

Une nouvelle approche de la relation vision-commande en robotique

Bernard Espiau, François Chaumette, Patrick Rives

▶ **To cite this version:**

Bernard Espiau, François Chaumette, Patrick Rives. Une nouvelle approche de la relation vision-commande en robotique. [Rapport de recherche] RR-1172, INRIA. 1990. inria-00075386

HAL Id: inria-00075386
https://inria.hal.science/inria-00075386

Submitted on 24 May 2006

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

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

INRIA

UNITE DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Original

Rapports de Recherche

N° 1172

Programme 6
Robotique, Image et Vision

UNE NOUVELLE APPROCHE DE LA RELATION VISION-COMMANDE EN ROBOTIQUE

Bernard ESPIAU
François CHAUMETTE
Patrick RIVES

Mars 1990

RR-1172

UNE NOUVELLE APPROCHE DE LA RELATION VISION-COMMANDE EN ROBOTIQUE

A NEW APPROACH TO VISUAL BASED SERVOING IN ROBOTICS

Bernard Espiau*, François Chaumette**, Patrick Rives***

* *ISIA-ENSMP, rue Claude Daunesse, Sophia Antipolis, 06565 Valbonne, France*

** *IRISA-INRIA campus de Beaulieu 35042 Rennes-cedex, France*

*** *INRIA-centre de Sophia Antipolis, 2004 Route des Lucioles, 06565 Valbonne, France*

Résumé

Ce rapport présente les concepts et les résultats relatifs à une nouvelle approche de la commande référencée vision en robotique. L'idée de base consiste à considérer un système de vision comme un capteur particulier associé à une tâche, et inséré dans une boucle de commande. Une fois effectuée l'etape nécessaire de modélisation, la problématique ainsi posée ressort alors de l'automatique, et met en avant des questions naturelles de stabilité et de robustesse.

Ce rapport est structuré comme suit: après un bref état de l'art en matière d'asservissement visuel, nous définissons les notions de base permettant la modélisation des interactions considérées. Le concept de *Torseur d'Interaction* est ainsi introduit, et son application aux images est ensuite effectuée, à travers la définition de quelques motifs essentiels qui constituent une représentation de l'objectif poursuivi. Le cadre général de la commande est ensuite décrit, à partir de la notion de fonction de tâche, et un rappel des résultats de stabilité de Claude SAMSON est donné. Le concept de tâche hybride est ensuite développé, et l'application aux capteurs visuels est explicitée.

Le rapport s'achève sur la présentation de divers résultats, expérimentaux et de simulation, et quelques lignes directrices pour de futures études sont tracées en conclusion.

Abstract

This research report describes some concepts and results related to a new approach of vision-based control in robotics. The basic idea consists in considering a vision system as a specific sensor dedicated to a task, and included in a control servo-loop. Once the necessary modelling stage is performed, the framework becomes the one of automatic control, and naturally stability and robustness questions arise.

This report is organized as follows: in the introduction, a short state-of-the-art in the area of visual servoing is given. Then, the basic concepts allowing to model the concerned interactions are given. The *Interaction Screw* is thus defined in a general way, and the application to images follows. Some essential image features, which are aimed to represent the user's goals are defined. Starting from the concept of task function, the general framework of the control is then described, and some stability results are recalled. The concept of hybrid task is also presented and then applied to visual sensors.

The report ends with the presentation of several simulation and experimental results, and some guidelines for future work are drawn in the conclusion.

Table des matières

1 Introduction 3

2 Modélisation 7
- 2.1 Généralités . . . 7
  - 2.1.1 Notations et rappels . . . 7
  - 2.1.2 Le torseur d'interaction . . . 8
  - 2.1.3 Notion de liaison virtuelle . . . 9
- 2.2 Cas du capteur d'images . . . 9
  - 2.2.1 Généralités . . . 9
  - 2.2.2 Quelques exemples de motifs usuels . . . 14

3 La commande 20
- 3.1 Cadre général . . . 20
  - 3.1.1 Le concept de fonction de tâche . . . 20
  - 3.1.2 Commande et stabilité . . . 21
  - 3.1.3 Tâches hybrides . . . 23
- 3.2 Application à l'utilisation d'un capteur visuel . . . 25

4 Quelques résultats expérimentaux 28
- 4.1 Une commande cinématique simplifiée . . . 28
- 4.2 Résultats de simulation . . . 29
  - 4.2.1 Positionnement . . . 30
  - 4.2.2 Suivi de route . . . 32
  - 4.2.3 Positionnement à l'entrée d'un tuyau . . . 35
- 4.3 Expérience réelle . . . 36

5 Conclusion 38

Chapitre 1

Introduction

L'utilisation de la vision a toujours été l'un des thèmes majeurs de la robotique. Les premières études en ce domaine ont principalement été motivées, dans les années 1970, par des problèmes de reconnaissance de formes. Les résultats de ces travaux, associés à une remarquable amélioration des moyens informatiques disponibles, font que l'on peut considérer que reconnaître, localiser ou inspecter une pièce sans mouvement font aujourd'hui partie de l'état de l'art industriel. Parallèlement, et en partie en raison de l'intérêt porté aux robots mobiles, les chercheurs se sont intéressés à des questions plus complexes : vision stéréoscopique, scènes d'extérieur par exemple. Par ailleurs, d'importants travaux ont été effectués dans le domaine du traitement de séquences d'images (vision dynamique) [BOU 88]. Les motivations originelles provenaient d'applications de télécommunications (codage avec compensation de mouvement), ou militaires (poursuite de cibles, reconnaissance d'objets en mouvement). L'algorithmique associée, du type traitement de signal, permettait d'espérer à moyen terme des traitements dits "temps réel", c'est-à-dire à la cadence vidéo.

Si l'on revient à présent à la robotique, il apparaît que s'est développée ces dernières années une tendance à l'utilisation de capteurs extéroceptifs au sein même de boucles de commande (actions réflexes), et non pas seulement comme informations utilisables dans des procédures de haut niveau. Une telle approche s'avère très utile lorsque l'on doit par exemple compenser de petites erreurs de positionnement, saisir des objets au vol sur un tapis roulant, suivre un joint de soudure, ou s'adapter à des incertitudes sur l'environnement. Des applications de ce type ont été réalisées aussi bien en robotique de manipulation qu'en téléopération ou robotique mobile. Toutefois, les capteurs utilisés étaient majoritairement du type proximètres ou "range finder" (acoustique par exemple). Or, il se trouve qu'il est également possible de considérer un capteur visuel embarqué comme étant un dispositif permettant de fournir les informations nécessaires à la réalisation de telles boucles de commande.

Compte tenu des particularités de la vision artificielle, il est ainsi nécessaire pour cette classe d'applications :

- d'être capable d'extraire de l'image des informations suffisantes pour réaliser la tâche désirée (en général de positionnement). Cela peut induire la conception de cibles particulières;

- de développer des algorithmes de commande suffisamment simples pour pouvoir

fonctionner à une cadence compatible avec la bande passante du système en boucle fermée, mais cependant suffisamment robustes pour tolérer les inévitables incertitudes et inconnues sur le capteur et l'environnement.

Ce contexte, souvent appelé "Asservissement Visuel" (ou "visual servoing") est celui dans lequel se situe le présent article.

Notons que cette approche est de nature totalement différente de celle regroupée généralement sous le terme "Vision Dynamique" qui exploite la connaissance du mouvement du capteur mais sans le contrôler. Dans l'approche "Asservissement Visuel", le but consiste à définir une configuration particulière d'éléments caractéristiques dans l'image, qui constituera une *image cible* à atteindre, puis, partant d'une image courante prise sous un point de vue différent, de construire une commande permettant à la caméra de converger vers le point de vue correspondant à cette image cible.

Les premiers travaux, qui ont vu le jour dans la fin des années 70, se caractérisaient par des démarches heuristiques. On peut citer, par exemple, [AGI 79], [GIL 80]. Des travaux plus formalisés sont apparus dès l'année 82, mettant en avant trois types de problèmes :

- Le choix et l'extraction des informations visuelles utilisées dans la commande.
- Le choix des stratégies de commandes.
- L'analyse et la synthèse des lois de commande du point de vue de l'automatique.

Concernant le premier point, les éléments caractéristiques constituant l'image cible ont été choisis le plus souvent en fonction de l'existence préalable d'algorithmes permettant de les extraire en un temps raisonnable. De ce fait, il n'est pas étonnant que la majorité des travaux porte sur l'utilisation de primitives *contour* : points, segments de droite [RIV 87], [ESP 87], [FEN 79], ou de primitives *région* : surface, barycentre, axes principaux d'inerties [GIL 80], [WEI 84]. On peut citer également une approche originale [SAN 80] où l'image est décrite en termes de graphe relationnel dont les nœuds représentent la surface des faces et les arcs, la longueur des arêtes. De plus, une connaissance a priori sur la structure géométrique de la scène 3D est toujours supposée (par exemple: l'objet observé est un cube ayant une longueur d'arête connue). Dans la quasi totalité des travaux cités, les équations reliant le mouvement de la caméra et la structure 3D de la scène au mouvement observé dans l'image (ce que nous appellerons par la suite *torseur d'interaction*), ne sont pas établies explicitement. Par contre, l'application directe des équations de la perspective associée à une démarche intuitive suffisait à déterminer des relations simples entre la variation des primitives observées et certaines directions privilégiées dans le mouvement de la caméra, par exemple : accroissement ou diminution de la surface et mouvement translationnel suivant l'axe optique ou encore, position du centre de gravité dans l'image et mouvement de roulis et de lacet de la caméra [GIL 80].

La première étude relativement complète sur l'aspect "commande" de l'approche "Asservissement Visuel", a pour origine le Laboratoire de Robotique de Carnegie Mellon [SAN 83], [WEI 84], [WEI 87]. Les stratégies de contrôle sont répertoriées suivant quatre types de schéma :

- **Static Look-and-Move** : Le contrôle du robot est assuré par un asservissement classique en position dans l'espace cartésien, la consigne en position étant calculée à partir de l'information extraite de l'image. Le mode de fonctionnement consiste en un enchainement de séquences du type : le robot observe la scène, analyse l'image, calcule un déplacement de la caméra, l'exécute et attend l'arrêt de la caméra avant l'acquisition d'une nouvelle image. Les algorithmes de traitement d'images et ceux de génération de la commande sont considérés comme deux processus séquentiels.

- **Dynamic Look-and-Move** : C'est une structure identique à la précédente hormis le fait que calcul de la commande et traitement de l'image peuvent être réalisés en simultané bien que, la plupart du temps, à des cadences différentes.

- **Position-Based Visual Servoing** : Cette stratégie ainsi que la suivante se caractérise par la suppression de la boucle de retour de l'asservissement calculée à partir des données articulaires. La stabilité du robot est maintenant assurée uniquement par un calcul de la position du robot à partir des données fournies par la caméra.

- **Image-Based Visual Servoing** : Cette dernière stratégie, qui est celle que nous développerons par la suite, consiste à spécifier et à contrôler la tâche non plus en termes de régulation de la position de l'organe terminal mais plutôt en termes de régulation de grandeurs mesurées dans l'image vis à vis d'une image cible à atteindre.

Jusqu'à récemment la plupart des travaux relevait des deux premières catégories. Cela est dû principalement au temps de calcul prohibitif des algorithmes de traitement d'images qui excluait une fréquence d'échantillonnage suffisante pour assurer la stabilité de la commande. Du fait des progrès considérables réalisés dans le domaine des architectures de traitement d'images, il est maintenant possible de mettre en œuvre les deux dernières stratégies. Dans [FED 89b] et [COR 89], des résultats prometteurs ont été obtenus dans cette voie.

La partie *analyse et synthèse des lois de commande* reste l'un des domaines où les résultats sont les moins avancés. Encore une fois, c'est à Carnegie Mellon que l'on trouve les travaux les plus significatifs. Les deux premières stratégies sont relativement classiques et ne présentent pas un grand intérêt du point de vue de la commande. Par contre, les deux dernières ont été le sujet de deux types de recherche : l'étude de l'application entre l'espace des torseurs associés au mouvement de la caméra et l'espace des champs de vitesse observés entre deux images, le deuxième axe d'études étant la synthèse de la commande et l'analyse de sa stabilité. Concernant le premier point, les travaux ont portés principalement sur la nature de l'application mettant en correspondance le mouvement et la structure géométrique dans la scène tridimensionnelle avec le mouvement et la structure géométrique observés dans l'image. Des résultats ont été obtenus concernant la *sensibilité* vis à vis du choix des arrangements de motifs dans l'image [WEI 87], [FED 89a], [KAB 88]; ces résultats portent sur des cas simplifiés : par exemple, mouvements purement translationnels de la caméra, restriction au cas où l'application est régulière. Il faut noter également qu'aucune méthodologie générale

permettant de dériver une forme analytique de cette application n'est proposée (dans [KAB 88], par exemple, le gradient de l'application est estimé par une méthode directe de perturbations). Pour ce qui est de l'analyse des lois de commande et de la synthèse du contrôleur, elles ont été abordées sous trois aspects : découplage, commande adaptative, commande par modèle de référence. Ces trois points sont discutés dans [WEI 87]. Une analyse de stabilité aboutissant à la synthèse d'un contrôleur peut être également trouvée dans [WEI 84] et [COR 89].

Il se trouve que les travaux cités en référence s'intéressent en général assez peu au choix des attributs visuels à utiliser en relation avec les applications. Par ailleurs, l'aspect commande ne s'inscrit en général pas dans une analyse de type robotique, c'est-à-dire, prenant en compte certains aspects spécifiques, tels les effets des erreurs de modélisation et des approximations.

Cet article se propose donc d'apporter au problème de l' "Asservissement visuel" un point de vue plus général, sous deux aspects : un premier point consistera à proposer une amorce de méthodologie générale pour la conception de tâches utilisant un capteur visuel dans leur boucle de commande. Cette étape de modélisation sera alors suivie par une prise en compte de l'aspect commande, principalement abordé sous l'angle "robustesse". Cette analyse, partiellement décrite dans [RIV 89], s'appuie sur une approche générale originale de la commande des robots, présentée de façon approfondie en [SAM 90b]. La commande a pour objectif, dans notre cas précis, le positionnement robuste d'une caméra embarquée vis à vis de l'environnement, la tâche étant alors directement exprimée en termes d'écart par rapport à une image cible.

L'article est structuré ainsi : une première partie (section 2) est consacrée à l'aspect modélisation. Après le rappel d'un modèle général des interactions entre un capteur et son environnement, le cas particulier des images est abordé et quatre exemples significatifs d'attributs sont traités : les points, les droites, les cercles et les sphères. La deuxième partie (section 3) de l'article s'intéresse aux aspects commande : en section 3.1, les concepts fondamentaux sont rappelés : fonction de tâche, redondance et tâche hybride, schéma de commande, analyse de stabilité. La section 3.2 est consacrée à l'application de cette approche au cas de la vision, à travers l'angle "robustesse". Enfin la section 4 présente quelques résultats de simulation et expérimentaux.

