\delta\mathbf{R}_i \; {}^c\mathbf{R}_o {}^o\mathbf{P}_i + {}^c\delta\mathbf{t}_i + {}^c\mathbf{t}_o \right) = \frac{\partial^c \mathbf{t}_o}{\partial^c \mathbf{t}_o} = \mathbf{I}_3 \tag{2.43}$$

La Jacobienne relative à la vitesse de translation s'obtient de la même manière que pour le vecteur de position :

$$\frac{\partial^c \mathbf{P}_i}{\partial^c \mathbf{v}} = \frac{\partial^c \delta\mathbf{t}_i}{\partial^c \mathbf{v}} = \frac{\partial^c \mathbf{v}}{\partial^c \mathbf{v}} \Delta t_i = \Delta t_i \; \mathbf{I} \tag{2.44}$$

Afin d'obtenir la dérivée du point 3D par rapport à $\theta\mathbf{u}$, on commence par éliminer les composants de la fonction de projection qui ne dépendent pas du vecteur d'orientation :

$$\frac{\partial^c \mathbf{P}_i}{\partial \theta \mathbf{u}} = \frac{\partial}{\partial \theta \mathbf{u}} \left({}^c\delta\mathbf{R}_i \; {}^c\mathbf{R}_o {}^o\mathbf{P}_i + {}^c\delta\mathbf{t}_i + {}^c\mathbf{t}_o \right) = {}^c\delta\mathbf{R}_i \frac{\partial}{\partial \theta \mathbf{u}} \left({}^c\mathbf{R}_o {}^o\mathbf{P}_i \right) \tag{2.45}$$

Dans cette équation le seul élément qui dépend de $\theta\mathbf{u}$ est la matrice de rotation ${}^c\mathbf{R}_o$. La difficulté dans ce calcul est que la dérivée d'une matrice par rapport à un vecteur ne peut pas être estimée de façon directe. Pour contourner ce problème, on utilise la dérivée temporelle de la matrice de rotation définie par :

$$\frac{d^c \mathbf{R}_o}{dt} = {}^c\dot{\mathbf{R}}_o = \left[{}^c\boldsymbol{\omega} \right]_\times {}^c\mathbf{R}_o \tag{2.46}$$

Sachant que les coordonnées du point ${}^o\mathbf{P}_i$ sont constantes, l'équation (2.46) nous permet d'écrire :

$$\begin{aligned} {}^c\delta\mathbf{R}_i \frac{d^c \mathbf{R}_o}{dt} {}^o\mathbf{P}_i = {}^c\delta\mathbf{R}_i \frac{d}{dt} \left({}^c\mathbf{R}_o {}^o\mathbf{P}_i \right) &= {}^c\delta\mathbf{R}_i \left[{}^c\boldsymbol{\omega} \right]_\times {}^c\mathbf{R}_o {}^o\mathbf{P}_i & (2.47) \\ &= -{}^c\delta\mathbf{R}_i \left[{}^c\mathbf{R}_o {}^o\mathbf{P}_i \right]_\times {}^c\omega & (2.48) \end{aligned}$$

Par ailleurs, nous savons que la relation entre la dérivée du vecteur $\theta\mathbf{u}$ et la vitesse de rotation ω est donnée par :

$$\omega = \mathbf{L}_\omega^{-1}(\theta, \mathbf{u}) \frac{d(\theta\mathbf{u})}{dt} \tag{2.49}$$

où $\mathbf{L}_\omega^{-1}$ est donnée par [Mal98] :

$$\mathbf{L}_\omega^{-1}(\theta, \mathbf{u}) = \mathbf{L}_\omega^{-1} = \mathbf{I} + \frac{\theta}{2} \text{sinc}^2 \left(\frac{\theta}{2} \right) \left[\mathbf{u} \right]_\times + \left(1 - \text{sinc}(\theta) \right) \left[\mathbf{u} \right]_\times^2 \tag{2.50}$$

En substituant (2.49) dans (2.48), on obtient :

$$ {}^{c}\delta\mathbf{R}_i \frac{d}{dt}\left({}^{c}\mathbf{R}_o {}^{o}\mathbf{P}_i\right) = -{}^{c}\delta\mathbf{R}_i \left[{}^{c}\mathbf{R}_o {}^{o}\mathbf{P}_i\right]_\times \mathbf{L}_\omega^{-1} \frac{d(\theta\mathbf{u})}{dt} \tag{2.51} $$

La dérivée du point ${}^{c}\mathbf{P}_i$ par rapport au vecteur d'orientation $\theta\mathbf{u}$ est donné par :

$$ \frac{\partial {}^{c}\mathbf{P}_i}{\partial(\theta\mathbf{u})} = -{}^{c}\delta\mathbf{R}_i \left[{}^{c}\mathbf{R}_o {}^{o}\mathbf{P}_i\right]_\times \mathbf{L}_\omega^{-1} \tag{2.52} $$

Rappelons que dans le modèle de mouvement cosidéré, le vecteur vitesse de rotation à un axe fixe $\mathbf{u}_\omega = \frac{{}^{c}\omega}{\|{}^{c}\omega\|}$ avec une norme uniformément variée (2.14). Avant d'entamer le calcul de la dérivée de la position du point 3D par rapport à la vitesse de rotation, il est important de comprendre qu'un telle rotation pendant un intervalle de temps Δt se traduit par une rotation qui correspond au vecteur :

$$ \theta\mathbf{u}_\omega = \theta_\omega \cdot \mathbf{u}_\omega = \frac{1}{2}\gamma\Delta t^2 + {}^{c}\omega\Delta t \tag{2.53} $$

La dérivée de la matrice de rotation par rapport à la vitesse ω se calcule ainsi de la même manière que la dérivée de l'orientation ${}^{c}\mathbf{R}_o$ par rapport au vecteur $\theta\mathbf{u}$. Comme seul la matrice $\delta\mathbf{R}_i$ dépend de la vitesse de rotation, la dérivée de la position du point par rapport à la vitesse de rotation s'obtient aussi à partir de l'équation (2.45). En procédant comme pour la matrice d'orientation, nous calculons d'abord à partir de (2.45) la dérivée temporelle du déplacement en rotation :

$$ \frac{d^{c}\delta\mathbf{R}_i}{dt} {}^{c}\mathbf{R}_o {}^{o}\mathbf{P}_i = \left[\mathbf{L}_\omega^{-1}(\theta_\omega, \mathbf{u}_\omega)\frac{\theta\mathbf{u}_\omega}{dt}\right]_\times {}^{c}\delta\mathbf{R}_i {}^{c}\mathbf{R}_o {}^{o}\mathbf{P}_i \tag{2.54} $$

où, par analogie avec (2.50) :

$$ \begin{aligned} \mathbf{L}_\omega^{-1}(\theta_\omega, \mathbf{u}_\omega) = \mathbf{L}_{\dot\omega}^{-1} &= \mathbf{I} + \frac{\theta_\omega}{2}\mathrm{sinc}^2\left(\frac{\theta_\omega}{2}\right)[\mathbf{u}_\omega]_\times \\ &+ \left(1 - \mathrm{sinc}(\theta_\omega)\right)[\mathbf{u}_\omega]_\times^2 \end{aligned} \tag{2.55} $$

Par ailleurs :

$$ \frac{d\theta\mathbf{u}_\omega}{dt} = \frac{d}{dt}\left(\frac{1}{2}\gamma\Delta t_i^2 + {}^{c}\omega\Delta t_i\right) = \frac{d^{c}\omega}{dt}\,\Delta t_i = {}^{c}\dot\omega\,\Delta t \tag{2.56} $$

En substituant (2.56) dans (2.54) nous obtenons la dérivée du déplacement en orientation par rapport à la vitesse de rotation :

$$ \frac{{}^{c}\delta\mathbf{R}_i}{d^{c}\omega} {}^{c}\mathbf{R}_o {}^{o}\mathbf{P}_i = -\Delta t_i \left[{}^{c}\delta\mathbf{R}_i\ {}^{c}\mathbf{R}_o {}^{o}\mathbf{P}_i\right]_\times\ \mathbf{L}_{\dot\omega}^{-1} \tag{2.57} $$

Ainsi, la forme générale de la matrice Jacobienne du point 3D est obtenue en substi-

tuant (2.43), (2.52), (2.44) et (2.57) dans (2.42) :

$$\mathbf{J}_{P_i} = \begin{bmatrix} \mathbf{I} \ , & -{}^c\delta\mathbf{R}_i \left[{}^c\mathbf{R}_o {}^o\mathbf{P}_i\right]_\times \mathbf{L}_\omega^{-1} \ , & \Delta t_i \ \mathbf{I} \ , & -\Delta t_i \left[{}^c\delta\mathbf{R}_i {}^c\mathbf{R}_o {}^o\mathbf{P}_i\right]_\times \ \mathbf{L}_{\dot{\omega}}^{-1} \end{bmatrix} \tag{2.58}$$

À partir de la relation (1.26), la matrice Jacobienne du point image peut alors être obtenue à partir de la Jacobienne du point dans l'espace comme suit :

$$\mathbf{J}_{m_i} = \frac{\partial \mathbf{m}_i}{\partial \mathbf{x}} = \frac{\partial \mathbf{m}_i}{\partial \mathbf{P}_i} \frac{\partial \mathbf{P}_i}{\mathbf{x}} = \begin{bmatrix} \frac{1}{z_i} & 0 & -\frac{x_i}{z_i^2} \\ 0 & \frac{1}{z_i} & -\frac{y_i}{z_i^2} \end{bmatrix} \mathbf{J}_{P_i} \tag{2.59}$$

Enfin, la Jacobienne correspondant à l'ensemble des points $\mathbf{m}_i \quad \forall i = 1..n$ est une matrice de dimensions $2n \times 12$ construite à partir de la concaténation verticale des matrices relatives aux points 2D :

$$\mathbf{J}_m = \begin{bmatrix} \mathbf{J}_{m_1}^T, \ \mathbf{J}_{m_2}^T, \cdots, \ \mathbf{J}_{m_n}^T \end{bmatrix}^T \tag{2.60}$$

2.3.2 Algorithme d'estimation

2.3.2.1 Choix du temps de référence

Le principe de causalité suggère que le temps de référence doit correspondre à l'instant d'acquisition de la première FI. Cependant, le but étant l'estimation temps-réel de la pose et de la vitesse ce choix n'est pas judicieux. En effet, un tel choix [AAALM06] signifierait que la pose estimée correspond à l'instant d'acquisition de la première FI mais estimée après l'acquisition de la dernière FI. Ceci introduit un retard systématique de $(n-1)\ T_a$ (T_a étant la période d'acquisition) dans l'estimation de la pose et la vitesse.

Afin de supprimer ce retard, une approche intuitive consiste à intégrer la vitesse estimée sur l'intervalle de temps en partant de la pose à l'instant t_0. Cependant, cette approche introduit des calculs superflus et amplifie le bruit dans la pose estimée puisque la vitesse est intégrée entre le temps de référence et l'instant courant. La solution qui nous parait la plus judicieuse pour estimer la pose et la vitesse à l'instant courant est de choisir t_0 comme l'instant d'acquisition de la dernière image. La conséquence de ce choix est que tous les autres points de l'image sont capturés à des instants négatifs ce qui n'influe en rien l'algorithme d'estimation.

2.3.2.2 Suivi de points dans l'image

Cette méthode d'estimation requiert la mise en correspondance de n points 2D et 3D. Cette tâche peut être assurée par l'implémentation d'un algorithme de suivi des points dans l'image permettant de positionner les FI dans la zone de l'image contenant les primitives désirées. Dans ce cas, la tâche de positionnement des FI dans l'image devient

critique. En effet, si une fenêtre d'acquisition est mal positionnée dans l'image, la FI acquise ne contiendra pas tout ou partie de l'information désirée. Cependant, si l'on dispose d'une estimation valide de la pose et de la vitesse, il est relativement simple, à partir du modèle de mouvement et de la fonction de projection, de prédire la position d'un point dans l'image aux prochaines m acquisitions ($m \leq n$). Par conséquent, la méthode d'estimation proposée intègre dans le même algorithme le calcul de pose et de vitesse et le suivi des points dans l'image.

2.3.2.3 Ordonnancement parallèle

Dans un mode d'acquisition classique, chaque image acquise doit contenir un nombre de primitives suffisant pour l'estimation ou la commande. Ainsi, les calculs peuvent être lancés à chaque acquisition, ce qui permet l'implémentation sérielle de l'algorithme. Dans un mode d'acquisition séquentielle une seule primitive visuelle est capturée à chaque acquisition. La quantité d'information ainsi acquise ne permet pas d'estimer le vecteur d'état. Une solution possible à ce problème consiste à conserver les n derniers points acquis. Ainsi, le tableau contenant ces points est mis à jour à chaque nouvelle acquisition. Cependant, cette solution n'est pas optimale sur deux points : le premier est que la fréquence d'acquisition se trouve bridée par les calculs (Fig. 2.6). Le second est que l'acquisition d'un seul point représente une information insuffisante pour permettre une actualisation efficace des données et de l'état estimé. L'algorithme d'estimation doit donc mieux équilibrer la quantité d'informations d'acquise et la fréquence d'estimation, sans que la fréquence d'acquisition (plus rapide) se trouve limitée par l'estimation (plus lente). L'implémentation d'un tel algorithme est possible en utilisant la programmation parallèle comme représenté sur la figure 2.7.

Après avoir initialisé le vecteur d'état, l'acquisition et l'estimation se déroulent en parallèle. L'estimation basée sur la méthode de Newton-Raphson débute après l'acquisition d'un certain nombre de points. Si le nombre de points acquis à chaque estimation peut être modulé ($1 \leq m \leq n$), l'estimation est toujours basée sur les n derniers points acquis. Pendant le calcul, l'acquisition continue de tourner. La prédiction des positions des m prochains points dans l'image (pour le positionnement des FI) est ensuite calculée à partir de la pose et de la vitesse estimée.

2.4 Estimation d'état par filtrage de Kalman étendu

Dans la section précédente, une méthode d'estimation de pose et de vitesse par une approche d'optimisation non linéaire à partir de l'acquisition d'un nombre suffisant de correspondances a été présentée. Cependant, bien que cette méthode présente certains avantages, elle présente en contrepartie certains inconvénients notamment en vision rapide.

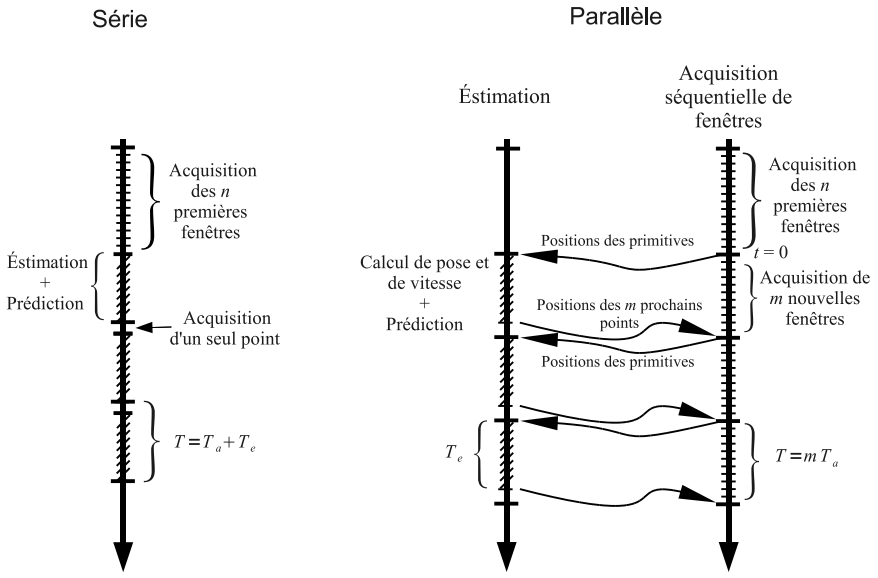


FIGURE 2.6 – Ordonnancement série et parallèle pour un système d'acquisition séquentielle de fenêtres d'intérêt.

En effet, hormis le fait qu'elle soit relativement simple à implémenter, l'un des avantages de cette méthode est qu'elle permet de calculer l'état qui minimise directement l'erreur dans l'image. De plus, l'algorithme d'estimation ne requiert pas de connaissance préalable sur les statistiques du bruit de modèle et de mesure, ce qui lui permet de converger même en présence d'erreurs de modèles et de bruits de mesure. Cependant, la mesure par vision peut être perturbée par plusieurs sources d'erreurs telles que la variation d'éclairage, les erreurs d'étalonnage de la caméra, etc. Dans ces circonstances, la minimisation de l'erreur de mesure ne représente pas la solution optimale dans la mesure où ce bruit est propagé dans l'estimation. De plus, l'algorithme de Newton-Raphson, comme les autres algorithmes d'optimisation non linéaire, ne tiennent pas explicitement compte de la dynamique du système (continuité de l'état), ce qui peut être gênant dans une application de commande. Par ailleurs, un point qui nous intéresse particulièrement en vision rapide est le coût calculatoire. De ce point de vue, la minimisation des moindres carrés, comme nous l'avons vu précédemment, nécessite un historique des mesures afin de contraindre suffisamment le système d'équations. Ceci se traduit en pratique par la nécessité de calculer le modèle de projection et la Jacobienne relative à tous les points considérés à chaque estimation même s'il n'y a qu'un seul nouveau point acquis.

Une autre approche pour l'estimation rapide de la pose et de la vitesse consiste à exploiter "à la volée" les données acquises. En d'autres termes, au lieu d'accumuler les informations et de les traiter dès que la quantité de données est suffisante, on traite directement chaque donnée acquise. Ainsi, l'incrément apporté par chaque observation peut permettre de converger vers la solution désirée. Ceci est possible en utilisant une méthode

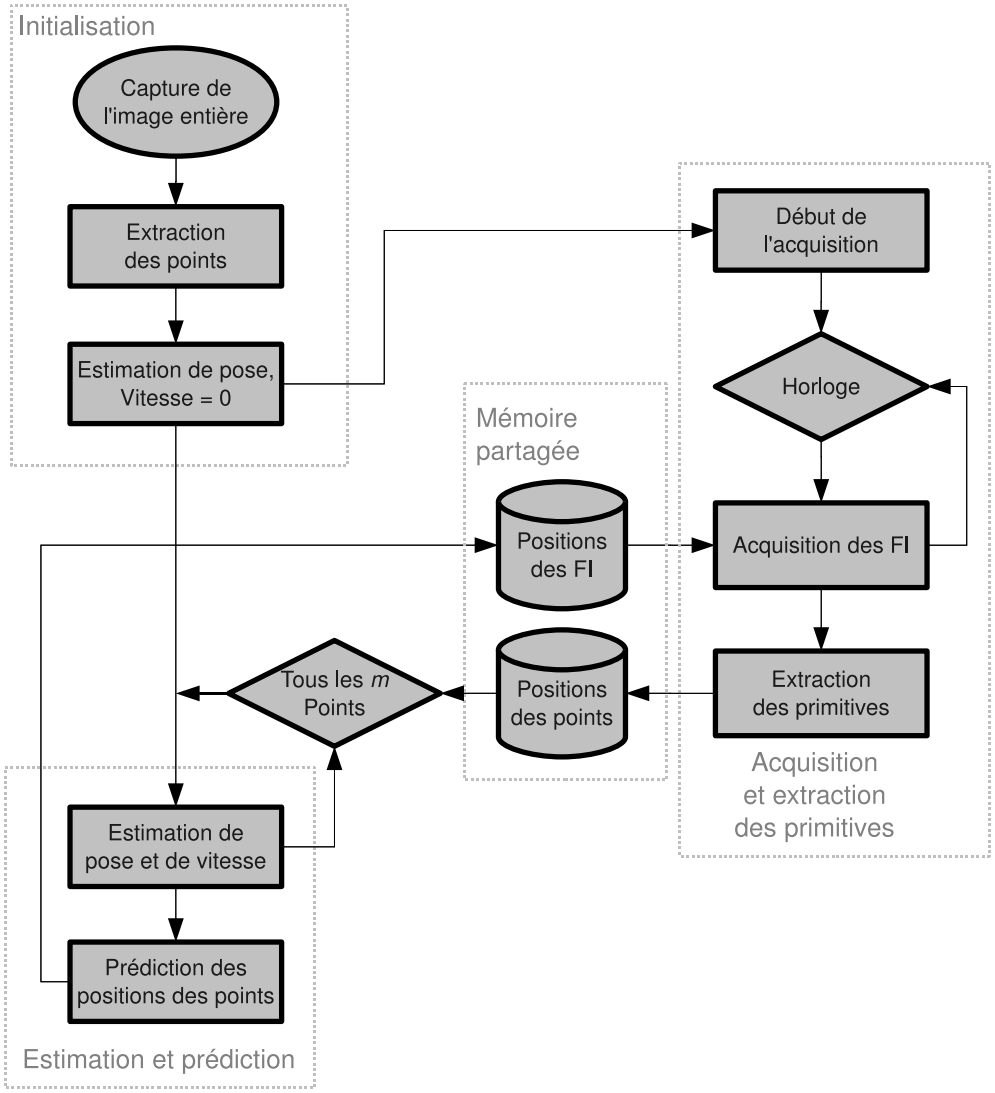


FIGURE 2.7 – Structure de l'algorithme d'estimation de pose et de vitesse par minimisation des moindres carrés à partir d'une acquisition séquentielle de fenêtres d'intéret.

telle que le filtrage de Kalman qui permet d'exploiter aussi bien des mesures redondantes que des mesures incomplètes[3] [WB95] sans nécessiter l'historique des mesures. En effet, dans un filtre de Kalman, la mise à jour de l'état est basée sur une estimation de l'état précédent et une observation courante. Par ailleurs, en tenant compte explicitement de la dynamique du système, le filtre de Kalman permet d'estimer l'état optimal (dans le sens des moindres carrés) d'un système à condition que les bruits sur le modèle d'évolution et la mesure soient de type Gaussiens à moyenne nulle. Par ailleurs, bien que l'optimalité de la solution ne soit démontrée que pour les systèmes linéaires, l'utilisation de cette méthode a déja fait ses preuves dans des systèmes non linéaires grâce au Filtre de Kalman Étendu (EKF), notamment en robotique [LK90, WW92, JU97].

Tout d'abord, nous allons présenter, dans ce qui suit, les équations du filtre de Kalman et du filtre de Kalman étendu avant de les utiliser dans la mesure de pose et de vitesse rapide.

2.4.1 Estimation de pose et de vitesse

La contrainte majeure dans l'utilisation du filtrage de Kalman est la nécessité d'avoir une estimation à priori des paramètres statistiques des bruits, ce qui n'est pas toujours trivial. De plus, ces paramètres peuvent varier au cours du temps ce qui peut influer négativement sur la solution jusqu'à compromettre la stabilité du filtre. Afin de pallier à ces problèmes, les paramètres du filtre peuvent être adaptés en ligne grâce au filtrage de Kalman adaptatif (AEKF) [MT76]. Ainsi, plusieurs méthodes d'adaptation peuvent être envisagées selon les propriétés du système considéré. Dans le cas où les paramètres du bruit varient quantitativement il suffit de mettre à jour les matrices de covariances relatives au bruit sur l'état et sur la mesure à partir des résidus calculés après la mise à jour [JLV99]. Par contre, dans des applications où le modèle d'évolution varie dans le temps d'autres approches consistent à commuter entre plusieurs modèles [BRM97] ou à estimer en ligne la matrice de transition [WU00].

L'estimation de pose, et éventuellement de vitesse, s'inscrit en robotique dans deux contextes relativement différents : les tâches d'estimation pure (pour le suivi par exemple) et les applications de commandes basées vision. La différence majeure entre ces deux applications c'est l'accessibilité du vecteur de commande $\boldsymbol{u}$. En effet, dans les applications d'estimation, l'entrée du système n'est généralement pas accessible. Dans ce cas, à moins d'introduire des capteurs supplémentaires pour l'estimation de l'état tel qu'un capteur gyroscopique [ACVT04] ou inertiel [YN01, GV05], l'entrée du système est considérée comme une perturbation. De plus, l'entrée $\boldsymbol{u}$ ne présente souvent pas les caractéristiques d'un bruit Gaussien à moyenne nulle. Une solution possible pour compenser cette pertur-

3. Une information est incomplète quand elle ne permet pas de décrire complètement l'état du système (plusieurs solutions possibles)

bation consiste à considérer l'entrée $\boldsymbol{u}$ comme une variable supplémentaire qui doit être estimée [LK90]. Une approche plus classique, lorsque le signal d'erreur de mesure calculée à postériori présente une certaine stationnarité pendant un certain intervalle de temps, consiste à calculer les paramètres statistiques de la perturbation ce qui permet d'adapter les matrices de covariances et de compenser la composante continue de la perturbation [FJS01]. Cependant, un compromis doit être fait. En effet, le calcul stochastique est basé sur les n dernières mesures. Dans le cas où n est choisi faible, les valeurs statistiques sur le bruit de mesure sont moins significatives. Dans le cas contraire où n est important, la condition de stationnarité pendant l'intervalle de temps plus important devient moins valide. Par ailleurs, du fait du nombre de données à traiter l'estimation de ces paramètres induisent un coût de calcul relativement lourd pour un système temps-réel. C'est pourquoi une implémentation récursive de ce filtre à été proposée dans [LSV06, FJS01] où il à été démontré que l'estimation résistait mieux aux changements abruptes de trajectoires que dans un filtre de Kalman étendu non adaptatif.

Cependant, dans une application de commande dynamique, la question de l'estimation de l'accélération ne se pose plus puisque il s'agit tout simplement de l'entrée du système calculée par la loi de commande qui est donc parfaitement connue. Dans ce cas un filtrage de Kalman étendu est suffisant. Comme ce qui nous intéresse dans ce chapitre est l'estimation pour la commande, nous considérons le modèle d'évolution du système $f(\mathbf{x}, \boldsymbol{u})$ défini par l'équation (2.24) où les déplacements correspondent au modèle de mouvement à accélération constante pour une configuration de caméra déportée (Eq. (2.15) et (2.16)). L'état du système à l'instant k est défini par $\mathbf{x}_k = (\mathbf{t}_k^T, \theta\mathbf{u}_k^T, \mathbf{v}_k^T, \omega_k^T)^T$ et son entrée, qui représente l'accélération du repère mobile, par $\boldsymbol{u} = ({}^c\mathbf{a}^T, {}^c\dot{\omega}^T)^T$. L'accélération angulaire de l'objet γ est obtenue en projetant l'accélération d'entrée ${}^c\dot{\omega}$ sur l'axe de la vitesse de rotation :

$$\gamma = {}^c\dot{\omega}^T \ \mathbf{u}_\omega \tag{2.61}$$

Si l'entrée du système $\boldsymbol{u}$ n'est pas connue (hors application de commande dynamique), alors l'accélération est considérée comme étant nulle ce qui revient à adopter le modèle de mouvement à vitesse constante.

La fonction d'évolution du système étant définie par les équations, (2.13), (2.14), (2.15) et (2.16), où l'intervalle de temps Δt_i représente la période d'acquisition du système de vision T_a qui est la même que la période d'estimation du filtre T_e. Ainsi, la prédiction de l'état du système à l'instant k à partir de l'état précédent $\hat{\mathbf{x}}_{k-1|k-1}$ et de l'entrée $\boldsymbol{u}_{k-1}$ est donnée par :

$${}^c\hat{\mathbf{t}}_{k|k-1} = {}^c\hat{\mathbf{t}}_{k-1|k-1} + {}^c\hat{\mathbf{v}}_{k-1|k-1}T_e + \frac{1}{2}{}^c\mathbf{a}T_e^2 \tag{2.62}$$

$$\left[\theta\mathbf{u}_{k|k-1}\right]_\times = \operatorname{logm}\left(e^{\left[{}^c\hat{\omega}_{k-1|k-1}T_e+\frac{1}{2}\gamma_{\mathbf{u}_\omega}T_e^2\right]_\times} e^{\left[\theta\mathbf{u}_{k-1|k-1}\right]_\times} \right) \tag{2.63}$$

$${}^c\hat{\mathbf{v}}_{k|k-1} = {}^c\hat{\mathbf{v}}_{k-1|k-1} + {}^c\mathbf{a}\ T_e \tag{2.64}$$

$${}^c\hat{\omega}_{k|k-1} = {}^c\hat{\omega}_{k-1|k-1} + \gamma\ T_e \tag{2.65}$$

où logm représente le logarithme matriciel.

On peut noter que la plupart des équations décrivant l'évolution du système sont linéaires ce qui permet de simplifier la forme de la matrice de transition et d'avoir des propriétés proches d'un système linéaire. Cependant, l'évolution de la rotation n'est pas linéaire. En réalité, un autre choix des paramètres qui rendrait le modèle d'évolution complètement linéaire consisterait à adopter la dérivée du vecteur de rotation $\dot{\theta\mathbf{u}}$ comme paramètre de vitesse angulaire dans le vecteur d'état à la place de ω. Dans ce cas l'équation (2.63) deviendrait :

$$\widehat{\theta\mathbf{u}}_{k|k-1} = \widehat{\theta\mathbf{u}}_{k-1|k-1} + \dot{\theta\mathbf{u}}_{k-1|k-1}\ T_e \tag{2.66}$$

Cependant, un tel choix ne se révèle pas très judicieux car il compromettrait la cohérence du modèle de mouvement. En effet, intégrer la dérivée du vecteur rotation introduit une accélération systématique en plus de l'accélération d'entrée ce qui revient à ajouter des perturbations au modèle.

Afin de compléter la forme de la Jacobienne du modèle d'évolution $\boldsymbol{A}_k$, il suffit d'effectuer un développement limité de l'évolution du vecteur de rotation :

$$\widehat{\theta\mathbf{u}}_{k|k-1} \approx \widehat{\theta\mathbf{u}}_{k-1|k-1} + T_e\ \frac{d\theta\mathbf{u}_{k-1|k-1}}{dt} = \widehat{\theta\mathbf{u}}_{k-1|k-1} + T_e\ \mathbf{L}_\omega\ \omega_{k-1|k-1} \tag{2.67}$$

où $\mathbf{L}_\omega$ exprime la variation du vecteur d'orientation par rapport à la vitesse de rotation ω (1.19).

Ainsi, à partir des équations (2.62), (2.67), (2.64) et (2.65) la matrice Jacobienne du modèle de projection se présente comme suit :

$$\boldsymbol{A}_k = \begin{bmatrix} \mathbf{I} & 0 & T_e\mathbf{I} & 0 \\ 0 & \delta\mathbf{R} & 0 & T_e\mathbf{L}_\omega \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \tag{2.68}$$

En ce qui concerne l'étape de mise à jour, la mesure représente la position des primitives dans l'image. Le fait que les données soient traitées après chaque acquisition et que l'estimation ne requière pas l'historique des données implique qu'il n'est pas nécessaire d'appliquer le modèle de projection à temps différé. Ainsi, l'équation d'observation représente tous simplement la fonction de projection classique. La matrice Jacobienne

de celle-ci peut alors être déduite de la matrice d'interaction classique ou de la matrice Jacobienne calculée précédemment en posant $\Delta t_i = 0$.

Ainsi, la matrice Jacobienne du modèle de projection est donnée par :

$$\boldsymbol{C}_k = \begin{bmatrix} \frac{1}{z_i} & 0 & -\frac{x_i}{z_i^2} \\ 0 & \frac{1}{z_i} & -\frac{y_i}{z_i^2} \end{bmatrix} \left[\mathbf{I},\ -\left[{}^c\mathbf{P}\right]_\times \mathbf{L}_\omega^{-1},\ 0,\ 0\right] \tag{2.69}$$

Finalement, l'algorithme d'estimation par filtrage de Kalman étendu se présente comme suit :

Prédiction

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}) \tag{2.70}$$

$$\boldsymbol{F}_{k|k-1} = \boldsymbol{A}_k\ \boldsymbol{F}_{k-1|k-1}\boldsymbol{A}_k^T + \boldsymbol{Q} \tag{2.71}$$

Mise à jour

$$\tilde{\mathbf{y}}_k = \mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1}) \tag{2.72}$$

$$\boldsymbol{S}_k = \boldsymbol{C}_k\ \boldsymbol{F}_{k|k-1}\ \boldsymbol{C}_k^T + \boldsymbol{R} \tag{2.73}$$

$$\boldsymbol{K}_k = \boldsymbol{F}_{k|k-1}\boldsymbol{C}_k^T\boldsymbol{S}_k^{-1} \tag{2.74}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \boldsymbol{K}_k\ \tilde{\mathbf{y}}_k \tag{2.75}$$

$$\boldsymbol{F}_{k|k} = (\mathbf{I} - \boldsymbol{K}_k\ \boldsymbol{C}_k)\ \boldsymbol{F}_{k|k-1} \tag{2.76}$$

2.4.2 Coût calculatoire

Un point crucial dans l'implémentation d'algorithmes d'estimation par vision rapide est le coût calculatoire. De ce point de vue, il est intéressant de constater que le filtre de Kalman étendu se divise en deux parties au sens de l'évolution du nombre d'opérations par rapport aux paramètres d'acquisition. En effet, le nombre total d'opérations (multiplications + additions) qu'implique la partie prédiction est fixe quel que soit le nombre de points acquis. Cependant, l'exécutions de cette partie est nécessaire à chaque acquisition. Par contre, dans la partie mise à jour, le nombre d'opérations est linéairement proportionnel au nombre de points acquis m sauf dans l'inversion de la matrice $\boldsymbol{S}_k$ qui est de dimention $2m \times 2m$ et qui a une complexité de calcul de $O(m^3)$. Au vu de ces propriétés, et indépendamment des capacités du système de vision utilisé, la question que l'on peut se poser est la suivante : afin de réduire le temps de traitement des données, est-il plus intéressant d'acquérir tous les points au même moment où de les acquérir de manière séquentielle comme proposé ci-dessus ?

Pour répondre à cette question, il suffit de calculer le nombre d'opérations nécessaires dans les deux cas de figure.

Soit n le nombre total des points et m le nombre de points acquis dans chaque fenêtre d'intérêt. Le nombre total d'opérations N que requiert cet algorithme pour l'estimation est donné par [4] :

$$N = \frac{n}{m} M_p + M_e \; n + 2m \; (2m - 1) \; (4m - 2) \tag{2.77}$$

où M_p est le nombre d'opérations de la partie prédiction et M_e le nombre d'opérations de la partie mise à jour pour chaque point acquis.

Ainsi, dans une acquisition globale et séquentielle nous avons respectivement $m = n$ et $m = 1$. Afin de déterminer quelle approche est la plus coûteuse en termes de calculs, il suffit pour cela de calculer le nombre d'opérations induites par chaque méthode :

$$N_c = M_p + M_e \; 2n + 2n \; (2n - 1) \; (4n - 2) \tag{2.78}$$

$$N_s = (M_p + M_e + 8) \; n \tag{2.79}$$

où N_c et N_s représentent respectivement le nombre d'opérations dans une acquisition globale et séquentielle. Nous pouvons constater que N_s est de complexité $O(n)$ alors que N_c est de complexité $O(n^3)$. [5]

Dans notre application, le nombre d'opérations élémentaires dans la prédiction et l'estimation sont : $M_p = 192$ et $M_e = 348$. La figure 2.8 donne le nombre d'opérations nécessaires dans chaque méthode en fonction du nombre total de points observés sur l'objet n. On peut ainsi voir que le nombre d'opérations nécessaire dans une acquisition globale croit considérablement plus vite que dans une acquisition séquentielle. Par ailleurs, comme le nombre de points nécessaires pour l'estimation de l'état est de $n \geq 6$ alors l'acquisition séquentielle est plus intéressante du point de vue du coût de calcul par rapport à une acquisition globale quel que soit le nombre de points.

De plus, comme le montre la figure 2.6, pour la même fréquence d'acquisition des primitives, la fréquence d'acquisition séquentielle est plus importante. Ainsi, l'horizon de prédiction est beaucoup plus court que dans une acquisition globale. De plus, comme l'estimation est mise à jour à chaque acquisition, alors le mode d'acquisition séquentiel devrait assurer un meilleur suivi des primitives dans l'image que le mode d'acquisition global.

4. Le nombre d'opérations concernant l'inversion correspond à la méthode d'élimination de Gauss-Jordan.

5. Dans les algorithmes les plus optimisés, la complexité peut être baissée jusqu'à $O(n^{2,376})$[CW87].

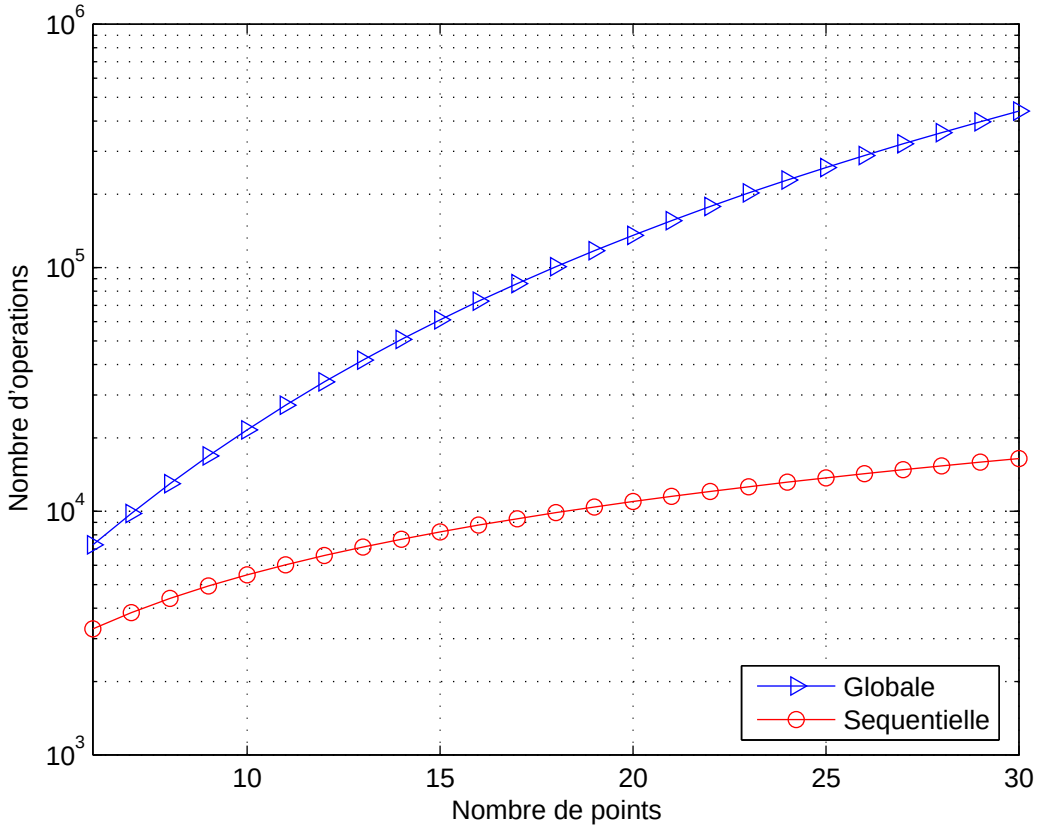


FIGURE 2.8 – Nombre d'opérations nécessaires pour l'estimation de pose et de vitesse dans une acquisition globale et dans une acquisition séquentielle à partir de la même quantité de données.

2.5 Conclusion

Nous avons abordé dans ce chapitre les aspects théoriques de la vision rapide. En particulier, deux méthodes d'estimation de pose et de vitesse basées sur l'acquisition séquentielle de fenêtres d'intérêt ont été présentées. La première, basée sur la minimisation des moindres carrés dans l'espace de mesure, consiste à accumuler suffisamment de données puis d'effectuer une estimation à travers un algorithme d'optimisation non linéaire. Dans ce contexte, l'expression de la matrice jacobienne qui exprime la variation des positions des primitives dans l'image par rapport à la variation des paramètres de la pose et de la vitesse a été développée. Cette méthode présente l'avantage de ne pas nécessiter de connaissance à priori sur le système. Cependant, l'estimation requiert le traitement de toutes les données à chaque nouvelle acquisition. La seconde méthode basée sur un filtrage de Kalman étendu permet le calcul de la mise à jour de l'état à partir d'une information incomplète, ce qui est particulièrement intéressant avec un système d'acquisition séquentiel. De plus, la prise en compte du modèle d'évolution permet de mieux approcher la solution optimale, et donc, de filtrer les données. De plus, l'acquisition séquentielle permet de réduire le nombre d'opérations nécessaires pour l'estimation de l'état par rapport à une acquisition classique dans laquelle les points sont acquis simultanément. L'inconvénient de l'estimation par filtrage de Kalman est qu'elle nécessite la connaissance apriori des paramètres statistiques des bruits de modèles et de mesures sous peine d'obtenir une solution erronée pouvant conduire à la divergence de l'algorithme. Le tableau 2.5 présente un comparatif entre les deux méthodes proposées.

Newton-Raphson	Filtre de Kalman étendu
- Fonction de projection faisant intervenir la pose et la vitesse.	+ Fonction de projection classique.
+ Minimisation dans l'espace des mesures.	+ Minimisation dans l'espace d'état.
- Mise à jour à partir des n dernières acquisitions.	+ Mise à jour basée sur une estimation de l'état et une nouvelle mesure.
+ Faible risque de minimum local (car associé à une tâche de suivi).	- Risque de divergence.
+ Minimisation de l'erreur non compromise par le bruit.	- Solution non optimale si les paramètres statistiques ne sont pas connus ou s'ils ne sont pas de type Gaussien à moyenne nulle.
- Prédiction pour le suivi sur les m prochaines acquisitions.	+ Prédiction sur l'acquisition suivante seulement.
- Traitement de toutes les données (fonction de projection + Jacobienne) quel que soit le nombre de points acquis.	+ Traitement limité aux nouvelles données acquises.

TABLE 2.1 – Comparaison entre l'estimation de pose et de vitesse par le méthode de Newton-Raphson et par filtrage de Kalman étendu.

