



**HAL**  
open science

# Contributions à la modélisation et à la commande en asservissement visuel

Ezio Malis

► **To cite this version:**

Ezio Malis. Contributions à la modélisation et à la commande en asservissement visuel. Robotique [cs.RO]. Université de Rennes 1, 1998. Français. NNT: . tel-04652091

**HAL Id: tel-04652091**

**<https://inria.hal.science/tel-04652091v1>**

Submitted on 18 Jul 2024

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

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



Distributed under a Creative Commons Attribution 4.0 International License

# 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

**Ezio MALIS**

Équipe d'accueil: TEMIS, puis VISTA (IRISA, RENNES)

École doctorale: Informatique, Traitement du Signal et Télécommunications

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

Titre de la thèse :

**Contributions à la modélisation et à  
la commande en asservissement visuel**

Soutenue le 4 novembre 1998 devant la commission d'examen composée de :

Jean-Jacques	FUCHS	Président
Radu	HORAUD	Rapporteurs
Claude	SAMSON	
Roberto	CIPOLLA	Examineurs
François	CHAUMETTE	
Sylvie	BOUDET	
Jacques	POT	(Membre invité)



---

*a mia mamma*





## Remerciements

Je tiens tout d'abord à remercier les membres du jury de cette thèse :

- Jean-Jacques FUCHS, professeur à l'Université de Rennes I, pour avoir présidé le jury.
- Radu HORAUD, directeur de recherche à l'INRIA Rhône-Alpes, et Claude SAMSON, directeur de recherche à l'INRIA Sophia-Antipolis, pour m'avoir fait l'honneur d'être les rapporteurs de cette thèse. Leur regard critique a toujours été constructif et leur nombreuses observations ont grandement amélioré la qualité du manuscrit.
- Roberto CIPOLLA, directeur de recherche à l'Université de Cambridge, pour avoir accepté de participer à ce jury et m'avoir donné la possibilité de continuer ma recherche au sein de son équipe dans le cadre du projet européen VIGOR.
- Jacques POT et Sylvie BOUDET respectivement chef de groupe et ingénieur à la DER d'EDF à Chatou, pour m'avoir donné la possibilité d'effectuer cette thèse en collaboration avec l'INRIA et avoir mis à ma disposition le matériel nécessaire.
- François CHAUMETTE, chargé de recherche à l'IRISA Rennes, pour m'avoir dirigé avec rigueur durant ma thèse tout en m'accordant une très grande marge de liberté. Ce manuscrit n'aurait pas été ce qu'il est sans ses innombrables critiques et remarques (sans parler des corrections des fautes de français et des équations involontairement fausses qui l'ont quelque peu gêné pendant la lecture!).

Cette thèse a été réalisée en partie à la DER d'EDF à Chatou. Mes remerciements s'adressent donc aux agents, prestataires, doctorants et stagiaires qui ont rendu plus agréable ma permanence à Paris ; un merci aussi à tous mes amis "parisiens", et particulièrement à Alberto, Houman et Jean-Xavier (surtout en souvenir de nos conversations "scientifiques").

Un très grand merci aux membres de l'ancienne équipe TEMIS (desormais séparée en TEMICS et VISTA) et surtout à Eric pour ses conseils (non seulement informatiques), et à Etienne et Patrick que j'ai dérangé souvent avec mes questions de mathématiques. En particulier je voudrais remercier Armel et Maeva, Lionel et Géraldine, Marc et Eva, Christophe et Stéphanie, Gwenaëlle, Pierre et Fabrice pour tout ce temps agréable passé ensemble, et pour avoir su fêter, sans me blesser, moi le "rital", le succès du passage en demi-finale de la France, après les tirs au but, ce 3 juillet 1998 (France-Italie: 0-0, ce ne fut pas une défaite!).

Enfin, un dernier merci à Caroline pour m'avoir supporté ces deux dernières années (en espérant qu'elle pourra me supporter encore longtemps) ainsi que tous les membres de sa famille pour leur accueil chaleureux.



# Table des matières

<b>Notations</b>	<b>11</b>
<b>Introduction</b>	<b>13</b>
<b>1 État de l'art et modélisation</b>	<b>17</b>
1.1 Modélisation . . . . .	17
1.1.1 L'espace projectif comme extension de l'espace euclidien . . . . .	17
1.1.2 Les matrices de transformation homogènes . . . . .	19
1.1.3 Les matrices de projection perspective . . . . .	20
1.1.4 Modélisation de la caméra . . . . .	20
1.2 L'asservissement visuel . . . . .	22
1.2.1 Le principe de la commande par fonction de tâche . . . . .	23
1.2.2 Commande dynamique . . . . .	27
1.2.3 Commande en vitesse . . . . .	29
1.3 Asservissement visuel 3D . . . . .	32
1.3.1 Modélisation . . . . .	32
1.3.2 Commande et analyse de la stabilité . . . . .	35
1.4 Asservissement visuel 2D . . . . .	36
1.4.1 Modélisation . . . . .	36
1.4.2 Commande et analyse de la stabilité . . . . .	39
1.5 Conclusion . . . . .	42
<b>2 Reconstruction projective et euclidienne</b>	<b>45</b>
2.1 Introduction . . . . .	45
2.2 Modélisation . . . . .	46
2.2.1 La relation entre deux images . . . . .	48
2.2.2 La contrainte épipolaire . . . . .	48
2.3 Estimation . . . . .	50
2.3.1 Estimation linéaire de la matrice fondamentale . . . . .	50
2.3.2 Estimation de l'épipole et d'une matrice d'homographie . . . . .	51
2.3.3 Estimation linéaire d'une homographie à partir de points non coplanaires	53
2.3.3.1 Changement de coordonnées et normalisation des données . . . . .	54
2.3.3.2 Réintroduction des contraintes non linéaires . . . . .	55
2.3.4 Estimation d'une homographie à partir de points ou droites coplanaires	57

2.4	De la reconstruction projective à la reconstruction euclidienne . . . . .	58
2.4.1	Paramètres intrinsèques exacts . . . . .	58
2.4.2	Paramètres intrinsèques erronés . . . . .	60
2.5	Résultats expérimentaux . . . . .	62
2.5.1	Scène synthétique . . . . .	62
2.5.2	Cible planaire . . . . .	63
2.5.3	Cible non planaire . . . . .	63
2.5.3.1	Déplacement nul . . . . .	63
2.5.3.2	Rotation pure . . . . .	64
2.5.3.3	Translation pure . . . . .	65
2.5.3.4	Déplacement quelconque . . . . .	65
2.5.4	Scène réelle . . . . .	68
2.5.5	Expériences avec un robot et une caméra embarquée . . . . .	68
2.5.5.1	Rotation pure . . . . .	70
2.5.5.2	Translation pure . . . . .	72
2.5.5.3	Déplacement quelconque . . . . .	74
2.6	Conclusion . . . . .	76
<b>3</b>	<b>Asservissement visuel mono-caméra</b>	<b>77</b>
3.1	Asservissement visuel $2D\frac{1}{2}$ . . . . .	77
3.1.1	Modélisation . . . . .	78
3.1.1.1	Contrôle de l'orientation . . . . .	78
3.1.1.2	Contrôle de la position . . . . .	79
3.1.2	Commande proportionnelle . . . . .	81
3.1.2.1	Loi de commande . . . . .	81
3.1.2.2	Boucle fermée . . . . .	82
3.1.2.3	Analyse de la robustesse de la stabilité aux erreurs sur les paramètres caméra . . . . .	84
3.1.2.4	Analyse de la robustesse de la stabilité aux erreurs sur les paramètres caméra et caméra/effecteur . . . . .	94
3.1.3	Commande robuste en présence d'erreurs sur les paramètres caméra . . . . .	96
3.1.3.1	Loi de commande . . . . .	97
3.1.3.2	Analyse de la robustesse de la stabilité . . . . .	101
3.1.4	Commande robuste en présence d'erreurs sur les paramètres caméra et caméra/effecteur . . . . .	102
3.1.4.1	Loi de commande . . . . .	102
3.1.4.2	Analyse de la robustesse de la stabilité . . . . .	103
3.2	Asservissement visuel $2D\frac{3}{4}$ . . . . .	104
3.2.1	Modélisation . . . . .	104
3.2.2	Comparaison des asservissements $2D\frac{1}{2}$ et $2D\frac{3}{4}$ . . . . .	105
3.2.3	Commande proportionnelle . . . . .	107
3.2.3.1	Loi de commande . . . . .	108
3.2.3.2	Boucle fermée . . . . .	109

3.2.3.3	Analyse de la robustesse de la stabilité aux erreurs sur les paramètres caméra . . . . .	110
3.2.3.4	Analyse de la stabilité et de la robustesse aux erreurs sur les paramètres caméra et caméra/effecteur . . . . .	112
3.3	Conclusion . . . . .	112
<b>4</b>	<b>Résultats expérimentaux en asservissement visuel mono-caméra</b>	<b>115</b>
4.1	Conditions expérimentales . . . . .	115
4.2	Cible plane . . . . .	116
4.3	Cible non-planaire . . . . .	122
4.3.1	Calibration correcte . . . . .	123
4.3.2	Calibration grossière . . . . .	123
4.3.3	Influence du choix de la distance $\hat{d}^*$ . . . . .	126
4.4	Comparaison entre les méthodes d'asservissement visuel . . . . .	130
4.5	Commande robuste . . . . .	151
4.6	Sensibilité des méthodes d'asservissement visuel au bruit de mesure . . . . .	154
4.6.1	Résultats pour une cible plane inclinée . . . . .	155
4.6.2	Résultats pour une cible parallèle au plan image . . . . .	159
4.6.3	Résultats pour une cible 3D . . . . .	163
4.7	Conclusion . . . . .	167
<b>5</b>	<b>Asservissement visuel multi-caméras</b>	<b>169</b>
5.1	État de l'art . . . . .	169
5.2	Une tâche d'asservissement multi-caméras . . . . .	171
5.2.1	Modélisation . . . . .	172
5.2.2	Commande . . . . .	174
5.2.3	Analyse de la stabilité . . . . .	175
5.3	Résultats expérimentaux . . . . .	179
5.3.1	Comparaison de la précision entre asservissement mono et multi-caméras	180
5.3.2	Comparaison entre les asservissements multi-caméras 2D et 2D $\frac{1}{2}$ . . .	187
5.3.2.1	Calibration grossière . . . . .	187
5.3.2.2	Calibration médiocre . . . . .	196
5.4	Conclusion . . . . .	204
	<b>Conclusion et perspectives</b>	<b>205</b>
	<b>Annexes</b>	<b>211</b>
<b>A</b>	<b>Représentation des rotations</b>	<b>211</b>
A.1	Définitions et propriétés . . . . .	211
A.2	Choix de la représentation de la rotation . . . . .	213
A.3	Calcul de $\omega$ . . . . .	214
A.4	Calcul du jacobien . . . . .	216

<b>B</b>	<b>Analyse de la stabilité de l'asservissement visuel mono-caméra</b>	<b>219</b>
B.1	Définitions . . . . .	219
B.1.1	Conditions suffisantes pour la positivité d'une matrice . . . . .	220
B.1.2	Conditions suffisantes pour la positivité du produit de matrices positives	224
B.2	Asservissement visuel $2D_{\frac{1}{2}}$ . . . . .	227
B.2.1	Démonstration du théorème 1 . . . . .	227
B.2.2	Démonstration du théorème 2 . . . . .	227
B.2.3	Démonstration du théorème 3 . . . . .	232
B.2.4	Démonstration du théorème 4 . . . . .	233
B.2.5	Démonstration du théorème 5 . . . . .	235
B.2.6	Démonstration du théorème 6 . . . . .	236
B.3	Asservissement visuel $2D_{\frac{3}{4}}$ . . . . .	237
B.3.1	Démonstration du théorème 7 . . . . .	237
B.3.2	Démonstration du théorème 8 . . . . .	237
B.3.3	Démonstration du théorème 9 . . . . .	239
<b>C</b>	<b>Asservissement visuel avec contrôle de la translation de la caméra</b>	<b>241</b>
C.1	Modélisation . . . . .	241
C.2	Loi de commande . . . . .	242

<b>Bibliographie</b>	<b>244</b>
----------------------	------------

# Notations

## Règles générales

- minuscule: scalaire
- minuscule gras: vecteur
- majuscule gras: matrice

## Mathématiques

$\mathbb{P}^n$  : espace projectif à  $n$  dimensions

$\mathbb{R}^n$  : espace euclidien à  $n$  dimensions

$\mathbf{U}$  : matrice orthonormale par SVD

$\mathbf{S}$  : matrice des valeurs singulières par SVD

$\mathbf{V}$  : matrice orthonormale par SVD

$\sigma$  : valeur singulière

$\lambda$  : valeur propre

$[\mathbf{v}]_{\times}$  : matrice de préproduit vectoriel associé au vecteur  $\mathbf{v}$

## Robotique

$\boldsymbol{\tau}$  : vecteur des forces extérieures appliquées au robot

$\boldsymbol{\eta}$  : vecteur des forces centrifuges, de Coriolis, de gravité et de frottement

$\mathbf{f}_g$  : vecteur des forces de gravité

$\mathbf{f}_c$  : vecteur des forces centrifuges et de Coriolis

$\mathbf{f}_r$  : vecteur des forces de frottement

$\mathbf{r}$  : vecteur représentant la situation de l'effecteur du robot

$\mathbf{q}$  : vecteur des coordonnées articulaires du robot

$\mathbf{J}$  : matrice jacobienne du robot



## Asservissement visuel

**T** : matrice homogène de changement de repère cartésien  
**t** : vecteur de translation  
**R** : matrice de rotation  
**u** : axe de rotation  
 $\theta$  : angle de rotation  
**v** : vecteur vitesse dans l'espace cartésien (torseur cinématique)  
**v** : vecteur vitesse en translation  
 $\omega$  : vecteur vitesse en rotation  
**W** : matrice de changement de repère pour le torseur cinématique  
**H** : matrice hessienne  
**e** : fonction de tâche  
**s** : signal capteur (dans notre cas un point image)  
*H* : torseur d'interaction  
**L** : matrice d'interaction relative à la fonction de tâche  
**Q** : matrice du système en boucle fermée

## Reconstruction Projective et Euclidienne

$\mathcal{F}$  : repère cartésien  
 $\mathcal{O}$  : centre de projection  
 $\mathcal{P}$  : point dans l'espace euclidien  
**x** : vecteur contenant les coordonnées d'un point de l'espace euclidien  
 $\Pi$  : plan de projection ou plan image  
 $\pi$  : plan dans l'espace euclidien  
 $\pi$  : vecteur contenant les coordonnées homogènes d'un plan de l'espace euclidien  
**n** : normale à un plan  
*d* : distance du centre de projection à un plan  
**P** : matrice de projection perspective  
**A** : matrice de changement de repère mètre/pixels dans l'image  
*f* : focale de la caméra  
**m** : point image en coordonnées métriques ou normalisées  
**p** : point image en coordonnées pixels  
**E** : matrice essentielle  
**F** : matrice fondamentale  
**H** : matrice d'homographie dans l'espace euclidien  
**G** : matrice d'homographie dans l'espace projectif  
**e** : épipole  
**l** : droite épipolaire

# Introduction

Le contrôle des mouvements d'un robot en utilisant la vision par ordinateur, ou asservissement visuel, intéresse depuis quelques années beaucoup d'industriels. Par exemple, la Direction des Études et Recherche d'Électricité de France est en train de valider différentes techniques d'asservissement visuel afin de réaliser des opérations de maintenance dans les centrales nucléaires. Les applications pratiques considérées sont des tâches de positionnement contrôlant les six degrés de liberté de l'effecteur du robot afin de saisir des outils et les utiliser, positionner des objets et réaliser des tâches d'inspection. La grande généralité des techniques d'asservissement visuel devrait faciliter leur extension à des applications dans la robotique manufacturière, spatiale ou mobile.

Les capteurs visuels apportent aux systèmes robotiques une plus grande connaissance de l'environnement dans lequel ils évoluent. L'utilisation des caméras pourra peut être un jour (cependant encore lointain) rendre les robots complètement autonomes dans des milieux inconnus ou difficilement modélisables. En effet, les progrès récents effectués en vision par ordinateur, comme la reconstruction projective et euclidienne d'objets réels, leur reconnaissance et leur suivi, ont ouvert de nouvelles perspectives. Certes, l'utilisation de manière robuste de la plupart de ces techniques dans une boucle de commande à la cadence vidéo n'est pas encore possible. Toutefois, on peut dès maintenant s'interroger sur la manière d'intégrer une partie de ces innovations en vision par ordinateur dans un système robotique et sur les améliorations qu'il est possible d'obtenir en ce qui concerne les techniques d'asservissement visuel.

L'utilisation des techniques d'asservissement visuel sur un site industriel demande souvent à la fois une grande précision, une grande facilité de mise en œuvre, une grande portabilité et surtout une grande fiabilité. Ces quatre aspects pratiques ouvrent des problèmes théoriques très intéressants :

- afin d'obtenir une grande précision, il est nécessaire dans certaines applications, comme par exemple le positionnement d'objets très encombrants, d'utiliser plusieurs caméras. A cause de la grande taille de l'objet considéré, les caméras observent généralement des parties différentes de la scène. Il s'agit donc de résoudre le problème de la fusion des données provenant de plusieurs capteurs.
- afin d'obtenir une grande simplicité de mise en œuvre, il est souhaitable d'éliminer, ou simplifier, toutes les procédures de calibration du système, souvent trop lourdes et coûteuses (surtout lors de l'utilisation de plusieurs caméras).

- afin d’obtenir une grande portabilité des méthodes d’asservissement, il est souhaitable de concevoir un schéma de commande avec un besoin minimal de connaissances a priori sur les objets observés par les caméras. Le cas idéal serait de n’utiliser aucune connaissance du modèle de l’objet.
- afin d’obtenir une grande fiabilité tout en ayant éliminé la phase de calibration, les lois de commande doivent, autant que possible, présenter une grande robustesse aux incertitudes sur les paramètres du système et assurer la convergence à partir de n’importe quelle position initiale.

Bien que les méthodes d’asservissement visuel existantes soient, pour certaines applications, très performantes, beaucoup de problèmes théoriques sont encore ouverts en ce qui concerne les tâches de positionnement considérées ici. En effet, en l’état actuel, aucune technique d’asservissement visuel ne réunit à la fois toutes les caractéristiques évoquées.

Le premier objectif de notre travail a donc été de poser les bases de la conception d’une part, de techniques d’asservissement visuel qui ne nécessitent ni la connaissance du modèle de l’objet observé, ni une calibration précise du système, et d’autre part, de lois de commande robustes aux incertitudes sur les paramètres. Le deuxième objectif est l’extension de ces techniques d’asservissement visuel à l’utilisation de plusieurs caméras dans la boucle de commande.

Dans le cadre de notre étude, on s’intéressera au positionnement d’une ou plusieurs caméras, embarquées sur un robot manipulateur, par rapport à un objet dont on peut extraire des points caractéristiques. Puisque le suivi d’objets réels n’est pas encore possible à cadence élevée, on utilisera des objets marqués à partir desquels il est facile d’extraire des informations géométriques à la cadence vidéo. Nous espérons que l’extension des techniques de commande au cas d’objets réels sera quasi immédiate, aux problèmes d’erreurs de mesure près.

On supposera aussi que le problème de la mise en correspondance des points observés dans les images a été résolu. C’est une hypothèse nécessaire, car il est extrêmement difficile d’effectuer une mise en correspondance automatique sans aucune connaissance du modèle de l’objet observé et ce problème, loin d’être résolu, fait encore l’objet de nombreux travaux.

On verra que, sous certaines hypothèses vérifiées par la plupart des robots industriels, on peut considérer que la dynamique du robot n’intervient pas dans la boucle de commande, et on focalisera donc notre attention sur une commande de type cinématique.

La thèse est organisée de la manière suivante :

- dans le premier chapitre, nous donnons quelques définitions utiles avant de discuter l’état de l’art en asservissement visuel. Ensuite, nous décrivons en détail les deux techniques les plus couramment utilisées, c’est-à-dire l’asservissement visuel 2D (où le positionnement de la caméra est contrôlé dans l’image) et l’asservissement visuel 3D

(où le positionnement de la caméra est contrôlé dans l'espace cartésien).

- dans le deuxième chapitre, on étudie le problème de la reconstruction projective et euclidienne (à un facteur d'échelle près) à partir de deux images d'un objet statique dont le modèle est inconnu. Nous nous intéressons à des méthodes linéaires afin de disposer d'algorithmes compatibles avec la cadence vidéo. Nous mettons en évidence que la meilleure solution pour obtenir une reconstruction euclidienne robuste est de passer par la reconstruction projective d'une matrice d'homographie. Un algorithme pour l'estimation linéaire de cette matrice est alors proposé.
- dans le troisième chapitre, nous proposons deux nouvelles méthodes d'asservissement visuel basées sur la reconstruction euclidienne partielle décrite dans le chapitre précédent. Afin de réaliser une tâche de positionnement, on utilise tout d'abord une loi de commande proportionnelle. On étudie alors le système en boucle fermée afin de déterminer de manière analytique la robustesse de la loi de commande aux incertitudes sur les paramètres intrinsèques de la caméra et sur la calibration de la matrice de passage caméra/effecteur. Ensuite, on montre que, grâce à la forme découplée des systèmes considérés, il est possible d'augmenter considérablement le domaine de robustesse avec une adaptation des gains de la commande proportionnelle.
- dans le quatrième chapitre, nous présentons tout d'abord les résultats expérimentaux relatifs aux nouvelles méthodes d'asservissement visuel décrites dans le chapitre précédent. Ensuite, nous effectuons une comparaison expérimentale des techniques d'asservissement visuel les plus courantes et de celles que nous avons développées.
- dans le cinquième chapitre, nous étendons les schémas d'asservissement visuel à l'utilisation de plusieurs caméras n'observant pas la même partie de la scène. Nous modélisons l'asservissement visuel multi-caméras dans le cadre de la commande référencée capteur et nous étudions la stabilité du système. Nous appliquons enfin les considérations théoriques à une tâche de positionnement d'un objet encombrant en utilisant deux caméras embarquées sur un robot manipulateur.

Enfin, dans la conclusion, nous mettons en évidence les contributions principales de notre travail et nous présentons quelques perspectives de recherche.



# Chapitre 1

## État de l’art et modélisation

Dans ce chapitre, après avoir rappelé quelques notions de base en géométrie projective, nous décrivons le modèle géométrique d’une caméra et définissons les entités qui seront utilisées dans cette étude. Ensuite, nous présentons l’état de l’art en asservissement visuel avec un regard particulier en ce qui concerne les résultats obtenus par Claude Samson et al. [Samson 91] en commande des robots manipulateurs et en commande référencée capteur. En effet, leur approche générale, basée sur la régulation d’une *fonction de tâche*, peut être appliquée au cas particulier d’une commande basée sur la vision. Enfin, nous décrivons en détail les deux méthodes classiques d’asservissement visuel présentes dans la littérature (à savoir l’asservissement visuel 2D et 3D) et discutons de leurs avantages et inconvénients respectifs.

### 1.1 Modélisation

Les robots et les objets qu’ils manipulent évoluent dans l’espace cartésien (c’est-à-dire un espace euclidien à trois dimensions muni d’un repère orthonormé). Par contre, le modèle de la caméra qu’on utilisera par la suite est un modèle projectif. Les liens entre les espaces projectif et euclidien peuvent être établis en réalisant une représentation projective de ce dernier avec les coordonnées homogènes. Ces liens sont bien connus en robotique et en vision par ordinateur, car un changement de coordonnées cartésiennes peut être représenté par une matrice de transformation homogène.

#### 1.1.1 L’espace projectif comme extension de l’espace euclidien

Dans ce paragraphe, on rappelle quelques notions de géométrie projective (voir [Mohr 96] pour de plus amples détails). L’espace projectif  $\mathbb{P}^n$  peut être vu géométriquement comme une “extension” de l’espace euclidien  $\mathbb{R}^n$  car il permet la manipulation des entités géométriques “à l’infini”.

En effet, un point  $\mathcal{P}$  dans un espace euclidien à  $n$  dimensions, exprimé dans un repère donné par le vecteur  $\mathbf{x} = [ X_1 \ X_2 \ \cdots \ X_n ]^T \in \mathbb{R}^n$ , peut être décrit dans l’espace projectif  $\mathbb{P}^n$  en utilisant les coordonnées homogènes  $(\mathbf{x}, 1) = [ X_1 \ X_2 \ \cdots \ X_n \ 1 ]^T \in \mathbb{R}^{n+1}$ .

De plus, une direction  $[ X_1 \ X_2 \ \cdots \ X_n ]^T$  dans l'espace euclidien, qui correspond à un "point à l'infini", peut être représentée en utilisant les coordonnées homogènes par le point projectif  $(\mathbf{x}, 0) = [ X_1 \ X_2 \ \cdots \ X_n \ 0 ]^T$ .

Dans le cas général, un point dans l'espace projectif  $\mathbb{P}^n$  est donc représenté par un vecteur à  $n + 1$  composantes. Ses éléments sont définis à un facteur d'échelle près et au moins un d'entre eux est différent de zéro. Les points euclidiens peuvent être reconstruits à partir des points projectifs en divisant toutes les coordonnées par la dernière coordonnée projective, évidemment seulement si celle-ci n'est pas nulle.

Un hyper-plan dans l'espace euclidien est l'ensemble de tous les points de  $\mathbb{R}^n$  dont les coordonnées satisfont une équation linéaire. En utilisant les coordonnées homogènes, l'hyper-plan peut être représenté dans l'espace projectif par un vecteur non nul  $\boldsymbol{\pi}$  (de dimension  $n + 1$ ), défini à un facteur d'échelle près. Un hyper-plan très particulier dans l'espace euclidien est l'ensemble de tous les "points à l'infini", ce qui correspond à un idéal "hyper-plan à l'infini", et peut être représenté par  $\boldsymbol{\pi}_\infty = [ 0 \ 0 \ \cdots \ 0 \ 1 ]^T$ .

Tous les points dans  $\mathbb{P}^n$  appartenant à un hyper-plan satisfont l'équation linéaire  $\boldsymbol{\pi}^T(\mathbf{x}, 1) = 0$ . Cette équation présente une symétrie entre les coordonnées de l'hyper-plan  $\boldsymbol{\pi}$  et les coordonnées du point  $\mathcal{P}$  car  $\boldsymbol{\pi}^T(\mathbf{x}, 1) = (\mathbf{x}, 1)^T \boldsymbol{\pi}$ . En effet, l'espace formé par tous les hyper-plans peut être considéré comme un autre espace projectif appelé le dual de l'espace original  $\mathbb{P}^n$ . Un *principe de dualité* extrêmement important découle de cette symétrie [Mohr 96]:

**Principe de dualité:** *N'importe quelle propriété projective obtenue en utilisant des points et des hyper-plans possède un résultat symétrique, dans lequel les rôles des hyper-plans et des points sont inversés.*

N'importe quelle matrice  $(m + 1, n + 1)$  de rang plein peut être utilisée pour réaliser une application projective entre deux espaces  $\mathbb{P}^m$  et  $\mathbb{P}^n$ . Une application projective inversible de  $\mathbb{P}^n$  en lui-même est appelée une homographie et correspond à un changement de coordonnées projectives obtenu en utilisant une matrice de dimension  $(n + 1, n + 1)$ . Si on compare un changement de coordonnées dans les espaces projectif et euclidien, on peut voir que l'espace projectif est moins contraint que l'espace euclidien. En effet, un changement de coordonnées euclidiennes, réalisé en utilisant une matrice de transformation homogène, doit transformer l'hyper-plan à l'infini  $\boldsymbol{\pi}_\infty$  en lui-même.

La géométrie projective est un outil puissant qui permet d'obtenir des simplifications non seulement au niveau de la modélisation mais aussi au niveau calculatoire. Avant de modéliser le capteur caméra, on rappellera donc brièvement quelques notions de base, d'abord sur les matrices de transformation homogènes de dimension  $(4 \times 4)$  (qui réalisent une transformation de  $\mathbb{R}^3$  en  $\mathbb{R}^3$ ), puis sur les matrices de projection perspective de dimension  $(3 \times 4)$  (qui réalisent une transformation de  $\mathbb{R}^3$  en  $\mathbb{R}^2$ ).

### 1.1.2 Les matrices de transformation homogènes

Soit  $\mathcal{F}^*$  le repère de référence dans l'espace cartésien. On l'appellera aussi le repère désiré. Toutes les entités géométriques exprimées dans ce repère seront caractérisées par l'indice  $*$ . La matrice de transformation homogène entre  $\mathcal{F}^*$  et un deuxième repère  $\mathcal{F}$  (appelé le repère courant) est :

$${}^d\mathbf{T}_c = \begin{bmatrix} {}^d\mathbf{R}_c & {}^d\mathbf{t}_c \\ 0 & 1 \end{bmatrix} \quad (1.1)$$

où  ${}^d\mathbf{R}_c$  et  ${}^d\mathbf{t}_c$  sont respectivement la matrice de rotation et le vecteur de translation entre  $\mathcal{F}^*$  et  $\mathcal{F}$  (“d” à gauche et “c” à droite indiquent que le repère courant est exprimé dans le repère désiré). La transformation inverse est donnée par :

$${}^d\mathbf{T}_c^{-1} = \begin{bmatrix} {}^d\mathbf{R}_c^T & -{}^d\mathbf{R}_c^T {}^d\mathbf{t}_c \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{R}_d & {}^c\mathbf{t}_d \\ 0 & 1 \end{bmatrix} \quad (1.2)$$

Les matrices de transformation homogènes permettent d'exprimer le changement de coordonnées homogènes des points et des plans de l'espace cartésien de manière très simple :

- un point  $\mathbf{x}^*$  dans l'espace cartésien, exprimé dans le repère  $\mathcal{F}^*$ , est représenté par les coordonnées homogènes  $[X^* \ Y^* \ Z^* \ W^*]^T \in \mathbb{R}^4$ . Comme on l'a déjà vu, si la quatrième coordonnée homogène est nulle, le point est “à l'infini” (c'est-à-dire qu'il représente une direction dans l'espace cartésien). Autrement, on peut poser  $W^* = 1$ , sans perte de généralité, pour représenter un point fini. Ce point est exprimé dans le nouveau repère  $\mathcal{F}$  de la manière suivante :

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = {}^c\mathbf{T}_d \begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} \quad (1.3)$$

- un plan dans l'espace cartésien, exprimé dans le repère  $\mathcal{F}^*$ , est caractérisé par son vecteur normal  $\mathbf{n}^*$  et sa distance  $d^*$  à l'origine  $\mathcal{O}^*$  du repère de référence. En utilisant les coordonnées homogènes, il est représenté par le vecteur  $\boldsymbol{\pi}^* = [\mathbf{n}^{*T} \ -d^*]^T$ . Le même plan, noté  $\boldsymbol{\pi} = [\mathbf{n}^T \ -d]^T$  dans le repère  $\mathcal{F}$ , est exprimé de la manière suivante :

$$\boldsymbol{\pi} = {}^c\mathbf{T}_d^{-T} \boldsymbol{\pi}^* \quad \Rightarrow \quad \begin{bmatrix} \mathbf{n} \\ -d \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{R}_d \mathbf{n}^* \\ -d^* - {}^c\mathbf{t}_d^T {}^c\mathbf{R}_d \mathbf{n}^* \end{bmatrix} \quad (1.4)$$

où la matrice  ${}^c\mathbf{T}_d^{-T}$  est appelée la matrice *duale* de  ${}^c\mathbf{T}_d$ .

Une matrice de transformation homogène permet aussi de représenter la situation d'un solide dans l'espace cartésien par rapport à un repère de référence. Toutefois, une telle représentation est redondante car seulement trois variables indépendantes suffisent pour identifier une matrice de rotation dans le groupe des rotations  $SO_3$ . C'est pourquoi la situation d'un solide, comme par exemple l'effecteur d'un robot, est représentée par un vecteur  $\mathbf{r} \in SE_3$  (où  $SE_3$  est le “Special Euclidean Group” isomorphe à  $\mathbb{R}^3 \times SO_3$ ) contenant trois paramètres pour la translation et trois paramètres représentant la rotation. Différentes représentations sont possibles, comme par exemple les angles d'Euler ou les quaternions. Par la suite, on utilisera  $\mathbf{u} \theta$ , c'est-à-dire la multiplication de l'axe de rotation  $\mathbf{u}$  par l'angle de rotation  $\theta$  autour de cet axe.



### 1.1.3 Les matrices de projection perspective

Soit le *centre de projection*  $\mathcal{O}^*$  l'origine du repère absolu. Le plan de projection  $\Pi$  est parallèle au plan  $(\vec{x}, \vec{y})$  et la distance  $f$  entre  $\mathcal{O}^*$  et  $\Pi$  est appelée la *distance focale*. L'intersection  $\mathbf{m}_0^*$  de l'axe  $\vec{z}$  et du plan image (voir Figure 1.1) est appelée le *point principal*.

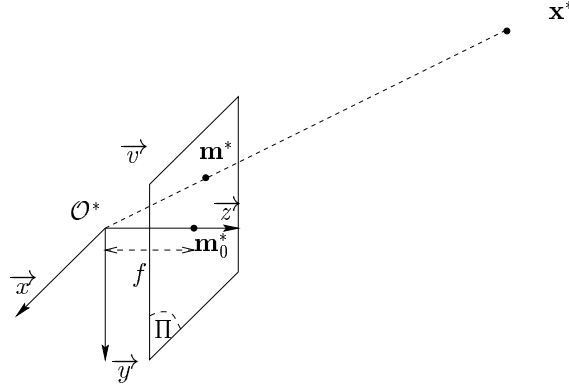


FIG. 1.1 – *Projection perspective*

La projection perspective d'un point  $\mathbf{x}^* \in \mathbb{P}^3$  en un point  $\mathbf{m}^* = [f \ x^* \ f \ y^* \ f]^T \in \mathbb{P}^2$  dans le plan de projection  $\Pi$  peut s'écrire :

$$\frac{Z^*}{f} \mathbf{m}^* = \mathbf{P}^* \begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} \quad (1.5)$$

où  $\mathbf{P}^* = [ \mathbf{I}_3 \ \mathbf{0} ]$  est la matrice de projection de dimension  $(3 \times 4)$  et  $Z^*$  est la profondeur du point dans le repère  $\mathcal{F}^*$ .

Comme on l'a vu précédemment, un changement de repère dans l'espace cartésien est représenté par une matrice de transformation homogène. Après un déplacement de la caméra de  $\mathcal{F}^*$  en  $\mathcal{F}$ , le point de coordonnées  $\mathbf{x}^*$  dans  $\mathcal{F}^*$  (et  $\mathbf{x}$  dans  $\mathcal{F}$ ) est donc projeté au point de coordonnées  $\mathbf{m} = [f \ x \ f \ y \ f]^T \in \mathbb{P}^2$  suivant la relation :

$$\frac{Z}{f} \mathbf{m} = \mathbf{P}^* \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \mathbf{P}^* {}^c\mathbf{T}_d \begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} \quad (1.6)$$

où  $\mathbf{P} = [ {}^c\mathbf{R}_d \ {}^c\mathbf{t}_d ]$  est la matrice de projection  $(3 \times 4)$  exprimée dans le nouveau repère et  $Z$  est la profondeur du point dans le repère  $\mathcal{F}$ . On peut fixer  $f = 1$ , sans perte de généralité car, de toute façon, les coordonnées d'un point dans un espace projectif sont définies à un facteur d'échelle près. On va voir maintenant que les matrices de projection perspective sont utilisées afin de modéliser les caméras avec un modèle sténopé.

### 1.1.4 Modélisation de la caméra

La caméra, décrite en utilisant le modèle sténopé (pour plus de détails voir par exemple [Horaud 93] ou [Faugeras 93]), réalise une projection perspective d'un point  $\mathbf{x} \in \mathbb{P}^3$  dans un

point image  $\mathbf{p} \in \mathbb{P}^2$  de coordonnées homogènes  $\mathbf{p} = [u \ v \ 1]^T$  exprimées en pixels (voir [Wiles 96] pour des modèles de caméra différents). Comme on l'a vu précédemment, le point  $\mathbf{x}$  est projeté en un point de coordonnées métriques homogènes  $\mathbf{m} = [x \ y \ 1]^T$ .

La transformation projective entre les coordonnées métriques et les coordonnées pixels d'un point image est :

$$\mathbf{p} = \mathbf{A} \mathbf{m} \quad (1.7)$$

où  $\mathbf{A}$  est la matrice des paramètres intrinsèques de la caméra qui réalise le changement des coordonnées projectives dans  $\mathbb{P}^2$  :

$$\mathbf{A} = \begin{bmatrix} f k_u & -f k_u \cot(\phi) & u_0 \\ 0 & f k_v / \sin(\phi) & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & \alpha_{uv} & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.8)$$

où  $u_0$  et  $v_0$  sont les coordonnées en pixels du point principal,  $k_u$  et  $k_v$  sont les facteurs d'échelle suivant l'axe  $\vec{u}$  et l'axe  $\vec{v}$  (en pixels/mètres), et  $\phi$  est l'angle entre ces axes (voir Figure 1.2).

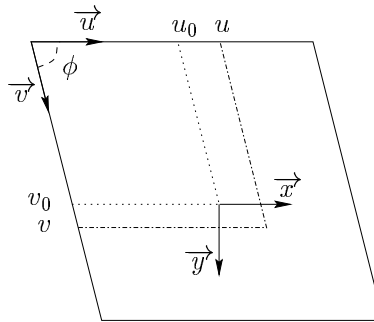


FIG. 1.2 – Transformation mètres pixels

En appliquant le changement de coordonnées projectives, à partir de l'équation (1.6), on obtient :

$$\frac{Z}{f} \mathbf{p} = \mathbf{A} \mathbf{P}^* \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \quad (1.9)$$

La relation entre les coordonnées homogènes de  $\mathbf{x}$  et  $\mathbf{p}$  est linéaire car la caméra est supposée être sans distorsion (voir [Zhang 96] pour un modèle caméra avec distorsion). Si les paramètres intrinsèques de la caméra sont parfaitement connus, ce qui nécessite une calibration de la caméra très précise [Robert 93], on peut travailler indifféremment avec les coordonnées normalisées ou exprimées en pixels. Une hypothèse plus réaliste en pratique est de considérer que les paramètres intrinsèques de la caméra sont grossièrement connus. On utilisera donc des méthodes de reconstruction projective afin de concevoir des schémas d'asservissement robustes aux incertitudes sur ces paramètres. Il faut toutefois souligner que, comme on le verra dans le chapitre suivant, une normalisation des données est nécessaire quand on désire effectuer une reconstruction projective à partir des points exprimés en coordonnées pixels [Hartley 97].

## 1.2 L'asservissement visuel

Le principe de l'asservissement visuel consiste à prendre en compte des informations visuelles issues d'une ou plusieurs caméras dans la boucle de commande d'un robot afin d'en contrôler le mouvement. Les techniques d'asservissement visuel utilisent la perception afin d'améliorer l'action et augmentent donc la flexibilité et la précision des systèmes robotiques [Hashimoto 93a]. La boucle perception-action est complétée par la vision active [Aloimonos 87] [Bajcsy 88] où le mouvement du robot est contrôlé afin d'améliorer la perception (par exemple dans la réalisation d'une tâche d'insertion [Yoshimi 94], dans la reconstruction d'une scène [Marchand 96] ou dans la calibration du système de vision [Heraud 97]). Les systèmes de commande utilisant la vision sont généralement classifiés, tout d'abord, suivant le niveau auquel la vision intervient dans la structure de la boucle de contrôle du robot [Sanderson 80] :

- Une première structure de contrôle peut être appelée asservissement visuel bas niveau ("direct visual servo" en anglais). Dans cette structure, illustrée en Figure 1.3(a), le système de vision fournit directement une estimation de l'état du robot et se substitue au contrôleur bas niveau du robot [Sanderson 83]. Il est évident qu'il est alors nécessaire de fournir une estimation de l'état à une cadence très élevée. Une étude en simulation de l'influence de différentes perturbations sur l'estimation temps réel de l'état a été effectuée en [Bishop 94]. Dans le cas où les articulations du robot sont observées directement par la caméra, cette structure de commande a été utilisée pour des robots planaires avec deux [Wijesoma 93] ou trois axes [Weiss 87]. Dans le cas où la caméra est embarquée et observe un point [Kelly 97] ou déportée et observe l'effecteur du robot [Kelly 96], cette structure de commande a été utilisée pour un robot planaire à deux axes. Font exception, enfin, les travaux décrits en [Hashimoto 93b] et [Gangloff 98] sur le contrôle des six axes d'un robot avec une caméra embarquée observant une cible de plusieurs points.
- Une structure de contrôle, beaucoup plus répandue dans la littérature [Corke 93] [Hutchinson 96], utilise les techniques d'asservissement visuel dans une boucle haut niveau qui tourne à une fréquence plus lente (la cadence vidéo est de 25 Hz) que celle du contrôleur bas niveau du robot (de l'ordre de 100 Hz). Dans cette structure de contrôle, illustrée en Figure 1.3(b), qu'on peut donc qualifier de haut niveau, le système de vision fournit un état de consigne au contrôleur du robot qui le réalise. Cette méthode présente des avantages certains de simplicité, de portabilité et de robustesse, car le robot est considéré comme un outil de positionnement indépendant du système de vision (par exemple, on sépare les singularités du robot du contrôleur visuel). Les performances du contrôleur visuel dépendent alors seulement de sa conception et du retard introduit par le système de vision dans la boucle, car la réalisation des consignes est assurée par le contrôleur bas niveau. Une étude de l'influence de ce retard sur le système en boucle fermée a été effectuée en [Corke 96], et plus récemment, une relation entre le retard et l'erreur de poursuite d'un système bouclé avec une commande PID a été établie dans [Krautgartner 98].

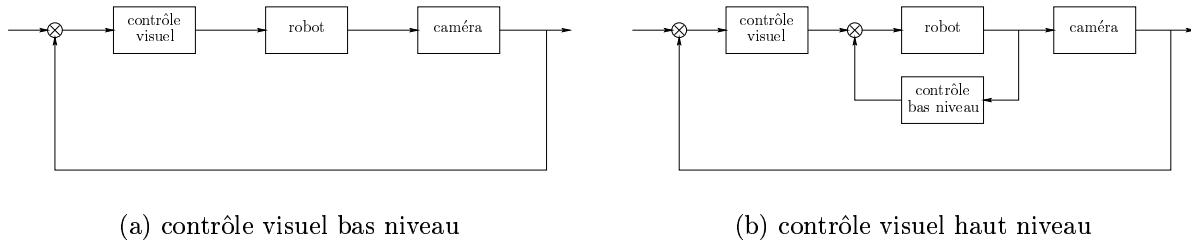


FIG. 1.3 – Structures d'asservissement visuel

Une deuxième classification des systèmes d'asservissement visuel est effectuée suivant l'espace d'état dans lequel s'effectue le contrôle du robot [Sanderson 80]. Par exemple, on peut éventuellement observer directement la configuration articulaire du robot, mais aussi estimer la position de l'effecteur du robot dans l'espace cartésien, ou utiliser simplement les images de l'objet par rapport auquel on désire positionner le robot. Cette deuxième classification sera mieux détaillée dans le cadre très général de la commande référencée capteur basée sur le principe de la commande par fonction de tâche.

### 1.2.1 Le principe de la commande par fonction de tâche

On peut montrer [Samson 91] qu'une tâche robotique peut généralement être exprimée comme la régulation, sur un certain horizon temporel  $[0, T]$ , d'une fonction  $\mathbf{e}(\mathbf{q}, t)$  (où  $\mathbf{q}$  est le vecteur contenant la configuration articulaire du robot) de classe  $C^2$  et de dimension  $m$ , appelée *fonction de tâche*. Des fonctions de tâche classiques sont :

- $\mathbf{e}(\mathbf{q}, t) = \mathbf{q} - \mathbf{q}^*(t)$  où  $\mathbf{q}^*(t)$  est une trajectoire désirée dans l'espace articulaire,
- $\mathbf{e}(\mathbf{q}, t) = \mathbf{r}(\mathbf{q}) - \mathbf{r}^*(t)$  où  $\mathbf{r}^*(t)$  est une trajectoire désirée dans l'espace cartésien,
- $\mathbf{e}(\mathbf{q}, t) = \bar{\mathbf{e}}(\mathbf{s}(\mathbf{q}), \mathbf{s}^*(t))$  où  $\mathbf{s}^*(t)$  est une trajectoire désirée dans l'espace du capteur.

Ce dernier type de fonction de tâche nous intéresse particulièrement dans le cas où les informations capteur  $\mathbf{s}(\mathbf{q}, t)$  utilisées pour appréhender la valeur de la fonction de tâche à l'instant  $t$  sont issues d'une caméra. De manière générale, comme cela est précisé en [Samson 91], le problème de la régulation de  $\mathbf{e}$  est bien posé si la fonction de tâche possède certaines propriétés. L'une d'elles, qui signifie que la tâche est réalisable, est l'existence et l'unicité d'une *trajectoire idéale* de classe  $C^2$ , notée  $\mathbf{q}_r(t)$ , telle que  $\mathbf{e}(\mathbf{q}_r(t), t) = 0, \forall t \in [0, T]$  et  $\mathbf{q}_r(0) = \mathbf{q}_0$ , où  $\mathbf{q}_0$  est une condition initiale donnée. Une autre condition, extrêmement importante, est la *régularité* du jacobien de la tâche,  $\frac{\partial \mathbf{e}}{\partial \mathbf{q}}(\mathbf{q}, t)$ , autour de  $\mathbf{q}_r(t)$ . En effet, ce jacobien lie l'espace différentiel articulaire à l'espace différentiel de la fonction de tâche :

$$\dot{\mathbf{e}}(\mathbf{q}, t) = \frac{\partial \mathbf{e}}{\partial \mathbf{q}}(\mathbf{q}, t) \dot{\mathbf{q}} + \frac{\partial \mathbf{e}}{\partial t}(\mathbf{q}, t) \quad (1.10)$$

et

$$\ddot{\mathbf{e}}(\mathbf{q}, t) = \frac{\partial \mathbf{e}}{\partial \mathbf{q}}(\mathbf{q}, t) \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (1.11)$$

avec :

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, t) = \begin{bmatrix} \vdots \\ \dot{\mathbf{q}}^T \mathbf{H}_i(\mathbf{q}, t) \dot{\mathbf{q}} \\ \vdots \end{bmatrix} + 2 \frac{\partial^2 \mathbf{e}}{\partial \mathbf{q} \partial t}(\mathbf{q}, t) \dot{\mathbf{q}} + \frac{\partial^2 \mathbf{e}}{\partial t^2}(\mathbf{q}, t) \quad (1.12)$$

où  $\mathbf{H}_i (i = 1, \dots, n)$  est la  $i$ -ème matrice hessienne de  $\mathbf{e}$  par rapport à  $\mathbf{q}$ . Lorsque toutes les conditions requises sont satisfaites, la fonction de tâche est dite *admissible*, ce qui permet alors la synthèse de lois de commande efficaces, dont la robustesse sera liée à la plus ou moins grande *admissibilité* de la tâche.

Dans notre cas, les informations capteur sont des informations visuelles. Il est alors évident qu'une autre condition, extrêmement importante à respecter, est que les informations visuelles nécessaires à la réalisation de la tâche soient toujours présentes dans l'image (condition de *visibilité* de la tâche). Soient  $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots\}$  l'ensemble des informations visuelles élémentaires  $\mathbf{s}_i$  théoriquement disponibles,  $s_{min}$  le nombre minimal d'informations visuelles nécessaires à la réalisation de la fonction de tâche et  $\mathcal{S}_{obs} = \{\mathbf{s}_i : \mathbf{s}_i \in \mathcal{S}, \mathbf{s}_i \in \mathcal{I}\}$  l'ensemble des informations visuelles réellement observées ( $\mathcal{I}$  étant le domaine de définition des informations visuelles considérées, c'est-à-dire l'image). La condition de visibilité peut alors s'écrire :

$$\dim(\mathcal{S}_{obs}) \geq s_{min}, \quad \forall t \quad (1.13)$$

Comme on le verra par la suite, cette condition n'est généralement pas prise directement en compte dans la commande. Ceci implique que, suivant le type de fonction de tâche choisi, on aura une probabilité plus ou moins grande que les informations visuelles restent dans le champ de vision de la caméra.

La configuration de la caméra par rapport au robot peut aussi avoir une influence sur la visibilité de la tâche. En effet, si la caméra est embarquée comme en Figure 1.4(a), le robot est contrôlé en observant l'objet sur lequel on désire le positionner, alors que, si la caméra est déportée comme en Figure 1.4(b), on peut contrôler les mouvements du robot en observant sa configuration articulaire, son effecteur (ce type de contrôle est appelé "Endpoint Open Loop" en anglais), ou l'effecteur et l'objet en même temps (ce type de contrôle est appelé "Endpoint Closed Loop" en anglais). De manière générale, les systèmes "Endpoint Open Loop" sont moins contraignants du point de vue du temps de calcul car le problème de vision est simplifié par le fait qu'un seul objet doit être suivi dans l'image, mais, d'un autre côté, ces systèmes n'ont pas de contrôle direct sur l'objet, dont la position doit alors être parfaitement connue par rapport à la caméra. Les différentes techniques d'asservissement visuel qu'on décrira par la suite s'appliquent indifféremment aux deux configurations.

Comme on l'a déjà indiqué, la deuxième classification des techniques d'asservissement visuel peut être faite sur la base du choix de la fonction de tâche. En effet, on peut par exemple construire la fonction de tâche des manières suivantes :

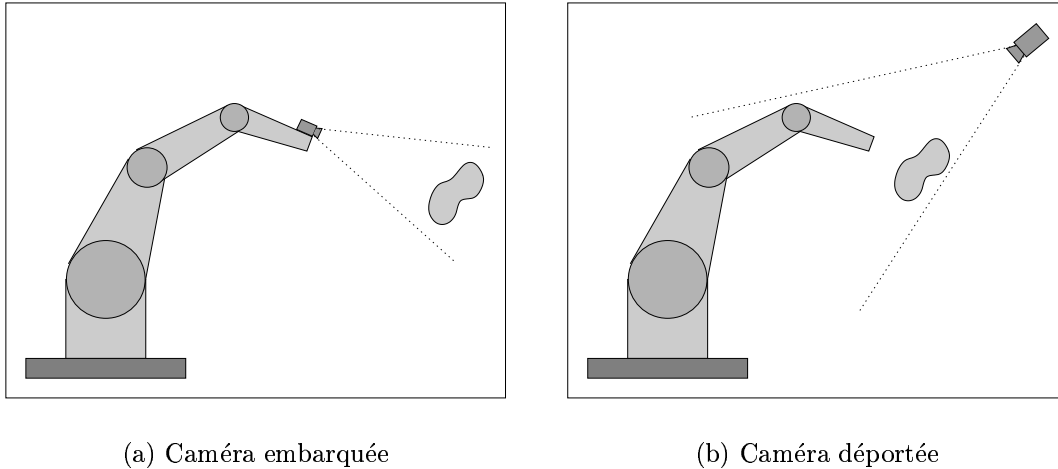


FIG. 1.4 – *Asservissement visuel mono-caméra*

- $\mathbf{e}(\mathbf{q}) = \mathbf{r}(\mathbf{s}(\mathbf{q})) - \mathbf{r}(\mathbf{s}^*(t))$ . La position  $\mathbf{r}$  de l'effecteur du robot est estimée à partir des informations visuelles. Cette approche est appelée *asservissement visuel 3D* car le contrôle de la caméra s'effectue dans l'espace cartésien [Wilson 96] [Grosso 96] [Martinet 97] [Tonko 97].
- $\mathbf{e}(\mathbf{q}, t) = \mathbf{C}(\mathbf{s}(\mathbf{q}) - \mathbf{s}^*(t))$  où  $\mathbf{C}$  est une matrice qui permet de tenir compte d'une éventuelle redondance d'information. La fonction de tâche est construite à partir de la différence entre les informations visuelles courantes et désirées. Cette approche est appelée *asservissement visuel 2D* car le contrôle s'effectue dans l'image [Chaumette 90] [Espiau 92] [Hashimoto 93a] [Papanikolopoulos 93a]. Cette méthode a été appliquée à une caméra déportée par [Dornaika 95].
- $\mathbf{e}(\mathbf{q}, t) = \bar{\mathbf{q}}(\mathbf{s}(\mathbf{q}) - \bar{\mathbf{q}}^*(\mathbf{s}^*(t)))$ . La configuration articulaire du robot est observée par une caméra déportée comme en Figure 1.4(b). On parle dans ce cas d'asservissement visuel dans l'espace articulaire. La caméra qui observe le robot dispose généralement d'un grand champ de vision pour que toutes les articulations du robot soient visibles. Cette méthode a été utilisée pour un robot à un axe par [Skaar 87] et étendue au cas plus général d'un robot six axes [Skaar 90] [Chen 94] mais en utilisant au moins deux caméras.

Les deux premières méthodes sont les plus répandues dans la littérature et seront détaillées davantage par la suite. Le paragraphe 1.3 est dédié à la description de l'asservissement visuel 3D, tandis que le paragraphe 1.4 à celle de l'asservissement visuel 2D.

Une des difficultés majeures pour la construction de la fonction de tâche est la définition de la trajectoire désirée  $\mathbf{s}^*(t)$ . En effet, il est nécessaire de construire la fonction de tâche de telle manière qu'elle soit nulle si les informations visuelles courantes coïncident avec les informations visuelles de référence :  $\mathbf{e}(\mathbf{q}, t) = 0$  si  $\mathbf{s}(\mathbf{q}(t)) = \mathbf{s}^*(t)$ . Une première méthode pour définir les informations désirées est basée sur leur modélisation en utilisant des connaissances a priori sur la scène observée. Cette méthode demande, entre autres, une connaissance parfaite de la géométrie du système de vision pour que  $\mathbf{s}^*(t)$  puisse physiquement exister. Une

technique pour éviter cette phase de modélisation consiste à effectuer un apprentissage expérimental de l'image de référence. Dans ce cas, on a alors  $\mathbf{s}^*(t) = \mathbf{s}^*$ , ce qui permet d'éviter la difficulté de générer une trajectoire  $\mathbf{s}^*(t)$ . Considérons, par exemple, la tâche classique du positionnement par rapport à un objet d'une caméra embarquée sur un robot. Après avoir appris l'image correspondant à la position désirée de la caméra, et après avoir déplacé la cible et/ou le robot, on peut définir la fonction de tâche sur la base des informations obtenues à partir des deux vues de la cible. Une fonction de tâche nulle implique alors que la caméra a rejoint sa position désirée avec une précision qui est indépendante des erreurs de calibration du système de vision. Bien que, de manière générale, les informations visuelles désirées  $\mathbf{s}^*(t)$  puissent varier dans le temps [Feddema 89] [Berry 97], on supposera par la suite que la configuration à atteindre dans l'image est fixe ( $\frac{\partial \mathbf{s}^*}{\partial t} = 0$ ) et qu'elle a été déterminée lors d'une phase d'apprentissage. Nous reviendrons tout de même dans le chapitre 3 sur les aspects de génération de trajectoire  $\mathbf{s}^*(t)$  afin de discuter de leur intérêt sur la stabilité du système.

Si on peut assurer qu'il existe  $\mathbf{q}^*$  tel que  $\mathbf{e}(\mathbf{q}^*) = \mathbf{C}(\mathbf{s}(\mathbf{q}^*) - \mathbf{s}^*) = 0$ , on peut donc écrire (en utilisant la notation des *multi-indices* introduite par H. Whitney) le développement de Taylor de la fonction de tâche ( $\mathbf{e}(\mathbf{q}) = \mathbf{C}(\mathbf{s}(\mathbf{q}) - \mathbf{s}^*) = \bar{\mathbf{e}}(\mathbf{s}(\mathbf{q}))$ ) de la manière suivante (où  $\mathbf{R}(\mathbf{s})$  ( $\mathbf{s} - \mathbf{s}^*$ ) contient les termes du développement supérieurs aux dérivées secondes) :

$$\begin{aligned} \bar{\mathbf{e}}(\mathbf{s}) &= \bar{\mathbf{e}}(\mathbf{s}^*) + \left. \frac{\partial \bar{\mathbf{e}}}{\partial \mathbf{s}} \right|_{\mathbf{s}=\mathbf{s}^*} (\mathbf{s} - \mathbf{s}^*) + \frac{1}{2} (\mathbf{s} - \mathbf{s}^*)^T \left. \frac{\partial^2 \bar{\mathbf{e}}}{\partial \mathbf{s}^2} \right|_{\mathbf{s}=\mathbf{s}^*} (\mathbf{s} - \mathbf{s}^*) + \mathbf{R}(\mathbf{s}) (\mathbf{s} - \mathbf{s}^*) = \\ &= \left[ \left. \frac{\partial \bar{\mathbf{e}}}{\partial \mathbf{s}} \right|_{\mathbf{s}=\mathbf{s}^*} + \frac{1}{2} (\mathbf{s} - \mathbf{s}^*)^T \left. \frac{\partial^2 \bar{\mathbf{e}}}{\partial \mathbf{s}^2} \right|_{\mathbf{s}=\mathbf{s}^*} + \mathbf{R}(\mathbf{s}) \right] (\mathbf{s} - \mathbf{s}^*) = \mathbf{C}(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \mathbf{g}) (\mathbf{s} - \mathbf{s}^*) \quad (1.14) \end{aligned}$$

où  $\mathbf{C}$  est une matrice qui dépend généralement des informations visuelles courantes  $\mathbf{s}$  et désirées  $\mathbf{s}^*$ , de la matrice  $\mathbf{A}$  contenant les paramètres intrinsèques de la caméra, et d'un vecteur  $\mathbf{g}$  contenant les paramètres géométriques relatifs à la scène observée. Comme on le verra par la suite, un des plus grands problèmes pour l'analyse de la stabilité des systèmes d'asservissement visuel est la difficulté de modéliser ou même d'obtenir une forme analytique de la matrice  $\mathbf{C}$ . Ce problème est crucial en asservissement visuel 3D, alors qu'on ne le rencontre pas en asservissement visuel 2D.

Une fois que la fonction de tâche a été définie, on doit concevoir une loi de commande afin de la réguler à zéro. Cette loi de commande doit présenter des caractéristiques suffisantes de robustesse afin d'assurer la convergence de l'asservissement visuel en présence d'erreurs de mesure et de calibration de la géométrie du système. Même si on a vu que la précision de positionnement peut être rendue indépendante des erreurs de calibration par des techniques d'apprentissage, ces erreurs ont une grande influence sur la manière dont le système arrive à la convergence. Pour cette raison, il sera important d'étudier la robustesse de l'asservissement visuel par rapport aux erreurs de calibration. On va voir maintenant quelles solutions ont été proposées en littérature afin de contrôler les mouvements d'un robot par la régulation à zéro d'une fonction de tâche.

## 1.2.2 Commande dynamique

Il est bien connu que le comportement dynamique d'un robot rigide est décrit par l'équation :

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) \quad (1.15)$$

où  $\mathbf{q}$  est le vecteur articulaire du robot,  $\boldsymbol{\tau}$  est le vecteur des forces extérieures appliquées,  $\mathbf{M}$  la matrice  $n \times n$  ( $n = \dim(\mathbf{q})$ ) d'énergie cinétique, et  $\mathbf{f} = \mathbf{f}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}_g(\mathbf{q}) + \mathbf{f}_r(\mathbf{q}, t)$  est le vecteur qui rassemble les contributions des forces de gravité  $\mathbf{f}_g(\mathbf{q})$ , centrifuges et de Coriolis  $\mathbf{f}_c(\mathbf{q}, \dot{\mathbf{q}})$ , et de frottement  $\mathbf{f}_r(\mathbf{q}, t)$ . La dérivée temporelle de la matrice d'inertie et le vecteur des forces de Coriolis vérifient la relation :

$$\frac{1}{2}\dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{q}}^T \mathbf{f}_c(\mathbf{q}, \dot{\mathbf{q}}) \quad \forall \mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n \quad (1.16)$$

Soit  $(\mathbf{q}, \dot{\mathbf{q}})$  le vecteur d'état, l'équation *d'état* du système est donc :

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{M}^{-1}(\mathbf{q}) \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{M}^{-1}(\mathbf{q}) \end{bmatrix} \boldsymbol{\tau} \quad (1.17)$$

où la variable de contrôle est le vecteur  $\boldsymbol{\tau}$ . À partir des équations (1.10) et (1.11), on peut écrire l'équation d'état dans l'espace de la fonction de tâche :

$$\frac{d}{dt} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \mathbf{M}^{-1}(\mathbf{q}) \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{h} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \mathbf{M}^{-1}(\mathbf{q}) \end{bmatrix} \boldsymbol{\tau} \quad (1.18)$$

où  $\mathbf{h}$  est définie par l'équation (1.12). Différentes méthodes ont été proposées afin de contrôler ce système non-linéaire. Une première technique, dite de linéarisation entrée sortie par retour d'état statique [Isidori 95], effectue tout d'abord la linéarisation de (1.18) dans le cas idéal :

$$\boldsymbol{\tau} = \mathbf{M} \left( \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \right)^{-1} (\mathbf{u} - \mathbf{h}) + \mathbf{f} \quad (1.19)$$

où  $\mathbf{u}(= \ddot{\mathbf{e}})$  est le nouveau vecteur de commande. La linéarisation est effectuée à la fois au niveau de la dynamique du robot ( $\mathbf{M}$  et  $\mathbf{f}$ ) et à la fois au niveau de la tâche ( $\frac{\partial \mathbf{e}}{\partial \mathbf{q}}$  et  $\mathbf{I}$ ). Le contrôle du système linéarisé peut ensuite s'effectuer avec un retour proportionnel dérivé de forme générale [Samson 91] :

$$\mathbf{u} = -k \mathbf{G} (\dot{\mathbf{e}} + \lambda \mathbf{D} \mathbf{e}) \quad (1.20)$$

où  $\mathbf{G}$  et  $\mathbf{D}$  sont deux matrices positives,  $k$  et  $\lambda$  deux scalaires positifs. À partir des équations (1.19) et (1.20), on obtient finalement une forme de commande linéarisante dans l'espace de la tâche :

$$\boldsymbol{\tau} = -k \mathbf{M} \left( \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \right)^{-1} \mathbf{G} (\lambda \mathbf{D} \mathbf{e} + \dot{\mathbf{e}}) + \mathbf{f} - \mathbf{M} \left( \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \right)^{-1} \mathbf{h} \quad (1.21)$$



Naturellement, cette commande idéale suppose une connaissance parfaite de tous les termes la constituant, ce qui n'est ni possible, ni même souvent souhaité. On considérera donc finalement de façon plus réaliste le schéma de commande suivant, généralisation du précédent :

$$\boldsymbol{\tau} = -k \widehat{\mathbf{M}} \left( \frac{\partial \widehat{\mathbf{e}}}{\partial \mathbf{q}} \right)^{-1} \mathbf{G} \left( \lambda \mathbf{D} \widehat{\mathbf{e}} + \frac{\partial \widehat{\mathbf{e}}}{\partial \mathbf{q}} \widehat{\dot{\mathbf{q}}} + \frac{\partial \widehat{\mathbf{e}}}{\partial t} \right) + \widehat{\mathbf{f}} - \widehat{\mathbf{M}} \left( \frac{\partial \widehat{\mathbf{e}}}{\partial \mathbf{q}} \right)^{-1} \widehat{\mathbf{h}} \quad (1.22)$$

dans lesquels les “chapeaux” indiquent que l'on utilise des mesures (approximations, estimations, ...) des grandeurs considérées. La commande (1.22) englobe de nombreux schémas existants : commande dynamique, commande en vitesse, commande adaptative indirecte, etc.

Cette méthode de contrôle a été appliquée au cas d'une caméra embarquée sur un robot à six degrés de liberté par [Hashimoto 93b]. La fonction de tâche est construite directement dans l'image en observant une cible de quatre points.

Une analyse de la stabilité du système (1.15) rebouclé par la commande (1.22), avec l'hypothèse d'un bruit additif sur la fonction de tâche  $\widehat{\mathbf{e}} = \mathbf{e} + \mathbf{b}_e(t)$ , a été réalisée par Samson et al. [Samson 91] dans un cadre non-linéaire, et a permis d'exhiber des conditions suffisantes de stabilité (bornitude de  $\|\mathbf{e}(\mathbf{q}, t)\|$  et, de plus, comportement asymptotique), qui sont principalement de deux classes :

- des conditions portant sur les **gains** de l'asservissement. En particulier, si le gain  $k$  est choisi suffisamment grand, la dynamique du système peut être négligée comme on le verra dans la section suivante.
- des conditions portant sur les **modèles** utilisés. Les plus importantes concernent la régularité du jacobien de la tâche. D'autres, liées à la dynamique du robot, ne sont pas trop contraignantes (en particulier en raison du fait que la matrice  $\mathbf{M}(\mathbf{q})$  est symétrique et définie positive).

L'analyse de stabilité effectuée étant de caractère général, la modélisation explicite des erreurs sur les paramètres intrinsèques de la caméra n'a pas été prise en compte. On ne dispose donc pas de résultats analytiques sur l'étendue du domaine de stabilité du système en fonction des erreurs de calibration.

Une autre méthode de commande est basée sur la technique du jacobien transposé proposée par [Takegaki 81] (et qui peut être obtenue à partir de l'équation (1.22) en posant  $\mathbf{G} = \mathbf{I}$ ,  $\left( \frac{\partial \widehat{\mathbf{e}}}{\partial \mathbf{q}} \right)^{-1} = \widehat{\mathbf{M}}^{-1} \left( \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \right)^T$ ,  $k \widehat{\mathbf{M}} = \mathbf{K}_v$ ,  $k \lambda \mathbf{D} = \mathbf{K}_p$ ,  $\widehat{\mathbf{f}} = \mathbf{f}_g(\mathbf{q}) + \mathbf{f}_r(\mathbf{q}, t)$  et  $\frac{\partial \widehat{\mathbf{e}}}{\partial t} = \widehat{\mathbf{h}} = 0$ ) :

$$\boldsymbol{\tau} = - \left( \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \right)^T \mathbf{K}_p \mathbf{e} - \mathbf{K}_v \dot{\mathbf{q}} + \mathbf{f}_g(\mathbf{q}) + \mathbf{f}_r(\mathbf{q}, t) \quad (1.23)$$

où  $\mathbf{K}_p$  et  $\mathbf{K}_v$  sont deux matrices symétriques et positives, et où seule la linéarisation des forces de gravité  $\mathbf{f}_g$  et de frottement  $\mathbf{f}_r$  est nécessaire. En effet, grâce à l'identité (1.16), l'utilisation de la matrice d'inertie et des forces de Coriolis dans la loi de commande n'est pas

nécessaire afin de démontrer la stabilité du système dans le cas idéal.

Cette méthode a été appliquée au cas d'un capteur visuel déporté [Kelly 96] ou embarqué [Kelly 97] dans le cadre d'une commande par asservissement visuel d'un robot plan à deux axes observant un point (dans ce cas, et avec des simplifications supplémentaires sur les paramètres géométriques, la stabilité asymptotique locale du système a pu être démontrée). Dans le cas d'une caméra embarquée, la fonction de tâche peut être construite dans l'espace articulaire ou dans l'image. Une étude comparative des deux méthodes a été effectuée par [Reyes 98]. Par rapport à la technique de linéarisation entrée/sortie, le comportement du système est complexe à cause du couplage entre les variables d'état dans la loi de commande. Ceci produit des trajectoires moins contrôlables dans l'image et il sera probablement difficile d'assurer la condition de visibilité tout au long de la tâche dans le cas général du contrôle d'un robot à six degrés de liberté.

Les techniques de commande qu'on vient de décrire tiennent compte du modèle dynamique du robot. Toutefois, comme il a été mis en évidence par [Samson 91], en effectuant une commande avec un gain  $k$  très grand, il est possible de négliger la dynamique du robot et de réaliser une commande dite "en vitesse".

### 1.2.3 Commande en vitesse

Considérons l'équation en boucle fermée idéale du système (1.18) rebouclé par la commande (1.20) :

$$\ddot{\mathbf{e}} + k \mathbf{G} (\dot{\mathbf{e}} + \lambda \mathbf{D} \mathbf{e}) = 0 \quad (1.24)$$

où la fonction de tâche converge asymptotiquement vers zéro. Si on considère une commande à grands gains ( $k \rightarrow \infty$ ) et dans les cas où les accélérations et les perturbations ne sont pas trop grandes, l'équation (1.24) devient :

$$\dot{\mathbf{e}} + \lambda \mathbf{D} \mathbf{e} = 0 \quad (1.25)$$

Donc, quand le gain  $k$  est assez grand, l'équation (1.24) a une allure qui se rapproche de celle de l'équation (1.25) et on cherche à réaliser approximativement un comportement exponentiel pour l'erreur de tâche en considérant seulement la cinématique du robot. En effet, la dérivée de la fonction de tâche peut être obtenue à partir de l'équation (1.10) :

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{r}_c} \mathbf{v} + \frac{\partial \mathbf{e}}{\partial t} \quad (1.26)$$

$$= \frac{\partial \mathbf{e}}{\partial \mathbf{r}_c} \frac{\partial \mathbf{r}_c}{\partial \mathbf{r}} \dot{\mathbf{r}} + \frac{\partial \mathbf{e}}{\partial t} \quad (1.27)$$

$$= \frac{\partial \mathbf{e}}{\partial \mathbf{r}_c} \frac{\partial \mathbf{r}_c}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{e}}{\partial t} \quad (1.28)$$

où on appellera par la suite :

- $\mathbf{L} = \frac{\partial \mathbf{e}}{\partial \mathbf{r}_c}$  la matrice d'interaction qui lie la variation de la fonction de tâche par rapport à la vitesse  $\mathbf{v}$  de la caméra. Comme la fonction de tâche est construite à partir des informations visuelles, cette matrice peut être aussi écrite  $\mathbf{L} = \frac{\partial \mathbf{e}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{r}}$  mettant ainsi en évidence sa dépendance avec  $\mathbf{s}$ . Cette matrice dépend généralement des paramètres intrinsèques de la caméra et de certains paramètres géométriques de la scène observée.
- $\mathbf{W} = \frac{\partial \mathbf{r}_c}{\partial \mathbf{r}}$  la matrice exprimant le passage entre la vitesse de la caméra et la vitesse de l'effecteur du robot :  $\mathbf{v} = \mathbf{W} \dot{\mathbf{r}}$ . Elle contient la rotation  ${}^c\mathbf{R}_e$  et la translation  ${}^c\mathbf{t}_e$  entre le repère caméra  $\mathcal{F}_c$  et le repère de l'effecteur  $\mathcal{F}_e$  :

$$\mathbf{W} = \begin{bmatrix} \mathbf{I} & [{}^c\mathbf{t}_e]_{\times} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} {}^c\mathbf{R}_e & 0 \\ 0 & {}^c\mathbf{R}_e \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{R}_e & [{}^c\mathbf{t}_e]_{\times} {}^c\mathbf{R}_e \\ 0 & {}^c\mathbf{R}_e \end{bmatrix} \quad (1.29)$$

La calibration de  ${}^c\mathbf{R}_e$  et  ${}^c\mathbf{t}_e$  peut être effectuée grâce à une des méthodes proposées par [Dornaika 95] [Rémy 97]. Toutefois, on cherchera à réaliser une technique d'asservissement visuel assez robuste afin d'éviter toute procédure de calibration trop complexe.

- $\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{r}}{\partial \mathbf{q}}$  la matrice jacobienne du robot exprimant le passage entre la vitesse de l'effecteur et la vitesse articulaire du robot :  $\dot{\mathbf{r}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}$ . Nous supposons par la suite que le jacobien du robot est parfaitement connu et inversible.

Nous allons à présent supposer que la consigne accessible à l'utilisateur n'est autre que la vitesse articulaire désirée  $\dot{\mathbf{q}}_c$ , comme c'est le cas pour beaucoup de robots industriels. Alors, si l'on est capable d'assurer  $\dot{\mathbf{q}}_c \simeq \dot{\mathbf{q}}$ , nous pouvons nous contenter de considérer en tant que grandeur de pseudo-commande, une "consigne de vitesse dans  $se_3$ ". Si on peut supposer la matrice  $\mathbf{W}$  parfaitement connue, on peut alors considérer comme consigne la vitesse de la caméra  $\mathbf{v}$ . Sinon, dans le cas plus général, on considérera la vitesse de l'effecteur du robot  $\dot{\mathbf{r}}$ .

En supposant pour simplifier l'analyse, bien que cela ne soit pas strictement nécessaire [Samson 91], que l'objet considéré est immobile (c'est-à-dire  $\frac{\partial \mathbf{e}}{\partial t} = 0$ ), idéalement,  $\dot{\mathbf{r}}$  aurait donc la forme :

$$\dot{\mathbf{r}} = -\lambda \mathbf{W}^{-1} \mathbf{L}^{-1} \mathbf{D} \mathbf{e} \quad (1.30)$$

et l'utilisation de (1.30) dans (1.27) nous redonne bien le comportement souhaité (1.25). Toutefois, comme nous l'avons déjà vu, seules des approximations peuvent être utilisées dans (1.30), et la forme générale de  $\dot{\mathbf{r}}$  utilisée en pratique sera donc :

$$\dot{\mathbf{r}} = -\lambda \widehat{\mathbf{W}}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{e}} \quad (1.31)$$

où  $\widehat{\mathbf{L}}^{-1}$  est une estimation de l'inverse de la matrice  $\mathbf{L}$  et  $\widehat{\mathbf{W}}^{-1}$  est une approximation de l'inverse de la matrice  $\mathbf{W}$ . En utilisant (1.31) dans (1.27) on obtient donc :

$$\dot{\mathbf{e}} = -\lambda \mathbf{L} \mathbf{W} \widehat{\mathbf{W}}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{e}} \quad (1.32)$$

Afin de fermer la boucle, il est nécessaire d'exprimer la fonction de tâche estimée  $\widehat{\mathbf{e}}$  en fonction de  $\mathbf{e}$ . À partir de l'équation (1.14), on obtient :

$$\widehat{\mathbf{e}} = \mathbf{C}(\widehat{\mathbf{s}}, \mathbf{s}^*, \widehat{\mathbf{A}}, \widehat{\mathbf{g}}) (\widehat{\mathbf{s}} - \mathbf{s}^*) \quad (1.33)$$

On suppose en général que les informations visuelles sont entachées d'un bruit additif  $\widehat{\mathbf{s}} = \mathbf{s} + \mathbf{b}_s$ . Donc, comme  $\mathbf{s} - \mathbf{s}^* = \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1} \mathbf{e} = \mathbf{C}^+(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \mathbf{g}) \mathbf{e}$ , on peut écrire :

$$\widehat{\mathbf{e}} = \mathbf{E} \mathbf{e} + \mathbf{b}_e \quad (1.34)$$

où  $\mathbf{b}_e = (\mathbf{C}(\widehat{\mathbf{s}}, \widehat{\mathbf{s}}^*, \widehat{\mathbf{A}}, \widehat{\mathbf{g}}) - \mathbf{C}(\mathbf{s}, \mathbf{s}^*, \widehat{\mathbf{A}}, \widehat{\mathbf{g}}))(\mathbf{s} - \mathbf{s}^*) + \mathbf{C}(\widehat{\mathbf{s}}, \widehat{\mathbf{A}}, \widehat{\mathbf{g}}) \mathbf{b}_s$  est un bruit additif sur la fonction de tâche et :

$$\mathbf{E}(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \widehat{\mathbf{A}}, \mathbf{g}, \widehat{\mathbf{g}}) = \mathbf{C}(\mathbf{s}, \mathbf{s}^*, \widehat{\mathbf{A}}, \widehat{\mathbf{g}}) \mathbf{C}^+(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \mathbf{g}) \quad (1.35)$$

On fait donc une distinction entre les erreurs dues à une mauvaise calibration du système, intervenant dans  $\mathbf{E}$ , et les erreurs de mesure intervenant dans  $\mathbf{b}_e$ . Afin de démontrer la convergence de la fonction de tâche vers zéro, on supposera par la suite que le bruit est négligeable  $\mathbf{b}_s \approx 0 \Rightarrow \mathbf{b}_e \approx 0$  (dans le cas contraire, la bornitude de la fonction de tâche peut être démontrée comme en [Samson 91]). Le système en boucle fermée peut enfin s'écrire :

$$\dot{\mathbf{e}} = -\lambda \mathbf{L} \mathbf{W} \widehat{\mathbf{W}}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{D} \mathbf{E} \mathbf{e} = -\lambda \mathbf{Q}(\mathbf{e}) \mathbf{e} \quad (1.36)$$

Il est bien connu qu'une condition suffisante à respecter pour assurer la décroissance de  $\|\mathbf{e}\|$  est  $\mathbf{Q}(\mathbf{e}) > 0 \forall \mathbf{e}, t$ . En effet, en multipliant à droite et à gauche l'équation (1.36) on obtient :

$$\mathbf{e}^T \dot{\mathbf{e}} = -\lambda \mathbf{e}^T \mathbf{Q}(\mathbf{e}) \mathbf{e} \quad (1.37)$$

Donc, si  $\mathbf{Q}(\mathbf{e})$  est une matrice positive (au sens où une matrice  $\mathbf{Q}$  de dimension  $(m \times m)$  est positive si  $\mathbf{e}^T \mathbf{Q} \mathbf{e} > 0, \forall \mathbf{e} \neq 0 \in \mathbb{R}^m$ ), la dérivée de la norme de la fonction de tâche est négative :

$$\frac{1}{2} \frac{d\|\mathbf{e}\|^2}{dt} = -\lambda \mathbf{e}^T \mathbf{Q}(\mathbf{e}) \mathbf{e} < 0 \quad (1.38)$$

La condition suffisante de stabilité peut donc s'écrire :

$$\mathbf{Q} = \mathbf{L} \mathbf{W} \widehat{\mathbf{W}}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{D} \mathbf{E} > 0 \quad (1.39)$$

**Remarque :** Dans certains cas, la stabilité s'étudie plus facilement à partir de la boucle fermée sur  $\widehat{\mathbf{e}}$  que sur  $\mathbf{e}$ . Toutefois, dans notre cas la dérivée de  $\widehat{\mathbf{e}}$  s'écrit :

$$\frac{d\widehat{\mathbf{e}}}{dt} = \frac{\partial \widehat{\mathbf{e}}}{\partial \mathbf{e}} \frac{d\mathbf{e}}{dt} = \mathbf{E} \dot{\mathbf{e}} \quad (1.40)$$

et, en utilisant l'équation (1.32), la boucle fermée sur  $\hat{\mathbf{e}}$  est :

$$\frac{d\hat{\mathbf{e}}}{dt} = -\lambda \mathbf{E} \mathbf{L} \mathbf{W} \widehat{\mathbf{W}}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{D} \hat{\mathbf{e}} \quad (1.41)$$

L'analyse de la stabilité semble donc présenter une difficulté équivalente à partir des équations (1.36) ou (1.41) car elles diffèrent seulement par la position de la matrice  $\mathbf{E}$ .

Bien que le schéma de contrôle soit simplifié, car la dynamique et la cinématique du robot ne sont pas prises en compte, la démonstration de la stabilité du système en tenant compte des erreurs de calibration intervenant dans  $\mathbf{L}$ ,  $\mathbf{W}$  et  $\mathbf{E}$  reste un problème complexe. La robustesse de la loi de commande par rapport aux erreurs de calibration, mais aussi par rapport aux erreurs de mesure, dépend de la nature de la fonction de tâche utilisée pour réaliser l'asservissement visuel. Nous analysons maintenant plus en détail les deux techniques d'asservissement visuel les plus répandues dans la littérature.

## 1.3 Asservissement visuel 3D

### 1.3.1 Modélisation

Le principe de l'asservissement visuel 3D est de contrôler la caméra dans l'espace cartésien en construisant la fonction de tâche à partir de l'estimation du déplacement de la caméra entre deux images. Soient  $\mathcal{F}_0$  le repère associé à l'objet observé,  $\mathcal{F}_d = \mathcal{F}^*$  et  $\mathcal{F}_c = \mathcal{F}$ , les repères associés à la caméra dans sa position désirée et courante (voir Figure 1.5).

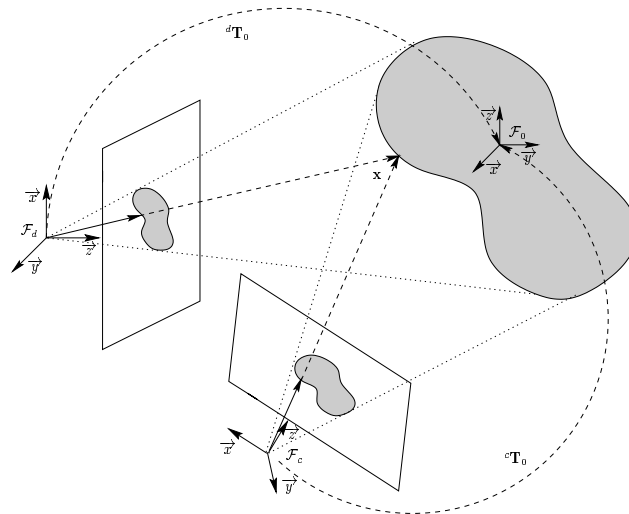


FIG. 1.5 – Modélisation de l'asservissement visuel 3D

En connaissant les coordonnées, exprimées dans  $\mathcal{F}_0$ , d'au moins quatre points de l'objet [Lowe 91] [Dementhon 95] (c'est-à-dire que le modèle 3D de l'objet, représenté par un vecteur de paramètres géométriques  $\mathbf{g}$ , est supposé être connu), il est possible, à partir de leur projection dans l'image, de calculer la position désirée de la caméra (représentée dans la

Figure 1.5 par la matrice de transformation  ${}^d\mathbf{T}_0(\mathbf{s}^*, \mathbf{A}, \mathbf{g})$ ) et la position courante de la caméra (représentée par la matrice de transformation  ${}^c\mathbf{T}_0(\mathbf{s}, \mathbf{A}, \mathbf{g})$ ). Le déplacement de la caméra à réaliser pour atteindre la position désirée peut donc être obtenu :

$$\mathbf{T}(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \mathbf{g}) = {}^c\mathbf{T}_0(\mathbf{s}^*, \mathbf{A}, \mathbf{g}) \left( {}^d\mathbf{T}_0(\mathbf{s}, \mathbf{A}, \mathbf{g}) \right)^{-1} = \begin{bmatrix} {}^c\mathbf{R}_d(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \mathbf{g}) & {}^c\mathbf{t}_d(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \mathbf{g}) \\ 0 & 1 \end{bmatrix} \quad (1.42)$$

Ce déplacement peut théoriquement être effectué en une seule fois (c'est-à-dire en boucle ouverte), ou de manière plus robuste en boucle fermée car une mauvaise calibration du système induit des erreurs sur l'estimation de la matrice de transformation homogène. En effet, dans la réalité, seule une estimation des paramètres du système est disponible, et la matrice de transformation homogène mesurée est :

$$\hat{\mathbf{T}}(\mathbf{s}, \mathbf{s}^*, \hat{\mathbf{A}}, \hat{\mathbf{g}}) = {}^c\hat{\mathbf{T}}_0(\mathbf{s}^*, \hat{\mathbf{A}}, \hat{\mathbf{g}}) \left( {}^d\hat{\mathbf{T}}_0(\mathbf{s}, \hat{\mathbf{A}}, \hat{\mathbf{g}}) \right)^{-1} = \begin{bmatrix} {}^c\hat{\mathbf{R}}_d(\mathbf{s}, \mathbf{s}^*, \hat{\mathbf{A}}, \hat{\mathbf{g}}) & {}^c\hat{\mathbf{t}}_d(\mathbf{s}, \mathbf{s}^*, \hat{\mathbf{A}}, \hat{\mathbf{g}}) \\ 0 & 1 \end{bmatrix} \quad (1.43)$$

À partir de la localisation 3D, on dispose non seulement du déplacement de la caméra mais aussi des coordonnées courantes  $\mathbf{x}$  et désirées  $\mathbf{x}^*$  des points de la cible exprimées respectivement dans le repère courant  $\mathcal{F}$  et désiré  $\mathcal{F}^*$  de la caméra. Afin de contrôler l'effecteur du robot, plusieurs façons de procéder ont été proposées. Toutes sont basées sur le contrôle direct de la rotation car, à partir de la matrice de rotation  ${}^c\mathbf{R}_d$ , on peut calculer aisément une représentation minimale  $\mathbf{e}_\omega(\mathbf{u}, \theta)$  (comme par exemple  $\mathbf{e}_\omega(\mathbf{u}, \theta) = \mathbf{u} \sin(\frac{\theta}{2})$ ,  $\mathbf{e}_\omega(\mathbf{u}, \theta) = \mathbf{u} \sin(\theta)$  ou  $\mathbf{e}_\omega(\mathbf{u}, \theta) = \mathbf{u}\theta$ ) où  $\mathbf{u}$  et  $\theta$  sont respectivement l'axe de rotation et l'angle de rotation. En ce qui concerne les degrés de liberté en translation, ils peuvent être commandés en utilisant :

- la translation  ${}^c\mathbf{t}_d$  [Wilson 96] [Martinet 97] ou la même translation dans un mouvement de vissage [Daucher 97]. Le principal avantage de cette méthode est que le contrôle direct de la trajectoire de la caméra dans l'espace cartésien permet de définir des tâches qui sont proches de celles effectuées en robotique classique. Par contre, il n'y a alors strictement aucun contrôle dans l'image. Les informations visuelles utilisées pour l'estimation de la position peuvent donc sortir de l'image au cours de l'asservissement, ce qui empêche la réalisation de la tâche ;
- les coordonnées  $\mathbf{x}$  d'un point de référence 3D [Martinet 96]. Le principal avantage de cette méthode est que l'on a un contrôle (même s'il n'est pas directement dans l'image) d'un point de la cible. Cela augmente en pratique la probabilité pour la cible de rester dans le champ de vision de la caméra pendant l'asservissement. Il faut remarquer que, si on choisit le point 3D de référence comme origine du repère  $\mathcal{F}_0$  attaché à la cible, cette méthode est équivalente à contrôler la translation  ${}^c\mathbf{t}_0 = \mathbf{x}$  afin qu'elle devienne égale à la translation  ${}^d\mathbf{t}_0 = \mathbf{x}^*$ .

Si la caméra n'est pas parfaitement calibrée, ou si il y a des erreurs sur le modèle 3D de la cible, les positions courante et désirée de la caméra ne seront pas correctement estimées et présenteront un biais par rapport à leur valeur réelle. Ces erreurs d'estimation sont

compensées par une commande en boucle fermée et il n'y a alors pas d'inconvénient à la convergence. Toutefois, les erreurs de calibration influencent grandement la manière dont le système arrive à la convergence, surtout quand le déplacement de la caméra est important. Il faut aussi remarquer que, de manière générale, l'étape de reconstruction à partir des informations visuelles peut rendre la fonction de tâche assez sensible aux erreurs de mesure.

Si on note  $\mathbf{e}_v$  la partie de la fonction de tâche relative au contrôle de la translation, la fonction de tâche  $\mathbf{e} = [\mathbf{e}_v \quad \mathbf{e}_\omega]^T$  correspondant aux deux cas décrits précédemment peut être exprimée de la manière suivante :

- dans le cas du contrôle de la translation, on a  $\mathbf{e}_v = {}^c\mathbf{t}_d$  et on obtient une matrice d'interaction de la forme :

$$\mathbf{L} = \begin{bmatrix} -\mathbf{I}_3 & [{}^c\mathbf{t}_d]_\times \\ \mathbf{0} & \mathbf{L}_\omega \end{bmatrix} \quad (1.44)$$

où  $[{}^c\mathbf{t}_d]_\times$  est la matrice de pré-produit vectoriel associée au vecteur  ${}^c\mathbf{t}_d$  et la matrice  $\mathbf{L}_\omega$  est la matrice jacobienne liée à la représentation de la rotation choisie. Nous présentons en Annexe A la détermination de  $\mathbf{L}_\omega$  pour  $\mathbf{e}_\omega = \mathbf{u}\theta$ . La forme de  $\mathbf{L}_\omega$  pour  $\mathbf{e}_\omega = \mathbf{u}\sin(\frac{\theta}{2})$  et  $\mathbf{e}_\omega = \mathbf{u}\sin(\theta)$  est donnée dans [Samson 91].

- dans le cas du contrôle d'un point dans le repère caméra, on a  $\mathbf{e}_v = \mathbf{x} - \mathbf{x}^*$  et on obtient une matrice d'interaction de la forme :

$$\mathbf{L} = \begin{bmatrix} -\mathbf{I}_3 & [\mathbf{x}]_\times \\ \mathbf{0} & \mathbf{L}_\omega \end{bmatrix} \quad (1.45)$$

Dans les deux cas, on observe dans la matrice d'interaction un découplage de la rotation par rapport à la translation. Ceci est une propriété intéressante comme nous le verrons dans le chapitre 3. Le schéma bloc correspondant à l'asservissement visuel 3D est représenté sur la Figure 1.6.

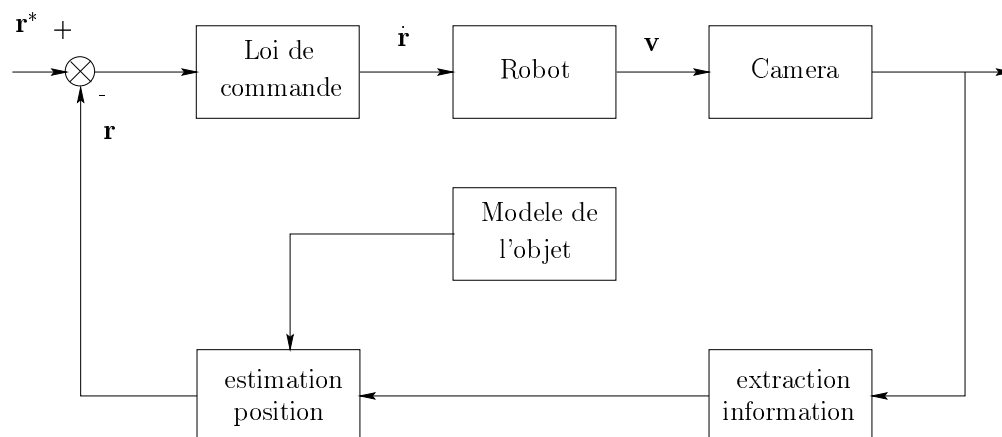


FIG. 1.6 – Schéma à bloc de l'asservissement visuel 3D

### 1.3.2 Commande et analyse de la stabilité

On “souhaite” pouvoir obtenir comme équation du système commandé :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (1.46)$$

Si on suppose que la cible est statique et que la matrice de passage caméra/effecteur est parfaitement connue, la loi de commande correspondante est donnée par :

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}^{-1} \widehat{\mathbf{e}} \quad (1.47)$$

En fait, on obtient avec cette commande :

$$\dot{\mathbf{e}} = -\lambda \mathbf{L} \widehat{\mathbf{L}}^{-1} \widehat{\mathbf{e}} = -\lambda \mathbf{L} \widehat{\mathbf{L}}^{-1} \mathbf{E} \mathbf{e} \quad (1.48)$$

où la matrice  $\mathbf{E} = \mathbf{E}(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \widehat{\mathbf{A}}, \mathbf{g}, \widehat{\mathbf{g}})$  dépend des paramètres de calibration du système. Le système en boucle fermée peut s'écrire :

– dans le cas du contrôle de la translation :

$$\mathbf{Q} = \begin{bmatrix} \mathbf{I}_3 & \begin{bmatrix} {}^c\mathbf{t}_d - {}^c\widehat{\mathbf{t}}_d \\ \mathbf{L}_\omega \widehat{\mathbf{L}}_\omega^{-1} \end{bmatrix}_\times \\ 0 & \mathbf{L}_\omega \widehat{\mathbf{L}}_\omega^{-1} \end{bmatrix} \mathbf{E}(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \widehat{\mathbf{A}}, \mathbf{g}, \widehat{\mathbf{g}}) \quad (1.49)$$

– dans le cas du contrôle du point 3D :

$$\mathbf{Q} = \begin{bmatrix} \mathbf{I}_3 & \begin{bmatrix} \mathbf{x} - \widehat{\mathbf{x}} \\ \mathbf{L}_\omega \widehat{\mathbf{L}}_\omega^{-1} \end{bmatrix}_\times \\ 0 & \mathbf{L}_\omega \widehat{\mathbf{L}}_\omega^{-1} \end{bmatrix} \mathbf{E}(\mathbf{s}, \mathbf{s}^*, \mathbf{A}, \widehat{\mathbf{A}}, \mathbf{g}, \widehat{\mathbf{g}}) \quad (1.50)$$

Dans les deux cas, la modélisation de  $\mathbf{L} \widehat{\mathbf{L}}^{-1}$  est très simple. Par contre, l'analyse de la stabilité paraît très difficile à réaliser. En effet, la modélisation de la matrice  $\mathbf{E}$  est problématique car elle dépend de la matrice  $\mathbf{C}$  (voir équation (1.35)) dont il semble impossible d'avoir la forme analytique. En effet, l'estimation de la fonction de tâche est très généralement obtenue avec des méthodes itératives soit linéaires [Dementhon 95], soit non-linéaires [Lowe 91].

De plus, les erreurs de mesure dans l'image peuvent avoir une grande influence sur la fonction de tâche estimée en raison de l'instabilité potentielle des calculs de pose (notamment si quatre points coplanaires sont utilisés [Chaumette 98]).

Comme l'erreur commise sur l'estimation de la fonction de tâche ne peut pas être exprimée en fonction des erreurs de calibration de la caméra et des erreurs sur le modèle de la cible, on suppose en général que :

– le modèle géométrique de la cible est parfaitement connu :  $\widehat{\mathbf{g}} = \mathbf{g}$  ;

– les paramètres intrinsèques de la caméra sont parfaitement connus :  $\widehat{\mathbf{A}} = \mathbf{A}$ .



Par conséquent, on pose non seulement  $\mathbf{E} = \mathbf{I}$ , mais aussi  $\mathbf{Q} = \mathbf{I}$ , ce qui est une hypothèse très forte.

De plus, même en supposant un monde parfait, si la caméra est contrôlée complètement dans l'espace cartésien, on ne peut pas assurer la condition de visibilité de la tâche. La cible peut alors sortir de l'image si le déplacement initial de la caméra est trop important. C'est pourquoi il est souhaitable, d'une manière ou d'une autre, d'introduire un contrôle exprimé directement dans l'image.

## 1.4 Asservissement visuel 2D

### 1.4.1 Modélisation

Le principe de l'asservissement visuel 2D est de contrôler les mouvements de la caméra en construisant la fonction de tâche de manière à utiliser directement les informations de l'image sans passer par une estimation (ou calcul) explicite des erreurs de position dans l'espace cartésien (voir Figure 1.7).

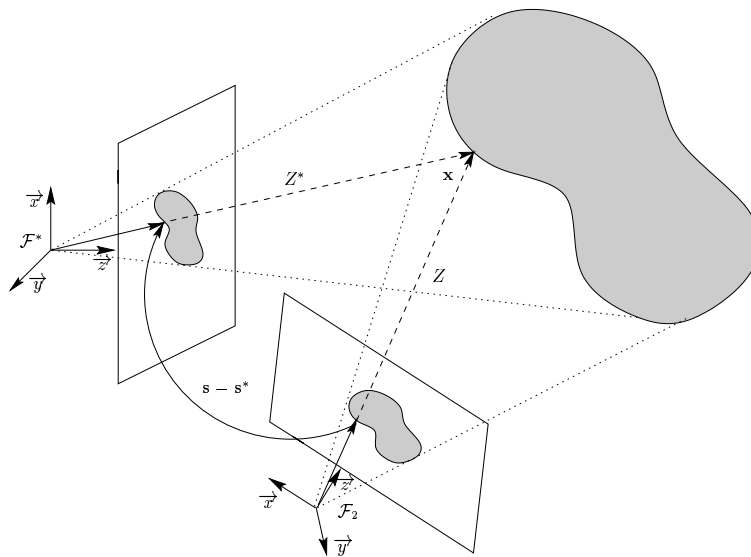


FIG. 1.7 – Modélisation de l'asservissement visuel 2D

Différents types d'informations visuelles extraites dans l'image peuvent être utilisés. Par exemple, [Bien 93] utilise les moments centraux de la surface observée et [Hager 95] utilise des invariants projectifs. Le comportement du système commandé par asservissement visuel dépend du choix des informations visuelles utilisées [Feddema 93] [Chaumette 98], de leur nombre et leur configuration [Hashimoto 98], et du point de vue duquel ces informations sont observées [Sharma 95] [Sharma 94] [Nelson 96]. On s'intéressera par la suite au cas où les informations visuelles sont les coordonnées de points, bien qu'il ait été mis en évidence que ce n'est pas toujours le meilleur choix [Chaumette 98].

Il faut souligner que nous nous intéressons à des tâches de positionnement contrôlant les six degrés de liberté de la caméra, mais le même formalisme peut s'appliquer au contrôle d'un nombre inférieur d'informations visuelles, comme par exemple le centrage d'un point dans l'image, le positionnement par rapport à une sphère ou à un cylindre (pour une classification des différentes tâches visuelles, voir [Chaumette 93]). Les degrés de liberté restants peuvent alors être utilisés pour réaliser une tâche secondaire telle que l'évitement des butées articulaires et des singularités du robot [Nelson 95] [Marchand 96], ou l'évitement d'obstacles et des occultations [Marchand 98].

Soit  $\mathbf{s}_i$  la valeur courante des informations visuelles observées par la caméra (c'est-à-dire un paramètre représentant la position dans l'image d'un point, d'une droite, ou d'une ellipse, etc.) et  $\mathbf{s}_i^*$  la valeur désirée de  $\mathbf{s}_i$  à atteindre dans l'image. Considérons par exemple le cas d'un point de coordonnées  $\mathbf{s}_i = [u_i \ v_i]^T$  en pixels dans le plan image. La dérivée des coordonnées du point dans l'image est liée au torseur cinématique de la caméra  $\mathbf{v} = [\mathbf{v}^T \ \boldsymbol{\omega}^T]^T$  par la relation:

$$\dot{\mathbf{s}}_i = \frac{\partial \mathbf{s}_i}{\partial \mathbf{r}}(\mathbf{s}_i, Z_i, \mathbf{A}) \mathbf{v} = \mathbf{L}_{s_i}(\mathbf{s}_i, Z_i, \mathbf{A}) \mathbf{v} \quad (1.51)$$

où  $\mathbf{L}_{s_i}(\mathbf{s}_i, Z_i, \mathbf{A})$  est dite la matrice d'interaction associée à  $\mathbf{s}_i$ . On a :

$$\mathbf{L}_{s_i}(\mathbf{s}_i, Z_i, \mathbf{A}) = \begin{bmatrix} \alpha_u & \alpha_{uv} \\ 0 & \alpha_v \end{bmatrix} \begin{bmatrix} -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i y_i & -(1+x_i^2) & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & (1+y_i^2) & -x_i y_i & -x_i \end{bmatrix} \quad (1.52)$$

où  $\mathbf{A}$  est la matrice contenant les paramètres intrinsèques de la caméra (voir équation (1.8)),  $Z_i$  est la profondeur du point 3D correspondant au point image considéré, et où les coordonnées métriques  $x_i$  et  $y_i$  sont fonction des coordonnées pixels  $u_i$  et  $v_i$  et des paramètres intrinsèques de la caméra par la relation :

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha_u} & -\frac{\alpha_{uv}}{\alpha_u \alpha_v} \\ 0 & \frac{1}{\alpha_v} \end{bmatrix} \begin{bmatrix} u_i - u_0 \\ v_i - v_0 \end{bmatrix} \quad (1.53)$$

Une méthode générale de calcul du torseur d'interaction pour des informations visuelles plus complexes (comme par exemple les segments, droites, cercles, sphères et cylindres) est donnée dans [Chaumette 90].

Un minimum de quatre points est nécessaire pour réaliser une tâche de positionnement rigide. Théoriquement, trois points sont suffisants mais, dans ce cas, le système peut rencontrer une singularité de la matrice d'interaction [Michel 93] ou atteindre un minimum global non désiré de la norme de la fonction de tâche (car il existe quatre positions différentes de la caméra pour lesquelles les trois points se projettent dans la même configuration dans l'image).

En général, il est possible d'utiliser la redondance d'informations afin de rendre plus robuste la loi de commande aux erreurs de mesure. Si l'on considère  $p$  points dans l'image, le vecteur contenant toutes les informations visuelles  $\mathbf{s} = [\mathbf{s}_1^T \ \mathbf{s}_2^T \ \dots \ \mathbf{s}_p^T]^T$  est de dimension  $2p$ . La fonction de tâche  $\mathbf{e}$ , correspondant à la régulation de  $\mathbf{s} - \mathbf{s}^*$ , est définie par:

$$\mathbf{e} = \mathbf{C} (\mathbf{s} - \mathbf{s}^*) \quad (1.54)$$

où  $\mathbf{C} = \frac{\partial \mathbf{e}}{\partial \mathbf{s}}$  est une matrice de dimension  $(m \times 2p)$ ,  $m$  étant la dimension de la fonction de tâche (soit 6 pour réaliser une liaison rigide). Le jacobien de la fonction de tâche, de dimension  $(m \times 6)$ , est donc obtenue en multipliant la matrice  $\mathbf{C}$  par la matrice d'interaction  $\mathbf{L}_s = \frac{\partial \mathbf{s}}{\partial \mathbf{r}_c}$ , de dimension  $(2p \times 6)$ , associée aux informations visuelles :

$$\mathbf{L}(\mathbf{s}, \mathbf{z}, \mathbf{A}) = \frac{\partial \mathbf{e}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{r}_c} = \mathbf{C} \mathbf{L}_s(\mathbf{s}, \mathbf{z}, \mathbf{A}) = \mathbf{C} \begin{bmatrix} \frac{\partial \mathbf{s}_1}{\partial \mathbf{r}}(\mathbf{s}_1, Z_1, \mathbf{A}) \\ \frac{\partial \mathbf{s}_2}{\partial \mathbf{r}}(\mathbf{s}_2, Z_2, \mathbf{A}) \\ \vdots \\ \frac{\partial \mathbf{s}_p}{\partial \mathbf{r}}(\mathbf{s}_p, Z_p, \mathbf{A}) \end{bmatrix} \quad (1.55)$$

où  $\mathbf{z} = [Z_1 \ Z_2 \ \dots \ Z_p]^T$  est le vecteur contenant les profondeurs de tous les points considérés.

