N° d'ordre : 2067

THÈSE

Présentée devant

L'UNIVERSITÉ DE RENNES I

Pour obtenir

le grade de: Docteur de l'Université de Rennes I
Mention: Traitement du Signal et Télécommunications

par

Armel CRETUAL

Equipe d'accueil: TEMIS, puis VISTA
IRISA, RENNES

Ecole doctorale: Informatique, Traitement du Signal et Télécommunications

Composante universitaire: Structures et Propriétés de la Matière (S.P.M.)

Titre de la thèse

Asservissement visuel à partir d'informations de mouvement dans l'image

Soutenue le 10 Novembre 1998, devant la commission d'examen composée de :

M.	Jean-Jacques	FUCHS	Président
MM.	Jan-Olof	EKLUNDH	Rapporteurs
	Patrick	RIVES	
MM.	Patrick	BOUTHEMY	Examinateurs
	François	CHAUMETTE	
	Philippe	MARTINET	

Remerciements

J'adresse de sincères remerciements à Patrick RIVES et Jan-Olof EKLUNDH qui ont très amicalement accepté la tâche d'être rapporteur de cette thèse.

Je remercie également Patrick BOUTHEMY et Philippe MARTINET pour leur participation au jury en tant qu'examinateur.

De la même manière, je remercie Jean-Jacques FUCHS pour son rôle de président du jury.

Il va de soi qu'on ne peut pas passer trois années dans une équipe sans tisser des liens. Que toutes les personnes de l'ex-équipe TEMIS et de ses deux descendants VISTA et TEMICS qui ont permis d'assurer que cette période se passe au mieux, aussi bien sur le plan humain que sur le plan scientifique, soient également assurées de mes remerciements. En particulier, je remercie Fabien SPINDLER pour sa patience face à mes nombreux problèmes au jour le jour.

Mes plus vifs remerciements s'adressent à François CHAUMETTE qui m'a encadré pendant ces trois années de travail. Sa très grande disponibilité au quotidien associée à une invariable bonne humeur et la confiance qu'il m'a, je crois, témoignée en me laissant suivre mon chemin, ont contribué à faire de cet encadrement un modèle du genre.

Enfin, je pense que ces trois années de travail n'auraient jamais abouti sans la présence de Nathalie (alias Maeva), qui a eu l'extrême patience de m'écouter parfois parler de dérivation des paramètres quadratiques du mouvement et autres, qui a également relu avec moult attention ce manuscrit et corrigé de nombreuses erreurs, mais qui a aussi, et surtout, eu le bon goût de me rappeler que tout n'est pas asservissement visuel. Elle sait combien je la remercie pour tout cela et le reste.

Table des matières

Introduction **5**

1 Etat de l'art **9**
1.1 Caractérisation du mouvement dans l'image 9
1.1.1 Champ dense 9
1.1.2 Modèle de mouvement 10
1.1.3 Carte binaire du mouvement 18
1.2 La poursuite de cibles 19
1.3 Navigation de robots mobiles et évitement d'obstacles 22
1.4 Tâches de positionnement 28

2 Asservissement visuel géométrique à partir de mesures dynamiques **35**
2.1 Rappels sur l'asservissement visuel 2D 36
2.1.1 Principe 36
2.1.2 Schéma de commande 37
2.2 Principe de l'utilisation d'information 2D à partir de mesures visuelles dynamiques 39
2.3 Poursuite d'un objet mobile en pan et tilt 40
2.3.1 Un rapide tour d'horizon de travaux existants 40
2.3.2 Loi de commande associée 42
2.3.3 Simulation 43
2.3.4 Expérimentations 46
2.3.5 Suivi de piéton 51
2.4 Liaison rigide avec un objet mobile 54
2.4.1 Loi de commande 55
2.4.2 Simulations 57
2.4.3 Discussion 69
2.5 Pourquoi n'appliquer cette méthode qu'à des tâches de suivi? 69
2.5.1 Obtention des données nécessaires à la commande 70
2.5.2 Positionnement par rapport à une scène 72

3 Asservissement visuel $\frac{d2D}{dt}$ 77
3.1 Comparaison des asservissements visuels géométrique et dynamique 78
3.1.1 Comparaison schématique 78
3.1.2 Asservissement visuel dynamique 79
3.2 Relations d'interaction des paramètres du mouvement 80
3.2.1 Expression des dérivées des paramètres du modèle quadratique du mouvement projeté 80
3.2.2 Calcul des matrices Q_{p_i} 82
3.3 Idées pour la commande 85
3.3.1 Contexte 85
3.3.2 Cas d'une matrice d'interaction de rang plein 88
3.3.3 Cas d'une matrice d'interaction de rang non-plein 91

4 Exemple de tâches en asservissement visuel $\frac{d2D}{dt}$ 99
4.1 Poursuite en *pan* et *tilt* d'un objet mobile 100
4.1.1 Présentation de la tâche 100
4.1.2 Loi de commande associée 100
4.1.3 Résultats de simulation 102
4.1.4 Résultats sur site robotique 105
4.1.5 Comparaison avec l'approche par contrôle de la position estimée du centre de l'objet 108
4.2 Suivi rigide d'un objet mobile 109
4.2.1 Présentation de la tâche 109
4.2.2 Loi de commande associée 109
4.2.3 Résultats de simulation 113
4.3 Positionnement de la caméra parallèlement à un plan 117
4.3.1 Présentation de la tâche 117
4.3.2 Loi de commande associée à la tâche d'alignement 118
4.3.3 Loi de commande associée à la tâche d'alignement couplée avec la fixation 124
4.3.4 Résultats de simulation 129
4.3.5 Résultats sur site robotique 138
4.4 Suivi d'une trajectoire d'observation d'une surface plane 146
4.4.1 Présentation de la tâche 146
4.4.2 Loi de commande associée 148
4.4.3 Résultats de simulation 152
4.4.4 Résultats sur site robotique 156

Conclusion 163

A Précision de l'algorithme RMRm d'estimation des paramètres d'un modèle polynomial de mouvement apparent dans l'image 167
A.1 Tests dans le cas de mouvements connus de la caméra 167
A.2 Comparaison avec une estimation basée sur des mesures géométriques . . . 169

B Influence des paramètres de calibration sur le modèle de mouvement apparent dans l'image 173
B.1 Modélisation de la projection 173
B.2 Passage pixel-mètre des paramètres du mouvement 174
B.3 Influence de la distorsion radiale 176

Bibliographie 178

Introduction

Pendant longtemps, la robotique ne s'est concentrée que sur la réalisation d'actions parfaitement définies au préalable. Pour donner un exemple parmi tant d'autres, nous pouvons citer, la saisie par un robot manufacturier d'un objet se trouvant à une position totalement connue.

Mais, depuis quelques années déjà, les robots ont franchi un grand pas en effectuant désormais des tâches dans un environnement, au moins partiellement inconnu. La nécessité, dans ce cas, de réagir aux éventuels changements de configuration du milieu, oblige à appréhender l'espace dans lequel le robot évolue. Un des capteurs extero-ceptifs (c'est-à-dire acquérant des données sur le milieu extérieur d'un système, contrairement aux capteurs proprio-ceptifs) fournissant la plus grande richesse d'information, est la caméra vidéo. C'est donc tout naturellement que de nombreux travaux se sont penchés sur le couplage de la vision et de la robotique. Ce domaine de recherche a pris le nom d'asservissement visuel quand il s'agit de commander les mouvements d'un robot à partir d'informations visuelles.

Le besoin d'extraire des informations fiables des images acquises implique l'utilisation d'un traitement d'image qui peut se révéler coûteux en temps de calcul. Cela vient alors en contradiction avec la nécessité de contrôler les mouvements du robot à une cadence élevée pour garantir les plus grandes réactivité et stabilité possibles. C'est pourquoi, la plupart des études en asservissement visuel ont, dans un premier temps, éludé le problème en ne considérant que des objets dédiés à l'application visée, et en particulier, permettant de simplifier au maximum l'algorithme d'extraction des primitives visuelles associées.

Les images les plus aisées à traiter sont les images binaires. Ainsi, les premiers objets utilisés en asservissement visuel ont été des cibles conçues spécifiquement pour obtenir une image binaire grâce à un simple seuillage de la luminance de chaque pixel. Considérons, par exemple, un objet du type disques noirs sur fond blanc, et que l'objectif du traitement d'image est de déterminer la position du centre de gravité de ces disques. Même si une image de cet objet peut être bruitée, il reste simple d'en déduire une image binaire permettant de séparer les pixels correspondant à un point d'un des disques blancs, de ceux correspondant à un point du fond noir. Il est alors élémentaire de retrouver la position dans l'image de la projection de leur centre de gravité.

Cependant, le monde n'est pas fait, loin de là, uniquement d'objets aussi simples. Par conséquent, il apparaît nécessaire dorénavant d'étendre le champ d'application de l'asservissement visuel à des environnements plus réalistes. Nous devons alors nous demander quels algorithmes de traitement d'image nous pouvons mettre en œuvre pour y parvenir.

Une partie de la réponse peut peut-être se trouver dans le monde animal, lorsque l'on sait que nombreuses sont les espèces dont l'œil n'est capable de détecter que des objets en mouvement, ou du moins, est beaucoup plus réceptif aux variations temporelles dans le champ de vision qu'aux différences d'intensité lumineuse reçue entre deux zones de ce champ. Parmi cette diversité, nous pouvons noter certains reptiles (les vipères, par exemple), les insectes hyménoptères (les mouches, les abeilles, pour ne citer que ceux-là), les poissons cartilagineux (les requins, les raies, entre autres), etc.. Ainsi, lorsque l'on observe une scène complexe, il semble plus facile d'extraire des informations de déplacement apparent dans l'image, que des données géométriques.

Cela peut se vérifier en pratique, lorsque l'on compare, en terme de temps de calcul, les divers algorithmes de traitement d'image permettant d'extraire des informations de mouvement, à ceux fournissant des informations telles que les positions de points singuliers (des coins, par exemple). Les progrès de la puissance informatique aidant, il est désormais possible de traiter les variations entre deux images successivement acquises par une caméra pour obtenir de telles mesures de mouvement, à une cadence suffisamment élevée pour que ces données puissent être utilisées dans une boucle de contrôle robotique.

C'est donc l'utilisation de telles informations visuelles de mouvement apparent - que nous qualifions de dynamiques - dans un contexte d'asservissement visuel, que nous nous proposons d'étudier dans cette thèse. Pour cela, nous nous sommes inspirés de l'approche par fonction de tâche appliquée à l'asservissement visuel géométrique, telle qu'elle a été développée préalablement au sein de l'équipe.

Ce mémoire se divise en quatre chapitres, qui portent, respectivement, sur les points suivants :

- Nous dressons, dans le premier chapitre, un état de l'art des divers travaux réalisés en vision robotique, dans lesquels le mouvement apparent dans l'image est pris en compte. Au préalable, nous donnons rapidement quelques manières d'aborder ce mouvement projeté. Les travaux en question sont ensuite classés en trois catégories distinctes, selon l'objectif de la tâche à réaliser. Il s'agit, dans l'ordre, de tâches de suivi d'objet en mouvement, de navigation de robots mobiles autonomes, et enfin, de positionnement par rapport à un objet de référence.
- Dans le deuxième chapitre, nous donnons une première utilisation possible du mouvement dans l'image en asservissement visuel. En se basant sur la définition de la vitesse, à savoir qu'elle est la dérivée de la position, nous proposons de retrouver les coordonnées dans l'image de points d'intérêt appartenant à un objet donné, à

partir de l'estimation, à chaque itération, de la vitesse apparente de cet objet, et de la connaissance initiale des positions des points considérés.
Nous montrons alors que nous pouvons utiliser les lois de commande développées en asservissement visuel géométrique, pour contraindre ces points à atteindre une position désirée dans l'image. Il s'agit, en fait, d'asservissement visuel géométrique à partir de mesures visuelles dynamiques. Nous présentons enfin deux applications de cette approche : d'une part, le suivi d'un objet mobile en *pan* et *tilt*, c'est-à-dire en contrôlant les deux degrés de liberté en rotation panoramique et inclinaison, de façon à conserver l'objet centré dans l'image, et d'autre part, le suivi rigide d'un objet, c'est-à-dire avec l'objectif de maintenir la caméra immobile par rapport à l'objet.

- Nous proposons ensuite un élargissement du champ d'application de l'asservissement visuel. Contrairement au chapitre 2, où la prise en compte de la vitesse dans l'image n'est qu'un moyen d'obtenir des informations visuelles géométriques dans un environnement complexe, le mouvement apparent est, dans cette nouvelle approche, utilisé en tant que mesure directe ainsi que pour définir la consigne à atteindre par le système. Ce concept est appelé asservissement visuel dynamique.
  En calquant l'approche par fonction de tâche, nous définissons une relation entre les variations des paramètres de mouvement dans l'image et les déplacements de la caméra, qui est une équation différentielle non-linéaire. Nous proposons donc quelques idées pour la linéariser, et pouvoir alors commander la caméra en accélération ou en combinant des contraintes en accélération sur certains axes et en vitesse sur d'autres.

- Enfin, dans un dernier chapitre, nous développons l'étude de quelques tâches robotiques entrant dans le cadre de l'asservissement visuel dynamique, et nous présentons des résultats tant en simulation, que sur site robotique réel. Il s'agit tout d'abord de tâches de suivi d'objet mobile, dont la finalité est identique à celles détaillées au chapitre 2, mais où la consigne est spécifiée par la nécessité de conserver une vitesse apparente nulle. Ensuite, nous étudions une tâche de positionnement du plan de projection de la caméra parallèle à un objet planaire. Nous adjoignons à cette première tâche une contrainte de fixation, c'est-à-dire le fait, pour l'objet, de rester centré dans l'image. Enfin, nous présentons une tâche de suivi de trajectoire parallèlement à une surface plane. Cela consiste à contraindre la caméra à se déplacer parallèlement à ce plan, avec une vitesse apparente donnée.

Après avoir dressé un bilan de cette thèse, nous proposons, en conclusion, quelques perspectives de recherche pouvant être menées consécutivement à notre étude.

Chapitre 1

Etat de l'art

Dans ce chapitre, les diverses manières d'aborder le mouvement apparent dans une séquence d'images sont d'abord exposées. Cette partie ne se prétend pas exhaustive des nombreux travaux effectués sur la détection, la spécification ou l'estimation du mouvement. Le lecteur intéressé par de plus amples détails sur l'analyse du mouvement 2D pourra se reporter à l'étude bibliographique effectuée sur le sujet dans la thèse [Odobez 94]. Le but de ce paragraphe est simplement de donner les principales caractérisations possibles du mouvement, ainsi que quelques références qui s'y rattachent, et ce afin de situer le contexte dans lequel se sont effectuées les études d'asservissement visuel basées sur le mouvement apparent dans l'image. Nous présentons succintement dans la suite du chapitre, certaines idées exploitées dans ces diverses études, ainsi que quelques résultats les concernant. Ainsi, nous exposons dans une seconde partie des applications de suivi d'objets mobiles, tandis que la troisième partie concerne des tâches de positionnement par rapport à un objet donné. Enfin, la quatrième partie de ce chapitre se rapporte à des travaux relatifs à la navigation de robots mobiles, en particulier dans le cadre de l'évitement d'obstacles.

1.1 Caractérisation du mouvement dans l'image

1.1.1 Champ dense

La détermination d'un champ dense de mouvement entre deux images successives consiste à retrouver pour chaque pixel de la première image, le pixel de la seconde qui est la projection du même point de l'espace tridimensionnel observé par la caméra. On associe à chacun de ces pixels son vecteur de déplacement en abscisse et ordonnée, qui correspond à la différence des positions dans les deux images. Ce vecteur peut prendre des valeurs non-entières, c'est-à-dire qu'on peut associer à un pixel de coordonnées entières de la première image, un pixel "virtuel" qui soit composé de parties de pixels de coordonnées entières de la seconde image.

Un des articles fondateurs de ce domaine est celui d'Horn et Schunck, [Horn 81]. Ces auteurs définissent une approche locale de recherche du déplacement, en chacun des points de l'image, basée sur le principe de conservation de l'illumination. Ce principe repose sur l'idée que l'illumination émise par un point de l'espace tridimensionnel, et reçue par la caméra, reste inchangée entre deux instants proches. A un niveau d'illumination reçue en un pixel de la caméra correspond un niveau de gris de ce pixel sur l'image. Si l'on appelle I la fonction qui à un pixel (x, y) de l'image associe son niveau de gris, ce principe peut se traduire par l'équation suivante:

$$\frac{dI}{dt} = 0 \text{ c'est-à-dire } \frac{\partial I}{\partial x}\dot{x} + \frac{\partial I}{\partial y}\dot{y} + \frac{\partial I}{\partial t} = 0$$

où $\dot{x}$ et $\dot{y}$ sont, respectivement les composantes selon l'axe des abscisses et celui des ordonnées du vecteur de vitesse recherché, et $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, $\frac{\partial I}{\partial t}$ les gradients spatio-temporels de I.

La résolution du système formé de l'ensemble de ces équations, est en pratique difficile. Ainsi, lorsque le gradient spatial ($\partial I/\partial x$ ou $\partial I/\partial y$) est trop faible - dans une zone sans texture ou lors d'un déplacement le long d'un contour, par exemple - il est impossible de discriminer quelle est la réelle direction du mouvement. Il s'agit du problème d'ouverture. Un moyen de le résoudre, est de considérer le champ de mouvement comme continu, en se basant sur l'idée que deux points proches sur un même objet auront des déplacements apparents similaires. Cependant, cette contrainte de continuité est évidemment fausse sur les contours des objets puisqu'alors, deux points proches peuvent très bien appartenir à deux objets différents et donc avoir des mouvement également différents. Il faut alors introduire une contrainte de relaxation qui permet de faire apparaître les discontinuités du champ.

Il s'avère en définitive, que, de plus, le calcul d'un champ dense est très délicat et que la plupart des méthodes se base sur des algorithmes itératifs, donc lourds à mettre en œuvre. Le lecteur intéressé par ce domaine de l'analyse d'image pourra consulter, par exemple [Black 94, Mémin 98] pour des améliorations récentes de l'approche exposée brièvement ci-dessus, ou encore l'article [Barron 94] qui recense et fait une critique de la plupart des méthodes repertoriées.

En conclusion, le calcul d'un champ dense de mouvement est à l'heure actuelle d'une complexité trop grande et donc d'un coût en temps de calcul trop lourd pour pouvoir être utilisé dans une boucle robotique suffisamment réactive.

1.1.2 Modèle de mouvement

Lorsque la caméra observe un objet rigide, les déplacements apparents des pixels qui lui correspondent sont liés les uns aux autres par la contrainte géométrique 3D de la conservation des distances entre chacun des points de l'objet. En conséquence, le champ dense de déplacement de l'ensemble des pixels correspondant à un même objet peut être

très bien approximé par un modèle paramétrique. L'estimation du mouvement apparent dans l'image peut alors se restreindre au calcul de ces paramètres.

Ainsi, plusieurs travaux, parmi lesquels nous pouvons citer par exemple [Campani 90, Bergen 92, Odobez 95], se sont attachés à définir des algorithmes permettant de déterminer les paramètres d'un modèle du mouvement apparent d'une zone de l'image. Cette zone peut correspondre, par exemple à un objet donné, pour peu que l'image ait été préalablement segmentée, ou encore à l'ensemble de l'image, si l'on cherche à determiner l'influence du mouvement de la caméra par rapport à la scène observée. Dans chacune de ces études, le modèle considéré est polynomial. De même, elles se basent toutes les trois sur un schéma multi-résolution d'estimation, ce qui permet, en construisant des pyramides d'images sous-échantillonnées à partir des deux images du couple initial, d'appréhender des mouvements apparents relativement importants. La première de ces approches est cependant limitée par la quantité d'information nécessaire à l'estimation qui contraint à ne travailler que sur des zones de l'image de taille suffisante (au moins 70×70 pixels). Dans la seconde, une condition nécessaire pour une bonne estimation est que le mouvement apparent ne provienne que du mouvement propre de la caméra. L'algorithme d'estimation d'un modèle de mouvement, le "RMRm", présenté dans le troisième de ces articles est celui que nous avons utilisé pour nos expérimentations sur site réel. Le principal de ses avantages, par rapport au précédent, est qu'il comprend une phase de rejet des déplacements locaux incohérents avec le modèle global, ce qui en assure la robustesse, même dans le cas où le champ n'est pas parfaitement homogène. Il est présenté brièvement à la fin de cette section.

Calcul du modèle de mouvement apparent d'un objet rigide

Comme nous l'avons écrit précédemment, les contraintes de conservation de la distance pour chaque couple de points d'un objet rigide observé par une caméra, peuvent être utilisées pour le calcul du mouvement apparent dans l'image. Nous rappelons ici, en nous inspirant de plusieurs travaux effectués dans ce cadre, comme par exemple [Waxman 83, Aloimonos 88, Subarrao 89, Negahdaripour 92], l'expression de l'approximation polynomiale au second ordre du mouvement projeté sur l'image d'un objet rigide.

On se place sous l'hypothèse d'une caméra sténopée (cf. [Horaud 93]), schématisée sur la figure (1.1). On considère, sans perte de généralité, que la distance focale est égale à 1. Sur la figure (1.1), le repère caméra et ses éléments (centre et axes) sont repérés par l'indice c, et respectivement, le repère objet par l'indice o. L'axe optique de la caméra est confondu avec l'axe Z_c de son repère. On désigne par P le premier point de l'objet rencontré sur cet axe, et p sa projection sur le plan image. De même, on désigne par M un point quelconque de la surface de l'objet, qui se projette sur le plan image en m. Enfin, le mouvement 3D de l'objet au point P est repéré par la flèche noire, et cette vitesse induit un déplacement 2D dans l'image du point p, qui est repéré par la flèche grise.

Les mouvements relatifs d'un des repères en jeu par rapport à l'autre sont caractérisés

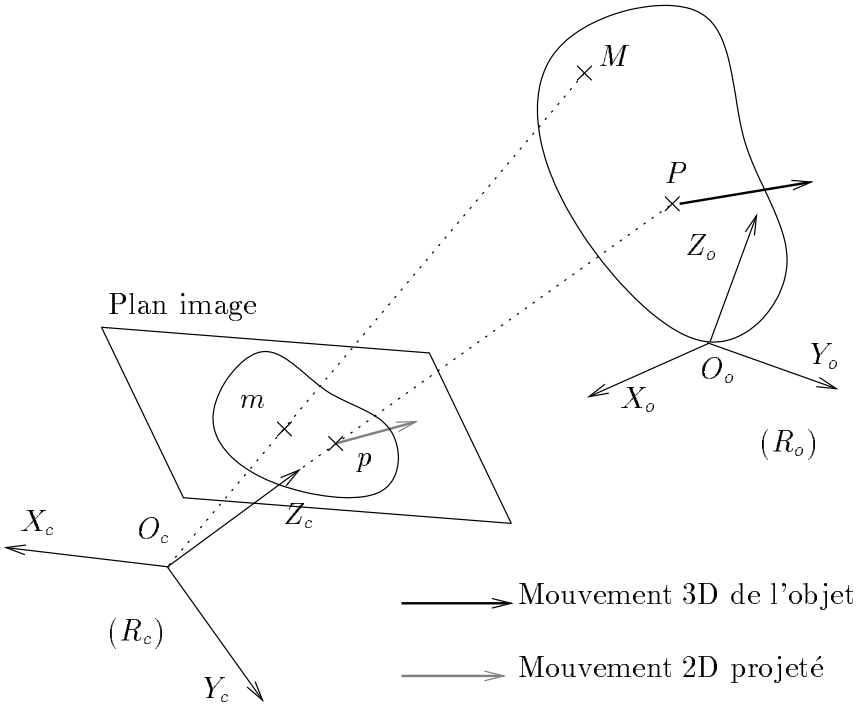


FIG. 1.1 – *Modèle de projection de la caméra. Positions relatives des repères caméra et objet.*

par un torseur cinématique que l'on exprime dans le repère de la caméra. Dans tous les calculs qui suivront, l'expression de la composante de translation d'un torseur sera donnée au centre optique de la caméra.

On note le torseur cinématique du mouvement de l'objet par rapport à la caméra:

$$v(R_o, R_c)(O_c) = \begin{pmatrix} T(O_c) \\ \Omega \end{pmatrix} = \begin{pmatrix} T_o(O_c) - T_c(O_c) \\ \Omega_o - \Omega_c \end{pmatrix}$$

où T et Ω représentent respectivement les vitesses de translation et de rotation de l'objet par rapport à la caméra. De même, T_o et T_c sont respectivement les vitesses de translation de l'objet et de la caméra par rapport à un repère fixe de l'espace dans lequel ils évoluent, auxquelles correspondent respectivement les vitesses de rotation Ω_o et Ω_c.

On désigne par $(X, Y, Z)^T$ les coordonnées du point M dans le repère de la caméra. A tout instant, ce point M est lié à l'objet. Sa vitesse relativement au repère caméra et exprimée dans ce repère est notée $T(M)$. D'après la relation fondamentale de la cinématique, cette contrainte physique se traduit par:

$$T(M) = T(O_c) + \Omega \wedge \overrightarrow{O_c M} \tag{1.1}$$

En notant $T = (T_x, T_y, T_z)^T$ et $\Omega = (\Omega_x, \Omega_y, \Omega_z)^T$ dans le repère (R_c), la relation (1.1 peut s'écrire sous forme scalaire :

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} + \begin{pmatrix} \Omega_y Z - \Omega_z Y \\ \Omega_z X - \Omega_x Z \\ \Omega_x Y - \Omega_y X \end{pmatrix} \tag{1.2}$$

Considérons maintenant la projection des points de l'objet sur le plan image de la caméra. En notant $(x, y)^T$ les coordonnées dans le plan image du point m, on obtient les relations:

$$x = \frac{X}{Z} \quad \text{et} \quad y = \frac{Y}{Z} \tag{1.3}$$

Par dérivation temporelle de ces équations et en remplaçant par les composantes de la vitesse du point M donnée en (1.2), on détermine la vitesse 2D du point m:

$$\begin{cases} \dot{x} = & \frac{1}{Z}[T_x + \Omega_y Z - \Omega_z Y - \frac{X}{Z}(T_z + \Omega_x Y - \Omega_y X)] \\ \dot{y} = & \frac{1}{Z}[T_y + \Omega_z X - \Omega_x Z - \frac{Y}{Z}(T_z + \Omega_x Y - \Omega_y X)] \end{cases} \tag{1.4}$$

Dans le repère (R_c), le point P a pour coordonnées $(0, 0, Z_p)^T$. Le développement en série de Taylor de la surface de l'objet, calculé au point M, s'écrit, toujours dans le repère (R_c):

$$Z = Z_p + \left(\frac{\partial Z}{\partial X}\right)_P X + \left(\frac{\partial Z}{\partial Y}\right)_P Y + \frac{1}{2}\left[\left(\frac{\partial^2 Z}{\partial X^2}\right)_P X^2 + \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_P XY + \left(\frac{\partial^2 Z}{\partial Y^2}\right)_P Y^2\right] + \ldots$$

que l'on choisit de noter:

$$Z = Z_p + \gamma_1 X + \gamma_2 Y + \zeta_1 X^2 + \zeta_2 XY + \zeta_3 Y^2 + \mathcal{O}(X^3, Y^3) \tag{1.5}$$

où $\mathcal{O}(X^3, Y^3)$ représente l'ensemble des termes d'ordre au moins égal à 3 dans ce polynome en X et Y.

En remplaçant Z par cette valeur dans le système (1.4), le développement en série de Taylor des composantes de la vitesse de m devient:

$$\begin{cases} \dot{x} &= \frac{T_x}{Z_p} + \Omega_y - \frac{x}{Z_p}(\gamma_1 T_x + T_z) - y\left(\frac{\gamma_2 T_x}{Z_p} + \Omega_z\right) \\ &+ x^2\left(\frac{\gamma_1 T_z}{Z_p} + \Omega_y - \zeta_1 T_x\right) + xy\left(\frac{\gamma_2 T_z}{Z_p} - \Omega_x - \zeta_2 T_x\right) - y^2(\zeta_3 T_x) + \mathcal{O}(x^3, y^3) \\ \dot{y} &= \frac{T_y}{Z_p} - \Omega_x - x(\frac{\gamma_1 T_y}{Z_p} - \Omega_z) - \frac{y}{Z_p}(\gamma_2 T_y + T_z) \\ &- x^2(\zeta_1 T_y) + xy\left(\frac{\gamma_1 T_z}{Z_p} + \Omega_y - \zeta_2 T_y\right) + y^2\left(\frac{\gamma_2 T_z}{Z_p} - \Omega_x - \zeta_2 T_y\right) + \mathcal{O}(x^3, y^3) \end{cases}$$

Dans certains cas, et en particulier lorsque l'on considère un point M proche du point P, la surface de l'objet peut être approximée par un plan d'équation $Z = Z_p + \gamma_1 X + \gamma_2 Y$.

Le mouvement apparent dans l'image est alors décrit par le modèle quadratique simplifié à huit paramètres suivant:

$$\left\{ \begin{array}{rcl} \dot{x} & = & c_1 + a_1 x + a_2 y + q_1 x^2 + q_2 xy \\ \dot{y} & = & c_2 + a_3 x + a_4 y + q_3 y^2 + q_4 xy \end{array} \right. \tag{1.6}$$

où les termes c_i, a_i et q_i représentent respectivement les coefficients constants, affines et quadratiques du modèle polynomial de mouvement de degré 2. Les valeurs de ces termes en fonction de la vitesse relative de l'objet par rapport à la caméra et des paramètres de l'équation de l'approximation planaire de l'objet autour du point P sont données par les relations:

$$\left\{ \begin{array}{ll} c_1 = v_x + \Omega_y & c_2 = v_y - \Omega_x \\ a_1 = -(\gamma_1 v_x + v_z) & a_2 = -(\gamma_2 v_x + \Omega_z) \\ a_3 = -\gamma_1 v_y + \Omega_z & a_4 = -(\gamma_2 v_y + v_z) \\ q_1 = \gamma_1 v_z + \Omega_y & q_2 = \gamma_2 v_z - \Omega_x \\ q_3 = q_2 & q_4 = q_1 \end{array} \right.$$

où le vecteur v désigne la translation de l'objet au facteur d'échelle près de la profondeur, c'est-à-dire $(v_x, v_y, v_z) = \dfrac{1}{Z_p} T$.

Il faut noter que, dans le cas où l'objet observé est réellement planaire, ce modèle de mouvement projeté est exact.

Dans la plupart des cas, l'estimation des paramètres du modèle de mouvement se limite à celle des paramètres affines. Cela se justifie par le fait que les paramètres purement quadratiques sont en général très bruités.

Autre expression du modèle affine

Dans les articles [Bouthemy 93, Cipolla 97a], une autre formulation du modèle affine est proposée, en utilisant des descripteurs du mouvement. Cela fait apparaître le modèle affine comme la somme de mouvements particuliers dans l'image, telle, par exemple, la divergence produite par un mouvement de zoom de la caméra. Ce modèle est donné par:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} tr_x \\ tr_y \end{pmatrix} + \left[div \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + rot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} + hyp1 \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + hyp2 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right] \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.7}$$

Par unicité du développement affine du mouvement, on peut établir les liens entre ces descripteurs et les paramètres de l'équation (1.6):

$$\left\{ \begin{array}{rclrcl} tr_x & = & c_1 & tr_y & = & c_2 \\ div & = & \dfrac{a_1 + a_4}{2} & rot & = & \dfrac{a_3 - a_2}{2} \\ hyp1 & = & \dfrac{a_1 - a_4}{2} & hyp2 & = & \dfrac{a_2 + a_3}{2} \end{array} \right.$$

Chacun de ces descripteurs correspond à une déformation particulière de l'image:

- tr_x et tr_y, respectivement, à une translation pure de l'image le long des axes x et y
- div, à une expansion ou une contraction de l'image autour de son centre
- rot, à une rotation de l'image autour de son centre
- $hyp1$ et $hyp2$, à des déformations dites "hyperboliques" correspondant à une expansion le long d'un axe et d'une contraction le long d'un second; il s'agit des axes $\overrightarrow{x}$ et $\overrightarrow{y}$ de l'image pour le terme $hyp1$ et des bissectrices de ces axes pour $hyp2$

On donne un exemple de champ affine complexe sur la figure (1.2) a., ainsi que l'ensemble des champs correspondant à sa décomposition selon ses descripteurs: la composante constante (b), la divergence (c), la rotation (d) et les deux champs hyperboliques (e et f).

Un modèle affine simplifié à quatre paramètres est parfois utilisé pour décrire le mouvement apparent. Il correspond à l'approximation du modèle précédent dans lequel on considère les termes hyperboliques comme nuls. Ces deux paramètres étant fonction de l'orientation, dans le repère de la caméra, du plan tangent à l'objet au point P, le modèle simplifié est d'autant plus juste que ce plan est parallèle au plan image.

L'algorithme RMRm

Pour estimer les paramètres du modèle du mouvement, nous avons utilisé l'algorithme RMRm [Odobez 95]. Nous le résumons à présent, dans ses grandes lignes.

Considérons deux images successivement acquises par la caméra. La première étape de l'algorithme RMRm consiste à construire deux pyramides multi-résolutions à partir de chacune de ces images, où chaque niveau (hormis, le niveau 0, qui correspond à l'une des images) est obtenu par un filtrage gaussien de l'image au niveau précédent. La taille de l'image du niveau i correspond à une division par deux, dans les deux directions, de la taille de l'image au niveau $i-1$.

Soit Θ_t le vecteur des huit paramètres du modèle quadratique de mouvement à l'instant t. Au niveau de résolution la plus grossière de la pyramide, la première estimation de ce vecteur correspond à la minimisation, par rapport à Θ_t du critère suivant :

$$C(\Theta_t) = \sum_p \rho(r(p, \Theta_t)) \tag{1.8}$$

avec :

$$r(p, \Theta_t) = \nabla I(p,t).w_{\Theta_t}(p) + I_t(p,t)$$

qui est l'Equation de Contrainte du Mouvement Apparent (ECMA) classique, où :

- les points p sont tous les points de l'image ;

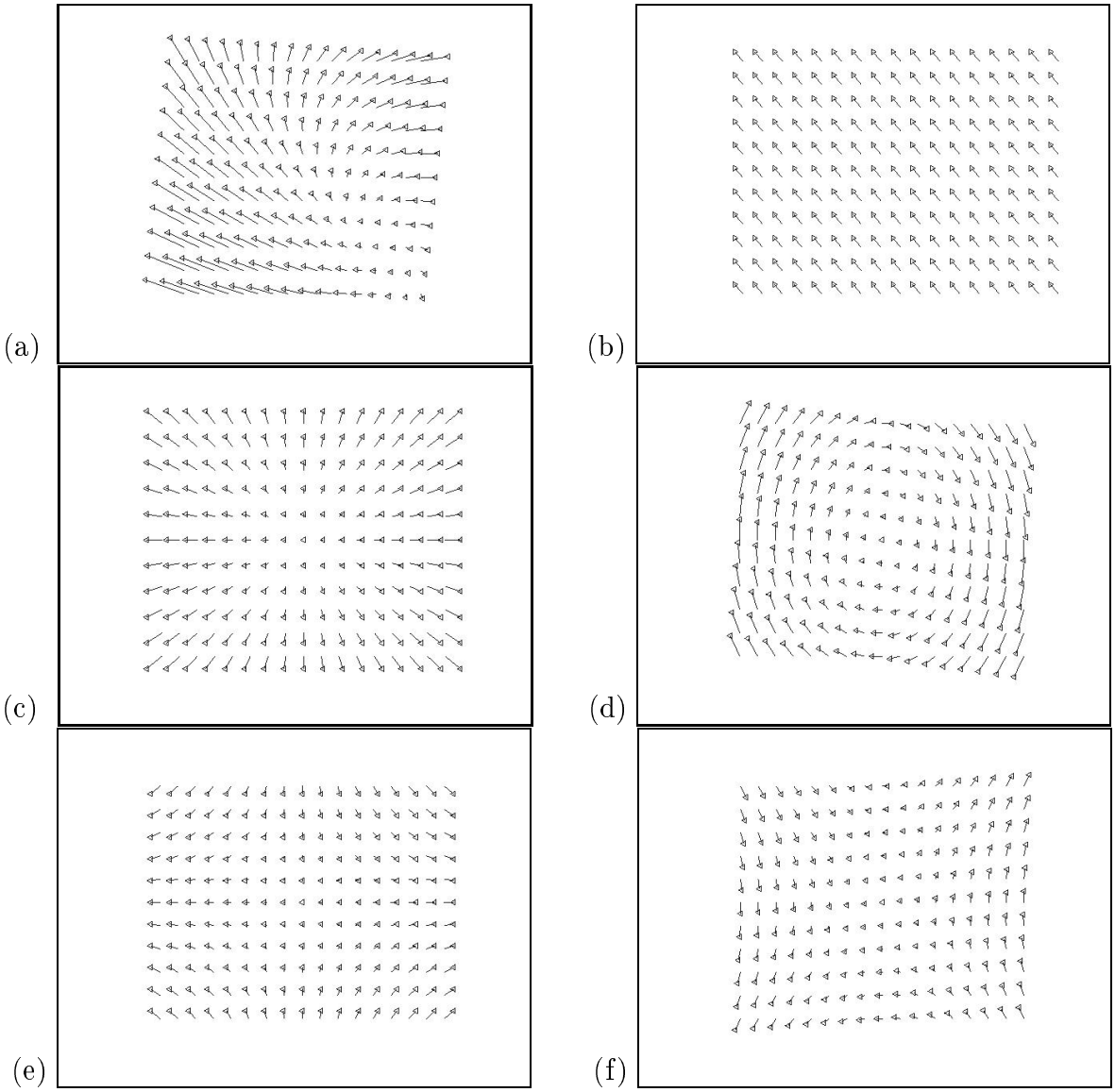


FIG. 1.2 – *(a) Un champ affine de mouvement et sa décomposition en la somme d'un déplacement constant (b), d'une divergence (c), d'une rotation (d) et de deux champs hyperboliques (e et f)*

- I est la fonction d'intensité lumineuse, qui à chaque pixel associe sa valeur en niveau de gris ;
- ∇I et I_t sont, respectivement, le gradient spatial et la dérivée temporelle de I ;
- $w_{\Theta_t}(p)$ est le vecteur vitesse calculé au point p, en appliquant le modèle défini par la valeur d'estimation courante de Θ_t ;
- ρ est un estimateur robuste, et plus précisément la fonction bi-pondérée de Tukey. Cet estimateur permet, au cours du processus itératif, de pondérer l'adéquation du déplacement du point p avec l'estimation courante du modèle de mouvement Θ_t. Plus cette adéquation est faible plus le poids est proche de 0. Cette estimateur permet donc de rejeter les points dont le mouvement est aberrant par rapport au mouvement dominant dans l'image, et par conséquent de robustifier l'estimation.

Remarque : L'ensemble des points p considérés peut être un sous-ensemble de l'image qui est spécifié au départ par l'utilisateur.

Le processus d'estimation des paramètres du mouvement suit alors une stratégie incrémentale. Soit Θ_t^k, l'estimée de Θ_t à l'itération k. Nous pouvons noter :

$$\Theta_t^k = \Theta_t + \Delta\Theta_t^k$$

où $\Delta\Theta_t^k$ représente le raffinement donné par l'estimation à l'itération k. Cette valeur est donnée par la relation :

$$\Delta\Theta_t^k = \arg\min_{\Delta\Theta_t^k} \sum_p \rho(r'(\Delta\Theta_t^k))$$

où :

$$r'(\Delta\Theta_t^k) = \nabla I(p + w_{\widehat{\Theta}_t^k}(p), t+1).w_{\Delta\Theta_t^k}(p) + I(p + w_{\widehat{\Theta}_t^k}(p), t+1) - I(p,t)$$

En notant, pour chacune des variables en jeu, $\widehat{x}$ l'estimation de x, nous obtenons alors :

$$\widehat{\Theta}_t^k = \widehat{\Theta}_t + \widehat{\Delta\Theta}_t^k$$

Ce processus est itéré, et les incréments sont cumulés jusqu'à ce qu'un critère de convergence prédéfini soit atteint.

L'estimation au niveau de résolution i (plus fin) est initialisée par la valeur obtenue au niveau $i+1$ (plus grossier). Ce schéma utilise un processus d'estimation aux moindres carrés itératif repondéré. Il ne fait appel qu'au calcul des dérivées spatio-temporelles de la fonction d'intensité.

1.1.3 Carte binaire du mouvement

La carte binaire du mouvement apparent dans l'image désigne une fonction qui, à chaque pixel de l'image, associe une valeur binaire, selon que ce pixel correspond à un point 3D qui a été ou non en mouvement par rapport à l'image précédente. Cette opération porte le nom de détection d'un (ou des) objet(s) mobile(s).

Avec une caméra fixe

Dans le cas où la caméra est fixe, la résolution de ce problème est simple. Si l'on considère des conditions idéales, c'est-à-dire sans bruit dans l'image, un pixel correspondant à un point immobile conservera la même intensité lumineuse au cours du temps. Ainsi, il suffit de déterminer si le niveau de gris d'un pixel a changé pour savoir s'il correspond à un point mobile. En pratique, les erreurs dues au bruit peuvent être compensées en n'effectuant pas la comparaison directement entre pixels, mais en seuillant la différence entre les moyennes obtenues sur un voisinage de ces pixels. Cette méthode très grossière donne une idée approximative de la position des zones mobiles, et elle peut être complétée, si besoin est, par des tests statistiques locaux, comme dans l'étude [Hsu 84, Bouthemy 90].

Avec une caméra mobile

Le véritable problème de la construction de cette carte binaire du mouvement se pose dans le cas où la caméra est elle-même mobile. Alors, en appliquant les méthodes décrites ci-dessus, tous les points en mouvement relatif par rapport à la caméra sont détectés mobiles, et en particulier tous les points statiques dans un repère fixe de l'espace d'évolution de la caméra. Il convient alors de discriminer le mouvement dans l'image dû à celui de la caméra de celui dû à des objets réellement mobiles. Les études réalisées sur ce sujet, parmi lesquelles nous pouvons citer [Irani 92, Odobez 97, Nagel 94], se basent généralement sur le fait que les éventuels objets mobiles représentent une partie minoritaire de la scène observée par rapport au fond immobile. Dans les deux premières, le principe est de déterminer le mouvement dominant dans l'image, qui correspond à la projection du mouvement de la caméra, puis de compenser la seconde image par l'opposé de ce mouvement, de façon à simuler une absence de déplacement de la caméra. L'image obtenue peut alors être comparée à la première par les méthodes exposées dans le cas d'une caméra fixe. Dans la troisième de ces études, l'approche utilisée consiste à déterminer les frontières des zones mobiles par l'emploi d'un test statistique local sur le champ dense du mouvement. Ayant ces frontières, les zones connexes peuvent être assimilées à des objets mobiles, et les autres au fond immobile.

Ce type de méthodes s'apparente plus à de la segmentation au sens du mouvement, et la détection des objets mobiles est plutôt une fonctionnalité alors offerte. De tels algorithmes de segmentation visent à regrouper les pixels de l'image en zones de mouvement apparent

cohérent. Cela offre en particulier, la possibilité de séparer des objets mobiles du fond de la scène. De plus, la segmentation permet d'affecter une étiquette à chaque zone afin de déteminer si deux pixels détectés en mouvement appartiennent à un même objet ou non. Ces algorithmes ne fonctionnent malheureusment pas en temps réel. Nous ne les utiliserons donc pas dans notre étude, même s'ils nous auraient certainement permis d'améliorer notre approche en nous permettant de distinguer clairement l'objet d'intérêt (par exemple, lors de la tâche de positionnement parallèlement à un plan que nous étudions au chapitre 4). Le lecteur intéressé par ce domaine d'étude pourra se reporter par exemple aux articles [Wang 94, Mitiche 96, Stiller 97].

1.2 La poursuite de cibles

L'application de poursuite d'un objet mobile est l'une de celles qui a drainée le plus de travaux dans le domaine de l'utilisation d'informations visuelles dynamiques dans des boucles de commande robotiques. A l'heure actuelle, les recherches menées dans cette optique semblent s'être concentrées sur un suivi de type surveillance. Le principe est de commander une tête articulée de telle manière qu'un objet d'intérêt soit toujours observé par la caméra, ou au moins tant qu'il n'est pas caché par un second objet. Par tête articulée, on désigne abusivement ici tous les systèmes, qu'ils soient monoculaires ou binoculaires, disposant de deux degrés de liberté de rotation en panoramique et inclinaison (*pan and tilt*). Les systèmes binoculaires disposent généralement d'au moins un degré de liberté supplémentaire, à savoir l'orientation relative des deux caméras qui les composent. Nous ne nous intéresserons pas dans ce paragraphe aux mouvements spécifiques de telle ou telle tête articulée, mais uniquement aux rotations des deux degrés de liberté qui permettent de maintenir un objet d'intérêt dans le champ de vision. D'autant plus que parfois, la tâche imposée à cette tête articulée consiste à faire suivre l'objet par une des caméras et à asservir la seconde sur les mouvements de la première, de manière à maintenir la vergence, c'est-à-dire à assurer que les deux champs de vue soient centrés sur le même point. C'est le cas par exemple dans l'article [Capurro 97], où le but premier est de maintenir cette vergence, même lorsque l'objet est mobile. En fait, les seuls mouvements de l'objet qui peuvent être correctement appréhendés sont des translations perpendiculairement à l'axe joignant les deux caméras. Le seul paramètre du mouvement à estimer est donc celui de divergence. Signalons que dans ce cas particulier, l'utilisation de caméras log-polaires permet de robustifier à la fois la fusion des zones communes aux deux images et le calcul de la composante de divergence du mouvement apparent dans l'image.

Une autre application de poursuite d'objets mobiles, grâce à des têtes stéréo, consiste à déterminer la position tridimensionnelle de l'objet puis à effectuer un contrôle simultané des deux caméras. Ces travaux, comme par exemple [Vieville 95] ne font pas explicitement appel au mouvement dans l'image. Nous les aborderons un peu plus en détail au chapitre

suivant.

Dans ce cadre, la partie de commande du suivi en *pan* et *tilt* d'un objet mobile peut être considérée comme relativement simple puisqu'elle consiste à ne contrôler que deux degrés de liberté de manière à conserver un point particulier de l'objet (son centre par exemple), ou l'ensemble de l'objet, visible. La principale difficulté, lorsque les objets considérés sont "réels", revient à localiser leur projection dans l'image. Dans le paragraphe suivant, nous donnons quelques-unes des techniques rencontrées dans la littérature pour déterminer la projection de l'objet mobile. Pour la plupart des approches citées, nous ne détaillerons pas la commande en rotation qui permet d'assurer le suivi. Nous nous concentrerons seulement sur l'algorithme de vision qui fournit l'information utilisée en entrée de cette boucle de contrôle. Simplement, dans certains cas, la phase de détection étant nettement détachée de celle du suivi, nous expliquerons alors brièvement comment ce suivi est effectué.

Localisation de l'objet mobile

La localisation de l'objet mobile peut s'effectuer selon deux objectifs différents. Dans un cas, il s'agit simplement de détecter l'objet mobile à l'initialisation, tandis que dans l'autre, la tâche de poursuite nécessite la détermination de la position de l'objet dans chaque image, ce qui revient à le suivre tout au long de la séquence acquise.

Lorsque la caméra est fixe

Nous avons noté dans le paragraphe précédent que la détection d'un objet mobile est quasi évidente, pour peu que la caméra soit immobile. Ce principe est utilisé dans [DawsonHowe 96] qui effectue une simple différence d'images pour une application de surveillance de piétons depuis la fenêtre d'un bâtiment. Cependant, afin d'observer un angle large, la détection ne se fait que sur une partie de l'image, par différence avec le fond appris au début de l'application. Et, lorsque le piéton suivi sort de la zone considérée, la détection s'effectue dans la zone contigüe. Une approche semblable est définie dans [Yamane 98]. Cette étude, uniquement axée sur le suivi de personne, consiste à déterminer une fenêtre de l'image dans laquelle se trouve un piéton. Le principe n'est plus d'effectuer une différence avec le fond supposé immobile, mais de combiner des connaissances a priori sur la nature d'un piéton (assimilé plus ou moins à un rectangle vertical), des informations en niveau de gris (la texture des vêtements est supposée presque uniforme) et des informations de mouvement (un champ global de mouvement est calculé, et une segmentation visant à regrouper les zones de mouvement cohérent est effectuée). La superposition des zones détectées, tant au sens du mouvement qu'à celui de l'uniformité du niveau de gris, permet de repérer le piéton. Mais, comme dans l'étude précédente, l'analyse se fait avec une caméra fixe. Un mouvement de rotation panoramique n'est généré que lorsque le piéton sort du champ de vue, qui plus est à une vitesse de rotation constante prédéfinie et

donc indépendante de celle de la personne suivie.

Quelques approches utilisent le mouvement apparent de l'objet pour en assurer le suivi, mais sans s'intéresser à sa détection; c'est le cas par exemple de [Grosso 93, Araújo 96, Batista 97]. Dans ces trois articles, le suivi est réalisé en mesurant à chaque itération le mouvement de l'objet d'intérêt, ramené au préalable dans la zone fovéale, et en maintenant sa vitesse apparente nulle. Dans le premier, la cible est simplement désignée par l'utilisateur. Dans les deux derniers, la détection n'est faite qu'à l'initialisation et s'effectue par différence d'images successives, avec une caméra qui est, pendant cette phase, immobile.

En utilisant le mouvement connu de la caméra

Les études sur le suivi d'objet réels ne se sont pas cantonnées au cas d'une caméra fixe. Ainsi, une part importante des travaux utilisant le mouvement apparent d'un objet afin de le suivre par une cellule pan et tilt, se base sur le principe que ce mouvement projeté propre de l'objet est incohérent avec celui du fond sur lequel se déplace l'objet. Si l'on se réfère à l'expression du mouvement apparent donnée en (1.6), il apparaît que l'influence d'une rotation de la caméra ne dépend pas de la profondeur de la scène observée. Or, dans le cas du suivi par mouvements panoramique et d'inclinaison, seules des rotations sont mises en œuvre, si le centre optique est confondu avec le centre de la rotation. Donc, le mouvement apparent du fond immobile peut être déterminé a priori, en connaissant les vitesses angulaires de la caméra, mesurées par odométrie. Cette technique est utilisée, par exemple, dans les articles [Murray 94, Murray 95, Arias 98]. La détection de l'objet mobile se fait entre deux images successives, en effectuant une transformation affine de la seconde image par l'opposé du modèle a priori, puis en appliquant les méthodes de différences pixel à pixel entre cette nouvelle image et la première. Une des principales limites de cette méthode est qu'elle nécessite une mesure précise du mouvement de la caméra, afin d'obtenir un modèle a priori du mouvement du fond fiable. De plus la condition de superposition du centre otique et du centre de rotation est, en pratique, délicate à assurer. Cela génère une erreur du modèle de mouvement, puisqu'il est alors impossible de garantir que ce mouvement est uniquement composé d'une rotation. La phase de suivi est par contre différente dans chacune de ces approches. Dans [Murray 94], la détection est effectuée entre chaque couple d'images. De plus, pour robustifier cette technique aux erreurs d'odométrie, l'image binaire de différence est traitée par des outils morphologiques d'érosion et de dilatation. L'hypothèse sous-jacente à ce traitement, est que les fausses alarmes correspondent à des contours, donc d'épaisseur faible, alors que le masque binaire de l'objet d'intérêt est plus large. Dans [Murray 95], la technique décrite ci-dessus n'est utilisée que lors de la phase de détection de l'objet mobile. Une fois détecté, son mouvement dans l'image est calculé à chaque itération et le principe de la commande en vitesse angulaire consiste à maintenir cette vitesse apparente nulle.

Une approche similaire est utilisée dans [Daniilidis 98]. Là encore, le mouvement du fond est déterminé par le modèle a priori correspondant aux vitesses de rotation mesurées. Par contre, l'algorithme de détection n'effectue pas la différence entre la première image et une compensation de la seconde par le mouvement opposé. Le principe est de calculer un champ dense de mouvement entre les deux images, ou au moins la composante normale au gradient spatial du déplacement de chaque pixel, lorsque cela est possible. Il suffit ensuite de comparer ce champ avec celui a priori, et de conserver les zones où il y a incohérence entre les deux. Cette méthode est robustifiée en ne considérant que les points de fort gradient spatial.

En déterminant la composante du mouvement apparent due au déplacement de la caméra

Une idée assez proche, mais qui évite le problème de devoir mesurer finement les mouvements de la caméra, consiste à calculer directement le modèle du mouvement du fond de la scène. Il est alors possible de suivre le schéma présenté ci-dessus, qui consiste à compenser la seconde image du mouvement estimé, et à effectuer la différence avec la première image. Cette approche est, par exemple, utilisée dans les articles [Bartholomeus 93, Nordlund 96]. La principale limite de cette approche est que le mouvement global de l'image est considéré comme étant celui induit par les déplacements de la caméra. Cela n'est vrai que lorsque la majeure partie de l'image correspond au fond de la scène. Une des contraintes est donc que la projection de l'objet mobile occupe une part minoritaire de l'image. Une autre contrainte, qui est d'ailleurs vraie pour presque toutes les techniques de détection présentées auparavant, est qu'en cas d'objets mobiles multiples, aucune distinction n'est faite entre les diverses zones déterminées comme étant mobiles.

Une robustification de cette approche est proposée dans [Eklundh 95]. Tout d'abord, le calcul du champ de mouvement du fond ne tient pas compte des pixels de l'image correspondant au masque de l'objet, avec l'hypothèse que le déplacement de l'objet d'intérêt entre deux itérations est limité. Cela permet donc de suivre un objet d'une taille plus importante. Ensuite, le mouvement de chaque zone connexe est calculé, et la fusion éventuelle de plusieurs de ces zones s'effectue selon un critère de cohérence de mouvement. Cela permet, d'une part, de différencier l'objet d'intérêt d'autres objets en mouvement, et d'autre part, de résoudre les problèmes d'occultations partielles.

1.3 Navigation de robots mobiles et évitement d'obstacles

Un robot mobile, à savoir un robot susceptible de se déplacer dans un espace non restreint, à la différence d'un robot manipulateur, doit, dans de nombreuses applications, se mouvoir dans un environnement peu ou mal connu (dans des tâches d'exploration par

exemple). Cela pose évidemment des problèmes de sûreté de l'engin. Même dans le cas où l'environnement est parfaitement connu, les problèmes mécaniques, du type glissement d'une roue ou d'une patte sur le sol, rendent très imprécises les mesures odométriques qui pourraient permettre de le situer dans son espace d'évolution. Afin d'éviter une collision, le système doit être à même d'appréhender au moins localement cet environnement.

Une solution couramment répandue consiste à déterminer la présence d'obstacles éventuels à une distance donnée du robot, en particulier dans la direction de déplacement, par des techniques de type radar ou détecteurs de proximité. Ces derniers fournissent une information assez pauvre, en ce sens qu'elle est généralement du type binaire: "il y a un obstacle dans ma direction de déplacement" ou "il n'y a pas d'obstacle dans ma direction de déplacement". La stratégie associée consiste alors simplement à ne rien faire, c'est-à-dire à continuer le mouvement à la même vitesse et dans la même direction si aucun obstacle n'est détecté ou au contraire à changer le cap, généralement d'un angle prédéfini. La trajectoire qui en résulte est donc relativement aléatoire et il est difficile de remplir une tâche aussi simple que "se rendre d'un point A donné à un point B donné".

L'apport de la vision est, bien entendu, de pouvoir enrichir l'information acquise sur le milieu environnant. Une première approche consiste à reconstruire au fur et à mesure du déplacement l'espace d'évolution du véhicule. Les inconvénients sont nombreux. Outre que cela nécessite une bonne calibration de la caméra et du robot, la question d'une détermination précise de la position du robot dans cet environnement, ou plus simplement de ses déplacements effectués, se pose en raison des problèmes mécaniques précédemment évoqués. Ces approches ne sont généralement valables que dans des cas où l'environnement est suffisamment structuré - un ensemble de polyèdres, par exemple -, et que l'on possède un a priori sur cette structure.

Par exemple, dans l'étude [Vassallo 98], un robot mobile se déplace uniquement dans un couloir. Sachant que l'environnement peut être décrit comme un ensemble formé de quatre plans (le sol, les deux murs latéraux et le fond), l'algorithme de vision extrait quatre droites significatives: les intersections du sol avec les deux murs latéraux, et celles du fond avec ces mêmes murs latéraux. Le principe utilisé pour la navigation est de retrouver la direction dans laquelle pointe le robot, assimilée à l'intersection des deux premières droites, et de la contrôler afin qu'elle apparaisse au milieu des deux dernières, et ce sans tenir compte d'éventuels obstacles.

Le temps à collision

Une information importante, lorsqu'un robot mobile se déplace dans un environnement inconnu, n'est pas tant la distance à laquelle se trouve un obstacle placé sur sa trajectoire, mais plutôt le temps qui se passera avant une collision avec cet obstacle, s'il continue son déplacement à la même vitesse. Ce temps est appelé le temps à collision présenté, par exemple dans [Meyer 92]. L'intérêt d'utiliser le mouvement dans l'image pour assister la

trajectographie d'un robot mobile est que cette information de temps à collision y apparaît implicitement. Dans l'article [Cipolla 97a] par exemple, les auteurs montrent que cette valeur peut être simplement obtenue, pour peu que l'on puisse déterminer la part due à la translation le long de l'axe optique dans le champ de mouvement apparent. Plus précisément, le temps à collision, que l'on notera par la suite t_c vaut:

$$t_c = \frac{Z_p}{T_z} \tag{1.9}$$

où Z_p est la profondeur le long de l'axe optique, et T_z, la vitesse de translation le long de ce même axe. On s'aperçoit ici que si l'on est en mesure d'obtenir de façon fiable la vitesse T_z, il est alors possible de remonter à l'information de profondeur.

Dans ce même article, les auteurs montrent également que lorsqu'une surface plane est observée, l'analyse du champ du mouvement apparent permet d'en déterminer l'orientation. Ces idées sont exploitées, par exemple, dans l'article [Dev 94]. En supposant donnée une carte de segmentation de l'image, chaque zone correspondant à une surface plane observée, un modèle affine de mouvement est estimé sur chacune de ces zones, à partir d'une mesure du champ dense. Le temps à collision de chacune de ces zones est calculé à partir de la composante de divergence du modèle affine. Par un seuillage sur ces valeurs, le système détermine les zones correspondant à un risque de collision si la vitesse de déplacement du robot reste inchangée.

[Grosso 95] se base sur la même information mais utilise un système de caméras stéréo afin de robustifier le calcul du temps à collision. L'objectif est de construire une carte des profondeurs relatives dans la zone de l'espace qui fait face au robot, toujours afin de discriminer les directions dangereuses des autres.

Détection d'obstacles frontaux en recherchant les incohérences du champ de mouvement

Tout en ne souhaitant pas reconstruire explicitement la scène tridimensionnelle d'évolution du robot, au fur et à mesure que celui-ci se déplace, plusieurs études ont été menées afin de déterminer à chaque instant la carte bidimensionnelle des directions à éviter ou non. [Nelson 89], par exemple, cherche à déterminer ce que les auteurs appellent les zones de divergences directionnelles, c'est-à-dire les zones de rupture du champ de mouvement. Une étiquette est ensuite associée à l'intérieur de chacune de ces zones selon qu'elle est jugée potentiellement dangereuse. Leur approche a été testée avec un système observant un plan devant lequel se trouve un objet volumique. Lorsque la caméra est soumise à un mouvement de rotation, que ce soit autour de l'axe optique, ou orthogonalement à cet axe, il n'apparaît pas de zones dangereuses. Lors d'un mouvement de zoom avant, toute la zone proéminente est désignée à risque. Mais leur approche semble poser quelques problèmes lors de translation parallèlement au plan du fond, puisque dans ce cas, les deux cotés de la protubérance sont jugés dangereux, alors même qu'un de ces côtés, au moins, s'éloigne de la caméra.

Des méthodes plus robustes ont été développées depuis. Dans [Young 92], le système étudié est sensé se déplacer dans un environnement extérieur. La ligne d'horizon est déterminée par des critères géométriques (un contour de fort gradient, rectiligne, plus ou moins horizontal et qui va de part et d'autre de l'image). A chaque itération, un champ dense de mouvement est calculé, et une nouvelle ligne d'horizon est déterminée selon un critère de mouvement (le mouvement apparent du ciel est nul). Les ruptures dans cette ligne permettent de caractériser un obstacle proéminent ou un trou, selon que la déviation se fait vers le haut ou vers le bas. Dans [Giachetti 98], le principe utilisé est de calculer un champ dense de mouvement, rendu possible dans ce cas sous la contrainte de faibles déplacements dans l'image (moins de 5 pixels). Cela permet, d'une part, de déterminer le mouvement tridimensionnel de la caméra, et d'autre part, de détecter les éventuels obstacles par la recherche de zones de l'image ayant un mouvement incohérent avec le mouvement global. Une approche similaire est proposée dans [SantosVictor 96], où un champ dense du mouvement est une nouvelle fois calculé. A partir de ce champ, un modèle affine correspondant au mouvement apparent du sol (supposé planaire) est déterminé, et une incohérence entre le champ brut et son modèle sera assimilée à la projection d'un obstacle. Enfin, dans [Dev 98], un champ dense du mouvement est également calculé à chaque itération et conservé uniquement sur les zones de fort gradient. Ce champ est comparé avec un champ affine défini au préalable, correspondant à des seuils de risque de collision. Le champ de référence est fonction de la vitesse du robot et est établi en utilisant l'idée que, par exemple, un mouvement apparent de forte amplitude au centre de l'image est signe d'un danger plus grand qu'un mouvement de même amplitude à la périphérie de l'image. En définitive, cela permet de déterminer une carte binaire dans l'image, précisant si un point sera laissé à gauche ou à droite de la trajectoire. L'idée générale exploitée par ses trois articles est présentée sur la figure (1.3). On y présente tout d'abord la scène tridimensionnelle (a) dans lequel le robot évolue et telle qu'elle est observée par la caméra, puis le champ de mouvement apparent dans l'image (b), sous l'hypothèse d'une translation le long de l'axe optique, et enfin, la carte binaire des zones présentant un danger (c). Cette carte est obtenue à partir de la segmentation du champ dense de mouvement et en déterminant sur chacune des zones ainsi construite s'il s'agit d'un obstacle potentiel, indiqué alors en noir, à partir de l'amplitude des vecteurs de déplacement.

Navigation dans un couloir

Une part importante des travaux effectués sur la navigation de robots mobiles concerne le déplacement dans des environnements structurés d'intérieur de bâtiment, et en particulier, le long de corridors. Dans la plupart des cas, la tâche que doit remplir le système est de se déplacer parallèlement aux deux murs verticaux qui forment le couloir. Dans ce cadre, l'approche proposée dans [Dev 97] est de contraindre le robot de manière à obtenir deux champs symétriques sur deux zones verticales de part et d'autre de l'image, permettant de contrôler à la fois l'orientation et la distance relative à chacun des murs.

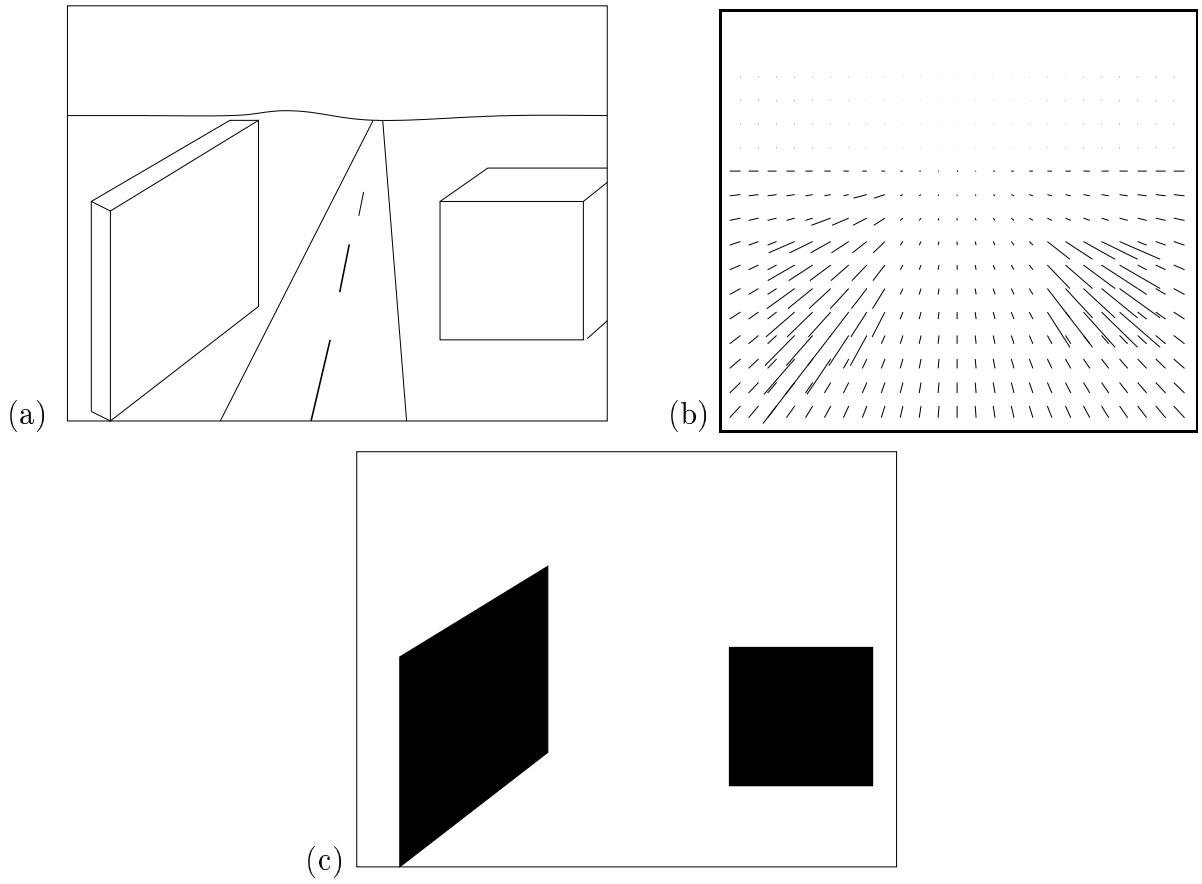


FIG. 1.3 – *Détection d'obstacles potentiels : scène observée (a), champ dense du mouvement projeté (b), carte binaire des obstacles (c).*

Dans un même but, plusieurs auteurs ont cherché à s'inspirer de modèles vivants, et en particulier du vol des insectes. Ainsi, dans l'article [Duchon 98], les auteurs commencent par démontrer qu'une mouche se déplace entre deux murs en fonction du mouvement 2D qu'elle perçoit sur chacun de ces murs. Par exemple, elle volera bien au centre si les deux murs sont immobiles et aura, au contraire, tendance à se rapprocher d'un mur se déplaçant dans la même direction qu'elle, l'autre restant fixe. En suivant cet exemple, les auteurs ont muni un robot mobile d'une caméra à grand angle lui permettant d'observer simultanément les deux murs. La commande utilisée est similaire à celle présentée précédemment, avec une prise en compte des obstacles latéraux qui sont traités comme un rétrécissement du couloir. [Coombs 93] et [SantosVictor 95] exploitent la même idée, à la différence que, dans le second de ces articles, le système utilisé consiste en un robot mobile à l'avant duquel deux caméras sont pointées dans des directions latérales. Donc, lorsque l'engin se déplace dans le couloir, chacune de ces caméras fait face à l'un des murs.

