

Asservissements visuels rapides d'un robot manipulateur à six degrés de liberté

Suivi de cible et de profilé

Jacques Gangloff

Asservissements visuels rapides d'un robot manipulateur à six degrés de liberté

Suivi de cible et de profilé

Asservissements visuels rapides d'un robot manipulateur à six degrés de liberté

Suivi de cible et de profilé

Jacques Gangloff

Thèse de l'Université Louis Pasteur

25 avril 2006

*Université Louis Pasteur
Ecole Nationale supérieure de Physique de Strasbourg*

REMERCIEMENTS

Je voudrais tout d'abord remercier Eric Ostertag, Professeur à l'Ecole Nationale Supérieure de Physique de Strasbourg pour m'avoir accueilli au sein du Groupe de Recherche en Automatique et Vision Robotique du LSIIT et m'avoir soutenu tout au long de ma thèse.

Je remercie très sincèrement François Chaumette et Jean Gallice et Pierre-Louis Wendel pour l'intérêt qu'ils ont témoigné à l'égard de mon travail en acceptant d'être rapporteurs.

J'ai été très honoré que Prof. Dr.-Ing. Gerd Hirzinger ai bien voulu prendre part au jury de ma thèse et je l'en remercie beaucoup.

C'est Gabriel Abba, Maître de conférences, qui m'a convaincu de me lancer dans ce travail. Je veux lui adresser mes plus profonds remerciements pour m'avoir guidé, conseillé et soutenu sans faille au cours des trois années que j'ai passé sous sa direction.

Je tiens à adresser des remerciements tout particuliers à Michel de Mathelin, Maître de conférences, pour l'investissement très important qu'il a mis dans l'encadrement de cette thèse. Sa grande rigueur, son perfectionnisme et ses vastes connaissances ont permis de baliser l'avancée de ce travail jusqu'à son aboutissement.

Mes plus chaleureux remerciements vont également à Hansjörg Sage. Dans le cadre de nos recherches nous avons pu développer une collaboration fructueuse qui déboucha sur plusieurs publications et une amitié sincère.

Je voudrais enfin remercier Guillaume Morel, Christophe Doignon et tous les autres membres du Groupe de Recherche en Automatique et Vision Robotique pour leur soutien et leurs conseils.

Je ne peux clore cette partie sans remercier ma famille et plus particulièrement mon épouse, Valérie, pour m'avoir soutenu et encouragé avec abnégation et tendresse.

TABLE DES MATIÈRES

1. Introduction	1
1.1 Les différents types d'asservissements par vision	3
1.1.1 Les asservissements où la caméra est extérieure	4
1.1.2 Les asservissements <i>eye in hand</i>	4
1.1.3 Les asservissements 3D	4
1.1.4 Les asservissements 2D	7
1.1.5 Les asservissements 2D 1/2	9
1.1.6 Les asservissements d2D/dt	10
1.1.7 Les asservissements visuels indirects	10
1.1.8 Les asservissements visuels directs	10
1.2 Les asservissements visuels rapides	11
1.2.1 Prédiction du mouvement de la cible	12
1.2.2 Prise en compte des dynamiques du manipulateur	13
1.3 Contributions de cette thèse	15
1.3.1 Le suivi de cible	16
1.3.2 Le suivi de profilé	16
1.4 Plan du mémoire	17
2. Modélisation de la caméra et du robot	19
2.1 Notions de base	19
2.1.1 Définition de l'attitude d'un repère	19
2.1.2 Définition des repères	20
2.2 Modélisation de la caméra	23
2.2.1 Modèle de projection d'un point dans l'image	23
2.2.2 Le capteur visuel	25
2.2.3 Compensation de la distorsion	26
2.2.4 Modélisation dynamique de la caméra	27
2.3 Modélisation d'un robot	28
2.3.1 Modélisation cinématique d'un robot	29
2.3.2 Modélisation dynamique d'un robot	29
2.4 Modélisation du robot SCEMI 6P01	30
2.4.1 Présentation générale	31
2.4.2 Modélisation géométrique	32
2.4.3 Le modèle dynamique	33
2.4.4 Identification du modèle dynamique d'un axe asservi en vitesse	40

3. Modélisation de l'asservissement visuel	41
3.1 Modélisation dynamique d'une boucle de vision	41
3.1.1 Définition du Jacobien de la caméra	42
3.1.2 Modélisation du dispositif de déplacement opérationnel virtuel	44
3.1.3 Commande de la boucle de vision	46
3.2 Le correcteur GPC	48
3.2.1 Mise en équation du système	48
3.2.2 La fonction de coût	49
3.2.3 Calcul des prédictions de la sortie	50
3.2.4 Calcul de la solution optimale	52
3.2.5 Implémentation du correcteur GPC	53
3.2.6 Réglage du correcteur GPC	54
3.2.7 Utilisation du GPC dans la boucle de vision	56
3.3 La plateforme expérimentale	57
3.3.1 Schéma général	57
3.3.2 Le capteur de vision	57
3.3.3 Le système d'acquisition d'images	59
3.3.4 Le système temps-réel	59
3.4 Simulation de l'asservissement par vision	59
3.4.1 Simulation du robot	60
3.4.2 Simulation de la caméra	60
3.4.3 Interface graphique de la simulation	60
4. Suivi de cible	63
4.1 Principe du suivi de cible	63
4.1.1 Définition de la grandeur de mesure de la boucle de vision	63
4.1.2 Commande de la boucle de vision	64
4.1.3 Suivi de cible par asservissement visuel 3D	66
4.1.4 Suivi de cible par asservissement visuel 2D	67
4.2 Simulations et expériences	70
4.2.1 Description de la cible	72
4.2.2 Traitements d'image	72
4.2.3 Influence du bruit sur les pixels : comparaison des approches 2D et 3D	73
4.2.4 Réponse de l'asservissement à un échelon de déplacement de la cible	75
4.2.5 Etude de la réponse à une trajectoire	80
4.3 Conclusion	86
5. Le suivi de profilé	87
5.1 Exploitation de l'information visuelle	88
5.1.1 Définition d'un profilé	88
5.1.2 Propriétés géométriques d'un profilé	89
5.1.3 Propriétés de la projection perspective d'une courbe 3D	90

5.1.4	Modélisation des primitives visuelles	90
5.1.5	Calcul de la position du profilé	91
5.2	Détermination de la section courante le long du profilé	97
5.2.1	Détermination de la position le long du profilé	97
5.2.2	Recherche de la section courante	97
5.3	Asservissement du suivi de profilé	102
5.3.1	La mesure de l'asservissement	102
5.3.2	Le contrôle de la boucle	104
5.4	Suivi de profilé avec prédiction de la trajectoire	104
5.4.1	Comparaison des 2 approches d'asservissement : l'approche classique et l'approche prédictive	105
5.4.2	Détermination des positions futures des sections courantes	105
5.4.3	Définition des mesures et des consignes	106
5.4.4	La boucle de contrôle	107
5.5	Simulations et expériences	107
5.5.1	Comparaison des stratégies GPC1 et PI	109
5.5.2	Etude de la précision de suivi en fonction de la vitesse	110
5.5.3	Comparaison des stratégies GPC1 et GPC2	113
5.6	Conclusion	117
6.	Conclusion générale	119
6.1	Contributions principales	119
6.2	Perspectives	120
Annexe		123
A.	Démonstrations	125
A.1	Démonstration du lemme 5.1	125
A.2	Démonstrations	126
B.	Résultats et calculs	129
B.1	Modèle géométrique du SCEMI	129
B.2	Fonctions de transfert en vitesse des axes du robot SCEMI	130
B.3	Définition de la décomposition en tangage, roulis et lacet	133
B.4	Calcul du Jacobien J_p	135
B.5	Calcul du Jacobien image dans le cas où les primitives sont des points	137
B.6	Calcul de l'attitude d'un objet grâce à la projection de n points coplanaires	139
B.7	Calcul récursif des équations Diophantiennes du GPC	141
B.7.1	Résolution récursive de l'équation Diophantienne (3.23)	141
B.7.2	Résolution récursive de l'équation Diophantienne (3.32)	143
B.8	Expression de K_1 , K_2 et K_3	146

C. Le logiciel	149
C.1 Le logiciel de contrôle de la boucle de vision	149
C.1.1 Cahier des charges	149
C.1.2 Choix du langage de programmation	150
C.1.3 Choix de l'ordinateur et du système d'exploitation	150
C.1.4 Choix de la structure logicielle	152
C.1.5 Organigramme du programme d'asservissement	153
C.1.6 Chronogramme de l'asservissement par vision	153

Chapitre 1

Introduction

L'adjonction de la faculté de vision aux robots, permet d'améliorer de manière significative leurs performances et permet le développement d'applications nouvelles. En effet, les tâches que doivent réaliser les robots peuvent toujours se ramener à des interactions précises avec leur environnement. Par exemple, lorsqu'il s'agit d'un bras manipulateur, le robot interagit avec une pièce et doit réaliser des mouvements précis par rapport à celle-ci. S'il s'agit d'un robot mobile, ce dernier doit se déplacer selon une trajectoire précise par rapport à son environnement. En l'absence de capteur donnant une mesure directe de la position par rapport aux objets environnants, le robot est «aveugle». Dans une telle situation, qui concerne d'ailleurs la majorité des robots, seule une tâche apprise au préalable et répétée dans un environnement connu et invariant peut être envisagée. La position relative par rapport à l'environnement est alors estimée de manière indirecte en fonction d'une mesure des mouvements du robot. Ainsi, une légère modification de la configuration des objets environnants par rapport à la référence peut entraîner de graves conséquences sur le déroulement de la tâche à accomplir. Par exemple, s'il s'agit de réaliser un travail de soudure au dixième de millimètre près et que l'objet à souder est décalé de 1 millimètre par rapport à la référence, tous les points de soudure seront décalés de cette même distance.

Divers capteurs peuvent être utilisés afin de permettre aux systèmes robotiques de pouvoir appréhender l'environnement. Les plus couramment utilisés sont les capteurs à ultrasons, les télémètres laser, les capteurs d'effort et les caméras.

Le capteur à ultrasons a l'avantage de pouvoir être utilisé dans un environnement où la visibilité est faible mais il fournit une information relativement imprécise. Il est souvent utilisé dans les applications de robotique mobile en association avec d'autres capteurs. Le télémètre laser donne quant à lui une image très précise de l'environnement, mais il nécessite un balayage de l'espace, ce qui est très coûteux en temps. Il est souvent utilisé en extérieur dans des environnements totalement inconnus et difficilement modélisables où il permet de reconstruire la géométrie du terrain même si ce dernier est très peu marqué. Le capteur d'effort fournit également au robot un

moyen d'appréhender un peu mieux son environnement. Il est souvent utilisé pour des applications de montage, et plus particulièrement pour des tâches d'insertion. Il permet d'avoir une information sur les forces et les couples d'interaction entre le robot (ou une pièce qui est dans la pince) et son environnement (voir par exemple [MMB98]). Ces trois capteurs ont la particularité de réaliser une mesure active sur l'environnement, c'est à dire qu'ils fournissent eux-mêmes l'énergie pour la grandeur à mesurer.

Une caméra exerce quant à elle une mesure passive, sans interaction avec l'environnement. C'est le capteur visuel dont les caractéristiques se rapprochent le plus de l'oeil humain. Elle permet d'obtenir, en une seule acquisition, une image de l'environnement situé dans son champ de vision. Sur ce point, elle est donc plus rapide que le télémètre laser. Mais, tout comme ce dernier, la caméra est également un capteur précis. Un traitement adéquat de l'image permet, par exemple, d'en extraire une mesure de la vitesse relative de la caméra par rapport à la scène vue par des techniques de flot optique [Pap92]. Par ailleurs, si on dispose d'un modèle géométrique de la scène, il est possible de déterminer avec une bonne précision la position de la caméra par rapport à cette scène [Cha90, DRLR91, FMZ91]. Cette position peut être reconstituée sans connaissance *a priori* de la scène dans le cas où plusieurs caméras sont utilisées.

La caméra est souvent montée sur l'organe terminal du robot, c'est à dire la partie du robot possédant le plus de degrés de liberté par rapport à la base. Pour un robot manipulateur, il s'agit de la pince ou de l'outil. Pour un robot mobile, elle peut être fixée directement sur le véhicule ou sur une tourelle orientable en tangage et roulis (de la même manière qu'un oeil humain) qui est elle-même solidaire du châssis du véhicule. La caméra est généralement montée directement sur le robot et observe l'environnement. Mais il arrive qu'elle soit utilisée pour visualiser le robot et son environnement. Dans ce cas, elle peut être fixe ou montée sur un autre robot [NK94]. Néanmoins, dans la littérature, les applications mettant en oeuvre un tel système restent marginales.

L'utilisation de l'information visuelle dans les systèmes robotiques a souvent été fortement contrainte par les ressources de puissance de calcul. En effet, l'exploitation de la vision requiert une phase de traitement d'image qui est très coûteuse en temps de calcul. Ainsi, la période de mesure des premières applications de vision se comptait en dizaines de secondes [SI73]. A cette époque, seule une exploitation en boucle ouverte de la mesure de vision était possible. La technique consistait à prendre une image de la scène, puis à la traiter, et enfin à déplacer le robot à l'endroit désiré en fonction de la mesure visuelle (cette approche est dénommée *look and move* en anglais). Outre son extrême lenteur à réagir à des modifications de l'environnement, cette technique est très sensible aux erreurs de modélisation de la caméra et du robot.

Au cours de ces dix dernières années, les progrès en matière de puissance de calcul des ordinateurs ont permis aux applications robotiques basées sur la vision de devenir de plus en plus «temps réel». La période de mesure des systèmes de vision diminuant (1 milliseconde pour les plus rapides actuellement [NI98]), il est désormais

possible d'inclure la mesure extraite de l'image dans une boucle d'asservissement. On parle alors d'asservissement par vision (*visual servoing* en anglais). La grandeur à asservir est celle qui est extraite de l'image, à savoir une position relative entre la caméra et son environnement (ou une vitesse relative dans le cas d'un traitement par flot optique). La tâche robotique à effectuer peut donc être redéfinie comme une trajectoire référencée par rapport à la scène. Tout mouvement de la scène agit comme une perturbation pour l'asservissement qui tend à l'annuler.

Ainsi, grâce à cet asservissement, il n'est plus nécessaire de référencer précisément la position des objets par rapport au manipulateur. Pour reprendre l'exemple de la tâche de soudure, une modification de la position de la pièce entraîne immédiatement un mouvement du robot tendant à replacer l'organe terminal dans la position désirée par rapport à la pièce. A la limite, si l'asservissement est assez rapide, on peut imaginer que cette tâche de soudure puisse être réalisée alors que la pièce est en mouvement par rapport au robot.

L'augmentation de la rapidité des systèmes d'asservissement par vision requiert une modélisation de la boucle de vision tenant compte des dynamiques du manipulateur et du capteur. Cette problématique est récente. Jusqu'à présent, dans la plupart des applications d'asservissement visuel, la dynamique du manipulateur était considérée comme instantanée par rapport à la bande passante de la mesure visuelle. Ainsi, elle pouvait être négligée et donc être modélisée par un transfert unitaire. Les asservissements visuels rapides (ceux dont la fréquence d'échantillonnage est supérieure à 50 Hz) requièrent une modélisation précise des dynamiques du manipulateur et du système de vision afin d'être contrôlés de manière optimale, c'est à dire en réalisant le meilleur compromis entre performance et stabilité.

Dans cette introduction, nous donnons un descriptif des principaux types d'asservissements visuels existants. Nous les avons classés en fonction de plusieurs critères pour faciliter les comparaisons. Nous consacrons ensuite une section aux asservissements visuels rapides en décrivant les principes de base qui gouvernent leur mise en oeuvre. Nous présentons les contributions apportées par notre travail à cette problématique. Enfin, nous donnons le plan de ce mémoire.

1.1 Les différents types d'asservissements par vision

Une classification des asservissements par vision a été établie par Sanderson *et al* dans [SW80]. Bien qu'elle ait été réalisée en 1980 et que la discipline a beaucoup évolué ces dernières années, elle reste encore valable dans ses grandes lignes. Nous la compléterons en tenant compte des développements les plus récents.

Cette classification s'effectue en prenant en compte 3 critères :

1. Le positionnement de la caméra :

- La caméra n'a pas de liaison mécanique avec le robot asservi par vision. Nous parlerons de *configuration où la caméra est extérieure* pour désigner ce type d'asservissement.
- La caméra est montée sur l'organe terminal du robot. Cette configuration est dénommée «*eye in hand*» dans la littérature anglo-saxonne.

2. Les grandeurs asservies :

- La référence et la mesure sont définies par une attitude. Un modèle géométrique de l'objet d'intérêt est nécessaire pour estimer la mesure. On parle alors d'asservissement «3D».
- La référence, la mesure et la loi de commande de l'asservissement sont définies dans le plan image. Dans ce cas, on parle d'asservissement référencé capteur ou plus prosaïquement d'asservissement «2D».
- La référence est une image mais la loi de commande est établie à partir de grandeurs exprimées dans le plan image et dans le repère caméra. Chaumette *et al.* ont baptisé ce mode de commande «2D 1/2» [Cha98, Mal98].
- La référence de l'asservissement est un champ de vitesse dans le plan image. Chaumette *et al.* [Cha98] ont donné à ce type d'asservissement l'appellation «d2D/dt».

3. Le type de commande appliquée au robot :

- Le robot est commandé *via* le contrôleur de position articulaire fournit par le constructeur du robot. On dénommera ce mode de fonctionnement *asservissement visuel indirect*.
- La boucle de vision fournit directement les références aux contrôleurs de vitesse ou de couple articulaire des axes du robot. Ce type d'asservissement est défini par *asservissement visuel direct*.

1.1.1 Les asservissements où la caméra est extérieure

La figure 1.1 décrit la configuration où la caméra est extérieure au robot qui est asservi. Elle est positionnée de manière à ce que l'organe terminal et les objets situés dans l'espace de travail du robot soient dans son champ de vision. Elle peut être fixe ou montée sur un autre système mécanique. Dans cette configuration, l'information visuelle permet non seulement la mesure de l'attitude ou du changement d'attitude d'un objet situé dans l'espace de travail du robot, mais également la mesure de l'attitude ou du changement d'attitude de l'organe terminal.

1.1.2 Les asservissements *eye in hand*

C'est la configuration la plus répandue. La caméra est attachée à l'organe terminal du manipulateur de manière à ce que les objets situés dans l'espace de travail soient dans son champ de vision (voir figure 1.2). Dans cette configuration, la caméra est utilisée pour mesurer l'attitude ou le changement d'attitude d'un objet par rapport à l'organe terminal du robot.

1.1.3 Les asservissements 3D

Dans un asservissement 3D, la référence est exprimée sous la forme d'une attitude. L'attitude d'un repère par rapport à un autre est définie par une translation et une rotation. Dans la configuration *eye in hand*, il s'agit de l'attitude p^* d'un repère

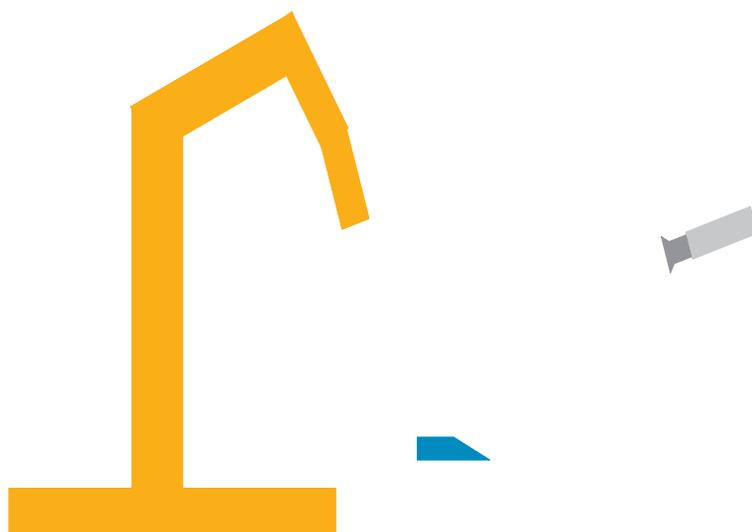


Fig. 1.1: Description de la configuration où la caméra est extérieure



Fig. 1.2: Description de la configuration *eye in hand*

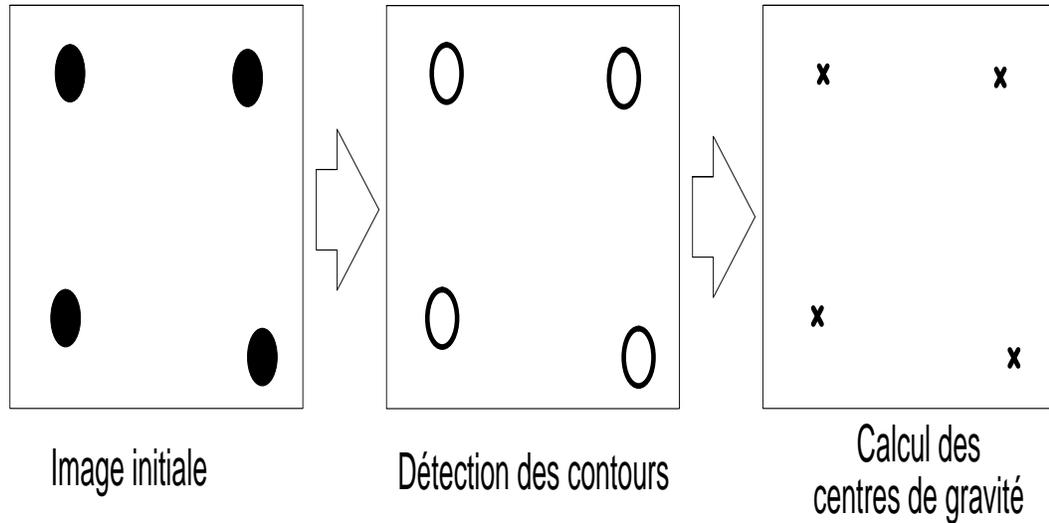


Fig. 1.3: Traitement d'image d'une cible constituée de disques coplanaires

lié à un objet vu par la caméra par rapport à un repère lié à l'organe terminal. La mesure utilisée dans l'asservissement est une estimation \hat{p} de l'attitude courante p entre l'organe terminal et l'objet.

Cette mesure s'obtient grâce aux primitives extraites de l'image et à un modèle géométrique de la cible. Mais cette mesure est très sensible aux erreurs de calibration de la caméra. Grâce à l'asservissement, $\hat{p} = p^*$ en régime établi. Mais, comme $p \neq \hat{p}$ à cause des erreurs de modélisation, la position réelle du robot par rapport à la cible est biaisée. Ce problème est néanmoins contournable en réalisant un apprentissage de la référence p^* . Dans ce cas, le robot est placé par un opérateur de manière à ce que $p = p^*$. Une estimation \hat{p}^* de la consigne est alors déterminée dans cette configuration. Si la consigne de l'asservissement est \hat{p}^* , on a donc en régime établi $\hat{p} = \hat{p}^*$. Comme, généralement, la relation qui lie l'attitude réelle p à son estimée mesurée \hat{p} est bijective, on en déduit que si $\hat{p}^* = \hat{p}$, alors $p = p^*$.

De nombreux articles traitent de techniques d'estimation de l'attitude entre une ou plusieurs caméras et un objet dont la géométrie est connue. La complexité des calculs nécessaires à cette estimation croît avec la complexité de l'objet. C'est pourquoi les objets sont souvent simplifiés au maximum dans les asservissements visuels. Par exemple dans [MDGD92], la cible est constituée d'un tétraèdre dont les sommets sont matérialisés par des points lumineux.

De nombreuses manipulations mettent en oeuvre une cible constituée uniquement de plusieurs disques coplanaires sur un fond uniforme [Mal98, HEK96]. Le traitement d'image se limite alors à une détection des contours des disques et à un calcul du centre de gravité des points de contour afin de déterminer le centre de chaque disque (*cf.* figure 1.3). Le modèle de la cible, à savoir la position relative de tous les centres des disques, est connu par le système de vision qui peut donc calculer une estimation de l'attitude de l'objet. Il a été montré qu'il est nécessaire de connaître la projection

d'au moins 4 primitives de type points pour pouvoir obtenir cette attitude de manière univoque (voir [Yua89] par exemple). Si seulement 3 projections sont disponibles, l'attitude ne peut être déterminée de manière unique. Néanmoins, la connaissance de l'attitude à un instant d'échantillonnage précédent permet, dans certains cas, d'estimer la nouvelle attitude en choisissant celle qui se rapproche le plus de la précédente.

Le problème de la reconstruction de l'attitude à partir de la projection de 3 ou 4 points est abordé par de nombreux auteurs. Dans certains cas, une solution analytique à ce problème est proposée [FB81, HLON91, DD92, HCL89, DRLR89]. D'autres auteurs proposent une solution par les moindres carrés [Low91, Gol94, LHF90].

Un des développements les plus récents en matière d'asservissement 3D est décrit dans [WHB96]. Une cible qui contient 5 trous circulaires non coplanaires est utilisée. Le suivi est réalisé suivant 5 degrés de liberté grâce à un bras manipulateur. Le système de vision utilisé est capable de mesurer la position de la projection des 5 trous dans une image binarisée de la cible à la cadence de 61Hz. Partant du constat que la binarisation d'une image est une opération introduisant beaucoup de bruit dans la détection des contours, un filtrage de Kalman de la mesure de l'attitude est réalisé. Ce dernier comprend un observateur très simple du mouvement de la cible par rapport à la caméra où la vitesse relative de déplacement est approximée par une constante. Le filtre de Kalman est également utilisé pour prédire la meilleure estimation¹ de la position future des projections des trous dans l'image. Cette estimation est utilisée par le dispositif de traitement d'image pour affiner la position des zones d'intérêt dans l'image. Il s'agit d'un asservissement visuel indirect car il utilise les asservissements de position des axes du robot. Le correcteur de la boucle de vision est du type Proportionnel Dérivé (PD) mais son réglage est réalisé expérimentalement, en l'absence de modélisation dynamique du manipulateur.

1.1.4 Les asservissements 2D

Dans un asservissement 2D, le signal de référence est exprimé sous la forme de primitives visuelles dans l'image. Une primitive est une forme géométrique élémentaire (point, segment de droite, portion d'ellipse,...). Elle sert à modéliser la projection d'un objet dans le plan image.

Le pionnier en matière d'asservissement 2D fut Weiss [WS87]. Il étudia le lien entre le torseur cinématique du repère de la caméra et les vitesses de déplacement des primitives dans l'image. Cette relation sera baptisée *Jacobien image* par la suite dans la littérature. Son travail fut développé entre autres par Feddema [FM89], notamment du point de vue expérimental. Espiau *et al.* proposèrent un cadre plus formel au concept de Jacobien image qu'ils baptisèrent *matrice d'interaction* dans [ECR92]. Dans cet article, les auteurs étendent le concept de Jacobien image à différents types de primitives visuelles : droites, plans, cercles, sphères. Ils y décrivent

¹ Au sens du minimum de la variance de l'erreur

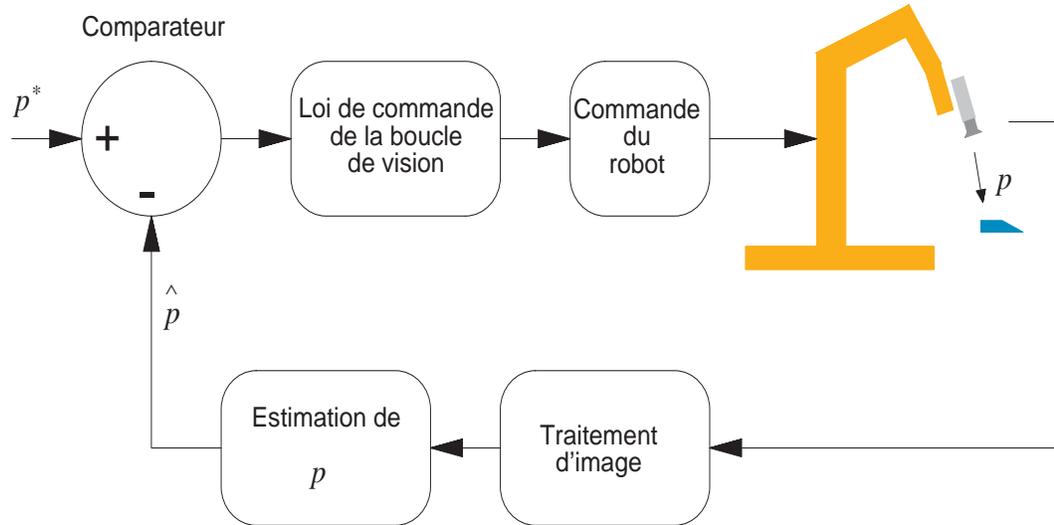


Fig. 1.4: Structure d'un asservissement visuel 3D

également une simulation de suivi dans laquelle la caméra doit se déplacer à vitesse constante au-dessus de la voie de droite d'un modèle de route constitué de 3 droites parallèles.

La plupart des travaux qui traitent de l'asservissement 2D utilisent des primitives constituées de points. Ces points peuvent par exemple être situés à l'intersection de segments dans l'image ou encore être extraits du centre de gravité de la projection de disques. En fait, les cibles les plus couramment rencontrées sont simplement constituées de plusieurs disques coplanaires. De telles cibles ont l'avantage de nécessiter un temps de traitement d'image faible.

Un des principaux avantages de l'asservissement 2D est qu'il n'est pas nécessaire de connaître un modèle de la cible. Dans le cas où les primitives sont des points, si on dispose d'une estimation de la profondeur de la cible (c'est à dire son éloignement par rapport à la caméra suivant l'axe optique), les informations contenues dans l'image suffisent à déterminer le déplacement de la cible par rapport à la caméra. Différentes techniques peuvent être utilisées pour estimer cette profondeur. Lorsqu'il s'agit d'une expérience de suivi de cible, on fait souvent 2 approximations sur la profondeur : on considère que celle-ci ne varie pas avec le temps et on estime que tous les points de la cible ont la même profondeur (la taille de la cible est souvent négligeable par rapport à la distance entre la cible et la caméra). Il est également possible de recourir à un algorithme adaptatif qui permet d'estimer en ligne la profondeur de la cible [PK93].

L'asservissement 2D est également adapté à des cibles complexes ne contenant aucune primitive simple. Une technique de flot optique² (*optical flow* en anglais) est alors utilisée pour déterminer le mouvement de zones d'intérêt dans l'image

² En toute rigueur, le flot optique consiste à calculer un champ dense de vitesses dans l'image ; il est cependant courant d'étendre cette notion à l'estimation du mouvement de zones d'intérêt dans l'image.

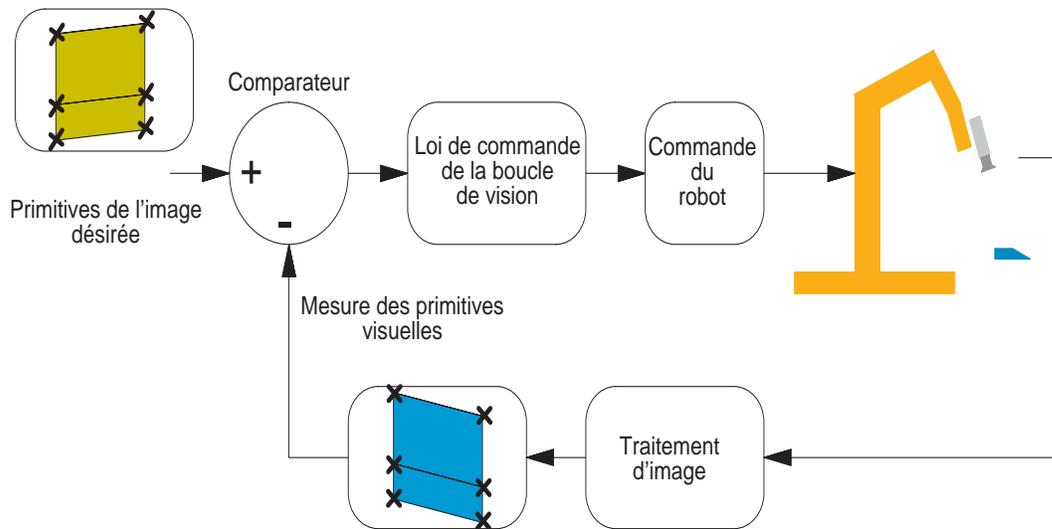


Fig. 1.5: Structure d'un asservissement visuel 2D

[PKK93, Pap92]. Dans ses grandes lignes, celle-ci consiste à minimiser une fonction de coût $\mathcal{O}(\Delta x, \Delta y)$:

$$\mathcal{O}(\Delta x, \Delta y) = \sum_{x=x_0}^{x_0+dx} \sum_{y=y_0}^{y_0+dy} [W_t(x, y) - W_0(x + \Delta x, y + \Delta y)]^2 \quad (1.1)$$

où $W_t(x, y)$ représente la luminance du pixel de coordonnées (x, y) dans une fenêtre rectangulaire W_t de dimension (dx, dy) englobant une région de l'image de la cible acquise à l'instant courant t . Le terme $W_0(x + \Delta x, y + \Delta y)$ représente la luminance du pixel de coordonnées $(x + \Delta x, y + \Delta y)$ dans une fenêtre de mêmes dimensions et position contenue dans l'image de référence. Ainsi, le minimum de $\mathcal{O}(\Delta x, \Delta y)$ est obtenu pour des valeurs de Δx et Δy représentant le déplacement de la région considérée entre les deux images. On obtient donc une approximation du déplacement moyen des pixels de cette région de l'image.

1.1.5 Les asservissements 2D 1/2

Récemment, Chaumette *et al.* [Cha98, MCB98] ont proposé une variante de la loi de commande habituellement utilisée, pour un asservissement dans l'image de la position d'une cible.

Les auteurs ont baptisé cet asservissement «2D 1/2» car il utilise une combinaison d'informations exprimées pour certaines d'entre elles dans l'image et pour d'autres dans le repère caméra. Avec cette technique, il est possible de contraindre la trajectoire d'un point de la cible à se projeter suivant une droite dans l'image. L'objectif est un meilleur contrôle de la position de la cible dans l'image lors de régimes transitoires pour des écarts importants entre la position courante de la ca-

méra et sa position de référence. (éviter entre autres qu'un des points ne sorte de l'image).

1.1.6 Les asservissements d2D/dt

Dans ce type d'asservissement, la grandeur asservie est une vitesse relative entre la caméra et la cible. La référence est définie dans le plan image par un champ de vitesse des points. La mesure peut être établie par des techniques classiques de flot optique (*cf.* section précédente).

Une fonction quadratique est souvent choisie pour modéliser le champ de vitesse [Cha98]. La loi de commande est définie par rapport aux paramètres de cette fonction.

Ce type d'asservissement est à un stade émergent. Actuellement, les traitements d'image qu'il nécessite sont encore très longs et fournissent des informations très bruitées. Des applications pratiques ont néanmoins été réalisées. Par exemple, un asservissement de l'alignement de l'axe optique d'une caméra animée d'un mouvement de translation suivant la direction de cette translation est décrit dans [SBC96].

1.1.7 Les asservissements visuels indirects

Dans un asservissement visuel indirect, le contrôleur du robot fournit par le constructeur est utilisé pour réaliser l'asservissement des positions des coordonnées articulaires.

La plupart des applications d'asservissement par vision réalisées jusqu'à ce jour sont en fait des asservissements visuels indirects. Outre le fait qu'il est parfois matériellement difficile de modifier le contrôleur fourni par le constructeur, son intégration dans la boucle de vision permet de simplifier grandement la conception du système. En effet, ce dernier est souvent pourvu d'une interface de dialogue avec un système extérieur permettant d'avoir accès à toutes sortes de fonctions : contrôle de vitesse Cartésienne dans différents repères, gestions des singularités, gestion des sécurités... Néanmoins, une telle architecture n'est absolument pas adaptée à un asservissement par vision visant à atteindre de bonnes performances dynamiques. En effet, lorsque la cadence d'échantillonnage de l'information visuelle est suffisamment grande par rapport aux constantes de temps mécaniques du robot, des boucles internes de position ne peuvent que ralentir l'asservissement par vision par l'introduction de retards supplémentaires. Son rôle de stabilisation dans la boucle, valable lorsque la période de traitement d'image est grande, n'est plus du tout justifiée dans le cas d'un asservissement par vision rapide. Aussi, ce type d'asservissement correspond à des applications dont le but recherché est autre que la rapidité.

1.1.8 Les asservissements visuels directs

Nous ne sommes actuellement qu'au début de l'émergence des asservissements visuels directs. En 1984, Weiss suggérait, au vu de résultats de simulations, que la

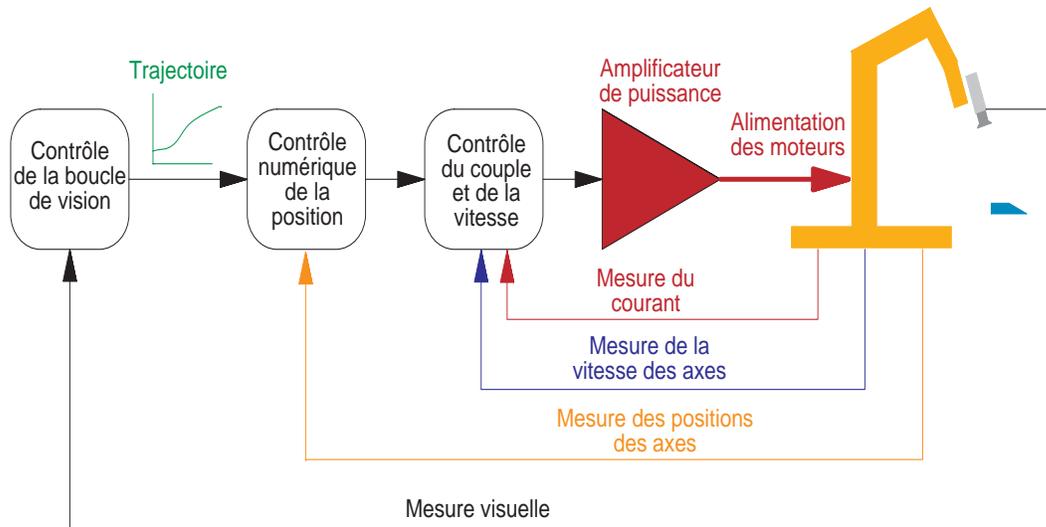


Fig. 1.6: Structure d'un asservissement visuel indirect

période d'échantillonnage de la vision devrait être inférieure à $33ms$ afin de garantir la convergence du contrôleur adaptatif d'un asservissement visuel à 2 degrés de liberté.

Corke fut l'un des pionniers en matière d'asservissement visuel direct [CG96, Cor96]. Il proposa différentes stratégies de contrôle pour des systèmes simples à 2 ou 3 degrés de liberté. Il étudia par exemple l'influence d'une commande en couple ou d'une commande en vitesse sur la boucle de vision. En raison de l'utilisation d'une caméra CCIR standard, la fréquence des asservissements visuels qu'il réalisa était limitée à $50Hz$. Néanmoins, une telle cadence est tout à fait compatible avec une structure d'asservissement visuel direct.

Les asservissements visuels directs devraient se généraliser dans un proche avenir. En effet, la croissance exponentielle de la puissance de calcul conjuguée avec l'apparition de systèmes de vision de plus en plus rapides pousse à adopter un telle structure de commande afin de mieux tirer parti des possibilités dynamiques offertes par le robot. La manipulation décrite dans [NI98] donne un exemple des dernières nouveautés en matière de système de vision. Les auteurs utilisent un nouveau capteur d'image qui intègre dans le même composant un dispositif d'acquisition et de traitement rapide d'image. Ce capteur est ainsi capable de réaliser des traitements comme le calcul du flot optique sur une matrice de 32×32 pixels à la cadence de $1kHz$. Néanmoins, la résolution de ce capteur n'est pas encore compatible avec son utilisation dans l'asservissement d'un robot suivant 6 degrés de liberté.

1.2 Les asservissements visuels rapides

Un asservissement visuel rapide, comme son nom l'indique, met en oeuvre un système de vision rapide dont la cadence d'échantillonnage et de traitement de l'image

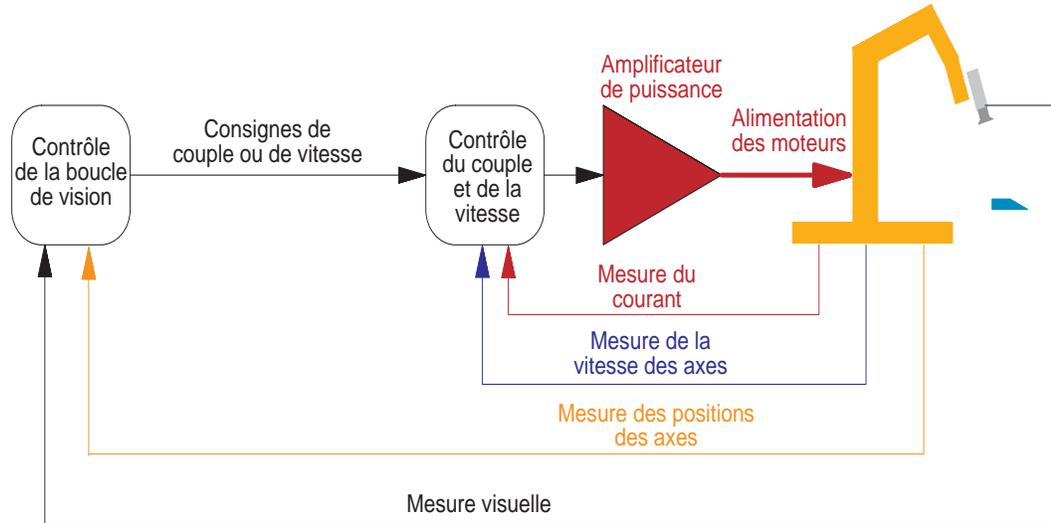


Fig. 1.7: Structure d'un asservissement visuel direct

permet d'exploiter au maximum les performances dynamiques du manipulateur. Nous considérerons comme rapide un asservissement visuel à une cadence $\geq 50\text{Hz}$.

Pour de tels asservissements, la dynamique du manipulateur n'est pas négligeable par rapport à la bande passante de la mesure visuelle. Le contrôle de la boucle de vision doit donc tenir compte d'un modèle dynamique du système.

La modélisation des dynamiques du robot dans une boucle de vision permet de régler le correcteur de l'asservissement de manière à obtenir une réponse optimale.

De plus, dans un contexte de suivi de cible, le modèle dynamique de la cible permet d'établir une prédiction de sa position. Cette prédiction peut être exploitée pour améliorer la performance du traitement d'image et pour augmenter la rapidité du suivi en utilisant un contrôleur prédictif, par exemple.

1.2.1 Prédiction du mouvement de la cible

Paradoxalement, la nécessité de réaliser une prédiction du mouvement d'un objet qu'on désire suivre est née du manque de puissance pour le traitement d'image. En effet, dans une tâche de suivi, l'image est souvent découpée en zones d'intérêt. Il s'agit de petites imagerie rectangulaires dans l'image contenant les primitives de l'objet qui serviront au calcul de la loi de commande (que ce soit un asservissement 2D ou 3D). Ainsi, le traitement d'image est limité à ces imagerie qui, réunies, ne constituent qu'une fraction très faible de l'image entière. L'avantage est double :

1. Le transfert des données brutes de l'image est limité aux pixels contenus dans les imagerie.
2. Le traitement d'image, très gourmand en puissance de calcul, est limité aux imagerie.

La contrepartie à ce fractionnement de l'image est la nécessité de constamment réactualiser la position des imagerie en fonction du mouvement de l'objet, afin qu'elles soient toujours centrées par rapport aux primitives qu'elles englobent. La technique la plus simple consiste à réactualiser cette position *a posteriori*, en considérant que la vitesse de déplacement de la primitive dans l'image est suffisamment faible pour qu'elle ne sorte pas de l'imagerie à l'acquisition suivante. Néanmoins, si la vitesse de la cible est trop importante ou si la fréquence d'acquisition de l'image est trop faible, une telle approche n'est plus viable. La solution consiste alors à réaliser une prédiction de la position future de la primitive dans l'image suivante et de placer l'imagerie suivante autour de cette position.

La prédiction de la position future de la cible est souvent réalisée grâce à un filtre de Kalman. Ce dernier permet d'obtenir la meilleure prédiction au sens de la minimisation de la variance de l'erreur. Néanmoins, il requiert un observateur du mouvement de la cible. Pour réaliser cet observateur, la plupart des auteurs font l'approximation d'un mouvement uniforme de la cible [WHB96, Cor96, CS93]. Il s'avère que la réalité du mouvement de la cible est souvent très différente d'un mouvement uniforme. Ainsi, un changement brusque dans le mouvement de la cible provoque un régime transitoire du filtre qui conduit à une mauvaise estimation transitoire de la position de la cible. C'est pour éviter ce phénomène que Bensalah propose dans [Ben96] un filtre de Kalman modifié capable de s'adapter très rapidement aux «ruptures de modèle».

La prédiction du mouvement futur de la cible peut également être exploitée dans un contexte d'asservissement prédictif. Dans ce cas, les signaux de référence futurs sont transmis à un correcteur prédictif qui réalise le contrôle de la boucle en fonction de l'évolution future de la trajectoire de la cible (voir [Cor96], par exemple).

Dans le cas particulier d'un asservissement 2D, il a été montré [Ben96] que l'estimation du mouvement de la cible et sa prise en compte dans la loi de commande permet d'annuler les erreurs de traînage.

1.2.2 Prise en compte des dynamiques du manipulateur

La prise en compte des dynamiques du manipulateur dans l'élaboration du contrôleur de la boucle de vision est un problème récent. Même si certains auteurs mentionnaient déjà ce point en 1979 [HP79], à l'époque, la lenteur de la cadence d'échantillonnage ne permettait pas de mettre en pratique un asservissement visuel direct.

En effet, une telle structure ne peut s'envisager que dans le cas d'un asservissement visuel rapide. Lorsque la cadence d'échantillonnage de la boucle de vision est trop faible, l'asservissement imbriqué des positions articulaires est requis pour stabiliser le système. La structure de contrôle direct va donc de pair avec un asservissement visuel rapide (voir section 1.1.8).

Weiss, constata en 1984 au vu de résultats de simulations [Wei84] que la fréquence d'échantillonnage de la vision d'un système à 3 degrés de liberté devrait être d'au moins 300Hz pour obtenir une boucle stable et précise. Le contrôleur de la boucle de

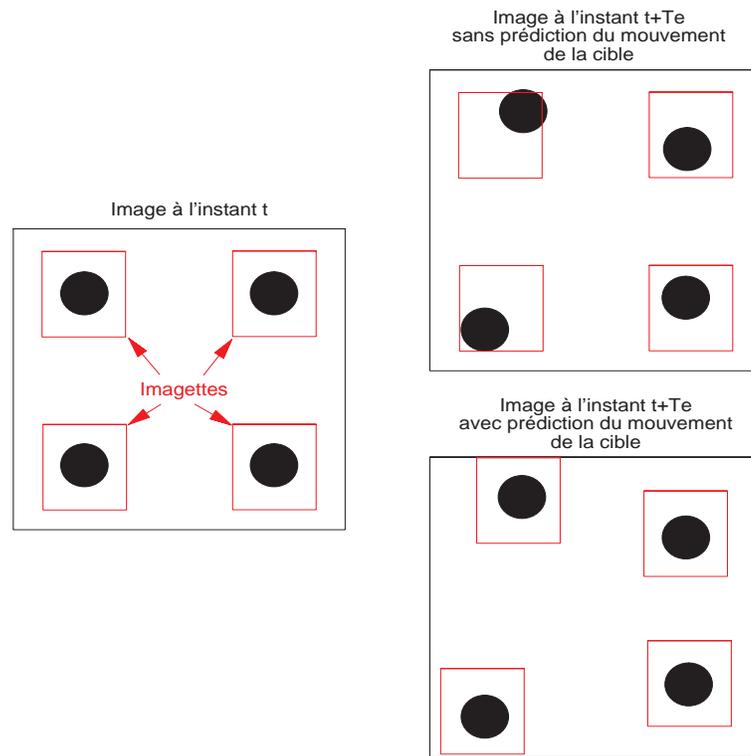


Fig. 1.8: Influence de la prédiction du mouvement de la cible sur la position des imagettes

vision était du type adaptatif et permettait ainsi de suivre les variations de modèle en fonction de la configuration du robot. L'identification des paramètres du modèle dynamique du système requérait une cadence d'échantillonnage très élevée afin d'être fiable, d'où les 300Hz.

Corke [CG96, Cor96] fut le premier à étudier de manière détaillée l'influence des dynamiques du manipulateur dans le contrôle de la boucle de vision. Les paramètres du contrôleur de la boucle de vision sont réglés grâce à une technique de placement de pôles. Le modèle dynamique du manipulateur est déterminé hors ligne (contrairement à ce qui est fait dans [Wei84]). Aussi, une cadence d'échantillonnage de 50Hz suffit à réaliser l'asservissement. Néanmoins, cette étude est restreinte à des systèmes relativement simples : au maximum 3 degrés de liberté sont asservis.

1.3 Contributions de cette thèse

Notre travail s'inscrit dans un contexte d'accroissement exponentiel de la puissance des microprocesseurs. Il a débuté à un moment où la réalisation d'un asservissement par vision d'un robot manipulateur suivant 6 degrés de liberté devenait possible à une fréquence d'échantillonnage suffisamment grande pour pouvoir entièrement tirer partie des possibilités dynamiques d'un robot. Néanmoins, afin d'exploiter cette amélioration des performances de la vision, une modélisation plus approfondie de la boucle d'asservissement tenant compte des dynamiques du manipulateur fut nécessaire. Une de nos contributions à cette discipline a consisté à proposer un modèle dynamique d'un robot manipulateur à 6 degrés de liberté utilisé dans une boucle de vision.

Dans notre approche générale de modélisation dynamique de la boucle de vision, le robot manipulateur est inséré dans une boucle d'asservissement visuel direct avec commande en vitesse. Les boucles de position au niveau des axes du robot sont désactivées et la vision se substitue aux capteurs articulaires pour fournir le retour de mesure de position. Tout l'asservissement est synchrone avec la cadence d'échantillonnage de l'image.

Le modèle dynamique du robot en boucle ouverte est très non-linéaire. Cependant, les asservissements de vitesse articulaire du robot réalisent un découplage et une linéarisation du comportement dynamique. Ainsi, le modèle dynamique que nous proposons est une linéarisation autour d'un point de fonctionnement des dynamiques du robot asservi en vitesse.

Nous avons validé ce modèle au travers de 2 expériences : un suivi de cible et un suivi de profilé. Les deux expériences sont cadencées à une fréquence d'échantillonnage suffisamment grande pour nécessiter la modélisation dynamique de la boucle de vision que nous avons proposée.

Nous avons pu utiliser une technique de synthèse optimale de correcteurs prédictifs qui tient compte de ce modèle dynamique. Elle a permis un accroissement important de la bande passante de l'asservissement tout en maintenant la stabilité.

Nous avons comparé les performances des asservissements contrôlés avec un correcteur classique à action Proportionnelle-Intégrale (PI) à celles obtenues avec un

correcteur GPC (*Generalized Predictive Controller*). Le correcteur PI est réglé expérimentalement tandis que le GPC est optimisé en fonction du modèle dynamique de la boucle. Les résultats expérimentaux ont permis de montrer la supériorité du correcteur GPC sur le correcteur PI validant ainsi la modélisation dynamique que nous proposons.

1.3.1 Le suivi de cible

La tâche de suivi de cible consiste à asservir à une référence la position d'une cible par rapport à une caméra suivant 6 degrés de liberté [GMA98]. Tout mouvement de la cible agit comme une perturbation pour l'asservissement et entraîne donc un mouvement de la caméra et par conséquent du robot. De plus, en modifiant le signal de référence, il est possible de générer une trajectoire par rapport à la cible.

Le suivi de cible a été réalisé avec une caméra rapide de résolution 640×240 , qui permet d'acquérir 120 images non entrelacées par seconde. Grâce à l'emploi d'une cible contenant des primitives très simples (des disques coplanaires) il a été possible de réaliser un asservissement visuel à la fréquence d'échantillonnage de 120 Hz. Nous n'utilisons pas de prédiction du mouvement de la cible. Nous avons préféré concentrer nos efforts sur la modélisation dynamique du robot.

Dans le cas d'un suivi de cible, le modèle dynamique de la boucle de vision que nous proposons s'applique de la même manière aux asservissements 2D et aux asservissements 3D. Nous avons ainsi été amenés à expérimenter les 2 configurations.

Dans le cas 2D, contrairement à la plupart des approches qui réalisent un contrôle dont les états sont exprimés dans le plan image sous forme de primitives (*cf.* par exemple [HEK96]), nous avons réalisé un contrôle dont les états sont exprimés sous la forme d'un vecteur contenant 6 coordonnées d'attitude. Nous avons ainsi introduit une nouvelle stratégie de contrôle en 2D applicable au suivi de cible.

1.3.2 Le suivi de profilé

Dans la seconde expérience, nous avons réalisé un suivi de profilé tridimensionnel. Le but est d'asservir la position de la caméra suivant 6 degrés de liberté par rapport à une section d'un profilé et de réaliser un déplacement de la caméra le long de ce profilé.

L'asservissement a été réalisé avec une caméra CCIR standard. Le traitement alterné des trames paires et impaires a permis de cadencer l'asservissement à 50 Hz.

Peu de travaux ont déjà été entrepris dans ce domaine [LWH98]. Un exemple d'asservissement 2D pour un suivi de profilé est décrit dans [ECR92]. Mais il se limite au suivi d'un profilé droit.

Dans ce travail, la manipulation de suivi de profilé a été abordée dans le cas 3D. Il est nécessaire de connaître la section du profilé ; cependant, il n'est pas nécessaire de connaître la courbure du profilé.

Nous proposons deux stratégies originales de commande pour la tâche de suivi. Ces stratégies se distinguent par le fait que l'une d'entre elles tient compte, dans

le suivi, de l'évolution future de la courbure du profilé par le biais de la fonction prédictive du GPC.

1.4 Plan du mémoire

La suite de ce travail est divisée en quatre parties. Le chapitre 2 est consacré à la modélisation du robot et de la caméra. Après une brève introduction des notions mathématiques de base requises en asservissement par vision (section 2.1), nous présentons le modèle de la caméra tant du point de vue géométrique que dynamique (section 2.2). Puis, nous exposons dans la section 2.3 la modélisation cinématique et dynamique d'un robot. Cette modélisation générale est appliquée au robot que nous avons utilisé pour nos expériences dans la section 2.4.

Dans le chapitre 3, nous décrivons la modélisation de l'asservissement visuel. La section 3.1 est consacrée à sa modélisation dynamique. Pour cela, nous introduisons la notion de *dispositif de déplacement opérationnel virtuel* (section 3.1.2). La section 3.2 est consacrée à une description détaillée du correcteur GPC. Nous avons choisi ce correcteur pour réaliser nos asservissements visuels à cause de ses caractéristiques adaptatives et prédictives.

La suite de ce chapitre est consacrée à la description du système expérimental. La section 3.3 présente la plateforme expérimentale dans son ensemble. Finalement, la section 3.4 présente la simulation du système de vision que nous avons développée.

Dans le chapitre 4, nous présentons l'application de suivi de cible. Une structure de contrôle générale, commune aux asservissements 2D et 3D est présentée dans la section 4.1. Dans cette section, nous comparons également l'approche 2D classique à notre approche. La section 4.2 présente les résultats expérimentaux qui permettent de démontrer la validité de notre modèle dynamique.

