



**HAL**  
open science

## Positionnement par rapport à un objet plan de forme inconnue par asservissement visuel 2D

Christophe Collewet, François Chaumette, Laurence Wallian, Philippe  
Marchal

► **To cite this version:**

Christophe Collewet, François Chaumette, Laurence Wallian, Philippe Marchal. Positionnement par rapport à un objet plan de forme inconnue par asservissement visuel 2D. [Rapport de recherche] RR-3419, INRIA. 1998. inria-00073270

**HAL Id: inria-00073270**

**<https://inria.hal.science/inria-00073270v1>**

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.

***Positionnement par rapport à un objet plan de  
forme inconnue par asservissement visuel 2D***

Christophe Collewet , François Chaumette , Laurence Wallian, Philippe Marchal

**No 3419**

Mai 1998

\_\_\_\_\_ THÈME 3 \_\_\_\_\_



*Rapport  
de recherche*



## Positionnement par rapport à un objet plan de forme inconnue par asservissement visuel 2D

Christophe Collewet\*, François Chaumette†, Laurence Wallian\*,  
Philippe Marchal\*

Thème 3 — Interaction homme-machine,  
images, données, connaissances  
Projet Vista

Rapport de recherche n 3419 — Mai 1998 — 60 pages

**Résumé :** Dans ce rapport nous décrivons la façon de réaliser une tâche particulière de positionnement par asservissement visuel 2D dans le cas où la consigne visuelle, la situation de l'objet observé et ses dimensions sont inconnues. Par contre, nous supposons l'objet plan et immobile. Nous traitons dans un premier temps le cas d'un objet filiforme et nous montrons ensuite comment l'approche retenue peut se généraliser à un objet dont trois points caractéristiques existent puis, enfin, au cas d'un objet de forme quelconque. La loi de commande utilisée est basée sur la maximisation d'un critère au cours du mouvement. Des résultats d'expérimentations, y compris en environnement non calibré, concluent ce rapport.

**Mots-clé :** Asservissement visuel 2D, consigne visuelle inconnue, tâche de positionnement, approche critère.

*(Abstract: pto)*

\* Cemagref Rennes, {christophe.collewet}{laurence.wallian}{philippe.marchal}@cemagref.fr

† francois.chaumette@irisa.fr

## Positioning task relative to a planar object of any shape by 2D visual servoing

**Abstract:** In this report we describe the way to achieve a particular positioning task by 2D visual servoing when the desired value of the visual features, the pose and dimension of the observed object are unknown. On the other hand, we assume this object is planar and motionless. First, we deal with a threadlike object and then we show how our approach can be generalized to an object with three particular points and, finally, to an object of any shape. The control scheme we used is based on the minimization of a criterion during the motion. Experimental results, including uncalibrated environments, conclude this report.

**Key-words:** 2D visual servoing, unknown desired value of the visual features, positioning task, criteria approach.

## Sommaire

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>L'approche asservissement visuel 2D</b>	<b>6</b>
<b>3</b>	<b>Étude d'un objet simple: le segment</b>	<b>7</b>
3.1	Spécification de la tâche . . . . .	7
3.2	Centrage de l'image . . . . .	10
3.2.1	Utilisation de quatre informations visuelles . . . . .	10
3.2.2	Utilisation de trois informations visuelles . . . . .	10
3.3	Détermination d'une fonction de tâche secondaire . . . . .	11
3.3.1	Résultats de simulations non bruitées . . . . .	14
3.3.2	Résultats de simulations bruitées . . . . .	16
3.4	Scrutation d'informations visuelles . . . . .	19
3.4.1	Modélisation de la taille de l'image à l'aide d'informations dynamiques . . . . .	19
3.4.2	Modélisation de la taille de l'image à l'aide d'informations statiques . . . . .	21
3.4.3	Application à la commande . . . . .	25
3.4.4	Simulations non bruitées . . . . .	27
3.4.5	Simulations bruitées . . . . .	27
3.4.6	Discussion sur l'erreur de positionnement . . . . .	27
<b>4</b>	<b>Étude d'un objet plan de forme quelconque</b>	<b>30</b>
4.1	Étude du cas où trois points caractéristiques existent . . . . .	30
4.1.1	Spécification de la tâche principale . . . . .	31
4.1.2	Modélisation de la surface de l'objet dans l'image . . . . .	33
4.2	Résultats expérimentaux — cas où trois points de l'objet sont connus . . . . .	35
4.2.1	Description de la plate-forme expérimentale . . . . .	35
4.2.2	Vérification des résultats . . . . .	36
4.2.3	Résultats expérimentaux . . . . .	37
4.2.4	Comportement de la loi de commande en environnement non calibré . . . . .	46
4.3	Étude d'un objet plan de forme quelconque . . . . .	48
4.3.1	Choix des informations visuelles et obtention d'une loi de commande . . . . .	49
4.3.2	Résultats expérimentaux . . . . .	50
<b>5</b>	<b>Conclusion et perspectives</b>	<b>51</b>

## 1 Introduction

Les premières utilisations de robots sont apparues aux USA dans les années 1960 pour les besoins de l'industrie manufacturière [28]. Ils étaient alors cantonnés à des rôles d'automates mécaniques programmables où les trajectoires effectuées étaient apprises puis répétées quelque soit leur environnement [5].

L'évolution des capteurs extéroceptifs, et en particulier des capteurs de vision, a révélé, avec l'émergence de l'électronique et de l'informatique, les potentialités d'utilisation des robots. Tout d'abord leur degré de flexibilité a été accru. Ils ne sont plus dédiés à une unique tâche bien définie mais à plusieurs tâches ou à des tâches évoluées où l'environnement proche du robot est susceptible de se modifier.

C'est ainsi que l'utilisation du capteur vision permet maintenant à un système robotisé de suivre un objet mobile, de le saisir éventuellement ou de modifier sa trajectoire en cas d'obstacle ou d'occlusion de l'objet. Citons par exemple [1], [3], [25] et [16].

Parallèlement à cette évolution l'utilisation de la vision a permis d'augmenter la précision du robot. En effet, les premières interactions vision / commande fonctionnaient selon le principe de la boucle ouverte : le robot se déplaçait après une représentation de la scène puis, le cas échéant, se déplaçait une nouvelle fois suite à une modification de la scène. La précision obtenue dépendait directement des capteurs, des asservissements articulaires et de la justesse des modèles utilisés. La première utilisation de la vision en boucle fermée est due à [24] qui décrit comment un capteur de vision pouvait augmenter la précision du positionnement. On parlait alors de retour par vision (*visual feedback*). C'est à Hill et Park [15] que l'on doit l'apparition du terme asservissement visuel (*visual servoing*). Plusieurs approches ont ensuite vu le jour, elles ont été caractérisées dans [22] en deux grandes classes suivant l'utilisation faite des informations visuelles : une première approche nommée asservissement en situation (*position-based control*) et une seconde nommée asservissement basé image (*image-based control*).

Dans un asservissement en situation, les informations visuelles sont extraites de l'image afin d'obtenir la position et l'orientation de la caméra par rapport à l'objet. Différentes approches répondent à ce problème : l'utilisation des techniques photogrammétriques, de stéréovision, d'obtention de la profondeur grâce au mouvement. Elles sont clairement détaillées dans [7].

Par contre, les techniques d'asservissement visuel basé image ne nécessitent pas de reconstruction 3D, les raisonnements sont réalisés directement au niveau du capteur. En effet, à partir d'un vecteur  $\underline{s}$  de dimension  $m$  représentant les informations visuelles courantes, on cherche à atteindre un motif  $\underline{s}^*$  correspondant à la position finale souhaitée du porteur. Le choix des informations visuelles et l'obtention de la relation les liant aux mouvements du porteur sont deux points fondamentaux de cette approche. Cette relation lie la variation

des informations visuelles  $\delta \underline{s}$  à la variation de la situation du porteur  $\delta \underline{r}$  (de dimension  $n$  où  $n$  est le nombre de degrés de liberté contrôlés) par linéarisation de l'expression fortement non linéaire liant  $\underline{s}$  et  $\underline{r}$ . C'est une matrice de dimension  $m \times n$  appelée, suivant les auteurs, matrice d'interaction, jacobien de la tâche ou jacobien de l'image. Elle peut être obtenue sous forme analytique dans le cas d'informations visuelles plus ou moins simples (points [12] et [13], droites, cercles, cylindres, sphères, ellipses [4]). Ce sont en fait les primitives géométriques les plus couramment utilisées [2].

Pourtant, si l'on veut appréhender des objets de nature plus complexe, de telles informations visuelles de type géométrique paraissent insuffisantes comme le soulignent Wells et al. dans [27]. D'autres types d'informations visuelles sont proposés pouvant caractériser la globalité de l'image, par exemple dans [6] les moments d'inertie, dans [2] la projection de l'image sur une ligne et la transformée de Fourier et, dans [27], les descripteurs de Fourier. Dans ces cas, la principale difficulté réside dans l'obtention de la matrice d'interaction et, aujourd'hui, seules des approximations ou des techniques d'apprentissage ont pu appréhender de telles informations visuelles [27], [2].

Par ailleurs, d'autres travaux, basés eux aussi sur des techniques d'apprentissage, permettent l'élaboration de lois de commande qui évitent l'inversion de la matrice d'interaction et par là même contournent les problèmes de singularité [18], [26], [14]. D'autres techniques permettent dans ce but de sélectionner les informations visuelles garantissant ainsi la robustesse de l'asservissement [19], [26].

Tous ces travaux supposent que la consigne visuelle soit parfaitement connue, c'est-à-dire que l'on puisse décrire précisément l'image finale souhaitée. Cette hypothèse, forte, limite grandement le champ applicatif. Peu d'auteurs s'en affranchissent, excepté dans [9] où, grâce à l'utilisation d'informations visuelles dynamiques de type «vitesse dans l'image», des objets plans de forme quelconque peuvent être traités. Nous nous intéresserons dans ce rapport au cas où la consigne, supposée de nature géométrique, est inconnue ou trop complexe et par là même supposée inconnue aussi. Pour ne pas restreindre la généralité du problème, nous ne ferons aucune hypothèse quant à la forme de l'objet observé, par contre, nous le supposerons plan et animé d'une vitesse nulle. On se limitera aussi dans cette étude à la réalisation d'une tâche de positionnement consistant à positionner la caméra en face de l'objet.

Ce rapport sera donc articulé comme suit. Après un rappel de l'approche asservissement visuel 2D, le cas d'un objet filiforme est ensuite traité pour positionner une caméra en face de cet objet. Pour ce faire, une approche fonction de tâche est tout d'abord explorée. Après avoir montré pourquoi la loi de commande ainsi obtenue n'est pas robuste vis-à-vis d'informations visuelles bruitées, une approche basée sur la maximisation d'un critère est proposée. Une identification en ligne par les moindres carrés récurrents avec facteur d'oubli permet d'évaluer ce critère et par suite d'atteindre la position souhaitée. On montre ensuite comment cette approche se généralise au cas d'un objet comportant trois points caractéristiques, puis au cas d'un objet plan de forme quelconque. Des résultats expérimentaux,



y compris en environnement non calibré, valident ensuite les simulations. Finalement, les principaux acquis ainsi que les travaux futurs concluront ce rapport.

## 2 L'approche asservissement visuel 2D

Comme on l'a vu dans le chapitre introductif, les asservissements visuels 2D font l'objet de différentes approches. Mais tous nécessitent la connaissance de la matrice d'interaction [11], elle permet en effet de lier la vitesse des informations visuelles au mouvement du porteur :

$$\dot{\underline{s}} = L_{\underline{s}}^T T_c \quad (1)$$

où :

- $T_c$  est le torseur cinématique. Il représente la vitesse de la caméra par rapport à l'objet et s'écrit  $T_c = (\underline{V}^T, \underline{\Omega}^T)^T$  où  $\underline{V} = (v_x, v_y, v_z)^T$  et  $\underline{\Omega} = (\omega_x, \omega_y, \omega_z)^T$  sont respectivement les vitesses de translation et de rotation exprimées dans le repère de la caméra ;
- $\dot{\underline{s}}$  est la vitesse des informations visuelles ;
- $L_{\underline{s}}^T$  est la matrice d'interaction relative aux informations visuelles  $\underline{s}$ .

L'approche fonction de tâche introduite par Samson [21] en commande référencée capteur puis appliquée au capteur vision par Chaumette [4] permet d'appréhender le problème comme une régulation à zéro d'une fonction de tâche  $\underline{e}_1$  telle que :

$$\underline{e}_1(\underline{r}, t) = C(\underline{s}(\underline{r}, t) - \underline{s}^*) \quad (2)$$

où :

- $C$  est une matrice de dimension  $n \times m$  dite de combinaison, elle permet de prendre en compte les cas où  $m$  est différent de  $n$  (avec  $m$  et  $n$  définis dans le paragraphe d'introduction) ;
- $\underline{s}^*$  représente la valeur des informations visuelles à la position finale souhaitée du porteur ;

- $\underline{e}_1$  est le vecteur fonction de tâche, de dimension  $n$ . Il dépend du temps  $t$  et de la situation du porteur  $\underline{r}$ .

Quand les degrés de liberté ne sont pas tous contraints par la réalisation de  $\underline{e}_1$ , on peut, grâce au formalisme de redondance décrit dans [21] et [20], faire cohabiter une autre tâche dite secondaire,  $\underline{e}_1$  étant alors la tâche la plus prioritaire dite principale. Certaines applications utilisent à profit cette façon de faire [8], [23] et [17].

La tâche secondaire est représentée par un coût secondaire  $h_s$  à minimiser de gradient  $\underline{g}_s = \frac{\partial h_s}{\partial \underline{r}}$ . Si  $L_{\underline{s}}^T$  est de rang plein, Samson a montré dans [21] et [20] qu'une fonction de tâche réalisant  $\underline{e}_1 = 0$  sous la contrainte de minimisation de  $h_s$  s'écrivait :

$$\underline{e} = L_{\underline{s}}^{T+} (\underline{s} - \underline{s}^*) + \alpha \left( \mathbb{I}_n - L_{\underline{s}}^{T+} L_{\underline{s}}^T \right) \underline{g}_s^T \quad (3)$$

où :

- $\alpha$  est un scalaire positif ;
- $L_{\underline{s}}^{T+}$  est la pseudo-inverse de  $L_{\underline{s}}^T$  ;
- $\mathbb{I}_n - L_{\underline{s}}^{T+} L_{\underline{s}}^T$  est un opérateur de projection dans le noyau de  $L_{\underline{s}}^T$ .

On peut alors, dans le cas d'un objet fixe par rapport à la caméra, obtenir le torseur cinématique  $T_c$  par :

$$T_c = -\lambda \underline{e} \quad (4)$$

Cette relation assure un comportement exponentiel découplé de l'erreur de tâche ( $\dot{\underline{e}} = -\lambda \underline{e}$ ) [4].

Ces principales relations étant établies, on se propose dans le chapitre suivant de poser le problème dans le cas d'un objet simple mais pour une tâche complexe.

### 3 Étude d'un objet simple: le segment

#### 3.1 Spécification de la tâche

On se propose d'étudier le positionnement d'une caméra par rapport à un objet filiforme de longueur  $L$  à l'aide d'asservissements visuels. La dimension de cet objet est supposée

inconnue tout comme sa position dans l'espace. Il est caractérisé par un segment d'extrémités  $[m_1, m_2]$  tels que  $\underline{Om}_1 = (x_1, y_1, z_1)^T$  et  $\underline{Om}_2 = (x_2, y_2, z_2)^T$ , coordonnées exprimées dans le repère de la caméra. Par convention ce repère est tel que son centre  $O$  et l'axe des  $z$  coïncident respectivement avec le centre optique et avec l'axe optique de la caméra (cf. figure 1).

Cet objet se projette sur le plan image par transformation perspective respectivement en  $M_1$  et  $M_2$  tel que  $\underline{OM}_1 = (X_1, Y_1, f)^T$  et  $\underline{OM}_2 = (X_2, Y_2, f)^T$  suivant :

$$\underline{OM} = \frac{f}{z} \underline{Om} \quad (5)$$

avec  $f$  la distance focale que l'on supposera, sans perte de généralité, égale à l'unité par la suite.

L'image obtenue est caractérisée par le segment  $[M_1, M_2]$  de longueur  $l$ . On dira que la tâche est réalisée quand l'axe optique de la caméra est orthogonal à  $[m_1, m_2]$  et l'image centrée par rapport à l'axe  $y$ . On aura alors  $z_1 = z_2$  et la caméra sera en face de l'objet considéré.

Bien que l'objet soit très simple, réaliser la tâche de positionnement choisie n'est pas trivial. En effet, supposons la dimension  $L$  de l'objet connue et l'image centrée. On peut écrire :

$$\begin{cases} Y_1 &= \frac{l}{2} \\ Y_2 &= -\frac{l}{2} \end{cases} \quad (6)$$

la consigne visuelle  $l$  s'obtient en considérant la caméra en face de l'objet :

$$l = \frac{L}{z^*} \quad (7)$$

soit :

$$\begin{cases} Y_1^* &= \frac{1}{2} \frac{L}{z^*} \\ Y_2^* &= -\frac{1}{2} \frac{L}{z^*} \end{cases} \quad (8)$$

on en déduit les composantes en  $y$  dans le repère de la caméra :

$$\begin{cases} y_1 &= \frac{L}{2} \frac{z_1}{z^*} \\ y_2 &= -\frac{L}{2} \frac{z_2}{z^*} \end{cases} \quad (9)$$

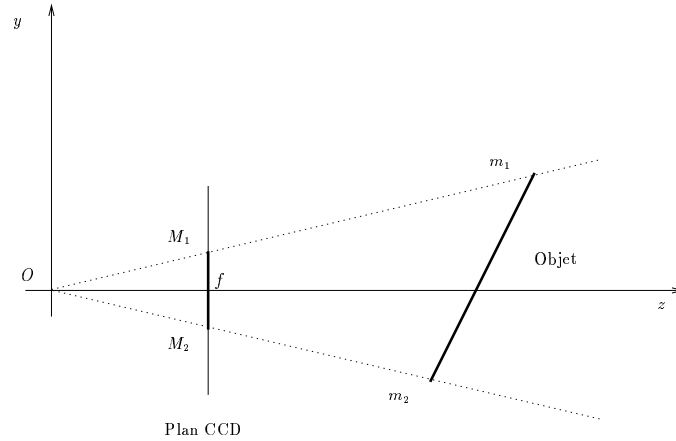


FIG. 1 – Projection de l'objet sur le plan image de la caméra

Par ailleurs, la contrainte de dimension de l'objet impose :

$$(y_1 - y_2)^2 + (z_1 - z_2)^2 = L^2 \quad (10)$$

d'où, en utilisant (9) dans (10), il vient :

$$\left(\frac{L}{2z^*}\right)^2 (z_1 + z_2)^2 + (z_1 - z_2)^2 - L^2 = 0 \quad (11)$$

Par conséquent, une infinité de position du segment permet de centrer l'image sans pour autant réaliser la tâche. Toutefois, les équations ci-dessus nous indiquent une façon possible de réaliser la tâche de positionnement : centrer l'image puis, scruter d'autres positions (celles vérifiant (11)) selon une loi de commande restant à définir mais maintenant l'image centrée. On rencontrera alors la position recherchée assurant  $z_1 = z_2 = z^*$ . Signalons en outre que, curieusement, la connaissance éventuelle de la dimension de l'objet n'apporte rien.

