

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE MOULOU MAMMERI, TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT AUTOMATIQUE

## MEMOIRE DE MAGISTER

en Automatique

Option: **Traitement d'Images et Reconnaissance de Formes**

présenté par

**TOUZOUTI Nassima**

*Ingénieur U.M.M.T.O.*

## Commande prédictive visuelle d'un bras manipulateur

Mémoire soutenu le 01 /03 /2015 devant le jury d'examen composé de :

**DJENNOUNE Saïd**, Professeur à l'UMMTO .....Président.  
**DIAF Moussa**, Professeur à l'UMMTO .....Rapporteur.  
**LAGHROUCHE Mourad**, Professeur à l'UMMTO .....Examinateur.  
**MAIDI Ahmed**, Maitre de conférences A à l'UMMTO .....Examinateur.  
**TOUAT Mohand Ouachour**, Maitre de conférences B à l'UMMTO ...Examinateur.

A ma mère, à ma  
mère, à ma mère,  
à mon père.

# Remerciements

Remercier est une façon unique de regarder autour de soi pour mieux se rendre compte des personnes qui nous entourent.

Je tiens à remercier les membres de jury. Plus qu'une marque de courtoisie, à mes yeux ils sont l'expression sincère de ma gratitude envers eux. Merci donc à DJENNOUNE Saïd, professeur à UMMTO, LAGHROUCHE Mourad, professeur à UMMTO, MAIDI Ahmed, maître de conférences A à UMMTO ainsi que TOUAT Mohand Achour, maître de conférences B à UMMTO, j'ai apprécié leurs remarques avisées et leurs conseils bienveillants.

Merci à LAGHROUCHE Mourad, professeur à l'UMMTO, qui m'a écouté, supporté et surtout pour son soutien moral.

Je souhaite également remercier OAUDEH Nourdine, chef de département robotique au Centre de Développement Technologique Avancé (CDTA), pour m'avoir accueilli pendant ce travail et pour m'avoir apporté sa confiance.

Il m'est difficile de dissocier dans ces remerciements CHILLALI Wardia et REZGUI Slimane MAA à l'UMMTO ainsi que MENSOURI Rachid, professeur à UMMTO qui n'ont pas épargné leurs efforts pour m'aider tout au long de cette période. Je leur dois beaucoup et même davantage.

Enfin je remercie ma famille qui m'a soutenu pendant cette période parfois cahotante. Finalement, je remercie mon tendre Soufiane qui a partagé avec moi (et de manière synchrone!) toutes les périodes de ce travail pendant lesquelles le moral oscille souvent comme les variations du baromètre dans les caraïbes.

## Résumé

Dans ce travail, l'asservissement visuel 2D est adressé par la commande prédictive non linéaire. Les tâches d'asservissement visuel sont formulées en un problème d'optimisation non linéaire dans le plan image. L'approche proposée, nommée commande prédictive visuelle, qui tient en compte dans les calculs et d'une manière simple, des contraintes 2D et 3D. En effet, la prédiction d'image sur un horizon fini joue un rôle central pour les grands déplacements. La prédiction d'image est obtenue grâce à un modèle. Nous distinguons un modèle global non linéaire et un modèle local basé sur la matrice d'interaction. Les avantages et les inconvénients des deux modèles sont cités. Finalement, des résultats de simulation sur un bras manipulateur à 6 degrés de liberté en utilisant le modèle local, sont présentés. La convergence du robot est obtenue par la résolution d'un problème d'optimisation, avec la prise en compte de contrainte de visibilité. Les résultats montrent l'efficacité de l'approche proposée en comparant avec l'asservissement visuel 2D classique.

**Mots clés-** L'asservissement visuel, la commande prédictive.

## Abstract

In this work, image-based visual servoing is addressed via nonlinear model predictive control. The visual servoing task is formulated into a nonlinear optimization problem in the image plane. The proposed approach, named visual predictive control, can easily and explicitly take into account 2D and 3D constraints. Furthermore, the image prediction over a finite prediction horizon plays a crucial role for large displacements. This image prediction is obtained thanks to a model. The choice of this model is discussed. A nonlinear global model and a local model based on the interaction matrix are considered. Advantages and drawbacks of both models are pointed out. Finally, simulations results for an eye-in-hand 6 d.o.f manipulator robot using a local model are presented. The convergence of robot motion has been obtained through nonlinear constraint optimization, which take into considerations the visibility constrain. Simulations present the capabilities and the efficiency of the proposed approach by a comparison with the classical image-based visual servoing.

**Key words-** Visual servoing, predictive control.



---

# Sommaire

<b>Introduction générale</b>	1
<b>Chapitre I. Commande référencée vision</b>	4
I. Introduction	4
II. La robotique	5
II.1 Définition	5
II.2 Les composants d'un système robotique	5
II.3 Les robots manipulateurs	6
II.3.1 Description	6
II.3.2 Transformation homogène	7
II.3.3 La vitesse d'un corps rigide	8
II.3.4 Les définitions usuelles en robotique	10
III. La commande référencée vision	11
III.1 Définition	11
III.2 Les primitives visuelles	13
III.2.1 Définition	13
III.2.2 Modèle sténopé d'une caméra	13
III.2.3 Le choix des informations visuelles	15
III.3 Classification des systèmes d'asservissement visuel	16
III.3.1 Les configurations robot/caméra	16
III.3.1.1 Configuration caméra déportée (eye-to-hand)	16
III.3.1.2 Configuration caméra embarquée (eye-in-hand)	16

III.3.2 Le type de mesure	17
III.3.2.1 Asservissement visuel 3D	17
III.3.2.2 Asservissement visuel 2D	18
III.3.2.3 Asservissement visuel hybride	19
III.3.3 Le type de commande	20
III.3.3.1 La commande cinématique	20
III.3.3.2 La commande dynamique	20
III.4 Les principales étapes de la mise en œuvre d'un asservissement visuel 2D	20
III.4.1.L'extraction des primitives visuelles	21
III.4.1.1 L'algorithme d'extraction des informations visuelles	21
III.4.1.2.La mise en correspondance	23
III.4.2 La détermination de ma matrice d'interaction	25
III.4.3 L'élaboration de la loi de commande	29
III.4.3.1 Le formalisme de la fonction de tâche	29
III.4.3.2 La commande cinématique	30
III.4.3.3 Le choix de la matrice de combinaison C	31
III.4.3.4 L'approximation de la matrice d'interaction	32
IV. Difficultés de mise en œuvre	33
V. Conclusion	34
<b>Chapitre II. Commande prédictive visuelle d'un bras manipulateur</b>	36
I. Introduction	36
II. Commande prédictive	37
II.1. Concept de base de la commande prédictive	37
II.2 Définition et principe	38
II.3 Les éléments de la commande prédictive	39
II.4 Le choix des horizons	41
II.4 1. Le choix de l'horizon de prédiction $H_p$	41
II.4.2. Le choix de l'horizon de commande $H_c$	41
III. La commande prédictive et l'asservissement visuel 2D	42
III.1 La commande prédictive visuelle (CPV)	42
III.2 La structure de la commande à modèle interne	45
III.3. Formulation mathématique	46
III.4. Le modèle de prédiction d'image	49
III.4.1 Le modèle global non linéaire	49

III.4.2 Le modèle local à base de la matrice d'interaction	51
III.5. Les étapes à suivre pour la mise en œuvre la CPV	52
IV. Conclusion	52
<b>Chapitre III. Commande prédictive visuelle pour un modèle local et les résultats de simulation</b>	54
I. Introduction	54
II. La structure de la commande à base d'AV 2D	55
III. La modélisation d'asservissement visuel 2D en boucle ouverte	55
III.1. Le modèle du dispositif de déplacement opérationnel virtuel	57
III.2. Le modèle du capteur visuel	58
IV. Le régulateur proportionnel	60
V. Le régulateur prédictif	62
VI. Les résultats de simulations	64
VI. 1. L'asservissement visuel 2D	65
VI.1.1. Les résultats obtenus en fonction de choix de la matrice d'interaction estimée $\hat{L}_s$	65
VI.1.2. L'inconvénient d'asservissement visuel 2D	69
VI.2. La résolution du problème	69
VI.2.1. Une combinaison entre les différents choix de la matrice d'interaction Estimée	69
VI.2.2. La commande prédictive visuelle	72
VII. Conclusion	74
<b>Conclusion générale</b>	75
<b>Bibliographie</b>	



---

# Liste des figures

I.1. Les composants d'un système robotique	5
I.2. Schématisation d'un système mécanique articulé	6
I.3. Asservissement en position	11
I.4. Asservissement en vitesse	11
I.5. Schéma de commande par asservissement visuel	12
I.6. Le modèle sténopé d'une caméra	14
I.7. Les configurations robot-caméra	17
I.8. Asservissement visuel 3D	18
I.9. Asservissement visuel 2D	18
I.10. Asservissement visuel 1/2D	19
I.11. Détection de points d'intérêt en utilisant le détecteur de Harris	23
I.12. Deux pixels $p_1$ et $p_2$ correspondants	24
I.13. La mise en correspondance entre deux images	24
I.14. Le couplage robot-image	26
II.1. Le principe de la commande prédictive	38
II.2. Les éléments de MPC	40
II.3. La structure de commande à modèle interne	45
II.4. Le principe de la prédiction d'image	48
III.1. La boucle fermée d'asservissement visuel 2D	54
III.2. Système de commande en asservissement visuel 2D	55
III.3. Le modèle VCMD linéaire	57

III.4. Les repères: base du robot, caméra et objet	58
III.5. Bloc de transformation homogène	59
III.6. Le modèle utilisé pour la prédiction	61
III.7. Le schéma bloc de la commande prédictive visuelle en utilisant un modèle local.	63
III.8. La situation de l'objet	64
III.9. Les résultats de simulations pour un AV 2D pour différents choix de $\hat{L}$	66
III.10. L'asservissement visuel 2D classique : situation (1) et situation (2)	69
III.11. L'asservissement visuel 2D classique avec la prise en compte de contrainte de visibilité	70
III.12. Les résultats de simulation en utilisant la commande prédictive visuelle	72

## Liste des tableaux

I.1. Les différents repères utilisés en robotique	8
---	---

## Liste des abréviations

### En Français

**AV.** Asservissement Visuel.

**2D.** Deux Dimensions.

**3D.** Trois Dimensions.

**CPV.** Commande Prédictive Visuelle.

### En Anglais

**IBVS.** Image Based Visual Servoing.

**PBVS.** Position Based Visual Servoing.

**MPC.** Model Predictive Control.

**VCMD.** Virtual Cartesian Motion Device.



---

# Introduction générale

Les machines, ou plus précisément les robots, font maintenant partie intégrante de notre vie. L'apparition des premiers robots était aux USA en 1960. Ils étaient cantonnés à des rôles d'automates mécaniques programmables où les trajectoires effectuées étaient apprises puis répétées quelque soit leur environnement. Cependant, le robot est souvent appelé à effectuer des tâches qui exigent une interaction avec son environnement. En absence de capteur adéquat, donnant une information sur cet environnement, le robot reste aveugle.

La volonté des chercheurs de reproduire les capacités humaines de perception et d'action dans les systèmes robotisés a conduit à l'intégration de données issues d'une caméra. L'utilisation d'une information visuelle en robotique, nécessite l'utilisation de la mesure extraite de l'image dans la boucle d'asservissement du robot. On parle alors d'*asservissement visuel* qui est une nouvelle discipline qui combine la vision par ordinateur, l'automatique et la robotique. C'est à Hill et Park [6] que l'on doit l'apparition du terme asservissement visuel (*visual servoing*).

L'asservissement visuel consiste à contrôler les mouvements d'un robot à partir d'un ensemble d'informations visuelles extraites des images acquises par une ou plusieurs caméras vidéos montées sur (ou observant) un robot. Il est classé suivant l'espace dans lequel s'effectue le contrôle du robot. Il existe trois approches: l'asservissement visuel 2D, l'asservissement visuel 3D et les asservissements visuels hybrides [17] [18] [1].

L'utilisation des techniques d'asservissement visuel fait, depuis plusieurs années, l'objet de nombreux travaux de recherche. Les domaines d'applications sont très variés: le domaine aérien [7] [8], le domaine médical [9] etc.

L'asservissement visuel 2D donne des résultats satisfaisants sur le plan stabilité et convergence lorsque l'erreur entre la position initiale et finale est faible [19]. Pour les grands déplacements ou pour le mouvement de rotation, le mouvement de la caméra peut entraîner la

sortie de certaines primitives visuelles de son champ visuel qui peut être désastreux pour la loi de commande.

Cependant, l'asservissement visuel 2D nécessite le calcul de la matrice d'interaction (le jacobien de l'image), pièce maîtresse de l'asservissement visuel qui exprime la relation entre les variations des primitives visuelles dans le plan image et le torseur cinématique de la caméra dans l'espace cartésien. Cette matrice peut être difficile à obtenir et son inverse pose quelques fois des problèmes de singularités [19].

Le point faible de l'asservissement visuel 2D c'est de ne pas prendre en considération des contraintes dans l'élaboration de la loi de commande. On distingue la contrainte 2D, appelées aussi la contrainte de visibilité, c'est garantir que les primitives visuelle restent dans le champ visuel de la caméra. Les contraintes 3D comme les limites de l'espace de travail et les contraintes mécaniques telles que les butées articulaires, ainsi que la limitation des actionneurs.

De nombreux travaux ont été faits pour résoudre ces cas critiques. Le premier consiste à choisir des primitives visuelles adéquates [19], ainsi que l'utilisation des lois de commande avancées telles que la commande adaptative [31] et la commande prédictive [32].

En effet, le but d'asservissement visuel 2D (réguler à zéro l'erreur entre les primitives courantes  $s(t)$  et les primitives de référence  $s^*$  dans le plan image) est formulé avec la prise en compte des contraintes en un problème d'optimisation, et lorsque le modèle du système est disponible, la stratégie de commande prédictive est bien adaptée pour résoudre ce genre de problème.

Cette approche s'appelle *la commande prédictive visuelle* qui est caractérisée par la prise en compte des contraintes dans le problème d'optimisation ainsi que la prédiction de l'image sur un horizon fini joue un rôle crucial pour réaliser des configurations difficiles. La prédiction de l'image est basée sur la connaissance d'un modèle. Il peut être *un modèle global non linéaire* qui combine le modèle du robot et de la caméra. La prédiction de l'image peut également être obtenue grâce à *un modèle local* en utilisant la matrice d'interaction.

C'est pourquoi l'utilisation de principe de la commande prédictive dans la tâche d'asservissement visuel 2D présente un fort potentiel que ce mémoire étudié à travers trois chapitres.

Le premier chapitre est consacré à décrire le principe d'asservissement visuel. Nous rappelons d'abord les notions de base en robotique et nous passerons ensuite vers la commande référencée vision. Nous présentons les différentes configurations entre un robot et un capteur visuel. Après énumération des techniques d'asservissement visuel, nous terminerons par une classification des différents types d'architectures existants pour la commande du robot et nous nous attardons sur l'asservissement visuel 2D.

Dans le deuxième chapitre, nous présentons brièvement la théorie de la commande prédictive ainsi que son principe de base. Nous passerons vers la présentation de la commande prédictive visuelle qui permet d'appliquer le principe de la commande prédictive sur les tâches d'asservissement visuel 2D, donc revient à résoudre un problème d'optimisation non linéaire sous contraintes dans le plan image. La prédiction de l'image est obtenue grâce à un modèle, le choix de ce dernier est discuté. Un modèle global non linéaire et un modèle local basé sur la matrice d'interaction sont considérés.

Le troisième chapitre est réservé à la méthode utilisée pour prédire les primitives visuelles futures dans le plan image. Nous présentons un modèle locale basé sur la matrice d'interaction. Premièrement, une structure de commande à base d'asservissement visuel 2D est présentée. Un régulateur proportionnel a été considéré. Nous passerons à la présentation d'un régulateur prédictif présenté par un modèle locale de prédiction ainsi qu'un bloc d'optimisation qui contient une fonction coût basée sur l'erreur dans le plan image ainsi que les contraintes. Le mouvement du robot est convergé vers la position désirée par la résolution d'un problème d'optimisation sous contrainte 2D. Les résultats de simulation permettent de comparer entre les résultats obtenus à partir de l'asservissement visuel 2D en appliquant une loi proportionnelle et celle en appliquant une loi prédictive.

La fin de mémoire est réservée à la conclusion sur l'ensemble des applications réalisées et aux perspectives de continuité.

---

# Commande référencée vision

## I. Introduction

La volonté des chercheurs de reproduire les capacités humaines de perception et d'action dans les systèmes robotisés a conduit à l'intégration de données issues de capteurs extéroceptifs, et plus particulièrement de celles issues d'une caméra. L'objectif affiché est que les capteurs visuels fournissent une information suffisamment riche pour permettre aux robots de réaliser, de manière autonome, des tâches dans des environnements partiellement connus, ou complètement inconnus.

Ce chapitre a pour but de décrire le principe général du processus d'utilisation de la vision dans la commande des robots. Ce processus est connu sous le nom d'*asservissement visuel*. Le chapitre est composé de deux parties: dans la première, nous rappelons des éléments de la robotique et dans la deuxième, nous focalisons sur la commande. Cette dernière permet d'évoquer les différentes configurations entre un robot et le capteur visuel. Après énumération des techniques d'asservissement visuel, nous terminerons par une classification des différents types d'architectures existants pour la commande du robot et nous nous attardons sur l'asservissement visuel 2D. Nous terminerons ce chapitre par une conclusion.

Ce chapitre n'a pas pour vocation de fournir un état de l'art exhaustif du domaine, le lecteur intéressé pourra consulter [1] pour un état de l'art jusqu'en 1995, [2] et [3] pour une synthèse des travaux effectués jusqu'en, respectivement, 2000 et 2003.

## II. Robotique

### II.1 Définition

Un robot est un dispositif mécanique chargé d'effectuer une ou plusieurs tâches d'une manière plus ou moins autonomes. Il perçoit son environnement et utilise cette information pour planifier une action dans le but d'atteindre un objectif fixé.

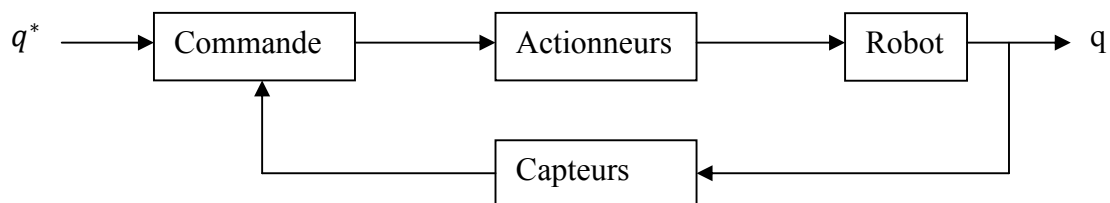
*La perception* est cruciale pour les robots. Ils déterminent les informations nécessaires pour l'accomplissement de la tâche. Elle peut être réalisée par deux types de capteurs tel que:

- *Capteurs proprioceptifs* permettent d'avoir une information sur l'état interne du robot: l'angle de l'articulation sur un bras manipulateur ou le nombre de tours de roue sur un robot mobile.
- *Capteurs extéroceptifs* permettent de connaître la situation du robot par rapport à son environnement.

*La planification*: le robot décide de l'itinéraire et les conditions de son bon accomplissement. Le robot doit *agir* et s'asservir sur les déplacements qu'il doit réaliser en vue de l'accomplissement de sa mission.

### II.2. Les composants d'un système robotique

Le système robotique est composé de quatre parties principales (figure I.1):



**Fig I.1.** Les composants d'un système robotique.

- Une *structure mécanique* qui sera le squelette du robot.
- Le second élément correspond *aux actionneurs* (servo-moteurs) qui vont permettre au robot d'effectuer réellement ses actions. Ces servo-moteurs seront commandés par la partie commande en interaction avec les informations transmises par les capteurs.
- La troisième partie correspond aux différents *capteurs* sensoriels équipant le robot pour une application particulière.
- Enfin *la partie commande* qui va permettre au robot d'analyser les données provenant

des capteurs et d'envoyer les ordres relatifs aux servo-moteurs. La partie commande est matérialisée physiquement par un micro-ordinateur.

Actuellement, les robots peuvent être classés en fonction de leurs caractéristiques de mobilité, avec une base fixe ce sont *les robots manipulateurs* et *des robots mobiles* pour une base mobile. La plus grande classe des robots, ce sont les bras manipulateurs qui seront étudiés dans ce manuscrit. Ils font des tâches répétitives mais souvent précises. Ces robots envahissent de nombreux domaines de l'automatisation industrielle moderne et sont principalement concernés par des tâches exécutées dans un environnement structuré.

## II.3. Les robots manipulateurs

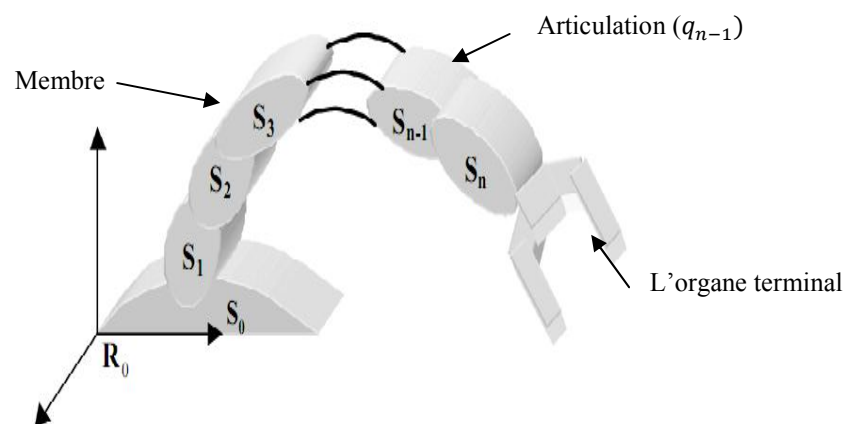
### II.3.1 Description

Un robot manipulateur est un système mécanique poly-articulaire composé de  $n$  corps rigides  $S_i$ ,  $i=1, \dots, n$ , appelés communément *membres*, reliés les uns aux autres en série ou en parallèle par l'intermédiaire d'*articulations* parmi lesquelles on distingue communément:

- *les articulations angulaires* caractérisées par un mouvement de rotation,
- *les articulations prismatiques* caractérisées par un mouvement de translation.

Le but de cet ensemble mécanique est de mener l'*organe terminal* du robot vers un lieu géométrique imposé par la tâche. Une structure mécanique articulée est schématiquement représentée en figure I.2.

A la base de la chaîne articulée est associé un *système de coordonnées de référence*  $R_0$  pour le robot. Le membre final est équipé d'un *effecteur* (organe terminal) adapté à l'exécution des tâches.



**Fig I.2.** Schématisation d'un système mécanique articulé.

### II.3.2. Transformation homogène

Dans ce manuscrit, l'espace de travail du robot est représenté par  $\mathcal{T}$ , qui est l'ensemble de positions et d'orientations qui peuvent être atteints par l'effecteur du robot et il est de dimension égale à  $m$ . L'effecteur du robot est un corps rigide qui se déplace arbitrairement dans un espace de travail à trois dimensions, alors dans le cas général  $\mathcal{T} = SE^3 = \mathfrak{R}^3 \times SO^3$  qui représente le produit de l'espace des coordonnées de positions  $\mathfrak{R}^3$  avec le groupe des rotations  $SO^3$ , alors  $\mathcal{T}$  de dimension  $m=6$ . Mais sa dimension peut être inférieure dans certaines applications où l'espace de travail est un sous-espace de  $SE^3$ . Par exemple, si nous considérons une tâche qui nécessite seulement des mouvements de translations (par exemple saisir et placer un objet), alors  $\mathcal{T} = \mathfrak{R}^3$  et  $m=3$ .

Typiquement, les tâches robotiques sont spécifiées par rapport à un ou plusieurs repères. Par exemple, une caméra peut fournir des informations sur la localisation d'un objet par rapport à son repère, tandis que la configuration utilisée pour saisir l'objet peut être spécifiée par rapport à un système de coordonnées attaché à l'objet.

Nous représentons les coordonnées d'un point  $P$  par rapport au repère  $i$  par la notation  $P^i$ . Étant donné deux repères,  $i$  et  $j$ , la matrice de rotation qui représente l'orientation de repère  $j$  par rapport au repère  $i$  est notée par  $R_j^i$ . La position de l'origine du repère  $j$  par rapport au repère  $i$  est notée par le vecteur  $t_j^i$ . La position et l'orientation d'un repère déterminent ce qu'on appelle *situation* et notée  $T_j^i = \begin{bmatrix} R_j^i & t_j^i \\ 0 & 1 \end{bmatrix}$ . Le repère monde est considéré si l'exposant principal,  $i$ , n'est pas spécifié.

Nous pouvons aussi utiliser la situation pour spécifier la transformation homogène qui est utilisée pour désigner le changement de coordonnées d'un point. En particulier, si nous donnons  $P^j$  (les coordonnées d'un point  $P$  par rapport au repère  $j$ ), et  $T_j^i$ , nous obtenons les coordonnées de  $P$  par rapport au repère  $i$  par l'application de la règle de transformation homogène:

$$P^i = T_j^i(P^j) \quad (I.1)$$

$$= R_j^i P^j + t_j^i \quad (I.2)$$

Dans la suite, nous allons utiliser la notation  $T_j^i$  pour référer soit à une transformation homogène, soit à une situation qui est spécifiée respectivement par une matrice de rotation et

un vecteur de translation  $R_j^i$  et  $t_j^i$ . Nous devons faire plusieurs transformations pour obtenir le changement de coordonnées désiré:

$$P^i = T_j^i(P^j) \quad (\text{I.3})$$

$$= T_j^i(T_k^j(P^k)) \quad (\text{I.4})$$

$$= (T_j^i T_k^j)(P^k) \quad (\text{I.5})$$

$$= T_k^i(P^k) \quad (\text{I.6})$$

La matrice de rotation et de translation correspondant sont données par:

$$R_k^i = R_j^i R_k^j \quad (\text{I.7})$$

$$t_k^i = R_j^i t_k^j + t_j^i \quad (\text{I.8})$$

Les repères qui sont nécessaires en robotique sont fréquemment présentés par les exposants/indices suivants:

e	Le repère attaché à l'effecteur du robot
o	Le repère attaché à l'objet
0	Le repère attaché à la base du robot
c	Le repère attaché à la caméra

**Tab I.1.** Les différents repères utilisés en robotique

Quand  $\mathcal{T} = SE^3$ ,  $x_e \in \mathcal{T}$  représenter la situation de l'effecteur par rapport au repère monde. Dans ce cas, nous préférons souvent paramétrer une situation en utilisant un vecteur de translation et trois angles de rotation (les angles d'Euler; roulis, tangage et lacet [4]). Il est souvent pratique de représenté la situation par le vecteur  $r \in \mathfrak{R}^6$ , plutôt que  $x_e \in \mathcal{T}$ . Cette notation peut être facilement adaptée au cas où  $\mathcal{T} \subseteq SE^3$ . Par exemple, quand  $\mathcal{T} = \mathfrak{R}^3$ , nous allons paramétrer l'espace de travail par  $r = [x, y, z]^T$ .