Chapitre 2

Modélisation

2.1 Généralités

2.1.1 Notations et rappels

Nous utiliserons dans la suite les quelques notations mathématiques suivantes :

Soit e l'espace affine euclidien à trois dimensions, l'espace vectoriel associé étant $\mathbb{R}^3$. L'espace de configuration d'un corps rigide (et d'ailleurs celui des repères) est le groupe de Lie des déplacements de e, appelé traditionnellement SE_3 (Special Euclidean Group) isomorphe à $\mathbb{R}^3 \times SO_3$ où SO_3 est le groupe des rotations. C'est une variété différentielle de dimension 6. Un élément de SE_3, appelé une "situation" (c'est à dire position et orientation en raison de l'isomorphisme précédent) est noté $\bar{r}$. L'espace tangent à SE_3 en l'identité est simplement noté se_3, et son dual, ou espace cotangent, se_3^*. se_3 est une algèbre de Lie isomorphe à l'algèbre de Lie des champs équiprojectifs de e dans $\mathbb{R}^3$, ce qui signifie que tout élément (champ) de se_3 n'est autre que le traditionnel torseur. Classiquement, un torseur H est aussi défini par son vecteur u et la valeur de son champ en un point O de e. On notera donc $H = (H(O), u)$.

Lorsque l'on sera amené à choisir des repères, ils seront notés génériquement F. Un torseur donné exprimé dans F est alors un vecteur de $\mathbb{R}^6$. Enfin, la vitesse d'un repère F_i par rapport à un repère F_j est notée T_{ij}. C'est aussi le classique torseur cinématique.

Le produit de deux torseurs est l'application bilinéaire associée à $\begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix}$. Il s'écrit, quelque soit le point O considéré :

$$H_1 \bullet H_2 = < u_1, H_2(O) > + < u_2, H_1(O) > \tag{2.1}$$

où $<,>$ est le produit scalaire usuel entre deux vecteurs de $\mathbb{R}^3$. Soit S un espace de torseurs. Le produit de torseurs induit un isomorphisme entre S et son dual S^*. La forme linéaire associé Φ est :

$$\Phi_{(H(O),u)}(H_1(O), u_1) = (H(O), u) \bullet (H_1(O), u_1)$$

Ainsi, S^* est lui-même un espace de torseurs.

2.1.2 Le torseur d'interaction

Ces préliminaires étant posés, intéressons-nous d'une façon générale à la notion de capteur. Nous restreindrons en fait notre étude au cas où :

- Hypothèse H_1 : un capteur est **entièrement défini par la donnée d'une application différentiable de SE_3 dans $\mathbb{R}^p$.**

Cette hypothèse implique en particulier que, pour un capteur donné, seules des modifications de nature géométrique de l'environnement sont susceptibles de faire varier la sortie du capteur. Lions à présent un repère F_T à l'environnement utile, c'est-à-dire à la partie de l'environnement "perçue" par le capteur, et un autre, F_S, au capteur lui-même. Le repère de référence peut être indifféremment F_S ou F_T, ou même, dans le cas où l'environnement est géométriquement invariant au cours du temps, n'importe quel repère F_o associé. s peut donc s'écrire $s(F_S, F_T)$. Allons à présent un peu plus loin et supposons que la mobilité du capteur est obtenue à travers un système de coordonnées généralisées, q, réalisé physiquement, qui constitue une carte de SE_3. Alors, si les objets perçus sont eux-mêmes mobiles, de façon autonome, s s'écrit aussi $s = s(q,t)$, le paramètre temps t, représentant cette contribution du mouvement des objets. Les 6 coordonnées q_i sont par exemple relatives aux articulations d'un manipulateur rigide portant la caméra. A noter qu'une sous-variété de SE_3 peut également être considérée : c'est le cas par exemple d'un robot mobile dans le plan, à liaisons supposées holonomes.

Examinons à présent une composante s_j de s. En vertu des préliminaires mathématiques énoncés plus haut, et de la définition de s_j, nous savons que sa différentielle au point $\bar{r}$, $ds_{j|_{\bar{r}}}$, est une application linéaire de $se_{3|_{\bar{r}}}$ dans $\mathbb{R}$. Par ailleurs, on sait que la différentielle de toute fonction analytique d'une variété M dans $\mathbb{R}$ peut être identifiée avec un élément de l'espace cotangent. Dans notre cas, ceci implique que la différentielle de s_j en $\bar{r}$ n'est autre qu'un élément de se_3^*, c'est à dire un torseur. Rappelant qu'un élément de se_3 est aussi un torseur cinématique T, nous pouvons finalement écrire au point $\bar{r}$ de SE_3 la relation fondamentale :

$$\dot{s}_j = H_j \bullet T_{ST} \tag{2.2}$$

où :

- T_{ST} est la vitesse relative du repère F_T par rapport au repère F_S
- $\bullet$ est le produit de torseurs défini plus haut
- H_j est un torseur, dont l'expression dépend à la fois des caractéristiques de l'environnement et du capteur lui-même. Il caractérise complètement les interactions entre capteur et environnement et nous l'appellerons donc "Torseur d'interaction". Nous verrons par la suite qu'il joue un rôle important aussi bien dans les aspects modélisations que dans la commande elle-même.

2.1.3 Notion de liaison virtuelle

Soit T^* un mouvement virtuel au point $\bar{r}$ laissant invariante la sortie capteur s_j. T^* est solution de l'équation :

$$H_j \bullet T^* = 0 \tag{2.3}$$

et est donc un torseur réciproque de H_j.

Revenons à présent au capteur complet, caractérisé par sa sortie s de dimension p. L'ensemble des mouvements T^* laissant s invariant est alors le sous-espace S^* réciproque du sous-espace S de torseurs engendré par l'ensemble $\{H_1 \cdots H_j \cdots H_p\}$.

Plus précisément, nous pouvons alors dire qu'un ensemble de contraintes compatibles et indépendantes, de la forme $s(\bar{r}) - s_j = 0$ constitue une **liaison virtuelle** entre les objets de l'environnement (T) et le capteur (S). A une position $\bar{r}$ où ces contraintes sont satisfaites, la dimension N de l'espace réciproque de S dans se_3 est appelée la **classe** de la liaison virtuelle au point $\bar{r}$.

Cette notion est clairement une extension immédiate du concept de liaison entre solides tel qu'il est classiquement défini en théorie des mécanismes. La notion de liaison virtuelle, qui peut d'ailleurs se transformer en liaison réelle dans le cas de capteurs de contact, nous permettra ainsi de spécifier les tâches robotiques référencées capteurs que nous souhaitons réaliser d'une façon à la fois simple et cohérente avec la démarche classique connue sous le nom de commande hybride.

Remarques :

- Soit $m = 6 - N$. Lorsque $m = p$, la dimension du signal s est correcte, ce qui signifie intuitivement qu'elle correspond bien au nombre de "degrés de liberté" que l'on peut et souhaite contrôler à partir de s. Le cas $p > m$ présente souvent cependant un intérêt pratique, par exemple par les effets de filtrage qu'il peut impliquer. Nous ne l'excluerons donc pas, et le prendrons même en compte explicitement dans la partie commande.

- Avec un abus de notation évident, la relation (2.2) peut aussi s'écrire :

$$\dot{s}_j = L_j^T \, T_{ST} \quad \text{où} \quad L_j^T = H_j \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix} \tag{2.4}$$

  L_j^T est la représentation matricielle du torseur d'interaction H_j pour un repère F et un point O donnés. De même, la représentation matricielle de l'ensemble $\{H_1 \cdots H_p\}$ est appelée "Matrice d'Interaction" et notée L^T. Avec le même abus de notation, on peut également noter que $S^* = \text{Ker } L^T$.

2.2 Cas du capteur d'images

2.2.1 Généralités

Considérons un modèle idéal de caméra, dont la géométrie est de type projection perspective, tel que représenté Figure 1.

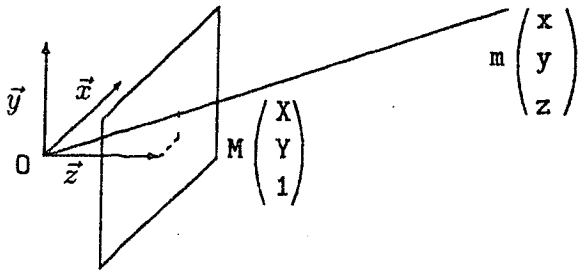


Figure 1 : Modèle de la caméra

Le repère $(O, \vec{x}, \vec{y}, \vec{z})$ donné Figure 1 sera, par défaut, celui dans lequel seront exprimées toutes les grandeurs nécessaires, par exemple, le torseur cinématique $(v(O), \omega)$, noté par la suite T, les coordonnées des points, les torseurs d'interaction, etc.

Sans la moindre perte de généralité, la distance focale est supposée égale à 1, de sorte que tout point m de e ayant pour coordonnées $\bar{x} = (x\, y\, z)^T$ se projette sur le plan image en M de coordonnées $\bar{X} = (X\, Y\, 1)^T$ avec :

$$\bar{X} = \frac{1}{z}\, \bar{x} \tag{2.5}$$

- **Remarque** : A ce modèle géométrique, il conviendrait d'ajouter un modèle photométrique. Toutefois, nous ne considérerons pas dans cet article que des mesures photométriques puissent être utilisées en tant que signaux capteurs, tels que définis au chapitre 2.1. En effet, notre hypothèse H_1 n'est généralement pas satisfaite avec de telles mesures, en raison de leur haute sensibilité aux conditions d'éclairement, qui peuvent varier indépendamment de tout déplacement. Lorsque le contrôle précis de celle-ci est possible, nous sommes alors quasiment dans le domaine de la vision avec capteurs actifs, qui n'est point notre propos.

  D'autre part, le calcul du torseur d'interaction s'avèrerait extrêmement hasardeux avec des mesures photométriques, en raison de fortes incertitudes de modélisation : les hypothèses simplificatrices usuelles (surface parfaitement Lambertienne, éclairage diffus) sont généralement complètement irréalistes et les erreurs induites rendraient ainsi non significatif le calcul de H.

Considérons, dans e, un solide rigide auquel peuvent être liées des primitives 3D (points, lignes, coins, etc...). Nous appellerons "motif dans la scène" un tel ensemble de primitives. Une configuration de ce motif est un élément P de l'ensemble $\mathcal{P}$ de toutes ses configurations possibles. Lorsque ce motif caractérise complètement la situation du corps auquel il est lié, la dimension de $\mathcal{P}$ est égale à 6.

Soit U un ouvert de $\mathcal{P}$ contenant P et ϕ un système local de coordonnées tels que (U, ϕ) forme une carte de $\mathcal{P}$ autour de P. On notera $w = \phi(P)$; $w \in \mathbb{R}^n$.

Soit l'application "projection perspective" notée f, telle que $f(P) = Q \in \mathcal{Q}$; $\dim \mathcal{Q} = m < n$. Q est un "motif dans l'image". On suppose alors :

- Hypothèse H_2 : P appartient à un ouvert $U' \subseteq U$ de $\mathcal{P}$ tel que dans U' le jacobien de f soit surjectif.

On note $V = f(U')$. Soit (V, ψ) une carte locale de $\mathcal{Q}$ autour de Q : $\bar{q} = \psi(Q)$; $\bar{q} \in \mathbb{R}^m$. Alors, $\bar{q} = \psi \circ f \circ \phi^{-1}(w)$, et le Jacobien $\frac{\partial \bar{q}}{\partial w}$ est de rang plein $\forall P \in U'$.

Enfin, le groupe des déplacements opère sur $\mathcal{P}$ par l'application δ de telle sorte que $\bar{q} = \psi \circ f \circ \delta(\bar{r})$. On fait alors l'hypothèse suivante :

- Hypothèse H_3 : les déplacements sont restreints à un ouvert W de SE_3 tel que $\delta(W) \subseteq U'$.

En pratique, le choix d'une carte pour $\mathcal{P}$ peut s'avérer inutilement contraignant. On utilisera donc souvent un paramétrage $\bar{p}$ de dimension $n' > n$ et l'on supposera alors :

- Hypothèse H_4 : l'application $P = g(\bar{p})$ est différentiable et de jacobien surjectif $\forall P \in \mathcal{P}$.

Un cas particulier est évidemment $w = \bar{p}$. Nous choisirons alors comme "signaux capteurs", une fonction $s = \sigma(\bar{q})$, $\forall \bar{q} \in \psi(V)$, telle que σ soit un difféomorphisme de V dans un ouvert de $\mathbb{R}^m$, le cas le plus fréquent étant $s = \bar{q}$.

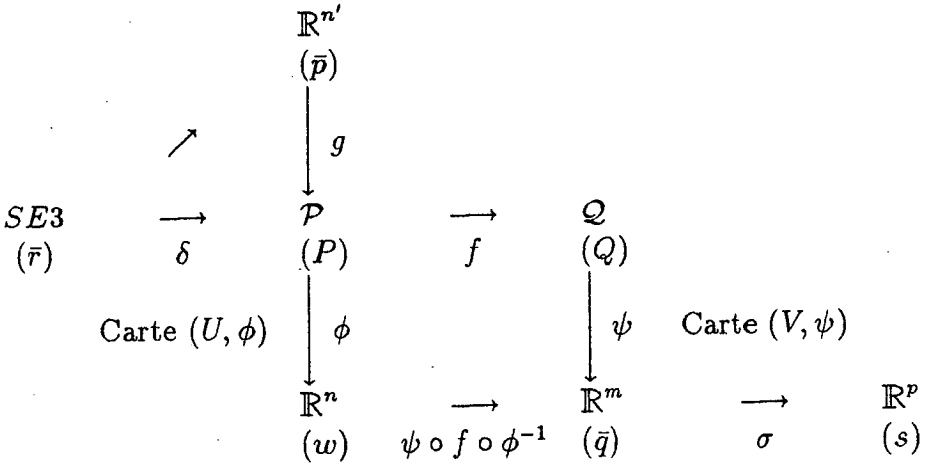


Figure 2

Les diverses hypothèses et définitions ci-dessus permettent de se placer dans un cadre de travail convenable. Lorsqu'elles sont respectées, elles ont en fait pour l'utilisateur les conséquences suivantes :

- la paramétrisation $\bar{p}$ est complète et unique, au sens où une seule est nécessaire pour représenter n'importe quelle configuration du motif dans $\mathcal{P}$,
- la paramétrisation de $\bar{q}$ est minimale,
- les applications σ, σ^{-1}, $\psi \circ f \circ \phi^{-1}$, $\psi \circ f \circ \delta$, $\psi \circ f \circ g$, etc... sont différentiables et leur matrice jacobienne est de rang plein.

Ces trois conséquences suffisent pour autoriser le calcul la matrice d'interaction associée à $\bar{q}$. Toutefois, il peut malheureusement arriver en pratique que l'une de ces hypothèses soit transgressée, comme c'est le cas dans l'exemple des cercles donné en section 2.2.2. L'apparition de singularités isolées peut alors induire des problèmes au niveau de la commande elle-même.

