



HAL
open science

Stratégies de commande référencées multi-capteurs et gestion de la perte du signal visuel pour la navigation d'un robot mobile

David Folio

► **To cite this version:**

David Folio. Stratégies de commande référencées multi-capteurs et gestion de la perte du signal visuel pour la navigation d'un robot mobile. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 2007. Français. NNT: . tel-00172507

HAL Id: tel-00172507

<https://theses.hal.science/tel-00172507>

Submitted on 17 Sep 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Présentée au

Laboratoire d'Analyse et d'Architecture des Systèmes

pour l'obtention du

Doctorat de l'Université Paul Sabatier
Spécialité Systèmes Automatiques

par

David FOLIO

Soutenu par le Fond Social Européen

STRATÉGIES DE COMMANDE RÉFÉRENCÉES MULTI-CAPTEURS ET GESTION DE LA PERTE DU SIGNAL VISUEL POUR LA NAVIGATION D'UN ROBOT MOBILE

Soutenu le 11 Juillet 2007 devant le jury

Président: Michel DEVY

Rapporteurs: François CHAUMETTE
Seth HUTCHINSON

Examineurs: Bernard BAYLE
Michel COURDESSES

Directeur de thèse: Viviane CADENAT

Invité: Philippe SOUÈRES

Remerciements

Les travaux présentés dans ce manuscrit ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) du CNRS¹. Je tiens pour cela à remercier les directeurs successifs du LAAS, Malik GHALLAB et Raja CHATILA, pour m'avoir accueilli au sein de leur laboratoire.

Lire et juger une thèse n'est pas une tâche aisée à accomplir. Aussi, je tiens tout particulièrement à exprimer ma reconnaissance à tous les membres du jury de ma thèse :

- Michel DEVY, directeur de recherches au LAAS-CNRS, pour avoir présidé ce jury et m'avoir admis dans son groupe de recherche Robotique Action et Perception (RAP).
- François CHAUMETTE, directeur de recherches à l'IRISA, responsable de l'équipe LAGADIC, et Seth HUTCHINSON, *professor of Electrical and Computer Engineering (University of Illinois at Urbana-Champaign)*, pour m'avoir fait l'honneur d'accepter la lourde tâche de rapporter sur mes travaux.
- Bernard BAYLE, maître de conférences à l'École Nationale Supérieure de Physique de Strasbourg (Université Louis Pasteur), pour sa lecture pertinente de mon manuscrit.
- Michel COURDESSES, professeur à l'Université Paul SABATIER, pour sa collaboration et son aide précieuse tout au long de ma thèse.
- Philippe SOUÈRES, chargé de recherches au LAAS-CNRS, pour sa participation et pour l'intérêt qu'il a bien voulu porter à mes travaux.
- Viviane CADENAT, maître de conférences à l'Université Paul SABATIER, pour avoir encadré mes travaux de recherche, et à qui cette thèse doit beaucoup. Un grand merci, Viviane, pour m'avoir soutenu et guidé tout au long de ces trois années de thèses et même celles qui les ont précédées. . .

Il m'est très difficile de remercier toutes les personnes qui ont contribué de près ou de loin à ce travail. Je veux néanmoins exprimer toute ma gratitude à Armelle BARELLI et Christophe CHAUMIER de la délégation régionale Midi-Pyrénées du CNRS qui au titre du Fonds Social Européen ont permis le financement de mes travaux. Je tiens aussi à exprimer ma reconnaissance à Michel TA IX et Frédéric LERASLE, tous deux maîtres de

1. Centre National de la Recherche Scientifique

conférences à l'Université Paul SABATIER, qui m'ont plus particulièrement apporté leur aide pour la réalisation de cette thèse.

J'adresse également dans une plus large mesure mes remerciements à toutes les personnes, collègues permanents, post-doct, doctorants, ingénieurs, stagiaires de l'ex-groupe RIA², ou aujourd'hui des groupes RAP, RIS³ et GEPETTO, dont l'ambiance a donné une autre dimension à ce travail. Un merci tout particulier à Sara et Matthieu pour leurs conseils éclairés sur l'utilisation de G^{en}M et leur support technique concernant l'utilisation des robots. Petite dédicace aux collègues du bureau 69-3, ayant dû me supporter tout au long de cette thèse. Un petit coucou aux anciens, certains déjà partis pour de nouveaux horizons : Frédéric Py, Olivier Lefebvre, Nicolas Do Huu, Joan Sola, Nacho, Aurélie Claudic, Sylvain Argentieri, Thierry Peynot... Je n'oublie pas ceux qui n'ont encore fini, Gauthier Hattenberger, Panagiotis Theodorakopoulos, Vincent Montreuil, Ayman Zureiki... à qui je souhaite bon courage et bonne continuation.

Je tiens aussi à remercier **Athos** et **Aramis**, deux des trois mousquetaires Super Scout, à qui cette thèse doit vraiment beaucoup... J'exprime également toute ma gratitude aux stagiaires qui ont contribué à l'avancement de mes travaux : David Flavigné, Medhi Yahia-Ouahmed, Tovo Ramaherirary.

Mes remerciements vont enfin aux enseignants de l'Université Paul SABATIER, aux personnels de l'AIP-PRIMECA⁴, ainsi qu'à toutes les personnes qui m'ont apporté leur aide lors des divers enseignements que j'ai eu l'opportunité de dispenser. Merci à tous pour votre accueil et soutien.

Un grand MERCI à ma famille qui m'a soutenu pendant ces 28 années, et sans qui rien de tout cela n'aurait été possible. Finalement, je ne saurais conclure sans un dernier remerciement à **Yasmina**, pour m'avoir supporté pendant ces deux dernières années (en espérant qu'elle pourra me supporter encore très longtemps) ainsi que tous les membres de sa famille pour leur accueil chaleureux...

2. Robotique et Intelligence Artificielle

3. Robotique et Interactions

4. Atelier Inter-établissement de Productique – Pôle de Ressources Informatiques pour la MECAnique

Table des matières

Table des figures

Table des matières

Introduction	15
Commande des robots mobiles	17
Problématique	18
Organisation du mémoire	19
I. Cadre de l'étude	21
I.1 Commande référencée vision	21
I.1.1 Asservissement visuel 3D	22
I.1.2 Asservissement visuel $2D\frac{1}{2}$	23
I.1.3 Asservissement visuel 2D	24
I.1.4 Modélisation	26
I.1.4.1 La plateforme robotique	26
I.1.4.2 Modélisation du système robotique	27
I.1.4.3 Modèle de la caméra	29
I.1.4.4 Interaction caméra/environnement	31
I.1.5 Synthèse d'un asservissement visuel 2D	33
I.1.5.1 Formalisme des fonctions de tâche	33
I.1.5.2 Application du formalisme des fonctions de tâche à la synthèse d'un asservissement visuel	36
I.1.5.3 Un exemple applicatif : réalisation d'une tâche de positionnement référencée vision	36
I.2 Navigation en environnement encombré	40

I.2.1	Évitement d'obstacles	41
I.2.2	Méthode des champs de potentiels	42
I.2.2.1	Principe général de la méthode des potentiels	42
I.2.2.2	Méthode des potentiels rotatifs	43
I.2.3	Formalisme du suivi de chemin	45
I.3	Commande référencée multi-capteurs en environnement encombré	47
I.3.1	Commande de la platine	48
I.3.2	Application conjointe du formalisme des fonctions de tâche et du suivi de chemin	50
I.3.3	Réalisation de tâches de navigation référencée vision en environnement encombré	51
I.3.3.1	Simulation d'une tâche de navigation référencée multi-capteurs exploitant (I.39)	53
I.3.3.2	Expérimentation d'un asservissement visuel et évitement de collision par la méthode des potentiels rotatifs	54
I.3.4	Bilan	55
I.4	Conclusion	56
II.	Contribution à l'évitement des occultations	57
II.1	Introduction	57
II.1.1	État de l'art	58
II.1.2	Modélisation de l'occultation	58
II.1.2.1	Projection de l'objet occultant sur le plan image	59
II.1.2.2	Distances relatives à l'objet occultant	60
II.2	Évitement simultané d'occultations et de collisions	61
II.2.1	Synthèse d'un correcteur d'évitement d'occultations et de collisions	61
II.2.2	Stratégie de commande	64
II.2.3	Résultats	65
II.2.3.1	Exemple : évitement de deux obstacles cylindriques	66
II.2.3.2	Exemple : évitement de deux murs	69
II.2.4	Bilan	70
II.3	Évitement de collision et d'occultation	71
II.3.1	Tâche d'évitement de collision $e_{co}(q)$	72
II.3.2	Tâche d'évitement d'occultation $e_{os}(q)$	73
II.3.3	Stratégie de commande	74
II.3.4	Résultats	76
II.3.4.1	Simulations	76
II.3.4.2	Expérimentation	79
II.3.5	Bilan	82
II.4	Discussion	82
II.4.1	Évitement simultané d'occultations et de collisions	82
II.4.2	Évitement découplé d'occultations et de collisions	84
II.4.3	Bilan	85

III. Contribution à la reconstruction du motif visuel	87
III.1 Introduction	88
III.2 Mouvement de la caméra et variation des informations visuelles	90
III.2.1 Le cas simple du point	90
III.2.2 Les autres primitives géométriques usuelles	92
III.2.3 Les moments de l'image	94
III.2.3.1 Les moments classiques	94
III.2.3.2 Les moments centrés	95
III.2.3.3 Choix du vecteur d'informations visuelles	96
III.2.3.4 Mise en équation du problème	96
III.2.4 Bilan et modélisation du problème	97
III.3 Le cas du point : vers une résolution analytique	100
III.3.1 Résolution dans le cas du torseur constant	100
III.3.1.1 Détermination de $z(t)$	100
III.3.1.2 Détermination de $X(t)$ et de $Y(t)$	102
III.3.1.3 Bilan	104
III.3.2 Extension : prise en compte de la variation du torseur cinématique	105
III.3.2.1 Variation par morceaux du torseur cinématique	105
III.3.2.2 Torseur cinématique variable	107
III.4 Cas général : résolution numérique	112
III.5 Conclusion	115
IV. Navigation référencée vision avec gestion de la perte du signal visuel	117
IV.1 Réalisation d'une tâche référencée vision	118
IV.1.1 Gestion de la perte du motif visuel suite à un défaut de la caméra	118
IV.1.1.1 Résultats de simulation	119
IV.1.1.2 Expérimentations	127
IV.1.2 Amélioration des performances de l'asservissement	128
IV.1.3 Bilan	131
IV.2 Asservissement visuel et évitement de collision avec gestion de la perte du motif visuel	131
IV.2.1 Stratégie de commande avec gestion de l'occultation du signal visuel	132
IV.2.2 Réalisation d'un asservissement visuel et évitement de collision par la méthode des potentiels rotatifs	134
IV.2.3 Réalisation d'une tâche référencée multi-capteurs par exploitation conjointe du formalisme des fonctions de tâche et du suivi de chemin	136
IV.2.4 Bilan	139
IV.3 Discussion	139

Conclusion	143
<hr/>	
Annexe	147
<hr/>	
A. Description du formalisme du suivi de chemin	149
<hr/>	
B. Résolution numérique d'équations différentielles	151
<hr/>	
B.1 Schéma d'Euler	152
B.2 Méthode de Runge–Kutta	153
B.3 Méthode d'Adams	153
B.3.1 Schéma d'Adams–Bashforth	154
B.3.2 Schéma d'Adams–Moulton	155
B.3.3 Méthode prédicteur/correcteur	155
B.4 Méthode BDF (<i>Backward Differentiation Formulas</i>)	156
<hr/>	
Glossaire	159
<hr/>	
Définitions	159
Notations	160
Acronymes	161
<hr/>	
Bibliographie	165
<hr/>	
Liste de mes publications	175
<hr/>	
Résumé	177
<hr/>	

Table des figures

1	Description de la problématique.	19
I.1	Asservissement visuel 3D.	22
I.2	Asservissement visuel $2D\frac{1}{2}$	23
I.3	Asservissement visuel 2D.	24
I.4	Le Super Scout.	26
I.5	Modèle du robot muni d'une caméra montée sur une platine.	27
I.6	La caméra	29
I.7	Simulation d'un asservissement visuel 2D.	38
I.8	Évolution de la fonction de tâche e_{AV}	38
I.9	La commande cinématique $\dot{q}_{AV} = (v, \omega, \varpi)$	39
I.10	Expérimentation d'un asservissement visuel 2D : trajectoire du robot.	39
I.11	Expérimentation d'un asservissement visuel 2D : évolution du motif visuel.	40
I.12	Paramètres pour l'évitement d'obstacle.	42
I.13	Exemple de distribution de potentiel classique.	43
I.14	Potentiel répulsif rotatif.	44
I.15	Évitement d'obstacle par suivi de chemin.	46
I.16	Principe d'évolution de la fonction μ_{coll}	52
I.17	Asservissement visuel et évitement de collision par application conjointe du formalisme des fonctions de tâche et du suivi de chemin.	53

I.18 Les mesures du risque de collision.	53
I.19 Asservissement visuel et évitement de collision par la méthode des potentiels rotatifs.	54
I.20 Évolution de la consigne de commande \dot{q}	55
II.1 Projection d'un obstacle sur le plan image.	59
II.2 Possibilité de passage du motif visuel en-dessous de l'objet occultant.	59
II.3 Les distances relative à l'objet occultant.	60
II.4 Simulation : évitement de deux obstacles cylindriques.	66
II.5 Le plan image.	67
II.6 Les mesures du risque de collision.	67
II.7 Les mesures du risque d'occultations.	68
II.8 Évolution de la fonction de basculement μ	68
II.9 Asservissement visuel et évitement simultané d'occlusions et de collisions.	69
II.10 Données d'évitement.	70
II.11 Différents types d'occultations.	71
II.12 Trajectoire du robot [FOLIO, EDSYS2005].	76
II.13 Données d'évitement [FOLIO, EDSYS2005].	77
II.14 Trajectoire du robot [FOLIO et CADENAT, JNRR2005].	78
II.15 Données d'évitements.	79
II.16 Asservissement visuel et évitement découplé d'occultations et de collisions.	79
II.17 Les mesures des capteurs ultrasons.	80
II.18 Les mesures du risque d'occultations.	80
II.19 Les vitesses effectivement appliquées par le robot.	81
II.20 Les vitesses synthétisées.	81
II.21 Trajectoire du robot.	83
II.22 Évitement simultané d'occultations et de collisions.	83
II.23 Trajectoire du robot.	84
II.24 Évitement découplé d'occultations et de collisions.	84
III.1 Le lien vision/mouvement.	89
III.2 Projection d'un objet quelconque dans l'image.	92
III.3 Estimation de χ_{k+1}	99
III.4 Les intervalles d'intégration.	106

IV.1	Commande référencée vision et estimation.	118
IV.2	Navigation référencée vision avec gestion de la perte du motif visuel.	119
IV.3	Erreurs pour le cas des points : $s = (X_1, Y_1 \dots, X_n, Y_n)^T$, avec $T_{ech} = 100$ ms.	120
IV.4	Erreurs pour le cas des points : $s = (X_1, Y_1 \dots, X_n, Y_n)^T$, avec $T_{ech} = 50$ ms.	120
IV.5	Les erreurs dans le cas de l'ellipse, avec $T_{ech} = 100$ ms.	123
IV.6	Les erreurs dans le cas de l'ellipse, avec $T_{ech} = 50$ ms.	123
IV.7	Erreur locale et globale.	124
IV.8	Les erreurs $\varepsilon_{ellipse}$ de reconstruction de l'ellipse.	124
IV.9	Les erreurs sur les moments, avec $T_{ech} = 100$ ms.	126
IV.10	Les erreurs sur les moments, avec $T_{ech} = 50$ ms.	126
IV.11	Trajectoire du robot obtenue pour les différentes méthodes numériques.	127
IV.12	Sous-échantillonnage de la période du capteur de vision.	129
IV.13	Erreurs pour le cas des points : $s = (X_1, Y_1 \dots, X_n, Y_n)^T$, avec $T_{ech} = 50$ ms.	130
IV.14	Les erreurs dans le cas de l'ellipse, avec $T_{ech} = 50$ ms.	130
IV.15	Les erreurs sur les moments, avec $T_{ech} = 50$ ms.	131
IV.16	Commande référencée multi-capteurs et estimation.	133
IV.17	Asservissement visuel et évitement de collision par la méthode des potentiels rotatifs.	135
IV.18	Mesures des risques de collision et d'occultation.	135
IV.19	Erreurs d'estimation, avec $T_{ech} = 50$ ms.	136
IV.20	Trajectoire du robot.	137
IV.21	Mesures d'évitement.	138
IV.22	Les différents risques.	138
IV.23	Erreurs d'estimation, avec $T_{ech} = 50$ ms.	139
A.1	Description du formalisme du suivi de chemin.	149
B.1	Principe de la méthode d'Adams.	154
B.2	Evolution de du coefficient C_E	156
B.3	Méthode BDF.	157

Introduction

“I can’t define a robot, but I know one when I see one.”

Joseph ENGELBERGER

L’UN des plus grands rêves de l’Homme est de pouvoir faire exécuter par des acteurs autres que lui-même des tâches ou activités qu’il considère comme aliénantes, dangereuses ou simplement ennuyeuses et non gratifiantes. L’apparition progressive de machines puis de la notion d’automatisation a permis de modifier la nature des travaux effectués par les hommes, donc de déporter cette aspiration vers des réponses issues des progrès des techniques et des sciences. Ainsi, l’émergence de la robotique au cours des années 1960 a su apporter de nouvelles réponses à cette attente. Avec les progrès constants dans ce domaine, les robots tendent à accomplir des tâches de plus en plus complexes avec une intervention moindre de l’homme pour les guider. Ils deviennent ainsi plus autonomes, et ils interagissent de mieux en mieux avec leur environnement pour accomplir la mission qui leur a été assignée, le robot devient alors une “machine intelligente”.

Vers la fin des années 1970, la robotique devient un thème de recherche à part entière stimulé par la robotique manufacturière et les travaux sur les bras manipulateurs. Mais, c’est surtout à partir des années 1980 qu’un intérêt croissant a été accordé à la recherche en robotique mobile motivé d’une part par l’augmentation de la puissance de calcul embarquée et l’amélioration des capteurs (miniaturisation, robustesse, etc.); et d’autre part par la plus grande diversité d’applications offertes. Toutefois, il apparaît rapidement

que, si pour la plupart des bras manipulateurs (dotés en général d'un nombre important de degrés de liberté), à *tout* chemin de l'espace des configurations⁵ correspond un mouvement exécutable dans l'espace réel, cette propriété fondamentale n'est souvent plus valable pour quelques types de robots mobiles. En effet, certains robots mobiles, à roues, ne peuvent pas se déplacer dans l'axe de leurs roues du fait de la contrainte *non-holonome* de roulement sans glissement, problème bien connu en mécanique. On souhaite alors faire effectuer aux robots mobiles des mouvements permettant de tenir compte de ces contraintes mécaniques.

La problématique de la navigation et du pilotage des robots, longtemps cantonnée au domaine de la robotique de manipulation, s'ouvre aujourd'hui vers de nombreux autres champs d'application, tels que : les véhicules, les drones aériens, les engins sous-marins, la chirurgie, etc. Dans tous les cas, il s'agit de faire évoluer des systèmes de façon sûre dans des milieux imparfaitement connus en contrôlant les interactions entre le robot et son environnement. Ces interactions peuvent prendre différents aspects : actions de la part du système robotique (se positionner par rapport à un objet, manœuvrer pour se garer, etc.), réactions vis-à-vis d'événements provenant du monde qui l'entoure (éviter des obstacles, poursuivre une cible mobile, etc.). Le degré d'autonomie du robot réside alors dans sa capacité à prendre en compte ces interactions à tous les niveaux de la tâche robotique. En premier lieu, lors de la planification, cela se fera par l'acquisition, la modélisation et la manipulation des connaissances sur l'environnement et sur l'objectif à réaliser. Ensuite, durant l'exécution, il s'agira d'exploiter les données perceptuelles pour adapter au mieux le comportement du système aux conditions de la mission qu'il doit réaliser. Ainsi, classiquement, une tâche de navigation se décompose en plusieurs phases :

- ① **Perception** : selon l'objectif fixé et les difficultés rencontrées, le système de navigation acquiert les informations nécessaires pour l'accomplissement de la tâche ;
- ② **Modélisation** : le robot procède à l'analyse de ces données perceptuelles afin de produire une représentation interprétable de l'environnement ;
- ③ **Localisation** : pour décider des déplacements, le robot doit déterminer sa situation⁶ par rapport au modèle produit ;
- ④ **Planification** : le robot décide de l'itinéraire, puis détermine la trajectoire ou les mouvements nécessaires, ainsi que les conditions de leur réalisation ;
- ⑤ **Action** : le robot doit s'asservir sur les déplacements décidés pour l'accomplissement de sa mission.

La perception et la modélisation de l'environnement en vue de la planification des actions est un thème majeur en robotique [LAUMOND, 2001, RIVES et DEVY, 2001]. Il se décline en de nombreuses variantes selon la connaissance a priori dont on dispose : elle peut être complète permettant ainsi la planification hors-ligne de la tâche, ou bien absente, nécessitant alors une acquisition en ligne du modèle durant une phase d'exploration. De même qu'il est important de prendre en compte le processus de perception très tôt dans la description de la tâche, il est tout aussi indispensable de contrôler l'interaction entre le robot et son environnement lors de son exécution. Cela se traduit par la prise en compte explicite d'informations perceptuelles, d'une part, dans la constitution de boucles de commande robustes et, d'autre part, dans la détection d'événements externes nécessitant une modification du comportement du robot. Dans les deux cas, il s'agit

5. Une définition est proposée dans le glossaire à la page 159.

6. La situation d'un corps décrit sa position et son orientation par rapport à un repère.

de rendre robuste le système face à une certaine variabilité des conditions d'exécution de la mission. Cette variabilité peut provenir d'erreurs de mesures ou de modèle, dues aux capteurs ou au système commandé. Mais elle peut également provenir du monde dans lequel le système évolue qui peut être incertain ou dynamique. Pour pallier à ces incertitudes, certains auteurs [QUINLAN, 1994], [KHATIB *et al.*, 1997], [LAMIRAUX *et al.*, 2004] proposent des solutions permettant de déformer la trajectoire initialement planifiée, de manière à éviter les obstacles non prévus. Cependant, ce genre d'approche, qualifiée de globale, se base sur la modification d'un chemin ou d'une trajectoire préalablement établis. Par conséquent, elles requièrent la construction d'une carte pour générer l'itinéraire, ainsi que la localisation du robot par rapport à celui-ci. D'autres types d'approches, dite locales, que nous considérons dans notre étude, permettent d'intégrer directement les informations sensorielles dans la boucle de commande. Sur la base de ces informations l'objectif est alors de synthétiser un asservissement qui guide le robot vers son but tout en tenant compte des variations du milieu où évolue le système, sans connaissance a priori sur celui-ci. Toutefois, ne disposant pas d'un modèle de l'environnement, les méthodes locales souffrent généralement des problèmes de minima locaux : c'est-à-dire des configurations où la commande appliquée est *nulle*, bien que le robot ne se trouve pas encore à la situation désirée. Néanmoins cette approche permet la synthèse de lois de commande très *réactives*, *précises* et *robustes*.

Commande des robots mobiles

La synthèse de lois de commande permettant de piloter efficacement un robot mobile apparaît comme un enjeu important dans la définition d'une tâche robotique. En effet, il faut pouvoir asservir le système robotique de manière à réaliser les mouvements désirés, tout en tenant compte de la variabilité et de l'incertitude que l'on peut avoir de l'environnement. La littérature regorge de méthodes concernant la commande des robots mobiles. Généralement, ces lois de commande sont posées en termes de régulation à zéro d'une erreur entre la situation du véhicule et une situation définie a priori par la trajectoire, le chemin ou la posture de référence. Si les techniques classiques de commande non-linéaire offrent des solutions pour les problèmes de suivi de chemin ou de trajectoire, telles que celles proposées par exemple dans [SAMSON, 1992] [DE LUCA *et al.*, 1998], il est en revanche plus complexe d'asservir un robot en une situation prédéfinie. En effet, il a été montré qu'il n'existe pas de retour d'état continu permettant d'atteindre cet objectif [BROCKETT, 1983]. En outre, les lois de commande synthétisées sont généralement de type retour d'état et nécessitent donc la reconstruction de l'état du robot en se basant sur les informations sensorielles.

On distingue deux types de capteurs : les *capteurs proprioceptifs* (odomètres, gyroscopes...) qui permettent d'avoir une information sur l'état interne du robot, et les *capteurs extéroceptifs* (caméras, ultrasons, lasers, etc.) qui renseignent sur la situation du robot par rapport à son environnement. L'utilisation exclusive des capteurs proprioceptifs paraît extrêmement limitative. En effet, ils fournissent uniquement des informations sur l'état interne du système. Ainsi, on ne peut espérer localiser précisément un robot en se basant uniquement sur ces mesures, du fait des problèmes de dérives et de précision des capteurs, et des incertitudes associées à la modélisation de l'ensemble robotique. De plus, ces informations proprioceptives ne tiennent pas compte de la variabilité de

l'environnement. En revanche, l'exploitation des données extéroceptives permet d'avoir une vision plus précise du véhicule dans son environnement, et d'avoir des indications sur les événements extérieurs. Les informations fournies sont souvent utilisées dans des procédures de haut niveau pour compléter ou corriger le processus de commande. Par exemple les travaux développés dans [LAMIRAUX *et al.*, 2004] proposent une méthode de déformation de la trajectoire planifiée en fonction de la distance aux obstacles non prévus détectés par un laser. Mais là encore il s'agit de réaliser une tâche de navigation dans un espace dont le modèle est connu a priori permettant la planification d'une trajectoire préalable. La construction d'une carte de l'espace dans lequel évolue le robot nécessite généralement une phase d'apprentissage, et peut s'avérer longue et fastidieuse (d'autant plus si l'on doit en changer fréquemment, ou si l'on recherche une grande précision). Pour la réalisation de missions robotiques disposant d'un minimum d'informations sur le contexte de la tâche et de l'environnement, il paraît particulièrement intéressant d'intégrer les mesures sensorielles directement dans la boucle de commande, on parle alors de la *commande référencée capteur*.

La commande référencée capteur consiste à exprimer les tâches robotiques à réaliser, non plus dans l'espace des configurations, mais directement dans l'espace du capteur sous la forme d'une relation locale entre le robot et son environnement. On cherche ainsi à définir des commandes *en boucle fermée* sur les informations provenant des capteurs et non plus sur la configuration du robot. De ce fait les lois de commande généralement synthétisées dans le cadre de cette approche sont de type *retour de sortie* et permettent de réaliser les tâches désirées de manière plus précise. Ainsi, toute tâche pouvant s'exprimer sous la forme d'une relation locale entre le robot et l'environnement sera réalisable par le biais de la commande référencée capteur. Toutefois, cette approche requiert l'utilisation de capteurs rapides, capables de fournir les mesures à une fréquence compatible avec la fréquence d'échantillonnage de l'asservissement. En effet, les travaux en commande référencée capteurs ont surtout porté sur des capteurs choisis pour la simplicité et l'efficacité d'obtention des informations fournies. Ainsi, le suivi d'un mur à une distance donnée ou l'évitement d'un obstacle peut être défini sur la base de données provenant de capteurs de proximité (ultrasons, laser, etc.), comme le proposent VICTORINO *et al.* [VICTORINO *et al.*, 2001] ou KHATIB, M. [KHATIB, 1996]. Mais, parmi les capteurs extéroceptifs existants, la vision a plus particulièrement retenu l'attention des roboticiens. En effet, il s'agit d'un capteur capable de fournir des informations très riches (selon le traitement mis en œuvre), permettant ainsi de réaliser des tâches très variées. La commande référencée capteurs devient alors la *commande référencée vision* [HUTCHINSON *et al.*, 1996] [CHAUMETTE, 2002]. Ce type d'asservissement consiste à prendre en compte des informations visuelles, issues généralement d'une ou plusieurs caméras embarquées sur le robot ou déportées, pour contrôler le mouvement du système robotique. Les techniques d'asservissement visuel exploitent des données perceptuelles de nature 2D ou 3D provenant d'une image. Ces informations sont alors exploitées dans la boucle de commande pour guider le robot dans l'accomplissement de sa mission.

Problématique

Dans le cadre de nos travaux, nous nous intéresserons à la définition de schémas de commande en boucle fermée sur des informations visuelles. L'aspect réactif vis-à-

vis d'événements extérieurs perturbant le déroulement de la tâche sera au cœur de nos préoccupations. En effet, l'exploitation des seules données visuelles, comme nous le verrons dans la suite de ce manuscrit, ne permet pas de définir des tâches robotiques élaborées. L'asservissement visuel est classiquement basé sur la seule régulation à zéro de l'erreur entre ce que le robot voit réellement et ce qu'il doit voir lorsque la tâche est parfaitement réalisée. Cette approche s'avère donc mal adaptée pour traiter les problèmes liés à la présence d'obstacles dans l'environnement du robot. C'est précisément ces problèmes que nous avons voulu étudier dans le cadre de cette thèse. De précédents travaux [CADENAT, 1999] les ont déjà abordés et proposaient de premières solutions. L'approche considérée consistait à définir des lois de commande *référencée multi-capteurs*, c'est-à-dire exploitant différents types de mesures sensorielles permettant de caractériser d'une part l'objectif de la mission qui sera défini à partir d'informations visuelles, et, d'autre part, d'éviter les éventuelles collisions. Toutefois les techniques proposées considéraient uniquement des obstacles dont la hauteur étaient inférieure à celle de la caméra, et qui, par conséquent, ne pouvaient pas perturber les informations visuelles. Notre intention, ainsi que notre contribution, est de nous soustraire à cette contrainte, et de considérer des obstacles pouvant occulter (ou non) le motif visuel. Notre objectif est donc finalement de définir un ensemble de lois de commande référencées multi-capteurs permettant à un robot mobile de réaliser une tâche référencée vision dans un environnement encombré d'obstacles susceptibles d'occulter le motif visuel (voir figure 1). Pour cela, nous nous appuyerons sur les techniques d'asservissement visuel et d'évitement d'obstacles existantes, et nous chercherons à les étendre afin de gérer les phénomènes d'occultation. Il s'agit ainsi d'une part de synthétiser des asservissements performants exploitant au mieux les données perceptuelles nécessaires à la réalisation de la mission, et d'autre part de pouvoir intégrer d'autres types de mesures pour naviguer sans collision ni occultation dans un espace contraint.

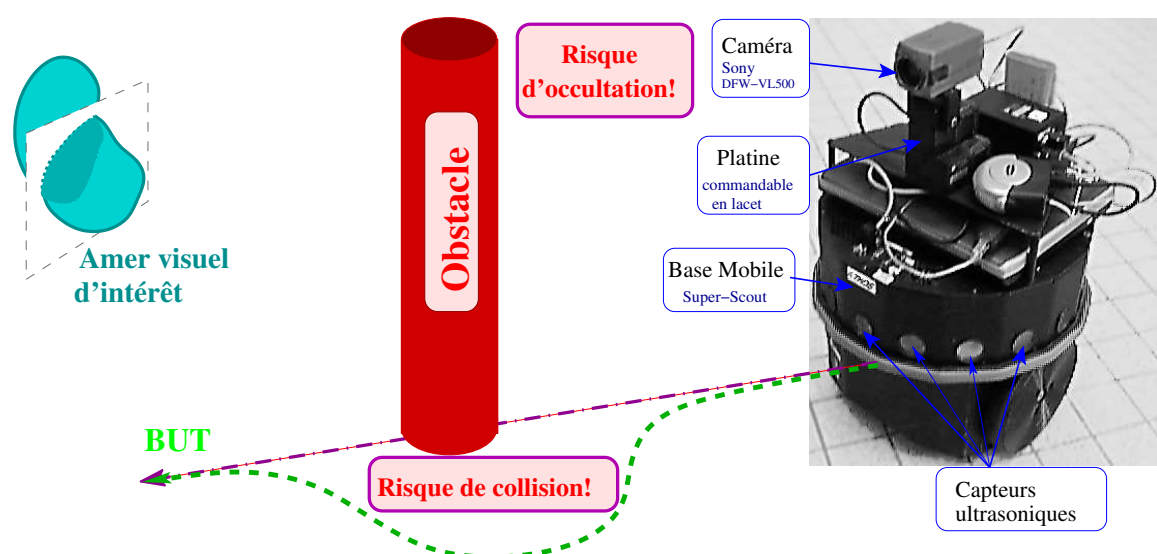


FIG. 1 – Description de la problématique.

Organisation du mémoire

Cet ouvrage est organisé en quatre chapitres structurés de la manière suivante. Dans le chapitre I, nous présentons le cadre de l'étude et rappelons les principaux concepts et formalismes sur lesquels sont basés nos travaux. Ensuite, dans le chapitre II, nous proposons deux méthodes permettant d'éviter conjointement les occultations et les collisions pendant l'exécution d'une tâche référencée vision en environnement encombré. Ces approches s'avérant limitées dans certains cas, nous avons cherché à étendre ces résultats en développant, dans le chapitre III, des techniques capables de reconstruire le signal visuel lorsque celui-ci n'est plus disponible, de manière à pouvoir tolérer les occultations si cela est nécessaire. Nous avons ainsi modélisé le problème d'estimation selon le type d'indice visuel considéré puis proposé deux classes de méthodes permettant de le résoudre. Dans le chapitre IV, nous exploitons ces résultats dans le contexte applicatif de la navigation référencée vision et les insérons au sein de nos boucles de commande. Nous validons alors l'ensemble de ces approches en présentant des résultats de simulation et d'expérimentation. Finalement, nous concluons sur les travaux réalisés et présentons les perspectives qui peuvent être envisagées.

Chapitre I

Cadre de l'étude

DANS le cadre de notre étude, nous nous intéressons au problème de la navigation d'un robot mobile guidé par la vision dans un environnement encombré. Plus particulièrement, les travaux décrits ici ont été réalisés au sein du groupe RAP du LAAS. Les recherches menées au sein de ce groupe portent sur l'étude et la conception de machines autonomes, intégrant des capacités de perception, d'action et de raisonnement. Ces machines doivent interagir rationnellement avec un environnement variable et évolutif pour la réalisation de tâches diverses. Parmi les thématiques de recherche abordées par le groupe, une attention particulière est donnée à la commande des robots mobiles autonomes. En effet, cette dernière pose un problème de recherche très riche dont l'une des difficultés provient des contraintes non-holonomes de roulement sans glissement qui interdisent les déplacements latéraux instantanés. Cette problématique intervient sur bon nombre d'architectures de robots mobiles à roues, dont le robot mobile "Super Scout" que nous avons considéré dans le cadre de nos travaux. Le Super Scout développé par la société Nomadic Technologies, est mis à notre disposition par l'AIP-PRIMECA de Toulouse. Ce robot est composé d'une base mobile de type char, dotée de seize capteurs ultrasons, et d'une caméra montée sur une platine commandable en lacet.

Nous rappellerons dans un premier temps quelques éléments de base concernant la commande d'un robot mobile basée sur la vision, avant de présenter les techniques de navigation en environnement contraint.

I.1 Commande référencée vision

La commande référencée vision [CORKE, 1996] [HUTCHINSON *et al.*, 1996] [CHAUMETTE, 2002] [CHAUMETTE et HUTCHINSON, 2006, 2007] consiste à contrôler les mouvements d'un système robotique en utilisant des informations visuelles, notées \mathbf{s} , issues d'une ou plusieurs caméras (ou plus généralement d'un capteur de vision) embarquées ou non sur le système.

De nombreux travaux sont basés sur l'exploitation de ces données pour réaliser différents objectifs tels que le positionnement face à un objet, son suivi, sa saisie, etc. La première utilisation de la vision en boucle fermée est due à SHIRAI et INOUE [1973] qui décrivent comment un capteur de vision pouvait augmenter la précision du positionnement. On parlait alors de retour par vision (*visual feedback*). Mais c'est à HILL et PARK [1979] que l'on doit l'apparition du terme asservissement visuel (*visual servoing*). Plusieurs approches ont depuis vu le jour. SANDERSON et WEISS [1980] en propose ainsi deux grandes classes selon l'utilisation des informations visuelles : l'asservissement visuel 3D (ou *position-based control*) et l'asservissement visuel 2D (ou *image-based control*). Notons qu'il existe aussi des approches intermédiaires plus récentes telles que l'asservissement visuel $2D\frac{1}{2}$ [MALIS, 1998] ou $\frac{d2D}{dt}$ [CRÉTUAL, 1998B]. Nous présentons ci-après les structures de commande 3D et $2D\frac{1}{2}$, avant de détailler l'asservissement visuel 2D que nous avons plus particulièrement considéré dans le cadre de nos travaux.

I.1.1 Asservissement visuel 3D

L'asservissement visuel 3D utilise en entrée de la boucle de commande des informations tridimensionnelles, à savoir la situation \mathbf{r} de la caméra, par rapport à l'objet d'intérêt. La tâche à réaliser s'exprime alors sous la forme d'une situation de référence \mathbf{r}^* à atteindre (voir figure I.1). La commande repose ainsi sur la détermination de la situation \mathbf{r} de la caméra, à partir des informations visuelles extraites de l'image (voir figure I.1.b).

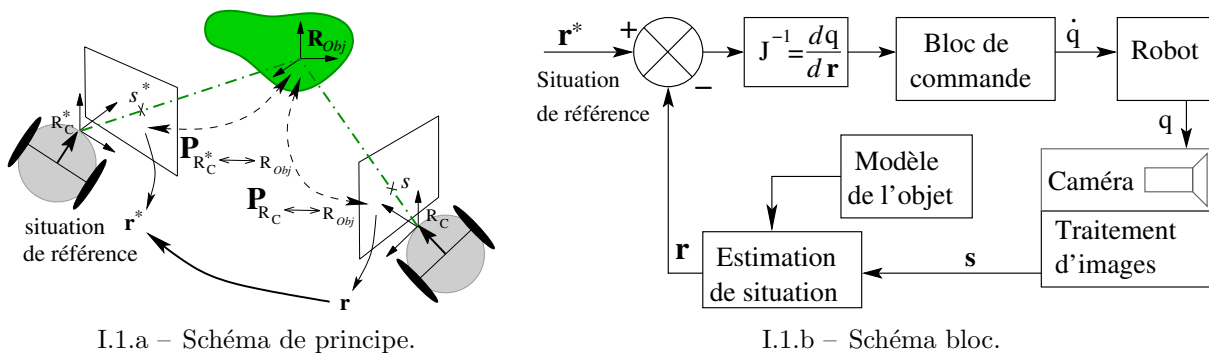


FIG. I.1 – Asservissement visuel 3D.

De nombreuses méthodes permettent d'estimer la situation d'une caméra par rapport à un objet à partir de l'image perçue de cet objet. Elles reposent très généralement sur la connaissance a priori d'un modèle (3D ou 2D) de l'objet et des paramètres intrinsèques de la caméra. Ces méthodes utilisent des informations visuelles de différentes nature, telles que des points [HORAUD *et al.*, 1989] [HARALICK *et al.*, 1989], des droites [DHOME *et al.*, 1989]... Toutefois des travaux, issus des recherches en reconstruction 3D par vision dynamique, permettent d'estimer le modèle de l'objet d'intérêt, ou de localiser la caméra à partir de mesures de mouvement 2D ou 3D, élargissant ainsi les tâches considérées.

Finalement, l'asservissement visuel 3D s'exprimant directement dans l'espace des configurations¹ permet la définition de lois de commande extrêmement simples pour aller itérativement d'une situation à une autre. Toutefois, l'inconvénient qui en résulte est qu'aucun contrôle véritable dans l'image n'est effectué ce qui implique que l'objet peut

1. Une définition est proposée dans le glossaire à la page 159.

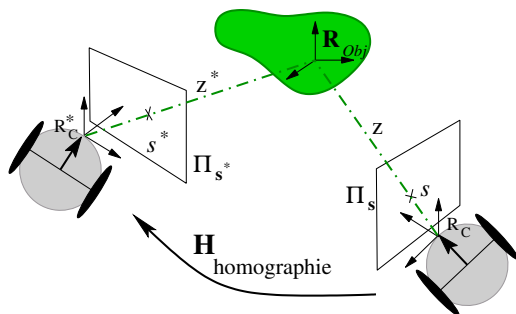
très bien être perdu. De plus, cette approche requiert une interprétation du motif visuel pour caractériser la situation du robot, et nécessite donc une reconstruction de l'état du système. On y retrouve alors les mêmes problèmes d'incertitude et de précision des schémas de commande des robots mobiles classiques.

I.1.2 Asservissement visuel $2D\frac{1}{2}$

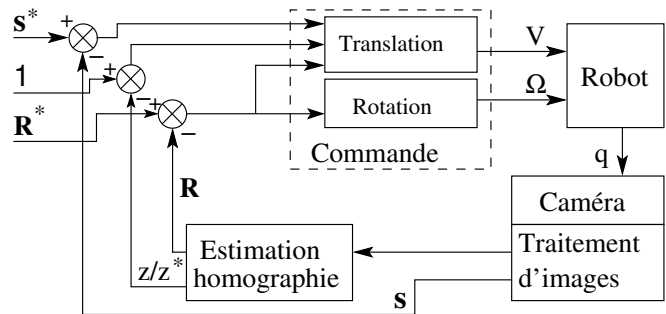
L'asservissement visuel $2D\frac{1}{2}$ [MALIS, 1998][MALIS et CHAUMETTE, 2000][FANG *et al.*, 2002, 2005] exploite des informations à la fois de nature 2D et 3D. Cette technique est basée sur l'estimation de l'homographie, notée \mathbf{H} , qui relie l'image d'au moins trois points entre différents plans projectifs [FAUGERAS, 1993] (voir figure I.2). L'homographie est une application projective bijective, correspondant à une transformation linéaire entre deux plans projectifs. Plus précisément, elle permet d'établir (à un facteur d'échelle α près) une bijection entre un objet de l'espace 3D et une image 2D, ou bien entre deux images 2D d'un même objet. Dans ce dernier cas, elle permet de lier les projections d'un ensemble de points 3D P_i , appartenant à un même plan, sur deux plans image Π_1 et Π_2 par la relation :

$$p_{i/\Pi_2} = \alpha \mathbf{H} p_{i/\Pi_1}, \quad \forall \alpha \in \mathbb{R}^* \quad (\text{I.1})$$

où p_{i/Π_1} et p_{i/Π_2} représentent respectivement la projection des points P_i sur les plans Π_1 et Π_2 . Dans le cas d'un objet plan, un minimum de quatre points appariés sur les images courante et désirée permet l'estimation de l'homographie par la résolution d'un système linéaire. Dans le cas d'un objet quelconque, un appariement de huit points est nécessaire.



I.2.a – Schéma de principe.



I.2.b – Schéma bloc.

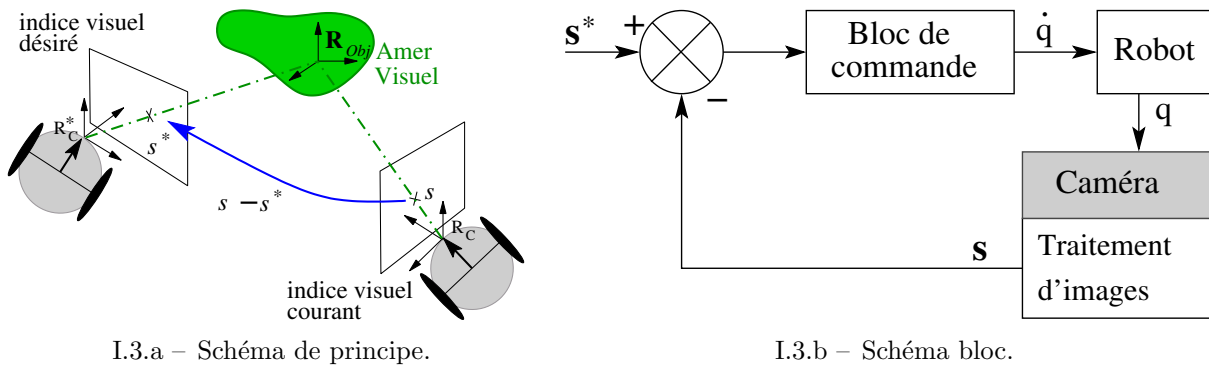
FIG. I.2 – Asservissement visuel $2D\frac{1}{2}$

Ainsi, l'asservissement visuel $2D\frac{1}{2}$ repose sur la détermination de l'homographie \mathbf{H} entre l'image courante et l'image désirée. En effet, à partir de cette homographie il est possible de calculer le déplacement en rotation \mathbf{R} que la caméra doit effectuer pour atteindre la situation spécifiée, ainsi que la direction de son déplacement en translation. Cette matrice \mathbf{H} fournit également le rapport z/z^* entre les distances courante et désirée de la caméra à l'objet (*cf.* figure I.2.a). Comme la translation à réaliser n'est connue qu'à un facteur d'échelle α près, un asservissement visuel purement 3D est impossible. Cependant, il est possible d'aboutir à une solution grâce à la combinaison des informations 2D (fournies par l'image) et 3D disponibles. Le vecteur des informations visuelles \mathbf{s} est donc défini sur la base de donnée 2D et 3D. Ainsi, l'asservissement visuel $2D\frac{1}{2}$ est une

approche intermédiaire entre l'asservissement 3D et 2D. La boucle d'asservissement ainsi synthétisée permet de séparer la rotation et la translation de la caméra, et d'obtenir un fort découplage de la loi de commande (voir figure I.2.b). L'avantage majeur de cette approche est que la connaissance du modèle géométrique 3D de l'objet n'est plus nécessaire. La seule information 3D utilisée dans la synthèse de la commande est la profondeur désirée z^* d'un point de l'objet.

I.1.3 Asservissement visuel 2D

Les techniques d'asservissement visuel 2D utilisent directement les informations visuelles, notée \mathbf{s} , extraites de l'image (voir figure I.3). La tâche à réaliser est alors spécifiée directement dans l'image en termes d'indices visuels de référence \mathbf{s}^* à atteindre. La loi de commande consiste alors à contrôler le mouvement de la caméra de manière à annuler l'erreur entre les informations visuelles courantes $\mathbf{s}(t)$ et le motif désiré \mathbf{s}^* (voir figure I.3.b). Cette approche permet donc de s'affranchir de l'étape de reconstruction 3D de la cible et des problèmes qui y sont liés.



I.3.a – Schéma de principe.

I.3.b – Schéma bloc.

FIG. I.3 – Asservissement visuel 2D.

Le choix des informations visuelles et l'obtention de la relation les liant au mouvement de la caméra sont deux aspects fondamentaux de l'asservissement visuel 2D. Cette relation, obtenue par dérivation des informations sensorielles \mathbf{s} par rapport à la situation \mathbf{r} de la caméra, est définie par une matrice appelée jacobienne de l'image ou *matrice d'interaction*. La synthèse de la commande repose sur l'élaboration d'une méthode de calcul explicite de cette matrice souvent associée à des primitives géométrique simples, telles que des points [FEDDEMA et MITCHELL, 1989], des droites, des cercles, des ellipses [CHAUMETTE, 1990], ou encore des moments de l'image [CHAUMETTE, 2004]. Dans ce contexte, les lois de commande synthétisées dépendent de la nature des informations visuelles choisies. Par conséquent la tâche robotique à réaliser est elle aussi dépendante de l'objet observé par la présence ou non d'informations visuelles dont on est capable de calculer la matrice d'interaction associée. C'est pourquoi, certains travaux ont proposé des extensions, en traitant des informations visuelles pouvant caractériser l'aspect global de l'image. Ainsi, par exemple, l'exploitation des informations visuelles dynamiques [CRÉTUAL, 1998A,B] [SANTOS-VICTOR et SANDINI, 1997] a conduit à l'élaboration de l'asservissement visuel $\frac{d^2D}{dt}$. Néanmoins, les asservissements visuels 2D sont, d'une manière générale, des lois de commandes relativement rapides à calculer, puisqu'il n'y a pas de phase de reconstruction 3D. Ce gain de temps n'est pas négligeable puisque le délai d'application de la commande est

réduit, ce qui contribue à la stabilité du système [WUNSCH et HIRZINGER, 1997]. De plus, les asservissements basés sur l'image permettent la réalisation de tâches de manière très efficace et précise. C'est ainsi que ce type de commande se rencontre de plus en plus dans différents domaines d'application. Citons par exemple la conduite d'engins sous-marins [RIVES et BORRELLY, 1997A,B], de véhicules autonomes personnels [MARTINET et THIBAUD, 2000] ou agricoles [KHADRAOUI *et al.*, 1998]. La robotique chirurgicale apparaît également comme un champ d'application privilégié des dernières avancées de l'asservissement visuel 2D [KRUPA *et al.*, 2003] [VITRANI *et al.*, 2006].

Différentes techniques de l'automatique classique permettent de synthétiser des lois de commande exploitant des données visuelles. Par exemple, MARTINET *et al.* [1998] utilisent les techniques de commande \mathcal{H}_∞ , développées dans [DOYLE *et al.*, 1989], pour synthétiser une commande robuste aux erreurs de modélisation. On peut aussi noter les travaux de D. BELLOT [BELLOT et DANÈS, 2001B,A] [BELLOT, 2002] qui portent sur la synthèse et l'analyse de lois de commande par asservissement visuel à base d'Inégalités Linéaires Matricielles (*Linear Matrix Inequality, LMI*). Les commandes ainsi synthétisées garantissent a priori certaines propriétés telles que la convergence, la non-saturation des actionneurs, la visibilité de l'objet, la satisfaction de contraintes 3D pendant le déplacement, etc. Plus récemment GAO, B. [GAO, 2006] a proposé une extension de ces dernières approches. La méthode développée permet de caractériser un correcteur et un domaine de stabilité associé prenant en compte une incertitude bornée sur la profondeur des points de la cible, la visibilité des indices visuels et les contraintes sur la vitesse et l'accélération de la caméra. Toutefois ces approches sont généralement conservatives, c'est-à-dire que la commande générée est très faible, et donc que le délai de convergence est long.

Une autre approche consiste à exploiter le formalisme des fonctions de tâches proposé par SAMSON C., LE BORGNE M. et ESPIAU B. [SAMSON *et al.*, 1991]. Ce formalisme permet de modéliser la tâche considérée sous la forme d'une *fonction de tâche*², puis à synthétiser un asservissement permettant de la réguler à zéro. Cette technique offre un cadre rigoureux pour la synthèse de lois de commande référencées capteur. Initialement conçue pour des bras manipulateurs [CHAUMETTE, 1990] [ESPIAU *et al.*, 1992], ce formalisme ne peut être directement étendu au cas des robots mobiles non-holonomes du fait de la contrainte de roulement sans glissement. En effet, cette contrainte entrave les mouvements de la caméra, et les tâches robotiques nécessitent alors la réalisation de manœuvres qui peuvent conduire à la perte du motif visuel. Une première solution, proposée par PISSARD-GIBOLLET et RIVES [1995], a consisté à introduire (au moins) un degré de liberté supplémentaire permettant à la caméra de se déplacer indépendamment du robot mobile. Ainsi, bien que la base mobile reste non-holonome, les mouvements de la caméra deviennent holonomes, et il est alors possible d'appliquer le formalisme des fonctions de tâches. Plus récemment, un cadre plus générique, le formalisme des *fonctions transverses* [MORIN et SAMSON, 2003], a vu le jour. Cette approche permet la synthèse de schémas de commande pour la stabilisation de systèmes non-linéaires.

Les travaux menés dans le cadre de cette thèse reposent sur le formalisme des fonctions de tâches. Ainsi nous chercherons d'abord à modéliser le robot Super Scout considéré, la caméra embarquée, ainsi que l'interaction entre cette dernière et son environnement. Nous présenterons ensuite brièvement le formalisme des fonctions de tâche que nous exploiterons pour définir un asservissement visuel 2D.

2. Nous détaillons le formalisme des fonctions de tâche au paragraphe I.1.5.1.

I.1.4 Modélisation

L'asservissement visuel requiert la détermination du lien entre l'évolution des indices visuels dans l'image et les mouvements de la caméra. Dans ce but, nous décrirons en premier lieu la plateforme expérimentale considérée avant de déterminer son modèle cinématique. Nous nous intéresserons ensuite à la modélisation de la caméra et à celle de son interaction avec son environnement.

I.1.4.1 La plateforme robotique

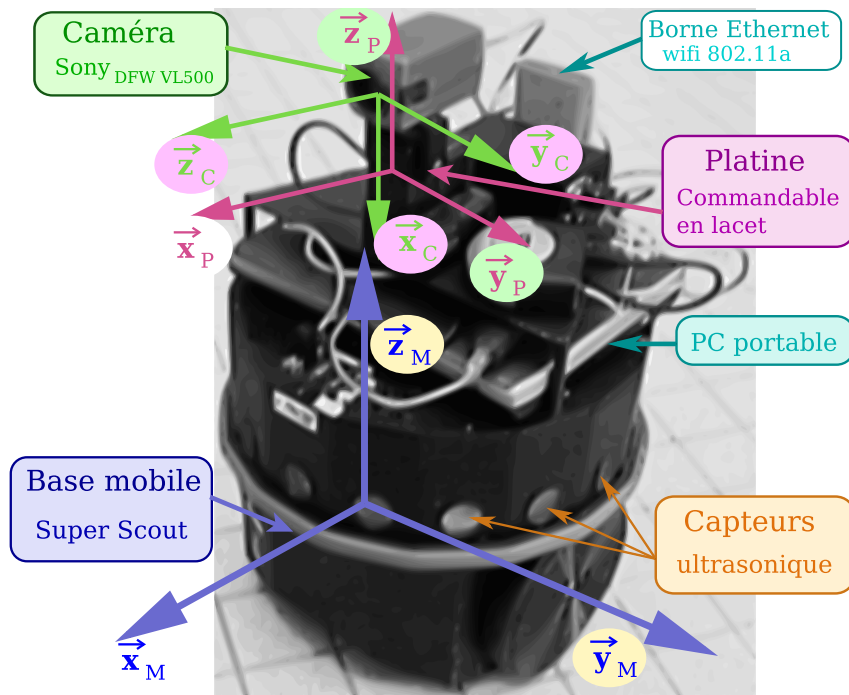


FIG. I.4 – Le Super Scout.

Les lois de commande qui sont présentées ici ont été implémentées sur le robot mobile Super Scout présenté sur la figure I.4. Ce robot a été développé par la société Nomadic Technologies, et mis à notre disposition par l'AIP-PRIMECA de Toulouse. Le Super Scout est constitué d'une base mobile de type *char* principalement destinée à une utilisation en intérieur. D'architecture cylindrique (35 cm de diamètre pour 35 cm de hauteur), et pèse 15kg environ (sans les batteries). Il possède deux roues motrices indépendantes commandables en vitesse. Le Super Scout est également doté d'une ceinture de seize capteurs ultrasoniques, de contacteurs (*bumpers*) et d'odomètres. Il comprend un système de commande multiprocesseurs bas niveau qui commande à la fois les capteurs, le mouvement ainsi que la communication. À un niveau plus élevé, le Super Scout est piloté par un PC-portable embarqué, évoluant sous une distribution Linux et disposant d'une architecture logicielle nommée $G^{\text{en}}M$ (*Generator of Module*) élaborée au LAAS [FLEURY, 1996] [FLEURY *et al.*, 1997] [FLEURY et HERRB, 2001]. De plus, notre plateforme est équipée d'une caméra couleur SONY DFW-VL 500. Enfin, pour faciliter la mise en œuvre de commandes référencées vision, la caméra a été montée sur une platine orientable en lacet (*i.e.* rotation autour de l'axe \vec{z}_P) commandable en vitesse.

I.1.4.2 Modélisation du système robotique

Notre objectif ici est d'établir la modélisation de notre système robotique représentée sur la figure I.5. Nous définissons :

- $\mathcal{R}_0(O, \vec{x}_0, \vec{y}_0, \vec{z}_0)$ le repère lié à la scène ;
- $\mathcal{R}_M(M, \vec{x}_M, \vec{y}_M, \vec{z}_M)$ le repère lié à la base mobile ;
- $\mathcal{R}_P(P, \vec{x}_P, \vec{y}_P, \vec{z}_P)$ le repère lié à la platine ;
- D_x la longueur de l'entraxe $[MP]$, valant 10 cm ;
- $\mathcal{R}_C(C, \vec{x}_C, \vec{y}_C, \vec{z}_C)$ le repère lié à la caméra ;

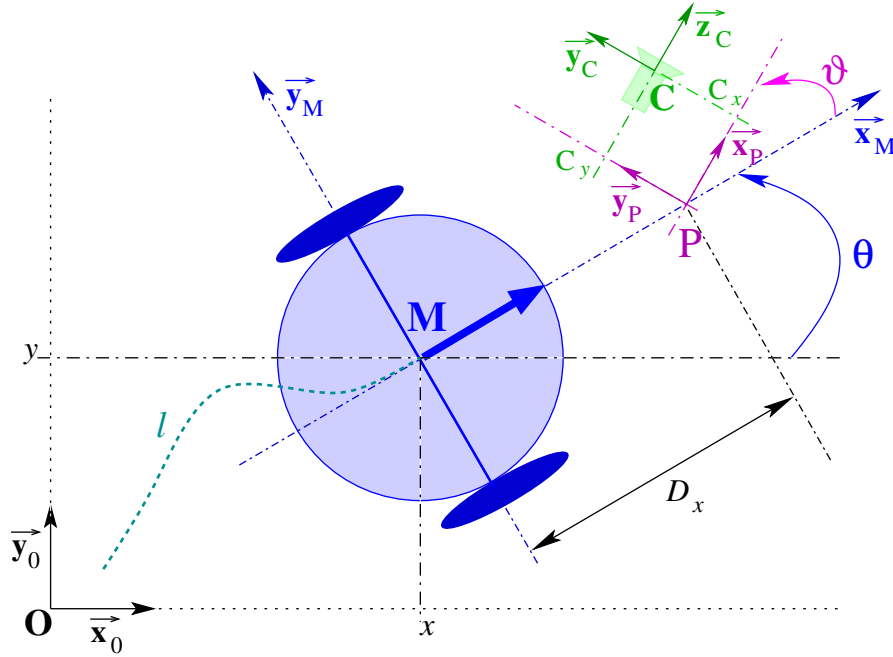


FIG. I.5 – Modèle du robot muni d'une caméra montée sur une platine.

La position de la base mobile est décrite dans le plan par le couple (x, y) désignant les coordonnées du centre M de la base mobile dans le repère global lié à la scène \mathcal{R}_0 ; son orientation est définie par rapport à l'axe \vec{x}_0 par l'angle θ . La position relative de la caméra par rapport à la platine est décrite par le vecteur $\vec{PC} = (C_x \ C_y \ 0)^T$ défini dans le repère \mathcal{R}_P de la platine, son orientation étant donnée par l'angle $\vartheta = \widehat{\vec{x}_M \vec{x}_P}$. Par ailleurs, nous définissons le vecteur des paramètres q , déduit des capteurs proprioceptifs, par :

$$q = (l \ \theta \ \vartheta)^T \quad (\text{I.2})$$

où l représente l'abscisse curviligne du point M sur la trajectoire du robot par rapport au repère de la scène \mathcal{R}_0 . Si nous dérivons ce vecteur q (I.2) par rapport au temps, nous obtenons :

$$\dot{q} = \frac{dq}{dt} = (v \ \omega \ \varpi)^T \quad (\text{I.3})$$

où v , ω désignent respectivement les vitesses linéaire et angulaire de la base mobile, ϖ correspondant à la vitesse de rotation de la platine par rapport au robot. Notre robot étant commandable en vitesse, \dot{q} sera notre vecteur de commande. Le tableau I.1 présente les vitesses limites des actionneurs du Super Scout muni de sa platine.

Base mobile			Platine		
v_{\min}	0.1 m/s	ω_{\min}	1 deg/s	ϖ_{\min}	1 deg/s
v_{\max}	1 m/s	ω_{\max}	90 deg/s	ϖ_{\max}	90 deg/s

TAB. I.1 – Vitesses minimales et maximales du système robotique.

La commande référencée vision consiste à définir le mouvement de la caméra de manière à amener le motif de l'image courante vers un motif prédéfini. Ce mouvement peut être décrit par son torseur cinématique, noté $\mathcal{T}_{C/\mathcal{R}_0}$, par rapport au repère de la scène \mathcal{R}_0 et exprimé dans le repère de la caméra \mathcal{R}_C [CHAUMETTE, 1990][PISSARD-GIBOLLET et RIVES, 1991]. Classiquement, ce torseur est défini par : $\mathcal{T}_{C/\mathcal{R}_0} = (V_{C/\mathcal{R}_0}^T \ \Omega_{\mathcal{R}_C/\mathcal{R}_0}^T)^T$, où $V_{C/\mathcal{R}_0} = (V_{\vec{x}_C} \ V_{\vec{y}_C} \ V_{\vec{z}_C})^T$ et $\Omega_{\mathcal{R}_C/\mathcal{R}_0} = (\Omega_{\vec{x}_C} \ \Omega_{\vec{y}_C} \ \Omega_{\vec{z}_C})^T$ désignent les vitesses de translation et de rotation de la caméra par rapport à l'environnement. Comme notre système robotique est commandable en vitesse, il est nécessaire de déterminer $\mathcal{T}_{C/\mathcal{R}_0}$ en fonction des commandes \dot{q} disponibles. Pour cela, dans un premier temps, nous décrivons brièvement le modèle cinématique de la base mobile, avant de présenter le modèle cinématique de l'ensemble robotique portant la caméra.

Le modèle cinématique de la base mobile, muni d'une platine commandable en lacet, est donné classiquement par les relations suivantes :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \\ \dot{\vartheta} = \varpi \end{cases}$$

Le torseur cinématique, $\mathcal{T}_{M/\mathcal{R}_0} = (V_{M/\mathcal{R}_0}^T, \Omega_{\mathcal{R}_M/\mathcal{R}_0}^T)^T$ de la base mobile par rapport au repère de la scène \mathcal{R}_0 , s'exprime de la manière suivante dans le repère de la scène \mathcal{R}_0 :

$$\mathcal{T}_{M/\mathcal{R}_0}^{\mathcal{R}_0} = \begin{pmatrix} V_{M/\mathcal{R}_0}^{\mathcal{R}_0} \\ \Omega_{M/\mathcal{R}_0}^{\mathcal{R}_0} \end{pmatrix} = \begin{pmatrix} (v \cos \theta \ v \sin \theta \ 0)^T \\ (0 \ 0 \ \omega)^T \end{pmatrix} \quad (\text{I.4})$$

La détermination du mouvement $\mathcal{T}_{C/\mathcal{R}_0}$ de la caméra par rapport au repère de la scène \mathcal{R}_0 se déduit du résultat classique de mécanique :

$$\mathcal{T}_{C/\mathcal{R}_0}^{\mathcal{R}_C} = \begin{pmatrix} V_{C/\mathcal{R}_0}^{\mathcal{R}_C} \\ \Omega_{\mathcal{R}_C/\mathcal{R}_0}^{\mathcal{R}_C} \end{pmatrix} = \begin{pmatrix} V_{C/\mathcal{R}_M}^{\mathcal{R}_C} + V_{M/\mathcal{R}_0}^{\mathcal{R}_C} + \Omega_{\mathcal{R}_M/\mathcal{R}_0}^{\mathcal{R}_C} \wedge \overrightarrow{MC}^{\mathcal{R}_C} \\ \Omega_{\mathcal{R}_C/\mathcal{R}_M}^{\mathcal{R}_C} + \Omega_{\mathcal{R}_M/\mathcal{R}_0}^{\mathcal{R}_C} \end{pmatrix} \quad (\text{I.5})$$

Tous calculs faits [CADENAT, 1999] [FOLIO, 2003], le torseur cinématique de la caméra par rapport au repère de la scène \mathcal{R}_0 exprimé dans le repère \mathcal{R}_C , s'écrit comme suit :

$$\mathcal{T}_{C/\mathcal{R}_0}^{\mathcal{R}_C} = \begin{pmatrix} V_{\vec{x}_C} \\ V_{\vec{y}_C} \\ V_{\vec{z}_C} \\ \Omega_{\vec{x}_C} \\ \Omega_{\vec{y}_C} \\ \Omega_{\vec{z}_C} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ -\sin \vartheta & C_x + D_x \cos \vartheta & C_x \\ \cos \vartheta & -C_y + D_x \sin \vartheta & -C_y \\ 0 & -1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} = \mathbf{J}(q)\dot{q} \quad (\text{I.6})$$

où $\mathbf{J}(q)$ représente la jacobienne qui lie le mouvement de la caméra aux vitesses de la base mobile et de la platine. Notons que cette jacobienne comporte trois lignes de zéro,

correspondant aux mouvements non admissibles de la caméra. En effet, celle-ci ne peut se déplacer en translation sur l'axe \vec{x}_C , ni pivoter autour des axes \vec{y}_C et \vec{z}_C . Ces résultats sont donc conformes à la géométrie du robot. Cela nous permet alors de réduire le torseur cinématique \mathcal{T}_r de la caméra en ne considérant que les degrés de liberté commandables de cette dernière :

$$\mathcal{T}_r = \begin{pmatrix} V_{\vec{y}_C} \\ V_{\vec{z}_C} \\ \Omega_{\vec{x}_C} \end{pmatrix} = \begin{pmatrix} -\sin \vartheta & C_x + D_x \cos \vartheta & C_x \\ \cos \vartheta & -C_y + D_x \sin \vartheta & -C_y \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} = \mathbf{J}_r(q)\dot{q} \quad (\text{I.7})$$

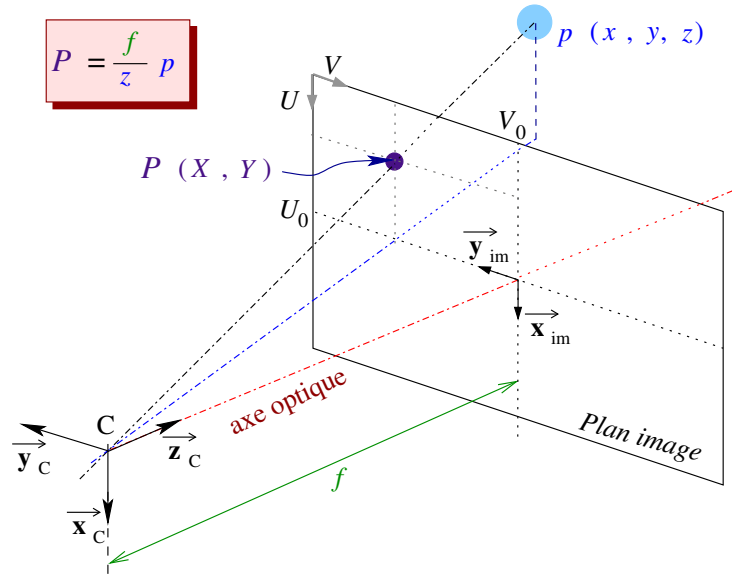
où $\mathbf{J}_r(q)$ désigne la jacobienne réduite de la caméra. En outre, comme son déterminant est non nul, $\det(\mathbf{J}_r) = D_x \neq 0$, la matrice \mathbf{J}_r est par conséquent régulière et inversible. La contrainte de non-holonomie n'intervient donc plus dans le mouvement de la caméra.

I.1.4.3 Modèle de la caméra

Notre robot est muni d'une caméra SONY DFW-VL 500 [DFWVL500] (voir figure I.6.a), numérique couleur supportant la norme IEEE1394. Cette caméra possède des pixels carrés permettant de ne pas modifier les proportions des objets perçus. Elle délivre des images au format VGA non compressé. Elle propose différentes résolutions (160×120 , 320×240 ou 640×480) et différents modes de compression pour l'acquisition des images vidéo (YUV444, YUV422 ou YUV411).



