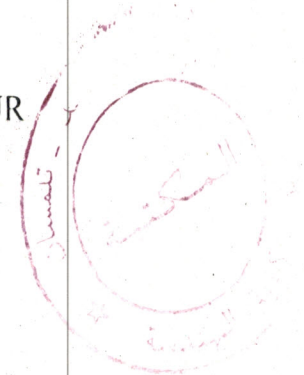


Mémoire

Présenté à

L'UNIVERSITÉ ABOU BEKR BELKAÏD
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE



Pour l'obtention du diplôme de

Magister API (Automatique – Productique – Informatique)

Option : **AUTOMATIQUE**

Par

M^r HADJ ABDELKADER MOHAMMED AMINE

Thème

SUIVI DE CHEMIN POUR UN ROBOT MOBILE NON-
HOLONOME PAR ASSERVISSEMENT VISUEL

Soutenu en Juin 2005

devant la commission d'examen

Président :

M^r N. GHOUALI

Professeur

UABB – Tlemcen

*A mes chers parents et grands-parents,
à toute ma famille,
à tous ceux qui me sont chers.*



Avant-propos

Ce travail a été effectué au Laboratoire d'Automatique de Tlemcen (LAT) sous la direction de M. BRAHIM CHERKI, Maître de conférences à l'université ABOU BEKR BELKAÏD - TLEMCCEN. Durant ces cinq dernières années lors desquelles nous l'avons connu, moi et mes deux camarades de promotion Choukri et Boumediene, il a été pour nous beaucoup plus qu'un professeur ou directeur de thèse, en raison de ses admirables valeurs humaines. D'autre part, ses compétences scientifiques, ses encouragements ainsi que ses critiques objectives m'ont été très bénéfiques pour mener à bien la réalisation de mon travail. En plus, il n'a pas épargné le moindre effort pour m'aider tout au long de mon parcours universitaire. Je lui dois beaucoup et il m'est difficile de trouver les bons mots pour le remercier. Qu'il trouve ici le témoignage de ma haute considération et de toute ma reconnaissance.

Je remercie les membres de jury ayant accepté de juger ce travail :

M. N. GHOUALI, Professeur à l'université ABOU BEKR BELKAÏD - TLEMCCEN, qui me fait l'honneur de présider ce jury,

M. F. KHELFI, Maître de conférences à l'université d'Oran - Es-Sénia et M. C. KARA TERKI, Chargé de cours à l'université ABOU BEKR BELKAÏD - TLEMCCEN pour avoir bien voulu juger ce travail.

Mes profonds remerciements vont à mon cousin et co-encadreur, Hicham, pour avoir accepté la lourde tâche d'être rapporteur. Son aide technique ainsi que ses réponses précises à mes innombrables questions m'ont permis de comprendre les divers aspects théoriques et pratiques de l'asservissement visuel. Je tiens à lui exprimer ma gratitude et à lui passer un grand merci (et je lui promet de ne plus le déranger autant!).

Je voudrai également remercier :

- Mes collègues de Magister API, pour les inoubliables moments que nous avons vécus ensemble pendant ces trois dernières années d'études post-graduées. Merci à vous : Choukri, Boumediene, Amine, Djelloul, Ahmed, Latéfa, Mounira, Sidi Mohamed et les autres,

- Les copains des autres promotions de Magister, Signaux et systèmes, SRT et Mécanique avancée, pour leurs amitiés inconditionnelles et leurs sourires permanents. Je ne citerai pas les noms par peur d'en oublier quelques uns,
- Les enseignants de l'université de Tlemcen en particulier, et ceux des autres universités pour leurs cours de valeurs inestimables qu'ils ont soigneusement assurés et qui nous ont permis d'acquérir des connaissances solides dans notre discipline. Je les remercie également pour leurs constantes et permanentes préoccupations,
- Les thésards du laboratoire LASC à l'université de Metz, Saber, M'hammad, Anas et Djamel, pour m'avoir bien accueilli durant mon stage en France. Je ne manquerai pas aussi de remercier Smaïl pour l'inoubliable journée que nous avons passée chez lui, à Longwy, et pour ses CD-Rom précieux,
- Tous les autres copains que je côtoyais dans la vie quotidienne : dans la mosquée, au cybercafé et les week-ends. Je citerai particulièrement : Mustapha, Kamel, Azedine, Abderrafie et Abdelbasset, Karim et H'mida. Je salue également les futurs docteurs en médecine qui étaient des vrais amis dès les premiers moments de notre rencontre et jusqu'à présent : mon cher cousin Hichem, Fouzi, Marouane, Sidi Mohammed et Sid-Ahmed ; je vous souhaite un avenir brillant à vous tous car c'est ce que vous méritez vraiment. Je n'oublierai pas non plus de remercier Abdennour et Mohammed, Samira et Wassila, ainsi que les copains d'outre mer : Amina, Sabri et Rania.

Durant les moments difficiles que j'ai vécus récemment, de nombreux nouveaux amis étaient présents et m'ont considérablement aidé à surpasser cette épreuve. Je ne pourrai jamais remercier assez Dr. Zoubida R. qui a pris soin de moi tout au long de mon hospitalisation, je lui souhaite de réussir dans tous ces futurs examens et dans sa vie. Mes remerciements aux Dr. Mesli et Dr. Benallal et aussi à Noureddine et aux jeunes infirmiers qui n'attendaient pas de primes pour bien prendre soin des malades auxquels je souhaite une proche guérison. Un grand merci également à mon voisin Mohammed qui m'a aidé énormément dans cette période.

Pour conclure, je demanderai à Dieu seul de bien récompenser tous ceux qui étaient toujours présents à mes côtés, en particulier : mes parents, mes frères, mes grands parents et les autres membres de ma famille qui m'ont supporté pendant toutes mes 25 années sans se lasser. Sans eux, rien de tout cela n'aurait été possible.

Table des matières

Notations	4
Introduction générale	7
1 Etat de l'art et formulation du problème	11
1.1 Introduction [24][27]	11
1.2 Les techniques d'asservissement visuel	13
1.2.1 L'asservissement visuel 3D [5] [10]	13
1.2.2 L'asservissement visuel 2D [3] [5]	15
1.2.3 L'asservissement visuel 2D1/2 [23] [24][25]	17
1.2.4 Conclusion	19
1.3 Formulation du problème	20
2 Système de vision et géométrie épipolaire	24
2.1 Introduction	24
2.2 Espace projectif et transformation homogène	25
2.2.1 La géométrie projective [13] [24]	25
2.2.2 Les matrices de transformation homogène [17]	26
2.3 Processus de formation d'image	28
2.3.1 Les matrices de projection perspective [24]	28
2.3.2 Modélisation de la caméra [3][13]	29
2.4 Reconstruction projective	30
2.4.1 La relation entre deux images [24]	32
2.4.2 La contrainte épipolaire [25][27]	32
2.5 Estimation de la géométrie épipolaire	35
2.5.1 Etat de l'art [27][28]	35
2.5.2 Estimation linéaire de la matrice fondamentale [23]	36
2.5.3 Estimation de l'homographie à partir de points ou droites coplanaires [24]	37

2.6	Reconstruction des paramètres tridimensionnels	39
2.6.1	Reconstruction à partir de la matrice d'homographie [12]	40
2.6.2	Reconstruction à partir de la matrice essentielle [25]	47
2.7	Conclusion	48
3	Modélisation et commande du robot mobile	49
3.1	Les contraintes non holonomes [30]	50
3.2	Modélisation	52
3.2.1	Modèle cinématique [2]	52
3.2.2	Chemins et trajectoires [19]	53
3.2.3	Modélisation dans un repère de Frenet [19][30]	53
3.3	Etude de la commandabilité du système	56
3.3.1	Commandabilité [19]	56
3.4	Synthèse de la loi de commande	58
3.4.1	Passage au modèle en forme chaînée [31]	58
3.4.2	Commande pour le suivi de chemin [33]	60
3.4.3	Application au robot mobile	64
3.4.4	Suivi d'un chemin droit	65
3.5	Conclusion	66
4	Aspects pratiques et résultats de simulation	67
4.1	Aspects pratiques	67
4.2	Résultats et commentaires	69
4.3	Conclusion	74
	Conclusion générale	76

Table des figures

1-1	Principe de l'asservissement visuel 3D	13
1-2	Schéma synoptique de l'asservissement visuel 3D	14
1-3	Principe de l'asservissement visuel 2D	16
1-4	Schéma bloc de l'asservissement visuel 2D	17
1-5	Schéma bloc de l'asservissement visuel 2D1/2	19
1-6	Stratégie globale de navigation	22
1-7	Configuration considérée	23
2-1	Projection perspective	28
2-2	Transformation mètres-pixels	30
2-3	Déplacement de la caméra	31
2-4	Contrainte épipolaire et parallaxe virtuelle	34
3-1	Exemple d'une roue	51
3-2	Configuration d'un robot mobile	53
3-3	Modélisation dans un repère de Frenet	54
4-1	La boucle fermé du système de navigation par asservissement visuel	68
4-2	Points dans l'image en positions initiale et désirée	71
4-3	Positions initiale et désirée de la caméra dans l'espace de travail	71
4-4	Trajectoires des points sur le plan image	72
4-5	Trajectoire de la caméra dans l'espace de travail	73
4-6	Déviations latérales de la caméra	74
4-7	Déviations angulaires de la caméra	75

Notations

Règles générales

- minuscule : scalaire
- minuscule gras : vecteur
- majuscule gras : matrice

Mathématiques

- \mathbb{P}^n : espace projectif de dimension n
- \mathbb{R}^n : espace euclidien de dimension n
- \mathbf{U} : matrice orthonormale par SVD
- \mathbf{S} : matrice des valeurs singulières par SVD
- \mathbf{V} : matrice orthonormale par SVD
- σ : valeurs singulière
- λ : valeur propre
- $[\mathbf{v}]_x$: matrice de préproduit vectoriel associée au vecteur \mathbf{v}
- \mathcal{U} : classe de fonctions dépendantes du temps
- \mathcal{K} : compact de \mathbb{R}^n
- $[f, g]$: Crochet de Lie de deux champs de vecteur f et g
- $[f, g][k]$: k^e coordonnée de $[f, g]$

Asservissement visuel

- \mathbf{T} : matrice homogène de changement de repère cartésien
- \mathbf{t} : vecteur de translation
- \mathbf{R} : matrice de rotation
- \mathbf{u} : axe de rotation

θ : angle de rotation
 s : signal capteur
 r : attitude d'un repère

Géométrie épipolaire

\mathcal{F} : repère cartésien
 \mathcal{O} : centre de projection
 \mathcal{P} : point dans l'espace euclidien
 \mathbf{x} : vecteur contenant les coordonnées d'un point de l'espace euclidien
 Π : plan de projection ou plan image
 π : plan dans l'espace euclidien
 $\boldsymbol{\pi}$: vecteur contenant les coordonnées homogènes d'un plan de l'espace euclidien
 \mathbf{n} : normale à un plan
 d : distance du centre de projection à un plan
 \mathbf{P} : matrice de projection perspective
 \mathbf{K} : matrice de changement de repère mètre/pixel dans l'image
 f : distance focale de la caméra
 \mathbf{m} : point image en coordonnées métriques ou normalisées
 \mathbf{p} : point image en coordonnées pixels
 \mathbf{F} : matrice fondamentale
 \mathbf{E} : matrice essentielle
 \mathbf{H} : matrice d'homographie dans l'espace euclidien
 \mathbf{G} : matrice d'homographie dans l'espace projectif
 \mathbf{e} : épipole
 l : droite épipolaire

Robotique

Γ : courbe définissant le chemin à suivre
 \mathcal{M} : point de référence du robot mobile
 V : vitesse linéaire du robot mobile
 ω : vitesse angulaire du robot mobile
 \mathbf{x}_r : vecteur d'état du robot mobile
 \mathbf{u}_r : vecteur des variables de commande du robot

s : abscisse curviligne sur le chemin

$c(s)$: courbure du chemin au point s

y : déviation latérale par rapport au chemin

θ : déviation angulaire par rapport au chemin

z_i : variables d'état du système chaîné

m_i : variable de commande du système chaîné

K_p : gain proportionnel du correcteur

K_d : gain dérivée du correcteur

Introduction Générale

Depuis quelques années, les développements technologiques des capteurs sensoriels ont permis leur implantation sur les robots mobiles. L'intérêt principal de cette intégration est de fournir au robot des informations sensorielles sur son environnement et d'ouvrir ainsi de nombreuses et de nouvelles perspectives en robotique, particulièrement dans le domaine de la navigation des robots.

La navigation des robots mobiles en environnement structuré est un thème de recherche central depuis plusieurs années. Classiquement, le problème est traité en deux temps. Tout d'abord, une étape de planification permet de déterminer un chemin ou une trajectoire de consigne réalisable par le robot, c'est-à-dire dans l'espace libre et vérifiant les contraintes **non holonomes**¹ de roulement sans glissement caractéristiques des robots mobiles. Dans un deuxième temps, une étape de synthèse définit une loi de commande permettant au véhicule de suivre le chemin ou la trajectoire spécifiée.

Le problème de la commande des robots mobiles a suscité l'intérêt des chercheurs en Automatique pendant les quinze dernières années. Il recouvre classiquement trois sous-problèmes :

- La poursuite de trajectoire : où le robot doit atteindre et suivre une trajectoire dans l'espace cartésien (c'est-à-dire un chemin géométrique auquel est associée une loi temporelle) à partir d'une configuration initiale donnée (sur la trajectoire ou pas),
- Le suivi de chemin : le robot doit atteindre et suivre un chemin dans l'espace cartésien à partir d'une configuration initiale donnée (sur le chemin ou pas),
- La stabilisation en une configuration fixe : le robot doit atteindre une configuration désirée à partir de sa configuration courante.

Ces trois problèmes sont généralement posés en terme de régulation à zéro d'une erreur entre la configuration courante du véhicule et la consigne définie par la trajectoire, le chemin ou la posture de référence. Les deux premiers problèmes ont été résolus à l'aide des techniques classiques de com-

¹Les contraintes non holonomes dans un système mécanique sont dues aux contacts des roues avec le sol. Elles empêchent le système de se déplacer dans la direction donnée par l'axe des roues motrices.

mande non linéaire. Le problème de stabilisation en un point est par contre beaucoup plus complexe, puisqu'il a été montré qu'il ne peut pas exister de retour d'état continu permettant de stabiliser un robot dans une configuration prédéfinie. Ce problème ne peut donc être traité comme un cas particulier du suivi de chemin ou de poursuite de trajectoire, et d'autres classes de lois de commande doivent être mises en oeuvre. Deux approches ont été proposées et consistent à synthétiser des correcteurs instationnaires ou discontinus. Le lecteur intéressé pourra trouver une description précise et complète de ces méthodes dans [19].

La résolution de ces différents problèmes de commande requiert la détermination de l'état du robot et donc sa reconstruction par le biais d'une étape de localisation. Celle-ci ne peut être effectuée uniquement sur la base des données fournies par les capteurs proprioceptifs du robot qui ne prennent pas en compte les problèmes dus aux interactions avec l'environnement comme le glissement des roues sur le sol. On ne peut donc espérer localiser précisément un véhicule en position et orientation uniquement à l'aide de ces données, et il s'avère nécessaire de faire appel aux capteurs extéroceptifs qui renseignent sur la situation du robot dans son environnement. Ainsi, les capteurs de force qui quantifient le contact avec les éléments extérieurs du robot, les capteurs proximétriques ou télémétriques (ultra-sons, lasers,...) qui détectent des objets situés à plus au moins grande distance du véhicule, ou bien les caméras qui permettent de percevoir l'environnement sont autant d'exemples de capteurs extéroceptifs.

Parmi les capteurs extéroceptifs existants, la vision apparaît comme un capteur privilégié pour réaliser des tâches robotiques. En effet, du fait de la richesse du signal vidéo, les informations que l'on peut extraire d'une image sont très variées et de plus ou moins haut niveau selon le traitement mis en oeuvre. Néanmoins, comme nous l'avons mentionné précédemment, les informations choisies pour effectuer une tâche doivent pouvoir être extraites en temps réel. Seules des **primitives**² simples seront donc exploitables en commande. Il existe deux structures de commande référencée vision [17] : l'une repose sur un asservissement en situation de la caméra par rapport à l'environnement, tandis que l'autre est basée sur la régulation à zéro d'une erreur dans l'image.

Ainsi, dans le premier cas, la tâche est décrite en terme d'une situation à atteindre entre la caméra et l'environnement (en général, un objet face auquel la caméra doit être positionnée). Il est alors nécessaire d'effectuer une interprétation de l'image de manière à reconstruire la situation courante

²On appelle primitive toute forme géométrique simple pouvant être extraite d'une image, par exemple : un point, un segment de droite, une ellipse, etc...

de la caméra. La littérature met en exergue deux approches distinctes selon le niveau du contrôle (*static look and move* si l'acquisition et le mouvement se font séquentiellement, et *dynamic look and move* s'ils se font simultanément). Dans l'asservissement visuel **2D**, par contre, la tâche n'est plus décrite en termes d'une situation entre la caméra et l'environnement, mais d'un motif à atteindre dans l'image. La commande est synthétisée uniquement sur la base d'informations visuelles. Cette dernière approche permet donc de s'affranchir de l'étape d'interprétation et les problèmes qui y sont liés (temps de calcul et erreurs d'estimation). Une description générale de ces techniques est donnée dans le chapitre suivant.

Outre ces deux techniques principales d'asservissement visuel, d'autres structures sont envisageables, dont la plus réputée pour ses performances est la technique d'asservissement visuel **2D1/2**. En fait, cette approche constitue, en quelque sorte, un couplage des deux méthodes que nous avons décrites précédemment. Appliquée à des robots manipulateurs à 6 degrés de liberté, cette technique a permis d'avoir des résultats aux propriétés particulièrement intéressantes (fort découplage de la loi de commande, robustesse vis-à-vis des erreurs de modélisation et de calibration). Elle repose principalement sur des calculs de **l'homographie** qui permettent de reconstruire partiellement l'état du système (en particulier, la composante de rotation dans son vecteur d'état). Une homographie est une application linéaire de l'espace projectif de dimension 2, \mathbb{P}^2 , en lui-même. On présentera plus de détails sur le calcul de cette homographie et ses propriétés dans le chapitre 2.

Les états reconstruits seront directement contrôlés dans l'espace cartésien par des lois de commandes adaptées au type du robot utilisé. En plus, un contrôle partiel dans l'image permet de commander les autres états exprimant la translation du système. Il sera donc plus simple qu'en asservissement visuel **2D**. Une étude complète de l'application de cette technique à un robot manipulateur à 6 degrés de liberté, avec analyse de la robustesse et de la stabilité du système, se trouve dans [24].

Dans notre travail, nous avons utilisé la technique d'asservissement visuel **2D1/2** afin de réaliser un suivi de trajectoire pour un robot mobile non holonome (du type *tricycle*) dans un environnement d'intérieur. Le chemin dans l'espace de travail est indiqué par des marquages qui peuvent être suivis par un capteur de vision. Bien que le principe de la méthode **2D1/2** est le même, les procédures de mise en oeuvre ainsi que les techniques de commande utilisées sont totalement différentes en raison des contraintes non holonomes qui entravent les mouvements du robot mobile. Dans la suite, Nous allons présenter l'organisation du document et les sujets abordés dans chaque chapitre de ce manuscrit.

Organisation du manuscrit

Ce mémoire est structuré de la manière suivante :

- Dans le premier chapitre, nous dresserons un bref état de l'art des divers travaux relatifs à l'asservissement visuel. Nous décrirons brièvement les principales structures de commande permettant la réalisation de tâches robotiques à partir d'informations visuelles extraites d'images fournies par un capteur de vision. Nous présenterons ensuite de manière plus précise, les différents schémas de commande **3D**, **2D** et **2D1/2**. Nous nous attacherons à préciser les avantages et inconvénients de ces trois approches et les critères qui nous ont permis de choisir l'une de ces techniques. Enfin, nous terminerons par une formulation générale de notre problème.
 - Dans le deuxième chapitre, nous rappellerons quelques résultats issus de la géométrie épipolaire. Nous discuterons, en particulier, les formulations matricielles de la contrainte épipolaire. Nous présenterons ensuite des méthodes d'estimation de la géométrie épipolaire et de reconstruction du déplacement partiel de la caméra et de la géométrie de la scène. Nous tiendrons à donner un bref état de l'art des méthodes les plus utilisées dans ce domaine et à détailler les calculs pour celles que nous avons utilisées.
 - Dans le troisième chapitre, nous décrirons l'aspect contrôle dans notre travail. Après avoir discuté d'une manière générale les contraintes non holonomes qui entravent les mouvements du robot mobile considéré, nous présenterons les différents modèles exploités dans de tels schémas de commande. Nous discuterons également de la commandabilité du système et nous donnerons en détails les étapes de la synthèse d'une loi de commande permettant de réaliser notre objectif.
 - Dans le quatrième chapitre, nous allons discuter d'une manière grossière les aspects pratiques du suivi de trajectoire, pour le robot non holonome, par asservissement visuel. Ensuite, nous nous intéresserons d'une façon plus détaillée à la présentation des différents résultats que nous avons obtenus en simulation et qui prouvent l'efficacité de la technique de commande utilisée. Des commentaires seront ajoutés pour mieux éclaircir le contenu.
- Enfin, dans la conclusion, nous mettons en évidence les contributions principales de notre travail et nous présentons quelques perspectives de recherche.

Chapitre 1

Etat de l'art et formulation du problème

Dans ce chapitre, nous présentons un bref regard sur les différentes techniques d'asservissement visuel. Nous décrirons brièvement les principales structures de commande permettant la réalisation de tâches robotiques à partir des informations visuelles extraites d'images fournies par un capteur de vision. Nous présentons ensuite de manière plus précise les différents schémas de commande **2D**, **3D** et **2D1/2**. Nous nous attacherons à préciser les avantages et inconvénients de ces trois approches et les critères qui nous ont permis de choisir l'une de ces techniques. Enfin, nous terminerons par une formulation générale de notre problème.

1.1 Introduction [24][27]

L'intégration des données capteurs et plus particulièrement celles issues d'une caméra dans la boucle de commande des robots permet un élargissement important du domaine d'application de la robotique et une amélioration considérable de la précision des positions obtenues. De nombreux travaux ont donc été menés dans ce sens et ont donné lieu à une littérature abondante.

L'objectif de l'asservissement visuel est l'intégration en temps réel d'informations visuelles issues d'une ou plusieurs caméras dans la boucle de commande du robot afin d'en contrôler les mouvements. De manière très classique, on peut classer les différentes techniques d'asservissement visuel en fonction de mesures utilisées en entrée du module de commande. Nous décrirons donc les techniques d'asservissement visuel **3D** et **2D**, qui ont fait l'objet de la majorité des travaux de chercheurs dans le domaine depuis de longues années [38][11][15][17]. Nous présenterons ensuite l'une des approches récentes dans le domaine, à savoir l'asservissement visuel **2D 1/2**, qui a été introduites par F. Chau-

mette et al. [24][25]. Nous tiendrons à détailler les avantages et les inconvénients de chacune de ces techniques.

Une autre classification des systèmes de commande par vision peut être faite en utilisant deux critères :

- l'utilisation ou non d'une boucle interne,
- l'espace de contrôle.

Dans le cas où le système ne possède pas de boucle interne, le système de vision fournit directement une estimation de l'état du robot aux articulations, se substituant à ses contrôleurs. La stabilisation est réalisée en utilisant seulement un retour sur le système de vision. Dans ce cas de figure, le système doit fournir une estimation de l'état à une cadence élevée. En raison de l'influence importante des perturbations sur l'estimation temps réel de l'état, ce type de schéma a été peu utilisé. Cependant, il existe plusieurs travaux, notamment sur des robots à six degrés de liberté, exploitant cette structure [15][38].

Dans le cas où le schéma de commande utilisé est hiérarchique (utilisation d'une boucle interne), le système de vision fournit une consigne au contrôleur du robot qui calcule les déplacements à réaliser dans l'espace articulaire par l'intermédiaire d'une boucle fermée interne qui stabilise le système. En utilisant une telle structure, la cadence du processus peut être amenée au temps réel vidéo de 40 *ms*. Le processus de vision et le système robotique sont alors clairement disjoints, ce qui permet une plus grande portabilité et simplicité des processus de commande par vision. On pourra se reporter à [6] pour une étude sur les performances dynamiques de cette structure.

Les systèmes robotisés munis d'une caméra peuvent également être distingués par le positionnement des caméras :

- lorsque la caméra n'a pas de liaison mécanique avec le robot asservi par vision, on parlera d'asservissement visuel déporté. La caméra est positionnée de manière à ce qu'elle observe soit le robot, soit les objets d'intérêt dans la scène, ou les deux à la fois. Elle peut être fixe ou montée sur un autre système mécanique,
- lorsque la caméra est montée sur l'organe terminal du robot manipulateur ou sur le robot mobile, on parlera d'asservissement visuel embarqué. Cette configuration est la plus répandue [15][17].