Chapitre 3

Asservissement visuel cinématique par vision rapide

Nous avons abordé dans le chapitre précédent les aspects modélisation et estimation liés à l'acquisition non simultanée de primitives visuelles dans l'image. Ce chapitre propose une démarche de synthèse de lois de commande à partir de ce type de primitives (spatio-temporelles).

3.1 Introduction

Dans le cadre de l'asservissement visuel, le choix des primitives [TFC09, AEH02] et des paramètres les décrivant [TC05, IO05, KI07b] est primordial afin de réaliser la synthèse de lois de commande permettant d'obtenir les meilleures propriétés de linéarisation et de découplage. Dans un contexte de vision rapide, il s'agit d'élaborer la loi de commande avec les meilleures performances pour des vitesses de déplacement importantes [CRE91, BC95, MBG96, CC01b]. En effet, les déplacements à vitesses élevées nécessitent généralement des trajectoires de références ce qui implique que les poses désirées sont proches des poses courantes. Le problème de découplage est alors moins critique dans la mesure où la commande par vision permet d'assurer une linéarisation autour du point de fonctionnement.

Dans le contexte de l'asservissement visuel haute performance, les primitives acquises de façon séquentielle sont particulièrement intéressantes du point de vue commande car leurs positions dans l'image dépendent aussi bien de la pose que de la vitesse. On peut donc imaginer qu'il est possible de développer à partir de ces primitives des lois de commande permettant de réguler ces deux grandeurs (c.-à-d. la pose du robot, sa vitesse ou les deux à la fois). L'estimation de l'erreur entre la vitesse actuelle et désirée doit, en effet, permettre l'amélioration du suivi de trajectoire. Cependant, la particularité que présente la fonction de projection de dépendre de plus de paramètres extrinsèques que de degrés de liberté

soulève une problématique particulière en synthèse de commande par vision aussi bien du point de vue dimensionnel que fonctionnel.

Le problème de l'utilisation de primitives spatio-temporelles en asservissement visuel a fait l'objet d'une étude approfondie dans [C98, CC01a, CC01b]. Les les primitives visuelles étudiées sont les déplacements dans l'image représentés par un modèle de mouvement quadratique [Ban85, Sub89]. Deux approches ont alors été proposées pour l'utilisation de ce genre de primitives en asservissement visuel. La première consiste à reconstruire des primitives purement géométriques en intégrant le mouvement des points dans l'image. La seconde consiste à utiliser directement le mouvement des primitives des points dans l'image. Dans la première méthode, le vecteur de commande obtenu est un torseur cinématique. Celui-ci permet de construire un schéma de commande cinématique. Dans la seconde, approche la dérivée temporelle de la fonction de tâche s'exprime en fonction de l'accélération et d'une fonction non linéaire (quadratique) qui dépend de la vitesse. La synthèse de la loi de commande à partir de cette primitives revient alors à implémenter une commande dynamique.

Dans la suite de ce chapitre, nous allons présenter une fonction de tâche qui dépend de la pose et de la vitesse. À partir de celle-ci, différentes lois de commande présentant des particularités distinctes seront proposées. Nous verrons ensuite qu'il est possible d'élaborer une méthode d'estimation rapide de pose et de vitesse en utilisant les mêmes formalismes. Étant donné que du point de vue pratique la primitive de base qui convient le mieux à l'acquisition par régions d'intérêt est le point, nous présenterons la matrice d'interaction relative à cette primitive dans le contexte de l'acquisition non simultanée des primitives. Enfin, pour satisfaire ou améliorer les performances du système de vision en termes de coût calculatoire, une méthode rapide d'estimation de la matrice d'interaction sera proposée.

3.2 Synthèse de lois de commande cinématiques

3.2.1 La fonction de tâche

Dans les applications de commande de robot, la fonction de tâche définit un critère à optimiser qui s'apparente dans le contexte de l'optimisation non linéaire à la fonction de coût. La différence entre ces deux approches est la prise en compte explicite de la structure particulière de l'espace euclidien dans le calcul de la correction (commande ou estimation). Dans l'asservissement visuel 2D classique, cette fonction représente l'erreur entre la position des primitives visuelles observées et désirées dans l'image, transformées par la matrice de combinaison en un vecteur à l éléments correspondant aux l degrés de liberté à contrôler. Ainsi, dans les applications où le mouvement est défini dans l'espace, la dimension du vecteur calculé par la fonction de tâche est de 6×1.

En réalité, tous les modèles de projection utilisés dans l'élaboration de schémas de

commande par vision utilisent des hypothèses qui permettent de simplifier les lois de commande obtenues (homogénéité des millieux traversés par la lumière, taux de remplissage de capteur = 100%, etc.). Cependant, l'effet de certains phénomènes indésirables augmente avec la vitesse de déplacement du robot. En effet, comme nous l'avons déjà vu en §1.4.1, les images ne sont pas acquises instantanément par les capteurs de vision mais intégrées sur le temps d'exposition du capteur ce qui se traduit par des images plus ou moins brouillées lorsque le mouvement observé est significativment rapide par rapport au temps d'exposition.

En réalité, l'effet de ce phénomène n'est pas très significatif dans notre cas puisque la méthode d'acquisition par fenêtres d'intérêt présentée dans le chapitre précédent favorise un temps d'exposition très faible. En effet, l'acquisition non simultanée implique que les primitives ne sont pas exposées au même instant. Cela signifie que l'énergie accumulée dans la rétine est réinitialisée après l'acquisition de chaque fenêtre d'intérêt. Afin d'assurer des fréquences d'acquisition importantes, il est nécessaire de réduire le temps d'exposition de chaque fenêtre d'intérêt qui pourrait contenir une ou plusieurs primitives. Par conséquent, le phénomène de trainée se trouve fortement réduit. Un objet se déplaçant par exemple à une vitesse de 1m/s à 1m de profondeur d'une caméra dotée d'un capteur 1024x1024 pixels où le temps d'exposition est réglé à 0.1ms [1] ne se déplace pas plus de 0.1 pixel dans l'image durant la phase d'exposition. Ce déplacement dans l'image n'est en réalité pas plus important que la précision de détection des primitives visuelles en statique. On peut ainsi admettre que l'effet de trainée d'une cible observée dans l'image reste négligeable dans cet ordre de rapport entre le temps d'exposition et les vitesses observées.

Si une vitesse de cible relativement importante n'a pas beaucoup d'influence sur la précision de détection, elle est plus significative quand il s'agit de l'acquisition d'un ensemble de points. En effet, le temps entre l'instant d'acquisition de la première et de la $n^{\text{ième}}$ primitive $\Delta t_n = (n-1)\ T_a$ est du même ordre que la période de la commande $T_c = n\ T_a$ ce qui représente le temps entre deux poses successives dans la trajectoire de référence. Les déformations dans l'image sont ainsi du même ordre.

Considérons l'acquisition en mode global de deux images successives. Si la première image est acquise à $t = t_0$ alors l'image suivante est acquise à $t = t_0 + T_c$ où T_c est la période d'acquisition globale des images. Les instants d'acquisition de la $\text{i}^{\text{ième}}$ primitive en mode séquentiel est donné par : $t = t_0 + (i-1)T_a$ [2]. L'instant d'acquisition de la première primitive en acquisition séquentielle correspond à l'instant d'acquisition de la première image globale. La position de la primitive est ainsi le même dans les deux modes d'acquisition. Par contre, l'instant d'acquisition de la dernière primitive en mode séquentiel

1. L'acquisition avec un tel temps d'exposition est tout à fait réalisable avec un éclairage artificiel.
2. Pour la même fréquence d'acquisition des primitives que dans le premier cas c.-à-d. la fréquence d'acquisition des fenêtres d'intérêt est n fois supérieur à la fréquence d'acquisition des images globales, où n est le nombre de primitives.

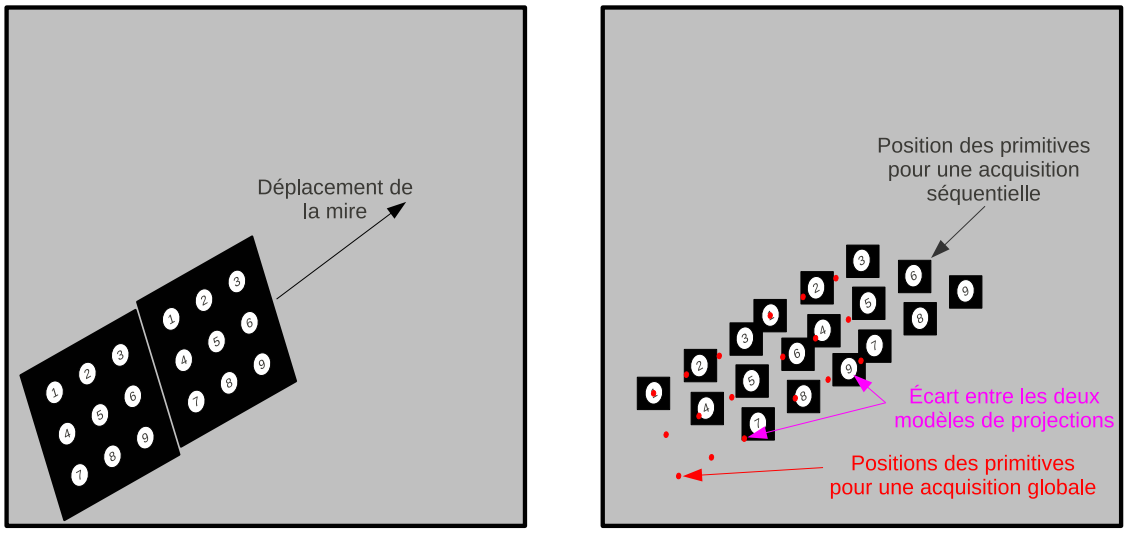


FIGURE 3.1 – Écart entre les positions des points dans l'image pour une acquisition classique et pour une acquistion séquentielle.(a) Deux positions successives de la cible en acquisition classique. (b) Positions des primitives dans une acquisition globale et dans une acquisition séquentielle de fenêtres d'intérêts.

$(t = t_0 + (n-1)T_a)$ est beaucoup plus proche de la seconde image acquise en mode global $(t = t_0 + nT_a)$ que de la première $(t = t_0)$. La position de cette primitive en mode séquentiel est ainsi bien plus proche de la pose suivante correspondant à une acquisition globale que de la pose courante. On peut ainsi en déduire que les déformations dans l'image sont du même ordre de grandeur que les distances parcourues par les primitives entre deux instants successifs (Fig. 3.1).

La question que l'on pourrait se poser est : si la prise en compte des distorsions dues à l'acquisition séquentielle est pertinente du point de vue métrologique, en est-il de même pour l'asservissement visuel ?

En effet, contrairement à l'estimation des paramètres extrinsèques de la caméra qui restent sensibles aux erreurs sur le modèle de la cible et sur les paramètres intrinsèques de la caméra, l'asservissement visuel 2D est, lui, robuste face à ce type d'erreurs. En réalité, cette hypothèse n'est que partiellement vraie. Dans une commande point-à-point où la trajectoire n'est pas contrainte, la fonction de tâche est définie par rapport à la position actuelle et la position finale. Si l'on considère que le processus de commande requiert un certain nombre d'itérations pour converger vers la position désirée, les déformations dues au mode d'acquisition deviennent "négligeables" devant les erreurs entre les primitives actuelles et désirées. Par ailleurs, étant donné que pour ce type de commande les positions initiales et finales sont caractérisées par une vitesse nulle, ces déformations ne devraient pas introduire des erreurs statiques. Dans ces conditions, négliger les déformations dans

l'élaboration de la commande n'est pas critique.

Dans les applications de suivi de trajectoire, la distance entre les positions successives des primitives désirées dans l'image sont relativement faibles (pour peu que la trajectoire soit lisse). Les gains de la commande sont par conséquent plus élevés que dans le cas du point-à-point. Ainsi, pour des vitesses de déplacement relativement élevées, les déformations dans l'image sont nécessairement du même ordre que les distances entre les positions successives des primitives dans l'image (Fig. 3.1). Dans ce cas, la prise en compte des distorsions induites par le mode d'acquisition et le mouvement devient cruciale.

Par ailleurs, nous avons vu que le modèle de projection lié à l'acquisition séquentielle des primitives faisait intervenir aussi bien la pose que la vitesse. Cette propriété ouvre des perspectives nouvelles en terme de commande car elle permet de mesurer et donc de réguler la vitesse en plus de la pose du robot comme le ferait par exemple une commande proportionnelle dérivée.

A partir de la fonction de projection (2.5) la fonction de tâche correspondant au modèle de mouvement à vitesse constante peut être écrite en fonction des paramètres de pose et de vitesse de l'objet[3] :

$$\mathbf{e} = \mathbf{C}\left(\mathbf{m}(\mathbf{r}, \boldsymbol{\tau}) - \mathbf{m}^*(t)\right) \tag{3.1}$$

où $\mathbf{m}(\mathbf{r}, \boldsymbol{\tau})$ représente la projection des primitives visuelles dans le plan image. Celle-ci dépend de la pose de l'objet par rapport à la caméra $\mathbf{r}$ et de son torseur cinématique $\boldsymbol{\tau} = [\mathbf{v}^T, \omega^T]^T$. $\mathbf{m}^*(t)$ est la position désirée des primitives et $\mathbf{C}$ est la matrice de combinaison. Pour ne pas surcharger ces notations, on notera : $\mathbf{m} = \mathbf{m}(\mathbf{r}, \boldsymbol{\tau})$ et $\mathbf{m}^* = \mathbf{m}^*(t)$.

La fonction de tâche est typiquement définie comme un vecteur de dimension 6 afin de contrôler les 6 ddl de l'objet mobile.

À partir de l'équation précédente (Eq. 3.1), la dérivée de la fonction de tâche est calculée comme suit :

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} + \frac{\partial \mathbf{e}}{\partial \boldsymbol{\tau}} \frac{d\boldsymbol{\tau}}{dt} - \mathbf{C}\, \frac{d\mathbf{m}^*}{dt} + \dot{\mathbf{C}}\, (\mathbf{m} - \mathbf{m}^*) \tag{3.2}$$

De la même manière qu'en §1.2.1, il est généralement admis en asservissement visuel que la variation de la matrice de combinaison est généralement négligeable devant la variation du signal $\mathbf{m}$. La dérivée de la fonction de tâche peut ainsi être approximée par :

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} + \frac{\partial \mathbf{e}}{\partial \boldsymbol{\tau}} \frac{d\boldsymbol{\tau}}{dt} - \mathbf{C}\, \frac{d\mathbf{m}^*}{dt} \tag{3.3}$$

On peut noter que le calcul de la dérivée temporelle de la vitesse n'est pas contradictoire avec l'hypothèse de vitesse constante. En effet, la vitesse n'est supposée constante que durant un intervalle de temps limité correspondant au temps d'acquisition de l'ensemble

3. Dans les applications de robotique rapide il n'est souvent pas possible d'embarquer la caméra du fait des accélérations importantes de l'organe terminal, c'est pourquoi les formulations suivantes seront développées dans un contexte de configuration déportée de la caméra.

des points dans l'image. Par ailleurs, même si tel n'était pas le cas, le calcul de la différentielle de la fonction de tâche en fonction de la vitesse permettrait de caractériser l'influence de la variation de la vitesse sur les positions des points dans l'image, et ce, que la vitesse de la cible soit constante ou pas. Un exemple analogue à celui-ci est l'estimation de pose d'un objet statique où l'on calcule un incrément sur les paramètres de pose à partir de la dérivée de la pose même si elle est en réalité constante. Ainsi, les deux premiers termes du membre de droite de l'équation précédente peuvent être calculés comme suit :

$$\frac{\partial \mathbf{e}}{\partial \mathbf{r}}\frac{d\mathbf{r}}{dt} = \mathbf{C}\frac{\partial \mathbf{m}}{\partial \mathbf{r}}\mathbf{A}\boldsymbol{\tau} = \mathbf{C}\ \mathbf{L}_{\tau}\boldsymbol{\tau} \tag{3.4}$$

$$\frac{\partial \mathbf{e}}{\partial \boldsymbol{\tau}}\frac{d\boldsymbol{\tau}}{dt} = \mathbf{C}\frac{\partial \mathbf{m}}{\partial \boldsymbol{\tau}}\dot{\boldsymbol{\tau}} = \mathbf{C}\ \mathbf{L}_{\dot{\tau}}\dot{\boldsymbol{\tau}} \tag{3.5}$$

où $\mathbf{L}_{\boldsymbol{\tau}}$ et $\mathbf{L}_{\dot{\tau}}$ représentent respectivement la relation entre la variation des positions des points 2D dans l'image et la variation de la pose et du torseur cinématique de l'objet dont l'expression sera donnée en §3.3. La matrice $\mathbf{A}$ permet d'exprimer la dérivée des paramètres de pose en fonction du torseur cinématique. En pratique nous n'avons pas besoin de calculer formellement cette matrice ni la dérivée partielle $\frac{\partial \mathbf{m}}{\partial \mathbf{r}}$ du fait que l'expression de la matrice d'interaction $\mathbf{L}_{\boldsymbol{\tau}}$ peut être retrouvée directement à partir de la dérivée temporelle de la fonction de tâche. Rappelons, par ailleurs, que la matrice de combinaison est considérée comme étant constante, ce qui n'empêche pas qu'elle puisse être modifiée durant la commande comme pour une commande adaptative.

Ainsi, en substituant (3.4) et (3.5) dans (3.3) la dérivée de la fonction de tâche par rapport au temps peut être écrite en fonction du torseur cinématique $\boldsymbol{\tau}$ et de l'accélération $\dot{\boldsymbol{\tau}}$ comme suit :

$$\dot{\mathbf{e}} = \mathbf{C}\mathbf{L}_{\boldsymbol{\tau}}\boldsymbol{\tau} + \mathbf{C}\mathbf{L}_{\dot{\tau}}\dot{\boldsymbol{\tau}} - \mathbf{C}\frac{d\mathbf{m}^*}{dt} \tag{3.6}$$

Cette équation détermine ainsi la variation de la fonction de tâche en fonction du torseur cinématique et de l'accélération. Elle peut ainsi être utilisée pour calculer la vitesse et l'accélération qui permettent de converger vers la solution désirée. Cependant, dans un asservissement visuel cinématique le vecteur de commande représente le vecteur des vitesses articulaires du robot dont la dimension ne peut dépasser 6 pour un robot non redondant. Étant donné que la dérivée de la fonction de tâche est donnée en fonction de 12 paramètres, se pose alors la question de la façon d'utiliser tout ou partie de ces paramètres dans le calcul du vecteur de commande. En d'autres termes, comment doit-on effectuer la synthèse de lois de commande à partir de ces primitives visuelles et du modèle de projection qui leur est associé ?

3.2.2 Asservissement visuel à partir de primitives spatio-temporelles

La démarche classique dans la synthèse de commande cinématique par vision est d'imposer une décroissance exponentielle de la fonction de tâche [ECR92, HHC96, CH06]. L'équation différentielle permettant d'obtenir ce comportement peut être écrite sous forme :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \tag{3.7}$$

où $\lambda > 0$ est un gain proportionnel qui permet de régler la vitesse de convergence.

En substituant (3.7) dans (3.6) on obtient :

$$-\lambda \mathbf{e} = \mathbf{C}\mathbf{L}_{\boldsymbol{\tau}} \boldsymbol{\tau} + \mathbf{C}\mathbf{L}_{\dot{\boldsymbol{\tau}}} \dot{\boldsymbol{\tau}} - \mathbf{C}\frac{d\mathbf{m}^*}{dt} \tag{3.8}$$

Cette équation définit un cadre général pour l'asservissement visuel dans lequel différentes lois de commande peuvent être envisagées selon que le vecteur de commande soit calculé à partir du torseur cinématique, de l'accélération ou de la combinaison des deux.

3.2.2.1 Commande à partir du torseur cinématique

Étant donné que la commande cinématique consiste à déterminer une vitesse de commande, la solution la plus immédiate consiste à n'exploiter que le torseur cinématique et d'ignorer la contribution de l'accélération. En effet, dans une commande cinématique, on considère que la vitesse du robot est constante entre deux échantillons successifs. Ainsi, l'accélération peut être considérée comme étant nulle durant cet intervalle de temps. Le vecteur de commande peut donc être calculé à partir de (3.8) en supposant une accélération nulle. On obtient ainsi :

$$-\lambda \mathbf{e} = \mathbf{C}\mathbf{L}_{\boldsymbol{\tau}} \boldsymbol{\tau} + \frac{\partial \mathbf{e}}{\partial t} \tag{3.9}$$

d'où

$$\boldsymbol{\tau} = (\mathbf{C}\mathbf{L}_{\boldsymbol{\tau}})^{-1} \left[-\lambda \mathbf{C}(\mathbf{m} - \mathbf{m}^*) - \frac{\partial \mathbf{e}}{\partial t} \right] \tag{3.10}$$

Ainsi pour $\mathbf{C} = \widehat{\mathbf{L}_{\boldsymbol{\tau}}}^{+}$, et sachant qu'on ne dispose que d'une estimation de la matrice d'interaction, l'expression du vecteur de commande est donnée par :

$$\boldsymbol{\tau} = -\lambda \widehat{\mathbf{L}_{\boldsymbol{\tau}}}^{+} (\mathbf{m} - \mathbf{m}^*) - \widehat{\frac{\partial \mathbf{e}}{\partial t}} \tag{3.11}$$

Le schéma de commande correspondant à cette régulation est représenté sur la figure 3.2.

L'équation de fermeture de boucle est obtenue à partir de (3.8), (3.11) et (3.9) :

$$\dot{\mathbf{e}} = -\lambda \widehat{\mathbf{L}_{\boldsymbol{\tau}}}^{+} \mathbf{L}_{\boldsymbol{\tau}} \mathbf{e} - \widehat{\mathbf{L}_{\boldsymbol{\tau}}}^{+} \mathbf{L}_{\boldsymbol{\tau}} \widehat{\frac{\partial \mathbf{e}}{\partial t}} + \frac{\partial \mathbf{e}}{\partial t} \tag{3.12}$$

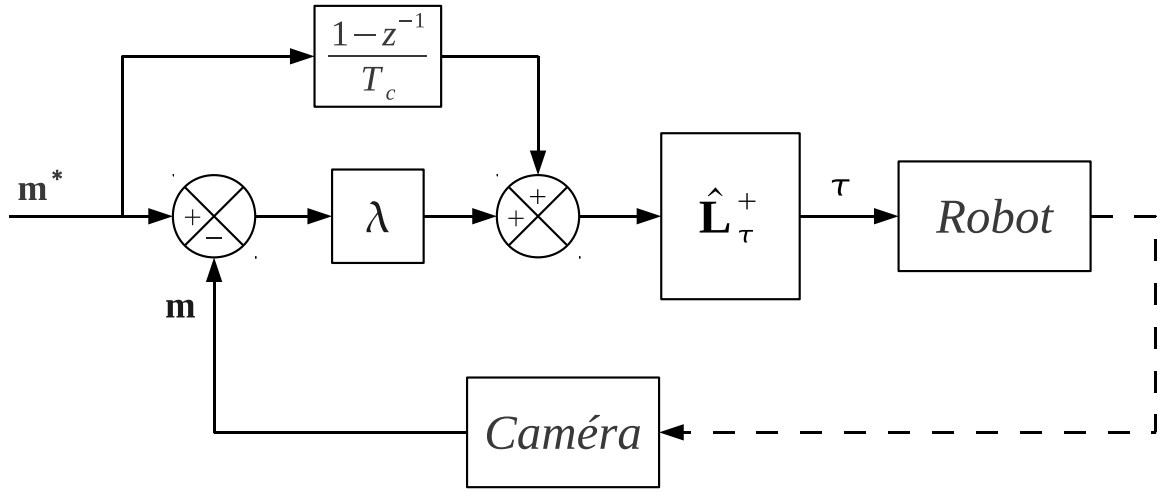


FIGURE 3.2 – Asservissement visuel à partir du torseur cinématique

Nous pouvons constater de cette équation que la condition de stabilité asymptotique locale de la loi de commande est donnée par :

$$\widehat{\mathbf{L}_{\boldsymbol{\tau}}}^{+} \mathbf{L}_{\boldsymbol{\tau}} > 0 \tag{3.13}$$

Par ailleurs, l'erreur de suivi n'est nulle que si :

$$\widehat{\mathbf{L}_{\boldsymbol{\tau}}}^{+} \mathbf{L}_{\boldsymbol{\tau}} \widehat{\frac{\partial \mathbf{e}}{\partial t}} = \frac{\partial \mathbf{e}}{\partial t} \tag{3.14}$$

Cette commande a l'avantage d'être simple. Cependant, négliger la contribution de l'accélération dans cette loi de commande ne permet pas de tirer profit des capacités de suivi de trajectoire. En effet, même si l'on admet que l'équation (3.14) est bien vérifiée, le suivi de trajectoire en pratique peut comporter des erreurs car la vitesse réelle du robot ne correspond pas forcément à la vitesse de commande et cela pour deux raisons principales. La première est que la vitesse cartésienne calculée par la commande est transformée en vitesse articulaire. Le fait que cette transformation entre les vitesses cartésiennes et articulaires ne soit pas linéaire implique qu'une intégration d'une vitesse articulaire constante ne correspond pas à une vitesse cartésienne constante. Ceci représente des erreurs de modèle pouvant perturber la commande. La seconde raison est que les vitesses articulaires envoyées par la commande haut niveau ne sont pas celles exécutées par le robot. En effet, pour éviter des discontinuités entre les vitesses de commande successives, la boucle de régulation bas niveau du robot intègre, en général, un interpolateur destiné à lisser le profil de vitesse. Ceci signifie que la vitesse envoyée au robot n'est atteinte qu'après un certain intervalle de temps.

Ainsi, le fait de négliger l'accélération dans la commande (Fig. 3.2) revient à éliminer

le retour capteur qui permet de corriger l'erreur entre la vitesse désirée et la vitesse réelle du robot. Noter que le terme $\widehat{\mathbf{L}_{\boldsymbol{\tau}}}^{+}\frac{d\mathbf{m}^*}{dt}$, représente une anticipation sur la vitesse, et non pas une régulation, et ne permet en aucun cas de compenser cette erreur. C'est pourquoi si l'on veut obtenir un suivi précis de la vitesse, il est nécessaire de ne pas négliger la contribution de l'accélération. Néanmoins, cette commande peut être destinée aux applications de robotique rapide lorsque les contraintes sur la précision de suivi ne sont pas critiques.

3.2.2.2 Commande à partir de l'accélération

Le signal sur lequel le système est asservi ayant un caractère spatio-temporel, les primitives visuelles peuvent aussi bien être perçues comme des positions de consigne que comme des vitesses désirées dans l'image. Cette caractéristique nous offre la possibilité de choisir l'une et/ou l'autre dimension du signal. Par ailleurs, dans certaines applications [SBC96, CC00, CC01b, SA05, Mar07], il est plus intéressant de définir la tâche de suivi de vitesse à partir des déplacements qu'à partir des positions. Dans ce type d'applications l'accent est mis sur la vitesse désirée définie par la trajectoire des primitives dans l'image. Ainsi, contrairement à la commande précédente exploitant le torseur cinématique (ce qui donne lieu à des erreurs de suivi de vitesse) c'est l'accélération calculée à partir des déformations dans l'image qui est utilisée. Ainsi, la vitesse de commande envoyé au régulateur bas niveau, obtenue par intégration de l'accélération, est utilisée pour la régulation de la vitesse.

À partir de l'équation (3.8), si l'on ne considère que la contribution de l'accélération dans la régulation du système on obtient :

$$-\lambda\mathbf{e} = \mathbf{C}\mathbf{L}_{\dot{\boldsymbol{\tau}}}\dot{\boldsymbol{\tau}} + \frac{\partial\mathbf{e}}{\partial t} \tag{3.15}$$

L'expression de l'accélération est ainsi donnée par :

$$\dot{\boldsymbol{\tau}} = (\mathbf{C}\mathbf{L}_{\dot{\boldsymbol{\tau}}})^{-1}\left(-\lambda(\mathbf{m}-\mathbf{m}^*) - \frac{\partial\mathbf{e}}{\partial t}\right) \tag{3.16}$$

En posant $\mathbf{C} = \widehat{\mathbf{L}_{\dot{\boldsymbol{\tau}}}}^{+}$, l'accélération est calculée par :

$$\dot{\boldsymbol{\tau}} = -\lambda\widehat{\mathbf{L}_{\dot{\boldsymbol{\tau}}}}^{+}(\mathbf{m}-\mathbf{m}^*) - \widehat{\frac{\partial\mathbf{e}}{\partial t}} \tag{3.17}$$

La loi de commande cinématique s'obtient par intégration de l'accélération. Cependant, comme en pratique les commandes par vision sont numériques, l'équation (3.17) est discrétisée et exprimée sous forme récurrente :

$$\boldsymbol{\tau}_i = T_c\ \dot{\boldsymbol{\tau}}_i + \boldsymbol{\tau}_{i-1} \tag{3.18}$$

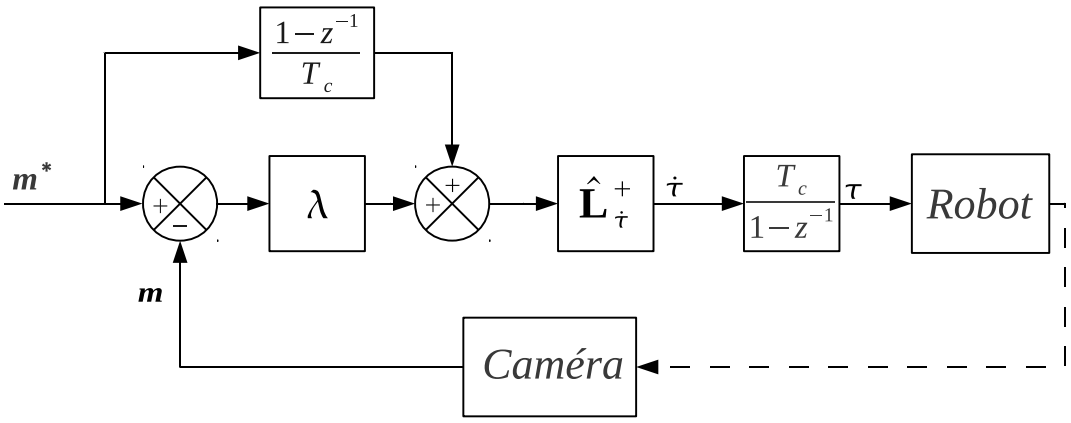


FIGURE 3.3 – Asservissement visuel cinématique à partir de l'accélération

où $\boldsymbol{\tau}_{i-1}$ représente le torseur cinématique à l'instant précédent et T_c la période d'échantillonnage de la commande.

Finalement, la loi de commande pour le suivi de vitesse (Fig. 3.3) s'écrit comme suit :

$$\boldsymbol{\tau}_i = T_c \left(-\lambda \widehat{\mathbf{L}_{\dot{\tau}}}^{+} (\mathbf{m} - \mathbf{m}^*) + \widehat{\mathbf{L}_{\dot{\tau}}} \frac{d\mathbf{m}^*}{dt} \right) + \boldsymbol{\tau}_{i-1} \tag{3.19}$$

Une particularité intéressante de cette commande est que l'intégrateur joue le rôle de filtre passe bas. De ce fait, cette commande est moins sensible vis-à-vis du bruit que la commande précédente. Par ailleurs, étant donné que la vitesse est régulée, il n'est plus nécessaire d'effectuer une anticipation sur la vitesse à partir de la consigne. Cela nous permet d'obtenir le schéma de commande de la figure 3.3. Cependant, cette commande présente certains inconvénients. Le premier est qu'elle ne peut être implémentée que pour du suivi de trajectoire. En effet, l'obtention de l'accélération est basée sur les déformations dans l'image. Or, les états initial et final du robot sont, à priori, statiques (pas de déformation dans l'image). Par ailleurs, étant donné que cette commande est basée sur le mouvement et non pas sur la position, une dérive peut être constatée lors du suivi.

3.2.2.3 Commande à partir du torseur cinématique et de l'accélération

L'utilisation ne permet qu'une extraction partielle des informations exploitables dans la commande. En revanche l'utilisation de la vitesse et de l'accélération de façon complémentaire et cohérente devrait permettre de tirer profit des avantages des deux commandes précédentes.

L'accélération peut s'écrire comme étant la dérivée numérique du torseur cinématique :

$$\dot{\boldsymbol{\tau}}_i = \frac{\boldsymbol{\tau}_i - \boldsymbol{\tau}_{i-1}}{T_c} \tag{3.20}$$

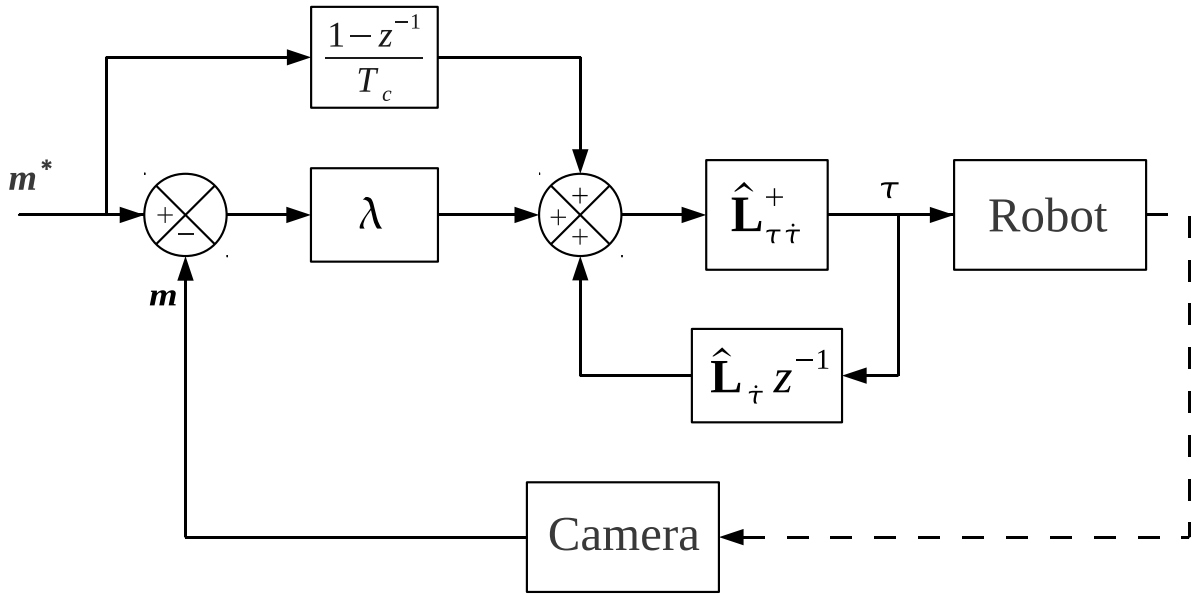


FIGURE 3.4 – Asservissement visuel à partir du torseur cinématique et de l'accélération

En substituant dans l'équation (3.8), on obtient :

$$\begin{aligned} -\lambda \mathbf{e} &= \mathbf{C}\mathbf{L}_{\boldsymbol{\tau}} \boldsymbol{\tau}_i + \mathbf{C}\mathbf{L}_{\dot{\boldsymbol{\tau}}} \left(\frac{\boldsymbol{\tau}_i - \boldsymbol{\tau}_{i-1}}{T_c} \right) + \frac{\partial \mathbf{e}}{\partial t} \\ &= \mathbf{C} \left(\mathbf{L}_{\boldsymbol{\tau}} + \frac{1}{T_c} \mathbf{L}_{\dot{\boldsymbol{\tau}}} \right) \boldsymbol{\tau}_i - \mathbf{C}\mathbf{L}_{\dot{\boldsymbol{\tau}}} \boldsymbol{\tau}_{i-1} + \frac{\partial \mathbf{e}}{\partial t} \end{aligned} \tag{3.21}$$

En notant :

$$\mathbf{L}_{\tau\dot{\tau}} = \mathbf{L}_{\boldsymbol{\tau}} + \frac{1}{T_c} \mathbf{L}_{\dot{\boldsymbol{\tau}}} \tag{3.22}$$

et en choisissant :

$$\mathbf{C} = \widehat{\mathbf{L}_{\tau\dot{\tau}}}^{+} \tag{3.23}$$

on obtient l'équation du torseur cinématique :

$$\boldsymbol{\tau}_i = -\lambda \widehat{\mathbf{L}_{\tau\dot{\tau}}}^{+} (\mathbf{m} - \mathbf{m}^*) + \widehat{\mathbf{L}_{\tau\dot{\tau}}}^{+} \widehat{\mathbf{L}_{\dot{\boldsymbol{\tau}}}} \boldsymbol{\tau}_{i-1} - \frac{\partial \mathbf{e}}{\partial t} \tag{3.24}$$

Notons que cette loi de commande se présente sous forme récursive puisque l'expression du vecteur de commande $\boldsymbol{\tau}_i$ est donnée en fonction de sa valeur à l'instant précédent $\boldsymbol{\tau}_i = f(\mathbf{r}, \boldsymbol{\tau}_{i-1}, \mathbf{m}, \mathbf{m}^*)$. Cette particularité par rapport aux lois de commande classiques fait que l'équation de fermeture de boucle à aussi une forme récursive. Par conséquent, l'établissement de la condition de convergence de la commande n'est pas trivial.

Aussi, la forme récurente de la loi de commande traduit le fait que le vecteur de commande intègre, en plus de la régulation de pose, une régulation de vitesse calculée

sur la base de l'intégrale de l'accélération correspondant à l'erreur de vitesse. Ce terme permet ainsi de réguler la vitesse sur la base de la vitesse désirée et de la vitesse du robot à l'instant précédent. Le schéma de commande correspondant est représenté sur la figure 3.4.

3.2.3 Asservissement visuel virtuel

Nous avons vu dans la section §1.3.4 que les schémas d'asservissement visuel peuvent bien être utilisés pour la commande que pour l'estimation de pose. Cette dualité est aussi vérifiée pour les lois de commandes présentées ci-dessus qui peuvent être utilisées pour l'estimation de pose et de vitesse. Cependant nous nous contenterons dans ce qui suit de développer les schémas de commandes qui nous sembles les plus pertinents pour l'estimation de pose et de vitesse. Dans cette perspective, deux versions d'asservissements visuels virtuels sont présentées : une version complète et une version simplifiée.

3.2.3.1 Version explicite

L'expression du torseur cinématique qui permet de converger vers la pose de l'objet par rapport à la caméra peut être obtenu de la même manière que dans le cadre de l'asservissement visuel réel (3.24), avec $\mathbf{m}^*$ qui représente la position actuelle des primitives dans l'image et $\widehat{\mathbf{m}}$ la position des points calculée à partir du modèle de projection (2.5).

L'expression de l'accélération peut être obtenue à partir de l'équation (3.8) de la même manière que pour le torseur cinématique. En effet, en substituant (3.18) dans (3.8) on obtient :

$$\begin{aligned} -\lambda \mathbf{e} &= \mathbf{C}\mathbf{L}_{\boldsymbol{\tau}}(T_c\dot{\boldsymbol{\tau}}_i + \boldsymbol{\tau}_{i-1}) + \mathbf{C}\mathbf{L}_{\dot{\boldsymbol{\tau}}}\dot{\boldsymbol{\tau}}_i - \mathbf{C}\,\frac{d\mathbf{m}^*}{dt} \\ &= \mathbf{C}(T_c\mathbf{L}_{\boldsymbol{\tau}} + \mathbf{L}_{\dot{\boldsymbol{\tau}}})\dot{\boldsymbol{\tau}}_i + \mathbf{C}\mathbf{L}_{\boldsymbol{\tau}}\boldsymbol{\tau}_{i-1} - \mathbf{C}\frac{d\mathbf{m}^*}{dt} \end{aligned} \tag{3.25}$$

où $\boldsymbol{\tau}_{i-1}$ représente le torseur cinématique à l'instant précédent et $\dot{\boldsymbol{\tau}}_i$ l'accélération moyenne de la cible entre les instants successifs t_{i-1} et t_i.

Ainsi, l'accélération peut s'écrire comme suit :

$$\dot{\boldsymbol{\tau}}_i = \left(T_c\ \mathbf{C}(\mathbf{L}_{\boldsymbol{\tau}} + \frac{1}{T_c}\mathbf{L}_{\dot{\boldsymbol{\tau}}})\right)^{-1}\left(-\lambda\mathbf{e} - \mathbf{C}\mathbf{L}_{\boldsymbol{\tau}}\boldsymbol{\tau}_{i-1} + \mathbf{C}\frac{d\mathbf{m}^*}{dt}\right) \tag{3.26}$$

en posant $\mathbf{C} = \widehat{\mathbf{L}_{\tau\dot{\tau}}}^{+}$ on obtient à partir de (3.26) l'expression de l'accélération permettant

d'estimer la vitesse :

$$\dot{\boldsymbol{\tau}}_i = \frac{1}{T_c}\left(-\lambda\widehat{\mathbf{L}_{\tau\dot{\tau}}}^{+}(\widehat{\mathbf{m}}-\mathbf{m}^*) - \widehat{\mathbf{L}_{\tau\dot{\tau}}}^{+}\left(\widehat{\mathbf{L}_{\boldsymbol{\tau}}}\boldsymbol{\tau}_{i-1} - \frac{d\mathbf{m}^*}{dt}\right)\right) \tag{3.27}$$

Notons que l'accélération s'exprime également en fonction du torseur cinématique à l'instant précédent. Cette propriété ne témoigne pas d'une intégration mais d'une dérivation numérique. En effet, le terme $\widehat{\mathbf{L}_{\boldsymbol{\tau}}}\boldsymbol{\tau}_{i-1} - \dot{\mathbf{m}}$ représente la différence entre la vitesse calculée et la vitesse désirée dans l'image. La matrice d'interaction $\widehat{\mathbf{L}_{\dot{\tau}\tau}}^{+}$ permet ensuite de compenser cette erreur. Le terme $\widehat{\mathbf{L}_{\tau\dot{\tau}}}^{+}\frac{d\mathbf{m}^*}{dt}$ dans (3.27) peut ainsi être considérée comme une anticipation sur l'accélération.