Une des limites de cette approche est que les divers systèmes présentés ne tiennent

pas compte des obstacles frontaux. C'est pourquoi, dans [SantosVictor 97], les auteurs ont complété leur système par une caméra frontale qui permet de gèrer les éventuels problèmes de collision de la même manière que ce qu'ils proposaient dans [SantosVictor 96]. De même, dans [Coombs 98], le système avec une caméra grand angle est complété par une caméra d'angle plus faible, qui permet de traiter également les risques de collision. Enfin, [Toepfer 98] utilise la même idée mais dans un cas monoculaire, en étudiant séparément le mouvement apparent à la périphérie de l'image et dans la zone fovéale.

Utilisation du foyer d'expansion pour contrôler le cap

Il existe une autre information importante pour la navigation; il s'agit du foyer d'expansion ("focus of expension" ou FOE en anglais), qui correspond à la projection du point à l'infini vers lequel un engin mobile se dirige à un instant donné. La position de ce point particulier est également implicitement présente dans le mouvement apparent dans l'image. Dans le cas où le mouvement de la caméra est une pure translation, tous les vecteurs de déplacement instantané dans l'image proviennent (ou se dirigent, si le mouvement le long de l'axe optique est en sens inverse de cet axe) d'un seul et même point, le FOE. Dans le cas, plus général, où le mouvement de la caméra est aussi composé de rotations, il convient de séparer l'influence des deux parties du mouvement, afin de ne travailler que sur le champ dû à la translation. On présente sur la figure (1.4), deux champs affines de mouvement sur lesquels le point correspondant au FOE est représenté par une croix. Sur le champ de gauche, le mouvement apparent est la conséquence d'un mouvement purement translationnel de la caméra et le FOE apparaît donc au point où le champ s'annule. Le mouvement apparent du champ de gauche a été obtenu en ajoutant une rotation au mouvement de translation précédent. La translation étant inchangée, le FOE est resté à la même place, mais, dans ce cas, le champ ne s'annule plus en ce point particulier.

Dans l'article [Fermüller 98], la tâche traitée consiste à se déplacer vers un point donné de l'espace. On suppose que ce point peut toujours être retrouvé dans l'image. Le schéma suivi consiste à diriger la caméra vers le point d'intérêt et à orienter le robot au cours du mouvement de façon à ce que le FOE se superpose avec ce point. L'idée est que le champ observé doit se ramener à un champ purement divergent. Une analyse précise du champ de déplacement permet de déterminer plusieurs éléments tels que la vitesse tridimensionnelle du mobile, la projection sur l'image du centre de rotation instantané et le FOE. Il suffit alors de contrôler le système de manière à ramener ce point sur le point d'intérêt.

Dans le même esprit, l'objectif de la tâche présentée dans [Sundareswaran 96] est d'aligner, par une commande en rotation, l'axe optique de la caméra avec la direction de translation inconnue suivant laquelle elle se déplace. Cette application a été validée sur un robot manipulateur mais est parfaitement valable pour un robot mobile. La première idée développée est similaire à celle de l'article précédent, à savoir que la position du FOE est calculée et ramenée au centre de l'image. Une deuxième idée, qui ne fait jamais

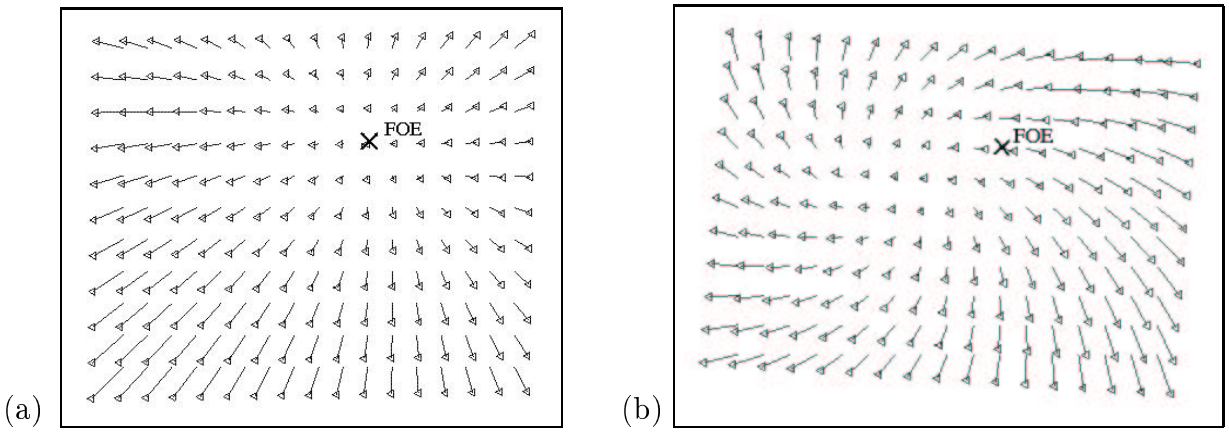


FIG. 1.4 – *Position du FOE sur deux champs de mouvement affine : pour un déplacement purement translationnel de la caméra (a), et pour la composée du même déplacement en translation avec une rotation (b)*

appel à des données géométriques de l'image, consiste à chercher à superposer le modèle de mouvement courant avec un modèle divergent. Pour cela, les auteurs montrent qu'il faut annuler la partie due à la translation des paramètres constants du modèle affine de mouvement. Ainsi, en déterminant l'influence de la rotation sur ces valeurs, il est alors possible de commander les vitesses angulaires du robot afin de minimiser ces paramètres.

1.4 Tâches de positionnement

Positionner une caméra par rapport à un objet consiste à définir des contraintes géométriques sur la position tridimensionnelle relative entre la caméra et l'objet. De nombreux travaux ont été menés dans ce cadre, et la plupart utilise des informations de type géométrique dans l'image pour remplir la tâche désignée.

Lorsque l'objet reste immobile, il semble qu'il y ait un antagonisme entre la notion de positionnement, devant amener à un état statique à l'équilibre, et la définition d'une contrainte sur le mouvement apparent dans l'image. En effet, lorsque cet équilibre est atteint, il n'y a a priori plus de mouvement du robot. Les mesures de mouvement apparent dans l'image sont alors nulles, et il est donc impossible d'extraire des informations sur la structure de la scène observée, telles par exemple, la profondeur le long de l'axe optique, ou l'orientation d'une surface. C'est pourquoi les quelques travaux qui prennent en compte le mouvement dans l'image pour effectuer des tâches de positionnement, le font en général en complément d'une contrainte géométrique. En particulier, lorsque l'objet est potentiellement mobile, l'utilisation des informations de mouvement apparent permet, à la convergence, d'effectuer un rejet des perturbations dues à ce mouvement propre de l'objet.

Utilisation du mouvement apparent pour robustifier une tâche de positionnement basée sur des informations géométriques

Une des principales causes d'échec pour des tâches de positionnement basées sur des informations géométriques est l'existence d'un déplacement propre de l'objet sur lequel le robot doit s'asservir. Les systèmes couplant vision et robotique fonctionnant, sauf dans de rares exceptions, au mieux à la cadence vidéo, si la loi de commande ne tient pas compte d'un éventuel déplacement de l'objet entre deux itérations, le robot se positionne par rapport à une donnée qui n'est plus exacte. Ainsi, plusieurs études utilisant des informations géométriques simples dans l'image pour effectuer un positionnement précis, ont été robustifiées par une mesure de la vitesse de l'objet, et ce, afin de prédire sa position et supprimer les erreurs de traînage. Nous pouvons citer dans ce cadre [Chaumette 93, Bensalah 95, Wilson 93, Feddema 93]. Cependant, une fois encore, ces travaux s'attachent à valider l'aspect robotique avant celui du traitement d'image en ne s'intéressant qu'à des objets dédiés à l'application considérée.

Une autre étude couplant l'utilisation d'informations géométriques et dynamiques est présentée dans [Martinet 96]. Le principe est de contrôler conjoitement la position d'un point d'intérêt dans l'image et sa vitesse apparente, de manière à stabiliser la convergence, en particulier pour éviter les risques d'oscillations lorsque le gain de la loi de contrôle est élevé. Cette approche se heurte à un problème théorique. En effet, le déplacement du robot est contraint de deux manières différentes qui peuvent être contradictoire : d'une part, afin d'amener le point dans la position désirée qui lui est associée et d'autre part, de manière à assurer une décroissance exponentielle de cette vitesse.

Plusieurs études ont été menées avec un objectif similaire, mais en considérant des objets plus réels. Nous en présenterons trois dans ce paragraphe, qui toutes font appel aux approches développées pour le suivi et présentées dans le paragraphe 1.2. Le choix de les classer sous l'étiquette de tâches de positionnement s'est fait avec l'idée que les objectifs qu'elles visaient nécessitent une précision beaucoup plus fine. En effet, l'objectif de chacune de ces trois applications est la préhension d'un objet par un bras manipulateur. Réaliser une telle tâche implique d'évidentes contraintes quant au positionnement du robot par rapport à la cible.

C'est le cas de l'article [Luo 88], où l'objet à saisir peut ou non être mobile. Même si le contexte est simplifié - mouvement de l'objet dans une seule direction de l'espace - un algorithme uniquement basé sur la détection de la position de l'objet ne pourrait pas garantir le succès de la tâche. Il est donc robustifié par une estimation du mouvement de l'objet afin de prédire la position qu'il atteindra à l'itération suivante.

Le projet mené par Nagel, dont l'objectif est le démontage automatisé de pièces d'un moteur de voiture, utilise une idée similaire, présentée dans plusieurs publications, comme par exemple [Gengenbach 96]. A la différence de l'étude précédente, la caméra n'est plus embarquée sur le bras du robot manipulateur, mais déportée sur un second bras, également

articulé. L'objectif de l'algorithme de vision est double, puisqu'il est sensé à la fois suivre l'organe préhenseur et la pièce à démonter, mais également se positionner au mieux pour observer la scène - en particulier pour éviter une occultation lorsque le préhenseur se rapproche de sa cible. Pour chacun des deux objets à suivre, sa position est déterminée par la superposition d'une vue de son modèle tridimensionnel, parfaitement connu, sur l'image. Comme précédemment, la phase d'approche est robustifiée contre les mouvements éventuels, par une mesure du champ de déplacement, afin de prédire les positions de l'effecteur et de la pièce.

Enfin, une troisième étude a été menée par l'équipe d'Allen et est présentée dans [Allen 93]. Dans ce cas, l'objet n'est pas déterminé a priori comme dans les deux études précédentes. Au contraire, le système doit se saisir de l'objet mobile présent dans la scène, sans aucune spécification sur sa structure, sous l'hypothèse, bien sûr, qu'il ne peut y avoir qu'un seul objet en mouvement. Il n'est donc pas possible de s'appuyer sur des données géométriques connues de la cible. Pour réaliser cet objectif, le système conçu est composé d'un bras robotique et d'une paire de caméras déportées, éloignées l'une de l'autre. Cet ensemble est parfaitement calibré, et l'objet mobile se déplace sur un plan horizontal toujours observable par les deux caméras. Le bras venant saisir l'objet par le dessus, le positionnement vertical est relativement facile à assurer, vu que cet axe n'est dégénéré dans aucune des deux images. La principale difficulté revient alors à déterminer la position de l'objet dans le plan horizontal. Cela est effectué par une détection du mouvement, dans chacune des images acquises par les deux caméras. Cette opération est relativement simple car les caméras restent immobiles, et le principe utilisé est de déterminer un champ dense de mouvement et de ne conserver que les pixels dont la norme du vecteur de déplacement est supérieure à un certain seuil. Le bras est alors à même de suivre l'objet mobile. Finalement, la préhension est encore une fois robustifiée en se servant des mesures de vitesses obtenues lors du calcul du champ de mouvement.

Définition des contraintes sur la trajectoire lors d'une tâche de positionnement basée sur des informations géométriques

Dans l'article [Colombo 95], les auteurs se donnent pour objectif de positionner la caméra par rapport à un objet planaire de façon rigide, c'est-à-dire qu'il n'existe qu'une seule position de l'espace possible pour la caméra. L'objet considéré est immobile, et on suppose qu'on peut toujours extraire son contour. L'image finale désirée est apprise au préalable. La méthode s'appuie sur le fait que la déformation d'un objet planaire due à un déplacement quelconque de la caméra est bien approximée par un modèle affine, à savoir une translation de son centre, et une transformation linéaire de la position de chacun des points du contour autour de ce centre. Cette transformation est calculée finement entre l'image initiale et l'image désirée. Le principe de la commande repose sur une séparation en deux tâches distinctes. La première consiste, par des méthodes classiques en asservissement visuel géométrique, à ramener le centre courant de l'objet sur la position désirée qui

lui correspond, par une commande en rotation sur les axes panoramique et d'inclinaison de la caméra. La seconde se base sur le calcul entre deux images successivement acquises du modèle affine de mouvement dans l'image, exprimé sous sa forme par descripteurs (cf. équation (1.7)). Le principe est alors d'imposer que la matrice de transformation calculée soit égale à la transformation désirée, à un facteur d'échelle près. L'intérêt de cette séparation est, en particulier que, si la commande avait entièrement été basée sur une transformation affine globale de l'image (translation et déformation linéaire), l'objet aurait pu sortir de l'image, ce qui n'est pas possible lorsque la position du centre est contrainte.

Une autre étude qui associe des contraintes géométriques et dynamiques dans l'image est présentée dans l'article [Grosso 96]. L'objectif des applications qui y sont développées est non seulement de positionner un objet par rapport à une cible, mais également de suivre une trajectoire déterminée pour atteindre la position désirée. L'exemple traité est celui qui consiste à amener le manipulateur du robot vers sa cible suivant une trajectoire rectiligne. Le système est observé par une caméra déportée, et pour s'affranchir du problème, classique en vision, lié à l'estimation de la profondeur, tous les mouvements sont réalisés dans un plan. La contrainte géométrique est simplement définie par l'obligation pour l'extrémité de l'effecteur, de se superposer avec un point donné de la cible. Ces deux positions dans l'image sont supposées pouvoir être extraites à chaque itération. Un champ dense est calculé autour de l'effecteur du robot afin de déterminer, par une minimisation aux moindres carrés, la vitesse projetée de son extrémité, et ce, avec une précision plus importante qu'en utilisant simplement la différence entre les positions mesurées de l'effecteur. La commande qu'impose la tâche géométrique est contrainte de façon à ce que ce vecteur soit colinéaire avec le segment joignant l'effecteur et sa cible, ce qui assure une trajectoire rectiligne. Le système de vision déporté étant monoculaire, il ne peut pas y avoir de contrôle de la trajectoire dans la direction de l'axe optique. Cela revient donc à contraindre le déplacement de l'effecteur sur un plan dont la direction est définie par l'axe optique et la direction de translation dans l'image.

Contrainte sur le modèle de mouvement projeté

Il existe également des études traitant d'asservissement visuel, où les déplacements de la caméra sont uniquement contraints par le mouvement apparent dans l'image. Par exemple, dans l'article [Questa 95], le but est de positionner la caméra parallèlement à un plan. L'idée exploitée est que lorsque la scène observée est planaire, l'orientation de ce plan par rapport à l'axe optique de la caméra apparaît explicitement dans les termes affines du mouvement projeté, et en particulier, dans les termes hyperboliques des descripteurs affines. Plus précisément, les tangentes des angles formés par le plan avec les axes $\overrightarrow{x}$ et $\overrightarrow{y}$ du repère de la caméra (respectivement les termes γ_1 et γ_2 de l'équation (1.6)), apparaissent dans ces termes hyperboliques, facteurs des translations de la caméra le long de ces mêmes axes (T_x et T_y). Connaissant la translation de la

caméra, par une mesure odométrique, la direction de plus grande pente est calculée. Le couplage d'une translation dans cette direction avec des rotations autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$, de façon à maintenir toujours le même point au centre, assure de pouvoir ramener la caméra parallèle au plan objet. Cette approche rencontre un problème aux abords de la convergence, puisqu'alors les translations orthogonalement à l'axe optique deviennent faibles, ce qui empêche d'obtenir des mesures significatives sur l'orientation de la caméra. Une seconde tâche peut être adjointe, qui consiste à contrôler le temps à collision de la caméra sur le plan objet, par une commande en translation le long de l'axe optique.

Dans l'article [SantosVictor 97], les auteurs utilisent la même idée, mais pour des tâches légèrement différentes, par le contexte dans lequel elles sont mises en applications. Le robot considéré est un engin mobile, et deux objectifs peuvent être imposés au système, selon que la caméra est portée par le robot et observe l'environnement qui l'entoure, ou, au contraire, que la caméra soit fixe et observe le robot en mouvement. Dans le premier cas, qui est le plus similaire à l'étude précédente, à la différence près que seule l'orientation autour d'un axe doit être prise en compte, l'engin se déplaçant sur un plan horizontal, le robot doit venir se positionner face à un mur, avec un temps à collision devant rester constant; les auteurs parlent alors d'"ego-docking". Dans le second cas, un plan est fixé à l'avant du robot et ce dernier doit venir se positionner face à la caméra qui l'observe; il s'agit alors d'"eco-docking". Ces applications auraient également pu être rattachées à la catégorie des tâches de navigation de robot mobile. Cependant, nous avons choisi de les présenter ici, car l'objectif recherché est différent. En effet, le but n'est pas d'assurer le libre déplacement du robot mobile dans son environnement, c'est-à-dire de définir les trajectoires impossibles, mais au contraire sa trajectoire est contrainte par la nécessité imposée au robot d'atteindre une certaine position dans l'espace.

La seconde partie de l'article [Sundareswaran 96] utilise également l'idée de définir une contrainte visuelle en terme d'un modèle de champ de mouvement 2D à atteindre. Nous l'avons cependant placé dans le cadre des travaux sur la navigation de robot mobile (même s'il ne met pas non plus en œuvre un matériel de ce type), car le positionnement de la caméra dans l'espace ne se fait pas par rapport à un objet de référence observé, mais en fonction du bras robotique sur lequel elle est montée.

Conclusion

Après avoir rappelé brièvement diverses manières de caractériser le mouvement apparent entre deux images successives d'une séquence, nous avons présenté quelques travaux utilisant l'une de ces approches afin de commander un système robotique. Nous avons séparé les études utilisant le mouvement apparent en trois catégories: le suivi d'un objet mobile, la navigation et le positionnement. La différenciation était effectuée selon l'objectif à réaliser par le robot. Une autre manière de les classer aurait pu être considérée. Dans

la plupart des cas, le traitement effectué pour extraire l'information réellement utilisée en entrée de la loi de commande comporte plusieurs phases, celle visant à obtenir des données sur le mouvement apparent n'étant que l'une d'entre elles. Ainsi, la classification aurait pu être basée sur le type de cette information donnée en entrée du bloc de commande, et correspondant à la sortie du bloc de traitement préliminaire.

Dans certaines des études présentées, l'analyse du mouvement permet d'extraire une information géométrique pertinente, qui ne pourrait pas, ou alors avec des algorithmes plus coûteux en temps de calcul, être obtenue directement. C'est le cas pour la plupart des travaux sur le suivi d'objet mobile, où une détection au sens du mouvement permet de déterminer la position de l'objet d'intérêt dans l'image. Nous pouvons également inclure dans cette catégorie, les travaux de détection d'obstacles par recherche des incohérences du champ dense de mouvement. L'approche que nous avons considérée pour réaliser une tâche de poursuite d'objet mobile, présentée dans [Crétual 98c], et détaillée dans le chapitre suivant, se place dans ce cadre. En effet, l'objectif étant de suivre un objet complexe, l'information de position de cet objet est obtenue à partir d'une mesure de son mouvement, que l'on intègre.

Dans d'autres approches, l'information de mouvement sert à robustifier une tâche définie par des contraintes géométriques. Il en est ainsi de la plupart des tâches de positionnement présentées, où l'information géométrique permet soit de prédire la position suivante de l'objet sur lequel le système s'asservit, soit de contraindre la trajectoire suivie pendant la convergence.

Enfin, quelques-uns des travaux présentés dans ce chapitre, se basent sur une contrainte directe sur le champ de mouvement dans l'image. Nous pouvons citer dans ce cadre les travaux de navigation dans un corridor, dont le principe est d'égaler les deux champs de mouvement latéraux, ainsi que les quelques cas où le but est de contrôler directement certains des paramètres du modèle affine de mouvement ([Sundareswaran 96, Grosso 95]).

En définitive, une classification selon le type d'information réellement prise en compte recouperait à peu de choses près celle que nous avons utilisée, si l'on considère que les deux principaux aspects évoqués dans le paragraphe sur la navigation (l'évitement d'obstacles et le déplacement dans un couloir) sont bien distincts. Cela montre que le choix de se baser sur telle ou telle caractérisation du mouvement pour remplir une tâche robotique est évidemment fortement lié à l'objectif donné pour cette application.

L'approche que nous avons choisi de suivre est celle de [Sundareswaran 96]. Cette étude, qui est à l'origine du sujet de thèse sur lequel nous avons travaillé, nous a permis de développer l'approche de l'asservissement visuel dynamique, que nous présentons au chapitre 3, et dont nous donnons quelques applications au chapitre 4 et dans l'article [Crétual 97]. En effet, cette approche visant à définir la contrainte visuelle comme un champ de mouvement 2D entre deux images à atteindre, nous est apparu comme novatrice dans le domaine de l'asservissement visuel.

Chapitre 2

Asservissement visuel géométrique à partir de mesures dynamiques

Quel que soit le schéma utilisé par chacune des méthodes d'asservissement visuel, un point qui leur est commun est la nécessité de travailler en temps-réel, c'est-à-dire d'être en mesure d'actualiser suffisamment souvent la commande envoyée aux actionneurs du robot. Dans le cas de l'asservissement visuel, la contrainte temporelle la plus forte concerne généralement le temps utilisé par l'algorithme de traitement d'images pour extraire l'information visuelle pertinente. La plupart des méthodes se basant sur des informations géométriques de l'image - à savoir, par exemple, les coordonnées de certains points de référence, ou les paramètres d'équations de droites - cette contrainte sur le traitement d'image est minimisée par l'utilisation d'objets fortement dédiés à la tâche effectuée et ayant la plus grande "simplicité visuelle" possible. Par ce terme, on désigne les objets qui sont, par exemple, aisément binarisables par un simple seuillage. Ainsi, si l'objectif de la tâche d'asservissement visuel est de positionner un ou plusieurs points de l'objet de référence à une position donnée dans l'image, un fort contraste entre ce ou ces points et le reste de l'objet permet évidemment d'optimiser le traitement d'image. C'est le cas dans de nombreuses publications, telles que [Espiau 92, Hashimoto 96].

Après avoir rappelé, dans le premier paragraphe de ce chapitre, en quoi consiste l'asservissement visuel géométrique, nous développons l'idée développée dans ce chapitre est qu'il est possible de s'affranchir de la contrainte sur le contenu de la scène observée en retrouvant la position des points d'intérêt à partir d'une mesure du mouvement dans l'image. En effet, connaissant la vitesse d'un point à chaque instant, on retrouve sa position par simple intégration. Ce principe simple est brièvement expliqué dans le second paragraphe.

Disposant d'une estimation de la position d'un ou plusieurs point d'intérêt de l'image, il est alors possible d'appliquer des lois de commandes identiques à celles qui ont pu être définies dans des travaux où cette position est considérée comme directement donnée par le traitement de l'image. Nous présenterons dans ce chapitre deux tâches robotiques qui

peuvent être remplies en suivant ce principe. La première est une tâche de suivi d'un objet mobile par une commande en rotation panoramique et en inclinaison de la caméra, en maintenant le centre de gravité estimé de cet objet au centre de l'image. La seconde consiste à effectuer un suivi rigide d'un objet mobile, c'est-à-dire à maintenir constante la position tridimensionnelle de l'objet dans le repère caméra. Cette contrainte peut se traduire par le maintien à la même position dans l'image des projections de quatre points de contrôle de l'objet. Les coordonnées de ces quatre points image sont retrouvées en approximant leur vitesse par l'application d'un modèle du mouvement dominant estimé. Deux modèles de mouvement 2D sont étudiés et leurs performances sont comparées. Enfin, pour clore ce chapitre, nous nous posons la question de savoir si des problèmes d'asservissement visuel autres que ceux de suivi peuvent être résolus par cette méthode.

2.1 Rappels sur l'asservissement visuel 2D

2.1.1 Principe

Le principe de l'asservissement visuel géométrique (asservissement visuel 2D) consiste à piloter un robot, sur l'effecteur duquel est montée une caméra, ou qui est observé par une caméra déportée, de telle manière que des informations géométriques de l'image viennent se superposer avec des positions désirées. Nous ne considérerons par la suite, que le cas où la caméra est embarquée sur le robot. Plus précisément, dans l'exemple que donnons sur la figure (2.1), le but de la tâche est de ramener les quatre sommets d'un carré, apparaissant dans l'image courante dans les positions correspondant aux croix sur la figure a, sur les positions désirées, données par les cercles sur la figure b. Les primitives géométriques utilisées peuvent prendre d'autres formes que des positions de points, telles, par exemple les paramètres qui caractérisent des droites.

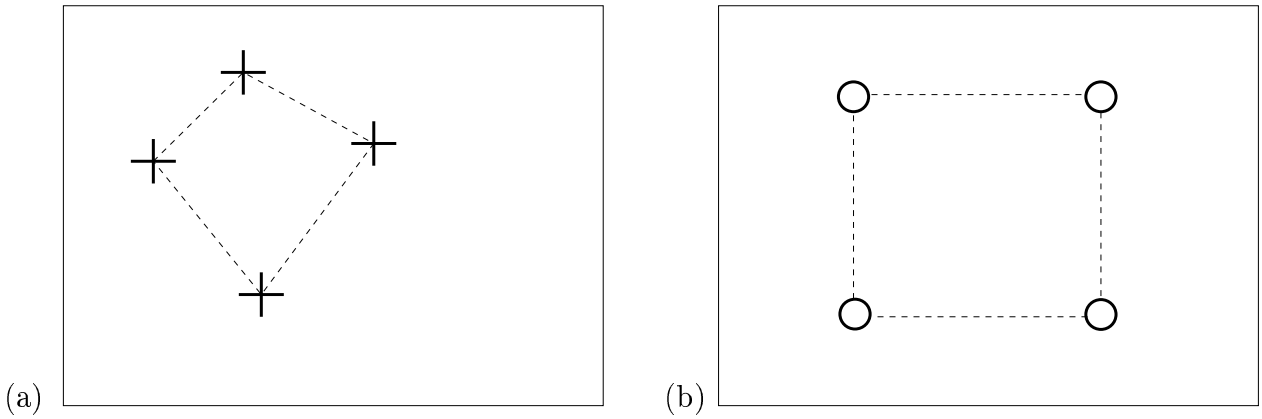


FIG. 2.1 – *Position courante du carré dans l'image (a) et position désirée (b)*

Nous résumons cette approche dans le schéma bloc de la figure (2.2), où : s désigne le vecteur des informations géométriques extraites de l'image, s^* sa valeur désirée, et (T, Ω) le torseur cinématique de la caméra qui est calculé par la loi de commande. La différence avec la méthode présentée au chapitre précédent, (cf. le schéma bloc de la figure (2.3)) réside uniquement dans le fait que le bloc "traitement d'image" de ce schéma de commande était décomposé en trois parties : une extraction du mouvement dans l'image sous forme d'un modèle polynomial, une approximation de la vitesse en chaque point à partir de ce modèle de mouvement, et enfin, l'estimation de la position de points d'intérêt par intégration de leur vitesse donnée par le modèle.

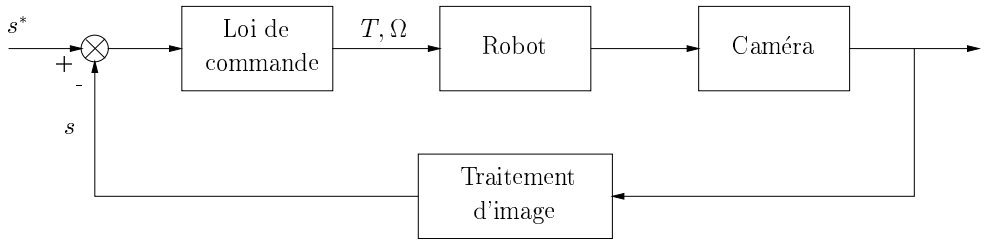


FIG. 2.2 – *Schéma bloc de l'asservissement visuel géométrique*

2.1.2 Schéma de commande

Dans le schéma de l'asservissement visuel géométrique, une des étapes essentielles consiste à définir une fonction d'erreur qui caractérise la différence entre l'observation de la scène observée, à un instant donné, et celle de la scène désirée, c'est-à-dire celle qui correspond à l'accomplissement de la tâche à remplir. La tâche est remplie lorsque cette fonction d'erreur est nulle et seulement dans ce cas. Cette erreur e dépend linéairement d'un vecteur d'informations visuelles s extraites de l'image, et n'est fonction que de la position relative de la caméra par rapport à l'objet d'intérêt.

Si l'objet d'intérêt observé est immobile, la dérivée $\dot{s}$ de s peut être exprimée comme une relation linéaire du torseur cinématique V_c entre la caméra et un repère fixe, sous la forme :

$$\dot{s} = L\ V_c \tag{2.1}$$

où L est appelée la matrice d'interaction associée à s.

La fonction tâche e, qui doit être regulée pour atteindre une valeur nulle, est définie par :

$$e = C(s - s^*)$$

où s^* représente la valeur désirée de s et C est une matrice, que l'on choisit, et qui permet, entre autres, d'assurer que la taille du vecteur e soit la même que celle du torseur cinématique V_c.

Remarque : Il peut, bien-sûr, arriver que la dimension de l'espace de configuration du système que l'on souhaite contraindre soit inférieur à 6. Les deux relations précédentes se récrivent alors en considérant la restriction de V_c aux seules vitesses considérées pour le contrôle.

On peut, en combinant les deux relations précédentes, définir une loi de commande de type gradient, en imposant une décroissance exponentielle de gain λ de l'erreur ($\dot{e} = -\lambda e$). Nous obtenons alors :

$$V_c = -\lambda \left(CL\right)^{-1} e \tag{2.2}$$

Si L est de rang maximal, c'est-à-dire de rang égal à la dimension n de l'espace de configuration du système, il est toujours possible de choisir la matrice C de façon à ce que la matrice CL, de dimension $n \times n$, soit inversible.

En pratique, il est souvent difficile d'obtenir une valeur précise de la matrice L, de même que pour le vecteur e. Ils sont donc approximés respectivement par $\widehat{L}$ et $\widehat{e}$. Par conséquent, si nous rebouclons le système en remplaçant V_c par sa valeur donnée par la loi de commande, dans la relation (2.2), nous obtenons :

$$e = \left(CL\right) \left(C\widehat{L}\right)^{-1} \widehat{e}$$

Une condition de convergence de e vers zéro est donnée par la positivité de la matrice :

$$K = \left(CL\right) \left(C\widehat{L}\right)^{-1}$$

ce qui montre un second intérêt de la matrice C, dont le choix peut stabiliser le système. Le choix optimal, si L était parfaitement estimée, serait de considérer $C = L^+$, la pseudo-inverse de L. Généralement, on choisit donc $C = \widehat{L}^+$, la pseudo-inverse de $\widehat{L}$. Un autre choix possible, lorsque s est de la même dimension que V_c, est de fixer simplement la matrice C égale à l'identité.

Remarque : Lorsque l'objet est en mouvement, la relation (2.1) s'écrit :

$$\dot{s} = L\ V_c + \frac{\partial s}{\partial t}$$

où le deuxième terme rend compte de l'influence du mouvement propre de l'objet.

2.2 Principe de l'utilisation d'information 2D à partir de mesures visuelles dynamiques

Nous considérons dans ce cas que l'information visuelle s se réduit à la position d'un point dans l'image. Ainsi, nous notons $s = (x, y)^T$ la projection sur l'espace image 2D d'un point M de l'espace opérationnel 3D, et $\dot{s}$ sa vitesse apparente dans l'image. Ce vecteur s peut être évidemment retrouvé à partir de l'évolution de $\dot{s}$ au cours du temps, pour peu que l'on ait sa valeur initiale s_0 à l'instant 0 par la relation :

$$s = s_0 + \int_O^t \dot{s}\ dt \tag{2.3}$$

Cette estimation de la position s est soumise à deux problèmes. Tout d'abord, elle nécessite de connaître parfaitement la position initiale s_0, sans quoi un biais apparaît qui ne peut pas être corrigé. De même, un bruit dans la mesure de la vitesse $\dot{s}$ entraîne une erreur d'estimation de s. Si ce bruit n'est pas un bruit blanc additif, il apparaît alors également un biais.

La relation (2.3) peut être exprimée sous forme discrète par :

$$s_k = s_0 + \sum_{i=1}^{k} \dot{s}_i\ \delta t_i \tag{2.4}$$

avec $\dot{s}_i$ la i$^{\text{ème}}$ mesure de $\dot{s}$, et δt_i le temps écoulé entre les (i-1)$^{\text{ème}}$ et i$^{\text{ème}}$ mesures.

Comme l'estimation d'un champ dense de mouvement est très coûteuse en temps de calcul et revient à faire un suivi de chaque point de l'image, le mouvement d'un point est reconstruit par application d'un modèle estimé. Le modèle utilisé pour approximer la vitesse apparente doit être choisi de façon à optimiser le compromis entre précision et rapidité de l'estimation. Dans les faits, on utilise le modèle quadratique simplifié à huit paramètres présenté dans le chapitre précédent, le modèle affine à six paramètres déduit du précédent, ou encore le simple modèle constant à deux paramètres.

Ce dernier modèle, même s'il peut paraître simpliste, nous sera malgré tout très utile pour réaliser le suivi d'un objet par une commande en rotation panoramique et en inclinaison. En effet, vu qu'alors seuls deux degrés de liberté sont contraints, l'estimation de la position d'un point de cet objet grâce au modèle constant est suffisante.

Au contraire, lorsque l'on souhaite réaliser un suivi rigide de l'objet mobile, nous verrons que le modèle affine de mouvement est insuffisant. Il faut alors recourir au modèle quadratique pour pouvoir réaliser cet objectif.

Le schéma bloc de la figure (2.3) résume le principe de la méthode où s et s^* représentent respectivement la position estimée courante et la position désirée dans l'image du point d'intérêt, $(T, \Omega)^T$ le torseur cinématique commandé de la caméra et a_i les informations visuelles dynamiques fournies par l'algorithme d'estimation du mouvement et permettant de retrouver $\dot{s}$.

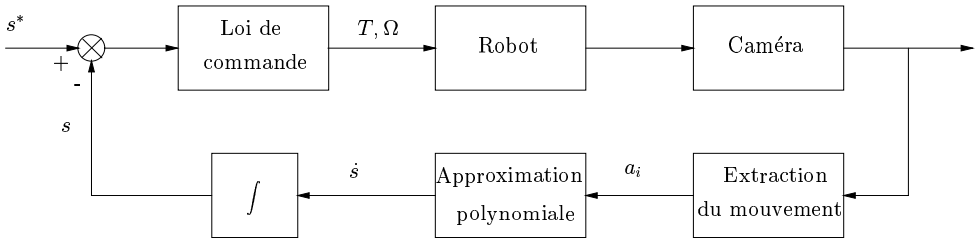


FIG. 2.3 – *Schéma bloc : Asservissement visuel basé sur des informations géométriques retrouvées à partir de mesures dynamiques*

Ce schéma bloc est très similaire au schéma général de l'asservissement géométrique de la figure (2.2), puisque nous réalisons également dans ce cas un asservissement 2D. Nous avons simplement explicité la manière d'obtenir l'information visuelle utilisée en entrée du bloc de commande. Les trois blocs inférieurs correspondent donc au bloc "Traitement d'image" de la figure (2.2).

Enfin, ce schéma de contrôle peut être évidemment généralisé à une commande basée sur le positionnement de plusieurs points de l'image en considérant le vecteur s comme composé des coordonnées de l'ensemble de ces points.

2.3 Poursuite d'un objet mobile en pan et tilt

Le principe de cette tâche de poursuite est de contrôler les deux degrés de liberté de la caméra en rotation autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$, respectivement appelés panoramique (*pan*) et inclinaison (*tilt*), de façon à amener et à maintenir au centre de l'image la projection d'un objet d'intérêt.

Nous ne traitons pas ici des problèmes d'occultations, par un autre objet, de l'objet d'intérêt, en considérant que ce dernier est toujours observable par la caméra. Nous ne nous intéressons pas non plus à la sélection de l'objet d'intérêt dans le cas ou plusieurs objets sont susceptibles d'être mobiles dans la scène observée. L'objet d'intérêt est le premier détecté en mouvement.

2.3.1 Un rapide tour d'horizon de travaux existants

De nombreux articles ont été publiés sur ce sujet. Des contraintes temps-réel sont évidemment à prendre en compte. La vitesse de l'objet étant inconnue et pouvant être élevée, le système de poursuite doit être suffisamment réactif pour ne pas risquer de laisser l'objet sortir du champ de vue. En ce sens, plusieurs travaux ont été menés dans des cas où le traitement de l'image a pu être simplifié par l'utilisation d'objets fortement dédiés à cette application.

Outre les divers travaux présentés dans le chapitre précédent [Bartholomeus 93, Nordlund 96, Allen 93] et qui utilisent le mouvement apparent pour détecter et suivre un

objet mobile, de nombreux articles ont été publiés sur ce même thème mais qui n'utilisent que des primitives géométriques. Afin d'extraire rapidement la position de l'objet d'intérêt dans l'image, [Bensalah 95, Corke 93, Hashimoto 93, Papanikolopoulos 95, Chaumette 93, Jung 98, Heuring 96] utilisent des marqueurs visuels du type points noirs sur fond blanc, rectangle blanc sur fond noir ou encore des diodes électro-luminescentes. Ces algorithmes sont en général très performants et robustes à la complexité du mouvement propre de l'objet. Ils ne sont par contre pas très robustes aux occultations, et ce à cause de la nécessité évidente de toujours observer les marques visuelles. Leurs limites ne sont pas tant dans leur capacité de réaction dans des situations délicates (mouvement rapide et/ou changeant brusquement, etc.), mais dans la diversité des objets pouvant être considérés. Leur intérêt est donc du domaine de la robotique, afin de valider un schéma de commande, mais pas du point de vue de la vision.

Une série de travaux a été menée par plusieurs équipes afin de définir et de construire des têtes robotiques binoculaires très performantes. Ces recherches présentées dans [Madden 95, Wavering 95, Du 94, Milios 93] sont très axées sur l'aspect technologique, tant sur l'organe de vision que sur le matériel robotique. De même, [Nakabo 98] présente un système "eye-in-hand", dont la caméra est optimisée technologiquement de façon à acquérir des images à la fréquence de 1000 Hz. Cependant, hormis [Du 94] qui réalise le suivi par recherche de points de forts gradients, les autres systèmes valident leurs algorithmes de suivi sur des cibles qui sont facilement séparables du fond de la scène (l'objet mobile évoluant en général devant un fond de couleur uniforme).

Une façon de contourner le problème est d'utiliser, comme dans l'article [Feddema 90] des objets "réels" connus, ce qui permet là encore d'optimiser le traitement d'image. Dans le même esprit, [Okhotsimsky 98] présente des résultats très robustes de préhension d'un objet mobile, mais avec un objet parfaitement connu, soumis de plus à des contraintes mécaniques également connues, qui restreignent son mouvement.

Plusieurs articles tels [Inoue 92, Papanikolopoulos 95, Christensen 95, Reid 96, Smith 97, Nickels 98] proposent une autre approche qui, cherchant un compromis entre réalisme des objets à suivre et rapidité de l'extraction des informations visuelles, met en œuvre un algorithme de vision ne nécessitant pas de cible dédiée. Le principe est de suivre des points correspondant à des coins. Dans ce cas, le suivi des points a une signification avant tout dans des termes de vision. Il s'agit d'abord de retrouver un point d'intérêt d'une image dans l'image suivante avant d'envisager de contrôler le système robotique pour maintenir de ce point dans une position désirée. Ce suivi visuel est réalisé par des techniques de mise en correspondance locale autour du point en question, comme par exemple la méthode SSD (appelée encore technique du "block matching"). L'aspect robotique est similaire aux travaux présentés auparavant, mais la partie vision de cette approche est plus générale. Le problème de l'observabilité des points considérés est par contre toujours présent, même si le nombre redondant de coins détectés permet d'en perdre une partie. Enfin, [Arsénio 97, Cipolla 97b] proposent de suivre la cible par recalage sur chaque image d'une courbe déformable ("snake") sur son contour.

2.3.2 Loi de commande associée

Dans ce cas particulier, on ne cherche à suivre qu'un point d'intérêt, le centre de gravité (c.d.g.) de la cible. La relation (2.4) donnant, à chaque itération, une estimation de la position du c.d.g., il est possible de commander les mouvements de la caméra de façon à l'amener au centre de l'image. La loi de commande que nous avons appliquée est une loi classique de type Newton, telle que celle présentée dans l'article [Espiau 92] et améliorée dans [Chaumette 93].

Soit le vecteur d'informations visuelles $s = (x, y)^T$ où x et y représentent respectivement l'abscisse et l'ordonnée du c.d.g. estimé dans le repère image. La position désirée s^* est le centre de l'image, c'est-à-dire $s^* = (0, 0)^T$. On considère le vecteur d'erreur, ou fonction de tâche, $e = s - s^* = s$. En utilisant le fait que seuls les degrés de liberté de rotation autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$ du repère caméra sont utilisés par la commande, et donc que les mouvements sont nuls sur les quatre autres axes, on obtient la relation d'interaction suivante :

$$\dot{e} = L \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} + \frac{\partial e}{\partial t} \text{ avec } L = \begin{bmatrix} xy & (-1 - x^2) \\ (1 + y^2) & -xy \end{bmatrix} \tag{2.5}$$

où $\frac{\partial e}{\partial t}$ représente la contribution du mouvement propre de la cible sur les variations de e dans l'image et Ω désigne le vecteur de vitesse de rotation de la caméra.

En spécifiant une décroissance exponentielle de l'erreur avec un gain λ ($\dot{e} = -\lambda e$), la loi de commande est donnée par :

$$\begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} = \frac{-\lambda}{1 + x^2 + y^2} \begin{pmatrix} y \\ -x \end{pmatrix} - L^{-1} \widehat{\frac{\partial e}{\partial t}} \tag{2.6}$$

Le premier terme de cette loi de commande permet seulement d'atteindre la convergence quand l'objet à suivre est immobile. Le second terme permet d'éviter les erreurs de traînage dans le cas, plus intéressant, où l'objet est en mouvement. Il est estimé de la manière suivante [Chaumette 93] :

$$\widehat{\frac{\partial e}{\partial t}} = \widehat{\dot{e}} - L\widehat{\Omega}$$

où $\widehat{\dot{e}} = \widehat{\dot{s}}$ est directement donné par l'algorithme d'estimation du mouvement et $\widehat{\Omega}$ est le mouvement de rotation de la caméra mesuré par odométrie. Cette estimation est filtrée par un filtre de Kalman à accélération constante et bruit corrélé, identique à celui présenté dans [Chaumette 93]. Un amélioration de ce filtrage, que nous n'avons pas prise en compte, a été apportée dans [Bensalah 95], en y ajoutant un algorithme de détection et d'estimation de saut, le Generalized Likehood Ratio (GLR) test.

Une autre loi de commande basée sur le Generalized Predictive Controller (GPC) a été proposée dans [Gangloff 98]. Cette approche permet de tenir compte explicitement de

la dynamique du manipulateur et donc de compenser également les erreurs de trainage en les considérant comme des perturbations à rejeter.

2.3.3 Simulation

Cette tâche de poursuite en *pan* et *tilt* a été simulée dans le contexte suivant :

- l'objet est placé à 1 m de la caméra ;
- son centre apparaît au départ à la position $s_0 = (-0.15, 0.1)^T$ exprimée en mètre dans le repère image pour une distance focale égale à 1 ;
- il est soumis à un mouvement complexe composé d'une suite de translations et de rotations constantes par morceaux ; les translations sont de l'ordre de 20 à 30 cm/s et les rotations de 0.1 à 0.2 rad/s ;
- le modèle de mouvement considéré pour estimer la vitesse du point d'intérêt est le modèle constant à deux paramètres (c_1 et c_2) ;
- la valeur de λ est fixée à 1.5.

Remarque : Il convient de ne pas confondre le modèle à accélération constante du filtre de Kalman, et le modèle constant du mouvement apparent dans l'image. Le premier se rapporte au mouvement propre de l'objet, et permet de considérer un ensemble assez large de mouvement, de façon à prédire la position de l'objet dans l'image, grâce à la phase de prédiction du filtre. Le second permet de réaliser une approximation du mouvement apparent de l'objet, afin d'optimiser le temps de calcul entre deux itérations.

Nous présentons sur la figure (4.1), le mouvement tridimensionnel de l'objet dans un repère fixe dans lequel le système évolue. Il s'agit des composantes de translations (a) et de rotation (b), qui sont toutes constantes par morceaux et atteignent des valeurs de l'ordre de 0.3 m/s et 0.3 rad/s. Sur chacune des courbes, la composante en x est en trait plein, celle en y, en trait pointillé, et celle en z en trait discontinu. Sur la figure (2.5), on montre le déplacement que subit le centre de gravité de l'objet dans l'image lorsque la caméra reste immobile, avec, respectivement, les composantes x en trait plein et y en trait pointillé. Le déplacement final est de l'ordre de 0.8 m dans chaque direction.

La figure (2.6) montre les comparaisons entre d'une part (a), les abscisses estimée (x_{est}, en trait plein) et vraie (x_{vrai}, en trait pointillé) du point d'intérêt, et d'autre part (b), ses ordonnées estimée (y_{est}, en trait plein) et vraie (y_{vrai}, en trait pointillé). Toutes ces coordonnées sont exprimées en mètre pour une focale de 1. Les vitesses angulaires (en deg/s) correspondant à cette simulation (Ω_x, en trait plein et Ω_y, en trait pointillé pour respectivement, les rotations autour de l'axe $\overrightarrow{x}$ et $\overrightarrow{y}$) sont exposées sur la figure (2.7).

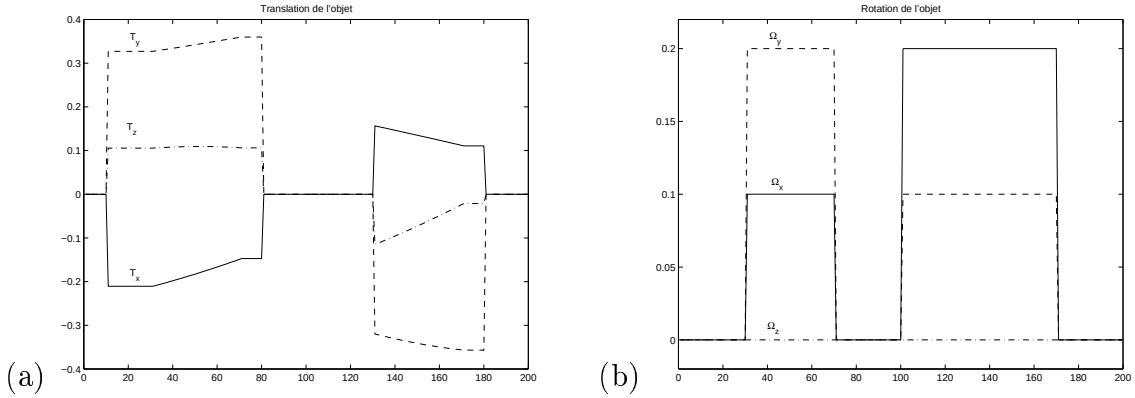


FIG. 2.4 – *Mouvement de l'objet dans le repère fixe, avec les translations (a) et les rotations (b). $T_{obj,x}$ et $\Omega_{obj,x}$ en trait plein (—), $T_{obj,y}$ et $\Omega_{obj,y}$ en trait discontinu (– · –), $T_{obj,z}$ et $\Omega_{obj,z}$ en trait pointillé (- - -)*

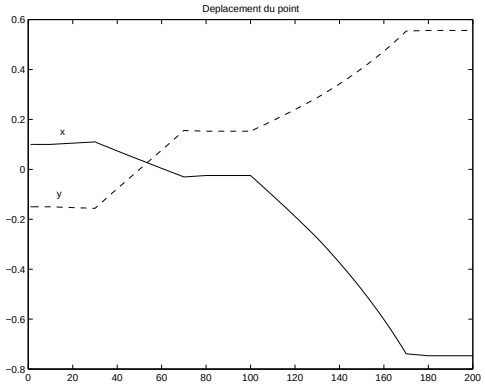


FIG. 2.5 – *Déplacement dans l'image du centre de l'objet avec une caméra immobile*

On constate que malgré la simplicité du modèle de mouvement considéré, l'erreur entre les positions estimée et exacte est faible (moins de 2 mm, soit moins de 2% de l'erreur initiale). Il reste cependant une erreur résiduelle qui n'est pas compensée mais qui reste bornée. Ce phénomène est facilement explicable par le fait que, dans la première phase de la commande qui consiste à ramener l'objet au centre de l'image, l'imprécision du modèle de mouvement est sensible et un biais apparaît dans l'intégration. Cependant, lorsque l'objet est ramené au centre de l'image, le modèle constant est fiable et il n'y a pas de biais supplémentaire dans l'estimation de la position. De nouvelles simulations nous ont montré que lorsqu'on considère un modèle quadratique du mouvement pour estimer la position, le biais est inexistant grâce à la bonne adéquation entre ce modèle et le mouvement réel. Enfin, on remarque que la sollicitation de la commande en rotation reste modérée (moins de 15 deg/s) malgré le mouvement relativement rapide de l'objet et sa position initiale éloignée du centre l'image.

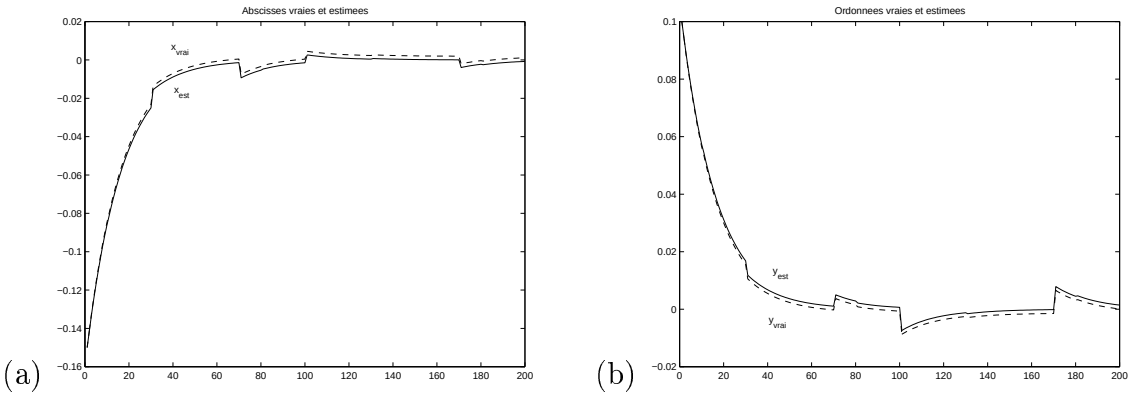


FIG. 2.6 – *(a) Comparaison des abscisses estimée et vraie du point suivi. (b) Comparaison des ordonnées estimée et vraie du point suivi.*

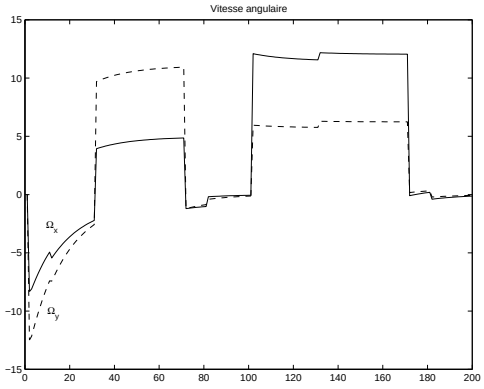


FIG. 2.7 – *Vitesse angulaire commandée (en deg/s)*

De plus, à chaque changement dans le mouvement propre de l'objet, le point d'intérêt subit un déplacement modéré dans l'image, qui est ensuite compensé. Ce phénomène s'explique très simplement par le temps de réaction nécessaire pour que le système puisse corriger cette variation, alors qu'il se trouve alors dans un régime permanent.

2.3.4 Expérimentations

Conditions des expérimentations

La tâche de poursuite d'une cible a été testée sur la cellule de vision robotique de l'IRISA, composée d'un robot cartésien à 6 degrés de liberté sur l'effecteur duquel est monté une caméra CCD. Les images acquises par une carte SunVideo et traitées par une station de travail UltraSparc sont de taille 256×256 pixels. Une image initiale type est présentée sur la figure (2.8) sur laquelle l'objet d'intérêt est la photographie de forme rectangulaire au premier plan. Il apparaît que l'extraction d'un point particulier de cet objet est délicate. De même, une segmentation par niveau de gris ne permet pas de le distinguer du fond qui est également texturé.

FIG. 2.8 – *Image initiale de la séquence acquise*

Détection de l'objet mobile

Avant de pouvoir procéder à l'étape de poursuite, il est nécessaire d'avoir désigné lequel des objets présents dans la scène est celui d'intérêt. Cette phase de détection permet de déterminer le masque binaire de la projection de l'objet à suivre. Cette détection est effectuée sur le seul critère du mouvement de l'objet d'intérêt. Ainsi, le suivi s'effectuera sur le premier objet en mouvement dans l'image. Pour éviter un amalgame entre deux parties mobiles dans l'image qui ne correspondraient pas au même objet physique, on se place sous l'hypothèse qu'un seul objet peut-être mobile pendant cette étape.

Si la caméra reste statique tant que cette phase n'est pas achevée, la projection de l'objet mobile sur un fond également statique peut être simplement déterminée par une différence seuillée pixel à pixel de deux images successives. En pratique, à cause du bruit dans l'image, on utilise des images déduites de celles acquises par la caméra par un moyennage local sur un carré de 3×3 pixels. Le centre de gravité du masque binaire obtenu est considéré comme le point de référence s_0 qui doit être amené à l'origine du repère image.

Le masque binaire résultant de la détection de l'objet par son mouvement est présenté sur la figure (2.9). Ce n'est pas directement ce masque qui est utilisé comme la zone de l'image sur laquelle s'effectue l'estimation du mouvement, mais celui que l'on obtient en lui appliquant une dilatation par l'élément structurant correspondant à un carré de 5 pixels de côté. La fenêtre d'estimation est la zone rectangulaire englobant ce masque. Dans cette fenêtre, la distinction entre les points correspondant à l'objet et les autres se fait en fixant les valeurs de pondérations de ces premiers pixels à 1 et les autres à 0, lors de l'initialisation de l'algorithme RMRm (cf. [Odobez 95]). Enfin, de manière à ce que le masque se superpose au mieux à l'objet au cours de la poursuite, la fenêtre d'estimation est déplacée à chaque itération dans la direction du mouvement estimé, à sa position prédite.

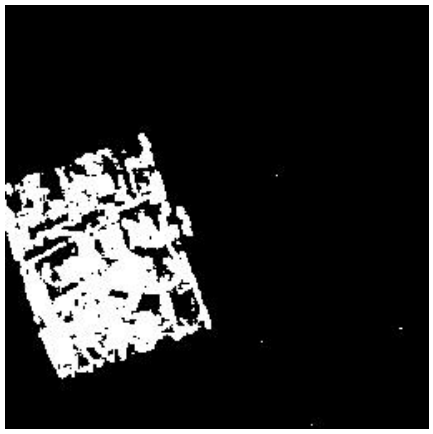


FIG. 2.9 – *Masque binaire de l'objet résultant de la phase de détection*

Validation de l'approche

Afin d'évaluer les performances de l'algorithme de suivi du centre de gravité de l'objet d'intérêt, nous devons comparer les positions réelle et estimée de ce point à chaque itération. Cependant, l'objet présenté sur la figure (2.8) ne permet pas l'extraction en temps réel de la position 2D d'un point désigné. Il est donc nécessaire de tester la tâche de poursuite sur un objet dont on peut facilement extraire la position du centre de gravité, sans pour autant utiliser cette information dans la boucle de commande. Il s'agit

du carré défini par ses quatre coins présenté sur la figure (2.10), sur laquelle il apparaît clairement que le traitement d'image à mettre en œuvre pour en extraire les positions 2D est simple. Ayant ces données à chaque itération et le temps écoulé entre les deux acquisitions d'images, on remonte à la vitesse 2D de chacun des points. En appliquant un modèle affine du mouvement, de manière à garantir une précision suffisante sur l'estimation des paramètres constants, nous arrivons à un système sur-dimensionné de 8 équations et 6 inconnues résolu par moindres carrés. La projection du centre de gravité de ce carré est calculé comme le point d'intersection des deux diagonales du carré, vu que si le barycentre n'est pas conservé par projection, l'intersection de deux droites, par contre, l'est. Cette mesure est plus robuste aux déformations affines que la simple moyenne des positions des quatre coins.

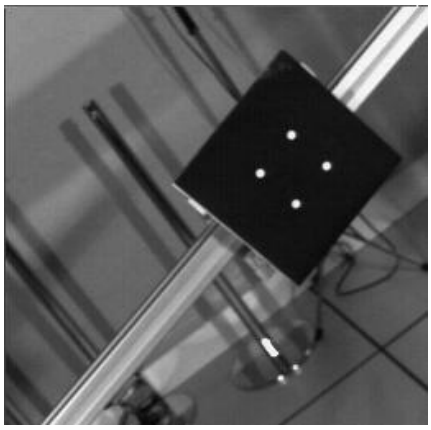


FIG. 2.10 – *Image initiale de la séquence de validation*

L'objet a été animé d'un mouvement de translation le long du rail qui apparaît de chaque côté de la cible sur la figure (2.10). Ce mouvement consiste en des phases de déplacement alternativement vers la droite et vers la gauche, à vitesse constante, entrecoupés de phases d'arrêt de 4 secondes. Les ralentissements et les démarrage se font à accélération constante. La norme de la vitesse est de 8 cm/s dans la première moitié de l'expérience et de 30 cm/s dans la seconde.

La différence entre les positions estimée et mesurée du centre de gravité est présentée sur la figure (2.11) (en trait plein selon l'axe $\overrightarrow{x}$ de l'image, en trait pointillé selon l'axe $\overrightarrow{y}$). Il apparaît que cette différence est toujours inférieure au demi-pixel, ce qui valide l'absence de biais lors de l'estimation par intégration de la vitesse apparente.

L'algorithme d'estimation de la vitesse 2D étant différent selon que l'objet utilisé est le plan texturé "réel" ou le carré "quatre points", on doit s'interroger sur l'influence du mode d'estimation sur l'erreur. Afin de comparer la qualité de l'algorithme RMRm à celle obtenue par la méthode simple de résolution du système linéaire dans le cas du carré "quatre points", des mesures ont été effectuées dans un environnement connu. Cela

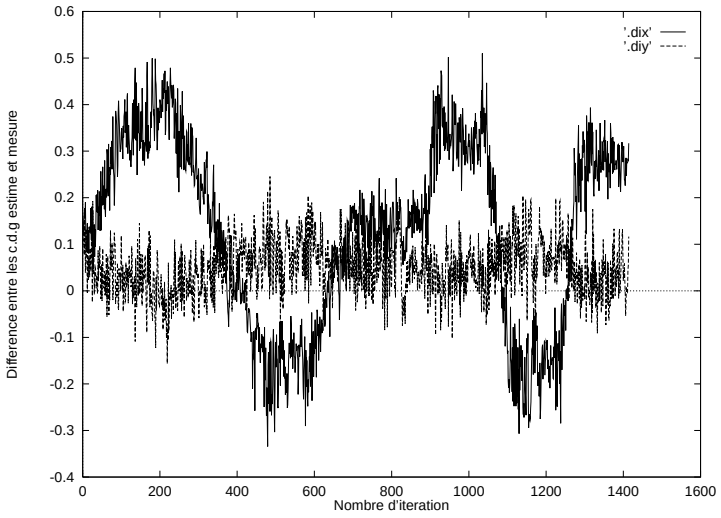


FIG. 2.11 – *Test de validation. Différence entre la position estimée du centre de gravité du carré "quatre points" et sa position mesurée (en pixel)*

signifie que la distance de la caméra à l'objet, les vitesses en rotation de la caméra et en translation de l'objet sont connues. L'intérêt de cette connaissance n'était pas de pouvoir vérifier l'adéquation de cette mesure avec ce qui était attendu - en particulier à cause des incertitudes sur la calibration du système (à la fois le robot manipulateur et le banc en translation) - mais de pouvoir se placer dans des conditions similaires, de façon à pouvoir comparer les bruits sur la mesure.

Les résultats obtenus (cf Annexe A) prouvent que la confiance que l'on peut accorder à l'algorithme RMRm est forte, en particulier en ce qui concerne l'estimation des paramètres constants. Dans le cas de déplacements faibles de l'objet, l'algorithme RMRm donne même des résultats nettement moins bruités.

Résultats

Les conditions expérimentales considérées lors des essais de la tâche de poursuite de l'objet "réel" avec estimation par l'algorithme RMRm étaient semblables à celles du test de validation, à savoir des déplacements le long du rail, à des vitesses de 8 cm/s puis de 30 cm/s, à environ 1 m de la caméra.

Le déplacement estimé du centre de gravité de l'objet est présenté sur la figure (2.12) (en trait plein selon l'axe $\overrightarrow{x}$, en trait pointillé selon l'axe $\overrightarrow{y}$), avec une vitesse de 8 cm/s de l'itération 1 à l'itération 700 environ et une vitesse de 30 cm/s de l'itération 800 à l'itération 1400. La convergence initiale est rapidement obtenue (moins de 40 itérations), et ce malgré, d'une part, une erreur initiale de 40 pixels et d'autre part, le fait que le premier mouvement de la cible ait tendance à l'éloigner du centre. A chaque changement brusque dans le mouvement propre de la cible (démarrage ou arrêt), un dépassement

apparaît, dû au temps de réaction du filtrage de Kalman, mais la convergence est de nouveau rapidement obtenue, dès que l'on se retrouve dans une phase de mouvement permanent. De plus, la convergence est toujours obtenue avec très peu d'oscillations. Ce dépassement est d'environ 10 pixels et compensé en 20 itérations, lorsque la vitesse est de 8 cm/s, et d'environ 25 pixels et compensé en 50 itérations, lorsque la vitesse est de 30 cm/s.

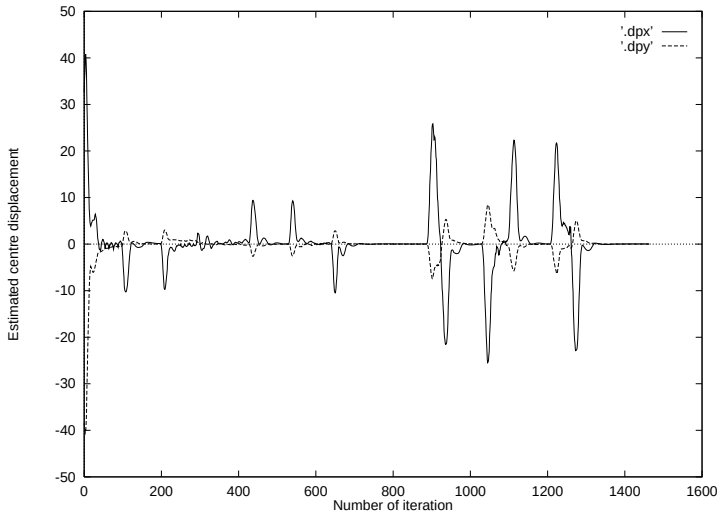


FIG. 2.12 – *Suivi de l'objet "réel". Déplacement estimé du centre de gravité en pixel*

Les vitesses de rotation calculées à chaque itération, sont présentées sur la figure (2.13) (en trait plein autour de l'axe $\overrightarrow{x}$, en trait pointillé autour de l'axe $\overrightarrow{y}$). Une vitesse de 4 deg/s environ est nécessaire pour suivre l'objet lorsqu'il se déplace à 8 cm/s. Lorsque l'objet est animé d'une translation à 30 cm/s, le système doit réagir par une rotation de l'ordre de 15 deg/s. Dans le premier cas, la convergence étant rapidement obtenue et le régime permanent étant assez long entre un démarrage et un arrêt, la vitesse de rotation atteint un palier. Au contraire, dans le second cas, les rotations Ω_x et Ω_y atteignent leurs valeurs maximales juste avant l'arrêt de l'objet. Cependant, il n'apparaît pas de perturbation nette dans la régulation de la position du centre de gravité estimée.

Une partie de la séquence acquise est présentée sur la figure (2.14). Il s'agit du début de la séquence, où apparaît l'image initiale, c'est-à-dire la première traitée pendant la phase de suivi après la détection, puis une image sur dix du premier "aller-retour" de la cible. Sur chacune de ces images, le centre de gravité estimé de l'objet est représenté par une croix blanche (+), et le centre de l'image par un carré noir ($\diamond$). Lorsqu'il y a superposition des deux, c'est le centre de gravité qui est mis au premier plan.

Il apparaît clairement que la poursuite est bien réalisé, en particulier pendant les phases de mouvement à vitesse constante et pendant les arrêts. Les dépassements lors des changements de vitesse sont rapidement compensés. La poursuite est robuste aux

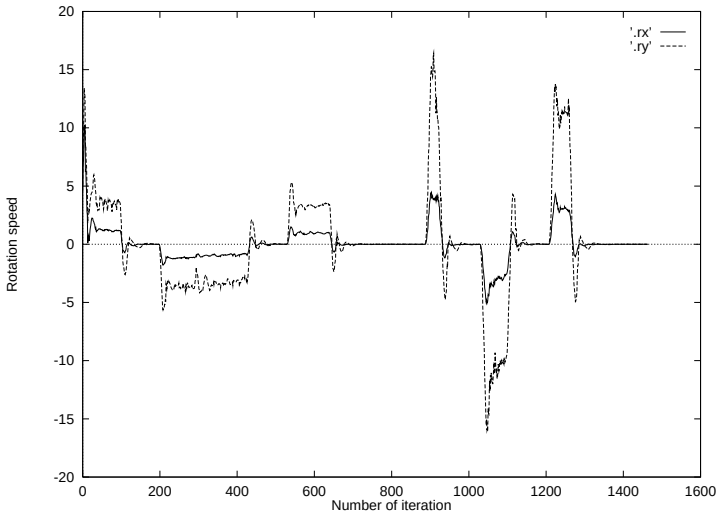


FIG. 2.13 – *Suivi de l'objet "réel". Vitesses de rotation Ω_x et Ω_y calculées par la loi de commande (en deg/s)*

mouvements de perturbations qui apparaissent derrière l'objet d'intérêt, généré par le déplacement d'un piéton, et ce parce que le mouvement est uniquement estimé sur le masque de l'objet. On peut également remarquer que le suivi est également robuste aux différences d'illumination qui peuvent apparaître (l'objet est un peu plus brillant lorsqu'il est sur la gauche du rail, plus sombre à droite). Enfin, "à l'œil nu", on note que le centre de gravité estimé correspond au même point de l'objet.