Cette dernière configuration est celle que nous avons considérée dans notre travail.

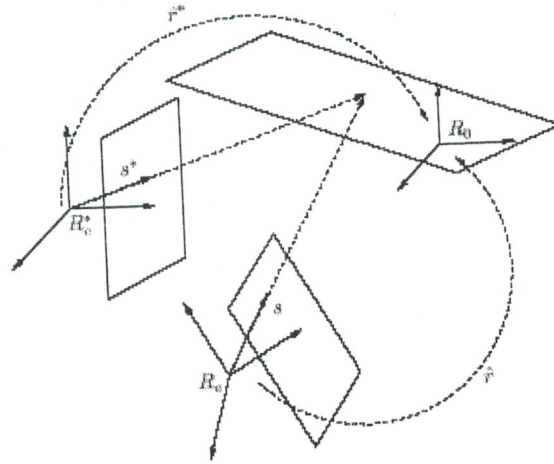


FIG. 1-1: Principe de l'asservissement visuel 3D

1.2 Les techniques d'asservissement visuel

Les premiers travaux en asservissement visuel ont porté sur des tâches de positionnement à l'aide d'une caméra embarquée sur un robot de type manipulateur. Il existe également de nombreux travaux sur l'utilisation de caméras déportées. Par ailleurs, l'utilisation de l'asservissement visuel dans la robotique mobile connaît depuis quelques années un essor important, notamment pour les applications de conduite automatique de véhicules [30]. La difficulté essentielle réside alors dans l'élaboration de lois de commande prenant en compte les contraintes non holonomes de ce type de robots.

1.2.1 L'asservissement visuel 3D [5] [10]

Comme son nom l'indique, l'asservissement visuel 3D (ou *position-based visual servoing*) utilise en entrée de la boucle de commande des informations tridimensionnelles exprimées dans un repère euclidien. Plus précisément, si l'on considère une tâche de positionnement rigide contrôlant les six degrés de liberté de la caméra (voir la figure 1-1), la consigne et la mesure sont exprimées sous la forme d'un changement de repère décrivant la position et l'orientation d'un repère lié à l'objet d'intérêt par rapport au repère lié à la caméra. On notera par la suite r , élément de \mathbb{R}^6 , une représentation d'un tel changement de repère et \hat{r} le résultat d'une estimation de r . La boucle de commande consiste alors à estimer à chaque itération le changement de repère $\hat{r}(t)$ pour la position courante de la caméra et à amener progressivement la caméra à sa position désirée \hat{r}^* par rapport à l'objet.

On peut bien sûr aussi ne considérer qu'un sous-ensemble des six paramètres de r ou fixer des contraintes entre ces paramètres afin de réaliser des tâches contraignant un nombre inférieur de de-

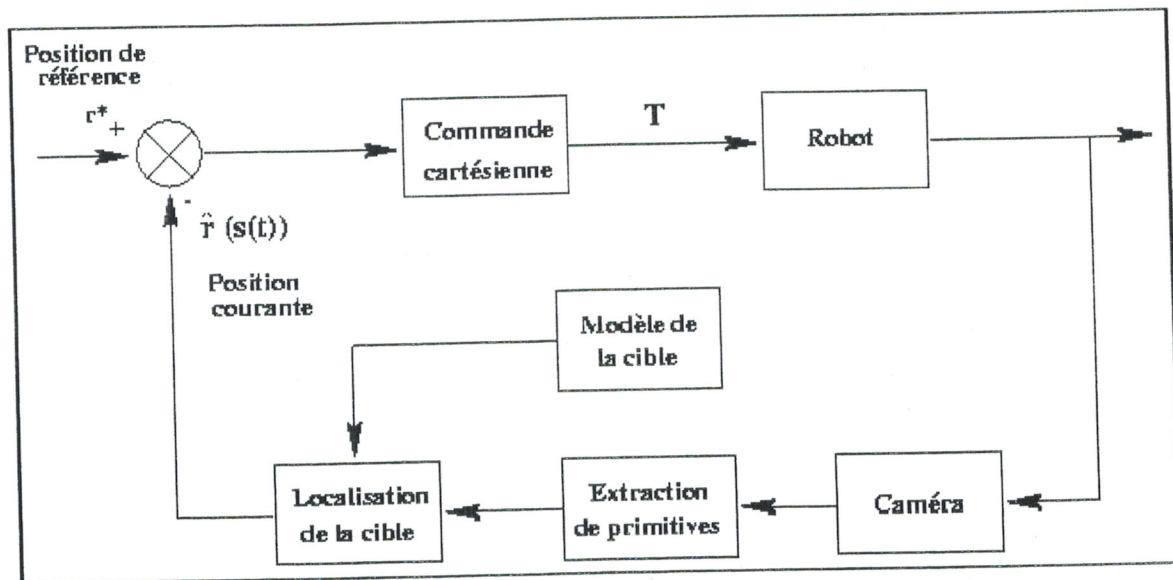


FIG. 1-2: Schéma synoptique de l'asservissement visuel 3D

grés de liberté entre la caméra et la scène considérée.

En théorie, une seule itération est suffisante et une commande du robot en position permet d'atteindre la position désirée. En pratique évidemment, les diverses erreurs de modèle et de calibration nécessitent un rebouclage et un asservissement en vitesse pour assurer la stabilité du système. Un schéma synoptique de ce type d'asservissement visuel est donné par la figure 1-2.

De nombreuses méthodes existent pour estimer la position d'une caméra par rapport à un objet en utilisant une image de cet objet. Elles reposent très généralement sur la connaissance à priori d'un modèle 3D de l'objet et des paramètres intrinsèques de la caméra. Ces méthodes utilisent comme informations visuelles soit des points, soit des droites, voire des ellipses résultant de la projection de cercles ou de sphères ou encore des objets cylindriques. Par contre, très peu de méthodes combinent les différents types d'informations visuelles.

Concernant les aspects de calibration, on peut dire qu'une calibration grossière est généralement suffisante pour assurer la convergence des lois de commande utilisées. Ceci est grâce à la boucle fermée employée. Dans cet aspect, on peut distinguer deux cas de figure dans le choix de la consigne :

- soit la consigne est fixée à priori. La position atteinte après la convergence sera alors différente de celle désirée en raison du biais introduit dans l'estimation de $\hat{r}(t)$. La situation est identique si le modèle 3D de l'objet n'est pas parfaitement exact.

- soit la consigne \hat{r}^* est calculée de la même manière que les mesures $\hat{r}(t)$, c'est-à-dire en utilisant une image de l'objet acquise à partir d'une position d'apprentissage. On se retrouve alors dans la situation paradoxale où la consigne et les mesures sont toutes les deux biaisées, mais la situation atteinte après la convergence des lois de commande est correcte aux erreurs de mesures près.

Puisqu'en pratique, il est extrêmement difficile d'obtenir une calibration exacte du système, nous utiliserons la seconde méthode afin d'obtenir l'état désiré du système. Notre objectif sera de fournir une commande adéquate entre l'état initial et l'état final, en passant par des états intermédiaires pour avoir plus de précision dans le suivi de la trajectoire choisie.

L'asservissement visuel **3D** présente l'avantage de fournir des mesures directement dans l'espace euclidien, ce qui permet l'emploi de lois de commande extrêmement simples pour aller itérativement d'une position à une autre, voire une certaine planification pour éviter par exemple les singularités du robot ou des obstacles connus de la scène. L'inconvénient qui en résulte est qu'aucun contrôle véritable dans l'image n'est effectué, ce qui implique que l'objet d'intérêt peut très bien sortir du champ de vue de la caméra pendant son déplacement, entraînant alors l'arrêt de la loi de commande. Un tel phénomène peut se produire pour des erreurs importantes introduites dans les paramètres intrinsèques de la caméra, dans le modèle du robot ou le passage caméra-robot.

1.2.2 L'asservissement visuel 2D [3] [5]

Les techniques d'asservissement visuel **2D** (*image-based visual servoing*) utilisent directement les informations visuelles s extraites de l'image, c'est-à-dire qu'elles ne nécessitent pas la phase d'estimation de \hat{r} contrairement à l'asservissement visuel **3D**. Les lois de commande consistent alors à contrôler le mouvement de la caméra afin que les mesures dans l'image $s(t)$ atteignent une valeur désirée s^* , voire suivre une trajectoire spécifiée $s^*(t)$ (voir la figure 1-3).

Nous souhaitons que notre motif de référence, qui est représenté par un rectangle noir dans cette figure, occupe une certaine position dans l'image. Pour réaliser cette tâche, nous effectuons en un premier temps, à partir d'une image prise par la caméra, une extraction des primitives visuelles présentes dans l'image. Une fois le motif extrait, nous pouvons comparer sa position courante à la position désirée. La différence obtenue est le facteur essentiel pris en compte dans la synthèse de la loi de commande.

Différents types d'informations extraites de l'image peuvent être utilisés. La plupart des travaux

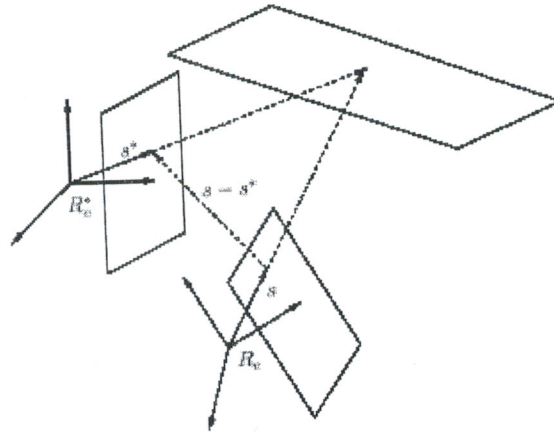


FIG. 1-3: Principe de l'asservissement visuel 2D

en asservissement visuel 2D sont basés sur l'utilisation de coordonnées de points dans les images. Mais il s'avère que l'utilisation de telles primitives peut dans certains cas être problématique. D'autres primitives géométriques peuvent éventuellement être choisies (comme les segments de droites, les ellipses ou les centres de régions, etc.) [7] [8], mais les traitements d'images permettant l'extraction de telles informations seront plus lourds, ce qui réduira nettement les performances du système.

Dans tous les cas de figure, le comportement du système commandé dépendra fortement des informations visuelles choisies, de leur nombre et leur configuration, mais aussi du point de vue à partir duquel elles sont observées [5][34].

Un autre facteur important à connaître dans cet asservissement est l'approximation que nous avons de la distance à la cible dans l'image. Bien que cette information ne peut être connue qu'à un facteur d'échelle près, elle s'avère nécessaire pour déterminer si l'objectif de l'asservissement a été achevé ou pas. Dès que la commande est déterminée, elle est transmise au robot qui effectue le déplacement adéquat. Ainsi, cette boucle, englobant l'extraction des primitives, la détermination de la profondeur approximative et l'application de la commande calculée, doit être effectuée tant que le motif de l'image courante n'a pas atteint sa position désirée. La figure 1-4 donne le schéma bloc de l'asservissement visuel 2D.

Les consignes sont donc exprimées sous la forme d'indices visuels à atteindre dans l'image (s^*), c'est-à-dire par le biais des informations visuelles qui ont été choisies pour caractériser l'objet d'intérêt dans l'image. Là aussi, ces consignes peuvent être soit fixées a priori, soit déterminées par apprentissage pour éviter que les erreurs dans la calibration du système influencent la convergence des lois

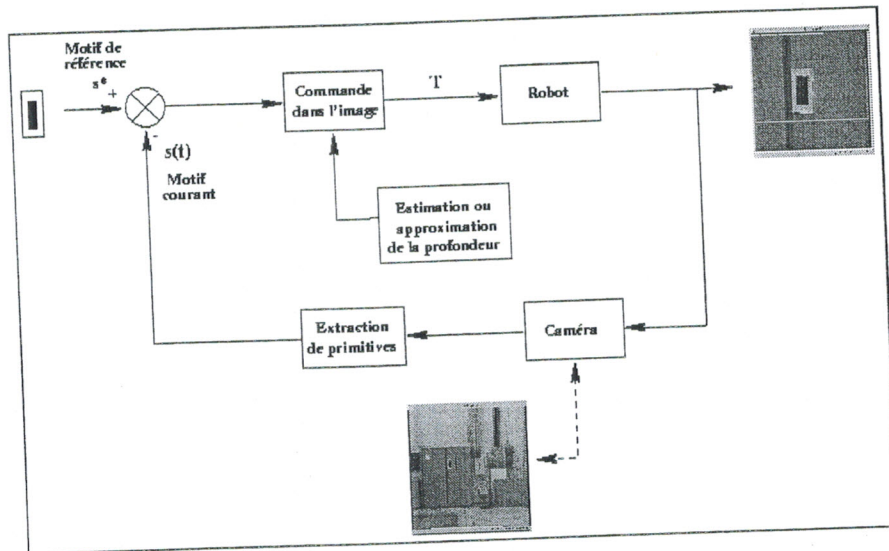


FIG. 1-4: Schéma bloc de l'asservissement visuel 2D

de commande.

En comparant avec l'asservissement visuel 3D, nous pouvons remarquer que le modèle de la cible n'est plus nécessaire, mais il faut calculer en ligne à partir des indices visuels une approximation de la profondeur entre la caméra et l'objet par rapport auquel nous souhaitons positionner le robot. Par contre, les trajectoires des primitives visuelles dans l'image peuvent correspondre à des mouvements non réalisables par le robot en question (par exemple, un mouvement latéral pour un robot non holonome) et être parfois surprenantes. Ceci vient du fait que ces trajectoires ont été déterminées directement dans l'image sans prise en considération des singularités du système robotique.

1.2.3 L'asservissement visuel 2D1/2 [23] [24][25]

L'asservissement visuel 2D1/2 a été développée récemment par F. Chaumette et *al.* [23][24]. Il est fondé sur une régulation simultanée dans l'image d'une part et sur les informations provenant d'une reconstruction partielle du 3D d'autre part. Cette reconstruction euclidienne est qualifiée de 'partielle' du fait que le déplacement en translation de la caméra entre les différentes prises de vue ne peut être obtenu qu'à un facteur d'échelle près. La reconstruction du déplacement en rotation est, quant à elle, déterminée avec exactitude.

Plus précisément, cette approche est basée sur l'estimation de l'**homographie**¹ qui relie les co-

¹Une homographie est une application linéaire qui relie les coordonnées de points, appartenant à un même plan, sur plusieurs images.

ordonnées de points dans plusieurs images [1] [13]. Dans le cas d'un objet plan, un minimum de quatre points appariés sur les images courante et désirée permet une telle estimation par la résolution d'un simple système linéaire (voir la section 2.5.3). Dans le cas d'un objet quelconque, l'appariement de huit points permet d'aboutir, là-encore par la résolution de systèmes linéaires. A partir de l'homographie obtenue, il est possible de calculer le déplacement en rotation 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 d/d^* entre les distances courante et désirée de la caméra à l'objet. Comme la translation à réaliser n'est connue qu'à un facteur d'échelle près, un asservissement visuel purement **3D** est impossible. Cependant, la combinaison des informations **2D** et **3D** disponibles permet d'aboutir à une solution aux propriétés particulièrement intéressantes.

En effet, l'erreur de rotation estimée est contrôlée directement dans l'espace cartésien avec les degrés de liberté en rotation du robot (trois degrés de liberté en cas d'un robot manipulateur et un seul degré dans le cas de notre robot mobile). En plus, le mouvement de translation (le long de l'axe optique pour les robots manipulateurs et dans la direction d'avancement pour les robots mobiles) est contraint de manière à ce que le rapport des profondeurs entre la caméra et l'objet tende vers 1, ce qui veut dire que les deux plans images courant et désiré sont finalement confondus. Notons toutefois que, pour les robots mobiles, cette dernière étape n'est possible que lorsque l'objet d'intérêt est situé dans la direction d'avancement du robot. Le schéma de commande dans l'asservissement visuel **2D1/2** est donné par la figure 1-5.

Un tel schéma de commande intègre donc à la fois un contrôle de la trajectoire de la caméra dans son espace de travail et un contrôle de la trajectoire de certaines primitives dans l'image. De cette manière, un fort découplage de la loi de commande est assuré et, en plus, l'objet d'intérêt restera dans le champ de vision de la caméra à cause du contrôle partiel effectué dans l'image.

L'avantage majeur de cette approche est que la connaissance d'un modèle géométrique **3D** de l'objet 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 tridimensionnelle utilisée dans la commande est la profondeur désirée approximative d'un point de l'objet, soit un besoin d'informations à priori bien moindre qu'en asservissement visuel **3D** et **2D**.

Des résultats formels sur la stabilité et sur la robustesse aux erreurs de modélisation pour ce type

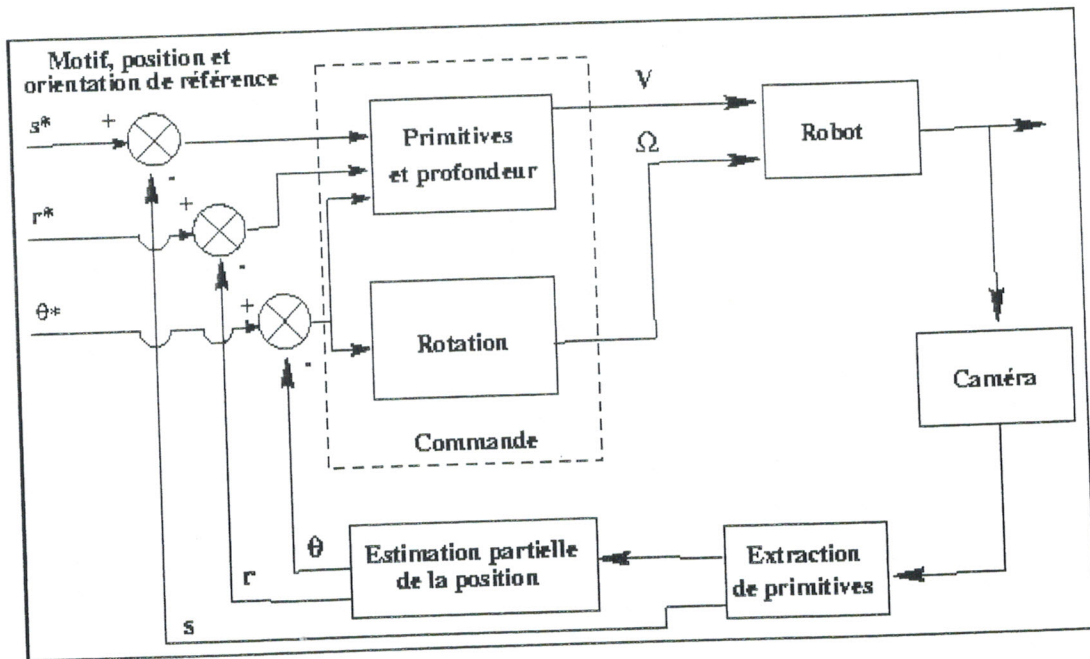


FIG. 1-5: Schéma bloc de l'asservissement visuel 2D1/2

d'asservissement visuel appliqué à un bras manipulateur ont été obtenus dans le cas où les informations tridimensionnelles sont estimés à partir de la matrice d'homographie relative à un plan de référence [24].

1.2.4 Conclusion

En récapitulation, nous pouvons constater que les asservissement visuels 3D nécessitent, en général, la connaissance de l'objet observé. L'étape d'estimation des paramètres 3D peut s'avérer délicate en particulier lorsque les paramètres de la caméra ou le modèle de l'objet sont erronés. En outre, la contrainte de visibilité est difficile à intégrer à la structure de commande. En effet, les trajectoires des informations visuelles dans l'image peuvent être inadéquates car les mouvements du robot sont contrôlés directement dans l'espace de travail. Au contraire, le contrôle direct des mouvements du robot dans l'espace Cartésien permet d'obtenir des trajectoires du robot très satisfaisantes.

Les techniques d'asservissement visuels 2D permettent, quant à elles, une prise en compte plus directe de la contrainte de visibilité car le signal d'erreur à réguler est directement construit à partir d'informations extraites de l'image. Cependant, l'objet d'intérêt peut tout de même quitter l'image durant l'asservissement lorsque les erreurs de modélisation ou les déplacements à effectuer sont importants. En outre, la trajectoire du robot dans son espace de travail n'est pas du tout contrôlée et peut conduire à un échec de l'asservissement si les déplacements à effectuer ne prennent pas en

considération les contraintes qui entravent les mouvements du robot.

En raison des inconvénients de chacune des techniques citées précédemment, nous avons opté, dans notre travail, pour l'utilisation de la technique d'asservissement visuel **2D1/2**, car d'une part, cette technique ne nécessite pas des connaissances sur l'objet d'intérêt ni une reconstruction de sa géométrie et, d'autre part, elle permet de garantir la visibilité de l'objet à cause du contrôle partiel effectué dans l'image. En plus, le contrôle des mouvements de rotation directement dans l'espace Cartésien nous permettra d'utiliser des lois de commande relativement simples dans lesquelles on pourra facilement prendre en compte les contraintes cinématiques de robot.

Cette technique nécessite donc un calcul de l'homographie à chaque itération de la boucle de commande qui nous permettra de déterminer les différentes rotations à effectuer entre chaque couple d'images clefs. L'estimation de l'homographie est sensible aux erreurs de modélisation et de calibration du capteur. Cependant, dans le cas particulier où les objets d'intérêt sont planaires, l'expérience a montré qu'elle est suffisamment robuste non seulement par rapport aux erreurs de modélisation et de calibration de la caméra, mais aussi par rapport aux inévitables bruits de mesures. Le fait que les objets d'intérêt sont planaires nous est bénéfique également du point de vue du temps de calcul, puisque l'estimation de l'homographie se réduit à la résolution de systèmes linéaires simples.

Dans les situations où les images courante et désirée sont très éloignées, une instabilité se produit dans le calcul de l'homographie, conduisant ainsi à des résultats entièrement incorrects dans les relations entre ces prises de vue. L'utilisation d'images relais s'avère donc indispensable dans ce schéma de commande. Les méthodes de calcul de l'homographie utilisées dans ce type d'asservissement visuel sont présentées dans le chapitre suivant.

Dans la suite de chapitre, nous allons présenter la formulation du problème, en détaillant la configuration du robot mobile et du système de vision utilisé.

1.3 Formulation du problème

Dans ce mémoire, on traite la navigation des robots mobiles dans un environnement interne par vision artificielle et plus particulièrement, le suivi d'un chemin défini par des amers ou des indices visuels. Typiquement, les amers visuels sont extraits du système de vision mono-oculaire ou stéréo.

Connaissant le modèle et la position **3D** de ces amers, l'attitude (**pose**² en anglais) de la caméra est calculée et une localisation globale du robot dans son environnement **3D** est mise à jour. Après la localisation du système, les mouvements du robot sont contrôlés en minimisant une fonction d'erreur définie dans l'espace euclidien.

Dans notre approche, nous supposons l'utilisation d'un seul capteur visuel pour localiser et commander le système. Ce capteur (la caméra) est rigidement fixé sur le robot mobile considéré. Afin d'observer un environnement aussi statique que possible (sans être gêné par les obstacles dynamiques), la caméra est censée observer le plafond ou le sol de l'environnement.

La stratégie de navigation dans ce problème est composée de cinq étapes :

1. Phase d'apprentissage supervisée : dans cette étape, l'utilisateur téléguide le robot dans son espace de travail. Le robot acquiert un panel d'images clefs représentatives des lieux. Ces images sont ensuite indexées (en respectant un ordre topologique) et organisées dans une base de données. Dans le cas idéal, ces images doivent fournir un échantillon représentatif de tous les points de vues qui pourraient être atteints pendant le déplacement du robot. Cette base d'images est souvent dénommée : la **mémoire visuelle** du robot.
2. Phase de localisation : dans la seconde étape, le robot acquiert une image à un emplacement non spécifié de son espace de travail. Il se localise ensuite par rapport à la base d'images apprises en identifiant l'image de la base la plus proche de l'image courante.
3. Sélection de la destination : une fois le robot localisé dans l'**espace image** (c'est-à-dire par rapport à la base d'images), une configuration finale, spécifiée à travers une image requête dans la base d'images, est choisie.
4. Planification de chemin : une suite d'images ordonnées de la base, appelées **images relais**, est construite grâce à l'indexation topologique de la base. Elle décrit "l'itinéraire" à emprunter pour atteindre l'image cible à partir de l'image courante. Les configurations injoignables par le robot (à cause des contraintes non holonomes) sont évitées par le fait que le robot a été lui-même téléguidé précédemment dans son espace de travail.
5. L'itinéraire déterminé par la suite d'images relais est interpolé et parcouru en s'appuyant sur les techniques de commande les plus adaptées au type du robot utilisé.

Cette stratégie de navigation est illustrée dans le schéma donné par la figure 1-6.

²Dans la suite, on désignera par pose un élément de **SE(3)** (Special Euclidean Group), c'est-à-dire la caractérisation de la situation d'un solide par sa position et son orientation dans un repère choisi.