### II.3.3. La vitesse d'un corps rigide

Dans cette section, nous présentons brièvement la vitesse d'un corps dans l'espace de travail. Considérant que l'effecteur du robot se déplace dans l'espace de travail avec  $\mathcal{T} = SE^3$ . Le mouvement est décrit par la vitesse de rotation  $\Omega(t) = [\omega_x(t), \omega_y(t), \omega_z(t)]^T$  et la vitesse de translation  $v(t) = [v_x, v_y, v_z]^T$  par rapport au repère base. Soit P un point rigidement lié à

l'effecteur du robot, ces coordonnées par rapport au repère base sont  $[x, y, z]^T$ . La dérivée de coordonnées de P par rapport au temps, exprimées dans le repère base sont données par:

$$\dot{x} = z\omega_y - y\omega_z + v_x \quad (\text{I.9})$$

$$\dot{y} = x\omega_z - z\omega_x + v_y \quad (\text{I.10})$$

$$\dot{z} = y\omega_x - x\omega_y + v_z \quad (\text{I.11})$$

Qui peuvent être écrit sous forme vectorielle comme suit:

$$\dot{P} = \Omega \times P + v \quad (\text{I.12})$$

Cela peut être écrit de manière concise sous forme matricielle en notant que le produit vectoriel peut être représenté en termes d'une matrice anti-symétrique:

$$sk(P) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \quad (\text{I.13})$$

Ce qui nous permet d'écrire:

$$\dot{P} = -sk(P)\Omega + v \quad (\text{I.14})$$

La vitesse de translation  $v$  et de rotation  $\Omega$  définissent ce qui est connu en robotique *le torseur cinématique*:

$$\dot{r} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (\text{I.15})$$

Notez que  $\dot{r}$  représente également la dérivée de  $r$  lorsque la matrice de rotation,  $R$ , est paramétrée par l'ensemble des rotations autour des axes d'Euler.

Définir la matrice 3x6  $A(P) = [I_3] - sk(P)$  où  $I_3$  représente la matrice identité 3x3. Alors (I.14) peut être réécrit comme suit:

$$\dot{P} = A(P)\dot{r} \quad (\text{I.16})$$

Nous supposons maintenant un point exprimé par rapport à l'effecteur du robot,  $P^e$ , et nous voulons déterminer le mouvement de ce point par rapport au repère base lorsque le robot est en mouvement. Combinant (I.1) et (I.16), nous obtenons:

$$\dot{P} = A(T_e^b(P^e))\dot{r} \quad (\text{I.17})$$

### II.3.4. Les définitions usuelles en robotique

Nous ne faisons dans ce paragraphe qu'une présentation simplifiée des notions de base en robotique, pour plus de détails nous invitons le lecteur à consulter [5].

*La situation* de l'effecteur (position et orientation) par rapport au repère de référence peut être représentée par six variables, appelées *variables opérationnelles* et notées  $r$ , sont nécessaires et suffisantes pour représenter une situation: trois pour la position et trois pour l'orientation.

*L'espace articulaire* décrit les positions  $q$  (*variables articulaires*) qui fournissent les positions respectives de chaque axe par rapport à une position de référence. La dimension  $m$  de cet espace est appelée le nombre de degrés de liberté (ddl) du robot.

Chaque association articulation-membre constitue *un degré de liberté* du robot manipulateur qui est défini comme étant le nombre de paramètres indépendants capables de représenter l'ensemble des situations que peut atteindre son effecteur. Il dépend du nombre d'axes du robot, donc il varie de un à six.

Les mouvements d'un robot sont produits en appliquant *de forces* ou *de couples*, notés  $F$ , qui sont transmis aux différents axes du robot par l'intermédiaire d'actionneurs. Les effets de ces actionneurs sur l'état interne du robot sont mesurés par des capteurs proprioceptifs. Ces états sont représentés par les variables articulaires.

$q$  et  $r$  sont reliés par *un modèle géométrique direct*:  $r = MG(q)$  qui permet de calculer la situation de l'effecteur correspondant à une position quelconque  $q$ . Il dépend des caractéristiques du robot (longueur des bras, diamètre des roues, etc.). *Le modèle géométrique inverse* permet d'exprimer les variables articulaires en fonction de la situation de l'organe terminal.  $r^*$  ( $q^* = MGINV(r^*)$ ). Le schéma d'un asservissement classique en position dans l'espace opérationnel est représenté par la figure I.3.

*Le modèle cinématique direct* d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles (torseur cinématique) en fonction des vitesses articulaires  $\dot{r} = J(q)\dot{q}$  où  $J(q) = \frac{\partial r}{\partial q}$  désigne *le jacobien* du manipulateur. L'objectif du *modèle cinématique inverse* est de calculer, à partir d'une configuration  $q$  donnée, les vitesses articulaires  $\dot{q}$  qui assurent à l'organe terminal une vitesse opérationnelle  $\dot{r}$  imposée.

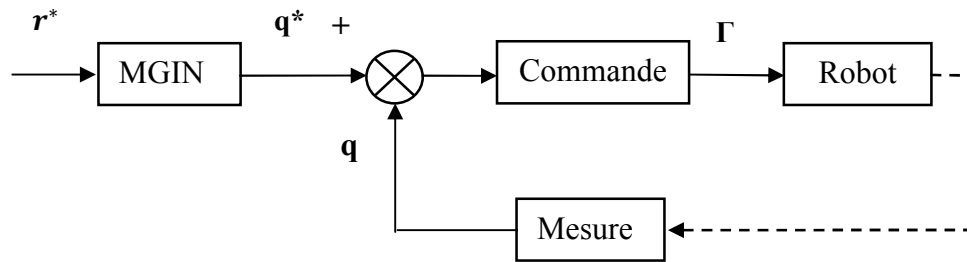


Fig I.3. Asservissement en position.

Pour réaliser un asservissement en vitesse dans l'espace opérationnel, on utilise, quand cela est possible, *le jacobien inverse*  $J^{-1}(q)$  (figure I.4).

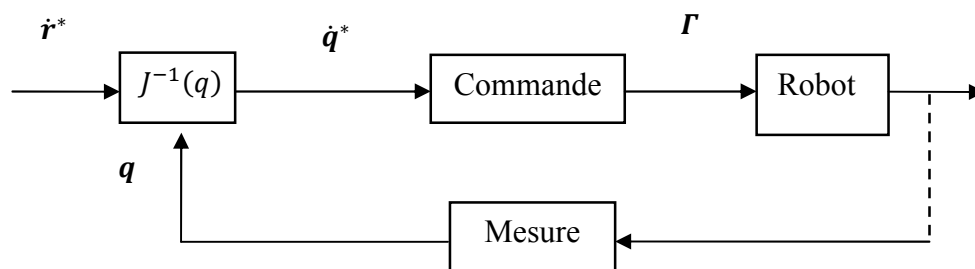


Fig I.4. Asservissement en vitesse.

### III. La commande référencée vision

#### III.1. Définition

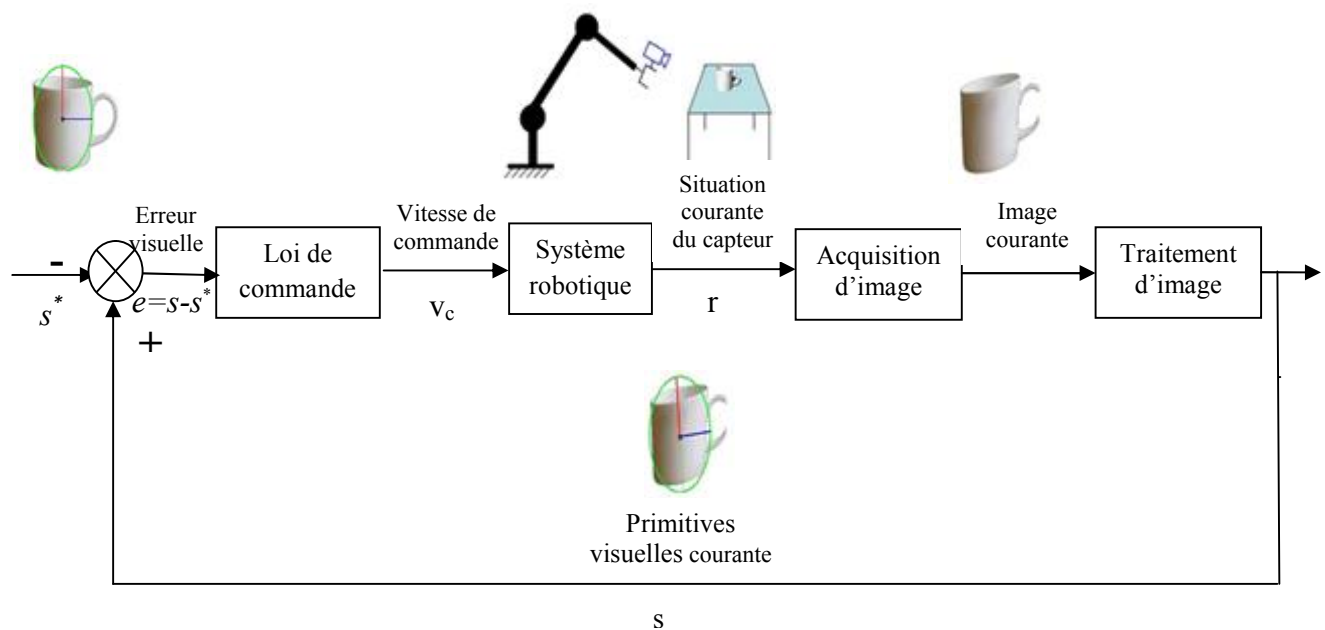
Par définition, *la commande référencée vision* ou *asservissement visuel AV* (*Visual servoing* en anglais) consiste à contrôler les mouvements d'un système robotique en intégrant les informations fournies par un capteur de vision dans un système de commande en boucle fermée. La figure I.5 expose le schéma classique de la commande incluant les différents maillons de la chaîne de traitement:

$s^*$ : le vecteur de *primitives visuelles de références* qui est obtenu à partir de l'image de référence.

$s$ : le *vecteur des primitives courantes* obtenu à partir de l'acquisition de l'image courante.

$e = s - s^*$ : erreur visuelle est la différence entre le vecteur de primitives courantes et le vecteur des primitives de références.

$v_c$ : le signal de commande ou la vitesse de référence de déplacement de la caméra qui est envoyé au *système robotique* dans le but de déplacer son effecteur c'est-à-dire avoir *une nouvelle situation de la caméra*.



**Fig I.5.** Schéma de commande par asservissement visuel

Après l'acquisition de la nouvelle image, elle va passer par *un traitement* dans le but d'extraire des nouvelles primitives.

En ce qui concerne le capteur de vision, les premiers travaux concernant l'utilisation d'un capteur de vision pour la commande des robots manipulateurs remontent dans les années 70. C'est à Hill et Park [6] que l'on doit l'apparition du terme asservissement visuel (*visual servoing*).

Il implique la mise en œuvre d'au moins 3 sous-systèmes :

- Un capteur visuel qui est une caméra.
- Un dispositif d'acquisition/traitement, autrement dit un ordinateur, qui est chargé du traitement d'images et de la commande.
- Un sous-système mécanique actionné représentant le robot qui est chargé de l'exécution de cette commande. Le robot génère ses mouvements de façon à ce que sa cible visuelle atteigne une certaine configuration dans l'image qu'il perçoit (il faut assurer que l'objet reste dans le champ visuel de la caméra pour qu'il soit toujours visible pendant l'exécution de la tâche).

L'asservissement visuel prend en considération plusieurs domaines de recherche, y compris la modélisation du robot (géométrique, cinématique, dynamique), la commande des systèmes en temps réel, la vision par ordinateur (l'extraction des informations visuelles, calibration de la caméra). Il touche aussi:

- Le domaine aérien: les applications les plus courantes sont la surveillance civile avec le suivi de structure de linéique (routes, lignes électriques), l'assistance au pilotage (hélicoptère), appontage automatique d'avions [7], les applications militaires (drones) [8].
- Le domaine médical [9].
- La conduite d'un véhicule sous-marin [10].

### III.2. Les primitives visuelles

Pour appliquer la vision dans le domaine robotique, il faut être en mesure de répondre à certaines questions telle que: *Quelle est la situation de l'objet? Quel type d'objet (son modèle)? Quelle est sa vitesse de déplacement? etc.*

Pour répondre à ces questions, il faut faire des mesures au niveau de l'image ce qu'on appelle *les caractéristiques visuelles* (primitives, indices ou informations), qui nous informent sur la scène. Elles sont considérées comme un point clé pour commander un robot.

#### III.2.1. Définition

On définit *une information visuelle* «s», également appelée *signal capteur* par analogie avec la commande référencée capteur, par la donnée d'une application différentielle de  $SE^3$  dans  $\mathbb{R}$ . Ces informations visuelles sont l'équivalent vectoriel du signal délivré par un capteur classique. Cette définition implique que la valeur d'une information visuelle ne dépend que de la situation entre la caméra et son environnement et de la situation de l'objet.

#### III.2.2. Modèle sténopé d'une caméra

Afin d'utiliser à bon escient les différentes informations visuelles, une phase préalable de modélisation est, comme pour tout capteur, indispensable. Cette phase consiste à identifier les paramètres du modèle de la caméra (distance focale, dimensions d'un pixel, ...). Il existe différents modèles permettant de représenter géométriquement une caméra. Le modèle sténopé, illustré sur la figure suivante est le plus couramment utilisé. Ce modèle constitué principalement par une projection perspective, permet d'établir la correspondance entre l'univers 3D et sa projection 2D dans l'image, le lecteur intéressé trouvera de plus amples détails dans [11] et [12].

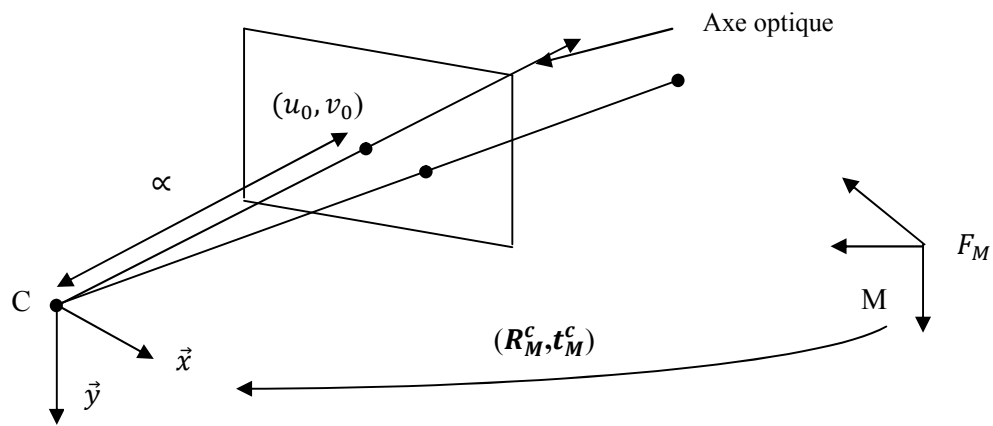
Par convention, l'axe *optique* est porté par l'axe  $Z^c$ , définissant ainsi *le plan image*  $\mathcal{P}(Z = \alpha)$ , où  $\alpha$  est *la distance focale* de la caméra. *Le point principal*  $(u_0, v_0)$  est l'intersection de l'axe optique  $(C, Z^c)$  avec le plan image. A partir des coordonnées de point

$P^c = (X^c, Y^c, Z^c)$  dans le repère caméra  $\{C\}$ , les coordonnées de sa projection  $p = (x, y)$  sur le plan image 2D, est déduite d'une application directe du théorème de Thalès (Figure I.6):

$$x = \alpha \frac{X^c}{Z^c}; \quad y = \alpha \frac{Y^c}{Z^c} \quad (\text{I.18})$$

où sous forme matricielle, en coordonnées homogènes:

$$\tilde{p} = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{P}^c = C \tilde{P}^c \quad (\text{I.19})$$



**Figure I.6.** Le modèle sténopé d'une caméra

où  $C$  est une matrice de dimension  $3 \times 4$  est appelée *matrice de la caméra*.

La caméra délivre une mesure sous forme d'un tableau de pixels. La position d'un pixel s'exprime par les coordonnées pixelliques  $f = (u, v)$  dont l'origine est, par convention, le coin supérieur gauche de l'image. Le modèle de la caméra représenté sur la figure I.6 relie les coordonnées pixelliques aux coordonnées métriques: il dépend des coordonnées pixelliques  $(u_0, v_0)$  du point principal et des dimensions  $(\rho_u, \rho_v)$  d'un pixel. Le modèle s'exprime par:

$$\begin{cases} u = u_0 + x/\rho_u \\ v = v_0 + y/\rho_v \end{cases} \quad (\text{I.20})$$

où sous forme matricielle:

$$f = \begin{pmatrix} \frac{1}{\rho_u} & 0 & u_0 \\ 0 & \frac{1}{\rho_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tilde{p} = k\tilde{p} = kC\check{P}^c \quad (I.21)$$

où  $k$  est la *matrice de paramètres de la caméra*. On regroupe généralement les matrices  $k$  et  $C$  en une *matrice de calibration*  $K$ , regroupant l'ensemble des paramètres de la caméra.

$$K = kC = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (I.22)$$

Avec  $\alpha_u = \frac{\alpha}{\rho_u}$  ;  $\alpha_v = \frac{\alpha}{\rho_v}$ .

Les coordonnées métriques utilisées sont des coordonnées normalisées, indépendantes de la focale:

$$x = \frac{x^c}{z^c}; \quad y = \frac{y^c}{z^c} \quad (I.23)$$

La connaissance *des paramètres intrinsèques*  $\xi_{in} = (\rho_u, \rho_v, \alpha_u, \alpha_v, u_0, v_0)$  est nécessaire pour pouvoir interpréter les mesures pixelliques de la caméra en termes géométriques, et ainsi faire le lien entre l'image et l'environnement. L'opération qui consiste à identifier ces paramètres est appelée *calibration de la caméra*.

En particulier, l'écriture paramétrique de la situation entre la caméra et l'effecteur du robot constitue les six *paramètres extrinsèques* de la caméra  $\xi_{ex} = P_c^e$ . Leur estimation fait aussi partie du processus de calibration. En effet, une bonne approximation de  $\xi_{ex}$  est nécessaire pour obtenir une commande précise.

### III.2.3. Le choix des informations visuelles

Les performances de l'asservissement visuel peuvent être atteintes par l'utilisation des mesures qui contiennent des informations riches. Ces informations doivent être sensibles à la situation de l'objet c'est-à-dire elles changent si la position ou l'orientation de l'objet change. Les informations de références doivent être sélectionnées et elles seront atteintes par l'application d'une séquence de commande. Elles sont principalement de deux types :

- **Les informations visuelles 2D (de nature géométrique):** dans ce cas, c'est la primitive point qui reste encore la plus couramment employée [13]. Des primitives de type droite peuvent être utilisées avec des caméras perspectives [13] et même avec des

caméras catadioptriques [14], voire des primitives de type cylindre, sphère ou ellipse [13], ainsi que des moments [15]. Cependant, si l'on veut appréhender des objets de nature plus complexe, de telles informations visuelles paraissent insuffisantes.

- **Les informations visuelles 3D:** le contrôle de la caméra est effectué dans l'espace en fonction des grandeurs 3D à asservir, choisies pour définir la situation de la caméra  $s=[R,t]$  dans le cas où le modèle de l'objet est connu.

Signalons que nous ne considérerons dans la suite de ce manuscrit que des primitives visuelles de type géométrique, nous allons considérer les informations visuelles les plus simples: les coordonnées d'un point dans l'image.

### III.3. Classification des systèmes d'asservissement visuel

Une classification bien connue permet de qualifier les différents types d'asservissement visuel selon les trois critères suivants: la configuration spatiale robot/caméra, le type de mesure et le type d'architecture de commande utilisée.

#### III.3.1. Les configurations robot/caméra

En asservissement visuel, on rencontre deux types de configurations caméra/robot selon l'application souhaitée:

##### III.3.1.1 Configuration caméra déportée (*eye-to-hand*)

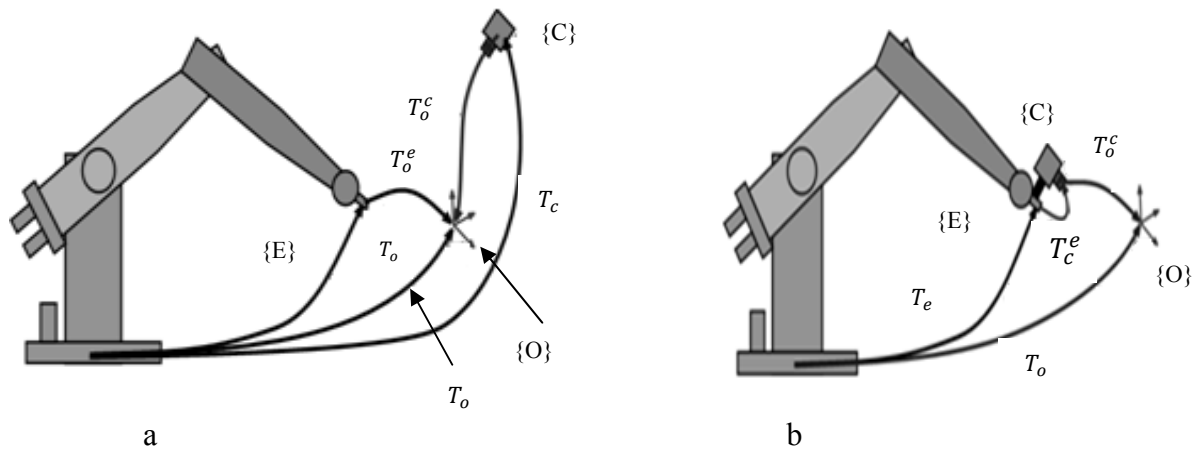
La caméra n'a pas de liaison mécanique avec le robot asservi. La caméra est positionnée de façon à ce qu'elle observe l'organe terminal du robot, la configuration articulaire ou à la fois le robot et les objets d'intérêt situés dans l'espace du travail. Elle peut être fixée sur une base immobile comme sur un autre système mécanique, elle est bien illustrée sur la figure I.7.a.

Ce type de positionnement permet des applications nécessitant un faible volume de travail et se prête bien à la robotique chirurgicale [16] où il faut observer à la fois l'outil et l'objet.

##### III.3.1.2 Configuration caméra embarquée (*eye-in-hand*)

C'est la configuration la plus répandue. Dans cette configuration, le système de vision (une ou plusieurs caméras) est rigidement lié au robot (par exemple, fixée à l'organe terminal dans le cas d'un robot manipulateur) de façon à ce qu'il observe les objets situés dans l'espace du travail (figure I.7.b). Ce type de positionnement permet des applications nécessitant un

grand volume de travail (surtout si le robot est mobile) et se prête bien à des applications de suivis de cibles.



**Fig I.7.** Les configurations robot-caméra  
**a).** La configuration eye-to-hand; **b).** La configuration eye-in-hand.  
 et la représentation des différents repères: base, effecteur robot {E},  
 caméra {C}, objet {O}.

La structure où le capteur est embarqué sur l'effecteur, structure réputée comme étant beaucoup plus performante qu'une structure où le capteur est déporté.

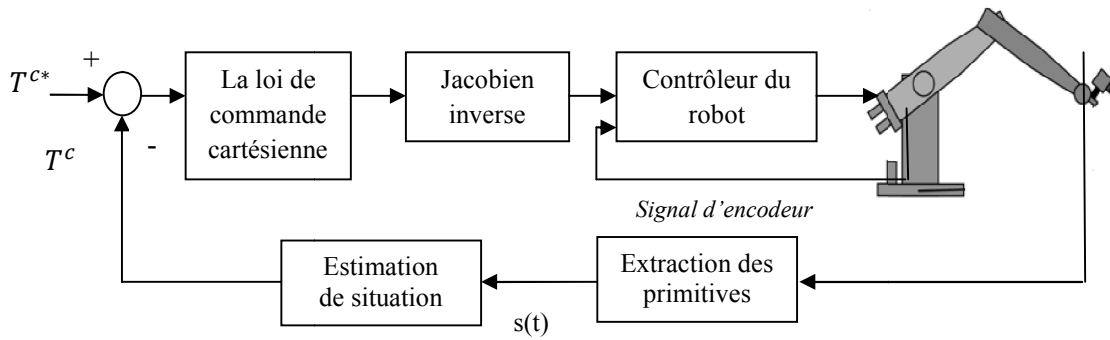
### III.3.2. Le type de mesure

Nous allons, dans cette partie, présenter les différents schémas de commande en asservissement visuel en fonction de l'espace dans lequel s'effectue le contrôle du robot, soit dans l'espace 3D ou dans l'espace image. Chaque approche ayant ses défauts et ses qualités, des méthodes hybrides ont été envisagées. Afin d'obtenir une explication plus approfondie, nous invitons le lecteur à se reporter aux articles tutoriels [1], [17] et [18].

#### III.3.2.1 Asservissement visuel 3D

Dans un asservissement visuel 3D (PBVS: Position Based Visual Servoing), la consigne est définie dans l'espace en terme d'attitude ou situation entre la caméra et un ou plusieurs objets dans la scène. Pour ce faire, la caméra doit être calibrée et un modèle de l'objet est généralement nécessaire.

Parmi les avantages liés à l'utilisation d'une telle approche, on peut citer celui de l'espace de contrôle qui est équivalent à l'espace opérationnel du robot, et qui permet de simplifier la définition de la tâche robotique.