I.6.a – Caméra SONY DFW-VL 500



I.6.b – Modèle sténopé de la caméra

FIG. I.6 – La caméra

Nous considérons dans l'ensemble de notre étude le modèle de caméra dit *sténopé* ou "trou d'épingle" (*pinhole camera model*), présenté sur la figure I.6.b. Ce modèle est le plus couramment utilisé en vision par ordinateur, car il permet de modéliser finement la plupart des capteurs projectifs et de simplifier les équations mises en jeu [HORAUD et MONGA, 1995]. D'autant plus acceptable que l'on utilise des focales de faible dimension, le modèle sténopé prend pour hypothèse que tous les rayons passent par un seul point : le *centre optique* C . De ce fait, les points sont projetés sur le plan image par une projection

perspective. Ainsi, un point p de la scène, de coordonnées $(x \ y \ z)^T$, exprimé dans le repère caméra \mathcal{R}_C , est projeté sur le plan image en un point P , de coordonnées métriques $(X, Y)^T$, selon la relation suivante :

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \frac{f}{z} & 0 & 0 \\ 0 & \frac{f}{z} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

où f représente la distance focale de la caméra (cf. figure I.6.b). En introduisant la matrice des paramètres intrinsèques, nous pouvons relier les coordonnées métrique $(X, Y)^T$ de P à leurs coordonnées pixel (U, V) selon la relation suivante :

$$\begin{pmatrix} U \\ V \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & 0 & U_0 \\ 0 & -\alpha_v & V_0 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Matrice des paramètres intrinsèques}} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (\text{I.8})$$

où U_0 et V_0 correspondent aux coordonnées de la projection orthogonale du centre optique C dans le plan image (cf. figure I.6.b), et α_u et α_v représentent respectivement la taille des pixels selon les lignes et les colonnes. Les paramètres $(U_0, V_0, \alpha_u, \alpha_v)$, appelés *paramètres intrinsèques* de la caméra, sont obtenus après calibrage de cette dernière [ZHANG, 2000], et ne dépendent que du capteur optique proprement dit.

Sur notre plate-forme robotique, la caméra SONY DFW-VL 500 est configurée pour fournir des images vidéo compressées au format YUV422 à la résolution 640×480 . Nous avons établi les valeurs de la focale f et du centre optique au moyen de la **Camera Calibration Toolbox for Matlab** [BOUGUET, 2007]. Les différents paramètres intrinsèques de notre caméra sont réunis dans le tableau I.2. En particulier, la valeur de la distance focale étant faible, le modèle sténopé apparaît bien adapté pour modéliser notre capteur.

Centre optique	$U_0 = 253.287 \text{ pix}$ $V_0 = 312.018 \text{ pix}$	Taille des pixels	$1 \text{ pix} = 7.4 \mu\text{m}$
Focale	$f = 5.685 \text{ mm}$	I.2.b – Données constructeur	

I.2.a – Données de calibrage

$$\hookrightarrow \alpha_u = \alpha_v = \frac{f}{\text{pix}}$$

TAB. I.2 – Paramètres intrinsèques

Ainsi grâce à la relation (I.8), il est possible de projeter les points exprimés dans le repère de la caméra \mathcal{R}_C sur le plan image. Si les coordonnées de ces points sont données dans le repère de la scène \mathcal{R}_0 , il faut de plus déterminer la matrice des *paramètres extrinsèques*. Celle-ci décrit la situation de la caméra par rapport au repère du robot \mathcal{R}_M , et peut être obtenue par un calibrage dit “bras/oeil” (*hand-eye*) [TSAI et LENZ, 1989] [HORAUD et DORNAIKA, 1995]. Ainsi, connaissant la situation du robot dans l'environnement, nous pouvons projeter les points 3D de la scène dans le repère de la caméra \mathcal{R}_C , et donc sur le plan image.

A présent, nous avons établi le lien entre la position des points de la scène exprimés dans le repère caméra et les mesures effectuées dans l'image. Il reste à présent à modéliser l'interaction entre la caméra et l'environnement.

I.1.4.4 Interaction caméra/environnement

Comme nous l'avons préalablement énoncé, dans le cadre de l'asservissement visuel 2D, le choix des informations visuelles et l'obtention de la relation caractérisant leur variation sont deux points fondamentaux de cette approche. Les informations visuelles $\mathbf{s}(q, t)$ caractérisent les mesures effectuées au moyen de la caméra, en fonction de la configuration q du robot, et du temps t pris en tant que paramètre indépendant. On supposera dans l'ensemble de nos travaux que la valeur d'une information visuelle $\mathbf{s}(q, t)$ ne dépend que de la situation de la caméra par rapport à une cible observée. Ainsi, nous considérons que seul le mouvement de la caméra, et éventuellement celui de la cible, sont susceptibles de faire varier la valeur du signal sensoriel³. En différentiant $\mathbf{s}(q, t)$ par rapport au temps, il est possible d'exprimer la variation des informations visuelles en fonction des mouvements de la caméra [CHAUMETTE, 1990] :

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

où l'on reconnaît :

- $\frac{\partial \mathbf{s}}{\partial \mathbf{r}}$: la *matrice d'interaction* ou jacobienne de l'image, notée $\mathcal{L}(\mathbf{s}, \mathbf{z})$, que nous détaillerons ci-après.
- $\mathbf{J}(q) = \frac{\partial \mathbf{r}}{\partial q}$: le jacobien du robot, qui ne dépend que de la géométrie du robot et de sa configuration. Il est ici défini entièrement par la relation (I.6).
- $\dot{q} = \frac{dq}{dt}$: la commande en vitesse de notre système robotique (I.3).
- $\frac{\partial \mathbf{s}}{\partial t}$: terme qui est dû au mouvement propre de l'objet par rapport à la caméra. Toutefois, dans l'ensemble de nos travaux, nous considérons un amer visuel **fixe**, impliquant donc : $\frac{\partial \mathbf{s}}{\partial t} = 0$.

Ainsi, par le biais du torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}^{\mathcal{R}_C}$ précédemment défini (I.6), nous pouvons relier la variation des informations sensorielles au mouvement réalisé par la caméra, soit :

$$\dot{\mathbf{s}}(q, t) = \mathcal{L}(\mathbf{s}, \mathbf{z}) \mathcal{T}_{C/\mathcal{R}_0}^{\mathcal{R}_C} \quad (\text{I.9})$$

On retrouve ici la notion d'action réflexe où les stimuli du capteur, $\dot{\mathbf{s}}$, sont aussitôt retranscrits par une action, $\mathcal{T}_{C/\mathcal{R}_0}^{\mathcal{R}_C}$.

La matrice d'interaction

La matrice d'interaction $\mathcal{L}(\mathbf{s}, \mathbf{z})$ permet de relier le mouvement de la caméra aux informations visuelles. Cette matrice dépend non seulement de la nature des informations visuelles choisies, mais aussi de la situation de la caméra par rapport à l'objet observé, ou, plus particulièrement, de la profondeur \mathbf{z} , c'est-à-dire la distance entre la caméra et les informations sensorielles $\mathbf{s}(q, t)$. Différentes méthodes existent pour caractériser la matrice d'interaction. Ainsi, certains auteurs [HOSODA et ASADA, 1994] [JÄGERSAND *et al.*, 1997] utilisent des méthodes d'apprentissage permettant une estimation en ligne pour obtenir \mathcal{L} . Dans les deux cas, les auteurs traitent des informations visuelles de type point et c'est directement le produit des matrices ($\mathcal{L}\mathbf{J}$) qui est estimé. Toutefois, avec ces approches il n'est pas possible de démontrer la stabilité des lois de commande associées [CHAUMETTE, 1998B]. Une autre méthode consiste à formuler une expression analytique, à partir du modèle géométrique de l'objet considéré. Ainsi, F. CHAUMETTE propose une méthode générale

3. Nous ne considérerons pas les signaux basés sur des mesures photométriques dans l'image.

de calcul analytique de la matrice d'interaction pour différentes primitives géométriques simples, telles que des points, des cercles, des sphères, des ellipses, etc. [CHAUMETTE, 1990] ou des moments de l'image [CHAUMETTE, 2004].

Remarque I.1: *Le calcul de la matrice d'interaction $\mathcal{L}(\mathbf{s}, \mathbf{z})$ nécessite généralement une mesure ou un modèle de la profondeur⁴ \mathbf{z} . Toutefois, cette information n'étant pas toujours disponible, il est nécessaire, dans ce cas, d'utiliser un modèle ou une approximation de la matrice d'interaction, notée $\hat{\mathcal{L}}$.*

Exemple de matrice d'interaction : le cas du point

Par exemple⁵, dans le cas d'un point p de coordonnées $(x, y, z)^T$ dans \mathcal{R}_C qui se projette par projection perspective sur le plan image en un point P de coordonnées $(X, Y)^T$ (cf. paragraphe I.1.4.3) la matrice d'interaction $\mathcal{L}(P, z)$ se déduit des équations du flot optique, et s'écrit comme suit [CHAUMETTE, 1990] :

$$\mathcal{L}(P, z) = \begin{pmatrix} \mathcal{L}(X, z) \\ \mathcal{L}(Y, z) \end{pmatrix} = \begin{pmatrix} -\frac{f}{z} & 0 & \frac{X}{z} & \frac{XY}{f} & -\left(f + \frac{X^2}{f}\right) & Y \\ 0 & -\frac{f}{z} & \frac{Y}{z} & \left(f + \frac{Y^2}{f}\right) & -\frac{XY}{f} & -X \end{pmatrix} \quad (\text{I.10})$$

où f désigne la focale de la caméra (voir table I.2.a). Cette expression de la matrice d'interaction suppose que le torseur cinématique est de dimension (6×1) , et entièrement défini par la relation (I.6). Or, nous avons considéré avec l'équation (I.7) un torseur cinématique réduit \mathcal{T}_r de dimension (3×1) ne contenant que les degrés de liberté réellement commandables. De ce fait, il est nécessaire de rendre cette matrice compatible en ne conservant que les colonnes correspondant à ces degrés de liberté. Nous obtenons alors la matrice d'interaction réduite \mathcal{L}_r suivante :

$$\mathcal{L}_r = \begin{pmatrix} \mathcal{L}_X \\ \mathcal{L}_Y \end{pmatrix} = \begin{pmatrix} 0 & \frac{X}{z} & \frac{XY}{f} \\ -\frac{f}{z} & \frac{Y}{z} & \left(f + \frac{Y^2}{f}\right) \end{pmatrix} \quad (\text{I.11})$$

Nous pouvons généraliser ce résultat dans le cas d'un amer visuel constitué de $k > 1$ points. Le vecteur des informations sensorielles $\mathbf{s}(q, t)$ est alors défini par : $\mathbf{s} = (X_1, Y_1, \dots, X_j, Y_j, \dots, X_k, Y_k)$, où les couples (X_j, Y_j) représentent les coordonnées métriques du $j^{\text{ème}}$ point P_j de l'amer projeté dans le plan image de la caméra. La matrice d'interaction résultante est alors donnée par :

$$\mathcal{L}_s = (\mathcal{L}_{P_1} \ \dots \ \mathcal{L}_{P_k})^T = (\mathcal{L}_{X_1}^T \ \mathcal{L}_{Y_1}^T \ \dots \ \mathcal{L}_{X_k}^T \ \mathcal{L}_{Y_k}^T)^T$$

où chaque ligne \mathcal{L}_{P_j} correspond à la matrice d'interaction d'un point P_j définie, dans notre cas, par la relation (I.11).

Remarque I.2: *Différents types de primitives peuvent être construits sur la base de points : centres de gravité, segments, polygones, cercles, etc. [CHAUMETTE, 1990].*

Nous allons à présent exploiter ces résultats pour synthétiser une loi de commande référencée vision 2D.

4. Les éléments sur l'obtention de la profondeur \mathbf{z} sont détaillés au chapitre III dans le paragraphe III.2.1.

5. Le lecteur pourra trouver l'expression de la matrice d'interaction pour d'autres types de primitives dans chaumette90.

I.1.5 Synthèse d'un asservissement visuel 2D

Notre objectif est de synthétiser une loi de commande référencée vision, permettant de piloter un robot mobile de manière à atteindre un but défini a priori dans l'image. Rappelons, que le principe de l'asservissement visuel 2D (*cf.* paragraphe I.1.3) consiste à contrôler le mouvement de la caméra de manière à annuler l'erreur entre les informations visuelles courantes $\mathbf{s}(q, t)$ et le motif désiré \mathbf{s}^* . Pour cela, nous nous basons sur le formalisme des fonctions de tâche. À cet effet, nous présentons brièvement ce formalisme que nous appliquerons ensuite pour synthétiser une loi de commande référencée vision.

I.1.5.1 Formalisme des fonctions de tâche

Le formalisme des fonctions de tâche⁶ a été introduit par SAMSON *et al.* [1991] pour définir un cadre unificateur englobant la plupart des schémas de commande existants pour les robots manipulateurs holonomes. Il fut ensuite étendu au cas des robots mobiles non-holonomes [PISSARD-GIBOLLET, 1993]. Ainsi dans le cadre de ce formalisme, la tâche robotique sera définie par une fonction de sortie particulière : *la fonction de tâche*, notée $\mathbf{e}(q, t)$, qui dépend de la configuration q du robot, et du temps t pris en tant que paramètre indépendant. Plus précisément, on peut montrer que l'objectif peut s'exprimer comme la régulation à zéro, sur un certain horizon temporel $t \in [0, T]$, d'une fonction de tâche $\mathbf{e}(q, t)$ de classe \mathcal{C}^2 définie dans \mathbb{R}^n ⁷. Les fonctions de tâche peuvent s'exprimer en termes de données proprioceptives ou extéroceptives selon l'application considérée et les capteurs disponibles. Par exemple, des fonctions de tâche classiques sont :

- $\mathbf{e}(q, t) = q(t) - q^*$, où q^* est la configuration de référence à atteindre dans l'espace des configurations du robot ;
- $\mathbf{e}(q, t) = \mathbf{r}(q, t) - \mathbf{r}^*$, où \mathbf{r}^* est la situation de référence à atteindre dans un repère de base ;
- $\mathbf{e}(q, t) = \mathbf{s}(q, t) - \mathbf{s}^*$, où \mathbf{s}^* est un motif désiré à atteindre dans l'espace du capteur.

Cependant, comme cela est précisé dans [SAMSON *et al.*, 1991], pour que ce problème de régulation de $\mathbf{e}(q, t)$ soit bien posé, la fonction de tâche doit posséder certaines propriétés, notamment l'unicité de la trajectoire $q_r(t)$ solution de l'équation : $\mathbf{e}(q, t) = 0, \forall t \in [0, T]$. Or, la fonction de tâche étant non-linéaire en q , la résolution de cette équation peut conduire à la définition de plusieurs trajectoires solutions distinctes. Pour éviter cela, une condition initiale q_0 est introduite, telle que : $q_r(0) = q_0$ vérifie : $\mathbf{e}(q_0, 0) = 0$. Notons qu'une trajectoire vérifiant cette propriété est dite *trajectoire idéale*. Cependant, il se peut qu'aucune solution n'existe, excepté la condition initiale, ou, au contraire, qu'il y en ait une infinité. Dans ce dernier cas on parle alors de *tâche redondante* que nous présenterons un peu plus loin. Outre l'unicité et l'existence de la trajectoire, il est nécessaire d'une part que la régulation de la fonction de tâche $\mathbf{e}(q, t)$ à zéro impose la convergence de la trajectoire du robot vers la trajectoire solution désirée, et d'autre part qu'une petite variation de $\mathbf{e}(q, t)$ n'induisse pas une grande variation de la configuration q du robot. Lorsque toutes les conditions requises sont satisfaites, la fonction de tâche est dite *admissible*, ce qui permet alors la synthèse de lois de commande efficaces. La notion de ρ -*admissibilité* offre alors un cadre rigoureux, réunissant les conditions *suffisantes* que doivent vérifier les fonctions de tâche pour que le problème de commande soit bien posé [SAMSON *et al.*, 1991]. Nous rappelons brièvement ci-après cette notion.

6. Nous ne saurions trop recommander aux lecteurs intéressés la lecture de [SAMSON *et al.*, 1991].

7. n représente ici le nombre de degrés de liberté, correspondant à la dimension du vecteur q

Notion de ρ -admissibilité

La ρ -admissibilité permet de prouver l'existence d'une application bijective de classe \mathcal{C}^1 dont la réciproque est aussi de classe \mathcal{C}^1 , c'est-à-dire l'existence d'un *difféomorphisme* entre l'ensemble des couples (q, t) et (e, t) . Ce difféomorphisme permet alors d'établir un lien régulier entre l'espace d'état et l'espace capteur dans lequel s'exprime la commande. La ρ -admissibilité est définie plus formellement de la façon suivante [SAMSON *et al.*, 1991] :

Définition I.1 Notion de ρ -admissibilité

$e(q, t)$ est une fonction de tâche ρ -admissible sur l'ensemble $C_{\rho, T}$ des couples (q, t) défini sur l'intervalle de temps $[0; T]$ si et seulement si la fonction $F(q, t) = (e(q, t), t)$ admet un difféomorphisme de classe \mathcal{C}^k de $C_{\rho, T}$ sur la boule fermée B_ρ centrée en 0 et de rayon ρ .

Le lecteur intéressé pourra trouver une définition plus complète de la ρ -admissibilité dans [SAMSON *et al.*, 1991]. Comme le souligne le théorème I.1 suivant proposé par [SAMSON *et al.*, 1991], la ρ -admissibilité d'une fonction de tâche est principalement liée à l'existence du difféomorphisme mentionné plus haut.

Théorème I.1 Condition de ρ -admissibilité

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

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

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

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

Formalisme des tâches redondantes

Une tâche redondante est une tâche qui ne contraint pas tous les degrés de liberté du robot, par conséquent la jacobienne de la fonction de tâche n'est pas carrée. Elle n'est donc pas ρ -admissible, et l'équation $e(q) = 0$ admet une infinité de trajectoires solutions $q_r(t)$. Ainsi, une tâche redondante est *sous-contrainte* : il n'existe pas de difféomorphisme entre l'espace d'état et l'espace capteur. L'ensemble des trajectoires solutions $q_r(t)$ forment alors un ensemble dense dans l'espace d'état : on peut ainsi passer d'une trajectoire à l'autre de façon continue [SAMSON *et al.*, 1991]. L'idée du formalisme des tâches redondantes est d'exploiter la redondance du robot pour réaliser *au mieux* une tâche secondaire en utilisant les degrés de liberté laissés disponibles par la tâche principale e_1 . La tâche secondaire étant

modélisée sous la forme d'une fonction de coût h , le problème de commande consiste à minimiser $h(q)$ sous la contrainte que la tâche prioritaire e_1 soit parfaitement exécutée (*i.e.* $e_1 = 0$). Ce problème d'optimisation est équivalent à réguler à zéro la fonction de tâche globale $e(q)$ suivante :

$$e(q) = W^+ e_1 + \beta (\mathbb{I} - W^+ W) g \quad (\text{I.12})$$

où $g = \frac{\partial h}{\partial q}$ représente le gradient de la fonction de coût $h(q)$, $\beta > 0$ est un réel positif, et \mathbb{I} définit la matrice identité de dimension n . $W^+ = W^T(WW^T)^{-1}$ est la pseudo-inverse à droite de W . Cette matrice W doit être choisie telle que :

$$\ker(W) = \ker\left(\frac{\partial e_1}{\partial q}\right) \quad \text{et} \quad \left(\frac{\partial e_1}{\partial q} W^T\right) > 0 \quad (\text{I.13})$$

En effet, sous ces conditions, SAMSON *et al.* [SAMSON *et al.*, 1991] montrent que la jacobienne de la tâche $J_e = \frac{\partial e}{\partial q}$ est elle-même définie positive au voisinage de $q_r(t)$, assurant la ρ -admissibilité de la fonction de tâche globale $e(q)$.

Remarque I.3 (choix de W): Si la jacobienne $J_{e_1} = \frac{\partial e_1}{\partial q}$ de la fonction de tâche redondante e_1 est de rang plein, un choix simple permettant de satisfaire les conditions (I.13) est de prendre : $W = J_{e_1} = \frac{\partial e_1}{\partial q}$.

Synthèse de la commande

Le formalisme des fonctions de tâches consiste à spécifier l'objectif à réaliser sous la forme d'une régulation à zéro d'une fonction particulière $e(q)$. Ainsi, afin d'assurer la convergence de celle-ci vers zéro nous imposons à la fonction de tâche de décroître de manière exponentielle. Dans la mesure où notre système robotique est commandable en vitesse, par l'intermédiaire de \dot{q} (I.3), dans le cadre de nos travaux, nous considérons uniquement des schémas de commande cinématiques. Le comportement désiré en boucle fermée peut donc être décrit par :

$$\dot{e}(q) = J_e \dot{q} = -\lambda e(q) \quad \forall \lambda > 0 \quad (\text{I.14})$$

où λ est un gain scalaire positif ou une matrice définie positive, permettant de fixer la vitesse de décroissance exponentielle de $e(q)$. La matrice $J_e = \frac{\partial e}{\partial q}$ représente la jacobienne de la fonction de tâche. Si $e(q)$ est ρ -admissible, c'est-à-dire que J_e est inversible, cela suffit pour s'assurer que le problème de commande est bien posé, et assure la détermination de la commande \dot{q} de manière unique. Toutefois, lorsque la fonction de tâche $e(q)$ est définie par (I.12), le calcul de J_e s'avère en général difficile, et la synthèse de la commande requiert une estimation de cette matrice, notée \hat{J}_e [CHAUMETTE, 1990]. La loi de commande (I.14) s'écrit alors : $\dot{q} = -\hat{J}_e^{-1} \lambda e(q)$. La dynamique de la fonction de tâche $e(q)$ devient alors en réalité : $\dot{e}(q) = -J_e \hat{J}_e^{-1} \lambda e(q)$, et la stabilité de la loi de commande nécessite que le produit de matrices $(J_e \hat{J}_e^{-1})$ soit défini positif. Or, si les conditions (I.13) sont satisfaites (voir aussi la remarque I.3), la jacobienne J_e est définie positive autour de la trajectoire solution $q_r(t)$. Dans ce cas, une solution satisfaisant la condition précédente $(J_e \hat{J}_e^{-1}) > 0$ consiste à choisir \hat{J}_e égale à la matrice identité \mathbb{I} , et la loi de commande ci-dessus se simplifie alors comme suit [ESPIAU *et al.*, 1992] :

$$\dot{q} = -\lambda e(q) \quad (\text{I.15})$$

I.1.5.2 Application du formalisme des fonctions de tâche à la synthèse d'un asservissement visuel

Notre but est de synthétiser une loi de commande permettant la réalisation d'une tâche de navigation référencée vision. Pour ce faire, nous exploitons le formalisme des fonctions de tâches que nous venons brièvement de présenter. Classiquement, un asservissement visuel permettant d'amener $\mathbf{s}(q)$ vers le motif désiré \mathbf{s}^* est défini par la fonction de tâche \mathbf{e}_{AV} suivante [CHAUMETTE, 1990] :

$$\mathbf{e}_{AV} = \mathbf{C}(\mathbf{s}(q) - \mathbf{s}^*) \quad (\text{I.16})$$

où \mathbf{C} est une matrice dite de combinaison. Le fait d'introduire cette matrice \mathbf{C} peut perturber la convergence des indices visuels $\mathbf{s}(q)$ vers leur consigne \mathbf{s}^* , en introduisant des minima locaux. En effet, la régulation de \mathbf{e}_{AV} n'assure que la convergence de $\mathbf{C}\mathbf{s}$ vers $\mathbf{C}\mathbf{s}^*$, il n'y a donc pas équivalence entre la régulation de \mathbf{e}_{AV} et celle de $(\mathbf{s} - \mathbf{s}^*)$. Néanmoins, la matrice de combinaison permet de pouvoir prendre en compte un nombre d'informations capteur supérieur au nombre de degrés de liberté, n , de la caméra contraints par la tâche référencée vision. En effet, comme nous l'avons vu au paragraphe I.1.5.1 précédent, la satisfaction de la condition ρ -admissibilité est associée à la régularité de la jacobienne de la fonction de tâche \mathbf{e}_{AV} . Cette condition est satisfaite en choisissant la matrice de combinaison \mathbf{C} de rang plein n , et telle que le produit des matrices $(\mathbf{C}\mathbf{L}_r\mathbf{J}_r)$ soit aussi de rang plein n . Dans la mesure où la matrice jacobienne du robot \mathbf{J}_r est une matrice régulière, la stabilité et la convergence de la fonction de tâche sont assurées par la condition suffisante suivante [CHAUMETTE, 1998B] :

$$\mathbf{C}\mathcal{L}(\mathbf{s}, z) > 0 \quad (\text{I.17})$$

Un choix judicieux⁸, et communément effectué, consiste à fixer la matrice de combinaison \mathbf{C} égale à un modèle ou à une approximation de la pseudo-inverse à gauche de la matrice d'interaction $\mathcal{L}(\mathbf{s}, z)$, notée : $\hat{\mathcal{L}}^+ \simeq (\mathcal{L}^T\mathcal{L})^{-1}\mathcal{L}^T$ (voir aussi la remarque I.1).

Enfin, pour assurer la convergence de \mathbf{e}_{AV} vers zéro nous imposons à la fonction de tâche de décroître de manière exponentielle vers zéro, comme nous l'avons précédemment introduit. Ainsi, la commande référencée vision se déduit de (I.14) et s'écrit :

$$\mathbf{J}_{AV}\dot{q}_{AV} = -\lambda_{AV}\mathbf{e}_{AV} \quad \forall \lambda_{AV} > 0 \quad (\text{I.18})$$

où λ_{AV} permet de fixer la vitesse de décroissance exponentielle, et $\mathbf{J}_{AV} = \frac{\partial \mathbf{e}_{AV}}{\partial q}$ représente la jacobienne de la fonction de tâche \mathbf{e}_{AV} . Ainsi, la régularité de la jacobienne \mathbf{J}_{AV} assure la ρ -admissibilité de la fonction de tâche référencée vision \mathbf{e}_{AV} , et garantit que le problème de commande est bien posé.

I.1.5.3 Un exemple applicatif : réalisation d'une tâche de positionnement référencée vision

Notre objectif ici est de réaliser une tâche de positionnement par rapport à un amer visuel donné, à l'aide de la commande (I.18) référencée vision, présentée ci-dessus.

8. Le lecteur intéressé pourra trouver davantage de détails sur le choix de \mathbf{C} dans [CHAUMETTE, 1998A] et [CHAUMETTE et HUTCHINSON, 2006].

Dans le cadre de notre étude, nous considérons un amer visuel **fixe** caractérisé par un ensemble de $k \geq 4$ points (voir la remarque I.4) extrait par un processus de traitement d'image. Le vecteur des informations sensorielles $\mathbf{s}(q)$ est alors défini par : $\mathbf{s} = (X_1, Y_1, \dots, X_j, Y_j, \dots, X_k, Y_k)$, où les couples (X_j, Y_j) représentent les coordonnées métriques du $j^{\text{ème}}$ point de l'amer projeté en P_j dans le plan image de la caméra. Comme nous l'avons vu précédemment, la matrice d'interaction résultante est donnée par :

$$\mathcal{L}_r = \left(\mathcal{L}_{X_1}^T \quad \mathcal{L}_{Y_1}^T \quad \dots \quad \mathcal{L}_{X_k}^T \quad \mathcal{L}_{Y_k}^T \right)^T \quad (\text{I.19})$$

où chaque ligne \mathcal{L}_{X_j} et \mathcal{L}_{Y_j} est définie pour un point P_j par la relation (I.11).

Remarque I.4 (type d'amer considéré): *Il est bien connu que si le motif visuel est constitué d'au plus trois points, des singularités et des problèmes de minima locaux peuvent se produire c'est-à-dire que la commande est nulle pour $\mathbf{s} \neq \mathbf{s}^*$. En effet, la matrice d'interaction devient singulière si ces points sont colinéaires. Par ailleurs, la projection de trois points sur le plan image peut correspondre jusqu'à quatre situations de la caméra [HORAUD, 1987]. Ainsi, l'utilisation d'au moins quatre points permet de limiter ces problèmes.*

La commande référencée vision se déduit alors simplement à partir de la relation (I.18), et nécessite la détermination de la jacobienne \mathbf{J}_{AV} . Cette matrice se déduit à partir des relations (I.9) et (I.16) comme suit :

$$\mathbf{J}_{AV} = \frac{\partial \mathbf{e}_{AV}}{\partial q} = \mathcal{C} \mathcal{L}_r \mathbf{J}_r \quad (\text{I.20})$$

Dans la mesure où la profondeur z , nécessaire au calcul de la matrice d'interaction \mathcal{L}_r , n'est pas toujours disponible (voir également la remarque I.1), on considère couramment une estimation de \mathcal{L}_r , notée : $\widehat{\mathcal{L}}_c = \mathcal{L}_r(\mathbf{s}^*, z^*)$, où la matrice d'interaction est choisie constante et est évaluée à la situation désirée en \mathbf{s}^* . La matrice de combinaison, permettant de satisfaire la condition (I.17), est alors définie par : $\mathcal{C} = \widehat{\mathcal{L}}_c^+(\mathbf{s}^*, z^*)$. La jacobienne \mathbf{J}_{AV} (I.20) ci-dessus se simplifie de la façon suivante : $\widehat{\mathbf{J}}_{AV} = \widehat{\mathcal{L}}_c^+ \widehat{\mathcal{L}}_c \mathbf{J}_r = \mathbf{J}_r$, où \mathbf{J}_r désigne la jacobienne réduite du robot définie par la relation (I.7). La propriété de ρ -admissibilité ne dépend plus alors que de la structure mécanique du robot. Comme nous l'avons précisé à la section I.1.4.2, \mathbf{J}_r est une matrice régulière : la propriété de ρ -admissibilité est donc satisfaite. Toutefois, la jacobienne de la fonction de tâche \mathbf{e}_{AV} ainsi obtenue n'est qu'une estimée, notée $\widehat{\mathbf{J}}_{AV}$, la valeur exacte de cette matrice ne pouvant être déterminée que si la matrice d'interaction \mathcal{L}_r est parfaitement connue à chaque itération. Le correcteur permettant de réguler \mathbf{e}_{AV} à zéro s'écrit alors simplement : $\dot{q}_{AV} = -\mathbf{J}_r^{-1} \lambda \mathbf{e}_{AV}$, soit finalement :

$$\dot{q}_{AV} = -\mathbf{J}_r^{-1} \lambda \mathcal{C} (\mathbf{s} - \mathbf{s}^*) \quad (\text{I.21})$$

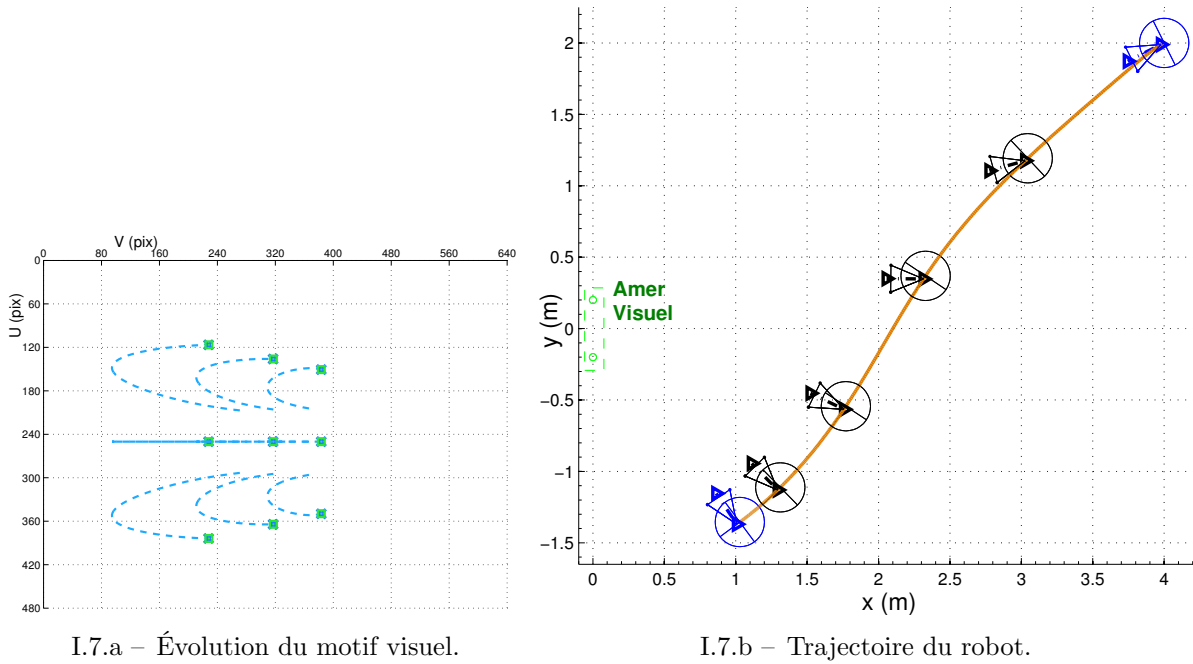
Remarque I.5: *Dans cette approche la décroissance exponentielle de la fonction de tâche \mathbf{e}_{AV} est perdue, puisque :*

$$\dot{\mathbf{e}}_{AV} = -\mathbf{J}_{AV} \lambda \widehat{\mathbf{J}}_{AV}^{-1} \mathbf{e}_{AV} = -\mathbf{J}_{AV} \lambda \mathbf{J}_r^{-1} \mathbf{e}_{AV}$$

Ce phénomène est d'autant plus accentué que le produit $(\mathbf{J}_{AV} \mathbf{J}_r^{-1})$ est différent de la matrice identité, c'est-à-dire dans notre cas lorsque le signal visuel \mathbf{s} se trouve loin du motif désiré \mathbf{s}^ . De plus, il est difficile de garantir la stabilité lorsque le robot se trouve loin de la trajectoire de référence. Cependant, l'expérience montre que le domaine de convergence de \mathbf{e}_{AV} est en général plus étendu.*

Simulation d'une tâche de navigation référencée vision

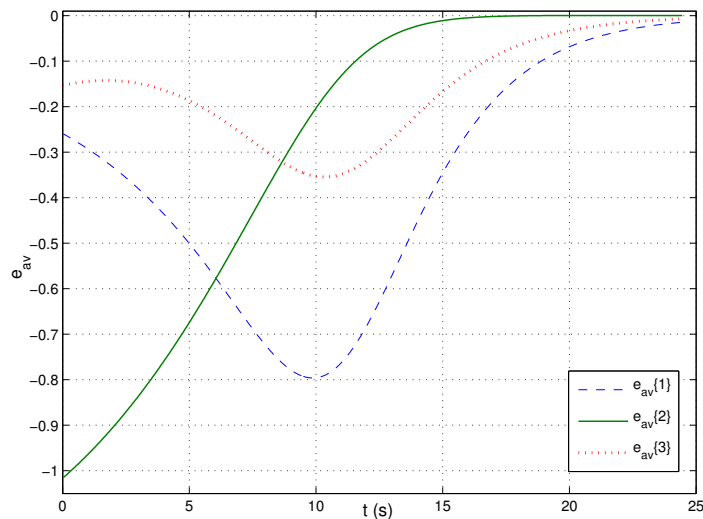
Afin d'illustrer l'asservissement visuel, nous présentons ci-après un exemple de simulation, où la tâche référencée vision choisie consiste à positionner la caméra relativement à un amers visuel caractérisé par neuf points (*cf.* remarque I.4). Notons que la période d'échantillonnage a été choisie égale à 150 ms, valeur correspondant à notre plate-forme expérimentale⁹. La trajectoire du robot et l'évolution correspondante des indices visuels sont représentées sur les figures I.7.a et I.7.b, et montrent que la tâche de navigation est parfaitement réalisée.



I.7.a – Évolution du motif visuel.

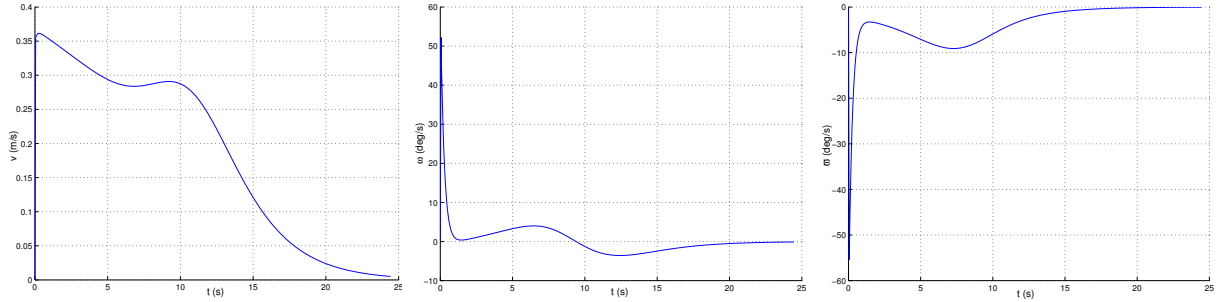
I.7.b – Trajectoire du robot.

FIG. I.7 – Simulation d'un asservissement visuel 2D.

FIG. I.8 – Évolution de la fonction de tâche e_{AV} .

9. Cette période d'échantillonnage correspond à la période de l'asservissement et du processus d'extraction des mesures perceptuelles (*cf.* chapitre IV).

La figure I.8 décrit la progression de la fonction tâche référencée vision e_{AV} au cours de la mission. Celle-ci est réglée à zéro, mais sa décroissance n'est pas exponentielle puisque nous avons implanté la commande (I.21) (*cf.* remarque I.5). Enfin la figure I.9 présente la commande référencée vision \dot{q}_{AV} appliquée au robot durant la tâche de navigation. Un choix judicieux du gain λ_{AV} , qui définit la vitesse de décroissance de la fonction de tâche e_{AV} , permet de s'assurer que la commande ne sature pas les actionneurs. Ainsi, dans notre exemple, en fixant $\lambda_{AV} = 0.35$ la mission se réalise parfaitement tout en garantissant l'admissibilité des vitesses envoyées au robot (*cf.* tableau I.1).



I.9.a – Vitesse linéaire v (m/s) I.9.b – Vitesse angulaire ω (deg/s) I.9.c – Vitesse de la platine ϖ

FIG. I.9 – La commande cinématique $\dot{q}_{AV} = (v, \omega, \varpi)$.

Expérimentation

Nous avons également cherché à expérimenter la loi de commande référencée vision sur le robot Super Scout de notre étude. Pour cela, nous utilisons ici un amer visuel constitué de quatre disques de couleur qui facilite et accélère le processus d'extraction des informations visuelles dans l'image.

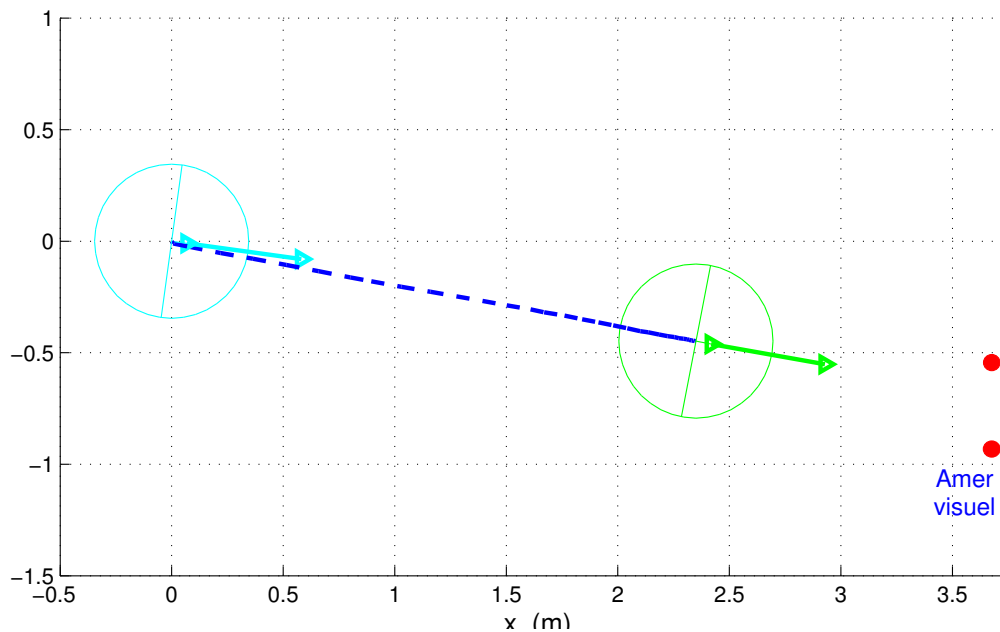
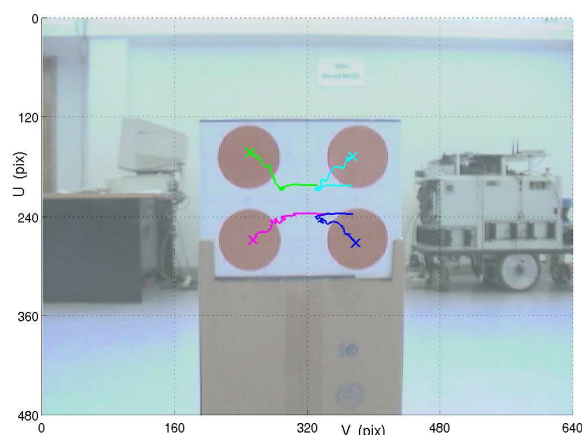
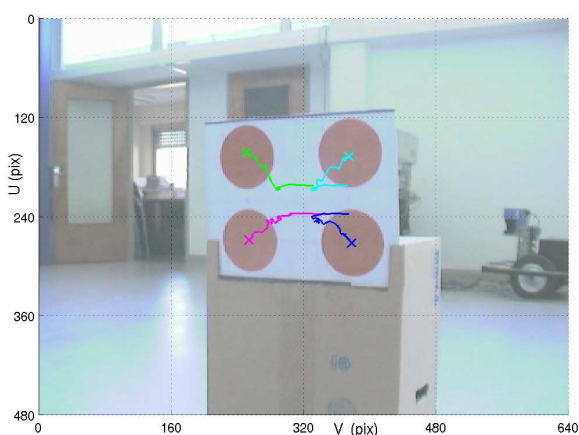


FIG. I.10 – Expérimentation d'un asservissement visuel 2D : trajectoire du robot.

La tâche de navigation choisie consiste à positionner la caméra face à une cible fixe. Nous définissons en premier lieu le motif de référence (*cf.* figure I.11.a) vers lequel la caméra embarquée doit converger. Ensuite, le robot est positionné à sa configuration initiale de sorte que la cible soit dans le champ de vision de la caméra. L'asservissement visuel est ensuite exécuté. La figure I.10 présente alors l'évolution de la position du robot obtenue à partir des mesures odométriques disponibles sur le Super Scout. L'évolution du véhicule est donc ici définie par rapport à un repère relatif fixé au début de la tâche de navigation. La figure I.11.b montre la vue de la caméra à la fin de la tâche. Le robot ne parvient pas ici à converger exactement vers le motif de référence \mathbf{s}^* du fait que, pour le gain statiques λ_{AV} utilisé, à la fin de la tâche, la consigne de commande devient trop faible pour être réalisable par le robot. Une solution pour remédier à cet inconvénient consisterait à utiliser un gain variable pour assurer une meilleure convergence de la tâche.



I.11.a – Image désirée.



I.11.b – Image à la fin de la tâche.

FIG. I.11 – Expérimentation d'un asservissement visuel 2D : évolution du motif visuel.

I.2 Navigation en environnement encombré

La commande référencée vision, présentée dans la section précédente, permet de réaliser des tâches de navigation grâce à l'utilisation des indices visuels dans la boucle de commande. Toutefois, cette approche suppose que rien ne vient entraver les déplacements du robot, ni gêner la caractérisation du signal visuel. Elle s'avère donc limitée lorsque l'environnement autour du robot est encombré d'obstacles. Ainsi, à lui seul, l'asservissement visuel ne permet pas de décrire des tâches robotiques élaborées et de tenir compte d'événements extérieurs. Afin de se rapprocher des situations réelles, il paraît nécessaire d'exploiter des données sensorielles de nature différente, dans le but de caractériser d'une part l'objectif de la mission, et, d'autre part, de gérer la présence éventuelle d'obstacles. Nous nous plaçons dans la continuité des travaux de CADENAT, V. [CADENAT, 1999] qui proposait des schémas de commande référencés multi-capteurs permettant à un robot mobile de réaliser des tâches de navigation référencées vision, dans un environnement encombré d'obstacles non occultants. Notre intention dans cette section est de rappeler l'ensemble de ces travaux sur lesquels nous avons bâti nos propres stratégies de commande. Nous décrirons donc d'abord les techniques d'évitement de collision les mieux adaptées au contexte de la commande référencée capteur, avant de montrer comment les coupler avec l'asservissement visuel pour exécuter des tâches référencée vision élaborées.

I.2.1 Évitement d'obstacles

Dans le domaine de l'évitement d'obstacle, la recherche se divise traditionnellement en deux grandes catégories : l'approche globale et l'approche locale.

- ▷ Dans les méthodes globales, la tâche robotique est définie au moyen d'un planificateur. Pour générer un chemin ou une trajectoire, le planificateur dispose d'une carte de l'environnement où la position des obstacles est supposée connue. Il est alors possible de produire un itinéraire reliant la situation initiale à la situation finale en évitant les collisions avec les obstacles connus. Néanmoins, cette approche requiert que le monde dans lequel évolue le robot soit parfaitement maîtrisé et statique. De plus, elle suppose généralement de pouvoir localiser le robot dans son environnement. Il faut alors intégrer les incertitudes de localisation de manière à ce que cela ne perturbe pas les mouvements du robot au voisinage des obstacles. Toutefois, il existe des extensions de l'approche globale visant à tenir compte de la variabilité de l'environnement. Ainsi, des solutions sont proposées par exemple dans [QUINLAN, 1994] où l'auteur propose l'algorithme de "bande élastique" (*elastic band*) qui permet, durant l'exécution de la tâche, de déformer la trajectoire initialement conçue, de manière à éviter les obstacles non prévus. Cette approche fut appliquée à des robots mobiles non-holonomes par KHATIB *et al.* [1997], puis adaptée à des systèmes plus complexes tels que des robots mobiles avec remorque par LAMIRAUX *et al.* [2004]. D'autres travaux exploitent l'approche de la fenêtre dynamique (*Dynamic Window Approach – DWA*) pour commander des robots omnidirectionnels (*free flying robot*) [FOX *et al.*, 1997]. Cette approche a ensuite été étendue aux robots non-holonomes dans [BROCK et KHATIB, 1999]. Toutes ces méthodes se basent cependant sur la modification d'un chemin ou d'une trajectoire préalablement établis. Par conséquent, elles requièrent la construction d'une carte pour générer l'itinéraire, ainsi que la localisation du robot par rapport à celui-ci.
- ▷ Les méthodes locales, que nous considérons dans notre étude, utilisent en temps réel les informations perceptuelles qui renseignent sur l'environnement proche du robot. Sur la base de ces données, l'objectif est alors de synthétiser une commande qui guide le robot vers son but tout en tenant compte de la présence d'obstacles. Les capteurs les plus couramment employés sont les lasers, et les capteurs à infrarouge ou ultrasons qui permettent de caractériser localement l'obstacle le plus proche du robot, c'est-à-dire le *risque de collision*. Toutefois, ne disposant pas d'un modèle des obstacles et de l'environnement, les méthodes locales souffrent généralement des problèmes de minima locaux.

Les stratégies de contournement d'obstacles, développées dans cet ouvrage, s'appuient sur les mesures proximétriques provenant des seize capteurs ultrasons disponibles sur notre robot, le Super Scout. Ainsi, lorsque le véhicule entre dans le voisinage d'un obstacle, délimité par l'enveloppe ξ_+ située à une distance d_+ de l'obstacle, les capteurs ultrasoniques, après traitement des données, fournissent (voir figure I.12) :

- la distance d_{coll} entre le centre M du robot et le point B le plus proche de l'obstacle ;
- l'orientation α entre la tangente \vec{T} à l'obstacle et la direction \vec{x}_M du véhicule.

Remarque I.6: *Nous supposons que la distance entre deux obstacles reste supérieure à $2d_+$, afin de permettre au robot de circuler efficacement entre différents obstacles. De cette manière nous pouvons considérer à tout instant uniquement l'obstacle le plus proche. De plus, nous admettons qu'il n'y a pas d'obstacle dans le voisinage de la cible qui empêcherait la finalisation de la mission.*

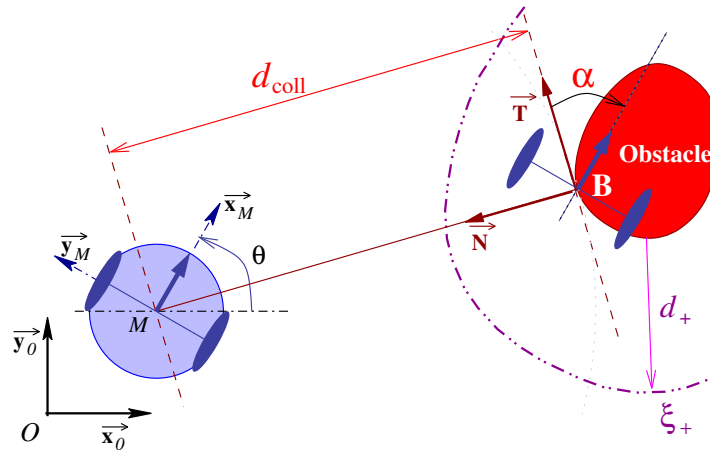


FIG. I.12 – Paramètres pour l'évitement d'obstacle.

Remarque I.7: Selon l'orientation de la tangente à l'obstacle il existe deux angles α . Nous définissons une tangente orientée unique \vec{T} , choisie lorsque le robot entre dans le voisinage de l'obstacle en fonction du sens de contournement, noté σ . Le choix du sens de contournement dépendra de la stratégie de commande adoptée.

Ainsi, nous nous proposons de présenter ici deux méthodes locales de contournement d'obstacles : la méthode des potentiels introduite par [KHATIB, 1980] et le formalisme du suivi de chemin proposé par [SAMSON, 1992]. Puis nous décrirons comment combiner ces approches à notre commande référencée vision introduite précédemment à la section I.1.5.2.

I.2.2 Méthode des champs de potentiels

I.2.2.1 Principe général de la méthode des potentiels

L'idée du formalisme des potentiels est de créer un champ de potentiel artificiel, qui va guider les déplacements du robot dans son environnement. Classiquement, le but à atteindre joue le rôle d'un potentiel **attractif**, *a contrario* les obstacles, ou plus généralement ce que le robot doit éviter, sont modélisés par des potentiels **répulsifs** (voir figure I.13). Le robot se comporte alors comme une particule chargée qui glisse le long du gradient de potentiels résultant de la somme des différents potentiels en cherchant à atteindre le minimum (*e.g.* le but). L'approche des potentiels reste bien adaptée à la problématique du contrôle du mouvement des robots en milieu encombré, dans la mesure où la force dérivant du potentiel permet de calculer une direction de déplacement et une accélération proportionnelle à son intensité (voir figure I.13.b). Initialement proposé par KHATIB, O. [KHATIB, 1980], il existe dans la littérature différents formalismes utilisant cette approche, comme la méthode des potentiels classiques [KHATIB, 1986], la méthode des potentiels généralisés [KROGH, 1984], la méthode des potentiels rotations [KHATIB, 1996], l'histogramme de champs de vecteurs [BORENSTEIN et KOREN, 1991], etc. Le problème essentiel est alors de minimiser l'influence des minima locaux, appelés dans ce cadre *puits de potentiels*. Les puits de potentiels apparaissent lorsque les fonctions de potentiels deviennent *antagonistes*. Par exemple la figure I.13.b présente un puits de potentiel derrière l'obstacle par rapport au but. Plus particulièrement, d'un point de vue expérimental, la commande générée oscille, ou devient trop faible pour pouvoir contrôler le robot. Il s'avère en général difficile de détecter efficacement les puits de potentiels. De ce fait, le

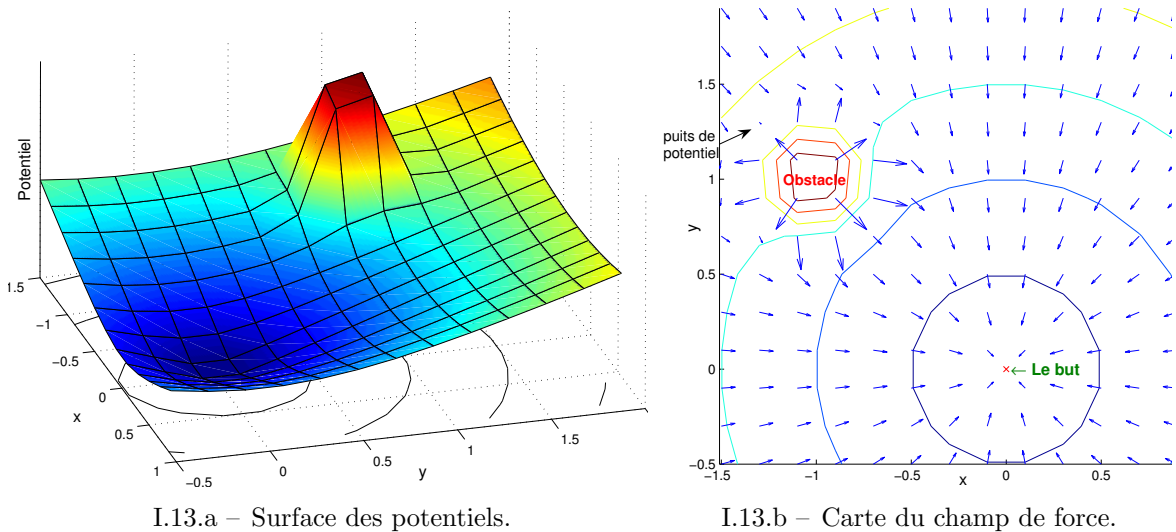


FIG. I.13 – Exemple de distribution de potentiel classique.

robot peut se trouver immobilisé dans une configuration, autre que la situation désirée, dont il ne peut pas sortir. Une première solution consiste à apporter une perturbation sur la configuration du robot lorsque celui-ci se trouve dans un puits de potentiel, pour tenter de l'en faire sortir. Afin de remédier à ces problèmes, nous nous proposons d'utiliser la *méthode des potentiels rotatifs* [CADENAT *et al.*, 1999B] qui s'inspire des méthodes dites de *champ de vortex* [MEDIO et ORIOLO, 1991]. Cette approche repose sur la séparation des effets attractif et répulsif des différentes forces.

I.2.2.2 Méthode des potentiels rotatifs

Dans les méthodes des potentiels classiques [KHATIB, 1980] [KHATIB, 1996] la force répulsive repousse simplement le robot au-delà d'une zone qui délimite le domaine de répulsion. La force correspondante tend alors vers zéro, et le robot s'immobilise en l'absence de toute force attractive. L'idée de la méthode des potentiels rotatifs est de générer un champ de potentiel autour de l'obstacle qui *impose* au robot de suivre une enveloppe particulière, dite enveloppe de sécurité. Ainsi, le potentiel répulsif doit être capable de maintenir la commande du robot au voisinage de l'obstacle afin de pouvoir le contourner, et cela même en l'absence de potentiel attractif. Pour cela, l'idée de base est d'une part d'assurer que la norme de la force répulsive soit non nulle sur l'enveloppe de sécurité, et d'autre part d'adapter son orientation en fonction de la distance à l'obstacle de telle sorte que le robot puisse contourner l'obstacle sans l'aide d'un potentiel attractif.

À cet effet, à l'aide de la figure I.14 qui présente les notations utilisées, nous définissons trois enveloppes autour de chaque obstacle :

1. la première, notée ξ_- , est située à une distance $d_- > 0$ et définit la région où l'obstacle est considéré comme réellement dangereux pour le robot ;
2. la seconde, notée ξ_0 , se trouve à une distance $d_0 > d_-$ et représente l'enveloppe de sécurité que le robot doit emprunter pour contourner l'obstacle ;
3. la troisième, notée ξ_+ , se trouve à une distance $d_+ > d_0$ et caractérise la zone à partir de laquelle l'obstacle est pris en considération, c'est-à-dire la limite de la zone d'influence de l'obstacle.

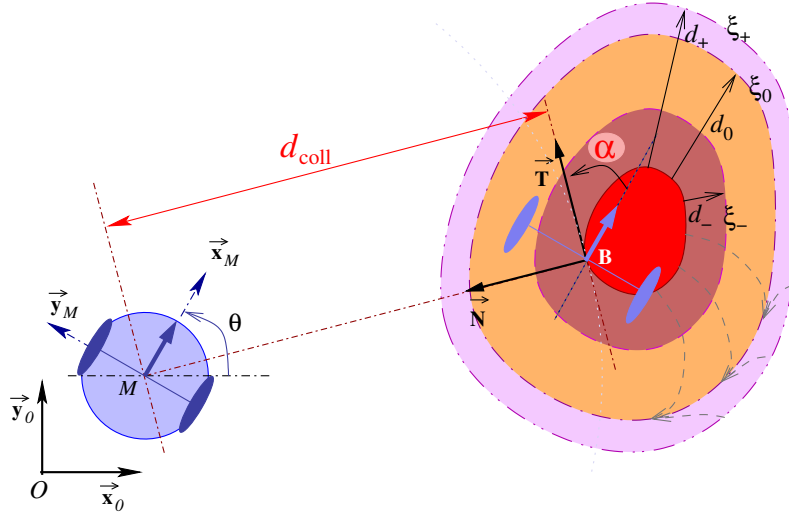


FIG. I.14 – Potentiel répulsif rotatif.

Nous définissons alors l'orientation de la force répulsive de telle sorte qu'elle soit :

- orthogonale à la surface de l'obstacle lorsque le robot en est très proche ;
- tangente à l'obstacle lorsque le robot se trouve sur l'enveloppe ξ_0 ;
- légèrement orientée vers l'obstacle, lorsque le robot se trouve entre les enveloppes ξ_0 et ξ_+ de manière à imposer au robot de suivre l'enveloppe ξ_0 .

Afin de déterminer une telle force, nous considérons la fonction de potentiel suivante :

$$U_{\text{coll}}(d_{\text{coll}}) = \begin{cases} \frac{1}{2}k_1 \left(\frac{1}{d_{\text{coll}}} - \frac{1}{d_+} \right)^2 + \frac{1}{2}k_2 (d_{\text{coll}} - d_+)^2 & \text{si } d_{\text{coll}} < d_+ \\ 0 & \text{sinon} \end{cases} \quad (\text{I.22})$$

où k_1 et k_2 sont deux gains scalaires positifs. Le premier terme, prépondérant à proximité de l'obstacle, tend vers l'infini pour garantir la non-collision. Le second terme quant à lui, dominant au voisinage de ξ_+ , permet au potentiel de conserver une valeur suffisamment importante pour pouvoir commander le robot sur l'enveloppe ξ_0 . Sur la base de la fonction de potentiel (I.22) nous définissons séparément le module et l'orientation de la force répulsive, notés respectivement F et β :

- F se déduit de la dérivée de la fonction de potentiel U_{coll} par rapport à d_{coll} , soit :

$$F = -\frac{\partial U_{\text{coll}}}{\partial d_{\text{coll}}}(d_{\text{coll}}) = k_1 \frac{1}{d_{\text{coll}}^2} \left(\frac{1}{d_{\text{coll}}} - \frac{1}{d_+} \right) - k_2 (d_{\text{coll}} - d_+) \quad (\text{I.23})$$

- L'orientation β , exprimée dans le repère lié au robot, se définit en fonction de la distance à l'obstacle d_{coll} et de l'orientation relative du robot par rapport à l'obstacle, α (cf. figure I.14), de la manière suivante :

$$\beta = \alpha - \frac{\pi}{2} \frac{d_{\text{coll}}}{d_0} + \frac{\pi}{2} \quad (\text{I.24})$$

Ainsi, à proximité de l'obstacle l'angle β tend vers $\alpha + \frac{\pi}{2}$ produisant une force répulsive normale à l'obstacle. De même, sur l'enveloppe ξ_0 , $d_{\text{coll}} = d_0$, $\beta = \alpha$, et le robot subit une force tangentielle non nulle qui permet au robot de suivre l'enveloppe de sécurité ξ_0 . Si le robot s'écarte de cette enveloppe et pénètre dans le couloir délimité par ξ_0 et ξ_+ l'angle β diminue et la force est alors légèrement dirigée vers l'obstacle, ramenant le véhicule vers ξ_0 .

À présent nous sommes en mesure de déterminer, sur la base des équations (I.23) et (I.24), la commande de la base mobile, c'est-à-dire les vitesses linéaire et angulaire assurant l'évitement d'obstacle. En effet, les lois de la mécanique classique permettent de lier les forces appliquées à un système à sa dynamique, et donc d'établir la commande de ce système. Pour ce faire, il est nécessaire de définir un point P_F d'application de cette force. Le choix le plus simple pour un système holonome consiste à choisir son centre d'inertie. En revanche, dans le cas d'un système non-holonome, c'est-à-dire ayant des contraintes cinématiques non intégrables, le problème est plus complexe, car un tel système ne peut être commandé en plaçant le point d'application P_F sur l'axe des roues \vec{y}_M . En effet, du fait de la contrainte non-holonome le robot ne peut se déplacer dans l'axe des roues, et par conséquent la force ne peut s'appliquer. Un choix judicieux consiste à positionner le point d'application P_F sur l'axe \vec{x}_M du robot, et en particulier de le faire coïncider avec le centre P de la platine. La commande d'évitement s'exprime alors comme suit :

$$v_{\text{coll}} = k_{\text{coll}}^v F \cos \beta \quad (\text{I.25})$$

$$\omega_{\text{coll}} = k_{\text{coll}}^\omega \frac{F}{D_x} \sin \beta \quad (\text{I.26})$$

où k_{coll}^v et k_{coll}^ω sont deux gains scalaires positifs qui permettent de fixer la dynamique de la commande d'évitement.

Remarque I.8: *Selon le principe fondamental de la dynamique nous savons qu'une force est proportionnelle à une accélération. Cependant, quand un robot peut être commandé en vitesse et que seul son modèle cinématique est pris en compte, la force virtuelle produite peut être considérée comme directement proportionnelle à la vitesse.*

Finalement, l'approche des potentiels rotatifs permet de synthétiser une loi de commande contraignant le robot à se déplacer le long d'une enveloppe de sécurité ξ_0 . En ce sens, elle s'apparente aux méthodes de suivi de chemin que nous présentons au paragraphe suivant. Ainsi, en définissant un chemin de contournement ξ_0 autour de chaque obstacle les problèmes de puits de potentiels inhérents aux méthodes classiques sont ici limités [CADENAT, 1999].

1.2.3 Formalisme du suivi de chemin

Le formalisme du suivi de chemin consiste à synthétiser une loi de commande assurant la convergence géométrique du robot vers un chemin de référence défini a priori. Plusieurs approches et implémentations ont été proposées, citons par exemple [SAMPEI *et al.*, 1991, 1995][SAMSON, 1992][FREZZA *et al.*, 1998]. Nous exploitons dans le cadre de nos travaux l'approche développée par SAMSON [1992]¹⁰, et la présentons ici comme un cas particulier du suivi de trajectoire appliqué au contournement d'obstacle [HAMEL, 1996][CADENAT, 1999].

Application du suivi de chemin à l'évitement d'obstacle

CADENAT *et al.* [1999A] proposent d'appliquer le formalisme du suivi de chemin au cas de l'évitement d'obstacle. L'idée consiste à définir un chemin de contournement, notée ξ_0 , situé à une distance d_0 de l'obstacle (voir figure I.15). En projetant orthogonalement le

10. Une description générale du formalisme du suivi de chemin est disponible en annexe A.

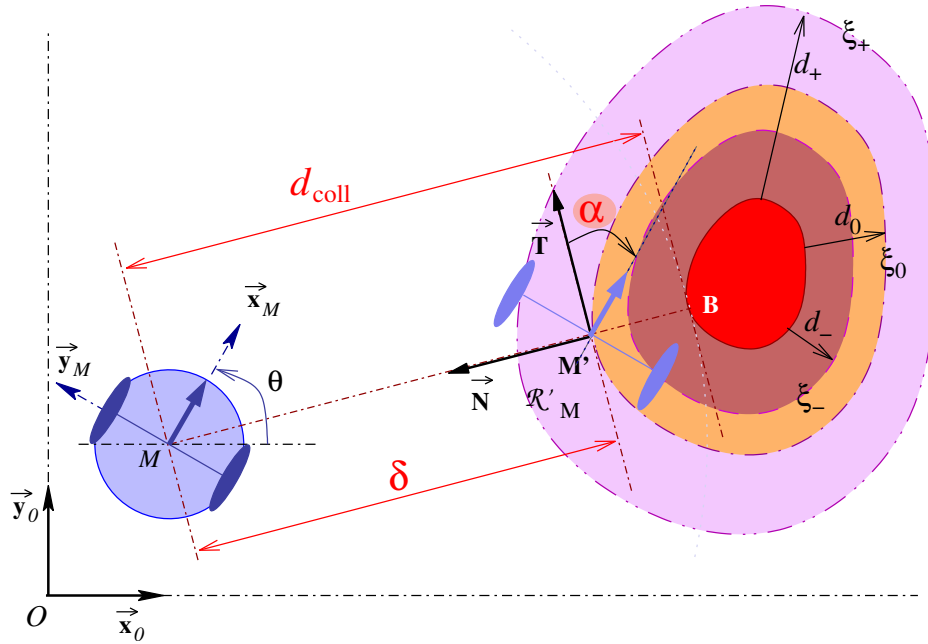


FIG. I.15 – Évitement d'obstacle par suivi de chemin.

robot sur l'enveloppe ξ_0 on peut définir un repère de Frenet \mathcal{R}'_M d'origine M' . Rappelons que le système robotique, par le biais des capteurs ultrasoniques embarqués, peut caractériser la distance d_{coll} et l'angle α , exprimés dans le repère de Frenet \mathcal{R}'_M . L'erreur de positionnement latéral est alors définie par $y_e = \delta = (d_{\text{coll}} - d_0)$, et l'erreur d'orientation par $\theta_e = \alpha$. La dynamique du couple (δ, α) par rapport au repère \mathcal{R}'_M se déduit de (A.2) et s'écrit comme suit [CANUDAS DE WIT *et al.*, 1993] :

$$\begin{cases} \dot{\delta} &= v_{\text{coll}} \sin \alpha \\ \dot{\alpha} &= \omega_{\text{coll}} - \frac{\bar{\chi} v_{\text{coll}} \cos \alpha}{1 + \sigma \bar{\chi} \delta} \end{cases} \quad (\text{I.27})$$

où $\bar{\chi} = \frac{\sigma}{R}$ représente la courbure de l'obstacle et R désigne son rayon de courbure. σ est défini selon le sens de contournement (*cf.* remarque I.7), de la manière suivante :

- $\sigma = +1$ si le robot contourne l'obstacle dans le sens trigonométrique ;
- $\sigma = -1$ si le robot contourne l'obstacle dans le sens horaire ;
- $\sigma = 0$ si le contour de l'obstacle est localement une ligne droite.

Dans le formalisme du suivi de chemin [SAMSON, 1992], l'objectif est d'assurer la convergence géométrique du centre M du véhicule vers le chemin à suivre ξ_0 *indépendamment de la vitesse linéaire du robot*. Notre but est donc maintenant de déterminer une loi de commande ω_{coll} permettant d'amener le couple (δ, α) à zéro, étant donné une vitesse linéaire non nulle définie a priori. Ainsi, une loi de commande permettant de stabiliser le système (I.27) en zéro est donné par [CADENAT *et al.*, 1999A] :

$$v_{\text{coll}} = v_0 \neq 0 \quad (\text{I.28})$$

$$\omega_{\text{coll}} = -v_{\text{coll}} (\kappa \delta + \alpha + 2\kappa \sin \alpha - \bar{\chi} \cos \alpha) \quad (\text{I.29})$$

où κ est un gain positif, la vitesse linéaire v_{coll} étant fixée constante et égale à $v_0 \neq 0$. [CADENAT *et al.*, 1999A] propose en outre une preuve de la stabilité de ce correcteur, ainsi qu'une méthode permettant de choisir le gain κ .