Ces préliminaires étant établis, le calcul de la matrice d'interaction se ramène au calcul de l'expression $\frac{\partial \bar{q}}{\partial P}\frac{\partial P}{\partial r}$, et le plus souvent, en pratique, à celui de $\frac{\partial \bar{q}}{\partial \bar{p}}\frac{\partial \bar{p}}{\partial r}$, chacun des termes étant une matrice de rang plein en raison des hypothèses considérées. Toutefois cette forme n'est pas toujours la plus appropriée à un calcul analytique de L^T, et l'on peut parfois utiliser une autre méthode.

En effet, les primitives du motif dans la scène peuvent généralement être spécifiées par une équation de la forme :

$$h(\bar{x}, \bar{p}) = 0 \tag{2.6}$$

où h définit la nature de la primitive et où la valeur de $\bar{p}$ correspond à une de ses configurations.

- **Remarque** : Il existe parfois une fonction h capable de représenter l'ensemble des primitives constituant le motif dans la scène. Cette fonction n'est en général pas différentiable. Aussi calcule-t-on en pratique, la matrice d'interaction associée à chaque primitive, la matrice d'interaction globale étant simplement constituée de la réunion des matrices d'interaction associées aux différentes primitives du motif.

En utilisant (2.5), l'équation (2.6) s'écrit :

$$h(z\bar{X}, \bar{p}) = 0 \tag{2.7}$$

soit

$$h'(\bar{X}, z, \bar{p}) = 0 \tag{2.8}$$

Le théorème des fonctions implicites permet alors de dire autour d'une solution $\bar{x}_0$ de (2.6), sous la condition ici triviale $\frac{\partial h'}{\partial z} \neq 0$, qu'il existe une fonction unique μ telle que :

$$z = \mu(\bar{X}, \bar{p}) \tag{2.9}$$

Considérons tour à tour les différents types de primitives que l'on peut rencontrer : points, droites, primitives planes et tridimensionnelles. Nous allons nous attacher à obtenir de manière explicite la fonction μ et la projection dans l'image de $h(\bar{x}, \bar{p}) = 0$ que nous écrirons sous la forme :

$$g(\bar{X}, \bar{q}) = 0 \tag{2.10}$$

- Cas du point : Soit m_i un point de e de coordonnées $\bar{x}_i = (x_i\, y_i\, z_i)^T$. On a :

$$h(\bar{x}, \bar{p}) = \begin{cases} h_1 = x - x_i = 0 \\ h_2 = y - y_i = 0 \\ h_3 = z - z_i = 0 \end{cases} \tag{2.11}$$

Pour ce cas trivial, μ est donnée à partir de $h_3 : z = z_i$ et g par :

$$g(\bar{X}, \bar{q}) = \begin{cases} g_1 = X - X_i = 0 \\ g_2 = Y - Y_i = 0 \end{cases} \tag{2.12}$$

où X_i et Y_i sont les coordonnées de la projection dans l'image de m_i.

- Cas de la droite et des primitives planes : h est alors de dimension 2 ($h = (h_1\ h_2)^T$). On peut supposer par exemple que h_2 représente le plan auquel appartient la primitive (dans le cas de la droite, l'écriture de h_2 n'est pas unique). μ est alors construite à partir de $h'_2(\bar{X}, z, \bar{p}) = 0$. En utilisant (2.9), $h'_1(\bar{X}, z, \bar{p}) = 0$ donne :

$$\tilde{h}(\bar{X}, \bar{p}) = 0 \ \text{ avec dim } \tilde{h} = 1 \tag{2.13}$$

qui s'écrit, après reparamétrisation $g(\bar{X}, \bar{q}) = 0$.

- Cas des primitives tridimensionnelles (sphères, cylindres, tores,..., ou même intersection de plusieurs primitives tridimensionnelles) : Pour ce cas également, il est possible d'exhiber les fonctions μ et g données par les équations (2.9) et (2.10). En effet, on s'intéresse alors au contour de la projection dans l'image de la primitive. La fonction $g(\bar{X}, \bar{q})$ est alors donnée par l'équation des limbes et la particularité des points 3D correspondant aux points contours peut s'exprimer sous la forme $h'_0(\bar{X}, z, \bar{p}) = 0$, avec dim $h'_0 = 1$, qui fournit la fonction $z = \mu(\bar{X}, \bar{p})$.

g (motif dans l'image) est la projection de h (motif de la scène), et l'on peut choisir $s = \bar{q}$ ou $s = \sigma(\bar{q})$.

Le calcul de la matrice d'interaction L^T peut à présent s'effectuer en remarquant que l'hypothèse de rigidité implique $\dot{g} = 0$, $\forall \bar{X} \in Q$. D'où, en dérivant (2.10) :

$$\frac{\partial g}{\partial \bar{q}}(\bar{X}, \bar{q})\ \dot{\bar{q}} = -\frac{\partial g}{\partial \bar{X}}(\bar{X}, \bar{q})\ \dot{\bar{X}}\ ,\ \forall \bar{X} \in Q \tag{2.14}$$

Par ailleurs, la dérivation de (2.5) nous donne les équations bien connues de l'"optic flow", que nous écrivons sous la forme :

$$\dot{\bar{X}} = L^T_{of}(\bar{X}, z)\ T \tag{2.15}$$

où T représente la vitesse de la caméra par rapport à la scène et où:

$$L^T_{of} = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \tag{2.16}$$

En utilisant (2.9) dans (2.16), il vient :

$$L^T_{of}(\bar{X}, z) = L'^T_{of}(\bar{X}, \bar{p}) \tag{2.17}$$

Finalement, (2.14), (2.15) et (2.17) donnent :

$$\frac{\partial g}{\partial \bar{q}}(\bar{X}, \bar{q})\ \dot{\bar{q}} = -\frac{\partial g}{\partial \bar{X}}(\bar{X}, \bar{q})\ L'^T_{of}(\bar{X}, \bar{p})\ T\ ,\ \forall \bar{X} \in Q \tag{2.18}$$

que l'on peut résoudre, soit, après utilisation explicite de (2.10) dans (2.18), par identification terme à terme, soit de la façon suivante :

La matrice L^T recherchée est de dimension $m \times 6$. On choisit alors m points $\bar{X}_i$ de Q, et l'on obtient :

$$\begin{cases} \dfrac{\partial g}{\partial \bar{q}}(\bar{X}_i, \bar{q}) & = & \alpha_i^T(\bar{q}) \\ -\dfrac{\partial g}{\partial \bar{X}}(\bar{X}_i, \bar{q}) L_{of}^T(\bar{X}_i, \bar{p}) & = & \beta_i^T(\bar{p}, \bar{q}) \end{cases} \tag{2.19}$$

La matrice d'interaction s'écrit alors :

$$L^T(\bar{p}, \bar{q}) = \begin{pmatrix} \alpha_1^T \\ \vdots \\ \alpha_m^T \end{pmatrix}^{-1} (\bar{q}) \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_m^T \end{pmatrix} (\bar{p}, \bar{q}) \tag{2.20}$$

- **Remarque** : Il apparaît dans cette dernière équation (2.20) que le calcul de L^T n'est possible que si la matrice $(\alpha_1 \cdots \alpha_m)$ est inversible c'est à dire si la représentation de g par les paramètres $\bar{q}$ est minimale.

2.2.2 Quelques exemples de motifs usuels

Les Points

Soit un point m_i de e de coordonnées $\bar{x}_i$; on a alors tout simplement $\bar{p} = \bar{x}_i = (x_i\, y_i\, z_i)^T$ et $\bar{q} = \bar{X}_i = (X_i\, Y_i)^T$. La représentation matricielle L_{X_i} et L_{Y_i} des deux torseurs d'interaction H_{X_i} et H_{Y_i} , exprimée dans F et au point O est donnée par (2.16) et s'écrit :

$$\begin{array}{l} L_{X_i}^T = [\ -1/z_i \quad 0 \quad X_i/z_i \quad X_iY_i \quad -(1+X_i^2) \quad Y_i\ ] \\ L_{Y_i}^T = [\ 0 \quad -1/z_i \quad Y_i/z_i \quad 1+Y_i^2 \quad -X_iY_i \quad -X_i\ ] \end{array} \tag{2.21}$$

- **Remarque** : Il est aisé de vérifier que si l'on exprime les deux torseurs d'interaction $H_{X|_{F,O}}$ et $H_{Y|_{F,O}}$ au point m au lieu du point O, ces torseurs se réduisent alors à des glisseurs (i.e. leur champ en m est nul).

On peut définir de multiples signaux capteurs à partir des points. Considérons par exemple le cas d'un segment 3D d'extrémités les points m_1 et m_2. Ce segment peut être représenté dans l'image soit par les coordonnées de ses extrémités M_1 et M_2, et on a alors $L_{X_1}, L_{Y_1}, L_{X_2}, L_{Y_2}$ donnés par (2.21), soit par les paramètres suivants : la longueur l, l'orientation α et les coordonnés X_c et Y_c du milieu du segment.

On peut facilement calculer les torseurs d'interaction associés à ce paramétrage et on obtient :

$$L_l^T = \begin{bmatrix} \lambda_1 \cos\alpha & \lambda_1 \sin\alpha \\ \lambda_2 l - \lambda_1(X_c \cos\alpha + Y_c \sin\alpha) & l[X_c \cos\alpha \sin\alpha + Y_c(1+\sin^2\alpha)] \\ -l[X_c(1+\cos^2\alpha) + Y_c \cos\alpha \sin\alpha] & 0 \end{bmatrix}$$

$$L_\alpha^T = \begin{bmatrix} -\lambda_1 \sin\alpha/l & \lambda_1 \cos\alpha/l \\ \lambda_1(X_c \sin\alpha - Y_c \cos\alpha)/l & -X_c \sin^2\alpha + Y_c \cos\alpha \sin\alpha \\ X_c \cos\alpha \sin\alpha - Y_c \cos^2\alpha & -1 \end{bmatrix}$$

$$L_{X_c}^T = \begin{bmatrix} -\lambda_2 & 0 \\ \lambda_2 X_c - \lambda_1 l \cos\alpha/4 & X_c Y_c + l^2 \cos\alpha \sin\alpha/4 \\ -(1 + X_c^2 + l^2 \cos^2\alpha/4) & Y_c \end{bmatrix}$$

$$L_{Y_c}^T = \begin{bmatrix} 0 & -\lambda_2 \\ \lambda_2 Y_c - \lambda_1 l \sin\alpha/4 & 1 + Y_c^2 + l^2 \sin^2\alpha/4 \\ -X_c Y_c - l^2 \cos\alpha \sin\alpha/4 & -X_c \end{bmatrix} \quad (2.22)$$

avec $\lambda_1 = (z_1 - z_2)/z_1 z_2$ et $\lambda_2 = (z_1 + z_2)/2z_1 z_2$.

De la même manière, il est possible de calculer le torseur d'interaction de tous capteurs basés sur les points tels que surface, centre de gravité d'un polygone, etc...

Les droites

Une droite de l'espace 3D peut être représentée comme l'intersection de deux plans :

$$h(\bar{x}, \bar{p}) = \begin{cases} a_1 x + b_1 y + c_1 z + d_1 = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \end{cases} \quad (2.23)$$

Excepté le cas dégénéré où la droite passe par le centre de projection ($d_1 = d_2 = 0$), cette droite se projette sur le plan image sous la forme d'une droite d'équation :

$$AX + BY + C = 0 \text{ avec } \begin{cases} A = a_1 d_2 - a_2 d_1 \\ B = b_1 d_2 - b_2 d_1 \\ C = c_1 d_2 - c_2 d_1 \end{cases} \quad (2.24)$$

La représentation (A, B, C) n'est pas minimale, c'est pourquoi il nous faut choisir une autre représentation des droites 2D. La représentation la plus couramment utilisée, à savoir $\bar{q} = (a, b)$ ne paraît pas, elle non plus, adéquate car elle nécessite l'emploi de 2 cartes ($Y = aX + b$, $X = aY + b$) et la condition de différentiabilité n'est pas satisfaite au passage d'une carte à l'autre. Nous choisissons alors la représentation $\bar{q} = (\rho, \theta)$ et l'équation d'une droite $\mathcal{D}$ est donnée par :

$$g(\bar{X}, \bar{q}) = X \cos\theta + Y \sin\theta - \rho = 0 \quad (2.25)$$

- **Remarque :** On a bien entendu $\theta = \arctan(B/A)$ et $\rho = -C/\sqrt{A^2 + B^2}$. De plus, l'ambiguïté de cette représentation (la même droite peut être paramétrée indifféremment par $(\rho, \theta + 2k\pi)$ et $(-\rho, \theta + (2k+1)\pi)$) est levée en partie en définissant un sens aux droites, ce qui permet de fixer le signe de ρ. Nous verrons

que l'ambiguïté dûe aux multiples choix possibles de θ, d'une part, n'a aucune influence sur la valeur des torseurs d'interaction associés à ρ et θ et, d'autre part, peut facilement être levée lors de l'exécution de toute tâche utilisant des droites comme signaux capteurs (voir section 4.1.2).

