

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Automatique
Laboratoire de Commande des Processus



Thèse

Présentée en vue de l'obtention du titre de
Docteur en Sciences

en Automatique
par

Noureddine OUADAH

Magistère en Robotique de l'Ecole Militaire Polytechnique

Intitulé :

LE CONTRÔLE VISUEL APPLIQUÉ DANS LA ROBOTIQUE MOBILE

Soutenue publiquement le 27 / 10 / 2011 à 10h à la Nouvelle bibliothèque devant le jury
composé de :

M. S. BOUCHERIT	Professeur à l'ENP	Président
M. HAMERLAIN	Directeur de recherche au CDTA	Rapporteur
F. BOUDJEMA	Professeur à l'ENP	Rapporteur
M. TADJINE	Professeur à l'ENP	Examineur
M. DEVY	Directeur de recherche au LAAS-CNRS, Toulouse, France	Examineur
M. ATTARI	Professeur à l'USTHB	Examineur
N. ACHOUR	Maître de conférences « A » à l'USTHB	Examineur
V. CADENAT	Maître de conférences à l'UPS, Toulouse, France	Invitée

Année universitaire : 2011/2012

Thèse préparée au sein du Laboratoire de Commande des Processus de l'ENP

10, Avenue Hassen Badi, BP 182, 16200 El-Harrach, Alger



A la mémoire de ma chère Maman

A mon Epouse et mon Papa

A mes Frères et Sœurs

A ma belle famille, Fella et Menad

A « Kram » & « Youès »

REMERCIEMENTS

Je voudrais tout d'abord remercier les membres du jury pour avoir accepté de juger ce travail : Mr Mohamed Said BOUCHERIT, Professeur à l'ENP d'Alger qui m'a fait l'honneur de présider ce jury. Je remercie Mme Noura ACHOUR, Maître de conférences « A » à l'USTHB d'Alger et Messieurs Mohamed TADJINE, Professeur à l'ENP d'Alger, Michel DEVY, Directeur de recherche au LAAS-CNRS de Toulouse et Mokhtar ATTARI, Professeur à l'USTHB d'Alger qui ont eu la gentillesse d'accepter de juger cette thèse. Je tiens à exprimer encore une fois mes remerciements à Mr DEVY pour m'avoir accueilli chaleureusement au LAAS, et aider à m'intégrer au sein du groupe RAP.

Je voudrais adresser mes plus vifs remerciements à Mlle Viviane CADENAT, Maître de conférences à l'UPS de Toulouse pour ses conseils, son soutien, sa rigueur scientifique, sa grande patience, et surtout d'avoir accepté de participer à ce jury en qualité d'invitée.

Mes remerciements vont également à Mr Mustapha HAMERLAIN, Directeur de recherche et de la division « *Productique et Robotique* » du CDTA, ainsi qu'à Mr Farès BOUDJEMA, Professeur à l'ENP d'Alger pour avoir accepté de diriger cette thèse.

Je remercie également tous les membres du pôle RIA du LAAS-CNRS pour leur gentillesse, leur disponibilité et leur contribution de près ou de loin à cette thèse, et tout spécialement Thierry GERMA avec qui j'ai mené les expérimentations, et Patrick DANÈS pour ses précieux conseils et orientations.

Un grand merci à mes collègues de la division « *Productique et Robotique* » pour leurs aides et encouragements (Mr CHETTOUH, Mlle AZOUAOUI et Mlle OURARI), ainsi qu'à tous les autres sans exception.

Je ne peux clore cette partie sans remercier ma famille, et plus particulièrement mon épouse pour m'avoir supporté et soutenu avec beaucoup de patience et de tendresse.

العنوان: التحكم المرئي لروبوت متحرك

ملخص:

المشكلة الرئيسية التي تم تناولها في هذا العمل هي دراسة وتطبيق تحكم مرئي على روبوت متحرك مزود بكاميرا. المهام المنقاة هي التوقع بالنسبة لهدف معروف، وتتبع هدف متحرك بتحكم متعدد الحواس. في هذا الإطار، قمنا أولاً ببرمجة المهمة الأولى من خلال تحكم مرئي كلاسيكي مبني على أساس مبدأ "وظيفة المهمة". بعد ذلك أدرجنا تحكم جديد باستعمال نظام انزلاقي قادر على قيادة الروبوت بشكل صحيح حتى في حالة تحرك الهدف، وقد تم اختبار ذلك بالمحاكاة قبل تجريبيها ميدانياً على روبوت حقيقي متحرك مخصص للاستعمال الداخلي. أخيراً، اقترحنا تقنية جديدة تأخذ بعين الاعتبار معلومات من حواس مختلفة، و تم التحقق من فعاليتها تجريبياً في إطار وظيفة بدء وضمان استمرارية التفاعل بين الروبوت ومستخدم معروف، مجهز بـ tag RFID النتائج المتحصل عليها أثبتت متانة أحسن للتقنية بفضل الجمع بين قوانين تحكم متعددة (كلاسيكي و انزلاقي) و كفاءة عالية نتيجة استعمال معلومات حسية مختلفة (الصورة والكشف بتكنولوجيا RFID).
المفاتيح: التحكم المرئي، التعرف على الأوجه، التحكم بنظام الانزلاق، الكشف بتكنولوجيا RFID.

Résumé :

Le problème principal traité dans ce travail est l'étude et la mise en œuvre d'un asservissement visuel sur un robot mobile. Les missions choisies sont le positionnement référencé vision et la poursuite multimodale de cible mobile. La première tâche de positionnement a été d'abord réalisée par l'implémentation d'un asservissement visuel classique, basé sur le principe de la « fonction de tâche ». Par la suite, nous avons introduit les modes glissants afin d'implémenter un asservissement visuel robuste capable de positionner le robot correctement même pour le cas d'un utilisateur mobile. Ces deux techniques ont été testées en simulation, puis implémentées expérimentalement sur un robot mobile d'intérieur. Enfin, une nouvelle approche hybride et multimodale a été proposée et validée expérimentalement dans le cadre d'une tâche d'initiation et de maintien de l'interaction entre le robot et un utilisateur connu, muni d'un tag RFID. Les résultats obtenus ont montré une meilleure robustesse due à la combinaison de lois de commande (classique et régimes glissants), et une amélioration de l'efficacité grâce à l'utilisation d'informations sensorielles différentes (image et détection RFID).

Mots clés : Asservissement visuel, Détection RFID, Modes glissants, Reconnaissance de visages.

Title: Visual Control Applied in Mobile Robotics

Abstract:

This work mainly addresses the problem of studying and implementing a visual servoing on a mobile robot. Chosen missions are image based positioning, and multimodal mobile target tracking. The positioning task has been performed first by implementing a classical visual servoing, based on the "task function" theory. Subsequently, we introduced the sliding mode technique to implement a robust visual servoing able to control the robot efficiently even in the case of a mobile user. Both control techniques have been tested in simulation, then implemented in real time on an indoor mobile robot. Finally, a new hybrid and multimodal approach has been proposed and experimentally validated in the context of initiating and maintaining an interaction task between the robot and a known user, equipped with an RFID tag. The obtained results have shown a good robustness generated by control laws combination (classical and sliding modes), and a more efficient control strategy using several sensory information (image and RFID detection).

Keywords: Faces recognition, RFID detection, Sliding modes control, Visual servoing.

TABLE DES MATIERES

	Page
Chapitre 1 INTRODUCTION GENERALE	1
1. INTRODUCTION	1
2. PROBLEMATIQUE	2
3. LE PROJET « COMMROB »	4
4. ORGANISATION DU MEMOIRE	4
5. VALORISATIONS	5
6. CONCLUSION	6
Chapitre 2 CONTEXTE DE L'ETUDE & ETAT DE L'ART	7
1. INTRODUCTION	7
2. COMMANDE REFERENCEE CAPTEURS	8
2.1 Formalisme des fonctions de tâche	8
2.1.1 Notion de ρ -admissibilité	9
2.1.2 Formalisme des tâches redondantes	9
2.2 Commande référencée vision et Asservissement Visuel	11
2.2.1 Asservissement Visuel « 3D »	12
2.2.2 Asservissement Visuel « 2D »	14
2.2.3 Asservissement Visuel « 2D ½ »	18
2.2.4 Asservissement Visuel « d2D/dt »	19
2.2.5 L'Asservissement Visuel « rapide »	21
2.3 Asservissement Visuel et Robustesse	22
2.4 Poursuite référencée capteurs de cible mobile	23
3. CONCLUSION	24
Chapitre 3 ASSERVISSEMENT VISUEL 2D D'UN ROBOT MOBILE NON- HOLONOME	26
1. INTRODUCTION	26
2. MODELISATION DU SYSTEME ROBOTIQUE	27
2.1 Système de vision	27
2.1.1 Modèle de la caméra	27

2.1.2	Modèle de la PTU	32
2.1.3	Détection et reconnaissance visuelle de l'utilisateur	33
A.	Les détecteurs	33
B.	La reconnaissance des visages	34
C.	Les suivis	34
2.2	Modèle du système robotique	36
2.3	Interaction Caméra/Environnement	39
2.3.1	Matrice d'interaction	40
2.3.2	Cas du point	40
2.4	Le système RFID	42
2.4.1	Description du capteur	44
2.4.2	Evaluation de la faisabilité	44
2.5	Environnement logiciel du robot	45
2.5.1	Présentation de <i>Genom</i>	45
2.5.2	Contexte d'utilisation	48
2.5.3	Architecture logicielle embarquée	48
3.	IMPLEMENTATION D'UN ASSERVISSEMENT VISUEL 2D	50
3.1	Synthèse du contrôleur classique	51
3.2	Simulations	54
3.3	Implémentation expérimentale	58
4.	CONCLUSION	60

Chapitre 4 SYNTHÈSE ET IMPLEMENTATION DE LOIS DE COMMANDE
ROBUSTES & MULTIMODALES

1.	INTRODUCTION	61
2.	ASSERVISSEMENT VISUEL PAR LES MODES GLISSANTS	62
2.1	Principes de la commande par modes glissants	62
2.2	Bouclage par les modes glissants classiques	63
2.2.1	Formalise de la commande à structure variable	63
2.2.2	Approche de <i>Fillipov</i>	64
2.2.3	Approche d' <i>Utkin</i>	66
2.2.4	Approche de <i>Sira-Ramirez</i>	68
A.	Le régime glissant	68
B.	Condition d'existence et unicité du contrôle équivalent	69

C. Rejets des perturbations	71
D. Approche du degré relatif	72
2.2.5 Composantes de la commande	74
2.3 Synthèse d'un Asservissement Visuel par Modes Glissants	77
2.4 Stabilité et convergence du contrôleur	80
2.5 Simulations	83
2.6 Implémentation expérimentale	85
3. COMMANDE CLASSIQUE MULTIMODALE APPLIQUEE AU GUIDAGE & A LA POURSUITE D'UN UTILISATEUR	87
3.1 Suivi visuel de l'utilisateur par PTU	87
3.1.1 Formulation du problème	87
3.1.2 Lois de commande	88
3.1.3 Résultats expérimentaux	89
3.2 Poursuite multimodale de l'utilisateur	90
3.2.1 Formulation du problème	90
3.2.2 Lois de commande	90
3.2.3 Résultats expérimentaux	93
4. COMMANDE ROBUSTE MULTIMODALE POUR LA POURSUITE D'UTILISATEUR	98
4.1 Asservissement Visuel Hybride	99
4.2 Asservissement Multimodal Hybride	100
4.3 Résultats expérimentaux	101
4.3.1 Cas d'un utilisateur visible	102
4.3.2 Cas d'un utilisateur visible par moment	104
5. CONCLUSION	108
 Chapitre 5 CONCLUSION GENERALE ET PERSPECTIVES	 109
 REFERENCES.....	 112

LISTE DES FIGURES

	Page
Chapitre 1	
Fig.1.1 Prototype de caddy intelligent « <i>Inbot</i> »	3
Fig.1.2 Le robot guide « <i>Rackham</i> »	3
Chapitre 2	
Fig.2.1 Espace de configuration d'un capteur	8
Fig.2.2 Asservissement visuel « 3D »	13
Fig.2.3 Asservissement visuel « 2D »	15
Fig.2.4 Trajectoire des points de la cible dans l'image	16
Fig.2.5 Asservissement visuel « $2D\frac{1}{2}$ »	18
Fig.2.6 Exemples de champs de vitesse	19
Fig.2.7 Asservissement visuel « $d2D/dt$ »	20
Chapitre 3	
Fig.3.1 Modèle géométrique de la caméra	26
Fig.3.2 Tourelle Pan/Tilt (PTU)	30
Fig.3.3 Angle de vue de la caméra	30
Fig.3.4 Détection et reconnaissance de visages	32
Fig.3.5 Séquence d'appel des composants d'ICU	33
Fig.3.6 Repères du système Robot-Platine-Caméra	34
Fig.3.7 Composants d'un système RFID (a) Un Tag, (b) Deux modèles d'antennes	40
Fig.3.8 Prototype de multiplexage RF pour 8 antennes	41
Fig.3.9 Champ de détection azimutal avec 8 antennes RFID	43
Fig.3.10 Architecture logicielle de l'application	47
Fig.3.11 Test 1 de simulation d'un asservissement visuel 2D : (a) Trajectoires des indices visuels, (b) Evolution de la fonction de tâche, (c) Trajectoire du robot	52
Fig.3.12 Variables de contrôle de l'asservissement visuel 2D pour le test 1	53
Fig.3.13 Test 2 de simulation d'un asservissement visuel 2D : (a) Trajectoires des indices visuels, (b) Evolution de la fonction de tâche, (c) Trajectoire du robot	54
Fig.3.14 Variables de contrôle de l'asservissement visuel 2D pour le test 2	55

Fig.3.15 Trajectoires des indices visuels en expérimentation	56
Fig.3.16 Test expérimental de l'asservissement visuel 2D classique (a) Evolution de la fonction de tâche, (b) Erreur de positionnement de la cible	57
Fig.3.17 Variables de contrôle de l'asservissement visuel 2D en expérimentation	57

Chapitre 4

Fig.4.1 Espace d'état et surface de glissement	64
Fig.4.2 Dynamique de glissement selon <i>Filippov</i>	65
Fig.4.3 Principe de la commande à structure variable	67
Fig.4.4 Phénomène du <i>Chattering</i>	76
Fig.4.5 Exemple de fonctions de commutation	76
Fig.4.6 Test de simulation de l'asservissement visuel 2D par modes glissant	83
Fig.4.7 Variables de contrôle de l'asservissement visuel 2D par modes glissant	84
Fig.4.8 Trajectoires des indices visuels en expérimentation	85
Fig.4.9 Test expérimental de l'asservissement visuel 2D par modes glissants : (a) Evolution de la fonction de tâche (b) Erreur de positionnement de la cible .	86
Fig.4.10 Variables de contrôle de l'asservissement visuel 2D en expérimentation	86
Fig.4.11 Erreurs de position du visage dans l'image	89
Fig.4.12 Vitesses des axes de la PTU	89
Fig.4.13 Manœuvres d'aide au recouvrement du signal visuel	92
Fig.4.14 Exemple de tâche de poursuite multimodale d'utilisateur	93
Fig.4.15 Séquences d'exécution de la poursuite d'utilisateur mobile par une commande multimodale	94
Fig.4.16 Signaux issus de la tâche de poursuite multimodale	95
Fig.4.17 Positions du visage de l'utilisateur en poursuite multimodale	96
Fig.4.18 Principe de la poursuite d'utilisateur par l'asservissement multimodal hybride	101
Fig.4.19 Test expérimental de l'asservissement visuel hybride (a) Trajectoire du robot, (b) Trajectoire des indices visuels dans l'image	102
Fig.4.20 Evolution des erreurs en asservissement visuel hybride	103
Fig.4.21 Signaux de commandes en asservissement visuel hybride	104
Fig.4.22 Test expérimental de l'asservissement multimodal hybride (a) Trajectoire du robot, (b) Trajectoire des indices visuels dans l'image	105
Fig.4.23 Evolution des erreurs en asservissement multimodal hybride	105
Fig.4.24 Signaux de commandes en asservissement multimodal hybride	106

LISTE DES TABLEAUX

	Page
Chapitre 3	
Tableau 2.1 Résultats du calibrage de la caméra	31
Chapitre 4	
Tableau 4.1 Ratio de contact visuel considérant 1 à 4 personnes	97

NOMENCLATURE

C	Matrice de combinaison	u	Coordonnée horizontale dans l'image
C_x	Abscisse de la caméra / PTU(m)	u_i	Coordonnée u du centre de l'image
C_y	Ordonnée de la caméra / PTU (m)	v	Coordonnée verticale dans l'image
D_x	Distance robot-PTU (m)	v_i	Coordonnée v du centre de l'image
e	Fonction de tâche	v_r	Vitesse linéaire du robot (m/s)
E	Erreur de position du visage dans l'image (pix)	x	Abscisse du robot (m)
E_p	Erreur de positionnement de la cible (pix)	y	ordonnée du robot (m)
E_u	Erreur horizontale du visage dans l'image (pix)	z	Distance caméra-cible (m)
E_v	Erreur verticale du visage dans l'image (pix)	φ_p	Angle Pan de la PTU (°)
f	Distance focale de la caméra (mm)	φ_t	Angle Tilt de la PTU (°)
H	Homographie de la cible	σ	Surface de glissement
J	Jacobienne du robot	λ	Gains de l'asservissement visuel
J_r	Jacobienne réduite du robot	Ω	Vitesse de rotation (rad/s)
K_p	Gain proportionnel du PID	μ_{rcv}	Moyenne du ratio de contact visuel
K_i	Gain intégral du PID	σ_{rcv}	Ecart type du ratio de contact visuel
K_d	Gain dérivatif du PID	τ_r	Torseur cinématique réduit
l	Abscisse curviligne du robot (m)	μ	Flag de Switch de l'approche hybride
L	Matrice d'interaction	τ	Torseur cinématique de la caméra
L_r	Matrice d'interaction réduite	θ	Orientation de la base mobile (rad)
q	Vecteur configuration du robot	μ_V	Indicateur cible détectée par ICU
\dot{q}	Vecteur commande du robot	μ_R	Indicateur cible détectée par RFID
R_0	Repère lié à la scène	ω_r	Vitesse angulaire du robot (rad/s)
R_M	Repère lié à la base mobile	ω_p	Vitesse angulaire Pan de la PTU (rad/s)
R_P	Repère lié à la PTU	ω_t	Vitesse angulaire Tilt de la PTU (rad/s)
R_C	Repère lié à la caméra		
s	Indices visuels		
t	Le temps (s)		

CHAPITRE 1

INTRODUCTION GÉNÉRALE

1. INTRODUCTION

Depuis son apparition, l'homme a toujours rêvé de pouvoir faire exécuter par des acteurs autres que lui-même des tâches ou activités qu'il considère comme aliénantes, dangereuses ou simplement ennuyeuses [45]. Ainsi, l'émergence de la robotique au cours des années 60 a su apporter de nouvelles réponses à cette attente, ce qui s'est traduit par une grande volonté des chercheurs à reproduire les capacités humaines de perception et d'action dans les systèmes robotisés. Quelques années plus tard, ces recherches ont conduit à l'intégration de données issues de capteurs extéroceptifs dans la boucle de commande des robots, comme celles fournies par une caméra. En pratique, cela a permis un élargissement important du domaine d'application de la robotique, et une amélioration considérable de la précision obtenue. Dans ce chapitre, nous introduisons le contexte général des travaux menés dans le cadre de cette étude, et qui consiste à implémenter une commande référencée capteurs sur un robot mobile, en vue d'initier et de maintenir une tâche d'interaction Homme/Robot.

2. PROBLEMATIQUE

Les travaux de recherche menés dans le cadre de cette étude s'intéressent au problème de définition de schémas de commande en boucle fermée, en utilisant des informations sensorielles de diverses natures. L'aspect réactif vis-à-vis des événements extérieurs perturbant le déroulement de la tâche est au cœur de nos préoccupations. En effet, l'exploitation des seules données visuelles, comme nous le verrons dans la suite de ce manuscrit, ne permet pas de définir des tâches robotiques élaborées. De plus, l'asservissement visuel classique est basé sur la seule régulation à zéro, et de manière basique, de l'erreur entre ce que le robot voit réellement et ce qu'il doit voir lorsque la tâche est parfaitement réalisée [45]. Cette approche s'avère alors incomplète et peu adaptée pour être exploitée dans une vraie tâche robotique (navigation autonome, interaction Homme/Robot, etc.). En effet, elle est incapable de traiter les problèmes liés à la perte du signal visuel (occultation, sortie du champ de vision, etc.). Aussi, elle présente des sensibilités importantes vis-à-vis des erreurs de modélisation, ou des perturbations (mouvements de la cible, erreurs d'estimation des primitives, etc.). C'est précisément ces problèmes que nous avons voulu étudier dans le cadre de cette thèse.

Plusieurs travaux de recherche ont déjà abordé une partie de ces problèmes et ont proposé des solutions. Dans [9], l'approche considérée consiste à définir des lois de commande référencées multi-capteurs, c'est-à-dire, qu'elle exploite différents types de mesures sensorielles permettant ainsi de caractériser d'une part l'objectif de la mission qui sera définie à partir d'informations visuelles, et d'autre part, d'éviter les éventuelles collisions. Toutefois cette approche considère uniquement des obstacles dont la hauteur est inférieure à celle de la caméra, et qui, par conséquent, ne pouvaient pas perturber les informations visuelles. Par la suite, *D. Folio* [45] a apporté des contributions en considérant des obstacles pouvant occulter (ou non) le motif visuel. Il a alors défini un ensemble de lois de commande référencées multi-capteurs permettant à un robot mobile de réaliser une tâche référencée vision dans un environnement encombré d'obstacles susceptibles d'occulter le motif visuel.

Nous nous sommes alors appuyés dans le cadre de la réalisation de nos travaux sur ces techniques d'asservissement référencés capteurs, en particulier sur l'asservissement visuel 2D, et en se fixant comme objectif l'amélioration de leurs performances en termes de contrôle. Nous proposons aussi des solutions en vue d'étendre leur champ d'application, en rendant ces techniques capables de gérer le phénomène de la perte du signal visuel. Il s'agit alors d'une

part de synthétiser des asservissements visuels avec une meilleure robustesse et précision, en employant de nouvelles techniques de commande. D'autre part, nous avons introduit l'aspect perception multimodale, avec l'exploitation de données perceptuelles de diverses natures (visuelle et RFID) afin d'employer de telles techniques dans la réalisation de missions robotiques complexes.

Plus précisément, cette étude se penche sur le cas d'une tâche d'interaction Homme/Robot dans un espace intérieur encombré. Nous nous intéressons donc à des missions de type « guidage » et « poursuite d'utilisateur mobile », comme spécifié dans le cadre du projet Européen *CommRob* [25]. Dans ce projet, l'objectif est l'étude et la réalisation d'un prototype de caddy intelligent et interactif (voir figure 1.1), comme détaillé ci-après. De ce fait, les travaux relatifs à notre étude sont réalisés en forte collaboration avec d'autres chercheurs membres du groupe *RAP* du *LAAS-CNRS*, dont la partie « *Perception Multimodale de l'Homme* », qui est traitée dans le cadre de la thèse de doctorat de *T. Germa* [53]. Enfin, une phase importante de notre travail consiste en l'implémentation expérimentale des techniques développées sur la plateforme robotique *Rackham* (voir figure 1.2) pour validation, avant leur portage sur le robot *Inbot*.



Fig. 1.1 : Prototype de caddy intelligent « *Inbot* »



Fig. 1.2 : Le robot guide « *Rackham* »

3. LE PROJET *COMMROB*

Le projet européen STREP *CommRob* (COMMunication with and among ROBots) [25] est un projet financé par la division FP6 de la Commission Européenne. Ce projet comporte cinq partenaires académiques ou industriels *i.e.* *FZI* (Karlsruhe, Allemagne), *KTH* (Stockholm, Suède), *LAAS-CNRS* (Toulouse, France), *TUW* (Vienne, Autriche) et *Zenon* (Athènes, Grèce). Ce projet vise à développer un robot assistant capable d'évoluer en environnement humain très encombré. Il a pour but de mettre en avant les avancées dans le domaine de la communication de haut niveau avec/entre robots. Au delà de la communication par la parole, la communication multimodale considère différents percepts (déplacements, gestes, etc.) afin de définir une interaction Homme/Robot plus riche [53]. Différents scénarios d'utilisation sont définis afin de mettre en avant les différentes fonctionnalités développées au sein du projet, ainsi que leur intégration. Nous listons ci-après les scénarios relatifs à notre travail :

- **Scénario S3** : Ce scénario est défini tel que l'utilisateur agit comme agent maître. Les déplacements du robot se font grâce à la perception multimodale. Le robot accompagne alors l'utilisateur en respectant une certaine distance de sécurité, tout en détectant et évitant les obstacles. Les fonctionnalités mises en œuvre dans ce scénario sont l'identification et le suivi multimodal (vision + RFID).
- **Scénario S4** : Ce scénario est défini tel que le robot a l'initiative des mouvements. Il doit alors s'assurer de la présence de l'utilisateur et rester à une certaine distance.

4. ORGANISATION DU MEMOIRE

Le présent mémoire est organisé en quatre chapitres structurés de la manière suivante : Le premier chapitre est une introduction générale aux travaux de recherche menés dans cette étude. Dans le chapitre 2, nous présentons le contexte de l'étude en rappelant en premier les principaux formalismes sur lesquels sont basés ces travaux, pour en suite exposer un état de l'art de l'asservissement visuel et de la commande référencée capteurs, en général. Le chapitre 3 regroupe deux parties principales : la première est dédiée à la modélisation de la plateforme robotique considérée dans cette étude, alors que la seconde traite le problème de la synthèse et l'implémentation d'un asservissement visuel 2D classique sur l'homme. Puis, des évaluations en simulation et en expérimentation sont détaillées afin de valider l'approche proposée.

L'approche classique s'avérant limitée dans certains cas (cible mobile, occultée ou sortie du champ de vision, etc.), nous avons cherché à améliorer ces résultats en développant, dans le chapitre 4, des techniques plus robustes, capables de recouvrir le signal visuel lorsque celui-ci n'est plus disponible. Ainsi dans une première partie, nous avons introduit une nouvelle loi de commande visuelle robuste basée sur les régimes glissants, puis avons proposé une approche hybride combinant à la fois les deux lois de commande classique et robuste. Dans une seconde partie, nous avons proposé deux approches multimodales utilisant les données fournies par le système de détection RFID au sein des boucles de commande. L'ensemble de ces approches ont été validés, et les résultats d'expérimentation sont présentés. Finalement, nous concluons sur les travaux réalisés et présentons les perspectives qui peuvent être envisagées.

5. VALORISATION

Les travaux effectués dans le cadre de cette thèse ont été valorisés à travers les publications suivantes :

5.1 Publications internationales

- Ouadah, N., Ourak, L., Boudjema, F. Car-like Mobile Robot Oriented Positioning by Fuzzy Controllers. 2008, *The International Journal of Advanced Robotic Systems*, Vol. 5, N° 3, pp. 249-256.
- Ouadah, N., Cadenat, V., Lerasle, F., Hamerlain, M., Germa, T., Boudjema, F. Multi-Sensor-Based Control Strategy for Initiating and Maintaining Interaction between a Robot and a Human. 2011, *Advanced Robotics*, vol. 25 n° 09-10, pp 1249-1270.

5.2 Communications internationales

- Ouadah, N., Ourak, L., Boudjema, F., Hamerlain, M. Implementation of an oriented positioning on a car-like mobile robot by fuzzy control. *32th Annual Conference of the IEEE Industrial Electronics Society (IECON'06)*, Paris, France, Novembre 06-10, 2006.
- Ouadah, N., Boudjema, F., Hamerlain, M. Target Tracking by Non-Holonomic Mobile Robot using Visual Control. *The 6th IFAC Symposium on*

Intelligent Autonomous Vehicles (IAV'07), Toulouse, France, Septembre 03-05, 2007.

- Ouadah, N., Cadenat, V., Boudjema, F., Hamerlain, M. Image Based Robust Visual Servoing on Human Face to Improve Human/Robot Interaction. *European conference on mobile robots (ECMR'09)*, Mlini/Dubrovnic, Croatia, Septembre 23-25, 2009.

6. CONCLUSION

Ce chapitre constitue une introduction générale à l'étude menée dans le cadre de cette thèse de doctorat. Nous avons exposé brièvement nos motivations, et cité des travaux récents sur la commande référencée capteurs de robots mobiles, puis exposé le cadre du travail (Projet *CommRob*) et la problématique traitée. Nous avons enfin présenté l'organisation du mémoire, et les valorisations résultantes de ce travail. Le prochain chapitre est dédié à l'état de l'art de la commande référencée capteurs, et plus particulièrement à l'asservissement visuel.

CHAPITRE 2

CONTEXTE DE L'ETUDE & ETAT DE L'ART

1. INTRODUCTION

Nous traitons dans notre étude le problème de la navigation d'un robot mobile non-holonome dans un environnement intérieur, par une commande référencée capteurs. Il s'agit plus précisément de réaliser des tâches de type poursuite et guidage d'utilisateur, en se basant sur les informations issues des capteurs embarqués sur le robot. L'identification de l'utilisateur est faite à partir des informations sensorielles.

Dans la première partie de ce chapitre, nous rappelons quelques notions de base concernant la commande référencée capteurs, et plus particulièrement le Contrôle (ou l'asservissement) visuel. Nous citons alors les différents types l'asservissement visuel, en présentant l'état de l'art de cette discipline.

Dans la deuxième partie, nous nous intéressons à l'application visée dans notre étude. Nous exposons quelques cas de systèmes robotiques existants, utilisant des d'informations sensorielles pour réaliser la poursuite de cible mobile.

2. COMMANDE REFERENCEE CAPTEURS

Nous présentons tout d'abord un rappel sur le cadre théorique de la commande référencée capteur développée durant les années 90. Ce cadre permet de définir une méthodologie complète de programmation de tâches robotiques allant de la spécification jusqu'à la synthèse des lois de commande. Cette méthodologie peut être appliquée à une grande variété de capteurs tels que les caméras, les capteurs de force ou les capteurs télémétriques [97], [38] et [93]. D'autres notions de base de la commande référencée capteur sont plus détaillées dans l'ouvrage de *Samson et al.* [97] apparu en 1991.

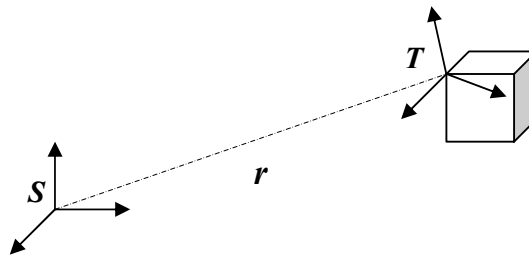


Fig. 2.1 : Espace de configuration d'un capteur

Soit un capteur rigidement monté sur le robot, représenté par un repère S . Il fournit un signal s lié à la perception d'un objet cible dans son environnement (voir figure 2.1). Un repère T est associé à l'objet cible. Nous posons, comme hypothèse de base, que le signal s ne dépend que de la configuration relative r (position et orientation) entre le repère du capteur et un repère de référence lié à l'environnement ($s : \mathbb{C} \rightarrow \mathbb{R}$, où \mathbb{C} est l'espace de configuration du repère associé au capteur). Cela signifie que si le capteur est statique par rapport à l'environnement, alors le signal s est stationnaire. Cette propriété est vérifiée pour une grande classe de capteurs tels que les capteurs de force, les capteurs visuels (primitives géométriques) ou télémétriques [104].

2.1 Formalisme des fonctions de tâche

Le formalisme des fonctions de tâche a été introduit par *Samson et al.* [97], mais son utilisation était limitée aux schémas de commande des bras manipulateurs. Il fut ensuite étendu au cas des robots mobiles non-holonomes par *Pissard* [91]. Ainsi, dans le cadre de ce formalisme, une tâche robotique peut être exprimée comme la régulation sur un certain horizon temporel $t \in \{0, T\}$ d'une fonction $e(q, t)$ de classe C^2 , appelée fonction de tâche (où q est le vecteur de configuration du robot).

Les fonctions de tâche peuvent être exprimées en termes de données proprioceptives ou extéroceptives selon l'application considérée et les capteurs disponibles sur le robot [45]. Nous présentons ci-après quelques exemples de fonctions classiques :

- $e(q,t) = q(t) - q^*(t)$ où $q^*(t)$ est une trajectoire désirée dans l'espace des configurations.
- $e(q,t) = r(q) - r^*(t)$ où $r^*(t)$ est une trajectoire désirée dans l'espace cartésien.
- $e(q,t) = s(q,t) - s^*(t)$ où $s^*(t)$ est une trajectoire désirée dans l'espace du capteur.

Cependant, *Samson et al.* [97] exigent dans leur travaux des conditions pour que le problème de régulation de $e(q, t)$ soit bien posé :

- La fonction de tâche doit posséder la propriété de l'unicité de la trajectoire $q_r(t)$ solution de l'équation : $e(q, t) = \theta$, $\forall t \in \{0, T\}$. Or, la fonction de tâche étant non-linéaire en q , la résolution de cette équation peut conduire à la définition de plusieurs trajectoires solutions distinctes. Pour éviter cela, une condition initiale q_0 est introduite, telle que : $q_r(0) = q_0$ vérifie $e(q_0, 0) = \theta$. Notons qu'une trajectoire vérifiant cette propriété est dite trajectoire idéale. Cependant, il se peut qu'aucune solution n'existe, excepté la condition initiale, ou au contraire, qu'il y en ait une infinité de solutions [45]. Dans ce dernier cas on parle alors de tâche redondante que nous présentons à la section « 2.1.2 ».
- D'une part, il est nécessaire que la régulation de la fonction de tâche $e(q, t)$ à zéro impose la convergence de la trajectoire du robot vers la trajectoire solution désirée, et d'autre part qu'une petite variation de $e(q, t)$ n'induisse pas une grande variation de la configuration q du robot.

Lorsque toutes les conditions requises sont satisfaites, la fonction de tâche est dite *admissible*, ce qui permet alors la synthèse de lois de commande efficaces. La notion de ρ -*admissibilité* offre alors un cadre rigoureux, réunissant les conditions suffisantes que doivent vérifier les fonctions de tâche pour que le problème de commande soit bien posé [97]. Nous rappelons brièvement ci-après cette notion.

2.1.1. Notion de ρ -admissibilité

La ρ -*admissibilité* permet de prouver l'existence d'une application bijective de classe C^1 dont la réciproque est aussi de classe C^1 , c'est-à-dire l'existence d'un difféomorphisme entre l'ensemble des couples (q, t) et (e, t) . Ce difféomorphisme permet alors d'établir un lien régulier entre l'espace d'état et l'espace capteur dans lequel s'exprime la loi de commande.

Ce lien est important dans la définition d'une fonction de tâche ρ -admissible, comme le souligne le théorème énoncé ci-dessous [97] :

- Théorème : Condition de ρ -admissibilité

Soit $\{e(q, t), q_0\}$ une fonction de tâche admettant une trajectoire idéale $q_r(t)$ définie sur l'intervalle de temps $\{0, T\}$. Alors :

- Si $e(q, t)$ est de classe C^2
- Si la jacobienne de la tâche $J_e = \frac{\partial e}{\partial q}$ est inversible au voisinage de la trajectoire $q_r(t)$
- Si l'horizon de temps reste fini : $T < \infty$

Il existe un rayon $\rho > 0$ tel que la tâche considérée soit « ρ -admissible ».

Ainsi, la notion de ρ -admissibilité permet de montrer d'une part qu'il y a équivalence entre la synthèse de la loi de commande dans l'espace d'état et la synthèse dans l'espace capteur, et d'autre part que la trajectoire de référence $q_r(t)$ est unique. Finalement, la condition essentielle permettant d'établir la ρ -admissibilité est de définir $e(q, t)$ de telle sorte que sa jacobienne soit inversible au voisinage de la trajectoire solution. Il nous suffit donc pour cela de contraindre autant de degrés de liberté que d'actionneurs disponibles sur le robot. L'obtention de cette propriété se réduit alors à un problème de modélisation de la tâche à effectuer. Néanmoins, certaines applications ne nécessitent pas l'utilisation de tous les degrés de liberté disponibles, et donc toute tâche robotique ne satisfait pas la propriété de ρ -admissibilité [45]. La fonction de tâche est alors redondante, et son formalisme est introduit dans la section suivante.

2.1.2 Formalisme des tâches redondantes

Une tâche redondante est une tâche qui ne contraint pas tous les degrés de liberté du robot, par conséquent la jacobienne de la fonction de tâche n'est pas carrée. Elle n'est donc pas ρ -admissible, et l'équation $e(q) = 0$ admet une infinité de trajectoires solutions $q_r(t)$. Ainsi, une tâche redondante est sous-contrainte : il n'existe pas de difféomorphisme entre l'espace d'état et l'espace capteur. L'ensemble des trajectoires solutions $q_r(t)$ forment alors un ensemble dense dans l'espace d'état : on peut ainsi passer d'une trajectoire à l'autre de façon continue [97].

L'idée du formalisme des tâches redondantes est d'exploiter la redondance du robot pour réaliser au mieux une tâche secondaire en utilisant les degrés de liberté laissés disponibles par la tâche principale e_I . La tâche secondaire étant modélisée sous la forme d'une fonction de coût h , le problème de commande consiste à minimiser $h(q)$ sous la contrainte que la tâche prioritaire e_I soit parfaitement exécutée (i.e. $e_I = 0$). Ce problème d'optimisation est équivalent à réguler à zéro la fonction de tâche globale $e(q)$ suivante [45] :

$$e(q) = W^+ e_1 + \beta (I - W^+ W) g \quad (2.1)$$

où $g = \frac{\partial h}{\partial q}$ représente le gradient de la fonction de coût $h(q)$, β est un réel positif, et I définit la matrice identité de dimension n . La matrice W^+ représente la pseudo-inverse à droite de la matrice W , calculée à l'aide de l'expression suivante « $W^T (W \cdot W^T)^{-1}$ ». Enfin, la matrice W doit être choisie telle que :

$$\ker(W) = \ker\left(\frac{\partial e_1}{\partial q}\right) \quad \text{et} \quad \left(\frac{\partial e_1}{\partial q} \times W^T\right) > 0 \quad (2.2)$$

En effet, sous ces conditions, *Samson et al.* [97] montrent que la jacobienne de la tâche $J_e = \frac{\partial e}{\partial q}$ est elle-même définie positive au voisinage de $q_r(t)$, assurant la ρ -admissibilité de la fonction de tâche globale $e(q)$.

Remarque : Afin de satisfaire les deux conditions données dans la formulation (2.2), il suffit

de prendre $W = \frac{\partial e_1}{\partial q}$

2.2 Commande référencée vision et Asservissement Visuel

La commande référencée vision représente la variété la plus répandue en termes de commande référencée capteurs, dans le domaine de la robotique. Elle consiste à contrôler les mouvements d'un système robotique en utilisant des informations visuelles, notées s , issues d'un ou plusieurs capteurs de vision, embarqués ou non sur le système [29], [62] et [63]. Dans la littérature, de nombreux travaux sont basés sur l'exploitation de ces données pour réaliser différentes tâches robotiques (positionnement face à un objet, suivi, saisie, etc.). La première

utilisation de la vision en boucle fermée date des années soixante dix, ou *Shirai & Inoue* [99] l'utilise pour augmenter la précision du positionnement. On parlait alors de retour visuel (Visual Feedback). C'est à partir de 1979 que ce type de loi de commande est appelé Asservissement Visuel (Visual Servoing) par *Hill & Park* [59].

Malheureusement, l'évolution de ce type de commande a été freinée, par les limitations des moyens de calcul, jusqu'à la fin des années 80. Au début des années 90, plusieurs études se sont alors intéressées à cette discipline émergente, essentiellement pour des applications sur des bras manipulateurs [10],[90] et [92]. Les résultats encourageants obtenus de ces travaux ont fait apparaître des recherches plus approfondies, appliquées à d'autres types de robots [29], [62]. Des études plus récentes, se sont ensuite intéressées à d'autres problèmes liés à l'application d'un asservissement visuel sur un robot, à savoir la saturation des actionneurs [14], ou encore la visibilité de la cible [18], [46]. Enfin, des capteurs de vision non conventionnels (caméra omnidirectionnel, etc) ont été introduits dans des applications robotiques comme dans [89] et [56].