Le chapitre 5 est consacré à l'application de suivi de profilé. Dans la section 5.1, après avoir défini un profilé, nous décrivons une méthode originale d'estimation de l'attitude d'une section du profilé par rapport à la caméra. La détermination de la section courante le long du profilé en fonction du modèle géométrique du robot est exposée dans la section 5.2. Les 2 stratégies de commande de l'asservissement sont décrites dans les sections 5.3 et 5.4. Finalement, la section 5.5 présente les résultats expérimentaux. Une comparaison des 2 stratégies de commande y est présentée.

Chapitre 2

Modélisation de la caméra et du robot

2.1 Notions de base

Quelques notions fondamentales propres aux asservissements par vision sont décrites dans cette section. Elles sont communes à tous les asservissements de l'attitude de l'organe terminal du robot manipulateur à partir de mesures fournies par une caméra.

2.1.1 Définition de l'attitude d'un repère

En robotique, il est souvent nécessaire de définir la position relative d'un corps par rapport à un autre : la position des différents axes du robot par rapport à sa base, la position de l'organe terminal par rapport à une pièce, *etc.* Lorsqu'il s'agit d'une position relative suivant 6 degrés de liberté, on parle alors *d'attitude*. Une attitude est définie mathématiquement par une translation et une rotation qui permettent de passer d'un repère à un autre. On définit l'attitude (*pose* en anglais) p_{12} du repère direct R_2 par rapport au repère direct R_1 par une translation T_{12} et une rotation R_{12} . Si O_1 est l'origine du repère R_1 et O_2 est l'origine du repère R_2 , alors le vecteur de translation T_{12} est défini par $\overrightarrow{O_1O_2}$. Soient \vec{x}_1, \vec{y}_1 et \vec{z}_1 les vecteurs unitaires du repère R_1 et \vec{x}_2, \vec{y}_2 et \vec{z}_2 les vecteurs unitaires du repère R_2 . La rotation R_{12} est celle qui transforme \vec{x}_1 en \vec{x}_2 , \vec{y}_1 en \vec{y}_2 et \vec{z}_1 en \vec{z}_2 . La rotation R_{12} peut être définie par une matrice :

$$R_{12} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (2.1)$$

où $[r_{11} \ r_{21} \ r_{31}]^T$ sont les coordonnées de \vec{x}_2 exprimées dans le repère R_1 , $[r_{12} \ r_{22} \ r_{32}]^T$ sont les coordonnées de \vec{y}_2 exprimées dans le repère R_1 et $[r_{13} \ r_{23} \ r_{33}]^T$ sont les coordonnées de \vec{z}_2 exprimées dans le repère R_1 . Si $P_{R_2} \in \mathbb{R}^3$ est un vecteur contenant

les coordonnées d'un point P exprimées dans le repère R_2 , alors, les coordonnées de ce point exprimées dans le repère R_1 sont données par :

$$P_{R_1} = R_{12}P_{R_2} + T_{12} \quad (2.2)$$

L'attitude peut elle aussi être définie sous forme matricielle. On utilise alors le formalisme des matrices homogènes. Celles-ci présentent l'intérêt de pouvoir composer plusieurs attitudes par simple multiplication d'une série de matrices. Ainsi, la matrice homogène de transformation $M_{12} \in \mathbb{R}^{4 \times 4}$ entre le repère R_1 et le repère R_2 est définie comme suit :

$$M_{12} = \begin{pmatrix} R_{12} & T_{12} \\ 0 & 1 \end{pmatrix} \quad (2.3)$$

Ce formalisme impose l'utilisation de quaternions pour définir les coordonnées d'un point. Ainsi l'équation (2.2) est équivalente à :

$$\begin{pmatrix} P_{R_1} \\ 1 \end{pmatrix} = M_{12} \begin{pmatrix} P_{R_2} \\ 1 \end{pmatrix} \quad (2.4)$$

Soient P_{R_3} , les coordonnées de P dans un repère R_3 et M_{23} , la matrice homogène de transformation entre R_2 et R_3 . Les coordonnées de P dans R_1 sont données par :

$$P_{R_1} = M_{12} \cdot M_{23} \cdot P_3 \quad (2.5)$$

L'attitude peut également être définie de manière unique par un vecteur de 6 coordonnées. On parle alors de coordonnées opérationnelles [Coi92]. Parmi ces 6 coordonnées, 3 définissent la translation et 3 autres définissent la rotation. S'il n'y a pas d'ambiguïté pour la définition de la translation, il n'en est pas de même pour la rotation. Il existe en effet plusieurs conventions permettant de décomposer cette rotation en 3 rotations élémentaires [Coi92]. Celle qui est la plus largement utilisée consiste à décomposer la rotation en 3 rotations θ_t , θ_r et θ_l respectivement autour des axes x , y , et z . On appelle aussi ces angles tangage, roulis et lacet. Dans l'annexe B.3 nous revenons plus largement sur la définition de ces angles.

2.1.2 Définition des repères

On appelle R_c le repère lié à la caméra. Par convention, R_c est défini comme suit¹ (voir figure 2.1) :

- **Son origine est au centre optique de la caméra.**
- **L'axe z_c est confondu avec l'axe optique et est dirigé vers la scène.**
- L'axe x_c est vertical dans l'image et dirigé vers le haut.
- L'axe y_c est horizontal et dirigé tel que R_c soit direct.

¹ Les conventions communément admises dans la littérature sont notées en gras, les autres sont celles que nous avons adoptées.

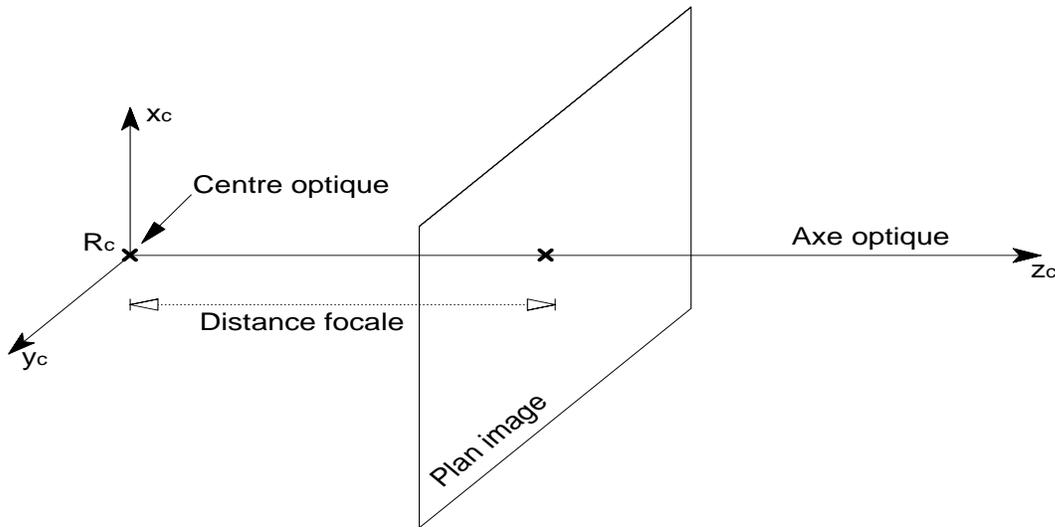


Fig. 2.1: Définition du repère caméra

Indice	Repère associé
c	Repère caméra
o	Repère lié à un objet
b	Repère lié à la base du robot
t	Repère lié à l'organe terminal du robot

Tab. 2.1: Convention pour les indices des repères

On désigne par R_o un repère lié à un objet situé dans l'espace de travail du robot. La matrice homogène M_{co} définit l'attitude de R_o par rapport à R_c .

Dans un asservissement par vision d'un robot, la grandeur à asservir est souvent l'attitude de R_o (ou d'un autre repère lié à l'objet) par rapport à un repère R_t lié à l'organe terminal. Lorsque la caméra est extérieure (*cf.* figure 2.2), l'estimation des transformations M_{ct} et M_{co} permet de déterminer la position de l'objet par rapport à l'organe terminal. Lorsque la caméra est attachée au robot, elle est généralement solidaire de l'organe terminal. Dans ce cas, il est judicieux de choisir $R_c = R_t$ pour simplifier les calculs (*cf.* figure 2.3). La position de l'objet par rapport à l'organe terminal est donc directement obtenue par l'estimation de M_{co} .

On a souvent besoin de définir la position de l'organe terminal par rapport à la base du robot. Pour cela, on utilise R_b , un repère fixe qui est solidaire de la base du robot.

Le tableau 2.1 récapitule la signification des différents indices utilisés pour les repères.

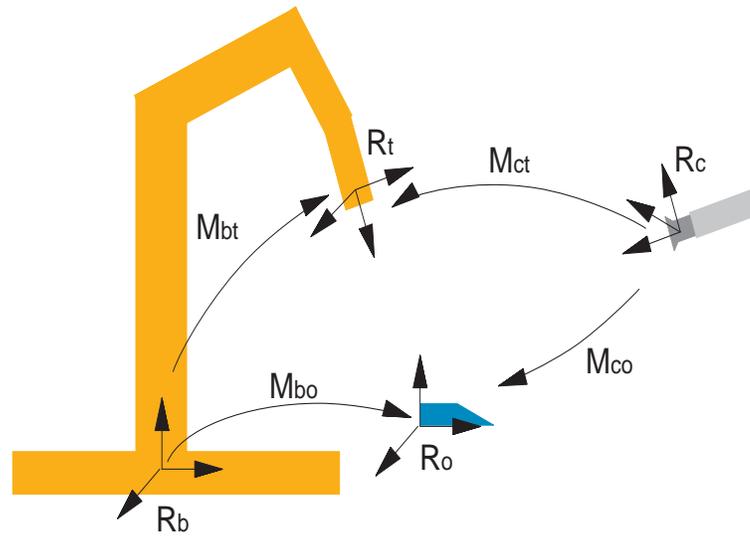


Fig. 2.2: Description de la configuration où la caméra est extérieure

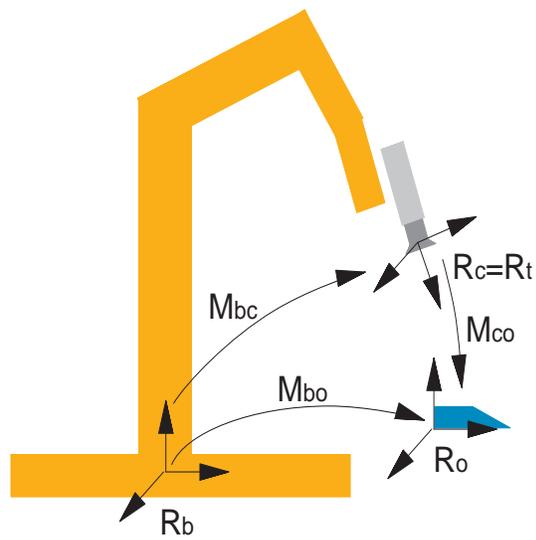


Fig. 2.3: Description de la configuration *eye in hand*

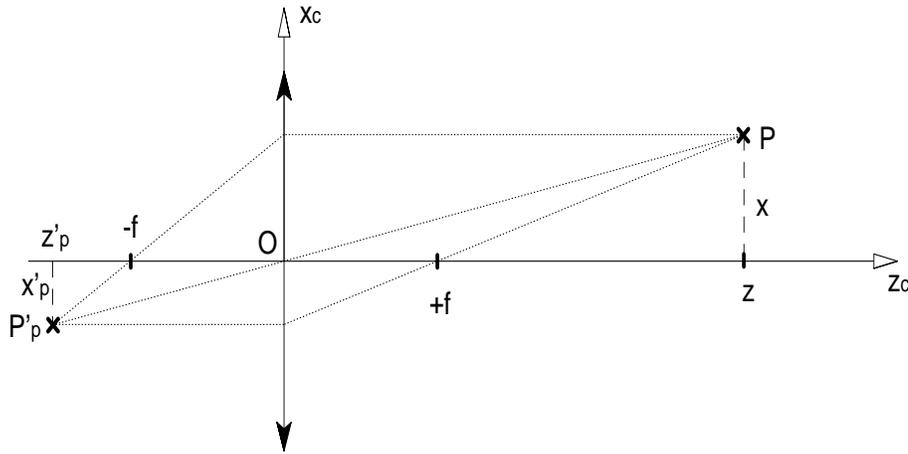


Fig. 2.4: Modèle d'une lentille mince

2.2 Modélisation de la caméra

Cette section présente la modélisation d'une caméra. Dans ce cadre, nous introduisons un modèle de la distorsion de l'image et nous décrivons le moyen de la compenser. Nous abordons ensuite la modélisation dynamique d'une caméra. Pour cela, certains aspects de la technologie de ces capteurs sont abordés.

2.2.1 Modèle de projection d'un point dans l'image

Nous supposons que l'objectif de la caméra est constitué d'une ou plusieurs lentilles qui peuvent se modéliser par une seule lentille mince convergente comme le décrit la figure 2.4. On définit un repère $R_c(x_c, y_c, z_c)$ associé à la lentille mince. Soit P un point lumineux dont l'image à travers la lentille est P'_p . La coordonnée de P suivant l'axe z_c est z et la coordonnée de P'_p suivant ce même axe est z'_p . L'image de P est réelle si $z > f$ où f est la longueur focale de la lentille. En appliquant la relation fondamentale d'une lentille mince on obtient :

$$\frac{1}{z} - \frac{1}{z'_p} = \frac{1}{f} \quad (2.6)$$

Soient x et x'_p les coordonnées respectives de P et P'_p suivant l'axe x_c . Grâce au théorème de Thalès nous obtenons (voir figure 2.4) :

$$\frac{x}{x'_p} = \frac{z}{z'_p} \quad (2.7)$$

Or, d'après (2.6), on a :

$$\frac{z}{z'_p} = 1 - \frac{z}{f} \quad (2.8)$$

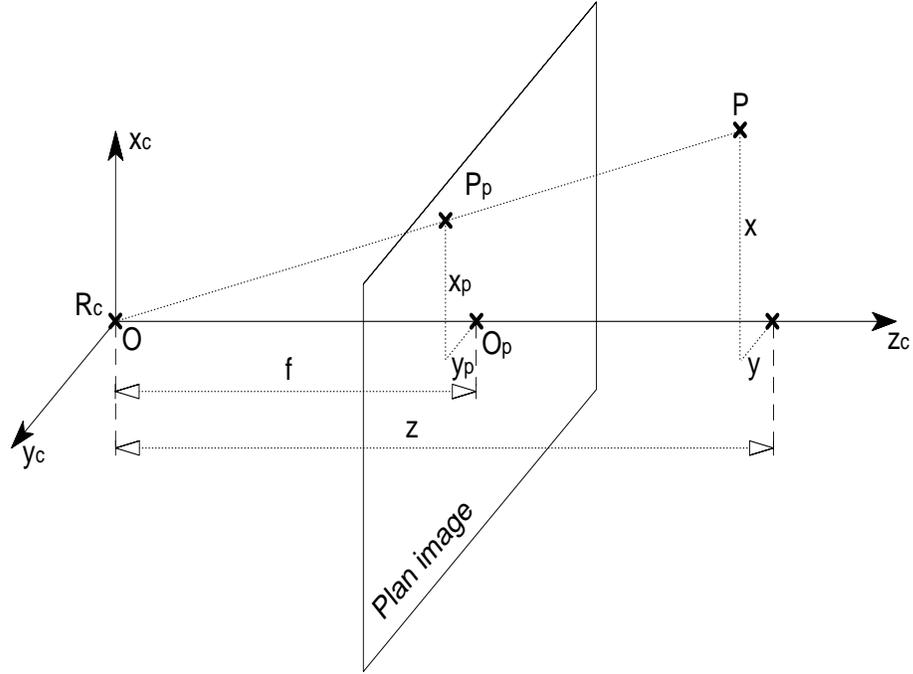


Fig. 2.5: Description de la projection perspective

D'où :

$$x'_p = \frac{fx}{f-z} \quad (2.9)$$

Comme dans la plupart des cas $z \gg f$, on réalise l'approximation suivante :

$$x'_p \approx -\frac{fx}{z} \quad (2.10)$$

Cette relation définit une projection perspective de centre O (le centre de la lentille) sur un plan perpendiculaire à l'axe optique z_c situé à l'abscisse $-f$. Cette projection réalise une inversion de l'image. Par commodité, on transforme souvent l'image par une symétrie centrale de centre O de manière à la redresser. Dans ce cas, la composition de la projection et de la symétrie centrale conduit à :

$$x_p = -x'_p \approx \frac{fx}{z} \quad (2.11)$$

La composition de ces 2 transformations équivaut à projeter P par une projection de centre O sur un plan image virtuel situé à l'abscisse f comme l'illustre la figure 2.5. En raisonnant de la même manière suivant y_c , on obtient :

$$y_p = -y'_p \approx \frac{fy}{z} \quad (2.12)$$

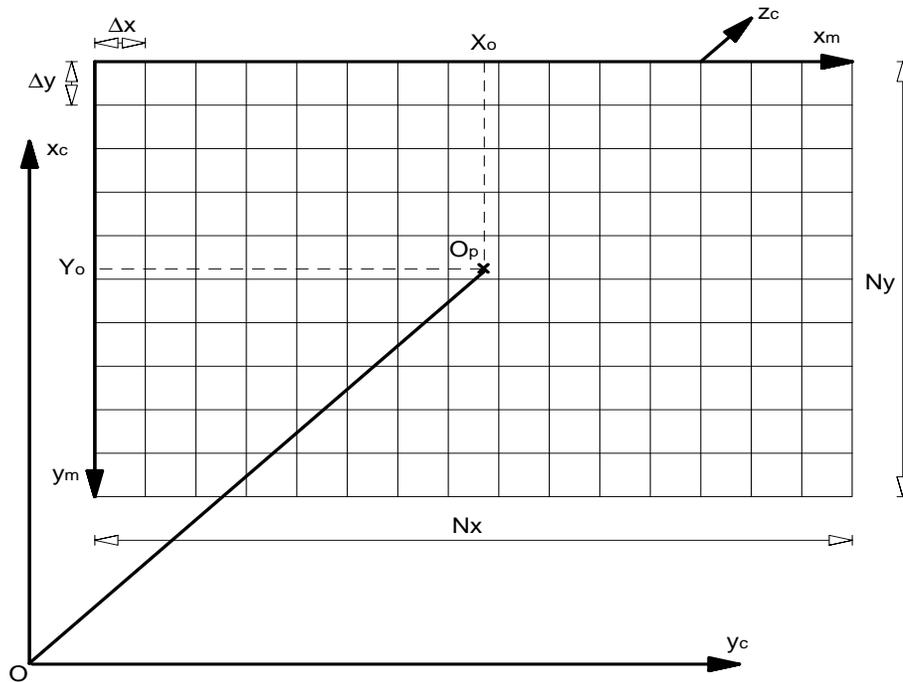


Fig. 2.6: Modélisation d'une matrice CCD

2.2.2 Le capteur visuel

La plupart des caméras modernes utilisent une matrice de capteurs rectangulaires élémentaires appelés *pixels* réalisés dans un matériau semi-conducteur. Le capteur le plus répandu est dénommé CCD (*Charge Coupled Device*). Nous ne rentrerons pas dans les détails technologiques. Pour plus d'informations sur le sujet, le lecteur peut se référer au livre de Corke [Cor96].

Chaque pixel se comporte comme un intégrateur de la quantité de photons reçue. Les photons sont transformés en charges électriques et donc à l'issue du temps d'intégration, la charge totale contenue dans un pixel est sensiblement proportionnelle à la quantité de photons reçue. Ce temps d'intégration peut être réglé sur la plupart des caméras CCD. On parle alors d'obturateur électronique. La matrice CCD est placée au niveau du plan image réel de la caméra (situé à l'abscisse $-f$). Néanmoins, nous pouvons la modéliser par commodité comme étant placée à l'abscisse f . Par la connaissance de la luminance associée à chaque pixel, on obtient donc un échantillonnage spatial de l'image.

Soient N_x et N_y respectivement le nombre de colonnes et de lignes de la matrice CCD (cf. figure 2.6); $(N_x \times N_y)$ est appelé résolution du capteur CCD. Soient Δ_x et Δ_y les dimensions d'un pixel. On définit G_x et G_y , respectivement les grandissements

suivant les axes x_c et y_c du repère R_c par :

$$G_x = \frac{f}{\Delta_y} \quad (2.13)$$

$$G_y = \frac{f}{\Delta_x} \quad (2.14)$$

Ainsi, les relations de projection perspective (2.11) et (2.12) peuvent être réécrites de la manière suivante :

$$\begin{aligned} X_p &\approx G_x \frac{x}{z} \\ Y_p &\approx G_y \frac{y}{z} \end{aligned} \quad (2.15)$$

où X_p et Y_p sont les coordonnées de P_p dans R_c exprimées en pixels et non en mètres.

Remarques

- La matrice CCD réalise un échantillonnage spatial de l'image. Donc, en toute rigueur, X_p et Y_p devraient être des valeurs discrètes. Néanmoins, l'information visuelle est rarement utilisée sous sa forme brute. Grâce à des traitements de l'image, on arrive à en extraire une information *sub-pixel*, donc en fraction de pixels. Les équations (2.15) se justifient donc dans ce contexte.
- Dans les relations (2.15), nous considérons que le repère associé aux coordonnées X_p et Y_p est R_c (voir figure 2.5). Dans la pratique, l'axe optique z_c coupe le capteur CCD approximativement en son centre. Néanmoins, en fonction des réglages de la lentille, sa position est susceptible de fluctuer légèrement (tout comme les grandissements G_x et G_y). Nous appelons X_0 et Y_0 les coordonnées de ce point d'intersection (O_p) dans un repère $R_m(x_m, y_m)$ solidaire du capteur (voir figure 2.6). Les grandeurs G_x , G_y , X_0 et Y_0 sont appelées *paramètres intrinsèques de la caméra*.

2.2.3 Compensation de la distorsion

La distorsion la plus significative qui affecte les images produites par un objectif constitué de lentilles est la distorsion radiale. Tous les points de l'image sont translatés suivant des directions définies par des droites passant par ces points et par l'intersection O_p de l'axe optique avec le plan image. Cette translation est d'autant plus importante que le point est éloigné de O_p . Les objectifs à focale courte ont tendance à déformer l'image plus que les objectifs à grande focale.

Tsai [Tsa87] propose un modèle de distorsion simplifié qui approxime l'amplitude de la translation des pixels par une fonction du carré de la distance à O_p . Soient $[X_p \ Y_p]^T$ les coordonnées en pixels d'un point P_p dans l'image obtenu par projection d'un point P (voir équation (2.15)) et $[X_d \ Y_d]^T$ les coordonnées du point P_p ayant subit la distorsion. D'après Tsai :

$$\begin{bmatrix} X_d \\ Y_d \end{bmatrix} = \frac{1}{D} \begin{bmatrix} X_p \\ Y_p \end{bmatrix} \quad (2.16)$$

avec

$$D = 1 + K_d \left(X_p^2 + \left(\frac{G_x}{G_y} \right)^2 Y_p^2 \right) \quad (2.17)$$

où $K_d > 0$ est le coefficient de distorsion. Les coordonnées $[X_d \ Y_d]^T$ sont celles mesurées dans l'image. Le coefficient de distorsion peut être estimé par des techniques de calibration. Grâce à la connaissance de K_d on peut compenser la distorsion de l'image. Soient $[X_c \ Y_c]^T$ les coordonnées corrigées de P_p :

$$\begin{bmatrix} X_c \\ Y_c \end{bmatrix} = D' \begin{bmatrix} X_p \\ Y_p \end{bmatrix} \quad (2.18)$$

avec

$$D' = 1 + K_d \left(X_d^2 + \left(\frac{G_x}{G_y} \right)^2 Y_d^2 \right) \quad (2.19)$$

Comme la distorsion est généralement relativement faible, on considère que $X_d \approx X_p$ et $Y_d \approx Y_p$ dans l'expression de D' . Par conséquent, $D' \approx D$ et finalement $[X_c \ Y_c]^T \approx [X_p \ Y_p]^T$.

2.2.4 Modélisation dynamique de la caméra

Avec une caméra CCD standard, le transfert des niveaux de luminance des pixels se fait séquentiellement et de manière analogique. Il en résulte un signal vidéo analogique continu dont la tension varie en fonction du temps suivant la valeur des niveaux de gris des pixels balayés. Le balayage de l'image s'effectue toujours de la gauche vers la droite et de haut en bas. Lorsque l'image est entrelacée, toutes les lignes paires, puis toutes les lignes impaires sont balayées alternativement. On définit une trame comme étant le signal résultant du balayage de toute l'image. Une image entrelacée est donc constituée d'une trame paire et d'une trame impaire. Une image non entrelacée ne comprend qu'une seule trame.

Le processus d'acquisition d'une image peut se décomposer en 3 étapes :

1. Le temps d'intégration T_i au cours duquel les éléments sensibles intègrent les charges produites par les photons qui les atteignent.
2. Le transfert des charges : les charges électriques contenues dans les éléments sensibles sont transférées dans un registre à décalage. A partir de cet instant, la mesure est terminée jusqu'à la prochaine trame.
3. Les registres à décalage sont vidés séquentiellement afin de générer le signal vidéo.

Comme le montre la figure 2.7, le signal vidéo de la trame $n - 1$ est émis pendant que la trame n est acquise. Il y a donc un retard de la durée T_e d'une trame entre le moment où l'image commence à être mesurée et le moment où elle commence à

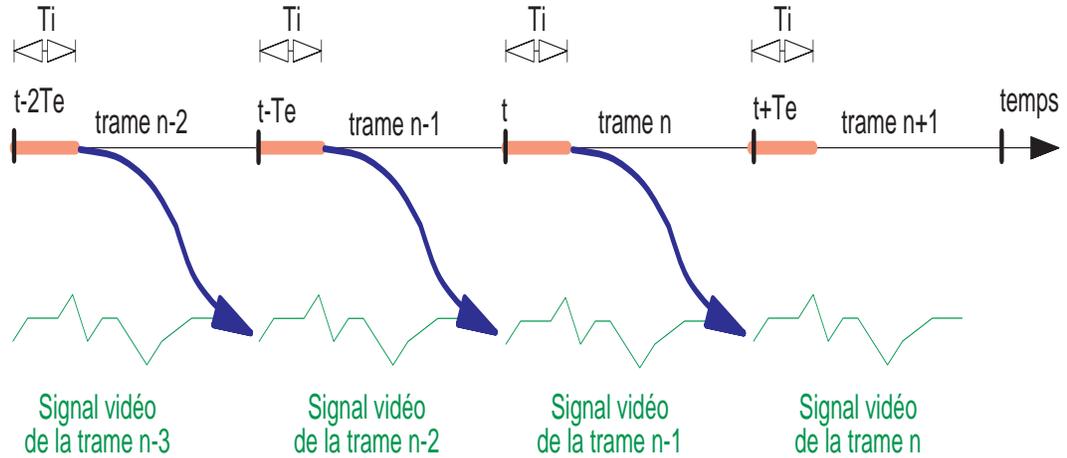


Fig. 2.7: Chronogramme du processus d'acquisition d'images d'une caméra CCD

être transférée. Comme l'émission du signal vidéo dure approximativement le temps d'une trame, il existe en fait un retard de $2T_e$ entre le moment où l'image commence à être mesurée et le moment où son transfert et sa digitalisation au niveau du système de vision sont achevés. La durée T_i correspond à la période de l'obturateur électronique. Etant donné que l'image ne peut être mesurée instantanément, on peut considérer que «l'instant moyen» de mesure de la trame n est $t + T_i/2$. Ainsi, si T_i tend vers 0, le retard pur T_r induit par le capteur de vision est de $2T_e$. Par contre, si $T_i = T_e$, ce qui est la configuration classique, $T_r = 1.5T_e$.

Les systèmes de vision modernes (*i.e.*, le dispositif qui permet de numériser l'image) permettent de lire l'image en même temps qu'elle est acquise. De plus, il est possible de connaître exactement la ligne de l'image qui est couramment numérisée. Ainsi, dans un contexte d'asservissement par vision, il est possible de synchroniser une boucle de vision sur l'instant $t + T_e/2$ correspondant au milieu d'une trame (voir figure 2.8). Il est donc possible de réaliser un traitement de la moitié de l'image ayant déjà été acquise tout en continuant, en parallèle, d'acquérir l'autre moitié de l'image. Avec une telle technique, si $T_i = T_e$, on a $T_r = T_e$. On en déduit que la fonction de transfert en z d'un modèle simplifié du processus de vision est z^{-1} .

Il est clair que le processus de traitement d'image et de calcul des commandes (en rouge sur la figure 2.8) introduit un second retard pur. Ce dernier correspond plutôt à une caractéristique de la commande ; aussi, nous en avons tenu compte dans la fonction de transfert modélisant le robot.

2.3 Modélisation d'un robot

Dans cette section nous décrivons la modélisation cinématique et dynamique d'un robot. Cette modélisation est générale à tous les types de manipulateurs.

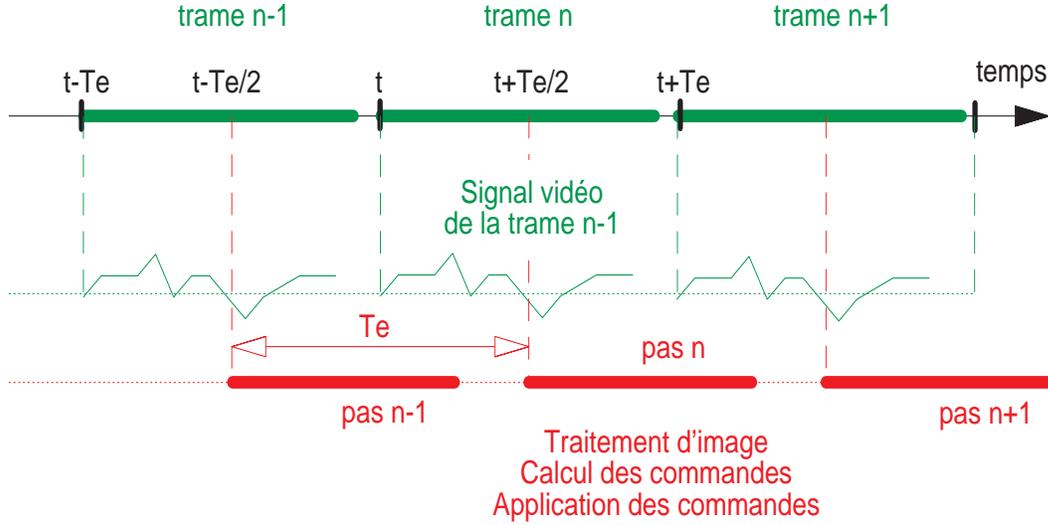


Fig. 2.8: Chronogramme de l'acquisition et du traitement d'images

2.3.1 Modélisation cinématique d'un robot

Un robot à structure cinématique ouverte est constitué de plusieurs corps reliés entre eux par des axes. Ces derniers réalisent généralement une liaison en rotation ou en translation. L'ensemble des paramètres définissant les angles de rotation ou les longueurs de translation des axes sont appelés *coordonnées articulaires* et seront notées q .

On modélise un robot en attachant un repère à chaque axe. On considère le robot comme un chaînage de $n + 1$ corps numérotés de 0 à n . Le corps 0 représente la base du robot tandis que l'organe terminal porte le numéro n . La position du corps k est repérée par rapport au corps $(k - 1)$ grâce à la matrice homogène $M_{(k-1)(k)}$. Cette matrice définit la transformation entre le repère R_{k-1} attaché au corps $(k - 1)$ et le repère R_k attaché au corps k exprimée dans le repère R_{k-1} . La matrice $M_{(k-1)(k)}$ dépend de q_k , la coordonnée articulaire de l'axe k , on la notera donc $M_{(k-1)(k)}(q_k)$.

Le modèle géométrique du robot est l'expression, en fonction des coordonnées articulaires, de l'attitude de l'organe terminal du robot par rapport à sa base. Il est défini par la matrice homogène $M_{0n}(q)$ où $q = [q_1 \dots q_n]^T$ est le vecteur des coordonnées articulaires. On a donc :

$$M_{0n}(q) = M_{01}(q_1)M_{12}(q_2)\dots M_{(n-1)n}(q_n) \quad (2.20)$$

2.3.2 Modélisation dynamique d'un robot

Le modèle dynamique d'un robot peut être calculé à partir de l'équation de Euler-Lagrange :

$$F = \frac{d}{dt} \frac{\partial \chi}{\partial \dot{q}} - \frac{\partial \chi}{\partial q} + \frac{\partial \varpi}{\partial q} \quad (2.21)$$

où :

- χ est l'énergie cinétique du robot.
- ϖ est l'énergie potentielle du robot.
- $q \in \mathbb{R}^n$ est le vecteur des n positions articulaires du robot.
- $F \in \mathbb{R}^n$ est le vecteur des n actions (couple ou force) externes appliquées aux n axes du robot.

Il est découle l'équation différentielle non-linéaire bien connue [SV89] :

$$\begin{aligned} D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f_f &= f_g \\ \text{avec } f_g - f_f &= F \end{aligned} \quad (2.22)$$

où :

- $D(q) \in \mathbb{R}^{n \times n}$ est la matrice d'inertie du robot manipulateur.
- $C(q, \dot{q})\dot{q} \in \mathbb{R}^n$ est le vecteur des effets centrifuges et de Coriolis.
- $g(q) \in \mathbb{R}^n$ est le vecteur des charges engendrées par la gravité.
- $f_f \in \mathbb{R}^n$ représente les actions dûes aux frottements.
- $f_g \in \mathbb{R}^n$ est un vecteur force généralisé représentant les actions des moteurs du robot.

Le vecteur force généralisé f_g contient les couples et les forces qu'appliquent les actionneurs du robot sur les axes. Ces grandeurs sont toujours asservies ; leur valeur peut être contrôlée avec précision et rapidité. La plupart des robots intègrent également un asservissement de vitesse sur chaque axe. C'est alors la variable \dot{q} qui est contrôlée. L'objectif de ces asservissements de vitesse est de linéariser le comportement dynamique du robot qui, comme l'indique l'équation (2.22), est par nature fortement non-linéaire.

Grâce aux boucles de vitesse, on peut faire l'approximation d'un découplage des dynamiques de chaque axe : le mouvement d'un axe i n'a pas d'influence en première approximation sur un axe j , si $i \neq j$. Soit $F_i(z)$ la fonction de transfert liant la consigne de vitesse \dot{q}_i^* de l'asservissement de vitesse du $i^{\text{ème}}$ axe à la vitesse articulaire \dot{q}_i de cet axe. On a :

$$F_i(z) = \frac{\dot{q}_i}{\dot{q}_i^*} \quad (2.23)$$

Ces fonctions de transfert $F_i(z)$ sont définies autour d'un point de fonctionnement q , étant donnée la non-linéarité du modèle. Elles dépendent donc de la configuration du robot définie par q . En pratique, c'est essentiellement la variation de la matrice d'inertie $D(q)$ qui est à l'origine de la variation des fonctions de transfert.

2.4 Modélisation du robot SCEMI 6P01

Nous exposons dans cette section la modélisation cinématique et dynamique du manipulateur que nous avons utilisé lors de nos expériences : le SCEMI 6P01. Pour cela, nous reprenons les principes exposés dans la section 2.3 que nous appliquons au cas particulier de ce robot.

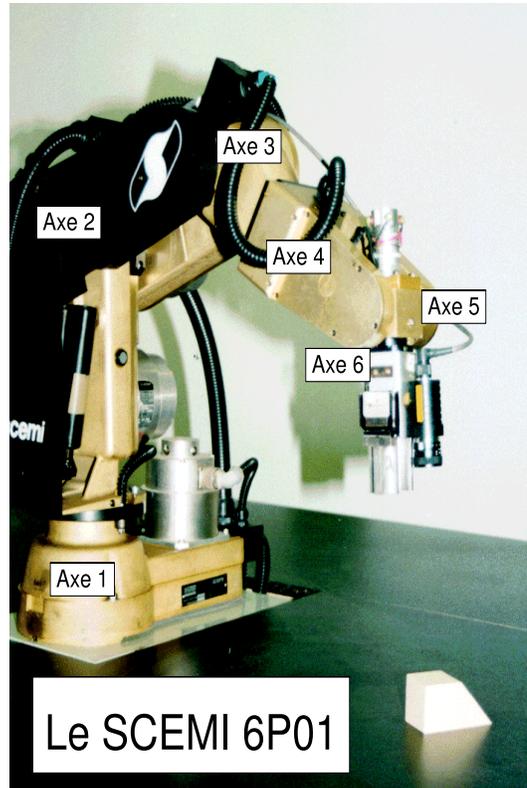


Fig. 2.9: Le robot SCEMI 6P01

2.4.1 Présentation générale

Le robot SCEMI 6P01 est un bras manipulateur industriel à 6 degrés de liberté. Son espace de travail est délimité par une demi-sphère d'environ 70cm de rayon. Il possède 6 axes en rotation. La figure 2.9 en donne un aperçu. Ses 3 derniers axes (4, 5 et 6) forment une rotule. Tous les axes sont mûs par des moteurs à courant continu : les 3 premiers axes sont équipés de moteurs AXEM à rotor plat de 400W chacun et les 3 derniers axes sont entraînés par des moteurs à rotor cylindrique standards de 50W chacun. Chaque moteur est pourvu d'un réducteur. Pour les 3 premiers axes, il s'agit d'un réducteur de type *Harmonic drive*, présentant comme principales caractéristiques d'avoir un jeu très faible mais une élasticité importante. Le SCEMI est muni d'un système de compensation des forces de gravité. Il s'agit de ressorts qui agissent sur les axes subissant la gravité (les axes 2 et 3) de manière à la compenser. En pratique, la compensation n'est cependant parfaitement réalisée qu'en certains points.

Du point de vue électrique, chaque moteur est asservi en courant grâce à une boucle analogique. Cet asservissement a pour fonction le contrôle du couple moteur (dans un moteur à courant continu, le couple est proportionnel au courant d'induit). Il joue également un rôle de protection : une saturation de la référence de cette boucle permet de limiter le courant à la valeur maximale tolérée par le moteur. Cette boucle

de courant est imbriquée dans une boucle analogique de vitesse. La mesure de vitesse est obtenue grâce à une génératrice tachymétrique fixée sur chaque arbre moteur.

De plus, chaque axe est muni d'un codeur incrémental couplé à l'arbre moteur. Celui-ci fournit une mesure de position relative sous forme de créneaux de tension. Classiquement, cette mesure permet la réalisation d'un asservissement numérique de position de chaque axe. Etant donné que cette mesure n'est pas absolue, une phase préalable de mise en référence est nécessaire avant l'utilisation du robot.

2.4.2 Modélisation géométrique

Un robot série (par opposition à un robot parallèle) est classiquement constitué d'un chaînage de corps solides inter-connectés entre eux par des axes. Ces derniers sont principalement de 2 types : rotation ou translation. Ainsi, pour un robot constitué de $n + 1$ corps solides chaînés numérotés de $0 \dots n$, il y a n axes numérotés de $1 \dots n$ qui les relient. Afin de modéliser la posture du robot, on fixe à chaque corps solide du robot un repère R_i , $i = 0 \dots n$. Il y a une infinité de façons de positionner chaque repère. La transformation entre un repère R_{i-1} lié à un corps $i - 1$ et un repère R_i lié au corps suivant est entièrement définie par 6 paramètres (3 translations et 3 rotations). Néanmoins, en imposant 2 contraintes sur la position relative de ces 2 repères, ce nombre est réduit à 4. Les 2 contraintes sont :

1. L'axe x_{i-1} de R_{i-1} est perpendiculaire à l'axe z_i de R_i .
2. L'axe x_{i-1} de R_{i-1} et l'axe z_i de R_i sont concourants.

De plus, afin de fixer la position des repères par rapport aux corps solides, il est convenu que z_i est support de l'axe i .² Ces contraintes constituent la convention de Denavit-Hartenberg (voir [SV89] et [Cra89]) classiquement utilisée en robotique. Ainsi, la transformation entre R_{i-1} et R_i peut être décomposée en 4 opérations élémentaires :

1. Rotation de z_{i-1} vers z_i d'un angle α_{i-1} autour de l'axe x_{i-1} .
2. Translation de z_{i-1} vers z_i d'une distance a_{i-1} le long de l'axe x_{i-1} .
3. Rotation de x_{i-1} vers x_i d'un angle θ_i autour de l'axe z_i .
4. Translation de x_{i-1} vers x_i d'une distance d_i le long de l'axe z_i .

Avec α_i , a_i , θ_i et d_i qui sont désignés comme les paramètres de Denavit-Hartenberg. Parmi ces 4 paramètres, α_i et a_i sont constants et dépendent de la géométrie du corps i tandis que d_i est variable si l'axe i constitue une liaison prismatique (θ_i restant constant) et θ_i est variable si l'axe i est une liaison rotoïde (d_i restant constant).

Le tableau 2.2 donne les valeurs des paramètres de Denavit-Hartenberg du modèle géométrique du robot SCEMI. Les valeurs minimales et maximales de la rotation de chaque axe sont précisées.

² Notez que le choix de la direction de z_i est laissé libre ; il existe donc 2 directions possibles pour z_i , l'une étant l'opposée de l'autre.

i	a_i	α_i	d_i	θ_i	$\min(\theta_i)$	$\max(\theta_i)$
1	0	$\pi/2$	r_1	q_1	-2.11	2.11
2	d_3	π	0	q_2	0.13	1.958
3	0	$-\pi/2$	0	q_3	0	1.83
4	0	$\pi/2$	r_4	q_4	-2.99	2.99
5	0	$-\pi/2$	0	q_5	-1.56	1.56
6	0	0	0	q_6	-2.99	2.99

Tab. 2.2: Paramètres de Denavit-Hartenberg du robot SCEMI

La matrice homogène $M_{(i-1)(i)}$ de transformation entre R_{i-1} et R_i s'exprime avec ces 4 paramètres dans le repère R_{i-1} de la manière suivante [Cra89] :

$$M_{(i-1)(i)} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}_{R_{i-1}} \quad (2.24)$$

On peut ainsi définir 6 matrices homogènes réalisant les transformations entre deux corps successifs. Ces matrices ont été reportées en annexe B.1 pour le cas du robot SCEMI.

D'autre part, afin de définir la position de la caméra par rapport au centre de la rotule constituée par les axes 4, 5 et 6, on introduit une matrice homogène supplémentaire, M_{6c} . Elle donne la transformation entre R_6 et R_c , le repère caméra. Par convention, l'origine du repère R_c est placée au centre optique de la caméra, son axe z_c est perpendiculaire au plan image de la caméra, son axe x_c est vertical et dirigé vers le haut dans l'image et son axe y_c est horizontal. Ainsi, comme le montre la figure 2.10, M_{6c} se limite à une translation définie par 3 paramètres : T_{xc} , T_{yc} et T_{zc} :

$$M_{6c} = \begin{bmatrix} 1 & 0 & 0 & T_{xc} \\ 0 & 1 & 0 & T_{yc} \\ 0 & 0 & 1 & T_{zc} \\ 0 & 0 & 0 & 1 \end{bmatrix}_{R_6} \quad (2.25)$$

Nous choisissons le repère de base du robot R_b égal à R_0 . Par conséquent, la transformation M_{bc} entre le repère de base du robot, R_b et le repère caméra, R_c , exprimée dans le repère de base s'obtient par multiplication de toutes les matrices homogènes entre elles :

$$M_{bc} = M_{01}M_{12}M_{23}M_{34}M_{45}M_{56}M_{6c} \quad (2.26)$$

2.4.3 Le modèle dynamique

Le modèle dynamique du robot SCEMI peut être décomposé en 3 parties :

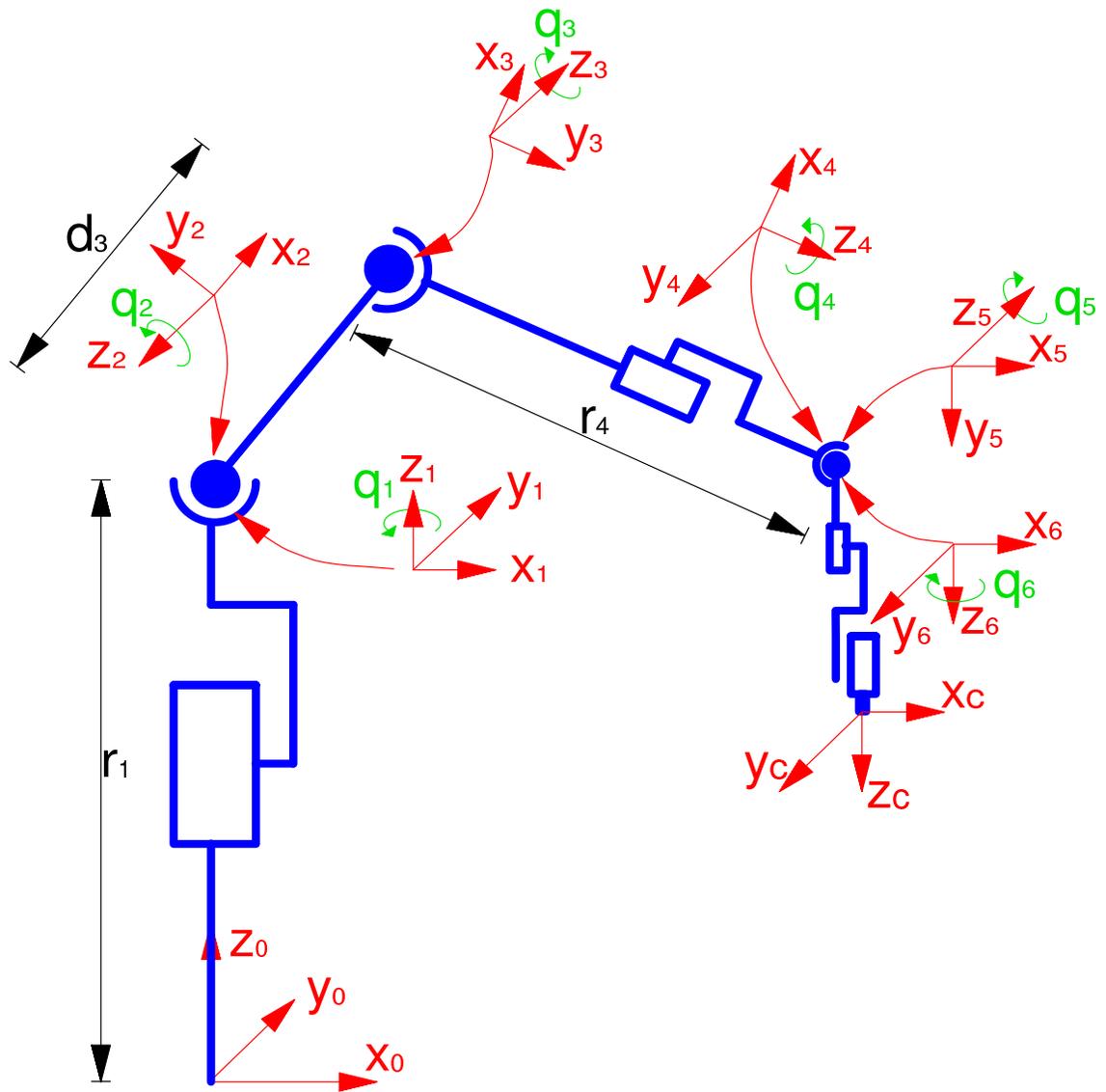


Fig. 2.10: Le modèle géométrique du robot SCEMI

1. Le modèle dynamique des actionneurs
2. Le modèle dynamique des transmissions
3. Le modèle dynamique du robot

Afin de modéliser encore plus finement le robot, on peut également tenir compte d'un modèle évolué des frottements. Ce modèle a été étudié en détails par Hansjörg Sage dans le cadre de sa thèse de doctorat [Sag97, SMA⁺99].

Le modèle dynamique des actionneurs

Le robot SCEMI est constitué de 6 actionneurs qui sont des moteurs à courant continu. Ils sont alimentés par des amplificateurs de tension qui sont des hacheurs à 4 quadrants. La tension à leurs bornes peut être commandée dans une plage comprise entre $-U_{max}$ et $+U_{max}$ où U_{max} est la tension continue d'alimentation des hacheurs. De plus, le courant d'induit du moteur peut circuler dans n'importe quel sens quelque soit la tension d'induit [FF94]. Le hacheur étant un amplificateur linéaire de tension, il existe une matrice diagonale $K_h \in \mathbb{R}^{6 \times 6}$ contenant les rapports entre le vecteur des tensions de commande u du hacheur et le vecteur des tensions d'induit, u_m . L'équation modélisant le hacheur est donc la suivante :

$$u_m = \begin{cases} K_h u & \text{si } -U_{max}I \leq K_h u \leq +U_{max}I \\ U_{max}I & \text{si } K_h u > +U_{max}I \\ -U_{max}I & \text{si } K_h u < -U_{max}I \end{cases} \quad (2.27)$$

Pour un moteur à courant continu, l'équation différentielle liant le courant induit, la vitesse de rotation du rotor et la tension d'induit est la suivante [SN94] :

$$Ri_m + L \frac{di_m}{dt} + K_e \frac{dq_m}{dt} = u_m \quad (2.28)$$

où :

- $R \in \mathbb{R}^{6 \times 6}$ est une matrice diagonale contenant les valeurs des résistances des induits.
- $i_m \in \mathbb{R}^6$ est un vecteur contenant les courants induits.
- $L \in \mathbb{R}^{6 \times 6}$ est une matrice diagonale contenant les inductances des induits.
- $K_e \in \mathbb{R}^{6 \times 6}$ est une matrice diagonale contenant les coefficients de force contre-électromotrice.
- $q_m \in \mathbb{R}^6$ contient les mesures de position des arbres moteurs.
- $u_m \in \mathbb{R}^6$ contient les tensions d'induit appliquées aux moteurs.

Puisque tous les moteurs du robot SCEMI sont à aimants permanents, le vecteur couple moteur $\tau_m \in \mathbb{R}^6$ est proportionnel au courant induit :

$$\tau_m = K_i i_m \quad (2.29)$$

avec $K_i = K_e \in \mathbb{R}^6$, le vecteur des coefficients de couple des moteurs.³ L'équation différentielle décrivant le comportement de la partie mécanique du moteur est la

³ On peut noter que pratiquement toutes les commandes d'actionneur en robotique sont basées sur un contrôle du couple (moteur à courant continu, moteur synchrone auto-piloté, moteur

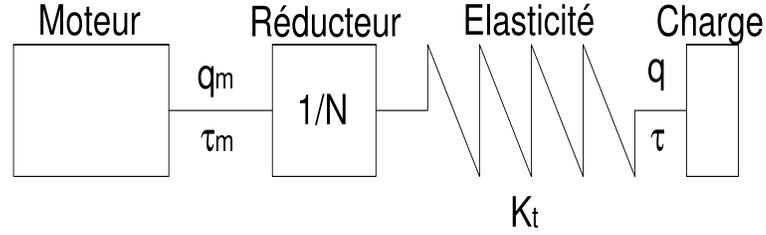


Fig. 2.11: Modélisation de la flexibilité de la transmission

suivante :

$$\tau_m = J_m \ddot{q}_m + N^{-1} \tau + f_v \dot{q}_m \quad (2.30)$$

où :

- $J_m \in \mathbb{R}^{6 \times 6}$ est une matrice diagonale contenant toutes les inerties de l'arbre moteur.
- $N \in \mathbb{R}^{6 \times 6}$ est une matrice diagonale telle que $N > I$ contenant les rapports de réduction des engrenages des axes.
- $\tau \in \mathbb{R}^6$ est un vecteur contenant les couples résistants dus à la charge appliquée à chaque axe.
- $f_v \in \mathbb{R}^6$ contient les coefficients de couples de frottements visqueux de chaque arbre moteur.

Le modèle dynamique des transmissions

Dans le cas du robot SCEMI, les flexibilités sont loin d'être négligeables. Elles sont principalement présentes sur les 3 premiers axes et sont dues aux réducteurs de type *Harmonic Drive*. Une modélisation de la flexibilité des transmissions d'un robot est proposée dans [Spo87]. Celle-ci est modélisée par un ressort de raideur K_t situé entre la sortie du réducteur et la charge (*cf.* figure 2.11). La liaison entre le moteur et la charge n'est donc pas rigide ($q \neq q_m/N$). Le couple τ appliqué à la charge est donc généré par la différence entre les angles q_m/N et q :

$$\tau = K_t \left(\frac{q_m}{N} - q \right) \quad (2.31)$$

où K_t est une matrice diagonale 6×6 contenant les coefficients de raideur des ressorts.

Le modèle dynamique du robot

Le modèle dynamique du robot SCEMI découle de l'équation générale de la dynamique (2.22) présentée dans la section 2.3. Nous l'appliquons ici au cas particulier

asynchrone avec commande vectorielle ou moteur à réluctance variable). Ce contrôle du couple est réalisé avec une boucle d'asservissement plus ou moins compliquée. Ainsi, dans le cas du moteur à courant continu, un simple correcteur PI permet de réaliser cette fonction. Nous reviendrons sur les détails de cet asservissement dans cette même section.

d'un robot possédant 6 axes en rotation :

$$\Gamma = \frac{d}{dt} \frac{\partial \chi}{\partial \dot{q}} - \frac{\partial \chi}{\partial q} + \frac{\partial \varpi}{\partial q} \quad (2.32)$$

avec :

- χ , l'énergie cinétique du robot.
- ϖ , l'énergie potentielle du robot.
- $q \in \mathbb{R}^6$, le vecteur des 6 positions angulaires des axes du robot.
- $\Gamma \in \mathbb{R}^6$, le vecteur des 6 couples extérieurs appliqués aux 6 axes du robot par les actionneurs.

Il en découle l'équation différentielle non-linéaire suivante :

$$\begin{aligned} D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + F_v\dot{q} + F_g(q) &= \tau \\ \text{avec } \tau - F_v\dot{q} - F_g(q) &= \Gamma \end{aligned} \quad (2.33)$$

où :

- $D(q) \in \mathbb{R}^{6 \times 6}$ est la matrice d'inertie des bras du robot.
- $C(q, \dot{q})\dot{q} \in \mathbb{R}^6$ est le vecteur des couples centrifuges et des couples de Coriolis.
- $g(q) \in \mathbb{R}^6$ est le vecteur des couples engendrés par la gravité.
- $F_g(q) \in \mathbb{R}^6$ est le vecteur des couples extérieurs provoqués par les ressorts de compensation de la gravité.
- $F_v \in \mathbb{R}^{6 \times 6}$ est la matrice des frottements visqueux des arbres de sortie du réducteur.

Les asservissements imbriqués de courant et de vitesse

Les actionneurs de tous les robots industriels sont asservis en couple (s'il s'agit d'un actionneur rotatif) ou en force (s'il s'agit d'un actionneur en translation). En effet, la maîtrise de cette grandeur est primordiale pour commander le mouvement d'un robot ne serait-ce que pour les tâches requérant un contrôle en effort. Mais bien souvent, la consigne de couple ou de force est elle-même une grandeur de commande d'une boucle externe de vitesse. Cette boucle de vitesse joue un rôle de linéarisation du modèle dynamique de l'actionneur. Dans le cas où l'actionneur est un moteur électrique, la maîtrise du couple va de pair avec la maîtrise du courant. Sa valeur, étant données les caractéristiques des amplificateurs de puissance, ne doit en aucun cas dépasser une limite maximale. C'est pourquoi, la grandeur de commande de la boucle de vitesse est saturée par une limite I_{max} .