La bonne réalisation de la tâche se décompose donc en deux parties :

- centrer l'image ; il s'agit en fait de choisir judicieusement les informations visuelles.
- rendre égaux  $z_1$  et  $z_2$ . Là, la démarche est plus délicate. *A priori* deux voies sont possibles : (i) déterminer une fonction de tâche secondaire qui s'annule en  $z_1 = z_2$ , (ii) détecter directement cette égalité grâce à des informations extraites de l'image. Dans les deux cas de figure, les mouvements devront appartenir au noyau de  $L_{\frac{z}{2}}^T$  afin de maintenir l'image centrée.

Nous allons détailler dans les paragraphes suivants les différentes voies possibles.

## 3.2 Centrage de l'image

### 3.2.1 Utilisation de quatre informations visuelles

L'approche la plus immédiate est de choisir le vecteur  $\underline{s}$  tel que :

$$\underline{s} = (X_1, X_2, Y_1, Y_2)^T \quad (12)$$

et de chercher à atteindre le motif  $\underline{s}^*$  donné par :

$$\underline{s}^* = (0, 0, Y, -Y)^T \quad (13)$$

avec  $Y = l_c/2$  où  $l_c$  est la taille désirée du segment  $[M_1M_2]$  dans l'image.

En se rappelant que la matrice d'interaction dans le cas d'un point de coordonnées  $(X, Y)$  s'écrit ([12], [13] et [4]) :

$$L_{\underline{s}}^T = \begin{pmatrix} -1/z & 0 & X/z & XY & -1 - X^2 & Y \\ 0 & -1/z & Y/z & 1 + Y^2 & -XY & -X \end{pmatrix} \quad (14)$$

on obtient :

$$L_{\underline{s}=\underline{s}^*}^T = \begin{pmatrix} -1/z_1 & 0 & 0 & 0 & -1 & Y \\ -1/z_2 & 0 & 0 & 0 & -1 & -Y \\ 0 & -1/z_1 & Y/z_1 & 1 + Y^2 & 0 & 0 \\ 0 & -1/z_2 & -Y/z_2 & 1 + Y^2 & 0 & 0 \end{pmatrix} \quad (15)$$

Les profondeurs  $z_1$  et  $z_2$  étant inconnues, on fait l'approximation  $z_1 = z_2 = z$  où  $z$  est une approximation de  $z^*$ , profondeur atteinte une fois la tâche réalisée.

On utilise ensuite (3) et (4) avec  $\alpha = 0$  pour réaliser le centrage souhaité.

### 3.2.2 Utilisation de trois informations visuelles

Une autre façon d'opérer est de choisir :

$$\underline{s} = (X_1, X_2, Y_1 + Y_2)^T \quad (16)$$

et :

$$\underline{s}^* = (0, 0, 0)^T \quad (17)$$

En utilisant la propriété qui lie un paramétrage  $\underline{P}'$  à un autre paramétrage  $\underline{P}$  relatif à la même primitive géométrique [4], à savoir :

$$L^T(\underline{P}') = \frac{\partial \underline{P}'}{\partial \underline{P}} L^T(\underline{P}), \quad (18)$$

on obtient la matrice d'interaction souhaitée :

$$L^T \left| \begin{array}{l} x_i=0 \\ z_i=z \end{array} \right. = \begin{pmatrix} -\frac{1}{z} & 0 & 0 & 0 & -1 & Y_1 \\ -\frac{1}{z} & 0 & 0 & 0 & -1 & Y_2 \\ 0 & -\frac{2}{z} & \frac{Y_1 + Y_2}{z} & 2 + Y_1^2 + Y_2^2 & 0 & 0 \end{pmatrix} \quad (19)$$

Si l'on veut comme en 3.2.1 imposer une taille  $l_c$  d'image, on introduit un coût secondaire  $h_s$  à minimiser :

$$h_s = \frac{1}{2} (l - l_c)^2 \quad (20)$$

La loi de commande décrite par (3) nécessite le calcul de  $\underline{g}_s$ . On a :

$$\underline{g}_s \left| \begin{array}{l} x_i=0 \\ z_i=z \end{array} \right. = (l - l_c) \left( L_{Y_1} \left| \begin{array}{l} x_1=0 \\ z_1=z \end{array} \right. - L_{Y_2} \left| \begin{array}{l} x_2=0 \\ z_2=z \end{array} \right. \right) \quad (21)$$

soit :

$$L_{Y_i} \left| \begin{array}{l} x_i=0 \\ z_i=z \end{array} \right. = (0 \quad -1/z \quad Y_i/z \quad 1 + Y_i^2 \quad 0 \quad 0)^T \quad (22)$$

et, simplifications faites, il vient :

$$\underline{g}_s \left| \begin{array}{l} x_i=0 \\ z_i=z \end{array} \right. = l(l - l_c) (0 \quad 0 \quad 1/z \quad Y_1 + Y_2 \quad 0 \quad 0)^T \quad (23)$$

Le centrage et la tâche secondaire sont ensuite réalisés grâce à (3) et (4).

Grâce aux méthodes de centrage que nous venons de détailler, nous allons pouvoir, dans les deux cas, élaborer les lois de commande permettant la bonne réalisation de la tâche. À ce stade il s'agit maintenant d'obtenir  $z_1 = z_2$ .

### 3.3 Détermination d'une fonction de tâche secondaire

La première étape étant réalisée, on va chercher une fonction de tâche secondaire  $e_2$  qui permet  $z_1 = z_2$ . Pour cela, on se place dans les conditions du paragraphe 3.2.1 où on utilise

quatre informations visuelles. Dans ce cas de figure, les seuls mouvements qui maintiennent l'image centrée et constante sont tels qu'ils laissent les informations visuelles invariantes : ce sont donc des mouvements qui appartiennent au noyau de la matrice d'interaction  $L_{|X_i=0}^T$  (l'image étant centrée on a  $\underline{s} = \underline{s}^*$ ). Cependant, les profondeurs étant inconnues on doit se contenter du noyau de  $L_{|X_i=0, z_i=z}^T$ . De plus, parmi ces mouvements, seuls nous intéressent ceux qui génèrent un mouvement dans le plan ( $yOz$ ). Ils sont de la forme :

$$T_s = (0, z\omega_x (1 + Y^2), 0, \omega_x, 0, 0)^T \quad (24)$$

la loi de commande est donc du type :

$$T_c = -\lambda L_{\substack{X_i=0 \\ z_i=z}}^{T+} (\underline{s} - \underline{s}^*) + T_s \quad (25)$$

où dans (24), et donc dans (25),  $\omega_x$  est à déterminer en fonction d'une tâche secondaire  $e_2$ .

Connaissant  $T_c$ , on en déduit la loi d'évolution de  $\underline{s}$  :

$$\underline{s}|_{k+1} = \underline{s}|_k + \Delta t L^T T_c \quad (26)$$

où  $\Delta t$  est la période d'acquisition des images.

On impose à la fonction de tâche secondaire  $e_2$  recherchée d'être fonction des informations visuelles disponibles et de s'écrire de la façon la plus simple possible :

$$e_2(\underline{\epsilon}) = \alpha \epsilon_1 + \beta \epsilon_2 + \gamma \quad (27)$$

dans lequel on a introduit le vecteur  $\underline{\epsilon} = (\epsilon_1, \epsilon_2)$  avec :

$$\begin{cases} \epsilon_1 &= Y_1 - Y \\ \epsilon_2 &= Y_2 + Y \end{cases} \quad (28)$$

On souhaite que cette fonction s'annule en  $z_1 = z_2 = z^*$ , on impose donc la contrainte suivante :

$$e_2|_{k+1, z_1=z_2=z^*} = 0 \quad (29)$$

Grâce à (27) puis (26), on exprime (29) uniquement en fonction de grandeurs connues à l'instant  $k$ . On en déduit alors  $\gamma$  en fonction de  $\alpha$  et de  $\beta$ . En effet, on peut montrer que  $\gamma$  se met sous la forme :

$$\gamma(\alpha, \beta) = \frac{b_1(\alpha, \beta)z^* + b_2(\alpha, \beta)}{b_3(\alpha, \beta)z^*} \quad (30)$$

où  $b_1$ ,  $b_2$  et  $b_3$  ont une forme peu sympathique qu'il est inutile de présenter ici.

Pour éviter que  $\gamma(\alpha, \beta)$  dépende de  $z^*$  qui est inconnu, on impose  $b_2 = 0$ . Cette nouvelle contrainte fixe le choix du coefficient  $\beta$ , fonction de grandeurs connues à l'instant  $k$  et de  $\alpha$  :

$$\beta(\alpha) = \alpha \frac{Y_1^2 - Y_1 Y_2 + \beta_1 Y_1 + \beta_2 Y_2 + \beta_3}{Y_2^2 - Y_1 Y_2 - \beta_2 Y_1 - \beta_1 Y_2 - \beta_3} \quad (31)$$

où on a posé :

$$\begin{cases} \beta_1 &= -Y \frac{1 + 2z^2 (1 + Y^2)^2}{1 + z^2 (1 + Y^2)^2} \\ \beta_2 &= \frac{Y}{1 + z^2 (1 + Y^2)^2} \\ \beta_3 &= 2Y (1 + Y^2) \frac{\omega_x}{\lambda} \end{cases} \quad (32)$$

$\beta(\alpha)$  étant connu, on peut exprimer maintenant  $\gamma(\alpha)$  grâce à (30) :

$$\gamma(\alpha) = \alpha \gamma_0 \frac{Y_1 + Y_2 + \gamma_1}{Y_2^2 - Y_1 Y_2 - \beta_2 Y_1 - \beta_1 Y_2 - \beta_3} P(Y_1, Y_2) \quad (33)$$

avec :

$$\begin{cases} \gamma_0 &= -\frac{\lambda \Delta t z^2 (1 + Y^2)}{2 (1 + z^2 (1 + Y^2)^2)} \\ \gamma_1 &= -\frac{2\omega_x (1 + z^2 (1 + Y^2)^2)}{\lambda z^2 (1 + Y^2)} \end{cases} \quad (34)$$

et :

$$\begin{aligned} P(Y_1, Y_2) &= Y_1^3 Y_2 + Y_2^3 Y_1 - 2Y_1^2 Y_2^2 + \beta_2 (Y_1^3 - Y_2^3) \\ &+ \beta_1 (Y_1^2 Y_2 - Y_2^2 Y_1) + \gamma_2 Y_1^2 - \beta_3 Y_2^2 + 2Y_1 Y_2 + \gamma_3 (Y_1 - Y_2) + \gamma_4 \end{aligned} \quad (35)$$

dans lequel on a posé :

$$\begin{cases} \gamma_2 &= 2Y (1 + Y^2) \frac{\omega_x}{\lambda} - 1 \\ \gamma_3 &= 2Y \frac{1 + Y^2 + \lambda \Delta t}{\lambda \Delta t} \\ \gamma_4 &= -4Y^2 \frac{1 + Y^2}{\lambda \Delta t} \end{cases} \quad (36)$$

L'utilisation de (31) et (33) dans (27) conduit à :

$$e_2|_{k+1} = \alpha e_2(0) \frac{z_1 - z_2}{z_1 z_2} (Y_1^2 - Y_1 Y_2 + \beta_1 Y_1 + \beta_2 Y_2 + \beta_3) \quad (37)$$



avec :

$$e_2(0) = \frac{\lambda \Delta t z}{2Y} \quad (38)$$

En imposant :

$$\alpha = \frac{1}{e_2(0) (Y_1^2 - Y_1 Y_2 + \beta_1 Y_1 + \beta_2 Y_2 + \beta_3)} \quad (39)$$

on obtient finalement :

$$e_2|_{k+1} = \frac{z_1 - z_2}{z_1 z_2} \quad (40)$$

Cette fonction convient parfaitement : elle ne s'annule qu'en  $z_1 = z_2$ , elle est proportionnelle à l'erreur de positionnement  $z_1 - z_2$  et surtout toutes les grandeurs sont connues au moment du calcul de  $e_2$ .

On propose donc la loi de commande définie par (25) dans laquelle on impose :

$$\omega_x|_k = K_p e_2|_k \quad (41)$$

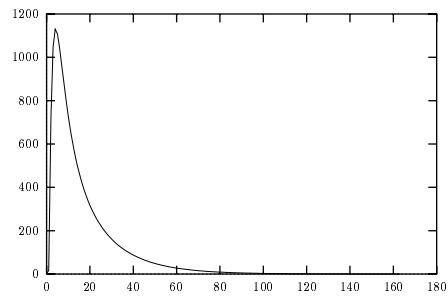
où  $K_p$  est un gain proportionnel, soit en fonction des coefficients  $\alpha$ ,  $\beta$  et  $\gamma$  :

$$\omega_x|_k = K_p (\alpha|_{k-1} \epsilon_1|_k + \beta|_{k-1} \epsilon_2|_k + \gamma|_{k-1}) \quad (42)$$

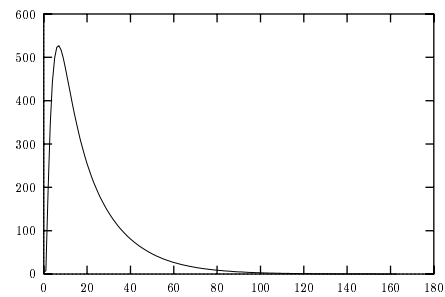
Le comportement de cette loi de commande fait l'objet des paragraphes suivants. Cependant, on s'aperçoit d'ores et déjà que pour l'initialiser il faudra imposer pour la première itération une valeur  $\omega_x$  non nulle. De plus, conformément à (40), on pourra choisir indifféremment son signe puisque la différence des profondeurs est connue par  $e_2$  dès la deuxième itération. Pratiquement, on a choisi une valeur de  $\omega_x$  suffisamment importante pour assurer une variation significative de  $\underline{z}$ .

### 3.3.1 Résultats de simulations non bruitées

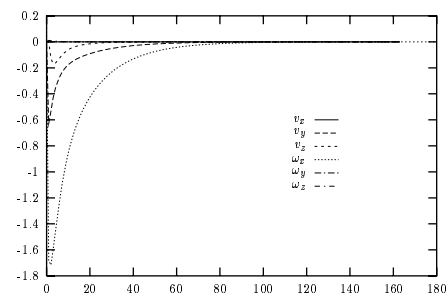
Les figures 2(a), 2(b), 2(c) et 2(d) représentent, une fois le centrage obtenu, les évolutions respectives de  $\epsilon_1|_k$ ,  $\epsilon_2|_k$ , des composantes du torseur cinématique à l'instant  $k$  et de  $(z_1 - z_2)|_k$ . Elles nous permettent de constater que cette loi de commande permet de réaliser correctement la tâche de positionnement souhaitée et cela sans information sur la situation ou sur la taille de l'objet. Une erreur de position initiale  $z_1 - z_2$  de 100 mm (après centrage) est réduite à zéro en une centaine d'itérations.



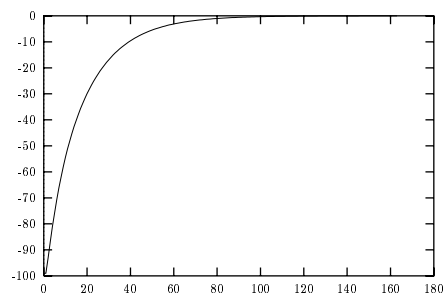
(a) Erreur dans le capteur relative à l'information visuelle  $Y_1$  ( $\mu\text{m}$ ) en fonction du nombre d'itérations



(b) Erreur dans le capteur relative à l'information visuelle  $Y_2$  ( $\mu\text{m}$ ) en fonction du nombre d'itérations



(c) Composantes du torseur cinématique (m/s ou rad/s) en fonction du nombre d'itérations



(d) Erreur de positionnement (mm) en fonction du nombre d'itérations

FIG. 2 — Utilisation d'une fonction de tâche secondaire — Informations visuelles non bruitées —

Par ailleurs, on peut remarquer qu'en fait le mouvement réalisé n'appartient pas exactement au noyau de la matrice d'interaction. En effet, les signaux d'erreur  $\epsilon_1$  et  $\epsilon_2$  augmentent fortement au début du mouvement, il s'en suit bien évidemment des commandes de compensation, en particulier sur  $v_z$ , pour maintenir l'image invariante. Par contre, quand  $z_1 \rightarrow z_2$  l'approximation  $z_1 = z_2 = z$  devient valide et les signaux  $\epsilon_1$  et  $\epsilon_2$  tendent alors vers zéro.