Dans le cas de l'asservissement visuel 3D, la matrice  $\mathbf{C}$  était imposée par le choix de la fonction de tâche; dans le cas présent, c'est le choix de la matrice  $\mathbf{C}$  qui détermine la fonction de tâche :

- une solution simple largement adoptée est de choisir  $\mathbf{C} = \mathbf{I}_m$  avec  $m = 2p$ . Afin d'avoir une fonction de tâche admissible, il faudrait choisir  $m = 6$  (donc 3 points), mais ce faisant, comme nous l'avons déjà dit, il est possible de rencontrer une singularité de la matrice d'interaction ou d'atteindre un minimum global non désiré de la norme de la fonction de tâche. Si  $m > 6$  (donc pour  $p > 3$  points), la dimension du vecteur de sortie (ou de tâche) est supérieure à celle du vecteur de commande (le vecteur vitesse de la caméra est de dimension 6) et on sort du cadre défini par l'admissibilité de la fonction de tâche. En effet, le nombre d'informations visuelles que l'on désire contrôler est supérieur aux degrés de liberté disponibles et il existe alors des trajectoire irréalisables, c'est-à-dire pour lesquelles il n'existe pas de mouvement de la caméra correspondant [Chaumette 98]. Si une telle trajectoire est nécessaire pour atteindre la convergence, le robot ne pourra alors pas la réaliser et le système atteindra un minimum local de la norme de la fonction de tâche.
- une solution qui permet de tenir compte de la redondance d'informations est de choisir  $\mathbf{C} = \mathbf{L}_s^+(\mathbf{s}^*, \mathbf{z}^*, \mathbf{A})$  [Chaumette 90] [Espiau 92], c'est-à-dire la pseudo-inverse de la matrice d'interaction calculée pour  $\mathbf{s} = \mathbf{s}^*$  et  $\mathbf{z} = \mathbf{z}^*$ , où  $\mathbf{z}^*$  est une valeur approximée de  $\mathbf{z}$  à la position désirée de la caméra. En général, comme les valeurs réelles de  $\mathbf{z}^*$  et  $\mathbf{A}$  ne sont pas connues, on choisit une estimation de la matrice réelle :  $\mathbf{C} = \mathbf{L}_s^+(\mathbf{s}^*, \hat{\mathbf{z}}^*, \hat{\mathbf{A}})$ .
- une autre solution qui permet de tenir compte de la redondance d'informations est de choisir  $\mathbf{C} = \mathbf{L}_s^T(\mathbf{s}^*, \hat{\mathbf{z}}^*, \hat{\mathbf{A}})$  [Hashimoto 93b]. Ce choix ne nécessite pas l'inversion de la matrice d'interaction.

Le schéma bloc de l'asservissement visuel 2D est donné en Figure 1.8. Par rapport à celui de l'asservissement visuel 3D, on peut remarquer que le modèle géométrique de la cible n'est plus nécessaire, mais il faut toutefois introduire dans la commande une estimation ou une approximation des profondeurs des points.

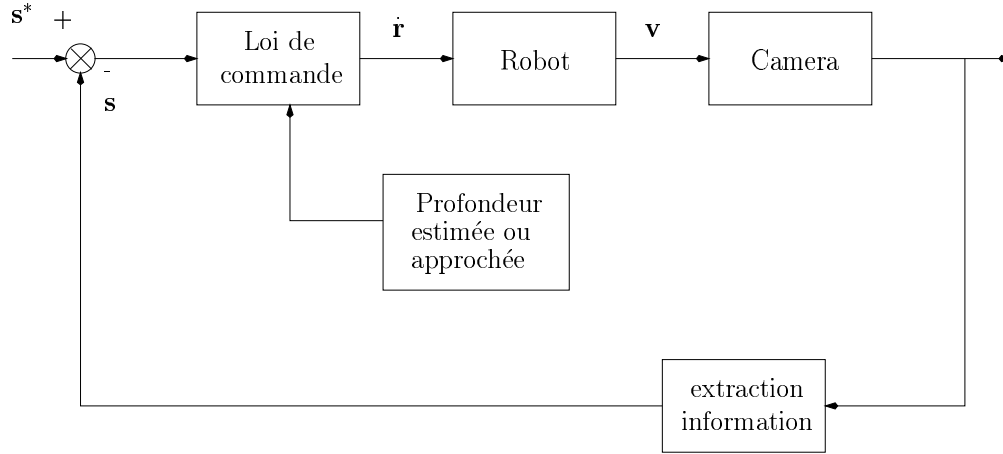


FIG. 1.8 – Schéma bloc de l'asservissement visuel 2D

### 1.4.2 Commande et analyse de la stabilité

Si on impose encore une fois une convergence exponentielle découplée de la fonction de tâche ( $\mathbf{D} = \mathbf{I}_m$ ) et si on suppose la cible statique et la matrice de passage caméra/effecteur parfaitement connue, la loi de commande (1.31) devient:

$$\mathbf{v} = -\lambda \hat{\mathbf{L}}^{-1} \hat{\mathbf{e}} \quad (1.56)$$

Il faut souligner que, dans une tâche de positionnement, un contrôleur proportionnel est suffisant car le système à contrôler est déjà de nature intégrale et le contrôleur bas niveau du robot est généralement un PID. Une intégration supplémentaire est utile seulement dans le cas de poursuite d'une cible mobile à vitesse constante [Chaumette 91]. Le cas d'une cible à trajectoire connue a été étudié par [Allen 93a] [Hashimoto 95], et le cas d'une cible avec changement abrupt de vitesse par [Bensalah 95]. D'autres lois de commande ont été utilisées: une étude comparative entre une commande proportionnelle intégrale, une commande à placement de pôles et une commande LQG a été effectuée par [Papanikolopoulos 93b] [Papanikolopoulos 93a] [Papanikolopoulos 94] et une commande optimale de type LQ a été utilisée par [Hashimoto 93b]. De plus, une commande de type GPC a été récemment appliquée à l'asservissement visuel [Gangloff 98].

Le choix de  $\hat{\mathbf{L}}$  est souvent lié au choix de la matrice  $\mathbf{C}$  car  $\hat{\mathbf{L}} = \mathbf{C}\hat{\mathbf{L}}_s$ . Différents choix pour  $\hat{\mathbf{L}}_s$  sont donc possibles:

- si  $\mathbf{C} = \mathbf{I}_m$  alors:
  - la matrice d'interaction peut être approchée par une équation aux différences finies et estimée en effectuant une série de mouvements de calibration orthogonaux

ou en ligne (l'estimation est ensuite mise à jour avec une formule de Broyden de rang 1) [Hosoda 94] [Suanto 97] [Jägersand 97], ou en utilisant un réseau de neurones [Suh 93]. En ajoutant un filtrage prédictif, la technique d'estimation en ligne de la matrice d'interaction a été adaptée à des tâches de poursuite par [Piepmeier 98]. Cette méthode ne nécessite aucun modèle à priori du système mais elle est assez lourde à mettre en œuvre (le choix de l'initialisation de la matrice d'interaction est notamment capitale pour éviter des mouvements initiaux peu orthodoxes) et présente un caractère local (l'approximation est valable autour de la zone d'apprentissage).

- la matrice d'interaction peut être calculée à chaque itération sur la base du modèle donné par les équations (1.52) et (1.53), donc :

$$\widehat{\mathbf{L}} = \mathbf{L}_s(\mathbf{s}, \widehat{\mathbf{z}}, \widehat{\mathbf{A}}) \quad (1.57)$$

La matrice d'interaction dépend de la profondeur  $\mathbf{z}$  des points utilisés. Une estimation de la profondeur peut être obtenue en utilisant, comme dans le cas de l'asservissement visuel 3D, un algorithme de calcul de pose (si le modèle géométrique de la cible est connu) [Hashimoto 91], ou en utilisant un algorithme de vision dynamique (si le déplacement de la caméra est connu) [Feddema 90] [Papanikolopoulos 94].

Dans le premier cas, on ne peut pas calculer analytiquement l'équation différentielle qui régit le comportement du système en boucle fermée, d'où l'énorme difficulté de démontrer la stabilité du système. Dans le deuxième cas, la matrice  $\mathbf{Q}$  de la boucle fermée du système est donnée par :

$$\mathbf{Q} = \mathbf{L}_s(\mathbf{s}, \mathbf{z}, \mathbf{A}) \mathbf{L}_s^+(\mathbf{s}, \widehat{\mathbf{z}}, \widehat{\mathbf{A}}) \quad (1.58)$$

Il faut remarquer que, même dans le cas idéal où  $\widehat{\mathbf{A}} = \mathbf{A}$  et  $\widehat{\mathbf{z}} = \mathbf{z}$ , si  $p > 3$  alors  $\mathbf{Q} \neq \mathbf{I}$ . De plus, comme on l'a déjà vu, ce choix peut conduire le système à atteindre un minimum local, voire une singularité de la matrice d'interaction [Chaumette 98].

- si  $\mathbf{C} = \mathbf{L}_s^+(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}})$ , alors :
  - la matrice d'interaction peut être choisie constante :  $\widehat{\mathbf{L}}_s = \mathbf{L}_s(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}})$ , donc :

$$\widehat{\mathbf{L}} = \mathbf{C}\widehat{\mathbf{L}}_s = \mathbf{L}_s^+(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}})\mathbf{L}_s(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}}) = \mathbf{I} \quad (1.59)$$

Dans ce cas, la matrice d'interaction  $\widehat{\mathbf{L}}$  utilisée dans la commande ne peut être singulière si la position désirée a été choisie correctement. L'utilisation d'une matrice constante contraint moins la trajectoire des points dans l'image et ceux-ci peuvent éventuellement sortir du champ de vision de la caméra. Toutefois, la matrice d'interaction réelle peut toujours devenir singulière au cours de la tâche. En

effet, la matrice  $\mathbf{Q}$  de la boucle fermée du système dépend toujours de la matrice d'interaction réelle :

$$\mathbf{Q} = \mathbf{L}\widehat{\mathbf{L}}^{-1} = \mathbf{L}_s^+(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}}) \mathbf{L}_s(\mathbf{s}, \mathbf{z}, \mathbf{A}) \quad (1.60)$$

Dans un voisinage de la positions désirée ( $\mathbf{s} \approx \mathbf{s}^*$  et  $\mathbf{z} \approx \mathbf{z}^*$ ), et dans le cas idéal où la calibration est parfaite ( $\widehat{\mathbf{A}} = \mathbf{A}$  et  $\widehat{\mathbf{z}}^* = \mathbf{z}^*$ ), ce type de commande est stable car il réalise un découplage de la boucle fermée ( $\mathbf{Q} \approx \mathbf{I} > 0$ ).

- la matrice d'interaction  $\widehat{\mathbf{L}}_s$  peut être estimée à chaque itération. Toutefois, ce cas de figure est plus coûteux en temps de calcul sans pour autant apporter d'améliorations. Il ne présente donc pas un grand intérêt en pratique.
- si  $\mathbf{C} = \mathbf{L}_s^T(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}})$ , alors :
  - la matrice d'interaction peut être choisie telle que  $\widehat{\mathbf{L}}_s = \mathbf{L}_s^{T+}(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}})$ , donc :

$$\widehat{\mathbf{L}} = \mathbf{C}\widehat{\mathbf{L}}_s = \mathbf{L}_s^T(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}})\mathbf{L}_s^{T+}(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}}) = \mathbf{I} \quad (1.61)$$

La matrice  $\mathbf{Q}$  de la boucle fermée du système dépend toujours de la matrice d'interaction réelle et elle est donnée par :

$$\mathbf{Q} = \mathbf{L}_s^T(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{A}}) \mathbf{L}_s(\mathbf{s}, \mathbf{z}, \mathbf{A}) \quad (1.62)$$

Par rapport au choix précédent, dans un voisinage de la position désirée ( $\mathbf{s} \approx \mathbf{s}^*$  et  $\mathbf{z} \approx \mathbf{z}^*$ ), et dans le cas idéal où la calibration est parfaite ( $\widehat{\mathbf{A}} = \mathbf{A}$  et  $\widehat{\mathbf{z}}^* = \mathbf{z}^*$ ), ce type de commande est stable car  $\mathbf{Q} \approx \mathbf{L}_s^T \mathbf{L}_s > 0$ , mais on ne réalise pas de découplage de la boucle fermée.

- la matrice d'interaction peut être estimée à chaque itération. Là encore, cela ne présente guère d'intérêt en pratique.

On peut noter que, dans tous les cas, et contrairement à l'asservissement visuel 3D, la matrice  $\mathbf{E}$  qui intervient dans le calcul de la matrice  $\mathbf{Q}$  de la boucle fermée vaut  $\mathbf{E} = \mathbf{I}$  car la matrice  $\mathbf{C}$  ne dépend pas des paramètres réels  $\mathbf{A}$  et  $\mathbf{g}$  et est donc parfaitement connue. L'analyse de la robustesse des lois de commande 2D aux erreurs de calibration semble donc plus simple que celle des lois de commande 3D (on rappelle qu'en asservissement visuel 3D, il semble impossible d'exprimer la matrice  $\mathbf{E}$  de manière analytique). Toutefois, le domaine de convergence et la robustesse des lois de commande 2D n'ont pas pu jusqu'ici être déterminés de manière analytique dans le cas général, car les erreurs de calibration interviennent dans la matrice d'interaction de manière non triviale (voir équations (1.52) et (1.53)). Malgré le couplage et la non-linéarité du système, sa stabilité a été étudiée par [Espiau 93] dans les cas simplifiés suivants :

- 3 points dont on connaît parfaitement les coordonnées 3D (donc la profondeur  $Z$ ). Ce cas est toutefois peu réaliste car si on connaît les coordonnées 3D de 3 points et leur

projection dans l'image, on peut alors retrouver la matrice des paramètres intrinsèques de la caméra.

- 2 points de même profondeur et une caméra commandée seulement en translation. La matrice d'interaction est choisie constante et calculée avec les informations visuelles désirées.
- 3 points de même profondeur, les paramètres de la caméra sont connus et le point principal coïncide avec le centre de l'image. La matrice d'interaction est choisie constante et calculée avec les informations visuelles désirées.

Dans les deux derniers cas, les résultats sont donc valables seulement autour de la position finale. Même si, d'un point de vue théorique, on ne peut rien conclure sur la stabilité du système dans le cas général, on peut observer expérimentalement la convergence de l'asservissement visuel 2D même en présence d'erreurs de calibration très importantes. Pour le type de commande avec une matrice d'interaction constante, la convergence n'est pas seulement réalisable dans un voisinage de la position désirée, comme on pourrait le penser, mais dans un domaine bien plus large [Espiau 93]. Toutefois, la convergence peut ne pas être assurée si le déplacement de la caméra est trop important.

En ce qui concerne la sensibilité au bruit de mesure, contrairement à l'asservissement visuel 3D, l'absence de l'étape de localisation rend l'asservissement visuel 2D plus robuste à ce type d'erreur. On rappelle que le bruit sur la fonction de tâche dû aux erreurs de mesure peut s'écrire (voir équation (1.34)) :

$$\mathbf{b}_e = (\mathbf{C}(\hat{\mathbf{s}}, \hat{\mathbf{s}}^*, \hat{\mathbf{A}}, \hat{\mathbf{g}}) - \mathbf{C}(\mathbf{s}, \mathbf{s}^*, \hat{\mathbf{A}}, \hat{\mathbf{g}}))(\mathbf{s} - \mathbf{s}^*) + \mathbf{C}(\hat{\mathbf{s}}, \hat{\mathbf{A}}, \hat{\mathbf{g}}) \mathbf{b}_s \quad (1.63)$$

À proximité de la convergence ( $\mathbf{s} \approx \mathbf{s}^*$ ), on a donc :

$$\mathbf{b}_e \approx \mathbf{C}(\hat{\mathbf{s}}, \hat{\mathbf{A}}, \hat{\mathbf{g}}) \mathbf{b}_s \quad (1.64)$$

et l'amplification du bruit de mesure  $\mathbf{b}_s$  dans l'image dépend du conditionnement de la matrice  $\mathbf{C} = \frac{\partial \mathbf{e}}{\partial \mathbf{s}}$ . Dans le cas de l'asservissement 3D, cette matrice est impossible à modéliser, mais il est bien connu que, dans une procédure de reconstruction, une petite variation des mesures peut se traduire par une grande variation de la grandeur estimée si le nombre de points utilisés est faible.

## 1.5 Conclusion

Les systèmes d'asservissement visuel ont été étudiés depuis une vingtaine d'années et sont désormais en phase de validation afin de les rendre enfin opérationnels sur des sites industriels. Ces systèmes présentent des caractéristiques différentes et le choix d'une technique plutôt qu'une autre n'est pas toujours facile. Il est donc intéressant de pouvoir effectuer une "comparaison quantitative" entre les différentes techniques. Ceci est possible seulement

entre des caractéristiques communes mesurables, comme par exemple la sensibilité au bruit de mesure. Dans certains cas, on ne peut qu'effectuer une "comparaison qualitative" comme dans le cas du comportement dynamique de l'asservissement. L'objectif n'étant pas de déterminer la meilleure technique, mais plutôt la technique la mieux adaptée pour une application donnée, nous définissons cinq critères de comparaison et nous les appliquons aux techniques 2D et 3D sur la base des considérations théoriques discutées dans ce chapitre :

- **Information** nécessaire pour la modélisation. On peut distinguer l'information mesurée dans l'image de celle a priori nécessaire pour la mise en œuvre de l'asservissement (modèle de la cible, profondeur des points). Il est évident que le nombre d'informations a priori nécessaires est directement lié à la portabilité du schéma d'asservissement. Vu la nature différente de ces informations, on ne peut qu'effectuer une comparaison qualitative. En ce qui concerne les informations mesurées, les deux techniques d'asservissement visuel 2D et 3D nécessitent au moins quatre points mis en correspondance dans les deux images. L'asservissement visuel 3D nécessite la connaissance de la structure de l'objet observé tandis que, dans l'asservissement visuel 2D, on peut suppléer la connaissance de la profondeur par une approximation ou une estimation en ligne.
- **Performance.** Nous considérons comme principal indice de la performance la cadence. Afin de faire abstraction de l'architecture du système de vision, de l'architecture du contrôleur du robot et du robot lui-même, seul le délai introduit par le contrôleur visuel doit être mesuré. Il est clair que d'autres indices de performance peuvent être utilisés tel que le rapport entre le temps de convergence et l'énergie dépensée. Le problème du retard introduit par le système de vision dans la boucle fermée se pose pour toutes les techniques d'asservissement visuel. Toutefois, un étage d'estimation supplémentaire est introduit dans l'asservissement visuel 3D, ou dans le cas de l'estimation en ligne de la profondeur pour l'asservissement visuel 2D.
- **Sensibilité** au bruit de mesure. Une comparaison quantitative est possible si toutes les techniques utilisent les mêmes informations visuelles. Normalement, le bruit de mesure devrait avoir une influence seulement sur le comportement du système à la convergence. Toutefois, quand des techniques de reconstruction sont utilisées, il peut avoir également une influence sur la stabilité du système. Les systèmes contrôlant directement dans l'image sont donc théoriquement moins sensibles aux erreurs de mesure. Une technique d'estimation basée sur la minimisation de critères non linéaires peut diminuer la sensibilité de l'asservissement 3D, mais elle diminue également sa performance.
- **Comportement** dynamique. La trajectoire de l'effecteur dans l'espace et la trajectoire des informations visuelles dans l'image peuvent être comparées qualitativement. Une mesure quantitative peut être établie sur la probabilité d'assurer la visibilité de la tâche tout au long de l'asservissement. Dans le cas de l'asservissement visuel 2D, les trajectoires de l'effecteur du robot peuvent être assez surprenantes car son mouvement est contrôlé directement dans l'image. De plus, la présence d'éventuelles singularités de



la matrice d'interaction peut poser des problèmes d'admissibilité de la tâche (minima locaux). Par opposition, dans le cas de l'asservissement visuel 3D, les trajectoires des informations visuelles dans l'image peuvent être assez surprenantes car le robot est contrôlé directement dans l'espace cartésien. Dans le premier cas, le problème de la visibilité de la tâche est moins délicat car on a un contrôle "direct" des informations visuelles dans l'image. Dans le deuxième cas, même en absence d'erreurs de calibration, il est difficile d'assurer que l'objet ne sorte pas du champ de vue de la caméra.

- **Robustesse** aux erreurs de calibration. Une comparaison quantitative peut s'effectuer en ce qui concerne les erreurs sur les paramètres intrinsèques de la caméra et sur la transformation caméra/effecteur. Les erreurs dues aux caractéristiques géométriques propres à chaque technique d'asservissement peuvent être comparées qualitativement. Mis à part les cas très simples, il est très difficile de produire des résultats formels sur la robustesse de ces systèmes vis à vis des erreurs de calibration. Ceci est dû dans le cas de l'asservissement visuel 2D au couplage entre la commande en translation et en rotation, et à l'intervention des erreurs de calibration de manière non triviale dans la matrice d'interaction, ce qui rend l'étude analytique de la boucle fermée très compliquée même dans le cas simple où seulement 3 points sont utilisés. Dans le cas de l'asservissement visuel 3D, même s'il présente un découplage de la rotation par rapport à la translation, la difficulté de la démonstration est due d'une part à l'absence de contrôle des informations visuelles observées (problème de visibilité de la tâche), et d'autre part, même si un contrôle indirect dans l'image est introduit, à l'impossibilité de modéliser la matrice d'erreur entre la fonction de tâche mesurée et la fonction de tâche réelle.

Chaque critère de comparaison a un poids différent selon la tâche, le matériel disponible, etc. Sur la base de ces observations, un schéma d'asservissement visuel qui puisse réunir les points positifs de l'asservissement visuel 2D et 3D devrait avoir les caractéristiques suivantes:

- permettre un contrôle direct dans l'image afin d'assurer la visibilité de la tâche.
- permettre le découplage de la loi de commande en rotation de celle en translation.
- utiliser un nombre minimum d'informations a priori sur la structure de l'objet observé.

Nous reviendrons dans le chapitre 4 sur les différentes techniques d'asservissement visuel décrites dans ce chapitre afin de les comparer expérimentalement à celles que nous avons développées. Nous allons voir dans le chapitre suivant qu'il est possible d'obtenir une estimation du déplacement, à un facteur d'échelle près, entre deux images d'un objet en utilisant un nombre minimal d'informations sur ce dernier. À partir de cette estimation, nous présenterons dans le chapitre 3 un schéma d'asservissement visuel permettant à la fois le découplage de la loi de commande en rotation de celle en translation, un contrôle direct dans l'image, et assurant la visibilité de la tâche. Nous déterminerons également les conditions analytiques pour assurer la robustesse de la stabilité des lois de commande.

# Chapitre 2

## Reconstruction projective et euclidienne

Dans ce chapitre, nous étudions le problème de la reconstruction partielle du déplacement d'une caméra entre deux vues d'un objet statique dont le modèle est inconnu. Après un bref état de l'art, nous proposons d'estimer ce déplacement à partir d'une homographie. En effet, cette solution semble la plus adaptée dans le cas considéré d'une tâche de positionnement par asservissement visuel, car l'objectif est d'amener les deux images, initialement éloignées, à converger l'une vers l'autre.

### 2.1 Introduction

Le problème classique de la reconstruction du déplacement d'une caméra entre deux vues (“structure from motion” en anglais) est, par sa propre nature, non-linéaire. Par conséquent, l'approche habituelle pour résoudre ce problème est divisée en deux étapes :

- on utilise d'abord un algorithme linéaire pour estimer la géométrie épipolaire sans tenir compte des contraintes non-linéaires ;
- on affine ensuite l'estimation avec un algorithme de minimisation non-linéaire [Jerian 91].

Nous nous intéresserons ici seulement à la première étape, car il est difficile d'obtenir une estimation initiale précise et la deuxième étape est en général très sensible à l'initialisation. De plus, nous nous limiterons à une méthode linéaire afin de disposer d'un algorithme tournant à la cadence vidéo.

Différentes méthodes ont été proposées pour résoudre linéairement le problème de “structure from motion”. Elles sont généralement basées sur l'estimation de la matrice fondamentale [Luong 96] si la caméra n'est pas calibrée (on utilise alors les points de l'image exprimés en coordonnées pixels), ou de la matrice essentielle [LonguetHiggins 81, Hartley 97] si la caméra est calibrée (on utilise alors les points de l'image exprimés en coordonnées normalisées). La matrice fondamentale doit être de rang deux alors que la matrice essentielle doit

satisfaire les conditions de Huang-Faugeras [Huang 89], qui sont des contraintes non-linéaires et qui sont imposées a posteriori, après l'estimation linéaire.

À partir de la matrice essentielle, on peut estimer la rotation de la caméra et sa translation à un facteur d'échelle près. Comme on le détaillera mieux dans le prochain chapitre, [Basri 98] a récemment proposé un schéma de commande très proche de l'asservissement visuel 3D en se basant sur cette estimation partielle. Le facteur d'échelle en translation étant inconnu, il est estimé en utilisant la mesure de la translation effectuée par le robot à la première itération de l'asservissement. Ensuite, la translation et la rotation de la caméra sont contrôlées exactement comme dans l'asservissement visuel 3D.

Le déplacement partiel de la caméra peut aussi être estimé à partir de la matrice d'homographie relative à un plan de référence lié à la cible [Faugeras 88, Zhang 95]. La matrice d'homographie peut être estimée en même temps que l'épipole en utilisant, par exemple, un des algorithmes présentés dans [Boufama 95, Heeger 92], ou après que l'épipole ait été estimé [Shashua 96]. Si plus de deux images d'un même objet sont disponibles, on pourra consulter [Irani 96, Irani 98, Vieville 96]. Si de plus, on a la connaissance a priori que 4 points de la cible sont coplanaires, on pourra consulter [Criminisi 98].

On démontrera par la suite que l'estimation des paramètres de mouvement est plus robuste si celle-ci est effectuée à partir d'une matrice d'homographie plutôt qu'à partir de la matrice fondamentale, notamment quand l'épipole n'est pas défini dans l'image (par exemple, si le mouvement de la caméra est une rotation pure ou si la cible est plane [Longuet-Higgins 84]). Comme les algorithmes classiques estiment la matrice d'homographie en même temps que l'épipole, le nombre d'inconnues n'est pas minimal. De plus, si on utilise l'algorithme présenté dans [Boufama 95], il existe trois configurations épipolaires où il est impossible d'extraire la matrice d'homographie.

Comme la connaissance de l'épipole n'est pas nécessaire pour l'estimation de l'homographie, on propose un algorithme similaire à celui présenté dans [Couapel 95] pour l'estimation directe de la matrice d'homographie. On verra que cette méthode est assez robuste même dans les cas particuliers. Ceci est particulièrement important en asservissement visuel car une tâche de positionnement s'achève quand le déplacement de la caméra est nul, ce qui correspond à une rotation pure nulle. De plus, comme la cible est a priori inconnue, elle peut être plane (ou tous les points observés sur la cible peuvent être proches d'un plan), ce qui correspond à un cas particulier correctement traité par une estimation directe de la matrice d'homographie.

## 2.2 Modélisation

Dans une tâche de positionnement par asservissement visuel, l'objectif est de concevoir une loi de commande afin que la caméra, initialement dans la position définie par le repère  $\mathcal{F}$ , rejoigne une position désirée définie par le repère  $\mathcal{F}^*$  (voir Figure 2.1 où les entités géomé-

triques exprimées dans le repère  $\mathcal{F}^*$  sont caractérisées par l'indice  $*$ ). Il est donc important d'établir la relation qui lie les deux projections perspectives d'un objet statique au déplacement effectué par la caméra.

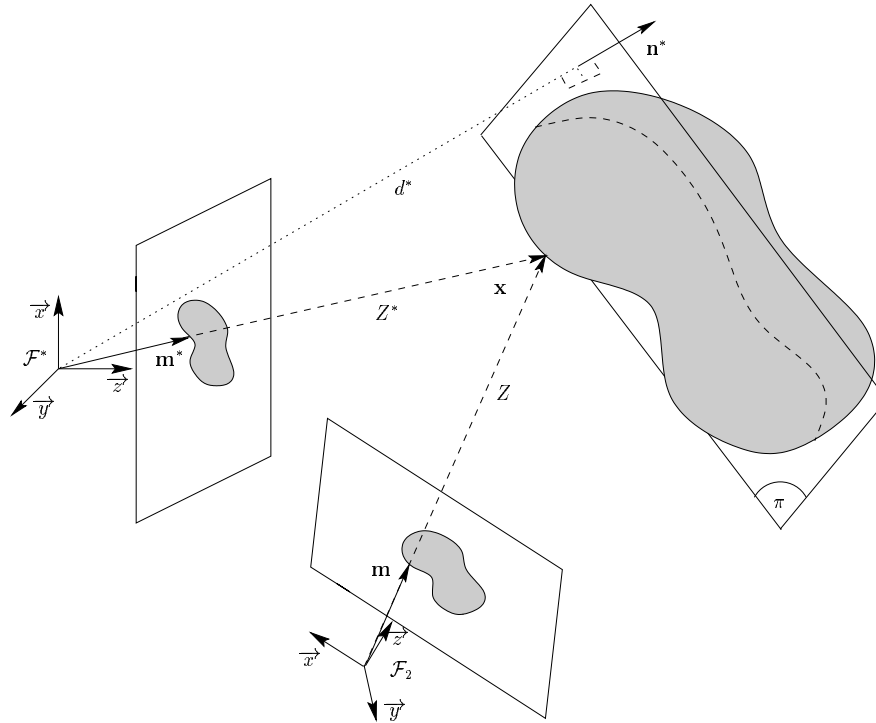


FIG. 2.1 – Déplacement de la caméra

Considérons un plan de référence  $\pi$  attaché à l'objet (voir Figure 2.1), décrit par le vecteur  $\boldsymbol{\pi}^* = [\mathbf{n}^* \quad -d^*]^T$  (on rappelle que  $\mathbf{n}^*$  est la normale au plan et  $d^*$  la distance entre l'origine de  $\mathcal{F}^*$  et le plan). On définit  $d(\mathbf{x}^*, \boldsymbol{\pi}^*) = \boldsymbol{\pi}^{*T}(\mathbf{x}^*, 1)$  la distance signée entre le point 3D, de coordonnées homogènes  $(\mathbf{x}^*, 1) = [X^* \quad Y^* \quad Z^* \quad W^*]^T$ , et le plan :

- si  $d(\mathbf{x}^*, \boldsymbol{\pi}^*) = 0$ , le point appartient à  $\pi$  ;
- si  $d(\mathbf{x}^*, \boldsymbol{\pi}^*) < 0$ , le point est entre  $\pi$  et la caméra ;
- si  $d(\mathbf{x}^*, \boldsymbol{\pi}^*) > 0$ , le point est au delà de  $\pi$ .

La coordonnée homogène  $W^*$  de  $\mathbf{x}^*$  peut donc s'écrire :

$$W^* = \frac{Z^* \mathbf{n}^{*T} \mathbf{m}^*}{d^*} - \frac{d(\mathbf{x}^*, \boldsymbol{\pi}^*)}{d^*} \quad (2.1)$$

car  $\mathbf{m}^* = \frac{1}{Z^*} \mathbf{P}^*(\mathbf{x}^*, 1) = \frac{1}{Z^*} \mathbf{x}^*$  (voir équation (1.5)). On va voir maintenant que cette représentation de la quatrième coordonnée homogène joue un rôle fondamental dans la définition de la relation entre les deux images de l'objet.

### 2.2.1 La relation entre deux images

Afin de lier les deux projections perspectives de la cible, considérons un point  $\mathbf{x}^*$  reconstruit à partir de ses coordonnées dans le plan image :

$$\begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} = Z^* \mathbf{P}^{*+} \mathbf{m}^* + \mathbf{P}^{*\perp} \quad (2.2)$$

où  $\mathbf{P}^{*+} = \mathbf{P}^{*T}(\mathbf{P}^* \mathbf{P}^{*T})^{-1} = [\mathbf{I}_3 \ \mathbf{0}]^T$  est la pseudo-inverse de la matrice de projection  $\mathbf{P}^*$  et  $\mathbf{P}^{*\perp} = [0 \ 0 \ 0 \ W^*]^T$  est le vecteur engendré par son noyau. À partir de (2.1), l'équation (2.2) peut s'écrire :

$$\begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} = Z^* \mathbf{N}_\pi^* \mathbf{m}^* + \boldsymbol{\varepsilon}_\pi^* \quad (2.3)$$

où  $\mathbf{N}_\pi^* = [\mathbf{I}_3 \ \mathbf{n}^*/d^*]^T$  et  $\boldsymbol{\varepsilon}_\pi^* = [\mathbf{0} \ \varepsilon]^T$  (avec  $\varepsilon = -d(\mathbf{x}^*, \boldsymbol{\pi}^*)/d^*$ ). À partir de l'équation  $Z\mathbf{m} = \mathbf{P}(\mathbf{x}, 1)$ , la relation entre les points projetés dans les deux images est donc :

$$Z\mathbf{m} = Z^* \mathbf{P} \mathbf{N}_\pi^* \mathbf{m}^* + \mathbf{P} \boldsymbol{\varepsilon}_\pi^* = Z^* \mathbf{H} \mathbf{m}^* + \varepsilon \mathbf{t} \quad (2.4)$$

où  $\mathbf{H} = \mathbf{P} \mathbf{N}_\pi^* = \mathbf{R} + \mathbf{t} \mathbf{n}^{*T}/d^*$  est la matrice d'homographie relative au plan de référence. Si ce dernier est le plan à l'infini  $\boldsymbol{\pi}_\infty$ , on a  $\mathbf{n}^*/d^* = \mathbf{0}$  et  $\varepsilon = 1$  ; l'équation (2.4) devient alors :

$$Z\mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + \mathbf{t} \quad (2.5)$$

où  $\mathbf{R} = \mathbf{H}_\infty$  est la matrice d'homographie relative au plan de l'infini (car elle fait correspondre entre les deux projections les points appartenant au plan de l'infini).

### 2.2.2 La contrainte épipolaire

L'équation (2.4) peut facilement être écrite dans le système de coordonnées en pixels en utilisant l'équation (1.7) :

$$Z\mathbf{p} = Z^* \mathbf{A} \mathbf{H} \mathbf{A}^{-1} \mathbf{p}^* + \varepsilon \mathbf{A} \mathbf{t} \quad (2.6)$$

On remarque qu'on peut multiplier les deux côtés de cette équation par un scalaire non nul sans perte de généralité. Pour éviter toute confusion, on écrit l'équation (2.6) sous la forme :

$$\alpha \mathbf{p} = \mathbf{G} \mathbf{p}^* + \beta \mathbf{e} \quad (2.7)$$

où  $\forall k \neq 0$  ;

- $\alpha = k Z/Z^*$  est un scalaire non nul qui peut être choisi toujours positif ;
- $\beta = k \varepsilon/Z^*$  est un scalaire qui est nul si le point appartient au plan de référence ;
- $\mathbf{G} = k \mathbf{A} \mathbf{H} \mathbf{A}^{-1}$  est l'homographie exprimée dans le système de coordonnées en pixels ;
- $\mathbf{e} = k \mathbf{A} \mathbf{t}$  est l'épipole dans l'image courante (c'est-à-dire la projection de  $\mathcal{O}^*$  dans l'image courante).

À partir de seulement deux images, et sans connaissance a priori sur la géométrie de l'objet ou sur le mouvement de la caméra, il est impossible de déterminer le facteur d'échelle  $k$ .

Considérons maintenant trois points 3D  $\mathcal{P}_i$   $\{i = 1, 2, 3\}$  de l'objet. Ces points, choisis de manière à ne pas être colinéaires dans les images, définissent un plan de référence noté  $\pi$ . La structure de la cible étant inconnue dans le cas général, tous les points pourraient être sur le même plan. On suppose pour le moment qu'il existe un certain nombre de points n'appartenant pas au plan de référence et on verra dans la suite comment traiter le cas d'un objet planaire. Si un point  $\mathcal{P}_j$  de l'objet n'appartient pas à  $\pi$ , la droite  $(\mathcal{O}^*\mathcal{P}_j)$  et le plan  $\pi$  s'intersectent au point 3D noté  $\mathcal{P}'_j$  (voir la Figure 2.2).

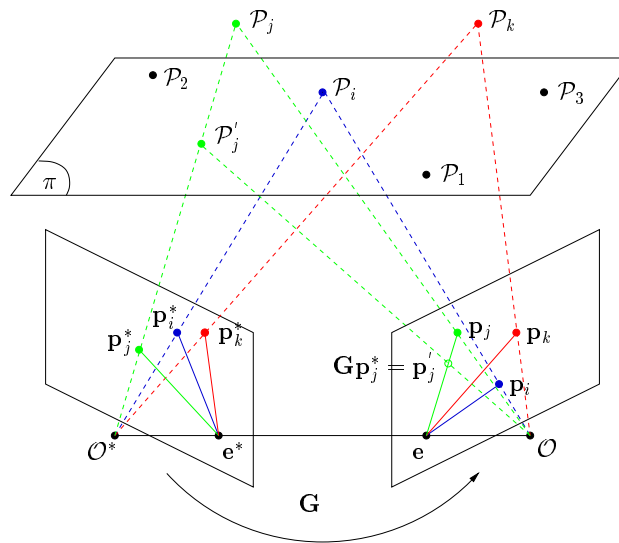


FIG. 2.2 – Contrainte épipolaire et parallaxe virtuelle

$\mathcal{P}_j$  et  $\mathcal{P}'_j$  se projettent au même point  $\mathbf{p}_j^*$  dans l'image désirée et en deux points différents ( $\mathbf{p}_j$  et le point virtuel  $\mathbf{p}'_j = \mathbf{G}\mathbf{p}_j^*$ ) dans l'image courante (ce qu'on appelle l'effet de parallaxe). La droite définie par  $\mathbf{p}_j$  et  $\mathbf{p}'_j$  est représentée, en coordonnées projectives, par  $\mathbf{l}_j = \mathbf{p}_j \wedge \mathbf{p}'_j$  [Faugeras 93], soit encore :

$$\mathbf{l}_j = \mathbf{p}_j \wedge \mathbf{G}\mathbf{p}_j^* \quad (2.8)$$

Les quatre points  $\mathcal{O}$ ,  $\mathcal{O}^*$ ,  $\mathcal{P}_j$  et  $\mathcal{P}'_j$  étant coplanaires, les points  $\mathbf{p}_j$ ,  $\mathbf{p}'_j$  et l'épipôle  $\mathbf{e}$  sont donc colinéaires et l'on a :

$$\mathbf{l}_j^T \mathbf{e}_j = 0 \quad (2.9)$$

En utilisant l'équation (2.8), on a donc :

$$\mathbf{p}_j^T \mathbf{F} \mathbf{p}_j^* = 0 \quad (2.10)$$

où  $\mathbf{F} = [\mathbf{e}]_{\times} \mathbf{G}$  est la matrice fondamentale ( $[\mathbf{e}]_{\times}$  est la matrice de préproduit vectoriel associée au vecteur  $\mathbf{e}$ ). Dans le cas général,  $\mathbf{F}$  est de rang 2 et  $\det(\mathbf{F}) = 0$ , ce qui implique une contrainte non-linéaire sur les éléments de  $\mathbf{F}$  [Luong 96].

## 2.3 Estimation

Dans cette section, nous décrivons les algorithmes d'estimation linéaire de la géométrie épipolaire les plus utilisés dans la littérature, à savoir basés sur :

- l'estimation de la matrice fondamentale
- l'estimation conjointe de l'épipole et d'une matrice d'homographie
- l'estimation d'une matrice d'homographie

Nous proposons au paragraphe 2.3.3 une nouvelle méthode d'estimation d'une matrice d'homographie. Il faut souligner que cette matrice, liant généralement des points appartenant à un même plan, sera estimée ici à partir de points non coplanaires grâce à l'effet de parallaxe qu'on vient de décrire.

### 2.3.1 Estimation linéaire de la matrice fondamentale

L'approche classique pour estimer la géométrie épipolaire est connue sous le nom d'algorithme des huit points [LonguetHiggins 81, Hartley 97]. En effet, il est possible de simplifier le problème en effectuant une estimation linéaire des éléments de la matrice fondamentale à partir de l'équation (2.10) :

$$u_j^* u_j f_{11} + v_j^* u_j f_{12} + u_j f_{13} + u_j^* v_j f_{21} + v_j^* v_j f_{22} + u_j^* f_{31} + v_j f_{23} + v_j^* f_{32} + f_{33} = 0 \quad (2.11)$$

Si on considère par exemple une image de taille  $500 \times 500$  pixels, on remarque que les coefficients associés à  $f_{11}, f_{12}, f_{21}, f_{22}$  sont de l'ordre de 250000, les coefficients associés à  $f_{13}, f_{23}, f_{31}, f_{32}$  de l'ordre de 500, alors que le coefficient associé à  $f_{33}$  est toujours égal à 1. L'importance d'une normalisation des données a été mise en évidence par [Hartley 97]. En effet, afin d'obtenir une bonne estimation, il est nécessaire d'effectuer un changement de coordonnées qui permette d'obtenir des coefficients ayant à peu près tous le même ordre de grandeur.

L'équation (2.10) est valable pour chaque couple de points  $(\mathbf{p}_j, \mathbf{p}_j^*)$ . Si l'on dispose de  $n$  couples, il est donc possible d'obtenir un système linéaire :

$$\mathbf{C}_f \mathbf{f} = 0 \quad (2.12)$$

où  $\mathbf{C}_f$  est la matrice des coefficients de dimension  $(n \times 9)$  et  $\mathbf{f}$  est un vecteur contenant les 9 éléments de  $\mathbf{F}$ . Le système (2.12) est homogène et, comme  $\mathbf{F}$  est définie à un facteur d'échelle près, un minimum de 8 couples de points est nécessaire pour résoudre (2.12). En présence de bruit, le problème d'estimation linéarisée peut être écrit :

$$\min_{\mathbf{f}} \|\mathbf{C}_f \mathbf{f}\| \quad \text{sous contrainte} \quad \|\mathbf{f}\| = 1 \quad (2.13)$$

La solution de ce problème est obtenue à partir de la décomposition en valeurs singulières (SVD) de la matrice  $\mathbf{C}_f^T \mathbf{C}_f = \mathbf{V} \mathbf{S} \mathbf{V}^T$ , où  $\mathbf{S}$  est une matrice diagonale. La solution  $\bar{\mathbf{f}}$  du système est la colonne de  $\mathbf{V}$  correspondant à la valeur singulière minimale (0 en absence de bruit).

On remarque que, si l'épipole n'est pas défini dans l'image (par exemple si le mouvement est une rotation pure ou si l'objet est plan [LonguetHiggins 84]), la matrice fondamentale est aussi indéfinie, ce qui implique une estimation instable autour de ces cas particuliers. On verra par la suite que la méthode que nous proposons est capable de fournir de meilleurs résultats malgré ces problèmes.

L'algorithme précédent ne tient pas compte de la contrainte que la matrice fondamentale doit être de rang 2. Cette contrainte est généralement prise en compte en utilisant un algorithme non-linéaire avec comme première estimée le résultat du système linéaire [Deriche 94, Luong 96]. Comme notre objectif est de réaliser des tâches robotiques à la cadence vidéo, les méthodes non-linéaires ne semblent pas adaptées. Toutefois, on comparera les résultats de notre méthode avec ceux obtenus en utilisant la méthode non-linéaire décrite dans [Deriche 94]. On verra aussi plus loin que, si les paramètres intrinsèques de la caméra sont connus, la contrainte  $\det(\mathbf{F}) = 0$  peut être prise en compte a posteriori en calculant la matrice essentielle.

### 2.3.2 Estimation de l'épipole et d'une matrice d'homographie

Afin de simplifier l'estimation de la matrice  $\mathbf{F}$ , Boufama et al. [Boufama 95] effectuent un changement de coordonnées projectives en utilisant 4 points dans chaque image. Ces points sont choisis de telle manière que jamais trois d'entre eux ne soient colinéaires dans les images. Soient  $\mathbf{M} = \mathbf{M}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4)$  et  $\mathbf{M}^* = \mathbf{M}^*(\mathbf{p}_1^*, \mathbf{p}_2^*, \mathbf{p}_3^*, \mathbf{p}_4^*)$  les matrices de changement de coordonnées de dimensions  $(3 \times 3)$ .

Les points image  $\tilde{\mathbf{p}}_j = [\tilde{u}_j \ \tilde{v}_j \ \tilde{w}_j]^T$  et  $\tilde{\mathbf{p}}_j^* = [\tilde{u}_j^* \ \tilde{v}_j^* \ \tilde{w}_j^*]^T$  dans le nouveau système de coordonnées sont :

$$\begin{cases} \tilde{\mathbf{p}}_j &= \mathbf{M}^{-1} \mathbf{p}_j \\ \tilde{\mathbf{p}}_j^* &= \mathbf{M}^{*-1} \mathbf{p}_j^* \end{cases} \quad (2.14)$$

En utilisant les trois premiers points, on peut écrire les matrices de changement de coordonnées sous la forme :

$$\begin{cases} \mathbf{M} &= [\mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_3] \\ \mathbf{M}^* &= [\mathbf{p}_1^* \ \mathbf{p}_2^* \ \mathbf{p}_3^*] \end{cases} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \\ \lambda_1^* & 0 & 0 \\ 0 & \lambda_2^* & 0 \\ 0 & 0 & \lambda_3^* \end{bmatrix} \quad (2.15)$$

où les scalaires inconnus  $\lambda_i$  et  $\lambda_i^*$  peuvent être calculés, à un facteur d'échelle près, en utilisant



le quatrième point :

$$\left\{ \begin{array}{l} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \\ \begin{bmatrix} \lambda_1^* \\ \lambda_2^* \\ \lambda_3^* \end{bmatrix} \end{array} \right. = \begin{array}{l} \lambda_4 [ \mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_3 ]^{-1} \mathbf{p}_4 \\ \lambda_4^* [ \mathbf{p}_1^* \ \mathbf{p}_2^* \ \mathbf{p}_3^* ]^{-1} \mathbf{p}_4^* \end{array} \quad (2.16)$$

En choisissant  $[ \tilde{\mathbf{p}}_1 \ \tilde{\mathbf{p}}_2 \ \tilde{\mathbf{p}}_3 ] = [ \tilde{\mathbf{p}}_1^* \ \tilde{\mathbf{p}}_2^* \ \tilde{\mathbf{p}}_3^* ] = \mathbf{I}_3$  pour les trois premiers points, la matrice d'homographie  $\tilde{\mathbf{G}}$ , relative au plan  $\pi$  défini par ces trois points, est diagonale quand elle est exprimée dans le nouveau système de coordonnées :

$$\tilde{\mathbf{G}} = \mathbf{M}^{-1} \mathbf{G} \mathbf{M}^* = \begin{bmatrix} \tilde{g}_u & 0 & 0 \\ 0 & \tilde{g}_v & 0 \\ 0 & 0 & \tilde{g}_w \end{bmatrix} \quad (2.17)$$

La matrice fondamentale dans le nouveau système de coordonnées s'écrit de la manière suivante :

$$\tilde{\mathbf{F}} = [\tilde{\mathbf{e}}]_{\times} \tilde{\mathbf{G}} = \begin{bmatrix} 0 & -\tilde{e}_z \tilde{g}_v & \tilde{e}_y \tilde{g}_w \\ \tilde{e}_z \tilde{g}_u & 0 & -\tilde{e}_x \tilde{g}_w \\ -\tilde{e}_y \tilde{g}_u & \tilde{e}_x \tilde{g}_v & 0 \end{bmatrix} \quad (2.18)$$

où  $\tilde{\mathbf{e}} = \mathbf{M}^{-1} \mathbf{e}$  est l'épipoles exprimé dans le nouveau système de coordonnées. Sous cette nouvelle forme, on voit que la matrice fondamentale est toujours de rang deux. La contrainte non linéaire sur les éléments de  $\mathbf{F}$  a donc pu être introduite par un simple changement de base. Toutefois, ce changement rend l'algorithme dépendant du choix des points qui constituent la nouvelle base.

L'équation (2.10) s'écrit finalement :

$$\tilde{v}_j^* \tilde{u}_j \tilde{f}_{12} + \tilde{w}_j^* \tilde{u}_j \tilde{f}_{13} + \tilde{u}_j^* \tilde{v}_j \tilde{f}_{21} + \tilde{u}_j^* \tilde{w}_j \tilde{f}_{31} + \tilde{w}_j^* \tilde{v}_j \tilde{f}_{23} + \tilde{v}_j^* \tilde{w}_j \tilde{f}_{32} = 0 \quad (2.19)$$

En injectant l'équation (2.18), on obtient une équation polynomiale de degré deux à quatre inconnues (deux inconnues pour l'épipoles et deux pour la matrice d'homographie diagonale puisque ces entités sont définies à un facteur d'échelle près). Il est possible d'écrire cette équation sous la forme suivante [Boufama 95] :

$$\mathbf{C}_{\tilde{\mathbf{f}}} \tilde{\mathbf{f}} = 0 \quad (2.20)$$

où  $\tilde{\mathbf{f}} = [ -\tilde{e}_z \tilde{g}_v \ \tilde{e}_y \tilde{g}_w \ \tilde{e}_z \tilde{g}_u \ -\tilde{e}_x \tilde{g}_w \ -\tilde{e}_y \tilde{g}_u \ \tilde{e}_x \tilde{g}_v ]^T$ . Le nouveau système est linéaire et homogène à 6 inconnues. Au moins cinq points n'appartenant pas à  $\pi$  sont donc nécessaires pour résoudre de manière linéaire le problème. Si  $m$  ( $m \geq 5$ ) points sont disponibles, la matrice  $\mathbf{C}_{\tilde{\mathbf{f}}}$  des coefficients est de dimension  $(m \times 6)$ . Le problème est résolu en considérant la SVD de la matrice  $\mathbf{C}_{\tilde{\mathbf{f}}}^T \mathbf{C}_{\tilde{\mathbf{f}}} = \mathbf{V} \mathbf{S} \mathbf{V}^T$ . Encore une fois, la solution  $\tilde{\mathbf{f}}$  du système est la colonne

de  $\mathbf{V}$  correspondant à la valeur singulière minimale (0 en absence de bruit). Après avoir estimé le vecteur  $\tilde{\mathbf{f}}$ , les inconnues originales peuvent être obtenues en effectuant le changement de coordonnées inverse.

Malheureusement, il existe trois cas singuliers où la matrice d'homographie ne peut pas être estimée avec cet algorithme :

- si  $\tilde{\mathbf{e}} = [1 \ 0 \ 0]^T$ , seul  $\tilde{g}_v/\tilde{g}_w$  peut être estimé à partir de  $f_4$  et  $f_6$  ;
- si  $\tilde{\mathbf{e}} = [0 \ 1 \ 0]^T$ , seul  $\tilde{g}_u/\tilde{g}_w$  peut être estimé à partir de  $f_2$  et  $f_5$  ;
- si  $\tilde{\mathbf{e}} = [0 \ 0 \ 1]^T$ , seul  $\tilde{g}_u/\tilde{g}_v$  peut être estimé à partir de  $f_1$  et  $f_3$ .

Dans ces cas de figure, l'estimation de la matrice d'homographie est donc impossible car nous ne pouvons pas obtenir à la fois  $\tilde{g}_u$ ,  $\tilde{g}_v$  et  $\tilde{g}_w$ . Si ces cas peuvent être détectés, un autre algorithme peut être alors utilisé. Toutefois, en présence de bruit, la détection de ces cas particuliers est un problème très délicat et, en cas de fausse détection, l'estimation ne sera pas précise car l'estimation de la valeur zéro pour 4 des 6 coefficients de  $\mathbf{f}$  est très sensible aux erreurs numériques.

Le principal avantage de cet algorithme est que, tout en dégénéralisant dans les cas particuliers ainsi que dans les cas dégénérés (rotation pure, objet plan), il fournit une estimation de la matrice d'homographie qui est toujours définie, contrairement à la matrice fondamentale. Il faut toutefois remarquer que, dans cet algorithme, l'estimation de la matrice d'homographie dépend de celle de l'épipole, et le nombre d'inconnues n'est donc pas minimal. Pour ces raisons, on présente maintenant une méthode pour déterminer directement la matrice d'homographie sans avoir besoin d'estimer l'épipole.

### 2.3.3 Estimation linéaire d'une homographie à partir de points non coplanaires

L'approche, similaire à celle proposée dans [Couapel 95], est basée sur la contrainte que toutes les droites épipolaires s'intersectent à l'épipole. En effet, pour chaque ensemble de trois droites épipolaires (2.8), on a :

$$\left| \mathbf{l}_j \ \mathbf{l}_k \ \mathbf{l}_l \right| = \left| \mathbf{p}_j \wedge \mathbf{G}\mathbf{p}_j^* \ \mathbf{p}_k \wedge \mathbf{G}\mathbf{p}_k^* \ \mathbf{p}_l \wedge \mathbf{G}\mathbf{p}_l^* \right| = 0 \quad (2.21)$$

Il faut remarquer que cette équation, qui exprime le fait que trois droites s'intersectent en un même point, est la duale de l'équation (2.10) qui exprime le fait que trois points sont alignés (par conséquent les cas dégénérés sont les mêmes). Toutefois, l'équation (2.21) est non-linéaire par rapport aux éléments de la matrice d'homographie. De plus, cette équation est vérifiée par n'importe quelle matrice d'homographie. Afin d'indiquer quelle est la matrice d'homographie que l'on désire estimer, et pour simplifier le calcul de  $\mathbf{G}$ , un changement particulier de coordonnées projectives est effectué.

Couapel [Couapel 95], comme Boufama [Boufama 95], a proposé de réaliser un changement de coordonnées projectives en utilisant quatre points dans chaque image. Toutefois, au

lieu de choisir un quatrième point quelconque dans l'image, Couapel utilise le centre de gravité des trois premiers points. En choisissant à nouveau  $[\tilde{\mathbf{p}}_1 \ \tilde{\mathbf{p}}_2 \ \tilde{\mathbf{p}}_3] = [\tilde{\mathbf{p}}_1^* \ \tilde{\mathbf{p}}_2^* \ \tilde{\mathbf{p}}_3^*] = \mathbf{I}_3$ , la matrice d'homographie  $\tilde{\mathbf{G}}$  dans le nouveau système de coordonnées est diagonale :  $\tilde{\mathbf{G}} = \mathbf{M}^{-1}\mathbf{G}\mathbf{M}^* = \text{diag}(\tilde{g}_u, \tilde{g}_v, \tilde{g}_w)$ . L'équation (2.21) peut donc s'écrire dans le nouveau système de coordonnées :

$$|\tilde{\mathbf{p}}_j \wedge \tilde{\mathbf{G}}\tilde{\mathbf{p}}_j^* \quad \tilde{\mathbf{p}}_k \wedge \tilde{\mathbf{G}}\tilde{\mathbf{p}}_k^* \quad \tilde{\mathbf{p}}_l \wedge \tilde{\mathbf{G}}\tilde{\mathbf{p}}_l^*| = 0 \quad (2.22)$$

Après calcul, (2.22) peut être écrite sous la forme :

$$\mathbf{A}_{\tilde{g}}\mathbf{y} = \mathbf{b}_{\tilde{g}} \quad (2.23)$$

où  $\mathbf{A}_{\tilde{g}}$  est la matrice des coefficients,  $\mathbf{b}_{\tilde{g}}$  est un vecteur et, après avoir fixé  $\tilde{g}_w = 1$  :

$$\mathbf{y} = [\tilde{g}_u \quad \tilde{g}_v \quad \tilde{g}_u/\tilde{g}_v \quad \tilde{g}_v/\tilde{g}_u \quad 1/\tilde{g}_v \quad 1/\tilde{g}_u]^T \quad (2.24)$$

La solution du système linéaire est  $\bar{\mathbf{y}} = \mathbf{A}_{\tilde{g}}^+\mathbf{b}_{\tilde{g}}$  (où  $\mathbf{A}_{\tilde{g}}^+$  est la pseudo-inverse de  $\mathbf{A}_{\tilde{g}}$ ). Par cette méthode, on obtient les inconnues  $\tilde{g}_u = \bar{y}_1$ ,  $\tilde{g}_v = \bar{y}_2$  et les paramètres de contrôle (appelé *erreurs de cohérence* sur la qualité de la solution)  $\bar{y}_1\bar{y}_6 = \bar{y}_2\bar{y}_5 = \bar{y}_3\bar{y}_4$  qui doivent être égaux à 1. Malheureusement, ces contraintes non-linéaires ne sont pas nécessairement satisfaites en présence de mesures bruitées. Aussi propose-t-on la méthode décrite à présent.

### 2.3.3.1 Changement de coordonnées et normalisation des données

À la différence de Boufama [Boufama 95] et Couapel [Couapel 95], les matrices de changement de coordonnées  $\mathbf{M}$  et  $\mathbf{M}^*$  sont construites en utilisant seulement trois points de référence dans l'image. Dans ce cas, les matrices de transformation sont  $\mathbf{M} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_3]$  et  $\mathbf{M}^* = [\mathbf{p}_1^* \ \mathbf{p}_2^* \ \mathbf{p}_3^*]$  (on rappelle que n'importe quelle matrice non singulière de dimension  $(3 \times 3)$  peut être utilisée pour réaliser un changement de coordonnées dans  $\mathbb{P}^2$ ).

Le choix des points de référence influence les résultats de l'estimation, surtout dans le cas d'une fausse mise en correspondance. Sous l'hypothèse que tous les points soient correctement appariés, on a intérêt à choisir les trois points qui maximisent l'aire des triangles associés dans les deux images afin d'obtenir une estimation la plus robuste possible. Ce critère de choix peut être effectué de manière automatique. Toutefois, on verra dans les résultats expérimentaux qu'il est également possible d'obtenir une estimation correcte en utilisant trois points qui ne sont pas très écartés dans les deux images.

En choisissant une fois de plus  $[\tilde{\mathbf{p}}_1 \ \tilde{\mathbf{p}}_2 \ \tilde{\mathbf{p}}_3] = [\tilde{\mathbf{p}}_1^* \ \tilde{\mathbf{p}}_2^* \ \tilde{\mathbf{p}}_3^*] = \mathbf{I}_3$ , la matrice d'homographie  $\tilde{\mathbf{G}}$  dans le nouveau système de coordonnées est diagonale :  $\tilde{\mathbf{G}} = \mathbf{M}^{-1}\mathbf{G}\mathbf{M}^* = \text{diag}(\tilde{g}_u, \tilde{g}_v, \tilde{g}_w)$ . Il faut remarquer que le changement de coordonnées a aussi l'effet de normaliser les données. Comme on l'a déjà indiqué, cette normalisation est importante pour obtenir une estimation robuste dans l'espace projectif [Hartley 97]. En effet, la matrice de changement de coordonnées est donnée par :

$$\mathbf{M}^{-1} = \frac{1}{\det(\mathbf{M})} \begin{bmatrix} v_2 - v_3 & u_3 - u_2 & u_2v_3 - u_3v_2 \\ v_3 - v_1 & u_1 - u_3 & v_1u_3 - u_1v_3 \\ v_1 - v_2 & u_2 - u_1 & u_1v_2 - v_1u_2 \end{bmatrix} \quad (2.25)$$

où  $\det(\mathbf{M}) = u_1v_2 - u_1v_3 - v_1u_2 + v_1u_3 + u_2v_3 - u_3v_2 \neq 0$  si les trois points ne sont pas alignés. Un point dans le nouveau système de coordonnées est donc :

$$\tilde{\mathbf{p}} = \mathbf{M}^{-1}\mathbf{p} = \frac{1}{\det(\mathbf{M})} \begin{bmatrix} (v_2 - v_3)u + (u_3 - u_2)v + u_2v_3 - u_3v_2 \\ (v_3 - v_1)u + (u_1 - u_3)v + v_1u_3 - u_1v_3 \\ (v_1 - v_2)u + (u_2 - u_1)v + u_1v_2 - v_1u_2 \end{bmatrix} \quad (2.26)$$

et les coordonnées du point ont maintenant toutes le même ordre de grandeur.

On utilise ensuite l'équation (2.22), qui est homogène et polynomiale de degré trois. Contrairement à (2.18), l'équation (2.22) ne dépend pas de l'épipole et contient seulement trois inconnues. Après calcul, (2.22) peut être écrite sous la forme :

$$\mathbf{C}_{\tilde{g}}\mathbf{x} = 0 \quad (2.27)$$

où :

$$\mathbf{x} = [ \tilde{g}_u^2\tilde{g}_v \quad \tilde{g}_v^2\tilde{g}_u \quad \tilde{g}_u^2\tilde{g}_w \quad \tilde{g}_v^2\tilde{g}_w \quad \tilde{g}_w^2\tilde{g}_u \quad \tilde{g}_w^2\tilde{g}_v \quad \tilde{g}_u\tilde{g}_v\tilde{g}_w ]^T \quad (2.28)$$

et où la  $j$ -ème ligne de la matrice des coefficients  $\mathbf{C}_{\tilde{g}}$  (qui a une perte de rang dans les cas dégénérés mais pas dans les cas singuliers de la méthode proposée dans [Boufama 95]) peut être écrite en fonction des coordonnées des points dans l'image :

$$\begin{aligned} c_{1j} &= \tilde{w}_i\tilde{w}_j\tilde{v}_k\tilde{u}_k^*(\tilde{u}_j^*\tilde{v}_i^* - \tilde{u}_i^*\tilde{v}_j^*) + \tilde{w}_i\tilde{w}_k\tilde{v}_j\tilde{u}_j^*(\tilde{u}_i^*\tilde{v}_k^* - \tilde{u}_k^*\tilde{v}_i^*) + \tilde{w}_j\tilde{w}_k\tilde{v}_i\tilde{u}_i^*(\tilde{u}_k^*\tilde{v}_j^* - \tilde{u}_j^*\tilde{v}_k^*) \\ c_{2j} &= \tilde{w}_i\tilde{w}_j\tilde{u}_k\tilde{v}_k^*(\tilde{u}_i^*\tilde{v}_j^* - \tilde{u}_j^*\tilde{v}_i^*) + \tilde{w}_i\tilde{w}_k\tilde{u}_j\tilde{v}_j^*(\tilde{u}_k^*\tilde{v}_i^* - \tilde{u}_i^*\tilde{v}_k^*) + \tilde{w}_j\tilde{w}_k\tilde{u}_i\tilde{v}_i^*(\tilde{u}_j^*\tilde{v}_k^* - \tilde{u}_k^*\tilde{v}_j^*) \\ c_{3j} &= \tilde{v}_i\tilde{v}_k\tilde{w}_j\tilde{u}_j^*(\tilde{u}_i^*\tilde{w}_k^* - \tilde{u}_k^*\tilde{w}_i^*) + \tilde{v}_i\tilde{v}_j\tilde{w}_k\tilde{u}_k^*(\tilde{u}_j^*\tilde{w}_i^* - \tilde{u}_i^*\tilde{w}_j^*) + \tilde{v}_j\tilde{v}_k\tilde{w}_i\tilde{u}_i^*(\tilde{u}_k^*\tilde{w}_j^* - \tilde{u}_j^*\tilde{w}_k^*) \\ c_{4j} &= \tilde{u}_i\tilde{u}_k\tilde{w}_j\tilde{v}_j^*(\tilde{v}_i^*\tilde{w}_k^* - \tilde{v}_k^*\tilde{w}_i^*) + \tilde{u}_i\tilde{u}_j\tilde{w}_k\tilde{v}_k^*(\tilde{v}_j^*\tilde{w}_i^* - \tilde{v}_i^*\tilde{w}_j^*) + \tilde{u}_j\tilde{u}_k\tilde{w}_i\tilde{v}_i^*(\tilde{v}_k^*\tilde{w}_j^* - \tilde{v}_j^*\tilde{w}_k^*) \\ c_{5j} &= \tilde{v}_j\tilde{v}_k\tilde{u}_i\tilde{w}_i^*(\tilde{u}_j^*\tilde{w}_k^* - \tilde{u}_k^*\tilde{w}_j^*) + \tilde{v}_i\tilde{v}_k\tilde{u}_j\tilde{w}_j^*(\tilde{u}_k^*\tilde{w}_i^* - \tilde{u}_i^*\tilde{w}_k^*) + \tilde{v}_i\tilde{v}_j\tilde{u}_k\tilde{w}_k^*(\tilde{u}_i^*\tilde{w}_j^* - \tilde{u}_j^*\tilde{w}_i^*) \\ c_{6j} &= \tilde{u}_j\tilde{u}_k\tilde{v}_i\tilde{w}_i^*(\tilde{v}_j^*\tilde{w}_k^* - \tilde{v}_k^*\tilde{w}_j^*) + \tilde{u}_i\tilde{u}_k\tilde{v}_j\tilde{w}_j^*(\tilde{v}_k^*\tilde{w}_i^* - \tilde{v}_i^*\tilde{w}_k^*) + \tilde{u}_i\tilde{u}_j\tilde{v}_k\tilde{w}_k^*(\tilde{v}_i^*\tilde{w}_j^* - \tilde{v}_j^*\tilde{w}_i^*) \\ c_{7j} &= \tilde{u}_i\tilde{v}_k\tilde{w}_j(\tilde{u}_k^*\tilde{v}_j^*\tilde{w}_i^* - \tilde{u}_j^*\tilde{v}_i^*\tilde{w}_k^*) + \tilde{u}_k\tilde{v}_i\tilde{w}_j(\tilde{u}_j^*\tilde{v}_k^*\tilde{w}_i^* - \tilde{u}_i^*\tilde{v}_j^*\tilde{w}_k^*) + \tilde{u}_i\tilde{v}_j\tilde{w}_k(\tilde{u}_k^*\tilde{v}_i^*\tilde{w}_j^* - \tilde{u}_j^*\tilde{v}_k^*\tilde{w}_i^*) + \\ &+ \tilde{u}_j\tilde{v}_i\tilde{w}_k(\tilde{u}_i^*\tilde{v}_k^*\tilde{w}_j^* - \tilde{u}_k^*\tilde{v}_i^*\tilde{w}_j^*) + \tilde{u}_k\tilde{v}_j\tilde{w}_i(\tilde{u}_j^*\tilde{v}_i^*\tilde{w}_k^* - \tilde{u}_i^*\tilde{v}_k^*\tilde{w}_j^*) + \tilde{u}_j\tilde{v}_k\tilde{w}_i(\tilde{u}_i^*\tilde{v}_j^*\tilde{w}_k^* - \tilde{u}_k^*\tilde{v}_i^*\tilde{w}_j^*) \end{aligned}$$

Il existe  $n!/(6(n-3)!)$  possibilités de choisir trois droites épipolaires différentes dans un ensemble de  $n$  droites épipolaires (une droite pour chaque point dans l'image). On obtient donc  $m = n!/(6(n-3)!)$  équations à sept inconnues. Au moins huit points (les trois points de référence et cinq points supplémentaires) sont donc nécessaires pour résoudre le problème, exactement comme dans les algorithmes décrits précédemment. La matrice des coefficients  $\mathbf{C}_{\tilde{g}}$  est de dimension  $(m \times 7)$ . Le problème est une fois de plus résolu en effectuant la SVD de la matrice  $\mathbf{C}_{\tilde{g}}^T\mathbf{C}_{\tilde{g}}$ .

### 2.3.3.2 Réintroduction des contraintes non linéaires

Finalement, les inconnues originales peuvent être obtenues en résolvant le système linéaire homogène suivant :

$$\begin{bmatrix} -\bar{x}_2 & \bar{x}_5 & -\bar{x}_7 & \bar{x}_7 & -\bar{x}_4 & \bar{x}_4 & -\bar{x}_6 & \bar{x}_6 & 0 \\ \bar{x}_1 & 0 & \bar{x}_3 & 0 & \bar{x}_7 & 0 & \bar{x}_5 & 0 & -\bar{x}_6 \\ 0 & -\bar{x}_3 & 0 & -\bar{x}_1 & 0 & -\bar{x}_2 & 0 & -\bar{x}_7 & \bar{x}_3 \end{bmatrix}^T \begin{bmatrix} \tilde{g}_u \\ \tilde{g}_v \\ \tilde{g}_w \end{bmatrix} = 0 \quad (2.29)$$

Cette deuxième estimation linéaire a l'effet d'imposer a posteriori les contraintes non linéaires sur les éléments du vecteur solution. Contrairement à l'algorithme proposé par Boufama, la matrice d'homographie peut être estimée plus précisément car la dimension du problème est réduite (l'estimation de l'épipole est séparée de l'estimation de l'homographie). On verra dans la section 2.5 que cette méthode fournit effectivement des résultats plus précis.

Il faut aussi souligner que, même si les cas particuliers pour le calcul de la géométrie épipolaire sont les mêmes que si on calcule la matrice fondamentale ou la matrice d'homographie, cette dernière est toujours définie. Dans le cas où l'épipole est indéfini, Torr et al. [Torr 98] ont récemment proposé d'estimer des matrices d'homographie afin d'obtenir de manière robuste la structure d'une scène dans une séquence d'images. Cela confirme la validité de notre approche.

Considérons par exemple le cas d'une rotation pure. La solution du système (2.12) est  $\mathbf{f} = \mathbf{0}$ . En absence de bruit, ce cas particulier peut être détecté et on peut utiliser un algorithme différent. Toutefois, en présence de bruit, la matrice fondamentale est estimée en imposant la contrainte  $\|\mathbf{f}\| = 1$  (car  $\mathbf{f}$  est une colonne d'une matrice orthonormale). Sans une détection correcte du cas particulier, ceci peut conduire à des résultats particulièrement instables car la bonne solution devrait satisfaire la contrainte  $\|\mathbf{f}\| = 0$ .

Au contraire, les solutions des systèmes (2.27) et (2.29) satisfont toujours la contrainte  $\|\mathbf{x}\| = 1$  et  $\|\tilde{\mathbf{g}}\| = 1$  respectivement. Ces contraintes, imposées en effectuant la décomposition en valeurs singulières de la matrice des coefficients, sont vérifiées même dans les cas particuliers. En présence de bruit, l'estimation du déplacement autour de ces configurations particulières sera donc plus précise si elle est calculée à partir de la matrice d'homographie plutôt qu'à partir de la matrice fondamentale. Ceci est particulièrement important en asservissement visuel, car une tâche de positionnement est réalisée quand le déplacement de la caméra est nul, ce qui correspond à une rotation pure dont la matrice de rotation est égale à l'identité.

Considérons maintenant le cas d'une cible plane. N'importe quel point dans l'image peut être choisi comme épipole. N'importe quel vecteur  $\mathbf{f}$  est donc une solution du système (2.12). En absence de bruit, ce cas particulier peut une nouvelle fois être détecté et un autre algorithme peut être utilisé. Toutefois, en présence de bruit, n'importe quelle solution peut être prise comme la bonne solution. Au contraire, comme l'objet est plan, il existe seulement une matrice d'homographie relative au plan des trois points de référence choisis. La solution obtenue en présence de bruit sera donc une approximation de la matrice d'homographie réelle même dans ce cas particulier. Ceci est très important car il est souhaitable de pouvoir obtenir l'estimation du déplacement de la caméra sans aucune connaissance a priori sur la géométrie de la cible.

### 2.3.4 Estimation d'une homographie à partir de points ou droites coplanaires

Nous décrivons à présent une méthode purement linéaire que l'on peut utiliser si l'on dispose de points ou de droites coplanaires. Si cette connaissance a priori est disponible, la matrice d'homographie peut en effet être estimée très simplement :

- **Points** : pour tous les points appartenant au plan de référence, l'équation (2.7) s'écrit sous la forme suivante :

$$\alpha_i \mathbf{p}_i = \mathbf{G}\mathbf{p}_i^* \quad \{i = 1, 2, 3, \dots, p\} \quad (2.30)$$

puisque alors  $\beta = 0$ . On rappelle que la matrice d'homographie  $\mathbf{G}$  est définie à un facteur d'échelle près. Un des éléments de  $\mathbf{G}$  ou son déterminant peut être donc fixé égal à un sans perte de généralité. On obtient donc un système linéaire à  $n + 8$  inconnues et  $3n$  équations, que l'on peut résoudre si l'on dispose d'au moins 4 points.

- **Droites** : l'équation de ces droites peut être obtenue à partir de points ou estimée directement dans l'image. La droite  $\mathbf{l}_i$  passant par deux points  $\mathbf{p}_i$  et  $\mathbf{p}_0$  s'écrit :

$$\mathbf{l}_i = \mathbf{p}_i \wedge \mathbf{p}_0 \quad (2.31)$$

Comme les deux points sont coplanaires, ils sont liés par la matrice d'homographie, et cette équation devient :

$$\mathbf{l}_i = \mathbf{G}\mathbf{p}_i^* \wedge \mathbf{G}\mathbf{p}_0^* \quad (2.32)$$

et, comme  $[\mathbf{G}\mathbf{v}]_{\times} = \det(\mathbf{G})\mathbf{G}^{-T} [\mathbf{v}]_{\times} \mathbf{G}^{-1}$ , on a :

$$\mathbf{l}_i = \det(\mathbf{G})\mathbf{G}^{-T}\mathbf{p}_i^* \wedge \mathbf{p}_0^* \quad (2.33)$$

Si on note  $\mathbf{l}_i^* = \mathbf{p}_i^* \wedge \mathbf{p}_0^*$  l'équation de la droite dans l'image désirée, la relation entre les deux droites s'écrit :

$$\alpha'_i \mathbf{l}_i = \mathbf{G}^{-T}\mathbf{l}_i^* \quad \{i = 1, 2, 3, \dots, l\} \quad (2.34)$$

où  $\alpha'_i$  est un scalaire non nul et  $\mathbf{G}^{-T}$  est la matrice *duale* de  $\mathbf{G}$ . Encore une fois, si l'on dispose d'au moins quatre droites mises en correspondance dans les deux images, il est possible d'estimer la matrice d'homographie duale (et donc la matrice d'homographie) par simple résolution d'un système linéaire.

Par contre, si on dispose de droites non coplanaires, il n'est pas possible d'estimer la géométrie épipolaire à partir de deux images (de même que la matrice d'homographie) [Faugeras 93].

## 2.4 De la reconstruction projective à la reconstruction euclidienne

À partir de la géométrie épipolaire, il est possible de remonter à la géométrie euclidienne. Il faut distinguer deux cas de figure :

- soit les paramètres intrinsèques de la caméra sont exactement connus (ce qui implique une calibration parfaite). Dans ce cas, on peut estimer exactement la rotation et la translation (à un facteur d'échelle près) de la caméra et la structure de la scène (à un facteur d'échelle près) [Tsai 84] [Faugeras 88]. Une information particulièrement intéressante qu'on peut obtenir est le rapport entre deux profondeurs, qui est donc indépendant du facteur d'échelle ;
- soit les paramètres intrinsèques sont entachés d'erreur (ce qui correspond à une calibration grossière de la caméra, plus réaliste d'un point de vue pratique). Dans ce cas, l'estimation du déplacement de la caméra et de la structure de la scène sera entachée d'erreur, mais, comme nous allons le voir, grâce à la connaissance de la matrice d'homographie du plan de l'infini, certains paramètres, comme l'angle de rotation ou le rapport entre les profondeurs, peuvent encore être estimés exactement.

Il faut remarquer que, dans le cas où la calibration de la caméra n'est pas parfaite, grâce à la connaissance de la matrice d'homographie du plan de l'infini, il est possible d'obtenir l'expression analytique des paramètres estimés en fonction des paramètres réels et des erreurs de calibration. Ceci nous permettra de trouver les conditions analytiques sur la stabilité de notre schéma d'asservissement visuel et sur sa robustesse par rapport aux erreurs de calibration de la caméra.

### 2.4.1 Paramètres intrinsèques exacts

**Matrice essentielle :** La matrice  $\mathbf{F}$  est estimée à partir des coordonnées exprimées en pixels dans l'image. Quand la caméra est calibrée, on peut utiliser les coordonnées métriques, et les deux images sont liées par la matrice essentielle  $\mathbf{E}$  [LonguetHiggins 81]. Cette dernière peut être estimée de la manière suivante :

$$\hat{\mathbf{E}} = \mathbf{A}^T \mathbf{F} \mathbf{A} \quad (2.35)$$

où on rappelle que  $\mathbf{A}$  est donné par (1.8). Le problème de l'estimation de la matrice essentielle est non-linéaire car la matrice essentielle réelle  $\mathbf{E}$  doit vérifier les contraintes établies par Huang et Faugeras [Huang 89] :  $\sigma_1 = \sigma_2$  et  $\sigma_3 = 0$  (où  $\sigma_1$ ,  $\sigma_2$  et  $\sigma_3$  sont les valeurs singulières de  $\mathbf{E}$ ). En effet,  $\mathbf{E}$  peut s'écrire comme le produit d'une matrice antisymétrique et d'une matrice de rotation :

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (2.36)$$

Ces contraintes peuvent être imposées a posteriori en utilisant l'algorithme de Tsai et Huang [Tsai 84]. En effet, la matrice essentielle estimée peut être décomposée en utilisant la SVD :

$$\hat{\mathbf{E}} = \mathbf{U} \hat{\mathbf{S}} \mathbf{V}^T \quad (2.37)$$

où  $\mathbf{U}$  et  $\mathbf{V}$  sont deux matrices orthonormales et  $\widehat{\mathbf{S}} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$  est une matrice diagonale. Après avoir imposé les contraintes de Huang-Faugeras ( $\widehat{\mathbf{S}}' = \text{diag}(\sigma, \sigma, 0)$  où par exemple  $\sigma = (\sigma_1 + \sigma_2)/2$ ), la matrice  $\widehat{\mathbf{S}}'$  peut se décomposer comme :

$$\widehat{\mathbf{S}}' = [\mathbf{t}']_{\times} \mathbf{R}' = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.38)$$

où  $[\mathbf{t}']_{\times}$  est une matrice antisymétrique et  $\mathbf{R}'$  une matrice de rotation. Il faut souligner qu'il existe deux couples de solutions possibles. Enfin, la matrice essentielle est décomposée comme :

$$\mathbf{E} = (\mathbf{U}[\mathbf{t}']_{\times} \mathbf{U}^T)(\mathbf{U}\mathbf{R}'\mathbf{V}^T) \quad (2.39)$$

où  $[\mathbf{t}]_{\times} = \mathbf{U}[\mathbf{t}']_{\times} \mathbf{U}^T$  est la matrice de préproduit vectoriel associée au vecteur de translation  $\mathbf{t}$  et  $\mathbf{R} = \mathbf{U}\mathbf{R}'\mathbf{V}^T$  est la matrice de rotation. Enfin, on peut choisir parmi les deux couples de solutions celle qui satisfait le fait que l'objet observé se trouve devant la caméra. Cette méthode de décomposition a été prouvée comme étant optimale par Hartley [Hartley 92]. Cependant, si les contraintes de Huang et Faugeras ne sont pas respectées par la matrice  $\widehat{\mathbf{S}}$ , la solution obtenue sera évidemment entachée d'erreurs.

En conclusion, à partir de la matrice essentielle estimée, on peut directement obtenir la matrice de rotation  $\mathbf{R}$  et la direction de translation  $\mathbf{t}/\|\mathbf{t}\|$ . De plus, les paramètres extrinsèques suivants peuvent ensuite être estimés : la translation  $\mathbf{t}/d^*$  à un facteur d'échelle près et le vecteur  $\mathbf{n}^*$  normal à  $\pi$ . Ces paramètres sont importants car ils seront utilisés dans la conception de notre schéma d'asservissement.

**Matrice d'homographie euclidienne :** La matrice correspondant à la matrice d'homographie  $\mathbf{G}$  dans le domaine calibré est la matrice d'homographie  $\mathbf{H}$ . Celle-ci peut être estimée de la manière suivante :

$$\widehat{\mathbf{H}} = \mathbf{A}^{-1} \mathbf{G} \mathbf{A} \quad (2.40)$$

De plus, la matrice d'homographie réelle peut s'écrire comme la somme d'une matrice de rotation et d'une matrice de rang 1 [Faugeras 88] :

$$\widehat{\mathbf{H}} = k \mathbf{H} = k \left( \mathbf{R} + \frac{\mathbf{t}}{d^*} \mathbf{n}^{*T} \right) \quad (2.41)$$

À partir de la matrice d'homographie estimée, on peut donc calculer  $k$ ,  $\mathbf{R}$ ,  $\mathbf{t}/d^*$ , et  $\mathbf{n}^*$ . Afin de reconstruire ces paramètres de mouvement, on peut utiliser un des algorithmes proposés en [Faugeras 88] ou [Zhang 95].

Malheureusement, dans le cas général, il existe deux solutions différentes. Si la cible est planaire, l'indétermination peut être éliminée si l'on dispose d'informations additionnelles (par exemple en connaissant le vecteur normal au plan de référence). Si la cible n'est pas



plane, l'indétermination est éliminée en considérant un autre plan de référence et en choisissant la solution commune aux deux couples [Faugeras 88]. En asservissement visuel, le choix n'est à effectuer qu'une seule fois, à l'initialisation du processus, car le vecteur normal  $\mathbf{n}^*$  ne varie pas au cours de l'asservissement.

Il faut aussi remarquer que d'autres paramètres sur la structure de l'objet observé peuvent être reconstruits à partir de la connaissance de la matrice d'homographie. Par exemple, on utilisera les paramètres suivants afin de concevoir notre schéma d'asservissement visuel :

- le rapport  $r$  entre les distances  $d$  et  $d^*$  des centres de projection  $\mathcal{O}$  et  $\mathcal{O}^*$  à  $\pi$  :

$$r = \frac{d}{d^*} = \det(\mathbf{H}) = 1 + \mathbf{n}^{*T} \mathbf{R} \mathbf{t}_{d^*} \quad (2.42)$$

avec  $\mathbf{t}_{d^*} = \mathbf{t}/d^*$ .

- le rapport entre la profondeur courante et désirée d'un point 3D :

$$\begin{cases} \frac{Z}{Z^*} = \frac{\mathbf{n}^{*T} \mathbf{m}^*}{\mathbf{n}^T \mathbf{m}} r & \text{si } \mathbf{m} \in \pi \\ \frac{Z}{Z^*} = \frac{\|[\mathbf{t}]_{\times} \mathbf{R} \mathbf{m}^*\|}{\|[\mathbf{t}]_{\times} \mathbf{m}\|} & \text{si } \mathbf{m} \notin \pi \end{cases} \quad (2.43)$$

- le rapport  $\rho$  entre la coordonnée  $Z$  d'un point 3D et  $d^*$  :

$$\begin{cases} \rho = \frac{Z}{d^*} = \frac{r}{\mathbf{n}^T \mathbf{m}} & \text{si } \mathbf{m} \in \pi \\ \rho = \frac{Z}{d^*} = \frac{Z}{Z^*} \frac{\|\mathbf{t}/d^*\|}{\|\mathbf{t}/Z^*\|} & \text{si } \mathbf{m} \notin \pi \end{cases} \quad (2.44)$$

avec  $\mathbf{t}/Z^* = \frac{Z}{Z^*} \mathbf{m} - \mathbf{R} \mathbf{m}^*$ .

Naturellement, les mêmes paramètres peuvent aussi être reconstruits à partir de l'estimation de la matrice essentielle. Toutefois, les résultats de simulation et expérimentaux qui seront présentés par la suite montrent que l'estimation est plus robuste si elle est obtenue à partir de la matrice d'homographie. On montre à présent comment il est possible, dans le domaine mal calibré, d'obtenir une forme analytique des paramètres de mouvement en fonction des paramètres réels et des erreurs de calibration de la caméra.

## 2.4.2 Paramètres intrinsèques erronés

En présence d'erreurs de calibration, le point mesuré dans l'image  $\hat{\mathbf{m}}$  peut s'écrire en fonction du point image réel  $\mathbf{m} = \mathbf{A}^{-1} \mathbf{p}$  sous la forme :

$$\hat{\mathbf{m}} = \hat{\mathbf{A}}^{-1} \mathbf{p} = \delta \mathbf{A} \mathbf{m} \quad (2.45)$$

où  $\hat{\mathbf{A}}$  est la matrice utilisée à la place de  $\mathbf{A}$  et  $\delta \mathbf{A} = \hat{\mathbf{A}}^{-1} \mathbf{A}$ . La matrice essentielle estimée est donnée par :

$$\hat{\mathbf{E}} = \hat{\mathbf{A}}^T \mathbf{F} \hat{\mathbf{A}} = k \delta \mathbf{A}^{-T} \mathbf{E} \delta \mathbf{A}^{-1} \quad (2.46)$$

Cette matrice peut s'écrire comme la multiplication d'une matrice de préproduit vectoriel et d'une matrice similaire à une matrice de rotation :

$$\widehat{\mathbf{E}} = k\delta\mathbf{A}^{-T} [\mathbf{t}]_{\times} \mathbf{R}\delta\mathbf{A}^{-1} = k\delta\mathbf{A}^{-T} [\mathbf{t}]_{\times} \delta\mathbf{A}^{-1}\delta\mathbf{A}\mathbf{R}\delta\mathbf{A}^{-1} = \det(\delta\mathbf{A}) [\delta\mathbf{A}\mathbf{t}]_{\times} \widehat{\mathbf{H}}_{\infty} \quad (2.47)$$

où  $\widehat{\mathbf{H}}_{\infty} = k\delta\mathbf{A}\mathbf{R}\delta\mathbf{A}^{-1}$ . Par ailleurs, la matrice d'homographie estimée est similaire à la matrice réelle :

$$\widehat{\mathbf{H}} = \widehat{\mathbf{A}}^{-1}\mathbf{G}\widehat{\mathbf{A}} = k\delta\mathbf{A}\mathbf{H}\delta\mathbf{A}^{-1} \quad (2.48)$$

où  $k$  est un facteur d'échelle inconnu. La matrice d'homographie estimée peut donc s'écrire comme la somme d'une matrice similaire à une matrice de rotation et d'une matrice de rang 1 :

$$\widehat{\mathbf{H}} = k\delta\mathbf{A}\mathbf{R}\delta\mathbf{A}^{-1} + k\delta\mathbf{A}\frac{\mathbf{t}}{d^*}\mathbf{n}^{*T}\delta\mathbf{A}^{-1} = \widehat{\mathbf{H}}_{\infty} + \widehat{\mathbf{t}}_{d^*}\widehat{\mathbf{n}}^{*T} \quad (2.49)$$

où  $\widehat{\mathbf{t}}_{d^*} = k\|\mathbf{n}^{*T}\delta\mathbf{A}^{-1}\|\delta\mathbf{A}\mathbf{t}_{d^*}$  et  $\widehat{\mathbf{n}}^{*T} = \frac{\mathbf{n}^{*T}\delta\mathbf{A}^{-1}}{\|\mathbf{n}^{*T}\delta\mathbf{A}^{-1}\|}$ . Les valeurs propres de  $\mathbf{R}$  dépendent de l'angle de rotation  $\theta$ , et le vecteur propre correspondant à la valeur propre unitaire est l'axe de rotation  $\mathbf{u}$ . La matrice  $\widehat{\mathbf{H}}_{\infty}$  n'est pas une matrice de rotation, mais elle est similaire à  $\mathbf{R}$ , ce qui implique que les deux matrices ont les mêmes valeurs propres et que les vecteurs propres de  $\widehat{\mathbf{H}}_{\infty}$  sont les vecteurs propres de  $\mathbf{R}$  multipliés par la matrice  $\delta\mathbf{A}$ .

On supposera par la suite, pour simplifier les calculs, que l'on peut calculer  $\widehat{\mathbf{H}}_{\infty}$ . Pour cela, il faut disposer d'au moins 4 images du même objet ou de connaissances supplémentaires sur la scène telles que 3 points de l'infini ou, par exemple, savoir que le plan de référence est parallèle au plan image [Robert 95] [Vieville 96].

Dans ce cas, l'angle de rotation estimé  $\widehat{\theta}$  et l'axe de rotation estimé  $\widehat{\mathbf{u}}$  peuvent être calculés directement à partir de  $\widehat{\mathbf{H}}_{\infty}$ , et il peuvent donc être exprimés en fonction des valeurs réelles et des erreurs de calibration :

$$\widehat{\theta} = \theta \quad \text{et} \quad \widehat{\mathbf{u}} = \frac{\delta\mathbf{A}\mathbf{u}}{\|\delta\mathbf{A}\mathbf{u}\|} \quad (2.50)$$

Il faut souligner que, de même que l'angle de rotation  $\theta$ , le rapport  $r$  peut être calculé sans erreur :

$$\widehat{r} = \det(\widehat{\mathbf{H}}/k) = \det(\mathbf{H}) = r \quad (2.51)$$

où  $k$  est donné par [Vieville 96] :

$$k = \sqrt[3]{\det(\widehat{\mathbf{H}}_{\infty})} \quad (2.52)$$

De plus, comme  $\widehat{\mathbf{n}} = \frac{1}{k}\widehat{\mathbf{H}}_{\infty}^{-T}\widehat{\mathbf{n}}^* = (\delta\mathbf{A}^{-T}\mathbf{R}\delta\mathbf{A}^T)(\frac{\delta\mathbf{A}^{-T}\mathbf{n}^*}{\|\delta\mathbf{A}^{-T}\mathbf{n}^*\|}) = \frac{\delta\mathbf{A}^{-T}\mathbf{n}}{\|\delta\mathbf{A}^{-T}\mathbf{n}\|}$ ,  $\widehat{\rho}$  est donné par :

$$\widehat{\rho} = \frac{\widehat{r}}{\widehat{\mathbf{n}}^T\widehat{\mathbf{m}}} = \frac{r}{\mathbf{n}^T\mathbf{m}} = \|\delta\mathbf{A}^{-T}\mathbf{n}^*\|\rho \quad (2.53)$$

Comme on l'a déjà dit, les équations précédentes sont valides si  $\widehat{\mathbf{H}}_{\infty}$  est connue. En pratique, nous avons utilisé la méthode proposée par [Faugeras 88] pour estimer  $\mathbf{R}$ ,  $\mathbf{n}$  et  $\mathbf{t}_{d^*}$ . Comme les erreurs introduites par l'utilisation de cette méthode, conçue pour opérer dans le cas calibré, sont sûrement plus importantes que si on avait utilisé la décomposition à partir de  $\widehat{\mathbf{H}}_{\infty}$ , on obtiendra une borne supérieure des performances d'une telle estimation.

## 2.5 Résultats expérimentaux

### 2.5.1 Scène synthétique

Nous avons effectué des simulations afin de comparer l'algorithme de reconstruction euclidienne proposé dans ce chapitre avec les autres méthodes classiques. Deux cibles différentes, chacune composée de 12 points, ont été utilisées pour la simulation. La première est planaire, tandis que la deuxième possède quatre points coplanaires et les autres distribués dans un volume de  $20\text{cm}^3$ . Naturellement, aucune connaissance a priori de la structure de l'objet n'est utilisée pour estimer les déplacements de la caméra. Pour chaque déplacement, on a répété 100 fois l'estimation en ajoutant un bruit sur les coordonnées des points dans l'image (avec un écart type de 1 pixel). On calcule les erreurs suivantes (les chapeaux indiquent la valeur estimée):

- Erreur en rotation: l'angle  $\theta_r$  extrait de la matrice de rotation  $\mathbf{R}\hat{\mathbf{R}}^{-1}$ . Cet angle représente la distance entre les deux rotations  $\mathbf{R}$  et  $\hat{\mathbf{R}}$ , soit la longueur minimale de la géodésique partant de  $\mathbf{R}$  et terminant en  $\hat{\mathbf{R}}$ .
- Erreur en translation: l'angle  $\theta_t$  entre les deux vecteurs normalisés  $\mathbf{t}/\|\mathbf{t}\|$  et  $\hat{\mathbf{t}}/\|\hat{\mathbf{t}}\|$ . En effet, puisque la reconstruction est obtenue à un facteur d'échelle près, seule la direction de translation peut être comparée.

De plus, on calcule l'erreur médiane, l'écart médian ainsi que les erreurs maximales (en valeur absolue) sur 100 tirages pour chaque expérience. Nous allons comparer les algorithmes suivants:

- L'algorithme linéaire de [Boufama 95] pour l'estimation de l'homographie présentée au paragraphe 2.3.2. La méthode sera indiquée avec le sigle **VPL** et les courbes des résultats correspondants seront marquées par des carrés.
- L'algorithme "linéarisé" d'estimation de la matrice d'homographie que nous avons proposé au paragraphe 2.3.3. La méthode sera indiquée avec le sigle **HML** et les courbes des résultats correspondants seront marquées par des cercles.
- L'algorithme linéaire des "huit points", utilisant des données normalisées, pour l'estimation de la matrice fondamentale présentée au paragraphe 2.3.1 [Hartley 97]. La méthode sera indiquée avec le sigle **EML** et les courbes des résultats correspondants seront marquées par des triangles.
- L'algorithme non-linéaire d'estimation de la matrice essentielle décrit dans [Deriche 94] et initialisé avec les résultats obtenus par l'algorithme **EML**. La méthode sera indiquée avec le sigle **EMN** et les courbes des résultats correspondants seront marquées par des losanges.

Les algorithmes **EML** et **EMN** ont été testés en utilisant le programme Fmatrix développé par Z. Zhang<sup>1</sup>.

---

1. le programme Fmatrix est disponible sur <http://www.inria.fr/robotvis/personnel/zzhang/zzhang-eng.html>

## 2.5.2 Cible planeaire

Les résultats décrits sur la Figure 2.3 ont été obtenus en utilisant la cible plane. Le déplacement de la caméra a été fixé à  $\mathbf{t} = [-38 \ 38 \ 45]^T$  cm et  $\mathbf{u}\theta = [64 \ -16 \ 138]^T$  degrés. Comme prévu, considérer une cible plane n'est pas favorable pour les algorithmes **EML** et **EMN** basés sur l'estimation de la matrice essentielle car, dans ce cas, l'épipoles n'est pas défini. En effet, avec ces algorithmes, on obtient des erreurs médianes importantes (45 degrés pour la rotation et 55 degrés pour la direction de translation) quelque soit le nombre de points considérés. Les résultats fournis par l'algorithme **HML** sont satisfaisants car plus précis et plus stables (l'erreur médiane est de 5 degrés).

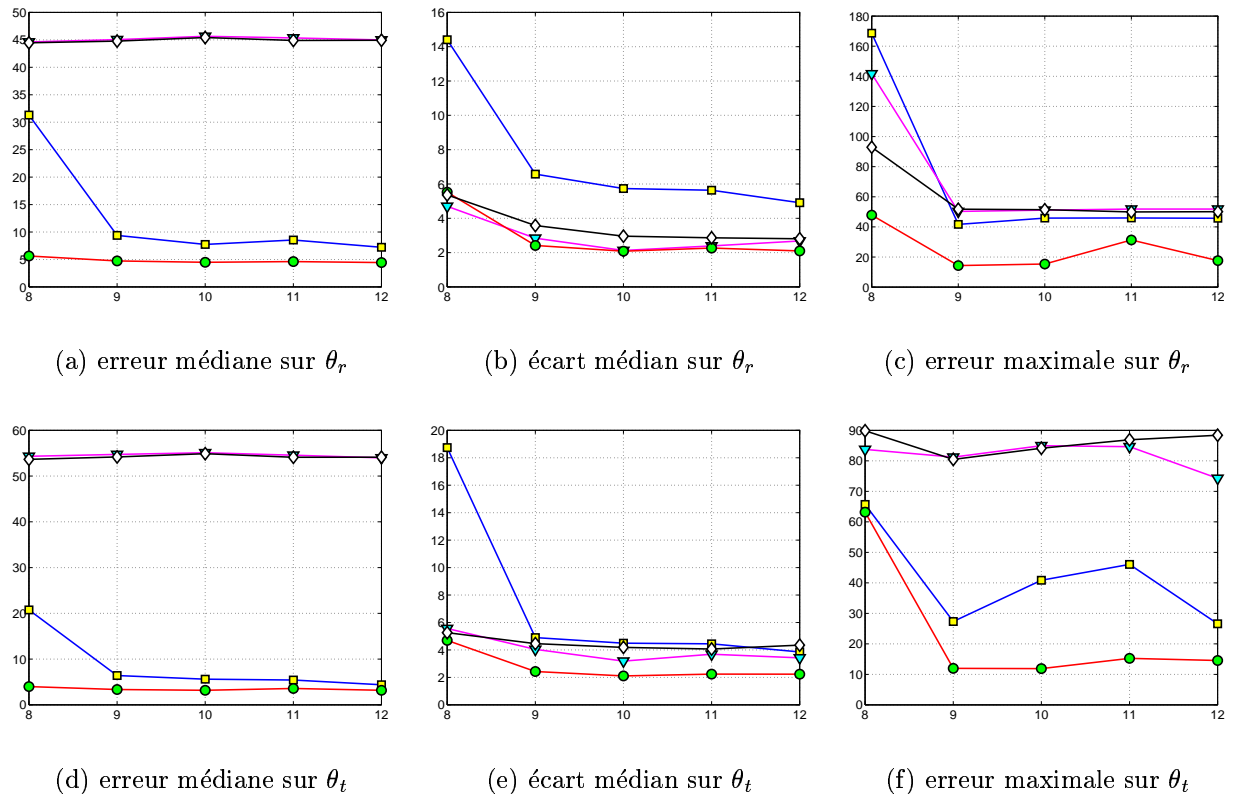


FIG. 2.3 – Comparaison des méthodes de reconstruction pour un objet plan en fonction du nombre de points considérés

## 2.5.3 Cible non planeaire

On utilise maintenant la cible non plane.

### 2.5.3.1 Déplacement nul

On teste tout d'abord la robustesse des algorithmes dans le cas où la caméra n'a pas été déplacée, ce qui correspond, dans le cas d'une caméra commandée par asservissement visuel

à la phase finale de convergence. On peut voir sur la Figure 2.4 que les résultats fournis dans ce cas sont plus stables que dans le cas où la cible était plane. En effet, l'erreur maximale est plus faible, même en n'utilisant que 8 points (on rappelle que les 4 premiers points sont coplanaires). Dans ce cas, les erreurs maximales des algorithmes **VPL** et **HML** sont inférieures à 10 degrés. Les algorithmes **EML** et **EMN** fournissent des résultats comparables, mais c'est encore une fois l'algorithme **HML** qui donne les meilleurs résultats.

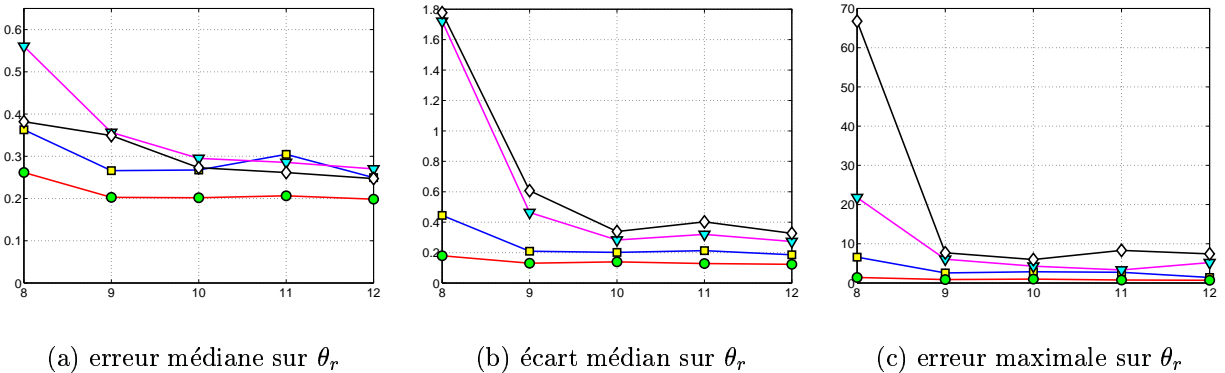


FIG. 2.4 – Comparaison des méthodes de reconstruction pour un déplacement nul en fonction du nombre de points considérés

### 2.5.3.2 Rotation pure

On teste à présent la robustesse des algorithmes à une rotation pure. Les résultats pour un déplacement en rotation  $\mathbf{u}\theta = [6 \ 6 \ 6]^T$  degrés sont présentés sur la Figure 2.5. Dans ce cas encore une fois dégénéré pour l'estimation de la géométrie épipolaire (puisque l'épipole est alors indéfini dans l'image), l'algorithme **HML** fournit des résultats plus précis et robustes que l'algorithme **VPL**, car l'homographie n'est pas estimée conjointement à l'épipole. Comme attendu, les algorithmes **EML** et **EMN** sont moins robustes que les algorithmes **HML** et **VPL**. Ces résultats confirment que, dans les cas particuliers, l'utilisation d'une matrice d'homographie est préférable afin d'obtenir le déplacement de la caméra. Comme les quatre premiers points considérés sont coplanaires, tous les algorithmes donnent les pires résultats avec seulement 8 points (théoriquement cinq points, n'appartenant pas à  $\pi$ , sont nécessaires). Ceci explique la grande différence, principalement pour les méthodes **EML** et **EMN**, des résultats en passant de 8 à 9 points.