Dans le cas du robot SCEMI, les asservissements qui contrôlent les couples sont des asservissements analogiques du courant d'induit, i_m , des moteurs à courant continu. Ces boucles sont contrôlées par des correcteurs à action Proportionnelle de matrice de gain proportionnel $K_{pi} \in \mathbb{R}^{6 \times 6}$. La matrice K_{pi} est diagonale car les axes sont découplés. Le vecteur de référence de l'asservissement de courant est noté i_m^* . Cet asservissement est imbriqué dans une boucle analogique de vitesse externe contrôlée par 2 correcteurs, un PI en série et un PD sur la mesure de vitesse. Un dispositif d'anti-saturation permet d'éviter au terme intégral de prendre une valeur

prohibitives, et ainsi d'éviter des grands dépassements en cas de saturation de la boucle interne de courant. $K_{pv} \in \mathbb{R}^{6 \times 6}$ est la matrice de gain proportionnel de cette boucle, $K_{iv} \in \mathbb{R}^{6 \times 6}$ est la matrice de gain intégral et $K_{dv} \in \mathbb{R}^{6 \times 6}$ est la matrice de gain dérivé. Les matrices K_{pv} , K_{iv} et K_{dv} sont diagonales car il n'y a aucune action croisée entre deux boucles. Le vecteur de référence de la boucle de vitesse est noté \dot{q}^* .

Le modèle dynamique complet du robot

La figure 2.12 donne une représentation synthétique du modèle dynamique du robot. Certaines non-linéarités comme les jeux, les *offsets* des amplificateurs ou encore les frottements secs ont été omises. Sur cette figure, les sous-parties du modèle ont été différenciées par des couleurs.

Quelques remarques d'ordre pratique concernant ce modèle s'imposent :

1. Les boucles de courant ont une bande passante de l'ordre de $1kHz$ sur le robot SCEMI. Ceci implique qu'en regard des constantes de temps mécaniques des moteurs, $\frac{J_m}{f_v}$, qui sont de l'ordre de quelques dizaines de ms , le retour de la force contre-électromotrice, $K_e \dot{q}_m$, se comporte comme une perturbation lente par rapport à la dynamique de la boucle de courant.
2. Seuls les axes 2 et 3 sont influencés de manière significative par la gravité. En effet, les axes du poignet du robot (4, 5 et 6) sont très légers et courts. Le couple dû à la gravité est donc très faible sur ces axes. L'axe 1 est parallèle à la direction du vecteur d'accélération de la gravité, donc aucun couple n'est produit. Sur le SCEMI, un système de compensation de la gravité constitué de ressorts permet d'annuler approximativement cette dernière quelque soit la configuration du robot. On peut donc raisonnablement faire l'approximation $g(q) \approx 0$.
3. L'effet des couples centrifuges et des couples de Coriolis ne se font ressentir qu'en cas de rotation très rapide des axes. Dans le cas d'un asservissement par vision, les vitesses de déplacement ne sont pas encore telles que ces couples aient une influence significative. De plus, les boucles de vitesse tendent à linéariser les effets de ces couples dont l'action peut être modélisée comme une perturbation. Nous considérerons donc que $C(q, \dot{q})\dot{q} = 0$.
4. La variation de l'inertie s'observe surtout sur les deux premiers axes. En effet, c'est pour ces 2 axes que la distance de l'axe de rotation au centre de gravité de la masse située en aval est la plus susceptible de varier d'une manière importante. Il est néanmoins possible de tenir compte d'un modèle approché de l'inertie $D(q)$ pour quelques configurations du robot. Ainsi, dans le cas de l'axe 1, on mesurera $D_1(q_1)$ pour les axes 2 et 3 repliés vers le haut (inertie minimale), pour les axes 2 et 3 pliés à 45 degrés (inertie moyenne) et pour les axes 2 et 3 horizontaux (inertie maximale).

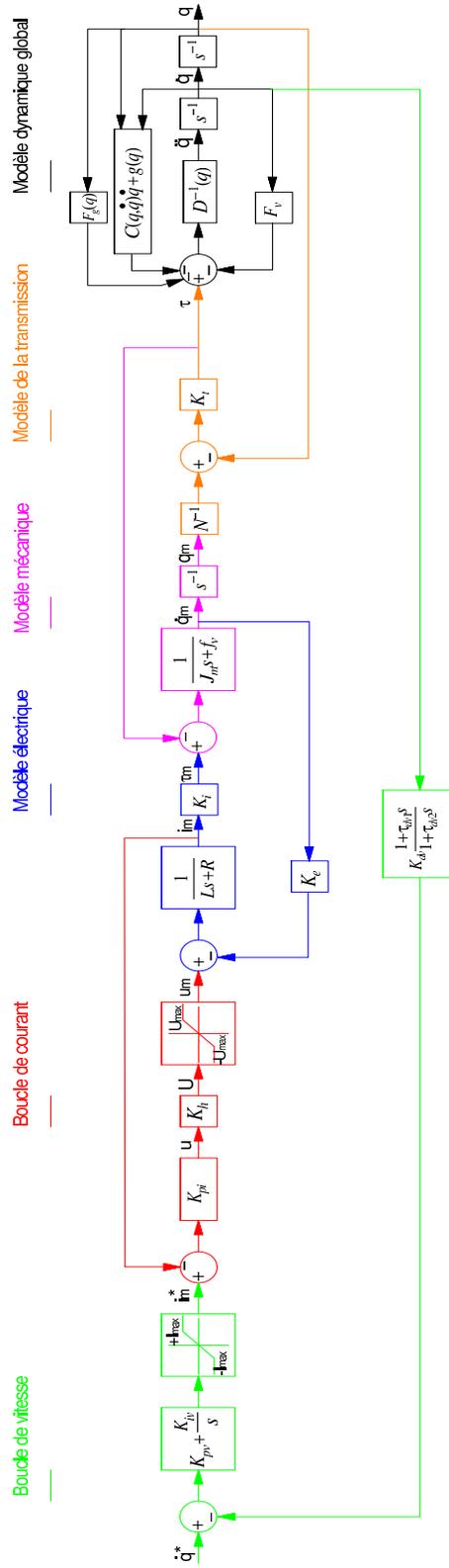


Fig. 2.12: Modèle dynamique complet du robot

2.4.4 Identification du modèle dynamique d'un axe asservi en vitesse

Nous faisons l'hypothèse que les asservissements de vitesse articulaire permettent de découpler et de linéariser le comportement dynamique du robot. Aussi, les couplages entre les axes et les effets des non-linéarités sont modélisés par des perturbations.

La fonction de transfert $F_i(z)$ entre la référence de vitesse \dot{q}^* et la mesure de vitesse articulaire du $i^{\text{ème}}$ axe du robot est identifiée sous la forme d'un modèle ARMAX [Lju87]. Comme nous l'avons vu dans la section précédente, cette fonction de transfert varie en fonction de la position q du robot. Aussi, nous réalisons l'identification pour une position nominale du robot autour de laquelle les expériences sont effectuées.

Nous avons donc identifié les 6 fonctions de transfert $F_i(z)$ avec une période d'échantillonnage de $1ms$. Pour ce faire, une séquence binaire pseudo-aléatoire est appliquée en référence de la boucle de vitesse de l'axe identifié. La mesure de la vitesse est faite grâce à un oscilloscope numérique qui échantillonne à la fréquence de $1kHz$ le signal analogique de la génératrice tachymétrique ainsi que le signal analogique de référence de vitesse.

L'oscilloscope que nous utilisons est pourvu d'un système de transfert des données numérisées. Aussi, est il possible de traiter ces données avec un programme tel que MATLAB et de réaliser l'identification paramétrique de $F_i(z)$.

Pour le robot SCEMI, ces fonctions de transfert ont été identifiées à l'ordre 6. Leur valeurs numériques ainsi que leurs réponses indicielles sont données en annexe B.2.

Chapitre 3

Modélisation de l'asservissement visuel

3.1 Modélisation dynamique d'une boucle de vision

Dans la configuration *eye in hand*, la grandeur à asservir est souvent l'attitude p d'un repère lié à un objet contenu dans l'espace du travail du robot par rapport à un repère solidaire de l'organe terminal.

Une attitude est définie de manière unique par 6 coordonnées opérationnelles. Afin que l'asservissement soit commandable, il est nécessaire de pouvoir maîtriser ces 6 paramètres.

Dans le cas d'un asservissement visuel direct, les grandeurs de commande appliquées au robot sont une consigne de vitesse \dot{q}^* ou une consigne d'accélération \ddot{q}^* des coordonnées articulaires. Celles-ci résultent de la transformation des commandes de vitesse \dot{p}^* ou d'accélération \ddot{p}^* générés par la boucle de vision dans l'espace des coordonnées articulaires q .

Il est donc nécessaire de faire une transformation de l'espace des coordonnées opérationnelles vers celui des coordonnées articulaires afin de générer les consignes \dot{q}^* ou \ddot{q}^* pour les asservissements des axes du robot. Cette transformation est obtenue grâce à la matrice Jacobienne du robot. Etant donné que la grandeur mesurée asservie (p) et la grandeur de commande de l'asservissement visuel (\dot{p}^* ou \ddot{p}^*) sont toutes les deux dans l'espace des coordonnées opérationnelles, on peut considérer le robot et sa matrice Jacobienne comme un *dispositif de déplacement opérationnel virtuel*.

Dans un contexte d'asservissement visuel rapide, les dynamiques du manipulateur doivent être prises en compte afin de réaliser une commande optimale de la boucle de vision. Si le robot est modélisé par un dispositif de déplacement opérationnel virtuel, il est donc intéressant d'établir la fonction de transfert de ce dispositif entre les commandes \dot{p}^* ou \ddot{p}^* et le déplacement effectif p en sortie.

Nous proposons dans cette section un calcul de la fonction de transfert entre \dot{p}^* et p dans le cas où les axes du robot sont asservis en vitesse. Ces asservissements internes de vitesse permettent de linéariser et de découpler le comportement dyna-

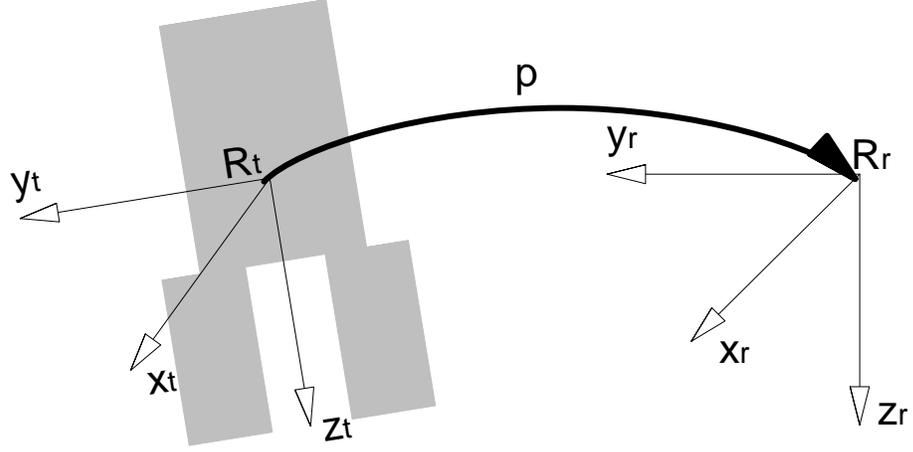


Fig. 3.1: Définition des repères

mique de chaque axe. Notre approche consiste à projeter les dynamiques articulaires du robot dans l'espace des coordonnées opérationnelles.

3.1.1 Définition du Jacobien de la caméra

Nous supposons dans cette section que le robot possède 6 axes et est capable de déplacer l'organe terminal suivant 6 degrés de liberté. Soit R_t , un repère lié à l'organe terminal du robot et R_r , un repère fixe par rapport à la base du robot. On définit p comme étant l'attitude de R_r par rapport à R_t (voir figure 3.1). La position de R_r par rapport à R_t dépend des coordonnées articulaires $q = [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]^T$ du robot. Soit M_{bt} , la matrice homogène de transformation entre le repère de base du robot, R_b , et le repère R_t . Cette matrice homogène se calcule grâce au modèle géométrique du robot et dépend donc de q , nous la noterons $M_{bt}(q)$. Soit $M_{tr}(q)$, la matrice homogène de transformation entre R_t et R_r qui dépend elle aussi de q :

$$M_{tr}(q) = M_{bt}(q)^{-1} M_{br} \quad (3.1)$$

où M_{br} est la matrice homogène de transformation entre R_b et R_r . La transformation M_{br} est constante car le repère R_r est supposé être fixe par rapport à R_b . Soit T_6 , la transformation permettant de convertir une matrice homogène en 6 coordonnées opérationnelles (voir annexe B.3 pour la définition de T_6). On a donc :

$$p(q) = T_6(M_{tr}(q)) \quad (3.2)$$

L'attitude p dans le repère R_t étant dépendante des coordonnées articulaires, on la note $p(q)$. On peut donc réécrire $p(q)$ comme un vecteur 6×1 de fonctions du vecteur q :

$$p(q) = \begin{bmatrix} t_1(q) \\ \vdots \\ t_6(q) \end{bmatrix}_{R_t} \quad (3.3)$$

On en déduit que $\dot{p}(q)$, la dérivée par rapport au temps de $p(q)$ est donnée par :

$$\dot{p}(q) = \frac{d}{dt}[T_6(M_{tr}(q))] = \begin{bmatrix} \nabla t_1^T(q) \\ \vdots \\ \nabla t_6^T(q) \end{bmatrix}_{R_t} \dot{q} \quad (3.4)$$

d'où :

$$\dot{p}(q) = \begin{pmatrix} \frac{\partial t_1}{\partial q_1}(q) & \cdots & \frac{\partial t_1}{\partial q_6}(q) \\ \vdots & \ddots & \vdots \\ \frac{\partial t_6}{\partial q_1}(q) & \cdots & \frac{\partial t_6}{\partial q_6}(q) \end{pmatrix} \dot{q} \triangleq J_c(q)\dot{q} \quad (3.5)$$

La matrice $J_c(q)$ permet de transformer les vitesses des coordonnées articulaires en vitesses des coordonnées opérationnelles. Dans le cas d'un asservissement visuel, pour une configuration *eye in hand*, on prend $R_t = R_c$ où R_c est le repère lié à la caméra solidaire de l'organe terminal (voir section 2.1.2, page 20). Pour cette raison, nous appelons J_c , *Jacobien de la caméra*.

Enfin, si on désire contrôler le robot comme un dispositif de déplacement opérationnel virtuel, on utilisera J_c^{-1} pour convertir des références opérationnelles \dot{p}^* en références articulaires \dot{q}^* compatibles avec la commande du robot asservi en vitesse.

Remarques

- L'expression analytique de $J_c(q)$ est très lourde pour un robot à 6 degrés de liberté. En pratique, il n'est pas nécessaire de la calculer explicitement. En effet, le calcul de $\frac{\partial t_i}{\partial q_j}(q)$ peut être approximé par des différences finies :

$$\frac{\partial t_i}{\partial q_j}(q) \approx \frac{\Delta t_i}{\Delta q_j}(q) = \frac{t_i \left(\begin{bmatrix} q_1 \\ \vdots \\ q_{j-1} \\ q_j + \Delta q_j \\ q_{j+1} \\ \vdots \\ q_6 \end{bmatrix} \right) - t_i \left(\begin{bmatrix} q_1 \\ \vdots \\ q_j \\ \vdots \\ q_6 \end{bmatrix} \right)}{\Delta q_j} \quad (3.6)$$

Ainsi, le calcul de $J_c(q)$ peut se réaliser numériquement avec une précision suffisante en réalisant le calcul décrit par l'équation (3.6) pour toutes les combinaisons de i et de j .

- Dans le cas d'une tâche de suivi, R_r est lié rigidement à l'objet qui est suivi. Ainsi, si cet objet se déplace, on ne peut pas considérer que R_r est fixe. Néanmoins, $J_c(q)$ est réactualisé à chaque période d'échantillonnage en fonction de l'évolution de q . Aussi, à l'échelle d'une période d'échantillonnage, étant

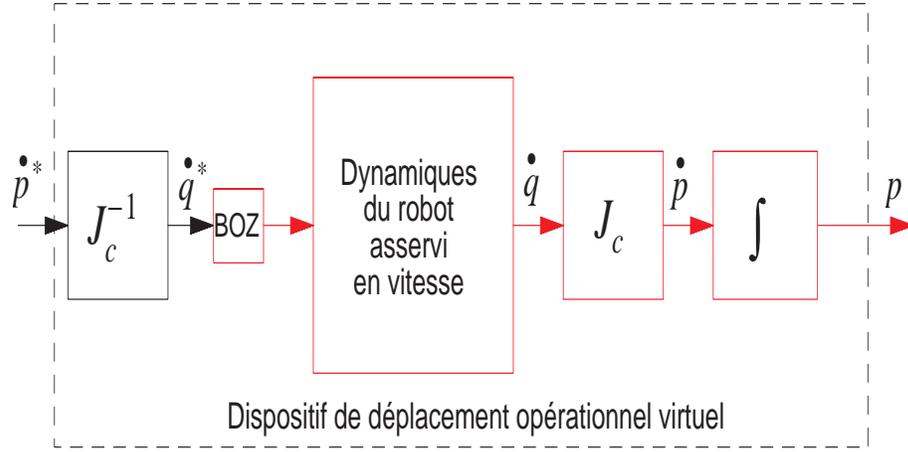


Fig. 3.2: Modèle du dispositif de déplacement opérationnel virtuel

donnée une vitesse de déplacement réaliste de l'objet, on peut faire l'approximation que R_r est fixe. Cette approximation est d'autant plus juste que la période d'échantillonnage est courte.

3.1.2 Modélisation du dispositif de déplacement opérationnel virtuel

Dans cette section, nous nous proposons d'établir le modèle dynamique du dispositif de déplacement opérationnel virtuel reliant la commande \dot{p}^* à l'attitude p décrit par la figure 3.2.

Le robot possède 6 axes permettant de positionner l'organe terminal suivant 6 degrés de liberté. Soit $F(z) \in \mathbb{R}^{6 \times 6}$, la fonction de transfert du robot asservi en vitesse. Nous considérons que la matrice $F(z)$ est diagonale ; elle contient sur sa diagonale les fonctions de transfert $F_i(z)$ des 6 boucles de vitesse articulaire du robot. Nous faisons donc l'hypothèse que les couplages entre les axes sont négligeables, ce qui est valide lorsque le robot est asservi en vitesse. De plus, nous supposons que les fonctions de transfert F_i sont normalisées, donc $\lim_{z \rightarrow 1} F(z) = I_6$.

La fonction de transfert $F(z)$ est une linéarisation du modèle dynamique du robot asservi en vitesse autour d'une position articulaire q . En effet, ce modèle dynamique est susceptible de varier lorsque q varie. Néanmoins, dans la pratique, on peut considérer que ce modèle ne varie de manière significative que pour des positions extrêmes du robot.

Les non-linéarités telles que les forces centrifuges, les forces de Coriolis ou les effets de la gravité sont négligées dans cette modélisation (voir sections 2.3 et 2.4 pour plus de détails). Etant donné que le robot est asservi en vitesse, on peut considérer qu'elles agissent comme des perturbations.

Soient les matrices J_c et J_c^{-1} , respectivement le Jacobien de la caméra et son

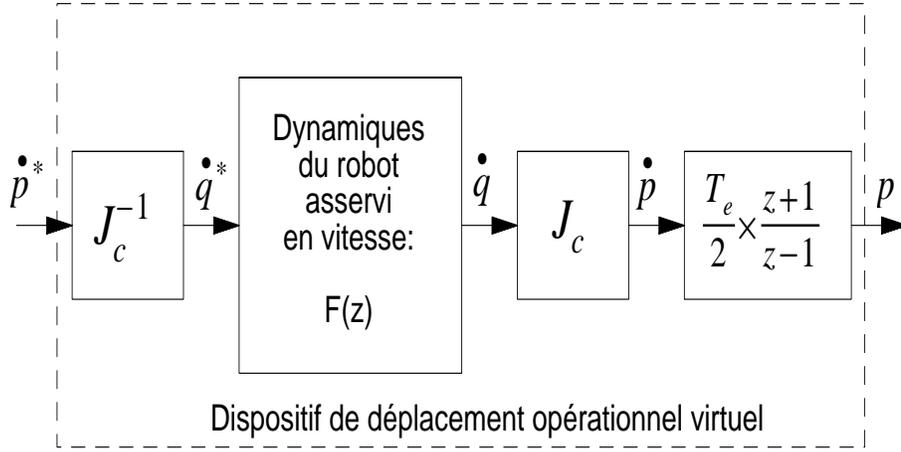


Fig. 3.3: Modèle linéarisé autour de q du dispositif de déplacement opérationnel virtuel

inverse. La relation

$$\dot{q}^* = J_c^{-1} \dot{p}^* \quad (3.7)$$

permet de traduire les consignes de vitesse opérationnelle \dot{p}^* dans l'espace articulaire. Les consignes de vitesse articulaire \dot{q}^* sont appliquées aux boucles de vitesse par l'intermédiaire d'un bloqueur d'ordre 0. Le robot asservi en vitesse étant modélisé par $F(z)$, on a donc :

$$\dot{q} = F(z) \dot{q}^* \quad (3.8)$$

Les mesures de vitesse articulaire \dot{q} sont exprimées dans l'espace opérationnel par :

$$\dot{p} = J_c \dot{q} \quad (3.9)$$

Une intégration de \dot{p} par la méthode des trapèzes permet d'obtenir p :

$$p = \frac{T_e}{2} \frac{z+1}{z-1} \dot{p} \quad (3.10)$$

où T_e est la période d'échantillonnage de l'asservissement visuel. Nous en déduisons le modèle dynamique du dispositif de déplacement opérationnel virtuel linéarisé autour d'une configuration q du robot décrit par la figure 3.3.

Des équations 3.7, 3.8, 3.9 et 3.10, on déduit la fonction de transfert $H(z)$ du dispositif de déplacement opérationnel virtuel :

$$H(z) = \frac{p}{\dot{p}^*} = \frac{T_e}{2} \frac{z+1}{z-1} J_c F(z) J_c^{-1} \quad (3.11)$$

Soit $G(z)$, la fonction de transfert entre \dot{p}^* et \dot{p} :

$$G(z) = \frac{\dot{p}}{\dot{p}^*} = J_c F(z) J_c^{-1} \quad (3.12)$$

Comme $\lim_{z \rightarrow 1} F(z) = I_6$, on a $\lim_{z \rightarrow 1} G(z) = I_6$. Donc, d'après le théorème de la valeur finale, l'effet d'un échelon de la consigne de vitesse \dot{p}_k^* de l'axe k sur la vitesse \dot{p}_i de l'axe i , s'annule avec le temps. De plus, l'erreur statique de la réponse \dot{p}_i à un échelon \dot{p}_i^* est nulle.

Remarques

- La relation 3.10 est en fait une approximation. Elle est d'autant plus fautive que T_e augmente. En effet, la fonction de transfert $\frac{T_e z+1}{2(z-1)}$ réalise une intégration par la méthode des trapèzes définis par 2 échantillons successifs. Aussi, plus la période est faible et plus le résultat se rapproche de l'intégrale réelle. Il est néanmoins possible de déterminer $H(z)$ avec une bonne précision même si la période d'échantillonnage du système T_e est grande. Pour cela, il est nécessaire de disposer d'un modèle de $F_i(z)$ identifié à une période T_c très courte. Dans un premier temps, on en déduit $G(z)$, puis $H(z)$ pour cette période T_c . Comme T_c est faible, $H(z)$ est donc connu avec une bonne précision. Puis grâce à un calcul standard de conversion de la période d'échantillonnage d'une fonction de transfert, il est possible d'en déduire $H(z)$ pour la période T_e avec une bonne précision.

Par conséquent, nous avons identifié les fonctions de transfert $F_i(z)$ du robot SCEMI à une période d'échantillonnage de $1 \text{ ms} < T_e$ avant de calculer $H(z)$ pour la période T_e (voir section 2.4.4).

- Le Jacobien de la caméra J_c est défini pour une position q des axes du robot (voir équation (3.5)). Néanmoins, on peut considérer qu'il reste sensiblement constant dans une région autour de q . De plus les fonctions de transfert $F_i(z)$ ne peuvent être considérées constantes que dans une région autour de q (du moins pour les premiers axes du robot). Aussi, le modèle de transfert linéaire $H(z)$ du dispositif de déplacement opérationnel virtuel n'est valide que dans une région autour de q . Dans le contexte d'utilisation du robot dans un asservissement par vision, la position q du robot est amenée à changer de manière importante. La fonction de transfert $H(z)$ est donc également susceptible de varier dans des proportions importantes. Aussi, afin de réaliser un contrôle optimal de la boucle de vision, il est intéressant de faire appel à un correcteur adaptatif qui puisse intégrer en temps réel les variations de $H(z)$ par séquençement des gains en fonction de q .

3.1.3 Commande de la boucle de vision

Soit p , l'attitude d'un repère de référence fixe R_r par rapport à un repère R_t lié à l'organe terminal (voir figure 3.1 page 42). On suppose que p peut être estimé grâce à un capteur de vision fixé sur l'organe terminal. L'attitude p est asservie à une référence p^* grâce à la boucle d'asservissement décrite par la figure 3.4.

Les dynamiques du robot suivant les 6 coordonnées de p sont modélisées par la fonction de transfert $H(z)$ qui est la linéarisation autour de la position courante du

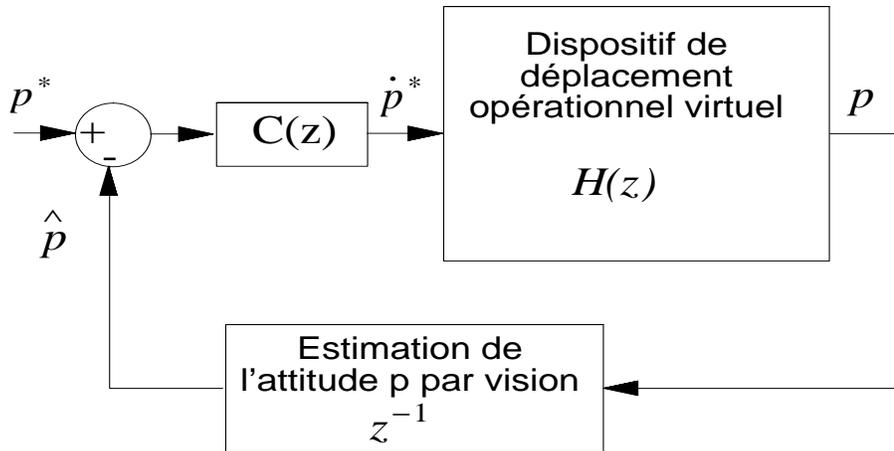


Fig. 3.4: Modèle général d'une boucle de vision

robot des dynamiques du dispositif de déplacement opérationnel virtuel décrit dans la section 3.1.2, page 44.

L'estimation \hat{p} de p est réalisée grâce à l'information visuelle. Or, d'après le modèle dynamique de la caméra décrit dans la section 2.2.4, page 27, cette estimation introduit un retard pur d'une période d'échantillonnage T_e si nous faisons l'hypothèse que le temps de traitement d'image est inférieur à T_e ; sa fonction de transfert est donc z^{-1} .

Deux types de correcteurs ont été utilisés pour le contrôle de la boucle de vision : un correcteur à action Proportionnelle-Intégrale (PI) et un correcteur prédictif *Generalized Predictive Controller* (GPC) dont le principe est décrit dans la section 3.2 page 48. Dans les deux cas, on a négligé les couplages entre les composantes de p ce qui équivaut à considérer que $H(z)$ est diagonale. L'asservissement peut donc être décomposé en 6 boucles indépendantes contrôlant chacune un degré de liberté.

Le correcteur PI est réglé de manière expérimentale pour obtenir les performances maximales tout en conservant la stabilité de la boucle. Le correcteur GPC est calculé en fonction du modèle dynamique du robot. Il est optimum autour de la position q du robot pour laquelle $H(z)$ a été calculé.

Le modèle dynamique $F(z)$ des boucles de vitesse articulaires du robot est requis pour le calcul de $H(z)$. Ce dernier est obtenu grâce à une identification préalable (voir section 2.4.4, page 40).

Le modèle $H(z)$ du dispositif de déplacement opérationnel virtuel varie en fonction de la position q du robot. Aussi, grâce à sa nature adaptative, le correcteur GPC peut être recalculé lorsque cela est nécessaire au gré des variations de $H(z)$ afin de toujours garantir une réponse optimale. Ce n'est pas le cas du correcteur PI qui doit être réglé de manière à conserver la stabilité même dans les configurations les plus défavorables.

3.2 Le correcteur GPC

Le correcteur GPC (*Generalized Predictive Controller*) a été introduit en 1987 par Clarke *et al.* dans [CMT87]. On peut définir le GPC comme étant une version simplifiée du correcteur LQG (*Linear Quadratic Gaussian*). Ils ont en commun la minimisation d'une fonction de coût dépendant de l'intégrale de l'erreur au carré et de l'énergie de la commande. La différence réside dans le fait que, pour le GPC, cette fonction est minimisée sur un horizon fuyant fini, tandis que pour le LQG, cette minimisation est faite sur un horizon s'étendant jusqu'à l'infini.

Cette section a pour objectif de fournir des informations détaillées sur les aspects théoriques du GPC. L'implémentation concrète du correcteur est également présentée en annexe. Néanmoins, le lecteur désirant obtenir des informations sur les variantes du GPC peut se référer au livre de Bitmead *et al.* [BGW90] ou de Dion *et al.* [DP96].

3.2.1 Mise en équation du système

Soit un système linéaire à une entrée et une sortie dont le comportement est modélisé par l'équation réursive suivante :

$$A(q^{-1})y(t) = B(q^{-1})u(t - T_e) + \frac{C(q^{-1})}{D(q^{-1})}\xi(t) \quad (3.13)$$

où :

- $u(t)$ est l'entrée du système,
- $y(t)$ est la sortie du système,
- $\{\xi(t)\}$ est une séquence de variables aléatoires indépendantes telles que $E\{\xi(t)\} = 0$,
- T_e est la période d'échantillonnage,
- q^{-1} est un opérateur de retard d'une période d'échantillonnage,
- A, B, C, D sont des polynômes en q^{-1} avec A, C et D moniques.¹

Les polynômes A, B, C, D sont respectivement d'ordre n_a, n_b, n_c, n_d :

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad (3.14)$$

$$B(q^{-1}) = b_0q^{-d} + b_1q^{-d-1} + \dots + b_{n_b}q^{-n_b-d} \quad (3.15)$$

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_{n_c}q^{-n_c} \quad (3.16)$$

$$D(q^{-1}) = 1 + d_1q^{-1} + \dots + d_{n_d}q^{-n_d} \quad (3.17)$$

où d est un entier positif et dT_e est le retard pur du système. Si $U(z) = \mathcal{Z}\{u(t)\}$, $Y(z) = \mathcal{Z}\{y(t)\}$ et $\Psi(z) = \mathcal{Z}\{\xi(t)\}$ avec \mathcal{Z} représentant la transformée en z , on peut mettre le système sous la forme suivante :

$$Y(z) = z^{-d-1} \frac{B(z^{-1})}{A(z^{-1})} U(z) + \frac{C(z^{-1})}{D(z^{-1})A(z^{-1})} \Psi(z) \quad (3.18)$$

$$= T_z(z^{-1})U(z) + P_z(z^{-1})\Psi(z) \quad (3.19)$$

¹ Un polynôme est monique si son coefficient de plus haut degré est égal à 1.

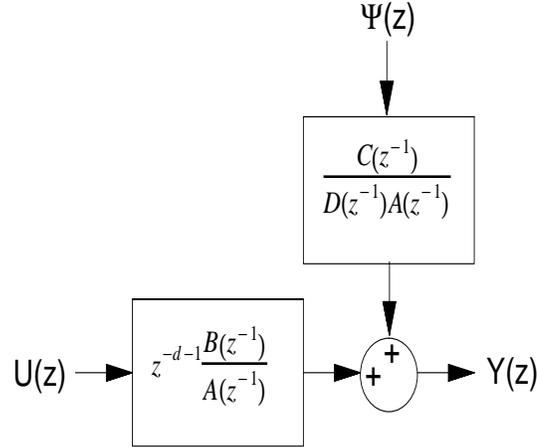


Fig. 3.5: Modèle linéaire du système

où T_z est la fonction de transfert du système et P_z est la fonction de transfert définissant le modèle de bruit (voir figure 3.5).

Dans le cas du correcteur GPC, on fixe $D(q^{-1}) = 1 - q^{-1} = \Delta$. On impose donc une partie du modèle du bruit afin de donner au correcteur synthétisé certaines caractéristiques. En effet, P_z contenant un intégrateur, le correcteur pourra ainsi annuler des perturbations constantes. Le polynôme $C(q^{-1})$ sera choisi de manière à modéliser le spectre du bruit ξ afin de générer un contrôleur qui filtre les bruits du système.

L'équation (3.13) peut donc réécrite sous la forme suivante :

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t - T_e) + C(q^{-1})\xi(t) \quad (3.20)$$

avec :

$$\begin{aligned} \Delta y(t) &= (1 - q^{-1})y(t) = y(t) - y(t - T_e) \\ \Delta u(t) &= (1 - q^{-1})u(t) = u(t) - u(t - T_e) \end{aligned} \quad (3.21)$$

L'équation (3.20) décrit donc l'évolution de la variation de la sortie Δy en fonction de la variation de l'entrée Δu entre deux instants d'échantillonnage successifs.

3.2.2 La fonction de coût

Le correcteur GPC est un correcteur prédictif. Son objectif est de minimiser l'erreur quadratique entre les prédictions de la sortie et les consignes futures. Cela signifie qu'il doit réaliser une estimation des prédictions des valeurs de la sortie y aux instants d'échantillonnage futurs en fonction des valeurs futures de l'entrée u . Soit r la grandeur de référence de l'asservissement dont la valeur est supposée connue non seulement à l'instant t présent mais également aux N_2 instants d'échantillonnage futurs. Le correcteur GPC génère, à chaque instant d'échantillonnage, N_u commandes

Δu de manière à minimiser la fonction de coût suivante : $J(u, t) =$

$$E \left\{ \sum_{j=N_1}^{N_2} [y(t + jT_e) - r(t + jT_e)]^2 + \lambda \sum_{j=1}^{N_u} [\Delta u(t + (j-1)T_e)]^2 \right\}$$

avec : $N_u < N_2$ et $\Delta u(t + jT_e) = 0 \quad \forall j \geq N_u$ (3.22)

où N_1 , N_2 et N_u sont des entiers strictement positifs et λ est un scalaire positif. Ces termes sont définis comme suit :

- N_1 est l'horizon d'initialisation,
- N_2 est l'horizon de prédiction,
- N_u est l'horizon de commande,
- λ est la pondération de la commande.

On peut remarquer que la fonction $J(u, t)$ comprend deux termes. Le premier, $\sum_{j=N_1}^{N_2} [y(t + jT_e) - r(t + jT_e)]^2$, est la somme des erreurs au carré entre les sorties et les signaux de référence futurs. Le second, $\sum_{j=1}^{N_u} [\Delta u(t + (j-1)T_e)]^2$, est un terme proportionnel à «l'énergie» fournie par la commande.²

Ce terme est pondéré par λ . Ainsi, plus λ est faible, et moins «l'énergie» de la commande est pénalisée dans la fonction de coût, et par conséquent, plus le correcteur GPC synthétisé est «énergique», et donc sa réponse est rapide. Ce terme permet d'éviter les signaux de commande trop importants pouvant saturer le système.

3.2.3 Calcul des prédictions de la sortie

Dans cette section, nous développons le calcul de $\hat{y}(t + jT_e)$, la prédiction à l'instant t de y à j pas d'échantillonnage en avance. Ce calcul nécessite la résolution de 2 équations Diophantiennes [BGW90].

Afin d'alléger les notations, un polynôme $X(q^{-1})$ sera simplement noté X dans cette section.

Soient E_j et F_j les solutions de l'équation Diophantienne suivante :

$$C = A\Delta E_j + q^{-j}F_j \quad (3.23)$$

avec :

$$E_j = 1 + e_1^{(j)}q^{-1} + \dots + e_{j-1}^{(j)}q^{-j+1} \quad (3.24)$$

$$F_j = f_0^{(j)} + f_1^{(j)}q^{-1} + \dots + f_{nf}^{(j)}q^{-nf} \quad (3.25)$$

et $nf^{(j)} = \max(na, nc - j)$. La résolution des équations (3.23) de manière récursive est décrite en annexe B.7. En réécrivant (3.20) pour le j -ème pas d'échantillonnage, on obtient :

$$A\Delta y(t + jT_e) = B\Delta u(t + (j-1)T_e) + C\xi(t + jT_e) \quad (3.26)$$

² En faisant une analogie avec la mécanique dans le cas où la grandeur de commande u est un couple.

En introduisant (3.23) dans cette équation, nous obtenons :

$$A\Delta y(t + jT_e) = B\Delta u(t + (j - 1)T_e) + A\Delta E_j \xi(t + jT_e) + F_j \xi(t) \quad (3.27)$$

En isolant $\xi(t)$ dans (3.20) et en utilisant cette relation dans (3.27) on obtient après simplification par $A\Delta$:

$$\begin{aligned} y(t + jT_e) &= \frac{F_j}{C} y(t) + E_j \xi(t + jT_e) \\ &\quad + \frac{BE_j}{C} \left(\frac{C}{AE_j} u(t + (j - 1)T_e) - \frac{F_j}{AE_j} u(t - T_e) \right) \end{aligned} \quad (3.28)$$

Grâce à l'équation Diophantienne (3.23), nous pouvons écrire que :

$$\begin{aligned} \frac{C}{AE_j} u(t + (j - 1)T_e) - \frac{F_j}{AE_j} u(t - T_e) &= \\ \Delta u(t + (j - 1)T_e) + q^{-j} \frac{F_j}{AE_j} u(t + (j - 1)T_e) - \frac{F_j}{AE_j} u(t - T_e) &= \\ &= \Delta u(t + (j - 1)T_e) \end{aligned} \quad (3.29)$$

En combinant (3.28) et (3.29) nous obtenons finalement :

$$y(t + jT_e) = \frac{F_j}{C} y(t) + \frac{E_j B}{C} \Delta u(t + (j - 1)T_e) + E_j \xi(t + jT_e) \quad (3.30)$$

Dans cette dernière équation, toutes les références à des valeurs de la perturbation pour des instants d'échantillonnage passés et présent ont été supprimées. Il ne subsiste donc plus que la combinaison linéaire $E_j \xi(t + jT_e)$ de valeurs futures de la perturbation ou du bruit. Or ces valeurs sont par définition indépendantes de signaux mesurables à l'instant t . Il est donc clair que la prédiction optimale à l'instant t de $y(t + jT_e)$ notée $\hat{y}(t + jT_e)$ au sens du minimum de la variance de l'erreur, obtenue grâce à des grandeurs dont la valeur est connue à l'instant t est donnée par :

$$\hat{y}(t + jT_e) = \frac{F_j}{C} y_t + \frac{E_j B}{C} \Delta u(t + (j - 1)T_e) \quad (3.31)$$

On peut noter que ce résultat est classique pour la détermination de prédictions optimales [Åst70]. Il met clairement en évidence que le terme C permet de modifier la dynamique de l'observateur (le prédicteur de la sortie) de la même façon que dans un filtre de Kalman. Ainsi, l'ajustement de C dans le modèle (3.13) permet de conférer au correcteur des propriétés de filtrage du bruit.

Dans l'équation (3.31), le terme $E_j B \Delta u(t + (j - 1)T_e)$ est une combinaison linéaire de valeurs de Δu à des instants d'échantillonnage compris entre les instants $t - (nb + d)T_e$ et $t + (j - 1 - d)T_e$. Mais la fonction de coût (3.22), implique seulement les valeurs futures et la valeur présente de Δu . Aussi allons nous chercher à séparer

ces valeurs des valeurs passées dans l'équation (3.31). Cette séparation peut être obtenue grâce à la résolution d'une seconde équation Diophantienne en G_j et H_j

$$E_j B = G_j C + q^{-j} H_j \quad (3.32)$$

avec :

$$G_j = g_0^{(j)} + g_1^{(j)} q^{-1} + \dots + g_{j-1}^{(j)} q^{-j+1} \quad (3.33)$$

$$H_j = h_0^{(j)} + h_1^{(j)} q^{-1} + \dots + h_{nh}^{(j)} q^{-nh} \quad (3.34)$$

et $nh = \max(nc, nb + d - 1)$. Les équations Diophantiennes (3.32) peuvent être résolues récursivement. Ce calcul est proposé en annexe B.7.

L'utilisation de (3.32) dans (3.31) conduit à :

$$\hat{y}(t + jT_e) = \frac{F_j}{C} y_t + G_j \Delta u(t + (j - 1)T_e) + \frac{H_j}{C} \Delta u(t - T_e) \quad (3.35)$$

3.2.4 Calcul de la solution optimale

Soit $f_p \in \mathbb{R}^{N_2 - N_1 + 1}$, un vecteur constitué de la prédiction de la réponse libre du système pour les instants d'échantillonnage $\geq t + N_1 T_e$ (horizon d'initialisation) et $\leq t + N_2 T_e$ (horizon de prédiction) :

$$f_p = \begin{bmatrix} \frac{H_{N_1}}{C} \Delta u(t - T_e) + \frac{F_{N_1}}{C} y(t) \\ \frac{H_{N_1+1}}{C} \Delta u(t - T_e) + \frac{F_{N_1+1}}{C} y(t) \\ \vdots \\ \frac{H_{N_2}}{C} \Delta u(t - T_e) + \frac{F_{N_2}}{C} y(t) \end{bmatrix} \quad (3.36)$$

Soit $\tilde{u} \in \mathbb{R}^{N_u}$, le vecteur contenant les incréments de u sachant que $\Delta u(t + jT_e) = 0 \forall j \geq N_u$ (cf. équation (3.22)) :

$$\tilde{u} = [\Delta u(t), \Delta u(t + T_e), \dots, \Delta u(t + (N_u - 1)T_e)]^T \quad (3.37)$$

Soit $\hat{Y} \in \mathbb{R}^{N_2 - N_1 + 1}$, le vecteur contenant les prédictions des sorties à partir de l'horizon d'initialisation :

$$\hat{Y} = [\hat{y}(t + N_1 T_e), \hat{y}(t + (N_1 + 1)T_e), \dots, \hat{y}(t + N_2 T_e)]^T \quad (3.38)$$

On peut donc réécrire l'équation (3.35) pour $N_1 \leq j \leq N_2$ sous forme matricielle :

$$\hat{Y}_i = G \tilde{u} + f_p \quad (3.39)$$

avec $G \in \mathbb{R}^{(N_2 - N_1 + 1) \times N_u}$ défini comme suit :

$$G = \begin{pmatrix} g_{N_1-1} & \dots & g_0 & \dots & \dots & 0 \\ g_{N_1} & g_{N_1-1} & \dots & g_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ g_{N_u-1} & g_{N_u-2} & g_{N_u-3} & \dots & \dots & g_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{N_2-1} & g_{N_2-2} & g_{N_2-3} & \dots & g_{N_2-N_u+1} & g_{N_2-N_u} \end{pmatrix} \quad (3.40)$$

Les termes g_k avec $0 \leq k \leq N_2 - 1$ sont les coefficients du polynôme G_j (voir équation (3.33), solution de l'équation Diophantienne (3.32)). La fonction de coût quadratique (3.22) devient donc :

$$J = (\hat{Y} - R)^T (\hat{Y} - R) + \lambda \tilde{u}^T \tilde{u} \quad (3.41)$$

où $R \in \mathbb{R}^{N_2 - N_1 + 1}$ est un vecteur contenant les signaux de référence futurs à partir de l'horizon d'initialisation :

$$R = [r(t + N_1 T_e), \dots, r(t + N_2 T_e)]^T \quad (3.42)$$

La valeur minimale de J au sens des moindres carrés est donc obtenue pour $\tilde{u} = \tilde{u}^*$ avec :

$$\tilde{u}^* = (G^T G + \lambda I)^{-1} G^T (R - f_p) = K(R - f_p) \quad (3.43)$$

La matrice $K \in \mathbb{R}^{N_u \times N_2 - N_1 + 1}$ est la matrice de gain optimal du correcteur GPC.

3.2.5 Implémentation du correcteur GPC

L'équation (3.43) permet d'obtenir la valeur présente et les valeurs futures optimales pour les incréments de la commande jusqu'à l'horizon de commande. Mais seule la première valeur $\tilde{u}_1^* = \Delta u(t)^*$ de \tilde{u}^* est utilisée. A l'instant d'échantillonnage suivant $t + T_e$, la commande optimale \tilde{u}^* est recalculée. La commande $u(t)$ est donc générée de la manière suivante :

$$u(t) = u(t - T_e) + \Delta u(t)^* \quad (3.44)$$

avec :

$$\begin{aligned} \tilde{u}_1^* = \Delta u(t)^* &= k_{11} [r(t + N_1 T_e) - (H_{N_1} \Delta u_c(t - T_e) + F_{N_1} y_c(t))] \\ &+ k_{12} [r(t + (N_1 + 1) T_e) - (H_{N_1 + 1} \Delta u_c(t - T_e) + F_{N_1 + 1} y_c(t))] \\ &\vdots \\ &+ k_{1(N_2 - N_1 + 1)} [r(t + N_2 T_e) - (H_{N_2} \Delta u_c(t - T_e) + F_{N_2} y_c(t))] \end{aligned} \quad (3.45)$$

Avec $C \Delta u_c(t - T_e) = \Delta u(t - T_e)$ et $C y_c(t) = y(t)$. Ces termes peuvent être calculés grâce aux équations récursives suivantes :

$$\Delta u_c(t - T_e) = \Delta u(t - T_e) + (1 - C) \Delta u_c(t - T_e) \quad (3.46)$$

$$y_c(t) = y(t) + (1 - C) y_c(t) \quad (3.47)$$

Les k_{1j} sont les éléments de la première ligne de la matrice de gain optimal K . On peut remarquer que si le modèle est invariant, un seul calcul préalable du gain K est nécessaire. Ce calcul peut être fait au moment de l'initialisation de l'asservissement. Le calcul de $u(t)$ est donc une simple équation récursive dont les variables sont les

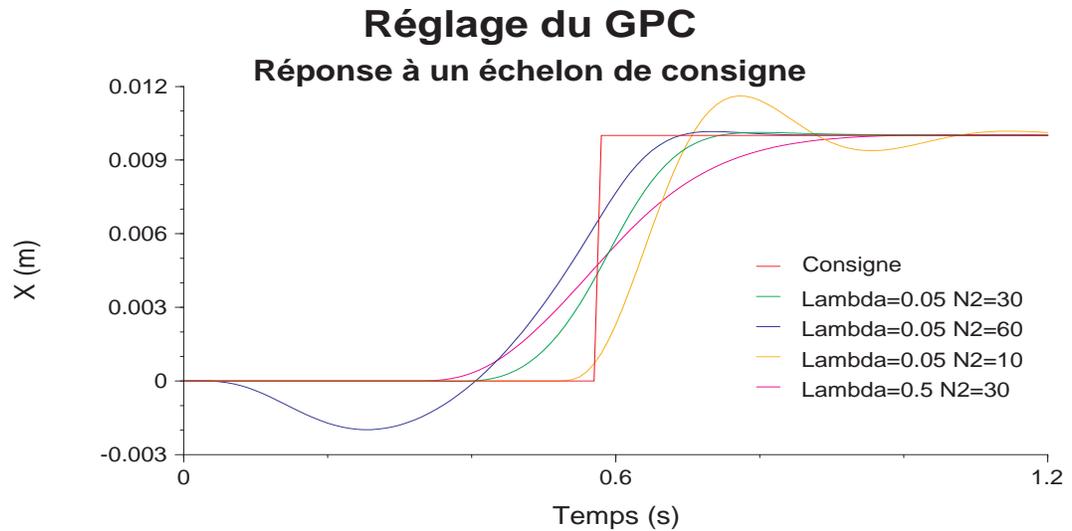


Fig. 3.6: Influence de N_2 et λ sur la réponse indicielle

mesures et les incréments de commande passés filtrés par $\frac{1}{C}$ ainsi que les consignes futures.

Le GPC est également adapté aux systèmes dont le modèle varie avec le temps. Dans ce cas, le gain K doit être réactualisé (à chaque pas d'échantillonnage ou quand cela s'avère nécessaire). Le calcul de K comprend, entre autres, une inversion d'une matrice $N_u \times N_u$ (cf. équation (3.43)). Si on envisage une implémentation temps-réel d'un tel calcul, il est donc nécessaire de limiter l'horizon de commande à une valeur raisonnable.

3.2.6 Réglage du correcteur GPC

La synthèse du correcteur GPC dépend de 4 paramètres de réglage :

- l'horizon d'initialisation N_1 ,
- l'horizon de prédiction N_2 ,
- l'horizon de commande N_u ,
- et la pondération de la commande λ .

Influence de N_1

La modification de l'horizon d'initialisation permet d'ajuster la fenêtre d'optimisation à une zone précise de la réponse. On peut ainsi choisir de ne pas faire intervenir les premiers échantillons dans l'optimisation lorsque, par exemple, le système contient plusieurs retards purs. En pratique, on prendra généralement $N_1 = d + 1$.

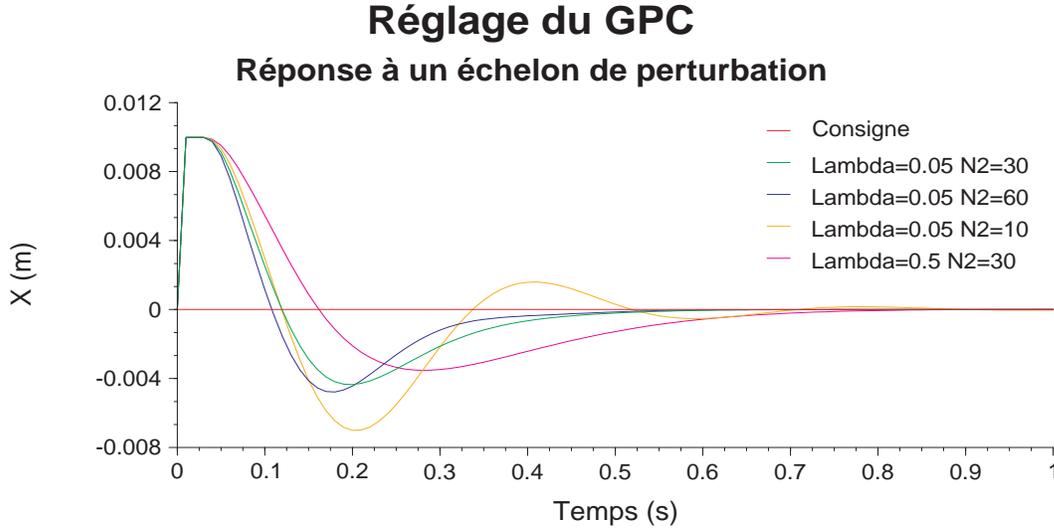


Fig. 3.7: Influence de N_2 et λ sur la réjection d'une perturbation indicielle

Influence de N_u

On choisit en général un horizon de commande N_u relativement réduit. Ce choix est entièrement justifié si le signal de référence de la boucle ne varie pas. Dans ce cas, seules les perturbations, qui par nature sont aléatoires, sont susceptibles de modifier la sortie du système. Il n'est donc pas nécessaire de calculer un nombre important de commandes à l'avance étant donné que seule la première est effectivement appliquée au système et qu'aucune prédiction ne peut être faite sur les perturbations.

Par contre, si le signal de référence n'est pas constant, le fait de ne faire intervenir dans l'optimisation qu'un nombre limité de commandes par rapport à l'horizon de prédiction N_2 a des effets indésirables sur la réponse. Ainsi, lorsque l'échelon commence à entrer dans la zone supérieure de la fenêtre d'optimisation (c'est à dire que l'instant d'occurrence de la prédiction de l'échelon est $t + N_2 T_e$), si $N_u \ll N_2$, il est impossible d'atteindre la référence à l'instant $t + N_2 T_e$ en seulement N_u variations de la consigne puisque $\Delta u(t) = 0$ si $t \geq N_u T_e$. L'erreur quadratique $\sum_{j=N_1}^{N_2} [y(t + jT_e) - r(t + jT_e)]^2$ sera donc d'autant plus grande que l'instant prévu de l'échantillon est éloigné de l'instant $t + N_u T_e$ de la dernière variation de la commande. La minimisation de l'erreur conduit donc à générer des commandes qui sont certes optimales au sens du critère (3.22), mais qui sont très éloignées de la réponse optimale obtenue grâce à un correcteur LQG. Néanmoins, on peut noter que le correcteur GPC tend vers le correcteur LQG lorsque N_2 et N_u tendent vers l'infini.

Les figures 3.6 et 3.7 illustrent ce comportement du GPC. Ainsi, la réponse à un échelon de consigne pour $N_u = 6$ et $N_2 = 30$ est beaucoup plus proche de la réponse avec un horizon infini que celle pour $N_u = 6$ et $N_2 = 60$. Le début de la réponse pour $N_2 = 60$, part dans un premier temps dans le sens opposé au sens de l'échelon. Ce comportement ne se retrouve pas pour $N_2 = 30$.

Il faut donc conserver un horizon de commande N_u en rapport avec l'horizon de prédiction N_2 . Par contre, pour le rejet de perturbation (*cf.* figure 3.7), l'augmentation de l'horizon de prédiction par rapport à l'horizon de commande n'a pas d'influence.

Influence de N_2

On choisit en général N_2 tel que $N_2 T_e$ soit de l'ordre de grandeur de la constante de temps dominante du système en boucle fermée. Ainsi, la fenêtre d'optimisation contient toute la réponse du système. La figure 3.7 illustre l'influence de l'horizon de prédiction. La constante de temps dominante de ce système en boucle fermée est de l'ordre de $30 \times T_e$. Les réponses pour $N_2 = 30$ et $N_2 = 60$ sont pratiquement identiques (celle pour $N_2 = 60$ est légèrement meilleure). Par contre, la réponse se dégrade nettement pour $N_2 = 10$. Dans ce cas, l'optimisation ne se fait que sur le début de la réponse. On peut ainsi constater que cette réponse est superposable à celle pour $N_2 = 30$ dans les premiers instants mais elle se démarque ensuite.

Influence de λ

De tous les paramètres de réglage du GPC, λ est celui dont l'influence est la plus évidente. En effet, il permet de pondérer l'influence des commandes dans l'optimisation et ainsi permet de générer un correcteur plus ou moins «énergique» donc plus ou moins rapide. Les figures 3.7 et 3.6 illustrent cela de manière évidente.

3.2.7 Utilisation du GPC dans la boucle de vision

Le GPC est particulièrement bien adapté à l'asservissement visuel. En effet, comme nous l'avons expliqué dans la section 3.1, le modèle dynamique du dispositif de déplacement opérationnel virtuel varie en fonction de la position q du robot. Aussi, il est souhaitable de réaliser un contrôle de la boucle qui tienne compte de ces dynamiques et qui puisse également s'adapter à leurs variations.

Soit $H(z)$, la fonction de transfert du dispositif de déplacement opérationnel virtuel autour d'une position q . Etant donné que le processus d'acquisition d'image introduit un retard d'une période, la fonction de transfert en boucle ouverte du modèle linéarisé de l'asservissement visuel est donc $z^{-1}H(z)$ (voir figure 3.4 page 47). Le modèle (3.13), page 48, est donc défini de la manière suivante :

$$\frac{B(q^{-1})}{A(q^{-1})} = H_i(q^{-1}) \quad (3.48)$$

où $H_i(q^{-1})$ est la $i^{\text{ème}}$ fonction de transfert de la diagonale de $H(z)$ dans laquelle on a remplacé z^{-1} par l'opérateur de retard q^{-1} .