Calculons à présent les torseurs d'interaction correspondant à cette représentation. L'équation (2.14) s'écrit dans le cas présent sous la forme :

$$\dot{\rho} + (X \sin\theta - Y \cos\theta)\dot{\theta} = \cos\theta \dot{X} + \sin\theta \dot{Y} \ , \ \forall (X,Y) \in \mathcal{D} \tag{2.26}$$

Par ailleurs, on obtient la fonction μ de (2.9) à partir de h_1 ou h_2 :

$$1/z = \begin{cases} -(a_1 X + b_1 Y + c_1)/d_1 & \text{si } d_1 \neq 0 \\ -(a_2 X + b_2 Y + c_2)/d_2 & \text{si } d_2 \neq 0 \end{cases} \tag{2.27}$$

A partir de l'équation (2.25), on écrit X en fonction de Y si $\cos\theta \neq 0$ (ou Y en fonction de X sinon) et l'équation (2.26) peut alors s'écrire, en utilisant (2.16) et (2.27) :

$$(\dot{\theta}/\cos\theta)\ Y + (\dot{\rho} + \tan\theta\ \dot{\theta}) = Y\ K_1(\bar{p},\bar{q})T + K_2(\bar{p},\bar{q})T \ , \ \forall Y \in \mathbb{R} \tag{2.28}$$

On en déduit aussitôt :

$$\begin{cases} \dot{\theta} &= -K_1(\bar{p},\bar{q})\ \cos\theta\ T \\ \dot{\rho} &= (K_2(\bar{p},\bar{q}) + K_1(\bar{p},\bar{q}))\ \sin\theta\ T \end{cases} \tag{2.29}$$

d'où :

$$\begin{array}{ll} L_\theta^T = [& \lambda_\theta \cos\theta \quad \lambda_\theta \sin\theta \quad -\lambda_\theta \rho \quad -\rho\cos\theta \quad -\rho\sin\theta \quad -1 \ ] \\ L_\rho^T = [& \lambda_\rho \cos\theta \quad \lambda_\rho \sin\theta \quad -\lambda_\rho \rho \quad (1+\rho^2)\sin\theta \quad -(1+\rho^2)\cos\theta \quad 0 \ ] \end{array} \tag{2.30}$$

$$\begin{array}{lll} \text{avec} & \lambda_\theta = & (a_2 b_1 - a_1 b_2)/\sqrt{A^2 + B^2} \\ \text{et} & \lambda_\rho = & [(c_2 a_1 - c_1 a_2)\cos\theta + (c_2 b_1 - c_1 b_2)\sin\theta]/\sqrt{A^2 + B^2} \end{array}$$

Les cercles

Considérons maintenant le cas d'un cercle dans l'espace 3D. Il peut être représenté comme l'intersection d'une sphère et d'un plan et on a alors :

$$h(\bar{x},\bar{p}) = \begin{cases} (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \\ (z - z_0) - \alpha_0 (x - x_0) - \beta_0 (y - y_0) = 0 \end{cases} \tag{2.31}$$

Ce cercle se projette dans le plan image sous la forme d'une ellipse (sauf dans les cas dégénérés où le cercle se projette sous la forme d'un segment) qui peut être représentée par :

$$g(\bar{X},\bar{q}) : X^2 + A_1 Y^2 + 2A_2 XY + 2A_3 X + 2A_4 Y + A_5 = 0 \tag{2.32}$$

où

$$\begin{aligned}
A_1 &= [b^2(x_0^2+y_0^2+z_0^2-r^2)+1-2by_0]/A_0 \\
A_2 &= [ab(x_0^2+y_0^2+z_0^2-r^2)-bx_0-ay_0]/A_0 \\
A_3 &= [ac(x_0^2+y_0^2+z_0^2-r^2)-cx_0-az_0]/A_0 \\
A_4 &= [bc(x_0^2+y_0^2+z_0^2-r^2)-cy_0-bz_0]/A_0 \\
A_5 &= [c^2(x_0^2+y_0^2+z_0^2-r^2)+1-2cz_0]/A_0
\end{aligned}$$

avec $A_0 = a^2(x_0^2+y_0^2+z_0^2-r^2)+1-2ax_0 \quad (\neq 0)$
et $a = -\alpha_0 c\ ,\ b = -\beta_0 c\ ,\ c = -1/(\alpha_0 x_0 + \beta_0 y_0 - z_0)$

Remarques :

- La représentation $\bar{q} = (A_1, \cdots, A_5)$ a été choisie pour représenter les ellipses car elle a l'avantage d'être non ambiguë contrairement à la représentation plus couramment utilisée $\bar{q} = (e, a, b, X_c, Y_c)$, où les paramètres sont respectivement l'orientation, le grand axe, le petit axe et les coordonnées du centre de l'ellipse.

- Les paramètres représentant une ellipse $\mathcal{E}$ peuvent s'exprimer à partir des moments d'inertie $\sum_{X\in\mathcal{E}}\sum_{Y\in\mathcal{E}} X^i Y^j, \forall (i,j) \in \mathbb{N}^2, i+j \leq 2$ qui sont facilement mesurables dans une image numérisée.

La fonction $\mu(\bar{X}, \bar{p})$, nécessaire pour le calcul des torseurs d'interaction, s'obtient facilement à partir de h_2 :

$$1/z = aX + bY + c \tag{2.33}$$

et on obtient, après de fastidieux calculs qui ne seront pas détaillés ici, les résultats suivants :

$$\begin{aligned}
L_{A_1}^T &= \begin{bmatrix} 2bA_2 - 2aA_1 & 2A_1(b-aA_2) & 2bA_4 - 2aA_1A_3 \\ 2A_4 & 2A_1A_3 & -2A_2(A_1+1) \end{bmatrix} \\
L_{A_2}^T &= \begin{bmatrix} b-aA_2 & bA_2 - a(2A_2^2 - A_1) & a(A_4 - 2A_2A_3) + bA_3 \\ A_3 & 2A_2A_3 - A_4 & A_1 - 2A_2^2 - 1 \end{bmatrix} \\
L_{A_3}^T &= \begin{bmatrix} c-aA_3 & a(A_4 - 2A_2A_3) + cA_2 & cA_3 - a(2A_3^2 - A_5) \\ -A_2 & 1 + 2A_3^2 - A_5 & A_4 - 2A_2A_3 \end{bmatrix} \\
L_{A_4}^T &= \begin{bmatrix} A_3b + A_2c - 2aA_4 & A_4b + A_1c - 2aA_2A_4 & bA_5 + cA_4 - 2aA_3A_4 \\ A_5 - A_1 & 2A_3A_4 + A_2 & -2A_2A_4 - A_3 \end{bmatrix} \\
L_{A_5}^T &= \begin{bmatrix} 2cA_3 - 2aA_5 & 2cA_4 - 2aA_2A_5 & 2cA_5 - 2aA_3A_5 \\ -2A_4 & 2A_3A_5 + 2A_3 & -2A_2A_5 \end{bmatrix}
\end{aligned} \tag{2.34}$$

- **Remarque** : Il est à noter que la matrice d'interaction L^T associée à l'ellipse est toujours de rang 5, excepté dans le cas où la projection du cercle 3D est un cercle centré dans l'image. On a alors :

$$A_1 = 1, A_2 = A_3 = A_4 = 0, A_5 = -r^2/z_0^2 \Leftrightarrow a = b = x_0 = y_0 = 0 \qquad (2.35)$$

Les vecteurs L_{A_1} et L_{A_2} sont alors nuls et la matrice L^T est de rang 3. Il s'agit là d'une singularité (au sens perte de rang) isolée puisque la matrice L^T est de rang 5, même dans les autres cas où le cercle 3D se projette dans l'image sous la forme d'un cercle :

$$A_1 = 1, A_2 = 0 \Leftrightarrow \begin{cases} a = b = 0 \\ \text{ou} \\ a = 2x_0/(x_0^2 + y_0^2 + z_0^2 - r^2)\,, \\ b = 2y_0/(x_0^2 + y_0^2 + z_0^2 - r^2) \end{cases} \qquad (2.36)$$

Les sphères

Etudions enfin le cas de la sphère, exemple de primitive tridimensionnelle. On a :

$$h(\bar{x}, \bar{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \qquad (2.37)$$

Calculons tout d'abord les fonctions $g(\bar{X}, \bar{q})$ et $\mu(\bar{X}, \bar{p})$. En utilisant l'équation (2.5) dans (2.37), on obtient :

$$(X - \frac{x_0}{z})^2 + (Y - \frac{y_0}{z})^2 + (1 - \frac{z_0}{z})^2 - \frac{R^2}{z^2} = 0 \qquad (2.38)$$

soit un polynôme du deuxième degré en $1/z$:

$$\frac{1}{z^2}(x_0^2 + y_0^2 + z_0^2 - R^2) - \frac{2}{z}(x_0X + y_0Y + z_0) + X^2 + Y^2 + 1 = 0 \qquad (2.39)$$

Les points appartenant au contour de l'image d'une sphère sont tels que l'intersection entre leur ligne de vue et la sphère soit unique. Dans le cas présent, cette propriété est équivalente au fait que le discriminant Δ de (2.39) soit nul. On a donc :

$$(x_0X + y_0Y + z_0)^2 - (x_0^2 + y_0^2 + z_0^2 - R^2)(X^2 + Y^2 + 1) = 0 \qquad (2.40)$$

qui s'écrit après reparamétrisation :

$$g(\bar{X}, \bar{q}) : X^2 + A_1Y^2 + 2A_2XY + 2A_3X + 2A_4Y + A_5 = 0 \qquad (2.41)$$

où

$$\begin{aligned} A_1 &= (R^2 - x_0^2 - z_0^2)/(R^2 - y_0^2 - z_0^2) \\ A_2 &= x_0y_0/(R^2 - y_0^2 - z_0^2) \\ A_3 &= x_0z_0/(R^2 - y_0^2 - z_0^2) \\ A_4 &= y_0z_0/(R^2 - y_0^2 - z_0^2) \\ A_5 &= (R^2 - x_0^2 - y_0^2)/(R^2 - y_0^2 - z_0^2) \end{aligned}$$

L'image d'une sphère est donc généralement une ellipse (un cercle dans le cas où $x_0 = y_0 = 0$).

Par ailleurs, la fonction μ est facilement obtenue à partir de la racine double de (2.39) correspondant à $\Delta = 0$ et on a :

$$1/z = aX + bY + c \text{ avec } \begin{cases} a = x_0/(x_0^2 + y_0^2 + z_0^2 - R^2) \\ b = y_0/(x_0^2 + y_0^2 + z_0^2 - R^2) \\ c = z_0/(x_0^2 + y_0^2 + z_0^2 - R^2) \end{cases} \tag{2.42}$$

Le calcul des torseurs d'interaction associés à une ellipse représentée par $\bar{q} = (A_1, ..., A_5)^T$ a déjà été effectué dans le cas du cercle, aussi les résultats sont-ils donnés par les équations (2.34) avec pour a,b et c les valeurs particulières précédentes. La matrice L^T associée est toujours de rang 3, quelque soit la configuration de la sphère. Trois paramètres sont donc suffisants pour caractériser l'image d'une sphère (ce résultat paraît logique puisque quatre paramètres sont suffisants pour représenter une sphère). On peut choisir les paramètres A_3, A_4 et A_5 car la matrice d'interaction associée est toujours de rang plein 3, alors que, si l'on utilise A_1 ou A_2, on a encore L_{A_1} et L_{A_2} nuls dans le cas où l'image de la sphère est un cercle.

Chapitre 3

La commande

3.1 Cadre général

Nous plaçons cette étude dans le contexte plus vaste de la commande référencée capteurs. Une approche générale du problème dûe à Samson [SAM 87], [SAM 90b] a été développée par ailleurs et nous nous contenterons ici d'en rappeler les caractéristiques principales.

3.1.1 Le concept de fonction de tâche

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

$$\Gamma = M(q)\ddot{q} + N(q, \dot{q}, t), \dim(q) = \dim M = n \tag{3.1}$$

où :

- Γ est le vecteur des forces extérieures appliquées,
- M la matrice d'énergie cinétique
- N rassemble les contributions des forces de gravité, centrifuges, de Coriolis, et de frottement.

L'équation (3.1) est l'équation **d'état** du système, et le vecteur d'état naturellement associé est $(q, \dot{q})$ (on suppose par ailleurs qu'un actionneur est associé à chaque degré de liberté). La tâche à effectuer peut alors être spécifiée par l'utilisateur comme une **fonction de sortie** associée à (3.1). Le problème sera alors bien posé si le passage entre l'"espace de commande" et l'"espace de sortie" est régulier en un certain sens.

Plus précisément, on peut montrer [SAM 90b] que l'objectif de l'utilisateur peut généralement être exprimé comme la régulation (à zéro) d'une certaine fonction C^2 de dimension n, $e(q,t)$ appelée "fonction de tâche", sur un certain horizon $[0, T]$. Quelques exemples immédiats de telles fonctions sont :

$$e(q,t) = q - q_d(t) \tag{3.2}$$

où $q_d(t)$ est une trajectoire articulée désirée,

$$e(q,t) = x(q) - x_d(t) \tag{3.3}$$

où $x_d(t)$ est par exemple la position désirée du poignet d'un robot à 3 degrés de liberté, exprimée dans un espace cartésien.

De nombreuses autres situations sont décrites en [SAM 90b]. Dans le cas particulier où l'on utilise des capteurs, tels que définis au paragraphe 2.1.2, il apparait d'ores et déjà que le vecteur $s(q,t)$ participera à la constitution de la fonction de tâche, d'une façon qui sera précisée plus loin.

Comme cela est précisé en [SAM 87] et [SAM 90b], le problème de régulation de e est bien posé si celle-ci possède certaines propriétés. L'une d'elles est l'existence et l'unicité d'une **trajectoire idéale** C^2, notée $q_r(t)$, telle que $e(q_r(t),t) = 0, t \in [0,T]$ et $q_r(0) = q_0$ où q_0 est une condition initiale donnée. Une autre, extrêmement importante, est la **régularité** du Jacobien de la tâche, $\frac{\partial e}{\partial q}(q,t)$, autour de $q_r(t)$. Lorsque toutes les conditions requises sont satisfaites, la fonction de tâche est dite "admissible", ce qui permet **alors** la synthèse de lois de commande efficaces, dont la robustesse sera liée à la plus ou moins grande "admissibilité" de la tâche.

3.1.2 Commande et stabilité

Nous donnons ici seulement une idée **intuitive** de l'approche utilisée et du type de résultats obtenus, et renvoyons à [SAM 87], [SAM 90b] pour une description complète. Considérons d'abord simplement le problème du découplage et de la linéarisation de (3.1) dans le cas idéal. Une commande les réalisant est de la forme :

$$\Gamma = M(q)u + N \tag{3.4}$$

où u est par exemple un retour proportionnel dérivé sur l'erreur $q - q_d$.

Appliquons la même méthode à présent dans l'espace de sortie. Nous pouvons écrire :

$$\dot{e}(q,t) = \frac{\partial e}{\partial q}(q,t)\dot{q} + \frac{\partial e}{\partial t}(q,t) \tag{3.5}$$

et

$$\ddot{e}(q,t) = \frac{\partial e}{\partial q}(q,t)\ddot{q} + f(q,\dot{q},t) \tag{3.6}$$

avec :

$$f(q,\dot{q},t) = \begin{bmatrix} \vdots \\ \dot{q}^T W_i(q,t)\dot{q} \\ \vdots \end{bmatrix} + 2\frac{\partial^2 e}{\partial q \partial t}(q,t)\dot{q} + \frac{\partial^2 e}{\partial^2 t}(q,t) \tag{3.7}$$

où $W_i(q,t)(i = 1,\cdots,n)$ est la dérivée partielle de la i-ème colonne de $\left(\frac{\partial e}{\partial q}\right)^T(q,t)$ par rapport à q.

La fonction e étant supposée admissible, on peut combiner les équations (3.1) et (3.6), ce qui donne :

$$\Gamma = M\left(\frac{\partial e}{\partial q}\right)^{-1}\ddot{e} + N - M\left(\frac{\partial e}{\partial q}\right)^{-1} f \tag{3.8}$$

et par analogie avec (3.4) une forme de commande linéarisante dans l'espace de la tâche serait :

$$\Gamma = M\left(\frac{\partial e}{\partial q}\right)^{-1} u' + N - M\left(\frac{\partial e}{\partial q}\right)^{-1} f \tag{3.9}$$

avec un retour proportionnel-dérivé de forme générale :

$$u' = -kG\left(\mu De + \dot{e}\right) \tag{3.10}$$

G et D étant des matrices positives, k et μ des scalaires positifs, tous devant être fixés par l'utilisateur.

Naturellement, la commande idéale (3.9) (3.10) suppose une connaissance parfaite de tous les termes la constituant ce qui n'est ni possible, ni même souvent souhaité. On considèrera donc finalement, de façon plus réaliste le schéma de commande suivant, généralisation du précédent :

$$\Gamma = -k\hat{M}\left(\frac{\widehat{\partial e}}{\partial q}\right)^{-1} G\left(\mu De + \frac{\widehat{\partial e}}{\partial q}\dot{q} + \frac{\widehat{\partial e}}{\partial t}\right) + \hat{N} - \hat{M}\left(\frac{\widehat{\partial e}}{\partial q}\right)^{-1} \hat{f} \tag{3.11}$$

dans lesquels les "chapeaux" indiquent que l'on choisit des modèles (approximations, estimations...) des grandeurs considérées.