### 3.3.2 Résultats de simulations bruitées

Pour tester la robustesse de l'algorithme précédent, on se propose de bruite les informations visuelles comme suit :

$$\begin{cases} Y_1 &= Y_1 + b_1 \\ Y_2 &= Y_2 + b_2 \end{cases} \quad (43)$$

où  $b_i$  est un bruit uniforme dans l'intervalle  $[-p_y/10, p_y/10]$ ,  $p_y$  étant la hauteur des pixels.

Comme le montrent les courbes 3(a), 3(b), 3(c) et 3(d), la loi de commande est sensible au bruit même si son amplitude est faible. Une analyse du phénomène s'impose.

Pour ce faire, on va tout d'abord se placer dans un cas simple — celui où la consigne est atteinte et le bruit nul — et on va exposer, dans ce cas, une façon plus rapide d'obtenir la fonction  $e_2$ .

La consigne étant réalisée, seul le mouvement appartenant à  $\text{Ker } L_{|\underline{s}=\underline{s}^*, z_1=z_2=z}^T$  existe, le torseur cinématique s'écrit :

$$T_c = (0, z(1+Y^2)\omega_x|_k, 0, \omega_x|_k, 0, 0)^T \quad (44)$$

Connaissant  $T_c$ , on en déduit la loi d'évolution de  $\underline{s}$  :

$$\underline{s}|_{k+1} = \underline{s}^* + \Delta t L_{|\underline{s}=\underline{s}^*}^T T_c \quad (45)$$

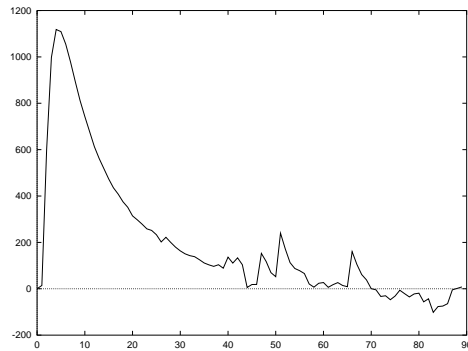
par suite, il vient :

$$Y_1|_{k+1} - Y = \frac{(1+Y^2)\Delta t \omega_x|_k (z - z_1)}{z_1} \quad (46)$$

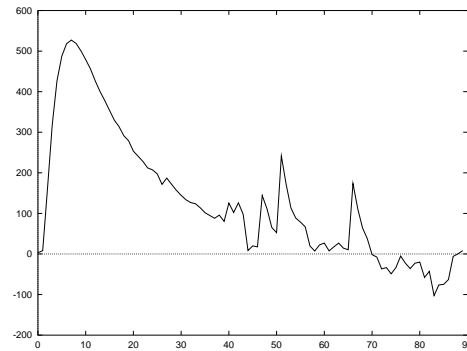
$$Y_2|_{k+1} + Y = \frac{(1+Y^2)\Delta t \omega_x|_k (z - z_2)}{z_2} \quad (47)$$

La seule façon d'obtenir un terme proportionnel à l'erreur de positionnement  $z_1 - z_2$  est de réaliser  $\alpha ((Y_1|_{k+1} - Y) - (Y_2|_{k+1} + Y))$  soit :

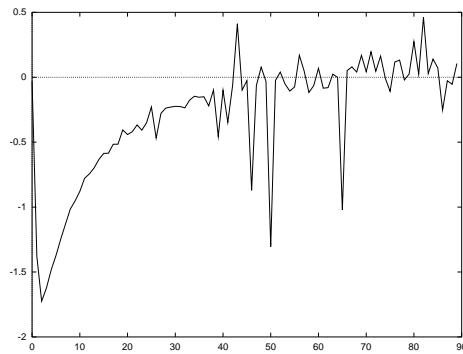
$$\alpha ((Y_1|_{k+1} - Y) - (Y_2|_{k+1} + Y)) = \alpha \frac{z(1+Y^2)\Delta t \omega_x|_k (z_1 - z_2)}{z_1 z_2} \quad (48)$$



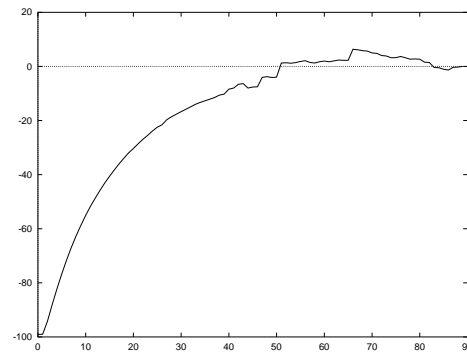
(a) Erreur dans le capteur relative à l'information visuelle  $Y_1$  ( $\mu\text{m}$ ) en fonction du nombre d'itérations



(b) Erreur dans le capteur relative à l'information visuelle  $Y_2$  ( $\mu\text{m}$ ) en fonction du nombre d'itérations



(c) Composante  $\omega_x$  du torseur cinématique (m/s ou rad/s) en fonction du nombre d'itérations



(d) Erreur de positionnement (mm) en fonction du nombre d'itérations

FIG. 3 — Utilisation d'une fonction de tâche secondaire — Informations visuelles bruitées —

On impose alors :

$$\alpha = \frac{1}{z(1+Y^2)\Delta t \omega_x|_k} \quad (49)$$

et finalement on obtient :

$$e_2|_{k+1} = \frac{z_1 - z_2}{z_1 z_2} \Big|_k \quad (50)$$

soit :

$$\omega_x|_{k+1} = \frac{K_p}{z(1+Y^2)\Delta t \omega_x|_k} ((Y_1|_{k+1} - Y) - (Y_2|_{k+1} + Y)) \quad (51)$$

Considérons maintenant le cas où les informations visuelles sont bruitées conformément à (43). Dans ces conditions (48) devient :

$$e_2|_{k+1} = \alpha \frac{z(1+Y^2)\Delta t \omega_x|_k (z_1 - z_2)}{z_1 z_2} - \alpha(b_2 - b_1) \quad (52)$$

et si l'on conserve pour  $\alpha$  l'expression donnée en (49) :

$$e_2|_{k+1} = \frac{z_1 - z_2}{z_1 z_2} \Big|_k - \frac{b_2|_{k+1} - b_1|_{k+1}}{z(1+Y^2)\Delta t \omega_x|_k} \quad (53)$$

Cette expression se décompose en deux termes, un premier non bruité et un second bruité. À la vue de ce second terme, nous pouvons tirer les conclusions suivantes :

- au début du mouvement appartenant à  $\text{Ker } L_{|\underline{x}=\underline{x}^*, z_1=z_2=z}^T$ , l'erreur de positionnement  $z_1 - z_2$  est importante et, par suite, la commande  $\omega_x|_k$  l'est aussi. Ce second terme est donc négligeable devant le premier qui lui, au contraire, est grand ;
- quand on s'approche de  $z_1 = z_2$ , la commande  $\omega_x|_k$  tend vers zéro et cette fois c'est le premier terme qui devient négligeable devant le second. Le bruit est en quelque sorte amplifié. La fonction  $e_2|_{k+1}$  et donc la commande  $\omega_x|_{k+1}$  est essentiellement du bruit.

Le problème vient du fait que  $\lim_{z_1 \rightarrow z_2} \alpha = \infty$ . Mais, si on imposait une valeur bornée de  $\alpha$ , la commande  $\omega_x|_k$  étant décroissante, d'après (48) le signal d'erreur  $e_2|_{k+1}$  décroîtrait alors rapidement vers zéro  $\omega_x|_k$  n'étant plus compensée par  $\alpha$ .

Par conséquent, cette façon de faire ne peut conduire à un positionnement précis, une autre approche est indispensable. Elle est décrite dans le prochain paragraphe.

### 3.4 Scrutation d'informations visuelles

La façon de centrer l'image à l'aide de quatre informations visuelles limite en fait notre marge de manœuvre puisque non seulement les informations visuelles restent invariantes au cours du mouvement mais l'image aussi.

On se place alors dans les conditions du paragraphe 3.2.2 où trois informations visuelles sont utilisées.

Une fois l'image centrée, on applique une loi de commande du type :

$$T_c = -\lambda L \left. \begin{matrix} T^+ \\ x_i=0 \\ z_i=z \end{matrix} \right| (\underline{s} - \underline{s}^*) + T_s \quad (54)$$

avec :

$$T_s = \left( 0 \quad z\omega_x \frac{2 + Y_1^2 + Y_2^2}{2} \quad 0 \quad \omega_x \quad 0 \quad 0 \right)^T \quad (55)$$

où  $\omega_x$  est à déterminer.

Si  $\underline{s}$  est proche de  $\underline{s}^*$ , seul  $T_s$  génère un mouvement. Il s'apparente à une pseudo-rotation d'axe  $x$ . Intuitivement, on imagine assez facilement que la taille du segment dans l'image  $l = Y_1 - Y_2$  peut fournir des informations. On pense en particulier qu'elle passe par un maximum une fois la caméra en face de l'objet, cette hypothèse a été validée grâce à des simulations.

Afin d'élaborer une commande basée sur la taille du segment, on va s'efforcer dans le paragraphe suivant d'en trouver une modélisation.

#### 3.4.1 Modélisation de la taille de l'image à l'aide d'informations dynamiques

Soit  $\underline{x}(t)$  les coordonnées du point observé par la caméra à l'instant  $t$  dans son repère. Elle est soumise pendant  $\Delta t$  à une vitesse de rotation  $\underline{V}$  et à une vitesse de translation  $\underline{\Omega}$  constantes telles que  $T_c = (\underline{V}^T, \underline{\Omega}^T)^T$ . On a alors la relation suivante :

$$\dot{\underline{x}}(t) = -\underline{V} - \underline{\Omega} \wedge \underline{x}(t) \quad (56)$$

où :

$$\underline{V} = (v_x, v_y, v_z)^T \quad (57)$$

$$\underline{\Omega} = (\omega_x, \omega_y, \omega_z)^T \quad (58)$$

On peut réécrire (56) de la façon suivante :

$$\dot{\underline{x}}(t) + \underline{\Omega} \underline{x}(t) = -\underline{V} \quad (59)$$

avec :

$$\Omega = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (60)$$

Appliquons la transformée de Laplace à (59), il vient :

$$\underline{x}(p) = (p\mathbb{I}_3 + \Omega)^{-1} \underline{\nu}(p) \quad (61)$$

où on a posé :

$$\underline{\nu}(p) = \underline{x}(t_k) - \frac{V(t_k)}{p} \quad (62)$$

et  $t_k = k\Delta t$ .

Simplifications faites, il vient :

$$\underline{x}(p) = \frac{1}{p(p^2 + \omega^2)} \begin{pmatrix} p^2 + \omega_x^2 & \omega_z p + \omega_x \omega_y & -\omega_y p + \omega_x \omega_z \\ -\omega_z p + \omega_x \omega_y & p^2 + \omega_y^2 & \omega_x p + \omega_y \omega_z \\ \omega_y p + \omega_x \omega_z & -\omega_x p + \omega_y \omega_z & p^2 + \omega_z^2 \end{pmatrix} \underline{\nu}(p) \quad (63)$$

où on a posé  $\omega = \sqrt{\Omega_c^2}$ .

On restreint la résolution de (56) au cas où seuls  $\omega_x$  et  $v_y$  sont non nuls. On obtient alors par transformation inverse :

$$\begin{cases} x(t) &= x(t_k) \\ y(t) &= y_1|_k \cos(\Delta\theta_k) + z_1|_k \sin(\Delta\theta_k) - v_y(k) \frac{\sin(\Delta\theta_k)}{\omega_x} \\ z(t) &= -y_1|_k \sin(\Delta\theta_k) + z_1|_k \cos(\Delta\theta_k) + v_y(k) \frac{1 - \cos(\Delta\theta_k)}{\omega_x} \end{cases} \quad (64)$$

où on a posé :

$$\Delta\theta_k = \omega_x(t - t_k) \quad (65)$$

On en déduit la position des points  $m_1 = (x_1, y_1, z_1)^T$  et  $m_2 = (x_2, y_2, z_2)^T$ , extrémités du segment observé, et par suite, par transformation perspective la taille de l'image :  $l = Y_1 - Y_2 = \frac{y_1}{z_1} - \frac{y_2}{z_2}$ .

On simplifie cette relation d'une part en supposant que  $t = t_k$  à l'ordre un et d'autre part en imposant la relation suivante entre  $v_y$  et  $\omega_x$  pour réaliser un mouvement appartenant au noyau de  $L_{|X_i=0}^T$  :

$$v_y(k) = 2\omega_x|_k \frac{z_1|_k z_2|_k (1 + a_k^2)}{z_1|_k + z_2|_k} \quad (66)$$

avec  $a_k = \frac{y_1|_k}{z_1|_k} = -\frac{y_2|_k}{z_2|_k}$  (si l'on suppose que la tâche principale maintienne l'image correctement centrée).

Après simplification il vient :

$$l(t) = Y_1 - Y_2 = l(t_k) - 2\omega_x|_k \frac{(1 + a_k^2) (z_1|_k - z_2|_k) (t_k - t)}{z_1|_k + z_2|_k} \quad (67)$$

Pour  $t_k \leq t < t_k + \Delta t$ ,  $\omega_x$  est constant d'où :

$$\frac{\partial \theta_x}{\partial t} = \omega_x|_k \Rightarrow \theta_x - \theta_x|_k = \omega_x|_k (t - t_k) \quad (68)$$

et finalement on en déduit l'expression cherchée sur l'intervalle  $[k\Delta t, (k+1)\Delta t[$  :

$$l(\theta_x) = l(\theta_x|_k) - 2 \frac{(1 + a_k^2) (z_1|_k - z_2|_k) (\theta_x|_k - \theta_x)}{z_1|_k + z_2|_k} \quad (69)$$

Malheureusement, cette expression est en pratique inutilisable  $z_1|_k$  et  $z_2|_k$  étant inconnus.

Pour obtenir néanmoins une représentation analytique de  $l = f(\theta_x)$  on va raisonner, dans le paragraphe suivant, à partir d'informations statiques.

### 3.4.2 Modélisation de la taille de l'image à l'aide d'informations statiques

Soient  $m_1$  et  $m_2$  les extrémités du segment considéré et  $m_0$  un point de l'espace tels que  $m_1 = (y^*, z^*)^T$ ,  $m_2 = (-y^*, z^*)^T$  et  $m_0 = (0, z_0)^T$  coordonnées exprimées dans le repère de la caméra en position souhaitée. On suppose que le mouvement réalisé lors de la maximisation de la taille de l'image résulte d'une rotation d'axe  $x$  passant par  $m_0$  et d'angle  $\theta_x$  puis d'une translation  $\mathcal{T}$  (cf. figure 4) . Les coordonnées des points  $m_1$  et  $m_2$  s'expriment alors dans le repère de la caméra par :

$$\begin{cases} m_1' &= \mathcal{R} (m_1 - m_0) + \mathcal{T} \\ m_2' &= \mathcal{R} (m_2 - m_0) + \mathcal{T} \end{cases} \quad (70)$$

où :

$$\mathcal{R} = \begin{pmatrix} \cos \theta_x & \sin \theta_x \\ -\sin \theta_x & \cos \theta_x \end{pmatrix}$$

et :

$$\mathcal{T} = \begin{pmatrix} 0 \\ z_0 \end{pmatrix}$$



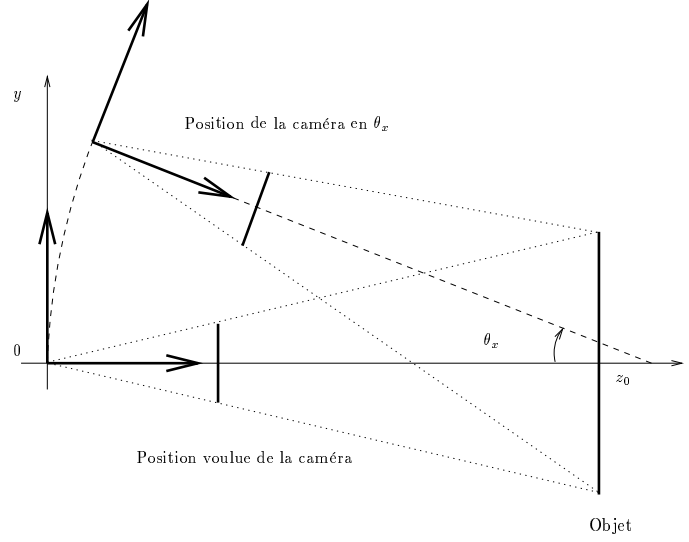


FIG. 4 – Modélisation de la taille de l'objet dans l'image en fonction de  $\theta_x$

On en déduit par transformation perspective les coordonnées des points  $M_1 = (X_1, Y_1)^T$  et  $M_2 = (X_2, Y_2)^T$  images des points  $m_1$  et  $m_2$  :

$$\begin{cases} X1 &= 0 \\ Y1 &= -\frac{y^* \cos \theta_x + (z^* - z_0) \sin \theta_x}{y^* \sin \theta_x - (z^* - z_0) \cos \theta_x - z_0} \\ X2 &= 0 \\ Y2 &= -\frac{y^* \cos \theta_x - (z^* - z_0) \sin \theta_x}{y^* \sin \theta_x + (z^* - z_0) \cos \theta_x + z_0} \end{cases} \quad (71)$$

On cherche à maintenir par le biais de  $z_0$  une image centrée, c'est-à-dire à annuler  $Y_1 + Y_2$  et en fait, par analogie avec le paragraphe précédent, à réaliser un mouvement appartenant au noyau de la matrice d'interaction  $L_{X_i=0}^T$ . Deux valeurs de  $z_0$  conviennent :

$$\begin{cases} z_{01} &= \frac{1}{2} \frac{2z^* \cos \theta_x - z^* + \sqrt{z^{*2} - 4y^{*2}(\cos \theta_x - 1) \cos \theta_x}}{\cos \theta_x - 1} \\ z_{02} &= \frac{1}{2} \frac{2z^* \cos \theta_x - z^* - \sqrt{z^{*2} - 4y^{*2}(\cos \theta_x - 1) \cos \theta_x}}{\cos \theta_x - 1} \end{cases} \quad (72)$$

On lève cette ambiguïté en se plaçant en  $\theta_x = \pi/2$  :

$$\begin{cases} z_{01}(\pi/2) & = 0 \\ z_{02}(\pi/2) & = z^* \end{cases} \quad (73)$$

on élimine ainsi le cas irréaliste  $z_0 = z_{01}$ .