L'approche de l'asservissement visuel répond à une idée intuitive du mouvement associé à la vision: le robot génère ses mouvements de façon à ce que sa cible visuelle atteigne une certaine configuration dans l'image qu'il perçoit du monde à l'aide d'une ou plusieurs caméras [65]. Il existe plusieurs types d'asservissement visuel. Par exemple, on peut utiliser dans la boucle de commande des informations tridimensionnelles, tirées du traitement d'images. On peut aussi utiliser directement les primitives visuelles, ou faire employer un mélange des deux (données 3D et données 2D en même temps) [75]. Une classification détaillée des différentes structures de commande en asservissement visuel est présentée aux sections suivantes.

2.2.1 Asservissement visuel « 3D »

Considérer la caméra comme un capteur de pose est sans doute l'approche la plus simple. Dans l'asservissement visuel 3D, la référence est exprimée sous la forme d'une attitude, notée r^* , d'un repère lié à la cible vue par la caméra par rapport à un repère lié au robot (voir figure 2.2). Cette mesure s'obtient grâce aux primitives extraites de l'image et à un modèle géométrique de la cible, mais elle est très sensible aux incertitudes sur ce modèle ainsi qu'aux erreurs de calibrage de la caméra [47]. Néanmoins, ce problème peut être contourné par la boucle fermée, ce qui évite les problèmes de convergence du système [75].

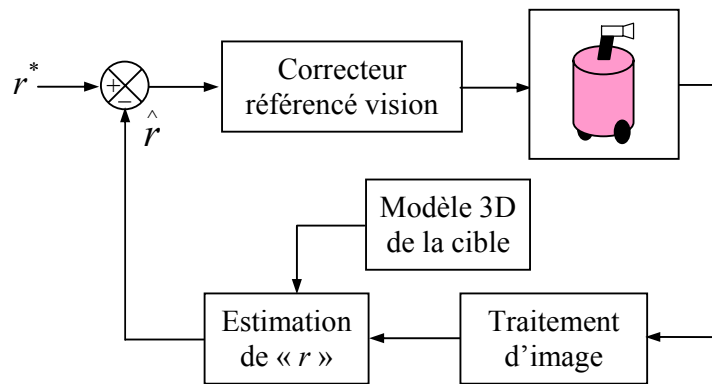


Fig. 2.2 : Asservissement visuel « 3D »

De nombreux travaux ont traité le problème de l'estimation de l'attitude r entre une ou plusieurs caméras et une cible dont la géométrie est connue. La complexité des calculs nécessaires à cette estimation croît avec la complexité de la cible. C'est pourquoi les cibles sont souvent simplifiées au maximum dans les asservissements visuels. Par exemple dans [82], la cible est constituée d'un tétraèdre dont les sommets sont matérialisés par des points lumineux. D'une autre part, de nombreux travaux mettent en œuvre une cible constituée uniquement de plusieurs disques coplanaires sur un fond uniforme comme dans [75]. Le traitement d'image se limite alors à une détection des contours des disques et à un calcul du centre de gravité des points de contour afin de déterminer le centre de chaque disque. Le modèle de la cible, à savoir la position relative de tous les centres des disques, est connu par le système de vision qui peut donc calculer une estimation de l'attitude de la cible. Il a été montré qu'il est nécessaire de connaître la projection d'au moins 4 primitives de type "point" pour pouvoir obtenir cette attitude de manière unique [109]. Si seulement trois projections sont disponibles, l'attitude ne peut être déterminée de manière unique. Toutefois, des techniques issues des recherches en reconstruction 3D par vision dynamique permettent d'estimer le modèle et de localiser la caméra à partir de mesures de mouvement 2D ou 3D, limitant ainsi l'utilisation d'un modèle de l'objet [45].

Enfin, comme l'asservissement visuel 3D s'exprime directement dans l'espace des configurations du robot, la définition de lois de commande est extrêmement simple pour aller itérativement d'une situation à une autre. Mais, il faut noter que dans cette méthode, il n'y a strictement aucun contrôle dans l'image, donc les informations visuelles nécessaires peuvent sortir de l'image pendant l'asservissement, ce qui empêche la réalisation de la tâche. De plus, cette approche requiert une interprétation du motif visuel pour caractériser la situation du robot, et nécessite donc une reconstruction de l'état du système. On y retrouve alors les

mêmes problèmes d'incertitude et de précision des schémas de commande des robots mobiles classiques [45]. De très récentes études ont montré l'efficacité de ce type de loi de commande, pour des applications d'assemblages en micromanipulation [101].

2.2.2 Asservissement visuel « 2D »

Alors que l'asservissement visuel 3D implique la connaissance d'un modèle tridimensionnel de la cible et de pouvoir la situer dans un repère commun, et de définir une consigne d'asservissement en termes de position et orientation, l'asservissement visuel 2D travaille au niveau de l'image, avec des primitives visuelles (indices) définies au préalable [36]. C'est dans l'image que la reconnaissance de la cible a lieu par ses indices, et c'est toujours dans l'image que la consigne, que doivent atteindre les indices, est définie comme indiqué sur la figure 2.3. L'algorithme n'a donc pas besoin de connaître la structure 3D de la scène et les calculs exploitent seulement l'image [65].

Une primitive est une forme géométrique élémentaire (point, segment de droite, portion d'ellipse, etc). Elle sert à modéliser la projection d'un objet dans le plan image. Les premiers travaux traitant de l'asservissement 2D ont utilisé des primitives constituées de points. Ces points peuvent par exemple être situés à l'intersection de segments dans l'image, ou encore être extraits du centre de gravité de la projection de disques. En fait, les cibles les plus couramment rencontrées sont simplement constituées de plusieurs disques coplanaires. De telles cibles ont l'avantage de requérir un temps de traitement d'image faible. Il faut noter que ce choix de primitives simples et faciles à extraire était motivé par les limitations existantes à l'époque, en termes de ressources de calcul [86].

Le pionnier en matière d'asservissement visuel 2D fut *Weiss et al.* [108]. Il étudia le lien entre le torseur cinématique du repère de la caméra et les vitesses de déplacement des primitives dans l'image. Cette relation sera baptisée « Jacobienne de l'image » par la suite dans la littérature. Son travail fut développé entre autres par *Feddema et al.* [41], notamment du point de vue expérimental. Quelques années plus tard, *Espiau et al.* [38] proposèrent un cadre plus formel au concept de la Jacobienne de l'image, qu'ils baptisèrent matrice d'interaction, notée $L_{(s,z)}$. Dans leur travail, les auteurs étendent ce concept de matrice d'interaction à différents types de primitives visuelles telles que : droites, plans, cercles, sphères. Par la suite, d'autres types de primitive ont été introduits en asservissement visuel comme les moments de l'image [16], ou encore très récemment, les primitives photométriques (luminance en tout point de l'image) [24].

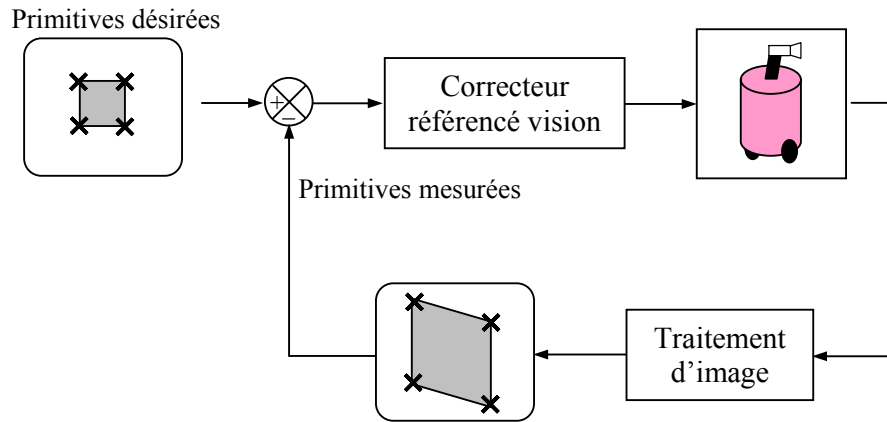


Fig. 2.3 : Asservissement visuel « 2D »

Pour ce qui est du problème du calcul de la matrice d'interaction $L_{(s,z)}$, ses éléments s'écrivent, par définition, en fonction des paramètres suivants :

- Les constantes issues du calibrage du capteur visuel.
- L'état courant des indices visuels s .
- La profondeur z .

La question est donc de savoir comment mettre à jour cette matrice dans la boucle d'asservissement. Initialement, deux cas de figure étaient possibles, selon que $L_{(s,z)}$ est calculée hors ligne ou en ligne :

- $L_{(s,z)}$ calculée hors ligne :

Dans ce cas, elle n'est calculée qu'une seule fois avant l'asservissement, en prenant les indices visuels au point d'équilibre souhaité s^* comme suit : ($L_{(s,z)} = L_{(s^*,z^*)}$). Cette approche allège les calculs dans la boucle d'asservissement tout en préservant sa convergence [65]. Toutefois, des difficultés sont enregistrées quand la position initiale est loin de celle désirée [86].

- $L_{(s,z)}$ calculée en ligne :

Nous considérons dans ce cas qu'elle est calculée à chaque itération en fonction des primitives visuelles courantes s comme suit : ($L_{(s,z)} = L_{(s(q,t), \hat{z})}$). Cette matrice dépend alors d'une information métrique (la profondeur z) [65]. Une estimation de z (notée \hat{z}) peut se faire si on dispose d'un modèle 3D de la cible [12].

Ces deux possibilités présentent chacune un certain nombre d'avantages, mais aussi des inconvénients [12] :

- D'une part, le calcul de $L_{(s,z)}$ en ligne contraint fortement la trajectoire que doivent suivre les informations visuelles dans l'image pour atteindre la position désirée. Par exemple, si l'on choisit des points, ce choix implique que chaque point doit aller en ligne droite vers sa position désirée (voir figure 2.4.a). Cependant, ce comportement souhaitable dans l'image peut parfois entraîner des mouvements inadéquats de la caméra (singularité), voire même impossible à réaliser. La caméra peut ainsi atteindre un minimum local de l'erreur dans l'image [86].
- D'autre part, calculer $L_{(s,z)}$ hors ligne ne contraint quasiment pas la trajectoire dans l'image, comme l'illustre la figure 2.4.b. Du coup, et même si nous ne pouvons pas l'expliquer, il s'avère en pratique que la commande est alors beaucoup moins sujette aux problèmes des mouvements irréalisables et des minima locaux. Par contre, il est tout à fait possible que la cible sorte du champ de vue de la caméra au cours du positionnement [86].

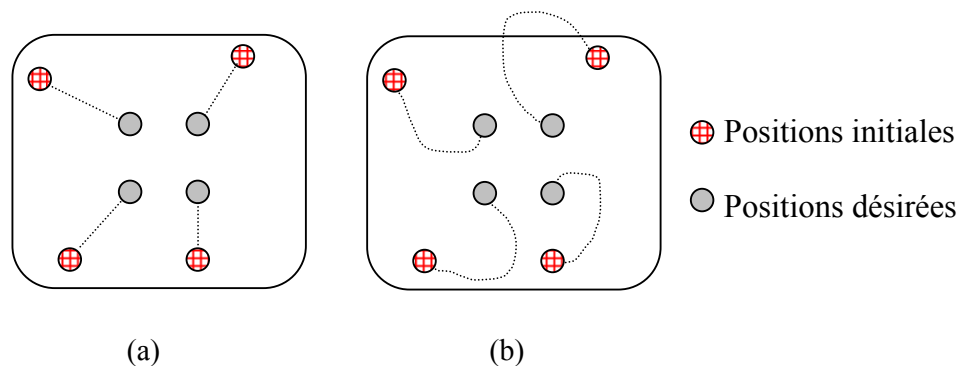


Fig. 2.4 : Trajectoires des primitives visuelles dans l'image

Des études plus récentes ont testé d'autres possibilités de calcul de la matrice d'interaction, que celles citées plus haut. Il a été constaté qu'une combinaison linéaire des deux formes précédentes donnait de meilleurs résultats [80], [64]. Il en résulte une matrice d'interaction de la forme $L_{(s,z)} = \{L_{(s^*,z^*)} + L_{(s(q,t), \hat{z})}\}/2$. Toutefois, Il existe une possibilité de rencontrer une singularité de la matrice $L_{(s,z)}$ pour tous les cas précédents, ce qui entraîne une instabilité ou même une divergence [12]. L'asservissement 2D est également adapté à des cibles complexes ne contenant aucune primitive simple. Une technique de flot optique (*Optical Flow* en anglais) est alors utilisée pour déterminer le mouvement de zones d'intérêt

dans l'image [90]. Il est montré dans [39] une robustesse de cette méthode par rapport aux variations des paramètres intrinsèques de la caméra.

Néanmoins, les asservissements visuels 2D sont, d'une manière générale, des lois de commandes relativement rapides à calculer, puisqu'il n'y a pas de phase de reconstruction 3D. Ce gain de temps n'est pas négligeable puisque le délai d'application de la commande est réduit, ce qui contribue à la stabilité du système [45]. De plus, les asservissements basés sur l'image permettent la réalisation de tâches de manière très efficace et précise. C'est ainsi que ce type de commande se rencontre de plus en plus dans différents domaines d'application. Citons par exemple la conduite d'engins sous-marins [95], de robot de type voiture en milieu urbain [96], de véhicules autonomes personnels [83], d'engins agricoles [71] ou encore d'engins volants [8]. La robotique chirurgicale apparaît également comme un champ d'application privilégié des dernières avancées de l'asservissement visuel 2D comme dans [73] et [106], tout comme la réalité virtuelle [81].

Enfin, la synthèse d'un asservissement visuel 2D repose directement sur le formalisme des fonctions de tâche, présenté précédemment. Cette technique offre un cadre rigoureux pour la synthèse de lois de commande référencées capteurs [45]. Initialement conçu pour des bras manipulateurs [10] [38], ce formalisme ne peut être directement étendu au cas des robots mobiles non-holonomes du fait de la contrainte de roulement sans glissement. En effet, cette contrainte entrave les mouvements de la caméra, et les tâches robotiques nécessitent alors la réalisation de manœuvres qui peuvent conduire à la perte du motif visuel. Une première solution, proposée par *Pissard-Gibollet et al.* [92], a consisté à introduire (au moins) un degré de liberté supplémentaire permettant à la caméra de se déplacer indépendamment du robot mobile. Ainsi, bien que la base mobile reste non-holonyme, les mouvements de la caméra deviennent holonomes, et il est alors possible d'appliquer le formalisme des fonctions de tâche. Plus récemment, un cadre plus générique, communément appelé le formalisme des fonctions transverses [85] a vu le jour. Cette approche permet la synthèse de schémas de commande pour la stabilisation de systèmes non-linéaires.

Toutefois, l'asservissement visuel seul ne permet pas, dans certains cas, de décrire des tâches robotiques élaborées et de tenir compte d'événements extérieurs. De ce fait, d'autres études ont abordé le problème d'exploitation des données sensorielles de nature différente, dans le but de caractériser d'une part l'objectif de la mission, et d'autre part, de gérer la présence éventuelle d'obstacles [9] [45]. Ainsi, un capteur de force a été couplé à une caméra pour réaliser le positionnement du bras manipulateur par commande référencée capteurs [77],

alors que les données des capteurs ultrasons sont exploitées pour éviter des obstacles statiques lors d'une tâche d'asservissement visuel dans [44].

2.2.3 Asservissement visuel « $2D \frac{1}{2}$ »

Cette technique, proposée par *Chaumette et al.* [12] [76], est une variante de la loi de commande habituellement utilisée pour un asservissement 2D. Nous décrivons à présent cette approche d'asservissement visuel, qui a été développée dans le cadre de la thèse d'*Ezio Malis* [75]. Elle est particulièrement intéressante puisqu'elle ne présente pas la plupart des inconvénients respectifs des deux approches précédemment présentées. Il s'agit de l'asservissement visuel $2D \frac{1}{2}$, car les informations utilisées comme mesures et consignes sont, pour certaines d'entre elles, exprimées directement dans l'image et, pour les autres, exprimées dans le repère de la caméra.

Plus précisément, cette approche est basée sur l'estimation de l'homographie H qui relie l'image de trois points [40]. Dans le cas d'une cible plane, un minimum de quatre points appariés sur les images courantes et désirées permet une telle estimation par la résolution d'un simple système linéaire. Dans le cas d'une cible quelconque, l'appariement de huit points permet d'aboutir, là encore par la résolution de systèmes linéaires [12]. A partir de l'homographie obtenue, il est possible de calculer le déplacement en rotation R que la caméra doit effectuer pour atteindre sa position spécifiée, ainsi que la direction de son déplacement en translation. L'homographie fournit également le rapport z / z^* entre les distances courantes et désirées de la caméra à la cible (voir figure 2.5). On peut ainsi séparer les boucles d'asservissement en rotation et en translation de la caméra, ce qui permet :

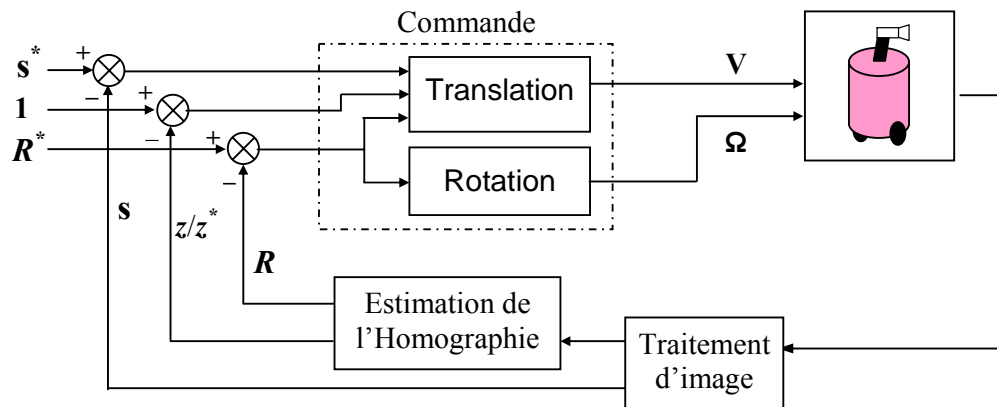


Fig. 2.5 : Asservissement visuel « $2D \frac{1}{2}$ »

- Un fort découplage de la loi de commande.

- Un contrôle partiel dans l'image permettant de conserver la cible en permanence dans le champ de vision de la caméra.
- Une étude de la stabilité et du domaine de convergence de la loi de commande.

Il a été ainsi obtenu des conditions suffisantes de stabilité prenant en compte les erreurs de calibrage et les erreurs d'estimation de l'homographie qui en découlent. De ce fait, pour des positions initiales de la caméra très éloignées de la position désirée [12], la convergence est obtenue là où l'asservissement visuel 2D échoue. L'avantage majeur de cette approche est que la connaissance d'un modèle géométrique 3D de la cible n'est plus nécessaire, ce qui lui donne un domaine d'application très important à partir du moment où le motif à atteindre est préalablement déterminé, par exemple, lors d'une phase d'apprentissage. La seule information 3D utilisée dans la commande est la profondeur désirée z^* approximative d'un point de la cible, soit, un besoin d'informations à priori bien moindre qu'en asservissement visuel 3D et 2D.

2.2.4 Asservissement visuel « $d2D/dt$ »

Les méthodes que nous avons décrites jusqu'à présent reposaient sur l'utilisation d'informations visuelles géométriques (coordonnées de points, paramètres représentant l'image d'une droite, etc.). Les contraintes imposées sont donc que ces primitives géométriques existent dans la scène, mais surtout qu'il est possible de les extraire et de les suivre par traitement d'image à une cadence assez élevée afin de conserver la robustesse et la stabilité des lois de commande [12]. Actuellement, les systèmes les plus fiables sont limités soit au suivi de segments, soit au suivi de points caractéristiques ou de formes simples.

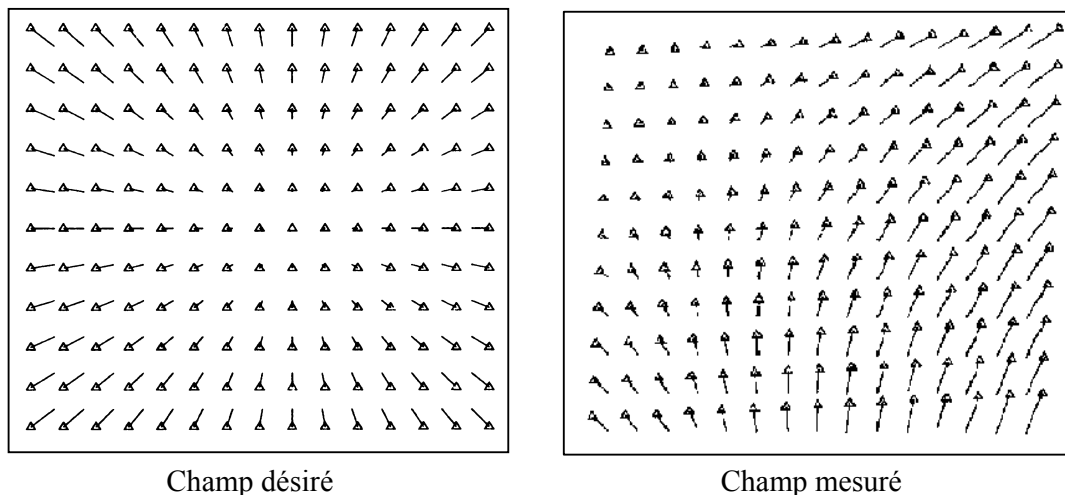


Fig. 2.6 : Exemples de champs de vitesse

En pratique, on utilise le plus souvent des marqueurs disposés sur la scène observée afin d'extraire les positions de leur projection perspective dans l'image. D'autres travaux ont permis de supprimer ces contraintes fortes par l'emploi d'informations visuelles dynamiques comme dans [100]. Ce type d'asservissement visuel utilise une vitesse relative entre la caméra et la cible comme grandeur asservie. La référence est définie dans le plan image par un champ de vitesse des points. Le principe de la commande consiste alors à contrôler les mouvements de la caméra de telle sorte que le mouvement 2D mesuré atteigne un champ de vitesse désirée (voir figure 2.6), d'où l'appellation d'asservissement visuel $d2D/dt$. Contrairement au cas de mesure géométrique, la mesure s ainsi que sa valeur désirée s^* correspondent désormais à des paramètres P_i du modèle de mouvement 2D dans l'image. Dans ce cas alors, la loi de commande ne contraint plus seulement le torseur cinématique de la caméra τ mais aussi son accélération Γ (voir figure 2.7). La mesure s peut être établie par des techniques classiques de flot optique. Une fonction quadratique est souvent choisie pour modéliser le champ de vitesse [12]. La loi de commande est définie par rapport aux paramètres de cette fonction. Par ailleurs, l'emploi de ces nouvelles informations visuelles permet d'appréhender de manière plus simple des tâches extrêmement délicates à modéliser en utilisant des informations visuelles géométriques.

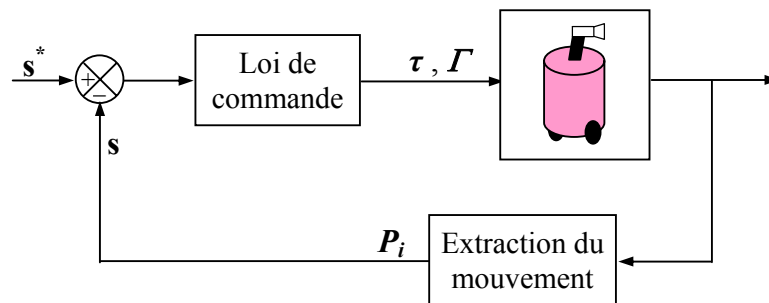


Fig. 2.7 : Asservissement visuel « $d2D/dt$ »

L'asservissement visuel $d2D/dt$ n'a pas connu une très grande utilisation, et reste aujourd'hui encore à un stade émergent, car les traitements d'image qu'il nécessite sont encore très longs et fournissent des informations très bruitées. Des applications pratiques ont néanmoins été réalisées. Par exemple, un asservissement de l'alignement de l'axe optique d'une caméra animée d'un mouvement de translation suivant la direction de cette translation est décrite dans [100]. On note aussi que, dans le cadre de sa thèse, *A. Crétual* [33] a réalisé une tâche de positionnement telle que le plan image de la caméra soit parallèle au plan d'une cible observée [31]. Dans un cadre plus général, la relation linéaire entre les variations de l'information visuelle et la vitesse de la caméra, représentée par la matrice d'interaction pour

des informations visuelles géométriques, n'est généralement plus valide, en particulier en raison de l'apparition de termes liés à l'accélération de la caméra et de termes quadratiques. Actuellement, les travaux sont plus orientés vers l'élaboration de lois de commande en accélération ou, si besoin, combinant accélération et vitesse.

Il faut finalement signaler que, même si des progrès énormes ont été faits récemment, les algorithmes de traitements d'image utilisés sont encore relativement coûteux en temps de calcul si l'on considère un modèle affine du mouvement (et encore davantage pour un modèle quadratique) [12]. Ils fournissent en outre des résultats beaucoup plus bruités que ceux employés pour extraire des informations visuelles géométriques. À des cadences de l'ordre de 1 Hertz, la validation expérimentale des techniques d'asservissement visuel $d2D/dt$ est donc particulièrement délicate. Par ailleurs, il est possible de remonter à des informations visuelles géométriques par simple intégration des paramètres de mouvement estimés (si une localisation initiale est disponible) [12] [100] [34]. On peut alors utiliser directement les lois de commande qui ont été développées en asservissement visuel 2D. Cette approche a été validée sur une tâche de poursuite d'une cible quelconque (tel un piéton par exemple) en contrôlant l'orientation de la caméra [32].

Les résultats obtenus de l'asservissement visuel dans diverses applications, ainsi que l'amélioration rapide des performances des systèmes de vision, tant du point de vue de la cadence d'acquisition que de la vitesse de traitement, a poussé les chercheurs, ces dernières années, à essayer d'élargir les capacités de cette technique de commande. Pour cela, on a vu apparaître une variante plus adaptée aux applications industrielles, dite asservissement visuel rapide [86].

2.2.5 L'Asservissement visuel « rapide »

Un asservissement visuel rapide, comme son nom l'indique, met en œuvre un système de vision très rapide dont la cadence d'échantillonnage et de traitement de l'image permet d'exploiter au maximum les performances dynamiques du robot. Il est considéré comme rapide tout asservissement visuel à une cadence $> 50\text{Hz}$ [86]. Ce nouveau type, contrairement aux autres traités avant, permet de tirer partie de toutes les possibilités d'un robot manipulateur en termes d'accélération et de vitesse [47]. Néanmoins, afin d'obtenir un contrôle optimal, il est nécessaire de modéliser précisément les dynamiques du manipulateur impliqué dans la boucle de vision [69]. De plus, dans un contexte de suivi de cible mobile, le modèle dynamique de ce dernier permet d'établir une prédiction de sa position. Cette prédiction peut être exploitée pour améliorer la performance du traitement d'image et pour

augmenter la rapidité du suivi en utilisant par exemple un contrôleur prédictif. On note aussi que pour ce type d'asservissement, la structure directe est sûrement la plus convenable [86].

Le premier chercheur à avoir étudié de manière détaillée l'influence des dynamiques du manipulateur dans le contrôle de la boucle de vision fut *Corke* dans [29] et [30]. Les paramètres du contrôleur de la boucle de vision sont réglés grâce à une technique de placement de pôles. Le modèle dynamique du manipulateur est déterminé hors ligne, contrairement à ce qui est fait dans [107]. Aussi, une cadence de 50Hz suffit à réaliser l'asservissement. Néanmoins, cette étude est restreinte à des systèmes relativement simples : au maximum, 3 degrés de liberté sont asservis. Enfin, l'un des travaux traitant largement ce type d'asservissement est celui réalisé dans le cadre de la thèse de *J. Gangloff* [47]. L'auteur a proposé une modélisation dynamique d'un robot à 6 degrés de liberté, qui a été appliquée à deux études d'asservissement visuel :

- La première application est un suivi de cible où deux stratégies d'asservissement ont été expérimentées : l'approche 3D et l'approche 2D avec deux correcteurs différents (correcteur classique PI et correcteur GPC).
- La seconde application est un suivi de profilé, où il a été mis en évidence l'utilité de l'approche prédictive.

2.3 Asservissement Visuel et Robustesse

La robustesse est une propriété centrale et déterminante dans le choix d'une loi de commande, plus particulièrement pour les systèmes à modèle partiellement ou complètement inconnu, ou ceux ayant un modèle non-linéaire. Pour ce qui est de l'asservissement visuel, plusieurs études ont été menées pour évaluer la stabilité et la robustesse de ce type de commande [78] en vue d'apporter des améliorations. Par exemple, les auteurs dans [84] et [79] proposent une loi de commande robuste vis-à-vis des erreurs de calibration. De même, des améliorations de la robustesse par rapport aux bruits induits par le traitement d'image, pour l'extraction des primitives ont été réalisées dans [27] et [26].

D'autres approches ont proposé la modification du type de loi de commande utilisée en asservissement visuel. Ainsi, *Hashimoto et al.* [58] ont utilisé une commande optimale de type LQG pour annuler la fonction de tâche et asservir un bras manipulateur. De même, une commande de type LQ est utilisée dans [90], alors qu'une technique robuste basée sur la commande de type H_∞ est employée pour guider un véhicule dans [70], ou encore un retour d'état continu instationnaire dans le cas d'un robot mobile non-holonomes [102].

Des travaux plus récents ont fait appel à des outils mathématiques, comme les LMI (*Linear Matrix Inequality*) pour améliorer la robustesse de la loi de commande en boucle fermée [35], [19] et [4]. Les commandes ainsi synthétisées garantissent a priori certaines propriétés telles que la convergence, la non-saturation des actionneurs, la visibilité de l'objet, la satisfaction de contraintes 3D pendant le déplacement, etc. De même, *B. Gao* [48] a proposé une extension de ces dernières approches dont la méthode développée permet de caractériser un correcteur et un domaine de stabilité associé, prenant en compte une incertitude bornée sur la profondeur des points de la cible, la visibilité des indices visuels et les contraintes sur la vitesse et l'accélération de la caméra. Toutefois, ces méthodes sont généralement conservatives, c'est-à-dire que la commande générée est très faible, et donc que le délai de convergence est long [45]. Un récapitulatif détaillé et complet de l'utilisation de ces techniques numériques en asservissement visuel est dans le livre de *Chesi et al.* [20].

Enfin, d'autres types de loi de commande, réputées robustes, peuvent être utilisées en vue d'améliorer les performances en robustesse de l'asservissement visuel. Quelques études ont utilisé la commande à structure variable basée sur les régimes glissants, mais, en majorité, son application concernait l'asservissement visuel de bras manipulateurs. On cite comme exemple les travaux de *Zanne et al.* [110] ou encore ceux de *Joon-Kil et al.* [67]. Toutefois, les auteurs de [3] ont implémenté une commande par modes glissants pour la navigation d'un robot mobile par asservissement visuel.

2.4 Poursuite référencée capteurs de cible mobile

Nous nous intéressons dans cette partie à la tâche de poursuite de cible mobile par une commande référencée capteurs. Un grand nombre de travaux de recherche traitant ce problème ont employé un capteur de vision (caméra ou banc stéréo), mais rares sont ceux qui ont utilisé un asservissement visuel proprement dit. Le plus souvent, ces travaux se focalisent plus sur l'aspect détection de la cible, son identification et le tracking sensoriel, et beaucoup moins sur l'aspect stratégie ou lois de commande garantissant la poursuite. On peut citer le travail de *Kwon et al.* [74] dans lequel un robot mobile poursuit une personne, en utilisant les informations issues de deux caméras non calibrées, sans que la loi de commande ne soit explicite. Aussi, dans [17] une poursuite de personne mobile a été implémentée en utilisant un régulateur classique de type proportionnel, référencé vision.

Néanmoins, certains travaux de recherche en asservissement visuel ont traité ce cas de figure, mais en considérant, dans leur majorité, le mouvement de la cible comme une perturbation à rejeter le plus rapidement et le plus efficacement possible. On peut citer les

exemples des travaux publiés dans [99] et [90]. D'autres études utilisent des connaissances a priori sur la trajectoire de l'objet comme dans [57] et [94]. Enfin, l'introduction d'un intégrateur dans la loi de commande est très classique en automatique pour éliminer l'erreur de trainage, mais l'utilisation de cette technique ne permet de traiter que les cas où la cible se déplace à vitesse constante [15]. À partir de ces résultats, d'autres travaux ont utilisé des estimateurs pour mesurer, le plus précisément possible, la vitesse de la cible dans l'image. Cela permet d'aboutir à une commande adaptative, où un filtre de *Kalman* est généralement utilisé pour lisser l'estimation [28]. De même, un modèle d'état à accélération constante et bruit corrélé a été employé pour l'estimation de la vitesse de la cible [11]. Aussi, *Bensalah et al.* [5] ont utilisé l'algorithme du GLR (*Generalised Likelihood Ratio*) afin de détecter, d'estimer et de compenser d'éventuels sauts en direction ou en amplitude de la vitesse de la cible.

Cependant, il existe dans la littérature des techniques de détection et de poursuite de personne mobile employant d'autre type de perception que la vision artificielle, comme le capteur à laser [55]. Plus récemment, des travaux se sont focalisés sur les technologies dites « émergentes », comme celles basées sur les réseaux sans fil, les ultrasons, les infrarouges [98] ou les badges Radio Fréquence (Tag RFID) [68]. Plus encore, des études ont montré que la combinaison de plusieurs informations sensorielles donne de meilleures performances en termes de détection [2], et peut s'avérer plus efficace pour développer des tâches robotiques plus évoluées. Les auteurs dans [72] ont utilisé à la fois une caméra omnidirectionnelle et un capteur laser pour suivre un utilisateur. De même, une caméra thermique a été associée à une caméra classique pour prendre en charge et gérer des occultations dans une tâche de poursuite [23].

Dans le cadre de cette thèse, nous nous intéressons à implémenter une tâches de positionnement par rapport à une personne connue, puis une poursuite en utilisant une commande référencée capteurs. Nous commençons alors au chapitre 3 par synthétiser un asservissement visuel 2D reposant sur le formalisme des fonctions de tâche. Par la suite, nous nous focalisons au chapitre suivant sur la nature de la loi de commande employée, en introduisant la technique des modes glissants dans l'asservissement visuel, en vue d'obtenir une meilleure robustesse vis-à-vis des erreurs de calibration de la caméra, des erreurs de mesure des indices visuels et celles liées aux mouvements de la cible. Enfin, nous introduisons le concept de la commande Multimodale dans le cadre d'une tâche d'interaction Homme/Robot.

3. CONCLUSION

Dans ce chapitre, nous avons rappelé quelques notions de base sur la commande référencée capteurs, et plus particulièrement celles de l'asservissement visuel en vue de son application sur un robot mobile non-holonome. Puis, nous avons présenté un état de l'art de cette discipline et cité ses différents types. Le chapitre suivant est consacré, à la fois, à la modélisation du système robotique considéré dans cette étude, et à la synthèse d'un asservissement visuel 2D basé sur le formalisme des fonctions de tâche. Des résultats de simulations et d'implémentation expérimentale seront présentés pour évaluer les performances de cet asservissement appliqué sur l'homme.

CHAPITRE 3

ASSERVISSEMENT VISUEL 2D D'UN ROBOT MOBILE NON-HOLONOME

1. INTRODUCTION

Les techniques de commande référencées capteurs, présentées dans le chapitre précédent, permettent à un robot de réaliser des tâches de diverses natures. Dans ce chapitre, nous proposons d'étudier le problème de l'utilisation de ces techniques pour implémenter un asservissement visuel d'un robot mobile non-holonome sur l'homme. Il s'agit de faire converger les indices visuels caractérisant un utilisateur connu, vers une configuration prédéfinie, en se basant uniquement sur la détection et la reconnaissance de son visage.

Dans un premier temps, nous commençons par présenter une modélisation explicite et détaillée de la plateforme utilisée (robot *Rackham*), puis présentons les fonctionnalités des deux systèmes de détection et d'identification de l'utilisateur (vision artificielle et RFID). Ensuite, nous exposons les principes de base de l'asservissement visuel 2D nécessaires à l'élaboration de ce travail. Enfin, nous présentons les résultats de simulation et ceux obtenus de l'implémentation expérimentale sur le robot considéré.

2. MODELISATION DU SYSTEME ROBOTIQUE

L'implémentation d'une loi de commande sur un système quelconque ne peut se faire correctement sans passer par une phase de modélisation (ou d'identification du modèle). Cette dernière doit être menée avec rigueur, en vue d'obtenir le modèle le plus proche de la réalité, et garantir de meilleures performances en termes de stabilité et de précision. En effet, l'implémentation d'une loi de commande référencée vision sur un robot mobile requiert la détermination du lien entre l'évolution des indices visuels dans l'image et les mouvements de la caméra. Dans ce but, nous décrivons dans le paragraphe suivant la plateforme expérimentale utilisée dans ce travail. Nous nous intéressons aussi à la modélisation des capteurs embarqués, et à leurs interactions avec l'environnement.

2.1 Système de vision

Le robot considéré est muni d'un système de vision active, composé d'une caméra CCD classique, délivrant des images à une résolution de 320x240 pixels. Elle est montée sur une tourelle commandable en tangage et en lacet, appelé PTU (Pan/Tilt Unit en Anglais). Ci-après la présentation du modèle du capteur de vision utilisé :

2.1.1 Modèle de la caméra

Dans ce travail, nous considérons le modèle dit *sténopé* de la caméra (ou trou d'épingle) [45], comme il est détaillé ci-après. Ce modèle est le plus couramment utilisé en vision par ordinateur, car il permet de modéliser finement la plupart des capteurs projectifs et de simplifier les équations mises en jeu [59]. Dans ce modèle, le point I dans le plan image est appelé point principal. La droite perpendiculaire au plan image passant par I , comme le montre la figure 3.1, constitue l'axe optique. Le point C placé sur l'axe optique à une distance f du plan image est le centre de projection, alors que f représente la distance focale de la caméra. Du moment où on utilise une focale de faible dimension, le modèle sténopé prend pour hypothèse que tous les rayons passent par le centre optique C . De ce fait, les points sont projetés sur le plan image par une projection perspective [45].