Dans l'asservissement visuel classique, le torseur cinématique calculé représente une correction de la pose courante du robot. Si le processus est stable, l'intégration de la vitesse calculée permet de converger vers la pose désirée. Dans la méthode proposée, la commande virtuelle fournit deux vecteurs correspondant à la correction de la pose et de la vitesse relative entre le robot et la cible. De la même manière, si le processus converge, l'intégration du torseur cinématique permet de converger vers la pose courante. De plus, l'intégrale de l'accélération permet de converger vers la vitesse courante.

L'estimation de la pose est obtenue par intégration du torseur cinématique calculé dans l'équation (3.4) où la position et l'orientation sont actualisées respectivement à partir du vecteur vitesse de translation et du vecteur vitesse de rotation ([PC05]) :

$$^c\mathbf{t}_o(i) = {}^c\mathbf{t}_o(i-1) + \mathbf{v}_{\boldsymbol{\tau}}\, T_c \tag{3.28}$$

$$^c\mathbf{R}_o(i) = e^{[\omega_{\boldsymbol{\tau}}\ T_c]_\times}\, {}^c\mathbf{R}_o(i-1) \tag{3.29}$$

où $\mathbf{v}_{\boldsymbol{\tau}}$ et ω_τ sont les composantes en translation et en rotation du torseur ($\tau = [\mathbf{v}_{\boldsymbol{\tau}}^T, \omega_\tau^T]^T$).

De la même manière, la mise à jour de la vitesse est obtenue par intégration de l'accélération :

$$\mathbf{v}_\tau(i) = T_c\ \dot{\mathbf{v}}_{\dot{\boldsymbol{\tau}}} + \mathbf{v}_\tau(i-1) \tag{3.30}$$

$$\omega_\tau(i) = e^{[\dot{\omega}_{\dot{\boldsymbol{\tau}}}\ T_c]_\times}\, \omega_\tau(i-1) \tag{3.31}$$

où $\dot{\mathbf{v}}_{\dot{\boldsymbol{\tau}}}$ et $\dot{\omega}_{\dot{\boldsymbol{\tau}}}$ désignent respectivement les accélérations en translation et en rotation obtenus à partir de l'accélération $\dot{\boldsymbol{\tau}}$.

Le schéma d'asservissement visuel virtuel est représenté sur la figure 3.5.

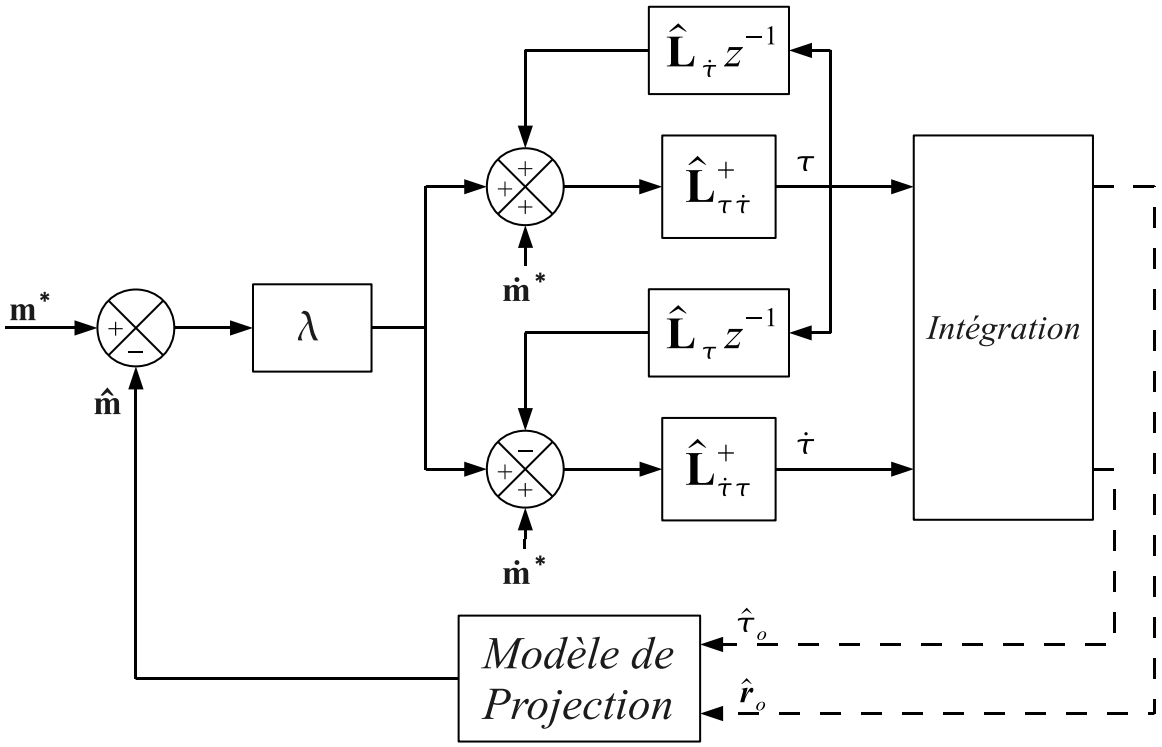


FIGURE 3.5 – Asservissement visuel virtuel pour l'estimation de pose et de vitesse.

3.2.3.2 Version simplifiée

Le schéma de commande présenté précédemment exploite l'ensemble des données disponibles afin de calculer la commande la plus "complète" possible. Cependant, il peut être intéressant de simplifier cette commande soit pour augmenter la vitesse d'exécution de la boucle de régulation en réduisant le nombre de calculs nécessaires soit pour réduire la complexité qui affecte directement le degré de non linéarité de la loi de commande. La seconde partie de l'équation (3.6), qui représente l'anticipation sur la consigne pour la compensation des erreurs de suivi, peut être négligée. Ainsi, l'équation (3.6) devient :

$$\dot{\mathbf{e}} = \mathbf{C}\mathbf{L} \begin{bmatrix} \boldsymbol{\tau} \\ \dot{\boldsymbol{\tau}} \end{bmatrix} \tag{3.32}$$

avec $\mathbf{L} = [\mathbf{L}_{\boldsymbol{\tau}}, \mathbf{L}_{\dot{\boldsymbol{\tau}}}]$

On posant $\mathbf{C} = \widehat{\mathbf{L}}^{+}$ on obtient la commande virtuelle suivante :

$$\begin{bmatrix} \boldsymbol{\tau} \\ \dot{\boldsymbol{\tau}} \end{bmatrix} = -\lambda \widehat{\mathbf{L}}^{+} (\widehat{\mathbf{m}} - \mathbf{m}^*) \tag{3.33}$$

L'estimation de la pose et de la vitesse est alors obtenue par intégration du torseur cinématique et de l'accélération comme définis dans les équations (3.28), (3.29), (3.30) et

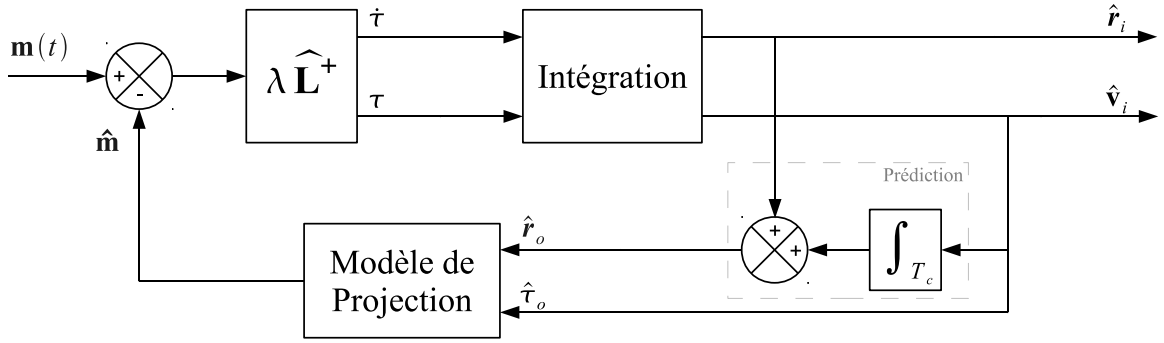


FIGURE 3.6 – Asservissement visuel virtuel simplifié.

(3.31).

Notons que négliger la variation de la consigne induit une erreur de suivi sur la pose et la vitesse. Cependant, ceci n'est pas critique car la commande est virtuelle. Il suffit alors d'exécuter la commande sur un nombre suffisant d'itérations pour assurer la convergence. Pour une implémentation temps réel de cette commande virtuelle, il est cependant intéressant e limiter au maximum le nombre d'itérations nécessaire à la convergence sans devoir augmenter le gain de la commande λ (ce qui a pour effet d'amplifier les bruit dans la boucle de régulation). À cette fin, la pose à l'instant i peut être prédite à partir de la pose et de la vitesse à l'instant $i-1$. L'estimation de la pose est initialisée non pas à la dernière pose estimée mais à la pose prédite :

$$ {}^c\hat{\mathbf{t}}_o(i) = {}^c\mathbf{t}_o(i-1) + \mathbf{v}_{\hat{\tau}} T_e \tag{3.34} $$

$$ {}^c\hat{\mathbf{R}}_o(i) = {}^c\mathbf{R}_o(i-1)\; e^{([\omega_{\hat{\tau}} T_e]_\times)} \tag{3.35} $$

$\mathbf{v}_{\hat{\tau}}$ and $\omega_{\hat{\tau}}$ étant respectivement les vitesses de translation et de rotation estimées $\hat{\boldsymbol{\tau}}_o = [\mathbf{v}_{\hat{\boldsymbol{\tau}}}, \omega_{\hat{\boldsymbol{\tau}}}]^{\mathrm{T}}$.

Les positions des points prédits sont alors obtenues par simple projection à partir des poses prédites ce qui permet d'assurer un meilleur tracking des primitives visuelles dans l'image. Le schéma de commande résultant est donné par la figure 3.6.

3.2.4 Asservissement visuel combiné : Réel-Virtuel

L'implémentation de l'asservissement visuel par acquisition séquentielle requiert une estimation de la pose et la vitesse de l'objet par rapport à la caméra. Dans certaines applications de robotique manipulatrice, le modèle de l'objet par rapport auquel le robot doit se positionner peut être connu. Dans ce cas de figure, il est plus intéressant d'utiliser toutes les informations à disposition afin d'améliorer le comportement de la commande. Dans cet esprit, une commande hybride comportant une boucle de commande réelle ainsi

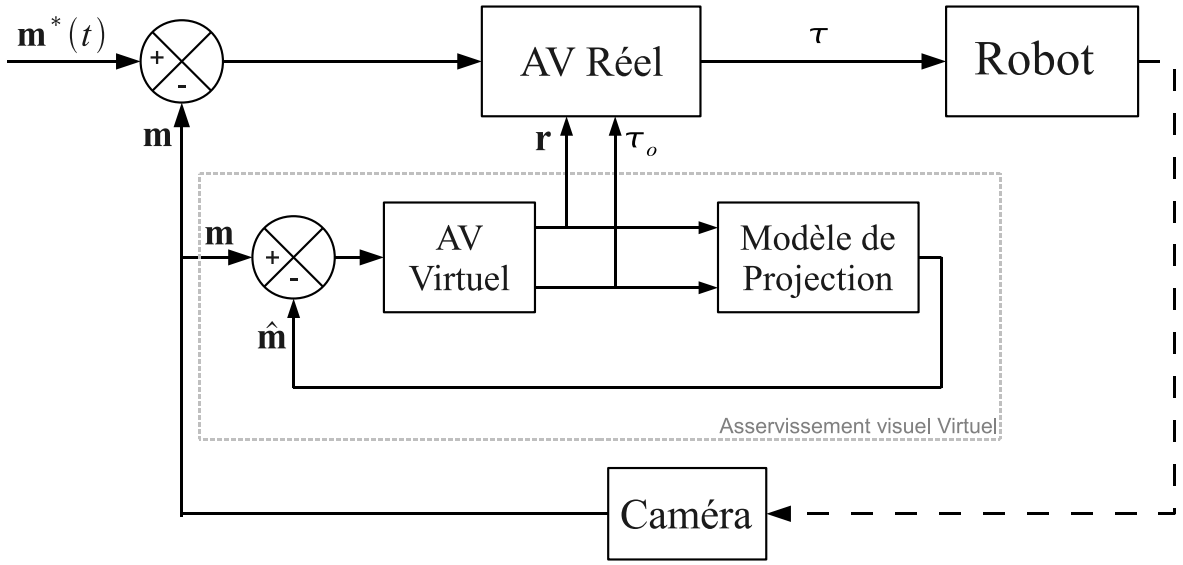


FIGURE 3.7 – Asservissement visuel combiné réel-virtuel par acqsuisition non séquentielle.

qu'une boucle de commande virtuelle peut être mise en place. La boucle virtuelle permet d'estimer la pose et la vitesse relatives entre la cible et la caméra nécessaires au calcul de la matrice d'interaction alors que la boucle d'asservissement réel permet, quant à elle, de réguler le signal dans l'image. Le schéma de commande illustrant cette configuration est présenté en figure 3.7.

Dans ce schéma de commande, le bloc "Commande" représente une des lois de commande présentées au §3.2.2 et le bloc "Cmd Virtuelle" représente l'une des commandes virtuelles présentées en §3.2.3. On peut noter que les erreurs d'estimation de pose et de vitesse pouvant résulter d'éventuelles erreurs sur les paramètres de la caméra ou sur le modèle de l'objet affectent de façon plus conséquente la boucle virtuelle (en raison des erreurs éventuelles sur le modèle de projection) que la boucle de commande. Cependant, l'estimation n'est utilisée que pour le calcul des matrices d'interactions. Par conséquent, le bruit de mesure pouvant affecter l'estimation de la matrice est bien moins nuisible pour la commande que quand c'est le signal de régulation qui est affecté par ce bruit.

3.3 Calcul de la matrice d'interaction

Les deux matrices d'interactions $\mathbf{L}_{\boldsymbol{\tau}}$ et $\mathbf{L}_{\dot{\boldsymbol{\tau}}}$, qui lient respectivement le mouvement des primitives dans l'image aux vitesses et accélérations du robot, jouent un rôle primordial dans la construction des lois de commande présentées dans ce chapitre. Leurs expressions peuvent être obtenues comme dans le cas classique [ECR92] à partir de la dérivée temporelle du modèle de projection (2.5). Le calcul de cette matrice est réalisé en deux étapes : tout d'abord la vitesse des points dans l'espace est exprimée en fonction de la

pose et de la vitesse de l'objet, ce qui correspond à la matrice d'interaction 3D, ensuite, la matrice d'interaction 2D est déduite de cette dernière en exploitant la relation entre la vitesse des points 3D et celle des points dans l'image.

La matrice d'interaction s'obtient de façon triviale à partir de la jacobienne de la fonction de projection présenté dans §2.3.1.

Ainsi, nous pouvons écrire :

$$\frac{d^c\mathbf{P}_i}{dt} = \frac{\partial^c\mathbf{P}_i}{\partial\mathbf{x}}\frac{d\mathbf{x}}{dt} \tag{3.36}$$

$$\mathbf{x} = \begin{bmatrix} {}^c\mathbf{t}_o \\ \theta\mathbf{u} \\ {}^c\mathbf{v}_o \\ {}^c\omega_o \end{bmatrix} \tag{3.37}$$

La dérivée du vecteur $\mathbf{x}$ par rapport au temps peut être définie de façon triviale :

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} \frac{d^c\mathbf{t}_o}{dt} \\ \frac{d\theta\mathbf{u}}{dt} \\ \frac{d^c\mathbf{v}_o}{dt} \\ \frac{d^c\omega_o}{dt} \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{v}_o \\ \mathbf{L}_\omega\omega \\ {}^c\dot{\mathbf{v}}_o \\ {}^c\dot{\omega}_o \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{L}_\omega & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau} \\ \dot{\boldsymbol{\tau}} \end{bmatrix} \tag{3.38}$$

A partir de cette relation et de l'expression de la jacobienne du point 2D (2.59), la dérivée temporelle du modèle de projection est donnée par :

$$\frac{d^c\mathbf{P}_i}{dt} = \left[\mathbf{I}\,, -{}^c\delta\mathbf{R}_i\left[{}^c\mathbf{R}_o{}^o\mathbf{P}_i\right]_\times\mathbf{L}_\omega^{-1}\,, \Delta t_i\,\mathbf{I}\,,\ -\Delta t_i\left[{}^c\delta\mathbf{R}_i{}^c\mathbf{R}_o{}^c\mathbf{P}_i\right]_\times\ \mathbf{L}_{\dot{\omega}}^{-1}\right] \tag{3.39}$$

$$\begin{bmatrix} \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{L}_\omega & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau} \\ \dot{\boldsymbol{\tau}} \end{bmatrix}$$

$$= \left[\mathbf{I}\,,\quad -{}^c\delta\mathbf{R}_i\left[{}^c\mathbf{R}_o{}^o\mathbf{P}_i\right]_\times\ ,\ \Delta t_i\,\mathbf{I}\,,\ -\Delta t_i\left[{}^c\delta\mathbf{R}_i{}^c\mathbf{R}_o{}^c\mathbf{P}_i\right]_\times\ \mathbf{L}_{\dot{\omega}}^{-1}\right] \tag{3.40}$$

$$\begin{bmatrix} \boldsymbol{\tau} \\ \dot{\boldsymbol{\tau}} \end{bmatrix} \tag{3.41}$$

On en déduit directement l'expression de la matrice d'interaction liée à ce modèle de projection :

$$\mathbf{L_P} = \left[\mathbf{I}\,,\quad -{}^c\delta\mathbf{R}_i\left[{}^c\mathbf{R}_o{}^o\mathbf{P}_i\right]_\times\ ,\quad \Delta t_i\,\mathbf{I}\,,\quad -\Delta t_i\left[{}^c\delta\mathbf{R}_i{}^c\mathbf{R}_o{}^c\mathbf{P}_i\right]_\times\ \mathbf{L}_{\dot{\omega}}^{-1}\right] \tag{3.42}$$

On peut aussi distinguer les matrices d'interactions relatives à la vitesse et à l'accélération :

$$\mathbf{L_P} = [\mathbf{L_{P\tau}} \ , \ \mathbf{L_{P\dot{\tau}}}] \tag{3.43}$$

avec

$$\mathbf{L_{P\tau}} = \left[\mathbf{I} \ , \quad -{}^c\delta\mathbf{R}_i \left[{}^c\mathbf{R}_o {}^o\mathbf{P}_i\right]_\times\right] \tag{3.44}$$

$$\mathbf{L_{P\dot{\tau}}} = \left[\Delta t_i \, \mathbf{I} \ , \quad -\Delta t_i \left[{}^c\delta\mathbf{R}_i {}^c\mathbf{R}_o {}^c\mathbf{P}_i\right]_\times \ \mathbf{L}_{\dot{\omega}}^{-1}\right] \tag{3.45}$$

3.3.1 Matrice d'interaction du point 2D

La matrice d'interaction 2D définit ainsi la relation entre le mouvement du point 2D dans l'image et le mouvement de l'objet décrit pas son torseur cinématique et son accélération :

$$\frac{d\mathbf{m}_i(t)}{dt} = \mathbf{L}_{\mathbf{m}_i} \begin{bmatrix} {}^c\boldsymbol{\tau}_o \\ {}^c\dot{\boldsymbol{\tau}}_o \end{bmatrix} \tag{3.46}$$

Cette matrice peut facilement être déduite de la matrice d'interaction 3D comme suit :

$$\frac{d\mathbf{m}_i(t)}{dt} = \frac{\partial \mathbf{m}_i(t)}{\partial {}^c\mathbf{P}_i(t)} \frac{d\,{}^c\mathbf{P}_i(t)}{dt} = \frac{\partial \mathbf{m}_i(t)}{\partial {}^c\mathbf{P}_i(t)} \mathbf{L}_{\mathbf{P}_i} \begin{bmatrix} {}^c\boldsymbol{\tau}_o \\ {}^c\dot{\boldsymbol{\tau}}_o \end{bmatrix} \tag{3.47}$$

où $\dfrac{\partial \mathbf{m}_i(t)}{\partial {}^c\mathbf{P}_i(t)}$ est la matrice bien connue liant le mouvement du point 3D et celui du point 2D donnée par (1.26).

Par identification terme à terme entre les équations (3.47) et (3.46), la matrice d'interaction 2D s'écrit sous la forme :

$$\begin{aligned} \mathbf{L}_{\mathbf{m}_i} &= \frac{\partial \mathbf{m}_i(t)}{\partial {}^c\mathbf{P}_i(t)} \ \mathbf{L}_{\mathbf{P}_i} \\ &= \begin{bmatrix} \dfrac{1}{z_i} & 0 & -\dfrac{x_i}{z_i^2} \\ 0 & \dfrac{1}{z_i} & -\dfrac{y_i}{z_i^2} \end{bmatrix} \mathbf{L}_{\mathbf{P}_i} \end{aligned} \tag{3.48}$$

3.3.2 Calcul de la matrice d'interaction complète

La matrice d'interaction complète relative aux n points de l'image est obtenue par concaténation verticale des matrices pour chaque point dans l'image. On obtient ainsi

une matrice de dimension $2n \times 12$:

$$\mathbf{L_m} = \begin{bmatrix} \mathbf{L_{m_1}} \\ \mathbf{L_{m_2}} \\ \vdots \\ \mathbf{L_{m_n}} \end{bmatrix} \tag{3.49}$$

En pratique, les variables de pose et de vitesse nécessaires au calcul de cette matrice sont estimées à l'itération précédente :

$$\widehat{\mathbf{L}}_{\mathbf{m}}(t_i) = \mathbf{L}\left(\hat{\mathbf{r}}(t_{i-1}), \hat{\boldsymbol{\tau}}_o(t_{i-1})\right) \tag{3.50}$$

Nous avons pu établir ci-dessus l'expression exacte de la matrice d'interaction en fonction de la pose, de la vitesse et du modèle 3D de l'objet. Notons que cette matrice respecte complètement la rigidité de l'objet car elle tient compte de la nature non linéaire de l'espace cartésien dans l'intégration du mouvement. En effet, on peut noter à partir de l'équation (3.45) que le calcul de la rotation de l'objet dépend de l'exponentielle matricielle qui est une fonction non linéaire du temps. Étant donné que les points dans l'image ne sont pas acquis au même moment, la construction de cette matrice requiert l'évaluation de cette fonction pour chaque point ce qui peut poser des problèmes de temps de calcul par rapport à fréquence d'acquisition. Dans les applications temps réel, il n'est pas exclu de faire un compromis entre coût calculatoire et degré de précision. Par exemple, afin d'éviter l'inversion de la matrice d'interaction à chaque itération dans les applications d'asservissement visuel, une solution consiste à choisir la matrice de combinaison comme étant l'inverse de la matrice d'interaction à la configuration désirée ($\mathbf{C} = \mathbf{L}^{*+}$) [ECR92] ou comme étant la transposée de la matrice d'interaction. En vision rapide, en particulier, le temps de calcul est un critère important. Pour des implémentations dans des unités de traitements où la puissance de calcul est limitée tels que les PC, il est intéressant de disposer d'une estimation rapide de la matrice d'interaction sans pour autant affecter de façon significative la vitesse de convergence du processus.

3.4 Estimation approchée de la matrice d'interaction

L'approximation de la matrice d'interaction peut s'effectuer de deux manières. La première consiste à approximer la matrice complète élément par élément à partir de son expression exacte. Cependant, dans cette approche, la précision de l'approximation n'est pas la même pour chaque élément ce qui fait perdre à la matrice résultante son homogénéité. La deuxième méthode consiste à approximer la fonction à partir de laquelle la matrice est dérivée (c.-à-d. le modèle de projection) plutôt que la matrice elle même. Intuitivement, si l'on suppose que le modèle de mouvement simplifié est proche du modèle réel alors leurs

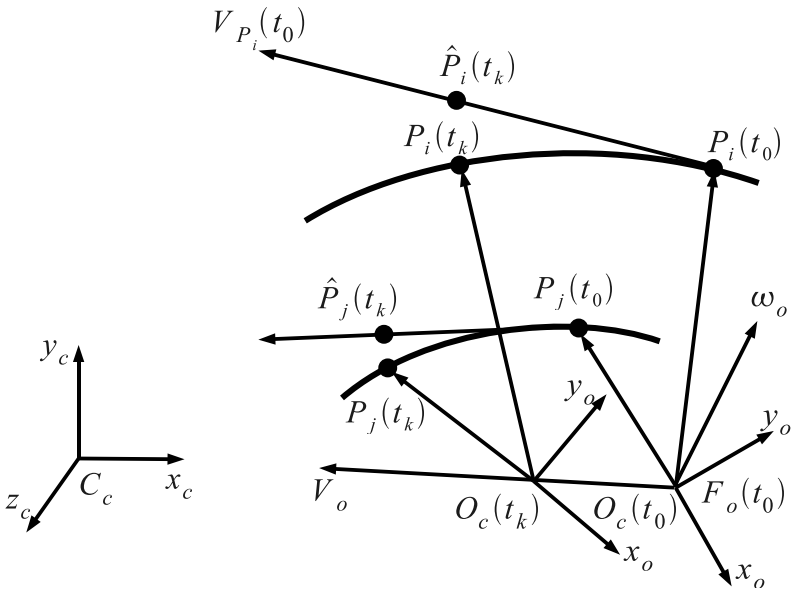


FIGURE 3.8 – Intégration des mouvements des points dans l'espace 3D

matrices respectives doivent l'être aussi. De plus, contrairement à la première méthode, la cohérence entre les éléments est préservée.

3.4.1 Approximation du modèle de mouvement

Soit $\mathbf{F}_o(t)$ le repère lié à un objet en mouvement avec une vitesse de translation ${}^c\mathbf{v}_o$ et une vitesse de rotation ${}^c\omega_o$ par rapport à la camera (Fig. 3.8), et soit $\mathbf{P}_i(t)$ et $\mathbf{P}_j(t)$ deux points de cet objet. Notons que la vitesse relative de l'objet par rapport à la caméra peut être causée soit par le mouvement de l'objet soit par le mouvement de la caméra. Dans la suite, nous considèrerons que le mouvement est dû au déplacement de l'objet pour les mêmes raisons [4] que celles déjà évoquées en §2.2.3.3.

Le développement au premier ordre de l'équation du mouvement du point sur un intervalle de temps Δt_k correspond à une hypothèse de vitesse constante du point dans l'espace sur cette même période. La position et la vitesse d'un point $\mathbf{P}_i(t)$ appartenant à l'objet peut donc être approximée par :

$${}^c\tilde{\mathbf{P}}_i(t_i) = {}^c\mathbf{P}_i(t_0) + \Delta t_i {}^c\dot{\mathbf{P}}_i(t_0) \tag{3.51}$$

$${}^c\tilde{\dot{\mathbf{P}}}_i = {}^c\dot{\mathbf{P}}_i(t_0) + \Delta t_i {}^c\ddot{\mathbf{P}}_i(t_0) \tag{3.52}$$

4. La configuration caméra déportée est la plus probable du fait des fortes accélérations subies par l'outil final qui peuvent endommager les systèmes électroniques et optiques du capteur.

La vitesse et l'accélération du point 3D peuvent s'écrire sous la forme suivante :

$$^c\dot{\mathbf{P}}_i(t) = {}^c\mathbf{v}_o + {}^c\omega_o \times {}^c\mathbf{OP}_i \tag{3.53}$$

$$\begin{aligned} ^c\ddot{\mathbf{P}}_i(t) &= {}^c\dot{\mathbf{v}}_o + 2{}^c\omega_o \times {}^c\mathbf{v}_o + {}^c\dot{\omega}_o \times {}^c\mathbf{OP}_i \\ &+ {}^c\omega_o \times ({}^c\omega_o \times {}^c\mathbf{OP}_i) \end{aligned} \tag{3.54}$$

où $^c\mathbf{OP}_i$ est le vecteur $^o\mathbf{P}_i$ représenté dans le repère de la caméra :

$$^c\mathbf{OP}_i = {}^c\mathbf{R}_o(t_i)\,{}^o\mathbf{P}_i \tag{3.55}$$

Bien que ce modèle de mouvement soit simple (ce qui doit permettre une implémentation temps réel dans un système de vision rapide), il a l'inconvénient de ne pas respecter la contrainte de rigidité de l'objet. En effet, le déplacement d'un ensemble de points est considéré comme rigide si les distances entre ces points restent fixes dans le temps quelque soit la base orthonormée dans laquelle les vecteurs sont écrits :

$$\forall \mathbf{P}_i(t) \in \mathbf{S},\ \forall \mathbf{P}_j(t) \in \mathbf{S},\ \forall t,\ \|\overrightarrow{\mathbf{P}_i(t)\mathbf{P}_j(t)}\| = C^{te} \tag{3.56}$$

Or, comme nous pouvons le voir dans la figure 3.8, ce modèle de mouvement ne respecte pas cette contrainte ce qui induit des erreurs sur le calcul de la position des points 3D et 2D. En effet, l'intégration du mouvement des points $\mathbf{P}_i(t_0)$ et $\mathbf{P}_j(t_0)$ entre les instants t_0 et t_k conduit respectivement aux points $\mathbf{P}_i(t_k)$ et $\mathbf{P}_j(t_k)$ dans le cas d'un modèle de mouvement rigide (déplacement du repère de l'objet) et aux points $\hat{\mathbf{P}}_i(t_k)$ et $\hat{\mathbf{P}}_j(t_k)$ dans le cas de l'approximation du premier ordre de mouvement (vitesses constantes des points dans l'espace cartésien). Le mouvement n'est rigide que si la distance entre les deux points est conservée. Or, cette condition n'est vérifée que si les deux points en question ont la même vitesse instantanée. Ce qui signifie, comme le montre l'équation (3.53) que la vitesse de rotation est nulle.

Notons cependant que le problème de rigidité ne se pose que dans le cas dynamique et non dans le cas statique. En réalité, une réflexion similaire peut être menée dans le contexte de l'asservissement visuel classique où la relation entre la vitesse des points et le torseur cinématique définie par la matrice n'est vérifiée qu'à un instant donné, alors que la vitesse est appliquée durant toute la période d'échantillonnage. Cette difficulté liée à une linéarisation locale de la fonction de projection peut donc s'interpréter comme un non respect de la rigidité de l'objet. En effet, l'application des vitesses instantanées des points durant la période d'échantillonnage conduit, théoriquement, à une déformation de l'objet. Par conséquent, l'utilisation de la matrice d'interaction approximée ne pose pas plus de problèmes au regard de la contrainte de rigidité que la matrice d'interaction classique.

En outre, ce modèle de mouvement, utilisé uniquement afin de réaliser une estimation rapide de la matrice d'interaction, engendre des erreurs du second ordre qui modifient de façon souvent insignifiantes les directions de convergence.

3.4.2 Approximation de la matrice d'interaction

Les équations (3.53) et (3.54) peuvent s'écrire sous forme matricielle en fonction du torseur cinématique et de l'accélération de l'objet ($^c\boldsymbol{\tau}_o$ and $^c\dot{\boldsymbol{\tau}}_o$) exprimés dans le repère caméra :

$$^c\dot{\mathbf{P}}_i(t) = \mathbf{L}_{3d_i}\, ^c\boldsymbol{\tau}_o \tag{3.57}$$

$$^c\ddot{\mathbf{P}}_i(t) = [\boldsymbol{H}_{\mathbf{P}_i},\ \mathbf{L}_{3d_i}] \begin{bmatrix} ^c\boldsymbol{\tau}_o \\ ^c\dot{\boldsymbol{\tau}}_o \end{bmatrix} \tag{3.58}$$

où $\mathbf{L}_{3d_i}$ est la matrice d'interaction classique du point 3D :

$$\mathbf{L}_{3d_i} = \begin{bmatrix} -\mathbf{I}\ , & [^c\mathbf{OP}_i]_\times \end{bmatrix}$$

et $\boldsymbol{H}_{\mathbf{P}_i}$ représente l'accélération du point 3D en fonction de la vitesse de l'objet :

$$\boldsymbol{H}_{\mathbf{P}_i} = \begin{bmatrix} 2\,[^c\omega_o]_\times\ , & \left[[^c\mathbf{OP}_i(t)]_\times\, ^c\omega_o\right]_\times \end{bmatrix}$$

En substituant (3.57) et (3.58) dans (3.52), la vitesse du point 3D peut s'écrire en fonction du torseur cinématique et de l'accélération de l'objet :

$$\hat{\dot{\mathbf{P}}}_i(t + \Delta t_i) = \mathbf{L}_{\hat{\mathbf{P}}_i} \begin{bmatrix} ^c\boldsymbol{\tau}_o \\ ^c\dot{\boldsymbol{\tau}}_o \end{bmatrix} \tag{3.59}$$

où $\mathbf{L}_{\hat{\mathbf{P}}_i}$ est la matrice d'interaction 3D donnée par :

$$\mathbf{L}_{\hat{\mathbf{P}}_i} = -\left[\mathbf{L}_{3d_i} + \Delta t_i \boldsymbol{H}_{\mathbf{P}_i}, \Delta t_i \mathbf{L}_{3d_i}\right] \tag{3.60}$$

Comme espéré, cette dernière expression est bien plus simple que celle obtenue avec le modèle de mouvement exact donné par (3.44) et (3.45).

Pour compléter le calcul comme dans (3.48), la matrice d'interaction 2D est obtenue par :

$$\mathbf{L}_{\hat{\mathbf{m}}_i} = \begin{bmatrix} \dfrac{1}{z_i} & 0 & -\dfrac{x_i}{z_i^2} \\ 0 & \dfrac{1}{z_i} & -\dfrac{y_i}{z_i^2} \end{bmatrix} \mathbf{L}_{\hat{\mathbf{P}}_i} \tag{3.61}$$

Enfin, la matrice d'interaction complète d'un ensemble de points est obtenue par concaté-

nation verticale de chaque matrice d'interaction relative à la primitive qui lui correspond :

$$\mathbf{L}_{\widehat{\mathbf{m}}} = \begin{bmatrix} \mathbf{L}_{\widehat{\mathbf{m}}_1} \\ \mathbf{L}_{\widehat{\mathbf{m}}_2} \\ \vdots \\ \mathbf{L}_{\widehat{\mathbf{m}}_n} \end{bmatrix} \tag{3.62}$$

Nous pouvons remarquer que, contrairement à la matrice exacte, l'expression de la matrice approchée ne dépend pas explicitement du modèle 3D de l'objet, ce qui laisse à penser qu'elle est mieux adaptée dans les applications où le modèle 3D de la cible n'est pas bien connu. Par ailleurs, le fait que les deux matrices dépendent de la vitesse ne pose pas de problème majeur car une estimation de la vitesse est disponible aussi bien en mesure (vitesse estimée) qu'en commande (vitesse de commande).

3.5 Conclusion

Nous avons abordé dans ce chapitre le problème de la synthèse de lois de commande à partir de primitives visuelles acquises à des instants différents. Ainsi, à partir de la fonction de tâche qui présente la particularité de dépendre de la pose et de la vitesse entre la caméra et la cible, différents schémas de commande présentant des propriétés différentes ont été proposés. En effet, nous avons vu que le calcul du torseur cinématique qui permet d'effectuer le suivi de trajectoires rapides dans l'image pouvait se calculer soit à partir de la matrice d'interaction liant le mouvement des primitives aux vitesses du robot, soit à travers l'intégrale de l'accélération calculé directement à partir des erreurs de projections dans l'image, soit en combinant les deux approches précédentes. De plus, nous avons vu que le formalisme développé pouvait aussi être utilisé pour l'estimation de pose et de vitesse. Enfin, la matrice d'interaction correspondant au point a été présentée et une méthode d'approximation rapide de cette matrice permettant d'augmenter la vitesse de calcul sans affecter la stabilité ou la précision de la commande a été proposée.

Chapitre 4

Commande dynamique par vision rapide

4.1 Introduction

Dans le chapitre précédent, nous nous sommes intéressés à la commande cinématique des robots par vision rapide. Comme nous avons pu le voir, les lois de commande proposées peuvent être exploitées aussi bien pour la commande que pour l'estimation d'état (pose et vitesse de la plate-forme). La commande cinématique présente cependant certaines limitations en termes de performances. En effet, les commandes cinématiques ont le "défaut" de négliger les effets de la dynamique du robot. Ces efforts représentent une perturbation dans la boucle de commande. Pour les robots à faible inertie et à faible vitesse, ces perturbations sont généralement faibles, donc assez bien rejetées par la boucle de régulation "bas niveau". Cependant, ces efforts sont d'autant plus importants que la vitesse de déplacement du robot ou son inertie sont élevées. Dans ce cas, il devient nécessaire de compenser ces perturbations afin d'obtenir de meilleures performances.

Dans la perspective d'améliorer les performances de la commande par vision pour des robots rapides, ce chapitre se propose d'étendre l'application de la vision rapide en général, dont l'acquisition par fenêtres d'intérêt en particulier, à la commande dynamique en couple des robots par vision. De plus, nous avons vu en §1.5 que la mesure par vision est, d'un point de vue conceptuel, mieux appropriée à la commande des robots parallèles que la mesure articulaire. La difficulté est alors d'effectuer la mesure dans l'espace cartésien. Aussi, nous allons voir dans ce chapitre que la vision rapide par acquisition séquentielle de fenêtres d'intérêt est une approche intéressante à la commande dynamique des robots parce qu'elle permet aussi bien d'augmenter la fréquence d'acquisition que d'estimer la pose et la vitesse. Ainsi, deux types de commandes seront proposés :

– Dans le premier type (commande 3D), le vecteur de commande, qui représente le couples envoyés aux actionneurs, est calculé à partir des trajectoires cartésiennes

désirées de la plate-forme et des pose et vitesse de cette dernière estimées par les algorithmes proposés dans les chapitres précédents.
- Le second type de commande permet d'obtenir une décroissance du second ordre de la fonction de tâche à partir d'une régulation dans l'image.

Dans ce qui suit, nous allons tout d'abord nous intéresser à la synthèse de commandes dynamiques 3D à partir de l'acquisition séquentielle de fenêtres d'intérêt. Une analyse de la stabilité des schémas de commande proposés en fonction des gains du correcteur et des facteurs d'amortissement de l'estimateur nous permettra, d'une part, de démontrer la stabilité en boucle fermée du schéma de commande et, d'autre part, de fournir les outils adéquats pour effectuer les différents réglages de la commande. Ensuite, nous verrons comment on peut utiliser le vecteur de commande dans l'estimation de mouvement en vue d'améliorer le comportement de la commande. Enfin, une commande en couple calculé par vision rapide référencée image sera présentée et comparée à la commande 3D.

4.2 Utilisation de l'estimation de pose et de vitesse par vision rapide pour la commande dynamique

Notons que le critère évoqué en §1.5 qui impose une vitesse d'acquisition élevée peut théoriquement être satisfait grâce au mode d'acquisition par fenêtres d'intérêt. Par ailleurs, les trois méthodes d'estimation de pose et de vitesse présentées précédemment (la minimisation des moindres carrés, l'asservissement visuel virtuel ainsi que le filtrage de Kalman étendu) respectent toutes la contrainte qui préconise que l'estimation de vitesse soit obtenue autrement que par dérivation numérique afin d'obtenir une mesure correcte de la vitesse. En effet, dans l'estimation de la pose et de la vitesse par minimisation des moindres carrés (§2.3, page 61) et par asservissement visuel virtuel (§3.2.3, page 88), partageant fondamentalement le même principe de minimisation de fonction de coût, la vitesse est obtenue respectivement par intégration de l'erreur de vitesse et d'accélération. Dans l'estimation par le filtrage de Kalman étendu, les paramètres de vitesse font partie du vecteur d'état et sont donc mis à jour à partir du modèle d'évolution et des nouvelles mesures. Une solution directe pour construire une commande en couple calculé par vision rapide est de substituer la mesure de la configuration cartésienne de la pose et la vitesse de la plate-forme par l'une des méthodes proposées d'estimation par vision dans le schéma de commande (Fig. 1.11, page 44). Cette méthode conduit au schéma de commande de la figure (4.1).

Il reste cependant à satisfaire le critère fondamental du point de vue de l'automatique non linéaire : garantir un temps de latence minimal sur la reconstruction de pose et de vitesse par vision.