Posons maintenant  $v = \frac{y^*}{z^*}$  et supposons  $|\theta_x| \leq \frac{\pi}{2}$ , il vient après simplification :

$$l_m = 2v \frac{2 \cos \theta_x + 1 - \sqrt{1 - 4v^2(\cos \theta_x - 1) \cos \theta_x}}{2v^2(\cos \theta_x - 1) + 1 + \sqrt{1 - 4v^2(\cos \theta_x - 1) \cos \theta_x}} \quad (74)$$

On peut vérifier, comme on s'y attend, que  $l_m(\theta_x)$  est une fonction  $2\pi$  périodique, paire, qu'elle s'annule en  $\theta_x = \pi/2$  et qu'en  $\theta_x = 0$  elle vaut bien  $2\frac{y^*}{z^*}$  qui est en fait la taille de l'image quand la caméra se situe en face de l'objet.

De plus, pour  $|\theta_x| \leq \frac{\pi}{2}$  on peut montrer qu'elle admet un unique maximum en  $\theta_x = 0$ .

Cette fonction, relativement complexe, peut s'écrire plus simplement dans le cas où  $v$  est faible par une valeur approchée  $l_a$  de  $l_m$  :

$$l_a = 2v \cos \theta_x \quad (75)$$

La figure 5 représente en simulation l'erreur en pixel entre la taille de l'image  $l$  et la taille de l'image modélisée respectivement par (74) puis par (75). Les conditions de simulation étant les suivantes :

- les calculs concernant le torseur d'interaction ont été réalisés pour les vraies valeurs de  $z_i$  mais en  $s = s^*$  ;
- $\omega_x = -0.05$  rad/s ;
- $m_1 = (0.1, 0.7)^T$  et  $m_2 = (-0.1, 0.7)^T$  (en mètre).

Pour se donner une idée des ordres de grandeur, on a représenté sur la figure 6 les variations de  $l(\theta_x)$ ,  $l_m(\theta_x)$  et  $l_a(\theta_x)$ .

Comme on peut le vérifier l'hypothèse d'un mouvement de pseudo rotation de rayon  $z_0 = z_{02}$  variable et de centre  $m_0$  est assez réaliste puisque l'erreur relative maximum est de l'ordre de 0.6 %.

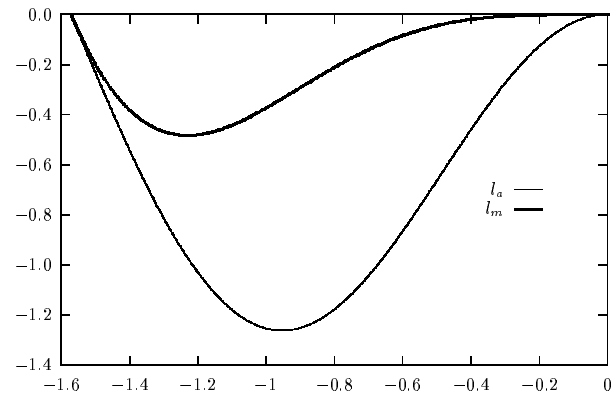


FIG. 5 – Erreur de modélisation :  $l - l_m$  et  $l - l_a$  (pixel) en fonction de  $\theta_x$  (rad)

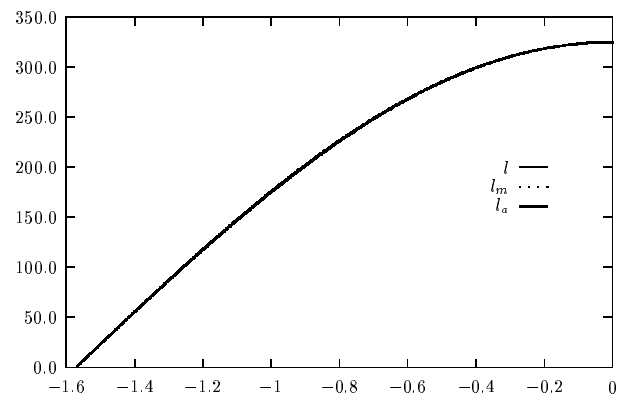


FIG. 6 – Variations de  $l$ ,  $l_m$  et  $l_a$  (pixel) en fonction de  $\theta_x$  (rad)

Les résultats obtenus sont donc tout à fait satisfaisants, cependant que ce soit  $l_m$  ou  $l_a$  ces fonctions dépendent, par le biais de  $v$ , de la situation désirée et par conséquent inconnue de l'objet. Une phase d'identification ou d'apprentissage en ligne est donc nécessaire. La mise en œuvre d'une telle identification de  $l_m$  peut s'avérer délicate puisque le modèle obtenu est non linéaire. De plus, la fonction à utiliser est en pratique  $l_m(\theta_x - \theta_{xc})$  où  $\theta_{xc}$  est la valeur angulaire où  $l_m$  admet un maximum. C'est en fait cette valeur qu'il est important de connaître mais qui est totalement inconnue. Par conséquent, en l'absence d'estimé initial correct, l'utilisation d'un algorithme d'identification non linéaire, du type Levenberg-Marquardt par exemple, conduirait inévitablement à l'échec. Aussi, dans un but de simplification, on va examiner le cas où on fait l'hypothèse que le point  $m_0$  est fixe. Le passage des points  $m_i$  aux points  $m_i'$  est toujours réalisé par (70) mais avec  $m_0 = (0, z^*)^T$  et  $\mathcal{T} = (t_y, z^*)^T$ .

Les points  $M_1$  et  $M_2$  deviennent :

$$\begin{cases} X1 &= 0 \\ Y1 &= -\frac{y^* \cos \theta_x - t_y}{y^* \sin \theta_x - z^*} \\ X2 &= 0 \\ Y2 &= -\frac{y^* \cos \theta_x + t_y}{y^* \sin \theta_x + z^*} \end{cases} \quad (76)$$

On cherche la valeur de  $t_y$  qui annule  $Y_1 + Y_2$  par analogie avec un mouvement du noyau de  $L_{X_i=0}^T$ . Il vient :

$$t_y = \frac{y^{*2} \cos \theta_x \sin \theta_x}{z^*} \quad (77)$$

que l'on injecte dans  $l_m = Y_1 - Y_2$  qui devient :

$$l_m = 2v \cos \theta_x \quad (78)$$

On retrouve l'expression obtenue précédemment quand on supposait  $v$  faible. En fait, il s'avère que cette expression peut s'utiliser sous d'autres hypothèses moins restrictives que  $v$  faible comme le présupposait déjà la figure 5 puisque la valeur de  $v$  était relativement importante ( $v \approx 0.14$ ) et la modélisation étonnamment correcte.

### 3.4.3 Application à la commande

Au cours du mouvement, dont la forme de la loi de commande est donnée par (54), on réalise une identification en ligne de  $l$  en supposant  $l$  modélisé par  $l_a$  que l'on développe en :

$$l_a = c_1 \cos \theta_x + c_2 \sin \theta_x \quad (79)$$

On obtient ainsi une équation linéaire par rapport aux paramètres  $c_1$  et  $c_2$  que l'on peut identifier par une méthode classique des moindres carrés avec facteur d'oubli  $\gamma$ . On identifie ensuite  $l_a$  à  $l_{max} \cos(\theta_x - \theta_{xc})$  conformément à (75) ou (78) pour obtenir la valeur de  $\theta_{xc}$  recherchée. Par suite, il vient :

$$\theta_{xc} = -\arctan \frac{c_2}{c_1} \quad (80)$$

Puis, pour obtenir une décroissance quasi exponentielle de l'erreur angulaire  $\theta_x - \theta_{xc}$ , on calcule la commande par :

$$\omega_x = K_x(\theta_x - \theta_{xc}) \quad (81)$$

où  $K_x$  est un scalaire positif permettant d'ajuster la vitesse de décroissance de cette erreur.

Il ne reste plus qu'à déterminer la façon de faire pendant le temps où le résultat de l'apprentissage n'est pas suffisamment correct pour utiliser  $\theta_{xc}$ . On propose simplement d'imposer  $\omega_x$  à une valeur  $\omega_0$  constante. Cependant, afin de ne pas conduire à un éventuel changement de signe de  $\omega_x$  une fois les résultats de l'apprentissage corrects, il convient de déterminer le signe à appliquer à  $\omega_0$ .

Ce problème se résout grâce à la phase initiale décrite dans 3.2.2. On suppose  $\alpha$  suffisamment petit pour que l'image soit centrée dans le capteur sans que la consigne  $l_c$  soit atteinte. Par suite, on peut montrer en utilisant (3), (23) puis (4) dans lesquelles on aura préalablement remplacé  $Y_1$  par  $l/2$  et  $Y_2$  par  $-l/2$  que le torseur cinématique devient :

$$\underline{T}_c = \begin{pmatrix} 0 & 0 & \frac{-\alpha\lambda(l-l_c)}{z} & 0 & 0 & 0 \end{pmatrix}^T \quad (82)$$

et en estimant la variation des informations visuelles par :

$$\dot{s} = L_{|X_i=0}^T \underline{T}_c \quad (83)$$

il vient :

$$\dot{Y}_1 + \dot{Y}_2 = \frac{1}{2}\alpha\lambda \frac{z_1|_k - z_2|_k}{z_1|_k z_2|_k} \frac{l^2(l-l_c)}{z} \quad (84)$$

on en déduit alors le signe de  $\omega_0$  directement relié à celui de  $z_1|_k - z_2|_k$  :

$$\text{signe}(\omega_0) = \text{signe}(z_1|_k - z_2|_k) = \text{signe}\left(\frac{\dot{Y}_1 + \dot{Y}_2}{l-l_c}\right) \quad (85)$$

La loi de commande étant complètement définie, on va la tester grâce à des simulations. Elles font l'objet des paragraphes suivants.

### 3.4.4 Simulations non bruitées

Les figures 7(a), 7(b), 7(c) et 7(d) montrent, une fois l'image centrée et ajustée à  $l = l_c$ , le comportement de l'algorithme proposé.

Tout d'abord, on s'aperçoit que l'erreur de modélisation est très faible. On remarque aussi que  $\theta_{xc}$  devient rapidement stable et, finalement, cette bonne modélisation couplée à une bonne identification conduit à une convergence rapide.

Nous allons donc tester dans le paragraphe suivant la robustesse de l'algorithme vis-à-vis d'une image bruitée.

### 3.4.5 Simulations bruitées

Les informations visuelles sont bruitées conformément à (43) mais cette fois dans l'intervalle  $[-2p_y, 2p_y]$ , soit par un bruit uniforme d'amplitude 2 pixels. Pour ces simulations seule la durée d'apprentissage a été modifiée, elle a été portée à 20 itérations.

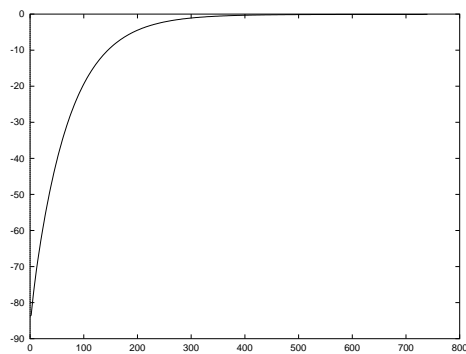
À la vue des figures 8(a), 8(b) et 8(c) on remarque tout d'abord nettement la présence de la période initiale destinée à l'apprentissage, période pendant laquelle  $\omega_x$  est constant. Par ailleurs, la consigne angulaire  $\theta_{xc}$  est cette fois un peu plus bruitée mais se stabilise toujours relativement vite. On peut noter aussi la faible influence du bruit sur  $\omega_x$  ou sur l'erreur de positionnement vis-à-vis des oscillations qu'il aurait pu engendrer. En effet, comme le modèle recopie bien la réalité on peut imposer une forte valeur de  $\gamma$  et ainsi filtrer fortement le bruit.

### 3.4.6 Discussion sur l'erreur de positionnement

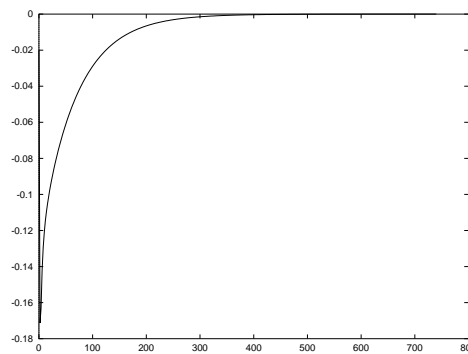
L'algorithme décrit ci-dessus repose sur deux points importants :

- Tout d'abord sur une modélisation correcte de la taille du segment dans l'image. On a vu que la validité du modèle dépendait soit de la situation du porteur par rapport à l'objet par l'intermédiaire de la variable  $v$  soit de l'hypothèse selon laquelle la trajectoire du porteur, dans le repère de l'objet, serait une pseudo rotation autour d'un point fixe. Cette dernière hypothèse n'est en pratique pas très exploitable car difficilement vérifiable. On s'efforcera donc de vérifier  $v$  faible même si cette hypothèse est plus forte comme on l'a vu lors de simulations.

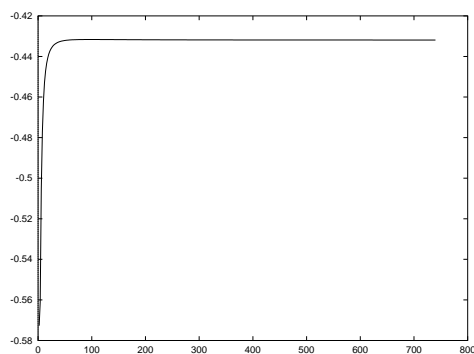
Si besoin était, n'oublions pas qu'une identification non linéaire pourrait toujours être envisagée avec les restrictions apportées dans le paragraphe 3.4.2.



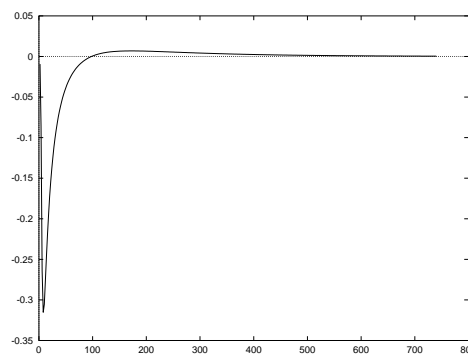
(a) Évolution de l'erreur de positionnement  $z_1 - z_2$  (mm) en fonction du nombre d'itérations



(b) Évolution de  $\omega_x$  (rad/s) en fonction du nombre d'itérations

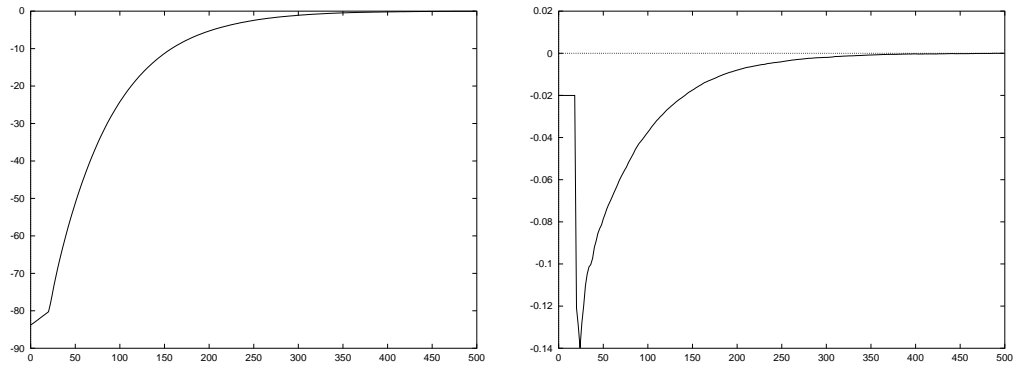


(c) Évolution de  $\theta_{xc}$  estimé (rad) en fonction du nombre d'itérations



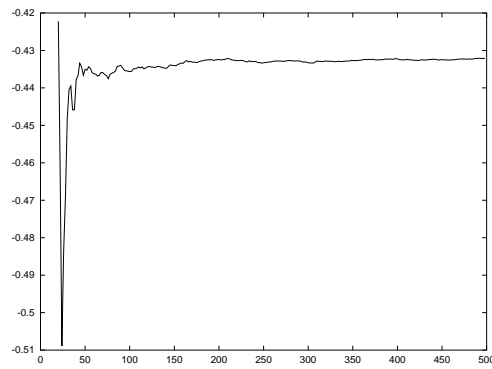
(d) Évolution de l'erreur de modélisation  $l - l_a$  (pixel) en fonction du nombre d'itérations

FIG. 7 — Maximisation de la taille de l'objet dans l'image — Informations visuelles non bruitées —



(a) Évolution de l'erreur de positionnement  $z_1 - z_2$  en fonction du nombre d'itérations

(b) Évolution de  $\omega_x$  en fonction du nombre d'itérations



(c) Évolution de  $\theta_{xc}$  estimé en fonction du nombre d'itérations

FIG. 8 — Maximisation de la taille de l'image — Informations visuelles bruitées par un bruit d'amplitude deux pixels —



- Il repose aussi sur la qualité de l'identification. Une première contrainte vient immédiatement à l'esprit : disposer d'un nombre suffisant de couples  $(\theta_x|_k, l_k)$  puis, une deuxième concernant la répartition de ces points. En effet, pour réaliser un apprentissage global de la courbe  $l(\theta_x)$  et non pas un apprentissage local (qui serait trop dépendant du bruit) il faut, dans la mesure du possible, disposer d'autant de couples au début du mouvement qu'à la fin. La conséquence de ces deux contraintes est de devoir limiter  $\omega_x$  en imposant  $K_x$  suffisamment faible. Pour les mêmes raisons on a intérêt à choisir  $\omega_0$  faible.