Ainsi, un point P dans le champ de vision de la caméra se projette dans l'image le long d'une droite passant par p et C (voir figure 3.1). Choisissons comme système de coordonnées lié à la caméra le repère R_C : l'origine de ce repère se trouve en C alors que le plan x_c-y_c de ce repère est parallèle au plan image, et l'axe z_c est confondu avec l'axe optique. Soient x , y et z

les coordonnées de P dans le repère R_C . Dans ce même repère, les coordonnées de p (sa projection sur le plan image) sont x', y' et z' , exprimés comme suit :

$$\begin{cases} x' = f \frac{x}{z} \\ y' = f \frac{y}{z} \\ z' = f \end{cases} \quad (3.1)$$

Les coordonnées homogènes de P sont $(x, y, z, 1)$ alors que celles de p sont $(x', y', z', 1)$, donc cette écriture peut se mettre sous la forme matricielle comme suit :

$$\begin{pmatrix} sx' \\ sy' \\ sz' \\ s \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3.2)$$

$$\text{avec } M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \text{ et } s = \frac{z}{f}.$$

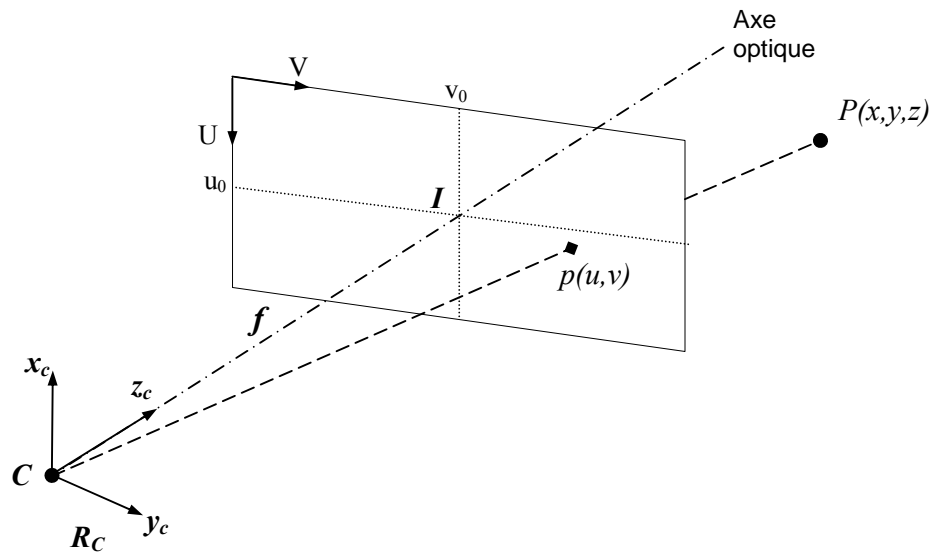


Fig. 3.1 : Modèle géométrique de la caméra

Les points image sont mesurés en pixels dans un repère bidimensionnel $U-V$ associé à l'image (voir figure 3.1). Afin de pouvoir écrire la matrice de transformation du repère

caméra au repère image, nous devons introduire les paramètres suivants : \mathbf{u}_0 , \mathbf{v}_0 qui sont les coordonnées de \mathbf{C} dans le repère image (en pixels). Soit k_u et k_v les facteurs d'échelle verticale et horizontale respectivement, exprimés en pixels/mm. En effet, les pixels d'une caméra sont rarement carrés. La transformation du repère caméra au repère image s'écrit (pour le point \mathbf{p}) [86]:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \\ w_0 \end{pmatrix} \quad (3.3)$$

C'est une transformation affine représentant un changement d'échelle, une rotation et une translation. La composante w étant toujours nulle, on peut ignorer la troisième ligne et écrire cette transformation sous la forme d'une matrice 3×4 . Cette transformation représente une application linéaire de l'espace projectif vers le plan projectif :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \mathbf{K} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \quad (3.4)$$

$$\text{avec } \mathbf{K} = \begin{pmatrix} -k_u & 0 & 0 & u_0 \\ 0 & k_v & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En multipliant les matrices \mathbf{K} et \mathbf{M} (projection perspective suivie d'une transformation affine) nous pouvons écrire les équations du modèle géométrique de la caméra. Soit la relation entre les coordonnées caméra (x, y, z) du point \mathbf{P} et ses coordonnées image (u, v) comme suit :

$$\begin{cases} u = -k_u f \frac{x}{z} + u_0 \\ v = k_v f \frac{y}{z} + v_0 \end{cases} \quad (3.5)$$

$$\text{avec } \mathbf{K} \times \mathbf{M} = \begin{pmatrix} -k_u & 0 & u_0/f & 0 \\ 0 & k_v & v_0/f & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix}$$

En multipliant les éléments de la matrice $\mathbf{K} \times \mathbf{M}$ par f (ce qui ne change pas le résultat, puisque les coordonnées homogènes sont définies à un facteur multiplicatif près), on obtient une autre matrice \mathbf{I}_C tel que :

$$\mathbf{I}_C = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.6)$$

avec : $\alpha_u = -k_u \cdot f$ et $\alpha_v = k_v \cdot f$

C'est une application linéaire de l'espace projectif vers le plan projectif exprimant la transformation perspective :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \mathbf{I}_C \begin{pmatrix} x \\ y \\ z \\ s \end{pmatrix} \quad (3.7)$$

Ce modèle comporte quatre paramètres : α_u , α_v , u_0 et v_0 . Ce sont ces paramètres qui décrivent les propriétés physiques du capteur, et qui sont estimés après calibrage. Notons que la distance focale ne peut être calculée explicitement, ce qui nous amène à utiliser la valeur nominale indiquée par le constructeur de la caméra. En introduisant des coordonnées caméra sans dimension telles que :

$$\mathbf{X} = x/z \quad \mathbf{Y} = y/z \quad \mathbf{Z} = 1$$

On peut maintenant écrire la relation entre les coordonnées image et les coordonnées caméra :

$$\begin{cases} u = \alpha_u X + u_0 \\ v = \alpha_v Y + v_0 \end{cases} \quad (3.8)$$

Soit sous forme matricielle :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (3.9)$$

$$\text{avec : } A = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Donc, on peut décomposer la matrice I_C en une transformation affine caméra/image (la matrice A) et une transformation projective :

$$I_C = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.10)$$

Comme il a été mentionné, le calibrage de la caméra est une étape nécessaire, afin d'identifier ses paramètres intrinsèques. Pour cela, nous avons procédé au calibrage au moyen de l'application « *Camera Calibration Toolbox* » sous *Matlab* disponible sur le net [7]. Les paramètres intrinsèques correspondant à la caméra numérique utilisée au cours de nos expérimentations sont réunis dans le tableau 3.1.

Tableau 3.1 : Résultats du calibrage de la caméra

Distance focale (donnée constructeur)	$f = 4.5 \text{ mm}$
Centre optique	$u_0 = 122.71 \text{ pix}$ $v_0 = 159.05 \text{ pix}$
Facteur d'échelle vertical	$\alpha_u = -331.75 \text{ pix/m}$
Facteur d'échelle horizontal	$\alpha_v = 441.36 \text{ pix/m}$

2.1.2 Modèle de la PTU



Fig. 3.2 : Tourelle Pan/Tilt (PTU)

Le robot considéré dans cette étude est équipé d'une caméra montée sur une tourelle à deux axes de type « PTU-D46 », de la marque « Directed Perception » (figure 3.2). Elle est aussi appelée *Pan/Tilt Unit* (PTU), car elle permet à la caméra d'avoir deux degrés de liberté :

- Le pan w_p est la rotation autour de l'axe \vec{z}_p du repère R_P lié à la PTU (voir figure 3.6). La position angulaire est notée φ_p .
- Le tilt w_t est la rotation autour de l'axe \vec{y}_p du repère R_P lié à la PTU (voir figure 3.6). La position angulaire est notée φ_t .

Plus de détails concernant les repères du système robotique considéré sont donnés dans la section 2.2 de ce chapitre. Il est important de noter que le fait que la caméra ait la possibilité de tourner dans le plan horizontal permet de diminuer considérablement l'angle mort de la perception (voir figure 3.3).

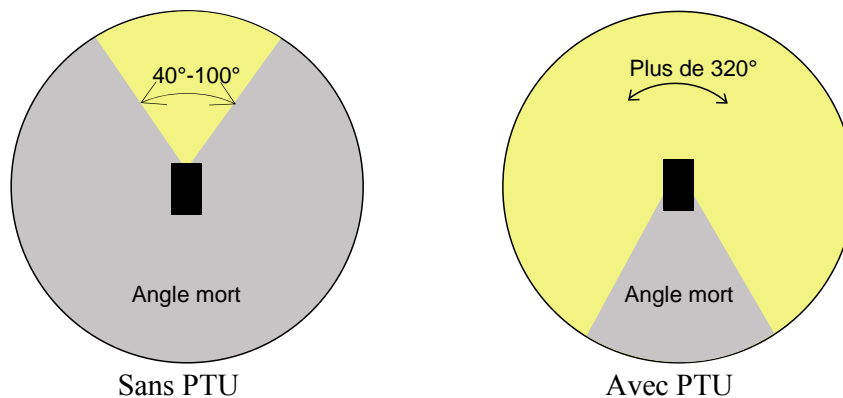


Fig. 3.3 : Angle de vue de la caméra

La PTU montée sur le robot *Rackham* possède les caractéristiques techniques suivantes :

- Vitesse de rotation maximale : 35 °/s
- Résolution : 0,01°
- Charge supportée : 4,08 kg
- Recalibrage automatique au démarrage
- Contrôle par liaison série RS-232

2.1.3 Détection et reconnaissance visuelle de l'utilisateur

Le software (ou module) de détection et reconnaissance de visage, appelé *ICU*, est un outil dédié aux modalités de suivi de personnes, et qui a été développé au *LAAS-CNRS* [105]. Il se base sur les filtrages particuliers ainsi que sur plusieurs méthodes de détection et d'évaluation. Ainsi, la détection de visage et un algorithme de reconnaissance de visages « *Eigenfaces* » permettent de cibler les recherches, alors que des mesures de contour et de couleur valident les choix du suivi [105]. Le module *ICU* est utilisé pour les tâches d'interaction Homme / Robot sur des distances allant de 1 à 3 m autour du robot. Ce module comprend différentes modalités de détection / suivi de personnes, de reconnaissance de visage basée sur les Visages Propres [49] [50] dans le but d'identifier une personne tout au long du suivi. Cette personne peut être apprise pendant l'interaction à condition que l'utilisateur soit à une distance < 1 m. La bibliothèque *OpenCV* a aussi été utilisée pour l'extraction d'indices visuels de bas niveau comme les contours ou la détection de visages. Les fonctions implémentées dans *ICU* sont organisées en trois catégories principales :

A. Les détecteurs :

Il existe trois détecteurs différents dans *ICU* permettant d'initialiser les filtres particuliers.

- a. Le détecteur de visage utilise le classificateur basé sur les masques de *Haar* décrit par *Viola et al.* dans [105]. Le résultat de ce détecteur est visualisable par l'appel de la modalité « *Face Detection* ». De plus, la position de chaque visage détecté est stockée, à chaque itération, dans une mémoire partagée (poster « *Results* » du module *ICU*).
- b. Le détecteur de blobs en mouvement utilise la différence entre l'image courante et l'image précédente pour extraire toutes les zones en mouvement dans l'image. Le résultat de ce détecteur est visualisable par l'appel de la modalité « *Motion Detection* ».
- c. Le détecteur de blobs de couleur peau permet d'extraire toutes les zones de l'image dont la couleur est semblable à celle d'une couleur de peau moyenne

(chargée depuis un fichier contenant un histogramme appris a priori). Ce détecteur ne sert que dans la boucle de suivi, c'est pourquoi son résultat n'est pas visualisable.

B. La reconnaissance de visages :

Elle se base sur la méthode des visages propres décrite dans [49] et [50]. Il est possible d'ajouter ou de supprimer des personnes de la liste des personnes connues du robot (requête *FaceRecLoadSombody*, et requête *FaceRecUnloadSombody*). De plus, il est possible d'apprendre de nouvelles personnes afin d'enrichir la base de connaissance (requête *FaceRecAddSomebody*). Enfin, Il est possible de se focaliser sur une seule personne (requête *FaceRecSetPeople*). Le résultat de la reconnaissance est visualisable par l'appel de la modalité « *Face Recognition* ». Quelques résultats de reconnaissance sont visibles sur la figure 3.4. Basée sur la reconnaissance de visage, une table de présence est mise à jour à chaque itération, permettant de savoir à n'importe quel moment qui est présent ou pas dans le champ de vision du robot.

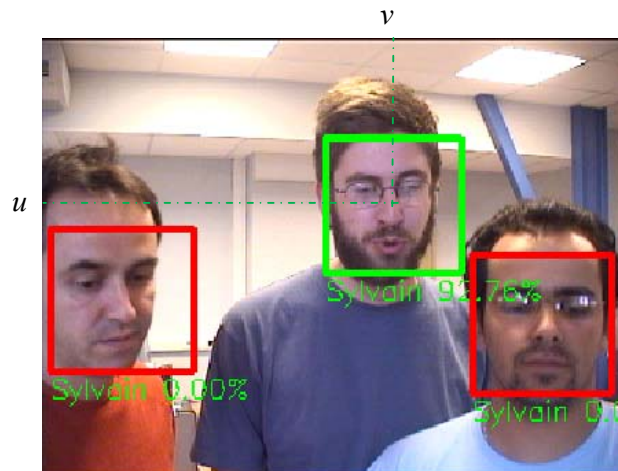


Fig. 3.4 : Détection et reconnaissance de visages

C. Les suivis :

Il existe deux principales modalités de suivi.

- a. Le suivi à mi-distance (de 1 à 3m), appelé « *Body Tracking* », est la modalité principale d'*ICU*. Elle permet de suivre dans le flot vidéo une personne grâce à un modèle de contour et de couleur décrits dans [6], [49] et [50]. L'initialisation se fait à l'aide principalement de la détection de visage appuyée par la reconnaissance. Il est alors possible de suivre une personne bien définie.

- b. Le suivi d'utilisateur lointain (distance > 3m), appelé « *Motion Monitoring* », sert de point d'entrée au mode de transition automatique. Le suivi est basé sur la détection de mouvement et permet de traquer des blobs en mouvement à de longues distances (jusqu'à 5m). De la même manière que pour le « *Body Tracking* », la prédiction utilise des mesures de couleurs et de mouvement.

Le module *ICU* utilise séquentiellement les composants cités ci-dessus dans son processus (voir figure 3.5). Les fonctions utilisées par plusieurs traqueurs en parallèle ne sont alors appelées qu'une seule fois. Toutes ces modalités sont accessibles de manière individuelle par le superviseur. Cependant le module inclut un mode automatique permettant le Switch entre les modalités jugées les plus pertinentes. Un automate à état fini gère les changements automatiques de modalité, du suivi à longue distance à l'interaction proximale [43].

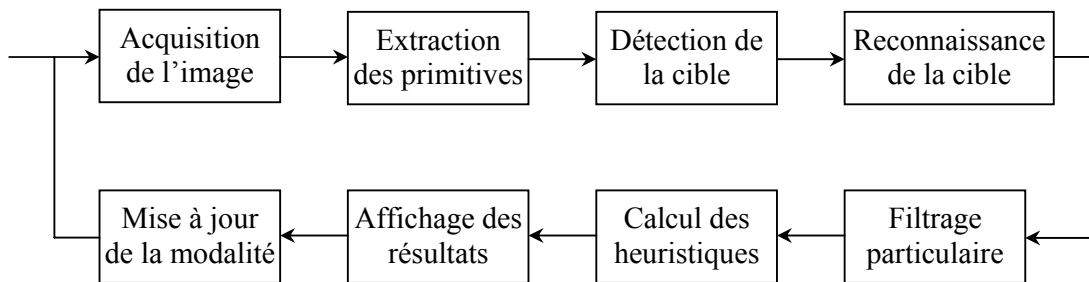


Fig. 3.5 : Séquence d'appel des composants d'*ICU*.

Enfin, et en vue de réaliser des tâches de navigation référencée vision, il est évident qu'on a besoin d'un outil performant de traitement d'image capable de nous fournir les informations nécessaires à l'implémentation de nos lois de commande. En effet, l'entrée de ces lois de commande n'est autre que les informations visuelles décrivant la position de la cible dans l'image. Pour notre application, et comme il a été mentionné avant, nous avons besoin d'au moins 4 points appartenant à la cible. De ce fait, nous avons opté pour l'utilisation du module *ICU* pour l'extraction de ces primitives. Ce module détecte tous les visages présents dans l'image, puis procède à leur identification. Le rectangle vert signifie que la personne suivie a bien été identifiée, tandis que les rectangles rouges pointent sur les visages non reconnus (voir figure 3.4). Ainsi, nous utilisons dans cette étude les coordonnées métriques des quatre coins du rectangle qui englobe le visage de la personne ciblée comme informations visuelles, notées *s*.

2.2 Modèle du système robotique

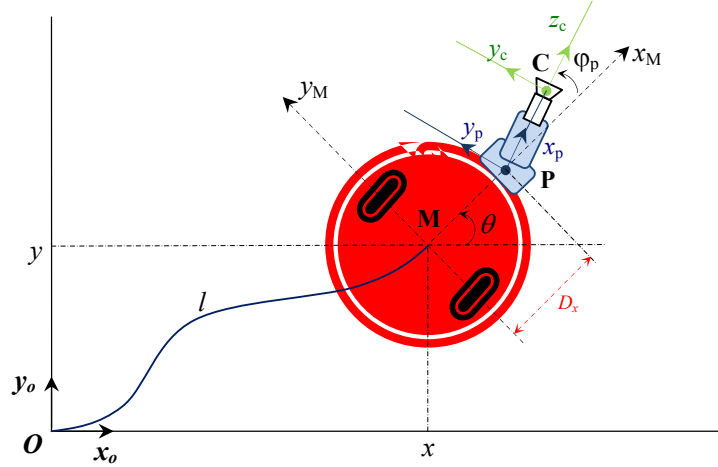


Fig. 3.6 : Repères du système Robot-Platine-Caméra

Le robot *Rackham* est une plateforme mobile de type « *iRobot B21r* » [52]. Afin de l'utiliser dans ce travail, il est d'abord nécessaire d'établir une modélisation rigoureuse de ce dernier. Pour cela, nous utilisons les notations suivantes [45]:

- $\mathcal{R}_0(\mathcal{O}, \vec{x}_o, \vec{y}_o, \vec{z}_o)$ est le repère lié à la scène,
- $\mathcal{R}_M(\mathcal{M}, \vec{x}_M, \vec{y}_M, \vec{z}_M)$ est le repère lié à la base mobile,
- $\mathcal{R}_P(\mathcal{P}, \vec{x}_P, \vec{y}_P, \vec{z}_P)$ est le repère lié à la PTU,
- $\mathcal{R}_C(\mathcal{C}, \vec{x}_C, \vec{y}_C, \vec{z}_C)$ est le repère lié à la caméra,
- D_x est la longueur du segment de droite $\{MP\}$.

La position de la base mobile est décrite dans le plan par le couple (x, y) désignant les coordonnées du centre M de la base mobile dans le repère global lié à la scène \mathcal{R}_0 (figure 3.6). Son orientation est définie par rapport à l'axe \vec{x}_o par l'angle θ . L'orientation horizontale de la platine (Pan) est donnée par l'angle entre \vec{x}_M et \vec{x}_P , noté ϕ_p , tandis que son orientation verticale (Tilt) noté ϕ_t est donnée par l'angle entre \vec{x}_P et \vec{z}_C . La position relative de la caméra dans le repère de la platine est décrite par le vecteur $\vec{PC} = (C_x, C_y, \theta)^T$ défini dans le repère \mathcal{R}_P [45]. Pour les besoins de cette étude, nous définissons le vecteur de configuration du robot, noté q , comme suit :

$$q = (l, \theta, \phi_p)^T \quad (3.11)$$

où l représente l'abscisse curviligne du point M sur la trajectoire du robot par rapport au repère de la scène R_θ . La dérivée de ce vecteur par rapport au temps n'est autre que le vecteur de commande, noté \dot{q} , et décrit par:

$$\dot{q} = \frac{dq}{dt} = (v_r, \omega_r, \omega_p)^T \quad (3.12)$$

avec v_r et ω_r qui désignent respectivement les vitesses linéaire et angulaire de la base mobile, alors que ω_p correspond à la vitesse de rotation horizontale de la platine par rapport au robot (rotation autour de l'axe \vec{z}_p).

L'ensemble de notre système robotique étant commandable en vitesse, nous utilisons alors tout au long de ce travail le modèle cinématique décrivant son comportement. Il s'agit donc de déterminer le torseur cinématique de la caméra, noté $\tau_{C/Ro}$, par rapport au repère de la scène R_o , en fonction des composantes du vecteur de commande \dot{q} de notre robot. Classiquement, ce torseur cinématique est constitué des composantes des vitesses de translation et de rotation. Il est défini alors comme suit :

$$\tau_{C/Ro} = (V_{C/Ro}^T, \Omega_{Rc/Ro}^T)^T \quad (3.13)$$

avec $V_{C/Ro} = (V_{Xc} \ V_{Yc} \ V_{Zc})^T$ et $\Omega_{Rc/Ro} = (\Omega_{Xc} \ \Omega_{Yc} \ \Omega_{Zc})^T$

La commande référencée vision consistant à définir le torseur cinématique de la caméra de manière à satisfaire un motif image prédéfini, le torseur $\tau_{C/Ro}$ est alors exprimé dans le repère caméra R_C [10]. Dans un premier temps, nous décrivons brièvement le modèle cinématique de la base mobile, muni d'une platine commandable en lacet. Il est donné classiquement par les relations suivantes [45]:

$$\begin{cases} \dot{x} = v_r \cos(\theta) \\ \dot{y} = v_r \sin(\theta) \\ \dot{\theta} = \omega_r \\ \dot{\varphi}_p = \omega_p \end{cases} \quad (3.14)$$

Le torseur cinématique τ_{M/R_0} de la base mobile, exprimé dans le repère de la scène R_0 s'écrit alors comme suit :

$$\tau_{M/R_0}^{R_0} = \begin{bmatrix} V_{M/R_0}^{R_0} \\ \Omega_{M/R_0}^{R_0} \end{bmatrix} = \begin{bmatrix} \{ v_r \cos(\theta) & v_r \sin(\theta) & 0 \}^T \\ \{ 0 & 0 & \omega_r \}^T \end{bmatrix} \quad (3.15)$$

La détermination du mouvement τ_{C/R_0} de la caméra par rapport au repère de la scène R_0 se déduit du résultat classique de mécanique [9] [45] :

$$\tau_{C/R_0}^{R_0} = \begin{bmatrix} V_{C/R_0}^{R_0} \\ \Omega_{C/R_0}^{R_0} \end{bmatrix} = \begin{bmatrix} V_{C/R_M}^{R_C} + V_{M/R_0}^{R_C} + \Omega_{R_M/R_0}^{R_C} \wedge MC^{R_C} \\ \Omega_{R_C/R_M}^{R_C} + \Omega_{R_M/R_0}^{R_C} \end{bmatrix} \quad (3.16)$$

Après développements et calculs, on arrive à écrire l'expression du torseur cinématique de la caméra, par rapport au repère de la scène R_0 sous forme matricielle, tel que :

$$\tau_{C/R_0}^{R_0} = \begin{bmatrix} V_{x_C} \\ V_{y_C} \\ V_{z_C} \\ \Omega_{x_C} \\ \Omega_{y_C} \\ \Omega_{z_C} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ -\sin(\varphi_p) & C_x + D_x \cos(\varphi_p) & C_x \\ \cos(\varphi_p) & -C_y + D_x \sin(\varphi_p) & -C_y \\ 0 & -1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} v_r \\ \omega_r \\ \omega_p \end{bmatrix} \quad (3.17)$$

Ce torseur s'écrit alors comme le produit d'une matrice dite « Jacobienne du robot » notée $J(q)$, liant le mouvement de la caméra aux vitesses de la base mobile et de la platine, avec \dot{q} le vecteur de commande du robot. Notons que cette jacobienne comporte trois lignes de zéro, correspondant aux mouvements non admissibles de la caméra. En effet, celle-ci ne peut se déplacer en translation sur l'axe \vec{x}_C , ni pivoter autour des axes \vec{y}_C et \vec{z}_C . Ces résultats sont donc conformes à la géométrie du robot [45]. Il est alors possible d'utiliser une forme réduite du torseur cinématique τ_r de la caméra, en ne considérant que les degrés de liberté commandables de cette dernière comme suit :

$$\tau_r = J_r(q) \times \dot{q} \quad (3.18)$$

$$\text{avec : } J_r(q) = \begin{bmatrix} -\sin(\varphi_p) & C_x + D_x \cos(\varphi_p) & C_x \\ \cos(\varphi_p) & -C_y + D_x \sin(\varphi_p) & -C_y \\ 0 & -1 & -1 \end{bmatrix}$$

La matrice $J_r(q)$ est appelée alors la jacobienne réduite du robot. Cette matrice est par définition régulière et inversible, car son déterminant est toujours non nul ($\det(J_r) = D_x \neq 0$). La contrainte de non-holonomie n'intervient donc plus dans le mouvement de la caméra [45].

En plus du modèle cinématique développé plus haut, nous avons besoin d'établir le lien entre le mouvement de la caméra, décrit par son torseur cinématique, et le mouvement des indices visuels dans l'image. Cela revient à modéliser l'interaction entre la caméra et l'environnement. C'est ce qui fait l'objet du paragraphe suivant :

2.3 Interaction caméra/environnement

Dans le cadre d'un asservissement visuel 2D, le choix des informations visuelles et l'obtention de la relation caractérisant leur variation sont deux points fondamentaux de cette approche [45]. Les informations visuelles $s(q,t)$ caractérisent les mesures effectuées au moyen de la caméra, en fonction de la configuration q du robot, et du temps t pris en tant que paramètre indépendant. On supposera dans l'ensemble de cette étude que la valeur d'une information visuelle $s(q,t)$ ne dépend que de la situation de la caméra par rapport à la cible observée. Ainsi, nous considérons que seul le mouvement de la caméra et éventuellement celui de la cible sont susceptibles de faire varier la valeur du signal sensoriel. En différentiant $s(q,t)$ par rapport au temps t , il est possible d'exprimer la variation des informations visuelles en fonction des mouvements de la caméra :

$$\dot{s}(q,t) = \frac{\partial s}{\partial q} \frac{dq}{dt} + \frac{\partial s}{\partial t} = \frac{\partial s}{\partial r} \frac{\partial r}{\partial q} \frac{dq}{dt} + \frac{\partial s}{\partial t} \quad (3.19)$$

Avec :

- $\frac{\partial s}{\partial r}$ est la matrice d'interaction ou jacobienne de l'image, notée $L_{(s,z)}$, que nous détaillons au paragraphe 2.3.1.

- $\frac{\partial r}{\partial q}$ est la jacobienne du robot, noté $\mathbf{J}(q)$, et qui ne dépend que de la géométrie du robot et de sa configuration q ,
- $\dot{q} = \frac{dq}{dt}$ est le vecteur de commande en vitesse de notre système robotique (voir section 2.2),
- $\frac{\partial s}{\partial t}$ est le terme qui est dû au mouvement propre de l'objet par rapport à la caméra.

Dans le cas où on considère un amer visuel immobile, ou encore si la cible est mobile mais nous ne disposons pas de moyens d'estimation efficaces et robustes pour ce terme, alors il est mis égal à zéro.

Ainsi, par le biais du torseur cinématique de la caméra $\mathcal{T}_{C/R_0}^{R_C}$, défini précédemment dans (3.17), la variation des informations sensorielles peut être exprimée par l'expression suivante :

$$\dot{s}(q,t) = L_{(s,z)} J(q) \dot{q} + \frac{\partial s}{\partial t} = L_{(s,z)} \mathcal{T}_{C/R_0}^{R_C} + \frac{\partial s}{\partial t} \quad (3.20)$$

2.3.1 Matrice d'interaction

La matrice $L_{(s,z)}$ représente le lien entre le mouvement de la caméra et l'évolution des indices visuels. Elle dépend non seulement de la nature des informations visuelles choisies, mais aussi de la situation de la caméra par rapport à l'objet observé, ou, plus précisément de la distance entre la caméra et la cible observée (appelée communément la profondeur). Différentes méthodes existent pour caractériser la matrice d'interaction. Ainsi, dans certains travaux, des méthodes d'apprentissage ont été utilisées pour l'estimation en ligne de cette matrice [66] [61]. Dans ces deux cas, les auteurs traitent des informations visuelles de type points, et c'est directement le produit des matrices $L_{(s,z)} J(q)$ qui est estimé. Toutefois, avec ces approches il n'est pas possible de démontrer la stabilité des lois de commande associées [13].

D'autres travaux ont adopté le choix de formuler une expression analytique de la matrice d'interaction, suivant le modèle géométrique de la cible et les primitives utilisées. Ainsi, *F. Chaumette* a proposé une méthode générale de calcul analytique de la matrice d'interaction pour différentes primitives géométriques simples, telles que des points, des

cercles, des sphères, des ellipses, etc [10], ou des moments de l'image [16]. Nous détaillons ci-après le cas de primitive de type point, et qui a été adopté pour cette étude :

2.3.2 Cas du point

Soit un point \mathbf{p} de coordonnées $(\mathbf{x}, \mathbf{y}, \mathbf{z})^T$ dans \mathbf{R}_C , et qui se projette par projection perspective sur le plan image en un point \mathbf{P} de coordonnées métriques $(\mathbf{X}, \mathbf{Y})^T$ (voir paragraphe 2.1.1). La matrice d'interaction $\mathbf{L}_{(p,z)}$ se déduit des équations du flot optique, et s'écrit comme suit [10] [45] :

$$\mathbf{L}_{(p,z)} = \begin{bmatrix} -\frac{f}{z} & 0 & \frac{X}{z} & \frac{XY}{f} & -(f + \frac{X^2}{f}) & Y \\ 0 & -\frac{f}{z} & \frac{Y}{z} & (f + \frac{Y^2}{f}) & -\frac{XY}{f} & -X \end{bmatrix} \quad (3.21)$$

où f désigne la distance focale de la caméra.

On note que le fait d'avoir une matrice d'interaction pour un point \mathbf{P} de dimension (2×6) suppose que le torseur cinématique est de dimension (6×1) , comme il est défini par la relation (3.17). Or, nous avons déjà considéré précédemment avec l'équation (3.18) un torseur cinématique réduit $\boldsymbol{\tau}_r$ de dimension (3×1) ne contenant que les degrés de liberté réellement commandables. De ce fait, il est nécessaire de ne conserver que les colonnes correspondant à ces degrés de liberté, afin de rendre cette matrice compatible. Nous obtenons alors la matrice d'interaction réduite \mathbf{L}_r suivante :

$$\mathbf{L}_r = \begin{bmatrix} 0 & \frac{X}{z} & \frac{XY}{f} \\ -\frac{f}{z} & \frac{Y}{z} & (f + \frac{Y^2}{f}) \end{bmatrix} \quad (3.22)$$

Il est possible dès lors de généraliser ce résultat dans le cas d'un amer visuel constitué de k points. Le vecteur des informations sensorielles \mathbf{s} est alors défini par : $\mathbf{s} = (\mathbf{X}_1, \mathbf{Y}_1, \dots, \mathbf{X}_k, \mathbf{Y}_k)^T$, où chaque couple $(\mathbf{X}_j, \mathbf{Y}_j)$ représente les coordonnées métriques du point \mathbf{P}_j de la cible, projeté dans l'image. La matrice d'interaction globale correspondant à la cible observée

est constituée alors par la superposition des matrices de tous les points de la cible considérée, ce qui donne:

$$L_{(s, z)} = \begin{bmatrix} L_{(P_1, z_1)} \\ \vdots \\ L_{(P_k, z_k)} \end{bmatrix} \quad (3.23)$$

où chaque ligne $L_{(P_j, z_j)}$ correspond à la matrice d'interaction d'un point P_j défini par la relation (3.22). Enfin, différents types de primitives peuvent être construits sur la base de points : centres de gravité, segments, polygones, cercles, etc [45]. Ces résultats sont exploités ultérieurement dans la phase de synthèse d'un asservissement visuel 2D sur notre robot mobile.

2.4 Le système RFID

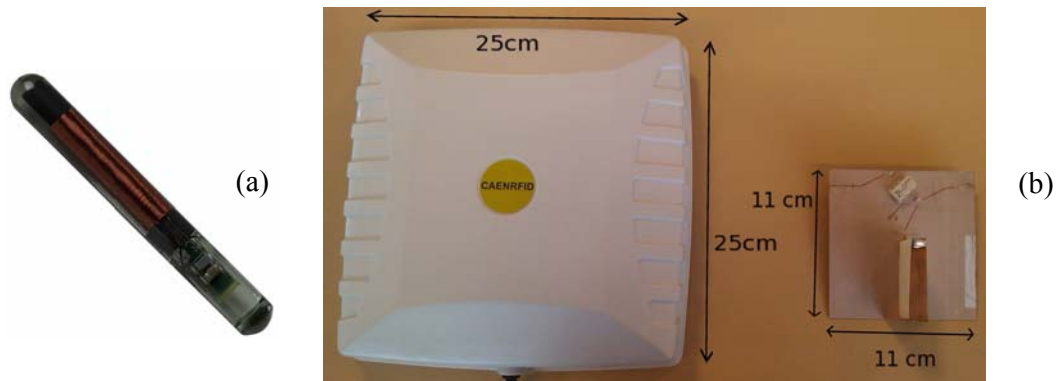


Fig. 3.7 : Composants d'un système RFID (a) Un Tag, (b) Deux modèles d'antennes

Ce type de système de détection et d'identification rentre dans la catégorie dite « technologies émergentes » dans le domaine de la robotique. Il est composé principalement de deux parties : Une partie active qui se compose généralement d'une ou de plusieurs antennes RFID comme celles montrées sur la figure 3.7.b, connectées à un lecteur. La deuxième partie qui est passive est composée d'un badge RFID passif (appelé aussi Tag, figure 3.7.a). L'antenne émet des ondes radio d'une certaine puissance sur un périmètre constant. Le Tag est équipé d'une électronique dédiée qui lui permet d'être détecté par le lecteur via une ou plusieurs antennes à la fois. La puissance des ondes radio émises permet au

Tag ensuite d'envoyer une réponse contenant un identifiant unique capté par le lecteur au travers des antennes [52]. L'utilisation d'un Tag passif dans ce type de système est très large, du fait que contrairement aux Tags actifs, il est très peu coûteux et offre une durée de vie illimitée du fait qu'il ne nécessite pas de batterie embarquée. De plus, il délivre une information explicite sur l'identité de la personne qui le porte même si l'information concernant sa position est très vague. Il est tout de même possible de détecter et d'identifier plusieurs Tags à la fois.

Dans le cadre de cette étude, le système RFID d'identification de personnes est composé de huit antennes, un Tag passif, un lecteur à quatre entrées et un multiplexeur 4/8 comme il est montré dans la figure 3.8. Le robot *Rackham* étant équipé d'un système de vision (*voir section 2.1*), la précision et la richesse des informations de la vision couleur active ont été combinées avec l'identification RFID, afin d'améliorer les performances de la détection de l'utilisateur dans un milieu d'intérieur encombré. A notre connaissance, ce type d'approche a été proposé dans la littérature pour la première fois dans [51].

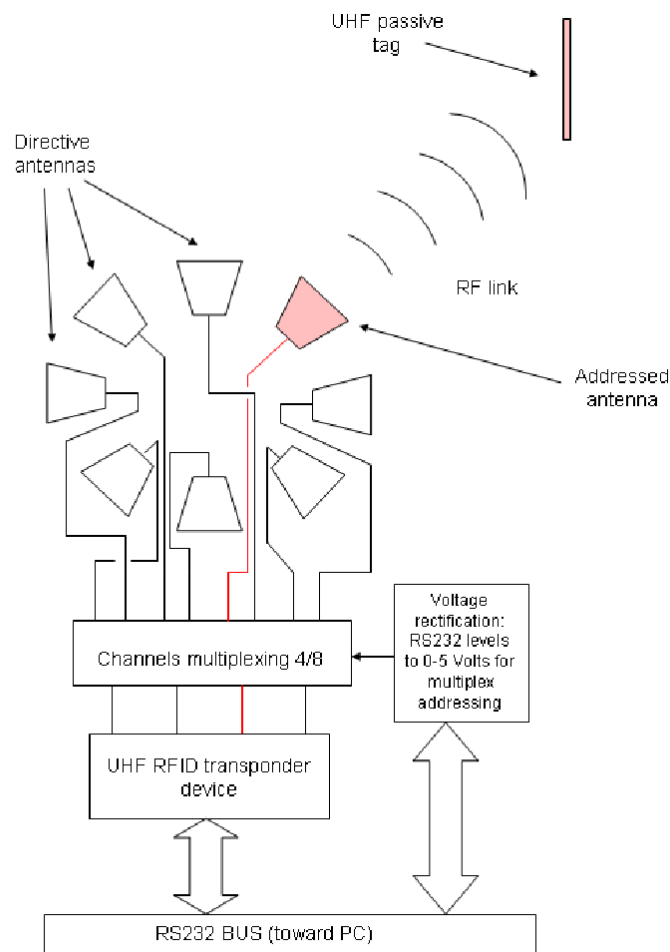


Fig. 3.8 : Prototype de multiplexage RF pour 8 antennes.

2.4.1 Description du capteur

Dans le cadre du projet *Commrob* [25], un système expérimental RFID a été développé par *T. Germa* [53], afin de réaliser une étude de faisabilité sur la détection de personnes sur le robot *Rackham*. Le capteur se compose de [52] :

- Un lecteur RFID multiprotocoles de marque « CAENRFID A941 » fonctionnant à 870MHz,
- Huit antennes RFID directionnelles capables de détecter un badge passif porté par l'utilisateur,
- Un prototype de carte de multiplexage RF afin d'adresser les 8 antennes de manière séquentielle (figure 3.8).

Ce système permet de discrétiser l'espace autour du robot en 24 zones, dépendant uniquement du nombre d'antennes qui détecte simultanément le même badge (voir figure 3.9). Un même badge peut donc être détecté tout autour du robot dans un périmètre allant de 0.5m (i.e. diamètre approximatif du robot) à 4.5m (i.e. portée maximale de l'antenne). Afin de caractériser le modèle d'observation de la ceinture d'antennes, une étude expérimentale détaillée a été réalisée dans le cadre de la thèse de *T. Germa* [53]. Dans cette étude, le modèle d'observation du système RFID complet (8 antennes à la fois) a été déterminé sur la base des histogrammes normalisés associés aux trois cas possibles de détection d'un Tag, à savoir par une seule antenne, par deux antennes en même temps ou par trois antennes à la fois (voir figure 3.9). Le modèle du capteur résultant de cette étude a permis d'approximer les histogrammes en distance et en azimut par un modèle gaussien. A partir de ce modèle, une étude de faisabilité, décrite au paragraphe suivant, a été effectuée pour caractériser les performances globales du capteur.

2.4.2 Évaluations de la faisabilité

Le système RFID a été monté sur le robot *Rackham* puis évalué dans un contexte encombré [53]. Il a été procédé à l'évaluation du nombre de vrais positifs (détections effectives) dans une zone de 81m^2 autour du robot. Des obstacles ont été ajoutés aléatoirement selon une loi uniforme dans la zone de détection des antennes de façon incrémentale. La vérité terrain est basée sur le rapport entre les zones occultées par les obstacles et la surface totale de la zone de détection. Par la suite, cette situation a été reproduite sur le terrain en observant les résultats de détection d'un badge en fonction du nombre d'obstacles.

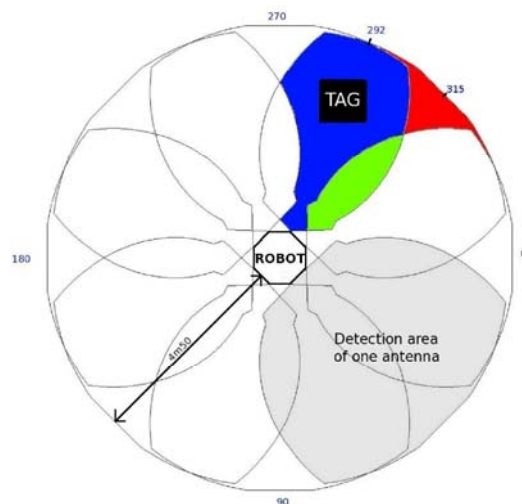


Fig. 3.9 : Champ de détection azimutal avec 8 antennes RFID

Les résultats expérimentaux obtenus ont été très proches des résultats théoriques [53]. La distribution du système étant régie par les occultations, il a été observé davantage de faux rejets lorsque le nombre d'occultations augmente. Malgré cela, le taux de vraies détections reste satisfaisant, même lors de forts encombrements (70% en moyenne pour 7 personnes présentes autour du robot). De plus, très peu de fausses détections ont été observées (réflexions, détections sur la mauvaise antenne, etc) dans la pratique [52] [53].

2.5 Environnement logiciel du robot

Après avoir exposé les aspects liés au matériel (actionneurs et capteurs) embarqué sur notre plateforme robotique, ainsi que certains aspects logiciels (détecteur de visage, détecteur RFID), nous présentons dans ce paragraphe l'architecture logicielle de contrôle du système en question. Il s'agit d'une structure modulaire sous l'environnement de programmation temps réel **Genom** (GENerator Of Modules), conçu et développé au *LAAS-CNRS* dans les années 90s [1] [43].