Dans cette expression générale, nous autorisons tous les termes, à l'exception de μ, D et G, à être des fonctions de q et de t, voire même de $\dot{q}$ pour $k, \hat{f}$ et $\hat{N}$. La commande (3.11) englobe de nombreux schémas existants : commande dynamique, commande en vitesse, commande adaptative indirecte, etc.

Une analyse de la stabilité du système (3.1) rebouclé par la commande (3.11) a été réalisée par Samson [SAM 90b] dans un cadre non-linéaire, et a permis d'exhiber des conditions suffisantes de stabilité (bornitude de $||e(t)||$ et de plus, comportement asymptotique), qui sont principalement de deux classes :

- des conditions portant sur les **gains** de l'asservissement. Il s'agit là de réglages laissant une plus ou moins grande plage de manœuvre à l'utilisateur

- des conditions portant sur les **modèles** utilisés. Parmi celles-ci, certaines concernent les paramètres dynamiques du robot et ne sont pas trop contraignantes (en particulier en raison de la symétrie-définie-positivité de $M(q)$). Une autre, beaucoup plus critique et liée à la tâche, est la suivante :

$$\frac{\partial e}{\partial q}\left(\frac{\widehat{\partial e}}{\partial q}\right)^{-1} > 0 \tag{3.12}$$

(au sens où une matrice A $(n \times n)$ est positive si $x^T Ax > 0$, $\forall x \neq 0 \in \mathbb{R}^n$). Nous reviendrons plus loin sur l'utilisation de cette condition essentielle qui est une façon de caractériser la **robustesse** de la tâche aux incertitudes et aux approximations effectuées.

On peut toutefois d'ores et déjà remarquer que, lorsque l'on s'intéresse à l'organe terminal du robot, on peut écrire $\frac{\partial e}{\partial q} = \frac{\partial e}{\partial r}\frac{\partial r}{\partial q}$, où $\frac{\partial r}{\partial q}$ est le jacobien classique du robot.

Si celui-ci est parfaitement connu et inversible, le choix $\frac{\widehat{\partial e}}{\partial q} = \frac{\widehat{\partial e}}{\partial \bar{r}} \frac{\partial \bar{r}}{\partial q}$ permet de réduire la condition (3.12) à :

$$\frac{\partial e}{\partial \bar{r}} \left(\frac{\widehat{\partial e}}{\partial \bar{r}} \right)^{-1} > 0 \tag{3.13}$$

3.1.3 Tâches hybrides

Introduction

Très souvent, la régulation de signaux capteurs n'est pas l'unique objectif visé, et cette tâche doit être combinée avec une autre, par exemple le suivi d'une trajectoire.

En général, la spécification d'un tel problème conduit en une première étape à la définition d'un vecteur de tâche référencée capteur, $e_1(q,t)$, constitué de $m \leq n$ composantes indépendantes, et dont la régulation constitue la partie de la tâche globale nécessitant l'usage de capteurs extéroceptifs. La façon d'obtenir un tel vecteur dans le cas d'un capteur visuel sera indiquée plus loin. Le second objectif, par exemple un certain mouvement du capteur lui-même, pourrait a priori être représenté par un deuxième vecteur $e_2(q,t)$. Toutefois, il faudrait alors être capable de s'assurer que e_1 et e_2 peuvent être regroupés en un seul vecteur $e(q,t)$ **admissible**. Les deux tâches concernées doivent alors être compatibles et indépendantes, ce qui signifie grossièrement, en termes de liaisons virtuelles que le deuxième objectif peut être atteint grâce à l'ensemble des mouvements réalisables laissés libres par la liaison virtuelle associée à la tâche référencée capteurs.

En fait, on peut montrer qu'une façon plus riche de poser le problème consiste à l'immerger dans celui des "tâches redondantes". Dans cette approche, la tâche e_1 est considérée comme prioritaire, et e_2 apparaît alors comme devant représenter la minimisation d'un coût secondaire, sous contrainte de réalisation parfaite de la régulation de e_1. Rappelons à présent les principaux résultats obtenus en la matière [SAM 88] [SAM 90b].

Le formalisme de la redondance

Soit E_c un certain espace de configuration de travail, de dimension n, dont un élément est appelé x. Dans notre cas cet espace sera généralement SE_3, et x sera $\bar{r}$. On suppose que le passage de E_c à l'espace articulaire est régulier, i.e. $\frac{\partial x}{\partial q}$ est inversible. Soit e_1 la tâche principale de dimension m et de Jacobien J_1^x $(= \frac{\partial e_1}{\partial x})$ dans E_c, et soit h_s le coût secondaire à minimiser, de gradient $g_s^x = \frac{\partial h_s}{\partial x}$.

La minimisation de h_s sous la contrainte $e_1 = 0$ nécessite la détermination du sous-espace des mouvements laissés libres par cette contrainte. Cela est équivalent à connaître le noyau de J_1^x, Ker J_1^x (ou l'image de $J_1^{x^T}$, $R(J_1^{x^T})$) le long de la trajectoire idéale, c'est-à-dire en d'autres termes, à connaître n'importe quelle matrice $m \times n$ de rang plein, W, telle que :

$$R(W^T) = R\left(J_1^{x^T}\right) \tag{3.14}$$

le long de la trajectoire idéale du robot.

Une fois cette matrice déterminée, alors, on peut montrer assez facilement [SAM 88] [SAM 90b] qu'une fonction de tâche réalisant l'objectif de minimisation de h_s sous la

contrainte $e_1 = 0$ est :

$$e = W^+ e_1 + \alpha \left(I_n - W^+ W\right) g_s^x \tag{3.15}$$

où :

- α est un scalaire positif,
- W^+ la pseudo-inverse de W
- $(I - W^+W)$ un opérateur de projection orthogonale sur le noyau de W, donc sur celui de J_1^x.

L'on imagine aisément que le calcul du Jacobien $\frac{\partial e}{\partial q}$ associé à (3.15), éventuellement requis dans la commande, peut être complexe. La condition de positivité (3.12) s'avère alors de quelque utilité. On peut en effet montrer que si la matrice W, en plus de (3.14), satisfait la propriété :

$$J_1^x W^T > 0 \tag{3.16}$$

le long de la trajectoire idéale, alors, dans des "circonstances normales" (voir [SAM 90b] pour de plus amples détails), le Jacobien de e dans E_c est tel que :

$$\frac{\partial e}{\partial x}\left(I_n + \gamma\left(I_n - W^+W\right)\right) > 0 \tag{3.17}$$

le long de la trajectoire idéale, $\forall \gamma \geq \gamma_m(\alpha) \geq 0$.

Sachant que $\frac{\partial e}{\partial q} = \frac{\partial e}{\partial x}\frac{\partial x}{\partial q}$, la condition (3.12) est alors satisfaite par le choix :

$$\left(\widehat{\frac{\partial e}{\partial q}}\right)^{-1} = \left(\frac{\partial x}{\partial q}\right)^{-1}\left(I_n + \gamma\left(I_n - W^+W\right)\right) \tag{3.18}$$

De plus, si α est "suffisamment petit", alors $\gamma_m = 0$, $\frac{\partial e}{\partial x}$ est positif, et l'on peut choisir :

$$\widehat{\frac{\partial e}{\partial q}} = \frac{\partial x}{\partial q} \tag{3.19}$$

Remarques :

- Lorsque e_1 est construit à partir de signaux capteurs et que h_s exprime un suivi de trajectoire dans SE_3, la tâche représentée par (3.15) porte alors le nom de "tâche hybride" ; $\frac{\partial x}{\partial q}$ est alors souvent le jacobien de base du robot, et la propriété ci-dessus conduisant au choix (3.19), explique pourquoi certains schémas plus classiques dits de "commande hybride" peuvent fonctionner, même sans utilisation d'une expression explicite de la variation des signaux capteurs dans l'équation de commande.

- Une situation intéressante est celle dans laquelle le sous-espace Ker $J_1^x(q,t)$ est invariant, $\forall t$, sur l'ensemble des solutions de $e_1(q,t) = 0$. Dans le cas de capteurs, ceci se produit lorsque le sous-espace S (évalué dans un repère lié au système de capteurs ou à la caméra) est invariant dans l'ensemble des positions du système de mesure pour lesquelles la contrainte de liaison virtuelle (ici $e_1 = 0$) est satisfaite. Dans ces conditions, la connaissance de S est suffisante pour déterminer $W(t)$, voire même, dans certains cas, W constante, satisfaisant (3.14).

Le cas spécifique des signaux capteurs

[SAM 90a]

Appliquons à présent cette approche à l'utilisation de signaux capteurs tels que définis au paragraphe 2.1. Rappelons que nous considérons un vecteur s de dimension p ; on travaillera dans SE_3 (i.e. "$x = \bar{r}$" dans les développements précédents), et l'on rappelle également que le Jacobien de s dans SE_3 correspond à la matrice d'interaction L^T. L est de dimension $6 \times p$ et de rang m, $N = 6 - m$ étant la classe de la liaison virtuelle associée au problème (cf. paragraphe 2.1.3).

On s'intéresse à la régulation du vecteur s autour d'une valeur ou d'une trajectoire désirées $s^*(t)$. Soit $D(t)$ une matrice, dite de "combinaison", de dimension $m \times p$, et telle que la matrice DL^T soit de rang plein m le long de la trajectoire idéale du robot. La tâche principale prend alors la forme :

$$e_1 = D(t)\ (s(\bar{r}, t) - s^*(t)) \tag{3.20}$$

La matrice D permet, entre autres, de prendre en compte l'existence d'un nombre de capteurs p supérieur à la dimension m des contraintes qu'ils introduisent réellement.

Le Jacobien de e_1 dans SE_3 s'écrit alors :

$$J_1^{\bar{r}} = \frac{\partial e_1}{\partial \bar{r}} = DL^T \tag{3.21}$$

et l'on peut aisément montrer que $R(J_1^{\bar{r}^T}) = R(L)$. En utilisant (3.15), la tâche à réguler s'écrit finalement :

$$e = W^+ D\ (s(\bar{r}, t) - s^*(t)) + \alpha \left(I_6 - W^+ W\right) g_s^{\bar{r}} \tag{3.22}$$

Idéalement, W doit satisfaire la propriété (3.14), qui devient alors $R\left(W^T\right) = R(L)$; cela est aussi équivalent à affirmer que les lignes de W sont formées de vecteurs de base de S.

Quant à la propriété (3.16), elle s'écrit :

$$DL^T W^T > 0 \tag{3.23}$$

qui lie les choix de D et W.

Finalement (3.14) et (3.23) sont par exemple satisfaites par les options : $D = WL$ ou $D = WL^{T+}$.

- **Remarque** : Le choix du coût secondaire à minimiser est discuté en détail en [SAM 90b].

3.2 Application à l'utilisation d'un capteur visuel

Remarquons que, comme évoqué précédemment, les modèles de torseurs d'interaction (donc L) peuvent être mal connus. Le vecteur de tâche (3.22) sera donc par exemple calculé en choisissant $D = W\hat{L}$ ou $D = W\hat{L}^{T+}$, $\hat{L}$ étant un modèle de L, et la satisfaction

des propriétés (3.14) et (3.16) nécessitera respectivement de garantir $R\left(\hat{L}\right) = R(L)$ et $DL^T W^T > 0$.

En effet, on a vu en section 2.2 que, si l'on utilise comme signaux-capteurs des informations fournies par une caméra embarquée, les matrices d'interaction L^T associées à ces informations sont de la forme $L^T(\bar{p}, \bar{q})$ où $\bar{q} = \bar{q}(\bar{r}, t)$ est mesurable dans l'image et où $\bar{p} = \bar{p}(\bar{r}, t)$ représente l'information tridimensionnelle des primitives considérées. Cette information tridimensionnelle étant a priori inconnue, un choix d'un modèle $\hat{L}$ de L est donc nécessaire. Plusieurs possibilités nous sont offertes :

- $\hat{L} = L(\hat{\bar{p}}, \bar{q})$ si l'on dispose par ailleurs d'un algorithme d'estimation de $\bar{p}$.
- $\hat{L} = L(\bar{p}^*, \bar{q})$ où $\bar{p}^*$ représente la valeur de $\bar{p}$ à la position $s = s^*$ réalisant $e_1 = 0$. Des hypothèses sur la forme et sur les dimensions des primitives considérées sont alors nécessaires. Nous y reviendrons plus loin.
- $\hat{L} = L(\hat{\bar{p}}^*, \bar{q})$ où $\hat{\bar{p}}^*$ est une estimation de $\bar{p}^*$ si l'on ne dispose a priori d'aucune information tridimensionnelle.

Les choix précédents nécessitent un calcul de la matrice D à chaque pas de la boucle de commande. Cela n'est pas forcément souhaitable, notamment dans le cas où D est choisi égal à $W\hat{L}^{T^+}$, en raison du temps de calcul des pseudo-inverses prohibitif. Mais surtout, il faut parfois tenir compte d'un éventuel passage dans une singularité isolée (c'est le cas, par exemple, lorsque l'on veut, à partir d'un cercle 3D, obtenir une certaine ellipse dans l'image et lorsque la loi de commande positionne, à un instant t, la caméra de telle sorte que le cercle 3D se projette sous la forme d'un cercle centré dans l'image (voir section 2.2.2)). Un tel passage à une singularité entraîne une perte de rang de L^T, une modification de W, d'où une perte de dimension de la tâche principale e_1. Le traitement de ces singularités peut s'avérer fort complexe et coûteux aussi préfère-t-on avoir un modèle $\hat{L}$ constant calculé lors de la définition de la tâche. On peut choisir :

- $\hat{L} = L(\bar{p}^*, \bar{q}^*)$, noté également $L_{|s=s^*}$, représentant la valeur de la matrice d'interaction pour la position correspondant au motif choisi $s = s^*$.

La condition de positivité (3.23) n'est alors assurée, quelque soit le choix de D, que dans un voisinage de la position souhaitée $s = s^*$. Heureusement, cette condition n'est que suffisante et nous verrons dans les exemples proposés en section 4 que la convergence de la loi de commande est obtenue, même à partir de positions initiales très éloignées de la position souhaitée.

Ce choix de $\hat{L}$ nécessite la connaissance de $\bar{p}^*$, ce qui revient à faire des hypothèses sur la forme et sur les dimensions de la scène 3D. Ces hypothèses sont souvent effectuées lors de la définition même de la tâche, et ne paraissent alors pas trop contraignantes : en effet, si la tâche consiste par exemple à un positionnement en face d'une porte, on doit supposer qu'il y a une porte dans la scène et qu'on peut détecter des capteurs représentant cette porte (par exemple, ses 4 coins). De plus, si on veut positionner la caméra à une distance fixée de la porte, on doit connaître ses dimensions afin de calculer le motif exact à atteindre dans l'image.