2.3.5 Suivi de piéton

L'application précédente a été conçue pour suivre a priori n'importe quel type d'objet mobile. Cependant, plus le modèle de mouvement est en adéquation avec le mouvement réel de l'objet, et plus, évidemment, la qualité du suivi sera optimale. En particulier, dans le cas d'un objet planaire et rigide, le modèle quadratique de mouvement est en théorie exact, et on peut considérer que, lorsque l'objet d'intérêt est en translation orthogonalement à l'axe optique le modèle constant en est une bonne approximation. Dans le cas d'un objet réellement tridimensionnel et déformable, le modèle constant est forcément une approximation beaucoup plus grossière du mouvement réel.

Cependant, nous avons souhaité testé les limites de notre algorithme en considérant un objet plus complexe que celui choisi dans le paragraphe précédent. Dans cette nouvelle experimentation, le système doit suivre un piéton. Cet "objet" remplit toutes les conditions défavorables pour l'algorithme. Un piéton est non-rigide, son mouvement, dans le cas d'aller-retour n'est pas une translation pure, sa vitesse n'est pas constante par morceaux.

Les résultats de cette expérience sont présentés sur la figure (2.15), avec, comme précédemment, une image sur dix du début de la séquence. La superposition du centre de

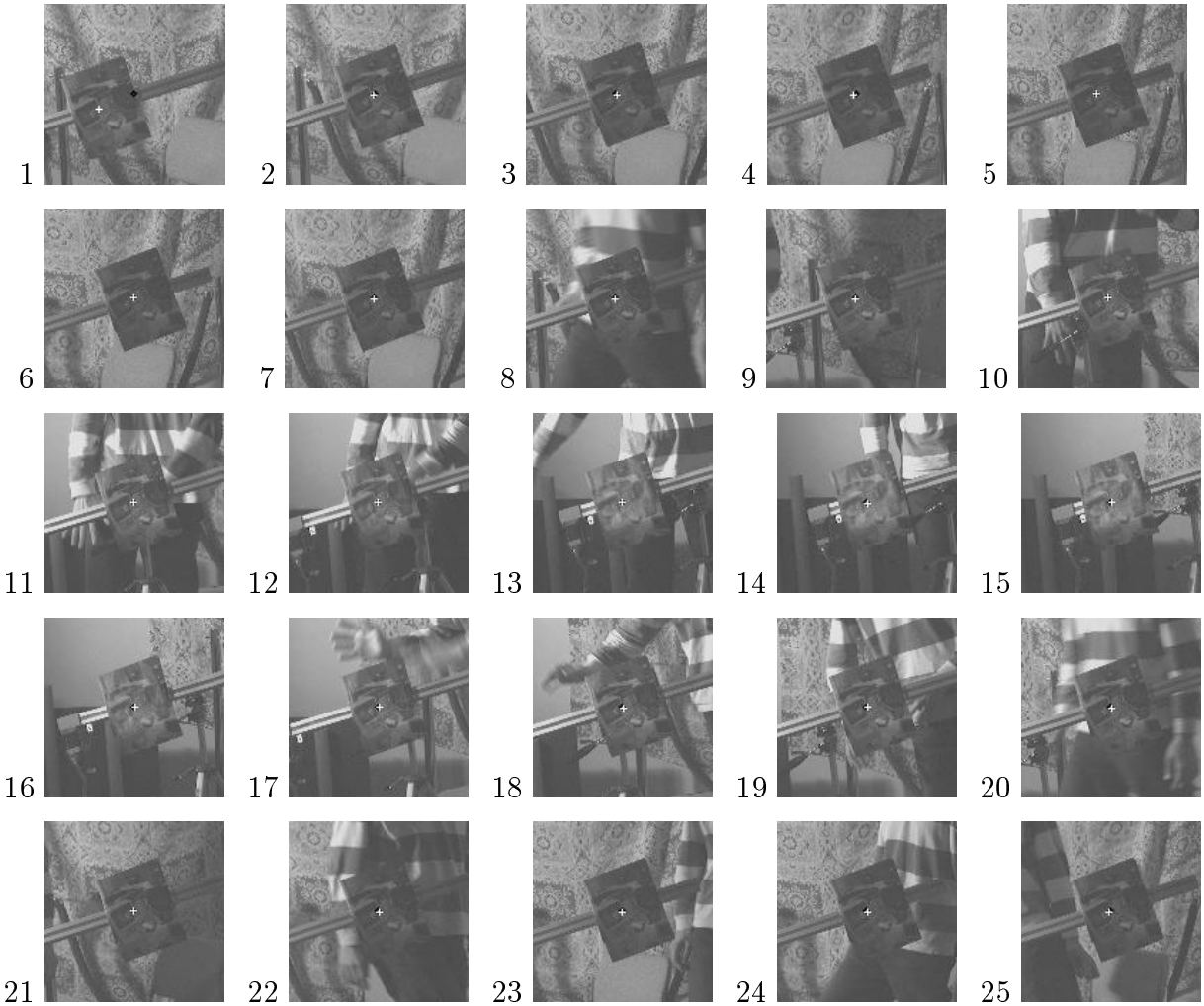


FIG. 2.14 – *Suivi de l'objet "réel". Une image sur dix de la séquence acquise, soit environ deux images par seconde. La croix (+) représente de le centre de gravité estimé, et le carré noir (◇) le centre de l'image.*

gravité estimé du piéton (représenté par une croix noire (+)) sur le centre de l'image (représenté par un carré noir(⋄)) est ici moins bonne que dans le cas précédent. L'erreur est toutefois maintenue dans un cercle d'environ 15 pixels autour du centre de l'image. Ce phénomène est facilement explicable par la complexité du mouvement du piéton et donc, la relative inadéquation du modèle considéré dans le filtrage de Kalman. Cependant, il serait assez illusoire de vouloir déterminer un filtre avec un modèle plus réaliste, ce qui de plus restreindrait considérablement le champ d'application. En fait, ce modèle à accélération constante semble être un bon compromis entre simplicité d'implémentation et applicabilité d'une part, et précision d'autre part.

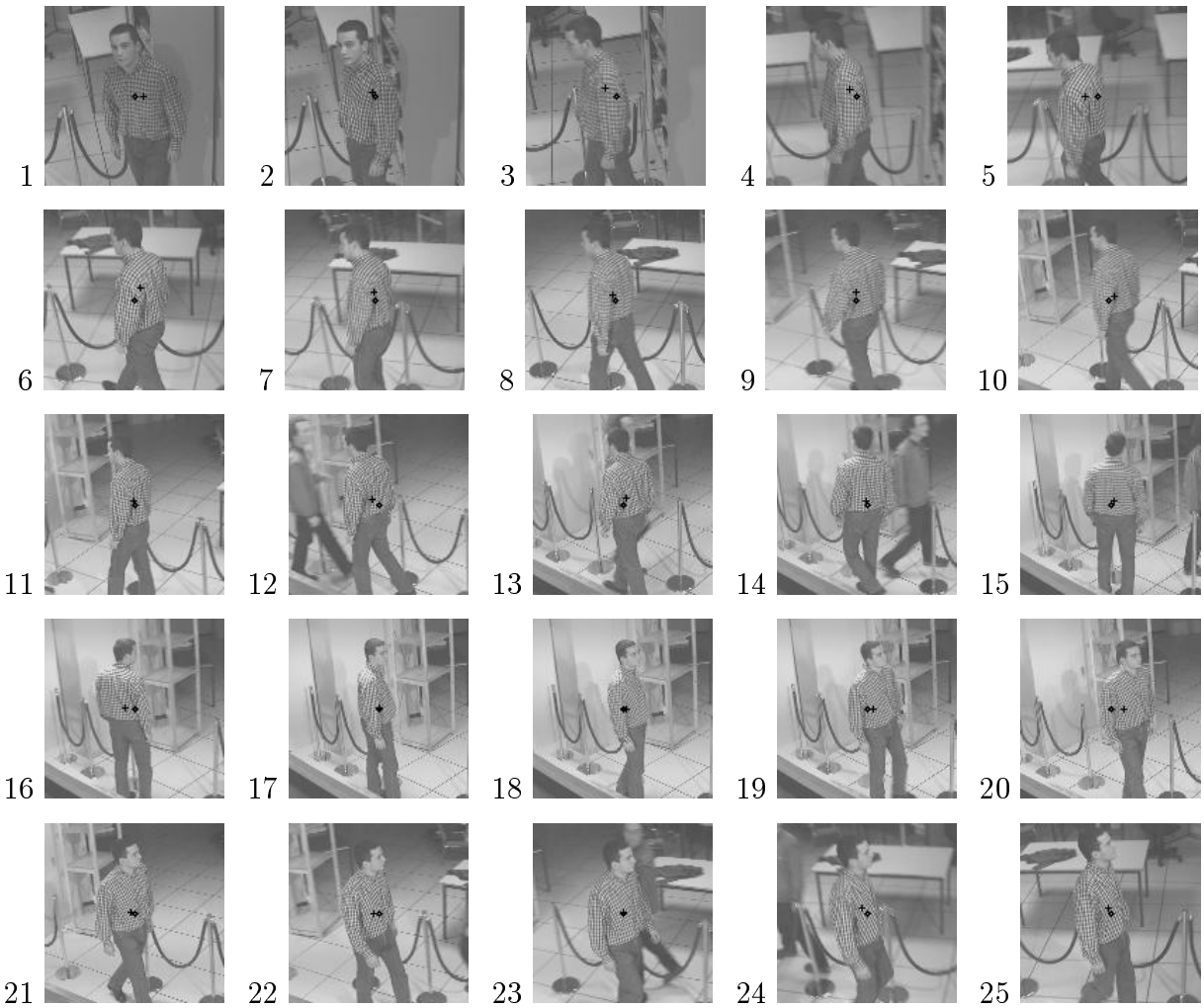


FIG. 2.15 – *Suivi de piéton. Une image sur dix de la séquence acquise, soit environ deux images par seconde. La croix (+) représente de le centre de gravité estimé, et le carré (⋄) le centre de l'image.*

Il est assez difficile de discuter de la qualité de l'estimation de la position du centre de gravité, dans le sens où on cherche à déterminer si le centre de gravité initial reste le même au cours du temps. En effet, ce point est plusieurs fois caché pendant la séquence, en particulier lorsque le piéton a effectué un demi-tour. Cependant, il n'est pas faux d'affirmer que le centre de gravité estimé à chaque itération correspond assez bien à celui qui pourrait être déterminé par des critères psycho-visuels. Une seconde remarque est que cet algorithme de suivi est finalement beaucoup plus robuste que ceux qui consistent à détecter sur chaque image un point d'intérêt déterminé au préalable. De tels algorithmes se révéleraient incapables de gérer le problème d'occultation lors du demi-tour, alors que même pendant cette phase relativement délicate, à cause des déformations propres de l'"objet", notre algorithme se montre très stable (images 13 à 17). Cela valide la robustesse de l'algorithme RMRm, en particulier le rejet des mouvements parasites lors de l'estimation du mouvement dominant.

Malgré les bons résultats obtenus, l'algorithme a tout de même montré ses limites quand l'"objet" subit de trop grandes déformations. En particulier, lorsque le piéton s'accroupit et donc "se tasse", l'adéquation entre le masque de détection et la région de l'image correspondant au piéton est mauvaise. Il peut alors apparaître alors une erreur d'estimation relativement importante dans la direction verticale. De même, les mouvements de recul ou d'avancée du piéton dans la direction de l'axe optique, qui produisent une déformation de type contraction ou divergence dans l'image, ne sont pas pris en compte par le modèle constant à deux paramètres.

Une application de cette approche

Cette approche a été appliquée dans le cadre d'une collaboration avec l'IFREMER présentée dans [Crétual 98b]. L'objectif de l'étude consistait à stabiliser l'acquisition d'images sous-marines par une caméra embarquée dans un engin sous-marin autonome. Lorsque cet engin est immobile, des phénomènes de courants ou de remous viennent perturber sa position en créant des mouvements de tangage et de roulis. Ainsi, si la caméra reste fixe par rapport au corps de l'appareil, il apparaît un mouvement dans l'image. En appliquant la technique décrite précédemment, et en considérant que l'objet à suivre est l'ensemble de la scène observée, dont le mouvement apparent est dû à celui de l'engin, on parvient à compenser ce mouvement par une commande de l'orientation de la caméra.

2.4 Liaison rigide avec un objet mobile

Dans ce paragraphe, on étend l'idée développée pour le suivi d'un point d'intérêt au cas de plusieurs points. En effet, ayant un modèle du mouvement dans l'image, il est possible d'estimer la vitesse apparente de chacun des points de cette image.

Considérons s un ensemble de n points de l'image, avec s_i les coordonnées réelles de

chacun de ces points sur la i-ème image d'une séquence acquise par la caméra.

$$s_i = (x_{1,i}, \ldots, x_{n,i}, y_{1,i}, \ldots, y_{n,i})^T$$

Ayant un modèle paramétrique du mouvement entre chacune des images et la position initiale s_0, il est possible de retrouver une estimation $\widehat{s}_i = (\widehat{x}_{1,i}, \ldots, \widehat{x}_{n,i}, \widehat{y}_{1,i}, \ldots, \widehat{y}_{n,i})$ de la position des points en utilisant la relation (2.4) et, pour un modèle affine de mouvement par exemple, l'approximation de leur vitesse apparente suivante:

$$\widehat{\dot{x}}_i = c_1 + a_1\widehat{x}_{i-1} + a_2\widehat{y}_{i-1}$$
$$\widehat{\dot{y}}_i = c_2 + a_3\widehat{x}_{i-1} + a_4\widehat{y}_{i-1}$$

Il est alors possible d'appliquer des lois de commande conçues pour le suivi de plusieurs points significatifs de l'objet.

2.4.1 Loi de commande

La notion de suivi rigide, qui découle directement du principe de liaison rigide virtuelle de [Espiau 92] appliquée à un objet mobile, consiste à maintenir constante la position relative tridimensionnelle entre la caméra et l'objet qu'elle observe. Pour cela, l'ensemble des mouvements de la caméra doit être commandé, c'est-à-dire trois translations et trois rotations. A priori, le maintien de trois points non-alignés de l'image dans la même position permet d'assurer cette liaison. Cependant, il a été prouvé que dans ce cas, le système peut passer par des singularités, en particulier lorsque le centre optique appartient au cylindre défini par le cercle circonscrit à ces trois points, et de direction orthogonale au plan de ce cercle (cf. [Michel 93]). De plus, à une position donnée des trois points dans l'image, correspond quatre positions de la caméra dans l'espace. C'est pourquoi on utilise généralement une information visuelle redondante avec au moins quatre points, même si dans ce cas-là, le système peut converger vers des minima locaux, à partir de positions initiales très éloignées (cf. [Chaumette 98] pour de plus amples détails). Pour notre application, nous travaillons avec quatre points ; dans ce cas, il n'y pas de risque de convergence vers un minimum local, vu que l'on reste toujours au voisinage de la position désirée.

Soit $\widehat{s}$ le vecteur des positions reconstruites, liées aux mouvements de la caméra par la relation :

$$\dot{\widehat{s}} = \underbrace{\begin{bmatrix} -1/\widehat{Z_1} & 0 & \widehat{x}_1 & \widehat{x}_1\widehat{y}_1 & -1-\widehat{x}_1^2 & \widehat{y}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1/\widehat{Z_4} & 0 & \widehat{x}_4 & \widehat{x}_4\widehat{y}_1 & -1-\widehat{x}_4^2 & \widehat{y}_4 \\ 0 & -1/\widehat{Z_1} & \widehat{y}_1 & 1+\widehat{y}_1^2 & -\widehat{x}_1\widehat{y}_1 & -\widehat{x}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -1/\widehat{Z_4} & \widehat{y}_4 & 1+\widehat{y}_4^2 & -\widehat{x}_4\widehat{y}_4 & -\widehat{x}_4 \end{bmatrix}}_{L} \begin{pmatrix} T_x \\ T_y \\ T_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} + \frac{\widehat{\partial s}}{\partial t}$$

où $\widehat{Z}_i$ désigne une estimation de la profondeur Z_i du point 3D correspondant au point $(\widehat{x}_i, \widehat{y}_i)$ de l'image, et (T, Ω) est le torseur cinématique de la caméra. Comme dans la relation (2.5), le terme $\frac{\widehat{\partial s}}{\partial t}$ représente la contribution du mouvement propre de l'objet.

Les points d'intérêt, virtuels, sont désignés sur la première image par l'utilisateur. Ce choix de s_0 peut a priori être quelconque. En particulier, il n'est pas nécessaire que les points en question appartiennent à l'objet devant être suivi. Il n'y a pas de correspondance entre les points virtuels d'intérêt et les points 3D de la scène qui sont désignés au départ. On ne cherche pas à suivre les points initiaux mais simplement à donner les positions initiales qui seront mises à jour par application du modèle de mouvement de l'objet suivi. Cependant, afin d'assurer une observation suffisante des déformations dans l'image, il est préférable de respecter les deux conditions suivantes : premièrement, que les points soient suffisament éloignés les uns des autres et deuxièmement, qu'il n'y ait pas de triplet de points alignés. Par exemple, sur la figure (2.16), les croix blanches représentent un choix de quatre points pouvant être retenu pour une poursuite rigide de l'objet présenté sur la figure (2.8). On peut noter en particulier que ces points ne correspondent pas forcément à des points significatifs de l'objet (c'est-à-dire, par exemple, avec des gradients spatiaux élévés dans deux directions), qui pourraient éventuellement être suivis par un algorithme de recherche de points singuliers.

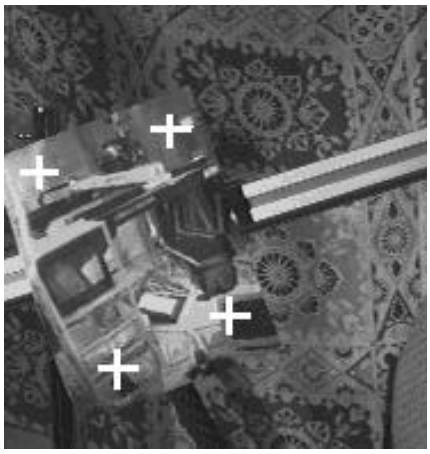


FIG. 2.16 – *Choix de quatre points d'intérêt virtuels pour une poursuite rigide du carré*

La position désirée s^* pour chacun des points est celle qu'ils avaient sur l'image initiale, c'est-à-dire $s^* = s_0$. Le vecteur d'erreur associé à la tâche est alors défini par $e = C(\widehat{s} - s_0)$ où C est une matrice constante. De la même manière que la relation (2.6) a été établie, une décroissance exponentielle de l'erreur ($\dot{e} = -\lambda e$) conduit à la commande :

$$\begin{pmatrix} T \\ \Omega \end{pmatrix} = -C^+ \widehat{L}^+ (\lambda \widehat{e} + \widehat{\frac{\partial e}{\partial t}})$$

où :

- $\widehat{e}$ désigne une mesure ou une estimation de e
- $\widehat{L}$ représente une approximation de L correspondant à sa valeur à la convergence dans laquelle les Z_i sont estimés visuellement. Dans la pratique, une même valeur est attribuée pour les quatre profondeurs Z_i. Enfin, $\widehat{L}^+$ est la pseudo-inverse de cette approximation et C est choisie égale à $\widehat{L}$. La loi de commande se réduit donc à :

$$\begin{pmatrix} T \\ \Omega \end{pmatrix} = -\lambda \widehat{e} - \widehat{\frac{\partial e}{\partial t}}$$

- $\widehat{\frac{\partial e}{\partial t}}$ est, comme dans le cas de la commande pour le suivi d'un seul point, une estimation de la variation dans l'image due au mouvement propre de l'objet, donnée par (cf. [Chaumette 93]) :

$$\widehat{\frac{\partial e}{\partial t}} = C\widehat{\frac{\partial s}{\partial t}} = C\dot{\widehat{s}} - \begin{pmatrix} \widehat{T} \\ \widehat{\Omega} \end{pmatrix}$$

où $\widehat{T}$ et $\widehat{\Omega}$ représentent respectivement la mesure par odométrie de la vitesse de translation et de rotation de la caméra. Cette estimation de la contribution du mouvement de l'objet au mouvement apparent dans l'image est également filtrée par un filtre de Kalman avec un modèle à accélération constante et bruit corrélé.

2.4.2 Simulations

La tâche de suivi rigide a été simulée dans un contexte similaire à celui utilisé pour le cas du suivi d'un seul point (cf. paragraphe 2.3.3), à savoir que la distance entre l'objet et la caméra est de 1 m, que λ vaut 1,5 et que l'ordre de grandeur des mouvements est de 20 à 30 cm/s pour les translations et 5 à 12 deg/s pour les rotations. La cible considérée est planaire mais n'est pas parallèle au plan de projection de la caméra, ce qui implique des erreurs dans le choix qui a été fait pour L. L'orientation est choisie de 12 degrés autour de l'axe x, 18 autour de l'axe y. Enfin, les quatre points considérés sont positionnés sur la première image aux coordonnées, exprimées en mètre, (-0.2, -0.1), (-0.2, 0.1), (0.2, 0.1) et (0.2, -0.1).

Dans un premier temps, et pour avoir une référence permettant de discuter des performances de notre algorithme, le suivi rigide est simulé en utilisant directement les vraies positions des quatre points considérés. Ensuite, la tâche est simulée en reconstruisant ces positions. Le modèle de mouvement utilisé n'est pas le modèle constant, comme dans le cas du suivi en *pan* et *tilt*, car il est évidemment trop pauvre pour estimer finement la vitesse apparente de chacun des points, en conduisant à un système sous-dimensionné. Deux modèles différents ont donc été utilisés : tout d'abord le modèle affine, puis le modèle quadratique simplifié à huit paramètres. Dans chacun des cas (positions estimées ou

non), deux cas de mouvements sont considérés : un mouvement composé uniquement de translations dans le repère fixe (cf. figure (2.17) a), et un mouvement complexe composé à la fois des translations du mouvement précédent et de rotations (cf. figure (2.17) b). Dans chacun de ces cas, les vitesses sont constantes par morceaux. Enfin, les simulations ont été effectuées en considérant les mesures non-bruitées.

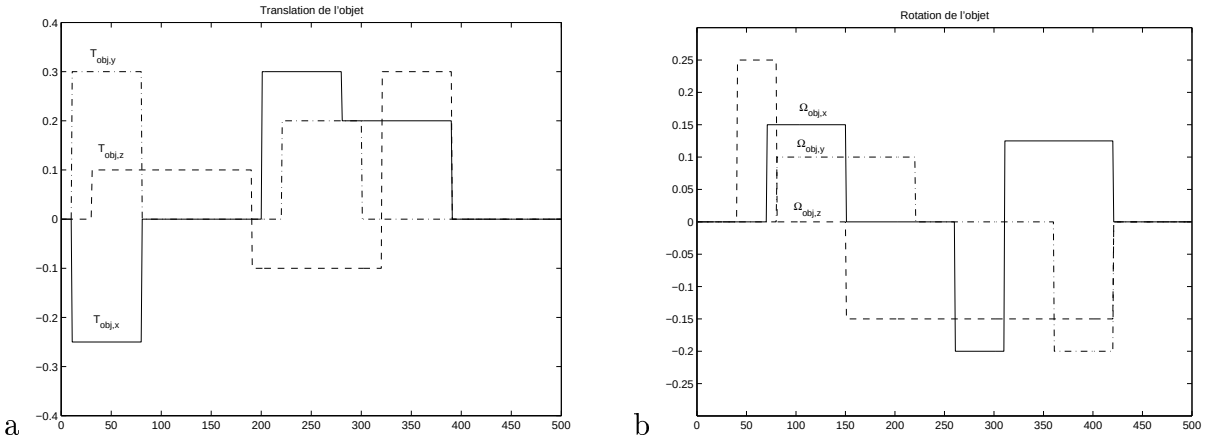


FIG. 2.17 – *Vitesses de translation (a) et de rotation (b) de l'objet dans le repère fixe.* $T_{obj,x}$ *et* $\Omega_{obj,x}$ *en trait plein (—),* $T_{obj,y}$ *et* $\Omega_{obj,y}$ *en trait discontinu (– · –),* $T_{obj,z}$ *et* $\Omega_{obj,z}$ *en trait pointillé (- - -)*

Dans chacune des expériences de simulation de ce paragraphe, quatre courbes sont présentées :

- les vitesses de translation et de rotation de la caméra dans son repère, données respectivement en m/s et rad/s ;
- l'abscisse et l'ordonnée vraies d'un des quatre points considérés. On ne donne que les résultats concernant un point, car le comportement est similaire pour les quatre points.

De plus, pour chacune des expériences dans laquelle les positions sont estimées, ces courbes sont complétées par :

- l'abscisse et l'ordonnée estimées d'un des quatre points considérés, respectivement superposée avec la vraie valeur qui lui correspond
- les erreurs d'estimation sur l'axe des abscisses et celui des ordonnées

Utilisation de la position réelle

Il apparaît nettement sur les figures (2.18), (2.19), (2.20) et (2.21) que, lorsque les positions réelles sont utilisées dans la boucle de commande, le suivi rigide est bien réalisé.

Seuls quelques dépassements dus au temps de réaction du système lors des changements du mouvement de l'objet sont observés, et ceux-ci sont très faibles par rapport à la taille de l'image.

Utilisation de la position réelle des points. Déplacement de l'objet en translation

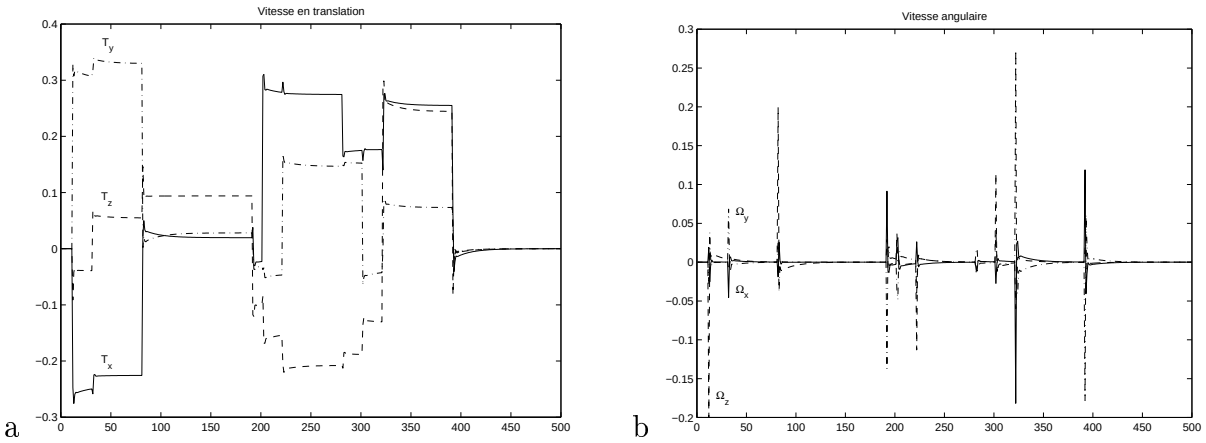


FIG. 2.18 – *Vitesses de translation (a) et de rotation (b) commandées. T_x et Ω_x en trait plein (—), T_y et Ω_y en trait pointillé (- - -), T_z et Ω_z en trait discontinu (– · –)*

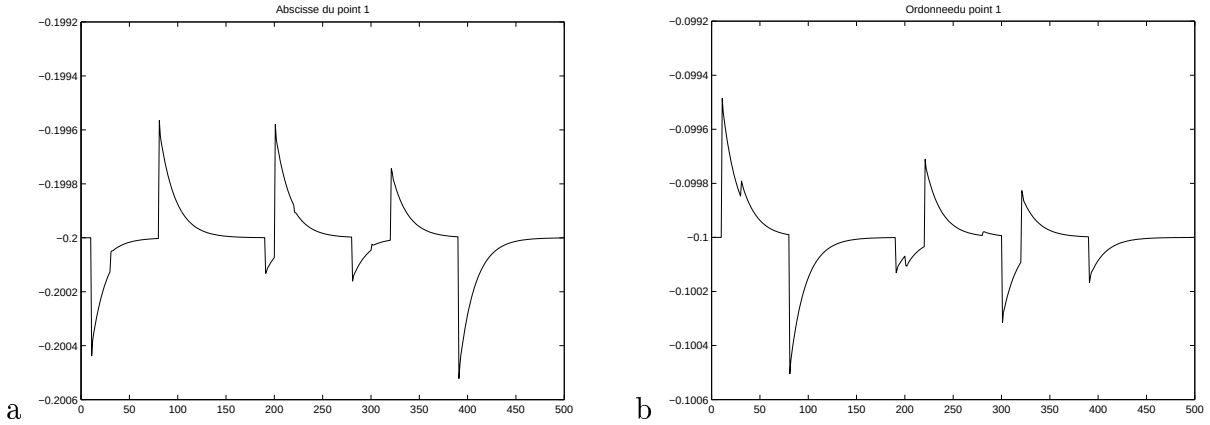


FIG. 2.19 – *Abscisse (a) et ordonnée (b) d'un des points considérés.*

Que le mouvement de l'objet soit une série de translations pures ou qu'il soit plus complexe, les vitesses de translation commandées de la caméra correspondent bien à celles de l'objet, en particulier pendant les phases stationnaires. En ce qui concerne les rotations, si elles sont très bien suivies lors des mouvements complexes, elles devraient

Utilisation de la position réelle des points. Déplacement de l'objet en translation et rotation

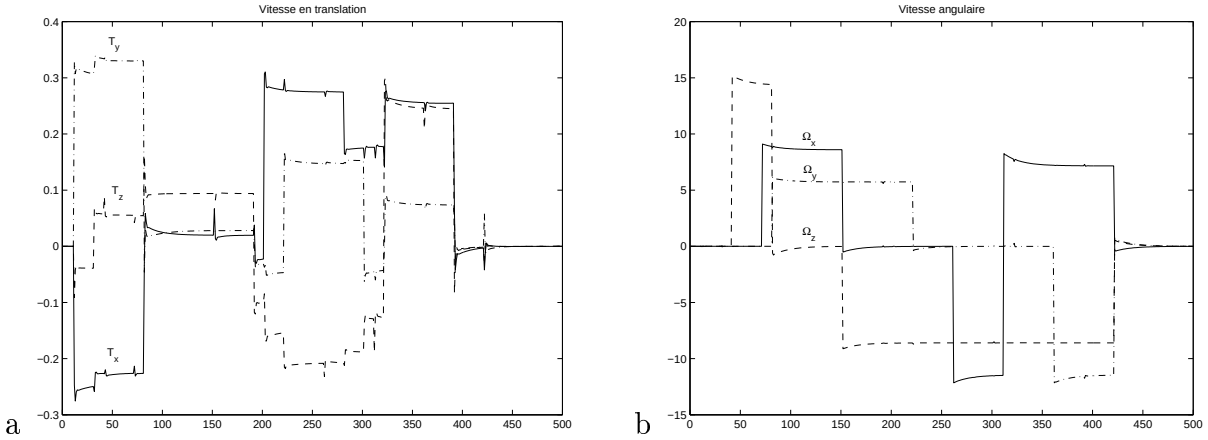


FIG. 2.20 – *Vitesses de translation (a) et de rotation (b) commandées. T_x et Ω_x en trait plein (——), T_y et Ω_y en trait pointillé (- - -), T_z et Ω_z en trait discontinu (– · –)*

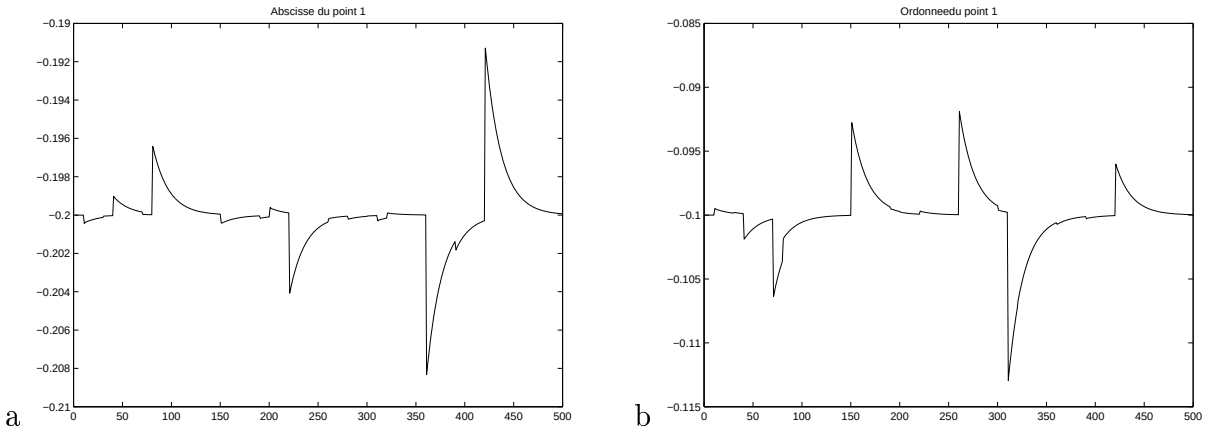


FIG. 2.21 – *Abscisse (a) et ordonnée (b) d'un des points considérés.*

être nulles dans le cas d'une translation pure de l'objet. Or, il apparaît, lors de chaque changement de la direction de translation de l'objet, des sollicitations sur chacun des axes de rotation de la caméra, même si ces vitesses reprennent très rapidement une valeur nulle. Ce phénomène est dû à l'approximation faite dans la matrice d'interaction L par sa valeur à la convergence. En effet, lors des changements du mouvement de l'objet, les points ne se trouvent plus à leurs positions désirées, et c'est alors que l'erreur d'approximation sur la matrice d'interaction est la plus forte. Les trois composantes du vecteur d'erreur e qui correspondent à une rotation relative de l'objet par rapport à la caméra lors du calcul $C(s - s^*)$ sont biaisées et donc non-nulles. Ce biais est rapidement éliminé dès que la commande en translation ramène l'objet dans la position désirée.

Estimation de la position avec un modèle affine de mouvement

Dans ce paragraphe, l'estimation de la position des points d'intérêt est effectuée avec une approximation de leur vitesse apparente par le modèle affine du mouvement. Nous verrons que, même si ce modèle possède six paramètres et qu'également, six degrés de liberté du robot sont contraints par la loi de commande, le suivi n'est pas correctement réalisé. Cela est dû au fait que le modèle affine ne permet pas de détecter le mouvement apparent correspondant aux rotations autour des deux axes orthogonaux à l'axe optique.

Dans la première expérience, le mouvement de l'objet se réduit à des translations dans le repère fixe. Les commandes calculées sont présentées sur la figure (2.22), la comparaison de la position réelle et estimée d'un des points considérés sur la figure (2.23) et enfin, les erreurs d'estimation pour le même point sur la figure (2.24).

Les résultats obtenus dans ce cas sont satisfaisants puisque les points sont correctement ramenés à leurs positions initiales après chaque perturbation due à un changement brusque du déplacement de l'objet (cf. figure (2.23)), et ce, même si la position réelle de ces points n'est pas utilisée dans la commande. Ce résultat est justifié et confirmé par le fait que l'erreur d'estimation est très faible et reste toujours inférieure à 10^{-2}mm (cf. figure (2.24)) (cela explique pourquoi, à l'échelle choisie, les deux courbes correspondant respectivement aux positions vraie et estimée sont indissociables sur la figure (2.23)). En effet, la loi de commande considérée est très robuste pour le suivi rigide lorsque les positions réelles sont utilisées; par conséquent si l'estimation est très fine, le suivi est également correctement réalisé.

L'erreur d'estimation est nettement plus importante, même si elle reste faible, lors des changements de mouvement de l'objet. Cela s'explique par la différence existant alors entre la position considérée pour estimer la vitesse, c'est-à-dire la position atteinte à l'itération précédente, et la position réelle. L'estimation de la vitesse étant biaisée, l'estimation de la position subit ce biais. Cependant, lorsque le système rejoint une phase stationnaire, correspondant à l'hypothése d'une accélération constante (dans ce cas nulle) considérée

Estimation de la position des points avec un modèle affine du mouvement 2D. Déplacement de l'objet en translation

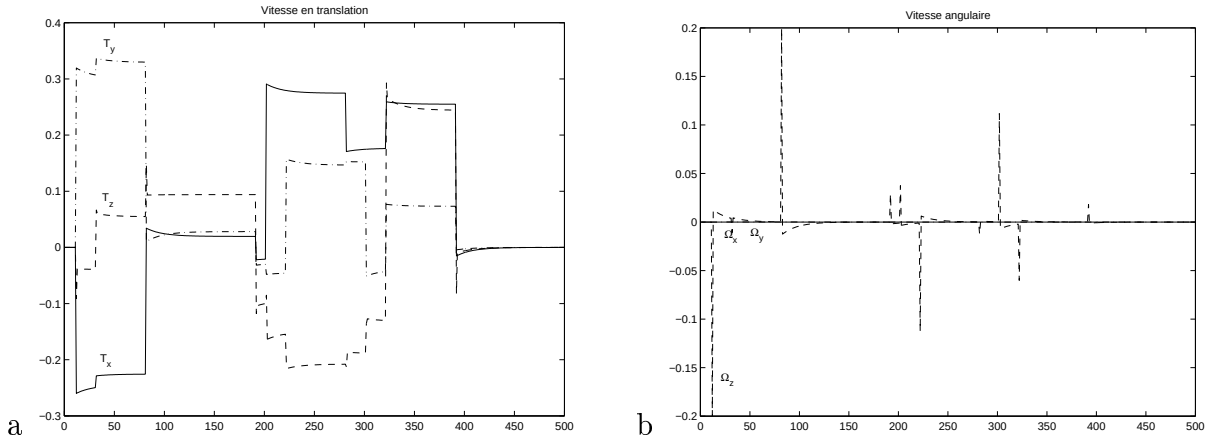


FIG. 2.22 – *Vitesses de translation (a) et de rotation (b) commandées. T_x et Ω_x en trait plein (—), T_y et Ω_y en trait pointillé (- - -), T_z et Ω_z en trait discontinu (– · –)*

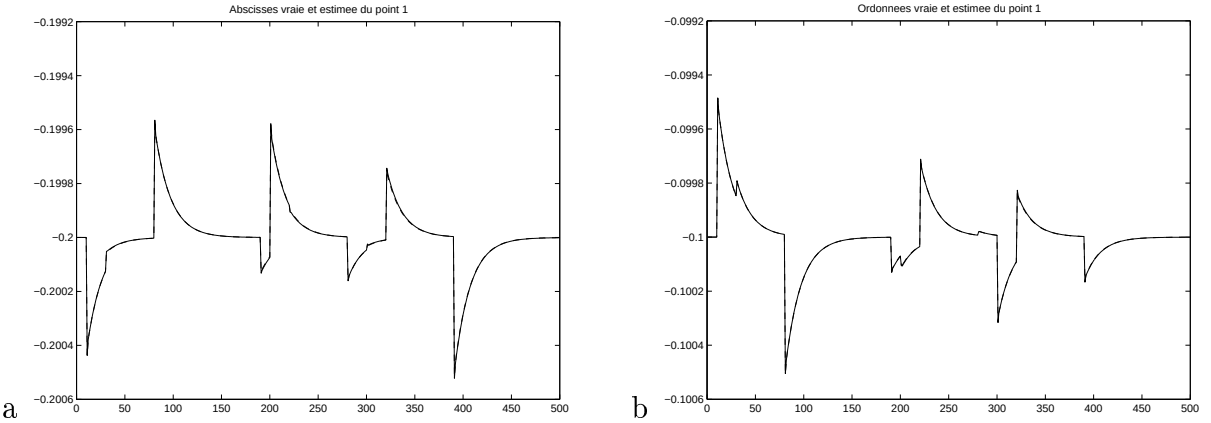


FIG. 2.23 – *Abscisse (a) et ordonnée (b) d'un des points considérés. Les valeurs estimées sont en trait plein (—), les vraies valeurs en trait pointillé (- - -)*

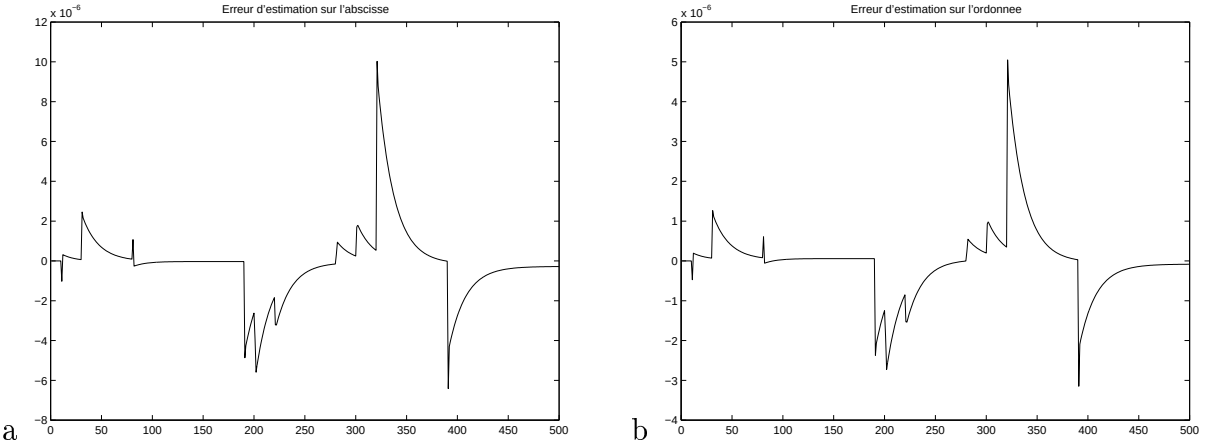


FIG. 2.24 – *Erreur d'estimation en abscisse (a) et ordonnée (b) du même point*

dans le modèle du filtre, l'erreur d'estimation est compensée.

Enfin, le phénomène qui apparaissait sur la commande en rotation en utilisant les vraies positions, ne se retrouve ici que dans la rotation autour de l'axe optique. La sollicitation "parasite" autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$ est ici nulle. L'approximation faite en ne considérant que la partie affine du mouvement, semble donc compenser exactement l'erreur générée par l'utilisation d'une matrice d'interaction approchée.

Dans la seconde expérience, les mouvements de l'objet se composent à la fois de translations et de rotations constantes par morceaux dans le repère fixe. Les commandes calculées sont présentées sur la figure (2.25), la comparaison de la position réelle et estimée d'un des points considérés sur la figure (2.26) et enfin, les erreurs d'estimation pour le même point sur la figure (2.27).

Dans ce cas, les résultats sont nettement moins bon que lorsque l'objet ne se déplace qu'en translation. En particulier, le système s'avère incapable de ramener les positions estimées à leurs valeurs désirées (cf. figure (2.26)). La convergence semble plutôt s'établir en direction d'un minimum local. De plus, les erreurs d'estimation sont élevées puisqu'elles atteignent des valeurs de l'ordre de 2 cm. En effet, les translations de la caméra sont nettement plus élevées que celles de l'objet (de l'ordre de 2 à 5 m/s). De plus, même lorsque l'objet se déplace également en rotation, aucune sollicitation n'est calculée autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$ du repère de la caméra. Nous expliquons, ci-dessous, les raisons de ce phénomène.

Considérons l'espace vectoriel d'évolution du vecteur $(\widehat{s} - s_0)$ lorsque le système est proche de la position de convergence. Cet espace de dimension au plus 6 est déterminé par une transformation linéaire du torseur cinématique du mouvement relatif de l'objet par rapport à la caméra. Ainsi, lorsque $\widehat{s} = s^*$ et que survient un mouvement de l'objet correspondant à un torseur cinématique (T_o, Ω_o), le vecteur $(\widehat{s} - s_0)$ devient (car on a un modèle affine) :

$$\widehat{s} - s_0 = \underbrace{\begin{bmatrix} (1-\gamma_1 x_1 - \gamma_2 y_1)/Z_p & 0 & -x_1/Z_p & 0 & 1 & -y_1 \\ (1-\gamma_1 x_2 - \gamma_2 y_2)/Z_p & 0 & -x_2/Z_p & 0 & 1 & -y_2 \\ (1-\gamma_1 x_3 - \gamma_2 y_3)/Z_p & 0 & -x_3/Z_p & 0 & 1 & -y_3 \\ (1-\gamma_1 x_4 - \gamma_2 y_4)/Z_p & 0 & -x_4/Z_p & 0 & 1 & -y_4 \\ 0 & (1-\gamma_1 x_1 - \gamma_2 y_1)/Z_p & -y_1/Z_p & -1 & 0 & x_1 \\ 0 & (1-\gamma_1 x_2 - \gamma_2 y_2)/Z_p & -y_2/Z_p & -1 & 0 & x_2 \\ 0 & (1-\gamma_1 x_3 - \gamma_2 y_3)/Z_p & -y_3/Z_p & -1 & 0 & x_3 \\ 0 & (1-\gamma_1 x_4 - \gamma_2 y_4)/Z_p & -y_4/Z_p & -1 & 0 & x_4 \end{bmatrix}}_{M} \begin{pmatrix} T_{o,x} \\ T_{o,y} \\ T_{o,z} \\ \Omega_{o,x} \\ \Omega_{o,y} \\ \Omega_{o,z} \end{pmatrix}$$

avec $Z = Z_p + \gamma_1 X + \gamma_2 Y$ l'équation du plan de la cible dans le repère de la caméra.

La matrice M est de rang 6, pour peu que les points considérés sur l'image soient correctement choisis, en particulier non-alignés. Cependant, l'espace de dimension 6 dans

Estimation de la position des points avec un modèle affine du mouvement 2D. Déplacement de l'objet en translation et rotation

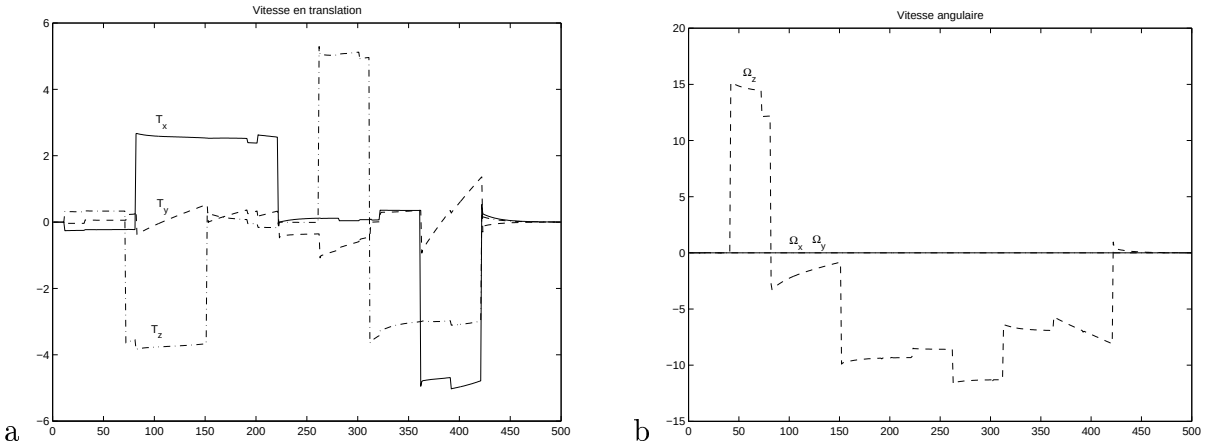


FIG. 2.25 – *Vitesses de translation (a) et de rotation (b) commandées. T_x et Ω_x en trait plein (—), T_y et Ω_y en trait pointillé (- - -), T_z et Ω_z en trait discontinu (– · –)*

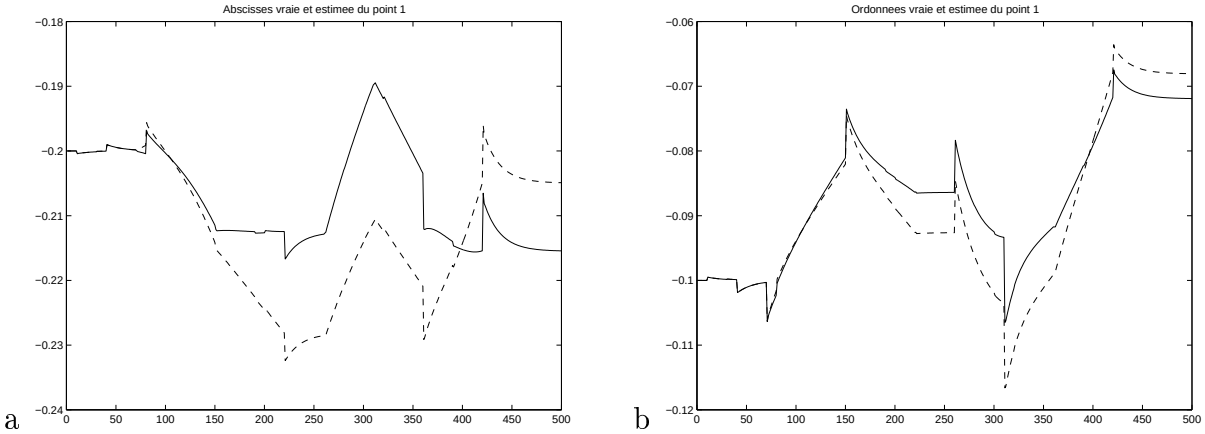


FIG. 2.26 – *Abscisse (a) et ordonnée (b) d'un des points considérés. Les valeurs estimées sont en trait plein (—), les vraies valeurs en trait pointillé (- - -)*

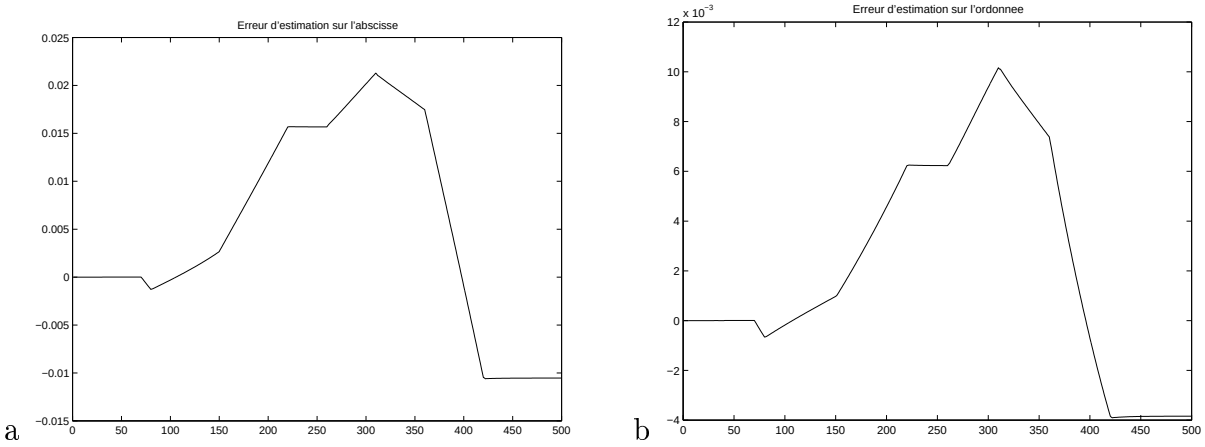


FIG. 2.27 – *Erreur d'estimation en abscisse (a) et ordonnée (b) du même point*

lequel évolue $(\widehat{s} - s_0)$ est différent de l'espace, également de dimension 6, d'évolution du vecteur $(s - s_0)$ correspondant aux vraies positions des points de contrôle. Le vecteur d'erreur e est donné par :

$$e = C(s - s_0) = (LL^T)^{-1} L^T M (T_o, \Omega_o)^T$$

Le produit de matrices $(L^T M)$ est de rang 4, ce qui implique une perte de l'information de déformation de la cible et se traduit par l'inobservabilité des variations de position des points due au mouvement de rotation autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$ du repère de la caméra. Il n'est pas possible de montrer simplement de façon analytique que le rang du produit $(L^T M)$ est 4, mais considérons les valeurs de ces deux matrices dans le cas particulier que nous avons étudié pour nos simulations :

$$L^T = \begin{bmatrix} -0.04 & -0.04 & -0.04 & -0.04 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.04 & -0.04 & -0.04 & -0.04 \\ -0.008 & -0.008 & 0.008 & 0.008 & -0.004 & 0.004 & 0.004 & -0.004 \\ 0.02 & -0.02 & 0.02 & -0.02 & 1.01 & 1.01 & 1.01 & 1.01 \\ -1.04 & -1.04 & -1.04 & -1.04 & -0.02 & 0.02 & -0.02 & 0.02 \\ -0.1 & 0.1 & 0.1 & -0.1 & 0.2 & 0.2 & -0.2 & -0.2 \end{bmatrix}$$

$$M = \begin{bmatrix} 0.0432 & 0 & 0.008 & 0 & 1 & 0.1 \\ 0.0416 & 0 & 0.008 & 0 & 1 & -0.1 \\ 0.0368 & 0 & -0.008 & 0 & 1 & -0.1 \\ 0.0384 & 0 & -0.008 & 0 & 1 & 0.1 \\ 0 & 0.0432 & 0.004 & -1 & 0 & -0.2 \\ 0 & 0.0416 & -0.004 & -1 & 0 & -0.2 \\ 0 & 0.0368 & -0.004 & -1 & 0 & 0.2 \\ 0 & 0.0384 & 0.004 & -1 & 0 & 0.2 \end{bmatrix}$$

Les valeurs singulières S_{aff} du produit $(L^T M)$ sont :

$$S_{aff} = (4.1664,\ 4.0464,\ 0.16,\ 2.6\ 10^{-4},\ 3.02\ 10^{-18},\ 0) \tag{2.7}$$

ce qui revient à dire, en tenant compte des erreurs de précision numérique, que deux de ces valeurs singulières sont nulles.

Le noyau de $(L^T M)$ est défini par l'espace vectoriel engendré par les deux vecteurs V_1 et V_2 suivants :

$$V_1 = (\underline{0.9717}, -0.0094, -0.2328, -0.0004, \underline{-0.0389}, -0.0016)^T$$
$$V_2 = (-0.0001, \underline{-0.9984}, 0.04, \underline{-0.0399}, 5.21\ 10^{-6}, -0.0096)^T$$

Si l'on s'intéresse aux termes significatifs en translation et en rotation (les valeurs soulignées) dans chacun de ces vecteurs, on peut noter que le noyau est constitué d'un vecteur V_1 qui lie v_x à Ω_y et d'un vecteur V_2 qui lie v_y à Ω_x.

Le terme dépendant de la translation T_z dans le vecteur V_1 (-0.2338), même s'il est plus faible, n'est pas négligeable devant celui dépendant de T_x (0.9717). Cette valeur apparaît dans le noyau parce que l'influence des translations le long de l'axe optique est plus difficile à observer dans l'image que celle d'une translation le long de l'axe x. En effet, la première apparaît dans les termes affines alors que la second apparaît dans les termes constants du mouvement. De plus, cette valeur non négligeable est observée dans V_1 et pas dans V_2, simplement à cause de la configuration particulière (un rectangle orienté selon l'axe $\overrightarrow{x}$ de l'image) retenue dans cet exemple.

Estimation de la position avec un modèle quadratique de mouvement

Dans ce paragraphe, les positions des points d'intérêt sont maintenant estimées à partir de la vitesse apparente déduite d'un modèle quadratique du mouvement. Dans le cas d'un objet planaire, ce modèle est, a priori, exact. Par conséquent, nous verrons que les problèmes de déplacements de l'objet non detectés dans le cas précédent sont ici résolus.

Dans la première expérience, les mouvements de l'objet se réduisent à des translations dans le repère fixe. Les commandes calculées sont présentées sur la figure (2.28), la comparaison de la position réelle et estimée d'un des points considérés sur la figure (2.29) et enfin, les erreurs d'estimation pour le même point sur la figure (2.30).

Les résultats obtenus sont très satisfaisants dans le sens où ils sont très proches de ceux obtenus dans le cas où les vraies positions sont utilisées, que ce soit du point de vue de la convergence des coordonnées des points vers les valeurs désirées (cf figure (2.29)), que de celui des commandes en translation et rotation (cf figure (2.28)). De plus, les erreurs d'estimation sont extrêmement faibles (inférieures à 10^{-3} mm). Le modèle quadratique simplifié étant exact, dans le cas où l'objet observé est planaire, il aurait, en théorie, dû conduire à une erreur d'estimation nulle. Le phénomène qui apparaît ici est dû, non pas à une approximation dans la modélisation, mais à la discrétisation de l'estimation; elle est ici effectuée à la cadence vidéo (une itération toutes les 40 ms).

Dans la seconde expérience, les mouvements de l'objet se composent à la fois de translations et de rotations constantes par morceaux dans le repère fixe. Les commandes calculées sont présentées sur la figure (2.31), la comparaison de la position réelle et estimée d'un des points considérés sur la figure (2.32) et enfin, les erreurs d'estimation pour le même point sur la figure (2.33).

De même qu'avec des mouvements uniquement de translation de l'objet, les résultats sont ici très proches de ceux obtenus pour une commande avec les vraies valeurs des positions de chaque point. Le fait d'utiliser un modèle qui permet de distinguer l'influence sur les mouvements dans l'image des translations de l'objet le long des axes orthogonaux à l'axe optique et celle des rotations autour de ces mêmes axes permet d'éviter les erreurs

Estimation de la position des points avec un modèle quadratique du mouvement 2D. Déplacement de l'objet en translation

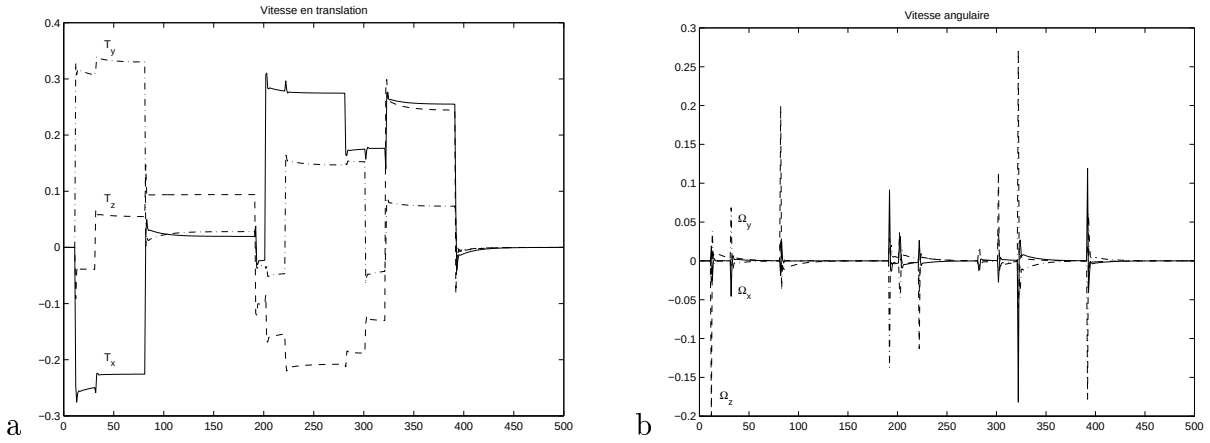


FIG. 2.28 – *Vitesses de translation (a) et de rotation (b) commandées. T_x et Ω_x en trait plein (—), T_y et Ω_y en trait pointillé (- - -), T_z et Ω_z en trait discontinu (– · –)*

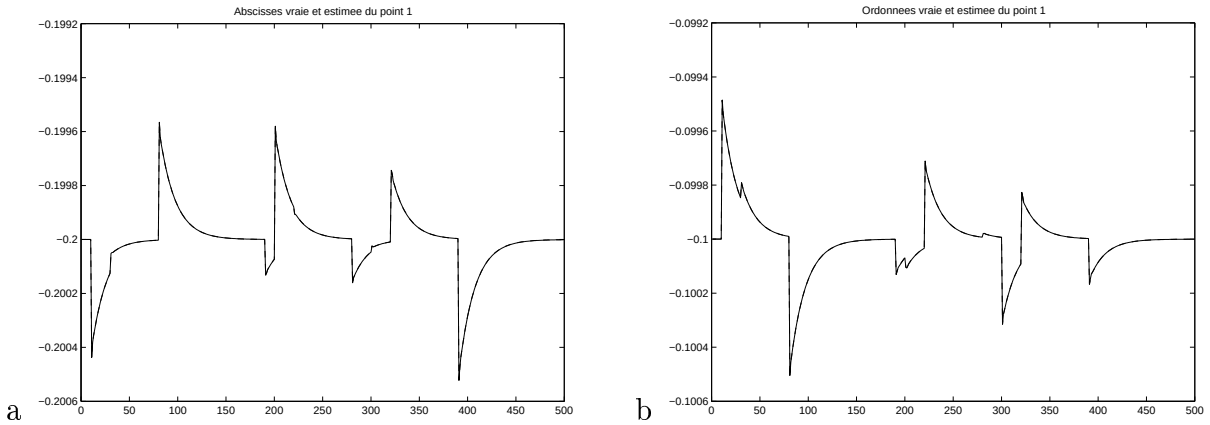


FIG. 2.29 – *Abscisse (a) et ordonnée (b) d'un des points considérés. Les valeurs estimées sont en trait plein (—), les vraies valeurs en trait pointillé (- - -)*

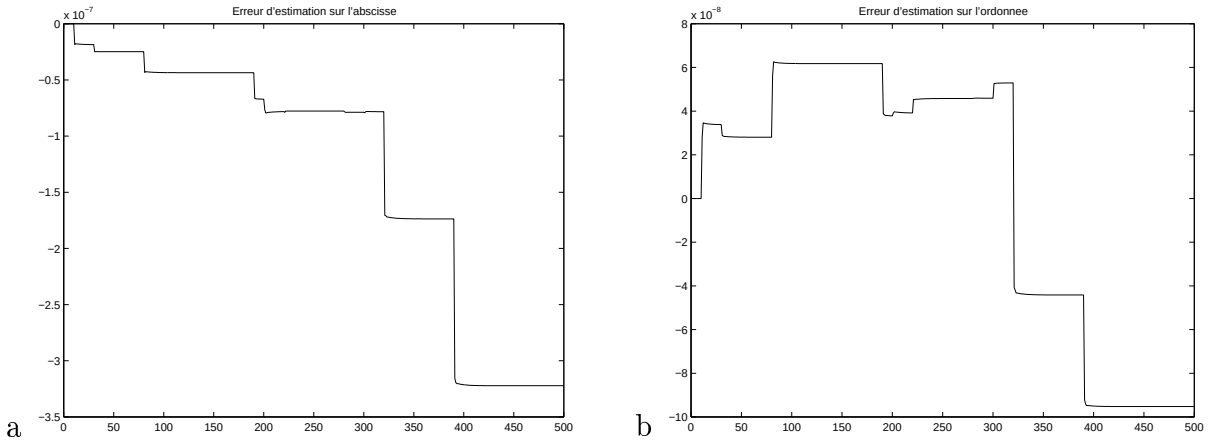


FIG. 2.30 – *Erreur d'estimation en abscisse (a) et ordonnée (b) du même point*

Estimation de la position des points avec un modèle quadratique du mouvement 2D. Déplacement de l'objet en translation et rotation

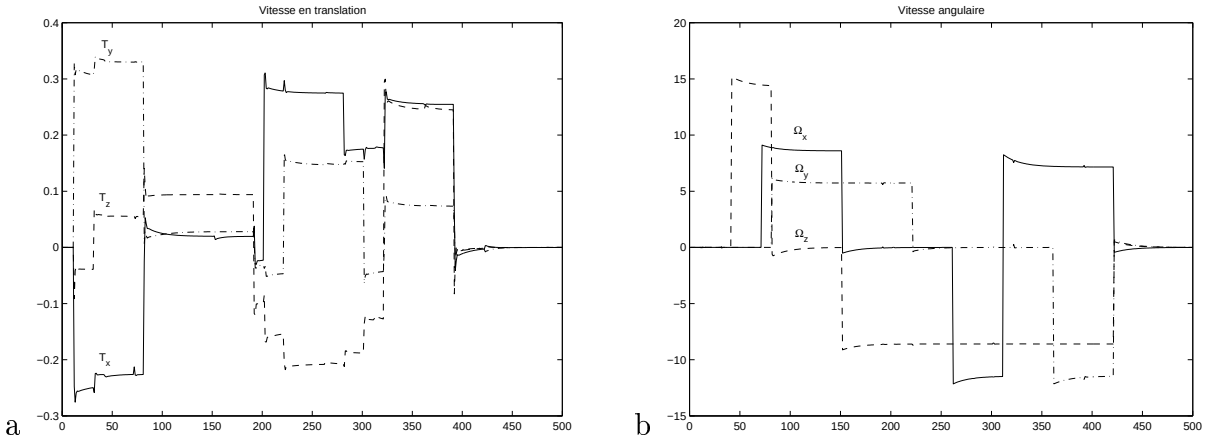


FIG. 2.31 – *Vitesses de translation (a) et de rotation (b) commandées. T_x et Ω_x en trait plein (—), T_y et Ω_y en trait pointillé (- - -), T_z et Ω_z en trait discontinu (– · –)*

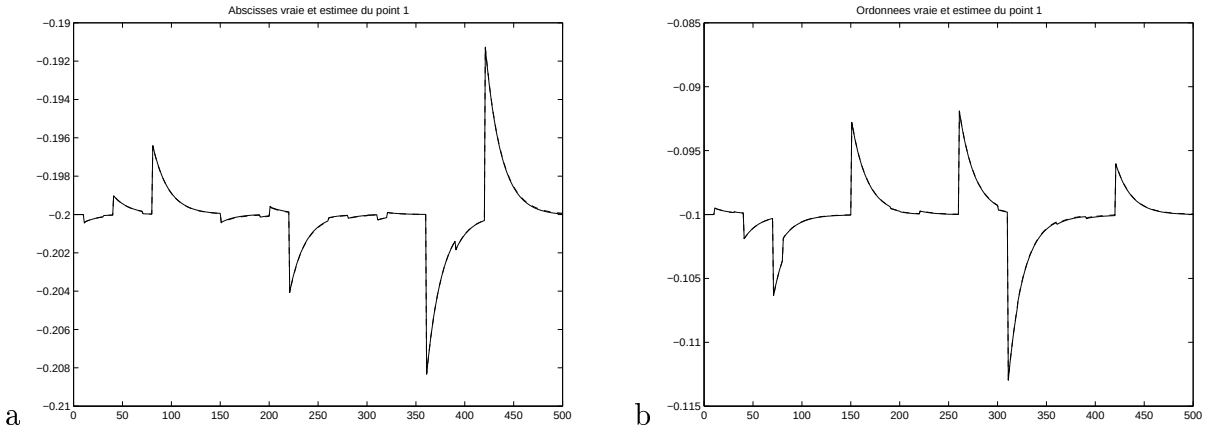


FIG. 2.32 – *Abscisse (a) et ordonnée (b) d'un des points considérés. Les valeurs estimées sont en trait plein (—), les vraies valeurs en trait pointillé (- - -)*

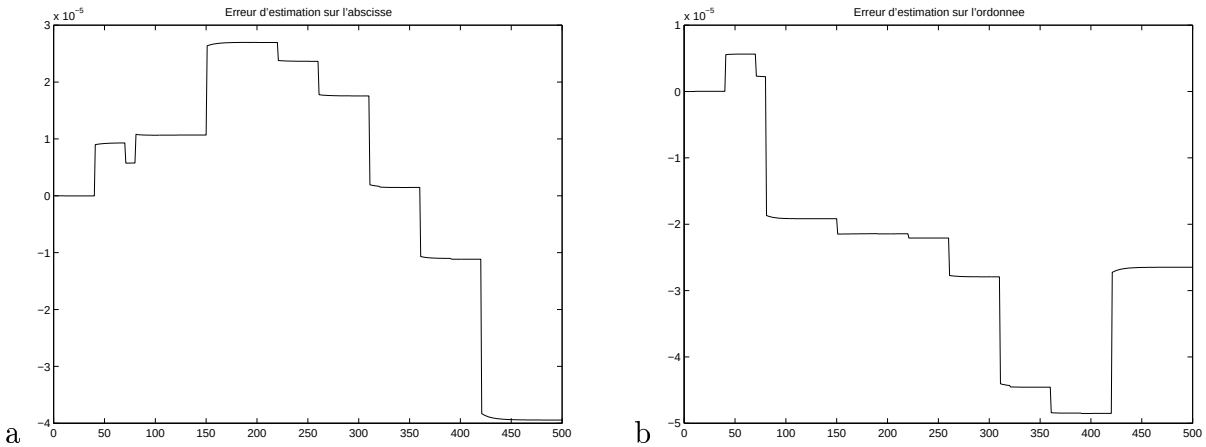


FIG. 2.33 – *Erreur d'estimation en abscisse (a) et ordonnée (b) du même point*

survenues lorsque seule la partie affine du mouvement est utilisée.