2.5.1 Présentation de Genom [43]

Le générateur de modules **Genom** est un outil d'aide à la conception d'architectures logicielles temps-réel. Il permet d'intégrer aisément et rapidement des fonctions opératoires dans des modules indépendants communicants chargés de leur exécution. Les fonctions peuvent être dynamiquement démarrées, interrompues ou (re)paramétrées en adressant aux modules des requêtes asynchrones paramétrées et non-bloquantes.

Les modules sont ainsi des serveurs standardisés qui prennent entièrement en charge l'exécution d'un ensemble de fonctions ou algorithmes sur une machine hôte. Il peut héberger plusieurs fonctions, synchrones ou asynchrones, et exécuter plusieurs traitements en parallèle. La stratégie de regroupement des fonctions dans un même module est généralement guidée par la notion de ressource : un module est responsable d'une ressource (physique ou logique), et intègre l'ensemble des fonctions nécessaires à son contrôle (les différents modes de traitements, les procédures d'interruption, de détection/reprise d'erreur, etc.). On peut, par exemple, concevoir un module pour gérer une caméra (production d'images, contrôle de paramètres), un actionneur (avec différents modes d'asservissement), un GPS ou encore pour maintenir des données telles que : une carte, une trajectoire, etc.

Plusieurs entités peuvent adresser des requêtes à un module : un opérateur, un autre module, ou encore un superviseur centralisé des modules. Lorsqu'un traitement se termine, que ce soit d'une façon nominale ou suite à une interruption ou à un échec, le module retourne au client qui lui a adressé la requête une réponse asynchrone, ou réplique. Un bilan d'exécution associé à cette réplique qualifie la façon dont s'est déroulé le traitement. Des données peuvent être produites ou récupérées par le traitement durant son exécution.

Les modules sont extrêmement simples à réaliser. Ils sont en effet automatiquement produits par **Genom** à partir d'une description synthétique qui ne nécessite pas de connaissance particulière sur les systèmes distribués temps-réel ou sur le système d'exploitation employé pour la machine cible. Cette description contient essentiellement la liste des services offerts par le module (la liste des requêtes que l'on peut adresser au module) et leurs caractéristiques : paramètres, fonctions mises en œuvre selon l'état du traitement, résultats qualitatifs possibles, données produites ou attendues, caractéristiques temporelles (dont la période pour les traitements périodiques) et logiques (services compatibles ou exclusifs), etc.

A partir de la description d'un module, **Genom** produit :

- Un module exécutable sous différents systèmes d'exploitation (actuellement VxWorks et UNIX),
- Les bibliothèques d'interface pour les clients (accès aux services et aux données produites),
- Un programme interactif qui permet de procéder aux tests préliminaires du module.

La synthèse automatique des modules s'appuie sur un modèle générique de module conforme aux pré-requis pour les systèmes temps-réel embarqués (exécution en temps constant, pas d'allocation dynamique de mémoire, traitements critiques non-préemptifs, etc.) et largement validé. Outre une facilité de développement (la description d'un module est indépendante du système d'exploitation et ne nécessite pas de connaissance particulière sur les systèmes distribués temps-réel), la génération automatique garantit l'adéquation de chaque module avec le modèle générique commun unique, ce qui est primordial pour la maîtrise de systèmes de dimensions importantes. En particulier, les interfaces sont standardisées et les tests d'intégration deviennent inutiles. La standardisation de la structure, du comportement et des interactions des modules permet d'automatiser les procédures d'intégration des modules dans le système. En particulier, **Genom** peut produire les bibliothèques d'interface aux modules dans n'importe quel langage informatique (C/C++, TCL, PROPICE, etc.). On peut de la même façon produire automatiquement des procédures de tests logiques ou temporels pour le système complet.

La structuration en modules permet :

- de masquer les spécificités algorithmiques dans chaque module,
- de découpler les développements (le module peut être produit avant même de disposer des algorithmes internes),
- de cloisonner les contraintes temporelles (spécifiées et gérées module par module),
- de délocaliser le contrôle de chaque ressource (le module peut gérer différents modes, y compris des procédures de reprise d'erreur locales),
- de hiérarchiser le système (hiérarchies de modules).

L'ensemble offre un système ouvert, communiquant et contrôlable :

- Requêtes et répliques composent le flux de contrôle. Elles permettent le suivi et le contrôle de l'ensemble des fonctions qui s'exécutent à bord du système. Les répliques sont directement adressées à l'émetteur de la requête. Les requêtes peuvent être émises par :
 - un ou plusieurs autres modules (hiérarchies de modules),
 - un opérateur (généralement pour la mise au point du module),
 - un système de contrôle d'exécution centralisé.

- Le flux de données entre les composants de l'architecture s'appuie sur un second protocole: les données produites sont exportées dans des structures ouvertement accessibles en lecture.

2.5.2 Contexte d'utilisation

Genom s'adresse en particulier aux systèmes embarqués complexes, tels que les robots mobiles autonomes ou les satellites, qui sont particulièrement exigeants sur le plan de l'informatique temps-réel distribuée car ils impliquent :

- La cohabitation de fonctions opératoires hétérogènes, tant par leur complexité algorithmique que par leur contraintes temps-réel (commande de capteurs et d'actionneurs, asservissements, surveillances, traitements et analyses d'images, calcul de trajectoires, etc.),
- L'intégration homogène des fonctions dans l'architecture de contrôle, qui suppose des comportements cohérents et prédictibles (démarrage, interruption, reprise d'erreur), et des modes d'interactions standards (contrôle, paramétrisation, transfert de données),
- La parallélisation, la distribution physique et l'embarquabilité des traitements,
- L'insertion, la mise à jour ou la réutilisation aisée des fonctions par des non spécialistes.

Genom permet de procéder à des prototypages rapides de systèmes complexes : dès que le système est spécifié et structuré en sous-systèmes (modules), la description des modules et donc leur synthèse est une opération quasi-immédiate. Les algorithmes internes à chaque module peuvent alors être affinés de manière incrémentale, et surtout sans interaction avec le reste du système. Le fichier de description de chaque module est un document de travail synthétique et complet qui décrit les services offerts, les données attendues et les données produites par le module. Les modules produits peuvent être évalués sur des stations de travail ou directement embarqués sur la machine cible temps-réel. Les modules respectent les contraintes propres aux systèmes temps-réel et ont été largement validés dans de nombreuses applications robotiques. **Genom** propose en outre des moyens pour évaluer in-situ les caractéristiques temporelles du système.

5.2.3 Architecture logicielle embarquée

Pour notre application, différents modules logiciels sont employés afin de réaliser la tâche désirée, à savoir les modules de perception comme *Caméra*, *ICU* et *RFID*, et ceux de

l'action comme *Platine*, *Rflex* et *VisuServ*. Ci-après une brève description des fonctionnalités de ces modules :

- *Caméra* : c'est le module qui gère le capteur de vision (résolution, paramètres, etc.), et fournit aux autres modules les images numériques acquises,
- *RFID* : c'est le module qui gère le système de détection RFID, et fournit aux autres modules les données relatives à la position de la cible dans le repère du robot R_M en coordonnées polaires, notées (d_{lug}, θ_{lug}) ,
- *ICU* : c'est le module qui réalise la détection et l'identification du visage de l'utilisateur ciblé (voir paragraphe 2.1.3),
- *Platine* : c'est le module qui gère la PTU et permet de la faire bouger en contrôlant soit en vitesse ou en position les deux moteurs (pan et tilt). Cela permet alors de changer l'orientation du capteur monté sur la PTU (ex Caméras, Swiss Ranger, etc.),
- *Rflex* : c'est le module qui s'interface avec les logiciels fournis par le constructeur du robot (*iRobot*), et qui exporte dans un poster la position odométrique du robot, corrigée par la mesure du gyroscope. Cette position fournit une bonne estimée des mouvements du robot. De plus, il importe les commandes (vitesse de rotation et de translation de la base mobile), et les envoie aux actionneurs pour exécution,
- *VisuServ* : c'est le module central de notre application. Comme son nom l'indique, il permet d'implémenter l'asservissement visuel sur le robot. Pour cela, et comme indiqué sur le schéma de la figure 3.10, ce module envoie les commandes aux modules des actionneurs *Platine* et *Rflex*, sur la base des données de perception reçues du module *ICU*.

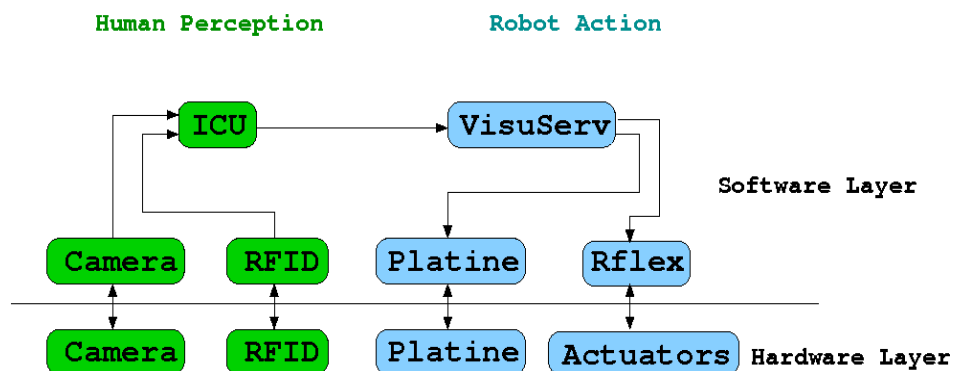


Fig. 3.10 : Architecture logicielle de l'application

La fusion des données issues des modules *Caméra* et *RFID* est réalisée au sein du module *ICU*. Les déplacements de la caméra (*Platine*) et du robot (*Rflex*) sont contrôlés par

le module *VisuServ*, que nous avons développé. Ce dernier implémente deux types de contrôleur (classique et modes glissants) utilisant les indices visuels lorsqu'ils sont disponibles, et les informations RFID dans les autres cas. Ces modules ont été implémentés au sein de l'architecture *LAAS-CNRS* [1], et s'exécutent simultanément à une cadence de 6Hz.

3. IMPLEMENTATION D'UN ASSERVISSEMENT VISUEL 2D

Dans cette partie, l'objectif est de synthétiser un asservissement visuel 2D permettant au robot mobile de se positionner de telle sorte que le visage de l'utilisateur statique ciblé (personne connue) atteigne une configuration désirée dans l'image de la caméra. Usuellement, la configuration désirée doit satisfaire deux contraintes exprimées dans l'environnement du robot comme suit:

- Avoir le visage centré dans l'image (le centre de gravité du visage se confond avec le centre optique)
- Etre positionné à une distance nominale par rapport au robot (le rectangle englobant le visage est de taille égale à celle de la configuration désirée).

Comme il a été précisé au chapitre précédent, le principe de l'asservissement visuel 2D (*voir paragraphe 2.2.2*) consiste à contrôler le mouvement de la caméra de manière à annuler l'erreur entre les informations visuelles courantes $s(q, t)$ et la configuration désirée s^* . De ce fait, pour la synthèse du contrôleur, nous nous basons sur le formalisme des fonctions de tâche détaillé à la section 2.1 du chapitre 2. Classiquement, la convergence de la fonction de tâche vers zéro est effectuée en lui imposant de décroître de manière exponentielle [45]. Dans la mesure où notre système robotique est commandable en vitesse, par l'intermédiaire de \dot{q} définie dans (3.12), nous considérons uniquement des schémas de commande cinématique tout au long de cette étude. Le comportement désiré du système en boucle fermée peut donc être décrit par :

$$\dot{e}(q) = J_e \dot{q} = -\lambda e(q) \quad (3.24)$$

où λ (le gain) est un scalaire positif ou une matrice définie positive, permettant de fixer la vitesse de décroissance exponentielle de $e(q)$. De plus, J_e qui est égale à $\frac{\partial e}{\partial q}$ est une matrice

représentant la jacobienne de la fonction de tâche. On rappelle que si $e(q)$ est ρ -admissible (i.e. J_e est inversible), cela suffit pour s'assurer que le problème de commande est bien posé, et assure la détermination de la commande \dot{q} de manière unique. Toutefois, lorsque la fonction de tâche $e(q)$ est définie par l'expression (2.1), le calcul de J_e s'avère en général difficile, et la synthèse de la commande requiert une estimation de cette matrice, notée \hat{J}_e [10] [45]. La loi de commande (3.24) s'écrit alors sous la forme :

$$\dot{q} = -\hat{J}_e^{-1} \lambda e(q) \quad (3.25)$$

La dynamique de la fonction de tâche $e(q)$ devient alors en réalité dépendante du produit de Matrices $(\hat{J}_e J_e^{-1})$. En effet, la stabilité de la loi de commande nécessite que ce produit soit égal à une matrice définie positive. Or, si les conditions mentionnées dans (2.2) (voir section 2.1 du chapitre 2) sont satisfaites, la jacobienne J_e est définie positive autour de la trajectoire solution $q_r(t)$. Dans ce cas, une solution satisfaisant la condition $(\hat{J}_e J_e^{-1}) > 0$ consiste à choisir \hat{J}_e égale à la matrice identité I , et la loi de commande ci-dessus se simplifie alors comme suit [38] :

$$\dot{q} = -\lambda e(q) \quad (3.26)$$

3.1 Synthèse du contrôleur classique

La synthèse d'un contrôleur permettant à un robot de réaliser une navigation référencée vision, en se basant sur le formalisme des fonctions de tâches passe nécessairement par la spécification de l'objectif à réaliser sous la forme d'une régulation à zéro d'une fonction particulière $e(q)$, dite *fonction de tâche*. Dans cette étude, nous avons opté pour un choix classique de la fonction de tâche pour un asservissement visuel 2D permettant d'amener les indices visuels courants s vers la configuration désirée s^* , comme suit :

$$e(q) = C (s(q) - s^*) \quad (3.27)$$

où C représente la matrice de combinaison qui permet de prendre en compte un nombre d'indices visuels supérieur au nombre de degrés de liberté de la caméra. Toutefois, l'introduction de cette matrice peut perturber la convergence des indices visuels, en introduisant des minimas locaux. En effet, la régulation de $e(q)$ vers zéro n'assure que la convergence de Cs vers Cs^* , ce qui ne garantit pas forcément la convergence de s vers s^* [45].

Comme il a été cité plus haut, la convergence de la fonction de tâche vers zéro est réalisée en lui imposant de décroître de manière exponentielle. Le comportement désiré en boucle fermée peut donc être décrit par :

$$\dot{e}(q) = -\lambda C (s(q) - s^*) \quad (3.28)$$

Le gain λ dans ce cas est une matrice diagonale (3x3) définie positive comme suit : $\lambda = \text{Diag}(\lambda_1, \lambda_2, \lambda_3)$. Les coefficients de la diagonale permettent de régler la vitesse de convergence. En utilisant l'expression (3.20), prise dans le cas d'une cible statique ou très peu mobile (cas de l'homme), on peut écrire :

$$\dot{e}(q) = C \dot{s}(q) = C L_{(s,z)} J(q) \dot{q} \quad (3.29)$$

Dans cette étude, nous considérons un amer visuel caractérisé par un ensemble de 4 points (coins du rectangle englobant le visage de l'utilisateur ciblé), extrait par le processus de traitement d'image *ICU*. Le vecteur des informations sensorielles s est alors défini par : $s = (X_1, Y_1, \dots, X_4, Y_4)$, où chaque couple (X_i, Y_i) représente les coordonnées métriques du $i^{\text{ème}}$ point de l'amer projeté en P_i dans le plan image de la caméra. Comme nous l'avons vu précédemment, la matrice d'interaction réduite résultante est donnée par :

$$L_r = \{ L_{P1}^T, \dots, L_{P4}^T \}^T \quad (3.30)$$

De même, on considère la jacobienne réduite du robot, notée J_r , afin de garantir une compatibilité des dimensions dans le produit de l'équation (3.29) qui s'écrit alors sous la forme :

$$C \dot{s}(q) = C L_r J_r \dot{q} \quad (3.31)$$

La condition de décroissance exponentielle de la fonction de tâche peut s'écrire alors comme suit:

$$C L_r J_r \dot{q} = -\lambda C (s(q) - s^*) \quad (3.32)$$

Comme nous l'avons cité au paragraphe 2.1.1 du chapitre 2, il est nécessaire de garantir la satisfaction de la condition de *ρ -admissibilité* de la fonction de tâche $e(q)$, qui est associée à la régularité de sa jacobienne. Cette condition est satisfaite en choisissant la matrice de combinaison C de rang plein n , et telle que le produit des matrices $(C L_r J_r)$ soit aussi de rang plein n . Dans la mesure où la matrice jacobienne du robot J_r est une matrice régulière (voir section 2.2), la stabilité et la convergence de la fonction de tâche sont assurées par la condition suffisante suivante [13].

$$C L_r > 0 \quad (3.33)$$

Afin de garantir la positivité de ce produit matriciel, une solution judicieuse consiste à fixer la matrice de combinaison C égale à une approximation de la pseudo-inverse à gauche de la matrice d'interaction, notée \hat{L}_r^+ , qui est donnée par le produit de matrices $\{(L_r)^T \cdot L_r\}^{-1} (L_r)^T$ [45]. Dans ce cas, on obtient :

$$J_r \dot{q} = -\lambda \hat{L}_r^+ (s(q) - s^*) \quad (3.34)$$

Finalement, le correcteur par asservissement visuel 2D assurant la régulation à zéro de la fonction de tâche se déduit alors facilement de l'expression précédente :

$$\dot{q} = -J_r^{-1} \lambda \hat{L}_r^+ (s(q) - s^*) \quad (3.35)$$

Comme cette approche se base sur l'estimation de la pseudo-inverse de L_r , la propriété de décroissance exponentielle peut être perdue puisque le comportement du système en boucle fermée est le suivant [45] :

$$\dot{e}(q) = -(\hat{L}_r^+ L_r) \lambda e(q) \quad (3.36)$$

En effet, le produit $(\hat{L}_r^+ L_r)$ peut être différent de la matrice identité I , car il dépend de l'estimation de la valeur de la profondeur z nécessaire pour le calcul de L_r . Usuellement, on palie à ce problème en faisant le choix d'utiliser une matrice d'interaction constante, calculée hors ligne à l'aide des indices visuels et profondeur de la configuration désirée (voir paragraphe 2.2.2 du chapitre 2). Dans ce cas, il est difficile de garantir la stabilité du système lorsque le robot se trouve très loin de la cible, c'est-à-dire lorsque le signal s se trouve loin du motif désiré s^* . Cependant, l'expérience montre que le domaine de convergence est en général plus étendu [45].

3.2 Simulations

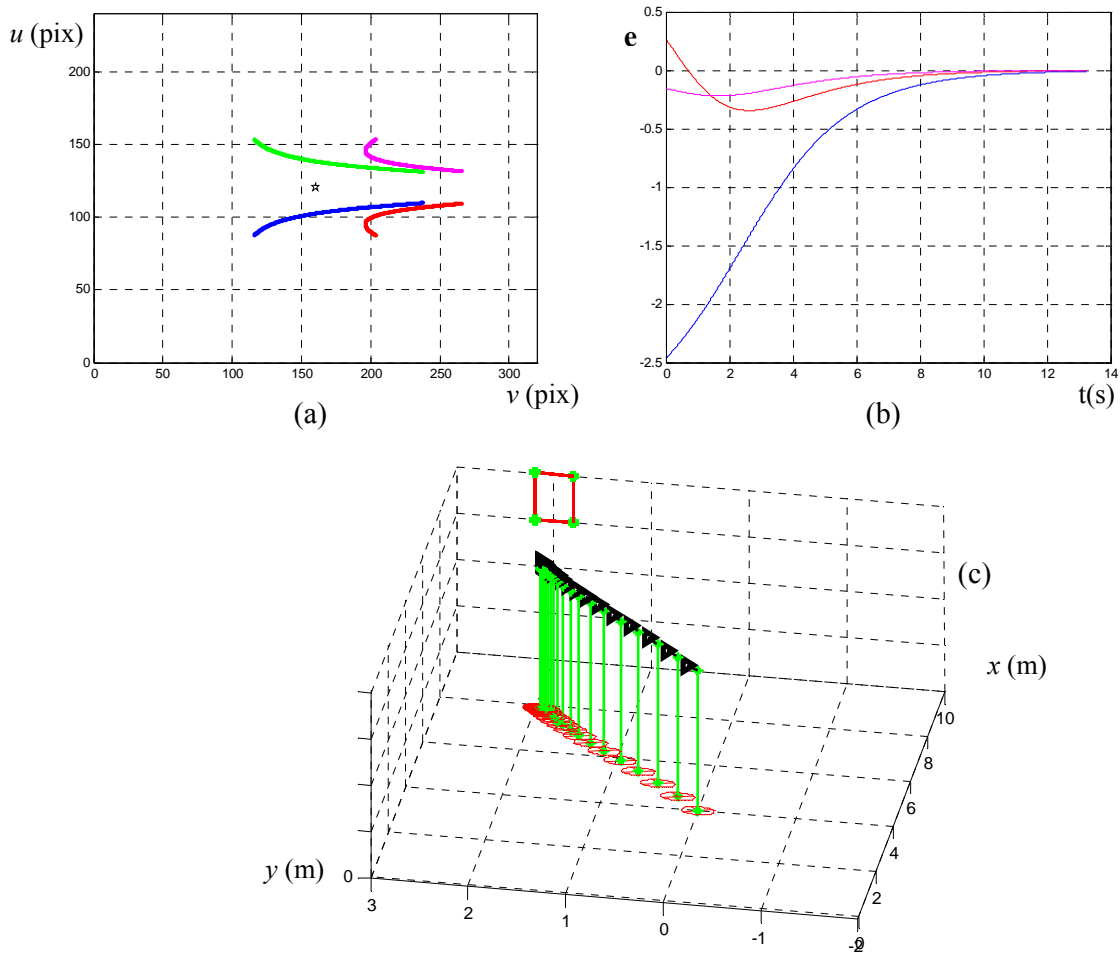


Fig. 3.11 : Test 1 de simulation d'un asservissement visuel 2D : (a) Trajectoires des indices visuels, (b) Evolution de la fonction de tâche, (c) Trajectoire du robot

Nous avons procédé à une série de tests de l'asservissement visuel 2D en simulation sous l'environnement *Matlab*, afin de mesurer les performances et limitations de cette approche. Nous présentons ci-après des exemples de simulation, où la tâche référencée vision choisie consiste à positionner le robot relativement à un utilisateur statique caractérisé par le rectangle qui englobe son visage (voir figure 3.11.c). On note qu'il est nécessaire que la configuration initiale du robot, notée $q_0(x_0, y_0, \theta_0, \varphi_{p0})$ soit choisie telle que la cible soit initialement présente dans le champ de vision de la caméra.

Nous présentons deux tests de simulation, où la cible (rectangle rouge à coins verts) est toujours positionnée à un point de l'environnement ayant comme coordonnée ($x = 10\text{m}$ et $y = 2\text{m}$) dans le repère global R_o lié à la scène (voir section 2.2). La figure 3.11.a montre les trajectoires dans l'image des quatre points constituant la cible pour le premier test, dans lequel le robot est positionné initialement à la configuration (4m, 0m, $\pi/6$ rad, 0 rad). Le centre de l'image étant désigné par une étoile noire, ces trajectoires montrent bien que la cible est centrée à la fin de la tâche. La trajectoire du robot présentée sur la figure 3.11.c confirme cela, et montre que la tâche de navigation est parfaitement réalisée.

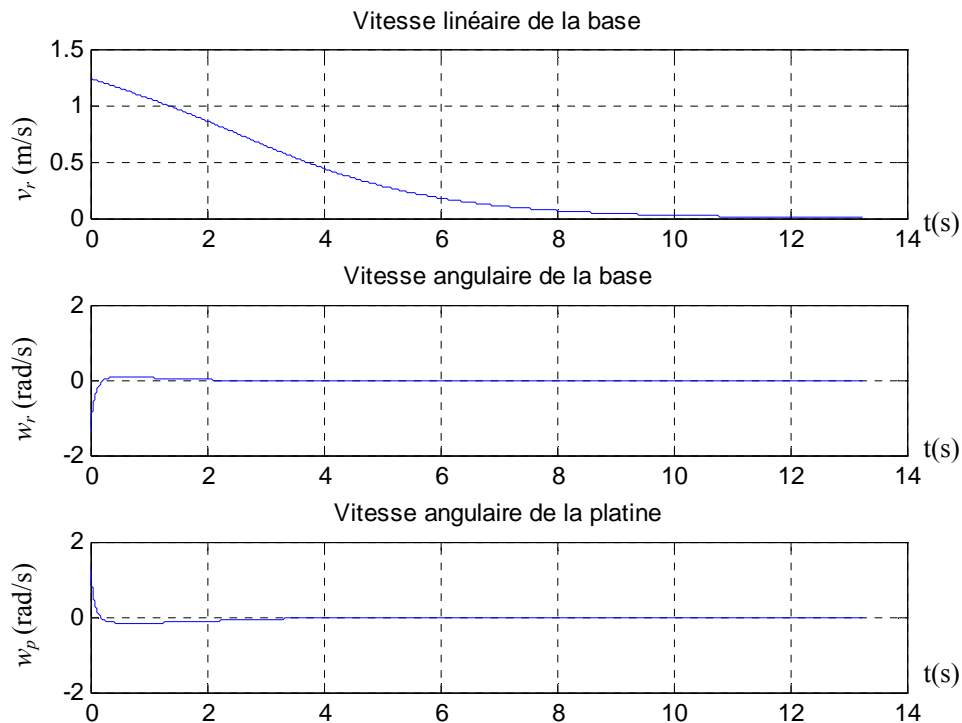


Fig. 3.12 : Variables de contrôle de l'asservissement visuel 2D pour le test 1

De même pour l'évolution des trois composantes de la fonction de tâche e qui convergent vers zéro de manière quasi simultanée (voir figure 3.11.b). Toutefois, ces

convergences ne sont pas toujours effectuées de manière exponentielle, comme il a été précisé à la section précédente. D'un point de vue commande, on présente dans la figure ci-dessus les composantes du vecteur de commande envoyées au robot pour effectuer cette tâche de navigation référencée vision. On remarque que pour les trois composantes, leurs amplitudes sont acceptables (inférieures aux valeurs de saturation des actionneurs). Cela est assuré par le biais d'un choix adéquat des valeurs de la diagonale de la matrice des gains λ . Enfin, les valeurs des commandes convergent aussi vers zéro, ce qui implique la convergence du système et la fin de la tâche.

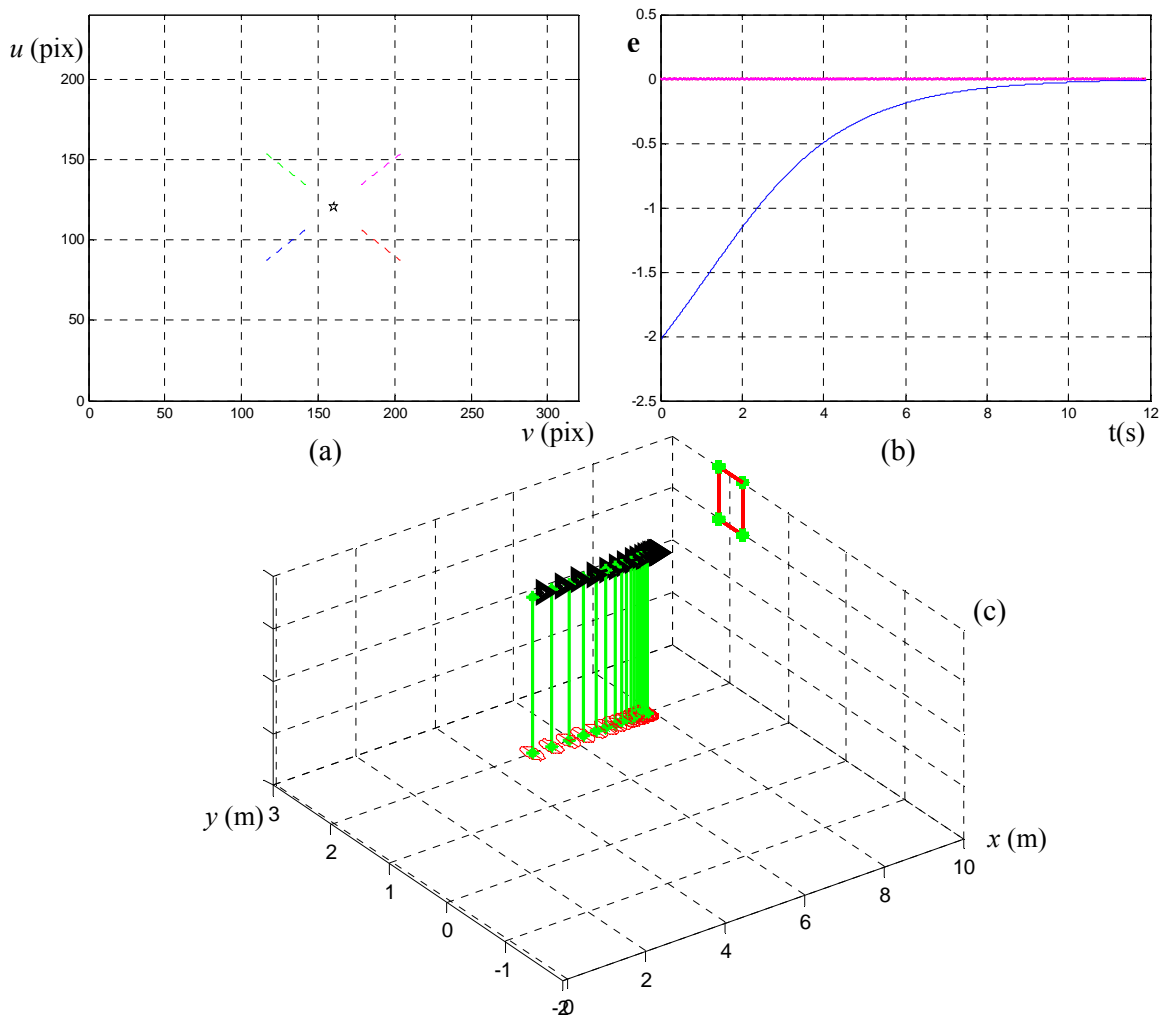


Fig. 3.13 : Test 2 de simulation d'un asservissement visuel 2D : (a) Trajectoires des indices visuels, (b) Evolution de la fonction de tâche, (c) Trajectoire du robot

Pour le deuxième test, la configuration initiale du robot est q (5m, 2m, 0 rad, 0 rad) où le robot est en face de la cible. De ce fait, le rectangle représentant la cible est alors déjà

centré dans l'image comme le montre la figure 3.13.a. Tout de même, le robot avance vers la cible jusqu'à ce que le rectangle ait les dimensions correspondantes à la configuration désirée. Les composantes de la fonction de tâche convergent alors vers zéro comme lors du test précédent, ce qui montre la bonne exécution de la tâche. Cela est confirmé par le tracé de la trajectoire du robot (voir figure 3.13.c)

De même que pour le test 1, les variables de commande demeurent bornées et assurent une exécution normale de la tâche, sans saturations ni oscillations. Elles tendent vers zéro à la fin de la tâche, ce qui signifie la convergence du système et la fin de la tâche. On remarque les très faibles valeurs des vitesses de rotation, à savoir ω_r et ω_p , qui sont évidemment dues au fait que le robot effectue un mouvement de translation pure, du moment que la cible est initialement centrée dans l'image.

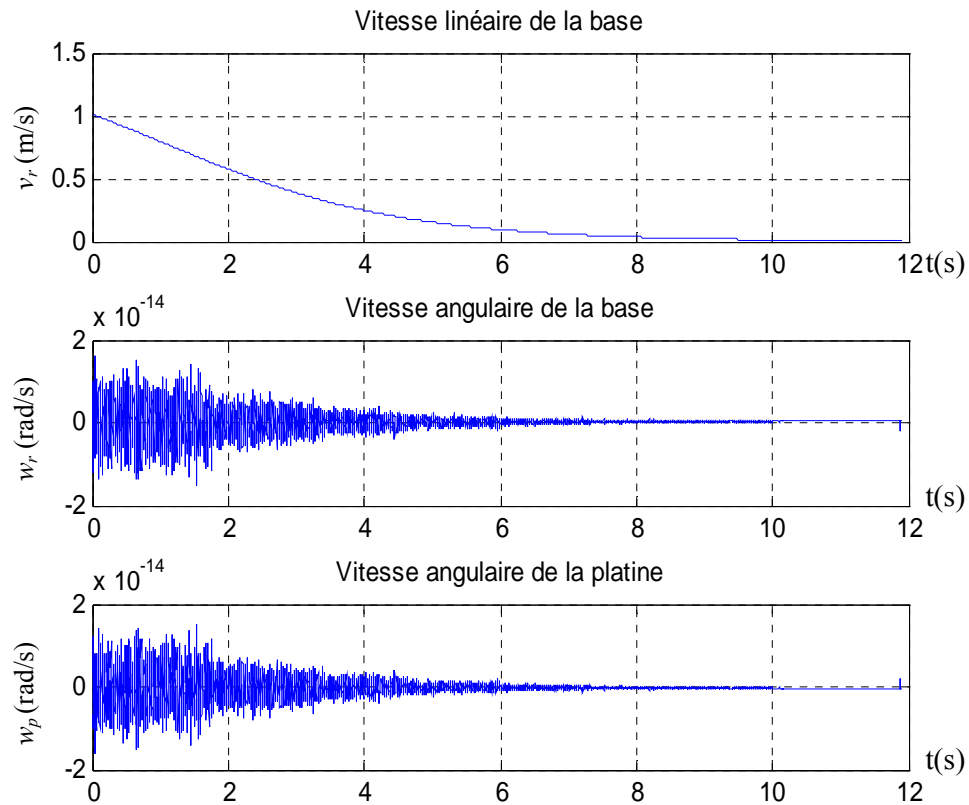


Fig. 3.14 : Variables de contrôle de l'asservissement visuel 2D pour le test 2

A travers ces résultats de simulation, nous pouvons avancer que l'approche de la commande par asservissement visuel 2D est validée, car la tâche converge de manière systématique et sans saturation des commandes ni déstabilisation du système dans tous les cas testés. Ces affirmations vont être consolidées par les tests expérimentaux que nous présentons ci-après.

3.3 Implémentation expérimentale

Nous présentons ici les résultats obtenus de l'implémentation expérimentale en temps réel sur le robot *Rackham* de l'asservissement visuel 2D classique. Afin de bien présenter ces résultats et faciliter leur interprétation, nous les exposons comme ceux des simulations, et rajoutons une nouvelle figure qui montre l'évolution d'une nouvelle variable appelée « *Erreur de positionnement* ». Comme à chaque instant t le centre du rectangle englobant le visage a pour coordonnées (u, v) dans l'image (voir figure 3.4), cette erreur notée E_p , représente la différence entre ses deux coordonnées horizontales : v (à l'état courant) et v_d (à la configuration désirée). L'évolution de cette nouvelle variable nous informe mieux sur la position du visage de l'utilisateur dans l'image, et donc du déroulement de la tâche.

De plus, on tient à préciser que contrairement aux tests de simulation, la configuration désirée ne correspond pas toujours à une position centrée dans l'image. En effet, cette dernière est mesurée de l'image à la position désirée, qui est choisie par l'utilisateur. Pour cela, nous avons indiqué par une croix rouge sur la figure 3.15 la position du rectangle à la configuration désirée pour chaque test (qui n'est pas la même nécessairement). De plus, et comme il s'agit d'un asservissement visuel sur l'homme, nous n'avons pas d'informations précises sur les positions réelles (3D) de l'utilisateur à l'instant initial ni pendant l'exécution de la tâche.

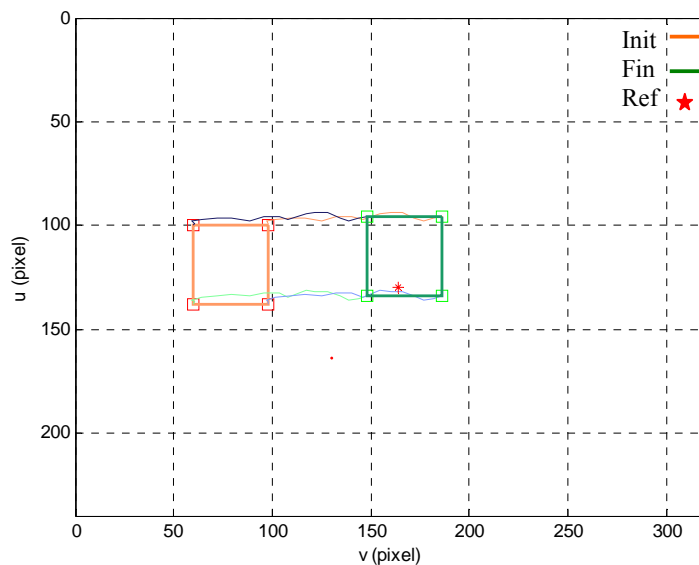


Fig. 3.15 : Trajectoires des indices visuels en expérimentation

La figure 3.15 présente l'évolution des indices visuels durant l'exécution de la tâche. On remarque que la position finale du visage (en vert) est très proche de la configuration désirée indiquée par l'étoile rouge. Cela est confirmé avec plus de précision sur la figure

3.16.b où E_p tend vers des valeurs très faibles ($< 5\text{pix}$) à la fin de la tâche. De même pour les trois composantes de la fonction tâche e représentées dans la figure 3.16.a. par contre, la taille de rectangle ne change presque pas.

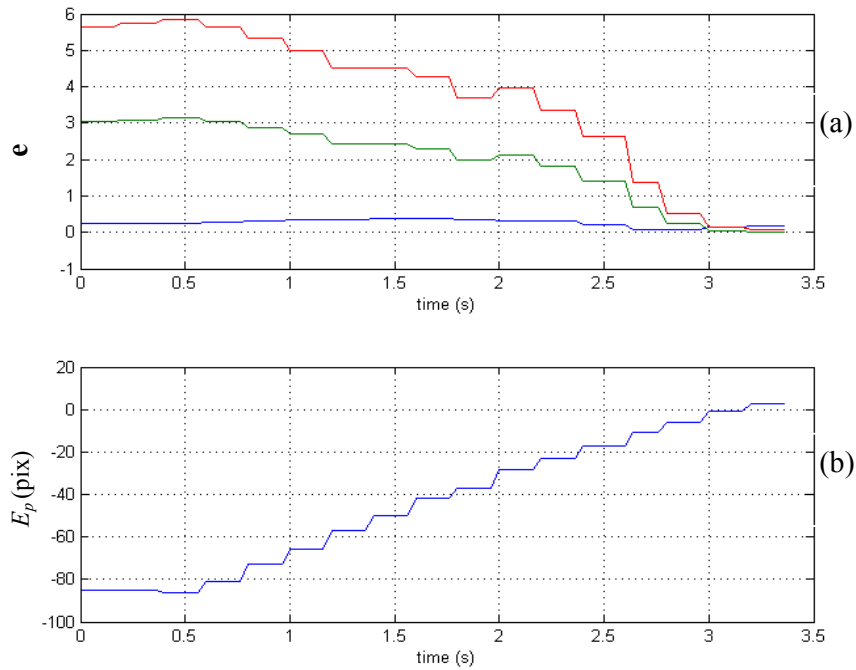


Fig. 3.16 : Test expérimental de l'asservissement visuel 2D classique (a) Evolution de la fonction de tâche, (b) Erreur de positionnement de la cible

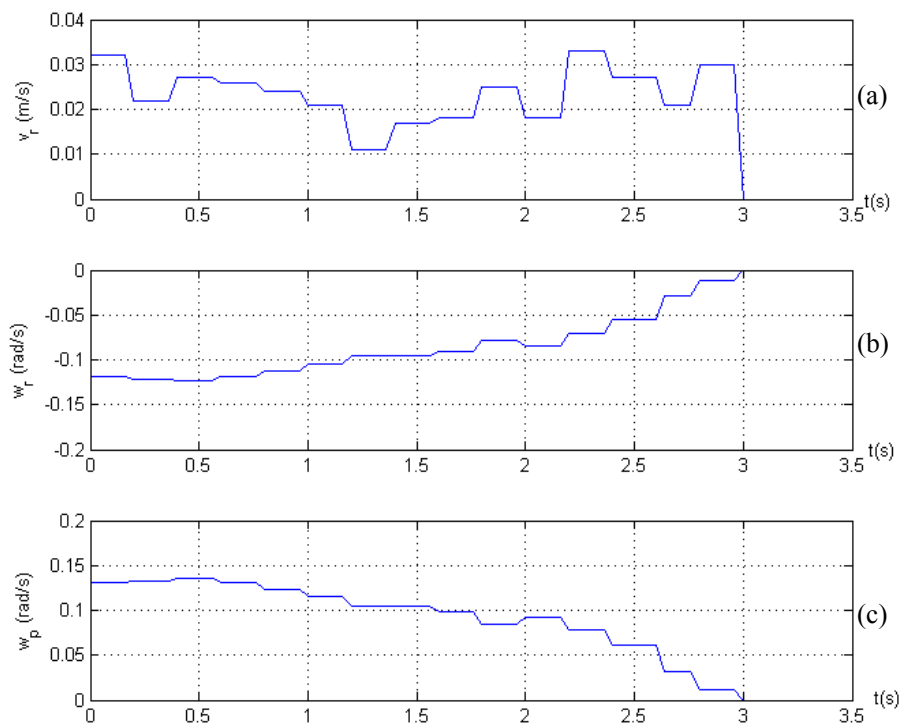


Fig. 3.17 : Variables de contrôle de l'asservissement visuel 2D

D'un autre côté, les vitesses injectées au robot et à la platine sont d'amplitudes bornées et admissibles, et s'annulent progressivement avec la convergence du motif visuel vers la position désirée. On note que la détection de visage fonctionne correctement, et permet de suivre l'utilisateur tout au long de la tâche. Toutefois, il a été constaté expérimentalement que l'évolution de la taille du rectangle n'est pas tout le temps cohérente avec la distance caméra-utilisateur. Cela peut perturber dans certains cas le bon déroulement de la tâche, vu que le calcul des indices visuels peut être affecté par ce problème.