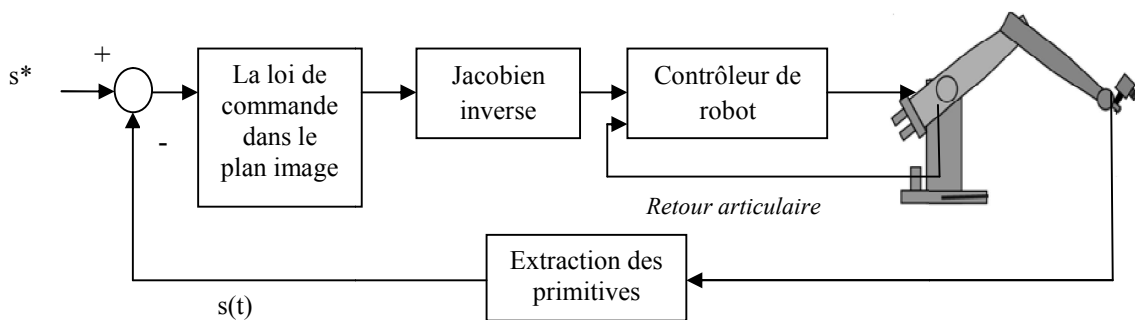


**Fig I.8.** Asservissement visuel 3D.

Parmi les inconvénients les plus cités, on peut noter le problème lié à la modélisation de l'objet, celui lié à la reconstruction et au calibrage de la caméra (paramètres intrinsèques et extrinsèques), le temps de calcul nécessaire à l'obtention de la mesure de la situation. Le principal défaut est qu'aucun contrôle n'est effectué dans l'image. Ainsi, l'objet observé peut sortir du champ de vision de la caméra, empêchant l'estimation de situation et donc la poursuite de la tâche, pour ne citer que les plus importants et les plus discutés.

### III.3.2.2. Asservissement visuel 2D

L'asservissement visuel 2D, couramment appelée «Image Based Visual Servoing IBVS», définit le contrôle dans l'espace du capteur (plan image). Cette approche semble être optimale au niveau temps de calcul puisque les informations extraites de l'image sont directement utilisées dans la boucle de commande. Les signaux capteurs  $s(t)$  sont constitués de primitives extraites de l'image caractérisant une forme observée dans celle-ci. Le schéma d'asservissement visuel correspondant est donné par la figure I.9.



**Fig I.9.** Asservissement visuel 2D.

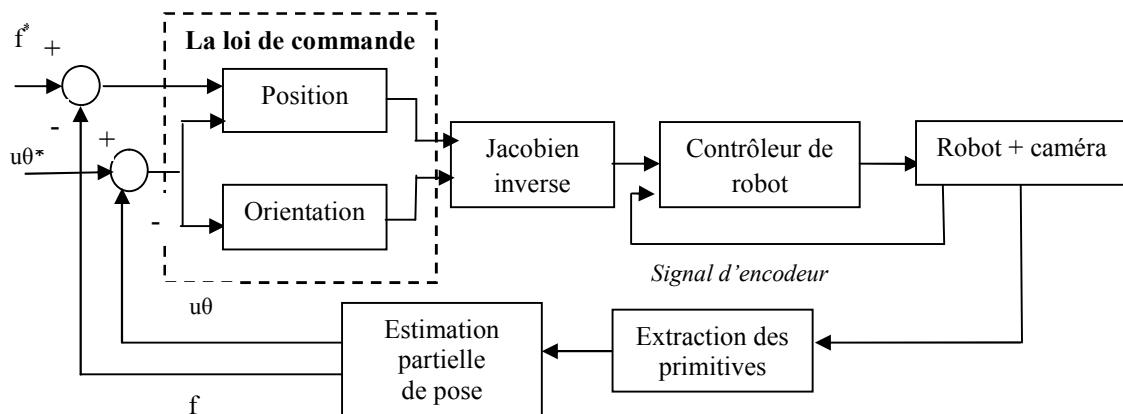
Il consiste, donc à contrôler les mouvements de la caméra afin que les mesures  $s(t)$  atteignent une configuration désirée exprimée par  $s^*$  dans l'image. Nous rappelons que le

vecteur consigne  $s^*$  peut être constant comme il peut être variable (une trajectoire à suivre par exemple).

En pratique, la loi de commande se montre robuste aux bruits de mesure, aux erreurs de modélisation du robot et d'étalonnage de la caméra [15]. Ceci n'est vrai que dans le cas de petits déplacements à réaliser. Dans le cas d'un déplacement plus important, ce type de commande peut conduire à des minima locaux, des singularités, ou encore des trajectoires de caméra inadéquates [19]. En effet, la commande dans l'espace image garantit un comportement satisfaisant des primitives visuelles sans garantir le comportement spatial de la caméra.

### III.3.2.3. Asservissement visuel hybride

Dans le but d'obtenir les avantages des deux approches précédentes tout en évitant leurs défauts, une approche est proposée qui permet de regrouper des informations 2D et 3D au sein d'un vecteur de dimension 6 dans le but de bénéficier un meilleur comportement 3D [20] ou garder les primitives dans le champ de vision du capteur. Cette approche permet un fort découplage de la loi de commande et un contrôle partiel dans l'image, qui permet de conserver l'objet dans le champ visuel de la caméra. Le schéma correspondant est donné par la figure I.10.



**Fig I.10:** Asservissement visuel 1/2D

où :

$u$  : vecteur unitaire ;

$\theta$  : angle de rotation.

Cette méthode est originalement proposée pour la configuration caméra embarquée. Contrairement à l'asservissement visuel 3D, elle n'a pas besoin de modèle géométrique de l'objet. En comparant avec l'asservissement visuel 2D, elle assure la convergence de la loi de commande dans l'espace entier de la tâche.

Cette méthode est basée sur l'estimation de déplacement de la caméra (la rotation et la translation de la caméra) entre la vue courante et désirée de l'objet.

### **III.3.3. Le type de commande**

Il existe principalement deux approches de commande permettant de contrôler un robot avec un retour visuel bouclé.

#### *III.3.3.1. La commande cinématique*

Elle est la plus répandue. Elle consiste à envoyer périodiquement des consignes de vitesses articulaires au contrôleur bas niveau du robot. Les effets dynamiques n'étant pas modélisés, la fonction de transfert du robot est considérée comme étant un intégrateur pur. Cette commande a l'avantage d'être facilement mise en œuvre car le régulateur bas niveau du robot permet de générer et d'appliquer une trajectoire de positions articulaires à partir de consignes de vitesses articulaires. Néanmoins, elle n'est pas adéquate pour des applications nécessitant une réactivité élevée du robot du fait de la non modélisation des effets dynamiques.

#### *III.3.3.2. La commande dynamique*

Elle tient compte d'une modélisation plus réaliste du comportement du robot. Cependant une identification des paramètres dynamiques (masses, inerties, frottements,...) du robot est nécessaire afin de mettre en œuvre une commande en force ou en couple. Cette architecture permet de tirer pleinement partie des possibilités dynamiques offertes par le robot pour réaliser des asservissements visuels rapides.

Dans le cadre de notre travail, nous nous positionnerons sur l'asservissement visuel 2D basé sur une commande cinématique.

### **III.4. Les principales étapes de la mise en œuvre d'un asservissement visuel 2D**

Le choix des primitives visuelles et l'obtention de la relation les liants aux mouvements de l'effecteur sont deux points fondamentaux de l'asservissement visuel 2D.

### III.4.1. L'extraction des primitives visuelles

L'utilisation du capteur visuel est particulièrement intéressante en raison de la grande richesse des informations qu'une caméra peut fournir et en raison de la grande variété des tâches qu'elle permet de réaliser. Cette grande richesse nécessite, cependant, de disposer d'algorithmes de traitement d'images particulièrement performants afin d'extraire, à une cadence proche de la cadence vidéos, les informations qui seront utilisées en commande.

#### III.4.1.1. L'algorithme d'extraction des informations visuelles

*Quels sont les endroits de l'image contenant le plus d'informations? On distingue essentiellement trois types de régions: les zones homogènes, les contours et les points d'intérêt qui feront l'objet de cette section.*

Un point dit *d'intérêt* peut se caractériser par une forte variation bidirectionnelle par rapport aux pixels voisins. Les primitives susceptibles d'être observées sont souvent assimilées visuellement à des coins, et correspondent à une forte variation bidirectionnelle du signal.

Plusieurs approches ont été développées afin d'extraire ces points dans une image. Nous citons le détecteur de Maravec [21], le détecteur de Shi et Tomasi [22] et le détecteur de Harris [23]. Dans la majorité des cas, le détecteur de Harris donne des meilleurs résultats. Nous renvoyons le lecteur à consulter [24] pour de plus amples détails sur les détecteurs de points d'intérêts. Nous nous concentrons sur le détecteur de Harris.

#### *Le détecteur de Harris*

Nous allons introduire une technique classique la plus utilisée pour extraire les primitives visuelles de type «points». Ces primitives sont distinctes dans l'image et elles sont souvent appelées *les points d'intérêt*, *les points saillants* ou *les points clés*. Le niveau de gris associé à un point d'intérêt diffère fortement des valeurs des autres pixels du voisinage (il est caractérisé par une forte variation bidirectionnelle du signal).

Soit  $s=(x,y)$  un point d'intérêt, et  $I(s)$  le niveau de gris en ce point, une mesure d'autocorrélation telle que:

$$E(s) = \sum_{u,v \in \mathcal{W}} w_{u,v} [I(x+u, y+v) - I(s)]^2 \quad (\text{I.24})$$

doit fournir une forte valeur.  $\mathcal{W}$  désigne une fenêtre autour du point, et  $w_{u,v}$  une pondération pour que la contribution de chaque pixel de la fenêtre soit dépendante de la distance au point

d'intérêt. Un développement de Taylor de  $I(x + u, y + v)$  (l'intensité  $I$  au voisinage du pixel  $(u, v)$ ) donné par :

$$I(x + u, y + v) = I(s) + uI_x + vI_y + \mathcal{O}(u^2, v^2) \quad (\text{I.25})$$

où  $I_x$  et  $I_y$  correspondent aux dérivées premières du signal (gradients du signal). En négligeant le terme  $\mathcal{O}(u^2, v^2)$ , la mesure d'autocorrelation devient alors:

$$E(s) = \sum_{u,v \in \mathcal{W}} w_{u,v} (u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2) \quad (\text{I.26})$$

Nous nous retrouvons avec la définition d'une quadratique, qui peut encore s'écrire sous la forme matricielle:

$$E(s) = \sum_{u,v \in \mathcal{W}} w_{u,v} X_{\mathcal{W}}^T M X_{\mathcal{W}}, \quad (\text{I.27})$$

$$\text{Avec } M = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \text{ et } X_{\mathcal{W}}^T = (u, v). \quad (\text{I.28})$$

Le filtre  $w_{u,v}$  utilisé est un filtre Gaussien:  $w_{u,v} = \exp -(u^2 + v^2) / 2\sigma^2$ .

La matrice  $M$  étant symétrique, ces valeurs propres obtenues ( $\lambda_1$  et  $\lambda_2$  telles que  $\lambda_1 \geq \lambda_2$ ) permettent de caractériser la zone considérée :

- $\lambda_1 = \lambda_2 = 0$  : la région considérée a une intensité approximativement constante.
- $\lambda_1 > \lambda_2 = 0$  : la zone correspond à un contour. Le vecteur propre associé à  $\lambda_1$  est perpendiculaire à ce contour ;
- $\lambda_1 > \lambda_2 > \lambda_t$  ( $\lambda_t$  étant un seuil): la zone contient une variation bidirectionnelle du signal. Elle correspond donc à un point d'intérêt.

Par voie de conséquence, Harris propose l'opérateur suivant pour détecter les coins dans une image :

$$H(s) = \det (M) - \beta \text{tr}^2 (M) \quad (\text{I.29})$$

Avec :  $\det (M) = \prod_i \lambda_i$ : déterminant de  $M$ .

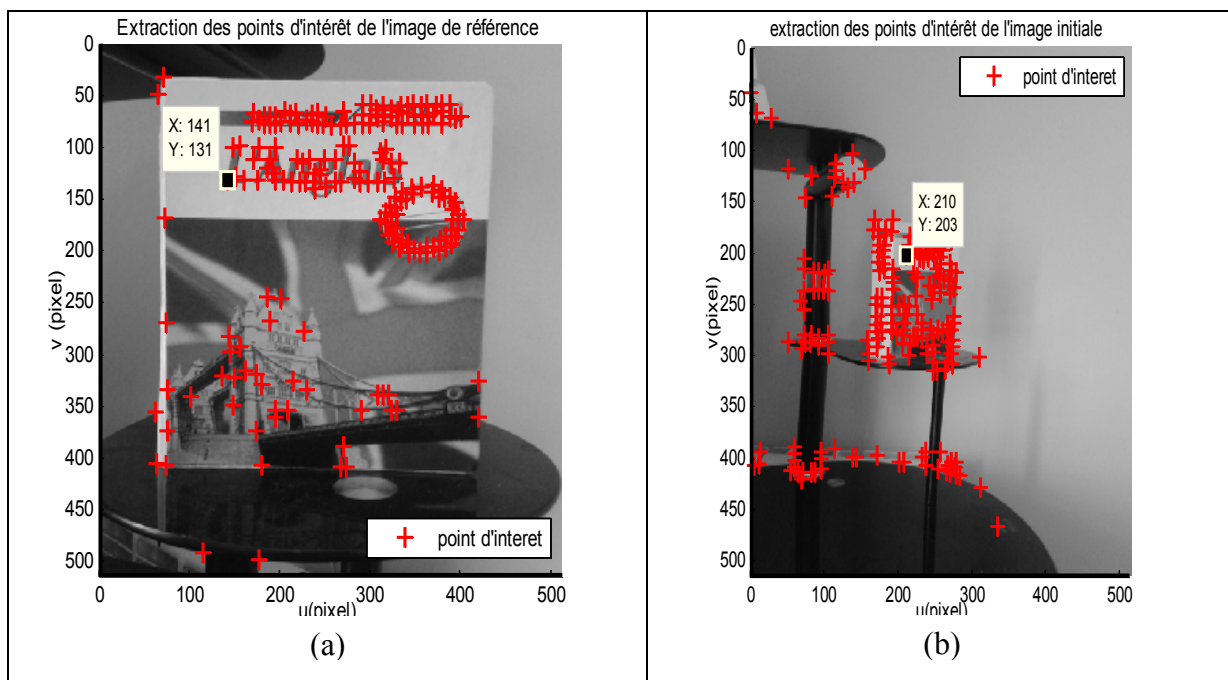
$\text{tr} (M) = \sum_i \lambda_i$ : la trace de  $M$ .

$\beta$ : est un scalaire positif.

Un point d'intérêt se situe aux maxima locaux de la matrice d'autocorrélation  $M$ . Les valeurs de  $H$  sont positives au voisinage d'un coin, négatives au voisinage d'un contour et faibles dans une région d'intensité constante.

La figure I.11 illustre un exemple de détecteur de Harris sur deux images. Nous pouvons observer que certains points obtenus sont très proches les uns des autres. Le choix de la taille de la fenêtre influence sur la répartition des points détectés. Dans notre cas, nous avons utilisé une fenêtre de taille  $[11 \ 11]$  et  $\beta=0.04$ .

Le détecteur de Harris est largement utilisé pour les applications de la mise en correspondance d'images.



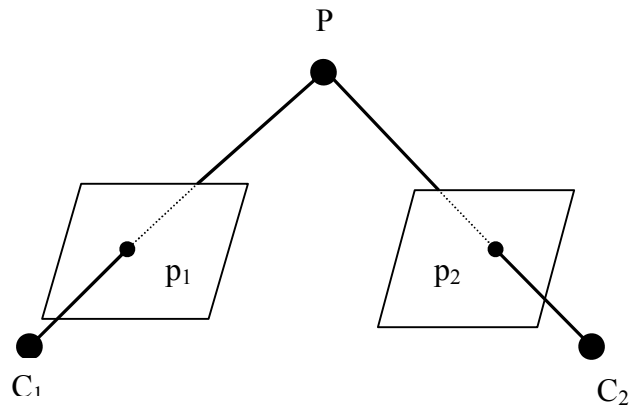
**Fig I.11.** Détection de points d'intérêt en utilisant le détecteur de Harris  
(a) Image désirée, (b) Image initiale.

#### III.4.1.2. La mise en correspondance

Dans les schémas d'asservissement visuel, on suppose qu'il existe un lien entre les images, initiale, courante et désirée. En effet, ils requièrent la mise en correspondance de primitives visuelles extraites de l'image initiale avec celles extraites de l'image désirée. Ces primitives sont ensuite suivies lors du mouvement de la caméra (et/ou de l'objet). Si une de ces étapes échoue, la tâche robotique ne pourra pas être réalisée [19]. Par exemple, s'il est impossible d'extraire des primitives visuelles communes aux images initiale et désirée ou si

les primitives visuelles sortent du champ de vision de la caméra durant ce mouvement (et/ou l'objet) alors la tâche ne pourra pas être réalisée.

La mise en correspondance d'images a pour objectif de déterminer les correspondances entre deux images d'une même scène. Considérons deux images 1 et 2 d'une scène (figure I.12), il s'agit ici de déterminer, pour un élément de l'image 1, l'élément qui lui correspond dans l'image 2 et éventuellement dans d'autres images.



**Fig I.12.** Deux pixels  $p_1$  et  $p_2$  correspondants

Les mesures de corrélation ont été très largement utilisées pour réaliser la mise en correspondance. L'idée est de définir une mesure de similarité entre les pixels de deux images. Le principe est de considérer, pour un pixel  $p_1$  de l'image 1, une fenêtre centrée en  $p_1$  et de calculer sa corrélation avec une fenêtre dans la deuxième image. Cette mesure est maximale si  $p_2$  correspond à  $p_1$ .

La figure I.12 illustre un exemple de la mise en correspondance entre deux images. On remarque que les points extraits de l'image gauche correspondent bien aux points extraits de l'image droite.

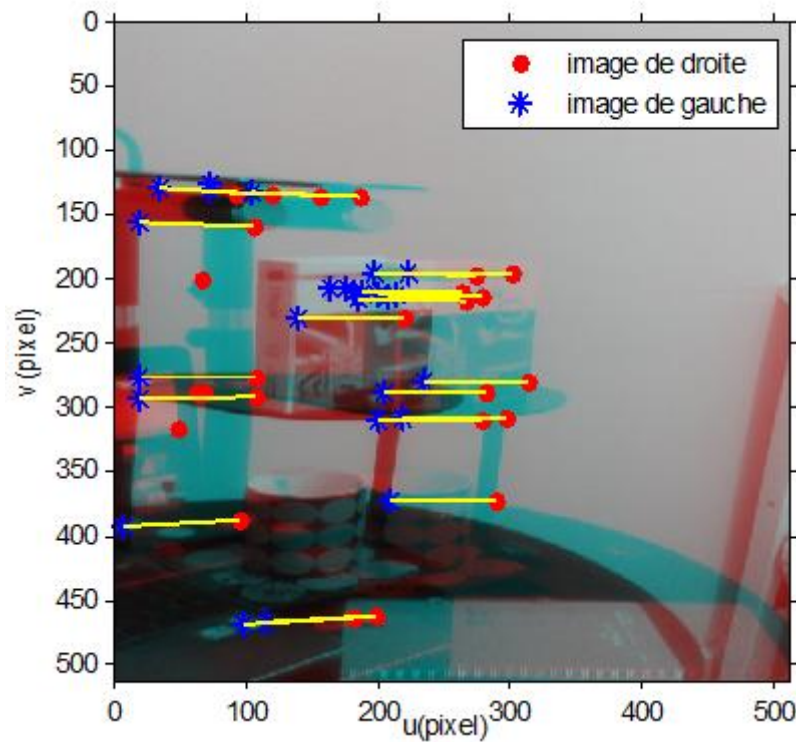


Fig I.13. La mise en correspondance entre deux images

### III.4.2. La détermination de la matrice d'interaction

L'asservissement visuel 2D nécessite le calcul de *jacobien d'image* [25] ou *matrice d'interaction* [26].

A partir d'un vecteur «*s*» représentant les primitives visuelles courantes, on cherche à atteindre un motif «*s\**» correspondant à la position finale souhaitée de l'effecteur. On considère que la valeur d'une primitive visuelle varie seulement en fonction des mouvements de la caméra ou des objets qu'elle perçoit.

Les mouvements de la caméra sont réalisés à l'aide des différents axes constituant le robot et la situation de la caméra ne dépend que de la valeur des coordonnées articulaires *q*. De plus, si les objets perçus sont eux-mêmes mobiles, alors «*s*» peut s'écrire:

$$s = s(q, t) \quad (\text{I.30})$$

où le paramètre temps «*t*» représente cette contribution du mouvement des objets.

La différentielle de «*s*» permet de relier les variations des primitives visuelles dans l'image aux mouvements de la caméra et de la scène. A partir de la dérivation de l'équation (I.31) on obtient facilement:

$$\dot{s} = \frac{\partial s}{\partial q} \dot{q} + \frac{\partial s}{\partial t} \quad (\text{I.31})$$

Le terme  $\frac{\partial s}{\partial q}$  peut en fait se décomposer sous la forme:

$$\frac{\partial s}{\partial q} = \frac{\partial s}{\partial r} \frac{\partial r}{\partial q} \quad (\text{I.32})$$

Si l'on choisit pour  $r$  la situation de la caméra,  $\frac{\partial s}{\partial r} = L_s$ : *le jacobien de la tâche* (matrice d'interaction), il n'est dépendant que de la tâche choisie, et en aucun cas de la configuration du robot. Il caractérise complètement les interactions entre le capteur et l'environnement et nous l'appellerons donc *torseur d'interaction*, dont l'expression dépend à la fois des caractéristiques de l'environnement et du capteur lui-même. Il est de dimension  $l \times m$  et il est exprimé dans le repère caméra.

$$L_s = \frac{\partial s}{\partial r} = \begin{bmatrix} \frac{\partial s_1(r)}{\partial r_1} & \dots & \frac{\partial s_1(r)}{\partial r_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_l(r)}{\partial r_1} & \dots & \frac{\partial s_l(r)}{\partial r_m} \end{bmatrix} \quad (\text{I.33})$$

Avec:

$l$ : le nombre des primitives visuelles extraites de l'image.

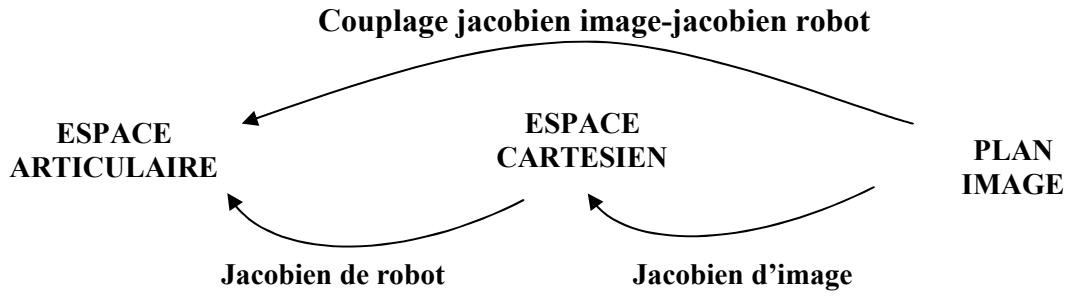
$m$ : la dimension de l'espace de la tâche  $\mathcal{T}$ .

$\frac{\partial r}{\partial q} = J_r$ : *le jacobien du robot* qui ne dépendant que de la configuration du robot et en aucun cas de la tâche.

Lorsque l'environnement perçu par la caméra est immobile, le terme  $\frac{\partial s}{\partial t} = 0$  et l'équation (I.32) s'écrit:

$$\dot{s} = L_s J_r \dot{q} \quad (\text{I.34})$$

La figure suivante illustre la relation entre le jacobien robot et le jacobien image.



**Fig I.14.** Le couplage robot-image

La relation (I.35) décrit la variation des primitives visuelles dans l'image par rapport à la situation de la caméra. Différents travaux ont permis de déterminer de façon algébrique cette matrice d'interaction pour différentes primitives. Dans le cas des caméras perspectives, la matrice d'interaction a été déterminée algébriquement pour des primitives visuelles de type point, droite, cercle, cylindre [13]. Dans le cas de caméras omnidirectionnelles, seules des primitives de type point et droite ont été exploitées [14].

Supposons que la caméra (l'effecteur du robot) se déplace avec une vitesse de translation  $v=[v_x, v_y, v_z]$  et une vitesse de rotation  $\Omega=[\omega_x, \omega_y, \omega_z]$  par rapport au repère caméra et elle observe un point de monde  $P=(X, Y, Z)$ . La vitesse de ce point par rapport au repère caméra est:

$$\dot{P} = -\Omega \times P - v \quad (\text{I.35})$$

Nous pouvons l'écrire sous forme scalaire comme suit:

$$\begin{aligned} \dot{X} &= Y\omega_z - Z\omega_y - v_x \\ \dot{Y} &= Z\omega_x - X\omega_z - v_y \\ \dot{Z} &= X\omega_y - Y\omega_x - v_z \end{aligned} \quad (\text{I.36})$$

En utilisant la projection perspective d'un point dans l'espace sur le plan image (l'équation I.19) et on prend la distance focale égale à l'unité, on trouve:

$$x = \frac{X}{Z}, \quad y = \frac{Y}{Z}$$

en dérivant ces termes par rapport au temps, on obtient:

$$\dot{x} = \frac{\dot{X}Z - X\dot{Z}}{Z^2}, \quad \dot{y} = \frac{\dot{Y}Z - Y\dot{Z}}{Z^2} \quad (\text{I.37})$$

En remplaçant  $X = xZ$  et  $Y = yZ$  dans l'équation (I.38) et on peut l'écrire sous forme matricielle comme suit:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (\text{I.38})$$

avec:

$$L_s = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \quad (\text{I.39})$$

C'est la matrice d'interaction pour un seul point qui relie la variation des coordonnées de ce point à la vitesse de la caméra par rapport au repère caméra.

