

# COOPÉRATION CAMÉRA EMBARQUÉE / CAMÉRA DÉPORTÉE POUR L'ASSERVISSEMENT VISUEL

Grégory Flandin, François Chaumette, Eric Marchand

IRISA - INRIA Rennes

Campus de Beaulieu, 35042 Rennes Cedex, France

prénom.nom@irisa.fr

**Résumé** - Il existe deux grands types d'architectures de contrôle d'un robot basées sur l'utilisation d'une caméra : la caméra est dite "embarquée" lorsqu'elle est rigidement liée à l'effecteur et dite "déportée" si elle observe le robot dans son espace de travail. Ces deux architectures présentent des différences techniques et surtout des avantages complémentaires par le caractère respectivement "local" et "global" de leur champ de vision. L'objectif de notre travail est de tirer partie de cette complémentarité dans un cadre coopératif. En effet, pour des tâches complexes en environnement naturel, nous ne pouvons plus nous limiter à l'utilisation d'une caméra embarquée : celle-ci, disposant d'une vue locale, n'a pas les moyens d'interagir avec tout son environnement contrairement à une caméra disposant d'une vue globale dont, en contrepartie, les faibles (voire inexistantes) capacités de manœuvres ne permettent pas d'explorer toutes les parties de la scène. Nous présentons un système intégrant ces deux architectures pour la réalisation simultanée de deux tâches. Dans un souci de robustesse, nous démontrons la stabilité de la loi de commande et comparons différentes techniques de coopération d'un point de vue expérimental.

**Mots clé** - coopération, asservissement visuel.

## 1 Introduction

Durant les quinze dernières années, l'asservissement visuel 2D embarqué a fait l'objet de nombreuses recherches [10, 3, 5, 7]. Au cœur de cette approche, réside le jacobien image  $L$  (ou matrice d'interaction). Il relie les variations de caractéristiques image  $s$  à la commande du robot généralement exprimée par son torseur cinématique  $T$  :  $\dot{s} = LT$ . Souvent, la loi de commande peut être exprimée comme régulant une fonction de tâche [9]. Si  $s^*$  est la valeur désirée de  $s$ , le but est de contrôler le robot de sorte que  $s - s^*$  décroisse vers 0. Une méthode simple pour y parvenir est d'appliquer la commande :

$$T = -\lambda \widehat{L}^+(s - s^*) \quad (1)$$

où  $\widehat{L}^+$  est la pseudo-inverse du jacobien estimé. Si  $\dim(s) \leq \dim(T)$ , on peut montrer que, lorsque  $\widehat{L}$  est recalculé à chaque itération, (1) assure une décroissance exponentielle de  $s - s^*$ . Mais, pour diverses raisons [1],  $\widehat{L}$  est souvent fixé à une matrice constante (calculée, en général, à la position désirée). Néanmoins, dans ce cas, la positivité des valeurs propres de  $L\widehat{L}^+$  assure la stabilité asymptotique locale de (1).

Dans [6], les résultats de [3] sont étendus au cas où la caméra est déportée. Soulignons le fait que, dans ce cas de figure, le jacobien image doit prendre en compte le passage du repère caméra au repère de commande du robot. Si l'on note  $[R, t]$  le déplacement correspondant ( $R$  étant la matrice de rotation et  $t$  le vecteur de translation), le jacobien déporté  $L_g$  est relié au jaco-

bien embarqué  $L$  par la relation :

$$L_g = -L \begin{pmatrix} R & -RS(-R^T t) \\ 0 & R \end{pmatrix} \quad (2)$$

où  $S(a)$  est la matrice antisymétrique associée au vecteur  $a$ . Dans [6], la commande est identique à (1).

Pour des tâches complexes en environnement naturel, nous ne pouvons plus nous limiter à l'utilisation d'un type de caméra. D'une part, la caméra locale (embarquée) ne peut pas interagir avec tout l'espace de travail et peut difficilement considérer des modifications importantes de son environnement. D'autre part, une caméra globale (déportée) n'est pas assez manœuvrable pour explorer la scène. Beaucoup d'articles traitent de l'utilisation de plusieurs caméras en vision par ordinateur : en général à des fins de reconstruction 3D par stéréovision, parfois pour de l'asservissement visuel basé sur une paire d'images [4, 6] mais rarement dans le cadre d'une coopération caméra globale / caméra locale [8]. Le travail que nous décrivons dans cet article constitue une première étape vers une telle coopération. Nous définissons et montrons la faisabilité d'une tâche d'asservissement visuel fusionnant une vue locale et une vue globale de la scène.