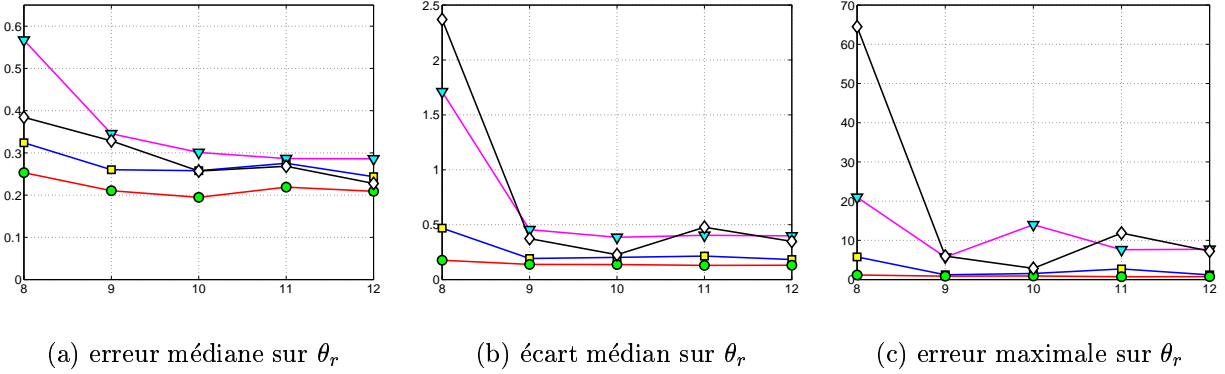


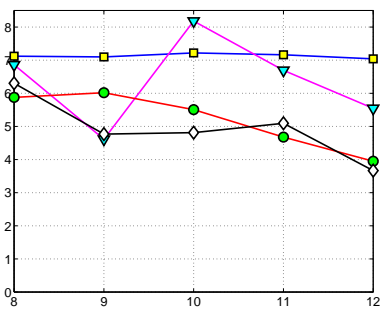
FIG. 2.5 – Comparaison des méthodes de reconstruction pour une rotation pure en fonction du nombre de points considérés

### 2.5.3.3 Translation pure

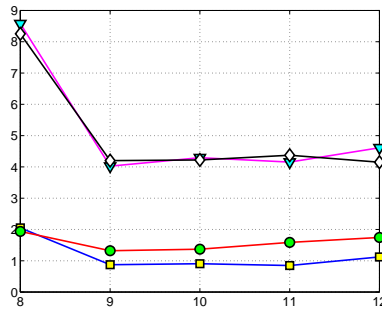
Dans cette simulation, le déplacement est une translation pure :  $\mathbf{t} = [6 \ 6 \ 6]^T$  cm. Il est bien connu que l'estimation de la matrice essentielle n'est pas robuste pour de tels mouvements [Luong 92]. En effet, comme on peut le voir en Figure 2.6, les méthodes **EML** et **EMN** peuvent être instables et produire une erreur sur la direction de translation de presque 90 degrés, ce qui implique une grande erreur médiane. Ce sont bien sûr des cas exceptionnels pouvant aussi s'expliquer par le fait qu'on applique un bruit de 1 pixel sur chaque point sans tenir compte de la réalité physique. L'algorithme **HML** peut, lui aussi, atteindre une erreur maximale de 75 degrés en translation (ce qui n'est pas du tout satisfaisant), mais l'erreur médiane et l'écart médian sont autour de 20 degrés. En ce qui concerne l'erreur en rotation, même si l'erreur médiane est presque la même pour toutes les méthodes, l'écart médian et l'erreur maximale sont divisés par deux avec les algorithmes **HML** et **VPL**. Toutefois, le biais de l'algorithme **VPL** est plus important que celui de l'algorithme **HML**, aussi bien pour l'erreur en translation qu'en rotation.

### 2.5.3.4 Déplacement quelconque

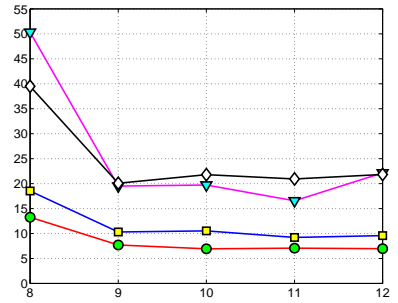
Enfin, les résultats obtenus avec un déplacement générique ( $\mathbf{t} = [-38 \ 38 \ 45]^T$  cm et  $\mathbf{u}\theta = [64 \ -16 \ 138]^T$  degrés) sont illustrés en Figure 2.7. Dans ce cas, l'algorithme **EMN** fournit, comme attendu, les meilleurs résultats, mais on peut remarquer que les résultats obtenus en utilisant l'algorithme **HML** en sont très proches, et donc satisfaisants par rapport à ceux obtenus en utilisant la méthode **EML**. Enfin, l'algorithme **VPL** donne ici des résultats assez médiocres, car l'estimation conjointe de l'épipoles et de la matrice d'homographie implique des erreurs importantes sur le déplacement de la caméra.



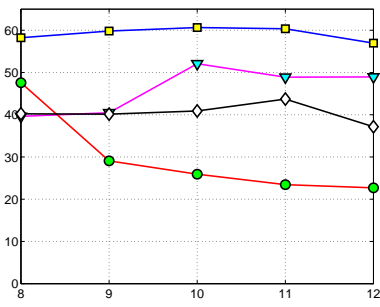
(a) erreur médiane sur  $\theta_r$



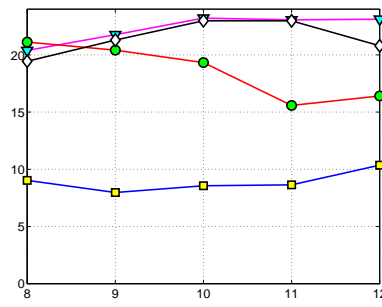
(b) écart médian sur  $\theta_r$



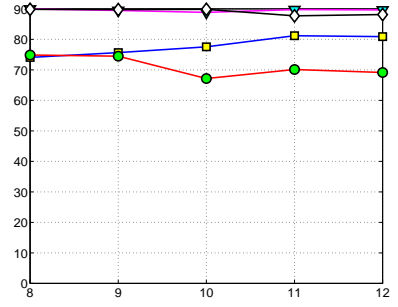
(c) erreur maximale sur  $\theta_r$



(d) erreur médiane sur  $\theta_t$



(e) écart médian sur  $\theta_t$



(f) erreur maximale sur  $\theta_t$

FIG. 2.6 – Comparaison des méthodes de reconstruction pour une translation pure en fonction du nombre de points considérés

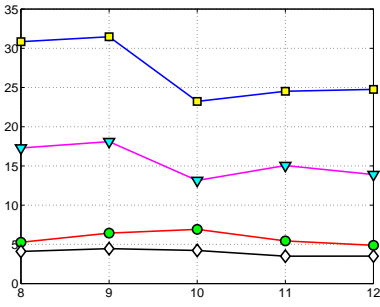
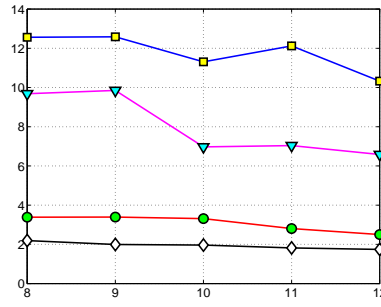
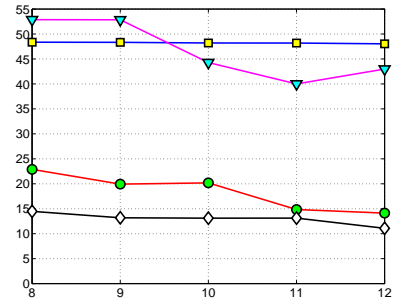
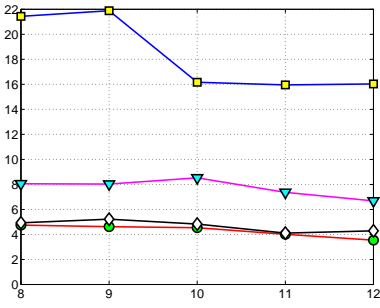
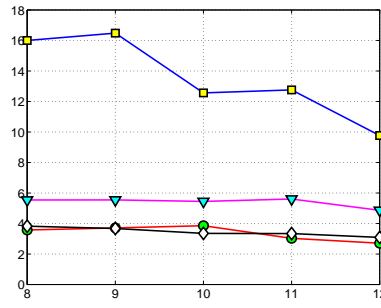
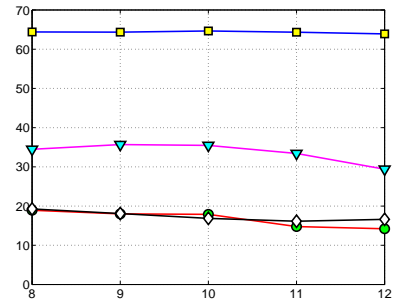
(a) erreur médiane sur  $\theta_r$ (b) écart médian sur  $\theta_r$ (c) erreur maximale sur  $\theta_r$ (d) erreur médiane sur  $\theta_t$ (e) écart médian sur  $\theta_t$ (f) erreur maximale sur  $\theta_t$ 

FIG. 2.7 – Comparaison des méthodes de reconstruction pour un déplacement générique en fonction du nombre de points considérés



## 2.5.4 Scène réelle

On considère maintenant une scène réelle<sup>2</sup> (voir Figure 2.8) et une caméra calibrée. Dans cette expérience, le déplacement de la caméra a été fixé à  $\mathbf{t} = [14 \ 6 \ -18]^T$  cm et  $\mathbf{u}\theta = [2.1 \ -3.1 \ -0.7]^T$  degrés. Les points, extraits et mis en correspondance en utilisant le programme Image Matching<sup>3</sup>, ont été numérotés de 1 à 28 (voir Figure 2.8a et Figure 2.8b). Les 3 premiers points ont été choisis arbitrairement comme points de référence pour effectuer le changement de coordonnées projectives, car ils sont bien dispersés dans les deux images. Les erreurs  $\theta_r$  et  $\theta_t$  sont respectivement tracées sur les Figure 2.8c et 2.8d en fonction du nombre de points utilisés dans l'estimation. Dans l'ensemble, l'algorithme **EMN** donne de meilleurs résultats que l'algorithme **EML** (sauf pour 13 points, ce qui est assez surprenant). En accord avec les résultats de simulation, l'algorithme **HML** fournit des résultats plus précis que les algorithmes **EML** et **VPL**. Toutefois, dans cette expérience, les résultats de l'algorithme **VPL** sont assez proches de ceux de l'algorithme **EML**. Il est assez surprenant que la méthode **HML** donne des résultats plus précis que la méthode **EMN**. Ceci est probablement dû au fait que le déplacement de la caméra n'est pas très grand par rapport à la dimension de la scène. Il faut souligner que le choix des trois points de référence est assez important pour la méthode **HML**, mais, comme on l'a déjà évoqué, il peut être effectué de manière automatique en sélectionnant les trois points qui maximisent l'aire des triangles associés dans les deux images.

## 2.5.5 Expériences avec un robot et une caméra embarquée

Les algorithmes d'estimation **HML** et **VPL** ont été intégrés dans la technique d'asservissement visuel  $2D\frac{1}{2}$  qui fait l'objet du chapitre suivant. L'objectif des expériences présentées ci-dessous est d'évaluer les performances des algorithmes de reconstruction du déplacement de la caméra. Le schéma d'asservissement ne sera donc pas détaillé ici.

En ce qui concerne la calibration de la caméra, on utilise la taille des pixels et la distance focale fournies par le constructeur de la caméra. Le centre de l'image a été utilisé comme point principal.

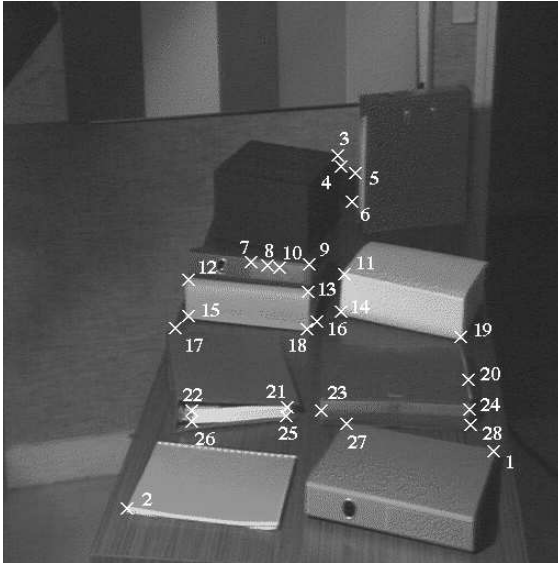
La cible est composée de douze disques blancs disposés sur trois plans parallèles (voir Figure 2.9). La disposition particulière des marqueurs est utilisée uniquement pour effectuer la mise en correspondance. Les informations visuelles considérées sont les coordonnées dans l'image du centre de gravité de chaque disque. Afin de prouver la validité des algorithmes d'estimation de la matrice d'homographie, même dans des conditions non optimales, les trois points de référence n'ont pas été choisis très dispersés dans l'image (voir Figure 2.9a où un carré blanc a été dessiné autour de chaque point de référence).

Dans ces expériences, nous n'avons pas testé la méthode **EMN** car, actuellement, elle ne semble pas être assez rapide pour pouvoir tourner à la cadence vidéo. La méthode **EML** n'a pas non plus été implémentée. En effet, à partir des résultats de simulation décrits

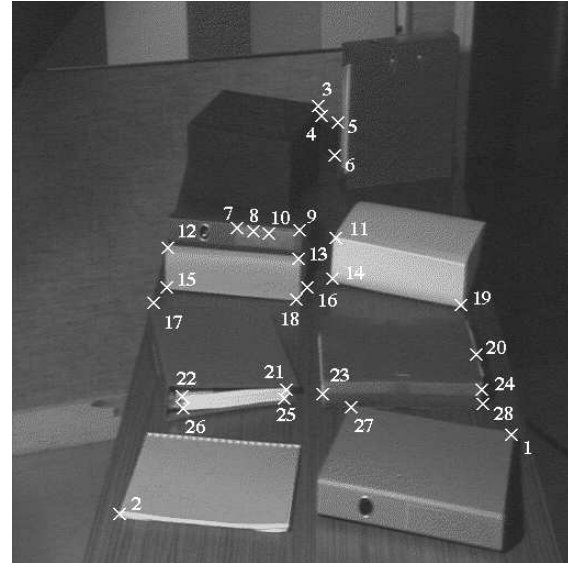
---

2. les images, fournies par le projet Syntim de l'INRIA, sont disponibles sur <http://www-syntim.inria.fr/syntim/analyse/paires-eng.html>

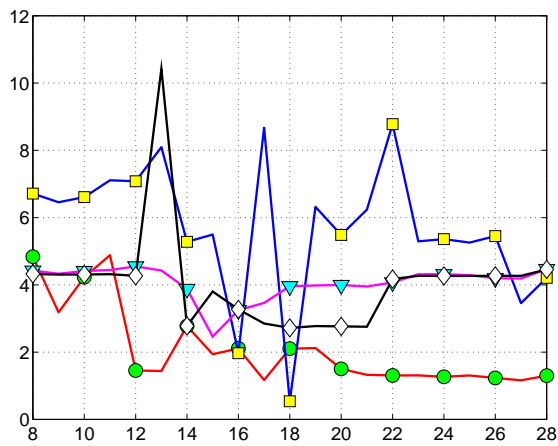
3. le programme Image Matching, développé par Z. Zhang, est disponible sur <http://www.inria.fr/robotvis/personnel/zzhang/zzhang-eng.html>



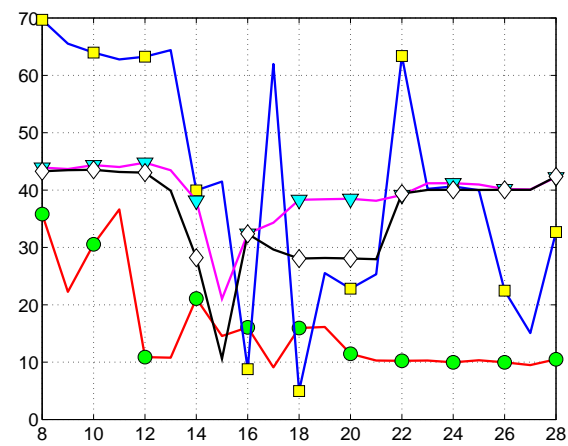
(a) image gauche



(b) image droite



(c) erreur en rotation



(d) erreur sur la direction de translation

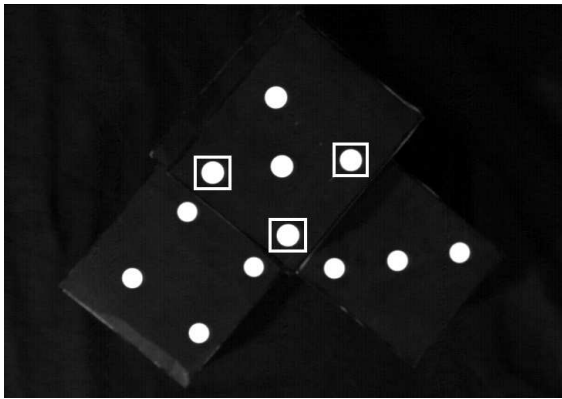
FIG. 2.8 – Comparaison des méthodes de reconstruction pour une scène réelle en fonction du nombre de points considérés

précédemment, on peut s'attendre à des résultats vraiment instables dans les cas où l'épipole est indéfini, ce qui malheureusement arrive quand la caméra rejoint sa position désirée. Pour cette raison, seules les méthodes **VPL** et **HML** ont été testées. Toutefois, exception faite pour les configurations particulières (qui inclut la phase finale de la convergence), on peut supposer que la méthode **EML** aurait fourni des résultats intermédiaires entre la méthode **VPL** et la méthode **HML**.

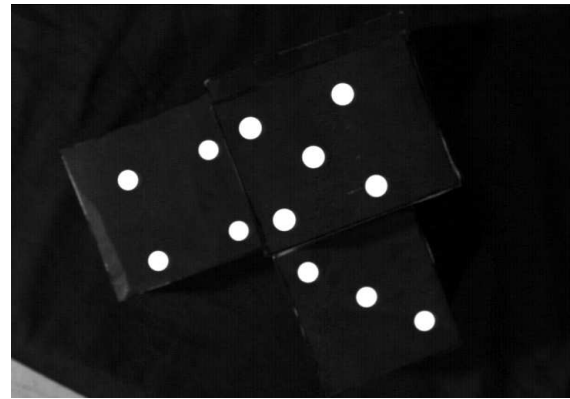
### 2.5.5.1 Rotation pure

Les résultats de l'asservissement visuel  $2D\frac{1}{2}$ , obtenus en effectuant un déplacement en rotation de  $-30$  degrés autour de l'axe  $\vec{z}$ , sont illustrés sur la Figure 2.10 (où nous avons donné en abscisse le nombre d'itérations). Afin de comparer les algorithmes **HML** et **VPL**, nous avons placé sur deux colonnes l'estimation du déplacement de la caméra (rotation et translation) ainsi que la loi de commande envoyée au robot (vitesse de translation et de rotation).

Les deux algorithmes **HML** et **VPL** fournissent de bons résultats même si l'épipole n'est pas défini. Toutefois, on peut remarquer que la rotation et la translation estimées en utilisant l'algorithme **HML** sont plus robustes que celles estimées en utilisant l'algorithme **VPL**. Ceci implique une loi de commande plus stable et un temps de convergence inférieure (160 iteration au lieu de 250).

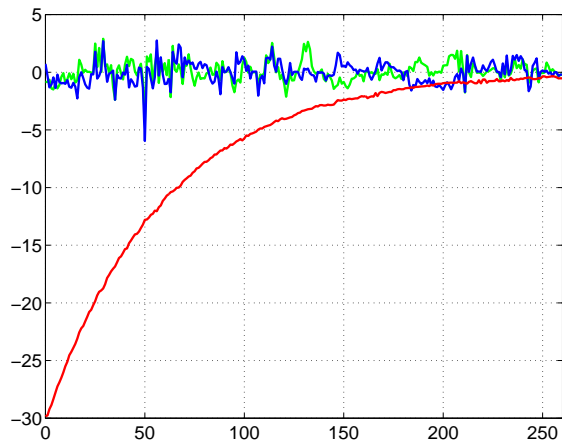
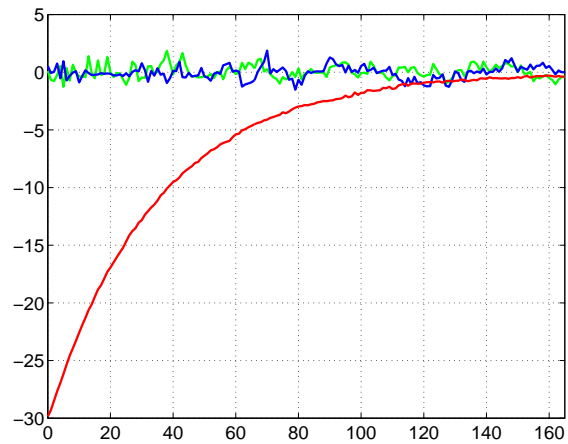
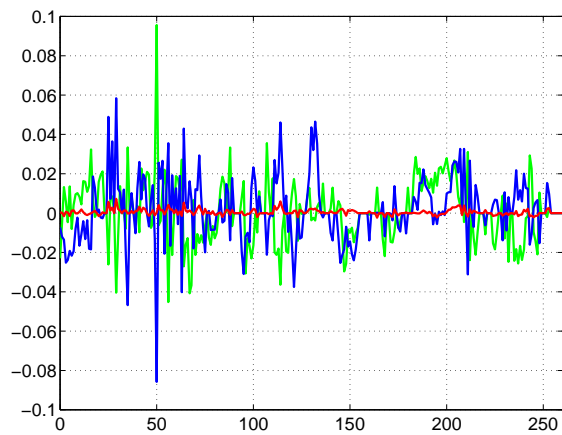


(a) Image finale

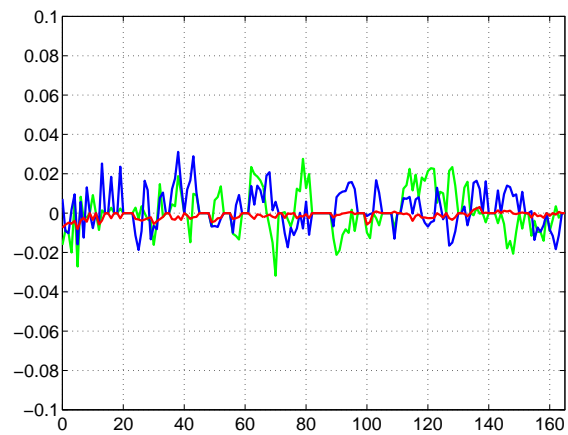


(b) Image initiale

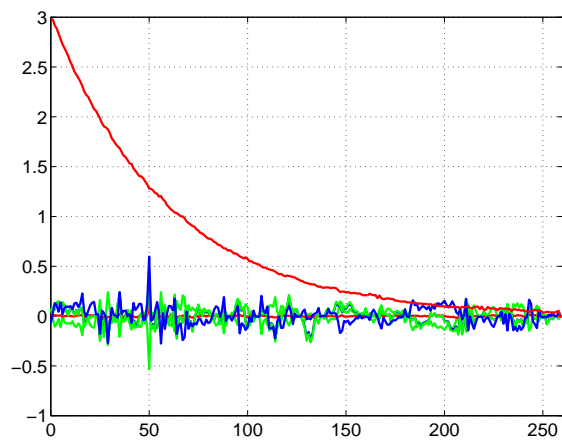
FIG. 2.9 – *Rotation pure de la caméra*

(a) rotation  $u\theta$  (deg) (VPL)(b) rotation  $u\theta$  (deg) (HML)

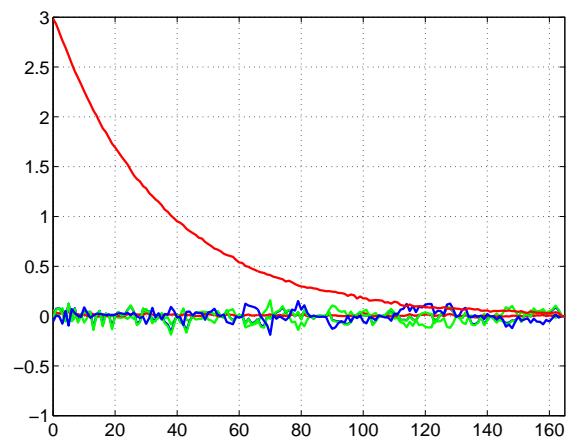
(c) translation (cm) (VPL)



(d) translation (cm) (HML)



(e) vitesse (cm/s et deg/s) (VPL)

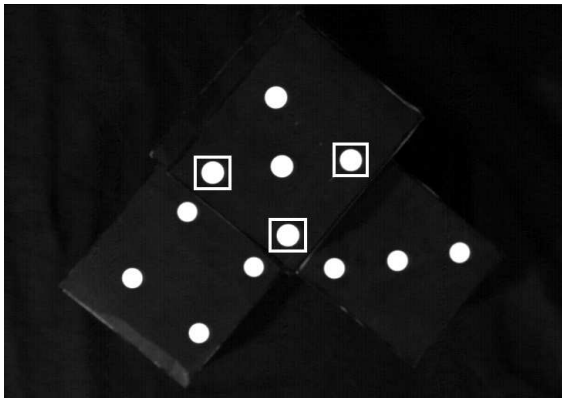


(f) vitesse (cm/s et deg/s) (HML)

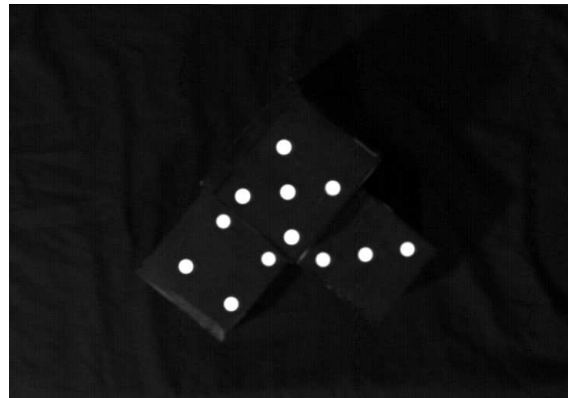
FIG. 2.10 – Résultats d'asservissement visuel pour une rotation pure de la caméra

### 2.5.5.2 Translation pure

Dans cette deuxième expérience, le déplacement de la caméra est une translation pure telle que l'épipole coïncide avec un point de référence dans l'image ( $\mathbf{e} = \mathbf{p}_1$ ). Les résultats sont illustrés en Figure 2.12 (organisée comme la Figure 2.10). On peut remarquer que, même si l'algorithme **HML** est plus robuste et stable, l'asservissement visuel  $2D\frac{1}{2}$  converge tout de même si on utilise l'algorithme **VPL**, grâce au fait que les erreurs sont compensées par la boucle fermée. Comme on peut le voir sur les courbes, de la première jusqu'à la cinquième itération, l'algorithme **VPL** est instable car proche d'une singularité. De plus, les erreurs dans l'estimation de la matrice d'homographie se répercutent sur la loi de commande calculée qui est donc plus stable en utilisant la méthode **HML**.

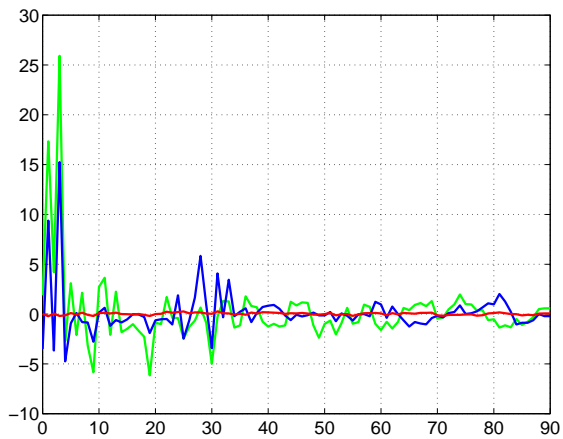
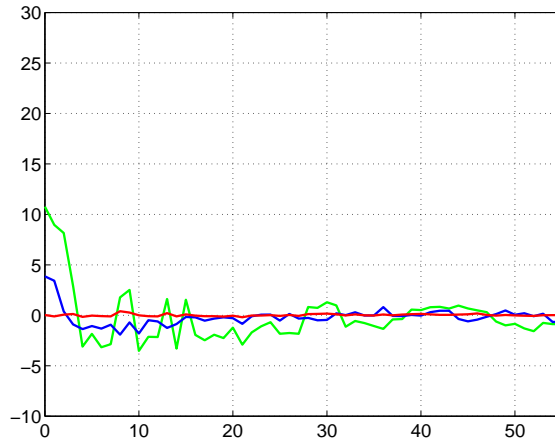
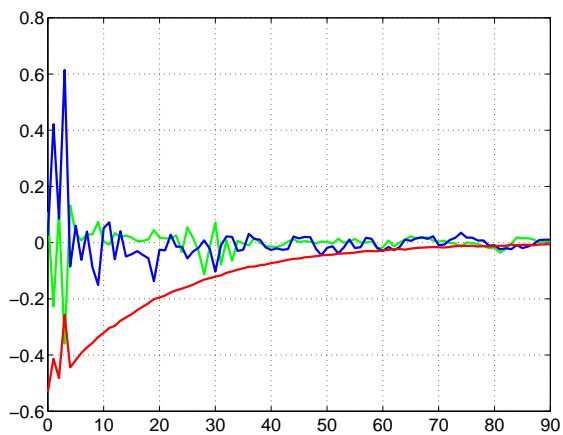


(a) Image finale

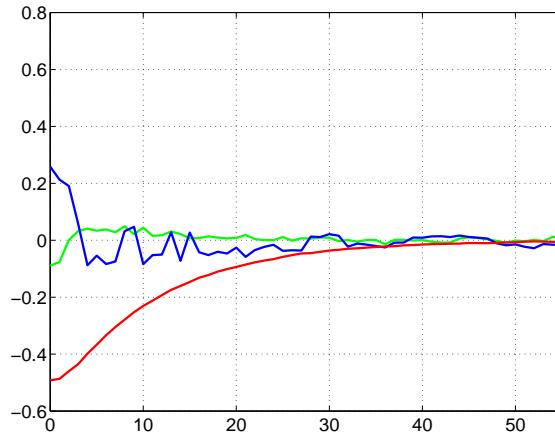


(b) Image initiale

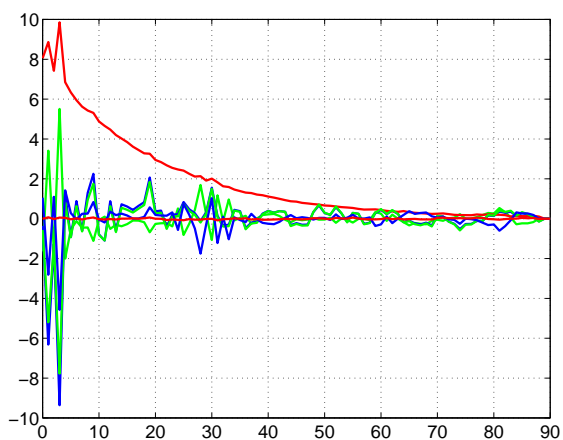
FIG. 2.11 – *Translation pure de la caméra*

(a) rotation  $u\theta$  (deg) (VPL)(b) rotation  $u\theta$  (deg) (HML)

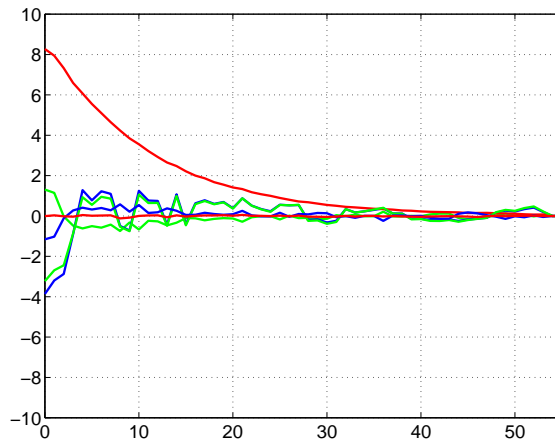
(c) translation (cm) (VPL)



(d) translation (cm) (HML)



(e) vitesse (cm/s et deg/s) (VPL)



(f) vitesse (cm/s et deg/s) (HML)

FIG. 2.12 – Résultats d'asservissement visuel pour une translation pure de la caméra

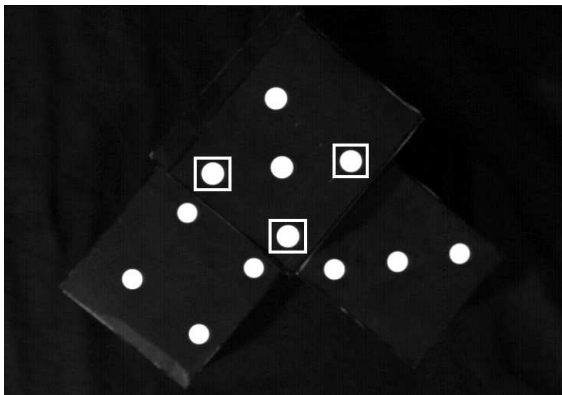
### 2.5.5.3 Déplacement quelconque

Dans cette dernière expérience, le déplacement de la caméra est quelconque (voir Figure 2.13) :

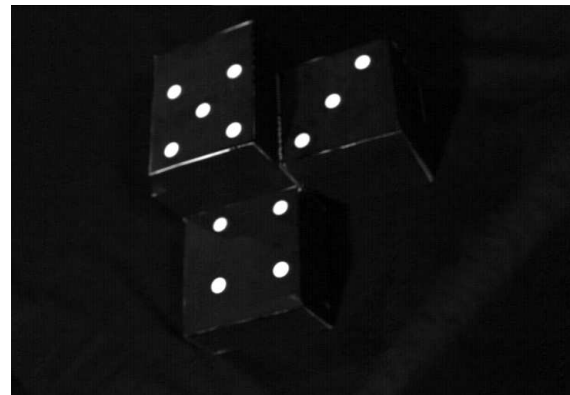
- $\mathbf{t} = [ -1.3 \quad 55.2 \quad 4.1 ]^T$  cm ;
- $\mathbf{u}\theta = [ 36.2 \quad -17.2 \quad 48.4 ]^T$  deg.

Encore une fois et en accord avec les résultats de simulation, l'algorithme **HML** fournit des résultats plus robustes que l'algorithme **VPL** (voir Figure 2.14). On peut notamment remarquer que l'asservissement converge 100 itérations plus tôt alors que la vitesse de convergence spécifié est la même dans les deux cas.

Par ailleurs, avec l'algorithme **HML**, l'estimation de la rotation entre la position initiale et la position désirée de la caméra est  $\hat{\mathbf{r}} = [ 34.8 \quad -14.9 \quad 48.3 ]^T$  degrés. La direction de translation estimée est  $\hat{\mathbf{t}}/\|\hat{\mathbf{t}}\| = [ -0.04 \quad 0.99 \quad 0.04 ]^T$  (tandis que la direction de translation réelle est  $\mathbf{t}/\|\mathbf{t}\| = [ -0.02 \quad 0.99 \quad 0.07 ]^T$ ). L'algorithme est donc assez précis malgré la calibration grossière de la caméra (l'erreur maximale en rotation est environ de 2 degrés, de même que l'erreur sur la direction de translation).

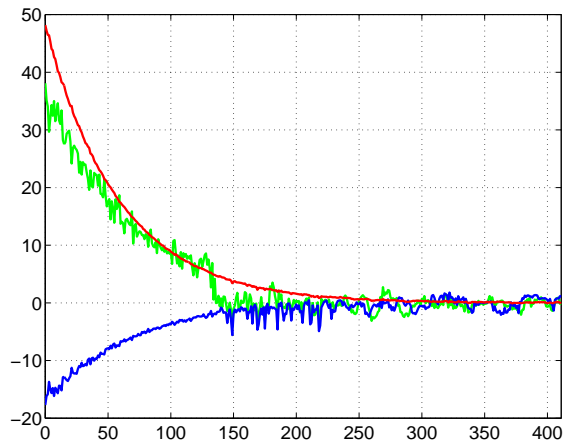
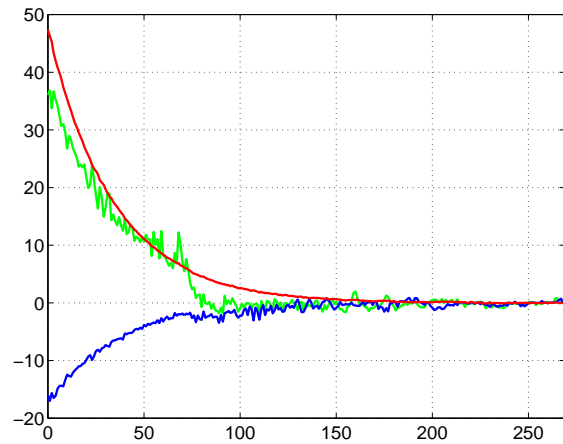
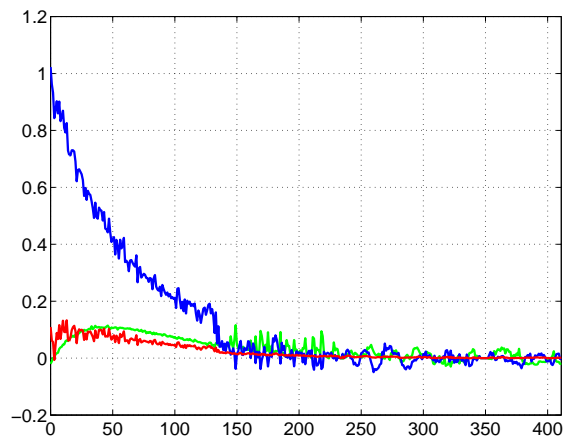


(a) Image finale

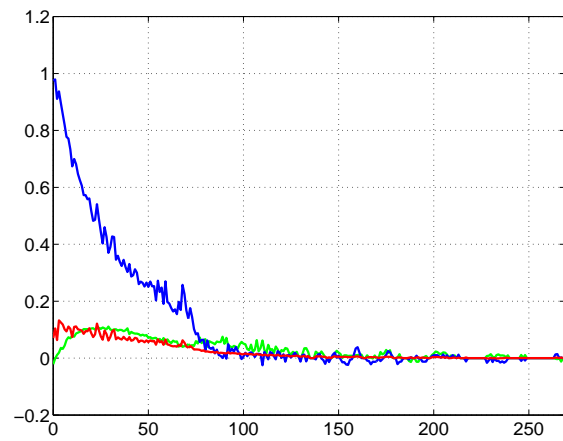


(b) Image initiale

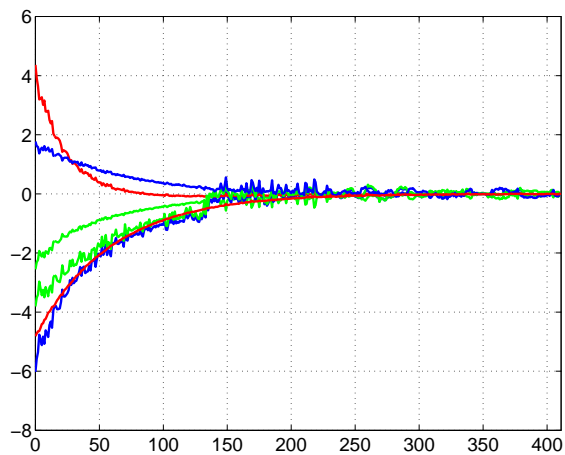
FIG. 2.13 – *Déplacement générique de la caméra*

(a) rotation  $u\theta$  (deg) (VPL)(b) rotation  $u\theta$  (deg) (HML)

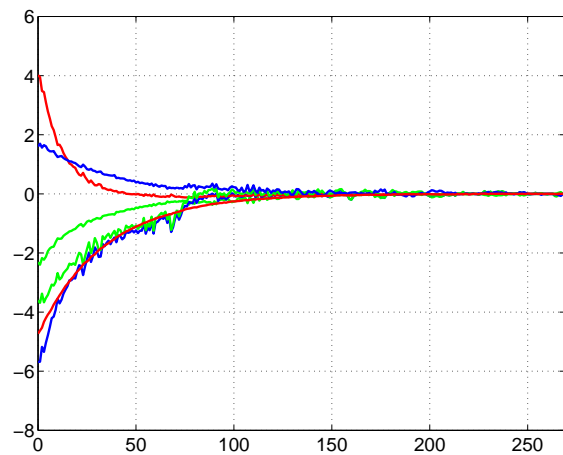
(c) translation (cm) (VPL)



(d) translation (cm) (HML)



(e) vitesse (cm/s et deg/s) (VPL)



(f) vitesse (cm/s et deg/s) (HML)

FIG. 2.14 – Résultats d'asservissement visuel pour un déplacement générique de la caméra



## 2.6 Conclusion

Dans ce chapitre, nous avons montré comment reconstruire le déplacement partiel d'une caméra à partir de deux images d'un objet inconnu. En effet, si les paramètres intrinsèques de la caméra sont parfaitement connus, on peut estimer la rotation et la translation à un facteur d'échelle près. De plus, des informations sur la structure de l'objet observé peuvent être estimées, comme par exemple la profondeur relative (le rapport entre la profondeur courante et désirée) des points de la cible. On verra dans le chapitre suivant que, à partir de ces informations, il est possible de concevoir des schémas d'asservissement visuel qui ne nécessitent pas de modèle 3D de l'objet observé.

Parmi les méthodes réalisant une reconstruction projective et euclidienne, nous avons mis en évidence que celles qui estiment une matrice d'homographie sont les mieux adaptées à être intégrées en asservissement visuel. Il faut souligner que les schémas d'asservissement présentés dans le chapitre suivant ne dépendent pas de la méthode utilisée pour la reconstruction (linéaire ou pas). Toutefois, une estimation linéaire est souhaitable afin d'obtenir des algorithmes qui tournent à la cadence vidéo. De plus, il est important de pouvoir gérer les cas particuliers. C'est pourquoi nous avons proposé un algorithme d'estimation de la matrice d'homographie qui n'utilise que des résolutions de systèmes linéaires. Les résultats expérimentaux et de simulation confirment la validité de notre approche.

Lorsque les paramètres intrinsèques de la caméra sont grossièrement connus, il est possible d'exprimer le déplacement estimé de la caméra en fonction du déplacement réel et des erreurs de calibration à partir de la matrice d'homographie à l'infini. Ceci nous permettra de démontrer de manière analytique la robustesse des lois de commande aux incertitudes sur les paramètres du système.

# Chapitre 3

## Asservissement visuel mono-caméra

Dans ce chapitre, on décrit deux nouvelles méthodes d'asservissement visuel basées sur la reconstruction partielle du déplacement de la caméra qui a été décrite dans le chapitre précédent. Comme nous le verrons, la boucle d'asservissement en rotation peut être découplée de celle en translation, permettant ainsi d'obtenir des résultats analytiques sur la stabilité de l'asservissement visuel et de démontrer la robustesse de nos méthodes par rapport aux erreurs de calibration du système.

### 3.1 Asservissement visuel $2D\frac{1}{2}$

On a vu dans le chapitre précédent que le déplacement de la caméra et la structure de l'objet peuvent être estimés (à un facteur d'échelle près) sans connaissance a priori sur le modèle 3D de l'objet, ce qui élargit grandement le domaine d'application de l'asservissement visuel. En effet, la rotation de la caméra entre deux prises de vue de l'objet (courante et désirée) peut être calculée à chaque itération de la loi de commande. Par conséquent, la boucle d'asservissement en rotation peut être découplée de celle en translation, ce qui permet d'obtenir une forme simple du jacobien de la tâche. Afin de contrôler les degrés de liberté en translation, on introduit les coordonnées image étendues d'un point de référence de la cible. Celles-ci sont obtenues à partir des coordonnées image métriques classiques, et en ajoutant une troisième coordonnée normalisée qui peut être mesurée à partir de la reconstruction partielle. Cette troisième coordonnée n'est rien d'autre qu'un rapport entre deux distances. En effet, seuls les rapports entre distances peuvent être utilisés dans la loi de commande car les distances ne sont estimées qu'à un facteur d'échelle près. Grâce à un choix judicieux de la troisième coordonnée, on peut construire une fonction de tâche telle que son jacobien soit triangulaire par bloc et ne présente pas de singularité. La structure de la boucle fermée du système est alors particulièrement simple et il est possible d'obtenir des résultats analytiques sur la stabilité du système. En effet, si les paramètres intrinsèques de la caméra ne sont pas parfaitement connus, la fonction de tâche estimée peut se calculer en fonction des erreurs de calibration. On peut donc, grâce à la structure triangulaire du système en boucle fermée, trouver de manière analytique le domaine de robustesse des lois de commande proposées. Ce domaine est défini par des conditions nécessaires et suffisantes pour la stabilité asymptotique globale en présence d'erreurs de calibration de la caméra. De plus, même si les paramètres

de calibration caméra/effecteur ne sont pas parfaitement connus, la boucle de commande en rotation est toujours découplée de celle en translation. Le système en boucle fermée étant encore triangulaire par bloc, il est toujours possible de prouver de manière analytique la robustesse de la stabilité de notre méthode d'asservissement. Enfin, encore une fois grâce à la structure simple du système, il est possible d'utiliser une loi de commande adaptative sur les gains du contrôleur proportionnel afin d'élargir le domaine de robustesse et de prendre en compte la condition de visibilité de la tâche directement dans la loi de commande.

### 3.1.1 Modélisation

On décrit le choix de la fonction de tâche qui sera utilisée afin de positionner la caméra par rapport à un objet inconnu. Le contrôle étant effectué en partie dans l'image et en partie dans l'espace cartésien, nous avons nommé cette technique asservissement visuel  $2D\frac{1}{2}$ .

#### 3.1.1.1 Contrôle de l'orientation

Afin de contrôler l'orientation de la caméra, on utilise naturellement la rotation  ${}^d\mathbf{R}_c$  entre les repères désiré  $\mathcal{F}^*$  et courant  $\mathcal{F}$  (qui doit devenir l'identité). En notant  $\mathbf{u}$  l'axe et  $\theta$  l'angle de rotation entre  $\mathcal{F}^*$  et  $\mathcal{F}$ , nous avons choisi le vecteur  $\mathbf{u}\theta$  (avec  $\|\mathbf{u}\| = 1$  et  $0 \leq \theta < 2\pi$ ) comme représentation de la rotation de la caméra entre les deux repères. La dérivée de  $\mathbf{u}\theta$  peut s'écrire en fonction du torseur cinématique  $\mathbf{v} = [ \mathbf{v}^T \quad \boldsymbol{\omega}^T ]^T$  de la caméra :

$$\frac{d(\mathbf{u}\theta)}{dt} = [ \mathbf{0} \quad \mathbf{L}_\omega ] \mathbf{v} \quad (3.1)$$

avec :

$$\mathbf{L}_\omega(\mathbf{u}, \theta) = \mathbf{I} - \frac{\theta}{2} [\mathbf{u}]_\times + \left( 1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})} \right) [\mathbf{u}]_\times^2 \quad (3.2)$$

où  $\text{sinc}(\theta)$  est la fonction sinus cardinal définie par :

$$\text{sinc}(\theta) = \begin{cases} 1 & \text{si } \theta = 0 \\ \frac{\sin(\theta)}{\theta} & \text{si } \theta \neq 0 \end{cases} \quad (3.3)$$

Le calcul de la matrice  $\mathbf{L}_\omega$  est décrit en Annexe A. Cette matrice est définie positive sauf pour  $\theta = k\pi \quad \forall k \in \mathbb{Z}^*$  (elle est semi-définie positive en  $\theta = (2k+1)\pi \quad \forall k \in \mathbb{Z}^*$  et elle n'est pas définie en  $\theta = 2k\pi \quad \forall k \in \mathbb{Z}^*$ ). Comme on le verra par la suite, le fait que la matrice soit semi-définie positive en  $\pi$  n'empêche pas la décroissance de l'erreur en rotation quelle que soit la valeur de  $\theta$ . Le déterminant de  $\mathbf{L}_\omega$  est donné par :

$$\det(\mathbf{L}_\omega) = 1/\text{sinc}^2(\frac{\theta}{2}) \quad (3.4)$$

Les singularités sont donc situés en  $\theta = 2k\pi \quad \forall k \in \mathbb{Z}^*$ , soit en dehors de notre espace de travail.

On peut aussi noter, vue la forme de l'équation (3.1), que le contrôle de l'orientation de la caméra peut être découplé du contrôle de la translation. Finalement, pour des angles  $\theta$  de faible amplitude, on a  $\mathbf{L}_\omega \approx \mathbf{I}$ .

### 3.1.1.2 Contrôle de la position

La position de la caméra peut être contrôlée conjointement dans l'espace cartésien et dans l'image. Pour garder la cible dans le champ de vision de la caméra, nous avons tout d'abord proposé d'utiliser trois informations visuelles indépendantes, à savoir les deux coordonnées d'un point de la cible dans l'image, et le rapport  $r = \frac{d}{d^*}$ , calculé à partir de l'équation (2.42) [Malis 97]. Toutefois, l'expression analytique du jacobien de la tâche n'est alors pas simple, ce qui rend très difficile l'analyse de la stabilité du schéma d'asservissement correspondant. On décrit à présent une méthode plus satisfaisante.

Considérons un point  $\mathbf{x}$ . Il est bien connu que la dérivée de ses coordonnées, dans le repère caméra courant  $\mathcal{F}$ , est donnée par :

$$\dot{\mathbf{x}} = \begin{bmatrix} -\mathbf{I}_3 & [\mathbf{x}]_{\times} \end{bmatrix} \mathbf{v} \quad (3.5)$$

où  $[\mathbf{x}]_{\times}$  est la matrice de pré-produit vectoriel associée au vecteur  $\mathbf{x}$ . En utilisant les équations de la projection perspective,  $\mathbf{x}$  se projette en un point image de coordonnées métriques  $\mathbf{m} = [x \ y \ 1]^T = [\frac{X}{Z} \ \frac{Y}{Z} \ 1]^T$ . On définit alors les *coordonnées image étendues* métriques  $\mathbf{m}_e$  de la manière suivante :

$$\mathbf{m}_e = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \\ \ln(Z) \end{bmatrix} \quad (3.6)$$

où  $z = \ln(Z)$  est une coordonnée supplémentaire. Même si la distance  $Z$  est inconnue, la différence  $z - z^* = \ln(Z/Z^*)$  peut être calculée à partir de l'équation (2.43).

La dérivée des coordonnées étendues peut s'écrire :

$$\dot{\mathbf{m}}_e = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{\dot{X}}{Z} - \frac{X}{Z} \frac{\dot{Z}}{Z} \\ \frac{\dot{Y}}{Z} - \frac{Y}{Z} \frac{\dot{Z}}{Z} \\ \frac{\dot{z}}{Z} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} 1 & 0 & -\frac{X}{Z} \\ 0 & 1 & -\frac{Y}{Z} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = -\mathbf{L}_v(\mathbf{m}_e) \dot{\mathbf{x}} \quad (3.7)$$

où  $\mathbf{L}_v(\mathbf{m}_e)$  est une matrice triangulaire supérieure :

$$\mathbf{L}_v(\mathbf{m}_e) = e^{-z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \\ 0 & 0 & -1 \end{bmatrix} \quad (3.8)$$

Le déterminant de cette matrice est :

$$\det(\mathbf{L}_v(\mathbf{m}_e)) = 1/Z^3 \quad (3.9)$$

et  $\mathbf{L}_v$  est donc singulière pour  $Z = 0$  et  $Z \rightarrow \infty$ . À partir des équations (3.5) et (3.7), on en déduit :

$$\dot{\mathbf{m}}_e = \begin{bmatrix} \mathbf{L}_v(\mathbf{m}_e) & \mathbf{L}_{(v,\omega)}(\mathbf{m}_e) \end{bmatrix} \mathbf{v} \quad (3.10)$$

où, comme  $\mathbf{x} = Z\mathbf{m}$  :

$$\mathbf{L}_{(v,\omega)}(\mathbf{m}_e) = Z \mathbf{L}_v(\mathbf{m}_e) [\mathbf{m}]_{\times} = \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \\ -y & x & 0 \end{bmatrix} \quad (3.11)$$

Les équations (3.1) et (3.10) peuvent se regrouper de la manière suivante :

$$\frac{d}{dt} \begin{bmatrix} \mathbf{m}_e \\ \mathbf{u}\theta \end{bmatrix} = \begin{bmatrix} \mathbf{L}_v & \mathbf{L}_{(v,\omega)} \\ \mathbf{0} & \mathbf{L}_{\omega} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{L}(\mathbf{m}_e)\mathbf{v} \quad (3.12)$$

avec :

$$\mathbf{L}(\mathbf{m}_e) = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & (1+y^2) & -xy & -x \\ 0 & 0 & -\frac{1}{Z} & -y & x & 0 \\ 0 & 0 & 0 & 1 - \beta(1+u_x^2) & \beta u_y u_x - \alpha u_z & \beta u_z u_x + \alpha u_y \\ 0 & 0 & 0 & \beta u_y u_x + \alpha u_z & 1 - \beta(1+u_y^2) & \beta u_z u_y - \alpha u_x \\ 0 & 0 & 0 & \beta u_z u_x - \alpha u_y & \beta u_z u_y + \alpha u_x & 1 - \beta(1+u_z^2) \end{bmatrix} \quad (3.13)$$

où  $\alpha = -\frac{\theta}{2}$  et  $\beta = 1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})}$ . La distance  $d^*$  est le seul paramètre inconnu de la matrice car on a  $Z = \rho d^*$  (voir équation (2.44)). Si le point  $\mathbf{m}_e$  n'appartient pas au plan de référence  $\pi$ , comme le rapport  $Z/Z^*$  peut être estimé (voir équation (2.43)), il peut être judicieux de tout exprimer en fonction de la profondeur  $Z^*$  (seule inconnue) car les calculs sont plus simples.

La matrice  $\mathbf{L}$  est donc triangulaire par bloc et toujours de rang plein sauf si  $Z = 0$ ,  $Z \rightarrow \infty$  ou  $\theta = 2k\pi \quad \forall k \in \mathbb{Z}^*$ . Comme nous le verrons par la suite, la forme particulière de  $\mathbf{L}$  permet de construire une loi de commande avec des propriétés particulièrement intéressantes.

On définit aussi les *coordonnées image étendues*  $\mathbf{p}_e$ , exprimées en pixels, de la manière suivante :

$$\mathbf{p}_e = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{B} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{p}_{e_0} \quad (3.14)$$

où  $\mathbf{p}_{e_0} = [u_0 \ v_0 \ 0]^T$  est un vecteur contenant les coordonnées du point principal et  $\mathbf{B}$  est la matrice de passage entre les coordonnées image étendues en mètres et en pixel :

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_{11} & 0 \\ 0 & 1 \end{bmatrix} \quad (3.15)$$

où  $\mathbf{A}_{11}$  est la sous-matrice ( $2 \times 2$ ) de  $\mathbf{A}$  contenant les dimensions des pixels (voir équation (1.8)). On obtient alors :

$$\dot{\mathbf{p}}_e = \mathbf{B} \dot{\mathbf{m}}_e = [ \mathbf{B} \mathbf{L}_v \quad \mathbf{B} \mathbf{L}_{(v,\omega)} ] \mathbf{v} \quad (3.16)$$

Suivant les besoins, on exprimera par la suite la fonction de tâche à partir des coordonnées image étendues exprimées en mètres ou en pixels.

### 3.1.2 Commande proportionnelle

Comme on l'a vu précédemment, une tâche générique de positionnement contrôlant les six degrés de liberté de la caméra peut être décrite par la régulation d'une fonction de tâche [Samson 91]. Dans notre cas, cette fonction de tâche est définie par :

$$\mathbf{e} = \begin{bmatrix} \mathbf{m}_e - \mathbf{m}_e^* \\ \mathbf{u}\theta \end{bmatrix} \quad (3.17)$$

où  $\mathbf{m}_e - \mathbf{m}_e^* = \left[ x - x^* \quad y - y^* \quad \ln\left(\frac{Z}{Z^*}\right) \right]^T$  peut être calculé à partir des images courante et désirée, et du rapport  $\frac{Z}{Z^*}$  (ou  $\rho$  si le point 3D correspondant à  $\mathbf{m}_e$  appartient à  $\pi$ ).

#### 3.1.2.1 Loi de commande

La dérivée de la fonction de tâche est liée à la vitesse de l'effecteur du robot par la relation :

$$\dot{\mathbf{e}} = \mathbf{L}\mathbf{W}\dot{\mathbf{r}} \quad (3.18)$$

où  $\mathbf{L}$  est la matrice donnée par l'équation (3.13) et  $\mathbf{W}$  est la matrice de passage entre la vitesse de la caméra et la vitesse de l'effecteur du robot donnée par l'équation (1.29). En supposant la cible immobile, la loi de commande proportionnelle classique pour obtenir une décroissance exponentielle de  $\mathbf{e}$  est :

$$\dot{\mathbf{e}} = -\lambda\mathbf{D}\mathbf{e} \quad (3.19)$$

où  $\lambda$  est un scalaire positif et  $\mathbf{D}$  est une matrice diagonale positive,  $\lambda$  et  $\mathbf{D}$  réglant la vitesse de convergence. À partir de (3.18) et (3.19), la vitesse de l'effecteur du robot est donc :

$$\dot{\mathbf{r}} = -\lambda\widehat{\mathbf{W}}^{-1}\widehat{\mathbf{L}}^{-1}\mathbf{D}\widehat{\mathbf{e}} \quad (3.20)$$

où  $\widehat{\mathbf{e}}$  est la fonction de tâche mesurée et  $\widehat{\mathbf{L}}$  et  $\widehat{\mathbf{W}}$  sont respectivement des approximations des matrices  $\mathbf{L}$  et  $\mathbf{W}$  obtenues en utilisant des mesures ou des approximations au lieu des données réelles. En particulier, comme  $\mathbf{L}$  dépend de la distance inconnue  $d^*$ , une valeur approximative  $\widehat{d}^*$  doit être choisie pendant la phase d'apprentissage et introduite dans  $\widehat{\mathbf{L}}^{-1}$ .

Toutefois, il n'est pas nécessaire que cette valeur soit déterminée avec une très grande précision car elle n'a pas une grande influence sur la stabilité du système. Plus précisément, elle influence le temps de convergence de la partie en translation et l'amplitude de l'erreur de poursuite due à une mauvaise compensation du mouvement en rotation. En ce qui concerne cette erreur de poursuite (qui existe si  $\widehat{d}^* \neq d^*$ ), elle est proportionnelle à la vitesse de rotation, et disparaît donc quand la caméra est orientée correctement. Nous verrons par la suite, lors de l'étude de la stabilité du système, quelles sont les plages d'erreur possible pour  $\widehat{d}^*$ .

Enfin, si on choisit  $\mathbf{D} = \mathbf{I}$ , la loi de commande est donnée par :

$$\dot{\mathbf{r}} = -\lambda \begin{bmatrix} {}^c\widehat{\mathbf{R}}_e^T & -{}^c\widehat{\mathbf{R}}_e^T \begin{bmatrix} {}^c\widehat{\mathbf{t}}_e \\ \times \end{bmatrix} \\ 0 & {}^c\widehat{\mathbf{R}}_e^T \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{L}}_v^{-1} & -\widehat{\mathbf{L}}_v^{-1}\widehat{\mathbf{L}}_{(v,\omega)}\widehat{\mathbf{L}}_\omega^{-1} \\ \mathbf{0} & \widehat{\mathbf{L}}_\omega^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{m}}_e - \widehat{\mathbf{m}}_e^* \\ \widehat{\mathbf{u}}\widehat{\theta} \end{bmatrix} \quad (3.21)$$

De plus, grâce à la forme particulière de la matrice  $\widehat{\mathbf{L}}_{\omega}^{-1}$  (dont le calcul est décrit en Annexe A), nous avons :

$$\widehat{\mathbf{L}}_{\omega}^{-1} \widehat{\mathbf{u}} \widehat{\theta} = \left( \mathbf{I} + \frac{\widehat{\theta}}{2} \text{sinc}^2 \left( \frac{\widehat{\theta}}{2} \right) [\widehat{\mathbf{u}}]_{\times} + (1 - \text{sinc}(\widehat{\theta})) [\widehat{\mathbf{u}}]_{\times}^2 \right) \widehat{\mathbf{u}} \widehat{\theta} = \widehat{\mathbf{u}} \widehat{\theta} \quad (3.22)$$

La loi de commande en rotation peut alors se calculer de manière très simple :  $\boldsymbol{\omega} = -\lambda \widehat{\mathbf{u}} \widehat{\theta}$ , et nous pouvons poser  $\widehat{\mathbf{L}}_{\omega}^{-1} = \mathbf{I}$  sans perte de généralité.

Le schéma bloc de notre loi de commande est présenté sur la Figure 3.1. Un tel système découplé permet d'obtenir la convergence dans tout l'espace de travail si on suppose le système parfaitement calibré et l'objet d'intérêt toujours visible. De plus, contrairement aux asservissements visuels 2D et 3D, on montrera par la suite qu'il est possible d'obtenir des conditions nécessaires et suffisantes pour la stabilité asymptotique locale, des conditions suffisantes pour la stabilité asymptotique globale en présence d'erreurs de calibration de la caméra, et des conditions suffisantes pour la stabilité asymptotique locale en présence d'erreurs de calibration de la matrice de passage caméra/effecteur.

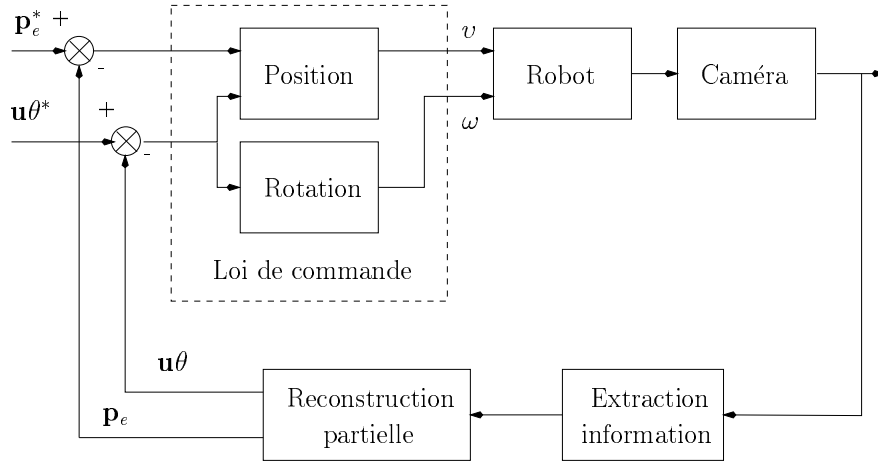


FIG. 3.1 – Schéma bloc de l'asservissement visuel  $2D\frac{1}{2}$

### 3.1.2.2 Boucle fermée

Afin d'obtenir l'équation différentielle caractérisant le système en boucle fermée, il est nécessaire d'exprimer la fonction de tâche estimée en fonction de la fonction de tâche réelle et des erreurs de calibration. Les coordonnées image étendues estimées sont :

$$\widehat{\mathbf{m}}_e = \widehat{\mathbf{B}}^{-1} \widehat{\mathbf{p}}_e - \widehat{\mathbf{B}}^{-1} \widehat{\mathbf{p}}_{e_0} \quad (3.23)$$

et, comme  $\widehat{\mathbf{p}}_e = \mathbf{p}_e$  car on suppose le bruit de mesure dans l'image négligeable, on a :

$$\widehat{\mathbf{p}}_e = \mathbf{p}_e = \mathbf{B} \mathbf{m}_e + \mathbf{p}_{e_0} \quad (3.24)$$

En substituant la relation (3.24) dans l'équation (3.23), on obtient :

$$\widehat{\mathbf{m}}_e = \delta \mathbf{B} \mathbf{m}_e + \widehat{\mathbf{B}}^{-1} (\mathbf{p}_{e_0} - \widehat{\mathbf{p}}_{e_0}) \quad (3.25)$$

où (en supposant par simplicité  $\mathbf{m} \in \pi$ ) :

$$\delta \mathbf{B} = \widehat{\mathbf{B}}^{-1} \mathbf{B} = \begin{bmatrix} \delta \mathbf{A}_{11} & 0 \\ 0 & 1 \end{bmatrix} \quad (3.26)$$

avec  $\delta \mathbf{A}_{11}$  la sous-matrice ( $2 \times 2$ ) de  $\delta \mathbf{A}$  contenant les dimensions des pixels (voir équation (2.45)). La relation (3.25) est valable pour n'importe quel point de l'image. En l'appliquant pour le point désiré, on a :

$$\widehat{\mathbf{m}}_e^* = \delta \mathbf{B} \mathbf{m}_e^* + \widehat{\mathbf{B}}^{-1} (\mathbf{p}_{e_0} - \widehat{\mathbf{p}}_{e_0}) \quad (3.27)$$

et on en déduit :

$$\widehat{\mathbf{m}}_e - \widehat{\mathbf{m}}_e^* = \delta \mathbf{B} (\mathbf{m}_e - \mathbf{m}_e^*) \quad (3.28)$$

qui est indépendant de  $\mathbf{p}_{e_0}$ . La fonction de tâche peut donc être enfin reconstruite de la manière suivante :

$$\widehat{\mathbf{e}} = \mathbf{E} \mathbf{e} = \begin{bmatrix} \mathbf{E}_v & 0 \\ 0 & \mathbf{E}_\omega \end{bmatrix} \mathbf{e} \quad (3.29)$$

avec  $\mathbf{E}_v = \delta \mathbf{B}$  et (voir équation (2.50)) :

$$\mathbf{E}_\omega = \mu \begin{bmatrix} \delta \mathbf{A}_{11} & \delta \mathbf{p}_0 \\ 0 & 1 \end{bmatrix} = \mu \delta \mathbf{A} \quad (3.30)$$

où  $\delta \mathbf{p}_0$  est le sous-vecteur contenant l'erreur sur le point principal et  $\mu = \frac{1}{\|\delta \mathbf{A} \mathbf{u}\|}$ . Les déterminants de  $\mathbf{E}_v$  et  $\mathbf{E}_\omega$  sont donnés par :

$$\det(\mathbf{E}_v) = \frac{\alpha_u \alpha_v}{\widehat{\alpha}_u \widehat{\alpha}_v} \quad (3.31)$$

$$\det(\mathbf{E}_\omega) = \mu^3 \frac{\alpha_u \alpha_v}{\widehat{\alpha}_u \widehat{\alpha}_v} \quad (3.32)$$

Le système en boucle fermée est donc :

$$\dot{\mathbf{e}} = \mathbf{f}(\mathbf{e}) = -\lambda \mathbf{Q}(\mathbf{e}) \mathbf{e} = -\lambda \mathbf{L} \mathbf{W} \widehat{\mathbf{W}}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{E} \mathbf{e} \quad (3.33)$$

La fonction  $\mathbf{f}$  est de classe  $C^\infty$  et définie dans un sous-ensemble  $S$  de  $SE_3$ . Il est facile de montrer l'existence et l'unicité du point d'équilibre :

**Proposition 1** : *Le seul point d'équilibre pour  $\mathbf{f}$ , c'est-à-dire un point  $\mathbf{e}^\circ \in S$  tel que  $\mathbf{f}(\mathbf{e}^\circ) = 0$ , est  $\mathbf{e}^\circ = 0$ .*



**Preuve de la proposition 1** *L'existence du point d'équilibre est évidente car si  $\mathbf{e} = 0$ , alors  $\mathbf{f}(\mathbf{e}) = \mathbf{Q}(\mathbf{e})\mathbf{e} = 0$ . Ce point d'équilibre est unique si et seulement si  $\det(\mathbf{Q}(\mathbf{e})) \neq 0$ ,  $\forall \mathbf{e} \in S$ .*

*Comme la matrice  $\mathbf{Q}$  est triangulaire par bloc, son déterminant est calculé facilement :*

$$\det(\mathbf{Q}) = \det(\mathbf{L}) \det(\mathbf{W}) \det(\widehat{\mathbf{W}}^{-1}) \det(\widehat{\mathbf{L}}^{-1}) \det(\mathbf{E}) \quad (3.34)$$

*Comme  $\det(\mathbf{W}) = \det(\widehat{\mathbf{W}}^{-1}) = 1$ ,  $\det(\mathbf{L}) = \det(\mathbf{L}_v) \det(\mathbf{L}_\omega)$ ,  $\det(\widehat{\mathbf{L}}^{-1}) = \det(\widehat{\mathbf{L}}_v^{-1}) \det(\widehat{\mathbf{L}}_\omega^{-1})$  (avec  $\det(\widehat{\mathbf{L}}_\omega^{-1}) = 1$ ) et  $\det(\mathbf{E}) = \det(\mathbf{E}_v) \det(\mathbf{E}_\omega)$ , on obtient :*

$$\det(\mathbf{Q}) = \det(\mathbf{L}_\omega) \det(\mathbf{L}_v) \det(\widehat{\mathbf{L}}_v^{-1}) \det(\mathbf{E}_v) \det(\mathbf{E}_\omega) \quad (3.35)$$

*soit finalement (voir équations (3.4), (3.9), (3.31), et (3.32)) :*

$$\det(\mathbf{Q}) = \nu^3 \mu^3 \frac{\alpha_u^2 \alpha_v^2}{\widehat{\alpha}_u^2 \widehat{\alpha}_v^2} \text{sinc}^2 \left( \frac{\theta}{2} \right) \quad (3.36)$$

*avec  $\nu = \frac{\widehat{d}^* \widehat{\rho}}{d^* \rho} = \frac{\widehat{d}^*}{d^*} \|\delta \mathbf{A}^{-T} \mathbf{n}^*\|$ . Donc  $\det(\mathbf{Q}) \neq 0, \forall \mathbf{e} \in S$  car  $\nu \neq 0$ ,  $\mu \neq 0$  et  $\text{sinc}^2 \left( \frac{\theta}{2} \right) \neq 0$ ,  $\forall \theta : -2\pi < \theta < 2\pi$ .*

En conclusion, la tâche ne présente pas de singularité, et si la fonction de tâche décroît, elle décroît vers 0 en absence d'erreurs de mesure.

Si le point de référence est choisi tel qu'il appartienne au plan de référence  $\pi$ , le seul cas dégénéré se présente si  $d^* = 0$  ou  $\rho = 0$ . Le cas  $d^* = 0$  apparaît quand le centre optique de la caméra appartient au plan de référence choisi (les points de ce plan se projettent alors sur une droite de l'image). Dans le cas d'un objet plan, la convergence est alors assurée dans le demi-espace en face de la cible. Dans le cas d'un objet non plan, la convergence peut être assurée dans tout l'espace en choisissant un autre plan de référence dès que le système s'approche de ce cas dégénéré. Le seul cas singulier se présente alors si  $\rho = 0$ , soit  $Z = 0$ .

Il est évident que, dans le cas où les paramètres géométriques du système sont parfaitement connus, on a  $\mathbf{Q} = \mathbf{I}$  si  $\mathbf{D} = \mathbf{I}$  et le système est parfaitement découplé et globalement asymptotiquement stable. Toutefois, dans la réalité, on ne peut pas connaître de manière parfaite tous les paramètres (surtout la distance  $d^*$ ) et une analyse de la robustesse de la stabilité du système en boucle fermée est nécessaire. On s'intéresse tout d'abord à la robustesse aux erreurs de calibration de la caméra, et ensuite à la robustesse aux erreurs de calibration de la caméra et de la matrice de passage caméra/effecteur.

### 3.1.2.3 Analyse de la robustesse de la stabilité aux erreurs sur les paramètres caméra

On va supposer pour le moment que la calibration des paramètres caméra/effecteur est parfaite ( $\mathbf{W} = \widehat{\mathbf{W}}$ ). Les seules erreurs possibles sont donc sur les paramètres intrinsèques de

la caméra et sur la distance approximée  $\hat{d}^*$ . La matrice  $\mathbf{Q}(\mathbf{e})$  caractérisant la boucle fermée du système est alors donnée par :

$$\mathbf{Q}(\mathbf{e}) = \begin{bmatrix} \mathbf{L}_v \hat{\mathbf{L}}_v^{-1} \mathbf{E}_v & \mathbf{L}_v [\mathbf{x} - \hat{\mathbf{x}}]_{\times} \mathbf{E}_\omega \\ \mathbf{0} & \mathbf{L}_\omega \mathbf{E}_\omega \end{bmatrix} \quad (3.37)$$

### Stabilité asymptotique locale

Afin d'étudier la stabilité asymptotique locale du système en boucle fermée, on utilise la méthode de la linéarisation. Il est bien connu que l'étude de la stabilité asymptotique locale d'un système est équivalente à l'étude de la stabilité du système linéaire invariant obtenu par linéarisation du premier. Dans certains cas, ce type d'analyse peut nous permettre d'extrapoler des conclusions sur la stabilité du système non linéaire. Toutefois, elle a un caractère strictement local, et la stabilité du système est étudiée dans un voisinage du point d'équilibre. Si l'état initial est éloigné du point d'équilibre, nous n'avons donc aucune connaissance de l'évolution du système, et une nouvelle analyse plus globale doit être effectuée. Une solution pour remédier à ce problème est de créer une trajectoire dans l'espace d'état, telle que le point d'équilibre soit toujours dans le voisinage du point initial. Cependant, cette méthode n'est pas toujours évidente à mettre en œuvre (notamment pour déterminer la vitesse de parcours de cette trajectoire) et demande généralement un temps de convergence élevé.

En considérant, comme nous l'avons fait, un point d'équilibre constant, la condition nécessaire et suffisante pour assurer la stabilité asymptotique locale est que les valeurs propres de la matrice  $\mathbf{Q}(\mathbf{e})$  soient à partie réelle négative. Dans notre cas, on peut très facilement démontrer le théorème suivant :

#### Théorème 1

*Le point d'équilibre  $\mathbf{e}^o$  du système différentiel (3.33), caractérisé par la matrice (3.37), est localement asymptotiquement stable si et seulement si :*

$$\frac{\hat{\alpha}_u}{\alpha_u} > 0, \quad \frac{\hat{\alpha}_v}{\alpha_v} > 0 \quad \text{et} \quad \frac{\hat{d}^*}{d^*} > 0 \quad (3.38)$$

La démonstration de ce théorème est donnée en Annexe B. En pratique, les conditions (3.38) sont très larges et facilement vérifiées. On remarque que ces conditions sont aussi nécessaires et suffisantes pour la stabilité asymptotique globale du sous-système de contrôle de l'orientation. Toutefois, il est bien connu que l'erreur peut augmenter considérablement avant la convergence à zéro. Sans un contrôle supplémentaire, la cible peut donc sortir du champ de vue de la caméra durant l'asservissement. En effet, il faut rappeler que l'équation différentielle (3.33) est contrainte par la condition de visibilité de la tâche. Pour cette raison, il est plus intéressant de chercher des conditions suffisantes pour assurer la décroissance de l'erreur à chaque itération, ce qui assure la stabilité asymptotique globale du système et, en même temps, augmente la probabilité que la condition de visibilité soit respectée.

## Stabilité asymptotique globale

Les conditions nécessaires et suffisantes obtenues lors de l'analyse de la stabilité locale ne sont plus suffisantes pour garantir la stabilité globale du système non-linéaire.

On rappelle qu'une condition suffisante pour assurer la décroissance de la fonction de tâche à chaque itération est la positivité de la matrice  $\mathbf{Q}(\mathbf{e}) \forall \mathbf{e} \neq 0$ . En effet, en multipliant à droite l'équation (3.33) par  $\mathbf{e}^T$ , on obtient :

$$\mathbf{e}^T \dot{\mathbf{e}} = -\lambda \mathbf{e}^T \mathbf{Q}(\mathbf{e}) \mathbf{e} \quad (3.39)$$

Donc, si  $\mathbf{Q}(\mathbf{e}) > 0, \forall \mathbf{e} \neq 0$  et si le gain  $\lambda$  est positif, la dérivée temporelle de la norme de la fonction de tâche est négative :

$$\frac{1}{2} \frac{d\|\mathbf{e}\|}{dt} = -\lambda \mathbf{e}^T \mathbf{Q}(\mathbf{e}) \mathbf{e} < 0 \quad (3.40)$$

et l'erreur  $\mathbf{e}$  décroît à chaque itération. Comme le seul point d'équilibre du système est  $\mathbf{e}^\circ = 0$ , la fonction de tâche décroît vers zéro et le système non-linéaire est globalement asymptotiquement stable (c'est-à-dire que la fonction de tâche décroît vers zéro à partir de n'importe quelle position initiale).

Dans notre cas, l'analyse de la stabilité du système peut se simplifier car, grâce à la forme particulière de la matrice  $\mathbf{L}_\omega$  (voir équation (3.2)) et de  $\mathbf{e}_\omega = \mathbf{u}\theta$ , on a :

$$\mathbf{e}_\omega^T (\mathbf{L}_\omega \mathbf{E}_\omega) \mathbf{e}_\omega = \mathbf{e}_\omega^T \mathbf{E}_\omega \mathbf{e}_\omega \quad (3.41)$$

On a donc :

$$\mathbf{e}^T \mathbf{Q}(\mathbf{e}) \mathbf{e} = \mathbf{e}^T \mathbf{Q}'(\mathbf{e}) \mathbf{e} \quad (3.42)$$

avec :

$$\mathbf{Q}'(\mathbf{e}) = \begin{bmatrix} \mathbf{L}_v \widehat{\mathbf{L}}_v^{-1} \mathbf{E}_v & \mathbf{L}_v [\mathbf{x} - \widehat{\mathbf{x}}]_\times \mathbf{E}_\omega \\ \mathbf{0} & \mathbf{E}_\omega \end{bmatrix} \quad (3.43)$$

L'analyse de la stabilité du système différentiel (3.33), caractérisé par la matrice (3.37) se réduit à l'analyse de la positivité de la matrice (3.43). Les conditions sous lesquelles la matrice  $\mathbf{Q}'(\mathbf{e})$  est positive sont données dans le théorème suivant :

### Théorème 2

*Le point d'équilibre  $\mathbf{e}^\circ$  du système différentiel (3.33), caractérisé par la matrice (3.37), est globalement asymptotiquement stable, seulement si les conditions (3.38) sont vérifiées, et si :*

$$\sigma = \frac{\alpha_u}{\widehat{\alpha}_u} + \frac{\alpha_v}{\widehat{\alpha}_v} - \sqrt{\left(\frac{\alpha_u}{\widehat{\alpha}_u} - \frac{\alpha_v}{\widehat{\alpha}_v}\right)^2 + \left(\frac{\alpha_{uv}}{\widehat{\alpha}_u} - \frac{\widehat{\alpha}_{uv} \alpha_v}{\widehat{\alpha}_u \widehat{\alpha}_v}\right)^2} > 0 \quad (3.44)$$

$$\sigma_1 = \sigma + 1 - \sqrt{(\sigma - 1)^2 + \|\mathbf{I} - \delta\mathbf{A}\|^2(1 + \bar{\gamma}^2)} > 0 \quad (3.45)$$

$$\sigma_2 = \sigma + 1 - \sqrt{(\sigma - 1)^2 + \|\delta\mathbf{p}_0\|^2} > 0 \quad (3.46)$$

$$\mu(\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|)^2\|\delta\mathbf{A}\|^2g^2(\bar{\gamma}) < \nu\sigma_1\sigma_2 \quad (3.47)$$

où  $\bar{\gamma} = \tan(\psi)$  est la tangente de l'angle de vision  $\psi$  ( $\bar{\gamma}$  est la valeur maximale de  $\gamma = \sqrt{x^2 + y^2}$ ) et :

$$g^2(\bar{\gamma}) = \frac{2 + \bar{\gamma}^2 + \sqrt{\bar{\gamma}^2(\bar{\gamma}^2 + 4)}}{2}(1 + \bar{\gamma}^2) \quad (3.48)$$

La démonstration de ce théorème est donnée en Annexe B. Le principe en est assez simple :

- les conditions (3.44) et (3.45) représentent des conditions suffisantes pour la positivité de la matrice  $\mathbf{L}_v\hat{\mathbf{L}}_v^{-1}\mathbf{E}_v$ . La positivité de cette matrice, qui est le premier bloc de la diagonale de  $\mathbf{Q}'$ , est nécessaire pour la positivité de  $\mathbf{Q}'$ .
- les conditions (3.44) et (3.46) représentent des conditions suffisantes pour la positivité de la matrice  $\mathbf{E}_\omega$ . La positivité de cette matrice, qui est le deuxième bloc de la diagonale de  $\mathbf{Q}'$ , est aussi nécessaire pour la positivité de  $\mathbf{Q}'$ .
- la condition (3.47) représente enfin une condition suffisante pour la positivité de la matrice  $\mathbf{Q}'$ , à condition que les deux précédentes soient satisfaites.

Les conditions suffisantes peuvent être analysées pour avoir une idée du domaine de convergence. Toutefois, elles dépendent de  $\delta\mathbf{A}$  et  $\mu$  (qui dépend de l'axe de rotation  $\mathbf{u}$ ). Pour simplifier l'analyse, on présente maintenant un corollaire dont la démonstration est également donnée en Annexe B.

## Corollaire au théorème 2

Le système différentiel (3.33), caractérisé par la matrice (3.37), est asymptotiquement globalement stable, seulement si les conditions (3.38) sont vérifiés, et si :

$$\sigma > 0 \quad (3.49)$$

$$\sigma_3 = \sigma + 1 - \sqrt{(\sigma - 1)^2 + (\|\mathbf{I} - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\|)^2(1 + \bar{\gamma}^2)} > 0 \quad (3.50)$$

$$\bar{\mu}(\nu(\|\mathbf{I} - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\|) + |1 - \nu|)^2(\|\delta\mathbf{A}_{11}\| + \sqrt{1 + \|\delta\mathbf{p}_0\|^2})^2g^2(\bar{\gamma}) < \nu\sigma_2\sigma_3 \quad (3.51)$$

où :

$$\bar{\mu} = \frac{\sqrt{2}}{\sqrt{\sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2\|\delta\mathbf{p}_0\|^2}}} \quad (3.52)$$

et :

$$\sigma' = \frac{1}{2} \left( \frac{\alpha_u^2}{\hat{\alpha}_u^2} + \frac{\alpha_v^2}{\hat{\alpha}_v^2} + \left( \frac{\alpha_{uv}}{\hat{\alpha}_u} - \frac{\hat{\alpha}_{uv} \alpha_v}{\hat{\alpha}_u \hat{\alpha}_v} \right)^2 - \sqrt{\left( \frac{\alpha_u^2}{\hat{\alpha}_u^2} + \frac{\alpha_v^2}{\hat{\alpha}_v^2} + \left( \frac{\alpha_{uv}}{\hat{\alpha}_u} - \frac{\hat{\alpha}_{uv} \alpha_v}{\hat{\alpha}_u \hat{\alpha}_v} \right)^2 \right)^2 - 4 \frac{\alpha_u^2 \alpha_v^2}{\hat{\alpha}_u^2 \hat{\alpha}_v^2}} \right) \quad (3.53)$$

Ces conditions suffisantes sont plus restrictives que les conditions imposées par le Théorème 2. Toutefois, ces nouvelles conditions ne dépendent pas de l'axe de rotation  $\mathbf{u}$ , et l'erreur  $\delta\mathbf{p}_0$  sur le point principal est séparée de l'erreur  $\delta\mathbf{A}_{11}$  sur la taille des pixels. De plus, la condition (3.46) n'est plus nécessaire car elle est automatiquement vérifiée si la condition (3.50) est vérifiée. L'analyse est donc simplifiée même si les conditions semblent plus complexes.

### Positivité des blocs de la diagonale

Tout d'abord, on analyse la condition (3.49) plus en détail. Cette condition est équivalente à la suivante :

$$\left( \frac{\alpha_{uv} \alpha_u}{\alpha_u \hat{\alpha}_u} - \frac{\hat{\alpha}_{uv} \alpha_v}{\hat{\alpha}_u \hat{\alpha}_v} \right)^2 < 4 \frac{\alpha_u \alpha_v}{\hat{\alpha}_u \hat{\alpha}_v} \quad (3.54)$$

Comme  $\cot(\phi) = -\frac{\alpha_{uv}}{\alpha_u}$  (voir équation (1.8)), on a :

$$\left( \cot(\hat{\phi}) \frac{\alpha_v}{\hat{\alpha}_v} - \cot(\phi) \frac{\alpha_u}{\hat{\alpha}_u} \right)^2 < 4 \frac{\alpha_u \alpha_v}{\hat{\alpha}_u \hat{\alpha}_v} \quad (3.55)$$

Soit  $\delta\phi = \hat{\phi} - \phi$  l'erreur commise sur l'angle  $\phi$ . En général, l'angle  $\hat{\phi}$  est fixé à  $\pi/2$ . On montre maintenant que ce choix n'a pas une grande influence sur la stabilité du système. En effet, on a  $\cot(\hat{\phi}) = 0$ ,  $\cot(\phi) = \cot(\hat{\phi} - \delta\phi) = \cot(\pi/2 - \delta\phi) = \tan(\delta\phi)$  et la condition (3.55) peut s'écrire :

$$\tan^2(\delta\phi) < 4 \frac{\hat{\alpha}_u \alpha_v}{\hat{\alpha}_v \alpha_u} \quad \Rightarrow \quad |\delta\phi| < \arctan \left( 2 \sqrt{\frac{\alpha_v / \hat{\alpha}_v}{\alpha_u / \hat{\alpha}_u}} \right) \quad (3.56)$$

En Figure 3.2, sont illustrées les bornes pour  $|\delta\phi|$  en fonction du rapport  $\alpha_u / \hat{\alpha}_u$  sur l'axe  $\vec{x}$  et du rapport  $\alpha_v / \hat{\alpha}_v$  sur l'axe  $\vec{y}$ .

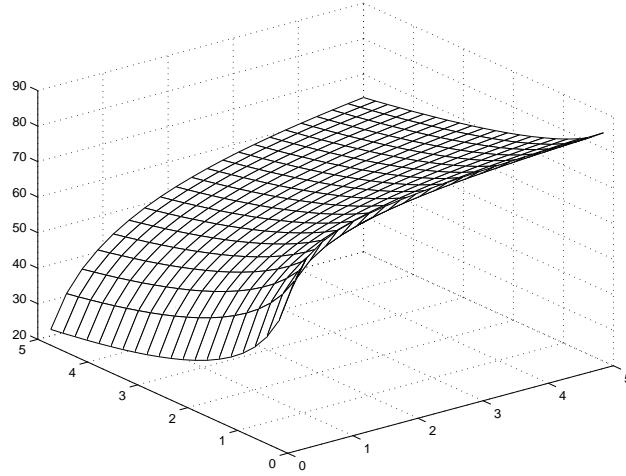


FIG. 3.2 – Bornes de stabilité pour  $|\delta\phi|$

Si les rapports sont bien estimés alors  $|\delta\phi| < 63$  dg. Dans le pire des cas considérés, quand le rapport estimé est  $\hat{\alpha}_u/\hat{\alpha}_v = 1/5$  et le rapport réel est  $\alpha_u/\alpha_v = 5$  (ce qui correspond à un pixel rectangulaire avec une largeur égale à cinq fois la hauteur calibrée comme un pixel dont la hauteur est égale à cinq fois la largeur!), on a  $|\delta\phi| < 22$  dg. Même dans un cas si extrême, cette condition peut être facilement respectée. Pour une caméra plus “normale” avec un rapport  $\alpha_u/\alpha_v = 3/4$ , si  $\hat{\alpha}_u/\hat{\alpha}_v = 4/3$  alors  $|\delta\phi| < 56$  dg, ce qui semble impossible à ne pas satisfaire.

On analyse maintenant la condition (3.50) sous l’hypothèse que la condition (3.49) est satisfaite. Cette condition peut être écrite de la manière suivante :

$$\|\mathbf{I} - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\| < 2\sqrt{\sigma}/\sqrt{1 + \bar{\gamma}^2} \quad (3.57)$$

ce qui implique :

$$\|\delta\mathbf{p}_0\| < 2\sqrt{\sigma}/\sqrt{1 + \bar{\gamma}^2} - \|\mathbf{I} - \delta\mathbf{A}_{11}\| \quad (3.58)$$

En Figure 3.3, les bornes pour  $\|\delta\mathbf{p}_0\|$  sont tracées en fonction du rapport  $\alpha_u/\hat{\alpha}_u$  sur l’axe  $\vec{x}$  et en fonction du rapport  $\alpha_v/\hat{\alpha}_v$  sur l’axe  $\vec{y}$  (pour obtenir ces résultats, on a fixé  $\bar{\gamma} = 0.364$ , ce qui correspond à une caméra avec un angle de vision de 20 dg, et  $\delta\phi = 20$  dg).

Dans le pire des cas considérés, quand  $\alpha_u/\hat{\alpha}_u = 0.5$  et  $\alpha_v/\hat{\alpha}_v = 2$  alors  $\|\delta\mathbf{p}_0\| < 0.44$ . Comme  $\|\delta\mathbf{p}_0\| = \sqrt{\hat{x}_0^2 + \hat{y}_0^2}$ , le vecteur 3D associé au point principal estimé doit être dans un cône d’ouverture  $\psi_0 < 24$  deg autour du vecteur exact.

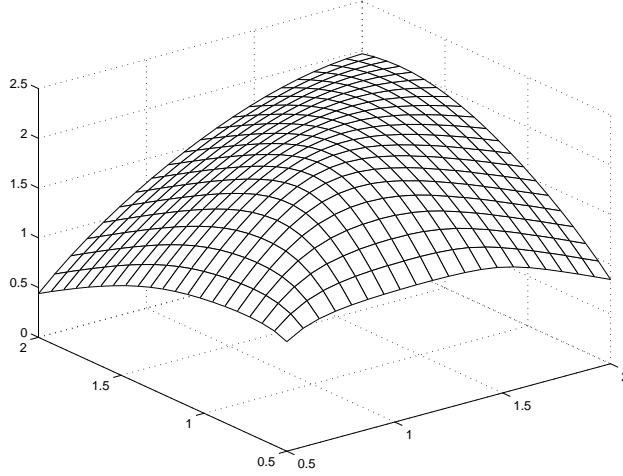


FIG. 3.3 – Bornes de stabilité pour  $\|\delta\mathbf{p}_0\|$

### Positivité de la matrice entière

Les conditions qu'on vient d'analyser sont facilement vérifiées en pratique. Toutefois, le domaine de stabilité n'est pas si grand car la condition (3.51) doit également être vérifiée. Cette dernière condition peut être écrite comme une inégalité du second degré (on rappelle que  $\nu = \|\delta\mathbf{A}^{-T}\mathbf{n}^*\|d^*/\hat{d}^*$ ) :

$$c_2\nu^2 + c_1\nu + c_0 < 0 \quad (3.59)$$

avec :

$$c_2 = \sqrt{2} \frac{(\|\mathbf{I} - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\| - \text{sign}(1 - \nu))^2 (\|\delta\mathbf{A}_{11}\| + \sqrt{1 + \|\delta\mathbf{p}_0\|^2})^2 g^2(\bar{\gamma})}{\sqrt{\sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2\|\delta\mathbf{p}_0\|^2}}} \quad (3.60)$$

$$c_1 = 2\sqrt{2}\text{sign}(1 - \nu) \frac{(\|\mathbf{I} - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\| - \text{sign}(1 - \nu))(\|\delta\mathbf{A}_{11}\| + \sqrt{1 + \|\delta\mathbf{p}_0\|^2})^2 g^2(\bar{\gamma})}{\sqrt{\sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2\|\delta\mathbf{p}_0\|^2}}} - \sigma_2\sigma_3 \quad (3.61)$$

$$c_0 = \sqrt{2} \frac{(\|\delta\mathbf{A}_{11}\| + \sqrt{1 + \|\delta\mathbf{p}_0\|^2})^2 g^2(\bar{\gamma})}{\sqrt{\sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2\|\delta\mathbf{p}_0\|^2}}} \quad (3.62)$$

Les coefficients de l'inégalité (3.59) dépendent du signe de  $1 - \nu$  qui peut être obtenu après le calcul de  $\nu$ . En fixant par exemple  $1 - \nu$  positif, la solution de l'inégalité du second degré est donnée par :

$$\nu_{min} = \frac{-c_1 - \sqrt{c_1^2 - 4c_0c_2}}{2c_0} < \nu < \frac{-c_1 + \sqrt{c_1^2 - 4c_0c_2}}{2c_0} = \nu_{max} \quad (3.63)$$

On va voir à présent plusieurs exemples pour analyser cette inégalité.

### Exemple 1 : distance $\hat{d}^*$ grossièrement connue

En premier exemple, on suppose une calibration de la caméra parfaite ( $\delta \mathbf{A}_{11} = \mathbf{I}_2$ ,  $\delta \mathbf{p}_0 = \mathbf{0}$ ,  $\sigma = \sigma_1 = \sigma_2 = 2$ ,  $\sigma' = 1$  et  $\nu = \frac{\hat{d}^*}{d^*}$ ). Dans ce cas, la condition (3.51) peut s'écrire :

$$\left(1 - \frac{\hat{d}^*}{d^*}\right)^2 g^2(\bar{\gamma}) < 4 \frac{\hat{d}^*}{d^*} \quad (3.64)$$

La solution de l'inégalité (3.64) est :

$$1 + 2 \frac{1 - \sqrt{g^2(\bar{\gamma}) + 1}}{g^2(\bar{\gamma})} < \frac{\hat{d}^*}{d^*} < 1 + 2 \frac{1 + \sqrt{g^2(\bar{\gamma}) + 1}}{g^2(\bar{\gamma})} \quad (3.65)$$

Les deux bornes correspondantes sont tracées en fonction de  $\bar{\gamma}$  sur la Figure 3.4.

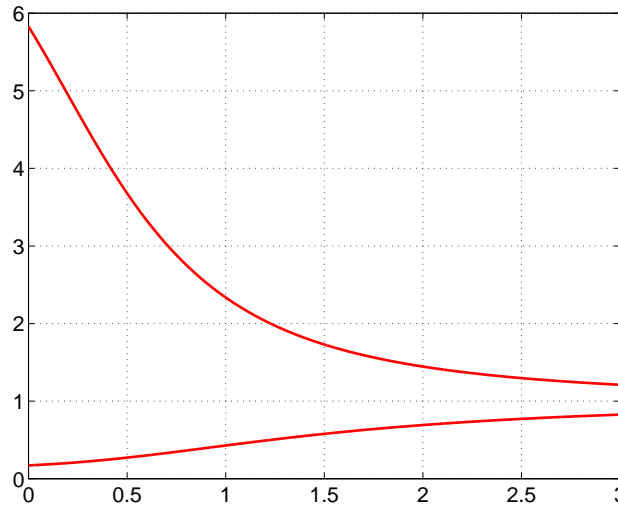


FIG. 3.4 – Bornes de stabilité pour la distance estimée

Ces résultats signifient que, si on considère une caméra avec un angle de vision de  $20^\circ$  (donc  $\bar{\gamma} = 0.364$ ), la condition de stabilité est vérifiée si  $\hat{d}^*/d^*$  est compris entre 0.24 et 4.22. Si la distance réelle  $d^*$  est 50 cm, le système converge asymptotiquement pour n'importe quelle position initiale dans l'espace de travail si  $\hat{d}^*$  est choisi entre 12 et 211 cm.

Ce résultat confirme définitivement la robustesse de notre loi de commande dans ce cas particulier. De plus, des résultats similaires peuvent être obtenus en considérant des erreurs de calibration de la caméra. Il est toutefois évident que le domaine de robustesse sera plus petit.

La condition (3.51) dépend des cinq paramètres intrinsèques de la caméra. On étudie à présent la stabilité avec  $\delta \mathbf{A}_{11}$  fixe et  $\delta \mathbf{p}_0$  variable, puis avec  $\delta \mathbf{A}_{11}$  variable et  $\delta \mathbf{p}_0$  fixe.



## Exemple 2 : distance $\hat{d}^*$ et $\delta\mathbf{A}_{11}$ grossièrement connus

On considère d'abord  $\alpha_u/\hat{\alpha}_u = \alpha_v/\hat{\alpha}_v = 1.5$  (ce qui signifie une erreur de 50% sur chaque dimension des pixels) et  $\delta\phi = 5$  dg. Les deux bornes de stabilité sont illustrées en Figure 3.5 en fonction de  $\|\delta\mathbf{p}_0\|$ .

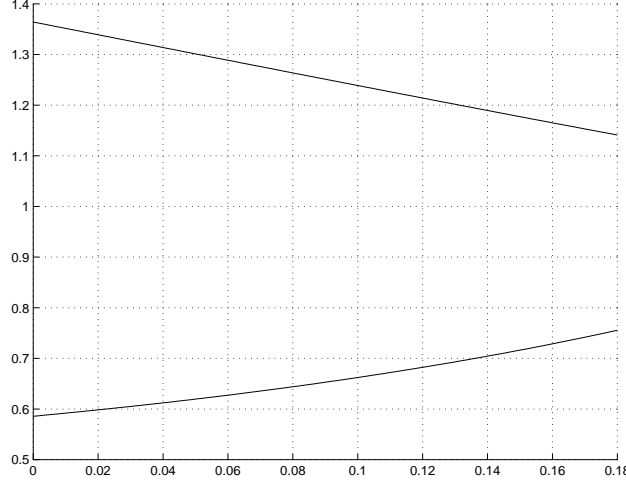


FIG. 3.5 – Bornes de stabilité pour  $\nu$  en fonction de  $\|\delta\mathbf{p}_0\|$

Ces résultats signifient que, si  $\|\delta\mathbf{p}_0\| = 0.0875$  (ce qui correspond à un cône avec une ouverture de 5 dg), alors  $\nu$  doit être compris entre 0.60 et 1.26. Pour obtenir une interprétation plus simple de cette condition, on va supposer maintenant que  $\nu \approx d^*/\hat{d}^*$  (ce qui signifie que la normale au plan de référence est  $\mathbf{n}^* \approx [0 \ 0 \ 1]^T$ ). Si la distance réelle  $d^*$  est à nouveau 50 cm, le système converge asymptotiquement à partir de n'importe quelle position initiale dans l'espace de travail si  $\hat{d}^*$  est choisi entre 30 et 63 cm.

Si cette hypothèse sur  $\mathbf{n}^*$  n'est pas vérifiée, le domaine de stabilité est plus petit. En effet,  $\nu = \|\delta\mathbf{A}^{-T}\mathbf{n}^*\|d^*/\hat{d}^*$  et on obtient alors :

$$\sqrt{\rho^{-1}(\delta\mathbf{A}^T\delta\mathbf{A})}\|\mathbf{n}^*\| \leq d^*/\hat{d}^* \leq \sqrt{\rho(\delta\mathbf{A}^{-T}\delta\mathbf{A}^{-1})}\|\mathbf{n}^*\| \quad (3.66)$$

où la définition du rayon spectral  $\rho(\mathbf{M})$  est rappelée en Annexe B. Comme  $\|\mathbf{n}^*\| = 1$ , la stabilité du système est assurée si (voir l'équation (3.63)) :

$$\nu_{min}\sqrt{\rho(\delta\mathbf{A}^T\delta\mathbf{A})} < d^*/\hat{d}^* < \sqrt{\rho^{-1}(\delta\mathbf{A}^{-T}\delta\mathbf{A}^{-1})}\nu_{max} \quad (3.67)$$

Dans le cas considéré, on a  $\sqrt{\rho^{-1}(\delta\mathbf{A}^{-T}\delta\mathbf{A}^{-1})} = 0.99$  et  $\sqrt{\rho(\delta\mathbf{A}^T\delta\mathbf{A})} = 1.57$ . On en déduit donc  $0.94 < d^*/\hat{d}^* < 1.25$  et  $\hat{d}^*$  doit être choisi entre 47 et 62 cm.

### Exemple 3 : distance $d^*$ et $\delta \mathbf{p}_0$ grossièrement connus

On fixe maintenant  $\|\delta \mathbf{p}_0\| = 0.1763$  (ce qui correspond à un cône de 10 dg d'ouverture) et à nouveau  $\delta \phi = 5$  dg. Les bornes supérieure et inférieure de l'inégalité (3.63) sont présentées en Figure 3.6 en fonction du rapport  $\alpha_u/\hat{\alpha}_u$  sur l'axe  $\vec{x}$  et en fonction du rapport  $\alpha_v/\hat{\alpha}_v$  sur l'axe  $\vec{y}$ .

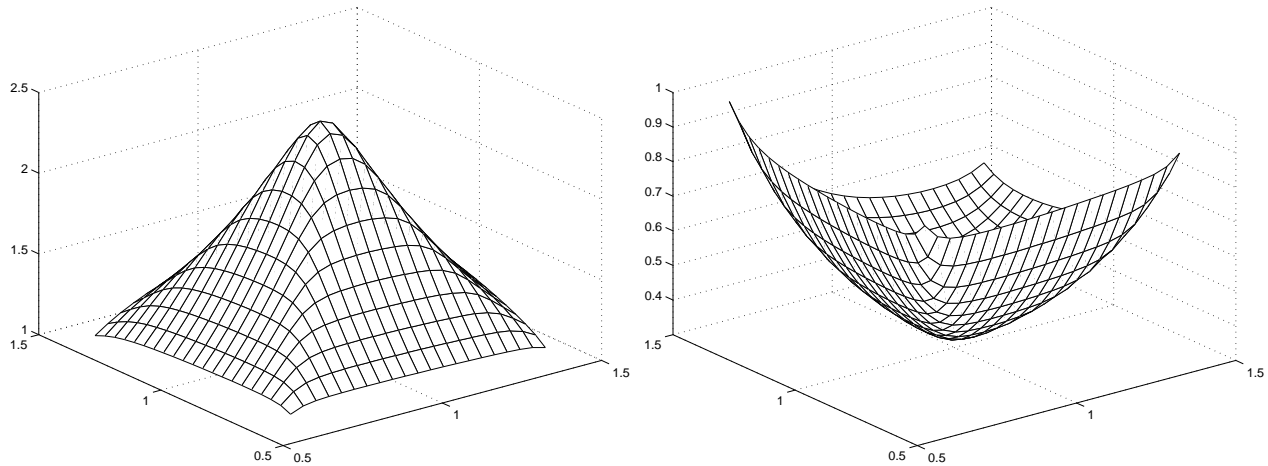


FIG. 3.6 – Bornes de stabilité pour  $\nu$  en fonction de  $\frac{\alpha_u}{\hat{\alpha}_u}$  et  $\frac{\alpha_v}{\hat{\alpha}_v}$

Pour une caméra standard avec  $\alpha_u/\alpha_v = 3/4$ , si  $\hat{\alpha}_u/\hat{\alpha}_v = 4/3$  alors on obtient  $\nu \approx d^*/\hat{d}^*$  compris entre 0.62 et 1.32. Si la distance réelle  $d^*$  est à nouveau 50 cm, le système converge asymptotiquement pour n'importe quelle position initiale dans l'espace de travail si  $\hat{d}^*$  est choisi entre 31 et 66 cm.

Il faut souligner que les conditions (3.49-3.51) sont plus restrictives que les conditions (3.44-3.47). Quand elles sont vérifiées, le système converge asymptotiquement pour n'importe quelle position initiale de la caméra. Si l'espace de travail est plus petit (c'est-à-dire si la position initiale de la caméra est toujours dans une région bien définie), l'analyse de la stabilité peut être effectuée à partir des conditions (3.44-3.47) en tenant compte des éventuelles restrictions sur l'espace de travail (qui vont jouer sur la valeur de  $\mu$ ). On peut alors obtenir une plage de robustesse plus grande. On remarque enfin que la condition (3.47) est seulement suffisante. La convergence peut donc être obtenue même pour des erreurs plus importantes.