Par contre, quand l'erreur de positionnement initiale  $z_1 - z_2$  est très faible et par suite l'erreur angulaire aussi, les deux contraintes précédentes ne peuvent être que difficilement remplies et une erreur de positionnement relativement importante peut apparaître. Le cas le plus défavorable que nous avons rencontré conduit à un comportement rapporté sur la figure 9 où une erreur de positionnement initiale faible ( $\approx 15$  mm) devient inférieure à 1.5 mm après asservissement.

Après avoir traité le cas des objets de type «segment», nous allons, dans le paragraphe suivant, voir comment transposer la démarche suivie et les résultats obtenus au cas d'un objet plan de forme quelconque.

## 4 Étude d'un objet plan de forme quelconque

Dans le paragraphe 3, on a réussi à réaliser la tâche de positionnement souhaitée en maximisant un critère — la taille de l'image de l'objet segment — au cours du mouvement. Dans le cas d'un objet décrit dans un espace de dimension deux une transposition naturelle consiste à maximiser la surface  $S$  de l'image de l'objet. Pour ce faire, il serait intéressant de disposer d'une modélisation de cette surface. L'obtenir dans le cas général n'est pas simple aussi, allons nous dans un premier temps simplifier le problème en supposant que l'on puisse extraire de l'image trois points caractéristiques de l'objet.

### 4.1 Étude du cas où trois points caractéristiques existent

En ne raisonnant que sur ces trois points, l'image à traiter devient équivalente à une image beaucoup plus simple comme celle décrite sur la figure 10, c'est-à-dire à un triangle de sommets  $M_1(X_1, Y_1)$ ,  $M_2(X_2, Y_2)$  et  $M_3(X_3, Y_3)$  respectivement images des points  $m_1(x_1, y_1, z_1)$ ,  $m_2(x_2, y_2, z_2)$  et  $m_3(x_3, y_3, z_3)$ . En adoptant la même démarche qu'en 3, on va tout d'abord spécifier :

- une tâche principale qui permette l'équivalent du centrage réalisé dans le paragraphe 3.2.2 ;

- une tâche secondaire qui conduise à la réalisation de  $z_1 = z_2 = z_3$ .

#### 4.1.1 Spécification de la tâche principale

Dans le cas de l'objet segment le centrage de l'image avait une signification bien précise, maintenant on s'aperçoit que plusieurs approches pourraient être utilisées comme chercher à positionner le centre de gravité par exemple. Cependant, si l'on veut pouvoir maximiser  $l$ ,  $h$  ou même directement le produit  $lh$ , on a tout intérêt à rester dans les conditions du paragraphe 3.2.2 où l'on sait comment varie  $l$  ou  $h$ . C'est-à-dire à imposer :

- $Y_1 + Y_2 = 0$  et  $X_1 = X_2$  en ce qui concerne la maximisation de  $l$  (le cas  $X_1 = X_2 = 0$  du paragraphe 3.2.2 était un cas particulier du cas  $X_1 = X_2$ );
- $X_1 + X_3 = 0$  en ce qui concerne la maximisation de  $h$ ;

ce qui conduit à définir  $\underline{s}$  par :

$$\underline{s} = (X_1 - X_2, X_1 + X_3, Y_1 + Y_2)^T \quad (86)$$

et la consigne visuelle par :

$$\underline{s}^* = (0, 0, 0)^T \quad (87)$$

Les informations visuelles étant choisies on peut exprimer la matrice d'interaction en  $\underline{s}^*$  et pour  $z_i = z$  ( $i = 1 \dots 3$ ) :

$$L_{z_i=z}^T \Big|_{\underline{s}^*} = \begin{pmatrix} 0 & 0 & 0 & 2X_1Y_1 & 0 & 2Y_1 \\ -\frac{2}{z} & 0 & 0 & X_1(Y_1 - Y_3) & -2(1 + X_1^2) & Y_1 + Y_3 \\ 0 & -\frac{2}{z} & 0 & 2(1 + Y_1^2) & 0 & -2X_1 \end{pmatrix} \quad (88)$$

et les deux vecteurs suivants appartenant au noyau de  $L_{z_i=z}^T$  :

$$\underline{T}_{sl} = ( -z\omega_x X_1 Y_3 \quad z\omega_x (1 + X_1^2 + Y_1^2) \quad 0 \quad \omega_x \quad 0 \quad -\omega_x X_1 )^T \quad (89)$$

$$\underline{T}_{sh} = ( -z\omega_y (1 + X_1^2) \quad 0 \quad 0 \quad 0 \quad \omega_y \quad 0 )^T \quad (90)$$

On appliquera donc une loi de commande de la forme suivante :

$$T_c = -\lambda L_{z_i=z}^T \Big|_{\underline{s}^*} (\underline{s} - \underline{s}^*) + T_{sl} + T_{sh} \quad (91)$$

dans laquelle  $\omega_x$  et  $\omega_y$  sont à déterminer.

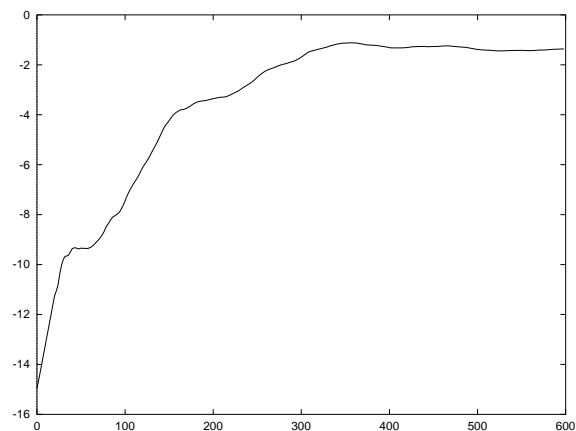


FIG. 9 – Évolution de l'erreur de positionnement en fonction du nombre d'itérations dans le cas où l'erreur initiale est faible — Informations visuelles bruitées par un bruit d'amplitude deux pixels —

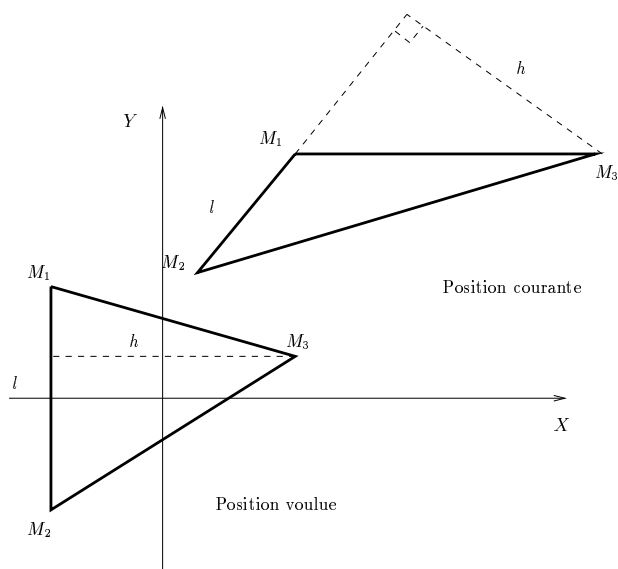


FIG. 10 – Triangle formé à l'aide des trois points caractéristiques ( $M_1, M_2, M_3$ ) — Position courante et position voulue —

#### 4.1.2 Modélisation de la surface de l'objet dans l'image

Soient  $m_1$ ,  $m_2$  et  $m_3$  les trois points caractéristiques de l'objet et  $m_0$  un point de l'espace tels que  $m_1 = (-x^*, y^*, z^*)^T$ ,  $m_2 = (-x^*, -y^*, z^*)^T$ ,  $m_3 = (x^*, r^*, z^*)^T$  et  $m_0 = (0, 0, z^*)^T$  coordonnées exprimées dans un repère fixe. On suppose que le mouvement réalisé lors de la maximisation de la surface de l'image résulte de trois rotations de centre  $m_0$  et d'une translation  $\mathcal{T}$ . Les coordonnées des points  $m_1$ ,  $m_2$  et  $m_3$  s'expriment alors dans le repère de la caméra par :

$$\begin{cases} m_1' &= \mathcal{R}(m_1 - m_0) + \mathcal{T} \\ m_2' &= \mathcal{R}(m_2 - m_0) + \mathcal{T} \\ m_3' &= \mathcal{R}(m_3 - m_0) + \mathcal{T} \end{cases} \quad (92)$$

où :

$$\mathcal{R} = \begin{pmatrix} C_{\theta_y} C_{\theta_z} & S_{\theta_y} S_{\theta_z} & S_{\theta_y} C_{\theta_y} \\ -C_{\theta_x} S_{\theta_z} - S_{\theta_x} S_{\theta_y} C_{\theta_z} & C_{\theta_x} C_{\theta_z} - S_{\theta_x} S_{\theta_y} S_{\theta_z} & S_{\theta_x} C_{\theta_y} \\ S_{\theta_x} S_{\theta_z} - C_{\theta_x} S_{\theta_y} C_{\theta_z} & -S_{\theta_x} C_{\theta_z} - C_{\theta_x} S_{\theta_y} S_{\theta_z} & C_{\theta_x} C_{\theta_y} \end{pmatrix}$$

avec les notations habituelles  $C_\theta = \cos \theta$  et  $S_\theta = \sin \theta$  et :

$$\mathcal{T} = \begin{pmatrix} t_x \\ t_y \\ z \end{pmatrix}$$

On en déduit par transformation perspective les coordonnées des points  $M_1 = (X_1, Y_1)^T$ ,  $M_2 = (X_2, Y_2)^T$  et  $M_3 = (X_3, Y_3)^T$  images des points  $m_1$ ,  $m_2$  et  $m_3$ . Puis, on cherche à maintenir l'image constante comme le ferait un mouvement appartenant au noyau de la matrice d'interaction. On résout donc, par rapport aux variables  $t_x$ ,  $t_y$ , et  $\theta_z$ , le système non linéaire suivant correspondant à  $\underline{s} = \underline{s}^*$  :

$$\begin{cases} X_1 - X_2 &= 0 \\ X_1 + X_3 &= 0 \\ Y_1 + Y_2 &= 0 \end{cases} \quad (93)$$

Son expression en fonction des inconnues est complexe et ne sera pas donnée ici. De plus, sa résolution analytique n'a malheureusement pas abouti. Toutefois, pour simplifier son expression on se propose de réaliser un développement en série de Taylor à l'ordre deux par rapport aux variables  $u = x^*/z^*$ ,  $v = y^*/z^*$  et  $w = r^*/z^*$ . De plus, on pose  $k_x = t_x/z^*$  et  $k_y = t_y/z^*$ . Tous calculs faits la résolution de (93) fournit :

$$\begin{cases} k_x &= 0 \\ k_y &= u \sin \theta_x \sin \theta_y \\ \theta_z &= 0 \end{cases} \quad (94)$$

Il s'en suit :

$$\begin{cases} l = Y_1 - Y_2 &= 2v \cos \theta_x \\ h = X_3 - X_1 &= 2u \cos \theta_y \end{cases} \quad (95)$$

On retrouve l'expression de  $l$  obtenue en 3.4.2, par contre le développement en série réalisé conduit à des expressions beaucoup trop simplifiées. En effet,  $l$  (respectivement  $h$ ) devrait dépendre de  $\theta_y$  (respectivement  $\theta_x$ ). De ce fait, les simulations de commande correspondantes ont échoué. Ne disposant pas d'une modélisation correcte, on ne peut donc pas au cours d'un même mouvement maximiser  $l$  et  $h$  simultanément. Par contre, on peut maximiser séquentiellement  $l$  puis  $h$ .

Pour que l'algorithme soit complet, il reste à déterminer les signes de  $\omega_x$  et  $\omega_y$  nécessaires à la phase d'apprentissage.

On procède pour cela comme en 3.4.3. Au cours d'une première phase de mouvement on cherche à atteindre la consigne visuelle tout en imposant une taille  $l_c$  de l'image. La fonction de tâche à réguler s'écrit :

$$\underline{e} = L_{\left| \begin{smallmatrix} \underline{s} = \underline{s}^* \\ z_i = z \end{smallmatrix} \right.}^{T+} (\underline{s} - \underline{s}^*) + \alpha \left( \mathbb{I}_6 - L_{\left| \begin{smallmatrix} \underline{s} = \underline{s}^* \\ z_i = z \end{smallmatrix} \right.}^{T+} L_{\left| \begin{smallmatrix} \underline{s} = \underline{s}^* \\ z_i = z \end{smallmatrix} \right.}^T \right) \underline{g}_s^T \quad (96)$$

$$\text{où } \underline{g}_s^T = \left( \frac{\partial h_s}{\partial \underline{r}} \right)^T \text{ avec } h_s = \frac{1}{2} (l - l_c)^2.$$

Supposons  $\alpha$  suffisamment petit pour que  $\underline{s} = \underline{s}^*$  et  $l \neq l_c$ . Alors, on peut montrer que  $\dot{s} = L_{\left| \begin{smallmatrix} \underline{s} = \underline{s}^* \\ z_i = z \end{smallmatrix} \right.}^T \underline{T}_c$  devient :

$$\dot{\underline{s}} = \left( \frac{\alpha \lambda h l (l - l_c)}{2z} \frac{z_1 - z_2}{z_1 z_2}, \frac{\alpha \lambda h l (l - l_c)}{2z} \frac{z_1 - z_3}{z_1 z_3}, \frac{\alpha \lambda l^2 (l - l_c)}{2z} \frac{z_1 - z_2}{z_1 z_2} \right)^T \quad (97)$$

on a donc accès, de la même façon qu'en 3.4.3, aux signes de  $z_1 - z_2$  et  $z_1 - z_3$  et par suite à ceux de  $\omega_x$  et  $\omega_y$ .

L'algorithme complet a été implanté en simulation sur des images bruitées, il a conduit à d'aussi bons résultats que ceux présentés en 3.4.5 avec les mêmes limitations qu'en 3.4.6.

Aucune représentation graphique de simulation ne sera proposée, elle serait redondante avec celle du paragraphes 3.4.5. Par contre les simulations ayant validé les différentes lois de commande, des résultats expérimentaux seront présentés dans les paragraphes suivants après une présentation des conditions d'expérimentation.

## 4.2 Résultats expérimentaux — cas où trois points de l'objet sont connus

### 4.2.1 Description de la plate-forme expérimentale

Le porteur est constitué d'un robot cartésien nécessaire aux trois degrés de liberté assurant les translations auquel on a ajouté une rotule (trois axes orthogonaux et concourants) assurant la réalisation des rotations (Figure 11).

Sur la rotule est fixée une caméra CCD Sony XC-8500CE. Elle possède la particularité d'exposer simultanément les trames paires et impaires et d'éviter ainsi le flou inter-trame. Elle dispose d'un capteur ayant la même résolution horizontale et verticale (pixels carrés de  $8.3 \mu\text{m} \times 8.3 \mu\text{m}$ ) et permet l'acquisition d'images de taille 782(H) x 582(V) au format CCIR. Elle est dotée d'un objectif de focale 12.5 mm.

Les acquisitions et traitements d'images sont réalisés par une carte GENESIS de la société Matrox. Elle est au format PC (bus PCI) et permet grâce à un processeur TMS320C80 de réaliser des traitements à grande vitesse. De plus, elle possède une carte graphique qui permet une visualisation indépendante du superviseur. La programmation de la carte est réalisée de manière transparente grâce à des bibliothèques C 32 bits dédiées à la carte.

Les asservissements articulaires sont réalisés par une carte d'axe DMC-1060 de la société GALIL. Elle est au format PC et permet l'asservissement des six axes du robot. Elle se programme en C.

Le superviseur est un PC équipé d'un Pentium à 200 MHz. Il permet de réaliser les calculs nécessaires à l'élaboration des lois de commande et de gérer les différentes cartes.

Par ailleurs, l'ensemble du dispositif a été calibré (jacobien du robot, caméra, matrice de passage de la caméra à la rotule).

L'ensemble de l'architecture a permis une implantation à la cadence vidéo des différentes lois de commande.

FIG. 11 – *Plate-forme expérimentale*

#### 4.2.2 Vérification des résultats

Lors des simulations il était facile d'obtenir la situation de la caméra par rapport à l'objet. Par contre, expérimentalement, cette orientation est beaucoup plus difficile à évaluer. Il s'agit en fait plus précisément de situer le capteur CCD lui-même par rapport à l'objet. La technique utilisée devra donc nécessairement faire intervenir la caméra. Deux approches sont possibles :

- mesurer à l'aide d'un dispositif à définir la situation de l'objet par rapport à un repère fixe. Puis, grâce à des calibrations (repère fixe  $\rightarrow$  porteur, porteur  $\rightarrow$  poignet, poignet  $\rightarrow$  caméra) obtenir cette situation par rapport à la caméra.
- utiliser une technique de localisation 3D. Pour ce faire une modélisation précise de l'objet ainsi qu'une caméra calibrée sont nécessaires.

L'une et l'autre des deux méthodes vont conduire, du fait d'erreurs de mesure, à une situation biaisée de l'objet par rapport à la caméra. Mais la seconde méthode est moins lourde et fait intervenir moins d'intermédiaires susceptibles d'entraîner des erreurs (passages repère fixe  $\rightarrow$  poignet et poignet  $\rightarrow$  caméra). En conséquence, notre choix s'est porté sur la deuxième méthode.

Nous avons choisi la méthode de localisation décrite dans [10]. Pour ce faire, outre les trois points utiles à l'algorithme de positionnement, nous avons ajouté un quatrième point nécessaire à la localisation 3D. Comme spécifié dans [10], il n'appartient pas au plan passant

par les trois autres points. De plus, il a été placé de telle sorte qu'une fois l'asservissement terminé sa projection soit confondue avec le centre du capteur. La figure 12 représente l'objet utilisé.