Cependant, dans certains cas, l'hypothèse de forme est la seule condition nécessaire pour calculer le motif (c'est le cas si l'on ne fixe pas la distance à atteindre entre la

caméra et l'objet). Alors, si $L^T(\bar{p}^*, \bar{q}^*)$, où la valeur de $\bar{p}^*$ est inconnue, peut s'écrire sous la forme :

$$L^T(\bar{p}^*, \bar{q}^*) = B^T(\bar{q}^*)\, C(\bar{p}^*, \bar{q}^*) \tag{3.24}$$

où C est une matrice régulière $n \times n$ positive et où B est de même rang que L, on peut choisir :

- $\hat{L} = B(\bar{q}^*)$ si la condition de positivité est satisfaite autour de s^*, c'est notamment le cas si C est diagonale ou si B est de rang plein.

Enfin, si les conditions précédentes ne sont pas remplies ou si l'on n'a aucune connaissance sur la scène (ce qui peut être le cas de tâches consistant à suivre des objets inconnus, le motif dans l'image étant alors construit par une vue initiale de l'objet), on peut choisir :

- $\hat{L} = L(\hat{\bar{p}}^*, \bar{q}^*)$ où $\hat{\bar{p}}^*$ est une estimation de $\bar{p}^*$ pouvant être très grossière, mais permettant cependant la convergence de la loi de commande dans un voisinage satisfaisant autour de la position $s = s^*$. Il est alors difficile d'assurer la condition de positivité, même pour $s = s^*$ puisque la valeur de $L_{|s=s^*}$ est alors inconnue.

Dans les différents exemples de la section suivante, les hypothèses de forme et de dimension ont été faites. On a donc $\hat{L} = L_{|s=s^*}$. De plus, la matrice D a toujours été choisie égale à $W\hat{L}^{T^+}$, où W est telle que $R\left(W^T\right) = R(\hat{L})$, en raison des meilleures propriétés de découplage des lois de commande obtenues par rapport au choix $D = W\hat{L}$.

Chapitre 4

Quelques résultats expérimentaux

Dans tous les cas, la caméra est embarquée sur un porteur mobile à 6 degrés de liberté et l'objectif consiste à positionner cette caméra d'une façon donnée relativement à une cible.

Avant de présenter ces résultats, nous allons spécifier la commande simplifiée que nous utilisons et montrer que la condition (3.12) y reste essentielle.

4.1 Une commande cinématique simplifiée

Nous allons chercher à réaliser approximativement un comportement exponentiel découplé pour l'erreur de tâche, à savoir :

$$\dot{e} = -\lambda e \tag{4.1}$$

L'équation (3.5) peut s'écrire :

$$\dot{e} = \frac{\partial e}{\partial \bar{r}} T + \frac{\partial e}{\partial t} \tag{4.2}$$

où le torseur cinématique T est relié à $\dot{q}$ par l'équation :

$$\dot{q} = \left(\frac{\partial \bar{r}}{\partial q}\right)^{-1} T \tag{4.3}$$

$\left(\frac{\partial \bar{r}}{\partial q}\right)^{-1}$ étant le jacobien inverse du robot. Nous allons à présent supposer, d'une part, que ce dernier est parfaitement connu, et, d'autre part, que la consigne accessible à l'utilisateur n'est autre que la vitesse articulaire désirée $\dot{q}_c$, comme c'est le cas pour beaucoup de robots industriels. Alors, si l'on est capable d'assurer $\dot{q}_c \simeq \dot{q}$ (par exemple par l'intermédiaire d'une commande à grands gains et dans les cas où les accélérations et les perturbations ne sont pas trop grandes), nous pouvons nous contenter de considérer comme grandeur de pseudo-commande, une "consigne de vitesse dans se_3", c'est à dire le torseur cinématique désiré T_c.

Idéalement, celui-ci aurait donc la forme :

$$T_c = \left(\frac{\partial e}{\partial \bar{r}}\right)^{-1} (-\lambda e - \frac{\partial e}{\partial t}) \tag{4.4}$$

et, si $T_c = T$, l'utilisation de (4.4) dans (4.2) nous redonne bien le comportement souhaité (4.1).

Toutefois, comme nous l'avons déjà vu, seules des approximations peuvent être utilisées dans (4.4), et la forme générale de T_c utilisée en pratique sera donc :

$$T_c = \left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} (-\lambda e - \widehat{\frac{\partial e}{\partial t}}) \tag{4.5}$$

L'utilisation de (4.5) dans (4.2) nous donne alors :

$$\dot{e} = -\lambda \frac{\partial e}{\partial \bar{r}} \left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} e - \frac{\partial e}{\partial \bar{r}} \left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} \widehat{\frac{\partial e}{\partial t}} + \frac{\partial e}{\partial t} \tag{4.6}$$

En supposant pour simplifier l'analyse, bien que cela ne soit pas strictement nécessaire [SAM 90b], $\frac{\partial e}{\partial t} = \widehat{\frac{\partial e}{\partial t}} = 0$, on constate alors, en utilisant (4.6) dans l'expression $\frac{\partial}{\partial t}(\frac{1}{2}\|e\|^2) = \dot{e}^T e$, que la condition (3.13) est suffisante à assurer la décroissance de $\|e\|$.

Revenons à présent à la fonction de tâche donnée par l'équation (3.22) et considérons le cas où les matrices W et D sont constantes, on a alors :

$$\frac{\partial e}{\partial t} = W^+ \frac{\partial e_1}{\partial t} + \alpha(I_n - W^+W)\dot{g}_s^{\bar{r}} \tag{4.7}$$

Le terme $\frac{\partial e_1}{\partial t}$ représente la contribution d'un éventuel mouvement autonome des cibles et est en général inconnu. Le choix $\widehat{\frac{\partial e_1}{\partial t}} = 0$, qui sera celui effectué dans la suite, peut conduire, dans le cas où ce mouvement existe, à une erreur de poursuite d'amplitude décroissant en fonction de λ. Par contre, si le coût secondaire choisi par l'utilisateur, permet de connaître $\dot{g}_s^{\bar{r}}$, ce qui est le cas pour tout suivi de trajectoire, on peut choisir :

$$\widehat{\frac{\partial e}{\partial t}} = \alpha\,(I_n - W^+W)\dot{g}_s^{\bar{r}} \tag{4.8}$$

Enfin, nous supposons dans les applications traitées que les choix des modèles d'interaction effectués permettent de satisfaire la condition (3.16), au moins autour de l'équilibre, et que α est suffisamment "petit" dans les tâches hybrides de forme (3.22) que nous utiliserons. Le choix (3.19) peut alors s'appliquer, ce qui signifie $\widehat{\frac{\partial e}{\partial \bar{r}}} = I$, et, finalement, la consigne en vitesse utilisée aura la forme simple :

$$T_c = -\lambda e - \lambda\alpha\,(I_n - W^+W)\dot{g}_s^{\bar{r}} \tag{4.9}$$

4.2 Résultats de simulation

Nous allons présenter dans cette partie les résultats de simulation de la réalisation de trois tâches, chacune effectuée respectivement à partir d'informations de type points (positionnement), droites (suivi de route) et cercles (positionnement à l'entrée d'un tuyau).

4.2.1 Positionnement

Supposons, par exemple, que l'on veuille placer la caméra par rapport à un objet plan, repérable par quatre points formant un carré. Les signaux-capteurs choisis pour réaliser ce positionnement sont les coordonnées des quatre points : $s = (X_1, \cdots, X_4, Y_1, \cdots, Y_4)$. La position finale souhaitée, par exemple parallèle au plan de l'objet et telle que l'image des quatre points forment un carré centré dans l'image, permet de construire le motif dans l'image $s^* = (-a, a, a, -a, a, a, -a, -a)$ où $a = l/2z^*$, l étant la longueur des côtés du carré et z^* la distance finale souhaitée entre la caméra et l'objet. On obtient facilement la matrice d'interaction associée à s^*, en utilisant l'équation (2.21) :

$$L^T_{|s=s^*} = \begin{pmatrix} -1/z^* & 0 & -a/z^* & -a^2 & -1-a^2 & a \\ -1/z^* & 0 & a/z^* & a^2 & -1-a^2 & a \\ -1/z^* & 0 & a/z^* & -a^2 & -1-a^2 & -a \\ -1/z^* & 0 & -a/z^* & a^2 & -1-a^2 & -a \\ 0 & -1/z^* & a/z^* & 1+a^2 & a^2 & a \\ 0 & -1/z^* & a/z^* & 1+a^2 & -a^2 & -a \\ 0 & -1/z^* & -a/z^* & 1+a^2 & a^2 & -a \\ 0 & -1/z^* & -a/z^* & 1+a^2 & -a^2 & a \end{pmatrix} \tag{4.10}$$

Il est possible, sans trop de difficultés, de calculer $L^{T+}_{|s=s^*}$ et on obtient :

$$L^{T+}_{|s=s^*} = \begin{pmatrix} z^*d_1 & z^*d_1 & z^*d_1 & z^*d_1 & -z^*d_2 & z^*d_2 & -z^*d_2 & z^*d_2 \\ -z^*d_2 & z^*d_2 & -z^*d_2 & z^*d_2 & z^*d_1 & z^*d_1 & z^*d_1 & z^*d_1 \\ -z^*d_3 & z^*d_3 & z^*d_3 & -z^*d_3 & z^*d_3 & z^*d_3 & -z^*d_3 & -z^*d_3 \\ -d_4 & d_4 & -d_4 & d_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_4 & -d_4 & d_4 & -d_4 \\ d_3 & d_3 & -d_3 & -d_3 & d_3 & -d_3 & -d_3 & d_3 \end{pmatrix} \tag{4.11}$$

$$\text{avec} \begin{cases} d_1 = -1/4\ ,\ d_2 = z^*(1+a^2)/4a^2 \\ d_3 = 1/8a\ ,\ d_4 = 1/4a^2 \end{cases}$$

La matrice $L^T_{|s=s^*}$ est de rang plein, aussi peut-on choisir pour W la matrice identité I_6. La matrice de combinaison est alors $D = L^{T+}_{|s=s^*}$ et on en déduit, en utilisant l'équation (3.22), la tâche e qui sera régulée par la loi de "commande " $T_c = -\lambda e$:

$$e = e_1 = D(s(\bar{r}, t) - s^*) \tag{4.12}$$

Remarques

- Si les dimensions du carré sont inconnues, il est alors impossible de connaître la distance finale z^* entre la caméra et l'objet. Un positionnement, à une distance inconnue de l'objet, est cependant possible : il suffit de se fixer la longueur souhaitée $2a$ du carré dans l'image et de prendre pour D la matrice :

$$D = \begin{pmatrix} d_1 & d_1 & d_1 & d_1 & -d_2 & d_2 & -d_2 & d_2 \\ -d_2 & d_2 & -d_2 & d_2 & d_1 & d_1 & d_1 & d_1 \\ -d_3 & d_3 & d_3 & -d_3 & d_3 & d_3 & -d_3 & -d_3 \\ -d_4 & d_4 & -d_4 & d_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_4 & -d_4 & d_4 & -d_4 \\ d_3 & d_3 & -d_3 & -d_3 & d_3 & -d_3 & -d_3 & d_3 \end{pmatrix} \tag{4.13}$$

En effet, on a alors :

$$(DL^T W^T)_{|s=s^*} = \begin{pmatrix} I_3/z^* & 0 \\ 0 & I_3 \end{pmatrix} > 0 \qquad (4.14)$$

- Les six degrés de liberté du robot sont utilisés pour réaliser cette tâche (e_1 est de dimension 6) puisque une seule situation de SE_3 entre la caméra et l'objet réalise $e_1 = 0$. Il est donc impossible d'introduire une tâche secondaire dans cet exemple de positionnement.

Sur la Figure 3 sont regroupés les résultats de simulation de cette tâche. Les deux fenêtres du haut représentent la position relative de la caméra (symbolisée par une pyramide) par rapport à l'objet. Les deux fenêtres du bas représentent la scène vue par la caméra. Les fenêtres de gauche correspondent à la position initiale et celles du milieu à la position finale. Enfin, les fenêtres de droite représentent le comportement de $\|s(\bar{r},t) - s^*\|$ (en haut) et de chaque composante de T_c (en bas) en fonction du temps. Notons que, pour cette simulation, λ a été fixé égal à 0.1.

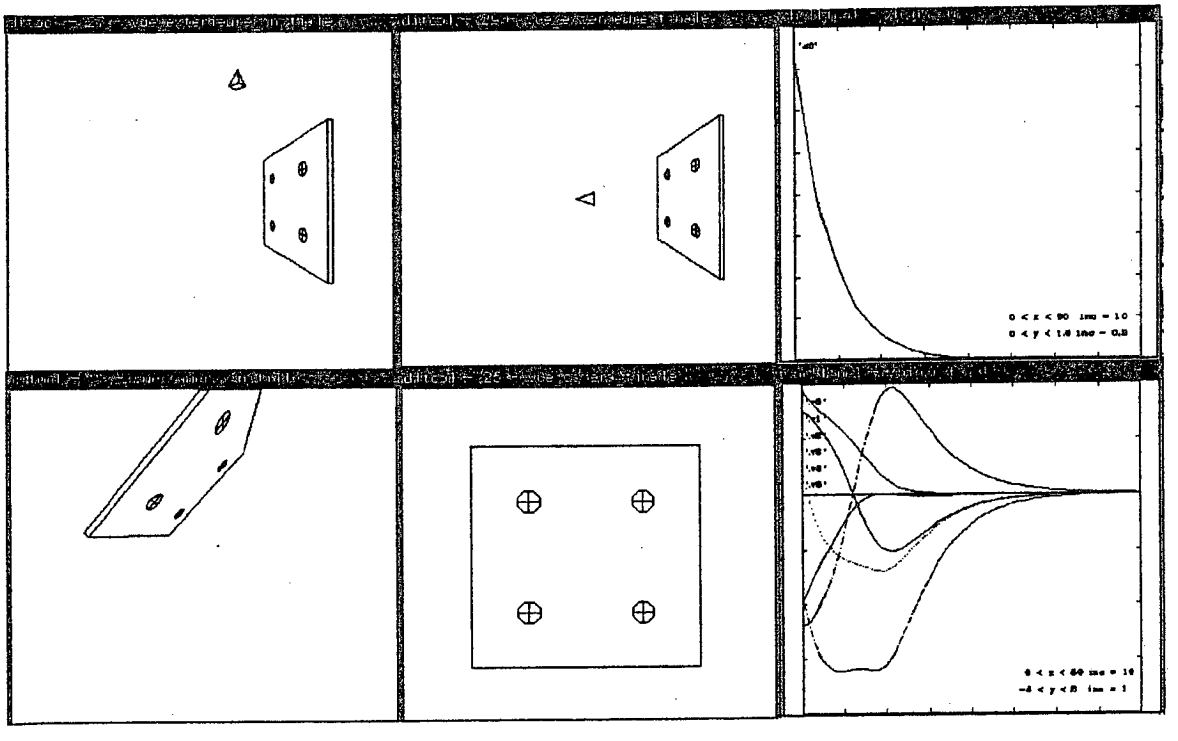


Figure 3 : Positionnement par rapport à un carré (sans bruit)

Si l'on introduit artificiellement du bruit sur les mesures effectuées dans l'image (bruit uniforme d'amplitude 2 pixels sur chaque coordonnée des points) et sur la position de la caméra (bruit uniforme d'amplitude 1cm sur les axes de translation et de 2 degrés sur les axes de rotation) pour simuler, d'une part, d'éventuelles erreurs de mesure et, d'autre part, une mauvaise calibration ou une mauvaise commande du robot, on obtient des résultats similaires, regroupés Figure 4, la convergence et la stabilité n'étant que peu perturbées par l'introduction des différents bruits.