On peut aussi exprimer cette matrice en fonction des coordonnées pixeliques d'un point telle que  $x = \frac{\alpha}{\rho_u}x + u_0$ ,  $y = \frac{\alpha}{\rho_v}y + v_0$  avec  $\alpha$ : la distance focale,  $(\rho_u, \rho_v)$  les dimensions d'un pixel,  $(u_0, v_0)$ : les coordonnées de centre optique. On peut les réécrire sous la forme suivante:

$$x = \frac{\rho_u}{\alpha} \bar{u}, \quad y = \frac{\rho_v}{\alpha} \bar{v} \quad (\text{I.40})$$

où  $\bar{u} = u - u_0$  et  $\bar{v} = v - v_0$ , On dérive (I.41):

$$\dot{x} = \frac{\rho_u}{\alpha} \dot{\bar{u}}, \quad \dot{y} = \frac{\rho_v}{\alpha} \dot{\bar{v}} \quad (\text{I.41})$$

On remplace (I.41) et (I.42) dans (I.39), on obtient:

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{v}} \end{pmatrix} = \begin{pmatrix} -\frac{\alpha}{\rho_u Z} & 0 & \frac{\bar{u}}{Z} & \frac{\rho_u \bar{u} \bar{v}}{\alpha} & -\frac{\alpha^2 + \rho_u^2 \bar{u}^2}{\rho_u \alpha} & \bar{v} \\ 0 & -\frac{\alpha}{\rho_v Z} & \frac{\bar{v}}{Z} & \frac{\alpha^2 + \rho_v^2 \bar{v}^2}{\rho_v \alpha} & -\frac{\rho_v \bar{u} \bar{v}}{\alpha} & -\bar{u} \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (\text{I.42})$$

Sans la moindre perte de généralité, on peut supposer que  $\rho_u = \rho_v = 1$  (un pixel carrée) ainsi que  $u_0 = v_0 = 0$ , alors  $L_s$  devient  $L_f$  avec  $f = [u, v]$ :

$$L_f = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{u}{Z} & uv & -(1+u^2) & v \\ 0 & -\frac{1}{Z} & \frac{v}{Z} & 1+v^2 & -uv & -u \end{pmatrix} \quad (\text{I.43})$$

La matrice d'interaction  $L_s$  fait intervenir une information 3D, c'est *la profondeur* des points dans le repère caméra et notée par «Z». Celle-ci peut être estimée en ligne en utilisant une reconstruction partielle 3D de la profondeur Z, ou hors ligne en l'estimant par la profondeur désirée  $Z_d$ , ce choix simplifie les calculs tout en maintenant la convergence du système [1].

Dans l'asservissement visuel, nous nous intéressons à déterminer la vitesse de la caméra (la vitesse du manipulateur),  $v_c = J_r \dot{q}$ , qu'il faut appliquer pour atteindre la valeur désirée « $s^*$ ». Ceci nécessite la résolution de l'équation (I.35) qui sera discutée dans le prochain paragraphe.

### III.4.3. L'élaboration de la loi de commande

#### III.4.3.1 Le formalisme de la fonction de tâche

Un formalisme introduit dans [27] permet d'exprimer une tâche robotique comme étant la régulation à zéro sur un horizon temporel d'une fonction  $e(q, t)$  appelée *fonction de tâche*.

Une fonction de tâche en asservissement visuel 2D est modélisée par un ensemble de primitives visuelles calculées à partir d'une image à chaque position du robot. Afin de construire une loi de commande robuste et stable ces primitives doivent être régularisées à une valeur souhaitée.

Un asservissement visuel 2D est tout à fait adapté pour réaliser un éventail beaucoup plus large de tâches qui s'expriment directement dans l'image, telles qu'asservir la caméra de sorte qu'un objet soit centré dans l'image.

Notre travail repose sur une approche qui consiste à spécifier une tâche de régulation dans l'image d'un ensemble de primitives visuelles dans le but de réaliser une tâche de positionnement.

Revenons à présent à la caractérisation des primitives visuelles où la caméra est embarquée sur l'effecteur d'un robot, alors « $s$ » peut s'écrire (équation I.31):

$$s=s(q, t)$$

Dans le cas d'asservissement visuel 2D, la fonction de tâche, également appelée *tâche référencée vision*, se définit par l'expression [25]:

$$e(q, t) = C[s(q, t) - s^*] \quad (I.44)$$

où:

$s$ : est le vecteur de mesures de dimension ' $l$ ' obtenu à partir des primitives visuelles courantes qui doivent être choisies convenablement.

$s^*$ : est la consigne sur les primitives visuelles de la tâche. Elle peut être constante ou variable en fonction du temps.

$(s-s^*)$ : est l'*erreur visuelle* que la tâche a pour objectif de minimiser.

$C$ : est une matrice, dite *de combinaison*, de dimensions  $m \times l$  avec  $m$  correspond au nombre de degrés de liberté du robot à commander et il faut que le nombre d'informations visuelles soit supérieur au nombre de degrés de libertés du robot.

### III.4.3.2 La commande cinématique

Consiste à trouver une relation cinématique entre la situation de l'objet, la situation du robot et les primitives extraites de l'image sans prendre en considération les effets dynamiques.

L'objectif de la loi de commande est de contrôler les mouvements de la caméra de façon à ce que l'erreur, définie entre les primitives de référence et les primitives mesurées dans l'image tende vers zéro.

Généralement, la régulation à zéro de la fonction de tâche (équation (I.45)) est réalisée par une commande cinématique stabilisante assurant une décroissance exponentielle du vecteur d'erreur « $e$ », à savoir:

$$\dot{e} = -\lambda e \quad (\text{I.45})$$

où:  $\lambda$  est un gain positif.

Nous présentons ici la commande cinématique dans le cas où la matrice de combinaison « $C$ » est constante. La dérivée temporelle de la fonction de tâche est alors égale à:

$$\dot{e} = C(\dot{s} - \dot{s}^*) \quad (\text{I.46})$$

En tenant compte de l'équation (I.32), cette expression peut s'écrire directement en fonction du torseur cinématique  $v_c$  du capteur visuel selon:

$$\dot{e} = CL_s v_c + C \frac{\partial s}{\partial t} - C \dot{s}^* \quad (\text{I.47})$$

ou encore:

$$\dot{e} = CL_s v_c + \frac{\partial e}{\partial t} \quad (\text{I.48})$$

Le terme  $\dot{e}$  représente les variations de «e» dues à la variation de s. En remplaçant le terme de gauche par l'équation (I.46), nous obtenons la commande cinématique caractérisant la décroissance exponentielle de la fonction de tâche, à savoir:

$$v_c = -(CL_s)^{-1}(\lambda e + \frac{\partial e}{\partial t}) \quad (\text{I.49})$$

La mise en œuvre d'une telle commande est d'ordre théorique, car elle suppose que:

- Les dynamiques de système soient négligeables et n'interviennent pas.
- La matrice d'interaction  $L_s$  soit exactement connue, ce qui implique la connaissance des paramètres du capteur visuel.

En pratique, la matrice d'interaction  $L_s$  est souvent construite sur des hypothèses fortes qui s'éloignent de son exactitude. Dans la majorité des cas, elle dépend des paramètres intrinsèques des capteurs visuels dont les valeurs sont généralement estimées expérimentalement. La loi de commande réellement appliquée est donc:

$$v_c = -(C\hat{L}_s)^{-1}(\lambda e + \frac{\partial e}{\partial t}) \quad (\text{I.50})$$

où  $\hat{L}_s$  est une valeur approchée ou estimée de la matrice d'interaction et  $\frac{\partial e}{\partial t}$  est la variation de e par rapport au temps. Si la scène considérée est immobile alors la consigne visuelle est constante, la loi de commande devient:

$$v_c = -\lambda(C\hat{L}_s)^{-1}e \quad (\text{I.51})$$

Toutefois, le torseur cinématique  $v_c$  est exprimé dans le repère caméra. Il doit être transformé vers le repère de base avant d'être utilisé dans la boucle de commande.

Nous nous intéressons à présent aux différents choix possibles pour C et  $\hat{L}_s$ .

#### III.4.3.3. Le choix de la matrice de combinaison C

Le choix de la matrice de combinaison C dépend du nombre d'informations visuelles  $l$  considéré:

1. Si  $l=m$ : la matrice C peut être choisie égale à la matrice identité. Dans ce cas, la fonction de tâche correspond directement à l'erreur visuelle et un comportement de

décroissance exponentielle pour tous les éléments du vecteur  $s$  est obtenu avec une loi de commande égale à:

$$v_c = -\lambda L_s^{-1}(s - s^*) \quad (\text{I.53})$$

2. Si  $l > m$ : la matrice  $C$  est de dimension  $m \times l$  et doit être de rang  $m$ . Elle peut être choisie comme étant la pseudo-inverse d'une estimation de la matrice d'interaction à la position désirée [25], soit  $C = \hat{L}_s^+ /_{s=s^*}$ . Une autre solution consiste à choisir  $C$  égale à la matrice identité de dimension  $l$  et à remplacer dans (I.52) l'inverse de la matrice d'interaction par sa pseudo-inverse  $L_s^+ = (L_s^T L_s)^{-1} L_s^T$ . Pour ce dernier choix nous obtenons alors la loi de commande suivante:

$$v_c = -\lambda \hat{L}_s^+(s - s^*) \quad (\text{I.54})$$

Cette équation constitue la base de l'asservissement visuel 2D.

#### III.4.3.4.L'approximation de la matrice d'interaction

La matrice d'interaction  $L_s$  ne peut pas être utilisée directement dans la loi de commande par contre une estimation ou une approximation de cette matrice est utilisée. Il existe plusieurs choix pour trouver l'estimé  $\hat{L}_s^+$  utilisable dans la loi de commande, on peut distinguer:

1.  $\hat{L}_s^+ = L_s^+(s^*, Z^*)$ : consiste à utiliser la matrice d'interaction à la position désirée, qui nécessite d'être calculée qu'une seule fois avec la valeur des primitives et des profondeurs de la position désirée. Dans ce cas,  $\hat{L}_s^+$  est constante et la profondeur à la position désirée pour chaque point est considérée, c'est-à-dire pas de paramètres 3D à estimer pendant l'exécution d'une tâche en asservissement visuel.
2.  $\hat{L}_s^+ = L_s^+(s, Z)$ : l'estimation de la matrice d'interaction à la situation courante qui permet d'assurer une décroissance exponentielle de l'erreur  $e$  mais nécessite des mesures au niveau d'images et une estimation de la profondeur à chaque itération de la loi de commande.
3.  $\hat{L}_s^+ = 1/2[L_s + L_s^*]^+$ : l'approximation de la matrice d'interaction est la moyenne entre les deux, la matrice d'interaction à la situation courante et à la situation désirée. Elle est aussi intéressante car elle assure un domaine de convergence plus étendu que les autres approximations [17].

#### IV. Difficultés de mise en oeuvre

L'asservissement visuel 2D permet la réalisation des tâches robotiques par comparaison des primitives visuelles extraites de l'image courante, avec une image de référence. Bien que l'AV 2D est robuste aux erreurs de modélisation, plusieurs inconvénients peuvent être mentionnés:

1. Quand la situation initiale est très éloignée de celle désirée, un asservissement visuel 2D peut échouer. Rapportons, dans ce cas de figure, l'utilisation avantageuse des asservissements visuel 2D 1/2 qui utilisent comme leur non le laisse supposer, à la fois des données 3D et 2D mais, sans toutefois nécessiter la connaissance de modèle [20].
2. La matrice d'interaction est considérée comme une pièce maîtresse de l'asservissement visuel 2D, cette matrice peut être difficile à obtenir et son inversion pose quelques fois des problèmes de singularités ou de divergence [19]. Ces problèmes surviennent lors de l'inversion de la matrice d'interaction. En effet, suivant le nombre de primitives choisies, des problèmes de dimensions peuvent intervenir. Dans ce cas, le jacobien de l'image n'est localement plus inversible et le système diverge.
3. L'évaluation de la matrice d'interaction n'est pas toujours triviale et dépend du type de primitives considérées. Dans l'expression de cette matrice intervient la quantité  $Z$  appelée *profondeur* ou distance relative entre la caméra et la cible. Cette information n'est pas directement mesurable car aucune reconstruction tridimensionnelle n'est effectuée en AV 2D. Dans la plupart des cas, c'est l'approximation de cette profondeur à la position finale (supposée connue) qui est utilisée. Cependant, ce choix entraîne des trajectoires non optimales des primitives visuelles dans le plan image, pouvant amener les primitives à sortir du champ visuel de la caméra et donc de rendre le calcul de la commande impossible [19].

Afin de pallier cette difficulté une méthode numérique du calcul de l'inverse de la matrice d'interaction par apprentissage, a été proposée [28]. Cette méthode dépend de la connaissance a priori de l'optimum global, et nécessite un grand nombre de points pour l'apprentissage.

- 4 Plusieurs inconvénients peuvent être mentionnés lorsque les primitives visuelles ne sont pas correctement choisies. Pour résoudre ce problème il faut concevoir des primitives visuelles adéquates. Dans [29] par exemple, les auteurs ont montré que le comportement du système dépend explicitement de genre des primitives visuelles

considérées. Par conséquent, des lignes, des cercles, des sphères, des cylindres, mais aussi des moments peuvent être utilisés. La loi de commande est généralement une loi exponentielle décroissante découplée. Cette solution ne permet pas de prendre en compte les contraintes.

5. Outre le problème classique des minima locaux et des singularités dans la matrice d'interaction mentionné ci dessus, la prise en compte de contraintes est un problème délicat en AV 2D. On peut distinguer des contraintes 2D et des contraintes 3D. Qui concerne la contrainte 2D, également appelée contrainte de visibilité, doit garantir que les mesures dans l'image restent dans le champ visuel de la caméra. Si la visibilité de la cible n'est plus assurée alors l'algorithme de commande est interrompu. Les contraintes 3D telles que les limites de l'espace de travail qui s'assurent que le robot réalise ses mouvements admissibles dans son espace de travail tout au long de la tâche.

Une façon de prendre en considération les contraintes est de combiner la planification et le suivi de trajectoire [30]. Cette solution assure une trajectoire optimale de la caméra dans l'espace cartésien et une visibilité des primitives (assurer les contraintes 2D et 3D).

Dans la troisième solution, les primitives visuelles sont généralement simples à savoir des primitives de type «points». Des lois de commande avancées telles que la commande adaptative [31] et la commande prédictive [32] ont été rapportés dans la littérature.

## V. Conclusion

Dans ce chapitre, nous avons montré que l'utilisant des informations délivrées par des caméras présente l'intérêt de pouvoir positionner plus librement et plus facilement le robot tout en autorisant la localisation d'objets. Cette approche, appelée "*commande référencée vision*" ou "*asservissement visuel*", a été initiée il y a plus d'une vingtaine d'années.

L'asservissement visuel permet de contrôler un robot grâce aux primitives visuelles acquises avec une ou plusieurs caméras. Les différentes approches d'asservissement visuel sont classées selon l'espace dans lequel la fonction de tâche est exprimée. Dans la littérature, nous pouvons distinguer 3 catégories d'asservissement visuel: l'asservissement visuel 3D, l'asservissement visuel 2D et l'asservissement visuel 2D 1/2.

Nous sommes focalisés sur l'asservissement visuel 2D. Cette stratégie est stable localement et, contrairement à toutes les méthodes existantes, elle ne nécessite pas une

connaissance sur les paramètres du modèle de l'objet. Seules des informations visuelles issues d'une image capturée à la position de référence et d'une image capturée à la position courante doivent suffire pour calculer la fonction de tâche et la loi de commande.

Nous nous intéressons ici aux lois de commande associées, qui sont des lois de commandes purement cinématiques. Le schéma de commande cinématique est basé sur le formalisme de la régulation à zéro de la fonction de tâche. A la convergence de l'asservissement visuel 2D, le robot atteint la position de référence, si les primitives visuelles observées coïncident avec celles de l'image de référence.

Nous nous intéressons aux asservissements visuels dont l'objectif peut être posé en terme de réalisation d'un positionnement désiré d'une caméra montée sur un bras manipulateur par rapport à une cible immobile appartenant à la scène observée.

La stratégie proposée dans le chapitre suivant est la commande prédictive non linéaire appliquée pour réaliser des tâches d'asservissement visuel 2D, dans le but de résoudre les problèmes rencontrés pendant la mise en œuvre de l'asservissement visuel 2D.

---

# Commande prédictive visuelle d'un bras manipulateur

## I. Introduction

Le régulateur a un rôle crucial dans un système d'asservissement visuel performant. La plupart des travaux réalisés en asservissement visuel ne prennent pas en considération la dynamique des manipulateurs.

Dans ce chapitre, une stratégie de commande est proposée pour effectuer des tâches d'asservissement visuel 2D. La technique utilisée se base sur une commande prédictive non linéaire. Elle présente l'avantage de tenir compte, explicitement, des contraintes mécaniques et/ ou visuelles. Un modèle de procédé assure la prédiction de l'image et permet de s'affranchir de la détermination de l'inverse de la matrice d'interaction.

Ce chapitre est composé de deux parties: dans la première partie, nous présentons brièvement la théorie de la commande prédictive ainsi que son principe de base. Dans la deuxième partie, les tâches d'asservissement visuel 2D sont réalisées par la commande prédictive non linéaire sous forme du problème d'optimisation non linéaire sous contraintes dans le plan image. La stratégie proposée s'appelle *la commande prédictive visuelle CPV* qui permet de prendre en compte les contraintes 2D et 3D. On commence cette partie par la présentation du principe de cette stratégie, la structure de commande ainsi que la formulation mathématique. La prédiction de l'image est obtenue grâce à un modèle et le choix de ce dernier est discuté. Un modèle global non linéaire et un modèle local basé sur la matrice d'interaction sont considérés. Nous terminerons par une conclusion.

## II. Commande prédictive

*La commande prédictive* ou *commande à horizon glissant* (ou Model Predictive Control (MPC) en anglais) fait, depuis 1960, l'objet de nombreux travaux de recherche et elle est, actuellement, l'une des commandes les plus utilisées en industrie dans des domaines très divers (alimentaire, chimie, etc.) [33]. Elle est née d'un besoin réel dans le monde industriel, qu'il s'agit d'avoir un système de régulation capable et de performances plus élevées que les régulateurs classiques tel que le régulateur de type PID, tout en respectant des contraintes de fonctionnement et de production toujours plus sévères.

Au début des années 60, Propoi [34] fut l'un des premiers à proposer explicitement une forme de commande prédictive basée sur un modèle.

La première génération de la commande prédictive appliquée en milieu industriel a été initiée par Richalet [34] sous le nom Identification et Commande (IDCOM) et par les ingénieurs de Shell sous le nom Dynamic Matrix Control (DMC). L'objectif est de poursuivre une référence mais les contraintes ne sont pas encore prises en compte.

La deuxième génération, qui apparaît au début des années 1980, permet la prise en compte de contraintes sur les entrées et les sorties en posant un problème quadratique (Quadratic Dynamic Matrix Control).

En fait, toutes ces variantes de stratégies de commande prédictive sont aujourd'hui regroupées sous le terme générique MPC, illustrant ainsi le rôle fondamental du modèle.

Une synthèse sur ces méthodes, où sont exposées les caractéristiques les plus représentatives, peut être trouvée dans CAMACHO et BORDONS [35]. La commande prédictive non-linéaire (NMPC) a également connu un essor conséquent, avec des applications convaincantes en termes de qualités de réglage [36].

### II.1. Le concept de base de la commande prédictive

Par apprentissage et par expérience, on pilote une voiture. On sait que telle action sur l'accélérateur, sur le frein ou sur le volant modifié, après un certain temps, la vitesse et la position du véhicule. Il faut anticiper, c'est-à-dire regarder, non pas au niveau de pare-chocs, mais «loin» devant, et qu'il ne faut pas attendre d'être sur l'obstacle pour freiner. On sait que telle action produit tel effet, et en inversant la proposition, on sait quelle action appliquer pour obtenir tel comportement [35].

Le principe de la commande prédictive se révèle donc simple à comprendre car issu du comportement humain. On peut en effet considérer cette stratégie de commande comme «naturelle». L'objectif est de calculer les entrées futures de telle manière que les sorties futures soient proches de la consigne.

## II.2. Définition et principe

La *commande prédictive* est une méthode de commande avancée qui utilise le modèle du système à commander (linéaire ou non) pour prédire son comportement futur et choisir la meilleure décision au sens d'un certain coût, tout en respectant les contraintes.

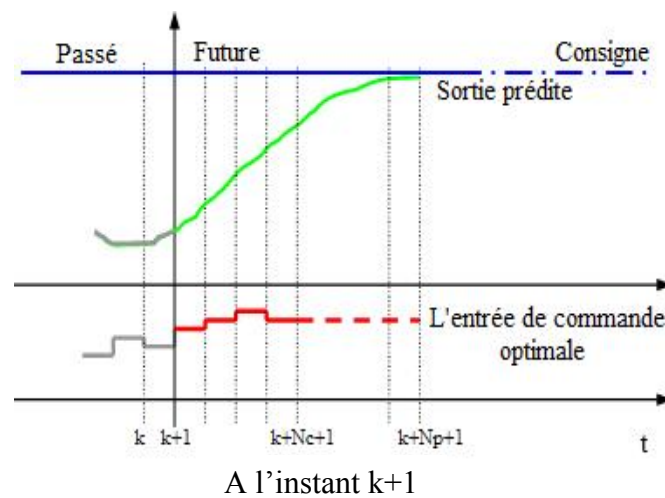
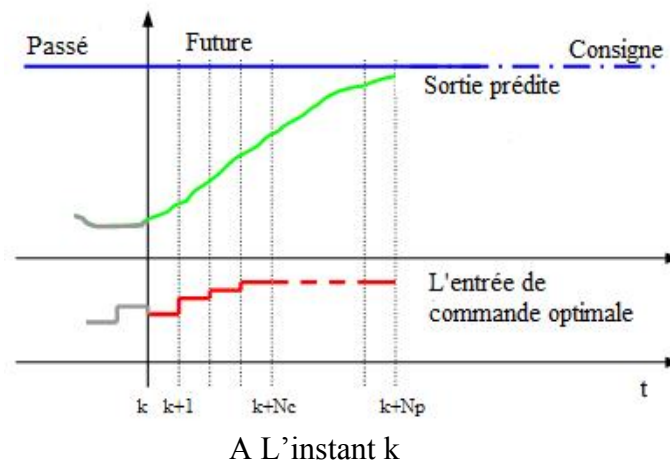
Le but de la commande prédictive est de déterminer, à chaque intervalle, une séquence de commande qui permet à la sortie du procédé de coïncider avec la consigne dans le futur [37].

Le principe de la commande prédictive consiste à utiliser un modèle dynamique du processus à l'intérieur d'un régulateur, en temps réel, afin d'anticiper le futur comportement du procédé. A chaque période d'échantillonnage, la séquence de commandes optimales est calculée par minimisation d'une fonction objectif et seul le début de cette séquence est appliqué au système [38].

La méthodologie de tous les régulateurs, appartenant à la famille de la commande prédictive (MPC), s'articule autour des points suivants:

1. Calculer les sorties prédites à chaque instant  $k$  pour un certain horizon  $H_p$  (dénommé horizon de prédiction). Les sorties prédites sont dénotées  $\hat{y}(k + i/k)$  où  $i=1,2,\dots,H_p$ .
2. Minimiser un critère à horizon fini, en fonction de l'écart entre la sortie prédite du système et la consigne. Des contraintes sur la sortie ou sur la commande peuvent être également imposées.
3. Obtenir une séquence de commandes futures  $\{u(k + i/k), i = 0, \dots, H_c - 1\}$  sur un horizon de commande  $H_c$  inférieure ou égale à  $H_p$ .
4. Le premier élément  $\{u(k)\}$  du vecteur du signal de commande optimale  $\{u(k + i/k), i = 0, \dots, H_c - 1\}$ , issu du problème précédent, est appliqué au système et le reste est rejeté car à l'instant suivant la nouvelle sortie  $y(k + 1)$  est disponible.
5. A l'instant d'échantillonnage suivant, l'horizon de prédiction glisse d'un pas et la procédure d'optimisation est répétée. D'où son nom *la commande optimale à horizon glissant*.

L'état actuel du système (mesuré ou estimé) est utilisé comme valeur initiale pour la prédiction [39]. Ce principe est illustré sur la figure suivante:



**Fig II.1.** Le principe de la commande prédictive.

### II.3. Les éléments de MPC

Durant ces dernières années, plusieurs approches de commande prédictive ont été développées, mais ont en commun les éléments suivants (la figure II.2):

#### a. Le modèle

Le modèle du système à commander joue un rôle central dans la commande prédictive. La détermination du modèle est faite de deux manières:

- en établissant les équations mathématiques des phénomènes qui régissent le fonctionnement du processus. On parle alors *de modèle de connaissance*.

- en appliquant des entrées au processus et en recueillant les réponses. On parle alors *de modèle de représentation*.

Le choix de modèle se fait, bien sûr, en fonction de la précision des objectifs à atteindre et des informations disponibles. Le modèle choisi permet de prédire le comportement dynamique du procédé, en fonction des commandes passées et futures.

### b. La trajectoire de référence

Elle représente le comportement désiré du procédé. Sa détermination est très importante, car elle représente les spécifications (performances dynamiques, stabilité) du procédé en boucle fermée. Elle peut être constante ou variable. Elle est notée  $y_{ref}$ .

### c. Le critère d'optimisation

Il est appelé aussi *critère de performance* ou *fonction coût*. C'est la traduction mathématique des objectifs de commande du procédé sous contraintes de fonctionnement. Il est en fonction de l'erreur entre la trajectoire de référence et la sortie prédite sur un horizon de prédiction fini  $H_p$ . La sortie prédite dépend de la séquence de commandes futures. Elle est notée  $U$  et composée de  $H_c$  commandes. La formulation mathématique de la fonction coût (cas discret multi-variable) est donnée par:

$$J(u) = \sum_{j=k+1}^{k+H_p} (y_{ref}(j) - y_p(j))^T Q (y_{ref}(j) - y_p(j)) \quad (\text{II.1})$$

avec,  $Q$ : matrice symétrique définie positive.

A cette fonction coût s'ajoute les contraintes inévitables dans le contexte expérimental ou industriel. Ces contraintes peuvent être de deux types:

- Les contraintes de fonctionnement: limitation des actionneurs en amplitude et/ou vitesse:

$$u_{min} \leq u_k \leq u_{max} \text{ et } \Delta u_{min} \leq u_k - u_{k-1} \leq \Delta u_{max} \quad (\text{II.2})$$

- Les contraintes physiques: limitation de certaines variables.

### d. La méthode de résolution

Le problème d'optimisation sous contraintes qui, à partir de l'écart entre le comportement constaté et celui voulu, va trouver la commande future à appliquer pour atteindre l'objectif établi. Il peut être résolu à l'aide de plusieurs méthodes. La différence

entre ces méthodes réside dans le type du critère de performance choisi et du modèle utilisé. Il peut aussi être transformé en un problème pénalisé (méthode des pénalités) et résolu ensuite par un algorithme d'optimisation (linéaire ou non linéaire) sans contraintes.