Dans le prochain paragraphe, on démontre que l'asservissement visuel  $2D\frac{1}{2}$  est également robuste en présence d'erreurs dans la transformation caméra/effecteur. Toutefois, à cause de la complexité du problème, seule la stabilité asymptotique locale a été démontrée au jour d'aujourd'hui. Cependant, on verra ensuite qu'en effectuant une commande robuste, il est possible de prouver la stabilité asymptotique globale de l'asservissement visuel  $2D\frac{1}{2}$  en présence de telles erreurs.

### 3.1.2.4 Analyse de la robustesse de la stabilité aux erreurs sur les paramètres caméra et caméra/effecteur

Si l'hypothèse d'une parfaite calibration du repère de passage caméra/effecteur n'est pas réaliste ( $\mathbf{W} \neq \widehat{\mathbf{W}}$ ), la matrice d'erreur intervenant dans  $\mathbf{Q}(\mathbf{e})$  s'écrit (voir équation (3.71)) :

$$\delta\mathbf{W} = \mathbf{W} \widehat{\mathbf{W}}^{-1} = \begin{bmatrix} {}^c\mathbf{R}_e & {}^c\widehat{\mathbf{R}}_e^T & 0 \\ 0 & {}^c\mathbf{R}_e & \widehat{\mathbf{R}}_e^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & [{}^c\mathbf{R}_e & {}^c\widehat{\mathbf{R}}_e^T c\mathbf{t}_e - c\widehat{\mathbf{t}}_e]_{\times} \\ 0 & \mathbf{I} \end{bmatrix} \quad (3.68)$$

En notant respectivement  $\delta\mathbf{R} = {}^c\mathbf{R}_e & {}^c\widehat{\mathbf{R}}_e^T$  et  $\delta\mathbf{t} = (\delta\mathbf{R}^T c\mathbf{t}_e - c\widehat{\mathbf{t}}_e)$  l'erreur commise sur la rotation et sur la translation, on a enfin :

$$\delta\mathbf{W} = \begin{bmatrix} \delta\mathbf{R} & 0 \\ 0 & \delta\mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{I} & [\delta\mathbf{t}]_{\times} \\ 0 & \mathbf{I} \end{bmatrix} \quad (3.69)$$

La matrice  $\mathbf{Q}(\mathbf{e})$  est donc donnée par :

$$\mathbf{Q}(\mathbf{e}) = \begin{bmatrix} \mathbf{L}_v \delta\mathbf{R} \widehat{\mathbf{L}}_v^{-1} \mathbf{E}_v & \mathbf{L}_v \delta\mathbf{R} (\delta\mathbf{R}^T \mathbf{L}_v^{-1} \mathbf{L}_\omega \delta\mathbf{R} - [\delta\mathbf{t}]_{\times} - \widehat{\mathbf{L}}_v^{-1} \widehat{\mathbf{L}}_\omega) \mathbf{E}_\omega \\ \mathbf{0} & \mathbf{L}_\omega \delta\mathbf{R} \widehat{\mathbf{L}}_\omega^{-1} \mathbf{E}_\omega \end{bmatrix} \quad (3.70)$$

et comme  $\mathbf{L}_v^{-1} \mathbf{L}_\omega = [\mathbf{x}]_{\times}$ ,  $\delta\mathbf{R}^T [\mathbf{x}]_{\times} \delta\mathbf{R} = [\delta\mathbf{R}^T \mathbf{x}]_{\times}$ , et  $\widehat{\mathbf{L}}_\omega^{-1} = \mathbf{I}$ , on obtient :

$$\mathbf{Q}(\mathbf{e}) = \begin{bmatrix} \mathbf{L}_v \delta\mathbf{R} \widehat{\mathbf{L}}_v^{-1} \mathbf{E}_v & \mathbf{L}_v \delta\mathbf{R} [\delta\mathbf{R}^T \mathbf{x} - \delta\mathbf{t} - \widehat{\mathbf{x}}]_{\times} \mathbf{E}_\omega \\ \mathbf{0} & \mathbf{L}_\omega \delta\mathbf{R} \mathbf{E}_\omega \end{bmatrix} \quad (3.71)$$

Il semble très difficile, compte tenu de la complexité des équations, de trouver des conditions simples pour la stabilité globale du système. On étudie alors seulement la stabilité locale qui peut nous renseigner sur le comportement global du système.

### Stabilité asymptotique locale

Par rapport au cas précédent décrit en 3.1.2.3, l'analyse est complexe en raison du fait que l'erreur en rotation  $\delta\mathbf{R}$  couple les composantes des matrices  $\mathbf{L}$  et  $\mathbf{E}$ , ce qui rend le calcul des valeurs propres de  $\mathbf{Q}(\mathbf{e})$  non trivial. Afin de démontrer la stabilité asymptotique locale, on s'intéresse alors à la positivité des sous matrices sur la diagonale. En effet, si ces matrices sont positives alors elles ont des valeurs propres positives. Comme les valeurs propres de la matrice  $\mathbf{Q}$  triangulaire à bloc sont les valeurs propres des sous-matrices sur la diagonale, on peut obtenir des conditions suffisantes pour la stabilité locale du système en boucle fermée.

### Théorème 3

*Le point d'équilibre  $\mathbf{e}^\circ$  du système différentiel (3.33), caractérisé par la matrice (3.71), est localement asymptotiquement stable si et seulement si  $\frac{d^*}{d^*} > 0$  et s'il existe trois angles  $\theta_1$ ,  $\theta_2$  et  $\theta_3$  tels que :*

$$|\theta_1| + |\theta_2| + |\theta_3| < \frac{\pi}{2} \quad (3.72)$$

et tels que :

- si  $\delta\theta$  est l'angle de rotation de la matrice de rotation  $\delta\mathbf{R}$  :

$$\cos(\delta\theta) = \frac{\text{trace}(\delta\mathbf{R}) - 1}{2} \geq \cos(\theta_1) \quad (3.73)$$

- si la matrice  $\delta\mathbf{A}$  est positive (voir les conditions (3.44) et (3.46)) :

$$\frac{\rho^{-1}((\delta\mathbf{A} + \delta\mathbf{A}^T)^{-1})}{2\sqrt{\rho(\delta\mathbf{A}^T\mathbf{A})}} \geq \cos(\theta_2) \quad (3.74)$$

- si  $\gamma^* = \sqrt{x^{*2} + y^{*2}}$  :

$$\frac{1 + \frac{\gamma^{*2}}{2} - \frac{1}{2}\sqrt{\gamma^{*4} + 4\gamma^{*2}}}{1 + \frac{\gamma^{*2}}{2} + \frac{1}{2}\sqrt{\gamma^{*4} + 4\gamma^{*2}}} \geq \cos(\theta_3) \quad (3.75)$$

L'interprétation de ce théorème, dont la démonstration est donnée en Annexe B, est assez simple. L'angle  $\theta_1$  règle le domaine de robustesse aux erreurs de calibration caméra/effecteur. Plus cet angle est proche de zéro et plus il faut avoir une calibration précise. De même, l'angle  $\theta_2$  règle le domaine de robustesse aux erreurs de calibration de la caméra. L'angle  $\theta_3$  règle, lui, le domaine de stabilité dans l'image.

Plus la calibration est précise, plus on peut choisir  $\theta_1$  et  $\theta_2$  proches de zéro et plus  $\theta_3$  est proche de  $\pi/2$ . Dans le cas limite  $\theta_1 = \theta_2 = 0$  (où la calibration est parfaite car  $\delta\theta = 0$  et  $\delta\mathbf{A} = \mathbf{I}$ ) et  $\theta_3 = \pi/2$ , la condition (3.75) est toujours vérifiée.

Un autre cas limite, très réaliste en pratique, se produit si le point de référence est choisi au point principal. Alors  $\gamma^* = \sqrt{x^{*2} + y^{*2}} = 0$  et la condition (3.75) est vérifiée pour  $\theta_3 = 0$ . On dispose alors d'un angle de  $\pi/2$  degrés pour les erreurs cumulées sur  $\delta\mathbf{R}$  et  $\delta\mathbf{A}$ .

Comme le montre la Figure (3.7) où est illustrée la borne pour  $|\theta_3|$  en fonction de  $\gamma^*$ , il est souhaitable de choisir le point de référence le plus possible au centre de l'image. Pour une caméra avec un angle de vision de 20 degrés, si le point de référence est choisi dans une zone centrée de l'image correspondant à un cône de 10 degrés ( $\gamma^* \leq 0.17$ ) alors  $\theta_3 = \pi/4$ . On dispose alors d'un angle de  $\pi/4$  pour les erreurs cumulées sur  $\delta\mathbf{R}$  et  $\delta\mathbf{A}$ .

Nous allons voir à présent qu'une commande proportionnelle robuste, basée sur le réglage automatique de gains adaptatifs, nous permet d'élargir énormément le domaine de robustesse, aussi bien en présence d'erreurs sur les paramètres de la caméra qu'en présence d'erreurs sur les paramètres caméra/effecteur.

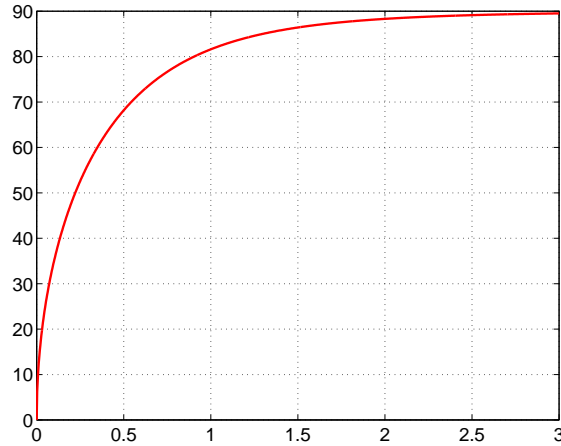


FIG. 3.7 – Borne de stabilité pour  $|\theta_3|$  (degrés) en fonction de  $\gamma^*$

### 3.1.3 Commande robuste en présence d’erreurs sur les paramètres caméra

On suppose ici que la matrice de transformation caméra/effecteur est parfaitement connue ( $\widehat{\mathbf{W}} = \mathbf{W}$ ). La structure simple du système à contrôler (le jacobien de la tâche est triangulaire supérieure par bloc) nous permet donc d’envisager une loi de commande de type adaptatif sur les gains de la matrice  $\mathbf{D}$ . Il faut distinguer deux types de commandes adaptatives [Samson 91]:

- adaptation des gains : ce type de commande a pour objectif d’améliorer la robustesse de la stabilité aux erreurs de calibration. L’adaptation est basée sur l’analyse du système en boucle fermée et n’apporte pas de connaissances supplémentaires sur les paramètres géométriques du système qui, eux, restent constants.
- adaptation des paramètres géométriques du système : ce type de commande a pour objectif d’améliorer non seulement la stabilité, mais aussi le comportement dynamique (et donc les performances) du système, par la mise à jour à chaque itération des paramètres géométriques du système.

On s’intéresse ici au premier type de commande adaptative, qu’on appellera commande robuste pour la différencier du deuxième type de commande, car on désire montrer que le domaine de robustesse aux erreurs de calibration (jusqu’ici défini par les conditions du Théorème 2) peut être étendu énormément.

Une fois la robustesse du système assurée, son comportement dynamique peut être amélioré soit par une adaptation des paramètres du système, soit par une augmentation des gains proportionnels. La première solution est préférable car une augmentation des gains réduit la marge de stabilité du système.

On rappelle que, dans notre cas, la fonction de tâche s'écrit :

$$\mathbf{e} = \begin{bmatrix} \mathbf{p}_e - \mathbf{p}_e^* \\ \mathbf{u}\theta \end{bmatrix} \quad (3.76)$$

où l'on utilise à présent les coordonnées étendues dans l'image  $\mathbf{p}_e^*$  exprimées en pixels. La matrice d'interaction est alors (voir équation (3.16)) :

$$\mathbf{L} = \begin{bmatrix} \mathbf{B}\mathbf{L}_v & \mathbf{B}\mathbf{L}_{(v,\omega)} \\ \mathbf{0} & \mathbf{L}_\omega \end{bmatrix} \quad (3.77)$$

Comme les matrices  $\mathbf{B}$  et  $\mathbf{L}_v$  sont triangulaires supérieures, leur produit est triangulaire supérieure, mais ce n'est pas le cas pour  $\mathbf{L}_\omega$ .

### 3.1.3.1 Loi de commande

On "souhaite" de nouveau une décroissance exponentielle de la fonction de tâche :

$$\dot{\mathbf{e}} = -\lambda \mathbf{D}(\mathbf{e}) \mathbf{e} \quad (3.78)$$

où  $\mathbf{D}$  est une matrice diagonale positive dont les éléments varient en fonction de l'état du système :

$$\mathbf{D}(\mathbf{e}) = \begin{bmatrix} \mathbf{D}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_\omega \end{bmatrix} \quad (3.79)$$

Précédemment, la matrice de contrôle à été choisie constante  $\mathbf{D} = \mathbf{I}_6$ . On va voir maintenant qu'un choix judicieux de cette matrice nous permet de démontrer des conditions nécessaires et suffisantes pour la stabilité asymptotique du système en boucle fermée.

Le "choix naturel" pour contrôler le système est, une fois de plus, une commande qui linéarise et découple l'état. La vitesse de l'effecteur du robot est donc cette fois (car  $\widehat{\mathbf{W}} = \mathbf{W}$ ) :

$$\dot{\mathbf{r}} = -\lambda \mathbf{W}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{D}(\widehat{\mathbf{e}}) \widehat{\mathbf{e}} \quad (3.80)$$

La matrice caractérisant le système différentiel non-linéaire (3.33) est :

$$\mathbf{Q}(\mathbf{e}) = \begin{bmatrix} \mathbf{B} \mathbf{L}_v \widehat{\mathbf{L}}_v^{-1} \widehat{\mathbf{B}}^{-1} \mathbf{D}_v & \mathbf{B} \mathbf{L}_v [\mathbf{x} - \widehat{\mathbf{x}}]_\times \widehat{\mathbf{L}}_\omega^{-1} \mathbf{D}_\omega \mathbf{E}_\omega \\ \mathbf{0} & \mathbf{L}_\omega \widehat{\mathbf{L}}_\omega^{-1} \mathbf{D}_\omega \mathbf{E}_\omega \end{bmatrix} \quad (3.81)$$

Il faut remarquer que, contrairement à la forme donnée en (3.29), on a ici  $\mathbf{E}_v = \mathbf{I}_3$  car la fonction de tâche est construite à partir des coordonnées exprimées en pixels. De plus, nous pouvons encore poser  $\widehat{\mathbf{L}}_\omega^{-1} = \mathbf{I}$  malgré la présence de la matrice  $\mathbf{D}_\omega$ , car on verra par la suite qu'un choix particulier de cette matrice (en l'occurrence  $\mathbf{D}_\omega = \lambda_\omega \mathbf{I}$ ) nous permet d'avoir encore  $\widehat{\mathbf{L}}_\omega^{-1} \mathbf{D}_\omega \mathbf{e}_\omega = \mathbf{D}_\omega \mathbf{e}_\omega$ .

Il faut aussi remarquer que les résultats qu'on va établir à présent sont également valables pour un vecteur d'erreur sur la translation  $\delta \mathbf{t}$  du repère caméra/effecteur quelconque si la

matrice d'erreur sur la rotation du repère caméra/effecteur est  $\delta\mathbf{R} = \mathbf{I}$ . Dans ce cas, les blocs sur la diagonale de  $\mathbf{Q}(\mathbf{e})$  sont en effet les mêmes.

La matrice des gains  $\mathbf{D}$  peut être construite afin que le système en boucle fermée vérifie le théorème suivant :

#### Théorème 4

Considérons le système non-linéaire d'équations différentielles ( $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ ) suivant :

$$\dot{\mathbf{x}} = -\lambda\mathbf{M}(\mathbf{x}) \mathbf{x} \quad \mathbf{x} \in S \quad (3.82)$$

où  $S = \bigcup_{i=1}^4 S_i$  est l'union des boules  $S_i$ , centrées à l'origine  $\mathbf{x}_i = 0$ , de frontière  $\Gamma_i$ , et  $\mathbf{M}(\mathbf{x})$  est une matrice triangulaire supérieure :

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \mathbf{M}_{11}(\mathbf{x}) & \mathbf{M}_{12}(\mathbf{x}) & \dots & \mathbf{M}_{1n}(\mathbf{e}) \\ \mathbf{0} & \mathbf{M}_{22}(\mathbf{x}) & \dots & \mathbf{M}_{2n}(\mathbf{x}) \\ \mathbf{0} & \mathbf{0} & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{nn}(\mathbf{x}) \end{bmatrix} \quad (3.83)$$

Soit la matrice  $\mathbf{M}(\mathbf{x})$  continue avec ses dérivées premières dans un domaine connexe borné et fermé  $\bar{S}$  (où  $\bar{S} = \bigcup_{i=1}^4 S_i \cup \Gamma_i$  est la fermeture de  $S$ ). Étant donné un état initial  $\mathbf{x}_0 = \mathbf{x}(t = 0) \in \bar{S}$ , si :

$$\mathbf{M}_{ii}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \in \bar{S} \quad (3.84)$$

$$\lim_{\mathbf{x}_i \rightarrow \Gamma_i} \|\mathbf{M}_{kj}(\mathbf{x})\| = 0 \quad \forall j > i, \forall k \geq i \quad (3.85)$$

alors il existe une solution unique  $\mathbf{x}(t)$  du système différentiel (3.83), telle que :

$$\mathbf{x}(t) \in S, \quad \forall t \quad (3.86)$$

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{0} \quad (3.87)$$

Le principe d'utilisation du théorème est le suivant (en supposant  $\mathbf{M} = \mathbf{Q}$ ) : quand une composante du vecteur d'état devient trop grande et risque de sortir de ses bornes, on diminue la valeur des éléments de la matrice  $\mathbf{D}$  (contenue dans  $\mathbf{Q}$ ) qui multiplient les variables qui sont la cause de l'erreur de poursuite. En imposant une décroissance vers zéro sur les valeur de  $\mathbf{D}$  (exponentielle en fonction de  $\mathbf{e}$ ) plus rapide que la croissance des autres éléments de la matrice  $\mathbf{Q}$  (polynomiale en fonction de  $\mathbf{e}$ ), les éléments qui ne sont pas sur la diagonale de  $\mathbf{Q}$  diminuent jusqu'au moment où les éléments sur la diagonale, qui sont eux positifs,

deviennent dominants, entraînant ainsi une décroissance de l'erreur. Dans notre cas, nous avons :

$$\begin{aligned}
 x_1 &= u \in \overline{S}_1 = [u_{min}, u_{max}] \\
 x_2 &= v \in \overline{S}_2 = [v_{min}, v_{max}] \\
 x_3 &= w \in \overline{S}_3 = [w_{min}, w_{max}] \\
 \mathbf{x}_4 &= \mathbf{u}\theta \in SO_3
 \end{aligned} \tag{3.88}$$

où seulement la première partie de l'état est contrainte car la rotation est libre d'évoluer dans  $SO_3$ . Les bornes  $[u_{min}, u_{max}]$  et  $[v_{min}, v_{max}]$  sont définies par la taille du capteur CCD. La borne  $[w_{min}, w_{max}]$  sur  $w = \ln(Z)$  peut être définie expérimentalement ou théoriquement, de telle manière, par exemple, que l'image ne soit jamais floue. La matrice de contrôle  $\mathbf{D}$  peut être choisie de la manière suivante :

$$\mathbf{D} = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & f_u(u) & 0 & 0 \\ 0 & 0 & f_u(u)f_v(v) & 0 \\ \hline 0 & 0 & 0 & \mathbf{I}_3 f_u(u)f_v(v)f_w(w) \end{array} \right] \tag{3.89}$$

où  $0 \leq f(x) \leq 1$  est une fonction en "cloche", symétrique par rapport à  $x_{med} = \frac{x_{min} + x_{max}}{2}$  :

$$f(x) = \begin{cases} e^{-\frac{(x-x_{med})^{2n}}{(x-x_{min})^m(x_{max}-x)^m}} & x_{min} < x < x_{max} \\ 0 & x = \{x_{min}, x_{max}\} \end{cases} \tag{3.90}$$

avec  $m$  et  $n$  deux paramètres utilisés pour régler la forme de la cloche. Les valeurs de  $m$  et  $n$  sont choisies de manière à avoir :

$$\begin{cases} f(x_{min} + \beta(x_{max} - x_{min})) \geq 1 - \alpha \\ f(x_{min} + \frac{\beta}{2}(x_{max} - x_{min})) \leq \alpha \end{cases} \text{ avec } \begin{cases} 0 < \alpha < 0.5 \\ 0 < \beta < 0.5 \end{cases} \tag{3.91}$$

La fonction étant symétrique par rapport à  $x_{med}$ , si ces dernières conditions sont vérifiées alors les conditions suivantes le sont aussi :

$$\begin{cases} f(x_{max} - \beta(x_{max} - x_{min})) \geq 1 - \alpha \\ f(x_{max} - \frac{\beta}{2}(x_{max} - x_{min})) \leq \alpha \end{cases} \text{ avec } \begin{cases} 0 < \alpha < 0.5 \\ 0 < \beta < 0.5 \end{cases} \tag{3.92}$$

Considérons par exemple une caméra ayant une image de taille  $600 \times 500$  pixels. On a donc :

$$\begin{cases} u_{min} = 0 \\ u_{med} = 300 \\ u_{max} = 600 \end{cases} \text{ et } \begin{cases} v_{min} = 0 \\ v_{med} = 250 \\ v_{max} = 500 \end{cases} \tag{3.93}$$

Si on veut, par exemple, que la fonction perde au plus 10% de sa valeur maximale ( $\alpha = 0.1$ ) à une distance du bord correspondant à 20% de la largeur, ou de la hauteur ( $\beta = 0.2$ ), on doit

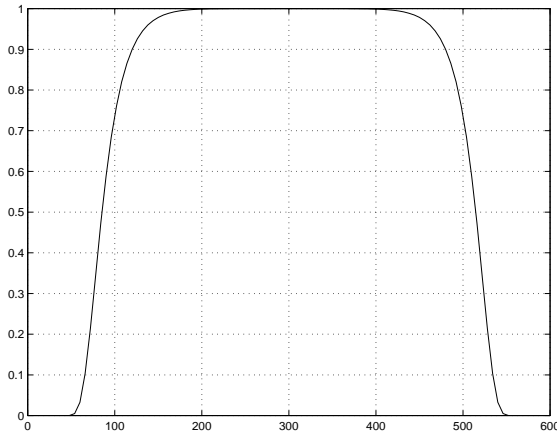


choisir  $n = m = 3$ . Par conséquent, la fonction est au moins à 90% de sa valeur maximale à 20% des bords de l'image (voir Figure 3.8) :

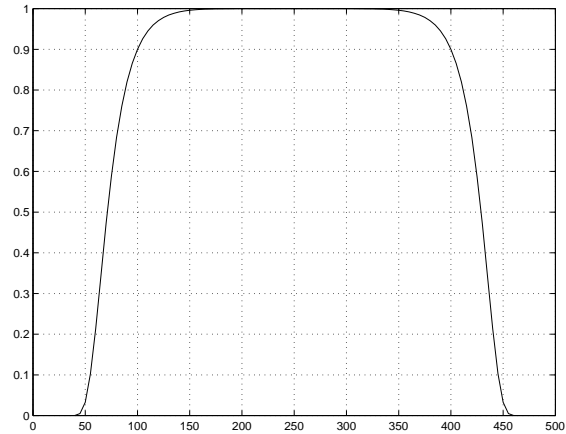
$$\begin{cases} f(u_{min} + 0.2(u_{max} - u_{min})) = f(120) \geq 0.9 \\ f(u_{min} + 0.1(u_{max} - u_{min})) = f(60) \leq 0.1 \\ f(u_{max} - 0.2(u_{max} - u_{min})) = f(480) \geq 0.9 \\ f(u_{max} - 0.1(u_{max} - u_{min})) = f(540) \leq 0.1 \end{cases}$$

$$\begin{cases} f(v_{min} + 0.2(v_{max} - v_{min})) = f(100) \geq 0.9 \\ f(v_{min} + 0.1(v_{max} - v_{min})) = f(50) \leq 0.1 \\ f(v_{max} - 0.2(v_{max} - v_{min})) = f(400) \geq 0.9 \\ f(v_{max} - 0.1(v_{max} - v_{min})) = f(450) \leq 0.1 \end{cases}$$

Dans cet exemple, on a construit  $f_u$  et  $f_v$  avec les mêmes paramètres  $n$  et  $m$ , mais rien n'empêche, comme on le verra dans le chapitre suivant (où on présente les résultats expérimentaux), de construire les deux fonctions avec des caractéristiques différentes.



(a) fonction à cloche  $f(u)$



(b) fonction à cloche  $f(v)$

FIG. 3.8 – Fonctions  $f(u)$  et  $f(v)$

La fonction  $f(u)f(v)$  est représentée sur la Figure 3.9 :

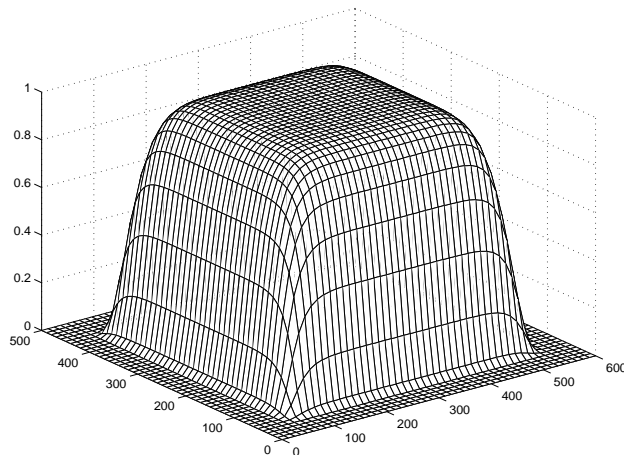


FIG. 3.9 – Fonction  $f(u)f(v)$

On peut construire de la même manière la fonction  $f(w)$  et la fonction  $f(u)f(v)f(w)$  (qui ne peut évidemment pas être visualisée car définie dans un espace à trois dimensions).

Les éléments de la matrice  $\mathbf{D}$  fonctionnent alors comme un filtre passe bande. Si l'erreur sur la distance relative  $w$  augmente à cause d'une mauvaise estimation de la rotation, alors les trois derniers éléments de la matrice  $\mathbf{D}$  diminuent et la commande en rotation n'est plus prise en compte. Un phénomène similaire se produit si seule l'erreur sur l'ordonnée  $v$  du point de référence augmente, alors les quatre derniers éléments de la matrice  $\mathbf{D}$  vont à zéro et la commande sur la distance relative et la rotation ne sont plus prises en compte. Enfin, le cas extrême se produit quand toutes les erreurs augmentent. Les cinq derniers éléments de la matrice  $\mathbf{D}$  vont alors à zéro et seule la commande de l'abscisse  $u$  est prise en compte.

### 3.1.3.2 Analyse de la robustesse de la stabilité

La loi de commande robuste qu'on vient de présenter n'a pas d'influence sur les conditions concernant la stabilité asymptotique locale du système, et les conditions nécessaires et suffisantes sont les mêmes que celles pour la stabilité locale du système commandé avec une loi classique (Théorème 1). En effet, les éléments de la matrice des gains  $\mathbf{D}$  sont positifs ou nuls, et les valeurs propres de la nouvelle matrice  $\mathbf{Q}$  sont égales à celles de l'ancienne multipliées par les éléments de  $\mathbf{D}$ . On étudie donc directement la stabilité globale du système.

### Stabilité globale

L'intérêt de la commande adaptative est, d'une part, l'extension du domaine de robustesse de la commande proportionnelle classique, et d'autre part, la possibilité de prendre en compte dans la commande la condition de visibilité de la tâche. En effet, si l'adaptation des gains est effectuée non plus pour le point de référence, mais pour le premier point de la cible qui est en train de sortir de l'image, on peut conserver en permanence la cible dans le champ de vue de la caméra.

Dans le cas d'une commande robuste à partir des coordonnées du point de référence, le domaine de robustesse de la stabilité globale du système est défini par le théorème suivant :

### **Théorème 5**

*Le point d'équilibre  $\mathbf{e}^\circ$  du système différentiel non-linéaire (3.33), caractérisé par la matrice (3.81), contrôlé en utilisant la matrice des gains définie par (3.89), est localement asymptotiquement stable si et seulement si :*

$$\frac{\hat{\alpha}_u}{\alpha_u} > 0, \quad \frac{\hat{\alpha}_v}{\alpha_v} > 0 \quad \text{et} \quad \frac{\hat{d}^*}{d^*} > 0 \quad (3.94)$$

*Ce système est globalement asymptotiquement stable seulement si les conditions (3.94) sont vérifiés et si :*

$$\delta \mathbf{A} > 0 \quad (3.95)$$

La démonstration de ce dernier théorème, obtenu facilement en montrant que le système rebouclé par la commande (3.89) vérifie les hypothèses du théorème 4, est donnée en Annexe B. Les conditions (3.94) sont nécessaires et il semble difficile qu'elles ne soient pas respectées en pratique. La condition (3.95) est seulement suffisante, mais, comme on l'a déjà vu lors de l'analyse des conditions du Théorème 2, elle est très facilement respectée en pratique.

On peut donc assurer qu'au moins le point de référence restera toujours dans l'image, ce qui augmente la probabilité qu'un certain nombre de points voisins le restent aussi, permettant ainsi de vérifier la condition de visibilité de la tâche. De plus, on montrera dans le chapitre suivant qu'il est généralement possible de conserver en permanence la cible dans le champ de vue de la caméra en effectuant l'adaptation des gains sur d'autres points que le point de référence.

## **3.1.4 Commande robuste en présence d'erreurs sur les paramètres caméra et caméra/effecteur**

### **3.1.4.1 Loi de commande**

Si la matrice de rotation  $\delta \mathbf{R}$  n'est pas égale à l'identité, le système est toujours triangulaire supérieur à bloc. Toutefois, comme on l'a déjà remarqué, la matrice  $\delta \mathbf{R}$  couple les variables d'état du premier sous-système. Une commande robuste du type de celle donnée par l'équation (3.89) n'a alors plus de sens car elle était fondée sur les découplages en cascade de chaque sous-système. Par contre, sur la base du corollaire du Théorème 4, on peut choisir dans ce cas une commande adaptative seulement sur les gains du sous-système en rotation :

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_v & 0 \\ 0 & \mathbf{D}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & 0 \\ 0 & \mathbf{I}_3 f(u) f(v) f(w) \end{bmatrix} \quad (3.96)$$

La matrice  $\mathbf{Q}$  de la boucle fermée est alors donnée par :

$$\mathbf{Q}(\mathbf{e}) = \begin{bmatrix} \mathbf{L}_v \delta \mathbf{R} \widehat{\mathbf{L}}_v^{-1} \mathbf{E}_v & \mathbf{L}_v \delta \mathbf{R} [\delta \mathbf{R}^T \mathbf{x} - \delta \mathbf{t} - \widehat{\mathbf{x}}]_{\times} \mathbf{D}_\omega \mathbf{E}_\omega \\ \mathbf{0} & \mathbf{L}_\omega \delta \mathbf{R} \widehat{\mathbf{L}}_\omega^{-1} \mathbf{D}_\omega \mathbf{E}_\omega \end{bmatrix} \quad (3.97)$$

### 3.1.4.2 Analyse de la robustesse de la stabilité

#### Stabilité globale

Il est évident que les conditions pour la stabilité du système sont similaires à celles données par le Théorème 3. La principale différence est qu'elles sont plus générales car elles sont valables pour la stabilité asymptotique globale :

#### Théorème 6

Le point d'équilibre  $\mathbf{e}^\circ$  du système différentiel (3.97) est globalement asymptotiquement stable si et seulement si  $\frac{d^*}{d^*} > 0$  et s'il existe trois angles  $\theta_1$ ,  $\theta_2$  et  $\theta_3$  tels que :

$$|\theta_1| + |\theta_2| + |\theta_3| < \frac{\pi}{2} \quad (3.98)$$

et tels que :

- Si  $\delta\theta$  est l'angle de rotation de la matrice de rotation  $\delta \mathbf{R}$  :

$$\cos(\delta\theta) = \frac{\text{trace}(\delta \mathbf{R}) - 1}{2} \geq \cos(\theta_1) \quad (3.99)$$

- Si la matrice  $\delta \mathbf{A}$  est positive (voir les conditions (3.44) et (3.46)) :

$$\frac{\rho^{-1}((\delta \mathbf{A} + \delta \mathbf{A}^T)^{-1})}{2\sqrt{\rho(\delta \mathbf{A}^T \mathbf{A})}} \geq \cos(\theta_2) \quad (3.100)$$

- Si  $\gamma = \sqrt{x^2 + y^2}$  :

$$\frac{1 + \frac{\gamma^2}{2} - \frac{1}{2}\sqrt{\gamma^4 + 4\gamma^2}}{1 + \frac{\gamma^2}{2} + \frac{1}{2}\sqrt{\gamma^4 + 4\gamma^2}} \geq \cos(\theta_3) \quad (3.101)$$

La démonstration de ce théorème est donnée en Annexe B. L'analyse des conditions du théorème est similaire à celle faite pour les conditions du Théorème 3. La principale différence est que la condition (3.101) doit être vérifiée non seulement dans un voisinage de la position finale, comme c'était le cas de la condition (3.75), mais dans toute l'image.

Nous verrons dans la section suivante un schéma d'asservissement avec des caractéristiques similaires à celles de l'asservissement visuel  $2D\frac{1}{2}$ , pour lequel toutes les démonstrations de stabilité sont simplifiées et qui fournit un domaine de robustesse des lois de commande beaucoup plus large. Toutefois, le gain de robustesse s'accompagne d'une augmentation de la sensibilité au bruit de mesure car, comme nous allons le voir, la fonction de tâche n'est alors conçue qu'à partir de paramètres reconstruits. D'autre part, paradoxalement, la condition de visibilité de la tâche est plus difficilement assurée car le contrôle ne s'effectue plus directement dans l'image.

## 3.2 Asservissement visuel $2D\frac{3}{4}$

On a vu dans la section précédente qu'il est possible de contrôler le mouvement du robot à la fois dans l'espace cartésien et à la fois dans l'image. Nous allons voir à présent qu'il est également possible de le contrôler dans un espace qui n'est plus l'image (car on n'utilise pas directement les coordonnées images des points), mais qui n'est pas non plus purement l'espace cartésien car la reconstruction euclidienne n'est obtenue qu'à un facteur d'échelle près. On rappelle en effet qu'à partir de la matrice essentielle ou de celle d'homographie, on peut estimer la rotation de la caméra et sa translation à un facteur d'échelle près.

Comme on l'a déjà indiqué dans le chapitre précédent, [Basri 98] a récemment proposé un schéma de commande très proche de l'asservissement visuel 3D. Sa méthode ne nécessite pas de modèle 3D de l'objet car la rotation et la translation (à un facteur d'échelle près) de la caméra sont estimées à partir de la matrice essentielle. Le facteur d'échelle en translation étant inconnu, celui-ci est estimé en utilisant la mesure de la translation effectuée par la caméra à la première itération de l'asservissement. Ensuite, la translation et la rotation de la caméra sont contrôlées exactement comme dans un asservissement visuel 3D classique.

Nous avons également réalisé ce type de schéma de contrôle. En effet, à partir de la matrice d'homographie, il est possible d'estimer le rapport entre la translation de la caméra et la distance  $d^*$ . Toutefois, exactement comme dans le cas de l'asservissement visuel 3D, si on contrôle la translation de la caméra entre la position courante et désirée, on n'assure aucun contrôle dans l'image. On a donc préféré deux autres solutions :

- si le contrôle direct de la translation de la caméra entre la position courante et désirée est nécessaire dans la tâche qu'on doit réaliser (parce que, par exemple, on souhaite aller en ligne droite d'un point à un autre), on contrôle deux degrés de liberté en rotation en utilisant un point de référence afin de le garder dans l'image. Le troisième degré de rotation, à savoir la rotation autour de l'axe optique, est estimé directement. Cette méthode très simple d'asservissement est décrite en Annexe C.
- si le contrôle de la trajectoire de la caméra n'est pas indispensable, nous proposons une nouvelle méthode qui contrôle un vecteur proportionnel aux coordonnées cartésiennes (estimées à un facteur d'échelle près) d'un point de référence de la cible. Ce schéma de contrôle, intermédiaire entre l'asservissement visuel  $2D\frac{1}{2}$  et l'asservissement visuel 3D, est appelée asservissement visuel  $2D\frac{3}{4}$ .

### 3.2.1 Modélisation

Comme on l'a déjà vu auparavant, à partir de la reconstruction partielle étudiée dans le chapitre 2, il est possible de déterminer les coordonnées dans l'espace cartésien d'un point  $\mathbf{x} = [X \ Y \ Z]$  à un facteur d'échelle près. Une méthode d'asservissement visuel similaire à celle proposée par [Martinet 96] peut alors être conçue sans aucune connaissance a priori du modèle de la cible observée. En effet, les seules mesures disponibles sont les coordonnées

d'un point dans l'image et le rapport  $\frac{Z}{Z^*}$  entre la profondeur courante et désirée de ce point.

On définit donc les *coordonnées cartésiennes réduites* :

$$\mathbf{x}_r = \frac{1}{Z^*} \mathbf{x} = \frac{Z}{Z^*} \mathbf{m} \quad (3.102)$$

À partir de cette définition et de l'équation (3.5), la dérivée des coordonnées cartésiennes réduites est liée au torseur cinématique de la caméra par la relation :

$$\dot{\mathbf{x}}_r = \left[ -\frac{1}{Z^*} \mathbf{I}_3 \quad [\mathbf{x}_r]_{\times} \right] \mathbf{v} \quad (3.103)$$

où  $[\mathbf{x}_r]_{\times}$  est la matrice de pré-produit vectoriel associée au vecteur  $\mathbf{x}_r$ . Le contrôle de l'orientation s'effectue de la même manière que pour le  $2D\frac{1}{2}$ . Les équations (3.1) et (3.103) peuvent donc se regrouper de la manière suivante :

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x}_r \\ \mathbf{u}\theta \end{bmatrix} = \begin{bmatrix} \mathbf{L}_v & \mathbf{L}_{(v,\omega)} \\ \mathbf{0} & \mathbf{L}_\omega \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{L}(\mathbf{x}_r, Z^*) \mathbf{v} \quad (3.104)$$

où  $\mathbf{L}_v = \frac{-1}{Z^*} \mathbf{I}_3$  et  $\mathbf{L}_{(v,\omega)} = [\mathbf{x}_r]_{\times}$ . La matrice d'interaction  $\mathbf{L}$  est à nouveau triangulaire par bloc et toujours de rang plein :

$$\mathbf{L}(\mathbf{x}_r, Z^*) = \begin{bmatrix} -\frac{1}{Z^*} & 0 & 0 & 0 & -\frac{Z}{Z^*} & \frac{Z}{Z^*} y \\ 0 & -\frac{1}{Z^*} & 0 & \frac{Z}{Z^*} & 0 & -\frac{Z}{Z^*} x \\ 0 & 0 & -\frac{1}{Z^*} & -\frac{Z}{Z^*} y & \frac{Z}{Z^*} x & 0 \\ 0 & 0 & 0 & 1 - \beta(1 + u_x^2) & \beta u_y u_x - \alpha u_z & \beta u_z u_x + \alpha u_y \\ 0 & 0 & 0 & \beta u_y u_x + \alpha u_z & 1 - \beta(1 + u_y^2) & \beta u_z u_y - \alpha u_x \\ 0 & 0 & 0 & \beta u_z u_x - \alpha u_y & \beta u_z u_y + \alpha u_x & 1 - \beta(1 + u_z^2) \end{bmatrix} \quad (3.105)$$

avec cette fois  $Z^* = \rho^* d^*$ , et toujours  $\alpha = -\frac{\theta}{2}$  et  $\beta = 1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})}$ . La forme de cette matrice est plus simple que celle obtenue pour la matrice (3.13) car le bloc sur la première diagonale est lui-même diagonal.

On va à présent comparer les deux schémas d'asservissement avec une loi proportionnelle dans le cas idéal où il n'y a pas d'erreurs de calibration du système.

### 3.2.2 Comparaison des asservissements $2D\frac{1}{2}$ et $2D\frac{3}{4}$

On montre maintenant que le fait d'imposer une convergence exponentielle découplée sur  $\mathbf{x}_r - \mathbf{x}_r^*$  ( $2D\frac{3}{4}$ ) ou sur  $\mathbf{m}_e - \mathbf{m}_e^*$  ( $2D\frac{1}{2}$ ) conduit à imposer une trajectoire rectiligne du point de référence dans l'image. En effet, dans le cas de l'asservissement visuel  $2D\frac{1}{2}$ , une convergence

exponentielle de  $\dot{\mathbf{m}}_e = -\lambda(\mathbf{m}_e - \mathbf{m}_e^*)$ , à partir des conditions initiales  $x(0) = x_0$ ,  $y(0) = y_0$ , et  $Z(0) = Z_0$  se traduit par le système différentiel suivant:

$$\begin{cases} \frac{dx(t)}{dt} &= -\lambda(x - x^*) \\ \frac{dy(t)}{dt} &= -\lambda(y - y^*) \\ \frac{d(\ln(\frac{Z}{Z^*}))}{dt} &= -\lambda \ln(\frac{Z}{Z^*}) \end{cases} \quad (3.106)$$

La solution de ce système est dans l'image :

$$\begin{cases} x(t) &= x^* - (x^* - x_0) \exp^{-\lambda t} \\ y(t) &= y^* - (y^* - y_0) \exp^{-\lambda t} \end{cases} \quad (3.107)$$

et, dans l'espace cartésien, la trajectoire suivante :

$$\begin{cases} X(t) = Z(t)x(t) &= (X^* - (X^* - X_0 \frac{Z^*}{Z_0}) \exp^{-\lambda t}) (\frac{Z_0}{Z^*})^{\exp^{-\lambda t}} \\ Y(t) = Z(t)y(t) &= (Y^* - (Y^* - Y_0 \frac{Z^*}{Z_0}) \exp^{-\lambda t}) (\frac{Z_0}{Z^*})^{\exp^{-\lambda t}} \\ Z(t) = Z^* (\frac{Z_0}{Z^*})^{\exp^{-\lambda t}} \end{cases} \quad (3.108)$$

On voit bien que la trajectoire dans l'image est une ligne droite alors que ce n'est pas le cas de la trajectoire dans l'espace cartésien.

Dans le cas de l'asservissement visuel  $2D\frac{3}{4}$ , La dérivée de  $\mathbf{m}_e$  est liée à la dérivée de  $\dot{\mathbf{x}}_r$  par la relation (voir équation (3.7)) :

$$\dot{\mathbf{m}}_e = \frac{1}{Z} \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} \dot{\mathbf{x}} = \frac{Z^*}{Z} \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} \dot{\mathbf{x}}_r \quad (3.109)$$

En imposant une convergence exponentielle de  $\dot{\mathbf{x}}_r = -\lambda(\mathbf{x}_r - \mathbf{x}_r^*)$ , on obtient, comme  $\mathbf{x}_r = \frac{Z}{Z^*} \mathbf{m}$  et  $\mathbf{x}_r^* = \mathbf{m}^*$  :

$$\dot{\mathbf{m}}_e = -\lambda \frac{Z^*}{Z} \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} (\mathbf{x}_r - \mathbf{x}_r^*) = -\lambda \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} (\mathbf{m} - \frac{Z^*}{Z} \mathbf{m}^*) \quad (3.110)$$

soit :

$$\dot{\mathbf{m}}_e = -\lambda \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \lambda \frac{Z^*}{Z} \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} \quad (3.111)$$

On obtient enfin le système suivant :

$$\begin{cases} \frac{dx(t)}{dt} &= -\lambda \frac{Z^*}{Z} (x - x^*) \\ \frac{dy(t)}{dt} &= -\lambda \frac{Z^*}{Z} (y - y^*) \\ \frac{d(\ln(\frac{Z}{Z^*}))}{dt} &= -\lambda (1 - \frac{Z^*}{Z}) \end{cases} \quad (3.112)$$

On remarque donc que, en imposant une convergence exponentielle sur  $\mathbf{x}_r$ , on obtient une trajectoire dans l'image qui est toujours rectiligne. Cependant, la convergence n'est plus exponentielle car la loi horaire est différente :

$$\begin{cases} x(t) = x^* - (x^* - x_0) \frac{\exp^{-\lambda t}}{\frac{Z^*}{Z_0} + (1 - \frac{Z^*}{Z_0}) \exp^{-\lambda t}} \\ y(t) = y^* - (y^* - y_0) \frac{\exp^{-\lambda t}}{\frac{Z^*}{Z_0} + (1 - \frac{Z^*}{Z_0}) \exp^{-\lambda t}} \end{cases} \quad (3.113)$$

En effet, la vitesse de convergence varie dans le temps. Par contre, la trajectoire 3D du point dans le repère de la caméra est cette fois rectiligne :

$$\begin{cases} X(t) = X^* - (X^* - X_0) \exp^{-\lambda t} \\ Y(t) = Y^* - (Y^* - Y_0) \exp^{-\lambda t} \\ Z(t) = Z^* - (Z^* - Z_0) \exp^{-\lambda t} \end{cases} \quad (3.114)$$

On peut noter que la trajectoire réalisée sera véritablement une ligne droite seulement dans le repère courant de la caméra mais pas dans le repère absolu, puisque l'orientation du repère de la caméra varie au cours de l'asservissement.

En conclusion, dans le cas de l'asservissement visuel  $2D\frac{3}{4}$ , on impose une convergence en ligne droite d'un point dans le repère de la caméra, mais aussi de sa projection correspondante dans l'image. Le comportement dans l'image est le même que celui de l'asservissement visuel  $2D\frac{1}{2}$  (seule la loi horaire change). Par conséquent, on peut s'attendre théoriquement à des performances très proches des deux méthodes d'asservissement. Ceci est sûrement vrai quand  $\frac{Z}{Z^*} \approx 1$ . Dans ce cas, nous avons  $\ln(\frac{Z}{Z^*}) \approx \frac{Z}{Z^*} - 1$ , et les coordonnées image étendues et les coordonnées cartésiennes réduites coïncident.

Toutefois, il faut souligner que l'estimation de l'état est plus sensible au bruit avec la méthode  $2D\frac{3}{4}$  car la reconstruction partielle est plus sensible aux erreurs de mesure qu'une utilisation directe des coordonnées des points dans l'image. De plus, des erreurs sur la calibration et l'estimation de la profondeur se répercutent de manière plus évidente sur les trajectoires dans l'image avec la méthode  $2D\frac{3}{4}$ . Même si on peut assurer la convergence du point de référence comme dans le cas de la méthode  $2D\frac{1}{2}$ , la condition de visibilité de la tâche est donc en pratique plus difficile à satisfaire si on n'utilise pas une commande robuste.

### 3.2.3 Commande proportionnelle

Dans le cas de l'asservissement visuel  $2D\frac{3}{4}$ , la tâche de positionnement peut être décrite par la régulation de la fonction de tâche suivante :

$$\mathbf{e} = \begin{bmatrix} \mathbf{x}_r - \mathbf{x}_r^* \\ \mathbf{u}\theta \end{bmatrix} \quad (3.115)$$



où  $\mathbf{x}_r^*$  peut être estimé à partir de l'image désirée. On cherche à nouveau à commander le système avec une commande proportionnelle (c'est-à-dire qu'on désire une convergence exponentielle de  $\mathbf{x}_r$  vers  $\mathbf{x}_r^*$  et  $\mathbf{u}\theta$  vers 0).

### 3.2.3.1 Loi de commande

La loi de commande est donc :

$$\dot{\mathbf{r}} = -\lambda \widehat{\mathbf{W}}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{e}} \quad (3.116)$$

où  $\widehat{\mathbf{e}}$  est la fonction de tâche définie par l'équation (3.115) et  $\widehat{\mathbf{L}}$  est une approximation de la matrice  $\mathbf{L}$  définie par l'équation (3.105).

Comme  $\mathbf{L}$  dépend encore de la distance inconnue  $d^*$ , une valeur approximative  $\widehat{d}^*$  doit être choisie pendant la phase d'apprentissage et introduite dans  $\widehat{\mathbf{L}}^{-1}$ . Toutefois, comme dans le cas de l'asservissement visuel  $2D\frac{1}{2}$ , il n'est pas nécessaire que cette valeur soit déterminée avec une très grande précision car elle n'a pas une grande influence sur la stabilité du système. On verra même par la suite que la plage d'erreur qu'on peut commettre sur l'estimation de cette distance est dans certains cas supérieure à celle obtenue pour l'asservissement visuel  $2D\frac{1}{2}$ .

Enfin, si on choisit  $\mathbf{D} = \mathbf{I}$ , la loi de commande est donnée par :

$$\dot{\mathbf{r}} = -\lambda \begin{bmatrix} {}^c \widehat{\mathbf{R}}_e^T & -{}^c \widehat{\mathbf{R}}_e^T \begin{bmatrix} {}^c \widehat{\mathbf{t}}_e \\ \times \end{bmatrix} \\ 0 & {}^c \widehat{\mathbf{R}}_e^T \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{L}}_v^{-1} & -\widehat{\mathbf{L}}_v^{-1} \widehat{\mathbf{L}}_{(v,\omega)} \widehat{\mathbf{L}}_\omega^{-1} \\ \mathbf{0} & \mathbf{L}_\omega \widehat{\mathbf{L}}_\omega^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{x}}_r - \widehat{\mathbf{x}}_r^* \\ \widehat{\mathbf{u}}\widehat{\theta} \end{bmatrix} \quad (3.117)$$

On peut remarquer que la boucle de contrôle en rotation est toujours découplée de celle en translation (voir la Figure 3.10). Encore une fois, nous pouvons poser  $\widehat{\mathbf{L}}_\omega^{-1} = \mathbf{I}$  sans perte de généralité.

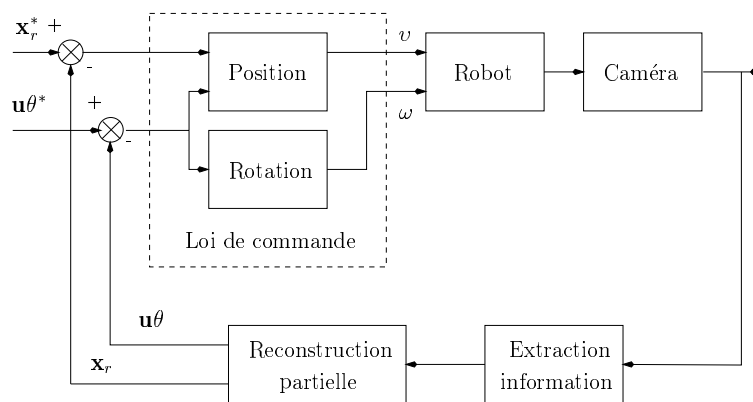


FIG. 3.10 – Schéma bloc de l'asservissement visuel  $2D\frac{3}{4}$

Comme pour l'asservissement visuel  $2D\frac{1}{2}$ , un tel système découplé permet d'obtenir la convergence dans tout l'espace de travail si on suppose le système parfaitement calibré et

l'objet toujours visible. De plus, on montrera dans la prochaine section qu'il est possible d'obtenir des conditions plus larges que celles obtenues pour l'asservissement visuel  $2D\frac{1}{2}$ , aussi bien pour la stabilité asymptotique globale en présence d'erreurs de calibration de la caméra, que pour la stabilité asymptotique locale en présence d'erreurs de calibration de la matrice de passage caméra/effecteur.

### 3.2.3.2 Boucle fermée

Afin de fermer la boucle, il est encore une fois nécessaire d'exprimer la fonction de tâche estimée en fonction de la fonction de tâche réelle et des erreurs de calibration. Les coordonnées cartésiennes réduites, courantes et désirées, s'écrivent :

$$\begin{cases} \mathbf{x}_r &= \frac{Z}{Z^*} \mathbf{m} = \frac{Z}{Z^*} \mathbf{A}^{-1} \mathbf{p} \\ \mathbf{x}_r^* &= \mathbf{m}^* = \mathbf{A}^{-1} \mathbf{p}^* \end{cases} \quad (3.118)$$

Les mêmes coordonnées estimées sont (on rappelle que  $\hat{\mathbf{p}} = \mathbf{p}$ , car on suppose le bruit de mesure dans l'image négligeable) :

$$\begin{cases} \hat{\mathbf{x}}_r &= \frac{\hat{Z}}{\hat{Z}^*} \hat{\mathbf{m}} = \frac{\hat{Z}}{\hat{Z}^*} \hat{\mathbf{A}}^{-1} \mathbf{p} \\ \hat{\mathbf{x}}_r^* &= \hat{\mathbf{m}}^* = \hat{\mathbf{A}}^{-1} \mathbf{p}^* \end{cases} \quad (3.119)$$

En injectant (3.118) dans (3.119) et en sachant que  $\frac{\hat{Z}}{\hat{Z}^*} = \frac{Z}{Z^*}$ , on obtient :

$$\begin{cases} \hat{\mathbf{x}}_r &= \frac{\hat{Z}}{\hat{Z}^*} \frac{Z^*}{Z} \delta \mathbf{A} \mathbf{x}_r = \delta \mathbf{A} \mathbf{x}_r \\ \hat{\mathbf{x}}_r^* &= \delta \mathbf{A} \mathbf{x}_r^* \end{cases} \quad (3.120)$$

On en déduit :

$$\hat{\mathbf{x}}_r - \hat{\mathbf{x}}_r^* = \delta \mathbf{A} (\mathbf{x}_r - \mathbf{x}_r^*) \quad (3.121)$$

La fonction de tâche estimée est alors :

$$\hat{\mathbf{e}} = \mathbf{E} \mathbf{e} = \begin{bmatrix} \mathbf{E}_v & 0 \\ 0 & \mathbf{E}_\omega \end{bmatrix} \mathbf{e} \quad (3.122)$$

avec  $\mathbf{E}_v = \delta \mathbf{A}$  et  $\mathbf{E}_\omega = \mu \delta \mathbf{A}$  (voir équation (2.50)). Le système en boucle fermée est donc :

$$\dot{\mathbf{e}} = \mathbf{f}(\mathbf{e}) = -\lambda \mathbf{Q}(\mathbf{e}) \mathbf{e} = \lambda \mathbf{L} \mathbf{W} \widehat{\mathbf{W}}^{-1} \widehat{\mathbf{L}}^{-1} \mathbf{D} \mathbf{E} \mathbf{e} \quad (3.123)$$

Exactement comme pour l'asservissement visuel  $2D\frac{1}{2}$ , il est facile de montrer l'existence et l'unicité du point d'équilibre :

**Proposition 2** *Le seul point d'équilibre pour  $\mathbf{f}$ , c'est-à-dire un point  $\mathbf{e}^\circ \in S$  tel que  $\mathbf{f}(\mathbf{e}^\circ) = 0$ , est  $\mathbf{e}^\circ = 0$ .*

Pour démontrer cette proposition, il n'y a strictement aucune différence avec la démonstration de la proposition 1.

### 3.2.3.3 Analyse de la robustesse de la stabilité aux erreurs sur les paramètres caméra

Si l'on suppose  $\mathbf{W} = \widehat{\mathbf{W}}$ , la matrice  $\mathbf{Q}(\mathbf{e})$  caractérisant la boucle fermée du système est donnée par :

$$\mathbf{Q}(\mathbf{e}) = \begin{bmatrix} \nu \delta \mathbf{A} & \mu [\mathbf{x}_r - \nu \widehat{\mathbf{x}}_r]_{\times} \delta \mathbf{A} \\ \mathbf{0} & \mu \mathbf{L}_{\omega} \delta \mathbf{A} \end{bmatrix} \quad (3.124)$$

On peut remarquer que, par rapport à l'asservissement visuel  $2D\frac{1}{2}$ , l'expression analytique de la boucle fermée est beaucoup plus simple. Par exemple, la sous-matrice supérieure sur la diagonale est constante. On verra par la suite que les conditions de stabilité sont également plus larges.

**Stabilité asymptotique locale** La démonstration de la stabilité asymptotique locale est identique à celle du Théorème 1 car le système est à nouveau triangulaire :

#### Théorème 7

*Le point d'équilibre  $\mathbf{e}^{\circ}$  du système différentiel (3.123) est localement asymptotiquement stable autour si et seulement si :*

$$\frac{\widehat{\alpha}_u}{\alpha_u} > 0, \quad \frac{\widehat{\alpha}_v}{\alpha_v} > 0 \quad \text{et} \quad \frac{\widehat{d}^*}{d^*} > 0 \quad (3.125)$$

En pratique, ces conditions sont facilement vérifiées. On remarque que ces conditions sont exactement les mêmes que pour le  $2D\frac{1}{2}$ .

#### Stabilité asymptotique globale

#### Théorème 8

*Le point d'équilibre  $\mathbf{e}^{\circ}$  du système différentiel (3.123) est globalement asymptotiquement stable, seulement si les conditions (3.125) sont vérifiées, et si :*

$$\sigma = \frac{\alpha_u}{\widehat{\alpha}_u} + \frac{\alpha_v}{\widehat{\alpha}_v} - \sqrt{\left(\frac{\alpha_u}{\widehat{\alpha}_u} - \frac{\alpha_v}{\widehat{\alpha}_v}\right)^2 + \left(\frac{\alpha_{uv}}{\widehat{\alpha}_u} - \frac{\widehat{\alpha}_{uv} \alpha_v}{\widehat{\alpha}_u \widehat{\alpha}_v}\right)^2} > 0 \quad (3.126)$$

$$\sigma_1 = \sigma + 1 - \sqrt{(\sigma - 1)^2 + \|\delta \mathbf{p}_0\|^2} > 0 \quad (3.127)$$

$$\mu \left(\frac{Z}{Z^*}\right)^2 (\nu \|\mathbf{I} - \delta \mathbf{A}\| + |1 - \nu|)^2 \|\delta \mathbf{A}\|^2 (1 + \overline{\gamma}^2) < \nu \sigma_1^2 \quad (3.128)$$

La démonstration de ce théorème est donnée en Annexe B. Les conditions suffisantes ont déjà été quasiment toutes analysées car, à part la condition (3.128), ce sont les mêmes que celles du Théorème 2.

Afin de montrer par un exemple le gain obtenu dans certains cas sur le domaine de robustesse, on suppose une calibration de la caméra parfaite ( $\delta \mathbf{A}_{11} = \mathbf{I}_2$ ,  $\delta \mathbf{p}_0 = \mathbf{0}$ ,  $\sigma = \sigma_1 = \sigma_2 = 2$ ,  $\sigma' = 1$  et  $\nu = \frac{\hat{d}^*}{d^*}$ ), la condition (3.128) peut alors s'écrire :

$$\left(1 - \frac{\hat{d}^*}{d^*}\right)^2 \left(\frac{Z}{Z^*} g(\bar{\gamma})\right)^2 < 4 \frac{\hat{d}^*}{d^*} \quad (3.129)$$

La solution de l'inégalité (3.129) est :

$$1 + 2 \frac{1 - \sqrt{\left(\frac{Z}{Z^*} g(\bar{\gamma})\right)^2 + 1}}{\left(\frac{Z}{Z^*} g(\bar{\gamma})\right)^2} < \frac{\hat{d}^*}{d^*} < 1 + 2 \frac{1 + \sqrt{\left(\frac{Z}{Z^*} g(\bar{\gamma})\right)^2 + 1}}{\left(\frac{Z}{Z^*} g(\bar{\gamma})\right)^2} \quad (3.130)$$

Les deux bornes sont tracées en Figure 3.11 en fonction de  $\bar{\gamma}$  et pour trois valeurs différentes de  $\frac{Z}{Z^*}$ . Les bornes obtenues pour le même exemple lors de l'analyse de la stabilité de l'asservissement visuel  $2D\frac{1}{2}$  sont tracées en traits discontinus.

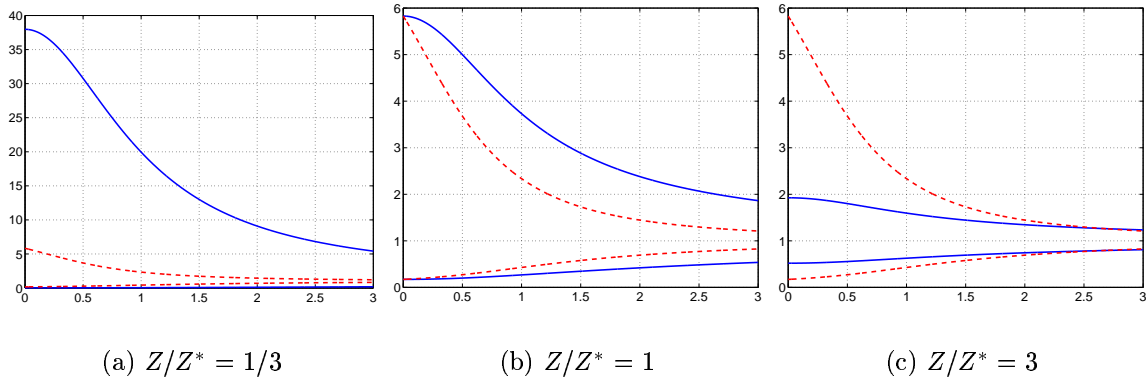


FIG. 3.11 – *Bornes de stabilité pour la distance estimée*

Si on considère encore une fois une caméra avec un angle de vision de  $20^\circ$  (donc  $\bar{\gamma} = 0.364$ ) et si la distance réelle  $d^*$  est 50 cm, le système converge asymptotiquement pour n'importe quelle position initiale dans l'espace de travail si :

- $\hat{d}^*$  est choisi entre 1.5 et 1688 cm pour  $\frac{Z}{Z^*} = 1/3$ .
- $\hat{d}^*$  est choisi entre 9 et 265 cm pour  $\frac{Z}{Z^*} = 1$ .
- $\hat{d}^*$  est choisi entre 27 et 93 cm pour  $\frac{Z}{Z^*} = 3$ .

Dans ce dernier cas, les bornes de stabilité sont plus faibles que pour l'asservissement visuel  $2D\frac{1}{2}$  car le rapport  $\frac{Z}{Z^*}$  est très grand et intervient sur le bloc non diagonal de la matrice de la boucle fermée (voir équation (3.124)), ce qui entraîne de plus fortes erreurs de poursuite si  $\hat{d}^*$  n'est pas correctement estimé. Comme pour l'asservissement visuel  $2D\frac{1}{2}$ , des résultats similaires peuvent être obtenus en considérant des erreurs de calibration de la caméra.

### 3.2.3.4 Analyse de la stabilité et de la robustesse aux erreurs sur les paramètres caméra et caméra/effecteur

Si l'hypothèse d'une calibration parfaite du repère de passage caméra/effecteur n'est pas réaliste ( $\mathbf{W} \neq \widehat{\mathbf{W}}$ ), la matrice  $\mathbf{Q}(\mathbf{e})$  est donnée par :

$$\mathbf{Q}(\mathbf{e}) = \begin{bmatrix} \nu \delta \mathbf{R} \delta \mathbf{A} & \mu \delta \mathbf{R} [\delta \mathbf{R}^T \mathbf{x} - \frac{1}{Z^*} \delta \mathbf{t} - \nu \widehat{\mathbf{x}}]_{\times} \delta \mathbf{A} \\ \mathbf{0} & \mu \mathbf{L}_{\omega} \delta \mathbf{R} \delta \mathbf{A} \end{bmatrix} \quad (3.131)$$

Il semble plus facile, par rapport à l'asservissement visuel  $2D\frac{1}{2}$ , de pouvoir démontrer des conditions pour la stabilité globale du système. Toutefois, on étudie ici seulement la stabilité locale car la stabilité globale peut être plus facilement assurée avec une commande robuste du même type que celle proposée pour l'asservissement visuel  $2D\frac{1}{2}$ .

### Stabilité asymptotique locale

#### Théorème 9

Le point d'équilibre  $\mathbf{e}^{\circ}$  du système différentiel (3.123), caractérisé par la matrice (3.131), est localement asymptotiquement stable si et seulement si  $\frac{\hat{d}^*}{d^*} > 0$  et s'il existe deux angles  $\theta_1$  et  $\theta_2$  tels que :

$$|\theta_1| + |\theta_2| < \frac{\pi}{2} \quad (3.132)$$

et tels que :

– si  $\delta\theta$  est l'angle de rotation de la matrice de rotation  $\delta \mathbf{R}$  :

$$\cos(\delta\theta) = \frac{\text{trace}(\delta \mathbf{R}) - 1}{2} \geq \cos(\theta_1) \quad (3.133)$$

– si la matrice  $\delta \mathbf{A}$  est positive :

$$\frac{\rho^{-1}((\delta \mathbf{A} + \delta \mathbf{A}^T)^{-1})}{2\sqrt{\rho(\delta \mathbf{A}^T \mathbf{A})}} \geq \cos(\theta_2) \quad (3.134)$$

La démonstration de ce théorème est très proche de celle du Théorème 3 et elle est donnée en Annexe B. On remarque que les conditions obtenues à présent sont forcément plus larges que celles du Théorème 3 car l'angle de stabilité  $\pi/2$  est partagé entre les erreurs de calibration des paramètres caméra et celles des paramètres caméra/effecteur.

## 3.3 Conclusion

Dans ce chapitre, nous avons proposé deux nouvelles méthodes d'asservissement visuel qui ne nécessitent pas les modèles des objets observés par la caméra. Ces méthodes sont

basées sur la reconstruction projective puis euclidienne du déplacement partiel de la caméra et de la structure (toujours à un facteur d'échelle près) de l'objet observé.

Comme la rotation est directement obtenue à partir de cette reconstruction, la boucle de commande en rotation est découplée de celle en translation. Cette dernière est contrôlée dans le cas du  $2D\frac{1}{2}$  en utilisant les coordonnées étendues dans l'image et, dans le cas du  $2D\frac{3}{4}$ , en utilisant les coordonnées cartésiennes réduites.

En présence d'erreurs sur les paramètres intrinsèques de la caméra, la reconstruction euclidienne n'est pas exacte, mais la matrice du système en boucle fermée est triangulaire par bloc. De plus, si la matrice de passage caméra/effecteur n'est pas parfaitement connue, la matrice du système en boucle fermée reste encore triangulaire par bloc. Il est donc possible d'étudier la robustesse de la stabilité de la boucle fermée aux incertitudes sur les paramètres du système.

En ce qui concerne la stabilité locale, le tableau 3.1 résume le type de conditions obtenues (CS pour conditions suffisantes et CNS pour conditions nécessaires et suffisantes) en fonction du type d'erreur (sur les paramètres caméra ou sur les paramètres caméra et caméra/effecteur) et de la loi de commande utilisée (commande proportionnelle ou proportionnelle robuste).

	erreurs caméra		erreurs caméra+effecteur	
	Proportionnelle	Robuste	Proportionnelle	Robuste
AV $2D\frac{1}{2}$	CNS	CNS	CS	CS
AV $2D\frac{3}{4}$	CNS		CS	

TAB. 3.1 – *Stabilité asymptotique locale*

On remarque qu'en présence d'erreurs sur la matrice de passage caméra/effecteur, l'analyse devient plus complexe et seules des conditions suffisantes peuvent être obtenues alors que, dans le cas d'erreurs sur les paramètres caméra, on a pu obtenir des conditions nécessaires et suffisantes très larges.

En ce qui concerne la stabilité globale, les résultats sont donnés dans le tableau 3.2.

	erreurs caméra		erreurs caméra+effecteur	
	Proportionnelle	Robuste	Proportionnelle	Robuste
AV $2D\frac{1}{2}$	CS	CN+S		CS
AV $2D\frac{3}{4}$	CS			

TAB. 3.2 – *Stabilité asymptotique globale*

Les conditions suffisantes obtenues en présence d'erreurs sur les paramètres de la caméra sont assez larges mais, en utilisant une commande robuste pour l'asservissement visuel  $2D\frac{1}{2}$ , on peut démontrer que les conditions nécessaires sont identiques à celles obtenues pour la stabilité locale et obtenir une condition suffisante très facile à respecter en pratique.

En présence d'erreurs de calibration caméra et caméra/effecteur, il est difficile d'obtenir des conditions sur la stabilité globale du système. Nous n'avons pas poussé davantage notre analyse car, en effectuant une commande robuste, on peut déterminer plus facilement des conditions suffisantes satisfaisantes.

Il faut souligner que toutes les conditions obtenues concernent le point de référence choisi pour effectuer l'asservissement. Si on peut assurer que ce point convergera à partir de n'importe quelle position initiale dans l'image, on n'a pas démontré que la visibilité de la tâche est assurée pendant tout l'asservissement. En effet, d'autres points de la cible peuvent, dans certains cas, sortir de l'image. Nous verrons dans le chapitre suivant comment remédier à ce problème pratique.

De plus, comme on le verra dans le chapitre suivant, la commande robuste peut s'appliquer également à l'asservissement visuel  $2D\frac{3}{4}$ . Cependant, nous n'avons pas encore démontré de manière formelle qu'elle permet de garder au moins le point de référence dans l'image.

Enfin, les résultats expérimentaux présentés dans le prochain chapitre, obtenus en utilisant un robot avec une caméra embarquée, confirment que nos schémas d'asservissement visuel apportent une amélioration importante du domaine de convergence et de la stabilité par rapport aux approches 2D et 3D classiques.

# Chapitre 4

## Résultats expérimentaux en asservissement visuel mono-caméra

Dans ce chapitre, nous présentons tout d'abord les résultats expérimentaux relatifs aux deux nouvelles méthodes d'asservissement visuel  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  proposées dans le chapitre précédent. Ensuite, nous les comparons aux méthodes les plus utilisées dans la littérature.

### 4.1 Conditions expérimentales

Les lois de commande ont été testées sur un robot manipulateur industriel MITSUBISHI PA10 à sept degrés de liberté (à la Direction des Etudes et Recherches d'EDF à Chatou) et sur un robot cartésien AFMA à six degrés de liberté (à l'IRISA). La caméra est embarquée sur l'effecteur du robot et observe un objet d'intérêt. Nous avons utilisé deux types d'objets :

- une cible plane composée de 7 disques blancs sur fond noir (voir Figure 4.1).
- une cible non planaire composée de 12 disques blancs sur fond noir (voir Figure 4.7).

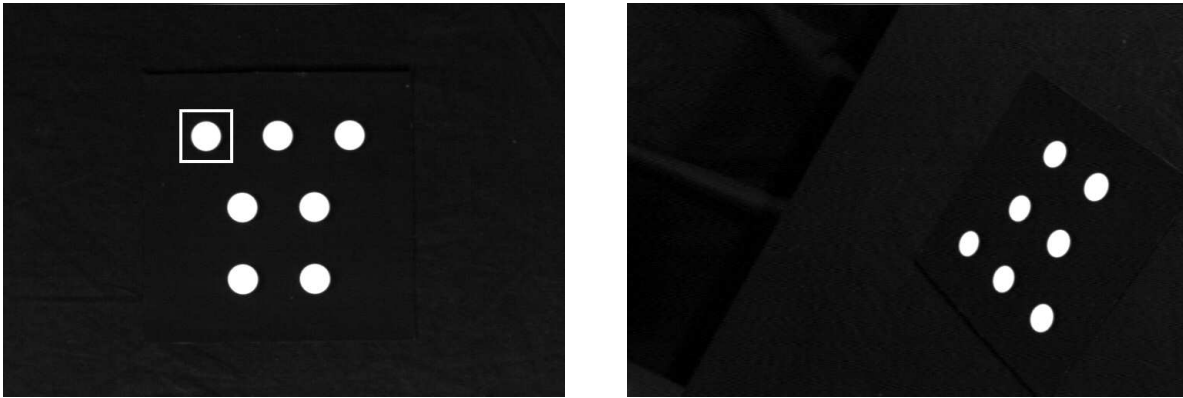
Les informations visuelles utilisées sont les coordonnées du centre de gravité de chaque disque. Avec de telles images, il est possible de réaliser la boucle d'asservissement visuel à la cadence vidéo en utilisant des cartes du commerce (en l'occurrence une carte EDIXIA IA2000). Signalons, mais le lecteur s'en apercevra, que nous n'avons absolument pas abordé les problèmes importants de traitement d'images et de mise en correspondance nécessaires à l'initialisation et au suivi des informations visuelles disponibles pour un objet véritablement complexe.

En ce qui concerne la calibration de la matrice de passage caméra/effecteur, nous l'avons mesurée "à la main" dans le cas du robot MITSUBISHI PA10. En ce qui concerne la calibration de la caméra, nous avons utilisé les paramètres fournis par le constructeur. On appellera alors cette configuration "calibration grossière" de la caméra. Afin de reproduire si besoin cette configuration dans le cas du système de l'IRISA, dont une calibration précise est disponible, nous avons introduit un erreur de 5 cm et 5 deg sur chaque composante en translation et en rotation de la matrice de passage caméra/effecteur, et une erreur de 20% sur les paramètres caméra.



## 4.2 Cible plane

Nous présentons d'abord les résultats de la tâche de positionnement par asservissement visuel  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  dans le cas d'une cible plane. Les images correspondant aux positions désirée et initiale sont illustrées sur la Figure 4.1(a) et 4.1(b) respectivement. Le déplacement de la caméra correspond à celui de la position 3 donnée dans le tableau 4.1. Le point caractéristique choisi pour réaliser l'asservissement est entouré par un carré, et il sera le même pour toutes les expériences conduites avec une cible plane. Il faut souligner que, comme l'indique le Théorème 2, le meilleur point de référence est celui qui est le plus au centre de l'image. Toutefois, nous avons voulu tester nos algorithmes dans des conditions non optimales afin de démontrer leur validité. De plus, on présente ici seulement les résultats relatifs à une commande proportionnelle classique, et nous utiliserons une commande robuste seulement quand la loi proportionnelle ne permet pas d'assurer la convergence du système.



(a) image correspondant à la position finale

(b) image correspondant à la position 3

FIG. 4.1 – Images de la cible plane pour la première expérience

Les résultats relatifs à cette expérience (en utilisant une calibration correcte du système et une la distance  $\hat{d}^*$  fixée à 50 cm, alors que sa valeur réelle est 60 cm) sont illustrés sur la Figure 4.3 pour l'asservissement visuel  $2D\frac{1}{2}$  et sur la Figure 4.4 pour l'asservissement visuel  $2D\frac{3}{4}$ . Les figures sont organisées de la manière suivante :

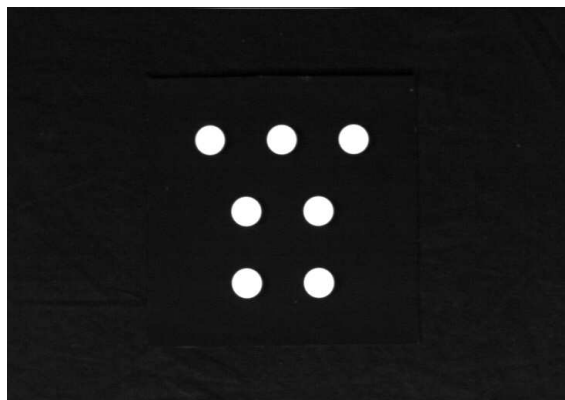
- sur la figure (a) sont représentées les erreurs sur les coordonnées des points dans l'image. L'asservissement visuel est arrêté quand l'erreur maximale est de un pixel.
- sur la figure (b), on peut voir la trajectoire des points dans l'image. La position initiale est marquée par un losange et la position désirée par un cercle. La position réellement atteinte par les points en fin d'asservissement est marqué par une croix. Si l'asservissement se termine avec succès, la croix est au centre du cercle. Dans le cas contraire, l'asservissement est arrêté dès qu'un point sort de l'image. Il faut remarquer que, pour des raisons de traitement d'images, seule la partie centrale du plan CCD est prise en compte. On considère donc qu'un point sort de l'image à 60 pixels du bord de celle-ci (de taille  $768 \times 512$ ) et non à 0 pixels.

- la vitesse de translation de la caméra (exprimée en cm/s) et sa vitesse de rotation (exprimée en deg/s) sont données sur les figures (c) et (d) respectivement.
- la figure (e) représente les coordonnées image étendues dans le cas de l’asservissement visuel  $2D\frac{1}{2}$ , et les coordonnées cartésiennes réduites dans le cas de l’asservissement visuel  $2D\frac{3}{4}$ . On rappelle qu’elles représentent la première partie de la fonction de tâche.
- la figure (f) décrit l’évolution de la rotation de la caméra entre sa position courante et celle désirée. On rappelle qu’elle représente la deuxième partie de la fonction de tâche.

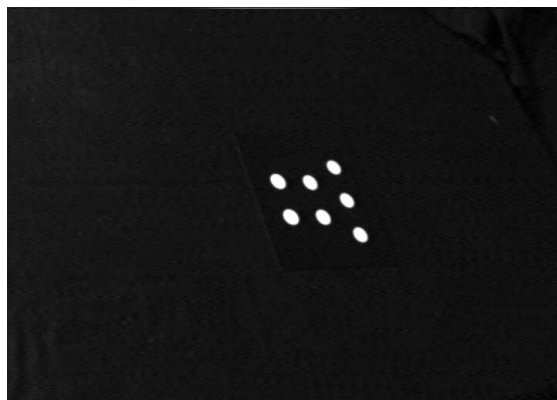
Dans cette première expérience, nous avons  $\frac{Z}{Z^*} \approx 1$ . Il n’est pas donc étonnant que les résultats illustrés sur la Figure 4.3 (asservissement visuel  $2D\frac{1}{2}$ ) et sur la Figure 4.4 (asservissement visuel  $2D\frac{3}{4}$ ) soient presque identiques. En effet, dans ce cas, on a  $\ln(\frac{Z}{Z^*}) \approx \frac{Z}{Z^*} - 1$ , et les coordonnées image étendues et les coordonnées cartésiennes réduites coïncident à la convergence. La fonction de tâche est alors la même pour les deux schémas d’asservissement.

Comme prévu, on peut remarquer sur les Figures 4.3b et 4.4b que la trajectoire du point de référence est presque rectiligne. Cela indique une bonne calibration du système. Par ailleurs, on peut signaler le nombre important d’itérations pour atteindre la convergence (1500 itérations, soit 60 secondes). Cela est dû à l’emploi d’un gain constant très faible afin que la vitesse calculée à la position initiale (très éloignée de la position désirée) ne soit pas trop élevée. Pour remédier à ce problème, nous verrons qu’il est possible d’utiliser un gain variable. Une perspective intéressante serait également de générer une trajectoire et de contrôler sa vitesse de parcours dans l’espace d’état.

Nous présentons maintenant une deuxième expérience pour laquelle le rapport  $\frac{Z}{Z^*}$  initial n’est pas proche de 1. Les images correspondant aux positions désirée et initiale sont illustrées sur les Figures 4.2a et 4.2b respectivement. Le déplacement de la caméra correspond à celui de la position 2 donnée dans le tableau 4.1. Même si  $\frac{Z}{Z^*}$  n’est pas proche de 1, les courbes des Figures 4.5 (asservissement visuel  $2D\frac{1}{2}$ ) et 4.6 (asservissement visuel  $2D\frac{3}{4}$ ) montrent que les deux schémas d’asservissement sont relativement similaires.

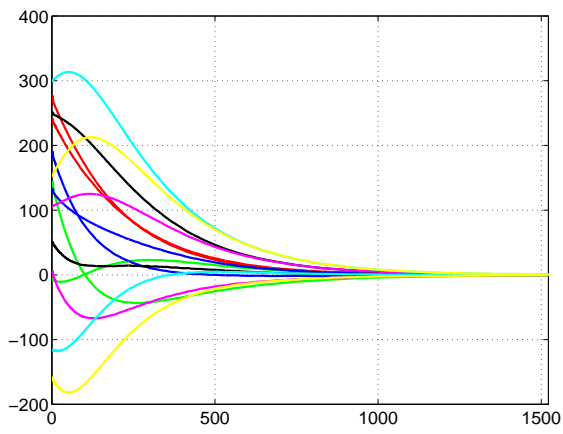


(a) image correspondant à la position finale

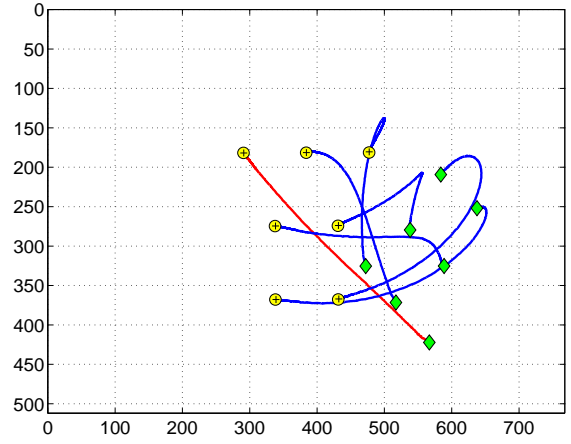


(b) image correspondant à la position 2

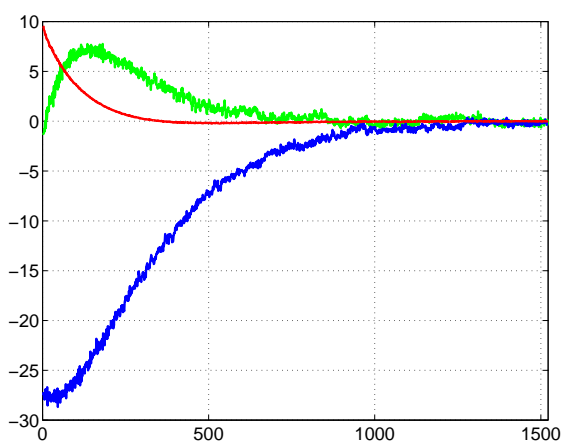
FIG. 4.2 – Images de la cible planaire pour la deuxième expérience



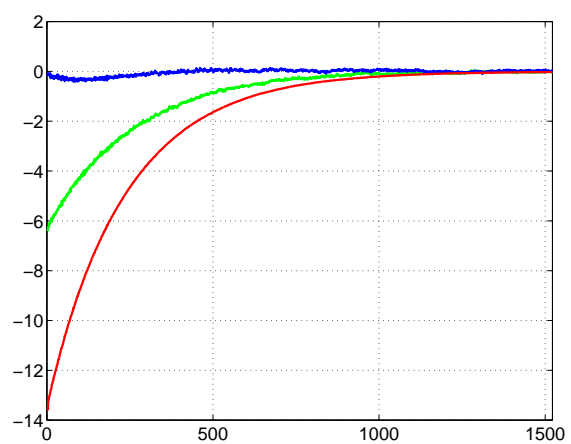
(a) erreur sur les coordonnées des points



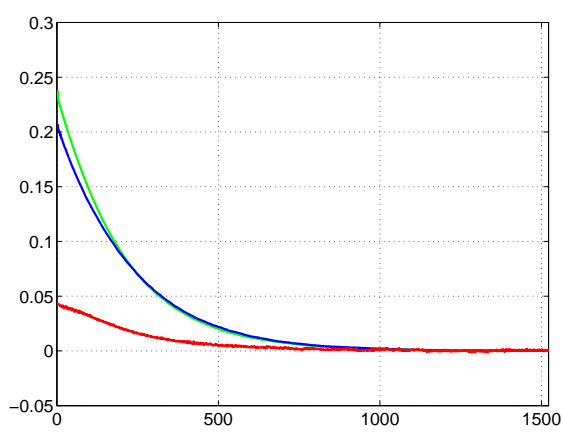
(b) trajectoire des points dans l'image



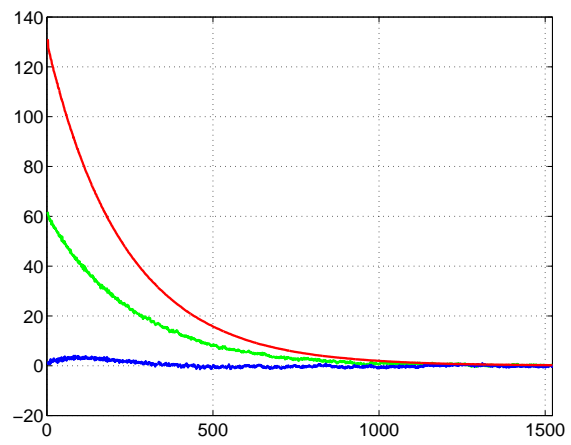
(c) vitesse de translation (cm/s)



(d) vitesse de rotation (deg/s)

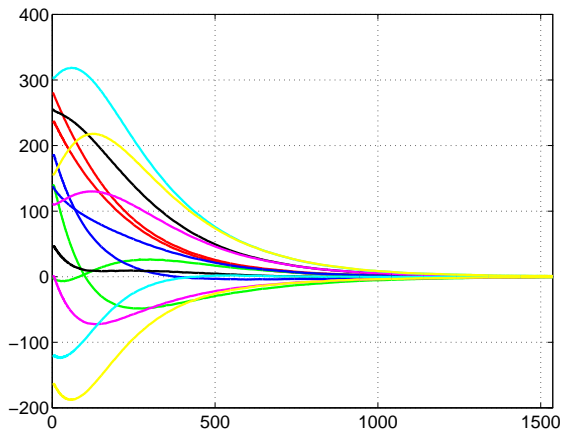


(e) coordonnées image étendues

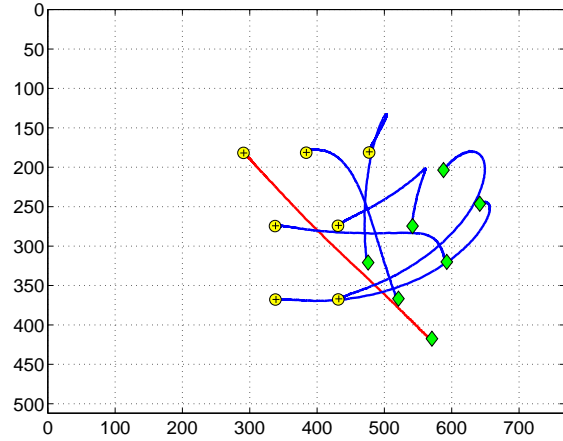


(f) rotation de la caméra (deg)

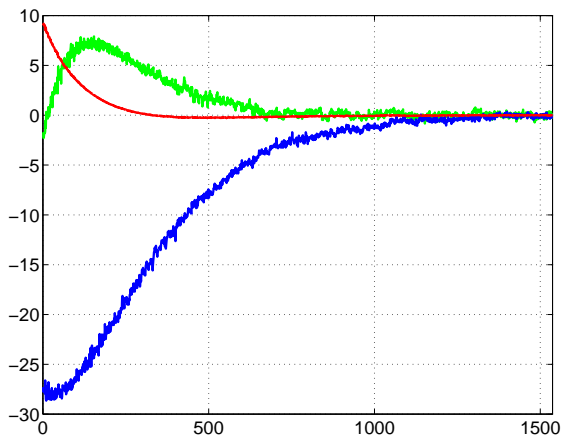
FIG. 4.3 – Position 3 : résultats de l'asservissement visuel  $2D\frac{1}{2}$



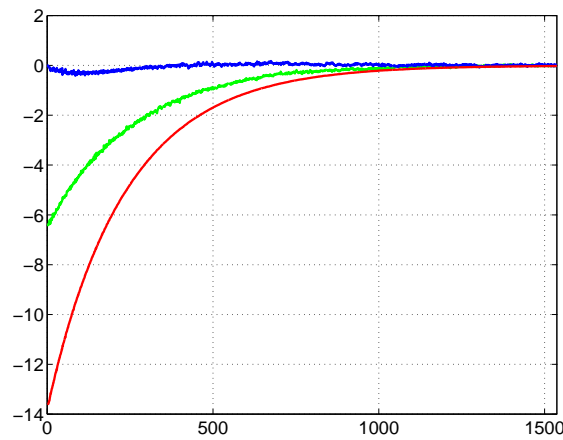
(a) erreur sur les coordonnées des points



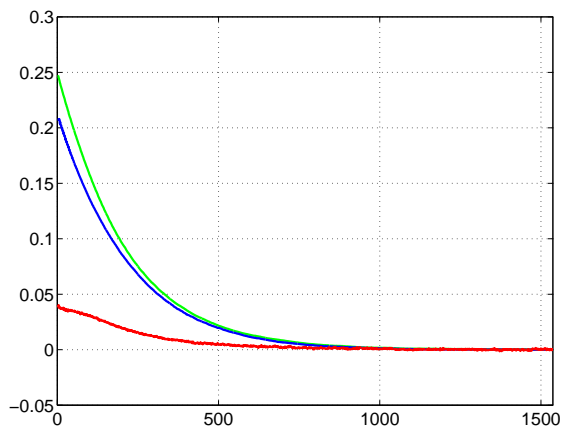
(b) trajectoire des points dans l'image



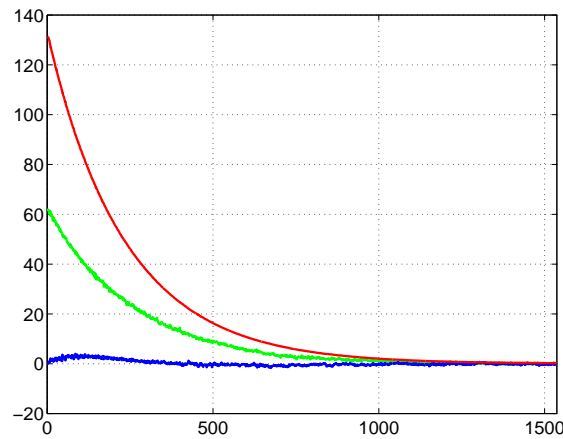
(c) vitesse de translation (cm/s)



(d) vitesse de rotation (deg/s)

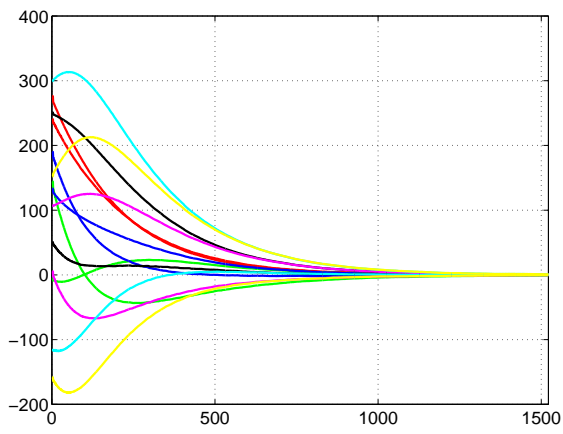


(e) coordonnées cartésiennes réduites

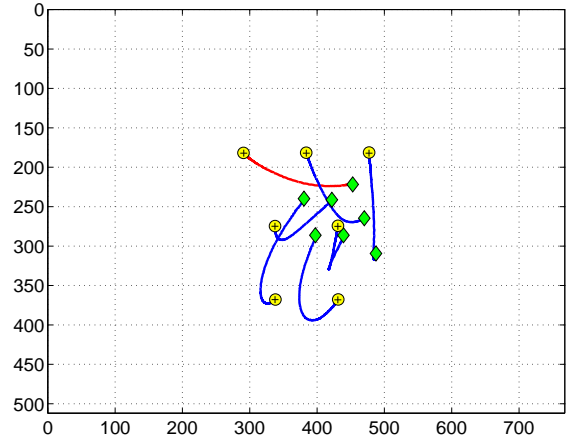


(f) rotation de la caméra (deg)

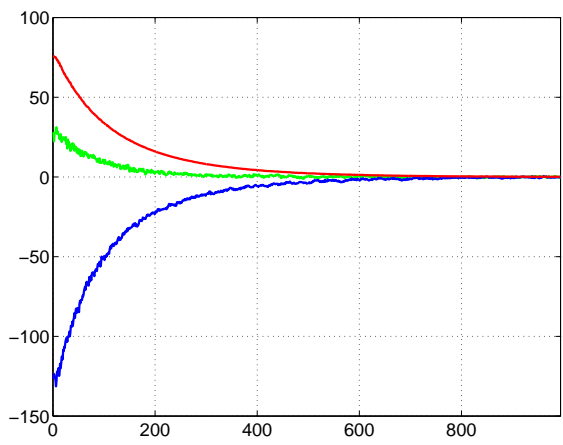
FIG. 4.4 – Position 3 : résultats de l'asservissement visuel  $2D\frac{3}{4}$



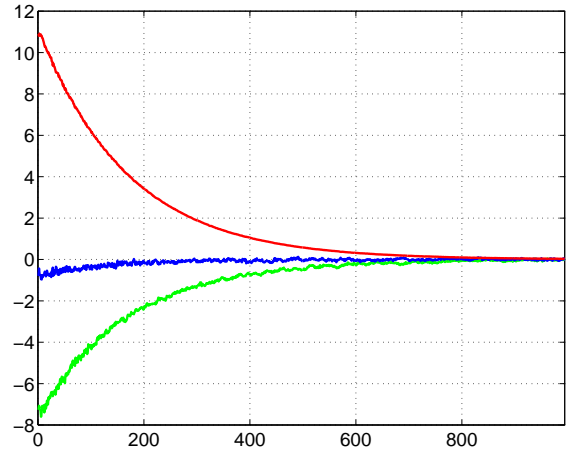
(a) erreur sur les coordonnées des points



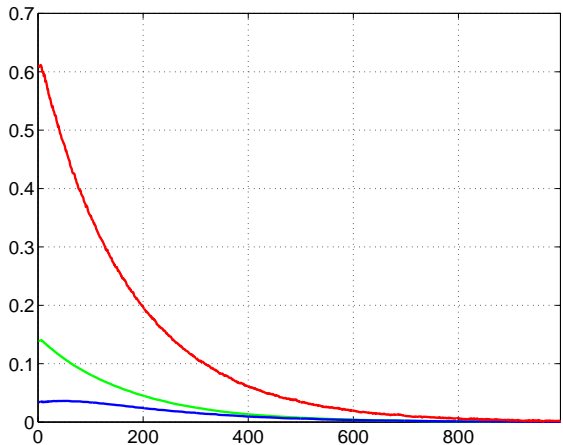
(b) trajectoire des points dans l'image



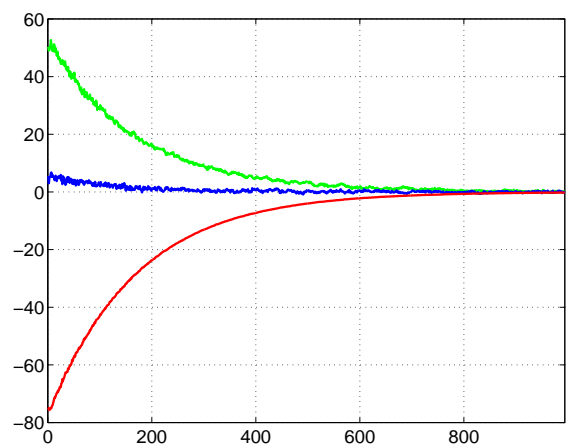
(c) vitesse de translation (cm/s)



(d) vitesse de rotation (deg/s)

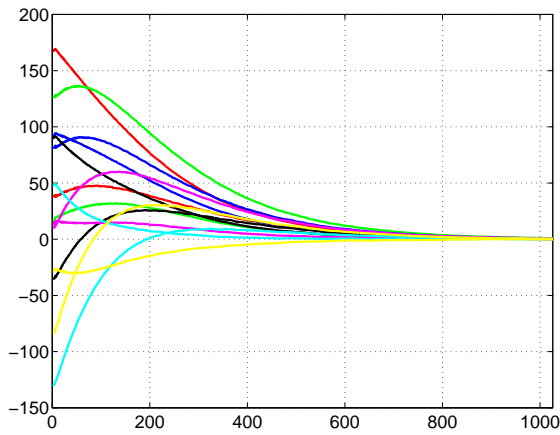


(e) coordonnées etendues dans l'image

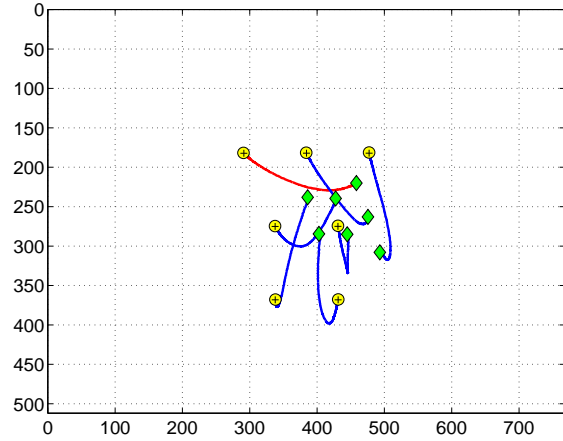


(f) rotation de la caméra (deg)

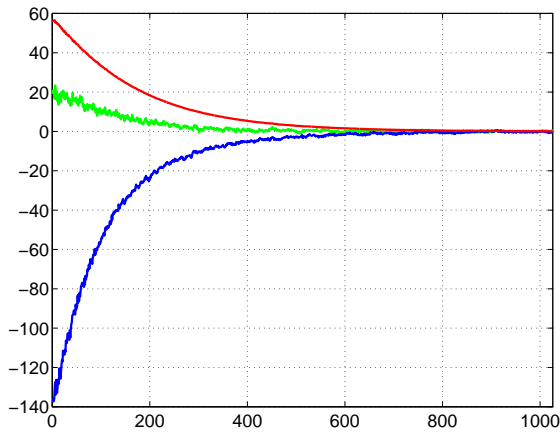
FIG. 4.5 – Position 2: résultats de l'asservissement visuel  $2D\frac{1}{2}$



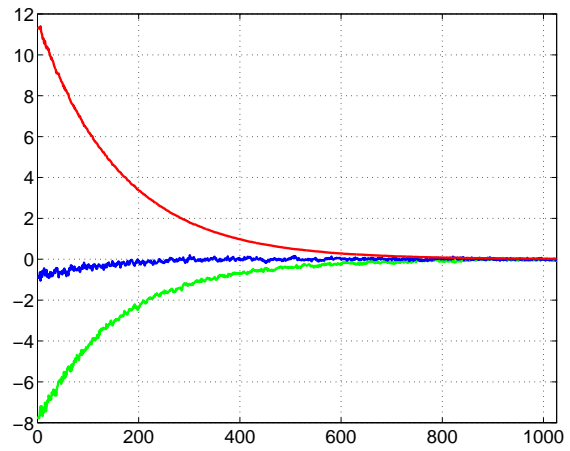
(a) erreur sur les coordonnées des points



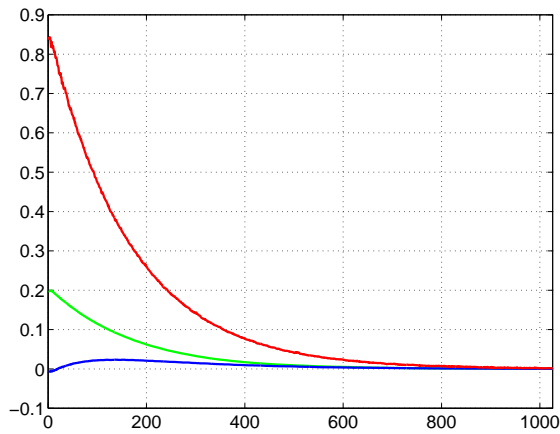
(b) trajectoire des points dans l'image



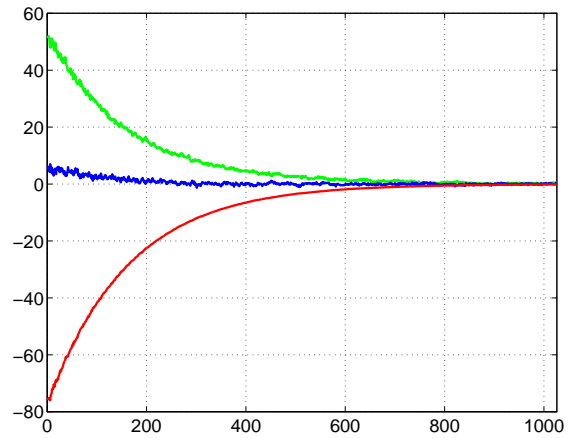
(c) vitesse de translation (cm/s)



(d) vitesse de rotation (deg/s)



(e) coordonnées étendues dans l'image



(f) rotation de la camera (deg)

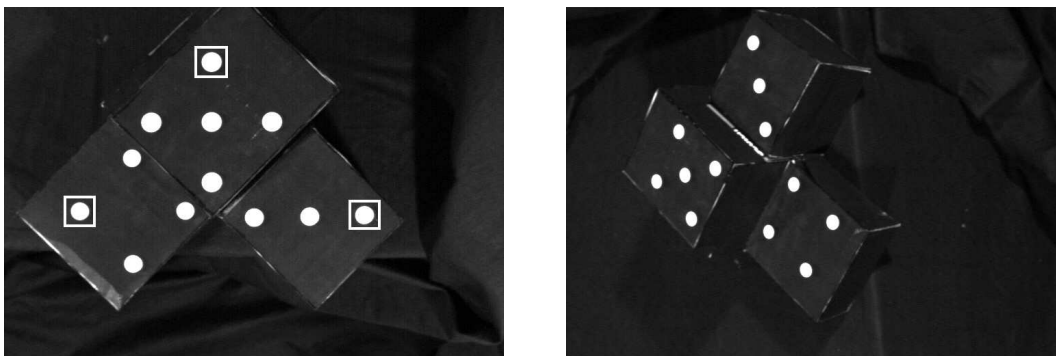
FIG. 4.6 – Position 2 : résultats de l'asservissement visuel  $2D\frac{3}{4}$

### 4.3 Cible non-planaire

Nous montrons à présent les résultats obtenus en utilisant la cible non-planaire. Comme nous l'avons déjà vu, les asservissements visuels  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  ont un comportement assez similaire. Nous ne présentons donc ici que les résultats obtenus pour l'asservissement visuel  $2D\frac{1}{2}$ .

Les images correspondant aux positions désirée et initiale sont illustrées sur la Figure 4.7a et 4.7b respectivement. Contrairement aux résultats donnés dans le Chapitre 2, les trois points définissant le plan de référence ont été choisis très dispersés dans l'image (voir la Figure 4.7a où ces points sont entourés par un carré) afin d'améliorer l'estimation du déplacement de la caméra.