En section 2, nous décrivons et modélisons précisément les tâches que chacune des caméras doit accomplir et montrons, en section 3, comment les fusionner en une seule. Nous verrons que leur interaction naturelle ne peut pas être ignorée et décrivons comment la prendre en compte de manière efficace et robuste. En section 4, nous présentons des résultats obtenus avec un robot cartésien six axes.

## 2 Description et modélisation

Dans notre système (voir figures 1 et 2), la caméra globale est fixe et contrôle les degrés de translation pour assurer le positionnement de l'effecteur (section 2.1) alors que la caméra locale contrôle son orientation pour centrer une cible fixe dans son image (section 2.2).

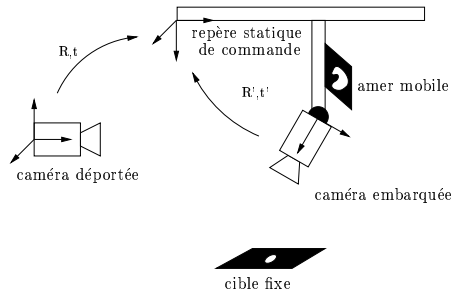


FIG. 1 – *Coopération caméra embarquée / caméra déportée*

L'amer mobile, d'où les caractéristiques de l'image globale sont extraites, est monté sur les organes de translation du robot. Ainsi les mouvements de rotation de l'effecteur n'ont pas d'influence sur l'image globale ce qui diminue l'interaction des deux tâches.

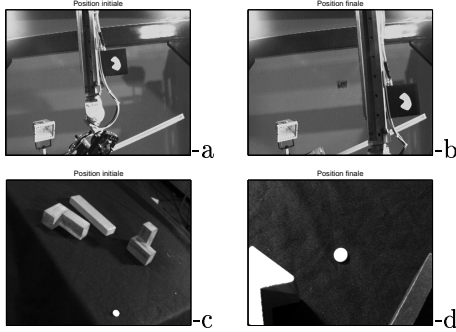


FIG. 2 – *Images initiales (à gauche) et désirées (à droite) pour le système déporté (ligne du haut) et celui embarqué (ligne du bas)*

### 2.1 Commande de translation

Pour assurer le positionnement de l'effecteur, l'image globale doit contrôler trois degrés de liberté. Nous devons donc extraire au moins trois caractéristiques indépendantes de l'image. Pour simplifier la démonstration de la stabilité, nous en choisissons uniquement trois : les coordonnées dans l'image du centre de gravité (c.d.g.) de l'amer observé et la surface projetée. Nous supposons, d'autre part, que l'amer est plan et parallèle au plan de l'image. En conséquence, le c.d.g. peut être considéré comme un point physique et son évolution est régie par l'équation :

$$\begin{pmatrix} \dot{X}_G \\ \dot{Y}_G \end{pmatrix} = \begin{pmatrix} \frac{1}{z_G} & 0 & -\frac{X_G}{z_G} \\ 0 & \frac{1}{z_G} & -\frac{Y_G}{z_G} \end{pmatrix} \begin{pmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{z}_G \end{pmatrix}$$

où  $(X_G, Y_G)$  sont les coordonnées du c.d.g. dans l'image à un mètre du centre optique et  $(x_G, y_G, z_G)$  sont les

coordonnées du c.d.g. de l'amer exprimées dans le repère de la caméra. D'autre part, l'évolution de la surface projetée  $S$  est donnée par :  $\dot{S} = -(2S/z_G)z_G$ . Les deux relations précédentes s'écrivent :

$$\begin{pmatrix} \dot{X}_G \\ \dot{Y}_G \\ \dot{S} \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{z_G} & 0 & -\frac{X_G}{z_G} \\ 0 & \frac{1}{z_G} & -\frac{Y_G}{z_G} \\ 0 & 0 & -\frac{2S}{z_G} \end{pmatrix}}_{L_0} \begin{pmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{z}_G \end{pmatrix} \quad (3)$$

Nous choisissons d'exprimer la commande de translation  $V$  dans le repère statique du robot. Le déplacement  $[R, t]$  du repère de la caméra embarquée au repère de commande est donc constant. Si l'on note  $s_T$  le vecteur de caractéristiques, les équations (2) et (3) conduisent à :

$$\dot{s}_T = L_T V = L_0 R V \quad (4)$$

$L_T$  est le jacobien de translation. La loi de commande associée, assurant une décroissance exponentielle de l'erreur, est donnée par :

$$V = -\lambda_T \widehat{L}_T^+ (s_T - s_T^*) \quad (5)$$

où  $s_T^*$  est le vecteur de caractéristiques désiré et  $\widehat{L}_T$  est le jacobien estimé. Dans le but de choisir convenablement  $\widehat{L}_T$ , nous proposons de déterminer le domaine de stabilité de la loi de commande (5) sous les hypothèses définies précédemment.