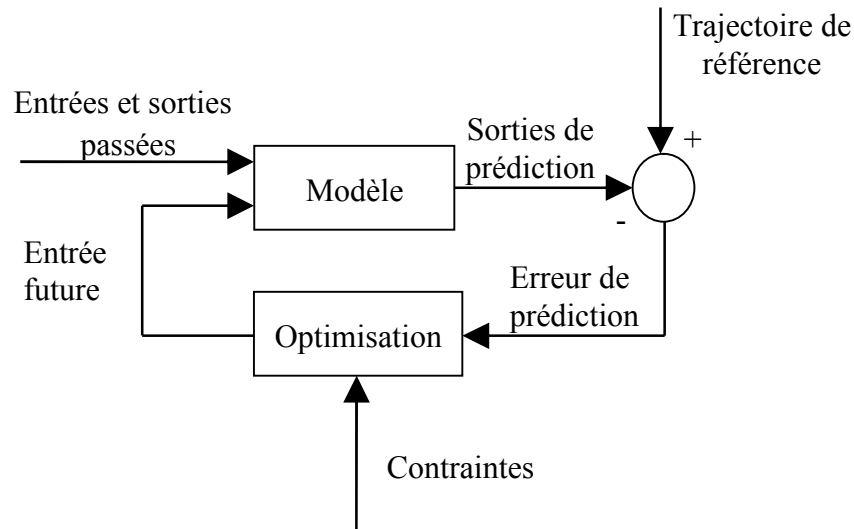


Fig II.2. Les éléments de MPC

#### II.4. Le choix des horizons

En ayant adopté cette démarche, deux concepts s'introduisent naturellement dans la commande prédictive:

- La longueur des suites d'arguments. C'est-à-dire l'horizon de commande  $H_c$ .
- La longueur sur laquelle est évalué le critère de performance. c'est-à-dire l'horizon de prédiction  $H_p$ .

##### II.4.1. Le choix de l'horizon de prédiction $H_p$

Le choix de l'horizon de prédiction  $H_p$  joue un rôle important tant par la quantité d'informations fournie à l'algorithme, que du point de vue de la faisabilité numérique du problème d'optimisation [39]. Il n'existe pas encore de méthode permettant de choisir l'horizon de prédiction optimal vis-à-vis du problème posé.

##### II.4.2. Le choix de l'horizon de commande $H_c$

En ce qui concerne le choix de l'horizon de commande  $H_c$ , une valeur élevée permet a priori, d'atteindre des objectifs plus difficiles. Cependant, le choix de  $H_c=1$  est reconnu comme étant suffisant dans la plupart des cas.

### III. La commande prédictive et l'asservissement visuel 2D

Le but de l'asservissement visuel est de réguler à zéro l'erreur  $e(t)$  entre les primitives visuelles courantes  $s(t)$  et les primitives de références  $s^*$ . Dans l'asservissement visuel 2D, les primitives visuelles sont exprimées dans l'image. La relation entre la vitesse de déplacement de la caméra  $v_c(t)$  et la variation dans le temps des primitives visuelles  $\dot{s}(t)$  est donnée par la matrice d'interaction  $L_s$ . Ainsi, en spécifiant une loi exponentielle décroissante découplée de l'erreur  $e(t)$ , on obtient l'entrée de commande appliqué au robot (chapitre I, section III.4.3.2):

$$v_c(t) = -\lambda \hat{L}_s^+ e(t) \text{ avec } \lambda > 0 \quad (\text{II.3})$$

où  $\hat{L}_s^+$  est l'approximation de la pseudo-inverse de la matrice  $L_s$ .

Les systèmes utilisés en asservissement visuel sont des systèmes robotiques soumis à des contraintes mécaniques (limitations des actionneurs, butée articulaires, ...). De plus, un des points faibles de l'asservissement visuel 2D réside dans le fait que les primitives visuelles peuvent sortir de l'image pendant la convergence du système (chapitre I, section IV).

La commande prédictive semble bien adaptée pour traiter ce problème contraint (contraintes mécaniques et contraintes de visibilité). Son principe, étendu à l'asservissement visuel, consiste à déterminer une séquence de commande qui minimise l'erreur définie entre les primitives de référence ( $\text{image}_{\text{réf}}$ ) et les primitives extraites à partir de l'image courante ( $\text{image}_{\text{cour}}$ ).

#### III.1. La commande prédictive visuelle (CPV)

Toutes les stratégies prédictives sont basées sur quatre points communs: une trajectoire de référence, un modèle dynamique de processus, une fonction de coût, et une méthode de résolution du problème d'optimisation, comme il était mentionné dans la section II.3.

Le point clé de l'approche prédictive est le modèle utilisé pour prédire le comportement des processus sur un horizon futur. Son choix influence sur la précision de suivi et sur le temps de calcul. Dans le cas de CPV, le procédé considéré se compose en général du système robotique et d'une caméra. Le système robotique peut être un robot mobile [32], un drone [40] ou un bras manipulateur [41]. Le système de vision peut être une caméra perspective [40] ou catadioptrique [41] quelle que soit sa configuration par rapport au robot, c'est-à-dire déportée ou embarquée. Les entrées du modèle sont les variables de commande du système robotique. Les sorties du modèle sont les primitives visuelles. Le modèle est utilisé pour prédire les

valeurs des primitives sur un horizon de prédiction, en prenant en compte les entrées de commande et la satisfaction des contraintes.

La commande prédictive a été appliquée pour la première fois dans la structure d'asservissement visuel dans [42]. Un modèle ARIMAX (Autoregressive and Integrated Moving Average with exogenous input) multivariable permet d'implémenter un régulateur GPC (Generalized Predictive Control) sur un bras manipulateur. Cette stratégie permet d'améliorer les performances en termes de rapidité de convergence, de précision de la tâche en comparant avec le régulateur proportionnel, mais les contraintes ne sont pas prises en considérations.

Par contre l'utilisation de la commande prédictive dans une structure d'asservissement visuel 2D revient à [16]. La CPV sans contraintes est utilisée pour compenser les mouvements des organes induits par la respiration et les mouvements cardiaques. Ces derniers peuvent être gênants pour le chirurgien lors de l'opération, en particulier pour les interventions chirurgicales nécessitant une bonne précision (par exemple: l'insertion d'aiguille et d'une suture). La propriété périodique de perturbations a été incluse dans le modèle pour permettre au robot de les anticiper. Les résultats expérimentaux montrent l'efficacité de la méthode proposée pour compenser les mouvements physiologiques complexes dans le but de suivre une trajectoire de référence avec le rejet des perturbations. Les primitives visuelles sont constituées de 4 diodes attachées à un morceau de tissu, en utilisant une caméra déportée.

Dans [43], SAUVÉE prend en compte des contraintes sur la commande, des contraintes de visibilité ainsi que des butées articulaires pour commander un bras manipulateur à 6 d.d.l. La caméra est déportée pour observer un objet attaché à l'effecteur du robot. La convergence du robot vers la position désirée est atteinte. Garantir la robustesse aux erreurs de calibration et aux bruits de mesures. Par contre, le temps de calcul pour résoudre le problème d'optimisation est supérieur à la période d'échantillonnage, ce qui ne permet pas l'implémentation en temps réel.

FERREIRA [44] réalise l'implémentation réelle de la CPV en utilisant des primitives visuelles de types points sur un robot PUMA à 6 d.d.l. La comparaison entre deux approches prédictives (GPC et MPC) et un régulateur PI, montre l'efficacité des stratégies prédictives en termes de rapidité que celle utilisant le régulateur PI. Dans la même année, ALLIBERT [45] utilise la CPV dans l'application en temps réel, sur un robot mobile avec une caméra catadioptrique embarquée, montre l'efficacité de l'approche pour la stabilisation du robot en un point.

Au début, la CPV est appliquée seulement pour réaliser des tâches de positionnement. Dans [46], les auteurs utilisent la CPV pour réaliser une tâche de suivi de trajectoire et d'évitement d'obstacles en utilisant un robot mobile. Le suivi de trajectoire est réalisé sur le plan simulation, à partir d'une situation initiale arbitraire du robot avec la prise en compte des contraintes sur l'amplitude de la commande (vitesse linéaire et angulaire). Les résultats de simulation montrent la robustesse du schéma de commande par rapport aux erreurs de modélisation (calibration de la caméra) et aux perturbations (bruit blanc ajouté à la sortie), avec la satisfaction des contraintes de visibilité présentées par une zone définie dans l'image, qui correspond à un obstacle dans l'espace de travail du robot.

Dans [32] et [41], l'objectif d'IBVS est formulé en un problème d'optimisation non-linéaire dans le plan image. L'application est réalisée sur un robot mobile et sur un bras manipulateur, avec une caméra catadioptrique. La CPV est formulée avec la prise en compte des contraintes mécaniques et des contraintes de visibilité. Elle montre son efficacité en comparant avec l'approche IBVS classique, surtout pour les configurations les plus difficiles telles que les grands déplacements ainsi que le mouvement de rotation de la caméra.

Pour éviter la complexité du modèle global non linéaire utilisé pour la prédiction de primitives visuelles dans l'image, un modèle local a été proposé dans [47] et [48]. Il est basé sur la discrétisation de la relation liant la dérivation temporelle des primitives visuelles à la vitesse de la caméra à travers le jacobien de l'image. La convergence du robot vers la position souhaitée est atteinte à travers la résolution d'un problème d'optimisation non linéaire sous contraintes de visibilité. Les performances de cette approche sont testées seulement sur le plan simulation et montre l'efficacité de la CPV surtout pour les grands déplacements. L'intérêt d'un modèle local est présenté dans [49] sans la prise en compte des contraintes.

ALLIBERT [50] a développé un algorithme de commande prédictive visuelle en utilisant un modèle global non linéaire et un modèle local pour prédire l'image future sur un horizon fini.

Le premier travail qui permet d'appliquer la commande prédictive pour les applications d'asservissement visuel 2D sur un bras mobile (une base mobile avec un bras manipulateur à 5 d.d.l) avec une caméra embarquée est présenté dans [51]. Le but est de déplacer le robot vers une cible (un tube avec un bouchon rouge). Les contraintes considérées sont les contraintes de visibilité (assurer la présence de la cible dans le champ visuel de la caméra pendant le mouvement du robot), les contraintes de localisation (le robot évite les obstacles pendant son mouvement vers la cible) ainsi que la contrainte sur la vitesse de translation et de

rotation du robot. Les résultats expérimentaux montrent l'efficacité et la validité du régulateur proposé.

Dans [52] une nouvelle architecture de la commande prédictive est présentée. La prédiction de l'image est faite par un modèle local en intégrant la trajectoire de référence. Des primitives visuelles de type moments d'image sont utilisées dans [53] avec un modèle local sur un bras manipulateur. Les résultats de simulation montrent l'efficacité de l'approche qui permet d'éviter les étapes de la mise en correspondance et l'estimation de la profondeur qui sont nécessaire pour les primitives visuelles de type points.

Dans [54], la CPV a été testée sur un système d'asservissement visuel 2D en temps réel, réalisé sur un bras manipulateur avec une caméra embarquée. Les résultats expérimentaux sont présentés avec l'utilisation des moments d'image comme primitives visuelles. Les auteurs ont utilisé un modèle local pour prédire les primitives visuelles dans le plan image.

La commande prédictive visuelle ne s'applique pas seulement sur les bras manipulateurs et les robots mobiles, mais touche aussi les robots volants. Dans [40], les auteurs proposent un algorithme d'évitement d'obstacles d'un drone avec des ailes fixes (UAV: Unmanned Aerial Vehicle). C'est le seul travail qui a permis de traiter le problème d'évitement d'obstacles, la visibilité de la cible ainsi que les contraintes sur l'entrée de commande pour la commande d'un UAV.

### III.2. La structure de commande à modèle interne

La structure de la commande considérée est une structure connue sous le nom de *commande à modèle interne* (CMI) [55] (Figure II.3).

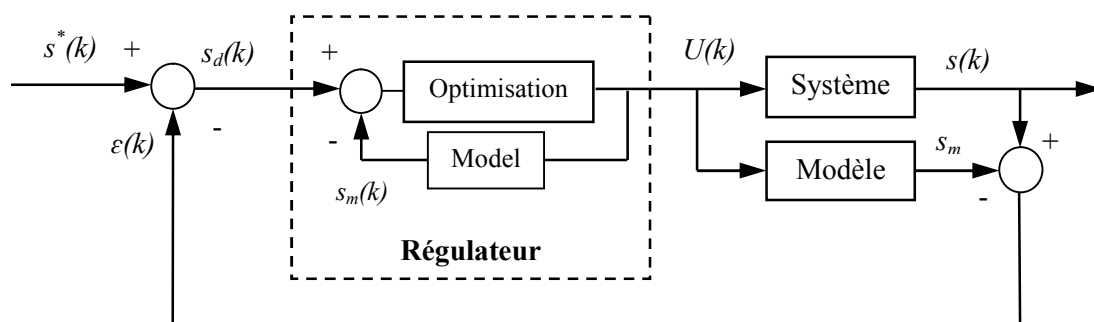


Fig II.3. La structure de commande à modèle interne.

Le bloc «système» contient le système robotique et la caméra. La variable  $U$  est l'entrée de commande du robot. La sortie  $s$  est la valeur courante des primitives visuelles. Pour l'asservissement visuel 2D,  $s^*$  est exprimée dans le plan image, comme des primitives visuelles de référence. Le signal d'erreur  $\varepsilon$  entre les primitives actuelles  $s$  et les valeurs prédites par le modèle du système  $s_m$ , représente toutes les erreurs de modélisation et les perturbations. Il s'exprime par:

$$\varepsilon(k) = s(k) - s_m(k) \quad (\text{II.4})$$

Le régulateur utilisé dans l'asservissement visuel 2D classique est remplacé, dans l'approche prédictive, par un algorithme d'optimisation. Celui-ci minimise la différence entre la primitive désirée  $s_d$  et la sortie prédite du modèle  $s_m$ . En effet, selon la figure II.3, on peut écrire :

$$\begin{aligned} s_d(k) &= s^*(k) - \varepsilon(k) \\ s_d(k) &= s^*(k) - (s(k) - s_m(k)) \\ s_d(k) - s_m(k) &= s^*(k) - s(k) \end{aligned} \quad (\text{II.5})$$

avec  $k$ : l'itération courante:

Par conséquent, atteindre les primitives de référence  $s^*$  par la sortie  $s$  de processus est équivalent à atteindre les primitives désirées  $s_d$  par la sortie du modèle  $s_m$ . Le modèle prédit le comportement des primitives sur un horizon de prédiction  $H_p$  fini. La différence  $s_d(k) - s_m(k)$  entre les primitives désirées et les primitives prédites est utilisée pour définir la fonction coût  $J$  pour être minimum par rapport à une séquence de commande  $\tilde{U}$ . Seul le premier élément  $U(k)$  de la séquence de commande optimale est réellement appliqué au système. A la prochaine période d'échantillonnage, en raison de perturbations et des erreurs de modélisation, les mesures sont mises à jour. L'horizon fini se déplace d'un pas en avant et la procédure recommence.

### III.3. Formulation mathématique du modèle de prédiction

La fonction coût  $J$  est définie comme une fonction quadratique de l'erreur, qu'il faut minimiser. Due à la structure CMI, la formulation mathématique de la CPV peut être écrite en temps discret comme:

$$\begin{aligned} \min J(U) \\ \tilde{U} \in \mathbb{R}^{m \times H_p} \end{aligned} \quad (\text{II.6})$$

avec:

$$J(U) = \sum_{j=k+1}^{k+H_p} [s_d(j) - s_m(j)]^T Q(j) [s_d(j) - s_m(j)] \quad (\text{II.7})$$

et:

$$s_d(j) = s^*(j) - \varepsilon(j) \quad (\text{II.8})$$

$$\begin{cases} x(j) = f(x(j-1), U(j-1)) \\ s_m(j) = h(x(j)) \end{cases} \quad (\text{II.9})$$

Les variables  $x \in \mathbb{R}^n$ ,  $U \in \mathbb{R}^m$  et  $s_m \in \mathbb{R}^p$  sont respectivement l'état, l'entrée et la sortie du modèle. Nous verrons, dans la section suivante, que l'état peut être choisi différemment selon le modèle de prédiction utilisé et selon les contraintes à satisfaire.

La première équation non linéaire de (II.9) décrit la dynamique du système où  $x(j)$  représente l'état prédit à l'instant  $j$ ,  $\forall j \in [k+1; k+H_p]$ . Pour  $j = k+1$ , l'état prédit  $s_m$  égale à l'état  $s$  du système à l'instant  $k$ . En outre, en cas d'erreur de modélisation et des perturbations, la deuxième boucle est assurée par le signal d'erreur  $\varepsilon(j)$  qui modifie en conséquence la trajectoire de référence.

La deuxième équation de (II.9) est l'équation de sortie. Pour calculer  $s_d(j)$ ,  $\forall j \in [k+1; k+H_p]$ , nous avons besoin de calculer l'erreur  $\varepsilon(j)$  définie par l'équation (II.4).

Cette erreur dépend de  $s_m(j)$  qui est disponible mais aussi  $s(j)$  qui est inconnu sur l'horizon de prédiction. Par conséquent, l'erreur  $\varepsilon(j)$  est supposée constante sur l'horizon de prédiction:

$$\varepsilon(j) = \varepsilon(k) = s(k) - s_m(k), \quad \forall j \in [k+1; k+H_p] \quad (\text{II.10})$$

Finalement,  $\tilde{U} = \{U(k), U(k+1), \dots, U(k+H_c), \dots, U(k+H_p-1)\}$  est une séquence optimale. De  $U(k+H_c+1)$  à  $U(k+H_p-1)$ , l'entrée de commande est constante et égale à  $U(k+H_c)$  avec  $H_c$  est l'horizon de commande. La matrice de pondération  $Q(j)$  est une matrice symétrique définie positive.

### Remarque:

*Le calcul de la séquence de commandes permet au robot de passer d'une position à l'instant  $k$  à une position à l'instant  $k+H_p$  qui correspond à la position de l'image de*

référence. Donc, la commande prédictive visuelle peut être assimilée à une planification de trajectoire implicite.

L'un des principaux avantages de la CPV est la capacité de traiter, explicitement, les contraintes dans le problème d'optimisation.

Trois types de contraintes sont distingués:

- *Contraintes sur l'état du système robotique.* Elle peut être une contrainte mécanique comme la limite de l'espace de travail où l'état représente la situation de la caméra.

$$x_{min} \leq x(k) \leq x_{max} \quad (\text{II.11})$$

- *Contraintes 2D* également nommées contraintes de visibilité pour assurer que les primitives visuelles restent dans le plan de l'image ou pour représenter des zones interdites dans l'image. Celui-ci peut être très utile pour faire face à l'évitement d'obstacle.

$$s_{min} \leq s_m(k) \leq s_{max} \quad (\text{II.12})$$

- *Contraintes de commande* telles que les limitations de l'amplitude et de la vitesse des actionneurs.

$$U_{min} \leq U(k) \leq U_{max} \quad (\text{II.13})$$

Ces contraintes sont ajoutées au problème (II.6) qui devient un problème d'optimisation non linéaire sous contraintes:

$$\begin{aligned} \min J(U) \\ \tilde{U} \in \mathbb{K} \end{aligned} \quad (\text{II.14})$$

où  $\mathbb{K}$  est le domaine de contraintes. Les contraintes (II.11), (II.12) et (II.13) peuvent être formulées par des fonctions non linéaires  $C(U)$  et  $C_{eq}(U)$  [56]. définit par:

$$\begin{cases} C(U) \leq 0 \\ C_{eq}(U) = 0 \end{cases} \quad (\text{II.15})$$

De nombreuses méthodes d'optimisation sous contraintes, sont disponibles. Dans notre cas, un programme quadratique séquentiel (SQP) [57] est utilisé.

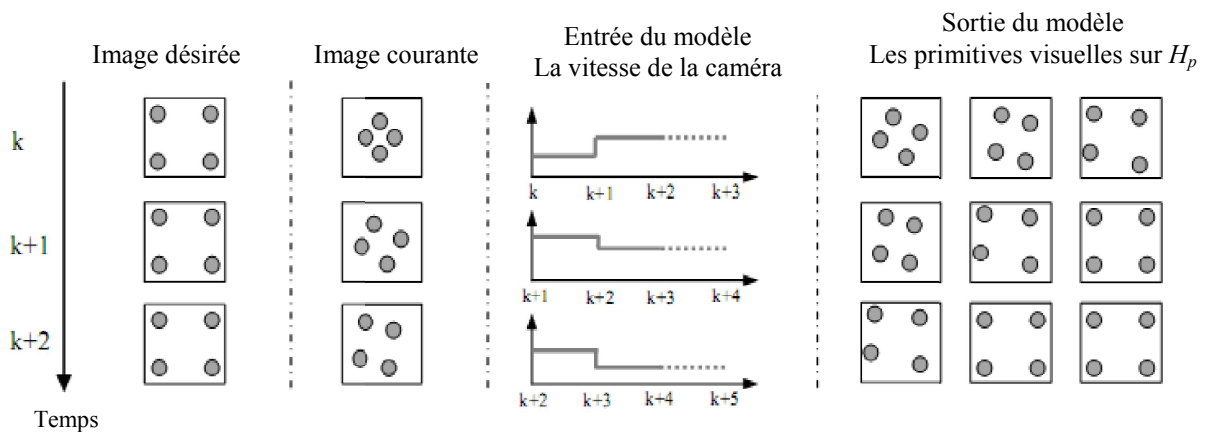
Dans un souci d'applicabilité temps réel, l'algorithme d'optimisation utilisé doit pouvoir satisfaire, à tout instant, les contraintes et ceci avec un temps de calcul le plus petit possible.

### III.4. Le modèle de prédiction d'image

Nous nous concentrons ici sur le modèle utilisé pour prédire l'évolution des primitives visuelles dans l'image.

Nous considérons une caméra perspective qui se déplace dans un espace à 6 degrés de liberté (d.d.l). Elle observe les primitives visuelles de type points qui sont fixes (cible immobile). Un point 3D, avec des coordonnées  $P=(X, Y, Z)$  dans le repère caméra, est projeté dans le plan image comme un point 2D de coordonnées  $s=(x, y)$ . La période d'échantillonnage est  $T_e$  et l'entrée de commande  $U$  est la vitesse de la caméra notée  $v_c = [v_x, v_y, v_z, w_x, w_y, w_z]$ .

Le rôle de ce modèle est de prévoir, sur l'horizon  $H_p$ , l'évolution des primitives visuelles par rapport à la vitesse de la caméra. Le principe de la prédiction de ces primitives dans l'image est représenté sur la figure II.4.



**Fig II.4.** Le principe de la prédiction d'image ( $H_p=3, H_c=2$ ).

Pour effectuer cette prédiction dans l'image, deux types de modèle peuvent être envisagées: *un modèle global non linéaire* et *un modèle local basé sur la matrice d'interaction*. L'identification du modèle, décrit ci-dessus en (II.9), est discuté en prenant en considération les deux cas présentés dans la section suivante.

#### III.4.1. Modèle globale non linéaire

Il est constitué du modèle du robot et le modèle de la caméra. L'entrée de commande du processus est la vitesse de la caméra  $v_c$ . Dans ce cas, l'état du système peut être la situation de la caméra dans le repère de la cible:  $x = (P_x, P_y, P_z, \theta_x, \theta_y, \theta_z)$ . L'équation dynamique peut être approximée par:

$$x(k+1) = x(k) + T_e v_c(k) = f(x(k), v_c(k)) \quad (\text{II.16})$$

La sortie du modèle est présentée par les primitives visuelles prédites  $s_m$ , qui sont exprimées dans le plan image. Dans le cas d'une caméra perspective, l'équation de sortie peut être écrite comme:

$$s_m(k) = \begin{pmatrix} x(k) \\ y(k) \end{pmatrix} = \begin{pmatrix} X(k)/Z(k) \\ Y(k)/Z(k) \end{pmatrix} = g(X(k), Y(k), Z(k)) \quad (\text{II.17})$$

où  $(X, Y, Z)_{R_c}$  sont les coordonnées d'un point dans le repère de la caméra. La transformation homogène entre le repère de la caméra et le repère de la cible, notée  $l(x)$ , peut être facilement déduite à partir de la connaissance de la situation de la caméra  $x(k)$ . Si les coordonnées du point sont connues par rapport au repère de la cible,  $(X, Y, Z)_{R_o}$ , alors les coordonnées du point dans le repère de la caméra,  $(X, Y, Z)_{R_c}$  sont données par:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_{R_c} = \begin{pmatrix} R(x) t(x) \\ 0_{1 \times 3} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_{R_o} = l(x(k)) \quad (\text{II.18})$$

Finalement, nous obtenons:

$$s_m(k) = g \circ l(x(k)) \quad (\text{II.19})$$

Les équations (II.9) sont à présent complètement identifiées par (II.16) et (II.19).

Ce modèle dynamique combine des données 2D et 3D et il est donc approprié pour traiter les contraintes 2D et/ou 3D. Ces contraintes sont exprimées sur les états et/ou sur les sorties du modèle de prédiction, respectivement, et peuvent être ajoutées facilement, au problème d'optimisation II.6.

Le modèle global non linéaire dispose d'un grand domaine de validité et donc il peut être utilisé pour les grands déplacements. Néanmoins, la prédiction sur l'horizon de prédiction nécessite beaucoup de temps. De plus, ce modèle a besoin des données 3D qui est la situation de la cible dans le repère caméra, ainsi que le modèle de la cible. Pour réduire la connaissance 3D, une solution peut être proposée, elle se base sur la linéarisation du modèle en utilisant la matrice d'interaction.

### III.4.2. Modèle locale à base de la matrice d'interaction

Soit le point  $f = [u, v]$  exprimé en coordonnées pexiliques tel que:

$$u = \frac{\alpha}{\rho_u} x + u_0 \text{ et } v = \frac{\alpha}{\rho_v} y + v_0 \quad (\text{II.20})$$

où:

$\alpha$ : la distance focale qui est égale à 1;

$(\rho_u, \rho_v)$ : les dimensions d'un pixel avec  $\rho_u = \rho_v = 1$ ;

$(u_0, v_0)$ : les coordonnées de centre optique tel que  $u_0 = v_0 = 0$ .