Il faut remarquer que, cette fois ci, on arrête l'asservissement visuel quand l'erreur maximale est de un demi-pixel. Ceci signifie que notre méthode d'estimation de l'homographie doit être très stable quand les deux vues de l'objet se superposent.



(a) image désirée

(b) image initiale

FIG. 4.7 – Images de la cible non plane

Nous présentons les résultats des trois expériences suivantes :

- dans la première, nous allons voir le comportement de l'asservissement visuel  $2D\frac{1}{2}$  pour une calibration correcte du système.
- dans la seconde, nous allons voir le comportement de l'asservissement visuel  $2D\frac{1}{2}$  en présence d'une calibration grossière des paramètres intrinsèques de la caméra et de la matrice de passage caméra/effecteur.
- dans la troisième, nous allons étudier l'influence du choix de la distance  $\hat{d}^*$  sur le comportement du système. Nous allons également voir que notre méthode de commande avec des gains adaptatifs permet de résoudre les problèmes de convergence créés par une mauvaise estimation de cette distance.

### 4.3.1 Calibration correcte

Dans cette première expérience, nous utilisons une calibration correcte du système. Les résultats correspondants sont illustrés sur la Figure 4.8.

On peut observer la décroissance exponentielle de la fonction de tâche vers zéro. La loi de commande est stable et ne présente pas de problèmes particuliers à la convergence bien qu'une partie de la fonction de tâche soit estimée à partir de la matrice d'homographie calculée avec des images très proches.

La trajectoire du point de référence peut être encore une fois facilement identifiée car elle est quasiment rectiligne dans l'image. On peut remarquer la convergence des coordonnées de chaque point à leur valeur désirée, ce qui démontre la réalisation correcte de la tâche de positionnement.

Rappelons que nous n'utilisons dans la commande aucune connaissance à priori sur la structure 3D de l'objet. De plus, ici encore, la distance  $\hat{d}^*$  a été fixée à 50 cm, alors que sa valeur réelle est de 60 cm.

### 4.3.2 Calibration grossière

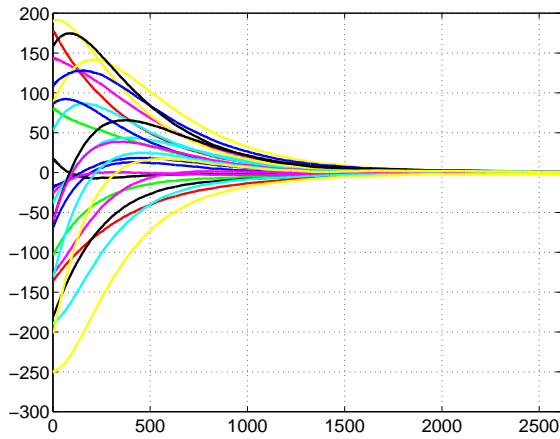
Afin de tester la robustesse de notre approche par rapport à une mauvaise calibration, nous avons ajouté une erreur de 20% sur les paramètres intrinsèques de la caméra et une erreur supplémentaire sur la translation (5 cm sur chaque axe) et sur la rotation (5 degrés sur chaque axe) de la matrice de transformation entre la caméra et l'effecteur du robot. Les résultats obtenus en utilisant la même position initiale que pour l'expérience précédente sont illustrés sur la Figure 4.9.

Comme on peut le voir sur les Figures 4.9(e) et 4.9(f), la convergence de la fonction de tâche n'est plus exactement exponentielle. Ceci est dû à la calibration grossière de la caméra, de la transformation caméra/effecteur et de l'estimation approximative de  $d^*$  (qui a une influence moins importante en utilisant une calibration correcte). Par conséquent, contrairement à l'expérience précédente, la trajectoire du point de référence dans l'image n'est plus une ligne droite. De même, la trajectoire des autres points dans l'image est nettement différente.

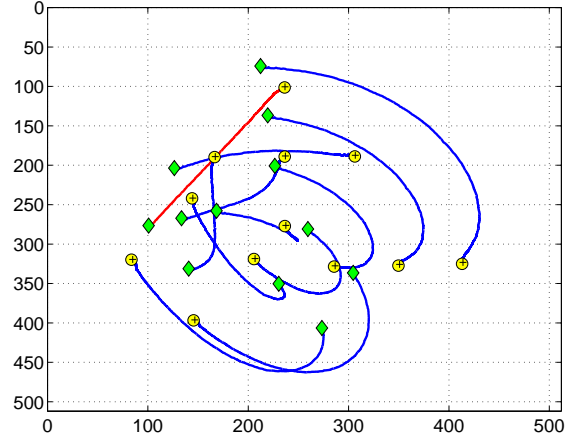
Toutefois, même dans ce cas très défavorable, on remarque la stabilité et la robustesse de la loi de commande. On peut aussi noter qu'un nombre plus important d'itérations est nécessaire pour atteindre la convergence.

Finalement, malgré toutes les erreurs introduites, on peut noter sur la Figure 4.9(a) la convergence des coordonnées de tous les points vers leurs valeurs désirées.

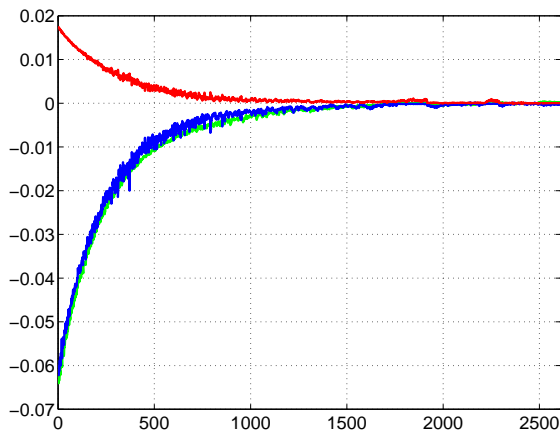




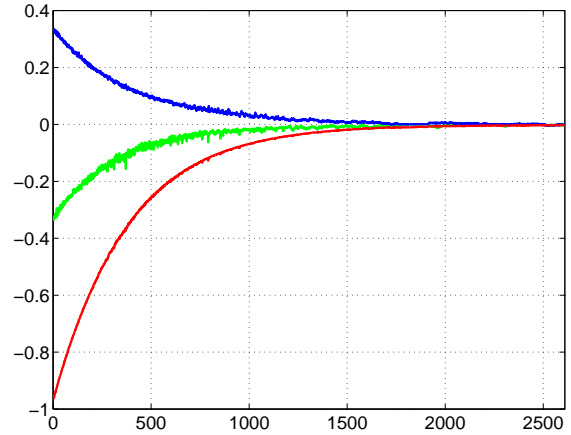
(a) erreur sur les coordonnées



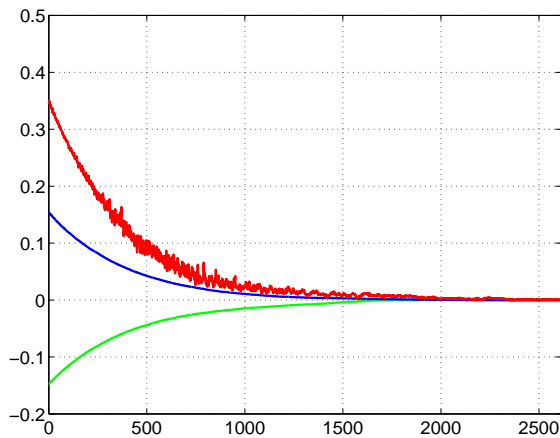
(b) trajectoire dans l'image



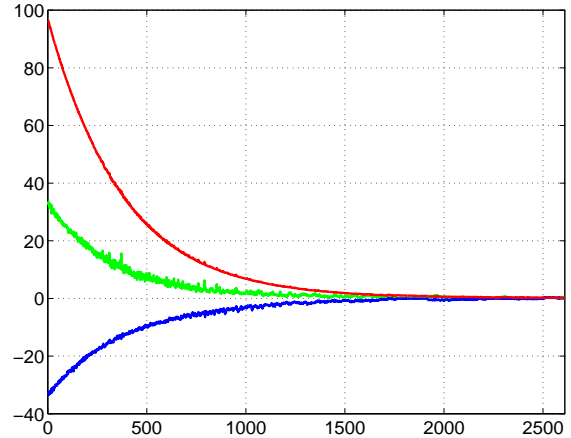
(c) vitesse de translation (cm/s)



(d) vitesse de rotation (deg/s)

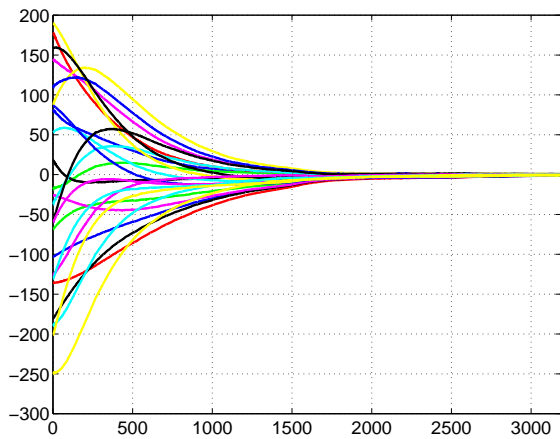


(e) coordonnées image étendues

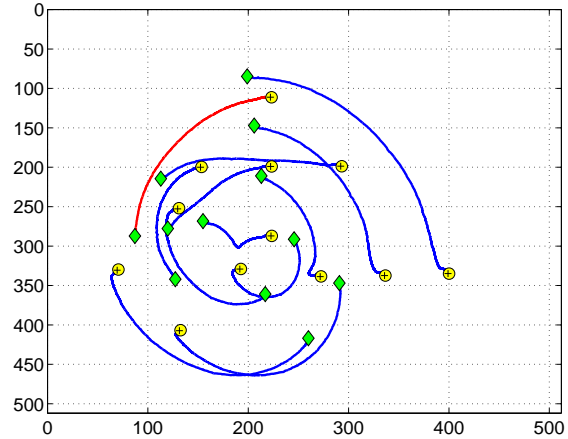


(f) rotation (deg)

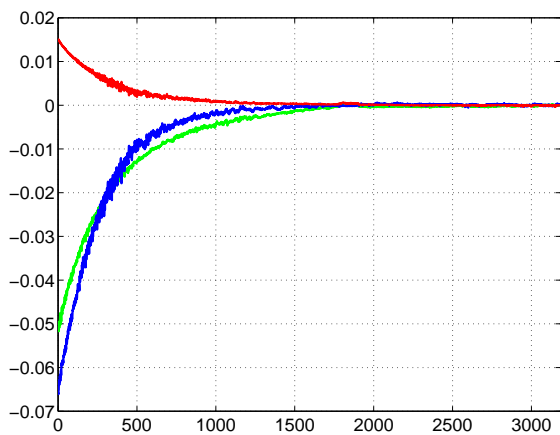
FIG. 4.8 – Résultats de l'asservissement visuel  $2D\frac{1}{2}$  avec une calibration correcte du système



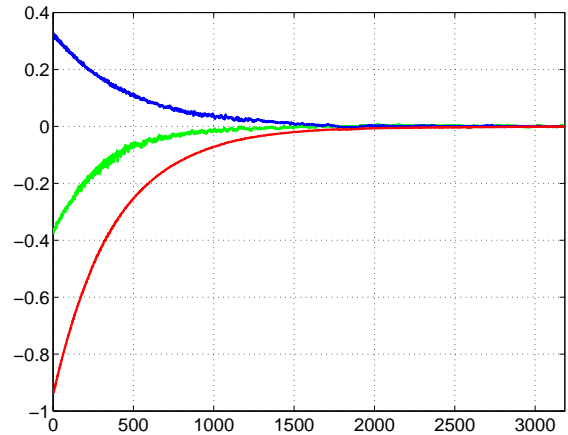
(a) erreur sur les coordonnées



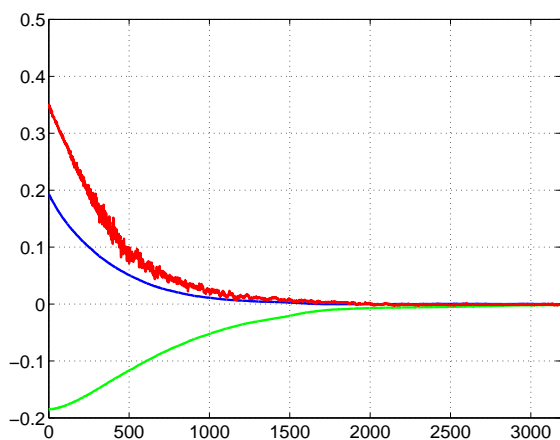
(b) trajectoire dans l'image



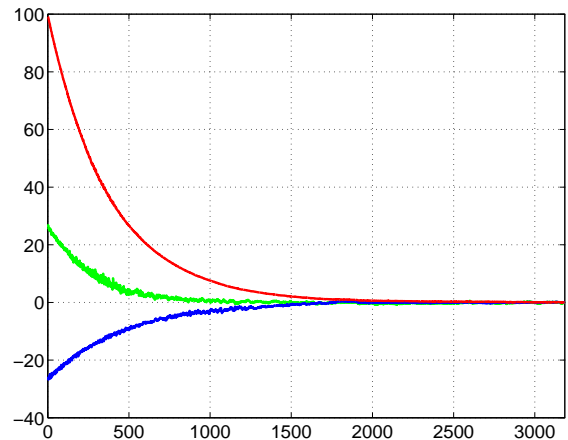
(c) vitesse de translation (cm/s)



(d) vitesse de rotation (deg/s)



(e) coordonnées image étendues



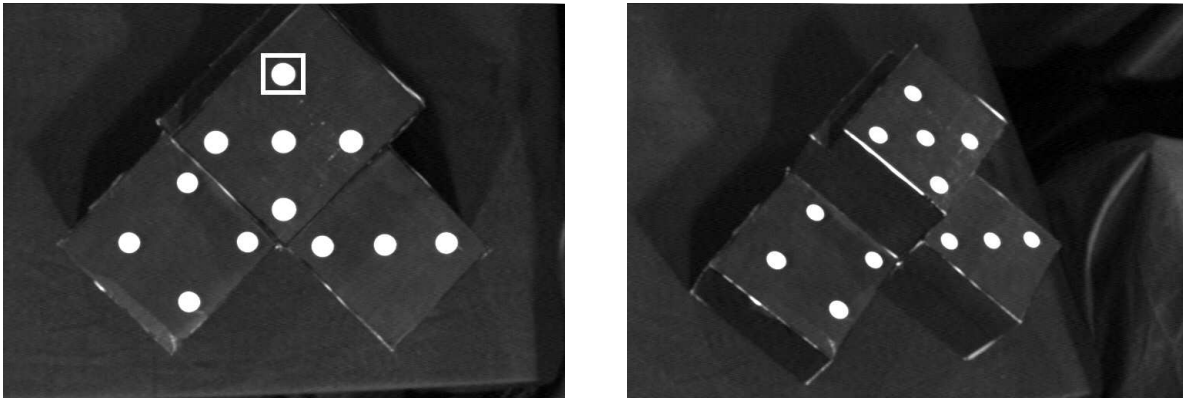
(f) rotation (deg)

FIG. 4.9 – Résultats de l'asservissement visuel  $2D\frac{1}{2}$  avec une calibration grossière du système

### 4.3.3 Influence du choix de la distance $\hat{d}^*$

Nous allons à présent vérifier expérimentalement les conclusions théoriques obtenues dans le chapitre précédent en ce qui concerne l'erreur d'estimation sur la distance  $\hat{d}^*$ . Nous allons considérer ici le cas d'une calibration correcte de tous les autres paramètres du système. On rappelle que nous avons obtenu une condition suffisante pour la stabilité globale de l'asservissement visuel  $2D\frac{1}{2}$ .

Afin de réaliser les expériences, un déplacement en rotation de la caméra de 30 degrés autour des axes  $\vec{x}$  et  $\vec{y}$  a été réalisé ainsi qu'un déplacement en translation afin que l'objet soit visible dans l'image à la position initiale (voir la Figure 4.10, où le point de référence choisi est entouré par un carré). Les rotations effectuées doivent provoquer une erreur de poursuite si  $\hat{d}^*$  ne correspond pas à la distance réelle.



(a) image désirée

(b) image initiale

FIG. 4.10 – Images de la cible non plane

Quatre valeurs différentes de  $\hat{d}^*$  ont été testées, à savoir  $\hat{d}^* = 60$  cm (qui est à peu près la vraie valeur),  $\hat{d}^* = 50$  cm,  $\hat{d}^* = 40$  cm, et enfin  $\hat{d}^* = 30$  cm. Les résultats sont illustrés sur la Figure 4.11. On peut voir sur la Figure 4.11(b) que la norme de l'erreur sur les coordonnées du point de référence augmente au fur et mesure que l'erreur d'estimation sur  $\hat{d}^*$  augmente. Toutefois, l'erreur sur la fonction de tâche diminue pour toutes les valeurs de  $\hat{d}^*$  (voir la Figure 4.11(c) où les tracés des quatre expériences sont presque superposés). On respecte donc la condition suffisante du Théorème 2 qui fixait à environ 20 cm la distance critique si tous les autres paramètres du système étaient exacts. En raison de l'importance de la rotation effectuée, il nous a été impossible d'utiliser une valeur encore plus faible de  $\hat{d}^*$  (pour tester si la norme de la fonction de tâche croit à partir de  $\hat{d}^* = 20$  cm) car les trajectoires dans l'image montrent un dépassement qui fait sortir le point de référence de l'image quand  $\hat{d}^* = 30$  cm. Le Théorème 2 ne prend en effet pas en compte la condition de visibilité de la tâche.

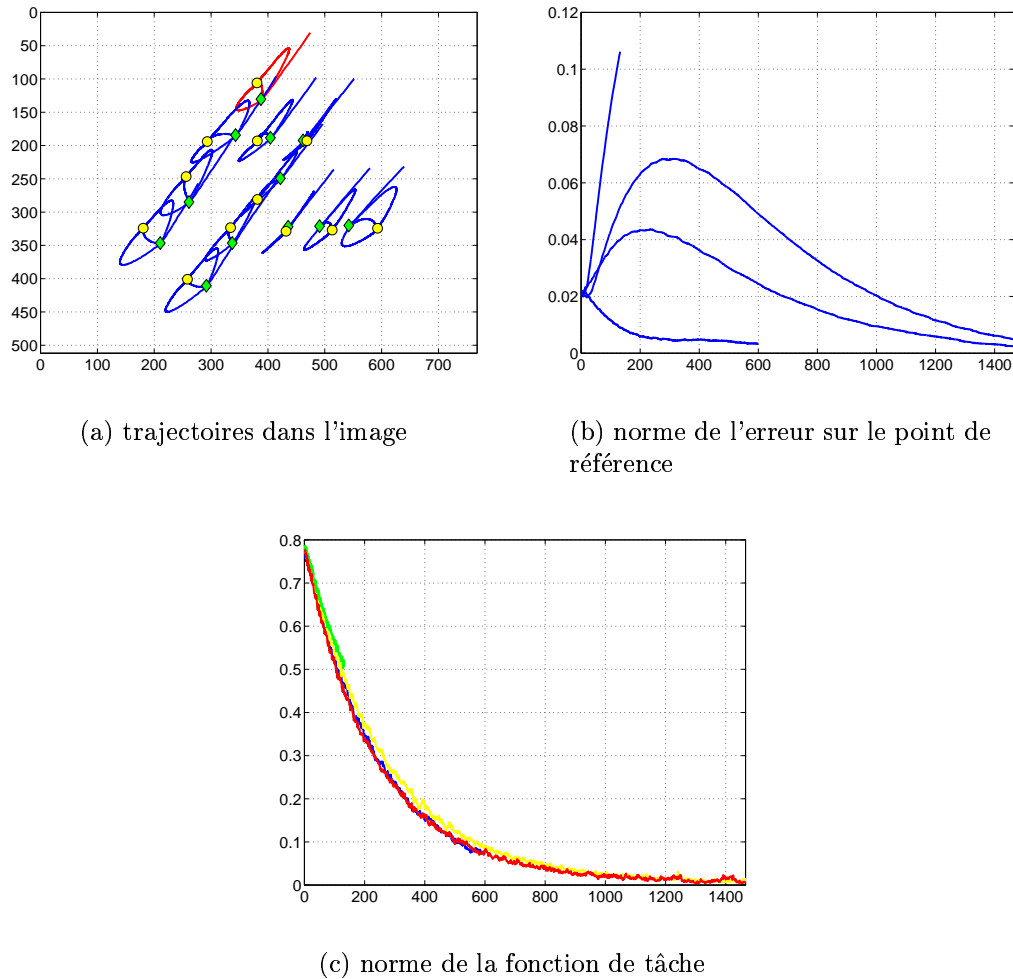


FIG. 4.11 – Résultats de l'asservissement visuel  $2D_{\frac{1}{2}}$  avec une distance  $\hat{d}^*$  variable

On effectue maintenant la même expérience pour  $\hat{d}^* = 30$  cm mais avec une loi de commande robuste afin de vérifier la condition de visibilité de la tâche. On rappelle que le Théorème 5 n'est valable que pour le point de référence. Il existe donc des cas critiques pour lesquels on peut observer un échec de l'asservissement.

L'exemple d'un cas critique est donné sur la Figure 4.12. L'objet est proche du bord supérieur de l'image. La rotation autour de l'axe  $\vec{z}$  est donc limité par la diminution du gain  $f_v$ . Toutefois, on ne peut pas empêcher l'objet de sortir en partie de l'image car le point de référence courant doit se déplacer vers le haut afin de rejoindre sa position désirée.

Afin de réaliser l'asservissement, il faudrait introduire un mouvement transitoire de zoom arrière qui permette à la cible d'effectuer la rotation autour de l'axe  $\vec{z}$  sans que la cible ne sorte de l'image. Une des perspectives de ce travail est donc de résoudre ces cas critiques. Signalons aussi qu'une perte de quelques informations visuelles n'a pas d'importance tant que le nombre d'informations disponibles est suffisant pour réaliser la tâche.

Une autre solution est de modifier le choix du point de référence au cours de l'asservissement, mais il faut alors assurer la continuité de la commande. En pratique, nous avons utilisé la coordonnée  $x$  du point le plus proche des bords droit et gauche de l'image, et la coordonnée  $y$  du point le plus proche des bords haut et bas de l'image. Cela permet de tenter de conserver l'ensemble des points de l'objet dans l'image tout en assurant la continuité de la loi de commande.

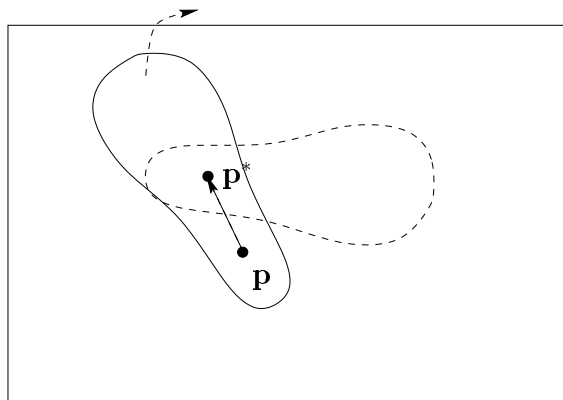
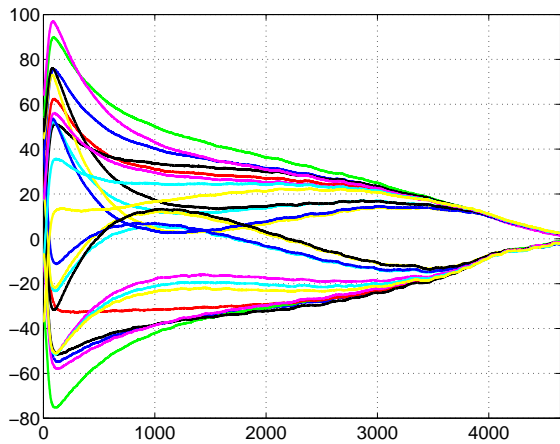


FIG. 4.12 – Cas d'échec de la commande robuste

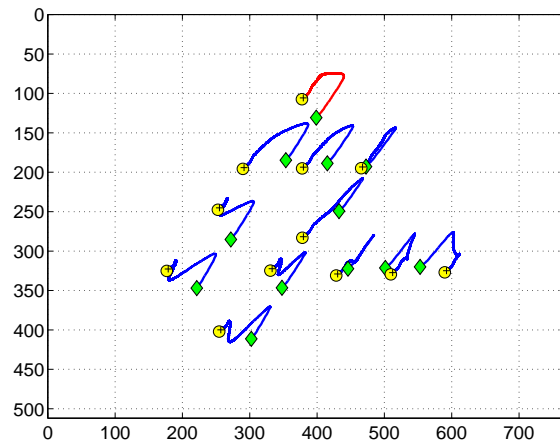
Les résultats obtenus sont illustrés sur la Figure 4.13. Dans cette expérience, le gain  $f_w$  est fixé égal à 1 alors que les gains  $f_u$  et  $f_v$  sont libres d'évoluer (voir paragraphe 3.1.3). On peut noter que  $f_u$  reste égal à 1 pendant tout l'asservissement car la cible reste toujours centrée horizontalement. Par contre, le gain  $f_v$  est initialement égal à 0.6 car le point inférieur de la cible est au bord de l'image.

Quand l'asservissement commence, la cible se déplace vers le haut de l'image et par conséquent  $f_v$  augmente. Toutefois, comme la cible occupe une très grande partie de l'image, le point supérieur de la cible s'approche à son tour du bord de l'image, ce qui entraîne une baisse rapide de  $f_v$ . Durant cette phase, on peut voir sur la Figure 4.13f que la norme de l'erreur sur les coordonnées du point de référence double d'amplitude. Ce dépassement est provoqué par la très mauvaise estimation de  $\hat{d}^*$ . La chute du gain  $f_v$ , provoquée par l'approche du point supérieur au bord de l'image, réduit la vitesse de rotation, ce qui permet à la cible de rester dans l'image. Le gain  $f_v$  augmente ensuite graduellement dès que la caméra s'approche de sa position désirée (où une mauvaise estimation de  $\hat{d}^*$  n'a plus aucune influence sur la stabilité du système comme nous l'avons démontré dans le Théorème 1).

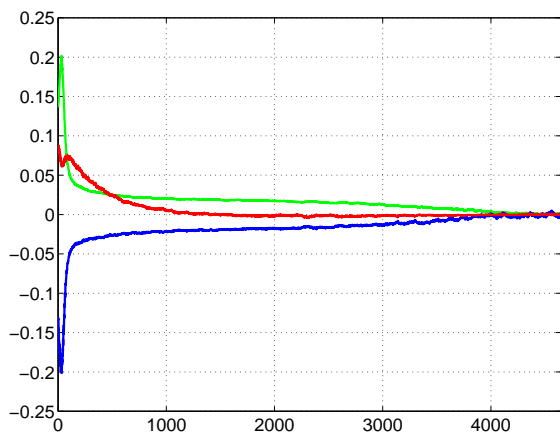
Il faut remarquer que le nombre d'itérations nécessaire pour arriver à la convergence est très important. Toutefois, l'objectif ici est seulement de permettre à la loi de commande de respecter la condition de visibilité de la tâche. La vitesse de convergence du système peut être augmentée en effectuant une commande adaptative sur les paramètres du système ou en augmentant graduellement les gains (des résultats de ce type seront présentés à la fin de ce chapitre).



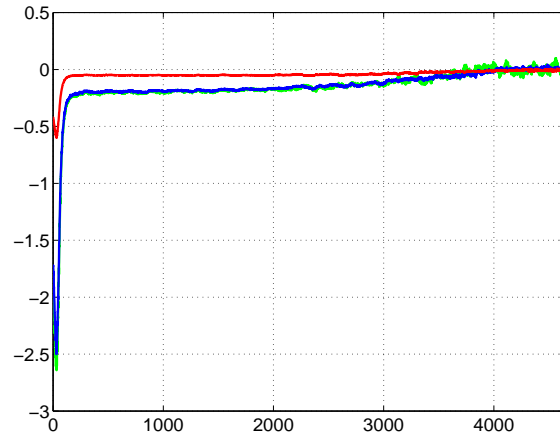
(a) erreur sur les coordonnées



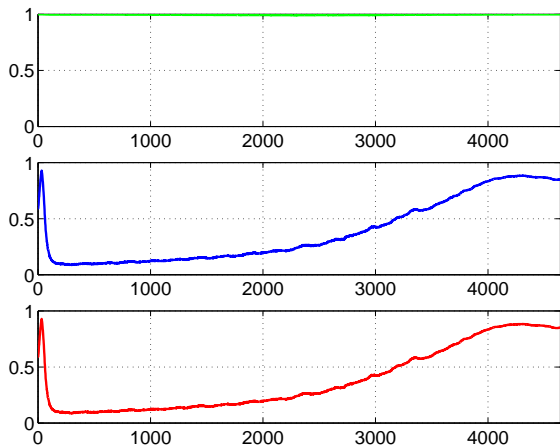
(b) trajectoire dans l'image



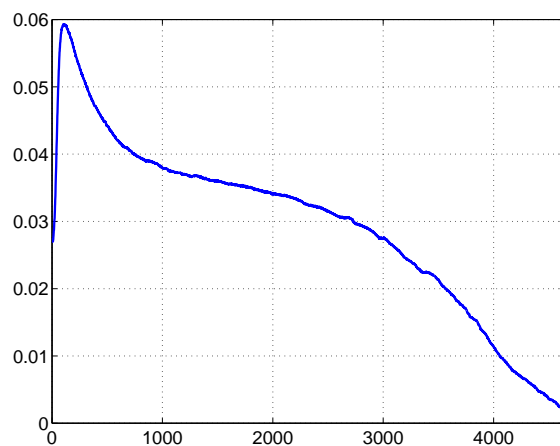
(c) vitesse de translation (cm/s)



(d) vitesse de rotation (deg/s)



(e) gains  $f_u$ ,  $f_v$  et  $f_u f_v$



(f) norme de l'erreur sur le point de référence

FIG. 4.13 – Résultats de l'asservissement visuel  $2D\frac{1}{2}$  avec une commande robuste

## 4.4 Comparaison entre les méthodes d'asservissement visuel

Nous comparons à présent les techniques d'asservissement visuel  $2D\frac{1}{2}$  (notée **AV 2D $\frac{1}{2}$** ) et  $2D\frac{3}{4}$  (notée **AV 2D $\frac{3}{4}$** ) par rapport aux schémas suivants :

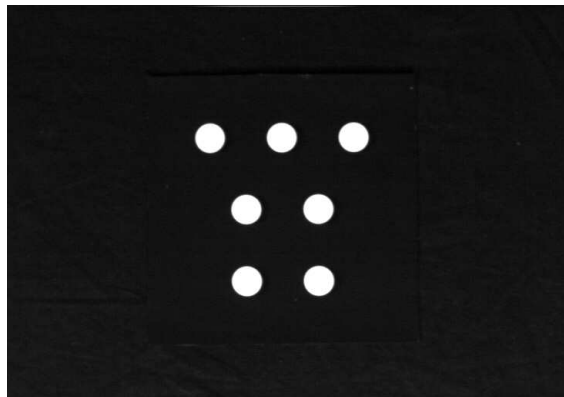
- asservissement visuel 2D (voir paragraphe 1.4) :
  - **AV 2Dc** : en utilisant une matrice d'interaction constante ;
  - **AV 2Dv** : avec une matrice d'interaction calculée à chaque itération.
- asservissement visuel 3D (voir paragraphe 1.3) en utilisant pour le calcul de pose la méthode non-linéaire de Lowe (qui semble être à ce jour la plus précise) :
  - **AV 3Dtr** : contrôle basé sur le déplacement de la caméra entre l'image courante et l'image désirée ;
  - **AV 3Dpr** : contrôle basé sur un point 3D et la rotation de la caméra entre l'image courante et l'image désirée ;
- asservissement visuel mixte (avec calcul de pose par la méthode de Lowe). **AV3Dpt** : le mouvement de la caméra est alors contraint de manière à effectuer une ligne droite entre la position initiale et désirée. Pour conserver la cible dans l'image, les coordonnées d'un point 2D sont également utilisées. Finalement, on considère aussi dans la fonction de tâche la rotation de la caméra autour de son axe optique. Cette nouvelle méthode d'asservissement visuel est décrite en Annexe C.

Les expérimentations ont été effectuées à partir de trois positions différentes, données dans le tableau 4.1, qui présentent les caractéristiques suivantes :

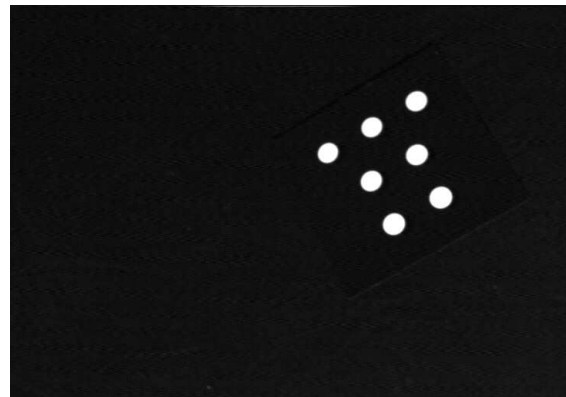
- la position 1 est une position de difficulté moyenne car le déplacement dans l'espace 3D et dans l'image n'est pas très grand (voir Figure 4.14b) ;
- la position 2 est une position difficile car même si l'objet est centré dans l'image à la position initiale, le déplacement dans l'espace 3D est important (voir Figure 4.14c) ;
- la position 3 est une position très difficile car le déplacement dans l'espace 3D est très important (surtout les rotation autour des axes  $\vec{x}$  et  $\vec{z}$ ) et la cible se trouve au bord de l'image (voir Figure 4.14d).

position	translation (mm)	rotation $\mathbf{u}\theta$ (deg)
1	39.6 278.7 -130.6	21.6 -16.0 27.2
2	367.7 557.5 -86.6	50.9 4.3 -75.0
3	-343.4 106.7 85.3	62.4 1.6 130.6

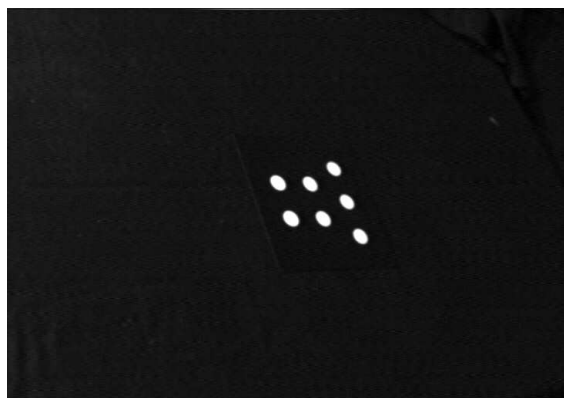
TAB. 4.1 – Déplacements de la caméra dans le repère désiré



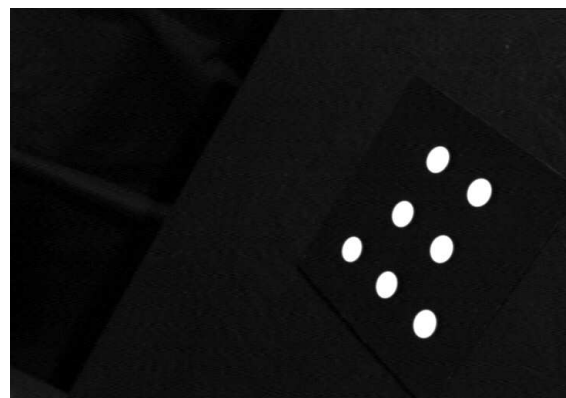
(a) image correspondante à la position finale



(b) image correspondante à la position 1



(c) image correspondante à la position 2



(d) image correspondante à la position 3

FIG. 4.14 –



Pour chaque position initiale et pour chaque méthode d'asservissement, nous présentons les résultats des expériences suivantes :

- asservissement visuel avec calibration parfaite (indiqué par "c00-e00").
- asservissement visuel avec erreurs de calibration sur la matrice de passage caméra/effecteur. Sur chaque composante en translation et en rotation, on ajoute une erreur de différentes amplitudes :
  - 5 cm et 5 deg (indiqué par "c00-e05");
  - 10 cm et 10 deg (indiqué par "c00-e10");
  - 20 cm et 20 deg (indiqué par "c00-e20").
- asservissement visuel avec erreurs de calibration de la caméra. L'angle entre les axes d'un pixel est fixé à  $\frac{\pi}{2}$  et le point principal est choisi au centre de l'image. Sur chaque rapport mètre/pixel, on ajoute une erreur de différentes amplitudes :
  - 30% (indiqué par "c30-e00");
  - 50% (indiqué par "c50-e00").
- asservissement visuel avec erreurs de calibration de la caméra et de la matrice de passage caméra/effecteur. Nous avons ajouté les erreurs suivantes :
  - 10 cm et 10 deg plus 30% (indiqué par "c30-e00");
  - 20 cm et 20 deg plus 50% (indiqué par "c50-e00").

Dans tous les cas, on utilise une loi proportionnelle afin de réguler la fonction de tâche à zéro ( $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ ). La valeur du gain proportionnel  $\lambda$  est déterminée de la manière suivante : nous calculons un gain  $\lambda_t$  tel que la vitesse de translation initiale soit de  $v_t$  cm/s et un gain  $\lambda_r$  tel que la vitesse de rotation initiale soit de  $v_r$  deg/s. La valeur du gain proportionnel est alors  $\lambda = \min(\lambda_t, \lambda_r)$  :

- pour la position 1, nous avons choisi  $v_t = 10$  cm/s et  $v_r = 10$  deg/s;
- pour les positions 2 et 3, nous avons choisi  $v_t = 15$  cm/s et  $v_r = 15$  deg/s;

Un tel choix du gain  $\lambda$  nous permettra de déterminer de manière qualitative la vitesse de convergence de chaque méthode. La valeur du gain et le nombre d'itérations nécessaires à la convergence (l'asservissement est arrêté quand l'erreur maximale est de 1 pixel) sont donnés en fonction de la calibration du système et de la méthode d'asservissement utilisée :

- dans le tableau 4.2, pour les résultats en partant de la position 1 ;
- dans le tableau 4.3, pour les résultats en partant de la position 2 ;
- dans le tableau 4.4, pour les résultats en partant de la position 3 ;

Dans chaque case de ces tableaux, la première valeur indiquée (sur la partie supérieure) est le nombre d'itérations, et la seconde est la valeur du gain  $\lambda$ . Une étoile indique que l'asservissement visuel n'a pas convergé (c'est-à-dire qu'un point est sorti de l'image).

POS 1	AV 2Dc	AV 2Dv	AV 2D $\frac{1}{2}$	AV 2D $\frac{3}{4}$	AV 3Dpr	AV 3Dtp	AV 3Dtr
c00-e00	$\frac{295}{0.41}$	$\frac{499}{0.23}$	$\frac{462}{0.26}$	$\frac{539}{0.26}$	$\frac{507}{0.26}$	$\frac{448}{0.28}$	$\frac{474}{0.26}$
c00-e05	$\frac{371}{0.42}$	$\frac{524}{0.23}$	$\frac{604}{0.26}$	$\frac{628}{0.26}$	$\frac{595}{0.26}$	$\frac{568}{0.28}$	$\frac{541}{0.26}$
c00-e10	$\frac{442}{0.41}$	$\frac{730}{0.23}$	$\frac{746}{0.25}$	$\frac{685}{0.26}$	$\frac{714}{0.26}$	$\frac{689}{0.28}$	$\frac{721}{0.26}$
c00-e20	$\frac{621}{0.37}$	$\frac{953}{0.23}$	$\frac{922}{0.23}$	$\frac{812}{0.26}$	$\frac{989}{0.25}$	$\frac{939}{0.26}$	$\frac{838}{0.24}$
c30-e00	$\frac{595}{0.49}$	$\frac{888}{0.22}$	$\frac{368}{0.29}$	$\frac{398}{0.29}$	$\frac{716}{0.25}$	$\frac{639}{0.27}$	$\frac{765}{0.25}$
c50-e00	$\frac{855}{0.50}$	$\frac{1128}{0.21}$	$\frac{397}{0.28}$	$\frac{356}{0.31}$	$\frac{905}{0.25}$	$\frac{875}{0.26}$	$\frac{874}{0.25}$
c30-e10	$\frac{539}{0.49}$	$\frac{805}{0.22}$	$\frac{575}{0.27}$	$\frac{574}{0.29}$	$\frac{707}{0.25}$	$\frac{623}{0.27}$	$\frac{716}{0.25}$
c50-e20	$\frac{634}{0.50}$	$\frac{1028}{0.21}$	$\frac{772}{0.26}$	$\frac{751}{0.26}$	*	$\frac{950}{0.26}$	*

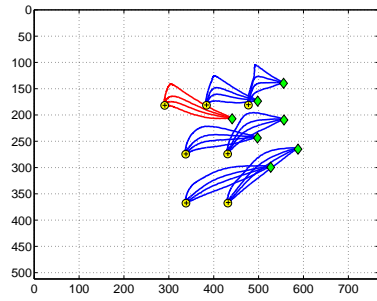
TAB. 4.2 – Comparaison des méthodes pour la position 1

Comme cela était prévisible, dès qu'on introduit des erreurs de calibration sur la matrice de passage caméra/effecteur, le temps de convergence augmente pour toutes les méthodes. Il en est de même pour les erreurs de calibration de la caméra, sauf pour nos deux méthodes d'asservissement visuel  $\mathbf{AV\ 2D}\frac{1}{2}$  et  $\mathbf{AV\ 2D}\frac{3}{4}$  car les erreurs introduites peuvent compenser l'erreur inévitable commise sur l'estimation "à vue" de la distance  $\hat{d}^*$  ou avoir un effet d'augmentation du gain.

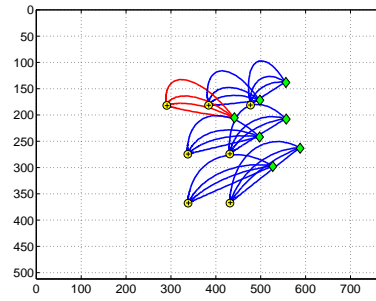
Nous remarquons que l' $\mathbf{AV\ 2Dc}$  est toujours la méthode la plus rapide à converger si on exclut le cas de figure décrit ci dessus. Le temps de convergence dépend à la fois du gain constant utilisé dans la commande, des erreurs de calibration du système, et des erreurs de mesure. On peut donc en déduire que les méthodes les plus rapides sont, d'une part, plus performantes et, d'autre part, moins sensibles aux erreurs de calibration et de mesure. Toutefois, il faut prendre cette conclusion avec une certaine précaution car la vitesse de convergence dépend du gain de la commande qui n'est pas le même pour toutes les méthodes. Le fait d'avoir choisi la même amplitude de vitesse initiale pour toutes les méthodes prouve cependant que les méthodes qui convergent plus rapidement sont celles qui déterminent des trajectoires les plus courtes à parcourir. Comme le niveau de bruit est ici de 1/10 de pixel, nous allons à présent nous intéresser à l'influence des erreurs de calibration sur le comportement du système. Une étude de la sensibilité au bruit de mesure sera présentée dans la prochaine section.

On peut remarquer sur la Figure 4.15 que les erreurs sur la matrice de passage caméra/effecteur ont à peu près le même effet sur la trajectoire des points dans l'image pour l'ensemble des méthodes. On peut aussi noter que les schémas d'asservissement visuel qui utilisent un calcul de pose sont plus sensibles aux erreurs de calibration, comme le démontre le fait que les méthodes  $\mathbf{AV\ 3Dpr}$  et  $\mathbf{AV\ 3Dtr}$  ne convergent pas pour des très fortes erreurs de calibration (voir la ligne "c50-e20" du tableau 4.2) et que les trajectoires des points de la cible s'éloignent sensiblement de la trajectoire obtenue en absence d'erreurs pour l'ensemble des méthodes 3D (comme on peut le voir sur les Figures 4.16 et 4.17).

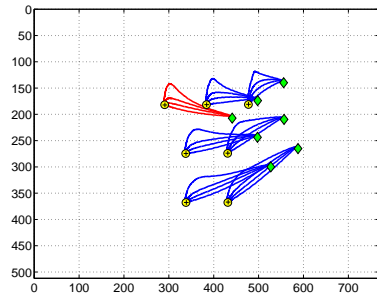
Les méthodes  $\mathbf{AV\ 2D}\frac{1}{2}$  et  $\mathbf{AV\ 2D}\frac{3}{4}$  sont, dans cette série d'expériences, les plus "robustes" aux erreurs de calibration de la caméra dans le sens où les trajectoires obtenues en présence d'erreurs s'éloignent très peu de celles obtenues en absence d'erreurs (voir les Figures 4.16(c) et 4.16(d)). Cela s'explique par le fait que, quand la caméra n'est pas très éloignée de sa position désirée, la matrice d'homographie est proche de l'identité ( $\mathbf{H} \approx \mathbf{I}$ ), et comme l'homographie estimée est  $\hat{\mathbf{H}} = \delta\mathbf{A}\mathbf{H}\delta\mathbf{A}^{-1} \approx \mathbf{I} \forall \delta\mathbf{A} \neq 0$ , ceci implique que les erreurs de calibration de la caméra n'ont que très peu d'effets en fin d'asservissement, puisqu'elles n'interviennent que sur une partie du jacobien de la tâche et sur une partie de la fonction de tâche (on rappelle que l'angle de rotation et le rapport entre les profondeurs ne sont pas influencés par les erreurs de calibration). Dans le cas de la méthode  $\mathbf{AV\ 2Dc}$ , les erreurs de calibration interviennent dans la matrice d'interaction constante de la même manière que pour des positions plus éloignées alors que, dans le cas de la méthode  $\mathbf{AV\ 2Dv}$ , ces erreurs sont couplées à l'état dans la matrice d'interaction, ce qui explique le comportement plus satisfaisant de l' $\mathbf{AV\ 2Dc}$  (voir Figures 4.16(a) et 4.16(b)). Ceci n'empêche pas la convergence mais seulement la manière dont le système arrive à la convergence.



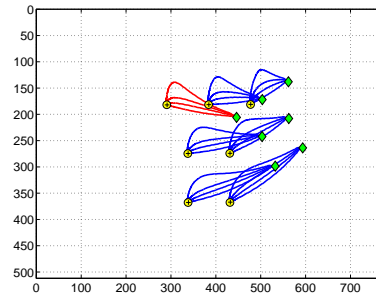
(a) AV 2Dc



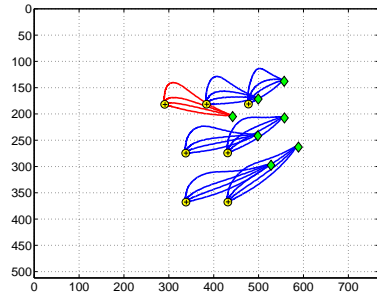
(b) AV 2Dv



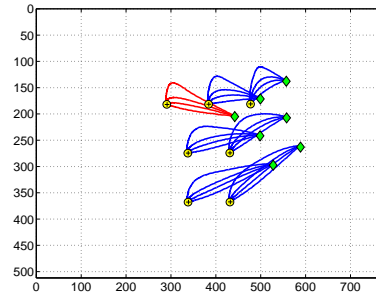
(c) AV 2D  $\frac{1}{2}$



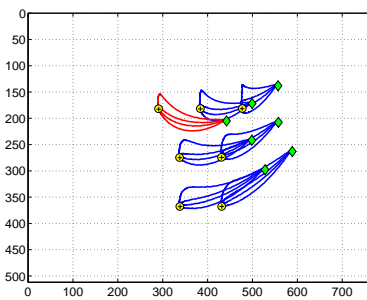
(d) AV 2D  $\frac{3}{4}$



(e) AV 3Dpr

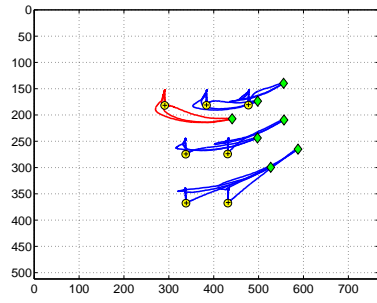


(f) AV 3Dtp

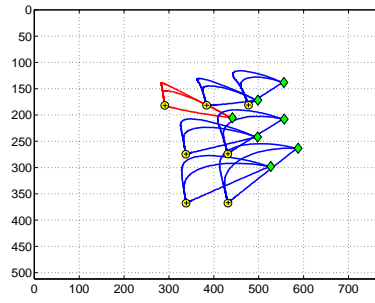


(g) AV 3Dtr

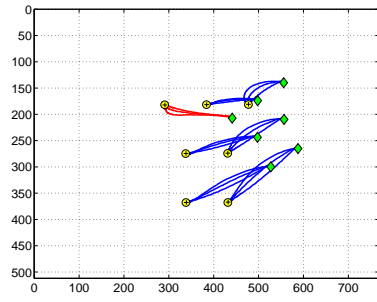
FIG. 4.15 – Position 1: Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration caméra/effecteur



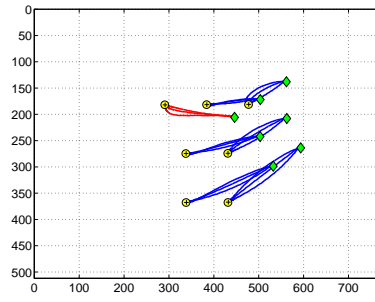
(a) AV 2Dc



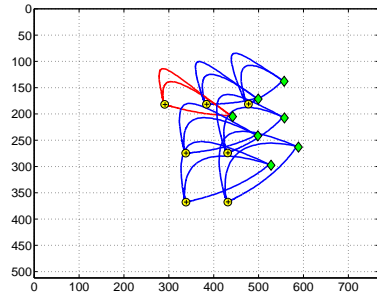
(b) AV 2Dv



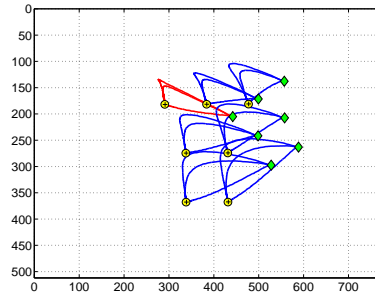
(c) AV 2D $\frac{1}{2}$



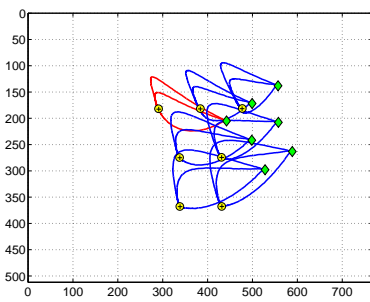
(d) AV 2D $\frac{3}{4}$



(e) AV 3Dpr

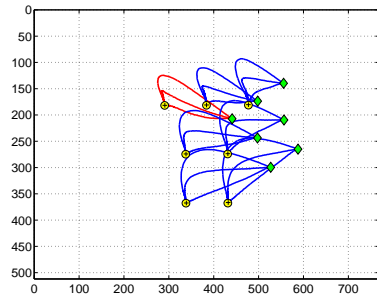


(f) AV 3Dtp

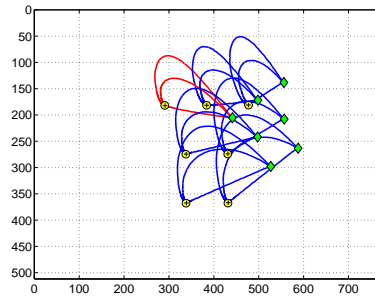


(g) AV 3Dtr

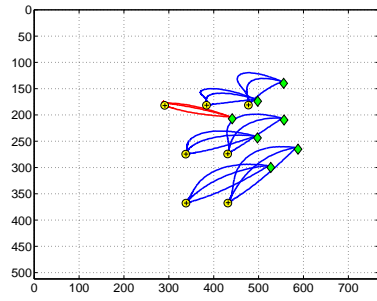
FIG. 4.16 – Position 1: Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration de la caméra



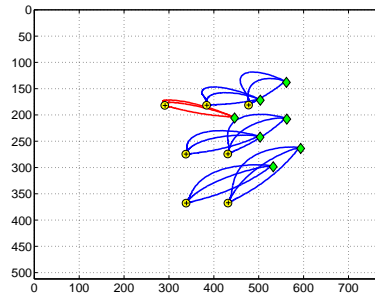
(a) AV 2Dc



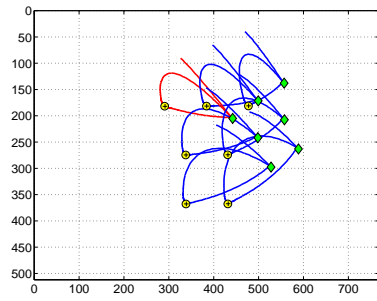
(b) AV 2Dv



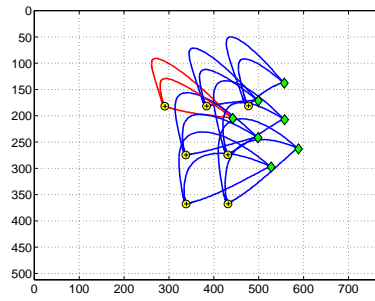
(c) AV 2D $\frac{1}{2}$



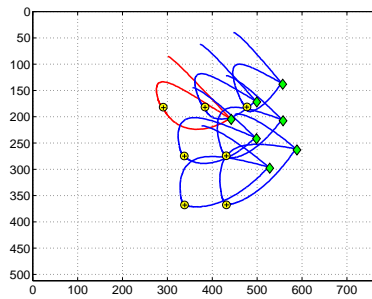
(d) AV 2D $\frac{3}{4}$



(e) AV 3Dpr



(f) AV 3Dtp



(g) AV 3Dtr

FIG. 4.17 – Position 1: Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration caméra et caméra/effecteur

POS 2	AV 2Dc	AV 2Dv	AV 2D $\frac{1}{2}$	AV 2D $\frac{3}{4}$	AV 2Dpr	AV 3Dtp	AV 3Dtr
c00-e00	*	$\frac{962}{0.13}$	$\frac{994}{0.14}$	$\frac{1026}{0.15}$	$\frac{988}{0.15}$	$\frac{761}{0.19}$	$\frac{1316}{0.13}$
c00-e05	*	$\frac{1536}{0.13}$	$\frac{1152}{0.15}$	$\frac{1120}{0.16}$	$\frac{1385}{0.16}$	$\frac{941}{0.19}$	$\frac{1343}{0.13}$
c00-e10	*	$\frac{1624}{0.13}$	$\frac{1089}{0.16}$	*	*	$\frac{1003}{0.19}$	$\frac{1386}{0.14}$
c00-e20	*	*	*	*	*	*	*
c30-e00	*	$\frac{1772}{0.13}$	$\frac{914}{0.15}$	$\frac{857}{0.16}$	$\frac{1153}{0.16}$	$\frac{1174}{0.19}$	$\frac{1190}{0.16}$
c50-e00	*	$\frac{2110}{0.13}$	$\frac{971}{0.15}$	$\frac{845}{0.16}$	*	$\frac{1510}{0.19}$	*
c30-e10	*	*	$\frac{1035}{0.17}$	$\frac{1026}{0.17}$	*	*	*
c50-e20	*	*	$\frac{1626}{0.17}$	$\frac{1341}{0.17}$	*	*	*

TAB. 4.3 – *Comparaison des méthodes pour la position 2*

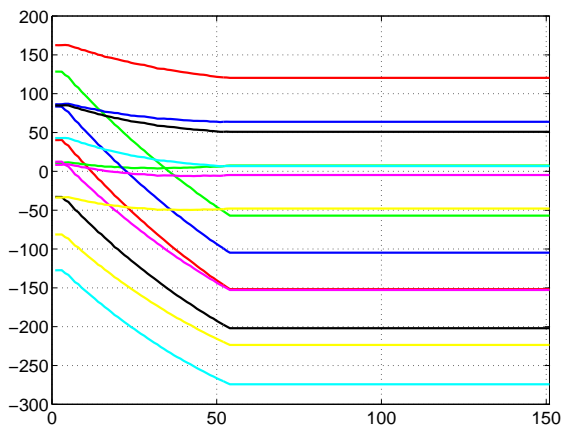
En considérant les résultats obtenus pour une position initiale de la caméra plus éloignée (voir le tableau 4.3), on remarque que l'**AV 2Dc** ne converge pas à partir de la position 2 car, comme indiqué en [Chaumette 98], le grand déplacement en rotation autour de l'axe  $\vec{z}$  provoque un mouvement de zoom avant de la caméra, et les points de la cible sortent de l'image (voir la Figure 4.22(a)). Cela est dû à un couplage important entre la vitesse de rotation autour de  $\vec{z}$  et de translation en  $\vec{z}$  causé par le choix des informations visuelles employées (les coordonnées des points).

Nous pouvons remarquer encore une fois que les méthodes **AV 2D $\frac{1}{2}$**  et **AV 2D $\frac{3}{4}$**  sont globalement les plus robustes aux erreurs de calibration (puisque les seules à converger quand d'importantes erreurs de calibration sont introduites). Comme la distance initiale  $Z(0)$  est très différente de la distance désirée  $Z^*$ , les deux schémas d'asservissement ont ici un comportement assez différent.

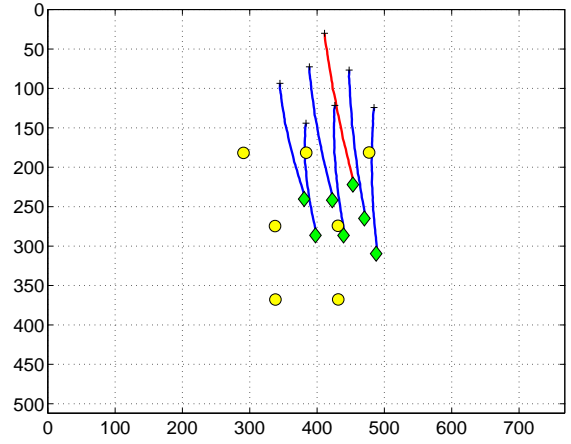
Comme toutes les autres méthodes, quand la calibration du système est vraiment mauvaise, nos schémas d'asservissement peuvent ne pas converger. On peut par exemple le voir sur la Figure 4.18, qui correspond à l'expérience "c00-e20". La sortie de la cible de l'image est provoquée par un dépassement de la coordonnée  $y$  bien que l'erreur en rotation et sur la distance relative diminuent. On verra par la suite qu'une commande robuste nous permettra d'éliminer ces cas de non convergence en conservant la cible dans l'image.

Encore une fois, les erreurs de calibration peuvent parfois se compenser, ce qui explique pourquoi nos méthodes d'asservissement convergent dans l'expérience "c50-e20" alors qu'elles ne convergent pas dans l'expérience "c00-e20". En effet, comme on peut le voir sur les Figures 4.19(b) et 4.19(c), les erreurs de calibration sur la matrice de passage caméra/effecteur ont tendance dans le cas présent à entraîner un déplacement vers le haut de l'image, alors que, comme on peut le voir sur les Figures 4.20(b) et 4.20(c), en présence d'erreurs de calibration de la caméra, la cible a plutôt tendance à se déplacer vers le bas de l'image. Par conséquent, nous pouvons voir sur les Figures 4.21(b) et 4.21(c) que les erreurs de calibration se compensent. Ce phénomène, imprévisible a priori, n'est pas propre à nos schémas d'asservissement et peut se présenter également pour les autres méthodes comme on le verra dans la prochaine expérience.

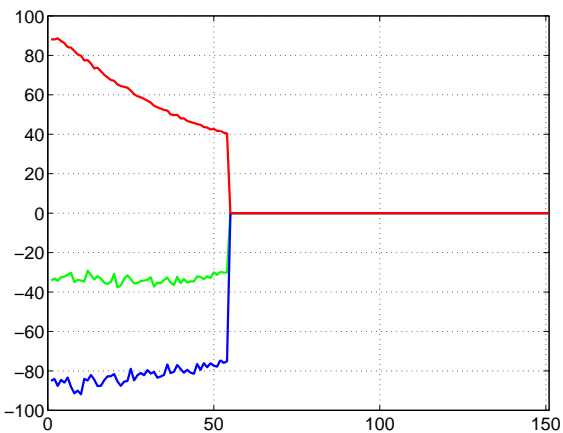




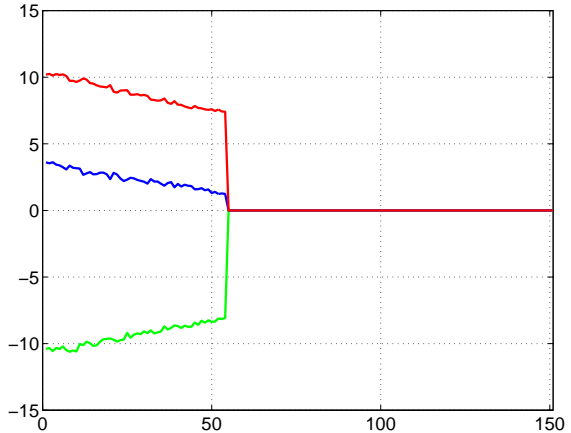
(a) erreur sur les coordonnées des points



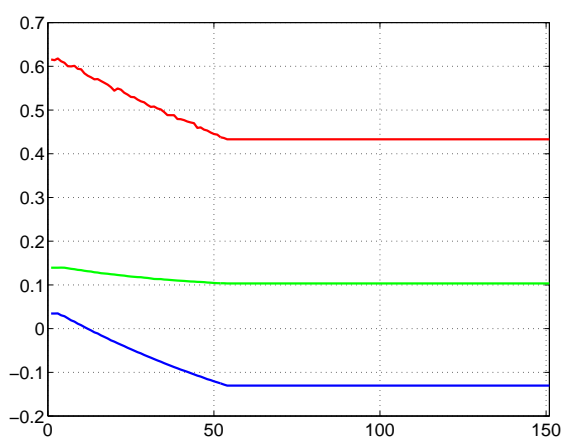
(b) trajectoire des points dans l'image



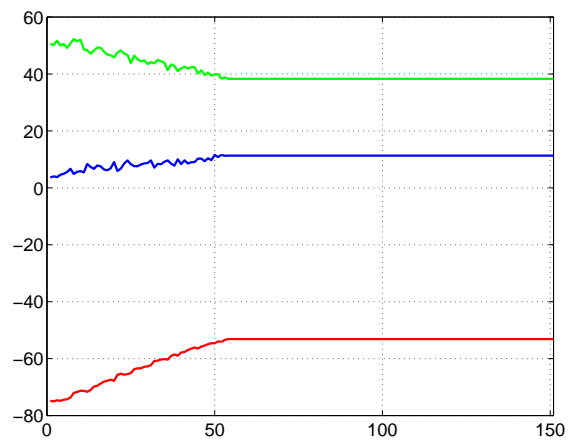
(c) vitesse de translation (cm/s)



(d) vitesse de rotation (deg/s)

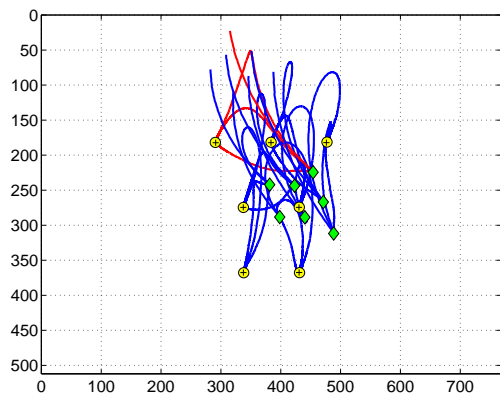


(e) coordonnées image étendues

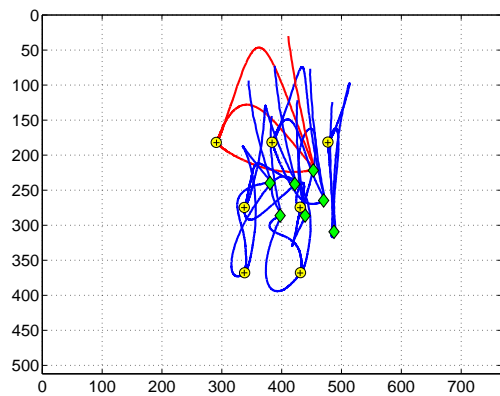


(f) rotation de la caméra (deg)

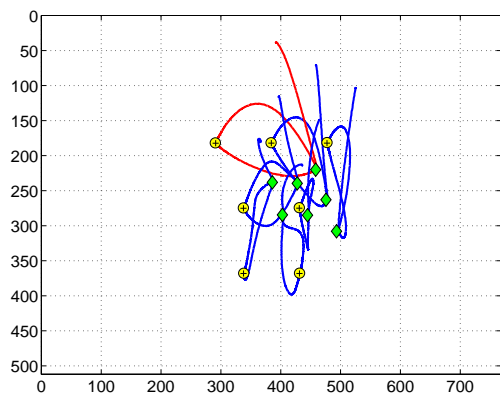
FIG. 4.18 – Position 2: résultats de l'AV 2D $\frac{1}{2}$  pour l'expérience "c00-e20"



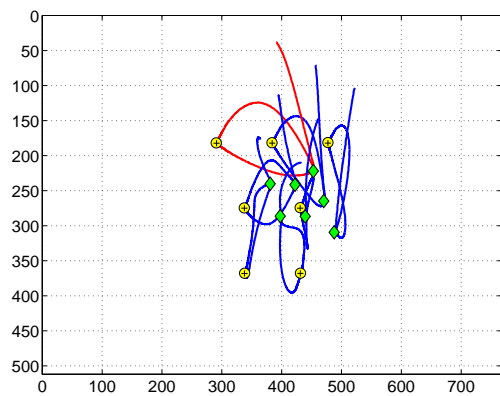
(a) AV 2Dv



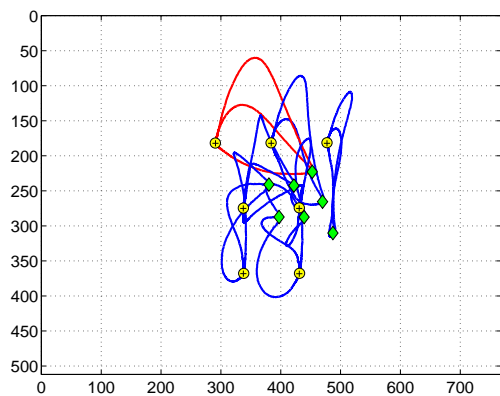
(b) AV 2D  $\frac{1}{2}$



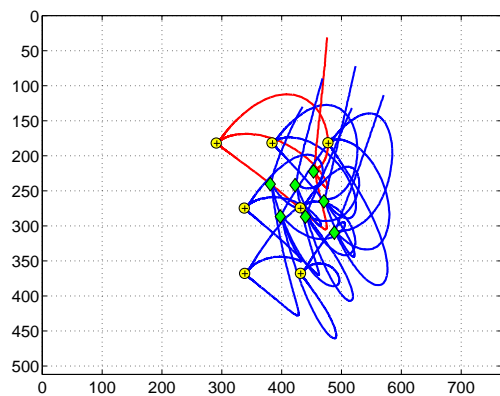
(c) AV 2D  $\frac{3}{4}$



(d) AV 3Dpr

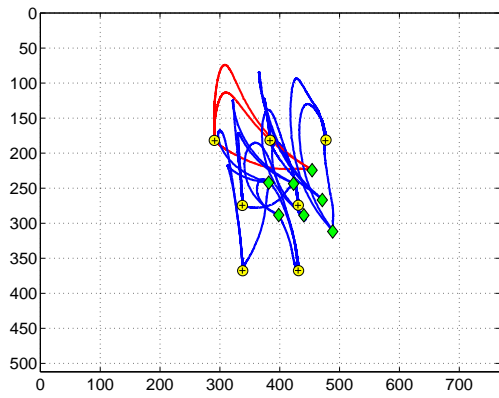


(e) AV 3Dtp

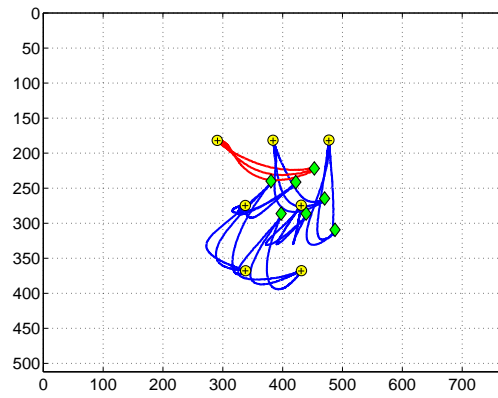


(f) AV 3Dtr

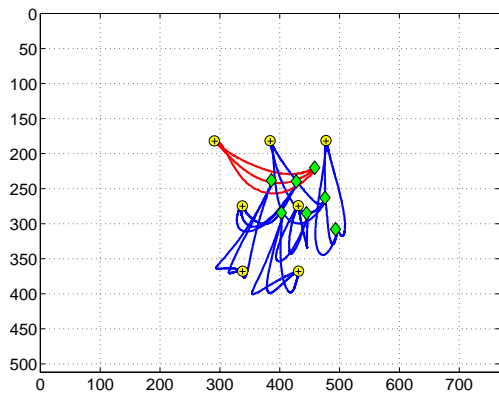
FIG. 4.19 – Position 2: Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration caméra/effecteur



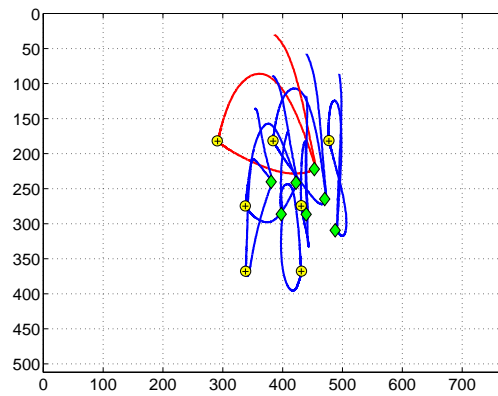
(a) AV 2D $v$



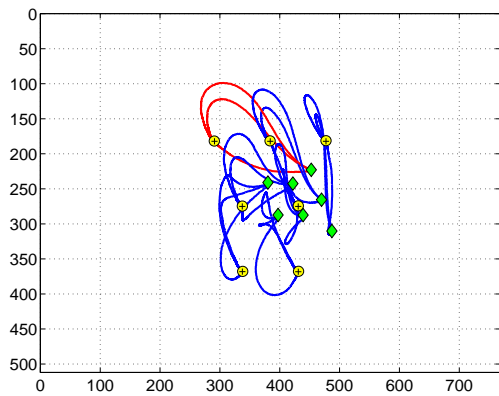
(b) AV 2D  $\frac{1}{2}$



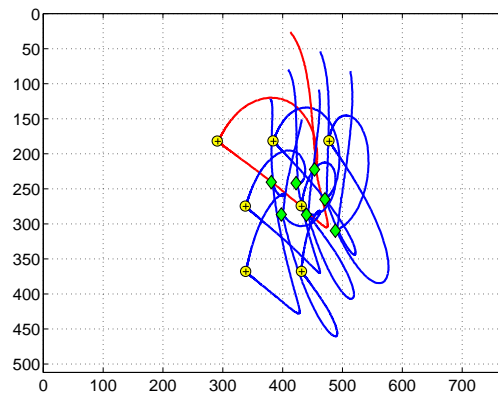
(c) AV 2D  $\frac{3}{4}$



(d) AV 3D $pr$

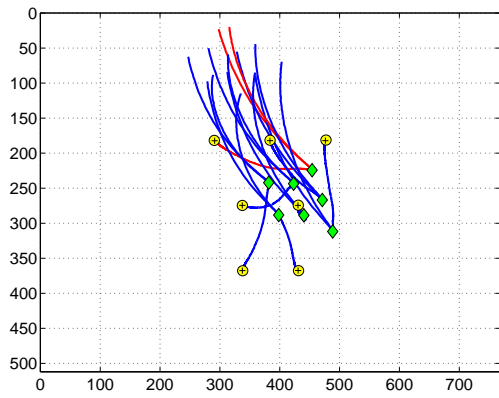


(e) AV 3D $tp$

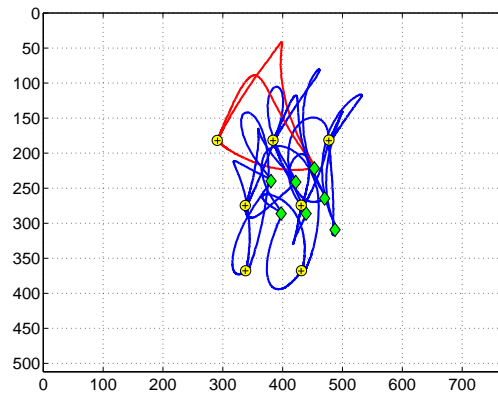


(f) AV 3D $tr$

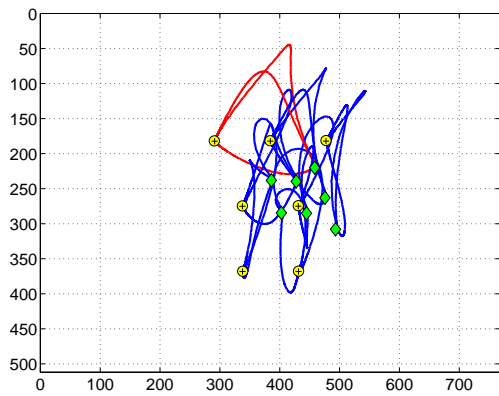
FIG. 4.20 – *Position 2: Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration caméra*



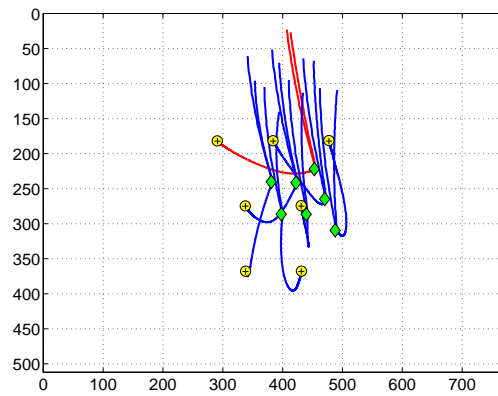
(a) AV 2D $v$



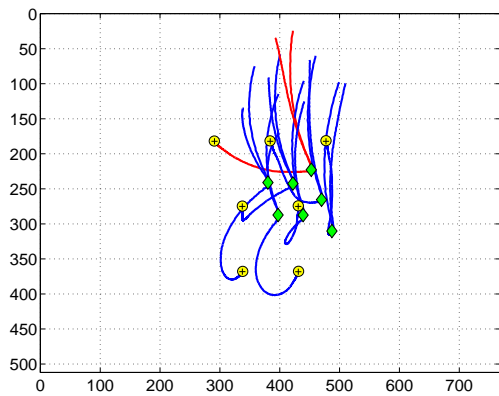
(b) AV 2D $\frac{1}{2}$



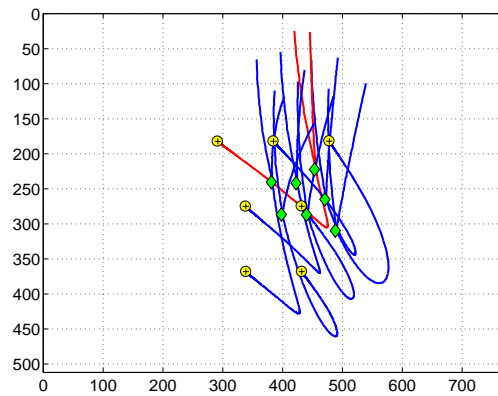
(c) AV 2D $\frac{3}{4}$



(d) AV 3D $pr$



(e) AV 3D $tp$



(f) AV 3D $tr$

FIG. 4.21 – Position 2: Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration caméra et caméra/effecteur

POS 3	AV 2Dc	AV 2Dv	AV 2D $\frac{1}{2}$	AV 2D $\frac{3}{4}$	AV 3Dpr	AV 3Dtp	AV 3Dtr
c00-e00	*	*	$\frac{1522}{0.10}$	$\frac{1537}{0.10}$	$\frac{1200}{0.10}$	$\frac{1473}{0.10}$	*
c00-e05	*	*	$\frac{1668}{0.10}$	$\frac{1669}{0.10}$	$\frac{1704}{0.10}$	$\frac{1750}{0.10}$	*
c00-e10	*	*	*	*	*	*	*
c00-e20	*	*	*	*	*	*	*
c30-e00	*	*	$\frac{1388}{0.10}$	$\frac{1384}{0.10}$	*	*	*
c50-e00	*	*	$\frac{1451}{0.10}$	$\frac{1443}{0.10}$	*	$\frac{1717}{0.10}$	*
c30-e10	*	*	$\frac{1406}{0.10}$	$\frac{1379}{0.10}$	$\frac{2014}{0.10}$	*	*
c50-e20	*	*	*	*	$\frac{2589}{0.10}$	$\frac{3800}{0.10}$	*

TAB. 4.4 – *Comparaison des méthodes pour la position 3*

Nous présentons à présent les résultats obtenus en partant de la position 3 sur le tableau 4.4. Nous pouvons remarquer que les asservissements visuels **AV 2Dc**, **AV 2Dv** et **AV 3Dtr** ne permettent pas d'atteindre la convergence même si la calibration du système est parfaite (il faut souligner que la position est extrêmement difficile) comme on peut le voir sur la Figure 4.22.

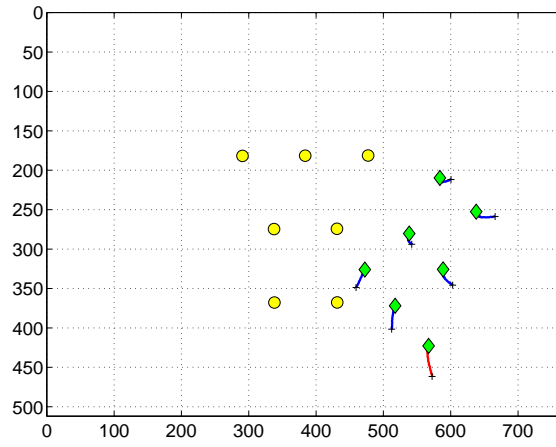
Dans le cas de l'asservissement visuel **AV 2Dv**, comme on l'a déjà vu pour l'asservissement visuel **AV 2Dc**, les problèmes de convergence sont ici liés au choix de l'information visuelle utilisée dans l'asservissement [Chaumette 98]. Dans le cas de l'**AV 2Dv**, le couplage entre la vitesse en rotation et de translation en  $\vec{z}$  provoque un mouvement de zoom arrière de la caméra, et la cible devient tellement petite dans l'image que l'asservissement doit être arrêté (si le robot n'a pas rejoint auparavant ses butées articulaires, voir Figure 4.22(b)).

Dans le cas de l'asservissement visuel **AV 3Dtr**, le problème est que, comme nous l'avons déjà dit à plusieurs reprises, l'asservissement n'assure aucun contrôle dans l'image (voir la Figure 4.22(c)).

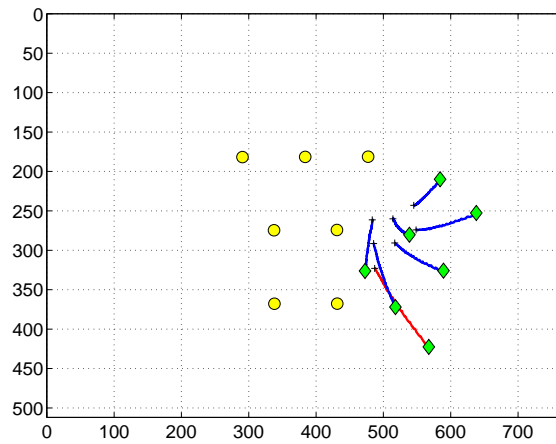
On remarque à nouveau que les erreurs de calibration peuvent se compenser comme le démontre le fait que la méthode **AV 3Dpr** ne converge pas pour "c50-e00" et "c00-e20" mais converge pour "c50-e20", ou que l'**AV 3Dtp** ne converge pas pour "c00-e20" mais converge pour "c50-e20". En effet, on peut voir cette fois sur les Figures 4.23(c) et 4.23(d) que, en présence d'erreurs de calibration sur la matrice de passage caméra/effecteur, la cible a plutôt tendance à se déplacer vers le bas de l'image, alors que, comme on peut le voir sur les Figures 4.24(b) et 4.24(c), en présence d'erreurs de calibration de la caméra la cible a plutôt tendance à se déplacer vers le haut de l'image. Par conséquent, les erreurs de calibration se compensent et nous pouvons observer sur les Figures 4.25(b) et 4.25(c) la convergence des méthodes considérées.

Vu les trajectoires dans l'image (Figures 4.23, 4.24 et 4.25), et les résultats obtenus auparavant, on ne peut que conclure que les erreurs de calibration ont un effet d'autant plus néfaste que la position initiale de la caméra est éloignée de sa position désirée. Pour remédier à ce problème, il serait intéressant de générer des trajectoires telles que la position initiale soit toujours proche de sa position désirée.

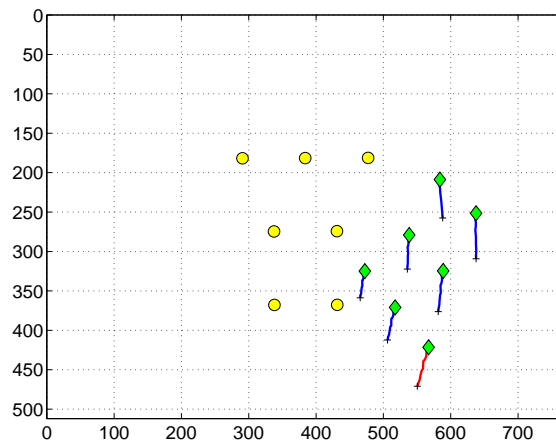
On peut finalement noter que l'**AV 2D $\frac{1}{2}$**  et l'**AV 2D $\frac{3}{4}$**  ont ici un comportement presque identique car la distance initiale  $Z(0)$  est très proche de la distance désirée  $Z^*$ .



(a) AV 2Dc

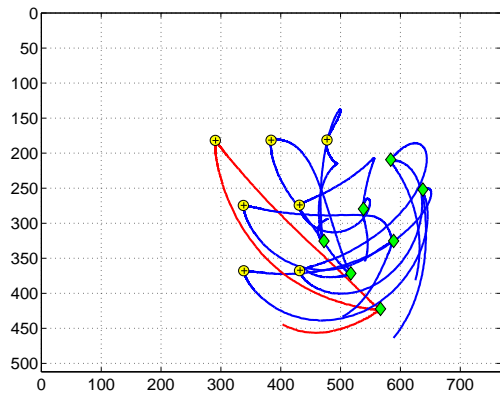


(b) AV 2Dv

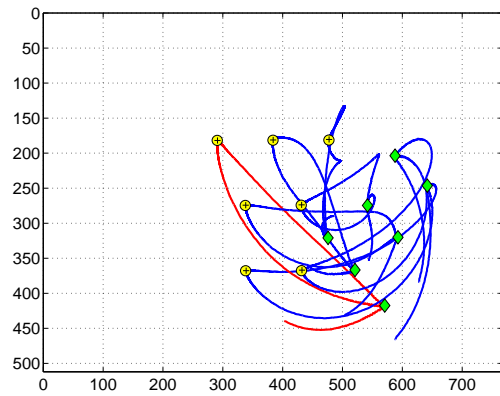


(c) AV 3Dtr

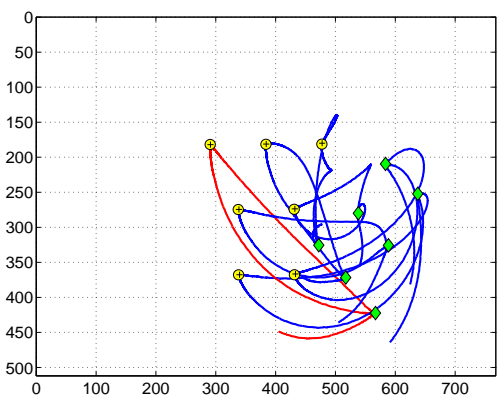
FIG. 4.22 – Position 3: problèmes de convergence indépendants de la calibration du système



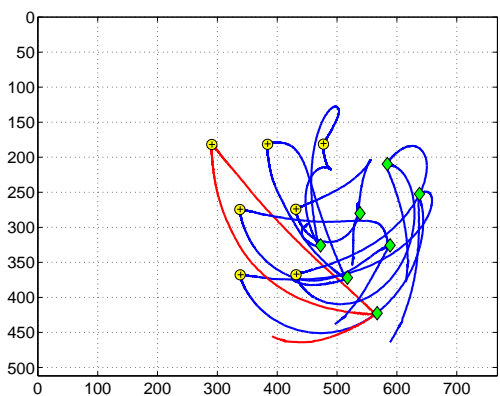
(a) AV  $2D \frac{1}{2}$



(b) AV  $2D \frac{3}{4}$



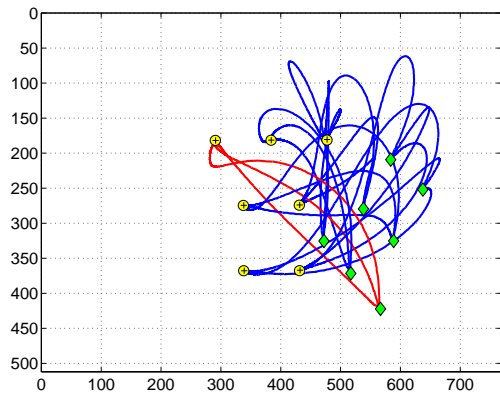
(c) AV 3Dpr



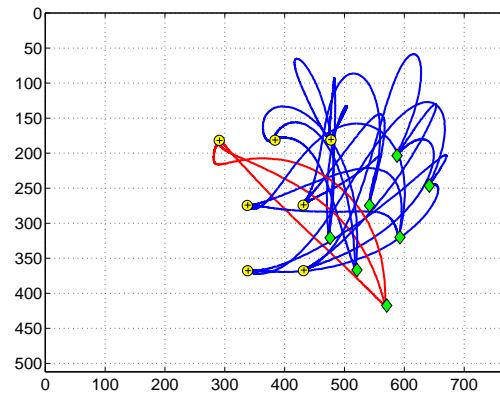
(d) AV 3Dtp

FIG. 4.23 – Position 3 : Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration caméra/effecteur

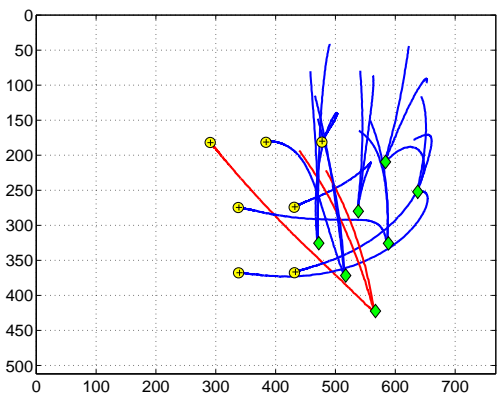




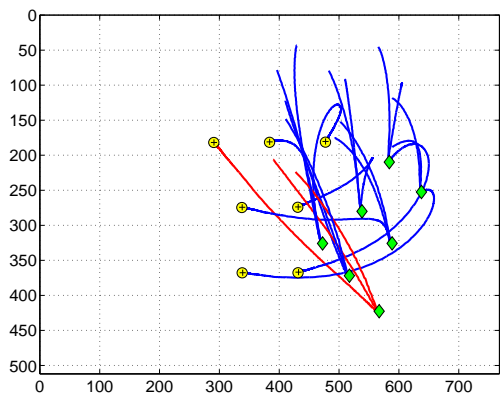
(a) AV  $2D \frac{1}{2}$



(b) AV  $2D \frac{3}{4}$

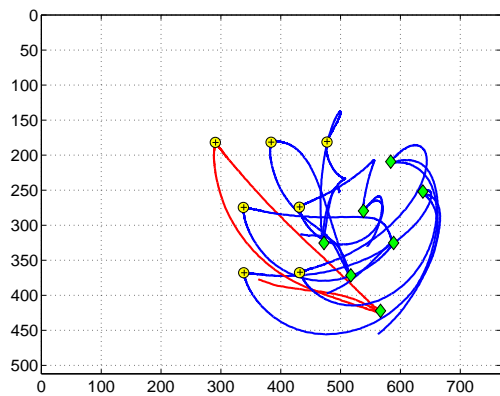


(c) AV 3Dpr

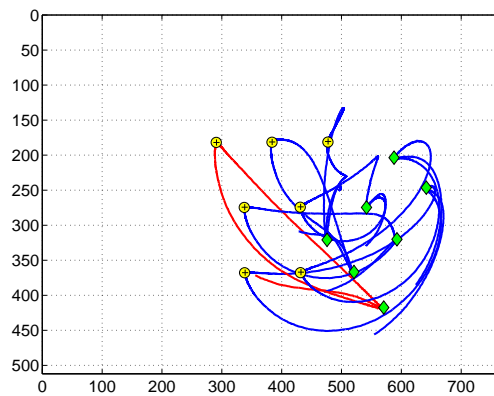


(d) AV 3Dtp

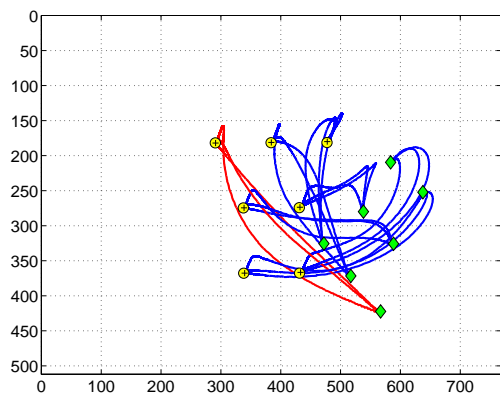
FIG. 4.24 – *Position 3*: Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration caméra



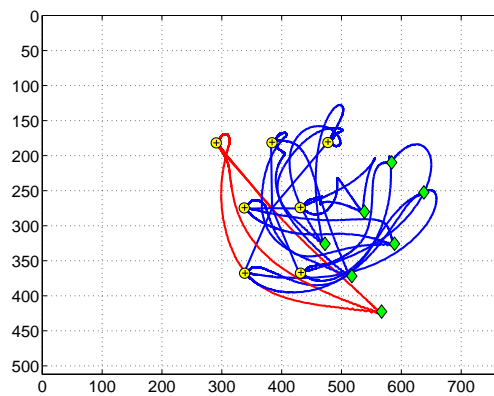
(a) AV  $2D \frac{1}{2}$



(b) AV  $2D \frac{3}{4}$



(c) AV  $3D_{pr}$



(d) AV  $3D_{tp}$

FIG. 4.25 – *Position 3*: Trajectoires dans l'image pour les différentes méthodes en fonction des erreurs de calibration caméra et caméra/effecteur

De manière plus générale, on peut synthétiser les résultats des trois séries d'expériences de la manière suivante :

- la différence de temps de parcours entre les différentes méthodes est significative pour des positions initiales pas trop éloignées de la position désirée et globalement est à l'avantage de l'asservissement visuel 2D.
- les erreurs de calibration caméra/effecteur interviennent de la même manière pour toutes les méthodes qui ont un contrôle dans l'image.
- la différence de comportement se remarque surtout avec les erreurs sur la calibration de la caméra. De plus, les systèmes sont d'autant plus robustes aux erreurs de calibration qu'ils sont basés sur un contrôle important dans l'image.
- les erreurs de calibration caméra et caméra/effecteur peuvent parfois se compenser.
- si la calibration est parfaite, les méthodes **AV 2D $\frac{1}{2}$** , **AV 2D $\frac{3}{4}$**  et **AV 3Dpr** ont un comportement très similaire. Toutefois, comme l'**AV 3Dpr** est réalisé à partir d'un calcul de pose, il est plus sensible aux erreurs de calibration.
- la méthode **AV 3Dtp** présente une plus grande robustesse aux erreurs de calibration que les autres méthodes 3D. Ceci est dû au fait que nous y avons introduit un contrôle direct dans l'image.
- les méthodes **AV 2D $\frac{1}{2}$**  et **AV 2D $\frac{3}{4}$**  donnent les meilleurs résultats en ce qui concerne la convergence à partir de positions initiales très éloignées.

## 4.5 Commande robuste

Nous allons voir que les cas où les asservissements visuels  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  ne convergeaient pas précédemment à cause d'une mauvaise calibration du système peuvent être éliminés par une commande robuste. La première expérience que nous présentons correspond à l'expérience "c00-e20" à partir de la position 2 montrée dans la Figure 4.18, alors que la deuxième correspond à l'expérience "c50-e20" à partir de la position 3.

Les résultats obtenus en utilisant l'AV  $2D\frac{1}{2}$  sont illustrés sur la Figure 4.26. Au tout début, les gains sont égaux à 1 car la cible est au centre de l'image. Quand l'asservissement commence, la cible se déplace rapidement vers le haut de l'image. Le gain  $f_v$  diminue alors pendant les 50 premières itérations. La vitesse de rotation diminue donc, et l'erreur de poursuite qui menait la cible hors de l'image s'arrête.

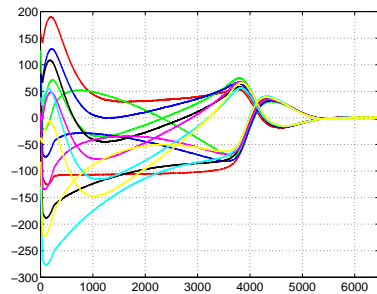
La cible commence alors à glisser vers la gauche (car la boucle de commande en  $u$  est stable), et ensuite, à revenir vers le centre. Le gain augmente progressivement vers 1 (itération 4000). A ce moment, les deux points en bas de l'image, qui ont dépassé leur positions désirées car le système n'est pas parfaitement calibré, se dirigent vers le bas de l'image. Le gain  $f_v$  commence à nouveau à diminuer jusqu'à environ 0.6. Entre temps, la rotation a continué sa décroissance vers zéro. Par conséquent, l'erreur de poursuite diminue et le gain peut croître vers 1 jusqu'à la convergence.

Dans la deuxième expérience, dont les résultats sont illustrés sur la Figure 4.27, nous utilisons l'AV  $2D\frac{3}{4}$  et nous montrons les limites de notre méthode en ce qui concerne le réglage des gains utilisés.

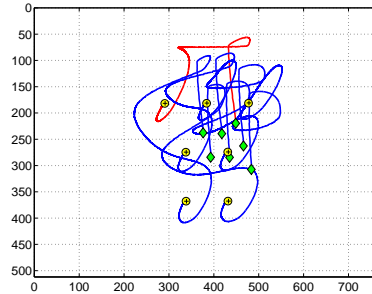
Au tout début, les gains valent  $f_u = 0.6$  et  $f_v = 0.5$  car la cible est au bord de l'image. Le gain  $f_u$  croît rapidement vers 1 car la cible se déplace horizontalement vers le centre de l'image. On peut remarquer des oscillations initiales provoquées par le fait que les erreurs de calibration poussent la cible vers l'extérieur mais, dans ce cas, le gain diminue et la cible revient vers le centre de l'image, ce qui provoque une nouvelle augmentation des gains. Ces oscillations peuvent être réduites en choisissant une décroissance moins rapide des gains mais elle ne peuvent pas être entièrement éliminées car elle sont également liées au retard introduit par le système de vision dans la boucle de commande.

Le point de la cible en bas de l'image se déplace d'abord vers la gauche et, quand il commence à remonter vers sa position désirée (itération 1400), le mouvement de rotation autour de  $\vec{z}$  amène un autre point vers le bas de l'image. Ceci provoque une petite diminution du gain (mais assez brusque), ce qui explique la diminution de la vitesse de la caméra. Ensuite le gain croît de manière continue jusqu'à la convergence.

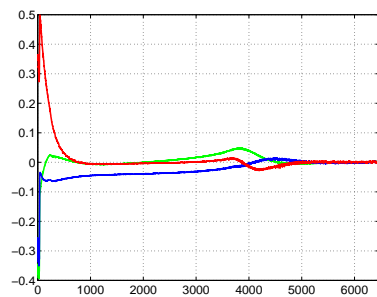
Il faut remarquer que la convergence est moins longue que dans l'expérience précédente (même si la position de départ est ici plus éloignée). En effet, afin d'augmenter la vitesse de convergence en fin d'asservissement, nous avons également augmenté le gain  $\lambda$ .



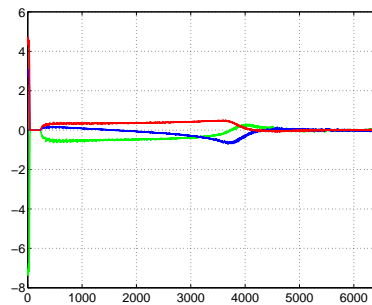
(a) erreur sur les coordonnées des points



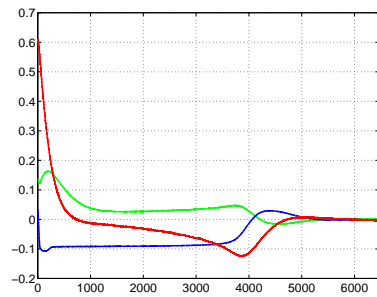
(b) trajectoire des points dans l'image



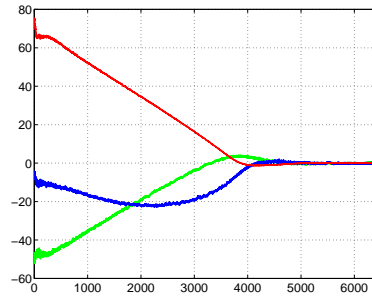
(c) vitesse de translation (cm/s)



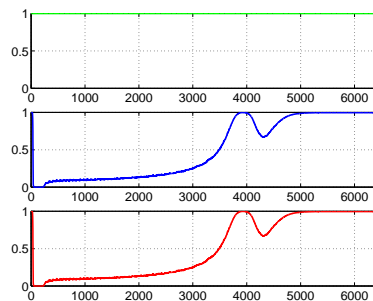
(d) vitesse de rotation (deg/s)



(e) coordonnées image étendues

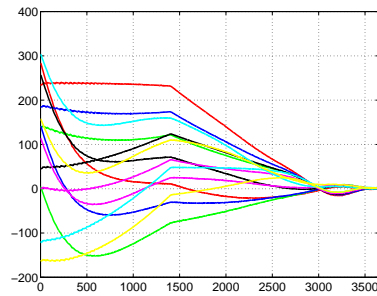


(f) rotation de la camera (deg)

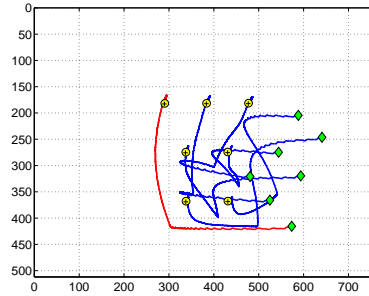


(g) gains  $f_u$ ,  $f_v$  et  $f_u f_v$

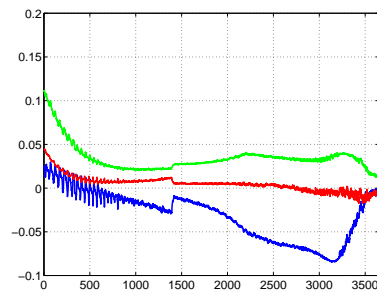
FIG. 4.26 – Position 2: commande robuste pour l'AV 2D $\frac{1}{2}$



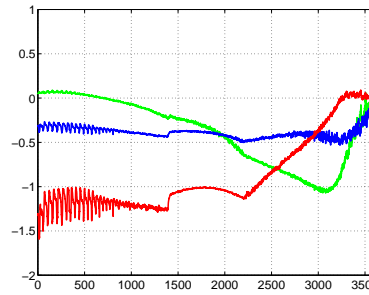
(a) erreur sur les coordonnées des points



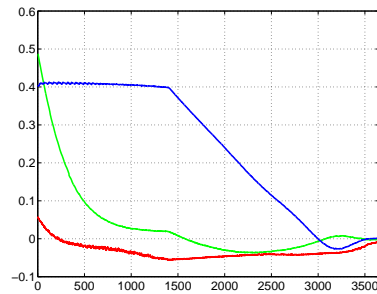
(b) trajectoire des points dans l'image



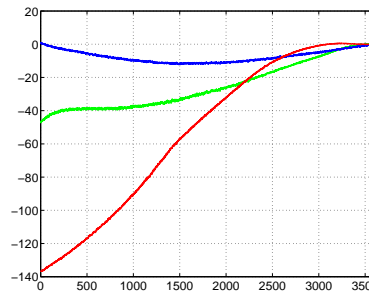
(c) vitesse de translation (cm/s)



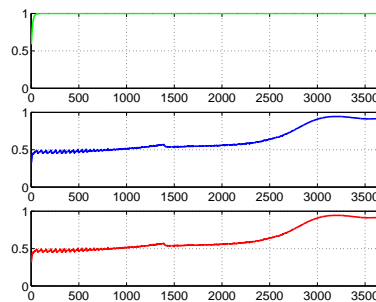
(d) vitesse de rotation (deg/s)



(e) coordonnées cartésiennes réduites



(f) rotation de la camera (deg)



(g) gains  $f_u$ ,  $f_v$  et  $f_u f_v$

FIG. 4.27 – Position 3: commande robuste pour l'AV 2D $\frac{3}{4}$

## 4.6 Sensibilité des méthodes d'asservissement visuel au bruit de mesure

Afin de tester la sensibilité des lois de commande aux erreurs de mesure, nous étudions le comportement des différents systèmes à la convergence. On peut signaler que nous avons utilisé une calibration aussi précise que possible afin d'obtenir des résultats indépendants des erreurs de calibration.

Nous avons comparé les méthodes suivantes : l'**AV 2Dc**, l'**AV 2D $\frac{1}{2}$** , l'**AV 3Dtr** et enfin l'**AV 3Dtp**. Nous n'avons pas jugé utile de tester l'**AV 2Dv**, l'**AV 2D $\frac{3}{4}$**  et l'**AV 3Dpr**. En effet, à la convergence, l'**AV 3Dtr** et l'**AV 3Dpr** ont exactement le même comportement. Il en est de même de l'**AV 2Dv** et de l'**AV 2Dc**, aux erreurs introduites dans la matrice d'interaction près. On rappelle que, pour l'**AV 2Dc**, la matrice d'interaction est calculée une fois pour toutes lors de l'apprentissage de l'image désirée, alors que pour l'**AV 2Dv**, ce calcul est effectué à chaque itération de la loi de commande. Comme la matrice d'interaction dépend des informations visuelles 2D et de l'estimation, par un calcul de pose, de la profondeur des points considérés, on peut en déduire que l'**AV 2Dv** sera légèrement plus instable à la convergence que l'**AV 2Dc** si les informations visuelles sont bruitées et si le calcul de pose ne fournit pas une estimation parfaite. De même, l'**AV 2D $\frac{1}{2}$**  et l'**AV 2D $\frac{3}{4}$**  ont le même comportement à la convergence, à la différence près que, pour le contrôle de la position de la caméra, on utilise des informations 2D en **AV 2D $\frac{1}{2}$**  alors que l'on utilise des informations issues de la reconstruction du déplacement en **AV 2D $\frac{3}{4}$** . L'**AV 2D $\frac{1}{2}$**  sera donc légèrement plus stable que l'**AV 2D $\frac{3}{4}$**  si l'estimation du déplacement n'est pas parfaite.