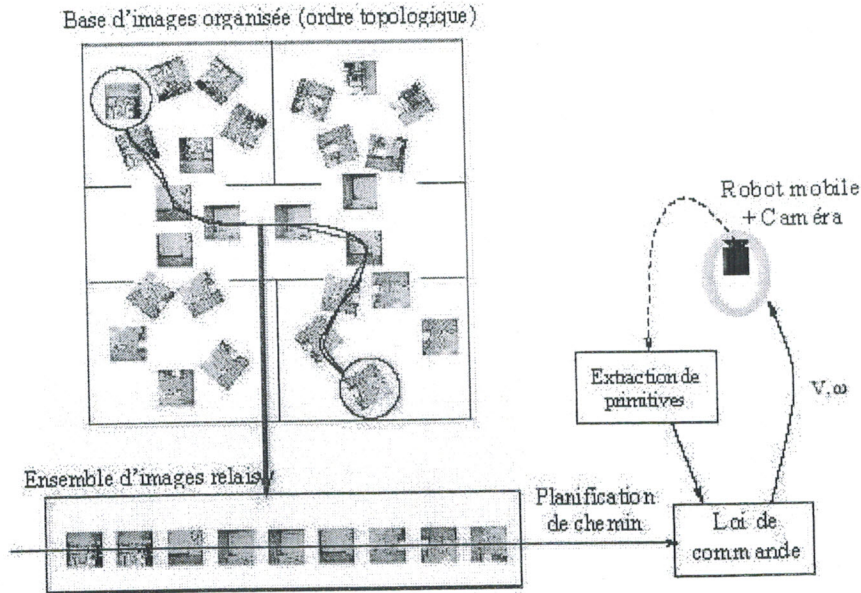


FIG. 1-6: Stratégie globale de navigation

Pour naviguer dans une base de données d'images, le robot mobile doit atteindre des configurations intermédiaires exprimées en tant qu'images désirées à visualiser sur la caméra. Donc, il nous faut, tout d'abord, trouver la relation entre chaque image relais et l'image relais suivante. Cette relation n'est autre que la translation et la rotation que le système doit effectuer pour passer d'une image relais à une autre. Ensuite, il faudra trouver la loi de commande permettant au robot de se déplacer le long du chemin défini par ces positions intermédiaires (données par les images relais). Finalement, le chemin entier sera suivi par une succession de mouvements élémentaires déterminés entre chaque couple d'images relais. Ce travail nous permet donc de calculer tous les mouvements non holonomes possibles du robot mobile, directement dans l'espace image.

Nous proposons dans ce travail d'utiliser la technique d'asservissement visuel **2D1/2** pour calculer ces translations et rotations. D'une manière générale, cette technique consiste en le calcul d'une homographie entre chaque couple d'images relais. A partir de cette homographie, il est possible de déterminer la rotation et la translation à un facteur d'échelle près entre ces images. Ces deux informations seront exploitées dans la commande du système. Dans la suite du document (et plus précisément au chapitre 2), on parlera des calculs d'homographie entre l'image courante (qui est acquise à l'instant actuel) et d'image désirée (représentant la prochaine image clef à atteindre).

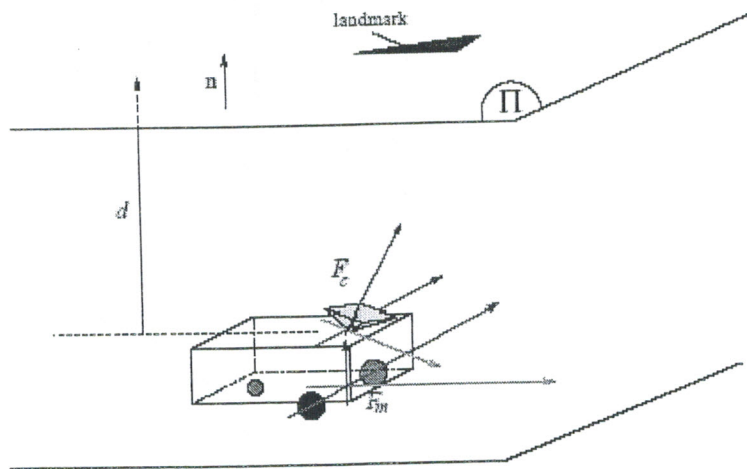


FIG. 1-7: Configuration considérée

Le système non holonome considéré dans ce travail est un robot mobile du type tricycle. On suppose que la caméra est rigidement fixée sur le robot mobile et qu'elle est dirigée vers le plafond de la scène avec un angle d'inclinaison (*tilt angle*) de 0.2 rad à peu près (voir la figure 1-7). Ceci permettra de visualiser une surface plus importante du plafond. Notons Π le plan représentant le plafond dans la scène, il est généralement parallèle au plan d'évolution du robot mobile. Le plan Π est censé contenir quelques amers artificiels planaires qui peuvent être suivis dans l'image \mathcal{I} par un système de vision. Nous utiliserons l'appellation anglaise **Landmark** pour distinguer ces amers planaires dans le reste du document. Notons \mathcal{L} l'une de ces Landmarks et \mathcal{P} un point se trouvant sur \mathcal{L} .

Nous supposons qu'un algorithme de suivi (*tracking algorithm*) peut nous fournir les coordonnées \mathbf{p} de \mathcal{P} à chaque itération de la boucle de commande (de tels paramètres sont des sorties standards des algorithmes de tracking). Dans la suite, on ne s'intéressera pas aux traitements d'images permettant d'effectuer la tâche d'extraction des primitives visuelles. Nous supposerons dorénavant que les coordonnées des points d'intérêts (constituants la Landmark) sont disponibles à chaque instant de l'asservissement.

Remarque 1.1 Cette dernière supposition est pratiquement correcte du fait que les primitives visuelles utilisées sont les plus simples possibles (primitives du type point). Pour des primitives plus complexes (telles que les segments de droites, les ellipses ou les contours), les traitements seront plus lourds et les informations du capteurs peuvent ne pas être disponibles en temps réels.

Le but est donc de relier, à chaque instant de l'asservissement, les images courante et désirée en déplaçant le robot non holonome le long d'un chemin adéquat.

Chapitre 2

Système de vision et géométrie épipolaire

2.1 Introduction

La localisation tridimensionnelle d'une caméra dans la scène qu'elle observe est un problème qui peut être traité de deux manières différentes. Le choix de l'une des méthodes par rapport à l'autre dépend naturellement des informations dont on dispose sur la caméra et sur la scène observée. Le premier type de méthodes utilise une mise en correspondance entre des points de l'espace Cartésien et leurs projections dans l'image de la caméra pour obtenir la pose de l'objet d'intérêt dans le repère lié à la caméra. Dans ces méthodes là, il est nécessaire de connaître le modèle de la scène ou, tout du moins, un certain nombre de ces primitives [10].

Dans le second groupe de ces méthodes de localisation, que nous décrirons plus en détail dans la suite de ce chapitre, le principe est de retrouver la géométrie de la scène observée à partir de la mise en correspondance de primitives géométriques dans plusieurs images. Dans ce cas, le modèle de la scène ou de la cible observée n'est plus nécessaire.

Si la calibration du capteur est connue, on parlera de reconstruction euclidienne ; sinon la reconstruction sera dite projective. Dans le cas d'une reconstruction euclidienne, le déplacement en translation de la caméra entre les prises de vue ne pourra être obtenu qu'à un facteur d'échelle près ; on parlera alors de **reconstruction partielle du déplacement** de la caméra.

Dans la suite de ce chapitre, nous rappelons tout d'abord quelques résultats de géométrie projective et de géométrie épipolaire. En particulier, nous donnons les formulations matricielles de la contrainte

épipolaire. Nous présentons ensuite quelques méthodes d'estimation de la géométrie épipolaire et de reconstruction partielle du déplacement de la caméra et de la géométrie de la scène. Nous tiendrons à détailler les calculs pour les méthodes que nous avons utilisées dans les programmes de simulation.

2.2 Espace projectif et transformation homogène

2.2.1 La géométrie projective [13] [24]

La géométrie projective constitue un outil puissant de modélisation en vision par ordinateur et est de ce fait très utilisée dans ce domaine. Elle permet une résolution élégante de problèmes primordiaux en vision, dont nous citons par exemple :

- l'estimation du déplacement partiel de la caméra entre deux prises de vue et la reconstruction de scène à partir de plusieurs images [13][27],
- l'estimation des paramètres de la caméra [27],
- le positionnement d'un robot relativement à une scène inconnue [25].

La caméra est un capteur qui réalise une projection d'un ensemble de points d'une scène tridimensionnelle dans un espace à deux dimensions : le plan image. Le cadre naturel pour étudier de manière simple de telles transformations est la géométrie projective. Nous allons voir qu'il est également intéressant d'englober l'espace cartésien de dimension n , \mathfrak{R}^n , dans l'espace projectif \mathbb{P}^n afin de manipuler simplement les transformations affines agissant sur \mathfrak{R}^n . En effet, cette opération va permettre de formuler linéairement ce type de transformation mais aussi de manipuler des entités géométriques à l'infini.

Soit \mathcal{P} un point dans un espace euclidien à n dimensions, exprimé dans un repère donné par le vecteur $\mathbf{x} = [X_1 \ X_2 \ \dots \ X_n]^T \in \mathfrak{R}^n$. Ce même point peut être décrit dans l'espace projectif \mathbb{P}^n en utilisant les coordonnées homogènes $(\mathbf{x}, 1) = [X_1 \ X_2 \ \dots \ X_n \ 1]^T \in \mathfrak{R}^{n+1}$. De plus, une direction $[X_1 \ X_2 \ \dots \ X_n]^T$ dans l'espace euclidien, qui correspond à un "point à l'infini", peut être représentée à l'aide des coordonnées homogènes par le point projectif $(\mathbf{x}, 0) = [X_1 \ X_2 \ \dots \ X_n \ 0]^T$.

Dans le cas général, un point dans l'espace projectif \mathbb{P}^n est représenté par un vecteur de $n + 1$ composantes. Ses éléments sont définis à un facteur d'échelle près et au moins un d'entre eux est différent de zéro. Les points euclidiens peuvent être reconstruits à partir des points projectifs en divisant toutes les coordonnées par la dernière coordonnée projective, évidemment seulement si celle-ci

n'est pas nulle.

Un hyperplan dans l'espace euclidien est l'ensemble de tous les points de \mathbb{R}^n dont les coordonnées satisfont une équation linéaire. En utilisant les coordonnées homogènes, l'hyperplan peut être représenté dans l'espace projectif par un vecteur non nul π (de dimension $n + 1$), défini à un facteur d'échelle près.

Un hyperplan très particulier dans l'espace euclidien est l'ensemble de tous les "points à l'infini", qui correspond à un "idéal hyperplan à l'infini", et qui est représenté par $\pi_\infty = [0 \ 0 \ \dots \ 0 \ 1]^T$.

Tous les points dans \mathbb{P}^n appartenant à un hyperplan vérifient l'équation linéaire :

$$\pi^T(\mathbf{x}, 1) = 0$$

Cette équation présente une symétrie entre les composantes de l'hyperplan π et les coordonnées du point \mathcal{P} , car en fait : $\pi^T(\mathbf{x}, 1) = (\mathbf{x}, 1)^T \pi$. D'autre part, l'espace formé par tous les hyperplans peut être considéré comme un autre espace projectif appelé le *dual* de l'espace original \mathbb{P}^n . Un *principe de dualité* extrêmement important découle de cette symétrie, il est défini dans [29] par :

Principe de dualité : *N'importe quelle propriété projective obtenue en utilisant des points et des hyperplans possède un résultat symétrique, dans lequel les rôles des hyperplans et des points sont inversés.*

N'importe quelle matrice de dimension $(m + 1, n + 1)$ de rang plein peut être utilisée pour réaliser une application projective entre deux espace \mathbb{P}^m et \mathbb{P}^n . Une application projective inversible de \mathbb{P}^n en lui-même est appelée une **homographie** et correspond à un changement de coordonnées projectives obtenu en utilisant une matrice de dimension $(n + 1, n + 1)$. Ce changement de coordonnées dans l'espace projectif est analogue au changement de coordonnées dans l'espace euclidien qui est effectué par les matrices de transformation homogène.

Avant de modéliser le capteur caméra, Nous rappelons brièvement les matrices de transformation homogène de dimension (4×4) (qui réalisent une transformation de \mathbb{R}^3 en lui même) et les matrices de projection perspective de dimension (3×4) (qui réalisent une transformation de \mathbb{R}^3 en \mathbb{R}^2).

2.2.2 Les matrices de transformation homogène [17]

Soit \mathcal{F}^* le repère de référence dans l'espace cartésien, appelé aussi le repère désiré. Toutes les entités géométriques exprimées dans ce repère seront caractérisées par l'indice *. *La matrice de*

transformation homogène entre \mathcal{F}^* et un deuxième repère \mathcal{F} (représentant le repère courant) est :

$${}^d\mathbf{T}_c = \begin{bmatrix} {}^d\mathbf{R}_c & {}^d\mathbf{t}_c \\ 0 & 1 \end{bmatrix} \quad (2.1)$$

où ${}^d\mathbf{R}_c$ et ${}^d\mathbf{t}_c$ sont respectivement la matrice de rotation et le vecteur de translation entre \mathcal{F}^* et \mathcal{F} ('d' à gauche et 'c' à droite indiquent que le repère courant est exprimé dans le repère désiré). La transformation inverse est donnée par :

$${}^c\mathbf{T}_d = \begin{bmatrix} {}^d\mathbf{R}_c^T & -{}^d\mathbf{R}_c^T {}^d\mathbf{t}_c \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{R}_d & {}^c\mathbf{t}_d \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

Les matrices de transformation homogènes permettent d'exprimer les changements de coordonnées homogènes des points et des plans de l'espace cartésien d'une manière très simple :

- un point \mathbf{x}^* de l'espace cartésien, exprimé dans le repère \mathcal{F}^* , est représenté par les coordonnées homogènes $[X^* \ Y^* \ Z^* \ W^*] \in \mathbb{R}^4$. Comme on l'a déjà vu, si la quatrième coordonnée homogène est nulle, le point est à l'infini (c'est-à-dire qu'il représente une direction dans l'espace cartésien). Autrement, on peut poser $W^* = 1$, sans perte de généralité, pour représenter un point fini. Ce point est exprimé dans le nouveau repère \mathcal{F} de la manière suivante :

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = {}^d\mathbf{T}_c \begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} \quad (2.3)$$

- un plan dans l'espace cartésien, exprimé dans le repère \mathcal{F}^* , est caractérisé par son vecteur normal \mathbf{n}^* et sa distance d^* à l'origine \mathcal{O}^* du repère de référence. En utilisant les coordonnées homogènes, il est représenté par le vecteur $\boldsymbol{\pi}^* = [\mathbf{n}^{*T} \ -d^*]^T$. Ce même plan, noté $\boldsymbol{\pi} = [\mathbf{n}^T \ -d]^T$ dans le repère \mathcal{F} , est exprimé de la manière suivante :

$$\boldsymbol{\pi} = {}^d\mathbf{T}_c^{-T} \boldsymbol{\pi}^* \quad \Rightarrow \quad \begin{bmatrix} \mathbf{n} \\ -d \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{R}_d \mathbf{n}^* \\ -d^* - {}^c\mathbf{t}_d^T {}^c\mathbf{R}_d \mathbf{n}^* \end{bmatrix} \quad (2.4)$$

où la matrice ${}^d\mathbf{T}_c^{-T}$ est appelée la matrice *duale* de ${}^d\mathbf{T}_c$.

Une matrice de transformation homogène permet aussi de représenter la situation d'un solide dans l'espace cartésien par rapport à un repère de référence. Cependant, une telle représentation est redondante car seulement trois variables suffisent pour identifier une matrice de rotation dans le groupe $\mathbf{SO}(3)$ des rotations. Nous choisirons plutôt la représentation des rotations par le vecteur de dimension 3, $\boldsymbol{\theta} = \mathbf{u}\theta$, où \mathbf{u} représente le vecteur unitaire porté par l'axe de la rotation et θ est l'angle

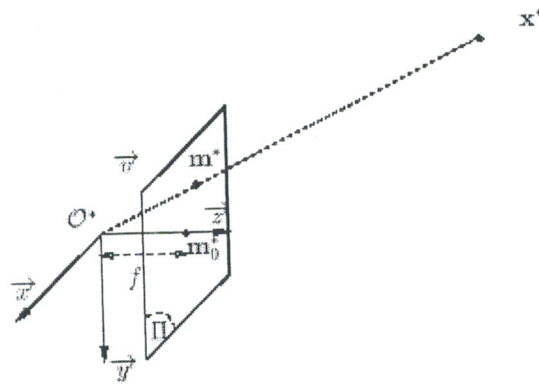


FIG. 2-1: Projection perspective

de rotation. D'une part, cette représentation de la rotation a l'avantage d'être minimale et peut facilement être obtenue à partir de la matrice de rotation associée. D'autre part, le jacobien liant les variations temporelles de cette paramétrisation et le torseur cinématique de la caméra présente une singularité uniquement pour $\theta = 2k\pi$ (avec $k \in \mathbb{Z}^*$), situées en dehors de l'espace de travail.

2.3 Processus de formation d'image

2.3.1 Les matrices de projection perspective [24]

Soit le *centre de projection*, O^* , l'origine du repère de référence. Le plan de projection Π est parallèle au plan (\vec{x}, \vec{y}) et la distance f entre O^* et Π est appelée *distance focale*. L'intersection m_0^* de l'axe \vec{z} et du plan image (voir la figure 2-1) est appelée le *point principal*.

La projection perspective d'un point $\mathbf{x}^* \in \mathbb{P}^3$ en un point $\mathbf{m}^* = [fx^* \ fy^* \ f] \in \mathbb{P}^2$ dans le plan de projection Π peut s'écrire :

$$\frac{Z^*}{f} \mathbf{m}^* = \mathbf{P}^* \begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} \quad (2.5)$$

où $\mathbf{P}^* = [\mathbf{I}_3 \ \mathbf{0}]$ est la *matrice de projection* de dimension (3×4) et Z^* est la profondeur du point dans le repère \mathcal{F}^* .

Comme on l'a vu précédemment, un changement de repère dans l'espace cartésien est représenté par une matrice de transformation homogène. Donc, après un déplacement de la caméra de \mathcal{F}^* en \mathcal{F} , le point de coordonnées \mathbf{x}^* dans \mathcal{F}^* (et \mathbf{x} dans \mathcal{F}) est projeté au point de coordonnées $\mathbf{m} = [fx \ fy \ f]$

$\in \mathbb{P}^2$ suivant la relation :

$$\frac{Z}{f} \mathbf{m} = \mathbf{P}^* \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \mathbf{P}^* {}^c \mathbf{T}_d \begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} \quad (2.6)$$

où $\mathbf{P} = [{}^c \mathbf{R}_d \quad {}^c \mathbf{t}_d]$ est la matrice de projection (3×4) exprimée dans le nouveau repère et Z est la profondeur du point dans le repère \mathcal{F} . Dans la suite, on peut fixer $f = 1$, sans perte de généralité, car les coordonnées de tout point fini dans l'espace projectif sont définis à un facteur d'échelle près. On va voir maintenant que ces matrices de projection perspective sont utilisées pour modéliser les caméras par un modèle sténopé.

2.3.2 Modélisation de la caméra [3][13]

La caméra, décrite en utilisant le modèle projectif dit *sténopé* (ou modèle *pinhole* en anglais), réalise une projection perspective d'un point $\mathbf{x} \in \mathbb{P}^3$ dans un point image $\mathbf{p} \in \mathbb{P}^2$ de coordonnées homogènes $\mathbf{p} = [u \ v \ 1]^T$ exprimées en pixels (rappelons qu'un point \mathbf{x} est projeté en un point de coordonnées métriques $\mathbf{m} = [x \ y \ 1]^T$.)

La transformation projective entre les coordonnées métriques et les coordonnées pixels d'un point dans l'image est :

$$\mathbf{p} = \mathbf{K} \mathbf{m} \quad (2.7)$$

où \mathbf{K} est la matrice des paramètres intrinsèques de la caméra qui réalise le changement des coordonnées projectives (de pixel en mètre) dans \mathbb{P}^2 . Elle est donnée par :

$$\mathbf{K} = \begin{bmatrix} f k_u & -f k_u \cot(\phi) & u_0 \\ 0 & f k_v / \sin(\phi) & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & \alpha_{uv} & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

où u_0 et v_0 sont les coordonnées en pixels du point principal, k_u et k_v sont les facteurs d'échelle suivant les axes \vec{u} et \vec{v} respectivement (en pixels/mètres) et ϕ est l'angle entre ces axes (voir la figure 2-2).

En appliquant le changement de coordonnées projectives, on obtient à partir de l'équation 2.6 :

$$\frac{Z}{f} \mathbf{p} = \mathbf{K} \mathbf{P}^* \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \quad (2.9)$$

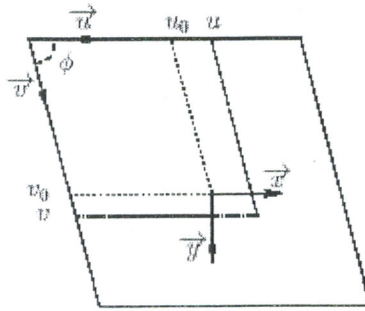


FIG. 2-2: Transformation mètres-pixels

Cette dernière équation représente le modèle général du processus de formation d'image.

Il existe dans la littérature d'autres modèles de caméra (projection orthographique, perspective faible, para perspective,...). Le lecteur intéressé pourra se référer à [13]. Notons également que la relation entre les coordonnées homogènes de \mathbf{x} et \mathbf{p} est linéaire car la caméra est supposée être parfaite (c'est-à-dire sans distorsions).

Si les paramètres intrinsèques de la caméra sont parfaitement connus, ce qui nécessite une calibration très précise de la caméra, on peut travailler indifféremment avec les coordonnées normalisées ou exprimées en pixels. Par contre, une hypothèse plus réaliste en pratique est de considérer que les paramètres intrinsèques de la caméra sont grossièrement connus. Ceci peut conduire à des erreurs dans les coordonnées normalisées des points projetés et peut influencer, de ce fait, la précision des asservissement réalisés.

Toutefois, l'utilisation de méthodes de reconstruction projective, que nous décrivons dans la suite, permet de concevoir des schémas d'asservissement robustes aux incertitudes sur ces paramètres, mais il faut tout de même souligner qu'une normalisation de données est nécessaire quand on veut effectuer une reconstruction projective à partir des points dont les coordonnées sont exprimées en pixels.

2.4 Reconstruction projective

Le problème classique de reconstruction du déplacement d'une caméra entre deux prises de vue, lorsqu'on ne dispose pas d'informations précises sur la géométrie de la scène est, en général, décomposé en deux étapes. La première consiste en l'estimation de la géométrie épipolaire à partir de la mise en correspondance d'un certain nombre de primitives géométriques dans les deux images. La seconde étape consiste ensuite à utiliser les relations liant les expressions matricielles de la contrainte

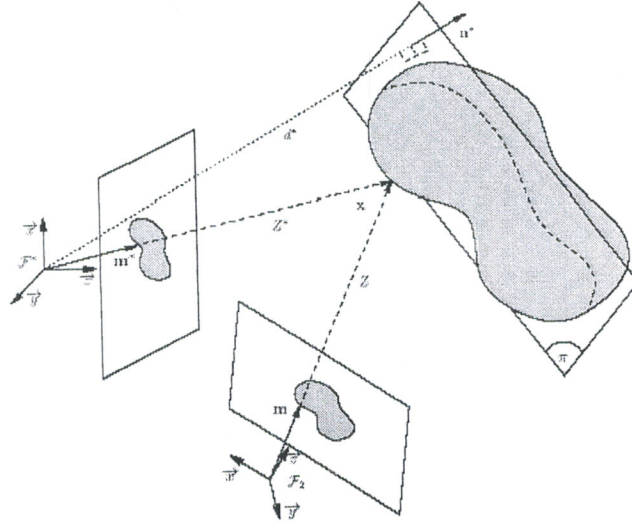


FIG. 2-3: Déplacement de la caméra

épipolaire pour construire des méthodes permettant de retrouver le déplacement partiel de la caméra et la structure de la scène. Dans ce paragraphe, nous allons décrire les relations matricielles qui nous intéresseront dans notre travail. Pour une description plus complète sur ce sujet, le lecteur pourra se reporter à [13][24].

Dans le contexte de l'asservissement visuel, l'objectif est de concevoir une loi de commande afin que la caméra, initialement dans la position définie par le repère \mathcal{F} rejoigne une position désirée définie par le repère \mathcal{F}^* (voir la figure 2-3 où les entités géométriques exprimés dans le repère \mathcal{F}^* sont caractérisées par l'indice *). Il est donc important d'établir la relation qui lie les deux projections perspectives d'un objet statique au déplacement effectué par la caméra.