A titre de comparaison avec l'estimation par un modèle affine de mouvement, lorsque l'on considère la matrice M_q (que l'on ne donnera pas explicitement ici) qui permet de passer des mouvements relatifs de l'objet par rapport à la caméra au vecteur $(\widehat{s} - s_0)$ dans le cas où l'estimation est basée sur un modèle quadratique du mouvement, les valeurs singulières S_{qua} du produit $(L^T M_q)$ obtenues pour notre situation choisie pour la simulation sont de :

$$S_{qua} = (4.21,\ 4.21,\ 0.16,\ 2.6\ 10^{-4},\ 2.47\ 10^{-6},\ 2.29\ 10^{-6})$$

Si l'on compare ces valeurs avec celles données en (2.7), on constate que même si les deux dernières valeurs de S_{qua} sont de l'ordre de $2\ 10^{-6}$, elles sont toutefois très nettement supérieures à la valeur singulière de S_{aff} que nous avons considérée nulle ($3.02\ 10^{-18}$).

2.4.3 Discussion

Comme il est bien souvent impossible de connaître l'espace de déplacement d'un objet mobile, et en particulier de savoir si son mouvement sera uniquement translationnel, notre méthode de suivi rigide basée sur une estimation de la position de quatre points en intégrant le modèle affine n'est pas applicable en pratique. Il est nécessaire d'utiliser un modèle plus complexe de mouvement, en l'occurrence le modèle quadratique simplifié. Là se pose un nouveau problème : celui de la précision de la mesure. Les paramètres quadratiques sont très délicats à estimer, et il semble que la précision supplémentaire apportée par ces paramètres lors de l'intégration de la vitesse de chacun des points par rapport au seul modèle affine peut être entièrement "noyée" dans le bruit, aussi bien des paramètres affines que des paramètres quadratiques.

Une autre limite à l'utilisation de cette technique pour des tâches de suivi est la question du temps de calcul. La nécessité d'utiliser les paramètres quadratiques implique un temps de calcul relativement long (cf. Annexe A), incompatible avec la réactivité nécessaire au suivi d'un objet dont les vitesses de déplacement ne sont pas connues a priori. En raison de ces problèmes actuels de temps de calcul, nous ne pouvons pas présenter de résultats expérimentaux sur cette partie.

2.5 Pourquoi n'appliquer cette méthode qu'à des tâches de suivi?

Dans ce chapitre, les seules applications présentées sont des tâches de suivi, que ce soit de la surveillance ou du suivi rigide. Il est intéressant de se demander si l'on peut utiliser les informations géométriques estimées par intégration du mouvement pour des tâches de positionnement telles que les liaisons virtuelles présentées dans [Chaumette 90].

Dans un premier temps, nous discutons de la nécessité de définir une consigne géométrique dans l'image, et comment on peut l'obtenir lorsque l'on traite des images complexes. Nous proposons, ensuite, une perspective de recherche concernant le positionnement de la caméra par rapport à une scène complexe, et qui fait appel à la méthode d'intégration présentée précédemment dans ce chapitre.

2.5.1 Obtention des données nécessaires à la commande

Les schémas de commande utilisés pour les deux tâches de suivi présentées précédemment, calqués sur ceux utilisés dans les cas où des mesures géométriques sont directement exploitables, nécessitent pour fonctionner :

- une matrice d'interaction liant linéairement les variations des positions à contrôler aux mouvements de la caméra ; cette matrice pouvant être estimée;
- une mesure ou une estimation de la position des points de contrôle à chaque itération;
- la donnée d'une image de référence, ou image désirée, correspondant à la connaissance des positions à atteindre dans le repère image pour l'ensemble des points considérés par la commande.
  Remarque : Dans la suite de ce paragraphe, cette image sera désignée de façon équivalente par les termes d'image de référence, désirée ou finale. La notion d'image initiale est réservée pour l'image acquise lors de la première itération de la phase de positionnement, dont le but est justement de superposer l'image acquise avec l'image désirée.

Quand le robot évolue dans un espace suffisamment proche de sa position désirée, l'approximation de la matrice d'interaction par sa valeur à la convergence permet d'atteindre cette position désirée par un schéma d'asservissement visuel géométrique. Pour définir cette matrice approchée, les données nécessaires sont les positions désirées des points 2D et les profondeurs des points 3D leur correspondant respectivement. Toutefois, une erreur sur ces profondeurs ne remet généralement pas en cause la convergence du système, la commande s'effectuant en boucle fermée. Une estimation grossière de ces valeurs par l'utilisateur est donc suffisante.

Dans le cas où des informations géométriques sont exploitables, c'est-à-dire, généralement, quand l'objet d'intérêt est conçu explicitement pour la tâche à remplir et donc parfaitement connu, l'image de référence peut être directement donnée par l'utilisateur sans même avoir besoin d'observer réellement l'objet à la position désirée. L'exemple le plus typique concerne le cas très simple où un seul point de contrôle est utilisé et où le but est de positionner la caméra de telle manière que ce point apparaisse au centre de l'image ou suive une trajectoire (circulaire par exemple). Pour des exemples plus complexes, une calibration de la caméra est nécessaire pour déterminer la projection de l'objet

à sa position tridimensionnelle désirée.

Une autre manière de procéder consiste à positionner, dans l'espace cartésien, le robot par rapport à l'objet dans la configuration tridimensionnelle souhaitée et à apprendre l'image alors formée. Une fois extraites les informations géométriques de cette image de référence, il faut associer à chaque primitive observée sur l'image initiale, son correspondant sur l'image de référence, c'est-à-dire associer au vecteur s le vecteur s^* qui convient. Là encore, la connaissance a priori de l'objet permet d'obtenir facilement cette mise en correspondance (voir par exemple les cibles à sept points disposés en π dans [Malis 97]).

L'estimation d'un modèle du champ de vitesse global - ou de plusieurs modèles correspondant chacun à un champ local, si l'on veut rendre plus finement compte des déformations d'une scène non-planaire - étant supposée toujours possible, l'estimation de la position de points est donc uniquement soumise à la connaissance de leurs valeurs initiales. Nous avons vu, dans le cas du suivi rigide, que ces points d'intérêt pouvaient, par exemple, être désignés par l'utilisateur. Les coordonnées dans l'image de ces points forment alors le vecteur s_0. Pour déterminer le vecteur s^* des positions de ces points dans l'image désirée, il faut effectuer une recherche du correspondant de chacun des points d'intérêt dans cette image désirée. En résumé, si l'on suppose cette mise en correspondance toujours possible, la désignation de s_0 fournit s^*. De manière complètement duale, la désignation des points d'intérêt dans l'image désirée, c'est-à-dire s^* permet, par la mise en correspondance, de retrouver s_0.

Supposons que le vecteur s_0 soit fourni initialement, que ce soit par l'utilisateur ou par un algorithme de traitement d'image permettant la recherche de points significatifs. Le problème est donc de pouvoir définir l'image de référence et surtout d'associer à chaque point de contrôle de cette image désirée son correspondant dans l'image initiale. Ce problème ne se posait pas lors du suivi en *pan* et *tilt*, car la position initiale du point de contrôle était donnée par l'étape de détection, et sa position désirée - le centre de l'image - était définie par le principe même de la tâche. Il ne se posait pas non plus pour le suivi rigide, car il suffisait de considérer quatre points de l'objet avec pour seule contrainte qu'ils ne soient pas alignés (même si en pratique, une répartition de ces points sur la plus grande partie possible de l'image est souhaitable, de manière à ce que leurs positions estimées ne se confondent pas dans le bruit). Et la position désirée pour ces quatre points était simplement celle donnée par celle de la première image, puisqu'il s'agit d'effectuer un rejet de perturbation sur cette position initiale.

Si l'on souhaite réaliser une tâche de positionnement par rapport à des points de référence, il semble difficile de chercher à associer des points de l'image désirée désignés par l'utilisateur avec leurs correspondants dans une autre image, car il y a des risques importants pour que ces points ne possèdent pas de caractères discriminants. Il faut donc pouvoir détecter des points singuliers à la fois dans l'image de référence et dans l'image initiale, afin de pouvoir ensuite les mettre en correspondance. Plusieurs articles, [Zhang 95] par exemple, se sont attachés à ce problème, en séparant nettement les deux étapes. La première consiste à détecter des points singuliers, c'est-à-dire des points autour

desquels les gradients spatiaux sont élevés dans deux directions, tel ce qui est effectué dans [Harris 88]. Il s'agit typiquement de coins d'objets. La seconde étape effectue une recherche de similarité sur l'ensemble des couples composés d'un point caractéristique de chaque image. Pour chacune de ces deux étapes, des seuils peuvent être fixés de façon à ne conserver que les couples appariés possédant la plus forte probabilité d'être réellement composés de points correspondants.

Une fois cette association effectuée, les images désirée et initiale sont données. En se basant sur l'image de référence, on peut définir la matrice d'interaction et grâce à l'image initiale, l'estimation de la position de chaque point par son mouvement est possible au cours de l'asservissement.

2.5.2 Positionnement par rapport à une scène

En suivant la méthodologie ci-dessus, il est a priori possible de positionner le robot par rapport à une scène rigide. Pour cela, nous proposons un schéma de commande composés de quatre étapes : apprentissage, mise en correspondance, positionnement grossier, positionnement fin.

Apprentissage En positionnant le robot dans sa configuration désirée, on acquiert l'image de référence. Sur cette image, on réalise la détection de points singuliers par la méthode de Harris et Stephens (cf. [Harris 88]). Un exemple est donné sur la figure (2.34), où les points singuliers sont marqués par une croix.

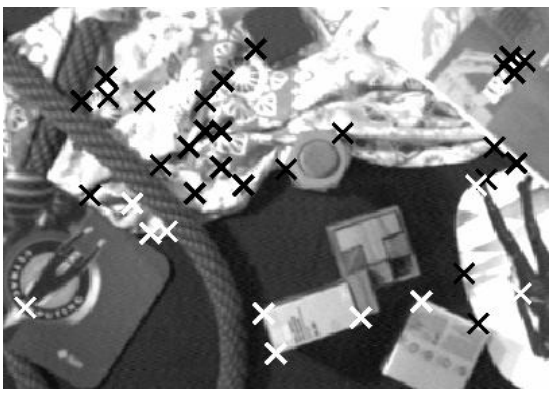


FIG. 2.34 – *Image désirée de la scène sur laquelle les points singuliers sont repérés par une croix*

Mise en correspondance Le robot se trouve maintenant dans une position différente de la position désirée, et le but de la tâche robotique est de l'y ramener. Dans cette position initiale, on réalise une nouvelle détection de points singuliers. A la suite de cette

opération, on effectue la phase de mise en correspondance par la méthode de corrélation proposée par [Zhang 95], en prenant pour référence l'ensemble des points détectés pendant l'apprentissage. Les résultats obtenus sont présentés sur la figure (2.35) avec en (a), l'image désirée sur laquelle les points singuliers conservés sont marqués par une croix et indicés, et en (b), l'image initiale où les points singuliers sont indiqués de la même manière. Sur ces deux images, deux points correspondants portent le même indice.

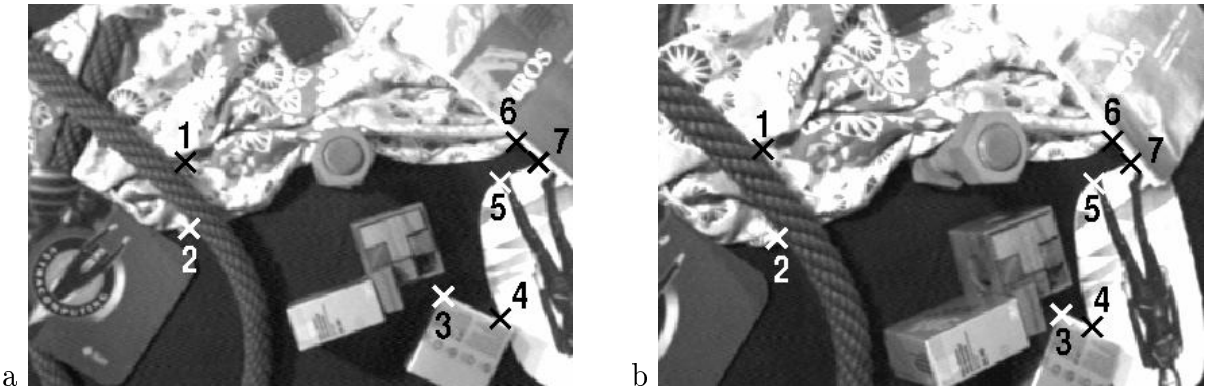


FIG. 2.35 – *Mise en correspondance des points singuliers de l'image désirées (a) et de l'image initiale (b)*

Cette technique est relativement coûteuse en temps de calcul, surtout si l'on doit considérer une déformation affine des voisinages des points d'intérêt, et non pas seulement un déplacement en translation dans l'image. Nous ne pouvons donc pas l'utiliser pour réaliser le positionnement en temps réel.

Positionnement grossier Pour réaliser cette étape, on utilise le schéma de commande donné dans le paragraphe précédent pour le suivi rigide. La seule différence est qu'il n'est pas possible d'assurer que seuls quatre points pourront être mis en correspondance pour les seuils utilisés. Et ce, même si l'on peut inclure dans la phase de mise en correspondance, un contrôle automatique de ces seuils de façon à ce que le minimum de points puissent être conservés, sous la contrainte que ce minimum soit au moins égal à quatre. La matrice L n'est donc plus de dimension 8×6 mais $2n \times 6$, où n est le nombre de points conservés.

Positionnement fin L'estimation de la position de points par intégration de leur vitesse offre le défaut de n'être jamais réactualisée par une mesure géométrique fiable. En conséquence, la commande qui utilise ce procédé permet simplement d'amener les positions estimées sur les positions désirées mais, en aucun cas, les véritables positions des points de contrôle considérés. Cependant, la robustesse de l'algorithme d'estimation du mouvement permet de penser qu'à la convergence de ce schéma de commande, les vrais correspondants des points de contrôle sont très proches des positions désirées. Il est alors

possible d'utiliser un schéma de commande où les positions des points de contrôle sont directement mesurées par un algorithme de mise en correspondance local afin d'affiner le positionnement du robot.

L'extraction d'informations géométriques d'une image complexe, qui était décrite comme contradictoire avec les contraintes temps-réel d'une tâche robotique, est ici rendue possible par la connaissance a priori de la proximité des points recherchés d'une position connue (la position finale). Cela permet d'optimiser cette extraction, en particulier en considérant des zones de recherche très restreintes.

On peut se demander pourquoi ce schéma de positionnement fin ne pourrait pas être utilisé dès le début du positionnement. Si tel était le cas, il se poserait le problème de la vitesse admissible pour le robot. En effet, si la position initiale est éloignée de la position finale, les mouvements commandés risquent d'être suffisamment importants pour que le déplacement dans l'image entre deux itérations soit supérieur aux limites admissibles pour une recherche locale. Un second argument en faveur du passage par le positionnement grossier est que, pendant cette étape, le système est robuste aux occultations partielles de la scène et en particulier des points de contrôle. Cela est dû encore une fois à la robustesse de l'algorithme RMRm aux mouvements locaux parasites.

Il existe malgré tout des travaux sur de l'asservissement visuel réalisé à partir de l'extraction de points singuliers dans l'image, comme par exemple l'article [Hager 98]. On peut citer également [Braud 97], même s'il n'entre pas dans le cadre de l'asservissement visuel, puisque l'objectif est simplement la reconnaissance d'un objet d'intérêt donné, mais il utilise un algorithme de traitement d'image similaire. Cependant, dans les deux cas, l'objet que l'on cherche à positionner précisement dans l'image est, au moins en partie, connu. Dans le premier, il s'agit de positionner un tournevis en extrayant les coordonnées de deux coins. La structure de l'objet est connue, à savoir que l'on sait comment se situent les coins par rapport aux droites qui peuvent être observées et il évolue devant un fond uni, ce qui permet d'optimiser la recherche des zones de fort gradient. Dans le deuxième article, une des données du système est le modèle tridimensionnel précis de l'objet, et là encore, il se trouve devant un fond très peu texturé.

Conclusion

Nous avons développé, dans ce chapitre, l'idée qu'un schéma d'asservissement visuel géométrique pouvait être réalisé à partir de mesures du type mouvement apparent entre deux images, par simple intégration de la vitesse des points d'intérêt pouvant être déduite de cette mesure dynamiques.

Grâce à des expérimentations de suivi d'objet mobile, réalisées en simulation et sur site réel pour le suivi en pan et tilt, nous avons pu montrer que l'estimation des positions utilisées dans la boucle de commande par cette méthode, n'était pas soumise à des problèmes de biais, classiques pourtant lorsque l'on intègre la dérivée d'une valeur pour

retrouver cette dernière. Ce résultat est dû, en grande partie, à la qualité de l'algorithme d'estimation d'un modèle de mouvement que nous avons utilisé.

Nous proposons à présent d'élargir le champ d'application de l'asservissement visuel, en ne cherchant plus seulement à remplir des contraintes géométriques dans l'image. En effet, nous étudions, dans les deux chapitres suivants, un schéma d'asservissement visuel, où les mesures dynamiques sont directement prises en compte dans la boucle de commande et non plus simplement pour permettre de retrouver des informations géométriques. Ainsi, la consigne est alors définie en termes de mouvement dans l'image.

Chapitre 3

Asservissement visuel $\frac{d2D}{dt}$

Dans le chapitre précédent, nous nous étions attachés à retrouver des informations géométriques dans l'image, à partir de mesures dynamiques, c'est-à-dire des termes homogènes à des données de vitesse dans l'image. Il nous était alors possible d'appliquer des méthodes d'asservissement visuel géométrique classiques, tout en n'ayant pas, à la différence de la plupart des travaux effectués dans ce domaine, recours à des objets dédiés à la tâche robotique considérée.

Dans ce chapitre, nous abordons une nouvelle approche de l'asservissement visuel. Tout en restant dans le cas où les mesures extraites de la séquence d'images sont dynamiques, nous ne cherchons plus désormais à remonter à des informations géométriques. Au contraire, nous proposons de calquer le schéma de l'asservissement visuel géométrique afin de contraindre directement les mesures dynamiques à atteindre une valeur souhaitée. Ainsi, nous ne cherchons pas à obtenir une image désirée, mais un champ particulier de mouvement entre deux images. Le fait de contraindre la vitesse apparente dans l'image, nous a naturellement amenés à dénommer ce schéma de commande l'asservissement visuel $\frac{d2D}{dt}$.

Dans un premier paragraphe, nous comparons cette nouvelle approche avec celle que nous avons considérée au chapitre précédent, et plus généralement, à l'asservissement visuel géométrique. Nous présentons, dans le second paragraphe, les relations qui lient les variations des informations visuelles dynamiques aux mouvements de l'objet par rapport à la caméra. Enfin, dans le troisième paragraphe, nous proposons quelques schémas de commande possibles pour cette nouvelle approche.

3.1 Comparaison des asservissements visuels géométrique et dynamique

Dans ce premier paragraphe, nous expliquons en quoi consiste l'asservissement visuel $\frac{d2D}{dt}$ (ou dynamique), en le comparant avec l'asservissement visuel géométrique rappelé au chapitre 2.

3.1.1 Comparaison schématique

Ce que nous proposons d'étudier dans ce chapitre, est la conception d'une nouvelle boucle de commande par asservissement visuel. La spécification visuelle de la configuration à atteindre n'est plus effectuée par des contraintes géométriques, comme présentés sur la figure (2.1), mais par des critères dynamiques, c'est-à-dire de vitesse apparente dans l'image. Plus précisément, nous souhaitons contrôler le robot de manière à ce que le champ de mouvement courant dans l'image, tel que celui présenté sur la figure (3.1) a, vienne égaler un champ désiré, comme par exemple le champ divergent de la figure (3.1) b.

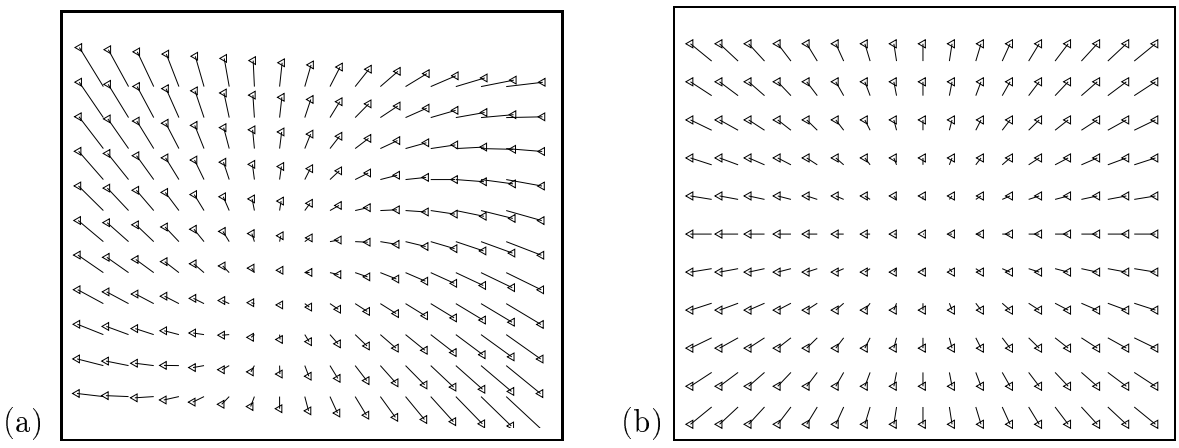


FIG. 3.1 – *Champs de mouvement courant (a) et désiré (b)*

Le principe que nous souhaitons développer possède donc un point commun avec l'asservissement visuel géométrique, puisque dans les deux cas, il s'agit de contraindre les mouvements de la caméra afin que les informations visuelles acquises par celle-ci atteignent des valeurs désirées. Pour réaliser l'objectif fixé par l'asservissement visuel dynamique, nous proposons donc de nous inspirer du schéma de commande développé dans le cas géométrique présenté dans [Espiau 92], rappelé au début du chapitre précédent.

3.1.2 Asservissement visuel dynamique

Pour pouvoir contrôler un système robotique, nous devons quantifier son état afin de pouvoir le comparer avec une configuration désirée. Si nous nous référons au paragraphe 1.1, nous pouvons distinguer trois manières d'aborder le mouvement dans l'image : l'expression d'un champ dense, l'estimation d'un modèle polynomial de mouvement, et enfin, une caractérisation binaire de présence ou non de mouvement en chaque pixel de l'image.

En ce qui concerne le champ dense, à savoir l'expression du déplacement de tous les pixels de l'image entre deux acquisitions, il paraît peu opportun de spécifier un champ désiré. En effet, même pour une image de petite taille, cela revient à utiliser un vecteur d'informations visuelles de grande dimension, et donc difficilement manipulable en pratique, d'autant plus que nous avons vu que les temps de calcul sont bien souvent prohibitifs.

Une carte binaire permettant de savoir si un pixel correspond à un point mobile par rapport à la caméra ou non n'apporte pas, comme son nom l'indique, une information continue sur la configuration du système. Par conséquent, cela ne nous permet pas de contrôler le robot selon un schéma similaire à celui que nous avons présenté ci-dessus, concernant l'asservissement visuel géométrique.

Pour ces raisons, nous avons choisi de spécifier le comportement désiré du système sous la forme d'un modèle de mouvement particulier dans l'image. Le modèle que nous considérerons est le modèle quadratique simplifié à huit paramètres, présenté au chapitre 1. Ce choix est également motivé par les relations fortes qui existent entre les paramètres affines de ce modèle et des mouvements particuliers de la caméra, mis à jour à travers les descripteurs du mouvement 2D, présentés dans la relation (1.7).

Pour commander le robot avec des contraintes géométriques dans l'image, nous devons agir sur la vitesse de la caméra. En utilisant maintenant des informations dynamiques extraites de l'image, nous "gagnons" donc un ordre de dérivation temporelle dans la mesure. Par conséquent, la loi de commande fournit également une sollicitation avec un ordre supplémentaire de dérivation, ce qui signifie que l'on agit sur les mouvements de la caméra en contraignant son accélération. Plus exactement, nous verrons, dans la suite de ce chapitre, que cette commande peut être définie comme la combinaison d'une contrainte en accélération sur certains axes et en vitesse sur d'autres.

Le principe de l'asservissement visuel dynamique est résumé sur le schéma bloc de la figure (3.2). Contrairement au schéma bloc du cas géométrique de la figure (2.2), la mesure s, ainsi que sa valeur désirée s^* correspondent désormais à des paramètres p_i du modèle de mouvement 2D. Et nous voyons que la loi de commande ne contraint plus seulement le torseur cinématique V de la caméra mais également son accélération Γ.

Nous avons vu que l'établissement d'une loi de commande en asservissement visuel géométrique passait par une phase de calcul des relations algébriques existants entre

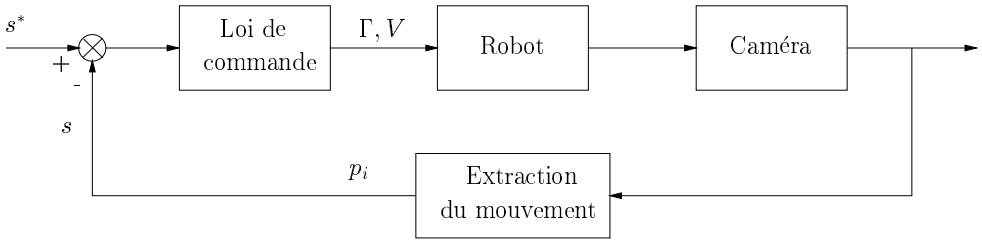


FIG. 3.2 – *Schéma bloc de l'asservissement visuel dynamique*

les variations des informations visuelles et le mouvement de la caméra. Par conséquent, dans le paragraphe suivant, nous donnons les dérivées des paramètres quadratiques du mouvement 2D, afin de déterminer les relations d'interaction entre ces paramètres et, d'une part l'accélération de la caméra, et d'autre part, sa vitesse.

3.2 Relations d'interaction des paramètres du mouvement

Si nous considérons le schéma de commande de l'asservissement visuel géométrique, nous voyons qu'une des étapes permettant de déterminer la loi de commande associée à une tâche donnée, est le calcul de la dérivée des informations visuelles considérées. L'objectif de cette phase est de pouvoir exprimer le lien entre l'évolution des paramètres et les mouvements de la caméra, supposés commandables, soit par une contrainte en vitesse, soit par une contrainte en accélération. Il devient alors possible, si ces relations sont inversibles, de contrôler les mouvements du robot, de manière à ce que le système suive une trajectoire dans l'espace des informations visuelles prises en compte.

Nous nous intéressons donc dans cette partie, aux relations d'interaction de chacun des paramètres du modèle quadratique du mouvement projeté, c'est-à-dire à l'expression de leur dérivée temporelle en fonction de l'accélération et de la vitesse de l'objet considéré par rapport à la caméra.

3.2.1 Expression des dérivées des paramètres du modèle quadratique du mouvement projeté

[Bandyopadhyay 85, Subarrao 89] s'étaient intéressés à un problème de reconstruction 3D à partir de l'analyse du mouvement. Ils ont alors été amenés à exprimer les gradients spatio-temporels dans l'image, afin de déterminer les paramètres d'un modèle du type :

$$\begin{cases} \dot{x} = c_1 + a_1 x + a_2 y + \tau_1 t \\ \dot{y} = c_2 + a_3 x + a_4 y + \tau_2 t \end{cases}$$

où les termes c_i et a_i correspondent à ceux de l'équation (1.6) et t représente le temps. En fait, chaque paramètre τ_i est la dérivée $\dot{c}_i$ de c_i.

Dans notre cas, nous ne devons pas simplement calculer la dérivée temporelle des termes constants du modèle quadratique de mouvement, mais de chacun des huit paramètres p_i de ce modèle. Si l'on reprend l'équation (1.6), nous vopyons que chaque paramètre p_i est une fonction linéaire du torseur cinématique V du mouvement de l'objet observé par rapport à la caméra :

$$p_i = L_{p_i}^T(r)\ V \tag{3.1}$$

où, pour chaque paramètre p_i, $L_{p_i}^T(r)$ est la matrice de dimension 1×6, qui exprime cette relation linéaire et est fonction de la position tridimensionnelle r de l'objet par rapport à la caméra. En particulier, $L_{p_i}^T(r)$ dépend de la profondeur de l'objet (Z_p) et de son orientation par rapport à la caméra (respectivement γ_1 et γ_2 autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$). Ces différentes matrices prennent les valeurs :

$$L_{c_1}^T = \left[\frac{1}{Z_p},\ 0,\ 0,\ 0,\ 1,\ 0\right] \qquad L_{c_2}^T = \left[0,\ \frac{1}{Z_p},\ 0,\ -1,\ 0,\ 0\right]$$

$$L_{a_1}^T = \left[-\frac{\gamma_1}{Z_p},\ 0,\ -\frac{1}{Z_p},\ 0,\ 0,\ 0\right] \qquad L_{a_2}^T = \left[-\frac{\gamma_2}{Z_p},\ 0,\ 0,\ 0,\ 0,\ -1,\ \right]$$

$$L_{a_3}^T = \left[0,\ -\frac{\gamma_1}{Z_p},\ 0,\ 0,\ 0,\ 1\right] \qquad L_{a_4}^T = \left[0,\ -\frac{\gamma_2}{Z_p},\ -\frac{1}{Z_p},\ 0,\ 0,\ 0\right]$$

$$L_{q_1}^T = \left[0,\ 0,\ \frac{\gamma_1}{Z_p},\ 0,\ 1,\ 0\right] \qquad L_{q_2}^T = \left[0,\ 0,\ \frac{\gamma_2}{Z_p},\ -1,\ 0,\ 0\right]$$

En se basant sur la relation linéaire (3.1), la dérivée temporelle d'un paramètre p_i peut s'écrire (abusivement) sous la forme :

$$\dot{p}_i = L_{p_i}^T(r)\ \dot{V} + \left(\frac{\partial L_{p_i}(r)}{\partial r}\dot{r}\right) V$$

Or, $\dot{r} = V$. La relation précédente peut donc s'écrire comme la somme d'une relation linéaire avec l'accélération et d'une forme quadratique du torseur cinématique :

$$\dot{p}_i = L_{p_i}^T\Gamma + V^T\ Q_{p_i}\ V \tag{3.2}$$

où :

- Γ représente l'accélération de l'objet par rapport à la caméra exprimée dans le repère caméra, c'est-à-dire $\Gamma = (\dot{T}_x,\ \dot{T}_y,\ \dot{T}_z,\ \dot{\Omega}_x,\ \dot{\Omega}_y,\ \dot{\Omega}_z)^T$;

- L_{p_i} sera désormais nommé vecteur d'interaction avec l'accélération associé à p_i ;
- Q_{p_i} est une matrice de dimension 6×6 liée à la forme quadratique qui associe à p_i la dépendance de sa dérivée par rapport à la vitesse de l'objet. Il y a donc plusieurs expressions possibles pour Q_{p_i}, et en particulier celles d'une matrice symétrique ou d'une matrice triangulaire. Entre ces formes équivalentes, nous choisirons, généralement, l'expression en un matrice triangulaire supérieure.

3.2.2 Calcul des matrices Q_{p_i}

Le calcul direct des matrices Q_{p_i} n'est pas évident à effectuer, en particulier parce que les composantes du vecteur r de position de l'objet dans l'espace n'apparaissent pas de manière complètement explicite dans l'expression des matrices $L_{p_i}^T$, mais sous la forme des valeurs Z_p, γ_1 et γ_2 qui ne dépendent que de r. Le calcul des dérivées $\dot{Z}_p$, $\dot{\gamma}_1$ et $\dot{\gamma}_2$ de ces trois valeurs, peut donc nous permettre de déterminer les dérivées des paramètres quadratiques du mouvement.

Ainsi, pour établir les valeurs des matrices Q_{p_i} nous avons calculé directement les dérivées des termes p_i à partir de leur forme développée. Nous obtenons donc, pour les paramètres constants :

$$\dot{c}_1 = -\frac{\dot{Z}_p T_x}{Z_p^2} + \frac{\dot{T}_x}{Z_p} + \dot{\Omega}_y \qquad \dot{c}_2 = -\frac{\dot{Z}_p T_y}{Z_p^2} + \frac{\dot{T}_y}{Z_p} - \dot{\Omega}_x \tag{3.3}$$

De la même manière, nous avons pour les paramètres affines :

$$\begin{cases} \dot{a}_1 &= \dfrac{\dot{Z}_p}{Z_p^2}(T_z + \gamma_1 T_x) - \dfrac{1}{Z_p}(\dot{T}_z + \gamma_1 \dot{T}_x + \dot{\gamma}_1 T_x) \\ \dot{a}_2 &= \dfrac{\gamma_2 \dot{Z}_p T_x}{Z_p^2} - \dfrac{1}{Z_p}(\gamma_2 \dot{T}_x + \dot{\gamma}_2) - \dot{\Omega}_z \\ \dot{a}_3 &= \dfrac{\gamma_1 \dot{Z}_p T_y}{Z_p^2} - \dfrac{1}{Z_p}(\gamma_1 \dot{T}_y + \dot{\gamma}_1 T_y) + \dot{\Omega}_z \\ \dot{a}_4 &= \dfrac{\dot{Z}_p}{Z_p^2}(T_z + \gamma_2 T_y) - \dfrac{1}{Z_p}(\dot{T}_z + \gamma_2 \dot{T}_y + \dot{\gamma}_2 T_y) \end{cases} \tag{3.4}$$

Et enfin, nous avons pour les paramètres quadratiques :

$$\dot{q}_1 = \frac{\gamma_1 \dot{Z}_p T_z}{Z_p^2} - \frac{\dot{\gamma}_1 T_z - \gamma_1 \dot{T}_z}{Z_p} - \dot{\Omega}_y \qquad \dot{q}_2 = \frac{\gamma_2 \dot{Z}_p T_z}{Z_p^2} - \frac{\dot{\gamma}_2 T_z - \gamma_1 \dot{T}_z}{Z_p} + \dot{\Omega}_x \tag{3.5}$$

Par ailleurs, en considérant l'équation (1.5) de l'approximation planaire autour du point P, premier point de l'objet rencontré sur l'axe optique, nous avons, pour tout point M de la surface de l'objet dans le voisinage de P, et de coordonnées $(X, Y, Z)^T$, la relation suivante :

$$\dot{Z} = \dot{Z}_p + \gamma_1 \dot{X} + \gamma_2 \dot{Y} + \dot{\gamma_1} X + \dot{\gamma_2} Y$$

c'est-à-dire en utilisant les valeurs de $\dot{X}$, $\dot{Y}$ et $\dot{Z}$ donnés par (1.4)

$$\begin{array}{l} T_z - \dot{Z}_p - \gamma_1(T_x + \Omega_y Z_p) - \gamma_2(T_y - \Omega_x Z_p) \\ -X(\Omega_y + \gamma_1^2 \Omega_y + \gamma_2 \Omega_z - \gamma_1 \gamma_2 \Omega_x + \dot{\gamma_1}) \\ +Y(\Omega_x + \gamma_2^2 \Omega_x + \gamma_1 \Omega_z - \gamma_1 \gamma_2 \Omega_y - \dot{\gamma_2}) = 0 \end{array} \tag{3.6}$$

La relation (3.6) est une relation affine en X et Y, valable pour tout point M de l'approximation planaire de la surface de l'objet autour du point P. Une condition nécessaire à la nullité du terme de gauche est donc que chacun des coefficients affines de cette fonction soit nul. On en déduit donc les égalités suivantes :

$$\begin{array}{rcl} \dot{Z}_p & = & T_z - \gamma_1(T_x + \Omega_y Z_p) - \gamma_2(T_y - \Omega_x Z_p) \\ \dot{\gamma_1} & = & -\Omega_y - \gamma_1^2 \Omega_y - \gamma_2 \Omega_z + \gamma_1 \gamma_2 \Omega_x \\ \dot{\gamma_2} & = & \Omega_x + \gamma_2^2 \Omega_x + \gamma_1 \Omega_z - \gamma_1 \gamma_2 \Omega_y \end{array} \tag{3.7}$$

Enfin, en remplaçant $\dot{Z}_p$, $\dot{\gamma_1}$ et $\dot{\gamma_2}$ par leur valeur respective dans les expressions des $\dot{p_i}$ données en (3.3), (3.4), et (3.5), nous obtenons :

- pour les paramètres constants

$$\left\{ \begin{array}{rcl} \dot{c}_1 & = & \gamma_1 v_x^2 + \gamma_1 v_x \Omega_y + \gamma_2 v_x v_y - \gamma_2 v_x \Omega_x - v_x v_z + \dfrac{\dot{T}_x}{Z_p} + \dot{\Omega}_y \\ \dot{c}_2 & = & \gamma_1 v_x v_y + \gamma_1 v_y \Omega_y + \gamma_2 v_y^2 - \gamma_2 v_y \Omega_x - v_y v_z + \dfrac{\dot{T}_y}{Z_p} - \dot{\Omega}_x \end{array} \right.$$

- pour les paramètres affines

$$\left\{ \begin{array}{rcl} \dot{a}_1 & = & -\gamma_1^2 v_x^2 - \gamma_1 \gamma_2 v_x v_y + \gamma_2 v_x \Omega_z + v_x \Omega_y - \gamma_1 v_z \Omega_y - \gamma_2 v_z v_y \\ & & +\gamma_2 v_z \Omega_x + v_z^2 - \dfrac{\gamma_1 \dot{T}_x}{Z_p} - \dfrac{\dot{T}_z}{Z_p} \\ \dot{a}_2 & = & \gamma_1 \gamma_2 v_x^2 - \gamma_2^2 v_x v_y + \gamma_2 v_x v_z - \gamma_1 v_x \Omega_z - v_x \Omega_x - \dfrac{\gamma_2 \dot{T}_x}{Z_p} - \dot{\Omega}_z \\ \dot{a}_3 & = & -\gamma_1^2 v_x v_y - \gamma_1 \gamma_2 v_y^2 + \gamma_1 v_y v_z + \gamma_2 v_y \Omega_z + v_y \Omega_y - \dfrac{\gamma_1 \dot{T}_y}{Z_p} + \dot{\Omega}_z \\ \dot{a}_4 & = & -\gamma_1 \gamma_2 v_x v_y - \gamma_2^2 v_y^2 - \gamma_1 v_y \Omega_z - v_y \Omega_x - \gamma_1 v_x v_z - \gamma_1 v_z \Omega_y \\ & & +\gamma_2 v_z \Omega_x + v_z^2 - \dfrac{\gamma_2 \dot{T}_y}{Z_p} - \dfrac{\dot{T}_z}{Z_p} \end{array} \right.$$

- pour les paramètres quadratiques

$$\begin{cases} \dot{q}_1 &= \gamma_1^2 v_x v_z + \gamma_1\gamma_2 v_y v_z - \gamma_1 v_z^2 - \gamma_2 v_z \Omega_z - v_z \Omega_y + \dfrac{\gamma_1 \dot{T}_z}{Z_p} + \dot{\Omega}_y \\ \dot{q}_2 &= \gamma_1\gamma_2 v_x v_z + \gamma_2^2 v_y v_z - \gamma_2 v_z^2 + \gamma_1 v_z \Omega_z + v_z \Omega_x + \dfrac{\gamma_2 \dot{T}_z}{Z_p} - \dot{\Omega}_x \end{cases}$$

Nous retrouvons bien dans ces expressions chacune des matrices $L_{p_i}^T$ facteur de l'accélération de l'objet dans le repère caméra. A partir de ces résultats, nous pouvons également exprimer chacune des matrices Q_{p_i}. Ainsi, nous avons, sous forme de matrices triangulaires supérieures :

$$Q_{c_1} = \begin{bmatrix} \frac{\gamma_1}{Z_p^2} & \frac{\gamma_2}{Z_p^2} & \frac{-1}{Z_p^2} & \frac{-\gamma_2}{Z_p} & \frac{\gamma_1}{Z_p} & 0 \\ 0 & 0 & \dots & & \dots & 0 \\ \vdots & \ddots & \ddots & & & \vdots \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix} \qquad Q_{c_2} = \begin{bmatrix} 0 & \frac{\gamma_1}{Z_p^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{\gamma_2}{Z_p^2} & \frac{-1}{Z_p^2} & \frac{-\gamma_2}{Z_p} & \frac{\gamma_1}{Z_p} & 0 \\ \vdots & \ddots & 0 & \dots & \dots & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix}$$

$$Q_{a_1} = \begin{bmatrix} \frac{-\gamma_1^2}{Z_p^2} & \frac{-\gamma_1\gamma_2}{Z_p^2} & 0 & 0 & \frac{1}{Z_p} & \frac{\gamma_2}{Z_p} \\ 0 & 0 & \frac{-\gamma_2}{Z_p^2} & 0 & 0 & 0 \\ \vdots & 0 & \frac{1}{Z_p^2} & \frac{\gamma_2}{Z_p} & \frac{-\gamma_1}{Z_p} & 0 \\ \vdots & & \ddots & 0 & \dots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix} \quad Q_{a_2} = \begin{bmatrix} \frac{-\gamma_1\gamma_2}{Z_p^2} & \frac{-\gamma_2^2}{Z_p^2} & \frac{\gamma_2}{Z_p^2} & \frac{-1}{Z_p} & 0 & \frac{-\gamma_1}{Z_p} \\ 0 & 0 & \dots & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & & & \vdots \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix}$$

$$Q_{a_3} = \begin{bmatrix} 0 & \frac{-\gamma_1^2}{Z_p^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{-\gamma_1\gamma_2}{Z_p^2} & \frac{\gamma_1}{Z_p^2} & 0 & \frac{1}{Z_p} & \frac{\gamma_2}{Z_p} \\ \vdots & 0 & 0 & \dots & \dots & 0 \\ \vdots & & \ddots & 0 & \dots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix} \quad Q_{a_4} = \begin{bmatrix} 0 & \frac{-\gamma_1\gamma_2}{Z_p^2} & \frac{-\gamma_1}{Z_p^2} & 0 & 0 & 0 \\ 0 & \frac{-\gamma_2^2}{Z_p^2} & 0 & \frac{-1}{Z_p} & 0 & \frac{-\gamma_1}{Z_p} \\ \vdots & 0 & \frac{1}{Z_p^2} & \frac{\gamma_2}{Z_p} & \frac{-\gamma_1}{Z_p} & 0 \\ \vdots & & \ddots & 0 & \dots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix}$$

$$Q_{q_1} = \begin{bmatrix} 0 & 0 & \frac{\gamma_1^2}{Z_p^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{\gamma_1\gamma_2}{Z_p^2} & 0 & 0 & 0 \\ \vdots & 0 & \frac{-\gamma_1}{Z_p^2} & 0 & \frac{-1}{Z_p} & \frac{-\gamma_2}{Z_p} \\ \vdots & & \ddots & 0 & \dots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix} \quad Q_{q_2} = \begin{bmatrix} 0 & 0 & \frac{\gamma_1\gamma_2}{Z_p^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{\gamma_2^2}{Z_p^2} & 0 & 0 & 0 \\ \vdots & 0 & \frac{-\gamma_2}{Z_p^2} & \frac{1}{Z_p} & 0 & \frac{\gamma_1}{Z_p} \\ \vdots & & \ddots & 0 & \dots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix}$$

3.3 Idées pour la commande

3.3.1 Contexte

Les paramètres du modèle quadratique du mouvement sont homogènes aux composantes du torseur cinématique de l'objet par rapport à la caméra. Nous appellerons dorénavant information visuelle dynamique, soit un de ces paramètres du modèle quadratique du mouvement, soit un terme obtenu par différence d'un de ces paramètres et d'une des composantes du torseur cinématique du mouvement entre l'objet et la caméra. Considérons un vecteur s composé de n informations visuelles dynamiques. Nous nous plaçons dans le cas d'une tâche visant à ramener s à la valeur s^*. Nous pouvons écrire, en utilisant les relations (3.2) définies pour chacune des composantes de s :

$$\dot{s} = L_s(r)\Gamma + f(V, r) \tag{3.8}$$

où :

- Γ et V représentent toujours l'accélération et le torseur cinématique de l'objet par rapport à la caméra, et r est la position de l'objet dans le repère de la caméra ;
- la matrice $L_s(r)$ de dimension $n\times 6$ est composée de chacune des matrices lignes faisant le lien entre chaque élément s_i de s et l'accélération Γ, données par les relations (3.1). Nous appellerons désormais $L_s(r)$ matrice d'interaction associée à s, et nous la noterons indifféremment sous cette forme ou, plus simplement, L ;
- $f(r, V)$ est une fonction vectorielle dont chaque composante est la forme quadratique associée, par les relations (3.2), à l'élément de s correspondant.

Remarque : Nous nous sommes placés ici dans le cas où le nombre de degrés de liberté commandés est égal à 6. Ce nombre peut évidemment être inférieur, et dans ce cas le vecteur Γ est remplacé par un de ses sous-vecteurs composés des accélérations sur les axes considérés. Dans la suite de ce paragraphe, nous considérerons cependant toujours que six degrés de libertés sont utilisés, sans perte de généralité.

Pour contrôler l'évolution du système, nous agissons sur les mouvements de la caméra. Nous devons donc les faire apparaître dans la relation précédente. En fait, les vecteurs représentant l'accélération et la vitesse de l'objet par rapport à la caméra peuvent s'écrire :

$$\Gamma = \Gamma_o - \Gamma_c \quad \text{et} \quad V = V_o - V_c$$

où les indices o et c représentent respectivement le mouvement de l'objet et de la caméra par rapport à un repère fixe galiléen dans lequel le système évolue. Commander le système consiste donc à agir sur les valeurs de Γ_c ou V_c. La relation d'interaction peut donc s'écrire :

$$\dot{s} = -L_s(r)\Gamma_c - f(V_c, r) + g_o(\Gamma_o, V_o, r)$$

où $g_o(\Gamma_o, V_o, r)$ est une fonction des mouvements propres de l'objet, que nous noterons simplement g_o dorénavant.

Nous pouvons récrire cette équation en regroupant les termes dépendant du mouvement propre de l'objet sous la forme plus générale $\dfrac{\partial s}{\partial t}$. Ainsi, nous obtenons :

$$\dot{s} = -L_s(r)\Gamma_c - f(V_c, r) + \frac{\partial s}{\partial t} \tag{3.9}$$

Remarque : Une nouvelle fois, l'équation précédente est écrite dans le cas où les 6 degrés de liberté de l'effecteur du robot sont contrôlés par le système. Si tel n'est pas le cas, et que Γ_c et V_c ne représentent que des sous-vecteurs de l'accélération et de la vitesse de la caméra, l'influence des composantes non-contrôlées sur les variations de s se retrouve également dans le terme $\dfrac{\partial s}{\partial t}$.

Une équation intégro-différentielle non-linéaire

Soit s^* la valeur désirée pour le vecteur s, et $e = C(s - s^*)$, l'erreur entre ces deux vecteurs, avec C une matrice de taille $6 \times n$ à définir. L'objectif de la commande est d'annuler chacune des composantes du vecteur e. Nous pouvons pour cela, en nous calquant sur la méthode utilisée dans [Espiau 92] pour l'asservissement visuel géométrique, appliquer une loi de commande de type gradient, en imposant une décroissance exponentielle de l'erreur de gain λ. Nous avons alors :

$$\lambda e = CL\ \Gamma_c + C\ f(V_c, r) - \frac{\partial e}{\partial t} \tag{3.10}$$

où $\dfrac{\partial e}{\partial t} = C\dfrac{\partial s}{\partial t}$.

Pour pouvoir contrôler les mouvements de la caméra à l'aide du robot sur lequel elle est embarquée, nous devons résoudre l'équation intégro-différentielle (3.10), c'est-à-dire, qu'à partir des mesures successives de e, nous devons être à même de déterminer l'accélération ou la vitesse de la caméra qui permettra d'obtenir le comportement décroissant souhaité.

Remarque : L'équation précédente est dite intégro-différentielle, et non pas seulement différentielle, car le terme intégral ou différentiel dépend du choix qui est fait de commander le robot en accélération ou en vitesse. S'il s'agit de commander en accélération, et donc de résoudre l'équation pour obtenir ce vecteur, l'équation est alors intégrale, car la vitesse est, par définition, une fonction intégrale de cette accélération. Au contraire, le choix de commander en vitesse conduit à la résolution d'une équation différentielle.

Cependant, cette équation est fortement non-linéaire, car la fonction f est quadratique pour les termes de vitesse de la caméra. Par conséquent, plutôt que d'utiliser des techniques de résolution d'une telle équation, nous proposerons des moyens de la linéariser, afin d'établir très simplement le vecteur de commande.

Si l'équation intégro-différentielle (3.10) est non-linéaire, nous devons cependant noter que la relation entre les variations de e et l'accélération de la caméra est, en revanche, linéaire. Cela signifie que si l'on considère f comme, non pas une fonction quadratique de la vitesse de la caméra, mais plus généralement comme une fonction mesurable de l'état du système, alors l'équation (3.10) peut s'écrire sous la forme :

$$\lambda e + f'(\text{état}) - \frac{\partial e}{\partial t} = CL\ \Gamma_c \tag{3.11}$$

où $f'(\text{état})$ vaut $-Cf(r, V)$. Nous ne considérons donc plus alors cette équation comme

étant intégro-différentielle, mais simplement nous cherchons à déterminer une relation linéaire entre l'accélération de la caméra et l'état du système considéré.

3.3.2 Cas d'une matrice d'interaction de rang plein

Supposons la matrice L de rang plein, c'est-à-dire 6. Nous pouvons choisir la matrice C de telle manière que la matrice CL soit inversible. Il est alors possible d'inverser la relation (3.11) afin d'obtenir une commande en accélération de notre système sous la forme :

$$\Gamma = (CL)^{-1} \left(\lambda e - Cf(V_c, r) + \frac{\partial e}{\partial t} \right) \tag{3.12}$$

V_c est obtenu par odométrie ou considéré comme perturbation (erreur de traînage) annihilée par estimation.

En réalité, il est souvent difficile d'obtenir une mesure précise de L. Nous travaillons donc avec une estimation ou une approximation de cette matrice, notée $\widehat{L}$. Il en est de même pour les valeurs des fonctions f et $\dfrac{\partial e}{\partial t}$. Elles sont donc estimées, respectivement, par les fonctions $\widehat{f}$ et $\widehat{\dfrac{\partial e}{\partial t}}$. La relation précédente s'écrit donc :

$$\Gamma_c = \left(C\widehat{L} \right)^{-1} \left(\lambda \widehat{e} - C\widehat{f}(V_c, r) + \widehat{\frac{\partial e}{\partial t}} \right)$$

où $\widehat{e}$ est la mesure de e donnée par l'algorithme d'estimation du mouvement apparent dans l'image.

Posons $K = (CL)(C\widehat{L})^{-1}$. Le comportement du système bouclé est, par conséquent, décrit par la relation :

$$\dot{e} = -\lambda K \widehat{e} + K \left(C\widehat{f} - \widehat{\frac{\partial e}{\partial t}} \right) - \left(Cf - \frac{\partial e}{\partial t} \right) \tag{3.13}$$

Il est délicat, à partir d'une telle relation de définir des contraintes permettant d'assurer la décroissance de l'erreur e. Nous pouvons toutefois remarquer que le comportement réel du système sera d'autant plus proche du comportement souhaité si la matrice K est proche de l'identité. Cette condition est notée :

$$K \approx \mathbb{I} \tag{3.14}$$

En utilisant une approximation un peu plus grossière, nous pouvons admettrons que la positivité de la matrice K suffit à maintenir la stabilité du système. Cela est d'autant

plus vrai lorsque les estimations du vecteur e et des fonctions f et $\frac{\partial e}{\partial t}$ sont de bonne qualité.

Plus précisément, si l'on note :

$$\begin{cases} \widehat{e} & = e + b_e(t) \\ \widehat{f} & = f + b_f(t) \\ \widehat{\frac{\partial e}{\partial t}} & = \frac{\partial e}{\partial t} + b_g(t) \end{cases}$$

où b_e, b_f et b_g représentent des fonctions de bruit, respectivement sur e, f et $\frac{\partial e}{\partial t}$, nous pouvons écrire la relation (3.13) sous la forme :

$$\dot{e} = -\lambda K e + (K - \mathbb{I}) \left(Cf - \frac{\partial e}{\partial t} \right) \underbrace{-\lambda b_e + KCb_f - Kb_g}_{b(t)}$$

où $b(t)$ symbolise le bruit global sur le système, que l'on neglige ici.

En admettant que le terme $\left(Cf - \frac{\partial e}{\partial t} \right)$ reste borné et que le bruit b est suffisamment "petit", nous pouvons démontrer, de la même manière que dans [Samson 91], que la positivité de la matrice K permet au moins d'assurer que l'erreur ne divergera pas. Par la suite, nous considérerons que cette condition, traduite par :

$$K > 0 \tag{3.15}$$

permet de conclure sur la stabilité du système.

De plus, lorsque l'on sait, a priori, que la fonction $\left(Cf - \frac{\partial e}{\partial t} \right)$ est nulle, c'est-à-dire, par exemple lorsque les six degrés de liberté de la caméra sont contraints et que l'objet de référence, sur lequel l'asservissement est réalisé, demeure immobile, la décroissance de l'erreur e vers un voisinage, également borné, autour de la valeur nulle est assuré par cette même condition.

Estimation de f et $\frac{\partial e}{\partial t}$

Nous supposons, dans un premier temps, que le mouvement propre de l'objet est nul. Nous avons donc $V = -V_c$ et $\frac{\partial e}{\partial t} = 0$. Admettons également que l'on dispose de mesures odométriques fiables nous permettant d'obtenir entre deux itérations successives le torseur cinématique de la caméra par rapport au repère fixe, que l'on note $(\widehat{T}_c, \widehat{\Omega}_c)^T$.

Reprenons les relations d'interaction des huit paramètres du modèle quadratique simplifié du mouvement, données au paragraphe 3.2. Les fonctions quadratiques $V_c^T\ Q_{p_i}\ V_c$ de l'équation (3.2), peuvent être décomposée comme la somme de produit de facteur pris dans l'ensemble :

$$\{v_{c,i},\ \Omega_{c,i},\ \gamma_1 v_{c,i},\ \gamma_2 v_{c,i}\} \quad \text{où} \quad i \in \{x, y, z\}$$

Nous pouvons montrer que chacun de ces facteurs peut être déterminé par la mesure odométrique de la vitesse de la caméra et par l'estimation des paramètres du modèle affine de mouvement apparent dans l'image. Pour cela, nous rappelons d'abord les valeurs de ces paramètres lorsque l'objet est immobile :

$$\begin{cases} c_1 = -v_{c,x} - \Omega_{c,y} & c_2 = -v_{c,y} + \Omega_{c,x} \\ a_1 = \gamma_1 v_{c,x} + v_{c,z} & a_2 = \gamma_2 v_{c,x} + \Omega_{c,z} \\ a_3 = \gamma_1 v_{c,y} - \Omega_{c,z} & a_4 = \gamma_2 v_{c,y} + v_{c,z} \end{cases}$$

Les termes $\Omega_{c,i}$ sont directement donnés par les mesures odométriques. A partir de ces mesures de rotation et des termes constants du mouvement, on retrouve les termes $v_{c,x}$ et $v_{c,y}$. De même, grâce à la mesure de $\Omega_{c,z}$ et des termes a_2 et a_3, on peut retrouver les valeurs des termes $\gamma_2 v_{c,x}$ et $\gamma_1 v_{c,y}$. Donc, connaissant celles de $v_{c,x}$ et $v_{c,y}$, nous pouvons également estimer les valeurs de γ_1 et γ_2. Et finalement, avec ces deux dernières valeurs et toujours celles de $v_{c,x}$ et $v_{c,y}$, nous pouvons déterminer $v_{c,z}$ à partir des paramètres a_1 et a_4. Plus explicitement, ces relations sont les suivantes :

$$\begin{cases} v_{c,x} = -c_1 - \Omega_{c,y} & v_{c,y} = -c_2 + \Omega_{c,x} \\ \gamma_1 = \dfrac{a_3 + \Omega_{c,z}}{v_{c,y}} & \gamma_2 = \dfrac{a_2 - \Omega_{c,z}}{v_{c,x}} \\ v_{c,z} = \dfrac{(a_1 - \gamma_1 v_{c,x}) + (a_4 - \gamma_2 v_{c,y})}{2} & \end{cases} \tag{3.16}$$

Nous considérons maintenant que l'objet n'est plus immobile, et que nous ne possédons aucune information a priori sur son mouvement. Il ne nous est pas possible de distinguer, dans les termes affines du mouvement, ce qui dépend uniquement du mouvement de la caméra du reste. En particulier, le raisonnement utilisé ci-dessus se basait sur la capacité à distinguer l'influence de la rotation de celle de la translation, et sur la possibilité de mesurer cette rotation. Cela n'est plus vrai dans le cas présent.

Si nous reprenons l'équation (3.9), nous pouvons l'écrire :

$$\dot{s} = -L\Gamma_c + f(V, r) + L\Gamma_o = -L\Gamma_c + \frac{\partial s}{\partial t}$$

où le second terme de la dernière égalité représente globalement l'influence du mouvement de l'objet (que ce soit par rapport à la caméra ou par rapport au repère fixe) sur les

variations de s. C'est ici un abus d'écriture par rapport à la relation (3.9), motivée par le fait que dans le cas où la matrice L est de rang plein, les composantes du vecteur V_c ne sont pas contraintes par la loi de commande et peuvent donc être considérées comme des perturbations sur le système.

Donc, en considérant trois images successives, nous pouvons déterminer deux valeurs successives de s notées $s(t)$ et $s(t+\delta t)$, et par conséquent exprimer une approximation de $\dot{s}$ sous la forme :

$$\dot{s} \approx \frac{s(t+\delta t) - s(t)}{\delta t} = \widehat{\frac{ds}{dt}} \tag{3.17}$$

Ainsi, grâce à une mesure odométrique de l'accélération $\widehat{\Gamma}$ de la caméra à l'instant t (à partir des variations de positions entre les instants $t-\delta t$ et t d'une part, et entre les instants t et $t+\delta t$ d'autre part), et à une estimation $\widehat{L}$ de la matrice L, nous pouvons estimer le terme d'influence du mouvement de l'objet par :

$$\frac{\partial s}{\partial t} \approx \widehat{\frac{\partial s}{\partial t}} = \widehat{\frac{ds}{dt}} + \widehat{L}\widehat{\Gamma} \tag{3.18}$$

En pratique, pour éviter les problèmes liés aux bruits de mesure, cette estimation peut être filtrée par un filtrage de Kalman.

3.3.3 Cas d'une matrice d'interaction de rang non-plein

Nous considérons maintenant le cas où la matrice L n'est pas de rang plein. Avec $e = C(s - s^*)$, nous pouvons toujours dans ce cas écrire la relation (3.10) et traduire l'équation (3.9) par la relation d'interaction :

$$\dot{e} = -CL\Gamma_c - Cf(V_c, r) - \frac{\partial e}{\partial t} \tag{3.19}$$

Par contre, nous ne pouvons plus déterminer le vecteur d'accélération de la caméra, permettant d'obtenir la décroissance exponentielle de e, avec la technique proposée ci-dessus. En effet, la matrice CL n'est alors plus inversible. Nous appelons m son rang, avec $m < 6$, et p la dimension de son noyau (on sait, par définition que $p = 6 - m$).

Un cas que nous devons également considérer est celui où le rang de la matrice L peut varier au cours de l'asservissement, selon la configuration alors atteinte par le système. Ce cas peut, par exemple, se présenter lorsque la caméra est parallèle au plan tangent à l'objet. En effet, les termes γ_1 et γ_2, représentant l'orientation de ce plan, sont alors nuls et les matrices L_{p_i} des relations (3.1) n'ont alors plus qu'un seul terme non-nul, au lieu de deux lorsque la caméra et l'objet ne sont pas parallèles. Ce cas apparaît, par exemple lors de la tâche de suivi de trajectoire que nous verrons au chapitre suivant.

Dans tous ces cas de figure, nous proposons de récrire l'équation (3.19) de façon à faire apparaître une nouvelle relation d'interaction où $\dot{e}$ serait exprimé en fonction de

p composantes de vitesse et m composantes d'accélération. Si la matrice d'interaction obtenue entre $\dot{e}$ et ce vecteur, composé à la fois de termes de vitesse et d'accélération, est de rang plein, il est alors possible de contraindre le déplacement de la caméra en contrôlant p axes en vitesse et m en accélération.

Une première idée consiste à négliger p termes d'accélération et à décomposer la fonction f sous la forme de la somme d'une relation linéaire des p termes de vitesse qui correspondent à ces accélérations et d'une fonction quadratique résultant de cette écriture et ne mettant en jeu que les m autres termes de vitesse. Cependant, l'approximation utilisée paraît trop abusive, et nous n'avons pas développé cette idée.

La solution que nous avons retenu consiste à discrétiser p termes d'accélération sous la forme :

$$\Gamma_{c,i}(t) = \frac{V_{c,i}(t) - V_{c,i}(t - \delta t)}{\delta t}$$

où $\Gamma_{c,i}$ et $V_{c,i}$ sont respectivement la i-ème composante du vecteur d'accélération et du torseur cinématique de la caméra, t est l'instant où l'on effectue cette discrétisation et, enfin, δt est un intervalle de temps.

Le choix des termes d'accélération à discrétiser n'est pas évident a priori, puisque le noyau de CL n'est pas forcément généré par p composantes de l'accélération. Cela signifierait en particulier que m colonnes de CL soient des vecteurs nuls. Nous verrons au chapitre suivant sur plusieurs exemples qu'il n'y a en général aucune colonne nulle, ce qui implique que le noyau est généré par des combinaisons linéaires mettant en jeu l'ensemble des axes commandés.

Cependant, en général ce choix est possible après une analyse de la tâche considérée. Il peut être motivé, par exemple, par la volonté d'obtenir une expression la plus simple possible de la relation d'interaction. Cela peut également être fait par des considérations "logiques", certes moins objectives. Par exemple, dans le cas du positionnement parallèle à plan associé avec une fixation dans l'image la commande est relativement découplée entre la translation et la rotation. L'un des choix possibles conduit à ne pas commander l'orientation par les axes de rotation et est donc rejeté.

Pour plus de clarté dans les notations, nous considérons que la discrétisation est effectuée pour les p premiers termes du vecteur Γ. Il n'y a pas de perte de généralité à restreindre notre étude à ce cas particulier puisqu'il suffit dans les cas contraires de récrire la relation (3.11) en transformant Γ par une permutation de ses composantes, de façon à faire apparaître celles qui sont discrétisées aux p premières places.

Nous notons Σ le vecteur de commande $(V_{c,1}, \ldots, V_{c,p}, \Gamma_{c,p+1}, \cdots, \Gamma_{c,6})^T$. En considé-

rant la discrétisation des p premiers termes d'accélération, nous pouvons écrire :

$$CL\Gamma = CL \begin{pmatrix} \frac{V_{c,1}(t)}{\delta t} \\ \vdots \\ \frac{V_{c,p}(t)}{\delta t} \\ \Gamma_{p+1} \\ \vdots \\ \Gamma_6 \end{pmatrix} - CL \begin{pmatrix} \frac{V_{c,1}(t-\delta t)}{\delta t} \\ \vdots \\ \frac{V_{c,p}(t-\delta t)}{\delta t} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

En posant :

$$CL = \begin{bmatrix} (CL)_1 & (CL)_3 \\ (CL)_2 & (CL)_4 \end{bmatrix}$$

où $(CL)_1$ est de dimension $p \times p$, $(CL)_2$ $m \times p$, $(CL)_3$ $p \times m$ et $(CL)_4$ $m \times m$, nous obtenons :

$$CL\Gamma = \begin{bmatrix} \frac{(CL)_1}{\delta t} & (CL)_3 \\ \frac{(CL)_2}{\delta t} & (CL)_4 \end{bmatrix} \Sigma - \underbrace{\begin{bmatrix} \frac{(CL)_1}{\delta t} & 0 \\ \frac{(CL)_2}{\delta t} & 0 \end{bmatrix} V_c(t-\delta t)}_{\text{noté } Cl(V_c)} \tag{3.20}$$

Afin de séparer, dans l'expression de la fonction f, la partie dépendant des termes de vitesse du vecteur de commande, nous écrivons :

$$Cf(V_c, r) = P \begin{pmatrix} V_{c,1}(t) \\ \vdots \\ V_{c,p}(t) \\ 0 \\ \vdots \\ 0 \end{pmatrix} + Ch(V_{c,p+1}, \ldots, V_{c,6}, r) \tag{3.21}$$

où h représente la partie de f qui ne dépend que des m derniers termes du vecteur vitesse et de r.