En ce qui concerne les asservissements basés sur un calcul de pose, à savoir l'**AV 3Dtr** et l'**AV 3Dtp**, nous avons comparé les résultats obtenus à partir de la méthode de Lagrange (méthode purement linéaire), la méthode de Dementhon (méthode linéaire itérative), et la méthode de Lowe (méthode non linéaire). En raison de l'architecture employée (à savoir que l'acquisition et le traitement des images sont effectués sur la carte IA 2000, en parallèle du calcul de pose et du calcul de la commande (sur une station Ultra Sparc), la cadence pour toutes les méthodes est de 25 Hz (soit 40 ms pour une itération des lois de commande). En utilisant une architecture plus simple (à savoir l'ensemble des calculs réalisés sur la carte IA 2000), la cadence serait encore de 25 Hz, sauf en cas d'utilisation de la méthode de Lowe (qui nécessite l'inversion d'une matrice (6x6) à chaque étape de la minimisation non linéaire).

Grâce à la cadence égale pour toutes les méthodes, le gain  $\lambda$  de l'asservissement a été fixé à 0.4 pour toutes les expérimentations. En considérant l'amplitude des oscillations sur la vitesse calculée de la caméra en régime stationnaire (c'est-à-dire autour de la position de convergence), on peut ainsi déterminer quantitativement l'effet des erreurs de mesure sur la stabilité des diverses lois de commande. On peut signaler que ces erreurs de mesure sont très faibles (de l'ordre du dixième de pixel) en raison de la simplicité des images considérées.

Plusieurs configurations ont été testées, à savoir :

- le positionnement par rapport à un objet plan (en l’occurrence celui présenté sur la Figure 4.1), et tel que le plan image soit incliné d’environ 30 degrés par rapport au plan de l’objet à la position désirée ;
- le positionnement par rapport à ce même objet plan, mais tel que le plan image soit parallèle au plan de l’objet à la position désirée ;
- le positionnement par rapport à un objet 3D (en l’occurrence celui présenté sur la Figure 4.7)

Les résultats correspondants sont représentés dans les Tableaux 4.5, 4.6 et 4.7, où l’écart médian  $a(x)$  sur chaque composante de la vitesse de translation et de rotation (respectivement en mm/s et  $\frac{1}{10}$ deg/s) a été calculé sur les 100 dernières itérations (soit environ 4 secondes). Afin de rendre plus lisible les résultats donnés dans ces tableaux, nous avons entouré deux fois le résultat correspondant au meilleur résultat, entouré une fois les résultats très proches, et souligné les résultats qui n’en sont pas trop éloignés. Les courbes associées sont données sur les Figures 4.28 et 4.29 pour la première configuration testée, 4.30 et 4.31 pour la deuxième, et 4.32 et 4.33 pour la troisième.

#### 4.6.1 Résultats pour une cible plane inclinée

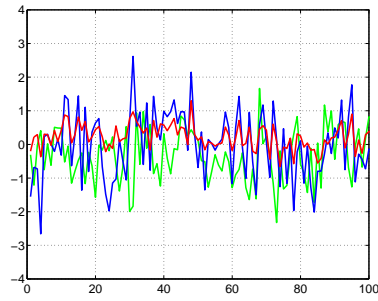
Pour cette première série d’expériences, et comme on pouvait s’y attendre, on peut noter que l’**AV 2Dc** est la méthode qui fournit les résultats les plus stables. Les asservissements **AV 3Dtp** et **AV 3Dtr** donnent également de bons résultats, sauf quand la méthode de Lagrange est employée pour le calcul de pose. On peut également noter que les résultats fournis par la méthode de Lowe sont plus robustes que ceux de la méthode de Dementhon, ce qui n’a rien de surprenant. De plus, comme l’objet est constitué d’un nombre de points assez important (7), on ne rencontre pas les problèmes d’instabilité du calcul de pose qui se produisent quand l’on considère 4 points. Finalement, l’**AV 2D $\frac{1}{2}$**  ne donne pas d’excellents résultats bien que l’estimation de l’homographie ait été effectuée avec la connaissance a priori que l’objet soit plan. En effet, à la convergence, il est impossible de distinguer entre les deux solutions possibles (et cependant très proches) pour le calcul du déplacement à partir de l’homographie. Une moins bonne robustesse de l’**AV 2D $\frac{1}{2}$**  par rapport à l’**AV 2D** est donc logique puisqu’une étape supplémentaire de calcul du déplacement de la caméra est introduite. Enfin, l’utilisation du modèle CAO pour le calcul de pose permet aux méthodes d’**AV 3D** de fournir des résultats plus robustes que quand l’on ne dispose pas de ce modèle CAO (ce qui est le cas de l’**AV 2D $\frac{1}{2}$** ).

De manière générale, on peut noter que les résultats sont nettement plus stables pour les composantes de translation en  $z$  et de rotation autour de  $z$ . Cela est dû à leur meilleur découplage par rapport aux composantes  $v_x$  (fortement couplée à  $w_y$ ) et  $v_y$  (fortement couplée à  $w_x$ ).

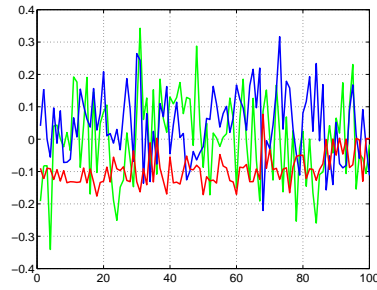


	AV 2Dc	AV 2D $\frac{1}{2}$	AV 3Dtr (Lag)	AV 3Dtr (Dem)	AV 3Dtr (Low)	AV 3Dtp (Lag)	AV 3Dtp (Dem)	AV 3Dtp (Low)
$a(v_x)$	<span style="border: 1px solid black; padding: 2px;">0.64</span>	4.08	3.89	<u>0.92</u>	<u>0.92</u>	3.73	<span style="border: 1px solid black; padding: 2px;">0.79</span>	<span style="border: 1px solid black; padding: 2px;">0.75</span>
$a(v_x)$	<span style="border: 1px solid black; padding: 2px;">0.79</span>	2.87	2.73	<span style="border: 1px solid black; padding: 2px;">0.85</span>	<span style="border: 1px solid black; padding: 2px;">0.73</span>	2.64	<u>1.07</u>	<span style="border: 1px solid black; padding: 2px;">0.74</span>
$a(v_z)$	<span style="border: 1px solid black; padding: 2px;">0.29</span>	1.08	0.89	<u>0.39</u>	<span style="border: 1px solid black; padding: 2px;">0.29</span>	0.90	<u>0.43</u>	<span style="border: 1px solid black; padding: 2px;">0.31</span>
$a(w_x)$	<span style="border: 1px solid black; padding: 2px;">0.10</span>	0.34	0.36	<span style="border: 1px solid black; padding: 2px;">0.11</span>	<span style="border: 1px solid black; padding: 2px;">0.10</span>	0.35	<u>0.14</u>	<span style="border: 1px solid black; padding: 2px;">0.10</span>
$a(w_y)$	<span style="border: 1px solid black; padding: 2px;">0.09</span>	0.51	0.52	<span style="border: 1px solid black; padding: 2px;">0.12</span>	<span style="border: 1px solid black; padding: 2px;">0.13</span>	0.47	<span style="border: 1px solid black; padding: 2px;">0.11</span>	<span style="border: 1px solid black; padding: 2px;">0.10</span>
$a(w_z)$	<span style="border: 1px solid black; padding: 2px;">0.04</span>	0.36	<u>0.05</u>	<u>0.05</u>	<u>0.05</u>	<span style="border: 1px solid black; padding: 2px;">0.04</span>	<span style="border: 1px solid black; padding: 2px;">0.04</span>	<u>0.05</u>

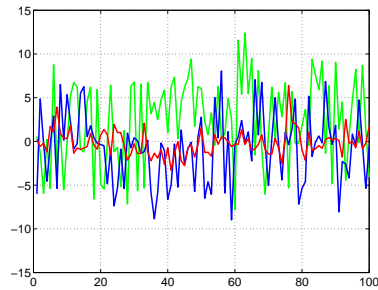
TABLE 4.5 – *Comparaison des méthodes pour un objet plan incliné*



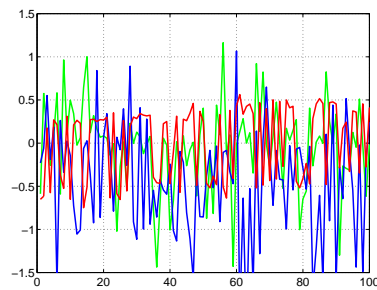
(a) vitesse de translation  
AV 2Dc



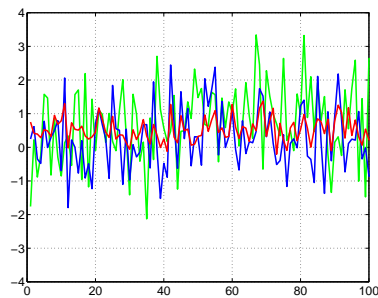
(b) vitesse de rotation  
AV 2Dc



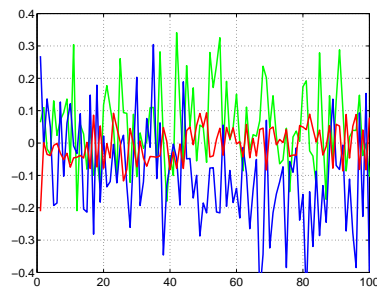
(c) vitesse de translation  
AV 2D $\frac{1}{2}$



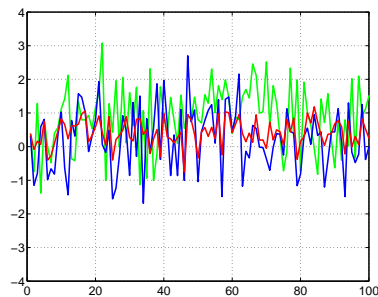
(d) vitesse de rotation  
AV 2D $\frac{1}{2}$



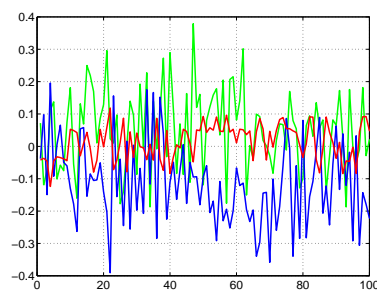
(e) vitesse de translation  
AV 3Dtr



(f) vitesse de rotation  
AV 3Dtr

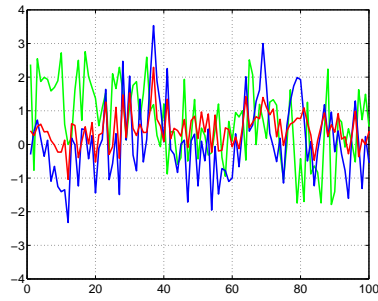


(g) vitesse de translation  
AV 3Dtp

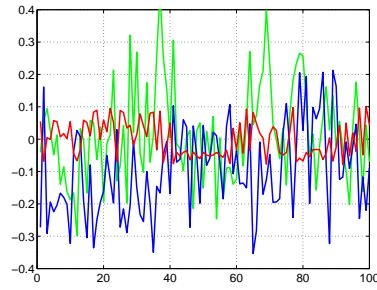


(h) vitesse de rotation  
AV 3Dtp

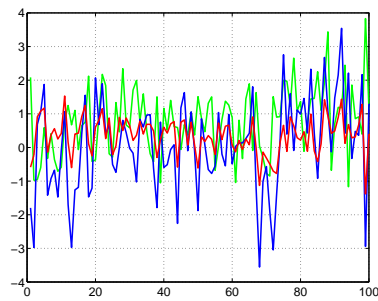
FIG. 4.28 – Niveau de bruit sur les lois de commande pour un objet plan incliné (I)



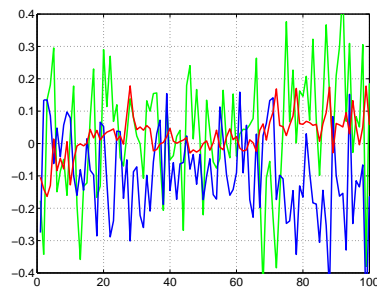
(a) vitesse de translation  
**AV 3Dtr (Dem)**



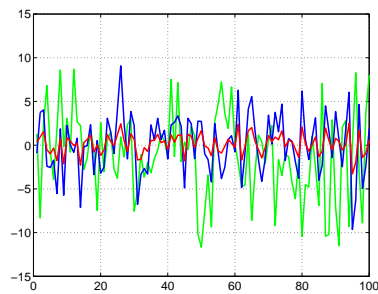
(b) vitesse de rotation  
**AV 3Dtr (Dem)**



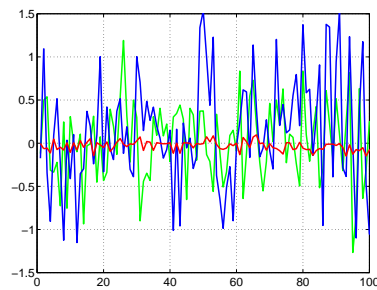
(c) vitesse de translation  
**AV 3Dtp (Dem)**



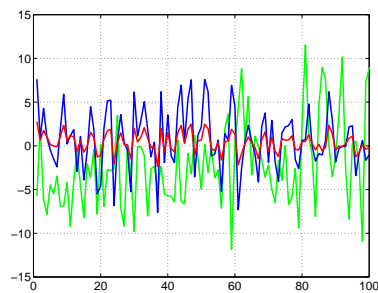
(d) vitesse de rotation  
**AV 3Dtp (Dem)**



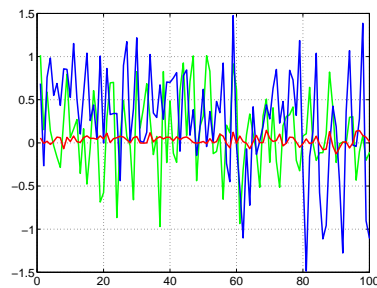
(e) vitesse de translation  
**AV 3Dtr (Lag)**



(f) vitesse de rotation  
**AV 3Dtr (Lag)**



(g) vitesse de translation  
**AV 3Dtp (Lag)**



(h) vitesse de rotation  
**AV 3Dtp (Lag)**

## 4.6.2 Résultats pour une cible parallèle au plan image

En considérant à présent le cas du positionnement de la caméra parallèlement à un objet plan (voir Tableau 4.6 et Figures 4.30 et 4.31), on peut effectuer les remarques suivantes :

Par rapport à la configuration précédente (positionnement par rapport à un objet plan incliné), l'ensemble des méthodes 2D et 3D fournissent des résultats beaucoup moins stables sur les composantes  $v_x, v_y, w_x$  et  $w_y$ .

D'un point de vue applicatif, il est regrettable que cette configuration soit moins stable que la configuration précédente, car le fait de considérer un objet incliné rend le domaine de visibilité de l'objet asymétrique à son plan de base.

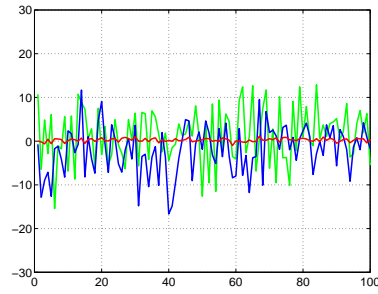
Ces résultats globalement moins satisfaisants sont dus, dans le cas de l'**AV 2D**, au conditionnement nettement moins bon de la matrice d'interaction [Hashimoto 98]. De même, pour l'**AV 3D**, le calcul de pose est alors connu pour être particulièrement instable. Cela est particulièrement vrai pour la méthode de Dementhon, puisqu'il est alors impossible de distinguer, parmi les deux solutions assez proches que cet algorithme fournit, la solution réelle (on aurait pu ajouter une contrainte de continuité sur la détermination de la pose, mais il aurait alors été impossible de converger à tout coup si la mauvaise solution avait été sélectionnée à la position d'apprentissage).

Par contre, l'**AV 2D** $\frac{1}{2}$  donne des résultats du même ordre de grandeur que dans le cas précédent. Cela est dû au fait que, comme l'on ne calcule que le déplacement de la caméra sans jamais remonter à la pose entre la caméra et l'objet, les cas objet plan incliné et objet plan parallèle au plan de l'image ne présentent aucune différence pour cette méthode.

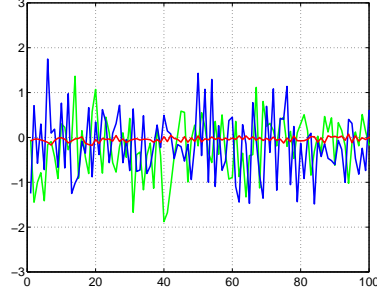
Pour ce qui est de la comparaison entre les différentes méthodes, on peut noter que les résultats d'**AV 3D** utilisant la méthode de Dementhon sont les plus médiocres. La méthode de Lagrange présente alors un intérêt en **AV 3D** si, pour des problèmes de temps de calcul, on ne peut pas utiliser la méthode non linéaire de Lowe. Finalement, il est assez difficile de distinguer, dans le cas présent, quelle est la méthode la plus robuste entre l'**AV 2Dc**, l'**AV 3Dtr** et l'**AV 3Dtp** si la méthode de Lowe est employée.

	AV 2Dc	AV 2D $\frac{1}{2}$	AV 3Dtr (Lag)	AV 3Dtr (Dem)	AV 3Dtr (Low)	AV 3Dtp (Lag)	AV 3Dtp (Dem)	AV 3Dtp (Low)
$a(v_x)$	<u>5.08</u>	<u>5.62</u>	<u>6.04</u>	16.15	<u>3.87</u>	<u>4.58</u>	7.29	<u>4.42</u>
$a(v_y)$	<u>4.30</u>	<u>4.39</u>	<u>5.71</u>	42.74	<u>5.72</u>	7.25	42.04	<u>6.11</u>
$a(v_z)$	<u>0.33</u>	<u>0.59</u>	<u>0.32</u>	2.15	<u>0.34</u>	<u>0.32</u>	2.73	<u>0.29</u>
$a(w_x)$	<u>0.49</u>	<u>0.48</u>	<u>0.65</u>	4.90	<u>0.65</u>	0.83	4.81	<u>0.70</u>
$a(w_y)$	<u>0.58</u>	<u>0.61</u>	<u>0.69</u>	1.86	<u>0.44</u>	<u>0.52</u>	0.84	<u>0.51</u>
$a(w_z)$	<u>0.04</u>	0.09	<u>0.05</u>	0.10	<u>0.04</u>	<u>0.06</u>	0.09	<u>0.05</u>

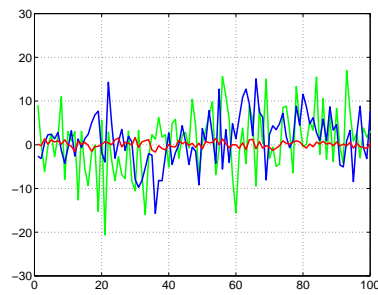
TAB. 4.6 – Comparaison des méthodes pour une cible parallèle au plan image



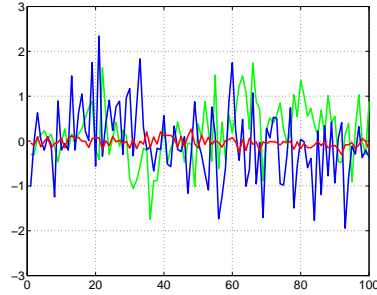
(a) vitesse de translation  
**AV 2Dc**



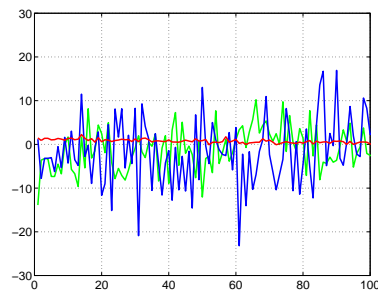
(b) vitesse de rotation  
**AV 2Dc**



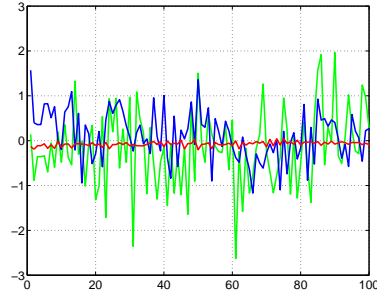
(c) vitesse de translation  
**AV 2D  $\frac{1}{2}$**



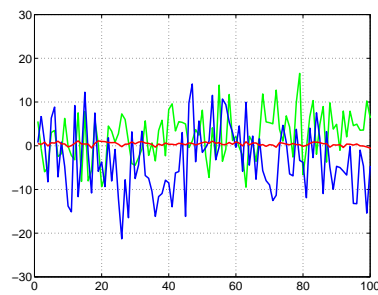
(d) vitesse de rotation  
**AV 2D  $\frac{1}{2}$**



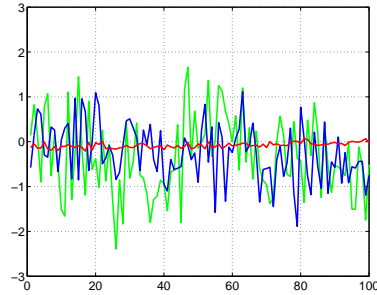
(e) vitesse de translation  
**AV 3Dtr (Low)**



(f) vitesse de rotation  
**AV 3Dtr (Low)**

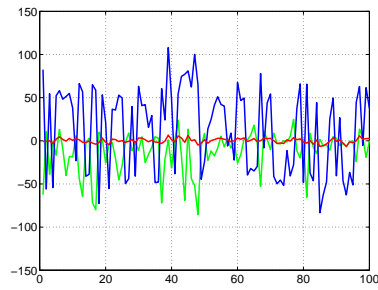


(g) vitesse de translation  
**AV 3Dtp (Low)**

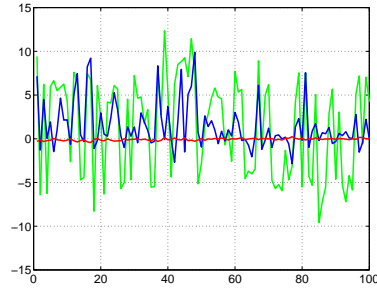


(h) vitesse de rotation  
**AV 3Dtp (Low)**

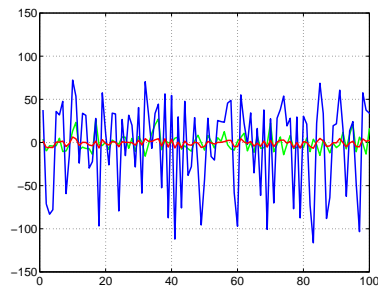
FIG. 4.30 – Niveau de bruit sur les lois de commande pour une cible parallèle au plan image (*I*)



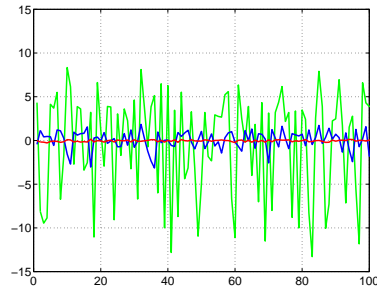
(a) vitesse de translation  
**AV 3Dtr (Dem)**



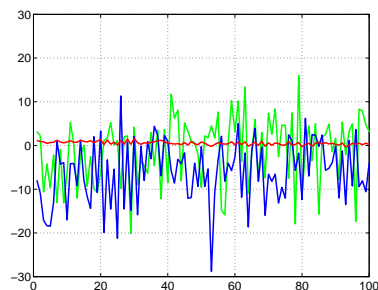
(b) vitesse de rotation  
**AV 3Dtr (Dem)**



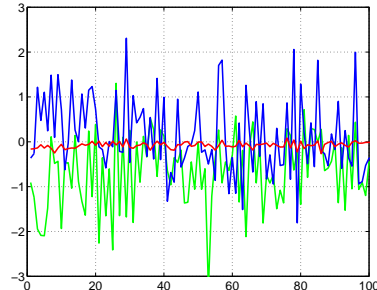
(c) vitesse de translation  
**AV 3Dtp (Dem)**



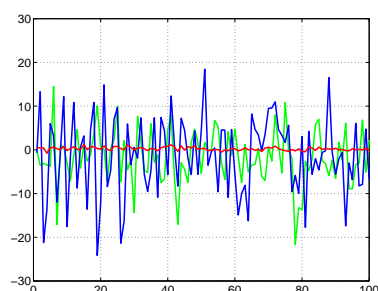
(d) vitesse de rotation  
**AV 3Dtp (Dem)**



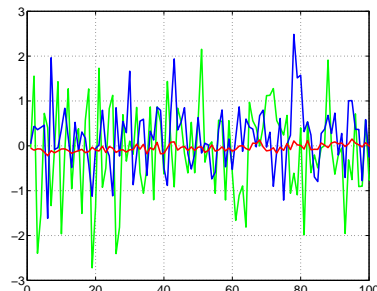
(e) vitesse de translation  
**AV 3Dtr (Lag)**



(f) vitesse de rotation  
**AV 3Dtr (Lag)**



(g) vitesse de translation  
**AV 3Dtp (Lag)**



(h) vitesse de rotation  
**AV 3Dtp (Lag)**

FIG. 4.31 – Niveau de bruit sur les lois de commande pour une cible parallèle au plan image (II)

### 4.6.3 Résultats pour une cible 3D

Nous présentons à présent les résultats obtenus sur la cible 3D composée de douze points. Les résultats relatifs à cette expérience sont résumés dans le Tableau 4.7. Les courbes correspondantes sont illustrées sur les Figures 4.32 et 4.33. Nous n'avons pas ici testé les asservissements visuels 3D en utilisant la méthode de Lagrange pour le calcul de pose, car la méthode de Dementhon est particulièrement bien adaptée à des objets 3D.

Nous pouvons remarquer que nous obtenons globalement les meilleurs résultats dans cette configuration, qui est malheureusement la moins pratique d'un point de vue applicatif. En effet, en utilisant un objet 3D, on peut être confronté au problème d'occultation d'une partie de l'objet par une autre.

La méthode la plus sensible au bruit de mesure est encore une fois l'**AV 2D $\frac{1}{2}$** . Ceci n'est pas étonnant car on rappelle, d'une part, que nous effectuons une estimation linéaire qui ne respecte pas exactement les contraintes non-linéaires du problème, et, d'autre part, que nous n'utilisons aucune information a priori sur la géométrie de l'objet. Pour obtenir des résultats plus robustes, il serait intéressant d'affiner les résultats obtenus par la méthode linéaire en utilisant une méthode non-linéaire, ce qui augmenterait sensiblement les temps de calcul. De plus, comme nous l'avons vu précédemment, les avantages de l'**AV 2D $\frac{1}{2}$**  se situent davantage dans leur grande robustesse aux erreurs de calibration.

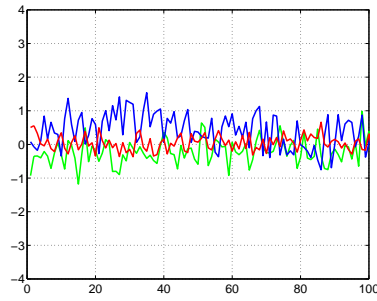
L'intérêt des méthodes non linéaires se retrouve également si l'on compare les résultats des **AV 3D** obtenus avec la méthode de Lowe ou la méthode Dementhon, puisqu'on obtient dans le premier cas des résultats tout à fait comparables à ceux obtenus en **AV 2Dc**. En effet, l'efficacité de la méthode non linéaire de Lowe, quand le nombre de points pris en compte est assez important, mais aussi le fait que les erreurs de mesure soient très faibles (de l'ordre d'un dixième de pixel) et que les erreurs de calibration soient également très faibles font que l'on dispose alors d'un capteur 3D quasiment parfait. Dans ce cas de figure (idéal), les techniques d'asservissement visuel 3D sont bien sûr intéressantes, même si l'on a vu que l'asservissement visuel 2D fournissait les résultats les plus robustes dans le cas du positionnement de la caméra par rapport à un objet plan incliné.

Finalement, et pour toutes les expériences réalisées, il semble impossible de distinguer une quelconque différence de robustesse aux erreurs de mesure entre l'**AV 3Dtr** et l'**AV 3Dtp**.

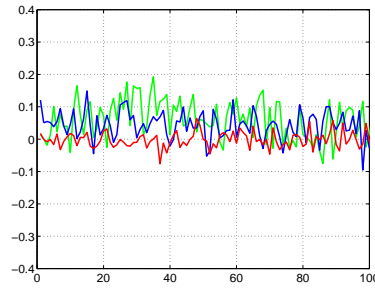


	AV 2Dc	AV 2D $\frac{1}{2}$	AV 3Dtr (Dem)	AV 3Dtr (Low)	AV 3Dtp (Dem)	AV 3Dtp (Low)
$a(v_x)$	<u>0.29</u>	1.80	<u>0.56</u>	<u>0.30</u>	<u>0.59</u>	<u>0.32</u>
$a(v_y)$	<u>0.48</u>	3.16	<u>0.73</u>	<u>0.41</u>	<u>0.75</u>	<u>0.39</u>
$a(v_z)$	<u>0.18</u>	0.42	<u>0.25</u>	<u>0.18</u>	<u>0.23</u>	<u>0.15</u>
$a(w_x)$	<u>0.05</u>	0.40	<u>0.09</u>	<u>0.05</u>	<u>0.09</u>	<u>0.05</u>
$a(w_y)$	<u>0.03</u>	0.21	<u>0.06</u>	<u>0.04</u>	<u>0.07</u>	<u>0.04</u>
$a(w_z)$	<u>0.02</u>	0.10	<u>0.04</u>	<u>0.02</u>	<u>0.04</u>	<u>0.02</u>

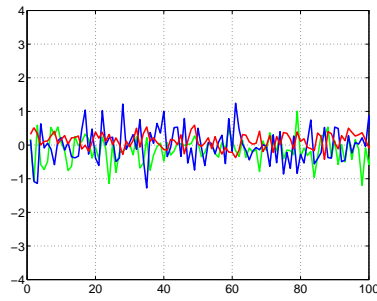
TAB. 4.7 – *Comparaison des méthodes pour la cible non-plane*



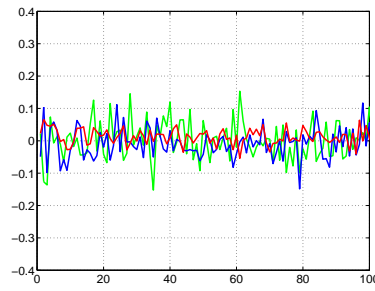
(a) vitesse de translation  
**AV 2Dc**



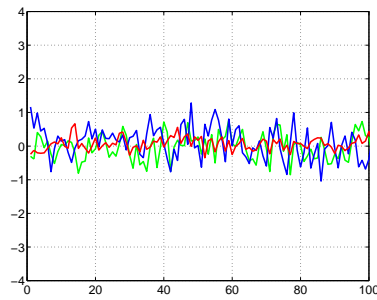
(b) vitesse de rotation  
**AV 2Dc**



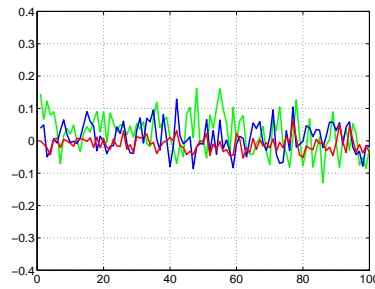
(c) vitesse de translation  
**AV 3Dtr (Low)**



(d) vitesse de rotation  
**AV 3Dtr (Low)**

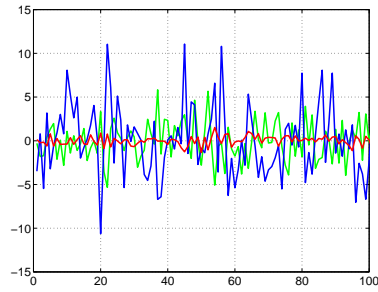


(e) vitesse de translation  
**AV 3Dtp (Low)**

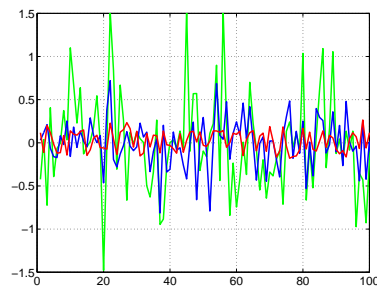


(f) vitesse de rotation  
**AV 3Dtp (Low)**

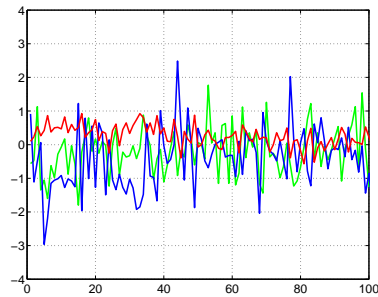
FIG. 4.32 – Niveau de bruit sur les lois de commande en utilisant un objet 3D (I)



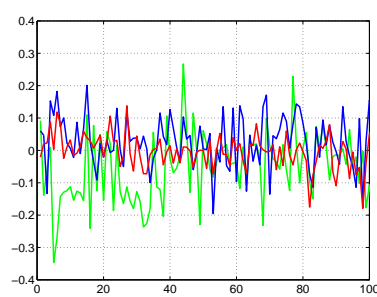
(a) vitesse de translation  
**AV 2D $\frac{1}{2}$**



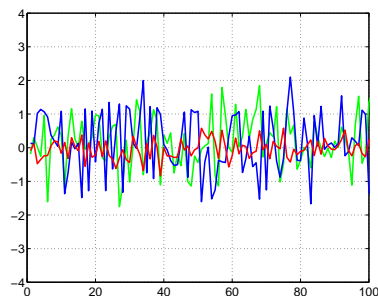
(b) vitesse de rotation  
**AV 2D $\frac{1}{2}$**



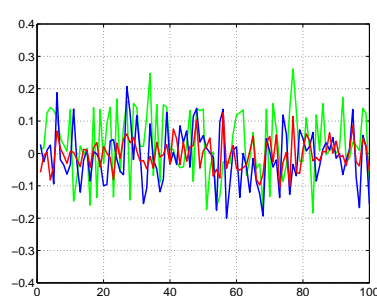
(c) vitesse de translation  
**AV 3Dtr (Dem)**



(d) vitesse de rotation  
**AV 3Dtr (Dem)**



(e) vitesse de translation  
**AV 3Dtp (Dem)**



(f) vitesse de rotation  
**AV 3Dtp (Dem)**

FIG. 4.33 – Niveau de bruit sur les lois de commande en utilisant un objet 3D (II)

## 4.7 Conclusion

Dans ce chapitre, nous avons vérifié expérimentalement les conclusions théoriques auxquelles nous étions arrivés. Nous pouvons donc reprendre les critères de comparaison définis dans l'introduction et les mettre à jour à la vue des résultats expérimentaux obtenus :

- **Information** nécessaire pour la modélisation. Par rapport à la technique 3D, les méthodes  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  ne nécessitent aucune connaissance a priori sur le modèle de l'objet, et par rapport à la technique 2D, les méthodes  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  ne nécessitent que l'introduction (grossière) de la profondeur d'un seul point. Si la cible est plane, les méthodes  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  nécessitent au moins quatre points mis en correspondance dans les deux images, exactement comme les méthodes 2D et 3D. Par contre, si la cible est quelconque les méthodes  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  nécessitent aux moins huit points.
- **Performance.** Tous les algorithmes présentés tournent à la cadence vidéo (mis à part le calcul de pose par la méthode de Lowe si l'on n'utilise pas un processeur très puissant). Le problème du retard introduit par le système de vision dans la boucle fermée se pose donc de manière identique pour toutes les techniques d'asservissement visuel. Si on considère également comme indice de performance le temps de convergence, on peut conclure qu'il est préférable d'utiliser en phase de convergence les méthodes d'asservissement visuel 2D, qui sont moins sensibles au bruit.
- **Sensibilité** au bruit de mesure. Les techniques d'asservissement visuel 2D sont moins sensibles au bruit de mesure que les méthodes pour lesquelles une estimation supplémentaire est nécessaire ( $2D\frac{1}{2}$ ,  $2D\frac{3}{4}$  et 3D). Ceci se traduit par une vitesse de convergence plus rapide et une meilleure stabilité en fin d'asservissement. Les méthodes d'asservissement visuel 2D sont donc les mieux adaptées pour finaliser la convergence.
- **Comportement** dynamique. Comme nous l'avons prévu de manière théorique, la méthode d'asservissement visuel 3D basée sur le contrôle du déplacement de la caméra estimé par la reconstruction de pose n'assure pas la condition de visibilité même si la calibration du système est parfaite. De même, l'asservissement visuel 2D peut présenter des problèmes de convergence si le déplacement de la caméra entre l'image courante et désirée est trop important. Les autres méthodes d'asservissement visuel ne présentent pas de problèmes particuliers de convergence et, de plus, dans le cas du  $2D\frac{1}{2}$  et du  $2D\frac{3}{4}$  la condition de visibilité de la tâche peut être prise en compte dans la commande.
- **Robustesse** aux erreurs de calibration. Toutes les méthodes d'asservissement considérées ont, à peu près, la même robustesse par rapport aux erreurs de calibration caméra/effecteur. Comme attendu, les méthodes basées sur une reconstruction de la pose de l'objet observé sont beaucoup plus sensibles aux erreurs de calibration de la caméra. De temps en temps, ces erreurs peuvent être compensées par les erreurs sur la matrice de passage caméra/effecteur. Les méthodes qui ont un contrôle direct ou indirect dans l'image sont généralement plus robustes aux erreurs de calibration. Les résultats confirment que les méthodes  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  donnent les meilleurs résultats en utilisant une commande robuste.



# Chapitre 5

## Asservissement visuel multi-caméras

Dans ce chapitre, nous étudions les techniques d'asservissement visuel dans le cas où plusieurs caméras sont utilisées afin de contrôler les mouvements du robot. L'objectif est d'améliorer la précision du positionnement de l'effecteur du robot dans le cas très général où plusieurs caméras embarquées sur cet effecteur n'observent pas nécessairement la même partie de la scène. On donnera un exemple d'une tâche de ce type dans un cas particulier où on utilise seulement deux caméras, mais toutes les considérations émises dans ce chapitre peuvent être généralisées au cas multi-caméras.

### 5.1 État de l'art

L'introduction de caméras supplémentaires dans la boucle d'asservissement visuel permet d'avoir une plus grande richesse d'informations sur la scène observée. En effet, en plus des deux classes que nous avons décrites dans le chapitre 1, une troisième classification des systèmes d'asservissement visuel peut être faite en fonction du nombre de caméras utilisées. On a vu dans le premier chapitre que la plus grande partie des systèmes d'asservissement visuel utilise une seule caméra, car l'architecture matérielle et logicielle est simplifiée, le temps de traitement d'image minimal, et le coût réduit. Toutefois, la perte d'information sur la profondeur peut compliquer la conception de la loi de commande si cette profondeur doit être estimée en ligne. Une solution à ce problème est l'utilisation de plusieurs caméras dans une configuration telle que l'objet observé soit dans le champ de vision de toutes les caméras (vision binoculaire, trinoculaire, etc.).

La configuration la plus connue utilisant deux caméras est dite stéréo (voir Figure 5.1), c'est-à-dire que l'on considère la partie de la scène où les champs de vision s'intersectent. Toutefois, afin de garder la généralité, on parlera plutôt de configuration bi-caméras. En effet, comme on le voit sur la Figure 5.1, les champs de vision des caméras peuvent ou non se croiser selon les cas considérés. Il apparaît donc clairement qu'une configuration stéréo est un cas particulier d'une configuration bi-caméras. On remarque que les deux caméras ne sont pas nécessairement toutes les deux embarquées ou déportées, mais l'une peut être embarquée et l'autre déportée. Encore une fois, comme dans le cas mono-caméra, les techniques d'asservissement visuel peuvent s'appliquer indifféremment à toutes ces configurations.

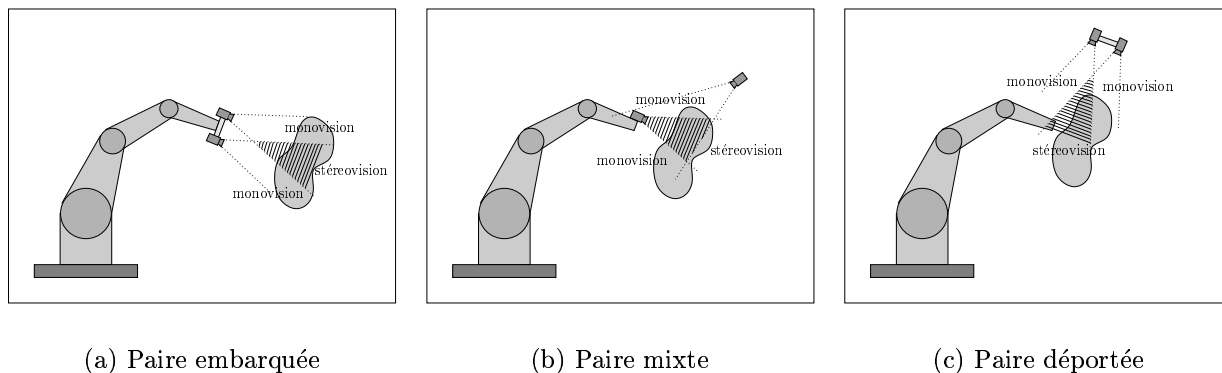


FIG. 5.1 – *Asservissement visuel multi-caméras*

La configuration la plus utilisée est la configuration stéréo car elle permet, si une calibration du système a été faite, d’obtenir une reconstruction euclidienne complète sur la scène observée. Par exemple, dans [Allen 93b], une paire stéréo déportée est utilisée afin de reconstruire la position d’un objet mobile à la cadence vidéo. Ensuite, l’effecteur du robot est commandé afin de poursuivre et saisir l’objet.

Par conséquent, une grande partie des systèmes stéréos a tendance à être commandée par un asservissement de type 3D, car ce mode de contrôle reste le plus “naturel” pour ceux qui raisonnent en termes de déplacement cartésien plutôt que de transformation 2D. Une première exception est donnée par le système d’asservissement visuel 2D proposé par [Maru 93], où les informations issues de deux caméras embarquées sur l’effecteur du robot sont utilisées pour mettre à jour le jacobien de la tâche. Ce système est de type “End-point Open Loop” et, comme on l’a déjà vu pour les systèmes mono-caméra, la précision de positionnement peut dépendre de la calibration du système, si on ne peut pas utiliser une technique d’apprentissage.

Bien que la connaissance de la structure de la cible ne soit plus nécessaire pour calculer le déplacement de l’effecteur du robot, la calibration de la paire stéréo reste un problème aussi délicat que la calibration mono-caméra. C’est pourquoi les systèmes “End-point Closed Loop” sont plus souvent utilisés. Nous rappelons que ces systèmes nécessitent une caméra déportée ou plusieurs afin d’observer en même temps l’effecteur du robot et l’objet considéré. La tâche de positionnement peut alors être effectuée en contrôlant directement dans l’image l’erreur entre la position de l’effecteur et celle de l’objet. Un tel système a été réalisé par [Hager 95] afin de contrôler d’abord un robot à trois degrés de liberté puis un robot à six degrés de liberté [Hager 97], en utilisant une technique d’asservissement visuel 2D où les informations visuelles considérées sont des invariants projectifs obtenus par un système stéréo non calibré.

Une technique d’asservissement visuel 3D, contrôlant un robot à cinq degrés de liberté et utilisant une paire stéréo déportée, a été proposée par [Cipolla 97]. Ce système utilise un modèle affine de la caméra, au lieu d’un modèle projectif, afin de reconstruire, à partir des points observés, la position de plans de référence sur l’objet et l’effecteur du robot. Ensuite, une technique d’asservissement visuel 3D est utilisée pour positionner le robot. Il faut sou-

ligner que l'approximation linéaire du modèle caméra simplifie grandement l'estimation et la loi de commande du robot, mais elle est valable seulement localement. Par conséquent, si le déplacement de la caméra est trop important, ce type de schéma d'asservissement peut rencontrer des problèmes de convergence.

Une dernière technique particulièrement intéressante, qui ne nécessite pas la calibration des caméras ni la calibration des repères caméras/effecteur, a été proposée par [Ruf 97] afin de contrôler le robot sans effectuer une reconstruction euclidienne. Cette méthode est basée sur l'estimation d'une matrice d'homographie projective, ce qui laisse espérer une extension presque immédiate des techniques d'asservissement visuel  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  au cas d'une paire stéréo. Il faut souligner toutefois, afin d'éviter toute confusion, que la matrice d'homographie de dimension  $(4 \times 4)$  estimée par [Ruf 97], liant les points de l'espace projectif n'est pas la même que celle présentée dans le Chapitre 2, liant des points d'un plan projectif et de dimension  $(3 \times 3)$ .

Même si l'utilisation de caméras supplémentaires dans le schéma d'asservissement visuel apporte une plus grande quantité d'informations, l'objet doit être dans le champ de vue de toutes les caméras. Plus le nombre de caméras utilisées est important, et plus il devient difficile d'avoir une grande zone d'intersection de leurs points de vue. De plus, il existe des tâches où les champs de vue des caméras ne peuvent pas se croiser, comme dans le cas, par exemple, du positionnement par rapport à un objet de très grande taille. Il est donc intéressant d'étudier aussi le cas d'un asservissement multi-caméras non forcément stéréo. On cherche donc, dans la suite de ce chapitre, à généraliser les lois de commande vues précédemment à l'utilisation de plusieurs caméras dont les champs de vision ne se recouvrent pas.

## 5.2 Une tâche d'asservissement multi-caméras

On décrit maintenant une tâche d'asservissement multi-caméras qui correspond à un besoin réel de la Direction des Études et Recherche d'Électricité de France. Il s'agit d'une intervention de manutention dans le bol d'un générateur de vapeur d'une centrale nucléaire. La tâche consiste à obturer le trou d'arrivée du circuit primaire d'un générateur de vapeur avec un couvercle circulaire d'environ 1 m de diamètre (voir Figure 5.2). Les caméras sont embarquées sur le robot et disposées autour du couvercle afin d'observer les bords du trou. Étant donné la grande dimension du couvercle, il est évident que :

- l'utilisation d'une seule caméra pourrait produire un "effet de levier", c'est-à-dire qu'une petite erreur sur le côté où se trouve la caméra pourrait induire une erreur plus importante du côté opposé. L'utilisation de plusieurs caméras est donc souhaitable ;
- les champs de vision des caméras ne peuvent pas se croiser, sauf si on utilise plusieurs paires stéréo ce qui est une hypothèse peu réaliste.



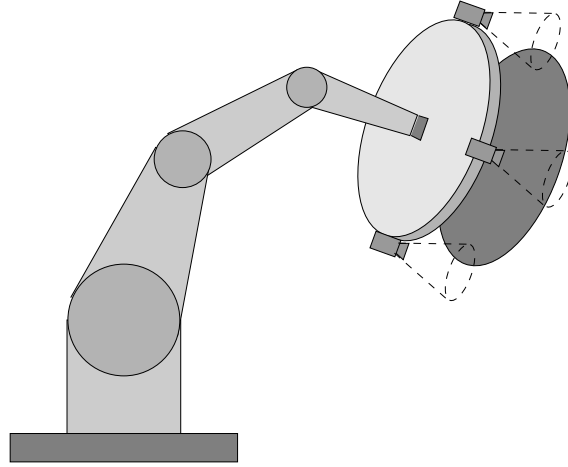


FIG. 5.2 – Pose du couvercle du circuit primaire d’un générateur de vapeur

Sur la Figure 5.2, on a représenté trois caméras mais leur nombre et leur disposition autour du couvercle est un problème encore ouvert. On fera ici l’hypothèse suivante :

*Des petites erreurs sur les informations visuelles d’une caméra impliquent qu’au moins une autre caméra du système est active (c’est-à-dire que la cible est dans son champ de vision).*

Il faut souligner que cette hypothèse est indépendante de notre connaissance de la calibration du système, mais elle dépend seulement de la précision qu’il est possible d’obtenir à partir de chaque caméra et de leur disposition dans le système. Cette hypothèse est donc vraie si deux caméras sont suffisamment “proches” et peut donc nous aider à déterminer le nombre minimal de caméras nécessaire pour réaliser la tâche de positionnement.

### 5.2.1 Modélisation

Soit  $\mathbf{s}_i$  le vecteur de dimension  $n_i$  contenant les signaux d’un capteur, et soit  $\mathbf{s} = [\mathbf{s}_1^T \ \mathbf{s}_2^T \ \dots \ \mathbf{s}_N^T]^T$  le vecteur de dimension  $n$  (où  $n = \sum_{i=1}^N n_i$ ) contenant l’ensemble des signaux de  $N$  capteurs. On suppose que chaque capteur peut contrôler tous les degrés de liberté de l’effecteur du robot, donc  $n_i \geq m$  (où en général  $m = 6$ ). Il faut souligner que même si on s’intéresse à une tâche d’asservissement visuel, le même formalisme peut être appliqué à l’utilisation de capteurs de nature différente.

La relation entre la dérivée des informations visuelles et la vitesse de l’effecteur du robot est donnée par:

$$\dot{\mathbf{s}} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \\ \vdots \\ \mathbf{L}_N \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 & 0 & \dots & 0 \\ 0 & \mathbf{W}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{W}_N \end{bmatrix} \dot{\mathbf{r}} = \mathbf{LW}\dot{\mathbf{r}} \quad (5.1)$$

où  $\mathbf{L}_i$  est la matrice d'interaction relative à la  $i$ -ème caméra, et  $\mathbf{W}_i$  la matrice de passage entre la vitesse de la  $i$ -ème caméra et la vitesse de l'effecteur du robot. La matrice d'interaction dépend toujours de la profondeur de l'objet. Dans le cas particulier où cet objet est dans le champ de vision d'au moins deux caméras, cette distance peut être estimée. Dans le cas contraire, elle sera simplement approximée pour chaque caméra, exactement comme dans l'asservissement visuel mono-caméra.

On construit une fonction de tâche  $\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*)$ , où  $\mathbf{C}$  est une matrice constante de rang plein et de dimension  $(m \times n)$ , qui permet de prendre en compte la redondance d'information. La dérivée de la fonction de tâche est :

$$\dot{\mathbf{e}} = \mathbf{C}\dot{\mathbf{s}} = \mathbf{CLW}\dot{\mathbf{r}} \quad (5.2)$$

La matrice  $\mathbf{C}$  doit être telle que la matrice  $\mathbf{CLW}$  soit de rang plein et peut être choisie de différentes façons :

- $\mathbf{C} = (\widehat{\mathbf{L}}^* \widehat{\mathbf{W}})^+$  est la pseudo-inverse de la matrice d'interaction calculée pour les valeurs désirées  $\widehat{\mathbf{L}}^* = \mathbf{L}(\mathbf{s}^*, \mathbf{g}^*)$ . Ce choix assure un meilleur découplage de la loi de commande à la convergence.
- $\mathbf{C} = (\widehat{\mathbf{L}}^* \widehat{\mathbf{W}})^T$  est la transposée de la matrice d'interaction calculée pour les valeurs désirées. Cette solution ne nécessite pas l'inversion du jacobien de la tâche, mais ne réalise pas le découplage de la loi de commande à la convergence.
- $\mathbf{C} = (\widehat{\mathbf{L}}^* \widehat{\mathbf{W}})^\#$  est la pseudo inverse à bloc de la matrice d'interaction:

$$(\widehat{\mathbf{L}}^* \widehat{\mathbf{W}})^\# = \begin{bmatrix} \kappa_1 \widehat{\mathbf{W}}_1^{-1} \widehat{\mathbf{L}}_1^{*+} & \kappa_2 \widehat{\mathbf{W}}_2^{-1} \widehat{\mathbf{L}}_2^{*+} & \kappa_3 \widehat{\mathbf{W}}_3^{-1} \widehat{\mathbf{L}}_3^{*+} & \dots & \kappa_N \widehat{\mathbf{W}}_N^{-1} \widehat{\mathbf{L}}_N^{*+} \end{bmatrix} \quad (5.3)$$

où  $\widehat{\mathbf{L}}_i^{*+}$  est la pseudo-inverse de la matrice  $\widehat{\mathbf{L}}_i^*$ ,  $\widehat{\mathbf{W}}_i^{-1}$  est l'inverse de la matrice  $\widehat{\mathbf{W}}_i$  et  $\kappa_i$  est un facteur de pondération positif tel que  $\sum_{i=1}^N \kappa_i = 1$ . Ce choix présente des caractéristiques intermédiaires par rapport aux deux choix précédents.

Il faut remarquer que la pseudo-inverse et la pseudo inverse à bloc sont des inverses généralisées. En effet, une matrice  $\mathbf{M}^{(-1)}$  est une inverse généralisée de  $\mathbf{M}$  si, et seulement si :

$$\mathbf{M} \mathbf{M}^{(-1)} \mathbf{M} = \mathbf{M} \quad (5.4)$$

La matrice  $\mathbf{M}^{(-1)}$  est respectivement une inverse à gauche ou à droite si  $\mathbf{M}^{(-1)} \mathbf{M} = \mathbf{I}$  ou  $\mathbf{M} \mathbf{M}^{(-1)} = \mathbf{I}$ . La matrice  $\mathbf{M}$  a une infinité d'inverses généralisées sauf si elle est carré et de rang plein car alors  $\mathbf{M}^{(-1)} = \mathbf{M}^{-1}$ . Parmi toutes les inverses généralisées d'une matrice  $\mathbf{M}$ , la pseudo-inverse  $\mathbf{M}^+$ , définie de manière unique, vérifie les quatre propriétés suivantes :

$$\mathbf{M} \mathbf{M}^+ \mathbf{M} = \mathbf{M} \quad (5.5)$$

$$\mathbf{M}^+ \mathbf{M} \mathbf{M}^+ = \mathbf{M}^+ \quad (5.6)$$

$$(\mathbf{M} \mathbf{M}^+)^T = \mathbf{M} \mathbf{M}^+ \quad (5.7)$$

$$(\mathbf{M}^+ \mathbf{M})^T = \mathbf{M}^+ \mathbf{M} \quad (5.8)$$

La propriété (5.5) est l'équivalent de la propriété (5.4). La pseudo-inverse a un rôle important dans la résolution des systèmes linéaires tels que  $\mathbf{y} = \mathbf{M}\mathbf{x}$ , car la solution  $\mathbf{x} = \mathbf{M}^+\mathbf{y}$  est la plus proche, au sens des moindres carrés, minimisant la norme  $\|\mathbf{y} - \mathbf{M}\mathbf{x}\|$ .

La pseudo-inverse à bloc que nous avons choisie vérifie les propriétés suivantes :

$$\mathbf{M} \mathbf{M}^\# \mathbf{M} = \mathbf{M} \quad (5.9)$$

$$\mathbf{M}^\# \mathbf{M} \mathbf{M}^\# = \mathbf{M}^\# \quad (5.10)$$

$$(\mathbf{M}^\# \mathbf{M})^T = \mathbf{M}^\# \mathbf{M} \quad (5.11)$$

et chaque bloc vérifie les quatre propriétés (5.5-5.8).

Il faut remarquer que toutes les matrices de combinaison  $\mathbf{C}$  données précédemment sont de rang  $m$ . Leur noyau est donc de dimension  $n-m$ . Seule la base du noyau est différente pour toutes les matrices. L'intérêt de l'un ou de l'autre choix sera plus clair lorsque l'on calculera la loi de commande et que l'on analysera la stabilité du système en boucle fermée. Toutefois, on peut voir tout de suite que le dernier choix présente des caractéristiques intéressantes. En effet, si on considère pour chaque caméra une fonction de tâche  $\mathbf{e}_i = \mathbf{C}_i \mathbf{s}_i$  avec  $\mathbf{C}_i = \widehat{\mathbf{W}}_i^{-1} \widehat{\mathbf{L}}_i^+$ , alors la fonction de tâche du système global est une moyenne pondérée des fonctions de tâche relatives à chaque caméra :

$$\mathbf{e} = \mathbf{C}\dot{\mathbf{s}} = \sum_{i=1}^N \kappa_i \mathbf{e}_i \quad (5.12)$$

Un tel choix de la matrice de combinaison conduit à une plus grande modularité. On peut construire une fonction de tâche pour chaque système et la fusion des données peut être effectuée dans un deuxième temps au niveau central. Par exemple, dans le cas de perte d'informations à cause de la défaillance d'un capteur, il suffit de ne plus ajouter la fonction de tâche correspondante. On peut même faire abstraction de la nature du capteur utilisé, à condition qu'il puisse contrôler tous les degrés de liberté du robot considéré. Par exemple, le même principe a été appliqué à l'utilisation conjointe d'une caméra et d'un capteur d'effort [Morel 98].

## 5.2.2 Commande

On impose une fois de plus une convergence exponentielle découplée de la fonction de tâche  $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ . La loi de commande théoriquement linéarisante est donc :

$$\dot{\mathbf{r}} = -\lambda(\mathbf{CLW})^{-1} \mathbf{e} \quad (5.13)$$

Comme on l'a déjà vu dans le cas de l'asservissement visuel mono-caméra, une loi de commande plus réaliste est la suivante :

$$\dot{\mathbf{r}} = -\lambda(\mathbf{CL}\widehat{\mathbf{W}})^{-1} \widehat{\mathbf{e}} \quad (5.14)$$

où les chapeaux indiquent les paramètres estimés ou approximatés. On choisit, comme dans [Chaumette 90], d'utiliser dans la commande la matrice d'interaction calculée dans la position d'apprentissage:  $\widehat{\mathbf{L}} = \mathbf{L}^*(\mathbf{s}^*, \widehat{\mathbf{g}}^*)$ . Par conséquent, en choisissant  $\mathbf{C} = (\widehat{\mathbf{L}}\widehat{\mathbf{W}})^\#$  définie par l'équation (5.3), on obtient :

$$(\mathbf{C}\widehat{\mathbf{L}}\widehat{\mathbf{W}})^{-1} = \left( \sum_{i=1}^N \kappa_i \widehat{\mathbf{W}}_i^{-1} \widehat{\mathbf{L}}_i^{*+} \widehat{\mathbf{L}}_i^* \widehat{\mathbf{W}}_i \right)^+ = \sum_{i=1}^N \kappa_i \mathbf{I}_m = \mathbf{I}_m \quad (5.15)$$

et la loi de commande est finalement :

$$\dot{\mathbf{r}} = -\lambda \widehat{\mathbf{e}} \quad (5.16)$$

En tenant compte de l'équation (5.12), on peut voir que cette loi de commande équivaut à une moyenne pondérée des lois de commande de chaque sous-système  $\dot{\mathbf{r}}_i = -\lambda \widehat{\mathbf{e}}_i$  :

$$\dot{\mathbf{r}} = -\lambda \sum_{i=1}^N \kappa_i \widehat{\mathbf{e}}_i = \sum_{i=1}^N \kappa_i \dot{\mathbf{r}}_i \quad (5.17)$$

### 5.2.3 Analyse de la stabilité

Afin de fermer la boucle, il faut calculer la fonction de tâche estimée en fonction de la fonction de tâche réelle et des erreurs de calibration du système. Dans le cas de l'asservissement visuel 2D, on a simplement  $\widehat{\mathbf{e}} = \mathbf{e}$  si on néglige le bruit de mesure. Le système en boucle fermée avec la loi de commande (5.16) est donc :

$$\dot{\mathbf{e}} = -\lambda(\mathbf{C}\widehat{\mathbf{L}}\widehat{\mathbf{W}})(\mathbf{C}\widehat{\mathbf{L}}\widehat{\mathbf{W}})^{-1}\mathbf{e} = -\lambda(\widehat{\mathbf{L}}\widehat{\mathbf{W}})^\#(\mathbf{L}\mathbf{W})\mathbf{e} \quad (5.18)$$

Une condition suffisante pour la stabilité du système est donc :

$$(\widehat{\mathbf{L}}^*\widehat{\mathbf{W}})^\#\mathbf{L}\mathbf{W} > 0 \quad (5.19)$$

On voit très bien ici l'intérêt de choisir comme matrice de combinaison la pseudo-inverse à bloc. En effet, grâce à l'équation (5.3), la condition (5.19) peut être aussi écrite de la manière suivante :

$$(\widehat{\mathbf{L}}^*\widehat{\mathbf{W}})^\#\mathbf{L}\mathbf{W} = \sum_{i=1}^N \kappa_i \widehat{\mathbf{W}}_i^{-1} \widehat{\mathbf{L}}_i^{*+} \mathbf{L}_i \mathbf{W}_i > 0 \quad (5.20)$$

et cette condition est vérifiée si :

$$\widehat{\mathbf{W}}_i^{-1} \widehat{\mathbf{L}}_i^{*+} \mathbf{L}_i \mathbf{W}_i > 0 \quad \forall i = \{1, 2, 3, \dots, N\} \quad (5.21)$$

Le système global sera donc stable si chaque sous-système est stable car la somme de deux matrices positives est positive. On peut donc étudier chaque sous-système séparément. Malheureusement, on ne connaît pas, au jour d'aujourd'hui, le domaine de stabilité de l'asservissement visuel 2D de manière formelle, et notre analyse s'arrêtera donc là. Toutefois, comme

on l'a vu dans le chapitre précédent, un système d'asservissement 2D mono-caméra est très robuste aux erreurs de calibration. On rappelle aussi que le domaine de stabilité est beaucoup plus large que ce que l'on pourrait imaginer suite à l'utilisation d'une approximation de la matrice d'interaction dans la loi de commande [Espiau 93]. On peut donc s'attendre à la même robustesse pour un système multi-caméras utilisant une technique d'asservissement visuel 2D.

**Remarque :** En utilisant le même procédé, il serait intéressant d'étudier si on peut simplifier l'étude de la stabilité locale d'un système d'asservissement 2D utilisant  $p$  points dans l'image. Si on considère  $N$  triplets de points non alignés, on peut construire  $N$  fonctions de tâche chacune contrôlant les six degrés de liberté de la caméra. La stabilité du système global peut donc se déduire de l'étude de la stabilité d'un simple triplet. Toutefois, il semble que le problème de l'analyse de la stabilité de trois points soit impossible à résoudre. Le problème est donc encore ouvert. En ce qui concerne les singularités d'un tel schéma de commande, elles ne sont pas les mêmes que celle présentes dans une configuration à trois points, mais sont liées au noyau de la matrice pseudo-inverse à bloc choisie. En effet, si un triplet est en singularité, il existe probablement (dans le cas général) au moins un autre triplet qui n'est pas en singularité. Il peut évidemment exister un cas où tous les triplets sont en configuration singulière et alors le rang de la matrice d'interaction sera inférieur à six.

Dans le cas d'un asservissement visuel  $2D\frac{1}{2}$  ou  $2D\frac{3}{4}$ , l'analyse de la stabilité est plus complexe car, pour chaque sous-système, on a  $\hat{\mathbf{e}}_i = \mathbf{E}_i \mathbf{e}_i$ . La fonction de tâche estimée est donc :

$$\hat{\mathbf{e}} = \sum_{i=1}^N \kappa_i \mathbf{E}_i \mathbf{e}_i \quad (5.22)$$

On ne peut donc pas exprimer  $\hat{\mathbf{e}}$  en fonction de  $\mathbf{e}$  sous la forme  $\hat{\mathbf{e}} = \mathbf{E} \mathbf{e}$  comme dans l'asservissement mono-caméra. On peut toutefois procéder de la manière suivante. La fonction de tâche peut s'écrire sous la forme :

$$\mathbf{e} = \begin{bmatrix} \kappa_1 \mathbf{I}_6 & \kappa_2 \mathbf{I}_6 & \dots & \kappa_N \mathbf{I}_6 \end{bmatrix} \mathbf{e}' = \mathbf{K} \mathbf{e}' \quad (5.23)$$

où  $\mathbf{e}'$  est le vecteur de dimension  $6N$  contenant les fonctions de tâche de chaque sous-système. On peut donc écrire :

$$\mathbf{e}' = \mathbf{K}^+ \mathbf{e} + (\mathbf{I}_{6N} - \mathbf{K}^+ \mathbf{K}) \phi \quad (5.24)$$

où  $(\mathbf{I}_{6N} - \mathbf{K}^+ \mathbf{K}) \phi$  est un vecteur (dont la forme n'est pas importante) appartenant au noyau de  $\mathbf{K}$ . Grâce à la forme très simple de  $\mathbf{K}$ , on a  $\mathbf{K}^+ = (\sum_{i=1}^N \kappa_i^2)^{-1} \mathbf{K}^T$ . En injectant les équations (5.24) et (5.23) dans l'équation (5.22), la fonction de tâche estimée peut donc s'écrire :

$$\hat{\mathbf{e}} = \frac{1}{\sum_{i=1}^N \kappa_i^2} \sum_{i=1}^N \kappa_i^2 \mathbf{E}_i \mathbf{e} + \mathbf{b}_e = \mathbf{E} \mathbf{e} + \mathbf{b}_e \quad (5.25)$$

où  $\mathbf{b}_e$ , qui peut être considéré comme un “biais” additif sur la fonction de tâche si les erreurs de calibration ne sont pas très importantes, a la forme :

$$\mathbf{b}_e = \sum_{i=1}^N \kappa_i \mathbf{E}_i \phi_i - \frac{1}{\sum_{i=1}^N \kappa_i^2} \sum_{i=1}^N \kappa_i^2 \mathbf{E}_i \left( \sum_{i=1}^N \kappa_i \phi_i \right) \quad (5.26)$$

On peut alors effectuer une approximation et négliger ce biais. Cette hypothèse peut être justifiée car, grâce à la forme très sympathique de la matrice  $\mathbf{E}_i = \text{diag}(\mathbf{E}_{v_i}, \mathbf{E}_{\omega_i})$  (diagonale à bloc, voir équation (3.29)), et des matrices  $\mathbf{L}_i$  et  $\mathbf{W}_i$  (triangulaires à bloc voir équations (3.13) et (1.29)), on peut séparer l’analyse de la stabilité en deux parties. Comme  $\mathbf{E}_{v_i} = \mathbf{I}$ , le “biais” est donc présent seulement sur le sous-système en rotation. On peut envisager alors une commande robuste du même type que celle décrite pour l’asservissement  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$ , en utilisant un gain adaptatif pour le contrôle de la rotation.

De plus, si on choisit une fonction de tâche pour chaque sous-système de rotation de la forme :

$$\mathbf{e}_{\omega_i} = [ \delta \mathbf{A}_i \mathbf{u}_i \theta_i ] \quad (5.27)$$

on a  $\hat{\mathbf{e}}_{\omega_i} = \mu_i \mathbf{e}_{\omega_i}$  avec  $\mu_i = \|\delta \mathbf{A}_i \mathbf{u}_i\|$ . La fonction de tâche  $\hat{\mathbf{e}}_{\omega_i}$  qui est utilisée dans la loi de commande peut être estimée directement à partir des données dans l’image. On obtient alors  $\mathbf{E}_i = \text{diag}(\mathbf{I}, \mu_i \mathbf{I})$  et :

$$\mathbf{b}_e = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sum_{i=1}^N \kappa_i \mu_i \phi_{\omega_i} - \frac{1}{\sum_{i=1}^N \kappa_i^2} \sum_{i=1}^N \kappa_i^2 \mu_i \left( \sum_{i=1}^N \kappa_i \phi_{\omega_i} \right) \end{bmatrix} \quad (5.28)$$

Si on suppose que  $\mu_i \approx 1$ , l’hypothèse de considérer  $\mathbf{b}_e$  comme un “bruit” additif sur la fonction de tâche est justifiée. En effet, les trois dernières composantes de  $\mathbf{b}_e$  deviennent :

$$\sum_{i=1}^N \kappa_i \phi_{\omega_i} - \frac{\sum_{i=1}^N \kappa_i^2}{\sum_{i=1}^N \kappa_i^2} \left( \sum_{i=1}^N \kappa_i \phi_{\omega_i} \right) \approx \sum_{i=1}^N \kappa_i \phi_{\omega_i} - \sum_{i=1}^N \kappa_i \phi_{\omega_i} \approx 0 \quad \forall \phi_{\omega_i} \quad (5.29)$$

Par conséquent, l’analyse de la stabilité des asservissements visuels multi-caméras  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$  se réduit à l’analyse de la stabilité de chaque sous-système. Comme chaque sous-système est stable sous les conditions vues dans le chapitre précédent, le système global est également stable, au moins sous les mêmes conditions.

Suivant le domaine de stabilité qu’on pourra obtenir de la loi de commande, on peut envisager deux types d’algorithme de plus haut niveau afin de porter à convergence le système

multi-caméras. Dans les cas où le domaine de convergence de chaque sous-système est relativement restreint (comme dans le cas de l'asservissement visuel 2D), on propose l'algorithme suivant :

- 1) Tester combien de caméras sont à la convergence (une caméra est à la convergence si l'erreur sur ses informations visuelles est inférieure à un certain seuil  $\epsilon_i$  :  $\|\mathbf{s}_i - \mathbf{s}_i^*\| < \epsilon_i$ ). Si toutes les caméras ont convergé, l'objectif est atteint, sinon aller à 2).
- 2) Sélectionner toutes les caméras qui ont convergé plus celle qui est la plus proche de la convergence (la caméra la plus proche de la convergence, notée  $j$ , est celle qui minimise  $\min \|\mathbf{s}_j - \mathbf{s}_j^*\|$  sous contrainte  $\|\mathbf{s}_j - \mathbf{s}_j^*\| > \epsilon_j$ ). Aller à 3).
- 3) Effectuer l'asservissement visuel en utilisant les caméras sélectionnées et après leur convergence, aller à 1).

Cet algorithme itératif ajoute donc à chaque étape une nouvelle caméra. Dans les cas où le domaine de convergence est très large (comme dans le cas de l'asservissement visuel 2D $\frac{1}{2}$ ), un algorithme plus satisfaisant peut être considéré :

- 1) Tester combien de caméras sont à la convergence. Si toutes les caméras ont convergé l'objectif est atteint, sinon aller à 2).
- 2) Sélectionner toutes les caméras qui ont convergé, plus celles qui ont la cible dans leur champ de vision (et qui sont dans leur domaine de stabilité). Aller à 3).
- 3) Effectuer l'asservissement visuel en utilisant les caméras sélectionnées et après leur convergence aller à 1).

Si les hypothèses sur la répartition équitable des caméras dans l'espace de travail sont satisfaites, alors les deux algorithmes convergent au pire en  $N$  itérations. Ceci implique que :

- une caméra à la convergence y reste à la fin de chaque étape. Ceci ne signifie pas que la caméra reste à la convergence tout le long de l'asservissement. En effet, seule une décroissance de la fonction de tâche globale peut être assurée.
- Si  $p$  (avec  $p < N$ ) caméras sont à la convergence, il existe au moins une autre caméra "active", c'est-à-dire avec sa cible dans le champ de vision et dans son domaine de stabilité.

### 5.3 Résultats expérimentaux

Nous allons appliquer les considérations théoriques présentées dans les sections précédentes à un système démonstratif simulant la pose du couvercle. Afin de démontrer la faisabilité de la tâche, nous n'avons utilisé que deux caméras. Elles sont montées aux extrémités d'une barre d'aluminium de 50 cm (dont le poids est d'environ 2 kg), embarquée sur un robot manipulateur comme en Figure 5.3.

On a utilisé le robot Mitsubishi PA10 disponible à la DER d'EDF Chatou et deux caméras Panasonic (diamètre de 12mm et focale de 6mm). Afin d'augmenter la rapidité du traitement d'images, on utilise deux cibles planaires avec seulement six points chacune. Les points sont disposés de manière différente afin de pouvoir distinguer les deux cibles. On a observé de manière expérimentale qu'à la distance de 50 cm une erreur d'environ 1 pixel sur les coordonnées des points de la caméra gauche se traduit par une erreur d'environ 10 pixels sur les coordonnées des points de la caméra droite.

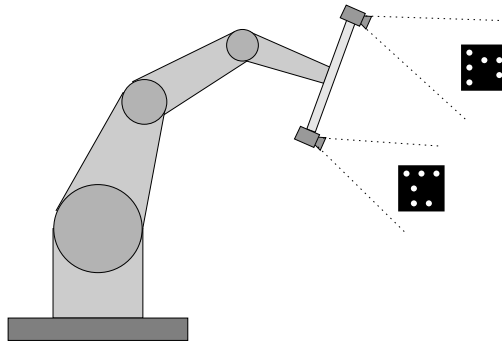


FIG. 5.3 – *Maquette simulant la pose d'un objet encombrant*

Afin de rendre le système robuste aux erreurs de calibration, on effectue un apprentissage préalable des informations visuelles correspondant à la position désirée de l'effecteur. La position d'apprentissage est telle que les plans images des deux caméras soient à peu près parallèles aux plans des cibles.

Une architecture optimale du contrôleur visuel aurait été obtenue avec la parallélisation et la synchronisation du système d'acquisition. Toutefois, nous avons été contraints d'utiliser une architecture différente : les deux caméras sont connectées à deux entrées d'une même carte d'acquisition et l'acquisition des deux images se fait de manière séquentielle. La cadence d'un tel système est deux fois plus lente que la cadence d'un système parallèle. En utilisant la carte EDIXIA IA 2000, la boucle complète (c'est-à-dire la numérisation des deux trames, le traitement des images et le calcul de la commande) a une durée de 80 ms (soit le double de la cadence vidéo) avec une durée entre les deux acquisitions d'environ 40 ms. Par conséquent, les vitesses de l'effecteur du robot durant l'asservissement ne pourront pas être trop élevées. En effet, le délai entre les acquisitions d'images (40 ms) n'est pas pris en compte dans la loi de commande. On choisit donc une convergence exponentielle de la fonction de tâche avec une vitesse de convergence  $\lambda = 0.1$  pour toutes les expériences.



### 5.3.1 Comparaison de la précision entre asservissement mono et multi-caméras

Comme nous l'avons vu dans le chapitre précédent, l'asservissement visuel 2D est très peu sensible aux erreurs de mesure. C'est donc cette méthode que l'on va utiliser pour cette expérience où l'asservissement est arrêté lorsque l'erreur maximale sur les coordonnées des points dans l'image est de 1 pixel. Afin d'obtenir une bonne précision de positionnement, on se place dans des conditions optimales pour l'asservissement visuel mono-caméra, à savoir que les cibles occupent une grande partie de l'image à la convergence car la distance entre la caméra et la cible n'est pas très élevée.

On effectue quatre expériences, correspondant à quatre déplacements différents de l'effecteur du robot afin que chacune des deux cibles soit présente dans les champs de vision de la caméra correspondante. D'abord, on effectuera l'asservissement en utilisant seulement une des deux caméras. Ensuite, à partir de la même position initiale, on effectuera l'asservissement en utilisant conjointement les deux caméras suivant la loi de commande (5.16) décrite dans la section précédente.

Les déplacements de l'effecteur pour chaque expérience sont donnés dans le tableau 5.1 (la rotation est donnée suivant la représentation  $\mathbf{u}\theta$ ). Le premier déplacement est une translation pure. Le deuxième présente une rotation principalement autour de l'axe  $\vec{x}$  et le troisième une rotation principalement autour de l'axe  $\vec{y}$ . Enfin, le quatrième déplacement présente une grande rotation autour des axes  $\vec{x}$  et  $\vec{y}$  et une grande translation.

position	translation (mm)	rotation (deg)
1	-5.7 -13.5 -34.0	0.0 0.01 0.01
2	18 -35.3 -21.3	-8.02 -5.66 1.33
3	-71.4 -26.5 -23	-4.56 13.60 -1.03
4	-101.1 -82.6 21.5	-16.18 19.84 0.6

TAB. 5.1 – Déplacements du repère terminal du robot

Les précisions mesurées après l'asservissement visuel mono-caméra sont données dans le tableau 5.2. Ces résultats confirment la grande précision que l'on peut obtenir avec une telle méthode d'asservissement. Il faut souligner que, même si les caméras et la matrice de

passage entre la caméra et l'effecteur du robot sont grossièrement calibrées, la précision de positionnement est élevée grâce à l'apprentissage des informations visuelles désirées.

position	translation (mm)	rotation (deg)
1	1.4 -0.8 0.3	-0.16 0.3 0.02
2	-1.2 -1.5 -0.5	-0.27 0.25 0.02
3	-0.3 -1.7 -0.8	-0.23 0.15 -0.00
4	-1.1 -1.6 -0.6	-0.22 0.36 -0.07

TAB. 5.2 – Précision de positionnement avec l'asservissement visuel mono-caméra

Les précisions mesurées après l'asservissement visuel multi-caméras sont données dans le tableau 5.3.

position	translation (mm)	rotation (deg)
1	-1.0 -0.5 0.2	-0.09 0.20 0.01
2	-0.5 -0.5 0.5	-0.07 0.14 0.00
3	-0.3 -0.9 0.4	-0.09 0.18 -0.01
4	-0.9 -0.5 -0.2	-0.12 0.20 -0.06

TAB. 5.3 – Précision de positionnement avec l'asservissement visuel multi-caméras

On remarque une amélioration de la précision du positionnement de l'effecteur du robot, principalement en translation, d'environ 1 mm sur la norme de l'erreur. Il faut souligner que l'amélioration sera encore plus évidente avec un objet plus encombrant, car la distance entre

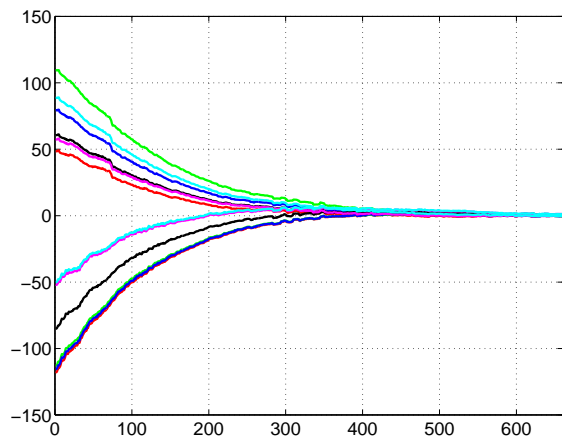
les deux caméras sera encore plus élevée.

En ce qui concerne l'asservissement visuel 2D multi-caméras, on donne respectivement sur les Figures 5.4, 5.5, 5.6 et 5.7 les résultats obtenus à partir des positions 1, 2, 3 et 4. Dans toutes les expériences, on utilise les paramètres intrinsèques de la caméra fournis par le constructeur et on ne tient pas compte des bagues qui ont été ajoutées sur chaque caméra afin d'améliorer la mise au point. La matrice homogène de transformation des coordonnées entre chaque repère caméra et l'effecteur du robot est effectuée "à la main". Enfin, les facteurs de pondération de la pseudo-inverse à bloc utilisée dans la commande ont été fixés à  $1/2$ .

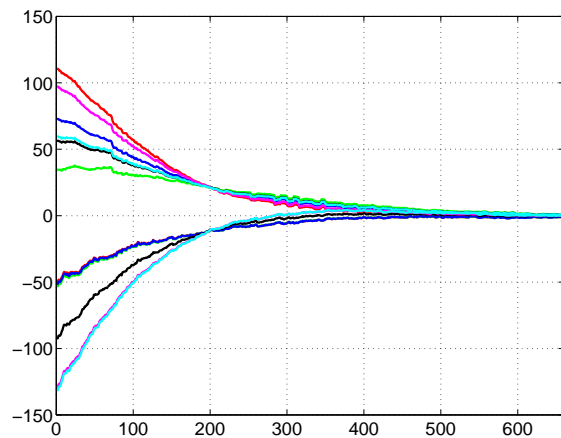
On peut remarquer le comportement un peu "ondulatoire" des informations visuelles dans l'image. Toutefois, la loi de commande fournie par le contrôleur visuel ne semble pas être très perturbée, comme le montrent les Figures 5.4(e) et 5.4(f), 5.5(e) et 5.5(f), 5.6(e) et 5.6(f) et enfin, 5.7(e) et 5.7(f). Cet effet est donc probablement dû au fait que la charge supplémentaire d'environ 2 kg, constituée par l'ensemble barre plus caméras, n'est pas prise en compte dans la boucle de commande bas niveau. Il est évident qu'une simple commande en vitesse, ne tenant pas compte de l'inertie du système, peut produire ce genre d'effet.

La loi de commande est robuste aux erreurs de calibration et elle assure la convergence de l'asservissement multi-caméras. Comme on le verra par la suite, on peut même se permettre de perturber cette loi de commande en ajoutant des erreurs de calibration, ce qui prouve que la grande robustesse de l'asservissement visuel 2D mono-caméra est conservée par la loi de commande constituée d'une moyenne pondérée des lois de commande relative à chaque caméra.

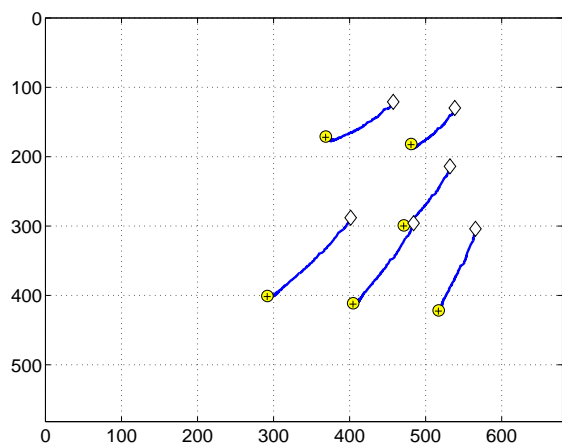
Le passage à l'utilisation de plusieurs caméras ne semble donc pas poser de problèmes particuliers de robustesse liés à la loi de commande. De même, les trajectoires des points dans les images de droite et de gauche ont une allure comparable aux trajectoires obtenues par les asservissements mono-caméra.



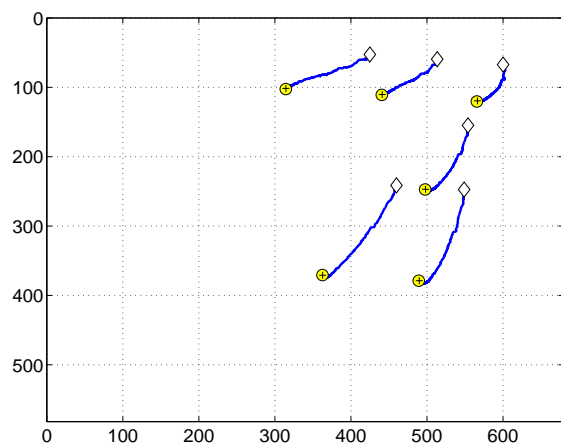
(a) erreurs sur les coordonnées de gauche



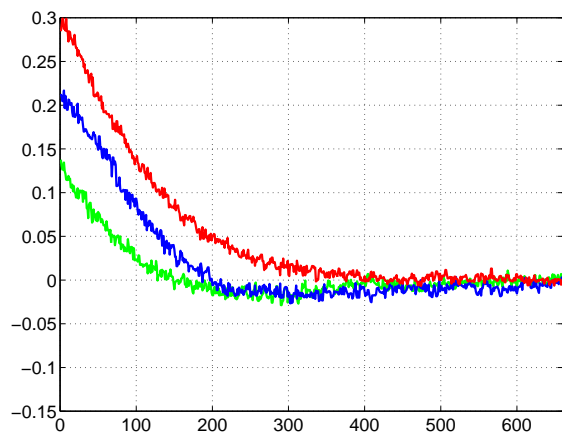
(b) erreurs sur les coordonnées de droite



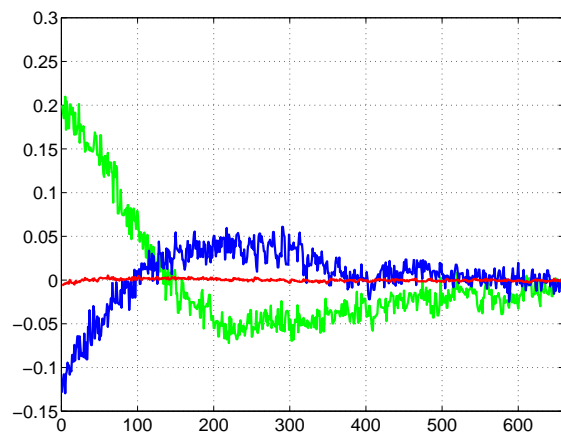
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

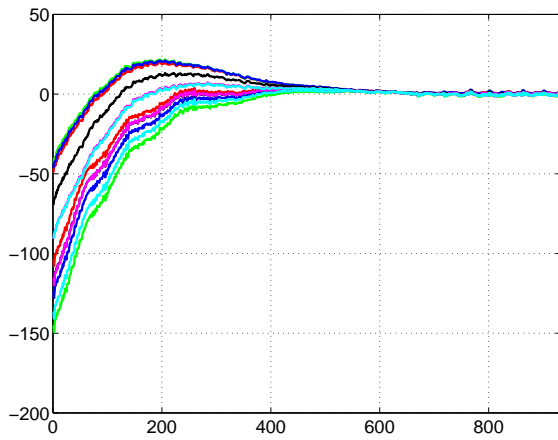


(e) vitesse en translation (cm/s)

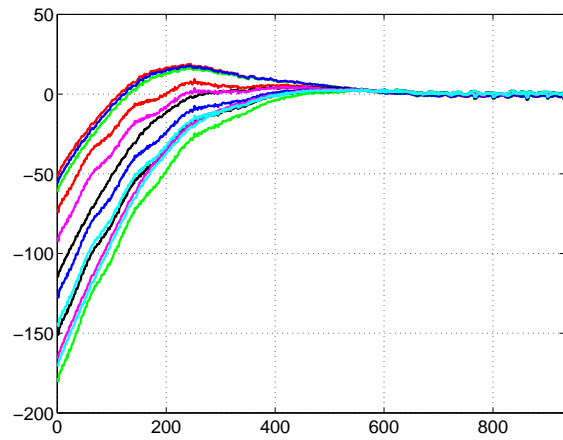


(f) vitesse en rotation (deg/s)

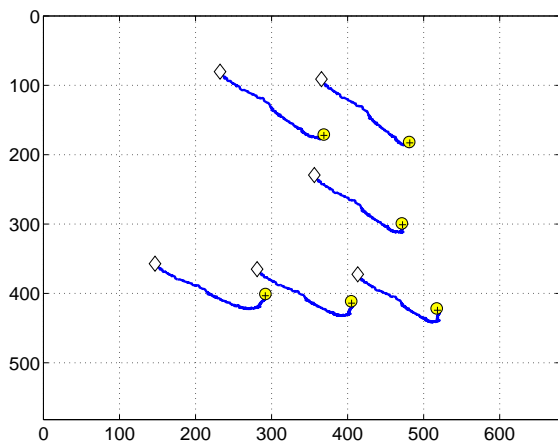
FIG. 5.4 – Première expérience



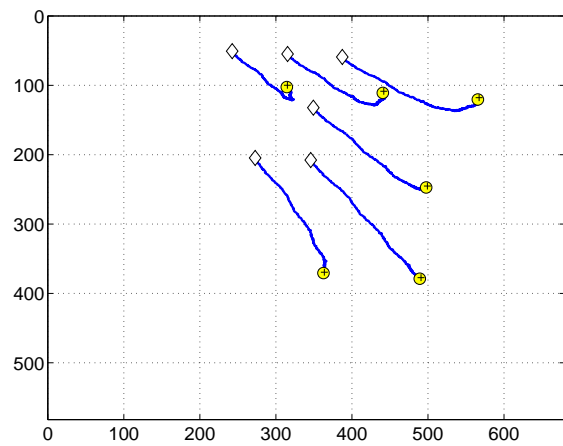
(a) erreurs sur les coordonnées de gauche



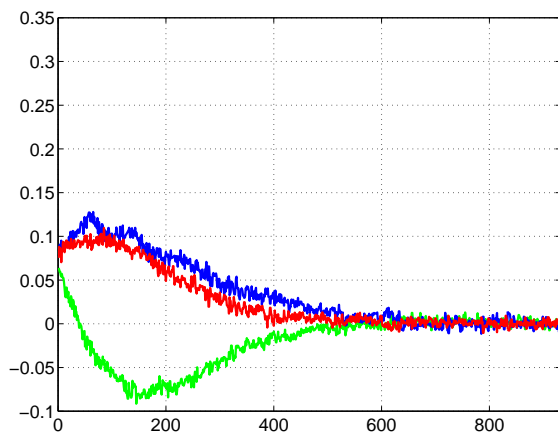
(b) erreurs sur les coordonnées de droite



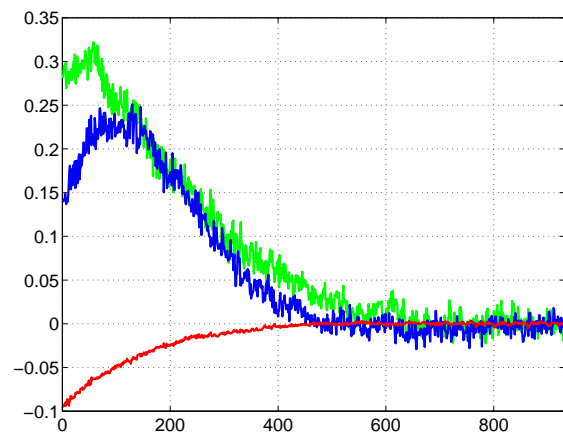
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

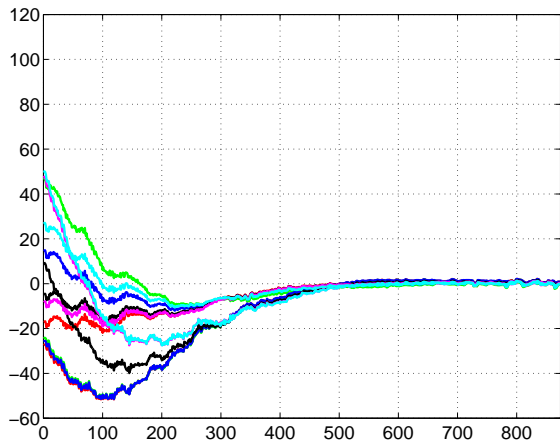


(e) vitesse en translation (cm/s)

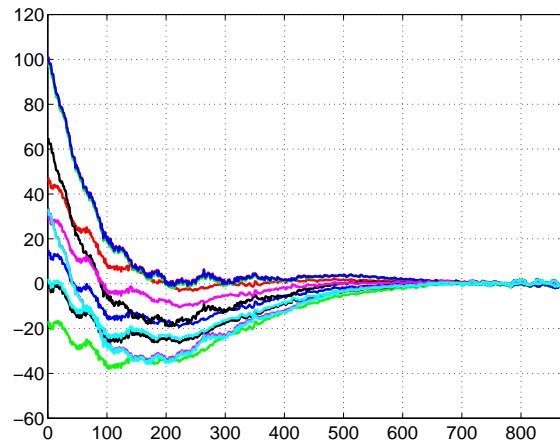


(f) vitesse en rotation (deg/s)

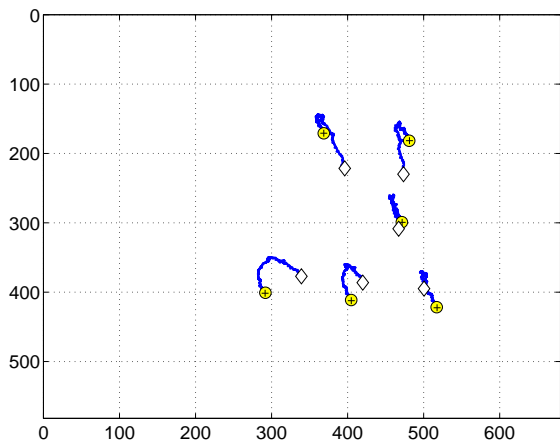
FIG. 5.5 – Deuxième expérience



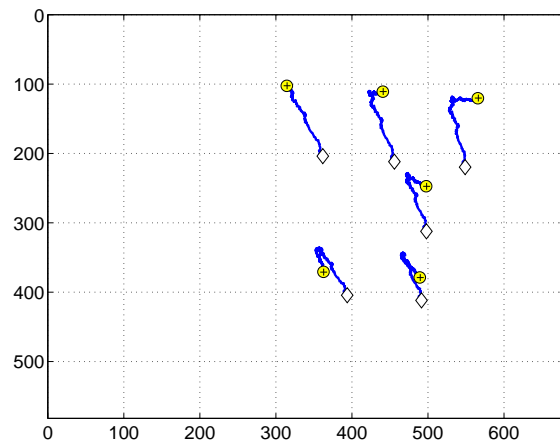
(a) erreurs sur les coordonnées de gauche



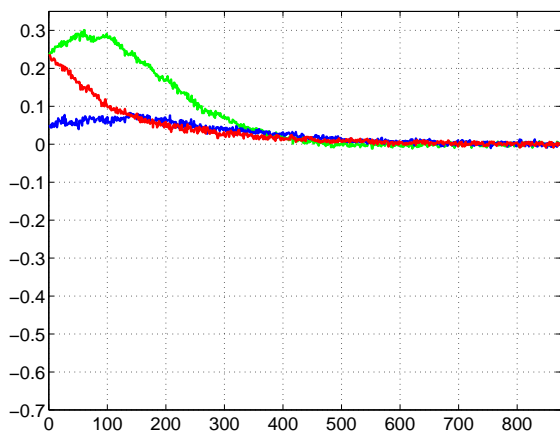
(b) erreurs sur les coordonnées de droite



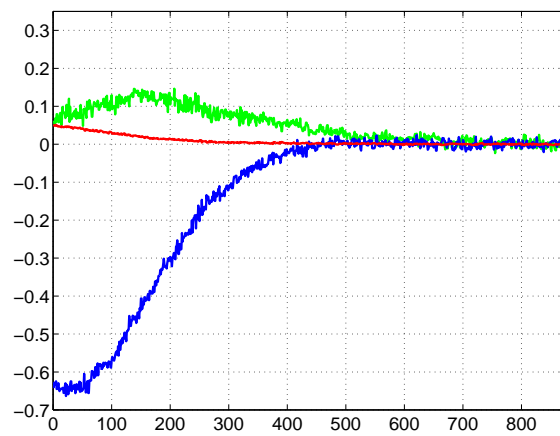
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

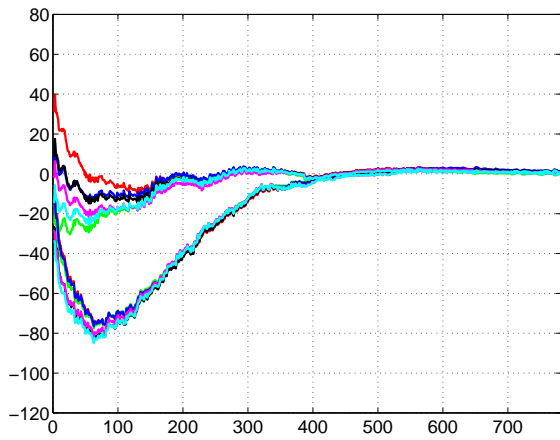


(e) vitesse en translation (cm/s)

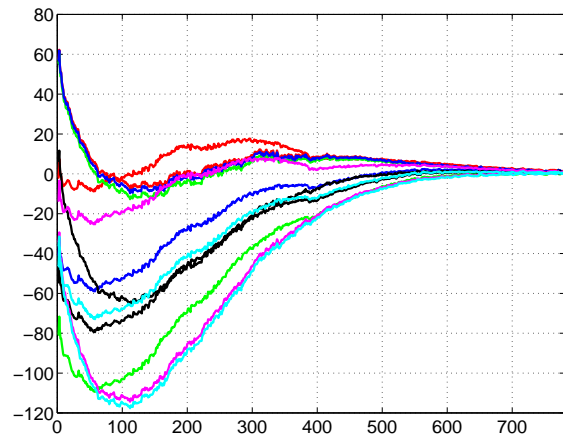


(f) vitesse en rotation (deg/s)

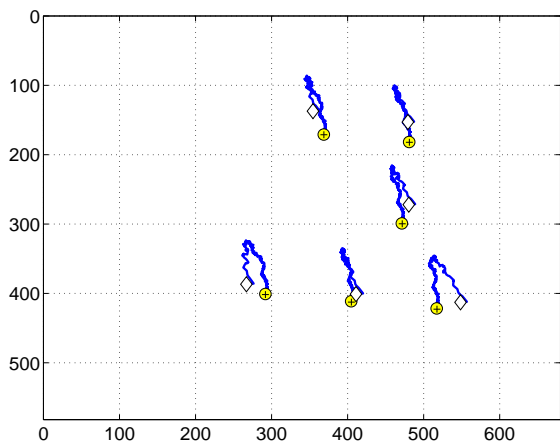
FIG. 5.6 – Troisième expérience



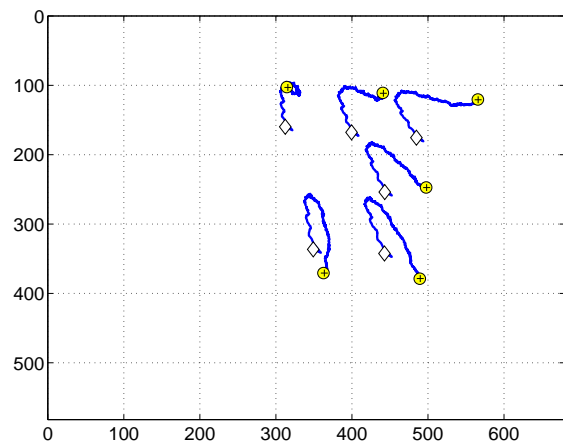
(a) erreurs sur les coordonnées de gauche



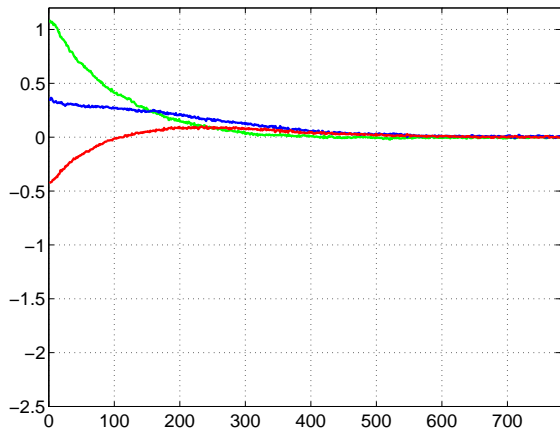
(b) erreurs sur les coordonnées de droite



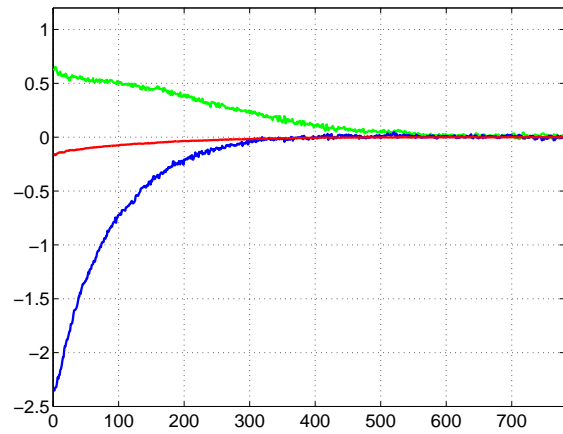
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite



(e) vitesse en translation (cm/s)



(f) vitesse en rotation (deg/s)

FIG. 5.7 – Quatrième expérience

### 5.3.2 Comparaison entre les asservissements multi-caméras 2D et $2D\frac{1}{2}$

Bien que l'asservissement visuel multi-caméras puisse être réalisé avec n'importe quel schéma d'asservissement (2D,  $2D\frac{1}{2}$ ,  $2D\frac{3}{4}$ , et 3D), on se limite ici à la comparaison des asservissements 2D et  $2D\frac{1}{2}$ . On réalise des déplacements un peu plus importants que ceux effectués précédemment. Il faut toutefois remarquer qu'il est difficile d'obtenir un grand déplacement du repère terminal du robot tout en gardant les deux cibles dans le champ de vue des caméras. Par exemple, une très faible rotation de l'effecteur autour de son axe  $\vec{z}$  fait sortir les cibles des images. De plus, les caméras ont une focale courte et le champ de vue est réduit. Les déplacements sont présentés dans le tableau 5.4 et sont similaires à ceux des positions 1 et 4 du tableau 5.1. Dans cette série d'expériences, l'asservissement visuel est arrêté dès que l'erreur maximale sur les coordonnées des points atteint 2 pixels car, comme on l'a déjà vu dans le chapitre précédent, l'asservissement visuel  $2D\frac{1}{2}$  est plus sensible aux erreurs de mesure que l'asservissement visuel 2D.

position	translation (mm)	rotation (deg)
1	-2.7 -11.7 -52.4	0.02 -0.03 -0.03
2	-105.9 -82.4 31.2	-17.36 21.46 0.5

TAB. 5.4 – Déplacements du repère terminal du robot

#### 5.3.2.1 Calibration grossière

Dans cette première série d'expériences, on utilise une calibration grossière du système de vision. Encore une fois, on utilise les paramètres intrinsèques de la caméra fournis par le constructeur et on ne tient pas compte des bagues qui ont été ajoutées sur chaque caméra afin d'améliorer la mise au point. La matrice homogène de transformation des coordonnées entre chaque repère caméra et l'effecteur du robot est effectuée "à la main".