#### ◆ Démonstration de la stabilité

Pour tenir compte de la calibration interne de la caméra, nous devons rappeler la transformation mètres-pixels. Si nous ignorons les distorsions radiales et supposons que les axes de l'image sont parfaitement orthogonaux :

$$\begin{pmatrix} \dot{X}_G \\ \dot{Y}_G \\ \dot{S} \end{pmatrix}_{\text{pixels}} = \underbrace{\begin{pmatrix} \frac{f}{l_x} & 0 & 0 \\ 0 & \frac{f}{l_y} & 0 \\ 0 & 0 & \frac{f^2}{l_x l_y} \end{pmatrix}}_A \begin{pmatrix} \dot{X}_G \\ \dot{Y}_G \\ \dot{S} \end{pmatrix}_m \quad (6)$$

où  $f$ ,  $l_x$  et  $l_y$  sont respectivement la focale, la largeur et la hauteur d'un pixel en mètres. D'après (4) et (6), le jacobien global de translation est du type  $L = AL_0R$  où  $A$ ,  $L_0$  et  $R$  sont régulières. Nous devons étudier la positivité des valeurs propres de  $L\widehat{L}^+$  ( $=L\widehat{L}^{-1}$  dans le cas présent). En notant  $\hat{x}$  l'estimée de  $x$  et  $R = dR \cdot \widehat{R}$ , nous obtenons  $L\widehat{L}^{-1} = AL_0 dR \widehat{L}_0^{-1} \widehat{A}^{-1}$ . Nous proposons d'étudier deux cas :

#### 1. $R = \widehat{R}$ (i.e. $dR = Id$ )

Dans ce cas, les valeurs propres de  $L\widehat{L}^{-1}$  sont :

$$\left( \frac{f \widehat{z}_G \widehat{l}_x}{l_x z_G \widehat{f}}, \frac{f \widehat{z}_G \widehat{l}_y}{l_y z_G \widehat{f}}, \frac{f^2 \widehat{l}_y S \widehat{z}_G \widehat{l}_x}{l_x z_G S \widehat{f}^2 l_y} \right)$$

En conséquence, la stabilité asymptotique locale du système est assurée puisque la condition  $f, \widehat{f}, l_x, l_y, \widehat{l}_x, \widehat{l}_y, z_G, \widehat{z}_G, S$  et  $\widehat{S} > 0$  n'est pas restrictive et ce quelles que soient les estimations  $\widehat{X}_G$  et  $\widehat{Y}_G$ .

## 2. $A = \widehat{A}$ et $L_0 = \widehat{L}_0$

A présent,  $L\widehat{L}^{-1} = AL_0dRL_0^{-1}A^{-1}$  donc  $dR$  et  $L\widehat{L}^{-1}$  sont semblables et ont les mêmes valeurs propres. Nous devons simplement étudier la positivité des parties réelles des valeurs propres de  $dR$  qui sont  $(1, e^{i\theta}, e^{-i\theta})$  où  $\theta$  est l'angle de rotation entre  $R$  et  $\widehat{R}$ . Ceci signifie que, quel que soit l'axe de rotation, le système est localement asymptotiquement stable si

$$-\frac{\pi}{2} < \theta < \frac{\pi}{2} \quad \square$$

A en juger par cette étude de stabilité, nous pouvons nous permettre de fixer  $\widehat{L}_T$  à une valeur constante tout en assurant la convergence de la commande. Dans notre application, nous avons choisi de fixer  $\widehat{X}_G, \widehat{Y}_G$  et  $\widehat{z}_G$  à leur valeur désirée,  $\widehat{z}_G$  étant grossièrement "estimé à la main". La valeur constante de  $R$  est également grossièrement estimée.

## 2.2 Commande en rotation

L'objectif de la seconde tâche est de contrôler l'orientation de la caméra embarquée de telle sorte que le centre d'une cible fixe apparaisse au centre de l'image (voir figure 2). Cette tâche assure la visibilité de la cible durant le positionnement.