La forme linéaire qui fait apparaître la matrice P peut toujours être obtenue. Cependant, si l'expression de la fonction f contient des termes de la forme $V_{c,i}.V_{c,j}$ avec i et j compris entre 1 et p, il faut alors décider de considérer l'une de ces vitesses comme une mesure et l'autre comme une entrée de commande. La volonté d'éviter les cas de ce type peut au choix des p termes d'accélération que l'on discrétise.

Notons $P = \begin{bmatrix} P_1 & P_3 \\ P_2 & P_4 \end{bmatrix}$, où P_i a les mêmes dimensions que $(CL)_i$. Les deux sous-matrices P_3 et P_4 n'ont aucune influence sur le calcul, car elles sont toujours multipliées par des vecteurs nuls. Nous pouvons donc les considérer toujours nulles. On a donc :

$$P \begin{pmatrix} V_{c,1}(t) \\ \vdots \\ V_{c,p}(t) \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{bmatrix} P_1 & 0 \\ P_2 & 0 \end{bmatrix} \Sigma$$

Dans la relation (3.21), tous les termes dépendant du temps qui sont mesurés, c'est-à-dire dans la matrice P et dans la fonction h, sont considérés comme mesuré à l'instant $t - \delta t$. Seuls les p termes de vitesse commandés dépendent de l'instant t. C'est cette manipulation, valable pour un intervalle de temps δt suffisamment petit, qui permet de prendre en compte la dépendance de la fonction f vis-à-vis du vecteur de commande Σ.

En définitive, en regroupant les relations (3.20) et (3.21), nous obtenons une nouvelle relation d'interaction :

$$\dot{e} = M\Sigma - Ch - Cl + \frac{\partial e}{\partial t} \tag{3.22}$$

avec :

$$M = \begin{bmatrix} \dfrac{(CL)_1}{\delta t} + P_1 & (CL)_3 \\ \dfrac{(CL)_2}{\delta t} + P_2 & (CL)_4 \end{bmatrix}$$

Nous admettrons qu'il existe un ensemble de p indices tels que la discrétisation des p éléments de l'accélération Γ_c correspondant à ces indices, nous permette d'obtenir une matrice M de rang plein. Dans le cas contraire, nous considérerons que le système n'est pas commandable.

Un exemple

Illustrons ces différents calculs par un exemple. Considérons le cas de la tâche de positionnement parallèlement à un plan couplée avec une tâche de fixation que nous verrons au paragraphe 4.3 dans le chapitre suivant. La relation d'interaction associée est

donnée par (voir équation (4.17)) :

$$\dot{e} = \begin{pmatrix} \gamma_1^2 v_x v_z + \gamma_1\gamma_2 v_y v_z - \gamma_1 v_z^2 - \gamma_2 v_z\Omega_z - v_z\Omega_y + \dfrac{\gamma_1 \dot{T}_z}{Z_p} \\ \gamma_1\gamma_2 v_x v_z + \gamma_2^2 v_y v_z - \gamma_2 v_z^2 + \gamma_1 v_z\Omega_z + v_z\Omega_x + \dfrac{\gamma_2 \dot{T}_z}{Z_p} \\ \gamma_1 v_x^2 + \gamma_1 v_x\Omega_y + \gamma_2 v_x v_y - \gamma_2 v_x\Omega_x - v_x v_z + \dfrac{\dot{T}_x}{Z_p} + \dot{\Omega}_y \\ \gamma_1 v_x v_y + \gamma_1 v_y\Omega_y + \gamma_2 v_y^2 - \gamma_2 v_y\Omega_x - v_y v_z + \dfrac{\dot{T}_y}{Z_p} - \dot{\Omega}_x \end{pmatrix}$$

Considérons dans un premier temps que le vecteur de commande, composé uniquement de termes d'accélération, vaut $\Gamma = (\dot{\Omega}_x, \dot{\Omega}_y, \dot{T}_x, \dot{T}_y)^T$. Nous avons donc :

$$CL = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1/Z_p & 0 \\ -1 & 0 & 0 & 1/Z_p \end{bmatrix}$$

La matrice L est de rang 2. Nous discrétisons les deux termes d'accélération angulaire $\dot{\Omega}_x$ et $\dot{\Omega}_y$. Nous avons alors :

$$P = \begin{bmatrix} 0 & -v_z & 0 & 0 \\ v_z & 0 & 0 & 0 \\ -\gamma_2 v_x & \gamma_1 v_x & 0 & 0 \\ -\gamma_2 v_y & \gamma_1 v_y & 0 & 0 \end{bmatrix}$$

On en déduit :

$$M = \begin{bmatrix} 0 & -v_z & 0 & 0 \\ v_z & 0 & 0 & 0 \\ -\gamma_2 v_x & 1/\delta t + \gamma_1 v_x & 1/Z_p & 0 \\ -1/\delta t - \gamma_2 v_y & \gamma_1 v_y & 0 & 1/Z_p \end{bmatrix}$$

qui est bien de rang 4.

Par ailleurs, nous avons :

$$h = \begin{pmatrix} \gamma_1 v_x^2 + \gamma_2 v_x v_y - v_x v_z \\ \gamma_1 v_x v_y + \gamma_2 v_y^2 - v_y v_z \\ \gamma_1^2 v_x v_z + \gamma_1\gamma_2 v_y v_z - \gamma_1 v_z^2 - \gamma_2 v_z\Omega_z \\ \gamma_1\gamma_2 v_x v_z + \gamma_2^2 v_y v_z - \gamma_2 v_z^2 + \gamma_1 v_z\Omega_z \end{pmatrix}$$

$$\text{et } l = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/\delta t & 0 & 0 \\ 1/\delta t & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \Omega_x \\ \Omega_y \\ T_x \\ T_y \end{pmatrix}$$

Loi de commande

En partant de la relation (3.22), dans le cas où M est inversible, on peut établir une loi de commande de type gradient, en imposant une décroissance exponentielle de gain λ de l'erreur :

$$\Sigma = -\widehat{M}^{-1} \left(\lambda \widehat{e} - C(\widehat{h} + \widehat{l}) + \widehat{\frac{\partial e}{\partial t}}) \right) \tag{3.23}$$

où, comme précédemment, $\widehat{M}$ est une approximation de la matrice M, $\widehat{h}$, $\widehat{l}$ et $\widehat{\frac{\partial e}{\partial t}}$ sont, respectivement, des estimations des fonctions h, l et $\frac{\partial e}{\partial t}$, et, enfin, $\widehat{e}$ est la mesure de e.

Dans le cas où l'objet observé est potentiellement mobile, la fonction $\frac{\partial e}{\partial t}$ est estimée selon un processus similaire à celui de l'équation (3.18). Les fonctions h et l peuvent être soit mesurées, soit estimées. Pour les estimer, nous effectuons la différence entre les variations de e et le vecteur $\widehat{M}\widehat{\Sigma}$, de la même manière que nous le faisions dans le cas d'une matrice L de rang plein entre $\dot{e}$ et $\widehat{L}\widehat{\Gamma}$.

Remarque sur la stabilité

Pour décrire le comportement du système bouclé, il faut exprimer le vecteur d'accélération à chaque instant. Cela est simple pour les m derniers termes de ce vecteur, car ils sont directement donnés par la loi de commande (3.23). Par contre, cela est beaucoup plus complexe pour les p premiers termes, car la loi de commande ne donne alors que les vitesses qui correspondent. Il est par conséquent très difficile d'évaluer la stabilité du système. Nous ne la traiterons donc pas ici.

Cas où des termes d'accélérations sont des vecteurs propres du noyau de CL

Nous considérons ici le cas où des vecteurs d'accélération sont des vecteurs de base du noyau de CL. Supposons que c'est le cas du terme Γ_i. Cela signifie donc que la colonne de la matrice CL qui lui correspond est nulle. Par conséquent, une discrétisation de Γ_i ne ferait apparaître aucun terme $V_i/\delta t$, mais seulement les termes issus de la matrice P. Au contraire, ne pas le discrétiser conduirait à laisser une colonne nulle dans la matrice M, qui ne pourrait donc pas être de rang plein.

Dans ce cas-là, il ne sert alors à rien de discrétiser, mais nous pouvons directement remplacer dans le vecteur de commande l'accélération Γ_i par la vitesse V_i qui lui correspond (en effectuant, évidemment les changements qui s'imposent dans la relation d'interaction). Ce cas correspond, par exemple à la tâche de positionnement de la caméra parallèlement à un plan dans laquelle le temps à collision est contrôlé, que nous verrons au chapitre suivant.

Cas où L est nulle

Lorsque la matrice L est nulle, nous arrivons au cas limite du cas abordé ci-dessus, où tous les termes d'accélération sont dans le noyau de CL. La relation d'interaction se réduit donc à la fonction quadratique f. Le principe est que pour chacun des termes de cette fonction, qui est un produit de deux composantes de vitesse, nous pouvons considérer l'un d'entre eux comme une entrée de commande et l'autre comme une mesure de l'état à l'instant précédent.

Plusieurs choix sont a priori possible pour écrire cette relation, selon que la matrice alors formée est triangulaire (supérieure ou inférieure), symétrique, etc..

Pour illustrer cette technique, considérons un exemple très simple, celui de la tâche étudiée dans [Sundareswaran 96] ayant pour objectif d'aligner l'axe optique avec la direction de translation de la caméra, où le vecteur s est composé de deux paramètres :

$$s = (c_1 - \Omega_y, c_2 + \Omega_x)^T$$

Si T_x et T_y sont constants (c'est le cas étudié), la relation d'interaction associée à s s'écrit :

$$\dot{s} = \begin{pmatrix} \gamma_1 v_x^2 + \gamma_1 v_x \Omega_y + \gamma_2 v_x v_y - \gamma_2 v_x \Omega_x - v_x v_z \\ \gamma_1 v_x v_y + \gamma_1 v_y \Omega_y + \gamma_2 v_y^2 - \gamma_2 v_y \Omega_x - v_y v_z \end{pmatrix}$$

Par conséquent, si l'on choisit, par exemple, de commander les axes de rotation, cette relation peut se récrire sous la forme linéaire diagonale suivante :

$$\dot{s} = \begin{bmatrix} -\gamma_2 v_x & \gamma_1 v_x \\ -\gamma_2 v_x & \gamma_1 v_y \end{bmatrix} \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} + \begin{pmatrix} v_x(\gamma_1 v_x + \gamma_2 v_y - v_z) \\ v_y(\gamma_1 v_x + \gamma_2 v_y - v_z) \end{pmatrix}$$

Conclusion

En nous inspirant des travaux réalisés en asservissement visuel géométrique, nous avons émis, dans ce chapitre, la possibilité de contrôler un système robotique par asservissement visuel basé sur des informations dynamiques. Le principe est de vouloir superposer le champ de mouvement mesuré entre deux images successivement acquises par une caméra montée sur le robot, avec un champ de mouvement désiré. C'est ce que nous appelons

asservissement visuel dynamique ou asservissement visuel $d2D/dt$. Plus précisément, nous cherchons à obtenir des valeurs données pour des paramètres d'un modèle polynomial du mouvement dans l'image.

En nous calquant, une fois encore, sur les méthodes de commandes utilisées en asservissement visuel géométrique, nous avons déterminé les relations liant les dérivées des paramètres du modèle quadratique de mouvement 2D et les mouvements de la caméra. Dans notre cas, contrairement au cas géométrique, ces relations sont non-linéaires. Nous avons donc proposé quelques idées pour établir une loi de commande permettant d'atteindre l'objectif fixé.

Deux cas se présentent selon que la matrice liant les dérivées des paramètres et l'accélération de la caméra soit de rang plein ou non. Si tel est le cas, il suffit de considérer les termes dépendant de la vitesse comme une mesure de l'état du système à l'instant précédent, et nous obtenons alors une relation linéaire entre l'accélération de la caméra et un vecteur d'informations que l'on peut extraire de l'image. Nous pouvons alors établir simplement une loi de commande en accélération.

Si, au contraire, cette matrice n'est pas de rang plein, il nous faut discrétiser certains termes du vecteur d'accélération, de façon à obtenir une relation linéaire entre des informations visuelles dynamiques et un vecteur de commande composé à la fois de termes d'accélération et de vitesse de la caméra.

Qui dit linéarisation d'une équation différentielle non-linéaire, dit approximations ou simplifications. Nous devons donc tester les techniques permettant de commander un système par asservissement visuel dynamique présentées dans ce chapitre, afin d'en évaluer les limites et les possibilités. Dans le chapitre suivant, nous appliquons ces méthodes dans plusieurs cas de tâches robotiques. Les résultats que nous obtenons, tant en simulation, que sur site robotique réel, nous permettent de valider ces différentes idées.

Chapitre 4

Exemple de tâches en asservissement visuel $\frac{d2D}{dt}$

Nous présentons dans ce chapitre différentes tâches robotiques mettant en œuvre les techniques d'asservissement visuel dynamique exposées dans le chapitre précédent. Ces différentes tâches ont été choisie de manière à illustrer chacun des cas abordés, à savoir lorsque la matrice d'interaction entre le vecteur de tâche choisi et l'accélération est de rang plein, lorsqu'elle n'est, au contraire, pas de rang plein, et même lorsqu'elle est nulle. Les tâches que nous abordons sont :

- le suivi en rotation panoramique et en inclinaison d'un objet mobile, de manière à le maintenir à la même position dans l'image ;
- le suivi rigide d'un objet afin de maintenir constante sa position et son orientation par rapport à la caméra ;
- le positionnement du plan de projection de la caméra parallèlement à un objet planaire.
- et enfin, le suivi d'une trajectoire parallèlement à une surface plane.

Pour chacune de ces tâches, nous donnons, après en avoir exposé les objectifs, la loi de commande associée. Ensuite, nous montrons des résultats d'expérimentations, d'abord en simulation, puis sur site robotique (sauf pour la tâche de suivi rigide).

4.1 Poursuite en *pan* et *tilt* d'un objet mobile

4.1.1 Présentation de la tâche

L'objectif de cette tâche de suivi est similaire à celui du suivi décrit dans le chapitre 2. En effet, le but est encore de maintenir un objet d'intérêt visible dans l'image. Toutefois, l'approche pour parvenir à cet objectif est différente, puisque nous ne cherchons pas dans ce nouveau cas, à retrouver des informations visuelles géométriques. Le principe se base ici sur l'idée qu'un objet observé dans l'image restera à sa position initiale si sa vitesse apparente reste nulle. Une différence avec le cas abordé au chapitre 2 est que l'objet ne peut pas être amené dans une position précise, mais simplement conservé à sa position initiale.

Cette tâche se place dans le cas où la matrice d'interaction entre le vecteur de mesure et l'accélération est de rang plein. Comme nous l'avons vu au chapitre précédent, la commande peut alors s'effectuer entièrement en accélération.

4.1.2 Loi de commande associée

Nous ne souhaitons contrôler dans cette tâche que le déplacement apparent de l'objet dans l'image, c'est-à-dire que nous ne nous intéressons pas aux déformations affines ou quadratiques. Ce déplacement global de l'objet est caractérisé par les paramètres constants du mouvement. La contrainte donnée par la tâche peut donc se traduire par la condition :

$$(c_1, c_2)^T = (0, 0)^T$$

Nous choisissons donc le vecteur $s = (c_1, c_2)^T$ comme vecteur de tâche, auquel est associée la valeur désirée $s^* = (0, 0)^T$.

Pour parvenir à cet objectif, nous commandons les deux mouvement de rotation de la caméra autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$. Les mouvements sur les autres axes sont fixés à zéro. L'objet d'intérêt est lui mobile, et son mouvement est a priori inconnu. Nous ne ferons donc d'hypothèse de nullité pour aucun des six paramètres du torseur cinématique caractérisant son mouvement par rapport à un repère galiléen fixe dans lequel le système évolue. Dans la suite de ce paragraphe, nous distinguerons les composantes des trois torseurs cinématiques en jeu par la règle suivante : les termes représentant le mouvement de l'objet par rapport au repère fixe porteront un indice o, les deux vitesses de rotation de la caméra auront un indice c. Enfin, le mouvement relatif de l'objet par rapport à la caméra sera noté sans indice.

Relation d'interaction

Pour établir la commande, il nous faut dans un premier temps, établir le lien entre les variations de s et les mouvements des axes retenus pour effectuer la tâche, qu'il s'agisse

de vitesses ou d'accélérations angulaires. Nous nous appuyons pour cela sur les équations (3.2). En utilisant la nullité des vitesses sur quatre des axes de la caméra, nous obtenons :

$$\dot{s} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \dot{\Omega}_{c,x} \\ \dot{\Omega}_{c,y} \end{pmatrix} + [\gamma_1(v_x + \Omega_y) + \gamma_2(v_y - \Omega_x) - v_z] \begin{pmatrix} v_x \\ v_y \end{pmatrix} + \begin{pmatrix} \frac{\dot{T}_{o,x}}{Z_p} + \dot{\Omega}_{o,y} \\ \frac{\dot{T}_{o,y}}{Z_p} - \dot{\Omega}_{o,x} \end{pmatrix}$$

Cependant, une écriture sous cette forme ne nous est pas très utile. En effet, la plupart des termes qui dépendent du mouvement de l'objet, que ce soit par rapport à la caméra ou au repère fixe, n'est pas explicitement mesurable, et ce, à cause de la méconnaissance totale que nous avons de ce mouvement. Nous préférons donc écrire cette relation sous une forme qui se contente de faire la distinction entre l'influence du mouvement de la caméra sur les variations de s et celle de l'objet. Nous écrivons donc simplement :

$$\dot{s} = L \begin{pmatrix} \dot{\Omega}_{c,x} \\ \dot{\Omega}_{c,y} \end{pmatrix} + \frac{\partial s}{\partial t} \tag{4.1}$$

avec :

$$L = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

et où le second terme rend compte, à la fois du mouvement propre de l'objet et des éventuels mouvements sur les axes non-contraints par la loi de commande.

Souhaitant contrôler deux axes à partir de la mesure de deux paramètres du mouvement, nous pouvons choisir le vecteur d'erreur e associé à la tâche directement égal à $s - s^*$, c'est-à-dire que la matrice C vaut $\mathbb{I}_2$. Dans le cas présent, L est parfaitement estimée à chaque itération ($\widehat{L} = L$). La stabilité et la convergence du système sont donc assurées, car la matrice :

$$K = (CL)(C\widehat{L})^{-1} = L\widehat{L}^{-1} = \mathbb{I}_2$$

remplit évidemment bien la condition de l'équation (3.14) d'être proche de l'identité.

Estimation de l'influence du mouvement de l'objet

A partir de trois images successives, acquises respectivement aux instants t_{k-1}, t_k, t_{k+1}, nous pouvons obtenir deux valeurs successives s_k et s_{k+1} de notre vecteur de tâche, grâce à l'algorithme d'estimation du mouvement opérant sur la zone de l'image correspondant à l'objet. Ayant ces deux mesures successives de s, il est possible d'approximer la valeur de $\dot{s}$, grâce également, à une mesure du temps écoulé entre les acquisitions de ces deux valeurs.

Si, enfin, nous pouvons obtenir par odométrie les variations des positions angulaires de la caméra entre deux instants, nous retrouvons les vitesses de rotation de la caméra $\Omega_{c,k}$ et $\Omega_{c,k+1}$ correspondant, respectivement aux valeurs s_k et s_{k+1} de s.

L'ensemble de ces données nous permet de définir à la fois une approximation discrète de $\dot{s}$ et de l'accélération angulaire. Plus précisément, nous avons :

$$\dot{s} \approx \frac{s_{k+1} - s_k}{\delta t} = \widehat{\frac{ds}{dt}} \quad \text{et} \quad \dot{\Omega}_c \approx \frac{\Omega_{c,k+1} - \Omega c,_k}{\delta t} = \widehat{\frac{d\Omega_c}{dt}}$$

avec δt le temps écoulé entre les deux instants des mesures s_k et s_{k+1} de s, assimilés aux valeurs médianes des instants d'acquisition des images correspondantes, c'est-à-dire :

$$\delta t = \frac{t_{k+1} - t_{k-1}}{2}$$

En définitive, en suivant la technique de l'équation (3.18), nous obtenons une estimation de l'influence du mouvement de l'objet sur $\dot{s}$ sous la forme :

$$\widehat{\frac{\partial s}{\partial t}} = \widehat{\frac{ds}{dt}} - L\widehat{\dot{\Omega}_c}$$

Cette estimation est en pratique complétée par un filtre de Kalman à accélération constante, et ce, afin de réduire l'influence des erreurs de mesure.

Loi de commande

La matrice L de la relation (4.1) étant inversible, nous pouvons appliquer une méthode de commande de type gradient, en imposant une décroissance exponentielle de gain λ de l'erreur $s - s^*$. En utilisant l'estimation définie auparavant du terme correspondant au mouvement de l'objet, la loi obtenue est :

$$\begin{pmatrix} \dot{\Omega}_{c,x} \\ \dot{\Omega}_{c,y} \end{pmatrix} = -L^{-1} \left(\lambda \widehat{s} + \widehat{\frac{\partial s}{\partial t}} \right) \tag{4.2}$$

où $\widehat{s}$ représente la mesure de s fournie par l'algorithme d'estimation du mouvement.

4.1.3 Résultats de simulation

Nous avons testé la loi de commande (4.2) en simulation. Dans le cas considéré, l'objet est placé au départ à une distance de 1 m de la caméra. L'acquisition fictive des images est considérée comme fonctionnant à une cadence fixe, à savoir la cadence vidéo (δt =40 ms). Enfin, le gain λ de la décroissance exponentielle est fixé à 5. Une valeur aussi élevée est nécessaire pour que le système soit à même de répondre rapidement aux perturbations générées par une variation dans le mouvement de l'objet.

L'objet est soumis à un mouvement composé de deux translations et d'une rotation à vitesses constantes. Chacune de ces phases est atteinte après une période d'accélération constante d'une durée de 10 à 20 itérations. Enfin, ces mouvements sont arrêtés par des décélérations du même type. Nous présentons sur la figure (4.1) ce mouvement de l'objet, avec respectivement la translation (en m/s) sur la figure a et la rotation (en deg/s) sur la figure b. Ces valeurs sont celles obtenues dans le repère fixe. Dans les deux cas, la composante en x est en trait plein, celle en y en trait pointillé et celle en z en trait discontinu.

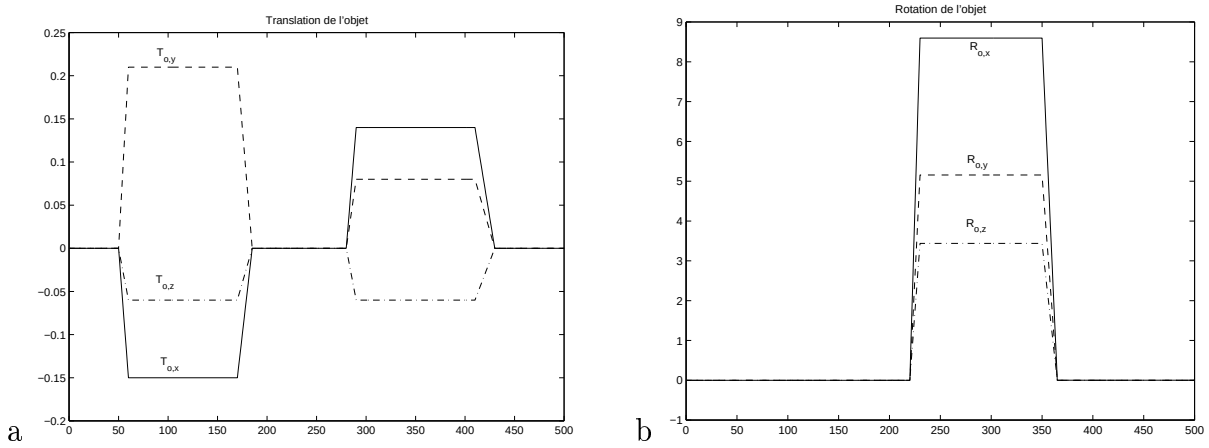


FIG. 4.1 – *Vitesses de translation (a) et de rotation (b) commandées, en m/s.* $T_{o,x}$ *et* $\Omega_{o,x}$ *en trait plein (—),* $T_{o,y}$ *et* $\Omega_{o,y}$ *en trait pointillé (- - -),* $T_{o,z}$ *et* $\Omega_{o,z}$ *en trait discontinu (– · –)*

Nous présentons quatre courbes de résultats concernant le suivi en *pan* et *tilt* d'un objet animé d'un tel mouvement. Il s'agit tout d'abord de la fonction de tâche, sur la figure (4.2) a, où le terme c_1 apparaît en trait plein et c_2 en trait pointillé. La figure (4.2) b montre les accélérations angulaires (en deg/s^2) calculées par la loi de commande (4.2) avec $\dot{\Omega}_{c,x}$ en trait plein et $\dot{\Omega}_{c,y}$ en trait pointillé. Les vitesses angulaires correspondantes (en deg/s) $\Omega_{c,x}$ et $\Omega_{c,y}$ sont présentées sur la figure (4.3) a. Enfin, la position du centre de l'objet (en m), supposée, sans perte de généralité, au centre de l'image au départ, est présentée sur la figure (4.3) b, avec la composante en x en trait plein, celle en y en trait pointillé. Cette position nous permet de juger de la qualité du suivi mais n'est, bien sûr, pas utilisée dans la loi de commande.

Nous pouvons constater que la tâche est bien remplie car les perturbations générées par le mouvement de l'objet, et qui apparaissent sur les paramètres constants du mouvement, sont très rapidement compensées à chaque variation importante du mouvement de l'objet. Pour obtenir un tel résultat, la sollicitation du système en terme d'accélération angulaire est cependant importante (jusqu'à 40 deg/s^2), même si la vitesse reste modérée.

Simulation. Suivi en pan et tilt d'un objet mobile par asservissement visuel dynamique

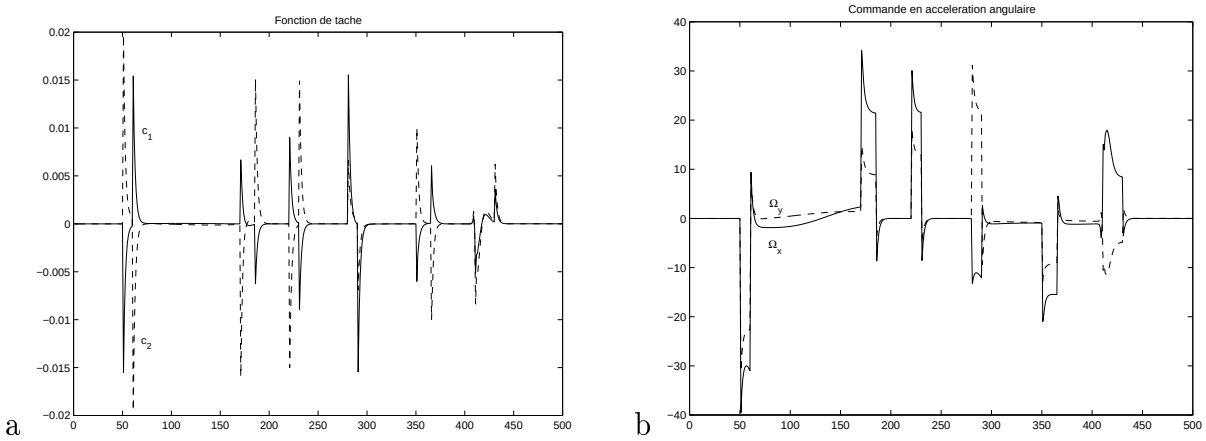


FIG. 4.2 – *Fonction de tâche, paramètres constants du mouvement (a). Commande en accélération angulaire (b)*

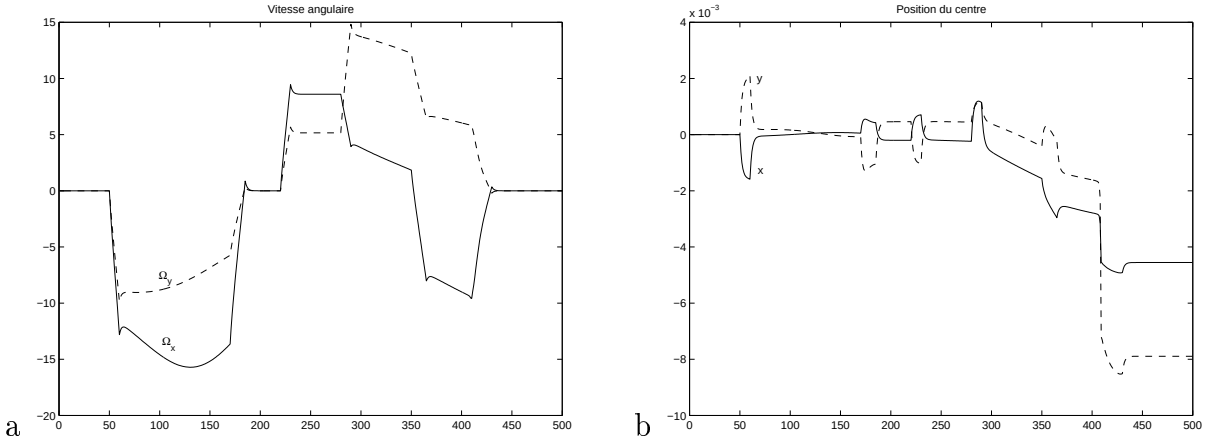


FIG. 4.3 – *Vitesse angulaire de la caméra (a). Position du centre de l'objet (b)*

La position du centre de l'objet subit une légère déviation (1 cm environ à la fin de l'asservissement) due à la discrétisation qui ne permet pas d'estimer très finement l'influence du mouvement de l'objet sur les variations de s. Cela est d'autant plus vrai lorsque l'accélération est rapide. Nous pouvons malgré tout considérer que pour une distance de 1 m entre le centre optique et le plan de projection de l'image, cette erreur est faible, compte tenu de la rapidité du mouvement de l'objet.

Pour montrer l'intérêt de l'estimation $\widehat{\frac{\partial s}{\partial t}}$ du mouvement propre de l'objet dans la boucle de commande, nous présentons quatre courbes de résultats obtenues avec une loi de commande similaire à celle de la relation (4.2), mais dans laquelle le terme $\widehat{\frac{\partial s}{\partial t}}$ est fixé à 0. Ces quatre courbes sont, comme précédemment : la fonction de tâche (figure (4.4) a), l'accélération angulaire (figure (4.4) b), la vitesse angulaire (figure (4.5) a) et la position du centre de l'objet (figure (4.4) b).

Les différences qui apparaissent dans ce cas, par rapport au précédent, sont surtout visibles avec l'évolution des paramètres constants du mouvement. En effet, contrairement au cas où l'estimation de l'influence du mouvement de l'objet est prise en compte dans la boucle de commande, nous nous apercevons ici, que les perturbations dues à des changements dans le mouvement de l'objet, sont beaucoup plus lentement compensées. Plus précisément, le système n'est pas capable, dans ce cas, de compenser les variations de s qui surviennent pendant les phases d'accélération et de décélération. Il s'agit de ce que l'on appelle une erreur de traînage. Ce phénomène peut également être noté sur les courbes de commandes en accélération angulaire, où les sollicitations sont beaucoup moins brusques que précédemment. Enfin, ces erreurs de traînage ont bien entendu, une forte influence sur la position de l'objet dans l'image. Son centre se retrouve à la fin de l'asservissement à environ 10 cm du centre dans chaque direction, alors que cette erreur finale était inférieure à 1 cm dans le cas précédent.

4.1.4 Résultats sur site robotique

Nous avons testé notre algorithme de suivi utilisant sur schéma d'asservissement visuel dynamique sur site robotique. Les conditions de l'expérimentation étaient les mêmes que dans le cas où le suivi en *pan* et *tilt* était basé sur le contrôle de la position estimée du centre de l'objet, au chapitre 2. Il s'agissait de l'objet présenté sur la figure (2.8), se déplaçant en translation le long d'un rail, à une distance d'environ 1 m de la caméra. Les déplacements ont consisté en des phases de translation à vitesse constante de 25 cm/s, alternativement vers la gauche et vers la droite, entrecoupées d'arrêt de 4 s. Les étapes d'accélération et de décélération entre ces phases sont effectuées avec une valeur absolue de 40 cm/s^2, c'est-à-dire qu'il faut quinze itérations, à la cadence vidéo, pour atteindre

Simulation. Suivi en pan et tilt d'un objet mobile par asservissement visuel dynamique sans estimation de l'influence du mouvement propre de l'objet

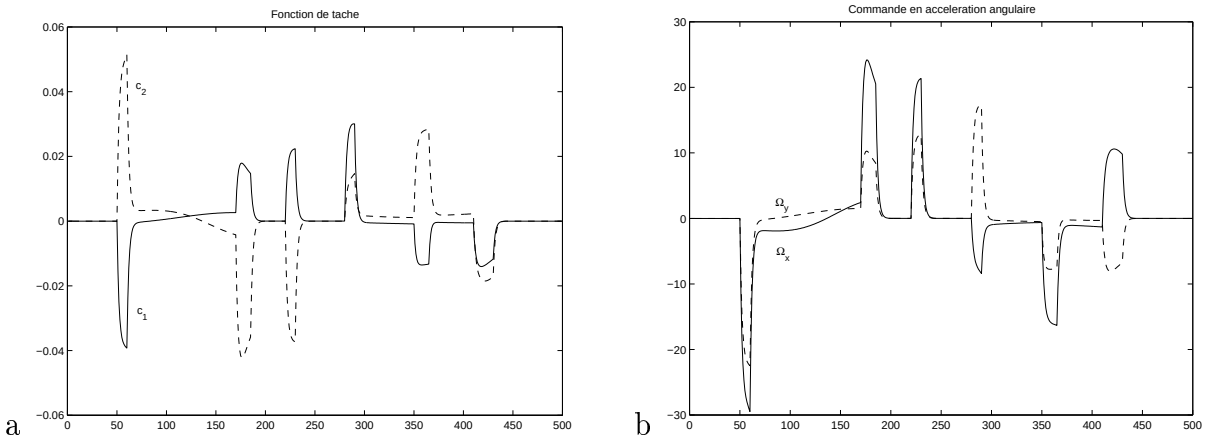


FIG. 4.4 – *Fonction de tâche, paramètres constants du mouvement (a). Commande en accélération angulaire (b)*

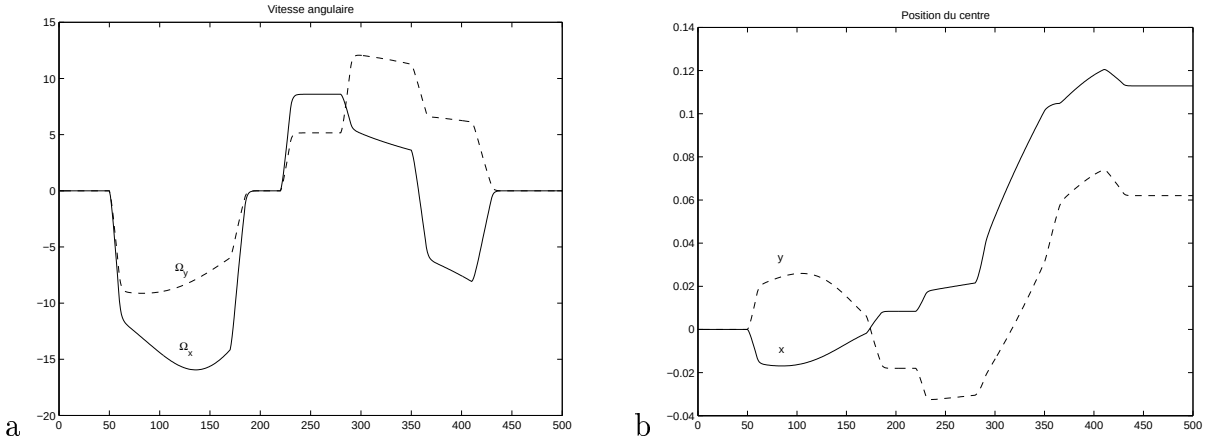


FIG. 4.5 – *Vitesse angulaire de la caméra (a). Position du centre de l'objet (b)*

la vitesse palier. Enfin, pour diminuer l'influence du bruit de mesure sur l'estimation du mouvement propre de l'objet, on utilise un filtrage de Kalman à accélération constante.

Nous présentons trois courbes de résultats correspondant à cette expérimentation. Il s'agit tout d'abord, de la fonction de tâche (en pixel/s), c'est-à-dire les termes constants du modèle de mouvement apparent de l'objet, sur la figure (4.6) a, où c_1 est en trait plein et c_2 en trait pointillé. Ensuite, nous montrons l'accélération angulaire (en deg/s^2) sur la figure (4.6) b, avec $\dot{\Omega}_{c,x}$ en trait plein et $\dot{\Omega}_{c,y}$ en trait pointillé. Enfin, sur la figure (4.7), les deux composantes de la position estimée du centre de gravité de l'objet (en pixels) sont tracées, avec x en en trait plein et y en trait pointillé.

Expérimentation sur site robotique. Suivi en pan et tilt d'un objet mobile par asservissement visuel dynamique

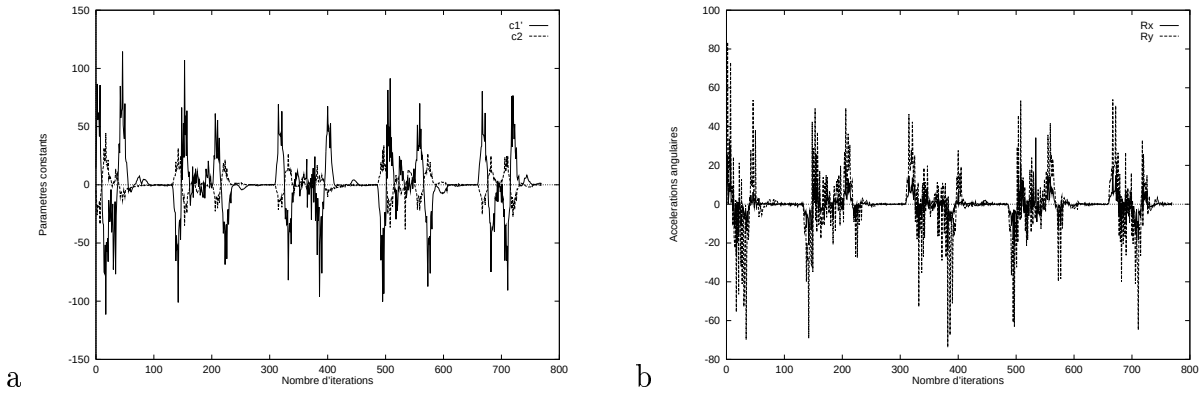


FIG. 4.6 – *Fonction de tâche, paramètres constants du mouvement (a). Commande en accélération angulaire (b)*

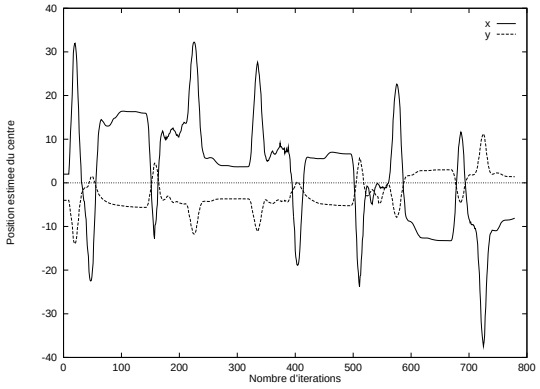


FIG. 4.7 – *Position du centre de l'objet*

Nous constatons, tout d'abord, sur la courbe d'évolution des paramètres constants, que le système oscille beaucoup, même s'il parvient à ramener ces valeurs à zéro. Cela n'est pas lié à un bruit de mesure important sur ces paramètres, mais à un bruit d'estimation sur l'influence du mouvement de l'objet dans l'image, ce qui génère une commande en accélération angulaire très oscillante et très brusque (des sollicitions jusqu'à près de 100 deg/s^2). Tout ceci s'explique en grande partie par la difficulté à estimer des termes liés à l'accélération de l'objet, que le filtre de Kalman n'a pas le temps d'affiner, vu la faible durée des phases d'accélération ou de décélération.

Nous pouvons également noter sur la figure (4.7), qu'il existe un biais sur la position du centre de l'objet. Cela provient des retards inévitables lorsque les paramètres c_1 et c_2 sont ramenés à zéro. En effet pendant ces périodes la vitesse de l'objet dans l'image n'est pas nulle, et le déplacement induit du centre de gravité n'est pas compensé. L'erreur au bout de 800 itérations (une trentaine de secondes) n'est pas catastrophique, puisqu'elle est de l'ordre d'une dizaine de pixels dans chaque direction, mais elle ne peut que croître.

4.1.5 Comparaison avec l'approche par contrôle de la position estimée du centre de l'objet

L'objectif de la tâche présentée ci-dessus est quasiment la même que celle présentée au cours du chapitre 2, à savoir, commander les deux degrés de rotation de la caméra en panoramique et inclinaison, afin de maintenir immobile un objet d'intérêt dans l'image. La seule différence est que, dans le cas où le principe est de contrôler la position estimée du centre de l'objet, une contrainte supplémentaire, consistant à amener cette position au centre de l'image, a été ajoutée.

Une seconde remarque en faveur de la méthode présentée dans le chapitre 2, est qu'elle permet de ramener l'objet au centre de l'image, tandis qu'avec celle du chapitre présent, l'objet est maintenu dans la position où il est apparu dans l'image au début de l'asservissement. Etant donné que généralement un objet mobile apparaît plutôt par un des bords de l'image, le fait de le maintenir dans cette position excentrée, offre de plus grands risques de sortie du champ de vue.

Il existe cependant un avantage à la méthode présentée dans ce paragraphe, en ce qui concerne le besoin de déterminer la position iniitiale de l'objet dans le cas de l'asservissement géométrique, qui n'existe pas dans le cas de l'asservissement visuel dynamique.

Les résultats obtenus sont meilleurs dans le cas de l'asservissement visuel géométrique basé sur des mesures dynamiques. En effet, là où l'erreur de position était inférieure au pixel, elle est de l'ordre de 10 pixels dans le cas présent. Cela permet facilement de déterminer laquelle des deux approches est préférable.

Cependant, nous avons souhaité exposer nos travaux concernant cette approche, car elle valide, malgré tout, dans un cas simple, la possibilité de contrôler un robot selon le schéma d'asservissement visuel dynamique, présenté au chapitre 3. Nous verrons dans

la suite de ce chapitrè deux tâches robotiques - le suivi d'une trajectoire d'observation d'une surface et le positionnement du plan de projection de la caméra parallèlement à un objet planaire - qui, contrairement à la tâche de suivi, ne peuvent pas être définies par des contraintes visuelles géométriques, sans une forte connaissance a priori de l'objet d'intérêt. Dans ces deux cas au moins, la commande du robot par asservissement visuel dynamique est donc nécessaire.

4.2 Suivi rigide d'un objet mobile

4.2.1 Présentation de la tâche

Comme pour le suivi en *pan* et *tilt* présenté dans le paragraphe précédent, il existe des similitudes entre cette tâche et celle que nous avons présenté en simulation dans le chapitre 2, au paragraphe 2.4. L'objectif consiste toujours à maintenir constante au cours du temps, la position tridimensionnelle d'un objet d'intérêt par rapport à la caméra. Et, en s'inspirant de l'idée utilisée pour la tâche précédente, nous cherchons à résoudre ce problème en maintenant nulle la vitesse de l'objet dans l'image en contrôlant les six degrés de liberté de la caméra.

Egalement, comme pour la poursuite en *pan* et *tilt*, cette tâche se situe dans le cas où la matrice d'interaction est de rang plein, 6 ici. Cependant, le vecteur d'informations visuelles dynamiques doit être correctement choisi pour obtenir cette condition.

Nous présenterons dans ce paragraphe, des résultats de simulation concernant cette tâche. Nous ne donnerons pas de résultats sur site robotique, car, comme nous le verrons, il est nécessaire de mesurer les paramètres quadratiques du mouvement. Une telle mesure ne peut être fiable qu'à une cadence de 1 Hz. Cette lenteur de l'estimation vient en contradiction avec la nécessité pour le système de réagir rapidement aux perturbations générées par le mouvement de l'objet mobile. Cela pose également un problème avec le fait que la loi de commande est spécifiée en termes d'accélération de la caméra, et il existe donc un risque d'obtenir des vitesses élevées avant que la commande ne soit rafraîchie.

4.2.2 Loi de commande associée

Pour assurer un suivi rigide de l'objet d'intérêt, il ne faut plus simplement que sa position - c'est-à-dire celle de son centre ou d'un quelconque autre point - reste inchangée dans l'image, mais, de plus, sa projection ne doit plus subir de déformations. Une première idée consiste à assimiler cette contrainte à l'absence de mouvement affine dans l'image. Cela revient donc à choisir $s = (c_1,\ c_2,\ a_1,\ a_2,\ a_3,\ a_4)^T$ comme vecteur de tâche (cf. équation (1.6)), dont la valeur désirée s^* correspond au vecteur nul de dimension 6.

Nous verrons cependant qu'il est nécessaire de considérer également les termes quadratiques.

Relation d'interaction

Nous avons vu dans le cas précédent, qu'il était inutile de faire apparaître explicitement les paramètres du mouvement propre de l'objet dans la relation d'interaction liant les variations de s au mouvement de la caméra. En fait, il est beaucoup plus approprié de rassembler l'ensemble des facteurs fonction de la vitesse ou de l'accélération de l'objet, dans un terme générique représentant l'influence du mouvement de l'objet. Ainsi, en se basant sur la relation (3.2), nous pouvons écrire :

$$\dot{s} = L\ \Gamma + \frac{\partial s}{\partial t} \tag{4.3}$$

avec :

$$L = \begin{bmatrix} -\frac{1}{Z_p} & 0 & 0 & 0 & -1 & 0 \\ 0 & -\frac{1}{Z_p} & 0 & 1 & 0 & 0 \\ \frac{\gamma_1}{Z_p} & 0 & \frac{1}{Z_p} & 0 & 0 & 0 \\ \frac{\gamma_2}{Z_p} & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{\gamma_1}{Z_p} & 0 & 0 & 0 & -1 \\ 0 & \frac{\gamma_2}{Z_p} & \frac{1}{Z_p} & 0 & 0 & 0 \end{bmatrix}$$

Le déterminant de la matrice L vaut :

$$\det L = \frac{\gamma_1^2 + \gamma_2^2}{Z_p^2}$$

Par conséquent, L sera de rang plein si et seulement si γ_1 ou γ_2 est non-nul, ce qui équivaut à ce que le plan de l'objet ne soit pas parallèle au plan de projection de l'image. Cette condition peut, au premier abord, sembler simple à remplir. Cependant, la position de l'objet parallèlement au plan image est celle qui permet la meilleur observation, c'est-à-dire que pour une profondeur Z_p donnée, elle génère la projection de taille maximale. C'est donc, a priori, une position qui pourra être souvent souhaitée lors d'une tâche robotique. Et, même si dans la position de départ, les valeurs de γ_1 et γ_2 ne sont pas nulles, rien ne garantit, que le système ne se retrouve pas, au cours de l'asservissement, dans une configuration où le plan de projection de la caméra est parallèle à l'objet à suivre ; la matrice L devient alors singulière. De plus, l'estimation des termes γ_1 et γ_2 n'est pas simple à réaliser en pratique, car elle dépend d'estimations préalable (voir équation

(3.16)). Aussi, lorsque ces deux valeurs sont faibles, une approximation par zéro peut être préférable à une estimation biaisée. Dans ce cas, la matrice $\widehat{L}$, approximant L est de rang 4, même si L est bien de rang 6.

Pour toutes ces raisons, il nous est paru plus judicieux de considérer un nouveau vecteur de tâche :

$$s = (c_1,\ c_2,\ a_1,\ a_2,\ a_3,\ a_4, q_1, q_2)^T$$

dont la valeur désirée s^* est le vecteur nul de dimension 8.

Avec ce nouveau vecteur, la relation d'interaction devient :

$$\dot{s} = L\ \Gamma + \frac{\partial s}{\partial t} \tag{4.4}$$

avec :

$$L = \begin{bmatrix} -\frac{1}{Z_p} & 0 & 0 & 0 & -1 & 0 \\ 0 & -\frac{1}{Z_p} & 0 & 1 & 0 & 0 \\ \frac{\gamma_1}{Z_p} & 0 & \frac{1}{Z_p} & 0 & 0 & 0 \\ \frac{\gamma_2}{Z_p} & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{\gamma_1}{Z_p} & 0 & 0 & 0 & -1 \\ 0 & \frac{\gamma_2}{Z_p} & -\frac{1}{Z_p} & 0 & 0 & 0 \\ 0 & 0 & -\frac{\gamma_1}{Z_p} & 0 & -1 & 0 \\ 0 & 0 & -\frac{\gamma_2}{Z_p} & 1 & 0 & 0 \end{bmatrix}$$

Le rang de cette nouvelle matrice L est de 6, quelques soient les valeurs des termes γ_1 et γ_2. Le vecteur de tâche est maintenant sur-dimensionné par rapport au nombre d'axes du robot que nous souhaitons contrôler. Aussi, nous définissons un vecteur d'erreur e associé à s sous la forme :

$$e = C\ S \qquad \text{avec} \qquad C = \widehat{L}^+$$

où $\widehat{L}$ représente une approximation de L, et $\widehat{L}^+$ sa pseudo-inverse. Nous choisissons d'approximer la matrice L en considérant les termes γ_1 et γ_2 nuls, et en laissant l'opérateur estimer grossièrement la profondeur Z_p par une valeur $\widehat{Z_p}$.

Nous pouvons donc écrire la relation d'interaction (4.4) sous la forme :

$$\dot{e} = CL\Gamma + C\frac{\partial s}{\partial t} \tag{4.5}$$

Loi de commande

A partir de la relation (4.5), nous pouvons établir une loi de commande de type gradient en imposant un décroissance exponentielle de gain λ de l'erreur. Nous obtenons :

$$\Gamma = -\lambda\widehat{e} - C\widehat{\frac{\partial s}{\partial t}} \tag{4.6}$$

où $\widehat{e}$ est l'estimation de e résultant de la mesure du vecteur s et de l'approximation de la matrice L.

L'estimation du terme représentant l'influence du mouvement propre de l'objet est réalisée sur un schéma similaire à celui présenté au paragraphe précédent et dans l'équation (3.18), avec :

$$\widehat{\frac{\partial s}{\partial t}} = \widehat{\frac{ds}{dt}} - \widehat{L}\widehat{\Gamma}$$

où $\widehat{\Gamma}$ représente une approximation de Γ obtenue par odométrie.

Analyse de la convergence

La loi de commande précédente correspond à la forme générale donnée à l'équation (3.13), pour laquelle nous avons vu que la convergence est atteinte si la matrice $K = (CL)(C\widehat{L})^{-1}$ est suffisament proche de l'identité (cf. relation (3.14)). Dans le cas présent, on a :

$$K = \widehat{L}^{+}L = \begin{bmatrix} \frac{\widehat{Z}_p}{Z_p} & 0 & -\frac{\gamma_1\widehat{Z}_p}{Z_p} & 0 & 0 & 0 \\ 0 & \frac{\widehat{Z}_p}{Z_p} & -\frac{\gamma_2\widehat{Z}_p}{Z_p} & 0 & 0 & 0 \\ \frac{\gamma_1\widehat{Z}_p}{2Z_p} & \frac{\gamma_2\widehat{Z}_p}{2Z_p} & \frac{\widehat{Z}_p}{Z_p} & 0 & 0 & 0 \\ 0 & 0 & -\frac{\gamma_2\widehat{Z}_p}{Z_p} & 1 & 0 & 0 \\ 0 & 0 & \frac{\gamma_1\widehat{Z}_p}{Z_p} & 0 & 1 & 0 \\ \frac{\gamma_2\widehat{Z}_p}{2Z_p} & -\frac{\gamma_1\widehat{Z}_p}{2Z_p} & 0 & 0 & 0 & 1 \end{bmatrix}$$

où $\widehat{Z}_p$ représente une estimation de la profondeur Z_p.

Les conditions de positivité de cette matrice K sont difficiles à déterminer précisément. Disons simplement que si γ_1 et γ_2 sont faibles (inférieurs à 0.2, c'est-à-dire des angles inférieurs à 10 degrés), la convergence est assurée si l'estimation $\widehat{Z}_p$ de Z_p est de signe correct. En effet, si $\gamma_1 = \gamma_2 = 0$, nous obtenons :

$$K = \begin{bmatrix} \dfrac{\widehat{Z}_p}{Z_p}\,\mathbb{I}_3 & 0 \\ 0 & \mathbb{I}_3 \end{bmatrix}$$

Nous avons effectué une étude expérimentale, en simulation, de la stabilité et de la convergence, que nous présentons à la fin du paragraphe suivant. Il s'avère, en fait, que l'erreur maximale admissible sur les valeurs de γ_1 et γ_2 est de l'ordre de 0.5, ce qui correspond à une erreur angulaire de 25 degrés environ.

4.2.3 Résultats de simulation

Les conditions de la simulation sont rigoureusement les mêmes que dans le cas du suivi en *pan* et *tilt* présenté au paragraphe précédent. En particulier, le mouvement de l'objet est représenté sur la figure (4.1). La valeur du gain λ est de 5.

Dans les résultats que nous présentons dans un premier temps, les valeurs initiales des termes γ_1 et γ_2, et qui donc, doivent être conservées au cours de l'asservissement, sont nulles. Il n'y a donc pas d'erreur d'approximation sur ces termes. En revanche, la profondeur Z_p est de 1 m au départ, et son approximation $\widehat{Z_p}$ est fixée à 1.25 m, ce qui représente une erreur de 25%.

Nous montrons différentes courbes correspondant à cette simulation. Il s'agit tout d'abord des paramètres du mouvement, présentés sur quatre figures distinctes : les termes constants sur la figure (4.8) a, avec c_1 en trait plein et c_2 en trait pointillé ; les termes affines dépendant de $\dot{x}$ sur la figure (4.8) b, avec a_1 en trait plein et a_2 en trait pointillé ; les termes affines dépendant de $\dot{y}$ sur la figure (4.9) a, avec a_3 en trait plein et a_4 en trait pointillé ; et enfin, les termes quadratiques sur la figure (4.9) b, avec q_1 en trait plein et q_2 en trait pointillé. Ensuite, nous présentons sur la figure (4.10), les accélérations données par la loi de commande, avec les translations (a), en m/s^2, et les rotations (b), en deg/s^2, où les composantes en x sont en trait plein, celles en y en trait pointillé, et celles en z en trait discontinu.

Il apparaît clairement, sur chacune des courbes représentant les composantes de la fonction de tâche, que l'objectif est atteint, c'est-à-dire que chacune de ces huit valeurs est rapidement ramenée à zéro, après chaque variation brusque du mouvement de l'objet. Ce résultat est obtenu malgré une légère oscillation des termes d'accélération en translation, due à l'erreur d'estimation de la profondeur.

Simulation. Suivi rigide d'un objet mobile par asservissement visuel dynamique

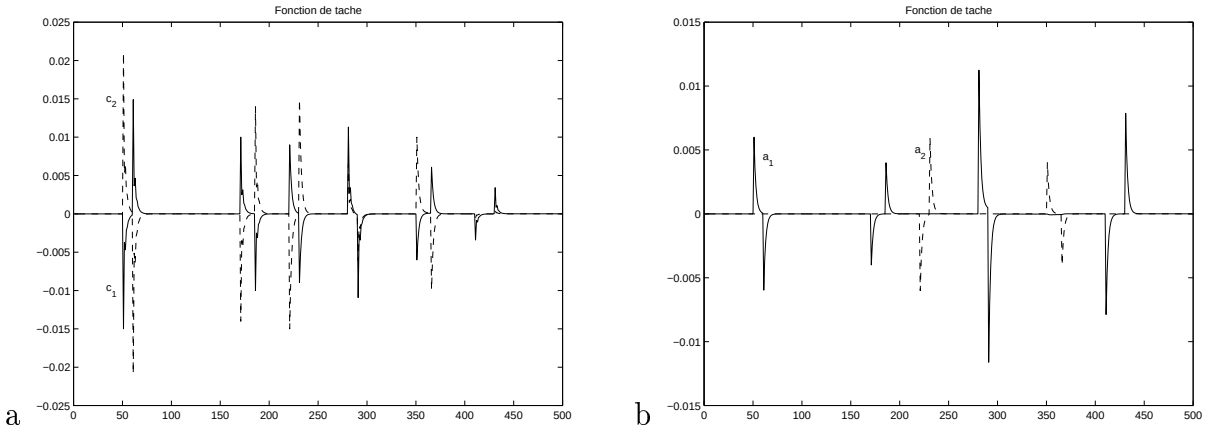


FIG. 4.8 – *Fonction de tâche : paramètres constants (a) et paramètres affines du développement de $\dot{x}$ (b)*

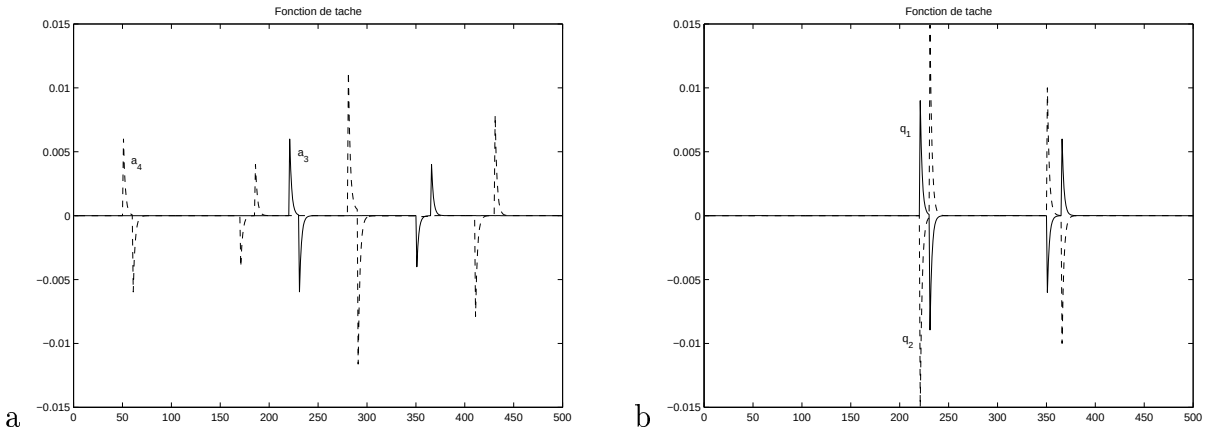


FIG. 4.9 – *Fonction de tâche : paramètres affines du développement de $\dot{y}$ (a) et paramètres quadratiques (b)*

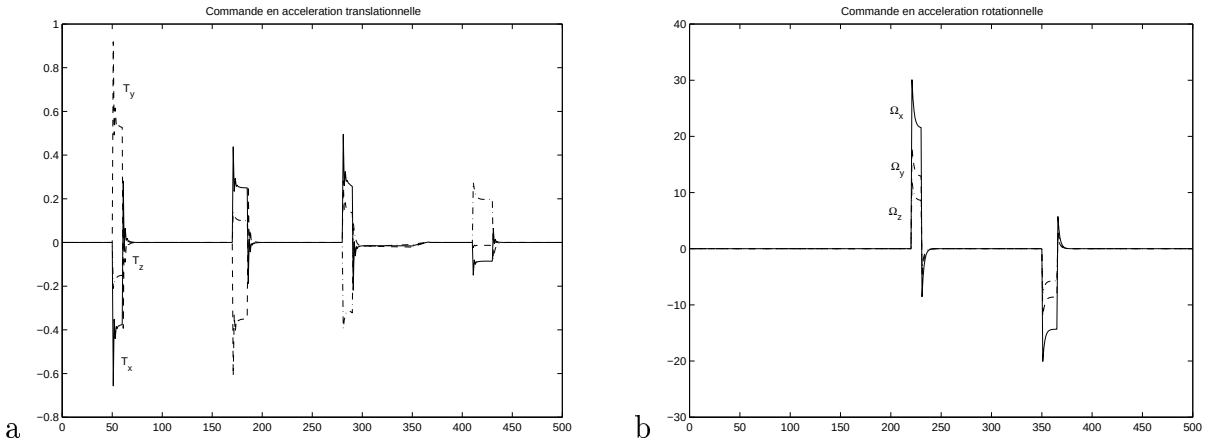


FIG. 4.10 – *Commande en accélération : translation (a) et rotation (b)*

Pour confirmer ces résultats, nous proposons une autre courbe, sur la figure (4.11). Il s'agit du déplacement (en m) du point de l'objet se projetant au début de l'asservissement à la position $(x,\ y)^T = (0.2,\ 0.2)^T$. La composante en x est en trait plein, celle en y, en trait pointillé.

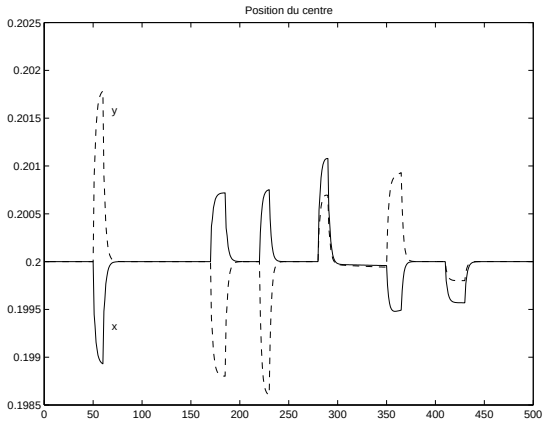


FIG. 4.11 – *Position d'un point de l'objet dans l'image*

Cette courbe confirme la qualité du suivi obtenu : l'erreur maximale est inférieure à 2 mm, et elle est nulle à la fin de l'asservissement, lorsque l'objet s'est à nouveau arrêté.

Influence de l'erreur d'approximation de L

Les résultats précédents ont été obtenus dans un cas où le biais d'estimation de L n'était dû qu'à l'erreur d'approximation de la profondeur. Afin de déterminer les limites de notre algorithme de suivi aux erreurs d'approximation des termes d'orientation γ_1 et γ_2, nous avons effectué des simulations dans un contexte similaire. La seule différence réside dans le fait que la valeur initiale de γ_1 n'est plus nulle. L'approximation de L continue pourtant à être obtenue avec des valeurs nulles pour ces deux termes.

Remarque : Il n'y a pas de perte de généralité à considérer seulement γ_1 non-nul. Les résultats sont identiques si γ_2 est inférieur ou égal à γ_1, en valeur absolue, ou s'il reste nul. En fait, seul importe la valeur maximale d'erreur sur l'un ou l'autre de ces deux paramètres.

Les résultats de cette analyse sont rapportés dans la figure (4.12), avec, respectivement, pour chaque plage de valeurs de γ_1, la plage angulaire approximative correspondant en degrés, le comportement du système et une courbe représentant la commande en accélération translationnelle illustrant l'allure de cette réponse, et ce pour la valeur la plus élevée de la plage considérée (sauf pour la dernière plage où γ_1 est choisi égal à 2).

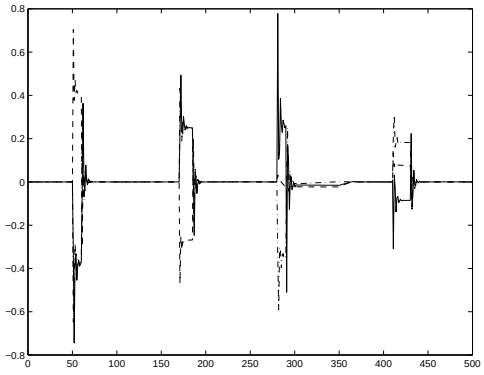
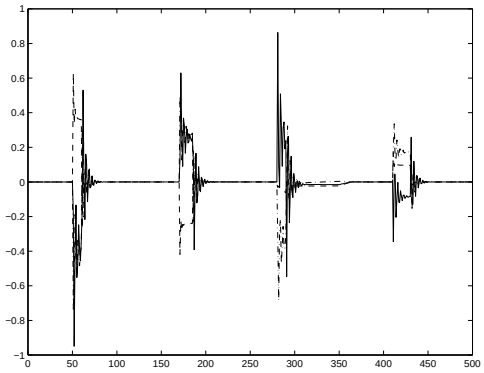
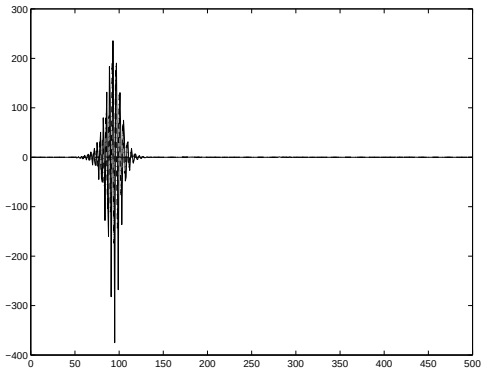
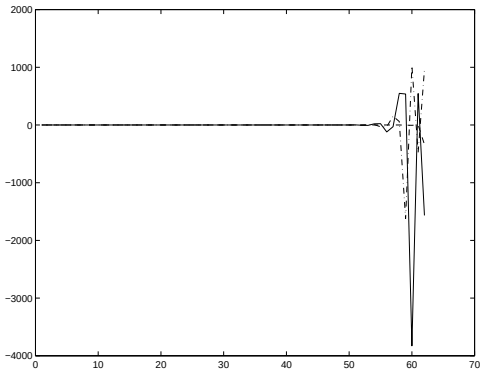
Valeur de γ_1	Comportement du système	Accélération en translation
$0.0 \rightarrow 0.5$ $0 \rightarrow 25$ deg	Convergence stable	
$0.5 \rightarrow 0.65$ $25 \rightarrow 35$ deg	Convergence Oscillations importantes	
$0.65 \rightarrow 1.0$ $35 \rightarrow 45$ deg	Oscillations irréalistes	
$1.0 \rightarrow +\infty$ $45 \rightarrow 90$ deg	Divergence	

FIG. 4.12 – *Comportement du système avec des valeurs de γ_1 non nulles*

Remarque : Ce que nous appelons "Oscillations irréalistes" désigne un comportement du système, où si la convergence est obtenue en simulation, il apparaît impossible qu'elle puisse l'être dans des conditions réelles, même en supposant l'absence de bruit de mesure.

Nous pouvons constater que l'erreur angulaire acceptable est de 25 degrés, ce qui est relativement faible. Cela indique qu'une phase d'estimation de l'orientation du plan est souhaitable avant de débuter le suivi. Elle peut être effectuée, par exemple, grâce à la mesure du modèle quadratique du mouvement, lorsque la caméra se déplace le long de son axe optique. En effet, les paramètres quadratiques valent alors $(-\gamma_1 v_z)$ et $(-\gamma_2 v_z)$ (cf. équation (1.6)), la valeur de v_z pouvant être obtenue à partir des paramètres affines a_1 et a_4, auxquels elle est alors égale (cf. équation (1.6)). A noter que cela nous permet également d'estimer la profondeur. Cependant, une telle estimation ne peut être faite que lorsque l'objet est immobile, et que ce caractère immobile est connu.