Considérons un plan de référence π attaché à l'objet (voir la figure 2-3), décrit par le vecteur $\pi^* = [\mathbf{n}^* \quad -d^*]^T$ (on rappelle que n^* est la normale au plan et d^* la distance entre l'origine de \mathcal{F}^* et le plan). On définit $d(\mathbf{x}^*, \pi^*) = \pi^{*T}(\mathbf{x}^*, 1)$ comme étant la distance signée entre le point 3D de coordonnées homogènes $(\mathbf{x}^*, 1) = [X^* \ Y^* \ Z^* \ W^*]^T$ et le plan, telle que :

- si $d(\mathbf{x}^*, \pi^*) = 0$, le point appartient à π ;
- si $d(\mathbf{x}^*, \pi^*) < 0$, le point est entre π et la caméra ;
- si $d(\mathbf{x}^*, \pi^*) > 0$, le point est au delà de π .

La coordonnée homogène W^* de \mathbf{x}^* peut s'écrire :

$$W^* = \frac{Z^* \mathbf{n}^{*T} \mathbf{m}^*}{d^*} - \frac{d(\mathbf{x}^*, \pi^*)}{d^*} \quad (2.10)$$

car $\mathbf{m}^* = \frac{1}{Z^*} \mathbf{P}^*(\mathbf{x}^*, 1) = \frac{1}{Z^*} \mathbf{x}^*$ (voir l'équation 2.5). On va voir maintenant que cette représentation de la quatrième coordonnée homogène joue un rôle fondamental dans la définition de la relation entre les deux images de l'objet.

2.4.1 La relation entre deux images [24]

Afin de lier les deux projections perspectives de la cible, considérons un point \mathbf{x}^* reconstruit à partir de ses coordonnées dans le plan image :

$$\begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} = Z^* \mathbf{P}^{*+} \mathbf{m}^* + \mathbf{P}^{*\perp} \quad (2.11)$$

où $\mathbf{P}^{*+} = \mathbf{P}^{*T} (\mathbf{P}^* \mathbf{P}^{*T})^{-1} = [\mathbf{I}_3 \ 0]^T$ est la pseudo-inverse de la matrice de projection \mathbf{P}^* et $\mathbf{P}^{*\perp} = [0 \ 0 \ 0 \ W^*]^T$ est le vecteur engendré par son noyau. A partir de 2.10, l'équation 2.11 peut s'écrire :

$$\begin{bmatrix} \mathbf{x}^* \\ 1 \end{bmatrix} = Z^* \mathbf{N}_\pi^* \mathbf{m}^* + \varepsilon_\pi^* \quad (2.12)$$

où $\mathbf{N}_\pi^* = [\mathbf{I}_3 \ \mathbf{n}^*/d^*]^T$ et $\varepsilon_\pi^* = [0 \ \varepsilon]^T$ (avec $\varepsilon = -d(x^*, \pi^*)/d^*$). Et finalement, à partir de l'équation $Z\mathbf{m} = \mathbf{P}(\mathbf{x}^*, 1)$, la relation entre les deux points projetés dans les deux images est donc :

$$Z\mathbf{m} = Z^* \mathbf{P} \mathbf{N}_\pi^* \mathbf{m}^* + \mathbf{P} \varepsilon_\pi^* = Z^* \mathbf{H} \mathbf{m}^* + \varepsilon t \quad (2.13)$$

où $\mathbf{H} = \mathbf{P} \mathbf{N}_\pi^* = \mathbf{R} + \mathbf{t} \mathbf{n}^{*T}/d^*$ est la *matrice d'homographie* relative au plan de référence. Si ce dernier est le plan à l'infini π_∞ , on a $\mathbf{n}^*/d^* = 0$ et $\varepsilon = 1$, l'équation 2.13 devient alors :

$$Z\mathbf{m} = Z^* \mathbf{R} \mathbf{m}^* + \mathbf{t} \quad (2.14)$$

où $\mathbf{R} = \mathbf{H}_\infty$ est la matrice d'homographie relative au plan à l'infini (car elle fait correspondre au deux projections les points appartenant au plan à l'infini).

2.4.2 La contrainte épipolaire [25][27]

L'équation 2.13 peut facilement être écrite dans le système de coordonnées en pixels en utilisant l'équation 2.7 :

$$Z\mathbf{p} = Z^* \mathbf{K} \mathbf{H} \mathbf{K}^{-1} \mathbf{p}^* + \varepsilon \mathbf{K} \mathbf{t} \quad (2.15)$$

La nature de cette dernière relation n'est pas changée si on multiplie chacun de ces membres par un scalaire α non nul. En tenant compte de cette remarque, on écrira dans la suite l'équation 2.15 sous la forme plus compacte suivante :

$$\alpha \mathbf{p} = \mathbf{G} \mathbf{p}^* + \beta \mathbf{e} \quad (2.16)$$

où $\forall k \neq 0$;

- $\alpha = kZ/Z^*$ est un scalaire non nul qui peut être choisi toujours positif;
- $\beta = k\varepsilon/Z^*$ est un scalaire qui est nul si le point appartient au plan de référence;
- $\mathbf{G} = k\mathbf{KHK}^{-1}$ est l'homographie exprimée dans le système de coordonnées en pixels (appelée plus fréquemment : *matrice de colinéation*);
- $\mathbf{e} = \mathbf{Kt}$ est l'épipole dans l'image courante (c'est-à-dire la projection de \mathcal{O}^* dans l'image courante).

A partir de seulement deux images, et sans connaissance à priori sur la géométrie de l'objet ou sur le mouvement de la caméra, il est impossible de déterminer le facteur d'échelle k .

Considérons maintenant trois points **3D** \mathcal{P}_i $\{i = 1, 2, 3\}$ de l'objet. Ces points, choisis de manière à ne pas être colinéaires dans les images, définissent un plan de référence noté π . La structure de la cible étant inconnue dans le cas général, tous les points pourraient être sur le même plan. On suppose pour le moment qu'il existe un certain nombre de points n'appartenant pas au plan de référence et on verra par la suite comment traiter le cas d'un objet planaire. Si un point \mathcal{P}_j n'appartient pas à π , l'intersection de la droite $(\mathcal{O}^*\mathcal{P}_j)$ et du plan π est le point **3D** noté \mathcal{P}'_j (voir la Figure 2-4). \mathcal{P}_j et \mathcal{P}'_j se projettent au même point \mathbf{p}_j^* dans l'image désirée et en deux points différents (\mathbf{p}_j et le point virtuel $\mathbf{p}'_j = \mathbf{G}\mathbf{p}_j^*$) dans l'image courante (ce qu'on appelle l'effet de *parallaxe*). La droite définie par \mathbf{p}_j et \mathbf{p}'_j est représentée, en coordonnées projectives, par $\mathbf{l}_j = \mathbf{p}_j \wedge \mathbf{p}'_j$ [13], soit encore :

$$\mathbf{l}_j = \mathbf{p}_j \wedge \mathbf{G}\mathbf{p}_j^* \quad (2.17)$$

Les quatre points \mathcal{O} , \mathcal{O}^* , \mathcal{P}_j et \mathcal{P}'_j étant coplanaires, les points \mathbf{p}_j , \mathbf{p}'_j et l'épipole \mathbf{e} sont donc colinéaires, et l'on a :

$$\mathbf{l}_j^T \mathbf{e}_j = 0 \quad (2.18)$$

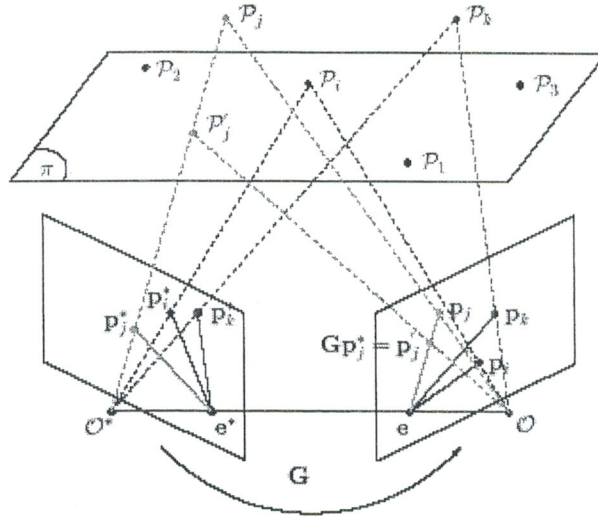


FIG. 2-4: Contrainte épipolaire et parallaxe virtuelle

En utilisant l'équation 2.17, on obtient :

$$\mathbf{p}_j \mathbf{F} \mathbf{p}_j^* = 0 \quad (2.19)$$

où $\mathbf{F} = [\mathbf{e}]_x \mathbf{G}$ est la *matrice fondamentale* ($[\mathbf{e}]_x$ est la matrice du préproduit vectoriel associée au vecteur \mathbf{e}). Dans le cas général, \mathbf{F} est de rang 2 et $\det(\mathbf{F}) = 0$, ce qui implique une contrainte non-linéaire sur les éléments de \mathbf{F} . La littérature concernant les propriétés, l'estimation et l'utilisation de la matrice fondamentale dans le domaine de la vision par ordinateur est très importante. On pourra par exemple se reporter à [13] [27] pour plus de détails.

Dans le système des coordonnées images métriques, la **matrice essentielle** \mathbf{E} introduite par H.C. Longuet-Higgins, joue un rôle similaire à la matrice fondamentale [14][20]. Elle vérifie la relation :

$$\mathbf{m}^T \mathbf{E} \mathbf{m}_f = 0 \quad (2.20)$$

Les différentes expressions matricielles exprimant la contrainte épipolaire liant les projections dans deux images d'un même point 3D sont très importantes et permettent une résolution élégante de problèmes difficiles dans le domaine de la vision. Nous allons, dans le paragraphe suivant, donner un bref aperçu des méthodes d'estimation de la géométrie épipolaire (c'est-à-dire des matrices \mathbf{G} , \mathbf{H} , \mathbf{F} , \mathbf{E} et du vecteur \mathbf{e}) et de son application dans l'estimation du déplacement partiel de la caméra et de la géométrie de la scène.

2.5 Estimation de la géométrie épipolaire

Nous considérons uniquement les cas où la caméra fournit une image sans distorsion optique ou géométrique et est décrite par le modèle sténopé. On pourra se reporter à [40] lorsque ces hypothèses ne sont pas valides. Nous ne présentons ici qu'un bref aperçu des méthodes d'estimation de la géométrie épipolaire. En revanche, nous invitons les lecteurs désirant plus de précisions à se reporter à [24][39].

Nous commençons cette section par présenter un bref état de l'art des techniques qui ont été utilisées pour l'estimation de la géométrie épipolaire. Nous donnerons ensuite les deux méthodes les plus utilisées dans le cadre de l'asservissement visuel, à savoir l'algorithme des huit points qui permet d'estimer la matrice fondamentale \mathbf{F} et l'algorithme d'estimation de la matrice d'homographie \mathbf{H} utilisé dans le cas particulier, où l'objet observé par la caméra est planaire (tel que les *Landmarks*).

2.5.1 Etat de l'art [27][28]

A partir de la mise en correspondance d'un ensemble de n points dans les deux images, il est possible de calculer la matrice fondamentale \mathbf{F} et/ou la colinéation \mathbf{G} relative à un plan de référence Π . Cela nécessite, dans le cas général, un nombre de points $n \geq 8$, et un nombre de points $n \geq 4$ dans le cas particulier où les points considérés sont coplanaires. L'estimation de la géométrie épipolaire est un problème, par nature, non linéaire. La matrice fondamentale doit être de rang 2 et la matrice essentielle doit satisfaire les conditions de Huang-Faugeras [16]. L'approche classique pour estimer la géométrie épipolaire est en général composée de deux étapes. Dans un premier temps, un algorithme linéaire fournit une solution. Cette solution est ensuite affinée en introduisant les contraintes non linéaires (par exemple imposer un rang de 2 à la matrice fondamentale) via un algorithme de minimisation. La solution obtenue en utilisant l'algorithme linéaire des huit points proposé dans [20][14] peut, par exemple, être utilisée pour initialiser les algorithmes non linéaires décrits dans [21].

Afin de simplifier l'estimation de la matrice fondamentale, celle-ci peut être décomposée en utilisant la relation $\mathbf{F} = [\mathbf{e}]_x \mathbf{G}$. L'estimation de la matrice fondamentale revient alors à estimer l'épipole et la matrice de colinéation. En effectuant un changement de base judicieux, Boufama et *al.* ont proposé d'estimer simultanément la matrice de colinéation et l'épipole, ce qui permet d'obtenir directement une matrice fondamentale de rang 2 [1]. Shashua et *al.* proposent d'estimer dans un premier temps l'épipole et ensuite la matrice d'homographie [35]. Si on cherche à estimer uniquement \mathbf{G} , on pourra par exemple utiliser les algorithmes décrits dans [26]. Lorsqu'on dispose de quatre points coplanaires, le problème est plus simple puisqu'il se résume à la résolution d'un système linéaire [9].

Certaines configurations entraînent une dégénérescence (réduction du rang) de la matrice fondamentale. Ces configurations sont décrites dans [21] ; parmi les plus fréquentes, notons les trois suivantes :

- lorsque le déplacement de la caméra est une rotation pure,
- lorsque la scène est composée d'un plan unique,
- lorsque tous les points **3D** se projettent sur une droite unique.

Dans le contexte de l'asservissement visuel, ces configurations ne sont pas rares. Notons en particulier qu'au voisinage de la position désirée, le déplacement à réaliser doit être nul. Nous nous trouvons alors dans la configuration du premier point cité précédemment. Nous préférons donc l'utilisation de la colinéation à celle de la matrice fondamentale, la colinéation relative à un plan ayant l'avantage de ne jamais être dégénérée. La matrice \mathbf{G} peut être estimée à partir d'un ensemble de droites mises en correspondance dans les deux images, ou encore en utilisant des contours modélisés par des fonctions polynomiales [13].

Pour estimer la colinéation relative au plan à l'infini, il faut disposer d'au moins quatre images de la scène ou de connaissances supplémentaires sur la scène, telles que trois points situés à l'infini, ou encore utiliser un plan de référence parallèle au plan image. Pour plus de précisions sur l'estimation de cette colinéation particulières, on pourra se reporter à [37]. Lorsque le mouvement de la caméra est une pure rotation ou une pure translation alors la matrice de colinéation à l'infini \mathbf{G}_∞ est facilement obtenue. En effet, dans le premier cas elle correspond à la matrice de colinéation \mathbf{G} relative à un plan quelconque de la scène et dans le second cas elle est proportionnelle à la matrice d'identité ($\mathbf{G}_\infty \propto \mathbf{I}$).

On présente maintenant quelques méthodes d'estimation de la géométrie épipolaire les plus intéressantes (du point de vue temps de calcul) et les plus utilisées dans le contexte de l'asservissement visuel.

2.5.2 Estimation linéaire de la matrice fondamentale [23]

L'approche classique pour estimer la géométrie épipolaire est connue sous le nom d'algorithme des huit points [20][14]. En effet, il est possible de simplifier le problème en effectuant une estimation linéaire des éléments de la matrice fondamentale à partir de l'équation 2.19 :

$$u_j^* u_j f_{11} + v_j^* u_j f_{12} + u_j f_{13} + u_j^* v_j f_{21} + v_j^* v_j f_{22} + u_j^* f_{31} + v_j f_{23} + v_j^* f_{32} + f_{33} = 0 \quad (2.21)$$

Si on considère, par exemple, une image dont la taille est 500×500 pixels, on remarque que les coefficients associés à f_{11} , f_{12} , f_{21} et f_{22} sont de l'ordre de 250000, les coefficients associés à f_{13} , f_{23} , f_{31} et f_{32} sont de l'ordre de 500, alors que le coefficient de f_{33} est toujours égal à 1. L'importance d'une normalisation de données à été mise en évidence par Hartley [14], car en fait, afin d'obtenir une bonne estimation de la matrice fondamentale, il est nécessaire d'effectuer un changement de coordonnées permettant d'avoir des coefficients ayant à peu près tous le même ordre de grandeur.

L'équation 2.19 est valable pour chaque couple de points $(\mathbf{p}_j, \mathbf{p}_j^*)$. Si l'on dispose de n couples, il est donc possible d'obtenir un système linéaire :

$$\mathbf{C}\mathbf{f} = 0 \quad (2.22)$$

où \mathbf{C} est la matrice des coefficients de dimension $(n \times 9)$ et \mathbf{f} est un vecteur contenant les 9 éléments de \mathbf{F} . Le système 2.22 est homogène et, comme \mathbf{F} est définie à un facteur d'échelle près, un minimum de 8 couples de points est nécessaire pour le résoudre. En présence de bruit, le problème d'estimation linéarisé peut s'écrire :

$$\min_{\mathbf{f}} \|\mathbf{C}\mathbf{f}\| \quad \text{sous contrainte } \|\mathbf{f}\| = 1 \quad (2.23)$$

La solution de ce problème est obtenue à partir de la décomposition en valeurs singulières (SVD) de la matrice $\mathbf{C}^T\mathbf{C} = \mathbf{V}\mathbf{S}\mathbf{V}^T$, où \mathbf{S} est une matrice diagonale. La solution $\bar{\mathbf{f}}$ du système est la colonne de \mathbf{V} correspondant à la valeur singulière minimale (0 en absence du bruit).

Cet algorithme ne tient pas compte de la contrainte que la matrice fondamentale doit être de rang 2. Cette contrainte est généralement prise en compte en utilisant un algorithme non linéaire avec comme première estimée le résultat du système linéaire. Comme notre objectif est de réaliser des tâches robotiques à la cadence vidéo, les méthodes non linéaires, généralement gourmandes en temps de calcul, ne semblent pas adaptées.

2.5.3 Estimation de l'homographie à partir de points ou droites coplanaires [24]

La méthode qu'on va présenter est purement linéaire et elle permet d'estimer la matrice d'homographie d'une manière très simple :

Cas de points coplanaires

Pour tous les points appartenant au plan de référence, l'équation 2.16 s'écrit sous la forme suivante :

$$\alpha_i \mathbf{p}_i = \mathbf{G} \mathbf{p}_i^* , \quad \{i = 1, 2, 3, \dots, p\} \quad (2.24)$$

puisqu'alors $\beta = 0$. On rappelle que la matrice d'homographie \mathbf{G} est définie à un facteur d'échelle près. Un des éléments de \mathbf{G} ou son déterminant peut donc être fixé égal à 1 sans perte de généralité. On obtient donc un système linéaire à $n + 8$ inconnues et $3n$ équations, que l'on peut résoudre si l'on dispose d'au moins quatre points.

Cas de droites coplanaires

L'équation de ces droites peut être obtenue à partir de points ou estimée directement dans l'image. La droite \mathbf{l}_i passant par les deux points \mathbf{p}_i et \mathbf{p}_0 s'écrit :

$$\mathbf{l}_i = \mathbf{p}_i \wedge \mathbf{p}_0 \quad (2.25)$$

Comme les deux points sont coplanaires, ils sont liés par la matrice d'homographie, et cette équation devient :

$$\mathbf{l}_i = \mathbf{G} \mathbf{p}_i^* \wedge \mathbf{G} \mathbf{p}_0^* \quad (2.26)$$

et comme $[\mathbf{G}\mathbf{v}]_x = \det(\mathbf{G}) \mathbf{G}^{-T} [\mathbf{v}]_x \mathbf{G}^{-1}$, on a :

$$\mathbf{l}_i = \det(\mathbf{G}) \mathbf{G}^{-T} \mathbf{p}_i^* \wedge \mathbf{p}_0^* \quad (2.27)$$

Si on note $\mathbf{l}_i^* = \mathbf{p}_i^* \wedge \mathbf{p}_0^*$ l'équation de la droite dans l'image désirée, la relation entre les deux droites s'écrit :

$$\alpha'_i \mathbf{l}_i = \mathbf{G}^{-T} \mathbf{l}_i^* , \quad \{i = 1, 2, 3, \dots, l\} \quad (2.28)$$

où α'_i est un scalaire non nul et \mathbf{G}^{-T} est la matrice *duale* de \mathbf{G} . Encore une fois, si l'on dispose d'au moins quatre droites mises en correspondance dans les deux images, il est possible d'estimer la matrice d'homographie duale (et donc la matrice d'homographie) par simple résolution d'un système linéaire.

2.6 Reconstruction des paramètres tridimensionnels

Une fois la géométrie épipolaire obtenue, il est possible de remonter à la géométrie euclidienne. C'est le problème bien connu de "structure from motion". On pourra donc obtenir une reconstruction géométrique de la scène observée par la caméra ("structure"), et le déplacement partiel de la caméra entre deux prises de vue ("motion"). Plus précisément, connaissant la géométrie épipolaire entre deux images I_c et I_f (c'est-à-dire la matrice fondamentale \mathbf{F} ou la matrice de colinéation \mathbf{G} relative à un plan de référence), on peut estimer la rotation de la caméra et sa translation à un facteur d'échelle près. La première étape consiste en l'estimation des matrices essentielle et/ou d'homographie à partir des matrices fondamentale et/ou de colinéation.

Il faut cependant distinguer deux cas de figure dans cette estimation :

- soit les paramètres intrinsèques de la caméra sont exactement connus (ce qui implique une calibration parfaite). Dans ce cas, on peut estimer exactement la rotation et la translation (à un facteur d'échelle près) de la caméra et la structure de la scène (à un facteur d'échelle près). Une information particulièrement intéressante qu'on peut obtenir est le rapport entre deux profondeurs, qui est donc indépendant du facteur d'échelle,
- soit les paramètres intrinsèques sont entachés d'erreurs (ce qui correspond à une calibration grossière de la caméra, plus réaliste d'un point de vue pratique). Dans ce cas, l'estimation du déplacement de la caméra et de la structure de la scène sera elle aussi entachée d'erreurs, mais grâce à la connaissance de la matrice d'homographie du plan de l'infini, certains paramètres, tels que l'angle de rotation ou le rapport entre les profondeurs, peuvent encore être estimés exactement.

L'influence des erreurs de modélisation et de calibration sur la reconstruction partielle du déplacement de la caméra a été étudiée dans plusieurs travaux en l'asservissement visuel [23][24]. Les résultats obtenus par les auteurs montrent qu'il est toujours possible de trouver des relations entre les variations des paramètres tridimensionnels reconstruits et les variations dans les paramètres intrinsèques et extrinsèques de la caméra, et ce dans le cas d'objets de forme quelconque. Par contre, dans le cas où les objets d'intérêt sont planaires, les travaux effectués ont montré l'existence d'une forte robustesse dans l'estimation du déplacement euclidien vis-à-vis les erreurs de modélisation et de calibration. On pourra, par exemple, consulter les travaux de E. Malis sur l'asservissement visuel 2D1/2 des robots manipulateurs pour plus de détails [25][26].

Dans la suite, nous présentons les méthodes utilisées pour déterminer les paramètres 3D de la scène et du déplacement de la caméra à partir de la géométrie épipolaire. Il existe deux approches

principales dans la littérature : dans la première approche, la reconstruction des paramètres tridimensionnels est effectuée à partir de la matrice d'homographie. Par contre, dans la deuxième approche, la reconstruction est faite à partir de la matrice essentielle. Dans les deux cas, les résultats sont obtenues en utilisant la décomposition en valeurs singulières **SVD** de ces deux matrices.

Nous ne détaillerons que la méthode de reconstruction à partir de la matrice d'homographie. Les résultats dans l'autre méthode peuvent être obtenus de manière similaire.

2.6.1 Reconstruction à partir de la matrice d'homographie [12]

La matrice correspondant à la matrice d'homographie **G** dans le domaine calibré est la matrice d'homographie **H**. Celle-ci peut être estimée de la manière suivante :

$$\hat{\mathbf{H}} = \mathbf{K}^{-1} \mathbf{G} \mathbf{K} \quad (2.29)$$

De plus, la matrice d'homographie réelle peut s'écrire comme la somme d'une matrice de rotation et d'une matrice de rang 1 :

$$\hat{\mathbf{H}} = k \mathbf{H} = k \left(\mathbf{R} + \frac{t}{d^*} \mathbf{n}^{*T} \right) \quad (2.30)$$

A partir de la matrice d'homographie estimée, on peut extraire les paramètres tridimensionnels suivants :

- les composantes du vecteur **n** normal au plan π^* ,
- la matrice de rotation **R**,
- le rapport $\frac{t}{d}$ de la translation sur la distance au plan.