Si l'objet observé est un point alors le mouvement de sa projection  $(X, Y)$  est relié à la commande de rotation  $\Omega = [\Omega_X, \Omega_Y, \Omega_Z]$  par :

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \underbrace{\begin{pmatrix} XY & -(1+X^2) & Y \\ 1+Y^2 & -XY & -X \end{pmatrix}}_{L_R} \Omega$$

$L_R$  est le jacobien de rotation. Si  $s_R = (X, Y)$  représente le vecteur des caractéristiques embarquées courantes,  $s_R^*$  celui désiré et  $\widehat{L}_R$  le jacobien estimé, la commande de suivi est donnée par :

$$\Omega = -\lambda_R \widehat{L}_R^+ (s_R - s_R^*) \quad (7)$$

Comme précédemment, nous pouvons montrer que le domaine de stabilité de cette commande est très large. Dans notre application,  $\widehat{L}_R$  est calculé une fois pour toutes en posant  $\widehat{X} = 0$  et  $\widehat{Y} = 0$ . Dans ce cas, une condition suffisante de stabilité asymptotique locale est  $f, \widehat{f}, l_x, \widehat{l}_x, l_y, \widehat{l}_y > 0$ . Le calcul  $\widehat{L}_R$  à chaque itération serait également très simple à implémenter.

## 2.3 Interaction

Alors que les mouvements de rotation n'influencent pas les caractéristiques de l'image globale, les mouvements de translation de l'effecteur induisent des mouvements dans l'image embarquée. Plus précisément, le mouvement de la projection de la cible fixe est relié à la commande en translation par :

$$\dot{s}_R = \underbrace{\begin{pmatrix} -1/z & 0 & X/z \\ 0 & -1/z & Y/z \end{pmatrix}}_{L_{RT}} R' V$$

où  $z$  est la profondeur de la cible fixe dans le repère de la caméra locale et  $R'$  est la rotation entre ce repère et le repère de commande en translation. Pour estimer  $\widehat{L}_{RT}$ , nous fixons  $\widehat{X}, \widehat{Y}$  à leur valeurs désirées 0 et  $\widehat{z}$  à une valeur constante  $z^*$ .  $R'$  est calculé grâce à l'odométrie du robot.

## 3 Coopération

A présent, nous souhaitons faire coopérer les tâches de rotation et de translation dans un schéma de commande unifié. D'après la modélisation précédente, le jacobien de la tâche globale est :

$$\widehat{L} = \begin{pmatrix} \widehat{L}_T & 0 \\ \widehat{L}_{RT} & \widehat{L}_R \end{pmatrix} = \begin{pmatrix} \widehat{L}_d \\ \widehat{L}_e \end{pmatrix} \left. \begin{array}{l} \text{déporté} \\ \text{embarqué} \end{array} \right\}$$

D'une part, utiliser la commande (1) où  $\widehat{L}$  présente un terme de couplage ( $\widehat{L}_{RT} \neq 0$ ) conduirait à des comportements indésirables dans l'image avec, éventuellement, une perte des amers si  $\widehat{L}$  n'était pas parfaitement estimé. Ces craintes ont pu être vérifiées au cours d'expérimentations ; c'est pourquoi nous souhaitons conserver le maximum d'indépendance dans la commande des tâches. De plus, si l'indépendance est préservée, la stabilité de la tâche globale sera assurée tant que chacune d'elles reste stable.

Mais d'autre part, fixer  $\widehat{L}_{RT} = 0$  introduit une imprecision de modélisation qui conduit à des erreurs de suivi (figure 3-b). Le problème est d'autant plus dangereux que les amers peuvent sortir de l'image. Afin de supprimer cette erreur de suivi tout en conservant l'indépendance, nous nous proposons de comparer deux types de méthodes :

### 1-Estimation de la perturbation

La translation de l'effecteur peut être vue comme une perturbation inconnue agissant sur les caractéristiques embarquées. A cause de cette perturbation inconnue sur  $s_R$ , nous devons écrire :

$$\dot{s}_R = \frac{\partial s_R}{\partial r} \frac{dr}{dt} + \frac{\partial s_R}{\partial t} = L_R \Omega + \frac{\partial s_R}{\partial t}$$

où  $r$  symbolise la position de l'effecteur. Si l'on souhaite assurer une décroissance exponentielle de  $s_R - s_R^*$ , la commande correspondante est donnée par [2] :

$$\Omega = \underbrace{-\lambda_R \widehat{L}_R^+ (s_R - s_R^*)}_{\text{loi de commande classique}} \quad \underbrace{-\widehat{L}_R^+ \frac{\partial s_R}{\partial t}}_{\text{compensation de l'erreur de traînage}} \quad (8)$$

Il faut donc estimer la part des variations de l'image due à la perturbation, c'est à dire  $\partial s_R / \partial t$ . Plusieurs techniques permettent de réaliser cette estimation :

**Odométrie :** Si l'on a accès à l'odométrie du robot,  $V$  peut être mesurée et l'estimation est donnée par  $\left(\frac{\partial s_R}{\partial t}\right) = L_{RT} V$ . Malheureusement, peu de robots peuvent se permettre une mesure précise de leurs mouvements de translation.