Les précisions de positionnement mesurées pour l'asservissement visuel 2D et  $2D\frac{1}{2}$  sont données respectivement dans les tableaux 5.5 et 5.6. Comme cela était prévisible, les précisions sont pratiquement équivalentes, car elles ne dépendent pas du schéma d'asservissement utilisé mais uniquement de l'erreur maximale atteinte dans l'image. Par conséquent, les précisions de positionnement obtenues dans ces expériences sont un peu moins élevées que celles obtenues dans les expériences précédentes, où l'asservissement était arrêté quand l'erreur maximale était de 1 pixel.



position	translation (mm)	rotation (deg)
1	-0.7 -0.7 0.3	-0.11 0.15 0.01
2	-0.8 -0.6 -1.0	0.20 0.20 -0.06

TAB. 5.5 – Précision de positionnement avec l’asservissement visuel multi-caméras 2D

position	translation (mm)	rotation (deg)
1	0.8 0.7 0.6	0.12 -0.12 -0.01
2	0.8 -0.1 -1.2	0.10 -0.10 0.01

TAB. 5.6 – Précision de positionnement avec l’asservissement visuel multi-caméras  $2D\frac{1}{2}$

En ce qui concerne les expériences d’asservissement visuel multi-caméras avec calibration grossière, on donne respectivement :

- sur la Figure 5.8, les résultats de l’asservissement visuel 2D obtenus à partir de la position 1.
- sur les Figures 5.9 et 5.10, les résultats de l’asservissement visuel  $2D\frac{1}{2}$  obtenus à partir de la position 1.
- sur la Figure 5.11, les résultats de l’asservissement visuel 2D obtenus à partir de la position 2.
- sur les Figures 5.12 et 5.13, les résultats de l’asservissement visuel  $2D\frac{1}{2}$  obtenus à partir de la position 2.

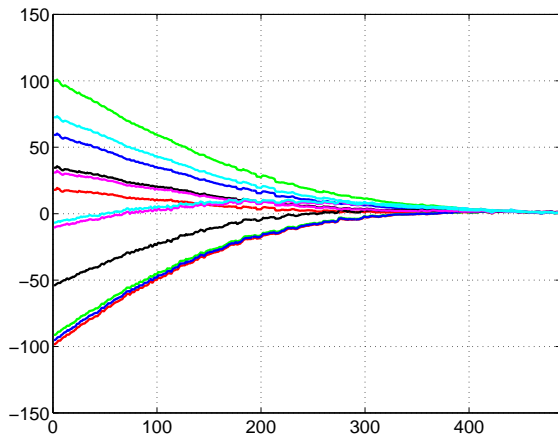
Dans le cas de l’asservissement visuel  $2D\frac{1}{2}$ , on donne la fonction de tâche estimée (c’est-à-dire les coordonnées étendues dans l’image et la rotation estimée) en plus de l’erreur sur les coordonnées des points, de la trajectoire des points dans l’image et de la loi de commande.

On peut voir sur les Figures 5.9(e) et 5.9(f) que l’asservissement visuel  $2D\frac{1}{2}$  est un peu plus sensible aux erreurs de mesure. En effet, entre les itérations 400 et 500, soit à proximité de la convergence, les estimations de la profondeur relative (voir Figures 5.10(a) et 5.9(b)) et de la rotation (voir Figures 5.10(a) et 5.9(b)) sont un peu bruitées, ce qui se répercute sur la loi de commande en translation. Toutefois, la loi de commande en boucle fermée est

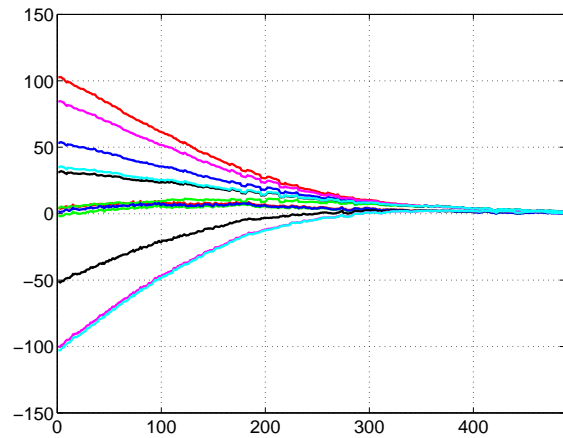
capable d'absorber ce type de perturbation et la convergence est réalisée à peu près avec le même nombre d'itérations.

Les trajectoires des points dans les images sont quasiment les mêmes à partir de la position 1 (voir les Figures 5.8(c) et 5.8(d) pour l'asservissement visuel 2D, et les Figures 5.9(c) et 5.9(d) pour l'asservissement visuel  $2D\frac{1}{2}$ ) et encore très proches à partir de la position 2 (voir les Figures 5.11(c) et 5.11(d) pour l'asservissement visuel 2D, et les Figures 5.12(c) et 5.12(d) pour l'asservissement visuel  $2D\frac{1}{2}$ ).

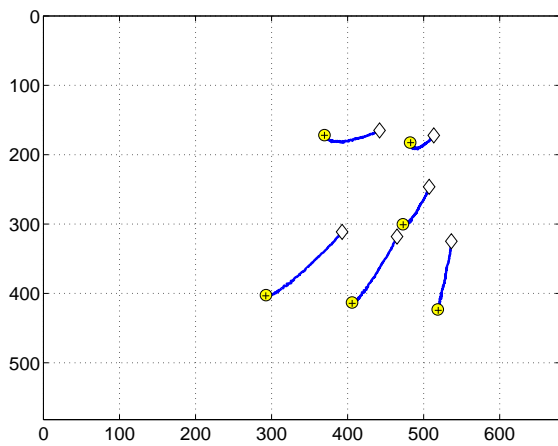
Les lois de commande obtenues en partant de la position 1 sont également très proches (voir les Figures 5.8(e) et 5.8(f) pour l'asservissement visuel 2D, et les Figure 5.9(e) et 5.9(f) pour l'asservissement visuel  $2D\frac{1}{2}$ ), alors que, en partant de la position 2, on peut remarquer une différence plus prononcée (voir les Figures 5.11(e) et 5.11(f) pour l'asservissement visuel 2D, et les Figures 5.12(e) et 5.12(f) pour l'asservissement visuel  $2D\frac{1}{2}$ ). Cela est dû au déplacement plus important réalisé.



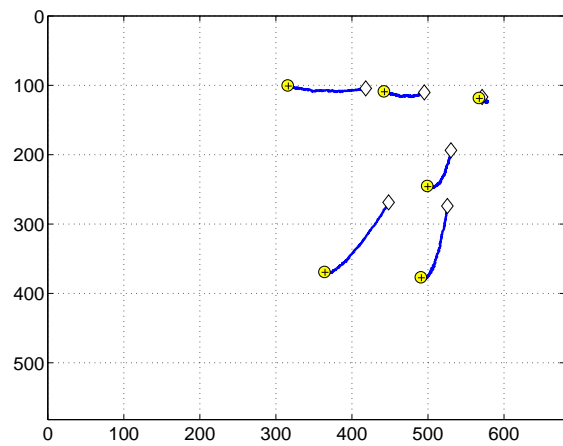
(a) erreurs sur les coordonnées de gauche



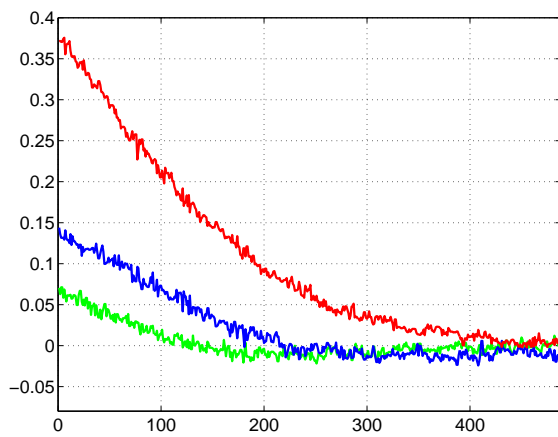
(b) erreurs sur les coordonnées de droite



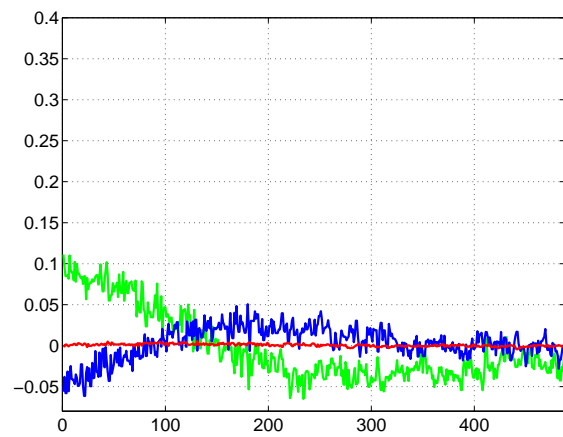
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

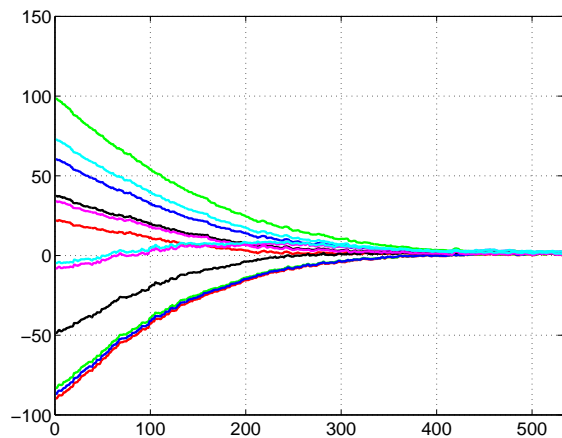


(e) vitesse en translation (cm/s)

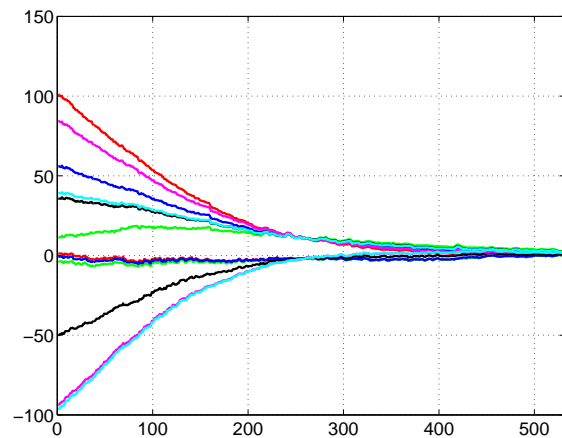


(f) vitesse en rotation (deg/s)

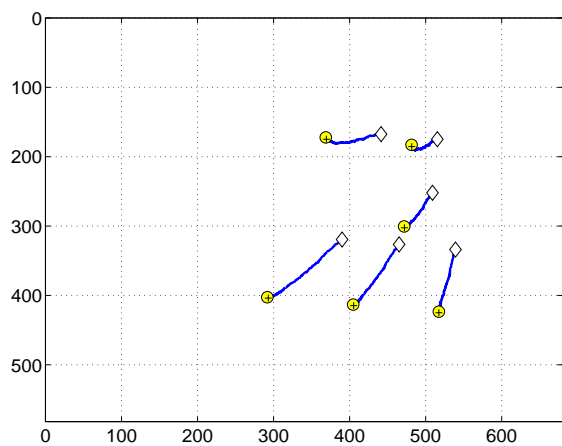
FIG. 5.8 – Asservissement visuel 2D avec calibration grossière: Première expérience



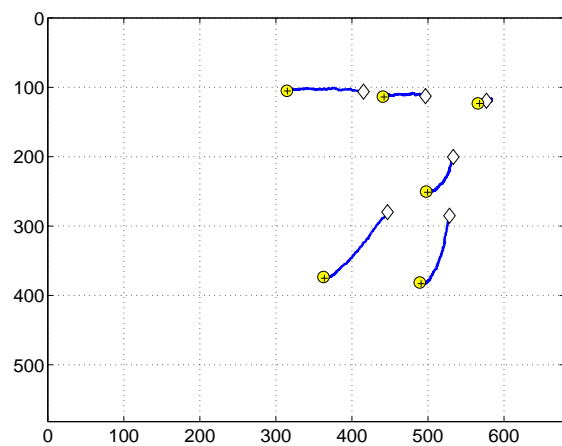
(a) erreurs sur les coordonnées de gauche



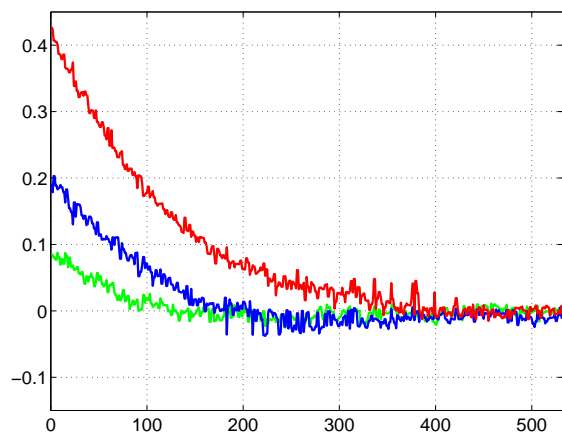
(b) erreurs sur les coordonnées de droite



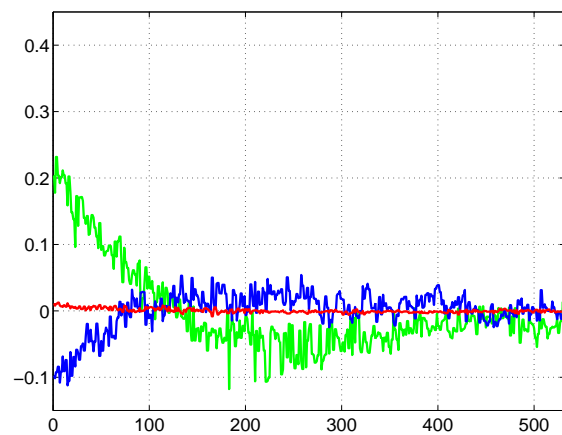
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

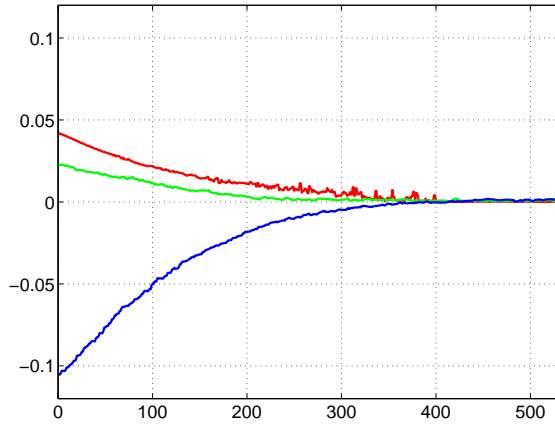


(e) vitesse en translation (cm/s)

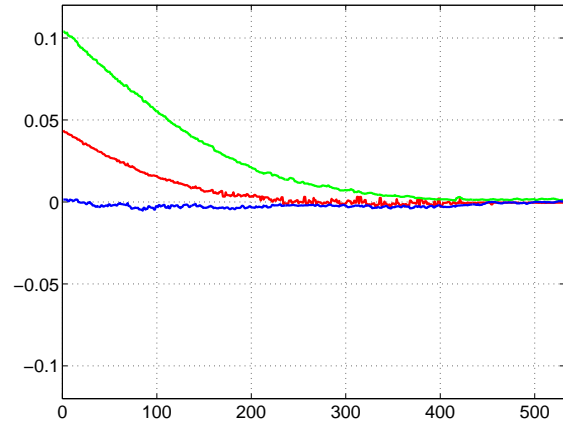


(f) vitesse en rotation (deg/s)

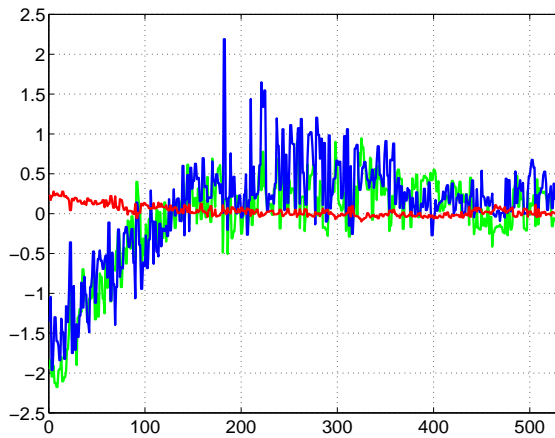
FIG. 5.9 – Asservissement visuel  $2D_2^1$  avec calibration grossière : Première expérience



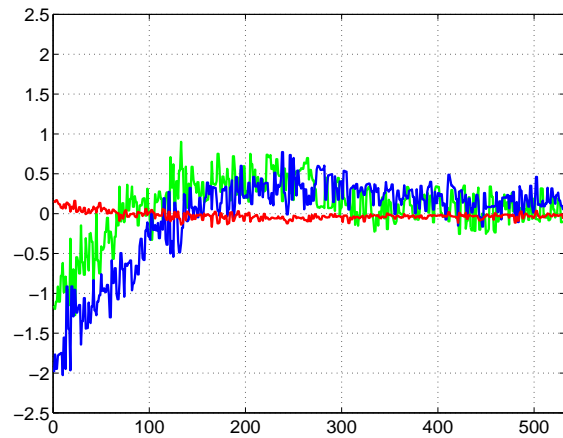
(a) coordonnées étendues de l'image gauche



(b) coordonnées étendues de l'image droite

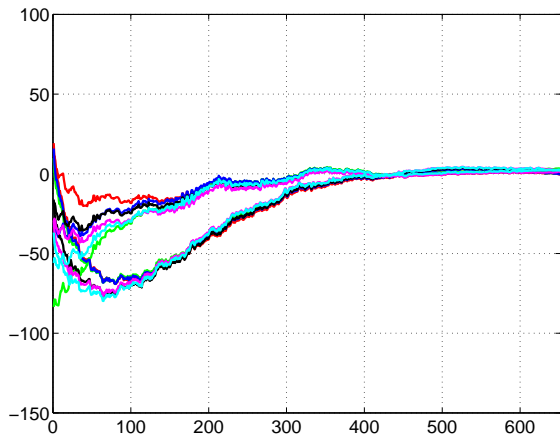


(c) rotation de la caméra gauche

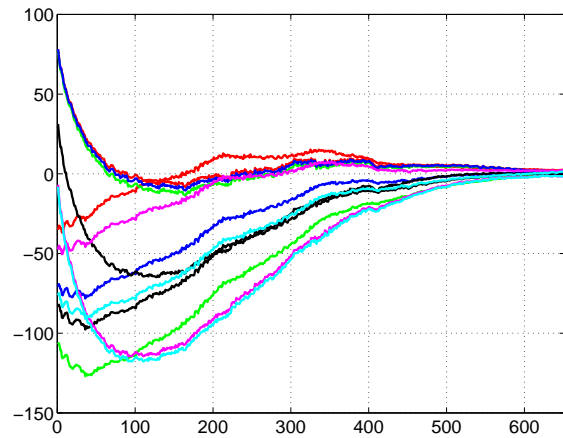


(d) rotation de la caméra droite

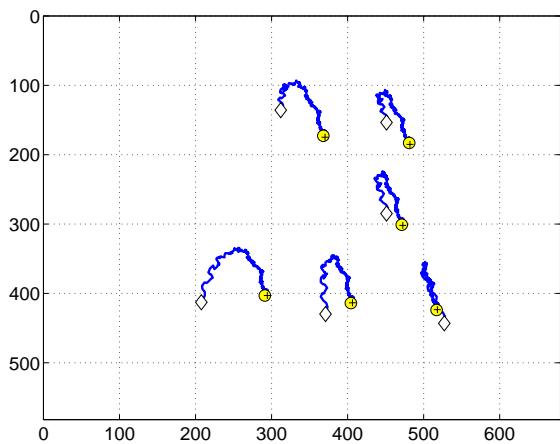
FIG. 5.10 – Asservissement visuel  $2D_2^1$  avec calibration grossière : Première expérience



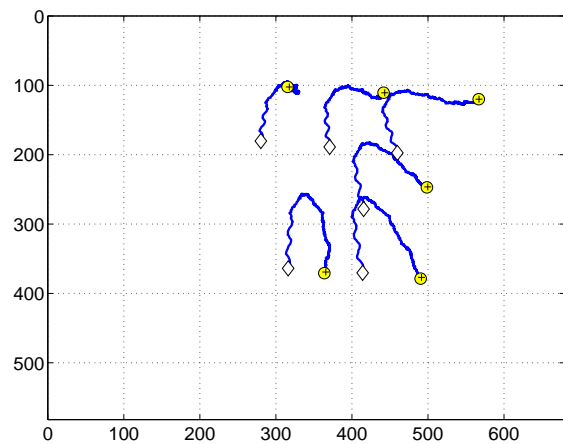
(a) erreurs sur les coordonnées de gauche



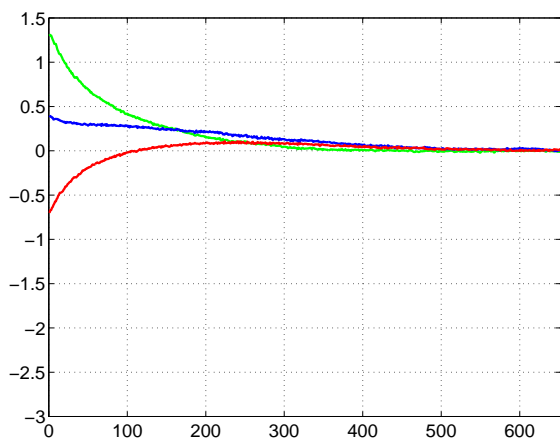
(b) erreurs sur les coordonnées de droite



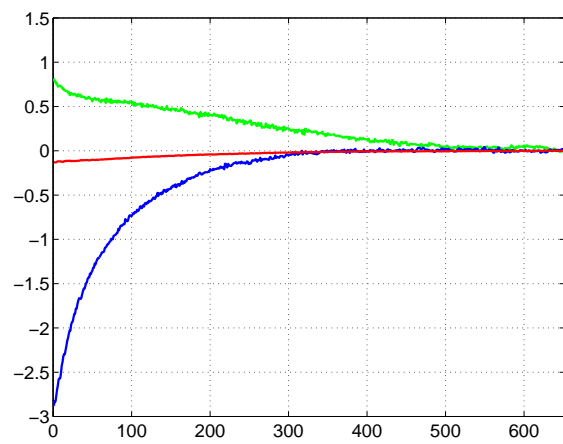
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

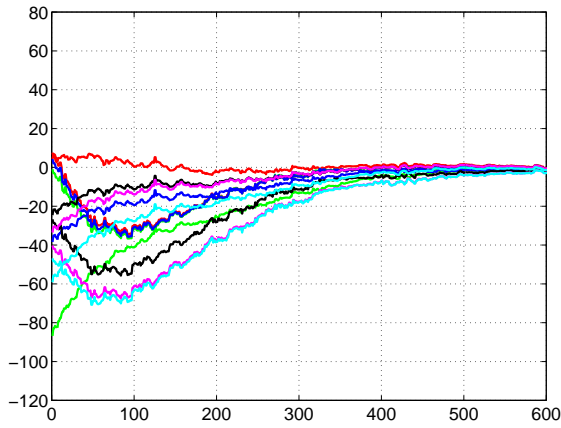


(e) vitesse en translation (cm/s)

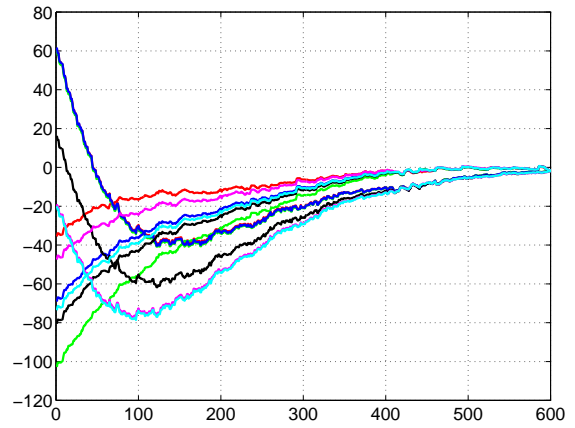


(f) vitesse en rotation (deg/s)

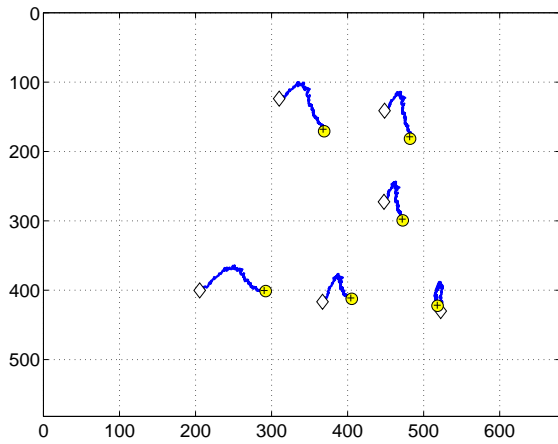
FIG. 5.11 – Asservissement visuel 2D avec calibration grossière : deuxième expérience



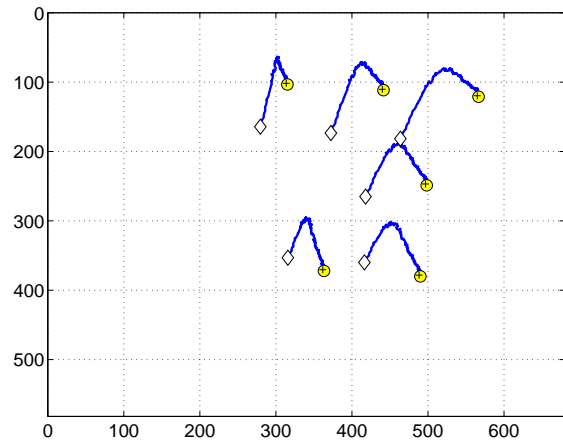
(a) erreurs sur les coordonnées de gauche



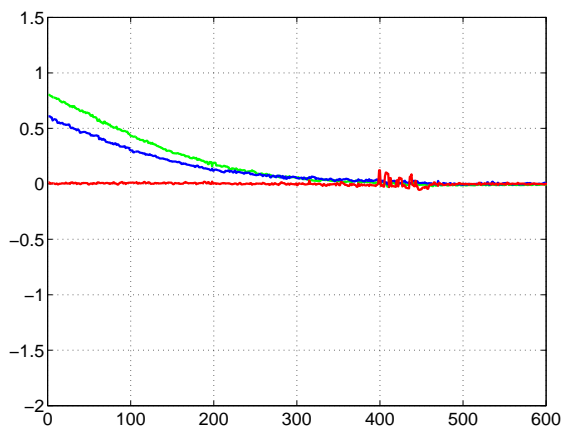
(b) erreurs sur les coordonnées de droite



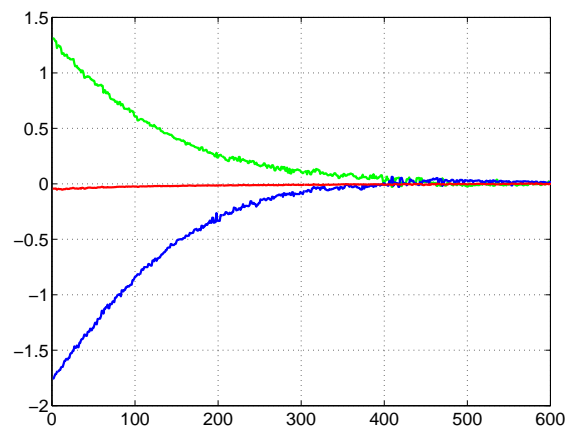
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

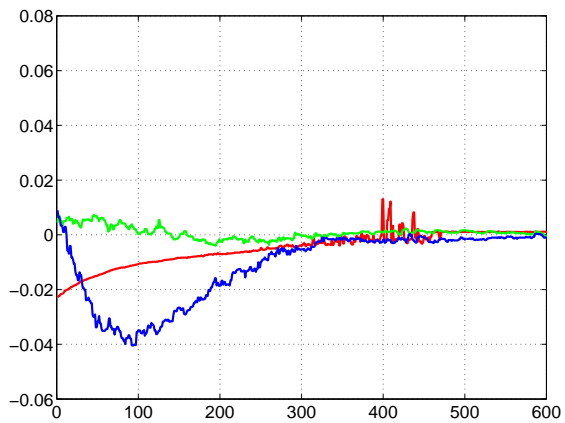


(e) vitesse en translation (cm/s)

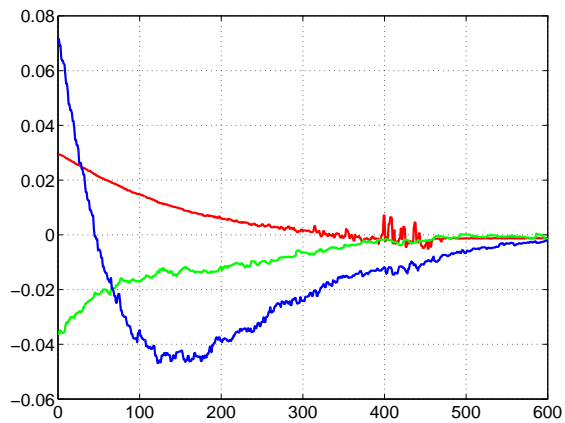


(f) vitesse en rotation (deg/s)

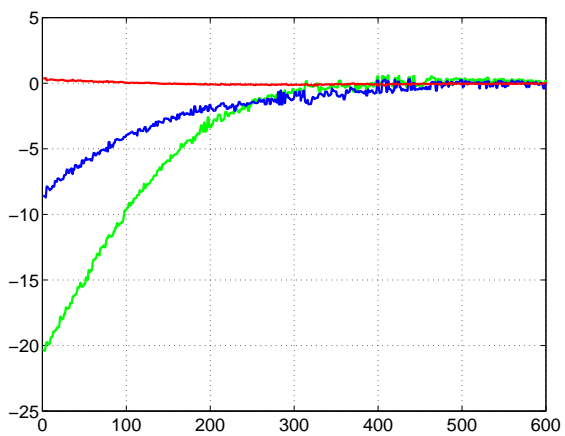
FIG. 5.12 – Asservissement visuel  $2D\frac{1}{2}$  avec calibration grossière: deuxième expérience



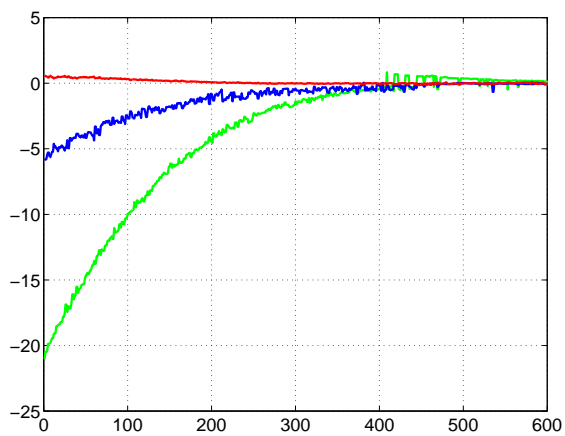
(a) coordonnées étendues de l'image gauche



(b) coordonnées étendues de l'image droite



(c) rotation de la caméra gauche



(d) rotation de la caméra droite

FIG. 5.13 – Asservissement visuel  $2D\frac{1}{2}$  avec calibration grossière : deuxième expérience



### 5.3.2.2 Calibration médiocre

Par rapport à la série d'expériences précédentes, on ajoute pour chaque caméra une erreur supplémentaire de 5 cm sur chaque composante en translation et une erreur de 5 degrés sur chaque composante en rotation sur la matrice de passage entre le repère caméra et celui de l'effecteur du robot. Toutefois, puisque la calibration a été faite "à la main", on ne peut pas savoir si on a augmenté exactement de ces valeurs l'erreur de calibration.

Les précisions mesurées pour l'asservissement visuel 2D et  $2D\frac{1}{2}$  sont respectivement données dans les tableaux 5.7 et 5.8. Encore une fois, les précisions de positionnement ne diffèrent pas beaucoup.

position	translation (mm)	rotation (deg)
1	-1.3 -0.8 -0.4	0.16 0.28 0.01
2	-0.8 -0.9 -1.1	-0.10 0.20 -0.06

TAB. 5.7 – Précision de positionnement avec l'asservissement visuel multi-caméras 2D

position	translation (mm)	rotation (deg)
1	0.7 -0.8 0.4	-0.26 -0.06 -0.01
2	0.8 -0.6 -0.9	-0.21 -0.19 -0.02

TAB. 5.8 – Précision de positionnement avec l'asservissement visuel multi-caméras  $2D\frac{1}{2}$

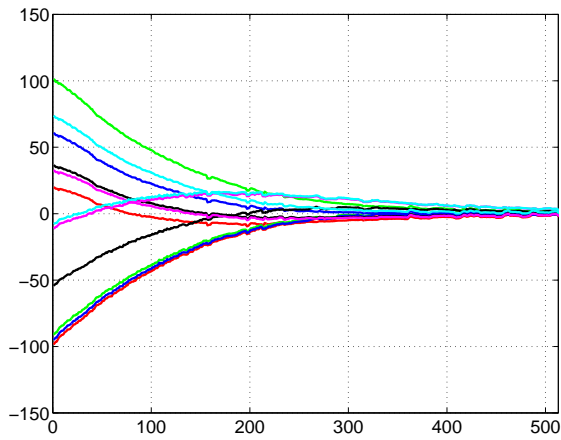
En ce qui concerne les expériences d'asservissement visuel multi-caméras avec calibration médiocre, on donne respectivement :

- sur la Figure 5.14, les résultats de l'asservissement visuel 2D obtenus à partir de la position 1.
- sur les Figures 5.15 et 5.16, les résultats de l'asservissement visuel  $2D\frac{1}{2}$  obtenus à partir de la position 1.
- sur la Figure 5.17, les résultats de l'asservissement visuel 2D obtenus à partir de la position 2.
- sur les Figures 5.18 et 5.19, les résultats de l'asservissement visuel  $2D\frac{1}{2}$  obtenus à partir de la position 2.

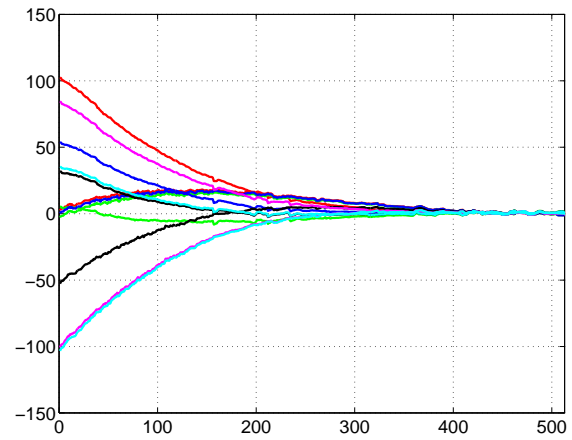
La loi de commande est une fois de plus stable et la convergence est assurée dans tous les cas. On remarque également, et notamment à la convergence de la deuxième expérience, une plus grande sensibilité de l'asservissement visuel  $2D\frac{1}{2}$  aux erreurs de mesure (voir l'estimation de la fonction de tâche sur la Figure 5.19).

Par rapport aux expériences précédentes, la trajectoire des points dans les images est vraiment différente à partir de la position 2 (voir les Figures 5.17(c) et 5.17(d) pour l'asservissement visuel 2D, et les Figures 5.18(c) et 5.18(d) pour l'asservissement visuel  $2D\frac{1}{2}$ ). Cela est dû au fait que le robot n'a pas pu se repositionner très précisément à la position 2 et que les erreurs de calibration interviennent de manière différente sur les deux schémas d'asservissement, comme on l'a déjà vu dans le cas de l'asservissement mono-caméra.

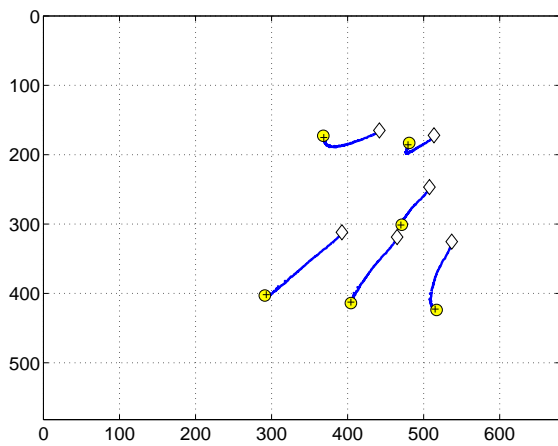
Il faut aussi remarquer que, dans le cas de la position 1 qui représente un déplacement faible des caméras, on obtient quasiment les mêmes trajectoires avec une calibration grossière ou médiocre (comparer les Figures 5.8(c) et 5.8(d) avec les Figures 5.14(c) et 5.14(d) pour l'asservissement visuel 2D, et comparer les Figures 5.9(c) et 5.9(d) avec les Figures 5.15(c) et 5.15(d) pour l'asservissement visuel  $2D\frac{1}{2}$ ). On peut donc conclure que les deux schémas d'asservissement présentent une bonne robustesse aux erreurs de calibration près de la convergence.



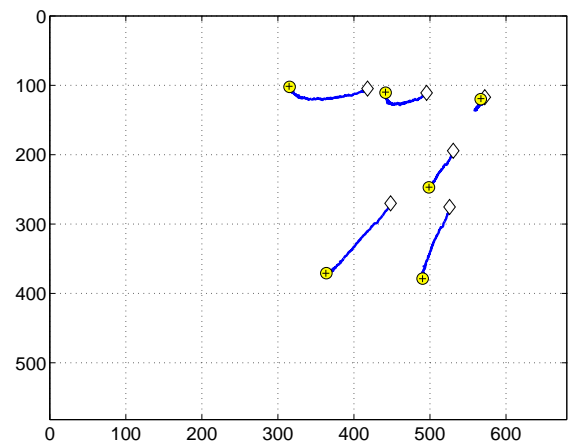
(a) erreurs sur les coordonnées de gauche



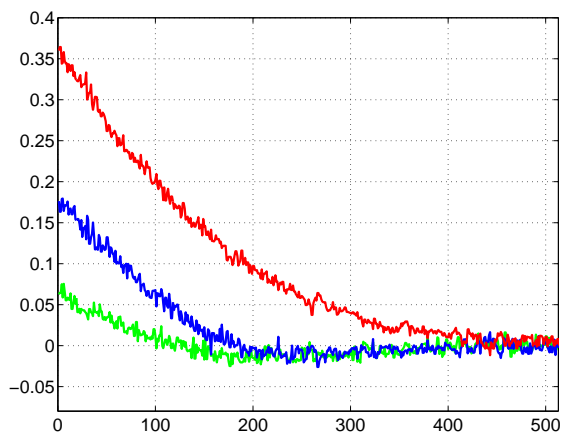
(b) erreurs sur les coordonnées de droite



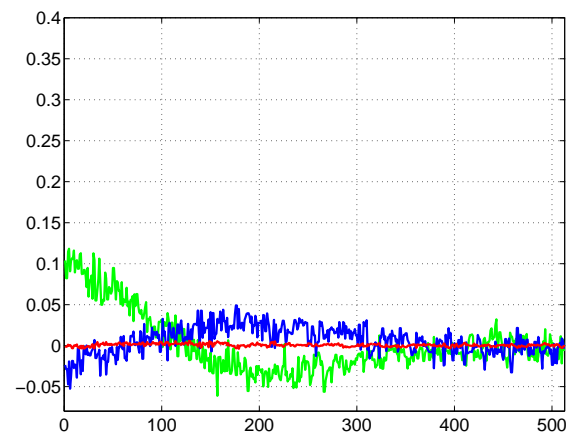
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

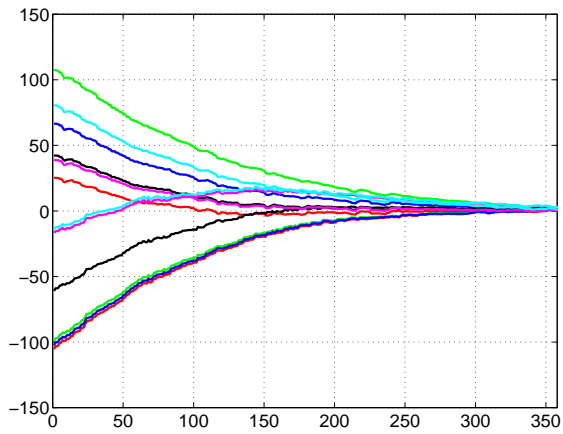


(e) vitesse en translation (cm/s)

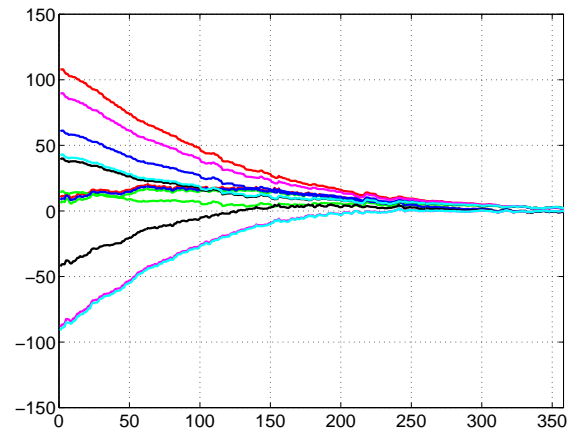


(f) vitesse en rotation (deg/s)

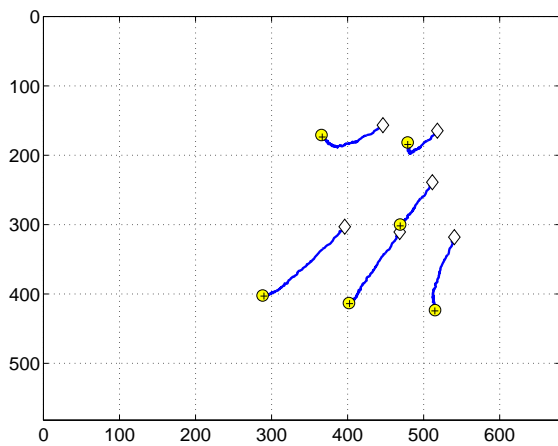
FIG. 5.14 – Asservissement visuel 2D avec calibration médiocre : Première expérience



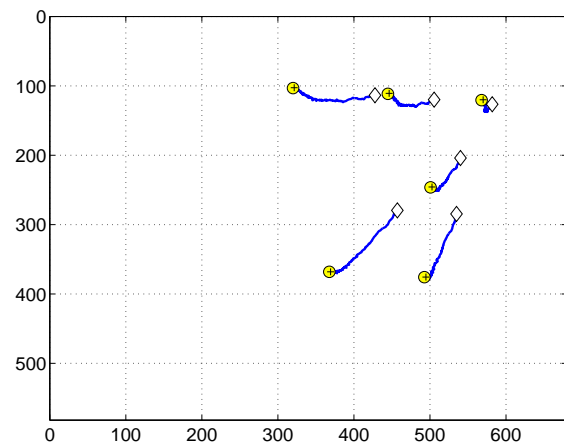
(a) erreurs sur les coordonnées de gauche



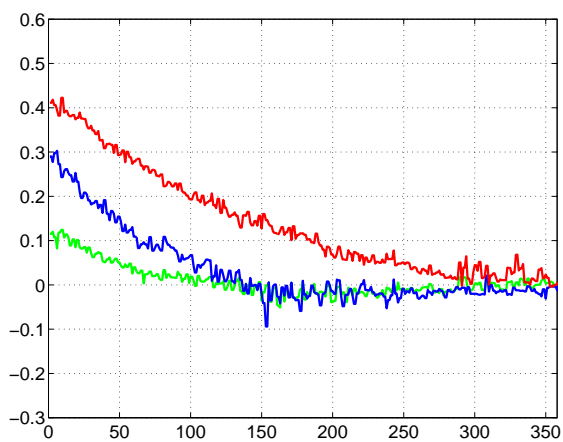
(b) erreurs sur les coordonnées de droite



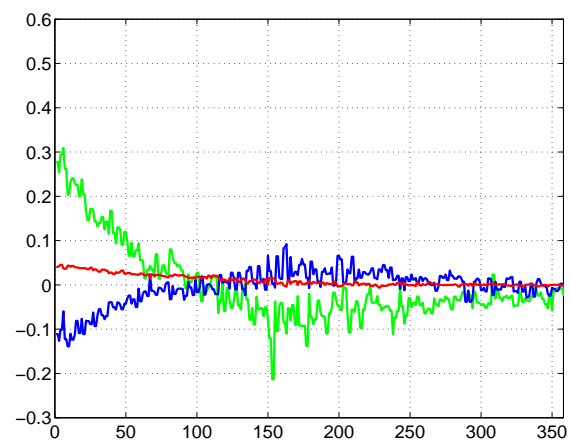
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

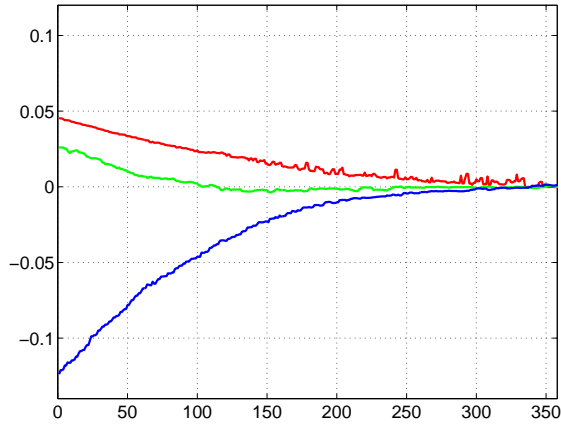


(e) vitesse en translation (cm/s)

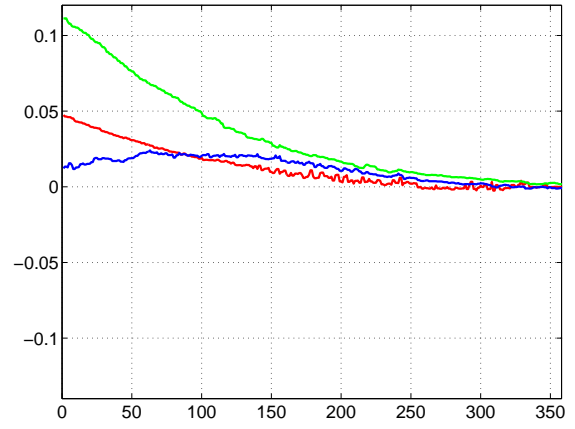


(f) vitesse en rotation (deg/s)

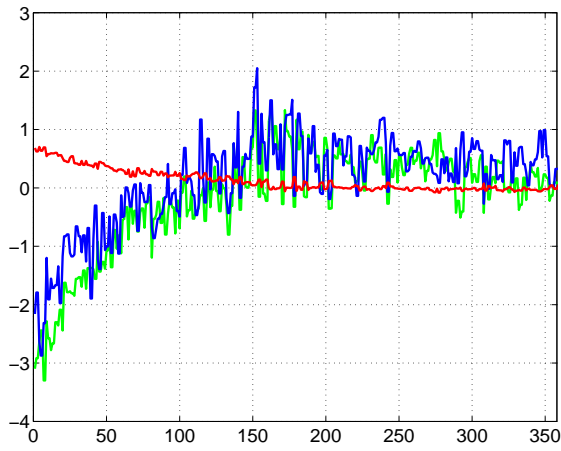
FIG. 5.15 – Asservissement visuel  $2D\frac{1}{2}$  avec calibration médiocre : Première expérience



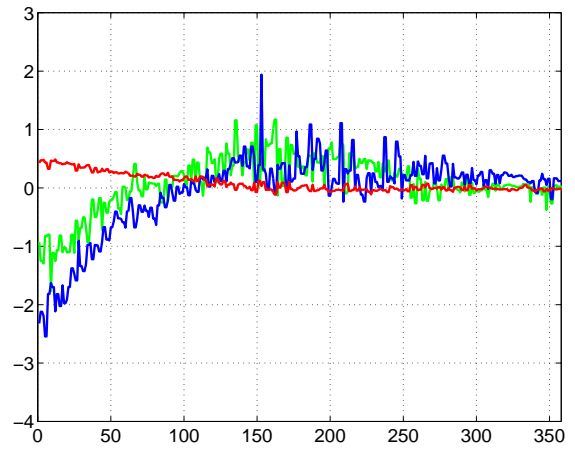
(a) coordonnées étendues de l'image gauche



(b) coordonnées étendues de l'image droite

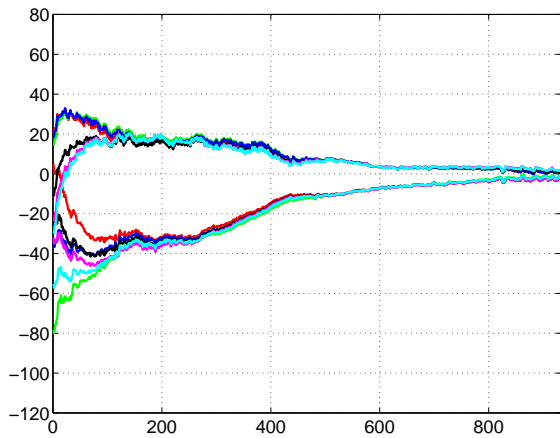


(c) rotation de la caméra gauche

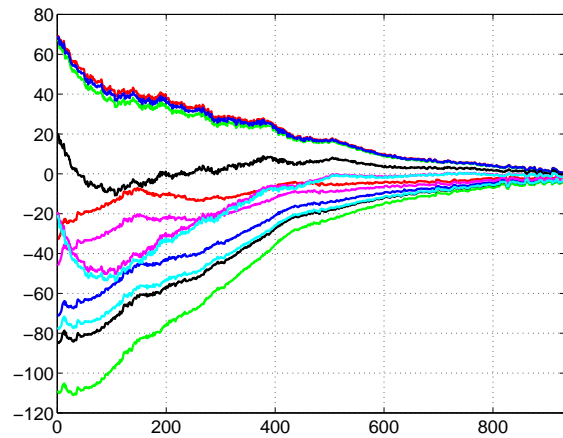


(d) rotation de la caméra droite

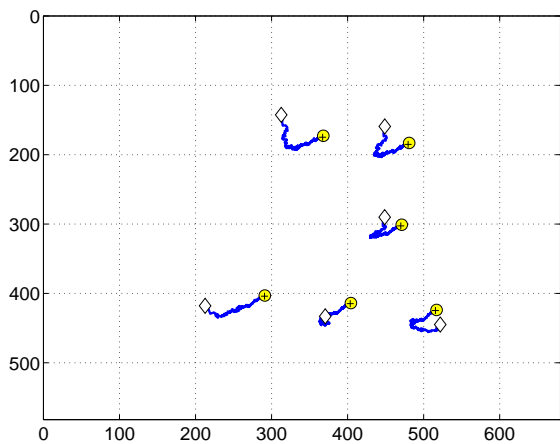
FIG. 5.16 – *Asservissement visuel  $2D\frac{1}{2}$  avec calibration médiocre : Première expérience*



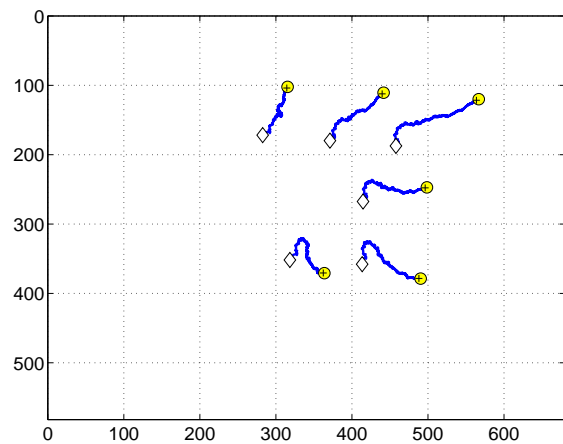
(a) erreurs sur les coordonnées de gauche



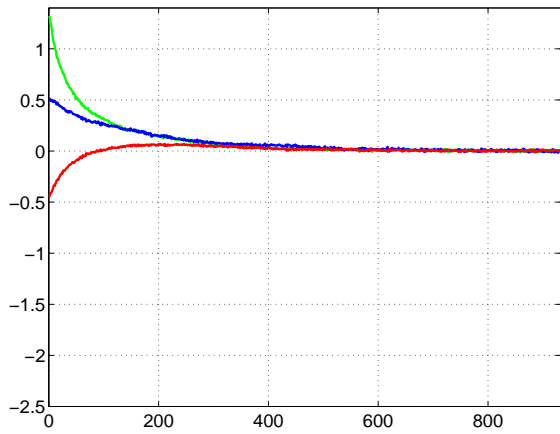
(b) erreurs sur les coordonnées de droite



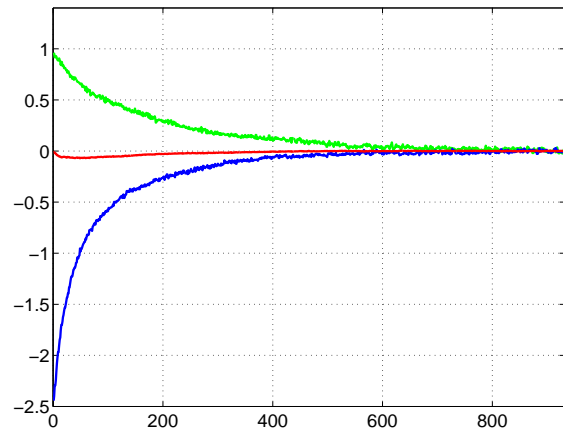
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

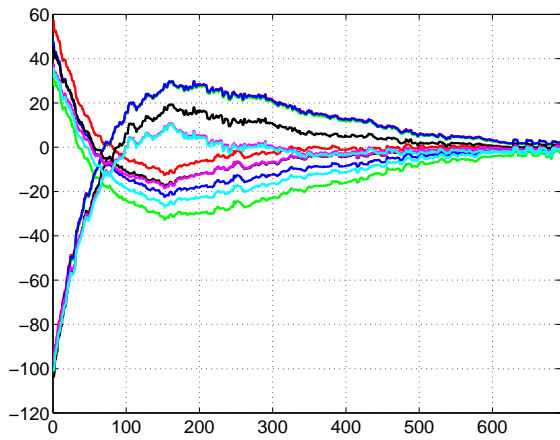


(e) vitesse en translation (cm/s)

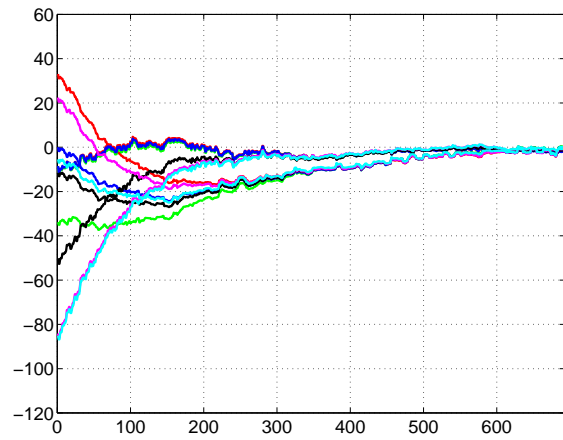


(f) vitesse en rotation (deg/s)

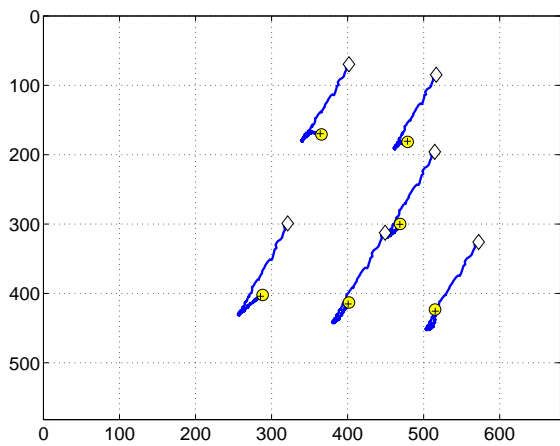
FIG. 5.17 – Asservissement visuel 2D avec calibration médiocre : deuxième expérience



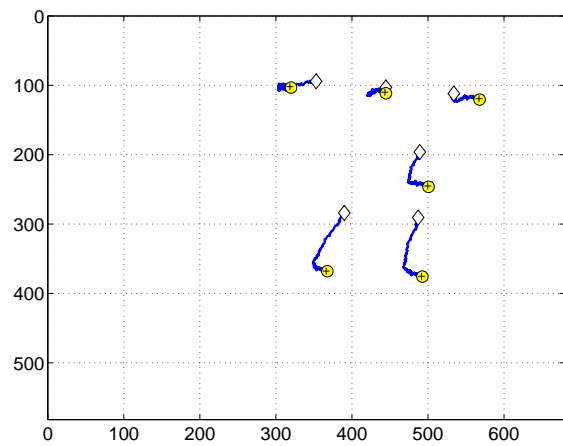
(a) erreurs sur les coordonnées de gauche



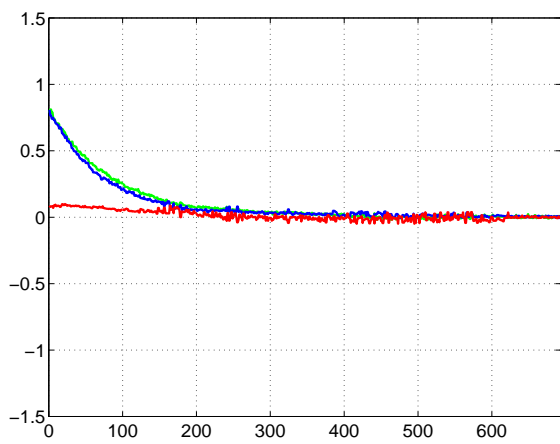
(b) erreurs sur les coordonnées de droite



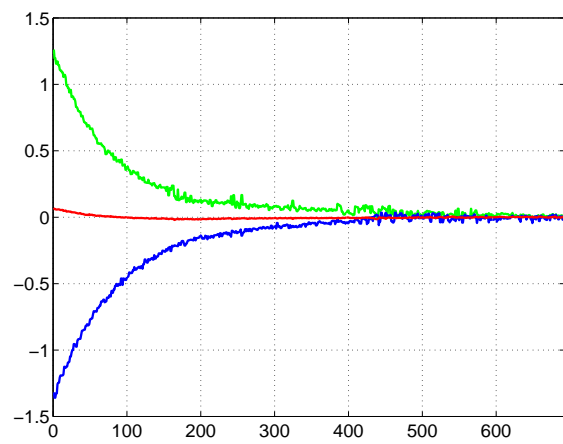
(c) trajectoire dans l'image de gauche



(d) trajectoire dans l'image de droite

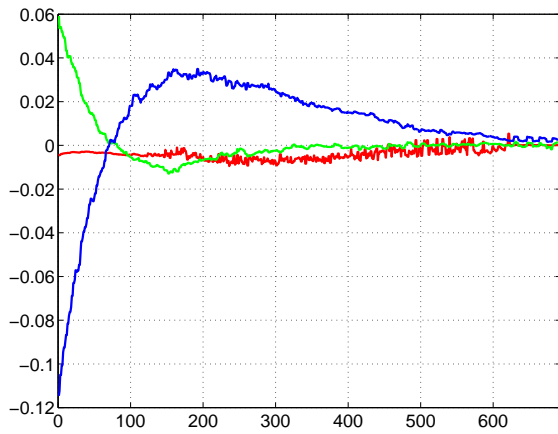


(e) vitesse en translation (cm/s)

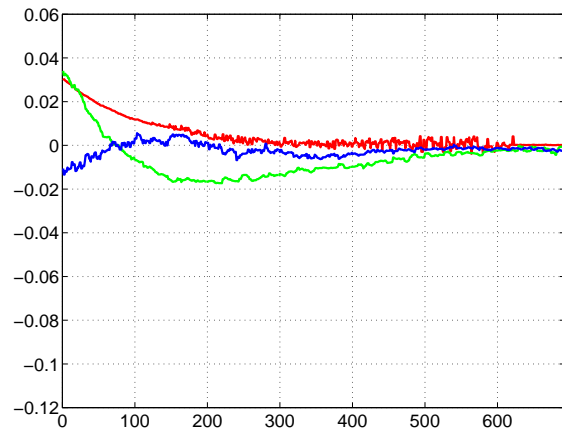


(f) vitesse en rotation (deg/s)

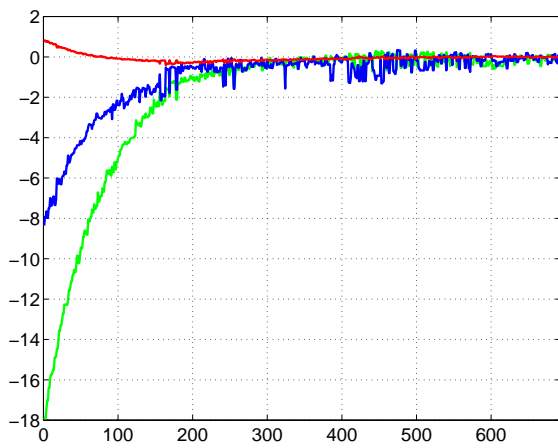
FIG. 5.18 – Asservissement visuel  $2D_2^1$  avec calibration médiocre : deuxième expérience



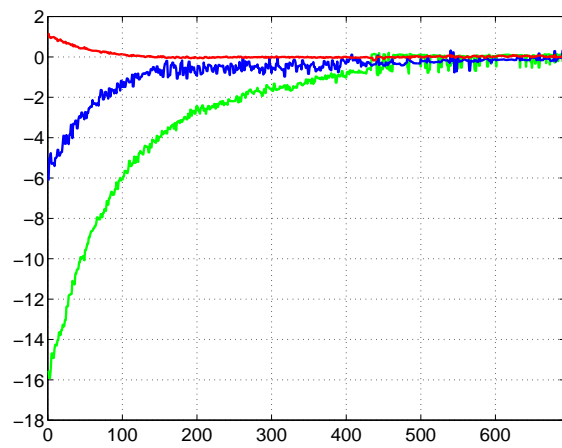
(a) coordonnées étendues de l'image gauche



(b) coordonnées étendues de l'image droite



(c) rotation de la caméra gauche



(d) rotation de la caméra droite

FIG. 5.19 – Asservissement visuel  $2D_{\frac{1}{2}}$  avec calibration médiocre : deuxième expérience



## 5.4 Conclusion

Dans ce chapitre, nous avons proposé une méthode d'asservissement visuel multi-caméras, inspirée de la commande référencée capteur et basée sur l'approche de la fonction de tâche. Notre schéma d'asservissement utilise plusieurs caméras dont les champs de vision ne se recouvrent pas. Dans ce cas, il n'est pas possible d'effectuer une reconstruction stéréo de l'objet observé et la fusion des données capteurs doit être effectuée à un niveau plus élevé.

Le choix particulier de la fonction de tâche nous a permis de simplifier la conception de la commande et l'analyse de la stabilité. En effet, on peut concevoir une fonction de tâche pour chaque caméra et construire une fonction de tâche globale qui n'est rien d'autre que la moyenne pondérée des précédentes. Par conséquent, la loi de commande peut être considérée comme une moyenne pondérée des lois de commande associées à chaque caméra. On montre que la stabilité du système global est assurée par la stabilité de chaque sous-système. L'analyse de la robustesse de la stabilité peut être alors simplifiée et reconduite à celle d'un système mono-caméra.

Enfin, les résultats expérimentaux, obtenus en utilisant un robot manipulateur avec deux caméras embarquées, montrent qu'il est possible d'améliorer la précision de positionnement de l'effecteur du robot par rapport à l'utilisation d'une seule caméra. Les expériences, conduites aussi bien avec l'asservissement visuel 2D que  $2D\frac{1}{2}$ , confirment la validité et la généralité de notre méthode, ainsi que la possibilité de l'étendre si besoin aux schémas  $2D\frac{3}{4}$  et 3D.

# Conclusion et perspectives

Le travail que nous venons de présenter a essayé de répondre aux problèmes théoriques et pratiques énoncés en introduction. Il a notamment porté sur la conception de méthodes qui ne nécessitent ni la connaissance du modèle de l'objet observé, ni une calibration précise du système. Les contributions principales de notre travail se situent à différents niveaux :

**Reconstruction projective et euclidienne :** après avoir étudié les différentes techniques de reconstruction projective et euclidienne, à partir de deux images d'un objet inconnu, nous avons mis en évidence que celles qui estiment une matrice d'homographie, afin d'en extraire le déplacement partiel de la caméra et la structure de la scène (à un facteur d'échelle près), sont les mieux adaptées à être intégrées dans les schémas d'asservissement visuel. Toutefois, le problème de l'estimation d'une matrice d'homographie est non linéaire si nous n'avons aucune connaissance a priori sur la structure de la cible (c'est-à-dire si l'objet n'est pas plan). Nous avons alors proposé, après avoir effectué une première linéarisation du problème, d'imposer les contraintes non-linéaires a posteriori en effectuant une deuxième estimation linéaire. Ceci nous a permis d'obtenir un algorithme assez rapide pour réaliser une boucle de commande à la cadence vidéo et fournissant des résultats robustes, même dans les cas dégénérés.

**Modélisation :** en ce qui concerne l'asservissement visuel mono-caméra, nous avons proposé deux nouvelles méthodes d'asservissement visuel qui ne nécessitent pas la connaissance des modèles 3D des objets observés. Ces méthodes sont basées sur la reconstruction projective puis euclidienne du déplacement partiel de la caméra et de la structure (à un facteur d'échelle près) de l'objet observé. Un choix judicieux de la fonction de tâche pour chaque méthode nous a permis d'obtenir un jacobien de la tâche triangulaire par bloc. En effet, la rotation est directement obtenue à partir de la reconstruction, et la boucle de commande en rotation est donc découplée de celle en translation. Cette dernière est contrôlée, dans le cas du  $2D\frac{1}{2}$ , en utilisant les coordonnées étendues dans l'image d'un point caractéristique et, dans le cas du  $2D\frac{3}{4}$ , en utilisant les coordonnées cartésiennes réduites. La forme très simple du jacobien de la tâche simplifie considérablement l'élaboration de la commande et l'analyse de la stabilité.

En ce qui concerne l'asservissement visuel multi-caméras, nous avons modélisé la fonction de tâche comme une moyenne pondérée des fonctions de tâches associées à chaque caméra. Ceci nous a permis de simplifier l'élaboration de la loi de commande et l'analyse de la stabilité.

**Commande et analyse de la stabilité :** en ce qui concerne l'asservissement visuel mono-caméra, après avoir utilisé une loi de commande proportionnelle classique afin de réguler la fonction de tâche à zéro, nous avons proposé une commande proportionnelle robuste afin de mieux exploiter les caractéristiques particulières du système (c'est-à-dire la rotation découplée de la translation).

Dans le premier cas, nous avons démontré de manière analytique la robustesse de la loi de commande aux erreurs sur les paramètres du système et nous avons obtenu :

- les conditions nécessaires et suffisantes pour la stabilité asymptotique locale, et des conditions suffisantes pour la stabilité asymptotique globale en présence d'erreurs sur les paramètres de la caméra ;
- des conditions suffisantes pour la stabilité asymptotique locale en présence d'erreurs sur les paramètres de la caméra et sur la transformation caméra/effecteur.

Dans le deuxième cas, le domaine de robustesse a pu être étendu considérablement et nous avons obtenu :

- les conditions nécessaires et suffisantes pour la stabilité asymptotique globale en présence d'erreurs sur les paramètres de la caméra ;
- des conditions suffisantes pour la stabilité asymptotique globale en présence d'erreurs sur les paramètres de la caméra et sur la transformation caméra/effecteur.

En ce qui concerne l'asservissement visuel multi-caméras, le choix particulier de la fonction de tâche nous a permis de simplifier la conception de la commande et l'analyse de la stabilité. En effet, l'étude de la stabilité du système global se réduit à l'étude de la stabilité de chaque sous-système. On peut donc utiliser les résultats obtenus en asservissement visuel mono-caméra afin de déduire de bonnes propriétés de robustesse de l'asservissement visuel multi-caméras.

**Expérimentations :** nous avons appliqué les considérations théoriques effectuées dans notre travail à des exemples concrets de positionnement de l'effecteur d'un robot manipulateur. Nous avons testé nos schémas d'asservissement sur un robot cartésien (à l'IRISA) et un robot redondant (à la DER EDF Chatou). Les résultats obtenus confirment que nos schémas d'asservissement apportent une amélioration importante du domaine de convergence.

Les résultats expérimentaux, obtenus en utilisant un robot manipulateur avec deux caméras embarquées, montrent qu'il est possible d'améliorer la précision de positionnement de l'effecteur du robot par rapport à l'utilisation d'une seule caméra. Les expériences réalisées aussi bien avec l'asservissement visuel 2D que  $2D\frac{1}{2}$  confirment la validité et la généralité de notre méthode d'asservissement multi-caméras.

Un grand nombre de problèmes restent cependant encore ouverts, et de nouveaux se présentent. Ainsi, nous proposons les extensions et les perspectives de recherche suivantes :

- afin de rendre notre schéma d’asservissement réellement indépendant de la connaissance de la structure des objets observés, il serait intéressant d’étudier les techniques de mise en correspondance automatique. Comme la mise en correspondance doit être effectuée une fois pour toute au début de l’asservissement, on pourrait éventuellement intégrer des techniques non linéaires efficaces (sous certaines hypothèses) mais généralement très coûteuses en temps de calcul. Parmi les méthodes existantes, on peut retenir les techniques de mise en correspondance par fenêtre de corrélation [Zhang 94] qui peuvent être rendues robustes aux occultations et aux variations d’illumination de la scène [Lan 97]. Toutefois, elles ne sont pas très bien adaptées pour de grandes rotations de la caméra. Une autre technique, plus robuste aux grandes rotations, utilise les images fovéales (invariantes en rotation) [Allezard 97]. Enfin, la méthode qui nous semble la mieux adaptée à notre besoin est celle proposée par [Schmid 97]. Avec cette méthode, utilisant des invariants projectifs, on pourrait effectuer une mise en correspondance à la fois robuste aux occultations, aux variations d’illumination et aux grands déplacements de la caméra.
- même si, à l’état actuel, on ne peut pas avoir une mise en correspondance automatique robuste en cas de grands déplacements de la caméra, on pourrait s’intéresser à l’extension de l’estimation de la matrice d’homographie au cas où des objets réels sont observés par la caméra. En effet, dans ce cas, on peut s’attendre à un plus grand niveau de bruit dans les images et à de fausses mises en correspondance. Il faut donc rendre plus robuste l’estimation de la matrice d’homographie :
  - soit en utilisant un algorithme d’estimation non linéaire, mais dans ce cas on tombe sur le problème de respecter la cadence vidéo.
  - soit en utilisant les informations de plusieurs images acquises pendant l’asservissement.
- le problème de l’estimation de la matrice d’homographie est lié à celui de l’estimation de la matrice d’homographie à l’infini. La possibilité d’estimer de manière simple cette matrice permet non seulement de valider définitivement toutes les démonstrations théoriques sur la stabilité de notre schéma d’asservissement visuel, mais ouvre aussi des perspectives intéressantes pour estimer les paramètres intrinsèques de la caméra [Vieville 96] [Zeller 96a]. De plus, l’auto-calibration en ligne des paramètres intrinsèques de la caméra pourrait être accompagnée par l’auto-calibration de la matrice de passage caméra/effecteur, et l’estimation de la profondeur de l’objet.
- une application très intéressante de l’estimation directe de la matrice d’homographie que nous avons proposée est celle de la détection d’obstacles. En effet, on pourrait intégrer dans notre schéma de contrôle le système proposé par [Zeller 96b] qui nécessite le calcul de la matrice d’homographie relative à un plan virtuel. Puisque le modèle des objets observés n’est pas nécessaire pour notre méthode d’asservissement visuel, on

peut alors s'intéresser à la détection d'obstacles et à la conception d'une loi de commande de plus haut niveau pour leur évitement.

- une autre perspective très intéressante est celle de la génération et de la planification de trajectoire. Dans le cas des asservissements visuels  $2D\frac{1}{2}$  et  $2D\frac{3}{4}$ , ce problème est simplifié par le fait que seule la trajectoire d'un point doit être spécifiée dans l'image. Dans le cas de l'asservissement 2D par contre, il est nécessaire de spécifier la position d'au moins quatre points de la cible de telle manière que la position relative entre les points corresponde à une réalité physique. À partir de deux vues d'un objet, il est possible de reconstruire la vue de ce même objet par une troisième caméra virtuelle [Faugeras 98]. Bien évidemment, on ne peut reconstruire que les parties visibles dans les deux premières images. Afin de résoudre le problème de la reconstruction d'une troisième vue à partir de deux images, il est nécessaire de fixer d'une manière ou d'une autre la "position" de la troisième caméra. Ceci peut être fait de deux manières différentes :
  - si la calibration est connue, la troisième vue peut être déterminée en fixant le repère caméra associé à la troisième vue.
  - si la calibration de la caméra est inconnue, la troisième vue peut être déterminée en fixant un certain nombre de points dans l'image (par exemple les épipoles relatifs aux deux premières caméras).

Ceci pourrait nous permettre de :

- planifier les trajectoires des informations visuelles dans l'image en générant  $\mathbf{s}^*(t)$ . On pourrait donc échantillonner la trajectoire dans l'image de telle manière à travailler aux "petites erreurs" ;
- planifier la vue suivante en fonction du conditionnement de la matrice d'interaction (évitement des singularités ou des minima locaux) ;
- prévoir à l'avance si un objet va sortir de l'image lors de l'asservissement visuel.

En ce qui concerne l'évitement d'obstacles, la relation d'ordre de la géométrie épipolaire (qui dit simplement si un point dans les deux premières vues est également visible dans la troisième vue virtuelle) pourrait nous permettre :

- d'éviter les auto-occultations de l'objet sur lui même ;
  - d'éviter les occultations d'un éventuel obstacle.
- afin de prendre définitivement en compte, dans la boucle de commande, la condition de visibilité de la tâche, il serait intéressant de compléter la loi de commande proportionnelle robuste de l'asservissement visuel  $2D\frac{1}{2}$ . En effet, il existe des cas particuliers qui ne sont pas encore traités par la loi commande telle que nous l'avons définie. On peut également envisager une commande robuste adaptative pour les méthodes 3D qui offrent un découplage entre rotation et translation.

- le seul inconvénient de l’asservissement visuel  $2D\frac{1}{2}$  est qu’il n’est pas le plus robuste près de la convergence. On pourrait alors définir un schéma d’asservissement convergent de n’importe quelle position et peu sensible au bruit par un enchaînement de tâches : la première tâche consisterait à amener la caméra dans un voisinage très proche de la cible ( $2D\frac{1}{2}$  avec une commande robuste) et la deuxième tâche consisterait à achever le positionnement avec une très grande précision ( $2D$  avec matrice d’interaction constante). En outre, afin d’améliorer le comportement dynamique de l’asservissement  $2D$  et de faciliter l’analyse de la stabilité, on peut utiliser la reconstruction projective pour définir les distances relatives des points de l’objet par rapport à un plan de référence. La distance de la caméra par rapport à ce plan demeure alors la seule inconnue, et aucune connaissance a priori sur le modèle de l’objet n’est plus nécessaire.



# Annexe A

## Représentation des rotations

### A.1 Définitions et propriétés

Il est bien connu qu'une rotation d'angle  $\theta$  autour d'un axe  $\mathbf{u}$  peut être représentée par une matrice de rotation  $\mathbf{R}(\mathbf{u}, \theta)$ . Cette matrice peut s'exprimer en fonction de  $\mathbf{u}$  et de  $\theta$  de différentes manières :

– sous forme exponentielle :

$$\mathbf{R}(\mathbf{u}\theta) = e^{[\mathbf{u}\theta]_{\times}} = \sum_{k=0}^{\infty} \frac{[\mathbf{u}\theta]_{\times}^k}{k!} \quad (\text{A.1})$$

où la matrice de pré-produit vectoriel  $[\mathbf{u}]_{\times}$  associée au vecteur  $\mathbf{u}$  est définie de la manière suivante :

$$[\mathbf{u}]_{\times} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \quad (\text{A.2})$$

– sous la forme due à Rodriguez :

$$\mathbf{R}(\mathbf{u}, \theta) = \mathbf{I} + \sin(\theta) [\mathbf{u}]_{\times} + (1 - \cos(\theta)) [\mathbf{u}]_{\times}^2 \quad (\text{A.3})$$

Cette dernière forme est la plus connue et la plus facile à manipuler. Nous allons voir à présent quelques propriétés qui nous seront utiles par la suite.

L'axe de rotation  $\mathbf{u} = [u_x \ u_y \ u_z]^T$  est choisi unitaire :

$$\mathbf{u}^T \mathbf{u} = 1 \quad (\text{A.4})$$

Par conséquent, on a :

$$\mathbf{u}^T \dot{\mathbf{u}} = 0 \quad (\text{A.5})$$



D'après la définition des matrices de préproduit vectoriel, on a :

$$[\mathbf{u}]_{\times} [\mathbf{v}]_{\times} = \mathbf{v} \mathbf{u}^T - (\mathbf{u}^T \mathbf{v}) \mathbf{I} \quad (\text{A.6})$$

Par conséquent :

– si  $\mathbf{v} = \mathbf{u}$ , grâce à la propriété (A.4), on a :

$$[\mathbf{u}]_{\times}^2 = \mathbf{u} \mathbf{u}^T - \mathbf{I} \quad (\text{A.7})$$

– si  $\mathbf{v} = \dot{\mathbf{u}}$ , grâce à la propriété (A.5), on a :

$$[\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} = \dot{\mathbf{u}} \mathbf{u}^T \quad (\text{A.8})$$

Une autre propriété est que :

$$[\mathbf{u}]_{\times} [\mathbf{v}]_{\times} - [\mathbf{v}]_{\times} [\mathbf{u}]_{\times} = \mathbf{v} \mathbf{u}^T - \mathbf{u} \mathbf{v}^T = [[\mathbf{u}]_{\times} \mathbf{v}]_{\times} \quad (\text{A.9})$$

De plus, en multipliant à droite la propriété (A.6) par  $[\mathbf{u}]_{\times}$ , on obtient :

$$[\mathbf{u}]_{\times} [\mathbf{v}]_{\times} [\mathbf{u}]_{\times} = -(\mathbf{u}^T \mathbf{v}) [\mathbf{u}]_{\times} \quad (\text{A.10})$$

car  $\mathbf{u}^T [\mathbf{u}]_{\times} = 0$ . Par conséquent :

– si  $\mathbf{v} = \mathbf{u}$ , on peut en déduire que les puissances entières de la matrice de préproduit vectoriel satisfont les propriétés suivantes :

$$[\mathbf{u}]_{\times}^{2k+1} = (-1)^k [\mathbf{u}]_{\times} \quad (\text{A.11})$$

$$[\mathbf{u}]_{\times}^{2k+2} = (-1)^k [\mathbf{u}]_{\times}^2 \quad (\text{A.12})$$

– si  $\mathbf{v} = \dot{\mathbf{u}}$ , on a :

$$[\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times} = 0 \quad (\text{A.13})$$

On peut, enfin, écrire une propriété très intéressante en multipliant à droite par  $\dot{\mathbf{u}}$  l'équation (A.7) et en tenant compte de l'équation (A.5) :

$$\dot{\mathbf{u}} = -[\mathbf{u}]_{\times}^2 \dot{\mathbf{u}} \quad (\text{A.14})$$

## A.2 Choix de la représentation de la rotation

Nous avons choisi comme représentation de la rotation le vecteur  $\mathbf{u}\theta$ , car nous verrons qu'elle est très intéressante par rapport à celles couramment employées. En effet, les représentations  $\mathbf{u}\sin(\theta)$  et  $\mathbf{u}\sin(\frac{\theta}{2})$  présentent respectivement une singularité de leur jacobien en  $\frac{\pi}{2}$  et  $\pi$ , alors que la représentation  $\mathbf{u}\theta$  implique seulement une singularité en  $2\pi$ .

Comme  $\cos(\theta) = \frac{1}{2}(\text{trace}(\mathbf{R}) - 1)$ , l'angle de rotation  $\theta$  peut être calculé à partir des éléments de la matrice de rotation :

$$\theta = \arccos\left(\frac{1}{2}(r_{11} + r_{22} + r_{33} - 1)\right) \quad (\text{A.15})$$

A la première itération de l'asservissement, on choisit  $\theta$  tel que  $0 \leq \theta \leq \pi$ . Pour les itérations suivantes, l'angle de rotation  $\theta$  est libre d'évoluer dans le domaine  $[0 ; 2\pi[$  afin d'éviter toute discontinuité de la représentation. Pour cela, on choisit la valeur de  $\theta$  tel que l'axe  $\mathbf{u}$  soit le plus proche de sa valeur précédente.

En sachant que  $[\mathbf{u}]_{\times} \sin(\theta) = \frac{\mathbf{R} - \mathbf{R}^T}{2}$ , le vecteur  $\mathbf{u}\theta$  peut être calculé comme suit :

$$\mathbf{u}\theta = \frac{1}{2} \frac{1}{\text{sinc}(\theta)} \begin{bmatrix} r_{32} - r_{23} \\ r_{31} - r_{13} \\ r_{21} - r_{12} \end{bmatrix} \quad \text{si} \quad \begin{cases} 0 \leq \theta < \pi \\ \pi < \theta < 2\pi \end{cases} \quad (\text{A.16})$$

où la fonction  $\text{sinc}(\theta)$ , définie par  $\sin(\theta) = \theta \text{sinc}(\theta)$ , est appelée le sinus cardinal. Cette fonction, illustrée sur la Figure A.1, est  $C^\infty$  et s'annule pour  $\theta = (2n + 1)\pi$  avec  $n \in \mathbb{Z}$ .

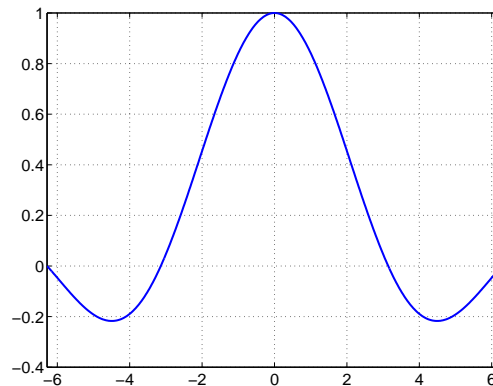


FIG. A.1 – *Fonction sinus cardinal*

L'équation (A.16) n'est pas utilisable pour  $\theta = \pi$  car on a alors  $\text{sinc}(\theta) = 0$ . Ce cas peut néanmoins être détecté (puisque nous connaissons la valeur de  $\theta$ ) et l'axe de rotation  $\mathbf{u}$  est alors le vecteur propre de la matrice de rotation  $\mathbf{R}$  associé à la valeur propre  $\lambda = 1$ .

On peut ainsi déterminer de manière unique la représentation  $\mathbf{u}\theta$  à partir de la matrice de rotation  $\mathbf{R}$ . Par ailleurs, la dérivée du vecteur  $\mathbf{u}\theta$  par rapport au temps s'écrit :

$$\frac{d(\mathbf{u}\theta)}{dt} = \mathbf{u}\dot{\theta} + \dot{\mathbf{u}}\theta \quad (\text{A.17})$$

Grâce aux propriétés montrées dans la section précédente, on obtient :

– en multipliant l'équation (A.17) par  $\mathbf{I} + [\mathbf{u}]_{\times}^2$  :

$$\mathbf{u}\dot{\theta} = (\mathbf{I} + [\mathbf{u}]_{\times}^2) \frac{d(\mathbf{u}\theta)}{dt} \quad (\text{A.18})$$

– en multipliant l'équation (A.17) par  $-[\mathbf{u}]_{\times}^2$  :

$$\dot{\mathbf{u}}\theta = -[\mathbf{u}]_{\times}^2 \frac{d(\mathbf{u}\theta)}{dt} \quad (\text{A.19})$$

Nous cherchons maintenant à exprimer la dérivée de  $\mathbf{u}\theta$  en fonction de la vitesse de rotation  $\omega$ .

### A.3 Calcul de $\omega$

Par définition, on a :

$$[\omega]_{\times} = \dot{\mathbf{R}}(\mathbf{u}, \theta) \mathbf{R}^T(\mathbf{u}, \theta) \quad (\text{A.20})$$

Comme il est assez complexe d'obtenir  $\omega$  à partir de la représentation exponentielle des matrices de rotation, nous allons l'obtenir à partir de la forme de Rodrigue, pour ensuite revenir à une représentation en fonction du vecteur  $\mathbf{u}\theta$ .

La dérivée de la matrice de rotation est donnée par :

$$\begin{aligned} \frac{d\mathbf{R}(\mathbf{u}, \theta)}{dt} &= \cos(\theta)\dot{\theta} [\mathbf{u}]_{\times} + \sin(\theta) [\dot{\mathbf{u}}]_{\times} + \\ &+ \sin(\theta)\dot{\theta} [\mathbf{u}]_{\times}^2 + (1 - \cos(\theta)) [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times} + (1 - \cos(\theta)) [\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} \end{aligned} \quad (\text{A.21})$$

On rappelle que la transposée de la matrice de rotation est :

$$\mathbf{R}^T(\mathbf{u}, \theta) = \mathbf{I} - \sin(\theta) [\mathbf{u}]_{\times} + (1 - \cos(\theta)) [\mathbf{u}]_{\times}^2 \quad (\text{A.22})$$

L'équation (A.20) se développe donc sous la forme suivante :

$$\begin{aligned} [\omega]_{\times} &= \sin(\theta) [\dot{\mathbf{u}}]_{\times} \\ &+ (1 - \cos(\theta)) [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times} + (1 - \cos(\theta)) [\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} \\ &- \sin^2(\theta) [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times} + (1 - \cos(\theta))^2 [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times}^3 \\ &- \sin(\theta)(1 - \cos(\theta)) [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times}^2 + \sin(\theta)(1 - \cos(\theta)) [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times}^2 \\ &+ \cos(\theta)\dot{\theta} [\mathbf{u}]_{\times} + \cos(\theta)(1 - \cos(\theta))\dot{\theta} [\mathbf{u}]_{\times}^3 - \sin^2(\theta)\dot{\theta} [\mathbf{u}]_{\times}^3 \\ &+ \sin(\theta)\dot{\theta} [\mathbf{u}]_{\times}^2 + \sin(\theta)(1 - \cos(\theta))\dot{\theta} [\mathbf{u}]_{\times}^4 - \cos(\theta)\sin(\theta)\dot{\theta} [\mathbf{u}]_{\times}^2 \\ &+ (1 - \cos(\theta))^2 [\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times}^2 - \sin(\theta)(1 - \cos(\theta)) [\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times} \end{aligned} \quad (\text{A.23})$$

Les termes contenant  $[\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times}$  sont nuls d'après la propriété (A.13). On trouve ensuite, grâce aux propriétés énoncées dans le premier paragraphe :

$$\begin{aligned}
[\boldsymbol{\omega}]_{\times} &= \sin(\theta) [\dot{\mathbf{u}}]_{\times} \\
&+ (1 - \cos(\theta)) [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times} + (1 - \cos(\theta)) [\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} \\
&- \sin^2(\theta) [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times} - (1 - \cos(\theta))^2 [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times} \\
&+ (\cos(\theta) - \cos(\theta)(1 - \cos(\theta)) + \sin^2(\theta)) [\mathbf{u}]_{\times} \dot{\theta} \\
&+ (\sin(\theta) - \sin(\theta)(1 - \cos(\theta)) - \cos(\theta) \sin(\theta)) [\mathbf{u}]_{\times}^2 \dot{\theta}
\end{aligned} \tag{A.24}$$

Les derniers termes étant nuls, on en déduit :

$$\begin{aligned}
[\boldsymbol{\omega}]_{\times} &= \sin(\theta) [\dot{\mathbf{u}}]_{\times} + [\mathbf{u}]_{\times} \dot{\theta} \\
&+ (1 - \cos(\theta)) [\mathbf{u}]_{\times} [\dot{\mathbf{u}}]_{\times} - (1 - \cos(\theta)) [\dot{\mathbf{u}}]_{\times} [\mathbf{u}]_{\times}
\end{aligned} \tag{A.25}$$

Enfin, grâce à la propriété (A.9), on obtient :

$$[\boldsymbol{\omega}]_{\times} = \sin(\theta) [\dot{\mathbf{u}}]_{\times} + [\mathbf{u}]_{\times} \dot{\theta} + (1 - \cos(\theta)) [[\mathbf{u}]_{\times} \dot{\mathbf{u}}]_{\times} \tag{A.26}$$

Le passage au vecteur  $\boldsymbol{\omega}$  est immédiat :

$$\boldsymbol{\omega} = \mathbf{u} \dot{\theta} + (\mathbf{I} \sin(\theta) + (1 - \cos(\theta)) [\mathbf{u}]_{\times}) \dot{\mathbf{u}} \tag{A.27}$$

On remarque que :

$$(1 - \cos(\theta)) = \frac{\theta^2}{2} \text{sinc}^2\left(\frac{\theta}{2}\right) \tag{A.28}$$

et :

$$\sin(\theta) = \theta \text{sinc}(\theta) \tag{A.29}$$

L'équation (A.27) peut alors s'écrire :

$$\begin{aligned}
\boldsymbol{\omega} &= \mathbf{u} \dot{\theta} + \left( \theta \text{sinc}(\theta) \mathbf{I} + \frac{\theta^2}{2} \text{sinc}^2\left(\frac{\theta}{2}\right) [\mathbf{u}]_{\times} \right) \dot{\mathbf{u}} \\
&= \mathbf{u} \dot{\theta} + \left( \text{sinc}(\theta) \mathbf{I} + \frac{\theta}{2} \text{sinc}^2\left(\frac{\theta}{2}\right) [\mathbf{u}]_{\times} \right) \dot{\mathbf{u}} \theta
\end{aligned} \tag{A.30}$$

En utilisant les propriétés (A.18) et (A.19), on obtient :

$$\begin{aligned}
\boldsymbol{\omega} &= (\mathbf{I} + [\mathbf{u}]_{\times}^2) \frac{d(\mathbf{u}\theta)}{dt} - \left( \text{sinc}(\theta) [\mathbf{u}]_{\times}^2 + \frac{\theta}{2} \text{sinc}^2\left(\frac{\theta}{2}\right) [\mathbf{u}]_{\times}^3 \right) \frac{d(\mathbf{u}\theta)}{dt} = \\
&= \left( \mathbf{I} + \frac{\theta}{2} \text{sinc}^2\left(\frac{\theta}{2}\right) [\mathbf{u}]_{\times} + (1 - \text{sinc}(\theta)) [\mathbf{u}]_{\times}^2 \right) \frac{d(\mathbf{u}\theta)}{dt}
\end{aligned} \tag{A.31}$$

Le jacobien inverse de  $\mathbf{u}\theta$  est donc :

$$\mathbf{L}_{\boldsymbol{\omega}}^{-1}(\mathbf{u}, \theta) = \mathbf{I} + \frac{\theta}{2} \text{sinc}^2\left(\frac{\theta}{2}\right) [\mathbf{u}]_{\times} + (1 - \text{sinc}(\theta)) [\mathbf{u}]_{\times}^2 \tag{A.32}$$

On remarque que l'on peut également obtenir une expression de ce jacobien en fonction du vecteur  $\mathbf{u}\theta$ . En effet, on a :

$$\frac{\theta}{2} \text{sinc}^2\left(\frac{\theta}{2}\right) = \sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k+1}}{(2k+2)!} \quad (\text{A.33})$$

et

$$\text{sinc}(\theta) = \sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k}}{(2k+1)!} \quad (\text{A.34})$$

En utilisant les propriétés (A.11) et (A.12), on obtient :

$$\mathbf{L}_{\omega}^{-1}(\mathbf{u}, \theta) = \mathbf{I} + \sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k+1}}{(2k+2)!} [\mathbf{u}]_{\times} + \sum_{k=1}^{\infty} -\frac{(-1)^k \theta^{2k}}{(2k+1)!} [\mathbf{u}]_{\times}^2 \quad (\text{A.35})$$

soit finalement :

$$\mathbf{L}_{\omega}^{-1}(\mathbf{u}\theta) = \sum_{k=0}^{\infty} \frac{[\mathbf{u}\theta]_{\times}^{2k+1}}{(2k+2)!} + \sum_{k=0}^{\infty} \frac{[\mathbf{u}\theta]_{\times}^{2k}}{(2k+1)!} \quad (\text{A.36})$$

Cette forme s'écrit directement en fonction du vecteur  $\mathbf{u}\theta$ , mais, comme elle n'est pas très pratique à manipuler, nous préférons utiliser la forme donnée par (A.32).

## A.4 Calcul du jacobien

On pose le problème de calculer l'inverse de la matrice :

$$\mathbf{L}_{\omega}^{-1} = (\mathbf{I} + \alpha [\mathbf{u}]_{\times} + \beta [\mathbf{u}]_{\times}^2) \quad (\text{A.37})$$

avec dans notre cas :

$$\begin{cases} \alpha &= \frac{\theta}{2} \text{sinc}^2\left(\frac{\theta}{2}\right) \\ \beta &= 1 - \text{sinc}(\theta) \end{cases} \quad (\text{A.38})$$

Le déterminant de la matrice  $\mathbf{L}_{\omega}^{-1}$  est donné par :

$$\Delta = \det(\mathbf{L}_{\omega}^{-1}) = (1 - \beta)^2 + \alpha^2 = \text{sinc}^2\left(\frac{\theta}{2}\right) \quad (\text{A.39})$$

Il est donc nul pour  $\theta = 2\pi$ , soit une singularité située en dehors de notre espace de travail.

On suppose que l'inverse de  $\mathbf{L}_{\omega}^{-1}$  a la forme :

$$\mathbf{M}^{-1} = (\mathbf{I} + x [\mathbf{u}]_{\times} + y [\mathbf{u}]_{\times}^2) \quad (\text{A.40})$$

On calcule alors :

$$\begin{aligned} \mathbf{M}\mathbf{M}^{-1} &= (\mathbf{I} + \alpha [\mathbf{u}]_{\times} + \beta [\mathbf{u}]_{\times}^2) (\mathbf{I} + x [\mathbf{u}]_{\times} + y [\mathbf{u}]_{\times}^2) \\ &= \mathbf{I} + (x + \alpha) [\mathbf{u}]_{\times} + (y + \beta + \alpha x) [\mathbf{u}]_{\times}^2 + (\beta x + \alpha y) [\mathbf{u}]_{\times}^3 + \beta y [\mathbf{u}]_{\times}^4 \end{aligned} \quad (\text{A.41})$$

et, grâce aux propriétés des matrices antisymétriques, on obtient :

$$\mathbf{M}\mathbf{M}^{-1} = \mathbf{I} + (x - \beta x + \alpha - \alpha y) [\mathbf{u}]_{\times} + (y - \beta y + \alpha x + \beta) [\mathbf{u}]_{\times}^2 \quad (\text{A.42})$$

d'où le système :

$$\begin{bmatrix} 1 - \beta & -\alpha \\ \alpha & 1 - \beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = - \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (\text{A.43})$$

qui a pour solution :

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{-1}{\Delta} \begin{bmatrix} 1 - \beta & \alpha \\ -\alpha & 1 - \beta \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{-1}{\Delta} \begin{bmatrix} \alpha \\ -\alpha^2 + (1 - \beta)\beta \end{bmatrix} \quad (\text{A.44})$$

c'est-à-dire, en tenant compte des équations (A.38) :

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -\frac{\alpha}{\Delta} \\ 1 - \frac{(1 - \beta)}{\Delta} \end{bmatrix} = \begin{bmatrix} -\frac{\theta}{2} \\ 1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})} \end{bmatrix} \quad (\text{A.45})$$

En conclusion, le jacobien lié à notre représentation de la rotation s'écrit ( $\frac{d(\mathbf{u}\theta)}{dt} = \mathbf{L}_{\omega}\boldsymbol{\omega}$ ) :

$$\mathbf{L}_{\omega} = \mathbf{I} - \frac{\theta}{2} [\mathbf{u}]_{\times} + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})}\right) [\mathbf{u}]_{\times}^2 \quad (\text{A.46})$$

On peut bien entendu vérifier que l'on a aussi  $\mathbf{L}_{\omega}\mathbf{L}_{\omega}^{-1} = \mathbf{I}$ ,  $\forall \theta : 0 \leq \theta < 2\pi$ .



# Annexe B

## Analyse de la stabilité de l'asservissement visuel mono-caméra

### B.1 Définitions

On rappelle ici quelques définitions utiles afin de prouver les théorèmes présentés dans cette annexe.

Le rayon spectral d'une matrice  $\mathbf{M}$ , de dimension  $(n \times n)$ , et de valeurs propres  $\lambda_1, \lambda_2, \dots, \lambda_n$  est :

$$\rho(\mathbf{M}) = \max_{1 \leq i \leq n} |\lambda_i| \quad (\text{B.1})$$

La norme  $l_2$  d'une matrice  $\mathbf{M}$ , de dimension  $(n \times n)$ , est définie comme :

$$\|\mathbf{M}\| = \sqrt{\rho(\mathbf{M}^T \mathbf{M})} \quad (\text{B.2})$$

Toute matrice se décompose en la somme d'une matrice antisymétrique ( $\mathbf{A}\mathbf{S} = \frac{1}{2}(\mathbf{M} - \mathbf{M}^T)$ ) et d'une matrice symétrique ( $\mathbf{S} = \frac{1}{2}(\mathbf{M} + \mathbf{M}^T)$ ). Si on considère seulement la partie symétrique  $\mathbf{S}$  de la matrice  $\mathbf{M}$  alors  $\|\mathbf{S}\| = \rho(\mathbf{S})$ . Si la matrice  $\mathbf{S}$  est définie positive, le produit scalaire  $\mathbf{x}^T \mathbf{S} \mathbf{x}$  peut être borné de la manière suivante :

$$\rho^{-1}(\mathbf{S}^{-1}) \|\mathbf{x}\|^2 \leq \mathbf{x}^T \mathbf{S} \mathbf{x} \leq \rho(\mathbf{S}) \|\mathbf{x}\|^2 \quad (\text{B.3})$$

où :

$$\rho^{-1}(\mathbf{S}^{-1}) = \min_{1 \leq i \leq n} |\lambda_i| \quad (\text{B.4})$$





où on a :

$$\mathbf{S}^2 = \left[ \begin{array}{c|c|c|c|c} \sigma_2 \mathbf{I}_2 & \mathbf{v}_2 & & & \\ \mathbf{v}_2^T & \varphi_2 & & & \\ \hline & & \mathbf{v}_3 & & \\ & & & \cdots & \\ \hline & & \mathbf{v}_3^T & \varphi_3 & \\ & & & & \\ \vdots & & & & \\ \hline & & & & \ddots \\ & & & & \\ \hline & & & & \mathbf{v}_{n-1}^T \\ & & & & \varphi_{n-1} \end{array} \right] \quad (\text{B.10})$$

Encore une fois, considérons le mineur principal de  $\mathbf{S}^2$  de dimension  $(3 \times 3)$  :

$$\mathbf{S}_{33}^2 = \begin{bmatrix} \sigma_2 \mathbf{I}_2 & \mathbf{v}_2 \\ \mathbf{v}_2^T & \varphi_2 \end{bmatrix} \quad (\text{B.11})$$

Le produit scalaire  $\mathbf{x}_3^T \mathbf{S}_{33}^2 \mathbf{x}_3$  peut être minoré de la manière suivante :

$$\mathbf{x}_3^T \mathbf{S}_{33}^2 \mathbf{x}_3 \geq \sigma_3 \mathbf{x}_3^T \mathbf{x}_3 \quad \forall \mathbf{x}_3 \in \mathbb{R}^3 : \mathbf{x}_3 \neq 0 \quad (\text{B.12})$$

où  $\sigma_3$  est la plus petite valeur propre de  $\mathbf{S}_{33}^2$  ( $\sigma_3$  doit être positive si on veut que la matrice  $\mathbf{S}_{33}^2$  soit positive). On peut donc écrire que :

$$\mathbf{x}_n^T \mathbf{S}^2 \mathbf{x}_n \geq \mathbf{x}_n^T \mathbf{S}^3 \mathbf{x}_n \quad \forall \mathbf{x}_n \in \mathbb{R}^n : \mathbf{x}_n \neq 0 \quad (\text{B.13})$$

où on a :

$$\mathbf{S}^3 = \left[ \begin{array}{c|c|c|c|c} \sigma_3 \mathbf{I}_3 & \mathbf{v}_3 & & & \\ \mathbf{v}_3^T & \varphi_3 & & & \\ \hline & & \cdots & & \\ & & & \ddots & \\ \hline & & & & \mathbf{v}_{n-1} \\ & & & & \\ \vdots & & & & \\ \hline & & & & \mathbf{v}_{n-1}^T \\ & & & & \varphi_{n-1} \end{array} \right] \quad (\text{B.14})$$

Par récurrence, on arrive à considérer le mineur principal de  $\mathbf{S}^{n-1}$ , de dimension  $(n \times n)$ , qui coïncide avec  $\mathbf{S}^{n-1}$  lui-même :

$$\mathbf{S}_{nn}^{n-1} = \begin{bmatrix} \sigma_{n-1} \mathbf{I}_{n-1} & \mathbf{v}_{n-1} \\ \mathbf{v}_{n-1}^T & \varphi_{n-1} \end{bmatrix} \quad (\text{B.15})$$

Le dernier produit scalaire peut être minoré de la manière suivante :

$$\mathbf{x}_n^T \mathbf{S}_{nn}^{n-1} \mathbf{x}_n \geq \sigma_n \mathbf{x}_n^T \mathbf{x}_n \quad \forall \mathbf{x}_n \in \mathbb{R}^n : \mathbf{x}_n \neq 0 \quad (\text{B.16})$$

où  $\sigma_n$  est la plus petite valeur propre de  $\mathbf{S}_{nn}^{n-1}$  (il est évident que, si  $\sigma_n > 0$ , la matrice  $\mathbf{S}_{nn}^{n-1}$  est positive). On peut donc enfin écrire que :

$$\mathbf{x}_n^T \mathbf{S}^1 \mathbf{x}_n \geq \mathbf{x}_n^T \mathbf{S}^2 \mathbf{x}_n \geq \cdots \geq \mathbf{x}_n^T \mathbf{S}^n \mathbf{x}_n > 0 \quad \forall \mathbf{x}_n \in \mathbb{R}^n : \mathbf{x}_n \neq 0 \quad (\text{B.17})$$

Les conditions nécessaires pour que la sous-matrice générique  $\mathbf{S}_{i+1,i+1}^i$  de dimensions  $(i+1 \times i+1)$  soit positive sont  $\sigma_i > 0$  et  $\varphi_i > 0$ . Il faut maintenant calculer la plus petite valeur propre  $\sigma_{i+1}$  de  $\mathbf{S}_{i+1,i+1}^i$  avec :

$$\mathbf{S}_{i+1,i+1}^i = \begin{bmatrix} \sigma_i \mathbf{I}_i & \mathbf{v}_i \\ \mathbf{v}_i^T & \varphi_i \end{bmatrix} \quad (\text{B.18})$$

Ses  $i+1$  valeurs propres sont données par :

$$\begin{cases} \lambda_{i+1} &= \frac{1}{2}(\sigma_i + \varphi_i + \sqrt{(\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2}) \\ \lambda_k &= \sigma_i \quad \forall k : 1 < k < i+1 \\ \lambda_1 &= \frac{1}{2}(\sigma_i + \varphi_i - \sqrt{(\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2}) \end{cases} \quad (\text{B.19})$$

On montre maintenant que  $\lambda_{i+1} \geq \lambda_k \geq \lambda_1$ . On montre d'abord que  $\lambda_k \geq \lambda_1$ . En effet :

$$\sigma_i \geq \frac{1}{2}(\sigma_i + \varphi_i - \sqrt{(\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2}) \Rightarrow \varphi_i - \sigma_i \leq \sqrt{(\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2} \quad (\text{B.20})$$

Si  $\sigma_i \geq \varphi_i$ , cette inégalité est évidemment vérifiée car la partie gauche de l'inégalité est négative ou nulle. Si  $\sigma_i < \varphi_i$ , on a :

$$(\varphi_i - \sigma_i)^2 \leq (\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2 \Rightarrow 0 \leq 4\|\mathbf{v}_i\|^2 \quad (\text{B.21})$$

On montre maintenant que  $\lambda_{i+1} \geq \lambda_k$ . On a :

$$\frac{1}{2}(\sigma_i + \varphi_i + \sqrt{(\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2}) \geq \sigma_i \Rightarrow \sqrt{(\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2} \geq \varphi_i - \sigma_i \quad (\text{B.22})$$

Ici encore, si  $\sigma_i \geq \varphi_i$ , cette inégalité est évidemment vérifiée. De même, si  $\sigma_i < \varphi_i$ , on a :

$$(\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2 \geq (\varphi_i - \sigma_i)^2 \Rightarrow 4\|\mathbf{v}_i\|^2 \geq 0 \quad (\text{B.23})$$

En conclusion, la valeur propre  $\lambda_1$  est toujours minimale, on a donc :

$$\sigma_{i+1} = \frac{1}{2}(\sigma_i + \varphi_i - \sqrt{(\sigma_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2}) \quad (\text{B.24})$$

La matrice  $\mathbf{S}$  est donc définie positive :

– si et seulement si :

$$\varphi_{i-1} > 0 \quad \forall i = \{1, 2, 3, \dots, n\} \quad (\text{B.25})$$

où  $\varphi_0 = \sigma_1$  est l'élément  $s_{11}$  de  $\mathbf{S}$ .