Du fait que toutes les distances dans l'espace projectif sont définies à un facteur d'échelle près, on peut écrire la matrice d'homographie sous la forme suivante :

$$\mathbf{H} = \mathbf{R} + \frac{t}{d^*} \mathbf{n}^{*T} = d^* \mathbf{R} + t \mathbf{n}^{*T} \quad (2.31)$$

En utilisant la décomposition en valeurs singulières **SVD**, **H** peut s'écrire de la manière suivante :

$$\mathbf{H} = \mathbf{U} \mathbf{S} \mathbf{V}^T$$

où **S** est une matrice diagonale et **U** et **V** sont des matrices orthonormales (c'est-à-dire $\mathbf{U}^T \mathbf{U} = \mathbf{V}^T \mathbf{V} = \mathbf{I}$). Les éléments de **S** sont les valeurs singulières de **H**, c'est-à-dire les racines carrées des valeurs propres

de $\mathbf{H}\mathbf{H}^T$. Ces valeurs propres λ_i sont positives et peuvent être classées dans un ordre décroissant $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

En utilisant cette décomposition, on obtient la nouvelle équation :

$$\mathbf{S} = d' \mathbf{R}' + \mathbf{t}' \mathbf{n}'^T \quad (2.32)$$

où \mathbf{R} , \mathbf{t} et \mathbf{n}^* sont liés à \mathbf{R}' , \mathbf{t}' et \mathbf{n}' par les relations suivantes :

$$\begin{cases} \mathbf{R} = s \mathbf{U} \mathbf{R}' \mathbf{V}^T \\ \mathbf{t} = \mathbf{U} \mathbf{t}' \\ \mathbf{n}^* = \mathbf{V} \mathbf{n}' \\ d^* = s d' \\ s = \det(\mathbf{U}) \det(\mathbf{V}) \end{cases} \quad (2.33)$$

Notons que \mathbf{R}' est aussi une matrice de rotation (c'est-à-dire $\mathbf{R}' \mathbf{R}'^T = \mathbf{I}$ et $\det(\mathbf{R}') = 1$).

Si $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ est la base canonique du repère de référence et en écrivant $\mathbf{n}' = x_1 \mathbf{e}_1 + x_2 \mathbf{e}_2 + x_3 \mathbf{e}_3$, l'équation 2.32 nous donne les trois équations suivantes :

$$\lambda_i \mathbf{e}_i = d' \mathbf{R}' \mathbf{e}_i + \mathbf{t}' x_i \quad \text{pour } i = 1, 2, 3 \quad (2.34)$$

Notons que du fait que \mathbf{n}^* a une norme unitaire et que \mathbf{V} est orthogonale, \mathbf{n}' a aussi une norme unitaire, c'est-à-dire $\sum_{i=1}^3 x_i^2 = 1$. L'élimination de \mathbf{t}' à partir des équations précédentes nous donne :

$$d' \mathbf{R}' (x_j \mathbf{e}_i - x_i \mathbf{e}_j) = \lambda_i x_j \mathbf{e}_i - \lambda_j x_i \mathbf{e}_j \quad \text{pour } i \neq j \quad (2.35)$$

Puisque \mathbf{R}' ne modifie pas la norme vectorielle, on obtient l'ensemble d'équations suivant :

$$\begin{cases} (d'^2 - \lambda_2^2) x_1^2 + (d'^2 - \lambda_1^2) x_2^2 = 0 \\ (d'^2 - \lambda_3^2) x_2^2 + (d'^2 - \lambda_2^2) x_3^2 = 0 \\ (d'^2 - \lambda_1^2) x_3^2 + (d'^2 - \lambda_3^2) x_1^2 = 0 \end{cases} \quad (2.36)$$

On peut considérer cet ensemble d'équations comme un système linéaire par rapport aux inconnues x_1^2 , x_2^2 et x_3^2 . Pour que ce système ait une solution non nulle, son déterminant doit être égal à zéro :

$$(d'^2 - \lambda_1^2)(d'^2 - \lambda_2^2)(d'^2 - \lambda_3^2) = 0 \quad (2.37)$$

On obtient donc différents cas, suivant le degré de multiplicité des valeurs singulières de \mathbf{H} :

1. $\lambda_1 \neq \lambda_2 \neq \lambda_3$ et donc $d' = \pm\lambda_2$,
2. $\lambda_1 = \lambda_2 \neq \lambda_3$ ou $\lambda_1 \neq \lambda_2 = \lambda_3$ et $d' = \pm\lambda_2$: dans ce cas, on considère seulement le cas où $\lambda_1 = \lambda_2$, l'autre possibilité étant symétrique,
3. $\lambda_1 = \lambda_2 = \lambda_3$ et $d' = \pm\lambda_2$.

Les solutions $d' = \pm\lambda_1$ et $d' = \pm\lambda_3$ sont en fait impossibles. Essayons de le prouver le dans le premier cas par exemple :

En supposant que $d' = \lambda_1$, l'équation 2.36 nous donne :

$$x_1 = 0 \quad \text{et} \quad (\lambda_1^2 - \lambda_3^2)x_2^2 + (\lambda_1^2 - \lambda_2^2)x_3^2 = 0$$

Puisque $\lambda_1 > \lambda_2 > \lambda_3$, ceci implique que $x_2 = x_3 = 0$, ce qui est impossible car \mathbf{n}' a une norme unitaire.

Si $\lambda_1 \neq \lambda_3$, et en utilisant le fait que \mathbf{n}' a une norme unitaire, on peut exprimer x_1 , x_2 et x_3 à partir de l'équation 2.36 de la manière suivante :

$$\begin{cases} x_1 = \varepsilon_1 \sqrt{\frac{\lambda_1^2 - \lambda_2^2}{\lambda_1^2 - \lambda_3^2}} \\ x_2 = 0 \\ x_3 = \varepsilon_3 \sqrt{\frac{\lambda_2^2 - \lambda_3^2}{\lambda_1^2 - \lambda_3^2}} \end{cases} \quad \varepsilon_1, \varepsilon_3 = \pm 1 \quad (2.38)$$

Etudions maintenant en détail ces trois cas. L'étude est divisée en deux parties suivant le signe de d' .

- Si $d' > 0$:

1. Suivant l'équation 2.34, on obtient : $\mathbf{R}'\mathbf{e}_2 = \mathbf{e}_2$. Donc \mathbf{R}' est une rotation autour de l'axe \mathbf{e}_2 .

On peut calculer cette matrice comme suit :

$$\mathbf{R}' = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.39)$$

A partir des équations 2.35 et 2.38, on trouve :

$$\begin{cases} \sin \theta = (\lambda_1 - \lambda_3) \frac{x_1 x_3}{\lambda_2} = \varepsilon_1 \varepsilon_3 \frac{\sqrt{(\lambda_1^2 - \lambda_2^2)(\lambda_2^2 - \lambda_3^2)}}{(\lambda_1 + \lambda_3)\lambda_2} \\ \cos \theta = \frac{\lambda_1 x_3^2 + \lambda_3 x_1^2}{\lambda_2} = \frac{\lambda_2^2 + \lambda_1 \lambda_3}{(\lambda_1 + \lambda_3)\lambda_2} \end{cases} \quad (2.40)$$

En remplaçant ces deux valeurs dans l'équation 2.34, on aboutit à :

$$\mathbf{t}' = (\lambda_1 - \lambda_3) \begin{bmatrix} x_1 \\ 0 \\ -x_3 \end{bmatrix} \quad (2.41)$$

2. Dans ce cas, on obtient $x_1 = x_2 = 0$ et $x_3 = \pm 1$, ce qui nous donne :

$$\begin{cases} \mathbf{R}' = \mathbf{I} \\ \mathbf{t}' = (\lambda_1 - \lambda_3)\mathbf{n}' \end{cases}$$

3. x_1, x_2 et x_3 ne sont pas définis et les équations 2.34 et 2.35 nous donnent :

$$\begin{cases} \mathbf{R}' = \mathbf{I} \\ \mathbf{t}' = 0 \end{cases}$$

Dans ce cas, le mouvement est une rotation pure et le vecteur normal est indéfini.

- Si $d' < 0$: en utilisant les mêmes méthodes, on obtient :

1. Les équations 2.34 nous donnent $\mathbf{R}'\mathbf{e}_2 = -\mathbf{e}_2$, ce qui implique que \mathbf{R}' est une symétrie (c'est-à-dire une rotation d'angle π) par rapport à un axe v perpendiculaire à \mathbf{e}_2 , c'est-à-dire dans le plan $(\mathbf{e}_1, \mathbf{e}_3)$. Soit $\varphi/2$ l'angle entre \mathbf{e}_1 et v . La matrice \mathbf{R}' peut ainsi s'écrire :

$$\mathbf{R}' = \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & -1 & 0 \\ \sin \varphi & 0 & -\cos \varphi \end{bmatrix} \quad (2.42)$$

avec :

$$\begin{cases} \sin \varphi = \frac{(\lambda_1 - \lambda_3)}{\lambda_2} x_1 x_3 = \varepsilon_1 \varepsilon_3 \frac{\sqrt{(\lambda_1^2 - \lambda_2^2)(\lambda_2^2 - \lambda_3^2)}}{(\lambda_1 - \lambda_3)\lambda_2} \\ \cos \varphi = \frac{\lambda_3 x_1^2 - \lambda_1 x_3^2}{\lambda_2} = \frac{\lambda_1 \lambda_3 - \lambda_2^2}{(\lambda_1 - \lambda_3)\lambda_2} \end{cases} \quad (2.43)$$

Et en remplaçant ces équations dans l'équation 2.34, on aboutit à :

$$\mathbf{t}' = (\lambda_1 + \lambda_3) \begin{bmatrix} x_1 \\ 0 \\ x_3 \end{bmatrix} \quad (2.44)$$

2. On a, encore une fois, $x_1 = x_2 = 0$ et $x_3 = \pm 1$. \mathbf{R}' est donc une symétrie par rapport à l'axe \mathbf{e}_3 :

$$\mathbf{R}' = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{t}' = (\lambda_1 + \lambda_3)\mathbf{n}'$$

3. L'équation 2.32 implique que $\mathbf{R}'\mathbf{x} = -\mathbf{x}$ pour tout vecteur \mathbf{x} dans le plan orthogonal à \mathbf{n}' . \mathbf{R}' est donc une symétrie par rapport à l'axe \mathbf{n}' . Ainsi, $\mathbf{R}' = -\mathbf{I} + 2\mathbf{n}'\mathbf{n}'^T$ et, à partir de l'équation 2.32, $\mathbf{t}' = -2d'\mathbf{n}'$.

Du fait qu'il n'y a aucun moyen de connaître le signe de d' à priori, nous avons en tout huit solutions dans le cas général (si les valeurs singulières sont distinctes). Mais si on revient à l'interprétation physique des résultats, on ne garde que deux solutions à la fin.

Proposition 2.1 *Puisque les points observés sont visualisés par les deux caméras, le problème de décomposition en valeurs singulières a seulement deux solutions physiques dans le cas général, et une seule lorsqu'il y a une valeur singulière double.*

Preuve. Soit \mathcal{P} un point dans le plan, visible à partir des deux caméras. Z et Z^* sont ces coordonnées suivant les axes des Z des deux repères associés aux deux caméras. La contrainte de visibilité dans les deux caméras impose que $Z > 0$ et $Z^* > 0$.

Avant de passer à la preuve, on rappelle les résultats suivants :

Soient $\mathbf{x} = [X \ Y \ Z \ 1]^T$ et $\mathbf{x}^* = [X^* \ Y^* \ Z^* \ 1]^T$ les vecteurs des coordonnées 3D du point \mathcal{P} dans les deux repères \mathcal{F} et \mathcal{F}^* , et soient $\mathbf{m} = [x \ y \ 1]^T$ et $\mathbf{m}^* = [x^* \ y^* \ 1]^T$ les vecteurs des coordonnées normalisées des projections du point \mathcal{P} sur les deux plans images.

Rappelons qu'à partir des équations de la projection perspective, on a $x^{(*)} = \frac{X^{(*)}}{Z^{(*)}}$ et $y^{(*)} = \frac{Y^{(*)}}{Z^{(*)}}$, ce qui nous donne :

$$\frac{X^{(*)}}{x^{(*)}} = \frac{Y^{(*)}}{y^{(*)}} = Z^{(*)} \quad (2.45)$$

et que, pour tous les points appartenant au plan de référence choisi, on a :

$$\mathbf{m}^* = \mathbf{H}\mathbf{m} \quad (2.46)$$

En combinant ces deux dernières équations, on obtient : $\alpha \mathbf{x}^* = \mathbf{H}\mathbf{x}$, pour $\alpha \in \mathcal{R}^*$. A partir de l'équation 2.31 et de l'équation $\mathbf{x}^* = \mathbf{R}\mathbf{x} + \mathbf{t} = \left(\mathbf{R} + \frac{\mathbf{t}\mathbf{n}^{*T}}{d^*}\right) \mathbf{x} = \mathbf{H}\mathbf{x}$ qui relie les coordonnées 3D du point dans les deux repères, on obtient $\alpha = d^*$. En plus, si :

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

alors :

$$\frac{Z^*}{Z} = \frac{h_{31}x + h_{32}y + h_{33}}{d^*} > 0$$

Z et Z^* maintiennent un signe constant quand \mathcal{P} varie dans le plan de référence, et d^* est constante, donc $h_{31}x + h_{32}y + h_{33}$ maintient aussi un signe constant quand \mathcal{P} varie dans le plan. Ainsi, la connaissance d'un point permet de déterminer le signe de d^* . Puisque l'on a : $d^* = sd'$ et du fait que s est déterminé à partir de \mathbf{H} , on a seulement quatre solutions suivant le signe de d' .

Les solutions correspondantes pour le vecteur normal \mathbf{n}^* sont \mathbf{n}_1^* , $-\mathbf{n}_1^*$, \mathbf{n}_2^* et $-\mathbf{n}_2^*$ (voir l'équation 2.38). \mathbf{x} appartient au plan de référence, tel que $\mathbf{n}^{*T}\mathbf{x} = d^*$. En utilisant l'équation 2.45, on obtient de la condition $Z > 0$ que :

$$\frac{\mathbf{n}^{*T}\mathbf{m}}{d^*} > 0$$

Ce qui ne laisse que deux solutions parmi les quatre obtenues. ■

Le résultat précédent a été déduit à partir de la contrainte de visibilité pour un point dans l'image. En écrivant cette contrainte pour tous les points observés, Longuet-Higgins a démontré la propriété intéressante suivante :

Proposition 2.2 [20] *Si tous les points observés ne sont pas assez proches du centre optique de l'une des caméras que de celui de l'autre caméra, alors une seule solution satisfait la condition de visibilité pour tous les points.*

Donc en résumé, la connaissance de \mathbf{H} permet d'obtenir huit solutions. En écrivant la contrainte de visibilité pour un point, on ne garde que deux solutions et si les points ne sont pas tous proches de l'une des caméras qu'il le sont par rapport à l'autre, alors la contrainte de visibilité ne laisse qu'une

seule solution.

En pratique, si on obtient plus qu'une solution, on doit utiliser des informations additionnelles pour éliminer l'indétermination. Par exemple, dans le cas d'une cible planaire, la connaissance du vecteur normal au plan de référence permet de résoudre l'ambiguïté. Si la cible n'est pas plane, l'indétermination est éliminée en considérant un autre plan de référence et en choisissant la solution commune aux deux couples. En asservissement visuel, le choix n'est à effectuer qu'une seule fois, à l'initialisation du processus, car le vecteur normal \mathbf{n}^* ne varie pas au cours de l'asservissement.

Il faut aussi remarquer que d'autres paramètres sur la structure de l'objet observé peuvent être reconstruits à partir de la connaissance de la matrice d'homographie. On peut citer, par exemple, les paramètres suivants :

- le rapport r entre les distances d et d^* des centres de projection \mathcal{O} et \mathcal{O}^* à π :

$$r = \frac{d}{d^*} = \det(\mathbf{H}) = 1 + \mathbf{n}^{*T} \mathbf{R} \mathbf{t}_{d^*} \quad (2.47)$$

avec $\mathbf{t}_{d^*} = \mathbf{t}/d^*$,

- le rapport entre la profondeur courante et désirée d'un point **3D** :

$$\begin{cases} \frac{Z}{Z^*} = \frac{\mathbf{n}^{*T} \mathbf{m}^*}{\mathbf{n}^T \mathbf{m}} r & \text{si } \mathbf{m} \in \pi \\ \frac{Z}{Z^*} = \frac{\|[\mathbf{t}]_x \mathbf{R} \mathbf{m}^*\|}{\|[\mathbf{t}]_x \mathbf{m}\|} r & \text{si } \mathbf{m} \notin \pi \end{cases} \quad (2.48)$$

- le rapport ρ entre la coordonnée Z d'un point **3D** et d^* :

$$\begin{cases} \rho = \frac{Z}{d^*} = \frac{r}{\mathbf{n}^T \mathbf{m}} & \text{si } \mathbf{m} \in \pi \\ \rho = \frac{Z}{d^*} = \frac{Z}{Z^*} \frac{\|\mathbf{t}/d^*\|}{\|\mathbf{t}/Z^*\|} & \text{si } \mathbf{m} \notin \pi \end{cases} \quad (2.49)$$

avec $\mathbf{t}/Z^* = \frac{Z}{Z^*} \mathbf{m} - \mathbf{R} \mathbf{m}^*$.

Naturellement, les mêmes paramètres peuvent être reconstruits à partir de l'estimation de la matrice essentielle. Toutefois, des résultats de simulation et expérimentaux dans les travaux en asservissement visuel **2D1/2** sur les robots manipulateurs ont montré que l'estimation est plus robuste si elle est effectuée à partir de la matrice d'homographie.

2.6.2 Reconstruction à partir de la matrice essentielle [25]

Si on considère que la caméra est calibrée, on peut utiliser les coordonnées métriques. Les deux images sont donc liées par la matrice essentielle \mathbf{E} . Cette dernière peut être estimée de la manière suivante :

$$\hat{\mathbf{e}} = \mathbf{K}^T \mathbf{F} \mathbf{K} \quad (2.50)$$

où la matrice \mathbf{K} est la matrice des paramètres intrinsèques donnée par l'équation 2.8. Le problème de l'estimation de la matrice essentielle est non linéaire car la matrice essentielle réelle \mathbf{E} doit vérifier les contraintes établies par Huang et Faugeras [16] : $\sigma_1 = \sigma_2$ et $\sigma_3 = 0$ (où σ_1 , σ_2 et σ_3 sont les valeurs singulières de \mathbf{E}). En effet, \mathbf{E} peut s'écrire comme le produit d'une matrice antisymétrique et d'une matrice de rotation :

$$\mathbf{E} = [\mathbf{t}]_x \mathbf{R} \quad (2.51)$$

La matrice essentielle estimée peut être décomposée en utilisant la **SVD** (décomposition en valeurs singulières) comme suit :

$$\hat{\mathbf{e}} = \mathbf{U} \hat{\mathbf{S}} \mathbf{V}^T \quad (2.52)$$

où \mathbf{U} et \mathbf{V} sont deux matrices orthonormales et $\hat{\mathbf{S}} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$ est une matrice diagonale. Après avoir imposé les contraintes de Huang-Faugeras ($\hat{\mathbf{S}}' = \text{diag}(\sigma, \sigma, 0)$ où on peut choisir par exemple $\sigma = (\sigma_1 + \sigma_2)/2$), la matrice $\hat{\mathbf{S}}'$ peut se décomposer de la manière suivante :

$$\hat{\mathbf{S}}' = [\mathbf{t}']_x \mathbf{R}' = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.53)$$

où $[\mathbf{t}']_x$ est une matrice antisymétrique et \mathbf{R}' est une matrice de rotation. Il faut souligner qu'il existe deux couples de solutions possibles. Finalement, la matrice essentielle est décomposée comme suit :

$$\mathbf{E} = (\mathbf{U} [\mathbf{t}']_x \mathbf{U}^T) (\mathbf{U} \mathbf{R}' \mathbf{V}^T) \quad (2.54)$$

Dans cette dernière équation, $(\mathbf{U} [\mathbf{t}']_x \mathbf{U}^T) = [\mathbf{t}]_x$ qui est la matrice de préproduit vectoriel associée au vecteur de translation \mathbf{t} et $(\mathbf{U} \mathbf{R}' \mathbf{V}^T) = \mathbf{R}$ qui est la matrice de rotation. Enfin, parmi les deux couples de solutions possibles, on peut choisir celle qui satisfait le fait que l'objet observé se trouve

devant la caméra.

Cette méthode de décomposition a été prouvée comme étant optimale par Hartley [14]. Cependant, si les contraintes de Huang-Faugeras ne sont pas respectées dans la matrice $\hat{\mathbf{S}}$, la solution obtenue sera évidemment entachée d'erreurs.

En conclusion, à partir de la matrice essentielle estimée, on peut directement obtenir la matrice de rotation \mathbf{R} et la direction de translation $\mathbf{t}/\|\mathbf{t}\|$. De plus, les paramètres extrinsèques suivant peuvent aussi être estimés : la translation \mathbf{t}/d^* à un facteur d'échelle près et le vecteur \mathbf{n}^* normal à π . Ces paramètres sont importants dans plusieurs schémas d'asservissement visuel.

2.7 Conclusion

Dans ce chapitre, nous avons modélisé l'environnement et le capteur caméra en plongeant les espaces de travail dans des espaces projectifs. Nous avons également vu qu'il est possible de remonter à la géométrie épipolaire liant deux images d'une scène statique. Les informations nécessaires à l'estimation de la géométrie épipolaire sont directement extraites de l'image et ne demandent pas de connaissance a priori sur la scène. Cependant, son estimation peut s'avérer délicate, en particulier dans le cas de la matrice fondamentale pour des configurations géométriques particulières (mouvements de rotations pures de la caméra, scène plane ...). Ces configurations étant très courantes dans le contexte de l'asservissement visuel, nous utiliserons plutôt la matrice de colinéation, celle-ci ayant l'avantage sur la matrice fondamentale de ne jamais être dégénérée.

L'estimation de la matrice de colinéation peut également être problématique dans le cas où les prises de vue sont très rapprochées les unes des autres (en raison de l'influence des bruits de mesure). Ce dernier point peut s'avérer très pénalisant, au voisinage de la position désirée, pour les techniques d'asservissement visuel où un calcul de l'homographie à chaque itération de la boucle de commande est nécessaire.

Pour éviter cela, nous effectuerons les calculs de la géométrie épipolaire entre les différentes images clés qu'on choisira suffisamment éloignées. Nos méthodes ne seront donc pas pénalisées par cette difficulté. Connaissant la géométrie épipolaire, nous avons également vu qu'il est possible de remonter au déplacement partiel de la caméra et d'obtenir des informations sur la géométrie de la scène. Cette reconstruction tridimensionnelle de l'état du système nous permettra de le localiser dans son environnement, opération indispensable dans de tels schémas de commande.

Chapitre 3

Modélisation et commande du robot mobile

Le but de ce chapitre est la synthèse de la loi de commande permettant d'effectuer le suivi de trajectoire pour le robot mobile non holonome considéré. En Automatique, l'utilisation de commandes en boucle fermée permet d'améliorer les performances du système et d'assurer le bon accomplissement de la tâche même en présence de perturbations externes et/ou d'erreurs de modélisation. Pour cela, les mesures fournies en temps réel par les capteurs du robot sont utilisées pour reconstruire l'état du système. Dans le reste de ce chapitre, on supposera que seulement la composante de l'état exprimant la rotation du système par rapport à sa trajectoire désirée est connue avec exactitude. L'état exprimant la distance entre le véhicule et la trajectoire (déviations latérales du robot par rapport à la trajectoire) n'est connu qu'à un facteur d'échelle près. Ces deux grandeurs sont supposées être déterminées par les calculs de l'homographie entre les différentes images relais définissant la trajectoire à suivre (voir le chapitre 2).

Notre étude sera limitée au cas d'un environnement sans obstacles dynamiques. Dans cette perspective, le contrôleur aura la tâche élémentaire d'exécuter les mouvements planifiés. Les obstacles statiques pourraient être pris en charge s'ils auraient été présents lors de la phase d'apprentissage effectuée avant le début de la navigation (voir la section 1.3).

La nature non holonome du véhicule tricycle est reliée à la supposition que les roues du robot roulent sans glissement sur le sol. Ceci implique la présence d'un ensemble de contraintes différentielles non intégrables, du premier ordre, sur les variables de configuration. Bien que ces contraintes non holonomes réduisent les mouvements instantanés que le véhicule peut effectuer, elles n'affectent pas sa commandabilité globale dans l'espace des configurations. Cette caractéristique unique conduit

à un déficit dans la synthèse des lois de commande qui vient s'aligner avec les nouvelles perspectives de recherche en planification de mouvements non holonomes. En effet, les applications de navigation de robots mobiles non holonomes ont déclenché la recherche de contrôleurs innovés qui peuvent être utilisés même pour d'autres classes de systèmes non linéaires.

Nous passons maintenant à la modélisation du robot mobile. Nous présenterons le modèle particulier utilisé dans le contexte de suivi de trajectoire. On passera ensuite à la synthèse de la loi de commande la plus adaptée à notre problème après avoir démontré, naturellement, la commandabilité du système. Des précisions sur la nature des équations modélisant les systèmes non holonomes seront aussi données dans la suite.