A travers les multiples tests expérimentaux effectués dans cette étude, nous pouvons avancer que ce type de régulateur donne de bonnes performances en stabilité et précision quand la cible est plus ou moins statique. Toutefois, en cas de cible mobile, les mouvements des indices visuels dans l'image perturbent considérablement le régulateur, et peuvent parfois entraîner la divergence du système, comme il a été constaté en expérimentation [87]. De plus, en cas de perte du signal visuel (occultations, panne du capteur ou encore sortie de la cible du champ de vision), le régulateur s'arrête et attend le retour du signal visuel pour continuer l'exécution de la tâche demandée. Nous proposons dans le chapitre suivant des solutions à ces problèmes de robustesse et d'efficacité de technique de commande.

4. CONCLUSION

La première partie de ce chapitre a été consacrée à la présentation et la modélisation du système robotique utilisé dans cette étude. Nous avons alors décrit le matériel contenu dans la plateforme expérimentale, puis l'architecture logicielle qui le contrôle. En deuxième partie, nous avons présenté la synthèse d'un asservissement visuel 2D sur l'homme, puis exposé les résultats des simulations et des expérimentations effectués sur la plateforme en question.

Les performances obtenues sont encourageantes, mais méritent d'être améliorées d'avantage, tant sur le plan loi de commande que sur le plan moyen de détection et d'identification de l'utilisateur. C'est ce qu'on traite au chapitre suivant.

CHAPITRE 4

SYNTHÈSE ET IMPLÉMENTATION DE LOIS DE COMMANDE ROBUSTES & MULTIMODALES

1. INTRODUCTION

L'asservissement visuel sur l'homme est prometteur si on se réfère aux résultats obtenus au chapitre précédent. Toutefois, il est jugé possible d'apporter des améliorations par rapport à deux aspects importants : le premier est lié au contrôle où à loi de commande en termes de robustesse. Le second aspect concerne la perception du fait qu'il est plus judicieux d'associer d'autres informations sensorielles à la vision artificielle, afin de pallier aux problèmes de non détection de l'utilisateur. Ainsi, nous proposons dans ce qui suit une nouvelle stratégie de commande référencée capteurs pour une tâche d'initiation et de maintien d'interaction Homme/Robot. Il s'agit d'une technique *Multimodale* utilisant la caméra et la détection par RFID, et *Hybride* car elle contrôle le robot via deux lois de commande : les régimes (ou modes) glissants réputés robustes, et le contrôleur classique présenté au chapitre précédent. Nous présentons d'abord les résultats de l'asservissement visuel robuste, puis ceux de la technique multimodale, et on termine par l'expérimentation de la stratégie globale.

2. ASSERVISSEMENT VISUEL PAR LES MODES GLISSANTS

Les performances obtenues de l'implémentation d'un asservissement visuel 2D sur l'homme sont encourageantes. Seulement, la loi de commande classique employée présente certains inconvénients comme la sensibilité aux erreurs de modélisation, aux bruits sur les informations visuelles ou encore aux mouvements de la cible. De ce fait, nous avons introduit une autre loi de commande réputée plus robuste. Notre choix s'est porté alors sur la technique de la commande par les modes glissants (ou régimes glissants) dont les principes de bases sont détaillés dans ce qui suit.

2.1 Principes de la commande par modes glissants

Les algorithmes de commandes classiques de type « Proportionnel », « Intégral » et « Dérivé » (PID) et variantes ont plus de cinquante ans de savoir faire. Cependant, pour les systèmes dynamiques qui présentent des variations paramétriques et des non linéarités pouvant être soit inhérentes au système, soit présentes en cours d'utilisation (hystérésis, couplage, etc.), ces algorithmes montrent leurs limites lors de leur implémentation [22]. C'est le cas pour notre commande classique par asservissement visuel 2D vis-à-vis des facteurs suivants [88] :

- Incertitudes de modélisation et de calibrage (Calcul de $L_{(s,z)}$),
- Erreurs de mesure des indices visuels s et de la profondeur z ,
- Mouvements propre de la cible.

Deux approches peuvent donc exister : une première consisterait à obtenir un modèle « précis » du système physique et lui appliquer un algorithme de commande adapté, et une seconde utiliserait un modèle « simplifié » et un algorithme robuste qui prend en compte toutes les incertitudes (bornées certes) [22]. Pour le cas de notre système, nous avons choisi d'utiliser plutôt une loi de commande réputée robuste. Parmi celles-ci, ce sont les algorithmes à structure variable qui retiennent notre attention, particulièrement ceux qui génèrent des régimes glissants.

Le problème d'une équation différentielle à second membre discontinu est qu'elle n'est pas définie au point de discontinuité, car l'existence et l'unicité des solutions en ce point ne sont pas garanties. *A. F. Filippov* [42] a établi dans ce sens une théorie mathématique systématique. Ces travaux théoriques ont donné naissance à la commande à structure variable par modes glissants en URSS, établie et élaborée par *S. V. Emelyanov* et ses collaborateurs

[37], [103]. La loi de commande des systèmes à structure variable varie d'une manière discontinue, et la commutation s'effectue en fonction de la position du point représentatif du système dans l'espace de phase par rapport à la fonction de commutation. Cette fonction de commutation est en fait une contrainte que la dynamique du système doit satisfaire. Lorsque c'est le cas, on dit que le système est en mode ou régime glissant. Sa dynamique est alors celle de cette fonction, et pendant le glissement le système est insensible aux variations paramétriques telles une charge inertielle variable, aux incertitudes comme un système mal identifié et à une classe de perturbations externes.

2.2 Bouclage par les modes glissants classiques

2.2.1 Formalisme de la commande à structure variable

Soit un système du type SISO (Simple Input Simple Output, ou Monovariante) représenté par l'équation d'état suivante :

$$\dot{x} = f(x, t) + g(x, t)u \quad (4.1)$$

L'état $x \in \mathcal{O}$, \mathcal{O} est un ouvert de \mathfrak{R}^n , $f(x, t)$ et $g(x, t)$ sont des champs de vecteurs mal connus, suffisamment différentiables et définis sur \mathfrak{R}^n avec $g(x, t) \neq 0 \forall x \in \mathcal{O}$. u est la commande définie dans \mathfrak{R} , $u : \mathcal{O} \rightarrow \mathfrak{R}$. Il s'agit alors de trouver :

- La fonction de commutation représentée par $\sigma(x) \in \mathfrak{R}$,
- Une commande à structure variable telle que la condition de glissement $\sigma(x) = 0$ soit atteinte en temps fini.

D'un point de vue physique, il s'agit en fait de :

- Construire une surface de commutation $\sigma(x) = 0$ qui représente une dynamique désirée du système, et qui est d'un ordre inférieur à celui-ci,
- Construire une commande à structure variable u telle que tout état $x(t)$ n'appartenant pas à la surface de commutation atteigne celle-ci en temps fini, et y demeure en dépit des incertitudes et variations paramétriques.

Sur la surface de commutation, il y a un régime glissant qui satisfait la dynamique désirée. Le système à structure variable est alors globalement asymptotiquement stable.

2.2.2 Approche de Filippov

Le problème de la résolution d'équations différentielles à second membre discontinu est traité par *Filippov* [42]. Soit le système général :

$$\dot{x}_i = f_i(x_1, x_2, \dots, x_n, t) \quad (4.2)$$

Les fonctions f_i sont continues mais présentent des discontinuités sur $\sigma(x) = 0$ avec $x = (x_1, x_2, \dots, x_n)^T$, $\sigma(x)$ étant la fonction de commutation. Posons $f = (f_1, f_2, \dots, f_n)^T$, et étant donné que le système est à structure variable, sa dynamique peut être décrite par deux structures :

$$f(x, t) = \begin{cases} f^+(x, t) & \text{si } \sigma(x) > 0 \\ f^-(x, t) & \text{si } \sigma(x) < 0 \end{cases} \quad (4.3)$$

La dynamique est définie partout dans l'espace d'état sauf sur $\sigma(x) = 0$. $\sigma(x)$ divise l'espace d'état en $\sigma^-(x)$ (où $\sigma(x) < 0$) et $\sigma^+(x)$ (où $\sigma(x) > 0$). f_N^+ et f_N^- sont les projections de $f^+(x, t)$ et $f^-(x, t)$ sur la normale à $\sigma(x)$ orientée de $\sigma^-(x)$ vers $\sigma^+(x)$ (voir figure 4.1). D'après un théorème de *Filippov*, si en chaque point de $\sigma(x)$, la condition $f_N^+ < 0$ et $f_N^- > 0$ est vérifiée, il existe alors une solution unique $x(t)$ pour le système (4.2) qui dépend de façon unique des conditions initiales [22].

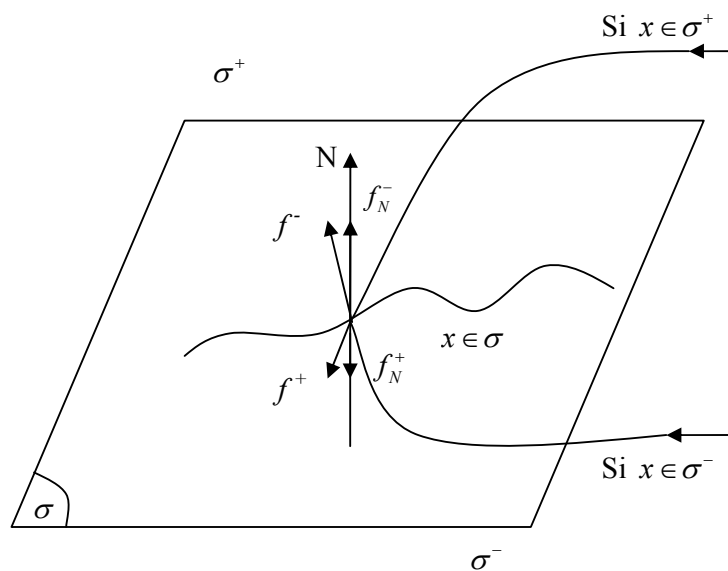


Fig. 4.1 : Espace d'état et surface de glissement

La discontinuité se produisant sur la surface $\sigma(x) = 0$, la dérivée de $\sigma(x)$ par rapport au temps est écrite comme :

$$\dot{\sigma}(x) = \sum_{i=1}^n \frac{\partial \sigma}{\partial x_i} \dot{x}_i = \sum_{i=1}^n \frac{\partial \sigma}{\partial x_i} f_i(x_1, x_2, \dots, x_n, t) = \langle N, f \rangle \quad (4.4)$$

où $\langle N, f \rangle$ représente le produit scalaire des vecteurs N et f .

On en déduit donc :

$$\begin{cases} f_N^+ < 0 \Rightarrow \sigma(x) > 0 \text{ et } \dot{\sigma}(x) < 0 \\ f_N^- > 0 \Rightarrow \sigma(x) < 0 \text{ et } \dot{\sigma}(x) > 0 \end{cases} \quad (4.5)$$

Donc les conditions $f_N^+ < 0$ et $f_N^- > 0$ sont équivalentes à $\sigma \cdot \dot{\sigma} < 0$. Cette inégalité représente la condition d'attractivité de la surface $\sigma(x)$ dans le plan de phase. Sur $\sigma(x) = 0$, *Filippov* décrit la dynamique du système comme une moyenne des deux structures de (4.3) par le champ de vecteur $f_0 = (f_{10}, f_{20}, \dots, f_{n0})^T$ [22].

$$\dot{x} = f_0(x) = \lambda f_0^+ + (1 - \lambda) f_0^-, \quad 0 \leq \lambda \leq 1 \quad (4.6)$$

où $f_0^+ = \lim_{s \rightarrow 0} f^+(x, t)$ et $f_0^- = \lim_{s \rightarrow 0} f^-(x, t)$ sont fonctions de $x(t)$, ainsi que λ qui est spécifiée de sorte que $f_0(x)$ (la dynamique moyenne) soit tangente à la surface $\sigma = 0$ (voir figure 4.2).

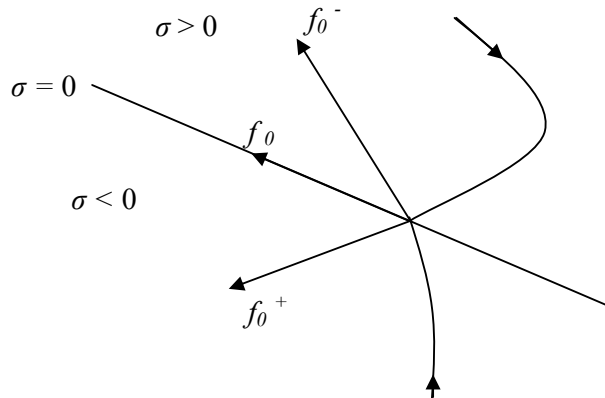


Fig. 4.2 : Dynamique de glissement selon *Filippov*

Le système bouclé en mode glissant est représenté donc par :

$$\dot{x} = \left[\frac{\langle \text{grad}(\sigma), f^- \rangle}{\langle \text{grad}(\sigma), (f^- - f^+) \rangle} \right] f^+ - \left[\frac{\langle \text{grad}(\sigma), f^+ \rangle}{\langle \text{grad}(\sigma), (f^- - f^+) \rangle} \right] f^- = f_0 \text{ et } x \in \sigma(x). \quad (4.7)$$

2.2.3 Approche d'Utkin [22]

Soit le système non linéaire affine en entrée défini par :

$$\dot{x} = f(x) + g(x)u \quad (4.8)$$

Avec un choix de fonction de commutation $\sigma(x)$, la première étape de cette approche est de déterminer la commande équivalente $u_e(x)$ de sorte que la trajectoire d'état reste sur la surface $\sigma(x) = 0$. Une fois celle-ci connue, le régime glissant idéal peut être décrit en substituant $u_e(x)$ dans l'équation (4.8). Cette commande équivalente est établie par la condition nécessaire de glissement, à savoir $\dot{\sigma}(x) = 0$, pour que la trajectoire d'état demeure sur $\sigma(x) = 0$. La dérivée par rapport au temps de $\sigma(x)$ donne, le long de la trajectoire de (4.8) :

$$\dot{\sigma}(x) = \left(\frac{\partial \sigma}{\partial x} \right)^T [f(x) + g(x)u_e] = 0 \quad (4.9)$$

Le calcul de $u_e(x)$ est donné alors par :

$$u_e(x) = - \left(\left(\frac{\partial \sigma}{\partial x} \right)^T g(x) \right)^{-1} \left(\frac{\partial \sigma}{\partial x} \right)^T f(x) \quad (4.10)$$

où l'existence de $\left(\left(\frac{\partial \sigma}{\partial x} \right)^T g(x) \right)$ est une condition nécessaire d'existence qui correspond à la condition de transversalité. Le terme $u_e(x)$ est interprété comme la commande moyenne lorsque celle-ci commute entre u_{max} et u_{min} (voir figure 4.3).

$$u_{min} = \min \{ u^+(x), u^-(x) \} < u_e(x) < u_{max} = \max \{ u^+(x), u^-(x) \} \quad (4.11)$$

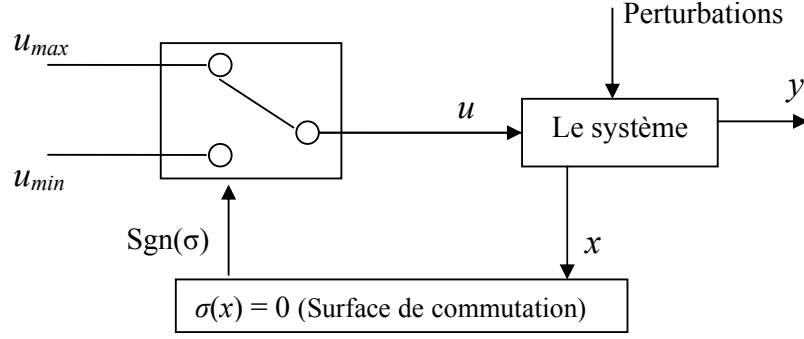


Fig. 4.3 : Principe de la commande à structure variable

Le système bouclé en régime glissant idéal est donc décrit par :

$$\begin{cases} \dot{x} = \left[I - g(x) \left(\left(\frac{\partial \sigma}{\partial x} \right)^T g(x) \right)^{-1} \left(\frac{\partial \sigma}{\partial x} \right)^T \right] f(x) \\ \sigma(x) = 0 \end{cases} \quad (4.12)$$

Les dynamiques de *Filippov* et *Utkin* sont généralement différentes. La condition d'attractivité $\dot{\sigma} < 0$ se résume donc à deux conditions :

$$\lim_{\sigma \rightarrow 0^+} \dot{\sigma} < 0 \quad \text{et} \quad \lim_{\sigma \rightarrow 0^-} \dot{\sigma} > 0 \quad (4.13)$$

De plus cette condition, qui est la condition nécessaire pour assurer le glissement, n'est pas suffisante pour satisfaire une convergence en temps fini. Afin d'éviter une convergence asymptotique, cette condition d'attractivité est généralement remplacée par la condition dite de η -attractivité pour laquelle le temps de convergence est borné par : $t_c \leq \frac{|\sigma(0)|}{\eta}$ en partant d'une condition initiale $\sigma(0)$. C'est cette condition qui a été employée dans cette étude.

$$\sigma \dot{\sigma} \leq -\eta |\sigma| \quad (4.14)$$

2.2.4 Approche de Sira-Ramirez

Le régime glissant est décrit par *Sira-Ramirez* en utilisant le contexte de la géométrie différentielle, et l'analyse d'un système en régime glissant y apparait très clairement [22].

A. Le régime glissant :

Soit le système non-linéaire autonome défini localement pour tout $x \in \mathbf{O}$, un ouvert de \mathfrak{R}^n :

$$\dot{x} = f(x) + g(x)u, \quad y = \sigma(x) \quad (4.15)$$

Les champs de vecteurs $f(x)$ et $g(x)$ sont mal connus et suffisamment différentiables définis sur \mathfrak{R}^n avec $g(x) \neq 0 \quad \forall x \in \mathbf{O}$. La commande u est définie dans \mathfrak{R} , $u : \mathbf{O} \rightarrow \mathfrak{R}$ et la sortie $\sigma(x)$ est continue, $\sigma(x) : \mathbf{O} \rightarrow \mathfrak{R}$. $f(x)$ est généralement appelé champ des vecteurs de dérive, alors que $g(x)$ est nommé champ des vecteurs de commande. La surface ou hypersurface de glissement est représentée par l'ensemble des vecteurs d'état dans \mathbf{O} satisfaisant la variable de glissement $\sigma(x) = 0$. Cet ensemble constituant la surface de glissement est défini par :

$$\sigma = \{x \in \mathbf{O} / \sigma(x) = 0\} \quad (4.16)$$

L'ensemble σ représente une sous variété de \mathbf{O} de dimension $n-1$, avec $\text{grad } \sigma \neq 0$ au moins localement, et orienté de la région où $\sigma(x) < 0$ vers celle où $\sigma(x) > 0$. Le système de commande à structure variable est comme suit :

$$u = \begin{cases} u^+(x) & \text{si } \sigma(x) > 0 \\ u^-(x) & \text{si } \sigma(x) < 0 \end{cases} \quad (4.17)$$

La commutation de la commande u s'effectue pour contraindre le système à converger vers la surface de glissement $\sigma(x)$ et y demeurer. Le système est alors dit « *en régime glissant idéal* » [22]. S'il est noté $L_f \sigma(x)$ la dérivée de Lie de la fonction scalaire $\sigma(x)$ suivant le champ de vecteurs f : $L_f \sigma(x) = \langle \text{grad } \sigma(x), f \rangle$ où $\langle \cdot, \cdot \rangle$ représente le produit scalaire de deux vecteurs. Demeurer sur la surface de glissement σ par l'utilisation

de (4.17) implique une commutation à une fréquence infinie, car pour des fréquences finies à chaque commutation, la trajectoire erre au-delà de la surface. Le système est en régime glissant si :

$$\lim_{\sigma \rightarrow 0^+} L_{(f+g.u^+)} \sigma < 0 \quad \text{et} \quad \lim_{\sigma \rightarrow 0^-} L_{(f+g.u^-)} \sigma > 0 \quad (4.18)$$

Ou bien,

$$\lim_{\sigma \rightarrow 0^+} \langle \text{grad} \sigma, (f + g.u^+) \rangle < 0 \quad \text{et} \quad \lim_{\sigma \rightarrow 0^-} \langle \text{grad} \sigma, (f + g.u^-) \rangle > 0 \quad (4.19)$$

Donc, dans le voisinage de la surface de commutation $\sigma(x) = 0$, les champs commandés sont dirigés vers celle-ci [22].

B. Conditions d'existence et unicité du contrôle équivalent :

En régime glissant idéal, la trajectoire d'état évolue le long de $\sigma(x) = 0$, et la commutation de la commande s'effectue à une fréquence infinie. Il peut être considéré qu'elle ait été produite par une commande continue au lieu d'une commande discontinue. C'est l'équivalence entre les deux types de commande qui est nommée « commande équivalente ». Si la trajectoire d'état est sur σ , la commande équivalente $u_e(x)$ l'y maintient. Cette commande équivalente satisfait donc les conditions d'invariance :

$$\begin{cases} \sigma(x) = 0 \\ \dot{\sigma}(x) = \frac{\partial \sigma}{\partial x} (f(x) + g(x)u_e(x)) = 0 \end{cases} \quad (4.20)$$

Cela peut être traduit par l'équation suivante :

$$L_{(f+gu_e)} \sigma(x) = L_f \sigma(x) + u_e(x) [L_g \sigma(x)] = 0 \quad (4.21)$$

Par invariance, il est entendu que la dynamique de (4.15) n'est régie que par l'ensemble de glissement σ . La commande équivalente $u_e(x)$ est alors exprimée comme :

$$u_e(x) = -\frac{L_f \sigma(x)}{L_g \sigma(x)} \quad (4.22)$$

En régime glissant idéal, la dynamique du système est :

$$\dot{x} = f + g u_e(x) = f(x) - g(x) \frac{L_f \sigma(x)}{L_g \sigma(x)} \quad (4.23)$$

Cette dynamique ne dépend alors que des coefficients de la surface de glissement. La condition nécessaire d'existence de la commande équivalente est :

$$L_g \sigma(x) = \langle \text{grad} \sigma, g \rangle \neq 0 \quad (4.24)$$

C'est la condition de transversalité. Elle indique que le champ de vecteurs de commande $g(x)$ ne peut pas être tangent à la surface de glissement σ [22].

Par rapport à l'unicité du contrôle équivalent, un théorème établit ce qui suit pour un système à structure variable défini par (4.15), comme dans [22] :

- Une condition nécessaire pour l'existence locale d'un régime glissant $\sigma(x) = 0$ sur un ouvert \mathbf{O} est qu'il existe localement une commande équivalente unique, continue, satisfaisant $u^-(x) < u_e < u^+(x)$ pour des commandes continues $u^-(x)$ et $u^+(x)$. En supposant la condition d'invariance (4.20) satisfaite pour deux commandes équivalentes u_{e1} et u_{e2} , on écrit :

$$\begin{aligned} \langle \text{grad} \sigma, (f + g u_{e1}) \rangle &= \langle \text{grad} \sigma, (f + g u_{e2}) \rangle = 0 \\ \langle \text{grad} \sigma, [g(u_{e1} - u_{e2})] \rangle &= 0 \\ (u_{e1} - u_{e2}) L_g \sigma(x) &= 0 \end{aligned} \quad (4.25)$$

Avec la condition de transversalité (4.24), on conclut que $u_{e1} = u_{e2}$.

Maintenant, si un régime glissant local existe sur \mathbf{O} pour la commande discontinue de (4.17), alors localement les relations suivantes sont valides :

$$L_{(f+g.u^+)}\sigma = \langle \text{grad } \sigma, (f + g.u^+) \rangle < 0 \quad (4.26)$$

$$L_{(f+g.u_e)}\sigma = \langle \text{grad } \sigma, (f + g.u_e) \rangle = 0 \quad (4.27)$$

$$L_{(f+g.u^-)}\sigma = \langle \text{grad } \sigma, (f + g.u^-) \rangle > 0 \quad (4.28)$$

Soustraire (4.27) de (4.26) et (4.28) de (4.27) donne :

$$\begin{cases} \langle \text{grad } \sigma, [(f + g.u^+) - (f + g.u_e)] \rangle < 0 \\ \langle \text{grad } \sigma, [(f + g.u_e) - (f + g.u^-)] \rangle < 0 \end{cases} \quad (4.29)$$

En utilisant le théorème de la valeur moyenne [22], il existe des fonctions continues $u_0^+(x)$ et $u_0^-(x)$ localement sur \mathbf{O} telles que :

$$\begin{cases} \langle \text{grad } \sigma, [(f + g.u^+) - (f + g.u_e)] \rangle = [u^+ - u_e] \left\langle \text{grad } \sigma, \frac{\partial (f + g.u_0^+)}{\partial u} \right\rangle < 0 \\ \langle \text{grad } \sigma, [(f + g.u_e) - (f + g.u^-)] \rangle = [u_e - u^-] \left\langle \text{grad } \sigma, \frac{\partial (f + g.u_0^-)}{\partial u} \right\rangle < 0 \end{cases} \quad (4.30)$$

où $u_0^+(x)$ et $u_0^-(x)$ satisfont respectivement :

$$u_e(x) < u_0^+(x) < u^+(x) \quad \text{et} \quad u^-(x) < u_0^-(x) < u_e(x)$$

Et donc :

$$u^-(x) < u_e(x) < u^+(x)$$

C. Rejets des perturbations :

Le système en boucle fermée commandé par $u_e(x)$ étant en régime glissant idéal, il peut être décrit par :

$$\dot{x} = f(x) + g(x)u_e(x) = \left[I - g(x)(\langle grad \sigma, g \rangle)^{-1} \cdot grad \sigma \right] f(x) = P(x) \cdot f(x) \quad (4.31)$$

où $P(x) = \left[I - g(x)(\langle grad \sigma, g \rangle)^{-1} \cdot grad \sigma \right]$ est l'opérateur de projection sur l'espace tangent à σ .

S'il est maintenant considéré un vecteur $w(x)$ agissant le long du champ de vecteurs de commande $g(x)$, $w(x) \in Span\{g(x)\}$ i.e. de la forme [22] :

$$w(x) = \alpha(x) \cdot g(x) \quad (4.32)$$

où $\alpha(x) \in \mathfrak{R}$,

Et si on applique l'opérateur de projection $P(x)$, il en résulte :

$$\begin{aligned} P(x)[\alpha(x)g(x)] &= \left[I - g(x)(\langle grad \sigma, g \rangle)^{-1} \cdot grad \sigma \right] g(x)\alpha(x) \\ &= \left[g(x) - g(x) \left((\langle grad \sigma, g(x) \rangle)^{-1} (\langle grad \sigma, g(x) \rangle) \right) \right] \alpha(x) \\ &= [g(x) - g(x)]\alpha(x) = 0 \end{aligned} \quad (4.33)$$

Donc, le glissement idéal est indépendant du champ de vecteurs $g(x)$. Du moment que cette indépendance est prouvée, si une perturbation agit sur le système dans la direction de $g(x)$, il est clair que son effet est nul. Ces perturbations sont appelées « *matching perturbations* » ou perturbations de recouvrement. Le régime glissant est donc indépendant de toute perturbation appartenant au « $Span\{g(x)\}$ » [22].

D. Approche du degré relatif :

Le système décrit par l'équation (4.15) possède un degré relatif $r \in \mathbf{N}^+$ au point x_0 si, pour $k \in \mathbf{N}^+$, $L_g L_f^k \sigma(x) = 0 \quad \forall x \in [x_0 \pm \varepsilon]$ pour $k < r-1$, et $L_g L_f^{r-1} \sigma(x_0) \neq 0$. Rappelons que la dérivée de Lie, notée $L_f \sigma(x)$, d'une fonction scalaire $\sigma(x)$ suivant le champ de vecteurs f est donnée par : $L_f \sigma(x) = \langle grad \sigma(x), f \rangle$. Aussi, il peut être écrit pour $k \in \mathbf{N}^+$, $L_f^k \sigma(x) = L_f(L_f^{k-1} \sigma(x))$ et $L_f^0 \sigma(x) = \sigma(x)$.

En fait, de manière plus simple, le degré relatif du système correspond au nombre minimal de dérivations de $y = \sigma(x)$ par rapport au temps pour y faire apparaître, explicitement, la commande u .

- *Condition de transversalité - Existence du glissement* : Il a été établi que la condition nécessaire d'existence de la commande équivalente est : $L_g \sigma(x) \neq 0$. Si $L_g \sigma(x)$ ne dépend pas localement de la commande u , c'est-à-dire $\frac{\partial L_g \sigma(x)}{\partial u(x)} = 0$, alors le changement de la commande de $u^+(x)$ à $u^-(x)$ dans le voisinage du point x_0 dans la structure variable n'affectera pas le signe de $L_g \sigma(x)$, donc un régime glissant ne peut pas exister localement sur $\sigma(x)$. Un régime glissant classique ou statique ne peut donc exister localement sur $\sigma(x)$ que si et seulement si le système possède un degré relatif égale à 1.
- *Régime glissant et bouclage linéarisant pour le cas de degré relatif supérieur à 1* : Si maintenant pour la sortie du système (4.15), le système possède un degré relatif $r > 1$, alors un régime glissant n'existera pas localement sur $\sigma(x) = 0$. Cependant, l'utilisation des travaux d'Isidori sur la stabilisation locale par retour d'état permet l'établissement d'une sortie auxiliaire $K(x)$ [22] :

$$\begin{aligned}
 K(x) &= \sigma^{(r-1)} + c_{r-2} \sigma^{(r-2)} + \dots + c_1 \dot{\sigma} + c_0 \sigma \\
 K(x) &= L_f^{r-1} \sigma(x) + c_{r-2} L_f^{r-2} \sigma(x) + \dots + c_1 L_f \sigma(x) + c_0 \sigma(x) \\
 K(x) &= z_r + c_{r-2} z_{r-1} + \dots + c_2 z_3 + c_1 z_2 + c_0 z_1
 \end{aligned} \tag{4.34}$$

avec $\mathbf{z} = [z_1 \ z_2 \ \dots \ z_r]^T$ et $z_i = \Phi_i(x) = L_f^{i-1} \sigma(x)$.

Le vecteur $\Phi(\mathbf{x}) = [\Phi_1(x), \Phi_2(x), \dots, \Phi_n(x)]^T$ constitue une transformation de coordonnées locale dont la matrice jacobienne est non singulière en x_0 .

Un régime glissant local existe sur $K(x_0)$ si :

$$\dot{K}(x) = L_f K(x) + u L_g K(x) \quad \text{et} \quad L_g L_f^{(r-1)} \sigma(x_0) \neq 0, \quad r > 1.$$

Donc idéalement $K(x_0) = 0$ et $z_r = -c_{r-2}z_{r-1} - \dots - c_2z_3 - c_1z_2 - c_0z_1$. Le système en régime glissant idéal associé à la nouvelle surface $K(x)$ est :

$$\begin{cases} \dot{z}_i = z_{i+1} & i = 1, \dots, r-2 \\ \dot{z}_{r-1} = z_r = -c_{r-2}z_{r-1} - \dots - c_1z_2 - c_0z_1 \\ \dot{z}_r = b(z) + a(z)u \\ \dot{z}_{r+j} = q(z_1, z_2, \dots, z_{r-1}, z_r, z_{r+1}, \dots, z_n) & j = 1, \dots, n-r \\ z_1 = \sigma(x) \end{cases} \quad (4.35)$$

avec $b(z) = L_f^r \sigma(\Phi^{-1}(z))$ et $a(z) = L_g L_f^{r-1} \sigma(\Phi^{-1}(z)) \neq 0$.

Le système est dit sous «*forme normale*». Le choix des coefficients c_0, c_1, \dots, c_{r-2} assure la stabilité asymptotique des $r-1$ premières coordonnées z_1, z_2, \dots, z_{r-1} , et donc aussi de z_r en faisant en sorte que tous les pôles de l'équation caractéristique $p^{(r-1)} + c_{r-2}p^{(r-2)} + \dots + c_1p + c_0 = 0$ appartiennent au demi plan gauche complexe [22]. Il peut être conclu que si un régime glissant existe localement sur $K(x_0)$, la sortie et ses $r-1$ premières dérivées approchent zéro asymptotiquement, et donc que le vecteur d'état du système approche $\sigma(x) = 0$ comme désiré. La commande équivalente est donnée par :

$$u_e(x) = -\frac{L_f K(x)}{L_g K(x)} = \{L_f^r \sigma + c_{r-2} L_f^{r-1} \sigma + \dots + c_1 L_f^2 \sigma + c_0 L_f \sigma\} / L_g L_f^{r-1} \sigma \quad (4.36)$$

Localement autour de x_0 sur $\sigma(x) = 0$, cette expression peut être écrite sous la forme :

$$u_e(x) = \frac{L_f^r \sigma(x)}{L_g L_f^{r-1} \sigma(x)} \quad (4.37)$$

2.2.5 Composantes de la commande

En régimes glissants, la commande discontinue u se compose en fait de deux termes : une composante haute fréquence u_{hf} et une autre basse fréquence u_{bf} comme suit :

$$u = u_{hf} + u_{bf} \quad (4.38)$$

La composante haute fréquence est filtrée par la bande passante du système, et le régime glissant n'est affecté que par u_{bf} qui est considérée comme la sortie d'un filtre passe-bas [22]. Ceci revient à établir que la commande équivalente qui est continue $u_e \cong u_{bf}$, représente la valeur moyenne de la commande discontinue u . Elle maintient l'état sur la surface de commutation $\sigma(x) = 0$ une fois celle-ci atteinte. La commande nécessite une composante discontinue haute fréquence Δu qui compense les incertitudes du modèle :

$$u = u_e + \Delta u \quad (4.39)$$

Cette composante discontinue Δu peut prendre diverses formes [103], [22] dont la plus simple et la plus usitée est :

$$\Delta u = -Sgn(\sigma(x)) \quad (4.40)$$

Ou encore d'autres formes, plus complexes, existantes dans la littérature [22] :

$$\Delta u = \alpha \cdot \sigma + \beta \cdot \dot{\sigma} + \lambda \cdot Sgn(\sigma) \quad \text{et} \quad \Delta u = \left(\alpha |\sigma| + \beta \left| \dot{\sigma} \right| + \lambda \right) \cdot Sgn(\sigma)$$

où les paramètres α , β et λ sont à régler, et Sgn désigne la fonction Signe(x).

La commande à structure variable générant un régime glissant constitue un outil puissant par sa robustesse car une fois ce régime atteint, le système devient à la fois indépendant des incertitudes du modèle, des perturbations agissant dans le champ de recouvrement (matching perturbations) et des variations paramétriques. Seuls les paramètres de la surface de commutation sont dominants et régissent le comportement du système [22].

En réalité, pour la commande, seule la partie haute fréquence Δu commute entre deux valeurs et donc l'utilisation de u_e réduit l'amplitude de la partie discontinue. La commutation de la partie discontinue de la commande devrait en théorie s'établir à une fréquence infinie, ce qui est difficile à réaliser en pratique. Alors, les délais inhérents à la commutation font que la trajectoire d'état erre légèrement au-delà de la surface de glissement. Ce phénomène à fréquence élevée est appelé *broutement* ou *chattering* (voir figure 4.4). Il est hautement

indésirable car il risque d'exciter des dynamiques hautes fréquences non considérées, de conduire à l'instabilité et à la limite, porter atteinte au système physique (détériorer les actionneurs).

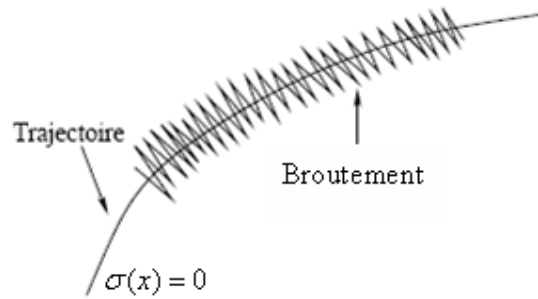


Fig. 4.4 : Phénomène du *Chattering*

Diverses méthodes sont utilisées pour l'atténuation du *chattering*, parmi lesquelles le remplacement de la composante discontinue Δu par une composante continue dans le voisinage immédiat de la surface de glissement [22] :

$$\Delta u = -K \cdot \text{cont}(\sigma(x)) \quad (4.41)$$

avec $\text{cont}(\sigma) = \frac{\sigma}{|\sigma| + \delta}$ et $\delta > 0$.

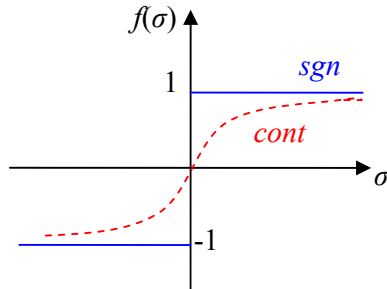


Fig. 4.5 : Exemples de fonctions de commutation

Aussi, pour atténuer ce phénomène, *Harashima* suggère l'introduction d'un terme intégral, alors que d'autres formes peuvent aussi contribuer à la réduction de ce dernier comme [22] :

$$\Delta u = -K \cdot \text{sat}(\sigma(x)) \quad (4.42)$$

avec $\text{sat}(\sigma) = \begin{cases} \sigma / \Phi & \text{si } |\sigma / \Phi| \leq 1 \\ \text{Sgn}(\sigma) & \text{si } |\sigma / \Phi| > 1 \end{cases}$ et $\Phi > 0$.

Une autre méthode consiste à utiliser une fonction sigmoïde telle $Tanh(\sigma)$, ou une fonction dont l'amplitude diminue à mesure que l'on s'approche de la surface de glissement telle $-|\sigma|^\alpha \cdot Sgn(\sigma)$. Le *chattering* est d'autant plus faible que le correcteur est bien calculé (paramètres de réglage optimaux). Généralement, ces divers outils ne sont utilisés que dans le voisinage immédiat de la surface de commutation, c'est-à-dire que loin de ce voisinage, la commande discontinue de la forme (4.40) ou (4.41) reste utilisée.