La matrice d'interaction liée à  $f$  est donnée par [17]:

$$L_f = \begin{pmatrix} -\frac{1}{z} & 0 & \frac{u}{z} & uv & -(1+u^2) & v \\ 0 & -\frac{1}{z} & \frac{v}{z} & 1+v^2 & -uv & -u \end{pmatrix} \quad (\text{II.21})$$

La valeur  $Z$  (la profondeur) du point 3D est exprimée dans le repère de la caméra. La relation entre la vitesse de la caméra  $v_c$  et la variation temporelle des primitives visuelles  $\dot{f}$  est donnée par:

$$\dot{f} = L_f v_c(t) \quad (\text{II.22})$$

avec une simple approximation de premier ordre, on obtient:

$$f(k+1) = f(k) + T_e L_f(k) v_c(k) \quad (\text{II.23})$$

Pour éviter l'estimation de la profondeur à chaque itération, sa valeur  $Z^*$ , donnée à la position de référence, peut être utilisée. Par conséquent, la matrice d'interaction (II.21) devient  $\hat{L}_s$  et dépend seulement des mesures courantes des primitives visuelles.

En considérant ici les primitives visuelles  $f$  comme l'état  $x$  du système, on obtient le système d'équations décrivant la dynamique du système et les sorties (équations II.9):

$$\begin{cases} f(k+1) = f(k) + T_e \hat{L}_f(k) v_c(k) \\ s_m(k) = f(k) = h(f(k)) \end{cases} \quad (\text{II.24})$$

Cette approximation locale du modèle ne nécessite pas de données 3D mais seulement la valeur approximative de  $Z^*$ . Les contraintes 2D peuvent être prises en compte puisque les

états et les sorties sont des primitives visuelles. En d'autres termes, aucune information n'est disponible sur la situation de la caméra. Alors, les contraintes 3D ne peuvent pas être traitées.

### III.5. Les étapes à suivre pour la mise en œuvre de la CPV

Pour implémenter un régulateur prédictif visuel, il faut suivre les étapes suivantes:

1. A l'instant actuel  $k$ , la sortie du système  $s(k)$  est mesurée.
2. Le signal d'erreur est calculé:  $\varepsilon(k) = s(k) - s_m(k)$ .
3. Considérer que cette erreur est constante le long de l'horizon de prédiction.

$$\varepsilon(j) = \varepsilon(k) = s(k) - s_m(k) \forall j \in [k + 1; k + H_p]$$

4. La trajectoire désirée  $s_d(j)$  est calculée  $\forall j \in [k + 1; k + H_p]$  en utilisant (II.8).
5. La sortie de modèle  $s_m(k)$ , nécessaire pour la prédiction, est initialisée par la sortie mesurée  $s(k)$  du système pour garantir le retour dans la structure de la commande à modèle interne (CMI). Cependant, le retour pour la seconde boucle est assuré par le signal d'erreur  $\varepsilon(j)$ .
6. Finalement, l'algorithme d'optimisation sous contraintes détermine la séquence de commande optimale. Pour  $U(k + H_p + 1)$  à  $U(k + H_p - 1)$ , l'entrée de commande est constante et égale à  $U(k + H_c)$ . Appliquer juste la première commande  $U(k)$  et refaire la même procédure pour la prochaine période d'échantillonnage.

## IV. Conclusion

Dans ce chapitre, nous avons montré que la problématique de l'asservissement visuel 2D a été reformulée dans le plan image en termes de problème d'optimisation non linéaire sous contraintes sur un horizon fini. L'avantage de cette formulation est la capacité de prendre en compte explicitement des contraintes de visibilité et des contraintes 3D dans la synthèse de la loi de commande, qui présente un réel avantage pour les applications réelles.

Le modèle du système, connaissance a priori du procédé (robot et caméra), joue un rôle crucial, similaire à celui de la matrice d'interaction dans l'approche asservissement visuel 2D. La CPV, combinée à une structure CMI, se révèle robuste vis-à-vis d'erreurs de modélisation, erreurs quasi inévitables sur la caméra.

Le choix du modèle de prédiction de l'image a été discuté. Le modèle local peut être moins efficace que le modèle global pour les configurations difficiles mais il ne nécessite pas

de données 3D. Dans d'autres cas, si la donnée 3D est disponible, CPVà modèle globale donne des résultats satisfaisants pour toutes les configurations initiales.

Dans le chapitre suivant, nous testerons les performances de cette approche sur le plan simulation en le comparant avec l'asservissement visuel 2D classique.

---

# La commande prédictive visuelle pour un modèle local et les résultats de simulation

## I. Introduction

Le modèle de prédiction joue un rôle clé dans une stratégie de commande prédictive. Pour effectuer cette prédiction dans l'image, deux types de modèle peuvent être envisagées: *un modèle global non linéaire* et *un modèle local*. Dans le but d'éviter la complexité du modèle global non-linéaire (modèle de robot+caméra), utilisé pour la prédiction d'image, une stratégie de commande prédictive à modèle local est étudiée dans ce chapitre. Elle est basée sur la matrice d'interaction. En utilisant une fonction coût, basée sur l'erreur dans le plan image, le mouvement du robot est convergé vers la position désirée par la résolution d'un problème d'optimisation non linéaire sous contrainte de visibilité.

Dans ce chapitre, une architecture d'asservissement visuel est développée dans le but de tester l'algorithme de la commande prédictive. Premièrement, une structure de commande à base d'asservissement visuel 2D est présentée. Ensuite, la modélisation d'un système d'asservissement visuel 2D, en boucle ouverte, est étudiée. Il est constitué du modèle dynamique d'un bras manipulateur qui est à son tour composé d'un bras manipulateur et d'une boucle interne, ce qu'on appelle *dispositif de déplacement cartésien virtuel* (Virtual Cartesian Motion Device: VCMD) et d'un capteur visuel. Un régulateur proportionnel a été considéré. Une présentation d'un régulateur prédictif ainsi que les résultats de simulation, qui permettent de comparer entre les résultats obtenus à partir de l'asservissement visuel 2D en appliquant une loi proportionnelle et celles en appliquant une loi prédictive. Nous proposerons aussi une solution pour assurer la visibilité des primitives visuelles dans le champ visuel de la caméra pendant son mouvement. Nous terminerons ce chapitre par une conclusion.

## II. La structure de la commande à base d'AV 2D

En considérant un AV 2D d'un bras manipulateur à 6 d.d.l avec une configuration embarquée. Le but principal d'asservissement visuel 2D est de commander l'effecteur du bras manipulateur, telle que les primitives visuelles actuelles atteignent les primitives désirées [18]. Généralement, des primitives visuelles de type «points» sont utilisées dans l'asservissement visuel 2D et elles sont souvent présentées par les coordonnées de centre de gravité ou des coordonnées des coins [58]. La figure III.1 présente une boucle fermée d'asservissement visuel 2D.

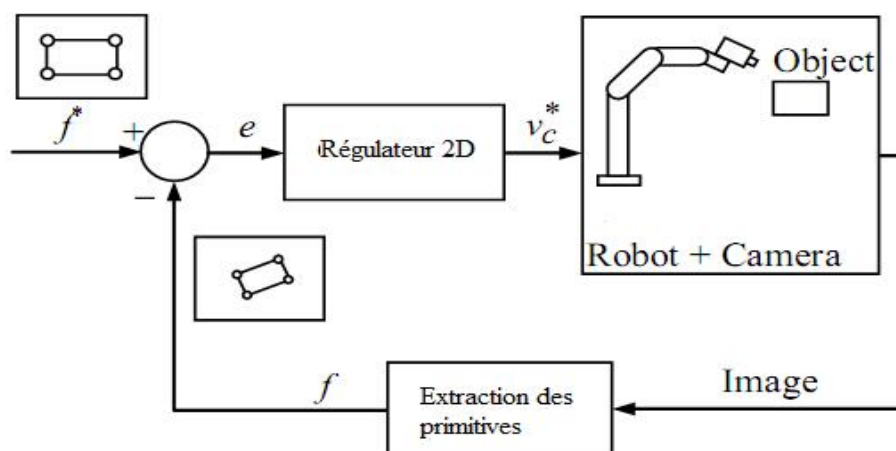


Fig III.1. La boucle fermée d'asservissement visuel 2D

Le vecteur des primitives de référence  $f^*$ , de la boucle fermée, est comparé au vecteur courant  $f$ , obtenu par le capteur visuel qui est composé d'une caméra et d'un bloc d'extraction des primitives visuelles. Le régulateur 2D traite l'erreur visuelle  $e$  et génère un signal de commande  $v_c^*$  (un vecteur de dimension  $6 \times 1$ ), qui représente le signal de la vitesse de référence de la caméra.

## III. La modélisation d'asservissement visuel 2D en boucle ouverte

L'architecture d'asservissement visuel 2D est composée de deux boucles: une boucle d'asservissement à niveau bas, pour commander la vitesse articulaire [42], ou bien une boucle interne, pour commander la vitesse de la caméra [59], et une autre externe, pour minimiser l'erreur entre les primitives actuelles et celles désirées en utilisant un régulateur 2D. La boucle interne est considérée rapide, parce qu'elle n'a besoin que d'information d'encodeur comme sortie du capteur. En contre partie, la boucle externe a besoin des informations extraites de

l'image acquise par la caméra. Dans ce travail, il est considéré que les vitesses articulaires sont commandées par une boucle d'asservissement articulaire. Cette caractéristique est disponible sur presque tous les robots industriels [52]. La figure II.2 présente ces deux boucles d'AV 2D.

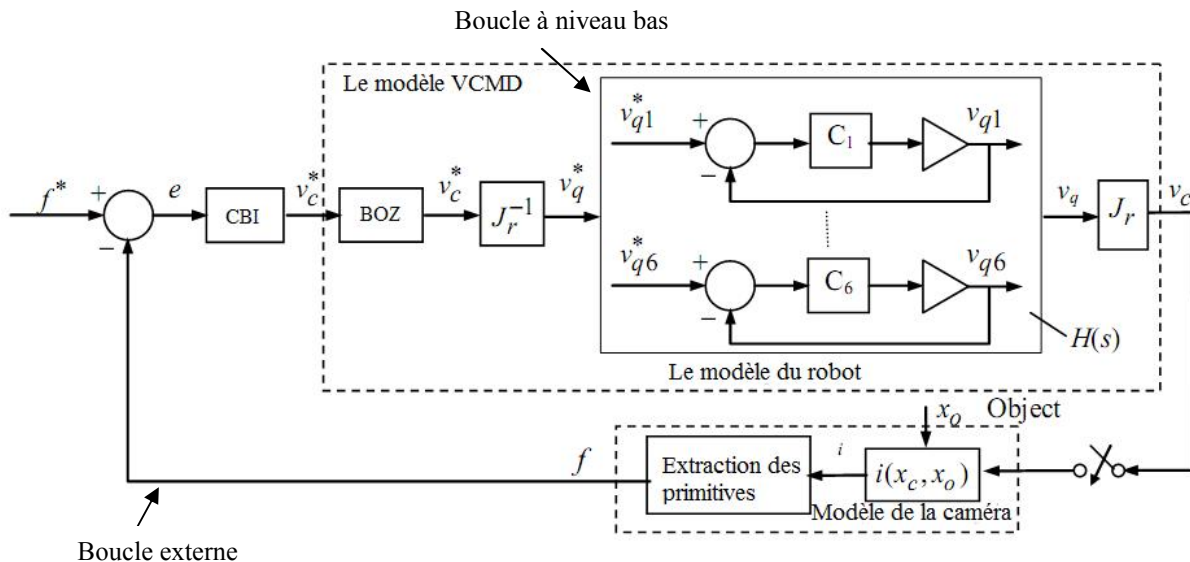


Fig III.2. Système de commande en asservissement visuel 2D.

Le modèle en boucle ouverte peut être décomposé en deux parties. La première inclut la dynamique du robot commandé, nommée *Dispositif de Mouvement Cartésien Virtuel* (VCMD : Virtual Cartesian Motion Device en anglais) [1]. La deuxième partie introduit le modèle de la caméra. Ces deux parties sont représentées dans la figure ci-dessus. Le vecteur de la vitesse de référence  $v_c^*$  présente l'entrée du modèle VCMD. Il est exprimé dans l'espace cartésien. Il doit être converti vers l'espace articulaire, dans le but de l'appliquer à 6 boucles de commande en vitesses articulaires du robot. Cette transformation est réalisée en utilisant le jacobien inverse du robot  $J_r^{-1}$ , ce qui donne  $v_q^*$  (la vitesse articulaire de référence). Chaque boucle de vitesse articulaire contient un régulateur C, qui assure les buts désirés des boucles à niveau bas. La vitesse articulaire  $v_q$  est convertie vers l'espace cartésien en utilisant le jacobien de robot  $J_r$ . Finalement, à partir de la vitesse de la caméra  $v_c$ , la situation  $x_c$  de la caméra peut être obtenue en utilisant une action intégrale.

Les régulateurs de vitesse  $C_i, i = \overline{1,6}$ , peuvent être conçus de tel sort que chaque axe articulaire soit découplé. Cette caractéristique est utilisée sur tous les robots industriels [52]. Le vecteur de référence  $v_c^*$  est élaboré par le régulateur 2D, dans le but de minimiser l'erreur  $e$  entre les primitives désirées  $f^*$  et les primitives mesurées  $f$ .

Le modèle d'asservissement visuel en boucle ouverte peut être vu comme un système multivariable tel que: les signaux d'entrée sont les composantes de  $v_c^*$  et les signaux de sorties sont les coordonnées des primitives  $f$ .

### III.1. Le modèle VCMD

Nous supposons que l'architecture de commande en AV 2D sépare entre la cinématique du robot et le régulateur visuel, ce qui permet de voir le robot comme un dispositif de mouvement cartésien. Il est appelé *Dispositif de Mouvement Cartésien Virtuel* (VCMD : Virtual Cartesian Motion Device en anglais). Le modèle VCMD est composé d'un bras manipulateur avec une boucle de commande interne. Il est non-linéaire parce que le jacobien du robot et son modèle dynamique sont en fonction des positions articulaires  $x_q$ .

#### *Hypothèses*

1. La configuration du robot est souvent choisie loin des singularités de  $J_r$ , alors  $J_r$  varie lentement avec  $x_q$ . Nous supposons que, pendant une période d'échantillonnage, le robot se déplace sur une petite distance. Ce qui implique que  $J_r$  est constante entre deux instants d'échantillonnage.
2. Le modèle dynamique du bras manipulateur est aussi non-linéaire et dû aux effets de frottement, de forces de Coriolis, centrifuge et gravitationnelle. Ces effets non-linéaires sont importants lorsque le robot se déplace avec une grande vitesse. Les régulateurs de la boucle interne,  $C_1$ - $C_6$  représentés sur la figure III.2, sont conçus dans le but d'éliminer ces effets non-linéaires qui sont considérés comme des perturbations. La matrice d'inertie est aussi non linéaire. Nous supposons que l'inertie des articulations varie lentement avec la position  $x_q$  du robot, donc la matrice d'inertie peut être considérée constante autour d'une position donnée du robot.

Par conséquent, il est possible de linéariser le modèle VCMD au voisinage de la position  $x_q$  du robot où le modèle dynamique du robot et  $J_r$  peuvent être constants. Compte tenu de ces hypothèses, le robot commandé en vitesse, peut être modélisé par une matrice de transfert  $H(s)$  qui est valable autour de la position d'équilibre  $x_q$  du robot:

$$v_q(s) = H(s)v_q^*(s) \quad (\text{III.1})$$

La matrice de transfert  $H(s)$  est considérée comme étant un intégrateur pur (chapitre I, section III.3.3.1). Si les régulateurs de vitesse  $C_i$ ,  $i = \overline{1,6}$ , de la boucle interne sont conçus

dans le but de commander la vitesse de la caméra  $v_c$ , le torseur cinématique de la caméra peut être commandé dans le système de coordonnées de la caméra, dans le but d'obtenir un modèle diagonale d'un bras manipulateur à 6 degrés de libertés [59]. Dans ce cas, il est possible d'assurer que chaque axe articulaire soit découplé. Le régulateur de vitesse est généralement proportionnel tel que  $C_i(s) = k_v$ , le VCMD peut être exprimé, sous forme d'un modèle continu suivant:

$$v_c(s) = H_0(s)v_c^*(s) = \frac{k_v}{s+k_v} I_6 v_c^*(s) \quad (\text{III.2})$$

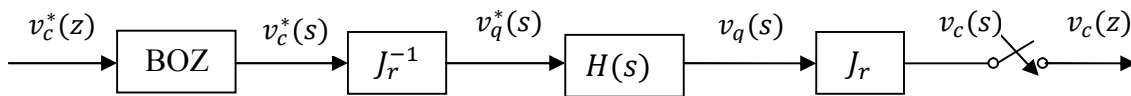
où  $I_6$  est une matrice identité de dimension 6.

En tenant compte de la structure de VCMD de la figure III.2 et de l'équation (III.2), il est nécessaire de déplacer le bloc BOZ (Bloqueur d'Ordre Zéro) de la figure III.2 après le régulateur 2D dont le but de convertir  $v_c^*(k)$  vers une valeur continue  $v_c^*(s)$  (figure III.3). Cette vitesse est considérée comme une entrée de la boucle ouverte. Le modèle VCMD est linéaire autour de point d'équilibre  $x_q$ , sa forme discrète obtenue est:

$$G(z) = (1 - z^{-1})Z\left(\frac{H_0(s)}{s}\right) \quad (\text{III.3})$$

où  $Z$  présente la transformée en  $z$ .

Le modèle (III.3) est une excellente approximation pour la conception du régulateur 2D [47].



**Fig III.3.** Le modèle VCMD linéaire.

### III.2. Le modèle du capteur visuel

En général, le capteur visuel est composé d'une caméra et d'un bloc de traitement d'images utilisé pour extraire les primitives à partir d'une image.

Soit  $x_0 \in R^6$  et  $x_c \in R^6$ , la position et l'orientation de l'objet et de la caméra, respectivement.

La caméra est modélisée par :

$$i: R^6 \times R^6 \rightarrow R^{2l} \quad (\text{III.4})$$

Nous considérons que l'objet est caractérisé par  $l'$  primitives visuelles de type *points*, définies comme suit

$$f = [f_1^T \dots f_l^T]^T \quad (\text{III.5})$$

où la primitive  $f_i$  présente les coordonnées pixeliques d'un point dans le plan image:

$$f_i = [u_i \ v_i]^T \quad (\text{III.6})$$

Les primitives visuelles sont obtenues en utilisant le modèle suivant de la caméra:

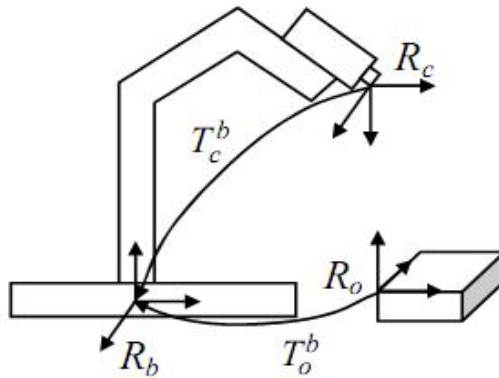
$$f = i(x_c, x_o) \quad (\text{III.7})$$

Le bloc de traitement d'images est modélisé comme suit:

$$f = g(i(x_c, x_o)) \quad (\text{III.8})$$

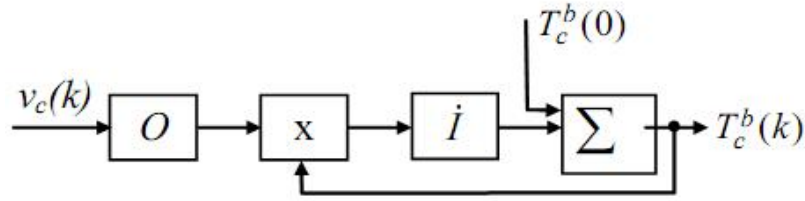
où  $g$  est une fonction qui modélise l'algorithme d'extraction des primitives visuelles.

Afin d'obtenir le modèle de capteur visuel, il faut d'abord déterminer les repères liés à la base du robot  $R_b$ , à la caméra  $R_c$  et à l'objet  $R_o$  (la figure III.4).



**Fig III.4.** Les repères: base du robot, caméra et objet.

Soient  $T_c^b$  et  $T_o^b$  les transformations homogènes entre les repères  $R_c$  et  $R_b$ ,  $R_o$  et  $R_b$ , respectivement. Pour obtenir la transformation homogène  $T_c^b(k)$ , qui dépend du mouvement de la caméra à l'instant discret  $k$ , un bloc de transformation homogène (Frame Motion : FM) à partir de [60] est utilisé (la figure III.5).



**Fig III.5.** Bloc de transformation homogène.

Le torseur cinématique de la caméra  $v_c = [v_x \ v_y \ v_z \ w_x \ w_y \ w_z]^T$  est traité par l'opérateur  $O$ . Ce qui donne la matrice suivante:

$$O(v_c) = \begin{bmatrix} 0 & -w_z & w_y & v_x \\ w_z & 0 & -w_x & v_y \\ -w_y & w_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{III.9})$$

Une multiplication de la matrice  $O(v_c)$  avec la situation précédente de la caméra par rapport au repère base et une action intégrale (bloc  $I$  de la figure III.5), le résultat est ajouté à  $T_c^b(0)$ . De cette manière  $T_c^b(k)$  (situation courante de la caméra/repère base) est obtenue et multipliée par la situation de l'objet  $x_o^b$  par rapport au repère base  $R_b$ , dans le but d'avoir les primitives visuelles  $x_f^c$  dans le repère caméra. Considérant  $\rho_x$  et  $\rho_y$ , les dimensions d'un pixel et  $(u_0, v_0)$ , les coordonnées de centre de l'image. Les coordonnées des points exprimés en pixel sont obtenues en appliquant la projection perspective :

$$u_i = \frac{x_i}{z_i} \rho_x + u_0 ; v_i = \frac{y_i}{z_i} \rho_y + v_0 \quad (\text{III.10})$$

Le modèle de capteur visuel trouvé a comme entrée, le torseur cinématique de la caméra  $v_c(k)$ , et a comme sortie les coordonnées des points  $f(k)$  dans le plan image. Ce modèle avec le modèle VCMD présentent le modèle en boucle ouverte d'asservissement visuel.

#### IV. Le régulateur proportionnel

L'objectif principal de l'asservissement visuel 2D est de minimiser l'erreur dans le plan image:

$$e(k) = f^* - f(k) \quad (\text{III.11})$$

où  $f^*$  est le vecteur des primitives désirées et qui est utilisé comme une consigne dans l'algorithme de commande.

On dérive (III.7) par rapport au temps, on obtient une relation entre  $\dot{f}$  et la vitesse de la caméra  $v_c \in R^6$  :

$$\dot{f} = \frac{\partial i}{\partial x_c} \dot{x}_c + \frac{\partial i}{\partial x_0} \dot{x}_0 \quad (\text{III.12})$$

Pour un objet statique, l'équation (III.12) devient:

$$\dot{f} = L(f, z)v_c \quad (\text{III.13})$$

où  $L = [L_1^T, L_2^T, \dots, L_l^T]^T$  est la matrice d'interaction.  $L_i$  est la matrice d'interaction de l' $i^{\text{ème}}$  point de coordonnées  $(u_i, v_i)$  dans le plan image, elle est définie comme suit [17]:

$$L_i = \begin{pmatrix} -\frac{\alpha}{z_i} & 0 & \frac{u_i}{z_i} & \frac{u_i v_i}{\alpha} & -\frac{(u_i^2 + \alpha^2)}{\alpha} & v_i \\ 0 & -\frac{\alpha}{z_i} & \frac{v_i}{z_i} & \frac{\alpha^2 + v_i^2}{\alpha} & -\frac{u_i v_i}{\alpha} & -u_i \end{pmatrix} \quad (\text{III.14})$$

où  $\alpha$  est la distance focale.  $z = [z_1 \dots z_m]^T$  est un vecteur telle que  $z_i$  présente la distance de la  $i^{\text{ème}}$  primitive au repère caméra.

Considérant  $v_c^*$  la sortie du régulateur 2D qui permet d'assurer une décroissance exponentielle de l'erreur :

$$\dot{e} = -\lambda e, \quad \lambda > 0 \quad (\text{III.15})$$

La loi proportionnelle suivante est obtenue :

$$v_c^* = -\lambda L^{-1} e(t) \quad (\text{III.16})$$

avec  $L \in R^{l \times 6}$  n'est pas une matrice carrée.  $L^{-1}$  de l'équation (III.16) est remplacée par sa pseudo inverse  $L^+ = (L^T L)^{-1} L^T$ .

Il faut assurer que le nombre minimum de points nécessaires pour obtenir un torseur cinématique défini de manière unique par la pseudo inverse de la matrice d'interaction  $L^+$ , est égale à 4 ( $l=4$ ) [26].

Le régulateur 2D, est conçu en utilisant, pas seulement une loi de commande proportionnelle, mais aussi des techniques de commande avancées, qui nécessitent la

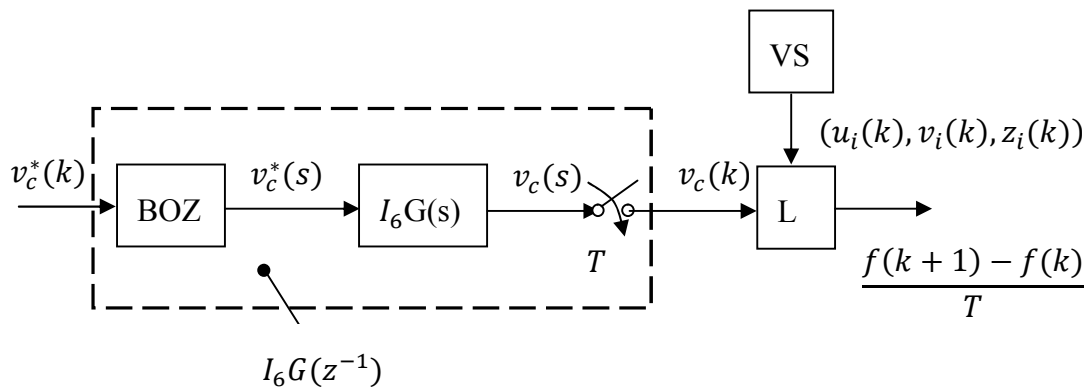
connaissance du modèle en boucle ouverte de système commander en asservissement visuel. La commande prédictive est l'une de ces techniques.

## V. Le régulateur prédictif

La commande prédictive est appliquée pour réaliser des tâches d'asservissement visuel 2D, dans le but de commander un bras manipulateur à 6 d.d.l, avec une configuration embarquée de la caméra. L'objet a une géométrie connue et présenté par 4 points.