En effet, l'estimation de la pose et de la vitesse peut être affectée par un retard avec

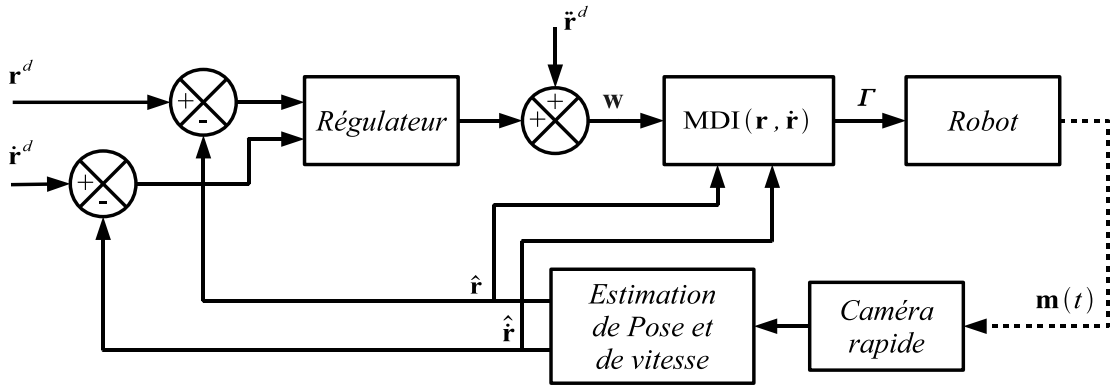


FIGURE 4.1 – Commande en couple calculé par vision rapide.

deux origines principales. La première est le temps d'acquisition des images, de transmission des données et de traitement. Cependant, la vitesse de calcul des microprocesseurs actuels associées à l'acquisition séquentielle de fenêtres d'intérêt, permet une réduction significative de ce temps. La seconde source de retard est inhérente aux méthodes d'estimation du fait de leur temps de convergence. En effet, les méthodes d'estimation par minimisation non linéaire (tel que l'asservissement visuel virtuel) requièrent souvent plusieurs itérations pour converger vers la solution désirée. Or, faute de temps, il est possible que l'implémentation temps réel de ce type d'algorithmes ne puisse effectuer le nombre d'itérations nécessaires pour atteindre la précision souhaitée, ce qui se traduit par une erreur de suivi des paramètres. En effet, la vitesse de calcul dépend principalement du système sur lequel l'algorithme est implémenté. Ainsi, dans une implémentation sur un système temps réel dédié, souhaitable pour des commandes dynamiques, la puissance de calcul d'un système correctement dimensionné permet d'effectuer un nombre important d'itérations afin d'assurer la convergence de l'algorithme. Cependant, dans une implémentation sur un système du type PC la puissance de calcul est assez limitée. Dans ce cas, l'estimation peut en effet ne pas converger assez rapidement.

De la même manière que pour l'estimation par asservissement visuel virtuel, l'algorithme de Kalman étendu peut aussi induire des effets de retard sur l'estimation. En effet, l'algorithme de Kalman étendu est en réalité un filtre qui, comme la plupart des filtres, introduit un déphasage. Par ailleurs, la solution calculée, est une moyenne entre la prédiction, calculée sur la base du modèle d'évolution, et la nouvelle mesure pondérée par les matrices de covariances qui traduisent le poids (degré de confiance) des deux estimations. Ainsi, un modèle d'évolution comportant un bruit de moyenne non nulle se traduit par le calcul d'une solution non optimale dont la moyenne de l'erreur d'estimation n'est pas nulle. Ceca a pour conséquence d'introduire du retard ou de l'avance sur la solution. Une première conclusion de cette discussion est que, indépendamment de l'architecture déployée, l'effet de retard sur l'estimation est un problème réel du point de vue pratique

et important du point de vue de l'automatique. Celui-ci nécessite donc une attention particulière dans la synthèse de commande. C'est pourquoi il est nécessaire de faire une étude de la stabilité de la commande dynamique face aux erreurs de suivi des paramètres de pose et de vitesse et au temps de latence.

4.3 Étude de la stabilité de la commande dynamique face aux retards sur l'estimation

Si les commandes dynamiques sont plus performantes que les commandes cinématiques, elles ont en contrepartie l'inconvénient d'être bien plus sensibles au temps de latence. En réalité, la stabilité des commandes dynamiques par vision a déjà fait l'objet d'une étude [CG93, CG96]. Cependant, cette étude ne fait intervenir que le temps de latence dû au temps d'acquisition et de traitement car le système considéré ne possédait que deux degrés de liberté (rotation en pan-tilt), la tête portant la caméra étant asservie dans l'image à partir d'un seul point. Notons que pour calculer la fonction de transfert qui permet d'analyser les performances et d'étudier la stabilité à partir de la position des pôles, l'auteur effectue une approximation du premier ordre de la fonction de transfert. De plus, dans cette étude, la vitesse a été obtenue par dérivation de la configuration estimée du robot. Ainsi, comme la vitesse n'est pas filtrée, les latences se répercutent de façon similaire sur la pose et la vitesse ce qui permet de garder une certaine cohérence temporelle entre ces deux mesures (avec l'inconvénient d'amplifier le bruit de mesure et de le propager sur la vitesse). Or, dans les méthodes d'estimation basées sur l'acquisition séquentielle de fenêtres d'intérêt, les estimations de la pose et la vitesse ne convergent pas de la même manière. Ainsi, la pose et la vitesse estimées peuvent avoir des retards différents à la même itération, ce qui peut être nuisible pour la commande. C'est pourquoi nous nous proposons d'étudier la stabilité du système dynamique face à des déphasages différents pour la pose et la vitesse qui seront représentés par des fonctions de transfert bien distinctes (Fig. 4.2).

Notons que notre objectif n'est pas de faire une étude complète de la stabilité et de la robustesse de la commande. Il s'agit de donner un aperçu du domaine de stabilité en fonction des différents gains. Cela nous permettra d'une part, de démontrer la stabilité de la commande et d'autre part, de fournir des éléments pour orienter le choix des gains de la commande.

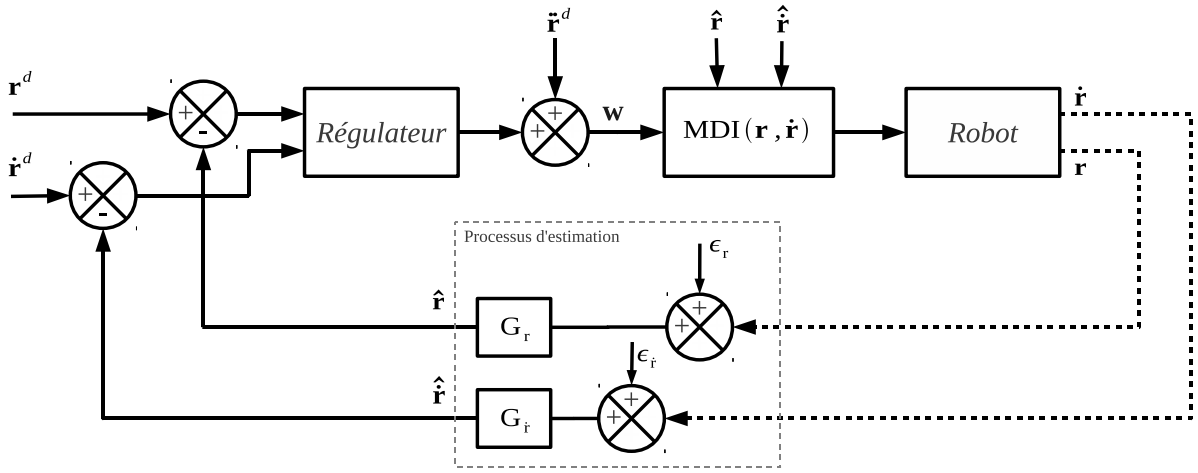


FIGURE 4.2 – Commande dynamique équivalente

4.3.1 Modélisation des différents éléments de la commande

4.3.1.1 Modélisation de la boucle de commande

Avant d'étudier la stabilité, il faut d'abord établir le modèle des différents blocs qui constituent le système, notamment : le robot lui-même, l'acquisition des données, le processus d'estimation et le correcteur. Si la modélisation des deux premiers éléments matériels (robot et caméra) est imposée par leur modèle physique, les deux autres éléments sont purement logiciels et ne dépendent ainsi que de la solution choisie tant pour l'estimation que pour la régulation.

Pour l'étude de la stabilité du système, le robot à m degrés de liberté commandé en couple sera représenté par m double intégrateur. Cette hypothèse signifie que le système est considéré comme étant localement linéarisé et découplé. Si la stabilité d'un tel système est démontrée, les éventuelles erreurs de modélisation et les perturbations qui peuvent exister en pratique peuvent être compensées par le correcteur en prenant soin de garder une certaine marge de stabilité sur le réglage des gains. Ainsi, l'étude de la stabilité de ce système, qui est de type MIMO (multi input-multi output), revient à étudier, de manière indépendante, les différentes grandeurs à réguler. Dans ce cas, il suffit d'étudier la stabilité de m systèmes SISO (single input-single output). Cependant, comme ces systèmes partagent le même schéma de commande, alors l'analyse de stabilité d'un seul système SISO est valable pour toutes les entrées-sorties.

4.3.1.2 Modèle du correcteur

Les deux correcteurs de base le plus souvent utilisés dans les commandes dynamiques sont les correcteurs PD et PID. La régulation PD associée à une anticipation est théorique-

ment le correcteur le plus simple dans dans le cas d'une commande en couple où le MDI estimé est suffisamment précis particulièrement par rapport aux erreurs statiques. En effet, les frottements secs liés au mouvement des moteurs, des réducteurs et des articulations sont difficilement identifiables. De plus, leur comportement peut évoluer au cours du temps du fait de l'usure. En pratique, une action intégrale peut être ajoutée en vue de réduire ces erreurs sans les compenser complètement du fait de la nature non linéaire des frottement. Par ailleurs, l'action intégrale augmente l'ordre du système ce qui rend ce qui rend sa commande plus délicate. Ainsi, le correcteur PID associe trois actions :

- Proportionnelle : pour la compensation des erreurs mesurées par la boucle de retour
- Dérivée : pour la compensation des erreurs dynamiques
- Intégrale : pour la compensation des erreurs statiques

Afin de rester générique il est préférable de traiter les deux choix de régulation. Néanmoins, vu que le correcteur PD peut être considéré comme un sous système du correcteur PID, il suffit d'établir le modèle de ce dernier. Ensuite, la fonction de transfert du correcteur PD peut être obtenue en annulant simplement l'action intégrale (gain intégral nul).

La sortie du correcteur est constituée de la somme des trois actions mentionnées précédemment pondérées respectivement par les gains K_p, K_I et K_d. Les fonctions de transfert en z respectives de ces actions sont :

$$H_p = K_p \tag{4.1}$$

$$H_I = K_I \frac{T_c}{1 - z^{-1}} \tag{4.2}$$

$$H_d = K_d \frac{1 - z^{-1}}{T_c} \tag{4.3}$$

où H_P, H_I et H_d sont respectivement les fonctions de transfert en z des composantes proportionnelle, intégrale et dérivée. Notons que ces composantes sont calculées séparément car, comme la vitesse n'est pas obtenue par dérivation numérique, différents blocs qui représentent le processus d'estimation vont s'ajouter en amont des blocs proportionnels et dérivés.

Par ailleurs, il faut noter que le dérivateur numérique amplifie d'autant plus les bruits de mesure que la période d'échantillonnage est faible, c'est pourquoi il est parfois associé à un filtre passe-bas. Cependant, cette démarche n'est pas très utilisée en robotique car les filtres numériques introduisent un déphasage qui peut perturber la commande au point de la déstabiliser [Lon06]. L'action du filtre est dans ce cas plus néfaste que bénéfique [1].

1. Il existe cependant une méthode qui consiste à construire un système multifréquences où la période de mesure est réglée de façon à avoir une fréquence d'acquisition beaucoup plus importante que la fréquence de commande. Le déphasage qui est introduit par le filtrage est alors important par rapport à la fréquence d'acquisition mais faible par rapport à la fréquence de commande, ce qui permet de réduire

Néanmoins, la mesure articulaire est souvent suffisamment précise pour se dispenser du filtrage. En revanche, la mesure par vision peut être plus bruitée que la mesure articulaire. L'utilisation d'un filtre sur la vitesse estimée peut alors s'avérer nécessaire.

En pratique, l'action intégrale peut aussi poser des problèmes quand l'erreur devient trop importante (on parle dans ce cas de "charge" ou accumulation de l'erreur). Cela peut perturber fortement la commande qui doit "décharger" l'erreur accumulée (intégrée). Pour remédier à ce problème, la solution consiste soit à borner la valeur de l'erreur, soit à ajouter un effet d'oubli en limitant le nombre d'échantillons sur lequel est calculée l'intégrale (les n derniers échantillons). Une autre solution consiste à ne pas activer l'action intégrale avant d'être assez relativement proche de la configuration finale.

4.3.1.3 Modélisation des processus d'acquisition et d'estimation

D'après le modèle d'acquisition de la caméra présenté au chapitre II, la caméra acquiert les imagettes à une fréquence plus importante que la fréquence de la boucle d'estimation (Fig. 2.6). Les images acquises sont ensuite exploitées par le processus d'estimation mais pas de la même manière selon la méthode d'estimation utilisée. En effet, il faut rappeler que dans l'estimation par filtrage de Kalman étendu, les données sont traitées à la volée. La mise à jour de la pose est ainsi disponible après l'acquisition de chaque point. Par conséquent, on peut considérer que la pose estimée est disponible immédiatement après acquisition. Cependant, un déphasage peut être induit par une éventuelle erreur de suivi des paramètres de pose et de vitesse due à l'algorithme lui même. Ce déphasage est d'autant plus important que le modèle d'évolution ainsi que le modèle de projection[2] sont entachés d'erreurs. Cependant, comme le processus est stochastique dépendant des matrices de covariance, il est relativement difficile de proposer un modèle réaliste de la transformée en z de ce processus même en faisant l'hypothèse de linéarité locale de la matrice d'observation $\boldsymbol{C}$ du filtre de Kalman (qui représente la Jacobienne $\mathbf{J}$ de la fonction de projection par rapport à l'état). C'est pourquoi nous nous limiterons dans cette étude au cas de l'estimation par des méthodes d'optimisation non linéaire.

Dans le cas de l'estimation par asservissement visuel virtuel, les données sont empilées puis traitées (Fig. 2.7, page 68). Par ailleurs, l'avantage de l'acquisition séquentielle et que les primitives sont acquis de façon progressive. Ainsi, à l'instant d'acquisition de la dernière primitive, toutes les autres primitives ont déjà été envoyées et ,éventuellement, ont été traitées. Cela réduit considérablement le temps de latence dû à l'acquisition et au traitement. Par conséquent, comme le processus d'acquisition est normalement plus rapide que le processus d'estimation, une implémentation parallèle de ces deux processus (acquisition et traitement) est assuré dans un délai d'une période contrairement à

l'effet du filtrage sur la boucle de commande.

2. Rappelons que dans l'estimation par filtrage de Kalman étendu, que le modèle de projection ne dépend pas de la vitesse.

l'acquisition globale qui nécessite deux périodes.

La pose est ainsi estimée avec un retard pur d'un échantillon. De plus, l'estimation par asservissement visuel virtuel induit à son tour, de la même manière que pour le filtre de Kalman étendu, un déphasage entre l'état réel et l'état estimé. Cependant, contrairement à l'estimation par filtrage de Kalman étendu, il est possible de calculer la transformée en z du processus d'estimation avec l'hypothèse de linéarité locale, ce qui permettra d'étudier la stabilité de commande dynamique en boucle fermée.

Pour calculer cette transformée, rappelons que l'équation récursive qui décrit l'évolution de l'état estimé (pose vitesse) est donnée par (voir §2.3) :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \Delta\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1} - \lambda\ \mathbf{J}^+\ (\widehat{\mathbf{m}} - \mathbf{m}) \tag{4.4}$$

où $\widehat{\mathbf{m}}$ et $\mathbf{m}$ représentent respectivement la position des primitives dans l'image obtenue par le modèle de projection à partir des paramètres de pose et de vitesse et les primitives observées par la caméra vers lesquelles l'algorithme d'estimation doit converger.

Pour de faibles erreurs dans l'image, ce qui est le cas lors d'un suivi de trajectoire, on peut écrire :

$$\widehat{\mathbf{m}} - \mathbf{m} = \widehat{\mathbf{J}}(\hat{\mathbf{x}} - \mathbf{x}) \tag{4.5}$$

où $\mathbf{x}$ représente la trajectoire réelle du robot [3].

Dans le cas où l'estimation de $\widehat{\mathbf{J}}$ est assez pricise et l'erreur $(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k)$ reste faible, alors, l'équation (4.4) peut ainsi être approximée par :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} - \lambda\ \widehat{\mathbf{J}}^+\ \mathbf{J}(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k) = \mathbf{x}_k - \lambda\ (\hat{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) \tag{4.6}$$

Le nombre d'itérations de l'algorithme d'estimation peut cependant être supérieur à 1, ce qui correspondrait à un système multifréquences dans lequel la fréquence de la boucle d'estimation est p fois supérieure à la boucle de commande. Même si cet aspect rend le système plus complexe, il est néanmoins possible de se ramener à un système classique. En effet, il suffit pour cela de calculer la fonction de transfert de la boucle d'estimation pour un nombre donné d'itérations à partir de l'équation récurrente qui exprime la sortie de l'estimateur en fonction de l'entrée et du nombre d'itérations.

La sortie de la boucle d'estimation à l'itération p de l'échantillon d'indice k est donnée par :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} - (1 - (1 - \lambda)^p)(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) \tag{4.7}$$

3. En effet, rappelons que pour l'estimation par asservissement visuel virtuel, cette trajectoire représente la configuration courante du robot vers laquelle l'algorithme doit converger et non la trajectoire cartésienne désirée que le robot doit suivre.

Comme le nombre d'itérations p est constant, on peut écrire :

$$\lambda_e = 1 - (1 - \lambda)^p \tag{4.8}$$

où λ_e peut être considéré comme le facteur d'amortissement pour un système équivalent à une seule itération.

Il faut tout de même signaler qu'en pratique, le fait d'effectuer plusieurs itérations avec un gain λ au lieu d'une seule avec le gain équivalent λ_e améliore la précision de la linéarisation locale qui est d'autant plus exacte que l'erreur entre la solution courante et désirée est faible. En d'autres termes, la linéarisation est plus précise sur le système où l'estimation converge sur plusieurs itérations que pour le système à une itération avec un gain équivalent. De plus, converger sur plusieurs itérations a un effet filtrant sur le bruit de mesure en raison de l'intégration. Par ailleurs, on peut remarquer que la valeur du gain équivalent (4.8) n'est bornée que si et seulement si $0 \leq \lambda \leq 2$ au quel cas λ_e tend vers 1 et donc $\hat{\mathbf{x}}_k$ vers $\mathbf{x}_{k-1}$.

L'équation récurrente (4.7) nous permet finalement d'écrire la fonction de transfert en z de la boucle d'estimation comme suit :

$$\hat{\mathbf{x}}(z) = z^{-1}\hat{\mathbf{x}}(z) - z^{-1}\lambda_e \left(\hat{\mathbf{x}}(z) - \mathbf{x}(z)\right) \tag{4.9}$$

$$\Leftrightarrow \quad (z + \lambda_e - 1)\hat{\mathbf{x}}(z) = \lambda_e \mathbf{x}(z) \tag{4.10}$$

$$\Leftrightarrow \quad \mathrm{F}_{est} = \frac{\lambda_e}{z + \lambda_e - 1} \tag{4.11}$$

4.3.1.4 Modélisation du robot + compensation de la dynamique

Rappelons que l'évolution de l'état d'un robot est obtenue par le modèle dynamique qui s'écrit sous la forme d'un système d'équations non linéaires[4] du second ordre dont l'expression générale est donnée dans le cas des robots parallèles par [PLA+08] :

$$\ddot{\mathbf{r}} = \mathbb{M}^{-1}\left(\mathbb{D}^{-T}\left(\Gamma - \mathbb{C}(\mathbf{r}, \dot{\mathbf{r}}) + \mathbb{G}(\mathbf{r}) + \sum_{j=1}^{k} \mathbb{J}_{jp}^{T}\mathbb{J}_{j}^{-1}\mathbb{H}_j\right)\right) \tag{4.12}$$

Comme le modèle du robot est non linéaire, il n'est pas possible de calculer sa fonction de transfert en z. Cependant, le rôle d'une commande en couple calculé est justement de linéariser et de découpler le système en estimant les efforts qu ?exercent les différents éléments du robot (corps, moteurs, liaisons, etc.) sur les actionneurs afin de les compenser.

4. Sauf pour certains robots particuliers qui présentent des propriétés de découplage cinématique où, dans ce cas, les équations peuvent être linéaires.

Ces efforts sont obtenus à de l'estimation du modèle dynamique (4.13) [KI07a] :

$$\Gamma = \widehat{\mathbb{D}}^T \left(\widehat{\mathbb{M}}\mathbf{w} + \widehat{\mathbb{C}}(\hat{\mathbf{r}}, \dot{\hat{\mathbf{r}}}) + \widehat{\mathbb{G}}(\hat{\mathbf{r}}) + \sum_{j=1}^{k} \widehat{\mathbb{J}}_{jp}^T \widehat{\mathbb{J}}_j^{-1} \widehat{\mathbb{H}}_j \right) \tag{4.13}$$

où le symbole $\hat{\bullet}$ signifie que la valeur est estimée.

Notons que cette équation, représentant le modèle dynamique inverse, permet de calculer de façon directe les couples articulaires découplant le système à partir de l'état courant de la plate-forme et du vecteur.

Ainsi, l'accélération du robot (4.12) en réponse aux couple obtenu par (4.13) peut s'écrire sous la forme suivante :

$$\ddot{\mathbf{r}} = \mathbf{w} + \eta \tag{4.14}$$

où η représente la perturbation appliquée à la commande lié aux erreurs d'estimation des paramètres dynamiques du robot :

$$\begin{aligned} \eta &= \left(\mathbb{M}^{-1} \mathbb{D}^{-T} \widehat{\mathbb{D}}^T \hat{\mathbb{M}} - \mathbf{I} \right) \mathbf{w} \\ &+ \mathbb{M}^{-1} \left(\mathbb{D}^{-T} \widehat{\mathbb{D}}^T \widehat{\mathbb{C}}(\hat{\mathbf{r}}, \dot{\hat{\mathbf{r}}}) - \mathbb{C}(\mathbf{r}, \dot{\mathbf{r}}) \right. \\ &+ \mathbb{D}^{-T} \widehat{\mathbb{D}}^T \widehat{\mathbb{G}}(\hat{\mathbf{r}}) - \mathbb{G}(\mathbf{r}) \\ &+ \left. \sum_{j=1}^{k} \left(\widehat{\mathbb{J}}_{jp}^T \widehat{\mathbb{J}}_j^{-1} \widehat{\mathbb{H}}_j - \mathbb{J}_{jp}^T \mathbb{J}_j^{-1} \mathbb{H}_j \right) \right) \end{aligned} \tag{4.15}$$

Par conséquent, le robot peut être modélisé comme étant un double intégrateur auquel on ajoute une perturbation représentant les erreurs de modèle. Évidemment, cette perturbation est d'autant plus faible que les paramètres dynamiques du robot sont estimés avec précision.

Si l'on considère que cette perturbation est négligeable devant le vecteur de commande alors l'accélération de la plate-forme peut être considérée comme étant constante entre deux échantillons successifs. Par ailleurs, la commande requiert une période pour le calcul des couples à partir de la pose et de la vitesse estimée, ce qui se traduit par une latence d'un échantillon sur le vecteur de commande. La fonction de transfert du robot avec une compensation de la dynamique en considérant le retard est ainsi obtenu par la multiplication d'un double intégrateur à un retard pur d'un échantillon :

$$\boldsymbol{H}_{robot} = \left(\frac{T_c}{1 - z^{-1}} \right)^2 z^{-1} = \frac{T_c^2 \, z}{(z-1)^2} \tag{4.16}$$

En réalité, qu'il s'agisse d'un robot sériel ou parallèle, un robot dont la dynamique est compensée peut être représenté par un double intégrateur. Cependant, comme cela a été

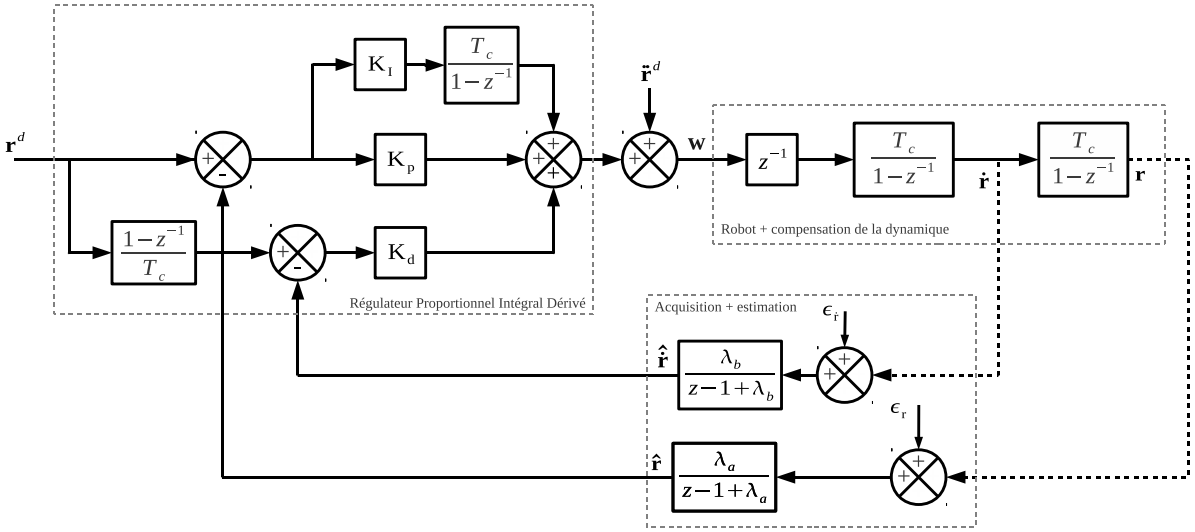


FIGURE 4.3 – Schéma de commande dynamique par vision avec une régulation PID

évoqué au début du chapitre, la mesure par vision est plus appropriée pour la commande dynamique des robots parallèles que pour la commande des robots sériels.

Par ailleurs, le retard dû au temps de calcul influe sur le signal mais aussi sur la linéarisation du système. Mais on peut considérer que dans le cas d'une commande dynamique, la période d'échantillonnage est suffisamment faible pour considérer que la linéarisation locale est tout de même assurée. En effet, il a été démontré qu'il est possible d'actualiser la dynamique à une fréquence bien plus faible que la période d'échantillonnage du signal sans que cela nuise de façon notable aux performances de la commande [MR82].

4.3.2 Analyse de stabilité de la commande avec un correcteur PID

Dans ce schéma de commande, l'estimation de la pose et de la vitesse est assurée par une boucle d'optimisation non linéaire associée à une régulation PID. En combinant les modèles établis précédemment, on obtient le schéma de commande représenté dans la figure 4.3.

La fonction de transfert en boucle fermée du système est dans ce cas donnée par :

$$F_{3dPID} = \frac{\left(\frac{K_d\,(z-1)}{T_c\,z} + K_p + \frac{K_I\,T_c\,z}{z-1}\right) {T_c}^2 z\,(z-1)^{-2}}{\left(1 + \left(\frac{K_d\,(z-1)\lambda_b}{T_c\,z^2(z-1+\lambda_b)} + \frac{K_p\,\lambda_a}{z(z-1+\lambda_a)} + \frac{K_I\,\lambda_a\,T_c}{(z-1+\lambda_a)(z-1)}\right) {T_c}^2 z\,(z-1)^{-2}\right)} \tag{4.17}$$

où λ_a et λ_b sont respectivement les facteurs d'amortissement équivalents de la boucle d'estimation de la pose et de la vitesse.

L'analyse de la stabilité du système consiste dans notre cas à étudier la position de ses pôles (racines du dénominateur) en fonction des paramètres de la commande (λ_a, λ_b, K_p, K_I et K_d).

Après quelques calculs, on peut montrer que le dénominateur de la fonction de transfert (4.17) peuvent respectivement s'écrire comme suit [5] :

$$\begin{aligned} D_{3dPID} &= z^6 + (\lambda_b - 5 + \lambda_a)\, z^5 + (\lambda_b\lambda_a - 4\lambda_b + 10 - 4\lambda_a)\, z^4 \\ &+ \left(K_I\lambda_a {T_c}^3 - 3\lambda_b\lambda_a + K_p\lambda_a {T_c}^2 + 6\lambda_a + 6\lambda_b + T_cK_d\lambda_b - 10\right) z^3 \\ &+ (3\lambda_b\lambda_a + 5 - 4\lambda_b - 3T_cK_d\lambda_b + T_cK_d\lambda_b\lambda_a - 4\lambda_a \\ &+ K_p\lambda_a {T_c}^2\lambda_b - K_I\lambda_a {T_c}^3 - 2K_p\lambda_a {T_c}^2 + K_I\lambda_a {T_c}^3\lambda_b)\, z^2 \\ &+ \left(\lambda_a + 3T_cK_d\lambda_b - 2T_cK_d\lambda_b\lambda_a - \lambda_b\lambda_a + \lambda_b - 1 + K_p\lambda_a {T_c}^2 - K_p\lambda_a {T_c}^2\lambda_b\right) z \\ &+ T_cK_d\lambda_b\lambda_a - T_cK_d\lambda_b \end{aligned} \tag{4.18}$$

La condition nécessaire et suffisante pour que le système en boucle fermée soit stable est que tous les pôles soient à l'intérieur du cercle unité. Étant donné que l'ordre du système est de $6(> 4)$, la résolution par radicaux de cette équation en fonction des différents paramètres n'est pas possible [Gal46]. Néanmoins, il existe des méthodes permettant de calculer numériquement les racines d'un polynôme quel que soit sont degré à travers les valeurs propres de sa matrice compagnon.

Si la vérification de la stabilité d'un tel système est simple (par calcul numérique des racines), il est assez difficile de faire une analyse détaillée et générique de sa stabilité en fonction des différents paramètres. En effet, d'une part l'absence d'une solution algébrique au problème des racines ne permet pas une étude formelle sur la position de ces derniers, et d'autre part, le nombre important des paramètres (cinq) rend difficile les études numériques où géométriques de la stabilité. Néanmoins, il est possible d'analyser le cas particulier, mais aussi le plus répandu, du réglage du PID dans une commande en couple calculé.

En effet, les commandes PID sont généralement réglées de façon à obtenir un pôle triple réel négatif ce qui correspond à la réponse la plus rapide sans dépassement. Pour

5. Les équations étant relativement complexes, l'expression du dénominateur a été obtenue en utilisant un logiciel de calcul symbolique (Matlab + Symbolic Math).

ce choix de réglage[6], la valeur des gains est donnée par [KD99] :

$$K_I = \omega_0^3 \tag{4.19}$$

$$K_p = 3\omega 0^2 \tag{4.20}$$

$$K_v = 3\omega_0 \tag{4.21}$$

où ω_0 est un réel correspondant à la position du pôle triple.

Ayant imposé les deux contraintes traduites par les trois équations précédentes, le nombre de paramètres est réduit à trois $(\omega_0, \lambda_a, \lambda_b)$. Il est ainsi possible de visualiser géométriquement le volume à l'intérieur duquel la commande est stable dans l'espace des paramètres. Pour cela il suffit de calculer la surface qui correspond à la limite de stabilité. Le principe de la méthode consiste à calculer le gain ω_0 en fonction de λ_a et λ_b pour lequel un des pôles du système a une norme unitaire. Cette surface est obtenue de façon numérique en calculant les gains maximal et minimal pour chaque couple de paramètres définis dans un intervalle déterminé. Étant donné que le gain minimal obtenu correspond à zéro sur tout l'intervalle, il suffit donc de ne tracer que le gain maximal. On obtient ainsi la figure 4.4. La commande est donc stable pour tout $0 < \omega_0 < \omega_{0max}$.

Tout d'abord, nous constatons à partir de cette figure que le gain limite est très faible ou nul pour des valeurs très petites ou très importantes des gains λ_a et λ_b. Cela s'explique par le fait que des valeurs trop faibles introduisent un retard important sur l'estimation à cause du temps de convergence important pour de faibles facteurs d'amortissement. Par ailleurs, dès que ces derniers dépassent la valeur limite qui est 2, le gain passe directement à zéro. En effet, les facteurs d'amortissement sont typiquement choisis inférieurs à 1. Une valeur supérieure à l'unité conduit à un dépassement de la solution désirée, alors qu'une valeur supérieur à 2 ne permet pas de converger comme nous avons pu le voir en §4.3.1.3. Il est aussi intéressant de noter que si le gain limite de la commande décroît progressivement quand les facteurs d'amortissement s'approchent de zéro, une valeur du gain proche de la limite 2 permet de conserver un gain important du correcteur. Ceci s'explique par le fait que le gain entre 1 et 2 n'introduit ni de retard ni d'avance puisque la valeur estimée "oscille" autour de la solution ce qui à pour effet d'introduire du bruit mais pas du déphasage. On peut en conclure que du point de vue de la stabilité, il vaut mieux avoir un facteur d'amortissement important même s'il laisse passer du bruit que d'avoir un facteur d'amortissement faible qui peut déstabiliser la commande.

Par ailleurs, on peut voir sur la même figure une arête droite correspondant à un gain maximal. Cela signifie que même si des facteurs d'amortissement plus faibles introduisent

6. Il serait éventuellement possible de faire la synthèse d'un correcteur qui permettrait d'améliorer le comportement du système mais, compte tenu de la complexité relative de cette tâche pour ce type de systèmes (système d'ordre 6), ce travail sortirait du cadre de ce chapitre. C'est pourquoi on se contentera d'utiliser un correcteur classique.

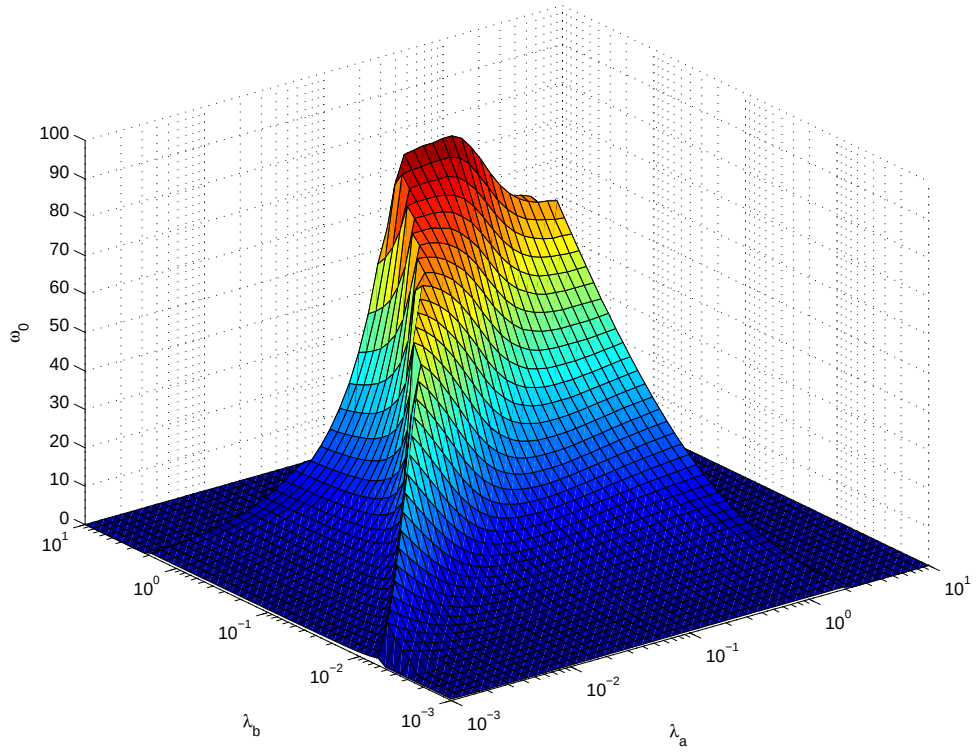


FIGURE 4.4 – Limite de stabilité de la commande dynamique par vision avec régulation PID en fonction des facteurs d'amortissement équivalents de la boucle d'estimation.

un déphasage plus important, réduire ces facteurs peut améliorer la marge de gain. En effet, il est plus important de garder un déphasage relatif acceptable entre la pose et la vitesse que de retard minimal sur l'un des deux. Le rapport entre les deux gains correspond à $\frac{\lambda_b}{\lambda_a} = 5.25$ ce qui signifie que dans cette plage de variation le paramètre de vitesse λ_b doit être supérieur au paramètre de pose λ_a. Le gain maximal, est atteint pour un couple $(\lambda_a, \lambda_b) = (0.25, 1.9)$. Il faut tout de même noter que ce réglage ne correspond pas forcément à la meilleure performance dynamique du système mais celui qui permet d'avoir le gain le plus important. Par ailleurs, le fait que la surface ne soit pas symétrique par rapport au plan $\lambda_a = \lambda_b$ et qu'elle soit décalée vers les λ_b positifs signifie que la commande a tendance à être plus sensible à des retards sur la vitesse qu'à des retards sur la pose.

4.3.3 Analyse de stabilité de la commande avec un correcteur PD

L'étude du domaine de stabilité de la commande proposée avec un correcteur proportionnel dérivé (PD) se fait de la même manière que pour le correcteur PID en annulant le gain intégral ($K_I = 0$).

La fonction de transfert en boucle fermée pour la commande avec correcteur PD est ainsi donnée par :

$$F_{3dPD} = \frac{\left(\frac{K_d(z-1)}{T_c z} + K_p\right) {T_c}^2 z\,(z-1)^{-2}}{\left(1 + \left(\frac{K_d(z-1)\lambda_b}{T_c z^2(z-1+\lambda_b)} + \frac{K_p\lambda_a}{z(z-1+\lambda_a)}\right) {T_c}^2 z\,(z-1)^{-2}\right)} \tag{4.22}$$

Le dénominateur de cette fonction de transfert est donné par :

$$\begin{aligned} D_{PD} &= z^5 + (\lambda_a - 4 + \lambda_b)\, z^4 + (-3\lambda_a + 6 + \lambda_b\lambda_a - 3\lambda_b)\, z^3 \\ &+ \left(-4 - 2\lambda_b\lambda_a + 3\lambda_b + K_p\lambda_a {T_c}^2 + 3\lambda_a + T_c K_d \lambda_b\right) z^2 \\ &+ \left(-2T_c K_d\lambda_b + T_c K_d\lambda_b\lambda_a - \lambda_a - K_p\lambda_a {T_c}^2 + \lambda_b\lambda_a - \lambda_b + K_p\lambda_a {T_c}^2\lambda_b + 1\right) z \\ &+ T_c K_d\lambda_b - T_c K_d\lambda_b\lambda_a \end{aligned} \tag{4.23}$$

La surface délimitant les gains sur la figure 4.5 est calculée de la même manière que pour la régulation PID. Ce que nous pouvons constater de particulier dans cette figure par rapport à la précédente est que dans cette commande plus le facteur λ_b est important (mais sans dépasser la valeur limite de 2) plus le gain maximal augmente, et ce, sur tout le domaine de stabilité. Le rapport entre le retard introduit par l'estimation de la pose et celui de l'estimation de la vitesse semble ainsi être moins important que pour un correcteur PID. De plus, comme pour la commande précédente mais de façon plus flagrante, la commande est bien plus sensible au retard sur la vitesse (correspondant à un facteur d'amortissement faible) qu'au retard sur la pose. On peut aussi noter que pour

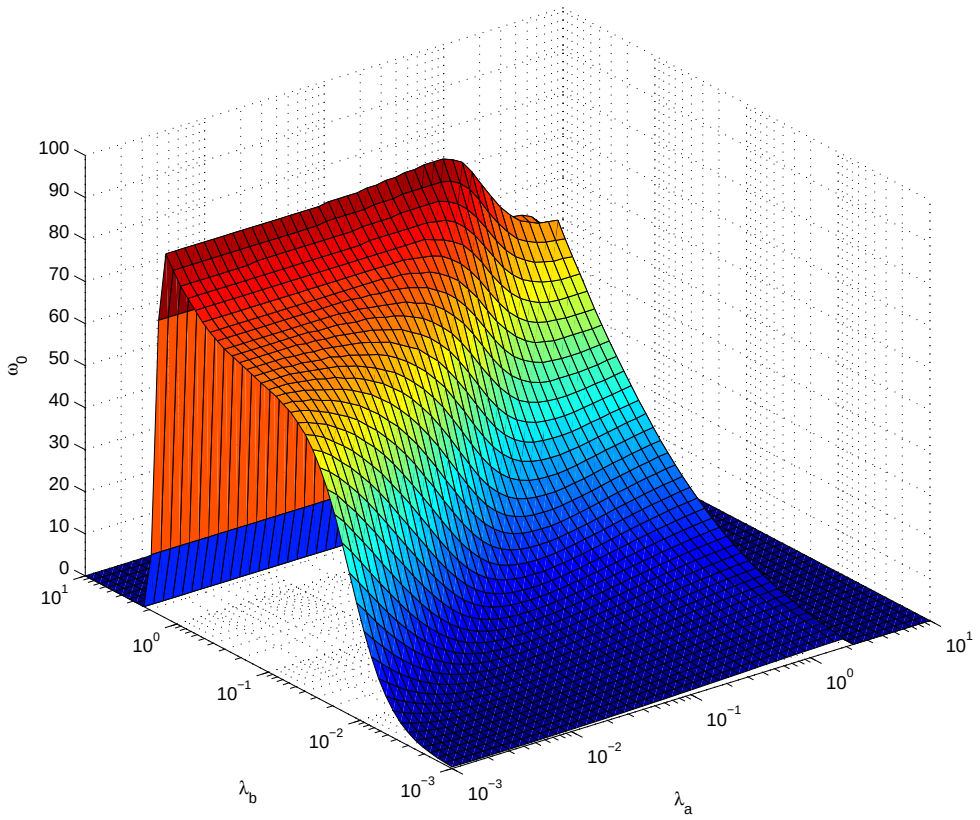


FIGURE 4.5 – Limite de stabilité de la commande dynamique par vision avec régulation PD en fonction des facteurs d'amortissement équivalents de la boucle d'estimation.

$\lambda_b < 5\lambda_a$ la valeur des gains à la limite de stabilité est sensiblement la même pour les deux correcteurs.

4.3.3.1 Discussion sur le réglage des paramètres

Il y a plusieurs critères qui doivent être pris en compte lors du réglage des paramètres du correcteur et de la boucle d'estimation. Côté estimation, comme il existe toujours des erreurs plus ou moins importantes dans les paramètres du modèle de projection (paramètres de la caméra, de la mire, etc.), l'estimation est toujours entachée d'erreurs. La propagation de ces erreurs est d'autant plus importante que les facteurs d'amortissement sont élevés. Par ailleurs, il faut que les facteurs d'amortissements soient réglés de façon à assurer une mesure avec un faible retard aussi bien sur la pose que sur la vitesse afin de ne pas compromettre la stabilité du système. Côté correcteur, il faut que les gains soient, à la fois, assez élevés pour d'assurer la régulation et suffisamment bas pour ne pas propager le bruit de mesure dans la commande et, bien entendu, ne pas dépasser le gain maximal au delà duquel la commande devient instable.