I.3 Commande référencée multi-capteurs en environnement encombré

Notre objectif est maintenant de synthétiser des lois de commande permettant à un robot mobile non-holonyme de naviguer dans un environnement contraint sur la base des données visuelles perçues par la caméra embarquée. Pour cela, il est nécessaire d'intégrer les fonctionnalités d'évitement d'obstacles que nous venons de présenter. La stratégie adoptée ici consiste à réaliser la tâche référencée vision lorsqu'il n'y a pas d'obstacle dans le voisinage du robot, et à basculer sur l'évitement d'obstacle lorsqu'un risque de collision se présente. Pour ce faire, il faut pouvoir *enchaîner* efficacement les différentes tâches robotiques. Il existe différentes techniques permettant d'enchaîner des tâches. Par exemple SOUÈRES, P. et CADENAT, V. [2003] proposent un formalisme d'enchaînement dynamique exploitant les propriétés de la géométrie différentielle. Le basculement entre les différentes tâches est alors basé sur la définition d'une structure différentielle sur l'espace d'état du robot sur lequel le séquençage dynamique peut être défini. MANSARD, N. [2006][MANSARD et CHAUMETTE, 2004, 2005A] utilise le formalisme des tâches redondantes pour proposer une méthode d'enchaînement de tâches. L'idée consiste d'une part à empiler des tâches élémentaires jusqu'à ce que le mouvement de robot soit complètement contraint et la situation désirée atteinte; et, d'autre part, à remplacer une tâche par une autre selon le contexte d'exécution. La méthode envisagée ici est la plus simple, et consiste à synthétiser séparément la commande d'évitement et celle guidant le robot vers son objectif, puis à fusionner les deux correcteurs ainsi obtenus par une combinaison convexe [PISSARD-GIBOLLET et RIVES, 1995] [CADENAT, 1999]. Il est ainsi possible de définir une loi de commande globale permettant de réaliser la tâche de navigation désirée. Celle-ci s'écrit :

$$\dot{q} = (1 - \mu_{\text{coll}})\dot{q}_{AV} + \mu_{\text{coll}}\dot{q}_{\text{coll}} \quad \forall \mu_{\text{coll}} \in [0; 1] \quad (\text{I.30})$$

où

- $\dot{q} = (v \ \omega \ \varpi)^T$ définit la commande en vitesse globale appliquée au robot. Nous rappelons que v et ω représentent les vitesses linéaires et angulaires de la base mobile, et ϖ la vitesse angulaire de la platine;
- $\dot{q}_{AV} = (v_{AV} \ \omega_{AV} \ \varpi_{AV})^T$ représente la commande permettant de réaliser la tâche de navigation référencée vision désirée;
- $\dot{q}_{\text{coll}} = (v_{\text{coll}} \ \omega_{\text{coll}} \ \varpi_{\text{coll}})^T$, où les vitesses v_{coll} et ω_{coll} sont déterminées par la commande assurant l'évitement de collision. La commande de la platine ϖ_{coll} reste encore à être défini (*cf.* paragraphe I.3.1);
- μ_{coll} est une fonction de la distance à l'obstacle d_{coll} qui permet de lisser le basculement d'un correcteur à l'autre.

Nous pouvons alors définir des schémas de commande référencées multi-capteurs basés d'une part sur l'utilisation de la commande référencée vision, et, d'autre part, sur les commandes exploitant les mesures fournies par les capteurs ultrasoniques embarqués. Néanmoins les techniques d'évitement d'obstacles proposées ne commandent que les vitesses linéaire v_{coll} , et angulaire ω_{coll} de la base mobile. L'utilisation seule de ces approches ne permet donc pas de contrôler la platine, et, de ce fait, le motif visuel peut être perdu durant la phase de contournement. Afin d'éviter ce phénomène, [CADENAT, 1999] propose de commander la platine de manière à compenser le mouvement de contournement de la base. Ainsi, nous présentons, tout d'abord au paragraphe I.3.1 une approche intégrant

la commande de la platine aux techniques de contournement d'obstacles précédemment décrites. Puis à la section I.3.2 nous proposons une première extension de ces travaux consistant à exploiter conjointement les formalismes des fonctions de tâches et du suivi de chemin afin de définir une tâche référencée multi-capteurs unique permettant à la fois de contourner les obstacles et de suivre l'amer visuel. s

I.3.1 Commande de la platine

Nous cherchons ici à définir la commande ϖ_{coll} de la platine lors de l'évitement de collision. Pour ce faire, nous nous appuyons sur les travaux de TSAKIRIS *et al.* [1996, 1997] où il s'agissait de stabiliser un robot mobile en une situation fixe tout en suivant un amer visuel d'intérêt à l'aide d'une caméra embarquée. La méthode employée repose sur la désolidarisation de la base mobile et de la platine. Nous avons adapté cette méthode à notre cas d'étude pour garantir la visibilité de l'amer pendant le contournement de l'obstacle. En effet, pendant la phase d'évitement, si le mouvement de la base n'est pas compensé, la valeur de la fonction de tâche référencée vision guidant le robot vers son but augmente et les indices visuels peuvent être perdus. Dans la mesure où la caméra est contrainte à se déplacer dans le plan horizontal, cette erreur se traduit par un déplacement latéral des indices visuels dans l'image, et affectera donc les ordonnées Y_j des points projetés. Pour cette raison, nous considérons que la tâche référencée vision à réaliser en phase d'évitement est de réguler à zéro une nouvelle fonction de tâche e_Y dépendant des ordonnées Y_j . En outre, puisque nous n'utilisons qu'une seule commande pour déplacer la caméra indépendamment de la base, nous considérons uniquement une erreur scalaire afin de satisfaire la propriété de ρ -admissibilité. Pour cela, nous nous proposons simplement de centrer l'amer dans l'image et définissons la fonction de tâche de centrage comme suit :

$$e_Y = Y_0 \quad (\text{I.31})$$

où Y_0 représente l'ordonnée d'un point du motif visuel, ou de son centre de gravité. Nous imposons alors à la dynamique de la fonction de tâche de décroître de manière exponentielle :

$$\dot{e}_Y = \mathcal{L}_{Y_0} \mathcal{T}_r = -\lambda_Y e_Y, \quad \lambda_Y > 0 \quad (\text{I.32})$$

où λ_Y est un gain positif scalaire, et \mathcal{L}_{Y_0} désigne la matrice d'interaction réduite définie uniquement pour l'ordonnée Y_0 de profondeur z_0 , soit :

$$\mathcal{L}_{Y_0} = \begin{pmatrix} -\frac{f}{z_0} & \frac{Y_0}{z_0} & f + \frac{Y_0^2}{f} \end{pmatrix} \quad (\text{I.33})$$

Remarque I.9: Dans le cas où Y_0 correspond à l'ordonnée du centre de gravité du motif visuel sur le plan image, la matrice d'interaction (I.33) n'est alors qu'une approximation de \mathcal{L}_{Y_0} . En effet, la matrice d'interaction associée aux centres de gravité se déduit des matrices d'interaction associées aux moments de l'image [CHAUMETTE, 2004] (cf. paragraphe III.2.3, page 94)

Il reste désormais à déterminer la commande de la platine satisfaisant cette dynamique. Dans ce but, nous exploitons l'idée proposée dans [DE LUCA et ORIOLO, 1990] [TSAKIRIS *et al.*, 1996, 1997] qui consiste à partitionner la jacobienne \mathbf{J}_r du robot de manière à séparer les

termes relatifs à la base de ceux de la platine, soit en reprenant la définition du torseur cinématique réduit de la caméra (I.7) :

$$\mathcal{T}_r = \begin{pmatrix} \mathbf{J}_{\text{base}} & \mathbf{J}_{\text{pl}} \end{pmatrix} \begin{pmatrix} \dot{q}_{\text{base}} \\ \dot{q}_{\text{pl}} \end{pmatrix} \quad (\text{I.34})$$

où

- \mathbf{J}_{base} désigne la matrice constituée des deux premières colonnes de la jacobienne \mathbf{J}_r , correspondant aux mouvements de la base mobile ;
- \mathbf{J}_{pl} est donnée par la troisième colonne de la jacobienne \mathbf{J}_r , correspondant au mouvement de la platine ;
- $\dot{q}_{\text{base}} = (v_{\text{coll}} \ \omega_{\text{coll}})^T$ représente les vitesses linéaire et angulaire de la base en phase de contournement, définies par les relations (I.25) et (I.26), ou (I.28) et (I.29) ;
- \dot{q}_{pl} désigne la commande de la platine en phase de contournement, c'est-à-dire la vitesse de rotation ϖ_{coll} .

En considérant les équations (I.32) et (I.34) nous pouvons établir :

$$\mathcal{L}_{Y_0} \mathbf{J}_{\text{base}} \dot{q}_{\text{base}} + \mathcal{L}_{Y_0} \mathbf{J}_{\text{pl}} \dot{q}_{\text{pl}} = -\lambda_Y \mathbf{e}_Y$$

D'où nous déduisons l'expression de la commande de la platine permettant de centrer la cible dans l'image en phase d'évitement :

$$\dot{q}_{\text{pl}} = \varpi_{\text{coll}} = -\frac{1}{\mathcal{L}_{Y_0} \mathbf{J}_{\text{pl}}} (\lambda_Y \mathbf{e}_Y + \mathcal{L}_{Y_0} \mathbf{J}_{\text{base}} \dot{q}_{\text{base}}) \quad (\text{I.35})$$

Remarque I.10: Le terme $(\mathcal{L}_{Y_0} \mathbf{J}_{\text{pl}})$ est un scalaire non nul du fait de la définition de la fonction de tâche \mathbf{e}_Y . La propriété de ρ -admissibilité est donc satisfaite, et la commande ϖ_{coll} peut donc être déterminée de manière unique et exacte.

Ainsi, la commande de la platine (I.35) intègre l'erreur de suivi de l'amer visuel, tout en compensant les déplacements de la base mobile en phase d'évitement. De cette manière, la conservation de l'amer visuel dans l'image est garantie pendant la phase de contournement d'obstacles. Enfin, le tableau I.3 suivant résume les différentes lois de commande d'évitement d'obstacles proposées.

Potentiels rotatifs	$v_{\text{coll}} = k_{\text{coll}}^v F \cos \beta$ (I.25)
	$\omega_{\text{coll}} = k_{\text{coll}}^\omega \frac{F}{D_F} \sin \beta$ (I.26)
	$\varpi_{\text{coll}} = -\frac{1}{\mathcal{L}_{Y_0} \mathbf{J}_{\text{pl}}} (\lambda_Y \mathbf{e}_Y + \mathcal{L}_{Y_0} \mathbf{J}_{\text{base}} \dot{q}_{\text{base}})$ (I.35)
Suivi de chemin	$v_{\text{coll}} = v_0$ (I.28)
	$\omega_{\text{coll}} = -v_{\text{coll}} (k\delta + \alpha + 2k \sin \alpha - \bar{\chi} \cos \alpha)$ (I.29)
	$\varpi_{\text{coll}} = -\frac{1}{\mathcal{L}_{Y_0} \mathbf{J}_{\text{pl}}} (\lambda_Y \mathbf{e}_Y + \mathcal{L}_{Y_0} \mathbf{J}_{\text{base}} \dot{q}_{\text{base}})$ (I.35)

TAB. I.3 – Commande d'évitement d'obstacles \dot{q}_{coll}

I.3.2 Application conjointe du formalisme des fonctions de tâche et du suivi de chemin

Notre objectif ici est de définir une fonction de tâche référencée multi-capteurs permettant d'une part à la base mobile de contourner l'obstacle, et, d'autre part, à la platine de suivre l'amer visuel. L'idée proposée est d'utiliser à la fois le formalisme des fonctions de tâches [SAMSON *et al.*, 1991] précédemment introduit à la section I.1.5.1, et le formalisme du suivi de chemin décrit au paragraphe I.2.3. Il s'agit ainsi de déterminer une fonction de tâche unique permettant de suivre une enveloppe de sécurité ξ_0 tout en assurant que le motif visuel reste dans le champ de vision de la caméra pendant le contournement. Afin de remplir le premier objectif il est nécessaire de stabiliser à zéro le système (I.27). Pour cela, nous avons choisi une approche reposant sur les techniques de régime glissant et consistant à réguler à zéro la variable de glissement γ suivante :

$$\gamma = \delta + \kappa\alpha, \quad \forall \kappa \in \mathbb{R}^* \quad (\text{I.36})$$

où κ est un gain scalaire non nul qui permet de déterminer la vitesse relative de convergence des erreurs de position δ et d'orientation α . En effet, une valeur importante de κ conduit à privilégier la convergence de α au détriment de celle de δ , et inversement si l'on choisit une faible valeur de κ . On peut montrer que la convergence de γ à zéro entraîne celle du couple (δ, α) , répondant ainsi au problème du suivi de chemin. Le lecteur intéressé pourra trouver une preuve de stabilité détaillée dans [CADENAT, 1999] et [SOUÈRES *et al.*, 1998]. Cependant, la régulation à zéro de γ ne permet que de fixer la vitesse angulaire ω_{coll} , la convergence vers le chemin de contournement étant définie de manière géométrique (*i.e.* indépendante de v_{coll}). Afin de fixer la vitesse linéaire v_{coll} , nous imposons au robot de suivre l'enveloppe de contournement ξ_0 avec une vitesse constante $v_0 \neq 0$ ¹¹ prédéfinie, ce qui conduit à réguler à zéro l'erreur : $(l - v_0 t)$, où l représente l'abscisse curviligne du centre M du robot. Nous pouvons alors définir la fonction de tâche \mathbf{e}_{ch} qui permet à la base mobile de contourner l'obstacle avec une vitesse $v_0 \neq 0$:

$$\mathbf{e}_{\text{ch}} = \begin{pmatrix} l - v_0 t \\ \delta + \kappa\alpha \end{pmatrix} \quad (\text{I.37})$$

De plus, il faut également assurer que le motif visuel reste dans le champ de vision de la caméra lors de la phase de contournement. Pour cela, nous reprenons la même idée consistant à centrer l'amer dans l'image en régulant à zéro la fonction de tâche $\mathbf{e}_Y = Y_0$ (I.31), où Y_0 représente l'ordonnée du centre du motif visuel dans le plan image. On peut alors définir une fonction de tâche de contournement d'obstacle \mathbf{e}_{coll} :

$$\mathbf{e}_{\text{coll}} = \begin{pmatrix} \mathbf{e}_{\text{ch}} \\ \mathbf{e}_Y \end{pmatrix} \quad (\text{I.38})$$

Cette fonction de tâche dépend à la fois des indices visuels et des mesures proximétriques fournis respectivement par la caméra et les ultrasons. Il s'agit donc d'une vraie tâche référencée multi-capteurs. De plus, les trois objectifs ainsi définis sont indépendants et compatibles. Leur régulation à zéro est donc possible et conduira à un mouvement cohérent

11. La vitesse linéaire constante $v_0 \neq 0$ doit être choisie assez faible pour que le robot ait le temps de ralentir suffisamment pour éviter toute collision, lorsqu'il se trouve dans la zone critique délimité par ξ_- .

du robot. En dérivant \mathbf{e}_{coll} par rapport au temps, il vient :

$$\dot{\mathbf{e}}_{\text{coll}} = \begin{pmatrix} v - v_0 \\ \dot{\delta} + \kappa \dot{\alpha} \\ \dot{Y}_0 \end{pmatrix}$$

À l'aide des relations (I.32) et (I.27), $\dot{\mathbf{e}}_{\text{coll}}$ peut se réécrire comme suit :

$$\dot{\mathbf{e}}_{\text{coll}} = \begin{pmatrix} 1 & 0 & 0 \\ \sin \alpha - \kappa \bar{\chi} & \kappa & 0 \\ \mathcal{L}_{Y_0} \mathbf{J}_r & & \end{pmatrix} \begin{pmatrix} v_{\text{coll}} \\ \omega_{\text{coll}} \\ \varpi_{\text{coll}} \end{pmatrix} + \begin{pmatrix} -v_0 \\ 0 \\ 0 \end{pmatrix}$$

De cette équation il est possible de déduire la commande de contournement :

$$\dot{q}_{\text{coll}} = \mathbf{J}_{\text{coll}}^{-1} (\dot{\mathbf{e}}_{\text{coll}} + B) , \quad \text{avec } \mathbf{J}_{\text{coll}} = \begin{pmatrix} 1 & 0 & 0 \\ \sin \alpha - \kappa \bar{\chi} & \kappa & 0 \\ \mathcal{L}_{Y_0} \mathbf{J}_r & & \end{pmatrix} \quad \text{et } B = \begin{pmatrix} -v_0 \\ 0 \\ 0 \end{pmatrix}$$

Notons que comme le déterminant de \mathbf{J}_{coll} est non nul :

$$\det(\mathbf{J}_{\text{coll}}) = -\kappa \left(\frac{C_x}{z} + \frac{C_y Y_0}{z} + 1 + Y_0^2 \right) \neq 0, \quad \forall \kappa \in \mathbb{R}^*$$

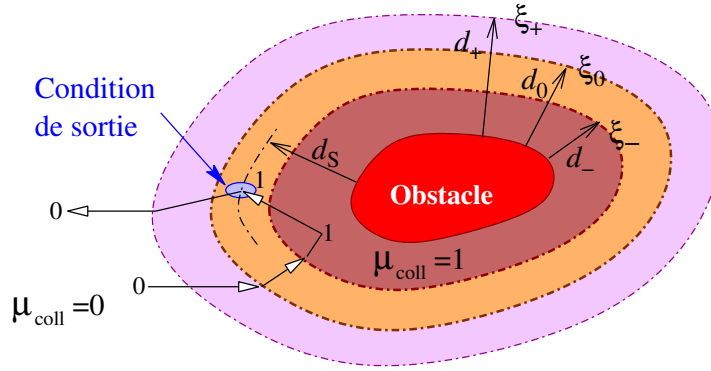
\mathbf{J}_{coll} est une matrice inversible et la propriété de ρ -admissibilité est satisfaite. Par ailleurs, la convergence de \mathbf{e}_{coll} vers zéro peut être assurée en imposant à la fonction de tâche de décroître de manière exponentielle, soit : $\dot{\mathbf{e}}_{\text{coll}} = -\lambda_{\text{coll}} \mathbf{e}_{\text{coll}}$, avec λ_{coll} un gain scalaire positif ou une matrice définie positive permettant de fixer la vitesse de décroissance. On obtient alors la commande d'évitement d'obstacle suivante :

$$\dot{q}_{\text{coll}} = \mathbf{J}_{\text{coll}}^{-1} (-\lambda_{\text{coll}} \mathbf{e}_{\text{coll}} - B) \quad (\text{I.39})$$

Nous avons ainsi défini une loi de commande référencée multi-capteurs permettant à notre robot mobile de contourner les obstacles, tout en suivant l'amer visuel. Nous allons à présent exploiter les schémas de commande d'évitement réunit dans le tableau I.3, ainsi que la relation (I.39), pour définir des stratégies de navigation référencée vision dans des environnement encombrés.

1.3.3 Réalisation de tâches de navigation référencée vision en environnement encombré

Notre objectif ici est de réaliser une tâche de positionnement par rapport à un amer visuel donné tout en évitant les éventuels obstacles. Notre stratégie de commande consiste à fusionner les différents correcteurs à l'aide de la relation (I.30). Cette stratégie repose sur la définition de la fonction μ_{coll} qui permet d'activer l'un ou l'autre correcteur en fonction de l'environnement local au robot. Ainsi lorsqu'il n'y a pas de risque de collision, *e.g.* le robot est hors de la zone d'influence ξ_+ de l'obstacle (voir figure I.16), μ_{coll} est fixé à 0 et seul \dot{q}_{AV} commande le robot. Inversement si le robot est à proximité d'un obstacle, μ_{coll} passe à 1 de manière à ce que le robot assure l'évitement au moyen de la commande \dot{q}_{coll} . Une évolution de μ_{coll} purement binaire n'est pas envisageable, car cela produirait

FIG. I.16 – Principe d'évolution de la fonction μ_{coll}

une discontinuité de la loi de commande (I.30). La variable μ_{coll} est donc définie comme une fonction continue de la distance d_{coll} , et elle dépend de la stratégie de commande souhaitée. Nous définissons μ_{coll} selon l'algorithme de commande suivant (*cf.* figure I.16) :

$$\begin{cases} \mu_{\text{coll}} = 0 & \text{si } d_{\text{coll}} \geq d_0 \quad \text{ET } \acute{\text{E}}\text{VITE}_{\text{coll}} = 0 \\ \mu_{\text{coll}} = \frac{d_{\text{coll}} - d_0}{d_- - d_0} & \text{si } d_{\text{coll}} \in [d_-, d_0] \quad \text{ET } \acute{\text{E}}\text{VITE}_{\text{coll}} = 0 \\ \mu_{\text{coll}} = \frac{d_{\text{coll}} - d_+}{d_S - d_+} & \text{si } d_{\text{coll}} \in [d_S, d_+] \quad \text{ET } \text{SORTIR}_{\text{coll}} = 1 \\ \mu_{\text{coll}} = 1 & \text{sinon} \end{cases} \quad (\text{I.40})$$

où

- $\acute{\text{E}}\text{VITE}_{\text{coll}}$ est un indicateur positionné à 1 lorsque le robot franchit l'enveloppe de sécurité ξ_- . Il permet de maintenir μ_{coll} à 1, et par conséquent la commande de contournement \dot{q}_{coll} , tant qu'il y a un risque de collision. Ainsi, les antagonismes entre les effets attractifs et répulsifs des différentes commandes s'en trouvent réduits.
- $\text{SORTIR}_{\text{coll}}$ est un indicateur de sortie qui passe à 1 lorsque la condition de sortie est satisfaite. Cette condition doit être vérifiée lorsque la tâche référencée vision \dot{q}_{AV} et la tâche d'évitement d'obstacle \dot{q}_{coll} deviennent compatibles. [CADENAT, 1999] propose de définir cet événement dès lors que l'obstacle est dépassé. Cela se définit par le fait que les directions de la base mobile et de la platine deviennent parallèle à l'obstacle, se traduisant par l'annulation des angles ϑ et α . Dans ce manuscrit, nous avons proposé de définir la condition de sortie lorsque les commandes \dot{q}_{AV} et \dot{q}_{coll} indiquent la même direction, soit : $\text{sign}(\dot{q}_{\text{AV}}) = \text{sign}(\dot{q}_{\text{coll}})$ ¹².
- d_S est la distance du robot à l'obstacle relevée par le capteur quand la condition de sortie intervient (*i.e.* lorsque $\text{SORTIR}_{\text{coll}}$ passe de 0 à 1). Elle permet de garantir la continuité de l'évolution de μ_{coll} au moment où le robot est autorisé à quitter l'enveloppe de contournement de l'obstacle.

Nous pouvons désormais illustrer la stratégie de commande référencée multi-capteurs permettant à un robot mobile muni d'une caméra de réaliser une tâche de navigation basée sur la vision tout en évitant les éventuelles collisions.

12. Si l'une des vitesses (v, ω, ϖ) est nulle, nous considérons que la direction correspondante est bonne.

I.3.3.1 Simulation d'une tâche de navigation référencée multi-capteurs exploitant (I.39)

Afin d'illustrer la réalisation d'une tâche référencée multi-capteurs nous présentons ici un exemple de simulation. La tâche de navigation considérée consiste à réaliser une tâche référencée vision tout en contournant deux obstacles de hauteur inférieure à celle de la caméra. Le premier obstacle est un muret de 4.55 m de longueur pour 20 cm de large, le second est un obstacle cylindrique de 30 cm de diamètre. Les enveloppes de contournement ξ_- , ξ_0 et ξ_+ sont fixées respectivement à 30 cm, 50 cm et 60 cm de l'obstacle. La trajectoire du robot et l'évolution correspondante des indices visuels sont représentées sur les figures I.17.a et I.17.b, et montrent que la tâche de navigation est parfaitement réalisée.

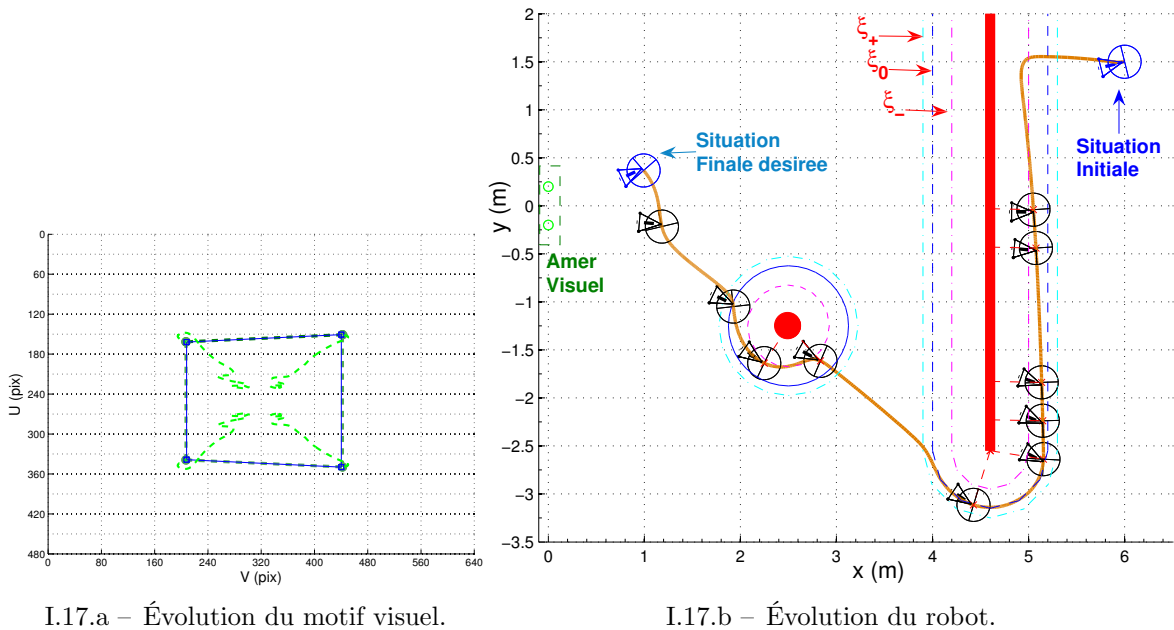


FIG. I.17 – Asservissement visuel et évitement de collision par application conjointe du formalisme des fonctions de tâche et du suivi de chemin.

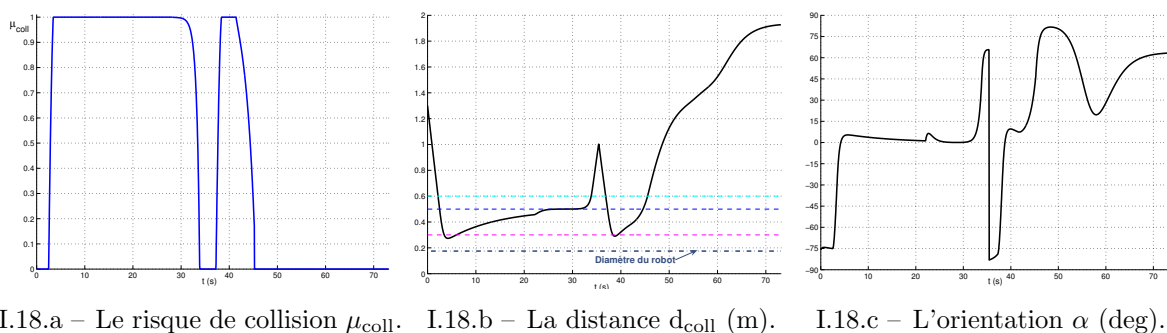


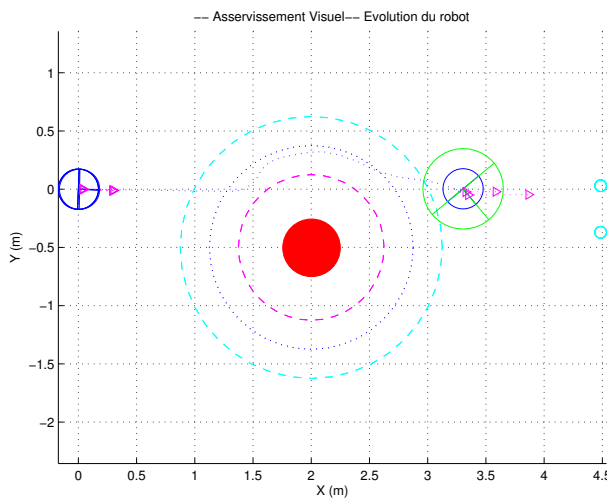
FIG. I.18 – Les mesures du risque de collision.

La figure I.18.a décrit la progression de la fonction de basculement μ_{coll} , et les figures I.18.b et I.18.c présentent respectivement l'évolution de la distance d_{coll} et de l'orientation α relative entre l'obstacle et le robot. Notons que le saut constaté sur l'angle α est lié au changement d'obstacle : le véhicule s'éloigne du premier obstacle et s'approche du second.

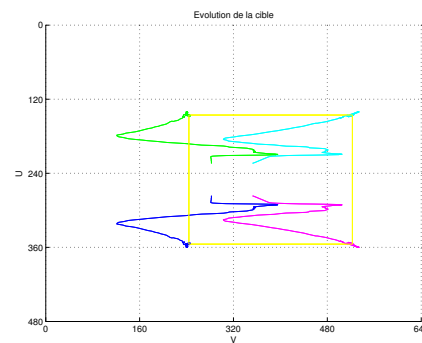
Dans la mesure où la distance entre les deux obstacles est supérieure à $2d_+$ (voir également la remarque I.6), il ne peut pas y avoir de risque de collision, *i.e.* $\mu_{\text{coll}} = 0$. De ce fait, seul l'asservissement visuel est appliquée au robot, et par conséquent le saut ne provoque pas de discontinuité sur la commande globale.

I.3.3.2 Expérimentation d'un asservissement visuel et évitement de collision par la méthode des potentiels rotatifs

Nous avons également expérimenté une tâche de navigation référencée multi-capteurs sur notre robot Super Scout. La mission considérée consiste à positionner la caméra face à un amer visuel dans un environnement encombré d'un obstacle. Cet obstacle est de forme cylindrique de 25 cm de diamètre. L'enveloppe de contournement ξ_0 est fixée à 1 m de l'obstacle, et les enveloppes ξ_- et ξ_+ sont positionnées à 10 cm de part et d'autre de ξ_0 . Nous avons choisi ici de contourner l'obstacle présent à l'aide des potentiels rotatifs.



I.19.a – Évolution du robot.



I.19.b – Évolution du motif visuel.

FIG. I.19 – Asservissement visuel et évitement de collision par la méthode des potentiels rotatifs.

Au début de la tâche, le robot se trouve loin de l'obstacle (*cf.* figure I.19.a). La fonction μ_{coll} est fixée à 0 et le véhicule est alors guidé uniquement par la commande référencée vision \dot{q}_{AV} . Il cherche à converger vers l'amer visuel (*cf.* figure I.19.b) sans prendre en compte l'obstacle présent. Dès qu'il franchit l'enveloppe de sécurité ξ_0 , μ_{coll} augmente, ainsi que la force répulsive correspondante. Le robot est alors commandé par la combinaison de la commande d'asservissement visuel et d'évitement de collision. Le mouvement du robot est alors modifié de telle sorte que l'obstacle soit évité en suivant l'enveloppe de contournement ξ_0 . Si le robot franchissait l'enveloppe ξ_- , μ_{coll} passerait à 1, et la base mobile et la platine seraient alors commandées indépendamment. Enfin, lorsque la condition de sortie est vérifiée, le robot est autorisé à quitter l'enveloppe de sécurité. La fonction μ_{coll} diminue progressivement jusqu'à s'annuler sur ξ_+ . Le robot n'est alors plus piloté que par l'asservissement visuel et continue de converger vers la cible. Enfin, la figure I.20 regroupe les différentes commandes appliquées sur le robot, la première ligne représente la commande globale appliquée à la base mobile (v, ω) et la platine (ϖ), la

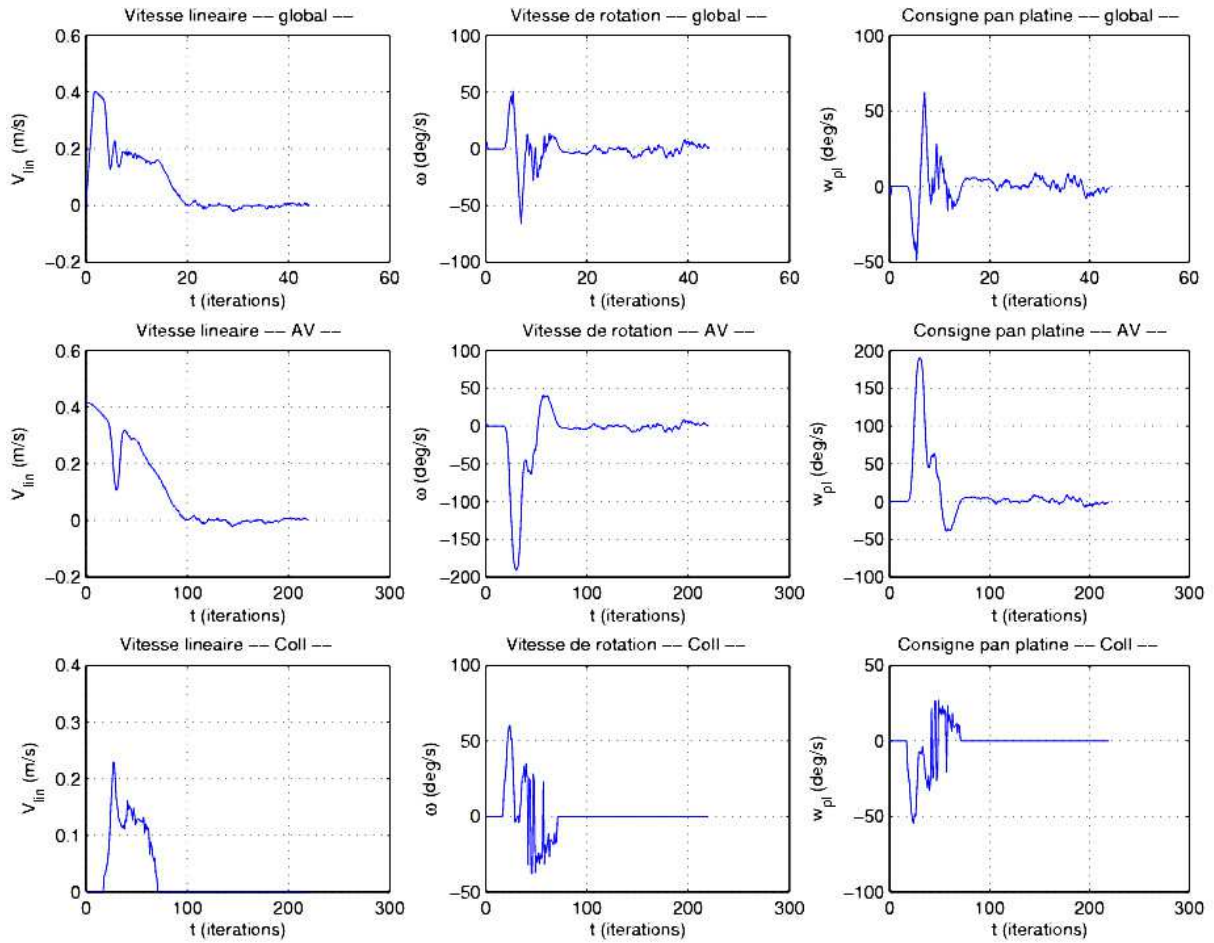


FIG. I.20 – Évolution de la consigne de commande \dot{q} .

seconde la commande référencée vision \dot{q}_{AV} , et enfin la troisième la commande d'évitement \dot{q}_{coll} calculée par la méthode des potentiels rotatifs. Grâce à un choix judicieux des divers gains, la commande globale envoyée au robot, définie par une combinaison linéaire des lois de commande référencées vision et d'évitement d'obstacle, est comprise dans les limites des vitesses données dans le tableau I.1. Les courbes des vitesses de rotation de la base mobile et de la platine mettent bien en évidence les différentes phases opératoires. Au début de l'exécution de la tâche, la base mobile cherche à converger vers la cible, alors que la commande de la platine compense ces mouvements pour conserver les indices visuels dans l'image. Puis, un obstacle est détecté entraînant une variation de la commande d'évitement, pendant que l'asservissement visuel tente toujours de converger vers la cible. La combinaison linéaire de ces deux correcteurs permet alors d'assurer le contournement. Enfin, le robot parvient à s'extraire de l'obstacle et converge alors simplement vers la cible, comme décrit sur la figure I.19.b.

I.3.4 Bilan

Nous avons présenté dans cette section différentes techniques de navigation référencée vision dans un environnement encombré. Les méthodes d'évitement proposées s'appuient sur la définition de lois de commande imposant au robot de se déplacer le long d'une

enveloppe de contournement ξ_0 . Cependant, il est également nécessaire de s'assurer de la visibilité du motif visuel durant la phase d'évitement en pilotant le mouvement de la platine. Ainsi, une première approche repose sur la définition de lois de commande "séparées" pour la base mobile d'une part, et la platine d'autre part. La première sera alors pilotée à l'aide de méthodes de contournement classiques (champs de potentiels, suivi de chemin, etc.), et la seconde de manière à assurer le suivi de l'amer visuel (*cf.* paragraphe I.3.1). Nous avons également proposé une approche alternative consistant à définir une fonction de tâche référencée multi-capteurs assurant conjointement le contournement de l'obstacle et le suivi de la cible (*cf.* paragraphe I.3.2). Ces deux approches conduisent à des résultats satisfaisants, et relativement proches (*i.e.* suivi d'une enveloppe de contournement). La dernière méthode présente néanmoins l'avantage d'utiliser un cadre formel (*i.e.* le formalisme des fonctions de tâches) permettant de garantir le bon conditionnement de la tâche grâce à la condition de ρ -admissibilité. Il devient maintenant possible pour un robot mobile de réaliser des missions guidées par la vision dans des environnements encombrés d'obstacles.

I.4 Conclusion

Dans ce chapitre, nous avons rappelé le principe de la commande référencée vision d'un robot mobile non-holonyme muni d'une caméra. Nous avons alors décrit brièvement le système robotique considéré, et établi les modèles nécessaires pour exprimer le lien entre le mouvement de la caméra et la variation des indices visuels. Nous avons ensuite présenté le formalisme des fonctions de tâche en rappelant les concepts fondamentaux que sont la ρ -admissibilité et les tâches redondantes. Puis, nous avons montré comment exploiter ce formalisme pour synthétiser une loi de commande référencée vision, que nous avons illustrée sur des exemples de simulation et d'expérimentation. Toutefois, l'asservissement visuel ne permettant pas de définir des tâches robotiques dans des environnements encombrés, nous avons présenté les bases de la navigation en présence d'obstacles. Nous nous sommes plus particulièrement intéressés à deux formalismes bien adaptés à l'approche référencée capteur : la méthode des potentiels et le formalisme du suivi de chemin. Ces formalismes ont permis de définir des lois de commande référencées multi-capteurs permettant à notre robot mobile de réaliser une tâche de navigation référencée vision malgré la présence d'obstacles. Néanmoins, les stratégies de navigation en environnement encombré proposées jusqu'ici ne permettent de gérer que des obstacles qui ne perturbent pas la caractérisation des informations visuelles. Elles reposent donc sur l'hypothèse que la hauteur des obstacles reste inférieure à celle de caméra, et que les phénomènes d'occultations ne peuvent se produire. Notre intention est maintenant de nous soustraire à cette contrainte, et de considérer des obstacles pouvant occulter (ou non) le motif visuel. Nous nous proposons ainsi d'étendre les travaux précédents et de définir dans la suite de ce manuscrit des lois de commande permettant à notre robot mobile de naviguer dans un environnement encombré tout en *gérant les éventuelles occultations et collisions*.

Chapitre II

Contribution à l'évitement des occultations

LES lois de commande présentées au chapitre I permettent à un robot mobile non-holonyme de réaliser une tâche de navigation référencée vision dans un environnement encombré d'obstacles. Toutefois, ces approches ne prennent compte que des obstacles qui ne viennent pas perturber l'acquisition du motif visuel. Nous proposons dans ce chapitre d'étendre les asservissements précédents, afin de pouvoir **éviter les phénomènes d'occultations**.

Nous nous proposons, dans un premier temps, d'introduire le phénomène d'occultation en présentant un aperçu de l'état de l'art de travaux en asservissement visuel gérant ces phénomènes. Puis, nous exposons les définitions nécessaires à l'élaboration de nos tâches de navigation d'évitement d'occultations. Enfin, nous détaillons les différentes stratégies de commande que nous avons mises en œuvre permettant à un robot mobile de réaliser une tâche basée sur la vision tout en évitant les collisions et les occultations.

II.1 Introduction

OCCULTATION (*nom, féminin*) XIV^{ème} siècle, emprunté du latin *occultatio* : action de cacher. **Astronomie** : Disparition passagère d'un astre qui est caché par un autre. Exemple : l'occultation d'une étoile par la Lune. Par analogie : action de cacher à la vue, de masquer. L'occultation des lumières pendant une alerte aérienne. L'occultation d'un signal de voie ferrée. **Figuratif** : Le fait de dissimuler, de rendre obscur. L'occultation de la vérité. *Dictionnaire de l'académie Française, neuvième édition, 1986.*

Classiquement, l'occultation se définit comme étant un phénomène de recouvrement apparent d'un élément par un autre. En asservissement visuel, cela se traduit par la perte des informations sensorielles généralement liée au masquage du motif visuel par un élément de l'environnement. Dans le cadre de notre étude, le fait que l'amer visuel puisse sortir du champ de vision de la caméra, sera également assimilé comme étant une occultation.

II.1.1 État de l'art

De premières ébauches de gestion des occultations ont été menées principalement dans le cadre de la robotique de manipulation où, en général, le nombre de degrés de liberté est important. Par exemple, certains auteurs utilisent des techniques issues du *tracking*, tels que WUNSCH, P. et HIRZINGER, G. [1997] qui proposent une méthode permettant de suivre visuellement un objet 3D tandis que les phénomènes d'occultations sont prédits et évités ; ou encore COMPORT *et al.* [2004] qui ont développé un algorithme *robuste* aux changements de luminosité, aux occultations et à la perte du motif visuel. Dans [MARCHAND et HAGER, 1998] [MANSARD et CHAUMETTE, 2005B] les auteurs ont choisi d'exploiter la redondance des bras manipulateurs pour réaliser une tâche référencée vision tout en évitant les occultations. Les travaux menés dans [MEZOUAR et CHAUMETTE, 2002] utilisent la planification de trajectoire des indices visuels dans l'image pour à la fois éviter les occultations dues aux mouvements du bras manipulateur, et maintenir le motif dans le champ de vision de la caméra. Dans la même lignée, SCHRAMM, F. et MOREL, G. [2006] ont proposé, plus récemment, une solution analytique à la planification dans l'image du motif visuel indépendant des paramètres de calibration. CHESI *et al.* [2004] présentent une approche garantissant la visibilité des informations visuelles \mathbf{s} , en partitionnant l'image. Ainsi, tant que \mathbf{s} reste à l'intérieur d'une sous-région de l'image une loi de commande classique est appliquée. Des que les indices visuels franchissent les limites prédéfinies, les auteurs proposent de ramener \mathbf{s} en effectuant des mouvements de rotations et de translations adéquat. D'autres auteurs [MALIS *et al.*, 1999] [CORKE et HUTCHINSON, 2001] [KYRKI *et al.*, 2004] préservent la visibilité du motif visuel en agissant sur certains degrés de liberté, ou en contrôlant le zoom [BENHIMANE et MALIS, 2003]. Dans [GARCIA-ARACIL et MALIS, 2004, GARCIA-ARACIL *et al.*, 2005] les auteurs présentent un formalisme qui modifie la tâche référencée vision nominale de manière à éliminer ou ajouter des éléments au vecteur des informations visuelles \mathbf{s} , permettant ainsi d'enlever les données occultées de ce vecteur. Enfin, récemment, REMAZEILLES *et al.* [2006] proposent une approche dérivée de l'asservissement visuel classique : l'asservissement visuel qualitatif. La caractéristique de la loi de commande proposée réside dans le fait qu'elle conduit à faire converger les mesures visuelles non pas vers une valeur désirée \mathbf{s}^* , mais plutôt vers un intervalle de confiance tout en tenant compte de contraintes, telles que la visibilité des informations sensorielles.

Dans le cadre de nos travaux nous nous intéressons à la définition de tâches de navigation référencées vision pour un robot mobile devant évoluer dans un environnement encombré d'obstacles susceptibles de perturber le signal visuel. Avant de présenter nos stratégies de navigation nous nous proposons de détailler la modélisation de l'occultation.

II.1.2 Modélisation de l'occultation

De manière générale, nous appelons objet occultant, noté \mathcal{O}_{cc} , tout motif dans l'image susceptible de gêner l'extraction des informations visuelles. Les stratégies d'évitement d'occultations, développées dans cet ouvrage, s'appuyant sur la caractérisation dans l'image de ce dernier, le processus de traitement d'image doit être à même de l'identifier et de l'extraire. Nous nous intéressons d'abord à la projection de l'objet occultant dans l'image, avant d'introduire les différentes distances utiles à nos stratégies de commande.

II.1.2.1 Projection de l'objet occultant sur le plan image

Soient $(\vec{x}_{im}, \vec{y}_{im})$ les axes du repère attaché au centre du plan image, (X_{min}, Y_{min}) et (X_{max}, Y_{max}) correspondant à ses bornes minimale et maximale. La figure II.1 illustre la

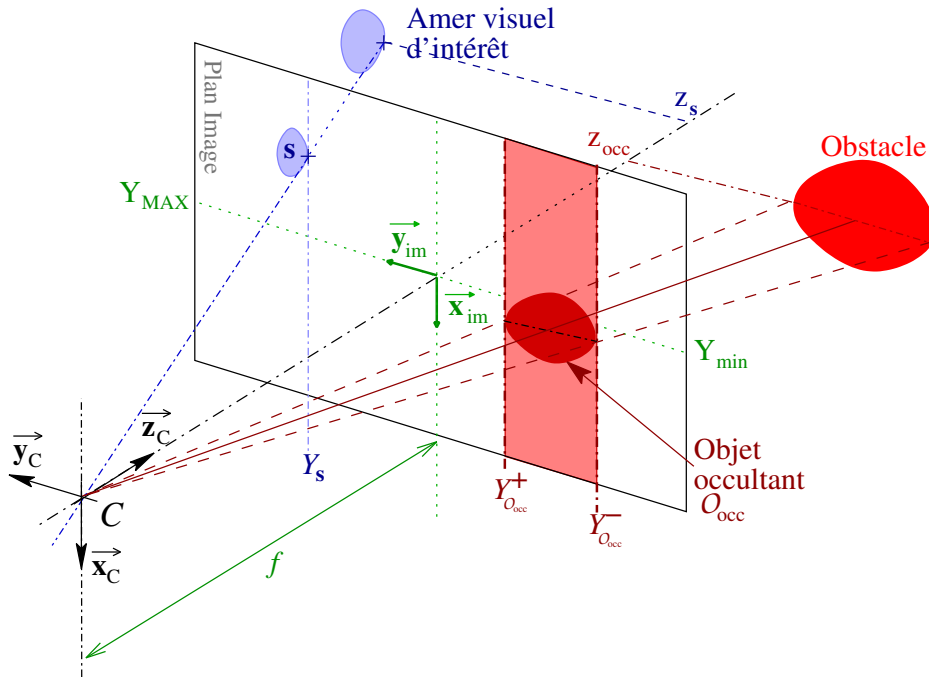
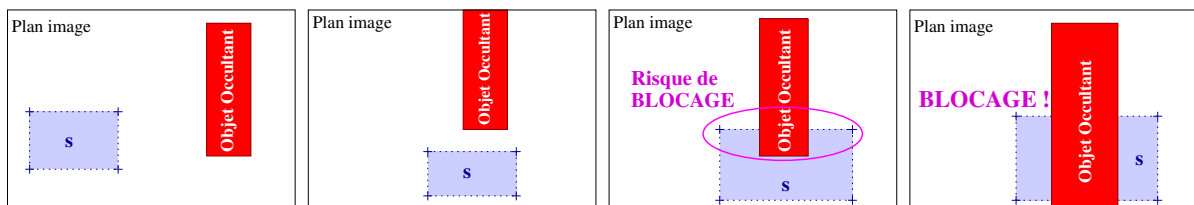


FIG. II.1 – Projection d'un obstacle sur le plan image.

projection d'un obstacle de la scène dans l'image définissant ainsi un objet occultant \mathcal{O}_{cc} . Nous notons alors $Y_{\mathcal{O}_{cc}}^+$ et $Y_{\mathcal{O}_{cc}}^-$ les ordonnées de ses bords gauche et droit. Nous avons choisi de construire nos stratégies d'évitement d'occultation uniquement sur ces deux mesures. Bien que ce choix puisse apparaître au premier abord conservatif, il présente l'avantage d'interdire au motif visuel de passer au-dessus ou en-dessous de \mathcal{O}_{cc} , ce qui permet d'éviter certaines situations de blocage, telles que celles représentées sur la figure II.2. Dans ce cas, le vecteur d'informations visuelles \mathbf{s} , constitué des coordonnées de quatre coins, est autorisé à passer sous l'objet occultant. Il est alors possible que le motif visuel se trouve de part et d'autre de ce dernier, ce qui peut conduire à un minimum local, comme le montre la figure II.2.d. En particulier, dans notre cas, ce choix est d'autant plus adéquat que notre système robotique permet à la caméra de se déplacer uniquement dans le plan horizontal.



II.2.a – Situation initiale. II.2.b – Passage en-dessous de \mathcal{O}_{cc} .

II.2.c – Risque de blocage.

II.2.d – Blocage !

FIG. II.2 – Possibilité de passage du motif visuel en-dessous de l'objet occultant.

II.1.2.2 Distances relatives à l'objet occultant

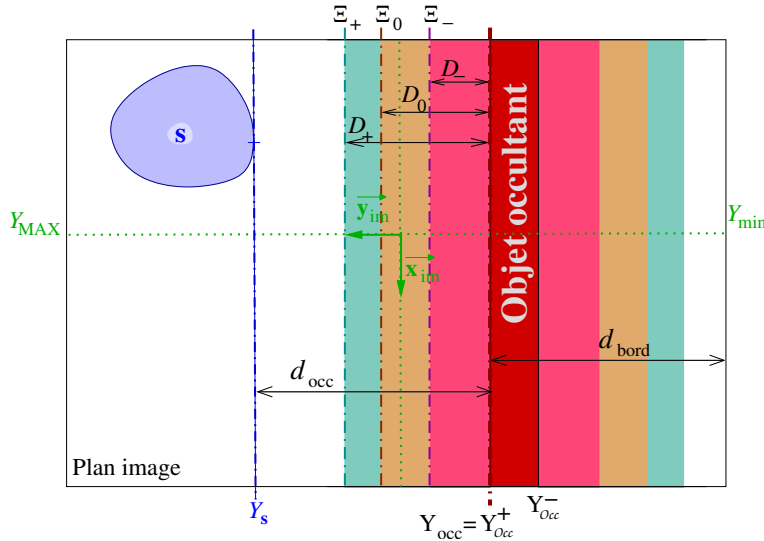


FIG. II.3 – Les distances relative à l'objet occultant.

Nous nous intéressons donc uniquement à la projection de l'objet occultant et du motif visuel sur l'axe des ordonnées \vec{y}_{im} . Afin de caractériser le risque d'occultation, nous définissons l'ordonnée Y_{occ} comme étant le bord de l'objet occultant le plus proche des informations visuelles (soit $Y_{occ} = Y_{occ}^+$ sur les figures II.1 et II.3). En l'absence d'objet occultant dans l'image Y_{occ} est définie par rapport à l'ordonnée Y_{min} ou Y_{max} du bord de l'image le plus proche de s . De cette manière, la perte du motif visuel s peut également être traitée comme une occultation. Pour construire nos stratégies d'évitement, nous introduisons maintenant les distances suivantes dans le plan image (voir figure II.3) :

- d_{occ} caractérise la *distance avant occultation*, c'est-à-dire la distance la plus courte entre l'ordonnée Y_{occ} et l'ordonnée Y_j de la composante du motif visuel le plus proche. La distance d_{occ} peut alors s'exprimer sous la forme suivante : $d_{occ} = \min_j |Y_j - Y_{occ}|$, qui peut se réduire en :

$$d_{occ} = |Y_s - Y_{occ}| \quad (II.1)$$

si Y_s représente l'ordonnée de la composante du motif visuel s la plus proche de l'objet occultant.

- d_{bord} caractérise la distance entre l'ordonnée Y_{occ} de l'objet occultant et le bord de l'image opposé au motif visuel s , soit :

$$d_{bord} = |Y_{occ} - Y_{bord}| \quad (II.2)$$

où Y_{bord} désigne l'ordonnée du bord de l'image vers lequel \mathcal{O}_{cc} doit sortir, c'est-à-dire Y_{max} ou Y_{min} . Si aucun objet occultant n'est présent dans l'image alors la distance d_{bord} est fixée à zéro.

- D_+ définit l'enveloppe Ξ_+ représentant la zone d'influence de l'objet occultant. Au-delà de cette limite on considérera qu'il n'y a pas de risque d'occultation.
- D_0 et D_- définissent respectivement les enveloppes Ξ_0 et Ξ_- qui délimitent les zones à l'intérieur desquelles le danger d'occultation devient important.

Ces différentes distances nous aideront à détecter et caractériser le risque d'occultation et donc à définir nos stratégies de navigation.

II.2 Évitement simultané d'occultations et de collisions

II.2.1 Synthèse d'un correcteur d'évitement d'occultations et de collisions

Notre objectif est ici de définir une tâche de navigation permettant à la fois d'éviter les collisions et les phénomènes d'occultations. Nous cherchons à exploiter le formalisme des tâches redondantes présenté à la section I.1.5.1. L'idée de ce formalisme est de définir une fonction de tâche principale redondante, notée $\mathbf{e}_{\text{occ}}(q)$, à réaliser en *priorité* (ici éviter les occultations), et une fonction de coût, notée $h_{\text{coll}}(q)$, caractérisant l'exécution d'un objectif *secondaire* (ici, éviter les collisions). Le formalisme des tâches redondantes permet alors de construire une fonction de tâche globale, notée $\mathbf{e}_o(q)$, telle que sa régulation à zéro minimise le critère $h_{\text{coll}}(q)$ sous la contrainte que la tâche primaire $\mathbf{e}_{\text{occ}}(q)$ soit parfaitement réalisée. La fonction de tâche globale $\mathbf{e}_o(q)$ garantissant l'exécution de la fonction de tâche redondante $\mathbf{e}_{\text{occ}}(q)$ est donnée par :

$$\mathbf{e}_o(q) = W_{\text{occ}}^+ \mathbf{e}_{\text{occ}}(q) + \beta_{\text{occ}} (\mathbb{I}_3 - W_{\text{occ}}^+ W_{\text{occ}}) g_{\text{coll}}$$

où $\beta_{\text{occ}} > 0$ est un gain scalaire, $W_{\text{occ}}^+ = W_{\text{occ}}^T (W_{\text{occ}} W_{\text{occ}}^T)^{-1}$ est la pseudo-inverse à droite de W_{occ} , et g_{coll} correspond au gradient du critère secondaire h_{coll} : $g_{\text{coll}} = \frac{\partial h_{\text{coll}}}{\partial q}$. Afin de satisfaire les conditions (I.13), la matrice W_{occ} est choisie égale à la jacobienne \mathbf{J}_{occ} de la tâche prioritaire $\mathbf{e}_{\text{occ}}(q)$ (cf. remarque I.3, page 35), soit : $W_{\text{occ}} = \mathbf{J}_{\text{occ}} = \frac{\partial \mathbf{e}_{\text{occ}}}{\partial q}$.

Nous nous proposons à présent de déterminer la tâche prioritaire $\mathbf{e}_{\text{occ}}(q)$, ainsi que le fonction de coût secondaire $h_{\text{coll}}(q)$ permettant d'éviter respectivement les occultations et les collisions. Nous proposons de définir $\mathbf{e}_{\text{occ}}(q)$ comme suit [FOLIO et CADENAT, ECMR2005] :

$$\mathbf{e}_{\text{occ}}(q) = \begin{pmatrix} \mathbf{e}_{\text{d}_{\text{occ}}} \\ \mathbf{e}_{\text{d}_{\text{bord}}} \end{pmatrix} \quad (\text{II.3})$$

où :

$$\begin{aligned} \mathbf{e}_{\text{d}_{\text{occ}}} &= \frac{1}{d_{\text{occ}}} - \frac{1}{D_{\text{max}}} \\ \mathbf{e}_{\text{d}_{\text{bord}}} &= d_{\text{bord}} \end{aligned} \quad (\text{II.4})$$

avec $D_{\text{max}} = \frac{1}{2} (Y_{\text{max}} - Y_{\text{min}})$ définissant une distance constante, correspondant à la valeur maximale que peut atteindre la distance d_{occ} dans l'image. La première composante $\mathbf{e}_{\text{d}_{\text{occ}}}$ permet de ramener le motif à une distance D_{max} de l'objet occultant dans l'image. Cette formulation présente l'avantage d'avoir la tâche d'évitement d'occultations active dans toute l'image. Ainsi, le terme $\mathbf{e}_{\text{d}_{\text{occ}}}$ augmente lorsque l'objet occultant \mathcal{O}_{cc} se rapproche du motif visuel \mathbf{s} , et devient infini lorsque la distance d_{occ} tend vers zéro. Inversement, $\mathbf{e}_{\text{d}_{\text{occ}}}$ décroît lorsque \mathcal{O}_{cc} s'éloigne de \mathbf{s} et s'annule lorsque $d_{\text{occ}} = D_{\text{max}}$, assurant que cette tâche est toujours définie. La seconde composante $\mathbf{e}_{\text{d}_{\text{bord}}}$ permet de faire sortir l'objet occultant de l'image, ce qui se produit lorsque la distance d_{bord} s'annule. Afin que le problème de commande soit bien posé, la régulation à zéro des deux composantes $\mathbf{e}_{\text{d}_{\text{occ}}}$ et $\mathbf{e}_{\text{d}_{\text{bord}}}$ de la fonction de tâche redondante $\mathbf{e}_{\text{occ}}(q)$ doit pouvoir être réalisée de manière *simultanée*. Pour cela, il est nécessaire que les deux tâches soient *compatibles*. Cette condition est remplie par construction par la définition des distances d_{occ} et d_{bord} .

La jacobienne \mathbf{J}_{occ} de la tâche prioritaire $\mathbf{e}_{\text{occ}}(q)$ s'écrit alors à partir de la relation (II.3) de la manière suivante :

$$\mathbf{J}_{\text{occ}} = \frac{\partial \mathbf{e}_{\text{occ}}}{\partial q} = \begin{pmatrix} \frac{\partial \mathbf{e}_{\text{d}_{\text{occ}}}}{\partial q}(\text{d}_{\text{occ}}) \\ \frac{\partial \mathbf{e}_{\text{d}_{\text{bord}}}}{\partial q}(\text{d}_{\text{bord}}) \end{pmatrix}$$

où $\frac{\partial \mathbf{e}_{\text{d}_{\text{occ}}}}{\partial q}(\text{d}_{\text{occ}})$ se déduit de (II.4) comme suit :

$$\frac{\partial \mathbf{e}_{\text{d}_{\text{occ}}}}{\partial q}(\text{d}_{\text{occ}}) = \frac{\partial}{\partial q} \left(\frac{1}{d_{\text{occ}}} - \frac{1}{D_{\text{max}}} \right) = -\frac{1}{d_{\text{occ}}^2} \frac{\partial d_{\text{occ}}}{\partial q}$$

et grâce à la définition de d_{occ} (II.1), il vient :

$$\frac{\partial d_{\text{occ}}}{\partial q} = \frac{\partial}{\partial q} |Y_{\text{s}} - Y_{\text{occ}}| = \varepsilon_{\text{occ}} \left(\frac{\partial Y_{\text{s}}}{\partial q} - \frac{\partial Y_{\text{occ}}}{\partial q} \right) \quad (\text{II.5})$$

avec $\varepsilon_{\text{occ}} = \text{sign}(Y_{\text{s}} - Y_{\text{occ}})$. De la même manière, la jacobienne $\frac{\partial \mathbf{e}_{\text{d}_{\text{bord}}}}{\partial q}(\text{d}_{\text{bord}})$ s'obtient en dérivant (II.2) :

$$\frac{\partial \mathbf{e}_{\text{d}_{\text{bord}}}}{\partial q}(\text{d}_{\text{bord}}) = \frac{\partial d_{\text{bord}}}{\partial q} = \frac{\partial}{\partial q} |Y_{\text{occ}} - Y_{\text{bord}}| = \varepsilon_{\text{bord}} \frac{\partial Y_{\text{occ}}}{\partial q} \quad (\text{II.6})$$

avec $\varepsilon_{\text{bord}} = \text{sign}(Y_{\text{occ}} - Y_{\text{bord}})$. Il reste, maintenant, à caractériser les matrices jacobiennes associées aux ordonnées $\frac{\partial Y_{\text{occ}}}{\partial q}$ et $\frac{\partial Y_{\text{s}}}{\partial q}$. À cet effet, rappelons que si Y est l'ordonnée d'un point sur le plan image, nous avons :

$$\frac{\partial Y}{\partial q} = \frac{\partial Y}{\partial r} \frac{\partial r}{\partial q} = \mathcal{L}(Y, z) \mathbf{J}(q)$$

où $\mathcal{L}(Y, z)$ et $\mathbf{J}(q)$ sont respectivement données par les relations (I.6) et (I.10). Toutefois nous avons montré que les jacobiennes $\mathcal{L}(Y, z)$ et $\mathbf{J}(q)$ se réduisent aux matrices \mathcal{L}_{r} et \mathbf{J}_{r} définies par les relations (I.7) et (I.11). Ainsi, les jacobiennes associées aux ordonnées $\frac{\partial Y_{\text{occ}}}{\partial q}$ et $\frac{\partial Y_{\text{s}}}{\partial q}$ s'écrivent :

$$\frac{\partial Y_{\text{occ}}}{\partial q} = \begin{pmatrix} -\frac{f}{z_{\text{occ}}} & \frac{Y_{\text{occ}}}{z_{\text{occ}}} & \left(f + \frac{Y_{\text{occ}}^2}{f} \right) \end{pmatrix} \mathbf{J}_{\text{r}} = \mathcal{L}_{Y_{\text{occ}}} \mathbf{J}_{\text{r}} \quad (\text{II.7})$$

$$\frac{\partial Y_{\text{s}}}{\partial q} = \begin{pmatrix} -\frac{f}{z_{\text{s}}} & \frac{Y_{\text{s}}}{z_{\text{s}}} & \left(f + \frac{Y_{\text{s}}^2}{f} \right) \end{pmatrix} \mathbf{J}_{\text{r}} = \mathcal{L}_{Y_{\text{s}}} \mathbf{J}_{\text{r}} \quad (\text{II.8})$$

Nous pouvons alors réécrire les relations (II.5) et (II.6) simplement sous la forme :

$$\frac{\partial d_{\text{occ}}}{\partial q} = \frac{\partial}{\partial q} |Y_{\text{s}} - Y_{\text{occ}}| = \varepsilon_{\text{occ}} (\mathcal{L}_{Y_{\text{s}}} - \mathcal{L}_{Y_{\text{occ}}}) \mathbf{J}_{\text{r}} \quad (\text{II.9})$$

$$\frac{\partial d_{\text{bord}}}{\partial q} = \frac{\partial}{\partial q} |Y_{\text{occ}} - Y_{\text{bord}}| = \varepsilon_{\text{bord}} \mathcal{L}_{Y_{\text{occ}}} \mathbf{J}_{\text{r}} \quad (\text{II.10})$$

Finalement, la matrice jacobienne \mathbf{J}_{occ} de la tâche redondante $\mathbf{e}_{\text{occ}}(q)$ se détermine de la manière suivante :

$$\mathbf{J}_{\text{occ}} = \frac{\partial \mathbf{e}_{\text{occ}}}{\partial q} = \begin{pmatrix} -\frac{\varepsilon_{\text{occ}}}{d_{\text{occ}}^2} (\mathcal{L}_{Y_{\text{s}}} - \mathcal{L}_{Y_{\text{occ}}}) \\ \varepsilon_{\text{bord}} \mathcal{L}_{Y_{\text{occ}}} \end{pmatrix} \mathbf{J}_{\text{r}} \quad (\text{II.11})$$

Remarque II.1: Le calcul des matrices d'interaction \mathcal{L}_{Y_s} et $\mathcal{L}_{Y_{occ}}$ requiert une mesure ou un modèle de leurs profondeurs respectives z_s et z_{occ} (cf. remarque I.1). Celles-ci peuvent être déterminées, par exemple¹, en utilisant des capteurs spécifiques (laser, système de stéréo-vision...), des prises de vue multiples, la dynamique de l'image, etc. Cela suppose alors, que l'on dispose de suffisamment d'informations permettant de caractériser efficacement z_s et z_{occ} , avant qu'un risque d'occultation ne se produise (e.g. avant que $d_{occ} < D_+$). Dans le cas où il est difficile de disposer de ces données, il est nécessaire de considérer des approximations des matrices $\hat{\mathcal{L}}_{Y_s}$ et $\hat{\mathcal{L}}_{Y_{occ}}$. La matrice W_{occ} est alors définie par une approximation de la jacobienne $\hat{\mathbf{J}}_{occ}$, donnée par :

$$W_{occ} = \hat{\mathbf{J}}_{occ} = \begin{pmatrix} -\frac{\varepsilon_{occ}}{d_{occ}^2} (\hat{\mathcal{L}}_{Y_s} - \hat{\mathcal{L}}_{Y_{occ}}) \\ \varepsilon_{bord} \hat{\mathcal{L}}_{Y_{occ}} \end{pmatrix} \mathbf{J}_r$$

en rappelant que les conditions (I.13) suivantes doivent être remplies :

$$\ker(\hat{\mathbf{J}}_{occ}) = \ker(\mathbf{J}_{occ}) \quad \text{et} \quad (\mathbf{J}_{occ} \hat{\mathbf{J}}_{occ}^T) > 0$$

Remarque II.2: Lorsqu'il n'y a pas d'objet occultant \mathcal{O}_{cc} dans l'image, et que la distance d_{occ} est définie par rapport aux bords de l'image, la profondeur z_{occ} est alors donnée par la profondeur du motif visuel z_s , soit : $z_{occ} = z_s$

Il reste à présent à définir la tâche secondaire h_{coll} permettant d'éviter les collisions. Ici, nous avons choisi d'exploiter le formalisme du suivi de chemin [SAMSON, 1992], présenté au paragraphe I.2.3. Rappelons que le système robotique, par le biais des capteurs ultrasoniques embarqués, peut caractériser la distance d_{coll} et l'angle α , exprimés dans un repère de Frenet \mathcal{R}'_M (voir figure I.15, page 46). Le suivi de chemin consiste alors à amener le véhicule à suivre une enveloppe de contournement ξ_0 située à une distance d_0 de l'obstacle, en régulant l'erreur de positionnement latéral $\delta = (d_{coll} - d_0)$ et l'erreur d'orientation $\theta_e = \alpha$. Dans ce formalisme, l'objectif est d'assurer la convergence géométrique du centre M du robot vers le chemin à suivre ξ_0 sous l'hypothèse que la vitesse linéaire v ne s'annule pas. De ce fait, dans notre cas, nous devons simplement établir un critère h_{coll} dont la minimisation permet la régulation du couple (δ, α) à zéro. À cet effet, nous proposons le critère quadratique suivant [FOLIO et CADENAT, ECMR2005] :

$$h_{coll} = \frac{1}{2}(\delta + \kappa\alpha)^2, \quad \forall \kappa \in \mathbb{R}^* \quad (\text{II.12})$$

où κ est un gain scalaire non nul qui permet de déterminer la vitesse relative de convergence des erreurs de position δ et d'orientation α . En effet, le terme $\delta + \kappa\alpha$ peut être vu comme une variable de glissement (cf. paragraphe I.3.2) dont la régulation à zéro assure celle de δ et α ². Ainsi, le critère h_{coll} (II.12) garantit le contournement des obstacles pendant l'évitement d'occultations, du moins tant que la réalisation de la tâche prioritaire

1. Les différents moyens d'obtention de la profondeur sont détaillés au chapitre III dans le paragraphe III.2.1.

2. Rappelons que le lecteur intéressé pourra trouver une preuve détaillée de ce résultat dans [CADENAT, 1999] et [SOUÈRES *et al.*, 1998].

$\mathbf{e}_{\text{occ}}(q)$ le permet. Enfin, le gradient g_{coll} de la tâche secondaire h_{coll} se déduit simplement de (II.12) comme suit :

$$g_{\text{coll}} = \frac{\partial h_{\text{coll}}}{\partial q} = (\delta + \kappa\alpha) \begin{pmatrix} \sin \alpha - \kappa \frac{\bar{\chi} \cos \alpha}{1 + \sigma \bar{\chi} \delta} \\ \kappa \\ 0 \end{pmatrix}, \quad \text{avec } \bar{\chi} = \frac{\sigma}{R}$$

en rappelant que la dynamique du couple (δ, α) par rapport au repère \mathcal{R}'_M est donnée par la relation (I.27) (*cf.* paragraphe I.2.3, page 45).