Le problème principal de la commande prédictive est de trouver un modèle qui permet de prédire le comportement futur du système le long d'un horizon de prédiction  $H_p$ . Le modèle est composé du modèle du robot avec le modèle de la caméra, ce qui donne un modèle global non-linéaire. Ce modèle complexe a été utilisé pour prédire l'évolution des primitives visuelles, dans le but de trouver une séquence de commande sur un horizon de prédiction  $H_p$ . Il génère des difficultés dans le développement des prédictions. Pour résoudre ce problème, un modèle local a été proposé. Il se base sur la relation entre la vitesse de la caméra  $v_c$  et les variations dans le temps des primitives visuelles  $\dot{f}$ , donnée par la matrice d'interaction.

En prenant en considération le modèle VCMD de la boucle interne et par la discrétisation de la relation (III.13), nous obtenons le modèle local de prédiction (figure III.6).



**Fig III.6.** Le modèle utilisé pour la prédiction.

A chaque période d'échantillonnage, le capteur visuel donne les coordonnées des primitives points  $(u_i(k), v_i(k))$ ,  $i = \overline{1,4}$ .

En utilisant la forme discrète de l'équation (III.13) et le modèle discret (III.3) de la boucle interne avec une période d'échantillonnage égale à T, on obtient :

$$f(k+1) = f(k) + TL_k G(z^{-1})v_c^*(k) \quad (\text{III.17})$$

où  $f(k+1)$  indique que la prédiction est calculée à l'instant discret  $k$ .

$L_k$  est la matrice d'interaction calculée en utilisant les primitives points  $(u_i(k), v_i(k))$  acquis à l'instant courant  $k$ . Ils sont obtenus par une caméra et un détecteur de points. On suppose qu'il est possible de calculer la profondeur  $z_i(k)$  des points pour chaque période d'échantillonnage.

La relation (III.17) présente la première étape pour prédire l'évolution des primitives visuelles. Déplaçant le modèle de prédiction (III.17) d'un pas en avant. Par récurrence on obtient:

$$\begin{aligned} f(k+2) &= f(k+1) + TL_{k+1}G(z^{-1})v_c^*(k+1) \\ f(k+i) &= f(k+i-1) + TL_{k+i-1}G(z^{-1})v_c^*(k+i-1) \\ f(k+H_p) &= f(k+H_p-1) + TL_{k+H_p}G(z^{-1})v_c^*(k+H_p) \end{aligned} \quad (\text{III.18})$$

Le calcul des primitives  $f(k+i)$  se base sur les primitives prédites à l'étape précédente  $f(k+i-1)$ . Ils sont aussi utilisées pour calculer la matrice d'interaction  $L_{k+i-1}$ . Il dépendent aussi de la commande future  $v_c^*(k+i-1)$ . La prédiction est initiée par la valeur de  $f(k)$  obtenue à l'instant  $k$  par le capteur visuel, en utilisant un détecteur de points d'intérêt.

Les erreurs de commande, exprimées dans l'espace cartésien, sont définies par:

$$e_c(k+i) = L^+(f^* - f(k+i)), i = \overline{1-H_p} \quad (\text{III.19})$$

La fonction coût à minimiser est définie sous forme quadratique. Elle est en fonction de l'erreur, exprimée dans l'espace cartésien, et le vecteur de commande sur l'horizon de commande  $H_c$ :

$$J = \frac{1}{2} \sum_{i=1}^{H_p} e_c^T(k+i) Q e_c(k+i) + \frac{1}{2} \sum_{i=0}^{H_c-1} v_c^{*T}(k+i) R v_c^*(k+i) \quad (\text{III.20})$$

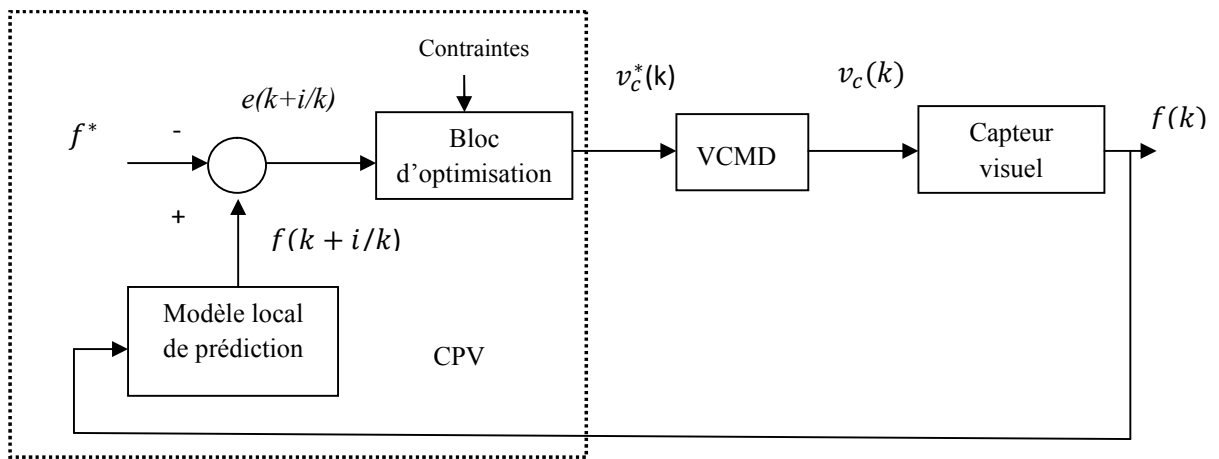
où  $Q$  et  $R$  sont des matrices symétriques définies positives.

Les limites de l'image présentent la contrainte de visibilité. Elle permet d'assurer que les primitives visuelles restent visibles. Elle est définie comme suit :

$$(u_i(k), v_i(k)) \in [u^{min}, v^{min}; u^{max}, v^{max}] \quad (\text{III.21})$$

Deux autres contraintes liées au robot sont souvent ajoutées. Il s'agit des contraintes sur la vitesse des actionneurs et des contraintes sur les butées articulaires.

La figure III.7 présente le schéma bloc de la commande prédictive visuelle avec un modèle local. Elle se compose, principalement, d'un régulateur prédictif à base d'image, qui est à son tour composé d'un modèle local de prédiction et d'un bloc d'optimisation. Le modèle de prédiction est utilisé pour prédire les primitives visuelles futures dans le but que les primitives courantes  $f$  atteignent les primitives de références  $f^*$ , qui sont considérées comme consigne. La commande prédictive consiste à calculer une séquence de commande optimale par la minimisation d'une fonction coût avec la prise en compte des contraintes en utilisant le *bloc optimisation*.



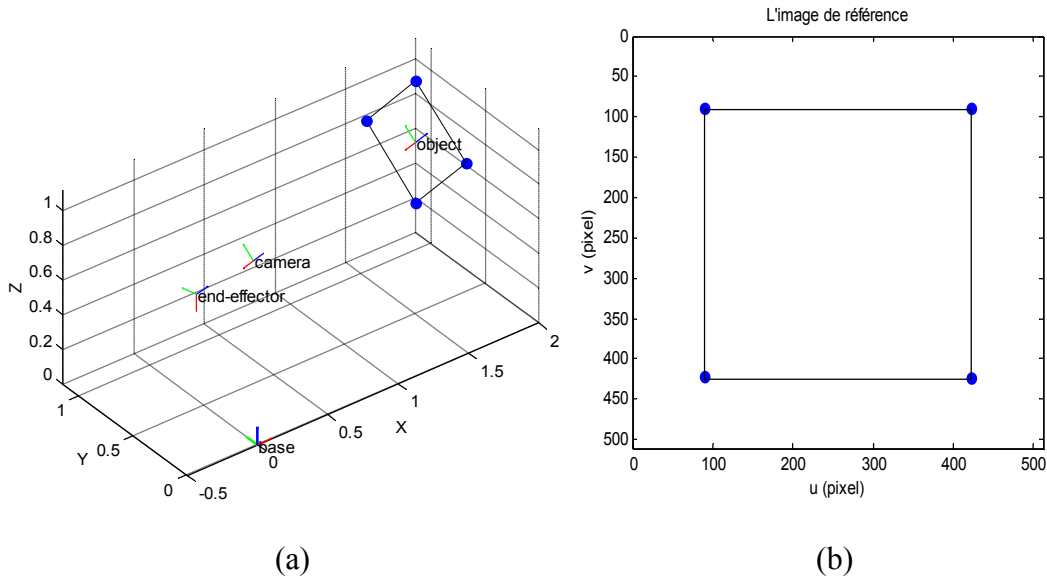
**Fig III.7.** Le schéma bloc de la commande prédictive visuelle en utilisant un modèle local.

## VI. Les résultats de simulation

La tâche de la commande consiste à positionner une caméra par rapport à une cible composée de 4 points. Ces 4 points forment un carré dans l'espace cartésien (figure III.8. (a).) La figure III.8. (b).présente l'image de référence. Les coordonnées des 4 points dans cette image sont égales à:

$$f^* = \begin{bmatrix} 422 & 422 & 89 & 89 \\ 89 & 422 & 422 & 89 \end{bmatrix}; \quad (\text{III.22})$$

Les paramètres intrinsèques de la caméra sont  $\rho_u = \rho_v = 1$ ,  $u_0 = v_0 = 256$ . Pour le modèle VCMD,  $k_v = 160$  et la période d'échantillonnage  $T=4/k_v$ .



**Fig III.8.** La situation de l'objet : (a): la situation désirée dans l'espace cartésien. (b) : l'image de référence 2D.

## VI.1. L'asservissement visuel 2D

### VI.1.1. Les résultats obtenus en fonction du choix de la matrice d'interaction estimée $\hat{L}_s$

L'AV 2D en boucle fermée présentée par la figure III.2 est testée sur le plan simulation en utilisant une loi proportionnelle avec un gain  $\lambda = 0.125$ . Les primitives désirées sont considérées comme une consigne. Pour la même situation initiale de la caméra par rapport au repère base, les coordonnées des points dans le pan image correspond à cette situation sont données comme suit:

$$f(0) = \begin{bmatrix} 224 & 399 & 287 & 112 \\ 70 & 183 & 358 & 245 \end{bmatrix} \quad (\text{III.23})$$

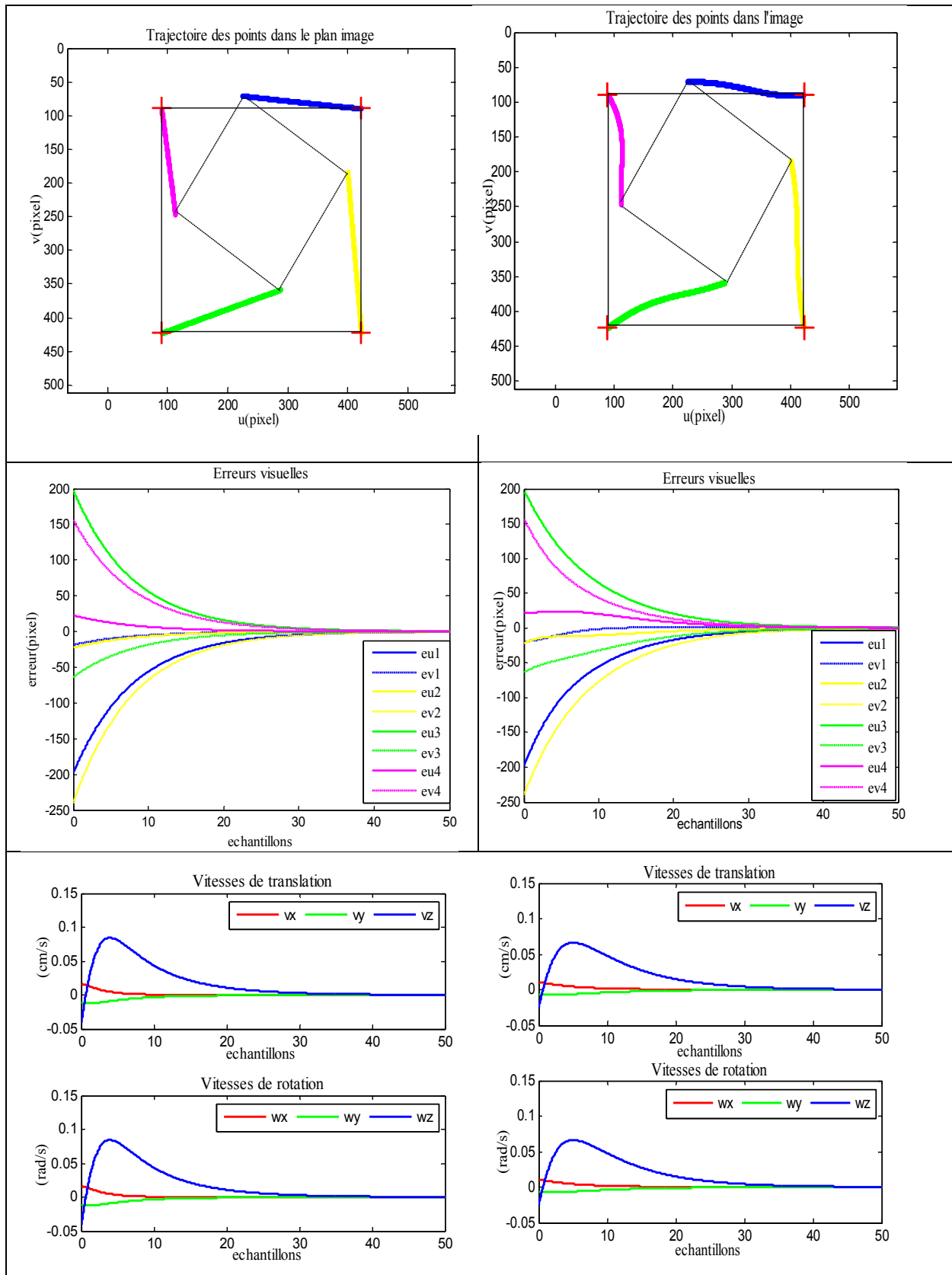
Il existe plusieurs choix pour la matrice d'interaction estimée  $\hat{L}_s$ . Elle peut être choisie comme suit:

- $\hat{L}_s = \hat{L}_{(s(t), Z(t))}$ , notée  $L_c$ : la matrice d'interaction est mise à jour à chaque itération.
- $\hat{L}_s = \hat{L}_{(s(t), Z^*)}$ , notée  $L_p$ : la profondeur est calculée ou mesurée à la position de référence notée  $Z^*$ . La matrice d'interaction varie seulement en fonction des valeurs actuelles des primitives visuelles.
- $\hat{L}_s = \hat{L}_{(s^*, Z^*)}$ , notée  $L_d$ : la matrice d'interaction est constante et correspond à sa valeur à la position de référence.

- $\hat{L}_s = \frac{1}{2} (\hat{L}_{(s^*, z^*)} + L_{(s(t), z(t))})$ , notée  $L_m$ : la matrice d'interaction est la moyenne entre la matrice d'interaction constante et courante.

Nous illustrons le comportement du schéma de commande pour chaque matrice. Le but de la tâche est de positionner la caméra de telle sorte que le carré soit centré dans l'image (figure III.8. (b)). Les résultats présentent la trajectoire des points dans le plan image, l'erreur visuelle pour chaque couple (u,v) ainsi que l'évolution de la commande (vitesse de translation et de rotation). Les résultats suivants (figure III.9) sont obtenus. La croix rouge présente la position désirée pour chaque point.

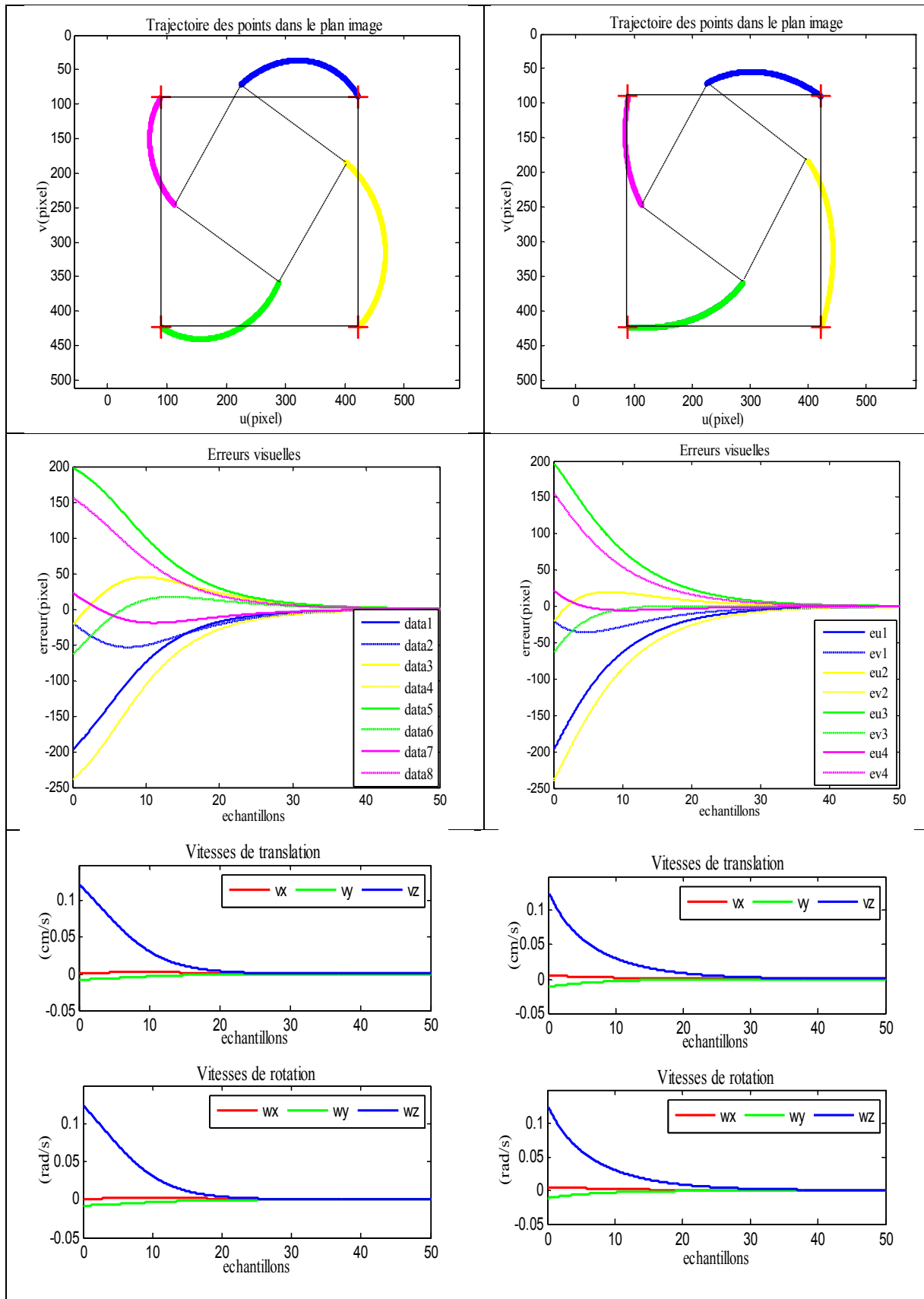
- Dans la figure III.9. (1), avec  $L_c$ : le mouvement de la caméra est une combinaison entre un mouvement de translation et un mouvement de rotation autour de son axe optique. Le schéma de commande tente d'assurer une décroissance exponentielle de l'erreur  $e$ . Les points dans l'image, s'effectuent des trajectoires sous forme des lignes droites à partir des positions initiales vers les positions désirées, comme mentionné dans [17]. Au début de l'asservissement, la vitesse de la caméra est grande. Ce qui signifie que la trajectoire de la caméra est loin d'être une ligne droite.
- Avec  $L_p$ : les résultats sont, approximativement, les mêmes que les résultats obtenus dans le premier cas, (figure III.9. (2)). La trajectoire des primitives visuelles tend vers une ligne droite.
- Avec  $L_d$ : la caméra se déplace vers la cible et tourne autour de son axe optique. Le système converge (l'erreur décroît vers zéro) et les points suivent une trajectoire circulaire. Ce choix peut entraîner, pour certaines positions initiales (la position initiale est loin de la position désirée ou pour un mouvement de rotation de la caméra), la sortie de certaines primitives du champ visuel de la caméra, (figure (III.9. (3))).
- Dans la figure III.9. (4), avec  $L_m$ : la trajectoire des primitives visuelles est circulaire. La vitesse de la caméra ne contient pas de grandes oscillations. Ce choix est performant en pratique.



(1):  $L_c$

(2):  $L_p$

**Fig III.9.** Les résultats de simulations pour un asservissement visuel 2D pour différents choix de  $\hat{L}$ .



(3) :  $L_d$

(4) :  $L_m$

**Fig III.9 .** Les résultats de simulations pour un asservissement visuel 2D pour différents choix de  $\hat{L}$ .

### VI.1.2. L'inconvénient d'asservissement visuel 2D

On prend en considération 2 situations initiales différentes de la caméra par rapport au repère base. Les coordonnées pixeliques des points pour chaque situation de la caméra, sont égales à:

$$f^{(1)}(0) = \begin{bmatrix} 408 & 271 & 188 & 308 \\ 244 & 377 & 279 & 162 \end{bmatrix} \quad (\text{III.24})$$

$$f^{(2)}(0) = \begin{bmatrix} 139 & 20 & 123 & 232 \\ 167 & 60 & 16 & 111 \end{bmatrix} \quad (\text{III.25})$$

Les coordonnées de référence sont:

$$f^* = \begin{bmatrix} 182 & 22 & 45 & 188 \\ 292 & 318 & 210 & 184 \end{bmatrix} \quad (\text{III.26})$$

Nous avons choisi  $\hat{L}_s = L_m$ , comme une estimée de la matrice d'interaction.

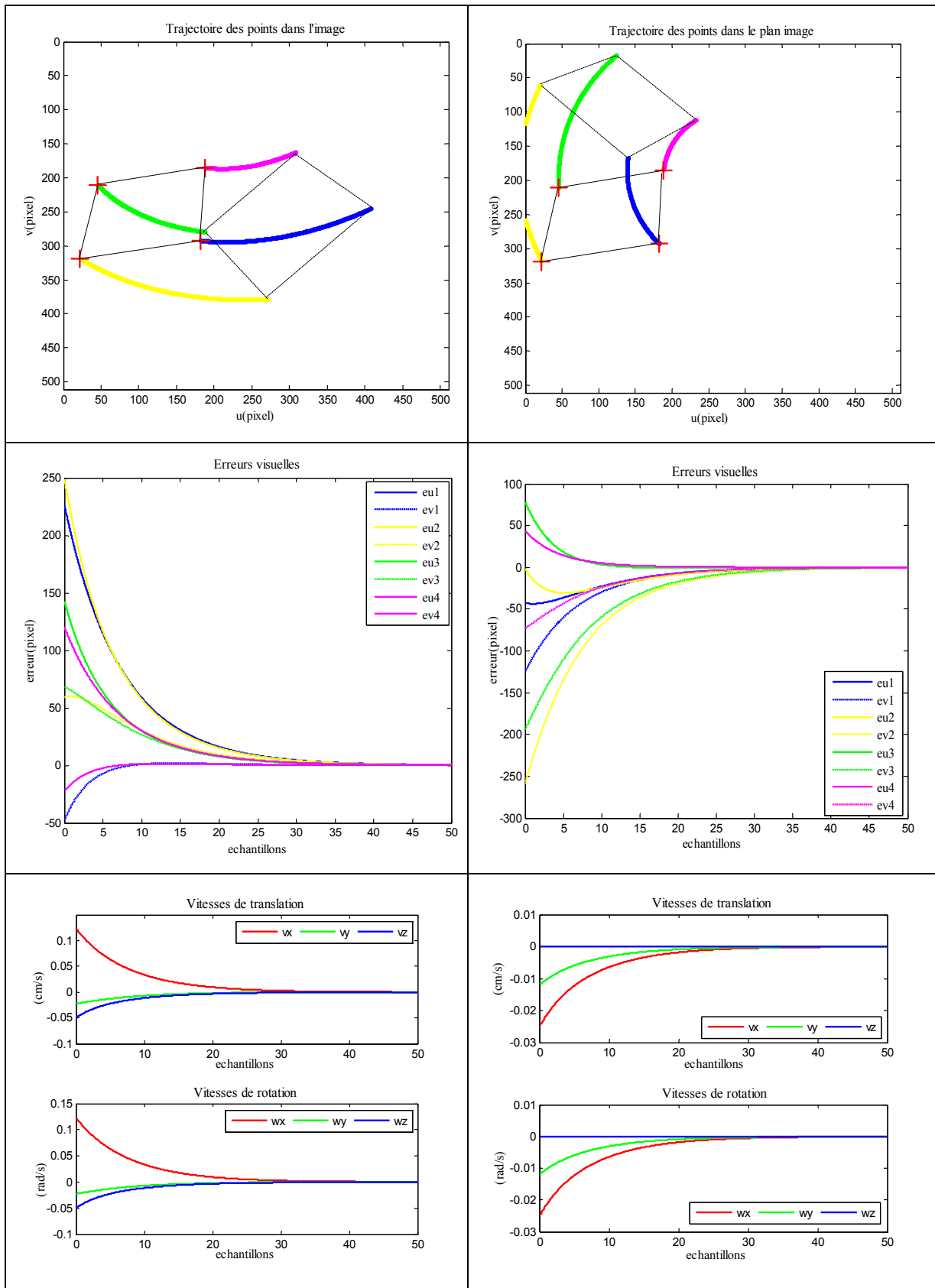
Nous obtenons les résultats présentés dans la figure III.10. Nous remarquons que le signal de la commande convergence et les primitives visuelles atteignent les primitives désirées pour les deux situations (1) et (2). Pour la situation (2), les primitives visuelles sortent de l'image (la trajectoire présentée en jaune). Dans le cas pratique, les résultats obtenus à la situation (2) conduisent à l'interruption de la tâche et l'arrêt du bras manipulateur. C'est parmi les limites de l'asservissement visuel 2D classique [19].

## VI.2. La résolution du problème

### VI.2.1. Une combinaison entre les différents choix de la matrice d'interaction estimée

Pour résoudre ce problème, nous proposons une solution qui permet aux primitives visuelles de rester dans l'image et ne pas sortir de champ visuel de la caméra pendant son mouvement. Pour cela, nous faisons un test sur le choix des matrices d'interactions estimées en fonction de la position des primitives visuelles dans l'image.

Si les points sont dans l'image et loin de la limite, on choisit la matrice d'interaction  $\hat{L}_s = L_m$  qui permet une convergence rapide vers les primitives désirées. Dès que les points atteignent la limite de l'image, nous choisissons  $\hat{L}_s = L_c$ , pour forcer les points à suivre une ligne droite dans le but d'éviter ces sorties.