### 4.2.3 Résultats expérimentaux

Les figures 13(a), 14(a), 14(b) et 13(b) montrent les comportements respectifs des composantes du torseur cinématique, de la consigne angulaire et de la position angulaire courante, de la taille de l'image réelle et modélisée et de l'erreur dans le capteur (définie comme étant  $\|\underline{g}\|$ ) lors de la maximisation de  $l$ . Les figures 15(a), 16(a), 16(b) et 15(b) représentent l'évolution des mêmes variables que précédemment mais lors de la maximisation de  $h$ . La figure 17 résume les différentes phases nécessaires à la réalisation de la tâche de positionnement souhaitée. On peut remarquer par ailleurs sur les figures 13(b) et 15(b) l'apparition de parasites, dûs aux moteurs du robot, réinjectés sur le signal vidéo.

Tout d'abord ces résultats expérimentaux confirment les simulations sur images bruitées, en particulier les figures 14(b) et 16(b) valident expérimentalement les résultats théoriques concernant les modélisations de  $l = f(\theta_x)$  et  $h = f(\theta_y)$ . On peut remarquer aussi sur les figures 14(a) et 16(a) l'obtention rapide d'une valeur de consigne angulaire qui s'affine au fur et à mesure de l'apprentissage puis qui se stabilise. Il est à noter, comme en simulation, la faible influence du bruit sur le calcul de cette valeur.

Par ailleurs, nous avons appliqué la méthode de Dementhon sur 100 images correspondant à la position initiale de la caméra et sur 100 images correspondant à la position finale<sup>1</sup>. Elle permet de calculer la position des trois points  $m_1$ ,  $m_2$  et  $m_3$  dans le repère de la caméra et l'orientation de la caméra par rapport à l'objet. On a caractérisé la matrice de rotation correspondante par les angles d'Euler, soient  $\gamma$ ,  $\beta$  et  $\alpha$  ces angles représentant respectivement les rotations autour des axes  $x$ ,  $y$  et  $z$ . Dans le cas de l'exemple présenté les résultats sont les suivants :

- Position initiale de la caméra par rapport à l'objet :
- Matrice de passage caméra/objet :

$$\begin{pmatrix} 0.9714 \pm 0.0000 & -0.2085 \pm 0.0003 & -0.1132 \pm 0.0003 \\ 0.1584 \pm 0.0002 & 0.9253 \pm 0.0000 & -0.3444 \pm 0.0002 \\ 0.1766 \pm 0.0003 & 0.3166 \pm 0.0002 & 0.9319 \pm 0.0000 \end{pmatrix}$$

- Angles d'Euler (en degrés) :

$$\gamma = 18.76 \pm 0.01 \quad \beta = 10.17 \pm 0.01 \quad \alpha = 9.26 \pm 0.01$$

---

1. L'acquisition d'un grand nombre d'images permet de donner une mesure et un intervalle de validation à  $\pm 2\sigma$ .



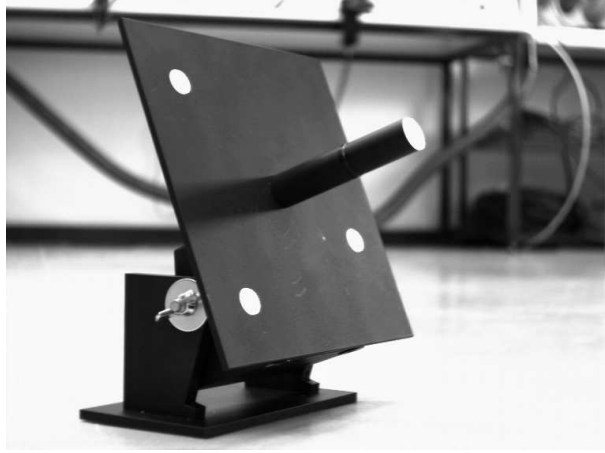


FIG. 12 – *Objet permettant la reconstruction 3D*

- Coordonnées des points caractéristiques (exprimées dans le repère de la caméra en mm) :

$$\begin{aligned} m_1 &= ( 21.01 \pm 0.03 \quad 26.61 \pm 0.07 \quad 577.51 \pm 0.07 )^T \\ m_2 &= ( 48.12 \pm 0.02 \quad -93.68 \pm 0.07 \quad 536.34 \pm 0.07 )^T \\ m_3 &= ( -84.93 \pm 0.01 \quad -84.20 \pm 0.07 \quad 523.67 \pm 0.08 )^T \end{aligned}$$

- Position finale de la caméra par rapport à l'objet :

- Matrice de passage caméra/objet :

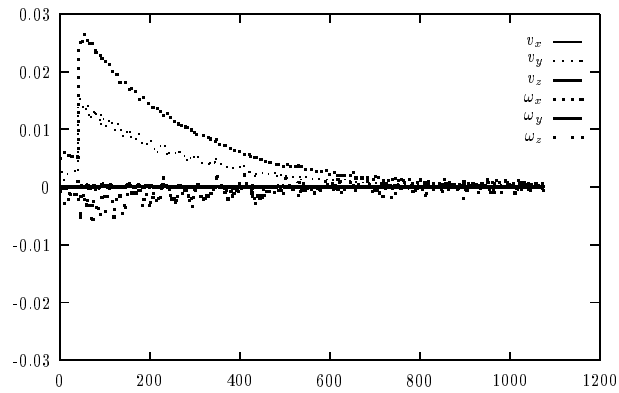
$$\begin{pmatrix} 0.9999 \pm 0.0000 & 0.0007 \pm 0.0005 & 0.0040 \pm 0.0004 \\ -0.0006 \pm 0.0005 & 0.9999 \pm 0.0000 & -0.0051 \pm 0.0004 \\ -0.0040 \pm 0.0004 & 0.0051 \pm 0.0004 & 0.9999 \pm 0.0000 \end{pmatrix}$$

- Angles d'Euler (en degrés) :

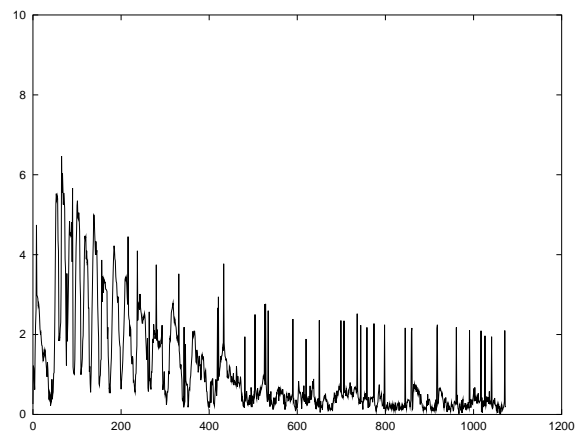
$$\gamma = 0.29 \pm 0.02 \quad \beta = -0.23 \pm 0.02 \quad \alpha = -0.039 \pm 0.02$$

- Coordonnées des points caractéristiques (exprimées dans le repère de la caméra en mm) :

$$\begin{aligned} m_1 &= ( 65.05 \pm 0.05 \quad 65.04 \pm 0.15 \quad 503.08 \pm 0.12 )^T \\ m_2 &= ( 64.96 \pm 0.03 \quad -64.96 \pm 0.15 \quad 502.41 \pm 0.12 )^T \\ m_3 &= ( -65.01 \pm 0.03 \quad -32.37 \pm 0.14 \quad 503.11 \pm 0.13 )^T \end{aligned}$$

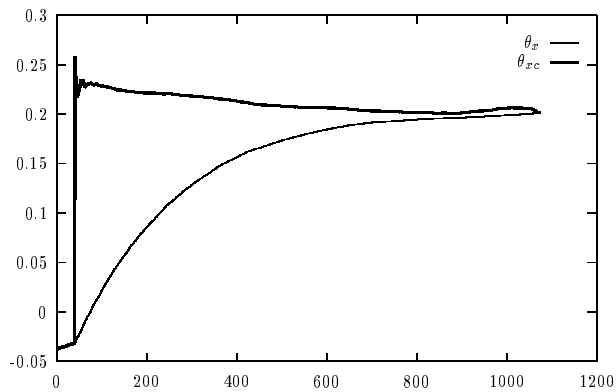


(a) Composantes du torseur cinématique (m/s ou rad/s) en fonction du nombre d'itérations

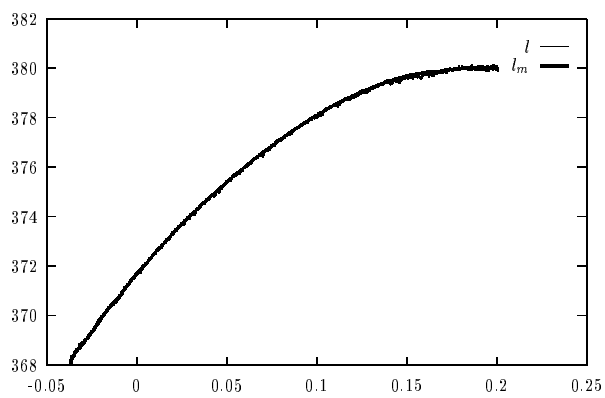


(b) Erreur dans le capteur (pixel) en fonction du nombre d'itérations

FIG. 13 – Résultats expérimentaux — cas où trois points de l'objet sont connus — Maximisation de  $l$

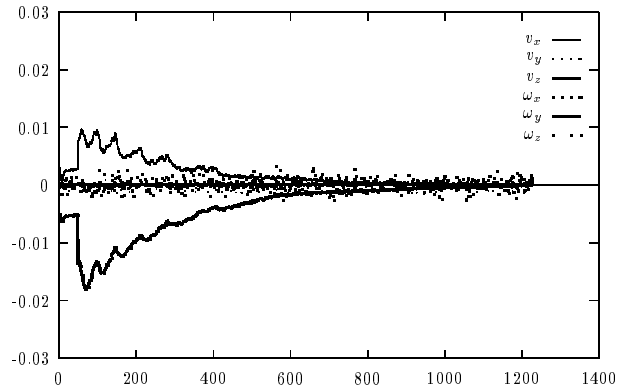


(a) Consigne angulaire (rad.) et position angulaire courante (rad.) en fonction du nombre d'itérations

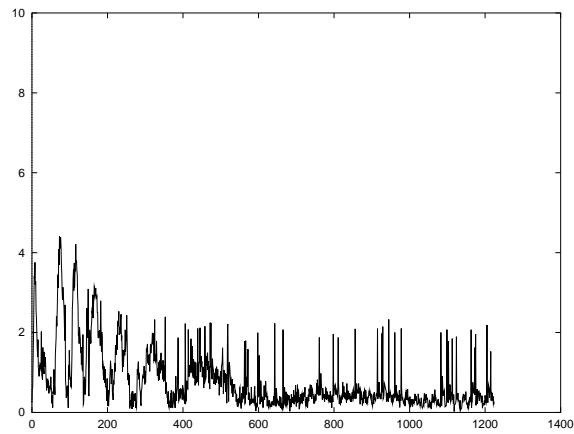


(b) Taille de l'image et taille modélisée (pixel) en fonction de  $\theta_x$

FIG. 14 – Résultats expérimentaux — cas où trois points de l'objet sont connus — Maximisation de  $l$

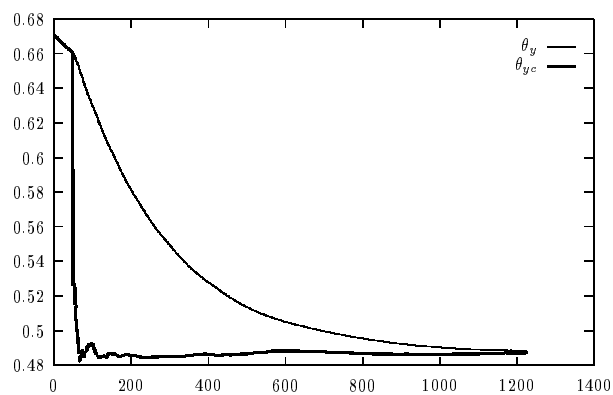


(a) Composantes du torseur cinématique (m/s ou rad/s) en fonction du nombre d'itérations

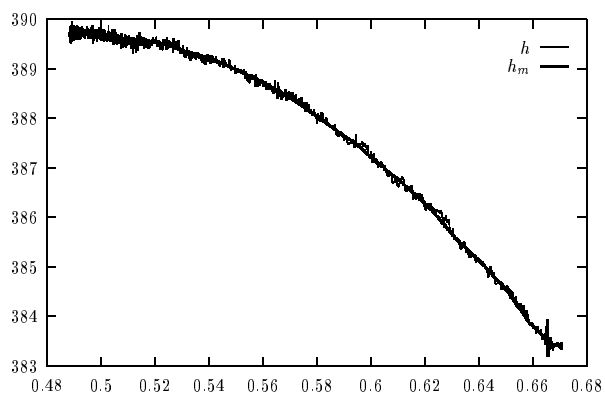


(b) Erreur dans le capteur (pixel) en fonction du nombre d'itérations

FIG. 15 – Résultats expérimentaux — cas où trois points de l'objet sont connus — Maximisation de  $h$



(a) Consigne angulaire (rad.) et position angulaire courante (rad.) en fonction du nombre d'itérations



(b) Taille de l'image et taille modélisée (pix.) en fonction de  $\theta_y$

FIG. 16 – Résultats expérimentaux — cas où trois points de l'objet sont connus — Maximisation de  $h$

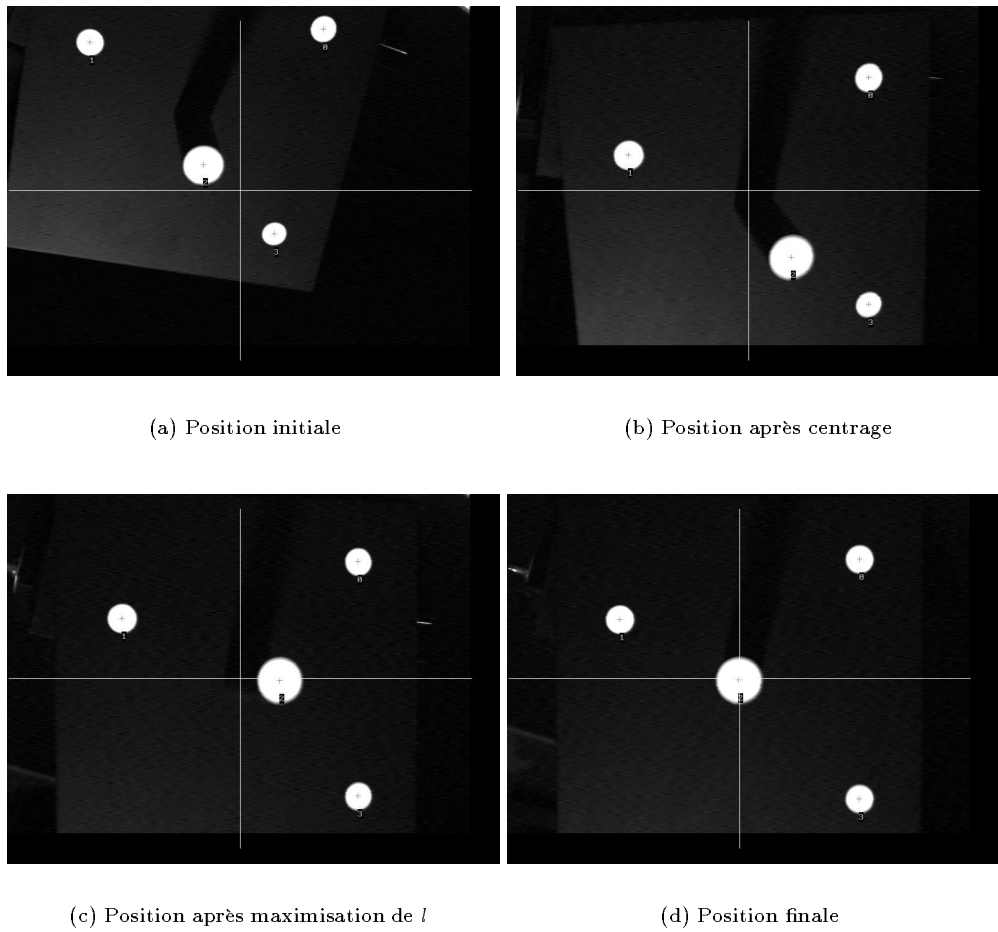


FIG. 17 – Résultats expérimentaux — cas où trois points de l'objet sont connus —

Si le positionnement était réalisé sans erreur, on devrait obtenir une matrice de rotation égale à l'identité et par conséquent des angles d'Euler tous nuls et, pour les points  $m_i$  les valeurs suivantes :

$$\begin{aligned} m_1 &= ( 65 \quad 65 \quad z^* )^T \\ m_2 &= ( 65 \quad -65 \quad z^* )^T \\ m_3 &= ( -65 \quad -32.5 \quad z^* )^T \end{aligned}$$

Par conséquent, les résultats obtenus sont tout à fait satisfaisants. Par ailleurs, on s'aperçoit une nouvelle fois que les valeurs des variables  $u$ ,  $v$  et  $w$  (respectivement  $\approx 0.13$ ,  $\approx 0.13$  et  $\approx 0.06$ ) sont relativement importantes et que les modélisations de  $l$  et  $h$  sont quand même correctes.

Pour valider ce résultat — obtenu pour une position initiale donnée — nous avons réalisé 25 autres essais en suivant la même procédure mais pour d'autres positions initiales choisies dans le volume tel que  $\gamma \in [-12^\circ, 27^\circ]$ ,  $\beta \in [-22^\circ, 35^\circ]$  et  $z_4 \in [443 \text{ mm}, 803 \text{ mm}]$ . Les résultats sont représentés sur la figure 18 où le numéro de l'essai est porté en abscisse et les valeurs moyennes de  $\gamma$  et  $\beta$  sur les 100 images en ordonnée<sup>2</sup>. De plus, des résultats statistiques descriptifs sont résumés dans le tableau 1.