Le GPC peut être réactualisé à chaque pas d'échantillonnage en fonction du modèle $z^{-1}H(z)$ du système. Il se prête donc parfaitement au contrôle de la boucle de vision.

Le GPC est aussi un correcteur prédictif. Cette caractéristique peut également être mise à profit pour un suivi de trajectoire. Cette trajectoire est définie par une variation de la consigne p^* de la boucle de vision en fonction du temps. La connaissance des N_2 signaux de référence futurs peut donc servir à définir le vecteur R (cf. équation 3.42).

3.3 La plateforme expérimentale

Cette plateforme a servi de base à toutes nos expériences d'asservissements visuels. Elle est construite autour d'un robot manipulateur à 6 degrés de liberté de type SCEMI 6P01. Les asservissements visuels réalisés avec ce système correspondent à la configuration classique de type *eye in hand*.

3.3.1 Schéma général

La figure 3.8 donne un aperçu de la structure du système expérimental. Nous avons choisi une architecture dans laquelle tous les traitements (traitements d'image et contrôle de la boucle) sont réalisés par la même machine afin de limiter au maximum les délais dûs au transfert des données. Nous avons opté de tout regrouper au niveau de l'ordinateur du système d'acquisition d'images. Pour de plus amples détails sur la structure logicielle du système, le lecteur pourra se référer à l'annexe C.1.

3.3.2 Le capteur de vision

Les capteurs visuels utilisés pour les manipulations sont des caméras CCD. Deux types de caméras ont servi aux expériences :

1. Une caméra CCD monochrome au standard CCIR permettant d'acquérir 25 images entrelacées par seconde, c'est à dire 50 trames par seconde. Sa résolution est de 752×582 pixels. L'intégration de la lumière pouvant être effectuée à chaque trame, ce capteur permet de fournir une mesure à la cadence de $50Hz$.
2. Une caméra CCD monochrome rapide pouvant acquérir jusqu'à 240 images non-entrelacées par seconde. Nous n'avons pas exploité cette cadence maximale, car à 240 images/s, la résolution est trop grossière. Nous avons donc limité la cadence à 120 images non entrelacées par seconde avec une résolution de 640×240 pixels.

L'objectif utilisé pour les manipulations a une longueur focale $f = 8.5mm$. Un tel objectif est bien adapté au robot que nous utilisons : Il permet de visualiser entièrement et avec une bonne netteté une cible située à environ $30cm$ de la caméra. Cette distance est compatible avec l'espace de travail du robot qui s'inscrit approximativement dans une demi-sphère de $70cm$ de rayon.

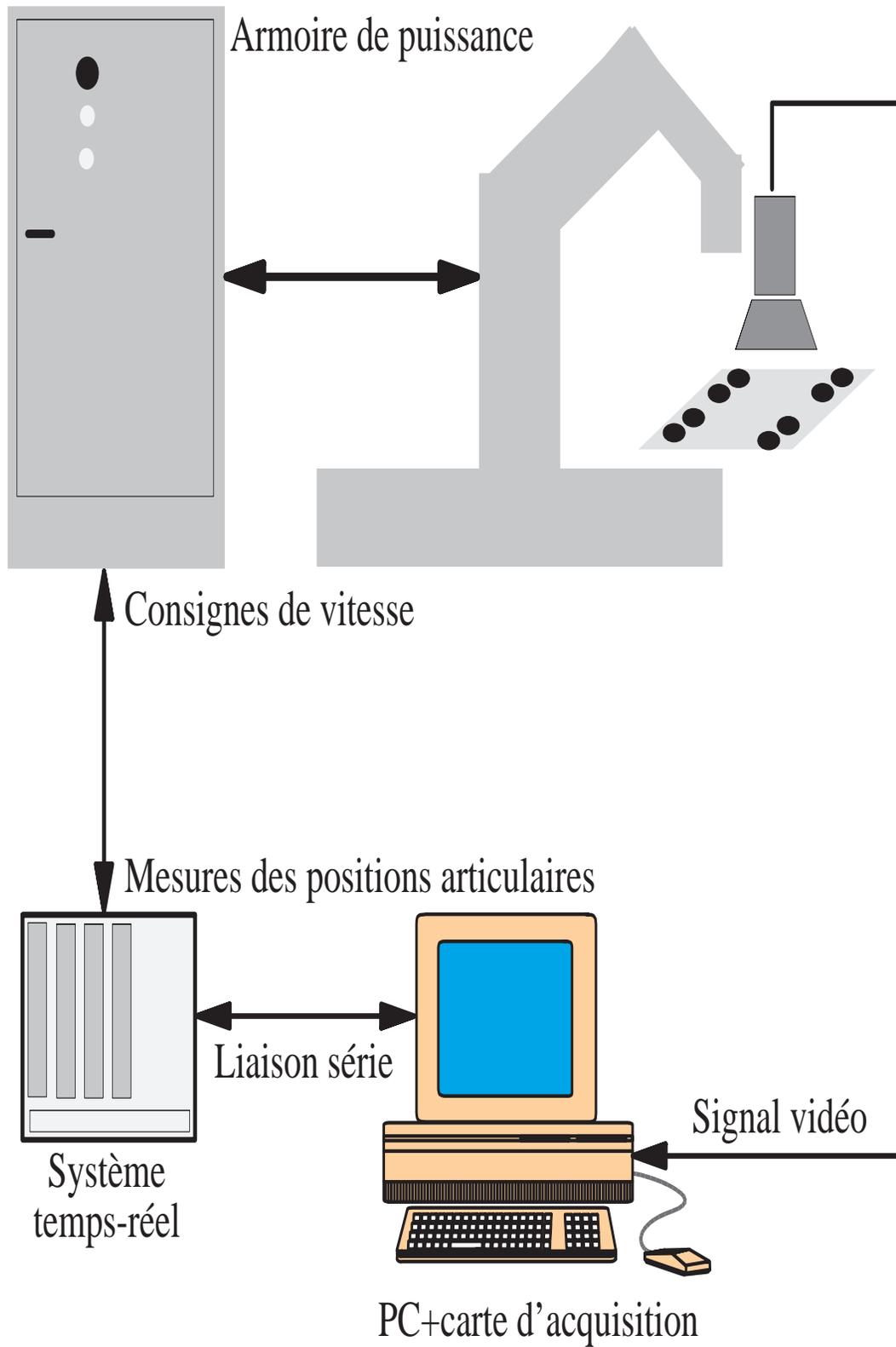


Fig. 3.8: Schéma de la plateforme expérimental

3.3.3 Le système d'acquisition d'images

Le système d'acquisition d'image (que nous appelons aussi *système de vision*) est constitué d'un ordinateur de type «PC» équipé d'une carte d'acquisition au format PCI de marque *Imaging Technologies*. Pour nos expériences, nous avons utilisé deux systèmes de vision. Le premier PC est un *Pentium Pro 200* muni d'une carte de numérisation rapide compatible avec la caméra rapide. Il a servi aux expériences de suivi de cible. Le second est un *Pentium II 300* équipé d'une carte de numérisation standard compatible avec la caméra CCIR 25 images/s qui fut utilisé dans l'expérience de suivi de profilé.

Les deux cartes d'acquisition ont des caractéristiques communes :

1. Le résultat de la numérisation de l'image est stocké dans une mémoire à double entrée intégrée à la carte. Cela signifie qu'il est possible de lire des pixels d'une image en cours d'acquisition. Cette caractéristique très intéressante a une influence sur le modèle dynamique de la caméra (voir section 2.2.4).
2. Le transfert de l'image dans la mémoire centrale du PC s'effectue à un débit de 120 Mo/s. Ce transfert a longtemps été considéré comme un goulet d'étranglement pour la réalisation pratique d'asservissements visuels rapides. Même si le temps de transfert de l'image n'est pas encore négligeable par rapport à la durée d'une période, il n'est plus prohibitif (voir annexe C.1 pour plus de précisions).

3.3.4 Le système temps-réel

Ce système multi-processeurs, multi-tâches temps réel est destiné à réaliser l'interface de bas niveau avec le matériel, en l'occurrence, les boucles analogiques de vitesse du robot. Il contient une carte de conversion numérique/analogique pour le blocage d'ordre 0 des références de vitesse et une carte possédant 6 compteurs pour la mesure des positions articulaires.

Ce système ne réalise aucun asservissement de position. Il sert uniquement d'interface et communique avec le système de vision *via* une liaison série bidirectionnelle. Certaines tâches de sécurité, comme un chien de garde (*watchdog*), lui sont également confiées (voir annexe C.1 pour plus de précisions sur l'architecture logicielle du système).

3.4 Simulation de l'asservissement par vision

La difficulté de mise au point d'un asservissement par vision augmente avec le nombre de degrés de liberté à asservir. Dans le cas où il faut asservir 6 degrés de liberté, une grande rigueur est nécessaire dans la modélisation du système. La simulation est un outil précieux pour déceler d'éventuelles erreurs de modèle ou simplement pour corriger les erreurs de programmation.

3.4.1 Simulation du robot

La simulation du robot tient compte des dynamiques des axes. Ces dynamiques ont été identifiées préalablement sur le SCEMI par la procédure décrite dans la section 2.4.4. Le modèle dynamique simulé correspond à une position nominale du robot (celle qu'a le robot sur la figure 2.9 page 31). En effet, on a vu dans la section 2.4.3 que ce modèle est susceptible de varier, notamment en raison de la variation de l'inertie $D(q)$ (cf. équation (2.33), page 37). Nous avons principalement utilisé la simulation pour tester la réponse à des petits échelons de position de la cible autour de la position nominale du robot. Dans ce contexte, le fait de ne pas simuler la variation des dynamiques n'engendre pas d'erreurs importantes.

En outre, un retard pur est simulé entre l'envoi des commandes et leur application effective sur les convertisseurs numérique-analogique qui sont modélisés par des bloqueurs d'ordre 0. Ce retard est justifié par le délai introduit par le calcul de la commande (voir le chronogramme C.2 page 155). Des «*offsets*» ainsi qu'un bruit blanc sont additionnés aux commandes afin de simuler les imperfections de l'asservissement analogique de vitesse.

3.4.2 Simulation de la caméra

La caméra est modélisée au niveau des primitives. Cela signifie que seules les primitives qui servent à l'asservissement sont simulées. La synthèse de l'image de la cible et le traitement d'image ne sont donc pas réalisés en simulation.

Nous avons voulu obtenir un modèle de caméra le plus proche possible de la réalité. Pour cela, nous avons simulé la distorsion de l'image ainsi que le bruit. Ces paramètres, ainsi que les paramètres de grandissement G_x et G_y ont été identifiés de manière précise sur la caméra par une technique de calibration [Tsa87].

3.4.3 Interface graphique de la simulation

Afin d'avoir un aperçu le plus réaliste possible du comportement du robot en asservissement par vision, nous avons développé une interface graphique visualisant le robot en 3D suivant la technique du «fil de fer» (voir figure 3.9). Une caméra virtuelle peut être déplacée autour du robot afin de visualiser ce dernier suivant différents angles. La simulation se déroule en temps réel, ce qui signifie que le robot simulé évolue à la même vitesse que le robot réel.

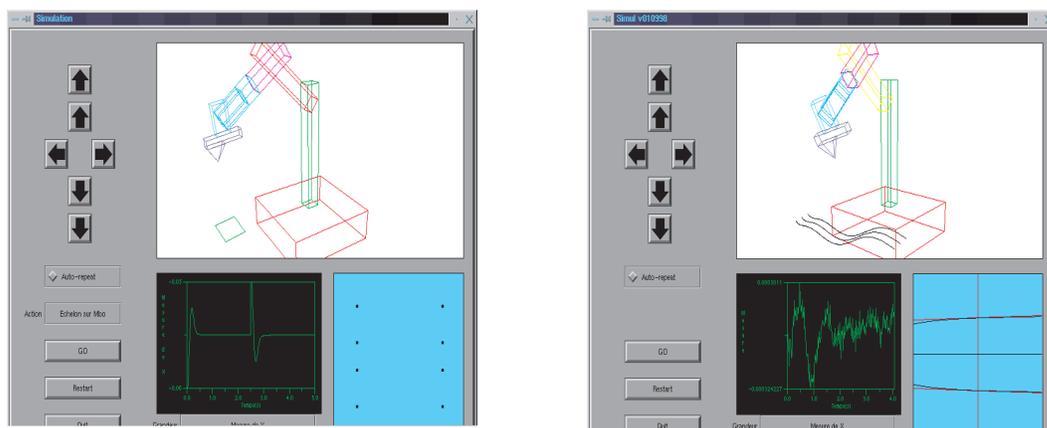


Fig. 3.9: Interface graphique de la simulation

Chapitre 4

Suivi de cible

Dans ce chapitre, nous décrivons une approche unifiée de contrôle de la boucle d'asservissement visuel dans une application de suivi de cible. Nous avons appliqué une stratégie de contrôle commune pour l'asservissement 2D et 3D de la boucle de vision. Celle-ci consiste à convertir la mesure en 6 coordonnées opérationnelles définissant l'attitude du repère attaché à la caméra par rapport à un repère fixe. Grâce au modèle dynamique du dispositif de déplacement opérationnel virtuel décrit dans la section 3.1.2, la synthèse du correcteur peut être réalisée de la même manière en 2D et en 3D.

4.1 Principe du suivi de cible

4.1.1 Définition de la grandeur de mesure de la boucle de vision

Soit R_o , un repère associé à l'objet qui constitue la cible. On définit par M_{co}^* , une transformation homogène de référence désirée entre le repère caméra, R_c , et R_o .

La mesure extraite de l'information visuelle est exprimée sous la forme d'une attitude p contenant 6 coordonnées opérationnelles. Cette attitude est celle d'un repère de référence, R_r , qui se trouve à la position désirée du repère caméra et qui est définie par rapport au repère caméra courant R_c .

Le repère de référence R_r est défini par :

$$M_{ro} = M_{co}^* \quad (4.1)$$

où M_{ro} est la transformation homogène entre R_r et R_o . On peut donc considérer qu'il existe une liaison virtuelle rigide entre R_r et R_o .

L'asservissement visuel consiste à asservir la mesure de p à une consigne p^* . Lorsque p a convergé vers p^* , et que la consigne p^* est nulle, on a $M_{co} = M_{co}^*$ ce qui est équivalent à $M_{cr} = I$.

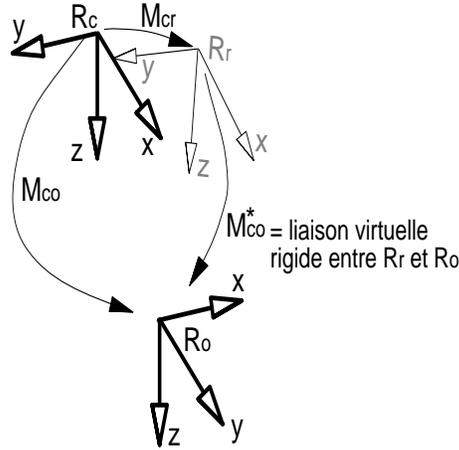


Fig. 4.1: Définition du repère R_r

Lorsque $p^* \neq 0$, l'asservissement se stabilise à une attitude $M_{cr} \neq I$. Grâce à p^* , on peut ainsi générer une trajectoire du repère caméra référencée par rapport à l'objet autour de la position de référence définie par M_{co}^* .

Tout mouvement de la cible entraîne un mouvement de R_r (voir figure 4.1). C'est pourquoi, le mouvement de la cible est modélisé comme une perturbation aléatoire de la mesure.

Dans l'approche 3D, c'est directement M_{co}^* qui définit de la position de référence ; tandis que dans l'approche 2D, c'est l'image de la cible lorsque $M_{co} = M_{co}^*$ qui définit cette référence.

Pour un asservissement 3D, l'estimation de M_{co} grâce à la connaissance du modèle géométrique de la cible permet de calculer p .

Dans le cas d'un asservissement 2D, on n'a aucun moyen de mesurer M_{co} . Néanmoins, grâce au Jacobien image, il est possible d'estimer M_{cr} , grâce à la technique que nous proposons dans ce chapitre.

4.1.2 Commande de la boucle de vision

La stratégie de contrôle de la boucle de vision que nous avons choisie est donc la même pour les asservissements 2D et 3D. Dans les deux cas, la mesure de p est asservie à une consigne p^* .

L'asservissement peut être considéré comme une boucle possédant deux signaux de consigne :

1. La position de référence M_{co}^* définie par une matrice homogène dans le cas 3D ou par une image dans le cas 2D.
2. La consigne p^* qui représente un déplacement par rapport à la position de référence défini par 6 coordonnées opérationnelles que ce soit en 3D comme en 2D.

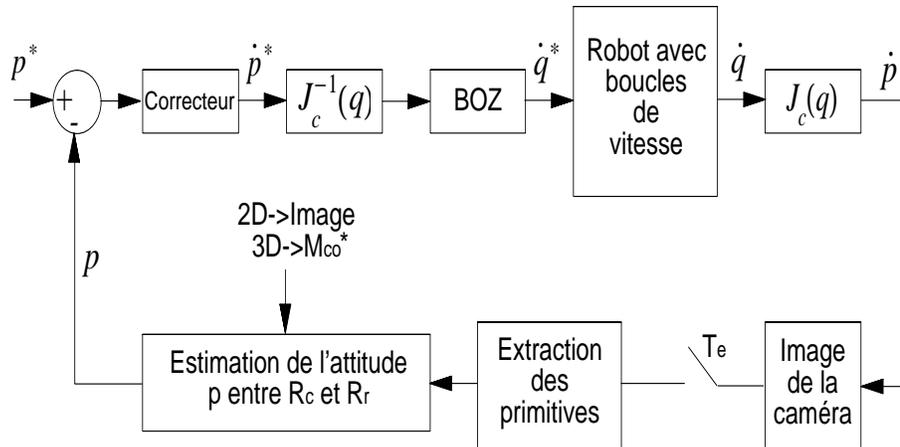


Fig. 4.2: Schéma de la boucle d'asservissement

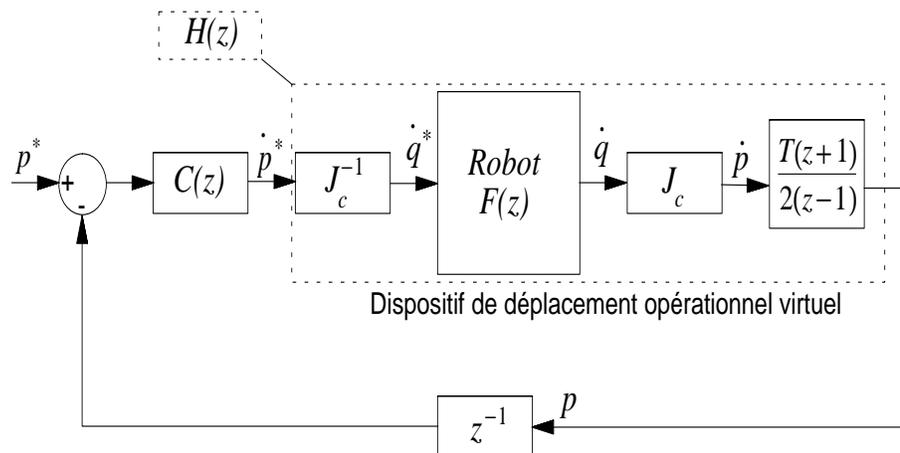


Fig. 4.3: Modèle linéaire de la boucle d'asservissement

La figure 4.2 donne un aperçu global de la boucle d'asservissement visuel d'un suivi de cible. La différence entre un asservissement 2D et un asservissement 3D réside dans la manière de déterminer p . D'après la modélisation dynamique du dispositif de déplacement opérationnel virtuel que nous proposons dans la section 3.1.2, la fonction de transfert $H(z)$ entre p^* et p peut être approximée par une matrice de transfert diagonale. Les dynamiques suivant les 6 coordonnées de p peuvent donc être considérées parfaitement découplées. On a donc décomposé l'asservissement de l'attitude p en 6 boucles indépendantes pour chacune des coordonnées opérationnelles. Le correcteur $C(z)$ est donc également une matrice diagonale contenant les 6 correcteurs des 6 boucles indépendantes. La figure 4.3 présente le modèle linéarisé de l'asservissement. La procédure d'acquisition d'image et d'extraction des primitives a été modélisée par un retard pur z^{-1} conformément au modèle dynamique de la caméra et du traitement d'image établi dans la section 2.2.4, page 27.

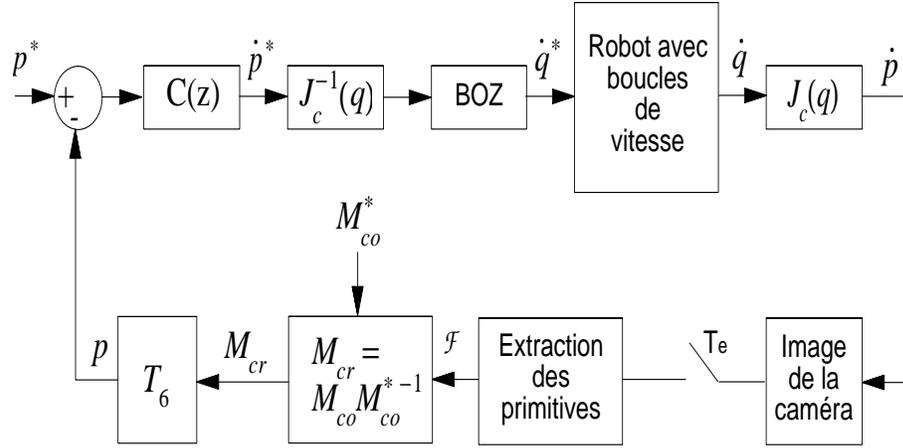


Fig. 4.4: Schéma général de la boucle d'asservissement pour un suivi en 3D

4.1.3 Suivi de cible par asservissement visuel 3D

Dans un asservissement visuel 3D, il est nécessaire de connaître le modèle géométrique de la cible qui doit être suivie. Grâce à ce modèle, et à condition que les primitives extraites de l'image de la cible soient assez riches, il est possible d'estimer la transformation M_{co} (voir annexe B.6, pour un exemple de calcul de M_{co} dans le cas où la cible est constituée de n points coplanaires). L'attitude de référence M_{co}^* entre la caméra et la cible est initialisée au début de l'asservissement.

La mesure p de la boucle de vision correspond à la transformation M_{cr} entre R_c et R_r . On a (puisque $M_{ro} = M_{co}^*$) :

$$M_{cr} = M_{co} M_{co}^{*-1} \quad (4.2)$$

Soit T_6 , l'opérateur transformant une matrice homogène en 6 coordonnées opérationnelles (voir annexe B.3 pour la définition de T_6). Il vient que

$$p = T_6(M_{cr}) \quad (4.3)$$

La figure 4.4 reprend la figure 4.2 dans le cas particulier d'un asservissement 3D.

4.1.4 Suivi de cible par asservissement visuel 2D

Définition du Jacobien image

Soit $\mathcal{F} \in \mathbb{R}^{2n}$, le vecteur des coordonnées courantes de n points dans l'image :

$$\mathcal{F} = \begin{bmatrix} x_{p1} \\ y_{p1} \\ x_{p2} \\ y_{p2} \\ \vdots \\ x_{pn} \\ y_{pn} \end{bmatrix} \quad (4.4)$$

Le Jacobien image $J_v(\mathcal{F}) \in \mathbb{R}^{2n \times 6}$ est une matrice reliant la vitesse $\dot{\mathcal{F}}$ des points dans l'image et le torseur cinématique \dot{r} du repère caméra. Cette relation est valable autour d'une image de référence définie par \mathcal{F} :

$$\dot{\mathcal{F}} = J_v(\mathcal{F})\dot{r} \quad (4.5)$$

Le torseur cinématique est défini par 6 coordonnées exprimée dans R_c : les trois premières sont les coordonnées du vecteur vitesse de déplacement de l'origine du repère et les 3 dernières sont les coordonnées du vecteur vitesse de rotation de ce repère. Les mouvements de translation et de rotation de ce repère sont relatifs à un repère fixe.

$$\dot{r} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{R_c} \quad (4.6)$$

Le Jacobien image est construit de la manière suivante lorsque les primitives sont des points (le calcul a été reporté en annexe B.5) :

$$J_v(\mathcal{F}) = \begin{bmatrix} -\frac{f}{z_1} & 0 & \frac{x_{p1}}{z_1} & \frac{x_{p1}y_{p1}}{f} & -\frac{f^2+x_{p1}^2}{f} & y_{p1} \\ 0 & -\frac{f}{z_1} & \frac{y_{p1}}{z_1} & \frac{f^2+y_{p1}^2}{f} & -\frac{x_{p1}y_{p1}}{f} & -x_{p1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\frac{f}{z_n} & 0 & \frac{x_{pn}}{z_n} & \frac{x_{pn}y_{pn}}{f} & -\frac{f^2+x_{pn}^2}{f} & y_{pn} \\ 0 & -\frac{f}{z_n} & \frac{y_{pn}}{z_n} & \frac{f^2+y_{pn}^2}{f} & -\frac{x_{pn}y_{pn}}{f} & -x_{pn} \end{bmatrix} \quad (4.7)$$

On peut noter que le Jacobien image dépend non seulement de \mathcal{F} mais aussi de la longueur focale f de la caméra et des coordonnées $z_1 \dots z_n$ des points de la cible suivant l'axe z du repère caméra (on parle aussi de profondeur). En fait, seules les

estimations de V_x , V_y et V_z sont dépendantes de la profondeur qui intervient comme un facteur d'échelle. Les vitesses angulaires ω_x , ω_y et ω_z peuvent, quant à elles, être estimées avec les seules informations contenues dans l'image.

Dans le cas d'un suivi de cible, on fait souvent l'approximation $z_i = z_0 \quad \forall i \in [1..n]$, où z_0 est une valeur mesurée une fois pour toute par un autre moyen que la vision. Etant donné qu'on ne peut avoir accès qu'à une estimation des z_i , et de la focale f de la caméra, on notera le Jacobien image correspondant \hat{J}_v pour le différentiel du Jacobien image réel J_v .

Approche 2D classique

On se place dans le cas où la cible est constituée de n points,. Soit $\mathcal{F}_0 \in \mathbb{R}^{2n}$, un vecteur contenant les n coordonnées des positions désirées des points dans l'image. Le vecteur $\Delta\mathcal{F} = (\mathcal{F} - \mathcal{F}_0)$ est la différence entre le vecteur $\mathcal{F} \in \mathbb{R}^{2n}$ des coordonnées courantes des points dans l'image et \mathcal{F}_0 .

Soit $\hat{J}_v^+(\mathcal{F})$, le pseudo-inverse de l'estimée du Jacobien image donné par :

$$\hat{J}_v^+(\mathcal{F}) = (\hat{J}_v^T(\mathcal{F})\hat{J}_v(\mathcal{F}))^{-1}\hat{J}_v^T(\mathcal{F}) \quad (4.8)$$

La loi de commande communément utilisée pour un asservissement 2D est la suivante :

$$\dot{r}^* = -K\hat{J}_v^+(\mathcal{F})\Delta\mathcal{F} - \hat{J}_v^+(\mathcal{F})\hat{\mathcal{F}}_{\downarrow} \quad (4.9)$$

où la matrice $K > 0$ est le gain proportionnel de l'asservissement et $\hat{\mathcal{F}}_{\downarrow}$ est l'estimation de la composante des vitesses des points dans l'image due au mouvement de la cible. Cette commande n'est valable que pour $n \geq 3$ pour un asservissement suivant 6 degrés de liberté. Néanmoins, afin de s'affranchir de certains problèmes de singularité, il est préférable de choisir $n \geq 4$ [MR93].

Le vecteur \dot{r}^* contient les consignes de vitesse pour le torseur cinématique du repère caméra. Ces consignes sont envoyées directement vers le contrôleur du robot. En effet, dans le cas d'un asservissement par vision indirect, c'est celui-ci qui réalise l'asservissement de \dot{r} .

La plupart des auteurs considèrent alors que la dynamique du manipulateur est égale à un transfert unitaire et donc que $\dot{r} = \dot{r}^*$. Avec cette hypothèse, on peut donc écrire que :

$$\dot{r} = -K\hat{J}_v^+(\mathcal{F})\Delta\mathcal{F} - \hat{J}_v^+(\mathcal{F})\hat{\mathcal{F}}_{\downarrow} \quad (4.10)$$

Or :

$$\dot{\mathcal{F}} = J_v(\mathcal{F})\dot{r} + \dot{\mathcal{F}}_m \quad (4.11)$$

où $J_v(\mathcal{F})$ est le Jacobien image réel et $\dot{\mathcal{F}}_m$ la composante réelle de la vitesse des points dans l'image due au déplacement de la cible. D'où :

$$\dot{\mathcal{F}} = J_v(\mathcal{F}) \left[-K\hat{J}_v^+(\mathcal{F})(\mathcal{F} - \mathcal{F}_0) - \hat{J}_v^+(\mathcal{F})\hat{\mathcal{F}}_{\downarrow} \right] + \dot{\mathcal{F}}_m \quad (4.12)$$

Cette loi de commande garantit donc la convergence exponentielle de \mathcal{F} vers \mathcal{F}_0 à condition que $J_v(\mathcal{F})\hat{J}_v^+(\mathcal{F})$ soit strictement positif. On pourra retrouver cette condition de convergence, ainsi qu'une analyse poussée de l'influence des erreurs de modélisation sur l'asservissement par vision dans [Esp95].

De plus, l'erreur statique de poursuite est nulle si $\hat{\mathcal{F}}_{\updownarrow} = \dot{\mathcal{F}}_{\updownarrow}$ et $J_v(\mathcal{F})\hat{J}_v^+(\mathcal{F}) = I$. Une technique d'estimation de $\hat{\mathcal{F}}_{\updownarrow}$ est décrite dans [Ben96].

L'approche 2D que nous proposons

Soit p , l'attitude du repère caméra désiré par rapport au repère caméra courant (voir section 4.1.1). Il existe un Jacobien J_p entre le torseur cinématique \dot{r} et la dérivée par rapport au temps \dot{p} de p (cf. annexe B.4) :

$$\dot{r} = J_p \dot{p} \quad (4.13)$$

où $J_p =$

$$\begin{bmatrix} -1 & 0 & 0 & 0 & T_z \cos \theta_t - T_y \sin \theta_t & -T_z \sin \theta_t \cos \theta_r - T_y \cos \theta_t \cos \theta_r \\ 0 & -1 & 0 & -T_z & T_x \sin \theta_t & -T_z \sin \theta_r + T_x \cos \theta_t \cos \theta_r \\ 0 & 0 & -1 & T_y & -T_x \cos \theta_t & T_y \sin \theta_r + T_x \sin \theta_t \cos \theta_r \\ 0 & 0 & 0 & -1 & 0 & -\sin \theta_r \\ 0 & 0 & 0 & 0 & -\cos \theta_t & \sin \theta_t \cos \theta_r \\ 0 & 0 & 0 & 0 & -\sin \theta_t & -\cos \theta_t \cos \theta_r \end{bmatrix} \quad (4.14)$$

et

$$p = \begin{bmatrix} T_x \\ T_y \\ T_z \\ \theta_t \\ \theta_r \\ \theta_l \end{bmatrix}_{R_c} \quad (4.15)$$

On peut remarquer que :

$$\lim_{p \rightarrow 0} J_p = -I \quad (4.16)$$

Si on considère que p^* est contenu dans une région proche de 0 et que l'asservissement est assez rapide pour maintenir p voisin de p^* en cas de déplacement de la cible, on peut faire l'approximation suivante :

$$\dot{p} \approx -\dot{r} \quad (4.17)$$

Soit J_v^+ , le pseudo-inverse du Jacobien image $J_v(\mathcal{F})$:

$$J_v^+ = (J_v^T J_v)^{-1} J_v^T \quad (4.18)$$

On a donc :

$$\dot{r} = J_v^+ \dot{\mathcal{F}} \quad (4.19)$$

En utilisant (4.19), l'équation (4.17) devient :

$$\dot{p} \approx -J_v^+ \dot{\mathcal{F}} \quad (4.20)$$

Soit \mathcal{F}_0 , le vecteur des paramètres des primitives correspondant à l'image de référence. Lorsque $\mathcal{F} = \mathcal{F}_0$, on a $p = p_0 = 0$, puisque à ce moment là $R_c = R_r$. En réalisant un développement au premier ordre de la relation entre p et \mathcal{F} autour de \mathcal{F} et en utilisant (4.20), nous obtenons :

$$p - p_0 = -J_v^+(\mathcal{F} - \mathcal{F}_0) + \mathcal{O}(\mathcal{F}^2) \quad (4.21)$$

où $\mathcal{O}(\mathcal{F}^2)$ est un terme d'erreur du second ordre. Finalement, nous obtenons la relation approchée entre p et \mathcal{F} suivante :

$$p \approx -J_v^+(\mathcal{F} - \mathcal{F}_0) \quad (4.22)$$

Grâce à cette relation, on peut convertir les mesures exprimées en terme de primitives, \mathcal{F} , en 6 coordonnées opérationnelles, p .

Dans la plupart des applications d'asservissements visuels 2D, le contrôle est réalisé exclusivement dans le plan image. Les grandeurs asservies sont liés aux primitives visuelles. Les paramètres définissant ces primitives constituent les états du système. Avec une telle approche, tous les états sont rarement commandables. Aussi, certains auteurs ont opté pour une décomposition du système en états commandables et non-commandables (voir [HEK96]). Cette solution présente l'inconvénient majeur que les états de l'asservissement ont peu de lien avec le système physique.

Dans notre approche, nous avons choisi de convertir ces états en 6 paramètres opérationnels de manière à rendre le système entièrement commandable. Ces 6 paramètres ont un rapport direct avec le système physique : ils représentent l'attitude p du repère caméra par rapport à un repère fixe, R_r . Aussi, grâce à la modélisation dynamique du dispositif de déplacement opérationnel virtuel qui réalise la commande en vitesse de p (voir section 3.1.2, page 44), on parvient à un découplage total des 6 boucles contrôlant les 6 coordonnées opérationnelles. La figure 4.5 reprend la figure 4.2 dans le cas particulier de l'asservissement 2D que nous proposons.

4.2 Simulations et expériences

Pour les expériences, nous utilisons la plateforme décrite dans la section 3.3. Tous les essais sont réalisés avec une caméra rapide qui acquiert 120 images non entrelacées par seconde. La boucle de vision est synchrone avec l'acquisition d'image ; sa période d'échantillonnage est donc de 8.33 ms. La figure 4.6 donne un aperçu de la manipulation de suivi de cible. Elle définit également la position des axes du

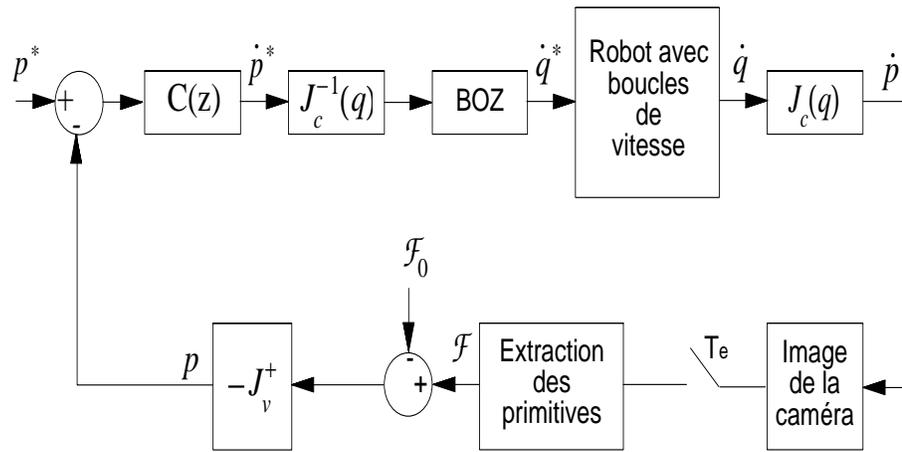


Fig. 4.5: Schéma général de la boucle d'asservissement pour un suivi en 2D

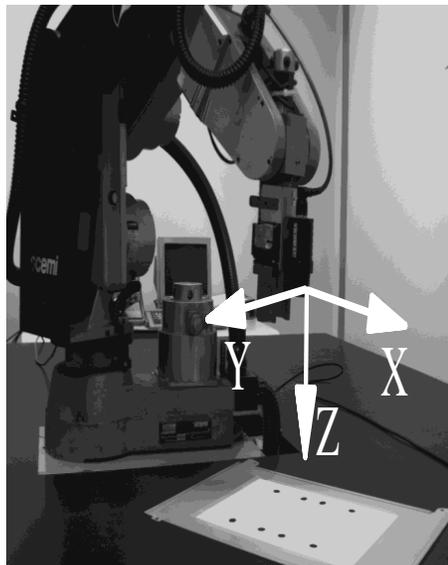


Fig. 4.6: Présentation de la manipulation de suivi de cible



Fig. 4.7: Description de la cible

repère caméra R_c . L'attitude de référence M_{co}^* est celle représentée sur cette figure : la caméra est approximativement à 30cm de la cible.

Nous comparons dans cette section les performances dynamiques de la boucle de vision contrôlée par un correcteur à action Proportionnelle-Intégrale (PI) classique à celles obtenues avec un correcteur GPC. Le correcteur PI a été réglé de manière empirique afin de réaliser un compromis entre la stabilité et les performances.

4.2.1 Description de la cible

La figure 4.7 représente à l'échelle 1/2 la cible qui a été utilisée lors des expériences. Elle est constituée de 8 disques noirs coplanaires imprimés sur une feuille blanche de format A4. Cette feuille est elle-même placée sur un support plan rigide de manière à garantir sa parfaite planéité.

Quatre est le minimum de points requis pour estimer de manière unique et sans singularité la mesure p , que ce soit en 2D ou en 3D [MR93, Yua89]. Le fait d'utiliser plus de points améliore sensiblement la qualité de la mesure en diminuant les effets du bruit sur les pixels.

4.2.2 Traitements d'image

Les primitives extraites de l'image sont des points : il s'agit des centres des projections des disques dans l'image. Ces projections forment 8 taches noires dans l'image. On peut faire l'approximation que les centres des disques de la cible se projettent sur les centres des taches dans l'image. Comme le contour d'un disque est un cercle, sa projection dans le plan image est une ellipse. Le centre d'une ellipse s'obtient en calculant le barycentre de tous ses points.

Des méthodes plus précises d'estimation du centre de l'ellipse peuvent être mises en œuvre [Léo90]. Nous ne les avons pas implémentées afin de minimiser le temps de traitement de l'image.

Le traitement d'image s'effectue donc en 2 temps :

1. Détection des points de contour des 8 taches dans l'image par filtrage passe-haut et seuillage.
2. Calcul des 8 barycentres des points de contours des 8 taches dans l'image.

Le filtrage passe-haut résulte du produit de convolution de l'image avec un masque 5×5 . Ce dernier contient les premiers échantillons de la réponse impulsionnelle d'un filtre de Canny [Can86]. Ce filtre a été choisi car il permet de détecter les contours de manière optimale.

L'image est décomposée en 8 imagettes centrées autour des projections des 8 disques. Seuls les pixels de ces imagettes sont transférés dans la mémoire centrale de l'ordinateur hôte de la carte d'acquisition. Le traitement d'image est également limité à ces imagettes. Cette décomposition de l'image en zones d'intérêt permet de gagner un temps considérable dans le traitement de l'image.

La position des taches dans l'image est susceptible de varier lorsque la cible se déplace. Il faut donc constamment réactualiser la positions des imagettes. Aucune prédiction du mouvement de la cible n'est réalisée : la position de l'imagette dans la nouvelle image est définie de manière à être centrée autour de la projection du disque dans l'image précédente (voir section 1.2.1, page 12, pour plus de précisions). Néanmoins, dans notre cas, la cadence d'acquisition d'image est assez grande pour considérer qu'une tache ne sort jamais de son imagette, étant données des vitesses et des accélérations réalistes pour la cible.

4.2.3 Influence du bruit sur les pixels : comparaison des approches 2D et 3D

La figure 4.8 présente les enregistrements comparés des 6 coordonnées de p pour un asservissement stationnaire¹ 2D et 3D. Un bruit blanc de variance 0.25 pixels^2 est rajouté aux coordonnées des 8 primitives ponctuelles. La boucle est contrôlée par un correcteur GPC dont les paramètres de réglage sont les suivants : $\lambda = 0.02$, $N_1 = 1$, $N_2 = 30$ et $N_u = 6$. Au vu de ces enregistrements, il apparaît clairement que la stratégie 2D est moins sensible au bruit que la stratégie 3D et ce particulièrement pour T_x , T_z et l'angle de roulis θ_r .

Ces résultats de simulation sont confirmés par l'expérience. Nous avons en effet constaté que, pour un même réglage du correcteur GPC, l'asservissement 2D est parfaitement stable, tandis que l'asservissement 3D est oscillant voir divergeant. Afin de mieux cerner ce problème, nous avons réalisé un enregistrement de la mesure p en boucle ouverte et comparé la variance de cette grandeur en 2D et en 3D. Le tableau 4.1 récapitule les résultats obtenus. On retrouve dans ces résultats la meilleure réjection du bruit en 2D suivant T_x , T_z et θ_r . Seule la coordonnée θ_l a une variance significativement plus faible en 3D. Or, lorsque le système est asservi, il oscille justement suivant T_x , T_z et θ_r .

¹ la cible reste fixe et la référence p^* est constante.

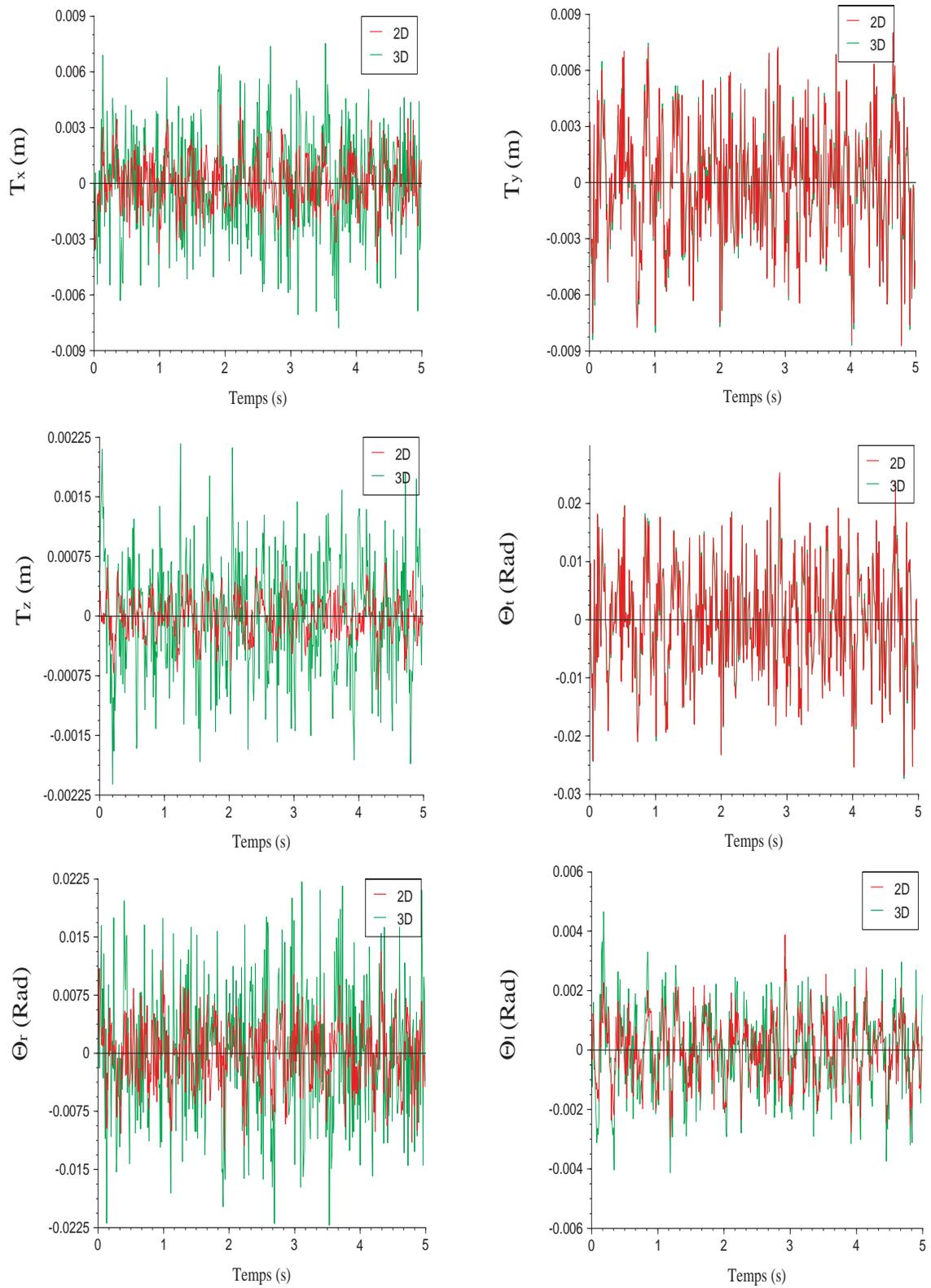


Fig. 4.8: Simulation de l'effet du bruit sur les pixels pour un asservissement stationnaire

Coordonnée	2D	3D
T_x	6.86e-9	1.2e-8
T_y	1.45e-8	1.28e-8
T_z	1.35e-10	6.80e-10
θ_t	1.12e-7	9.95e-8
θ_r	5.72e-8	1.05e-7
θ_l	2.50e-9	1.42e-9

Tab. 4.1: Comparaison de la variance du bruit sur p en boucle ouverte

On peut donc en conclure, que l'estimation de p est plus robuste au bruit avec une technique 2D qu'avec une technique 3D.

Afin de stabiliser l'asservissement 3D, il est nécessaire d'augmenter la pondération de la commande λ ce qui a pour effet de ralentir le système (voir section 3.2.6). Or notre objectif est au contraire la recherche des performances maximales. L'étude qui suit, porte donc exclusivement sur l'analyse des performances de la boucle en 2D.

4.2.4 Réponse de l'asservissement à un échelon de déplacement de la cible

Un échelon de déplacement de la cible n'est pas réalisable pratiquement car il impliquerait une accélération infinie de celle-ci. Néanmoins, en additionnant de manière logicielle un échelon à une des composantes de la mesure p , il est possible de simuler un déplacement instantané de la cible suivant la direction correspondante. Le recours à cet artifice n'est cependant valable que pour des petits échelons. Nous avons donc limité leur amplitude à 1cm pour les translations et 0.03 Rad pour les rotations.

Comparaison entre la simulation et l'expérience

La figure 4.9 présente la superposition des réponses simulées et expérimentales à un échelon suivant T_x de la boucle corrigée par un PI et un GPC. On constate que la réponse expérimentale du PI est assez proche de la simulation. Ce n'est pas le cas avec le GPC : la réponse réelle présente un dépassement nettement plus important que la réponse simulée. Son temps de montée est également plus important. Ce comportement est caractéristique des effets des saturations. En effet, outre le fait de ralentir la réponse, les saturations provoquent également une augmentation excessive du terme intégral du correcteur ce qui conduit à un dépassement important. Or les saturations n'ont pas été intégrées à la simulation, il est donc normal qu'on observe une telle différence.

Ce phénomène ne s'observe pas sur la réponse du PI car ce dernier est beaucoup plus lent que le GPC et ne provoque donc aucune saturation des commandes.

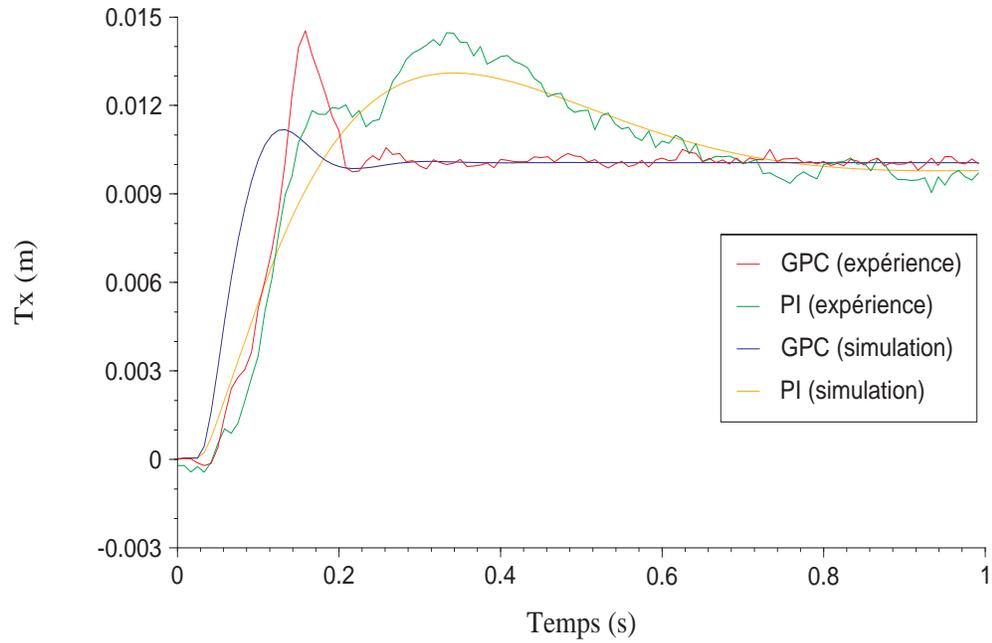


Fig. 4.9: Comparaison entre la simulation et l'expérience pour un échelon suivant T_x

Néanmoins, malgré la saturation, le temps d'établissement du GPC est pratiquement 4 fois plus faible que celui du PI (0.2s *vs.* 0.75s) ce qui confère au système des performances dynamiques nettement supérieures.

Comparaison des réponses à un échelon entre un PI et un GPC suivant les 6 degrés de liberté

Les figures 4.10 et 4.11 présentent les réponses expérimentales à un échelon de position de la cible suivant les 6 coordonnées de p . Nous avons superposé la réponse de la boucle contrôlée par un PI à celle de la boucle contrôlée par un GPC. Pour tous les 6 degrés de liberté T_x , T_y , T_z , θ_t , θ_r et θ_l , le correcteur GPC affiche des performances dynamiques nettement supérieures. Son temps d'établissement est environ 3 fois plus faible que celui du PI (0.25s en moyenne, contre 0.8s pour le PI) tandis que son temps de montée est toujours inférieure à celui du PI (0.12s en moyenne, contre 0.15s en moyenne pour le PI).

Ces résultats étaient prévisibles étant donné que le GPC est optimisé en fonction d'un modèle dynamique de la boucle de vision autour du point où sont réalisés les échelons.

Etude de l'influence de la variation du modèle dynamique sur la réponse indicielle

Le correcteur GPC est calculé en fonction du modèle dynamique du dispositif de déplacement opérationnel virtuel linéarisé autour de la position q du robot. Il

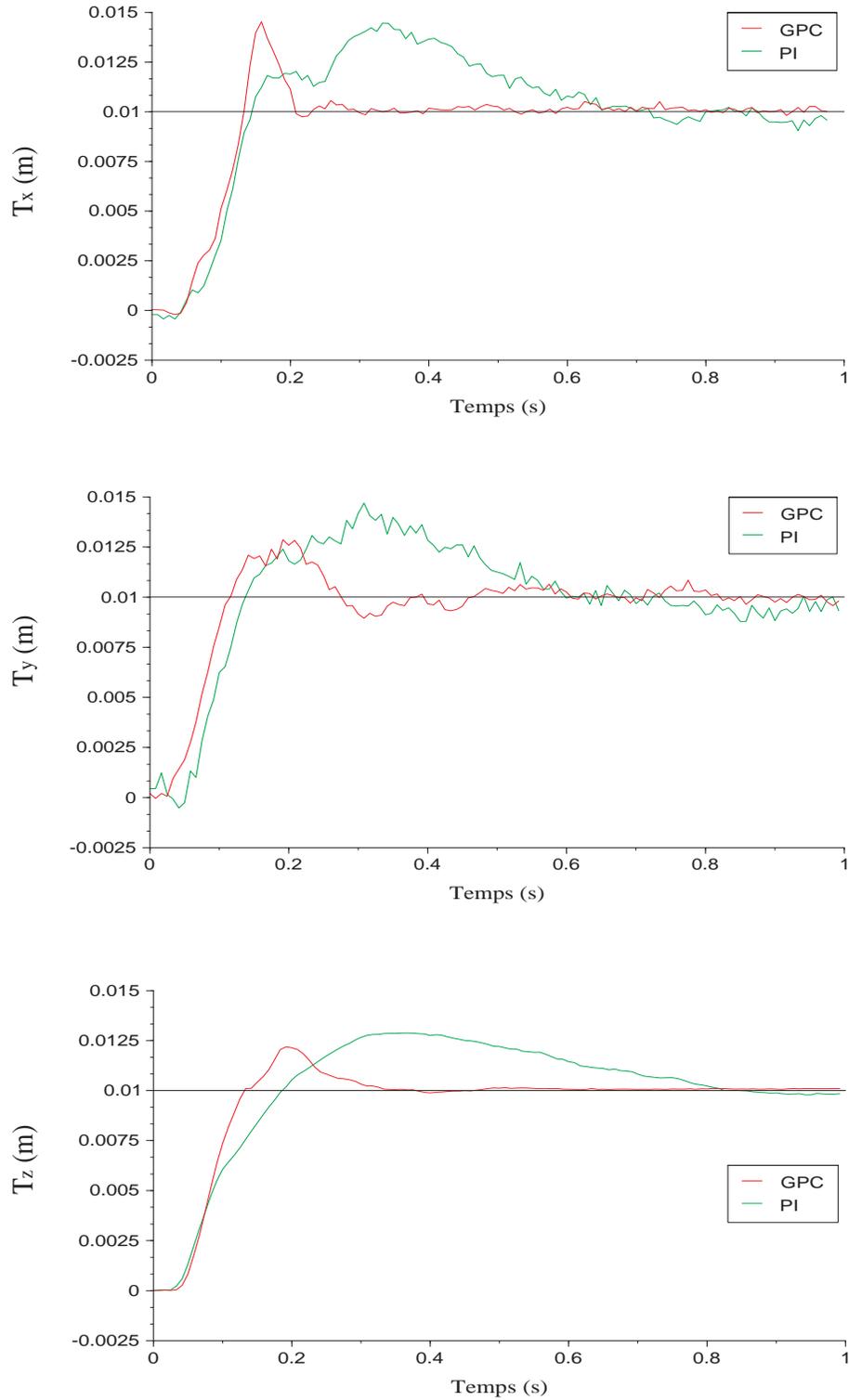


Fig. 4.10: Comparaison des réponses à des échelons de déplacement de la cible suivant T_x , T_y et T_z .