– et si :

$$\sigma_{i+1} > 0 \quad \forall i = \{1, 2, 3, \dots, n-1\} \quad (\text{B.26})$$

La dernière inégalité peut aussi s'écrire :

$$\sigma_i \varphi_i > \|\mathbf{v}_i\|^2 \quad \forall i = \{1, 2, 3, \dots, n-1\} \quad (\text{B.27})$$

On remarque que, si  $n = 1$  ou  $n = 2$ , les conditions sont exactement les mêmes que celles obtenues avec les autres méthodes (valeurs propres, Sylvester). Si  $n > 2$ , on trouve, avec la méthode qu'on vient de décrire, des conditions assez simples qui utilisent les résultats obtenus à l'itération précédente, mais qui peuvent être plus restrictives que celles données par les autres critères.

Il faut aussi remarquer que, si on effectue le raisonnement symétrique, le produit scalaire (B.5) peut être borné comme suit :

$$\sigma_n \mathbf{x}^T \mathbf{x} \leq \mathbf{x}^T \mathbf{S} \mathbf{x} \leq \psi_n \mathbf{x}^T \mathbf{x} \quad (\text{B.28})$$

où  $\psi_i$  est la plus grande valeur propre de chaque sous-matrice, donnée par :

$$\psi_{i+1} = \frac{1}{2}(\psi_i + \varphi_i + \sqrt{(\psi_i - \varphi_i)^2 + 4\|\mathbf{v}_i\|^2}) \quad (\text{B.29})$$

L'inégalité (B.28) nous sera utile dans la démonstration des Théorèmes 2 et 8.

## B.1.2 Conditions suffisantes pour la positivité du produit de matrices positives

Considérons  $n$  matrices  $\mathbf{M}_i$  définies positives (c'est-à-dire telles que  $\mathbf{x}^T \mathbf{M}_i \mathbf{x} > 0, \forall \mathbf{x} \neq 0$ ). On s'intéresse au problème de savoir sous quelles conditions la matrice produit est elle aussi définie positive :

$$\mathbf{M} = \mathbf{M}_n \mathbf{M}_{n-1} \cdots \mathbf{M}_1 = \prod_{i=1}^n \mathbf{M}_{n-i} > 0 \quad (\text{B.30})$$

Considérons un vecteur unitaire  $\mathbf{x}_0$ . Celui-ci est transformé, après multiplication par la matrice  $\mathbf{M}_1$ , en  $\mu_1 \mathbf{x}_1$  où  $\mu_1 = \|\mathbf{M}_1 \mathbf{x}_0\|$  et  $\mathbf{x}_1$  est un vecteur unitaire (voir Figure B.1).

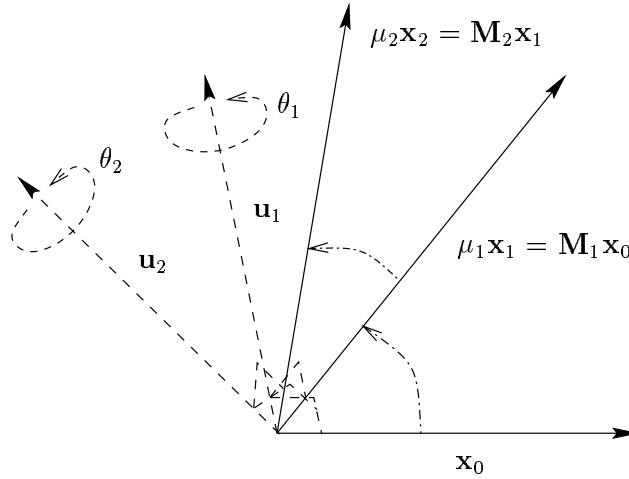


FIG. B.1 – rotation entre deux vecteurs

Le vecteur  $\mathbf{x}_0$  se transforme en  $\mathbf{x}_1$  par la multiplication d'une matrice de rotation  $\mathbf{R}_1$ . Parmi toutes les rotations possibles, on choisit celle dont l'axe de rotation  $\mathbf{u}_1$  est perpendiculaire au plan défini par  $\mathbf{x}_0$  et  $\mathbf{x}_1$  et l'angle de rotation  $\theta_1$  est l'angle entre les deux vecteurs. On peut calculer facilement :

$$\begin{cases} \mathbf{u}_1 &= \frac{[\mathbf{x}_0]_{\times} \mathbf{M}_1 \mathbf{x}_0}{\|[\mathbf{x}_0]_{\times} \mathbf{M}_1 \mathbf{x}_0\|} \\ \cos(\theta_1) &= \frac{\mathbf{x}_0^T \mathbf{M}_1 \mathbf{x}_0}{\|\mathbf{x}_0\| \|\mathbf{M}_1 \mathbf{x}_0\|} \end{cases} \quad (\text{B.31})$$

Comme, par hypothèse, la matrice  $\mathbf{M}_1$  est définie positive, le cosinus de l'angle de rotation l'est aussi et donc  $|\theta_1| \leq \frac{\pi}{2}$ . Le vecteur  $\mathbf{x}_1$  peut s'écrire :

$$\mathbf{x}_1 = e^{[\mathbf{u}_1 \theta_1]_{\times}} \mathbf{x}_0 = \mathbf{R}_1 \mathbf{x}_0 \quad (\text{B.32})$$

et on a  $\mathbf{M}_1 = \mu_1 \mathbf{R}_1$ . En répétant le même raisonnement, le vecteur  $\mu_1 \mathbf{x}_1$  est transformé en  $\mu_2 \mathbf{x}_2 = \mu_1 \mathbf{M}_2 \mathbf{x}_1$ . Par récurrence, on a donc :

$$\mu_n \mathbf{x}_n = \mu_{n-1} \mathbf{M}_n \mathbf{x}_{n-1} = \left( \prod_{i=1}^n \mathbf{M}_i \right) \mathbf{x}_0 \quad (\text{B.33})$$

où

$$\mu_n = \prod_{i=1}^n \|\mathbf{M}_i \mathbf{x}_{i-1}\| \quad (\text{B.34})$$

et

$$\mathbf{x}_n = \left( \prod_{i=1}^n \mathbf{R}_i \right) \mathbf{x}_0 \quad (\text{B.35})$$

Le  $i$ -ème axe de rotation et le  $i$ -ème angle de rotation de la matrice  $\mathbf{R}_i$  sont donnés par :

$$\begin{cases} \mathbf{u}_i &= \frac{[\mathbf{x}_{i-1}]_{\times} \mathbf{M}_i \mathbf{x}_{i-1}}{\|[\mathbf{x}_{i-1}]_{\times} \mathbf{M}_i \mathbf{x}_{i-1}\|} \\ \cos(\theta_i) &= \frac{\mathbf{x}_{i-1}^T \mathbf{M}_i \mathbf{x}_{i-1}}{\|\mathbf{x}_{i-1}\| \|\mathbf{M}_i \mathbf{x}_{i-1}\|} \end{cases} \quad (\text{B.36})$$

L'étude de la relation (B.30) se ramène donc à l'étude de :

$$\mathbf{x}_0^T \left( \prod_{i=1}^n \mathbf{M}_i \right) \mathbf{x}_0 = \mu_n \mathbf{x}_0^T \mathbf{x}_n = \mu_n \mathbf{x}_0^T \left( \prod_{i=1}^n \mathbf{R}_i \right) \mathbf{x}_0 > 0 \quad \forall \mathbf{x}_0 \neq 0 \quad (\text{B.37})$$

et, comme  $\mu_n > 0$ , il nous reste à trouver sous quelles conditions le produit de  $n$  matrices de rotation positives est lui aussi positif. Soit  $\varphi_i$  l'angle et  $\mathbf{v}_i$  l'axe de rotation de la matrice de rotation  $\mathbf{R}'_i$  définie par la formule  $\mathbf{R}'_i = \mathbf{R}_i \mathbf{R}'_{i-1}$  (avec  $\mathbf{R}'_1 = \mathbf{R}_1$ ). On a :

$$\cos\left(\frac{\varphi_i}{2}\right) = \cos\left(\frac{\theta_i + \varphi_{i-1}}{2}\right) + (1 - \mathbf{u}_i^T \mathbf{v}_{i-1}) \sin\left(\frac{\theta_i}{2}\right) \sin\left(\frac{\varphi_{i-1}}{2}\right) \quad (\text{B.38})$$

Par conséquent,  $\mathbf{R}'_i$  est positive si :

$$|\varphi_i| \leq |\theta_i + \varphi_{i-1}| \leq |\theta_i| + |\varphi_{i-1}| \leq \frac{\pi}{2} \quad (\text{B.39})$$

et, par récurrence,  $\mathbf{R}'_n = \left( \prod_{i=1}^n \mathbf{R}_i \right)$  est positive si :

$$|\varphi_n| \leq \sum_{i=1}^n |\theta_i| \leq \frac{\pi}{2} \quad (\text{B.40})$$

Donc, si la somme des valeurs absolues des angles de rotation de chaque matrice  $\mathbf{R}_i$  est inférieure à  $\pi/2$  alors la matrice  $\mathbf{M}$  est définie positive. Considérons à présent l'expression du cosinus du  $i$ -ème angle, donnée par (B.36). On obtient la minoration suivante :

$$\cos(\theta_i) = \frac{\mathbf{x}_{i-1}^T \mathbf{M}_i \mathbf{x}_{i-1}}{\|\mathbf{x}_{i-1}\| \|\mathbf{M}_i \mathbf{x}_{i-1}\|} \geq \frac{\|\mathbf{x}_{i-1}\|^2 \rho^{-1} ((\mathbf{M}_i + \mathbf{M}_i^T)^{-1})}{2 \|\mathbf{x}_{i-1}\|^2 \|\mathbf{M}_i\|} \geq \frac{\rho^{-1} ((\mathbf{M}_i + \mathbf{M}_i^T)^{-1})}{2 \sqrt{\rho(\mathbf{M}_i^T \mathbf{M}_i)}} \quad (\text{B.41})$$

La condition (B.40) est donc vérifiée si :

$$\sum_{i=1}^n |\theta_i| \leq \sum_{i=1}^n \left| \arccos\left(\frac{\rho^{-1}((\mathbf{M}_i + \mathbf{M}_i^T)^{-1})}{2\sqrt{\rho(\mathbf{M}_i^T \mathbf{M}_i)}}\right) \right| \leq \frac{\pi}{2} \quad (\text{B.42})$$

La condition (B.41) peut se simplifier dans les cas suivants :

- si  $\mathbf{M}_i$  est une matrice de rotation d'angle  $\theta_i$ , alors :

$$\frac{\rho^{-1}((\mathbf{M}_i + \mathbf{M}_i^T)^{-1})}{2\sqrt{\rho(\mathbf{M}_i^T \mathbf{M}_i)}} = \cos(\theta_i) \quad (\text{B.43})$$

- si  $\mathbf{M}_i$  est une matrice symétrique, alors :

$$\frac{\rho^{-1}((\mathbf{M}_i + \mathbf{M}_i^T)^{-1})}{2\sqrt{\rho(\mathbf{M}_i^T \mathbf{M}_i)}} = \frac{\rho^{-1}(\mathbf{M}_i^{-1})}{\rho(\mathbf{M}_i)} \quad (\text{B.44})$$

- si  $\mathbf{M}_i$  est une matrice quelconque, en tenant compte des résultats obtenus dans le paragraphe précédent, une autre minoration du cosinus de l'angle est :

$$\cos(\theta_i) = \frac{\mathbf{x}_{i-1}^T \mathbf{M}_i \mathbf{x}_{i-1}}{\|\mathbf{x}_{i-1}\| \|\mathbf{M}_i \mathbf{x}_{i-1}\|} \geq \frac{\sigma_n \|\mathbf{x}_{i-1}\|^2}{\psi_n \|\mathbf{x}_{i-1}\|^2} \geq \frac{\sigma_n}{\psi_n} \quad (\text{B.45})$$

où  $\sigma_n$  est la plus petite valeur propre de la dernière de la suite des matrices associées à la matrice  $(\mathbf{M}_i + \mathbf{M}_i^T)/2$  et  $\psi_n$  est la plus grande valeur propre de la dernière de la suite des matrices associées à la matrice  $\mathbf{M}_i^T \mathbf{M}_i$ .

Ces simplifications nous seront utiles dans les démonstrations des Théorèmes 3 et 6.

## B.2 Asservissement visuel $2D\frac{1}{2}$

### B.2.1 Démonstration du théorème 1

On démontre ici la stabilité asymptotique locale de l'asservissement visuel  $2D\frac{1}{2}$  avec une commande proportionnelle et en présence d'erreurs de calibration sur les paramètres caméra.

#### Preuve du théorème 1

Considérons le développement de Taylor au premier ordre du système différentiel non-linéaire (3.33) autour de  $\mathbf{e}^\circ$  :

$$\dot{\mathbf{e}} \approx -\lambda \mathbf{Q}(\mathbf{e}^\circ)(\mathbf{e} - \mathbf{e}^\circ) \quad (\text{B.46})$$

Le système linéarisé (B.46) est asymptotiquement stable si et seulement si les valeurs propres de  $\mathbf{Q}(\mathbf{e}^\circ)$  sont positives. Comme  $\mathbf{L}_\omega(\mathbf{e}^\circ) = \mathbf{I}$ , la matrice  $\mathbf{Q}(\mathbf{e}^\circ)$  est triangulaire et ses valeurs propres sont données par :

$$\begin{aligned} \lambda_1 &= \nu \alpha_u / \hat{\alpha}_u, & \lambda_3 &= \nu, & \lambda_5 &= \mu \alpha_u / \hat{\alpha}_u, \\ \lambda_2 &= \nu \alpha_v / \hat{\alpha}_v, & \lambda_4 &= \mu, & \lambda_6 &= \mu \alpha_v / \hat{\alpha}_v \end{aligned} \quad (\text{B.47})$$

et sont donc positives si et seulement si les conditions (3.38) sont vérifiées. Puisque le système linéarisé est asymptotiquement stable, la stabilité asymptotique locale, autour du point d'équilibre, du système est démontré.

### B.2.2 Démonstration du théorème 2

On démontre ici la stabilité asymptotique globale de l'asservissement visuel  $2D\frac{1}{2}$  avec une commande proportionnelle et en présence d'erreurs de calibration sur les paramètres caméra.

#### Preuve du théorème 2

On suppose que les conditions nécessaires et suffisantes (3.38), qui assurent la stabilité locale asymptotique du système, sont vérifiées. Comme on le sait, ce n'est pas une hypothèse contraignante en pratique, et elle doit être faite pour la démonstration suivante car ces conditions sont aussi nécessaires (mais pas suffisantes) pour la stabilité asymptotique globale du système. On rappelle qu'une condition suffisante pour la stabilité asymptotique globale du système (3.33) est la positivité de la matrice  $\mathbf{Q}'$  (voir équation (3.43)). Cette matrice est positive si et seulement si sa partie symétrique  $\mathbf{S} = (\mathbf{Q}' + \mathbf{Q}'^T)/2$  est positive. En notant  $\mathbf{e} = [\mathbf{e}_1^T \quad \mathbf{e}_2^T]^T$  un élément de  $\mathcal{S}$ , la matrice  $\mathbf{S}$  est positive si et seulement si :

$$\frac{1}{2} \mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 + \frac{1}{2} \mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 + \mathbf{e}_1^T \mathbf{S}_{12} \mathbf{e}_2 > 0, \quad \forall \mathbf{e} \neq 0 \quad (\text{B.48})$$

Les matrices  $\mathbf{S}_{11}$  et  $\mathbf{S}_{22}$  doivent être positives car, si  $\mathbf{e}_1 = 0$ , l'inégalité devient  $\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 > 0$  ( $\forall \mathbf{e}_2 \neq 0$ ), et, si  $\mathbf{e}_2 = 0$ , l'inégalité devient  $\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 > 0$  ( $\forall \mathbf{e}_1 \neq 0$ ). On va déterminer



maintenant les conditions pour leur positivité et, afin de simplifier l'inégalité (B.48), on va trouver une borne inférieure pour  $\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1$  et  $\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2$ , et une borne supérieure pour  $\mathbf{e}_1^T \mathbf{S}_{12} \mathbf{e}_2$ .

- Considérons d'abord la matrice  $\mathbf{S}_{11}$  (voir équation (3.37)) :

$$\mathbf{S}_{11} = \nu \begin{bmatrix} \delta \mathbf{A}_{11} + \delta \mathbf{A}_{11}^T & \delta \mathbf{m} \\ \delta \mathbf{m}^T & 2 \end{bmatrix} \quad (\text{B.49})$$

où  $\delta \mathbf{m} = [\hat{x} - x \quad \hat{y} - y]^T$ . Une condition nécessaire pour que cette matrice soit définie positive est  $\nu = \frac{d^*}{d^*} \|\delta \mathbf{A}^{-T} \mathbf{n}^*\| > 0$ . Cette condition est évidemment toujours respectée. De plus, la matrice  $\mathbf{S}_{11}$  est positive seulement si la matrice  $\delta \mathbf{A}_{11} + \delta \mathbf{A}_{11}^T$  est positive. Cette matrice symétrique est positive si et seulement si ses valeurs propres sont positives, et donc si et seulement si sa valeur propre minimale, appelé  $\sigma$ , est positive :

$$\sigma = \frac{\alpha_u}{\hat{\alpha}_u} + \frac{\alpha_v}{\hat{\alpha}_v} - \sqrt{\left(\frac{\alpha_u}{\hat{\alpha}_u} - \frac{\alpha_v}{\hat{\alpha}_v}\right)^2 + \left(\frac{\alpha_{uv}}{\hat{\alpha}_u} - \frac{\hat{\alpha}_{uv} \alpha_v}{\hat{\alpha}_u \hat{\alpha}_v}\right)^2} > 0 \quad (\text{B.50})$$

On retrouve ici la condition (3.44). On va maintenant supposer que cette condition est satisfaite. La matrice  $\mathbf{S}_{11}$  peut donc être minorée :

$$\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 \geq \mathbf{e}_1^T \mathbf{S}'_{11} \mathbf{e}_1, \quad \forall \mathbf{e}_1 \quad (\text{B.51})$$

où :

$$\mathbf{S}'_{11} = \nu \begin{bmatrix} \sigma \mathbf{I}_2 & \delta \mathbf{m} \\ \delta \mathbf{m}^T & 2 \end{bmatrix} \quad (\text{B.52})$$

Si la matrice  $\mathbf{S}'_{11}$  est positive alors  $\mathbf{S}_{11}$  l'est également. En rappelant que  $\|\delta \mathbf{m}\| = \|(\mathbf{I} - \delta \mathbf{A}) \mathbf{m}\|$ , la valeur propre minimale de  $\mathbf{S}'_{11}$  est :

$$\sigma'_1 = \nu(\sigma + 1 - \sqrt{(\sigma - 1)^2 + \|\delta \mathbf{m}\|^2}) = \nu(\sigma + 1 - \sqrt{(\sigma - 1)^2 + \|(\mathbf{I} - \delta \mathbf{A}) \mathbf{m}\|^2}) \quad (\text{B.53})$$

Encore une fois, la matrice  $\mathbf{S}'_{11}$  est positive si sa valeur propre minimale est positive. Comme  $\|(\mathbf{I} - \delta \mathbf{A}) \mathbf{m}\|^2 \leq \|(\mathbf{I} - \delta \mathbf{A})\|^2 \|\mathbf{m}\|^2$  et  $\|\mathbf{m}\|^2 = (1 + \gamma^2) \leq (1 + \bar{\gamma}^2)$  (où  $\gamma = \sqrt{x^2 + y^2}$  et  $\bar{\gamma}$  est sa valeur maximale),  $\sigma'_1$  peut être minoré comme suit :

$$\sigma'_1 \geq \nu(\sigma + 1 - \sqrt{(\sigma - 1)^2 + \|(\mathbf{I} - \delta \mathbf{A})\|^2 (1 + \bar{\gamma}^2)}) \quad (\text{B.54})$$

Donc, comme  $\nu > 0$ ,  $\sigma'_1$  est positif si :

$$\sigma_1 = \sigma + 1 - \sqrt{(\sigma - 1)^2 + \|(\mathbf{I} - \delta \mathbf{A})\|^2 (1 + \bar{\gamma}^2)} > 0 \quad (\text{B.55})$$

qui n'est rien d'autre que la condition (3.45). Ceci signifie que si  $\sigma > 0$  et  $\sigma_1 > 0$  alors  $\mathbf{S}_{11} > 0$  et  $\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1$  peut être minoré par :

$$\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 \geq \mathbf{e}_1^T \mathbf{S}'_{11} \mathbf{e}_1 \geq \sigma'_1 \|\mathbf{e}_1\|^2 \geq \nu \sigma_1 \|\mathbf{e}_1\|^2 \quad (\text{B.56})$$

- Considérons maintenant la matrice  $\mathbf{S}_{22}$  (voir équation (3.37)) :

$$\mathbf{S}_{22} = \mu \begin{bmatrix} \delta \mathbf{A}_{11} + \delta \mathbf{A}_{11}^T & \delta \mathbf{p}_0 \\ \delta \mathbf{p}_0^T & 2 \end{bmatrix} \quad (\text{B.57})$$

Une condition nécessaire pour la positivité de la matrice  $\mathbf{S}_{22}$  est  $\mu = \frac{1}{\|\delta \mathbf{A}_{11}\|} > 0$  qui est évidemment toujours satisfaite. On remarque que cette matrice a la même structure que la matrice  $\mathbf{S}_{11}$ , donc  $\mathbf{S}_{22}$  peut également être minoré par :

$$\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 \geq \mathbf{e}_2^T \mathbf{S}'_{22} \mathbf{e}_2, \quad \forall \mathbf{e}_2 \quad (\text{B.58})$$

où :

$$\mathbf{S}'_{22} = \mu \begin{bmatrix} \sigma \mathbf{I}_2 & \delta \mathbf{p}_0 \\ \delta \mathbf{p}_0^T & 2 \end{bmatrix} \quad (\text{B.59})$$

Encore une fois, la matrice  $\mathbf{S}'_{22}$  est positive si sa valeur propre minimale est positive. La valeur propre minimale de  $\mathbf{S}'_{22}$  est :

$$\sigma'_2 = \mu(\sigma + 1 - \sqrt{(\sigma - 1)^2 + \|\delta \mathbf{p}_0\|^2}) \quad (\text{B.60})$$

Donc, comme  $\mu > 0$ ,  $\sigma'_2$  est positif si :

$$\sigma_2 = \sigma + 1 - \sqrt{(\sigma - 1)^2 + \|\delta \mathbf{p}_0\|^2} > 0 \quad (\text{B.61})$$

ce qui donne la condition (3.46). Par conséquent, si  $\sigma > 0$  et  $\sigma_2 > 0$  alors  $\mathbf{S}_{22} > 0$  et  $\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2$  peut être minoré par :

$$\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 \geq \mathbf{e}_2^T \mathbf{S}'_{22} \mathbf{e}_2 \geq \sigma'_2 \|\mathbf{e}_2\|^2 \geq \mu \sigma_2 \|\mathbf{e}_2\|^2 \quad (\text{B.62})$$

- Considérons maintenant la matrice  $\mathbf{S}_{12}$  (voir équation (3.37)) :

$$\mathbf{S}_{12} = \mu Z \mathbf{L}_v ([\mathbf{m} - \nu \hat{\mathbf{m}}]_{\times}) \delta \mathbf{A} \quad (\text{B.63})$$

La norme  $l_2$  de  $\mathbf{S}_{12}$  peut être majorée de la manière suivante :

$$\|\mathbf{S}_{12}\| \leq \mu \|\mathbf{Z} \mathbf{L}_v\| \|[\mathbf{m} - \nu \hat{\mathbf{m}}]_{\times}\| \|\delta \mathbf{A}\| \quad (\text{B.64})$$

et, comme la norme d'une matrice de préproduit vectoriel associée à un vecteur est égale à la norme du vecteur, on obtient :

$$\begin{aligned} \|\mathbf{S}_{12}\| &\leq \mu \|\mathbf{Z} \mathbf{L}_v\| \|\delta \mathbf{A}\| \|\mathbf{m} - \nu \hat{\mathbf{m}}\| \\ &\leq \mu \|\mathbf{Z} \mathbf{L}_v\| \|\delta \mathbf{A}\| \|(1 - \nu) \mathbf{m} + \nu (\mathbf{m} - \hat{\mathbf{m}})\| \\ &\leq \mu \|\mathbf{Z} \mathbf{L}_v\| \|\delta \mathbf{A}\| \|(1 - \nu) \mathbf{m} + \nu (\mathbf{I} - \delta \mathbf{A}) \mathbf{m}\| \\ &\leq \mu \|\mathbf{Z} \mathbf{L}_v\| \|\delta \mathbf{A}\| \|\mathbf{m}\| (\nu \|\mathbf{I} - \delta \mathbf{A}\| + |1 - \nu|) \end{aligned} \quad (\text{B.65})$$

En notant :

$$g(\gamma) = \|Z\mathbf{L}_v\| \|\mathbf{m}\| = \sqrt{\frac{2 + \gamma^2 + \sqrt{\gamma^2(\gamma^2 + 4)}}{2}}(1 + \gamma^2) \quad (\text{B.66})$$

on peut remarquer que cette fonction ne s'annule jamais ( $g(\gamma) \neq 0 \quad \forall \gamma$ ) et qu'elle est monotone croissante en fonction de la variable positive  $\gamma$ . En effet, la dérivée de cette fonction est toujours positive :

$$\frac{dg(\gamma)}{d\gamma} = \frac{1}{2g(\gamma)} \frac{dg^2(\gamma)}{d\gamma} = \frac{1}{2g(\gamma)} \frac{\partial g^2(\gamma^2)}{\partial \gamma^2} \frac{d\gamma^2}{d\gamma} = \frac{\gamma}{g(\gamma)} \frac{\partial g^2(\gamma^2)}{\partial \gamma^2} > 0 \quad (\text{B.67})$$

avec :

$$\frac{\partial g^2(\gamma^2)}{\partial \gamma^2} = \frac{2\gamma^2 + 3}{2} + \frac{2\gamma^4 + 7\gamma^2 + 2}{2\sqrt{\gamma^2(\gamma^2 + 4)}} > 0 \quad (\text{B.68})$$

Donc, si  $\gamma \leq \bar{\gamma}$  alors  $g(\gamma) \leq g(\bar{\gamma})$ . Par conséquent, on peut encore majorer la norme de  $S_{12}$  :

$$\|\mathbf{S}_{12}\| \leq \mu(\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|)\|\delta\mathbf{A}\|g(\bar{\gamma}) \quad (\text{B.69})$$

On peut finalement trouver la dernière condition pour la positivité de la matrice  $\mathbf{S}$ . L'inégalité (B.48) est vérifiée si :

$$\frac{1}{2}\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 + \frac{1}{2}\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 > |\mathbf{e}_1^T \mathbf{S}_{12} \mathbf{e}_2| \quad (\text{B.70})$$

À partir de (B.56) et (B.62), on a :

$$\frac{1}{2}\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 + \frac{1}{2}\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 \geq \frac{1}{2}\nu\sigma_1\|\mathbf{e}_1\|^2 + \frac{1}{2}\mu\sigma_2\|\mathbf{e}_2\|^2 \quad (\text{B.71})$$

et, à partir de (B.69), on a :

$$|\mathbf{e}_1^T \mathbf{S}_{12} \mathbf{e}_2| \leq \mu(\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|)\|\delta\mathbf{A}\|g(\bar{\gamma})\|\mathbf{e}_1\|\|\mathbf{e}_2\| \quad (\text{B.72})$$

Donc, l'inégalité (B.70) est vérifiée si :

$$\frac{1}{2}\nu\sigma_1\|\mathbf{e}_1\|^2 + \frac{1}{2}\mu\sigma_2\|\mathbf{e}_2\|^2 > \mu(\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|)\|\delta\mathbf{A}\|g(\bar{\gamma})\|\mathbf{e}_1\|\|\mathbf{e}_2\| \quad (\text{B.73})$$

Cette inégalité a la solution suivante  $\forall \mathbf{e}_1$  et  $\forall \mathbf{e}_2$  (avec  $\mathbf{e} \neq 0$ ) :

$$(\mu(\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|)\|\delta\mathbf{A}\|g(\bar{\gamma}))^2 < 4\left(\frac{1}{2}\nu\sigma_1\right)\left(\frac{1}{2}\mu\sigma_2\right) \quad (\text{B.74})$$

qui finalement, peut simplement s'écrire :

$$\mu(\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|)^2\|\delta\mathbf{A}\|^2g^2(\bar{\gamma}) < \nu\sigma_1\sigma_2 \quad (\text{B.75})$$

où l'on retrouve la condition (3.47).

## Preuve du corollaire au théorème 2

On cherche tout d'abord une borne supérieure pour  $\mu = \frac{1}{\|\delta\mathbf{A}\mathbf{u}\|}$ . Le carré de la norme du vecteur  $\delta\mathbf{A}\mathbf{u}$  est :

$$\|\delta\mathbf{A}\mathbf{u}\|^2 = \mathbf{u}^T \delta\mathbf{A}^T \delta\mathbf{A}\mathbf{u} = \mathbf{u}^T \begin{bmatrix} \delta\mathbf{A}_{11}^T \delta\mathbf{A}_{11} & \delta\mathbf{A}_{11}^T \delta\mathbf{p}_0 \\ \delta\mathbf{p}_0^T \delta\mathbf{A}_{11} & 1 \end{bmatrix} \mathbf{u} \quad (\text{B.76})$$

Soit  $\sigma'$  la valeur propre minimale de  $\delta\mathbf{A}_{11}^T \delta\mathbf{A}_{11}$  :

$$\sigma' = \frac{1}{2} \left( \frac{\alpha_u^2}{\widehat{\alpha}_u^2} + \frac{\alpha_v^2}{\widehat{\alpha}_v^2} + \left( \frac{\alpha_{uv}}{\widehat{\alpha}_u} - \frac{\widehat{\alpha}_{uv} \alpha_v}{\widehat{\alpha}_u \widehat{\alpha}_v} \right)^2 - \sqrt{\left( \frac{\alpha_u^2}{\widehat{\alpha}_u^2} + \frac{\alpha_v^2}{\widehat{\alpha}_v^2} + \left( \frac{\alpha_{uv}}{\widehat{\alpha}_u} - \frac{\widehat{\alpha}_{uv} \alpha_v}{\widehat{\alpha}_u \widehat{\alpha}_v} \right)^2 \right)^2 - 4 \frac{\alpha_u^2}{\widehat{\alpha}_u^2} \frac{\alpha_v^2}{\widehat{\alpha}_v^2}} \right) \quad (\text{B.77})$$

On sait à partir de la forme de  $\delta\mathbf{A}_{11}^T \delta\mathbf{A}_{11}$  que  $\sigma'$  est positif. Le carré de la norme du vecteur  $\delta\mathbf{A}\mathbf{u}$  peut donc être majoré par :

$$\|\delta\mathbf{A}\mathbf{u}\|^2 \geq \mathbf{u}^T \delta\mathbf{A}'^T \delta\mathbf{A}' \mathbf{u} \quad (\text{B.78})$$

où :

$$\delta\mathbf{A}'^T \delta\mathbf{A}' = \begin{bmatrix} \sigma' \mathbf{I} & \delta\mathbf{A}_{11}^T \delta\mathbf{p}_0 \\ \delta\mathbf{p}_0^T \delta\mathbf{A}_{11} & 1 \end{bmatrix} \quad (\text{B.79})$$

La valeur propre minimale de  $\delta\mathbf{A}'^T \delta\mathbf{A}'$  est :

$$\frac{1}{2} \left( \sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}^T \delta\mathbf{p}_0\|^2} \right) \quad (\text{B.80})$$

et peut être minoré par :

$$\frac{1}{2} \left( \sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11} \delta\mathbf{p}_0\|^2} \right) \geq \frac{1}{2} \left( \sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2 \|\delta\mathbf{p}_0\|^2} \right) \quad (\text{B.81})$$

Par conséquent, le carré de la norme du vecteur  $\delta\mathbf{A}\mathbf{u}$  peut aussi être minoré par :

$$\|\delta\mathbf{A}\mathbf{u}\|^2 \geq \frac{1}{2} \left( \sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2 \|\delta\mathbf{p}_0\|^2} \right) \|\mathbf{u}\|^2 \quad (\text{B.82})$$

Enfin, comme  $\|\mathbf{u}\|^2 = 1$ , on a :

$$\mu = \frac{1}{\|\delta\mathbf{A}\mathbf{u}\|} \leq \frac{\sqrt{2}}{\sqrt{\sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2 \|\delta\mathbf{p}_0\|^2}}} \quad (\text{B.83})$$

Comme  $\|\delta\mathbf{A}\| \leq \|\delta\mathbf{A}_{11}\| + \sqrt{1 + \|\delta\mathbf{p}_0\|^2}$  et  $\|\mathbf{I}_3 - \delta\mathbf{A}\| \leq \|\mathbf{I}_2 - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\|$ , la partie gauche de la condition (3.47) peut être bornée comme suit :

$$\frac{\mu(\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|)^2\|\delta\mathbf{A}\|^2g^2(\bar{\gamma}) \leq \sqrt{2} \frac{(\nu(\|\mathbf{I} - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\|) + |1 - \nu|)^2(\|\delta\mathbf{A}_{11}\| + \sqrt{1 + \|\delta\mathbf{p}_0\|^2})^2g^2(\bar{\gamma})}{\sqrt{\sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2\|\delta\mathbf{p}_0\|^2}}} \quad (\text{B.84})$$

De plus, la condition (3.45) peut s'écrire :

$$\sigma_1 \geq \sigma_3 = (\sigma + 1 - \sqrt{(\sigma - 1)^2 + (\|\mathbf{I}_2 - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\|)^2(1 + \bar{\gamma}^2)}) > 0 \quad (\text{B.85})$$

qui est la condition (3.50). On remarque que si cette inégalité est vérifiée alors (3.46) est également vérifiée. La condition (3.46) n'est donc plus nécessaire. Finalement, la partie droite de la condition (3.47) peut être bornée de la manière suivante :

$$\nu\sigma_1\sigma_2 \geq \nu\sigma_2\sigma_3 \quad (\text{B.86})$$

et, en utilisant (B.84), on obtient la condition (3.51) :

$$\sqrt{2} \frac{(\nu(\|\mathbf{I} - \delta\mathbf{A}_{11}\| + \|\delta\mathbf{p}_0\|) + |1 - \nu|)^2(\|\delta\mathbf{A}_{11}\| + \sqrt{1 + \|\delta\mathbf{p}_0\|^2})^2g^2(\bar{\gamma})}{\sqrt{\sigma' + 1 - \sqrt{(\sigma' - 1)^2 + 4\|\delta\mathbf{A}_{11}\|^2\|\delta\mathbf{p}_0\|^2}}} < \nu\sigma_2\sigma_3 \quad (\text{B.87})$$

### B.2.3 Démonstration du théorème 3

On démontre ici la stabilité asymptotique locale d'un système différentiel non-linéaire, avec matrice triangulaire, commandé de manière classique et en présence d'erreurs de calibration caméra/effecteur.

#### Preuve du théorème 3

Les valeurs propres de la matrice  $\mathbf{Q}(\mathbf{e}^\circ)$  sont positives si :

$$\mathbf{L}_v^* \delta\mathbf{R} \widehat{\mathbf{L}}_v^{*-1} \mathbf{E}_v > 0 \quad (\text{B.88})$$

$$\delta\mathbf{R} \mathbf{E}_\omega > 0 \quad (\text{B.89})$$

car  $\mathbf{L}_\omega(\mathbf{e}^\circ) = \mathbf{I}$ . La première condition s'écrit aussi :

$$\delta\mathbf{R} \widehat{\mathbf{L}}_v^{*-1} \mathbf{E}_v \mathbf{L}_v^{*-T} = \delta\mathbf{R} \widehat{\mathbf{L}}_v^{*-1} \mathbf{E}_v \mathbf{L}_v^* \mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T} > 0 \quad (\text{B.90})$$

On a (voir équation 3.8) :

$$\widehat{\mathbf{L}}_v^{*-1} \mathbf{E}_v \mathbf{L}_v^* = \frac{\widehat{Z}^*}{Z^*} \begin{bmatrix} -1 & 0 & -\widehat{x}^* \\ 0 & -1 & -\widehat{y}^* \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta\mathbf{A}_{11} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & x^* \\ 0 & -1 & y^* \\ 0 & 0 & -1 \end{bmatrix} = \quad (\text{B.91})$$

$$= \frac{\widehat{Z}^*}{Z^*} \begin{bmatrix} \delta\mathbf{A}_{11} & \delta\mathbf{p}_0 \\ 0 & 1 \end{bmatrix} = \frac{\widehat{Z}^*}{Z^*} \delta\mathbf{A} \quad (\text{B.92})$$

La condition (B.90) s'écrit donc :

$$\frac{\widehat{Z}^*}{Z^*} \delta \mathbf{R} \delta \mathbf{A} \mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T} > 0 \quad (\text{B.93})$$

ce qui implique naturellement que  $\frac{\widehat{Z}^*}{Z^*} > 0$ , c'est-à-dire  $\frac{\widehat{d}^*}{d^*} > 0$ . La condition (B.89) n'est plus nécessaire car elle est incluse dans la condition (B.93) (on rappelle que  $\mathbf{E}_\omega = \mu \delta \mathbf{A}$ , avec  $\mu > 0$ ). De plus, considérons chaque matrice du produit (B.93) :

- $\delta \mathbf{R} > 0$  si l'angle de rotation  $\delta \theta < \frac{\pi}{2}$
- $\delta \mathbf{A} > 0$  si les conditions (3.44) et (3.46) sont vérifiées.
- $\mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T} > 0$  par définition car  $\mathbf{L}_v^*$  est toujours inversible.

Sur la base de l'étude effectuée dans le paragraphe (B.1.2), le produit des trois matrices est positif s'il existe trois angles  $\theta_1$ ,  $\theta_2$  et  $\theta_3$  associés respectivement aux matrices  $\delta \mathbf{R}$ ,  $\delta \mathbf{A}$  et  $\mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T}$  tels que  $|\theta_1| + |\theta_2| + |\theta_3| < \frac{\pi}{2}$  et tels que :

- dans le cas de la matrice de rotation  $\delta \mathbf{R}$  :

$$\frac{\rho^{-1}((\delta \mathbf{R} + \delta \mathbf{R}^T)^{-1})}{2\sqrt{\rho(\delta \mathbf{R}^T \mathbf{R})}} = \cos(\delta \theta) \geq \cos(\theta_1) \quad (\text{B.94})$$

- dans le cas de la matrice  $\delta \mathbf{A}$  :

$$\frac{\rho^{-1}((\delta \mathbf{A} + \delta \mathbf{A}^T)^{-1})}{2\sqrt{\rho(\delta \mathbf{A}^T \mathbf{A})}} \geq \cos(\theta_2) \quad (\text{B.95})$$

- dans le cas de la matrice symétrique  $\mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T}$  :

$$\frac{\rho^{-1}((\mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T} + \mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T})^{-1})}{2\sqrt{\rho((\mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T})^2)}} = \frac{\lambda_{min}^*}{\lambda_{max}^*} \geq \cos(\theta_3) \quad (\text{B.96})$$

où  $\lambda_{min}^*$  et  $\lambda_{max}^*$  sont respectivement la plus petite et la plus grande valeur propre de  $\mathbf{L}_v^{*-1} \mathbf{L}_v^{*-T}$  :

$$\lambda_{min}^* = 1 + \frac{x^{*2} + y^{*2}}{2} - \frac{1}{2} \sqrt{(x^{*2} + y^{*2})^2 + 4(x^{*2} + y^{*2})} \quad (\text{B.97})$$

$$\lambda_{max}^* = 1 + \frac{x^{*2} + y^{*2}}{2} + \frac{1}{2} \sqrt{(x^{*2} + y^{*2})^2 + 4(x^{*2} + y^{*2})} \quad (\text{B.98})$$

## B.2.4 Démonstration du théorème 4

On démontre ici la stabilité asymptotique globale d'un système différentiel non-linéaire, avec matrice triangulaire, commandé de manière adaptative.

### Preuve du théorème 4

On démontre d'abord que l'état est toujours borné :  $\mathbf{x}_i(t) \in S_i, \forall t$ . Supposons que  $\mathbf{x}_i$  soit

plus proche de sa frontière  $\Gamma_i$  que les autres variables  $\mathbf{x}_j \forall j > i$  de leur frontière  $\Gamma_j$ . La dérivée de la norme de l'erreur  $\mathbf{x}_i$  est :

$$\lim_{\mathbf{x}_i \rightarrow \Gamma_i} \frac{1}{2} \frac{d}{dt} \|\mathbf{x}_i\|^2 = \lim_{\mathbf{x}_i \rightarrow \Gamma_i} \mathbf{x}_i^T \mathbf{M}_{ii}(\mathbf{x}) \mathbf{x}_i + \lim_{\mathbf{x}_i \rightarrow \Gamma_i} \sum_{j=i+1}^n \mathbf{x}_i^T \mathbf{M}_{ij}(\mathbf{x}) \mathbf{x}_j \quad (\text{B.99})$$

D'après l'hypothèse (3.85),  $\|\mathbf{x}_j\|$  étant borné, on en déduit :

$$\lim_{\mathbf{x}_i \rightarrow \Gamma_i} \sum_{j=i+1}^n \mathbf{x}_i^T \mathbf{M}_{ij}(\mathbf{x}) \mathbf{x}_j \leq \lim_{\mathbf{x}_i \rightarrow \Gamma_i} \sum_{j=i+1}^n \|\mathbf{M}_{ij}(\mathbf{x})\| \|\mathbf{x}_i\| \|\mathbf{x}_j\| = 0 \quad (\text{B.100})$$

donc, grâce à l'hypothèse (3.84), la norme de l'erreur décroît :

$$\lim_{\mathbf{x}_i \rightarrow \Gamma_i} \frac{1}{2} \frac{d}{dt} \|\mathbf{x}_i\|^2 = \lim_{\mathbf{x}_i \rightarrow \Gamma_i} \mathbf{x}_i^T \mathbf{M}_{ii}(\mathbf{x}) \mathbf{x}_i < 0 \quad (\text{B.101})$$

La variable  $\mathbf{x}_i$  s'éloigne donc de ses bornes. Comme cela est vrai  $\forall i$ , alors  $\mathbf{x}(t) \in S, \forall t$ . Par conséquent, comme la matrice  $\mathbf{M}$  est une fonction continue de  $\mathbf{x}$ , la norme de ses sous-matrices  $\mathbf{M}_{ij}(\mathbf{x})$  est aussi bornée.

On démontre maintenant la convergence asymptotique du système, en prouvant que, si les erreurs  $\mathbf{x}_j (\forall j > i)$  tendent vers zéro alors l'erreur  $\mathbf{x}_i$  tend aussi vers zéro. La dérivée de la norme de l'erreur  $\mathbf{x}_i$  est :

$$\lim_{t \rightarrow \infty} \frac{1}{2} \frac{d}{dt} \|\mathbf{x}_i\|^2 = \lim_{t \rightarrow \infty} \mathbf{x}_i^T \mathbf{M}_{ii}(\mathbf{x}) \mathbf{x}_i + \lim_{t \rightarrow \infty} \sum_{j=i+1}^n \mathbf{x}_i^T \mathbf{M}_{ij}(\mathbf{x}) \mathbf{x}_j \quad (\text{B.102})$$

Comme  $\|\mathbf{x}_i\|$  et  $\|\mathbf{M}_{ij}(\mathbf{x})\|$  sont bornées et comme l'erreur sur  $\mathbf{x}_j$  tend vers zéro  $\forall j > i$ , on a :

$$\lim_{t \rightarrow \infty} \sum_{j=i+1}^n \mathbf{x}_i^T \mathbf{M}_{ij}(\mathbf{x}) \mathbf{x}_j \leq \lim_{t \rightarrow \infty} \sum_{j=i+1}^n \|\mathbf{M}_{ij}(\mathbf{x})\| \|\mathbf{x}_i\| \|\mathbf{x}_j\| = 0 \quad (\text{B.103})$$

Grâce à l'hypothèse (3.84), la dérivée de la norme de l'erreur sur  $\mathbf{x}_i$  décroît donc vers zéro :

$$\lim_{t \rightarrow \infty} \frac{1}{2} \frac{d}{dt} \|\mathbf{x}_i\|^2 = \lim_{t \rightarrow \infty} \mathbf{x}_i^T \mathbf{M}_{ii}(\mathbf{x}) \mathbf{x}_i \leq 0 \quad (\text{B.104})$$

Ceci signifie que si l'erreur  $\mathbf{x}_n$  tend vers zéro alors l'erreur sur  $\mathbf{x}_{n-1}$  tend également vers zéro. De même, si les erreurs  $\mathbf{x}_n$  et  $\mathbf{x}_{n-1}$  tendent vers zéro alors l'erreur  $\mathbf{x}_{n-2}$  tend vers zéro, et enfin, par récurrence, si les erreurs  $\mathbf{x}_n, \mathbf{x}_{n-1} \dots \mathbf{x}_2$  tendent vers zéro alors l'erreur  $\mathbf{x}_1$  tend vers zéro. Grâce à l'hypothèse (3.84), l'erreur  $\mathbf{x}_n$  tend vers zéro car  $\mathbf{M}_{nn}(\mathbf{x}) < 0$ . On a donc :

$$\lim_{t \rightarrow \infty} \|\mathbf{x}\| = 0 \quad (\text{B.105})$$

## B.2.5 Démonstration du théorème 5

On démontre ici la stabilité asymptotique globale de l'asservissement visuel  $2D\frac{1}{2}$  avec une commande proportionnelle robuste et en présence d'erreurs de calibration sur les paramètres caméra.

### Preuve du théorème 5

On démontre d'abord que les conditions sont nécessaires pour la stabilité globale, car elles sont nécessaires pour la stabilité locale. On démontre ensuite que les conditions sont suffisantes. Considérons maintenant le développement de Taylor au premier ordre du système différentiel non-linéaire (3.33) autour de  $\mathbf{e}^\circ$  :

$$\dot{\mathbf{e}} \approx -\lambda \mathbf{Q}(\mathbf{e}^\circ)(\mathbf{e} - \mathbf{e}^\circ) \quad (\text{B.106})$$

Le système linéarisé (B.106) est asymptotiquement stable si et seulement si les valeurs propres de  $\mathbf{Q}(\mathbf{e}^\circ)$  sont positives. Comme  $\mathbf{L}_\omega(\mathbf{e}^\circ) = \mathbf{I}$ , la matrice  $\mathbf{Q}(\mathbf{e}^\circ)$  est triangulaire et ses valeurs propres sont données par :

$$\begin{aligned} \lambda_1 &= \nu \alpha_u / \hat{\alpha}_u, & \lambda_3 &= \nu f_u(u) f_v(v), & \lambda_5 &= \mu \alpha_u / \hat{\alpha}_u f_u(u) f_v(v) f_w(w), \\ \lambda_2 &= \nu \alpha_v / \hat{\alpha}_v f_u(u), & \lambda_4 &= \mu f_u(u) f_v(v) f_w(w), & \lambda_6 &= \mu \alpha_v / \hat{\alpha}_v f_u(u) f_v(v) f_w(w) \end{aligned} \quad (\text{B.107})$$

Elles sont donc positives si et seulement si les conditions (3.38) sont vérifiées. Puisque le système linéarisé est asymptotiquement stable, la stabilité asymptotique locale, autour du point d'équilibre, du système est prouvé.

On peut poser  $x_1 = e_1$ ,  $x_2 = e_2$ ,  $x_3 = e_3$  et  $\mathbf{x}_4 = [e_4 \ e_5 \ e_6]^T$ , ce dernier sous-système pouvant évoluer librement. On va simplement vérifier que, en choisissant le contrôle (3.89), le système en boucle fermée satisfait les hypothèses du théorème. On remarque que  $\mathbf{D} \mathbf{E} = \mathbf{E} \mathbf{D}$ . Le système en boucle fermée s'écrit donc :

$$\mathbf{M} = \begin{bmatrix} \mathbf{L}_v \hat{\mathbf{L}}_v^{-1} \delta \mathbf{A} & \nu \mathbf{L}_v [\mathbf{x} - \hat{\mathbf{x}}]_\times \delta \mathbf{A} \\ 0 & \mu \mathbf{L}_\omega \delta \mathbf{A} \end{bmatrix} \mathbf{D} \quad (\text{B.108})$$

et les quatre blocs sur la diagonale sont positifs si :

$$\mathbf{M}_{11} = \nu \frac{\alpha_u}{\hat{\alpha}_u} > 0 \quad (\text{B.109})$$

$$\mathbf{M}_{22} = \nu \frac{\alpha_v}{\hat{\alpha}_v} f_u(u) > 0 \quad (\text{B.110})$$

$$\mathbf{M}_{33} = \nu f_v(v) f_w(w) > 0 \quad (\text{B.111})$$

$$\mathbf{M}_{44} = \mu \delta \mathbf{A} f_u(u) f_v(v) f_w(w) > 0 \quad (\text{B.112})$$

car on rappelle que  $\mathbf{x}_4^T (\mu \mathbf{L}_\omega \delta \mathbf{A} \mathbf{D}_\omega) \mathbf{x}_4 = \mathbf{x}_4^T (\mu \delta \mathbf{A} \mathbf{D}_\omega) \mathbf{x}_4$ . Les trois premières conditions donnent les conditions nécessaires du Théorème (5). La condition  $\mathbf{M}_{44} > 0$  n'est pas très contraignante comme cela a déjà été démontré (voir conditions (3.44) et (3.46)). Le scalaire  $\mu f_u(u) f_v(v) f_w(w)$  est toujours positif. Le sous-système est donc stable si  $\delta \mathbf{A} > 0$ , ce qui n'est rien d'autre que la condition (3.95).



On démontre maintenant que l'hypothèse 3.85 est vérifiée grâce au choix des gains adaptatifs de la matrice  $\mathbf{D}$ :

- si  $w \rightarrow w_{min}$  ou  $w \rightarrow w_{max}$ , les trois dernières colonnes de  $\mathbf{D}$  tendent vers zéro de manière exponentielle en fonction des variables d'état, donc les trois dernières colonnes de la matrice  $\mathbf{M}$  tendent également vers zéro.
- si  $v \rightarrow v_{min}$  ou  $v \rightarrow v_{max}$ , les quatre dernières colonnes de  $\mathbf{D}$  tendent vers zéro de manière exponentielle en fonction des variables d'état, donc les quatre dernières colonnes de la matrice  $\mathbf{M}$  tendent également vers zéro.
- si  $u \rightarrow u_{min}$  ou  $u \rightarrow u_{max}$ , les cinq dernières colonnes de  $\mathbf{D}$  tendent vers zéro de manière exponentielle en fonction des variables d'état, donc les cinq dernières colonnes de la matrice  $\mathbf{M}$  tendent également vers zéro.

## B.2.6 Démonstration du théorème 6

On démontre ici la stabilité asymptotique globale de l'asservissement visuel 2D 1/2 avec une commande robuste et en présence d'erreurs de calibration sur les paramètres caméra et caméra/effecteur.

### Preuve du théorème 6

Comme pour la démonstration du Théorème 3, grâce aux choix de  $\mathbf{D}_\omega$ , il suffit de démontrer que chaque sous-système sur la diagonale est stable pour que, grâce au corollaire du Théorème 4, le système global soit stable. Les blocs de la diagonale de la matrice  $\mathbf{Q}$  donnée en (3.97) sont positives si :

$$\mathbf{L}_v \delta \mathbf{R} \hat{\mathbf{L}}_v^{-1} \mathbf{E}_v > 0 \quad (\text{B.113})$$

$$\delta \mathbf{R} \mathbf{D}_\omega \mathbf{E}_\omega > 0 \quad (\text{B.114})$$

avec  $\mathbf{D}_\omega = \mathbf{I}f(u)f(v)f(w)$  où  $f(u)f(v)f(w) > 0$ . La première condition s'écrit aussi :

$$\delta \mathbf{R} \hat{\mathbf{L}}_v^{-1} \mathbf{E}_v \mathbf{L}_v^{-T} = \delta \mathbf{R} \hat{\mathbf{L}}_v^{-1} \mathbf{E}_v \mathbf{L}_v \mathbf{L}_v^{-1} \mathbf{L}_v^{-T} > 0 \quad (\text{B.115})$$

On a (voir équation (3.8)) :

$$\hat{\mathbf{L}}_v^{-1} \mathbf{E}_v \mathbf{L}_v = \frac{\hat{\mathbf{Z}}}{\mathbf{Z}} \delta \mathbf{A} \quad (\text{B.116})$$

La condition (B.115) s'écrit donc :

$$\frac{\hat{\mathbf{Z}}}{\mathbf{Z}} \delta \mathbf{R} \delta \mathbf{A} \mathbf{L}_v^{-1} \mathbf{L}_v^{-T} > 0 \quad (\text{B.117})$$

ce qui implique naturellement que  $\frac{\hat{\mathbf{Z}}}{\mathbf{Z}} > 0$ , c'est-à-dire  $\frac{\hat{d}^*}{d^*} > 0$ . La condition (B.114) n'est plus nécessaire car elle est incluse dans la condition (B.117). Les conditions du Théorème 6 se calculent de la même manière que celles du Théorème 3. La seule différence est que

maintenant la matrice  $\mathbf{L}_v^{-1} \mathbf{L}_v^{-T}$  n'est plus constante et calculée à la position désirée. La condition correspondante à la condition (B.96) s'écrit :

$$\frac{\rho^{-1}((\mathbf{L}_v^{-1} \mathbf{L}_v^{-T} + \mathbf{L}_v^{-1} \mathbf{L}_v^{-T})^{-1})}{2\sqrt{\rho((\mathbf{L}_v^{-1} \mathbf{L}_v^{-T})^2)}} = \frac{\lambda_{min}}{\lambda_{max}} \geq \cos(\theta_3) \quad (\text{B.118})$$

où  $\lambda_{min}$  et  $\lambda_{max}$  sont respectivement la plus petite et la plus grande valeur propre de  $\mathbf{L}_v^{-1} \mathbf{L}_v^{-T}$  :

$$\lambda_{min} = 1 + \frac{x^2 + y^2}{2} - \frac{1}{2}\sqrt{(x^2 + y^2)^2 + 4(x^2 + y^2)} \quad (\text{B.119})$$

$$\lambda_{max} = 1 + \frac{x^2 + y^2}{2} + \frac{1}{2}\sqrt{(x^2 + y^2)^2 + 4(x^2 + y^2)} \quad (\text{B.120})$$

## B.3 Asservissement visuel $2D\frac{3}{4}$

### B.3.1 Démonstration du théorème 7

On démontre ici la stabilité asymptotique locale de l'asservissement visuel  $2D\frac{3}{4}$  avec une commande classique et en présence d'erreurs de calibration sur les paramètres caméra.

#### Preuve du théorème 7

Considérons le développement de Taylor au premier ordre du système différentiel non-linéaire (3.123) autour de  $\mathbf{e}^\circ$  :

$$\dot{\mathbf{e}} \approx -\lambda \mathbf{Q}(\mathbf{e}^\circ)(\mathbf{e} - \mathbf{e}^\circ) \quad (\text{B.121})$$

Le système linéarisé (B.121) est asymptotiquement stable si et seulement si les valeurs propres de  $\mathbf{Q}(\mathbf{e}^\circ)$  sont positives. Celles-ci ont pour valeur :

$$\begin{aligned} \lambda_1 &= \nu \alpha_u / \hat{\alpha}_u, & \lambda_3 &= \nu, & \lambda_5 &= \mu \alpha_u / \hat{\alpha}_u, \\ \lambda_2 &= \nu \alpha_v / \hat{\alpha}_v, & \lambda_4 &= \mu, & \lambda_6 &= \mu \alpha_v / \hat{\alpha}_v \end{aligned} \quad (\text{B.122})$$

et sont positives si et seulement si les conditions (3.125) sont vérifiées.

### B.3.2 Démonstration du théorème 8

On démontre ici la stabilité asymptotique globale de l'asservissement visuel  $2D\frac{3}{4}$  avec une commande classique et en présence d'erreurs de calibration sur les paramètres caméra.

#### Preuve du théorème 8

On suppose que les conditions nécessaires et suffisantes (3.125), qui assurent la stabilité locale asymptotique du système, sont vérifiées. Soit  $\mathbf{S} = (\mathbf{Q} + \mathbf{Q}^T)/2$  la partie symétrique de  $\mathbf{Q}$ . En notant  $\mathbf{e} = [\mathbf{e}_1^T \ \mathbf{e}_2^T]^T$  un élément de  $\mathcal{S}$ , la matrice  $\mathbf{S}$  est positive si et seulement si :

$$\frac{1}{2}\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 + \frac{1}{2}\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 + \mathbf{e}_1^T \mathbf{S}_{12} \mathbf{e}_2 > 0, \quad \forall \mathbf{e} \neq 0 \quad (\text{B.123})$$

Comme dans la démonstration du Théorème 2, on va déterminer maintenant les conditions pour la positivité de  $\mathbf{S}_{11}$  et  $\mathbf{S}_{22}$  et, afin de simplifier l'inégalité (B.123), on va trouver une borne inférieure pour  $\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1$  et  $\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2$ , et une borne supérieure pour  $\mathbf{e}_1^T \mathbf{S}_{12} \mathbf{e}_2$ .

- Considérons les matrices  $\mathbf{S}_{11} = \nu(\delta\mathbf{A} + \delta\mathbf{A}^T)$  et  $\mathbf{S}_{22} = \mu(\delta\mathbf{A} + \delta\mathbf{A}^T)$ . Deux conditions nécessaires pour la positivité de ces deux matrices est  $\mu = \frac{1}{\|\delta\mathbf{A}\mathbf{u}\|} > 0$ , qui est toujours respecté, et  $\nu = \frac{1}{\|\delta\mathbf{A}\mathbf{u}\|} > 0$ , qui est satisfaite si  $\hat{d}^* > 0$ . De plus, les matrices  $\mathbf{S}_{11}$  et  $\mathbf{S}_{22}$  sont positives seulement si :

$$\mathbf{S}' = (\delta\mathbf{A} + \delta\mathbf{A}^T) = \begin{bmatrix} \delta\mathbf{A}_{11} + \delta\mathbf{A}_{11}^T & \delta\mathbf{p}_0 \\ \delta\mathbf{p}_0^T & 2 \end{bmatrix} > 0 \quad (\text{B.124})$$

Cette matrice symétrique est positive si et seulement si les valeurs propres de  $\delta\mathbf{A}_{11} + \delta\mathbf{A}_{11}^T$  sont positives, et donc si et seulement si sa valeur propre minimale, appelé  $\sigma$ , est positive :

$$\sigma = \frac{\alpha_u}{\hat{\alpha}_u} + \frac{\alpha_v}{\hat{\alpha}_v} - \sqrt{\left(\frac{\alpha_u}{\hat{\alpha}_u} - \frac{\alpha_v}{\hat{\alpha}_v}\right)^2 + \left(\frac{\alpha_{uv}}{\hat{\alpha}_u} - \frac{\hat{\alpha}_{uv} \alpha_v}{\hat{\alpha}_u \hat{\alpha}_v}\right)^2} > 0 \quad (\text{B.125})$$

On retrouve donc ici la condition (3.126). On va supposer maintenant que cette condition est satisfaite. La matrice  $\mathbf{S}'$  peut donc être minorée :

$$\mathbf{xS}'\mathbf{x} \geq \mathbf{x}^T \mathbf{S}'' \mathbf{x}, \quad \forall \mathbf{x} \quad (\text{B.126})$$

où :

$$\mathbf{S}'' = \begin{bmatrix} \sigma \mathbf{I}_2 & \delta\mathbf{p}_0 \\ \delta\mathbf{p}_0^T & 2 \end{bmatrix} \quad (\text{B.127})$$

Si la matrice  $\mathbf{S}''$  est positive alors la matrice  $\mathbf{S}'$  l'est aussi.

Encore une fois, la matrice  $\mathbf{S}''$  est positive si sa valeur propre minimale  $\sigma_1$  est positive :

$$\sigma_1 = \sigma + 1 - \sqrt{(\sigma - 1)^2 + \|\delta\mathbf{p}_0\|^2} > 0 \quad (\text{B.128})$$

ce qui donne la condition (3.127). Par conséquent, si  $\sigma > 0$  et  $\sigma_2 > 0$  alors  $\mathbf{S}_{11} > 0$  et  $\mathbf{S}_{22} > 0$ , et on obtient :

$$\mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 \geq \nu \mathbf{e}_1^T \mathbf{S}' \mathbf{e}_1 \geq \nu \sigma_1 \|\mathbf{e}_1\|^2 \quad (\text{B.129})$$

$$\mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 \geq \mu \mathbf{e}_2^T \mathbf{S}' \mathbf{e}_2 \geq \mu \sigma_1 \|\mathbf{e}_2\|^2 \quad (\text{B.130})$$

- Considérons maintenant la matrice  $\mathbf{S}_{12}$  :

$$\mathbf{S}_{12} = \mu([\mathbf{x}_r - \nu\hat{\mathbf{x}}_r]_{\times}) \delta\mathbf{A} = \mu([\mathbf{x}_r - \nu\delta\mathbf{A}\mathbf{x}_r]_{\times}) \delta\mathbf{A} \quad (\text{B.131})$$

La norme  $l_2$  de  $\mathbf{S}_{12}$  peut être majorée de la manière suivante :

$$\|\mathbf{S}_{12}\| \leq \mu (\nu \|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|) \|\mathbf{x}_r\| \|\delta\mathbf{A}\| \quad (\text{B.132})$$

d'où :

$$\begin{aligned} \|\mathbf{S}_{12}\| &\leq \mu \|\delta\mathbf{A}\| (\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|) \|\mathbf{x}_r\| \\ &\leq \mu \frac{Z}{Z^*} \|\delta\mathbf{A}\| (\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|) \|\mathbf{m}\| \end{aligned} \quad (\text{B.133})$$

On note  $g(\gamma) = \|\mathbf{m}\| = \sqrt{1 + \gamma^2}$ . On peut remarquer que cette fonction ne s'annule jamais ( $g(\gamma) \neq 0 \ \forall \gamma$ ) et qu'elle est monotone croissante en fonction de la variable positive  $\gamma$ . Par conséquent, on peut encore majorer la norme de  $S_{12}$  :

$$\|\mathbf{S}_{12}\| \leq \mu \frac{Z}{Z^*} (\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|) \|\delta\mathbf{A}\| g(\bar{\gamma}) \quad (\text{B.134})$$

On peut finalement trouver la dernière condition pour la positivité de la matrice  $\mathbf{S}$ . L'inégalité (B.123) est vérifiée si :

$$\frac{1}{2} \mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 + \frac{1}{2} \mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 > |\mathbf{e}_1^T \mathbf{S}_{12} \mathbf{e}_2| \quad (\text{B.135})$$

À partir de (B.129) et (B.130), on a :

$$\frac{1}{2} \mathbf{e}_1^T \mathbf{S}_{11} \mathbf{e}_1 + \frac{1}{2} \mathbf{e}_2^T \mathbf{S}_{22} \mathbf{e}_2 \geq \frac{1}{2} \nu \sigma_1 \|\mathbf{e}_1\|^2 + \frac{1}{2} \mu \sigma_1 \|\mathbf{e}_2\|^2 \quad (\text{B.136})$$

et, à partir de (B.134), on a :

$$|\mathbf{e}_1^T \mathbf{S}_{12} \mathbf{e}_2| \leq \mu \frac{Z}{Z^*} (\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|) \|\delta\mathbf{A}\| g(\bar{\gamma}) \|\mathbf{e}_1\| \|\mathbf{e}_2\| \quad (\text{B.137})$$

L'inégalité (B.135) est donc vérifiée si :

$$\frac{1}{2} \nu \sigma_1 \|\mathbf{e}_1\|^2 + \frac{1}{2} \mu \sigma_1 \|\mathbf{e}_2\|^2 > \mu \frac{Z}{Z^*} (\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|) \|\delta\mathbf{A}\| g(\bar{\gamma}) \|\mathbf{e}_1\| \|\mathbf{e}_2\| \quad (\text{B.138})$$

Cette inégalité a la solution suivante  $\forall \mathbf{e}_1$  et  $\forall \mathbf{e}_2$  (avec  $\mathbf{e} \neq 0$ ) :

$$\left( \mu \frac{Z}{Z^*} (\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|) \|\delta\mathbf{A}\| g(\bar{\gamma}) \right)^2 < 4 \left( \frac{1}{2} \nu \sigma_1 \right) \left( \frac{1}{2} \mu \sigma_1 \right) \quad (\text{B.139})$$

qui, finalement, peut simplement s'écrire :

$$\mu \left( \frac{Z}{Z^*} \right)^2 (\nu\|\mathbf{I} - \delta\mathbf{A}\| + |1 - \nu|)^2 \|\delta\mathbf{A}\|^2 g^2(\bar{\gamma}) < \nu \sigma_1^2 \quad (\text{B.140})$$

qui est la condition (3.128).

### B.3.3 Démonstration du théorème 9

On démontre ici la stabilité asymptotique locale de l'asservissement visuel 2D 3/4 avec une commande classique et en présence d'erreurs de calibration sur les paramètres caméra et caméra/effecteur.

### Preuve du théorème 9

Les valeurs propres de la matrice  $\mathbf{Q}(\mathbf{e}^\circ)$  sont positives si :

$$\delta\mathbf{R} \mathbf{E}_\nu = \nu \delta\mathbf{R} \delta\mathbf{A} > 0 \quad (\text{B.141})$$

$$\delta\mathbf{R} \mathbf{E}_\omega = \mu \delta\mathbf{R} \delta\mathbf{A} > 0 \quad (\text{B.142})$$

car  $\mathbf{L}_\omega(\mathbf{e}^\circ) = \mathbf{I}$ . Si  $\nu > 0$  (ce qui implique naturellement que  $\frac{\hat{z}^*}{z^*} > 0$ , c'est-à-dire  $\frac{\hat{d}^*}{d^*} > 0$ ) et comme  $\mu > 0$ , les deux conditions sont vérifiées si :

$$\delta\mathbf{R} \delta\mathbf{A} > 0 \quad (\text{B.143})$$

Considérons chaque matrice du produit (B.143) :

- $\delta\mathbf{R} > 0$  si l'angle de rotation  $\delta\theta < \frac{\pi}{2}$
- $\delta\mathbf{A} > 0$  si les conditions (3.44) et (3.46) sont vérifiées.

Sur la base de l'étude effectuée dans le paragraphe (B.1.2), le produit des deux matrices est positif s'il existe deux angles  $\theta_1$  et  $\theta_2$  associés respectivement aux matrices  $\delta\mathbf{R}$  et  $\delta\mathbf{A}$  tels que  $|\theta_1| + |\theta_2| < \frac{\pi}{2}$  et tels que :

- dans le cas de la matrice de rotation  $\delta\mathbf{R}$  :

$$\frac{\rho^{-1}((\delta\mathbf{R} + \delta\mathbf{R}^T)^{-1})}{2\sqrt{\rho(\delta\mathbf{R}^T\mathbf{R})}} = \cos(\delta\theta) \geq \cos(\theta_1) \quad (\text{B.144})$$

- dans le cas de la matrice  $\delta\mathbf{A}$  :

$$\frac{\rho^{-1}((\delta\mathbf{A} + \delta\mathbf{A}^T)^{-1})}{2\sqrt{\rho(\delta\mathbf{A}^T\mathbf{A})}} \geq \cos(\theta_2) \quad (\text{B.145})$$

# Annexe C

## Asservissement visuel avec contrôle de la translation de la caméra

Dans certaines applications, il est souhaitable de contrôler autant que possible la trajectoire que doit suivre la caméra pour réaliser son positionnement. Une translation dans un repère fixe semble donc particulièrement appropriée car elle permettra à la caméra d'aller en ligne droite de sa position courante à sa position désirée sous l'hypothèse d'une calibration parfaite du système [Mezouar 98]. Comme nous l'avons déjà vu, si on utilise alors comme mesure la rotation dans l'espace cartésien, il n'y a aucun contrôle dans l'image et la cible peut sortir du champ de vue de la caméra, même si la calibration du système est parfaite. Nous proposons ici une méthode pour introduire un contrôle direct dans l'image tout en contrôlant la translation dans l'espace cartésien.

### C.1 Modélisation

Comme on l'a vu dans le Chapitre 1, la dérivée de la position de  $\mathcal{F}_d$  exprimée dans  $\mathcal{F}_c$  peut être écrite en fonction du torseur cinématique de la caméra :

$${}^c\dot{\mathbf{t}}_d = -\mathbf{v} - [\boldsymbol{\omega}]_{\times} {}^c\mathbf{t}_d \quad (\text{C.1})$$

Considérons maintenant la position de  $\mathcal{F}_c$  exprimée dans  $\mathcal{F}_d$  :

$${}^d\mathbf{t}_c = -{}^c\mathbf{R}_d^T {}^c\mathbf{t}_d \quad (\text{C.2})$$

En dérivant cette dernière équation on obtient :

$${}^d\dot{\mathbf{t}}_c = -{}^c\dot{\mathbf{R}}_d^T {}^c\mathbf{t}_d - {}^c\mathbf{R}_d^T \dot{{}^c\mathbf{t}}_d \quad (\text{C.3})$$

En injectant (C.1) dans (C.3) on obtient :

$${}^d\dot{\mathbf{t}}_c = -{}^c\dot{\mathbf{R}}_d^T {}^c\mathbf{t}_d + {}^c\mathbf{R}_d^T \mathbf{v} + {}^c\mathbf{R}_d^T [\boldsymbol{\omega}]_{\times} {}^c\mathbf{t}_d = {}^c\mathbf{R}_d^T \mathbf{v} + {}^c\mathbf{R}_d^T ([\boldsymbol{\omega}]_{\times} - {}^c\mathbf{R}_d {}^c\dot{\mathbf{R}}_d^T) {}^c\mathbf{t}_d \quad (\text{C.4})$$

Comme par définition  $[\boldsymbol{\omega}]_{\times} = {}^d\dot{\mathbf{R}}_c {}^d\mathbf{R}_c^T = -{}^c\dot{\mathbf{R}}_d {}^c\mathbf{R}_d^T = {}^c\mathbf{R}_d {}^c\dot{\mathbf{R}}_d^T$ , on obtient finalement :

$${}^d\dot{\mathbf{t}}_c = {}^c\mathbf{R}_d^T \mathbf{v} = {}^d\mathbf{R}_c \mathbf{v} \quad (\text{C.5})$$

La boucle de commande en translation est donc découplée de celle en rotation. Afin de contrôler les trois degrés de liberté en rotation tout en gardant un contrôle dans l'image nous utilisons les coordonnées  $x$  et  $y$  d'un point de référence dans l'image et la rotation  $u_z\theta$  autour de l'axe  $\vec{z}$  estimée à partir de la matrice  ${}^d\mathbf{R}_c$ . On obtient alors (en supposant ici  $\hat{\mathbf{u}} \approx 0$  pour simplifier l'écriture) :

$$\begin{bmatrix} x \\ y \\ u_z\theta \end{bmatrix} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & (1+y^2) & -xy & -x \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \quad (\text{C.6})$$

## C.2 Loi de commande

Nous avons choisi tout naturellement la fonction de tâche suivante :

$$\mathbf{e} = \begin{bmatrix} {}^d\mathbf{t}_c \\ x \\ y \\ u_z\theta \end{bmatrix} \quad (\text{C.7})$$

À partir des équations (C.5) et (C.6), on obtient une matrice jacobienne de la tâche triangulaire inférieure par bloc :

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \quad (\text{C.8})$$

où :

$$\mathbf{L}_{11} = {}^c\mathbf{R}_d^T \quad (\text{C.9})$$

$$\mathbf{L}_{21} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} \\ 0 & -\frac{1}{Z} & \frac{y}{Z} \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{C.10})$$

$$\mathbf{L}_{22} = \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{C.11})$$

La loi de commande correspondant à une décroissance exponentielle de la fonction de tâche s'écrit donc :

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = -\lambda \hat{\mathbf{L}}^{-1} \hat{\mathbf{e}} \quad (\text{C.12})$$

où la matrice  $\widehat{\mathbf{L}}^{-1}$  peut être facilement calculée :

$$\widehat{\mathbf{L}}^{-1} = \begin{bmatrix} \widehat{\mathbf{L}}_{11}^{-1} & 0 \\ -\widehat{\mathbf{L}}_{22}^{-1}\widehat{\mathbf{L}}_{21}\widehat{\mathbf{L}}_{11}^{-1} & \widehat{\mathbf{L}}_{22}^{-1} \end{bmatrix} \quad (\text{C.13})$$

où :

$$\widehat{\mathbf{L}}_{11}^{-1} = {}^c\widehat{\mathbf{R}}_d \quad (\text{C.14})$$

et :

$$\widehat{\mathbf{L}}_{22}^{-1} = \frac{1}{1 + \widehat{x}^2 + \widehat{y}^2} \begin{bmatrix} -\widehat{x}\widehat{y} & (1 + \widehat{x}^2) & \widehat{x}(1 + \widehat{x}^2 + \widehat{y}^2) \\ -(1 + \widehat{y}^2) & \widehat{x}\widehat{y} & \widehat{y}(1 + \widehat{x}^2 + \widehat{y}^2) \\ 0 & 0 & (1 + \widehat{x}^2 + \widehat{y}^2) \end{bmatrix} \quad (\text{C.15})$$

Le jacobien inverse utilisé dans la commande est donc toujours de rang plein et ne présente aucune singularité dans l'espace de travail défini par  $0 < Z < \infty$ .





# Bibliographie

- [Allen 93a] P. K. Allen, A. Timcenko, Yoshimi B., 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.
- [Allen 93b] P. K. Allen, A. Timcenko, B. Yoshimi, P. Michelman. – Hand-eye coordination for robotic tracking and grasping. *Visual servoing*, éd. par K. Hashimoto, pp. 33–69. – World Scientific Press, 1993.
- [Allezard 97] N. Allezard, M. Dhôme. – Localisation automatique d'un objet texturé. – *Journées ORASIS 97*, Nice, 1997.
- [Aloimonos 87] Y. Aloimonos, I. Weiss, A. Bandopadhyay. – Active vision. *Int. Journal of Computer Vision*, 1(4):333–356, Janvier 1987.
- [Bajcsy 88] R. Bajcsy. – Active perception. *Proc. of the IEEE*, 76(8):996–1005, Août 1988.
- [Basri 98] R. Basri, E. Rivlin, I. Shimshoni. – Visual homing: Surfing on the epipoles. – *IEEE Int. Conf. on Computer Vision*, pp. 863–869, Bombay, Inde, Janvier 1998.
- [Bensalah 95] F. Bensalah, F. Chaumette. – Compensation of abrupt motion changes in target tracking by visual servoing. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, vol. 1, pp. 181–187, Pittsburgh, Pennsylvannie, Août 1995.
- [Berry 97] F. Berry, P. Martinet, J. Gallice. – Trajectory generation by visual servoing. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'97*, pp. 1066–1072, Grenoble, France, Septembre 1997.
- [Bien 93] Z. Bien, W. Jang, J. Park. – Characterization and use of feature-jacobian matrix for visual servoing. *Visual servoing*, éd. par K. Hashimoto, pp. 317–363. – World Scientific Press, 1993.
- [Bishop 94] B. Bishop, S. Hutchinson, M. Spong. – On the performance of state estimation for visual servo systems. – *IEEE Int. Conf. on Robotics and Automation*, pp. 168–173, 1994.
- [Boufama 95] B. Boufama, R. Mohr. – Epipole and fundamental matrix estimation using the virtual parallax property. – *IEEE Int. Conf. on Computer Vision*, pp. 1030–1036, Cambridge, USA, 1995.

- [Chaumette 90] F. Chaumette. – *La relation vision-commande: théorie et application à des tâches robotiques*. – Thèse de l'Université de Rennes I, IRISA, Juillet 1990.
- [Chaumette 91] F. Chaumette, P. Rives, B. Espiau. – Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. *IEEE Int. Conf. on Robotics and Automation*, 3:2248–2253, Avril 1991.
- [Chaumette 93] F. Chaumette, P. Rives, B. Espiau. – Classification and realization of the different vision-based tasks. *Visual servoing*, éd. par K. Hashimoto, pp. 199–228. – World Scientific Press, 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 D. Kriegman, G. Hager et A. Morse. – LNCIS 237, pp. 66–78, Springer Verlag, 1998.
- [Chen 94] W. Z. Chen, U. A. Korde, S. B. Skaar. – Position control experiments using vision. *Int. Journal of Robotics Research*, 13(3):199–208, Juin 1994.
- [Cipolla 97] R. Cipolla, N. Hollinghurst. – Visually guided grasping in unstructured environment. *Robotics and Autonomous Systems*, 19:337–346, 1997.
- [Corke 93] P. I. Corke. – Visual control of robot manipulators - A review. *Visual servoing*, éd. par K. Hashimoto, pp. 1–31. – World Scientific Press, 1993.
- [Corke 96] P. I. Corke, Malcolm C. Good. – Dynamic effects in visual closed-loop systems. *IEEE Trans. on Robotics and Automation*, 12(5):671–683, Octobre 1996.
- [Couapel 95] B. Couapel, K. Bainian. – Stereo vision with the use of a virtual plane in the space. *Chinese Journal of Electronics*, 4(2):32–39, Avril 1995.
- [Criminisi 98] A. Criminisi, I. Reid, A. Zisserman. – Duality, rigidity and planar parallax. – *European Conf. on Computer Vision, ECCV'98*, vol. 2, pp. 846–861, Fribourg, Allemagne, Juin 1998.
- [Daucher 97] N. Daucher, M. Dhome, J.T. Lapresté, G. Rives. – Speed command of a robotic system by monocular pose estimate. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 55–62, Grenoble, France, Septembre 1997.
- [Dementhon 95] D Dementhon, L. S. Davis. – Model-based object pose in 25 lines of code. *Int. Journal of Computer Vision*, 15(1/2):123–141, Juin 1995.
- [Deriche 94] R. Deriche, Z. Zhang, Q.-T. Luong, O. Faugeras. – Robust recovery of the epipolar geometry for an uncalibrated stereo rig. – *European Conf. on Computer Vision, ECCV'94*, Stockhölme, Suède, 1994.

- [Dornaika 95] F. Dornaika. – *Contribution à l'intégration vision/robotique: calibration, localisation et asservissement*. – Thèse de l'Institut National Polytechnique de Grenoble, Septembre 1995.
- [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.
- [Espiau 93] B. Espiau. – Effect of camera calibration errors on visual servoing in robotics. – *3rd Int. Symposium on Experimental Robotics*, Kyoto, Japon, Octobre 1993.
- [Faugeras 88] O. Faugeras, F. Lustman. – Motion and structure from motion in a piecewise planar environment. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.
- [Faugeras 93] O. Faugeras. – *Three-dimensionnal computer vision: a geometric viewpoint*. – MIT Press, Cambridge, Massachusetts, 1993.
- [Faugeras 98] O. Faugeras. – From geometry to variational calculus: theory and applications of three-dimensional vision. – *RFIA*, vol. 1, pp. 15–34, Clermont-Ferrand, France, Janvier 1998.
- [Feddemma 89] J. T. Feddemma, O. R. Mitchell. – Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700, Octobre 1989.
- [Feddemma 90] J. T. Feddemma, C.S. George Lee. – Adaptive 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, Octobre 1990.
- [Feddemma 93] J. T. Feddemma, C. S. G. Lee, O. R. Mitchell. – Feature-based visual servoing of robotic systems. *Visual servoing*, éd. par K. Hashimoto, pp. 105–164. – World Scientific Press, 1993.
- [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, Louvain, Belgique, Mai 1998.
- [Grosso 96] E. Grosso, G. Metta, A. Oddera, G. Sandini. – Robust visual servoing in 3-d reaching tasks. *IEEE Trans. on Robotics and Automation*, 12(5):732–742, Octobre 1996.
- [Hager 95] G. D. Hager. – Calibration-free visual control using projective invariance. – *IEEE Int. Conf. on Computer Vision*, pp. 1009–1015, Cambridge, USA, Juin 1995.
- [Hager 97] G. D. Hager, Z. Dodds. – A projective framework for constructing accurate hand-eye systems. – *Workshop on New Trends In Image-Based Robot Servoing (IROS'97)*, pp. 71–82, Grenoble, France, Septembre 1997.
- [Hartley 92] R. I. Hartley. – Estimation of relative camera positions for uncalibrated cameras. – *European Conf. on Computer Vision, ECCV '92*, Santa Margherita Ligure, Italie, LNCS 588, pp. 579–587, Mai 1992.

- [Hartley 97] R. I. Hartley. – In defense of the eight-point algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(6):580–593, Juin 1997.
- [Hashimoto 91] K. Hashimoto, T. Kimoto, T. Ebine, H. Kimura. – Manipulator control with image-based visual servo. – *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2267–2271, Sacramento, Californie, Avril 1991.
- [Hashimoto 93a] K. Hashimoto. – *Visual Servoing: Real Time Control of Robot manipulators based on visual sensory feedback*. – World Scientific Press, vol. 7 of *World Scientific Series in Robotics and Automated Systems*, Singapore, 1993.
- [Hashimoto 93b] K. Hashimoto, H. Kimura. – LQ optimal and nonlinear approaches to visual servoing. *Visual servoing*, éd. par K. Hashimoto, pp. 165–198. – World Scientific Press, 1993.
- [Hashimoto 95] K. Hashimoto, H. Kimura. – Visual servoing with non linear observer. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 484–489, Nagoya, Japon, Avril 1995.
- [Hashimoto 98] K. Hashimoto, T. Noritsugu. – Performance and sensitivity in visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 2321–2326, Louvain, Belgique, Mai 1998.
- [Heeger 92] D. J. Heeger, A. D. Jepson. – Subspace methods for recovering rigid motion i: Algorithm and implementation. *Int. Journal of Computer Vision*, 7(2):95–117, Juin 1992.
- [Horaud 93] R. Horaud, O. Monga. – *Vision par ordinateur, outils fondamentaux*. – Hermès, Paris, France, 1993.
- [Horaud 97] R. Horaud, S. Christy, R. Mohr. – Euclidean reconstruction and affine camera calibration using controlled robot motions. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 1575–1582, Septembre 1997.
- [Hosoda 94] K. Hosoda, M. Asada. – Versatile visual servoing without knowledge of true jacobian. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'94*, pp. 186–193, Septembre 1994.
- [Huang 89] T. S. Huang, O. Faugeras. – Some properties of the E matrix in two-view motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, Décembre 1989.
- [Hutchinson 96] S. Hutchinson, G. D. Hager, P. I. Corke. – A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, Octobre 1996.
- [Irani 96] M. Irani, P. Anandan. – Parallax geometry of pairs of points for 3d scene analysis. – *European Conf. on Computer Vision, ECCV'96*, pp. 17–30, 1996.
- [Irani 98] M. Irani, P. Anandan, D. Weinshall. – From reference frames to reference planes: multi-view parallax geometry and applications. –

- European Conf. on Computer Vision, ECCV'98*, vol. 2, pp. 829–845, Fribourg, Allemagne, Juin 1998.
- [Isidori 95] A. Isidori. – *Nonlinear Control Systems*. – Springer, 1995.
- [Jerian 91] C.P. Jerian, R. Jain. – Structure from motion - A critical analysis of methods. *IEEE Trans. on Systems, Man, and Cybernetics*, 21(3):572–588, Mai/Juin 1991.
- [Jägersand 97] M. Jägersand, O. Fuentes, R. Nelson. – Experimental evaluation of uncalibrated visual servoing for precision manipulation. – *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2874–2880, Albuquerque, New Mexico, Avril 1997.
- [Kelly 96] R. Kelly. – Robust asymptotically stable visual servoing of planar robots. *IEEE Trans. on Robotics and Automation*, 12(5):759–766, Octobre 1996.
- [Kelly 97] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, F. Reyes. – A stable visual servo controller for camera-in-hand robotic systems. – *8<sup>th</sup> Int. Conf. on Advanced Robotics, ICAR'97*, pp. 117–123, Monterey, Californie, Juillet 1997.
- [Krautgartner 98] P. Krautgartner, M. Vincze. – Performance evaluation of vision-based control tasks. – *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2315–2320, Louvain, Belgique, Mai 1998.
- [Lan 97] Z.D. Lan. – *Méthodes robustes en vision: application aux appareillages visuels*. – Thèse de l'Institut National Polytechnique de Grenoble, 1997.
- [LonguetHiggins 81] H. C. Longuet-Higgins. – A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, Septembre 1981.
- [LonguetHiggins 84] H.C. Longuet-Higgins. – The reconstruction of a scene from two projections: configurations that defeat the 8-point algorithm. – *Proceedings of the 1st Conf. on Artificial intelligence applications*, pp. 395–397, Denver, 1984.
- [Lowe 91] D.G. Lowe. – Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, Mai 1991.
- [Luong 92] Q.-T. Luong. – *Matrice fondamentale et calibration visuelle sur l'environnement: vers une plus grande autonomie des systèmes robotiques*. – Thèse de l'Université de Paris-sud/Orsay, Décembre 1992.
- [Luong 96] Q.-T. Luong, O. Faugeras. – The fundamental matrix: Theory, algorithms, and stability analysis. *Int. Journal of Computer Vision*, 17(1):43–75, Janvier 1996.
- [Malis 97] E. Malis, F. Chaumette, S. Boudet. – Positioning a coarse-calibrated camera with respect to an unknown planar object by 2D 1/2 visual servoing. – *5th IFAC Symposium on Robot Control (SYROCO'97)*, vol. 2, pp. 517–523, Nantes, France, Septembre 1997.

- [Marchand 96] E. Marchand. – *Stratégies de perception par vision active pour la reconstruction et l'exploration de scènes statiques*. – Thèse de l'Université de Rennes I, IRISA, Juin 1996.
- [Marchand 98] E. Marchand, G. Hager. – Dynamic sensor planning in visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 1988–1993, Louvain, Belgique, Mai 1998.
- [Martinet 96] P. Martinet, J. Gallice, D. Khadraoui. – Vision based control law using 3D visual features. – *World Automation Congress, WAC'96, Robotics and Manufacturing Systems*, vol. 3, pp. 497–502, Montpellier, France, Mai 1996.
- [Martinet 97] P. Martinet, N. Daucher, J. Gallice, M. Dhome. – Robot control using monocular pose estimation. – *Workshop on New Trends In Image-Based Robot Servoing (IROS'97)*, pp. 1–12, Grenoble, France, Septembre 1997.
- [Maru 93] N. Maru, S. Yamada H. Kase, A. Nishikawa, F. Miyazaki. – Manipulator control by visual servoing with the stereo vision. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 1866–1870, Yokohama, Japon, Juillet 1993.
- [Mezouar 98] Y. Mezouar. – Comparaison des lois 3d et 2d. – *JJCR*, pp. 6–10, Clermont-Ferrand, France, Juillet 1998.
- [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.
- [Mohr 96] R. Mohr, B. Triggs. – Projective geometry for image analysis. – Int. Symposium of Photogrammetry and Remote Sensing, Juillet 1996.
- [Morel 98] G. Morel, E. Malis, S. Boudet. – Impedance based combination of visual and force control. – *IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1743–1748, Louvain, Belgique, Mai 1998.
- [Nelson 95] B. Nelson, P.K. Khosla. – Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance. *Int. Journal of Robotics Research*, 14(3):255–269, Juin 1995.
- [Nelson 96] B. J. Nelson, P. K. Khosla. – Vision resolvability for visually servoed manipulation. *Journal of Robotic Systems*, 13(2):73–93, 1996.
- [Papanikolopoulos 93a] N. P. Papanikolopoulos, P. K. Kosla. – Adaptive robot visual tracking: Theory and experiments. *IEEE Trans. on Robotics and Automation*, 38(3):429–445, Mars 1993.
- [Papanikolopoulos 93b] N. P. Papanikolopoulos, P. K. Kosla, T. Kanade. – Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. *IEEE Trans. on Robotics and Automation*, 9(1):14–35, Février 1993.
- [Papanikolopoulos 94] N. P. Papanikolopoulos, B. Nelson, P. K. Khosla. – Six degree-of-freedom hand/eye visual tracking with uncertain parameters. –

- IEEE Int. Conf. on Robotics and Automation*, pp. 174–179, San Diego, Californie, Mai 1994.
- [Piepmeier 98] J. A. Piepmeier, G. V. McMurray, H. Lipkin. – Tracking a moving target with model independent visual servoing: a predictive estimation approach. – *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2652–2657, Louvain, Belgique, Mai 1998.
- [Reyes 98] F. Reyes, R. Kelly. – Experimental evaluation of fixed-camera direct visual controllers on a direct-drive robot. – *IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 2327–2332, Louvain, Belgique, Mai 1998.
- [Rémy 97] S. Rémy, M. Dhome, J. M. Lavest, N. Daucher. – Hand-eye calibration. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 1057–1065, Grenoble, France, Septembre 1997.
- [Robert 93] L. Robert. – *Perception stéréoscopique de courbes et de surfaces tridimensionnelles. Application à la robotique mobile*. – Thèse de l'École Polytechnique, INRIA Sophia-Antipolis, Mars 1993.
- [Robert 95] L. Robert, C. Zeller, O. Faugeras. – *Applications of non-metric vision to some visually guided robotics tasks*. – Rapport de Recherche n. 2584, INRIA, Juin 1995.
- [Ruf 97] A. Ruf, R. Horaud. – Visual trajectories from uncalibrated stereo. – *Workshop on New Trends In Image-Based Robot Servoing (IROS'97)*, pp. 83–92, Grenoble, France, Septembre 1997.
- [Samson 91] C. Samson, M. Le Borgne, B. Espiau. – *Robot Control: the Task Function Approach*. – Clarendon Press, vol. 22 of *Oxford Engineering Science Series*, Oxford, Royaume Uni, 1991.
- [Sanderson 80] A. C. Sanderson, L. E. Weiss. – Image based visual servo control using relational graph error signal. – *Proc. of the Int. Conf. on Cybernetics and Society*, pp. 1074–1077, Cambridge, MA, Octobre 1980.
- [Sanderson 83] A. C. Sanderson, L. E. Weiss. – Adaptive visual servo control of robots. – *Robot Vision*, éd. par A. Pugh, pp. 107–116, Bedford, UK, 1983.
- [Schmid 97] C. Schmid, R. Mohr. – Local greyvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1997.
- [Sharma 94] R. Sharma, S. Hutchinson. – On the observability of robot motion under active camera control. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 162–167, San Diego, Californie, Mai 1994.
- [Sharma 95] R. Sharma, S. Hutchinson. – Optimizing hand/eye configuration for visual-servo systems. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 172–177, Nagoya, Japon, 1995.
- [Shashua 96] A. Shashua, N. Navab. – Relative affine structure: canonical model for 3d from 2d geometry and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):873–883, Septembre 1996.



- [Skaar 87] S. B. Skaar, W. H. Brockman, R. Hanson. – Camera-space manipulation. *Int. Journal of Robotics Research*, 6(4):20–32, Décembre 1987.
- [Skaar 90] S. B. Skaar, W. H. Brockman, W. S. Jang. – Three-dimensional camera space manipulation. *Int. Journal of Robotics Research*, 9(4):22–39, Août 1990.
- [Suanto 97] H. Suanto, R. Sharma, V. Varma. – Image-based autodocking without calibration. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 974–979, Albuquerque, New Mexico, Avril 1997.
- [Suh 93] I. H. Suh, T. W. Kim. – Visual servoing of robot manipulators by fuzzy membership function based neural networks. *Visual servoing*, éd. par K. Hashimoto, pp. 285–315. – World Scientific Press, 1993.
- [Takegaki 81] M. Takegaki, S. Arimoto. – A new feedback method for dynamic control of manipulators. – *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 103, pp. 119–125, Juin 1981.
- [Tonko 97] M. Tonko, J. Shurmann, K. Schafer, H.-H. Nagel. – Visually servoed gripping of a used car battery. – *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 49–54, Grenoble, France, Septembre 1997.
- [Torr 98] P. Torr, A. W. Fitzgibbon, A. Zisserman. – Maintaining multiple motion model hypotheses over many views to recover matching and structure. – *IEEE Int. Conf. on Computer Vision*, pp. 485–491, Bombay, Indie, Janvier 1998.
- [Tsai 84] R. Y. Tsai, T. S. Huang. – Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):13–27, Janvier 1984.
- [Vieville 96] T. Vieville, C. Zeller, L. Robert. – Using collineations to compute motion and structure in an uncalibrated image sequence. *Int. Journal of Computer Vision*, 20(3):213–242, 1996.
- [Weiss 87] L. E. Weiss, A. C. Sanderson, C. P. Neuman. – Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417, Octobre 1987.
- [Wijesoma 93] S. W. Wijesoma, D.F.H. Wolfe, R.J. Richards. – Eye-to-hand coordination for vision-guided robot control applications. *Int. Journal of Robotics Research*, 12(1):65–78, Février 1993.
- [Wiles 96] C. Wiles, M. Brady. – On the appropriateness of camera models. – *European Conf. on Computer Vision - ECCV '96*, pp. 228–237. Springer, Avril 1996.
- [Wilson 96] W. J. Wilson, C. C. W. Hulls, G. S. Bell. – Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696, Octobre 1996.

- [Yoshimi 94] B. H. Yoshimi, P. K. Allen. – Active, uncalibrated visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 156–161, San Diego, Californie, Mai 1994.
- [Zeller 96a] C. Zeller, O. Faugeras. – *Camera Self-Calibration from Video Sequences: the Kruppa Equations Revisited*. – Rapport de Recherche n. 2793, INRIA, Février 1996.
- [Zeller 96b] Cyril Zeller. – *Calibration projective affine et euclidienne en vision par ordinateur*. – Thèse de l'École Polytechnique, INRIA Sophia-Antipolis, Février 1996.
- [Zhang 94] Z. Zhang, R. Deriche, O. Faugeras, Quang-Tuan Luong. – *A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry*. – Rapport de Recherche n. 2273, INRIA, Mai 1994.
- [Zhang 95] Z. Zhang, A. R. Hanson. – Scaled euclidean 3D reconstruction based on externally uncalibrated cameras. – *IEEE Symposium on Computer Vision*, Coral Gables, Floride, 1995.
- [Zhang 96] Z. Zhang. – On the epipolar geometry between two images with lens distortion. – *Int. Conf. on Pattern Recognition, ICPR'96*, vol. 1, pp. 407–411, Vienne, Autriche, Août 1996.

---

[The following text is extremely faint and largely illegible. It appears to be a list of references or a table of contents, but the specific content cannot be accurately transcribed.]

# Publications

## Journaux

- [1] E. Malis, F. Chaumette, S. Boudet. - 2D 1/2 visual servoing. - **Accepté à IEEE Transactions on Robotics and Automation.**
- [2] E. Malis, F. Chaumette, S. Boudet. - Camera displacement through the recovery of an homography: Application to 2D 1/2 visual servoing. - **Soumis à International Journal of Computer Vision.**

## Conférences internationales

- [3] E. Malis, F. Chaumette, S. Boudet. - 2D 1/2 visual servoing stability analysis with respect to camera calibration errors. - *11<sup>ème</sup> Congrès IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'98*, Volume 2, pages 691-697, Victoria, Canada, Octobre 1998.
- [4] E. Malis, F. Chaumette, S. Boudet. - Positioning a coarse-calibrated camera with respect to an unknown object by 2D 1/2 visual servoing. - *14<sup>ème</sup> Congrès IEEE International Conference on Robotics and Automation, ICRA'98*, Volume 2, pages 1352-1359, Leuven, Belgique, Mai 1998.
- [5] G. Morel, E. Malis, S. Boudet. - Impedance based combination of visual and force control. - *14<sup>ème</sup> Congrès IEEE International Conference on Robotics and Automation, ICRA'98*, Volume 2, pages 1743-1748, Leuven, Belgique, Mai 1998.

## Symposium et Workshop

- [6] E. Malis, F. Chaumette, and S. Boudet. - Positioning a coarse-calibrated camera with respect to an unknown planar object by 2D 1/2 visual servoing. - *IFAC Symposium on Robot Control, SYROCO'97*, Vol. 2, pp. 517-523, Nantes, France, Septembre 1997.
- [7] F. Chaumette, E. Malis, and S. Boudet. - 2D 1/2 visual servoing with respect to a planar object. - *Workshop on New Trends in Image-based Robot Servoing, IROS'97*, pp. 45-52, Grenoble, France, Septembre 1997.



## Abstract

Our work is concerned with vision and robotics. More precisely, it consists in the design of visual servoing methods that do not need the geometric model of the observed objects or a precise calibration of the system.

First, the problem of the Euclidean reconstruction (up to a scale factor), from two images of a unknown static object, has been studied. It has been highlighted that the best linear solution for a robust reconstruction of the camera displacement is to use an homography matrix. Thus, an algorithm for the linear estimation of this matrix has been proposed.

Then, two new visual servoing schemes, based on the partial Euclidean reconstruction, are proposed. The closed-loop system is analyzed in order to obtain the robustness domain of a proportional control law with respect to the system calibration errors. Later, it has been shown that, thanks to the decoupled structure of the considered systems, it is possible to considerably increase the robustness domain with an adaptive control law. The new methods have been tested on a Cartesian robot and on a redundant robot, and then compared to the classic ones. The obtained results confirm the improvement in the stability of our methods.

Finally, the visual servoing techniques have been extended to the use of several cameras observing different parts of the scene. The multi-camera visual servoing has been designed as a part of the task function approach. The particular choice of the task function allows us to simplify the design of the control law and the stability analysis. In the end, a positioning task of a cumbersome object has been realized using two cameras mounted on a robot arm.

**Keywords:** Vision, Robotics, Visual Servoing, Projective Reconstruction, Robust Control, Stability Analysis, Multi-Camera Visual Servoing.







## Résumé

Notre travail se situe dans le domaine de la vision robotique. Plus précisément, il a consisté à concevoir des méthodes d'asservissement visuel qui ne nécessitent ni la connaissance du modèle géométrique de l'objet observé, ni une calibration précise du système.

Nous avons tout d'abord étudié le problème de la reconstruction euclidienne (à un facteur d'échelle près) à partir de deux images d'un objet statique dont le modèle est inconnu. Nous avons mis en évidence que la meilleure solution linéaire pour obtenir une reconstruction robuste du déplacement de la caméra est d'utiliser une matrice d'homographie. Un algorithme pour l'estimation linéaire de celle-ci a été alors proposé.

Ensuite, nous avons proposé deux nouvelles méthodes d'asservissement visuel basées sur la reconstruction euclidienne partielle. Nous avons étudié le système en boucle fermée afin de déterminer analytiquement la robustesse des lois de commande aux incertitudes sur la calibration du système. Ensuite, nous avons montré que, grâce à la forme découplée des systèmes considérés, il est possible d'augmenter nettement le domaine de robustesse avec une adaptation des gains de la commande. Les nouvelles méthodes ont été testées sur un robot cartésien et un robot redondant, et comparées aux méthodes classiques. Les résultats obtenus confirment que nos schémas d'asservissement apportent une amélioration importante du domaine de convergence.

Enfin, nous avons étendu les schémas d'asservissement visuel à l'utilisation de plusieurs caméras n'observant pas la même partie de la scène. Nous avons modélisé l'asservissement multi-caméras dans le cadre de la commande référencée capteur. Le choix particulier de la fonction de tâche nous a permis de simplifier la conception de la commande et l'analyse de la stabilité. Nous avons finalement réalisé une tâche de positionnement d'un objet encombrant en utilisant deux caméras embarquées sur un robot manipulateur.

**Mots-Clefs :** Vision, Robotique, Asservissement visuel, Reconstruction projective, Commande robuste, Analyse de stabilité, Asservissement visuel multi-caméras.