D'autres méthodes de réduction de *chattering* existent, utilisant une combinaison PI de la fonction de glissement dans son voisinage immédiat ou encore un correcteur de type PID dans le voisinage immédiat de la surface de commutation comme dans [21] et [22]. Cependant, dans ce voisinage, il y a dégradation de la robustesse. Il faut donc faire un compromis entre le broutement et la robustesse de la commande. D'autres outils existent tels la commutation sur les dérivées de la commande dans la structure variable généralisée (GVS), introduite par *Fliess*, ou les modes glissants d'ordre supérieur (*HOSM*) [37] [22].

2.3 Synthèse d'un Asservissement Visuel par Modes Glissants

Dans cette partie, nous présentons la synthèse d'un nouveau type de contrôleur par asservissement visuel 2D, basé sur la théorie des régimes glissants, comme il a été procédé dans [87]. Ce contrôleur, comme pour le cas classique (*section 3 du chapitre 3*) doit permettre au robot mobile de se positionner face à l'utilisateur, tout en garantissant que les indices visuels lui correspondant atteignent la configuration désirée dans l'image. Pour cela, nous nous basons à la fois sur le formalisme des fonctions de tâche, déjà détaillé dans le chapitre précédent, ainsi que les principes des régimes glissants classiques présentés à la section précédente. L'objectif est de réaliser une navigation référencée vision plus robuste par rapport aux erreurs de modélisation, et aux bruits des mesures et du traitement d'image. De ce fait, un choix évident de la surface de glissement, notée σ , est de la prendre égale à la fonction de tâche définie dans (3.27) [87] :

$$\sigma(q,t) = C(s(q,t) - s^*) \quad (4.43)$$

où C dénote la matrice de combinaison utilisée dans le cas de la commande classique. Il est important de noter que le choix de cette matrice doit se faire soigneusement, pour ne pas augmenter la complexité de la synthèse d'une loi de commande par régimes glissants. Il est alors vivement conseillé de prendre cette matrice constante, et fonction de la matrice

d'interaction calculée hors ligne (voir section 2.2.2 du chapitre 2), afin de simplifier le calcul des gains à partir des conditions de glissement (voir section 2.4) [88].

Avant de déduire la loi de commande, il est nécessaire d'écrire l'expression de la première dérivée de la surface de glissement, notée $\dot{\sigma}$. En utilisant les équations (4.43) et (3.20), on peut écrire :

$$\dot{\sigma} = C L_{(s,z)} \tau_{C/R_o}^{R_c} + \frac{\partial s}{\partial t} \quad (4.44)$$

La loi de commande par régimes glissants, notée \dot{q}_s , est classiquement composée de deux termes : un premier nommé commande équivalente notée \dot{q}_e , plus un second terme discontinu notée Δq , ce qui donne :

$$\dot{q}_s = \dot{q}_e + \Delta q \quad (4.45)$$

Le contrôle équivalent est calculé à partir de la condition de l'existence du régime glissant, à savoir ($\sigma = 0$) et ($\dot{\sigma} = 0$). De ce fait, l'annulation de la première dérivée de la surface de glissement en employant les expressions (4.44) et (3.20) donne un contrôle équivalent sous la forme :

$$\dot{q}_e = - (C L_{(s,z)} J)^{-1} C \frac{\partial s}{\partial t} \quad (4.46)$$

De même que pour le cas de la commande classique, et par souci de compatibilité des dimensions des matrices dans l'expression (4.45), on considère la jacobienne du robot et la matrice d'interaction sous leurs formes réduites, notées respectivement J_r et L_r , ce qui donne :

$$\dot{q}_e = - (C L_r J_r)^{-1} C \frac{\partial s}{\partial t} \quad (4.47)$$

Mais comme le calcul de la matrice d'interaction est basé sur les mesures des indices visuels s et de la profondeur z , cette dernière est le plus souvent déduite d'un estimateur,

faute de moyen spécifique de mesure. A partir de là, on obtient une estimation de la matrice d'interaction réduite, notée \hat{L}_r . De plus, le terme $\frac{\partial s}{\partial t}$ quantifie théoriquement le mouvement propre de la cible par rapport à la caméra. Si la cible est complètement immobile, alors il est mis à zéro comme dans [45]. Par contre, pour cette étude ce n'est pas réellement le cas du moment qu'il s'agit d'un asservissement visuel sur l'homme, et dans le cadre d'une tâche d'interaction. Malheureusement, du fait que nous ne disposons pas de moyens d'estimation précis et robuste, nous considérons tout au long de cette étude uniquement une vague estimation de ce terme, que nous notons $\frac{\hat{\partial s}}{\partial t}$. A partir de ces hypothèses, l'expression du contrôle équivalent devient :

$$\dot{q}_e = -(C \hat{L}_r J_r)^{-1} \frac{\hat{\partial \sigma}}{\partial t} \quad (4.48)$$

avec $\frac{\hat{\partial \sigma}}{\partial t} = C \frac{\hat{\partial s}}{\partial t}$.

Par contre, le terme discontinu est exprimé par une des expressions, bien connues dans la littérature, comme le plus souvent l'expression d'un simple relais à gain constant :

$$\Delta q = -K_S \text{Sgn}(\sigma) \quad (4.49)$$

où le gain K_S est dans ce cas une matrice diagonale 3x3, et qui peut être écrite sous la forme $K_S = \text{diag}(k_{S1}, k_{S2}, k_{S3})$.

Enfin, l'expression finale de la loi de commande par modes glissants référencée vision peut être exprimée comme suit :

$$\dot{q}_s = -(C \hat{L}_r J_r)^{-1} \frac{\hat{\partial \sigma}}{\partial t} - K_S \text{Sgn}(\sigma) \quad (4.50)$$

On rappelle que \dot{q}_s est un vecteur de dimension trois. Contrairement au cas de la commande classique où le choix des gains se fait arbitrairement par la technique dite essai-erreur, les gains du contrôleur par régime glissants (élément de la diagonale de la matrice K_S)

doivent garantir la condition de glissement ($\sigma \dot{\sigma} < 0$), afin d'assurer la stabilité globale du système et la convergence de la loi de commande. Les détails liés à cette condition sont exposés dans ce qui suit.

2.4 Stabilité et convergence du contrôleur

Dans la théorie des modes glissants, les deux aspects stabilité et convergence de la loi de commande sont étroitement liés. Usuellement, l'étude de stabilité se fait par le choix d'une fonction candidate de *Lyapunov* définie positive notée V . La stabilité est alors conditionnée par le signe de la dérivée de cette fonction. Pour le cas de cette étude, la fonction choisie est de la forme :

$$V = \frac{1}{2} \sigma^T \sigma = \frac{1}{2} \sum_{i=1}^3 \sigma_i^2 \quad (4.51)$$

où σ_i représente la i^{eme} composante de σ : la surface de glissement.

Le choix de la fonction candidate, mise à part sa forme quadratique énergétique usuelle, se justifie par le fait que le système est totalement décrit par σ lorsqu'il est en mode de glissement [22]. Cette fonction étant toujours positive, le régulateur à modes glissants est asymptotiquement stable si on assure la négativité absolue de la dérivée de la fonction de *Lyapunov* ($\dot{V} < 0$). De l'expression précédente, cette dérivée s'écrit comme suit :

$$\dot{V} = \sum_{i=1}^3 \sigma_i \dot{\sigma}_i \quad (4.52)$$

Afin de garantir la négativité de cette fonction dérivée, il suffit de contraindre en même temps les trois composantes de la somme à être inférieures à des grandeurs strictement négatives [88]. Cette manière de faire, appelée la η -attractivité, assure une convergence en temps fini du système. Pour cela, nous décomposons la condition de glissement en trois inégalités telle que :

$$\text{Pour } i = 1 \text{ à } 3, \text{ vérifier que } \sigma_i \dot{\sigma}_i \leq -\eta_i |\sigma_i| \quad (4.53)$$

où η_i est une constante strictement positive, servant à régler la vitesse de convergence du système. La i^{eme} composante du vecteur $\dot{\sigma}$, notée $\dot{\sigma}_i$ s'écrit sous la forme :

$$\dot{\sigma}_i = C_i \dot{s} \quad (4.54)$$

où C_i est la i^{eme} ligne de la matrice de combinaison C . Du moment qu'on est dans le cas de la commande par régimes glissants, on peut écrire :

$$\dot{\sigma}_i = C_i \left\{ L_r J_r \dot{q}_s + \frac{\partial s}{\partial t} \right\} \quad (4.55)$$

En introduisant l'expression de la loi de commande par régimes glissants, exprimée dans (4.50), on obtient :

$$\dot{\sigma}_i = -C_i L_r (C \hat{L}_r)^{-1} \frac{\partial \hat{\sigma}}{\partial t} - C_i L_r J_r K_S \text{Sgn}(\sigma) + C_i \frac{\partial s}{\partial t} \quad (4.56)$$

Le résultat du produit $C_i L_r (C \hat{L}_r)^{-1}$ est le i^{eme} vecteur de base, sous l'hypothèse que l'estimation de la matrice d'interaction est suffisamment précise ($\hat{L}_r \cong L_r$). L'expression (4.56) se simplifie de manière considérable, surtout si on considère que $C_i \frac{\partial s}{\partial t} = \frac{\partial \sigma_i}{\partial t}$, ce qui donne :

$$\dot{\sigma}_i = -C_i L_r J_r K_S \text{Sgn}(\sigma) \quad (4.57)$$

Cette simplification est effectuée, sous une deuxième hypothèse sur la qualité de l'estimation du terme $\frac{\partial s}{\partial t}$. Elle a été effectuée en supposant que $\frac{\partial \hat{\sigma}_i}{\partial t} \cong \frac{\partial \sigma_i}{\partial t}$. On peut alors injecter l'expression (4.57) dans (4.53) afin de déduire les gains garantissant la condition de stabilité de la loi de commande par régimes glissants. En effet, chaque inégalité de l'expression (4.53) s'écrit comme suit :

$$-\sigma_i C_i L_r J_r K_S \text{Sgn}(\sigma) \leq -\eta_i |\sigma_i| \quad (4.58)$$

En considérant le fait que $|\sigma_i| = \sigma_i \text{Sgn}(\sigma_i)$, et sous la condition que $\sigma_i \neq 0$, on peut écrire alors :

$$\text{Sgn}(\sigma_i) C_i L_r J_r K_S \text{Sgn}(\sigma) \geq \eta_i \quad (4.59)$$

Rappelons que l'inconnue dans cette inégalité n'est autre que la matrice des gains définie par $\mathbf{K}_S = \text{diag}(k_{S1}, k_{S2}, k_{S3})$. A partir de cette écriture, on peut reformuler l'inégalité (4.59) de la manière suivante :

$$\text{Sgn}(\sigma_i) \sum_{j=1}^3 [M_{ij} k_j \text{Sgn}(\sigma_j)] \geq \eta_i \quad (4.60)$$

avec $M_i = C_i L_r J_r$

Comme \mathbf{M}_i est un vecteur de dimension 3, M_{ij} est la j^{eme} composante de ce vecteur. Chaque gain scalaire k_{Sj} , élément de la diagonale de \mathbf{K}_S , est alors calculé à partir de cette inégalité. Enfin, le calcul des gains du contrôleur par régimes glissants est équivalent à la résolution d'un système de trois inégalités défini dans (4.60). Les éléments des vecteurs \mathbf{M}_i , et qui représentent les coefficients du système d'inégalités, sont dépendants du temps t et de la configuration du robot \mathbf{q} , mais la résolution mathématique d'un tel système n'est pas une tâche compliquée [88].

Toutefois, il est important de noter qu'en pratique, la possibilité de rencontrer des situations dans lesquelles le système en question n'a pas de solution admissible existe. On peut citer à titre d'exemple le cas où les valeurs des gains engendrent des commandes non applicables sur le robot (supérieures aux valeurs admissibles). Dans ce type de configuration, on peut dire que le contrôleur à régimes glissants n'est malheureusement pas adéquat pour réaliser la tâche robotique spécifiée. Afin de palier à ce problème, plusieurs solutions sont envisageables. La solution proposée dans cette étude est de combiner ce régulateur réputé robuste, avec le contrôleur classique développé au chapitre précédent. L'objectif est de gérer au mieux les cas d'inadmissibilité des commandes à injecter au robot. Cette solution est détaillée plus loin (voir section 4.1).

2.5 Simulations

Le régulateur à régimes glissants classiques synthétisé dans la section précédente à été d'abord testé en simulation sous *Matlab*, avant d'être implémenté sur le robot réel. Pour cela, nous avons effectué plusieurs tests de la même tâche référencée vision utilisée pour le cas classique, à savoir positionner le robot relativement à un utilisateur connu, et caractérisé par le rectangle qui englobe son visage. Ce dernier est présent dans l'image de la caméra à la configuration initiale du robot. Nous présentons ci-dessous quelques résultats.

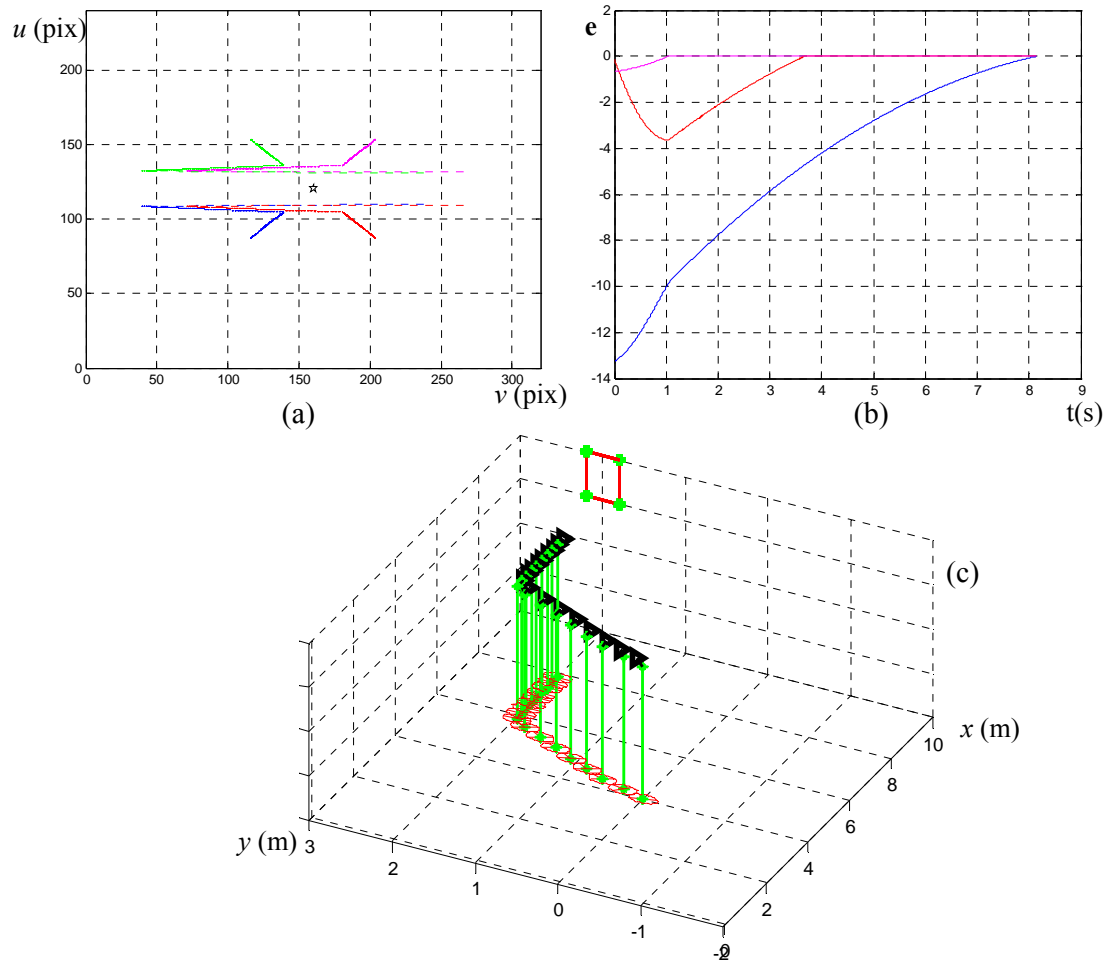


Fig. 4.6 : Test de simulation de l'asservissement visuel 2D par modes glissants :

(a) Trajectoires des indices visuels, (b) Evolution de la fonction de tâche

(c) Trajectoire du robot

Pour ce test, la cible est positionnée au point ayant comme coordonnées ($x = 10$ m, $y = 2$ m) dans le repère globale \mathbf{R}_o lié à la scène (voir figure 4.6.c). Dans ce même repère, le robot est positionné à la configuration initiale du test 1 de simulation du régulateur classique,

à savoir q_0 (4m, 0m, $\pi/6$ rad, 0 rad). La figure 4.6.a montre les trajectoires dans l'image des quatre points constituant la cible. Elle illustre clairement le fait que la cible est positionnée à la fin de la tâche sur le centre de l'image (l'étoile noire). Cette affirmation est consolidée par la trajectoire du robot montrée sur la figure 4.6.c, et en même temps par l'évolution des trois composantes de la fonction de tâche e qui convergent vers zéro une par une (voir figure 4.6.b) et signifiant la fin de la tâche. Cela indique clairement que la tâche de navigation a été correctement réalisée.

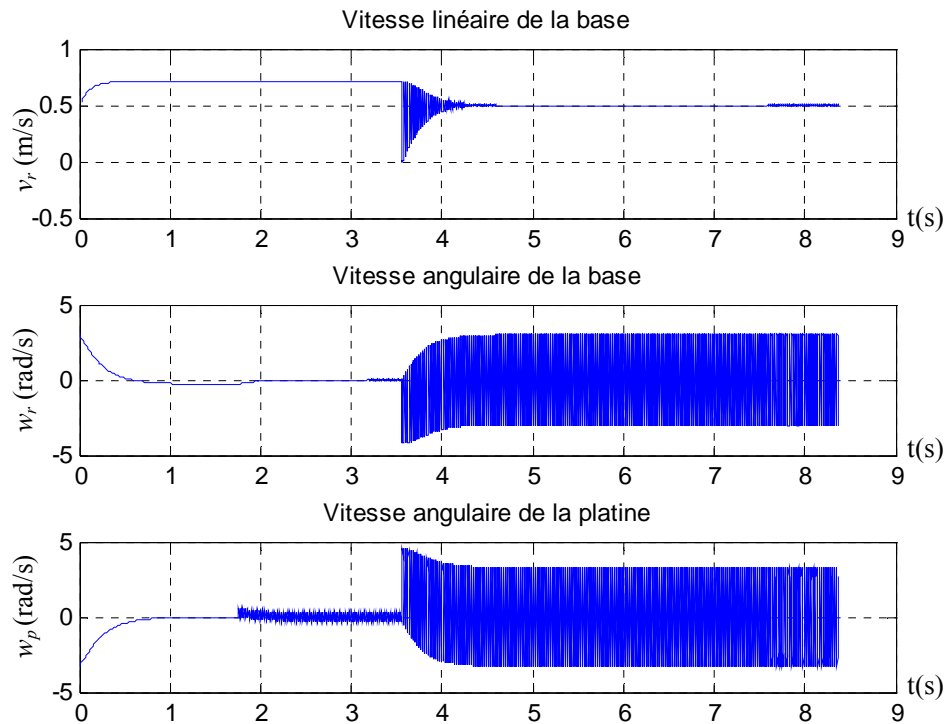


Fig. 4.7 : Variables de contrôle de l'asservissement visuel 2D par modes glissants

Concernant les grandeurs de contrôle du régulateur par modes glissants, présentées dans la figure ci-dessus, l'évolution des trois composantes se fait en deux phases distinctes :

- Phase 1 : Quand le robot se dirige vers une position face à la cible, les trois composantes de la commande ont des amplitudes bornées. Toutefois, un chattering apparait sur la vitesse envoyée à la platine, mais son amplitude est faible.
- Phase 2 : Quand le robot est en face de la cible et se dirige tout droit vers la position désirée en effectuant une simple translation, la vitesse v_r est presque constante jusqu'à la fin de la tâche. Toutefois, on constate la présence de chattering au début de cette phase, qui s'annule progressivement. Par contre, les vitesses de rotation présentent un

fort chattering, mais cela n'affecte pas la bonne conduite de la tâche car le robot converge correctement.

Les résultats des simulations nous ont permis de valider cette nouvelle approche d'asservissement visuel 2D basée sur les modes glissants. En effet, les tests effectués montrent que ce régulateur est aussi efficace que la commande classique, tout en assurant en théorie une meilleure stabilité. La présence de chattering est normale vu la nature même de la loi de commande, mais il existe dans la littérature plusieurs techniques qui permettent de le diminuer [21] [22]. Les performances de cette approche vont être vérifiées en expérimentation sur le robot *Rackham*, dans ce qui suit.

2.6 Implémentation expérimentale

Suite aux résultats obtenus en simulation, nous avons testé cette nouvelle technique de commande expérimentalement sur le robot *Rackham*. Identiquement au cas de la commande classique, nous avons rajouté une figure pour exposer l'évolution de l'erreur de positionnement ($E_p = v - v_d$). Aussi, la configuration désirée ne correspond pas à une position centrée dans l'image, et est désignée sommairement par une croix rouge (voir figure 4.8).

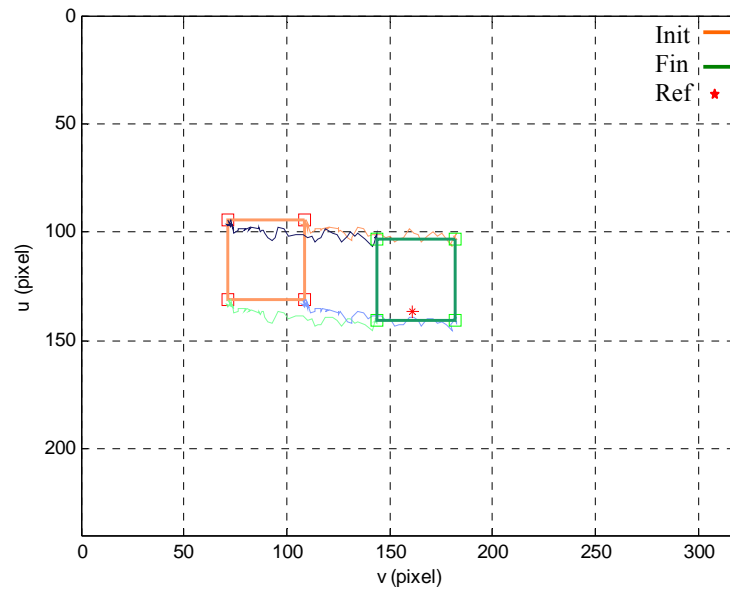


Fig. 4.8 : Trajectoires des indices visuels en expérimentation

L'évolution des indices visuels durant l'exécution de la tâche, exposée sur la figure 4.8, montre que celle-ci converge correctement. En effet, la position finale du visage en vert est presque confondue avec la configuration désirée (l'étoile rouge). Cela est confirmé avec

plus de précision sur la figure 4.9.b où E_p tend vers des valeurs très faibles (< 5 pix) à la fin de la tâche. De plus, les composantes de la fonction tâche convergent vers zéro comme il est montré sur la figure 4.9.a.

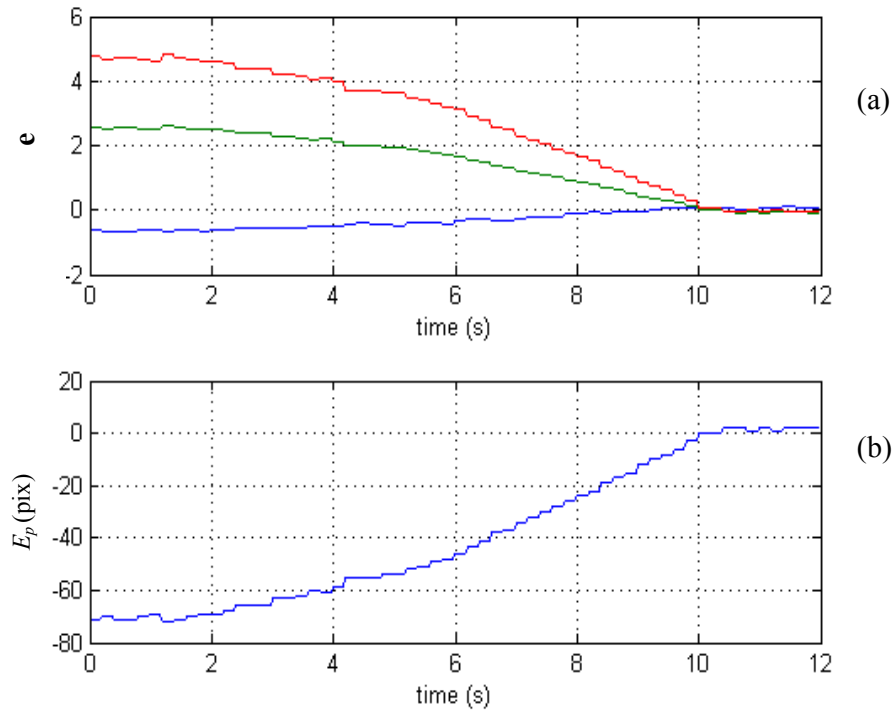


Fig. 4.9 : Test expérimental de l'asservissement visuel 2D par modes glissants :
 (a) Evolution de la fonction de tâche, (b) Erreur de positionnement de la cible

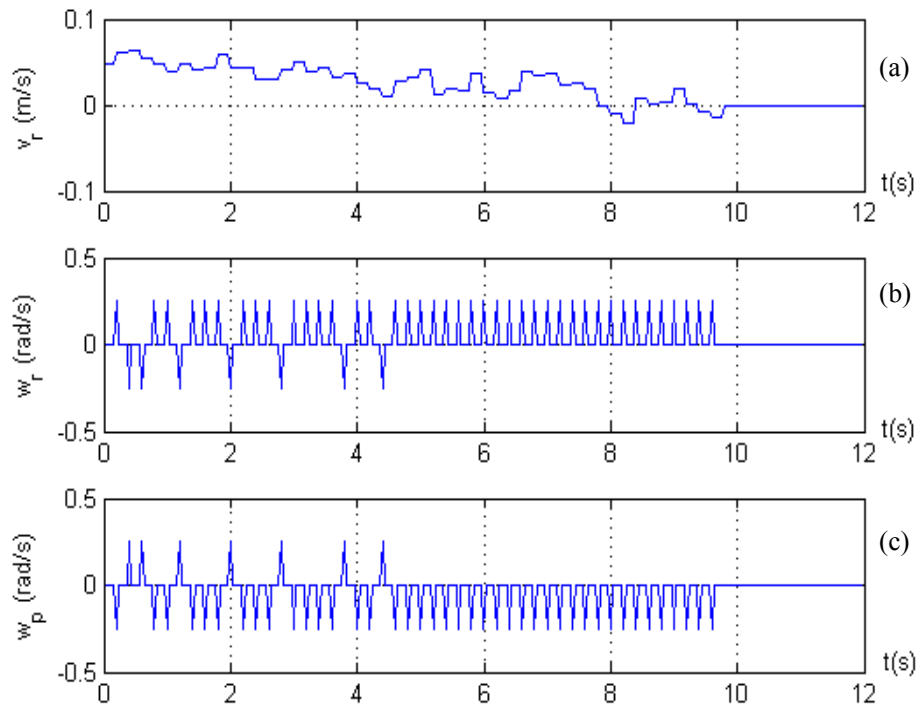


Fig. 4.10 : Variables de contrôle de l'asservissement visuel 2D en expérimentation

Concernant les commandes injectées au robot, elles sont de faibles amplitudes et s'annulent avec la convergence du motif visuel vers la position désirée. Toutefois, et comme en simulation, on note la présence de chattering essentiellement sur les vitesses de rotation. À travers les tests expérimentaux effectués dans cette étude, nous pouvons avancer que ce type de régulateur donne de bonnes performances en stabilité et précision. De plus, nous l'avons évalué pour des cas avec mouvements de la cible. Il a été constaté en expérimentation qu'il se comporte mieux que le régulateur classique, comme il est montré dans [87]. Enfin, la perte du signal visuel reste un cas d'échec pour ce type de commande basée uniquement sur l'information visuelle. Nous proposons dans la section suivante une solution à ce problème, en employant un autre type d'information sensorielle.

3. COMMANDE CLASSIQUE MULTIMODALE APPLIQUEE AU GUIDAGE & A LA POURSUITE D'UN UTILISATEUR

Dans le cadre du projet *Commrob* [25], l'objectif principal était de concevoir et de réaliser un prototype de caddy automatisé intelligent, capable de réaliser une multitude de tâches de manière complètement autonome. Pour le cas de notre étude, nos travaux se sont focalisés sur la réalisation de deux tâches spécifiques :

- *Suivi visuel de l'utilisateur par PTU (Guide me)* : Le robot doit être capable de garder le visage de l'utilisateur dans le champ de vision de sa caméra, quand il exécute des tâches de navigation autonome (e.g se diriger vers l'emplacement d'un produit).
- *Poursuite multimodale de l'utilisateur (Follow me)* : Le robot doit aussi pouvoir suivre l'utilisateur et rester près de lui dans les rayons du magasin, en utilisant une commande référencée capteurs (caméras + RFID).

3.1 Suivi visuel de l'utilisateur par PTU

3.1.1 Formulation du problème

Le robot *Rackham* étant équipé d'une PTU sur laquelle est montée la caméra, il est nécessaire d'implémenter des lois de commande des axes de la PTU, pour réaliser le suivi visuel et garantir que l'utilisateur demeure le plus possible dans le champ de vision de la caméra. Quand l'utilisateur est visible, on utilise la détection de visage effectué grâce au module *ICU* de détection et de reconnaissance des visages, développé au *LAAS-CNRS*. En absence d'information visuelle (l'utilisateur sort du champ de vision ou il est occulté par un

obstacle, etc.), nous employons alors les données fournies par le système RFID pour commander la platine afin d'orienter la caméra vers l'utilisateur, en vue de recouvrir le signal visuel. De cette manière, dès que le visage de l'utilisateur est détecté à nouveau, l'asservissement référencé vision reprend. Cet asservissement est réalisé à l'aide des lois de commande détaillées ci-après.

3.1.2 Lois de commande

Pour réaliser l'objectif de maintien de la visibilité de l'utilisateur, nous avons synthétisé deux régulateurs classiques de type PID pour la commande en vitesse de chacun des deux axes de la PTU. Ces régulateurs utilisent les informations fournies par *ICU*, à savoir les coordonnées (u, v) du centre de gravité du visage de l'utilisateur, et cela nous permet de définir la fonction d'erreur \mathbf{E} comme suit :

$$\mathbf{E} = \begin{bmatrix} E_u \\ E_v \end{bmatrix} = \begin{bmatrix} u - u_i \\ v - v_i \end{bmatrix} \quad (4.61)$$

où u_i et v_i représentent les coordonnées du centre de l'image.

Les deux régulateurs PID permettent de faire évoluer les composantes de \mathbf{E} vers zéro, tel que l'erreur en abscisse dans l'image, notée E_u , est régulée en agissant sur w_p , alors que l'erreur en ordonnée dans l'image, notée E_v , est annulée en agissant sur w_t . À partir de ces notations, les deux lois de commande employées sont données par :

$$\begin{cases} w_p = K_{pp} E_u + K_{ip} \int E_u dt + K_{dp} \frac{E_u}{dt} \\ w_t = K_{pt} E_v + K_{it} \int E_v dt + K_{dt} \frac{E_v}{dt} \end{cases} \quad (4.62)$$

avec $(K_{pp}, K_{ip}$ et $K_{dp})$ et $(K_{pt}, K_{it}$ et $K_{dt})$ sont respectivement les gains des deux régulateur PID des axes de la PTU.

Le modèle du système platine-caméra n'étant pas disponible, nous avons effectué le réglage des ces gains de façon expérimentale sans passer par une modélisation. Les tests de la phase expérimentale sont présentés ci-après.

3.1.3 Résultats expérimentaux

Nous avons évalué les performances de ces deux régulateurs par des tests en milieu intérieur. On mesure essentiellement la réponse du système et l'erreur des deux degrés de liberté de la caméra, pour une position initiale non centrée dans l'image. Les figures suivantes montrent les résultats obtenus pour un utilisateur statique.

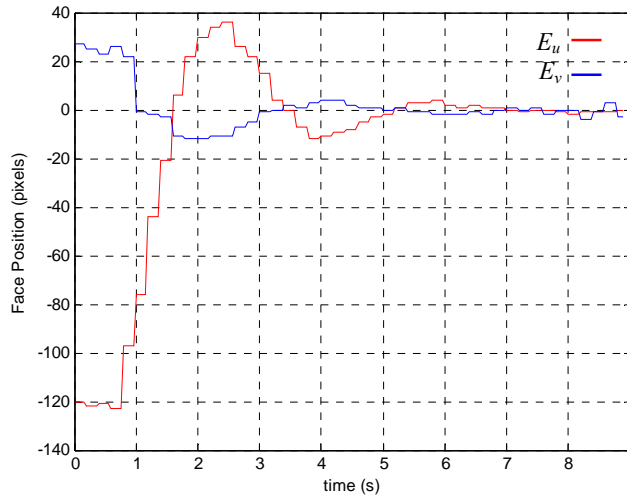


Fig. 4.11 : Erreurs de position du visage dans l'image

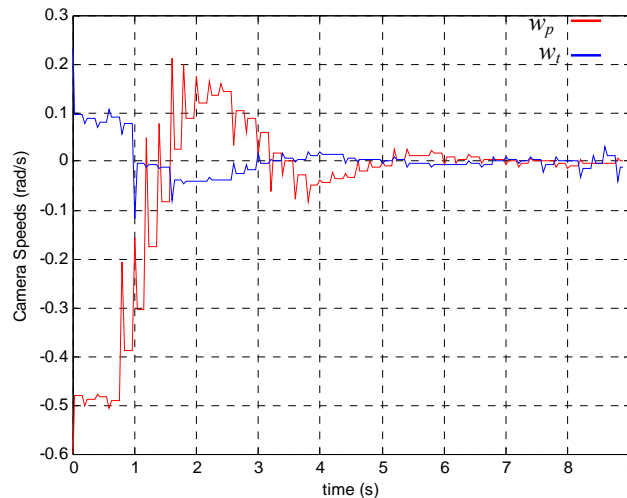


Fig. 4.12 : Vitesses des axes de la PTU

Nous estimons que ces performances sont satisfaisantes, car le temps de réponse du système est acceptable (convergence rapide, suivi parfait), ainsi que la précision de régulation. Ces tests nous ont permis de vérifier aussi que ces performances restent très bonnes, même dans le cas d'un utilisateur mobile. Dans la section suivante, nous allons exploiter les informations issues de l'algorithme multimodal de suivi de personne (vision + RFID) pour réaliser une poursuite d'un utilisateur mobile.

3.2 Poursuite multimodale de l'utilisateur

3.2.1 Formulation du problème

En plus de la caméra montée sur la PTU, le robot *Rackham* est équipé d'un système de détection sur 360° par RFID. Dans cette section, le but est d'implémenter des lois de commande pour permettre au robot de suivre l'utilisateur de manière efficace et en toute sécurité. Pour cela, la littérature propose diverses stratégies. Dans notre cas, où la caméra et le système RFID sont utilisés afin de détecter et reconnaître l'utilisateur, nous avons privilégié les approches basées sur la commande référencée capteurs comme dans [54].

Afin de simplifier la synthèse de la commande, nous avons choisi de dissocier les différents degrés de liberté de la caméra et du robot. Pour cela, nous avons analysé la structure mécanique de ce dernier. Celle-ci fait apparaître une redondance entre les deux entrées de commande w_r et w_p . Bien que, classiquement, cette propriété peut être utilisée pour réaliser un second objectif compatible avec la tâche de guidage (*section précédente*), nous avons choisi ici de ne pas considérer ce degré de liberté supplémentaire de façon à ce que la position horizontale de l'utilisateur dans l'image soit gérée par un seul correcteur. Nous avons de plus choisi de commander w_r plutôt que w_p afin de pouvoir orienter le robot tout entier vers l'utilisateur (et non pas uniquement la caméra), en vue d'améliorer la qualité d'exécution de la tâche robotique. Nous devons donc finalement synthétiser trois lois de commande (v_r , w_r , w_t), alors que w_p est fixée à zéro [52].

3.2.2 Lois de commande

Comme pour le cas du guidage d'utilisateur, les lois de commande développées reposent sur les informations fournies par l'algorithme de suivi de visage. Cependant, en plus des coordonnées (u, v) du centre de gravité du visage de l'utilisateur, nous employons le facteur d'échelle S fourni aussi par *ICU*, et qui caractérise vaguement la distance Homme/Robot [53]. Nous exploitons alors ces informations pour définir la fonction d'erreur E suivante :

$$E = \begin{bmatrix} E_u \\ E_v \\ E_S \end{bmatrix} = \begin{bmatrix} u - u_i \\ v - v_i \\ S - S_{soc} \end{bmatrix} \quad (4.63)$$

où S_{soc} représente la valeur du facteur d'échelle correspondant à une distance sociale acceptable, notée d_{soc} et fixée a priori.

L'objectif est maintenant de synthétiser les trois régulateurs PID permettant de réguler E à zéro, sachant que E_u représente l'erreur en abscisse dans l'image et peut être régulée à zéro en agissant sur la seule vitesse angulaire du robot w_r . De même, E_v correspond à l'erreur en ordonnée dans l'image et peut être progressivement annulée en pilotant la vitesse en tilt de la platine w_t . Enfin, E_s désigne l'erreur d'échelle sur laquelle on peut agir grâce à la vitesse linéaire du robot v_r [54]. A partir de ces notations, les lois de commande employées sont formulées par les expressions suivantes :

$$\begin{cases} v_r = K_{ps} E_s + K_{is} \int E_s dt + K_{ds} \frac{E_s}{dt} \\ w_r = K_{pu} E_u + K_{iu} \int E_u dt + K_{du} \frac{E_u}{dt} \\ w_t = K_{pv} E_v + K_{iv} \int E_v dt + K_{dv} \frac{E_v}{dt} \end{cases} \quad (4.64)$$

avec $(K_{ps}, K_{is}$ et $K_{ds})$, $(K_{pu}, K_{iu}$ et $K_{du})$ et $(K_{pv}, K_{iv}$ et $K_{dv})$ étant respectivement les gains des régulateur PID v_r , w_r et w_t . Comme au cas précédent, ils sont ajustés empiriquement pour garantir la stabilité de la boucle de commande et les performances souhaitées de l'asservissement.

Cependant, ces lois de commande ne peuvent être utilisées que lorsque l'utilisateur se trouve dans le champ de vision de la caméra. Quand son visage est perdu, les informations visuelles correspondantes ne sont plus accessibles, et cela nécessite de modifier la stratégie de commande envisagée. Pour cela, nous utilisons les informations fournies par le système RFID [52]. En effet, l'utilisateur d'intérêt portant un Tag RFID, cela permet de déterminer sa position en coordonnées polaires par rapport au repère du robot, à savoir, la distance robot-utilisateur notée d_{tag} (mesurée de manière grossière) ainsi que l'orientation notée θ_{tag} . La solution proposée est une technique semblable à celle utilisée dans le guidage, en vue de faciliter le recouvrement du signal visuel. Elle consiste à orienter la caméra vers le Tag RFID détecté, tout en rapprochant le robot de ce dernier, comme l'illustré la figure 4.13. En pratique, dès que l'utilisateur devient invisible, trois manœuvres sont enclenchées en même temps, à savoir [54] :

1. Translater le robot vers l'utilisateur en fonction des valeurs de d_{tag} et θ_{tag} , afin de garantir que ce dernier ne s'éloigne pas. De cette manière, nous essayons de continuer à satisfaire la contrainte sur la distance sociale d_{soc} , malgré la perte du signal visuel.

2. Tourner le robot à vitesse angulaire constante, jusqu'à ce qu'il soit orienté dans la bonne direction (vers le Tag RFID). Le sens de la rotation est déterminé à partir de la valeur de θ_{tag} .
3. Commander la PTU de telle sorte que son orientation en tilt φ_t soit ramenée à une configuration optimale pour le recouvrement du signal visuel.