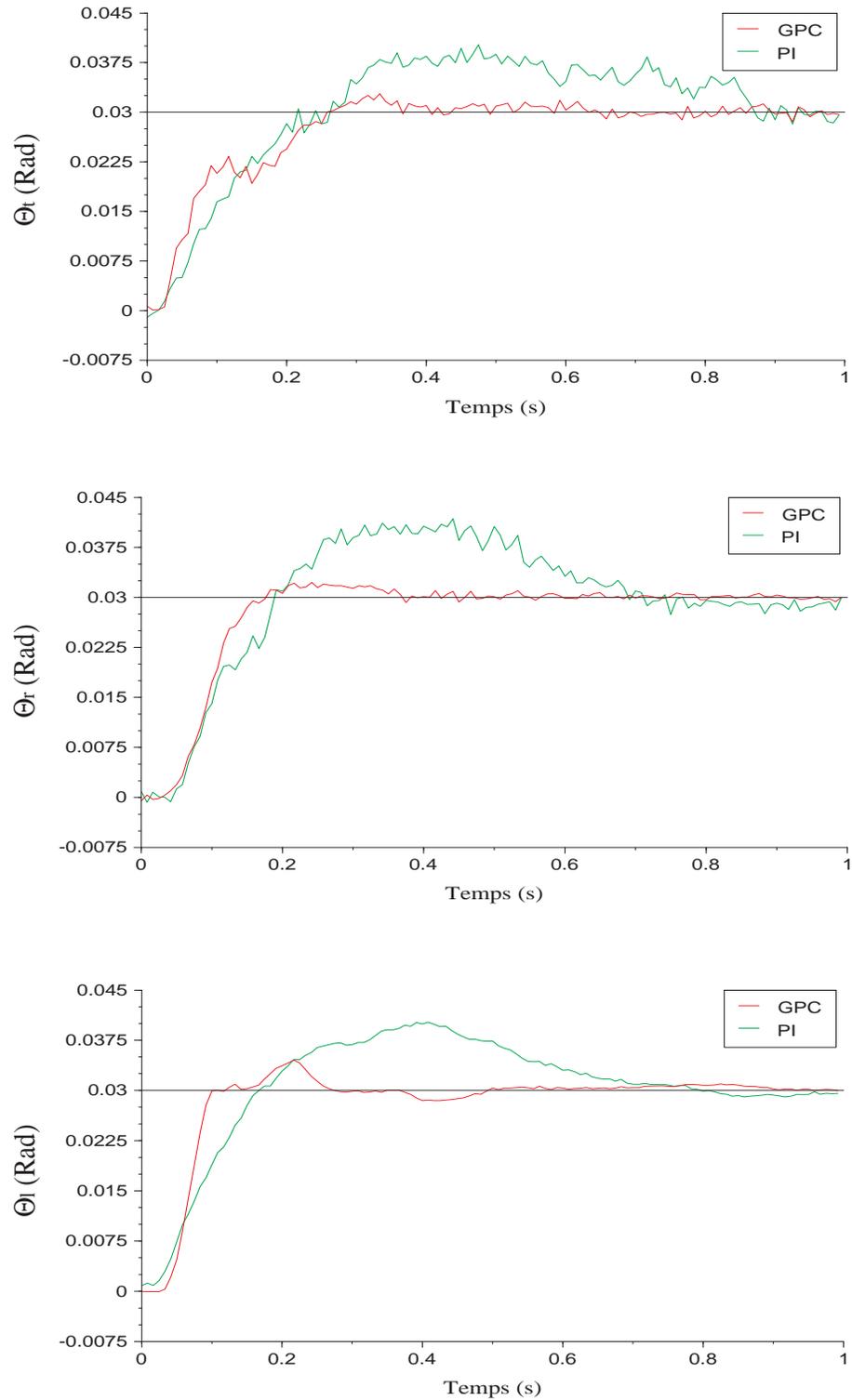


Fig. 4.11: Comparaison des réponses à des échelons de déplacement de la cible suivant θ_t , θ_r et θ_l .

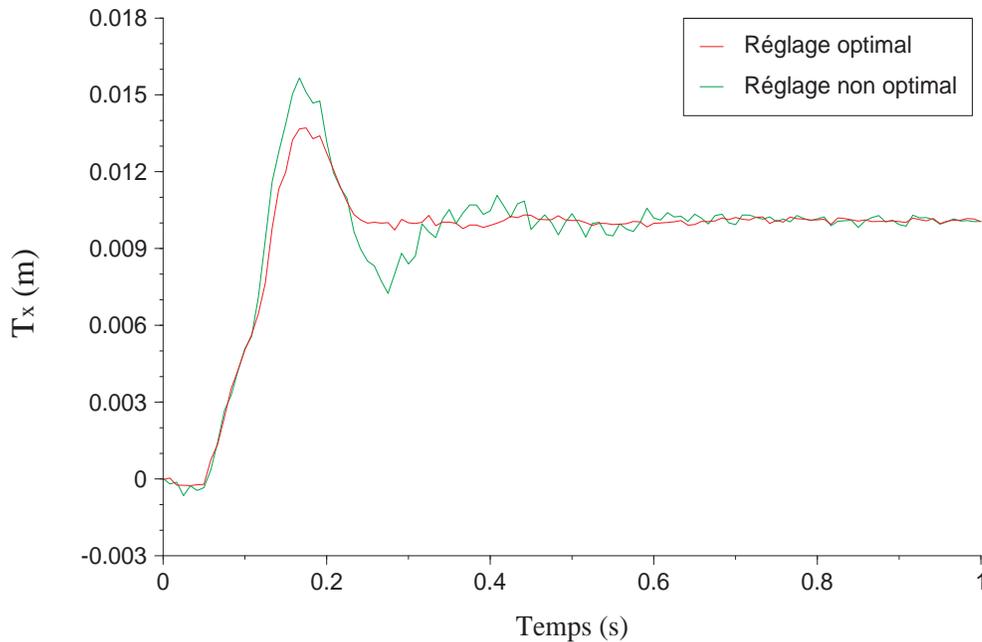


Fig. 4.12: Comparaison des réponses suivant T_x d'un correcteur GPC optimal et d'un correcteur GPC non optimal.

réalise donc un contrôle optimum en q . Dans cet essai, le GPC a été calculé pour une position q' du robot et un essai indiciel suivant T_x a été réalisé autour d'une position $q \neq q'$. La position q correspond à la configuration du le robot définie par la figure 4.6. La position q' diffère de la position q d'une rotation de 45 degrés de l'axe 6.

Il est évident, que la dynamique suivant T_x ne peut être identique pour les 2 configurations. Le GPC n'étant pas optimum, on peut s'attendre à une détérioration de la réponse par rapport au cas où le GPC est optimum. La figure 4.12 illustre clairement ce phénomène.

Cet essai est très important car il justifie l'utilisation d'un correcteur adaptatif comme le GPC qui puisse être réactualisé en fonction des variations du modèle dynamique. Dans nos essais, le correcteur n'est réactualisé qu'à l'initialisation de l'asservissement. Etant donné que les essais sont réalisés autour de la configuration initiale, on peut considérer qu'il reste optimum. Néanmoins, si le robot est amené à réaliser de grands déplacements, une réactualisation périodique du GPC peut s'avérer nécessaire. Celle-ci n'a pas été implémentée expérimentalement vu les contraintes en temps de calcul.

La figure 4.12 montre une détérioration de la réponse qui ne va cependant pas jusqu'à l'instabilité de l'asservissement. En effet, le GPC est relativement robuste aux erreurs de modélisation. Aussi, même dans le cas de grands déplacements de la cible (50 cm en translation , 45 degrés en rotation, par exemple) l'asservissement reste parfaitement stable.

Etude des couplages entre les coordonnées opérationnelles

Dans la modélisation du dispositif de déplacement opérationnel, on a négligé les termes de couplage $C_{ik}(z)$ de la matrice $G(z)$ en considérant qu'elle est diagonale (voir section 3.1.2, page 44). Les résultats donnés par les figures 4.13 et 4.14 montrent l'influence de ces termes. Ainsi, avec le correcteur GPC, un échelon de 1 cm suivant T_x provoque une perturbation rejetée en 0.3s sur T_z dont l'amplitude maximale atteint 4.5 mm. Ce même échelon provoque une perturbation de 0.02 Rad suivant l'angle de roulis également rejetée en 0.3s. On peut constater que le temps de réjection est du même ordre de grandeur que le temps d'établissement de l'échelon correspondant. Toujours pour le GPC, un échelon de 0.03 Rad suivant θ_r provoque une perturbation de 4 mm d'amplitude maximale sur T_x et une perturbation de 1.25 mm d'amplitude maximale sur T_z .

Le correcteur PI réalise une réjection systématiquement plus lente que le GPC (environ 3 fois plus lente) même si l'amplitude maximale de la perturbation n'est pas aussi importante.

Cet essai ouvre des perspectives pour l'amélioration du contrôle dynamique de l'asservissement : la prise en compte des termes de couplage $C_{ik}(z)$ dans le modèle dynamique peut être envisagée dans le cadre d'un asservissement GPC multivariable.

4.2.5 Etude de la réponse à une trajectoire

Réponse à une trajectoire de la cible

La figure 4.15 présente les réponses à un déplacement de la cible suivant selon une rampe de pente 5 cm/s en translation et 0.2 rad/s en rotation. Les enregistrements sont réalisés suivant les 6 degrés de liberté en comparant les réponses du PI à celles du GPC.

Dans cette expérience, la cible n'est pas physiquement déplacée. Tout comme pour les essais de réponse à un échelon, le déplacement de la cible est simulé en additionnant une rampe à la mesure. Aussi, l'enregistrement de la figure 4.15 représente l'addition de ce signal et de la réponse du système à la perturbation que représente le mouvement simulé de la cible.

Comme la classe du système est 2 (2 intégrateurs dans la boucle ouverte), l'erreur de poursuite doit être nulle. La figure montre que les réponses de la boucle contrôlée par un GPC et celles de la boucle contrôlée par un PI convergent vers la consigne triangulaire.

De plus, ces enregistrements exhibent à nouveau la supériorité du correcteur GPC sur le correcteur PI : pour tous les degrés de liberté, l'erreur quadratique de poursuite est nettement plus faible avec le GPC.

Réponse à une référence triangulaire

La figure 4.16 présente la réponse à une consigne triangulaire de T_x . C'est donc p^* qui varie et la cible qui reste fixe. La trajectoire est donc connue d'avance. Le

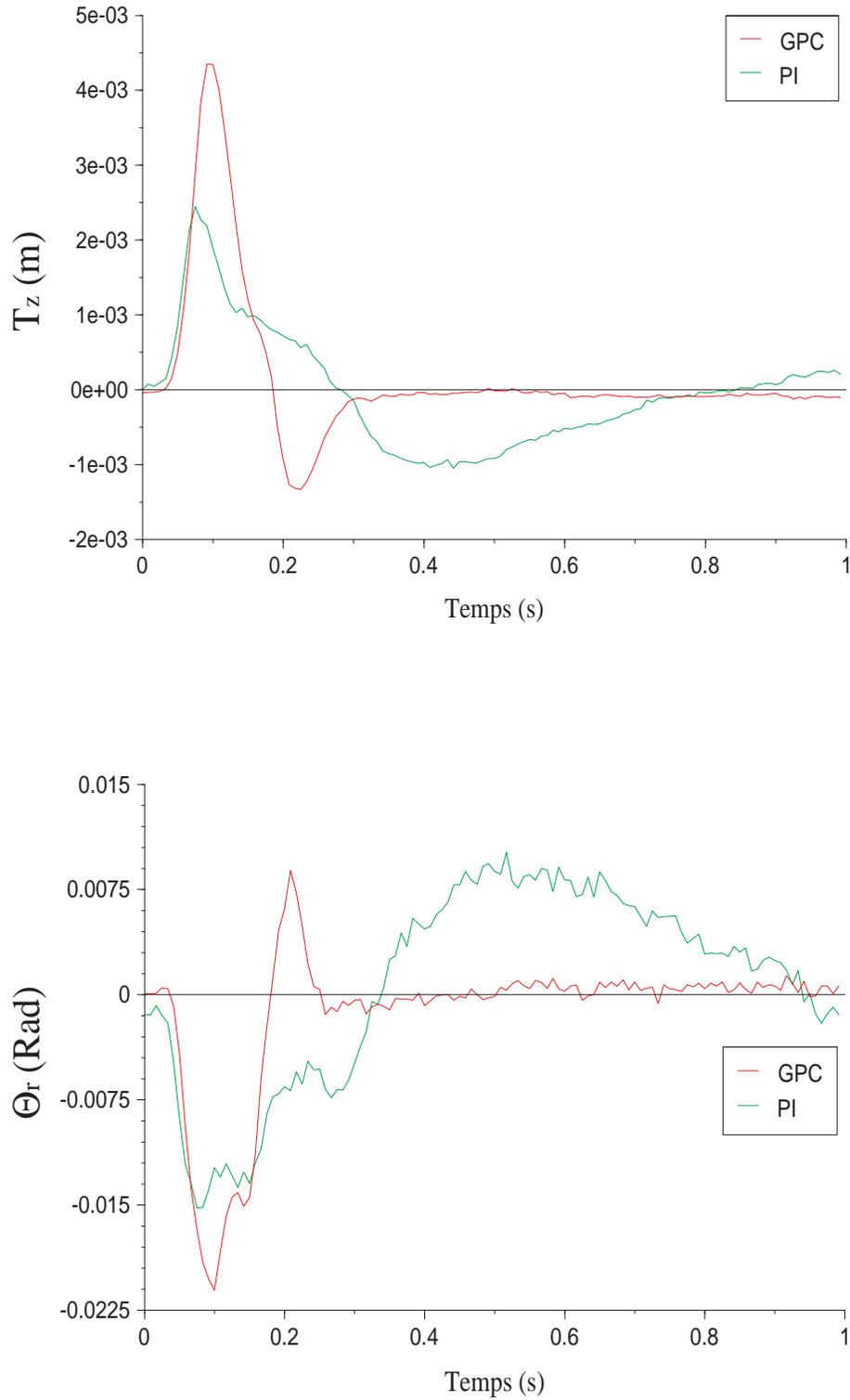


Fig. 4.13: Etude de l'influence d'un échelon de 1 cm suivant T_x sur les asservissements de T_z et θ_r

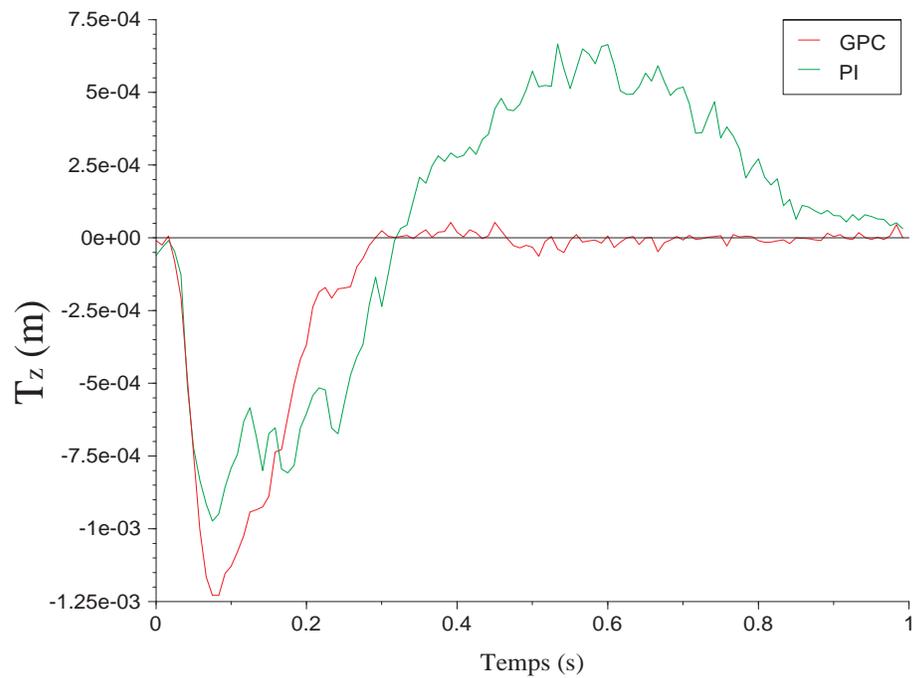
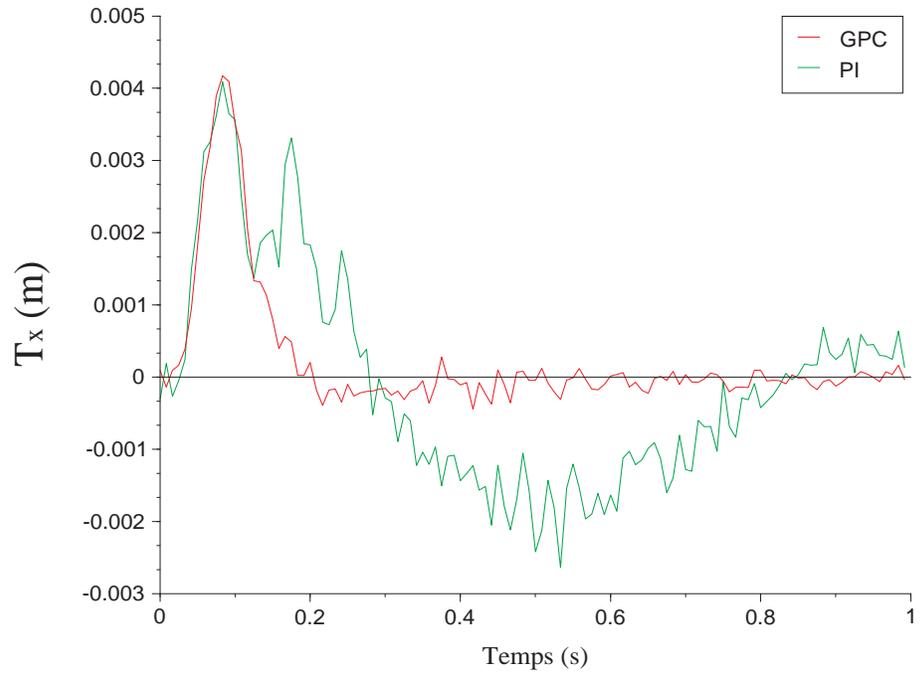


Fig. 4.14: Etude de l'influence d'un échelon de 0.03 Rad suivant θ_r sur les asservissements de T_x et T_z

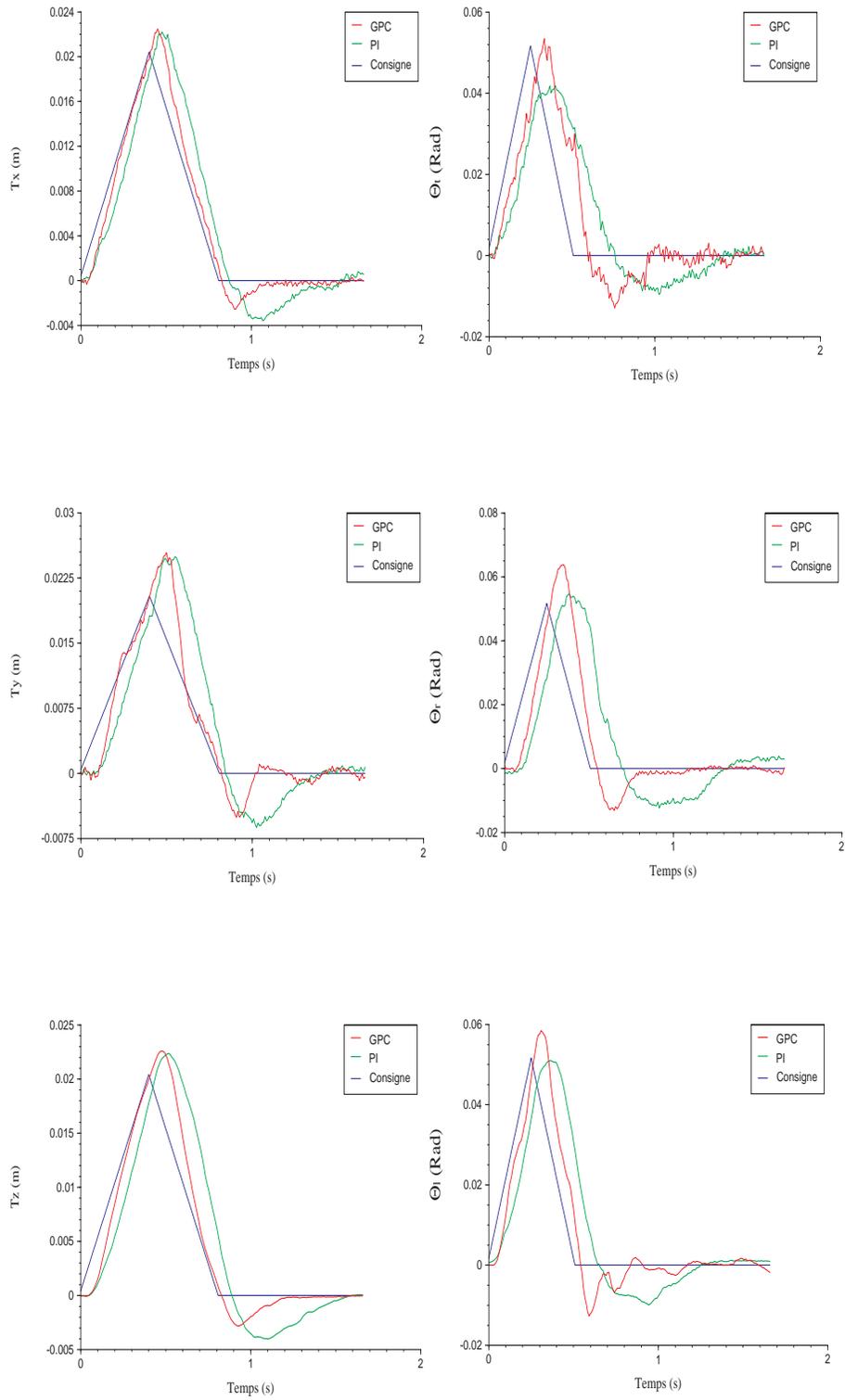


Fig. 4.15: Réponses à un déplacement de la cible à vitesse constante

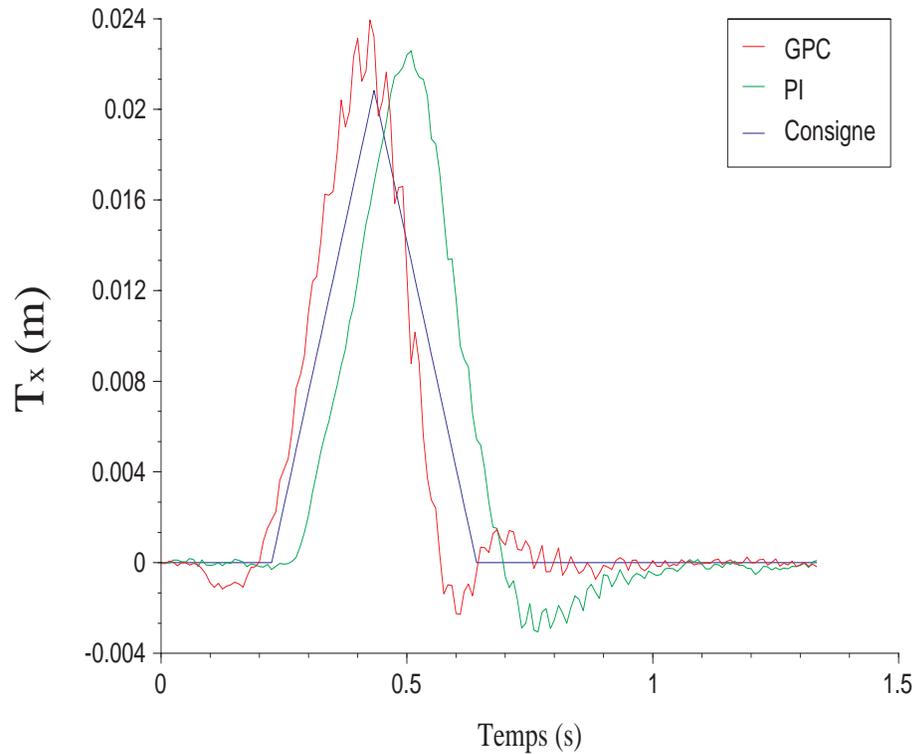


Fig. 4.16: Réponses à un signal de référence triangulaire

GPC étant un correcteur prédictif, les N_2 valeurs futures du signal de consigne échantillonné lui sont fournies (voir équation 3.42, page 53). Cette figure montre de manière évidente que la réponse du GPC est celle d'un correcteur prédictif : la mesure de T_x commence à changer avant que le signal de consigne n'ait commencé à évoluer. Le signal de référence a une pente de 20 cm/s. La figure 4.16 met clairement en évidence la supériorité du GPC par rapport au PI.

Trajectoire dans un plan parallèle au plan image

Dans cet essai, la référence p^* est définie de manière à décrire une trajectoire à la vitesse constante de 5 cm/s dans un plan parallèle au plan image (le long de T_x et T_y).

Les prédictions des signaux de consigne de T_x et T_y sont fournies au correcteur GPC. La figure 4.17 présente l'évolution de T_y en fonction de T_x . La trajectoire est un carré de 4 cm de côté. Le sens dans lequel la trajectoire est décrite est indiqué sur cette figure. On constate à nouveau que le GPC réalise un bien meilleur suivi de la consigne que le PI.

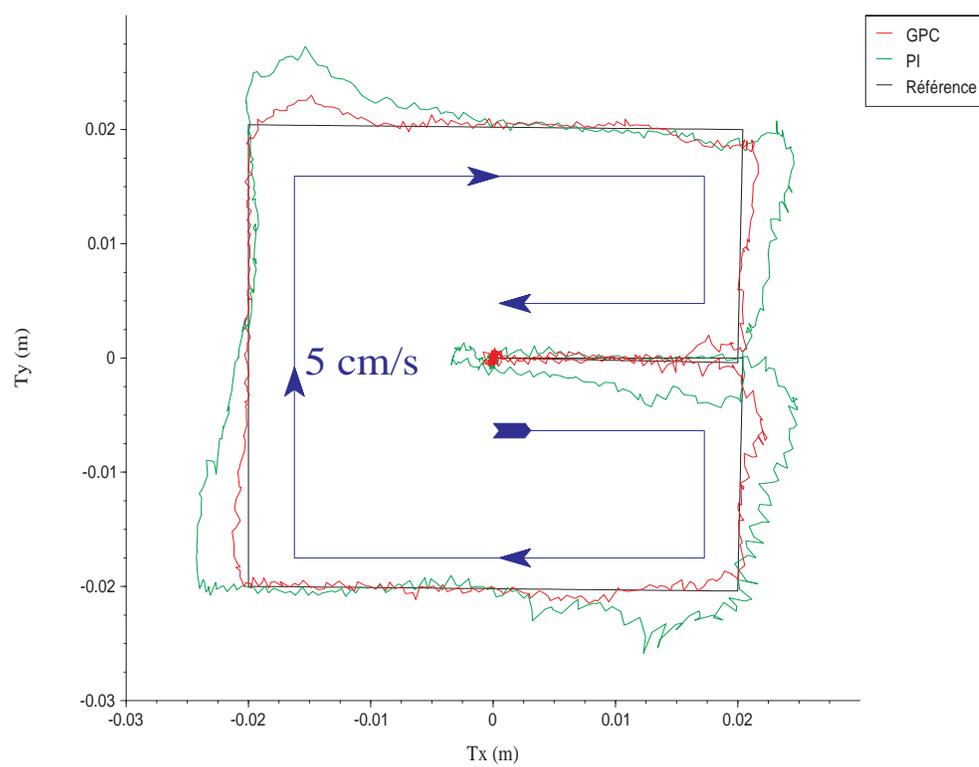


Fig. 4.17: Suivi d'une trajectoire dans un plan parallèle au plan image

4.3 Conclusion

Dans ce chapitre, nous avons utilisé l'application de suivi de cible pour valider notre modélisation dynamique de la boucle de vision. Les approches 2D et 3D ont été abordées. Etant donné que le modèle dynamique du robot dans la boucle de vision est défini par rapport aux coordonnées opérationnelles, et que l'asservissement 2D classique est réalisé exclusivement dans le plan image, nous avons proposé une variante basée sur une transformation des informations 2D en information 3D. Grâce à cette transformation, les 2 types d'asservissements (2D et 3D) peuvent être modélisés dynamiquement de la même façon.

Si les approches 2D et 3D sont identiques du point de vue du modèle, il n'en est pas de même dans la pratique. Les expériences ont mis en évidence que les mesures extraites de l'image par la technique 2D étaient nettement moins bruitées que celles extraites par une estimation 3D. Il en résulte que, à correcteur égal, l'asservissement 3D est très oscillant tandis que l'asservissement 2D est parfaitement stable.

Partant de ce constat, tous les essais visant à valider le modèle dynamique ont été réalisés pour un asservissement 2D. Pour ces essais, nous avons comparé la réponse d'une boucle contrôlée par un correcteur PI à la réponse de cette boucle contrôlée par un correcteur GPC. Le GPC a la particularité de pouvoir être implémenté de manière adaptative. Il est réglé de manière optimale en fonction du modèle dynamique de la boucle de vision. Le PI est réglé expérimentalement de manière à obtenir le meilleur compromis entre performances et stabilité.

Tous les essais que nous avons réalisés ont montré la supériorité du correcteur GPC sur le correcteur PI du point de vue des performances dynamiques. Ils valident de ce fait expérimentalement le modèle dynamique de la boucle de vision que nous avons proposé.

Chapitre 5

Le suivi de profilé

De nombreuses applications industrielles mettent en oeuvre des tâches de suivi de profilé. Par exemple, la dépose de colle le long d'un montant de portière de voiture ou autour d'un pare-brise, la réalisation d'un joint de soudure le long d'une arrête ou l'ébavurage d'une pièce de polymère moulée. Le suivi de profilé consiste à asservir la distance entre l'organe terminal du robot et la pièce, ainsi que l'angle. L'approche standard pour la réalisation d'une telle tâche consiste en un apprentissage de positions successives le long de la trajectoire à réaliser, la trajectoire entre deux positions apprises étant le plus souvent réalisée par interpolation linéaire. Cette approche présente trois inconvénients :

- Le profilé doit être précisément positionné par rapport au robot. Tout décalage de cette position se traduit par un décalage identique de la trajectoire réalisée par rapport à celle désirée. Quand il se produit un tel décalage, dans le meilleur des cas, seule la pièce sera perdue et dans le pire des cas c'est l'outil voir le robot qui peuvent être endommagés.
- L'apprentissage se fait «à vue». Souvent, le positionnement se fait au jugé de l'opérateur et sans aucun instrument de mesure précis. Cette façon de procéder est d'autant plus ardue que la courbure du profilé est plus complexe. Aussi, de nombreux essais et corrections sont nécessaires avant d'obtenir la trajectoire idéale d'où une perte importante de temps.
- Un nombre fini de positions sont mémorisés le long du profilé. C'est par interpolation linéaire ou spline que la trajectoire entre deux points est calculée. Ainsi, pour une courbure complexe, l'outil ne respecte la position de référence qu'aux points mémorisés par apprentissage.

Une autre approche consiste à munir le robot d'un capteur extéroceptif, c'est à dire capable de réaliser des mesures sur l'environnement de travail du robot. Dans la plupart des cas, il s'agit d'une caméra fixée sur l'organe terminal du robot. Ainsi, si l'objet contient des primitives visuelles assez riches, l'information visuelle peut donner à chaque nouvelle image une mesure de sa position par rapport à l'effecteur du robot. L'utilisation d'une caméra fixée à l'organe terminal du robot pour réaliser

une telle tâche de suivi de profilé ne présente aucun des trois inconvénients de l'approche standard. En effet,

- Aucune opération d'apprentissage préliminaire n'est nécessaire en dehors de la définition de la transformation de référence entre l'outil et le profilé.
- Il n'est pas nécessaire de référencer la position du profilé par rapport à la base du robot. De plus, le profilé peut être animé d'un mouvement par rapport au robot au cours de l'opération de suivi.
- A tout moment, l'asservissement par vision tend à annuler l'erreur de suivi et en tout point le long du profilé.

Des travaux antérieurs se sont intéressés au suivi de profilé. On peut assimiler la simulation de suivi de route décrite dans [ECR92] par Espiau *et al.* à une tâche de suivi de profilé. Son but est de maintenir la caméra à une certaine hauteur au-dessus de la voie de droite, dirigée dans le sens de la route. La route est matérialisée par 3 droites parallèles. Cette simulation met en oeuvre un asservissement par vision dans le plan image. Les primitives du Jacobien image sont des droites au lieu de points. Le calcul du noyau du Jacobien image fournit les directions qui laissent les primitives visuelles inchangées, c'est à dire les directions du suivi.

L'expérience de suivi de ligne décrite par Lange *et al.* dans [LWH98] est un asservissement visuel à 2 dimensions. La courbe à suivre est contenue dans un plan parallèle au plan image de la caméra. Le traitement d'image est très simple et peut donc être très rapide. Ils atteignent ainsi une cadence d'échantillonnage de 50 Hz. L'erreur du suivi n'excède pas 0.3 mm à une vitesse de 1.6 m/s. L'originalité de travail réside dans le fait que ses auteurs utilisent la vision pour prédire des consignes futures. Ils ont ainsi mis en oeuvre une stratégie de contrôle basée sur un correcteur prédictif. Néanmoins, le fait de n'asservir que deux degrés de liberté, simplifie grandement le problème.

Dans notre approche, le profilé est tridimensionnel. Il doit contenir au moins 3 courbes parallèles distinctes visibles dans l'image. C'est le minimum pour pouvoir estimer les 6 coordonnées opérationnelles de la transformation entre le repère caméra et le repère lié à une section du profilé. Notre asservissement visuel est basé sur les coordonnées 3D et non pas réalisé dans le plan image (voir section 1.1). Nous proposons une méthode pour reconstruire les 6 paramètres opérationnels à partir des tangentes à la projection du profilé. Un des problèmes à résoudre est la détermination, le long du profilé, de la section courante. En effet, la vision ne donne aucune information de position le long du profilé. Cette mesure doit être obtenue par un autre moyen. A cette fin, nous utilisons le modèle géométrique du robot calculé grâce aux valeurs des codeurs incrémentaux sur les axes du robot qui donnent une mesure des coordonnées articulaires.

5.1 Exploitation de l'information visuelle

5.1.1 Définition d'un profilé

Un profilé est défini comme suit :

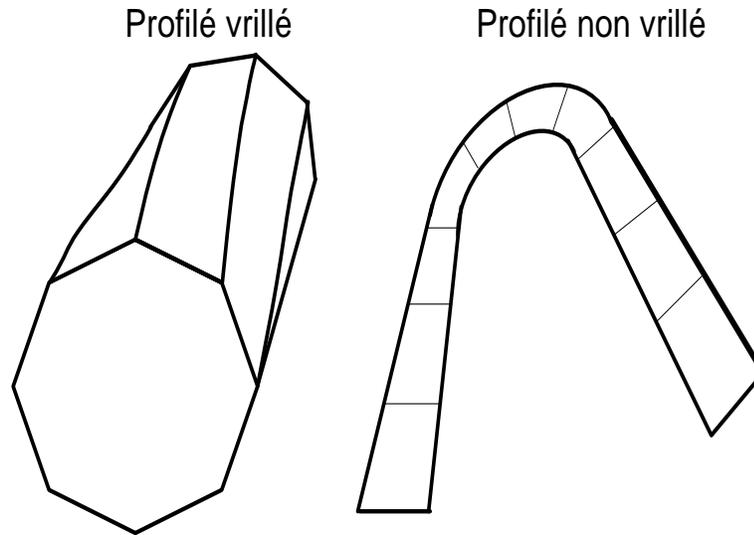


Fig. 5.1: Comparaison entre profilé vrillé et profilé non vrillé

Définition 5.1 (Profilé). Soit \mathcal{C} une courbe 3D quelconque. Soit \mathcal{S} une surface plane quelconque et O un point appartenant à \mathcal{S} . Un profilé est généré par le déplacement de \mathcal{S} le long de \mathcal{C} tel que $\mathcal{C} \cap \mathcal{S} = O$ et \mathcal{S} perpendiculaire au vecteur vitesse curviligne de \mathcal{C} en O ; \mathcal{C} est appelée génératrice du profilé; \mathcal{S} est la section du profilé.

Dans l'ensemble des profilés on peut distinguer 4 sous-ensembles particuliers : les profilés plans, les profilés filaires, les profilés non vrillés et les profilés vrillés. La section d'un profilé plan est un segment de droite et celle d'un profilé filaire est un point. Les profilés non vrillés peuvent être définis comme suit :

Définition 5.2 (Profilé non vrillé). Un profilé non vrillé peut être découpé en segments pour lesquels toutes les sections sont liées par une translation de même direction (dans le cas d'un segment linéaire) ou une rotation de même centre de rotation (dans le cas d'un segment courbe).

Les profilés qui ne correspondent pas à cette définition sont des profilés vrillés. La figure (5.1) donne un exemple de profilé vrillé comparé à un profilé non-vrillé. Dans la suite de ce chapitre, nous expliquerons pourquoi les profilés non vrillés présentent les caractéristiques requises pour l'algorithme de suivi. Aussi, lorsqu'il sera fait mention de profilé dans la suite, la caractéristique «non vrillé» sera toujours sous-entendue.

5.1.2 Propriétés géométriques d'un profilé

La section d'un profilé industriel classique contient généralement des arêtes. Si le profilé est non vrillé, ces arêtes génèrent des courbes parallèles \mathcal{A}_i le long du profilé. Etant donné que dans un segment du profilé, toutes les sections sont liées par une rotation de même centre ou une translation de même direction, les tangentes aux

courbes \mathcal{A}_i en des points contenus dans une même section sont parallèles. Dans le cas général, ces tangentes ne sont pas définies aux points de jonction entre deux segments où la transformation, translation ou rotation, change respectivement de direction ou de centre. C'est pourquoi, nous supposons dans ce qui suit l'existence de la dérivée première de la courbure en tout point du profilé, c'est à dire qu'il n'y a pas de discontinuité du vecteur vitesse curviligne.

5.1.3 Propriétés de la projection perspective d'une courbe 3D

Afin de déterminer la transformation entre le repère lié à la caméra et le profilé, nous utilisons une propriété bien connue de la projection perspective :

Lemme 5.1. Projection perspective de la tangente à une courbe 3D.

La projection perspective de la tangente en P à une courbe tridimensionnelle, \mathcal{C} , dérivable est identique à la tangente à la projection de \mathcal{C} au point projeté de P .

La démonstration de ce lemme est reportée en annexe A.1.

Cette propriété est exploitée par d'autres auteurs pour réaliser la reconstruction 3D d'un profilé à partir de son image [KW90, DD90].

Nous verrons dans la section 5.1.5, que la projection dans le plan image d'au moins 3 droites parallèles dont l'écartement deux à deux est connu, suffit à définir entièrement la transformation entre un repère lié à la caméra et un repère lié à ces droites. Ainsi, étant donné que les tangentes aux arrêtes d'un profilé non vrillé en des points qui sont contenus dans une même section sont parallèles, la connaissance de la projection dans le plan image d'au moins 3 arêtes du profilé et la connaissance de la géométrie d'une section suffisent pour calculer la transformation entre la caméra et une section du profilé.

5.1.4 Modélisation des primitives visuelles

L'utilisation du lemme 5.1 impose de pouvoir trouver avec une bonne précision la tangente aux projetées des arrêtes d'un profil dans l'image. Pour cela, les courbes dans l'image sont approximées par des équations polynômiales d'ordre élevé. De cette façon, tous les points d'une courbe contribuent à sa modélisation par opposition à des méthodes ne prenant en compte que les points dans un voisinage de l'endroit de la courbe où il est nécessaire de déterminer la tangente. La figure 5.2 donne un aperçu de la procédure de traitement d'image. Après une première détection globale des contours, grâce à un masque de convolution 3x3 de Sobel, un modèle polynômial des primitives dans l'image est déterminé. Puis, pour toutes les images suivantes, seules certaines imageries sont acquises. Cette restriction du traitement à certaines zones permet de limiter considérablement le temps de calcul. Ces imageries sont localisées le long du profilé en fonction de sa position dans l'image précédente. Ainsi, nous faisons la supposition que cette position ne varie pas de plus de la moitié de la hauteur d'une de ces imageries d'une image à l'autre (ceci fixe une limite supérieure

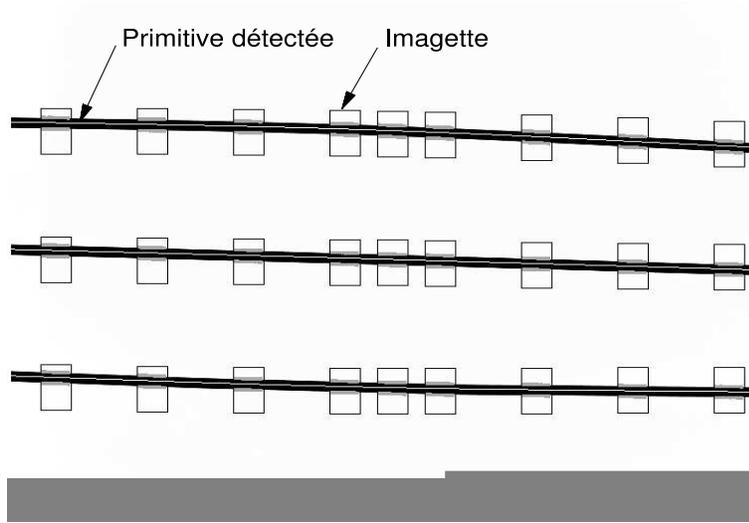


Fig. 5.2: Le traitement d'image

pour la vitesse de déplacement du profilé avec une certaine cadence d'acquisition des images). Un filtrage de Sobel est réalisée dans chacune de ces imagettes. Le calcul du barycentre de tous les points où le gradient dépasse un certain seuil, pondéré par ce même gradient, fourni les coordonnées sub-pixel d'un point théoriquement localisé sur la ligne médiane de la courbe correspondante. Ainsi, le traitement d'image donne la position de N points ($N = 9$ dans la figure 5.2) appartenant à chaque courbe. Ils sont utilisés pour déterminer un modèle polynômial par un calcul des moindres carrés.

5.1.5 Calcul de la position du profilé

Dans toute cette section, nous considérons que la caméra a une distance focale $f = 1$. Ceci ne remet pas en question la généralité des démonstrations car toute mesure dans l'image peut être convertie en une mesure dans un plan image virtuel de longueur focale $f = 1$. Soit $\mathcal{T}_{pi} = \mathcal{P}(\mathcal{T}_i)$, $i = 1..n$, les projections perspectives dans le plan image des tangentes \mathcal{T}_i au profilé à des points P_i appartenant à une même section :

$$\begin{aligned} \mathcal{T}_i &= \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}_{R_c} = k_i \vec{V}_i + P_i = k_i \begin{pmatrix} v_{xi} \\ v_{yi} \\ v_{zi} \end{pmatrix}_{R_c} + \begin{pmatrix} r_{xi} \\ r_{yi} \\ r_{zi} \end{pmatrix}_{R_c} \\ \mathcal{T}_{pi} &= \begin{pmatrix} x_{pi} \\ y_{pi} \\ z_{pi} \end{pmatrix}_{R_c} = k_{pi} \begin{pmatrix} u_{xi} \\ u_{yi} \\ 0 \end{pmatrix}_{R_c} + \begin{pmatrix} o_{xi} \\ o_{yi} \\ 1 \end{pmatrix}_{R_c} \end{aligned} \quad (5.1)$$

où k_i et k_{pi} sont des scalaires paramétrisant les tangentes et les \vec{V}_i sont des vecteurs directeurs unitaires (*i.e.* $\|\vec{V}_i\| = 1$). Ces équations sont exprimées dans un repère

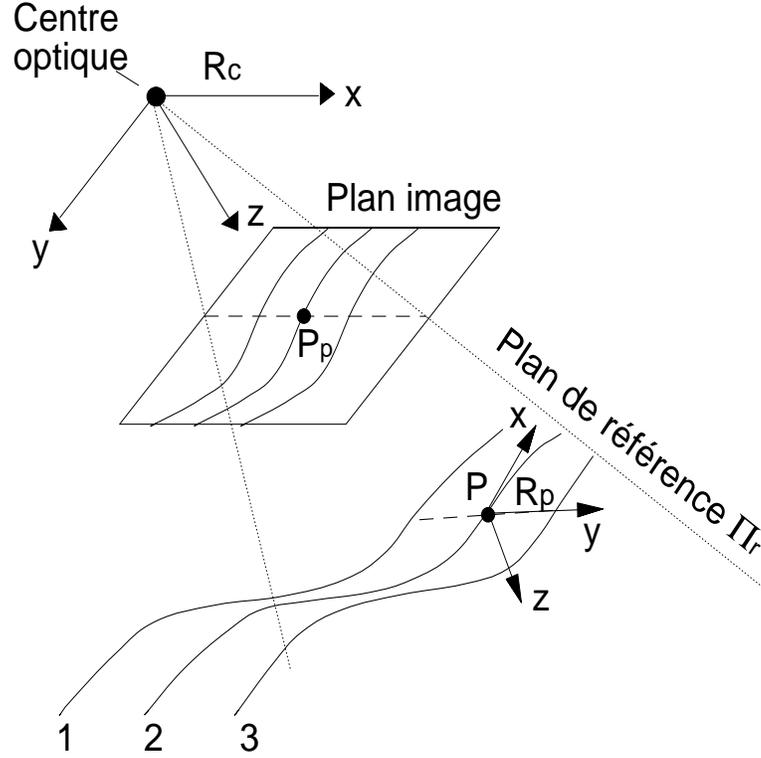


Fig. 5.3: Definition du plan de référence

R_c lié à la caméra. Ce repère est placé de telle sorte que son origine soit au centre optique de la caméra, que l'axe des z soit perpendiculaire au plan image, que l'axe des x soit vertical dans l'image et que l'axe des y soit horizontal (voir figure 5.3).

Les paramètres u_{xi} , u_{yi} , o_{xi} et o_{yi} sont déterminés grâce aux informations visuelles. Pour cela, il est nécessaire de connaître la position des projections dans l'image des points P_i appartenant à une même section. Nous faisons l'hypothèse d'une faible courbure des projections des arêtes du profilé dans l'image. Ainsi, nous pouvons faire l'approximation que les points appartenant à une même section dans l'image sont alignés suivant une droite perpendiculaire à la projection d'une des arêtes du profilé. On commet ainsi une erreur négligeable sur l'estimation des paramètres u_{xi} , u_{yi} , o_{xi} et o_{yi} des tangentes \mathcal{T}_{pi} (voir figure 5.4b).

Les paramètres tridimensionnels des droites \mathcal{T}_i et les paramètres de leur projection perspective \mathcal{T}_{pi} sont liés par les deux équations suivantes : (voir [HS93]) : $\forall i = 1 \dots n$,

$$u_{yi}r_{xi} - u_{xi}r_{yi} + (o_{yi}u_{xi} - o_{xi}u_{yi})r_{zi} = 0 \quad (5.2)$$

$$u_{yi}v_{xi} - u_{xi}v_{yi} + (o_{yi}u_{xi} - o_{xi}u_{yi})v_{zi} = 0 \quad (5.3)$$

La démonstration de ces équations se trouve en annexe A.2. Comme les droites \mathcal{T}_i

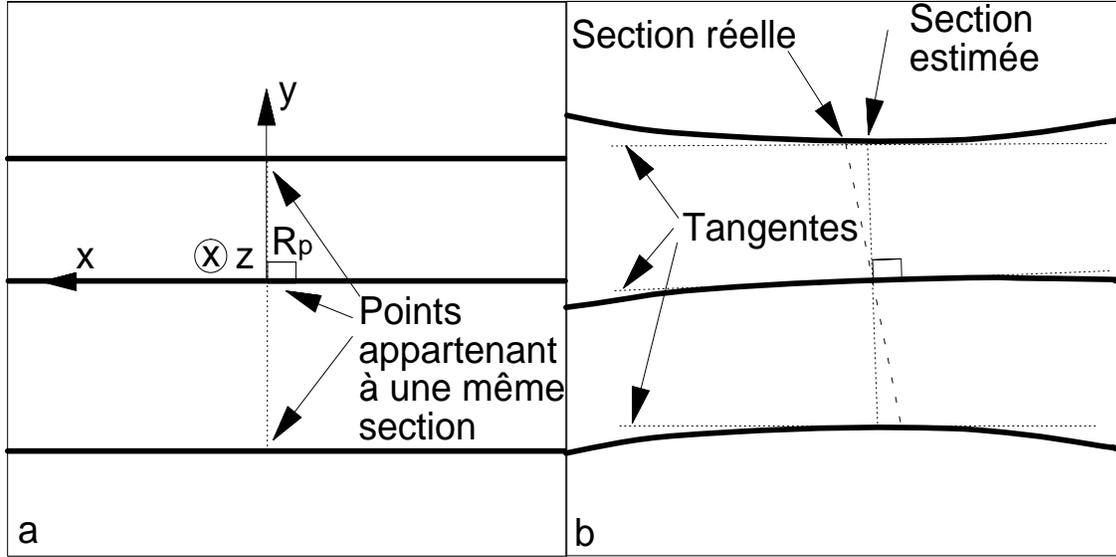


Fig. 5.4: Recherche de points dans l'image appartenant à une même section

sont parallèles :

$$\vec{V}_i = \begin{pmatrix} v_{xi} \\ v_{yi} \\ v_{zi} \end{pmatrix}_{R_c} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}_{R_c} = \vec{V} \quad \forall i = 1 \dots n \quad (5.4)$$

De plus, puisque $\|\vec{V}\| = 1$:

$$v_x^2 + v_y^2 + v_z^2 = 1 \quad (5.5)$$

Comme les paramètres des droites \mathcal{T}_{pi} sont connus, car mesurés dans le plan image (voir section 5.1.4), l'équation (5.3) contient 3 inconnues : v_x , v_y et v_z :

$$u_{yi}v_x - u_{xi}v_y + (o_{yi}u_{xi} - o_{xi}u_{yi})v_z = 0 \quad i = 1 \dots n \quad (5.6)$$

En résolvant ces équations pour $n \geq 2$ et en utilisant (5.5), nous obtenons \vec{V} , le vecteur directeur commun à toutes les tangentes \mathcal{T}_i .

Nous allons maintenant définir le modèle géométrique d'une section du profilé. Pour ce faire, définissons un repère, R_p , lié au profilé (voir les figures 5.4 et 5.3). Ce repère R_p est choisi de manière à ce que tous les points appartenant à la section considérée soient contenus dans le plan (yz) . Soient P_i , $i = 1 \dots n$, ces points :

$$P_i = \begin{pmatrix} 0 \\ P_{yi} \\ P_{zi} \end{pmatrix}_{R_p} \quad (5.7)$$

Les valeurs des paramètres P_{yi} et P_{zi} sont déterminés grâce à la connaissance de la géométrie de la section du profilé. D'autre part, nous choisissons l'origine de R_p

confondue avec l'un de ces points (le point central dans l'image). Soit $P \in \{P_i\}$ ce point central, alors :

$$P = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}_{R_p} \quad (5.8)$$

Ainsi, l'axe des x du repère R_p est choisi tangent au profilé en P . Afin de totalement fixer la position de R_p par rapport au profilé, nous imposons que y passe par un des points P_i .

Soit M_{cp} la transformation homogène entre le repère R_c lié à la caméra et le repère R_p lié au profilé exprimée dans R_c :

$$M_{cp} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}_{R_c} \quad (5.9)$$

Les composantes de la première colonne de M_{cp} sont connues. En effet, la sous-matrice 3×3 de rotation de M_{cp} contient 3 vecteurs orthonormés qui sont les coordonnées dans le repère R_c des 3 vecteurs unitaires \vec{x}_p , \vec{y}_p et \vec{z}_p du repère R_p . Etant donné que \vec{x}_p est choisi tangent au profilé et que cette direction est donnée par \vec{V} , on en déduit que $r_{11} = v_x$, $r_{21} = v_y$ et $r_{31} = v_z$. Ainsi, grâce à M_{cp} , les points P_i peuvent être exprimés dans R_c :

$$P_i = \begin{pmatrix} r_{xi} \\ r_{yi} \\ r_{zi} \end{pmatrix}_{R_c} = \begin{pmatrix} r_{12}P_{yi} + r_{13}P_{zi} + T_x \\ r_{22}P_{yi} + r_{23}P_{zi} + T_y \\ r_{32}P_{yi} + r_{33}P_{zi} + T_z \end{pmatrix}_{R_c} \quad i = 1 \dots n \quad (5.10)$$

D'où, en utilisant (5.10 dans (5.2)), on obtient :

$$\begin{aligned} & u_{yi}(r_{12}P_{yi} + r_{13}P_{zi} + T_x) - u_{xi}(r_{22}P_{yi} + r_{23}P_{zi} + T_y) \\ & + (o_{yi}u_{xi} - o_{xi}u_{yi})(r_{32}P_{yi} + r_{33}P_{zi} + T_z) = 0 \quad i = 1 \dots n \end{aligned} \quad (5.11)$$

Ces équations contiennent 9 inconnues : r_{12} , r_{13} , r_{22} , r_{23} , r_{32} , r_{33} , T_x , T_y , T_z . Les propriétés de la sous-matrice de rotation de M_{cp} fournissent 5 équations :

$$r_{12}^2 + r_{22}^2 + r_{32}^2 = 1 \quad (5.12)$$

$$r_{13}^2 + r_{23}^2 + r_{33}^2 = 1 \quad (5.13)$$

$$v_x r_{12} + v_y r_{22} + v_z r_{32} = 0 \quad (5.14)$$

$$v_x r_{13} + v_y r_{23} + v_z r_{33} = 0 \quad (5.15)$$

$$r_{12}r_{13} + r_{22}r_{23} + r_{32}r_{33} = 0 \quad (5.16)$$

La sixième équation est obtenue par la définition d'un plan de référence Π_r . Ce plan de référence Π_r (voir figure 5.3) est défini par le centre optique de la caméra qui est

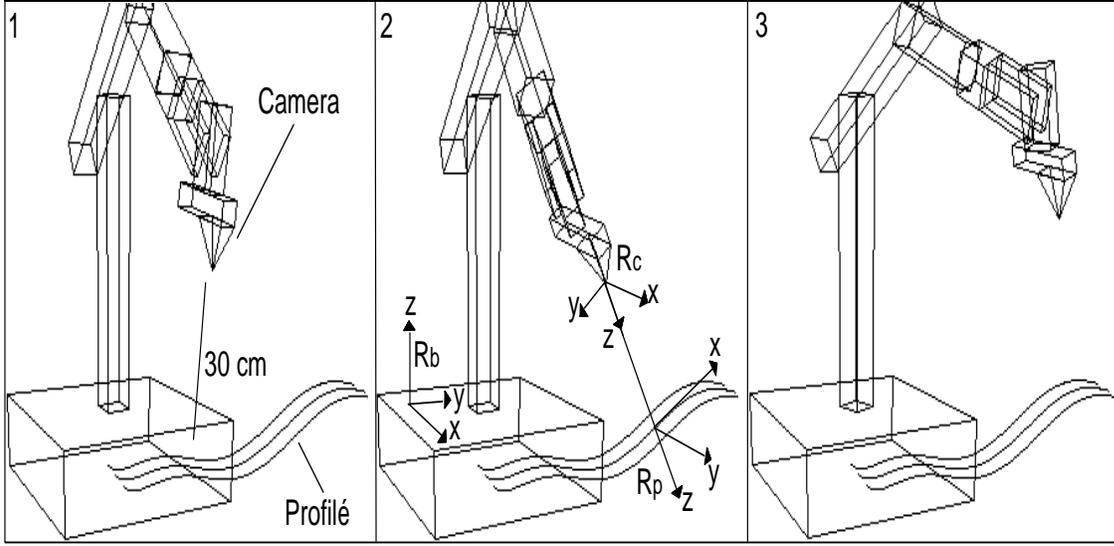


Fig. 5.5: Simulation du suivi de profilé

aussi l'origine du repère R_c , et une verticale dans le plan image passant par P_p , la projection de P . Ce plan fixe ainsi la position P de la section courante le long du profilé. Son équation est la suivante :

$$\Pi_r = \begin{pmatrix} \pi_x \\ \pi_y \\ \pi_z \end{pmatrix}_{R_c} \quad \text{avec} \quad \pi_y - a\pi_z = 0 \quad (5.17)$$

où a est un scalaire choisi de telle sorte que l'intersection entre Π_r et le plan image soit située dans l'image et que le déplacement le long du profilé se fasse à une vitesse V_f prédéfinie (Nous verrons dans la section 5.2 comment choisir ce paramètre a). Finalement, puisque P , l'origine de R_p , appartient à Π_r , on obtient l'équation :

$$T_y - aT_z = 0 \quad (5.18)$$

En conclusion, pour calculer M_{cp} , il est donc nécessaire d'avoir la projection dans l'image d'au moins 3 droites parallèles afin de fournir 3 équations (5.11) supplémentaires.