Nous venons de définir la fonction de tâche $\mathbf{e}_o(q)$ qui garantit à la fois l'évitement d'occultations et de collision. Il nous reste, désormais, à synthétiser la loi de commande \dot{q}_o régulant $\mathbf{e}_o(q)$ à zéro. Dans la mesure où la matrice $W_{\text{occ}} = \mathbf{J}_{\text{occ}}$ est choisie de manière à satisfaire les conditions (I.13), le jacobien $\mathbf{J}_o = \frac{\partial \mathbf{e}_o}{\partial q}$ de la fonction de tâche $\mathbf{e}_o(q)$ est une matrice définie positive, et $\mathbf{e}_o(q)$ est ρ -admissible. La synthèse de l'asservissement peut alors se simplifier comme indiquée en (I.15) [ESPIAU *et al.*, 1992] (*cf.* section I.1.5.1, page 35), en appliquant ce résultat à notre cas il vient :

$$\dot{q}_o = -\lambda_o \mathbf{e}_o(q) \quad \forall \lambda_o > 0 \quad (\text{II.13})$$

Nous avons donc défini une loi de commande \dot{q}_o d'évitement *simultané* d'occultations et de collisions. Nous proposons de valider ce correcteur dans le cadre de la réalisation d'une tâche de navigation référencée vision et d'évitement simultané d'occlusions et de collisions.

II.2.2 Stratégie de commande

Notre intention est de synthétiser une loi de commande permettant à un robot mobile de réaliser une tâche référencée vision dans un environnement encombré d'obstacles présentant un éventuel risque de collision et/ou d'occultation. À cet effet, il est nécessaire de combiner l'asservissement visuel (I.21), à la commande d'évitement simultané d'occultation et de collision (II.13) que nous venons d'introduire. Ainsi, de la même manière qu'au paragraphe I.3, la stratégie adoptée ici consiste à réaliser la tâche référencée vision lorsqu'il n'y a pas d'obstacle dans le voisinage du robot, et à basculer sur la commande (II.13) lorsqu'un risque d'occultation ou de collision se produit. Il est alors possible de définir la loi de commande globale permettant de réaliser la tâche de navigation désirée de la manière suivante :

$$\dot{q} = (1 - \mu)\dot{q}_{AV} + \mu\dot{q}_o \quad \forall \mu \in [0; 1] \quad (\text{II.14})$$

où \dot{q} représente la commande en vitesse appliquée au robot ; \dot{q}_{AV} et \dot{q}_o correspondent respectivement aux correcteurs définis par (I.21) et (II.13). La fonction μ permet de lisser le basculement d'un correcteur à l'autre en fonction de l'environnement local du robot. Elle dépend donc de la distance à l'obstacle d_{coll} mesurée dans l'environnement, et de la distance à l'objet occultant d_{occ} évaluée dans l'image. Afin de remplir nos objectifs de commande, nous souhaitons mettre en œuvre la stratégie de basculement décrite dans le tableau II.1.

	$\mu_{\text{occ}} = 0$	$\mu_{\text{occ}} = 1$
$\mu_{\text{coll}} = 0$	$\mu = 0 \Rightarrow \dot{q} = \dot{q}_{AV}$ aucun risque	$\mu = 1 \Rightarrow \dot{q} = \dot{q}_O$ risque d'occultation seule
$\mu_{\text{coll}} = 1$	$\mu = 1 \Rightarrow \dot{q} = \dot{q}_O$ risque de collision seule	$\mu = 1 \Rightarrow \dot{q} = \dot{q}_O$ risque d'occultation et de collision

TAB. II.1 – Stratégie d'évolution de la fonction de basculement μ .

La fonction μ peut alors s'exprimer comme suit [FOLIO et CADENAT, ECMR2005] :

$$\mu = (1 - \mu_{\text{occ}})\mu_{\text{coll}} + \mu_{\text{occ}} \quad \forall \mu_{\text{occ}}, \mu_{\text{coll}} \in [0; 1] \quad (\text{II.15})$$

où μ_{occ} et μ_{coll} caractérisent respectivement les risques d'occultations et de collisions. Ces paramètres sont définis par :

$$\begin{cases} \mu_{\text{occ}} = 0 & \text{si } d_{\text{occ}} > D_+ \\ \mu_{\text{occ}} = 1 & \text{si } d_{\text{occ}} < D_- \\ \mu_{\text{occ}} = \frac{D_+ - d_{\text{occ}}}{D_+ - D_-} & \text{sinon} \end{cases} \quad (\text{II.16})$$

$$\begin{cases} \mu_{\text{coll}} = 0 & \text{si } d_{\text{coll}} > d_+ \\ \mu_{\text{coll}} = 1 & \text{si } d_{\text{coll}} < d_- \\ \mu_{\text{coll}} = \frac{d_+ - d_{\text{coll}}}{d_+ - d_-} & \text{sinon} \end{cases} \quad (\text{II.17})$$

Ainsi, lorsqu'il n'y a pas de risque d'occultation ni de collision ($\mu_{\text{occ}} = 0$ et $\mu_{\text{coll}} = 0$), alors μ est fixé à zéro et seul l'asservissement visuel \dot{q}_{AV} est appliqué au robot. Quand un objet occultant franchit l'enveloppe délimitée par Ξ_+ (voir la figure II.3, page 60), le risque d'occultation μ_{occ} croît progressivement vers 1. Si dans le même temps, il n'y a aucun risque de collision, alors μ est déterminé uniquement à partir de μ_{occ} . Lorsqu'un obstacle est détecté dans le voisinage du véhicule (*i.e.* $d_{\text{coll}} < d_+$), le risque de collision μ_{coll} augmente pour atteindre 1 quand le risque de collision est maximal (*i.e.* $d_{\text{coll}} < d_-$). Durant cette phase, la valeur de μ dépend à la fois des risques de collision μ_{coll} et d'occultation μ_{occ} , et le robot est commandé par une combinaison linéaire de \dot{q}_{AV} et de \dot{q}_O . Enfin si les risques d'occultations ou de collisions deviennent maximal, alors seule la commande \dot{q}_O est appliquée au véhicule.

Finalement, nous avons défini un schéma de commande référencé multi-capteurs. En effet, celui-ci est basé d'une part sur l'exploitation des données perceptuelles pour effectuer la tâche référencée vision considérée en évitant les occultations; et d'autre part, sur les mesures fournies par les capteurs ultrasoniques embarqués pour éviter les collisions. De plus, la stratégie de commande devient également multi-capteurs puisque le basculement entre les deux correcteurs \dot{q}_{AV} et \dot{q}_O se fait à partir de mesures effectuées à la fois dans l'image et dans l'environnement local du robot.

II.2.3 Résultats

Dans le but d'illustrer et de valider la méthode proposée, la loi de commande (II.14) a été implantée sous *Matlab*. Rappelons que la période d'échantillonnage a été choisie

égale à 150 ms, valeur correspondant à notre plate-forme expérimentale³. Nous proposons ci-après deux exemples de simulation.

II.2.3.1 Exemple : évitement de deux obstacles cylindriques

L'objectif de la mission est ici de positionner la caméra embarquée face à un amer visuel composé de quatre points caractéristiques [FOLIO et CADENAT, ECMR2005]. Le robot évolue dans un environnement encombré de deux obstacles cylindriques susceptibles d'occulter ou de gêner la base mobile. Les distances des enveloppes D_- et D_+ ont été fixées respectivement à 30 et 80 pixels, et d_+ , d_0 , d_- à 0.6 m, 0.5 m et 0.4 m. Les figures II.4, II.5, II.6, II.7 et II.8 présentent les résultats obtenus.

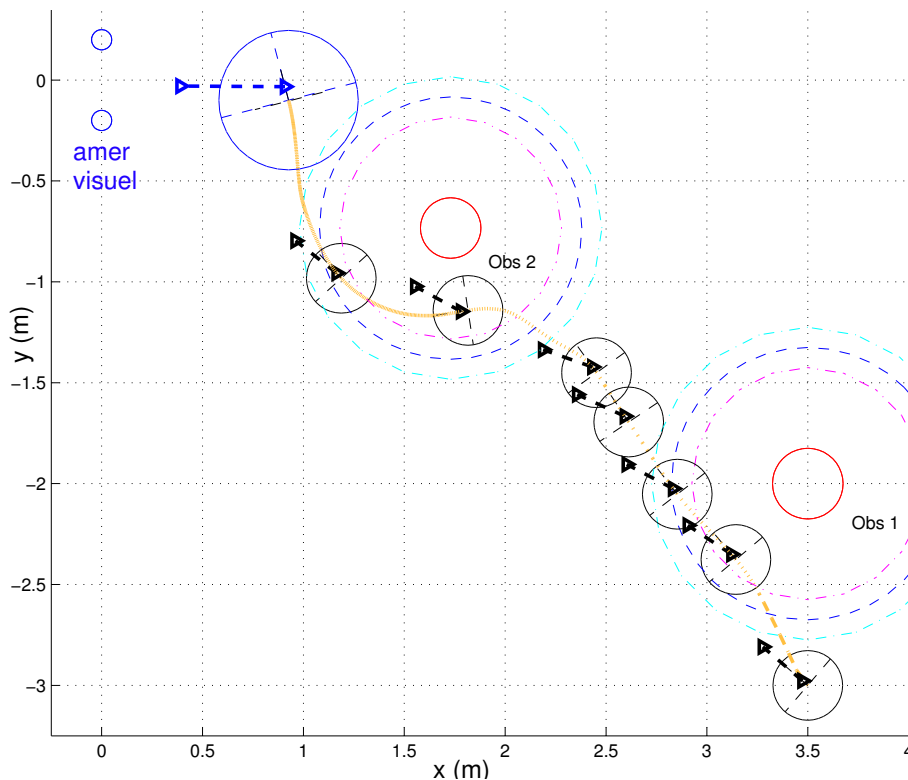


FIG. II.4 – Simulation : évitement de deux obstacles cylindriques.

Comme représenté sur les figures II.4 et II.5.a, au début de la tâche il n'y a pas de risque d'occultation ($\mu_{occ} = 0$, cf. figure II.7.a) ni de collision ($\mu_{coll} = 0$, cf. figure II.6.a), et μ est fixé à zéro (cf. figure II.8). Le robot est donc commandé uniquement par l'asservissement visuel \dot{q}_{AV} . Puis le véhicule entre dans le voisinage du premier obstacle, et le risque de collision μ_{coll} croît, tandis que μ_{occ} reste fixé à 0. La fonction de basculement μ augmente alors pour atteindre sa valeur maximale 1, et la commande globale \dot{q} bascule progressivement vers \dot{q}_O qui permet d'éviter l'obstacle (cf. figure II.4). Une fois le premier obstacle dépassé la fonction μ est ramenée de 1 à 0, et la loi de commande commute à nouveau vers l'asservissement visuel \dot{q}_{AV} . Toutefois, la réalisation de la tâche référencée vision contraint le motif visuel à se rapprocher de l'objet occultant⁴, comme le montre

3. Cette période d'échantillonnage correspond à la période de l'asservissement et du processus d'extraction des mesures perceptuelles (cf. chapitre IV).

4. L'objet occultant est ici la projection du second obstacle. Celui-ci induit à cet instant un risque d'occultation, mais pas de collision.

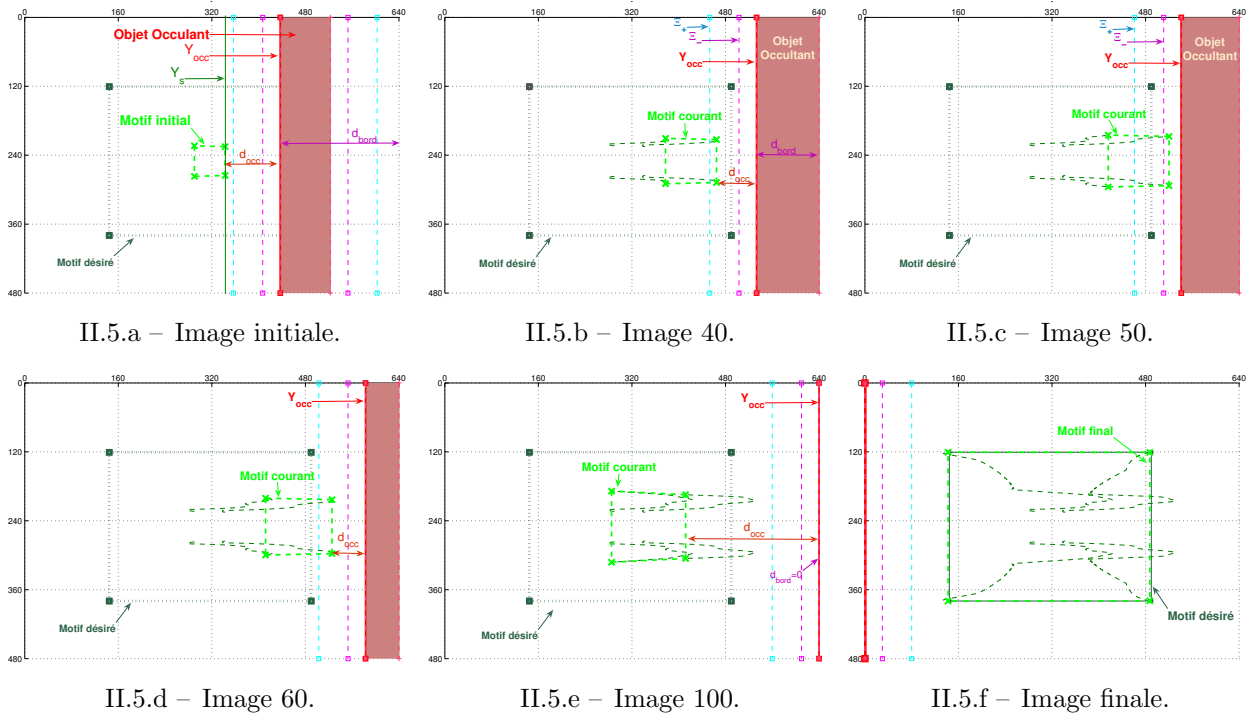


FIG. II.5 – Le plan image.

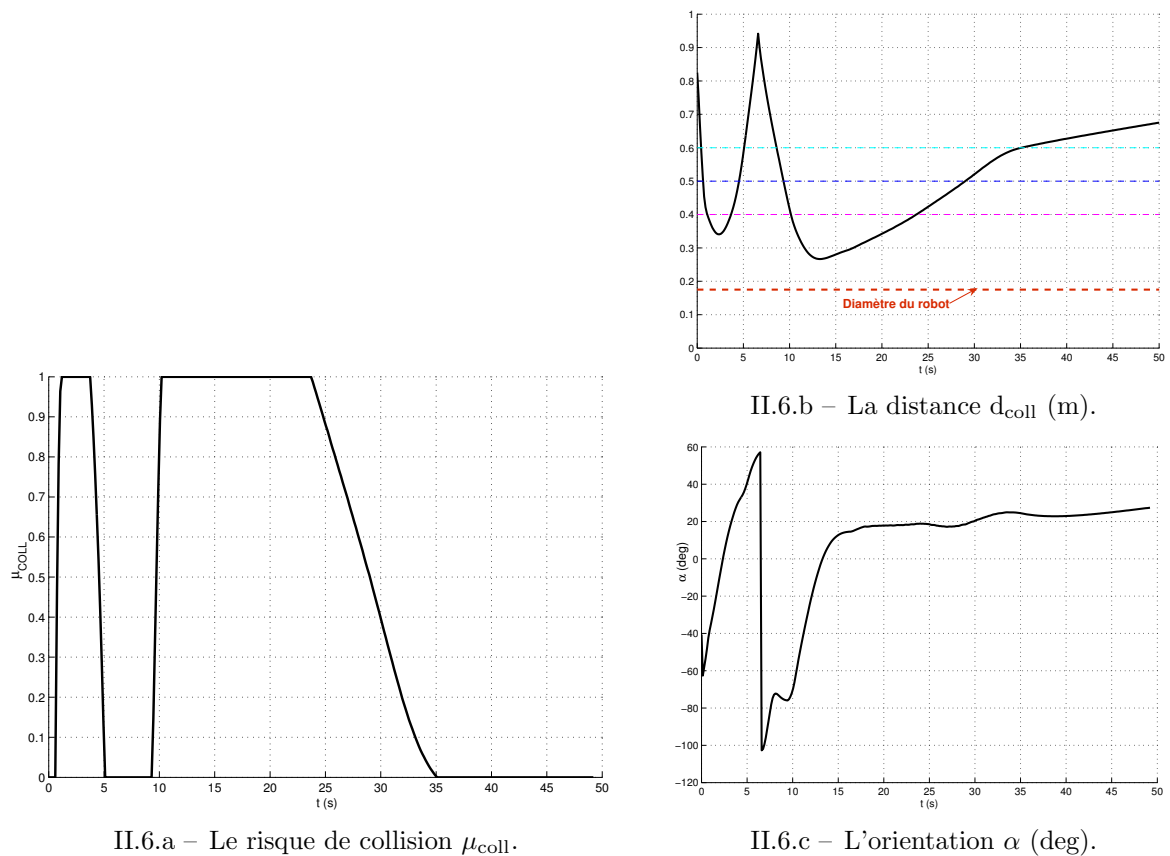
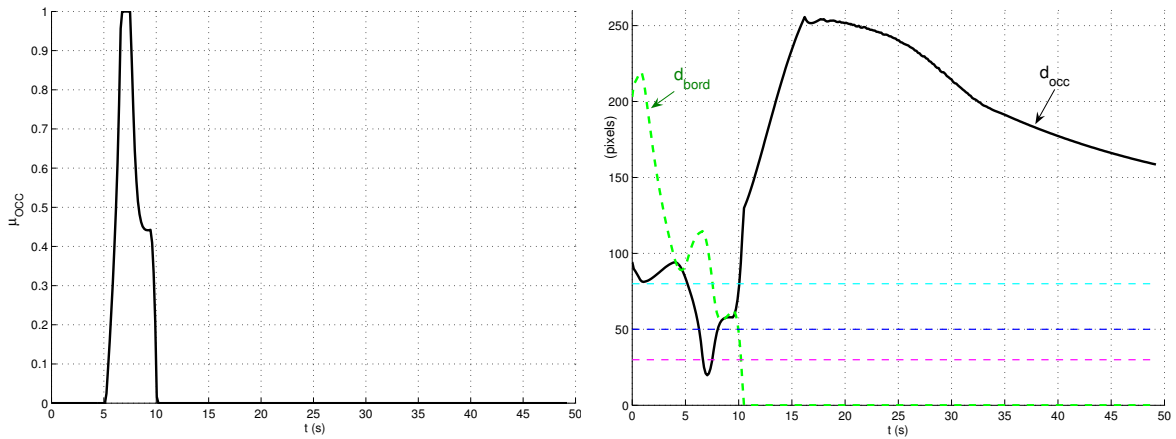


FIG. II.6 – Les mesures du risque de collision.

la séquence d'images proposée sur la figure II.5, et un danger d'occultation se produit (voir figure II.7). Le risque de collision μ_{coll} étant fixé à 0, $\mu = \mu_{occ}$ augmente pour



II.7.a – Le risque d’occultation μ_{occ} .

II.7.b – Les distances d_{occ} et d_{bord} (pixel).

FIG. II.7 – Les mesures du risque d’occultations.

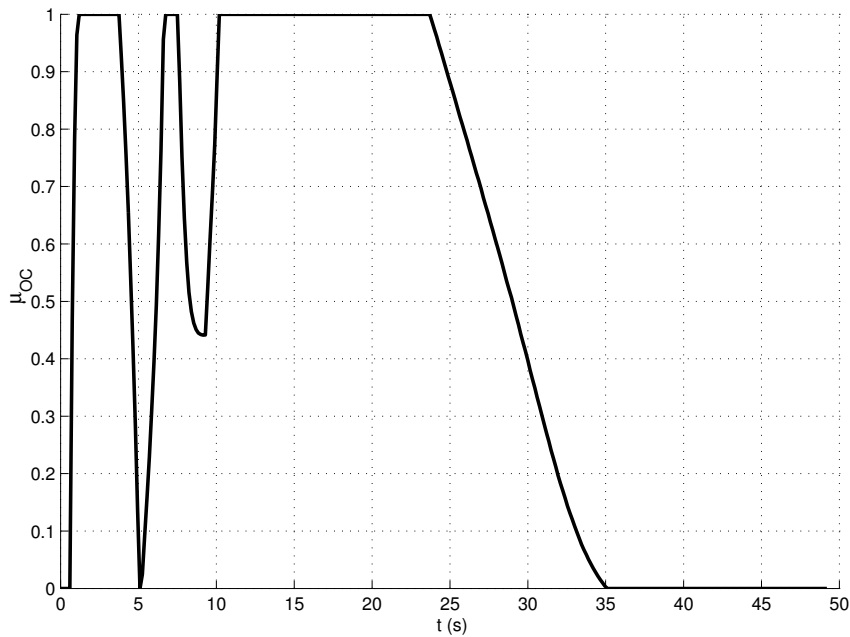


FIG. II.8 – Évolution de la fonction de basculement μ .

atteindre la valeur 1. La commande globale \dot{q} bascule à nouveau vers \dot{q}_o ce qui permet de progressivement de diminuer le risque d’occultation μ_{occ} et donc μ . Cependant avant d’éliminer complètement ce dernier, le robot entre dans le voisinage du second obstacle, et le danger de collision fait croître à nouveau μ vers 1. La commande d’évitement \dot{q}_o continue donc à être appliquée au véhicule qui parvient finalement à contourner le dernier obstacle, et à converger parfaitement vers la situation désirée \mathbf{s}^* comme le montre la figure II.5.f.

II.2.3.2 Exemple : évitement de deux murs

La mission consiste à présent à positionner la caméra embarquée face à un amér visuel dans un environnement encombré de deux murs [FOLIO et CADENAT, JNRR2005]. La position de ces derniers, celle de l'amér visuel, ainsi que la configuration initiale du robot ont été choisies de manière à provoquer des situations de collisions et d'occultations. Les distances des enveloppes D_- et D_+ ont été fixées respectivement à 30 et 80 pixels, et d_+ , d_0 , d_- à 0.7 m, 0.5 m et 0.4 m.

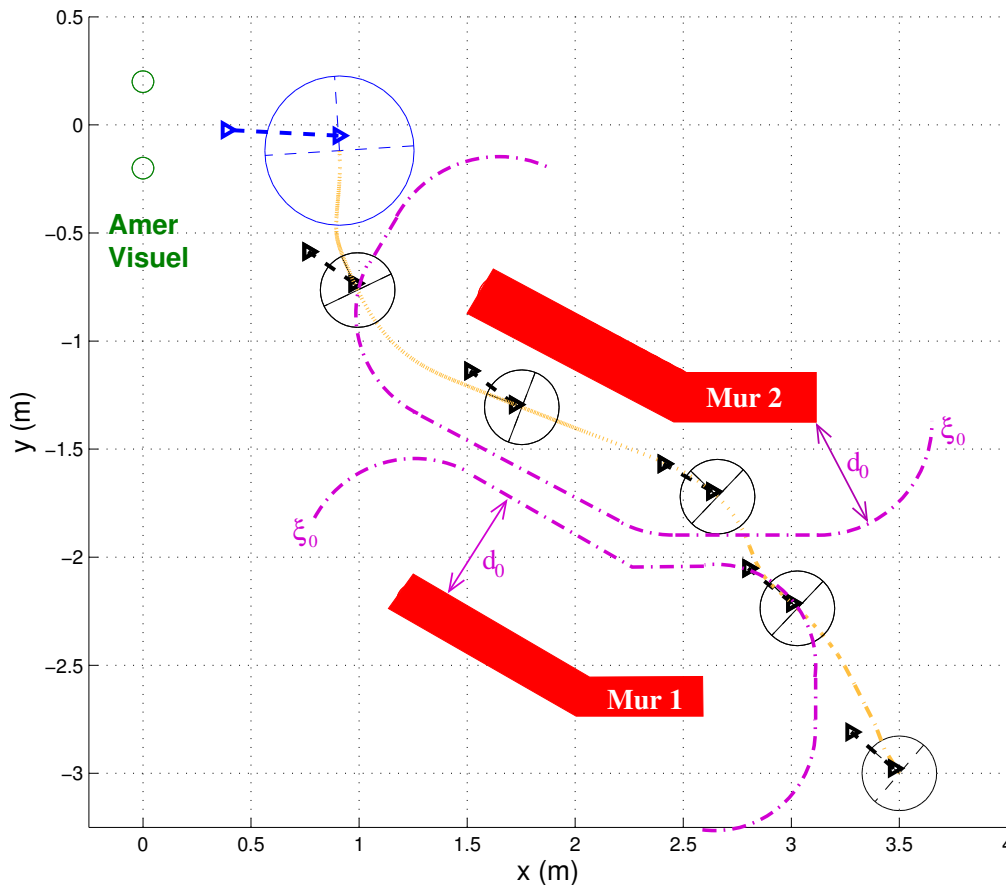


FIG. II.9 – Asservissement visuel et évitement simultané d'occlusions et de collisions.

Au début de la tâche il n'y a ni risque d'occultation, ni de collision, le robot est alors, dans un premier temps, commandé uniquement par l'asservissement visuel \dot{q}_{AV} . Toutefois, le déplacement de la caméra contraint rapidement le motif visuel \mathbf{s} à se rapprocher de l'objet occultant⁵, le risque d'occultation μ_{occ} commence par augmenter légèrement, et par conséquent la fonction de basculement μ également. Le robot est alors commandé par une combinaison linéaire des correcteurs \dot{q}_{AV} et \dot{q}_O . Cependant, le mouvement induit amène le véhicule à entrer dans le voisinage du mur 1. Le risque de collision μ_{coll} croît à son tour légèrement. Le système robotique cherche à réaliser la tâche de navigation référencée vision tout évitant les occultations, et dans la mesure où cela est possible les collisions également. La combinaison linéaire des commandes \dot{q}_{AV} et \dot{q}_O parvient à éloigner le véhicule du mur 1, et à réduire le risque d'occultation. Mais, ce faisant, la base mobile entre dans le

5. Ici, l'objet occultant correspond à la projection du mur 2 sur le plan image.

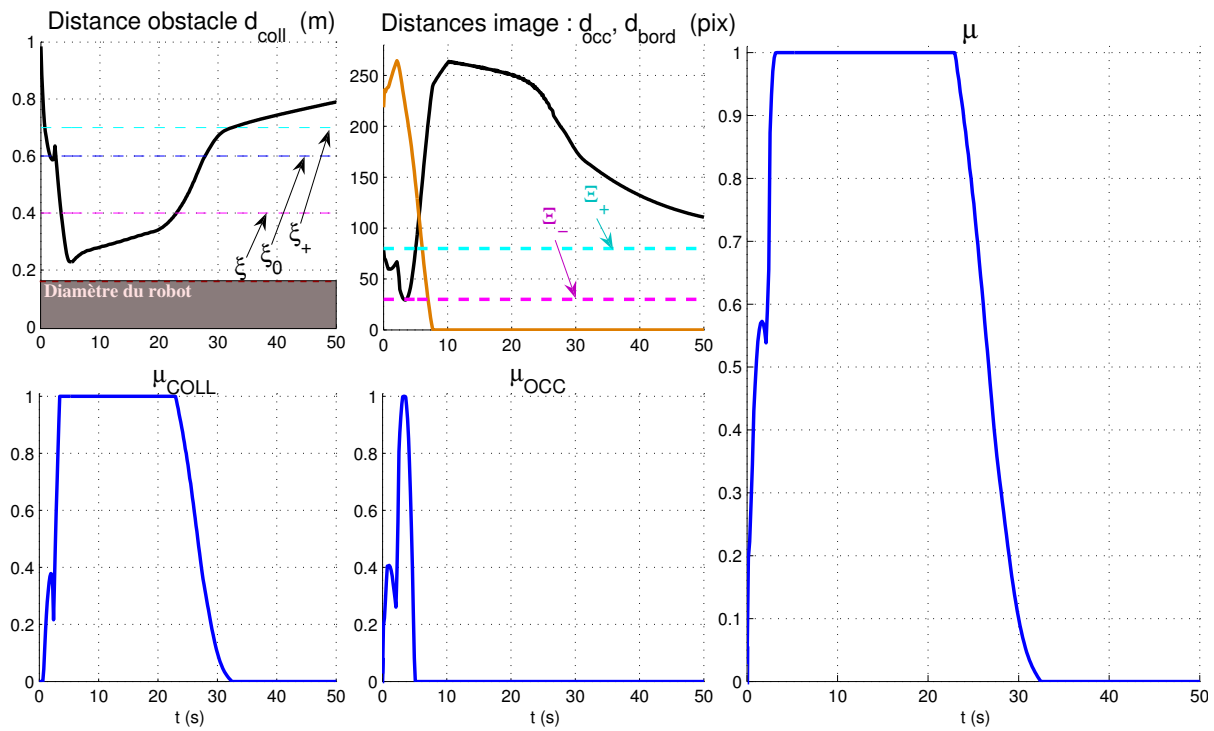


FIG. II.10 – Données d'évitement.

voisinage du mur 2. Les risques d'occultation μ_{occ} et de collision μ_{coll} croissent à nouveau tous deux, donc la fonction de basculement μ augmente aussi, et la commande globale \dot{q} bascule progressivement vers \dot{q}_O . Ainsi, celle-ci permet d'éviter parfaitement l'occultation, mais contraint le robot à se rapprocher dangereusement du mur 2, la non-collision n'étant assurée qu'au mieux (cf. figure II.10). Une fois le mur 2 dépassé le danger de collision μ_{coll} décroît, et la commande bascule à nouveau vers la commande d'asservissement visuel \dot{q}_{AV} . Finalement, le robot parvient à converger parfaitement vers la situation désirée \mathbf{s}^* .

II.2.4 Bilan

Nous avons ainsi défini un schéma de commande référencée multi-capteurs intégrant à la fois les données perceptuelles définies dans l'image, et les mesures proximétriques fournies par les capteurs ultrasoniques embarqués. Les premiers permettent de gérer l'exécution de la tâche de navigation basée sur la vision et l'évitement d'occultation, tandis que les secondes sont exploitées pour tenter d'assurer la non-collision. Cette méthode apparaît comme une première réponse au problème de gestion *simultanée* des occultations et des collisions. Toutefois, son principal inconvénient est que l'évitement d'obstacles n'est réalisé ici qu'*au mieux*. La non collision ne peut donc être pleinement garantie et le véhicule peut ainsi être amené à passer dans un voisinage très proche des obstacles, comme cela s'est produit dans le second exemple au paragraphe II.2.3.2. C'est pourquoi nous proposons une seconde méthode permettant d'améliorer la qualité du contournement.

II.3 Évitement de collision et d'occultation

La stratégie de commande mise œuvre à la section II.2 précédente ne permet pas d'assurer la non collision qu'au mieux. Notre objectif est, à présent, de définir une nouvelle approche d'évitement des collisions et des phénomènes d'occultations de manière *indépendante* afin de garantir la sécurité du robot au voisinage des obstacles. Pour cela, il est nécessaire de bien distinguer les trois situations suivantes :

- Cas ①** : Le même obstacle génère un danger de collision et d'occultation. Un risque d'occultation peut donc se produire en même temps qu'un risque de collision (*cf.* figure II.11.a) ;
- Cas ②** : Les risques de collision et d'occultations sont générés par deux obstacles distincts. Le phénomène d'occultation n'est donc pas forcément lié à l'obstacle que le robot est en train de contourner (*cf.* figure II.11.b) ;
- Cas ③** : Un obstacle peut être détecté dans l'image bien avant de devenir dangereux pour la base mobile. La présence d'un objet occultant dans l'image n'implique pas forcément qu'un risque de collision se présente (*cf.* figure II.11.c).

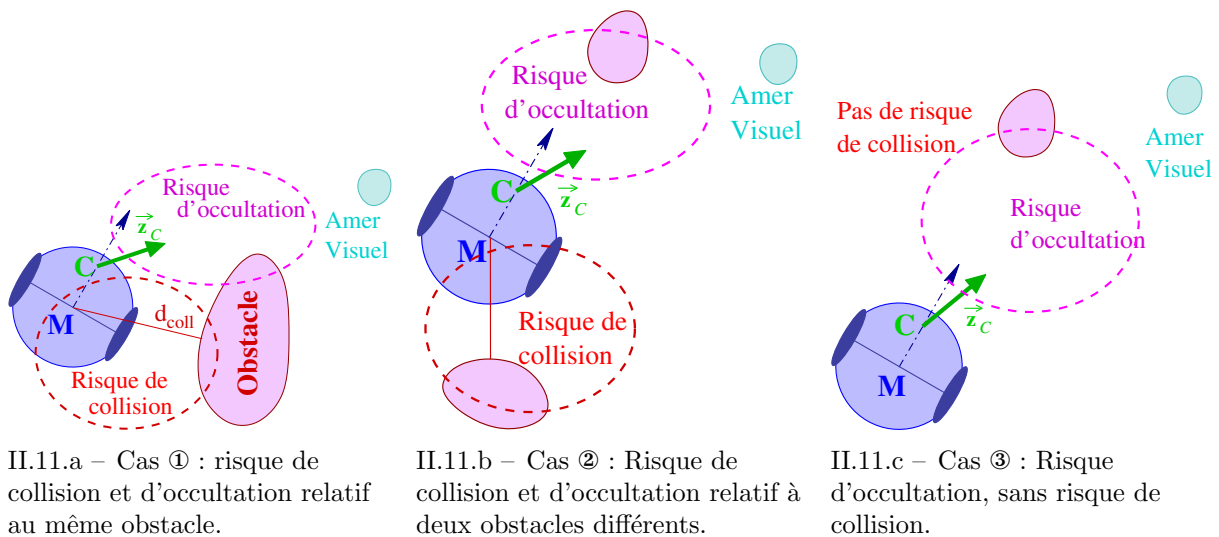


FIG. II.11 – Différents types d'occultations.

Il y a donc différents types d'occultations qu'il faut pouvoir détecter et gérer. L'idée est de définir deux tâches de navigation distinctes :

1. la première, notée $e_{co}(q)$, assurant la gestion conjointe des occultations et des collisions dans le voisinage des obstacles, avec comme impératif de garantir la sécurité du robot. Elle répond aux situations ① et ②.
2. la seconde, notée $e_{os}(q)$, contrôlant la non occultation du motif visuel lorsqu'il n'y a pas de risque de collision. Elle répond aux phénomènes d'occultations liés au cas ③.

De la même manière que précédemment, nous exploitons le formalisme des tâches redondantes présenté à la section I.1.5.1, pour définir les tâches de navigation $e_{co}(q)$ et $e_{os}(q)$. Ainsi, nous présentons d'abord la tâche garantissant la non-collision $e_{co}(q)$, puis la tâche d'évitement d'occultation $e_{os}(q)$, avant de décrire la stratégie de commande globale.

II.3.1 Tâche d'évitement de collision $e_{CO}(q)$

Nous souhaitons dans ce paragraphe déterminer la fonction de tâche $e_{CO}(q)$. Pour cela, nous proposons d'une part de définir une fonction de tâche redondante prioritaire, notée $e_{\text{coll}}(q)$, dédiée à l'évitement d'obstacles, et d'autre part de déterminer l'objectif secondaire $h_{\text{occ}}(q)$ de manière à éviter au mieux les occultations *au voisinage de ces derniers*. En appliquant le formalisme des tâches redondantes la fonction de tâche globale $e_{CO}(q)$ s'écrit :

$$e_{CO}(q) = \mathbf{J}_{\text{coll}}^+ e_{\text{coll}}(q) + \beta_{\text{coll}} (\mathbb{I}_3 - \mathbf{J}_{\text{coll}}^+ \mathbf{J}_{\text{coll}}) g_{\text{occ}} \quad \text{avec} \quad \begin{cases} \beta_{\text{coll}} > 0 \\ W_{\text{coll}} = \mathbf{J}_{\text{coll}} = \frac{\partial e_{\text{coll}}}{\partial q} \\ g_{\text{occ}} = \frac{\partial h_{\text{occ}}}{\partial q} \end{cases} \quad (\text{II.18})$$

Ainsi, cette fonction de tâche $e_{CO}(q)$ répond à l'impératif de sécurité du robot, que nous avons formulé précédemment, tout en traitant les occultations de type ① et ② qui peuvent se produire durant la tâche de contournement.

Déterminons dans un premier temps la fonction de tâche prioritaire $e_{\text{coll}}(q)$. Comme précédemment, la stratégie d'évitement de collision s'appuie sur les mesures d_{coll} et α fournies par les ultrasons disponibles sur le robot (voir la figure I.15, page 46). L'idée consiste à reprendre les principes développés au paragraphe I.3.2 (page 50), où il s'agissait d'allier le formalisme des fonctions de tâches [SAMSON *et al.*, 1991] et le formalisme du suivi de chemin [SAMSON, 1992]. Nous avons alors montré que la fonction de tâche $e_{\text{ch}}(q)$ (I.37) permet à la base mobile d'éviter l'obstacle en suivant une enveloppe de contournement ξ_0 avec une vitesse $v_0 \neq 0$. De plus, cette fonction de tâche est redondante puisqu'elle ne contraint que deux degrés de libertés sur les trois disponibles de notre système robotique. Elle est donc parfaitement adéquat à notre problème, et il vient [FOLIO et CADENAT, ECC-CDC2005, ROBIO2006] :

$$e_{\text{coll}}(q) = e_{\text{ch}}(q) = \begin{pmatrix} l - v_0 t \\ \delta + \kappa \alpha \end{pmatrix} \quad (\text{II.19})$$

La matrice jacobienne $\mathbf{J}_{\text{coll}} = \frac{\partial e_{\text{coll}}}{\partial q}$ de la tâche prioritaire $e_{\text{coll}}(q)$ se déduit simplement de la dynamique (I.27) du couple (δ, α) par rapport au repère de Frenet \mathcal{R}'_M (voir paragraphe I.2.3), comme suit :

$$\mathbf{J}_{\text{coll}} = \begin{pmatrix} 1 & 0 & 0 \\ \sin \alpha - \kappa \bar{\chi} & \kappa & 0 \end{pmatrix}$$

Il reste à définir la tâche secondaire minimisant le risque d'occultation. Nous proposons pour cela le critère $h_{\text{occ}} = \frac{1}{d_{\text{occ}}}$ qui diminue lorsque la distance d_{occ} croît, et inversement. Nous déduisons au moyen de la relation (II.9) le gradient g_{occ} associé :

$$g_{\text{occ}} = \frac{\partial h_{\text{occ}}}{\partial q} = -\frac{1}{d_{\text{occ}}^2} \frac{\partial d_{\text{occ}}}{\partial q} = -\frac{\varepsilon_{\text{occ}}}{d_{\text{occ}}^2} ((\mathcal{L}_{Y_s} - \mathcal{L}_{Y_{\text{occ}}}) \mathbf{J}_r)^T$$

Cette fonction de coût h_{occ} permet de minimiser le risque d'occultation ou de perte du motif visuel en phase de contournement. De plus, puisque la distance d_{occ} (II.1) est définie en fonction du bord de l'image le plus proche lorsqu'il n'y a pas d'objet occultant dans l'image, h_{occ} permet d'assurer la visibilité du motif visuel \mathbf{s} (*cf.* paragraphe I.3.1, page 48).

Nous avons défini entièrement une fonction de tâche $e_{CO}(q)$ permettant de contourner les obstacles tout en évitant au mieux les problèmes d'occultations. Il reste alors à définir la loi de commande \dot{q}_{CO} régulant $e_{CO}(q)$ à zéro. Dans la mesure où la matrice $W_{\text{coll}} = \mathbf{J}_{\text{coll}}$ est choisie de manière à satisfaire les conditions (I.13), le jacobien $\mathbf{J}_{CO} = \frac{\partial e_{CO}}{\partial q}$ est défini positif, et la fonction de tâche $e_{CO}(q)$ est ρ -admissible, et \dot{q}_O est alors donné par :

$$\dot{q}_{CO} = -\lambda_{CO} e_{CO}(q) \quad \forall \lambda_{CO} > 0 \quad (\text{II.20})$$

Nous avons ainsi déterminé une loi de commande \dot{q}_{CO} permettant d'éviter les collisions, tout en minimisant le risque de perte du motif visuel.

II.3.2 Tâche d'évitement d'occultation $e_{OS}(q)$

Nous cherchons à présent à déterminer une tâche $e_{OS}(q)$ garantissant l'évitement des occultations de type ③, c'est-à-dire celles qui se produisent loin des obstacles. Comme précédemment, nous utilisons le formalisme des tâches redondantes. Nous choisissons ici de définir la tâche redondante prioritaire $e_{\text{occ}}(q)$ pour assurer la non-occultation du motif visuel et la fonction de coût secondaire $h_s(q)$ de manière à réaliser au mieux la tâche de navigation référencée vision. Nous proposons alors la fonction de tâche globale $e_{OS}(q)$ suivante :

$$e_{OS}(q) = \mathbf{J}_{\text{occ}}^+ e_{\text{occ}}(q) + \beta_{\text{occ}} (\mathbb{I}_3 - \mathbf{J}_{\text{occ}}^+ \mathbf{J}_{\text{occ}}) g_s \quad \text{avec} \quad \begin{cases} \beta_{\text{occ}} > 0 \\ W_{\text{occ}} = \mathbf{J}_{\text{occ}} = \frac{\partial e_{\text{occ}}}{\partial q} \\ g_s = \frac{\partial h_s}{\partial q} \end{cases} \quad (\text{II.21})$$

Tout d'abord, caractérisons la tâche prioritaire $e_{\text{occ}}(q)$ d'évitement d'occultation. Nous proposons la fonction de tâche redondante suivante [FOLIO et CADENAT, ECC-CDC2005, RO-BIO2006] :

$$e_{\text{occ}}(q) = \begin{pmatrix} e_{d_{\text{occ}}} \\ e_{d_{\text{bord}}} \end{pmatrix} = \begin{pmatrix} \tan \left(\frac{\pi}{2} - \frac{\pi d_{\text{occ}}}{2 D_+} \right) \\ d_{\text{bord}} \end{pmatrix} \quad (\text{II.22})$$

Remarque II.3: Nous aurions pu définir la fonction de tâche $e_{d_{\text{occ}}}$ de la même manière qu'au paragraphe II.2.1 (cf. l'équation (II.4)). Toutefois, nous souhaitons montrer ici que différentes formulations peuvent être utilisées pour $e_{d_{\text{occ}}}$ afin de garantir la non occultation du motif visuel.

La première composante de $e_{\text{occ}}(q)$ permet d'éviter les occultations. En effet, le terme $e_{d_{\text{occ}}}$ tend vers l'infini lorsque l'objet occultant \mathcal{O}_{cc} se rapproche du motif visuel \mathbf{s} , et inversement décroît lorsque \mathcal{O}_{cc} s'éloigne de \mathbf{s} . Notons, que $\forall d_{\text{occ}} \geq D_+$ (i.e. pas de risque d'occultation), nous imposons à la composante $e_{d_{\text{occ}}}$ d'être maintenue à zéro. De la même manière qu'à la section II.2, la seconde composante fait sortir l'objet occultant de l'image, ce qui se produit lorsque la distance d_{bord} s'annule. Nous pouvons désormais déterminer la jacobienne \mathbf{J}_{occ} de la tâche prioritaire $e_{\text{occ}}(q)$ qui se déduit à partir des relations (II.9), (II.10) et (II.22) comme suit :

$$\mathbf{J}_{\text{occ}} = \frac{\partial e_{\text{occ}}}{\partial q} = \begin{pmatrix} -\frac{\varepsilon_{\text{occ}} \pi}{D_+} \frac{\pi}{2} \left(1 + \tan^2 \left(\frac{\pi}{2} - \frac{\pi d_{\text{occ}}}{2 D_+} \right) \right) (\mathcal{L}_{Y_s} - \mathcal{L}_{Y_{\text{occ}}}) \\ \varepsilon_{\text{bord}} \mathcal{L}_{Y_{\text{occ}}} \end{pmatrix} \mathbf{J}_r \quad (\text{II.23})$$

Définissons, maintenant, la tâche secondaire de manière à continuer la réalisation de l'asservissement visuel par le biais du critère quadratique $h_s(q)$ suivant [CADENAT, 1999] :

$$h_s(q) = \frac{1}{2}(\mathbf{s} - \mathbf{s}^*)^T(\mathbf{s} - \mathbf{s}^*)$$

L'introduction de ce critère garantit la réalisation de la tâche de navigation basée sur la vision pendant l'évitement d'occultations, du moins tant que la parfaite réalisation du mouvement le permet. Nous pouvons alors en déduire le gradient $g_s(q)$ associé :

$$g_s(q) = ((\mathbf{s} - \mathbf{s}^*)^T \mathcal{L}_r \mathbf{J}_r)^T$$

Nous venons de définir la fonction de tâche $e_{os}(q)$ d'évitement d'occultations tout en assurant la poursuite de la tâche de navigation référencée vision nominale. Il reste à définir la loi de commande \dot{q}_{os} régulant $e_{os}(q)$ à zéro. Dans la mesure où la matrice $W_{occ} = \mathbf{J}_{occ}$ est choisie de manière à satisfaire les conditions (I.13), le jacobien $\mathbf{J}_{os} = \frac{\partial e_{os}}{\partial q}$ est défini positif, et la fonction de tâche $e_{os}(q)$ est ρ -admissible. La commande \dot{q}_{os} est alors donnée par :

$$\dot{q}_{os} = -\lambda_{os} e_{os}(q) \quad \forall \lambda_{os} > 0 \quad (\text{II.24})$$

Nous avons ainsi défini une loi de commande \dot{q}_{os} permettant d'éviter les occultations loin des obstacles, tout en continuant à converger aux mieux vers le motif de référence \mathbf{s}^* .

II.3.3 Stratégie de commande

Notre objectif est de définir une loi de commande permettant à un robot mobile de naviguer sur la base d'un asservissement visuel en évitant les occultations et les collisions éventuelles. La stratégie envisagée ici consiste à basculer entre les trois correcteurs précédemment synthétisés (\dot{q}_{AV} , \dot{q}_{CO} et \dot{q}_{OS}) en fonction des risques de collision (μ_{coll}) et d'occultations (μ_{occ}), comme représenté dans le tableau II.2 (*cf.* remarque II.25).

	$\mu_{occ} = 0$	$\mu_{occ} = 1$
$\mu_{coll} = 0$	$\dot{q} = \dot{q}_{AV}$	$\dot{q} = \dot{q}_{OS}$
$\mu_{coll} = 1$	$\dot{q} = \dot{q}_{CO}$	$\dot{q} = \dot{q}_{CO}$

TAB. II.2 – Stratégie de basculement.

Remarque II.4: *Il est ici nécessaire de faire un choix sur les priorités à donner à la tâche d'évitement d'occultation face à celle d'évitement de collision. Comme la sécurité du robot est pour nous primordiale, nous avons conçu notre stratégie de commande de manière à privilégier l'évitement de collision par rapport à l'évitement d'occultation. De plus, nous avons défini la fonction de tâche e_{CO} de manière à minimiser le risque d'occultation. Ainsi, si μ_{coll} passe à 1, la commande \dot{q}_{CO} sera systématiquement appliquée au robot, garantissant la non collision et minimisant les problèmes d'occultation.*

Nous proposons alors la loi de commande globale suivante :

$$\dot{q} = (1 - \mu_{coll})(1 - \mu_{occ})\dot{q}_{AV} + (1 - \mu_{coll})\mu_{occ}\dot{q}_{OS} + \mu_{coll}\dot{q}_{CO} \quad \forall \mu_{coll}, \mu_{occ} \in [0; 1] \quad (\text{II.25})$$

où les commandes \dot{q}_{AV} , \dot{q}_{CO} et \dot{q}_{OS} sont données respectivement par les relations (I.21), (II.20) et (II.24).

Les fonctions μ_{coll} et μ_{occ} assurent le basculement continu d'une loi de commande à l'autre en fonction des risques d'occultation et de collision. Différents cas se présentent :

- ① Si $d_{\text{occ}} > D_+$ (pas de risque d'occultation) et si $d_{\text{coll}} > d_+$ (pas de risque de collision), μ_{occ} et μ_{coll} sont fixés à zéro et seul l'asservissement visuel (I.21) pilote le robot.
- ② Si la base mobile entre dans la zone définie par ξ_0 le danger de collision devient plus important et μ_{coll} augmente. Si le robot entre dans la région définie par ξ_- le danger de collision est maximal, μ_{coll} est fixé à 1 et un indicateur $\acute{\text{E}}\text{VITE}_{\text{coll}}$ est activé. Le robot est alors ramené sur l'enveloppe de contournement ξ_0 et la suit jusqu'à ce que la commande de contournement devienne compatible avec la commande référencée vision, c'est-à-dire lorsque les deux commandes indiquent la même direction : $\text{sign}(\dot{q}_{AV}) = \text{sign}(\dot{q}_{CO})$. Quand cette condition est remplie un indicateur $\text{SORTIR}_{\text{coll}}$ est activé, et la distance d_S est fixée à la distance d_{coll} correspondante. Le danger de collision μ_{coll} décroît jusqu'à ce que le robot franchisse ξ_+ . Ainsi, μ_{coll} s'écrit en de fonction de la distance d_{coll} comme suit :

$$\left\{ \begin{array}{lll} \mu_{\text{coll}} = 0 & \text{si } d_{\text{coll}} > d_0 & \text{et } \acute{\text{E}}\text{VITE}_{\text{coll}} = 0 \\ \mu_{\text{coll}} = \frac{d_{\text{coll}} - d_0}{d_- - d_0} & \text{si } d_{\text{coll}} \in [d_-, d_0] & \text{et } \acute{\text{E}}\text{VITE}_{\text{coll}} = 0 \\ \mu_{\text{coll}} = \frac{d_{\text{coll}} - d_+}{d_S - d_+} & \text{si } d_{\text{coll}} \in [d_S, d_+] & \text{et } \text{SORTIR}_{\text{coll}} = 1 \\ \mu_{\text{coll}} = 1 & \text{sinon} & \end{array} \right.$$

- ③ Si le motif visuel \mathbf{s} pénètre dans la région délimitée par Ξ_0 le danger d'occultation est plus important et μ_{occ} augmente progressivement. Le robot est donc commandé par une combinaison linéaire de \dot{q}_{AV} et \dot{q}_{OS} entre Ξ_0 et Ξ_- ; et si le véhicule entre dans le voisinage d'un obstacle pendant cette phase la commande \dot{q}_{CO} est également appliquée. Si une composante de \mathbf{s} franchit l'enveloppe Ξ_- le danger d'occultation est maximal et μ_{occ} reste fixé à 1 tant que la condition de sortie $\text{SORTIR}_{\text{occ}}$ n'est pas vérifiée. L'indicateur $\text{SORTIR}_{\text{occ}}$ est activé lorsque \mathcal{O}_{cc} a quitté l'image (*i.e.* $d_{\text{bord}} = 0$) ou lorsque \mathbf{s} est ramené sur l'enveloppe Ξ_0 (*i.e.* $d_{\text{occ}} \geq D_0$). À cet instant μ_{occ} décroît et s'annule dès que $d_{\text{occ}} \geq D_+$. Afin de définir μ_{occ} nous introduisons un indicateur $\acute{\text{E}}\text{VITE}_{\text{occ}}$ fixé à 1 dès que $d_{\text{occ}} \leq D_-$ et conservant cette valeur tant que la condition de sortie n'est pas vérifiée. La distance D_S est également définie comme étant la distance d_{occ} correspondant à cet événement. μ_{occ} est donné par :

$$\left\{ \begin{array}{lll} \mu_{\text{occ}} = 0 & \text{si } d_{\text{occ}} > D_0 & \text{et } \acute{\text{E}}\text{VITE}_{\text{occ}} = 0 \\ \mu_{\text{occ}} = \frac{d_{\text{occ}} - D_0}{D_- - D_0} & \text{si } d_{\text{occ}} \in [D_-, D_0] & \text{et } \acute{\text{E}}\text{VITE}_{\text{occ}} = 0 \\ \mu_{\text{occ}} = \frac{d_{\text{occ}} - D_+}{D_S - D_+} & \text{si } d_{\text{occ}} \in [D_S, D_+] & \text{et } \text{SORTIR}_{\text{occ}} \\ \mu_{\text{occ}} = 1 & \text{sinon} & \end{array} \right.$$

Nous avons ainsi défini un schéma de commande référencée multi-capteurs permettant à notre robot mobile de réaliser un asservissement visuel tout en évitant indépendamment les collisions et les occultations. Par ailleurs, nous avons proposé une approche permettant de gérer efficacement les différents situations de risque d'occultation pouvant se produire. Enfin, notre souhait de préserver l'intégrité du robot est assuré de part la définition de notre stratégie de commande (II.25), et la synthèse de la commande d'évitement d'obstacles \dot{q}_{CO} (II.20)

II.3.4 Résultats

La stratégie de commande (II.25) a été validée en simulation à l'aide de Matlab [FOLIO et CADENAT, ECMS2005,ECC-CDC]avant d'être appliquée à notre robot Super Scout [FOLIO et CADENAT, ROBIO2006]. Nous présentons ci-après deux résultats de simulation, puis nous montrerons un résultat expérimental.

II.3.4.1 Simulations

Nous avons choisi d'appliquer la nouvelle loi de commande (II.25) aux deux exemples introduits lors de l'illustration de la stratégie d'évitement simultané de collision et d'occultation. Toutefois, ici, dans les deux résultats de simulation, les distances des enveloppes D_- , D_0 et D_+ ont été fixées respectivement à 40, 70 et 80 pixels, et d_+ , d_0 , d_- à 0.7 m, 0.6 m et 0.4 m. Notons que la période d'échantillonnage a été choisie égale à 50 ms, valeur correspondant à notre plate-forme expérimentale.

Évitement de deux obstacles cylindriques

Le premier résultat est constitué d'un environnement encombré de deux obstacles cylindriques présentant un risque éventuel d'occultation et de collision [FOLIO, EDSYS2005]. L'objectif de la mission est, encore une fois, de positionner la caméra embarquée face à un amer visuel constitué de quatre points.

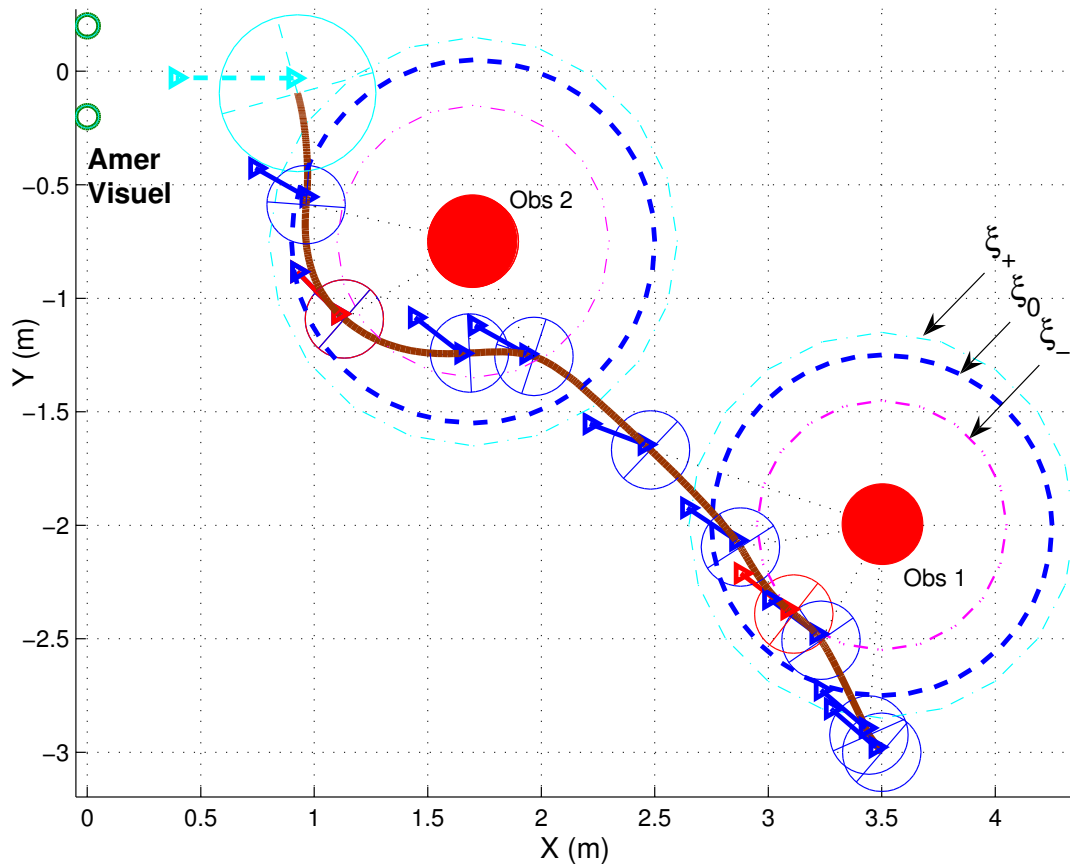


FIG. II.12 – Trajectoire du robot [FOLIO, EDSYS2005].

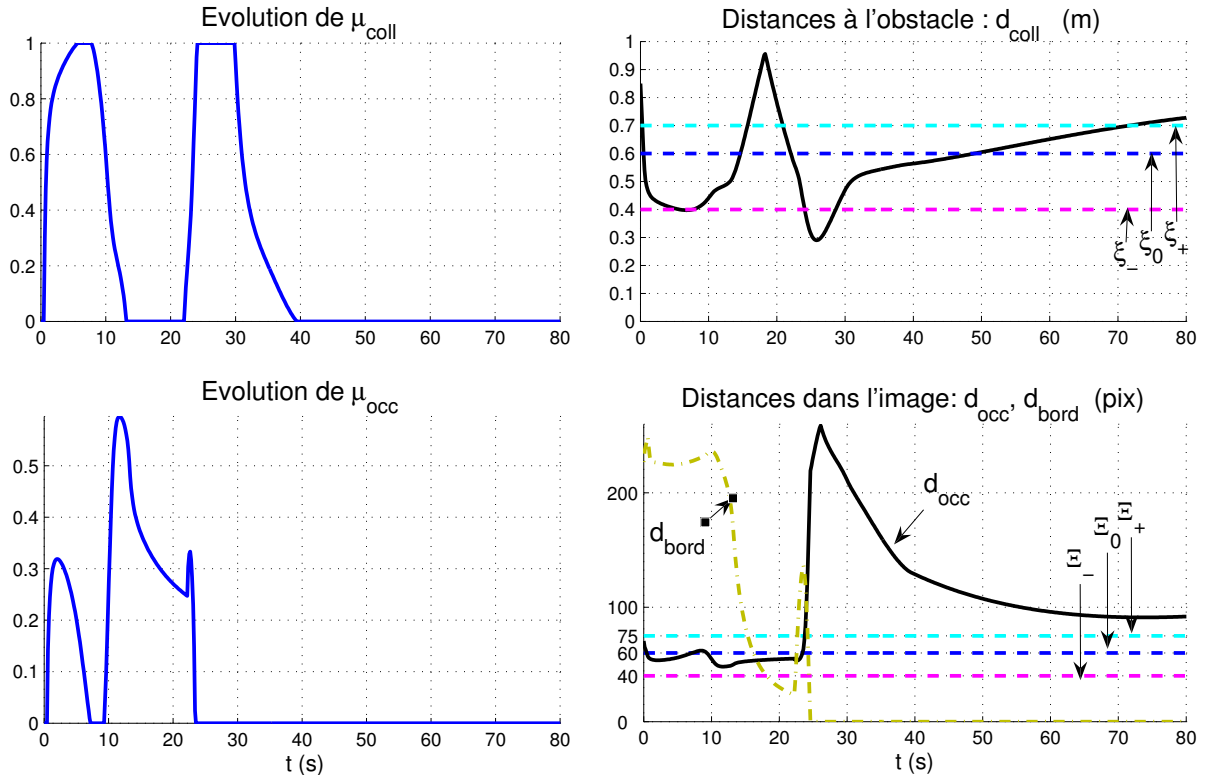


FIG. II.13 – Données d'évitement [FOLIO, EDSYS2005].

Comme représenté sur les figures II.12 et II.13, au début de la tâche il n'y a pas de risque d'occultation ($\mu_{occ} = 0$) ni de collision ($\mu_{coll} = 0$). Le robot est donc piloté uniquement par la commande référencée vision \dot{q}_{AV} . Puis le véhicule entre dans le voisinage du premier obstacle, le risque de collision μ_{coll} croît, et la commande globale \dot{q} bascule progressivement vers \dot{q}_{CO} qui permet d'éviter la collision. Pendant cette phase le danger d'occultation μ_{occ} augmente légèrement. De ce fait, une combinaison linéaire de \dot{q}_{CO} et \dot{q}_{OS} est alors appliquée au robot jusqu'à ce que μ_{occ} revienne à 0. Le risque de collision devenant maximal, $\mu_{coll} = 1$, le véhicule est piloté uniquement à partir de \dot{q}_{CO} . Une fois le premier obstacle dépassé μ_{coll} est ramené à 0, et la loi de commande privilégie progressivement \dot{q}_{AV} . Toutefois, la réalisation de la tâche référencée vision guidant le robot vers l'amer visuel, fait apparaître la projection du second obstacle dans le champ de vision de la caméra. Le risque d'occultation μ_{occ} augmente sensiblement, et la commande globale \dot{q} bascule progressivement vers \dot{q}_{OS} . Celle-ci permet de diriger le véhicule de sorte que l'occultation soit évitée. Le risque μ_{occ} commence alors à décroître. Cependant, avant que μ_{occ} ne s'annule, le robot entre dans le voisinage du second obstacle générant un risque de collision, μ_{coll} croît à nouveau pour atteindre 1. La seule commande \dot{q}_{CO} est appliquée au véhicule qui contourne l'obstacle tout en terminant au mieux l'évitement d'occultation. Une fois le dernier obstacle dépassé, μ_{occ} et μ_{coll} sont ramenés à 0, et seul l'asservissement visuel pilote le robot jusqu'à ce qu'il ait fini de converger vers la situation désirée \mathbf{s}^* . La mission est alors parfaitement réalisée.

Évitement de deux murs

La seconde simulation, présentée sur les figures II.14 et II.15, consiste toujours à positionner la caméra embarquée face à un amer visuel, mais dans un environnement encombré de deux murs [FOLIO et CADENAT, JNRR2005].

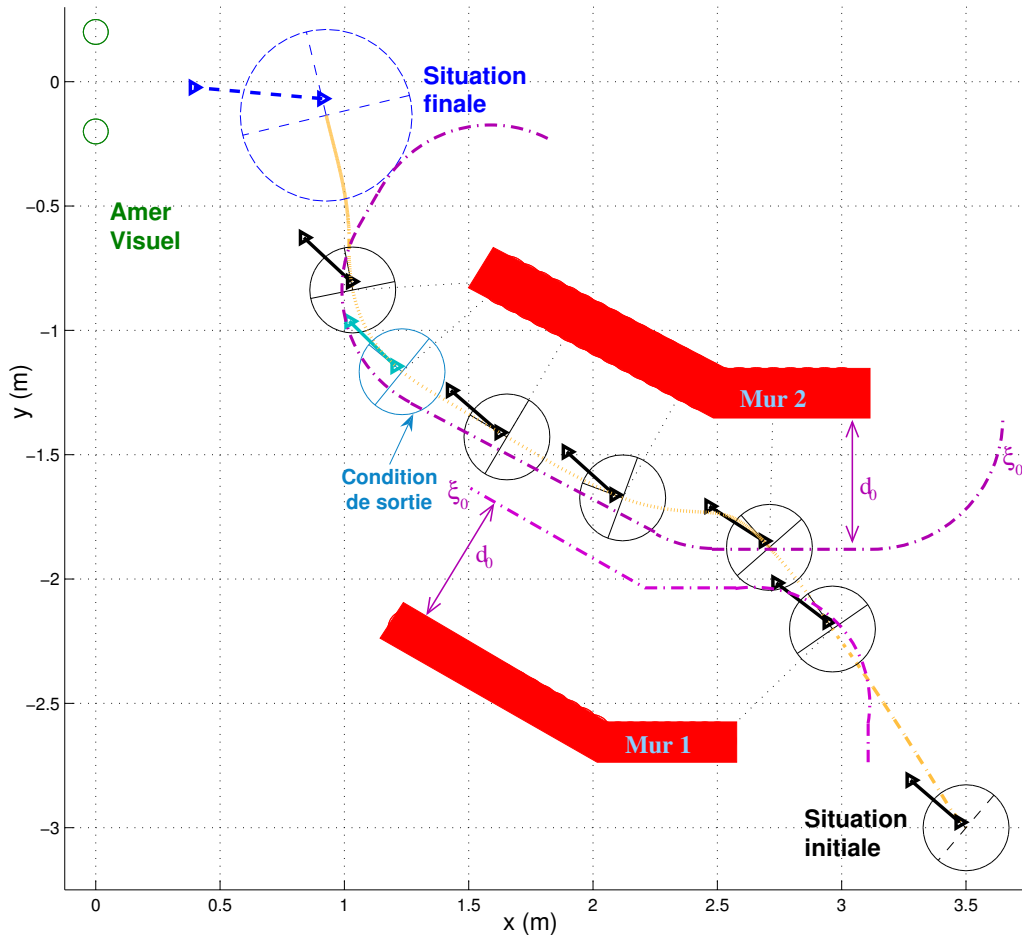


FIG. II.14 – Trajectoire du robot [FOLIO et CADENAT, JNRR2005].

Comme précédemment, au début de la tâche, seul l'asservissement visuel \dot{q}_{AV} dirige le véhicule. Toutefois, très vite, le mouvement induit par celui-ci contraint le motif visuel à se rapprocher de l'objet occultant correspondant à la projection du mur 2 sur le plan image, et le risque d'occultation augmente progressivement. N'incluant pas de composante d'évitement de collision, la combinaison linéaire des commandes \dot{q}_{AV} et \dot{q}_{OS} amène le véhicule à entrer dans le voisinage du mur 1. Le risque de collision augmente à son tour, et une combinaison des trois commandes \dot{q}_{AV} , \dot{q}_{OS} et \dot{q}_{CO} est alors appliquée. Toutefois, la commande globale \dot{q} bascule progressivement principalement vers \dot{q}_{CO} qui est prioritaire par rapport à \dot{q}_{AV} et \dot{q}_{OS} (cf. remarque II.4). Ainsi, l'application de \dot{q}_{CO} permet d'éviter parfaitement la collision des deux murs, tout en minimisant les risques d'occultations. Une fois le dernier mur dépassé, le danger de collision μ_{coll} décroît, et il n'y a plus alors de risque d'occultation. La commande globale \dot{q} peut à nouveau commuter vers la commande d'asservissement visuel, et le robot parvient à faire converger parfaitement le motif visuel vers la situation désirée \mathbf{s}^* .

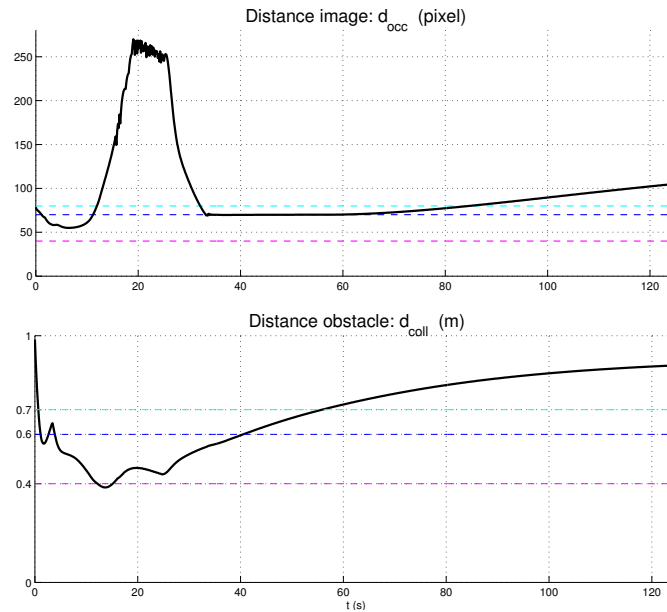


FIG. II.15 – Données d'évitements.

II.3.4.2 Expérimentation

Nous avons expérimenté la loi de commande (II.25) sur le robot Super Scout de notre étude [FOLIO et CADENAT, ROBIO2006]. L'objectif de la mission consiste ici à positionner la caméra face à un amer visuel dans un environnement encombré de deux obstacles. Le premier obstacle ne présente qu'un risque de collision pour la base mobile, tandis que le second génère principalement un danger d'occultation.