Cet asservissement, noté \dot{q}_R , basé sur les informations RFID et la configuration du robot sert à réaliser des manœuvres simples et synchronisées, de manière à maximiser la probabilité de recouvrement du signal visuel. Il est exprimé alors par l'expression suivante :

$$\dot{q}_R = (v_r(d_{tag}, \theta_{tag}), w_r(\theta_{tag}), w_p(\varphi_t)) \quad (4.65)$$

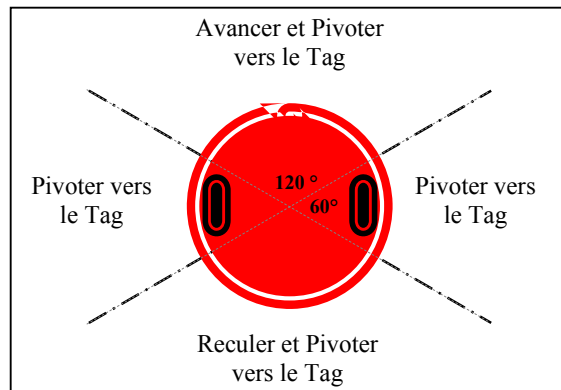


Fig. 4.13 : Manœuvres d'aide au recouvrement du signal visuel

En résumé, dès que le visage de l'utilisateur n'est plus dans l'image, le robot pivote et s'approche de lui s'il est détecté devant ou derrière, et pivote seulement si il est détecté à droit ou à gauche. Le robot ainsi se rapproche de l'utilisateur afin de le garder dans son voisinage, et faciliter la réinitialisation de l'algorithme de suivi de visage. Lorsque l'utilisateur est visuellement détecté à nouveau, la stratégie de commande rebascule automatiquement sur les PID référencés vision décrits plus haut dans l'équation (4.64).

Enfin, on note que la continuité de la loi de commande est assurée lors du changement de comportement entre le contrôle basé sur les RFID et celui basé sur la vision. En effet, la vitesse linéaire du robot est modifiée progressivement afin d'atteindre la valeur désirée. De même, la continuité des lois relatives w_r et w_t n'est pas gérée de manière explicite car elle correspondent à des systèmes suffisamment réactifs [53].

3.2.3 Résultats expérimentaux

Dans la cadre du projet *Commrob*, des expérimentations ont été réalisées dans une salle de 5x4 m² (voir figure 4.14), encombrée par d'autres personnes afin de valider l'approche proposée. Rappelons que notre objectif est de suivre un utilisateur à visage connue, et équipé d'un Tag RFID dans un lieu d'intérieur dynamique et encombré [52]. Les contraintes imposées sont :

- Le visage de l'utilisateur doit toujours être centré dans l'image quand il est visible,
- La distance sociale est fixée à $d_{soc} = 2m$, et doit être maintenue entre l'utilisateur et le robot tout au long de la tâche.

Pendant ces évaluations, et pour des raisons de sécurité, les vitesses linéaires et angulaires du robot ont été limitées respectivement à 0.4 m/s et 0.6 rad/s. Ces valeurs sont compatibles avec les déplacements de l'utilisateur [53]. La figure 4.14 décrit l'environnement dans lequel cette tâche a été testée, la trajectoire prédéfinie de l'utilisateur, ainsi que la trajectoire du robot résultant de l'exécution de la tâche robotique. L'orientation de la caméra est aussi indiquée sur la trajectoire du robot. De nombreuses séries de tests ont été effectuées selon le scénario suivant :

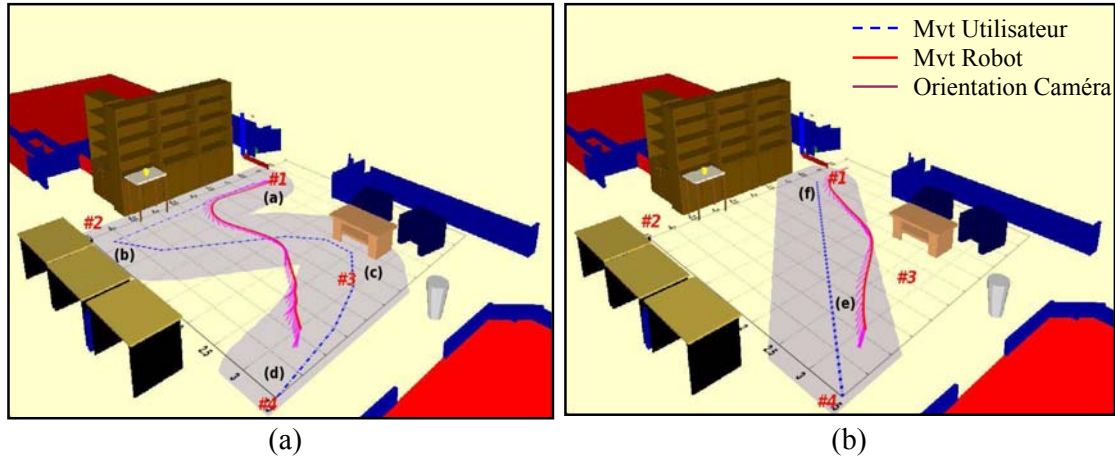


Fig. 4.14 : Exemple de tâche de poursuite multimodale d'utilisateur

- Phase 1 : l'utilisateur entre dans la salle et prend le Tag RFID déposé sur le robot stationné au point #1. Il va ensuite vers le point #2 pour se mettre en face du robot (figure 4.14.a). Dès que son visage est détecté, il se dirige vers le point #4 en passant par le point #3.

- Phase 2 : l'utilisateur est arrivé au point #4. Il va alors se retourner pour être en face du robot, puis il se redirige vers le point #1 en suivant une trajectoire rectiligne (figure 4.14.b).

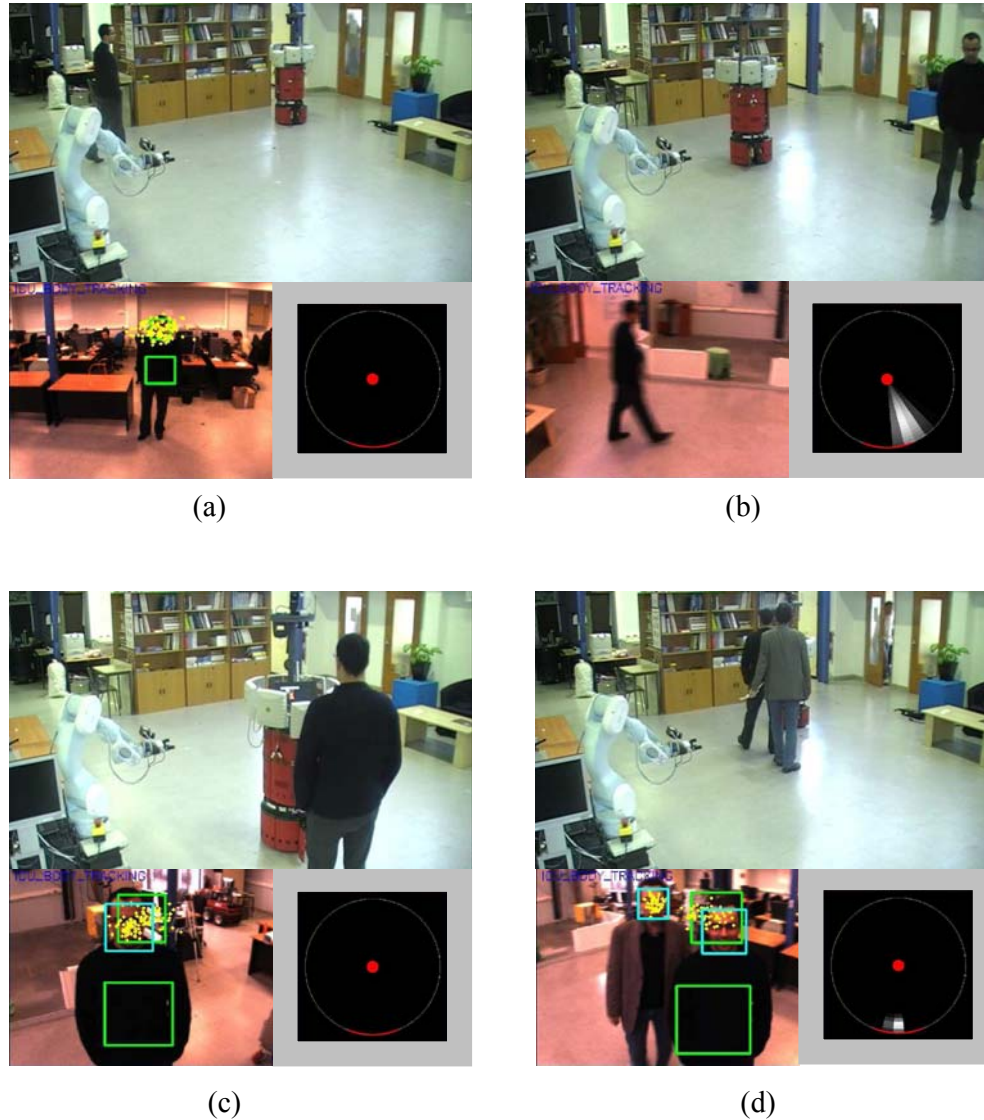


Fig. 4.15 : Séquences d'exécution de la poursuite d'utilisateur mobile par une commande multimodale

Durant cette tâche, le robot doit rester au voisinage de l'utilisateur en tentant de conserver au mieux la distance sociale. Nous montrons dans la figure 4.15 quatre Snapshot clés du flux vidéo associé à cette tâche. On expose alors à chaque fois une vue globale de la salle des tests (image du haut), l'image issue de la caméra du robot (image du bas à gauche),

ainsi que la cartes de saillance RFID (image du bas à droite) dans laquelle est indiquée la position Pan de la PTU (arc rouge).

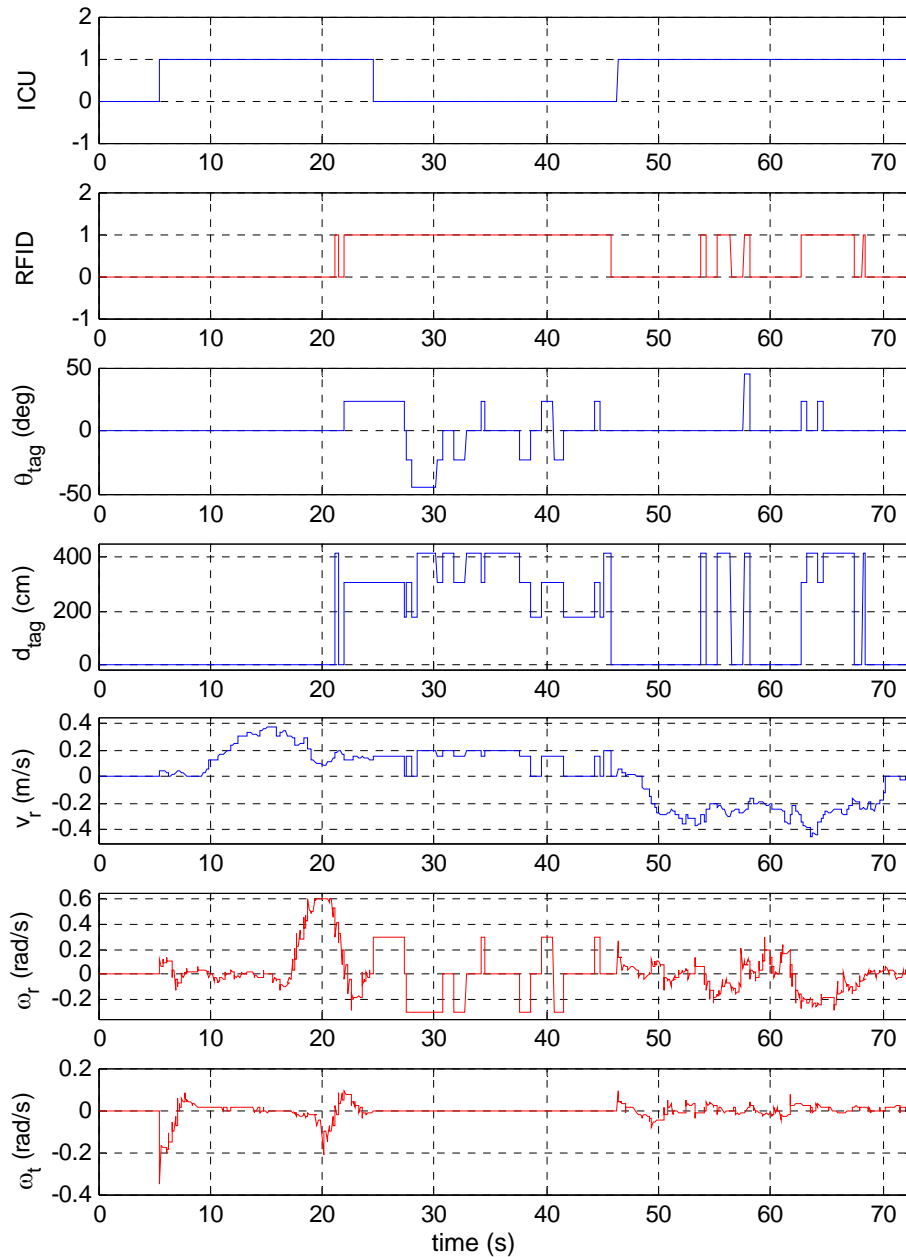


Fig. 4.16 : Signaux issus de la tâche de poursuite multimodale

L'initialisation de la mission est effectuée une fois le visage de l'utilisateur détecté et reconnu (figure 4.15.a). Le robot focalise alors son attention sur l'utilisateur pour centrer son visage dans l'image et commence à avancer vers lui grâce aux données visuelles. Dès que l'utilisateur se dirige vers le point #3, il reste présent dans l'image de la caméra mais son visage est perdu. Tout de même, le Tag RFID est détecté et le robot continue d'avancer en

direction de l'utilisateur (figure 4.15.b). Quand l'utilisateur arrive au point #4, le robot se dirige vers lui grâce aux informations RFID, jusqu'à ce que d_{tag} soit proche de d_{soc} . La tâche de poursuite est donc maintenue malgré la perte du signal visuel. Le système RFID guide le robot et la caméra afin de récupérer le contact visuel, et dès que l'utilisateur se retourne, son visage est détecté à nouveau et le contrôle visuel reprend (figure 4.15.c). A ce moment là, l'utilisateur commence à avancer pour aller au point #1, ce qui fait reculer le robot vers ce même point (figure 4.15.d) toujours par commande visuelle. La figure 4.14.b montre que les trajectoires du robot et de l'utilisateur vont dans la même direction durant cette phase. De manière globale, on peut dire que la tâche a été réalisée avec succès.

La figure 4.16 montre alors les différents signaux issus des modules *ICU*, *RFID* et *VisuServ* durant l'exécution de cette tâche, à savoir :

- Les deux flags nommés ICU et RFID, qui prennent la valeur 1 lorsque la cible est détectée respectivement dans l'image et par le système RFID et zéro dans le cas contraire,
- Les informations fournies par le système RFID (l'angle θ_{tag} et la distance d_{tag} mesurés),
- Les grandeurs de commande (v_r , w_r , w_t) calculées par le module *VisuServ* et envoyées aux actionneurs du robot.

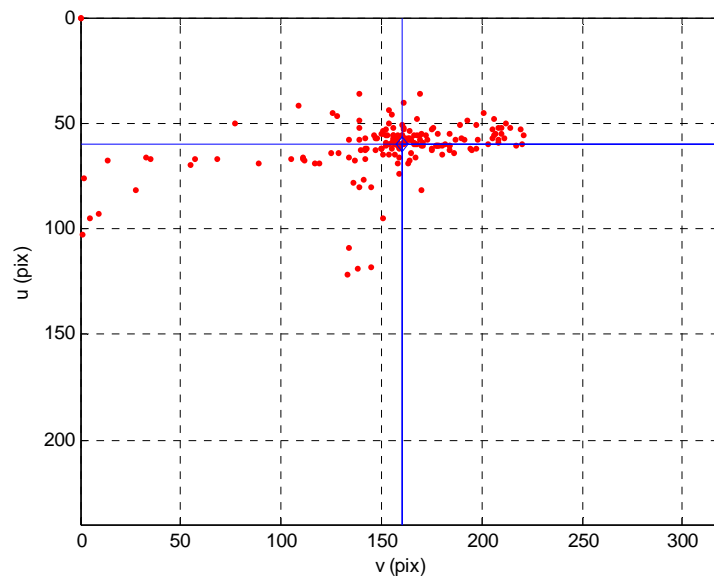


Fig. 4.17 : Positions du visage de l'utilisateur en poursuite multimodale

Nous présentons dans la figure 4.17 toutes les positions image du visage de l'utilisateur durant l'exécution de la tâche (points rouges). Dans la majorité des cas, le visage est proche de la configuration désirée désignée par l'intersection des deux droites bleues, ce

qui confirme une fois de plus l'efficacité des lois de commande référencées vision. Les quelques points positionnés à l'extrémité de l'image correspondent aux cas où l'utilisateur venait d'être détecté à nouveau par ICU, après avoir été perdu.

Pour une meilleure estimation des performances de la technique multimodale par rapport à celle utilisant uniquement la vision, nous avons procédé à une autre évaluation du système. Elle est basée sur un paramètre appelé le « Ratio de Contact Visuel » (RCV) défini par le rapport entre le temps où l'utilisateur est présent dans le champ de vision du robot et le temps total de la mission [53]. Cet indicateur mesure indirectement la robustesse de l'algorithme de suivi aux artefacts tels que les occultations ou les disparitions temporaires dues à la présence de passants. Le tableau 4.1 illustre cette évaluation, en considérant un nombre croissant de personnes présentes dans l'environnement du robot.

Tableau 4.1 : Ratio de contact visuel considérant 1 à 4 personnes

Perception utilisée	Nombre de personne au voisinage su robot								Total	
	1		2		3		4			
	μ_{rcv}	σ_{rcv}	μ_{rcv}	σ_{rcv}	μ_{rcv}	σ_{rcv}	μ_{rcv}	σ_{rcv}	μ_{rcv}	σ_{rcv}
Vision seule	0.21	0.11	0.22	0.02	0.18	0.05	0.22	0.06	0.21	0.04
Vision + RFID	0.94	0.08	0.84	0.14	0.94	0.13	0.83	0.19	0.86	0.14

On remarque que le RCV reste constant pour les deux cas de perception, mais que les résultats montrent une meilleure efficacité du système multimodal. En effet, l'introduction des données RFID permet de conserver la cible dans le champ de vision plus de $\mu_{rcv} = 80\%$ du temps de la tâche, et cela malgré la présence de personnes autour du robot (même pour le cas de 4 personnes). Toutefois, la valeur importante de l'écart-type (noté σ_{rcv}) est à notre avis due en grande partie aux déplacements aléatoires des passants, afin d'évaluer le système dans des conditions réalistes. On ne peut alors que remarquer que l'ajout des informations RFID à la vision permet d'améliorer la robustesse du système aux artefacts de l'environnement tels que les occultations longues, les sorties du champ de vue de la caméra, etc [53].

Suite aux résultats obtenus et jugés encourageants sur le robot *Rackham*, les mêmes expérimentations ont été reconduites sur le prototype de robot caddy *Inbot*. Ce dernier étant utilisé comme plateforme d'expérimentation principale dans le projet *CommRob*, l'aspect confidentiel du travail de certains partenaires du projet ne nous permet malheureusement pas de présenter en détail ces expérimentations. Il était notamment interdit de filmer les évaluations durant la présentation des scénarios [53]. Cependant, on présente ci-après une

rapide description des tests effectués dans le cadre du projet. Rappelons que l'objectif était d'effectuer deux tâches robotiques dans un environnement public encombré type grande surface, à savoir :

- 1- Guider un utilisateur novice vers un produit dans les rayons du supermarché (*Guide-Me*). Concrètement, l'utilisateur définit via l'interface tactile un produit à atteindre. Après planification du chemin vers ce produit, le robot commence sa mission en vérifiant périodiquement la présence de l'utilisateur dans l'image. Lorsque ce dernier s'éloigne du robot ($d_{soc} > 2m$), le robot s'arrête et l'attend. Dès que l'utilisateur s'approche à nouveau, le robot reprend son parcours en suivant la planification initiale.
- 2- Suivre l'utilisateur lors de ses déplacements dans les rayons (*Follow-Me*). Le robot réalise alors des mouvements pour rester dans le voisinage de l'utilisateur afin de conserver le plus souvent possible ce dernier dans son champ de vision. Lorsque l'utilisateur s'éloigne du robot, ce dernier s'avance vers lui afin de toujours conserver une distance sociale $d_{soc} < 2$ m. Lorsque le robot se trouve dans la zone d'interaction sociale, il s'arrête. Contrairement à *Rackham*, le robot *Inbot* ne recule pas lorsque l'utilisateur s'approche trop afin de lui permettre d'accéder au panier du caddy [53].

Enfin, une évaluation détaillée de ces deux tâches robotiques se trouve dans [53]. En résumé, la plupart des missions de guidage ont été réalisées avec succès. Parmi les échecs qui sont apparus, une partie est liée à un problème de planification ou de navigation (indépendant des thématiques traitées dans nos travaux). Dans le cas du suivi, 83% des missions ont été réalisées avec succès. Cependant, les cas d'échecs des deux tâches sont principalement dus à un problème de réinitialisation du suivi sur la base du RFID, car l'environnement de type supermarché augmente considérablement le nombre de fausses détections dues aux réflexions sur les structures métalliques environnantes [53].

4. COMMANDE ROBUSTE MULTIMODALE POUR LA POURSUITE D'UTILISATEUR

Nous avons jusque-là expérimenté deux types d'asservissement visuel 2D sur l'homme : l'un classique et l'autre basé sur les modes glissants, ainsi qu'une technique de commande classique multimodale à base de PID employée aussi pour le guidage et la

poursuite de l'homme. Les deux types d'asservissement visuel ont donné de bonnes performances : le premier est simple à implémenter et converge bien pour une cible statique, alors que le second est plus efficace en présence de perturbations (mouvements de la cible) mais ne s'applique pas facilement (problème du calcul des gains). Dans un premier temps, on propose de synthétiser un asservissement visuel qu'on appelle *Hybride*, fusionnant ces deux variantes en vue de bénéficier le plus possible de leurs avantages et obtenir de meilleures performances. C'est ce qui fera l'objet de la section suivante. En suite, on présente d'abord une autre variante plus complète de cet asservissement, employant à la fois l'information issue de l'image et les données RFID dans la boucle de commande, puis en dernier les résultats expérimentaux.

4.1 Asservissement Visuel Hybride

Notre objectif est de concevoir un algorithme de contrôle visuel applicable sur notre robot et capable de réaliser la tâche robotique tout en étant robuste et précis. Pour cela, nous proposons dans cette partie une nouvelle technique de commande référencée vision, que nous avons appelée « *Asservissement Visuel Hybride* ». Cette technique emploie à la fois les principes de base de l'asservissement visuel 2D classique, ainsi que les concepts propres de la commande par modes glissants.

En résumé, on définit une fonction de tâche $e(q,t)$ qui est en même temps utilisée comme surface de glissement pour le contrôleur à régimes glissants. À chaque itération, on tente de faire converger de deux manières différentes cette fonction vers zéro, suivant la configuration actuelle du système. En effet, si les gains calculés du contrôleur par modes glissants sont admissibles, on envoie alors ces commandes au robot. Dans le cas contraire, on utilise le contrôleur classique qui applique le principe de la décroissance exponentielle, afin de ramener la fonction de tâche à tendre vers zéro et assurer la bonne réalisation de la tâche robotique. En pratique, il s'agit de switcher entre les deux contrôleurs, ce qui donne une loi de commande hybride notée \dot{q}_H , dont l'expression est sous la forme :

$$\dot{q}_H = \mu \dot{q} + (1-\mu) \dot{q}_S \quad (4.66)$$

où \dot{q} et \dot{q}_S sont respectivement les deux contrôleurs référencés vision obtenus par asservissement visuel 2D classique (3.35), et asservissement visuel 2D par modes glissants (4.50).

Le paramètre μ permet alors de choisir entre les deux contrôleurs, suivant le résultat de la résolution du système d'inégalités décrivant la condition de stabilité du contrôleur à modes glissants. Par défaut, μ est égal à 0, et à chaque itération, si la solution \mathbf{K}_S obtenue du système (4.53) n'est pas admissible, μ est mis égal à 1, puis remis à 0 à l'itération suivante. Enfin, un choix judicieux des gains λ_i du contrôleur classique \dot{q} atténue considérablement l'effet de discontinuité sur \dot{q}_H , et qui résulte du basculement brutal d'un contrôleur à un autre [88].

Enfin, cette nouvelle technique de commande est prometteuse en termes de stabilité et de robustesse du contrôle, mais reste applicable uniquement quand l'utilisateur est présent dans le champ de vision de la caméra. Avant de la tester dans un contexte réel de tâche d'interaction Homme/Robot, nous avons choisi de l'améliorer encore d'avantage par l'utilisation des informations fournies par le système de détection RFID. C'est ce qui fait l'objet de la section suivante.

4.2 Asservissement Multimodal Hybride

Dans la dernière partie de ce travail, l'objectif final est la synthèse d'une technique de commande hybride référencée capteurs pour l'implémentation d'une tâche d'initialisation et de maintien d'interaction entre un utilisateur connu et le robot *Rackham*. D'un côté, cette technique est de nature multimodale combinant la vision par ordinateur à la détection par la technologie RFID, afin de combler la déficience du suivi visuel contre les occultations, les mouvements rapides de la cible, etc. De l'autre côté, elle sera basée à la fois sur la commande classique et les régimes glissants, en vue d'obtenir une convergence plus rapide et une meilleure robustesse vis-à-vis des perturbations et des erreurs de modélisation (erreur sur l'estimation de z , etc.).

De ce fait, le principe général de cette technique est décrit par l'organigramme de la figure 4.18. Quand l'utilisateur est visible ($\mu_V = 1$), l'asservissement visuel hybride développé dans la section 4.1 est utilisé. Dans le cas contraire ($\mu_V = 0$), si le Tag RFID est détecté ($\mu_R = 1$) alors on emploie l'asservissement basé sur les informations RFID conçu et testé pour la tâche de poursuite multimodale d'utilisateur dans la section 3.2. Enfin, si l'utilisateur n'est détecté par aucune modalité ($\mu_V = 0$ & $\mu_R = 0$), le robot arrête tout mouvement par mesure de

sécurité ($\dot{q} = 0$), et attend le retour d'un signal signifiant la présence de l'utilisateur (visuel ou RFID), comme illustré sur la figure suivante :

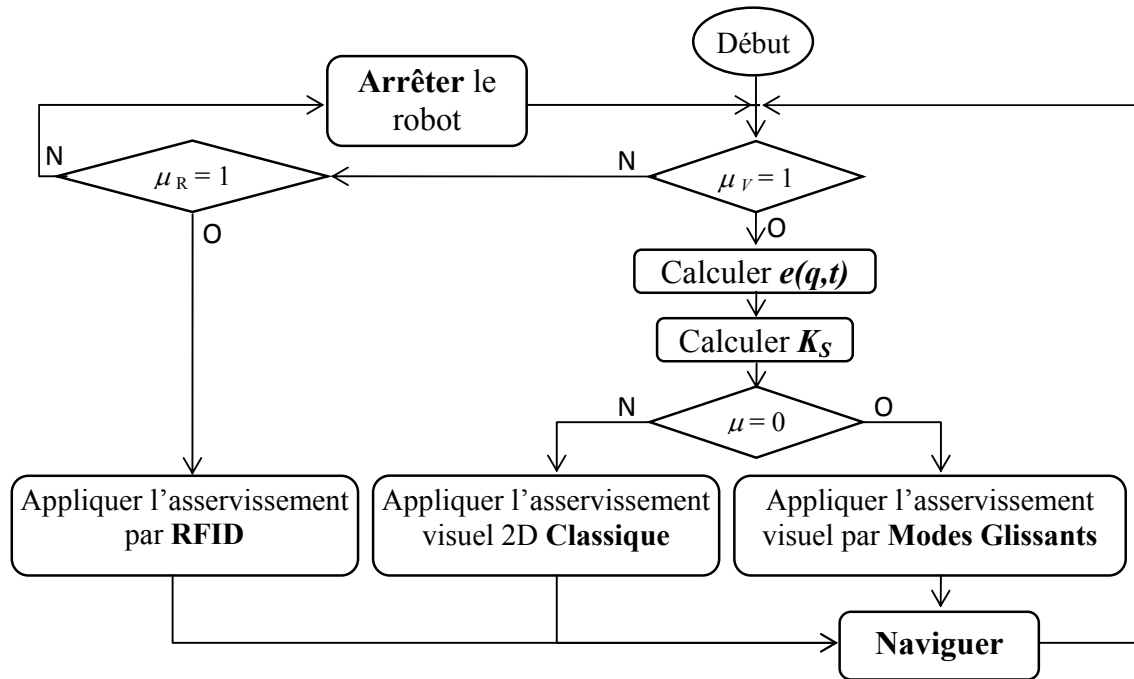


Fig. 4.18 : Principe de la poursuite d'utilisateur par asservissement multimodal hybride

De ce fait, l'expression du contrôleur \dot{q} de l'asservissement multimodal hybride est la suivante :

$$\dot{q} = \mu_V \dot{q}_H + \mu_R (1 - \mu_V) \dot{q}_R \quad (4.67)$$

On rappelle que \dot{q}_H et \dot{q}_R sont donnés respectivement par les expressions (4.66) et (4.65). Enfin, cette stratégie de commande a fait l'objet de multiples évaluations expérimentales pour une tâche d'interaction Homme/Robot. Les résultats de ces évaluations sont présentés dans ce qui suit.

4.3 Résultats expérimentaux

Nous exposons dans cette partie les résultats expérimentaux réalisés au *LAAS-CNRS* sur le robot *Rackham*. L'évaluation expérimentale de la stratégie globale de commande multimodale et hybride a été réalisée en deux phases. Dans la première phase, nous

considérons le cas d'une tâche d'interaction avec un utilisateur mobile, mais qui reste toujours dans le champ de vision de la caméra. L'objectif est d'évaluer en premier l'asservissement visuel hybride. Dans ce cas, nous n'utilisons pas encore les informations RFID. Dans la deuxième phase, nous considérons le cas d'une interaction avec un utilisateur libre de mouvement, afin de permettre de tester la stratégie globale. L'utilisateur porte alors un Tag RFID lors des expérimentations.

4.3.1 Cas d'un utilisateur visible

L'utilisateur est appelé dans ce cas à rester présent dans le champ de vision de la caméra, tout au long de la tâche d'initiation et de maintien de l'interaction Homme/Robot. Cela limite certes ses déplacements, mais permet tout de même de mieux évaluer les performances de la technique hybride par rapport aux deux autres techniques testées séparément. Nous présentons les résultats expérimentaux de la manière suivante : une figure montre le déroulement de la tâche (trajectoire du robot, trajectoires des indices visuels), une seconde figure pour reporter l'évolution des erreurs (fonction de tâche et erreur de positionnement), et une troisième figure pour les signaux de commande envoyés au robot ainsi que le type de contrôleur appliqué (classique ou modes glissants).

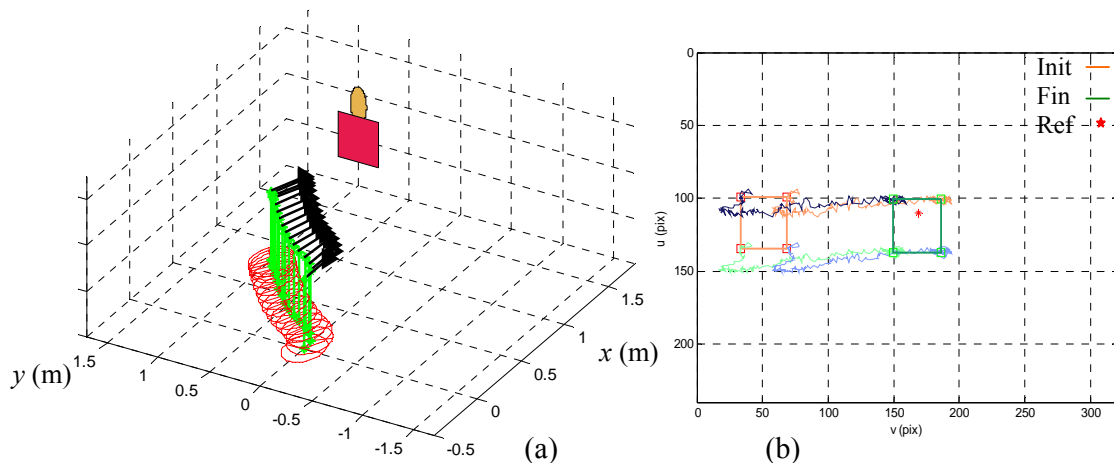


Fig. 4.19 : Test expérimental de l'asservissement visuel hybride
 (a) Trajectoire du robot, (b) Trajectoire des indices visuels dans l'image

La figure 4.19.a montre que le robot emprunte une trajectoire qui le mène progressivement vers l'utilisateur identifié. À la fin de la tâche, son visage est quasi centré dans l'image comme l'illustre la figure 4.19.b. Toutefois, les trajectoires des indices visuels

ne sont pas lisses, à cause des mouvements naturels de l'utilisateur, ainsi que le bruit sur la détection et la reconnaissance du visage.

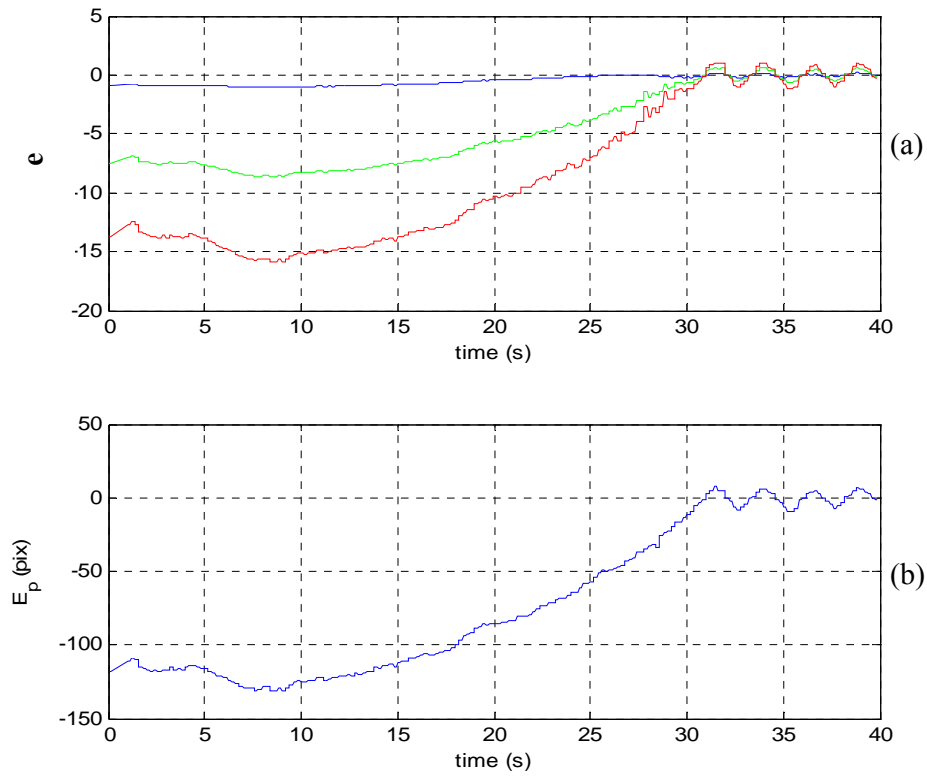


Fig. 4.20 : Evolution des erreurs en asservissement visuel hybride

Les composantes de la fonction de tâche tendent graduellement vers zéro, ainsi que pour l'erreur de positionnement E_p . Les oscillations apparues à la fin de la tâche sont dues essentiellement au basculement fréquent entre les deux lois de commande comme le montre la figure 4.21.a. Sur cette même figure, on constate que le régulateur à modes glissants fonctionne parfaitement bien, et fait converger le système vers l'utilisateur, jusqu'à ce que les gains prennent des valeurs inadmissibles lors de certaines itérations. Là, notre stratégie de commande utilise le contrôleur classique pour pallier à ce problème, et la tâche se termine correctement vu que le robot s'arrête à la position désirée.

Les amplitudes des grandeurs de commande restent bornées et admissibles, tout au long de l'exécution de la tâche (voir figure 4.21). Enfin, les oscillations présentes sur la vitesse linéaire du robot sont aussi dues aux changements fréquents de type de loi de commande, mais cela n'affecte en rien le bon déroulement de la tâche.

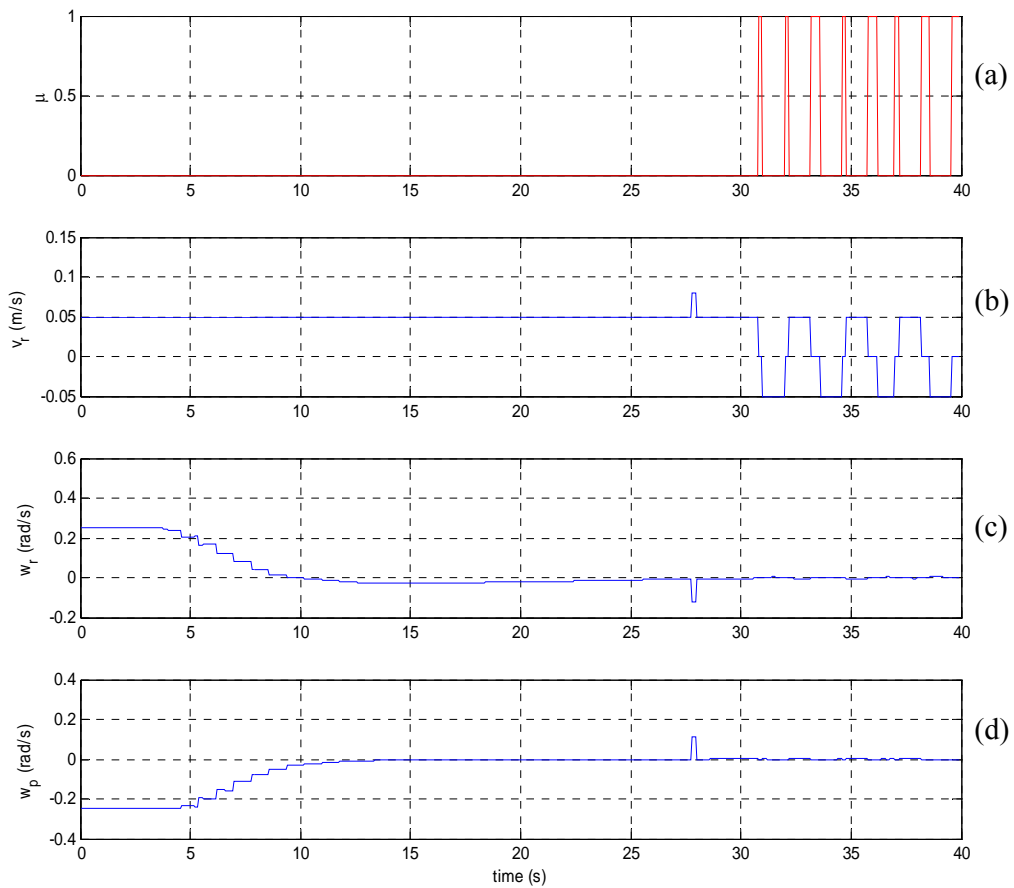


Fig. 4.21 : Signaux de commande en asservissement visuel hybride

4.3.2 Cas d'un utilisateur visible par moment

Pour ce cas plus réaliste, l'utilisateur bouge librement et peut quitter le champ de vision de la caméra durant le déroulement de la tâche d'interaction avec le robot. De plus, la condition de visibilité à l'état initial n'est plus nécessaire. Notre objectif est d'évaluer la qualité du comportement du robot en absence du signal visuel et ses capacités à le recouvrir. Cette façon de faire nous a permis de tester la stratégie de commande dans sa globalité (asservissement hybride et multimodale). Des résultats expérimentaux sont présentés ci-dessous où les deux premières figures sont similaires au cas précédent (déroulement de la tâche et erreurs de régulation). Par contre, la troisième figure consacrée aux signaux de la tâche contient l'évolution des commandes envoyées au robot ainsi que le type d'informations sensorielles employées par la stratégie de commande (visuelle ou RFID), et le type de contrôle appliqué (classique ou modes glissants).