4.3 Positionnement de la caméra parallèlement à un plan

4.3.1 Présentation de la tâche

Pour cette nouvelle tâche, le but est d'amener le plan de projection de l'image parallèle à un objet supposé planaire et immobile dans un repère fixe galiléen. Cet objectif de la tâche est schématisé sur la figure 4.13 sur laquelle:

- $\overrightarrow{n}$ représente le vecteur unitaire normal au plan de l'objet d'intérêt ;
- $\overrightarrow{Z_c}$ est le vecteur directeur de l'axe optique de la caméra, supposé orthogonal au plan de projection Π des images ;
- les exposants * sont associés à la position désirée de la caméra, tandis que les éléments associés à sa position initiale sont indiqués sans exposant ;
- les points P_i et P_d sont les points 3D de la surface de l'objet qui se projettent au centre de l'image, respectivement aux positions initiales et désirées.

Remarque: La position désirée n'est bien sûr pas unique. La caméra doit, en réalité, se positionner dans un sous-espace de l'espace euclidien dans lequel elle évolue. Cependant, par abus de langage, nous dirons, lorsque la caméra est dans ce sous-espace, qu'elle a atteint sa position désirée.

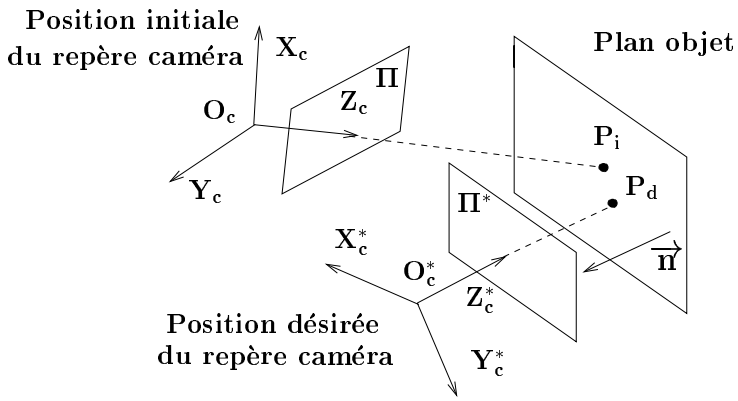


FIG. 4.13 – *Contexte et résultat attendu de la tâche*

Deux tâches différentes

Nous décrivons deux tâches associées à l'objectif fixé. La première consiste à simplement remplir la condition de parallélisme du plan Π et de l'objet. Cela revient à aligner l'axe optique $\overrightarrow{Z_c}$ de la caméra avec la normale $\overrightarrow{n}$ au plan de l'objet. Pour parvenir à cet objectif, les deux axes de rotation de la caméra orthogonaux à l'axe optique sont commandés. Nous appellerons cette tâche, la tâche d'alignement de la caméra.

Une seconde approche pour répondre au problème posé consiste à associer à la première tâche, décrite ci-dessus, une contrainte supplémentaire sur le positionnement de la caméra par rapport au plan observé. Il s'agit dans ce cas de compenser le mouvement apparent de l'objet dans l'image, dû à la rotation de la caméra, par des translations le long des axes $\overrightarrow{X_c}$ et $\overrightarrow{Y_c}$ de la caméra. Plus précisément, le but est de maintenir le même point au centre de l'image au cours de l'asservissement, c'est-à-dire, en particulier, que les points P_i et P_d doivent être identiques. La tâche jointe à la tâche d'alignement sera appelée dorénavant, la fixation.

Dans ce cadre, deux cas différents étudiés au chapitre 3 apparaissent. Tout d'abord, lorsque seul l'alignement est pris en compte, nous nous trouvons dans le cas particulier où la dérivée du vecteur de mesure ne dépend pas de l'accélération, c'est-à-dire que la matrice d'interaction L est nulle. Par contre, si la commande concerne également la fixation, la matrice d'interaction est alors de rang non-plein.

4.3.2 Loi de commande associée à la tâche d'alignement

Soit $Z = Z_p + \gamma_1 X + \gamma_2 Y$, l'équation du plan objet dans le repère de la caméra. La condition de parallélisme entre ce plan et celui de projection de la caméra s'énonce simplement sous la forme :

$$\left(\left(\frac{\partial Z}{\partial X}\right), \left(\frac{\partial Z}{\partial Y}\right)\right)^T = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4.7}$$

Si nous reprenons le développement au second ordre du mouvement projeté d'un objet (cf. l'équation 1.6 au chapitre 1), nous voyons que les termes d'orientation du plan observé, γ_1 et γ_2 apparaissent aussi bien dans les termes affines a_i, que dans les termes quadratiques q_i. Plus précisément, en tenant compte du fait que l'objet est immobile, et donc que son mouvement relatif par rapport à la caméra est l'opposé de celui de la caméra par rapport à un repère fixe, nous avons :

$$\begin{cases} a_1 = \gamma_1 v_x + v_z & a_2 = \gamma_2 v_x + \Omega_z \\ a_3 = \gamma_1 v_y - \Omega_z & a_4 = \gamma_2 v_y + v_z \\ q_1 = -\gamma_1 v_z - \Omega_y & q_2 = -\gamma_2 v_z + \Omega_x \end{cases} \tag{4.8}$$

en considérant, pour plus de simplicité dans les notations, que dorénavant le torseur cinématique de la caméra est représenté par $(T, \Omega)^T$ et le vecteur $v = (v_x, v_y, v_z)$ comme étant la translation de la caméra, au facteur d'échelle près de la profondeur de l'objet, $v = \frac{1}{Z_p} T$.

Une mauvaise voie

Une première idée consiste à considérer le cas où v_x et v_y ne sont pas simultanément nuls. La condition (4.7) est alors équivalente à :

$$\begin{bmatrix} v_x & -v_y \\ v_y & v_x \end{bmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = M \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} a_1 - a_4 \\ a_2 + a_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4.9}$$

car, dans ce cas, la matrice M est toujours inversible.

Dans l'article [Questa 95] où l'objectif est également de se positionner parallèlement à un plan, c'est cette modélisation qui est utilisée.

Remarque : Il faut noter que les termes $(a_1 - a_4)$ et $(a_2 + a_3)$ sont, à un facteur 2 près, les termes hyperboliques du développement affine par descripteur du mouvement donnés à l'équation (1.7).

A partir des dérivées des paramètres du mouvement, données au paragraphe 3.2, il est alors possible de calculer le lien entre les variations de ces termes hyperboliques et les mouvements de la caméra, et ce, afin de contrôler leur décroissance vers une valeur nulle.

Cependant, cette approche, même si elle est correcte en théorie, ne nous paraît pas applicable en pratique. En effet, elle nécessite le fait que la composante de la translation

de la caméra orthogonalement à l'axe optique ne puisse jamais être nulle. Cette contrainte est incompatible avec la tâche de fixation, puisqu'à la convergence, les rotations autour des axes $\overrightarrow{X_c}$ et $\overrightarrow{Y_c}$ sont nulles. Il en est donc de même des translations visant à compenser leurs effets. Il survient donc une contradiction au voisinage de la convergence, car les mesures dynamiques extraites des images, même si elles sont nulles, ne permettent pas d'assurer que la tâche est remplie.

Un choix plus judicieux de la fonction de tâche

Les termes γ_1 et γ_2 apparaissent également dans les termes quadratiques de l'équation (1.6), où ils sont alors facteurs de la translation le long de l'axe optique. Si cette translation v_z est non nulle, la condition (4.7) est alors équivalente à :

$$\begin{pmatrix} q_1 + \Omega_y \\ q_2 + \Omega_x \end{pmatrix} = - \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} v_z = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Ce choix paraît plus pertinent que le précédent, puisque dans ce cas, la nécessité de conserver v_z non nul ne vient pas en contradiction avec l'objectif de la tâche que nous considérons, même lorsque la fixation est prise en compte. L'éventuel problème qui peut se poser est celui de la collision de la caméra avec l'objet, lorsque la translation le long de l'axe optique est dirigée vers la cible. Ce problème peut être résolu par une commande de la translation T_z qui permet de maintenir constant le temps à collision ([Cipolla 97b]).

Le vecteur de mesure choisi est donc :

$$s = \begin{pmatrix} q_1 + \Omega_y \\ q_2 - \Omega_x \end{pmatrix} \quad \text{noté } s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

La valeur désirée s^* pour ce vecteur est $s^* = (0,0)^T$. Le vecteur d'erreur e est choisi égal à s, c'est-à-dire, $C = \mathbb{I}_2$.

Les équations (3.2) correspondant aux termes quadratiques du mouvement, nous permettent d'établir la relation d'interaction associée à s. Ainsi, nous obtenons :

$$\dot{s} = v_z \begin{pmatrix} -\Omega_y + \gamma_1(\gamma_1 v_x + \gamma_2 v_y - v_z) - \gamma_2 \Omega_z + \gamma_1 \dfrac{\dot{T}_z}{Z_p} \\ \\ \Omega_x + \gamma_2(\gamma_1 v_x + \gamma_2 v_y - v_z) + \gamma_1 \Omega_z + \gamma_2 \dfrac{\dot{T}_z}{Z_p} \end{pmatrix} \tag{4.10}$$

Nous choisissons dans un premier temps de considérer la translation T_z constante, c'est-à-dire l'accélération $\dot{T}_z$ nulle.

Nous nous trouvons dans le cas où la matrice d'interaction L est nulle, puisque $\dot{s}$ ne dépend pas des accélérations angulaires. Nous considérons donc le vecteur de commande $(\Omega_x, \Omega_y)^T$. La relation précédente peut alors s'écrire sous forme matricielle :

$$\dot{s} = M \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} + h \tag{4.11}$$

avec :

$$M = \begin{bmatrix} 0 & -v_z \\ v_z & 0 \end{bmatrix}$$

Nous notons $h = \delta_m + \delta_s$ avec :

$$\delta_m = \begin{pmatrix} -\gamma_2 v_z \Omega_z \\ \gamma_1 v_z \Omega_z \end{pmatrix} = \Omega_z \begin{pmatrix} s_2 \\ -s_1 \end{pmatrix}$$

$$\text{et} \quad \delta_s = \begin{pmatrix} \gamma_1 v_z(\gamma_1 v_x + \gamma_2 v_y - v_z) \\ \gamma_2 v_z(\gamma_1 v_x + \gamma_2 v_y - v_z) \end{pmatrix} = (v_z - \gamma_1 v_x - \gamma_2 v_y)s = \alpha s$$

Loi de commande

Dans la relation (4.11), les variations de s ne dépendent pas de l'accélération de la caméra. Il n'apparaît donc que des termes quadratiques en la vitesse de la caméra. Nous devons donc, si cela est possible, séparer les termes qui dépendent des vitesses de rotation autour des axes sur lesquels la commande s'applique, de ceux dépendant des axes non-contraints qui seront alors considérés comme des informations mesurées de notre système. Dans notre cas, les vitesses de rotation Ω_x et Ω_y sont uniquement facteurs de la translation T_z qui n'est pas commandée. Par conséquent, il est possible de spécifier un schéma de commande de type gradient, en spécifiant une décroissance exponentielle de l'erreur. Nous obtenons donc :

$$\begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} = -\widehat{M}^{-1} \ (\lambda s + \delta_s + \delta_m)$$

où l'estimation $\widehat{M}$ de M est donnée par :

$$\widehat{M} = \begin{bmatrix} 0 & -\widehat{v_z} \\ \widehat{v_z} & 0 \end{bmatrix}$$

dans laquelle nous pouvons utiliser l'approximation de v_z (cf. (1.6)) suivante :

$$v_z \approx \frac{a_1 + a_4}{2}$$

Cependant, l'approximation précédente peut être améliorée, en supposant que nous avons des mesures odométriques fiables des vitesses de translation de la caméra. En effet (voir équation (1.6)), la relation exacte liant v_z aux paramètres affines du mouvement est :

$$\begin{aligned} a_1 + a_4 &= 2v_z + \gamma_1 v_x + \gamma_2 v_y \\ &= 2v_z - s_1 \frac{\widehat{T_x}}{\widehat{T_z}} - s_2 \frac{\widehat{T_y}}{\widehat{T_z}} \end{aligned}$$

où $\widehat{T}$ représente la valeur mesurée par odométrie du vecteur T, et $\widehat{s}$ est l'estimation de s déduite de la mesure des paramètres quadratiques du mouvement et de la rotation Ω. Ainsi, nous considérerons désormais que v_z peut être estimé par :

$$\widehat{v_z} = \frac{1}{2}\left(a_1 + a_4 + \widehat{s}_1 \frac{\widehat{T_x}}{\widehat{T_z}} + \widehat{s}_2 \frac{\widehat{T_y}}{\widehat{T_z}}\right) \tag{4.12}$$

En utilisant le même procédé, nous pouvons approximer δ_s par :

$$\widehat{\delta_s} = \left(\widehat{v_z} + \widehat{s}_1 \frac{\widehat{T_x}}{\widehat{T_z}} + \widehat{s}_2 \frac{\widehat{T_y}}{\widehat{T_z}}\right)\widehat{s} \quad \text{noté} \quad \widehat{\alpha}\widehat{s} \tag{4.13}$$

En définitive, la loi de commande peut s'écrire :

$$\begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} = \frac{1}{\widehat{v_z}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \left((\lambda + \widehat{\alpha}) \begin{pmatrix} \widehat{s}_1 \\ \widehat{s}_2 \end{pmatrix} + \widehat{\Omega_z} \begin{pmatrix} \widehat{s}_2 \\ -\widehat{s}_1 \end{pmatrix}\right) \tag{4.14}$$

Analyse de la convergence

En remplaçant, par leur valeur donné par la loi de commande (4.14), les vitesses angulaires dans la relation d'interaction (4.11), nous obtenons :

$$\dot{s} = -M\widehat{M}^{-1}\left((\lambda + \widehat{\alpha})\widehat{s} + \widehat{\Omega_z}\begin{pmatrix} \widehat{s}_2 \\ -\widehat{s}_1 \end{pmatrix}\right) + \alpha s + \Omega_z \begin{pmatrix} s_2 \\ -s_1 \end{pmatrix}$$

En supposant que toutes les estimations effectuées sont parfaites, le comportement du système bouclé se réduit donc à :

$$\dot{s} = -\lambda s$$

La positivité de la matrice $K = M\widehat{M}^{-1}$, permet d'assurer la convergence. La matrice M et son estimation $\widehat{M}$ étant toujours de rang 2,la matrice K est donnée par :

$$K = \frac{v_z}{\widehat{v_z}}\mathbb{I}_2$$

La convergence sera donc assurée pour peu que l'estimation $\widehat{v_z}$ de v_z soit de signe correct.

Contrôle du temps à collision

Nous avons vu que lorsque nous supposons la translation T_z non nulle, le problème qui peut se poser est celui de la collision de la caméra avec l'objet d'intérêt - dans le cas, bien sûr, où cette translation amène la caméra dans la direction de l'objet. Or si nous considérons cette vitesse constante, le risque de collision est inévitable.

Considérons maintenant le cas où nous souhaitons contrôler la translation T_z de manière à maintenir le temps à collision t_c constant, ce qui revient à maintenir v_z constant, étant donné que $v_z = \frac{1}{t_c}$ (cf. 1.9). Le vecteur de tâche devient alors :

$$s = \begin{pmatrix} q_1 - \Omega_y \\ q_2 + \Omega_x \\ v_z \end{pmatrix}$$

La valeur désirée s^* pour ce vecteur peut prendre deux formes selon que l'on souhaite maintenir le temps à collision à sa valeur initiale $t_{c,0}$ ou le contraindre à atteindre une valeur prédéfinie t_c^*. Dans le second cas, nous avons :

$$s^* = \left(0, 0, \frac{1}{t_c^*}\right)^T$$

Le vecteur d'erreur considéré est $e = (s - s^*)$.

En utilisant l'équation (4.10) et les calculs intermédiaires qui nous ont permis d'aboutir aux relations (3.2), nous pouvons déterminer la dérivée du vecteur s :

$$\dot{s} = \begin{pmatrix} v_z \left(-\Omega_y + \gamma_1(\gamma_1 v_x + \gamma_2 v_y - v_z) - \gamma_2 \Omega_z + \gamma_1 \frac{\dot{T}_z}{Z_p}\right) \\ v_z \left(\Omega_x + \gamma_2(\gamma_1 v_x + \gamma_2 v_y - v_z) + \gamma_1 \Omega_z + \gamma_2 \frac{\dot{T}_z}{Z_p}\right) \\ -v_z \left(\gamma_1 v_x + \gamma_2 v_y + \gamma_1 \Omega_y - \gamma_2 \Omega_x - v_z\right) + \frac{\dot{T}_z}{Z_p} \end{pmatrix}$$

De même que dans le cas précédent, les accélérations angulaires $\dot{\Omega}_x$ et $\dot{\Omega}_y$ n'apparaissent pas dans le développement de $\dot{s}$. Ces deux termes sont donc des vecteurs propres du noyau de la matrice d'interaction L qui lie $\dot{s}$ avec le vecteur des accélérations le long des axes que l'on souhaite commander (c'est-à-dire les deux rotations autour des axes perpandiculaires à l'axes optique et la translation le long de cet axe optique). Comme nous l'avons vu au chapitre précédent, il faut dans ce cas-là considérer un nouveau vecteur de commande où apparaisse les vitesses correspondant aux accélérations présentes dans le noyau de L.

Nous écrivons donc la relation précédente, en découplant la commande en deux termes de vitesse angulaire, Ω_x et Ω_y, et un terme d'accélération en translation $\dot{T}_z$. Pour cela, en

effectuant les mêmes approximations que précédemment au voisinage de la convergence, nous écrivons la relation précédente sous la forme :

$$\dot{s} = M \begin{pmatrix} \Omega_x \\ \Omega_y \\ \dot{T}_z \end{pmatrix} + v_z s + \delta_m$$

avec :

$$M = \begin{bmatrix} 0 & -v_z & \frac{\gamma_1 v_z}{Z_p} \\ v_z & 0 & \frac{\gamma_2 v_z}{Z_p} \\ \gamma_2 v_z & -\gamma_1 v_z & \frac{1}{Z_p} \end{bmatrix} = \begin{bmatrix} 0 & -s_3 & -\frac{s_1}{Z_p} \\ s_3 & 0 & -\frac{s_2}{Z_p} \\ -s_2 & s_1 & \frac{1}{Z_p} \end{bmatrix} \quad \text{et} \quad \delta_m = \Omega_z \begin{pmatrix} -s_2 \\ s_1 \\ 0 \end{pmatrix}$$

La loi de commande obtenue en imposant une décroissance exponentielle de l'erreur e est alors :

$$\begin{pmatrix} \Omega_x \\ \Omega_y \\ \dot{T}_z \end{pmatrix} = -\widehat{M}^{-1}(\lambda \widehat{e} + \widehat{v_z}\widehat{s} + \delta_m) \tag{4.15}$$

où l'estimation $\widehat{e}$ de e est effectuée à partir de celle de s. Cette estimation $\widehat{s}$ de s est obtenue comme précédemment pour les deux premiers termes, et comme en (4.12) pour s_3 égal à v_z. De l'estimation de v_z et d'une mesure de T_z par odométrie, nous pouvons déduire une estimation de la profondeur Z_p. De l'ensemble de ces valeurs mesurées ou estimées, il est alors possible de déduire l'approximation $\widehat{M}$ de la matrice M. Enfin, le facteur $\widehat{\Omega_z}$ qui apparaît dans le terme δ_m est, comme précédemment, mesuré par odométrie.

Dans ce cas, la commande est spécifiée à la fois en accélération et en vitesse. Nous ne pouvons donc pas donner de conditions permettant d'assurer la convergence de l'erreur vers zéro.

4.3.3 Loi de commande associée à la tâche d'alignement couplée avec la fixation

Dans le cas de la tâche d'alignement, nous ne sommes pas intéressés à la question de la dimension de l'objet d'intérêt. Cependant, si nous supposons raisonnablement que cette dimension est limitée, il apparaît un problème lié à l'observation de ce plan. En effet, la commande en rotation génère un déplacement de l'objet dans l'image. Donc, lorsque l'erreur initiale est importante, ou que le plan observé est de dimension restreinte, ce

déplacement peut conduire l'objet à sortir du champ de vue de la caméra. C'est pourquoi nous avons associé à la tâche d'alignement une contrainte de fixation qui consiste à maintenir le même point au centre de l'image au cours de l'asservissement.

Plusieurs approches sont possibles pour résoudre ce problème ; nous en proposerons trois. La première se base sur une compensation directe de l'influence de la rotation par une commande en boucle ouverte. La seconde utilise un schéma d'asservissement visuel dynamique visant à maintenir la vitesse apparente du centre de l'image nulle. Enfin, la troisième méthode se base sur les résultats présentés au chapitre 2, et consiste à retrouver à chaque itération la position du point initialement au centre de l'image afin de la contraindre à demeurer identique à sa valeur initiale.

Commande en boucle ouverte

Pour définir cette commande, les tâches d'alignement et de fixation sont complètement séparées. Plus exactement, la tâche d'alignement est réalisée exactement avec la même loi de commande que celle qui est présentée dans l'équation (4.14). La fixation est obtenue en maintenant à une valeur nulle les termes constants c_1 et c_2 du développement quadratique du mouvement apparent, qui représentent donc la vitesse 2D du centre de l'image.

En nous inspirant de l'article [Questa 95], nous proposons d'atteindre cet objectif en utilisant la connaissance a priori que nous avons de ces termes. D'après l'équation (1.6), nous savons que $(c_1, c_2)^T = (-v_x + \Omega_y, -v_y - \Omega_x)^T$. Donc :

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \iff \begin{cases} v_x = & -\Omega_y \\ v_y = & \Omega_x \end{cases}$$

Dans la loi de commande (4.14), les translations T_x et T_y correspondent à des degrés de liberté non-contraints. L'influence de la rotation contrôlée pour l'alignement peut donc être compensée en commandant la translation selon la loi de commande :

$$\begin{pmatrix} T_x \\ T_y \end{pmatrix} = \widehat{Z_p} \begin{pmatrix} -\Omega_y \\ \Omega_x \end{pmatrix} \tag{4.16}$$

où $\widehat{Z_p}$ est l'estimation de Z_p déterminée par l'estimation $\widehat{v_z}$ de v_z (cf. 4.12) et la mesure de T_Z. Les valeurs Ω_x et Ω_y sont celles fournies par la loi de commande de la tâche d'alignement.

Le défaut de cette loi de commande est qu'elle fonctionne en boucle ouverte. A aucun moment nous n'utilisons une information mesurée ou estimée des paramètres c_1 et c_2 que nous cherchons à annuler. Donc toute erreur dans l'estimation de la profondeur Z_p générera un comportement différent de celui qui est attendu, sans avoir la possibilité de corriger cette déviation.

Commande en asservissement visuel dynamique

Dans cette seconde approche, nous proposons comme précédemment de contrôler la vitesse apparente du centre de l'image afin qu'elle reste nulle, ce qui assure a priori que le point qui se projette initialement dans cette position restera le même au cours de l'asservissement. Cependant, à la différence de ce qui est présenté ci-dessus, les termes constants du mouvement apparent sont pris en compte explicitement dans la boucle de commande, et ce afin d'éviter les erreurs de dérive dues à une estimation erronée de la profondeur. Nous utilisons pour cela un schéma d'asservissement visuel dynamique en considérant le vecteur de tâche :

$$s = (c_1, c_2, q_1 - \Omega_y, q_2 + \Omega_x)^T$$

dont la valeur désirée est :

$$s^* = (0, 0, 0, 0)^T$$

La relation d'interaction associée à ce vecteur, déterminée à partir des équations (3.2), s'écrit :

$$\dot{s} = \begin{pmatrix} \gamma_1 v_x^2 + \gamma_1 v_x \Omega_y + \gamma_2 v_x v_y - \gamma_2 v_x \Omega_x - v_x v_z + \dfrac{\dot{T}_x}{Z_p} + \dot{\Omega}_y \\ \gamma_1 v_x v_y + \gamma_1 v_y \Omega_y + \gamma_2 v_y^2 - \gamma_2 v_y \Omega_x - v_y v_z + \dfrac{\dot{T}_y}{Z_p} - \dot{\Omega}_x \\ \gamma_1^2 v_x v_z + \gamma_1 \gamma_2 v_y v_z - \gamma_1 v_z^2 - \gamma_2 v_z \Omega_z - v_z \Omega_y + \dfrac{\gamma_1 \dot{T}_z}{Z_p} \\ \gamma_1 \gamma_2 v_x v_z + \gamma_2^2 v_y v_z - \gamma_2 v_z^2 + \gamma_1 v_z \Omega_z + v_z \Omega_x + \dfrac{\gamma_2 \dot{T}_z}{Z_p} \end{pmatrix}$$

Les mouvements de la caméra sur les axes considérés pour la commande (translations le long des axes orthogonaux à l'axe optique et rotations autour de ces mêmes axes) apparaissent dans la relation précédente sous la forme d'une accélération. Cependant, en choisissant de commander le système en contraignant ces accélérations, la relation d'interaction s'écrit alors :

$$\dot{s} = \begin{bmatrix} 1/Z_p & 0 & 0 & 1 \\ 0 & 1/Z_p & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \dot{T}_x \\ \dot{T}_y \\ \dot{\Omega}_x \\ \dot{\Omega}_y \end{pmatrix} + f(V, r) \tag{4.17}$$

où $f(V, r)$ est une fonction quadratique du torseur cinématique V de la caméra et de la position r de l'objet dans le repère de la caméra.

Nous nous trouvons ici dans le cas, étudié dans le chapitre 3, où la matrice d'interaction qui lie $\dot{s}$ avec le vecteur de commande en accélération n'est pas de rang plein. Plus exactement, cette matrice est ici de rang 2. Il n'est donc pas possible de contrôler le système par une méthode de type gradient, en agissant uniquement sur l'accélération de la caméra. Pour atteindre l'objectif défini, nous devons reformuler l'équation précédente pour faire apparaître une relation d'interaction avec une matrice de rang plein, liant $\dot{s}$ à un vecteur composé de termes d'accélération et de termes de vitesse.

Nous devons donc discrétiser deux des termes d'accélération, comme nous l'avons vu dans le chapitre précédent. A priori plusieurs choix sont possibles, mais par souci d'homogénéité, nous n'étudierons que les deux cas où, la discrétisation s'applique soit à la translation soit à la rotation.

Considérant dans un premier temps que les accélérations $\dot{T}_x$ et $\dot{T}_y$ sont discrétisées, la relation d'interaction met en lien $\dot{s}$ avec le vecteur $(T_x,\ T_y,\ \dot{\Omega}_x,\ \dot{\Omega}_y)^T$. Cependant, dans ce cas, les dérivées des termes s_3 et s_4, qui représentent l'information d'orientation du plan observé, ne dépendent pas de la commande en rotation. Ce paradoxe nous a amené à rejeter cette approche.

Nous considérons maintenant que la discrétisation s'applique à l'accélération angulaire, c'est-à-dire :

$$\begin{pmatrix} \dot{\Omega}_x \\ \dot{\Omega}_y \end{pmatrix} = \frac{1}{\delta t} \begin{pmatrix} \Omega_x - \widehat{\Omega_x} \\ \Omega_y - \widehat{\Omega_y} \end{pmatrix}$$

où $\widehat{\Omega}$ est la mesure de la vitesse de rotation à l'itération précédente.

En considérant une nouvelle fois que la vitesse de translation T_z est constante, la relation d'interaction (4.17) s'écrit alors, comme nous l'avons vu dans l'exemple traité au chapitre précédent dans le cas d'une matrice L de rang non-plein, sous la forme :

$$\dot{s} = M \begin{pmatrix} \dot{T}_x \\ \dot{T}_y \\ \Omega_x \\ \Omega_y \end{pmatrix} + \delta_s + \delta_m$$

avec :

$$M = \begin{bmatrix} 1/Z_p & 0 & -\gamma_2 v_x & \frac{1}{\delta t} + \gamma_1 v_x \\ 0 & 1/Z_p & -\frac{1}{\delta t} - \gamma_2 v_y & \gamma_1 v_y \\ 0 & 0 & 0 & -v_z \\ 0 & 0 & v_z & 0 \end{bmatrix} \qquad \delta_s = (v_z - \gamma_1 v_x - \gamma_2 v_y) s = \alpha s$$

et $$\delta_m = \left(\left(-\frac{1}{\delta t} + v_z - \gamma_1 v_x - \gamma_2 v_y \right) \widehat{\Omega_y},\ \left(\frac{1}{\delta t} - v_z + \gamma_1 v_x + \gamma_2 v_y \right) \widehat{\Omega_x},\ s_4 \Omega_z,\ -s_3 \Omega_z \right)^T$$

La nouvelle matrice d'interaction M est maintenant inversible. Il est donc possible d'établir une loi de commande du type gradient en imposant une décroissance exponentielle de l'erreur de gain λ. En effectuant l'estimation $\widehat{s}$ de s de la même manière que précédemment, et en estimant respectivement v_x et v_y par $-c_1 + \widehat{\Omega_y}$ et $-c_2 - \widehat{\Omega_x}$, cette loi de commande est donnée par :

$$\begin{pmatrix} \dot{T}_x \\ \dot{T}_y \\ \Omega_x \\ \Omega_y \end{pmatrix} = -\widehat{M}^{-1} \left((\lambda + \widehat{\alpha})s + \begin{pmatrix} \left(\frac{1}{\delta t} - \alpha\right)\widehat{\Omega_y} \\ -\left(\frac{1}{\delta t} - \alpha\right)\widehat{\Omega_x} \\ \widehat{s_4}\widehat{\Omega_z} \\ -\widehat{s_3}\widehat{\Omega_z} \end{pmatrix} \right) \tag{4.18}$$

où $\widehat{M}$ représente l'estimation de la matrice M et $\widehat{\alpha}$ est l'estimation de α déjà présentée dans l'équation (4.13). L'estimation de M se fait à partir de celles de v_z et Z_p identiquement au cas de la tâche d'alignement, et celles des termes du type $\gamma_1 v_x$, $\gamma_2 v_y$, etc., par combinaison de l'estimation des paramètres affines et la mesure de Ω_z ou l'estimation de v_z (cf. l'équation (3.16)).

Commande basée sur l'estimation de la position du centre initial de l'image

L'objectif de la tâche de fixation consiste à effectuer un "rejet de perturbation" de la position du centre de l'image. Dans les deux méthodes précédentes, la méthode utilisée consistait à annuler les variations de cette position à chaque itération, c'est-à-dire sa vitesse apparente. La première approche présente le défaut de fonctionner en boucle ouverte et d'être peu robuste aux erreurs d'estimation de la profondeur. Ce problème était résolu avec la seconde approche. Cependant, en contraignant simplement la vitesse du centre de l'image à rester nulle, il existe un risque de dérive due au temps de réaction du système pour ramener cette vitesse apparente à zéro.

Nous proposons donc une troisième méthode qui s'inspire de l'approche présentée au chapitre 2. Le principe est d'estimer, à chaque itération, la position $(x,\ y)^T$ du point se projetant au centre de l'image initiale, en intégrant sa vitesse (cf. [Crétual 98a]). Il s'agit ensuite d'appliquer des techniques développées pour l'asservissement visuel géométrique, afin de ramener cette position au centre de l'image. Ainsi en choisissant pour vecteur de tâche $s = (x, y, q_1 - \Omega_y, q2 + \Omega_x)^T$, de valeur désirée $s^* = (0, 0, 0, 0)^T$, et en nous basant sur les équations (3.1) et (4.11), nous obtenons la relation d'interaction suivante (sous l'hypothèse d'une translation T_z constante) :

$$\dot{s} = L\ (T_x, T_y, \Omega_x, \Omega_y)^T + \delta_s$$

avec :

$$L = \begin{bmatrix} \frac{-1+\gamma_1 x+\gamma_2 y}{Z_p} & 0 & -1-x^2 & -xy \\ 0 & \frac{-1+\gamma_1 x+\gamma_2 y}{Z_p} & -xy & 1+y^2 \\ \frac{\gamma_1^2 v_z}{Z_p} & \frac{\gamma_1\gamma_2 v_z}{Z_p} & 0 & -v_z \\ \frac{\gamma_1\gamma_2 v_z}{Z_p} & \frac{\gamma_1^2 v_z}{Z_p} & v_z & 0 \end{bmatrix}$$

$$\text{et} \quad \delta_s \begin{pmatrix} \Omega_z y + v_z(x - \gamma_1 x^2 - \gamma_2 xy) \\ -\Omega_z x + v_z(y - \gamma_1 xy - \gamma_2 y^2) \\ -\gamma_1 v_z^2 - \gamma_2 v_z \Omega_z \\ -\gamma_2 v_z^2 + \gamma_1 v_z \Omega_z \end{pmatrix}$$

Nous pouvons établir une loi de commande de type gradient, avec une décroissance exponentielle de l'erreur de gain λ, de la forme :

$$(T_x, T_y, \Omega_x, \Omega_y)^T = -\widehat{L}^{-1}(\lambda s - \widehat{\delta_s}) \tag{4.19}$$

En pratique, les termes les termes $\gamma_1 x$, $\gamma_2 y$, etc. sont considérés négligeables devant 1, en particulier au voisinage de la convergence ; il en est de même pour les termes $\gamma_1 x^2$, $\gamma_2 xy$, etc. devant x ou y et pour les termes $\gamma_1\gamma_2 v_z$, $\gamma_2^2 v_z$. La matrice L et le vecteur δ_s résultant de ces diverses approximations, sont alors estimés en utilisant les méthodes décrites précédemment.

Contrôle du temps à collision

Nous ne présentons pas ici les techniques qui permettent de prévenir une éventuelle collision de la caméra avec l'objet d'intérêt, dans le cas de la tâche qui couple l'alignement et la fixation. Elles sont identiques à ce qui était proposé pour le contrôle de l'alignement seul, et on n'apporterait rien à les détailler ici.

4.3.4 Résultats de simulation

Les diverses lois de commande présentées dans le paragraphe précédent ont été expérimentées en simulation. Afin de pouvoir les comparer les unes aux autres, les conditions expérimentales choisies sont identiques dans tous les cas considérés, à savoir :

– le temps de cycle considéré correspond à la cadence vidéo, c'est-à-dire 40 ms ;

- les erreurs angulaires initiales sont de 30 degrés autour de l'axe $\overrightarrow{x}$ et de 35 degrés autour de l'axe $\overrightarrow{y}$;
- le plan est situé à une distance de 1 m de la caméra ;
- la valeur initiale de la translation le long de l'axe optique est de 10 mm/s ;
- la rotation autour de l'axe optique Ω_z est fixée à zéro tout au long de l'asservissement, sans perte de généralité ;
- enfin, le gain λ de la décroissance exponentielle de l'erreur vaut 0,3.

Alignement

Nous présentons ici les résultats concernant la tâche d'alignement, sans prendre en compte la fixation. Cela concerne d'une part le comportement du système lorsqu'il est soumis à la loi de commande (4.14), et également lorsque cette loi est complétée par un contrôle du temps à collision (cf. (4.15)).

Dans chacun des deux cas, les translations le long des axes $\overrightarrow{x}$ et $\overrightarrow{y}$ de la caméra ont été fixées à une valeur nulle tout au long de l'asservissement. Pour le contrôle du temps à collision, la valeur désirée t_c^* a été choisie égale à 80 s, alors que sa valeur initiale est de 100 s (une translation T_z de 10 mm/s pour une profondeur de 1 m). Cela correspond à une valeur désirée v_z^* de 0.0125.

Pour chacune des approches, nous présentons quatre courbes, dont nous donnons les références d'abord pour le cas de l'alignement seul, puis pour le cas de l'alignement avec contrôle du temps à collision. Il s'agit :

- des composantes du vecteur de tâche s, issues des paramètres quadratiques du mouvement, avec s_1 en trait plein et s_2 en trait discontinu (figures (4.14) a et (4.16) a) ;
- des vitesses de rotation (en deg/s) données par la loi de commande, avec Ω_x en trait plein et Ω_y en trait discontinu (figures (4.14) b et (4.17) a) ;
- des erreurs angulaires (en deg), avec $\arctan(\gamma_1)$ en trait plein et $\arctan(\gamma_2)$ en trait discontinu (figures (4.15) a et (4.18) a) ;
- du temps à collision (en s) (figures (4.15) b et (4.18) b).

Dans le cas où le temps à collision est contrôlé, deux courbes supplémentaires sont exposées, à savoir la troisième composante du vecteur de tâche, v_z (figure (4.16) b), et la commande en accélération le long de l'axe optique, $\dot{T}_z$ (figure (4.17) b).

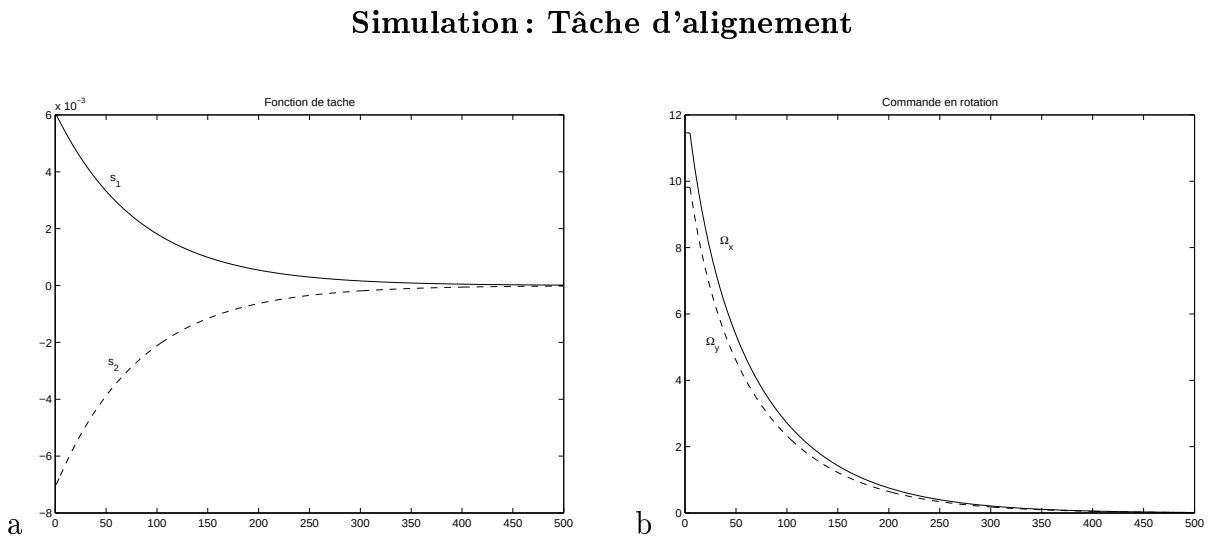


FIG. 4.14 – *Fonction d'erreur e (a). Vitesses angulaires données par la loi de commande (b)*

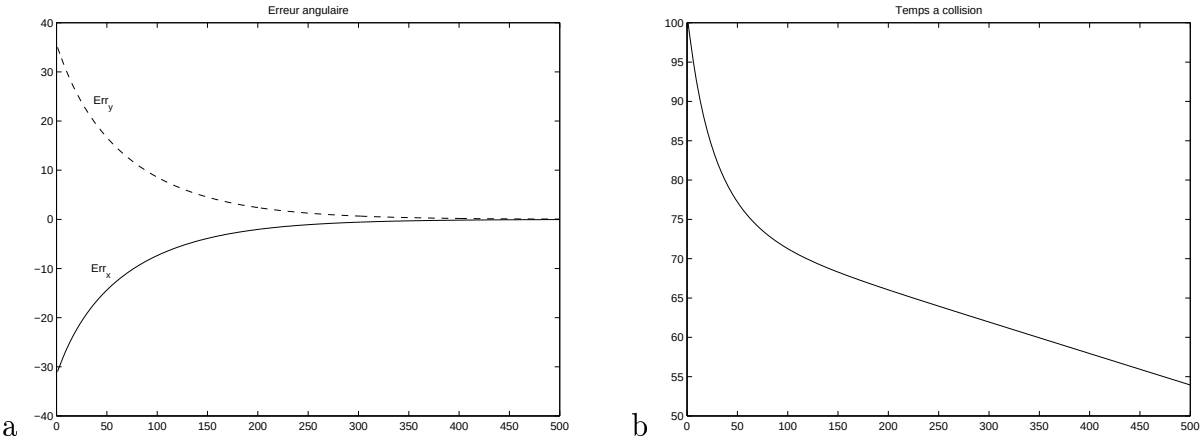


FIG. 4.15 – *Erreur angulaire (a). Temps à collision (b)*

Simulation : Tâche d'alignement avec contrôle du temps à collision

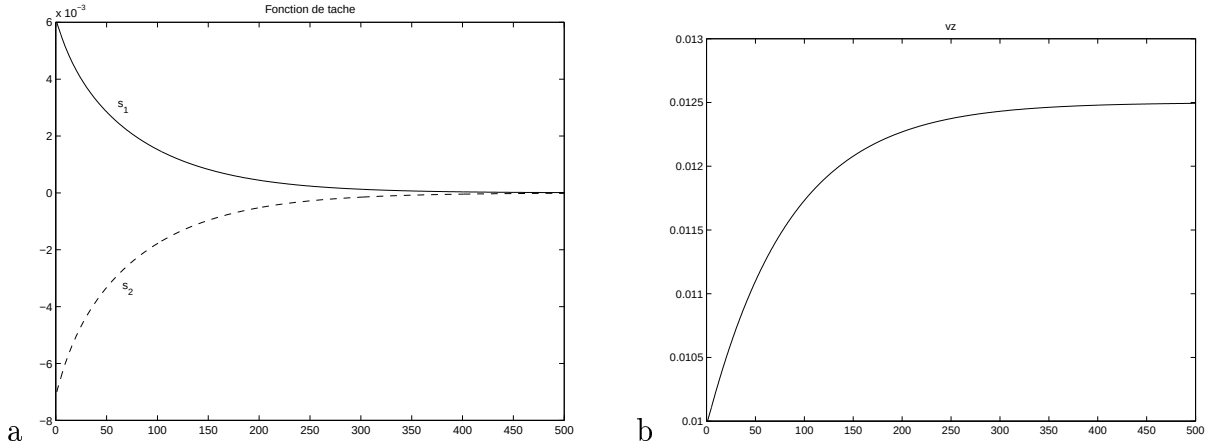


FIG. 4.16 – *Fonction d'erreur : Termes déduits des paramètres quadratiques (a) et v_z (b)*

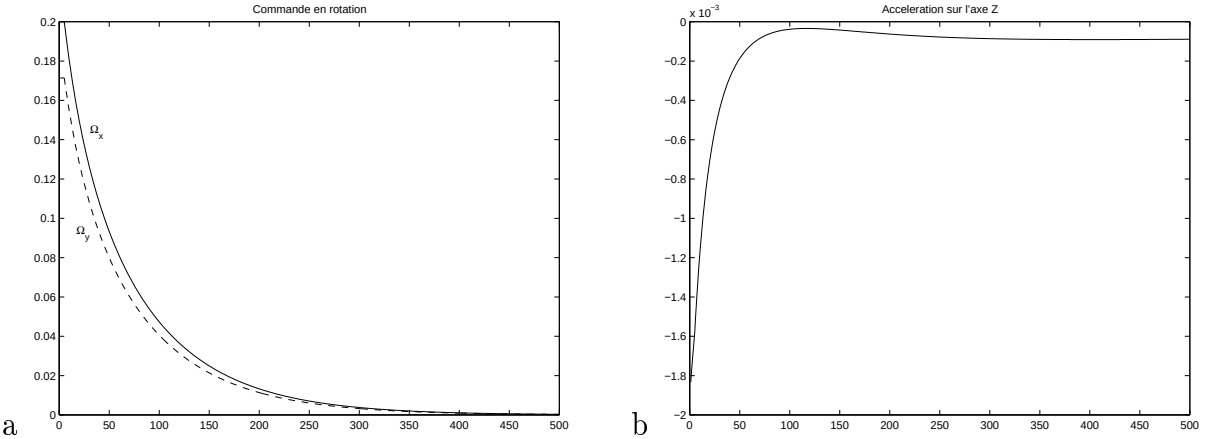


FIG. 4.17 – *Vitesses angulaires Ω_x, Ω_y (a) et accélération $\dot{T}_z$ (b) données par la loi de commande*

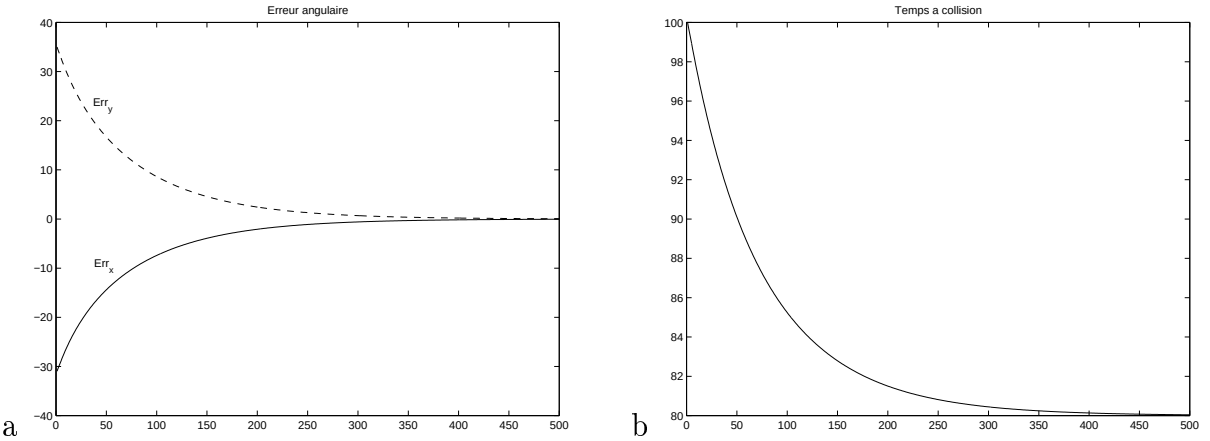


FIG. 4.18 – *Erreur angulaire (a). Temps à collision (b)*

Dans les deux cas, nous obtenons bien une décroissance exponentielle des termes du vecteur de tâche provenant des termes quadratiques du mouvement (s_1 et s_2) et les allures des courbes correspondantes sont très semblables. De même, le comportement du système est très similaire en ce qui concerne la commande en rotation calculée, et les erreurs angulaires, qui tendent bien vers zéro. Dans le premier cas, le temps à collision diminue en permanence, et atteint une décroissance linéaire lorsque la convergence est obtenue pour l'orientation de la caméra. Cela correspond parfaitement à un phénomène de zoom. En revanche, lorsque ce temps à collision est contrôlé, il converge bien vers la valeur souhaitée, ainsi, donc, que le terme v_z qui en est l'inverse. L'accélération le long de l'axe optique atteint une valeur palier négative, qui assure donc une décroissance linéaire de la translation proportionnelle à la diminution de la profondeur.

Alignement et fixation

Dans ce paragraphe, nous présentons les résultats de simulation concernant les trois approches considérées pour résoudre le problème conjoint de l'alignement et de la fixation. Cela concerne tout d'abord la loi de commande (4.16), où la fixation est contrôlée en boucle ouverte. Il s'agit ensuite de la loi de commande (4.18) qui permet de contrôler le système selon un schéma d'asservissement visuel dynamique. Enfin, nous présentons les résultats correspondant à la loi de commande (4.19), grâce à laquelle la position estimée du centre de l'image initiale est contrôlée par un asservissement visuel géométrique.

Dans chacun des cas, la profondeur est estimée à partir de la mesure de T_z et de l'estimation de v_z proposée dans l'équation (4.12). On considère les mesures odométriques comme exacte. Par conséquent, dans le cas où les mesures visuelles dynamiques ne sont pas bruitées, l'estimation de la profondeur ne l'est pas non plus.

Pour chacune des méthodes considérées, six courbes sont présentées. Les références données ci-dessous concernent, dans l'ordre, l'approche où la fixation est obtenue : par une commande en boucle ouverte, puis par une commande basée entièrement sur un schéma d'asservissement visuel dynamique, et enfin par le contrôle de la position estimée du centre de l'image initiale. Il s'agit :

- des composantes du vecteur de tâche s, issues des paramètres quadratiques du mouvement, avec s_1 en trait plein et s_2 en trait discontinu (figures (4.19) a, (4.22) a et (4.25) a) ;
- des termes constants du modèle de mouvement, avec c_1 en trait plein et c_2 en trait discontinu (figures (4.19) b, (4.22) b et (4.27) a) ;
- des vitesses de rotation (en deg/s) données par la loi de commande, avec Ω_x en trait plein et Ω_y en trait discontinu (figures (4.20) a, (4.23) a et (4.26) a) ;
- des commandes en translation le long de l'axe $\overrightarrow{x}$ en trait plein et de l'axe $\overrightarrow{y}$ en

trait discontinu - il s'agit de vitesses (en m/s) pour la commande en boucle ouverte et le contrôle de la position estimée du centre initial, et d'accélérations (en m/s^2) pour la commande par asservissement visuel dynamique - (figures (4.20) b, (4.23) b et (4.26) b) ;

- des erreurs angulaires (en deg), avec $\arctan(\gamma_1)$ en trait plein et $\arctan(\gamma_2)$ en trait discontinu (figures (4.21) a, (4.24) a et (4.27) a) ;
- de la position estimée (en m) du point se projetant au centre de la première image (figures (4.21) b, (4.24) b et (4.25) b).

En ce qui concerne les termes permettant de contrôler l'orientation ($-\gamma_1 v_z$ et $-\gamma_2 v_z$), les comportements sont similaires dans les trois cas, et aboutissent à la convergence, par une décroissance exponentielle. Les erreurs angulaires, qui dépendent fortement de ces valeurs, suivent donc également des trajectoires semblables et tendent vers zéro. Dans le premier et le troisième cas, les commandes en vitesse angulaire et en vitesse de translation sont, là encore, similaires. Dans le cas où l'on utilise un schéma d'asservissement visuel dynamique, la commande en accélération translationnelle est très brusque au départ, et revient rapidement à une valeur quasi nulle. Cela permet bien de contraindre les termes constants à atteindre zéro. Cependant, cela génère également une sollicitation très forte au départ de la vitesse angulaire (environ 3 fois ce qui est obtenu dans les deux autres cas). Cette commande en rotation revient ensuite (après 20 à 30 itérations), à un comportement proche de celui des deux autres approches.

Pour ce qui est des termes constants du mouvement, si dans les trois cas, ils convergent bien vers zéro, les comportements sont par contre différents les uns des autres. Dans la première approche, ils restent très faibles, et leur différence avec zéro est en réalité simplement due à la discrétisation du système (la profondeur varie pendant chaque itération et l'égalité obtenue au début de l'itération n'est plus exacte à la fin). La décroissance s'explique par le fait que les valeurs des rotations diminuent au cours du temps. Dans le cas de la boucle fermée sur ces paramètres, la perturbation générée par la rotation à laquelle est soumis le système à la première itération, est compensée rapidement, et il apparaît un léger dépassement, dû à la rapidité de réponse du système. Enfin, dans le troisième cas, cette perturbation initiale est compensée lentement, et s'il apparaît un dépassement, ce n'est pas comme précédemment à cause d'une commande en translation trop forte, mais pour compenser le déplacement obtenu pendant les 70 premières itérations.

Enfin, les résultats concernant la position du centre initial sont également très différents. Dans le premier cas, les composantes de ce vecteur restent très faibles (moins de 0.3 mm après 500 itérations). Cependant, nous assistons à une déviation qui parvient à un régime linéaire à la fin de l'asservissement. Dans le second cas, le déplacement est très important au départ, puis diminue légèrement, ce qui correspond au dépassement dans le contrôle des termes constants. A la fin de l'asservissement, nous retrouvons le même phénomène de déviation linéaire que précédemment. La cause est la même dans les deux

Simulation : Tâche d'alignement et fixation en boucle ouverte

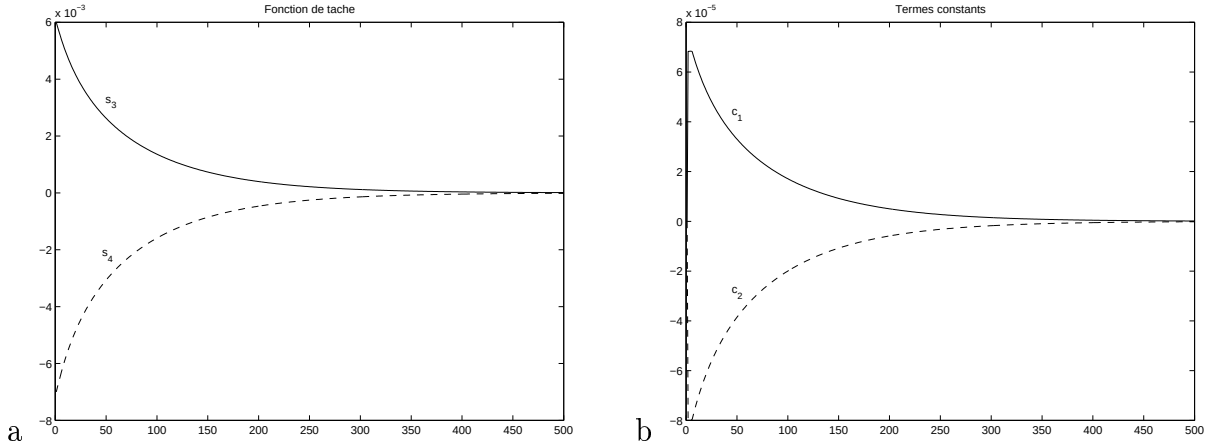


FIG. 4.19 – *Fonction d'erreur : Termes déduits des paramètres quadratiques (a). Paramètres constants du mouvement (b)*

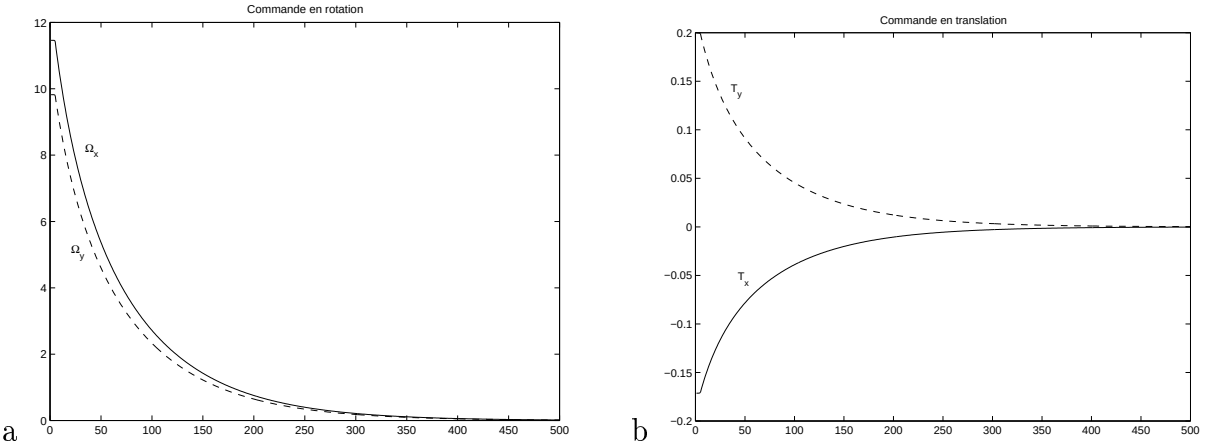


FIG. 4.20 – *Vitesses angulaires Ω_x, Ω_y (a) et de translation T_x, T_y (b) données par la loi de commande*

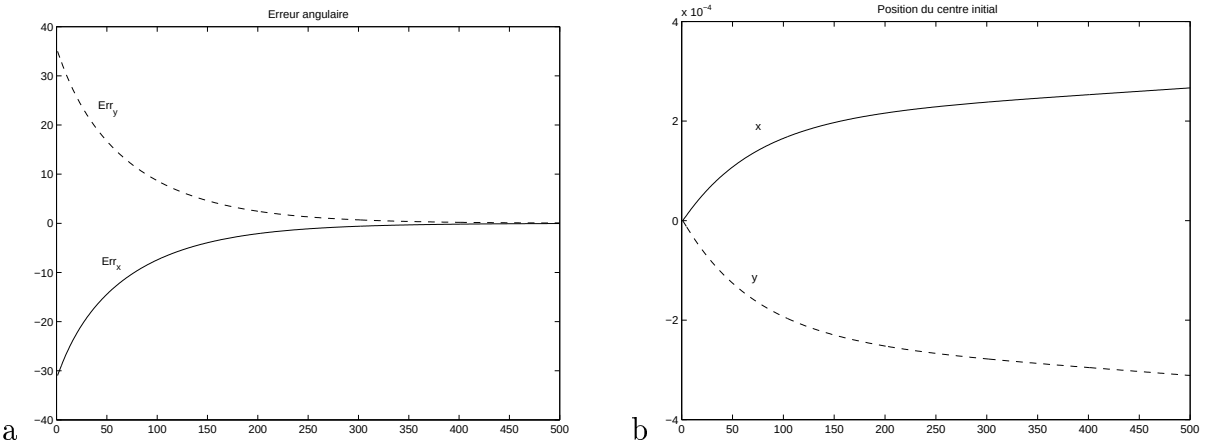


FIG. 4.21 – *Erreur angulaire (a). Position du centre initial (b)*

Simulation : Tâche d'alignement et fixation en boucle fermée

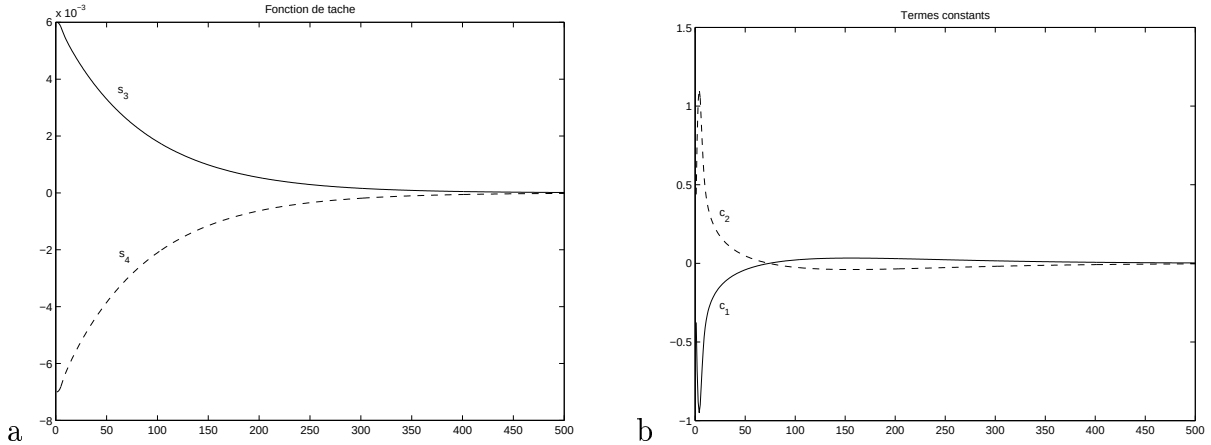


FIG. 4.22 – *Fonction d'erreur : Termes déduits des paramètres quadratiques (a). Paramètres constants du mouvement (b)*

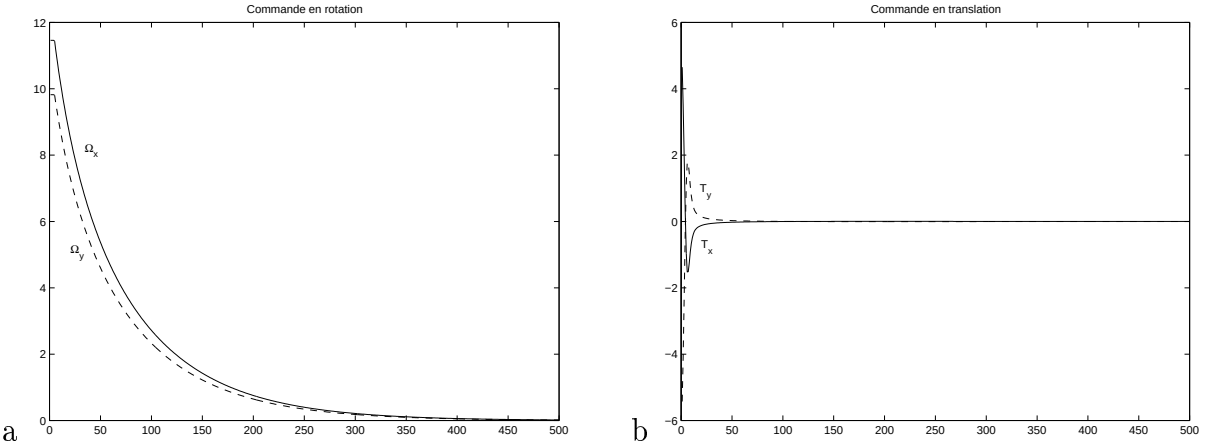


FIG. 4.23 – *Vitesses angulaires Ω_x, Ω_y (a) et accélérations $\dot{T}_x, \dot{T}_y$ (b) données par la loi de commande*

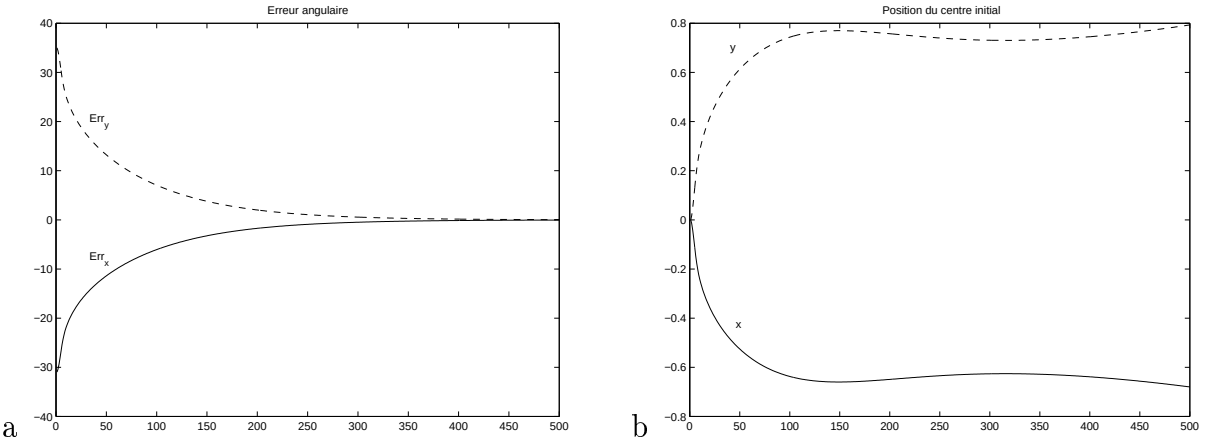


FIG. 4.24 – *Erreur angulaire (a). Position du centre initial (b)*

Simulation : Tâche d'alignement et fixation par contrôle de la position estimée du centre initial

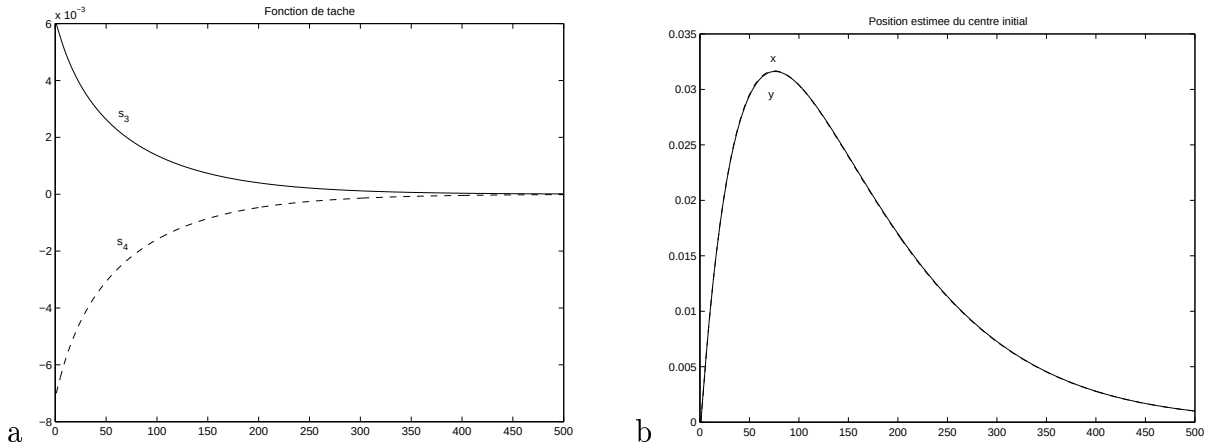


FIG. 4.25 – *Fonction d'erreur : Termes déduits des paramètres quadratiques (a). Position estimée du centre initial (b)*

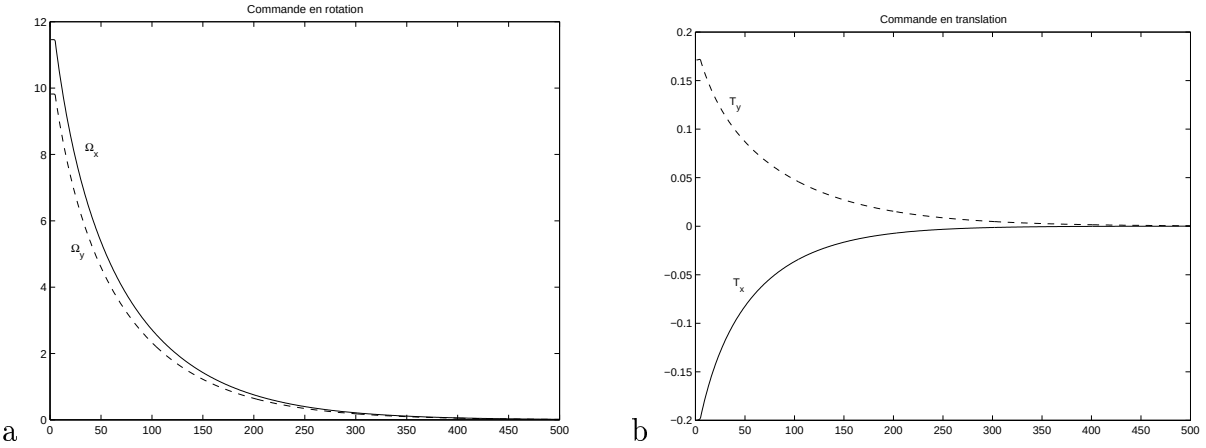


FIG. 4.26 – *Vitesses angulaires Ω_x, Ω_y (a) et de translation T_x, T_y (b) données par la loi de commande*

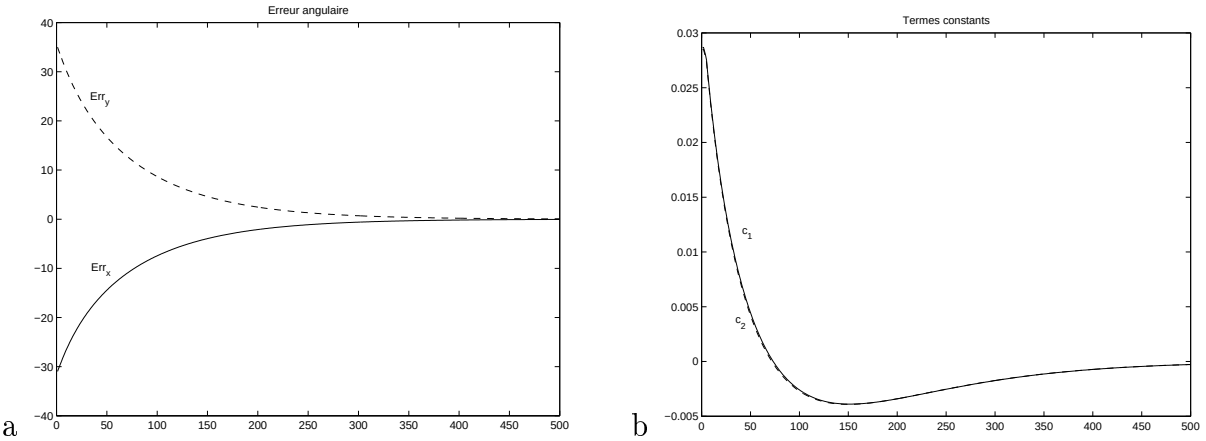


FIG. 4.27 – *Erreur angulaire (a). Paramètres constants du mouvement (b)*

cas. En fait, une fois la convergence atteinte pour l'orientation de la caméra, il n'y a plus de sollicitation en rotation, donc pas non plus en translation, puisqu'elle est sensée compenser les effets de la rotation. Le centre initial, qui s'était déplacé dans l'image pour diverses raisons (caractère discret du système, temps de réaction pour contrôler les termes constants du mouvement) est alors soumis à un mouvement de divergence pure résultant de la translation le long de l'axe optique. Par conséquent, il s'éloigne du centre de l'image de façon linéaire, proportionnellement à la vitesse T_z. Au contraire, dans le troisième cas, où cette position est explicitement contrôlée, elle est progressivement ramenée au centre de l'image, après s'en être éloignée à cause du temps de réaction nécessaire à la commande.