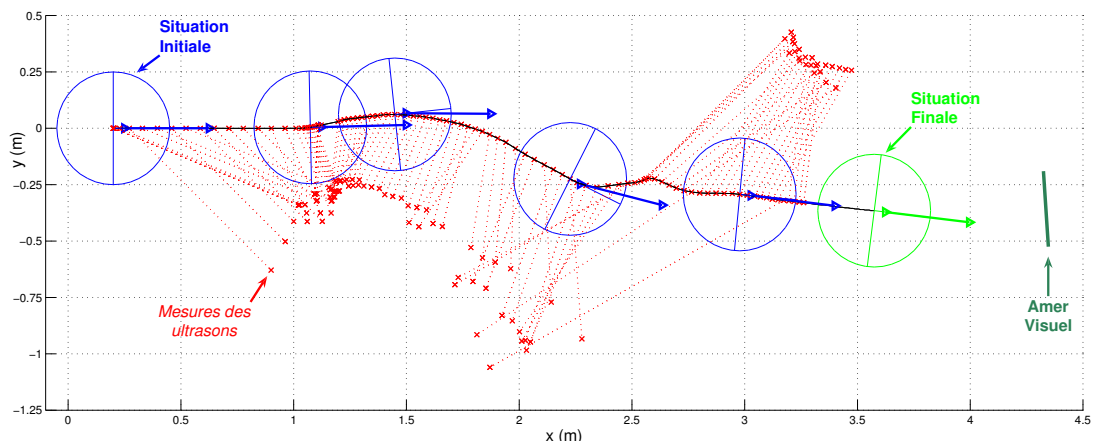


FIG. II.16 – Asservissement visuel et évitement découplé d'occultations et de collisions.

Comme le présentent les figures II.16, II.17 et II.18 la tâche de navigation est parfaitement réalisée : les collisions et les occultations sont parfaitement évitées durant toute la mission. Au début de la tâche il n'y a pas de risque de collision ($\mu_{coll} = 0$, cf. figure II.17.a) ni d'occultation ($\mu_{occ} = 0$, cf. figure II.18.a). Le robot est alors dirigé uniquement par la commande référencée vision \dot{q}_{AV} , et il commence à converger vers l'amer visuel. Puis la base mobile entre dans le voisinage du premier obstacle, et le risque de collision μ_{coll} croît pour atteindre sa valeur maximale. La commande globale \dot{q} bascule

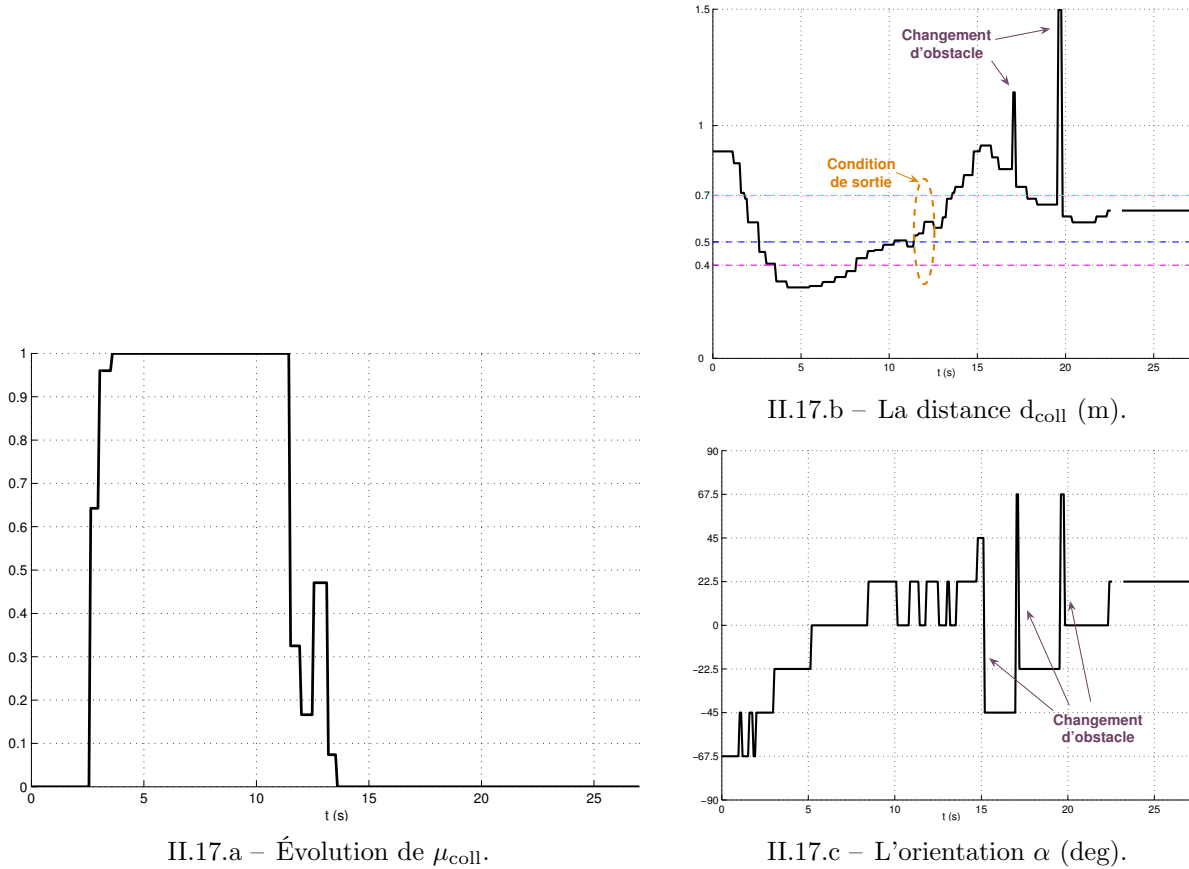


FIG. II.17 – Les mesures des capteurs ultrasons.

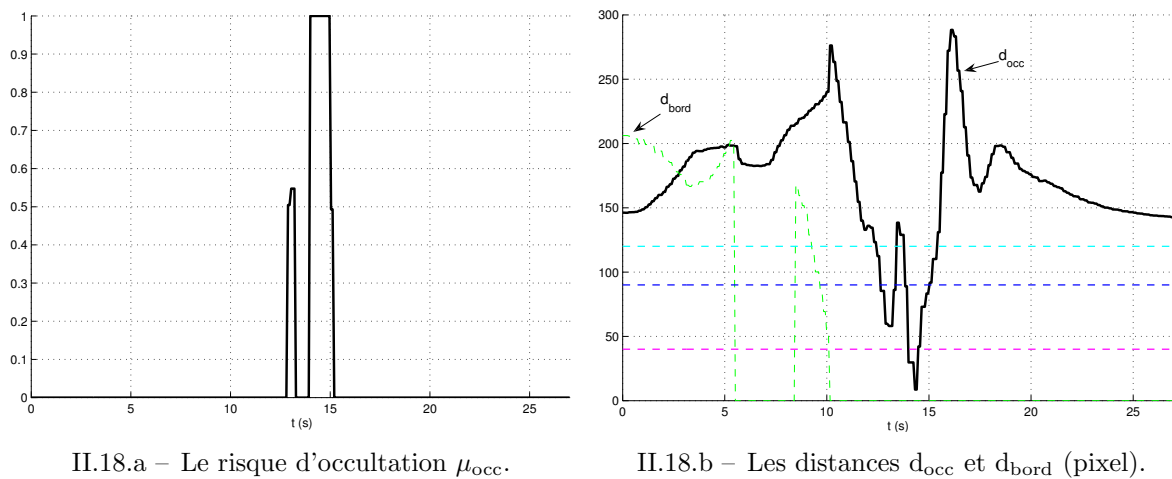


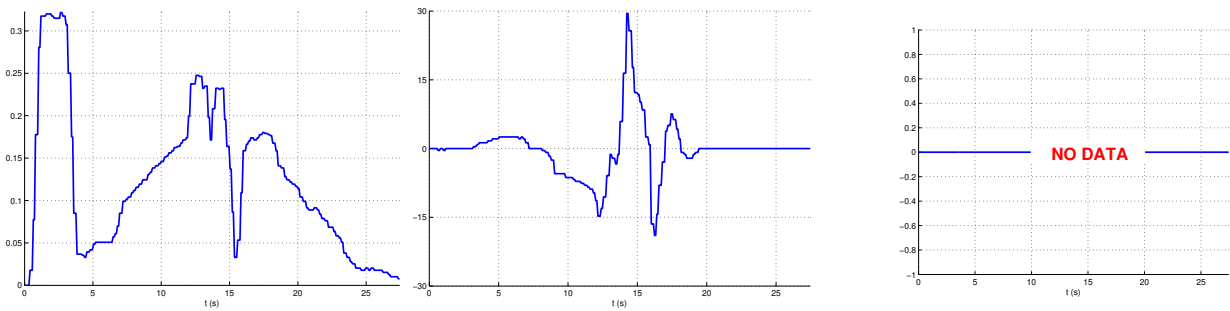
FIG. II.18 – Les mesures du risque d'occultations.

alors progressivement vers \dot{q}_{CO} , et le robot commence à éviter l'obstacle (représenté par les premières mesures des capteurs ultrasoniques sur la figure II.16, voir également la figure II.17.b) tout en minimisant le risque d'occultation au mieux. Toutefois, alors que le véhicule commence à quitter le voisinage du premier obstacle, un risque d'occultation se produit et μ_{occ} croît (cf. figure II.18.a). Ainsi, à ce moment, le robot doit alors éviter à la fois la collision et l'occultation. Lorsque la base mobile parvient à quitter le voisinage du premier obstacle, le risque d'occultation μ_{occ} continue à augmenter pour atteindre la

valeur maximale 1. Alors, seule la commande d'évitement d'occultation \dot{q}_{OS} est appliquée pour garantir un mouvement du robot sans occultation. Enfin, lorsqu'il n'y a plus aucun risque d'occultation, ni de collision, l'asservissement visuel est à nouveau appliqué au robot qui converge alors vers l'amer visuel de référence. La tâche de navigation est alors parfaitement réalisée.

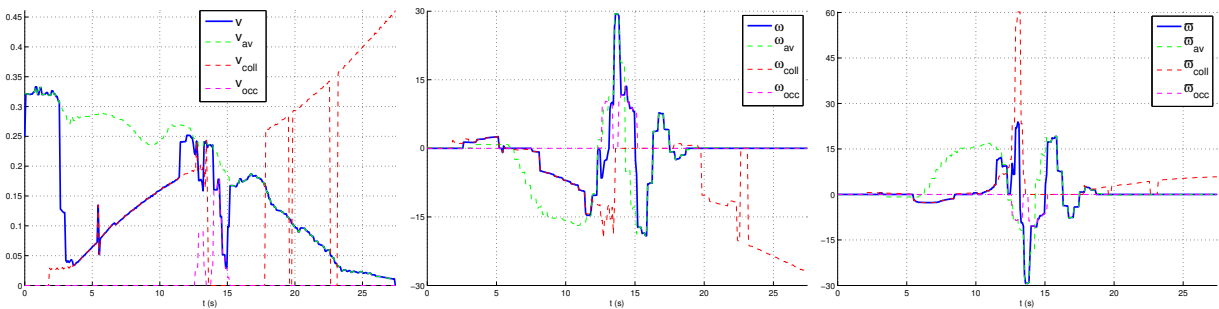
Remarque II.5: Dans la mesure où le second obstacle est très petit (environ 6 cm de diamètre) les 16 capteurs ultrasons ne parviennent pas toujours à le détecter. Ce phénomène, qui se produit deux fois au cours de la tâche de navigation, est symbolisé par deux sauts sur l'évolution de la distance d_{coll} (cf. figure II.17.b), et par les deux dernières variations sur l'orientation α (cf. figure II.17.c). Dans la mesure où ces sauts interviennent alors qu'il n'y a plus de risque de collision ($\mu_{\text{coll}} = 0$, cf. figure II.17.a), ils n'induisent pas de perturbation sur la commande appliquée au robot.

Les courbes II.19 et II.20 représentent respectivement les vitesses effectivement appliquées⁶ au robot et les vitesses calculées à partir de la commande globale (II.25). Comme nous pouvons le constater le Super Scout suit correctement la loi de commande globale déterminée précédemment sans retard, ni problème de saturation. Ce résultat illustre la bonne adéquation de notre stratégie de commande avec la boucle de commande bas niveau contrôlant les actionneurs du robot.



II.19.a – Vitesse linéaire v m/s . II.19.b – Vitesse angulaire ω deg/s. II.19.c – Vitesse de la platine ϖ .

FIG. II.19 – Les vitesses effectivement appliquées par le robot.



II.20.a – Vitesse linéaire v m/s. II.20.b – Vitesse angulaire ω deg/s. II.20.c – Vitesse de la platine ϖ .

FIG. II.20 – Les vitesses synthétisées.

6. Nous n'avons cependant pas de mesure de vitesses de la platine ϖ .

II.3.5 Bilan

Nous avons proposé une nouvelle stratégie de commande référencée multi-capteurs permettant d'éviter les occultations tout en garantissant la non-collision. Elle est basée sur la définition de trois correcteurs réalisant respectivement la tâche référencée vision (\dot{q}_{AV}), l'évitement de collision (\dot{q}_{CO}) et d'occultation (\dot{q}_{OS}). Il nous est alors possible de traiter les différents cas d'occultations présentés sur la figure II.11, et de réaliser la tâche de navigation désirée. Dans la mesure où il existe une solution permettant d'atteindre cet objectif tout en préservant la visibilité des indices perceptuels et l'intégrité du robot.

II.4 Discussion

Dans ce chapitre, nous nous sommes intéressés à la définition de tâches de navigation référencées vision pour un robot mobile devant évoluer dans un environnement encombré d'obstacles susceptibles de perturber le signal visuel $s(q, t)$. D'une manière générale, la grande majorité des travaux relatifs à la gestion des phénomènes d'occultations proposent, soit de planifier le mouvement de la caméra dans l'image, soit de gérer de manière robuste la perte des informations sensorielles. Nous avons choisi de proposer des stratégies de commande assurant à la fois l'évitement des occultations et des obstacles. Ainsi, dans un premier temps, nous avons proposé une loi de commande permettant d'éviter les occultations tout en minimisant le risque de collision. De ce fait, le robot peut être amené à passer dans un voisinage très proche des obstacles et sa sécurité ne peut être pleinement garantie. Nous avons alors proposé un second schéma de commande où l'évitement des occultations et des collisions est découplé afin de préserver l'intégrité du système robotique.

Cependant, bien que les différents schémas de commande soient bien conditionnés, leur combinaison peut générer des problèmes de minima locaux (*i.e.* la commande globale \dot{q} est nulle ailleurs qu'en la situation désirée). Ces problèmes sont liés à un comportement antagoniste des différents correcteurs en phase de transition (*e.g.* $\mu \in]0; 1[$). Pour limiter ces effets, il faut veiller à définir convenablement les paramètres de basculement de ces derniers (distances des enveloppes de sécurité, gains...). Mais le choix est souvent des plus délicats et nécessite une bonne expertise du système. De plus, il n'est pas toujours possible de trouver le bon compromis assurant un déroulement de la mission adéquat. En effet, l'ajout de l'évitement des phénomènes d'occultation sur-contraint la tâche robotique et réduit le nombre de trajectoires permettant d'atteindre la situation désirée. Considérons, par exemple, un environnement encombré de deux obstacles : un premier muret de hauteur inférieure à celle de la caméra, et un second obstacle cylindrique pouvant générer un risque d'occultation, comme illustré sur les figures II.21 et II.23. Nous nous proposons d'appliquer nos deux stratégies de commande (II.14) et (II.25) à cet environnement, en remarquant que, dans ce cas, le robot ne peut pas contourner le muret par le haut.

II.4.1 Évitement simultané d'occultations et de collisions

Nous appliquons, ici, la loi de commande (II.14) permettant d'éviter simultanément les occultations et les collisions. Comme représenté sur les figures II.22.a et II.22.c, au début de la tâche il n'y a pas de risque d'occultation ($d_{occ} > D_+$) ni de collision ($d_{coll} > d_+$). Le robot est donc commandé uniquement par l'asservissement visuel \dot{q}_{AV} . Puis le véhicule

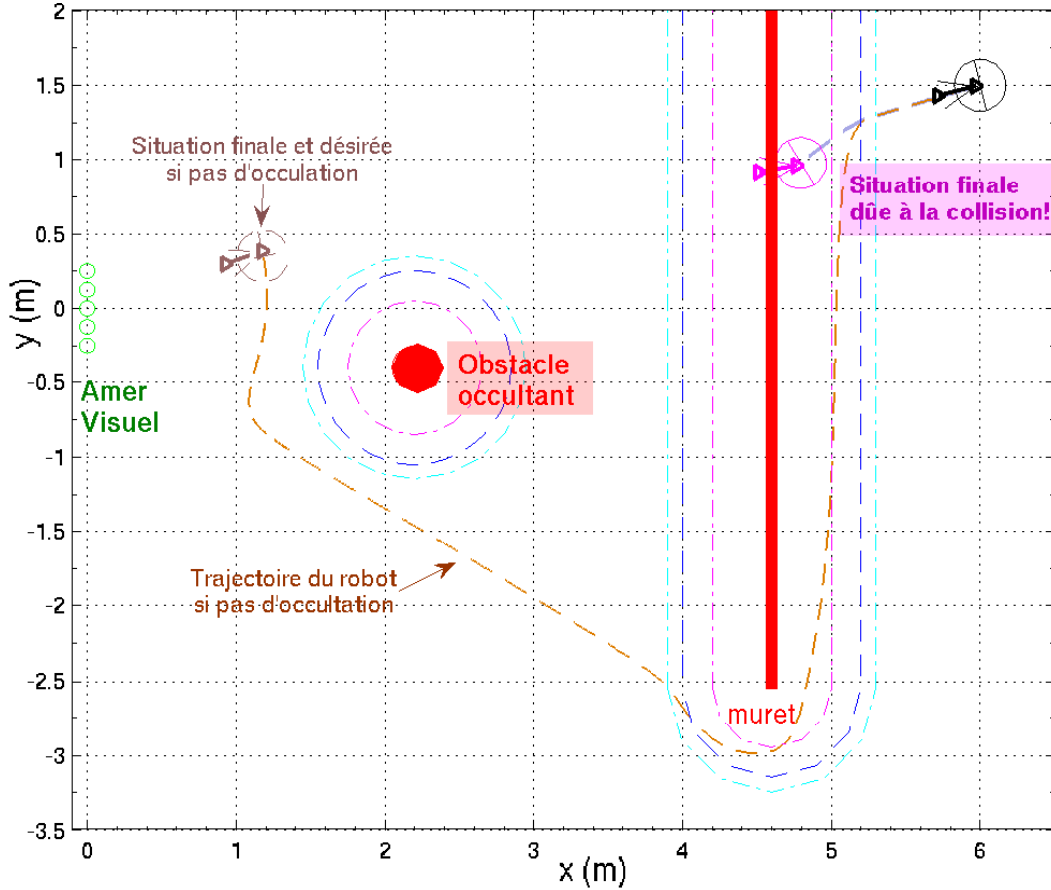


FIG. II.21 – Trajectoire du robot.

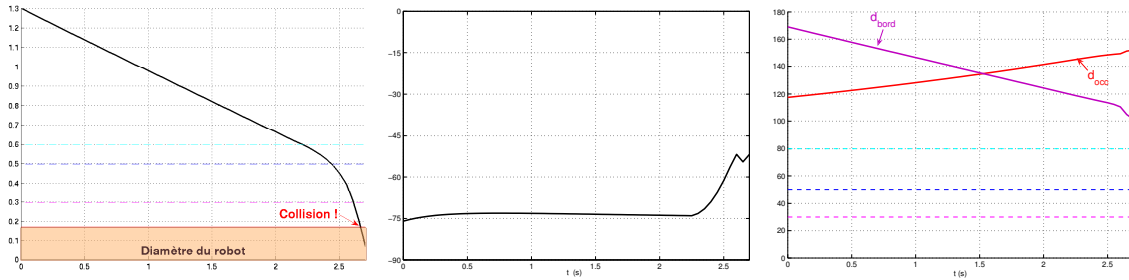
II.22.a – La distance d_{coll} (m). II.22.b – L'orientation α (deg). II.22.c – Les distances d_{occ} et d_{bord} (pixels).

FIG. II.22 – Évitement simultané d'occultations et de collisions.

entre dans le voisinage du muret, la distance d_{coll} diminue et le risque de collision μ_{coll} croît. La commande globale \dot{q} bascule progressivement vers \dot{q}_o qui permet d'assurer l'évitement d'occultation tout en contournant *au mieux* les obstacles (cf. figure II.4). Bien qu'il n'y ait pas de risque important d'occultation, dans la mesure où l'objet occultant est présent dans l'image (cf. figure II.22.c), la tâche e_o cherche à maximiser la distance d_{occ} . Le mouvement produit n'est alors pas suffisant pour contourner le muret, et le robot vient le heurter. La mission n'est donc pas réalisée convenablement.

II.4.2 Évitement découplé d'occultations et de collisions

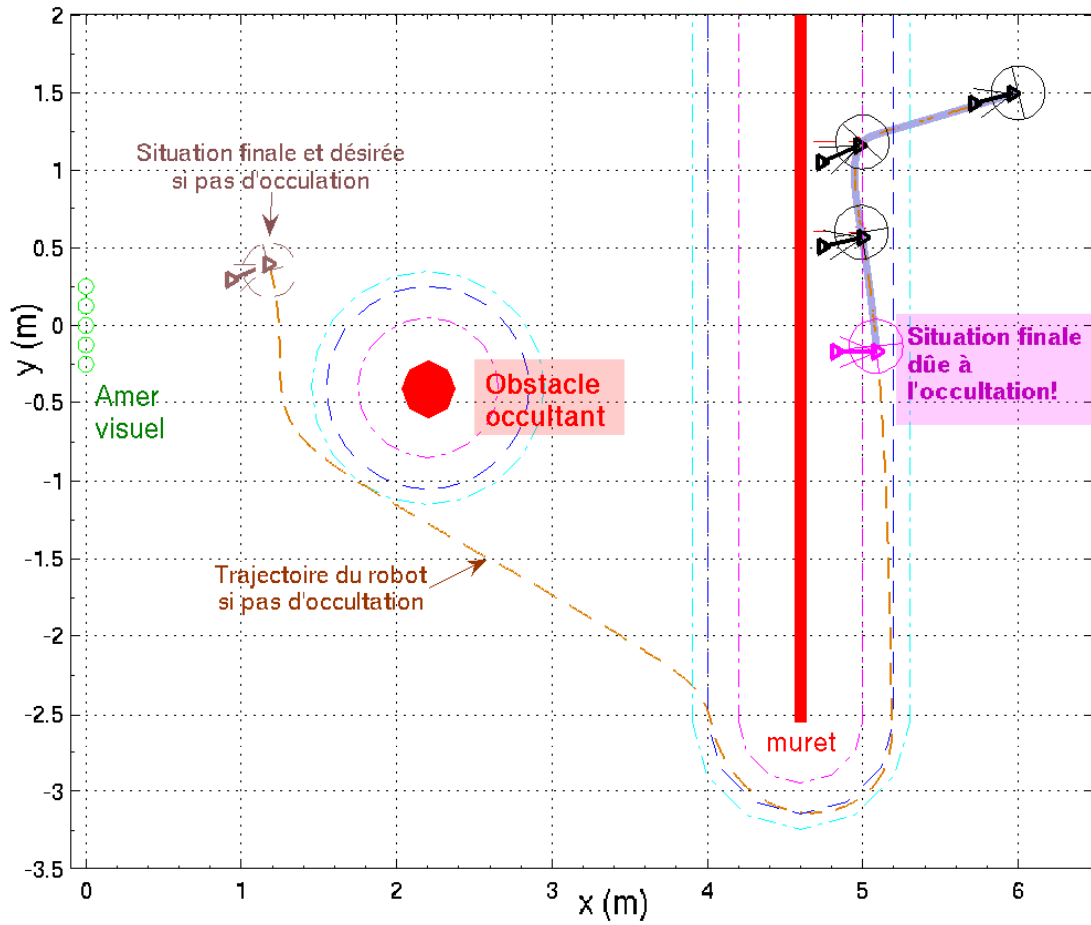
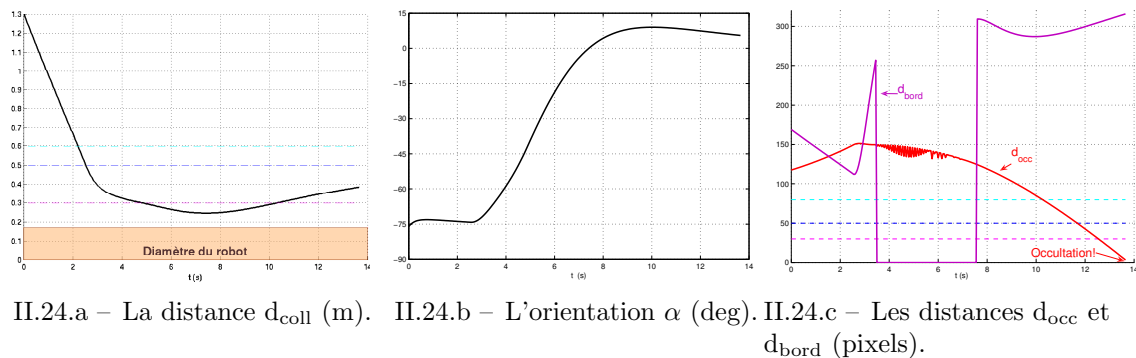


FIG. II.23 – Trajectoire du robot.



II.24.a – La distance d_{coll} (m). II.24.b – L'orientation α (deg). II.24.c – Les distances d_{occ} et d_{bord} (pixels).

FIG. II.24 – Évitement découplé d'occultations et de collisions.

Nous utilisons, à présent, la stratégie de commande (II.25) permettant d'éviter conjointement les collisions et les occultations. Comme précédemment, au début de la tâche, il n'y a pas de risque d'occultation ($d_{occ} > D_+$) ni de collision ($d_{coll} > d_+$) (voir figure II.24), le robot est alors commandé uniquement par l'asservissement visuel \dot{q}_{AV} . Puis, celui-ci entre dans le voisinage du muret, le risque de collision μ_{coll} croît, et la commande globale

\dot{q} bascule progressivement vers \dot{q}_{CO} . Ce correcteur permet d'assurer l'intégrité du robot puisqu'il garantit la non collision, tout en *minimisant* le risque d'occultation. Le robot commence donc à contourner le muret et la tâche est correctement réalisée jusqu'à ce que l'objet occultant entre dans le champ de vision de la caméra, et masque le motif visuel. Il est alors impossible de continuer d'éviter à la fois l'obstacle et l'occultation. Encore une fois, le véhicule s'arrête avant d'atteindre le but.

II.4.3 Bilan

Ainsi, l'environnement considéré permet de mettre en exergue les limites des stratégies proposées. Celles-ci sont restreintes à des missions pour lesquelles les occultations et les collisions peuvent être simultanément évitées. Notons d'ailleurs que, si le second obstacle cylindrique n'est pas occultant, le robot parvient parfaitement à effectuer la tâche de navigation à l'aide des deux schémas de commande précédents. Afin d'aller plus loin et de pouvoir élargir la gamme de missions réalisables, il est nécessaire d'étendre nos travaux pour tolérer la perte de tout ou partie du motif visuel d'intérêt

Chapitre III

Contribution à la reconstruction du motif visuel

COMME nous l'avons vu au chapitre II la réalisation d'une tâche de navigation référencée vision requiert de préserver l'intégrité du robot (*i.e.* mouvement sans collision) tout en garantissant la visibilité des informations visuelles. Pour répondre à cette problématique, nous avons présenté différentes lois de commande permettant d'éviter à la fois les collisions et les occultations. Mais ces méthodes ne sont envisageables que lorsque l'évitement conjoint de ces deux phénomènes est possible sans générer des minima locaux ou des situations de blocage de la mission.

Cependant, certaines tâches robotiques ne peuvent être réalisées qu'au prix d'une perte temporaire du signal visuel. Aussi chercher à éviter systématiquement les occultations n'est pas toujours la stratégie la plus pertinente. C'est pourquoi, nous cherchons à étendre les travaux précédents en développant des méthodes capables de prédire les indices visuels lorsque ceux-ci sont indisponibles. De cette manière, il est possible de réaliser une tâche de navigation référencée vision en tolérant les pertes partielles ou **totales** du motif d'intérêt.

Ce chapitre est organisé de la manière suivante. Dans un premier temps, nous introduisons la problématique considérée ainsi que les hypothèses sous-jacentes de notre étude. Puis nous rappellerons le lien existant entre la vision et le mouvement nous permettant de définir un système dynamique. Enfin, nous proposons un ensemble de méthode permettant de reconstruire le motif lorsqu'il n'est plus accessible depuis le capteur de vision.

III.1 Introduction

Dans ce chapitre nous nous focalisons essentiellement sur les situations où les données visuelles $\mathbf{s}(q, t)$ ne sont plus disponibles pendant l'exécution d'une tâche de navigation référencée vision en environnement encombré. En particulier, au cours d'une telle mission, différents problèmes peuvent survenir :

- occultation ou perte des données perceptuelles,
- déficience temporaire du processus de traitement d'image ou de la caméra,
- utilisation de la caméra pour une autre tâche plus prioritaire,
- etc.

Ainsi, l'hypothèse de base qui sous-tend notre étude est qu'il n'est plus possible d'obtenir **momentanément** les indices visuels $\mathbf{s}(q, t)$ à travers la caméra, et plus généralement nous considérons que l'**image entière** devient inaccessible. Notre objectif est donc ici de proposer des méthodes capables de traiter des pertes temporaires partielles ou **totales** du motif visuel d'intérêt, suffisamment rapidement pour pouvoir s'intégrer efficacement au sein de notre boucle de commande. Ainsi les techniques restreintes à la gestion des pertes *partielles* des données visuelles, telles que celles proposées dans [MALIS *et al.*, 1999], [GARCIA-ARACIL *et al.*, 2005], [COMPORT *et al.*, 2004], etc. ne répondent pas à nos exigences. Nous nous sommes donc principalement intéressés aux approches permettant de reconstruire le motif visuel $\mathbf{s}(q, t)$ dès qu'il devient inaccessible. Une première solution est de considérer les techniques issues du **tracking**, très largement abordé en vision et robotique. En effet, de nombreux travaux traitent le suivi d'un motif visuel de manière robuste aux occultations totales [FAVARO et SOATTO, 2003], [DRUMMOND et CIPOLLA, 2002], [JACKSON *et al.*, 2004], [VACCHETTI et LEPETIT, 2004], [LEPETIT et FUA, 2006], etc. Cependant comme elles reposent classiquement sur des mesures prélevées dans l'image (qui est considérée indisponible dans notre contexte), elles ne peuvent être retenues ici. Il est également possible d'aborder le problème à l'aide des techniques issues du traitement du signal telles que le filtre de Kalman [KALMAN, 1960] [MAYBECK, 1979], ou le filtre particulière [CRISAN et DOUCET, 2002]. Toutefois, ce type de techniques ne présente un réel intérêt lorsque la valeur initiale de l'état du système dynamique à reconstruire est mal connue voir inconnue. Or, dans notre cas, nous supposons que nous disposons toujours *d'au moins* une mesure du signal sensoriel, avant que ce dernier ne soit perdu. De plus, nous ne considérons pas ici d'éventuels bruits perturbant la caractérisation des différentes données (valeurs initiales, commande, paramètres de calibration...). De ce fait, synthétiser un filtre ne semble donc pas le plus pertinent dans ce contexte. Enfin, une dernière approche envisageable consiste à utiliser un modèle 2D ou 3D de l'objet d'intérêt et à exploiter la géométrie projective pour retrouver les données manquantes. Toutefois, ce choix conduit à localiser le robot par rapport à l'amer considéré, et nous rapproche alors des techniques d'asservissement visuel 3D, avec les inconvénients qui y sont liés (*cf.* chapitre I).

Nous avons donc finalement opté pour une solution alternative permettant d'estimer les données sensorielles en exploitant l'historique des mouvements de la caméra et des mesures des indices visuels $\mathbf{s}(q, t)$ avant leur perte (*cf.* table III.1). La stratégie envisagée consiste à développer une méthode reposant sur le lien entre la variation des informations visuelles $\dot{\mathbf{s}}$ et le mouvement de la caméra \mathcal{T}_{C/R_0} défini par la relation (I.9) que nous avons présentée au chapitre I (*cf.* figure III.1).

t_0	t_1	\dots	t_{k-1}	t_k	\rightarrow	t_{k+1}
\mathbf{s}_0	\mathbf{s}_1	\dots	\mathbf{s}_{k-1}	\mathbf{s}_k	perte	$\widehat{\mathbf{s}}_{k+1}$?
q_0	q_1	\dots	q_{k-1}	q_k	de $\mathbf{s}(q, t)$	
$\mathcal{T}_{C/\mathcal{R}_0}(t_0)$	$\mathcal{T}_{C/\mathcal{R}_0}(t_1)$	\dots	$\mathcal{T}_{C/\mathcal{R}_0}(t_{k-1})$	$\mathcal{T}_{C/\mathcal{R}_0}(t_k)$		
Historique					\rightarrow	Estimation

TAB. III.1 – Perte des informations visuelles

Hypothèses générales

Dans la suite de ce chapitre, et jusqu'à mention contraire, nous ne faisons aucune hypothèse concernant le système robotique portant la caméra. Nous nous plaçons donc dans le cas général d'un système à six degrés de liberté et $\mathcal{T}_{C/\mathcal{R}_0}$ est alors défini par :

$$\mathcal{T}_{C/\mathcal{R}_0} = \begin{pmatrix} V_{C/\mathcal{R}_0} \\ \Omega_{\mathcal{R}_C/\mathcal{R}_0} \end{pmatrix} = \left(V_{\bar{x}_C} \quad V_{\bar{y}_C} \quad V_{\bar{z}_C} \quad \Omega_{\bar{x}_C} \quad \Omega_{\bar{y}_C} \quad \Omega_{\bar{z}_C} \right)^T \quad (\text{III.1})$$

Nous supposons néanmoins que le mouvement de la caméra est *holonome*. Deux situations peuvent alors se présenter, soit le robot porteur est holonome et par conséquent le mouvement de la caméra l'est aussi ; soit il est non-holonome, et nous admettons que la caméra est en mesure de se déplacer indépendamment de celui-ci.

Par ailleurs, nous rappelons également que nous ne prenons en compte que le cas où l'amer visuel d'intérêt \mathcal{O} est **fixe** dans l'environnement et que seul le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ est susceptible de faire évoluer les mesures sensorielles $\mathbf{s}(q, t)$ dans l'image (*cf.* figure III.1). Enfin, nous supposons que le motif visuel $\mathbf{s}(q, t)$ est présent dans l'image *au moins* au début de la tâche robotique (*cf.* table III.1). De ce fait, initialement, nous considérons qu'il n'y a pas de problème d'occultation ni d'accessibilité de $\mathbf{s}(q, t)$.

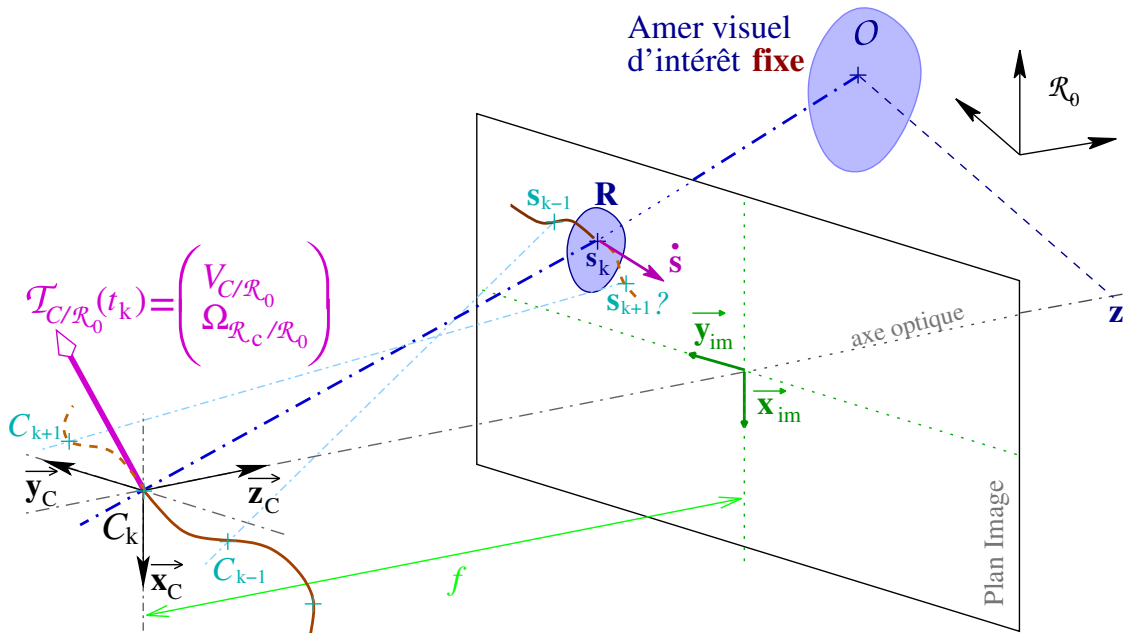


FIG. III.1 – Le lien vision/mouvement.

III.2 Mouvement de la caméra et variation des informations visuelles

Considérons un objet \mathcal{O} rigide et fixe de la scène. Ce solide se projette sur le plan image de la caméra en une région \mathbf{R} (cf. figure III.1) à laquelle est associée un ensemble de primitives visuelles $\mathbf{s}(q, t)$, et un vecteur $\mathbf{z}(q, t)$ décrivant leur profondeur. Comme nous l'avons préalablement introduit, la variation des informations visuelles $\dot{\mathbf{s}}$ est liée à la cinématique de la caméra $\mathbf{T}_{C/\mathcal{R}_0} = (V_{C/\mathcal{R}_0}^T, \Omega_{\mathcal{R}_C/\mathcal{R}_0}^T)^T$ par le biais de la matrice d'interaction $\mathcal{L}(\mathbf{s}, \mathbf{z})$ (cf. la relation (I.9), page 31) :

$$\dot{\mathbf{s}}(q, t) = \mathcal{L}(\mathbf{s}, \mathbf{z})\mathbf{T}_{C/\mathcal{R}_0}$$

Notre intention est d'exploiter cette relation pour estimer le motif visuel dès qu'il devient indisponible. Nous nous proposons de modéliser notre problème pour différents types de primitives avant de nous intéresser au cas général.

III.2.1 Le cas simple du point

Le point est une primitive couramment employée en asservissement visuel, car il s'agit d'une primitive très simple que l'on sait aisément extraire de l'image à l'aide d'algorithmes efficaces, tels que le détecteur de Harris [HARRIS et STEPHENS, 1988]¹. Il est aussi possible de considérer des ensembles de points relativement denses, mais également des formes géométriques plus significatives comme les centres de gravité, les segments, les polygones, etc.

Nous considérons donc ici un amer visuel constitué par un ensemble de n points. Le vecteur des informations sensorielles $\mathbf{s}(q, t)$ est alors défini par : $\mathbf{s} = (X_1, Y_1, \dots, X_j, Y_j, \dots, X_n, Y_n)^T$. Les couples (X_j, Y_j) représentent les coordonnées métriques du $j^{\text{ème}}$ point $p_j(x_j, y_j, z_j)$ de l'amer de la scène \mathcal{O} , projeté en P_j sur le plan image de la caméra (cf. figure I.6.b, page 29). Comme nous l'avons vu au chapitre I, la variation $\dot{\mathbf{s}}_j$ d'un point P_j s'écrit classiquement [CHAUMETTE, 1990] :

$$\dot{\mathbf{s}}_j = \begin{pmatrix} \dot{X}_j \\ \dot{Y}_j \end{pmatrix} = \mathcal{L}(P_j, z_j)\mathbf{T}_{C/\mathcal{R}_0} \quad (\text{III.2})$$

où $\mathcal{L}(P_j, z_j)$ est donnée par la relation (I.10) que nous rappelons ci-après :

$$\mathcal{L}(P_j, z_j) = \begin{pmatrix} \mathcal{L}(X_j, z_j) \\ \mathcal{L}(Y_j, z_j) \end{pmatrix} = \begin{pmatrix} -\frac{f}{z} & 0 & \frac{X}{z} & \frac{XY}{f} & -\left(f + \frac{X^2}{f}\right) & Y \\ 0 & -\frac{f}{z} & \frac{Y}{z} & \left(f + \frac{Y^2}{f}\right) & -\frac{XY}{f} & -X \end{pmatrix}$$

Nous cherchons à reconstruire la primitive visuelle $\mathbf{s}_j = (X_j, Y_j)$ associée au point P_j lorsque la caméra n'est plus en mesure de la fournir. Pour cela, nous proposons d'exploiter le lien existant entre $\dot{\mathbf{s}}_j$ et $\mathbf{T}_{C/\mathcal{R}_0}$ en résolvant le système dynamique (III.2). Cependant,

1. Nous ne considérons pas ici le problème, souvent difficile, de mise en correspondance et de suivi des points d'une image à l'autre.

l'expression de la matrice d'interaction $\mathcal{L}(P_j, z_j)$ dépendant *explicitement* de la profondeur z_j du point P_j considéré. Il est nécessaire de la déterminer pour résoudre ce système (III.2). La profondeur z_j peut être obtenue de différentes manières. Tout d'abord, elle peut être mesurée à l'aide de capteurs spécifiques, tels qu'un laser ou un système de stéréo-vision. Mais, ne disposant d'aucun de ces capteurs sur notre robot, nous devons nous orienter vers des approches permettant de la reconstruire. Par exemple, il est possible d'exploiter les principes du SFM (*Structure from Motion*) qui permet d'établir le modèle 3D de l'objet observé à partir de la dynamique des informations perceptuelles [JERIAN et JAIN, 1991], [AZARBAYEJANI et PENTLAND, 1995], [CHAUMETTE *et al.*, 1994, 1996], [SOATTO et PERONA, 1998A,B], [OLIENSIS, 2002], *etc.* Il est également envisageable d'utiliser un filtre de Kalman comme dans [MATTHIES *et al.*, 1989], ou bien des méthodes de localisation relatives à l'amer considéré [OBERKAMPF *et al.*, 1996], [THRUN *et al.*, 2001], *etc.* Cependant, là encore, ces différentes techniques supposent que l'image est disponible, et ne sont donc pas applicables dans notre cas. C'est pourquoi, nous proposons une solution alternative consistant à estimer *conjointement* la profondeur z_j et les indices visuels associés $\mathbf{s}_j = (X_j, Y_j)$. Pour cela, il est nécessaire d'établir le lien entre sa variation \dot{z}_j et le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0} = (V_{C/\mathcal{R}_0}^T, \Omega_{\mathcal{R}_C/\mathcal{R}_0}^T)^T$. Rappelant que le point p_j est fixe dans l'environnement, sa vitesse V_{p_j/\mathcal{R}_C} par rapport au repère de la caméra \mathcal{R}_C peut s'écrire de la manière suivante :

$$V_{p_j/\mathcal{R}_C} = -V_{C/\mathcal{R}_0} - \Omega_{\mathcal{R}_C/\mathcal{R}_0} \wedge p_j = \dot{p}_j \quad (\text{III.3})$$

La dynamique \dot{z}_j de la profondeur est alors donnée par la relation ci-après :

$$\begin{aligned} \dot{z}_j &= -V_{\mathbf{z}_C} - \frac{z_j Y_j}{f} \Omega_{\mathbf{x}_C} + \frac{z_j X_j}{f} \Omega_{\mathbf{y}_C} \\ &= \begin{pmatrix} 0 & 0 & -1 & -\frac{z_j Y_j}{f} & \frac{z_j X_j}{f} & 0 \end{pmatrix} \mathcal{T}_{C/\mathcal{R}_0} \\ \dot{z}_j &= \mathcal{L}_{z_j}(P_j, z_j) \mathcal{T}_{C/\mathcal{R}_0} \end{aligned} \quad (\text{III.4})$$

Donc, pour tout point P_j de l'amer visuel, nous pouvons caractériser entièrement la dynamique du triplé $\boldsymbol{\psi}_j = (X_j, Y_j, z_j)^T$ par :

$$\dot{\boldsymbol{\psi}}_j = \begin{pmatrix} \dot{X}_j \\ \dot{Y}_j \\ \dot{z}_j \end{pmatrix} = \begin{pmatrix} \mathcal{L}(X_j, z_j) \\ \mathcal{L}(Y_j, z_j) \\ \mathcal{L}_{z_j}(P_j, z_j) \end{pmatrix} \mathcal{T}_{C/\mathcal{R}_0} = \mathcal{L}(\boldsymbol{\psi}_j) \mathcal{T}_{C/\mathcal{R}_0} \quad (\text{III.5})$$

Ainsi, pour reconstruire une primitive visuelle de type point, il suffira d'intégrer le système d'équations différentielles (III.5). Ce résultat peut naturellement être étendu à un motif visuel constitué d'un ensemble de n points. Dans cette situation nous avons $\mathbf{s} = (X_1, Y_1, \dots, X_j, Y_j, \dots, X_n, Y_n)^T$, et il s'agit alors de reconstruire le vecteur $\boldsymbol{\psi} = (\boldsymbol{\psi}_1^T, \dots, \boldsymbol{\psi}_n^T)^T$, où $\boldsymbol{\psi}_j = (X_j, Y_j, z_j)^T$, en considérant le système d'équations différentielles suivant :

$$\dot{\boldsymbol{\psi}} = \begin{pmatrix} \dot{\boldsymbol{\psi}}_1 \\ \vdots \\ \dot{\boldsymbol{\psi}}_n \end{pmatrix} = \begin{pmatrix} \mathcal{L}(\boldsymbol{\psi}_1) \\ \vdots \\ \mathcal{L}(\boldsymbol{\psi}_n) \end{pmatrix} \mathcal{T}_{C/\mathcal{R}_0} = \mathcal{L}(\boldsymbol{\psi}) \mathcal{T}_{C/\mathcal{R}_0} \quad (\text{III.6})$$

III.2.2 Les autres primitives géométriques usuelles

Bien qu'il soit possible de construire différentes primitives géométriques sur la base du point, il est parfois intéressant de considérer d'autres paramétrisations du vecteur des indices visuels \mathbf{s} pour définir la tâche référencée vision. Nous nous intéressons ici aux primitives géométriques usuelles telles que les droites, les ellipses, les sphères, les cylindres, etc. Comme précédemment, nous cherchons à déterminer le système dynamique à résoudre pour reconstruire ces primitives lorsqu'elles ne peuvent plus être fournies par la caméra.

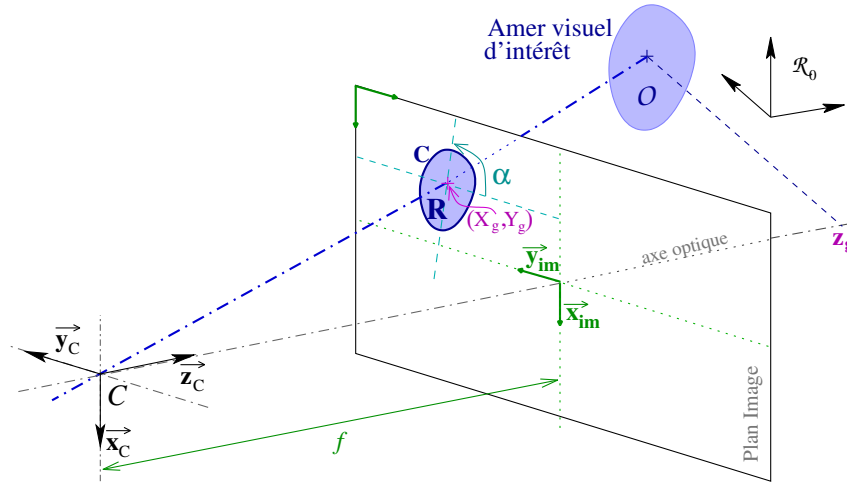


FIG. III.2 – Projection d'un objet quelconque dans l'image.

Considérons un objet \mathcal{O} fixe de la scène qui se projette dans l'image en une région \mathbf{R} ayant un contour fermé \mathbf{C} (cf. figure III.2). Nous admettons que cet objet est formé d'une surface continue, c'est-à-dire ne présentant pas de discontinuité sur la profondeur. Dans ce cas, la profondeur $z(q, t)$ de chaque point de l'objet \mathcal{O} peut s'exprimer par une fonction continue des points images $P_j (X_j, Y_j) \in \mathbf{R}$ de la manière suivante [CHAUMETTE, 1990, 2004], [ESPIAU *et al.*, 1992] :

$$\frac{1}{z_j} = \sum_{p \geq 0, q \geq 0} A_{pq} X_j^p Y_j^q, \quad \forall X_j, Y_j \in \mathbf{R} \quad (\text{III.7})$$

où les coefficients A_{pq} dépendent de la nature de l'objet \mathcal{O} considéré. Par exemple, dans le cas d'un objet plan dont l'équation s'exprime dans le repère caméra \mathcal{R}_C comme suit :

$$z_j = ax_j + by_j + c, \quad \forall p_j : (x_j, y_j, z_j) \in \mathcal{O}$$

l'équation III.7 se réduit à [CHAUMETTE, 2004] :

$$\frac{1}{z_j} = AX_j + BY_j + C, \quad \text{avec : } \begin{cases} C = A_{00} = \frac{1}{c}, & A = A_{10} = -\frac{a}{c}, & B = A_{01} = -\frac{b}{c} \\ \text{et } A_{pq} = 0 & \forall (p, q) = (1, 1) \text{ et } p, q > 1 \end{cases} \quad (\text{III.8})$$

Nous supposons qu'il est possible d'associer à l'objet d'intérêt \mathcal{O} des primitives géométriques usuelles (droites, sphères, cercles, etc.), et que la profondeur $z(q, t)$ s'exprime de manière implicite sous la forme (III.7). Notons $\mathbf{s} = (\pi_1, \dots, \pi_n)^T$ le vecteur des indices visuels considérés, où les composantes π_j correspondent aux différents paramètres

nécessaires pour caractériser la ou les primitives géométriques issues de l'objet \mathcal{O} observé (cf. remarque III.1). Dans ce cas, la variation $\dot{\mathbf{s}}$ peut s'écrire comme suit :

$$\dot{\mathbf{s}} = \begin{pmatrix} \dot{\pi}_1 \\ \vdots \\ \dot{\pi}_n \end{pmatrix} = \begin{pmatrix} \mathcal{L}(\pi_1, A_{pq}) \\ \vdots \\ \mathcal{L}(\pi_n, A_{pq}) \end{pmatrix} \mathcal{T}_{C/\mathcal{R}_0} = \mathcal{L}(\pi_1, \dots, \pi_n, A_{pq}) \mathcal{T}_{C/\mathcal{R}_0} \quad (\text{III.9})$$

où $\mathcal{L}(\pi_j, A_{pq})$ est la matrice d'interaction² associée à la $j^{\text{ème}}$ composante de \mathbf{s} . Cette matrice dépend de π_j et de la profondeur à travers les paramètres A_{pq} . Le vecteur des indices visuels considérés \mathbf{s} ne pourra donc être reconstruit par l'intégration de ce système que si ces paramètres sont connus³. Plusieurs approches sont envisageables. Il est tout d'abord possible d'utiliser le modèle 3D de l'objet s'il est disponible. Toutefois, comme nous l'avons déjà mentionné, nous souhaitons effectuer un nombre minimal d'hypothèses sur l'objet d'intérêt, et nous ne retenons donc pas cette solution. Une autre solution consisterait à caractériser les coefficients A_{pq} à chaque étape de l'estimation. En effet, ces coefficients peuvent, par exemple, s'obtenir au moyen d'une méthode de reconstruction 3D par vision dynamique [BOUKIR, 1993]. Cependant, l'obtention d'une bonne estimation des coefficients A_{pq} nécessite, dans ce cas, un mouvement particulier de la caméra. Enfin, une approche alternative serait de pouvoir exprimer la variation \dot{A}_{pq} , et de résoudre le système dynamique (III.9) en ajoutant ces nouvelles équations. Toutefois, l'expression de la variation \dot{A}_{pq} dépendant généralement du type de primitive géométrique considérée il serait difficile de proposer une résolution générique. Nous proposons donc plutôt de déterminer les coefficients A_{pq} en exploitant l'équation (III.7). Pour cela, il suffit de considérer un nombre l suffisant de points P_j appartenant à la région \mathbf{R} . Ensuite, connaissant leurs coordonnées $(X_j, Y_j, z_j)^T$, il suffit d'appliquer une méthode d'identification pour obtenir les paramètres recherchés. Cela nous conduit alors à travailler en quatre étapes selon l'algorithme III.1 (cf. remarque III.1).

Algorithme III.1 Reconstruction des primitives géométriques.

- 1: Considérer un ensemble de l points appartenant à la région \mathbf{R} dans l'image, et définir ainsi un vecteur $\boldsymbol{\psi} = (X_1, Y_1, z_1, \dots, X_l, Y_l, z_l)^T$ comme dans le paragraphe III.2.1.
 - 2: Intégrer le système d'équations différentielles (III.6) et en déduire une estimation $\hat{\boldsymbol{\psi}}$.
 - 3: Utiliser cette estimée $\hat{\boldsymbol{\psi}}$ pour déterminer les coefficients A_{pq} de la surface passant au plus près des l points par une méthode telle que les moindres carrés.
 - 4: Connaissant les coefficients A_{pq} , intégrer le système dynamique (III.9) et en déduire une estimation du vecteur des indices visuels $\mathbf{s} = (\pi_1, \dots, \pi_n)^T$.
-

Remarque III.1: *Si toutes les primitives n'appartiennent pas au même objet \mathcal{O} observé, il s'agit alors d'associer un ensemble de l points à chaque objet considéré, et de suivre l'algorithme III.1 proposé ci-dessus pour chacun de ces ensembles.*

Finalement, il est possible de reconstruire le motif visuel $\mathbf{s} = (\pi_1, \dots, \pi_n)^T$ associé à différentes primitives géométriques dès lors qu'une formulation analytique de leurs matrices d'interactions existe et dont toutes les composantes sont connues (ou tout au moins identifiables). Nous verrons au chapitre IV un exemple applicatif de l'algorithme III.1 pour le cas de l'ellipse.

2. Le lecteur intéressé trouvera dans [CHAUMETTE, 1990] une méthode générale de calcul de la matrice d'interaction pour diverses primitives géométriques et pour de multiples paramétrisations.

3. Nous retrouvons donc le problème de l'estimation de la profondeur déjà rencontré dans le cas du point, à ceci près que celle-ci n'apparaît plus maintenant de manière explicite.

III.2.3 Les moments de l'image

Les moments de l'image constituent un autre ensemble de primitives plus récemment considérées en asservissement visuel et qui offrent un certain nombre de propriétés intéressantes. Tout d'abord, ils fournissent une représentation générique et significative des objets de forme simple ou complexe en proposant une approche statistique. Ensuite, ils présentent l'avantage de traiter la forme de l'objet dans son ensemble, de sorte qu'il n'est plus nécessaire de chercher à mettre en correspondance les points d'une image à l'autre, mais uniquement des régions. Enfin, les moments permettent d'obtenir de très bonnes propriétés de découplage du mouvement de la caméra (*e.g.* séparation des mouvements de translation et de rotation) [TAHRI, 2004, TAHRI et CHAUMETTE, 2005].

Nous nous proposons de rappeler brièvement les définitions relatives aux moments 2D ainsi qu'aux moments centrés de l'image. Comme précédemment, nous considérons un objet \mathcal{O} fixe de la scène qui se projette en une région \mathbf{R} dans l'image ayant un contour fermé \mathbf{C} (*cf.* figure III.2).

Remarque III.2: *Afin de simplifier les expressions, nous supposons, dans cette section, que l'objet \mathcal{O} observé est plan. La profondeur $z(q, t)$ des points appartenant à \mathcal{O} est alors reliée aux coordonnées $(X_j, Y_j)^T$ des points de l'image par la relation (III.8). Le lecteur intéressé trouvera une méthode générale de calcul analytique de la matrice d'interaction associée aux moments de l'image pour un objet quelconque dans [CHAUMETTE, 2004].*

III.2.3.1 Les moments classiques

Le moment 2D de l'image m_{ij} d'ordre $i + j$ de l'objet \mathcal{O} est classiquement défini pour tous les points $P(X, Y)$ de la région \mathbf{R} de l'image par la relation suivante :

$$m_{ij} = \iint_{\mathbf{R}} X^i Y^j dXdY \quad (\text{III.10})$$

De manière similaire, pour un ensemble de n points $P_k(X_k, Y_k)$ appartenant à la région \mathbf{R} , on peut définir les moments m_{ij} sous la forme discrète ci-après :

$$m_{ij} = \sum_{k=1}^n X_k^i Y_k^j \quad (\text{III.11})$$

La variation \dot{m}_{ij} est alors donnée par [CHAUMETTE, 2004] :

$$\dot{m}_{ij} = \mathcal{L}_{m_{ij}} \mathbf{T}_{C/\mathcal{R}_0}$$

où $\mathcal{L}_{m_{ij}}$ s'écrit [TAHRI et CHAUMETTE, 2005] :

$$\mathcal{L}_{m_{ij}} = \begin{pmatrix} m_{V_{\vec{x}C}} \\ m_{V_{\vec{y}C}} \\ m_{V_{\vec{z}C}} \\ m_{\Omega_{\vec{x}C}} \\ m_{\Omega_{\vec{y}C}} \\ m_{\Omega_{\vec{z}C}} \end{pmatrix}^T = \begin{cases} m_{V_{\vec{x}C}} & = -i (Am_{ij} + Bm_{i-1,j+1} + Cm_{i-1,j}) - A\beta m_{ij} \\ m_{V_{\vec{y}C}} & = -j (Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) - B\beta m_{ij} \\ m_{V_{\vec{z}C}} & = (i + j + 3\beta) (Am_{i+1,j} + Bm_{i,j+1} + Cm_{i,j}) - C\beta m_{ij} \\ m_{\Omega_{\vec{x}C}} & = (i + j + 3\beta) m_{i,j+1} + j m_{i,j-1} \\ m_{\Omega_{\vec{y}C}} & = -(i + j + 3\beta) m_{i+1,j} - i m_{i-1,j} \\ m_{\Omega_{\vec{z}C}} & = i m_{i-1,j+1} - j m_{i+1,j-1} \end{cases} \quad (\text{III.12})$$

Le coefficient β vaut 1 ou 0 selon que les moments sont respectivement définis par les équations (III.10), ou (III.11). Notons que la matrice d'interaction $\mathcal{L}_{m_{ij}}$ dépend des moments d'ordre inférieur ou égal à $i + j + 1$. Enfin, $\mathcal{L}_{m_{ij}}$ est une fonction *implicite* de la profondeur $z(q, t)$ de l'objet \mathcal{O} observé, puisqu'elle nécessite la détermination des composantes A , B et C définissant l'équation du plan.

III.2.3.2 Les moments centrés

Pour un objet \mathcal{O} de la scène qui se projette en une région \mathbf{R} dans l'image de centre de gravité (X_g, Y_g) (cf. figure III.2), les moments centrés de l'image μ_{ij} d'ordre $i + j$ sont définis par :

$$\mu_{ij} = \iint_{\mathbf{R}} (X - X_g)^i (Y - Y_g)^j dX dY \quad (\text{III.13})$$

Comme précédemment, en considérant un ensemble de n points $P_k(X_k, Y_k)$ de la région \mathbf{R} , on peut également définir les moments centrés μ_{ij} sous la forme discrète suivante :

$$\mu_{ij} = \sum_{k=1}^n (X_k - X_g)^i (Y_k - Y_g)^j \quad (\text{III.14})$$

Les moments centrés μ_{ij} peuvent s'exprimer en fonction des moments m_{kl} comme suit :

$$\mu_{ij} = \sum_{k=0}^i \sum_{l=0}^j \binom{i}{k} \binom{j}{l} (-X_g)^{i-k} (-Y_g)^{j-l} m_{kl}$$

La variation $\dot{\mu}_{ij}$ des moments centrés se déduit de celle des moments classiques par la relation ci-après [CHAUMETTE, 2004] :

$$\dot{\mu}_{ij} = \mathcal{L}_{\mu_{ij}} \mathbf{T}_{C/\mathcal{R}_0}$$

où $\mathcal{L}_{\mu_{ij}}$ s'écrit [TAHRI et CHAUMETTE, 2005] :

$$\mathcal{L}_{\mu_{ij}} = \begin{pmatrix} \mu_{V_{\bar{x}C}} \\ \mu_{V_{\bar{y}C}} \\ \mu_{V_{\bar{z}C}} \\ \mu_{\Omega_{\bar{x}C}} \\ \mu_{\Omega_{\bar{y}C}} \\ \mu_{\Omega_{\bar{z}C}} \end{pmatrix}^T = \begin{cases} \mu_{V_{\bar{x}C}} &= -(i + \beta) A \mu_{i,j} - i B \mu_{i-1,j+1} \\ \mu_{V_{\bar{y}C}} &= -j A \mu_{i+1,j-1} - (j + \beta) B \mu_{i,j} \\ \mu_{V_{\bar{z}C}} &= -A \mu_{\Omega_{\bar{x}C}} + B \mu_{\Omega_{\bar{y}C}} (i + j + 2\beta) C \mu_{i,j} \\ \mu_{\Omega_{\bar{x}C}} &= (i + j + 3\beta) \mu_{i,j+1} + i X_g \mu_{i-1,j+1} \\ &\quad + (i + 2j + 3\beta) Y_g \mu_{i,j} - \gamma i \frac{\mu_{11}}{m_{00}} \mu_{i-1,j} - \gamma j \frac{\mu_{02}}{m_{00}} \mu_{i,j-1} \\ \mu_{\Omega_{\bar{y}C}} &= -(i + j + 3\beta) \mu_{i+1,j} - i (2i + j + 3\beta) X_g \mu_{i,j} \\ &\quad - j Y_g \mu_{i+1,j-1} + \gamma i \frac{\mu_{20}}{m_{00}} \mu_{i-1,j} + \gamma j \frac{\mu_{11}}{m_{00}} \mu_{i,j-1} \\ \mu_{\Omega_{\bar{z}C}} &= i \mu_{i-1,j+1} - j \mu_{i+1,j-1} \end{cases} \quad (\text{III.15})$$

où $\beta = 1$ et $\gamma = 4$ dans le cas où les moments sont définis par (III.13), et $\beta = 0$ et $\gamma = 1$ s'ils sont donnés par la relation (III.14). Tout comme précédemment, la matrice d'interaction $\mathcal{L}_{\mu_{ij}}$ associée au moment centré μ_{ij} dépend *implicitement* de la profondeur $z(q, t)$ de l'objet \mathcal{O} observé par le biais des paramètres A , B et C .

III.2.3.3 Choix du vecteur d'informations visuelles

Différents vecteurs d'informations visuelles $\mathbf{s}(q, t)$ peuvent être construits sur la base des moments de l'image pour la réalisation de tâches référencées vision [CORKE et HUTCHINSON, 2001], [CHAUMETTE, 2004], [TAHRI et CHAUMETTE, 2005], [MANSARD, 2006], etc. Néanmoins, il est en général souhaitable de ne considérer que les moments d'ordre inférieur à trois, afin de réduire la sensibilité au bruit. Une première idée serait d'utiliser ces six premiers moments pour contrôler les six degrés de liberté de la caméra, ce qui conduirait à choisir $\mathbf{s} = (m_{00}, m_{10}, m_{01}, \mu_{20}, \mu_{11}, \mu_{02})^T$. Cependant, dans ce cas, il s'avère que la matrice d'interaction $\mathcal{L}(\mathbf{s}, \mathbf{z})$ associée à \mathbf{s} est mal conditionnée, et la loi de commande associée est instable [CHAUMETTE, 2004]. Classiquement, un choix convenable consiste à sélectionner au moins les moments :

- d'ordre 0 : m_{00} , c'est-à-dire l'aire \mathcal{A} de la région \mathbf{R} de l'image ;
- d'ordre 1 : m_{10} et m_{01} , qui sont liés aux coordonnées du centre de gravité $X_g (= \frac{m_{10}}{m_{00}})$ et $Y_g (= \frac{m_{01}}{m_{00}})$ de \mathbf{R} .

La matrice d'interaction correspondante est alors facilement calculable à partir de la forme générale (III.12) (*cf.* remarque III.2). Puis, il s'agit de combiner des moments d'ordre supérieur permettant un bon conditionnement de la matrice d'interaction résultante de manière à contraindre autant de degrés de liberté que nécessaire. Il peut être ainsi intéressant de considérer l'orientation principale α de la région \mathbf{R} de l'image (*cf.* figure III.2), définie classiquement par :

$$\alpha = \frac{1}{2} \arctan \frac{2\mu_{11}}{\mu_{20} - \mu_{02}}$$

Il est également possible de combiner des moments de manière à conférer de bonnes propriétés à la loi de commande (*e.g.* découplage des mouvements de translation et de rotation). Dans ce contexte, les invariants de Hu [HU, 1962] (invariants aux translations, rotations et changement d'échelle) sont souvent utilisés. Pour davantage de détails sur ces derniers, le lecteur intéressé peut se référer à [CHAUMETTE, 2004] et à [TAHRI et CHAUMETTE, 2005].

III.2.3.4 Mise en équation du problème

Nous supposons maintenant que nous avons sélectionné un ensemble cohérent de primitives visuelles définies à partir des moments de l'image. Notons $\mathbf{s} = (m_1, \dots, m_n)^T$ le vecteur des indices visuels ainsi obtenu. Dans ce cas, $\dot{\mathbf{s}}$ s'écrit :

$$\dot{\mathbf{s}} = \begin{pmatrix} \dot{m}_1 \\ \vdots \\ \dot{m}_n \end{pmatrix} = \begin{pmatrix} \mathcal{L}(m_1, A, B, C) \\ \vdots \\ \mathcal{L}(m_n, A, B, C) \end{pmatrix} \mathcal{T}_{C/\mathcal{R}_0} = \mathcal{L}(m_1, \dots, m_n, A, B, C) \mathcal{T}_{C/\mathcal{R}_0} \quad (\text{III.16})$$

où $\mathcal{L}(m_k, A, B, C)$ peut se déduire des formes générales (III.12) et (III.15) (*cf.* remarque III.2) selon les moments m_k choisis. Ainsi, ici, reconstruire $\mathbf{s} = (m_1, \dots, m_n)^T$ reviendrait, a priori, à intégrer le système d'équations différentielles (III.16). Cela nous ramènerait alors à une situation identique aux primitives géométriques présentées au paragraphe III.2.2. Cependant, l'expression de la variation d'un moment m_{ij} d'ordre $i + j$ nécessite l'évaluation du moment d'ordre $i + j + 1$ [CHAUMETTE, 2004]. Dès que l'image devient inaccessible, il paraît alors difficile d'exprimer les moments m_{ij} sur la seule base du système dynamique (III.16). Une solution alternative consiste à déterminer

l'expression des moments m_{ij} à partir d'un ensemble de l points appartenant au contour \mathbf{C} de la région \mathbf{R} d'intérêt (cf. remarque III.3). Nous proposons alors de procéder selon les étapes définies par l'algorithme III.2.

Algorithme III.2 Reconstruction des moments de l'image.