**Estimation itérative :** Cette méthode nous vient directement de la théorie de la commande et consiste à insérer un intégrateur dans la boucle de commande. L'intégration est réalisée par le schéma itératif suivant :

$$\left(\frac{\partial s_R}{\partial t}\right)_{k+1} = \left(\frac{\partial s_R}{\partial t}\right)_k + \mu(s_{Rk} - s_R^*) \quad (9)$$

où  $\mu$  est un gain de compensation. Lorsque la stationnarité est atteinte,  $s_{Rk} = s_R^*$ .

**Observation/Prédiction :** Une autre méthode consiste à comparer la vitesse observée dans l'image et la vitesse prédite à partir de la commande précédente. La différence de ces deux termes doit correspondre à la perturbation. A l'instant  $k$ , l'estimation est :

$$\left(\frac{\partial s_R}{\partial t}\right)_k = \underbrace{\frac{s_{Rk}}{\Delta t}}_{\text{observation}} - \underbrace{\frac{s_{Rk/k-1}}{\Delta t}}_{\text{prédiction}}$$

où  $s_{Rk/k-1} = s_{Rk-1} + L_R \Omega_{k-1} \Delta t$ . Donc :

$$\left(\frac{\partial s_R}{\partial t}\right)_k = \frac{s_{Rk} - s_{Rk-1}}{\Delta t} - L_R \Omega_{k-1} \quad (10)$$

A l'instant  $k$ , la connaissance de  $\Omega_{k-1}$  peut être obtenue soit en supposant que  $\Omega_{k-1}$  est la commande précédente soit en la mesurant. Cette dernière technique est celle que nous avons implémentée puisque la première suppose un temps de réponse idéal du système. Notons que les mouvements de rotation sont toujours plus faciles à mesurer précisément que les mouvements de translation.

L'estimation donnée par (10) serait parfaite si les données n'étaient pas bruitées. Puisque ce schéma n'est pas itératif, il ne filtre pas le bruit sur les mesures de  $s_R$  et  $\Omega$ . Pour le filtrer, nous pouvons utiliser soit un simple filtre avec facteur d'oubli :

$$\left(\frac{\partial s_R}{\partial t}\right)_k = \alpha \left( \frac{s_{Rk} - s_{Rk-1}}{\Delta t} - L_R \cdot \Omega_{k-1} \right) + (1 - \alpha) \cdot \left(\frac{\partial s_R}{\partial t}\right)_{(k-1)} \quad (11)$$

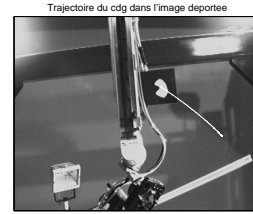
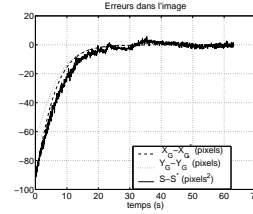
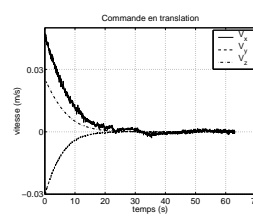
où  $\alpha$  est le facteur d'oubli, soit un filtre de Kalman comme celui proposé dans [2]. Evidemment, cette technique implique que l'on connaisse une approximation du modèle d'évolution de  $\frac{\partial s_R}{\partial t}$  mais il est bien connu que le filtre de Kalman est assez robuste vis à vis de cette modélisation. Nous avons choisi un modèle à vitesse constant et bruit corrélé.

Les techniques d'observation / prédiction peuvent résoudre le suivi de cible mobile. Dans ce cas, on ne fait pas de différence entre la perturbation due à la translation de l'effecteur et celle due au mouvement de la cible, mais ceci n'influence pas le comportement du système.