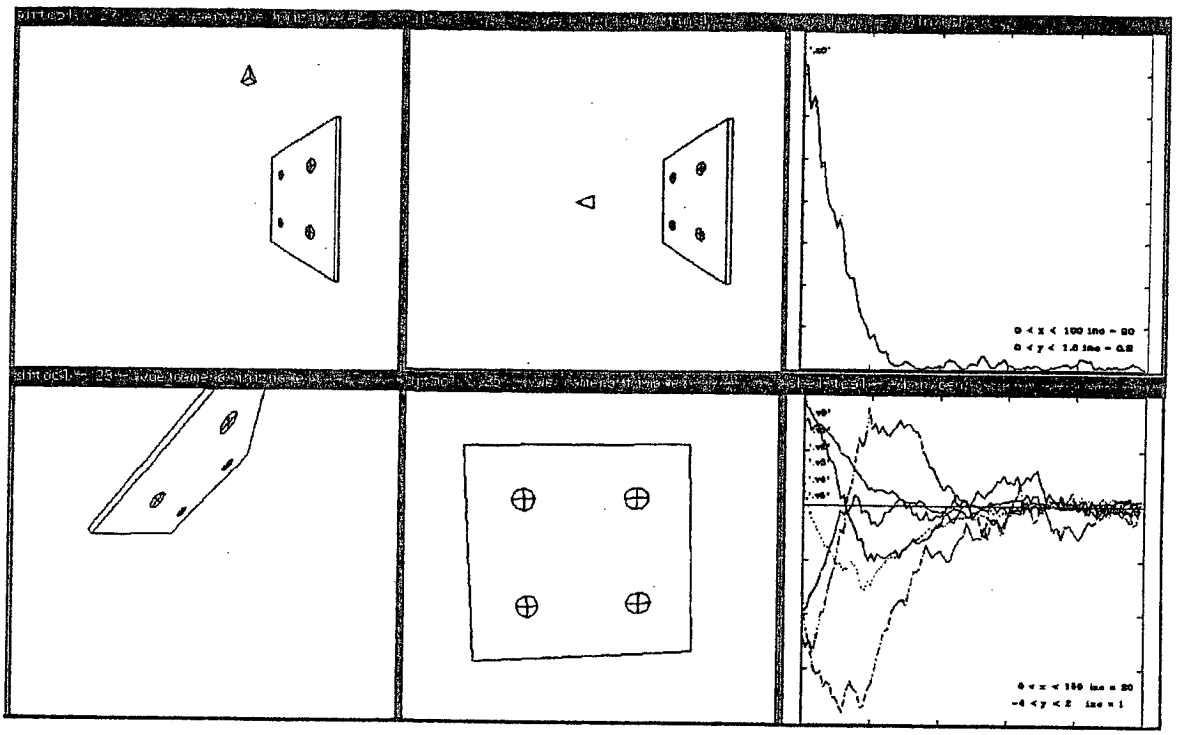


Figure 4 : Positionnement par rapport à un carré (avec bruit)

4.2.2 Suivi de route

Considérons à présent le cas où l'on souhaite positionner la caméra par rapport à une "route" de largeur l, symbolisée par trois droites parallèles coplanaires représentant les bandes latérales et centrale de cette route. La situation souhaitée de la caméra est choisie de telle sorte que :

- la caméra soit à une hauteur y^* et au milieu de la file droite de la route,
- l'axe $\vec{z}$ de la caméra coïncide avec sa direction et l'axe $\vec{y}$ avec sa verticale.

En utilisant les équations (2.23) et (2.25), on obtient aussitôt les fonctions $h(\bar{x}, \bar{p})$ et $g(\bar{X}, \bar{q})$ correspondant aux trois droites :

$$h_1(\bar{x}, \bar{p}^*) : \begin{cases} y + y^* = 0 \\ x + l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_1^* = \arctan(-l/4y^*) \\ \rho_1^* = 0 \end{cases} \tag{4.15}$$

$$h_2(\bar{x}, \bar{p}^*) : \begin{cases} y + y^* = 0 \\ x - l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_2^* = \arctan(l/4y^*) \\ \rho_2^* = 0 \end{cases} \tag{4.16}$$

$$h_3(\bar{x}, \bar{p}^*) : \begin{cases} y + y^* = 0 \\ x - 3l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_3^* = \arctan(3l/4y^*) \\ \rho_3^* = 0 \end{cases} \tag{4.17}$$

Les signaux-capteurs choisis pour réaliser cette tâche sont évidemment les paramètres représentant les trois droites : $s = (\theta_1, \rho_1, \theta_2, \rho_2, \theta_3, \rho_3)$ et on a : $s^* =$

$(\theta_1^*, 0, \theta_2, 0, \theta_3^*, 0)$. De plus, on peut facilement calculer la matrice d'interaction associée à s^* en utilisant l'équation (2.30) :

$$L^T_{|s=s^*} = \begin{pmatrix} -\cos^2\theta_1^*/y^* & -\cos\theta_1^*\sin\theta_1^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & sin\theta_1^* & -\cos\theta_1^* & 0 \\ -\cos^2\theta_2^*/y^* & -\cos\theta_2^*\sin\theta_2^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & sin\theta_2^* & -\cos\theta_2^* & 0 \\ -\cos^2\theta_3^*/y^* & -\cos\theta_3^*\sin\theta_3^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & sin\theta_3^* & -\cos\theta_3^* & 0 \end{pmatrix} \tag{4.18}$$

La matrice $L^T_{|s=s^*}$ est toujours de rang 5, et on a Ker $L^T_{|s=s^*} = (0\ 0\ 1\ 0\ 0\ 0)^T$.

Remarques :

- Comme on pouvait s'y attendre intuitivement, un mouvement de translation sur l'axe des $\vec{z}$, laisse invariant le motif considéré.

- Si l'on dispose d'un robot mobile aux trois degrés de liberté les translations sur les axes $\vec{x}$,$\vec{z}$ et la rotation autour de l'axe $\vec{y}$, cette même tâche peut être réalisée en utilisant les informations fournies par une seule droite : $s = (\theta_1, \rho_1)$. Par contre, avec un robot à six degrés de liberté, s doit être de dimension ≥ 5.

- L'ambiguïté de la représentation (ρ, θ), évoquée en section 2.2.2 et concernant les différents choix possibles de θ $(\theta + 2k\pi, \forall k \in \mathbb{N})$, est levée en modulant à 2π l'erreur $(\theta - \theta^*)$.

Appliquons à présent la démarche décrite en section 3.1.3 pour calculer le vecteur e. On peut prendre pour W la matrice 5×6 :

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.19}$$

La matrice de permutation D est choisie égale à $WL^{T+}_{|s=s^*}$ et, en utilisant l'équation (3.22), on obtient le vecteur de tâche e, permettant en outre de réaliser une tâche secondaire e_2 à choisir par l'utilisateur :

$$e = W^+WL^{T+}_{|s=s^*}(s(\bar{r}, t) - s^*) + \alpha\left(I_6 - W^+W\right)g_s^{\bar{r}} \tag{4.20}$$

La tâche secondaire peut consister à décrire une trajectoire le long l'axe $\vec{z}$, par exemple à avancer à vitesse constante V. Cette tâche secondaire peut s'écrire sous la forme $e_2 = z(t) - z_0 - Vt$, où $z(0) = z_0$. On en déduit le coût secondaire à minimiser $h_s = \frac{1}{2}(z(t) - z_0 - Vt)^2$. D'où :

$$g_s^{\bar{r}} = (0\ \ 0\ \ (z(t) - z_0 - Vt)\ \ 0\ \ 0\ \ 0)^T \tag{4.21}$$

Les tâches e_1 et e_2 sont alors compatibles et indépendantes puisqu'on obtient :

$$e = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} L^{T+}_{|s=s^*} (s(\bar{r},t) - s^*) + \alpha \begin{pmatrix} 0 \\ 0 \\ z(t) - z_0 - Vt \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.22}$$

et la consigne T_c est obtenue à partir de l'équation (4.9) en remarquant que $\dot{g}^{\bar{r}}_s = (0\ 0\ -V\ 0\ 0\ 0)^T$:

$$T_c = -\lambda\, e + \begin{pmatrix} 0 \\ 0 \\ V \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.23}$$

Des résultats de simulation sont présentés sur la Figure 5, configurée comme les figures précédentes. Les différents gains ont été fixés, pour cette simulation, aux valeurs suivantes : $\lambda = 0.1, \alpha = 10$ et la vitesse V a été choisie égale à 1.25 cm/s.

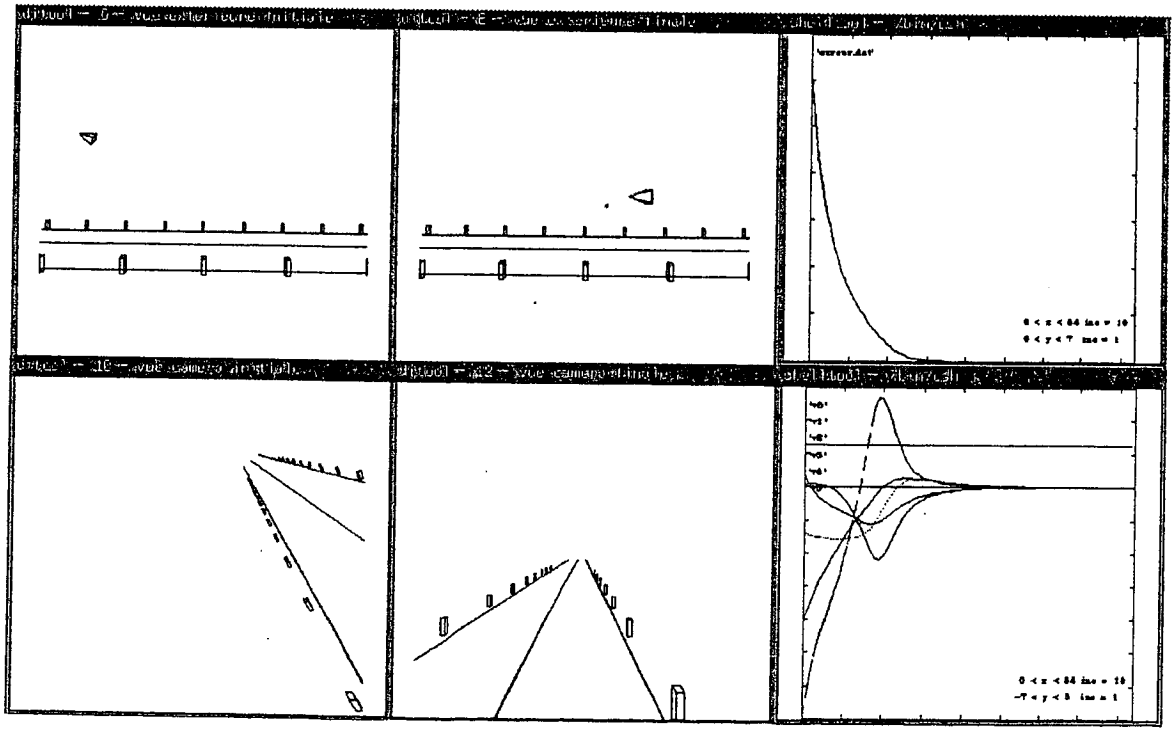


Figure 5 : Suivi de route (sans bruit)

4.2.3 Positionnement à l'entrée d'un tuyau

Considérons à présent une scène constitué d'un "tuyau" symbolisé par un cylindre de rayon r et de hauteur h. La tâche consiste à positionner la caméra à l'entrée de ce cylindre de manière à obtenir dans l'image deux cercles centrés de rayon respectivement $R_1 = r/z_1$ et $R_2 = r/z_2$ où z_1 est la distance souhaitée entre la caméra et l'entrée du tuyau et où $z_2 = z_1 + h$.

Comme on l'a vu en section 2.2.2, les vecteurs L_{A_1} et L_{A_2}, représentant les torseurs d'interaction des paramètres A_1 et A_2 caractérisant une ellipse, sont nuls pour une situation de la caméra telle que l'image d'un cercle 3D soit un cercle centré. Aussi ne peut-on pas prendre dans le vecteur s les signaux capteurs A_1 et A_2, ceux-ci étant inefficaces pour réaliser la tâche donnée. On choisit alors : $s = (A_{3_1}, A_{3_2}, A_{4_1}, A_{4_2}, A_{5_1}, A_{5_2},)$ où A_{i_j} est le paramètre A_i de l'ellipse j considérée. La position souhaitée de la caméra permet de construire le motif $s^* = (0, 0, 0, 0, -R_1^{*^2}, -R_2^{*^2})$. On obtient facilement la matrice d'interaction associée à s^*, en utilisant l'équation (2.34) :

$$L^T_{|s=s^*} = \begin{pmatrix} 1/z_1^* & 0 & 0 & 0 & 1+R_1^{*^2} & 0 \\ 1/z_2^* & 0 & 0 & 0 & 1+R_2^{*^2} & 0 \\ 0 & 1/z_1^* & 0 & -1-R_1^{*^2} & 0 & 0 \\ 0 & 1/z_2^* & 0 & -1-R_2^{*^2} & 0 & 0 \\ 0 & 0 & -2R_1^{*^2}/z_1^* & 0 & 0 & 0 \\ 0 & 0 & -2R_2^{*^2}/z_2^* & 0 & 0 & 0 \end{pmatrix} \tag{4.24}$$

La matrice $L^T_{|s=s^*}$ est de rang 5, et on a Ker $L^T_{|s=s^*} = (0\ 0\ 0\ 0\ 0\ 1)^T$. On choisit donc pour W la matrice 5×6 : $(I_5\ 0)$. Par ailleurs, une rotation θ_z autour de l'axe $\vec{z}$ laisse le motif invariant, aussi choisit-on comme tâche secondaire de faire tourner la caméra autour de son axe $\vec{z}$ à une vitesse constante ω : $e_2 = \theta_z(t) - \theta_{z_0} - \omega t$. Les tâches e_1 et e_2 sont alors compatibles et indépendantes et on a, en procédant comme précédemment :

$$T_c = -\lambda \begin{pmatrix} I_5 \\ 0 \end{pmatrix} D\,(s(\bar{r},t) - s^*) + \lambda\alpha \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \theta_z(t) - \theta_{z_0} - \omega t \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \omega \end{pmatrix} \tag{4.25}$$

où la matrice de permutation D est donnée par $D = W L^{T+}_{|s=s^*}$, soit :

$$D = \begin{pmatrix} d_1 & d_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_1 & d_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_1^2 z_2 d_3 & R_2^2 z_1 d_3 \\ 0 & 0 & -z_1 d_4 & z_2 d_4 & 0 & 0 \\ z_1 d_4 & -z_2 d_4 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.26}$$

$$\text{avec} \begin{cases} d_1 = \dfrac{(1+R_2^2) z_1 z_2}{(1+R_2^2) z_2 - (1+R_1^2) z_1}\ ,\ d_2 = \dfrac{-(1+R_1^2) z_1 z_2}{(1+R_2^2) z_2 - (1+R_1^2) z_1} \\ d_3 = \dfrac{-z_1 z_2}{2(R_1^4 z_2^2 + R_2^4 z_1^2)}\ ,\ d_4 = \dfrac{-1}{(1+R_2^2) z_2 - (1+R_1^2) z_1} \end{cases}$$

La Figure 6 présente les résultats obtenus en choisissant $\alpha = 10$, $\lambda = 0.1$ et $\omega = 4.5$ degrés/s.