3.1 Les contraintes non holonomes [30]

Les principales difficultés dans la commande d'un robot mobile en comparaison avec un robot manipulateur proviennent de la présence de contraintes mécaniques spécifiques. L'étude du paramétrage d'un système mécanique permet de définir les liaisons qui sont des contraintes limitant les possibilités de mouvement des solides constitutifs du système. Une liaison se traduit, schématiquement, par des relations de dépendances entre les paramètres du système \mathbf{q} , les vitesses généralisées $\dot{\mathbf{q}}$ et le temps t .

Dans le cas des bras manipulateurs, les solides constituant le robot ne sont astreints qu'à des conditions géométriques de contact entre eux. Ces relations de liaison ne font pas intervenir les vitesses. Elles sont dites *liaisons holonomes* et peuvent s'exprimer par des équations de la forme :

$$f(\mathbf{q}, t) = 0 \quad (3.1)$$

En revanche, dans le cas des robots mobiles à roues, en plus des conditions géométriques de contact, les solides peuvent être astreints à des conditions cinématiques de contact. Les relations de liaisons font alors intervenir les vitesses généralisées. Ces liaisons sont dénommées *liaisons non-holonomes* et leur comportement est régi par des contraintes de la forme :

$$f(\mathbf{q}, \dot{\mathbf{q}}, t) = 0 \quad (3.2)$$

On remarque qu'une liaison non-holonyme nécessite au préalable une liaison holonome traduisant le contact purement géométrique. Pour illustrer ce type de liaison, considérons une roue de centre C de coordonnées (x, y) et de rayon R dans un repère \mathcal{R}_0 (voir la figure 3-1).

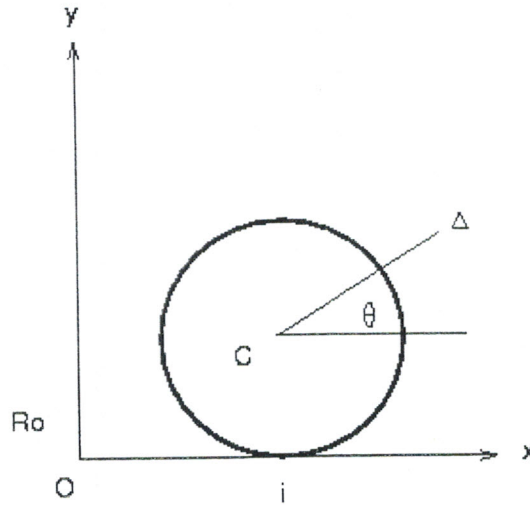


FIG. 3-1: Exemple d'une roue

La condition de contact géométrique est :

$$y = R \quad (3.3)$$

La condition cinématique de roulement sans glissement s'écrit :

$$\vec{V}(i) = 0 \quad (3.4)$$

où $\vec{V}(i)$ représente la vitesse de la roue au point i .

En utilisant le paramétrage (x, θ) illustré dans la figure 3-1, la relation dite parfois *semi-holonyme* s'écrit :

$$\dot{x} + R\dot{\theta} = 0 \quad (3.5)$$

Dans la pratique, la présence de liaisons non-holonomes dans un mécanisme se traduit par le fait que pour une configuration donnée du mécanisme, il existe certaines directions vers lesquelles ce mécanisme ne pourra pas se déplacer de façon instantanée. Par exemple, dans le cas d'une voiture, cette direction est donnée par l'axe liant les roues non directrices. Il est à noter toutefois que l'existence de ces contraintes n'entraîne pas une impossibilité d'atteindre une configuration désirée ; celle-ci pouvant être atteinte par le biais de manoeuvres. Par contre, l'existence de ces liaisons non-holonomes pose des problèmes difficiles au niveau du contrôle de tels mécanismes.

3.2 Modélisation

3.2.1 Modèle cinématique [2]

Dans ce problème de suivi de trajectoire, nous allons utiliser le modèle cinématique plutôt que le modèle dynamique. Par analogie avec la commande des robots manipulateurs, cela revient à utiliser un modèle de commande en vitesse plutôt qu'un modèle de commande en couples. Les principales raisons de ce choix sont les suivantes :

- Le modèle cinématique est plus simple que le modèle dynamique. En particulier, il ne fait pas intervenir un certain nombre de fonctions matricielles dont la connaissance précise repose sur celle des paramètres inertiels du véhicule et de ses actionneurs (masse, moments d'inertie, coefficients de réduction des moteurs électriques, etc.). Nombreuses sont les applications pour lesquelles il n'est pas nécessaire de connaître l'ensemble de ces grandeurs avec précision,
- pour les robots à motorisation électrique, il est fréquent de disposer d'un asservissement "bas niveau" en vitesse sur les moteurs, dont la grandeur en entrée est une vitesse désirée et ayant pour rôle d'asservir la vitesse du moteur à celle désirée. Si cet asservissement est bien conçu, l'écart entre ces deux vitesses est toujours faible, de sorte que la vitesse désirée peut, à son tour, être interprétée comme une variable de commande libre. Cet asservissement en vitesse permettra donc de découpler la partie dynamique de la partie cinématique du véhicule.

Nous allons maintenant essayer de déterminer le modèle cinématique de la base de notre robot mobile qui est du type *tricycle*.

La configuration d'une base mobile est décrite par le vecteur $\mathbf{x}_r = [x_r \ y_r \ \theta_r]^T$ où le couple (x_r, y_r) désigne les coordonnées du point de référence \mathcal{M} du robot, situé au milieu de l'axe des roues motrices, par rapport au repère $\mathcal{F}(O, \vec{i}, \vec{j}, \vec{k})$ lié à la scène, et l'angle θ_r représente l'orientation du véhicule par rapport à l'axe (O, \vec{i}) de ce même repère (voir la figure 3-2). Ces trois variables sont nécessaires et suffisantes pour définir une situation quelconque du robot dans \mathcal{F} . Enfin, désignons par $\mathcal{F}_M(\mathcal{M}, \vec{i}_M, \vec{j}_M, \vec{k}_M)$ le repère lié au robot mobile.

Le modèle cinématique du robot est alors classiquement défini par les relations suivantes :

$$\begin{cases} \dot{x}_r = V \cos \theta_r \\ \dot{y}_r = V \sin \theta_r \\ \dot{\theta}_r = \omega \end{cases} \quad (3.6)$$

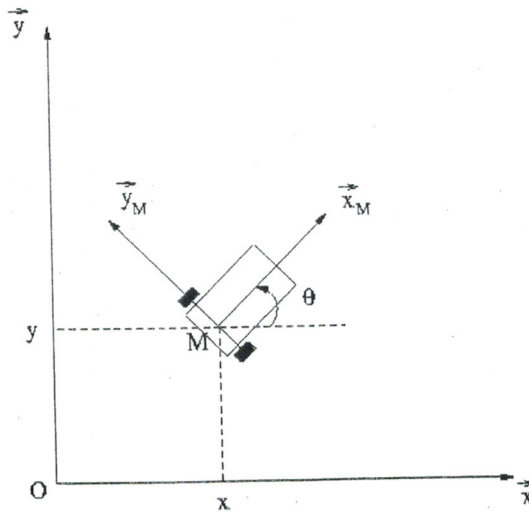


FIG. 3-2: Configuration d'un robot mobile

où V et ϖ désignent respectivement les vitesses linéaire et angulaire de la base mobile.

En mettant ces équations sous la forme matricielle, on pourra écrire :

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos \theta_r \\ \sin \theta_r \\ 0 \end{bmatrix} V + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \varpi \quad (3.7)$$

Le modèle cinématique 3.7 est, selon la terminologie utilisée en Automatique, un "système de commande" de type $\dot{\mathbf{x}}_r = f(\mathbf{x}_r, \mathbf{u}_r)$ avec $\mathbf{x}_r = [x_r \ y_r \ \theta_r]^T$ comme vecteur d'état et $\mathbf{u}_r = [V \ \varpi]$ comme commande du système.

3.2.2 Chemins et trajectoires [19]

Soit \mathcal{C} l'espace des configurations d'un robot mobile (c'est-à-dire le nombre minimum de paramètres qui permettent de localiser le système dans son environnement). Une **trajectoire** est une fonction de $[0, T]$ dans \mathcal{C} . Une trajectoire est dite *admissible* si elle est solution du système d'équations différentielles correspondant au modèle cinématique du robot mobile, incluant les contraintes sur les commandes, pour des conditions initiales et finales données. Un **chemin** est l'image dans \mathcal{C} d'une trajectoire. Un chemin *admissible* est donc l'image d'une trajectoire admissible.

3.2.3 Modélisation dans un repère de Frenet [19][30]

L'objectif de ce paragraphe est de généraliser les équations cinématiques précédentes lorsque le repère de référence est un repère de Frenet. Cette généralisation est celle qu'on va utiliser pour le

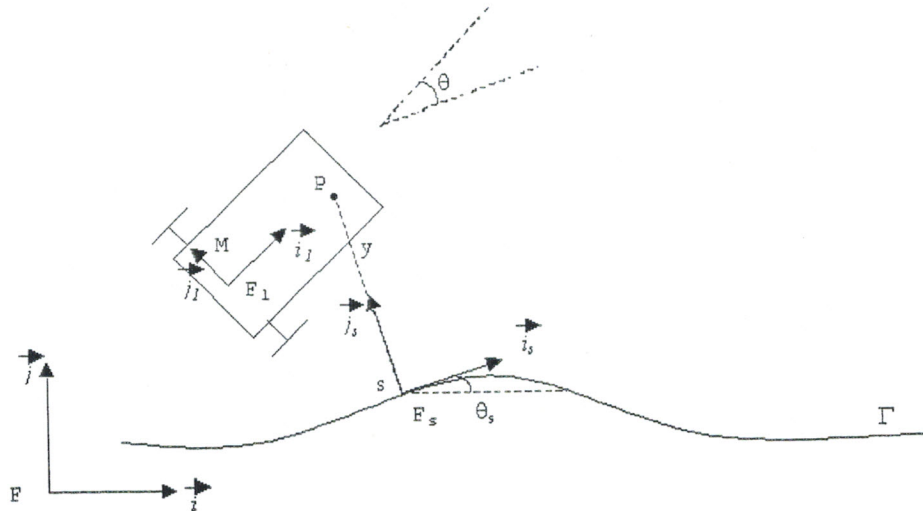


FIG. 3-3: Modélisation dans un repère de Frenet

suiivi de chemin. Considérons une courbe Γ du plan et les trois repères \mathcal{F} , \mathcal{F}_1 et \mathcal{F}_s représentés par la figure 3-3.

Le repère \mathcal{F} est le repère fixe de l'espace de travail, \mathcal{F}_1 est lié au robot avec, pour origine, le point M situé au milieu de l'axe des roues motrices et \mathcal{F}_s , indexé par l'abscisse curviligne s sur la courbe, est tel que le vecteur unitaire \vec{i}_s soit tangent à Γ . Considérons un point P lié au robot et soient (l_1, l_2) les coordonnées de P exprimées dans la base du repère \mathcal{F}_1 . On souhaite établir les équations de mouvement de P par rapport à la courbe Γ . Nous introduisons pour cela les trois variables s , y et θ définies comme suit :

- s représente l'abscisse curviligne du point S obtenu par projection orthogonale de P sur Γ ; afin que ce point existe et soit unique, le point P doit être suffisamment proche de la courbe. Plus précisément, si r désigne la borne inférieure (lorsque s parcourt la courbe Γ) du rayon de courbure $r(s)$ de Γ en s , P doit être situé à une distance de Γ inférieure à r . Nous supposons dans la suite que cette condition est satisfaite,
- y correspond à l'ordonnée de P dans le repère \mathcal{F}_s ; c'est aussi (au signe près) la distance de P à la courbe,
- $\theta = \theta_r - \theta_s$ représente l'orientation du robot par rapport au repère \mathcal{F}_s .

Déterminons maintenant \dot{s} , \dot{y} et $\dot{\theta}$.

En utilisant la définition de la courbure $c(s)$ de Γ en s , c'est-à-dire $c(s) = \partial\theta_s/\partial s$, on déduit direc-

tement de l'équation 3.6 que l'on a :

$$\dot{\theta} = \varpi - \dot{s}c(s) \quad (3.8)$$

Etant donné que l'on a $\overrightarrow{SP} = y\overrightarrow{j}_s$, on déduit tout d'abord, en utilisant l'égalité $d\overrightarrow{OS}/dt = \dot{s}\overrightarrow{i}_s$, que l'on a :

$$\begin{aligned} \frac{d\overrightarrow{OP}}{dt} &= \frac{d\overrightarrow{OS}}{dt} + \dot{y}\overrightarrow{j}_s - yc(s)\dot{s}\overrightarrow{i}_s \\ &= \dot{s}(1 - yc(s))\overrightarrow{i}_s + \dot{y}\overrightarrow{j}_s \end{aligned} \quad (3.9)$$

D'autre part, on a également, $\mathcal{MP} = l_1\overrightarrow{i}_1 + l_2\overrightarrow{j}_1$, d'où l'on déduit à partir de l'équation $d\overrightarrow{OM}/dt = V\overrightarrow{i}_1$, que l'on a :

$$\begin{aligned} \frac{d\overrightarrow{OP}}{dt} &= \frac{d\overrightarrow{OM}}{dt} + l_1\varpi\overrightarrow{j}_1 - l_2\varpi\overrightarrow{i}_1 \\ &= (V - l_2\varpi)\overrightarrow{i}_1 + l_1\varpi\overrightarrow{j}_1 \\ &= (V - l_2\varpi)(\cos\theta\overrightarrow{i}_s + \sin\theta\overrightarrow{j}_s) + l_1\varpi(-\sin\theta\overrightarrow{i}_s + \cos\theta\overrightarrow{j}_s) \\ &= [(V - l_2\varpi)\cos\theta - l_1\varpi\sin\theta]\overrightarrow{i}_s + [(V - l_2\varpi)\sin\theta + l_1\varpi\cos\theta]\overrightarrow{j}_s \end{aligned} \quad (3.10)$$

En projetant les équations 3.9 et 3.10 sur \overrightarrow{i}_s et \overrightarrow{j}_s et en utilisant également 3.8, on déduit finalement le système d'équations :

$$\begin{aligned} \dot{s} &= \frac{1}{1 - yc(s)} [(V - l_2\varpi)\cos\theta - l_1\varpi\sin\theta] \\ \dot{y} &= (V - l_2\varpi)\sin\theta + l_1\varpi\cos\theta \\ \dot{\theta} &= \varpi - \dot{s}c(s) \end{aligned} \quad (3.11)$$

Pour vérifier que ces équations représentent bien une généralisation du modèle cinématique utilisé, il suffit de prendre comme point P l'origine \mathcal{M} du repère \mathcal{F}_1 (c'est-à-dire $l_1 = l_2 = 0$) et de choisir l'axe (O, \overrightarrow{i}) du repère \mathcal{F} comme courbe Γ avec $s = x_r$. Dans ce cas, on a $c(s) = 0$ pour tout s et, en posant $y_r \equiv y$ et $\theta_r \equiv \theta$, on retrouve exactement le système 3.6.

Dans le cas général, si P est confondu avec \mathcal{M} , les équations 3.11 deviennent :

$$\begin{aligned} \dot{s} &= \frac{V}{1 - yc(s)} \cos \theta \\ \dot{y} &= V \sin \theta \\ \dot{\theta} &= \varpi - \dot{sc}(s) = \varpi - \frac{V \cos \theta}{\frac{1}{c(s)} - y} \end{aligned} \quad (3.12)$$

Remarque 3.1 *Ce système présente une singularité quand $y = \frac{1}{c(s)}$. Dans les situations pratiques, cette singularité est contournée en gardant le robot mobile proche au chemin Γ , et généralement, les chemins planifiés pour les robots non holonomes sont réguliers avec des courbures continues et bornées.*

Nous verrons, par la suite, qu'en utilisant les équations établies dans le repère de Frenet, on pourra obtenir une représentation simplifiée et même linéaire du système. Ceci sera visible après la transformation de ces équations en une forme canonique dite *chaînée*. Mais tout d'abord, étudions la commandabilité du système.

3.3 Etude de la commandabilité du système

Même dans des environnements statiques, commander un robot ou un système non holonome pour l'amener d'une configuration de départ à une configuration d'arrivée n'est pas une chose aisée. Ce problème a suscité un regain d'intérêt des automaticiens pour la robotique : les techniques de commande non linéaire trouvaient là une bonne application. De plus, il n'existe pas aujourd'hui d'algorithme général permettant de résoudre le problème pour n'importe quel système non holonome. Des solutions exactes sont connues seulement pour certaines classes de systèmes, mais pour les autres, on ne dispose que de méthodes approchées.

Nous allons à présent introduire la notion de commandabilité pour les systèmes non holonomes à l'aide de laquelle nous essayerons de prouver que le robot mobile que nous avons considéré est commandable.

3.3.1 Commandabilité [19]

Considérons une variété de dimension n et \mathcal{U} une classe de fonctions dépendant du temps et prenant leurs valeurs dans un compact \mathcal{K} de \mathbb{R}^m . Les systèmes non holonomes Σ sont généralement

de la forme :

$$\dot{\mathbf{x}} = f(\mathbf{x})\mathbf{u} + g(\mathbf{x}) \quad (3.13)$$

où \mathbf{u} est la commande du système. La i^e colonne de la matrice $f(\mathbf{x})$ est un champ de vecteur noté f_i , et $g(\mathbf{x})$ est appelé la *dérive*. Une *trajectoire admissible* est une solution du système différentiel vérifiant des conditions initiales et finales imposées et telle que \mathbf{u} appartient à \mathcal{U} .

Définition 3.1 *Le système Σ est localement commandable à partir de \mathbf{x} si l'ensemble des points accessibles à partir de \mathbf{x} le long d'une trajectoire admissible contient un voisinage de \mathbf{x} . Il est localement commandable en temps petit à partir de \mathbf{x} si l'ensemble des points accessibles à partir de \mathbf{x} le long d'une trajectoire admissible et avant un temps T contient un voisinage de \mathbf{x} pour tout T .*

Il convient de remarquer que la commandabilité locale en temps petit implique la commandabilité locale. La réciproque est fautive. Un système sera dit *commandable en temps petit* s'il est localement commandable en temps petit à partir de n'importe quel point.

Etablir les propriétés de commandabilité d'un système nécessite l'analyse de la structure de l'algèbre engendré par les champs de vecteur définissant le système et munie de l'opérateur **Crochet de Lie**. Rappelons que le crochet de Lie de deux champs de vecteur f et g est le champ de vecteur noté $[f, g]$ et égal à $\partial f.g - \partial g.f$. D'une façon explicite, la k^e coordonnée de $[f, g]$ est :

$$[f, g][k] = \sum_{i=1}^n \left(g[i] \frac{\partial}{\partial x_i} f[k] - f[i] \frac{\partial}{\partial x_i} g[k] \right)$$

Le théorème suivant procure un résultat puissant qui nous permet de tester la commandabilité en temps petit des systèmes symétriques (c'est-à-dire pour lesquels le compact \mathcal{K} est symétrique par rapport à l'origine) et sans dérive (c'est-à-dire pour lesquels on a $g(\mathbf{x}) = 0$).

Théorème 3.1 *Un système symétrique sans dérive est commandable en temps petit à partir de \mathbf{x} si et seulement si la dimension de l'espace vectoriel engendré par la famille des champs de vecteur f_i et de tous leurs crochets de Lie itérés est n au point \mathbf{x} .*

Ce théorème est plus connu sous le nom de **condition du rang de l'algèbre de Lie**. Une méthode pour tester cette condition consiste à construire une base de l'espace tangent au point considéré à partir d'une base de l'algèbre de Lie libre engendrée par les champs de vecteurs définissant le système.

Appliquons cette méthode à notre robot mobile. Le modèle cinématique donné par l'équation 3.7 est de la forme suivante :

$$\dot{\mathbf{x}} = f_1(\mathbf{x})V + f_2(\mathbf{x})\varpi \quad (3.14)$$

Les coordonnées du champ de vecteur $[f_1, f_2]$, qui est le crochet de Lie des champs de vecteur f_1 et f_2 sont :

$$[f_1, f_2](\mathbf{x}) = \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{pmatrix} \quad (3.15)$$

On vérifie aisément que la famille $\{f_1(\mathbf{x}), f_2(\mathbf{x}), [f_1, f_2](\mathbf{x})\}$ est une base de l'espace tangent en \mathbf{x} , et ceci pour tout \mathbf{x} . Le système est donc commandable en temps petit.

3.4 Synthèse de la loi de commande

3.4.1 Passage au modèle en forme chaînée [31]

L'existence de formes canoniques pour les modèles cinématiques des robots non-holonomes est essentielle au développement systématique de stratégies de contrôle pour ces systèmes. La structure canonique la plus utile à cette issue est la **forme chaînée**.

L'utilisation de la forme chaînée est très importante en robotique mobile. En plus de la simplification qu'elle apporte dans la synthèse de lois de commande, elle permet une extension directe des contrôleurs aux véhicules plus complexes (par exemple, les robots avec remorques). Toutes les lois de commande synthétisées pour des systèmes chaînés peuvent être appliquées aux autres types de robots mobiles, même si les équations régissant ces derniers ne peuvent pas être mises sous la forme chaînée.

Un système chaîné est de la forme suivante :

$$\begin{aligned}\dot{x}_1 &= u \\ \dot{x}_2 &= f_2(x_1)u \\ \dot{x}_3 &= f_3(x_1, x_2)u \\ &\vdots \\ \dot{x}_p &= f_p(x_1, \dots, x_p)u\end{aligned}$$

avec $x_i \in \mathbb{R}^{m_i}$ et $\sum_i m_i = n$.

En raison de cette forme particulière, il est possible d'utiliser une commande qui affecte le i^e sous-ensemble de coordonnées en laissant les coordonnées précédentes inchangées.

Pour les systèmes à deux entrées, il existe une représentation spécifique, dite *forme chaînée simple* $(2, n)$, elle est donnée par :

$$\begin{aligned}\dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \\ &\vdots \\ \dot{x}_n &= x_{n-1} u_1\end{aligned}\tag{3.16}$$

En pratique, ce dernier cas couvre beaucoup de modèles cinématiques de robots mobiles.

Bien qu'il est non-linéaire, le système chaîné 3.16 possède une structure fondamentale forte. Ceci apparaît clairement quand u_1 est choisie comme fonction du temps, de telle sorte qu'elle ne soit plus considérée comme une variable de commande. Dans ce cas, le système 3.16 devient un système linéaire à une entrée.

La conversion d'un système à deux entrée à la forme chaînée 3.16 se fait par :

1. un changement de coordonnées,
2. une transformation inversible de l'entrée.

Nous allons déterminer un changement de coordonnées et de variables de commande

$$(s, y, \theta, V, \varpi) \mapsto (z_1, z_2, z_3, m_1, m_2)$$

permettant de transformer 3.12 en un système chaîné de dimension 3 :

$$\begin{cases} \dot{z}_1 = m_1 \\ \dot{z}_2 = m_1 z_3 \\ \dot{z}_3 = m_2 \end{cases} \quad (3.17)$$

Posons tout d'abord :

$$z_1 = s, \quad m_1 = \dot{s} = \frac{V}{1 - yc(s)} \cos \theta$$

de sorte que l'on obtienne déjà $\dot{z}_1 = m_1$. Ceci implique que :

$$\begin{aligned} \dot{y} &= V \sin \theta \\ &= \frac{V}{1 - yc(s)} \cos \theta ((1 - yc(s)) \tan \theta) \\ &= m_1 (1 - yc(s)) \tan \theta \end{aligned}$$

Posons ensuite $z_2 = y$ et $z_3 = (1 - yc(s)) \tan \theta$ de sorte que l'équation ci-dessus devienne $\dot{z}_2 = m_1 z_3$.

Enfin, définissons :

$$m_2 = \dot{z}_3 = \left(-\dot{yc}(s) - y \frac{\partial c}{\partial s} \dot{s} \right) \tan \theta + (1 - yc(s)) (1 + \tan^2 \theta) \dot{\theta} \quad (3.18)$$

Les variables z_i et m_i vérifient bien 3.17.

Remarque 3.2 *On voit que dans le choix des variables du système chaîné, z_3 n'est pas définie pour $\theta = \frac{\pi}{2}[\pi]$. Cette condition est généralement évitée en choisissant un chemin approprié à suivre.*

3.4.2 Commande pour le suivi de chemin [33]

La méthode qu'on va utiliser exploite la structure interne des systèmes chaînés pour synthétiser le contrôleur. Ceci se fait en deux phases : dans la première phase, on suppose que l'une des entrées de commande est donnée et qu'elle satisfait quelques conditions générales. L'autre entrée de commande est donc synthétisée pour la stabilisation d'un *sous vecteur* de dimension $(n-1)$ du vecteur d'état du système. Dans la seconde phase, la synthèse est complétée en spécifiant la première entrée de commande de manière à garantir la convergence des variables restantes tout en stabilisant globalement

la boucle fermée.