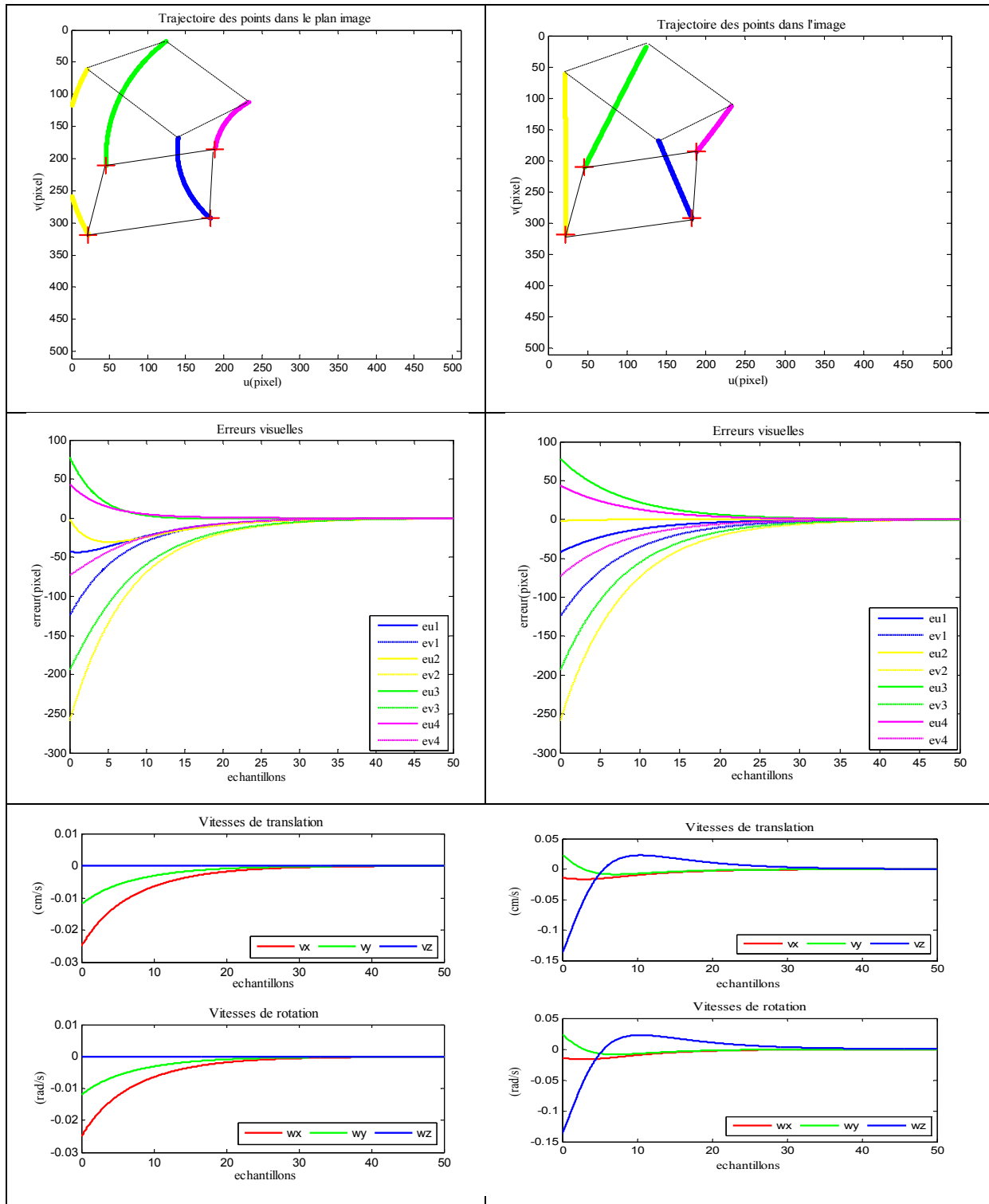


Situation (1)

Situation (2)

Fig III.10 L'asservissement visuel 2D classique: situation (1) et situation (2).

Nous considérons toujours la situation (2). Les résultats obtenus sont présentés dans la figure III.11.



(a)

(b)

**Fig III.11.** L'asservissement visuel 2D classique avec la prise en compte de la contrainte de visibilité. (a): avec la sortie des points. (b): sans la sortie des points.

Les résultats présentent la validité de la solution proposée. Nous remarquons d'après la figure III.11 (2) que le signal de commande converge et les primitives visuelles restent dans l'image pendant l'exécution de la tâche et atteignent les primitives désirées. Nous remarquons, aussi, que le signal d'erreur tend vers zéro.

### VI.2.2. La commande prédictive visuelle

La commande prédictive visuelle est testée sur le plan simulation en réalisant une extension de la toolbox IBVS [60]. La simulation est faite sous Matlab, elle se base sur le schéma bloc de la figure III.7 en utilisant un modèle local de prédiction.

A chaque instant «k», les prédictions (III.17) et (III.18) sont calculées et par la minimisation de la fonction coût (III.20), nous obtenons une séquence de commande future. En utilisant le principe de l'horizon glissant, seulement la première commande  $v_c^*(k)$  qui sera appliquée au système.

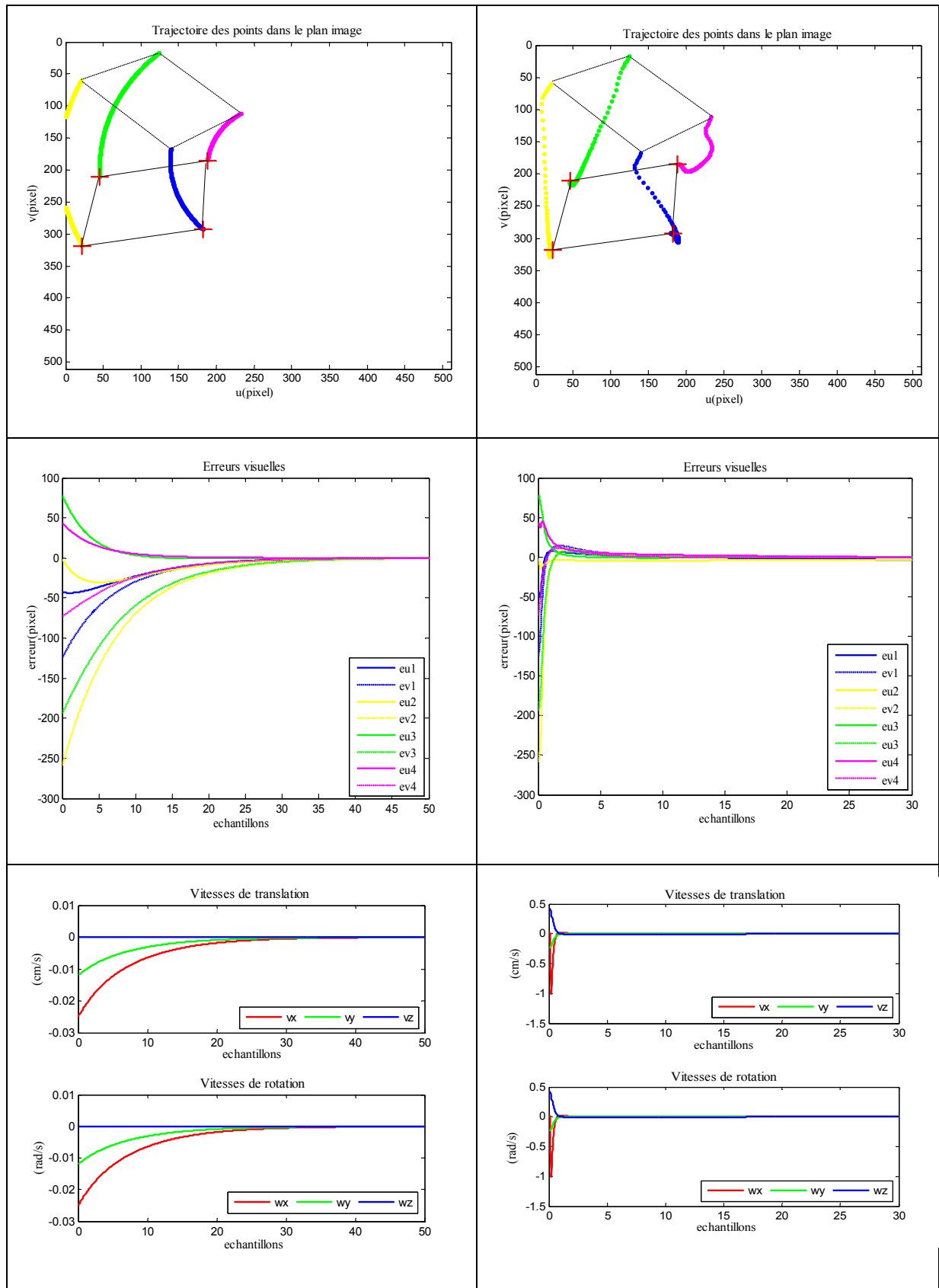
Les paramètres du régulateur sont :  $H_p = 4$ ,  $H_c = 1$ ,  $Q = I_8$ ;  $R = \partial * I_6$  avec  $\partial = 0.01$ . La contrainte de visibilité est défini par :  $u_{min} = v_{min} = 0$  et  $u_{max} = v_{max} = 512$ .

Les résultats de simulation présentés dans la figure III.12, nous permet de remarquer que malgré le mouvement de rotation effectué par la caméra autour de son axe optique, les primitives visuelles restent dans la fenêtre définie par  $u \in [0,512]$  et  $v \in [0,512]$ . Nous remarquons que la vitesse de la caméra selon l'axe x est responsable à la sortie des point. Au début de la simulation, le signal de commande diminue (la composante selon l'axe x) dans le but d'empêcher la sortie des points. Dès que les points s'éloignent de la limite de l'image, les vitesses augmentent dans le but d'atteindre la situation désirée. Les primitives visuelles atteignent les primitives désirées et l'erreur visuelle tend vers zéros.

Le régulateur prédictif visuel commande les primitives visuelles  $f^0$  (la position de départ du robot) vers les primitives désirées  $f^*$  (la position souhaitée) dans le but de réaliser une tâche de positionnement.

Autrement dit, à l'aide d'une fonction coût basée sur les erreurs dans le plan image, le mouvement du robot converge grâce à la résolution d'un problème d'optimisation non linéaire sous contrainte de visibilité qui est dû au mouvement de la caméra.

Les résultats obtenus présentent des meilleures performances. Les points convergent rapidement vers leurs positions de références, en assurant leur présence dans le champ visuel de la caméra (satisfaction de la contrainte 2D).



(a)

(b)

**Fig.III.12.** Les résultats de simulation en utilisant la commande prédictive visuelle.

(a): avec la sortie des points, (b): sans la sortie des points.

## VII. Conclusion

Dans ce chapitre, nous avons présenté une approche de modélisation qui permet d'obtenir un modèle multivariable en boucle ouverte d'un système d'asservissement visuel 2D. Les vitesses articulaires du robot sont contrôlées par des boucles internes donnant un modèle linéaire et découplé.

En utilisant le modèle VCMD et le modèle du capteur visuel, le modèle en boucle ouverte d'asservissement visuel a été obtenu.

La simulation d'un modèle dynamique d'un bras manipulateur commandé en utilisant l'asservissement visuel 2D a été présentée. La modélisation prend en compte le modèle du capteur visuel qui est constitué d'une caméra et d'un bloc d'extraction des caractéristiques visuelles et le modèle VCMD. Les résultats de simulations du modèle dynamique en boucle fermée sont présentés.

Une stratégie de commande prédictive est présentée pour résoudre le problème d'asservissement visuel, basée sur un modèle local. Nous avons utilisé le modèle du robot avec le modèle du capteur visuel pour développer le modèle de prédiction et en récurrence nous prédirons les primitives visuelles futures.

A l'aide d'une fonction cout basée sur les erreurs dans le plan image, le mouvement du robot converge grâce à la résolution d'un problème d'optimisation non linéaire sous contrainte de visibilité due au mouvement de la caméra. Les résultats de simulation montrent l'efficacité de la stratégie prédictive pour résoudre le problème de contraintes 2D.



---

# Conclusion générale

Nous avons essayé dans ce mémoire de montrer que le contrôle d'un robot en interaction avec son environnement physique est réalisé par un schéma de commande en boucle fermée directement sur des informations issues de capteurs visions. On parle alors d'*asservissement visuel* qui consiste à contrôler les mouvements d'un système robotique en intégrant les informations fournies par un capteur de vision dans un système de commande en boucle fermée.

Bien que les informations visuelles utilisées soient de bas niveau (points, droites, ellipses,...), il sera nécessaire d'utiliser des traitements spéciaux pour les extraire en temps réel (c'est-à-dire à la fréquence vidéo). Ces traitements spécifiques à chaque type de primitives, doivent permettre l'extraction des informations utiles à la commande. Dans la majorité des cas, le détecteur de Harris donne les meilleurs résultats.

L'asservissement visuel 2D permet la réalisation des tâches robotiques par comparaison d'information visuelles extraites de l'image courante, avec une image de référence. Bien que l'AV 2D est robuste aux erreurs de modélisation, plusieurs inconvénients peuvent être mentionnés tels que le problème de singularité de la matrice 'interaction qui conduit à la divergence du système. La prise en compte des contraintes est un problème délicat en asservissement visuel 2 D.

Dans le but de résoudre ces problèmes, nous avons présenté une stratégie de commande prédictive pour effectuer des tâches d'asservissement visuel 2D. La technique utilisée se base sur une commande prédictive non linéaire. Elle présente l'avantage de tenir compte explicitement des contraintes, mécaniques et/ ou visuelles. Un modèle de procédé assure la prédiction de l'image et permet de s'affranchir de la détermination explicite de la matrice d'interaction.

On distingue deux modèles pour la prédiction des primitives visuelles dans le plan image : un modèle global non linéaire composé du robot est d'une caméra et un modèle local basé sur la matrice d'interaction.

Pour les simulations nous avons choisi le modèle local qui permet la prise en compte des contraintes de visibilité. Ce choix est fait pour éviter la complexité du modèle global non linéaire. Les résultats de simulation, qui permettent de comparé entre les résultats obtenus à partir de l'asservissement visuel 2D en appliquant une loi proportionnelle et celle en appliquant une loi prédictive. Cette dernière présente des résultats satisfaisantes et efficace par rapport à la loi proportionnelle surtout au niveau de contraintes de visibilité telle que quelque soit la situation initiale de la caméra, la loi de commande converge en assurant que les primitives visuelles restent dans l'image pendant le mouvement de la caméra.

En perspective, nous souhaiterons réaliser ce travail sur le plan expérimental sur un bras manipulateur avec une caméra vidéo montée sur l'effecteur de ce dernier, un micro ordinateur, un contrôleur de vitesses. La communication entre le PC et le contrôleur est fait en utilisant Ethernet network. Les images sont acquises en utilisant la caméra vidéo FO124TC qui est équipée d'une liaison série 1394a.

# Bibliographie

- [1] HUTCHINSON.S; HAGAR.G.D; CORKE.P. *A tutorial on visual servo control*. IEEE Trans Robotics and Automation, vol 12(5), pp:651-671, 1996.
- [2] CHAUMETTE.F. *Asservissement visuel*. In W. Khalil, Ed., La commande des robots manipulateurs, Traité IC2, chapitre 3, Hermès, pp:105–150, 2002.
- [3] CHAUMEETE.F. *Avancées récentes en asservissement visuel*. In Journées Nationales de la Recherche en Robotique, JNRR'03, Clermont-Ferrand, pp:103–108, 2003.
- [4] BRUNO.S; LORENZO.S; LUIGI.V; GIUSEPPE.O. *Robotics: Modelling, Planning and Control*. Springer-Verlag, London, 2009.
- [5] DOMBRE.E. *Analyse et modélisation des robots manipulateurs*. Lavoisier, 2001.
- [6] HILL.J; PARK.W.T. *Real time control of a robot with a mobile camera*. 9<sup>th</sup> Int. Symp on Industrial Robots, pp: 233–246, 1979.
- [7] COUTARD.L. *Appontage automatique d'avions par asservissement visuel*. Thèse présentée pour obtenir le grade de Docteur de UNIVERSITÉ DE RENNES 1 sous le sceau de l'Université Européenne de Bretagne, 2012.
- [8] COURBON.J; MEZOUAR.Y; GENARD.N; MARTINET.P. *Vision-based navigation of unmanned aerial vehicles*. Elsevier Control Engineering Practice, vol 18, pp:789–799, 2010.
- [9] KRUPA.A. *Contributions à l'asservissement visuel échographique*. Thèse d'habilitation à diriger des recherches, l'université de Rennes 1.2012.
- [10] RIVES.P; BORRELLY.J.J. *Visual servoing techniques applied to an underwater vehicle*. IEEE Int. Conf. on Robotics and Automation, ICRA'97, Albuquerque, New Mexico, USA, pp: 1851–1856, 1997.
- [11] CORKE.P. *Robotics, Vision and Control: fundamental algorithms in Matlab*. Springer-Verlag, 2011.
- [12] DAVIES.E.R. *Computer and machine vision: Theory, Algorithms, Practicalities*. Elsevie, 2012.
- [13] CHAUMETTE.F. *La relation vision-commande: théorie et application des tâches robotiques*. Thèse de doctorat, Université de Rennes I, Institut de Formation Supérieur en Informatique et Communication, 1990.
- [14] HADJ-ADBLKADER.H; MEZOUAR.Y; MARTINET.P; CHAUMETTE.F. *Catadioptric visual Servoing from 3-D straight lines*. IEEE Transactions on Robotics, vol.24, pp: 652–665, 2008.

- [15] CHAUMETTE.F. *Image moments: a general and useful set of features for visual servoing*. IEEE Trans. on Robotics, vol.20 (4), 2004.
- [16] GINHOUX.R; GANGLOFF.J; DE MATHELIN.M; SOLER. L; SANCHEZ.A. MARESCAUX.J. *Active filtering of physiological motion in robotized surgery using predictive control*. IEEE Trans Robotics and Automation, vol.21, pp: 67-79, 2005.
- [17] CHAUMETTE.F; HUTCHINSON.S. *Visual servo control, part I: Basic approaches*. IEEE Robotics and Automation Magazine, vol.13, pp: 82-90, 2006.
- [18] CHAUMETTE.F; HUTCHINSON.S. *Visual servo control, part II: Advanced approaches*. IEEE Robotics and Automation Magazine, vol.14, pp: 109-118, 2007.
- [19] CHAUMETTE.F. *Potential problems of stability and convergence in image-based and position-based visual servoing*. The Confluence of Vision and Control, Lecture Note in Control and Informations Systems, Springer-Verlag, vol 237, pp: 66-78, 1998.
- [20] MALIS.E; CHAUMETTE.F; S.BOUDET.S. *2-1/2D visual servoing*. IEEE Transactions on Robotics and Automation, vol.15, pp: 238-250, 1999.
- [21] MARAVEC.H. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Tech Report CMU-RI-TR-3 Carnegie-Melon University, Robotics Institute, 1980.
- [22] SHI.J; TOMASIC. *Good features to track*. IEEE, 9<sup>th</sup> Conference on Computer Vision and Pattern Recognition, pp: 593-600, 1994.
- [23] HARRIS.C; STEPHENS.M. *A combined corner and edge detection*. Proceedings of the 4<sup>th</sup> Alvey Vision Conference, pp: 147-151, 1988.
- [24] YI.M; STEFANO.S; JANA.K; S.SHANKAR.S. *An invitation to 3D vision from images to geometric models*. Springer, Verlag, New York, vol. 24, 2004.
- [25] ESPIAU.B; CHAUMETTE.F; RIVES.P. *A new approach to visual servoing in robotics*. IEEE Transactions on Robotics and Automation, vol. 3, pp: 313-326. 1992.
- [26] HASHIMOTO.K; NORITSUGU.T. *Performance and Sensitivity in Visual Servoing*. Proceedings of IEEE, International Conference on Robotics & Automation, Leuven, Belgium, pp 2321-2326, 1998.
- [27] SAMSON.C; BORGNE.M.Le; ESPIAU.B. *Robot control: The task function approach*. Oxford University Press, England: Clarendon, 1992.
- [28] LAPRESTE.J.T; JURIE.F; DHOME.M ; CHAUMETTE.F. *Nouvelle approche pour le calcul du jacobien inverse en asservissement visuel 2D*. RFIA 2004, Toulouse, France, 2004.
- [29] MAHONY.R; CORKE.P; CHAUMETTE.F. *Choice of image features for depth-axis control in image-based visual servo control*. In: IEEE/RSJ Int. Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002.

- [30] CHESI.G. *Visual servoing path planning via homogeneous forms and LMI optimizations*. IEEE Transaction Robotics and Automation, vol. 25, pp: 281–291, 2009.
- [31] NASISI.O; CARELLI.R. *Adaptive servo visual robot control*. Robotics and Autonomous Systems, vol. 43, pp: 51–78, 2003.
- [32] ALLIBERT.G; COURTIAL.E; TOURE.Y. *Real-time visual predictive controller for image-based trajectory tracking of mobile robot*. In: 17<sup>th</sup> IFAC World Congress, Seoul, Korea, 2008.
- [33] QIN S.J; BADGWELL T.A. *A survey of industrial model predictive control technology*. Control Eng. Prac, vol.11 (7):733-764, 2003.
- [34] CAMACHO.E.F; BORDONS.C. *Model predictive control in process industry*. Springer, London, U.K. 1995.
- [35] CAMACHO.E. F; BORDONS.C. *Model Predictive Control*. Springer Berlin, 1999.
- [36] LARS. G; JURGEN. P. *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer-Verlag London Limited, 2011.
- [37] KOUVARITAKIS.B; CANNON.M. *Nonlinear predictive control: theory and practice*. The Institution of Engineering and Control, 2008.
- [38] GRUNE.L; PANNEK.J. *Nonlinear Model Predictive Control: Theory and Algorithms*. Communication and Control Engineering, Springer,2011.
- [39] FADLI.I. *Commande prédictive non linéaire d'un lit mobile simulé*. Thèse de doctorat, Institut National Polytechnique de Grenoble, 2006.
- [40] DAEWON L; HYON L; JIN KIM.H. *Obstacle Avoidance Using Image-based Visual Servoing Integrated with Nonlinear Model Predictive Control UAV-Image* 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC) Orlando, FL, USA, 2011.
- [41] ALLIBERT.G; COURTIAL.E; TOURE.Y. *Visual predictive control for manipulators with catadioptric camera*. In: IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 2008.
- [42] GANGLOFF.J; MATHELIN.F. *High speed visual servoing of a 6 DOF manipulator using MIMO predictive control*. Proceedings of the 2000 IEEE International Conference on Robotics & Automation San Francisco, CA, pp: 3751-3756, April 2000.
- [43] SAUVEE.M; POIGNET.P; DOMBRE.E; COURTIAL.E. *Image based visual servoing through nonlinear model predictive control*. Proceedings of the 45<sup>th</sup> IEEE Conference on Decision and Control, Manchester Grand Hyatt Hotel, San Diego, CA, USA, pp: 1776-1781, 2006.

- [44] FERREIRA.P; PINTO.J. *Visual based predictive control for a six degree of freedom robot*. In Proc of IEEE Conference on Emerging Technologies and Factory Automation, Prague, pp: 846-853, 2006.
- [45] ALLIBERT.G; COURTIAL.E; TOURE.Y. *Visual predictive control*. In: IFAC Workshop on Nonlinear Predictive Control for Fast System, Grenoble, France, 2006.
- [46] ALLIBERT.G; COURTIAL.E; TOURE.Y. *Aflat model predictive controller for trajectory tracking in image based visual servoing*. Proceedings of the 7<sup>th</sup> IFAC Symposium on Non Linear Control Systems, Pretoria, South Africa, 2007.
- [47] LAZAR.C; BURLACU.A. *Predictive control strategy for image based visual servoing of robot manipulators*. 9<sup>th</sup> WSEAS International Conference on Automation and Information (ICAI'08'), Bucharest, Romania, pp: 91-96, 2008.
- [48] LAZAR.C; BURLACU.A. *Visual servoing of robot manipulators using model based predictive control*. In Proc of 7<sup>th</sup> IEEE international Conference on Industrial Informatics, Cardiff, pp: 690-695, 2009.
- [49] ALLIBERT.G; COURTIAL.E. *What can prediction bring to image-based visual servoing?* In Proc. IEEE/RSJ International Conference Intell. Robots Syst, St.Louis, MO, pp. 5210–5215, 2009.
- [50] ALLIBERT.G, COURTIAL.E; CHAUMETTE.F. *Predictive control for constrained image based visual servoing*. IEEE Transaction on Robotics, vol. 26(5), pp: 933-939, 2010.
- [51] YING.W; HAOXIANG.L; CLARENCE W.DE.S. *An autonomos Mobile Grasping System using Visual Servoingf and Nonlinear Model Predictive Control*. 8<sup>th</sup> IEEE International Conference on Control and Automation, Xiamen, China, 2010.
- [52] LAZAR.C; BURLACU.A; COPOT.C. *Predictive Control Architecture for Visual Servoing of Robot Manipulator*. Preprints of the 18<sup>th</sup> IFAC World Congress Milano (Italy), 2011.
- [53] COPOT.C; LAZAR.C; BURLACU.A. *Predictive control of nonlinear visual servoing systems using image moments*. IET Control Theory Appl, vol.6, Iss.10, pp: 1486-1496, 2012.
- [54] BURLACU.A; COPOT.C; CERVERA.E; LAZAR.C. *Predictive control architecture for real-time image moments based servoing of robot manipulators*. Springer Science+Business Media New York 2013.
- [55] MORARI.M; ZAFIRIOU.E. *Robust Control*. Dunod, 1983.
- [56] CHONG E; ZAK.S.H, *An Introduction to Optimization*, 2<sup>nd</sup> edition. New York: Wiley, 2001.
- [57] NOCEDAL.J; WRIGHT.S.J. *Numerical Optimization*, 2<sup>nd</sup> Edition. Springer Series in Operations Research, Springer Verlag, 2006.

- [58] KRAGIC.D; CRISTENSEN.H. *Cue integration for visual servoing*. IEEE Trans on Robotics and Automation, vol.17, pp: 19-26, 2001.
- [59] FUJIMOTO.H. *Visual servoing of 6 dof manipulator by multirate control with depth identification*, Proc. of 42<sup>nd</sup> IEEE Conference on Decision and Control, Maui, Hawaii, pp: 5408- 5413, 2003.
- [60] CERVERA.E. *Visual Servoing Toolbox*. Jaume I University, Castello, 2003, <http://sourceforge.net/projects/vstooblox/>.