- 1: Considérer un ensemble de l points appartenant au contour \mathbf{C} de l'objet dans l'image, et définir ainsi un vecteur $\boldsymbol{\psi} = (X_1, Y_1, z_1, \dots, X_l, Y_l, z_l)^T$ comme dans le paragraphe III.2.1.
 - 2: Intégrer le système d'équations différentielles (III.6) et en déduire une estimation $\hat{\boldsymbol{\psi}}$.
 - 3: À l'aide de l'estimée $\hat{\boldsymbol{\psi}}$, calculer les différents moments m_{ij} nécessaires (cf. remarque III.3), et en déduire le vecteur des indices visuels $\mathbf{s} = (m_1, \dots, m_n)^T$.
-

Remarque III.3: *Il convient de choisir judicieusement l'ensemble des l points du contour \mathbf{C} . Dans le cas particulier de formes polygonales, ces points peuvent être définis par les l sommets du polygone considéré, et les moments correspondants peuvent être déterminés comme dans [SINGER, 1993] ou [STEGER, 1996]. Pour des formes plus élaborées il est possible d'approximer le contour par un polygone et d'obtenir une estimation des moments m_{ij} par le même procédé. Notons enfin qu'il est également possible de déterminer analytiquement les moments de l'image pour certaines primitives géométriques simples, telles que les cercles ou les ellipses.*

III.2.4 Bilan et modélisation du problème

Nous avons considéré le problème de la reconstruction de différents types de primitives visuelles à partir de la relation existant entre le mouvement de ces dernières dans l'image et celui de la caméra. Pour cela, il est nécessaire de disposer d'un modèle de la matrice d'interaction $\mathcal{L}(\mathbf{s}, \mathbf{z})$ dont tous les paramètres peuvent être caractérisés, et du mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$. Nous avons mis en exergue trois cas :

- ① $\mathcal{L}(\mathbf{s}, \mathbf{z})$ dépend *explicitement* de la profondeur $\mathbf{z}(q, t)$, et nous supposons qu'il est possible de relier sa variation $\dot{\mathbf{z}}$ au mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$. Nous pouvons alors exploiter directement les résultats obtenus dans le cas du point et déduire le vecteur des indices visuels \mathbf{s} en résolvant un système d'équations différentielles d'une forme similaire à (III.6).
- ② $\mathcal{L}(\mathbf{s}, \mathbf{z})$ dépend *implicitement* de la profondeur $\mathbf{z}(q, t)$, par le biais des composantes A_{pq} définies par (III.7). Il faut alors utiliser les résultats obtenus dans le cas du point pour reconstruire la profondeur $\mathbf{z}(q, t)$ et caractériser les paramètres A_{pq} . Il est ensuite possible de déterminer les informations visuelles initiales en résolvant (III.9).
- ③ Il est difficile d'obtenir tous les paramètres de $\mathcal{L}(\mathbf{s}, \mathbf{z})$ nécessaires au processus de reconstruction. C'est notamment le cas des moments de l'image. Le problème d'estimation peut néanmoins se résoudre s'il est possible de déterminer \mathbf{s} à partir de primitives visuelles plus simples (le point par exemple) qu'il suffit alors de calculer en résolvant le système d'équations différentielles correspondant. Nous nous ramenons ainsi aux cas ① ou ②.

De ce fait, pour reconstruire le motif visuel $\mathbf{s}(q, t)$ nous proposons de résoudre un système dynamique de la forme suivante :

$$\dot{\boldsymbol{\chi}} = \mathcal{L}(\boldsymbol{\chi})\mathcal{T}_{C/\mathcal{R}_0} = \varphi(\boldsymbol{\chi}, t) \quad (\text{III.17})$$

où :

- dans le cas ① : $\boldsymbol{\chi} = (\mathbf{s}^T, \mathbf{z}^T)^T$ et $\varphi(\boldsymbol{\chi}, t)$ se déduit de manière similaire à (III.6). Nous obtenons alors directement le motif visuel perdu.
- dans le cas ② : nous effectuons deux processus de reconstruction :
 1. Détermination des paramètres A_{pq} à partir de l points. On pose alors $\boldsymbol{\chi} = (X_1, Y_1, z_1, \dots, X_l, Y_l, z_l)^T$ et $\varphi(\boldsymbol{\chi}, t)$ est directement donné par (III.6).
 2. Estimation des primitives visuelles d'intérêt en considérant à présent $\boldsymbol{\chi} = (\pi_1, \dots, \pi_n)^T$, $\varphi(\boldsymbol{\chi}, t)$ s'obtenant de (III.9),
- dans le cas ③ : il se déduit soit du cas ①, soit du cas ② selon le type de primitives (points ou autres formes géométriques) utilisées pour retrouver les indices visuels d'intérêt.

Le bilan précédent a donc montré que la reconstruction du signal visuel lorsqu'il est perdu peut se ramener à la résolution du système d'équations différentielles (III.17). D'une manière plus précise nous considérons le problème suivant :

$$\begin{cases} \dot{\boldsymbol{\chi}}(t) = \varphi(\boldsymbol{\chi}, t) \\ \boldsymbol{\chi}_0 = \boldsymbol{\chi}(t_0) \end{cases} \quad \text{avec : } t_0 \leq t \leq t_{\text{fin}} < +\infty \quad (\text{III.18})$$

où t_0 et t_{fin} représentent respectivement les instants initial et final correspondant au début et à la fin de la tâche robotique. Le vecteur $\boldsymbol{\chi}_0$ définit les conditions initiales du système d'équations différentielles à l'instant t_0 . Il est donné par le motif visuel original \mathbf{s}_0 , et les différentes données complémentaires nécessaires, soit :

- dans le cas ① : la profondeur \mathbf{z}_0 ;
- dans le cas ② : les composantes A_{pq} .

Comme nous l'avons déjà mentionné, nous supposons que le signal perceptuel \mathbf{s} est toujours disponible dans l'image au début de la tâche, et qu'il est toujours possible d'extraire les l points nécessaires au calcul de A_{pq} à l'instant t_0 . De même, la profondeur \mathbf{z}_0 peut aussi être considérée connue, dans la mesure où elle peut être caractérisée en utilisant une procédure hors-ligne robuste (mesure physique, prises de vues multiples...) effectuée avant le lancement de la mission. Nous pouvons donc admettre que la valeur de $\boldsymbol{\chi}_0$ est toujours disponible.

La résolution directe du système (III.17) entre t_0 et t_{fin} s'avérant difficile, nous avons choisi d'intégrer ces équations sur la période d'échantillonnage de l'asservissement T_{ech} . Il s'agit donc pour nous de résoudre le système (III.17) pour tout t appartenant à l'intervalle $[t_k, t_{k+1}]$ avec $t_k = t_0 + kT_{\text{ech}}$ et $t_{k+1} = t_0 + (k+1)T_{\text{ech}}$ en prenant comme conditions initiales $\boldsymbol{\chi}_k = \boldsymbol{\chi}(t_k)$ (cf. figure III.3). Nous considérons donc finalement le problème suivant :

$$\begin{cases} \dot{\boldsymbol{\chi}}(t) = \varphi(\boldsymbol{\chi}, t) \\ \boldsymbol{\chi}_k = \boldsymbol{\chi}(t_k) \end{cases} \quad \text{avec : } t_k \leq t \leq t_{k+1} \quad (\text{III.19})$$

Le vecteur $\boldsymbol{\chi}_k = \boldsymbol{\chi}(t_k)$ est défini par le motif visuel \mathbf{s}_k et les informations complémentaires nécessaires (\mathbf{z}_k ou composantes A_{pq}) à l'instant t_k . Nous supposons ce vecteur connu, soit directement à partir des informations capteurs si l'image est disponible (cf. remarque III.4), soit reconstruit par notre estimateur si celle-ci n'est plus accessible. Il est alors possible de déduire $\boldsymbol{\chi}_{k+1} = \boldsymbol{\chi}(t_{k+1})$ connaissant $\boldsymbol{\chi}_k$ et le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ appliqué au moment t_k .

Remarque III.4: Rappelons que, si l'image n'est pas perturbée, \mathbf{s}_k peut être obtenue à l'aide du traitement d'images embarqué et z_k déterminée par des méthodes basées sur le SFM, le filtrage de Kalman [MATTHIES *et al.*, 1989] ou encore l'estimation relative de pose [OBERKAMPF *et al.*, 1996], [THRUN *et al.*, 2001], *etc.* Notons que, χ_0 étant connue, il est aussi possible de déterminer le vecteur χ_k en appliquant itérativement l'estimateur dès le début de la mission (même si l'image est encore disponible à l'instant t_k).

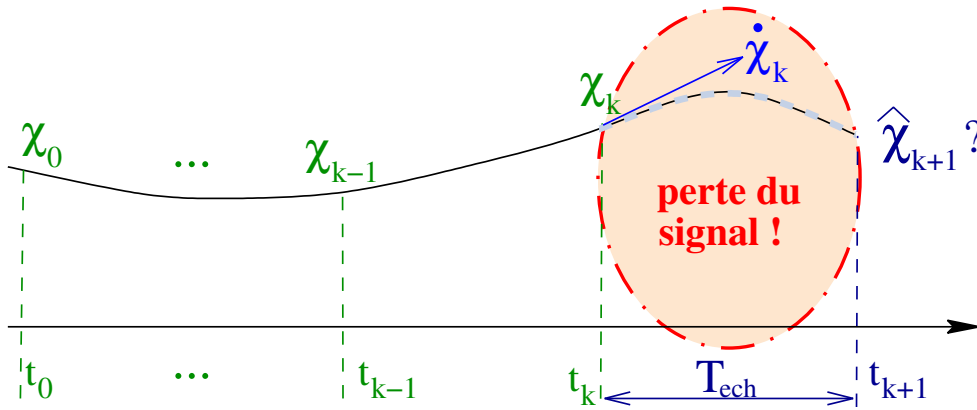


FIG. III.3 – Estimation de χ_{k+1} .

Finalement, nous avons montré que la reconstruction du signal visuel peut se ramener à la résolution du problème (III.19). Nous cherchons donc à intégrer les équations (III.17) pour tout t appartenant à l'intervalle $[t_k, t_{k+1}]$ en prenant comme conditions initiales $\chi_k = \chi(t_k)$ (*cf.* figure III.3). Afin que le problème soit bien posé, nous supposons que la fonction $\varphi(\chi, t)$ (III.17) est une fonction continue et qu'elle satisfait la condition de Lipschitz pour la variable χ (*cf.* théorème B.2 en annexe B). En effet, dans ce cas, le théorème B.1 de CAUCHY–LIPSCHITZ⁴ garantit l'existence d'une unique solution locale au système d'équations différentielles considéré.

Nous proposons dans la suite des techniques permettant de résoudre le problème (III.19). Néanmoins, le système d'équations (III.17) dépendant à la fois de la nature des primitives visuelles \mathbf{s} considérées et du type de robot employé, il est difficile de proposer des solutions pour le cas général. C'est pourquoi, nous nous sommes dans un premier temps focalisés sur une primitive particulière : le point. Nous formulerons dans ce cas trois solutions analytiques selon que le torseur cinématique peut être considéré constant, quasi-constant ou variable sur l'intervalle $[t_k, t_{k+1}]$. Ensuite, nous proposerons des solutions numériques permettant de traiter les cas plus complexes.

4. Le théorème B.1 de CAUCHY–LIPSCHITZ est également connu dans de nombreux pays sous le nom du théorème de PICARD–LINDELÖF, du nom des mathématiciens Émile PICARD (1856-1941) et Ernst LINDELÖF (1870-1946).

III.3 Le cas du point : vers une résolution analytique

Nous nous focalisons dans cette partie sur une primitive particulière : le point. En effet, nous avons montré précédemment l'intérêt de cette primitive dans la reconstruction de la profondeur $z(q, t)$, et de certains indices visuels tels que les centres de gravité, les segments, les polygones, etc. Nous considérons donc un point $p(x, y, z)$ de la scène projeté en $P(X, Y)$ sur le plan image. Comme nous l'avons préalablement énoncé la dynamique du triplé $\boldsymbol{\chi} = (X, Y, z)^T$ d'un point P de l'amer peut être décrite par les relations (I.10) et (III.4), et se réécire sous forme scalaire comme suit :

$$\dot{X} = -\frac{f}{z(t)}V_{\vec{x}_C}(t) + \frac{X(t)}{z(t)}V_{\vec{z}_C}(t) + \frac{X(t)Y(t)}{f}\Omega_{\vec{x}_C}(t) - \left(f + \frac{X(t)^2}{f}\right)\Omega_{\vec{y}_C}(t) + Y(t)\Omega_{\vec{z}_C}(t) \quad (\text{III.20})$$

$$\dot{Y} = -\frac{f}{z(t)}V_{\vec{y}_C}(t) + \frac{Y(t)}{z(t)}V_{\vec{z}_C}(t) + \left(f + \frac{Y(t)^2}{f}\right)\Omega_{\vec{x}_C}(t) - \frac{X(t)Y(t)}{f}\Omega_{\vec{y}_C}(t) - X(t)\Omega_{\vec{z}_C}(t) \quad (\text{III.21})$$

$$\dot{z} = -V_{\vec{z}_C}(t) - \frac{z(t)Y(t)}{f}\Omega_{\vec{x}_C}(t) + \frac{z(t)X(t)}{f}\Omega_{\vec{y}_C}(t) \quad (\text{III.22})$$

Notre intention est d'établir, à partir du système dynamique ci-dessus, l'expression analytique du triplé $\boldsymbol{\chi}(t) = (X(t), Y(t), z(t))^T$. Pour cela, nous nous proposons d'intégrer les équations précédentes pour tout $t \in [t_k; t_{k+1}]$. Nous considérons successivement trois cas selon que le torseur cinématique de la caméra $\boldsymbol{\mathcal{T}}_{C/\mathcal{R}_0}$ est considéré constant, quasi-constant (*i.e.* constant par morceaux) ou variable sur cet intervalle.

III.3.1 Résolution dans le cas du torseur constant

Dans un premier temps, nous nous proposons de résoudre le système d'équations différentielles précédent en considérant que le mouvement de la caméra décrit par son torseur cinématique $\boldsymbol{\mathcal{T}}_{C/\mathcal{R}_0} = (V_{C/\mathcal{R}_0}^T, \Omega_{\mathcal{R}_C/\mathcal{R}_0}^T)^T$ est *constant* sur l'intervalle $[t_k; t_{k+1}]$. Bien que cela ne soit pas toujours le cas, cette première approximation est en général acceptable pour des périodes d'échantillonnage T_{ech} suffisamment faibles. Ainsi, pour tout $t \in [t_k; t_{k+1}]$, les variations \dot{X} , \dot{Y} et \dot{z} peuvent se réécrire de la manière suivante :

$$\dot{X} = -\frac{f}{z(t)}V_{\vec{x}_{Ck}} + \frac{X(t)}{z(t)}V_{\vec{z}_{Ck}} + \frac{X(t)Y(t)}{f}\Omega_{\vec{x}_{Ck}} - \left(f + \frac{X(t)^2}{f}\right)\Omega_{\vec{y}_{Ck}} + Y(t)\Omega_{\vec{z}_{Ck}} \quad (\text{III.23})$$

$$\dot{Y} = -\frac{f}{z(t)}V_{\vec{y}_{Ck}} + \frac{Y(t)}{z(t)}V_{\vec{z}_{Ck}} + \left(f + \frac{Y(t)^2}{f}\right)\Omega_{\vec{x}_{Ck}} - \frac{X(t)Y(t)}{f}\Omega_{\vec{y}_{Ck}} - X(t)\Omega_{\vec{z}_{Ck}} \quad (\text{III.24})$$

$$\dot{z} = -V_{\vec{z}_{Ck}} - \frac{z(t)Y(t)}{f}\Omega_{\vec{x}_{Ck}} + \frac{z(t)X(t)}{f}\Omega_{\vec{y}_{Ck}} \quad (\text{III.25})$$

en posant :

$$\boldsymbol{\mathcal{T}}_{C/\mathcal{R}_0}(t_k) : \begin{cases} V_{\vec{x}_{Ck}} = V_{\vec{x}_C}(t_k), & V_{\vec{y}_{Ck}} = V_{\vec{y}_C}(t_k), & V_{\vec{z}_{Ck}} = V_{\vec{z}_C}(t_k) \\ \Omega_{\vec{x}_{Ck}} = \Omega_{\vec{x}_C}(t_k), & \Omega_{\vec{y}_{Ck}} = \Omega_{\vec{y}_C}(t_k), & \Omega_{\vec{z}_{Ck}} = \Omega_{\vec{z}_C}(t_k) \end{cases}$$

Remarque III.5: Afin d'alléger les notations ultérieures les composantes $V_{\vec{x}_{Ck}}$, $V_{\vec{y}_{Ck}}$, $V_{\vec{z}_{Ck}}$, $\Omega_{\vec{x}_{Ck}}$, $\Omega_{\vec{y}_{Ck}}$ et $\Omega_{\vec{z}_{Ck}}$ du torseur $\boldsymbol{\mathcal{T}}_{C/\mathcal{R}_0}$ à l'instant t_k seront respectivement notées $V_{\vec{x}_C}$, $V_{\vec{y}_C}$, $V_{\vec{z}_C}$, $\Omega_{\vec{x}_C}$, $\Omega_{\vec{y}_C}$ et $\Omega_{\vec{z}_C}$.

III.3.1.1 Détermination de $z(t)$

Dans un premier temps, intéressons-nous au calcul de $z(t)$. En dérivant l'équation différentielle (III.25), nous obtenons :

$$\ddot{z} = -\left(\dot{Y}z(t) + Y(t)\dot{z}\right)\frac{\Omega_{\vec{x}_C}}{f} + \left(\dot{X}z(t) + X(t)\dot{z}\right)\frac{\Omega_{\vec{y}_C}}{f} \quad (\text{III.26})$$

En multipliant (III.25) et (III.24) respectivement par $Y(t)$ et $z(t)$, et en sommant les équations obtenues il vient :

$$\dot{Y}z(t) + Y(t)\dot{z} = -fV_{\vec{y}_C} + fz(t)\Omega_{\vec{x}_C} - z(t)X(t)\Omega_{\vec{z}_C} \quad (\text{III.27})$$

De même, en multipliant (III.25) et (III.23) respectivement par $X(t)$ et $z(t)$ et en sommant, nous avons :

$$\dot{X}z(t) + X(t)\dot{z} = -fV_{\vec{x}_C} - fz(t)\Omega_{\vec{y}_C} + z(t)Y(t)\Omega_{\vec{z}_C} \quad (\text{III.28})$$

En remplaçant (III.27) et (III.28) dans (III.26), nous pouvons déduire l'expression de \ddot{z} suivante :

$$\ddot{z} = \Omega_{\vec{x}_C}V_{\vec{y}_C} - \Omega_{\vec{y}_C}V_{\vec{x}_C} - z(t)(\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) + \frac{\Omega_{\vec{z}_C}}{f}(z(t)X(t)\Omega_{\vec{x}_C} + z(t)Y(t)\Omega_{\vec{y}_C}) \quad (\text{III.29})$$

Dérivons maintenant une seconde fois l'équation différentielle ci-dessus. Il vient :

$$\ddot{\ddot{z}} = -\dot{z}(\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) + \frac{\Omega_{\vec{z}_C}}{f}\left(\Omega_{\vec{x}_C}(\dot{X}z + X\dot{z}) + \Omega_{\vec{y}_C}(\dot{Y}z + Y\dot{z})\right)$$

En réutilisant les expressions données par (III.27) et (III.28), $\ddot{\ddot{z}}$ peut se réduire à :

$$\ddot{\ddot{z}} + \dot{z}(\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2 + \Omega_{\vec{z}_C}^2) = -\Omega_{\vec{z}_C}(V_{\vec{x}_C}\Omega_{\vec{x}_C} + V_{\vec{y}_C}\Omega_{\vec{y}_C} + V_{\vec{z}_C}\Omega_{\vec{z}_C}) \quad (\text{III.30})$$

Posons : $\zeta(t) = \dot{z}$. L'équation (III.30) peut alors se réécrire sous la forme d'une équation différentielle du second ordre à coefficients constants avec un second membre constant, soit :

$$\ddot{\zeta} + A_1^2\zeta(t) = A_2 \quad \text{avec : } \begin{cases} A_1 = \sqrt{\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2 + \Omega_{\vec{z}_C}^2} \\ A_2 = -\Omega_{\vec{z}_C}(V_{\vec{x}_C}\Omega_{\vec{x}_C} + V_{\vec{y}_C}\Omega_{\vec{y}_C} + V_{\vec{z}_C}\Omega_{\vec{z}_C}) \end{cases} \quad (\text{III.31})$$

La résolution de cette équation différentielle du second ordre (III.31) nécessite de résoudre l'équation homogène :

$$\ddot{\zeta} + A_1^2\zeta(t) = 0, \quad (\text{III.32})$$

Deux cas peuvent se poser :

- ① Si $A_1 > 0$ (*i.e.* $\Omega_{\vec{x}_C} \neq 0$, $\Omega_{\vec{y}_C} \neq 0$ et $\Omega_{\vec{z}_C} \neq 0$), alors la solution générale $\zeta(t)$ de l'équation différentielle (III.31) s'écrit :

$$\zeta(t) = c_1 \sin(A_1(t - t_k)) + c_2 \cos(A_1(t - t_k)) + \frac{A_2}{A_1^2}$$

La solution générale de l'équation différentielle (III.25) s'obtient alors en intégrant $\zeta(t)$ entre t_k et t . Il vient donc :

$$\begin{aligned} \int_{t_k}^t \zeta(\tau) d\tau &= \int_{t_k}^t \left(c_1 \sin(A_1(\tau - t_k)) + c_2 \cos(A_1(\tau - t_k)) + \frac{A_2}{A_1^2} \right) d\tau \\ [z(\tau)]_{t_k}^t &= \left[-\frac{c_1}{A_1} \cos(A_1(\tau - t_k)) + \frac{c_2}{A_1} \sin(A_1(\tau - t_k)) + \frac{A_2}{A_1^2} \tau \right]_{t_k}^t \end{aligned}$$

Soit finalement :

$$z(t) = -\frac{c_1}{A_1} \cos(A_1(t - t_k)) + \frac{c_2}{A_1} \sin(A_1(t - t_k)) + \frac{A_2}{A_1^2}(t - t_k) + \frac{c_1}{A_1} + z_k \quad (\text{III.33})$$

Les constantes c_1 et c_2 se déduisent des conditions initiales du système dynamique à l'instant t_k , et à partir des dérivées \dot{z} et \ddot{z} de la relation ci-dessus (III.33). Il vient :

$$\begin{aligned} c_1 &= \ddot{z}_k \\ c_2 &= \dot{z}_k - \frac{A_2}{A_1^2} \end{aligned}$$

où :

$$\begin{aligned} \dot{z}_k &= -V_{\vec{z}_C} - \frac{z_k Y_k}{f} \Omega_{\vec{x}_C} + \frac{z_k X_k}{f} \Omega_{\vec{y}_C} \\ \ddot{z}_k &= \Omega_{\vec{x}_C} V_{\vec{y}_C} - \Omega_{\vec{y}_C} V_{\vec{x}_C} - z_k (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) + z_k \frac{\Omega_{\vec{z}_C}}{f} (X_k \Omega_{\vec{x}_C} + Y_k \Omega_{\vec{y}_C}) \end{aligned}$$

- ② Si $A_1 = 0$, c'est-à-dire si $\Omega_{\vec{x}_C} = \Omega_{\vec{y}_C} = \Omega_{\vec{z}_C} = 0$ (pas de rotation), l'équation différentielle (III.25) se simplifie en :

$$\dot{z} = -V_{\vec{z}_C}$$

La solution pour la profondeur $z(t)$ dans le cas particulier où il n'y a aucune rotation est alors donnée par :

$$z(t) = -V_{\vec{z}_C}(t - t_k) + z_k \quad (\text{III.34})$$

III.3.1.2 Détermination de $X(t)$ et de $Y(t)$

Intéressons-nous à présent au calcul de $X(t)$ et de $Y(t)$. L'équation (III.25) peut se réécrire de la manière suivante :

$$-z(t)Y(t) = \frac{f\dot{z} + fV_{\vec{z}_C} - z(t)X(t)\Omega_{\vec{y}_C}}{\Omega_{\vec{x}_C}} \quad (\text{III.35})$$

$$z(t)X(t) = \frac{f\dot{z} + fV_{\vec{z}_C} + z(t)Y(t)\Omega_{\vec{x}_C}}{\Omega_{\vec{y}_C}} \quad (\text{III.36})$$

En substituant (III.35) et (III.36) dans l'expression de la dérivée seconde de la profondeur (III.29), nous avons :

$$\begin{aligned} \ddot{z} - \Omega_{\vec{x}_C} V_{\vec{y}_C} + \Omega_{\vec{y}_C} V_{\vec{x}_C} + z(t) (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) &= \frac{\Omega_{\vec{z}_C}}{f\Omega_{\vec{x}_C}} \left(z(t)X(t) (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) - \Omega_{\vec{y}_C} f (\dot{z} + V_{\vec{z}_C}) \right) \\ \ddot{z} - \Omega_{\vec{x}_C} V_{\vec{y}_C} + \Omega_{\vec{y}_C} V_{\vec{x}_C} + z(t) (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) &= \frac{\Omega_{\vec{z}_C}}{f\Omega_{\vec{y}_C}} \left(z(t)Y(t) (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) + \Omega_{\vec{x}_C} f (\dot{z} + V_{\vec{z}_C}) \right) \end{aligned}$$

Différentes situations sont alors possibles :

- ① Si $\Omega_{\vec{x}_C} \neq 0$, $\Omega_{\vec{y}_C} \neq 0$ et $\Omega_{\vec{z}_C} \neq 0$, alors la solution des équations différentielles (III.23) et (III.24) se déduit simplement des deux relations précédentes :

$$\begin{aligned} X(t) &= \frac{f\Omega_{\vec{x}_C} \left(\ddot{z} - \Omega_{\vec{x}_C} V_{\vec{y}_C} + \Omega_{\vec{y}_C} V_{\vec{x}_C} + z(t) (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) \right) + \Omega_{\vec{z}_C} \Omega_{\vec{y}_C} f (\dot{z} + V_{\vec{z}_C})}{z(t)\Omega_{\vec{z}_C} (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2)} \\ Y(t) &= \frac{f\Omega_{\vec{y}_C} \left(\ddot{z} - \Omega_{\vec{x}_C} V_{\vec{y}_C} + \Omega_{\vec{y}_C} V_{\vec{x}_C} + z(t) (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) \right) - \Omega_{\vec{z}_C} \Omega_{\vec{x}_C} f (\dot{z} + V_{\vec{z}_C})}{z(t)\Omega_{\vec{z}_C} (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2)} \end{aligned}$$

Connaissant l'expression de $z(t)$ donnée par (III.33), nous pouvons calculer facilement \dot{z} et \ddot{z} . Les solutions générales de $X(t)$ et de $Y(t)$ s'écrivent alors comme suit :

$$X(t) = \frac{\Omega_{\vec{x}_C} f A_3 + \Omega_{\vec{y}_C} A_4}{\Omega_{\vec{z}_C} (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2)} z(t) \quad (\text{III.37})$$

$$Y(t) = \frac{\Omega_{\vec{y}_C} f A_3 - \Omega_{\vec{x}_C} A_4}{\Omega_{\vec{z}_C} (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2)} z(t) \quad (\text{III.38})$$

avec :

$$\begin{aligned} A_3 &= c_1 A_1 \cos(A_1(t - t_k)) - c_2 A_1 \sin(A_1(t - t_k)) - \Omega_{\vec{x}_C} V_{\vec{y}_C} + \Omega_{\vec{y}_C} V_{\vec{x}_C} + z(t) (\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2) \\ A_4 &= \Omega_{\vec{z}_C} f \left(c_1 \sin(A_1(t - t_k)) + c_2 \cos(A_1(t - t_k)) + \frac{A_2}{A_1^2} + V_{\vec{z}_C} \right) \end{aligned}$$

- ② Si $\Omega_{\vec{x}_C} = \Omega_{\vec{y}_C} = \Omega_{\vec{z}_C} = 0$ (*i.e.* pas de rotation), les équations (III.28) et (III.27) se simplifient en :

$$\begin{aligned} \dot{X}z(t) + X(t)\dot{z} &= -fV_{\vec{x}_C} \\ \dot{Y}z(t) + Y(t)\dot{z} &= -fV_{\vec{y}_C} \end{aligned}$$

En intégrant ces équations entre t_k et t il vient :

$$\begin{aligned} \int_{t_k}^t (\dot{X}z(\tau) + X(\tau)\dot{z}) d\tau &= -fV_{\vec{x}_C} \int_{t_k}^t 1 d\tau \Leftrightarrow [X(\tau)z(\tau)]_{t_k}^t = -fV_{\vec{x}_C} [\tau]_{t_k}^t \\ \int_{t_k}^t (\dot{Y}z(\tau) + Y(\tau)\dot{z}) d\tau &= -fV_{\vec{y}_C} \int_{t_k}^t 1 d\tau \Leftrightarrow [Y(\tau)z(\tau)]_{t_k}^t = -fV_{\vec{y}_C} [\tau]_{t_k}^t \end{aligned}$$

Finalement la solution des équations différentielles (III.23) et (III.24) dans le cas particulier où il n'y a pas de rotation est ainsi donnée par :

$$\begin{aligned} X(t) &= \frac{-fV_{\vec{x}_C}(t - t_k) + X_k z_k}{z(t)} \\ Y(t) &= \frac{-fV_{\vec{y}_C}(t - t_k) + Y_k z_k}{z(t)} \end{aligned}$$

où $z(t)$ est donnée par la relation (III.34).

- ③ Dans le cas où $\Omega_{\vec{x}_C} \neq 0$ et $\Omega_{\vec{y}_C} \neq 0$, mais $\Omega_{\vec{z}_C} = 0$, les équations (III.28) et (III.27) se simplifient désormais en :

$$\dot{X}z(t) + X(t)\dot{z} = -fV_{\vec{x}_C} - fz(t)\Omega_{\vec{y}_C} \quad (\text{III.39})$$

$$\dot{Y}z(t) + Y(t)\dot{z} = -fV_{\vec{y}_C} - fz(t)\Omega_{\vec{x}_C} \quad (\text{III.40})$$

Dans ce cas, sachant que $A_1 = \sqrt{\Omega_{\vec{x}_C}^2 + \Omega_{\vec{y}_C}^2}$ et $A_2 = 0$, la relation (III.33) se simplifie en :

$$z(t) = -\frac{c_1}{A_1} \cos(A_1(t - t_k)) + \frac{c_2}{A_1} \sin(A_1(t - t_k)) + \frac{c_1}{A_1} + z_k$$

En intégrant les équations (III.39) et (III.40) entre t_k et t il vient :

$$X(t) = \frac{-fV_{\vec{x}_C}(t-t_k) + X_k z_k - f\Omega_{\vec{y}_C} \left(-\frac{c_1}{A_1^2} \sin(A_1(t-t_k)) - \frac{c_2}{A_1^2} \cos(A_1(t-t_k)) + \left(\frac{c_1}{A_1} + z_k\right)(t-t_k) + \frac{c_2}{A_1^2} \right)}{z(t)}$$

$$Y(t) = \frac{-fV_{\vec{y}_C}(t-t_k) + Y_k z_k - f\Omega_{\vec{x}_C} \left(-\frac{c_1}{A_1^2} \sin(A_1(t-t_k)) - \frac{c_2}{A_1^2} \cos(A_1(t-t_k)) + \left(\frac{c_1}{A_1} + z_k\right)(t-t_k) + \frac{c_2}{A_1^2} \right)}{z(t)}$$

- ④ Si $\Omega_{\vec{x}_C} = 0$ et $\Omega_{\vec{y}_C} = 0$, avec $\Omega_{\vec{z}_C} \neq 0$, dans ce cas les relations (III.27) et (III.28) se réduisent à :

$$\dot{Y}z(t) + Y(t)\dot{z} = -fV_{\vec{y}_C} - z(t)X(t)\Omega_{\vec{z}_C} \Leftrightarrow \frac{d(Yz)}{dt} = -fV_{\vec{y}_C} - \Omega_{\vec{z}_C}X(t)z(t)$$

$$\dot{X}z(t) + X(t)\dot{z} = -fV_{\vec{x}_C} + z(t)Y(t)\Omega_{\vec{z}_C} \Leftrightarrow \frac{d(Xz)}{dt} = -fV_{\vec{x}_C} + \Omega_{\vec{z}_C}Y(t)z(t)$$

Les deux équations ci-dessus peuvent alors se mettre sous la forme matricielle suivante :

$$\begin{pmatrix} \frac{d(Xz)}{dt} \\ \frac{d(Yz)}{dt} \end{pmatrix} = \begin{pmatrix} 0 & \Omega_{\vec{z}_C} \\ -\Omega_{\vec{z}_C} & 0 \end{pmatrix} \begin{pmatrix} X(t)z(t) \\ Y(t)z(t) \end{pmatrix} - f \begin{pmatrix} V_{\vec{x}_C} \\ V_{\vec{y}_C} \end{pmatrix}$$

Nous pouvons en déduire les solutions $X(t)$ et $Y(t)$ ci-après :

$$X(t) = \frac{c_3 \cos(\Omega_{\vec{z}_C}(t-t_k)) + c_4 \sin(\Omega_{\vec{z}_C}(t-t_k)) - f \frac{V_{\vec{y}_C}}{\Omega_{\vec{z}_C}}}{z(t)}$$

$$Y(t) = \frac{-c_3 \sin(\Omega_{\vec{z}_C}(t-t_k)) + c_4 \cos(\Omega_{\vec{z}_C}(t-t_k)) + f \frac{V_{\vec{x}_C}}{\Omega_{\vec{z}_C}}}{z(t)}$$

où $z(t)$ est donnée par la relation (III.34). Les constantes c_3 et c_4 se déduisent à partir des conditions initiales X_k , Y_k et z_k du système dynamique à l'instant t_k , et sont données par :

$$c_3 = -X_k z_k - f \frac{V_{\vec{y}_C}}{\Omega_{\vec{z}_C}}$$

$$c_4 = Y_k z_k - f \frac{V_{\vec{x}_C}}{\Omega_{\vec{z}_C}}$$

III.3.1.3 Bilan

Nous disposons donc des expressions analytiques du triplé $\chi(t) = (X(t), Y(t), z(t))^T$ dans le cas où le torseur cinématique $\mathcal{T}_{C/\mathcal{R}_0}$ est considéré constant sur une période d'échantillonnage T_{ech} de l'asservissement. Connaissant le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ et la valeur de $\chi(t_k)$ à l'instant t_k , il est alors possible de déduire la position et la profondeur du point P pour tout $t \in [t_k; t_{k+1}]$, et ceci *quel que soit* le système robotique portant la caméra.

Cependant, ce résultat repose sur l'hypothèse que $\mathcal{T}_{C/\mathcal{R}_0}$ est constant sur cet intervalle. Cette première approximation n'est réellement satisfaisante que pour de faibles valeurs de T_{ech} , et dans les situations où la dynamique du système porteur de la caméra n'est pas trop importante. Bien qu'aujourd'hui la première hypothèse soit généralement vérifiée sur la plupart des plates-formes expérimentales, la seconde apparaît plus contraignante. Nous avons donc cherché à proposer une solution permettant de nous affranchir de cette condition sur le torseur cinématique.

Remarque III.6: *Les solutions présentées ici nous rapprochent des résultats obtenus en résolvant le système dynamique (III.3) pour le point p de coordonnées $(x, y, z)^T$, soit [SOATTO et PERRONA, 1995, SOATTO et al., 1996] :*

$$\dot{p} = -V_{C/\mathcal{R}_0}(t) - \Omega_{\mathcal{R}_C/\mathcal{R}_0}(t) \wedge p(t) \quad \leftrightarrow \quad p(t_{k+1}) = R(t_k)p(t_k) + T(t_k)$$

où $R \in SO(3)$ ⁵ et $T \in \mathbb{R}^3$ correspondent respectivement à la rotation et à la translation réalisées par le point p sur une période d'échantillonnage. R et T sont liés aux mouvements de rotation $\Omega_{\mathcal{R}_C/\mathcal{R}_0}(t)$ et de translation $V_{C/\mathcal{R}_0}(t)$ par une carte exponentielle (*exponential map*) [MURRAY et al., 1994].

III.3.2 Extension : prise en compte de la variation du torseur cinématique

Nous nous proposons ici d'étendre les résultats précédents afin de prendre en compte l'évolution du torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ sur l'intervalle $[t_k; t_{k+1}]$. Nous proposons deux approches distinctes. Dans la première, nous cherchons à rester indépendant du système robotique porteur, et il est alors difficile de prendre en compte la "vraie" variation du torseur. Nous avons donc proposé une approche où le torseur reste constant par morceaux sur l'intervalle de commande $[t_k; t_{k+1}]$ ce qui permet de bénéficier des expressions analytiques de $X(t)$, $Y(t)$ et $z(t)$ établies précédemment. La seconde approche cherche, quant à elle, à se rapprocher de la réalité où, en toute rigueur, seules les commandes \dot{q} envoyées au robot sont bloquées sur cet intervalle. Il s'agit alors de réécrire et de résoudre le système (III.19) en prenant en compte la jacobienne de l'ensemble porteur. Les expressions obtenues pour X , Y , et z sont alors intrinsèquement liées à ce dernier et uniquement valables pour tout système robotique de structure mécanique similaire. Nous présentons ci-après ces deux approches.

III.3.2.1 Variation par morceaux du torseur cinématique

Notre idée consiste à échantillonner l'intervalle de commande $[t_k, t_{k+1}]$ en N ⁶ sous-intervalles d'intégration de largeur $T_N = \frac{T_{\text{ech}}}{N}$ (cf. figure III.4), sur lesquels nous allons évaluer le torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$. Celui-ci sera donc considéré constant pour tout $t \in [t_n, t_{n+1}]$, où $t_n = t_k + (n - k)T_N$, avec $n = \{k, k + 1, \dots, k + N\}$, mais reste variable sur $[t_k; t_{k+1}]$. Or, comme : $\mathcal{T}_{C/\mathcal{R}_0} = \mathbf{J}(q)\dot{q}$, et que nous considérons les commandes \dot{q} envoyées au système robotique, le calcul de $\mathcal{T}_{C/\mathcal{R}_0}$ à chaque instant t_n passe par la détermination de la jacobienne du robot $\mathbf{J}(q)$ et donc de q à chacun de ces

5. $SO(3)$ désigne le *groupe spécial orthogonal* aussi appelé groupe de rotation.

6. N doit être de préférence choisi *entier* (non nul) afin de ne pas provoquer de problème de chevauchement des différentes périodes d'échantillonnage.

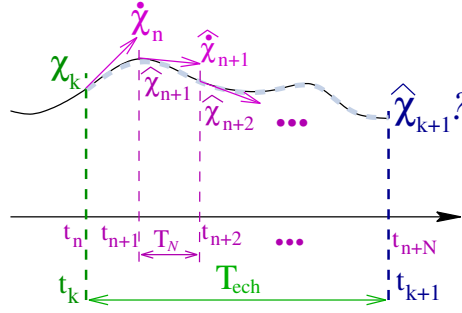


FIG. III.4 – Les intervalles d'intégration.

instants. Pour cela, nous proposons d'intégrer l'équation $\dot{q}(t) = \dot{q}_k$ entre t_k et t . Nous obtenons :

$$q(t) = (t - t_k) \dot{q}_k + q(t_k) \quad \forall t \in [t_k; t_{k+1}] \quad (\text{III.41})$$

où $q(t_k)$ correspond aux valeurs des différents paramètres q caractérisant la configuration de la structure mécanique portant la caméra. Ces valeurs peuvent être obtenues à l'aide des capteurs proprioceptifs (*e.g.* encodeurs embarqués). Connaissant $q(t)$ pour tout $t \in [t_k, t_{k+1}]$, il est alors possible de déduire la jacobienne $\mathbf{J}(q(t))$ et donc le torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}(q(t))$ à l'instant t_n . Celui-ci étant considéré constant sur l'intervalle $[t_n; t_{n+1}]$, les expressions analytiques établies dans le paragraphe précédent III.3.1 peuvent alors être utilisées pour caractériser le triplé $\boldsymbol{\chi}(t) = (X(t), Y(t), z(t))^T$ en $t = t_{n+1}$. Il suffit ensuite de réitérer ces calculs pour chaque instant $t_n \in [t_k; t_{k+1}]$ pour déterminer la valeur de $\boldsymbol{\chi}_{k+1}$. Cela conduit à l'algorithme III.3 suivant :

Algorithme III.3 Estimation par morceau.

Initialisation :

- Déterminer $\boldsymbol{\chi}_0$
- Fixer : $t_k = t_0$ et $\boldsymbol{\chi}_k = \boldsymbol{\chi}_0$

POUR CHAQUE intervalle de commande $[t_k; t_{k+1}]$ FAIRE

Fixer : $t_n = t_k$, $\boldsymbol{\chi}_n = \boldsymbol{\chi}_k$

POUR CHAQUE intervalle d'intégration $[t_n, t_{n+1}]$ FAIRE

Évaluer $q(t_n)$ à l'aide de la relation (III.41)

En déduire $\mathbf{J}(q(t_n))$ et $\mathcal{T}_{C/\mathcal{R}_0}(q(t_n))$

Calculer $\boldsymbol{\chi}(t_{n+1})$ avec les relations établies précédemment

$t_n = t_n + T_N$ /*itération du schéma*/

FIN POUR

$t_k = t_k + T_{ech}$ /*itération de la commande*/

FIN POUR

L'approche proposée ici constitue une première étape vers la prise en compte de l'évolution du torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ dans le processus de reconstruction du vecteur $\boldsymbol{\chi}(t)$. S'appuyant sur les résultats obtenus dans le cas d'un torseur cinématique constant, elle est très simple à mettre en œuvre et permet de rester indépendant de la structure du robot porteur. De plus, le torseur cinématique étant considéré variable sur l'intervalle $[t_k; t_{k+1}]$ (*i.e.* constant par morceaux), elle apparaît plus proche de la réalité. Enfin, l'intégration sur un sous-intervalle $[t_n; t_{n+1}]$ de $[t_k, t_{k+1}]$ peut permettre d'améliorer la qualité de la reconstruction de $\boldsymbol{\chi}(t)$ par un choix judicieux du nombre d'intervalle N .

III.3.2.2 Torseur cinématique variable

Jusqu'ici nous avons supposé que le torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ était constant ou tout au moins constant par morceaux sur l'intervalle $[t_k; t_{k+1}]$. Nous avons ainsi pu résoudre le problème dans le cas général d'un robot porteur à six degrés de liberté et proposer des solutions analytiques indépendantes de ce dernier. Cependant, bien que cette hypothèse soit généralement acceptable (elle est d'autant plus vraie que la dynamique du robot est lente et que la période d'échantillonnage de l'asservissement est faible), elle est rarement pleinement vérifiée. Comme nous l'avons vu au paragraphe précédent, en toute rigueur, seules les commandes \dot{q} envoyées au robot sont bloquées pendant l'intervalle de commande $T_{\text{ech}} = t_{k+1} - t_k$. Rappelons que $\mathcal{T}_{C/\mathcal{R}_0} = \mathbf{J}(q)\dot{q}$ et que la jacobienne \mathbf{J} dépend de q . Comme le robot se déplace sous l'action de la commande entre t_k et t_{k+1} , q varie nécessairement sur cet intervalle et le torseur cinématique n'est donc pas strictement constant. Nous avons donc cherché à nous affranchir de l'hypothèse sur le torseur cinématique de la caméra. Pour cela, il est nécessaire de considérer un système robotique particulier afin d'avoir une expression de la jacobienne \mathbf{J} . Nous avons donc naturellement choisi de nous focaliser sur notre plateforme robotique à trois degrés de liberté, où $\mathcal{T}_{C/\mathcal{R}_0}$ se réduit à $\mathcal{T}_r = (V_{\vec{y}_C}, V_{\vec{z}_C}, \Omega_{\vec{x}_C})^T$, avec $V_{\vec{x}_C} = \Omega_{\vec{y}_C} = \Omega_{\vec{z}_C} = 0$. Les résultats proposés, dans cette partie, seront donc restreints à des robots porteurs de structure mécanique similaire. Dans notre cas, le mouvement de la caméra est défini par le torseur cinématique réduit \mathcal{T}_r donné par l'équation rappelée ci-dessous (I.7) (cf. chapitre I, page 29) :

$$\mathcal{T}_r = \begin{pmatrix} V_{\vec{y}_C}(t) \\ V_{\vec{z}_C}(t) \\ \Omega_{\vec{x}_C}(t) \end{pmatrix} = \begin{pmatrix} -v \sin(\vartheta(t)) + \omega(C_x + D_x \cos(\vartheta(t))) + C_x \varpi \\ v \cos(\vartheta(t)) + \omega(-C_y + D_x \sin(\vartheta(t))) - C_y \varpi \\ -(\omega + \varpi) \end{pmatrix} = \mathbf{J}_r(q)\dot{q} \quad (\text{III.42})$$

Le mouvement de la caméra \mathcal{T}_r dépend donc de l'orientation $\vartheta(t)$ de la platine, et n'est donc pas strictement constant. Pour aller plus loin il est nécessaire d'avoir une expression de $\vartheta(t)$. Pour cela, rappelant que la dynamique de la platine est définie par : $\dot{\vartheta} = \varpi$, et que la commande de la platine $\varpi(t) = \varpi(t_k) = \varpi_k$ est constante sur $[t_k; t_{k+1}]$, nous intégrons cette relation entre t_k et t , il vient :

$$\vartheta(t) = (t - t_k) \varpi_k + \vartheta_k \quad (\text{III.43})$$

où $\vartheta_k = \vartheta(t_k)$ correspond à la valeur angulaire de la platine à l'instant t_k , mesurée par les capteurs proprioceptifs (e.g. encodeurs embarqués). Ainsi, en prenant en compte le fait que nous travaillons sur une plate-forme dotée de trois degrés de liberté, et que le torseur $\mathcal{T}_{C/\mathcal{R}_0}$ est maintenant considéré comme variable pour tout $t \in [t_k; t_{k+1}]$, le système d'équations différentielles à résoudre peut se réécrire comme suit :

$$\dot{X} = \frac{X(t)}{z(t)} V_{\vec{z}_C}(t) + \frac{X(t)Y(t)}{f} \Omega_{\vec{x}_C}(t) \quad (\text{III.44})$$

$$\dot{Y} = -\frac{f}{z(t)} V_{\vec{y}_C}(t) + \frac{Y(t)}{z(t)} V_{\vec{z}_C}(t) + \left(f + \frac{Y(t)^2}{f} \right) \Omega_{\vec{x}_C}(t) \quad (\text{III.45})$$

$$\dot{z} = -V_{\vec{z}_C}(t) - \frac{z(t)Y(t)}{f} \Omega_{\vec{x}_C}(t) \quad (\text{III.46})$$

Remarque III.7: Afin d'alléger les notations ultérieures les commandes v_k , ω_k et ϖ_k de \dot{q} à l'instant t_k seront respectivement notées v , ω et ϖ .

Remarque III.8: Rappelons que nous ne considérons pas ici les éventuels bruits sur la commande \dot{q} , ni sur la jacobienne \mathbf{J}_r du robot (e.g. lié au processus de calibration). Nous supposons donc que ces données sont parfaitement connues.

Comme précédemment, nous nous intéressons en premier lieu au calcul de $z(t)$, avant de déduire $X(t)$ et $Y(t)$.

Détermination de $z(t)$

Dérivons l'équation différentielle (III.46). Nous obtenons :

$$\ddot{z} = -\dot{V}_{\vec{z}_C} - \left(\dot{Y}z(t) + Y(t)\dot{z} \right) \frac{\Omega_{\vec{x}_C}(t)}{f} - \frac{Y(t)z(t)}{f} \dot{\Omega}_{\vec{x}_C}$$

En multipliant (III.46) et (III.45) respectivement par $Y(t)$ et $z(t)$, il vient :

$$\dot{Y}z(t) + Y(t)\dot{z} = -fV_{\vec{y}_C}(t) + fz(t)\Omega_{\vec{x}_C}(t) \quad (\text{III.47})$$

De plus, à l'aide de (III.42) nous déduisons les dérivées $\dot{V}_{\vec{z}_C}$ et $\dot{\Omega}_{\vec{x}_C}$ suivantes :

$$\dot{V}_{\vec{z}_C} = (-v \sin(\vartheta(t)) + D_x \omega \cos(\vartheta(t))) \varpi \quad (\text{III.48})$$

$$\dot{\Omega}_{\vec{x}_C} = 0 \quad (\text{III.49})$$

Nous obtenons alors une équation différentielle du second ordre avec second membre *non constant*, soit :

$$\begin{aligned} \ddot{z} + z(t)(\omega + \varpi)^2 &= C_x \varpi (\omega + \varpi) - (\omega + 2\varpi) V_{\vec{y}_C}(t) \\ \ddot{z} + z(t)A_1^2 &= A_2(t) \end{aligned} \quad (\text{III.50})$$

où $A_1 = (\omega + \varpi)$ et $A_2(t) = C_x \varpi (\omega + \varpi) - (\omega + 2\varpi) V_{\vec{y}_C}(t)$. Pour résoudre cette équation différentielle, il est nécessaire de trouver la solution de l'équation homogène :

$$\ddot{z} + A_1^2 z(t) = 0$$

Plusieurs situations peuvent alors se présenter :

① Si $A_1 \neq 0$, c'est-à-dire $\omega \neq -\varpi$, alors l'équation homogène admet comme solution :

$$z_H(t) = c_1 \sin(A_1(t - t_k)) + c_2 \cos(A_1(t - t_k))$$

où c_1 et c_2 sont des constantes dépendant des conditions initiales du système dynamique. Afin de déterminer la solution particulière, nous utilisons ici la méthode des coefficients indéterminés, et proposons une solution de la forme :

$$z_P(t) = B_1 \cos(\vartheta(t)) + B_2 \sin(\vartheta(t)) + B_3$$

En dérivant $z_P(t)$, il vient :

$$\begin{aligned} \dot{z}_P(t) &= -B_1 \dot{\vartheta} \sin(\vartheta(t)) + B_2 \dot{\vartheta} \cos(\vartheta(t)) \\ \ddot{z}_P(t) &= -B_1 \ddot{\vartheta} \sin(\vartheta(t)) - B_1 \dot{\vartheta}^2 \cos(\vartheta(t)) + B_2 \ddot{\vartheta} \cos(\vartheta(t)) - B_2 \dot{\vartheta}^2 \sin(\vartheta(t)) \end{aligned}$$

En remplaçant $z_P(t)$, $\dot{z}_P(t)$ et $\ddot{z}_P(t)$ dans (III.50), et en rappelant que $\dot{\vartheta} = \varpi$ et par conséquent $\ddot{\vartheta} = 0 \forall t \in [t_k; t_{k+1}]$, il est possible d'identifier les paramètres B_1 , B_2 et B_3 . On montre que si $\omega \neq 0$, alors :

$$\left. \begin{array}{l} B_1 = -D_x \\ B_2 = \frac{v}{\omega} \\ B_3 = -C_x \end{array} \right\} \Rightarrow z_P(t) = -D_x \cos(\vartheta(t)) + \frac{v}{\omega} \sin(\vartheta(t)) - C_x$$

Finalement la solution générale $z(t) = z_H(t) + z_P(t)$ de l'équation différentielle (III.46) s'écrit :

$$z(t) = c_1 \sin(A_1(t - t_k)) + c_2 \cos(A_1(t - t_k)) - D_x \cos(\vartheta(t)) + \frac{v}{\omega} \sin(\vartheta(t)) - C_x \quad (\text{III.51})$$

Les constantes c_1 et c_2 seront déterminées ultérieurement, une fois $X(t)$ et $Y(t)$ établis. Notons que cette solution n'est admissible que si $\omega \neq 0$ (*cf.* cas ②).

- ② Si $\omega = 0$ et $\varpi \neq 0$ la solution (III.51) n'est pas admissible. Cependant, pour ce cas particulier, l'équation différentielle (III.50) devient :

$$\ddot{z} + z(t)\varpi^2 = 2v\varpi \sin(\vartheta(t)) - C_x\varpi^2$$

De la même manière que précédemment, nous pouvons établir la solution homogène :

$$z_H(t) = c_1 \sin(\varpi(t - t_k)) + c_2 \cos(\varpi(t - t_k))$$

Nous proposons alors la solution particulière suivante :

$$z_P(t) = -v(t - t_k) \cos(\vartheta(t)) + \frac{v}{2\varpi} \sin(\vartheta(t)) - C_x$$

Finalement la formulation analytique de la profondeur s'écrit alors :

$$z(t) = c_1 \sin(\varpi(t - t_k)) + c_2 \cos(\varpi(t - t_k)) - v(t - t_k) \cos(\vartheta(t)) + \frac{v}{2\varpi} \sin(\vartheta(t)) - C_x \quad (\text{III.52})$$

Notons que cette solution n'est admissible que si $\varpi \neq 0$, dans le cas contraire on peut utiliser la solution établie au cas ①.

- ③ Si $A_1 = 0$, c'est-à-dire si $\omega = -\varpi$, l'équation différentielle (III.46) se simplifie en :

$$\dot{z} = -V_{\vec{z}_C}(t) = -v \cos(\vartheta(t)) - \omega D_x \sin(\vartheta(t))$$

La solution pour la profondeur $z(t)$ dans le cas particulier, où le mouvement de rotation de la platine compense exactement celui de la base mobile, s'obtient alors en intégrant l'équation ci-dessus entre t_k et t , soit :

$$\int_{t_k}^t \dot{z}(\tau) d\tau = -v \int_{t_k}^t \cos(\vartheta(\tau)) d\tau - \omega D_x \int_{t_k}^t \sin(\vartheta(\tau)) d\tau$$

connaissant l'expression de $\vartheta(t)$ définie par la relation (III.43), il vient :

$$[z(\tau)]_{t_k}^t = -\frac{v}{\varpi} [\sin(\vartheta(\tau))]_{t_k}^t + \frac{\omega}{\varpi} D_x [\cos(\vartheta(\tau))]_{t_k}^t$$

Si $\varpi = 0$, alors $z(t)$ s'écrit :

$$z(t) = -\frac{v}{\varpi} (\sin(\vartheta(t)) - \sin(\vartheta_k)) + \frac{\omega}{\varpi} D_x (\cos(\vartheta(t)) - \cos(\vartheta_k)) + z_k \quad (\text{III.53})$$

Notons que si $\varpi = 0$, la solution (III.53) ci-dessus n'est plus applicable.

- ④ Ainsi pour $A_1 = 0$ avec $\varpi = 0$, soit $\omega = 0$ également (*i.e.* pas de rotation), et donc $\vartheta(t) = \vartheta(t_k) = \vartheta_k$ pour tout $t \in [t_k; t_{k+1}]$, les solutions précédentes (III.52) et (III.53) ne sont pas admissibles. Toutefois, l'équation différentielle (III.46) se simplifie désormais en :

$$\dot{z} = -V_{\vec{z}_C}(t) = -v \cos(\vartheta_k)$$

La profondeur $z(t)$ dans le cas particulier où il n'y a aucune rotation s'obtient alors en intégrant l'équation précédente entre t_k et t :

$$z(t) = -v \cos(\vartheta_k)(t - t_k) + z_k \quad (\text{III.54})$$

Remarque III.9: Notons que dans le cas où la platine est animée d'aucun mouvement (*i.e.* $\varpi = 0$), le torseur cinématique devient véritablement constant et que la méthode proposée dans le paragraphe III.3.1 est pleinement applicable. Dans ce cas précis, nous obtenons d'ailleurs un résultat identique à celui pour les équations (III.34) et (III.54).

Détermination de $X(t)$ et de $Y(t)$

Intéressons-nous à présent au calcul de $X(t)$ en considérant l'équation différentielle (III.44) qui peut se réécrire sous la forme :

$$\frac{\dot{X}}{X(t)} = -\frac{\dot{z}}{z(t)}$$

En intégrant cette relation entre t_k et t , il vient :

$$X(t) = \frac{z_k X_k}{z(t)} \quad (\text{III.55})$$

L'expression de $Y(t)$ se déduit directement de (III.46), soit :

$$Y(t) = -f \frac{\dot{z} + V_{\vec{z}_C}(t)}{z(t)\Omega_{\vec{x}_C}(t)} = f \frac{\dot{z} + v \cos(\vartheta(t)) + D_x \omega \sin(\vartheta(t)) - C_y(\omega + \varpi)}{z(t)(\omega + \varpi)} \quad (\text{III.56})$$

Différents cas sont alors possibles :

- ① Si $A_1 \neq 0$, c'est-à-dire $\omega \neq -\varpi$, et avec $\omega \neq 0$, $z(t)$ est donné par la relation (III.51). En dérivant cette relation et en remplaçant \dot{z} par son expression dans (III.56), $Y(t)$ s'écrit :

$$Y(t) = -\frac{f}{z(t)} \left(c_1 \cos(A_1(t - t_k)) - c_2 \sin(A_1(t - t_k)) - D_x \sin(\vartheta(t)) - \frac{v}{\omega} \cos(\vartheta(t)) + C_y \right) \quad (\text{III.57})$$

Connaissant les conditions initiales ϑ_k , Y_k et z_k à l'instant t_k , nous pouvons établir les constantes c_1 et c_2 :

$$\begin{aligned} c_1 &= -\frac{Y_k z_k}{f} + D_x \sin(\vartheta_k) + \frac{v}{\omega} \cos(\vartheta_k) - C_y \\ c_2 &= z_k + D_x \cos(\vartheta_k) - \frac{v}{\omega} \sin(\vartheta_k) + C_x \end{aligned}$$

Néanmoins, cette solution et ces conditions initiales ne sont applicable uniquement tant que $\omega \neq 0$.

- ② Si $\omega = 0$ et $\varpi \neq 0$, la solution $Y(t)$ ci-dessus n'est plus admissible. Sur la base de $z(t)$ défini par la relation (III.52) et de sa dérivée que nous substituons dans (III.56), nous pouvons déduire l'expression de $Y(t)$ correspondante :

$$Y(t) = -\frac{f}{z(t)} \left(c_1 \cos(A_1(t - t_k)) - c_2 \sin(A_1(t - t_k)) + v(t - t_k) \sin(\vartheta(t)) - \frac{v}{2\varpi} \cos(\vartheta(t)) + C_y \right) \quad (\text{III.58})$$

Connaissant les conditions initiales à l'instant t_k , nous pouvons déterminer les constantes c_1 et c_2 :

$$\begin{aligned} c_1 &= -\frac{Y_k z_k}{f} + \frac{v}{2\varpi} \cos(\vartheta_k) - C_y \\ c_2 &= z_k - \frac{v}{2\varpi} \sin(\vartheta_k) + C_x \end{aligned}$$

Cependant, cette solution n'est admissible que si $\varpi \neq 0$. Dans le cas contraire, comme précédemment lors de la résolution de la profondeur, on peut utiliser la solution établie au cas ①.

- ③ Si $A_1 = 0$, c'est-à-dire si $\omega = -\varpi$, alors la relation (III.56) n'est plus applicable. Dans ce cas, la relation (III.47) se simplifie comme suit :

$$\dot{Y}z(t) + Y(t)\dot{z} = -fV_{\vec{y}_C}(t) = -fv \sin(\vartheta(t)) \quad (\text{III.59})$$

En intégrant cette équation entre t_k et t , et connaissant l'expression de $\vartheta(t)$ définie par la relation (III.43), il vient :

$$\begin{aligned} \int_{t_k}^t \left(\dot{Y}z(\tau) + Y(\tau)\dot{z} \right) d\tau &= -fv \int_{t_k}^t \sin(\vartheta(\tau)) d\tau \\ [z(\tau)Y(\tau)]_{t_k}^t &= f \frac{v}{\varpi} [\cos(\vartheta(\tau))]_{t_k}^t \end{aligned}$$

Ce qui nous conduit à la solution suivante :

$$Y(t) = \frac{fv(\cos(\vartheta(t)) - \cos(\vartheta_k)) + \varpi z_k Y_k}{\varpi z(t)} \quad (\text{III.60})$$

où $z(t)$ est donné par (III.52). Toutefois, cette solution n'est applicable que si $\varpi \neq 0$.

- ④ Ainsi pour $A_1 = 0$ avec $\omega = \varpi = 0$ (*i.e.* pas de rotation) et donc $\vartheta(t) = \vartheta(t_k) = \vartheta_k$ pour tout $t \in [t_k; t_{k+1}]$, en intégrant la relation (III.59), il vient :

$$\int_{t_k}^t \left(\dot{Y}z(\tau) + Y(\tau)\dot{z} \right) d\tau = -fv \sin(\vartheta_k) \int_{t_k}^t 1 d\tau$$

On obtient alors la solution $Y(t)$ suivante :

$$Y(t) = \frac{-fv \sin(\vartheta_k)(t - t_k) + Y_k z_k}{z(t)} \quad (\text{III.61})$$

Notons que $\vartheta(t) = \vartheta(t_k) = \vartheta_k, \forall t \in [t_k; t_{k+1}]$, le torseur cinématique est, encore une fois, véritablement constant sur cet intervalle (*cf.* remarque III.9). La méthode proposée dans le paragraphe III.3.1 est alors pleinement applicable.

Bilan

L'approche proposée dans cette partie prend pleinement en compte l'évolution du torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ dans la reconstruction des données $\chi(t) = (X(t), Y(t), z(t))$ pour tout t de l'intervalle de commande $[t_k; t_{k+1}]$. Bien que limitée au système robotique à trois degrés de liberté considéré au cours de notre étude, cette approche présente l'avantage d'être plus proche de la "réalité" que les techniques proposées précédemment, et donc de conduire à de meilleurs résultats en termes de qualité de l'estimation (*cf.* chapitre IV).

Cependant, cette méthode suppose que les données initiales χ_k , ainsi que les commandes \dot{q} et les paramètres de la jacobienne du robot \mathbf{J}_r sont parfaitement connus (*cf.* remarque III.8). Ainsi, si ces derniers sont bruités, les données estimées seront moins précises. Une solution pour limiter les effets du bruit serait de les modéliser et d'exploiter ce modèle, par exemple, à travers un filtre de Kalman [KALMAN, 1960] [MAYBECK, 1979].

III.4 Cas général : résolution numérique

Le système d'équations (III.19) dépendant de la nature des primitives visuelles \mathbf{s} considérées, des matrices d'interaction associées et de la structure mécanique du robot porteur, il s'avère difficile de le résoudre analytiquement dans le cas général. Il est donc indispensable de disposer de techniques capables de proposer des *solutions approchées* à ce système lorsqu'il ne peut être résolu analytiquement (ou difficilement).

Notre intention, à présent, est de proposer un cadre générale permettant de mettre en œuvre une méthode de caractérisation des données perceptuelles pour le plus grand nombre d'architecture robotique et de primitives visuelles. Nous nous proposons, pour cela, d'utiliser les techniques d'intégration numérique qui apparaissent bien adaptées dans ce contexte.

Préliminaires

Nous cherchons donc ici à résoudre *numériquement* le système d'équations différentielles (III.19) connaissant χ_k ainsi que les mouvements $\mathcal{T}_{C/\mathcal{R}_0}$ de la caméra à l'instant t_k . Pour cela, de la même manière qu'au paragraphe III.3.2.1, nous nous proposons d'échantillonner l'intervalle $[t_k; t_{k+1}]$ en N sous-intervalles (*cf.* figure III.4). Nous définissons donc le pas d'intégration T_N comme précédemment par : $T_N = \frac{T_{\text{ech}}}{N}$. Le choix de T_N est particulièrement important pour la qualité de l'estimation numérique obtenue puisque cette dernière est, généralement, d'autant meilleure que le pas d'intégration T_N est faible.

L'utilisation de techniques numériques pour la résolution de (III.19) requiert la détermination préalable d'une expression analytique de ce système dynamique, ce qui suppose de caractériser le vecteur χ et les différentes matrices d'interaction associées (*cf.* paragraphe III.2.4). De plus, les schémas d'intégrations employés ne sont pas mis en œuvre de la même manière, selon que le torseur cinématique $\mathcal{T}_{C/\mathcal{R}_0}$ est considéré constant ou variable sur l'intervalle $[t_k; t_{k+1}]$. Il est donc nécessaire de distinguer les deux cas suivants :

- ① $\mathcal{T}_{C/\mathcal{R}_0}$ est supposé constant sur l'intervalle $[t_k; t_{k+1}]$: dans ce cas une seule évaluation de $\mathcal{T}_{C/\mathcal{R}_0}$ à l'instant $t_n = t_k$ intervient dans le calcul de $\varphi(\chi, t)$, $\forall t \in [t_k; t_{k+1}]$.

- ② $\mathcal{T}_{C/\mathcal{R}_0}$ est considéré variable ou constant sur le sous-intervalle $[t_n; t_{n+1}] \subset [t_k; t_{k+1}]$, où $t_n = t_k + (n - k)T_N$ ($n = \{k, k + 1, \dots, k + N\}$) : la détermination de $\varphi(\boldsymbol{\chi}, t)$ pour tout $t \in [t_n; t_{n+1}]$ passe alors par l'évaluation du torseur $\mathcal{T}_{C/\mathcal{R}_0}(t_n)$, et donc de la jacobienne du robot $\mathbf{J}(q(t_n))$ à chaque instant t_n .

Résolution numérique

Nous supposons maintenant qu'une expression du système dynamique (III.19) est disponible, et nous nous intéressons à sa résolution. Ce système étant constitué d'équations différentielles ordinaires avec conditions initiales $(\boldsymbol{\chi}_k, t_k)$, il existe dans la littérature de nombreuses méthodes permettant de le résoudre numériquement [SHAMPINE et GORDON, 1975], [PRESS *et al.*, 1992], [DEMAILLY, 1996]... Dans le cadre de nos travaux, nous nous sommes plus particulièrement focalisés sur les schémas d'Euler, de Runge–Kutta, d'Adams et sur la méthode de Gear connue également sous le nom de *Backward Differentiation Formulas* (BDF). Nous rappelons rapidement ces différentes techniques ci-après⁷.

Schéma d'Euler Il s'agit de la méthode d'intégration la plus simple, et la moins précise car elle consiste à approximer la dérivée $\dot{\boldsymbol{\chi}} = \varphi(\boldsymbol{\chi}, t)$ à l'instant t_n par le schéma d'ordre 1 suivant :

$$\dot{\boldsymbol{\chi}}(\boldsymbol{\chi}_n, t_n) = \frac{\boldsymbol{\chi}_{n+1} - \boldsymbol{\chi}_n}{T_N}$$

ce qui conduit au résultat bien connu ci-après :

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + T_N \varphi(\boldsymbol{\chi}_n, t_n) \quad (\text{III.62})$$

La mise en œuvre du schéma d'Euler nécessite des périodes d'intégration T_N faibles afin de réduire les imprécisions.

Schémas de Runge–Kutta Les méthodes de Runge–Kutta (B.4) forment une famille importante de techniques permettant de résoudre numériquement des équations différentielles. La forme la plus classique est le schéma de Runge–Kutta d'ordre 4 (dénotée RK4), qui conduit à l'expression de $\widehat{\boldsymbol{\chi}}_{n+1}$ suivante :

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad (\text{III.63})$$

avec :

$$\begin{cases} K_1 &= T_N \varphi(\boldsymbol{\chi}_n, t_n) \\ K_2 &= T_N \varphi(\boldsymbol{\chi}_n + \frac{K_1}{2}, t_n + \frac{T_N}{2}) \\ K_3 &= T_N \varphi(\boldsymbol{\chi}_n + \frac{K_2}{2}, t_n + \frac{T_N}{2}) \\ K_4 &= T_N \varphi(\boldsymbol{\chi}_n + K_3, t_n + T_N) \end{cases}$$

Schémas d'Adams La méthode d'Adams consiste à utiliser l'historique des valeurs de $\boldsymbol{\chi}(t)$ pour approximer $\varphi(\boldsymbol{\chi}, t)$ par un polynôme $\mathcal{P}(t)$ de degré $\mathbf{D} - 1$. Elle fonctionne le plus souvent sous la forme d'une paire prédicteur/correcteur, connue sous le nom de schéma d'Adams–Bashforth–Moulton (ABM). Celui-ci

7. Une présentation détaillée de ces méthodes d'intégration numérique est proposée en annexe B.

consiste à *prédire* la valeur $\tilde{\chi}_{n+1}$ de χ_{n+1} au moyen d'un schéma d'Adams–Bashforth (B.7), puis de *corriger* cette valeur grâce à un schéma d'Adams–Moulton (B.10) pour déterminer $\hat{\chi}_{n+1}$. Le schéma ABM d'ordre 4 est donné par :

$$\tilde{\chi}_{n+1} = \chi_n + \frac{T_N}{24}(55\varphi_n - 59\varphi_{n-1} + 37\varphi_{n-2} - 9\varphi_{n-3}) \quad (\text{III.64})$$

$$\hat{\chi}_{n+1} = \chi_n + \frac{T_N}{24}(9\varphi(\tilde{\chi}_{n+1}, t_{n+1}) + 19\varphi_n - 5\varphi_{n-1} + \varphi_{n-2}) \quad (\text{III.65})$$

où $\varphi_i = \varphi(\chi_i, t_i)$ désigne la fonction $\varphi(\chi_i, t_i)$ à l'instant t_i .