La démarche proposée pour le réglage des gains consiste à ajuster tout d'abord les facteurs d'amortissement de façon à obtenir une mesure avec un bruit acceptable pour la commande. Après quoi, le réglage du gain de la commande peut être réalisé à l'aide de la fonction de transfert en suivant les méthodes classiques. Les performances de la commande sont ensuite analysées. Si des oscillations apparaissent, alors les gains de l'estimateur doivent être augmentés et les gains de la commande éventuellement baissés. Cette opération ne devrait pas compromettre la stabilité du système. Néanmoins, si cette opération engendre une commande trop bruitée alors les gains de la boucle de commande doivent être baissés. Sinon, les facteurs d'amortissement de la boucle d'estimation doivent être baissés et le calcul des gains de la commande renouvelé. Notons que, comme la vitesse estimée est forcément plus bruitée que la pose, alors le paramètre λ_b devrait être plus faible que λ_a. Or, dans ce cas, la limite de stabilité par rapport aux gains est quasiment la même pour les deux correcteurs PID et PD. Ainsi, il n'y pas de préférence en termes de stabilité entre les correcteurs PID et PD. Cependant, il peut y en avoir une en termes de performance.

4.4 Exploitation du vecteur de commande dans l'estimation d'état

Contrairement à une application de métrologie où l'on n'a pas forcément d'informations sur l'évolution du système autre que la mesure par vision, l'implémentation d'une commande dynamique permet de fournir une information supplémentaire au processus d'estimation (le vecteur de commande qui n'est autre que l'accélération appliquée). Cette

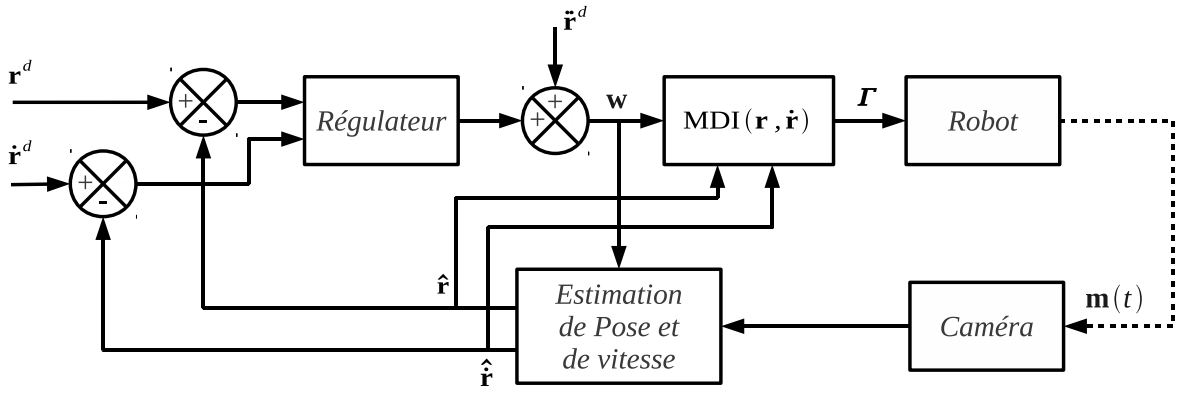


FIGURE 4.6 – Commande dynamique par vision exploitant l'accélération de commande.

information peut ainsi être exploitée par l'algorithme d'estimation pour tenir compte de l'accélération dans le modèle de mouvement (§2.2.3.4) et ainsi étendre le modèle de projection de l'hypothèse de vitesse constante (accélération négligée) à une hypothèse d'une accélération constante (le temps d'une période d'échantillonnage), ce qui est plus réaliste dans une commande dynamique. Le schéma de commande se trouve alors légèrement modifié, comme représenté dans la figure 4.6.

Il est vrai que l'accélération de commande est souvent très bruitée. Faire l'hypothèse que l'accélération est nulle revient finalement à commettre une erreur de 100% sur l'accélération. De plus, cette erreur, qui se répercute directement sur les retards de la pose et de la vitesse estimées, est d'autant plus importante que l'accélération est élevée. Par ailleurs, on peut espérer que même pour un système dont les paramètres dynamiques ont été grossièrement identifiés, l'erreur absolue sur l'accélération prédite par le modèle dynamique soit inférieure au cas où elle est négligée. Ainsi, on peut considérer que dans ce cas, l'utilisation de l'accélération de commande comme entrée du système permet : d'améliorer la précision du modèle de projection ; de réduire les erreurs et les retards qui peuvent affecter la pose et la vitesse estimées et donc ; d'améliorer la stabilité de la commande.

Cette approche est d'autant plus intéressante dans l'estimation d'état par filtrage de Kalman étendue que le changement de modèle de mouvement ne modifie presque pas la méthode d'estimation. En effet, comme nous avons pu le voir dans la section 2.4 La matrice de transition $\boldsymbol{A}$, le modèle de projection (classique) ainsi que la matrice d'observation $\mathbf{C}$ restent inchangés et ne sont donc pas plus complexes que pour un modèle d'évolution à vitesse constante. D'autre part, comme l'accélération est considérée comme une entrée et non comme une perturbation (via la matrice $\boldsymbol{B}$), le modèle d'évolution doit être plus précis et permette une meilleure estimation de l'état. Ainsi, comme pour l'estimation par optimisation non linéaire, l'estimation de l'état devrait être moins affectée par des effets de retard, et ainsi assurer une bonne stabilité de la commande. De plus, étant donné que

la sortie est filtrée, l'état estimé et le vecteur de commande calculé devraient être moins bruités.

4.5 Asservissement visuel dynamique des robots parallèles par vision rapide

Le principal avantage de l'asservissement visuel référencé image est que, contrairement aux approches basées sur une reconstruction dans l'espace cartésien, la régulation du système se fait dans le même espace que la mesure (c.-à-d. l'espace image). Pour cette raison, cette commande est souvent plus robuste face aux erreurs de modèle de l'objet et de la caméra. Cependant, la plupart des lois de commande développées autour de l'asservissement visuel 2D sont des commandes cinématiques obtenues à partir d'un modèle de décroissance exponentielle de la fonction de tâche [ECR92, CH06, CH07]. Le modèle d'évolution de cette dernière correspond à l'ordre du système asservi. En effet, un robot commandé en vitesse peut être modélisé comme étant un simple intégrateur et donc comme un système du premier ordre. Cependant, les équations régissant le système dans une commande dynamique sont typiquement du second ordre. Comme la réponse d'un système du premier ordre est plus lente que la réponse d'un système du second ordre en régime critique pour la même vitesse maximale, est n'est pas intéressant du point de vue des performances d'imposer une décroissance exponentielle du premier ordre au système dynamique. Ainsi, l'approche classique devient dans ce cas inappropriée, ce qui nécessite de développer une commande en imposant un autre type de décroissance de la fonction de tâche.

Par ailleurs, dans une commande cinématique, on définit une seule fonction de tâche qui s'annule lorsque la configuration désirée est atteinte. Cependant, le fait qu'aucune contrainte sur la vitesse ne soit explicitement formalisée se traduit par des erreurs de suivi pour une consigne variable dans le temps[7]. Afin de remédier à ce problème, nous définissons deux fonctions de tâches à 6 éléments chacune. La première a pour rôle de contraindre la position désirée et la seconde de contraindre la vitesse désirée à partir des mesures dans l'image. Notons cependant que cette approche diffère sensiblement de celle adoptée dans les méthodes d'estimation et dans la commande cinématique où l'on avait choisi une seule fonction de tâche à 12 éléments. Cette approche différente se justifie par le fait que contrairement aux méthodes précédentes, le type de réponse souhaité est d'ordre deux.

Nous définissons ainsi les deux fonctions de tâches comme suit :

7. Afin d'atténuer cet effet, une anticipation sur la vitesse est injectée dans la commande. Néanmoins, il ne s'agit que d'une anticipation et non d'une régulation ce qui ne permet pas de compenser les éventuelles erreurs

$$\mathbf{e} = \mathbf{C}(\mathbf{m} - \mathbf{m}^*) \tag{4.24}$$

$$\mathbf{e}' = \mathbf{C}'(\mathbf{m} - \mathbf{m}^*) \tag{4.25}$$

où $\mathbf{C}$ et $\mathbf{C}'$ représentent les matrices de combinaisons.

Rappelons que l'objectif de la commande est de calculer, à partir des mesures dans l'image, l'accélération qui permet d'assurer la convergence du système vers la configuration désirée définie par $\mathbf{m}^*$. Ainsi, l'équation qui nous permet d'obtenir une décroissance du second ordre de la fonction de tâche homogène à l'erreur de position est donnée par :

$$\ddot{\mathbf{e}} + \lambda_v \dot{\mathbf{e}} + \lambda_p \mathbf{e} = 0 \tag{4.26}$$

où λ_p et λ_v sont des matrices diagonales représentant les gains de la commande.

Nous n'utiliserons qu'une équation de contrainte car, comme nous allons le voir par la suite, cette équation permettra de contraindre les deux fonctions de tâches $\mathbf{e}$ et $\mathbf{e}'$ grâce à un choix judicieux des matrices de combinaisons.

Dans une acquisition globale, la dérivée de la fonction de tâche est calculée à partir des vitesses des primitives dans l'image, ce qui nécessite une dérivation numérique. Cela a pour conséquence d ?amplifier le bruit de mesure. La mesure par acquisition séquentielle des primitives dans l'image contient, quant à elle, aussi bien l'information de pose que l'information de vitesse [8]. C'est cette propriété qui rend possible le calcul de l'erreur en position (qui est homogène à une vitesse) et de l'erreur de vitesse (qui est homogène à une accélération) entre les trajectoires réelles et désirées à partir des simples mesures dans l'image (sans estimation des vitesses des primitives). Cette propriété peut ainsi être exploitée dans le cadre de la commande dynamique des robots parallèles. En effet, étant donné que la position des primitives dans l'image dépend aussi bien de la pose que de la vitesse, il s'agit de choisir $\mathbf{C}$ et $\mathbf{C}'$ de façon à ce que les deux fonctions de tâches ne dépendent respectivement que de la pose et que de la vitesse. Ces conditions peuvent être formalisées mathématiquement de la manière suivante :

$$\frac{\partial \mathbf{e}}{\partial \dot{\boldsymbol{\tau}}} = 0 \tag{4.27}$$

$$\frac{\partial \mathbf{e}'}{\partial \boldsymbol{\tau}} = 0 \tag{4.28}$$

8. Les primitives (en nombre suffisant) acquises de façon séquentielle contiennent autant d'informations que de variables dans le modèle de projection. Ainsi, pour un modèle de projection à accélération constante par exemple, une image reconstruite à partir de ce type de primitives contiendrait les informations de position, de vitesse et d'accélération. Cependant, pour une acquisition haute fréquence, l'accélération estimée serait sans doute bien plus bruitée que la pose et la vitesse.

Afin de trouver les matrices de combinaisons qui permettent de satisfaire ces deux conditions,exprimons tout d'abord les dérivées des fonctions de tâche à partir des vitesses et des accélérations. On obtient ainsi :

$$\dot{\mathbf{e}} = \mathbf{C}(\dot{\mathbf{m}} - \dot{\mathbf{m}}^*) = \mathbf{C}\mathbf{L}_{\boldsymbol{\tau}}(\boldsymbol{\tau} - \boldsymbol{\tau}^*) + \mathbf{C}\mathbf{L}_{\dot{\boldsymbol{\tau}}}(\dot{\boldsymbol{\tau}} - \dot{\boldsymbol{\tau}}^*) \tag{4.29}$$

$$\dot{\mathbf{e}}' = \mathbf{C}'(\dot{\mathbf{m}} - \dot{\mathbf{m}}^*) = \mathbf{C}'\mathbf{L}_{\boldsymbol{\tau}}(\boldsymbol{\tau} - \boldsymbol{\tau}^*) + \mathbf{C}'\mathbf{L}_{\dot{\boldsymbol{\tau}}}(\dot{\boldsymbol{\tau}} - \dot{\boldsymbol{\tau}}^*) \tag{4.30}$$

où $\boldsymbol{\tau}^*$ et $\dot{\boldsymbol{\tau}}^*$ représentent respectivement le torseur cinématique et l'accélération correspondant à la trajectoire de consigne.

Les équations (4.29) et (4.30) montrent que pour satisfaire les conditions (4.27) et (4.28), il suffit que :

$$\mathbf{C}\mathbf{L}_{\dot{\boldsymbol{\tau}}} = 0 \tag{4.31}$$

$$\mathbf{C}'\mathbf{L}_{\boldsymbol{\tau}} = 0 \tag{4.32}$$

Par ailleurs, on peut améliorer encore le comportement du système en découplant les éléments du torseur cinématique et de l'accélération en imposant les conditions secondaires :

$$\mathbf{C}\mathbf{L}_{\boldsymbol{\tau}} = \mathbf{I} \tag{4.33}$$

$$\mathbf{C}'\mathbf{L}_{\dot{\boldsymbol{\tau}}} = \mathbf{I} \tag{4.34}$$

Les matrices $\mathbf{C}$ et $\mathbf{C}'$ peuvent être obtenues en exploitant les propriétés de la pseudo-inverse. Pour cela, décomposons la pseudo-inverse de la matrice d'interaction en deux sous matrices :

$$\mathbf{L}_{\mathbf{m}}^{+} = \begin{bmatrix} \mathbf{L}_{sup}^{+} \\ \mathbf{L}_{inf}^{+} \end{bmatrix} \tag{4.35}$$

où $\mathbf{L}_{sup}^{+}$ et $\mathbf{L}_{inf}^{+}$ sont respectivement les sous matrices supérieure et inférieure de dimension $6 \times 2n$ de la pseudo-inverse de la matrice d'interaction.

Ainsi, pour $\mathbf{L}_{\mathbf{m}} = [\mathbf{L}_{\boldsymbol{\tau}}, \mathbf{L}_{\dot{\boldsymbol{\tau}}}]$ nous avons :

$$\mathbf{L}_{sup}^{+}\,\mathbf{L}_{\boldsymbol{\tau}} = \mathbf{I} \tag{4.36}$$

$$\mathbf{L}_{inf}^{+}\,\mathbf{L}_{\boldsymbol{\tau}} = 0 \tag{4.37}$$

$$\mathbf{L}_{sup}^{+}\,\mathbf{L}_{\dot{\boldsymbol{\tau}}} = 0 \tag{4.38}$$

$$\mathbf{L}_{inf}^{+}\,\mathbf{L}_{\dot{\boldsymbol{\tau}}} = \mathbf{I} \tag{4.39}$$

En définissant $\mathbf{C}$ comme une estimation de la partie supérieure de la pseudo-inverse de la matrice d'interaction :

$$\mathbf{C} = \widehat{\mathbf{L}}^{+}_{sup} \tag{4.40}$$

l'équation (4.29) s'écrit :

$$\dot{\mathbf{e}} = \widehat{\mathbf{L}}^{+}_{inf}\mathbf{L}_{\boldsymbol{\tau}}(\boldsymbol{\tau} - \boldsymbol{\tau}^{*}) + \widehat{\mathbf{L}}^{+}_{inf}\mathbf{L}_{\dot{\boldsymbol{\tau}}}(\dot{\boldsymbol{\tau}} - \dot{\boldsymbol{\tau}}^{*}) \tag{4.41}$$

À partir des propriétés (4.36) et (4.38), et sous l'hypothèse que $\widehat{\mathbf{L}}^{+}_{\mathbf{m}}\,\mathbf{L}_{\mathbf{m}} \approx \mathbf{I}$ l'équation (4.41) devient :

$$\dot{\mathbf{e}} = (\boldsymbol{\tau} - \boldsymbol{\tau}^{*}) \tag{4.42}$$

Ainsi, la dérivée de la fonction de tâche ne dépend que du torseur cinématique. Par conséquent, la fonction de tâche est indépendante de l'accélération. De plus, la condition secondaire (4.33) qui revient à découpler les éléments du torseur cinématique est bien vérifiée.

La même démarche peut être appliquée à la seconde fonction de tâche. Ainsi, en posant $\mathbf{C}' = \widehat{\mathbf{L}}^{+}_{inf}$ dans l'équation (4.30) la dérivée de la fonction de tâche $\mathbf{e}'$ s'écrit :

$$\dot{\mathbf{e}}' = \widehat{\mathbf{L}}^{+}_{sup}(\mathbf{m} - \mathbf{m}^{*}) = \widehat{\mathbf{L}}^{+}_{sup}\mathbf{L}_{\boldsymbol{\tau}}(\boldsymbol{\tau} - \boldsymbol{\tau}^{*}) + \widehat{\mathbf{L}}^{+}_{sup}\mathbf{L}_{\dot{\boldsymbol{\tau}}}(\dot{\boldsymbol{\tau}} - \dot{\boldsymbol{\tau}}^{*}) \tag{4.43}$$

A partir des propriétés (4.37) et (4.39) la dérivée de la fonction de tâche $\mathbf{e}'$ est donnée par :

$$\dot{\mathbf{e}}' = (\dot{\boldsymbol{\tau}} - \dot{\boldsymbol{\tau}}^{*}) \tag{4.44}$$

On peut ainsi déduire des équations (4.42) et (4.44) que la fonction de tâche estimée $\dot{\mathbf{e}}'$ n'est autre que la dérivée de la fonction de tâche $\mathbf{e}$ est estimée par :

$$\dot{\mathbf{e}} = \mathbf{e}' \tag{4.45}$$

La dérivée seconde de la fonction de tâche $\mathbf{e}$ peut ainsi s'écrire en fonction de la dérivée de $\mathbf{e}'$:

$$\ddot{\mathbf{e}} = \dot{\mathbf{e}}' = \widehat{\mathbf{L}}^{+}_{inf}(\dot{\mathbf{m}} - \dot{\mathbf{m}}^{*}) = \dot{\boldsymbol{\tau}} - \widehat{\mathbf{L}}^{+}_{inf}\dot{\mathbf{m}}^{*} \tag{4.46}$$

En substituant les équations (4.24), (4.25), (4.45) et (4.46) dans (4.26) on obtient la loi de commande suivante :

$$\dot{\boldsymbol{\tau}} = \widehat{\mathbf{L}}^{+}_{inf}\dot{\mathbf{m}}^{*} + \lambda_v\widehat{\mathbf{L}}^{+}_{inf}(\mathbf{m}^{*} - \mathbf{m}) + \lambda_p\widehat{\mathbf{L}}^{+}_{sup}(\mathbf{m}^{*} - \mathbf{m}) \tag{4.47}$$

En regroupant les deux sous matrices $\widehat{\mathbf{L}}^{+}_{inf}$ et $\widehat{\mathbf{L}}^{+}_{sup}$ et en définissant λ comme étant la

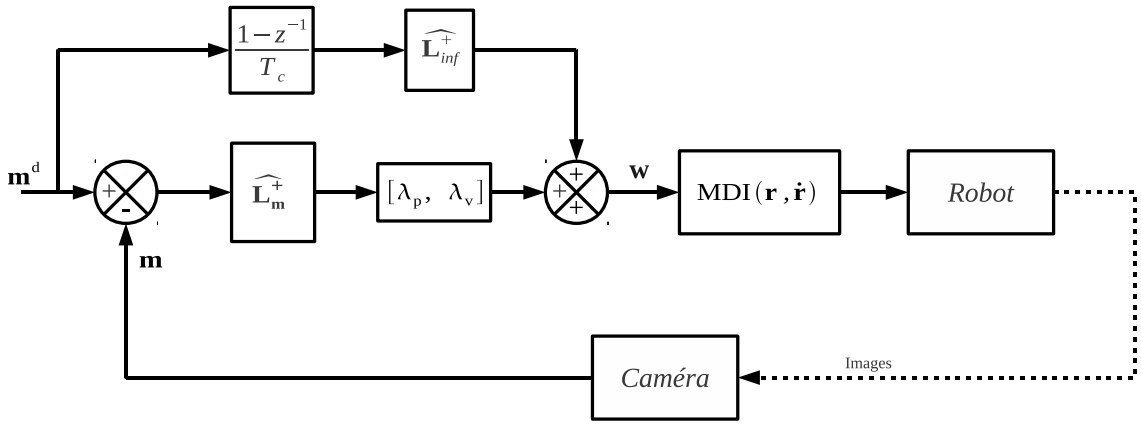


FIGURE 4.7 – Asservissement visuel 2D en couple calculé

concaténation horizontale des matrices de gains $\lambda = [\lambda_p, \lambda_v]$, on obtient une expression plus compacte de l'accélération de commande de l'asservissement visuel dynamique :

$$\dot{\boldsymbol{\tau}} = -\lambda \widehat{\mathbf{L}}^{+}_{\mathbf{m}}(\mathbf{m} - \mathbf{m}^{*}) + \widehat{\mathbf{L}}^{+}_{inf}\dot{\mathbf{m}}^{*} \tag{4.48}$$

ce qui nous conduit au schéma de commande illustré par la figure 4.7.

Il faut noter que le choix de la matrice de gain conditionne l'évolution de la fonction de tâche. Par ailleurs, comme on impose une décroissance du second ordre à la fonction de tâche $\mathbf{e}$ et que $\mathbf{e}'$ est sa dérivée, alors :

$$\mathbf{e} \to 0 \Rightarrow \mathbf{e}' \to 0 \tag{4.49}$$

Comme le calcul de la matrice d'interaction dépend de la pose et de la vitesse de la plate-forme du robot par rapport à la caméra, alors les algorithmes d'estimation de pose et de vitesse présentés précédemment peuvent être utilisés pour le calcul de la matrice d'interaction. Il faut aussi noter que les erreurs de mesure de la pose et de la vitesse n'ont qu'un effet minime sur la précision et la stabilité du système étant donné qu'elles n'affectent que la matrice d'interaction dont la précision n'est pas aussi critique que celle des mesures sur lesquelles est basée la régulation du système (primitives visuelles pour cette commande). Le schéma d'asservissement visuel en couple calculé incluant l'estimation de pose et de vitesse est représenté sur la figure 4.8.

Rappelons que les deux schémas de commande proposés dans cette section reposent sur l'hypothèse que l'estimation de la matrice d'interaction est suffisamment précise pour considérer que $\widehat{\mathbf{L}}^{+}_{\mathbf{m}}\mathbf{L}_{\mathbf{m}} \approx \mathbf{I}$. En réalité cette hypothèse est tout à fait réaliste pour deux raisons principales. La première est que les erreurs de pose et de vitesse sont normalement faibles pour des tâches de suivi de trajectoire. Ainsi, l'actualisation à chaque itération

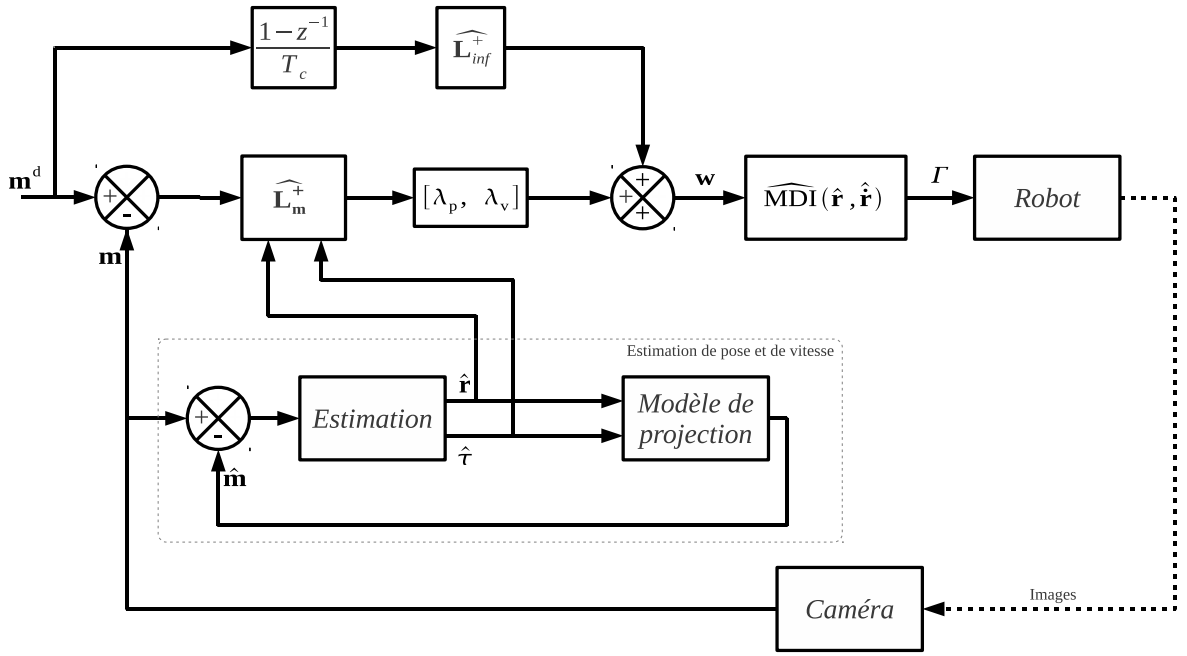


FIGURE 4.8 – Asservissement visuel 2D en couple calculé avec estimation de pose et de vitesse

des matrices de combinaisons devrait assurer une bonne linéarisation de la commande d'autant plus que la fréquence de commande est élevée. La seconde est que l'algorithme d'estimation permet l'obtention d'une mesure précise de la pose et de la vitesse, et donc, une bonne estimation de la matrice d'interaction.

4.6 Comparaison entre la commande dynamique 3D et l'asservissement visuel dynamique 2D

L'un des avantages les plus intéressants de l'asservissement visuel dynamique 2D est que le schéma de régulation n'est constitué que d'une seule boucle. En effet, la boucle d'estimation de ce schéma de commande n'intervenant que dans l'estimation des matrices d'interaction et du calcul de la dynamique contrairement à la commande dynamique 3D où la régulation est assurée à partir de la valeur estimée. Cette propriété permet ainsi d'éliminer les déphasages qui pourraient être introduits par le processus d'estimation améliorant ainsi la stabilité du système. En contre partie, il n'existe pas d'effet filtrant dans le calcul du vecteur de commande comme dans la commande 3D où le bruit est pondéré par les facteurs d'amortissement et intégré dans la valeur estimée (l'intégrale jouant le rôle de filtre). Par conséquent, le bruit de mesure risque d'être amplifié par les gains du correcteur et envoyé au robot. Néanmoins, comme la régulation se fait dans l'image, on peut s'attendre à ce que le bruit affectant le signal de régulation soit plus

Asservissement visuel 3D dynamique	Asservissement visuel 2D dynamique
Schéma de commande dynamique classique (mesure cartésienne).	Schéma de commande inédit.
Génération simple de la trajectoire cartésienne désirée.	Génération plus difficile de la trajectoire dans l'image.
Sensible erreurs de paramètres.	Généralement robuste face aux erreurs de paramètres.
Filtre le bruit.	Ne filtre pas le bruit.
Introduit déphasage.	N'introduit pas de déphasage.
Peut nécessiter plusieurs itérations (coût et vitesse calcul).	Commande calculée en une seule itération.

TABLE 4.1 – Comparaison entre l'asservissement visuel 3D dynamique et l'asservissement visuel 2D dynamique.

faible.

4.7 Conclusion

Différentes commandes en couple calculé basées vision ont été proposées dans ce chapitre. Ces schémas de commande peuvent être classés en deux catégories principales en fonction de leur espace de régulation : la commande dynamique par vision et l'asservissement visuel 2D dynamique. Les variantes de la commande 3D dépendent du choix de l'estimateur de pose et de vitesse de la plate-forme et du type de régulation. Nous avons ainsi pu voir que les performances de cette commande dépendent principalement du déphasage introduit par la boucle d'estimation de la pose et de la vitesse. En outre, nous avons constaté, grâce à l'étude de stabilité, que la commande est bien plus sensible au retard sur la vitesse qu'au retard sur la pose. Ainsi, deux solutions ont été proposées afin d'améliorer la stabilité du système. La première consiste à réduire le déphasage introduit par le processus d'estimation en jouant sur les gains où le nombre d'itérations. La seconde, quant à elle, propose d'utiliser l'accélération de commande comme anticipation sur le mouvement qui devient un mouvement à accélération constante plus réaliste pour une commande dynamique. Dans ce cas, la qualité de l'anticipation dépend de la précision du modèle dynamique du robot. Contrairement à la commande 3D, la commande 2D n'introduit pas de déphasage ce qui laisse présager de bonnes performances dynamiques et de larges marges de gains. De plus, comme toutes les commandes référencées image, cette commande devrait être plus robuste par rapport aux erreurs de modèle (caméra et modèle 3D de l'objet observé).

Chapitre 5

Résultats expérimentaux

Ce chapitre présente les résultats expérimentaux obtenus en estimation de pose et de vitesse de mouvements rapides, en asservissement visuel virtuel ainsi qu'en commande dynamique 2D et 3D d'un robot rapide.

5.1 Dispositif expérimental

Les lois de commande présentées dans ce chapitre ont été implémentées en C++ en utilisant la librairie de calcul numérique NT_2 [FS05] sur un PC Intel P4. L'acquisition est assurée par la caméra "Photon Focus MV-D1024-TrackCam" qui permet d'effectuer une acquisition séquentielle de fenêtres d'intérêt présentée en §2.2.1 sur laquelle est monté un objectif d'une focale de 12mm. Les primitives observées sont des pastilles blanches réfléchissantes sur fond noir. Ce choix de primitives permet d'extraire les positions des primitives par simple calcul de moment au premier ordre sur l'imagette seuillée, ce qui a pour effet de réduire le temps d'extraction de la primitive tout en gardant une précision correcte sur la détection.

Les estimations de la pose et de la vitesse par optimisation non linéaire ou par asservissement visuel virtuel sont basées sur la position de l'ensemble des points dans l'image. Par conséquent, il n'est pas intéressant d'effectuer le calcul après l'acquisition de chaque primitive car le coût calculatoire serait alors bien trop lourd par rapport à la quantité d'information acquise. Pour que la fréquence d'acquisition des fenêtres d'intérêt (qui peut être très élevée) ne soit pas "bridée" par l'estimation (qui requiert un certain temps de calcul) et afin que la quantité d'information acquise (nombre de points acquis) soit suffisamment importante pour relancer le traitement, les processus d'acquisition et de traitement ont été implémentés sur deux tâches différentes (Fig. 2.7).

La tâche principale qui réalise l'acquisition des FI est synchronisée avec la carte de commande du robot par une horloge qui délivre un signal d'une fréquence de 4kHz correspondant à la fréquence d'acquisition des fenêtres d'intérêt (c.-à-d. la fréquence d'acqui-

sition des points). L'ordre d'acquisition des points dans la mire est déterminé de façon aléatoire. A chaque acquisition d'une fenêtre d'intérêt, de dimension 24×24 pixels, la position de la primitive dans l'image est stockée dans une mémoire tampon partagée entre les deux tâches. Après un nombre d'acquisitions déterminé par l'utilisateur (10 dans notre cas), la tâche de traitement est lancée (Fig. 5.2). Comme nous allons le voir, toutes les expérimentations présentées requièrent une estimation de la pose et de la vitesse. Les positions des points à acquérir peuvent alors être prédites à partir de la pose et de la vitesse estimées en utilisant le modèle de projection (2.5). Les fenêtres d'intérêt sont ensuite centrées sur les positions prédites des primitives dans l'image.

Par ailleurs, après l'acquisition de chaque ensemble de primitives, la tâche d'acquisition vérifie que l'estimation est bien terminée afin de garantir la synchronisation entre les deux tâches. Si la tâche d'estimation n'est pas terminée avant la fin de l'acquisition des points, l'expérimentation est interrompue. Comme l'acquisition est synchronisée sur une horloge externe délivrée par un système temps-réel (carte dSpace DS1103), le fait que l'expérimentation se termine normalement confirme que la contrainte temps-réel est vérifiée même si le traitement est implémenté sur un système qui n'est pas temps-réel (système Windows XP). Ainsi, pour un traitement lancé toutes les 10 acquisitions, la fréquence de répétition de la tâche de traitement est de 400 Hz (fréquence d'échantillonnage de la carte de commande du robot).

Avant de procéder aux expérimentations la caméra et la mire ont été étalonner simultanément par ajustement de faisceaux [LVD98]. Les paramètres intrinsèques de la caméra (focale et centre optique) ainsi que les paramètres de distorsion "inverses" qui permettent de corriger la position des points dans l'image ont été estimés. Il faut aussi souligner que les paramètres de distorsion "directs" qui permettent d'appliquer les distorsions ont aussi été identifiés afin d'assurer un suivi précis des primitives à partir des prédictions obtenues par les positions et vitesses cartésiennes mesurées par la méthode d'estimation proposée.

Le manipulateur utilisé dans ces expérimentations est l'Orthoglide de l'IRCCyN [1] [CW03]. C'est est un robot parallèle possédant de bonnes performances dynamiques. En effet, ce robot permet d'atteindre des accélérations allant jusqu'à $10 m/s^2$, ce qui permet de tester les performances de notre approche. En vue des fortes accélérations du robot, la configuration adoptée est la configuration déportée afin de ne pas endommager la caméra. Par ailleurs, même si ce robot ne possède que 3 degrés de liberté en translation, les traitements implémentés (estimation et commande) considèrent les 6 degrés de liberté. Pour l'estimation cela permet de mesurer les éventuelles rotations qui ne sont pas supposées exister. En ce qui concerne la commande, seuls les éléments du vecteur de commande correspondant aux degrés de liberté actifs sont considérés, alors que les autres sont tout simplement ignorés. En outre, considérer tous les degrés de liberté de l'espace permet

1. Nous tenons à remercier l'équipe de recherche de robotique de l'IRRCyN d'avoir mis à notre disposition le robot Orthoglide sans lequel il n'aurait pas été possible d'obtenir ces résultats.

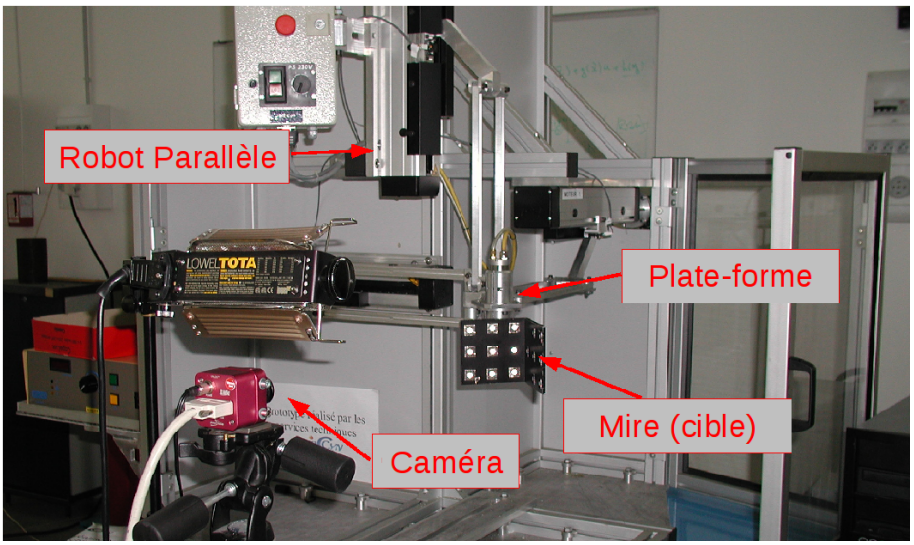


FIGURE 5.1 – Dispositif Experimental montrant le robot parallèle "Orthoglide" sur lequel est monté une mire 3D dont les primitives sont capturées par une caméra rapide permettant l'acquisition séquentielle de fenêtres d'intérêt.

Axe	x	y	z
Écart type position (10^{-5}m)	2.67	4.05	3.45
Écart type vitesse (10^{-3}m/s)	2.04	3.2	5.75

TABLE 5.1 – Écarts types du bruit d'estimation de la position et de la vitesse en statique.

de garder une certaine généricité de la couche de commande par rapport aux types de robots considérés du moment que la matrice cinématique inverse de ce dernier et son modèle dynamique inverse sont définis en fonction de la position et de la vitesse de la plate-forme.

5.2 Caractérisation du système de vision et de la rigidité du robot

Afin d'avoir une interprétation pertinente des résultats, il est nécessaire de caractériser les mesures obtenues par le système de vision et par le robot parallèle en termes de fiabilité et de précision. En effet, même si de nombreux travaux nous rendent confiants sur ce points, il est tout de même important en terme de rigueur de vérifier la fiabilité et la précision des différents moyens de mesure.

Tout d'abord, différentes poses statiques ont été estimées pendant plusieurs secondes à la fréquence de 400Hz afin de caractériser les bruits de mesure et d'estimation. Les écarts-types correspondant aux bruits d'estimation sur la pose et sur la vitesse obtenus sont présentés en Table 5.1.

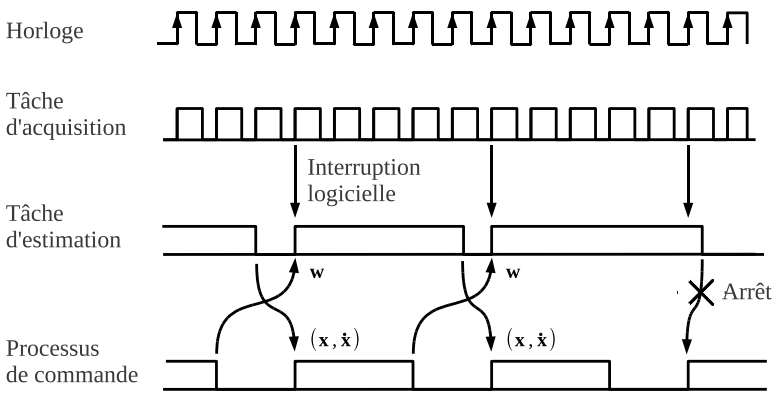


FIGURE 5.2 – Chronogramme des tâches d'acquisition et de traitement (PC) synchronisées avec le processus de commande (carte de commande du robot).

Notons que ces valeurs sont très faibles ce qui signifie que l'estimation de la pose ainsi que l'extraction des points sont stables. Cependant, il peut exister un biais entre les poses réelles et estimées. Néanmoins, la norme moyenne des erreurs de reprojection (0.13 pixel / point) indique que cet éventuel biais est également faible.

En ce qui concerne le robot parallèle, certains jeux et flexibilités dans les articulations ont été remarqués. Pour mesurer l'ampleur des mouvements résultant de ces défauts sur la plate-forme, la position de cette dernière a été mesurée par vision au moment où des efforts manuels y ont été appliqués alors que les actionneurs étaient bloqués (freins engagés). Les déplacements des primitives dans l'image et les déplacements cartésiens estimés par vision sont représentés respectivement sur les figures 5.3 et 5.4. Nous pouvons constater que les primitives dans l'image se déplacent pratiquement de la même manière ce qui signifie que la variation des positions des primitives est, en effet, due à un déplacement de la plate-forme. Ceci est confirmé par la variation des positions cartésiennes qui sont cohérentes avec les mouvements des primitives dans l'image.

Les déplacements maximaux mesurés de la plate-forme dans l'espace cartésien sont de :

$$\delta\mathbf{r} = [10.0\text{mm},\ 8.1\text{mm},\ 7.1\text{mm},\ 2.38^{\text{o}},\ 2.65^{\text{o}},\ 2.93^{\text{o}}]^{\text{T}}$$

et le déplacement moyen des points dans l'image est de $[\delta u,\ \delta v] = [\pm 13,\ \pm 10.1]$ ce qui est beaucoup plus important que les erreurs de reprojection.

Ces résultats montrent que la pose estimée par vision est bien plus précise que la pose calculée par le modèle géométrique direct. On peut aussi souligner que c'est l'un des avantages les plus intéressants dans l'utilisation des capteurs extéroceptifs.

Après la caractérisation de la fiabilité et de la précision du robot et du système de vision, la transformation base-caméra a été obtenue à partir de trajectoires lisses et à faibles vitesses pour éviter l'excitation des jeux et des flexibilités du robot.

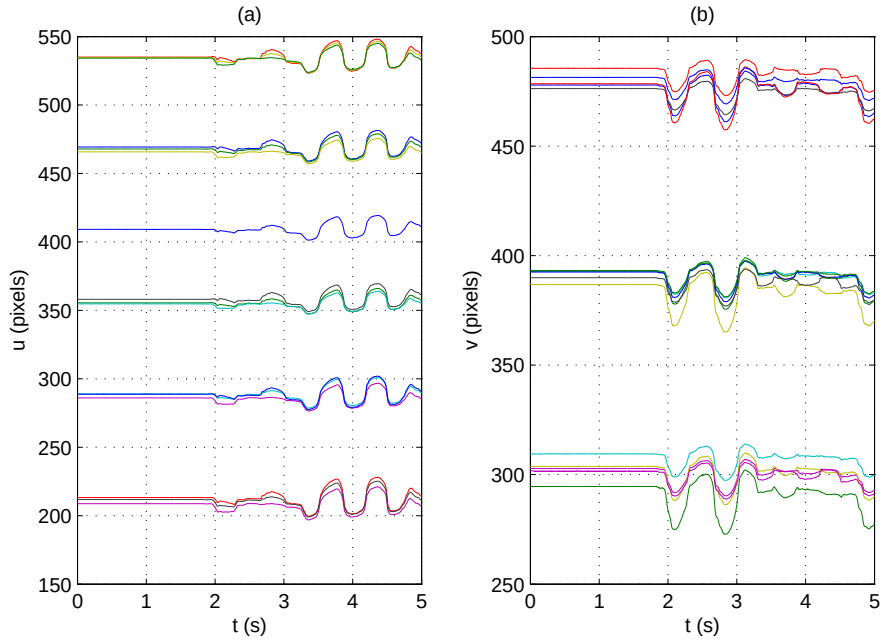


FIGURE 5.3 – Déplacements des primitives dans l'image du fait des efforts manuels appliqués sur la plate-forme lorsque le robots est à l'arrêt (freins engagés).