Dans la suite, on ne détaillera que la première étape de cette méthode. La loi de commande obtenue à cette phase résout le problème de suivi de trajectoire à condition que la vitesse d'avancement du véhicule soit non nulle. Pour plus de précision sur le reste de la méthode, le lecteur pourra se reporter à [19].

En étape préliminaire, on réordonne les variables du système chaîné en mettant :

$$\mathbf{X} = (X_1, X_2, \dots, X_{n-1}, X_n) = (x_1, x_n, \dots, x_3, x_2)$$

En conséquence directe, le système chaîné 3.16 devient :

$$\begin{aligned} \dot{X}_1 &= u_1 \\ \dot{X}_2 &= X_3 u_1 \\ \dot{X}_3 &= X_4 u_1 \\ &\vdots \\ \dot{X}_{n-1} &= X_n u_1 \\ \dot{X}_n &= u_2 \end{aligned} \tag{3.19}$$

ou bien :

$$\dot{\mathbf{X}} = h_1(\mathbf{X})u_1 + h_2u_2, \quad h_1(\mathbf{X}) = \begin{bmatrix} 1 \\ X_3 \\ X_4 \\ \vdots \\ X_n \\ 0 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{3.20}$$

Pour le robot tricycle, cette réadaptation est effectuée seulement en permutant la deuxième et la troisième coordonnée, chose que nous avons faite directement dans le passage à la forme chaînée.

Posons $\mathbf{X} = (X_1, \mathbf{X}_2)$, avec $\mathbf{X}_2 = (X_2, X_3, \dots, X_n)$. Dans la suite, nous poursuivons uniquement la stabilisation de \mathbf{X}_2 à 0. La stabilisation de tout le vecteur \mathbf{X} peut être effectuée en une seconde étape si on désire stabiliser le véhicule en une configuration fixe.

Quand u_1 est choisie comme une fonction du temps, le système chaîné 3.20 peut être ré-écrit de la manière suivante :

$$\begin{aligned} \dot{\tilde{X}}_1 &= 0 \\ \dot{\mathbf{X}}_2 &= \begin{bmatrix} 0 & u_1(t) & 0 & \cdots & \cdots & 0 \\ 0 & 0 & u_1(t) & 0 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & u_1(t) & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & u_1(t) \\ 0 & 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix} \mathbf{X}_2 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \end{aligned} \quad (3.21)$$

avec :

$$\tilde{X}_1 = X_1 - \int_0^t u_1(\tau) d\tau.$$

La première équation dans 3.21 montre clairement que, quand l'entrée u_1 est fixée à priori, le système n'est plus commandable. Cependant, la structure des équations différentielles définissant \mathbf{X}_2 rappelle la forme canonique de commandabilité des systèmes linéaires. En particulier, quand u_1 est constante et non nulle, le système 3.21 devient un système à temps invariant et sa deuxième partie est clairement commandable. En fait, la commandabilité est vérifiée lorsque $u_1(t)$ est une fonction continue par morceaux, bornée, et strictement positive (ou négative). Sous ces hypothèses, x_1 varie de façon monotone avec le temps, et la différentiation par rapport au temps peut être remplacée par la différentiation par rapport à X_1 , de sorte que :

$$\frac{d}{dt} = \frac{d}{dX_1} \dot{X}_1 = \frac{d}{dX_1} u_1$$

et ainsi :

$$\text{sign}(u_1) \frac{d}{dX_1} = \frac{1}{|u_1|} \cdot \frac{d}{dt}.$$

Donc, la deuxième partie du système peut être ré-écrite comme suit :

$$\begin{aligned}
 X_2^{[1]} &= \text{sign}(u_1)X_3 \\
 X_3^{[1]} &= \text{sign}(u_1)X_4 \\
 &\vdots \\
 X_{n-1}^{[1]} &= \text{sign}(u_1)X_n \\
 X_n^{[1]} &= \text{sign}(u_1)u_2'
 \end{aligned} \tag{3.22}$$

avec les définitions suivantes :

$$X_i^{[j]} = \text{sign}(u_1) \frac{d^j X_i}{dX_1^j} \quad \text{et} \quad u_2' = \frac{u_2}{u_1}$$

L'équation 3.22 représente un système linéaire à temps invariant. Une représentation entrée-sortie équivalente pour ce système est donnée par :

$$X_2^{[n-1]} = \text{sign}(u_1)^{n-1} u_2'$$

Un tel système est commandable et peut être stabilisé exponentiellement par une loi de commande de la forme :

$$u_2'(\mathbf{X}_2) = -\text{sign}(u_1)^{n-1} \sum_{i=1}^{n-1} K_i X_2^{[i-1]} \tag{3.23}$$

où les gains $K_i > 0$ sont choisis de manière à satisfaire le critère de stabilité de Hurwitz. Par conséquent, la loi de commande :

$$u_2(\mathbf{X}_2, t) = u_1(t) u_2'(\mathbf{X}_2) \tag{3.24}$$

stabilise asymptotiquement et globalement l'origine $\mathbf{X}_2 = 0$.

Dans le problème de suivi de chemin, le sous-vecteur d'état \mathbf{X}_2 exprime généralement les déviations latérale et angulaire du véhicule par rapport au chemin. Donc en stabilisant asymptotiquement $\mathbf{X}_2 = 0$, le robot finira par rejoindre le chemin planifié (par rapport auquel sont calculées ces déviations).

3.4.3 Application au robot mobile

Dans ce cas, le suivi de chemin nécessite l'annulation des variables z_2 et z_3 liées aux déviations latérale et angulaire indépendamment de la variable z_1 (où X_1 dans le cas général).

Le système chaîné représentant notre robot non holonome est donné par 3.17. Comme on l'a vu précédemment, la linéarité de ce système apparaît lorsqu'on le dérive par rapport à z_1 :

$$\begin{cases} z_1' = 1 \\ z_2' = z_3 \\ z_3' = m_3 \end{cases} \quad \text{avec } m_3 = \frac{m_2}{m_1}. \quad (3.25)$$

Grâce à la linéarité de ce système chaîné (3.25), une loi de commande simple et naturelle est donnée par :

$$m_3 = -K_d z_3 - K_p z_2 \quad \text{avec } (K_p, K_d) \in \mathfrak{R}^{+2} \quad (3.26)$$

A partir des équations 3.25 et 3.26, on obtient l'équation caractéristique du système :

$$z_2'' + K_d z_2' + K_p z_2 = 0 \quad (3.27)$$

En conséquent, tant que la vitesse linéaire du véhicule V est non nulle, les variables z_2 et z_3 convergent toutes les deux vers zéro. Puisque l'on a $z_2 = y$ et $z_3 = (1 - yc(s)) \tan \theta$, la loi de commande donnée par 3.26 assure une convergence à zéro des deux variables y et θ . Il s'en suit que le robot rejoindra le chemin planifié tant que sa vitesse d'avancement est non nulle.

Le choix des gains K_p et K_d permet de déterminer les performances du contrôleur proportionnel-dérivée et de définir une distance d'établissement pour la régulation de la déviation latérale y .

En prenant en compte le fait que la vitesse d'avancement V peut être négative, l'expression de m_2 peut être obtenue à partir de 3.26 comme suit :

$$m_2 = -|m_1| K_d z_3 - m_1 K_p z_2 \quad (3.28)$$

Ainsi, ω est calculée à partir des équations 3.12 et 3.18. On obtient donc :

$$\omega(y, \theta) = \frac{V \cos^3 \theta}{(1 - yc(s))^2} \left(\frac{dc(s)}{ds} \tan \theta y + c(s) \frac{\cos \theta}{1 - yc(s)} + c(s)(1 - yc(s)) \tan^2 \theta - K_p y \right) - \frac{|V \cos^3 \theta|}{(1 - yc(s))^2} K_d \tan \theta \quad (3.29)$$

Rappelons que cette loi de commande est valable uniquement sous les conditions suivantes :

$$y \neq \frac{1}{c(s)}, \quad V \neq 0, \quad \theta \neq \frac{\pi}{2}[\pi] \quad (3.30)$$

Notons que, du fait que la saturation des actionneurs n'est pas considérée, l'équation 3.29 peut donner des valeurs non appropriées à la vitesse angulaire. Cependant, à cause du fait que V est dépendante du temps, elle peut être adaptée pour assurer que la vitesse angulaire ω reste dans les limites de saturation. En plus, dans la navigation en milieux d'intérieur, les vitesses prennent des valeurs modérées qui sont loin d'atteindre les limites de saturations des actionneurs.

3.4.4 Suivi d'un chemin droit

Comme nous l'avons déjà dit, le robot mobile doit passer le long de son chemin par des positions intermédiaires, qui sont définies par les images relais. Il est clair que la portion du chemin entre chaque deux positions intermédiaires peut être considérée, sans perte de généralité, comme une ligne droite. En utilisant cette propriété, on aura $c(s) = 0, \forall s$, et de ce fait, la loi de commande donné par 3.29 peut se simplifier en :

$$\omega(y, \theta) = -V \cos^3(\theta) K_p y - |V \cos^3(\theta)| K_d \tan \theta \quad (3.31)$$

Du fait que cette loi de commande est un cas spécial de 3.29 et que le système chaîné consiste en un double intégrateur (voir l'équation 3.25), la stabilité asymptotique du système est assurée. En plus, la condition $c(s) = 0$ implique que la singularité $y = \frac{1}{c(s)}$ ne se produit jamais durant le déplacement du robot.

Les résultats de simulation que nous avons obtenues ont montré que cette dernière loi de commande marche bien aussi dans le cas où la portion du chemin entre les positions intermédiaires n'est pas parfaitement droite, mais d'une courbure plus ou moins modeste.

Remarque 3.3 *Le contrôle de l'abscisse curviligne du robot n'est pas possible par les lois de commandes décrites précédemment. Cependant, après que le robot rejoigne sa trajectoire, un arrêt peut être effectué dans l'une des positions définies par les images relais.*

3.5 Conclusion

Dans ce chapitre, nous avons traité les aspects de modélisation et de commande du robot mobile. Nous avons présenté le modèle le plus utilisé dans le contexte de suivi de chemin, qui est le modèle établi dans le repère de Frenet. Après cela, nous avons étudié la commandabilité du système considéré en utilisant la condition du rang de l'algèbre de Lie, il s'est avéré que le système est parfaitement commandable en temps petit. Dans la synthèse de la loi de commande, nous avons montré comment le passage à la forme chaînée permet d'apporter de grandes simplifications dans le calcul du contrôleur à cause de la forme relativement simple de ces systèmes et de sa ressemblance particulière avec les systèmes linéaires.

Nous avons ensuite montré la technique utilisée pour la synthèse d'une loi de commande pour le suivi de chemin et nous l'avons appliquée au robot non holonome considéré. Les résultats obtenus montrent clairement que les déviations latérale et angulaire du système s'annulent tant que la vitesse d'avancement du véhicule est non nulle. Une expression plus simplifiée pour la vitesse angulaire du robot mobile a été obtenue en considérant un chemin droit.

L'efficacité de loi de commande utilisée a été prouvée par les résultats de simulations que nous avons obtenus. Les aspects pratiques de ce travail ainsi que ces résultats sont présentés dans le chapitre suivant.

Chapitre 4

Aspects pratiques et résultats de simulation

Dans ce chapitre, nous allons discuter d'une manière grossière les aspects pratiques du suivi de trajectoire, pour le robot non holonome, par asservissement visuel. Ensuite, nous nous intéresserons d'une façon plus détaillée à la présentation des différents résultats que nous avons obtenus en simulation. Des commentaires seront ajoutés pour mieux éclaircir le contenu.

4.1 Aspects pratiques

Dans ce document, nous avons étudié un système de localisation et de commande pour un robot mobile non holonome en utilisant un capteur de vision. Les paramètres permettant de localiser le système dans son environnement peuvent être parfaitement estimés à partir des mesures (probablement bruitées) dans l'image. Nous avons également vu que la localisation n'est plus effectuée dans un espace continu, mais plutôt dans l'espace discret des positions pour lesquelles des images ont été apprises.

Du point de vue commande, il s'agit de stabiliser asymptotiquement vers zéro les états du système exprimant ses déviations latérale et angulaire par rapport au chemin suivi. Ces états sont régulièrement estimés à partir des informations visuelles par le calcul de l'homographie entre chaque couple d'images relais. En contrepartie, l'état exprimant l'abscisse curviligne du véhicule par rapport au chemin ne peut pas être déterminé. Par ailleurs, un arrêt du système peut être envisageable en l'une des positions définies par les images clés.

En combinant les systèmes de localisation et de commande, on obtient une boucle fermée com-

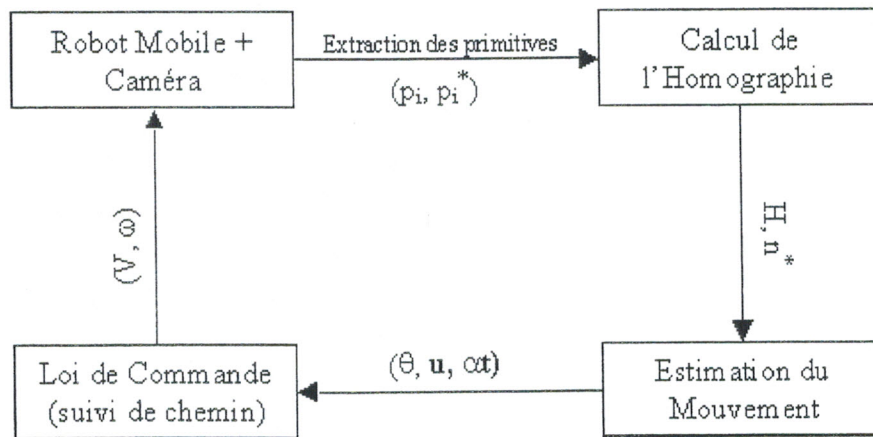


FIG. 4-1: La boucle fermée du système de navigation par asservissement visuel

plète pour ce **système de navigation par asservissement visuel**, qui est illustrée dans la figure 4-1. Dans la simulation que nous avons effectuée, seule l'extraction des primitives visuelles n'est pas prise en compte. On supposera donc que les informations visuelles nécessaires à l'accomplissement de cette tâche (coordonnées en pixels de points appartenant à la Landmark) sont disponibles durant toute la phase de l'asservissement.

En situation pratique, le robot mobile doit être équipé d'un ordinateur numérique, sur lequel s'exécutent les programmes permettant l'extraction des primitives visuelles à partir des images fournies par la caméra. Comme nous l'avons déjà dit auparavant, les primitives utilisées dans ce travail sont de type point. Du fait que les objets d'intérêt sont planaires, ces primitives sont largement suffisantes pour identifier une Landmark dans l'image. D'autre part, les algorithmes d'extraction de ces primitives visuelles (de type point) sont plus rapides que les algorithmes dédiés à d'autres types de primitives. En effet, actuellement, on n'utilise plus le balayage complet de l'image pour la détection des primitives visuelles mais on utilise plutôt le suivi (*tracking*). Du coup, on obtient un système purement temps réel. A titre indicatif, les trackeurs les plus appliqués sont à base des filtres de Kalman et très récemment, on a commencé à utiliser les filtres à particules.

En plus, nous avons déjà vu dans la section 2.5.3 que l'homographie reliant les projections de points dans plusieurs images peut être calculée par la résolution d'un simple système linéaire d'équations, opération qui se fait généralement en un court laps de temps. Donc, en pratique, il est tout à fait possible que le suivi de chemin pour le robot mobile non holonome, par cette technique d'asservissement visuel, soit effectué en temps réel.

4.2 Résultats et commentaires

Nous avons testé l'approche de commande de robot non holonome, proposée dans cette étude, dans un environnement de simulation. Comme nous l'avons mentionné précédemment, les tâches d'extraction de primitives visuelles ne sont pas prises en compte. La simulation établie consiste alors, d'une manière générale, aux tâches suivantes :

- calcul de l'homographie à partir des coordonnées des points de la Landmark,
- estimation du déplacement partiel entre les deux prises de vue considérées,
- localisation du système par rapport à son chemin et stabilisation asymptotique de l'état du système à des déviations latérale et angulaire nulles.

Rappelons que la tâche robotique fondamentale considérée ici est de rejoindre une image désirée dans l'espace image. La caméra utilisée est embarquée sur le robot mobile. Un angle d'inclinaison (*tilt angle*) de 0.2 rad est introduit entre l'axe optique de la caméra et l'axe vertical du repère \mathcal{F}_M associé au robot mobile.

La caméra visualise six points (constituant la Landmark). Ces points appartiennent à un plan qui est parallèle au plan d'évolution du robot mobile. La loi de commande présentée dans le chapitre précédent est utilisée dans la simulation pour contrôler les mouvements du robot.

Dans cette simulation, nous avons considéré le mouvement élémentaire du robot entre deux images relais. Pour cela, on choisit l'origine du repère de la scène \mathcal{F} comme position désiré pour le système robot mobile + caméra. On fixe la situation initiale du robot à une position donnée par le vecteur de translation \mathbf{t} et les angles de rotation α , β et γ , par rapport aux axes des X , Y et Z de \mathcal{F} respectivement.

Cependant, il est clair que seule la rotation par rapport à l'axe des Z est possible pour ce robot mobile et qu'une translation de ce véhicule dans le sens de cet même axe n'est pas réalisable. Ceci nous donne deux degrés de liberté contrôlables en translation et un seul en rotation. De ce fait, nous avons choisi les paramètres exprimant la situation du repère initial par rapport au repère final de la manière suivante :

$$\mathbf{t} = [-1 \quad -0.5 \quad 0]^T; \quad \alpha = 0, \beta = 0, \gamma = -\pi/5$$

Dans cette simulation, nous avons utilisé un certain nombre de fonctions d'une librairie MATLAB, utilisée pour la simulation des travaux d'asservissements visuels. Elle contient des fonctions permet-

tant de simuler les caméras, de tracer les formes géométriques qui représentent les primitives visuelles utilisées et de calculer les poses à partir de données visuelles.

En utilisant les fonctions de cette librairie, les positions des caméras, et donc du robot, peuvent être représentées dans les figures par des repères.

On commence par fixer les paramètres intrinsèques de la caméra comme suit :

$$\alpha_u = 657; \quad \alpha_v = 657; \quad \alpha_{uv} = 0; \quad u_0 = 302, \quad v_0 = 242$$

On obtient donc la matrice des paramètres intrinsèques suivante :

$$\mathbf{K} = \begin{bmatrix} 657 & 0 & 302 \\ 0 & 657 & 242 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

On choisit ensuite 6 points *coplanaires*, représentant la Landmark, dans le repère de la scène. Les coordonnées de ces points, en mètre et en pixel, sont calculés ensuite dans les positions initiale et désirée, en utilisant les matrices de transformations homogènes nécessaires et la matrice des paramètres intrinsèques donnée par 4.1. La figure 4-2 montre les positions initiales et désirées de ces points (en pixel) sur le plan image.

Dans la figure 4-3, on représente les repères associés aux caméras dans leurs positions initiales et désirées. Comme on peut le remarquer sur cette figure, la caméra est initialement orientée d'un angle de $-\pi/5$ par rapport à la position souhaitée. Ceci nous permettra de tester l'efficacité de la loi de commande dans le contrôle de l'orientation.

En utilisant l'algorithme donné à la section 2.5.3, nous avons réalisé une fonction pour l'estimation de la matrice d'homographie entre les deux positions de la caméra, à partir des coordonnées des points dans les positions initiale et désirée. Pour la résolution numérique du système d'équations mises en jeu, nous avons utilisé la méthode des moindres carrés.

Ensuite, par la méthode de reconstruction des paramètres tridimensionnels à partir de la matrice d'homographie, décrite dans le section 2.6.1, une autre fonction d'estimation des paramètres tridimensionnels du mouvement de la caméra et de la structure de la scène a été réalisée. Rappelons que

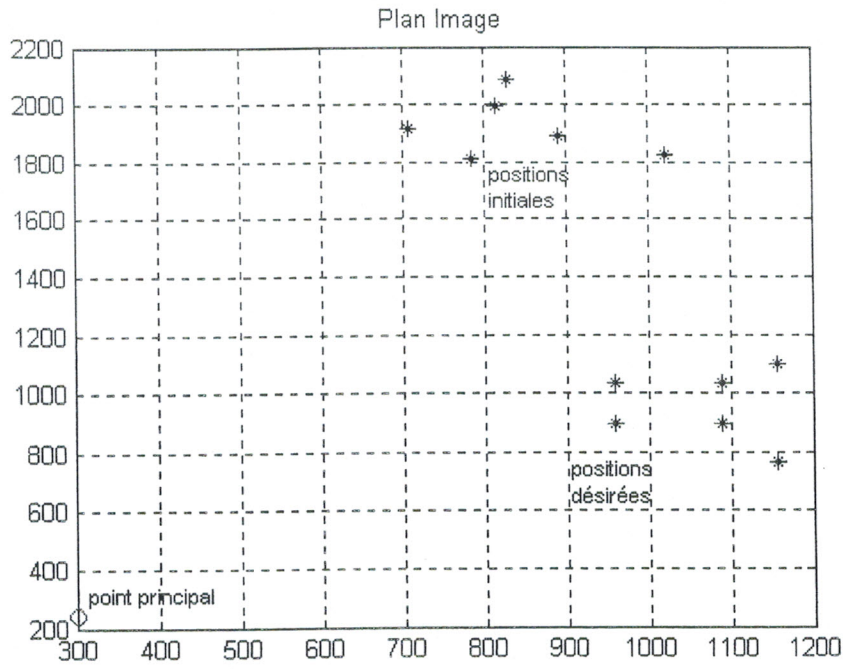


FIG. 4-2: Points dans l'image en positions initiale et désirée

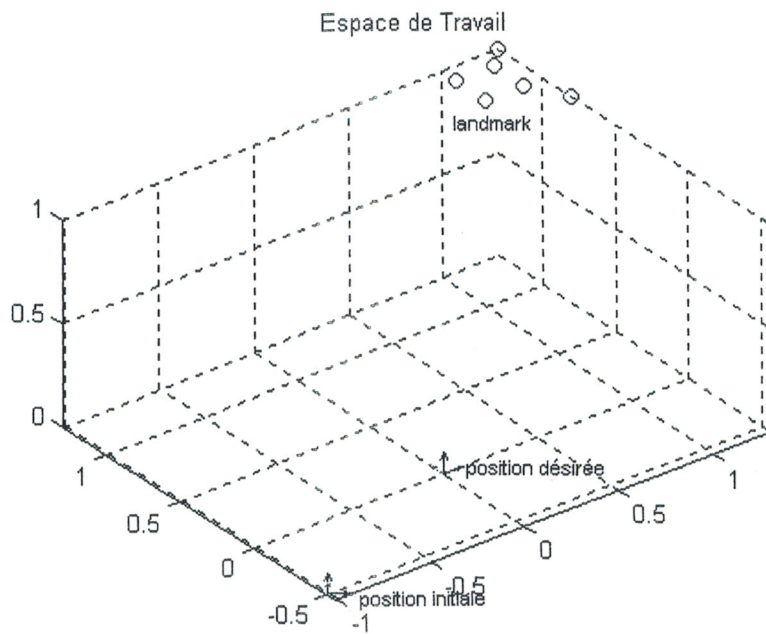


FIG. 4-3: Positions initiale et désirée de la caméra dans l'espace de travail

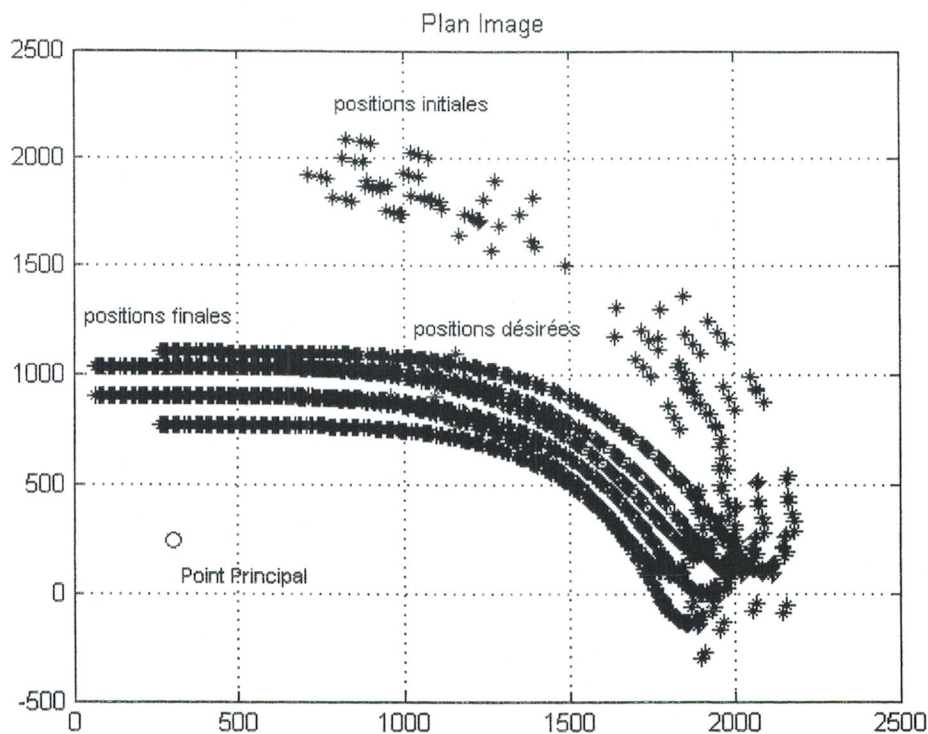


FIG. 4-4: Trajectoires des points sur le plan image

les paramètres **3D** qu'on peut obtenir à partir de la matrice d'homographie sont :

- les composantes du vecteur \mathbf{n}^* normal au plan π^* ,
- la matrice de rotation \mathbf{R} ,
- le vecteur de translation \mathbf{t} qu'on obtient à un facteur d'échelle près.