Cas particulier

Il est possible de trouver facilement une solution analytique à ces équations dans le cas particulier où on a un profilé plan constitué de 3 arêtes (voir figure 5.5). Dans ce cas, les tangentes à ces arêtes aux points d'une même section sont 3 droites parallèles coplanaires. Soient P_1 , P_2 , P_3 les 3 points de tangence définis dans le

repère R_p attaché à la section courante :

$$P_1 = \begin{pmatrix} 0 \\ d_1 \\ 0 \end{pmatrix}_{R_p} \quad P_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}_{R_p} \quad P_3 = \begin{pmatrix} 0 \\ -d_3 \\ 0 \end{pmatrix}_{R_p} \quad (5.19)$$

où d_1 et d_3 sont des valeurs connues donnant la distance entre les arêtes du profilé. De (5.11) découlent les 3 équations suivante :

$$u_{y1}(d_1 r_{12} + T_x) - u_{x1}(d_1 r_{22} + T_y) + (o_{y1} u_{x1} - o_{x1} u_{y1})(d_1 r_{32} + T_z) = 0 \quad (5.20)$$

$$u_{y2} T_x - u_{x2} T_y + (o_{y2} u_{x2} - o_{x2} u_{y2}) T_z = 0 \quad (5.21)$$

$$u_{y3}(-d_3 r_{12} + T_x) - u_{x3}(-d_3 r_{22} + T_y) + (o_{y3} u_{x3} - o_{x3} u_{y3})(-d_3 r_{32} + T_z) = 0 \quad (5.22)$$

En utilisant (5.21) et (5.18), on obtient :

$$T_x = \frac{u_{x2} a - o_{y2} u_{x2} + o_{x2} u_{y2}}{u_{y2}} T_z \quad (5.23)$$

De (5.14), il vient :

$$r_{12} = -\frac{v_y r_{22} + v_z r_{32}}{v_x} \quad (5.24)$$

En utilisant (5.18), (5.23) et (5.24) dans (5.20) et (5.22), on obtient :

$$u_{y1} \left(-\frac{d_1(v_y r_{22} + v_z r_{32})}{v_x} + \frac{T_z(u_{x2} a - o_{y2} u_{x2} + o_{x2} u_{y2})}{u_{y2}} \right) - u_{x1}(d_1 r_{22} + a T_z) + (o_{y1} u_{x1} - o_{x1} u_{y1})(d_1 r_{32} + T_z) = 0 \quad (5.25)$$

et

$$u_{y3} \left(\frac{d_3(v_y r_{22} + v_z r_{32})}{v_x} + \frac{T_z(u_{x2} a - o_{y2} u_{x2} + o_{x2} u_{y2})}{u_{y2}} \right) - u_{x3}(-d_3 r_{22} + a T_z) + (o_{y3} u_{x3} - o_{x3} u_{y3})(-d_3 r_{32} + T_z) = 0 \quad (5.26)$$

Seules 3 inconnues, r_{22} , r_{32} et T_z subsistent dans ces 2 équations. La résolution de (5.25) et (5.26) pour r_{22} et r_{32} , donne 2 relations :

$$r_{22} = K_2 T_z \quad (5.27)$$

$$r_{32} = K_3 T_z \quad (5.28)$$

De plus, l'utilisation de (5.24) nous donne :

$$r_{12} = K_1 T_z \quad (5.29)$$

L'expression de K_1 , K_2 et K_3 est reportée en annexe B.8. Finalement, l'utilisation de (5.29) et (5.27) dans la dernière équation (5.12) fournit la valeur de T_z^2 . Comme le profilé est toujours face à la caméra, $T_z > 0$. Nous en déduisons T_z et par conséquent, toutes les autres composantes de la matrice M_{cp} .

5.2 Détermination de la section courante le long du profilé

L'information visuelle ne donne aucune information quant à la position le long du profilé. Mais, afin de réaliser un suivi à une vitesse constante V_f le long du profilé, cette position doit être mesurée par un autre moyen. A cette fin, nous utilisons le modèle géométrique du robot.

5.2.1 Détermination de la position le long du profilé

Soit M_{bp}^- la transformation homogène entre R_b , le repère de base (voir figure 5.5) et R_p^- , le repère attaché à la section précédente (la section courante pendant la dernière période d'échantillonnage). Nous la notons M_{bp}^- afin de la distinguer de M_{bp} , la transformation entre le repère de base et le repère R_p attaché à la section courante. Il vient :

$$M_{bp}^- = M_{bc}^- M_{cp}^- \quad (5.30)$$

où M_{bp}^- est la transformation homogène entre le repère de base et le repère attaché à la caméra (ou repère caméra) obtenue par calcul du modèle géométrique du robot à l'instant d'échantillonnage précédent et utilisant les informations provenant des codeurs angulaires des axes (voir l'équation 2.26, page 33). Si V_f est la vitesse de suivi de profilé, alors la position théorique, M_{bp}^0 , du repère R_p attaché à la section courante (noté R_p^0 pour signifier qu'il s'agit de la position théorique de R_p) est donné par :

$$M_{bp}^0 = M_{bp}^- T_x \quad (5.31)$$

où T_x est une matrice homogène faisant une translation de distance $T_e V_f$ (où T_e est la période d'échantillonnage) suivant la direction de l'axe des x de R_p^- (voir figure 5.6). Cependant, M_{bp}^0 ne traduit la position de la section courante réelle que dans l'hypothèse où le profilé est droit et immobile et où le modèle cinématique du robot est parfaitement connu. Dans la réalité, ces conditions ne sont jamais remplies. Cette position doit donc être corrigée grâce à l'information visuelle.

5.2.2 Recherche de la section courante

Le but est de rechercher, grâce à l'information visuelle, la position du repère R_p attaché à la section courante sachant que l'on dispose de sa position théorique, M_{bp}^0 . La vision ne peut fournir aucune information de position le long du profilé. Aussi, faut-il imposer une contrainte fixant cette position. Ainsi, dans ce cas précis, il est naturel d'imposer que l'origine P du repère R_p soit contenue dans le plan $(yz)^0$ formé par les axes y et z du repère théorique R_p^0 . Si on considère que la distance de translation $T_e V_f$ d'une section courante à une autre est négligeable par rapport au rayon de courbure du profilé dans la région considérée, alors l'intersection entre $(yz)^0$ et la courbe de référence du profilé (celle qui sert de support au repère R_p) donne la

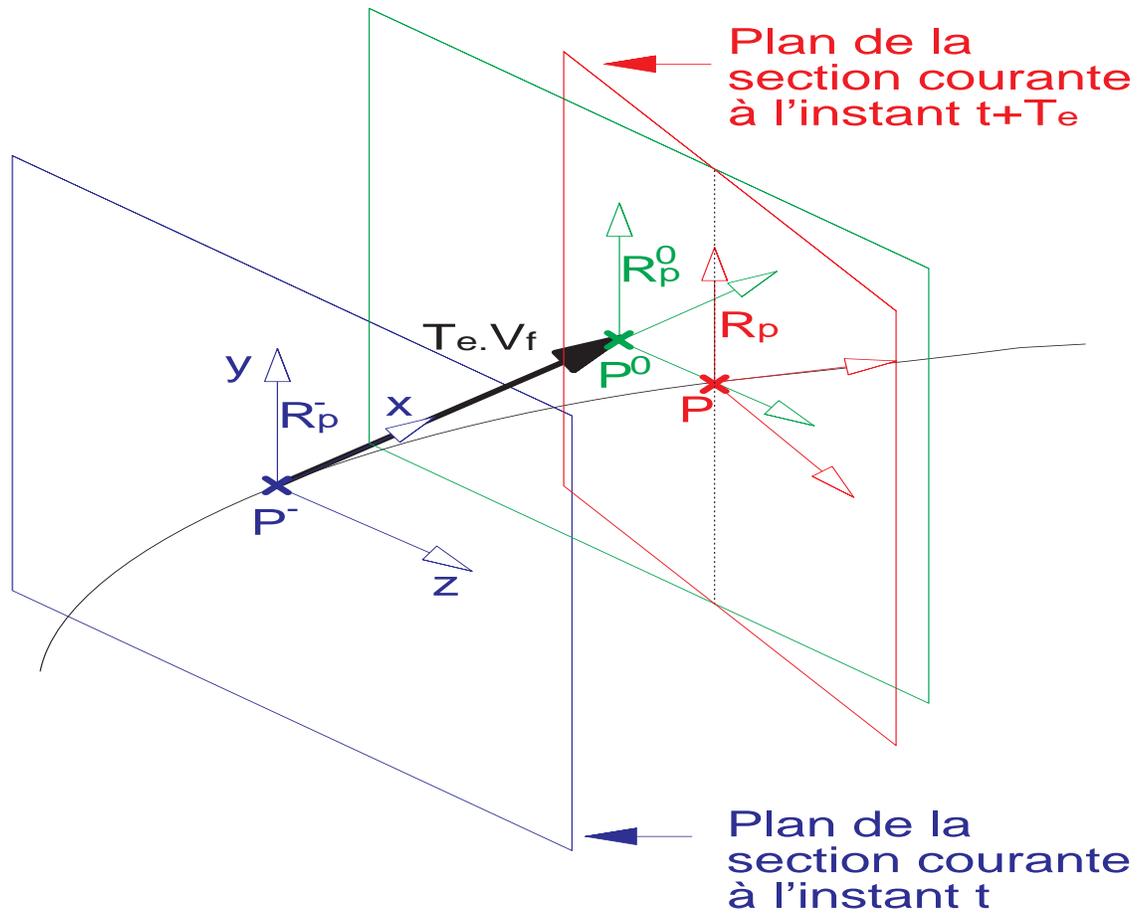


Fig. 5.6: Détermination de la section courante

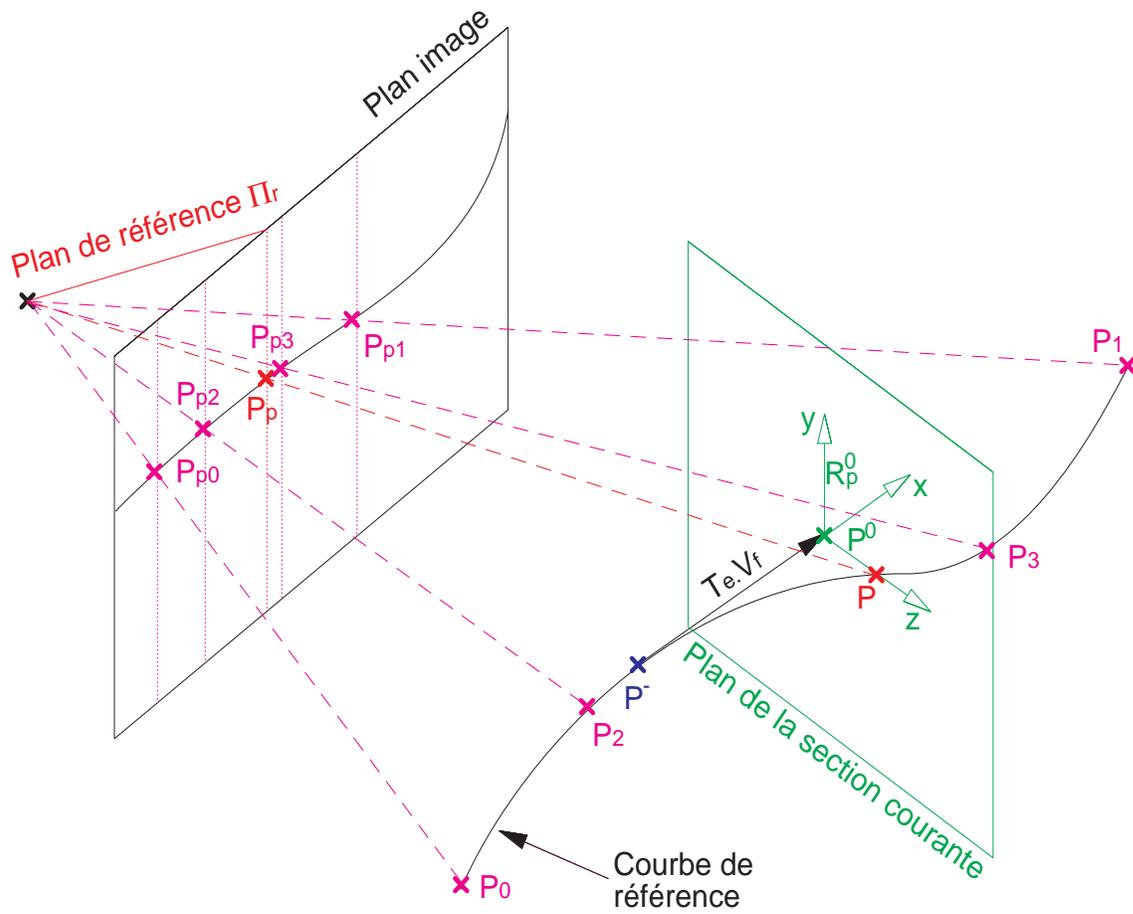


Fig. 5.7: Algorithme itératif de recherche de la section courante

position de P telle que la distance curviligne entre P^- et P soit approximativement égale à $T_e V_f$. Dans la pratique, cette condition est aisément vérifiée.

Le profilé est susceptible de se mouvoir lors de la tâche de suivi. Dans ce cas, la différence entre R_p^0 et R_p est d'autant plus grande que la vitesse de déplacement du profilé est grande. Néanmoins, seul un déplacement dans la direction du profilé ne peut être détecté. Un déplacement suivant les 5 autres degrés de liberté peut être mesuré par la vision et ainsi être compensé par l'asservissement visuel.

Recherche itérative de P

L'intersection entre le plan $(yz)^0$ et la courbe de référence ne peut être trouvée que par itération. Les itérations ont pour but de rechercher dans le plan image le point P_p , projeté du point P , tel que $P \in (yz)^0$. Ces itérations ne mettent en jeu que des informations provenant de l'image. Ainsi, la position de R_p^0 doit être exprimée dans le repère caméra :

$$M_{cp}^0 = M_{bc}^{-1} M_{bp}^0 \quad (5.32)$$

où M_{bc} est obtenue en calculant le modèle géométrique du robot à l'instant d'échantillonnage courant. Le paramètre variable dans cette procédure itérative est a , le scalaire définissant la position du plan de référence Π_r (voir équation (5.18), page 95) qui intersecte la courbe de référence en P . La condition qui doit être vérifiée est la suivante :

$$\mathcal{S} = \overrightarrow{P^0 P} \cdot \vec{x}^0 = 0 \quad (5.33)$$

où \vec{x}^0 est le vecteur directeur de l'axe x du repère R_p^0 . Cette condition est équivalente à $P \in (yz)^0$. Lorsque les itérations ont convergé, le plan de référence Π_r coupe la courbe de référence en P et le paramètre a est déterminé. En suivant la procédure décrite dans la section 5.1.5, on peut alors calculer M_{cp} , la transformation entre le repère caméra et le repère R_p lié à la section courante.

La figure 5.7 décrit le processus de recherche par itérations. Soit k le numéro de l'itération courante. Soit, a_k la valeur courante du paramètre a telle que le plan de référence Π_k correspondant coupe la projection de la courbe de référence en P_{pk} . Grâce aux calculs décrits dans la section 5.1.5, on en déduit une transformation M_{cpk} entre le repère caméra et un repère lié au profilé en P_k (P_k se projetant en P_{pk}). En fonction de la valeur du produit scalaire $\mathcal{S}_k = \overrightarrow{P^0 P_k} \cdot \vec{x}^0$ (5.33), on en déduit la position de $P_{p(k+1)}$ en faisant l'approximation d'une relation linéaire entre \mathcal{S}_k et a_k :

$$\begin{cases} \mathcal{S}_k &= K(a_k - a) \\ \mathcal{S}_{k-1} &= K(a_{k-1} - a) \end{cases} \quad (5.34)$$

De ces deux équations, on déduit a_{k+1} , la valeur réactualisée de a :

$$a_{k+1} = \frac{\mathcal{S}_k a_{k-1} - \mathcal{S}_{k-1} a_k}{\mathcal{S}_k - \mathcal{S}_{k-1}} \quad (5.35)$$

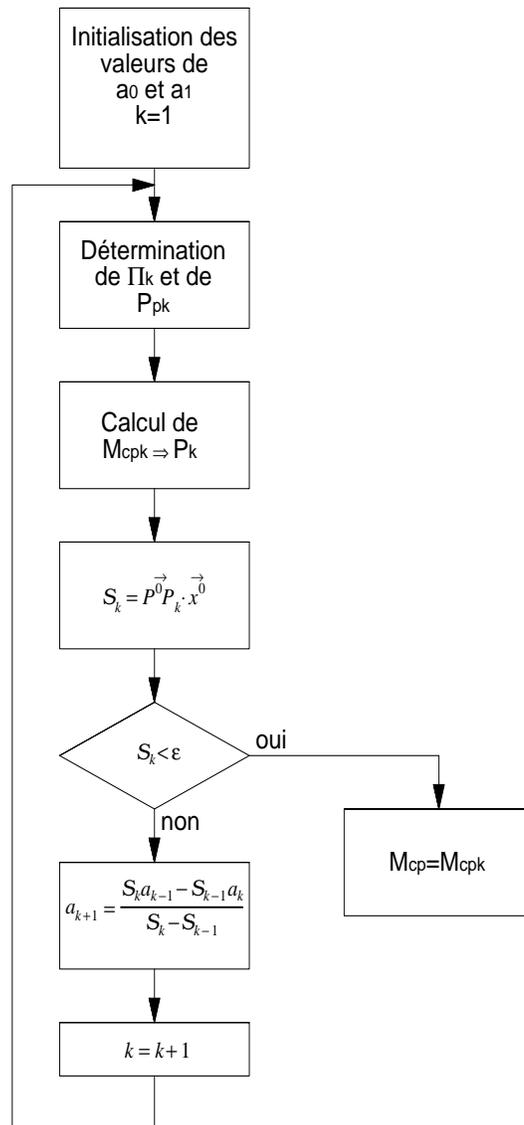


Fig. 5.8: Organigramme de l'algorithme récursif de recherche de la section courante

De a_{k+1} on déduit la position de Π_{k+1} et donc de $P_{p(k+1)}$. Cette procédure doit être initialisée en choisissant deux points P_{p0} et P_{p1} aux 2 extrémités de l'image correspondant à respectivement à a_0 et a_1 tels que :

$$a \in \text{à l'intervalle } [a_0 \ a_1] \quad (5.36)$$

La procédure itérative se poursuit n fois jusqu'à ce que $\mathcal{S}_n < \varepsilon$ où ε correspond à la limite de précision de calcul fixée à l'avance. Lorsque cette condition est remplie, on considère que :

$$a_n \approx a \quad (5.37)$$

$$\Pi_n \approx \Pi_r \quad (5.38)$$

$$P_{pn} \approx P_p$$

$$P_n \approx P$$

$$M_{cpn} \approx M_{cp} \quad (5.39)$$

En pratique, on a $\mathcal{S}_n < 10^{-7}$ pour $n = 3$. La figure 5.8 résume les principales étapes de cette procédure itérative.

5.3 Asservissement du suivi de profilé

Le contrôle de l'attitude de la caméra par rapport à la section courante du profilé est réalisé par un asservissement des coordonnées opérationnelles de la position courante de la caméra par rapport à sa position désirée. La référence de la boucle doit donc être nulle afin de garantir la coïncidence entre les positions courante et désirée de la caméra. La position désirée du repère caméra est remise à jour à chaque période d'échantillonnage et peut évoluer en fonction du mouvement du profilé et du déplacement de la section courante (la section du profilé qui sert de référence pour l'asservissement) le long du profilé. Elle est calculée grâce à une attitude de référence constante entre la caméra et la section courante qui est initialisée au début de l'asservissement.

5.3.1 La mesure de l'asservissement

La figure 5.9 donne un descriptif de la boucle d'asservissement. Les primitives visuelles sont extraites de l'image (voir section 5.1.4, page 90) et sont utilisées pour reconstruire la transformation M_{cp} entre le repère caméra, R_c et le repère lié à la section courante, R_p . Cette dernière est réactualisée à chaque période en fonction de V_f , la vitesse de suivi le long du profilé (voir section 5.2 page 97). Etant donné que l'information visuelle ne permet pas de reconstruire la position le long du profilé, le modèle géométrique du robot est utilisé. Il est calculé grâce aux mesures de position angulaires, q , provenant des codeurs incrémentaux des axes du robot.

Ensuite, M_{cp} est calculé grâce à la procédure décrite dans la section 5.1.5, page 91. La matrice homogène M_{cp}^* définit la transformation de référence entre la position

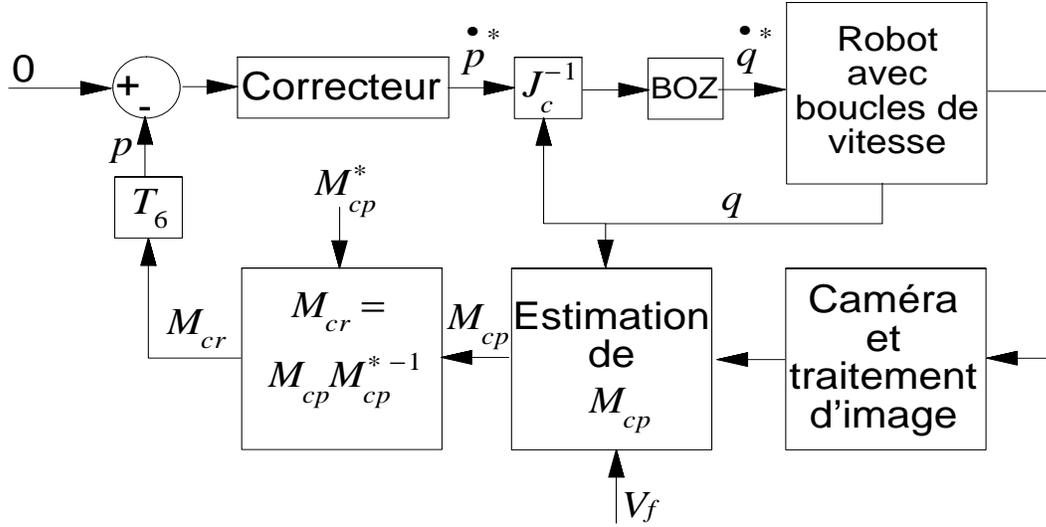


Fig. 5.9: Schéma-bloc de l'asservissement de suivi de profilé

du repère caméra et le repère lié à la section courante. Elle est initialisée au début de l'asservissement et correspond à la transformation entre la caméra et la section du profilé dont la projection est au milieu de l'écran. Cette situation initiale correspond donc à $a = 0$ dans l'équation (5.18), page 95.

Grâce à M_{cp}^* il est possible de calculer l'attitude du repère caméra désiré R_r par rapport au repère caméra courant R_c . Celle-ci est exprimée par la matrice homogène M_{cr} (cf. figure 5.10) :

$$M_{cr} = M_{cp} M_{cp}^{*-1} \quad (5.40)$$

Ainsi, lorsque $M_{cr} = I$, $M_{cp} = M_{cp}^*$. Cette transformation M_{cr} est entièrement définie par 6 coordonnées opérationnelles : 3 pour la translation et 3 pour la rotation. Afin de réaliser un asservissement qui soit commandable, il est impératif de convertir M_{cr} en ces 6 paramètres. La rotation peut être décomposée en une infinité de compositions de 3 rotations élémentaires. Nous avons choisi la décomposition en tangage, roulis, lacet. Soit T_6 , la transformation effectuant cette conversion (voir annexe B.3 pour la définition de T_6) :

$$p = \begin{bmatrix} T_x \\ T_y \\ T_z \\ \Theta_p \\ \Theta_r \\ \Theta_y \end{bmatrix} = T_6(M_{cc}) \quad (5.41)$$

Le vecteur p exprime donc la mesure de la position du repère caméra désiré par rapport au repère caméra courant. La référence de l'asservissement doit donc être nulle si on veut garantir la convergence de M_{cp} vers M_{cp}^* .

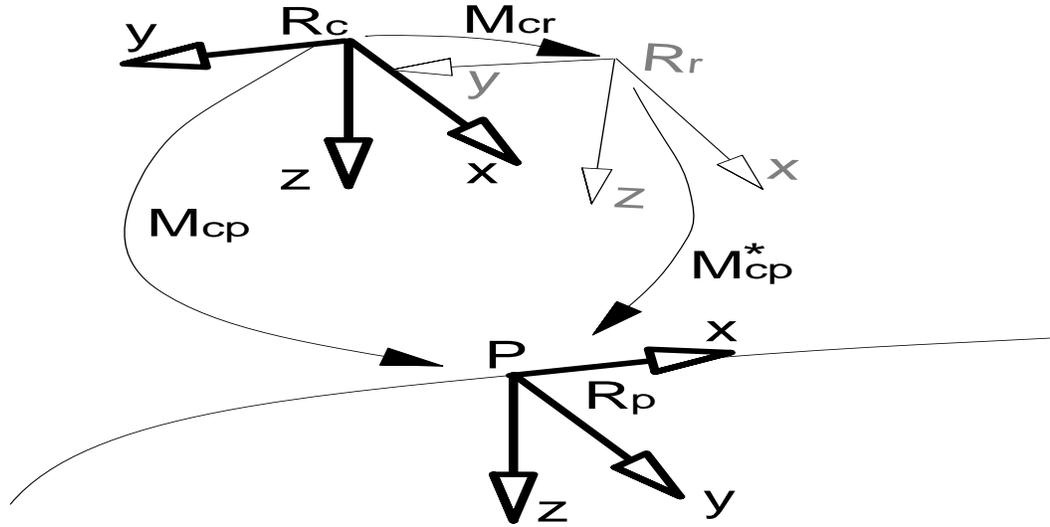


Fig. 5.10: Détermination de M_{cr}

5.3.2 Le contrôle de la boucle

Le contrôleur de la boucle fournit des commandes de vitesse des coordonnées opérationnelles \dot{p}^* . Celles-ci sont converties en références de vitesse \dot{q}^* pour les contrôleurs de vitesses articulaires grâce au Jacobien de la caméra J_c . Ce Jacobien est décrit dans la section 3.1.1.

La fonction de transfert $H(z)$ entre \dot{p}^* et p a été calculée dans la section 3.1.2. Elle modélise le comportement dynamique de ce que nous avons appelé le dispositif de déplacement opérationnel virtuel.

Deux types de contrôleurs ont été utilisés pour cette boucle : un correcteur à action Proportionnelle-Intégrale (PI) et un correcteur GPC (voir section 3.2). Le PI est ajusté de manière à obtenir les performances dynamiques maximales tout en préservant la stabilité du système. Le GPC quant à lui est obtenu par optimisation d'une fonction de coût qui prend en compte les dynamiques du manipulateur. Ces dynamiques sont celles du dispositif de déplacement opérationnel virtuel. Le modèle dynamique linéarisé complet de la boucle de vision est décrit dans la section 3.1.3.

5.4 Suivi de profilé avec prédiction de la trajectoire

L'image d'un profilé peut être exploitée afin d'en tirer une information quant à l'évolution future de la courbure le long de la trajectoire. En effet, le suivi s'effectuant à une vitesse constante, il est possible de calculer les positions des n sections du profilé qui se projettent dans l'image qui serviront de référence pour les n pas d'échantillonnage futurs. Néanmoins, toute la structure de l'asservissement doit être modifiée en conséquence.

5.4.1 Comparaison des 2 approches d'asservissement : l'approche classique et l'approche prédictive

Dans l'approche présentée dans la section 5.3, le signal de référence de l'asservissement est nul. La mesure est la position désirée du repère caméra exprimée dans le repère caméra actuel. La position désirée du repère caméra se déplace parallèlement au profilé de sorte que la transformation entre ce repère et la section courante soit égale à une transformation de référence M_{cp}^* . Cette modification de la position du repère caméra de référence agit donc comme une perturbation sur la boucle de vision. Cette perturbation peut être modélisée par un signal rampe, étant donné que la vitesse de déplacement est constante. Comme les correcteurs PI et GPC contiennent un intégrateur, et que le transfert entre consigne de vitesse et déplacement du robot contient également intrinsèquement un intégrateur, il y a annulation de cette perturbation par l'asservissement. Le repère caméra courant tend donc vers le repère caméra désiré si le signal de consigne de l'asservissement est nulle.

Néanmoins, une telle approche ne convient pas à un asservissement prédictif. En effet, il est nécessaire que ce soit une variation du signal de référence qui provoque le mouvement et non que ce dernier soit engendré par une perturbation. Il faut donc modifier la boucle d'asservissement, et plus particulièrement les mesures, de manière à ce que le signal de référence traduise la trajectoire de référence à effectuer pour le suivi. Les mesures doivent donc obligatoirement être exprimées dans un repère de référence fixe par rapport à la base du robot. Pour cela, il faut faire intervenir, dans le calcul de la mesure, le modèle géométrique du robot.

Comparée à l'approche classique, l'approche prédictive est donc plus complexe. Néanmoins, comme nous allons le voir, le fait de tenir compte des signaux de consigne futurs conduit à une amélioration de la précision de suivi.

5.4.2 Détermination des positions futures des sections courantes

Soient $P^{(1)}, P^{(2)}, \dots, P^{(N_2)}$, les prédictions des N_2 positions futures de la section courante le long de la courbe de référence. Le paramètre N_2 définit l'horizon de prédiction du correcteur GPC (*cf.* section 3.2).

La procédure qui permet d'obtenir les points $P^{(i)}$ est décrite par la figure 5.11. Cette procédure débute au point P qui est à l'origine du repère R_p lié à la section courante. Les points $P^{(i)}$ sont à l'intersection entre les plans des sections futures et la courbe de référence. Ces plans sont obtenus par translations successives suivant une direction tangente à la courbe de référence du profilé et d'une distance $V_f T_e$ comme le montre la figure 5.11. Pour chaque plan des sections futures, il est possible, grâce au calcul décrit dans la section 5.2, de déterminer les matrices homogènes $M_{cp}^{(1)}, M_{cp}^{(2)}, \dots, M_{cp}^{(N_2)}$. Ces matrices définissent les transformations entre le repère caméra courant et les repères $R_f^{(1)}, \dots, R_f^{(N_2)}$ associés aux N_2 positions futures de la section courante.

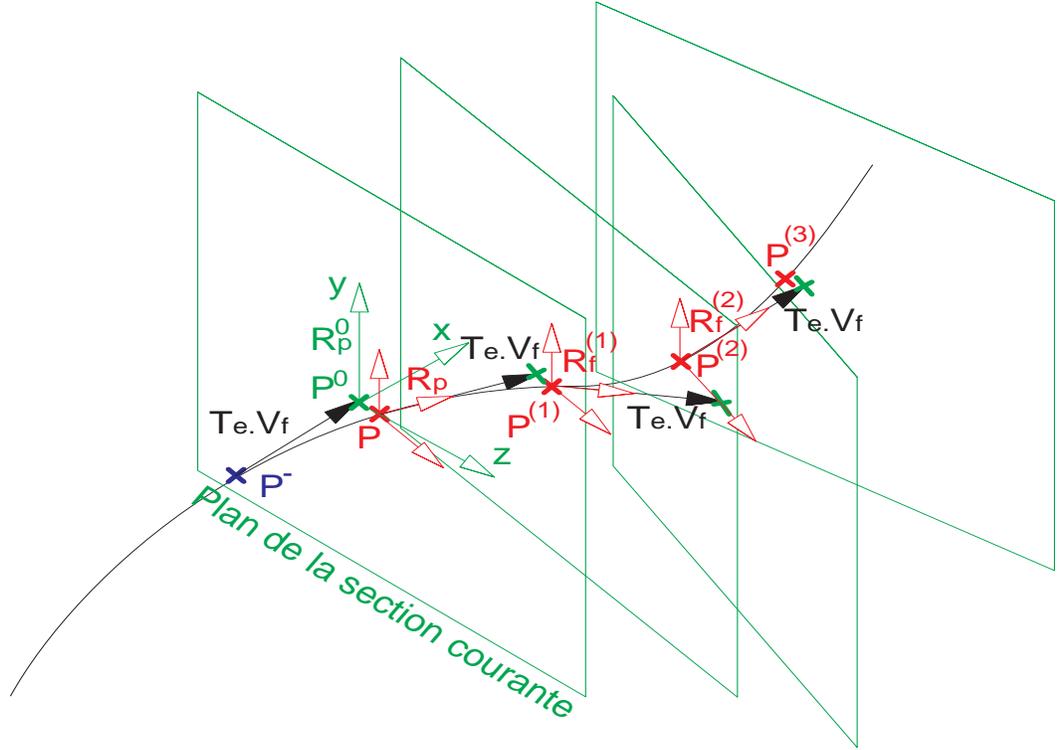


Fig. 5.11: Détermination des prédictions des positions de la section courante

Néanmoins, l'estimation de toutes les matrices homogènes n'est possible que si les projections de toutes les N_2 sections futures sont toutes contenues dans l'image.

5.4.3 Définition des mesures et des consignes

Soit M_{cp}^* la transformation de référence entre la caméra et la section courante. Les positions des repères $R_f^{(i)}$, $1 \leq i \leq N_2$, associés aux positions futures de la section courante sont définies par les matrices homogènes $M_{cp}^{(i)}$ par rapport au repère caméra courant R_c .

Exprimons par rapport à R_c , les positions $M_{cF}^{(i)}$ des repères $R_f^{(i)}$ qui sont à l'emplacement désiré du repère caméra aux N_2 instants d'échantillonnage futurs par rapport à R_c :

$$M_{cF}^{(i)} = M_{cf}^{(i)} \cdot M_{cp}^{*-1} \quad 1 \leq i \leq N_2 \quad (5.42)$$

Soit $M_{bc}(q)$, la matrice homogène définissant la position du repère caméra par rapport à la base du robot. Cette matrice est calculée grâce au modèle géométrique du robot ; elle dépend donc du vecteur des coordonnées articulaires q . Les positions des repères $R_f^{(i)}$ peuvent donc être exprimées dans le repère de base par les matrices

homogènes $M_{bF}^{(i)}$:

$$M_{bF}^{(i)} = M_{bc}(q) \cdot M_{cF}^{(i)} \quad (5.43)$$

On définit $p^{*(i)} = T_6(M_{bF}^{(i)})$, les vecteurs des coordonnées opérationnelles de référence pour la position future du repère caméra à l'instant $t + iT_e$ (t est l'instant courant). (voir annexe B.3 pour la définition de T_6). On désigne par p^* , le vecteur des N_2 attitudes $p^{*(i)}$.

Les mesures sont également définies par rapport au repère de base du robot. Il s'agit des coordonnées opérationnelles courantes p du repère caméra R_c par rapport à R_b :

$$p = T_6(M_{bc}(q)) \quad (5.44)$$

5.4.4 La boucle de contrôle

La figure 5.12 présente la structure de l'asservissement prédictif du suivi de profilé. Les consignes p^* de cet asservissement sont déterminées grâce à la mesure visuelle conformément à la procédure décrite dans la section précédente.

Afin de pouvoir contrôler les coordonnées opérationnelles de p , le contrôleur $C(z)$ applique des commandes \hat{p}^* . Ces commandes sont converties en références \hat{q}^* pour les boucles de contrôle des vitesses articulaires grâce au Jacobien de la caméra J_c défini dans la section 3.1.1. Dans ce cas particulier, le repère de référence fixe R_r est égal au repère de base du robot R_b et le repère lié à l'organe terminal R_t est égal au repère caméra R_c (configuration *eye in hand*).

Il existe une fonction de transfert $H(z)$ entre \hat{p}^* et p qui modélise les dynamiques du dispositif de déplacement opérationnel virtuel. Le calcul de cette fonction de transfert est donné dans la section 3.1.2. Ce modèle sert au calcul des 6 correcteurs GPC réalisant les asservissements des 6 coordonnées opérationnelles de p . Le modèle dynamique linéarisé complet de la boucle de vision est décrit dans le section 3.1.3.

Le correcteur GPC tient compte des N_2 consignes futures de p pour réaliser un contrôle prédictif du suivi. Celui-ci permet d'anticiper les réactions à l'évolution future de la courbure du profilé. Pour cela, on réalise donc $R = [p^{*(N_1)}, \dots, p^{*(N_2)}]^T$ dans l'équation 3.42 page 53 où N_1 est l'horizon d'initialisation du GPC.

5.5 Simulations et expériences

Le système expérimental de suivi de profilé est décrit par la figure 5.13. Le profilé est constitué de 3 lignes de 5mm d'épaisseur imprimées sur une bande de papier blanc de 21cm de largeur. La courbure de ce profilé peut être modifiée par manipulation de la bande de papier (voir figure). Le profilé utilisé lors des expérimentations a la courbure visualisée sur la figure 5.13. Pour donner un ordre de grandeur, la variation de hauteur du profilé est de 7 cm.

Les essais de suivi s'effectuent de gauche vers la droite par rapport à la figure 5.13. L'attitude de référence définie par M_{cp}^* correspond à celle décrite par la figure.

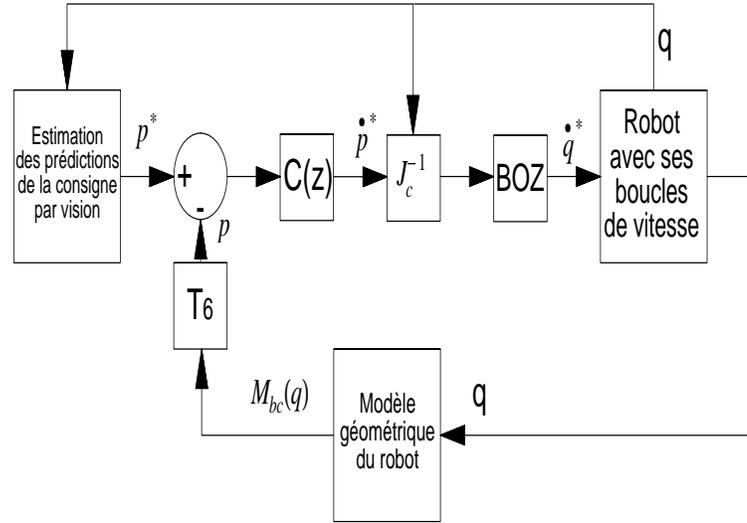


Fig. 5.12: Schéma général de l'asservissement prédictif du suivi de profilé

La caméra est à une distance de 30 cm du profilé. Lors des essais, le suivi s'effectue dans un premier temps suivant la partie plane du profilé, puis le long de la courbure imprimée au profilé. Dans les enregistrements, on peut donc distinguer ces 2 parties.

Etant donnée la courbure du profilé, la tâche de suivi provoque des perturbations essentiellement sur 3 des 6 coordonnées opérationnelles (voir figure 5.5.2 pour la définition des axes du repère caméra) :

1. T_z car le robot doit réaliser un mouvement de translation vers le haut pour suivre le profilé.
2. θ_t qui est la coordonnée de tangage correspondant à une rotation autour de l'axe x . Pour maintenir l'attitude constante entre la section courante et la caméra, le robot doit effectuer un mouvement de rotation suivant cet axe.
3. T_y , car l'axe y est dans la direction du suivi.

Tous les essais ont été réalisés à une période d'échantillonnage de 20ms. Cette période correspond à la durée d'une trame avec une caméra au standard CCIR.

Dans cette section nous appelons GPC1, la stratégie de contrôle de la boucle de vision par correcteur GPC qui ne tient pas compte des prédictions des consignes. Cette stratégie est décrite dans la section 5.3. Nous désignons par GPC2, la stratégie de contrôle prédictif présentée dans la section 5.4. Lorsque le correcteur est un PI, la contrôle est nécessairement non prédictif et donc conforme à ce qui est présenté dans la section 5.3.

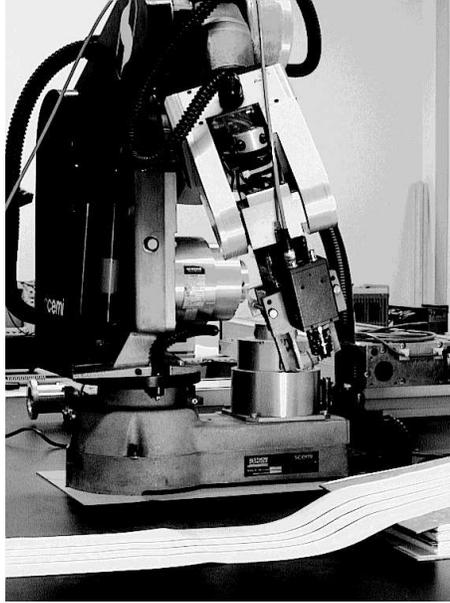


Fig. 5.13: Description de la manipulation de suivi de profilé

5.5.1 Comparaison des stratégies GPC1 et PI

Simulation

Nous avons simulé l'expérience de suivi de profilé. Le modèle de profilé utilisé est représenté sur la figure 5.5. Grâce à la simulation, il est possible d'extrapoler les performances d'un asservissement visuel dont la cadence d'échantillonnage serait 10 fois plus importante.

La figure 5.14 présente les enregistrements comparés des erreurs suivant la coordonnée opérationnelle θ_t pour un suivi simulé à la vitesse $V_f = 10\text{cm/s}$. Le contrôle de la boucle avec le PI est réalisé à la période d'échantillonnage de 20ms. Le contrôle de la boucle par la stratégie GPC1 est réalisée pour 2 périodes d'échantillonnage différentes : 20ms et 2ms.

Cette figure montre la supériorité du contrôle avec un correcteur GPC tenant compte des dynamiques du manipulateur sur le contrôle avec un correcteur PI réglé expérimentalement. De plus, le fait d'augmenter la cadence d'échantillonnage dans un rapport de 10, augmente très nettement la précision du suivi pour la stratégie GPC1. Les paramètres de réglage de ce correcteur sont les mêmes aux deux périodes d'échantillonnage : $N_1 = 1$, $N_2 = 30$, $N_u = 6$ et $\lambda = 0.1$. Ce dernier résultat montre qu'une amélioration de la rapidité du système de vision peut améliorer de manière très nette les performances dynamiques du suivi.

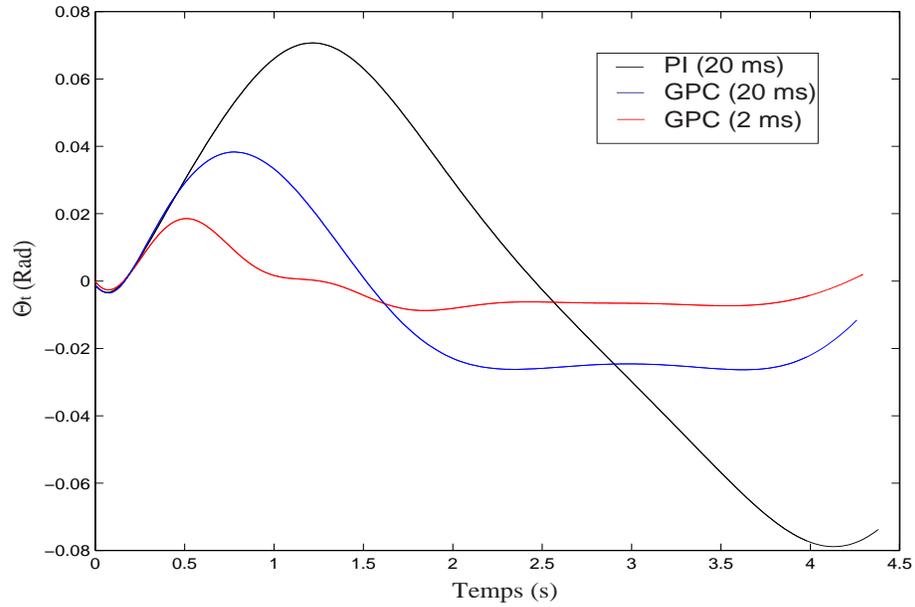


Fig. 5.14: Simulation des stratégies GPC1 et PI pour un suivi à 10 cm/s

Expérience

La figure 5.15 présente les enregistrements des erreurs suivant T_y , T_z et θ_t en fonction du temps pour un suivi de profilé à la vitesse $V_f = 2\text{cm/s}$. On distingue nettement sur ces courbes les 2 parties correspondant à la zone plane et à la zone incurvée du profilé. Cette figure démontre clairement la supériorité de la stratégie GPC1 sur la stratégie PI : l'intégrale de l'erreur au carré est nettement inférieure avec le contrôle GPC1 qu'avec le contrôle PI. Les paramètres de réglage du correcteur GPC sont les suivants : $N_1 = 1$, $N_2 = 30$, $N_u = 6$ et $\lambda = 0.1$.

On peut noter qu'avec le contrôle GPC1, la précision de suivi pour la coordonnée T_z est d'environ 2 dixièmes de millimètre ; pour la coordonnée θ_t , cette précision est de l'ordre de $5 \cdot 10^{-3}$ Rad.

5.5.2 Etude de la précision de suivi en fonction de la vitesse

Les enregistrements présentés par les figures 5.16 et 5.17 montrent respectivement l'évolution des erreurs suivant T_z et θ_t en fonction de la vitesse de suivi V_f .

On observe qu'il existe une relation sensiblement linéaire entre la vitesse de suivi et l'erreur maximale. Par exemple, pour la coordonnée θ_t , l'erreur maximale pour un suivi à 1 cm/s est de 0.006 Radians et pour un suivi à 5 cm/s, elle est de 0.033 Radians.

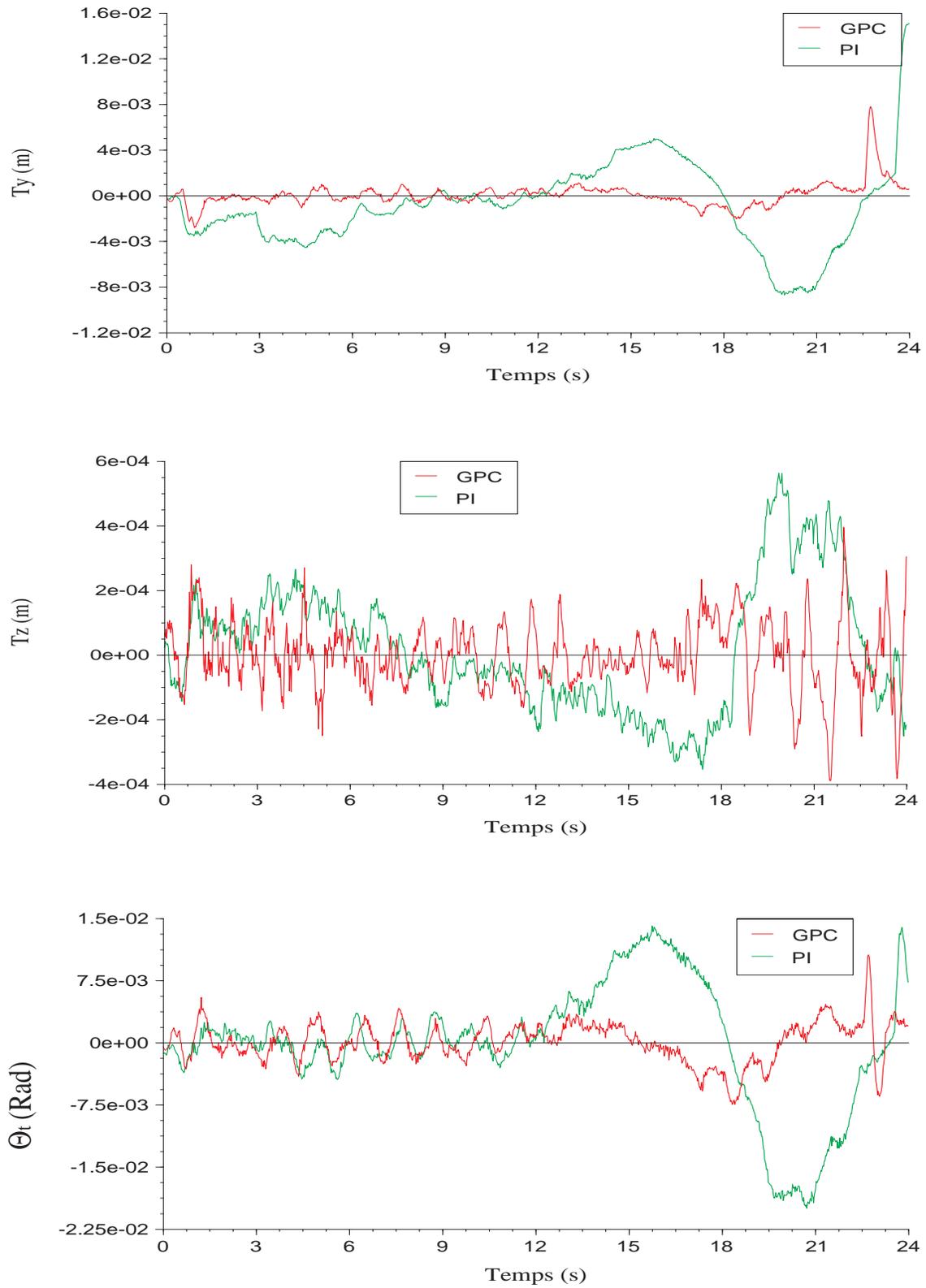


Fig. 5.15: Comparaison des stratégies GPC1 et PI pour un suivi à 2 cm/s

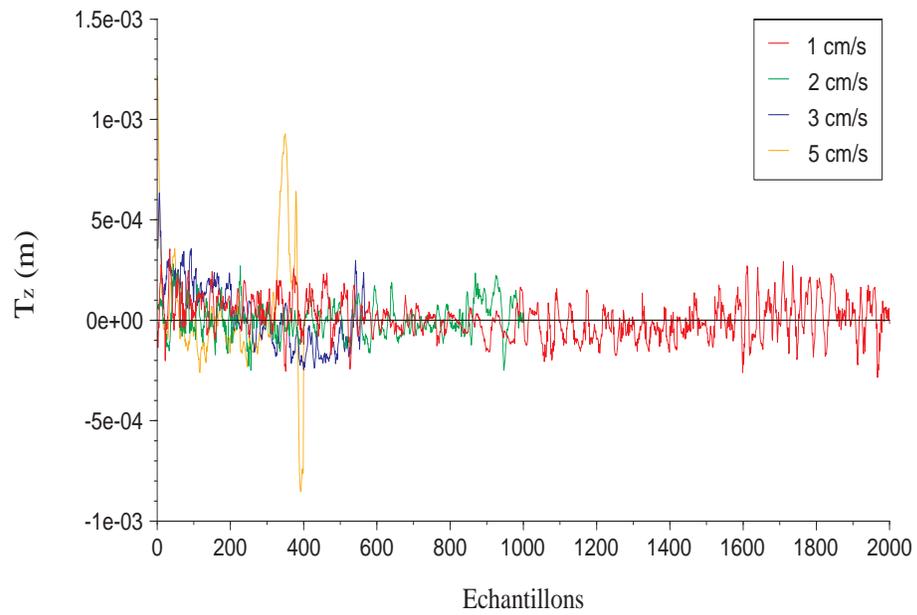


Fig. 5.16: Mesure de T_z en fonction pour différentes vitesses de suivi – Stratégie GPC1

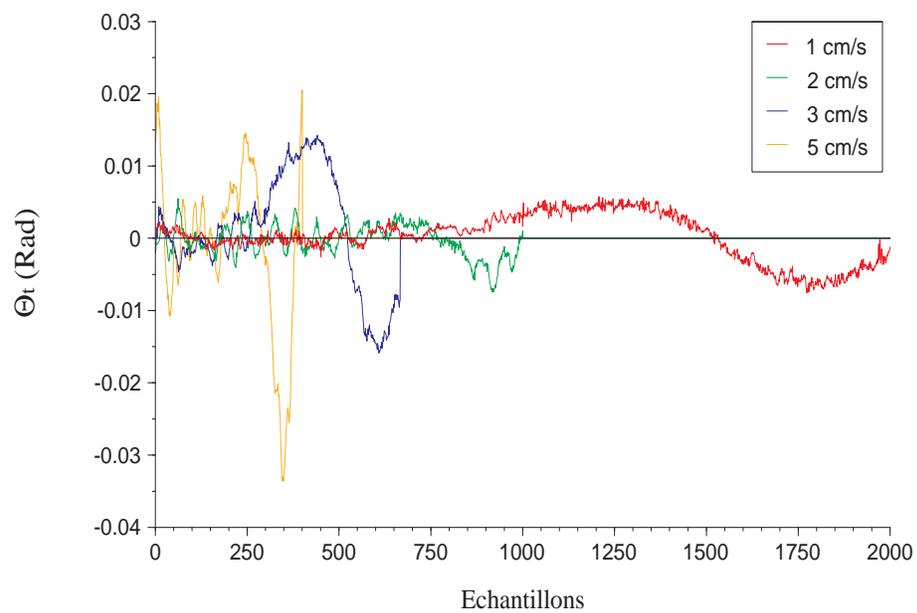


Fig. 5.17: Mesure de θ_t en fonction pour différentes vitesses de suivi – Stratégie GPC1

5.5.3 Comparaison des stratégies GPC1 et GPC2

Afin de réaliser une comparaison de ces 2 approches, il est nécessaire d'exprimer les mesures par rapport au même repère. Dans l'approche GPC1, les mesures sont exprimées dans le repère caméra courant tandis qu'avec l'approche GPC2, celles-ci sont exprimées par rapport au repère de base du robot.

Dans cette section, toutes les mesures ont été exprimées dans le repère caméra courant. Pour les 2 stratégies de contrôle, on donne les mesures sous forme de coordonnées opérationnelles du repère caméra désiré par rapport au repère caméra courant.

Simulation

La figure 5.18 montre les enregistrements des erreurs suivant T_y , T_z et θ_t pour une simulation du suivi à la vitesse de 8 cm/s. Une comparaison des stratégies de contrôle GPC1 (approche standard) et GPC2 (approche prédictive) est faite sur cette figure. Elle montre clairement la supériorité de l'approche prédictive : l'intégrale du carré de l'erreur est nettement plus faible.

Expérience

Les figures 5.19 et 5.20 montrent la comparaison expérimentale des erreurs suivant les 6 coordonnées d'attitude pour les approches GPC1 et GPC2. La vitesse de suivi est de 8 cm/s. Les paramètres de réglages du GPC sont identiques pour le contrôle de type GPC1 et GPC2 : $N_1 = 1$, $N_2 = 30$, $N_u = 6$ et $\lambda = 0.1$.

On constate immédiatement que, même si l'intégrale de l'erreur au carré est plus faible dans le contrôle GPC2 que dans le contrôle GPC1, l'amélioration n'est pas aussi marquante que le laissait envisager la simulation. De plus, on observe un biais très nette dans les erreurs suivant θ_t et T_y ; celui-ci existe aussi de façon moins prononcée suivant T_z .

Nous avons fait l'hypothèse que ce biais pourrait provenir de la mauvaise estimation des paramètres intrinsèques de la caméra. En effet, la détermination des N_2 valeurs des consignes futures p^* est réalisée à partir des informations visuelles qui donnent l'évolution future de la courbure du profilé. Si les paramètres intrinsèques de la caméra sont connus de manière imprécise, ces prédictions de consigne p^* sont donc biaisées. Ceci peut expliquer le biais observé sur les enregistrements de θ_t , T_y et T_z . En effet, ces 3 coordonnées sont celles qui varient le plus lors du suivi.