## 2-Redondance de tâches

Une autre méthode est de considérer la tâche de suivi

-a- Caméra déportée



-b- Caméra embarquée

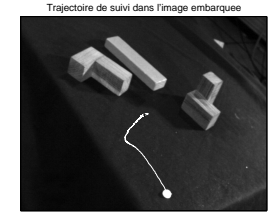
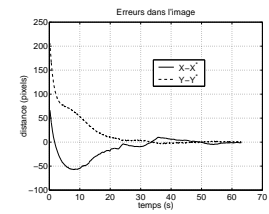
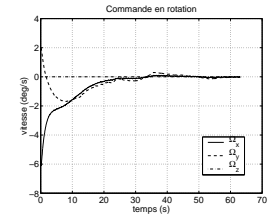


FIG. 3 – Commande indépendante des tâches

en rotation comme tâche principale. Puisqu'elle ne contraint pas tous les degrés de liberté du robot, nous pouvons utiliser le formalisme de la redondance de l'approche fonction de tâche [9, 3]. Une tâche secondaire  $g_s$  peut alors être achevée en régulant la fonction de tâche :

$$e = W^+(s_R - s_R^*) + (I - W^+W)\alpha_2 g_s^T$$

où  $W$  est une matrice de rang plein telle que  $\text{Ker } W = \text{Ker } \widehat{L}_e$  si on note  $\widehat{L}_e = [\widehat{L}_{RT} \quad \widehat{L}_R]$ . La matrice  $I - W^+W$  est donc un opérateur de projection sur le noyau de  $\widehat{L}_e$ . Cela signifie que la réalisation de la tâche secondaire n'aura pas d'effet sur la tâche principale tant que  $\widehat{L}_e$  est une bonne approximation de  $L_e$ .

Dans notre cas, le jacobien de la tâche principale  $L_e = (L_{RT} \quad L_R)$  est de rang plein donc nous pouvons poser  $W = \widehat{L}_e$ . De plus la tâche secondaire est une tâche de positionnement dans l'image déportée :  $g_s = C(s_T - s_T^*)$  où  $C$  est une matrice de combinaison. Par application de la commande  $T = -\lambda e$ , l'évolution de  $s_T$  lorsque  $s_R$  a convergé est donnée par  $s_T' = -\lambda \alpha_2 L_d (I - \widehat{L}_e^+ \widehat{L}_e) C (s_T - s_T^*)$ . Une condition nécessaire à la décroissance exponentielle de  $s_T$  est  $C = (\widehat{L}_d (I - \widehat{L}_e^+ \widehat{L}_e))^+$ . En conclusion, la fonction de tâche s'écrit :

$$e = \widehat{L}_e^+ (s_R - s_R^*) + (I - \widehat{L}_e^+ \widehat{L}_e) (\widehat{L}_d (I - \widehat{L}_e^+ \widehat{L}_e))^+ \alpha_2 (s_T - s_T^*)$$

## 4 Résultats expérimentaux

Cette section et les figures 3,5 montrent les résultats expérimentaux obtenus pour chacune des méthodes citées. Les gains de commande ont été choisis de la façon

suivante :  $\lambda_T = \lambda_R = 0.1$ . L'amer mobile d'où l'on extrait les caractéristiques déportées  $s_T$  est la tâche de forme complexe et inconnue que l'on aperçoit figure 2-a.  $s_T^*$  a été préalablement appris (figure 2-b) mais aurait tout aussi bien pu être fixé par un opérateur.

Lorsque l'on contrôle chaque tâche indépendamment, sans compensation (voir figure 3), la translation adopte un comportement exponentiel alors que la rotation ne peut pas s'achever tant que la translation n'a pas convergé. Ceci correspond à l'erreur de traînage.

### Estimation et compensation de l'erreur

Pour toutes ces méthodes, la commande en translation ne change pas et ses performances sont identiques à celles présentées figure 3.

Les résultats obtenus en utilisant un intégrateur sont présentés figure 5-a. Le gain était fixé à  $\mu = 0.01$ . Cette méthode améliore la vitesse de convergence mais  $\mu$  est assez difficile à ajuster. Les performances ne sont pas optimales du fait des variations de vitesse.

L'utilisation du filtre (11) pour lequel nous avons fixé  $\alpha = 0.7$  laisse apparaître un bruit résiduel significatif (figure 5-b) sur la commande même si l'effet de ce bruit sur les caractéristiques image est naturellement filtré par la dynamique du robot. Le filtre de Kalman a, quant à lui, de très bonnes performances (figure 5-c). Enfin, l'utilisation de l'odométrie semble également très efficace (figure 5-d) mais ses performances dépendent principalement de la précision des mesures du robot.