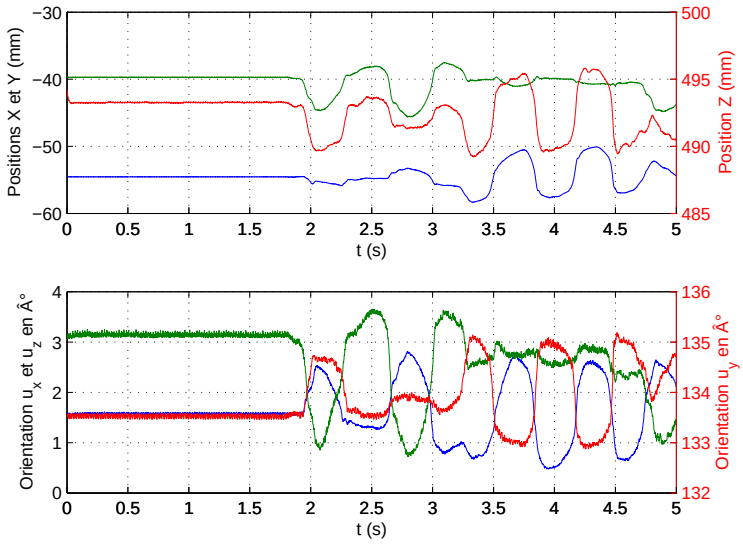


FIGURE 5.4 – Estimation par vision des déplacements de la plate-forme dans l'espace en raison des efforts manuels appliqués sur l'effecteur lorsque le robots est à l'arrêt (freins engagés).

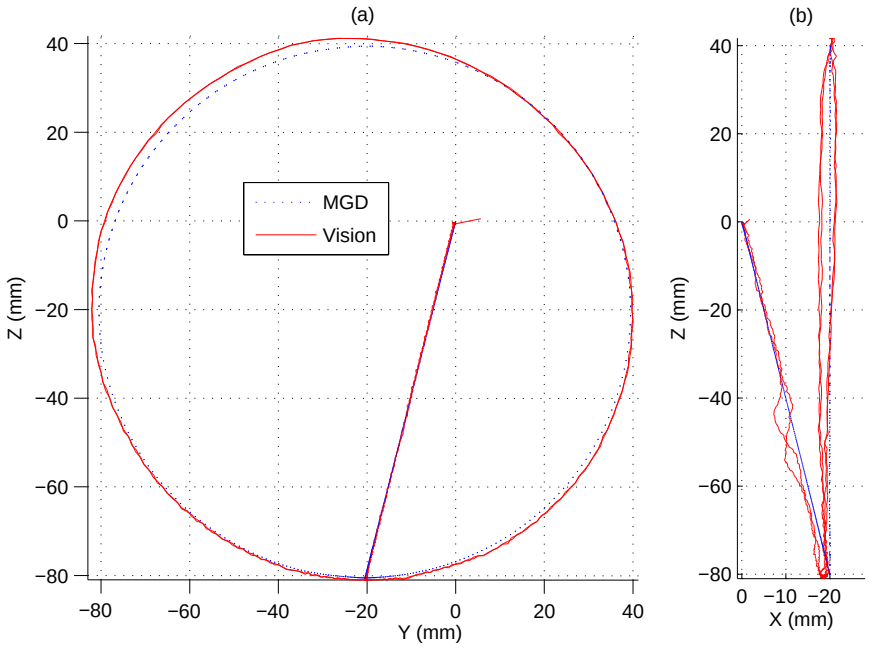


FIGURE 5.5 – Trajectoires de la plate-forme obtenues par vision et par le modèle géométrique direct du robot. (a) Trajectoires projetées sur le plan YZ. (b) Trajectoires projetées sur le plan XZ.

5.3 Estimation de pose et de vitesse

Pour la validation de l'approche proposée, une expérimentation d'estimation de pose et de vitesse de la plate-forme du robot parallèle a été réalisée. Pour évaluer la précision d'estimation de la pose et de la vitesse ainsi que les capacités de suivi de la méthode proposée, le robot a été commandé à 100% de ses performances. La trajectoire de la plate-forme du robot, obtenue par interpolation de degré 5, est composée de déplacements linéaires et circulaires (Fig. 5.5). Dans la première phase du mouvement, le robot se déplace de la position initiale vers le point de départ de la trajectoire circulaire en suivant une ligne droite d'environ 8cm. Dans la deuxième phase, le robot exécute deux trajectoires circulaires identiques d'un rayon de 6cm. Enfin, le robot retourne à sa position initiale en suivant une trajectoire linéaire. La vitesse maximale atteinte sur la partie circulaire de la trajectoire est d'environ $1m/s$ ce qui correspond à une accélération tangentielle de $16.4m/s^2$ et une accélération normale de $16.67m/s^2$.

5.3.1 Choix de la matrice d'interaction

Avant d'entamer les expérimentations, il reste à faire un choix entre les deux matrices d'interaction présentées en §3.3. D'une part, nous avons la forme exacte de la matrice d'interaction liée au modèle de projection et, d'autre part, nous avons une matrice d'interaction approchée (moins précise) mais qui peut être estimée plus rapidement. L'implé-

	Erreurs de reprojection		Erreurs de suivi	
	$\mathbf{L_m}$	$\mathbf{L_{\widehat{m}}}$	$\mathbf{L_m}$	$\mathbf{L_{\widehat{m}}}$
Moyenne	9.73e-2	9.87e-2	1.34e-1	1.35e-1
Variance	6.04e-4	6.37e-4	1.21e-3	1.23e-3
Écart type	2.46e-2	2.52e-2	3.48e-2	3.51e-2

TABLE 5.2 – Erreurs de reprojection et erreurs de suivi dans l'image exprimées en pixels en utilisant la forme exacte de la matrince d'interaction et la forme approximée.

mentation des deux matrices dans l'unité de traitement montre que le temps de calcul de la matrice d'interaction avec la forme exacte est environ 50% plus élevé qu'avec la forme simplifiée ($T_{\mathbf{L_m}} \approx 0.6ms, \quad T_{\mathbf{L_{\widehat{m}}}} \approx 0.4ms$). Ceci fait que la matrice approchée est mieux appropriée pour les applications à haute vitesse. Cependant, l'utilisation de la matrice approchée peut altérer la vitesse de convergence du processus. Pour définir la matrice la plus adaptée, la trajectoire décrite dans le paragraphe précédent a été exécutée et les positions des primitives dans l'image ont été enregistrées. Pour tester les deux matrices dans les mêmes conditions, l'estimation de la pose et de la vitesse ainsi que le suivi dans l'image ont été effectués hors-ligne avec les mêmes données acquises en utilisant les deux matrices en émulant une acquisition réelle. Les résultats ont été rapportés dans le tableau 5.2 où les erreurs résiduelles représentent les erreurs de reprojection ($\mathbf{m}_i - \mathbf{m}_i^*$). L'erreur de suivi est, elle, obtenue en comparant les positions prédites des points dans l'image aux positions mesurées.

Il ressort de ce tableau que les erreurs résiduelles obtenues en utilisant la matrice approchée sont plus importantes que les erreurs obtenues en utilisant la matrice exacte, ce qui était prévisible. Néanmoins, l'erreur obtenue par la matrice approchée ne représente pas plus de 0.015% de l'erreur résiduelle et 0.0075% de l'erreur par rapport à l'utilisation de la forme exacte ce qui n'est pas très significatif. En outre, l'utilisation de la matrice d'interaction approchée permet au processus d'estimation d'atteindre la fréquence de commande du robot qui est de 400 Hz. Étant donné le gain en temps de calcul par rapport à la très faible perte en précision, il semble qu'il est plus intéressant, dans l'état actuel de la puissance de calcul (puissance d'un PC standard) et de la fréquence d'acquisition, d'utiliser la matrice approchée. Ainsi, c'est cette dernière qui a été implémentée dans les expérimentations qui suivent.

5.3.2 Suivi d'un objet rapide avec estimation de la pose et de la vitesse

Cette expérimentation met en œuvre l'estimation de pose et de vitesse de mouvement rapide par la méthode d'asservissement visuel virtuel présentée en §3.2.3.2. La trajectoire décrite ci-dessus a été exécutée par le robot (commande basée modèle) et le mouvement

de la plate-forme a été estimé en temps-réel par le système de vision à la même fréquence que la commande du robot comme décrit en §5.1.

La figure 5.5 montre la trajectoire 3D reconstruite à partir des mesures articulaires en utilisant le MGD "idéal" (c.-à-d. sans estimer les jeux et flexibilités mises en évidence plus haut) et celle obtenue par vision, projetée sur le plan YZ (Fig. 5.5(a)) et XZ (Fig. 5.5(b)) du repère lié au robot. À première vue, les deux trajectoires semblent se confondre. Toutefois, on peut noter que l'estimation de la trajectoire basée vision n'est pas un cercle parfait à cause du dépassement en haut à gauche et en bas à droite par rapport à la trajectoire de référence. L'erreur entre les deux trajectoires peut être attribuée soit à l'estimation basée modèle en raison des flexibilités et des jeux mis en évidence ci-dessus, soit à l'estimation basée vision. Cette erreur peut s'expliquer par le fait que la trajectoire circulaire génère une force centripète qui a tendance à déplacer la plate-forme du robot vers l'extérieur de la trajectoire du fait des défauts dans les articulations. Ces déplacements sont mesurés par vision alors qu'ils n'apparaissent pas dans la trajectoire reconstruite à partir des mesures articulaires. Il est néanmoins important de souligner que l'erreur entre ces deux trajectoires (basée modèle et basée vision) est inférieure à l'amplitude des jeux et flexibilités mesurés. Notons également que dans la trajectoire estimée par vision, les deux cercles sont bien superposés ce qui valide la répétabilité de l'estimation par vision.

La figure 5.6(a) montre la trajectoire cartésienne obtenue par le MGD et la figure 5.6(b) la trajectoire obtenue par le système de vision représentée dans le repère de base du robot. Les deux trajectoires sont lisses et semblent similaires. On peut noter que la composante $\mathrm{X_c}$ de la position mesurée par vision semble la plus bruitée. En fait, la position obtenue à partir de la vision est exprimée dans le repère du robot où l'axe X du robot est confondu avec l'axe Z du repère de la caméra (l'axe correspondant à la profondeur qui est la direction la plus bruitée en vision).

L'erreur entre les deux trajectoires est illustrée sur la figure 5.6(c). Le bruit haute fréquence en statique peut avoir plusieurs sources. La première est issue des erreurs, plus ou moins importantes, sur le modèle 3D de la cible ainsi que sur les paramètres intrinsèques de la caméra. Ces erreurs résiduelles d'étalonnage affectent le modèle de projection et se répercutent sur l'estimation de la pose et de la vitesse. La seconde source a pour origine le bruit de détection des primitives visuelles qui est principalement causé par la variation de la lumière et le bruit dans les images acquises du fait des éventuelles perturbations électromagnétiques, ainsi que par le bruit d'échantillonnage et de quantification dans le processus d'acquisition. L'écart-type du bruit de détection en statique est de 3.174010^{-3} pixel pour un temps d'exposition de $80\mu s$.

Le bruit de détection est difficile à obtenir en dynamique. Toutefois, étant donné que le temps d'exposition est très court, le mouvement ne devrait pas affecter la précision de détection outre mesure. Par exemple, le déplacement d'une primitive, placée à environ 1 m de la caméra et se déplaçant parallèlement au plan de projection à une vitesse de

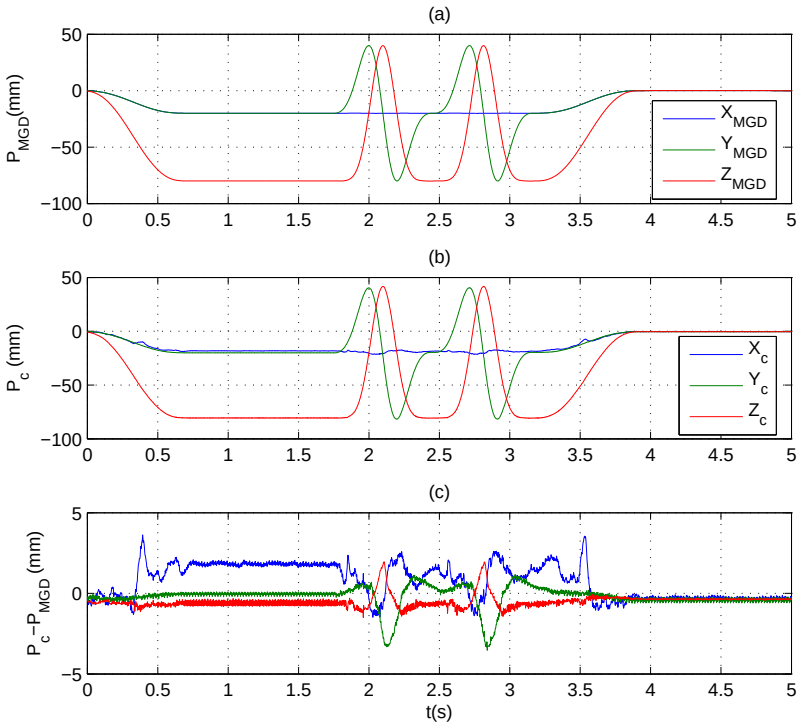


FIGURE 5.6 – Position cartésienne de la plateforme (a) obtenue à partir des mesures articulaires et (b) à partir de l'estimation par vision et (c) l'erreur de position entre les deux estimations.

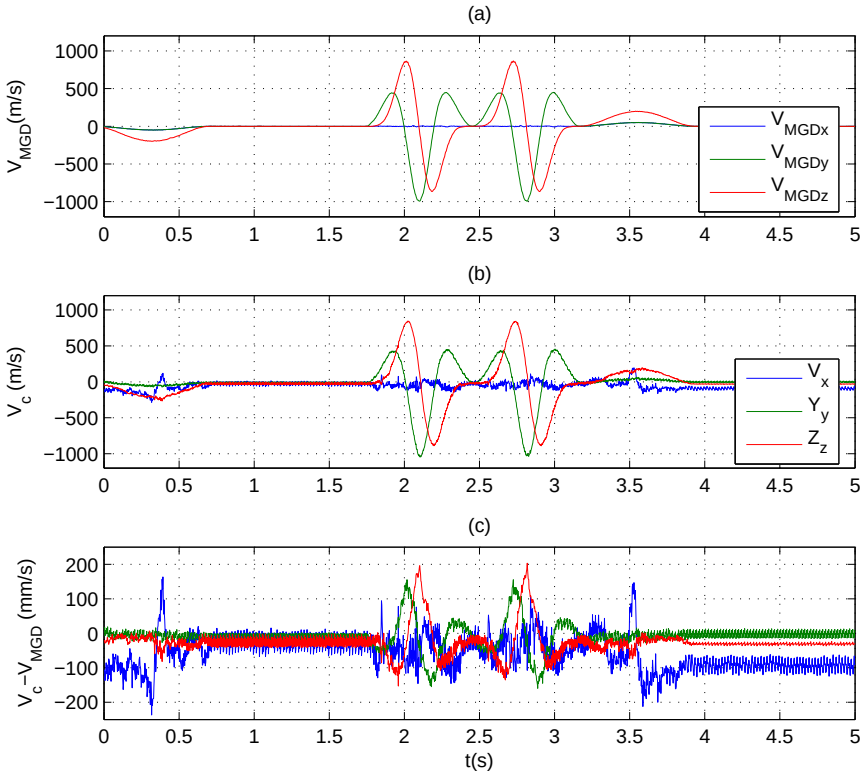


FIGURE 5.7 – Vitesse cartésienne de la plateforme (a) obtenue à partir des mesures articulaires et (b) à partir de l'estimation par vision et (c) l'erreur de position entre les deux estimations.

1m/s durant le temps d'exposition sélectionné ($\leq 80\mu m$), correspond à un déplacement dans l'image d'environ 0.096 pixel (avec la caméra utilisée). Ce déplacement qui peut perturber l'estimation est en réalité plus faible que l'erreur moyenne de reprojection en statique. Par conséquent, les erreurs relatives aux imperfections des modèles de la caméra et de la mire sont plus significatives que les erreurs de détection causées par la variation de la luminosité et du temps d'exposition en dynamique.

La figure 5.7 montre la vitesse de la plate-forme obtenue par dérivation numérique de la trajectoire cartésienne obtenue par le MGD (Fig. 5.7(a)) et la vitesse obtenue par vision (Fig. 5.7b). Tout comme la position, la vitesse V_x qui correspond à l'axe de profondeur de la caméra est la plus bruitée. Notons également qu'il y'a un retard d'environ 4 échantillons [2] sur la vitesse estimée. La cause de ce retard est que le processus d'estimation n'effectue qu'une seule itération ce qui n'est pas suffisant pour converger vers la solution désirée. Ce problème n'apparait pas dans l'estimation de position grâce au terme d'anticipation sur le suivi. Pour réduire ce retard, on peut augmenter le facteur d'amortissement λ en vue d'accélérer la vitesse de convergence. Toutefois, une valeur de gain trop élevée rend

2. Le retard a été calculé par inter-corrélation entre les deux signaux.

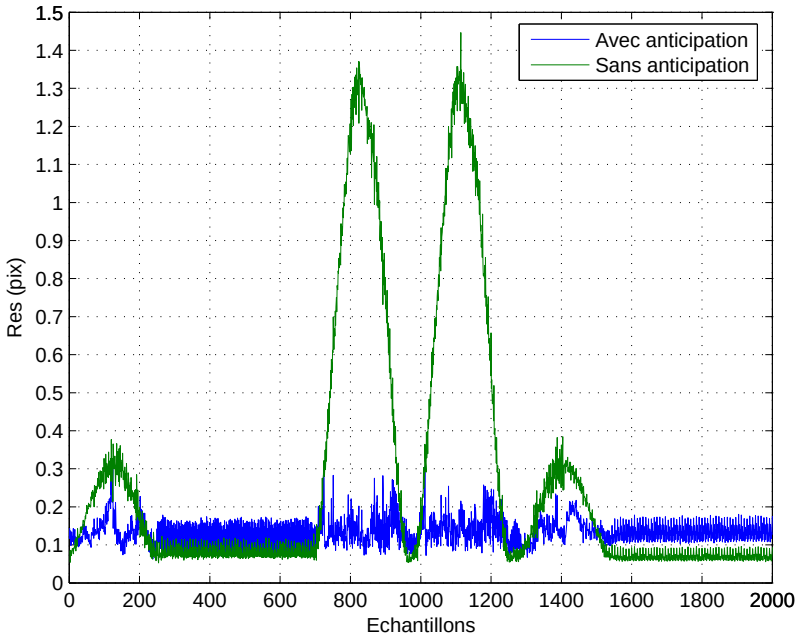


FIGURE 5.8 – Comparaison entre les erreurs de reprojection dans l'estimation de la pose et de la vitesse par asservissement visuel virtuel avec et sans terme d'anticipation.

le système plus sensible aux bruits. Pour avoir une réponse optimale, un compromis doit être fait.

Afin de montrer l'effet de l'anticipation sur le processus de suivi, la même expérience est reproduite sans ce terme. Les résidus normalisés de reprojection dans l'image sont représentés dans la figure 5.8.

En statique, l'erreur de suivi est un peu plus faible sans l'utilisation du terme anticipatif. Cela signifie que l'utilisation de ce terme en statique a un petit effet nocif sur la précision de l'estimation. Ceci s'explique tout simplement par le fait que l'estimation de la vitesse en statique n'est pas parfaitement nulle à cause du bruit (voir figure 5.7). Puisque la vitesse réelle est nulle, l'anticipation représente dans ce cas un bruit supplémentaire dans le processus d'estimation. Il est ainsi préférable du point de vue de la précision de négliger le terme d'anticipation en statique. Toutefois, en dynamique (échantillon $\in [700, 1500]$), nous pouvons clairement voir que le terme d'anticipation améliore considérablement la qualité du suivi. Ceci est confirmé par le tableau 5.3 qui présente une comparaison entre l'erreur de suivi dans les deux cas (anticipation utilisée ou non). Dans ce qui suis, nous utiliserons le terme d'anticipation en statique et en dynamique.

	Statique		Dynamique	
	Sans	Avec	Sans	Avec
Moyenne	0.0738	0.1287	0.8191	0.1424
Variance	2.04e-4	5.8e-4	1.1695	0.0017
Écart type	0.0143	0.0241	0.4117	0.0411
Max	0.14	0.18	1.44	0.31

TABLE 5.3 – Erreurs normalisées de suivi dans l'image (en pixels) avec et sans terme d'anticipation.

5.3.3 Estimation par filtrage de Kalman étendu

Afin d'effectuer une comparaison entre l'estimation par asservissement visuel virtuel et par filtrage de Kalman étendu dans les mêmes conditions (FKE), cette dernière à été testée en utilisant les données acquises précédemment (projections des primitives dans l'image). Cependant, rappelons que cette méthode nécessite une connaissance à priori des statistiques (matrices de covariance) sur les bruits de modèle. La matrice de covariance du bruit de mesure a été calculée à partir des erreurs de reprojection dans l'image obtenue dans l'expérimentation précédente. Aussi, en l'absence d'une mesure de l'accélération, le modèle de mouvement considéré est à vitesse constante. Par conséquent, l'erreur sur le modèle d'évolution n'est autre que l'accélération. La matrice de covariance sur l'erreur de modèle peut dans ce cas être choisie à partir des performances du robot (typiquement : accélération $\max/\sqrt{2}$).

La figure 5.9 représente la trajectoire obtenue en utilisant le MGD du robot et à partir des mesures articulaires, les positions estimées par le filtre de Kalman étendu et l'erreur entre les deux mesures. Nous remarquons tout d'abord que les résultats obtenues par cette méthodes d'estimation sont assez proches des résultats obtenues par l'asservissement visuel virtuel (AVV). Cependant, les erreurs semblent être légèrement supérieurs par rapport à cette dernière. Par ailleurs, les pics d'erreurs observés dans la figure 5.6 vers t = 0.4s et t = 3.5s n'apparaissent pas dans la figure 5.9. Ceci peut être dû soit à l'estimation par AVV soit au filtre de Kalman qui a pour effet de filtrer les mouvements brusques (qui pourraient être causés par les jeux dans les articulations du robot).

Contrairement à l'estimation de la position, la vitesse obtenue par le filtre de Kalman étendu (Fig. 5.10) est plus faible et moins bruitée que celle obtenue par l'asservissement visuel virtuel. En effet, on peut voir sur la figure 5.10(c) qu'il n'existe pratiquement pas de biais sur l'estimation de la vitesse et les pics d'erreurs sont moins importants que dans la méthode précédente. Cependant, le retard sur la vitesse estimée, qui est de 5 échantillons, est supérieur à celui de la méthode précédente.

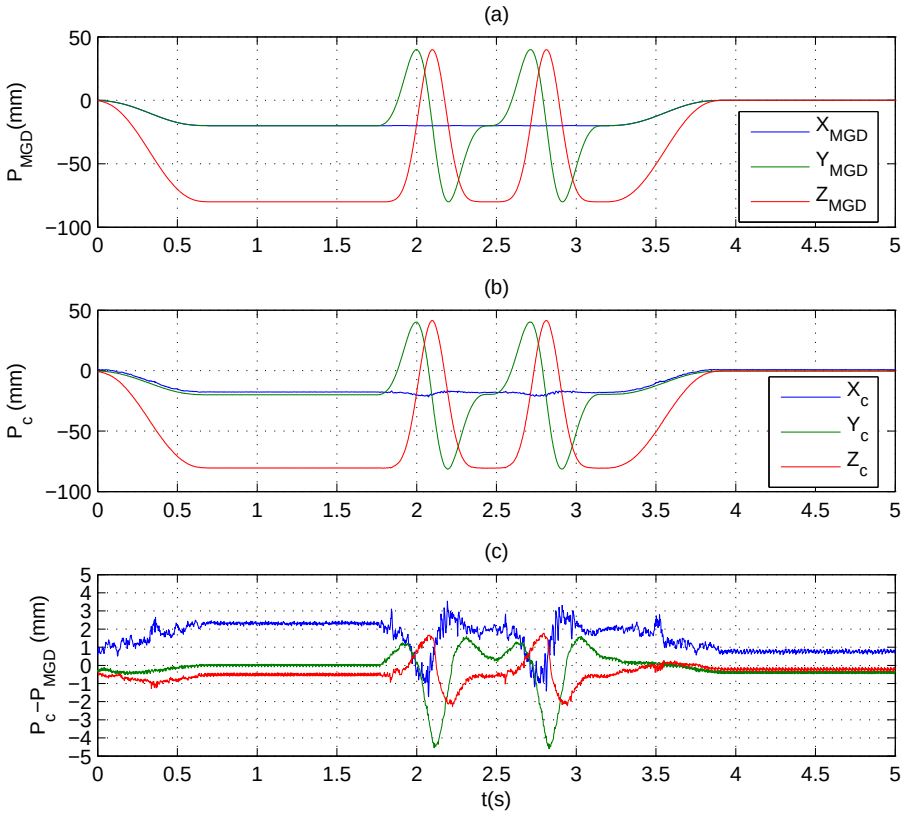


FIGURE 5.9 – Position de la plate-forme obtenue par (a) le modèle géométrique direct du robot et par (b) le filtre de Kalman étendu ainsi que (c) l'erreur entre les deux mesures.

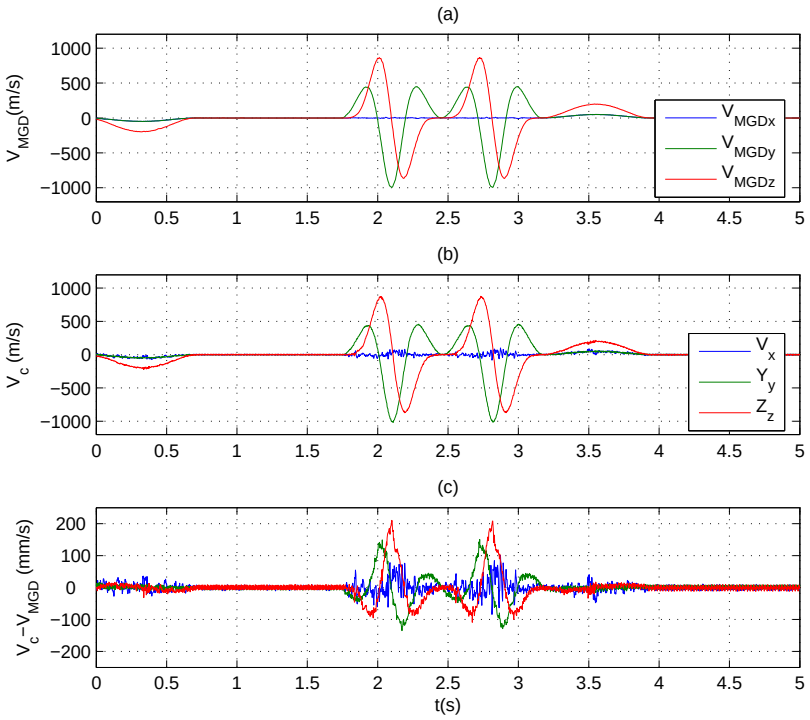


FIGURE 5.10 – Vitesse de la plate-forme obtenue par (a) le modèle géométrique direct du robot et par (b) le filtre de Kalman étendu ainsi que (c) l'erreur entre les deux mesures.

5.4 Asservissement visuel 3D dynamique

L'implémentation de la commande dynamique cartésienne par vision consiste tout d'abord à exprimer le modèle dynamique du robot en fonction de la position et de la vitesse de la plate-forme [PLA+08, Pac09]. L'estimation de l'état de la plate-forme pour la compensation de la dynamique et la régulation de la commande sont ensuite obtenues par vision. Les résultats d'estimation précédents montrent que les deux méthodes proposées sont pratiquement équivalentes en terme de précision. Cependant, hormis la précision, il est important en commande d'obtenir une mesure stable, robuste et avec le moins de retard possible. De ce point de vue, l'asservissement visuel virtuel semble être la plus appropriée car il est plus stable que le filtre de Kalman étendu (moins de risque de divergence). De plus, la mesure obtenue par l'AVV et moins sensible aux erreurs de modèle et comporte moins de retard. Par conséquent, c'est cette méthode d'estimation qui a été choisie pour la commande.

En réalité, l'utilisation directe de l'estimation de la pose et de la vitesse par la méthode d'asservissement visuel virtuel ne permet pas d'obtenir une commande stable. La cause principale d'instabilité est due au retard sur l'estimation de la vitesse à cause du nombre insuffisant d'itérations (voir §4.3.1.3). La solution qui a été adoptée pour remédier à ce problème consiste à utiliser le vecteur de commande dans l'estimation (voir Fig. 4.6).

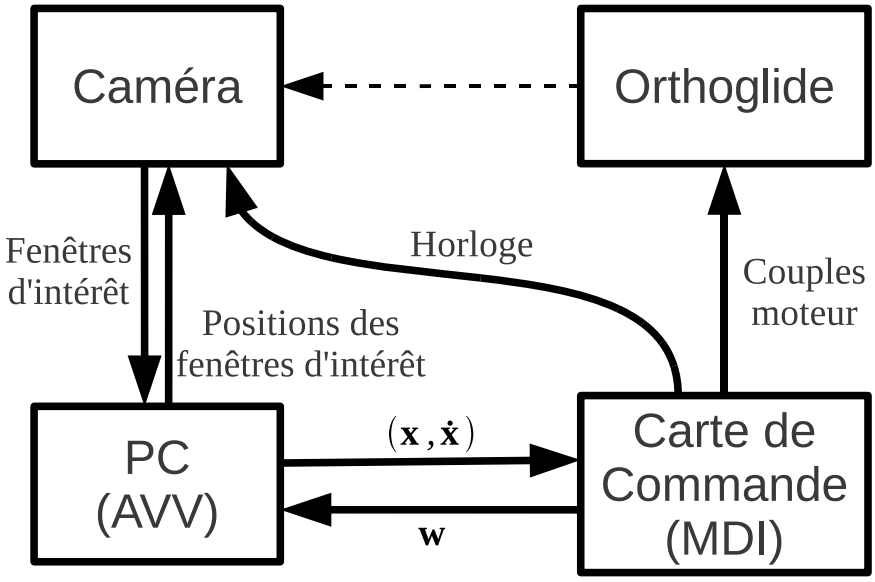


FIGURE 5.11 – Schéma synoptique de l'architecture du système.

Cependant, la mise en œuvre de cette commande nécessite de modifier légèrement l'architecture du système car la communication entre l'unité de traitement du système de vision et la carte de commande du robot (carte dSpace) doit pouvoir s'effectuer dans les deux sens. Ainsi, le vecteur de commande doit pouvoir être envoyé de la carte de commande du robot vers le processus d'estimation implémenté sur le PC qui effectue l'estimation de la pose et de la vitesse de la plate-forme. Cette estimation est ensuite renvoyée vers la carte de commande du robot qui utilise la mesure par vision dans la commande cartésienne aussi bien pour le calcul de la dynamique que pour la régulation du système. Pour ce faire, la communication s'effectue à travers la liaison RS-422 configurée à 333333 Bauds dans les deux sens. Les données envoyées de la carte de commande vers le PC sont trois nombres à virgule flottante (3×4 octets) qui représentent les accélérations de commande. Dans l'autre sens, les données envoyées du PC vers la carte de commande sont 6 nombres à virgule flottante qui représentent la position et la vitesse de la plate-forme. Le schéma synoptique du système de commande est représenté sur la figure 5.11.

Les résultats obtenus montrent que l'utilisation de cette approche permet non seulement de stabiliser la commande mais aussi de commander le robot au maximum de ses capacités. Cependant, les défauts mécaniques entre la plate-forme et les actionneurs, que la commande tente de compenser, induisent des oscillations plus ou moins fortes. Afin de réduire l'effet de ce phénomène, les gains de la commande ont été réduits de façon à ce que la fréquence propre de la commande en boucle fermée ne corresponde pas à la plage de fréquence qui excite les modes propres de la structure mécanique.

La figure 5.12 montre la trajectoire de référence, la trajectoire obtenue par la commande dynamique basée vision et la trajectoire correspondant à la commande articulaire.

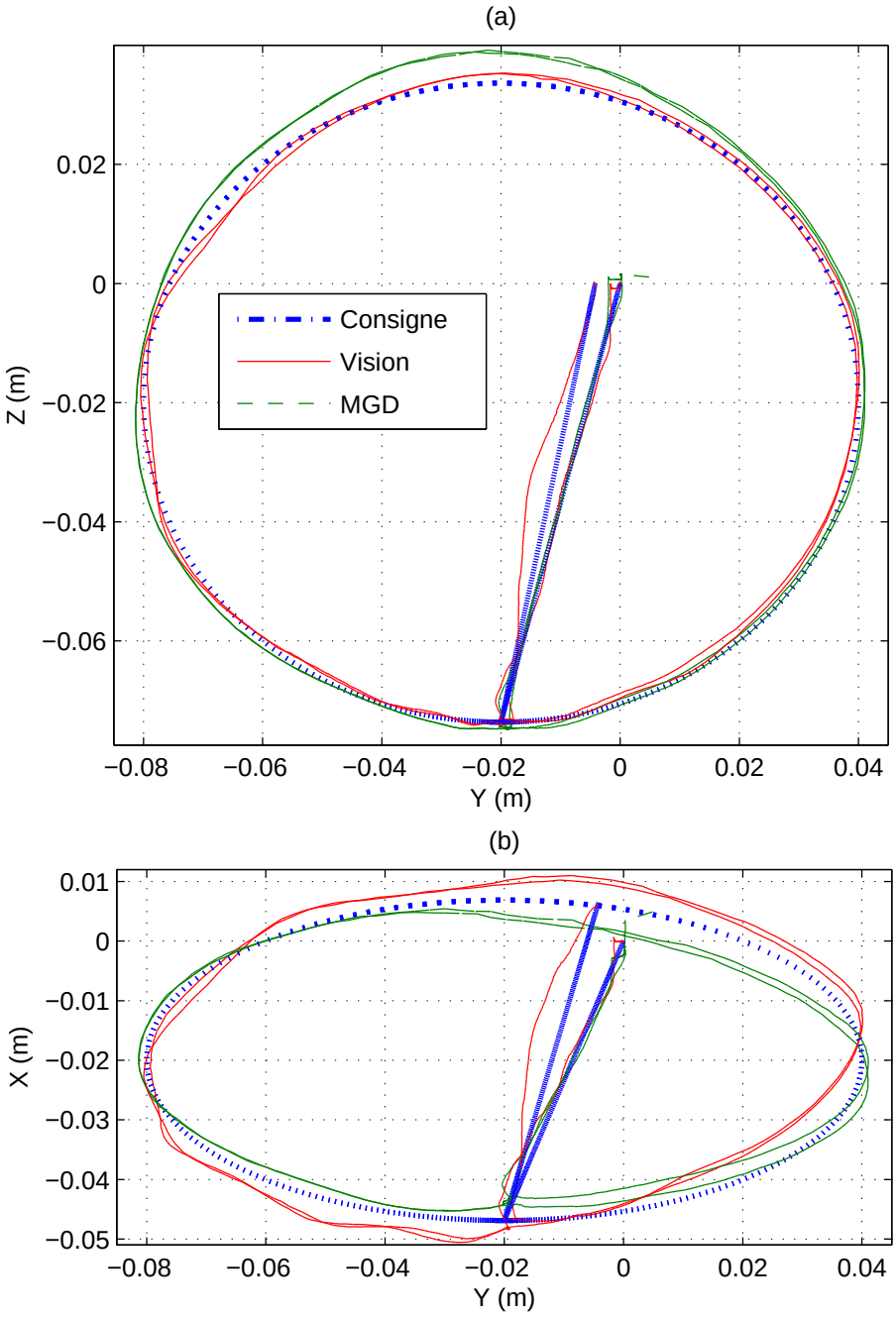


FIGURE 5.12 – Comparaison entre les trajectoires obtenues par la commande référencée vision et par la commande référencée modèle projetées sur les plan (a) YZ et (b) XY.

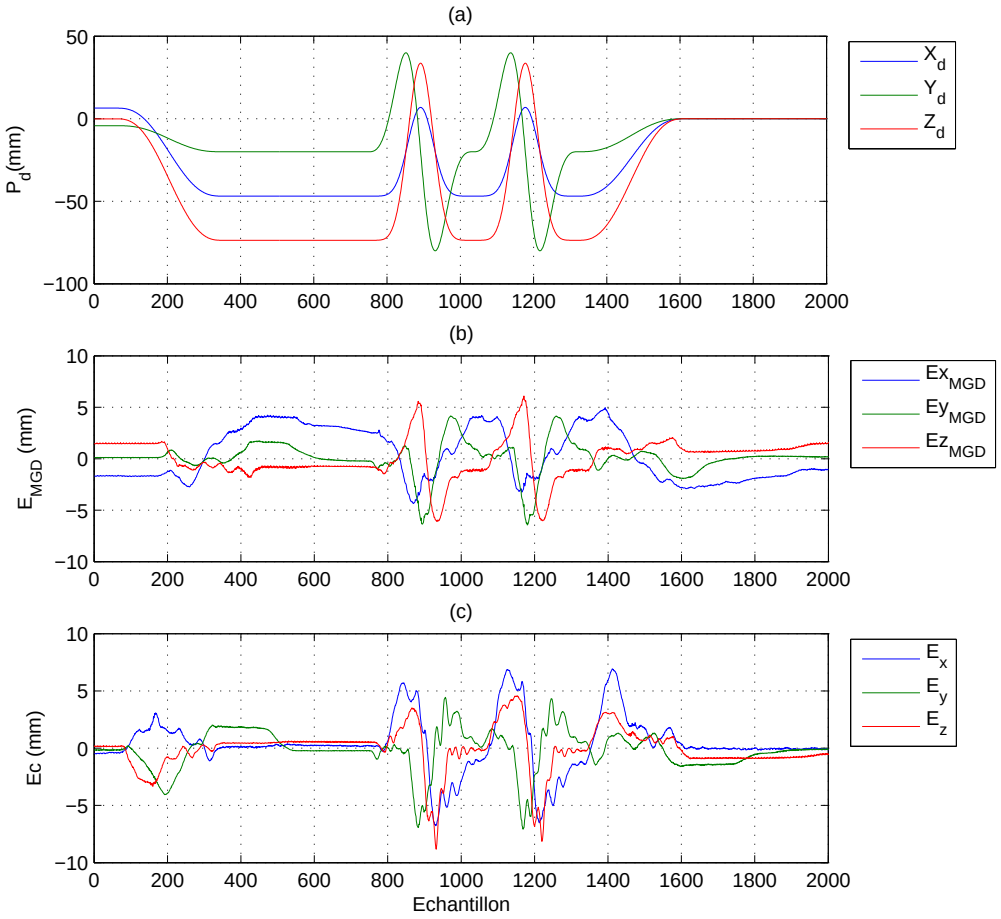


FIGURE 5.13 – (a) Trajectoire de consigne et les erreurs correspondant à (b) la commande référencée modèle et à (c) la commande référencée vision en fonction du temps.

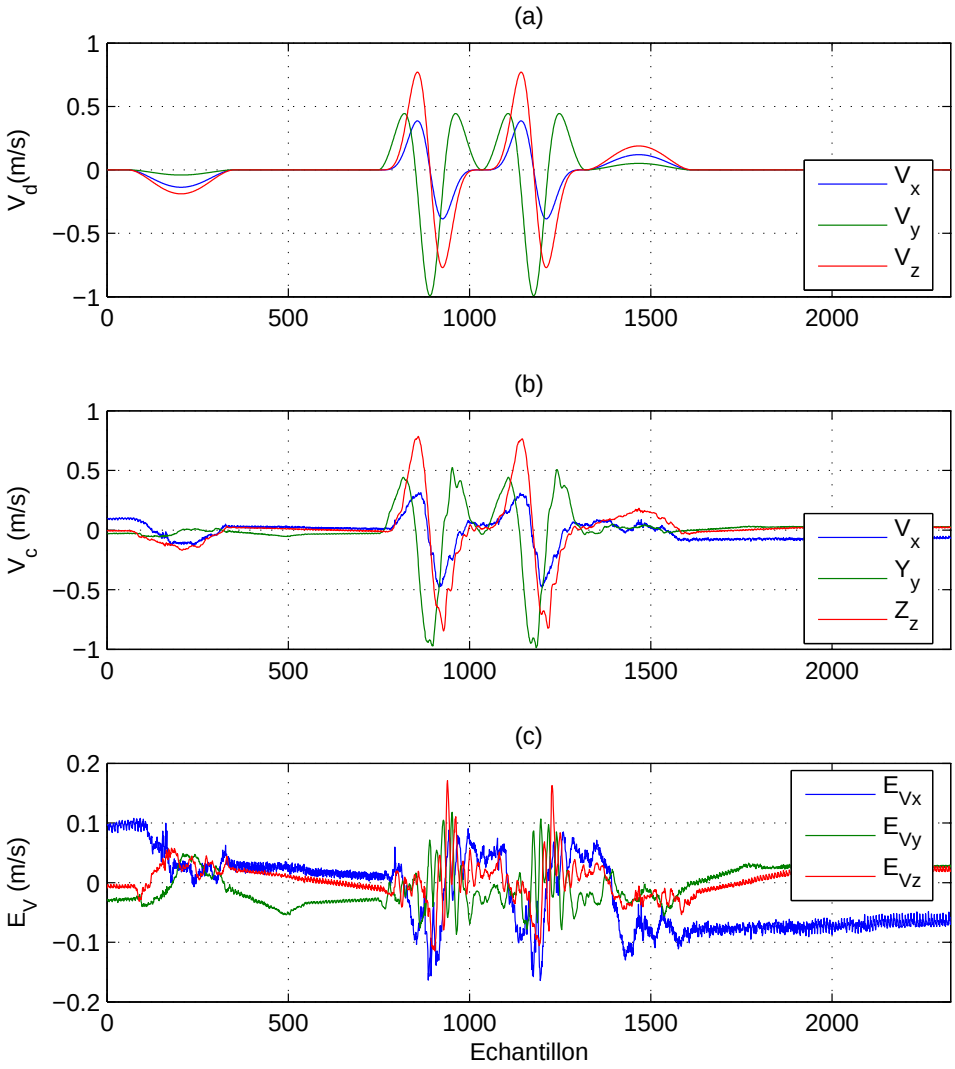


FIGURE 5.14 – (a) Vitesse de consigne et les erreurs correspondant (b) à la commande référencée modèle et (c) à la commande référencée vision.