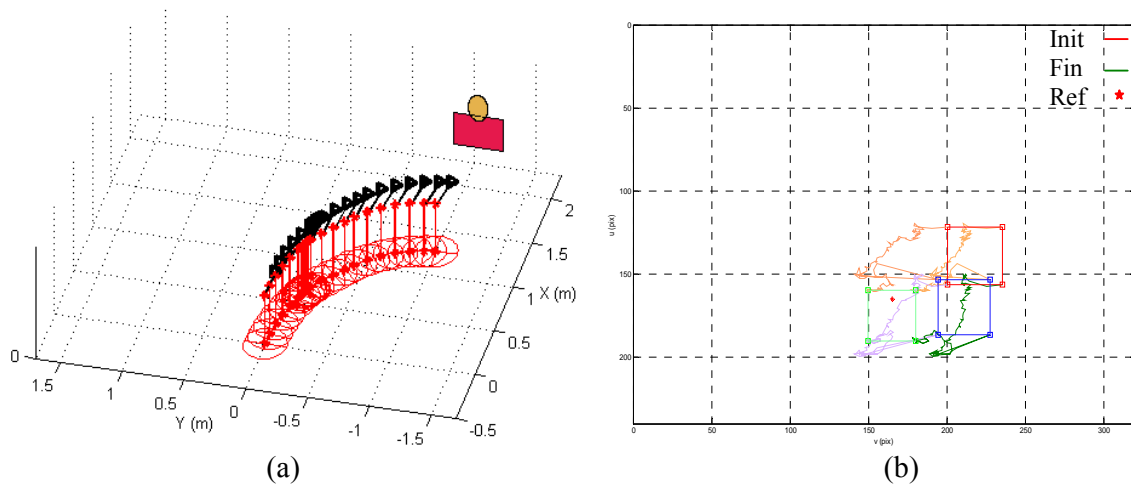


Fig. 4.22 : Test expérimental de l’asservissement multimodal hybride
 (a) Trajectoire du robot, (b) Trajectoire des indices visuels dans l’image

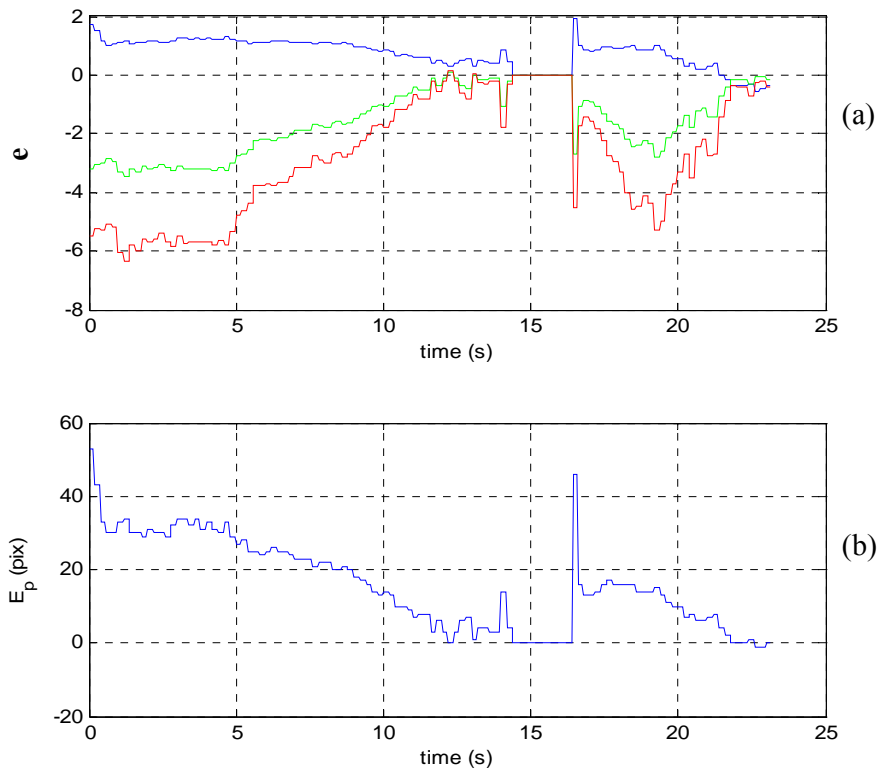


Fig. 4.23 : Evolution des erreurs en asservissement multimodal hybride

Pour ce test, la figure 4.22.a montre que le robot se dirige bien vers l’utilisateur identifié de manière correcte, bien qu’il s’arrête à un moment où le signal visuel est perdu. Cela est confirmé sur la figure 4.22.b, où on montre la position du visage de l’utilisateur. Initialement, il est positionné à droite (rectangle rouge) puis se trouve à la fin de la tâche

proche de la configuration désirée, désignée par une étoile. Le rectangle bleu montre la position du visage réapparu dans l'image, après que l'utilisateur ait été perdu par l'algorithme de suivi visuel du fait de ses déplacements. Cela signifie que le signal a été recouvert avec succès. L'évolution des composantes de la fonction de tâche e et de l'erreur de positionnement E_p confirme cela, du moment qu'elles sont toutes égales à zéro durant ce laps de temps (voir figure 4.23). Cependant, le bon déroulement de la tâche d'interaction n'a pas été affecté, et la stratégie de commande a réussi à faire converger vers zéro les grandeurs régulées, comme indiqué sur cette figure. De même que pour le cas de l'asservissement hybride, les indices visuels évoluent selon des trajectoires bruitées.

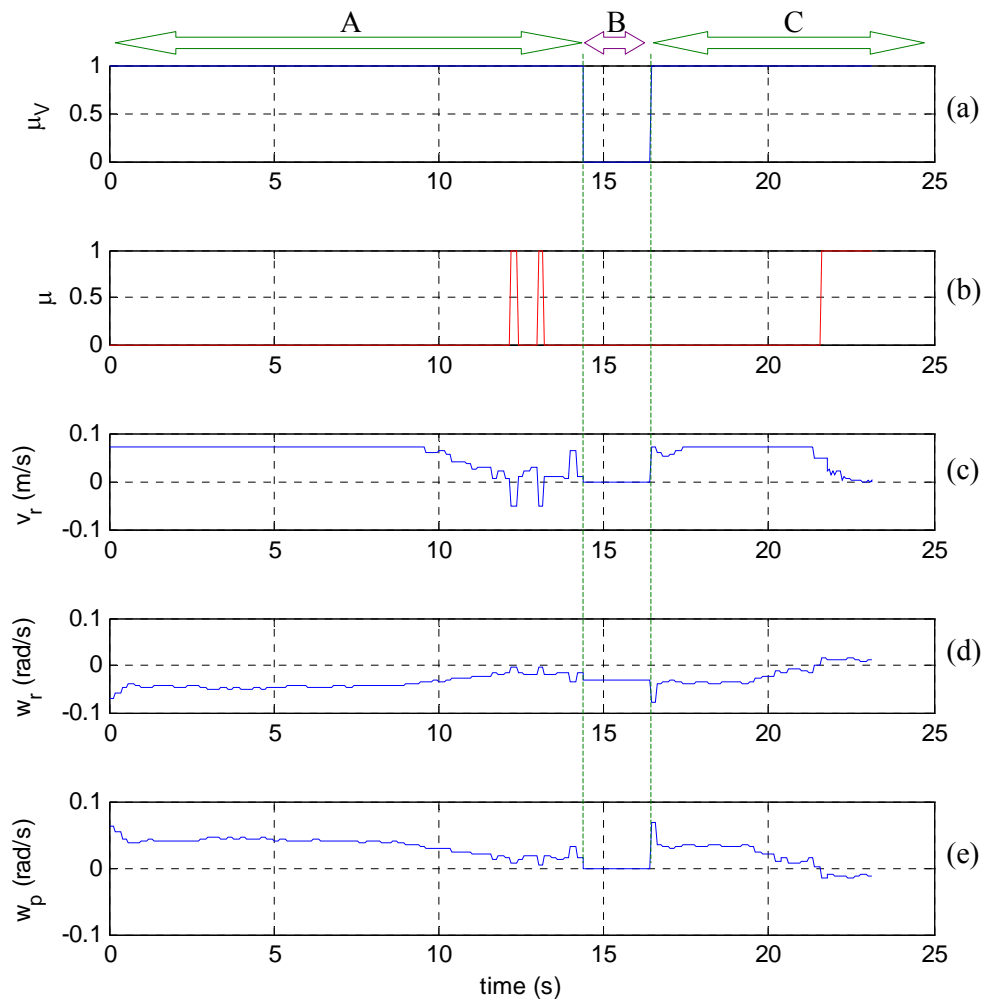


Fig. 4.24 : Signaux de commande en asservissement multimodal hybride

Les grandeurs de commande exposées dans la figure 4.24 évoluent de manières différentes durant les trois phases de l'exécution de la tâche :

- Phase « A » : Après initialisation de la tâche, l'utilisateur est visible (rectangle rouge), et le robot se dirige vers lui par asservissement visuel hybride. Les vitesses

diminuent au fur et à mesure qu'il se rapproche de sa cible, et les oscillations enregistrées sont dues essentiellement au Switch entre les deux lois de commande. Juste avant la convergence, l'utilisateur bouge et devient invisible.

- Phase « B » : l'utilisateur est hors du champ de vision, la translation du robot et la rotation de la PTU sont mises à zéro. Par contre, la rotation du robot est maintenue afin de l'orienter vers le Tag RFID détecté, en vue de retrouver le visage de l'utilisateur.
- Phase « C » : l'utilisateur est détecté à nouveau (rectangle bleu), et l'asservissement visuel hybride reprend et réussit à faire évoluer le robot correctement jusqu'à ce que l'erreur converge vers zéro. Les grandeurs de commandes évoluent de manière continue, et ont des valeurs admissibles qui s'annulent à la fin de la tâche.

A travers les expérimentations réalisées dans ce travail, on peut dire que la stratégie de commande Multimodale et Hybride proposée permet d'initier et de maintenir de manière efficace l'interaction entre le robot et l'utilisateur. D'un côté, la détection visuelle de l'utilisateur est améliorée significativement, car le détecteur de visage *ICU* prend en compte l'information RFID lors du calcul de la fonction de probabilité pour la reconnaissance (cas de plusieurs visages dans l'image). D'un autre côté, les données RFID se sont avérées très utiles lors de la perte du signal visuel. Le contrôleur basé RFID génère des consignes qui aident le robot à recouvrir ce signal, ce qui permet la reprise de l'interaction dans la majorité des cas. Enfin, la tâche d'interaction peut être lancée sans même que l'utilisateur visé ne soit initialement présent dans l'image. Cet avantage est précieux car il ne contraint pas l'utilisateur à se mettre en face du robot pour démarrer l'interaction, ce qui rend ce type de tâche plus utile et facilite son utilisation dans des scénarios réalistes. On peut alors imaginer de telle fonctionnalité dans plusieurs applications comme les robots d'aide à la personne (hôpitaux ou centres spéciaux), les robots guides (musée ou autre) ou les robots domestiques, etc.

Enfin, bien que les performances obtenues soient de qualité appréciable, il reste à compléter cette approche par de nouvelles fonctionnalités (e. g. évitement d'obstacles), et améliorer certains aspects comme nous le mentionnons dans la conclusion générale de ce manuscrit.

5. CONCLUSION

Ce chapitre a été consacré à la présentation de l'évaluation des techniques robustes et multimodales de commande référencées capteurs pour l'interaction Homme/Robot. En effet, après avoir rappelé les concepts théoriques de la commande par modes glissants, nous avons synthétisé un asservissement visuel dit *Hybride*, combinant ce type de commande réputé robuste à la commande classique développée au chapitre précédent. En suite, et pour pallier aux déficiences de la vision artificielle, nous avons introduit un autre moyen de perception qu'est le capteur RFID. Dans ce cadre, nous avons proposé deux stratégies de commande : une première dite classique testée puis validée sur le robot *Inbot* pour le suivi d'utilisateur, puis une seconde stratégie Multimodale et Hybride capable d'initier et de maintenir une tâche d'interaction entre un utilisateur connu et le robot *Rackham*. Les expérimentations ont montré que cette dernière stratégie est efficace et plus robuste, car elle permet de maintenir l'interaction même en cas de perte du signal visuel. Enfin, le recouvrement de ce signal est facilité par le recours au contrôleur référencé RFID.

CHAPITRE 5

CONCLUSION GÉNÉRALE & PERSPECTIVES

Nous nous sommes intéressés, à travers les travaux de recherche menés dans le cadre de cette étude, à la synthèse et l'implémentation de techniques de commande référencées capteurs sur une plateforme de robotique mobile de service, pour la réalisation d'une tâche d'interaction Homme/Robot. Plus concrètement, nos travaux visent à doter le robot de capacités lui permettant d'identifier son utilisateur, de conserver tant que possible un contact visuel avec lui et de le suivre (ou de le guider) dans un environnement d'intérieur dynamique et encombré. Afin de mieux répondre à cette problématique, présentée dans le chapitre 1, nous avons ainsi défini un ensemble de schémas de commande intégrant toutes les informations sensorielles pertinentes pour cette tâche robotique. L'aspect réactif, vis-à-vis des événements extérieurs perturbant le déroulement de la tâche, est au cœur de nos préoccupations et constitue un aspect fondamental de nos travaux. À ce titre, nous avons, en premier lieu rappelé, dans le chapitre 2 les notions essentielles sur lesquelles s'appuient nos stratégies de

navigation référencées capteurs, telles que le formalisme des fonctions de tâche. Sur cette base, nous avons synthétisé dans le chapitre 3 un asservissement visuel 2D classique, que nous avons testé en simulation puis expérimentalement sur le robot *Rackham* du *LAAS-CNRS*.

Par la suite, une technique d'asservissement visuel 2D par les régimes glissants a été proposée dans ce chapitre 4, en vue d'apporter une meilleure robustesse vis-à-vis des mouvements de l'utilisateur ciblé durant l'interaction. Cette nouvelle approche a aussi été d'abord testée en simulation, puis expérimentalement sur le robot. Suite aux résultats obtenus, nous avons proposé une autre variante de cette technique que nous avons appelé « *Asservissement Visuel Hybride* », mieux adaptée à notre système car elle combine la commande visuelle classique développée au chapitre 2 avec l'asservissement visuel robuste basé sur les régimes glissants.

Cependant, cette technique permet au robot d'interagir avec l'utilisateur uniquement quand il est visible. Il a été alors jugé nécessaire de l'améliorer en exploitant d'autres informations sensorielles, autres que celles fournies par la caméra. Pour cela, nous avons introduit les données du système de détection par RFID disponible sur le robot, en proposant dans le chapitre 4 deux stratégies de commande multimodale, utilisant les données visuelles et les informations RFID. Nous avons alors dans le cadre du projet *CommRob* [47] proposé une première variante classique basée sur des contrôleurs PID, et capable de réaliser le guidage et la poursuite d'utilisateur mobile dans un environnement de type Supermarché.

Dans un autre contexte applicatif, une deuxième stratégie de commande « *Robuste* » et « *Multimodale* » a été proposée en vue d'initier et de maintenir une tâche d'interaction avec un utilisateur connu, même en cas d'absence ou perte du signal visuel. Elle est basée sur l'asservissement visuel hybride et utilise aussi les données fournis par le système de détection RFID. Quand l'utilisateur est visible, le robot est piloté par l'asservissement visuel hybride alors que dans le cas contraire, un contrôleur basé sur les informations RFID prend le relais et contribue au maintien du robot au voisinage de l'utilisateur en vu de recouvrir le signal visuel [36]. Les expérimentations sur le robot *Rackham* ont permis de constater que la reprise du contact visuel est plus probable, grâce à l'exploitation des données RFID. Cela nous permet de conclure que la stratégie proposée est très prometteuse, et qu'en procédant à l'amélioration de certains aspects, il est plus que probable de l'utiliser dans un cadre commercial (en robotique de service).

Enfin, l'ensemble de nos travaux ont été valorisés par la parution de 3 articles de revues de renommée (dont 1 en collaboration), et la présentation de 4 articles de conférences internationales (dont 1 en collaboration) et 1 article de conférence nationale. De plus, ces

travaux ont permis d'ouvrir un certain nombre de perspectives intéressantes sur divers axes de recherche. Nous soulevons ici quelques unes :

- L'introduction de la commande à modes glissants a permis d'améliorer la robustesse du schéma de commande, mais le problème du chattering mérite sérieusement d'être traité afin de permettre une meilleure utilisation de cette commande sur les robots.
- Le contrôleur robuste utilisé dans cette étude rentre dans la catégorie des modes glissants classiques. Il serait alors intéressant d'étudier la possibilité d'employer d'autres types de contrôleurs, plus évolués et connus pour être plus efficaces (modes glissants d'ordre supérieur).
- L'ajout de fonctionnalités complémentaires, comme l'évitement d'obstacles dynamiques à la stratégie multimodale proposée procurera certainement une meilleure efficacité, et permettra une large applicabilité.

L'utilisation d'informations sensorielles complémentaires provenant de capteurs déportés paraît très intéressante pour ce type de tâche. Cela peut apporter une richesse d'information sur l'environnement, sans pour autant mobiliser de manière importante les ressources de calcul disponibles sur le robot.

REFERENCES

- [1] Alami, R., Chatila, R., Fleury, S., Ghallab, M., Ingrand, F. An architecture for autonomy. 1998, *International Journal of Robotic Research*, Special Issue on *Integrated Architectures for Robot Control and Programming*, vol. 17(4) pp 315-337.
- [2] Anne, M., Crowley, J. L., Devin, V., Privat, G. Localisation intra-bâtiment multi-technologies: RFID, Wifi, vision. *2nd French-speaking conference on Mobility and ubiquity computing (Ubimob'05)*, Grenoble, France, Mai 31 - Juin 3, 2005.
- [3] Becera, H. M., Sagues, C. A Sliding Mode Control Law for Epipolar Visual Servoing of Differential-Drive Robots. *IEEE International Conference on Intelligent Robots and Systems*, Nice, France, Septembre 22-26, 2008.
- [4] Bellot., D. Contribution à l'analyse et à la synthèse de schémas de commande référencés vision. 2002, *Thèse de doctorat, LAAS-CNRS, Université Paul Sabatier, Toulouse, France*.
- [5] Bensalah, F., Chaumette, F. Compensation of abrupt motion changes in target tracking by visual servoing. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95)*, Pittsburgh, USA, Août 5-9, 1995.
- [6] Béthes, L., Lerasle, F., Danès, P. Particle filtering strategies for visual tracking dedicated to h/r interaction. *International Workshop on Vision Based Human-Robot Interaction*, Palerme, Italie, Mars 17, 2006.
- [7] Bouguet, J. Y. Camera Calibration Toolbox for Matlab. Matlab Toolbox. Page web : http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [8] Bourquardez, O., Chaumette, F. Visual Servoing of an Airplane for Auto-Landing, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07)*, San Diego, USA, Octobre 29 - Novembre 02, 2007.
- [9] Cadenat., V. Commande référencée multi-capteurs pour la navigation d'un robot mobile. 1999, *Thèse de doctorat, LAAS-CNRS, Université Paul Sabatier, Toulouse, France*.
- [10] Chaumette, F. La relation Vision-Commande : Théorie et application à des tâches robotiques. 1990, *Thèse de doctorat. Université de Rennes I, Rennes, France*.
- [11] Chaumette, F., Santos, A. Tracking a moving object by visual servoing. *12th World congress IFAC*, Sydney, Australie, Octobre 1993 .
- [12] Chaumette, F. De la perception à l'action : l'asservissement visuel, de l'action à la perception : la vision active. 1998, *HDR, Université de Renne I*.

- [13] Chaumette, F. Potential problems of stability and convergence in image-based and position-based visual servoing. 1998, In Hager, G., Kriegman, D., Morse, A. The confluence of vision and control, LNCIS Series, vol. 237 pp 66-78, *Springer-Verlag*.
- [14] Chaumette, F., Marchand, E. A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing. 2001, *IEEE Transaction on Robotics and Automation*, vol. 17(5), pp 719-730.
- [15] Chaumette, F. Asservissement visuel. dans Khalil, W. La commande des robots manipulateurs 2002, éditions Hermès.
- [16] Chaumette, F. Image moments: a general and useful set of features for visual servoing. 2004, *IEEE Transactions on Robotics and Automation*, vol. 20(4) pp 713–723.
- [17] Chen, Z., Birchfield, S. T. Person Following with a Mobile Robot using Binocular Feature-Based Tracking, *IEEE International Conference on Intelligent Robots and Systems (IROS'07)*, San Diego, USA, Octobre 29 - Novembre 02, 2007.
- [18] Chesi, G. Vicino, A. Visual servoing for large camera displacements. 2004, *IEEE Transaction on Robotics and Automation*, vol. 20(4), pp 724-735.
- [19] Chesi, G. Visual servoing path-planning via homogeneous forms and LMI optimizations. 2009, *IEEE Transaction on Robotics and Automation*, vol. 25(2) pp 281-291.
- [20] Chesi, G., Hashimoto, K. Visual Servoing via Advanced Numerical Methods. 2010, *édition Springer-Verlag*.
- [21] Chettouh, M., Toumi, R., Hamerlain, M. High-order sliding modes for a robot driven by pneumatic artificial rubber muscles. 2008, *Advanced Robotics*, vol. 22 pp 689-704.
- [22] Chettouh, M., Commande d'un robot à muscles artificiels par les régimes glissants, et réduction du broutement. 2008, *Thèse de doctorat. Université des Sciences et de la Technologie Houari Boumediene (USTHB), Bab-Ezzouar, Alger, Algérie*.
- [23] Cielniak, G., Lilienthal, A., Duckett., T. Improved data association and occlusion handling for vision based people tracking by mobile robots. *International Conference on Intelligent Robots and Systems (IROS'07)*, San Diego, USA, Octobre 29 - Novembre 02, 2007.
- [24] Collewet, C., Marchand, E. Photometry-based visual servoing using light reflexion models. *IEEE International Conference on Robotics and Automation (ICRA'09)*, Kobe, Japon, Mai 12-17, 2009.
- [25] Communication with and among robots: Commrob. www.commrob.eu

- [26] Comport, A. I., Marchand, E. Chaumette, F., Pressigout, E. A Visual Servoing Control Law that is Robust to Image Outliers, *IEEE International Conference on Intelligent Robots and Systems*, Las Vegas, USA, Octobre 27 - 31, 2003.
- [27] Comport, A. I., Marchand, E. Chaumette, F. Statistically robust 2d visual servoing. 2006, *IEEE Transaction on Robotics and Automation*, vol. 22(2) pp 415–421.
- [28] Corke, P.I., Goods, M.C. Controller design for high performance visual servoing. 12th *World congress IFAC*, Sydney, Australie, Octobre 1993 .
- [29] Corke, P. I. Visual control of robots. 1996, *Research Studies Press Ltd*.
- [30] Corke, P. I., Good, M.C. Dynamic effects in visual closed-loop systems. 1996, *IEEE Transaction on Robotic and Automation*, vol. 12(5) pp 671-683.
- [31] Crétual, A., Chaumette, F. Positioning a camera parallel to a plane using dynamic visual servoing. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97)*, Grenoble, France, Septembre 7-11, 1997.
- [32] Crétual, A., Chaumette, F. Image-based visual servoing by integration of dynamic measurements. *IEEE International Conference on Robotics and Automation (ICRA'98)*, Louvain, Belgique, Mai 16-20,1998.
- [33] Crétual, A. Asservissement visuel à partir d'information de mouvement dans l'image. 1998, *Thèse de doctorat, Université de Rennes I, Rennes, France*.
- [34] Crétual, A., Chaumette, F., Sandini, G. Image-based positionning with respect to a non-structured scene using 2D image motion. *IEEE International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, 31 Octobre-05 Novembre, 2000.
- [35] Danès, P., Coutinho, D. F. Multicriteria Analysis of Visual Servos via Biquadratic Lyapunov Functions. *5th IFAC Symposium on Robust Control Design*, Toulouse, France, Juillet 05-07, 2006.
- [36] Dornaika, F. Contributions à l'intégration vision/robotique : calibrage, localisation et asservissement. 1995, *Thèse de doctorat, Institut National Polytechnique de Grenoble, France*.
- [37] Emelyanov, S. V. Taran, V. A. Sur une classe de systèmes de régulation automatique à structure variable. 1962, *Journal de l'académie des sciences d'URSS, Énergétique et Automatique*, vol. 3.
- [38] Espiau, B., Chaumette, F., Rives, P. A new approach to visual servoing in robotics. 1992, *IEEE Transactions on Robotics and Automation*, vol. 8(3) pp 313-326.
- [39] Espiau, B. Effect of camera calibration errors on visual servoing in robotics. 3rd *International Symposium on Experimental Robotics*, Kyoto, Japan, October 28-30, 1993.

- [40] Fang, Y., Dixon, W. E., Dawson, M., Shawda, P. Homography based visual servo regulation of mobile robot. 2005, *IEEE Transaction on System, Man and Cybernetics*, vol. 35(5), pp 1041-1050.
- [41] Feddema, J., Mitchell, O. Vision-guided servoing with feature-based trajectory generation. 1989, *IEEE Transactions on Robotics and Automation*, vol. 35(12) pp 2323-2327.
- [42] Filippov, A. F. Application of theory of differential equations with discontinuous right hand side to non linear control problems. *1st IFAC Congress, Izd. ANSSSR*, 1960.
- [43] Fleury, S., Herrb, M., Mallet., A. GenoM: Generator of Modules for Robots. <http://softs.laas.fr/openrobots/tools/genom.php>, 2005.
- [44] Folio, D., Cadenat, V. A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment, *44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Espagne, Décembre 12-15, 2005.
- [45] Folio, D. Stratégies de commande référencées multi-capteurs et gestion de la perte du signal visuel pour la navigation d'un robot mobile. 2007, *Thèse de doctorat, LAAS-CNRS, Université Paul Sabatier, Toulouse, France*.
- [46] Folio, D. Cadenat, V. Treating image loss using the vision/motion link: a generic framework, Dans *Computer Vision Book, éditions IN-TECH*.
- [47] Gangloff, J., Asservissements visuels rapides d'un robot manipulateur à six degrés de liberté : suivi de cible et de profilé. 1999, *Thèse de doctorat. Ecole Nationale Supérieure de Physique de Strasbourg, France*.
- [48] Gao., B. Contribution à la synthèse de commandes référencées vision 2D multicritères. 2006, *Thèse de doctorat, LAAS-CNRS, Université Paul Sabatier, Toulouse, France*.
- [49] Germa, T., Béthes, L., Lerasle, F. Simon, T. Data fusion and eigenface based tracking dedicated to a tour-guide robot. *International Conference on Vision Systems (ICVS'07)*, Bielefeld, Germany, Mars 21-24, 2007.
- [50] Germa, T., Lerasle, F., Danès, P., Béthes, L. Human/robot visual interaction for a tour-guide robot. *International Conference on Intelligent Robots and Systems (IROS'07)*, San Diego, USA, Octobre 29 - Novembre 02, 2007.
- [51] Germa, T., Lerasle, F., Ouadah, N., Cadenat, V. Vision and RFID-based Person Tracking in Crowds from a Mobile Robot, *IEEE International Conference on Intelligent Robots and Systems (IROS'09)*, Saint Louis, USA, Octobre 11-15, 2009.

- [52] Germa, T., Lerasle, F., Ouadah, N., Cadenat, V., Lemaire, C. Fusion de données visuelles et RFID pour le suivi de personnes en environnement encombré depuis un robot mobile. *Reconnaissance des Formes et Intelligence Artificielle (RFIA '2010)*, Caen, France, Janvier 19-22, 2010.
- [53] Germa, T. Fusion de données hétérogènes pour la perception de l'homme par un robot mobile. 2010, *Thèse de doctorat. LAAS-CNRS, Université Paul Sabatier, Toulouse, France.*
- [54] Germa, T., Lerasle, F., Ouadah, N., Cadenat. Vision and RFID data fusion for tracking people in crowds by a mobile robot. 2010, *Computer Vision and Image Understanding*, vol. 114(6), pp 641-651.
- [55] Gockley, R., Forlizzi, J. Simmons, R. Natural Person-Following Behavior for Social Robots. *ACM/IEEE International Conference on Human-Robot Interaction*, Arlington, Virginia, USA, Mars 9-11, 2007.
- [56] Hadj-Abdelkader, H., Mezouar, Y., Andreff, N. Martinet, P. Omnidirectional Visual Servoing from Polar Lines, *IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, USA, Mai 15-19, 2006.
- [57] Hashimoto, K., Kimura, H. Visual servoing with nonlinear observer. *IEEE International Conference on Robotics and Automation (ICRA'95)*, Nagoya, Japan, Mai 21-27, 1995.
- [58] Hashimoto, K., Ebine, T., Kimura, H. Visual servoing with hand-eye manipulator – optimal control approach. 1996, *IEEE Transaction on Robotics and Automation*, vol. 12(5) pp 766-774.
- [59] Hill, J., Park, W. T. Real Time Control of a Robot with a Mobile Camera. *9th International Symposium on Industrial Robot*, Washington, DC, Mars 13-15, 1979.
- [60] Horaud, R., Monga, O. Vision par ordinateur, outils fondamentaux (deuxième édition). 1995, *Traité des nouvelles technologies, série Informatique. Hermès, Paris, France.*
- [61] Hosoda, K. Asada, M. Versatile visual servoing without knowledge of true Jacobian. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, Munich, Allemagne, Septembre 12-16, 1994.
- [62] Hutchison, S., Hager, G., Corke, P. A tutorial on visual servo control. 1996, *IEEE Transaction on Robotic and Automation*, vol. 12(5) pp 651-670.
- [63] Hutchison, S., Chaumette, F. Visual servo control, Part I: Basic approaches. 2006, *IEEE Robotic and Automation Magazine*, vol. 13(4) pp 82-90.

- [64] Hutchison, S., Chaumette, F. Visual servo control, Part II: Advanced Approaches. 2007, *IEEE Robotic and Automation Magazine*, vol. 14(1) pp 109-118.
- [65] Huynh, F. Manipulation par la vision pour des tâches de surveillance et de saisie d'objet fixe ou mobile, 1999. *Thèse de doctorat, LAAS-CNRS, Université Paul Sabatier, Toulouse, France.*
- [66] Jägersand, M., Fuentes, O., Nelson, R. Experimental evaluation of uncalibrated visual servoing for precision manipulation". *IEEE International Conference on Robotics and Automation (ICRA '97)*, Albuquerque, New Mexico, USA, Avril 21-27, 1997.
- [67] Joon-Kil, K., Do-Wan, K., Seung-Joon, C. Sang-Chul, W. Image-Based Visual Servoing using Sliding Mode Control, *International Joint Conference, Bexcon, Korea*, Octobre 18-21, 2006.
- [68] Kanda, T., Shiomi, M., Perrin, L., Nomura, T., Ishiguro, H., Hagita., N. Analysis of people trajectories with ubiquitous sensors in a science museum. *International Conference on Robotics and Automation (ICRA '07)*, Rome, Italy, Avril 10-14, 2007.
- [69] Keçeci, F., Tonko, M., Nagel, H., Genbenbach, V. Improving visually servoed disassembly opération by automatic camera placement. *IEEE International Conference on Robotics and Automation (ICRA'98)*, Louvain, Belgique, Mai 16-20,1998.
- [70] Khadraoui, D. La Commande Référencée Vision pour le Guidage Automatique de Véhicules. 1996, *Thèse de doctorat, LASMEA, Université Blaise Pascal de Clermont-Ferrand, France.*
- [71] Khadraoui, D., Debain, C., Rouveure, R., Martinet, P., Bonton, P., Gallice, J. Vision based control in driving assistance of agricultural vehicules. 1998, *International Journal of Robotics Research*, vol. 17(10) pp 1040-1054.
- [72] Kobilarov, M., Sukhatme, G., Hyams, J., Batavia, P. People tracking and following with mobile robot using an omnidirectional camera and laser, *IEEE International Conference on Robotics and Automation (ICRA '06)*, Orlando, USA, Mai 15-19, 2006.
- [73] Krupa, A., Gangloff, J. Doignon, C., Mathelin, M. D. Morel, G., Leroy, J. Soler, L. Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. 2003, *IEEE Transactions on Robotics and Automation*, vol. 19 pp 842-853.
- [74] Kwon, H., Yoon, Y., Byung-Park, J., Kak, A. C. Person Tracking with a Mobile Robot using Two Uncalibrated Independently Moving Cameras, *IEEE International Conference on Robotics and Automation (ICRA '05)*, Barcelona, Spain, Avril 18-22, 2005.

- [75] Malis, E. Contribution à la modélisation et à la commande en asservissement visuel. 1998, *Thèse de doctorat, Université de Rennes I, Rennes, France*.
- [76] Malis, E., Chaumette, F., Boudet, S. Positioning a coarse calibrated camera with respect of unknown object by 2D $\frac{1}{2}$ visual servoing. *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 16-20, 1998.
- [77] Malis, E., Morel, G., Chaumette, F. Robot control from disparate multiple sensors. 2001, *International Journal of Robotics Research*, vol. 20(5), pp 364-378.
- [78] Malis, E., Chaumette, F. Theoretical improvement in the stability analysis of a new class of model free visual servoing methods. 2002, *IEEE Transaction on Robotics and Automation*, vol. 18(2) pp 176-186.
- [79] Malis, E. Visual servoing invariant to changes in camera intrinsic parameters. 2004, *IEEE Transaction on Robotics and Automation*, vol. 20(1) pp 72–81.
- [80] Malis, E., Benhimane, S. A unified approach to visual tracking and servoing. 2005, *International Journal of Robotics and Autonomous Systems*, vol. 52(1) pp 39-52.
- [81] Marchand, E., Courty, N. Controlling a camera in a virtual environment. 2002, *The Visual Computer Journal*, vol. 18(1), pp 1-19.
- [82] Martinet, P., Daucher, N., Gallice, J., Dhome, M. Robot control using 3D monocular pose estimation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Grenoble, France, September 8-12, 1997.
- [83] Martinet, P. Thibaud, C. Automatic guided vehicules: Robust controller design in image space. 2000, *Autonomous Robots*, vol. 8(1) pp 25-42.
- [84] Miura, J., Gangloff, K., de Mathelin, M. Robust and Uncalibrated Visual Servoing without Jacobian Using a Simplex Method, *IEEE International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Septembre 30 - Octobre 4, 2002.
- [85] Morin, P., Samson, C. Practical stabilization of driftless systems on Lie groups: the transverse function approach. 2003, *IEEE Transactions on Automatic Control*, vol. 48(9) pp 1496-1508.
- [86] Ouadah, N. Asservissement visuel d'un bras manipulateur. 2002, *Thèse de Magistère, Ecole Militaire Polytechnique (EMP), Alger*.
- [87] Ouadah, N., Cadenat, V., Boudjema, F., Hamerlain, M. Image Based Robust Visual Servoing on Human Face to Improve Human/Robot Interaction. *European conference on mobile robots (ECMR '2009)*, Mlini/Dubrovnic, Croatia, Septembre 23-25, 2009.

- [88] Ouadah, N., Cadenat, V., Lerasle, F., Hamerlain, M., Germa, T., Boudjema, F. Multi-Sensor-Based Control Strategy for Initiating and Maintaining Interaction between a Robot and a Human. 2011, *Advanced Robotics*, vol. 25 n° 9-10 pp 1249-1270.
- [89] Pagès, J., Collewet, C., Chaumette, F., Salvi, J. Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light. 2008, *IEEE Transaction on Robotics and Automation*, vol. 22(5), pp 1000-1010.
- [90] Papanikolopoulos, N.P., Khosla, P. K., Kanade, T. Visual tracking of a moving target by a camera mounted on a robot: combination of control and vision. 1993, *IEEE Transactions on Robotics and Automation*, vol. 9(1) pp 14-35.
- [91] Pissard-Gibollet, R. Conception et commande par asservissement visuel d'un robot mobile. 1993, *Thèse de doctorat, École des Mines de Paris, France*.
- [92] Pissard-Gibollet, R., Rives, P. "Applying visual servoing techniques to control a mobile hand-eye system. *IEEE International Conference on Robotics and Automation (ICRA'95)*, Nagoya, Japan, Mai 21-27, 1995.
- [93] Rives, P., Pissard-Gibollet, R., Pelletier, L. Sensor-based Tasks: From the Specification to the Control Aspects. *6th International Symposium on Robotics and Manufacturing*, Montpellier, France, May 28-30, 1996.
- [94] Rizzi, A., Koditschek D. An active visual estimator for dexterous manipulation. 1996, *IEEE Transactions on Robotics and Automation*, vol. 12(5) pp 697-713.
- [95] Rives, P. Borrelly, J. J. Underwater pipe inspection task using visual servoing techniques. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97)*, Grenoble, France, Septembre 7-11, 1997.
- [96] Royer, E., Bom, J., Dhome, M., Thuilot, B., Lhuillier, M., Marmoiton, F. Outdoor autonomous navigation using monocular vision, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, Edmonton, Canada, Août 2-6, 2005.
- [97] Samson, C., Borgne, M., L., Espiau, B. Robot Control: The task function approach. 1991, *Oxford science publications, Clarendon Press édition*.
- [98] Schulz, D., Fox, D., Hightower., J. People tracking with anonymous and ID-sensors using rao-blackwellised particle filters. *International Joint Conference on Artificial Intelligence (IJCAI'03)*, Acapulco, Mexico, Août 9-15, 2003.
- [99] Shirai, Y., Inoue, H. Guiding a Robot by Visual Feedback in Assembling Tasks. 1973, *Pattern Recognition*, vol. 5 pp 99-108.

- [100] Sundaeswaran, V., Bouthemy, P., Chaumette, F. Exploiting image motion for active vision in a visual servoing framework. 1996, *International Journal of Robotics Research*, vol. 15(6) pp 629-645.
- [101] Tamadazte, B., Lefort-Piat, N., Marchand, E. Dembélé, S. Microassembly of Complex and Solid 3D MEMS by 3D Vision-based Control, *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS'09)*, St Louis, Missouri, USA, Octobre 11-15, 2009.
- [102] Tsakiris, D., Rives, P., Samson, C. Extending Visual Servoing technique to Non holonomic mobile robots. 1998, In Hager, G., Kriegman, D., Morse, A. The confluence of vision and control, LNCIS Series, vol. 237 pp 106-117, *Springer-Verlag*.
- [103] Utkin, V. I. Sliding Modes in Control Optimization. 1992 , *Springer-Verlag*.
- [104] Victorino, A. C. La commande référencée capteur : une approche robuste au problème de navigation, localisation et cartographie simultanées pour un robot d'intérieur. 2002, *Thèse de doctorat, Université de Nice-Sophia Antipolis, France*.
- [105] Viola, P., Jones, M. Fast multi-view face detection. *International Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, USA, Juin 16-22, 2003.
- [106] Vitrani, M. A., Morel, G., Mitterhofer, H. Bonnet, N. Robust ultrasoundbased visual servoing for beating heart intracardiac surgery. *IEEE International Conference on Robotics and Automation (ICRA'07)*, Roma, Italy, Avril 10-14, 2007.
- [107] Weiss, L. Dynamic visual servo control of robots: an adaptative image-based approach. 1984, *Thèse de doctorat, Université de Carnegie-Mellon, USA*.
- [108] Weiss, L.E., Sanderson, A.C. Dynamic sensor-based control of robots with visual feedback. 1987, *IEEE Journal of Robotics and Automation*, vol. 3(5) pp 404-417 .
- [109] Yuan, J.S.C. A general photogrammetric method for determining object position and orientation. 1989, *IEEE Transactions on Robotics and Automation*, vol. 5(2) pp 129-142.
- [110] Zanne, P., Morel, G., Plestan, F. Robust Vision Based 3d Trajectory Tracking Using Sliding Mode Control, *IEEE International Conference on Robotics and Automation*, San Francisco, USA, Avril 24-28, 2000.