Méthode de Gear La méthode de Gear [GEAR, 1971], aussi connue sous le nom de *Backward Differentiation Formulas* (BDF) (B.14), consiste à utiliser le polynôme qui passe par les χ_i pour approximer les solutions du système d'équations différentielles considéré. Le schéma BDF d'ordre 4 que nous utiliserons dans nos travaux, permet d'écrire $\hat{\chi}_{n+1}$ comme suit :

$$\hat{\chi}_{n+1} = \frac{300}{137}\chi_n - \frac{300}{137}\chi_{n-1} + \frac{200}{137}\chi_{n-2} - \frac{75}{137}\chi_{n-3} + \frac{12}{137}\chi_{n-4} + \frac{60}{137}T_N\varphi(\tilde{\chi}_{n+1}, t_{n+1}) \quad (\text{III.66})$$

Notons que la méthode BDF est un schéma implicite qui nécessite une prédiction $\tilde{\chi}_{n+1}$ de χ_{n+1} . Celle-ci peut être fournie par l'un des schémas précédents. De plus le BDF est particulièrement bien adapté aux problèmes dits “raides” ou *stiff*. Nous verrons que cette propriété peut être intéressante dans le cas où la dynamique du système considéré varie rapidement (*cf.* chapitre IV).

Remarque III.10: *Les méthodes d'Adams et de Gear sont des schémas à pas multiple (cf. annexe B) où l'état du système $\chi(t)$ à t_{n+1} est construit à partir de l'historique des \mathbf{D} états antérieurs obtenus aux instants $t_n, t_{n-1}, \dots, t_{n-\mathbf{D}}$. Dans ce cas, il faut disposer d'au moins \mathbf{D} valeurs de l'état χ avant de pouvoir estimer $\hat{\chi}_{n+1}$. Pour pallier cet inconvénient, on peut soit considérer des ordres inférieurs, jusqu'à disposer de suffisamment de valeurs ; soit initialiser ces schémas avec des méthodes à pas simple, telles que les schémas d'Euler ou de Runge–Kutta.*

Ainsi ces intégrateurs numériques permettent d'obtenir une estimation χ_{k+1} , en itérant le schéma N fois sur une période d'échantillonnage du système. Ainsi, une fois le schéma numérique sélectionné, nous pouvons obtenir une estimation $\hat{\chi}$ de χ en suivant l'algorithme III.4.

Finalement, l'emploi d'intégrateurs numériques pour la résolution du système d'équations différentielles (III.19) est envisageable, dès lors qu'une formulation de $\varphi(\chi, t)$ et qu'un historique des valeurs successives de χ sont disponibles. Nous pouvons alors espérer résoudre notre système dynamique pour tout type de primitives visuelles, et ceci quelle que soit la structure du système robotique portant la caméra.

Algorithme III.4 Intégration numérique.

```

Initialisation :
- Déterminer  $\chi_0$ 
- Fixer :  $t_k = t_0$  et  $\chi_k = \chi_0$ 
POUR CHAQUE intervalle de commande  $[t_k; t_{k+1}]$  FAIRE
  SI (schéma à pas multiple) ET (pas fini d'initialiser) ALORS
    Initialisation des états antérieurs
  FIN SI
  SI torseur constant /*i.e. cas ①*/ ALORS
    Évaluer le mouvement de la caméra  $\mathcal{T}_{C/\mathcal{R}_0}(t_k)$  à l'instant  $t_k$ 
  FIN SI
  Fixer :  $t_n = t_k, \chi_n = \chi_k$ 
  POUR CHAQUE intervalle d'intégration  $[t_n, t_{n+1}]$  FAIRE
    SI torseur variable /*i.e. cas ②*/ ALORS
      Évaluer le mouvement de la caméra  $\mathcal{T}_{C/\mathcal{R}_0}(t_n)$  à l'instant  $t_n$ 
    FIN SI
    Déterminer la matrice d'interaction  $\mathcal{L}(\chi_n)$ 
    Calculer  $\hat{\chi}_{n+1}$  selon le schéma numérique choisi
     $t_n = t_n + T_N$  /*itération du schéma*/
  FIN POUR
   $t_k = t_k + T_{ech}$  /*itération de la commande*/
FIN POUR

```

III.5 Conclusion

Dans ce chapitre nous avons proposé un ensemble de méthodes permettant de reconstruire le motif visuel d'intérêt \mathbf{s} lorsqu'il ne peut plus être délivré par la caméra. La majorité des travaux s'intéressant à cette problématique s'appuient généralement sur des informations provenant de l'image (*e.g.* en exploitant sa dynamique). Toutefois, ici, nous nous sommes placés dans le cas où l'image entière devient temporairement indisponible. Nous nous sommes alors appuyés sur le lien entre la variation des indices visuels dans l'image et le mouvement de la caméra pour reconstruire les données manquantes. Nous avons, tout d'abord, modélisé le problème pour différents types de motifs visuels \mathbf{s} , à savoir :

- les points, entités élémentaires qui permettent de traiter explicitement la profondeur de l'objet d'intérêt ;
- les primitives géométriques simples, qui enrichissent les tâches de navigation référencées vision ;
- les moments de l'image qui permettent de considérer la forme de l'objet d'intérêt dans son ensemble.

Nous avons ensuite montré que le problème pouvait se résumer à la résolution du système d'équations différentielles (III.19). Nous avons alors développé différentes techniques de résolution analytique et numérique permettant de répondre à ce problème. Les premières offrent des solutions précises, mais il est nécessaire de disposer d'une formulation analytique de tous les éléments du système dynamique (III.19) pour pouvoir espérer

le résoudre. Elles sont valables pour tout type de plate-forme robotique, tant que le torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ est considéré sous sa forme la plus générique et constant sur une période d'échantillonnage T_{ech} . Pour tenir compte d'une certaine variabilité du mouvement de la caméra aux cours de l'intervalle de commande, il est alors nécessaire de particulariser selon l'architecture du robot porteur, et de connaître le lien entre le vecteur de commande \dot{q} et le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$. Les méthodes de résolution numérique conduisent à des solutions approchées mais présentent l'avantage d'être exploitables dans le cas le plus général : pour toute primitive visuelle et tout système mécanique. Les différentes approches proposées sont simples en termes de mise en œuvre, et donc facilement intégrables au sein de la boucle de commande. Cependant, comme nous le verrons par la suite, elles nécessitent, de bien connaître les différentes périodes d'échantillonnage (*i.e.* les périodes T_{ech} et T_N), et de disposer des informations nécessaires pour l'initialisation de nos algorithmes (historique des mouvements de la caméra ainsi que des mesures des indices visuels). Toutefois, nous nous sommes placés dans le cas où les données ne sont pas perturbées (*cf.* remarque III.8). Une solution pour limiter les effets des perturbations serait d'utiliser un filtre de Kalman [KALMAN, 1960] [MAYBECK, 1979].

Nous nous proposons maintenant d'illustrer les différentes approches développées dans le contexte applicatif de la navigation référencée vision. Nous présentons ainsi dans le chapitre IV quelques exemples permettant de valider ces techniques et de comparer leur efficacité.

Chapitre IV

Navigation référencée vision avec gestion de la perte du signal visuel

Nous avons présenté au chapitre III un ensemble d’approches permettant de reconstruire le motif visuel $\mathbf{s}(q, t)$ lorsqu’il devient indisponible. Comme nous l’avons préalablement établi, la mise en œuvre de nos techniques dépend à la fois du type d’informations visuelles $\mathbf{s}(q, t)$ considérées et de la nature du système robotique portant la caméra. Bien que nous ayons proposé des solutions pour le cas général où le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ est défini sur la base d’un système à six degrés de liberté, nous nous restreindrons, dans ce chapitre, à des applications concernant notre plate-forme robotique dotée seulement de trois degrés de liberté (*cf.* chapitre I). Nous nous proposons à présent de valider nos différentes méthodes d’estimation dans le cadre de diverses tâches de navigation référencées vision.

Cependant, il est un aspect important, sur lequel nous n’avons pas encore insisté : les méthodes proposées supposant le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ connu, elles ne permettent de définir des schémas de commande en boucle fermée (sur les mesures sensorielles) que lorsque $\mathcal{T}_{C/\mathcal{R}_0}$ est établi indépendamment des indices visuels. C’est, par exemple, le cas lors de la gestion des occultations pendant l’évitement d’obstacles (*cf.* section IV.2). Cependant, pour de “véritables” tâches de navigation référencée vision, où $\mathcal{T}_{C/\mathcal{R}_0}$ est déduit directement à partir des données perceptuelles estimées $\hat{\mathbf{s}}$ (et non plus celles mesurées), le schéma de commande devient alors une boucle ouverte (*cf.* figure IV.1). Dans ce cas, notre approche reste applicable pour des pertes du signal visuel de *courte durée*, et dans la mesure où le mouvement du robot (et celui de la caméra) n’est que peu perturbé. Elle apparaît alors comme une solution de “secours”, permettant de continuer à exécuter la tâche référencée vision “dans un mode dégradé” de manière à atteindre au mieux les objectifs fixés, malgré l’indisponibilité du motif visuel d’intérêt.

Nous nous intéressons donc désormais à ces deux cas, et présentons les résultats obtenus pour chacun d’entre-eux.

IV.1 Réalisation d'une tâche référencée vision

Notre objectif est ici de réaliser une tâche de positionnement par rapport à un amer visuel dans un environnement libre de tout obstacle en dépit de l'indisponibilité éventuelle du signal perceptuel $\mathbf{s}(q, t)$. De ce fait le robot est piloté par la commande \dot{q}_{AV} définie par la relation (I.18) (*cf.* chapitre I), que nous rappelons ci-après :

$$\dot{q}_{AV} = -\mathbf{J}_r^{-1} \lambda \mathcal{C}(\mathbf{s} - \mathbf{s}^*)$$

Le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ est alors commandé de manière différente selon que le signal sensoriel \mathbf{s} est disponible ou non. Deux cas peuvent se présenter :

- ① $\mathbf{s}(q, t)$ est acquis depuis la caméra : la commande \dot{q}_{AV} reste en boucle fermée sur les mesures du signal perceptuel.
- ② $\mathbf{s}(q, t)$ est indisponible : la commande \dot{q}_{AV} est évaluée à partir des données perceptuelles estimées $\hat{\mathbf{s}}$, et le schéma de commande devient alors une boucle ouverte (*cf.* figure IV.1).

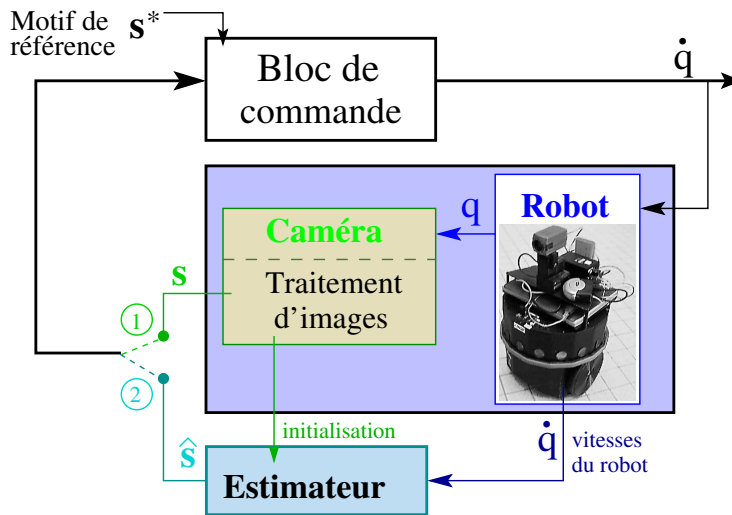


FIG. IV.1 – Commande référencée vision et estimation.

Nous nous proposons ici d'illustrer la pertinence de nos techniques de reconstruction de $\mathbf{s}(q, t)$ dans deux contextes. Dans le premier, nous cherchons à gérer la perte du motif visuel occasionnée par une panne temporaire de la caméra ou du processus de traitement d'images. Dans le second, nous nous intéressons à la prédiction du signal visuel entre deux acquisitions d'image, l'intérêt étant alors de pouvoir rafraîchir la loi de commande à une période d'échantillonnage plus élevée que celle du traitement d'images, et ainsi d'améliorer les performances de l'asservissement.

IV.1.1 Gestion de la perte du motif visuel suite à un défaut de la caméra

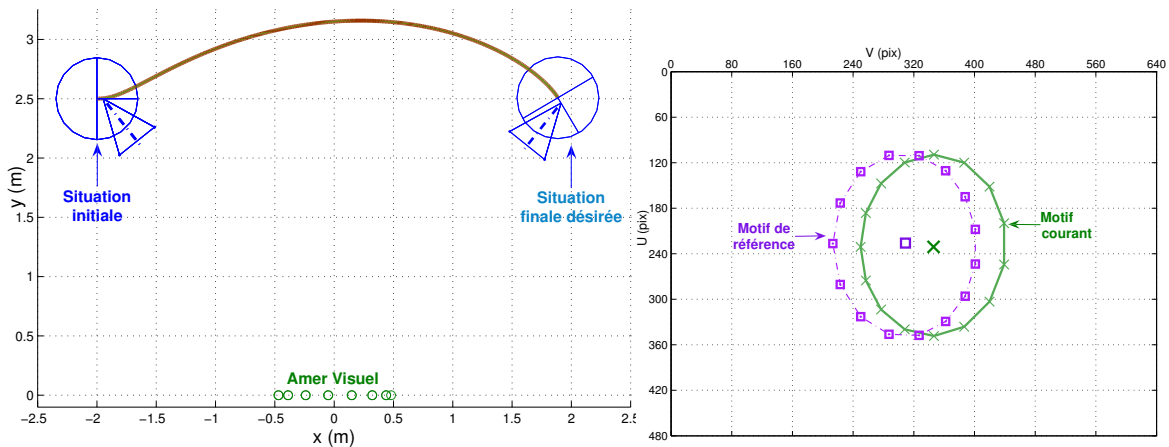
Notre objectif est ici de positionner la caméra embarquée relativement à un amer au moyen de la commande \dot{q}_{AV} (I.18), en dépit de la perte **totale** du motif visuel due à un défaut temporaire de la caméra ou du processus de traitement d'images. Nous illustrons les résultats obtenus par des exemples de simulation et d'expérimentation.

IV.1.1.1 Résultats de simulation

Afin d'évaluer l'efficacité de nos techniques d'estimation, nous avons effectué de nombreuses simulations à l'aide de **Matlab**, en considérant différents motifs visuels \mathbf{s} . Pour chaque cas étudié, nous nous sommes placés dans des conditions identiques de tests. Nous avons ainsi réalisé la même tâche de navigation en partant de la même configuration initiale pour atteindre le même motif visuel de référence \mathbf{s}^* . De manière similaire, nous avons simulé la panne de la caméra à l'issue de 10 itérations, et ce jusqu'à la fin de la mission. Les algorithmes à pas multiple disposent ainsi de suffisamment de mesures pour être initialisés convenablement¹.

Par ailleurs, nous avons cherché à étudier le comportement de ces techniques en fonction de la valeur du pas d'intégration T_N pour deux périodes d'échantillonnage T_{ech} différentes. La première ($T_{ech} = 50$ ms) correspond approximativement à celle de notre plate-forme robotique, la seconde ($T_{ech} = 100$ ms) permettant de valider nos estimateurs dans un contexte légèrement dégradé.

Nous nous proposons de synthétiser les résultats obtenus.



IV.2.a – Trajectoire du robot.

IV.2.b – Amer visuel d'intérêt.

FIG. IV.2 – Navigation référencée vision avec gestion de la perte du motif visuel.

Nous souhaitons ici réaliser la tâche de navigation proposée sur la figure IV.2.a. Elle consiste à positionner la caméra embarquée face à un amer constitué d'une ellipse (cf. figure IV.2.b). Cet amer peut donc être représenté à l'aide des trois primitives visuelles : des points appartenant à l'ellipse, l'ellipse elle-même, et enfin deux moments de l'image (l'aire et le centre de gravité). Cet amer nous permet donc d'illustrer les trois grandes manières de modéliser le problème de l'estimation des indices visuels évoquées au chapitre III et de valider les algorithmes de reconstruction associés. Nous présentons ci-après les résultats obtenus pour chacune de ces primitives visuelles.

Remarque IV.1: *Le simulateur réalise, d'une manière générale, les différents algorithmes proposés (cf. algorithmes III.1, III.2 et III.4). Notons, qu'afin de disposer de mesure de références suffisamment précise, le simulateur évolue à une période T_{Sim} nettement inférieure à celle de la commande T_{ech} (soit $T_{Sim} = 5$ ms au cours de nos simulations).*

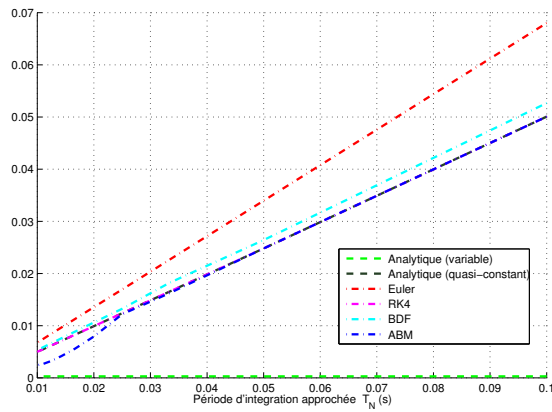
1. Rappelons que nous considérons les schémas ABM (III.65) et BDF (III.66) d'ordre 4, qui nécessitent au moins quatre valeurs antérieures pour être initialisés (cf. remarque III.10 et annexe B).

Les points

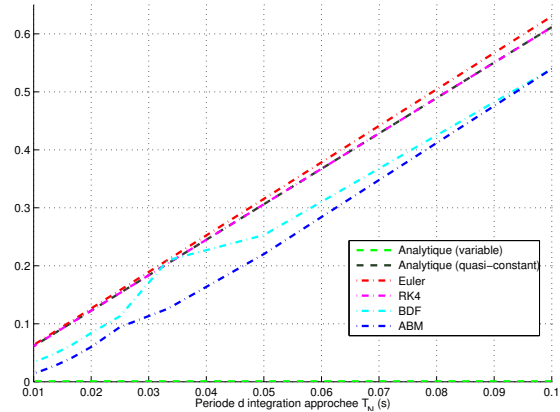
Au chapitre précédent nous avons mis en exergue l'intérêt de l'utilisation des points, d'une part pour sa facilité d'extraction, et d'autre part dans le cadre de la reconstruction d'autres informations perceptuelles plus évoluées. Nous nous intéressons donc, en premier lieu, à la reconstruction d'un motif visuel constitué d'un ensemble de $n = 10$ points (*cf.* figure IV.2.b). Nous définissons donc \mathbf{s} par :

$$\mathbf{s} = (X_1, Y_1, \dots, X_n, Y_n)^T \quad (\text{IV.1})$$

Comme nous l'avons précédemment montré, l'estimation du motif (IV.1) passe par la résolution analytique ou numérique du système (III.6), le vecteur $\boldsymbol{\chi}$ étant alors défini par $\boldsymbol{\chi} = (X_1, Y_1, z_1, \dots, X_n, Y_n, z_n)^T$. Les figures IV.3 et IV.4 présentent les résultats obtenus pour toutes les techniques de reconstruction développées (analytiques et numériques) lorsque la période d'échantillonnage de l'asservissement est fixée d'abord à 100 ms puis à 50 ms. Plus précisément, ces figures décrivent l'évolution de la valeur maximale² de $\|\mathbf{s} - \hat{\mathbf{s}}\|$, et de $\|\mathbf{z} - \hat{\mathbf{z}}\|$, en fonction du pas d'intégration T_N utilisé.

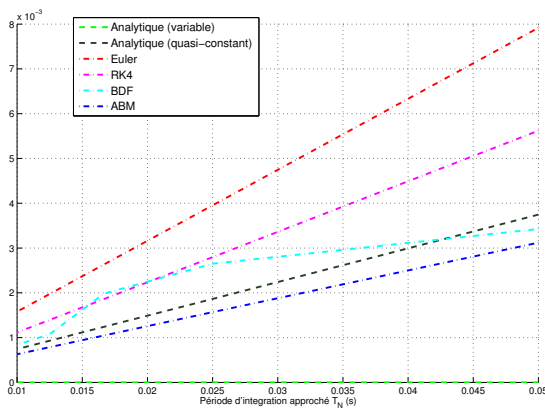


IV.3.a – Erreur maximale $\|\mathbf{s} - \hat{\mathbf{s}}\|$ (pixel) sur le motif visuel en fonction de T_N

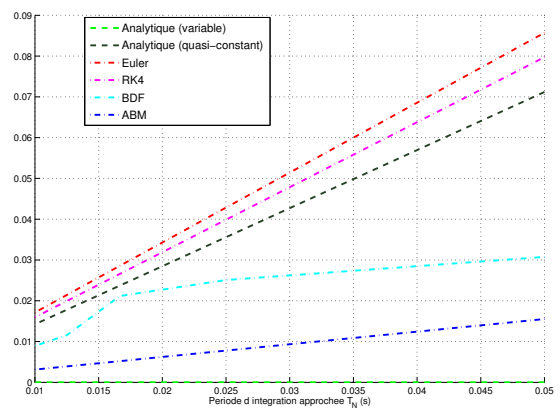


IV.3.b – Erreur maximale $\|\mathbf{z} - \hat{\mathbf{z}}\|$ (mm) sur la profondeur en fonction de T_N .

FIG. IV.3 – Erreurs pour le cas des points : $\mathbf{s} = (X_1, Y_1, \dots, X_n, Y_n)^T$, avec $T_{\text{ech}} = 100$ ms.



IV.4.a – Erreur maximale $\|\mathbf{s} - \hat{\mathbf{s}}\|$ (pixel) sur le motif visuel en fonction de T_N .



IV.4.b – Erreur maximale $\|\mathbf{z} - \hat{\mathbf{z}}\|$ (mm) sur la profondeur en fonction de T_N .

FIG. IV.4 – Erreurs pour le cas des points : $\mathbf{s} = (X_1, Y_1, \dots, X_n, Y_n)^T$, avec $T_{\text{ech}} = 50$ ms.

2. $\|\mathbf{x}\|$ correspond à la norme euclidienne du vecteur \mathbf{x} .

Comparons, tout d'abord, l'efficacité des différentes techniques développées. La méthode apparaissant comme la plus précise est celle qui repose sur la résolution analytique du système dynamique (III.6) avec le torseur $\mathcal{T}_{C/\mathcal{R}_0}$ variable (*cf.* paragraphe III.3.2.2) puisqu'elle conduit à des erreurs quasi-nulles³ tant sur les indices visuels \mathbf{s} que sur la profondeur z . Ce résultat semble tout à fait cohérent, cette méthode apparaissant comme la plus proche de la "réalité" de par l'intégration de l'évolution de $\mathcal{T}_{C/\mathcal{R}_0}$ sur $[t_k; t_{k+1}]$ dans le processus d'estimation. Toutefois, elle reste dédiée au système robotique de notre étude. En ce qui concerne les techniques de résolution approchées, l'algorithme d'Adams ABM apparaît ici comme celui qui offre les meilleurs résultats, tandis que les schémas à pas simple de Runge-Kutta RK4 et d'Euler sont les moins efficaces. La technique de résolution analytique avec torseur constant par morceaux⁴ (*cf.* paragraphe III.3.2.1) conduit à des résultats proches de ceux fournis par la méthode ABM. L'algorithme BDF, quant à lui, bien que relativement précis, ne montre pas ici toute l'étendue de ses capacités car il est mieux adapté aux situations où la commande varie de manière importante (*cf.* section IV.2).

Enfin, d'une manière générale, les erreurs restent faibles (inférieures au pixel pour le motif visuel et au millimètre pour la profondeur). De plus, pour les résolutions numériques, elles diminuent avec le pas d'intégration T_N . Les résultats obtenus démontrent donc la validité des approches proposées en l'absence de perturbations imprévues sur la caméra. En effet, rappelons que le robot est ici commandé en boucle ouverte sur les estimations $\hat{\mathbf{s}}$, tant que les mesures perceptuelles sont indisponibles.

L'ellipse

Nous considérons maintenant toujours le même amer représenté sur la figure IV.2.b, mais nous choisissons cette fois de le caractériser par une primitive géométrique plus complexe, l'ellipse. Celle-ci peut être modélisée par la quadrique suivante [CHAUMETTE, 1990] :

$$X^2 + E_1 Y^2 + 2E_2 XY + 2E_3 X + 2E_4 Y + E_5 = 0 \quad \forall (X, Y) \in \mathcal{C} \quad (\text{IV.2})$$

où les coefficients E_i désignent les paramètres de l'ellipse. Sur cette base, nous avons défini le vecteur des informations sensorielles comme suit :

$$\mathbf{s} = (E_1, E_2, E_3, E_4, E_5, X_1, Y_1, \dots, X_l, Y_l)^T \quad (\text{IV.3})$$

où les couples (X_j, Y_j) sont les coordonnées de $l = 25$ points P_j appartenant à l'ellipse considérée. À partir des relations (III.2) et (III.9), nous pouvons caractériser la variation $\dot{\mathbf{s}}$ qui peut s'écrire comme suit :

$$\dot{\mathbf{s}} = \begin{pmatrix} \mathcal{L}(E_1, A_{pq}) \\ \vdots \\ \mathcal{L}(E_5, A_{pq}) \\ \mathcal{L}(P_1, z_1) \\ \vdots \\ \mathcal{L}(P_l, z_l) \end{pmatrix} \mathcal{T}_{C/\mathcal{R}_0} = \mathcal{L}(E_1, \dots, E_5, A_{pq}, P_1, z_1, \dots, P_l, z_l) \mathcal{T}_{C/\mathcal{R}_0} \quad (\text{IV.4})$$

3. L'erreur constatée ici est de l'ordre de 10^{-10} et est donc liée à la précision des calculs numériques.

4. Le cas où le torseur cinématique de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ est supposé constant sur tout l'intervalle de commande $[t_k; t_{k+1}]$, correspond à $T_N = T_{\text{ech}}$.

où la matrice d'interaction $\mathcal{L}(P_j, z_j)$ d'un point P_j est donnée par la relation (I.11) pour notre plate-forme robotique, $\mathcal{L}(E_i, A_{pq})$ correspondant aux matrices d'interaction des paramètres E_i de l'ellipse⁵.

La reconstruction de \mathbf{s} requiert donc la détermination des coefficients E_i et des couples (X_j, Y_j) . Pour cela, dans la mesure où l'ellipse est une primitive géométrique plane, il est nécessaire de caractériser les composantes A_{pq} de l'équation du plan qui lui est associée. Nous nous trouvons donc dans le cas ② présenté à la section III.2.4. Le processus d'estimation s'effectue alors en trois étapes :

1. Reconstruire les points $P_j(X_j, Y_j)$,
2. Identifier les paramètres A_{pq} du plan,
3. Estimer les paramètres E_i de l'ellipse.

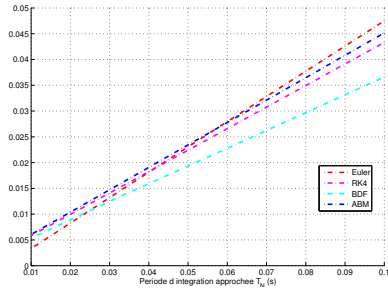
Nous nous proposons de suivre l'algorithme IV.1 ci-après :

Algorithme IV.1 Estimation des ellipses.

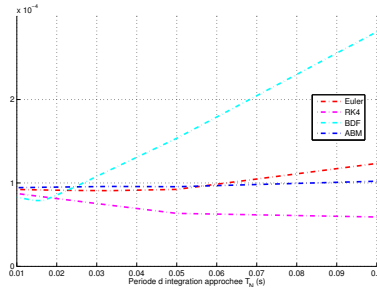
- 1: Initialisation :
 - Déterminer \mathbf{s}_0 et \mathbf{z}_0
 - Fixer $t_k = t_0$, $\mathbf{s}_k = \mathbf{s}_0$ et $\mathbf{z}_k = \mathbf{z}_0$
 - 2: POUR CHAQUE intervalle $[t_k; t_{k+1}]$ FAIRE
 - 3: Fixer : $t_n = t_k$, $\mathbf{s}_n = \mathbf{s}_k$ et $\mathbf{z}_n = \mathbf{z}_k$
 - 4: POUR CHAQUE intervalle $[t_n, t_{n+1}]$ FAIRE
 - 5: Déterminer le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}(t_n)$ à l'instant t_n
 - 6: Estimer les l points $\boldsymbol{\psi}_j = (X_j, Y_j, z_j)^T$ en résolvant le système d'équations différentielles (III.6).
 - 7: Identifier les composantes A_{pq} du plan de l'ellipse
 - 8: Estimer les paramètres E_i de l'ellipse sur la base du système dynamique (III.9) en considérant que $\pi_i = E_i$.
 - 9: Reconstruire le vecteur des informations visuelles de l'instant t_{n+1} :
 $\mathbf{s} = (E_1, E_2, E_3, E_4, E_5, X_1, Y_1, \dots, X_l, Y_l)^T$
 - 10: $t_n = t_n + T_N$ /*itération du schéma*/
 - 11: FIN POUR
 - 12: Reconstruire le vecteur des informations visuelles de l'instant t_{k+1} :
 $\mathbf{s} = (E_1, E_2, E_3, E_4, E_5, X_1, Y_1, \dots, X_l, Y_l)^T$
 - 13: $t_k = t_k + T_{\text{ech}}$ /*itération de la commande*/
 - 14: FIN POUR
-

La reconstruction des points ayant été analysée dans le paragraphe précédent, nous nous focaliserons ici sur l'étape 8 de l'algorithme IV.1 consistant à estimer les paramètres E_i de l'ellipse. Compte tenu de la complexité du système (III.9) à résoudre dans ce cas, seules les méthodes numériques peuvent être mises en œuvre pour extraire une solution \hat{E}_i . Nous présentons les résultats obtenus sur les figures IV.5 et IV.6 pour les périodes d'échantillonnage T_{ech} valant 100 ms et 50 ms. Ces figures décrivent l'évolution des erreurs maximales $|E_i - \hat{E}_i|$ en fonction de T_N . Encore une fois les écarts entre les mesures E_i et les estimations \hat{E}_i sont globalement faibles, démontrant l'efficacité de notre approche lorsque le mouvement de la caméra est peu perturbé. Cependant, contrairement au cas précédent

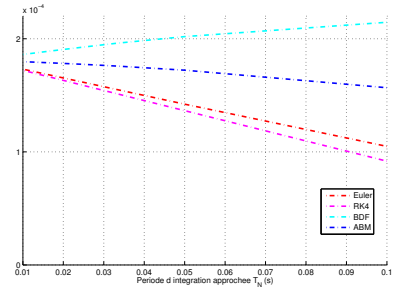
⁵. L'expression de la matrice d'interaction relative aux ellipses est disponible dans [CHAUMETTE, 1990].



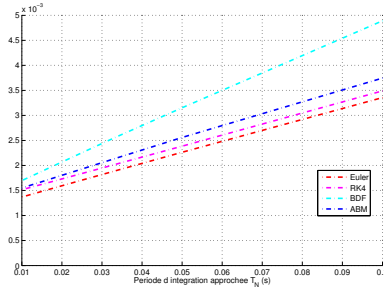
IV.5.a – Erreur maximale
 $|E_1 - \hat{E}_1|$.



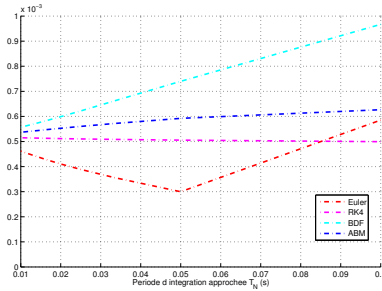
IV.5.b – Erreur maximale
 $|E_2 - \hat{E}_2|$.



IV.5.c – Erreur maximale
 $|E_3 - \hat{E}_3|$.

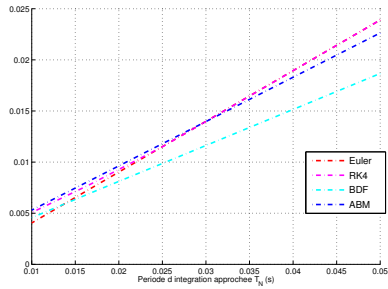


IV.5.d – Erreur maximale
 $|E_4 - \hat{E}_4|$.

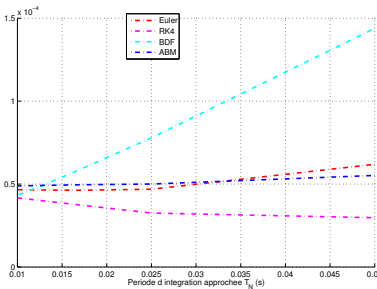


IV.5.e – Erreur maximale
 $|E_5 - \hat{E}_5|$.

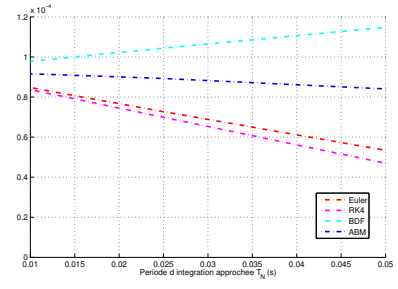
FIG. IV.5 – Les erreurs dans le cas de l'ellipse, avec $T_{ech} = 100$ ms.



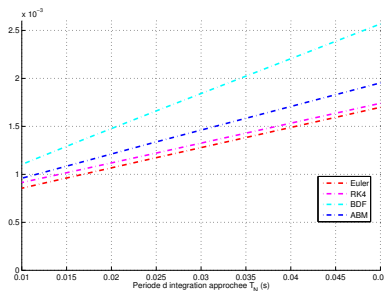
IV.6.a – Erreur maximale
 $|E_1 - \hat{E}_1|$.



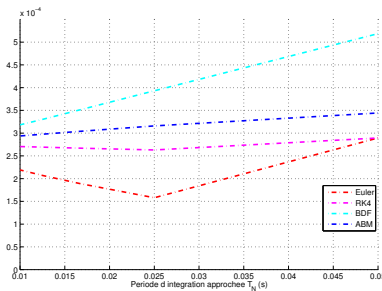
IV.6.b – Erreur maximale
 $|E_2 - \hat{E}_2|$.



IV.6.c – Erreur maximale
 $|E_3 - \hat{E}_3|$.



IV.6.d – Erreur maximale
 $|E_4 - \hat{E}_4|$.



IV.6.e – Erreur maximale
 $|E_5 - \hat{E}_5|$.

FIG. IV.6 – Les erreurs dans le cas de l'ellipse, avec $T_{ech} = 50$ ms.

où seuls des points étaient traités, les erreurs $|E_i - \hat{E}_i|$ ne diminuent pas systématiquement avec T_N (cf. figures IV.5.b, IV.5.c, IV.5.e, IV.6.b, IV.6.c et IV.6.e). Ce résultat, un peu singulier, est lié aux approximations cumulées lors des estimations successives effectuées aux étapes 6, 7 et 8 de l'algorithme IV.1 (cf. remarque IV.2).

Remarque IV.2: Dans le cas des méthodes d'intégration numérique, il y a classiquement deux types d'erreurs : d'une part, l'erreur liée la précision des calculs numériques et d'autre part, l'erreur de troncature associée au schéma numérique considéré. Cette dernière se décompose en (cf. figure IV.7) :

- l'erreur de troncature globale qui correspond à l'erreur entre la solution exacte $\chi(t)$ et celle calculée $\hat{\chi}$. Elle est de l'ordre $\mathcal{O}(T_N^r)$, où r est l'ordre de la méthode.
- l'erreur de troncature locale qui correspond à l'erreur commise à chaque pas du schéma numérique (elle est de l'ordre $\mathcal{O}(T_N^{r+1})$)

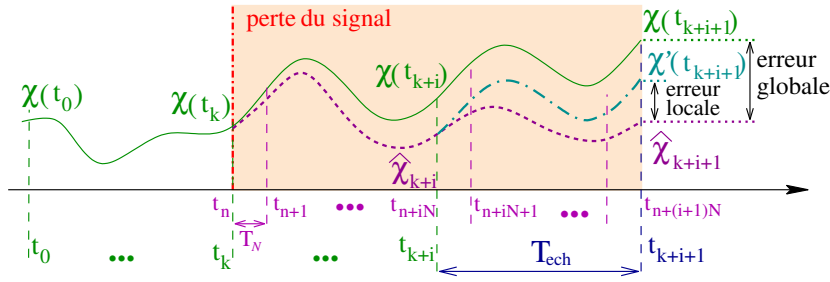
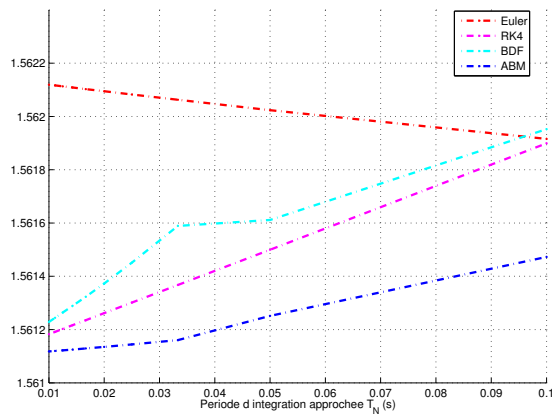


FIG. IV.7 – Erreur locale et globale.

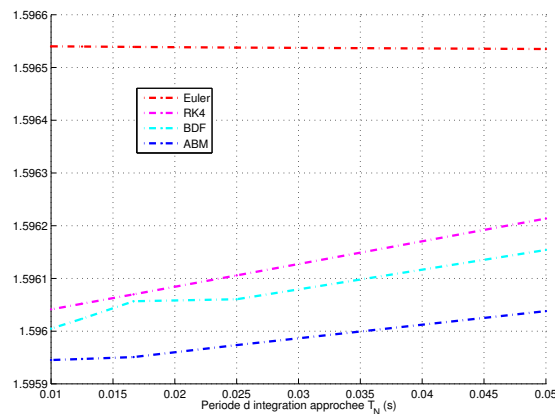
Dans la mesure où nous reconstruisons les l points $\psi_j = (X_j, Y_j, z_j)$ pour identifier d'abord les composantes A_{pq} du plan et puis en déduire une estimation des paramètres E_i , ces différentes erreurs s'accumulent à chaque pas T_N de nos schémas numériques (et ceci d'autant plus que nous augmentons le nombre de sous-intervalles N). L'erreur globale ne diminue donc plus systématiquement avec T_N .

De plus, il faut également noter que les points P_j ont été choisis comme appartenant à l'ellipse. Cette appartenance doit donc être encore vérifiée pour tous les points et les paramètres E_i estimés. La qualité de la reconstruction de \mathbf{s} peut donc être évaluée en déterminant l'erreur maximale suivante pour l'ensemble des l points :

$$\varepsilon_{\text{ellipse}} = \max_j (X_j^2 + E_1 Y_j^2 + 2E_2 X_j Y_j + 2E_3 X_j + 2E_4 Y_j + E_5)$$



IV.8.a – Erreur $\varepsilon_{\text{ellipse}}$, pour $T_{\text{ech}} = 100$ ms.



IV.8.b – Erreur $\varepsilon_{\text{ellipse}}$, pour $T_{\text{ech}} = 50$ ms.

FIG. IV.8 – Les erreurs $\varepsilon_{\text{ellipse}}$ de reconstruction de l'ellipse.

Les figures IV.8.a et IV.8.b décrivent l'évolution de cette erreur en fonction de T_N . Celle-ci n'étant pas nulle, cela implique que certains points $P_j(X_j, Y_j)$ ne sont plus systématiquement sur l'ellipse reconstruite. Cette décorrélation est due au fait que les points et les paramètres de l'ellipse sont déterminés par deux processus indépendants, en faisant abstraction du lien qui les unissait. Pour améliorer les performances de nos algorithmes, une première solution serait d'identifier l'ellipse qui passe au plus près des l points estimés. Cela nous ramène alors au cas ③ présenté à la section III.2.3.4, où nous utilisons des points pour reconstruire des primitives plus évoluées. Une seconde alternative serait de ramener les points estimés sur l'ellipse reconstruite, tout au moins ceux qui en sont trop éloignés.

Les moments de l'image

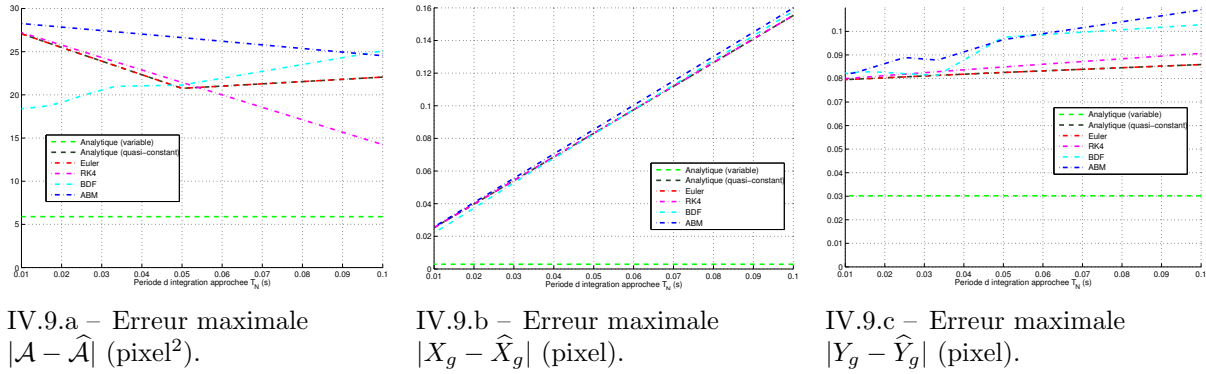
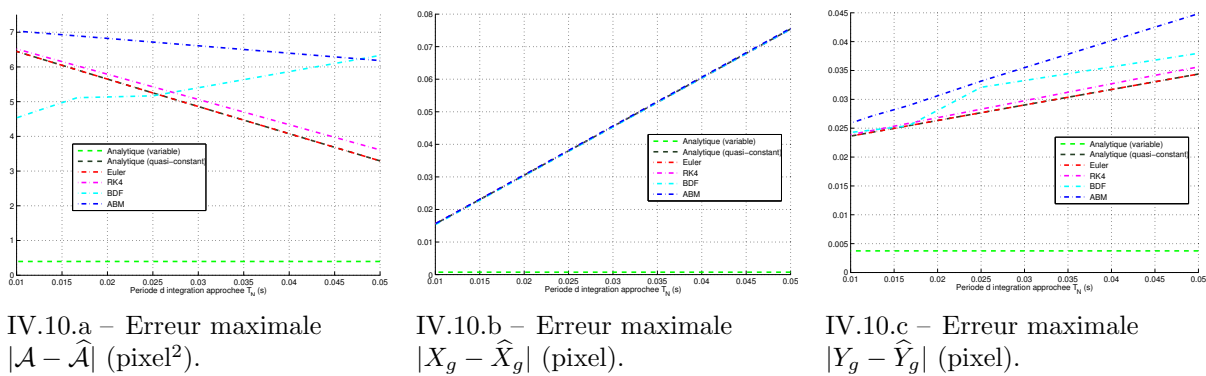
Outre les points et les primitives géométriques classiques, les moments de l'image constituent des données perceptuelles intéressantes pour la définition de lois de commande référencées vision. À ce titre, nous avons également évalué la pertinence de nos approches dans ce cas. Pour cela, nous avons considéré le motif visuel suivant (*cf.* figure IV.2.b) :

$$\mathbf{s} = (\mathcal{A}, X_g, Y_g, X_1, Y_1, \dots, X_l, Y_l)^T \quad (\text{IV.5})$$

où, pour la région \mathbf{R} de l'objet considéré dans l'image, $\mathcal{A} = m_{00}$ représente sa surface et (X_g, Y_g) désignent les coordonnées de son centre de gravité. Les couples (X_1, Y_1) à (X_l, Y_l) sont les coordonnées de $l = 15$ points appartenant au contour \mathbf{C} de \mathbf{R} . Comme nous l'avons préalablement évoqué, il paraît difficile de déterminer les moments m_{ij} sur la seule base du système dynamique (III.16), qui requiert la connaissance des moments d'ordre $i + j + 1$. De ce fait, nous reconstruirons ces moments sur la base de l'estimation des coordonnées (X_j, Y_j) de chaque point P_j (*cf.* remarque III.3). Nous proposons donc l'algorithme IV.2.

Algorithme IV.2 Estimation des moments de l'image.

- 1: Initialisation :
 - Déterminer \mathbf{s}_0 et \mathbf{z}_0
 - Fixer : $t_k = t_0$, $\mathbf{s}_k = \mathbf{s}_0$ et $\mathbf{z}_k = \mathbf{z}_0$
 - 2: POUR CHAQUE intervalle $[t_k; t_{k+1}]$ FAIRE
 - 3: Fixer : $t_n = t_k$, $\mathbf{s}_n = \mathbf{s}_k$ et $\mathbf{z}_n = \mathbf{z}_k$
 - 4: POUR CHAQUE intervalle $[t_n, t_{n+1}]$ FAIRE
 - 5: Déterminer le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}(t_n)$ à l'instant t_n
 - 6: Calculer une estimation des l points $\boldsymbol{\psi}_j = (X_j, Y_j, z_j)$ en résolvant le système d'équations différentielles (III.6).
 - 7: Déduire les moments de l'image nécessaires à partir des coordonnées des points calculées ci-dessus (*cf.* remarque III.3).
 - 8: Reconstruire le vecteur des informations visuelles de l'instant t_{n+1} :
 $\mathbf{s} = (\mathcal{A}, X_g, Y_g, X_1, Y_1, \dots, X_l, Y_l)^T$
 - 9: $t_n = t_n + T_N$
 - 10: FIN POUR
 - 11: Reconstruire le vecteur des informations visuelles de l'instant t_{k+1} :
 $\mathbf{s} = (\mathcal{A}, X_g, Y_g, X_1, Y_1, \dots, X_l, Y_l)^T$
 - 12: $t_k = t_k + T_{\text{ech}}$
 - 13: FIN POUR
-


 FIG. IV.9 – Les erreurs sur les moments, avec $T_{ech} = 100$ ms.

 FIG. IV.10 – Les erreurs sur les moments, avec $T_{ech} = 50$ ms.

Nous nous focalisons ici sur l'étape 7 de l'algorithme IV.2. Ainsi que nous l'avons rappelé précédemment, les moments sont calculés à partir des points estimés. Les méthodes de reconstruction utilisées sont donc celles développées pour ces primitives visuelles. Les figures IV.9 et IV.10 présentent donc les résultats obtenus pour les techniques analytiques et numériques. Tout comme précédemment, la résolution analytique avec le torseur \mathcal{T}_{C/R_0} variable reste la méthode la plus performante. Cependant, elle présente des erreurs plus importantes que dans le cas du point, où les écarts obtenus restaient de l'ordre de grandeur du bruit de calcul. Ce résultat semble cohérent puisque les erreurs d'estimation s'ajoutent lors des étapes 6 et 7 de notre algorithme. Cette accumulation explique également que l'erreur d'estimation (notamment sur l'aire \mathcal{A}) ne diminue plus systématiquement en fonction de T_N . Nous obtenons alors un résultat similaire à celui de l'ellipse. En ce qui concerne les autres techniques, les figures montrent que les schémas à pas simple Euler et RK4 apparaissent relativement efficaces, et ce d'autant plus que le pas d'intégration est faible. De même, l'algorithme BDF et la méthode de résolution analytique conduisent à des performances relativement proches de ces derniers, la technique la moins intéressante dans ce cas étant le schéma d'Adams ABM. Cela s'explique par le fait que cette dernière utilise une interpolation de la dérivée $\dot{\chi} = \varphi(\chi, t)$ pour reconstruire la valeur de $\chi(t)$, elle est donc plus sensible aux bruits occasionnés par les approximations successives.

Enfin, d'une manière générale, notons que les erreurs restent relativement faibles dans l'ensemble et démontrent l'intérêt de notre approche pour la gestion des pannes de la caméra ou du traitement d'images pendant une tâche référencée vision.

IV.1.1.2 Expérimentations

Nous avons expérimenté les estimateurs numériques sur le robot Super Scout de notre étude dans le cadre de la réalisation d'une tâche de positionnement référencée vision. À cet effet, nous utilisons ici un amer visuel constitué de quatre points de couleur qui facilite et accélère le processus d'extraction des informations visuelles dans l'image. Par ailleurs, sur la plate-forme expérimentale, la période d'échantillonnage de la commande est de l'ordre⁶ de $T_{ech} \simeq 50$ ms. Pour chaque schéma numérique, le robot effectue la même tâche de navigation : depuis la même configuration de départ, il doit atteindre un même motif de référence s^* (cf. figure IV.11). Au début de la mission, la commande \dot{q}_{AV} exploite les mesures fournies par le processus de traitement d'image, et le robot commence alors à converger vers le but. Dans le même temps, les différents algorithmes numériques sont également initialisés et lancés. Puis, comme précédemment, après 10 itérations, l'amer visuel est artificiellement occulté dans le but de simuler un défaut de la caméra. De ce fait, à partir de cet instant, la commande \dot{q}_{AV} est évaluée à partir des estimations du motif visuel \hat{s} .

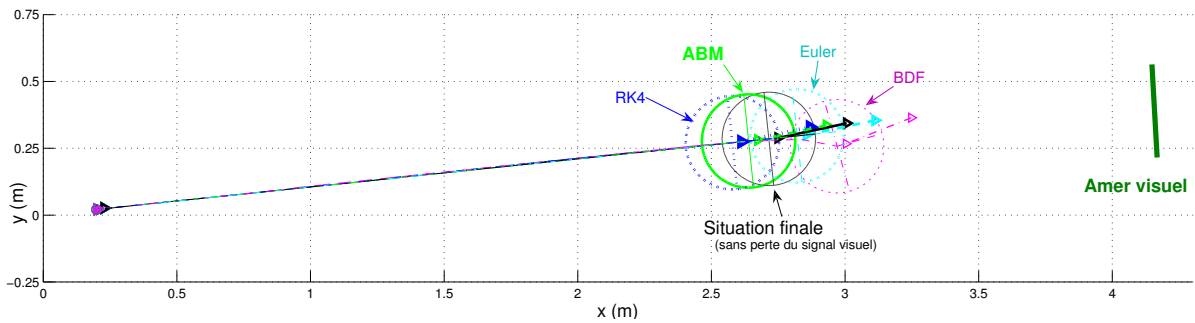


FIG. IV.11 – Trajectoire du robot obtenue pour les différentes méthodes numériques.

Schémas	s / z	erreur moyenne	écart type	erreur maximum
Euler	s (pix)	0.87	1.54	9.68
	z (m)	0.0706	0.13618	0.7232
RK4	s (pix)	0.79	1.19	7.02
	z (m)	0.0496	0.087037	0.44495
ABM	s (pix)	1.37	1.36	5.92
	z (m)	0.0224	0.0235	0.0935
BDF	s (pix)	2.17	1.89	7.69
	z (m)	0.0234	0.0271	0.1184

TAB. IV.1 – Synthèse des résultats sur les erreurs, sur la durée de toute la mission.

La figure IV.11 décrit l'évolution de la trajectoire du robot, et la table IV.1 synthétise les résultats correspondant aux différentes erreurs pour chaque algorithme numérique considéré. Comme nous pouvons le constater, malgré la perte du signal visuel au bout de 10 itérations, la caméra parvient à atteindre un voisinage de la situation finale désirée, démontrant la validité de notre approche dans des conditions expérimentales "réelles".

6. Dans la mesure où le système robotique est commandé depuis un PC portable embarqué évoluant sous Linux qui n'est pas temps réel, les périodes indiquées ne peuvent être qu'approximées.

En outre, pour la tâche référencée vision considérée, le schéma ABM apparaît comme la méthode la plus efficace puisqu'il présente le plus petit écart type et la plus faible erreur maximale. L'algorithme de Runge–Kutta RK4 offre lui aussi de bonnes performances, tandis que le schéma d'Euler est le moins précis. Enfin, l'algorithme BDF conduit à des résultats corrects, mais il est généralement plus efficace lorsqu'il y a de plus fortes variations sur la commande (*cf.* section IV.2). La table IV.1 montre de plus que l'erreur commise sur la reconstruction des primitives visuelles \mathbf{s} est ici peu importante quels que soient les schémas d'intégration numérique mis en œuvre, ce qui signifie que le système n'a été que peu perturbé pendant l'exécution de la mission.

Toutefois, les résultats présentés dans cette partie semblent moins bons que ceux obtenus en simulation. Cette différence est principalement liée aux contraintes expérimentales suivantes :

- les schémas d'intégration numérique nécessitent une caractérisation *précise* de la période d'échantillonnage T_{ech} du système. Or, dans la mesure où la plate-forme expérimentale ne dispose pas d'un système d'exploitation temps réel, T_{ech} ne peut être qu'approximée.
- nos techniques d'estimation supposent que le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ est entièrement connu. Cependant, ne disposant pas de toutes⁷ les mesures des vitesses réellement appliquées par le système robotique, nous avons exploité les commandes déterminées par l'asservissement visuel \dot{q}_{AV} . Cela suppose donc, que notre plate-forme robotique réalise *parfaitement* et *instantanément* les consignes en vitesse de notre stratégie de commande.
- nous n'avons pas intégré les erreurs de mesure propres au processus de traitement d'images utilisé qui sont de l'ordre du pixel dans nos résultats de simulation.

IV.1.2 Amélioration des performances de l'asservissement

Les approches référencées capteurs nécessitent l'utilisation de capteurs et de processus de traitement de données associés rapides, capables de fournir les mesures à une fréquence compatible avec la fréquence d'échantillonnage de l'asservissement T_{ech} . Les travaux en commande référencée vision ont donc surtout porté sur des processus choisis pour leur efficacité dans l'obtention des informations. Cependant, disposer des données requises à des cadences importantes s'avère parfois difficile. En particulier, dans notre contexte expérimental, la période de la commande T_{ech} est fixée, approximativement, à 50 ms, tandis que celle liée au capteur de vision est de l'ordre de 100 à 150 ms (selon l'environnement d'exécution). Notre commande est donc calculée sur la base des mêmes mesures du signal visuel, durant deux à trois pas d'exécution, ce qui peut parfois occasionner quelques problèmes notamment si le robot évolue à des vitesses élevées. Nous nous proposons donc ici d'utiliser nos techniques d'estimation pour prédire les valeurs des indices visuels \mathbf{s} entre deux acquisitions d'image. Plus précisément, il s'agit de sous-échantillonner la période du capteur $T_{\text{capt}} = CT_{\text{ech}}$ de manière à reconstruire le signal sensoriel à la fréquence de l'asservissement T_{ech} (*cf.* figure IV.12). Typiquement deux cas peuvent alors se présenter :

7. Sur notre robot Super Scout nous n'avons pas accès à la vitesse de la platine ϖ , et nous ne pouvons disposer que des vitesses relatives à la base mobile v et ω .

- ① Une mesure \mathbf{s} provenant du processus de traitement d'images est disponible à l'instant t_k de l'asservissement :
 - les conditions initiales χ_k de l'estimateur sont mises à jour.
 - la commande peut être calculée à partir de \mathbf{s} (cf. remarque IV.3).
- ② Il n'y a pas de mesure \mathbf{s} disponible :
 - l'estimateur est itéré sur la base de ses données reconstruites à l'instant précédent.
 - la commande est calculée à partir des valeurs reconstruites $\hat{\mathbf{s}}$ fournies par l'estimateur (cf. figure IV.1).

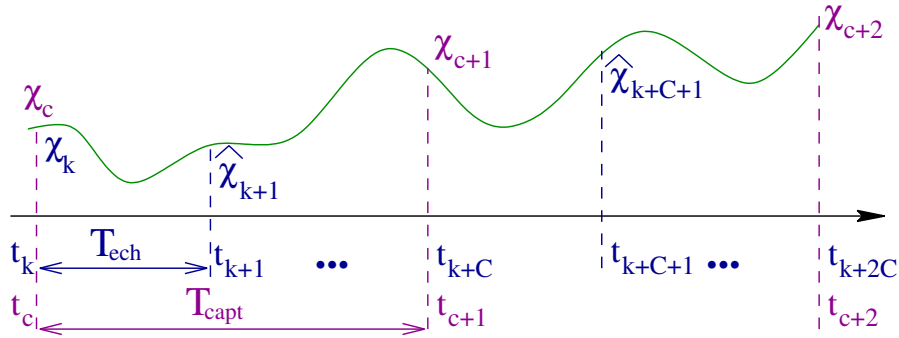


FIG. IV.12 – Sous-échantillonnage de la période du capteur de vision.

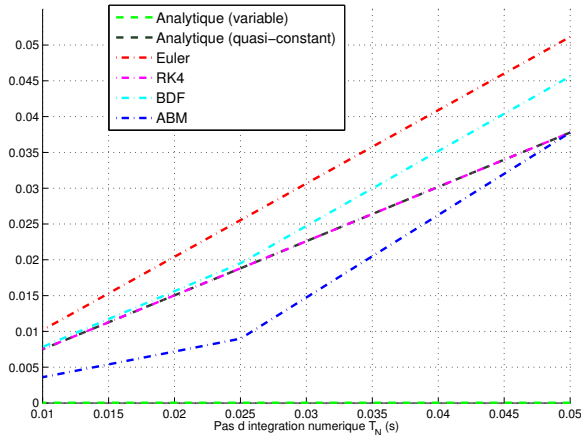
Remarque IV.3: *Nous ne nous intéressons pas ici aux problèmes de lissage du passage de la valeur estimée $\hat{\mathbf{s}}$ à la valeur mesurée \mathbf{s} . Cependant, compte tenu que l'estimateur est recalé sur la mesure \mathbf{s} à chaque instant t_c du capteur de vision, et que les erreurs observées sur les exemples précédents sont faibles, ces problèmes devraient être relativement limités.*

Afin d'évaluer l'efficacité de cette approche nous avons effectué de nombreuses simulations à l'aide de **Matlab**. Pour chaque cas étudié, nous nous sommes placés dans des conditions identiques de tests, similaires à ceux du paragraphe IV.1.1.1, soit :

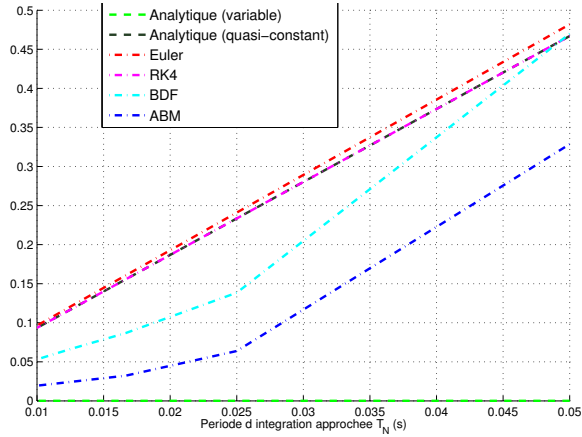
- la même tâche de positionnement que sur la figure IV.2.a.
- les mêmes indices visuels, définis successivement par des points, des ellipses et des moments de l'images, et donnés respectivement par les relations (IV.1), (IV.3) et (IV.5).

Nous ne considérons toutefois que le cas où la période d'échantillonnage de la commande T_{ech} vaut 50 ms. Par ailleurs, nous avons choisi de traiter un cas dégénéré où les mesures sensorielles \mathbf{s} fournies par le processus de traitement d'images sont rafraîchies toutes les $T_{capt} = 1$ s. Ainsi, sans sous-échantillonnage des mesures du capteur, la loi de commande évoluerait sur les mêmes valeurs de \mathbf{s} durant 20 itérations.

Les figures IV.13, IV.14 et IV.15 décrivent les courbes des erreurs obtenues pour les différents motifs visuels considérés (respectivement les points, les ellipses et les moments de l'image) en fonction de T_N . Comme l'illustrent ces figures nos algorithmes d'estimation conduisent à des résultats proches de ceux obtenus à la section précédente, avec des erreurs qui restent relativement faibles. Ainsi, la résolution analytique avec \mathcal{T}_{C/R_0} variable sur $[t_k; t_{k+1}]$ apparaît toujours comme la technique la plus précise. Parmi les résolutions approchées, la méthode d'Adams (ABM) donne de bons résultats, tandis que celle de Gear (BDF) semble moins efficace. De plus, tout comme précédemment, les erreurs ne

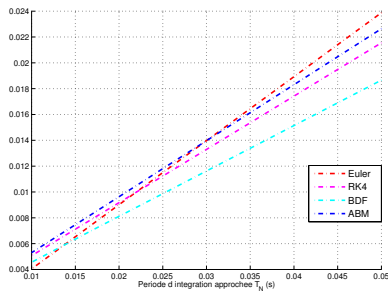


IV.13.a – Erreur maximale $\|s - \hat{s}\|$ (pixel).

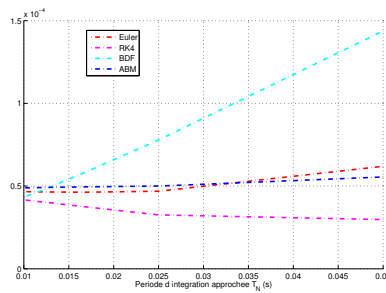


IV.13.b – Erreur maximale $\|z - \hat{z}\|$ (mm).

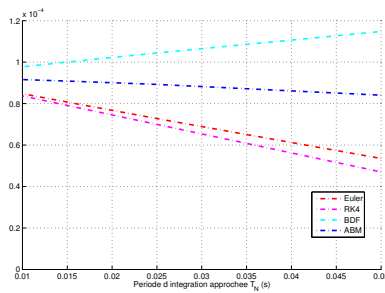
FIG. IV.13 – Erreurs pour le cas des points : $s = (X_1, Y_1, \dots, X_n, Y_n)^T$, avec $T_{ech} = 50$ ms.



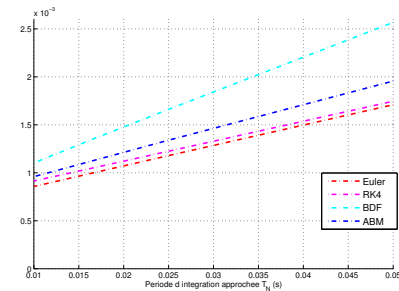
IV.14.a – Erreur maximale $|E_1 - \hat{E}_1|$.



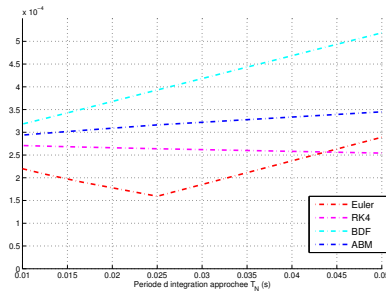
IV.14.b – Erreur maximale $|E_2 - \hat{E}_2|$.



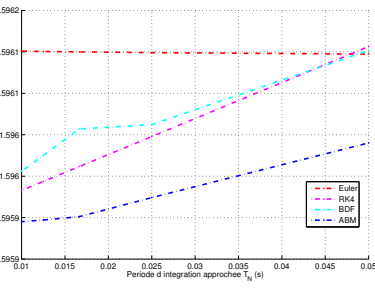
IV.14.c – Erreur maximale $|E_3 - \hat{E}_3|$.



IV.14.d – Erreur maximale $|E_4 - \hat{E}_4|$.



IV.14.e – Erreur maximale $|E_5 - \hat{E}_5|$.

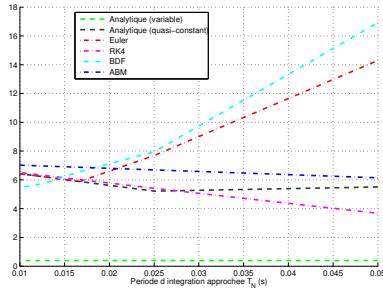


IV.14.f – Erreur $\epsilon_{ellipse}$.

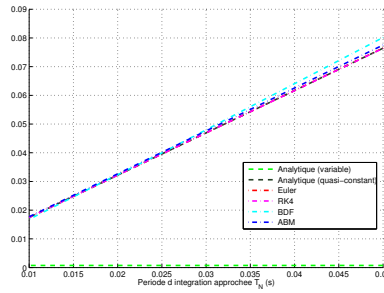
FIG. IV.14 – Les erreurs dans le cas de l'ellipse, avec $T_{ech} = 50$ ms .

diminuent plus systématiquement en fonction de T_N , du moins dans le cas de l'ellipse et des moments de l'image où il est nécessaire d'effectuer plusieurs approximations successives.

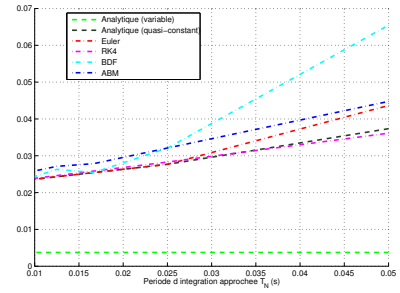
Finalement, nos algorithmes de reconstruction du motif visuel apparaissent comme un outil intéressant puisqu'ils permettent de fournir une estimation du motif visuel entre deux acquisitions d'images, et donc de rafraîchir la loi de commande à une cadence plus élevée que celle du capteur de vision. Ainsi, il est possible de considérer des processus de traitement d'images complexe, exploitant des données visuelles riches et denses.



IV.15.a – Erreur maximale $|\mathcal{A} - \hat{\mathcal{A}}|$ (pixel²).



IV.15.b – Erreur maximale $|X_g - \hat{X}_g|$ (pixel).



IV.15.c – Erreur maximale $|Y_g - \hat{Y}_g|$ (pixel).

FIG. IV.15 – Les erreurs sur les moments, avec $T_{ech} = 50$ ms.

IV.1.3 Bilan

Nous avons donc validé les différentes techniques d'estimation proposées au chapitre III dans le cadre de la réalisation d'une tâche de navigation référencée vision dans un environnement libre. Les résultats obtenus ont démontré la validité de notre approche dans ce contexte. Celle-ci, quoique conduisant à des lois de commande en boucle ouverte dans ce cas précis, constitue néanmoins une première réponse intéressante au problème de la gestion de la perte temporaire du signal visuel consécutif à un défaut ou une panne de la caméra ou du traitement d'images embarqué. Elle offre également des solutions pertinentes pour l'amélioration des performances de l'asservissement mis en œuvre.

Nous revenons maintenant vers la problématique qui a initié nos travaux et allons montrer comment intégrer ces techniques d'estimation au sein de nos lois de commande référencées multi-capteurs.

IV.2 Asservissement visuel et évitement de collision avec gestion de la perte du motif visuel

Nous nous intéressons donc à nouveau à la réalisation d'une tâche référencée vision dans un environnement encombré d'obstacles susceptibles d'occulter le motif visuel. Pour cela, nous avons élaboré au chapitre II différentes stratégies de navigation référencées multi-capteurs permettant d'éviter les collisions et les occultations. Cependant, les résultats obtenus ont montré que ces approches étaient restreintes aux missions pour lesquelles il existe une trajectoire permettant de rejoindre le motif désiré \mathbf{s}^* tout en préservant la visibilité des indices visuels \mathbf{s} et l'intégrité du système robotique.

Nous souhaitons maintenant étendre ces stratégies de manière à ce que l'occultation temporaire du motif visuel soit autorisée, si elle est nécessaire au bon déroulement de la tâche de navigation. Pour cela, nous nous proposons de synthétiser des lois de commande référencées multi-capteurs intégrant les méthodes d'estimation développées dans le chapitre III. Nous présentons, dans un premier temps, la stratégie de commande élaborée, avant de l'illustrer par le biais de deux exemples applicatifs.

IV.2.1 Stratégie de commande avec gestion de l'occultation du signal visuel

La stratégie de commande présentée ici repose sur un principe similaire à celui développé dans le chapitre II. Nous considérons donc trois correcteurs différents :

1. le premier, \dot{q}_{AV} , réalise la tâche de navigation référencée vision désirée (*i.e.* amener le motif \mathbf{s} à sa valeur de référence \mathbf{s}^*). Il est donc directement donné par l'équation (I.18).
2. le deuxième, \dot{q}_{OCC} , doit permettre de gérer les occultations de type ③ (*cf.* figure II.11, page 71), c'est-à-dire celles qui se produisent loin des obstacles. Il peut donc être pleinement défini par le correcteur \dot{q}_{OS} (II.24) qui préserve la visibilité du motif visuel tout en continuant à converger au mieux vers \mathbf{s}^* .
3. le troisième, \dot{q}_{COLL} , doit être conçu pour garantir la non collision et gérer la perte éventuelle du signal sensoriel si des occultations se produisent au voisinage de l'obstacle. Il permet donc de traiter les occultations de type ① et ② (*cf.* figure II.11, page 71). Nous nous intéresserons à la synthèse de \dot{q}_{COLL} dans la suite.