Par ailleurs les constantes utilisées dans l'algorithme pour l'ensemble des essais étaient :

- Phase de centrage :
  - $\alpha = 1.5$  ;
  - $l_c = 370$  pixels ;
  - $z = 60$  cm ;
  - $\lambda = 2$ .
- Phase de maximisation de  $l$  :
  - $\omega_0 = 0.005$  rad/s ;
  - $\lambda = 3$  ;
  - facteur d'oubli pour les moindres carrés = 0.99 ;
  - $K_x = 0.1$  ;
  - nombre d'acquisitions d'apprentissage = 40 ;
  - condition d'arrêt :  $|\theta_{xc} - \theta_x| < 1e - 3$  et  $|\theta_{xc,k} - \theta_{xc,k-1}| < 1e - 4$ .

---

2. Nous ne présentons pas de résultat concernant l'angle  $\alpha$  car l'information visuelle  $X_1 - X_2$  a pour consigne zéro ce qui conduit à le rendre nul.

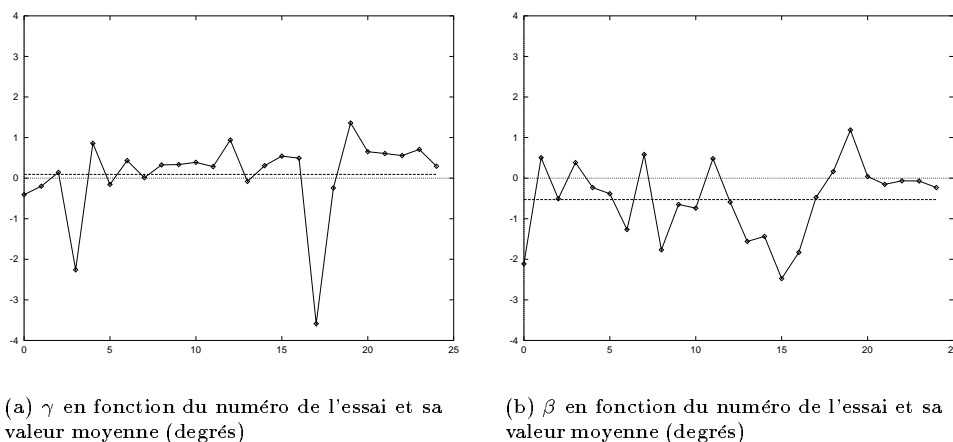


FIG. 18 — Résultats expérimentaux sur 25 essais — cas où trois points de l'objet sont connus —

$x$	$\bar{m} = \frac{1}{N} \sum_i x_i$	$\sup  x_i $	$\sigma$	$\inf  x_i $
$\gamma$	0.09	3.58	0.99	0.01
$\beta$	-0.53	2.47	0.91	0.04

TAB. 1 — Résultats expérimentaux sur 25 essais (en degrés) — cas où trois points de l'objet sont connus —



- Phase de maximisation de  $h$  :
  - $\omega_0 = 0.005$  rad/s;
  - $\lambda = 5$ ;
  - facteur d'oubli pour les moindres carrés = 0.995 ;
  - $K_y = 0.1$  ;
  - nombre d'acquisitions d'apprentissage = 50 ;
  - condition d'arrêt :  $|\theta_{yc} - \theta_y| < 1e - 3$  et  $|\theta_{yc,k} - \theta_{yc,k-1}| < 1e - 4$ .

Compte tenu du volume très important dans lequel les positions initiales étaient choisies, les résultats sont globalement bons. Cependant, certaines positions conduisent à des erreurs importantes. Les causes d'erreur ont déjà été détaillées en 3.4.6 lors des simulations et se vérifient ici aussi. En effet, on a vu qu'il était important que l'erreur angulaire initiale soit importante une fois la phase de centrage réalisée (et non pas en position initiale avant le début du mouvement de centrage). Malheureusement, à partir d'une configuration initiale caméra / objet donnée on ne peut pas évaluer cette erreur angulaire après centrage puisque, d'une part l'utilisation d'un algorithme de commande référencée vision ne permet pas de connaître la situation de la caméra une fois le centrage terminé et d'autre part la situation de l'objet est inconnue.

Par contre, au cours du mouvement, quand  $\theta_{xc}$  devient stable on connaît l'amplitude angulaire réalisée et, si elle s'avère trop faible, on peut décider de changer le sens de déplacement et imposer une valeur de  $\theta_{xc}$  de l'ordre d'une dizaine de degrés. On considère ensuite la position courante ainsi obtenue comme la nouvelle position initiale à partir de laquelle pourra débiter, dans des conditions favorables, un nouvel apprentissage.

Par ailleurs, pour un objet donné, on peut expérimentalement définir grossièrement un volume de l'espace pour lequel une fois le centrage réalisé les erreurs angulaires initiales seront suffisantes.

#### 4.2.4 Comportement de la loi de commande en environnement non calibré

Dans le but de comparer les performances de la méthode proposée par rapport à une démarche reconstructionniste, nous considérons maintenant le cas où la calibration réalisée est entachée d'erreurs. Deux types de calibration entrent en jeu :

- La calibration de la caméra : elle a été modélisée en considérant un déplacement  $(X_c, Y_c)$  par rapport au centre du capteur et en prenant en compte la distorsion radiale  $K_d$ . On obtient ainsi le modèle suivant :

$$\begin{cases} X_p &= X_c + F_x X (1 + K_d (X^2 + Y^2)) \\ Y_p &= Y_c + F_y Y (1 + K_d (X^2 + Y^2)) \end{cases} \quad (98)$$

Les résultats de la calibration ont donné les valeurs suivantes :

$$\begin{cases} X_c = 378.2 & (\text{pixels}) & Y_c = 299.8 & (\text{pixels}) \\ F_x = 1508.5 & & F_y = 1508.9 & \\ K_d = -0.08 & & & \end{cases}$$

- La calibration du porteur : elle intervient lors du calcul du jacobien inverse pour fournir, à partir du torseur cinématique, les consignes de vitesse articulaire. Dans cette transformation, le point le plus délicat est l'identification de la matrice  $M_{pc}$ . Elle exprime l'orientation et la translation du repère du porteur par rapport au repère de la caméra. L'identification de cette matrice a donné les valeurs suivantes :

$$\begin{pmatrix} -0.0024 & 0.9999 & -0.0001 & -89 \text{ mm} \\ -0.9999 & -0.0024 & -0.0062 & 3 \text{ mm} \\ -0.0062 & 0.0001 & 0.9999 & 46 \text{ mm} \end{pmatrix}$$

Nous avons, dans le cas d'un ensemble robot/caméra calibré, réalisé une tâche de positionnement conformément à l'algorithme proposé et avons obtenu les angles d'Euler suivants :

$$\gamma = 0.12 \quad \beta = -0.49 \quad \alpha = -0.02 \text{ (en degrés)}$$

Nous avons tout d'abord considéré le cas d'une mauvaise identification des paramètres intrinsèques de la caméra en imposant :

$$\begin{cases} X_c = 391 & (\text{pixels}) & Y_c = 291 & (\text{pixels}) \\ F_x = 1360 & & F_y = 1360 & \\ K_d = 0 & & & \end{cases}$$

$X_c$  et  $Y_c$  ont été choisis comme étant le milieu de l'image,  $F_x$  et  $F_y$  ont été minorés de 10 % par rapport aux valeurs obtenues par la calibration, la distorsion a été négligée.

À partir des mêmes positions du porteur et de l'objet, une fois la tâche de positionnement réalisée, nous avons obtenu cette fois les résultats suivants :

$$\gamma = 0.5 \quad \beta = -0.86 \quad \alpha = -0.06 \text{ (en degrés)}$$

La loi de commande s'avère donc peu sensible vis-à-vis des erreurs de calibration de la caméra.

Considérons maintenant le cas où la matrice  $M_{pc}$  a été incorrectement identifiée. Pour ce faire, une erreur de 5 degrés a été introduite sur chaque axe, et une erreur de 3 cm sur les translations. La matrice  $M_{pc}$  devient :

$$\begin{pmatrix} 0.0944 & 0.9924 & -0.0789 & -59 \text{ mm} \\ -0.9917 & 0.0868 & -0.0944 & 33 \text{ mm} \\ -0.0868 & 0.0871 & 0.9924 & 76 \text{ mm} \end{pmatrix}$$

Après positionnement de la caméra, suivant la procédure décrite ci-dessus, les angles suivants ont été obtenus :

$$\gamma = -1.46 \quad \beta = 1.71 \quad \alpha = 0.19 \text{ (en degrés)}$$

L'erreur d'orientation est cette fois plus importante. En fait, le biais introduit dans la matrice  $M_{pc}$  peut intervenir de deux façons :

- lors du calcul du jacobien du robot ;
- lors de la reconstruction des angles  $\theta_x$  et  $\theta_y$ .

Pour analyser le comportement de la loi de commande dans ces deux cas, une matrice incorrectement identifiée a été utilisée pour le calcul du jacobien et par contre, une matrice correctement identifiée a été utilisée pour la reconstruction des angles. Dans ces conditions, les résultats sont les suivants :

$$\gamma = -1.61 \quad \beta = 1.84 \quad \alpha = 0.23 \text{ (en degrés)}$$

Par conséquent, contrairement à ce que l'on imaginait *a priori*, c'est moins la reconstruction des angles qui influe sur l'erreur d'orientation que le jacobien du robot. On peut maintenant comprendre pourquoi cette erreur apparaît. En effet, les mouvements imposés lors des phases de maximisation et que l'on souhaite appartenir au noyau de la matrice d'interaction n'y appartiennent pas et, la tâche principale introduit alors des mouvements de compensation qui conduisent à une trajectoire différente de celle escomptée.

Finalement, pour apporter un réel plus à la méthode proposée par rapport à des méthodes de reconstruction 3D, il faudrait envisager dans le futur des mouvements réellement en boucle fermée et non plus des mouvements générés par la tâche secondaire qui sont, en fait, réalisés en boucle ouverte. Mais, comme on l'a vu en 3.3.2, cela semble difficile à réaliser.

### 4.3 Étude d'un objet plan de forme quelconque

On a vu dans le paragraphe 4.1.2 qu'il n'était pas possible de trouver une expression analytique de la surface de l'objet dans l'image en fonction de la situation de la caméra, mais que l'on pouvait aboutir à la position souhaitée grâce à deux mouvements successifs. On va chercher à appliquer ici la même procédure.

### 4.3.1 Choix des informations visuelles et obtention d'une loi de commande

Considérons dans le capteur une forme *a priori* quelconque et quatre points  $M_i$  pour  $i = 1, \dots, 4$  résultant de l'intersection du contour et des axes principaux d'inertie. Si les points  $M_1$  et  $M_2$  d'une part appartenaient à un objet 1 et les points  $M_3$  et  $M_4$  d'autre part appartenaient à un objet 2, on choisirait d'après le paragraphe 3.2.2 les informations visuelles suivantes :

$$\begin{cases} X_1 - X_2 & = & 0 \\ Y_1 + Y_2 & = & 0 \end{cases} \quad (99)$$

et

$$\begin{cases} Y_3 - Y_4 & = & 0 \\ X_3 + X_4 & = & 0 \end{cases} \quad (100)$$

Dans le cas d'un seul et même objet et du fait que les axes principaux d'inertie sont orthogonaux, la condition  $X_1 - X_2 = 0$  entraîne  $Y_3 - Y_4 = 0$  et par conséquent on peut choisir pour informations visuelles :

$$\underline{s} = (X_1 - X_2, X_3 + X_4, Y_1 + Y_2)^T \quad (101)$$

et :

$$\underline{s}^* = (0, 0, 0)^T \quad (102)$$

le torseur d'interaction calculé en  $\underline{s}^*$ ,  $Y_4 = Y_3$  et en  $z_i = z$  s'écrit :

$$L^T \Big|_{\substack{\underline{s}^* \\ Y_4=Y_3 \\ z_i=z}} = \begin{pmatrix} 0 & 0 & 0 & 2X_1Y_1 & 0 & 2Y_1 \\ -2/z & 0 & 0 & 0 & -2(1+X_3^2) & 2Y_3 \\ 0 & -2/z & 0 & 2(1+Y_1^2) & 0 & -2X_1 \end{pmatrix} \quad (103)$$

On s'intéresse une nouvelle fois aux vecteurs  $\underline{T}_{sl}$  et  $\underline{T}_{sh}$  appartenant au noyau de  $L^T|_{\underline{s}^*, Y_4=Y_3, z_i=z}$  :

$$\underline{T}_{sl} = ( -z\omega_x X_1 Y_3 \quad z\omega_x (1 + X_1^2 + Y_1^2) \quad 0 \quad \omega_x \quad 0 \quad -\omega_x X_1 )^T \quad (104)$$

$$\underline{T}_{sh} = ( -z\omega_y (1 + X_3^2) \quad 0 \quad 0 \quad 0 \quad \omega_y \quad 0 )^T \quad (105)$$

Ils doivent permettre, comme précédemment, au cours des mouvements qu'ils engendrent la maximisation tour à tour de  $l$  ( $Y_1 - Y_2$ ) puis  $h$  ( $X_3 - X_4$ ). Une fois la tâche réalisée, on doit obtenir une image du même type que celle qui est représentée sur la figure 19.

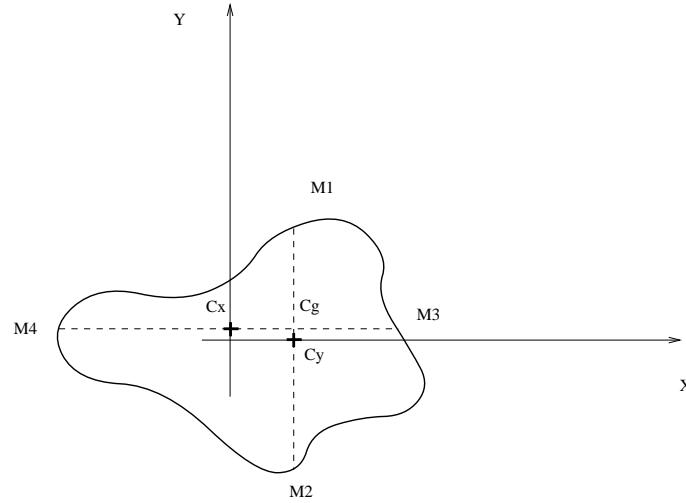


FIG. 19 – *Objet plan de forme quelconque — Motif désiré —*

#### 4.3.2 Résultats expérimentaux

Les figures 20(a), 21(a), 21(b) et 20(b) montrent les comportements respectifs des composantes du torseur cinématique, de la consigne angulaire et de la position angulaire courante, de la taille de l'image réelle et modélisée et de l'erreur dans le capteur lors de la maximisation de  $l$ . Les figures 22(a), 23(a), 23(b) et 22(b) représentent l'évolution des mêmes variables que précédemment mais lors de la maximisation de  $h$ . La figure 24 résume les différentes phases nécessaires à la réalisation de la tâche de positionnement souhaitée.

Le comportement de la loi de commande est grossièrement identique à celle du paragraphe 4.2.3. Cependant, lors de la maximisation de  $l$ , l'erreur dans le capteur (figure 20(b)) est cette fois beaucoup plus importante et, de plus ne commence à tendre vers zéro que plus tardivement. Cela se comprend aisément. En effet, les informations visuelles sont construites à partir des coordonnées  $(X_i, Y_i)$  de points qui ne représentent pas, à chaque itération, les images des mêmes points physiques puisque ces informations dépendent de la forme de l'image de l'objet. Par conséquent, les mouvements du noyau de  $L_s^T, Y_4=Y_3, z_i=z$  ne laissent pas les informations visuelles invariantes<sup>3</sup> et, il s'en suit une erreur dans l'image importante. Cette erreur conduit à des commandes de compensation provenant de la tâche principale et donc à une trajectoire différente de celle escomptée.

3. Les informations visuelles seraient invariantes si, au cours du mouvement, les axes d'inertie avaient une orientation constante.

Dans le cas de la maximisation de  $h$  apparaît le même phénomène. Il est clairement mis en évidence sur la figure 22(a) où des composantes non nulles suivant  $v_y$ ,  $\omega_x$  et  $\omega_z$  existent contrairement à (105).

Pour quantifier plus précisément la qualité du positionnement obtenu on a procédé comme suit. Tout d'abord, un objet plan constitué de quatre points marqués a été placé au même endroit que l'objet quelconque pour localiser la position de la caméra par rapport à l'objet suivant la même procédure qu'en 4.2.3. Puis, une fois la tâche de positionnement réalisée, l'objet quelconque a été remplacé par l'objet marqué pour localiser une nouvelle fois la caméra par rapport à l'objet. Les résultats obtenus sont les suivants :

- Position initiale de la caméra par rapport à l'objet :

- Angles d'Euler (en degrés) :

$$\gamma = 16.13 \pm 0.01 \quad \beta = 9.05 \pm 0.02 \quad \alpha = 16.56 \pm 0.03$$

- Position finale de la caméra par rapport à l'objet :

- Angles d'Euler (en degrés) :

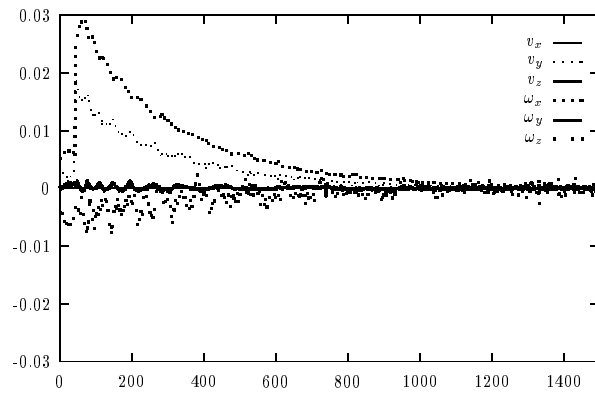
$$\gamma = 0.05 \pm 0.01 \quad \beta = 3.9 \pm 0.01 \quad \alpha = 45.10 \pm 0.03$$

Nous avons réalisé ensuite une série de 10 essais pour différentes positions initiales. Les résultats sont consignés d'une part dans le tableau 2 et d'autre part sur la figure 25 où le numéro de l'essai est porté en abscisse et les valeurs moyennes de  $\gamma$  et  $\beta$  sur les 100 images en ordonnée. Ces résultats nous confirment que le positionnement effectué est moins précis que dans le cas où trois points caractéristiques de l'objet sont disponibles dans l'image.