Les première et troisième méthodes sensées assurer la fixation, même si elles aboutissent à des résultats différents en ce qui concerne la position du point initialement au centre de l'image, semblent malgré tout se valoir. En effet, une fois la convergence atteinte pour l'orientation de la caméra, et donc, une fois parvenue à l'objectif principal, la première méthode aboutit à un déplacement plus faible, même si l'on se trouve alors dans une phase de divergence.

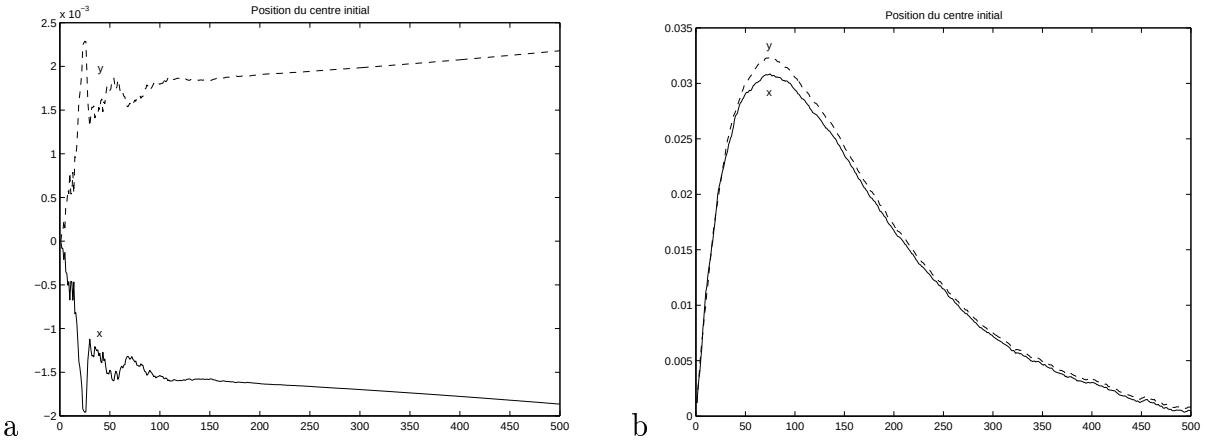
Pour être à même de les comparer au mieux, nous les avons testées en introduisant un bruit de mesure sur les paramètres du mouvement. Il s'agit dans tous les cas de bruits gaussiens centrés, d'écarts-type $3.10-6$ sur les termes constants, 6.10^{-6} sur les termes affines et 3.10^{-5} sur les termes quadratiques. L'estimation de la profondeur étant effectuée à partir de ces mesures, elle est donc également bruitée. Nous donnons sur la figure (4.28) les positions (en m) du centre initial correspondant à ces simulations, pour la commande en boucle ouverte (a) et celle inspirée des travaux du chapitre 2 (b).

Nous pouvons constater que malgré le bruit, le comportement de la troisième méthode est presque inchangé, et la position atteinte à la convergence est la même que dans le cas non-bruité. En revanche, malgré la faible amplitude du bruit, le déplacement est beaucoup plus important pour la boucle ouverte que précédemment. Cela est dû aux erreurs sur l'estimation de la profondeur, dont les effets ne sont pas corrigés par un retour d'information visuelle. Au final, le déplacement est alors plus important que lorsque l'on contrôle la position estimée du point d'intérêt.

4.3.5 Résultats sur site robotique

Les diverses approches pour résoudre le problème que nous nous étions fixé ont été implémentées sur la cellule robotique. Nous présentons dans ce paragraphe, quelques-uns des résultats obtenus. Une image de la scène à la position initiale est présentée sur la figure (4.29) a, et une image de la scène à la convergence sur la figure (4.29) b. On constate que l'objet planaire n'occupe pas la totalité de l'image au départ. De même, des objets non-planaires sont disposés sur le plan d'intérêt. Cependant, malgré ces deux types de perturbations, grâce à l'aspect robuste de l'algorithme RMRm qui rejette, lors

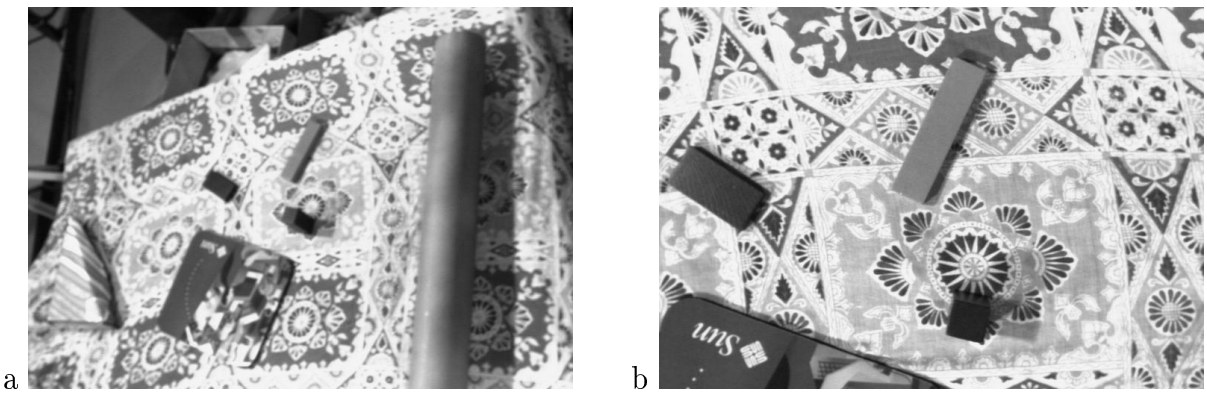
Simulation : Comparaison des résultats des commandes de fixation en boucle ouverte et par contrôle de la position estimée du centre de l'image initiale



a b

FIG. 4.28 – *Position vraie du centre initial en présence de bruit de mesure : fixation en boucle ouverte (a) et fixation par contrôle de l'estimation de cette position (b)*

de l'estimation, l'influence des pixels ayant un mouvement apparent non-conforme, nous verrons que la tâche peut être correctement réalisée.



a b

FIG. 4.29 – *Configuration de la scène : à la position initiale (a) et à la convergence (b)*

Contexte

Les différentes lois de commande présentées auparavant fonctionnent d'autant mieux que le temps de cycle est court. Cependant, l'algorithme d'estimation des paramètres du mouvement est lui d'autant plus précis que les images considérées sont grandes et que le nombre de niveaux de résolution est important. Par conséquent, plus la précision

demandée est fine, plus le temps de calcul est important. Afin d'obtenir un compromis entre ces deux contraintes temporelles opposées, nous avons choisi de travailler avec des images de taille 256 × 256 pixels, et de considérer quatre niveaux de résolution. Cela permet d'obtenir une fréquence d'environ 1 Hz.

A une telle cadence, il peut apparaître des problèmes d'oscillation si le gain de la loi de commande est trop élevé, et que le système est proche de la convergence. Cependant, lorsque l'on se situe loin de la convergence, le gain peut être a priori plus important. Nous avons donc choisi de ne pas utiliser un gain fixe mais dépendant du temps, afin d'accélérer la convergence tout en évitant les oscillations finales, avec :

$$\lambda = \lambda_{min} + \frac{\lambda_{max}}{n}$$

où λ_{min} est la valeur de λ assurant la stabilité à l'approche de la convergence, λ_{max} une valeur plus élevée et n le nombre d'itérations.

Une autre idée a priori possible pour régler le gain serait d'utiliser un gain fonction de l'erreur. Cette solution offre l'avantage d'être plus "souple", dans le sens où l'évolution de la valeur du gain s'adapte à chaque situation, en particulier elle dépend de la position initiale. Nous l'avons cependant rejetée, car le bruit sur l'erreur est ici trop important et le contrôle du gain s'en trouverait très perturbé.

Influence des paramètres de calibration

A partir de deux images successives, l'algorithme RMRm fournit un modèle de déplacement dans l'image, exprimé en pixels. Pour obtenir le modèle de mouvement en mètre, il faut utiliser les paramètres intrinsèques de la caméra, c'est-à-dire : la position (x_c, y_c) de l'intersection de l'axe optique et du plan de projection des images, les rapports mètre-pixel sur chaque axe (p_x sur l'axe $\overrightarrow{x}$ et p_y sur l'axe $\overrightarrow{y}$) et la distorsion radiale (K_d). L'influence de ces paramètres sur cette transformation est détaillée dans l'annexe B.

Pour la tâche précédente (poursuite de cible), une calibration grossière - à savoir (x_c, y_c) au centre de l'image, les rapports mètre-pixel fournit par le constructeur, et la non-prise en compte de la distorsion radiale - permettait d'obtenir des résultats corrects. Cependant, ces paramètres, et en particulier la distorsion radiale, ont une influence forte (cf. annexe B) sur l'estimation des paramètres quadratiques du mouvement apparent. Nous avons donc utilisé, pour cette tâche d'alignement, une calibration relativement fine de la caméra.

Alignement

Nous présentons ici les résultats concernant la tâche d'alignement seul. Nous ne donnerons pas de résultats correspondant au couplage de cette tâche avec un contrôle du temps à collision. En effet, la nécessité d'effectuer une commande en accélération pour la translation le long de l'axe optique produit un comportement très instable du système à

la cadence de 1 Hz. Ces résultats en situation réelle ne sont donc pas encore concluants. Ils pourront très vraisemblablement l'être en utilisant une plus grande puissance de calcul permettant d'augmenter très sensiblement la fréquence de contrôle.

La tâche d'alignement a été testée dans le contexte suivant : $\lambda_{min} = 0.04$, $\lambda_{max} = 0.1$ et $T_z = 10$ mm/s ; les erreurs angulaires initiales étaient d'environ 12 degrés autour de l'axe $\overrightarrow{x}$ et 15 degrés autour de l'axe $\overrightarrow{y}$; les vitesses des axes non-contraints par la commande, c'est-à-dire T_x, T_y et Ω_z, ont été fixées à zéro.

Les erreurs angulaires initiales sont relativement faibles pour deux raisons :

- des erreurs plus importantes conduisent à la sortie de l'objet du champ de vision de la caméra ;
- lorsque le mouvement apparent entre deux images est trop important - ce qui est le cas avec des erreurs initiales plus grandes - l'estimation des paramètres quadratiques est extrêmement bruitée et peut devenir inexploitable.

Nous verrons dans le paragraphe suivant, que l'ajout d'une tâche de fixation permet de résoudre ces deux problèmes. Cette amélioration était attendue pour le premier de ces problèmes, et c'est d'ailleurs la principale motivation qui nous a amené à concevoir la tâche de fixation. En ce qui concerne le second, il s'est avéré, au cours de nos expérimentations, que le fait de maintenir un même point au centre de l'image, c'est-à-dire d'avoir des paramètres constants du mouvement faibles, permet d'obtenir une estimation plus robuste des paramètres affines et quadratiques.

Les différentes courbes présentées sont respectivement :

- les composantes du vecteur de tâche s, sur la figure (4.30) a, avec s_1 en trait plein et s_2 en trait discontinu ;
- les vitesses de rotation (en deg/s) données par la loi de commande sur la figure (4.30) b, avec Ω_x en trait plein et Ω_y en trait discontinu ;
- les paramètres constants du mouvement sur la figure (4.31) a, avec c_1 en trait plein et c_2 en trait discontinu ;
- enfin, les erreurs angulaires (en deg) sur la figure (4.31) b, avec $\arctan(\gamma_1)$ en trait plein et $\arctan(\gamma_2)$ en trait discontinu.

Remarque : les erreurs angulaires sont calculées à partir d'une connaissance a priori de l'orientation du plan, et d'une mesure par odométrie de la position angulaire de la caméra dans son repère cartésien d'évolution. Ces informations ne sont bien évidemment pas utilisées dans la boucle de commande.

Expérimentation sur site robotique : Tâche d'alignement

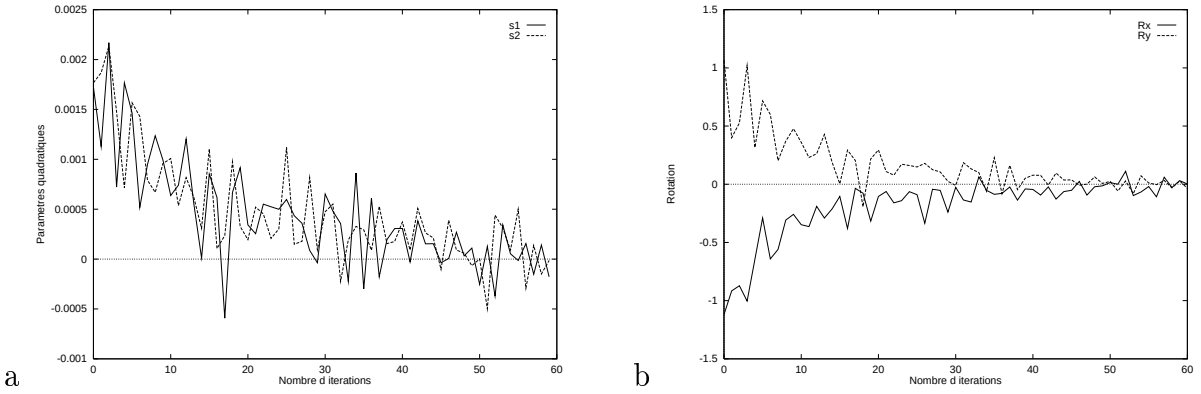


FIG. 4.30 – *Fonction de tâche (a). Vitesses angulaires données par la loi de commande*

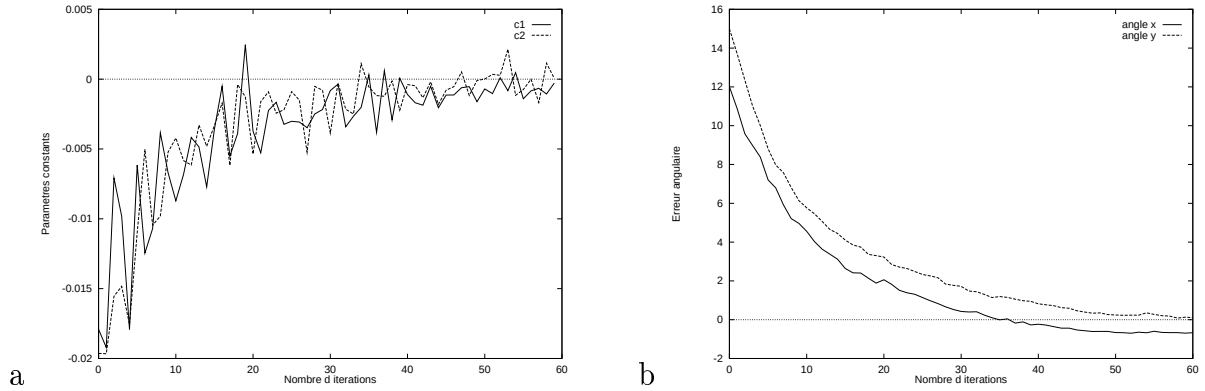


FIG. 4.31 – *Paramètres constants du mouvement (a). Erreurs angulaires (b)*

La première remarque que nous pouvons faire au vu de ces résultats, est que l'estimation des paramètres quadratiques du mouvement est très bruitée. Cependant, le comportement du système est globalement similaire à ce qui pouvait être attendu après la simulation ; la fonction de tâche et la commande en rotation convergent bien vers zéro. Les erreurs angulaires convergent également, et les écarts avec une valeur nulle sont simplement dus à l'imprécision sur l'orientation effective du plan observé. Enfin, les paramètres constants décroissent vers une valeur nulle, ce qui est normal, vu qu'ils ne dépendent en théorie que des rotations Ω_x et Ω_y (les translations T_x et T_y sont nulles), mais ils conservent un signe constant et leur décroissance est lente. Par conséquent, le déplacement du centre initial de l'image est élevé, puisqu'il atteint une valeur d'une centaine de pixels dans chaque direction.

Alignement et fixation

Les tâches couplant l'alignement et la fixation, utilisant des commandes en vitesse uniquement, c'est-à-dire avec d'une part, une fixation en boucle ouverte, et d'autre part, un contrôle de la position du point au centre de la première image, ont été implémentées sur la cellule robotique. En effet, en ce qui concerne la commande entièrement basée sur un schéma d'asservissement visuel dynamique, les sollicitations en accélération translationnelle, font apparaître un phénomène similaire à celui décrit lors du contrôle du temps à collision. En effet, une nouvelle fois, la fréquence d'exécution de l'asservissement rend très délicate la stabilité du système. Nous ne présenterons donc pas de résultats dans ce cas.

Nous avons observé que le comportement du système soumis à une commande en boucle ouverte pour assurer la fixation paraissait similaire à celui obtenu en contrôlant la position estimée du centre de l'image initiale, au moins en ce qui concerne l'alignement. Nous montrons l'ensemble des courbes significatives de l'asservissement dans le second cas, et, pour le premier cas, uniquement les courbes permettant de comparer l'efficacité de la tâche de fixation.

Pour la commande en boucle ouverte, par rapport au cas de l'alignement seul, les valeurs de λ_{min} et T_z sont restées inchangées. Par contre, l'estimation des paramètres quadratiques étant rendue plus robuste par la fixation, la valeur de λ_{max} à été augmentée jusqu'à une valeur de 0.2.

Dans le second cas, il est possible de fixer des valeurs de gain de décroissance différentes pour les commandes en rotation et en translation, et ce, à cause du bruit de mesure beaucoup plus faible sur le déplacement du centre de la première image (qui dépend de l'intégration du modèle affine) que sur les paramètres quadratiques du mouvement. Il s'agit donc de ne plus spécifier le gain par scalaire λ, mais par une matrice diagonale générée par le vecteur $(\lambda_{tra}, \lambda_{tra}, \lambda_{rot}, \lambda_{rot})^T$. La valeur de λ_{rot}, qui détermine la valeur de la vitesse de rotation, est la même que celle du scalaire λ considéré dans le cas où la fixation est obtenue par une commande en boucle ouverte. La précision dans l'estimation du déplacement du centre initial permet de choisir une valeur de λ_{tra}, correspondant à la commande en translation, plus élevée. Ainsi, sur le même schéma de gain adaptatif que précédemment, nous avons fixé $\lambda_{tra} = 0.3 + 0.06/n$.

La fixation nous permet également de considérer des erreurs angulaires initiales plus importantes que lorsque le système est simplement soumis à la contrainte d'alignement. Dans les deux expériences effectuées, elle étaient de 30 degrés autour de l'axe $\overrightarrow{x}$ et 25 autour de l'axe $\overrightarrow{y}$. Enfin, comme précédemment, la rotation autour de l'axe optique Ω_z est restée nulle tout au long de l'asservissement.

Les différentes courbes présentées sont respectivement :

– les composantes du vecteur de tâche s, issues des paramètres quadratiques du mou-

vement, sur la figure (4.32) a, avec s_3 en trait plein et s_4 en trait discontinu ;

- les composantes du vecteur de tâche correspondant à la position estimée du point se projetant au centre de l'image initiale sur la figure (4.32) b, avec x en trait plein et y en trait discontinu ;
- les vitesses de rotation (en deg/s) données par la loi de commande sur la figure (4.33) a, avec Ω_x en trait plein et Ω_y en trait discontinu ;
- les vitesses de translation (en mm/s) données par la loi de commande sur la figure (4.33) b, avec T_x en trait plein et T_y en trait discontinu ;
- les paramètres constants du mouvement sur la figure (4.34) a, avec c_1 en trait plein et c_2 en trait discontinu ;
- les erreurs angulaires (en deg) sur la figure (4.34) b, avec $\arctan(\gamma_1)$ en trait plein et $\arctan(\gamma_2)$ en trait discontinu.

Nous retrouvons, ici encore, malgré le bruit de mesure important, un comportement de même allure que ce que nous obtenions lors des essais en simulation. On observe la convergence des termes s_1 et s_2, ainsi que la décroissance des vitesses de rotation. Il y a également un dépassement des termes constants de façon à garantir la convergence de la position estimée du centre initial vers zéro. Enfin, les erreurs angulaires sont ramenées vers les mêmes valeurs (environ -1 degré autour de l'axe $\overrightarrow{x}$ et 0 autour de l'axe $\overrightarrow{y}$) que lorsque seul l'alignement est contrôlé, ce qui montre la répétabilité de nos différents algorithmes.

De plus, pour pouvoir comparer les performances des deux approches retenues pour leur capacité à réaliser la fixation, nous montrons deux autres courbes sur la figure (4.35), concernant la loi de commande en boucle ouverte. Il s'agit de la position (en m) du point se projetant au centre de la première image (a), et des paramètres constants du mouvement (b, en pixel/s).

Nous pouvons noter que, malgré le dépassement des termes constants lorsque le système répond à la première perturbation, le déplacement du point d'intérêt arrive très rapidement en phase de divergence, comme nous pouvions le prévoir avec les résultats en simulation dans le cas de mesures bruitées. Ce phénomène est principalement dû à l'erreur d'estimation de la profondeur lors des premières itérations, qui génère le déplacement de l'ordre de 20 pixels que l'on peut noter dès la troisième itération. Ce déplacement ne peut pas être compensé, puisque cette donnée n'est pas utilisée dans la loi de commande. En définitive, le déplacement du point après 60 itérations est de l'ordre de 40 à 50 pixels dans chaque direction. Nous pouvons facilement comparer ces valeurs au déplacement maximal de 15 pixels observé dans le cas précédent, et de l'ordre de 1 pixel à la convergence, pour conclure sur l'inefficacité de la fixation par une commande en boucle ouverte.

Expérimentation sur site robotique : Tâche d'alignement et fixation par contrôle de la position estimée du centre de la première image

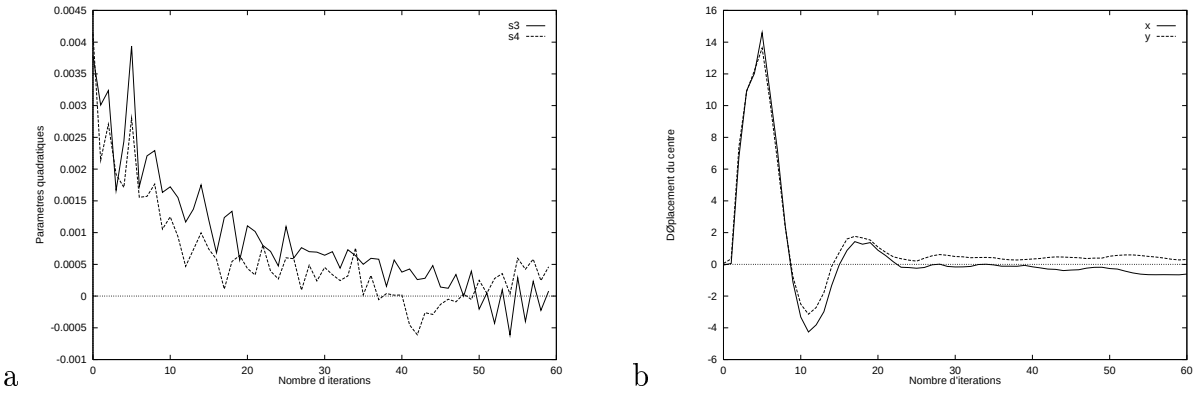


FIG. 4.32 – *Fonction de tâche : Termes issus des paramètres quadratiques du mouvement (a) et position estimée du centre de l'image initiale (b)*

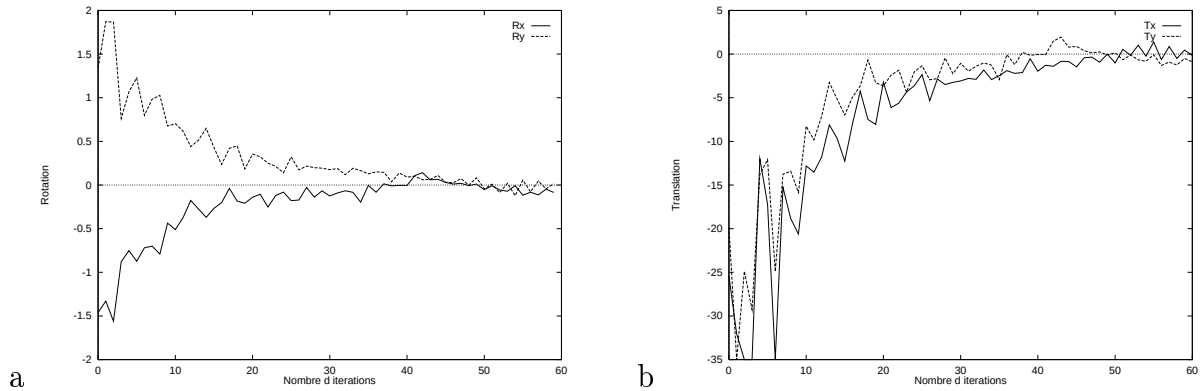


FIG. 4.33 – *Vitesses angulaires (a) et de translation (b) données par la loi de commande*

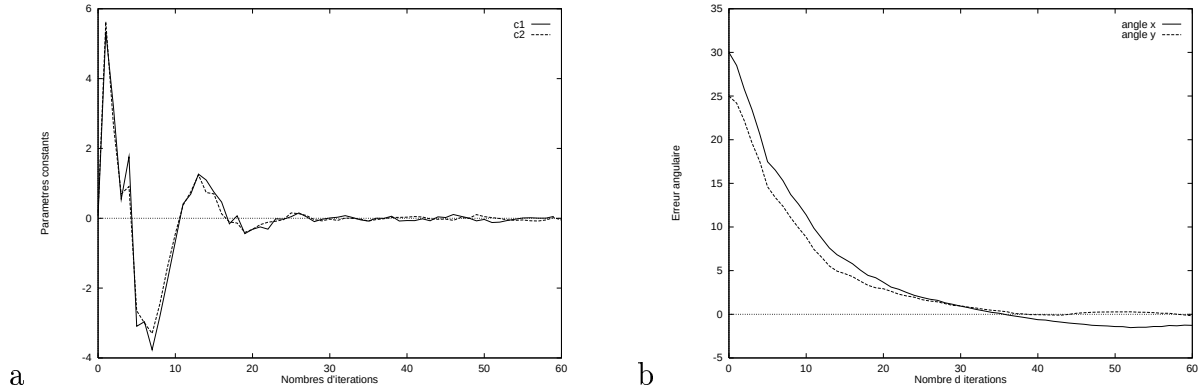


FIG. 4.34 – *Paramètres constants du mouvement (a). Erreurs angulaires (b)*

Expérimentation sur site robotique : Tâche d'alignement et fixation en boucle ouverte

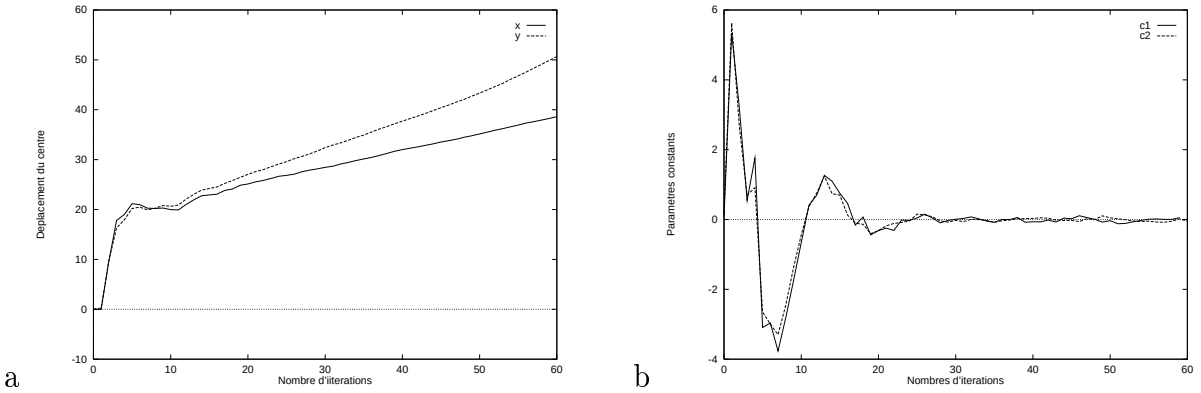


FIG. 4.35 – *Position estimée du centre de l'image initiale (a). Paramètres constants du mouvement (b)*

4.4 Suivi d'une trajectoire d'observation d'une surface plane

4.4.1 Présentation de la tâche

Nous supposons, pour cette tâche, que la caméra observe une surface tridimensionnelle. L'objectif fixé pour la caméra est de suivre une trajectoire parallèlement à cette surface. Cette trajectoire est contrôlée de telle manière que la séquence acquise réponde à des contraintes imposées, par exemple, par un algorithme de traitement d'image en aval de l'asservissement visuel et travaillant en parallèle à celui-ci. Pour plus de simplicité, nous considérerons dans nos expérimentations que cette surface est plane, même si les loi de commande que nous établirons restent valables pour des cas plus généraux.

Cette tâche se place dans le cas intermédiaire évoqué dans le chapitre 3, où la matrice d'interaction entre le vecteur de mesures et l'accélération est généralement de rang plein, mais peut ne plus l'être lorsque le système se trouve dans une configuration spécifique, en particulier, au voisinage de la convergence.

Considérons, toujours à titre d'exemple, la configuration présentée sur la figure 4.36. Nous supposons qu'un algorithme doit traiter successivement chacune des zones notées de 1 à 4, et que sa cadence d'exécution est de 1 Hz. La contrainte imposée à la trajectoire de la caméra consiste, dans un premier temps à suivre la bande dans laquelle se situent les diverses zones d'intérêt. Cela signifie que le mouvement de la caméra doit s'effectuer dans un plan parallèle à la surface d'intérêt, et dans une direction bien définie. De plus, cette trajectoire doit être parcourue à une vitesse telle qu'une seconde après avoir observé

la zone 1, la caméra observe exactement la zone 2, et ainsi de suite.

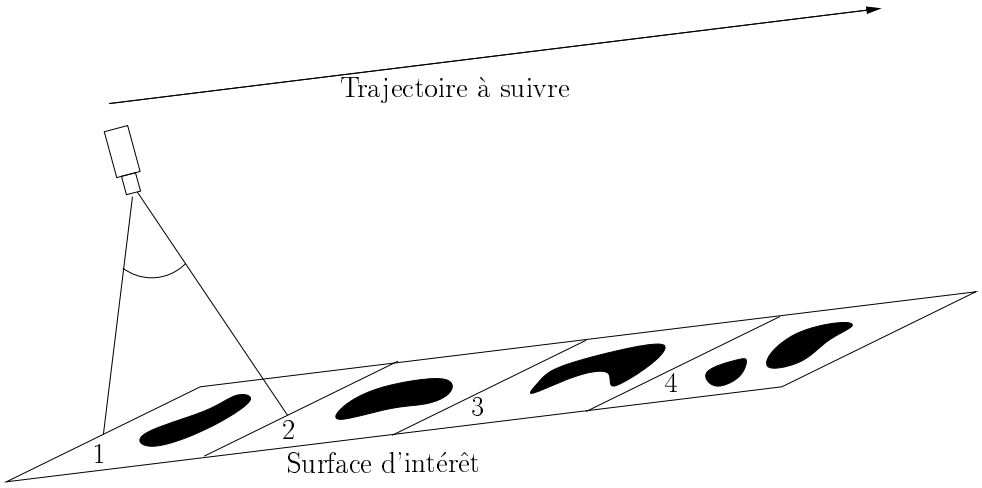


FIG. 4.36 – *Exemple de trajectoire associée à l'observation d'une surface plane*

En définitive, la tâche à laquelle nous nous intéressons dans ce paragraphe consiste, lorsque l'on observe une surface plane, à imposer à la caméra un déplacement dans un plan parallèle à cette surface, dans une direction particulière et à une vitesse donnée. Ces deux dernières contraintes sont spécifiées par l'algorithme de traitement d'image fonctionnant en aval de la tâche robotique. Nous pouvons scinder la tâche en deux parties : l'une consistant à placer la caméra parallèlement au plan de référence, et l'autre ayant pour objectif de se déplacer dans ce plan.

Cela revient donc à réaliser une liaison virtuelle plan sur plan et à se déplacer en utilisant les trois degrés de liberté qui conservent cette liaison. Une solution aurait pu consister à scinder la commande en une tâche principale de positionnement parallèle, et une tâche secondaire de suivi de trajectoire (voir [Espiau 92] pour le formalisme des tâches secondaires). Une autre solution aurait pu être de contrôler le positionnement parallèlement au plan en boucle fermé et le suivi de trajectoire en boucle ouverte en imposant des vitesses de déplacement de la caméra prédéfinies. Cette approche soulève tous les problèmes classiques liés à une commande en boucle ouverte, et ne paraît donc pas très intéressante.

Nous choisissons donc de contrôler tous les degrés de liberté mis en jeu, par une seule tâche qui contraint à la fois l'orientation de la caméra par rapport au plan, et le mouvement apparent dans l'image, afin de suivre la trajectoire désirée.

Supposons la première partie remplie (ce qui peut évidemment être fait par la réalisation de la tâche précédente) ; la vitesse apparente dans l'image étant imposée par le contenu et les objectifs de l'algorithme de traitement d'image, elle doit être contrôlée dans la seconde partie de la tâche. Une trajectoire planaire de la caméra peut donc être obtenue

en commandant les trois degrés de liberté qui la maintiennent dans un même plan, c'est-à-dire les deux translations orthogonales à l'axe optique et la rotation autour de ce même axe. L'information nécessaire pour effectuer ce contrôle est le champ de vitesse généré par ce mouvement.

Nous ne nous soucierons pas, ici, de l'éventuel traitement d'image en aval de la tâche de suivi de trajectoire d'observation. Les contraintes de vitesse apparente dans l'image seront fixées arbitrairement.

4.4.2 Loi de commande associée

Comme nous l'avons vu pour la tâche précédente, imposer à la caméra d'être parallèle à un plan de référence revient à imposer :

$$\left(\left(\frac{\partial Z}{\partial X}\right), \left(\frac{\partial Z}{\partial Y}\right)\right)^T = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Les termes d'orientation du plan γ_1 et γ_2 apparaissent dans les paramètres affines et dans les paramètres quadratiques du mouvement. Nous avons vu, avec l'équation (4.9), que l'utilisation des paramètres affines pour contrôler l'orientation de la caméra nécessite une translation orthogonalement à l'axe optique non-nulle. Dans le cas précédent, cette contrainte était incompatible avec la tâche de fixation adjointe à celle de positionnement parallèle. Au contraire, dans le cas présent, le fait d'imposer à la caméra de se déplacer dans un plan parallèle à la surface de référence, assure justement que cette translation soit non-nulle. Par conséquent, et toujours en se référant à l'équation (4.9), le contrôle de l'orientation peut être obtenu en imposant que les termes hyperboliques du développement affine (cf. équation 1.7) soient nuls, c'est-à-dire :

$$\begin{bmatrix} hyp1 \\ hyp2 \end{bmatrix} = \frac{1}{2}\begin{pmatrix} a_1 - a_4 \\ a_2 + a_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Lorsque le déplacement de la caméra est contraint de s'effectuer dans un plan parallèle à la surface de référence, nous avons vu que les seuls degrés de liberté possibles sont les translations T_x et T_y et la rotation Ω_z. De plus, dans ce cas, les termes d'orientation γ_1 et γ_2 sont nuls. La surface de référence est supposée immobile. Aussi, en reprenant l'équation (1.6), le modèle affine de mouvement se réduit à :

$$\begin{cases} \dot{x} &= -v_x + \Omega_z y \\ \dot{y} &= -vy - \Omega_z x \end{cases}$$

Cela signifie en particulier que trois des descripteurs affines du mouvement sont nuls : la divergence et les termes hyperboliques. Par conséquent, pour contrôler la vitesse apparente, il faut contraindre les termes constants et le rotationnel $1/2(a_3 - a_2)$ du modèle

affine.

En définitive, nous choisissons pour vecteur de tâche s :

$$s = (hyp1, hyp2, c_1, c_2, rot)^T$$

dont la valeur désirée est donnée par le vecteur s^* :

$$s^* = (0, 0, c_1^*, c_2^*, rot^*)^T$$

Relation d'interaction

Pour plus de simplicité, et sans perte de généralité, nous choisissons d'imposer la translation T_z nulle. Nous aurions pu contrôler cette translation en ajoutant au vecteur s la composante de divergence div, en particulier lorsque le positionnement parallèlement au plan est obtenu. Notre choix revient à réaliser ce contrôle en boucle ouverte.

Cela pose cependant moins de problèmes que dans le cas de la commande en fixation en boucle ouverte présentée précédemment. En effet, dans le cas de la fixation la source d'erreur provient des éventuels biais dans l'estimation de la profondeur, or dans le cas présent, il n'intervient pas d'estimation pouvant être erronée. Si le système robotique est correctement calibré, alors quand le plan image est parallèle à la surface d'intérêt, avoir T_Z nulle implique que la divergence soit également nulle.

Ainsi, en imposant que ce paramètre de divergence reste nul, aurait permis de compenser d'éventuelles erreurs de calibration du bras du robot, qui font que le déplacement le long de l'axe optique n'est pas nul, même si l'on spécifie $T_z = 0$. Nous avons simplement admis ici que ces déplacements pouvaient être négligés.

Ainsi, en se basant sur l'équation (3.2), la relation d'interaction associée à s peut s'écrire :

$$\dot{s} = L\ (\dot{T}_x, \dot{T}_y, \dot{\Omega}_x, \dot{\Omega}_y, \dot{\Omega}_z)^T + f(V, r)$$

avec :

$$L = \begin{bmatrix} \frac{\gamma_1}{Z_p} & -\frac{\gamma_2}{Z_p} & 0 & 0 & 0 \\ \frac{\gamma_2}{Z_p} & \frac{\gamma_1}{Z_p} & 0 & 0 & 0 \\ -\frac{1}{Z_p} & 0 & 0 & -1 & 0 \\ 0 & -\frac{1}{Z_p} & 1 & 0 & 0 \\ -\frac{\gamma_2}{Z_p} & \frac{\gamma_1}{Z_p} & 0 & 0 & 2 \end{bmatrix}$$

et :

$$f(V,r) = \begin{pmatrix} 1/2(v_x\Omega_y + v_y\Omega_x - \gamma_1^2 v_x^2 + \gamma_2^2 v_y^2 + (\gamma_2 v_x + \gamma_1 v_y)\Omega_z + (\gamma_1 v_x - \gamma_2 v_y)v_z) \\ 1/2(-v_x\Omega_x + v_y\Omega_y + \gamma_1\gamma_2(v_x^2 - v_y^2) - (\gamma_1^2 + \gamma_2^2)v_x v_y + \dots \\ \dots + (\gamma_2 v_y - \gamma_1 v_x)\Omega_z + (\gamma_2 v_x + \gamma_1 v_y)v_z) \\ \gamma_1 v_x^2 + \gamma_1 v_x\Omega_y + \gamma_2 v_x v_y - \gamma_2 v_x\Omega_x - v_x v_z \\ \gamma_1 v_x v_y + \gamma_1 v_y\Omega_y + \gamma_2 v_y^2 - \gamma_2 v_y\Omega_x - v_y v_z \\ 1/2(v_y\Omega_y + v_x\Omega_x + (\gamma_2^2 - \gamma_1^2)v_x v_y - \gamma_1\gamma_2(v_x^2 + v_y^2) + \dots \\ \dots + (\gamma_1 v_x + \gamma_2 v_y)\Omega_z + (\gamma_1 v_y - \gamma_2 v_x)v_z \end{pmatrix}$$

représente une fonction quadratique du torseur cinématique V de la caméra et de la position r de l'objet dans le repère de la caméra.

Au voisinage de la convergence, on a γ_1 et γ_2 proche de 0. Par conséquent, la matrice L n'est plus de rang plein 5 dans ce cas, mais simplement de rang 3. Il s'agit du cas intermédiaire étudié au chapitre 3. Nous devons donc chercher à exprimer les variations de s, non plus seulement en fonction de l'accélération de la caméra, mais d'un vecteur comprenant à la fois des composantes d'accélération et de vitesse. Pour cela nous devons discrétiser certaines composantes de l'accélération. La matrice L étant toujours de rang au moins 3, il suffit a priori de discrétiser seulement deux composantes.

Si nous reprenons le cas où l'on se situe au voisinage de la convergence, on note que les deux premières composantes de $\dot{s}$ ne dépendent plus de l'accélération. Plus précisément, la matrice L vaut alors :

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\dfrac{1}{Z_p} & 0 & 0 & -1 & 0 \\ 0 & -\dfrac{1}{Z_p} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Ces deux premiers termes sont les dérivées des paramètres hyperboliques, et permettent donc principalement de contrôler l'orientation du plan. Par conséquent, nous choisissons de discrétiser les accélérations angulaires autour des axes $\overrightarrow{x}$ et $\overrightarrow{y}$, de façon à faire apparaître un lien entre les dérivées des termes hyperboliques et les vitesses de rotation autour des mêmes axes. Ce choix a été jugé préférable à celui qui consistait à discrétiser les accélérations en translation le long de ces mêmes axes, car il semble plus logique que l'orientation soit directement contrôlée par des rotations.

Ainsi en effectuant l'approximation que les valeurs γ_1 et γ_2 sont nulles - ce qui est vrai dès que les deux premiers termes de s ont convergé vers leur valeur désirée -, la relation

d'interaction associée à s peut s'écrire :

$$\dot{s} = M\ (\dot{T}_x, \dot{T}_y, \Omega_x, \Omega_y, \dot{\Omega}_z)^T + \delta_s \tag{4.20}$$

avec :

$$M = \begin{bmatrix} 0 & 0 & v_y & v_x & 0 \\ 0 & 0 & -v_x & v_y & 0 \\ -\frac{1}{Z_p} & 0 & 0 & -\frac{1}{\delta t} & 0 \\ 0 & -\frac{1}{Z_p} & \frac{1}{\delta t} & 0 & 0 \\ 0 & 0 & v_x & v_y & 2 \end{bmatrix} \quad \text{et} \quad \delta_s = \left(0, 0, \frac{1}{\delta t}\widehat{\Omega_y}, -\frac{1}{\delta t}\widehat{\Omega_x}, 0\right)^T$$

et où $\widehat{\Omega_x}$ et $\widehat{\Omega_y}$ représentent respectivement les mesures des vitesses de rotation Ω_x et Ω_y à l'itération précédente.

Loi de commande

La matrice M de la relation (4.20) étant inversible, nous pouvons appliquer une commande de type gradient pour atteindre l'objectif fixé par la tâche. Pour cela, nous imposons une décroissance exponentielle de l'erreur, avec un gain λ. Cette loi de commande est donnée par :

$$(\dot{T}_x, \dot{T}_y, \Omega_x, \Omega_y, \dot{\Omega}_z)^T = -\widehat{M}^{-1}(\lambda(\widehat{s} - s^*) + \widehat{\delta s}) \tag{4.21}$$

où :

- $\widehat{M}$ est une approximation de M dans laquelle :
  - les termes v_x et v_y sont estimés à partir de la mesure des paramètres constants du mouvement et des mesures odométriques de la rotation de la caméra

$$v_x \approx \widehat{v}_x = -c_1 - \widehat{\Omega_y} \quad \text{et} \quad v_y \approx \widehat{v}_y = -c_2 + \widehat{\Omega_y}$$

  - la profondeur est calculée à partir des estimations précédentes de v_x et v_y et des mesures odométriques des translations T_x et T_y et, sachant qu'au moins une de ces deux valeurs est non-nulle, on a, par exemple

$$Z_p = \frac{\widehat{T}_x}{\widehat{v}_x}$$

- le temps δt est mesuré par l'horloge du système de contrôle ;

- $\widehat{s}$ représente la mesure de s par l'algorithme d'estimation du mouvement ;
- $\widehat{\delta_s}$ est une estimation de δ_s obtenue à partir des mesures de $\widehat{\Omega_x}$, $\widehat{\Omega_y}$ et δt.

Notons que pour éviter que la matrice M ne soit singulière au début de l'asservissement, celui-ci doit commencer avec une translation orthogonale à l'axe optique non-nulle.

4.4.3 Résultats de simulation

Nous avons testé la loi de commande (4.21) en simulation, dans les conditions suivantes :

- la cadence correspond à la fréquence vidéo, c'est-à-dire $\delta t = 0.04$ s ;
- la valeur du gain de décroissance λ est fixée à 0.5 ;
- le plan de référence est le plan horizontal d'équation $Z = 0$ dans le repère fixe ;
- la trajectoire imposée est composée de trois parties, c'est-à-dire que s^* prend trois valeurs différentes au cours de l'asservissement. Ces valeurs successives sont :
  - de l'itération 1 à l'itération 400 : $s^* = (0, 0, 0.05, 0, 0)^T$,
  - de l'itération 401 à l'itération 650 : $s^* = (0, 0, 0.05, 0, 0.314)^T$,
  - de l'itération 651 à l'itération 1000 : $s^* = (0, 0, 0.05, 0, 0)^T$,

  cela signifie donc que, dans un premier temps, le champ de mouvement correspondra à une translation le long de l'axe $\overrightarrow{x}$, puis à la composée de ce mouvement avec une rotation de l'image, et enfin, à nouveau, à un mouvement de translation le long de l'axe $\overrightarrow{x}$;
- les erreurs d'orientation initiales étaient de 30 degrés autour de l'axe $\overrightarrow{x}$ et 45 degrés autour de l'axe $\overrightarrow{y}$;
- la position initiale du centre optique de la caméra dans le repère fixe était au point de coordonnées $(0, 0, 1)^T$;
- les translations T_x et T_y étaient, au départ, respectivement égales à 0.2 et 0.05 m/s.

Il convient de noter que la valeur imposée au terme rotationnel du modèle affine de mouvement, pendant la seconde phase de la trajectoire, n'est pas choisi au hasard. En effet, une telle valeur imposée pendant 250 itérations à la cadence vidéo, permet d'obtenir théoriquement, au final, une rotation dans l'image d'à peu près 90 degrés.

Nous présentons six courbes de résultats concernant cette simulation dans les conditions décrites ci-dessus. Il s'agit tout d'abord de courbes rendant compte des composantes du vecteur de tâche, avec :

- les paramètres constants du modèle affine de mouvement, sur la figure (4.37) a, avec c_1 en trait plein et c_2 en trait pointillé ;
- les paramètres hyperboliques, sur la figure (4.37) b, avec $hyp1$ en trait plein et $hyp2$ en trait pointillé ;
- et enfin le paramètres rotationnel sur la figure (4.38) a.

Nous montrons ensuite les différentes valeurs des commandes permettant d'obtenir le résultat demandé pour la tâche, c'est-à-dire :

- les accélérations en translation orthogonalement à l'axe optique, sur la figure (4.38) b, avec $\dot{T}_x$ en trait plein et $\dot{T}_y$ en trait pointillé ;
- les vitesses de rotation autour des axes orthogonaux à l'axe optique, sur la figure (4.39) a, avec Ω_x en trait plein et Ω_y en trait pointillé ;
- l'accélération angulaire autour de l'axe optique, sur la figure (4.39) b.

Nous pouvons constater sur ces différentes courbes que, malgré les erreurs initiales sur les paramètres constants, dues à des translations T_x et T_y ne correspondant pas au départ aux valeurs désirées de s^*, et malgré, également, les erreurs angulaires initiales importantes, la convergence est rapidement obtenue pour les termes constants et hyperboliques. De plus, ces convergences sont obtenues sans oscillations des commandes en translation le long des axes $\overrightarrow{x}$ et $\overrightarrow{y}$ et en rotation autour de ces mêmes axes. La convergence du terme rotationnel vers sa valeur désirée pendant la deuxième phase du mouvement, est assez lentement obtenue. L'erreur n'a pas le temps de s'annuler complètement avant que s^* change à nouveau de valeur. On assiste alors à un phénomène parfaitement symétrique pour faire décroître le terme rotationnel de la valeur 0.314 à la valeur nulle. Ce comportement est confirmé par la réactivité relativement faible de la commande en accélération angulaire autour de l'axe optique. Une solution pour régler ce problème, pourrait être de fixer une vitesse de décroissance plus élevée pour le terme rotationnel.

Remarque : Lorsque nous avons établi la loi de commande (4.21), nous avons considéré que s^* était constant, et nous n'en avons donc pas tenu compte dans le calcul de la dérivée de l'erreur $(s - s^*)$. Cependant, dans le cas considéré pour cette simulation, s^* varie en deux occasions, et il aurait été plus juste d'en tenir compte, comme cela est fait dans [Berry 97], pour une commande en asservissement visuel géométrique. Cependant, dans

Simulation. Génération d'un trajectoire d'observation d'une surface plane

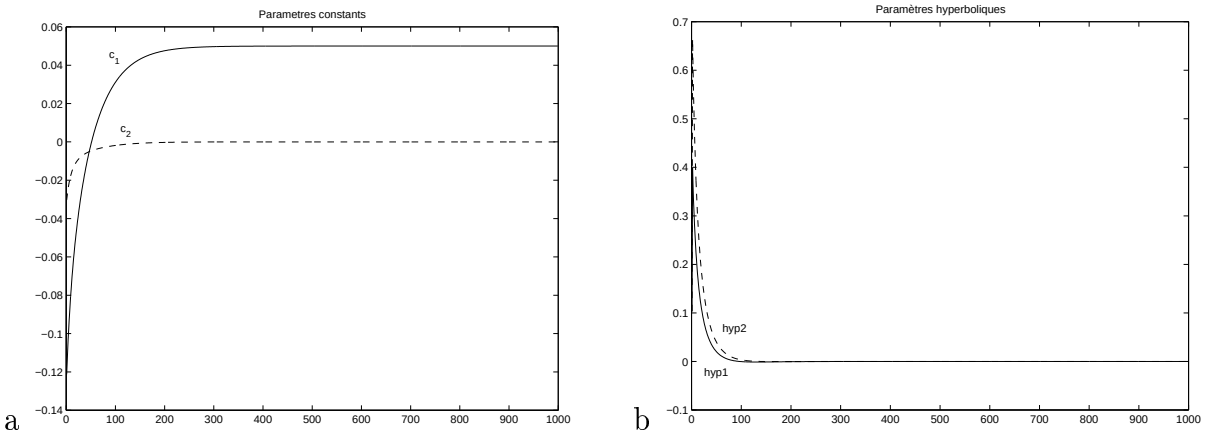


FIG. 4.37 – *Fonction de tâche : paramètres constants (a), paramètres hyperboliques(b)*

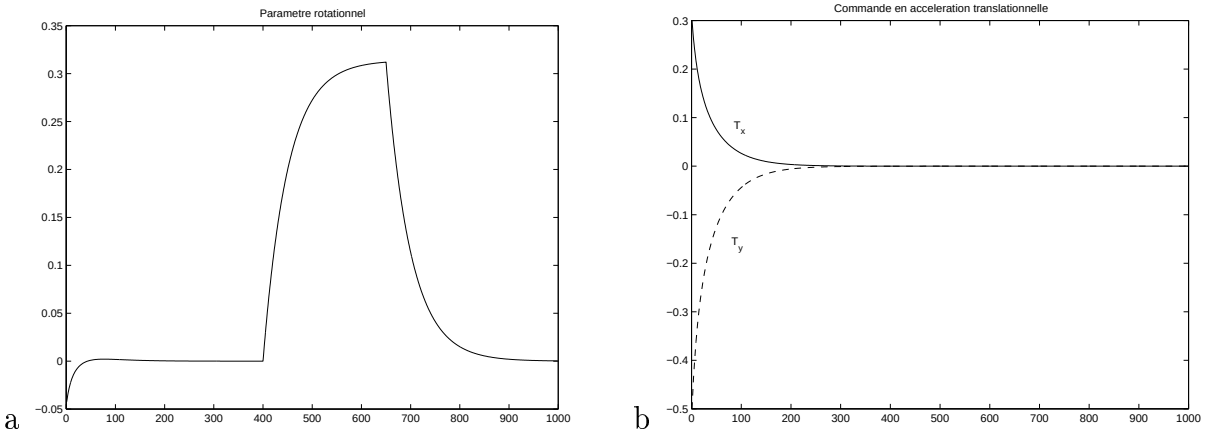


FIG. 4.38 – *Fonction de tâche : paramètre rotationnel (a). Commande en accélération translationnelle (b)*

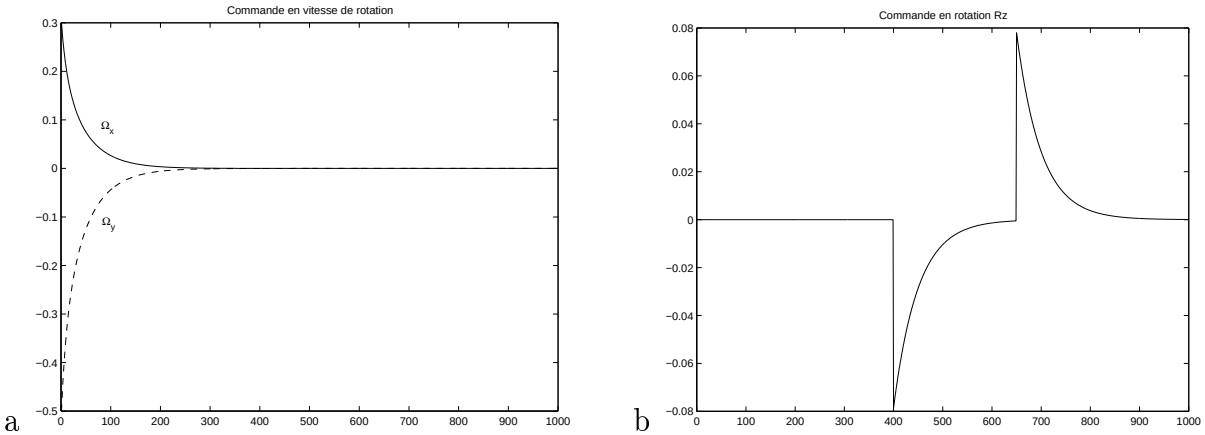


FIG. 4.39 – *Commande angulaire : vitesse autour des axes x et y (a), accélération autour de l'axe z (b)*

cet article, le mouvement du robot est justement généré par le fait que la consigne géométrique imposée est variable. Vu que nous travaillons avec une consigne en vitesse dans l'image, s^* n'a de raison de varier dans notre cas, que lorsque la trajectoire change de direction, de rayon de courbure, ou de vitesse de parcours. Nous n'avons donc pas jugé utile de prendre en compte les deux variations ponctuelles de s^* qui est constant par morceaux.

Pour illustrer les résultats obtenus, nous présentons deux nouvelles courbes. Il s'agit d'une part, de la position du centre optique de la caméra dans le repère fixe (cf. figure (4.40) a), et d'autre part, de la projection de ce même point sur un plan parallèle au plan de référence (cf. figure (4.40) b). Sur chacune de ces courbes, la position initiale est repérée par un rond et la position finale par un triangle.

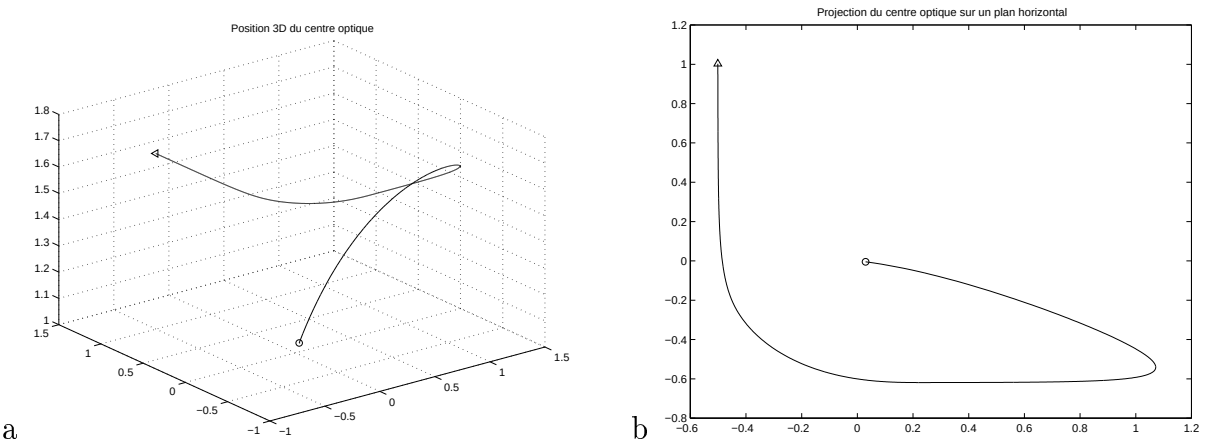


FIG. 4.40 – *Position 3D du centre optique de la caméra (a). Projection de cette position sur un plan parallèle au plan de référence (b)*

On peut noter sur ces deux courbes qu'une fois la convergence atteinte pour les termes hyperboliques du modèle affine, c'est-à-dire lorsque la caméra est correctement orientée par rapport à la surface de référence, la trajectoire imposée est correctement suivie. En particulier, la rotation de 90 degrés recherchée pendant la deuxième phase de cette trajectoire est bien obtenue. Il faut noter que la phase transitoire permettant d'orienter la caméra parallèlement au plan de référence, dure mois de 100 itérations.

Influence des erreurs d'approximation de M

Lorsque la loi de commande (4.21) a été déterminée, nous avons réalisé l'approximation consistant à négliger tous les termes ayant γ_1 ou γ_2 en facteur. Nous avons donc souhaité tester les limites de cette approximation lorsque les erreurs angulaires initiales sont plus importantes que celles considérées dans l'exemple de simulation précédent. Aussi, nous

avons effectué une nouvelle simulation avec des valeurs de γ_1 et γ_2 égales toutes les deux à 5, ce qui correspond à des erreurs angulaires de 80 degrés. Toutes les autres conditions de l'expérimentation précédente demeurent inchangées. Nous présentons sur la figure (4.41), la position 3D du centre optique de la caméra dans le repère fixe.

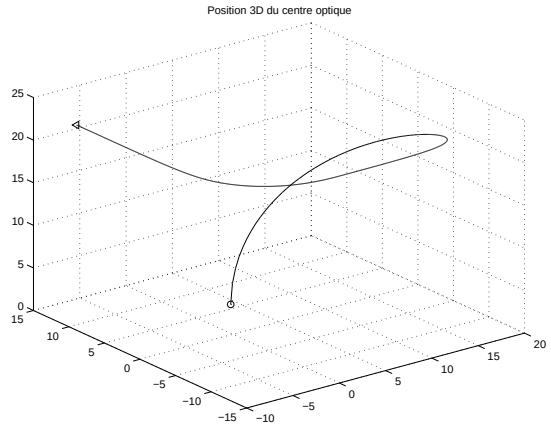


FIG. 4.41 – *Position 3D du centre optique de la caméra pour une erreur angulaire initiale de 80 degrés sur chaque axe*

Nous pouvons donc remarquer que la loi de commande est robuste à l'approximation considérée sur les valeurs γ_1 et γ_2. En effet, sur la courbe précédente, le contrôle de l'orientation est réalisé correctement. Une fois la convergence atteinte pour cette mise en parallèle, il est normal que la trajectoire soit bien suivie, vu qu'alors l'approximation réalisée est justifiée. Cependant, la hauteur du plan dans lequel la caméra se déplace dans ce cas (environ 22 m), est peu réaliste en pratique.

4.4.4 Résultats sur site robotique

Nous avons également testé la loi de commande (4.21) sur la cellule robotique. Les conditions de l'expérimentation étaient les suivantes :

- le plan de référence est horizontal dans le repère fixe, à un ou deux degrés près de précision ;
- les erreurs angulaires initiales étaient de 15 degrés autour de l'axe $\overrightarrow{x}$ et 12 autour de l'axe $\overrightarrow{y}$ - ces erreurs sont plus faibles qu'en simulation, car il se pose un problème d'observation du plan de référence lorsque ces angles sont trop élevés : nous n'avions pas abordé cette question en simulation, en supposant les dimensions du plan infinies ;

- les vitesses de translation initiales sont respectivement, $T_x = 10$ mm/s et $T_y = 5$ mm/s ;

- les images traitées par l'algorithme d'estimation du mouvement sont de taille 128×128 pixels, ce qui conduit à des itérations d'une durée de 300 ms ;
- au vu de la cadence de traitement d'image, le gain de décroissance λ est fixé à 0.2 ;
- les valeurs successives du vecteur s^* étaient :
  - de l'itération 1 à l'itération 140 : $s^* = (0, 0, 0. - 0.02, 0, 0)^T$,
  - de l'itération 141 à l'itération 250 : $s^* = (0, 0, 0.015, 0, 0)^T$,
  - de l'itération 251 à l'itération 320 : $s^* = (0, 0, 0.015, 0, -0.157)^T$,
  - de l'itération 321 à l'itération 500 : $s^* = (0, 0, 0.015, 0, 0)^T$,

  ce qui correspond donc à une translation le long de l'axe $\overrightarrow{x}$, puis une translation le long de l'axe $\overrightarrow{y}$, pendant laquelle il apparaît une phase de rotation autour de l'axe optique.

De même que pour l'expérimentation en simulation, nous montrons six courbes de résultats avec les conditions décrites ci-dessus. Il s'agit, comme précédemment, des courbes rendant compte des composantes du vecteur de tâche, avec :

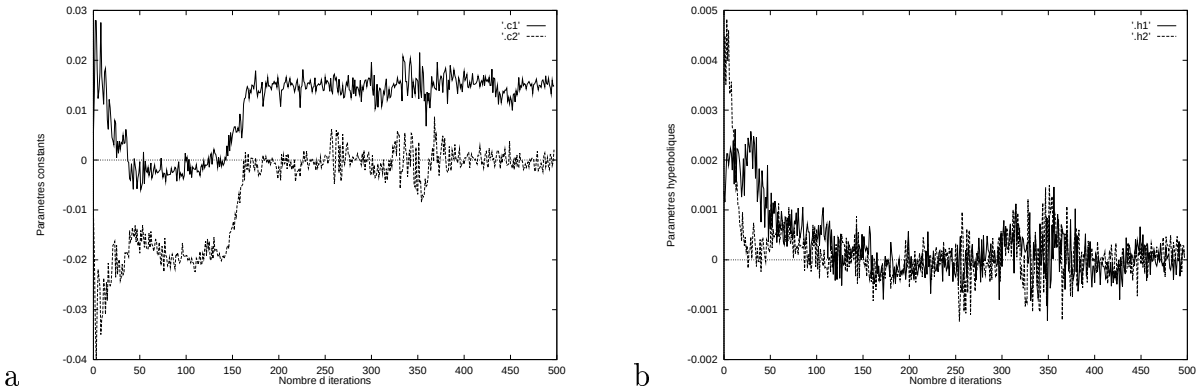
- les paramètres constants du modèle affine de mouvement, sur la figure (4.42) a, avec c_1 en trait plein et c_2 en trait pointillé ;
- les paramètres hyperboliques, sur la figure (4.42) b, avec $hyp1$ en trait plein et $hyp2$ en trait pointillé ;
- et enfin le paramètre rotationnel sur la figure (4.43) a.

Ensuite les différentes valeurs calculées par la loi de commande sont présentées, c'est-à-dire :

- les accélération en translation orthogonalement à l'axe optique, sur la figure (4.43) b, avec $\dot{T}_x$ en trait plein et $\dot{T}_y$ en trait pointillé ;
- les vitesses de rotation autour des axes orthogonaux à l'axe optique, sur la figure (4.44) a, avec Ω_x en trait plein et Ω_y en trait pointillé ;
- l'accélération angulaire autour de l'axe optique, sur la figure (4.44) b.