Afin de conforter cette hypothèse, nous avons reproduit ce phénomène en simulation. La figure 5.21, permet de comparer la simulation de l'erreur suivant θ_t dans le cas où une erreur de 5% est commise sur l'estimation des paramètres de grandissement G_x et G_y de la caméra au cas idéal. Il apparaît clairement que cette erreur d'estimation provoque une détérioration de la précision du suivi. Nous en déduisons que l'approche GPC2 permet d'obtenir un meilleur suivi à condition de réaliser une bonne calibration des paramètres intrinsèques de la caméra.

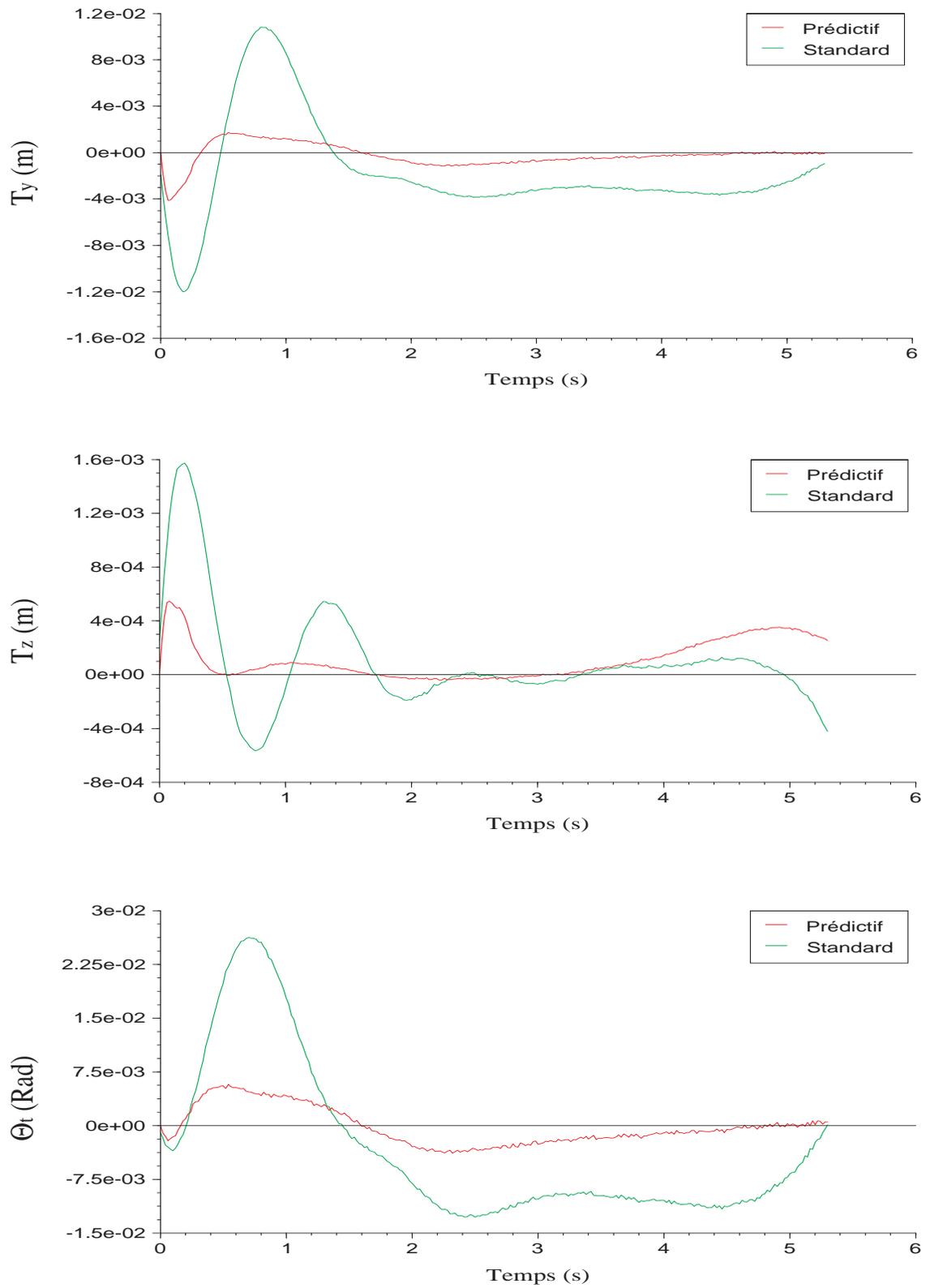


Fig. 5.18: Comparaison des stratégies GPC1 et GPC2 en simulation

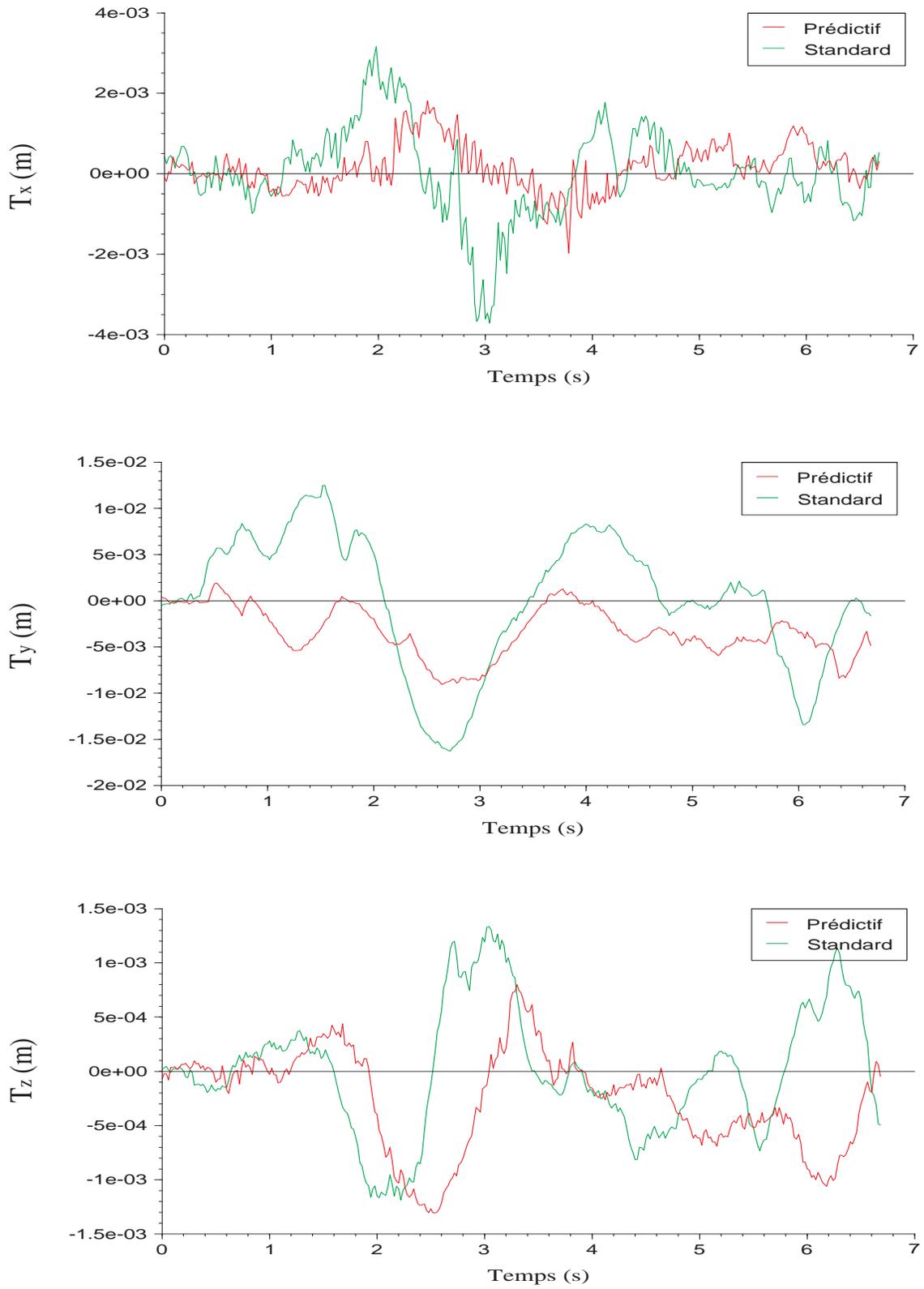


Fig. 5.19: Comparaison expérimentale des stratégies GPC1 et GPC2

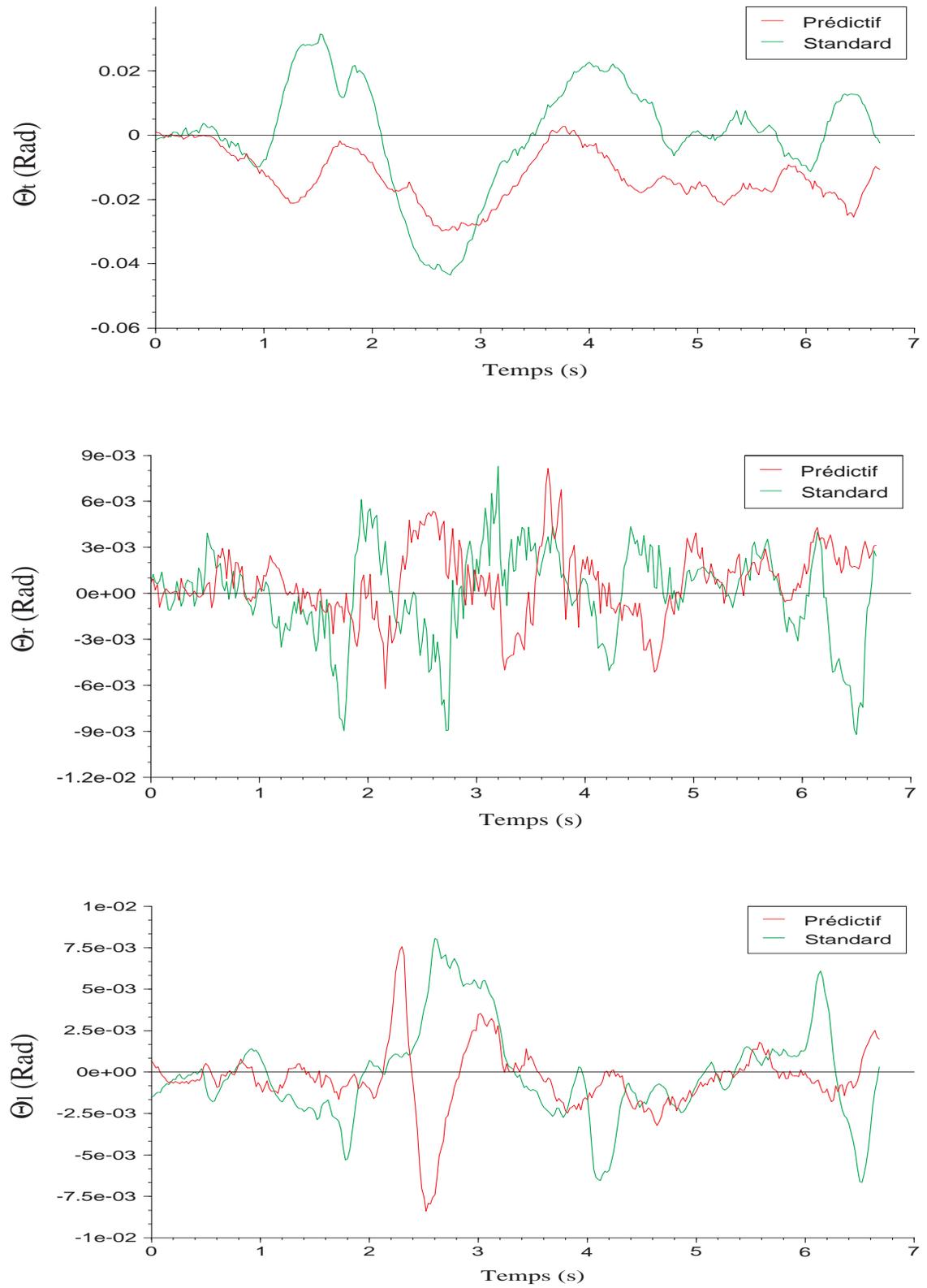


Fig. 5.20: Comparaison expérimentale des stratégies GPC1 et GPC2

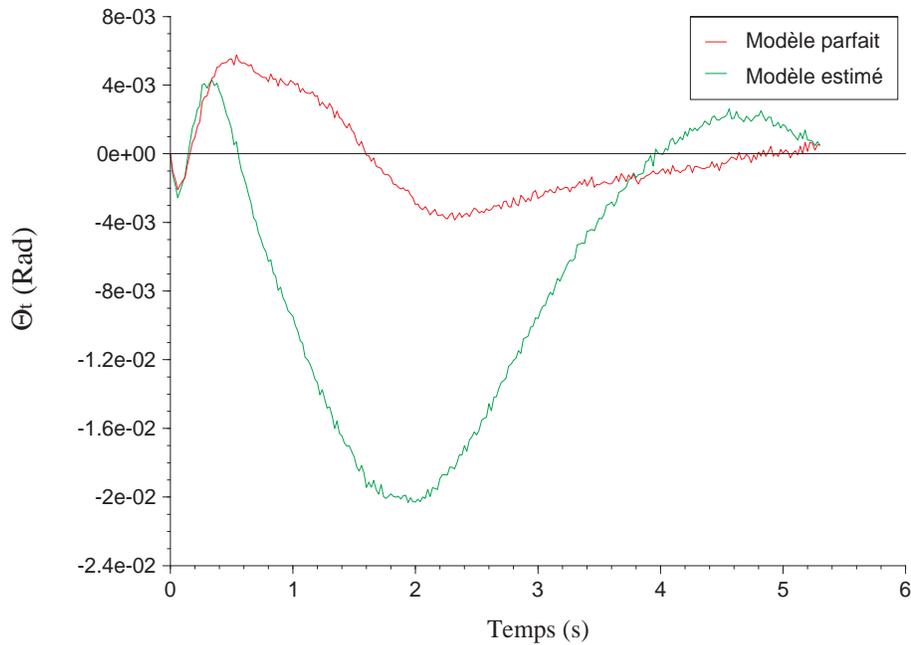


Fig. 5.21: Etude simulée de l'influence d'une mauvaise estimation du grandissement de la caméra – Approche GPC2

5.6 Conclusion

Dans ce chapitre nous décrivons une application d'asservissement visuel qui consiste à réaliser un suivi à vitesse constante le long d'un profilé suivant 6 degrés de liberté.

Cet asservissement est du type «3D» car la consigne est exprimée sous la forme d'une attitude de référence entre la caméra et une section du profilé. Nous décrivons une méthode originale d'estimation de cette attitude grâce à une combinaison d'informations extraites de l'image et d'informations provenant des capteurs de position articulaire du manipulateur. Un modèle géométrique de la section du profilé est requis pour réaliser cette estimation, mais la courbure de profilé est inconnue.

Nous avons testé 2 stratégies de contrôle pour la boucle de vision :

1. Une stratégie standard ne tenant pas compte de l'évolution future de la courbure du profilé.
2. Une stratégie prédictive où la mesure de l'évolution future de la courbure du profilé dans l'image sert à calculer des références futures pour un contrôleur GPC.

Pour la première stratégie, nous avons comparé un correcteur à action proportionnelle intégrale (PI) dont les paramètres ont été réglés de manière expérimentale à un correcteur GPC dont le réglage tient compte des dynamiques du manipulateur conformément au modèle que nous avons proposé. Les expériences ont permis de montrer la supériorité du contrôleur GPC par rapport au contrôleur PI classique,

validant de ce fait notre modèle dynamique.

Nous avons ensuite comparé l'approche prédictive à l'approche standard. Les expériences ont mis en évidence une amélioration de la précision du suivi grâce à la prise en compte des prédictions. Néanmoins, il a été établi qu'une mauvaise calibration de la caméra entraîne une détérioration de la réponse dans le cas de l'approche prédictive.

Chapitre 6

Conclusion générale

6.1 Contributions principales

L'augmentation des performances des systèmes de vision, tant du point de vue de la cadence d'acquisition que de la vitesse de traitement, a permis l'émergence d'asservissements visuels dits «rapides».

Ces nouveaux asservissements, contrairement à leurs aînés, permettent de tirer partie de toutes les possibilités d'un robot manipulateur en terme d'accélération et de vitesse. Néanmoins, afin d'obtenir un contrôle optimal, il est nécessaire de modéliser précisément les dynamiques du manipulateur impliqué dans la boucle de vision.

Dans ce travail, nous avons proposé une modélisation dynamique d'un robot à 6 degrés de liberté impliqué dans un asservissement visuel qui est basée sur la notion de *dispositif de déplacement opérationnel virtuel*. Partant du constat que la grandeur asservie peut toujours s'exprimer en terme de coordonnées opérationnelles (ou coordonnées d'attitude), notre modèle dynamique réalise une projection des dynamiques articulaires du robot dans l'espace de ces coordonnées opérationnelles.

Cette modélisation a été appliquée à deux études d'asservissement visuel.

La première est un suivi de cible. Il s'agit d'asservir à une constante l'attitude entre une caméra fixée sur l'organe terminal d'un robot et une cible située dans le champ de vision de cette caméra. L'asservissement est réalisé suivant 6 degrés de liberté : tout mouvement de la cible provoque un mouvement de la caméra et du robot de manière à conserver l'attitude de référence.

Deux stratégies d'asservissement ont été expérimentées : l'approche 3D qui définit la référence par une attitude et l'approche 2D qui définit la référence par l'image désirée de la cible. Pour l'approche 2D, nous avons proposé une méthode qui permet de convertir les informations dans le plan image en 6 coordonnées opérationnelles dans le cadre d'une tâche de suivi. Cette transformation permet de modéliser dynamiquement les asservissements 2D et 3D de la même manière *via* le dispositif de déplacement opérationnel virtuel.

Nous avons comparé expérimentalement les performances de la boucle de vision avec deux types de correcteurs : un correcteur classique à action proportionnelle-intégrale (PI) réglé de manière empirique et un correcteur GPC dont le réglage dépend du modèle dynamique que nous proposons. La supériorité du contrôleur GPC en terme de rapidité a permis de confirmer de manière formelle la validité de notre modélisation dynamique.

Les expériences ont également mis en lumière la supériorité de l'approche 2D par rapport à l'approche 3D. En effet, cette dernière conduit à un asservissement plus instable car l'estimation de la mesure est notablement plus bruitée.

La seconde application d'asservissement visuel que nous avons mise en oeuvre est un suivi de profilé. La tâche consiste à réaliser le suivi à vitesse constante d'un profilé dont la section est connue mais dont la courbure est inconnue. L'asservissement visuel permet de conserver l'attitude entre la caméra et une section du profilé. Nous avons proposé une méthode de détermination de cette attitude. Il s'agit d'un asservissement 3D car la référence est exprimée sous la forme d'une attitude.

Nous avons réalisé 2 structures de contrôle de la boucle de vision. L'une a une structure standard et l'autre réalise un contrôle prédictif du suivi en fonction de la mesure dans l'image de l'évolution future de la courbure du profilé. Cette dernière structure fait appel aux caractéristiques prédictives du GPC. Pour les deux structures, la modélisation dynamique de la boucle de vision fait intervenir le dispositif de déplacement opérationnel virtuel que nous avons introduit.

Nous avons dans un premier temps comparé les performances de la première structure d'asservissement avec les contrôleurs PI et GPC. Ces expériences ont confirmé la supériorité du contrôle GPC tenant compte du modèle dynamique sur le contrôle PI réglé expérimentalement. Dans un deuxième temps, nous avons comparé la structure standard à la structure prédictive, toutes deux contrôlées par un correcteur GPC. Les simulations ont mis en évidence une nette amélioration de la précision de suivi grâce à l'approche prédictive. Ces résultats ont été confirmés par les expériences mais dans une mesure moindre de ce que la simulation laissait présager.

Aussi nous avons entrepris de déterminer la cause de cette dégradation. Nous avons supposé que cette dégradation provenait de la mauvaise estimation des paramètres intrinsèques de la caméra. Une simulation de ce phénomène a permis de confirmer cette hypothèse. Nous en avons conclu que le suivi prédictif de profilé requerrait une bonne calibration de la caméra.

6.2 Perspectives

Ce travail possède de nombreuses extensions possibles. Nous en avons retenu 3 qui paraissent être plus fondamentales.

Dans le contrôle de la boucle, nous avons toujours négligé les couplages entre les différentes coordonnées opérationnelles du dispositif de déplacement opérationnel virtuel. Or ces couplages existent : des essais expérimentaux le montrent clairement. De plus ils ne sont pas négligeables. La prise en compte de ces couplages implique

un contrôle multivariable du système de vision. Ce contrôle peut être envisagé avec une version multivariable du GPC.

Nous avons montré dans ce travail la supériorité de l'approche 2D sur l'approche 3D pour une application de suivi. Néanmoins, cette étude se limite à une seule méthode d'estimation 3D de la mesure. Une étude comparative plus détaillée entre les 2 stratégies peut être envisagée ; elle comparerait toutes les techniques d'estimation 3D à l'approche 2D.

Nous avons mis en évidence une faiblesse du suivi de profilé prédictif dont la précision dépend de la calibration de la caméra. Nous proposons de mener une recherche pour rendre l'algorithme de contrôle prédictif robuste aux erreurs de modèle de la caméra.

ANNEXE

Annexe A

DÉMONSTRATIONS

A.1 Démonstration du lemme 5.1

Soit \mathcal{C} une courbe dérivable dans \mathbb{R}^3 définie par :

$$\mathcal{C} = \mathcal{C}(t) = \begin{pmatrix} F_x(t) \\ F_y(t) \\ F_z(t) \end{pmatrix} \quad (\text{A.1})$$

où t est le paramètre de la courbe. Soit P_0 un point appartenant à \mathcal{C} :

$$P_0 = P_0(t_0) = \begin{pmatrix} F_x(t_0) \\ F_y(t_0) \\ F_z(t_0) \end{pmatrix} \quad (\text{A.2})$$

La tangente à la courbe en P_0 est définie par P_0 et le vecteur tangent normalisé $\vec{\mathcal{D}}_0$ avec :

$$\vec{\mathcal{D}}_0 = \frac{1}{d_0} \begin{pmatrix} F'_x(t_0) \\ F'_y(t_0) \\ F'_z(t_0) \end{pmatrix} \quad (\text{A.3})$$

et

$$d_0 = \sqrt{F'^2_x(t_0) + F'^2_y(t_0) + F'^2_z(t_0)} \quad (\text{A.4})$$

Les points P appartenant à la tangente \mathcal{T} sont définis par $P = P_0 + \alpha \vec{\mathcal{D}}_0$ avec $\alpha \in \mathbb{R}$. Soit $P_1 \neq P_0$ un autre point de \mathcal{T} , $P_1 = P_0 + \alpha_1 \vec{\mathcal{D}}_0$ avec $\alpha_1 \neq 0$. Les projections perspectives de P_0 et P_1 dans le plan image d'une caméra de longueur focale f sont données par :

$$\mathcal{P}(P_1) = \begin{pmatrix} F_x(t_0) + \frac{\alpha_1}{d_0} F'_x(t_0) \\ F_y(t_0) + \frac{\alpha_1}{d_0} F'_y(t_0) \end{pmatrix} \frac{1}{\left(F_z(t_0) + \frac{\alpha_1}{d_0} F'_z(t_0) \right) f} \quad (\text{A.5})$$

$$\mathcal{P}(P_0) = \begin{pmatrix} F_x(t_0) \\ F_y(t_0) \end{pmatrix} \frac{1}{F_z(t_0) f} \quad (\text{A.6})$$

Ainsi, la projection perspective de \mathcal{T} dans le plan image, $\mathcal{T}_p = \mathcal{P}(\mathcal{T})$ est définie par la projection $\mathcal{P}(P_0)$ du point P_0 et par le vecteur directeur normalisé $\vec{\mathcal{D}}_p$:

$$\vec{\mathcal{D}}_p = \frac{1}{f} \begin{pmatrix} \frac{F_x(t_0) + \frac{\alpha_1}{d_0} F'_x(t_0)}{F_z(t_0) + \frac{\alpha_1}{d_0} F'_z(t_0)} - \frac{F_x(t_0)}{F_z(t_0)} \\ \frac{F_y(t_0) + \frac{\alpha_1}{d_0} F'_y(t_0)}{F_z(t_0) + \frac{\alpha_1}{d_0} F'_z(t_0)} - \frac{F_y(t_0)}{F_z(t_0)} \end{pmatrix} \times \frac{1}{d_p} \quad (\text{A.7})$$

où d_p est tel que $\|\vec{\mathcal{D}}_p\| = 1$.

$$\vec{\mathcal{D}}_p = \frac{\alpha_1}{f d_p d_0} \frac{1}{F_z^2(t_0) + \frac{\alpha_1}{d_0} F_z(t_0) f'_z(t_0)} \begin{pmatrix} F_z(t_0) F'_x(t_0) - F_x(t_0) F'_z(t_0) \\ F_z(t_0) F'_y(t_0) - F_y(t_0) F'_z(t_0) \end{pmatrix} \quad (\text{A.8})$$

Finalement, la pente de \mathcal{T}_p , projection de \mathcal{T} est égale à :

$$K_p = \frac{F'_y(t_0) F_z(t_0) - F_y(t_0) F'_z(t_0)}{F'_x(t_0) F_z(t_0) - F_x(t_0) F'_z(t_0)} \quad (\text{A.9})$$

D'autre part, soit $\mathcal{C}_p = \mathcal{P}(\mathcal{C})$ la projection de \mathcal{C} dans le plan image de la caméra :

$$\mathcal{C}_p = \begin{pmatrix} \frac{F_x(t)}{F_z(t)} \frac{1}{f} \\ \frac{F_y(t)}{F_z(t)} \frac{1}{f} \end{pmatrix} \quad (\text{A.10})$$

Le vecteur tangent à cette courbe en t_0 est donné par :

$$\vec{\mathcal{D}}_c = \begin{pmatrix} \frac{\frac{1}{f} \frac{F'_x(t_0) F_z(t_0) - F_x(t_0) F'_z(t_0)}{F_z^2(t_0)}}{\frac{1}{f} \frac{F'_y(t_0) F_z(t_0) - F_y(t_0) F'_z(t_0)}{F_z^2(t_0)}} \end{pmatrix} \frac{1}{d_c} \quad (\text{A.11})$$

où d_c est tel que $\|\vec{\mathcal{D}}_c\| = 1$. Puisque $\vec{\mathcal{D}}_c$ a la même pente qu'en (A.9), $\vec{\mathcal{D}}_c = \vec{\mathcal{D}}_p$, ce qui prouve le théorème.

A.2 Démonstration des relations (5.2) et (5.3)

Soit $\mathcal{T}_p = \mathcal{P}(\mathcal{T})$, la projection perspective d'une droite \mathcal{T} dans le plan image d'une caméra de longueur focale f :

$$\mathcal{T} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{R_c} = k \vec{V} + P = k \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}_{R_c} + \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}_{R_c} \quad (\text{A.12})$$

$$\mathcal{T}_p = \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix}_{R_c} = k_p \begin{pmatrix} u_x \\ u_y \\ 0 \end{pmatrix}_{R_c} + \begin{pmatrix} o_x \\ o_y \\ f \end{pmatrix}_{R_c} \quad (\text{A.13})$$

où k et k_p sont des scalaires paramétrisant \mathcal{T} et \mathcal{T}_p , \vec{V} est le vecteur directeur unitaire (*i.e.* $\|\vec{V}\| = 1$) de \mathcal{T} et P est un point de \mathcal{T} . Ces équations sont exprimées dans un

repère R_c lié à la caméra (*cf.* section 2.1.2, page 20, pour la définition de R_c). Les relations de projection perspective nous donnent :

$$\begin{aligned} x_p &= f \frac{x}{z} \\ y_p &= f \frac{y}{z} \end{aligned} \quad (\text{A.14})$$

En utilisant (A.12) et (A.13), on en déduit que :

$$k_p u_x + o_x = f \frac{kv_x + r_x}{kv_z + r_z} \quad (\text{A.15})$$

$$k_p u_y + o_y = f \frac{kv_y + r_y}{kv_z + r_z} \quad (\text{A.16})$$

En multipliant l'équation (A.15) par u_y et l'équation (A.16) par u_x , et en soustrayant ces 2 équations, on élimine le paramètre k_p :

$$o_x u_y - o_y u_x = f \frac{u_y(kv_x + r_x) - u_x(kv_y + r_y)}{kv_z + r_z} \quad (\text{A.17})$$

En mettant k en évidence dans cette expression, nous obtenons :

$$\begin{aligned} & r_z(o_y u_x - o_x u_y) - f(r_y u_x - r_x u_y) \\ & + k[v_z(o_y u_x - o_x u_y) - f(v_y u_x - v_x u_y)] = 0 \end{aligned} \quad (\text{A.18})$$

Comme cette équation doit être vérifiée pour tout k , nous en déduisons que :

$$f u_y r_x - f u_x r_y + (o_y u_x - o_x u_y) r_z = 0 \quad (\text{A.19})$$

$$f u_y v_x - f u_x v_y + (o_y u_x - o_x u_y) v_z = 0 \quad (\text{A.20})$$

Annexe B

RÉSULTATS ET CALCULS

B.1 Modèle géométrique du SCEMI

$$M_{01} = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{R_0} \quad (\text{B.1})$$

$$M_{12} = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin q_2 & \cos q_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{R_1} \quad (\text{B.2})$$

$$M_{23} = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & d_3 \\ -\sin q_3 & -\cos q_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{R_2} \quad (\text{B.3})$$

$$M_{34} = \begin{bmatrix} \cos q_4 & -\sin q_4 & 0 & 0 \\ 0 & 0 & 1 & r_4 \\ -\sin q_4 & -\cos q_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{R_3} \quad (\text{B.4})$$

$$M_{45} = \begin{bmatrix} \cos q_5 & -\sin q_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin q_5 & \cos q_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{R_4} \quad (\text{B.5})$$

$$M_{56} = \begin{bmatrix} \cos q_6 & -\sin q_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin q_6 & -\cos q_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{R_5} \quad (\text{B.6})$$

B.2 Fonctions de transfert en vitesse des axes du robot SCEMI

Les fonctions de transfert modélisant les boucles de vitesse du robot SCEMI ont été identifiées à l'ordre 6 pour la position nominale du robot (autour de laquelle tous les essais ont été réalisés). Elles ont la forme suivante :

$$F_i(z) = \frac{b_5 z^{-5} + b_4 z^{-4} + \dots + b_0}{z^{-6} + a_5 z^{-5} + \dots + a_0} \quad (\text{B.7})$$

Les tableaux B.1, B.2 et B.3 donnent les valeurs des coefficients des polynômes du numérateur et du dénominateur de ces fonctions de transfert pour les 6 axes indexés par l'entier i . La figure B.1 donne les réponses indicielles des 6 fonctions de transfert.

i	1	2
b_0	6.144337205368242e-02	-2.108051546582266e-01
b_1	-1.426793463863858e-03	1.132822664049553e+00
b_2	-6.156178260125461e-02	-2.440640516147716e+00
b_3	-6.655486455330641e-02	2.635220951217815e+00
b_4	2.176636953524022e-02	-1.425997457175104e+00
b_5	5.026236665731111e-02	3.094130079795092e-01
a_0	2.821623829721537e-01	1.694325964250128e-01
a_1	-1.320873224357864e+00	-1.653766869333608e+00
a_2	1.788929939776107e+00	5.877136313056599e+00
a_3	-6.791743332750140e-01	-1.039876562007935e+01
a_4	8.341719317811783e-01	9.933713752170028e+00
a_5	-1.901288029268752e+00	-4.927736676972851e+00

Tab. B.1: Paramètres des fonctions $F_i(z)$ pour les axes 1 et 2

i	3	4
b_0	1.618209448475971e-02	1.506568541905881e-01
b_1	-1.438083229839724e-02	1.724108963236753e-02
b_2	-4.155809667822411e-02	-1.692961613497715e-01
b_3	1.237903708619156e-03	-1.456914656999033e-01
b_4	4.856506636882708e-02	1.024656088198993e-01
b_5	5.137140461155544e-02	1.258927078619847e-01
a_0	-1.797587669060881e-01	3.255438618994087e-01
a_1	-3.178519644576170e-01	-1.178682444353927e+00
a_2	1.194466201461549e+00	1.272452494717649e+00
a_3	-4.715478142657912e-01	-1.613860830759655e-01
a_4	-1.059043989261378e-01	3.057038966400645e-01
a_5	-1.057985716708775e+00	-1.482363092372065e+00

Tab. B.2: Paramètres des fonctions $F_i(z)$ pour les axes 3 et 4

i	5	6
b_0	-7.946940757901037e-02	2.007411185150763e-01
b_1	1.605507873295317e-01	-1.052681950441290e-01
b_2	3.370462898710719e-02	-1.544416708388014e-01
b_3	-1.783610174272436e-01	-1.953271448163429e-01
b_4	-1.922559500395958e-02	1.155496070449986e-01
b_5	8.676908924591595e-02	1.533471959984967e-01
a_0	-3.615651480429528e-01	3.253912326535913e-01
a_1	1.389805655475779e+00	-1.168038531413835e+00
a_2	-1.872854366133754e+00	1.790117117127585e+00
a_3	1.125415137144070e-01	-1.409257917165848e+00
a_4	2.582165711364113e+00	1.255043399320783e+00
a_5	-2.846124880825251e+00	-1.778654389662978e+00

Tab. B.3: Paramètres des fonctions $F_i(z)$ pour les axes 5 et 6

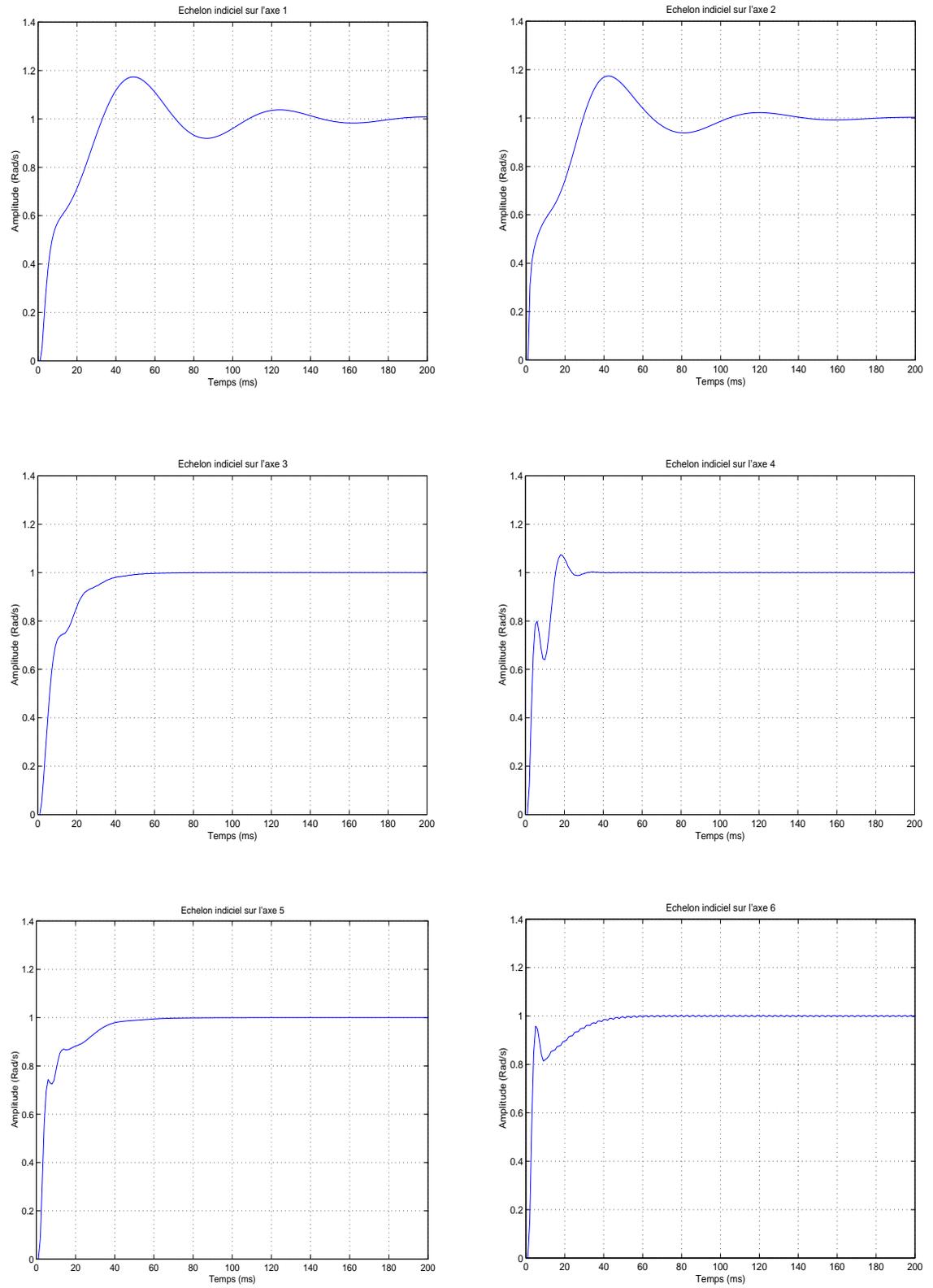


Fig. B.1: Réponses des $F_i(z)$ à un échelon de vitesse

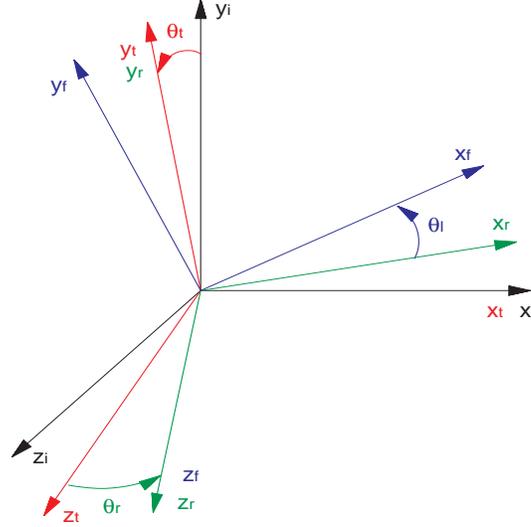


Fig. B.2: Définition des angles de tangage, roulis et lacet

B.3 Définition de la décomposition en tangage, roulis et lacet

La décomposition d'une rotation en tangage, roulis et lacet est décrite par la figure B.2. Le repère d'origine R_i est défini par les vecteurs \vec{x}_i , \vec{y}_i et \vec{z}_i . R_f est le repère final défini par \vec{x}_f , \vec{y}_f et \vec{z}_f . L'angle de tangage θ_t transforme le repère R_i en R_t défini par \vec{x}_t , \vec{y}_t et \vec{z}_t . L'angle de roulis θ_r transforme R_t en R_r défini par \vec{x}_r , \vec{y}_r et \vec{z}_r . Finalement, l'angle de lacet θ_l transforme R_r en R_f . Les matrices \mathcal{R}_{it} , \mathcal{R}_{tr} et \mathcal{R}_{rf} définissent respectivement les rotations de tangage, roulis et lacet :

$$\mathcal{R}_{it} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_t & -\sin \theta_t \\ 0 & \sin \theta_t & \cos \theta_t \end{pmatrix}_{R_i} \quad (\text{B.8})$$

$$\mathcal{R}_{tr} = \begin{pmatrix} \cos \theta_r & 0 & \sin \theta_r \\ 0 & 1 & 0 \\ -\sin \theta_r & 0 & \cos \theta_r \end{pmatrix}_{R_t} \quad (\text{B.9})$$

$$\mathcal{R}_{rf} = \begin{pmatrix} \cos \theta_l & -\sin \theta_l & 0 \\ \sin \theta_l & \cos \theta_l & 0 \\ 0 & 0 & 1 \end{pmatrix}_{R_r} \quad (\text{B.10})$$

l'avantage de la décomposition en tangage, roulis, lacet réside dans le fait qu'elle n'est pas singulière lorsque $R_i = R_f$ (contrairement à la décomposition en angles d'Euler par exemple).

Définition de T_6

Grâce à la décomposition d'une rotation en 3 rotations élémentaires, il est possible de convertir une matrice homogène en 6 coordonnées opérationnelles grâce à la transformation T_6 . Soit M une matrice homogène définie par :

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.15})$$

Alors $T_6(M)$ est défini par :

$$T_6(M) = \begin{bmatrix} T_x \\ T_y \\ T_z \\ \arctan2\left(\frac{-r_{23}}{\cos \theta_r}, \frac{r_{33}}{\cos \theta_r}\right) \\ \arcsin(r_{13}) \\ \arctan2\left(\frac{-r_{12}}{\cos \theta_r}, \frac{r_{11}}{\cos \theta_r}\right) \end{bmatrix} \quad (\text{B.16})$$

B.4 Calcul du Jacobien J_p

Le Jacobien J_p relie le torseur cinématique \dot{r} , de la vitesse d'un repère R_i par rapport à un repère R_f , à la dérivée par rapport au temps \dot{p} de l'attitude p entre R_i et R_f , le tout exprimé dans R_i .

Nous proposons un calcul de J_p dans le cas où les paramètres angulaires de p sont exprimés sous forme de tangage, roulis, lacet (*cf.* figure B.2).

Soit \dot{p} , le vecteur définissant la dérivée par rapport au temps des coordonnées opérationnelles de l'attitude p (*cf.* annexe B.3 pour la définition des angles) :

$$\dot{p} = \begin{bmatrix} \dot{T}_x \\ \dot{T}_y \\ \dot{T}_z \\ \dot{\theta}_t \\ \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (\text{B.17})$$

On adopte la notation suivante : ${}^y_x \vec{\Omega}_{R_z}$ désigne les coordonnées du vecteur vitesse de rotation d'un repère R_y par rapport à un repère R_y exprimées dans un repère R_z . Soit ${}^f_i \vec{\Omega}$, le vecteur vitesse de rotation de R_f par rapport à R_i :

$${}^f_i \vec{\Omega} = \dot{\theta}_t \vec{x}_i + \dot{\theta}_r \vec{y}_r + \dot{\theta}_l \vec{z}_f \quad (\text{B.18})$$

Exprimons ${}^f_i\vec{\Omega}$ en fonction de \vec{x}_i , \vec{y}_i et \vec{z}_i . On a :

$$\vec{y}_r = \vec{y}_i \cos \theta_t + \vec{z}_i \sin \theta_t \quad (\text{B.19})$$

$$\vec{z}_f = \vec{z}_t \cos \theta_r + \vec{x}_t \sin \theta_r \quad (\text{B.20})$$

Or :

$$\vec{x}_t = \vec{x}_i \quad (\text{B.21})$$

$$\vec{z}_t = \vec{z}_i \cos \theta_t - \vec{y}_i \sin \theta_t \quad (\text{B.22})$$

Il vient donc :

$$\vec{z}_f = \vec{x}_i \sin \theta_r - \vec{y}_i \sin \theta_t \cos \theta_r + \vec{z}_i \cos \theta_t \cos \theta_r \quad (\text{B.23})$$

D'où :

$$\begin{aligned} {}^f_i\vec{\Omega} &= \vec{x}_i(\dot{\theta}_t + \sin \theta_r \dot{\theta}_l) + \\ &\quad \vec{y}_i(\dot{\theta}_r \cos \theta_t - \dot{\theta}_l \sin \theta_t \cos \theta_r) + \\ &\quad \vec{z}_i(\dot{\theta}_r \sin \theta_t + \dot{\theta}_l \cos \theta_t \cos \theta_r) \end{aligned} \quad (\text{B.24})$$

Or, ${}^f_i\vec{\Omega} = -{}^i_f\vec{\Omega}$. D'où :

$$\begin{aligned} {}^i_f\vec{\Omega}_{R_i} &= \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = - \begin{bmatrix} 1 & 0 & \sin \theta_r \\ 0 & \cos \theta_t & -\sin \theta_t \cos \theta_r \\ 0 & \sin \theta_t & \cos \theta_t \cos \theta_r \end{bmatrix} \begin{bmatrix} \dot{\theta}_t \\ \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \\ &\triangleq -J_R[\dot{\Theta}] \end{aligned} \quad (\text{B.25})$$

Dans la partie qui suit, on notera ${}^y_x\vec{V}(O)_{R_z}$ les coordonnées dans R_z de la vitesse du point O dans le mouvement de R_y par rapport à R_x . On a donc :

$${}^f_i\vec{V}(O_f)_{R_i} = \begin{bmatrix} \dot{T}_x \\ \dot{T}_y \\ \dot{T}_z \end{bmatrix} \quad (\text{B.26})$$

où O_f est l'origine de R_f . Or :

$${}^f_i\vec{V}(O_f) = -{}^i_f\vec{V}(O_f) \quad (\text{B.27})$$

De plus :

$${}^i_f\vec{V}(O_i) = {}^i_f\vec{V}(O_f) + \overrightarrow{O_i O_f} \times {}^i_f\vec{\Omega} \quad (\text{B.28})$$

où O_i est l'origine de R_i . Si t_\times est défini de la manière suivante :

$$t_\times = \begin{bmatrix} 0 & -T_z & -T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \quad (\text{B.29})$$

On peut écrire que :

$${}^i_f \vec{V}(O_i)_{R_i} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = - \begin{bmatrix} \dot{T}_x \\ \dot{T}_y \\ \dot{T}_z \end{bmatrix} - t_{\times} J_R \dot{\Theta} \quad (\text{B.30})$$

Finalement, en regroupant (B.25) et (B.30), on obtient :

$$\begin{aligned} \dot{r} &= \begin{bmatrix} -I & -t_{\times} J_R \\ 0 & -J_R \end{bmatrix} \dot{p} \\ &\triangleq J_p \dot{p} \end{aligned} \quad (\text{B.31})$$

Remarque

Lorsque p tend vers 0, J_p tend vers $-I_6$. Donc, lorsque $R_i = R_f$, on a $\dot{r} = -\dot{p}$.

B.5 Calcul du Jacobien image dans le cas où les primitives sont des points

Soit R_c , un repère lié à une caméra (voir 2.1.2 et 2.2 pour la définition de R_c et la modélisation de la caméra). Soit \dot{r} , le torseur cinématique du repère R_c par rapport à un repère fixe R_b :

$$\dot{r} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{R_b} \quad (\text{B.32})$$

où V_x, V_y, V_z définissent la vitesse instantanée de déplacement de l'origine de R_c et $\omega_x, \omega_y, \omega_z$ sont les vitesses angulaires instantanées de rotation de R_c respectivement autour de ses axes x, y et z .

Soit P , un point fixe dans R_b dont les coordonnées instantanées P_c dans R_c sont données par :

$$P_c = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{R_c} \quad (\text{B.33})$$

La vitesse relative \dot{P}_c de P par rapport au repère R_c s'obtient par :

$$\dot{P}_c = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_{R_c} = - \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad (\text{B.34})$$

D'où :

$$\dot{K}_c = \begin{bmatrix} -\omega_y z + \omega_z y - V_x \\ -\omega_z x + \omega_x z - V_y \\ -\omega_x y + \omega_y x - V_z \end{bmatrix} \quad (\text{B.35})$$

En utilisant les relations de projection perspective suivantes (voir section 2.2) :

$$\begin{aligned} x_p &= G_x \frac{x}{z} \\ y_p &= G_y \frac{y}{z} \end{aligned} \quad (\text{B.36})$$

où

$$\mathcal{F} = \begin{bmatrix} x_p \\ y_p \end{bmatrix} \quad (\text{B.37})$$

sont les coordonnées en pixels de la projection de P dans le plan image, l'équation (B.35) devient :

$$\dot{P}_c = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_{R_c} = \begin{bmatrix} -\omega_y z + \omega_z \frac{zy_p}{G_y} - V_x \\ -\omega_z \frac{zx_p}{G_x} + \omega_x z - V_y \\ -\omega_x \frac{zy_p}{G_y} + \omega_y \frac{zx_p}{G_x} - V_z \end{bmatrix} \quad (\text{B.38})$$

En dérivant les équation (B.36), on obtient :

$$\begin{aligned} \dot{x}_p &= G_x \frac{\dot{x}z - x\dot{z}}{z^2} \\ \dot{y}_p &= G_y \frac{\dot{y}z - y\dot{z}}{z^2} \end{aligned} \quad (\text{B.39})$$

En utilisant (B.38) et (B.36) dans (B.39), il vient :

$$\begin{aligned} \dot{x}_p &= -\frac{G_x}{z} V_x + \frac{x_p}{z} V_z + \frac{x_p y_p}{G_y} \omega_x - \frac{G_x^2 + x_p^2}{G_x} \omega_y + \frac{y_p G_x}{G_y} \omega_z \\ \dot{y}_p &= -\frac{G_y}{z} V_y + \frac{y_p}{z} V_z + \frac{G_y^2 + y_p^2}{G_y} \omega_x - \frac{x_p y_p}{G_x} \omega_y - \frac{x_p G_y}{G_x} \omega_z \end{aligned} \quad (\text{B.40})$$

Cette équation peut se mettre sous forme matricielle :

$$\dot{\mathcal{F}} = J_v(\mathcal{F}) \dot{r} \quad (\text{B.41})$$

où

$$J_v(\mathcal{F}) = \begin{bmatrix} -\frac{G_x}{z} & 0 & \frac{x_p}{z} & \frac{x_p y_p}{G_y} & -\frac{G_x^2 + x_p^2}{G_x} & \frac{y_p G_x}{G_y} \\ 0 & -\frac{G_y}{z} & \frac{y_p}{z} & \frac{G_y^2 + y_p^2}{G_y} & -\frac{x_p y_p}{G_x} & -\frac{x_p G_y}{G_x} \end{bmatrix} \quad (\text{B.42})$$

La matrice $J_v(\mathcal{F})$ est le Jacobien image dans le cas où les primitives sont des points. Si le vecteur \mathcal{F} contient les coordonnées de n points, alors $J_v(\mathcal{F})$ est construit en empilant n fois la matrice élémentaire décrite par (B.42). On pourra trouver un calcul analogue de ce Jacobien image dans [HHC96].¹

¹ L'auteur y calcul le Jacobien image d'une caméra fixe qui observe un point de l'organe terminal d'un robot dont le torseur cinématique est \dot{r} . Par conséquent, tous les éléments de J_v sont de signe opposé.

B.6 Calcul de l'attitude d'un objet grâce à la projection de n points coplanaires

Soit R_o , un repère lié à un objet plan et R_c un repère lié à la caméra (*cf.* section 2.1.2, page 20 pour la définition de R_c). Le repère R_o est placé de manière à ce que tous les points de l'objet soient dans le plan (xy) de ce repère. Soit P , un point de l'objet, et P_p son projeté dans le plan image :

$$P = \begin{pmatrix} x_o \\ y_o \\ 0 \end{pmatrix}_{R_o} \quad P_p = \begin{pmatrix} x_p \\ y_p \\ f \end{pmatrix}_{R_c} \quad (\text{B.43})$$

où f est la longueur focale de la caméra. Soit M_{co} , la matrice homogène de transformation entre R_c et R_o exprimée dans R_c :

$$M_{co} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}_{R_c} \quad (\text{B.44})$$

D'après la loi de projection perspective, on a donc :

$$\begin{aligned} x_p &= f \frac{r_{11}x_o + r_{12}y_o + T_x}{r_{31}x_o + r_{32}y_o + r_{33}z_o + T_z} \\ y_p &= f \frac{r_{21}x_o + r_{22}y_o + T_y}{r_{31}x_o + r_{32}y_o + r_{33}z_o + T_z} \end{aligned} \quad (\text{B.45})$$

Mais la caméra fournit les valeurs X_p et Y_p de x_p et y_p en pixels. Il faut donc convertir les pixels en longueur. Pour cela, on définit G_x et G_y , respectivement les grandissements suivant x et y de la caméra :

$$\begin{aligned} x_p &= \frac{f}{G_x} X_p \\ y_p &= \frac{f}{G_y} Y_p \end{aligned} \quad (\text{B.46})$$

En divisant les équations (B.45) par T_z , et en exprimant les coordonnées des projections en pixels, on obtient :

$$X_p = \frac{\theta_1 x_o + \theta_2 y_o + \theta_3}{\theta_7 x_o + \theta_8 y_o + 1} \quad (\text{B.47})$$

$$Y_p = \frac{\theta_4 x_o + \theta_5 y_o + \theta_6}{\theta_7 x_o + \theta_8 y_o + 1} \quad (\text{B.48})$$

avec :

$$\begin{aligned}
\theta_1 &= \frac{r_{11}G_x}{T_z} \\
\theta_2 &= \frac{r_{12}G_x}{T_z} \\
\theta_3 &= \frac{T_x G_x}{T_z} \\
\theta_4 &= \frac{r_{21}G_y}{T_z} \\
\theta_5 &= \frac{r_{22}G_y}{T_z} \\
\theta_6 &= \frac{T_y G_y}{T_z} \\
\theta_7 &= \frac{r_{31}}{T_z} \\
\theta_8 &= \frac{r_{32}}{T_z}
\end{aligned} \tag{B.49}$$

En divisant (B.47) par (B.48), on obtient :

$$-Y_p y_o \theta_2 + X_p x_o \theta_4 + X_p y_o \theta_5 + X_p \theta_6 = Y_p (x_o \theta_1 + \theta_3) \tag{B.50}$$

Cette équation peut s'écrire sous forme matricielle :

$$[-Y_p y_o \quad X_p x_o \quad X_p y_o \quad X_p] \begin{pmatrix} \theta_2 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{pmatrix} = [Y_p x_o \quad Y_p] \begin{pmatrix} \theta_1 \\ \theta_3 \end{pmatrix} \tag{B.51}$$

Soit n , le nombre de points dont on a la projection dans l'image. Si $n \geq 4$, l'équation (B.51) admet une solution exacte ou au sens des moindres carrés pour α_{21} , α_{23} , α_{41} , α_{43} , α_{51} , α_{53} , α_{61} et α_{63} :

$$\begin{cases} \theta_2 = \alpha_{21}\theta_1 + \alpha_{23}\theta_3 \\ \theta_4 = \alpha_{41}\theta_1 + \alpha_{43}\theta_3 \\ \theta_5 = \alpha_{51}\theta_1 + \alpha_{53}\theta_3 \\ \theta_6 = \alpha_{61}\theta_1 + \alpha_{63}\theta_3 \end{cases} \tag{B.52}$$

En substituant ces résultats dans l'équation (B.47) on obtient :

$$X_p x_o \theta_7 + X_p y_o \theta_8 + X_p - \theta_1 x_o - \theta_3 - y_o (\alpha_{21}\theta_1 + \alpha_{23}\theta_3) = 0 \tag{B.53}$$

Cette équation mise sous forme matricielle devient :

$$[(x_o + \alpha_{21}y_o) \quad (1 + \alpha_{23}y_o) \quad -X_p x_o \quad -X_p y_o] \begin{pmatrix} \theta_1 \\ \theta_3 \\ \theta_7 \\ \theta_8 \end{pmatrix} = (x) \tag{B.54}$$

La résolution de (B.54) pour $n \geq 4$ permet de déterminer la valeur de θ_1 , θ_3 , θ_7 et θ_8 . Finalement, en remplaçant θ_1 et θ_3 dans l'équation (B.52) par leurs valeurs, on en déduit les valeurs de θ_2 , θ_4 , θ_5 et θ_6 .