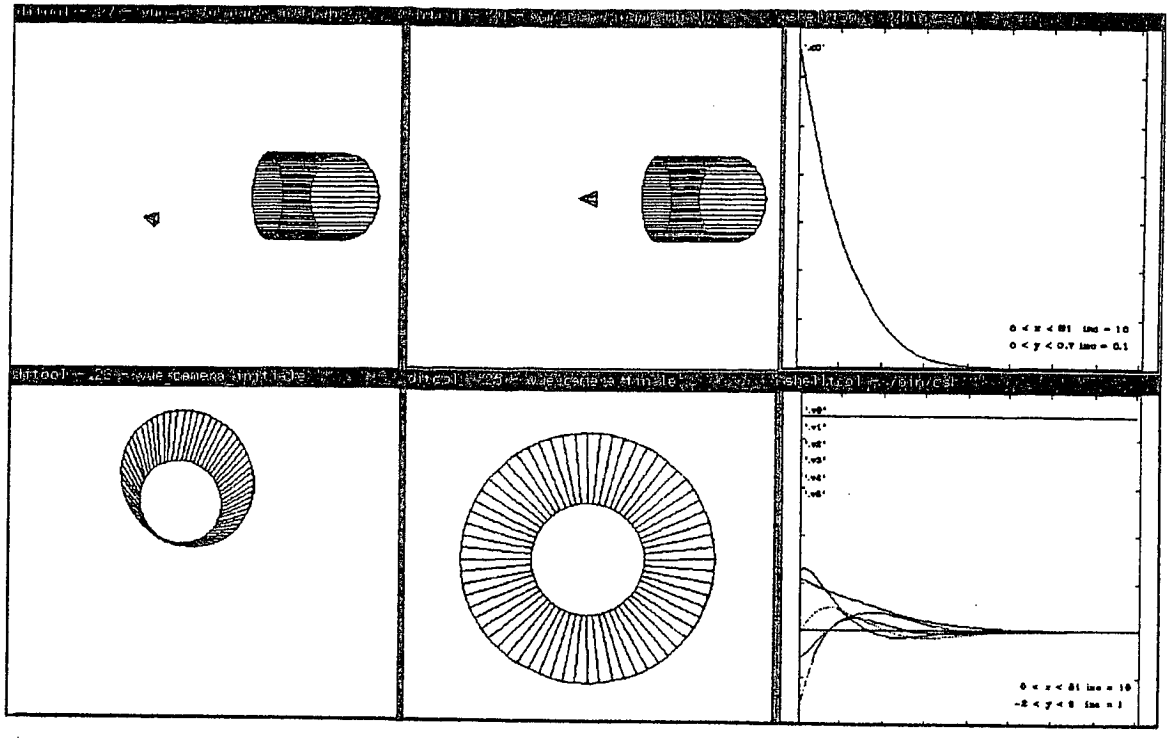


Figure 6 : Positionnement à l'entrée d'un tuyau (sans bruit)

4.3 Expérience réelle

La tâche de positionnement par rapport à un carré a été implantée sur notre banc expérimental composé d'une caméra CCD embarquée sur un robot manipulateur à 6 degrés de liberté. La calibration de la caméra qui permet de se ramener au modèle donné Figure 1 et l'identification de la matrice de passage entre le poignet du robot et le repère de la caméra ont été effectuées [CHAU 89]. La consigne T_c, calculée dans le repère de la caméra, peut donc être traduite sous forme de "commande dans l'espace articulaire" en utilisant le jacobien inverse J^{-1} du robot :

$$\dot{q}_c = J^{-1} T_c \tag{4.27}$$

Le calcul de J^{-1} est effectué, sur une carte spécifique à processeur 68020, avec une fréquence d'échantillonnage de 5 ms. Par ailleurs, la scène est constituée de quatre disques blancs sur fond noir. La simplicité des images acquises sur cette scène permet de calculer en moins de 20 ms le centre de gravité de l'image des disques, qui fournit s, et la consigne T_c, qui est transmise au processeur 68020. Les informations de toutes les trames paires sont donc utilisées à la cadence vidéo.

La Figure 7 présente une séquence d'images acquises lors d'un positionnement (avec $\lambda = 0.5$). Une image sur 10 est visualisée. Les courbes obtenues représentant $\|s\|$ et chaque composante de T_c sont similaires à celles obtenues en simulation.

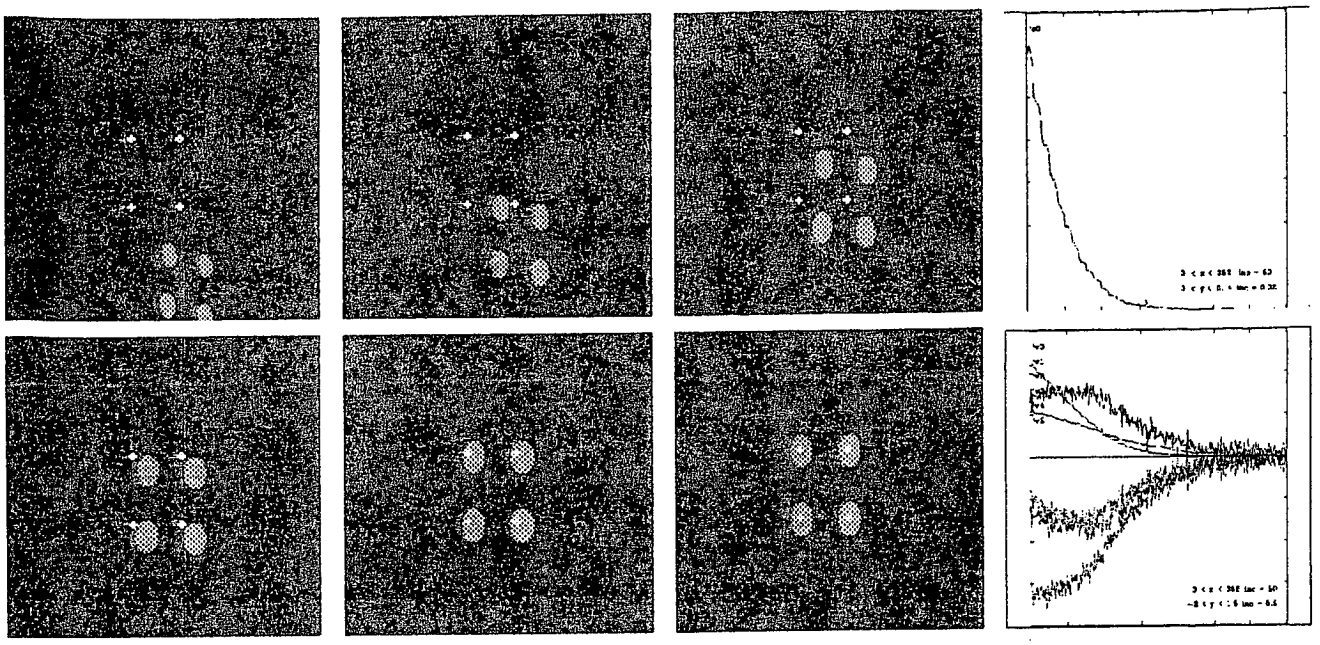


Figure 7 : Positionnement par rapport à un carré

Notons enfin que, lorsque l'objet se déplace suffisamment lentement pour rester dans le champ de la caméra entre deux acquisitions, on constate, comme annoncé en section 4.1, une erreur de poursuite décroissant en fonction de λ. Celle-ci se stabilise, pour une vitesse constante de l'objet, à environ une dizaine de pixels dans l'image pour un fort gain ($\lambda = 3$) permettant cependant de conserver la stabilité du système. Cette erreur s'annule en quelques itérations de la boucle de commande dès que l'objet reste fixe. Il est très certainement possible de diminuer cette erreur de poursuite en introduisant dans la consigne T_c une estimation de la vitesse de l'objet, ce qui peut correspondre à choisir comme consigne :

$$T_c = -\lambda e - \widehat{\frac{\partial e_1}{\partial t}} \tag{4.28}$$

Chapitre 5

Conclusion

Nous avons essayé de montrer dans ce rapport que l'évolution des techniques permet d'envisager, aujourd'hui, de contrôler l'interaction entre un processus et son environnement physique par un schéma de commande en boucle fermée directement sur des informations issues de capteurs de vision. L'exploitation d'une telle approche en robotique nécessite la réponse à deux questions:

- Quels types de tâches robotiques sont concernées par cette approche? Avec le corollaire suivant, peut-on définir des *classes de tâches génériques* et comment les spécifier?
- Comment, pour une classe de tâches données, peut-on construire une commande capable de la réaliser en garantissant de bonnes propriétés de convergence et de robustesse?

Pour répondre au premier point, rappelons l'hypothèse fondamentale qui soustend notre démarche : seules les modifications géométriques de l'environnement sont susceptibles de faire varier la sortie du capteur. De ce fait, les tâches que nous serons capables d'appréhender avec ce type d'approche seront celles qui nécessiteront un contrôle d'interactions de nature géométrique entre le robot et l'environnement. Typiquement, cela se traduira par le contrôle de l'attitude relative entre un repère lié au capteur et un repère lié à l'environnement. Un formalisme, issu de la *Mécanique*, a déjà été utilisé en robotique d'assemblage et consiste à spécifier cet attitude en termes de *liaisons* (point sur point, point sur droite, point sur plan...) représentant les contraintes désirées entre les deux repères et caractérisant les degrés de liberté nécessaires à une réalisation correcte de la tâche. Ce formalisme permet de définir les classes de tâches élémentaires (positionnement, suivi...) qui seront utilisées pour décrire une application plus complexe. Dans l'approche *asservissement visuel* que nous avons développée, nous avons établi pour un certain nombre de primitives géométriques simples dans la scène (points, droites, cercles, sphères), la structure et les propriétés du torseur d'interaction permettant ainsi d'associer à un motif de la scène une liaison définie dans le formalisme précédent. La principale difficulté rencontrée est liée à la richesse des signaux fournies par un capteur de vision, richesse qui fait qu'un type donné de liaisons peut être obtenu par une infinité de motifs différents. La préférence d'un motif particulier vis à vis d'un autre devra donc s'appuyer sur d'autres critères incluant des aspects liés au traitement d'images (robustesse d'extraction des informations dans l'image vis

à vis des bruits de mesure, coût d'extraction ...) et des aspects liés à la commande (sensibilité, découplage...). La prise en compte de ces aspects n'est encore qu'à l'état d'ébauche et beaucoup de travail reste à faire dans ce domaine.

Le deuxième point, concernant l'aspect "mise en œuvre de commandes robustes", a été poussé un peu plus loin en s'appuyant sur les résultats déjà acquis sur la commande robuste des robots manipulateurs. Pour ce qui est des tâches de positionnement vis à vis d'un objet statique, les résultats obtenus donnent toutes satisfactions. Cependant la commande réalisée ne prennant pas explicitement en compte une éventuelle mobilité des objets, il s'en suit des erreurs de trainage. Une commande plus sophistiquée incluant un estimateur de la vitesse de la cible est en cours de définition.

Enfin pour conclure, la mise en œuvre de commandes en boucle fermées sur des informations visuelles implique une compatibilité entre la fréquence d'acquisition des mesures (vision) et la fréquence d'échantillonage du processus à commander. Bien que les informations visuelles utilisées dans une approche *asservissement visuel* soient de bas niveau (points, droites, ellipses...), il est nécessaire d'utiliser des architectures spécialisées pour les extraire en Temps réel (c'est à dire à la fréquence d'acquisition TV). Un travail de spécification mené en collaboration avec le Laboratoire d'Electronique de Clermont-Ferrand, est en cours, devant déboucher sur la conception et la réalisation d'une carte spécialisée au format MAXBUS et bien adaptée à l'approche *asservissement visuel.*

Références

[AGI 79] G. J. Agin : *Real Time Control of a Robot with a Mobile Camera*, Technical Note 179, SRI International, February 1979.

[BOU 88] P. Bouthemy : *Modèles et méthodes pour l'analyse du mouvement dans une séquence d'images*, Technique et Science Informatiques, Vol. 7, n.6, pp 527-546, 1988.

[CHAU 89] F. Chaumette, P. Rives : *Réalisation et calibration d'un système expérimental de vision composé d'une caméra embarquée sur un robot manipulateur*, Rapport de recherche INRIA, n.994, Mars 1989.

[COR 89] P. I. Corke, R. P. Paul : *Video-Rate Visual Servoing for Robots*, First International Symposium on Experimental Robotics, Montreal, Canada, June 1989.

[ESP 87] B. Espiau, P. Rives : *Closed-Loop Recursive Estimation of 3D Features for a Mobile Vision System*, IEEE Int. Conf. on Robotics and Automation, Raleigh, North Carolina, USA, March 30- April 3, 1987.

[FED 89a] J. T. Feddema, C. S. G. Lee and O. R. Mitchell : *Automatic selection of image features for visual servoing of a robot manipulator*, Conf. IEEE Robotics and Automation, Scottsdale, Arizona, USA, May 14-19, 1989.

[FED 89b] J. T. Feddema, O. R. Mitchell : *Vision-Guided Servoing with Feature-Based Trajectory Generation*, IEEE Transaction on Robotics and Automation, Vol. 5, n.5, October 1989.

[FEN 79] C. L. Fennema, W. B. Thomson : *Velocity Determination in Scenes Containing Several Moving Objects*, Computer Graphics and Image Processing, 9:301-315, 1979.

[GIL 80] A. L. Gilbert and al. : *A Real-Time Video Tracking System*, IEEE Transaction on PAMI, Vol. PAMI-2, n.1, January 1980.

[KAB 88] M. Kabuka, E. McVey, P. Shironoshita : *An Adaptive Approach to Video Tracking*, IEEE Journal of Robotics and Automation, 4(2):228-236, April 1988.

[RIV 87] P. Rives : *Dynamic vision: theoretical capabilities and practical problems*, NATO Workshop on Kinematic and Dynamic Issues in Sensor Based Control, Italy, October 1987.