### Formalisme de la redondance

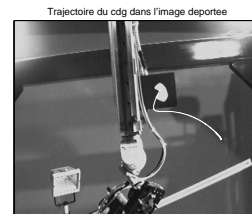
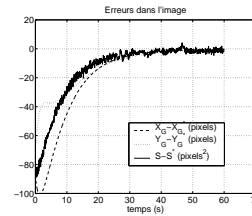
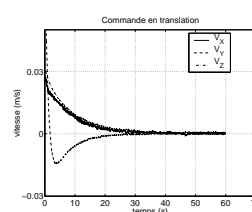
Le formalisme de la redondance présente l'avantage de spécifier l'importance d'une tâche par rapport à l'autre. La figure 4 montre que la tâche secondaire suit une convergence exponentielle à partir du moment où la tâche primaire a convergé. La réalisation de la tâche primaire semble influencée, au cours des premières itérations, par la tâche secondaire dans la mesure où son noyau n'est pas parfaitement estimé.

## 5 Conclusion

Nous avons développé un schéma de coopération intégrant une caméra fixe et une caméra mobile. En conservant l'indépendance entre les tâches, nous avons démontré la stabilité de la loi de commande associée. La plupart des techniques d'asservissement visuel ont été adaptées à la coopération et comparées expérimentalement. Toutes ont montré leur capacité à résoudre le problème considéré mais quelques une telles que le filtre de Kalman ou l'odométrie, se sont avérées plus efficaces en terme de robustesse.

L'application que nous avons développée constitue une base solide pour la construction de tâches de niveau supérieur. À l'avenir, nous souhaitons développer une stratégie d'exploration en environnement inconnu et nous intéresser à des problèmes tels que : construire un modèle de représentation de la connaissance permettant de fusionner des données incertaines ou partielles, planifier les mouvements de la caméra mobile dans le but d'améliorer la connaissance tout en évitant

### -a- Caméra déportée



### -b- Caméra embarquée

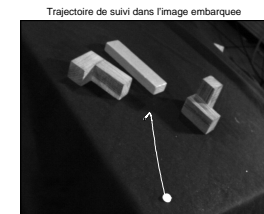
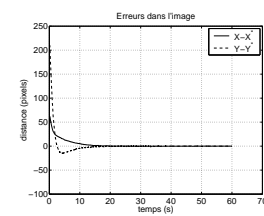
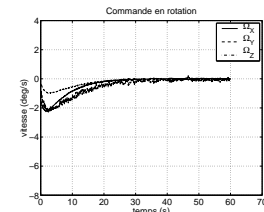


FIG. 4 – Utilisation du formalisme de la redondance

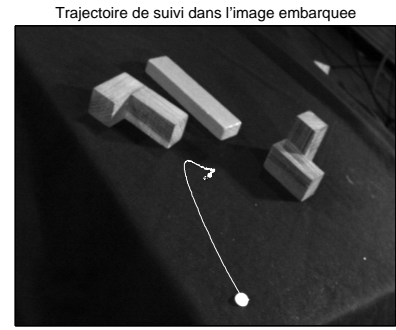
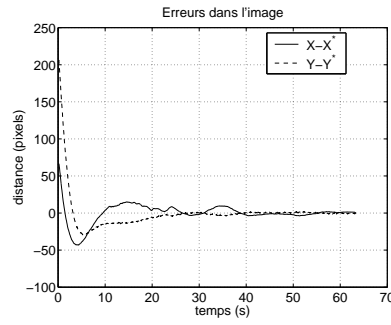
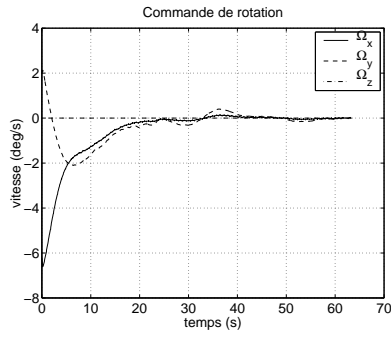
les obstacles, prendre des décisions en fonction de la connaissance actuelle et de la tâche à accomplir.

**Remerciements.** Cette étude a reçu le soutien du projet LARA de l'INRIA et de la région Bretagne.

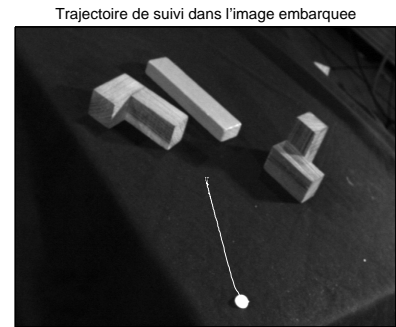
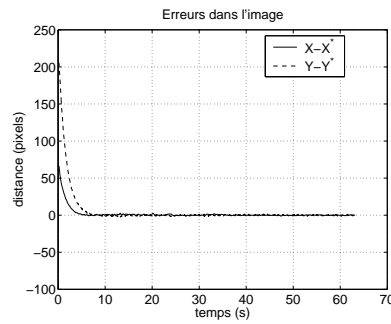
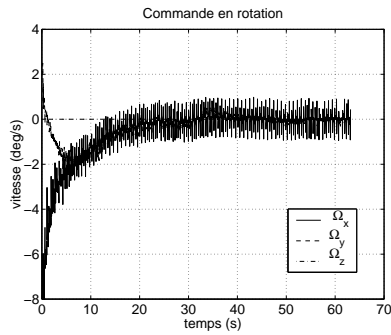
## Références