Afin de reconstituer tous les paramètres de la matrice homogène M_{co} , il faut déterminer la profondeur T_z . Celle-ci peut se déduire en utilisant la propriété d'orthogonalité de la matrice de rotation. Ainsi, en exprimant la normalité du premier vecteur-colonne de la matrice de rotation, on a :

$$r_{11}^2 + r_{21}^2 + r_{31}^2 = 1 \quad (\text{B.55})$$

D'où, en utilisant les relations (B.49), on obtient :

$$\left(\frac{\theta_1 T_z}{G_x}\right)^2 + \left(\frac{\theta_4 T_z}{G_y}\right)^2 + (\theta_7 T_z)^2 = 1 \quad (\text{B.56})$$

Sachant que T_z est toujours positif (l'objet est toujours situé devant la caméra), on en déduit T_z :

$$T_z = \frac{1}{\sqrt{\left(\frac{\theta_1}{G_x}\right)^2 + \left(\frac{\theta_4}{G_y}\right)^2 + \theta_7^2}} \quad (\text{B.57})$$

B.7 Calcul récursif des équations Diophantiennes du GPC

B.7.1 Résolution récursive de l'équation Diophantienne (3.23)

Soit l'équation Diophantienne (3.23) :

$$C = A\Delta E_j + q^{-j} F_j \quad (\text{B.58})$$

avec :

$$A = 1 + a_1 q^{-1} + \dots + a_{na} q^{-na} \quad (\text{B.59})$$

$$C = 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc} \quad (\text{B.60})$$

$$\Delta = 1 - q^{-1} \quad (\text{B.61})$$

$$E_j = e_0^{(j)} + e_1^{(j)} q^{-1} + \dots + e_{j-1}^{(j)} q^{-j+1} \quad (\text{B.62})$$

$$F_j = f_0^{(j)} + f_1^{(j)} q^{-1} + \dots + f_{nf}^{(j)} q^{-nf} \quad (\text{B.63})$$

Cette équation peut être mise sous la forme suivante :

$$\begin{aligned} & 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc} = \\ & [1 + (a_1 - 1)q^{-1} + (a_2 - a_1)q^{-2} + \dots + (a_{na} - a_{na-1})q^{-na} - a_{na} q^{-na-1}] \\ \times & [e_0^{(j)} + e_1^{(j)} q^{-1} + \dots + e_{j-1}^{(j)} q^{-j+1}] + f_0^{(j)} q^{-j} + f_1^{(j)} q^{-j-1} + \dots + f_{nf}^{(j)} q^{-j-nf} \end{aligned} \quad (\text{B.64})$$

L'ordre de $A\Delta E_j = na + j$ et l'ordre de $q^{-j}F_j = nf^{(j)} + j$. Etant donné que les ordres des deux membres de l'équation (B.64) doivent être identiques, on en déduit l'ordre nf du polynôme F_j :

$$\begin{cases} nc \leq na + j & \Rightarrow na + j = nf^{(j)} + j \\ nc > na + j & \Rightarrow nf^{(j)} + j = nc \end{cases} \quad (\text{B.65})$$

d'où :

$$nf^{(j)} = \max(na, nc - j) \quad (\text{B.66})$$

Pour l'horizon de prédiction $j - 1$, l'équation (B.58) s'écrit :

$$C = A\Delta E_{j-1} + q^{-j+1}F_{j-1} \quad (\text{B.67})$$

A partir de (B.58) et (B.67), on obtient :

$$A\Delta(E_j - E_{j-1}) + q^{-j+1}(q^{-1}F_j - F_{j-1}) = 0 \quad (\text{B.68})$$

On peut réécrire (B.68) de la manière suivante :

$$\begin{aligned} & [1 + (a_1 - 1)q^{-1} + \dots + (a_{na} - a_{na-1})q^{-na} - a_{na-1}q^{-na-1}] \\ & \times [(e_0^{(j)} - e_0^{(j-1)}) + (e_1^{(j)} - e_1^{(j-1)})q^{-1} + \dots + (e_{j-2}^{(j)} - e_{j-2}^{(j-1)})q^{-j+2} + e_{j-1}^{(j)}q^{-j+1}] \\ & - q^{-j+1}[f_0^{(j-1)} + (f_1^{(j-1)} - f_0^{(j)})q^{-1} + \dots + (f_{nf}^{(j-1)} - f_{nf-1}^{(j)})q^{-nf} - f_{nf}^{(j)}q^{-nf-1}] \\ & = 0 \end{aligned} \quad (\text{B.69})$$

On en déduit que :

$$e_k^{(j)} - e_k^{(j-1)} = 0 \quad \forall k = 0, \dots, j - 2 \quad (\text{B.70})$$

Par conséquent, nous avons :

$$E_j = E_{j-1} + e_{j-1}^{(j)}q^{-j+1} \quad (\text{B.71})$$

$$(\text{B.72})$$

Etant donné que $e_k^{(j)} = e_k^{(j-1)} \quad \forall (k, j) \in \mathbb{N}^{+2}$, nous pouvons simplifier les notations en écrivant que $e_k^{(j)} = e_k \quad \forall j$. Ainsi l'équation (B.69) devient :

$$\begin{aligned} & \overbrace{[1 + (a_1 - 1)q^{-1} + \dots + (a_{na} - a_{na-1})q^{-na} - a_{na-1}q^{-na-1}] \times e_{j-1}q^{-j+1}}^{A\Delta} \\ & = q^{-j+1} \times \underbrace{[f_0^{(j-1)} + (f_1^{(j-1)} - f_0^{(j)})q^{-1} + \dots + (f_{nf}^{(j-1)} - f_{nf-1}^{(j)})q^{-nf} - f_{nf}^{(j)}q^{-nf-1}]}_{F_{j-1} - q^{-1}F_j} \end{aligned} \quad (\text{B.73})$$

Et on en déduit que :

$$q^{-1}F_j = F_{j-1} - e_{j-1}A\Delta \quad (\text{B.74})$$

A partir de ces relations, nous pouvons définir l'algorithme de calcul récursif suivant :

Initialisation de la récurrence

Pour $j = 1$, l'équation (B.64) devient :

$$\begin{aligned} & 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc} = \\ & e_0 \times [1 + (a_1 - 1)q^{-1} + (a_2 - a_1)q^{-2} + \dots + (a_{na} - a_{na-1})q^{-na} - a_{na}q^{-na-1}] \\ & + f_0^{(1)}q^{-1} + f_1^{(1)}q^{-2} + \dots + f_{nf}^{(1)}q^{-1-nf} \end{aligned} \quad (\text{B.75})$$

avec $nf^{(1)} = \max(na, nc - 1)$. On en déduit les initialisations suivantes :

$$\begin{cases} e_0 &= 1 \\ f_i^1 &= c_{i+1} - e_0(a_{i+1} - a_i) \quad 0 \leq i \leq nf^{(1)} \end{cases} \quad (\text{B.76})$$

Calcul récursif de E et F

En utilisant l'équation (B.74), on en déduit la récurrence pour $j \geq 2$:

$$\begin{cases} e_{j-1} &= f_0^{j-1} \\ f_i^j &= f_{i+1}^{j-1} - e_{j-1}(a_{i+1} - a_i) \quad 0 \leq i \leq nf^{(j)} \end{cases} \quad (\text{B.77})$$

B.7.2 Résolution récursive de l'équation Diophantienne (3.32)

Soit l'équation Diophantienne (3.32) :

$$BE_j = CG_j + q^{-j}H_j \quad (\text{B.78})$$

où :

$$B = b_0q^{-d} + b_1q^{-1-d} + \dots + b_{nb}q^{-nb-d} \quad (\text{B.79})$$

$$E_j = e_0 + e_1q^{-1} + \dots + e_{j-1}q^{-j+1} \quad (\text{B.80})$$

$$C = 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc} \quad (\text{B.81})$$

$$G_j = g_0^j + g_1^jq^{-1} + \dots + g_{j-1}^jq^{-j+1} \quad (\text{B.82})$$

$$H_j = h_0^j + h_1^jq^{-1} + \dots + h_{nh}^jq^{-nh} \quad (\text{B.83})$$

L'équation (B.78) peut être mise sous la forme suivante :

$$\begin{aligned} & \overbrace{(b_0q^{-d} + b_1q^{-1-d} + \dots + b_{nb}q^{-nb-d})}^{P_1} (e_0 + e_1q^{-1} + \dots + e_{j-1}q^{-j+1}) = \\ & \overbrace{(1 + c_1q^{-1} + \dots + c_{nc}q^{-nc})}^{P_2} (g_0^j + g_1^jq^{-1} + \dots + g_{j-1}^jq^{-j+1}) \\ & + \overbrace{(h_0^jq^{-j} + h_1^jq^{-j-1} + \dots + h_{nh}^jq^{-j-nh})}^{P_3} \end{aligned} \quad (\text{B.84})$$

L'ordre des polynômes des 2 membres de cette équation doit être identique. D'où les relations suivantes :

$$\begin{cases} \text{Si } \text{ordre}(P_1) \leq \text{ordre}(P_2) & \text{alors } \text{ordre}(P_3) = \text{ordre}(P_2) \\ \text{Si } \text{ordre}(P_1) > \text{ordre}(P_2) & \text{alors } \text{ordre}(P_3) = \text{ordre}(P_1) \end{cases} \quad (\text{B.85})$$

Ce qui se traduit par :

$$\begin{cases} nb + d \leq nc & \Rightarrow nh = nc - 1 \\ nb + d > nc & \Rightarrow nh = nb + d - 1 \end{cases} \quad (\text{B.86})$$

On en déduit l'ordre nh du polynôme H :

$$nh = \max(nc, nb + d) - 1 \quad (\text{B.87})$$

Réécrivons l'équation (B.78) pour l'horizon de prédiction $j - 1$:

$$BE_{j-1} = CG_{j-1} + q^{-j+1}H_{j-1} \quad (\text{B.88})$$

En faisant la différence entre (B.78) et (B.88), on obtient :

$$C(G_j - G_{j-1}) + q^{-j+1}[q^{-1}H_j - H_{j-1} - q^{j-1}B(E_j - E_{j-1})] = 0 \quad (\text{B.89})$$

or (*cf.* équation (B.71)) :

$$E_j - E_{j-1} = e_{j-1}q^{-j+1} \quad (\text{B.90})$$

d'où :

$$C(G_j - G_{j-1}) - q^{-j+1}[H_{j-1} - q^{-1}H_j + Be_{j-1}] = 0 \quad (\text{B.91})$$

En réécrivant cette équation sous forme étendue nous obtenons :

$$\begin{aligned} & (1 + c_1q^{-1} + \dots + c_{nc}q^{-nc}) \\ & \times [(g_0^{(j)} - g_0^{(j-1)}) + (g_1^{(j)} - g_1^{(j-1)})q^{-1} + \dots + (g_{j-2}^{(j)} - g_{j-2}^{(j-1)})q^{-j+2} + g_{j-1}^{(j)}q^{-j+1}] \\ & - q^{-j+1}[h_0^{(j-1)} + (h_1^{(j-1)} - h_0^{(j)})q^{-1} + \dots + (h_{nh}^{(j-1)} - h_{nh-1}^{(j)})q^{-nh} - h_{nh}^{(j)}q^{-nh-1} \\ & + e_{j-1}(b_0q^{-d} + b_1q^{-1-d} + \dots + b_{nb}q^{-nb-d})] \\ & = 0 \end{aligned} \quad (\text{B.92})$$

Etant donné que C et q^{-j+1} sont premiers entre eux, on en déduit que :

$$g_k^{(j)} - g_k^{(j-1)} = 0 \quad \forall k = 0, \dots, j - 2 \quad (\text{B.93})$$

Par conséquent, nous avons :

$$G_j - G_{j-1} = g_{j-1}^j q^{-j+1} \quad (\text{B.94})$$

Etant donné que $g_k^j = g_k^{j-1} \forall (k, j) \in \mathbb{N}^{+2}$, nous pouvons simplifier les notations et écrire que $g_k^{(j)} = g_k \forall j$. Ainsi l'équation (B.91) devient :

$$Cg_{j-1}q^{-j+1} - q^{-j+1}[H_{j-1} - q^{-1}H_j + Be_{j-1}] = 0 \quad (\text{B.95})$$

d'où on peut déduire l'équation récursive suivante :

$$q^{-1}H_j = H_{j-1} - g_{j-1}C + e_{j-1}B \quad (\text{B.96})$$

A partir de ces relations, nous pouvons définir l'algorithme de calcul récursif suivant :

Initialisation de la récurrence

Pour $j = 1$, l'équation (B.84) devient :

$$e_0 \times (b_0 q^{-d} + b_1 q^{-d-1} + \dots + b_{nb} q^{-nb-d}) = (1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc}) \times g_0 + (h_0^{(1)} q^{-1} + h_1^{(1)} q^{-2} + \dots + h_{nh}^{(1)} q^{-1-nh}) \quad (\text{B.97})$$

On en déduit que :

$$\text{Si } d = 0 \quad \begin{cases} g_0 & = e_0 b_0 \\ h_i^{(1)} & = b_{i+1} e_0 - g_0 c_{i+1} \quad 0 \leq i \leq nh \end{cases} \quad (\text{B.98})$$

$$\text{Si } d > 0 \quad \begin{cases} g_0 & = 0 \\ h_i^{(1)} & = b_{i-d+1} e_0 \quad 0 \leq i \leq nh \end{cases} \quad (\text{B.99})$$

Calcul récursif de G et H

En réécrivant l'équation (B.96) de manière étendue on obtient :

$$\begin{aligned} & h_0^{(j)} q^{-1} + h_1^{(j)} q^{-2} + \dots + h_{nh}^{(j)} q^{-nh-1} = \\ & h_0^{(j-1)} + h_1^{(j-1)} q^{-1} + \dots + h_{nh}^{(j-1)} q^{-nh} \\ & - g_{j-1} (1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc}) \\ & + e_{j-1} (b_0 q^{-d} + b_1 q^{-1-d} + \dots + b_{nb} q^{-nb-d}) \end{aligned} \quad (\text{B.100})$$

Pour $j \geq 2$ on a donc :

$$\begin{cases} g_{j-1} & = \begin{cases} h_0^{(j-1)} + b_0 e_{j-1} & \text{si } d = 0 \\ h_0^{(j-1)} & \text{si } d > 0 \end{cases} \\ h_i^{(j)} & = h_{i+1}^{(j-1)} - c_{i+1} g_{j-1} + e_{j-1} b_{i+1-d} \quad 0 \leq i \leq nh \end{cases} \quad (\text{B.101})$$

B.8 Expression de K_1 , K_2 et K_3 dans (5.29), (5.27) et (5.28)

$$\begin{aligned}
K_1 = & (-V_z u_{x1} u_{y2} a d_3 u_{x3} + V_z u_{y1} o_{x2} u_{y2} d_3 u_{x3} \\
& - V_y o_{y3} u_{x3} u_{x1} u_{y2} a d_3 + V_y d_1 u_{y2} o_{y3} u_{x3} o_{y1} u_{x1} \\
& - V_y d_1 u_{x3} u_{y2} a o_{y1} u_{x1} + V_y d_1 u_{x3} u_{y2} a o_{x1} u_{y1} \\
& + V_y o_{y3} u_{x3} u_{y1} u_{x2} a d_3 - V_y o_{y3} u_{x3} u_{y1} o_{y2} u_{x2} d_3 \\
& - V_y d_1 u_{y2} o_{y3} u_{x3} o_{x1} u_{y1} - V_y d_1 u_{y3} o_{y2} u_{x2} o_{y1} u_{x1} \\
& + V_y d_1 u_{y3} o_{y2} u_{x2} o_{x1} u_{y1} - V_z d_1 u_{x1} u_{x3} u_{y2} a \\
& + V_y d_1 u_{y3} o_{x2} u_{y2} o_{y1} u_{x1} - V_y d_1 u_{y3} o_{x2} u_{y2} o_{x1} u_{y1} \\
& + V_y o_{x3} u_{y3} u_{x1} u_{y2} a d_3 - V_y o_{x3} u_{y3} u_{y2} o_{y1} u_{x1} d_3 \\
& + V_y o_{x3} u_{y3} u_{y2} o_{x1} u_{y1} d_3 - V_y d_1 u_{y2} o_{x3} u_{y3} o_{y1} u_{x1} \\
& + V_y d_1 u_{y2} o_{x3} u_{y3} o_{x1} u_{y1} - V_y o_{x3} u_{y3} u_{y1} u_{x2} a d_3 \\
& + V_y o_{x3} u_{y3} u_{y1} o_{y2} u_{x2} d_3 - V_y o_{x3} u_{y3} u_{y1} o_{x2} u_{y2} d_3 \\
& + V_y o_{y3} u_{x3} u_{y1} o_{x2} u_{y2} d_3 + V_z d_1 u_{x1} u_{y3} o_{x2} u_{y2} \\
& + V_y d_1 u_{y3} u_{x2} a o_{y1} u_{x1} - V_z d_1 u_{x1} u_{y3} o_{y2} u_{x2} \\
& - V_y d_1 u_{y3} u_{x2} a o_{x1} u_{y1} + V_y o_{y3} u_{x3} u_{y2} o_{y1} u_{x1} d_3 \\
& - V_y o_{y3} u_{x3} u_{y2} o_{x1} u_{y1} d_3 + V_z d_1 u_{x1} u_{y3} u_{x2} a \\
& - V_z u_{y2} o_{x1} u_{y1} d_3 u_{x3} + V_z d_1 u_{x1} u_{y2} o_{y3} u_{x3} \\
& - V_z d_1 u_{x1} u_{y2} o_{x3} u_{y3} + V_z u_{y1} u_{x2} a d_3 u_{x3} \\
& - V_z u_{y1} o_{y2} u_{x2} d_3 u_{x3} + V_z u_{y2} o_{y1} u_{x1} d_3 u_{x3}) / (d_1 u_{y2} d_3 (\\
& - u_{y1} V_y o_{y3} u_{x3} + u_{y1} V_y o_{x3} u_{y3} + u_{x1} u_{y3} V_z - u_{x1} V_x o_{y3} u_{x3} \\
& + u_{x1} V_x o_{x3} u_{y3} - u_{y1} V_z u_{x3} + V_x o_{y1} u_{x1} u_{x3} + o_{y1} u_{x1} u_{y3} V_y \\
& - V_x o_{x1} u_{y1} u_{x3} - o_{x1} u_{y1} u_{y3} V_y))
\end{aligned} \tag{B.102}$$

$$\begin{aligned}
K_2 = & -(-u_{y3}V_z u_{y1} o_{x2} u_{y2} d_3 - u_{y3}V_z u_{y1} u_{x2} a d_3 \\
& + u_{y3}V_z u_{y1} o_{y2} u_{x2} d_3 - u_{y3}V_z u_{y2} o_{y1} u_{x1} d_3 \\
& + u_{y3}V_z u_{x1} u_{y2} a d_3 + u_{y3}V_z u_{y2} o_{x1} u_{y1} d_3 \\
& - d_1 u_{y3} u_{x2} a u_{y1} V_z - d_1 u_{y3} o_{x2} u_{y2} u_{y1} V_z \\
& + d_1 u_{y3} o_{y2} u_{x2} u_{y1} V_z + u_{x3} d_1 u_{y2} a u_{y1} V_z \\
& + d_1 u_{y2} o_{x3} u_{y3} u_{y1} V_z - u_{x3} d_1 u_{y2} o_{y3} u_{y1} V_z \\
& - V_x o_{y3} u_{x3} u_{x1} u_{y2} a d_3 - V_x d_1 u_{y2} o_{y3} u_{x3} o_{x1} u_{y1} \\
& + V_x o_{y3} u_{x3} u_{y2} o_{y1} u_{x1} d_3 - V_x o_{y3} u_{x3} u_{y2} o_{x1} u_{y1} d_3 \\
& + V_x d_1 u_{y2} o_{y3} u_{x3} o_{y1} u_{x1} - V_x d_1 u_{x3} u_{y2} a o_{y1} u_{x1} \\
& + V_x d_1 u_{x3} u_{y2} a o_{x1} u_{y1} + V_x o_{y3} u_{x3} u_{y1} u_{x2} a d_3 \\
& - V_x o_{y3} u_{x3} u_{y1} o_{y2} u_{x2} d_3 + V_x o_{y3} u_{x3} u_{y1} o_{x2} u_{y2} d_3 \\
& + V_x d_1 u_{y3} u_{x2} a o_{y1} u_{x1} - V_x d_1 u_{y3} u_{x2} a o_{x1} u_{y1} \\
& - V_x d_1 u_{y3} o_{y2} u_{x2} o_{y1} u_{x1} + V_x d_1 u_{y3} o_{y2} u_{x2} o_{x1} u_{y1} \\
& + V_x d_1 u_{y3} o_{x2} u_{y2} o_{y1} u_{x1} - V_x d_1 u_{y3} o_{x2} u_{y2} o_{x1} u_{y1} \\
& + V_x o_{x3} u_{y3} u_{x1} u_{y2} a d_3 - V_x o_{x3} u_{y3} u_{y2} o_{y1} u_{x1} d_3 \\
& + V_x o_{x3} u_{y3} u_{y2} o_{x1} u_{y1} d_3 - V_x d_1 u_{y2} o_{x3} u_{y3} o_{y1} u_{x1} \\
& + V_x d_1 u_{y2} o_{x3} u_{y3} o_{x1} u_{y1} - V_x o_{x3} u_{y3} u_{y1} u_{x2} a d_3 \\
& + V_x o_{x3} u_{y3} u_{y1} o_{y2} u_{x2} d_3 - V_x o_{x3} u_{y3} u_{y1} o_{x2} u_{y2} d_3) / (d_1 \\
& u_{y2} d_3 (-u_{y1} V_y o_{y3} u_{x3} + u_{y1} V_y o_{x3} u_{y3} + u_{x1} u_{y3} V_z \\
& - u_{x1} V_x o_{y3} u_{x3} + u_{x1} V_x o_{x3} u_{y3} - u_{y1} V_z u_{x3} + V_x o_{y1} u_{x1} u_{x3} \\
& + o_{y1} u_{x1} u_{y3} V_y - V_x o_{x1} u_{y1} u_{x3} - o_{x1} u_{y1} u_{y3} V_y))
\end{aligned} \tag{B.103}$$

$$\begin{aligned}
K_3 = & (-d_1 u_{y1} V_y u_{y3} u_{x2} a + d_1 u_{y1} V_y u_{y3} o_{y2} u_{x2} \\
& - d_1 u_{y1} V_y u_{y3} o_{x2} u_{y2} + d_1 u_{y1} V_y u_{x3} u_{y2} a \\
& - d_1 u_{y1} V_y u_{y2} o_{y3} u_{x3} + d_1 u_{y1} V_y u_{y2} o_{x3} u_{y3} \\
& - d_1 u_{x1} V_x u_{y3} u_{x2} a + d_1 u_{x1} V_x u_{y3} o_{y2} u_{x2} \\
& - d_1 u_{x1} V_x u_{y3} o_{x2} u_{y2} + d_1 u_{x1} V_x u_{x3} u_{y2} a \\
& - d_1 u_{x1} V_x u_{y2} o_{y3} u_{x3} + d_1 u_{x1} V_x u_{y2} o_{x3} u_{y3} \\
& - u_{y1} V_x u_{x2} a d_3 u_{x3} - u_{y1} u_{x2} a d_3 u_{y3} V_y \\
& + u_{y1} V_x o_{y2} u_{x2} d_3 u_{x3} + u_{y1} o_{y2} u_{x2} d_3 u_{y3} V_y \\
& - u_{y1} V_x o_{x2} u_{y2} d_3 u_{x3} - u_{y1} o_{x2} u_{y2} d_3 u_{y3} V_y \\
& + u_{x1} V_x u_{y2} a d_3 u_{x3} + u_{x1} u_{y2} a d_3 u_{y3} V_y \\
& - V_x u_{y2} o_{y1} u_{x1} d_3 u_{x3} - u_{y2} o_{y1} u_{x1} d_3 u_{y3} V_y \\
& + V_x u_{y2} o_{x1} u_{y1} d_3 u_{x3} + u_{y2} o_{x1} u_{y1} d_3 u_{y3} V_y) / (d_1 u_{y2} d_3 (\\
& - u_{y1} V_y o_{y3} u_{x3} + u_{y1} V_y o_{x3} u_{y3} + u_{x1} u_{y3} V_z - u_{x1} V_x o_{y3} u_{x3} \\
& + u_{x1} V_x o_{x3} u_{y3} - u_{y1} V_z u_{x3} + V_x o_{y1} u_{x1} u_{x3} + o_{y1} u_{x1} u_{y3} V_y \\
& - V_x o_{x1} u_{y1} u_{x3} - o_{x1} u_{y1} u_{y3} V_y))
\end{aligned} \tag{B.104}$$

Annexe C

LE LOGICIEL

C.1 Le logiciel de contrôle de la boucle de vision

C.1.1 Cahier des charges

La réalisation du système expérimental répond à un certain nombre de contraintes. Ces dernières ne sont pas apparues immédiatement, mais se sont greffées au fur et à mesure que le projet a avancé. On peut les ranger en 2 catégories. Certaines consistent à diminuer le plus possible les délais de traitement et à synchroniser l'ensemble des tâches de manière à rendre l'asservissement le plus rapide possible :

- Tout l'asservissement doit être cadencé sur le rythme d'acquisition des images.
- Les boucles internes d'asservissement de position angulaire des axes doivent être désactivées.
- Le traitement de l'information visuelle introduit un retard pur égal à une période d'échantillonnage.

D'autres visent à améliorer l'ergonomie de la manipulation faisant ainsi gagner beaucoup de temps d'expérimentation :

- Toutes les fonctions de la manipulation sont accessibles en temps réel à partir d'un clavier d'ordinateur.
- En cas d'erreur du programme d'asservissement, le système doit se mettre dans un état de sécurité stable à partir duquel il peut rapidement et facilement sortir dès que le programme d'asservissement a été relancé.
- Le code du programme d'asservissement est le même que celui de la simulation à l'interface utilisateur près.

Cette dernière contrainte est peut-être celle qui fût la plus bénéfique en terme de gain de temps. En effet, les parties de code les plus critiques sont celles qui peuvent être partagées avec la simulation et donc être testées sans risque sur un ordinateur. Ainsi, une fois la partie de code mise au point, une simple re-compilation sur la plate-forme expérimentale permet de transférer la modification sur le système réel.

C.1.2 Choix du langage de programmation

Les contraintes temporelles de l'asservissement imposent un langage qui soit le plus proche possible des instructions du processeur. Le langage C ANSI répond à cette contrainte. La programmation directe en code assembleur est exclue étant donné le faible bénéfice en terme de performance qu'on pourrait en retirer par rapport au travail que cela réclamerait. De plus, nous nous sommes imposé une contrainte de portabilité entre le programme tournant sur le système réel et le programme de simulation. Seul un langage structuré évolué et normalisé permet d'obtenir cela. Le langage C présente néanmoins un inconvénient important lié à la rapidité. Contrairement au langage Pascal ou au langage Fortran, les erreurs survenues lors de l'exécution ne sont pas détectées. Ceci peut être gênant lors de calculs faisant intervenir des tableaux de données comme les calculs matriciels. Or, presque tous les modules du programme d'asservissement par vision contiennent de tels calculs. Là encore, le partage du code entre la simulation et l'expérience se révèle très bénéfique. En effet, sur certains compilateurs, il est possible d'activer une option qui permet de générer un code pourvu de mécanismes de contrôle pendant l'exécution. Ainsi, le code est toujours testé lors de la simulation puis implanté sans risque sur le système réel.

C.1.3 Choix de l'ordinateur et du système d'exploitation

L'ordinateur doit être évolutif. En effet, les tâches de traitement d'image requièrent d'importantes ressources de puissance de calcul. L'évolution de la cadence des processeurs suit une loi exponentielle. Il est donc judicieux de planifier une expérience à long terme en tablant sur l'évolution probable de la puissance de calcul des processeurs. Mais, afin de limiter au maximum le travail de portage du logiciel d'un système moins performant vers un système plus performant, il est souhaitable que les deux systèmes soient compatibles entre eux. A ce jour, seul le système d'exploitation MS-DOS garanti une compatibilité matérielle et logicielle ascendante sur une période de plus de 10 ans. Certes, comparé à d'autres systèmes comme UNIX, VXWORKS ou Windows-NT, il paraît très rudimentaire. Mais, comme nous allons le voir, sa simplicité est un atout.

Le système MS-DOS fut initialement prévu pour exécuter des applications 16 bits. Mais, avec l'apparition des processeurs 32 bits, fut introduite une extension au système baptisée *DOS Extender* qui permet d'exécuter des programmes contenant des instructions 32 bits. Ainsi, toute la puissance du processeur peut être exploitée. MS-DOS est un système monotâche. Cela signifie que le programme en cours d'exécution dispose de tout le temps processeur et ne peut être interrompu qu'en cas d'un événement émanant d'un périphérique tel que le clavier, l'horloge ou la souris. Le traitement de ces interruptions est très court et ne perturbe pratiquement pas l'exécution du programme principal. C'est là le principal avantage de MS-DOS. En effet, tous les systèmes d'exploitation modernes sont multitâches (UNIX, WINDOWS-NT, OS2...). Certains sont temps réels (VXWORKS, OS9, iRMX...). L'utilisation

d'un système d'exploitation multitâches qui n'est pas temps-réel pour une application d'asservissement où les contraintes temporelles sont très serrées est impossible. Par exemple, sous Windows-NT, une tâche qui se voit attribuer la priorité maximale est susceptible d'être interrompue pendant une durée de $50ms$ par le noyau du système. Ce temps correspond à l'exécution de tâches périodiques de maintenance du système qui ont une priorité absolue. Il est clair que si l'asservissement a une période d'échantillonnage de $20ms$, au moins 3 périodes seront manquées conduisant inmanquablement à un comportement saccadé. Ceci est inacceptable. Il reste l'option du système temps-réel. Outre le fait qu'un programme temps réel est plus difficile à mettre en oeuvre qu'un programme classique, le noyau temps réel, du fait de sa complexité, utilise une partie non négligeable de la puissance de calcul du processeur. Ainsi, on peut estimer à 20% la perte de puissance par rapport au système MS-DOS. Néanmoins, l'utilisation d'un tel système s'impose dans le cas où le nombre d'entrées et de sorties à gérer est tel que réaliser la même chose avec un programme écrit linéairement serait une gageure. Ce choix est donc principalement motivé par le degré de complexité du système. Dans notre cas, ce degré ne justifie pas l'utilisation d'un système temps réel.

Seuls les ordinateurs compatibles IBM-PC basés sur les processeurs Intel supportent le système MS-DOS. Ces ordinateurs sont des produits grand-public donc bon marché. La puissance de ces machines est certes moindre que celle des stations de travail mais la perspective d'une totale compatibilité efface largement une telle disparité. Il y a tout lieu de penser que cette compatibilité ascendante se prolongera encore durant de nombreuses années malgré l'abandon progressif de MS-DOS au profit de systèmes d'exploitation plus évolués et plus complexes. En effet, le standard matériel des «PC» a atteint une certaine stabilité depuis quelques années qui ne peut que se renforcer avec le temps.

Dans le cadre de ce travail, la flexibilité du système MS-DOS a pu être vérifiée. En 1996, la première expérience d'asservissement par vision mettait en oeuvre un PC à base de processeur Pentium cadencé à $90MHz$. La période d'échantillonnage de la boucle de vision était de $20ms$ et le processeur était occupé durant toute la période. En 1997, l'acquisition d'une nouvelle machine de vision à base de Pentium Pro à $200MHz$ a permis de réduire la période d'échantillonnage à $8.33ms$. Cette dernière machine n'est néanmoins pas assez puissante pour réaliser un suivi de profilé en raison notamment du traitement d'image plus conséquent. L'acquisition d'une machine basée sur un Pentium II cadencé à $300MHz$ a permis de régler immédiatement ce problème.

Il reste à déterminer si les «PC» respectent le cahier des charges sur un point : le temps de traitement de l'information visuelle. Le flux de données le plus important en terme de débit se situe à la sortie de la carte d'acquisition. En effet, une image digitale en 256 nuances de gris de résolution standard occupe environ $400Ko$. Les zones d'intérêts d'une image, qui sont les seules parties transférées lors de l'asservissement, couvrent environ 30% de l'image totale, soit $120Ko$. Une des contraintes qu'on s'est fixée est que le traitement d'image ne doit pas introduire un retard pur de plus d'une période d'échantillonnage. Une caméra CCIR standard fourni 50 trames par seconde

(une image = une trame paire + une trame impaire). L'asservissement par vision a donc une fréquence de 50Hz imposée par la caméra. Afin de laisser du temps au traitement d'image, le transfert de l'image ne doit pas prendre plus d'un cinquième de la période, soit 4ms . Le débit entre la carte et l'ordinateur hôte doit donc être d'au moins 30 Mo/s . Les machines compatibles IBM-PC sont actuellement pourvues d'un bus 32 bits dénommé «PCI». Ce bus permet de communiquer avec des cartes filles insérées dans des connecteurs situés sur la carte mère. Il dispose d'un mode de transfert appelé *bus master* où la carte fille prend le contrôle du bus et vient écrire directement dans la mémoire centrale de l'ordinateur. Durant ce transfert, le processeur est bloqué. Ce transfert peut s'effectuer à un débit maximum de 120 Mo/s . Ce débit est largement supérieur aux 30 Mo/s minimum imposés par le cahier des charges.

C.1.4 Choix de la structure logicielle

L'ordinateur hôte de la carte d'acquisition doit obligatoirement réaliser le traitement d'image afin de limiter au maximum le temps de transfert des images. De plus, nous imposons la synchronisation de tout l'asservissement sur la cadence d'acquisition des images. C'est donc la carte d'acquisition vidéo qui fournit le signal d'horloge pour l'asservissement. Aussi est-il naturel que ce soit l'ordinateur hôte de la carte d'acquisition qui réalise la tâche de contrôle de la boucle de vision puisqu'il a directement accès à l'information d'horloge *via* le bus PCI.

Cette structure se démarque par rapport à la structure classique où l'ordinateur hôte de la carte d'acquisition est chargé uniquement du traitement d'image et non du contrôle de la boucle de vision. Dans cette structure classique, le contrôle est confié à un autre système, le plus souvent géré par un système d'exploitation temps-réel tel que OS9, VXWORKS ou iRMX. Malheureusement, avec une telle configuration, la synchronisation de l'asservissement sur la période d'acquisition devient d'autant plus difficile que la fréquence d'échantillonnage augmente et que la liaison entre le système de vision et le système temps-réel est lente. Néanmoins, cette structure est entièrement justifiée dans le cas où, pour des raisons de puissance de calcul, les différents traitements doivent être réalisés en parallèle.

Nous avons tout de même choisi de confier les tâches de bas niveau – à savoir la gestion des entrées et des sorties avec le robot – à un système temps-réel. Ce choix a été motivé par une des contraintes du cahier des charges : en cas d'erreur du programme d'asservissement, le système doit se mettre dans un état de sécurité stable à partir duquel il peut rapidement et facilement sortir dès que le programme d'asservissement a été relancé. Notre principal souci fut de séparer les routines «sensibles» du reste du programme. La routine la plus sensible est celle qui écrit les consignes de vitesse \dot{q}^* dans les convertisseurs Numérique-Analogique (voir figure 2.12). En effet, l'envoi d'une valeur erronée peut induire des déplacements très rapides et incontrôlés des axes du robot et conduire à sa destruction. Aussi, cette partie de code doit être écrite avec beaucoup de soin et faire l'objet d'un minimum de modifications par la suite.

C.1.5 Organigramme du programme d'asservissement

La figure C.1 décrit le fonctionnement de l'asservissement du point de vu logiciel. Les organigrammes de la partie droite décrivent le fonctionnement des programmes du système temps réel. Il s'agit d'une routine principale de gestion des entrées/sorties avec le robot et de communication avec la machine de vision et d'une routine de chien de garde. Le chien de garde est un compteur qui est incrémenté à chaque période par la routine principale. La routine de chien de garde teste périodiquement si le compteur du chien de garde évolue. Si ce dernier reste figé, cela signifie que l'asservissement s'est arrêté intempestivement. Dans ce cas, par mesure de sécurité, les consignes de vitesse des axes du robot sont mises à 0. La partie gauche de la figure C.1 décrit le fonctionnement du programme d'asservissement par vision. Ce programme est purement séquentiel et s'inscrit dans une boucle qui n'est interrompue qu'à l'arrêt de l'asservissement. Au moment de l'arrêt, un code est envoyé au système temps réel à la place des commandes qui lui indique de se mettre dans un état de veille où le robot est mis hors tension. C'est au moment de l'initialisation de la boucle d'asservissement que la machine de vision envoi un signal de réveil au système temps-réel afin qu'il mette le robot sous tension. La programmation d'une boucle séquentielle impose de connaître avec précision la durée de chaque tâche élémentaire. De plus, aucune routine ne peut s'exécuter en tâche de fond avec une priorité moindre. Mais, en contrepartie, toute la puissance du processeur peut être employée pour le programme d'asservissement.

C.1.6 Chronogramme de l'asservissement par vision

La figure C.2 présente le diagramme temporel de l'asservissement. Les durées de toutes les tâches sont invariantes et connues avec précision. C'est la condition *sine qua non* pour une organisation optimale de toutes les tâches. En effet, en l'absence de mécanismes évolués de synchronisation tels que les sémaphores ou les boites aux lettres des systèmes temps-réel, seule la maîtrise rigoureuse de toutes les durées permet de réaliser un système robuste et fiable.

Ce chronogramme met en évidence que la majeure partie du temps processeur est consacré au transfert et au traitement de l'image. Le temps consacré au calcul de l'asservissement qui comprend le calcul d'un Jacobien image ou une reconstruction 3D, le calcul du Jacobien du robot et le calcul du correcteur apparaît négligeable en comparaison ($500\mu s$ vs entre 5 et 10ms). Cette différence s'explique par la différence du type de traitement à effectuer dans chaque cas. Le traitement d'image, qui se résume essentiellement à un filtrage par masque de convolution de type Sobel ou Canny [Can86], met en oeuvre des opérations arithmétiques élémentaires sur des entiers pour un volume important de données. Le calcul de l'asservissement implique des calculs matriciels en virgule flottante sur des réels codés sur 80bits (double) mais concernant un volume beaucoup moins important de données. De plus, ces calculs sont réalisés par un coprocesseur arithmétique dont la puissance permet de réaliser des opérations en virgule flottante avec la même rapidité que les opérations sur les

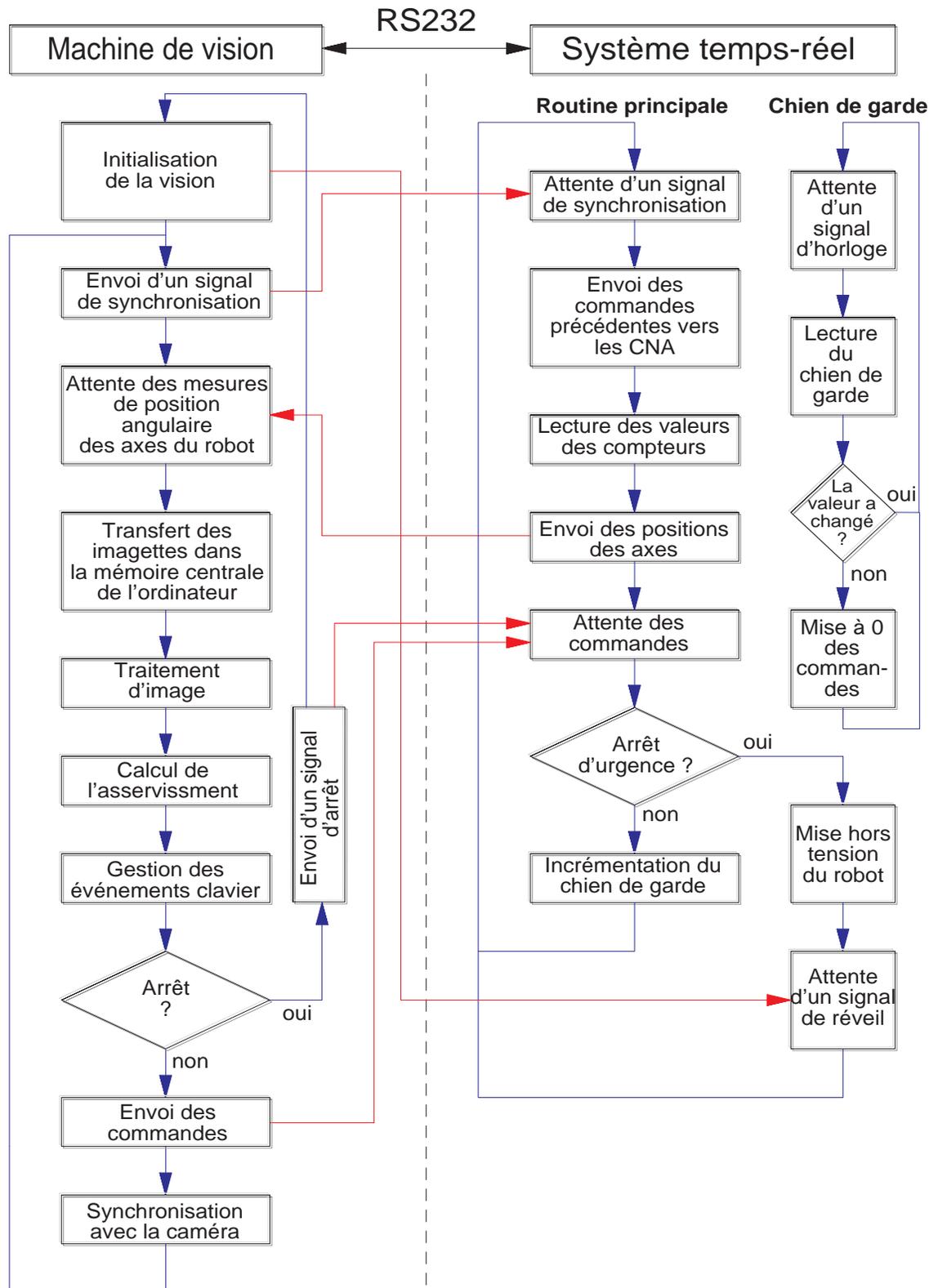


Fig. C.1: Organigramme du logiciel d'asservissement

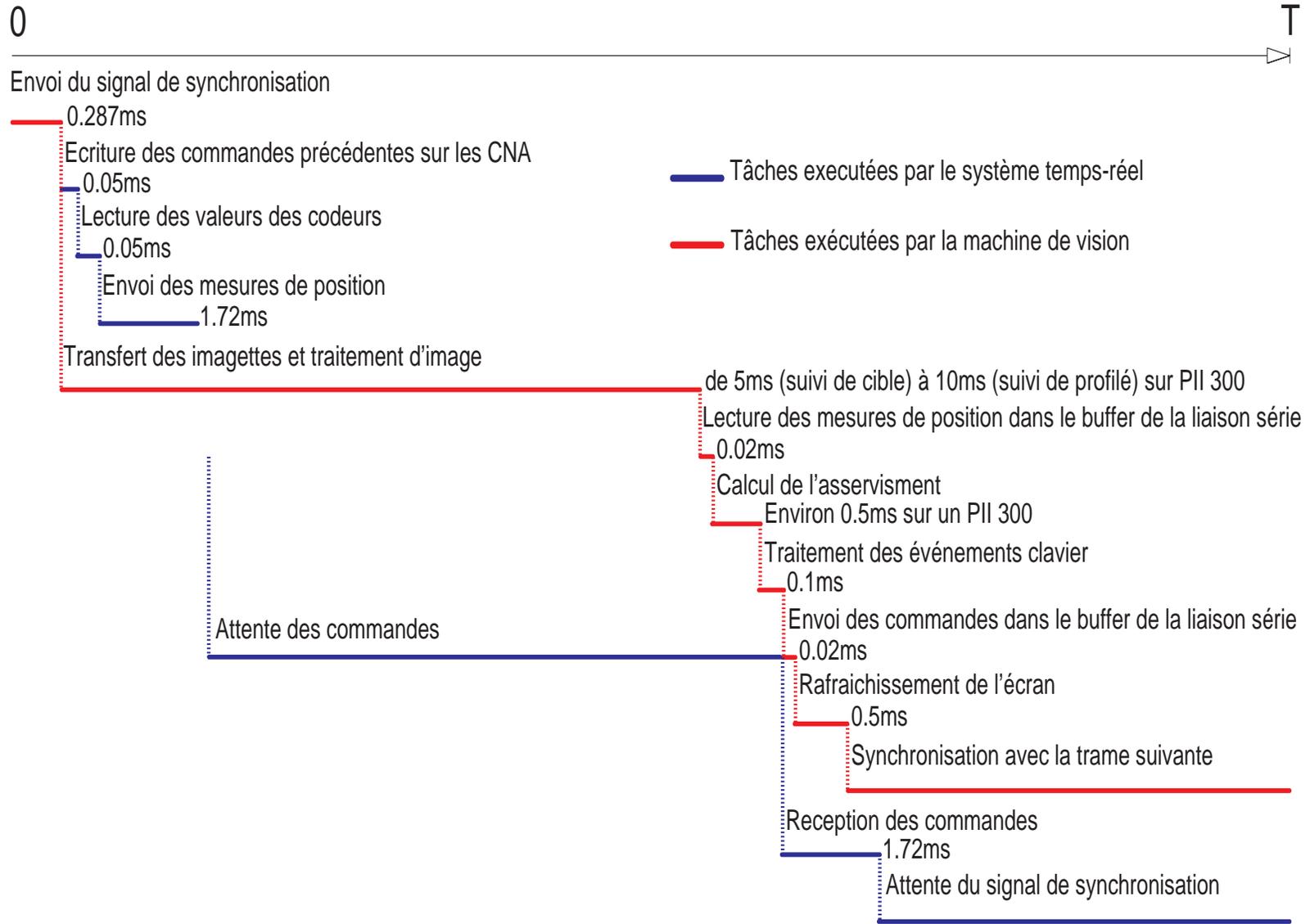


Fig. C.2: Chronogramme du programme d'asservissement

entiers.

On peut enfin constater sur ce chronogramme qu'une partie non négligeable du temps est consacré à la transmission sur la liaison série. En effet, la liaison s'opérant à la vitesse de 38400 bits/s, la transmission d'un octet de 8 bits avec un bit de début et deux bits de fin requiert donc $287\mu s$. Néanmoins, cette transmission est entièrement réalisée en tâche de fond par le matériel (UART 16650A), le temps processeur peut donc être consacré à une autre tâche.

BIBLIOGRAPHIE

- [Åst70] K.J. Åström. *Introduction to stochastic control theory*. Academic Press, New-York, USA, 1970.
- [Ben96] F. Bensalah. *Estimation du mouvement par vision active*. Thèse de doctorat, Université de Rennes 1, 1996.
- [BGW90] R. Bitmead, M. Gevers, et V. Wertz. *Adaptive optimal control, the thinking's man's GPC*. Prentice Hall, London, 1990.
- [Can86] J.F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 :769–798, 1986.
- [CG96] P.I. Corke et M.C. Good. Dynamic effects in visual closed-loop systems. *IEEE Transactions on Robotics and Automation*, 12(5) :671–683, 1996.
- [Cha90] F. Chaumette. *La relation vision-commande : théorie et application à des tâches robotiques*. Thèse de doctorat, Université de Rennes I, 1990.
- [Cha98] F. Chaumette. *Habilitation à diriger des recherches*. Université de Rennes I, 1998.
- [CMT87] D.W. Clarke, C. Mohtadi, et P.S. Tuffs. Generalized predictive control - part 1. The basic algorithm. *Automatica*, 23 :137–160, 1987.
- [Coi92] P. Coiffet. *La robotique – principes et applications*. Hermes, Paris, troisième édition, 1992.
- [Cor96] Peter I. Corke. *Visual control of robots*. Research Studies Press Ltd., Taunton, Somerset, U.K., 1996.
- [Cra89] J. Craig. *Introduction to robotics – Mechanics and control*. Addison-Wesley, Reading, Massachusetts, deuxième édition, 1989.
- [CS93] F. Chaumette et A. Santos. Tracking a moving object by visual servoing. Dans *Proceedings of the 12th IFAC World Congress*, pages 409–414, juillet 1993.
- [DD90] D. DeMenthon et L. S. Davis. Reconstruction of a road by local image matches and global 3D optimization. Dans *IEEE Int. conf. on robotics and automation*, tome 2, pages 1337–1342, mai 1990.
- [DD92] D. DeMenthon et L. S. Davis. Exact and approximate solutions of the perspective-three-point problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11 :1100–1105, 1992.

- [DP96] J-M Dion et D. Popescu. *Commande optimale, conception optimisée des systèmes*. Diderot, Paris, 1996.
- [DRLR89] M. Dhome, M. Richetin, J. Lapresté, et G. Rives. Determination of the attitude of 3D objects from a single perspective view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12 :1265–1278, 1989.
- [DRLR91] M. Dhome, M. Richetin, J.T. Lapresté, et G. Rives. Inverse perspective transformation using zero-curvature points—Application to the localization of some generalized cylinders from a single view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(2) :185–192, février 1991.
- [ECR92] B. Espiau, F. Chaumette, et P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3) :313–326, 1992.
- [Esp95] B. Espiau. *Sur les erreurs en asservissement visuel*. Rapports de recherche de l'INRIA no. 2619, juillet 1995.
- [FB81] M. A. Fischler et R. C. Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 :381–395, juin 1981.
- [FF94] J.-P. Ferrieux et F. Forest. *Alimentations à découpage, convertisseurs à résonance*. Masson, Paris, deuxième édition, 1994.
- [FM89] J. Feddema et O. Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Transactions on Robotics and Automation*, 5 :691–700, octobre 1989.
- [FMZ91] D. Forsyth, J.L. Mundy, et A. Zisserman. Invariant description for 3d object recognition and pose. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(10), octobre 1991.
- [GMA98] J. A. Gangloff, M. de Mathelin, et G. Abba. 6 DOF high speed dynamic visual servoing using GPC controllers. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2008–2013, 1998.
- [Gol94] R. Goldberg. Constrained pose refinement of parametric objects. *Int. Journal on Computer Vision*, 2 :181–211, 1994.
- [HCL89] R. Horaud, B. Canio, et O. Leboulloux. An analytic solution for the perspective 4-point problem. *Computer Vision Graphics Image Processing*, 1 :33–44, 1989.
- [HEK96] K. Hashimoto, T. Ebine, et H. Kimura. Visual servoing with hand-eye manipulator—optimal control approach. *IEEE Transactions on Robotics and Automation*, 12(5) :766–774, 1996.
- [HHC96] G. D. Hager, S. Hutchinson, et P. Corke. *A tutorial on visual servo control*. Tutorial TT3, 1996.
- [HLON91] R. M. Haralick, C. Lee, K. Ottenberg, et M. Nolle. Analysis and solutions of the three point perspective pose estimation problem. Dans *Proc.*

- of *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 592–598, 1991.
- [HP79] J. Hill et W. T. Park. Real time control of a robot with a mobile camera. Dans *Proceedings of the 9th ISIR*, pages 233–246, mars 1979.
- [HS93] R. M. Haralick et L. G. Shapiro. *Computer and robot vision*, tome 2. Addison-Wesley, Reading, Massachusetts, 1993.
- [KW90] K. Kanatani et K. Watanade. Reconstruction of 3D road geometry from images for autonomous land vehicles. *IEEE Transactions on Robotics and Automation*, 6(1), février 1990.
- [LHF90] Y. Liu, T. S. Huang, et O. D. Faugeras. Determination of camera location from 2D to 3D line and point correspondences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1 :28–37, 1990.
- [Lju87] L. Ljung. *System identification – Theory for the user*. Prentice-Hall, Englewood Cliffs, 1987.
- [Léo90] F. Léonard. *Contributions à la commande dynamique d'un robot industriel en boucle fermée par caméra embarquée*. Thèse de doctorat, Université Louis Pasteur de Strasbourg, 1990.
- [Low91] D. G. Lowe. Fitting parametrized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5 :441–450, 1991.
- [LWH98] F. Lange, P. Wunsch, et G. Hirzinger. Predictive vision based control of high speed industrial robot paths. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2646–2651, 1998.
- [Mal98] E. Malis. *Contributions à la modélisation et à la commande en asservissement visuel*. Thèse de doctorat, Université de Rennes I, 1998.
- [MCB98] E. Malis, F. Chaumette, et S. Boudet. Positioning a coarse-calibrated camera with respect to an unknown object by 2D 1/2 visual servoing. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1352–1359, 1998.
- [MDGD92] P. Martinet, N. Daucher, J. Gallice, et M. Dhome. Robot control using 3d monocular pose estimation. Dans *IEEE/RSJ Int. conf. on intelligent robots ans systems*, tome 4, pages 1–12, 1992.
- [MMB98] G. Morel, E. Malis, et S. Boudet. Impedance based combination of visual and force control. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1743–1748, 1998.
- [MR93] H. Michel et P. Rives. *Singularities in the determination of the situation of a robot effector from the perspective view of 3 points*. Rapports de recherche de l'INRIA no. 1850, 1993.
- [NI98] Y. Nakabo et M. Ishikawa. Visual impedance using 1 ms visual feedback system. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2333–2338, 1998.

- [NK94] B. Nelson et P. K. Khosla. Integrating sensor placement and visual tracking strategies. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1351–1356, 1994.
- [Pap92] N. P. Papanikolopoulos. *Controlled Active Vision*. Thèse de doctorat, Carnegie Mellon University, 1992.
- [PK93] N.P. Papanikolopoulos et P. K. Khosla. Adaptive robotic visual tracking : theory and experiments. *IEEE Transactions on Automatic control*, 38(3) :429–445, mars 1993.
- [PKK93] N.P. Papanikolopoulos, P. K. Khosla, et T. Kanade. Visual tracking of a moving target by a camera mounted on a robot : combination of control and vision. *IEEE Transactions on Robotics and Automation*, 9(1) :14–35, 1993.
- [Sag97] H. G. Sage. *Synthèse de correcteurs \mathcal{H}_∞* . Thèse de doctorat, Université Louis Pasteur de Strasbourg, 1997. VDI Verlag.
- [SBC96] V. Sundareswaran, P. Bouthemy, et F. Chaumette. Exploiting image motion for active vision in a visual servoing framework. *Int. Journal of robotics research*, 15(6) :629–645, décembre 1996.
- [SI73] Y. Shirai et H. Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5 :99–108, 1973.
- [SMA⁺99] H. Sage, M. de Mathelin, G. Abba, J. Gangloff, et E. Ostertag. Synthesis of robust \mathcal{H}_∞ controllers for industrial robot manipulators using nonlinear optimization. *Journal of Engineering design and automation*, 5(1), février 1999.
- [SN94] G. Segulier et F. Notelet. *Electrotechnique industrielle*. Lavoisier, Paris, deuxième édition, 1994.
- [Spo87] M.W. Spong. Modeling and control of elastic joint robots. *Journal of Dynamic Systems, Measurement and control*, 109 :310–319, septembre 1987.
- [SV89] M. W. Spong et M. Vidyasagar. *Robot dynamics and control*. Wiley, New York, 1989.
- [SW80] A.C. Sanderson et L.E. Weiss. Image-based visual servo control using relational graph error signals. Dans *Proc. IEEE Int. Conf. on Cybern. and Soc.*, pages 1074–1077, octobre 1980.
- [Tsa87] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses. *IEEE Journal of Robotics and Automation*, 3(4) :323–344, 1987.
- [Wei84] L. Weiss. *Dynamic visual servo control of robots : an adaptive image-based approach*. Thèse de doctorat, Carnegie-Mellon University, 1984.
- [WHB96] W. J. Wilson, C. Williams Hulls, et G. S. Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5) :684–696, 1996.

-
- [WS87] L.E. Weiss et A.C. Sanderson. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5) :404–417, octobre 1987.
- [Yua89] J.S.C. Yuan. A general photogrammetric method for determining object position an orientation. *IEEE Transactions on Robotics and Automation*, 5(2), avril 1989.