Après cela, nous avons introduit le modèle du système donné par 3.6 et 3.12, ainsi que la commande donnée par 3.31 dans des fichiers séparés. Les paramètres du contrôleur et la vitesse d'avancement du véhicule ont été choisis comme suit :

$$K_p = 35, \quad K_d = 15, \quad V = 0.02 \quad (4.2)$$

Les résultats obtenus sont données par les figures 4-4 et 4-5.

Sur la figure 4-4, on remarque que dans le plan image, les trajectoires des points passent approximativement par les positions désirées. On peut aussi voir clairement sur la figure 4-5 la trajectoire du système (robot+caméra) dans l'espace de travail. Chaque position du système est représentée par un repère dans l'espace **3D**.

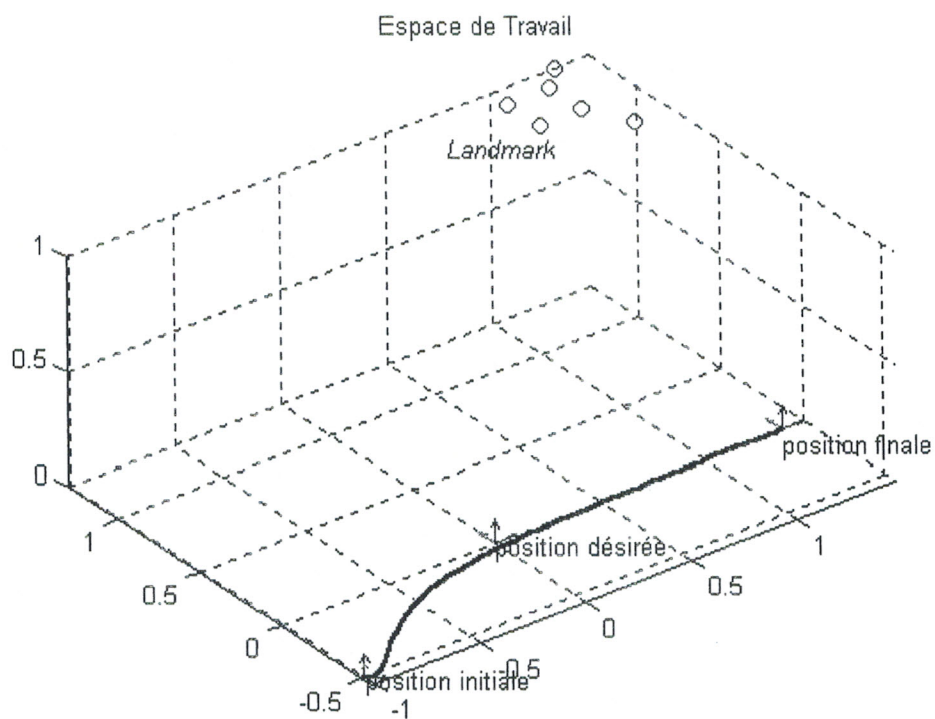


FIG. 4-5: Trajectoire de la caméra dans l'espace de travail

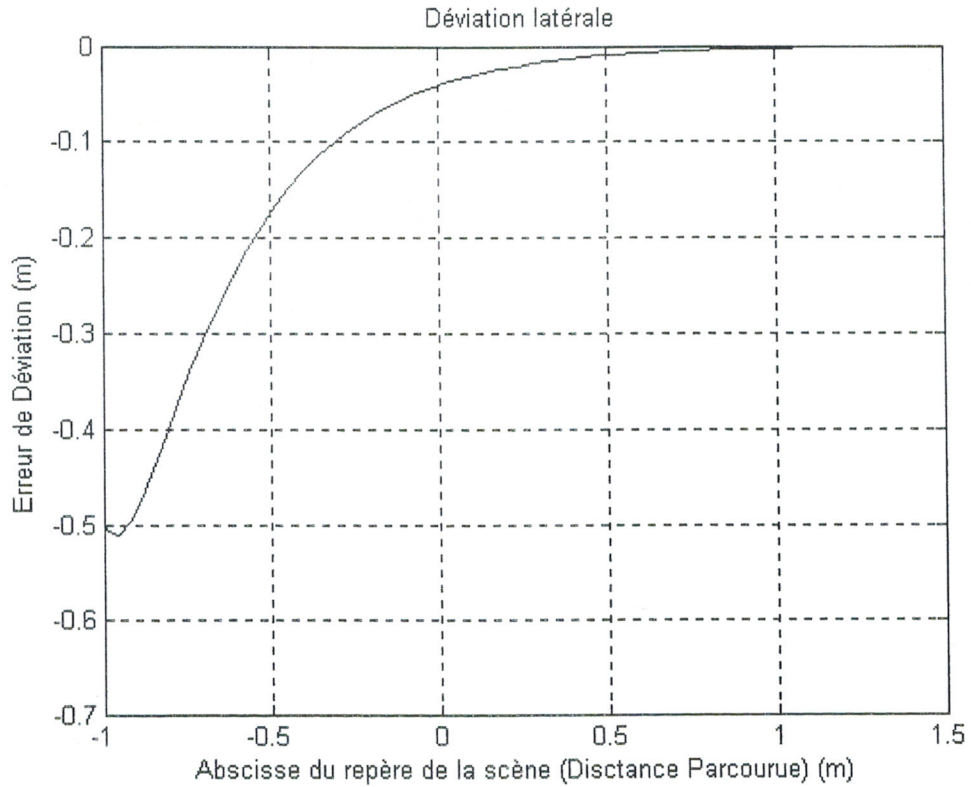


FIG. 4-6: Déviation latérale de la caméra

Rappelons que la loi de commande utilisée n'agit pas sur la coordonnée curviligne s du véhicule sur le chemin. Elle stabilise seulement les déviations latérale y et angulaire θ (voir la section 3.4.2).

L'évolution de ces deux dernières grandeurs est donnée par les figures 4-6 et 4-7.

4.3 Conclusion

Dans ce chapitre, nous avons exposé les résultats de simulation que nous avons obtenus. L'utilisation de la Toolbox Matlab dédiée à l'asservissement visuel nous a permis de voir, en plus du mouvement du système dans l'espace tridimensionnel, les trajectoires que prennent les primitives visuelles durant l'asservissement.

D'une manière générale, cette tâche de simulation consistait au calcul de l'homographie à partir des informations visuelles que nous avons supposées disponibles au cours de l'asservissement, sup-

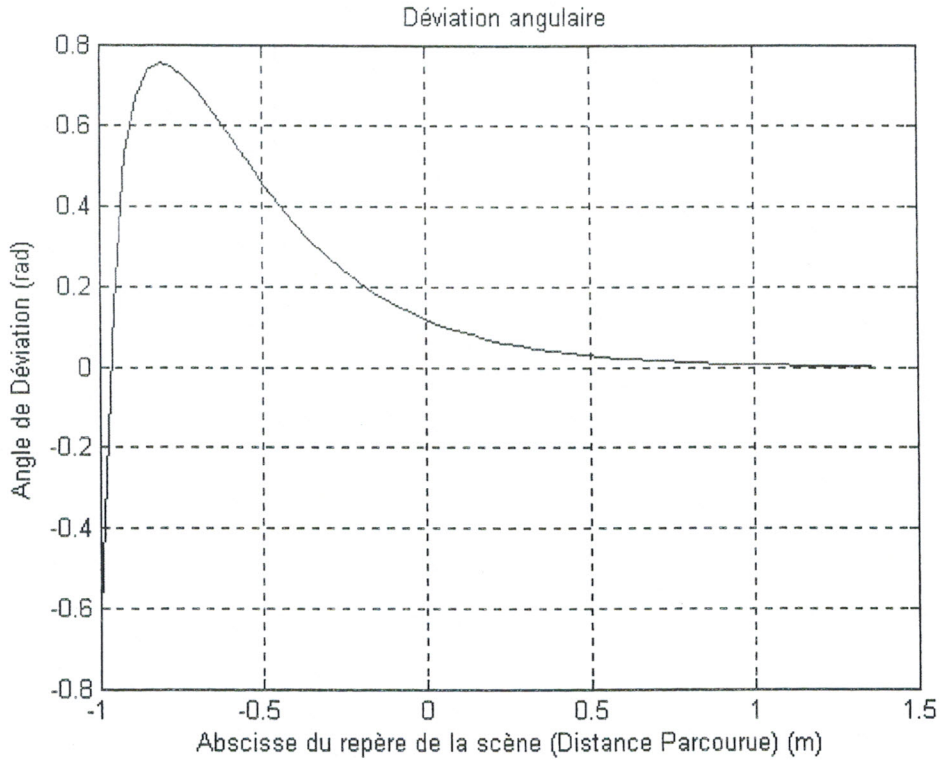


FIG. 4-7: Déviation angulaire de la caméra

position désormais possible en raison de la rapidité que procurent les algorithmes de tracking dans l'extraction des primitives visuelles. Il était nécessaire aussi de remonter aux informations **3D** pour trouver la relation entre les images relais. Finalement, la loi de commande permettait de rejoindre, puis de suivre, la trajectoire initialement planifiée dans la phase de l'apprentissage.

Le choix des gains dans la loi de commande permet d'ajuster la distance d'établissement, c'est-à-dire la distance au bout de laquelle le système rejoint le chemin qu'il doit suivre. En augmentant les gains de la commande, le système se rapproche plus rapidement de sa position souhaitée, mais des mouvements de rotations "brusques" apparaîtront dans les trajectoires de primitives visuelles sur le plan image.

Le suivi du chemin entier s'obtient par la succession des mouvements élémentaires entre chaque couple d'images relais. Nous n'avons pas pu présenter le résultat prouvant cette proposition à cause de la complexité des tracés qui en résultent.

Conclusion Générale

Les travaux présentés dans ce mémoire ont porté sur la localisation et la commande d'un robot mobile non holonome par les techniques d'asservissement visuel afin de lui permettre de suivre une trajectoire préalablement planifiée. Comme nous l'avons mentionné précédemment, l'application des techniques d'asservissement visuel aux robots mobiles est totalement différente de son application aux robots manipulateurs. Nous avons centré nos travaux sur l'approche d'asservissement visuel $2D1/2$, dont nous avons présenté le principe et les différents calculs qui s'y attachent dans le chapitre 2.

Une description générale de ce travail peut se résumer dans les points suivants :

- En utilisant les méthodes de calculs d'homographie relatives à la technique $2D1/2$, nous avons pu obtenir une localisation du robot mobile dans l'espace discret des diverses positions que peut prendre le robot en suivant sa trajectoire. Rappelons qu'à chacune de ces positions, on a fait correspondre l'image visualisée par la caméra à cet emplacement. Cette étape de localisation est cruciale au processus de navigation. Elle nous permet de situer le système par rapport au chemin considéré.
- L'état du système est exprimé ensuite par sa coordonnée curviligne le long du chemin à suivre et par ses déviations latérale et angulaire par rapport à ce même chemin. Ce sont ces deux dernières grandeurs qu'on tente de stabiliser asymptotiquement pour que le robot rejoigne la bonne trajectoire. La position du système sur son chemin est, quant à elle, contrôlée dans l'espace image. Cette étape, représentant la partie commande du processus de navigation, fait l'objet du chapitre 3.

Nous avons toutefois souligné le fait qu'il peut exister des configurations entraînant des dégénérescences dans les calculs d'homographie et que, pour les éviter, il faut que le chemin choisi soit régulier (avec des courbures continues et bornées) et que le robot y reste suffisamment proche.

En ce qui concerne les répercussions pratiques de ce travail, l'utilisation des algorithmes de tracking pour la détection de primitives visuelles, permettra au système de tourner en temps réel. Néanmoins, la rapidité de ce processus sera nettement altérée en cas d'utilisation de primitives plus complexes.

Une suite naturelle à ce travail serait de considérer une caméra dotée de plus de degrés de liberté au lieu de la fixer rigidement sur le robot mobile. Par cette supposition, l'intégration de tâches secondaires dans ce processus de navigation, telles que la détection d'obstacles sur le chemin et leurs évitements, semblerait être une perspective très intéressante.

Il s'avère cependant que les outils de traitements d'images et de vision par ordinateur, nécessaires au bon accomplissement de ces tâches, affecteront considérablement la rapidité du système dans la pratique. Mais d'un autre point de vue, les problèmes de contrôle posés dans une telle configuration seront plus délicats et pourront faire l'objet de plusieurs futurs travaux.

Bibliographie

- [1] B. Boufama & R. Mohr. *Epipole and fundamental matrix estimation using the virtual parallax property*. **IEEE** International Conference on Computer Vision, Pages 1030-1038, Cambridge, USA, 1995.
- [2] V. Cadenat. *Commande référencée multi-capteurs pour la navigation d'un robot mobile*. **PhD** thesis, Université Paul Sabatier, Toulouse, France, December 1999.
- [3] F. Chaumette. *La Relation Vision Commande : Théorie et Application à des Tâches Robotiques*. **PhD** thesis, Université de Rennes I, Rennes, France, July 1990.
- [4] F. Chaumette, P. Rives & B. Espiau. *The Task Function Approach Applied to Vision-Based Control*. In Proceedings of the **IEEE** International Conference on Robotics & Automation (**ICRA**), Pages 1392-1397, California, USA, April 1991.
- [5] F. Chaumette. *Potential Problems of Stability and Convergence in Image-Based and Position-Based Visual Servoing*. In A. Morse Springer Verlag G. Hager & D. Kriegman, editeur, *The Confluence of Vision and Control*. Lecture Notes in Control and Informations Systems, 1998.
- [6] P.I. Corke & M.C. Good. *Dynamic Effects in Visual Closed-Loop Systems*. **IEEE** Transactions on Robotics & Automation, Vol. 5, Pages 671-683, October 1996.
- [7] A. Crétual. *Asservissement visuel à partir d'informations de mouvement dans l'image*. **PhD** thesis, Université de Renne I, Renne, France, November 1998.
- [8] A. Crétual & F. Chaumette. *Image-Based Visual Servoing by Integration of Dynamic Measurements*. In Proceedings of the **IEEE** International Conference on Robotics & Automation (**ICRA**), Pages 1994-2001, Leuven, Belgium, May 1998.
- [9] A. Criminisi, I. Reid & A. Zisserman. *Duality, rigidity and planar parallax*. European Conference on Computer Vision, *ECCV'98*, Volume 2, Pages 846-861, Fribourg, Germany, June 1998.
- [10] D. Dementhon & L.S.Davis. *Model-Based Object Pose in 25 Lines of Code*. International Journal of Computer Vision, Pages 123-141, June 1995.
- [11] B. Espiau, F. Chaumette & P. Rives. *A New Approach to Visual Servoing in Robotics*. **IEEE** Transactions on Robotics & Automation, Vol. 8, no. 3, June 1992.

- [12] O. Faugeras & F. Lustman. *Motion and Structure from Motion in a Piecewise Planar Environment*. International Journal of Pattern Recognition and Artificial Intelligence. Vol. 2, Pages 485-508, 1988.
- [13] O. Faugeras. *Three Dimensional Computer Vision : A Geometric Viewpoint*. The MIT Press, Cambridge, Massachusetts, USA, 1993.
- [14] R. I. Hartley. *In defense of the eight-point algorithm*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 6, Pages 580-593, June 1997.
- [15] K. Hashimoto. *Visual servoing-real-time control of robot manipulators based on visual sensory feedback*. World Scientific Publishing Co. Pte. Ltd. NJ, USA, 1993.
- [16] T. S. Huang & O. Faugeras. *Some properties of the E matrix in two-view motion estimation*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 11, Pages 1310-1312, December 1989.
- [17] S. Hutchinson, G.D. Hager & P.I. Corke. *A Tutorial on Visual Servo Control*. *IEEE Transactions on Robotics & Automation*, Vol. 12, no. 5, October 1996.
- [18] J. Kosecka. *Visually Guided Navigation*. In Proceedings of the 4th International Symposium on Intelligent Robotic Systems (SIRS'96), Lisbon, Portugal, July 1996.
- [19] J.P. Laumond. *La Robotique Mobile, chapitre Planification de mouvement pour robots mobiles non holonomes*. Hermes Science Publications, Paris, France, 2001.
- [20] H. C. Longuet-Higgins. *A computer algorithm for reconstruction of a scene from two projections*. *Nature*, Pages 133-135, September 1981.
- [21] Q.T. Luong & O. Faugeras. *The fundamental matrix theory, algorithms and stability analysis*. *International Journal of Computer Vision*, January 1996.
- [22] Y. Ma, J. Kosecka & S. S. Sastry. *Vision guided Navigation for a Nonholonomic Mobile Robot*. *IEEE Transactions on Robotics & Automation*, 1999.
- [23] E. Malis, F. Chaumette & S. Boudet. *Positioning a Coarse-Calibrated Camera with Respect to an Unknown Object by 2D $\frac{1}{2}$ Visual Servoing*. In Proceedings of the IEEE International Conference on Robotics & Automation (ICRA), Pages 1352-1359, Leuven, Belgium, May 1998.
- [24] E. Malis. *Contribution à la modélisation et à la commande en asservissement visuel*. PhD thesis, Université de Rennes I, Rennes, France, November 1998.
- [25] E. Malis, F. Chaumette & S. Boudet. *2D 1/2 Visual Servoing*. *IEEE Transactions on Robotics & Automation*, Pages 238-250, April 1999.

- [26] E. Malis & F. Chaumette. *2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement*. International Journal of Computer Vision, Volume 1, Pages 79-97, June 2000.
- [27] Y. Mezouar. *Planification de trajectoires pour l'asservissement visuel*. PhD thesis, Université de Renne I, Renne, France, November 2001.
- [28] Y. Mezouar, A. Remazeilles, P. Gros & F. Chaumette. *Images Interpolation for Image-based Control Under Large Displacement*. In **IEEE** International Conference on Robotics & Automation (**ICRA**'2002), Volume 3, Pages 3787-3794, Washington DC, USA, May 2002.
- [29] R. Mohr & B. Triggs. *Projective Geometry for Image Analysis*. International Symposium of Photogrammetry and Remote Sensing, July 1996.
- [30] R. Pissard-Gibollet. *Conception et Commande par Asservissement Visuel d'un Robot Mobile*. PhD thesis, Ecole des Mines de Paris, Paris, France, December 1993.
- [31] C. Samson & K. Ait-Abderrahim. *Feedback Stabilization of a Nonholonomic Wheeled Mobile Robot*. In Proceedings of the **IEEE/RSJ** International Workshop on Intelligent Robots & Systems (**IROS**), Osaka, Japan, March 1991.
- [32] C. Samson, M.L. Borgne & B. Espiau. *Robot control : The task function approach*. Clarendon Press, Oxford, England, 1991.
- [33] C. Samson. *Control of Chained Systems : Application to Path Following and Time-Varying Stabilization of Mobile Robots*. **IEEE** Transactions on Automatic Control, 40(1) : 64-77, 1995.
- [34] R. Sharma & S. Hutchinson. *Optimizing Hand-Eye Configuration for Visual-Servo Systems*. In Proceedings of the **IEEE** International Conference on Robotics & Automation (**ICRA**), Pages 172-177, Nagoya, Japan, May 1995.
- [35] A. Shashua & N. Navab. *Relative affine structure : Theory and application to 3D reconstruction from perspective views*. **IEEE** International Conference on Vision and Pattern Recognition, Pages 483-489, 1994.
- [36] D. Tsakiris, P. Rives & C. Samson. *Extending Visual Servoing Techniques to Nonholonomic Mobile Robots*. The confluence of vision and control. Edited by G. Hager, D. Kriegman & A. Morse. Pages 106-117, Springer Verlag, 1998.
- [37] T. Viéville, C. Zeller & L. Robert. *Using collineations to compute motion and structure in an uncalibrated image sequence*. International Journal of Computer Vision, Volume 3, Pages 213-242, 1996.
- [38] L.E. Weiss & A.C Anderson. *Dynamic Sensor-Based Control of Robots with Visual Feedback*. **IEEE** Journal of Robotics & Automation, vol. 3, no. 5, Pages 404-417, 1987.

- [39] Z.Zang. *Determining the epipolar geometry and its uncertainty : a review*. INRIA Research Report n°1996, July 1996.
- [40] Z.Zang. *On the epipolar geometry between two images with lens distorsion*. International Conference on Pattern Recognition, Vol. 1, Pages 407-411, Vienna, Austria, August 1996.

Résumé

Dans ce mémoire, nous présentons un travail sur la navigation d'un robot mobile non holonome dans un environnement d'intérieur en utilisant seulement un capteur de vision embarqué sur le robot mobile. La tâche traitée est le suivi d'un chemin indiqué par des amers visuels. Le système de navigation utilise un ensemble d'images relais, acquises lors d'une phase d'apprentissage, pour permettre au robot d'atteindre une configuration finale exprimée par l'image visualisée sur la caméra à cet emplacement. L'état du système est exprimé dans un repère de Frenet pour pouvoir situer le robot par rapport à son chemin. La détermination de cet état pendant l'asservissement est effectuée par les calculs de l'homographie reliant chaque couple d'image relais, technique empruntée à l'approche d'asservissement visuel **2D1/2**. La loi de commande est ensuite synthétisée après passage à la forme chaînée afin de prendre en compte les contraintes non holonomes du système. Des simulations illustrent ces points et des résultats pour un robot mobile de type tricycle faisant le suivi d'amers artificiels sur le plafond de la scène sont présentés.

Mots Clés :

Robot mobile non holonome, géométrie épipolaire, relation homographique, systèmes chaînés.

Abstract

We present a framework for navigation of a nonholonomic mobile robot in and indoor environment, using only a monocular vision system embedded on the robot. We particularly focus on the path following problem. The path is indicated by some visual landmarks in the scene. The navigation system uses a set of relay images, obtained during an off-line learning step, to enable the robot to reach a goal configuration expressed by the image viewed on the camera at that position. The system state is determined in a Frenet frame to locate the robot with respect to the path. As practiced in the **2D1/2** visual servoing scheme, we compute the homography matrix between each two images to determine the system state during the servoing phase. The control law is designed after having transformed the system into a suitable chained form to overcome its nonholonomic constraints. Simulations illustrate these points and results for a cart-like robot which navigate using artificial landmarks on the ceiling are presented.

Keywords :

Nonholonomic mobile robot, epipolar geometry, homography relationship, chained systems.

Résumé

Dans ce mémoire, nous présentons un travail sur la navigation d'un robot mobile non holonome dans un environnement d'intérieur en utilisant seulement un capteur de vision embarqué sur le robot mobile. La tâche traitée est le suivi d'un chemin indiqué par des amers visuels. Le système de navigation utilise un ensemble d'images relais, acquises lors d'une phase d'apprentissage, pour permettre au robot d'atteindre une configuration finale exprimée par l'image visualisée sur la caméra à cet emplacement. L'état du système est exprimé dans un repère de Frenet pour pouvoir situer le robot par rapport à son chemin. La détermination de cet état pendant l'asservissement est effectuée par les calculs de l'homographie reliant chaque couple d'image relais, technique empruntée à l'approche d'asservissement visuel $2D1/2$. La loi de commande est ensuite synthétisée après passage à la forme chaînée afin de prendre en compte les contraintes non holonomes du système. Des simulations illustrent ces points et des résultats pour un robot mobile de type tricycle faisant le suivi d'amers artificiels sur le plafond de la scène sont présentés.

Mots Clés :

Robot mobile non holonome, géométrie épipolaire, relation homographique, systèmes chaînés.

Abstract

We present a framework for navigation of a nonholonomic mobile robot in an indoor environment, using only a monocular vision system embedded on the robot. We particularly focus on the path following problem. The path is indicated by some visual landmarks in the scene. The navigation system uses a set of relay images, obtained during an off-line learning step, to enable the robot to reach a goal configuration expressed by the image viewed on the camera at that position. The system state is determined in a Frenet frame to locate the robot with respect to the path. As practiced in the $2D1/2$ visual servoing scheme, we compute the homography matrix between each two images to determine the system state during the servoing phase. The control law is designed after having transformed the system into a suitable chained form to overcome its nonholonomic constraints. Simulations illustrate these points and results for a cart-like robot which navigate using artificial