- [1] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and A.S. Morse, editors, *The Confluence of Vision and Control*, pp 66–78. LNCIS Series, No 237, Springer-Verlag, 1998.
- [2] F. Chaumette and A. Santos. Tracking a moving object by visual servoing. In *12th IFAC World Congress*, volume 3, pp 643–648, Sidney, Australia, July 1993.
- [3] 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.
- [4] G.D. Hager. A modular system for robust positioning using feedback from stereo vision. *IEEE Trans. on Robotics and Automation*, 13(4):582–595, August 1997.
- [5] K. Hashimoto. *Visual Servoing: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapor, 1993.
- [6] R. Horaud, F. Dornaika, and B. Espiau. Visually guided object grasping. *IEEE Trans. on Robotics and Automation*, 14(4):525–532, August 1998.
- [7] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.
- [8] E. Marchand and G. Hager. Dynamic sensor planning in visual servoing. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pp 1988–1993, Belgium, May 1998.
- [9] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.
- [10] L.E. Weiss. Dynamic visual servo control of robots. An adaptive image based approach. Technical Report CMU-RI-TR-84-16, Carnegie-Mellon University, April 1984.

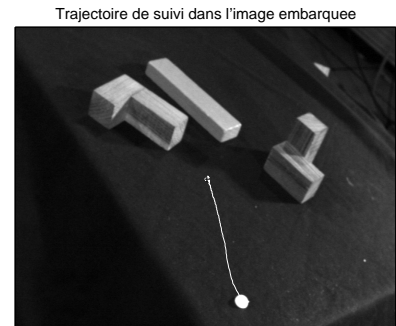
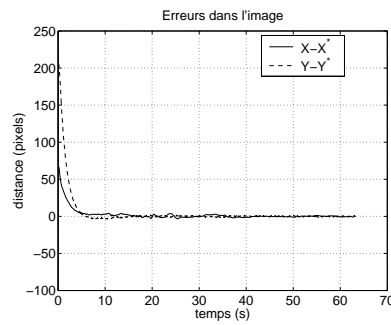
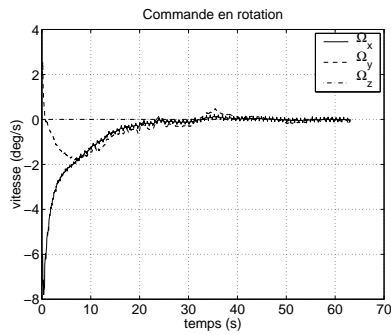
-a- Compensation par intégration



-b- Compensation par estimation + filtrage simple



-c- Compensation par estimation + filtrage de Kalman



-d- Compensation par odometrie

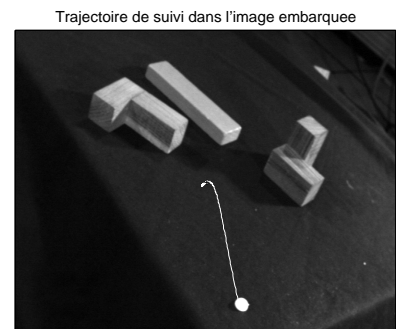
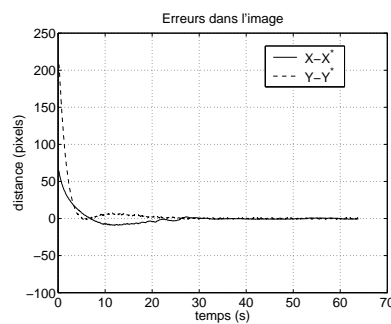
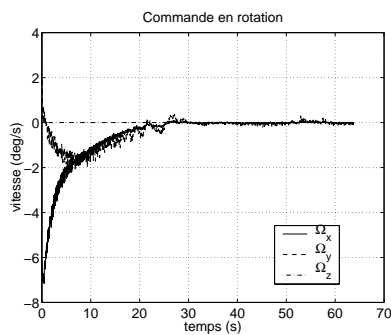


FIG. 5 – Tâche de suivi utilisant une estimation de  $\frac{\partial s_R}{\partial t}$