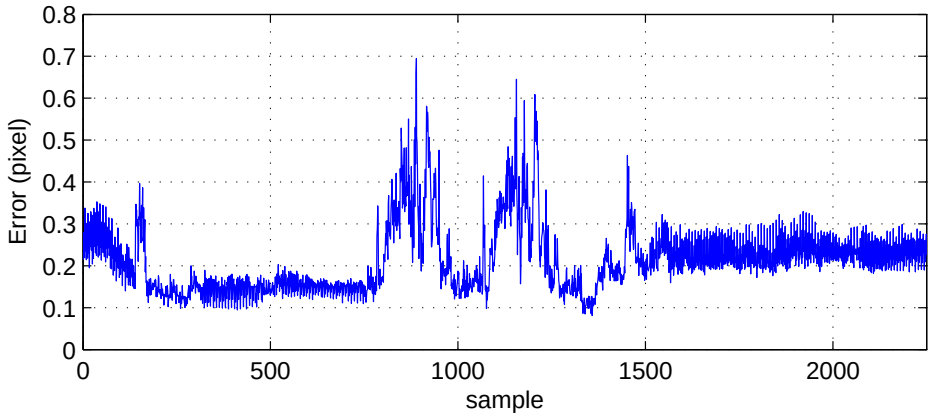


FIGURE 5.15 – Erreurs de suivi dans l'image en fonction du temps.

Ces deux dernières trajectoires ont été mesurées par vision afin de pouvoir comparer les trajectoires à partir de la même mesure qui semble la plus précise. Tout d'abord, nous pouvons remarquer que le cercle obtenu par la commande référencée modèle est plus large que la trajectoire de référence ce qui n'est pas le cas de la trajectoire obtenue par la commande basée vision. En effet, les rayons moyens de la commande basée modèle et de la commande par vision sont respectivement de 61.34mm et 60.20mm contre 60.0mm pour la consigne. De plus, la distance algébrique moyenne entre les trajectoires obtenues et la trajectoire de référence (distance normale) est plus faible dans la commande basée vision (2.74mm) que pour la trajectoire basée modèle (3.25mm). Ceci peut s'expliquer par le fait que la commande basée modèle ne détecte pas et, donc, ne peut pas compenser les erreurs causées par les imperfections mécaniques, contrairement à la commande par vision qui est asservie sur la pose de l'effecteur et, donc, permet de compenser plus ou moins ces erreurs. Cependant, les défauts mécaniques n'ont pas été prises en compte dans la synthèse de la loi de commande, chose qui devrait l'être si l'on veut exploiter au mieux la commande par vision. Toutefois, ce travail (prise en compte des flexibilités et des jeux dans la commande par vision) sort du contexte des travaux présentés dans le présent document.

La figure 5.13 représente respectivement la position désirée et les positions obtenues par la commande basée modèle et la commande par vision. On peut remarquer en premier lieu que les erreurs correspondant à la trajectoire obtenue par vision ne dépassent pas les erreurs de jeux dans les articulations. De plus, la comparaison entre les erreurs obtenues par les deux commandes montre que la commande par vision est la plus précise aussi bien en statique qu'en dynamique (voir tableau 5.4).

	Commande réf. modèle			Commande fér. vision		
	x	y	z	x	y	z
Moyenne (mm)	0.19	0.1	0.28	0.11	-0.08	-0.06
Écart type (mm)	2.42	1.47	1.76	2.21	1.6	1.64

TABLE 5.4 – Moyenne et écart type des erreurs sur les trajectoires obtenues par la commande référencée modèle et par l'asservissement visuel dynamique 3D.

La figure 5.15 montre les erreurs de suivi dans l'image. Notons que même à cette vitesse, l'erreur de suivi reste inférieure au pixel. Il apparait aussi que l'erreur dans l'image augmente avec l'accélération. Ceci peut être causé par des erreurs sur le modèle dynamique qui induisent des erreurs proportionnelles à l'accélération et qui se répercutent sur le modèle de projection. Néanmoins, les erreurs de reprojection dans l'image restent bien plus faibles que les déplacements des primitives dans l'image en raison des jeux et des flexibilités (Fig. 5.3).

5.5 Asservissement visuel 2D dynamique

Cette expérimentation consiste à implémenter le schéma d'asservissement visuel dynamique avec estimation de pose (Fig. 4.8) sur l'Orthoglide. L'architecture du système est la même que pour la commande 3D sauf que l'accélération de commande est calculée par le PC à partir des trajectoires de références dans l'image et des mesures acquises par vision. De plus, l'état du robot est reconstruit afin de compenser la dynamique du robot. Cette dernière est calculée dans la carte de commande du robot. Il faut aussi noter que pour cette commande, les données envoyées du PC vers la commande sont le vecteur de commande ainsi que les estimations de la pose et de la vitesse nécessaires à la compensation de la dynamique.

L'une des difficultés dans l'implémentation de l'asservissement visuel 2D en couples calculés est la génération de la trajectoire de référence dans l'image. En réalité il existe des techniques de synthèse robuste de trajectoires dans l'image ([MC02c]). Cependant, pour une première validation de la commande, la technique d'apprentissage a été préférée afin d'éliminer les éventuelles erreurs ou imprécisions sur la trajectoire synthétisée qui pourraient nous induire en erreur dans l'analyse des performances de la commande. Ainsi, la trajectoire désirée dans l'image a été obtenue par apprentissage. Les positions des primitives visuelles ont donc été enregistrées lors de l'exécution de la trajectoire cartésienne obtenue par la commande 3D dynamique. Il faut cependant noter que, comme la trajectoire de la plate-forme peut différer de la trajectoire de consigne (comme cela a été observé dans les expérimentations précédentes), c'est évidemment la trajectoire réalisée qui représentera la trajectoire de consigne pour les expérimentations suivantes. L'objectif de l'expérimentation étant de pouvoir exécuter la trajectoire apprise exclusivement à partir des données images en utilisant l'asservissement visuel 2D dynamique.

Pour des considérations techniques [3] la trajectoire de référence choisie n'est constituée que d'une phase correspondant à la trajectoire circulaire présentée dans les expérimentations précédentes. Afin d'améliorer les performances de la commande, un filtre moyenneur a été employé pour filtrer le bruit sur l'erreur de vitesse. Par ailleurs, les gains de la commande ont été choisis afin d'assurer une réponse critique du 3e ordre avec un pôle triple. Ainsi, pour une pulsation propre de 25rad/s on obtient les trajectoires représentées sur la figure 5.17 montrant la trajectoire de référence et la trajectoire mesurée dans l'image. On peut ainsi constater que la trajectoire obtenue est lisse et correctement superposée à la trajectoire de référence. L'erreur dans l'image est illustrée dans la même figure où l'on peut voir que l'erreur maximale commise sur la trajectoire est inférieure à l'erreur dans l'image due aux jeux et flexibilités sur les articulations. En théorie, ces erreurs peuvent

3. Le passage d'une phase de la trajectoire à une autre se fait manuellement à partir de l'interface de commande du robot. Ceci pose le problème de la répétabilité par rapport à la trajectoire apprise en raison du délai introduit par l'opérateur lors du passage entre deux phases.

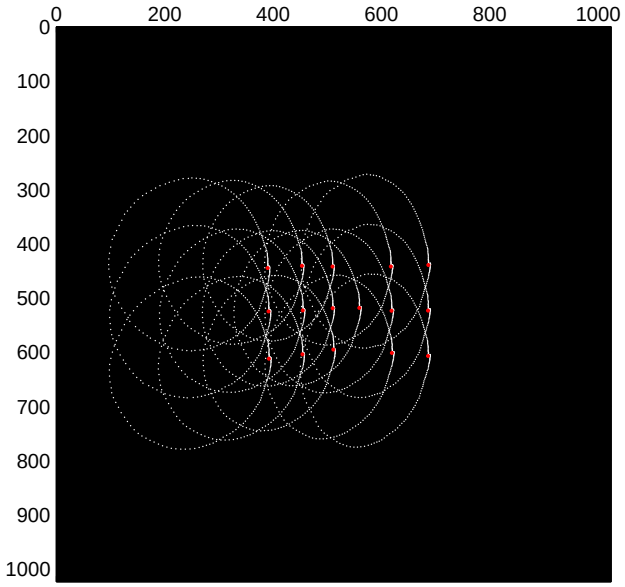


FIGURE 5.16 – Positions des 16 primitives dans l'image dans la trajectoire acquises (points blancs) et leurs positions initiale (points rouges).

être causées soit par une erreur de positionnement du robot soit par des erreurs de suivi des primitives dans l'image. Cependant, l'erreur de reprojection dans l'estimation de la pose et de la vitesse ne dépasse pas 0.5517 pixel, ce qui signifie que le suivi des primitives dans l'image est bien assuré. De plus, nous pouvons constater que l'erreur dans l'image des différents points de la cible varie de façon homogène, ce qui laisse à penser que les points détectés gardent plus ou moins la même distance dans l'image. Tout ceci montre que cette erreur est due à une erreur de positionnement dans l'espace et non par une erreur de suivi dans l'image. Le tableau 5.5 montre la moyenne et l'écart type sur les erreurs dans l'image entre la trajectoire de consigne et la trajectoire obtenue par l'asservissement visuel 2D dynamique [4].

	u	v
Moyenne	-0.7345	-0.5096
Eccart type	1.6378	1.7070

TABLE 5.5 – Erreurs sur le suivi de trajectoire dans l'image.

La figure 5.18 montre les trajectoires cartésiennes désirée et estimées par vision. On peut voir dans cette figure que la trajectoire cartésienne obtenue est lisse sur les trois axes. A partir de cette figure nous pouvons remarquer que l'angle de vue à partir duquel

4. Notons que cette erreur dans l'image, qui représente l'écart entre la trajectoire réelle et désirée du robot, ne doit pas être confondue avec les erreurs de reprojection dans les processus d'estimation de la pose et de la vitesse qui est indépendante de la commande.

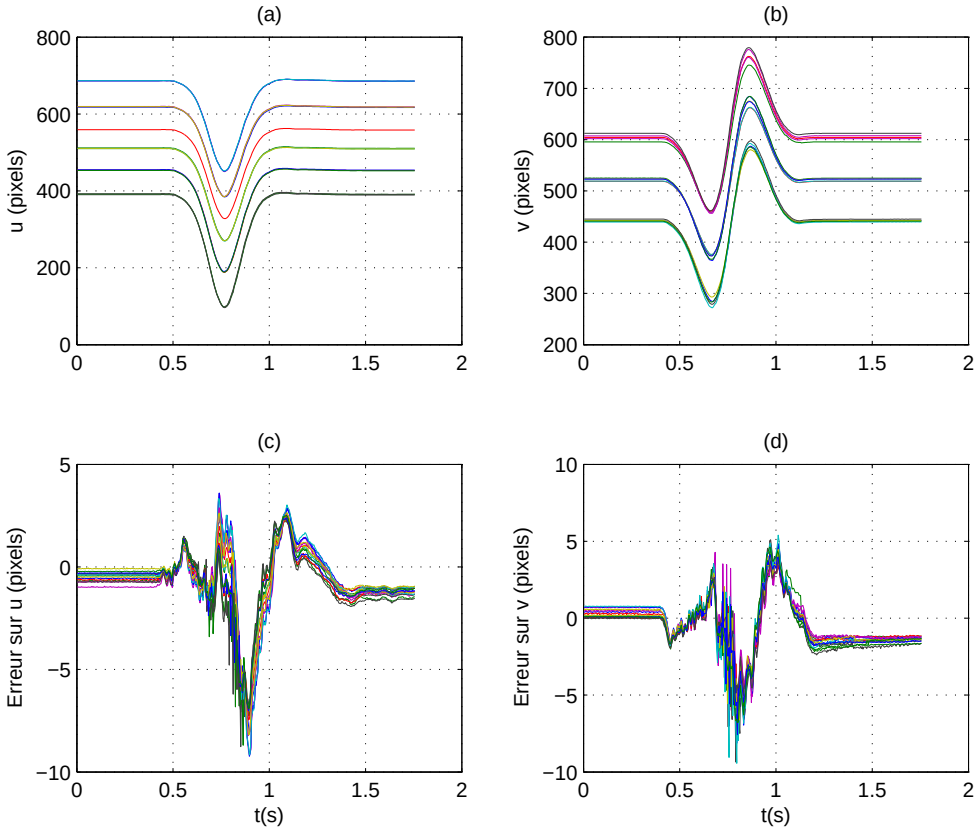


FIGURE 5.17 – Positions des primitives dans l'image pour pulsation propre de 25 rad/s selon (a) l'axe u et (b) l'axe v ainsi que les erreurs entre les positions réelle et désirée sur (c) u et (d) v.

la trajectoire parait être la plus proche de la trajectoire désirée est celui de la figure 5.18(a). Ceci s'explique très bien puisqu'il s'agit de la trajectoire vue par la caméra. On peut aussi constater sur cette même figure que le rayon de la trajectoire exécutée n'est pas plus large que la trajectoire désirée contrairement à ce qu'on a pu voir dans la commande référencée modèle.

L'évolution de la position de la plate-forme mesurée par vision ainsi que l'erreur de positionnement dans l'espace sont représentés sur la figure 5.19. Ce que l'on peut noter à partir de la courbe d'erreur (Fig. 5.19) est que l'erreur de positionnement est plus faible que pour la commande précédente. Le même constat peut être fait par rapport aux erreurs de vitesse (Fig. 5.20). Les statistiques sur les erreurs de position et de vitesse figurant sur le tableau 5.6, notamment les écarts types sur les trois axes, confirment ce constat. Ceci s'explique tout simplement par la robustesse de la commande 2D. En effet, contrairement à l'estimation où à l'asservissement visuel 3D dynamique, l'erreur dans l'image est calculée à partir de mesures acquises par la caméra et non estimée par un modèle de projection, ce qui rend la commande plus robuste face aux erreurs de modèle de la mire et de la caméra. Par ailleurs, cette commande n'ai pas affectée par le retard sur le signal de régulation comme c'est le cas pour la commande précédente.

	x	y	z
Moyenne	-0.4742	-0.7917	0.1786
Eccart type	1.4031	1.5946	1.5842

TABLE 5.6 – Erreur sur les positions cartésiennes.

Afin d'identifier ce qui limite les performances de la commande en termes de stabilité et de précision, les gains de la commande ont été augmentés jusqu'à l'apparition d'oscillations (gains initiaux multipliés par deux ce qui correspond à une pulsation propre de 50rad/s). En effet, la figure 5.21(a) montre les trajectoires cartésiennes obtenues par vision et par le MGD avec la même commande dynamique 2D. On peut voir dans l'agrandissement de cette trajectoire (Fig. 5.21(b)) que la trajectoire obtenue par vision révèle des oscillations qui apparaissent aussi dans l'image (Fig. 5.22) où les oscillations sont du même ordre de grandeur que les erreurs de suivi. Cependant, ces oscillations sont beaucoup plus faibles dans la trajectoire obtenue par le MGD. Ceci signifie que les faibles oscillations au niveau des articulations se traduisent par des déplacements plus importants au niveau de la plate-forme. L'explication la plus plausible est que les faibles oscillations au niveau des articulations sont amplifiées par les flexibilités et les jeux qui existent entre les actionneurs et la plate-forme.

Par ailleurs, les imperfections dans la mécanique introduisent des déphasages dans le mouvement entre les actionneurs et la plate-forme qui sont d'autant plus importants que la fréquence propre de la commande (proportionnelle aux gains) est élevée, ce qui peut faire basculer la commande dans un mode oscillatoire. Par conséquent, il est fort probable

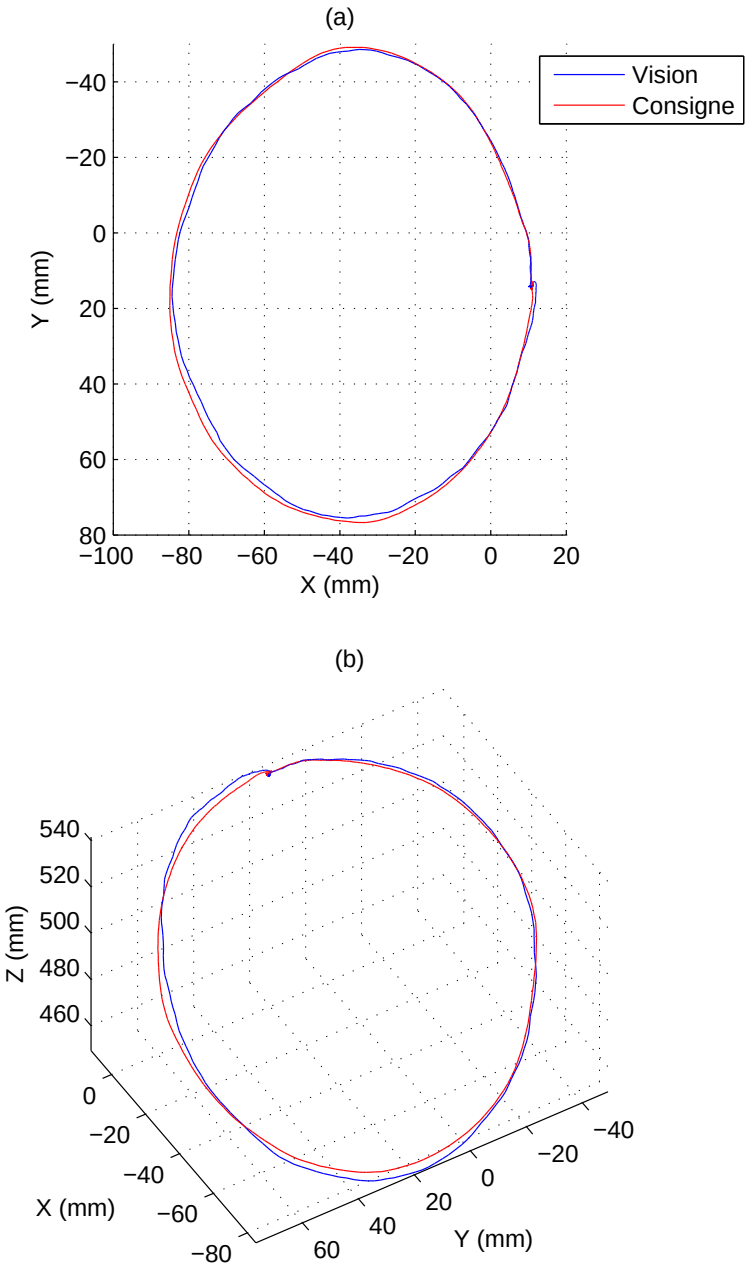


FIGURE 5.18 – Trajectoire cartésienne de consigne et celle obtenue par l'asservissement visuel 2D dynamique (a) vues par la caméra et (b) représentée dans l'espace.

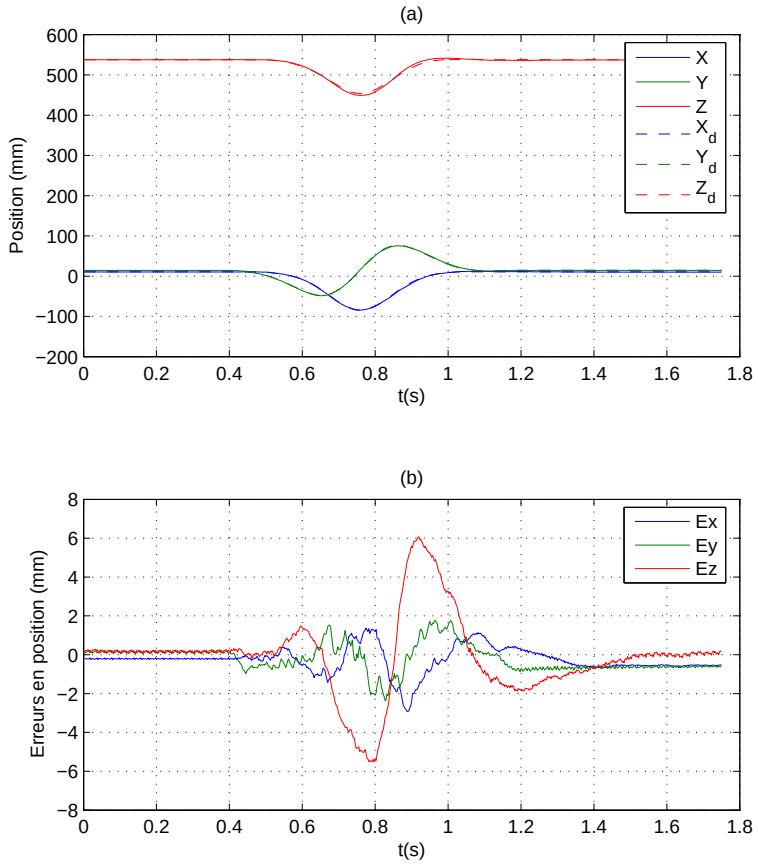


FIGURE 5.19 – (a) Positions cartésiennes de consigne et celles obtenue par l'asservissement visuel dynamique 2D et (b) l'erreur entre les deux.

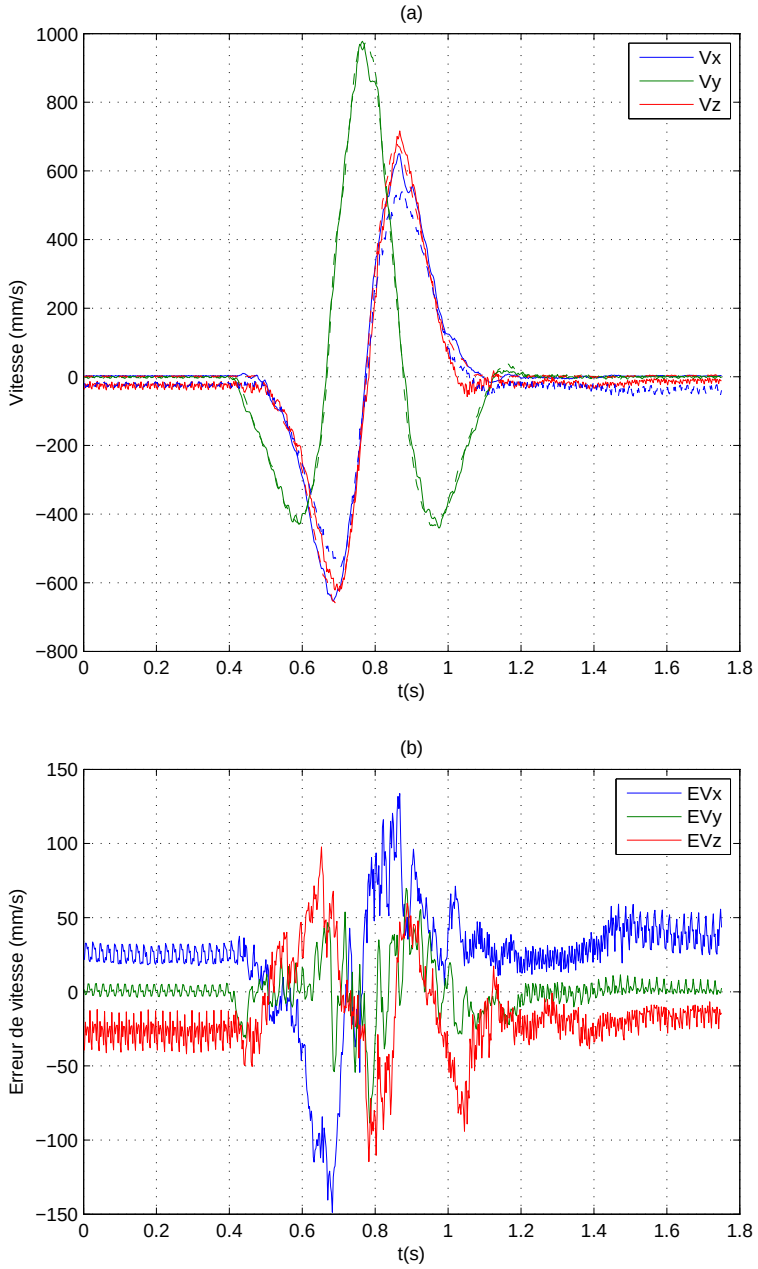


FIGURE 5.20 – (a) Positions cartésiennes de consigne et celles obtenue par l'asservissement visuel dynamique 2D et (b) l'erreur entre les deux.

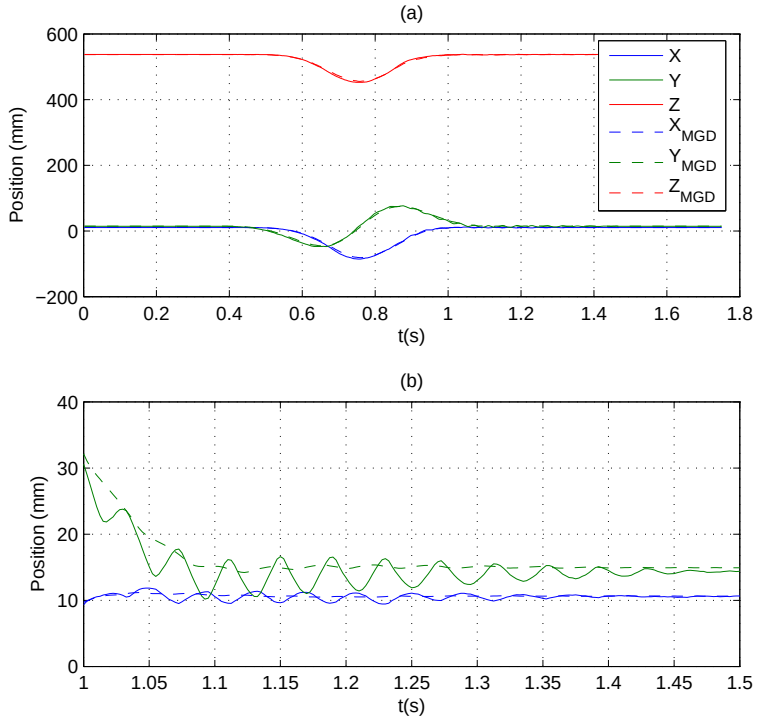


FIGURE 5.21 – Comparaison entre les positions cartésiennes. (a) vue normale et (b) vue agrandie.

que la limitation des gains de la commande, qui se répercute directement sur la précision de cette dernière, soit causée par les imperfections de la mécanique du robot et non pas par la commande référencée vision.

5.6 Conclusion

Nous avons présenté dans ce chapitre les différents résultats expérimentaux obtenus en estimation de pose et de vitesse (par asservissement visuel virtuel) ainsi qu'en asservissement visuel dynamique dans l'espace et dans l'image à partir d'une acquisition séquentielle rapide de fenêtres d'intérêt contenant les primitives visuelles. Concernant l'estimation, nous avons vu que le suivi des points dans l'image est précis (< 1 pixel) même au maximum des performances du robot. La précision de l'estimation de l'état de la plateforme n'a pas pu être déterminée de façon exacte en raison de l'absence d'un moyen de mesure alternatif fiable. Néanmoins, les faibles erreurs de reprojections des points dans l'image calculées à partir des estimations de pose et de vitesse indiquent que ces erreurs sont tout aussi faibles. Ceci nous a permis d'utiliser la mesure par vision comme mesure de référence.

L'utilisation de la mesure par vision de la plate-forme du robot dans la commande

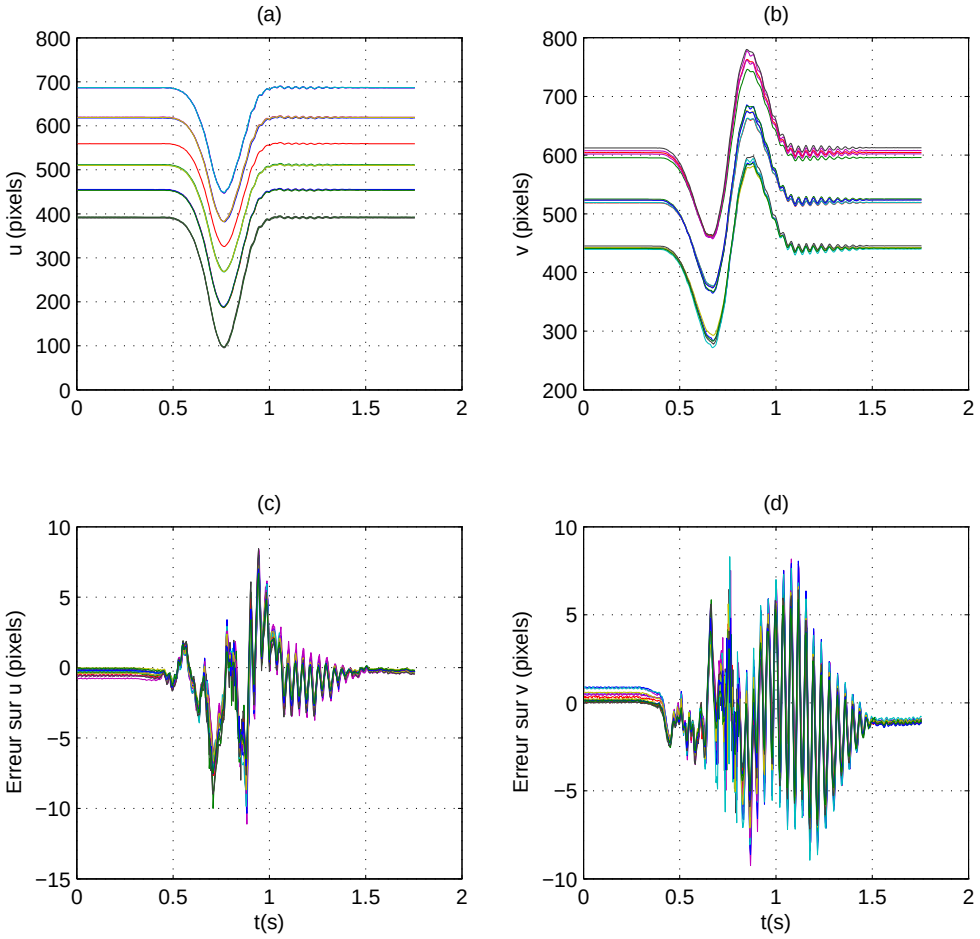


FIGURE 5.22 – Positions des primitives dans l'image pour pulsation propre de 50 rad/s selon (a) l'axe u et (b) l'axe v ainsi que les erreurs entre les positions réelle et désirée sur (c) u et (d) v.

dynamique a posé le problème des jeux et flexibilités existants dans la structure du robot ainsi que le retard dans l'estimation de la vitesse du fait du nombre insuffisant d'itérations. Ainsi, la combinaison de ces deux effets perturbaient considérablement la commande au point de la rendre instable. Notons tout de même que le retard lié au nombre d'itérations insuffisant ne représente pas un problème réel si les algorithmes d'estimation peuvent être implémentés sur une unité de traitement plus puissante (ordinateur plus puissant, DSP, etc.). Ne pouvant pas augmenter le nombre d'itérations (par rapport au matériel disponible) ni réduire les jeux dans la structure mécanique du robot, la solution à ce problème a été de réduire le retard sur l'estimation de la vitesse en utilisant l'accélération de commande comme suggéré en §4.4. Ainsi, cette approche a permis de stabiliser la commande et d'obtenir un meilleur suivi de trajectoire par rapport à la commande référencée modèle.

Le problème du retard dans l'estimation ne se pose pas en commande dynamique référencée image puisque le signal de régulation (points dans l'image) est obtenue par une caméra rapide (non estimé). Cette commande a permis d'améliorer la précision de la commande. Cependant, même si les performances de la commande restent limitées, des expérimentations montrent que les erreurs sont plus liées aux défauts de rigidité du robot qu'à la commande elle-même. On peut donc s'attendre à obtenir de meilleurs résultats sur des robots plus rigides.

Conclusion générale et perspectives

Améliorer les performances de la commande par vision est l'un des enjeux majeurs pour que cette solution technologique s'impose sur le plan industriel [KV09]. A cette fin, différentes contributions ont été présentées. Tout d'abord, nous nous sommes intéressés à la vitesse d'acquisition des systèmes de vision. En effet, la fréquence d'acquisition des systèmes classiques est trop faible pour espérer obtenir de bonnes performances dynamiques des systèmes commandés. Pour remédier à ce problème, nous avons proposé d'utiliser un mode d'acquisition par fenêtres d'intérêt. Ce mode d'acquisition particulier utilise la bande passante de l'interface de communication de manière plus efficace. Ceci permet d'augmenter la fréquence d'acquisition du système de vision et de satisfaire ainsi les exigences des commandes hautes performances. En outre, ce mode d'acquisition, qui consiste à ne récupérer que les régions d'intérêt dans la scène, n'altère pas la qualité du signal image puisque la résolution des fenêtres acquises n'est nullement affectée. De plus, le champ de vue global n'est pas réduit car les fenêtres peuvent être positionnées dans toute l'image. En réalité, il est même possible d'améliorer la qualité du signal acquis en utilisant des capteurs avec une résolution bien plus importante sans que cela n'ait de conséquence significative sur la fréquence d'acquisition. Par ailleurs, il est important de souligner que les méthodes d'estimation et de commande proposées ne sont pas limitées à l'acquisition séquentielle de fenêtres d'intérêt mais à tout mode d'acquisition qui permet d'acquérir des primitives à des instants différés. Il peut même s'agir d'une acquisition classique où les primitives seraient définies sur deux images différentes (des images successives par exemple).

Cependant, les primitives visuelles n'étant pas acquises au même instant, l'utilisation des méthodes de reconstruction euclidienne et d'asservissements visuels classiques ne peuvent pas être directement utilisées. En revanche, ces primitives "spatio-temporelle" ont l'avantage de contenir des informations sur l'état dynamique, mais aussi sur l'évolution de la scène observée. L'exploitation de cette propriété nous a permis de développer des méthodes d'estimation de pose et de vitesse indispensables à la synthèse d'une commande performante intégrant une compensation de la dynamique du système commandé. Ces méthodes ne font pas intervenir de dérivation numérique, ce qui permet d'améliorer la précision de l'estimation de vitesse par rapport à une estimation obtenue par dérivation numérique de la pose. Aussi, des schémas d'asservissement visuel cinématique basés sur

ces primitives ont été proposés pour la commande haute vitesse des robots légers. Par ailleurs, nous avons vu que les méthodes d'asservissement pouvaient être utilisées pour l'estimation de la pose et de la vitesse en utilisant le concept d'asservissement visuel virtuel. En adoptant une architecture logicielle multi-tâches, l'implémentation des méthodes d'acquisition et d'estimation sur un PC standard a permis d'atteindre une fréquence d'acquisition de 4 kHz et une fréquence d'estimation de 400 Hz. Les résultats obtenus montrent que l'estimation est précise même pour des mouvements à fortes accélérations.

La mesure de pose et de vitesse par vision a été exploitée pour l'estimation de la jacobienne ou de la matrice d'interaction nécessaires à l'implémentation de l'algorithme d'estimation lui même. De plus, nous avons vu que cette mesure 3D pouvait aussi être utilisée pour le calcul de la dynamique et même comme signal de régulation dans une commande en couple (asservissement visuel 3D dynamique). L'implémentation de la méthode d'estimation dans une application de commande nécessite cependant l'utilisation du vecteur de commande afin d'éliminer le retard dans le suivi de la vitesse. Les résultats obtenus sur l'Orthoglide ont montré que la commande par vision peut être plus précise que la commande articulaire.

L'asservissement visuel 2D dynamique utilise aussi l'aspect spatio-temporelle des primitives pour la régulation dans l'image de la pose et de la vitesse du robot. Comme la régulation est directement calculée à partir de la mesure dans l'image, le bruit et le retard susceptibles d'affecter, l'estimation de la pose et de la vitesse ont une incidence plus faible sur la commande. En effet, ces perturbations affectent la matrice d'interaction et la compensation de la dynamique mais pas le signal de régulation. La commande est donc plus robuste vis-à-vis d'erreurs sur les modèles de la caméra et de l'objet observé ainsi que par rapport au retard sur l'estimation.

Les résultats obtenus par la commande dynamique basée image sur une trajectoire apprise sont plus précis que ceux obtenus par l'asservissement visuel 3D dynamique qui sont à leur plus précis que les résultats de la commande basée modèle. Par ailleurs, les gains ont pu être augmentés pour la commande basée image, ce qui signifie que cette commande est plus stable et robuste que la commande dynamique 3D. Enfin, les différents résultats suggèrent que les erreurs sur les trajectoires exécutées en utilisant la commande basée vision sont principalement dues à la structure mécanique du robot qui comporte des imperfections au niveau de la structure mécanique (jeux et flexibilités).

Les perspectives de ces travaux se situent à plusieurs niveaux : la mesure, l'estimation, la commande et l'expérimentation. Concernant ce dernier point, d'autres expérimentations sont à prévoir afin d'illustrer de façon plus exhaustive les performances et les limites des méthodes proposées. Les expérimentations à effectuer en priorité sont l'implémentation des méthodes d'estimation et de commande proposées sur des robots à six degrés de liberté afin de pouvoir caractériser les performances aussi bien en translation qu'en rotation. Aussi, il serait intéressant d'établir une comparaison plus étendue entre les différentes

méthodes d'estimation (Asservissement visuel virtuel - optimisation non linéaire - Filtre de Kalman) ainsi qu'entre les commandes proposées (asservissements visuels 3D et 2D et commande articulaire) en termes de performances mais aussi en terme de stabilité et de robustesse vis-à-vis des différentes perturbations pouvant affecter le système. En outre, les conditions expérimentales pourraient également être améliorées au regard du système commandé (meilleure rigidité et plus de degrés de liberté) et du système de commande (caméra compatible temps-réel, unité de traitement plus puissante, capteur avec meilleure résolution, optimisation de l'architecture logicielle et matérielle, etc.).

Concernant la mesure, nous avons vu qu'un choix pertinent des différents paramètres liés au mode d'acquisition par fenêtres d'intérêt est primordial. En vision classique, la dispersion des primitives dans l'espace et dans l'image a une grande importance sur la qualité des résultats obtenus. Dans notre cas, la dispersion est spatiale mais aussi temporelle. Cela pose alors aussi le problème de l'ordre d'acquisition des primitives. Rappelons que dans les expérimentations présentées, l'ordre d'acquisition des points dans la mire avait été généré de façon aléatoire. En outre, d'autres paramètres tels que la dimension des fenêtres et le nombre de points contenues dans les fenêtres d'intérêt ont été définis de façon empirique. Le choix de ces paramètres peut être optimisé de façon à améliorer la précision et la robustesse de l'estimation et de la commande par vision.

Dans un contexte de commande de robots parallèles, nous avons vu que la mesure de la plate-forme était plus pertinente que la mesure articulaire. Il existe cependant des structures cinématiques pour lesquelles la pose de la plate-forme ne définit pas de façon unique la configuration du robot comme c'est le cas pour robot 3-RRR. Par ailleurs, il a été établi que la configuration des jambes est encore plus pertinente pour la commande des robots parallèles puisque, à notre connaissance, elle décrit de façon unique l'état du robot [DAM07, OAM10]. Il serait donc intéressant d'étendre les commandes développées à partir de la mesure de la plate-forme à la mesure des jambes [OBAMed].

Coté capteur, d'autres modes d'acquisition peuvent être utilisés afin d'obtenir des primitives encore plus intéressantes pour la commande haute performance. En effet, il existe des caméras dotées de capteurs permettant d'acquérir des images de façon instantanée (sans temps d'intégration) et qui sont associés à une électronique permettant un accès aléatoire au signal image au niveau du capteur [Neu]. Ainsi, les méthodes d'estimation et de commande développées pour l'acquisition séquentielle de fenêtres d'intérêt pourraient être étendues à "l'acquisition progressive des primitives". De plus, ce type d'acquisition semble être plus approprié à l'acquisition des primitives qui occupent un espace important dans l'image comme pour l'observation des jambes.

Sur le plan théorique, il serait intéressant d'étudier les singularités de la matrice d'interaction par rapport au nombre de points, à la configuration dans l'espace, mais aussi par rapport au temps d'acquisition et à la vitesse. Cette étude permettrait, comme dans le cas classique, de déterminer le nombre minimal de points qui permet de contraindre suff-

isamment le système ainsi que les configurations dégénérées en statiques et en dynamique qui peuvent altérer la convergence des algorithmes. Un autre point intéressant à développer est l'étude de la robustesse des approches d'estimation et de commande proposées vis-à-vis des erreurs potentielles sur les paramètres intrinsèques de la caméra, le modèle de la mire et sur le modèle de mouvement.

Par ailleurs, le principe d'incertitude d'Heisenberg stipule que la position et de vitesse d'un objet ne peuvent être connues simultanément qu'à une précision limitée. Même la précision annoncée est très faible, ce principe se reflète tout de même sur l'estimation et la commande par vision rapide. En effet, une acquisition séquentielle sur une fenêtre temporelle très courte ne permet pas d'observer des distorsions, ce qui se traduit par une perte de précision sur la vitesse. En revanche, comme le temps d'évolution est faible, les erreurs sur le modèle d'évolution ont une répercussion moindre sur le modèle de projection, ce qui permet d'obtenir une meilleure précision sur la pose. À l'inverse, un temps d'acquisition long joue plus en faveur de la vitesse au détriment de la pose qui devient plus sensible aux erreurs sur le modèle de mouvement. Il serait ainsi intéressant d'établir des critères quantitatifs par rapport à la fenêtre temporelle d'acquisition et de développer des méthodes permettant d'obtenir le meilleurs compromis en terme de temps d'acquisition.