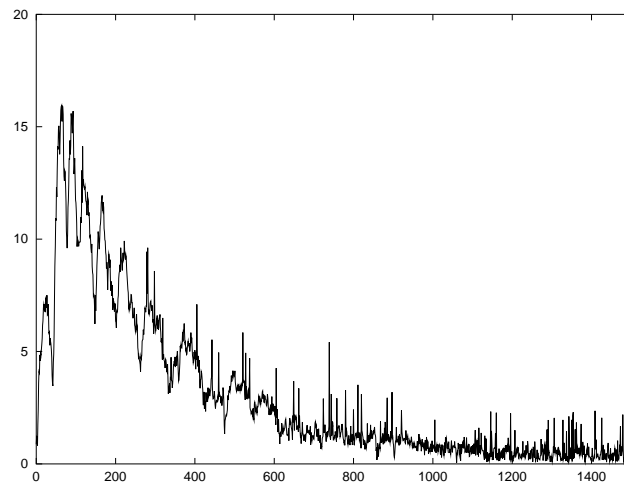
Par conséquent, dans le cas où l'on souhaite positionner une caméra en face d'un objet plan de forme quelconque, on privilégiera l'approche extraction de trois points caractéristiques physiques où, comme on l'a vu en 4.2.3, des performances importantes en terme d'erreur de positionnement peuvent être atteintes.

## 5 Conclusion et perspectives

Nous avons présenté dans ce rapport la commande d'un robot par asservissement visuel 2D à l'aide d'informations très simples et cela dans le cas où l'objet observé est plan, de forme quelconque et immobile. Une attention particulière a été portée au cas où trois points caractéristiques peuvent être extraits de l'image. Une discussion concernant les erreurs de positionnement rencontrées a conclu qu'elles pouvaient être limitées dans la mesure où un pré-positionnement relativement grossier de la caméra était possible.

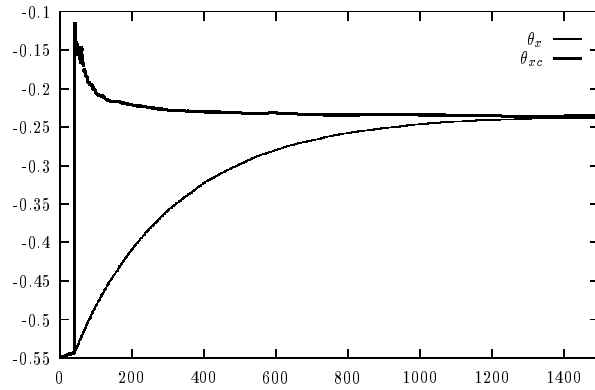


(a) Composantes du torseur cinématique (m/s ou rad/s) en fonction du nombre d'itérations

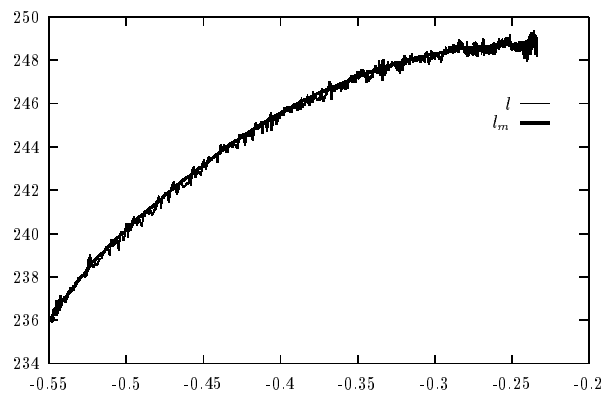


(b) Erreur dans le capteur (pix.) en fonction du nombre d'itérations

FIG. 20 – Résultats expérimentaux — cas d'un objet quelconque — Maximisation de  $l$



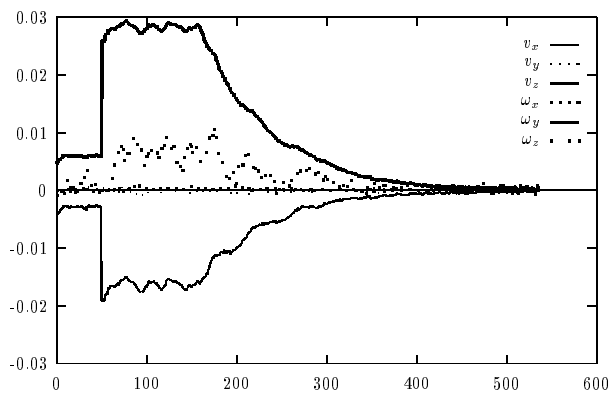
(a) Consigne angulaire (rad.) et position angulaire courante (rad.) en fonction du nombre d'itérations



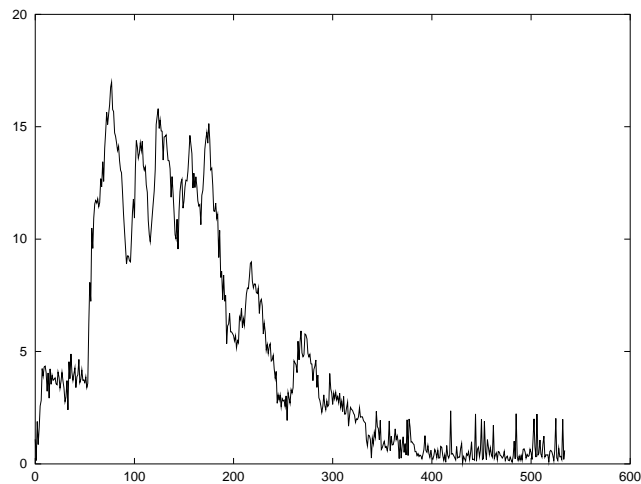
(b) Taille de l'image et taille modélisée (pix.) en fonction de  $\theta_x$

FIG. 21 – Résultats expérimentaux — cas d'un objet quelconque — Maximisation de  $l$



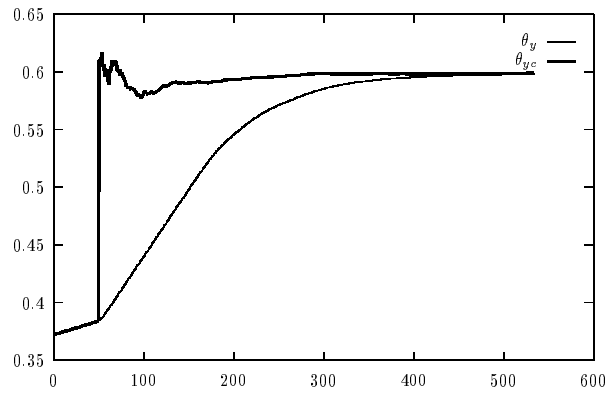


(a) Composantes du torseur cinématique (m/s ou rad/s) en fonction du nombre d'itérations

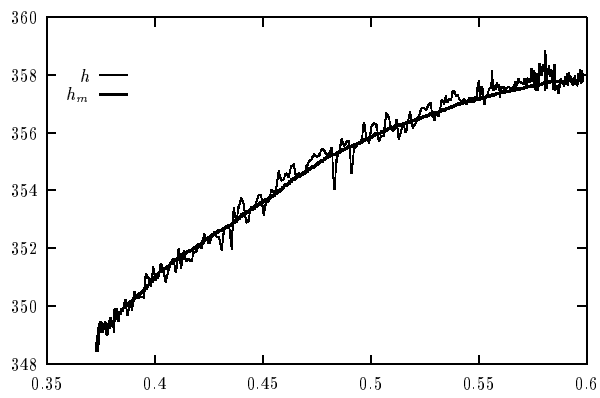


(b) Erreur dans le capteur (pix.) en fonction du nombre d'itérations

FIG. 22 – Résultats expérimentaux — cas d'un objet quelconque — Maximisation de  $h$

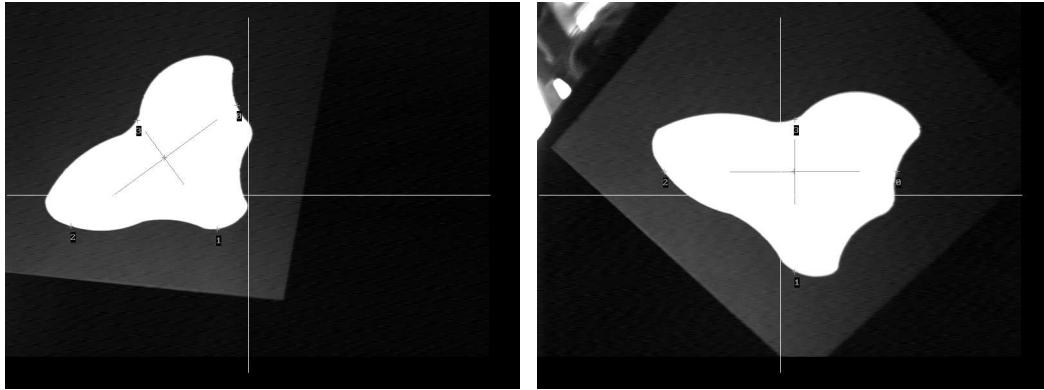


(a) Consigne angulaire (rad.) et position angulaire courante (rad.) en fonction du nombre d'itérations



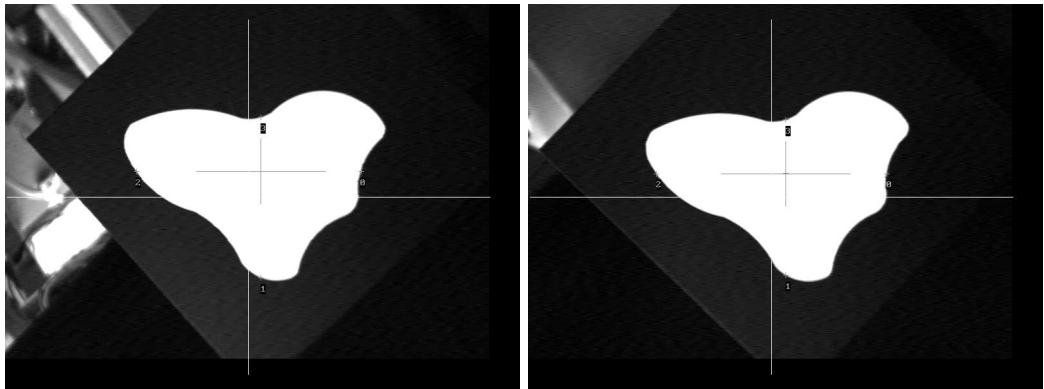
(b) Taille de l'image et taille modélisée (pix.) en fonction de  $\theta_y$

FIG. 23 – Résultats expérimentaux — cas où trois points de l'objet sont connus — Maximisation de  $h$



(a) Position initiale

(b) Position après centrage

(c) Position après maximisation de  $l$ 

(d) Position finale

FIG. 24 – Résultats expérimentaux — cas d'un objet quelconque —

Par ailleurs, plus particulièrement dans le cas où trois points caractéristiques existent, il serait intéressant d'évaluer dans le futur le gain éventuel d'une identification non linéaire en terme d'erreur de positionnement. Cette identification pourrait faire suite à la méthode présentée ici dès lors que l'erreur angulaire est suffisamment faible. La consigne angulaire pourrait être alors une bonne estimation initiale nécessaire à une optimisation non linéaire.

Une autre voie de recherche ayant toujours pour but l'extension du champ applicatif des asservissements visuels fera suite à ces travaux, elle concernera les cas où la consigne visuelle est cette fois supposée connue mais complexe.

## Références

- [1] M. Bertozzi and A. Broggi. Vision-based vehicle guidance. *IEEE Computer Magazine*, 7(30):49–55, July 1997.
- [2] Z. Bien, W. Jang, and J. Park. Characterisation and use of feature-jacobian matrix for visual servoing. In K. Hashimoto, editor, *Visual Servoing*, volume 7, pages 317–363. World Scientific, Singapore, 1993.
- [3] A. Castano and S. Hutchinson. Visual compliance: task-directed visual servo control. *IEEE Trans. on Robotics and Automation*, 10(3):334–341, June 1994.
- [4] F. Chaumette. *La relation vision-commande : théorie et application à des tâches robotiques*. Thèse de doctorat, Université de Rennes I, Rennes, Juillet 1990.
- [5] P. Coiffet. *La robotique : principes et applications*. Hermes, Paris, 1986. Traité des nouvelles technologies.
- [6] P.I. Corke. Video-rate robot visual servoing. In K. Hashimoto, editor, *Visual Servoing*, volume 7, pages 257–284. World Scientific, Singapore, 1993.
- [7] P.I. Corke. Visual control of robot manipulators — a review. In K. Hashimoto, editor, *Visual Servoing*, volume 7, pages 1–31. World Scientific, Singapore, 1993.
- [8] E. Coste-Manière, P. Couvignou, and P.K. Khosla. Visual servoing in the task-function framework: a contour following task. *Journal of Intelligent and Robotic Systems*, 12:1–21, 1995.
- [9] A. Crétual and F. Chaumette. Positioning a camera parallel to a plane using dynamic visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'97*, pages 43–48, Grenoble, Septembre 1997.
- [10] D.F. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.

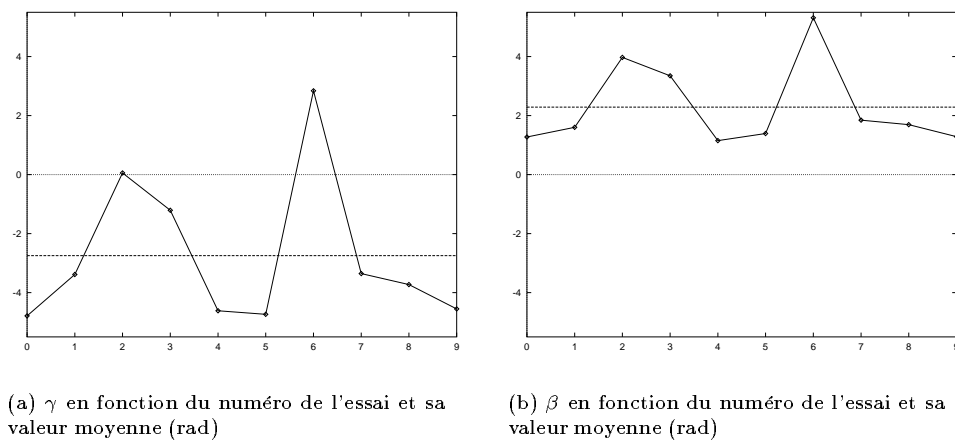


FIG. 25 – Résultats expérimentaux sur 10 essais — cas d'un objet quelconque —

$x$	$\bar{m} = \frac{1}{N} \sum_i x_i$	$\sup  x_i $	$\sigma$	$\inf  x_i $
$\gamma$	-2.75	4.79	3.28	0.05
$\beta$	2.29	5.31	1.35	1.15

TAB. 2 – Résultats expérimentaux sur 10 essais (en degrés) — cas d'un objet quelconque —

- 
- [11] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.
- [12] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell. Automatic selection of image features for visual servoing of a robot manipulator. In *Conf. IEEE Robotics and Automation*, pages 832–837, Washington, May 1989. IEEE Computer Society Press. Scottsdale, Arizona, USA.
- [13] J. T. Feddema and O. R. Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700, October 1989.
- [14] H. Hashimoto, T. Kubota, M. Kudou, and F. Harashima. Self-organizing visual servo system based on neural networks. *IEEE Control Systems*, April 1992.
- [15] J. Hill and W. T. Park. Real time control of a robot with a mobile camera. In *9th International Symposium on Industrial Robots*, pages 233–246, March 1979.
- [16] W. Jang and Z. Bien. Feature-based visual servoing of an eye-in-hand robot with improved tracking performance. In *IEEE International Conference on Robotics and Automation*, April 1991. Sacramento, California.
- [17] E. Marchand, F. Chaumette, and A. Rizzo. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS '96*, volume 3, pages 1083–1090, November 1996. Osaka, Japan.
- [18] W. T. Miller. Sensor-based control of robotic manipulators using a general algorithm. *IEEE Trans. on Robotics and Automation*, 3(2):157–165, April 1987.
- [19] N. P. Papanikolopoulos. Selection of features and evaluation of visual measurements during robotic visual servoing tasks. *Journal of Intelligent and Robotic Systems*, 13:279–304, 1995.
- [20] C. Samson, B. Espiau, and M. Le Borgne. Robot redundancy: an automatic control approach. In *NATO Advanced Research Workshop on Robots with Redundancy*, June 1988. Salo, Italia.
- [21] C. Samson, B. Espiau, and M. Le Borgne. *Robot control: the task function approach*. Oxford University Press, 1990.
- [22] A.C. Sanderson and L.E. Weiss. Image-based visual servo control using relational graph error signals. *Proc. IEEE*, pages 1074–1077, 1980.
- [23] A. Santos, B. Espiau, P. Rives, D. Simon, and V. Rigaud. Sensor-based control of holonomic autonomous underwater vehicles. Rapport de recherche 2609, INRIA, Juillet 1995.

- [24] Y. Shirai and H. Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5:99–108, 1973.
- [25] C. E. Smith and N. P. Papanikolopoulos. Grasping of static and moving objects using a vision-based control approach. *Journal of Intelligent and Robotic Systems*, 19:237–270, 1997.
- [26] I.H. Suh and T. W. Kim. Visual servoing of robot manipulators by fuzzy membership function based neural networks. In K. Hashimoto, editor, *Visual Servoing*, volume 7, pages 285–315. World Scientific, Singapore, 1993.
- [27] G. Wells, C. Venaille, and C. Torras. Promising research vision-based robot positioning using neural networks. *Image and Vision Computing*, 14:715–732, 1996.
- [28] M. Wilson. Developments in industrial robotics. *Computing and Control Engineering Journal*, 6(4):156–160, August 1995.



---

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY  
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN  
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

Éditeur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399