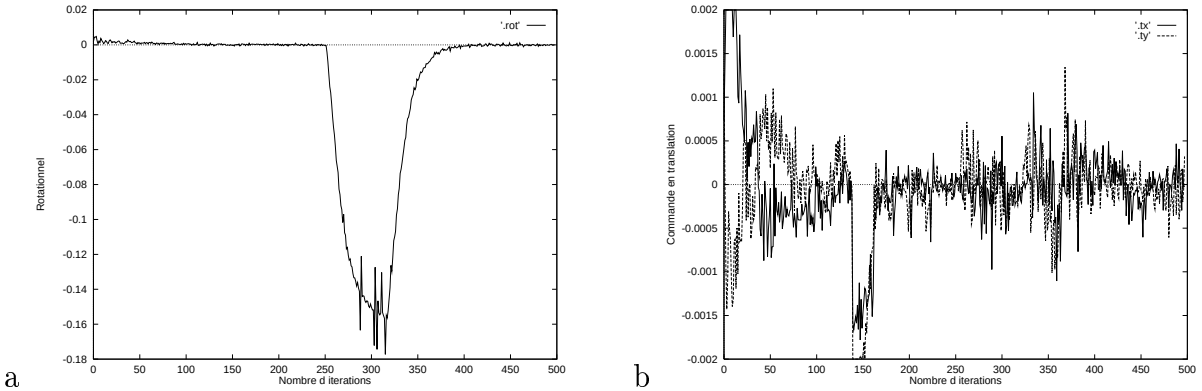
Chacune des composantes du vecteur de tâche converge bien vers les valeurs successives qui lui sont imposées. Cette convergence est atteinte malgré le bruit relativement élevé sur

Expérimentation sur site robotique. Génération d'un trajectoire d'observation d'une surface plane



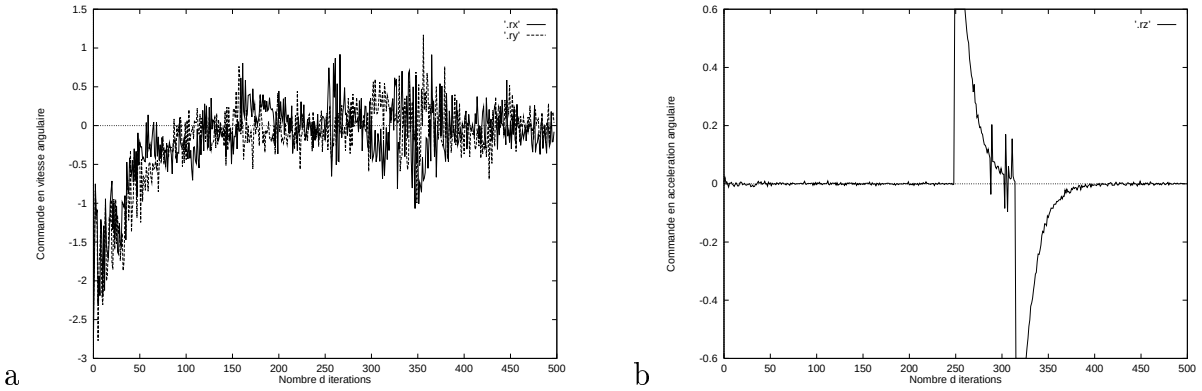
a b

FIG. 4.42 – *Fonction de tâche : paramètres constants (a), paramètres hyperboliques(b)*



a b

FIG. 4.43 – *Fonction de tâche : paramètre rotationnel (a). Commande en accélération translationnelle (b)*



a b

FIG. 4.44 – *Commande angulaire : vitesse autour des axes x et y (a), accélération autour de l'axe z (b)*

les commandes en translation et en vitesse angulaire autour des axes $\vec{x}$ et $\vec{y}$. L'une des explications de ce phénomène réside dans le fait qu'une partie de la commande est donnée en accélération, alors que la cadence de l'asservissement est peu élevée (un peu plus de 3 Hz). Le rafraîchissement des valeurs d'accélération est donc trop lent pour assurer un comportement plus "lisse" du système. Une autre explication est que les termes constants et les termes hyperboliques sont très interdépendants et donc, qu'une action pour corriger l'erreur sur un de ces paramètres a une influence sur un autre. Cela signifie, par exemple, qu'une commande en rotation pour contrôler l'orientation (c'est-à-dire les paramètres hyperboliques) fait varier les paramètres constants, puisque la vitesse angulaire apparaît explicitement dans leur expression.

Il faut noter en revanche, qu'en ce qui concerne le paramètre rotationnel et la commande en accélération angulaire autour de l'axe optique qui permet de le contrôler, le comportement est nettement moins bruité. Cela est justement dû à la relative indépendance de ce terme par rapport aux autres. De plus, au début de la tâche, lorsque la caméra n'est pas encore parallèle au plan, la valeur désirée de Ω est nulle, et la rotation Ω_z l'est aussi. Nous sommes donc dans une situation stable en ce qui concerne ce paramètre rotationnel et l'entrée de commande qui le régule. Lorsque la contrainte imposée au terme rotationnel n'est plus nulle, alors la convergence est atteinte pour l'orientation. Or, dans cette configuration, sa valeur est précisément réduite à celle de Ω_z. Nous pouvons donc considérer que le contrôle du terme Ω est découplé des autres termes. Enfin, toujours à propos du terme rotationnel, le même phénomène de réaction relativement lente, déjà remarqué en simulation, apparaît ici aussi.

Egalement, comme dans les résultats de simulation, nous montrons deux autres courbes illustrant le comportement du système pendant l'asservissement. Il s'agit de la trajectoire 3D du centre optique de la caméra, présentée sur la figure (4.45) a, et de sa projection sur un plan horizontal, c'est-à-dire sur un plan parallèle au plan de référence (cf. figure (4.45) b). De même qu'en simulation, la position de départ est repérée par un rond, et celle d'arrivée par un triangle.

Si le contrôle de la vitesse apparente dans l'image est bruité, on peut cependant remarquer que la direction de déplacement de la caméra est moins perturbée. Une fois la convergence atteinte pour l'orientation de la caméra par rapport au plan de référence, la trajectoire reste planaire. Sur la projection 2D des positions successives du centre optique, nous voyons que le comportement du mouvement de la caméra est proche de celui qui était souhaité.

Nous pouvons noter, en particulier lors de la dernière phase de translation dans l'image (après l'itération 320), que la hauteur de la caméra dans le repère fixe diminue. Cela est dû aux légères erreurs de calibration du passage poignet-caméra, qui font qu'un déplacement le long des axes $\vec{x}$ ou $\vec{y}$ génère un faible déplacement le long de l'axe $\vec{z}$. La caméra effectuant au cours de cette tâche une translation importante le long de ces deux premiers axes, le défaut est ici perceptible. Il reste cependant très limité puisqu'après un

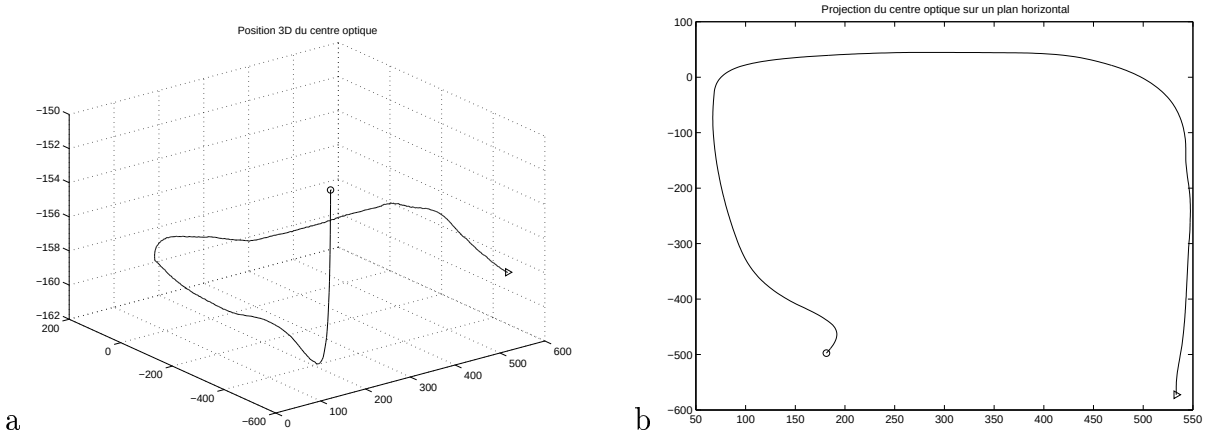


FIG. 4.45 – *Position 3D du centre optique de la caméra (a). Projection de cette position sur un plan parallèle au plan de référence (b)*

déplacement en translation orthogonalement à l'axe optique d'environ 1.5 m, le biais le long de l'axe optique est de l'ordre de 5 mm. Comme nous l'avons précisé au début de ce paragraphe, une commande de la divergence *div* dans l'image aurait sans doute permis d'éviter ce problème.

Remarque: La calibration du bras du robot permet en particulier de passer du repère de l'effecteur vers celui de la caméra. Elle permet donc également d'obtenir la position du centre optique. Cette information n'a évidemment pas été utilisée dans la boucle de commande pour assurer que le comportement du système corresponde à celui qui était spécifié géométriquement (trajectoire rectiligne, rotation, etc.), mais simplement pour pouvoir visualiser les courbes de la figure (4.45).

Conclusion

Nous avions émis, dans le chapitre précédent, quelques idées pour résoudre le problème de la commande d'un système robotique par asservissement visuel basé sur des informations dynamiques. Le principe général est de déterminer une relation d'interaction linéaire entre les paramètres de mouvement dans l'image et le déplacement de la caméra. L'objectif est alors de pouvoir contrôler le système en accélération. Cependant, il existe des cas où cette commande en accélération sur chacun des axes contraints n'est pas possible. Il faut alors discrétiser en partie la relation d'interaction, tout en continuant à rechercher la meilleure linéarisation, de façon à contrôler à la fois en accélération sur certains axes et en vitesse sur les autres.

Les expérimentations présentées dans ce chapitre permettent de valider cette approche.

En effet, pour chacune des tâches que nous avons étudiées ici, le comportement du système en simulation répondait aux exigences fixées. Cela prouve donc que la linéarisation de l'équation liant les variations des paramètres visuels dynamiques et les mouvements de la caméra, est valable, abstraction étant alors faite des erreurs de mesure. Elle l'est dans les cas où la matrice d'interaction avec le vecteur d'accélération est de rang plein, et que nous considérons alors les termes quadratiques en vitesse comme une mesure de l'état à l'itération précédente. Cela concerne notament, les deux tâches de poursuite d'objet mobile que nous avons vu dans ce chapitre, que ce soit par une commande pan et tilt ou une poursuite rigide.

La linéarisation est également valide lorsque cette matrice n'est pas de rang plein, et que nous devons alors discrétiser une ou plusieurs des composantes du vecteur d'accélération, de manière à pouvoir découpler le contrôle en vitesse sur les axes discrétisés et en accélération sur les autres. Les expérimentations rentrant dans ce cadre sont premièrement, l'alignement de la caméra avec un plan couplé avec une tâche de fixation et deuxièmement, la génération d'une trajectoire d'observation d'une surface.

Dans ce second cas, nous n'avions pas pu énoncer de principes généraux permettant de définir une condition de stabilité du système. Les résultats obtenus laissent malgré tout penser que les simplifications effectuées pour linéariser l'équation différentielle décrivant le système ne sont pas abusives.

Cependant, en ce qui concerne la tâche de suivi en pan et tilt, nous avons vu que la méthode proposée au chapitre 2, basée sur le contrôle de la position estimée du centre de la cible donne de meilleurs résultats. Il en est de même pour toutes les tâches qui peuvent être spécifiées par des contraintes géométriques, comme par exemple la fixation que l'on couple à la tâche d'alignement notament (principalement car le temps de calcul est incompatible avec une comande en accélération).

Certaines tâches présentées dans ce chapitre ne peuvent par contre pas être définies géométriquement sans une connaissance forte de l'environnement. La tâche d'alignement du plan de projection de l'image avec un plan est possible avec notre approche sans connaissance sur le contenu visuel de l'objet. Pour résoudre ce problème avec des primitives géométriques, il est nécessaire de posséder des informations telles que les distances relatives entre plusieurs couples de points, l'angle entre deux droites, etc.. Il en est de même pour la tâche de génération de trajectoire, qui est impossible à réaliser en asservissement visuel 2D, sans l'utilisation de marqueurs dédiés spécifiquement au problème.

Enfin, nous devons tout de même noter qu'une commande en accélération, telle que nous l'effectuons à présent, est difficilement compatible avec le temps de calcul nécessaire pour l'estimation du mouvement. Cela implique l'obligation d'utiliser des gains de décroissance faibles. Par conséquent, les convergences obtenues restent assez lentes.

Conclusion

Partant du constat que la plupart des travaux réalisés en asservissement visuel sont possibles, en pratique, uniquement grâce à l'utilisation d'objets dédiés à l'application visée, c'est-à-dire, en particulier, permettant d'optimiser le traitement d'image, nous nous sommes interrogés sur la réponse à apporter dans le cas d'un environnement réel. Sachant, d'autre part, qu'à partir d'une séquence d'images au contenu complexe, il est plus simple d'extraire des informations de mouvement apparent que des données géométriques, nous avons cherché à déterminer dans quelles mesures de telles informations visuelles dynamiques pourraient être intégrées dans une boucle de commande robotique.

Pour cela, nous nous sommes inspirés des travaux d'asservissement visuel basé sur des mesures géométriques dans l'image, eux-mêmes étant une application de la commande par fonction de tâche, dans le cas d'un capteur de vision. Nous avons donc montré qu'une telle approche est possible, en développant deux aspects différents.

Le premier concerne une technique d'asservissement visuel où la consigne est spécifiée par une contrainte géométrique, mais où la mesure est un modèle de champ de mouvement dans l'image. Pour être à même de comparer l'état courant du système avec l'état désiré, la vitesse des points d'intérêt considérés, déterminée par application du modèle estimé, est intégrée afin de remonter à une information géométrique. Une tâche de suivi d'objet mobile de type surveillance a été implémentée en se basant sur ce principe. Les résultats obtenus sont satisfaisants, même lorsque l'objet considéré possède un mouvement aussi complexe que celui d'un piéton.

Nous avons, dans un second temps, développé l'idée proposée initialement dans l'article [Sundareswaran 96] concernant une nouvelle technique d'asservissement visuel, où le mouvement dans l'image sert à déterminer la consigne à suivre par le système. Nous parlons alors d'asservissement visuel dynamique. Le principe consiste à définir une boucle de commande visant à obtenir l'égalité entre le champ de déplacement apparent entre deux images consécutives et un champ désiré. En suivant le modèle établi pour l'asservissement visuel géométrique, nous parvenons dans notre cas à une équation différentielle non-linéaire. L'application de quelques idées pour la linéariser nous a permis, malgré tout, de déterminer des lois de commandes relativement simples, que nous avons mises en pratique pour réaliser plusieurs tâches robotiques (poursuite d'objet mobile, positionnement de la caméra parallèlement à un plan de référence, suivi d'une trajectoire d'observation

d'une surface plane).

Les expérimentations, en simulation et sur site robotique, nous ont permis de passer du cadre théorique au domaine du réalisable en pratique. Cependant, même si la puissance de calcul utilisée nous a permis d'obtenir les résultats pratiques que nous avons présentés, nous restons encore au stade où cette approche n'est pas réellement applicable (hormis, tout de même, la tâche de suivi en pan et tilt du chapitre 2), au vu des temps de convergence nécessaires (près d'une minute pour le positionnement parallèle, à une précision d'un degré).

Cependant, en ce qui concerne les applications réalisées grâce au schéma d'asservissement visuel dynamique, malgré cette durée assez élevée entre deux itérations, la convergence souhaitée est correctement atteinte. De même, la répétabilité des résultats montre la robustesse de l'approche, en dépit des approximations effectuées, surtout lors de la linéarisation de la relation d'interaction entre les paramètres du mouvement et les déplacements de la caméra.

Dans l'ensemble des travaux réalisés dans le cadre de l'asservissement visuel, cette étude constitue plutôt une application faisant se rencontrer les techniques de commande par fonction de tâche et celle d'extraction du mouvement apparent dans l'image, qu'une base théorique pouvant être développée dans plusieurs directions (il semble, par exemple, peu probable de pouvoir asservir un système à partir d'information d'accélération dans l'image). Pour employer une métaphore, nous pourrions dire que sur l'arbre de l'asservissement visuel, les travaux présentés dans ce mémoire constitue plus une branche terminale, qu'une branche proche du tronc pouvant se ramifier.

Cependant, il reste encore de nombreux points théoriques à étudier à l'intérieur de cette étude. Il serait intéressant de se pencher sur une démonstration théorique de la stabilité, et d'en définir les limites, en particulier, dans le cas où la matrice d'interaction en facteur de l'accélération n'est pas de rang plein. De même, nous pourrions étudier plus précisément l'influence du retard introduit par la linéarisation sur cette stabilité.

Une voie également ouverte serait de voir si l'équation différentielle ne pourrait pas être résolue numériquement, en ligne, au cours de l'asservissement pour éviter ces éventuels problèmes dus au retard.

De manière plus pratique, comme nous l'avons vu, les informations de mouvement dans l'image contiennent des données sur la configuration de l'environnement dans lequel la caméra évolue. Par exemple, lorsque le modèle de mouvement apparent est calculé sous l'hypothèse de planarité de l'objet observé, la profondeur et l'orientation du plan moyen apparaissent explicitement dans les paramètres alors estimés. Or, on sait que l'un des problèmes en asservissement visuel géométrique réside dans l'approximation de la matrice d'interaction par méconnaissance de ces données. Nous pourrions donc, envisager de coupler l'utilisation d'informations géométriques et dynamiques, de manière à robustifier

l'asservissement visuel.

Dans le cadre d'applications de l'asservissement visuel dynamique, et en reprenant la métaphore précédente, même si l'asservissement visuel dynamique est plutôt une branche de l'arbre du couplage vision et robotique que l'une de ses racines, il peut également être associé à d'autres "branches" pour des applications complexes, tel que :

- la planification de l'observation d'une scène et sa reconstruction tridimensionnelle en utilisant les nombreux travaux effectués dans le domaine intitulé "structure from motion" ;
- l'exploration d'environnements inconnus : la prospection sous-marine par exemple, en se basant sur la tâche de génération de trajectoire d'observation de surface pour analyser les fonds marins, ou encore sur celle de suivi, afin de fixer l'attention sur une zone d'intérêt, malgré les mouvements parasites inévitables de tangage et de roulis de l'engin sous-marin ;
- la conception de systèmes robotiques copiés sur le monde vivant, par l'intermédiaire de la neuro-science, car, comme nous l'avons noté au départ, de nombreux êtres vivants ont une vision principalement basée sur la détection du mouvement.

Annexe A

Précision de l'algorithme RMRm d'estimation des paramètres d'un modèle polynomial de mouvement apparent dans l'image

Dans le premier paragraphe, nous donnons les résultats de quelques tests effectués sur la cellule robotique, où, en imposant un mouvement connu à la caméra, nous estimons les paramètres du mouvement apparent dans l'image, afin de les comparer à la valeur attendue. Dans le second, nous présentons également les résultats d'expérimentations permettant de comparer l'estimation du mouvement avec l'algorithme RMRm et celle obtenue par la technique décrite au chapitre 2, utilisant des mesures géométriques dans l'image, grâce à une cible de type "points blancs sur fond noir".

A.1 Tests dans le cas de mouvements connus de la caméra

Pour évaluer les performances de l'algorithme RMRm d'estimation des paramètres du mouvement, nous avons réalisé plusieurs tests sur notre cellule robotique. Ces tests ont consisté à déplacer la caméra avec un torseur cinématique constant. Plus précisément, pour chaque test, une seule composante de ce torseur cinématique possède une valeur non-nulle. Nous supposons que les consignes en vitesse imposées à la caméra sont respectées. De même, nous sommes en mesure d'obtenir de données sur la configuration du système dans son environnement, c'est-à-dire, en particulier que nous pouvons connaître la profondeur de champ dans l'axe optique (la variable Z_p dans le développement des paramètres quadratiques donnés dans la relation (1.6)) et l'orientation de la caméra par rapport au plan moyen de la surface observée (les valeurs γ_1 et γ_2).

L'objet observé est planaire et suffisamment texturé. Les mesures sont effectués à partir d'image de taille 256×256 pixels, et 4 niveaux de résolution sont pris en compte dans la construction des pyramides. Pour chaque mouvement effectué, les huit paramètres du modèle quadratique sont estimés.

Les valeurs que nous donnons sont exprimées en mètre/s sur le plan CCD de la caméra, pour une distance focale ramenée à 1. Dans tous les cas que nous avons considérés, lorsque la valeur attendue d'un paramètre était nulle, sa valeur moyenne était inférieure à 2.10^{-4}, pour un écart-type compris entre 2.10^{-5} et 2.10^{-4}. Nous ne donnons donc que les résultats concernant les paramètres ayant une valeur significative.

Vu que pour chacun des mouvements effectués, seule une composante du torseur cinématique est non-nulle, nous ne préciserons que la valeur de cette composante. De même, nous ne donnerons d'informations relatives à la configuration géométrique de la caméra (profondeur dans l'axe optique et orientation) que dans le cas d'un mouvement de translation.

Dans les quatre premiers cas, les valeurs attendues des paramètres non-nuls sont supposées rester constante. On donne donc leur valeur moyenne, ainsi que l'écart-type.

Mouvement : $\Omega_x = 0.01$ rad/s
$c_2 = 0.0099$ avec un écart-type de $2.4.10^{-4}$ (valeur attendue 0.01)
$q_2 = 0.0101$ avec un écart-type de $2.5.10^{-4}$ (valeur attendue 0.01)

Mouvement : $\Omega_y = 0.02$ rad/s
$c_1 = 0.0198$ avec un écart-type de $2.5.10^{-4}$ (valeur attendue 0.02)
$q_2 = 0.0196$ avec un écart-type de $4.5.10^{-4}$ (valeur attendue 0.02)

Mouvement : $\Omega_z = 0.01$ rad/s
$a_2 = 0.0104$ avec un écart-type de $2.5.10^{-4}$ (valeur attendue 0.01)
$a_3 = 0.0096$ avec un écart-type de $2.3.10^{-4}$ (valeur attendue 0.01)

Mouvement : $T_x = 0.01$ m/s, avec $\gamma_1 = \gamma_2 = 0$ et $Z_p \approx 0.8$ m
$c_1 = 0.0119$ avec un écart-type de $2.5.10^{-4}$ (valeur attendue ≈ 0.0125)

Dans les trois cas suivants, la caméra est soumise à un mouvement de translation et la profondeur dans l'axe optique varie au cours du test. Lorsque deux paramètres sont théoriquement égaux, nous donnons la moyenne de leur différence. De plus, à partir des valeurs initiales et finales de la profondeur Z_p, nous calculons les valeurs initiales et finales théoriques des paramètres considérés, que nous comparons alors aux valeurs estimées (les valeurs finales et initiales sont respectivement repérées par des indices f et i).

Mouvement : $T_z = 0.01$ m/s, avec $\gamma_1 = \gamma_2 = 0$, $Z_{p,i} \approx 1.1$ m et $Z_{p,f} \approx 0.5$ m
$a_1 - a_4 = 1.8.10^{-5}$ (en théorie $a_1 = a_4 = v_z$)
$a_{1,i} = 0.0093$ (valeur attendue ≈ 0.0091)
$a_{1,f} = 0.0196$ (valeur attendue ≈ 0.02)

Mouvement : $T_x = 0.01$ m/s, avec $\gamma_1 = 1$, $\gamma_2 = 0$, $Z_{p,i} \approx 1.2$ m et $Z_{p,f} \approx 0.85$ m
(en théorie $c_1 = -v_{c,x}$ et $a_1 = \gamma_1 v_{c,x}$)
$c_{1,i}$ = -0.0085 (valeur attendue $\approx$ -0.0083)
$c_{1,f}$ = -0.0117 (valeur attendue $\approx$ -0.0117)
$a_{1,i} = 0.0079$ (valeur attendue ≈ 0.0083)
$a_{1,f} = 0.0107$ (valeur attendue ≈ 0.0117)

Mouvement : $T_z = 0.01$ m/s, avec $\gamma_1 = 1$, $\gamma_2 = 0$, $Z_{p,i} \approx 1.2$ m et $Z_{p,f} \approx 0.55$ m
$a_1 - a_4 = -5.2.10^{-6}$ (en théorie $a_1 = a_4 = v_z$)
$a_{1,i} = 0.0083$ (valeur attendue ≈ 0.0083)
$a_{1,f} = 0.0182$ (valeur attendue ≈ 0.0181)
$q_{1,i} = 0.0083$ (valeur attendue ≈ 0.0083)
$q_{1,f} = 0.0172$ (valeur attendue ≈ 0.0181)

Ces quelques expérimentations nous permettent d'admettre la bonne précision de l'estimation des paramètres du modèle quadratique du mouvement apparent.

A.2 Comparaison avec une estimation basée sur des mesures géométriques

Au chapitre 2, nous avons été amenés, pour valider la pertinence de notre approche, à considérer une estimation des paramètres constants du mouvement se basant sur des mesures géométriques dans l'image, disponibles grâce à l'utilisation d'un objet de type "points blancs sur fond noir". Nous avons alors pu constater qu'il n'y a pas de biais dans l'estimation de la position du centre de l'objet par intégration de sa vitesse apparente déduite du modèle de mouvement. Pour que nous puissions également l'affirmer dans le cas où l'estimation de la vitesse est réalisée avec l'algorithme RMRm, nous devons comparer ces deux méthodes d'estimation du mouvement, en particulier, en ce qui concerne les paramètres constants du mouvement.

Rappelons tout d'abord le principe de l'estimation à partir de mesures géométriques. L'objet considéré est en pratique composé de quatre points, disposés aux sommets d'un carré. La carte de traitement d'image EDIXIA IA 1000, nous fournit, à la cadence vidéo, la position de la projection de ces points dans l'image, à une précision de l'ordre du dixième de pixel. Grâce à une mesure de ces positions dans deux images successives et du temps

écoulé entre les deux acquisitions, nous obtenons la vitesse de chaque point. L'application d'un modèle affine de mouvement permet d'obtenir un système linéaire sur-dimensionné (8 équations, 6 paramètres) que l'on résout par moindres carrés. L'utilisation du modèle affine, plutôt que d'un simple modèle à deux paramètres, permet de robustifier l'estimation des paramètres constants.

Dans les différentes expérimentations, l'objet dont le mouvement a été estimé, se déplace en translation sur le banc présenté au chapitre 2. Ce déplacement s'est effectué dans une direction orthogonale à l'axe optique, à une distance d'environ 1.25 m de l'objectif, sur une distance de 40 cm. La caméra quant à elle, reste immobile. Cependant, le robot est tout de même placé sous alimentation, afin de prendre en compte, dans l'estimation du mouvement, les parasites résultant sur le signal vidéo. On admet que dans une telle configuration, la vitesse apparente dans l'image reste constante, pour peu, bien-sûr, que celle de l'objet le long du banc le soit aussi. Pour l'algorithme RMRm, la taille des images acquises est de 128×128 pixels, et la surface de l'objet représente environ un quart de celle de l'image, c'est-à-dire que l'estimation se fait sur une zone de taille environ 60×80 pixels. Dans ces conditions, le temps de traitement est de 40 ms ; c'est le même que pour l'estimation à partir des informations géométriques fournies par la carte EDIXIA IA 1000.

Dans le tableau A.1, nous présentons les résultats des estimations pour chacune des deux méthodes. Pour chacune des vitesses de l'objet considérées ($\overrightarrow{V} = 0$ cm/s, 5 cm/s et 10 cm/s), nous donnons les vitesses apparentes moyennes sur chaque axe ($\overline{\dot{x}}$ et $\overline{\dot{y}}$), ainsi que l'écart-type correspondant. Tous ces résultats sont donnés en pixel/s. Nous ne donnons pas de résultats pour des vitesses supérieures de l'objet le long du banc car, pour de telles valeurs, le déplacement entre deux images successives est trop important, et peut faire échouer le traitement d'image.

	IA 1000		RMRm	
$\overrightarrow{V} = 0$ cm/s	$\overline{\dot{x}}$	$\overline{\dot{y}}$	$\overline{\dot{x}}$	$\overline{\dot{y}}$
	0.01	0.01	0.01	-0.01
	écart-type sur $\dot{x}$	écart-type sur $\dot{y}$	écart-type sur $\dot{x}$	écart-type sur $\dot{y}$
	6.31	7.05	0.33	0.34
$\overrightarrow{V} = 5$ cm/s	$\overline{\dot{x}}$	$\overline{\dot{y}}$	$\overline{\dot{x}}$	$\overline{\dot{y}}$
	38.76	-8.27	39.10	-8.86
	écart-type sur $\dot{x}$	écart-type sur $\dot{y}$	écart-type sur $\dot{x}$	écart-type sur $\dot{y}$
	6.77	8.81	2.66	0.87
$\overrightarrow{V} = 10$ cm/s	$\overline{\dot{x}}$	$\overline{\dot{y}}$	$\overline{\dot{x}}$	$\overline{\dot{y}}$
	76.27	-18.33	73.58	-18.57
	écart-type sur $\dot{x}$	écart-type sur $\dot{y}$	écart-type sur $\dot{x}$	écart-type sur $\dot{y}$
	7.91	8.94	3.17	0.99

TAB. A.1 – *Comparaison de l'estimation des paramètres constants du mouvement par un algorithme utilisant des informations géométriques et par l'algorithme RMRm*

Nous pouvons remarquer tout d'abord que les résultats obtenus sont cohérents, c'est-à-dire que : d'une part, pour une vitesse donnée de l'objet, les déplacements estimés dans l'image sont identiques pour les deux méthodes d'estimation, et d'autre part, lorsque la vitesse de l'objet est doublée, sa vitesse apparente dans l'image est également doublée.

Si nous regardons plus précisément les écarts-types associés à chacune des valeurs moyennes de vitesse apparente, nous pouvons noter que dans le cas où l'algorithme RMRm est utilisé, l'estimation est moins bruitée que dans le cas de l'algorithme se basant sur des mesures géométriques, et ce, malgré la précision de l'ordre du dixième de pixel sur ces mesures. En particulier, lorsque le mouvement est nul, l'écart-type est très faible dans le premier cas (moins de 0.5 pixel), alors qu'il est de l'ordre de 6 à 7 pixels dans le second.

Pour l'estimation par l'algorithme RMRm, l'écart-type semble être proportionnel à

la vitesse estimée. Ainsi, vu que la direction du mouvement est plutôt selon l'axe $\overrightarrow{x}$ de l'image, le bruit d'estimation est plus faible pour $\dot{y}$ que pour $\dot{x}$.

Au contraire, dans l'autre cas, même si le bruit augmente avec la vitesse, il est important quelque soit cette vitesse. On peut noter, également que ce bruit est toujours plus fort pour $\dot{y}$, alors que l'objet se déplace dans une direction plutôt orthogonale à cet axe. Cela peut s'expliquer par le fait que l'acquisition des images se fait, lorsqu'on utilise la carte IA 1000, par des trames successives correspondant, respectivement, aux lignes paires et impaires de l'image. Plus précisément, entre deux acquisitions successives de trames, permettant de reconstruire une image complète, il s'écoule un temps, certes faible (20 ms), mais pouvant générer un léger décalage, qui sera principalement ressenti dans la direction orthogonale à ces lignes, c'est-à-dire la direction $\overrightarrow{y}$ de l'image.

En conclusion, nous pouvons affirmer que le test de validation étudié au chapitre 2, et permettant de prouver que l'estimation de la position du centre de gravité de l'objet mobile était non-biaisée, est correcte, vu que l'estimation du mouvement apparent par l'algorithme RMRm est moins bruitée que lorsqu'on utilise des informations géométriques dans l'image.

Annexe B

Influence des paramètres de calibration sur le modèle de mouvement apparent dans l'image

Quelque soit l'algorithme utilisé pour l'estimation des paramètres du modèle de mouvement entre deux images successives, le résultat est fourni, dans un premier temps, sous la forme du modèle de déplacement dans l'image, exprimé en pixel. Pour obtenir, la correspondance avec le mouvement apparent en mètre, sur le plan CCD de la caméra, il faut connaître, outre le temps séparant l'acquisition des deux images, les paramètres intrinsèques de calibration de la caméra, c'est-à-dire : la position de l'intersection de l'axe optique et du plan CCD, le rapport mètre-pixel sur chaque axe et la distorsion radiale.

Nous donnons, dans un premier temps, le modèle de la projection perspective de l'espace tridimensionnel métrique observé par la caméra sur l'espace bidimensionnel pixelique de l'image. Puis nous donnons l'expression des paramètres de mouvement dans l'image exprimés en mètre en fonction du mouvement de la caméra, de la position dans l'espace de la scène qu'elle observe (profondeur et orientation) et de ses paramètres intrinsèques.

B.1 Modélisation de la projection

Nous rappelons succinctement dans ce paragraphe le modèle de projection sténopée d'une caméra, pour laquelle le phénomène de distorsion est supposé se réduire à une composante radiale.

Nous considérons que l'origine du repère de la caméra est située au centre de projection. L'axe $\overrightarrow{x}$ de ce repère est orienté de la même manière que l'axe des abscisses de l'image, et de même pour l'axe $\overrightarrow{y}$ et l'axe des ordonnées de l'image. La distance minimale du centre de projection au plan image, appelée distance focale, est notée f. L'intersection de

l'axe $\overrightarrow{z}$ du repère caméra avec le plan image a pour coordonnées (x_c, y_c) dans l'image numérisée.

Dans le repère caméra, la projection perspective d'un point $M = (X, Y, Z)^T$ de l'espace 3D, sur le plan image est $m = (x, y)^T$ avec :

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}$$

Soit K_d le coefficient de distorsion radiale, alors la position du point m réellement observé dans l'image est $m_d = (x_d, y_d)^T$ avec :

$$\begin{cases} x_d = x + K_d.x.(x^2 + y^2) \\ \\ y_d = y + K_d.y.(x^2 + y^2) \end{cases} \tag{B.1}$$

Nous obtenons alors, pour la position $m_p = (x_p, y_p)^T$ de m dans l'image, exprimée en pixel :

$$\begin{cases} x_p = x_c + \dfrac{x_d}{l_x} \\ \\ y_p = y_c + \dfrac{y_d}{l_y} \end{cases} \tag{B.2}$$

où l_x et l_y sont des coefficients proportionnels à la taille d'un pixel élémentaire.

A partir de ces équations, on peut écrire le modèle complet de la caméra :

$$\begin{cases} x_p = x_c + F_x.\dfrac{X}{Z} + K_{df}.F_x.\dfrac{X}{Z}.\left(\dfrac{X^2}{Z^2} + \dfrac{Y^2}{Z^2}\right) \\ \\ y_p = y_c + F_y.\dfrac{Y}{Z} + K_{df}.F_y.\dfrac{Y}{Z}.\left(\dfrac{X^2}{Z^2} + \dfrac{Y^2}{Z^2}\right) \end{cases} \tag{B.3}$$

où $F_x = f/l_x$, $F_y = f/l_y$ et $K_{df} = K_d.f^2$.

B.2 Passage pixel-mètre des paramètres du mouvement

Les paramètres dans l'équation (1.6) correspondent à un modèle de mouvement pour des coordonnées image exprimées en mètre. Ce sont les dérivées de ces paramètres qui définissent les relations d'interaction. Cependant, l'algorithme d'estimation du mouvement fournit un modèle de mouvement calculé avec des coordonnées image exprimées en pixel. Nous devons donc déterminer les relations de passage d'une représentation à l'autre. Nous

notons sans indice les paramètres qui dépendent d'une mesure en mètre et avec un indice p ceux qui dépendent d'une mesure en pixel.

A partir de la relation (B.3), nous pouvons exprimer les dérivées $\dot{x}_p$ et $\dot{y}_p$ en fonction de x, y et de leur dérivée. Pour obtenir le modèle de mouvement en pixel, nous devons alors exprimer x et y en fonction de x_p et y_p. La relation (B.3) étant non-linéaire, cette expression est délicate à établir. Par conséquent, pour obtenir le passage de la représentation mètre vers la représentation pixel, nous utilisons la relation linéaire :

$$\begin{cases} x_p = x_c + \dfrac{x}{l_x} \\ y_p = y_c + \dfrac{y}{l_y} \end{cases}$$

Nous pouvons donc écrire, d'une part :

$$\begin{cases} \dot{x}_p = \dfrac{\dot{x}}{l_x} \\ \dot{y}_p = \dfrac{\dot{y}}{l_y} \end{cases}$$

et d'autre part :

$$\begin{cases} x = x_p l_x - x_c \\ y = y_p l_y - y_c \end{cases}$$

En reportant ces valeurs dans la relation (1.6), nous obtenons :

$$\begin{cases} \dot{x_p} &= c_{1,p} + a_{1,p}x_p + a_{2,p}y_p + q_{1,p}x_p^2 + q_{2,p}x_p y_p \\ \dot{y_p} &= c_{2,p} + a_{3,p}x_p + a_{4,p}y_p + q_{1,p}x_p y_p + q_{2,p}y_p^2 \end{cases}$$

avec :

$$\begin{cases} c_{1_p} &= (c_1 - a_1 x_c - a_2 y_c + q_1 x_c^2 + q_2 x_c y_c)/l_x \\ c_{2_p} &= (c_2 - a_3 x_c - a_4 y_c + q_1 x_c y_c + q_2 y_c^2)/l_y \\ a_{1,p} &= a_1 - 2q_1 x_c - q_2 y_c & a_{2_p} &= (a_2 - q_2 x_c) l_y / l_x \\ a_{3,p} &= (a_3 - q_1 y_c) l_x / l_y & a_{4,p} &= a_4 - q_1 x_c - 2q_2 y_c \\ q_{1,p} &= q_1 l_x & q_{2,p} &= q_2 l_y \end{cases}$$

Ces relations s'écrivent sous forme matricielle :

$$\begin{pmatrix} c_{1_p} \\ c_{2_p} \\ a_{1,p} \\ a_{2_p} \\ a_{3,p} \\ a_{4,p} \\ q_{1,p} \\ q_{2,p} \end{pmatrix} \begin{bmatrix} 1/l_x & 0 & -x_c/l_x & -y_c/l_x & 0 & 0 & x_c^2/l_x & x_c y_c/l_x \\ 0 & 1/l_y & 0 & 0 & -x_c/l_y & -y_c/l_y & x_c y_c/l_y & y_c^2/l_y \\ 0 & 0 & 1 & 0 & 0 & 0 & -2x_c & -y_c \\ 0 & 0 & 0 & l_y/l_x & 0 & 0 & 0 & -x_c l_y/l_x \\ 0 & 0 & 0 & 0 & l_x/l_y & 0 & -y_c l_x/l_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -x_c & -2y_c \\ 0 & 0 & 0 & 0 & 0 & 0 & l_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & l_y \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ q_1 \\ q_2 \end{pmatrix}$$

En inversant cette relation linéaire, on obtient l'équation qui permet le passage des paramètres du modèle pixel vers le modèle mètre :

$$\begin{pmatrix} c_1 \\ c_2 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ q_1 \\ q_2 \end{pmatrix} \begin{bmatrix} l_x & 0 & x_c & y_c l_x/l_y & 0 & 0 & x_c^2/l_x & x_c y_c/l_y \\ 0 & l_y & 0 & 0 & x_c l_y/l_x & y_c & x_c y_c/l_x & y_c^2/l_y \\ 0 & 0 & 1 & 0 & 0 & 0 & 2x_c/l_x & y_c/l_y \\ 0 & 0 & 0 & l_x/l_y & 0 & 0 & 0 & x_c/l_y \\ 0 & 0 & 0 & 0 & l_y/l_x & 0 & y_c/l_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & x_c/l_x & 2y_c/l_y \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/l_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/l_y \end{bmatrix} \begin{pmatrix} c_{1_p} \\ c_{2_p} \\ a_{1,p} \\ a_{2_p} \\ a_{3,p} \\ a_{4,p} \\ q_{1,p} \\ q_{2,p} \end{pmatrix}$$

B.3 Influence de la distorsion radiale

Dans le calcul des paramètres du modèle quadratique du mouvement au chapitre 1, nous avions considéré, sans perte de généralité, la distance focale égale à 1. En reprenant ce calcul, et en remplaçant, dans l'équation (1.3), les valeurs de x et y par celles données à l'équation (B.1), nous obtenons un nouveau modèle de mouvement quadratique. Il faut noter que le modèle exact n'est plus quadratique dans ce cas, mais est en fait d'ordre 4, en raison de l'introduction du coefficient de distorsion radiale. L'approximation à l'ordre 2 de ce modèle est donnée par :

$$\begin{cases} \dot{x} &= c_1 + a_1 x + a_2 y + qd_1 x^2 + qd_2 xy + qd_3 y^2 \\ \dot{y} &= c_2 + a_3 x + a_4 y + qd_4 x^2 + qd_5 xy + qd_6 y^2 \end{cases} \tag{B.4}$$

où les différents paramètres en jeu valent respectivement :

$$\begin{cases} c_1 = f(v_x + \Omega_y) & c_2 = f(v_y - \Omega_x) \\ a_1 = -f(\gamma_1 v_x + v_z) & a_2 = -f(\gamma_2 v_x + \Omega_z) \\ a_3 = f(-\gamma_1 v_y + \Omega_z) & a_4 = -f(\gamma_2 v_y + v_z) \\ qd_1 = f(3K_{df} v_x + \gamma_1 v_z + (1 + 3K_{df})\Omega_y) & qd_2 = f(2K_{df} v_y + \gamma_2 v_z - (1 + 2K_{df})\Omega_x) \\ qd_3 = f K_{df}(v_x + \Omega_y) & qd_4 = f K_{df}(v_y - \Omega_x) \\ qd_5 = f(2K_{df} v_x + \gamma_1 v_z + (1 + 2K_{df})\Omega_y) & qd_6 = f(3K_{df} v_y + \gamma_2 v_z - (1 + 3K_{df})\Omega_x) \end{cases}$$

Nous pouvons donc constater que :

- la position du centre de projection (x_c, y_c) n'a pas d'influence sur le modèle ;
- les paramètres sont tous proportionnels à f, que l'on peut donc supposer égal à 1 sans perte de généralité ;

– la distorsion n'a une influence que sur les paramètres quadratiques. A titre d'illustration, la caméra avec laquelle les expérimentations ont été effectuées au chapitre 4 possède un coefficient de distorsion K_{df} de l'ordre de -0.14. Il n'est donc pas possible, par exemple, de négliger $3K_{df}$ devant 1 dans l'expression de qd_1.

Pour l'estimation des paramètres quadratiques, nous utilisons le résultat donné pour le modèle quadratique simplifié à huit paramètres, et la valeur du terme K_{df} de la calibration, supposée relativement fine, c'est-à-dire :

$$\begin{cases} q_1 = qe_1 - 3K_{df}c_1 \\ q_2 = qe_2 - 3K_{df}c_2 \end{cases}$$

où les termes qe_i sont les paramètres quadratiques issus de l'estimation du modèle à 8 paramètres.

Nous aurions cependant pu retrouver les paramètres déterminés dans le chapitre 1, sans avoir recours à la calibration de la caméra, en notant que :

$$\begin{cases} q_1 = qd_1 - 3qd_3 \\ q_2 = qd_6 - 3qd_4 \end{cases}$$

Cependant, ce calcul nous aurait obligé à déterminer le modèle quadratique complet du mouvement, à 12 paramètres. Cela aurait impliqué des temps de calcul prohibitifs à chaque itération.

Bibliographie

[Allen 93] P.K. Allen, A. Timcenko, B. Yoshimi, P. Michelman. – Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. on Robotics and Automation*, 9(2):152–165, Avril 1993.

[Aloimonos 88] Y. Aloimonos, I. Weiss, A. Bandyopadhyay. – Active vision. *International Journal of Computer Vision*, 1(4):333–356, Janvier 1988.

[Araújo 96] H. Araújo, J. Batista, P. Peixoto, J. Dias. – Pursuit control in binocular active vision system using optical flow. – *IAPR Int. Conf. on Pattern Recognition*, pp. 313–317, Vienna, Austria, Août 1996.

[Arias 98] S. Arias, E. Coste-Manière, G. Giraudon, T. Viéville. – Implementing oculomotor behaviors. – 1998. Disponible sur le site web : http://www.inria.fr/robotvis/personnel/vthierry/Demos/Mike/paper/paper.html.

[Arsénio 97] A. Arsénio, J. Santos-Victor. – Robust visual tracking by an active observer. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 1342–1347, Grenoble, France, Septembre 1997.

[Bandyopadhyay 85] A. Bandyopadhyay, J. Aloimonos. – *Perception of rigid motion from spatio-temporal derivatives of optical flow.* – Technical report n° 157, University of Rochester, Department of Computer Science, Mars 1985.

[Barron 94] J. Barron, D. Fleet, S. Beauchemin. – Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, Janvier 1994.

[Bartholomeus 93] M.G.P. Bartholomeus, B.J.A. Kröse, A.J. Noest. – A robust multi-resolution vision system for target tracking with a moving camera.

Computer Science in the Netherlands, éd. par H. Wijshof, pp. 52–63. – Amsterdam, CWI, Novembre 1993.

[Batista 97] J. Batista, P. Peixoto, H. Arajo. – Real-time visual behaviours with a binocular vision system. – *IEEE Int. Conf. on Robotics and Automation*, pp. 3391–3396, Albuquerque, NM, Avril 1997.

[Bensalah 95] F. Bensalah, F. Chaumette. – Compensation of abrupt motion changes in target tracking by visual servoing. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 181–187, Pittsburgh, Août 1995.

[Bergen 92] J.R. Bergen, P. Anandan, K.J. Hanna, R. Hingorani. – Hierarchical model-based motion estimation. – *European Conf. on Computer Vision*, vol. 588, pp. 237–252, Santa Margherita Ligurie, Italy, Mai 1992. Springer-Verlag.

[Berry 97] F. Berry, P. Martinet, J. Gallice. – Trajectory generation by visual servoing. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 1066–1072, Grenoble, France, Septembre 1997.

[Black 94] M. Black. – Recursive non-linear estimation of discontinuous flow fields. – *European Conf. on Computer Vision*, pp. 138–145, Stockholm, Suède, 1994. Springer-Verlag.

[Bouthemy 90] P. Bouthemy, P. Lalande. – Detection and tracking of moving objects based on statistical regularization method in space and time. – *European Conf. on Computer Vision*, pp. 307–311, Antibes, France, Avril 1990. Springer-Verlag.

[Bouthemy 93] P. Bouthemy, E. François. – Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, 10(2):157–182, Avril 1993.

[Braud 97] P. Braud, M. Dhome, J.T. Lapresté, B. Peuchot. – Reconnaissance, localisation et suivi d'objets polyédriques par vision multi-oculaire. *Technique et Science Informatiques*, 16(1):9–37, Janvier 1997.

[Campani 90] M. Campani, A. Verri. – Computing optical flow from an overconstrained system of linear algebraic equations. – *IEEE Int. Conf. on Computer Vision*, pp. 22–26, Osaka, Japon, Décembre 1990.

[Capurro 97] C. Capurro, F. Panerai, G. Sandini. – Dynamic vergence using log-polar images. *International Journal of Computer Vision*, 24(1):79–94, Janvier 1997.

[Chaumette 90] F. Chaumette. – *La relation vision-commande : théorie et application à des tâches robotiques.* – PhD. Thesis, Université de Rennes I, Juillet 1990.

[Chaumette 93] F. Chaumette, A. Santos. – Tracking a moving object by visual servoing. – *12^{th} World Congress IFAC*, vol. 9, pp. 409–414, Sydney, Australia, Juillet 1993.

[Chaumette 98] F. Chaumette. – Potential problems of stability and convergence in image-based and position-based visual servoing. *The Confluence of Vision and Control*, éd. par G. Hager, D. Kriegman, A.S. Morse. – Springer-Verlag, 1998.

[Christensen 95] H.I. Christensen, J. Horstmann, T. Rasmussen. – A control of theoretical approach to active vision. – *Asian Conf. on Computer Vision*, vol. 1035 of *Lecture Notes in Computer Science*, pp. 201–210, Singapore, Décembre 1995.

[Cipolla 97a] R. Cipolla, A. Blake. – Image divergence and deformation from closed curves. *International Journal of Robotics Research*, 16(1):77–96, Janvier 1997.

[Cipolla 97b] R. Cipolla, N. Hollinghurst. – Visually guided grasping in unstructured environments. *Elsevier - Robotics and Autonomous Systems*, 19:337–346, 1997.

[Colombo 95] C. Colombo, B. Allotta, P. Dario. – Affine visual servoing : A framework for relative positioning with a robot. – *IEEE Int. Conf. on Robotics and Automation*, pp. 464–471, Nagoya, Japan, Mai 1995.

[Coombs 93] D. Coombs, K. Roberts. – Centering behavior using peripheral vision. – *Int. Conf. on Computer Vision and Pattern Recognition*, pp. 440–445, New York City, NY, Juin 1993.

[Coombs 98] D. Coombs, M. Herman, T-H. Hong, M. Nashman. – Real-time obstacle avoidance using central flow divergence, and peripherical flow. *IEEE Trans. on Robotics and Automation*, 14(1):49–59, Février 1998.

[Corke 93] P.I. Corke, M.C. Good. – Controller design for high performance visual servoing. – *12th World congress IFAC*, vol. 9, pp. 395–398, Sydney, Australia, Juillet 1993.

[Crétual 97] A. Crétual, F. Chaumette. – Positionning a camera parallel to a plane using dynamic visual servoing. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 43–48, Grenoble, France, Septembre 1997.

[Crétual 98a] A. Crétual, F. Chaumette. – Image-based visual servoing by integration of dynamic measurements. – *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 1994–2001, Leuven, Belgique, Mai 1998.

[Crétual 98b] A. Crétual, F. Chaumette. – *Stabilisation dynamique d'une caméra pan and tilt par asservissement visuel sur des images sous-marines.* – Rapport de recherche, IRISA - IFREMER, Juillet 1998. Rapport final de convention.

[Crétual 98c] A. Crétual, F. Chaumette, P. Bouthemy. – Complex object tracking by visual servoing on 2D image motion. – *IAPR Int. Conf. on Pattern Recognition*, vol. 2, pp. 1251–1254, Brisbane, Australie, Août 1998.

[Daniilidis 98] K. Daniilidis, C. Krauss, M. Hansen, G. Sommer. – Real-time tracking of moving objects with an active camera. *Real-Time Imaging*, 4(1):3–20, Février 1998.

[DawsonHowe 96] K.M. Dawson-Howe. – *Active surveillance using dynamic background subtraction.* – Rapport de Recherche n° TCD-CS-96-06, Dublin, Ireland, Dept. of Computer Science, Trinity College, Août 1996.

[Dev 94] A. Dev, B.J.A. Kröse, L. Dorst, F.C.A. Groen. – Observer curve and object detection from the optic flow. – *SPIE Conference on Intelligent Robots and Computer Vision*, pp. xx–yy, Décembre 1994.

[Dev 97] A. Dev, B. Kröse, F. Groen. – Navigation of a mobile robot on the temporal development of the optic flow. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 558–563, Grenoble, France, Septembre 1997.

[Dev 98] A. Dev, B.J.A. Kröse, F.C.A. Groen. – "Where are you driving to?" Heading direction for a mobile robot from optical flow. – *IEEE*

Int. Conf. on Robotics and Automation, pp. 1578–1583, Leuven, Belgium, Mai 1998.

[Du 94] F. Du, M. Brady. – A four degree-of-freedom robot head for active vision. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(6):1439–1469, Décembre 1994.

[Duchon 98] A.P. Duchon, W.H. Warren, L.P. Kaelbling. – Ecological robotics. *Adaptative behavior*, 6(3/4):471–505, 1998. – Special issue on Biologically Inspired Models of Spatial Navigation.

[Eklundh 95] J-O. Eklundh, T. Uhlin, P. Nordlund, A. Maki. – Developing an active observer. – *Asian Conf. on Computer Vision : Recent developments in computer vision*, vol. 1035 of *Lecture Notes in Computer Science*, pp. 181–190, Singapore, 1995. – Invited session papers.

[Espiau 92] B. Espiau, F. Chaumette, P. Rives. – A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, Juin 1992.

[Feddema 90] J.T. Feddema, C.S.G. Lee. – Adaptative image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Trans. on Systems, Man, and Cybernetics*, 20(5):1172–1183, Septembre 1990.

[Feddema 93] J.T. Feddema, C.S.G. Lee, O.R. Mitchell. – Feature-based visual servoing of robotic systems. *Visual servoing. Real-time control of robot manipulators based on visual sensory feedback*, éd. par K. Hashimoto, pp. 71–104. – World scientific, 1993.

[Fermüller 98] C. Fermüller, L. Cheong, Y. Aloimonos. – 3d motion and shape representations in visual servo control. *International Journal of Robotics Research*, 17(1):4–18, Janvier 1998.

[Gangloff 98] J.A. Gangloff, M. de Mathelin, G. Abba. – 6 d.o.f. high speed dynamic visual servoing using GPC controllers. – *IEEE Int. Conf. on Robotics and Automation*, pp. 2008–2013, Leuven, Belgium, Mai 1998.

[Gengenbach 96] V. Gengenbach, H.H. Nagel, M. Tonko, K. Schäfer. – Automatic dismantling integrating optical flow into a machine vision-contolled robot system. – *IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1320–1325, Minneapolis, MN, Avril 1996.

[Giachetti 98] A. Giachetti, M. Campani, V. Torre. – The use of optical flow for road navigation. *IEEE Trans. on Robotics and Automation*, 14(1):34–48, Février 1998.

[Grosso 93] E. Grosso, D.H. Ballard. – Head-centered orientation strategies in animate vision. – *IEEE Int. Conf. on Computer Vision*, pp. 395–402, Berlin, Germany, Mai 1993.

[Grosso 95] E. Grosso, M. Tistarelli. – Active/dynamic stereo vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(11):1117–1127, Novembre 1995.

[Grosso 96] E. Grosso, G. Metta, A. Oderra, G. Sandini. – Robust visual servoing in 3-d reaching tasks. *IEEE Trans. on Robotics and Automation*, 12(5):732–742, Octobre 1996.

[Hager 98] G.D. Hager, K. Toyama. – X-vision: a portable substrate for real-time vision applications. – vol. 69, pp. 23–37, Janvier 1998.

[Harris 88] C.G. Harris, M. Stephens. – A combined corner and edge detector. – *Alvey Conference*, pp. 189–192, 1988.

[Hashimoto 93] K. Hashimoto, T. Ebine, K. Sakamoto, H. Kimura. – Full 3D visual tracking with nonlinear model-based control. – *American Control Conference*, pp. 3180–3185, San Francisco, California, Juin 1993.

[Hashimoto 96] K. Hashimoto, T. Ebine, H. Kimura. – Visual servoing with hand-eye manipulator - optimal control approach. *IEEE Trans. on Robotics and Automation*, 12(5):766–774, Octobre 1996.

[Heuring 96] J.J. Heuring, D.W. Murray. – Visual head tracking and slaving for visual telepresence. – *IEEE Int. Conf. on Robotics and Automation*, pp. 2908–2914, Minneapolis, MN, Avril 1996.

[Horaud 93] R. Horaud, O. Monga. – *Vision par ordinateur, outils fondamentaux.* – Hermès, Paris, France, 1993.

[Horn 81] B.K.P. Horn, B.G. Schunck. – Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[Hsu 84] Y.Z. Hsu, H.H. Nagel, G. Rekers. – New likehood test methods for change detection in image sequences. *Computer Vision, Graphics, and Image Processing*, 26:73–106, 1984.

[Inoue 92] H. Inoue, T. Tachikawa, M. Inaba. – Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation. – *IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1621–1626, Nice, France, Mai 1992.

[Irani 92] M Irani, B. Rousso, S. Peleg. – Detecting and tracking multiple moving objects using temporal integration. – *European Conf. on Computer Vision*, vol. 588, pp. 282–287, Santa Margherita Ligurie, Italy, Mai 1992. Springer-Verlag.

[Jung 98] D. Jung, J. Heinzmann, A. Zelinsky. – Range and pose estimation for visual servoing of a mobile robot. – *IEEE Int. Conf. on Robotics and Automation*, pp. 1226–1231, Leuven, Belgium, Mai 1998.

[Luo 88] R.C. Luo, R.E. Mullen Jr, D.E. Wessell. – An adaptative robotic tracking system using optical flow. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 568–573, Philadelphia, PA, Avril 1988.

[Madden 95] B.C. Madden, U.M. Cahn von Seelen. – *PennEyes: a binocular activ vision system.* – Rapport de Recherche n° MS-CIS-95-37/GRASP LAB 396, Department of Computer and Information Science, University of Pennsylvania, Grasp Laboratory, Décembre 1995.

[Malis 97] E. Malis, F. Chaumette, S. Boudet. – Positioning a coarse-calibrated camera with respect to an unknowmn planar object by 2D 1/2 visual servoing. – *5^{th} IFAC Symposium on Robot Control (SYROCO'97)*, vol. 2, pp. 517–523, Nantes, France, Septembre 1997.

[Martinet 96] P. Martinet, F. Berry, J. Gallice. – Use of first derivative of geometric features in visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, vol. 4, pp. 3413–3419, Minneapolis, MN, Avril 1996. IEEE.

[Meyer 92] F. Meyer, Bouthemy. P. – Estimation of time-to-collision maps from first order motion models and normal flows. – *IAPR Int. Conf. on Pattern Recognition*, pp. 78–82, La Haye, Pays-Bas, 1992.

[Michel 93] H. Michel, P. Rives. – *Singularities in the determination of the situation of a robot effector from the perspective view of 3 points.* – Rapport de Recherche n° 1850, INRIA, Février 1993.

[Milios 93] E. Milios, M. Jenkin, J. Tsotsos. – Design and performance of TRISH, a binocular robot head with torsional eye movements. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(1):51–68, Février 1993.

[Mitiche 96] A. Mitiche, P. Bouthemy. – Computation of image motion: a synopsis of current problems and methods. *Int. Journ. of Comp. Vis.*, 19(1):29–55, 1996.

[Mémin 98] E. Mémin, P. Pérez. – Optical flow estimation and object-based segmentation with robust techniques. *IEEE Trans. on Image Processing*, 7(5):703–719, Mai 1998.

[Murray 94] D. Murray, A. Basu. – Motion tracking with an active camera. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(5):449–459, Mai 1994.

[Murray 95] D.W. Murray, K.J. Bradshaw, P.F. Mc Lauchlan, I.D. Reid, P.M. Sharkey. – Driving saccade to pursuit using image motion. *International Journal of Computer Vision*, 16(3):205–228, Mars 1995.

[Nagel 94] H.H. Nagel, G. Socher, H. Kollnig, M. Otte. – Motion boundary detection in image sequences by local stochastic tests. – *European Conf. on Computer Vision*, pp. 305–315, Stockholm, Suède, 1994. Springer-Verlag.

[Nakabo 98] Y. Nakabo, M. Ishikawa. – Visual impedance using 1ms visual feedback system. – *IEEE Int. Conf. on Robotics and Automation*, pp. 2333–2338, Leuven, Belgium, Mai 1998.

[Negahdaripour 92] S. Negahdaripour, S. Lee. – Motion recovery from image sequences using only first order optical flow information. *International Journal of Computer Vision*, 9(3):163–184, Décembre 1992.

[Nelson 89] R.C. Nelson, J. Aloimonos. – Obstacle avoidance using flow field divergence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106, Octobre 1989.

[Nickels 98] K. Nickels, S. Hutchinson. – Weighting observations: the use of kinematic models in object tracking. – *IEEE Int. Conf. on Robotics and Automation*, pp. 1677–1682, Leuven, Belgium, Mai 1998.

[Nordlund 96] P. Nordlund, T. Uhlin. – Closing the loop: detection and pursuit of a moving object by a moving observer. *Image and Vision Computing*, 14(4):265–275, Mai 1996.

[Odobez 94] J.M. Odobez. – *Estimation, détection et segmentation du mouvement: une approche robuste et markovienne.* – PhD. Thesis, Université de Rennes I, Décembre 1994.

[Odobez 95] J.M. Odobez, P. Bouthemy. – Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, Décembre 1995.

[Odobez 97] J.M. Odobez, P. Boutheny. – *Video Data Compression for Multimedia Computing*, chap. Separation of moving regions from background in an image sequence acquired with a mobile camera, pp. 283–311. – Kluwer Academic, 1997.

[Okhotsimsky 98] D.E. Okhotsimsky, A.K. Platonov, I.R. Belousov, A.A. Boguslavsky, S.N. Emelianov, V.V. Sazonov, S.M. Sokolov. – Real-time hand eye system: interaction with moving object. – *IEEE Int. Conf. on Robotics and Automation*, pp. 1683–1688, Leuven, Belgium, Mai 1998.

[Papanikolopoulos 95] N.P. Papanikolopoulos, B. Nelson, P.K. Khosla. – Six degree-of-freedom hand/eye visual tracking with uncertain parameters. *IEEE Trans. on Robotics and Automation*, 11(5):725–732, Octobre 1995.

[Questa 95] P. Questa, E. Grossmann, G. Sandini. – Camera self orientation and docking maneuver using normal flow. – *SPIE AeroSense'95*, Orlando, Florida, Avril 1995.

[Reid 96] I.D. Reid, D.W. Murray. – Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision*, 18(1):41–60, Avril 1996.

[Samson 91] C. Samson, M. Le Borgne, B. Espiau. – *Robot control: the task function approach.* – Oxford University Press, 1991.

[SantosVictor 95] J. Santos-Victor, G. Sandini, F. Curotto, S. Garibaldi. – Divergent stereo in autonomous navigation: from bees to robots. *International Journal of Computer Vision*, 14(2):159–177, Mars 1995.

[SantosVictor 96] J. Santos-Victor, G. Sandini. – Uncalibrated obstacle detection using normal flow. *Machine Vision and Applications*, 9(3):130–137, 1996.

[SantosVictor 97] J. Santos-Victor, G. Sandini. – Embedded visual behaviors for navigation. *Elsevier - Robotics and Autonomous Systems*, 19(3-4):299–313, Mars 1997.

[Smith 97] C.E. Smith, S.A. Brandt, N.P. Papanikolopoulos. – Eye-in-hand robotic tasks in uncalibrated environments. *IEEE Trans. on Robotics and Automation*, 13(6):903–914, Décembre 1997.

[Stiller 97] C. Stiller. – Object-oriented estimation of dense motion fields. *IEEE Trans. on Image Processing*, 6(2):234–250, Février 1997.

[Subarrao 89] M. Subarrao. – Interpretation of image flow : a spatio-temporal approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(3):266–278, Mars 1989.

[Sundareswaran 96] V. Sundareswaran, P. Bouthemy, F. Chaumette. – Exploiting image motion for active vision in a visual servoing framework. *International Journal of Robotics Research*, 15(6):629–645, Décembre 1996.

[Toepfer 98] C. Toepfer, M. Wende, G. Baratoff, H. Neumann. – Robot navigation by combining central and peripheral optical flow detection on a space-variant map. – *IAPR Int. Conf. on Pattern Recognition*, vol. 2, pp. 1804–1807, Brisbane, Australie, Août 1998.

[Vassallo 98] R.F. Vassallo, H.J. Schneebeli, J. Santos-Victor. – Visual navigation: combining visual servoing and appearance based methods. – 6^{th} *Intl. Symposium on Intellignet Robotic Systems*, Edimbourg, Ecosse, Juillet 1998.

[Vieville 95] T. Vieville, E. Clergue, R. Encisco, H. Mathieu. – Experimenting with 3D vision on a robotic head. *Elsevier - Robotics and Autonomous Systems*, 14(1):1–28, Janvier 1995.

[Wang 94] J.Y.A. Wang, E.H. Adelson. – Representing moving images with layers. *IEEE Trans. on Image Processing*, 3(5):625–638, septembre 1994.

[Wavering 95] A.J. Wavering, H. Schneiderman, J.C. Fiala. – High-performance tracking with TRICLOPS. – *Asian Conf. on Computer Vision*, num-

ber 1035, pp. 171–180, Singapore, 1995. Lecture Notes in Computer Science. – Invited session papers.

[Waxman 83] A. Waxman, S. Ullman. – *Surface structure and 3-D motion from image flow: a kinematic analysis.* – Technical report n° CAR-TR-24, University of Maryland, Center for Automation Research, 1983.

[Wilson 93] W.J. Wilson. – Visual servo control of a robot using kalman filter estimates of robot pose relative to work-pieces. *Visual servoing. Real-time control of robot manipulators based on visual sensory feedback*, éd. par K. Hashimoto, pp. 71–104. – World scientific, 1993.

[Yamane 98] T. Yamane, Y. Shirai, J. Miura. – Person tracking by integrating optical flow and uniform brightness regions. – *IEEE Int. Conf. on Robotics and Automation*, pp. 3267–3272, Leuven, Belgium, Mai 1998.

[Young 92] G.S. Young, T.H. Hong, M. Herman, J.C.S. Yang. – Obstacle detection and terrain characterization using optical flow without 3-D reconstruction. – SPIE (édité par), *Inteligent Robots and Computer Vision IX*, vol. 1825, pp. 561–572, 1992.

[Zhang 95] Z. Zhang, R. Deriche, O. Faugeras, Q.T. Luong. – A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–120, Janvier 1995.

Visual servoing from image motion features

Abstract

This thesis is concerned with robotics and vision, and more precisely visual servoing. The general aim of a visual servoing scheme is to control a robot in order that visual features acquired by a camera become superimposed with a desired visual pattern.

Most of the works done in this area are based on the use of geometric features, such as coordinates of points in the image. Difficulties to extract points of interest involve, generally, to consider objects dedicated to the task. To take into account real objects, we use the extraction of parameters of motion between two images, called dynamic feature.

The first approach we develop in this dissertation consists in retrieving a geometric feature from a dynamic measurement with a simple integration step. This allows us to perform tasks such as the tracking of any mobile object.

We define, in a second approach, the concept of dynamic visual servoing. The principle is to try to obtain a desired 2D motion field in the image sequence. In the geometric case, variations of visual features are linearly linked to the camera velocity. On the contrary, in the dynamic case, the relation is quadratic and puts camera acceleration and speed into play. We thus give some propositions to linearize this relation in order to be able to control the system.

Then, we have applicated the dynamic visual servoing approach to different tasks: tracking of a moving object, positioning of the camera parallel to a plane, and trajectory tracking to observe a textured surface.

Keywords: Vision, Robotics, Visual Servoing, 2D Motion, Dynamic Visual Servoing

Résumé

Cette thèse s'inscrit dans le domaine de la vision robotique, et plus précisément celui de l'asservissement visuel. L'objectif général d'un asservissement visuel est de piloter un robot afin que les informations visuelles acquises par une caméra se superposent avec un motif visuel désiré.

La plupart des travaux effectués dans ce cadre se base sur l'utilisation d'informations géométriques, telles que les coordonnées de points dans l'image. Les difficultés d'extraire des points d'intérêt implique, en général, l'usage d'objets dédiés à la tâche considérée. Afin de prendre en compte des objets réels, nous utilisons l'extraction de paramètres de mouvement entre deux images, appelés information dynamique.

La première approche que nous développons dans ce mémoire consiste à retrouver une information géométrique à partir d'une mesure dynamique par une simple intégration. Cela nous permet de réaliser des tâches telles que la poursuite d'un objet mobile quelconque.

En calquant l'asservissement visuel géométrique, nous définissons, dans une seconde approche, le concept d'asservissement visuel dynamique. Le principe est de chercher à obtenir un champ de mouvement 2D désiré. Dans le cas géométrique, les variations de l'information visuelle sont linéairement liées à la vitesse de la caméra, ce qui permet d'établir assez facilement une loi de contrôle. Au contraire, dans le cas dynamique, la relation est quadratique et met en jeu l'accélération et la vitesse de la caméra. Nous donnons donc quelques propositions pour linéariser cette relation afin de pouvoir commander le système.

Nous avons ensuite appliqué les techniques de contrôle en asservissement visuel dynamique à différentes tâches : la poursuite d'objet mobile, le positionnement de la caméra parallèlement à un plan, et le suivi d'une trajectoire d'observation d'une surface.

Mots-clés : Vision, Robotique, Asservissement Visuel, Mouvement 2D, Asservissement Visuel Dynamique