Par ailleurs, même si les méthodes développées autour des primitives spatio-temporelles visent plutôt des applications de suivi de trajectoire, cela n'empêche pas d'élaborer des commandes point-à-point. Pour ces applications l'étude de la stabilité, du domaine de convergence et du découplage du système deviennent critiques car l'erreur entre la configuration courante et désirée peut s'avérer importante. Dans ce contexte, il serait opportun de déterminer les primitives qui améliorent les propriétés globales du système. En effet, l'acquisition par fenêtres d'intérêt n'est pas limitée à l'observation de points. En effet, il est possible d'acquérir d'autres primitives en déplaçant les régions d'intérêt de façon à les parcourir de façon séquentielle. Cependant, le mode d'acquisition qui semble être le plus adapté dans ce cas est, encore une fois, l'accès aléatoire au niveau du capteur car il permet d'acquérir la primitive visuelle de façon quasi-continue.

En ce qui concerne plus particulièrement la commande dynamique, nous avons vu que le système en boucle fermée était perturbé par plusieurs phénomènes relatifs à la vision, au robot et à la commande elle-même. Coté vision, l'implémentation de l'estimation de la pose et de la vitesse dans la commande dynamique 3D a nécessité l'utilisation du vecteur de commande comme entrée du système afin de palier au problème du retard sur la vitesse estimée. Cependant, la précision de l'estimation dépend dans ce cas de la précision du modèle dynamique du robot. Il serait ainsi plus intéressant, du point de vue de la précision, de rendre la mesure moins indépendante des aux modèles (dynamique en l'occurrence) soit en calculant le terme d'anticipation directement à partir de la mesure par vision (si la mesure n'est pas trop bruitée) ou en augmentant le nombre d'itérations de l'algorithme d'estimation afin de converger vers la solution désirée. Ces deux solutions sont possibles

en assurant l'implémentation des algorithmes sur une unité de traitement plus puissante ou dédié.

Nous avons vu lors des expérimentations que la principale perturbation sur la commande est causée par des jeux et des flexibilités mécaniques. La loi de commande implémentée n'ayant pas été pensée pour ce type de perturbations, l'apparition d'oscillations limitant les gains de régulation. Une perspective intéressante pour la commande est donc la synthèse de commandes tenant compte de cet aspect, chose qui serait difficilement envisageable par un moyen de mesure proprioceptif.

Enfin, on peut déduire à partir de l'état de l'art actuel et des contributions des travaux présentés dans ce document que même si la lumière est la forme d'énergie la plus utilisée dans les domaines les plus exigeants en précision et en vitesse tel que la gravure des circuits intégrés (échelle nanométrique), l'interférométrie laser (mesure micrométrique) ainsi que bien d'autres applications, elle reste encore sous-exploitée en commandes hautes performances référencées vision. Le défi à relever est de développer les moyens de mesures et les méthodes les plus à mêmes de tirer profit de tout le potentiel de la vision artificielle pour la perception et la commande.

Annexe A

Notations

Notations générales

Les symboles suivants dénotent :
$\mathbf{I}_n$: la matrice d'identité de dimensions $n \times n$.
$\dot{\bullet}$: la dérivée temporelle de $\bullet$.
$\ddot{\bullet}$: la dérivée seconde de $\bullet$.
$\tilde{\bullet}$: la représentation homogène du vecteur $\bullet$.
$\hat{\bullet}$: une estimation de $\bullet$.
$[\bullet]_\times$: la matrice antisymétrique associée au vecteur $\bullet$.
$\lceil \bullet \rceil$: les crochets de Lie.

Robotique

${}^a\mathbf{T}_b$: matrice de transformation homogène de b vers a.
${}^a\mathbf{R}_b$: matrice de rotation de b vers a.
${}^a\mathbf{t}_b$: position du repère b dans a.
$\theta\mathbf{u}$: vecteur d'orientation de direction $\mathbf{u}$ (vecteur unitaire) et d'amplitude θ.
q : variable articulaire.
$\mathbf{r}$: vecteur colonne à 6 éléments correspondant au paramètres de pose.
${}^a\mathbf{v}$: vitesse cartésienne exprimée dans le repère a.
${}^a\omega$: vitesse de rotation exprimé dans le repère a.
${}^a\boldsymbol{\tau}$: torseur cinématique exprimé dans le repère a.
$\mathbf{x}$: vecteur d'état.
${}^a\dot{\mathbf{v}}$: accélération cartésienne exprimée dans le repère a.
${}^a\dot{\omega}$: accélération de rotation exprimé dans le repère a.

${}^a\dot{\boldsymbol{\tau}}$: torseur dynamique exprimé dans le repère a.
${}^a\delta\mathbf{R}$: déplacement en rotation du repère a.
${}^a\delta\mathbf{t}$: déplacement en translation du repère a.
${}^a\delta\mathbf{T}$: matrice de transformation homogène décrivant le déplacement du repère a.

Vision et asservissement visuel

$\mathbf{K}$: matrice des paramètres intrinsèques de la caméra.
${}^a\mathbf{P}$: coordonnées du point 3D dans le repère a.
$\mathbf{m}$: point dans l'image en coordonnées normalisées.
$\mathbf{p}$: coordonnées du point dans l'image exprimé en pixels.
$\mathbf{s}$: signal obtenue par vision.
$\mathbf{s}^*$: signal de référence (désiré).
$\mathbf{e}$: fonction de tâche.
$\mathbf{C}$: matrice de combinaison.
$\mathbf{J}$: jacobienne du modèle de projection par rapport aux paramètres de pose $\mathbf{r}$.
$\mathbf{L}$: matrice d'interaction liée au modèle de projection.
$\mathbf{L}_{\boldsymbol{\tau}}$: matrice d'interaction qui exprime la variation de la position des points dans l'image au torseur cinématique.
$\mathbf{L}_{\dot{\boldsymbol{\tau}}}$: matrice d'interaction qui exprime la variation de la position des points dans l'image au torseur dynamique.
$\mathbf{L}_\omega$: jacobienne qui lie la dérivée temporelle du vecteur de rotation $(\frac{d\theta\mathbf{u}}{dt})$ à la vitesse de rotation ω.

Dynamique

$\mathbb{M}$: matrice d'inertie du robot.
$\mathbb{D}$: jacobienne du robot.
$\mathbb{C}$: forces de Coriolis et forces centrifuges.
$\mathbb{H}_j$: modèle dynamique de la jambe j.
$\mathbb{G}$: forces de gravité.
$\mathbb{J}$: jacobienne entre la dernière liaison de la jambe j et le repère de la plateforme.
$\mathbf{w}$: accélération de commande.
Γ : couple de commande.

Filtre de Kalman

$\boldsymbol{A}$: matrice de transition.
$\boldsymbol{B}$: matrice de commande.
$\boldsymbol{C}$: matrice d'observation.
$\boldsymbol{u}$: vecteur d'entrée.
$\mathbf{y}$: vecteur de sortie.
ν_k : bruit de mesure.
w : bruit sur le modèle d'évolution.
$\boldsymbol{R}$: matrice de covariance du bruit de mesure.
$\boldsymbol{Q}$: matrice de covariance des erreurs de modèle.
$\boldsymbol{F}$: matrice de covariance des erreurs sur l'état estimée.
$\boldsymbol{S}$: matrice de covariance de l'innovation.
$\boldsymbol{K}$: gain de Kalman.

Annexe B

Filtrage de Kalman

B.1 Le Filtre de Kalman linéaire

Le filtre de Kalman est un filtre récursif modélisé par une chaine de Markov perturbée par un bruit Gaussien. En d'autres termes, c'est une méthode de filtrage stochastique qui permet d'estimer l'état "le plus probable" d'un système à partir de mesures bruitées et d'un modèle d'évolution plus ou moins précis.

Afin d'illustrer ce principe, considérons un système linéaire échantillonné régi par son modèle d'évolution représenté dans l'espace d'état (Fig. B.1) :

$$\mathbf{x}_k = \boldsymbol{A}\ \mathbf{x}_{k-1} + \boldsymbol{B}\ \boldsymbol{u}_{k-1} + w_{k-1} \tag{B.1}$$

où $\boldsymbol{A}$ est la matrice $n \times n$ qui relie l'état précédent à l'état actuel. Elle traduit entre autres la dynamique du système. La matrice $\boldsymbol{B}$, de dimensions $n \times m$, appelée matrice de commande, représente le lien entre un éventuel vecteur d'entrée $\boldsymbol{u}_{k-1}$ du système à l'instant $k-1$ et l'état suivant [1]. Le vecteur $w_{k-1} \in \Re^n$ est un bruit Gaussien de moyenne

1. Noter que le vecteur d'entrée $\boldsymbol{u}$ est en italic pour le différencier du vecteur décrivant l'orientation $\mathbf{u}$.

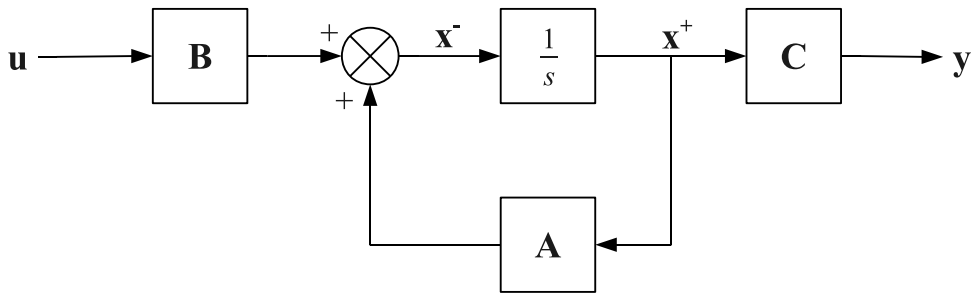


FIGURE B.1 – Représentation d'un système dans l'espace d'état.

nulle représentant la perturbation du système. Ce bruit est caractérisé par sa matrice de covariance $\boldsymbol{Q}$. Cette matrice nous informe entre autres sur le degré de précision du modèle. Le système peut être observé par un signal $\mathbf{y}_k \in \Re^m$ généralement délivré par un capteur. La mesure n'étant pas forcément faite dans l'espace d'état, la relation entre l'état du système et la mesure $\mathbf{y}$ est donnée par l'équation d'observation :

$$\mathbf{y}_k = \boldsymbol{C}\ \mathbf{x}_k + \nu_k \tag{B.2}$$

où $\boldsymbol{C}$ est une matrice $m \times n$ appelée matrice d'observation et ν_k le bruit de mesure modélisé par un bruit Gaussien de moyenne nulle et de covariance $\boldsymbol{R}$. Nous considérons que le système est stationnaire, ce qui implique que les matrices $\boldsymbol{A}$, $\boldsymbol{B}$ et $\boldsymbol{C}$ ainsi que les matrices de covariance $\boldsymbol{Q}$ et $\boldsymbol{R}$ sont constantes au cours du temps. À partir de (B.1), une estimation de l'état courant peut être calculée à partir d'une estimation de l'état précédent et de l'entrée du système :

$$\hat{\mathbf{x}}_{k|k-1} = \boldsymbol{A}\hat{\mathbf{x}}_{k-1|k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} \tag{B.3}$$

où $\hat{\mathbf{x}}_{k|k-1}$ est une estimation de l'état courant k à partir des connaissances à priori $k-1$.

Les sources d'erreurs de l'état estimé $\hat{\mathbf{x}}_{k|k-1}$ sont les erreurs dans le modèle d'évolution $\boldsymbol{Q}$ et les erreurs présentes dans l'estimation de l'état précédent. Ainsi, l'estimation prédite de la matrice de covariance des erreurs sur l'état prédit peut être calculée par :

$$\boldsymbol{F}_{k|k-1} = \boldsymbol{A}\ \boldsymbol{F}_{k-1|k-1}\ \boldsymbol{A}^T + \boldsymbol{Q} \tag{B.4}$$

Cette matrice permet entre autres d'estimer le degré de précision de l'état prédit ($\hat{\mathbf{x}}_{k|k-1}$).

Les équations (B.3) et (B.4) représentent l'étape de prédiction dans le filtre de Kalman. La mise à jour de l'état est calculée à partir de la mesure (Éq. B.2). On définit ainsi l'innovation comme étant l'erreur entre la mesure réelle et la mesure prédite :

$$\tilde{\mathbf{y}}_k = \mathbf{y}_k - \boldsymbol{C}_k\ \hat{\mathbf{x}}_{k|k-1} \tag{B.5}$$

Noter que la différence entre la prédiction et la mesure est projetée dans l'espace de mesure, ce qui permet d'intégrer des mesures incomplètes.

La matrice de covariance de l'innovation n'est rien d'autre que la somme de la matrice de covariance de l'erreur de mesure et celle de l'état prédit projeté dans l'espace de mesure :

$$\boldsymbol{S}_k = \boldsymbol{C}_k\ \boldsymbol{F}_{k|k-1}\ \boldsymbol{C}_k^T + \boldsymbol{R} \tag{B.6}$$

Étant donné que la prédiction aussi bien que la mesure sont entachées d'erreurs, il s'agit de déterminer un gain matriciel qui permet de pondérer la contribution de la prédiction et

de la mesure dans l'estimation de l'état courant. En effet, dans le cas où le bruit de sur le modèle est plus important que le bruit mesure alors c'est la contribution de cette dernière qui sera privilégiée et vice-versa. On peut démontrer que le gain optimal $\boldsymbol{K}$ appelé gain de Kalman qui permet de minimiser l'erreur d'estimation de l'état au sens des moindres carrés est donné par :

$$\boldsymbol{K} = \boldsymbol{C}_k \ \boldsymbol{F}_{k|k-1} \ \boldsymbol{C}_k^T \ \boldsymbol{S}_k^{-1} \tag{B.7}$$

Enfin, les mises à jour de l'état et de la matrice de covariance de l'erreur d'estimation "à postériori" sont calculés comme suit :

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\mathbf{y}}_k \tag{B.8}$$

$$\boldsymbol{F}_{k|k} = (\mathbf{I} - \boldsymbol{K}_k \ \boldsymbol{C}_k) \ \boldsymbol{F}_{k|k-1} \tag{B.9}$$

Nous pouvons remarquer que dans le cas idéal où la mesure ne comporte aucune erreur ($\boldsymbol{R} = 0$) le gain de Kalman devient $\boldsymbol{K} = \boldsymbol{C}^{-1}$. Dans ce cas, la mise à jour de l'état n'est basée que sur la mesure :

$$\hat{\mathbf{x}}_{k|k} = \boldsymbol{C}_k^{-1} \ \tilde{\mathbf{y}}_k$$

À l'inverse, si le modèle dévolution est parfait $\boldsymbol{F} = 0$, alors le gain de Kalman s'annule ce qui signifie qu'on n'a pas besoin de mesure :

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1}$$

B.2 Le filtre de Kalman étendu

Le filtre de Kalman étant destiné aux systèmes linéaires, son utilisation pour les systèmes non linéaires requiert la linéarisation de ces derniers.

Soit un système dynamique caractérisé par son modèle d'évolution discret :

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \boldsymbol{u}_{k-1}) + w_{k-1} \tag{B.10}$$

N'ayant pas la possibilité de connaitre la valeur du bruit sur le modèle à l'instant $(k-1)$, la prédiction de l'état courant du système à partir de l'état précédent est donnée par :

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}) \tag{B.11}$$

Comme dans le cas linéaire, cette prédiction comporte des erreurs dues aux hypothèses de modélisation w_{k-1} et à la propagation de l'erreur d'estimation de l'état précédent ($\mathbf{x}_k - \hat{\mathbf{x}}_{k-1|k-1}$) sur l'état courant. Si $f(\mathbf{x}_{k-1}, u_{k-1})$ est différentiable, la propagation de l'erreur

peut être approchée par un développement en série de Taylor du modèle d'évolution :

$$\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \approx \boldsymbol{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k-1|k-1}) + w_{k-1} \tag{B.12}$$

où $\boldsymbol{A}_k$ est la matrice Jacobienne de la fonction d'évolution f par rapport à l'état $\mathbf{x}$ calculé à l'instant k :

$$\boldsymbol{A}_k = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}} \tag{B.13}$$

Ainsi la matrice de covariance de l'erreur de prédiction est donnée par :

$$\mathbf{P}_{k|k-1} = \boldsymbol{A}_k \ \mathbf{P}_{k-1|k-1} \ \boldsymbol{A}_k^T + \boldsymbol{Q} \tag{B.14}$$

où $\mathbf{P}_{k-1|k-1}$ est la matrice de covariance de l'erreur sur la prédiction à l'instant précédent.

En outre, la mesure étant modélisée par :

$$\mathbf{y}_k = h(\mathbf{x}_k) + \nu_k \tag{B.15}$$

Une estimation du vecteur de mesure à partir de l'état prédit peut être obtenue par :

$$\hat{\mathbf{y}}_k = h(\hat{\mathbf{x}}_{k|k-1}) \tag{B.16}$$

Ainsi, l'erreur entre la mesure prédite et la mesure effectuée peut être estimée en effectuant une linéarisation au 1er ordre de l'équation (B.15) :

$$\mathbf{y}_k - \hat{\mathbf{y}}_k \approx \boldsymbol{C}_k \ (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) + \nu_k \tag{B.17}$$

où $\boldsymbol{C}_k$ est la matrice Jacobienne définie par :

$$\boldsymbol{C}_k = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1}} \tag{B.18}$$

Ainsi, l'innovation représente la différence entre la mesure effectuée et la mesure prédite :

$$\tilde{\mathbf{y}}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k \tag{B.19}$$

L'innovation étant entachée des bruits de mesures ν_k et des erreurs de l'état prédit propagées dans la prédiction de l'observation $\hat{\mathbf{y}}_k$, la matrice de covariance de l'innovation est estimée par :

$$\boldsymbol{S} = \boldsymbol{C} \ \boldsymbol{F}_{k|k-1} \ \boldsymbol{C}^T + \boldsymbol{R} \tag{B.20}$$

Enfin, le gain de Kalman ainsi que la mise à jour de l'état sont calculés de la même manière que dans le cas linéaire (équations (B.7), (B.8) et (B.9)).

Bibliographie

[AAALM06] O. Ait-Aider, N. Andreff, J.M. Lavest, and P. Martinet. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *Proceedings of the 9th European Conference on Computer Vision, ECCV'06*, volume 2, pages 56–68, Graz, Austria, May 7-13 2006.

[ACVT04] L. Armesto, S. Chroust, M. Vincze, and J. Tornero. Multi-rate fusion with vision and inertial sensors. In *Proc. IEEE International Conference on Robotics and Automation ICRA '04*, volume 1, pages 193–199 Vol.1, 2004.

[AEH02] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. *The International Journal of Robotique Research*, 21(8) :679–700, 2002.

[AHE01] N. Andreff, R. Horaud, and B. Espiau. Robot hand-eye calibration using structure-from-motion. *The International Journal of Robotics Research*, 20 :228–248, March 2001.

[ATS+09] L. Angel, A. Traslosheros, J. Sebastian, L. Pari, R. Carelli, and F. Roberti. Vision-based control of the robotenis system, 2009.

[ATYM93] P.K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transactions on Robotics and Automation*, 9(2) :152–165, 1993.

[Ban85] A Bandyopadhyay. Perception of rigid motion from spatio-temporal derivatives. Technical Report 157, University of Rochester, Departement of Computer Science, March 1985.

[BC95] F. Bensalah and F. Chaumette. Compensation of abrupt motion changes in target tracking by visual servoing. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 181–187, 1995.

[Ber92] J.R. Bergen. Hierarchical model-based motion estimation. In Springer, editor, *European Conference on Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 237–252, Santa Margherita, Italy, May 1992.

[BG05] G Barrette and C.M. Gosselin. Determination of the dynamic workspace of cable-driven planar parallel mechanisms. *Journal of Mechanical Design*, 127(2) :242–248, 2005.

[BRM97] K.J. Bradshaw, I.D. Reid, and D.W. Murray. The active recovery of 3D motion trajectories and their use in prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3) :219–234, 1997.

[Bru99] H. Bruyninckx. Dualities between serial and parallel "321" manipulators. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1532–1537, 1999.

[BWY06] C. Brecher, M. Weck, and T. Yamasaki. Controller-integrated predictive oscillation compensation for machine tools with parallel kinematics. *International Journal of Machine Tools and Manufacture*, 46(2) :142 – 150, 2006.

[C98] A. Crétual and . *Utilisation d'informations visuelles dynamiques en asservissement visuel*. PhD thesis, Université de Rennes 1, 1998.

[CC00] A. Crétural and F. Chaumette. Dynamic stabilisation of a pan and tilt camera for submarine image visualisation. *Computer Vision and Image Understanding*, 79 :47–69, 2000.

[CC01a] A. Crétual and F. Chaumette. Application of motion-based visual servoing to target tracking. *International Journal of Robotics Research*, 20(11) :878–890, November 2001.

[CC01b] A. Crétual and F. Chaumette. Visual servoing based on image motion. *International Journal of Robotics Research*, 20(11) :857–877, November 2001.

[CG93] P.I. Corke and M. Goods. Controller design for high performance visual servoing. In *12th World Congr. IFAC'93*, pages 395–398, Australia, July 1993.

[CG96] P.I. Corke and M.C. Good. Dynamic effects in visual closed-loop systems. *IEEE Transaction on Robotics and Automation*, 12(5) :671–683, October 1996.

[CGdMF06] L. Cuvillon, J. Gangloff, M. de Mathelin, and A. Forgione. Towards robotized beating heart tecabg : assessment of the heart dynamics using high-speed vision. *Computer Aided Surgery*, 11(5) :267 – 277, September 2006.

[CH01] P. I. Corke and S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transaction on Robotics and Automation*, 17(4) :507–515, August 2001.

[CH06] F. Chaumette and S. Hutchinson. Visual servo control, part I : Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4) :82–90, December 2006.

[CH07] F. Chaumette and S. Hutchinson. Visual servo control, part II : Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1) :109–118, March 2007.

[Cha04] F. Chaumette. Image moments : a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4) :713–723, 2004.

[CPP06] M. Callegari, M.C. Palpacelli, and M. Principi. Dynamics modelling and control of the 3-RCC translational platform. *Mechatronics*, 16(10) :589–605, December 2006.

[CRE91] F. Chaumette, P. Rives, and B. Espiau. Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2248–2253, Sacramento, California, April 1991.

[CRVG+04] G. L. Cembrano, A. Rodriguez-Vazquez, R. C. Galan, F. Jimenez-Garrido, S. Espejo, and R. Dominguez-Castro. A 1000 fps at 128 x 128 vision processor with 8-bit digitized I/O. *IEEE Journal of Solid-State Circuits*, 39(7) :1044–1055, 2004.

[CW87] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 1–6, New York, NY, USA, 1987. ACM.

[CW98] D. Chablat and P. Wenger. Working modes and aspects in fully parallel manipulators. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, pages 1964–1969, 1998.

[CW03] D. Chablat and P. Wenger. Architecture optimization of a 3-dof translational parallel mechanism for machining applications, the orthoglide. *IEEE Transactions on Robotics and Automation*, 19(3) :403–410, 2003.

[DAM07] T. Dallej, N. Andreff, and P. Martinet. Image-based visual servoing of the I4R parallel robot without proprioceptive sensors. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, pages 1709–1714, Rome, 2007.

[DAMM06] T. Dallej, N. Andreff, Y. Mezouar, and P. Martinet. 3D pose visual servoing relieves parallel robot control from joint sensing. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4291–4296, 2006.

[DC99] B. Dasgupta and P. Choudhury. A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators. *Mechanism and Machine Theory*, 34(6) :801–824, August 1999.

[DD95] D. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1/2) :123–141, June 1995.

[DH06] B. Denkena and C Holz. Advanced position and force control concepts for the linear direct driven hexapod palida. In *Chemnitz Parallel Kinematics Seminar*, pages 359–378, April 2006.

[ECR92] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3) :313–326, June 1992.

[EPK+03] T.G. Etoh, D. Poggemann, G. Kreider, H. Mutoh, A.J.P. Theuwissen, A. Ruckelshausen, Y. Kondo, H. Maruno, K. Takubo, H. Soya, K. Takehara, T. Okinaka, and Y. Takano. An image sensor which captures 100 consecutive frames at 1000000 frames/s. *IEEE Transactions on Electron Devices*, 50(1) :144–151, 2003.

[FJS01] M. Ficocelli and F. Janabi-Sharifi. Adaptive filtering for pose estimation in visual servoing. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 19–24 vol.1, 2001.

[FNTI06] N. Furukawa, A. Namiki, S. Taku, and M. Ishikawa. Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system. In *Proc. IEEE International Conference on Robotics and Automation*, pages 181–187, 2006.

[FS05] J. Falcou and J. Serot. E.V.E., an object oriented SIMD library. *Scalable Computing : Practice and Experience*, 6(4) :31–41, December 2005.

[Gal46] E. Galois. Sur les conditions de résolution des équations par radicaux. *Journal de mathématiques pures et appliquées*, pages 417–433, 1846.

[GdM03] J. Gangloff and M. de Mathelin. High-speed visual servoing of a 6 DOF manipulator using multivariable predictive control. *Advanced Robotics. Special issue : advanced 3D vision and its application to robotics*, 17(10) :993–1021, December 2003.

[GGdM+05] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M.M.A. Sanchez, and J. Marescaux. Active filtering of physiological motion in robotized surgery using predictive control. *IEEE Transactions on Robotics*, 21(1) :67–79, 2005.

[Gog04] G. Gogu. Structural synthesis of fully-isotropic translational parallel robots via theory of linear transformations. *European Journal of Mechanics - A/Solids*, 23(6) :1021–1039, November 2004.

[GV05] P. Gemeiner and M. Vincze. Motion and structure estimation from vision and inertial sensor data with high speed cmos camera. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2005*, pages 1853–1858, 2005.

[HHC96] S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5) :651–670, October 1996.

[HM95] R. Horaud and O. Monga. *Vision par ordinateur : Outils fondamentaux.* Hermes Sciences, 1995.

[HM02] T. Hamel and R. Mahony. Visual servoing of an under-actuated dynamic rigid-body system : an image-based approach. *IEEE Transactions on Robotics and Automation*, 18(2) :187–198, 2002.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[Hus96] M. Husty. An algorithm for solving the direct kinematics of general Gough-Stewart platforms. *Mechanism and Machine Theory*, 31(4) :365–380, 1996.

[HZ00] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[IMT92] M. Ishikawa, A. Morita, and N. Takayanagi. High speed vision system using massively parallel processing. In *Proc. lEEE/RSJ Int Intelligent Robots and Systems Conf*, volume 1, pages 373–377, 1992.

[INI96] I. Ishii, Y. Nakabo, and M. Ishikawa. Target tracking algorithm for 1 ms visual feedback system using massively parallel processing. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, pages 2309–2314, 1996.

[IO05] M. Iwatsuki and N. Okiyama. A new formulation of visual servoing based on cylindrical coordinate system. *IEEE Trans. on Robotics and Automation*, 21(2) :266–273, 2005.

[JLV99] L. Jetto, S. Longhi, and G. Venturini. Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation*, 15(2) :219–229, 1999.

[JU97] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, pages 182–193, 1997.

[KCN+00] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes. Stable visual servoing of camera-in-hand robotic systems. *IEEE/ASME Transactions on Mechatronics*, 5(1) :39–48, March 2000.

[KD99] Wisama K. and Etienne D. *Modélisation Identification et commande des robots*. Hermes Science Publications, 1999.

[KI07a] W. Khalil and O. Ibrahim. General solution for the dynamic modeling of parallel robots. *Journal of Intelligent and Robotic Systems*, 49 :19–37, 2007.

[KI07b] T. Komuro and M. Ishikawa. A moment-based 3D object tracking algorithm for high-speed vision. In *Proc. IEEE International Conference on Robotics and Automation*, pages 58–63, Roma, april 2007.

[KIIY03] T. Komuro, I. Ishii, M. Ishikawa, and A. Yoshida. A digital vision chip specialized for high-speed target tracking. *IEEE transactions on Electron Devices*, 50(1) :191–199, 2003.

[KV09] D. Kragic and M. Vincze. Vision for robotics. *Found. Trends Robot*, 1(1) :1–78, 2009.

[LD08] E. Laroche and J. Delavigne. Dynamical effects of vision-based position measurement. In *IFAC World Congress*, pages 7522–7527, Seoul, South Korea, 2008.

[LHW04] Y. Liu, A.W. Hoover, and I.D. Walker. A timing model for vision-based control of industrial robot manipulators. *IEEE Transactions on Robotics and Automation*, 20(5) :891–898, Oct. 2004.

[LK90] S. Lee and Y. Kay. An accurate estimation of 3-D position and orientation of a moving object for robot stereo vision : Kalman filter approach. In *Proc. IEEE International Conference on Robotics and Automation*, pages 414–419 vol.1, 1990.

[Lon06] R. Longchamp. *Commande numérique des systèmes dynamiques*. Presses Polytechniques et Universitaires Romandes, 2006.

[LSCH03] S.H. Lee, J.B. Song, W.C. Choi, and D. Hong. Position control of a stewart platform using inverse dynamics control with approximate dynamics. *Mechatronics*, 13(6) :605–619, July 2003.

[LSV06] V. Lippiello, B. Siciliano, and L. Villani. Adaptive extended Kalman filtering for visual motion estimation of 3d objects. *Control Engineering Practice*, 15(1) :123–134, May 2006.

[LVD98] J. M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration. In *Proceedings of ECCV98*, pages 158–174, Freiburg, Germany, June 1998.

[Mal98] E. Malis. *Contributions ï£¡ la modélisation et ï£¡ la commande en asservissement visuel*. PhD thesis, Université de Rennes I, IRISA, 1998.

[Mar07] E. Marchand. Control camera and light source positions using image gradient information. In *Proc. IEEE Int Robotics and Automation Conf*, pages 417–422, 2007.

[MB05] E. Malis and S. Benhimane. A unified approach to visual tracking and servoing. *Robotics and Autonomous Systems*, 52(1) :39–52, 2005.

[MBG96] P. Martinet, F. Berry, and J. Gallice. Use of first derivative of geometric features in visual servoing. In *Proc. IEEE International Conference on Robotics and Automation*, volume 4, pages 3413–3419 vol.4, 1996.

[MC02a] E. Malis and F. Chaumette. Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Transactions on Robotics and Automation*, 18(2) :176–186, 2002.

[MC02b] E. Marchand and F. Chaumette. Virtual visual servoing : a framework for real-time augmented reality. *EUROGRAPHICS*, 21(3) :289–298, septembre 2002.

[MC02c] Y. Mezouar and F. Chaumette. Path planning for robot image-based control. *IEEE Transaction on Robotics and Automation*, 18(4) :534–549, August 2002.

[MC10] E. Marchand and C. Collewet. Using image gradient as a visual feature for visual servoing. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'10*, Taipei, Taiwan, October 2010.

[MCB99] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Transaction on Robotics and Automation*, 15(2) :234–246, April 1999.

[MDP07] R. Mosqueron, J. Dubois, and M. Paindavoine. High-speed smart camera with high resolution. *EURASIP Journal Embedded System*, 2007(1) :1–16, 2007.

[Mer90] J.P. Merlet. An algorithm for the forward kinematics of general 6 dof parallel manipulators. Technical Report 1331, INRIA, 1990.

[Mer09] J. P. Merlet. *Parallel Robots*. Springer Netherlands, 2009.

[MR82] S. Megahed and M. Renaud. Minimization of the computation time necessary for the dynamic control of robot manipulators. In *Proc. of the 12th International Symposium on Industrial Robots/6th Conference on Industrial Robot Technology*, pages 469 –478, Paris, June 1982.

[MT76] K. Myers and B. Tapley. Adaptive sequential estimation with unknown noise statistics. 21(4) :520–523, 1976.

[NC03] O. Nasisi and R. Carelli. Adaptive servo visual robot control. *Robotics and Autonomous Systems*, 43(1) :51–78, April 2003.

[NdlORC+05] V. Nabat, M. de la O Rodriguez, O. Company, S. Krut, and F. Pierrot. Par4 : very high speed parallel robot for pick-and-place. In *Proc. IEEE/RSJ International Conference Intelligent Robots and Systems (IROS 2005)*, pages 553–558, 2005.

[Neu] www.neuricam.com.

[NITM00] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno. 1ms column parallel vision system and its application of high speed target tracking. In *Proc. IEEE International Conference on Robotics and Automation*, pages 650–655, San Francisco, USA, April 2000.

[OAM10] E. Özgür, N. Andreff, and P. Martinet. Vector-based dynamic modeling and control of the quattro parallel robot by means of leg orientations. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'10)*, pages 638 – 643, Anchorage, AK, 2010.

[OBAMed] E. Özgür, N. Bouton, N. Andreff, and P. Martinet. Dynamic control of the quattro robot by the leg edgels. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'11)*, Shanghai, China., 2011. (Submitted).

[OOY+03] R. Okada, J. Oaki, D. Yamamoto, N. Kondoh, H. Miyazaki, K. Uesugi, J. Amemiya, K. Shirakawa, and A. Kunimatsu. High-speed object tracking in ordinary surroundings based on temporally evaluated optical flow. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 1, pages 242–247, 2003.

[OZW02] P. R. Ouyang, W. J. Zhang, and F. X. Wu. Nonlinear pd control for trajectory tracking with consideration of the design for control methodology. In *Proc. IEEE International Conference on Robotics and Automation ICRA '02*, volume 4, pages 4126–4131, 2002.

[Pac09] F. Paccot. *Contributions ï£¡ la commande dynamique référencée capteur de robots parallèles.* PhD thesis, Université Blaise Pascal - Clermont II, Juin 2009.

[PAM09] F. Paccot, A. Andreff, and P. Martinet. A review on the dynamic control of parallel kinematic machines : Theory and experiments. *The International Journal of Robotics Research*, 28(3) :395–416, 2009.

[PC05] J. Park and W.K. Chung. Geometric integration on Euclidean group with application to articulated multibody systems. *IEEE Transactions on Robotics and Automation*, 21(5) :850–863, 2005.

[PLA+08] F. Paccot, P. Lemoine, N. Andreff, D. Chablat, and Philippe Martinet. A vision-based computed torque control for parallel kinematic machines. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1556–1561, April 2008.

[PML99] J.A. Piepmeier, G.V. McMurray, and H. Lipkin. A dynamic quasi-newton method for uncalibrated visual servoing. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1595–1600 vol.2, 1999.

[Pri02] G. Pritschow. Influence of the dynamic stiffness on the accuracy of PKM. In *Chemnitz Parallel Kinematic Seminar*, pages 313–333, 2002.

[RCGVS07] A. Ranftl, L. Cuvillon, J. Gangloff, and J. Vander Sloten. High speed visual servoing with ultrasonic motors. In *Proc. IEEE International Conference on Robotics and Automation*, Roma, april 2007.

[SA05] I.A.K. Saeed and N.V. Afzulpurkar. Real time, dynamic target tracking using image motion. In *Proc. IEEE International Conference on Mechatronics ICM '05*, pages 241–246, 2005.

[SBC96] V. Sundareswaran, P. Bouthemy, and F. Chaumette. Exploiting image motion for active vision in a visual servoing framework. *International Journal of Robotics Research*, 15(6) :629–645, December 1996.

[SELB91] C. Samson, B. Espiau, and M. Le Borgne. *Robot Control : The Task Function Approach*. Oxford University Press, 1991.

[SIT07] Y. Sumi, Y. Ishiyama, and F. Tomita. Robot-vision architecture for real-time 6-dof object localization. *Computer Vision and Image Understanding*, 105(3) :218–230, March 2007.

[SJ07] D. Santosh and C. Jawahar. Visual servoing in non-rigid environments : A space-time approach. In *Proc. IEEE International Conference on Robotics and Automation*, pages 10–14, Roma, Italy, April 2007.

[SNI04] T. Senoo, A. Namiki, and M. Ishikawa. High-speed batting using a multi-jointed manipulator. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1191–1196 Vol.2, 2004.

[Sub89] M. Subbarao. Interpretation of image flow : a spatio-temporal approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3) :266–278, 1989.

[TC05] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics and Automation*, 21(6) :1116–1127, 2005.

[TFC09] R. Tatsambon Fomena and F. Chaumette. Improvements on visual servoing from spherical targets using a spherical projection model. *IEEE Transactions on Robotics and Automation*, 25(4) :874–886, August 2009.

[TKH07] J. Takei, S. Kagami, and K. Hashimoto. 3,000-fps 3-D shape measurement using a high-speed camera-projector system. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007*, pages 3211–3216, 2007.

[TMCG02] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice. Position based visual servoing : keeping the object in the field of vision. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1624–1629, 2002.

[TO00] Camillo J. Taylor and James P. Ostrowski. Robust vision-based pose control. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2734–2740, April 2000.

[TZR99] J. Tlusty, J. Ziegert, and S. Ridgeway. Fundamental comparison of the use of serial and parallel kinematics for machines tools. *CIRP Annals - Manufacturing Technology*, 48(1) :351 – 356, 1999.

[URLP04] M. Ulrich, M. Ribi, P. Lang, and A. Pinz. A new high speed CMOS camera for real-time tracking application. In *Proc. IEEE International Conference on Robotics and Automation*, New Orleans, april 2004.

[VAC+02] M. Vincze, M. Ayromlou, S. Chroust, M. Zillich, W. Ponweiser, and D. Legenstein. Dynamic aspects of visual servoing and a framework for real-time 3D vision for robotics. *Lecture notes in computer science*, 2238 :101–121, 2002.

[WB95] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[WH91] K. J. Waldron and K. H. Hunt. Series-Parallel dualities in actively coordinated mechanisms. *The International Journal of Robotics Research*, 10(5) :473–480, 1991.

[WHB96] L.W Wilson, C.W Hulls, and G.S Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5) :684–696, October 1996.

[WJV+05] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics*, 24(3) :765–776, 2005.

[WM93] J. Wang and O. Masory. On the accuracy of a stewart platform. I. the effect of manufacturing tolerances. In *Proc. IEEE International Conference on Robotics and Automation*, pages 114–120, 1993.

[WSN87] L. Weiss, A. Sanderson, and C. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5) :404– 417, Oct 1987.

[WU00] P. Wira and J-P. Urban. A new adaptive kalman filter applied to visual servoing tasks. In *International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, pages 267–270, Brighton, UK, 2000.

[WW92] J. Wang and W.J. Wilson. 3D relative position and orientation estimation using kalman filter for robot control. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2638–2645 vol.3, 1992.

[YN01] S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *Proc. IEEE Virtual Reality*, pages 71–78, 2001.

[ZLWS03] J. Zhang, R. Lumia, J. Wood, and G. Starr. Delay dependent stability limits in high performance real-time visual servoing systems. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 1, pages 485–491, 27–31 Oct. 2003.

[ZP99] H. Zhang and J. P. Pstrowski. Visual servoing with dynamics : Control of an unmanned blimp. In *Proc. IEEE International Conference on Robotics and Automation*, pages 648–623, May 1999.

[ZR95] H. Zhuang and Z. S. Roth. *Camera Aided Robot Calibration*. CRC Press, Inc., Boca Raton, FL, USA, 1995.

Résumé

La vision artificielle est un moyen de perception très apprécié en robotique. Les applications les plus courantes de la vision en robotique manipulatrice sont l'estimation de pose et la commande. D'un point de vue conceptuel, l'utilisation de la vision artificielle permet d'améliorer les performances de la commande des robots en termes de précision et de robustesse (vis-à-vis des erreurs sur les paramètres géométriques du robot). De plus, cette mesure est d'autant plus pertinente pour les robots parallèles puisque l'état de ces derniers est généralement mieux défini par la pose de l'effecteur que par les mesures articulaires, traditionnellement utilisées en commande. Cependant, les systèmes de vision classiques ne permettent pas de satisfaire les exigences des commandes hautes performances à cause de leur période d'acquisition et de leur temps de latence trop grands. Pour pallier ce problème, l'approche proposée dans cette thèse est de procéder à une acquisition séquentielle de fenêtres d'intérêt. En effet, le fait de ne transmettre que les régions de l'image contenant les primitives visuelles utiles a pour effet de diminuer la quantité de données à transmettre ce qui permet réduire la période d'acquisition et le temps de latence. De plus, l'acquisition non simultanée des primitives offre la possibilité d'estimer la pose et la vitesse du robot de façon conjointe. Différentes méthodes d'estimation et plusieurs schémas de commandes cinématiques et dynamiques utilisant ce mode d'acquisition ont ainsi été proposés dans ce document. Les résultats expérimentaux obtenus en commande dynamique par vision d'un robot parallèle montrent, pour la première fois, que la commande référencée vision peut être plus performante que la commande articulaire.

Abstract

Visual sensing is very appreciated in manipulation robotics since it provides measures for pose estimation and robot control. Conceptually, vision allows improving the performance of manipulator robots control from accuracy and robustness (against kinematic parameters errors) points of view. In addition, vision-based control is particularly relevant for parallel robot manipulators. In fact, the state of these robots is usually better described by the pose of their mobile platform than by their articular joints. Thus, the use of vision simplifies the control of parallel robots. However, the typical vision systems do not fulfil the dynamic control constraints on the acquisition frequency and the latency. The approach proposed in this thesis is to perform a sequential acquisition of regions of interest which contain the useful visual features to cut down the data amount to transmit. Thus allows for reducing the acquisition period and latency. In addition, the sequential acquisition allows for estimating both the pose and the velocity of the robot platform. Thanks to this acquisition method, several control laws are proposed. The experimental results show that the performance of the proposed vision based dynamic control laws are, for the first time, better than classical dynamic control.