Il s'agit donc maintenant de basculer d'un correcteur à l'autre en fonction des risques d'occultation et de collision définis par les paramètres μ_{occ} et μ_{coll} . Pour cela, nous proposons de reprendre la loi de commande (II.25), qui se réécrit ici de la manière suivante :

$$\dot{q} = (1 - \mu_{coll})(1 - \mu_{occ})\dot{q}_{AV} + (1 - \mu_{coll})\mu_{occ}\dot{q}_{OCC} + \mu_{coll}\dot{q}_{COLL} \quad \forall \mu_{coll}, \mu_{occ} \in [0; 1] \quad (IV.6)$$

où \dot{q}_{AV} et \dot{q}_{OCC} sont définis par les relations (I.18) et (II.24), le correcteur \dot{q}_{COLL} restant encore à déterminer.

Remarque IV.4: *Cette stratégie permet de traiter les trois types d'occultations référencés sur la figure II.11. Cependant, si le signal visuel devient indisponible en dehors de la phase de contournement suite à un défaut de la caméra ou du processus de traitement d'image, le problème redevient similaire au cas présenté à la section précédente IV.1.1. Il peut donc être traité de la même manière.*

Nous nous intéressons maintenant à la définition de la commande de contournement d'obstacle \dot{q}_{COLL} . Pour cela, nous nous appuyons sur les résultats présentés dans le chapitre I et plus particulièrement sur les travaux de TSAKIRIS *et al.* [1996, 1997]. Comme nous l'avons déjà mentionné, l'approche proposée consiste à commander séparément la base et la platine pendant la phase d'évitement. Ainsi, la première sera pilotée de manière à contourner les obstacles, ce qui conduit à définir la commande de la base mobile $\dot{q}_{base} = (v_{COLL}, \omega_{COLL})^T$ à l'aide des techniques⁸ décrites dans le tableau I.3. La platine sera quant à elle commandée pour compenser le mouvement d'évitement effectué par la base, tout en suivant le motif visuel *mesuré* \mathbf{s} ou *estimé* $\hat{\mathbf{s}}$, selon qu'il est disponible ou non. Dans le premier cas, ϖ_{COLL} sera donné par la relation (I.35) que nous rappelons ci-après :

$$\varpi_{coll} = -\frac{1}{\mathcal{L}_{Y_0} \cdot \mathbf{J}_{pl}} (\lambda_Y e_Y + \mathcal{L}_{Y_0} \cdot \mathbf{J}_{base} \cdot \dot{q}_{base})$$

8. Bien évidemment, toute technique permettant de définir des vitesses linéaire v_{COLL} et angulaire ω_{COLL} de la base mobile de manière à contourner l'obstacle peut être utilisée ici.

Dans le second, cette dernière expression n'est plus exploitable et nous nous proposons de calculer la commande de la platine sur la base des indices visuels reconstruits. Il vient :

$$\widehat{\omega}_{\text{coll}} = -\frac{1}{\mathcal{L}_{\widehat{Y}_0} \cdot \mathbf{J}_{\text{pl}}} (\lambda_{\widehat{Y}} \mathbf{e}_{\widehat{Y}} + \mathcal{L}_{\widehat{Y}_0} \cdot \mathbf{J}_{\text{base}} \cdot \dot{q}_{\text{base}}) \quad (\text{IV.7})$$

où $\mathbf{e}_{\widehat{Y}} = \widehat{Y}_0$ et $\mathcal{L}_{\widehat{Y}_0}$ sont évaluées à partir des estimations du motif visuel fournies par les algorithmes développés dans le chapitre III. La commande appliquée à la platine en phase d'évitement peut alors être définie en effectuant un basculement continu entre ces deux correcteurs en fonction du risque d'occultation $\mu_{\text{occ}} \in [0; 1]$. Nous obtenons finalement l'expression suivante :

$$\varpi_{\text{COLL}} = (1 - \mu_{\text{occ}})\varpi_{\text{coll}} + \mu_{\text{occ}}\widehat{\omega}_{\text{coll}} \quad (\text{IV.8})$$

Remarque IV.5: *Il est important de noter que, lorsque l'occultation se produit au voisinage de l'obstacle, seule la platine est commandée en boucle ouverte sur l'estimée $\widehat{\mathbf{s}}$ des indices visuels nécessaires. La base mobile reste pilotée en boucle fermée sur les informations sensorielles renvoyées par les capteurs de proximité embarqués, ce qui permet de garantir la non collision.*

Finalement, nous avons proposé une nouvelle stratégie de commande référencée multi-capteurs permettant de réaliser une tâche de navigation guidée par la vision en environnement encombré. Elle intègre l'ensemble des données sensorielles pertinentes pour la tâche à réaliser, ainsi qu'un processus d'estimation permettant de reconstruire les indices visuels d'intérêt chaque fois que cela est nécessaire (*cf.* figure IV.16). Il est ainsi possible de réaliser des tâches de navigation plus complexes, c'est-à-dire des tâches dont le succès dépend de la capacité du robot à tolérer la perte momentanée du signal visuel.

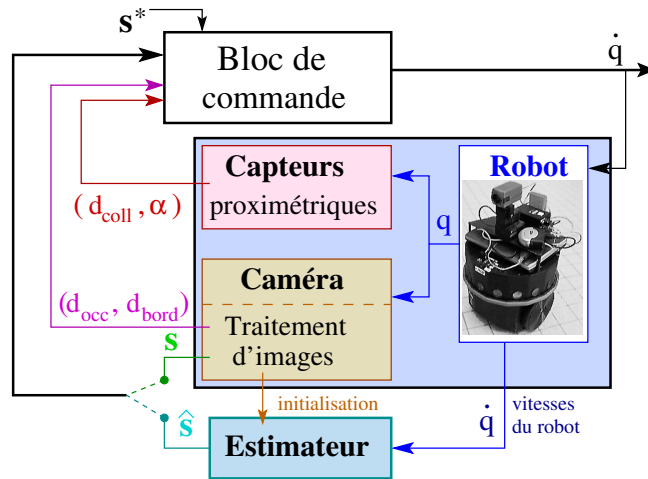


FIG. IV.16 – Commande référencée multi-capteurs et estimation.

Nous nous proposons maintenant de valider cette approche dans le cadre de la réalisation de deux tâches référencée vision en environnement encombré. Afin d'illustrer les différents schémas de commande \dot{q}_{COLL} possibles, nous mettons en œuvre les deux techniques d'évitement d'obstacles décrites dans le tableau I.3.

IV.2.2 Réalisation d'un asservissement visuel et évitement de collision par la méthode des potentiels rotatifs

Notre objectif est ici de valider la stratégie de commande référencée multi-capteurs proposée, lorsque l'évitement d'obstacle est effectué par la méthode des potentiels rotatifs (*cf.* paragraphe I.2.2.2). Dans ce cas, la commande d'évitement de collision est alors définie par le vecteur $\dot{q}_{COLL} = (v_{COLL}, \omega_{COLL}, \varpi_{COLL})^T$, où [FOLIO et CADENAT, ICINCO2007] :

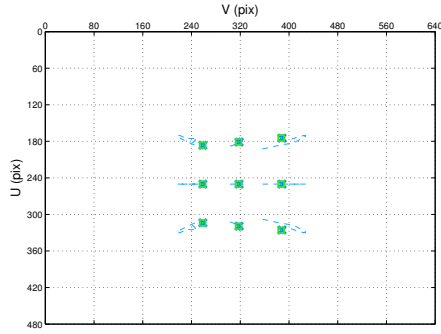
$$v_{COLL} = k_{coll}^v F \cos \beta \quad (I.25)$$

$$\omega_{COLL} = k_{coll}^\omega \frac{F}{D_F} \sin \beta \quad (I.26)$$

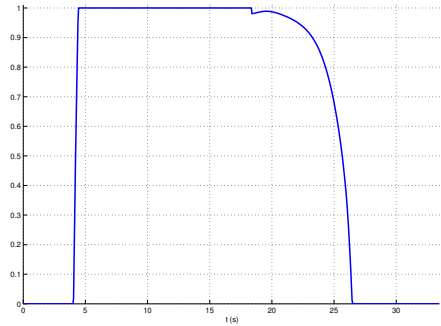
$$\varpi_{COLL} = (1 - \mu_{occ})\varpi_{coll} + \mu_{occ}\widehat{\varpi}_{coll} \quad (IV.7)$$

Nous présentons ici un exemple de simulation, permettant d'illustrer l'efficacité de notre stratégie décrite par (IV.6) avec gestion de la perte du signal visuel. Le motif visuel \mathbf{s} est ici constitué d'un ensemble de $n = 9$ points (*cf.* figure IV.17.a). La mission considérée consiste à positionner la caméra embarquée par rapport à cet amer tout en évitant deux obstacles : le premier est un mur, le second est un obstacle cylindrique de 30 cm de diamètre susceptible d'occulter l'amer visuel (*cf.* figure IV.17.d). Les enveloppes de contournement ξ_- , ξ_0 et ξ_+ sont fixées respectivement à 30 cm, 50 cm et 60 cm de l'obstacle. La période d'échantillonnage a été fixée à 50 ms, valeur correspondant à la boucle de commande de notre plate-forme expérimentale. Comme précédemment, nous avons réalisé cette tâche en utilisant les différents algorithmes d'estimation développés. Les résultats obtenus sont donnés sur les figures IV.17, IV.18 et IV.19.

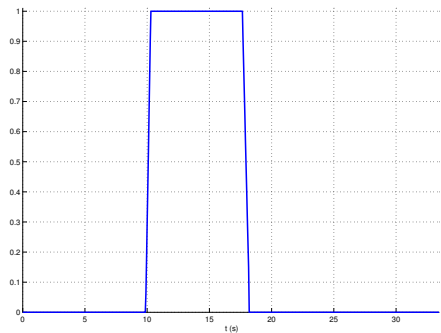
Les figures IV.17.a et IV.17.d représentent la trajectoire du robot et l'évolution des indices visuels et montrent que la tâche de navigation est parfaitement réalisée en dépit de l'occultation du motif visuel d'intérêt. Au début de la mission, il n'y a pas de risque d'occultation ($\mu_{occ} = 0$) ni de collision ($\mu_{coll} = 0$) comme le montrent les figures IV.17 et IV.18. Les algorithmes d'estimation peuvent donc être initialisés et lancés. Le robot est alors commandé uniquement par l'asservissement visuel \dot{q}_{AV} . Sous l'action de cette commande, le véhicule entre dans le voisinage du mur et le risque de collision croît pour atteindre 1. La commande globale \dot{q} bascule alors continûment vers \dot{q}_{COLL} , et le robot commence à s'écarter de l'obstacle. Pendant cette phase, le risque d'occultation augmente à son tour. Cependant, μ_{coll} étant fixé à 1, le correcteur \dot{q}_{COLL} reste appliqué au robot. La base mobile est donc pilotée de manière à garantir la non collision, tandis que, du fait de l'augmentation de μ_{occ} , la commande de la platine bascule progressivement de ϖ_{coll} (I.35) à $\widehat{\varpi}_{coll}$ (IV.7) calculée sur la base des indices visuels estimés. L'amer visuel d'intérêt peut donc être temporairement occulté par l'obstacle cylindrique, sans remettre en cause la réussite de la mission. Une fois l'obstacle occultant dépassé, μ_{occ} décroît, et ϖ_{COLL} bascule à nouveau vers ϖ_{coll} . Puis, les tâches d'asservissement visuel et de contournement d'obstacles indiquent la même direction, et le risque de collision μ_{coll} diminue à son tour, pour atteindre 0. Le robot est alors piloté par \dot{q}_{AV} jusqu'à ce qu'il ait fini de converger vers la situation désirée \mathbf{s}^* . La mission est alors parfaitement réalisée (*cf.* figure IV.17.a).



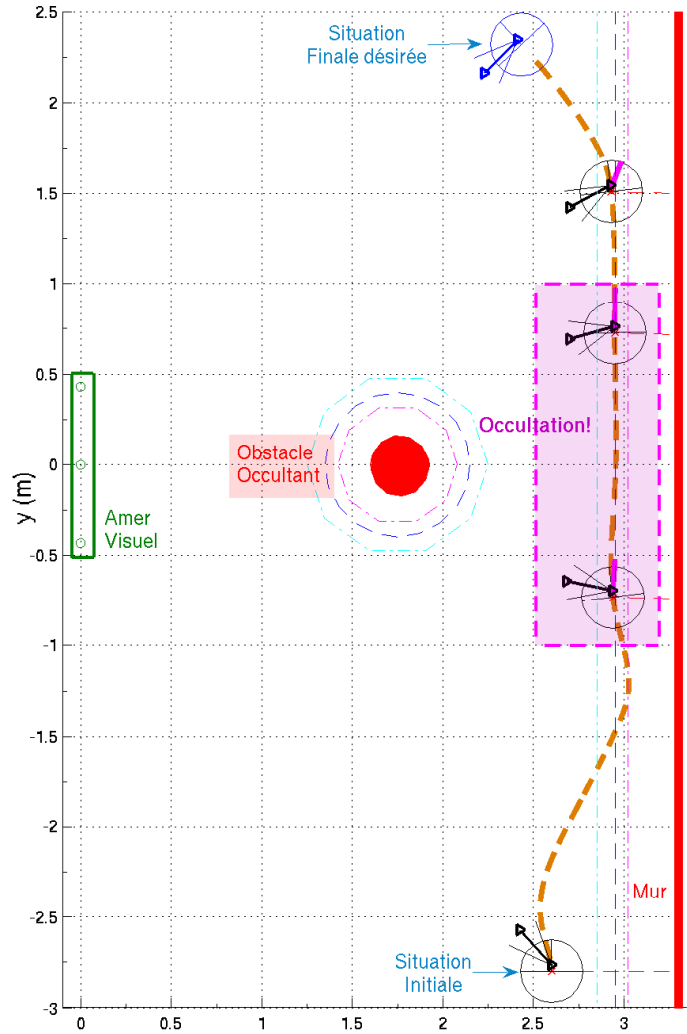
IV.17.a – Le motif visuel d'intérêt.



IV.17.b – Le risque de collision μ_{coll} .

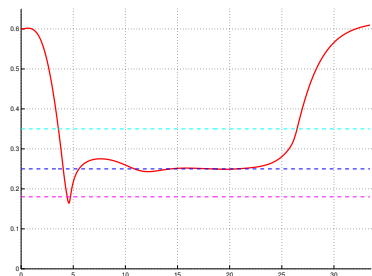


IV.17.c – Le risque d'occultation μ_{occ} .

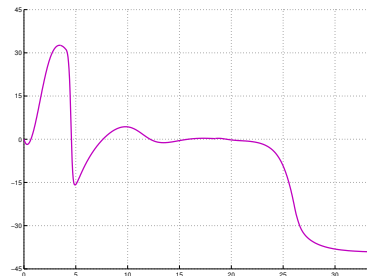


IV.17.d – Trajectoire du robot.

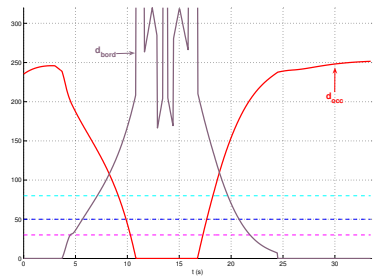
FIG. IV.17 – Asservissement visuel et évitement de collision par la méthode des potentiels rotatifs.



IV.18.a – La distance d_{coll} (m).



IV.18.b – L'orientation α (deg).



IV.18.c – Les distances d_{occ} et d_{bord} (pixels).

FIG. IV.18 – Mesures des risques de collision et d'occultation.

Comparons maintenant l'efficacité des différents algorithmes de reconstruction développés. La figure IV.19 présente la valeur maximale de $\|\mathbf{s} - \hat{\mathbf{s}}\|$ et de $\|\mathbf{z} - \hat{\mathbf{z}}\|$, en fonction du pas d'intégration T_N , utilisé. Tout comme dans le cas du point présenté

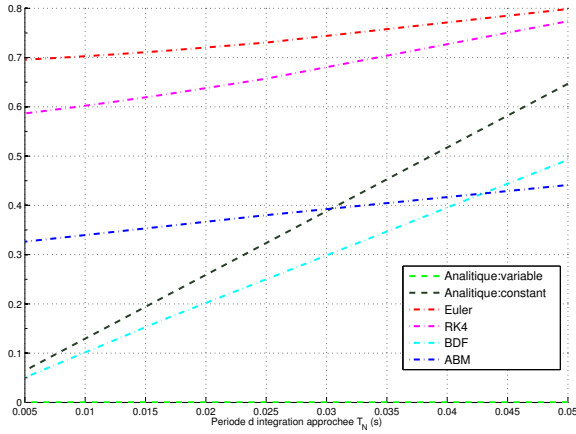
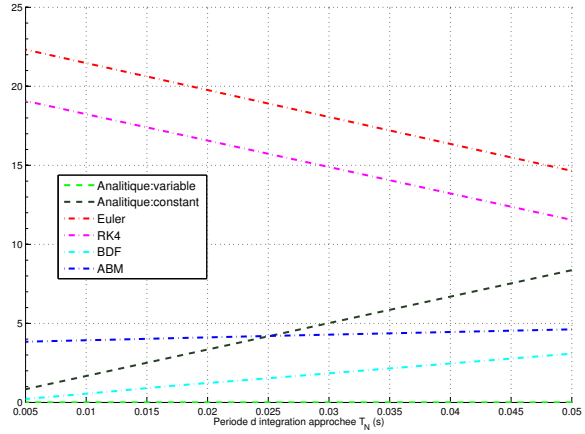

 IV.19.a – Erreur maximale $\|\mathbf{s} - \hat{\mathbf{s}}\|$ (pixel) sur le motif visuel en fonction de T_N

 IV.19.b – Erreur maximale $\|\mathbf{z} - \hat{\mathbf{z}}\|$ (mm) sur la profondeur en fonction de T_N .

 FIG. IV.19 – Erreurs d'estimation, avec $T_{\text{ech}} = 50$ ms.

au paragraphe IV.1.1.1, l'approche reposant sur la résolution analytique du système dynamique (III.17) avec le torseur $\mathcal{T}_{C/\mathcal{R}_0}$ variable est la plus précise. Parmi les techniques de résolution approchées, la méthode de Gear (BDF) semble ici la plus performante. En effet, du fait de l'évitement d'obstacles, la dynamique du système devient plus importante, et cette méthode s'avère mieux adaptée dans ce contexte. L'algorithme d'Adams conduit à des résultats convenables, tandis que les schémas à pas simple de Runge–Kutta RK4 et d'Euler sont les moins efficaces.

D'une manière générale, les erreurs restent ici relativement faibles (de l'ordre de quelques pixels pour le motif visuel et du millimètre pour la profondeur), et elles diminuent avec T_N . Les résultats obtenus démontrent donc la validité de nos algorithmes malgré la présence de perturbations imprévues sur la caméra liées au contournement des obstacles. L'intérêt du découplage base/caméra pendant la phase de contournement est ainsi mis en lumière.

IV.2.3 Réalisation d'une tâche référencée multi-capteurs par exploitation conjointe du formalisme des fonctions de tâche et du suivi de chemin

Notre intention est à présent d'appliquer notre stratégie de commande référencée multi-capteurs, lorsque l'évitement d'obstacle est réalisé au moyen du formalisme du suivi de chemin (*cf.* paragraphe I.2.3). Dans ce contexte, la commande de contournement d'obstacle est alors donnée par la commande $\dot{q}_{\text{COLL}} = (v_{\text{COLL}}, \omega_{\text{COLL}}, \varpi_{\text{COLL}})^T$ ci-après [FOLIO et CADENAT, ECC2007] :

$$v_{\text{COLL}} = v_0 \quad (\text{I.28})$$

$$\omega_{\text{COLL}} = -v_{\text{coll}} (k\delta + \alpha + 2k \sin \alpha - \bar{\chi} \cos \alpha) \quad (\text{I.29})$$

$$\varpi_{\text{COLL}} = (1 - \mu_{\text{occ}})\varpi_{\text{coll}} + \mu_{\text{occ}}\hat{\varpi}_{\text{coll}} \quad (\text{IV.7})$$

Afin d'illustrer l'efficacité de l'approche développée nous présentons ici un exemple de simulation. Le motif visuel \mathbf{s} est ici constitué d'un ensemble de $n = 25$ points. La tâche de navigation considérée consiste à réaliser une tâche référencée vision tout en contournant deux obstacles : le premier est un muret de 4.55 m de longueur pour 20 cm de large et de hauteur inférieure à celle de la caméra (*i.e.* pas de risque d'occultation), le second est un obstacle cylindrique de 30 cm de diamètre pouvant occulter l'amer visuel d'intérêt (*cf.* figure IV.20). Les enveloppes de contournement ξ_- , ξ_0 et ξ_+ sont fixées respectivement à 30 cm, 50 cm et 60 cm de l'obstacle. La période d'échantillonnage a été ici aussi fixée à 50 ms. De plus, pour analyser les performances de nos algorithmes de reconstruction, tout comme précédemment, nous avons appliqué successivement nos différentes méthodes en partant de la même configuration initiale pour atteindre le même motif visuel de référence \mathbf{s}^* . Ceux-ci sont initialisés et lancés dès le début de la mission. Les résultats obtenus sont donnés sur les figures IV.20, IV.21 et IV.23.

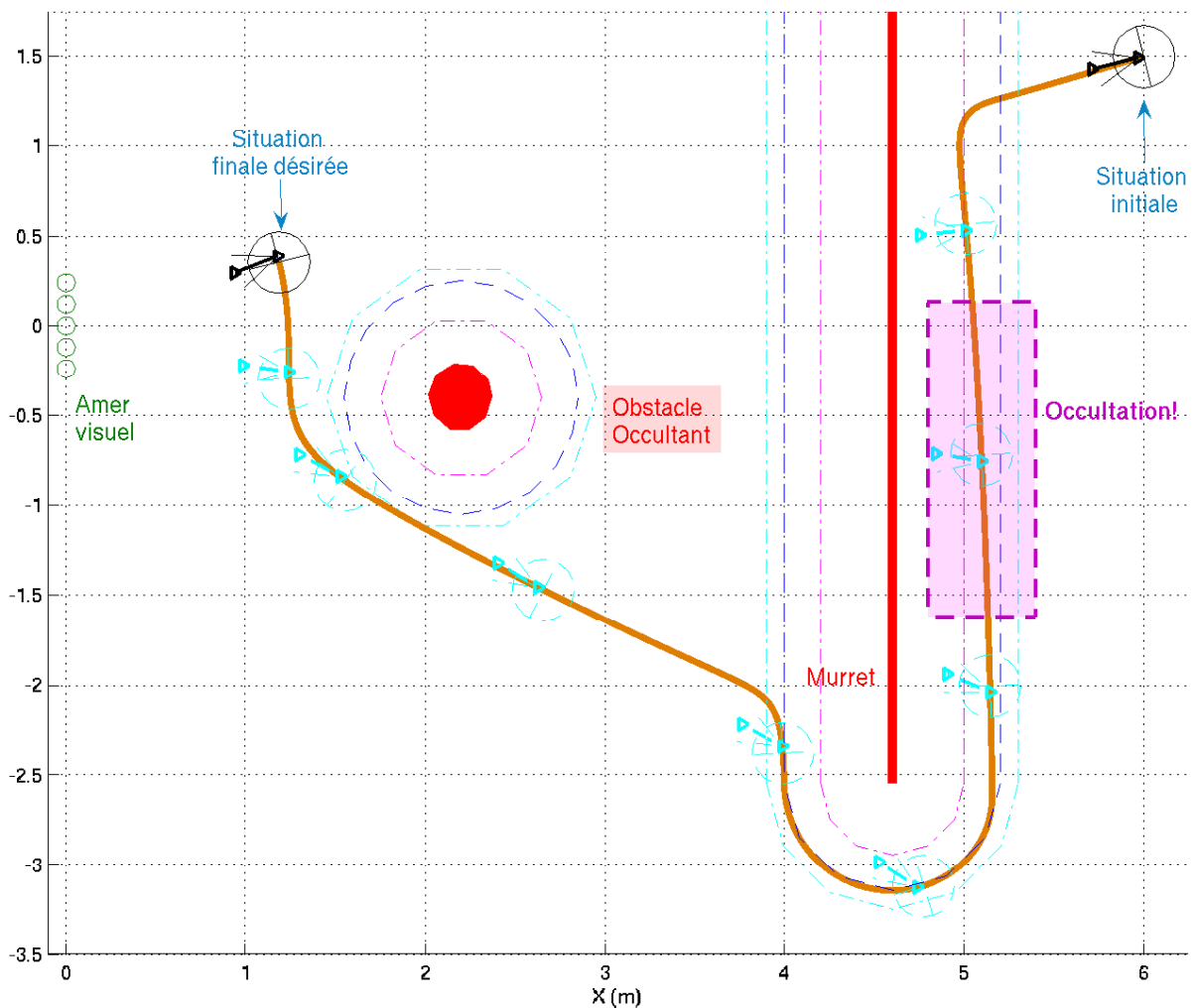


FIG. IV.20 – Trajectoire du robot.

Comme représenté sur les figures IV.20 IV.21 et IV.22 au début de la tâche il n'y a pas de risque d'occultation ($\mu_{occ} = 0$) ni de collision ($\mu_{coll} = 0$). Le robot n'est alors dirigé que

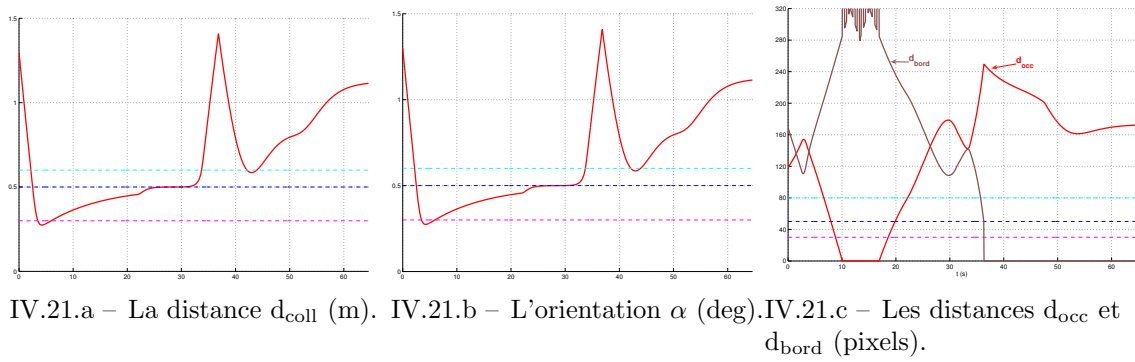


FIG. IV.21 – Mesures d'évitement.

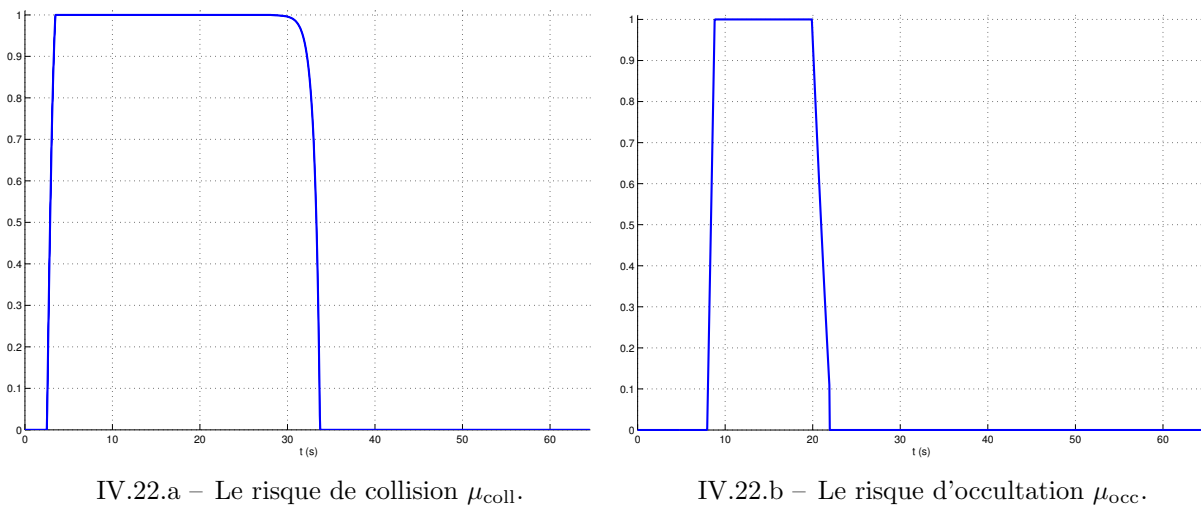
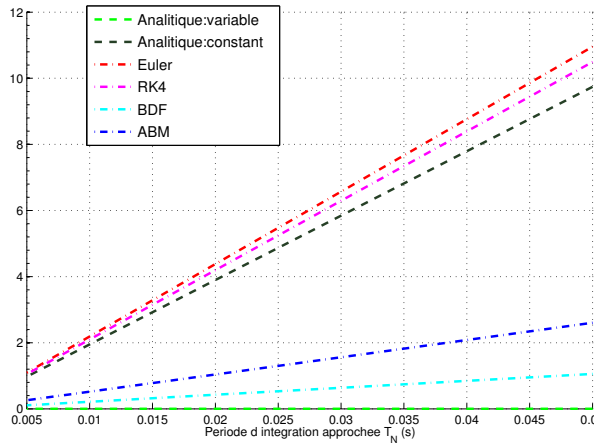


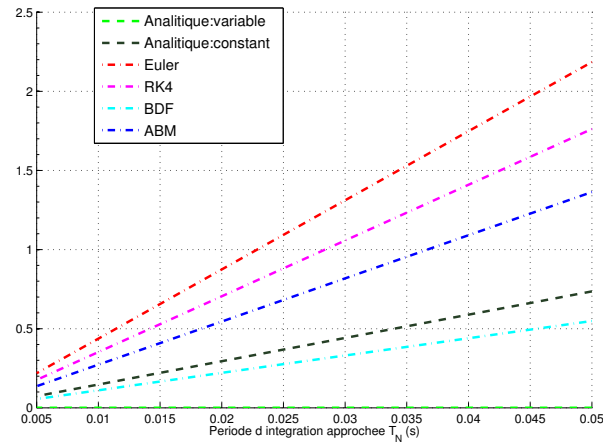
FIG. IV.22 – Les différents risques.

par l'asservissement visuel \dot{q}_{AV} et commence à converger vers \mathbf{s}^* . Puis le véhicule entre dans le voisinage du muret, le risque de collision μ_{coll} augmente pour atteindre sa valeur maximale 1, et la commande globale \dot{q} bascule vers \dot{q}_{COLL} . Le robot évite alors le muret en se rapprochant de l'enveloppe de contournement ξ_0 . Pendant cette phase l'obstacle cylindrique induit un risque d'occultation, et μ_{occ} augmente à son tour. La commande \dot{q}_{COLL} doit gérer l'occultation du motif visuel, en basculant la commande de la platine ϖ_{COLL} vers $\widehat{\varpi}_{coll}$ (IV.7) qui exploite l'estimation du signal visuel $\widehat{\mathbf{s}}$. Une fois l'obstacle occultant dépassé, le risque d'occultation μ_{occ} décroît, et ϖ_{COLL} bascule à nouveau vers ϖ_{coll} (I.35). Puis, une fois que le muret est lui aussi dépassé, le risque de collision μ_{coll} diminue vers 0. Le robot est piloté à nouveau par \dot{q}_{AV} jusqu'à ce qu'il ait fini de converger vers la situation désirée \mathbf{s}^* . La mission est alors parfaitement réalisée

Comparons maintenant l'efficacité des différentes techniques de reconstruction développées. La figure IV.23 présente la valeur maximale de $\|\mathbf{s} - \widehat{\mathbf{s}}\|$ et de $\|\mathbf{z} - \widehat{\mathbf{z}}\|$, en fonction du pas d'intégration T_N , obtenues pour nos algorithmes d'estimation (analytiques et numériques). Tout comme précédemment, l'approche reposant sur la résolution analytique



IV.23.a – Erreur maximale $\|\mathbf{s} - \hat{\mathbf{s}}\|$ (pixel) sur le motif visuel en fonction de T_N



IV.23.b – Erreur maximale $\|\mathbf{z} - \hat{\mathbf{z}}\|$ (mm) sur la profondeur en fonction de T_N .

FIG. IV.23 – Erreurs d'estimation, avec $T_{\text{ech}} = 50$ ms.

du système dynamique (III.17) avec le torseur $\mathcal{T}_{C/\mathcal{R}_0}$ variable est la plus précise. C'est encore la méthode de Gear qui est la plus performante, du fait de la variation plus importante dans la dynamique du système. L'algorithme d'Adams donne quant à lui de bons résultats, tandis que les schémas à pas simple d'Euler et de Runge–Kutta sont les moins efficaces. Enfin, comme précédemment, les erreurs restent ici relativement faibles (de l'ordre de quelques pixels pour le motif visuel et du millimètre pour la profondeur), et elles diminuent avec T_N . Les résultats obtenus démontrent donc la validité de nos algorithmes malgré la présence de perturbations imprévues sur la caméra liées au contournement des obstacles.

IV.2.4 Bilan

Nous avons donc validé nos différentes méthodes de reconstruction dans un contexte délicat, nécessitant de gérer à la fois les problèmes de collision et d'occultation. Nous avons également montré l'intérêt d'insérer ces algorithmes au sein de nos lois de commande référencées multi-capteurs, ce qui nous a conduit à étendre les stratégies précédemment développées. Il devient ainsi possible de réaliser des tâches de navigation plus complexes, au sens où leur succès est lié à la capacité du robot à tolérer la perte temporaire du motif visuel d'intérêt. Les deux missions présentées plus haut n'auraient en effet pas pu être effectuées à l'aide des stratégies proposées dans le chapitre II.

IV.3 Discussion

Nous avons cherché ici à valider les algorithmes de reconstruction développés dans le chapitre III. Nous avons ainsi considéré différents exemples de tâches de navigation référencées vision dans lesquelles le motif visuel \mathbf{s} pouvait être momentanément **entièrement** perdu. Tout d'abord, nous nous sommes intéressés à la réalisation d'un asservissement

visuel canonique pendant lequel la perte est due à une panne de la caméra. Ce premier exemple applicatif nous a permis de mettre en lumière l'intérêt d'utiliser ces algorithmes pour reconstruire le motif visuel et continuer à exécuter la mission, malgré la perte de ce dernier. Nous avons également montré leur pertinence pour différentes primitives visuelles correspondant aux trois cas mis en exergue dans le paragraphe III.2.4. Nous avons ainsi pu valider les trois types d'algorithmes de reconstruction élaborés au chapitre précédent. Sur cette base, nous avons de plus proposé d'exploiter ces résultats pour prédire les valeurs des indices visuels entre deux acquisitions d'image et ainsi rafraîchir la loi de commande à une fréquence plus élevée. De cette manière, les performances de l'asservissement peuvent être significativement améliorées. Enfin, nous sommes revenus vers la problématique qui a été au cœur de nos préoccupations : la réalisation de tâches référencées vision en environnement encombré. Nous avons alors étendu les stratégies de commande développées au chapitre II de manière à y intégrer les algorithmes d'estimation développés. Là encore, les résultats obtenus ont montré la validité de ces derniers dans un contexte plus exigeant.

Pour chaque application considérée, nous avons confronté les différentes techniques de reconstruction que nous avons élaborées. Nous nous proposons maintenant d'effectuer un bilan des résultats obtenus et de dresser une table comparative mettant en évidence leurs avantages et leurs inconvénients. Nous tenterons également de donner quelques éléments permettant de sélectionner la méthode la mieux appropriée en fonction du contexte (type d'indices visuels considérés, tâche à réaliser, etc.). Ainsi, tous les tests effectués ont montré que la résolution analytique du système dynamique (III.17) avec torseur $\mathcal{T}_{C/\mathcal{R}_0}$ variable (cf. paragraphe III.3.2.2) est de loin la plus précise. Ce résultat est tout à fait cohérent puisqu'en intégrant l'évolution de $\mathcal{T}_{C/\mathcal{R}_0}$ sur $[t_k; t_{k+1}]$ dans le processus d'estimation, cette méthode prend en compte la structure mécanique du robot porteur et apparaît donc comme la plus proche de la réalité. De fait, elle conduit à des erreurs négligeables, liées uniquement aux imprécisions des calculs numériques ou aux approximations effectuées (c'est notamment le cas pour les moments). De même, la résolution analytique avec le torseur $\mathcal{T}_{C/\mathcal{R}_0}$ constant par morceaux offre de bons résultats, quoique moins précise que la précédente. Cependant, ces deux approches ont été développées spécifiquement dans le cas du point, et ne peuvent donc être utilisées pour d'autres primitives visuelles. Afin de s'affranchir de ces restrictions, il est nécessaire de considérer les techniques de résolution numérique. Parmi celles-ci, la méthode de Gear (BDF) apparaît comme la plus performante, notamment dans le cas où la dynamique du système évolue rapidement (on se rapproche alors d'un problème dit *stiff*). Elle est donc particulièrement recommandée dans le cas de tâches de navigation nécessitant des fonctionnalités d'évitement d'obstacles, comme nous l'avons constaté dans la section précédente. La méthode à pas multiple d'Adams (ABM) semble également intéressante, quoique parfois limitée lorsque la reconstruction des indices visuels d'intérêt se fait en plusieurs étapes, cumulant ainsi les erreurs. Enfin, les méthodes à pas simple (Euler et Runge-Kutta) restent les moins précises du fait qu'elles ne prennent en compte aucun historique des indices visuels à estimer. Le tableau IV.2 synthétise les comparaisons de nos différents algorithmes.

Algorithmes	Avantages	Inconvénients	Intérêts
Analytique $\mathcal{T}_{C/\mathcal{R}_0}$ variable	Erreur quasi-nulle.	Dédié à notre plate-forme robotique et aux points.	++
Analytique $\mathcal{T}_{C/\mathcal{R}_0}$ constant	Permet de traiter différents systèmes robotiques portant la caméra.	Dédié aux primitives définies par des points.	+
Schéma d'Euler	Le plus simple de mise en œuvre.	Peu précis.	–
Schéma de Runge–Kutta RK4	Plus précis que le schéma d'Euler.	– Pas de prise en compte de l'historique. – Nécessite de nombreuses évaluations.	–
Schéma d'Adams ABM	– Approche prédiction/correction. – Possibilité d'estimer l'erreur de troncature locale.	– L'initialisation nécessite \mathbf{D} valeurs antérieures de χ obtenues aux instants $t_n \dots t_{n-\mathbf{D}}$. – Limitées dans les cas où l'on effectue des approximations successives du fait que χ est approximé sur la base de l'interpolation de $\dot{\chi} = \varphi(\chi, t)$.	+
Méthode de Gear BDF	La plus efficace lors de variations importantes sur le mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$.	L'initialisation nécessite \mathbf{D} valeurs antérieures de χ obtenues aux instants $t_n \dots t_{n-\mathbf{D}}$.	+

TAB. IV.2 – Comparaison des algorithmes développés.

Dans le cadre de nos travaux, nous nous sommes volontairement restreints à considérer principalement des schémas numériques classiques, allant jusqu'à l'ordre 4. D'une manière générale, les ordres supérieurs contribuent peu à l'amélioration des estimations numériques. Il est donc généralement préférable de chercher à réduire le pas d'intégration T_N . Cependant, lorsque le processus de reconstruction nécessite plusieurs phases d'estimation successives, notre approche perd en efficacité du fait du cumul des approximations effectuées (*i.e.* l'erreur ne diminue plus avec T_N). Toutefois, nous pouvons espérer perfectionner nos algorithmes en prenant en compte le lien existant entre les primitives “de haut niveau” à estimer et les primitives de “bas niveau” reconstruites pour déduire les précédentes. Enfin, notons que le moyen le plus efficace pour améliorer la qualité de nos estimations reste de diminuer la période d'échantillonnage de l'asservissement T_{ech} . Cela est d'autant plus envisageable que notre approche permet de fournir une estimation du motif visuel entre deux acquisitions d'images. La loi de commande peut donc être rafraîchie à une cadence plus élevée, et ses performances en sont significativement améliorées.

Par ailleurs, nos algorithmes nécessitent de bien connaître les différentes périodes d'échantillonnage (*i.e.* les périodes T_{ech} et T_N), et de disposer des données nécessaires à l'initialisation de nos méthodes. Ces conditions ne sont, néanmoins, pas toujours remplies

sur les diverses plates-formes expérimentales, conduisant à des performances moindres de nos techniques. En effet, nous nous sommes placés dans le cas où les données ne sont pas perturbées (*cf.* remarque III.8). Une solution pour limiter les effets des perturbation bruits serait de les modéliser et d'exploiter ce modèle, par exemple, à travers un filtre de Kalman [KALMAN, 1960] [MAYBECK, 1979]. Enfin, simples en termes de mise en œuvre, et facilement intégrables au sein de la boucle de commande, nos algorithmes permettent de disposer d'une bonne estimation du motif visuel, sans exploiter d'informations provenant de la caméra.

En conclusion, rappelons que l'ensemble des tests effectués ont montré que les erreurs d'estimations restent relativement peu importantes, démontrant l'efficacité de nos algorithmes. Grâce à nos méthodes, dès lors qu'une expression analytique de la matrice d'interaction associée à tout motif visuel \mathbf{s} peut être déterminée, il est possible de reconstruire le signal perceptuel lorsque la caméra n'est plus en mesure de le fournir.

Conclusion

DANS les travaux développés au cours de cette thèse, nous nous sommes intéressés à la gestion de la perte des données perceptuelles pendant la réalisation d'une tâche référencée vision en environnement encombré. Nous avons ainsi cherché à définir un ensemble de schémas de commande intégrant toutes les informations sensorielles pertinentes pour la tâche robotique. L'aspect réactif vis-à-vis d'événements extérieurs perturbant le déroulement de la tâche a été un aspect fondamental au cœur de nos préoccupations. À ce titre, nous avons, en premier lieu, rappelé, dans le premier chapitre, les notions de essentielles sur lesquelles s'appuient nos stratégies de navigation référencées vision en environnement encombré. Sur cette base, nous avons proposé une première contribution pour la gestion conjointe des collisions et des occultations durant l'exécution de la tâche référencée vision. Ainsi, nous avons développé, tout d'abord, une loi de commande permettant d'éviter les occultations tout en minimisant le risque de collision. Cependant, cette tâche peut amener le robot à passer dans un voisinage très proche des obstacles et sa sécurité ne peut être pleinement garantie. Nous avons alors synthétisé un second schéma de commande où l'évitement des occultations et des collisions est découplé afin de préserver l'intégrité du système robotique. Néanmoins, bien que les différents schémas de commande soient bien conditionnés, l'antagonisme des différentes tâches (asservissement visuel, contournement d'obstacle et évitement d'occultation) peut conduire à des problèmes de minima locaux. De plus, l'ajout de l'évitement des phénomènes d'occultation sur-contraint la tâche robotique, réduisant les trajectoires permettant de rejoindre le motif de référence. Nous avons alors cherché à étendre les stratégies élaborées de manière à pouvoir exécuter des tâches nécessitant la perte totale du motif visuel pour être réalisé convenablement. Pour cela, dans un premier temps, nous avons développé des algorithmes capables de reconstruire le signal visuel lorsque celui-ci n'est

plus disponible. Les techniques proposées s'appuient sur la connaissance du mouvement de la caméra $\mathcal{T}_{C/\mathcal{R}_0}$ et des mesures perceptuelles obtenues avant leur perte. Nous avons alors montré que le problème pouvait se résumer à la résolution d'un système d'équations différentielles, et nous avons proposé différentes méthodes de résolution analytique et numérique. Les premières offrent des solutions précises, restreintes à un contexte applicatif donné. Les secondes conduisent à des solutions approchées mais présentent l'avantage d'être exploitables dans le cas le plus général. Il devient ainsi possible d'obtenir une estimation de tout type de primitives visuelles, et ceci quelle que soit la structure du système robotique portant la caméra.

Enfin, dans le chapitre IV, nous avons démontré la validité de nos algorithmes dans le cadre de diverses tâches de navigation référencées vision au cours desquelles le motif visuel pouvait être perdu. Tout d'abord, nous avons considéré le cas de la réalisation d'un asservissement visuel canonique en environnement libre. Nous avons ainsi mis en lumière l'efficacité de nos méthodes lorsqu'une panne de la caméra se produit au cours de l'exécution. Nous avons ainsi montré tout son intérêt pour améliorer les performances de l'asservissement considéré. Nous sommes ensuite revenus vers notre problématique initiale : la réalisation de tâches de navigation référencée vision dans des environnements fortement contraints. Nous avons alors étendu les stratégies de commande développées dans le chapitre II, de manière à ce que les occultations totales du motif visuel d'intérêt puissent être tolérées si nécessaire. Pour cela, nous avons intégré les différents algorithmes développés dans ces stratégies, et montré la validité de notre approche en exécutant deux tâches de navigation non réalisables par les techniques proposées dans le chapitre II. Enfin nous avons également comparé les différents algorithmes élaborés, et donné quelques éléments permettant de sélectionner les méthodes les mieux adaptées en fonction du contexte applicatif.

Les travaux proposés ont permis d'ouvrir un certain nombre de perspectives intéressantes. Divers axes sont envisageables.

- ▷ En premier lieu, la plupart des résultats présentés ont été obtenus en simulation. Il serait donc souhaitable de compléter les validations proposées par des expérimentations menées d'abord sur notre Super Scout, puis sur d'autres plateformes robotiques telles que le manipulateur mobile jido par exemple.
- ▷ Ensuite, nous nous sommes volontairement restreints à des enchaînements relativement simples, réalisés par combinaison convexe des correcteurs. Or, ces techniques sont parfois difficiles à utiliser. En effet, d'une part, elles nécessitent un réglage souvent minutieux des paramètres gérant le séquençement, et, d'autre part, la phase de transition peut générer des minima locaux dus à la fusion des correcteurs. Une première extension de ces travaux serait donc de considérer des basculements plus évolués tels que les enchaînements dynamiques proposés par [SOUÈRES et CADENAT, 2003], ou encore les piles de tâches développées par [MANSARD et CHAUMETTE, 2004, 2005A] [MANSARD, 2006]. Ces techniques permettraient de s'affranchir des inconvénients mentionnés plus haut et donc d'améliorer la transition entre les tâches.
- ▷ De plus, en ce qui concerne la reconstruction des indices visuels, les solutions analytiques proposées sont restreintes dans le cas du point, et pour l'une d'entre elles, spécifique à la plate-forme robotique considérée. Bien qu'il soit a priori difficile

de proposer une solution pour le cas général, une extension naturelle serait de tenter de résoudre ce problème pour d'autres architectures robotiques et d'autres primitives visuelles.

- ▷ En outre, les exemples applicatifs ont été restreints au cadre de notre étude et donc appliqués à la réalisation de tâches de navigation référencées vision en environnement encombré. Il serait également intéressant d'exploiter nos résultats dans des domaines connexes. Ainsi, par exemple, nos approches peuvent contribuer à améliorer l'efficacité des processus de traitement d'images en aidant à détecter les défauts et à les corriger.
 - ▷ Par ailleurs, les solutions analytiques obtenues dans le cas du point fournissent directement une expression de la profondeur. À ce titre, elles peuvent donc également présenter un intérêt pour toute approche nécessitant la caractérisation de cette dernière. Elles peuvent ainsi être utilisées en complément des algorithmes de tracking, de reconstruction de pose de la caméra, etc.
 - ▷ Enfin, les techniques proposées ont été développées dans le cas d'un amer visuel fixe. Il serait également pertinent de s'affranchir de cette hypothèse et de traiter le problème d'un amer mobile en intégrant son mouvement.
-

Annexe

Annexe A

Description du formalisme du suivi de chemin

Le formalisme du suivi de chemin consiste à synthétiser une loi de commande assurant la convergence géométrique du robot vers un chemin de référence défini a priori. Plusieurs approches et implémentations ont été proposées, citons par exemple [SAMPEI *et al.*, 1991, 1995] [SAMSON, 1992] [FREZZA *et al.*, 1998], etc. Dans le cadre de nos travaux, nous avons plus particulièrement considéré l'approche développée par C. SAMSON [SAMSON, 1992].

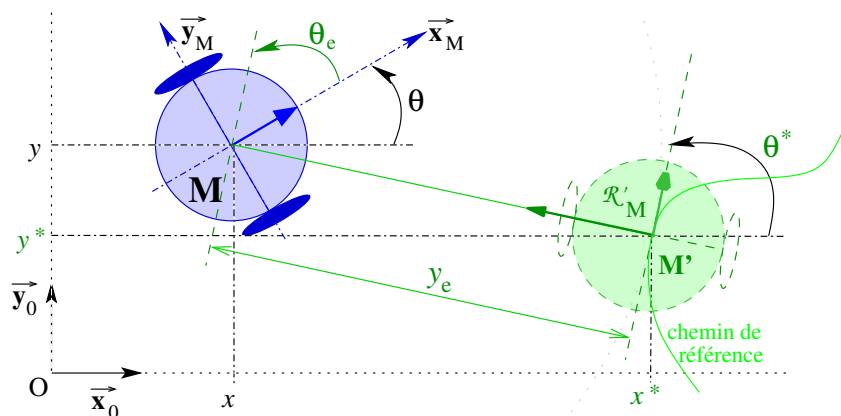


FIG. A.1 – Description du formalisme du suivi de chemin.

Le formalisme de suivi de chemin proposé dans [SAMSON, 1992] consiste à projeter orthogonalement le centre M du robot sur le chemin de référence en un point M' , de manière à définir un repère de Frenet \mathcal{R}'_M , comme représenté sur la figure A.1. La projection permet alors de définir un robot fictif que le véhicule réel doit rejoindre. Soit $Q = (x, y, \theta)^T$ un vecteur décrivant la posture d'un robot mobile, où le couple (x, y) représente les coordonnées du centre M de la base mobile, et θ l'orientation du véhicule dans le repère de la scène \mathcal{R}_0 ; de la même manière la configuration du robot de référence

est notée $Q^* = (x^*, y^*, \theta^*)^T$. L'erreur de configuration $Q_e = (x_e, y_e, \theta_e)^T$ entre les deux robots s'exprime alors dans le repère \mathcal{R}'_M de la manière suivante :

$$Q_e^{\mathcal{R}'_M} = \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} = \mathbf{P}_{\mathcal{R}'_M/\mathcal{R}_0} Q_e^{\mathcal{R}_0} = \mathbf{P}_{\mathcal{R}'_M/\mathcal{R}_0} \left(\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} - \begin{pmatrix} x^* \\ y^* \\ \theta^* \end{pmatrix} \right)$$

où $\mathbf{P}_{\mathcal{R}'_M/\mathcal{R}_0}$ représente la matrice de passage entre les repères \mathcal{R}_0 et \mathcal{R}'_M , et est donné par :

$$\mathbf{P}_{\mathcal{R}'_M/\mathcal{R}_0} = \begin{pmatrix} \cos \theta^* & \sin \theta^* & 0 \\ -\sin \theta^* & \cos \theta^* & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

La dynamique de l'erreur de configuration s'écrit alors comme suit :

$$\begin{pmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{pmatrix} = \begin{pmatrix} v^*(\chi^* y_e - 1) + v \cos \theta_e \\ -v^* \chi^* x_e + v \sin \theta_e \\ v\chi - v^* \chi^* \end{pmatrix} \quad (\text{A.1})$$

où $\chi^* = \omega^*/v^*$ et $\chi = \omega/v$ définissent respectivement les courbures du chemin de référence et du chemin suivi par le robot réel. v^* et ω^* représentent les vitesses linéaire et angulaire du véhicule de référence. De plus, comme ce dernier est défini par la projection orthogonale du robot réel sur le chemin de référence, x_e et \dot{x}_e restent toujours nulles. La vitesse de référence peut se déduire de (A.1) de la manière suivante :

$$v^* = \frac{v \cos \theta_e}{1 - \chi^* y_e}, \quad \text{sous la contrainte : } \left| \frac{1}{\chi^*} \right| > |y_e|$$

Cette contrainte implique que la distance entre le robot et le chemin doit toujours rester inférieure à son rayon de courbure, pour que l'unicité de la projection du point M soit garantie [SAMSON, 1992]. La dynamique de l'erreur se déduit alors de (A.1) par :

$$\begin{cases} \dot{y}_e = v \sin \theta_e \\ \dot{\theta}_e = v\chi_e \end{cases} \quad \text{où } \chi_e = \chi - \frac{\chi^* \cos \theta_e}{1 - \chi^* y_e} \quad (\text{A.2})$$

Le problème de suivi de chemin consiste à synthétiser une loi de commande en χ_e permettant d'amener le couple (y_e, θ_e) en $(0, 0)$, sous l'hypothèse que la vitesse linéaire v du robot est bornée, à dérivée bornée et *ne s'annule pas*. Cette dernière hypothèse permet de garantir la commandabilité du système (A.2) [CANUDAS DE WIT *et al.*, 1993]. Il existe différentes lois de commande non-linéaires permettant de répondre à ce problème, citons par exemple [SAMPEI *et al.*, 1991], [SAMSON, 1992], [CANUDAS DE WIT *et al.*, 1993], [HAMEL, 1996], [CADENAT, 1999], etc.

Annexe B

Résolution numérique d'équations différentielles

Nous nous intéressons ici à la résolution numérique d'un système d'équations différentielles ordinaires avec conditions initiales, de la forme :

$$\begin{cases} \dot{\boldsymbol{\chi}}(t) &= \varphi(\boldsymbol{\chi}, t) \\ \boldsymbol{\chi}_0 &= \boldsymbol{\chi}(t_0) \end{cases}$$

où $\boldsymbol{\chi}_0$ représente les conditions initiales du système dynamique, et $\varphi(\boldsymbol{\chi}, t)$ est une fonction continue qui satisfait la condition de LIPSCHITZ pour la variable $\boldsymbol{\chi}$ (cf. théorème B.2). En effet, dans ce cas, le théorème B.1 de CAUCHY–LIPSCHITZ garantit l'existence d'une unique solution locale au système d'équations différentielles considéré.

Théorème B.1 Cauchy–Lipschitz

Soit φ une fonction :

$$\begin{aligned} \varphi &: \mathbf{U} \times \mathbf{I} \rightarrow \mathbb{R} \\ (\boldsymbol{\chi}, t) &\mapsto \varphi(\boldsymbol{\chi}, t) \end{aligned}$$

décrivant l'équation différentielle : $\dot{\boldsymbol{\chi}} = \varphi(\boldsymbol{\chi}, t)$, soumise à la condition initiale $\boldsymbol{\chi}_k$ à $t_k \in \mathbf{I}$. Si la fonction $\varphi(\boldsymbol{\chi}, t)$ satisfait la condition de Lipschitz, alors il existe **une et une seule solution** $\boldsymbol{\chi}(t)$ de l'équation différentielle définie pour tout $t \in \mathbf{I}$.

Théorème B.2 Condition de Lipschitz

Soit φ une fonction :

$$\begin{aligned} \varphi &: \mathbf{U} \times \mathbf{I} \rightarrow \mathbb{R} \\ (\boldsymbol{\chi}, t) &\mapsto \varphi(\boldsymbol{\chi}, t) \end{aligned}$$

où $\mathbf{U} \subset \mathbb{R}$ et $\mathbf{I} \subset \mathbb{R}$ sont des intervalles de \mathbb{R} . La fonction $\varphi(\boldsymbol{\chi}, t)$ satisfait la condition Lipschitz, si et seulement si il existe une constante $L > 0$, telle que :

$$\exists L > 0 / \forall t \in \mathbf{I}, \forall (\boldsymbol{\chi}_a, \boldsymbol{\chi}_b) \in \mathbf{U}^2, \quad \|\varphi(\boldsymbol{\chi}_a, t) - \varphi(\boldsymbol{\chi}_b, t)\| \leq L \|\boldsymbol{\chi}_a - \boldsymbol{\chi}_b\| \quad (\text{B.1})$$

La plus petite valeur L est appelée la *constante de Lipschitz*.

Il existe dans la littérature de nombreuses méthodes permettant de résoudre numériquement ce type problème [SHAMPINE et GORDON, 1975], [PRESS *et al.*, 1992], [DEMAILLY, 1996]. . . Classiquement, les méthodes d'intégration numérique se divisent en deux catégories :

- les schémas à pas simple (ou unique) où l'état du système à l'instant t_{n+1} est construit à partir des seules informations de l'instant t_n :

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + \Phi(\boldsymbol{\chi}_n, t_n)$$

- les méthodes à pas multiple où l'état du système à l'instant t_{n+1} est construit à partir des N états antérieurs obtenus aux instants : $t_n, t_{n-1}, \dots, t_{n-N}$:

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + \Phi(\boldsymbol{\chi}_n, \boldsymbol{\chi}_{n-1}, \dots, \boldsymbol{\chi}_{n-N})$$

Il est également possible de distinguer les méthodes :

- *explicites* où $\widehat{\boldsymbol{\chi}}_{n+1}$ s'exprime uniquement à partir de terme connus à l'instant t_n .
- *implicites* où $\widehat{\boldsymbol{\chi}}_{n+1}$ s'exprime en fonction de termes dépendant de $\boldsymbol{\chi}_{n+1}$. Il faut alors disposer d'une première évaluation $\widetilde{\boldsymbol{\chi}}_{n+1}$ de $\boldsymbol{\chi}_{n+1}$ que l'on obtient au moyen d'une phase de prédiction. Ce type de schéma fonctionne alors généralement sur le principe prédiction/correction qui présente l'avantage d'offrir de bonnes propriétés de stabilité.

Définition B.1. Notons qu'une méthode numérique qui a une erreur globale en ordre $\mathcal{O}(T_N^r)$ avec $r > 0$ est dite méthode d'ordre r .

Nous présentons ci-après les intégrateurs numériques auxquels nous nous sommes plus particulièrement intéressés, à savoir les schémas d'Euler, de Runge–Kutta, d'Adams et à la méthode de Gear [GEAR, 1971] (ou BDF).

B.1 Schéma d'Euler

La méthode d'intégration la plus simple est le schéma d'Euler proposé par Leonhard Euler (1707-1783) [EULER, 1768]. La solution $\boldsymbol{\chi}_{n+1}$ est alors approximée par :

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + T_N \cdot \varphi(\boldsymbol{\chi}_n, t_n) \tag{B.2}$$

Cette formulation est aussi appelé schéma explicite à un pas, en opposition à la forme implicite suivante :

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + T_N \cdot \varphi(\widetilde{\boldsymbol{\chi}}_{n+1}, t_{n+1}) \tag{B.3}$$

La forme implicite du schéma d'Euler requiert une première estimation de $\widetilde{\boldsymbol{\chi}}_{n+1}$ qui peut, par exemple, être fournie par un schéma d'Euler explicite.

Bien qu'étant très simple de mise en œuvre, les schémas d'Euler sont des méthodes d'intégration d'ordre 1 (*cf.* remarque B.1), et souffrent souvent d'un manque de précision pour la résolution des systèmes d'équations différentielles.

Remarque B.1: Dans la plupart des méthodes d'intégration numérique on retrouve le schéma d'Euler à l'ordre 1.

B.2 Méthode de Runge–Kutta

Les méthodes de Runge–Kutta forment une famille importante d'approximation de solutions pour la résolution d'équations différentielles. Ces techniques furent initialement développées au début du XX^{ème} siècle par les mathématiciens allemands Carl Runge (1856-1927) et Martin Wilhelm Kutta (1867-1944). L'idée est que la valeur $\widehat{\boldsymbol{\chi}}_{n+1}$ est approximée par la somme de la valeur actuelle $\boldsymbol{\chi}_n$ et du produit de la taille de l'intervalle d'intégration T_N par la pente estimée. La forme générique des méthodes de Runge–Kutta est donnée par :

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + T_N \cdot \sum_{i=1}^{\mathbf{E}} b_i \varphi(\boldsymbol{\chi}_n(t_n + c_i T_N), t_n + c_i T_N)$$

Elle peut se ré-écrire sous la forme :

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + T_N \cdot \sum_{i=1}^{\mathbf{E}} b_i K_i \quad \text{avec :} \quad \begin{cases} K_1 = T_N \cdot \varphi(\boldsymbol{\chi}_n, t_n) \\ K_2 = T_N \cdot \varphi(\boldsymbol{\chi}_n + a_{21} K_1, t_n + c_2 T_N) \\ K_3 = T_N \cdot \varphi(\boldsymbol{\chi}_n + a_{31} K_1 + a_{32} K_2, t_n + c_3 T_N) \\ \vdots \\ K_i = T_N \cdot \varphi(\boldsymbol{\chi}_n + \sum_j^{\mathbf{E}} a_{i,j} K_j, t_n + c_i T_N) \end{cases} \quad (\text{B.4})$$

où \mathbf{E} désigne le nombre d'étapes du schéma a_{ji} (pour $1 \leq j \leq i \leq \mathbf{E}$), b_i et c_i (pour $1 \leq i \leq \mathbf{E}$) sont des coefficients définis de telle sorte que :

$$\sum_{i=1}^{\mathbf{E}} b_i = 1 \quad \text{et} \quad \sum_{i=1}^{\mathbf{E}} a_{ji} = c_j$$

La forme la plus classique est la méthode de Runge–Kutta d'ordre 4 (dénnotée RK4) qui fut construite autour des années 1980 par Dormand et Prince [DORMAND et PRINCE, 1980] :

$$\widehat{\boldsymbol{\chi}}_{n+1} = \boldsymbol{\chi}_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad \text{avec :} \quad \begin{cases} K_1 = T_N \cdot \varphi(\boldsymbol{\chi}_n, t_n) \\ K_2 = T_N \cdot \varphi(\boldsymbol{\chi}_n + \frac{K_1}{2}, t_n + \frac{T_N}{2}) \\ K_3 = T_N \cdot \varphi(\boldsymbol{\chi}_n + \frac{K_2}{2}, t_n + \frac{T_N}{2}) \\ K_4 = T_N \cdot \varphi(\boldsymbol{\chi}_n + K_3, t_n + T_N) \end{cases} \quad (\text{B.5})$$

La méthode RK4 est une méthode d'ordre 4, ce qui signifie que l'erreur commise à chaque étape est de l'ordre de $\mathcal{O}(T_N^5)$, alors que l'erreur totale accumulée est de l'ordre de $\mathcal{O}(T_N^4)$.

B.3 Méthode d'Adams

Déjà longtemps avant la parution des premières méthodes de Runge–Kutta, John Couch Adams (1819-1892) avait résolu numériquement des équations différentielles

[BASHFORTH, 1883]. Son idée était d'approximer $\varphi(\chi, t)$ par un polynôme $\mathcal{P}(t)$ de degré $\mathbf{D} - 1$ (cf. figure B.1) qui satisfait :

$$\mathcal{P}(t_j) = \varphi(\chi_j, t_j) \quad \text{pour } j = n, n-1, \dots, n-\mathbf{D}+1$$

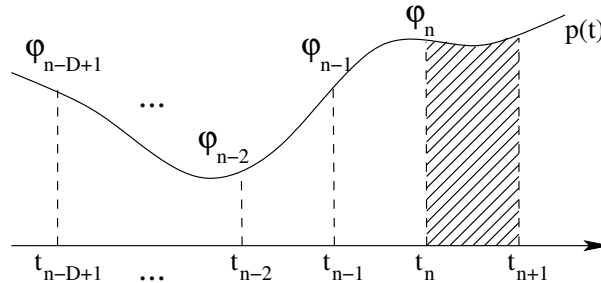


FIG. B.1 – Principe de la méthode d'Adams.

L'approximation de $\widehat{\chi}_{n+1}$ est alors obtenue à partir de la relation suivante :

$$\widehat{\chi}_{n+1} - \chi_n = \int_{t_n}^{t_{n+1}} \mathcal{P}(\tau) d\tau \quad (\text{B.6})$$

Les méthodes d'Adams sont ainsi des schémas d'intégration temporelle à pas multiples.

B.3.1 Schéma d'Adams–Bashforth

Le développement de Taylor de l'équation (B.6) permet d'exprimer $\widehat{\chi}_{n+1}$ en fonction de χ_n et de ses dérivées successives. Or, d'après l'équation différentielle (que l'on cherche à résoudre), $\dot{\chi}_n = \varphi_n$, $\ddot{\chi}_n = \dot{\varphi}_n \dots$, ce qui mène à une expression donnant $\widehat{\chi}_{n+1}$ en fonction de χ_n , φ_n et ses dérivées successives $\dot{\varphi}_n$, $\ddot{\varphi}_n \dots$. Ces dernières sont évaluées à l'aide du polynôme d'interpolation $p(t)$ (et de ses dérivées) basé sur les valeurs φ_n , $\varphi_{n-1} \dots$. Ainsi la méthode d'Adams–Bashforth s'écrit alors comme suit :

$$\widehat{\chi}_{n+1} = \chi_n + T_N \sum_{i=0}^{\mathbf{E}} A_{\mathbf{B}i} \cdot \varphi(\chi_{n-i}, t_{n-i}) + \text{LTE}^{\text{Ab}}, \quad \text{pour } 0 \leq \mathbf{E} \leq k \quad (\text{B.7})$$

où $A_{\mathbf{B}i}$ est donné à partir du polynôme de Lagrange comme suit :

$$A_{\mathbf{B}i} = \frac{(-1)^i}{i!(\mathbf{E}-i)!} \int_0^1 \prod_{j=0, j \neq i}^{\mathbf{E}} (u+j) du, \quad \text{pour } i = 0, \dots, \mathbf{E} \quad (\text{B.8})$$

L'erreur de troncation locale pour le schéma d'Adams-Bashforth (LTE^{Ab}) au pas t_n peut s'écrire de la façon suivante :

$$\begin{aligned} \text{LTE}^{\text{Ab}} &= \chi(t_{n+1}) - \widehat{\chi}_{n+1} = \chi^{(\mathbf{E}+1)}(\tau) T_N^{\mathbf{E}+1} \cdot \frac{1}{(\mathbf{E}-1)!} \int_0^1 \prod_{i=1}^{\mathbf{E}-1} (u+i) du \quad (\text{B.9}) \\ \text{LTE}^{\text{Ab}} &= \chi^{(\mathbf{E}+1)}(\tau) T_N^{\mathbf{E}+1} \cdot \beta_{\mathbf{E}}, \quad \text{pour } \tau \in [t_n, t_{n+1}] \end{aligned}$$

où $\beta_{\mathbf{E}}$ est appelé constante d'erreur de la méthode d'Adams–Bashforth. On montre qu'un schéma d'Adams–Bashforth comportant $\mathbf{E} + 1$ étapes (B.7) est une méthode d'ordre $r = \mathbf{E} + 1$.

B.3.2 Schéma d'Adams–Moulton

On procède de même que pour le schéma d'Adams–Bashforth, mais à partir du développement de Taylor *arrière* de (B.6) permettant d'exprimer χ_n en fonction de $\widehat{\chi}_{n+1}$, $\widehat{\chi}_{n+1}$, $\widehat{\chi}_{n+1}$, $\widehat{\chi}_{n+1}$, $\widehat{\chi}_{n+1}$. On obtient ainsi la relation suivante :

$$\widehat{\chi}_{n+1} = \chi_n + T_N \sum_{i=-1}^{\mathbf{E}-1} A_{M_i} \cdot \varphi(\chi_{n-i}, t_{n-i}) + \text{LTE}^{\text{AM}}, \quad \text{pour } 0 \leq \mathbf{E} \leq k \quad (\text{B.10})$$

où A_{M_i} est donné à partir du polynôme de Lagrange de la façon suivante :

$$A_{M_i} = \frac{(-1)^{i+1}}{(i+1)!(n-i-1)!} \int_0^1 \prod_{j=-1, j \neq i}^{\mathbf{E}-1} (u+j) du, \quad \text{pour } i = -1, 0, \dots, \mathbf{E}-1 \quad (\text{B.11})$$

L'erreur de troncature locale pour le schéma d'Adams–Moulton (LTE^{AM}) à l'étape t_n peut s'écrire comme suit :

$$\begin{aligned} \text{LTE}^{\text{AM}} &= \chi(t_{n+1}) - \widehat{\chi}_{n+1} = \chi^{(\mathbf{E}+1)}(\eta) T_N^{\mathbf{E}+1} \cdot \frac{1}{\mathbf{E}!} \int_0^1 \prod_{i=0}^{\mathbf{E}-1} (u+i) du \\ \text{LTE}^{\text{AM}} &= \chi^{(\mathbf{E}+1)}(\eta) T_N^{\mathbf{E}+1} \cdot \mu_{\mathbf{E}} \quad \text{pour } \eta \in [t_n, t_{n+1}] \end{aligned} \quad (\text{B.12})$$

où $\mu_{\mathbf{E}}$ est appelé constante d'erreur de la méthode d'Adams–Moulton. On montre qu'un schéma d'Adams–Moulton comportant $\mathbf{E}+1$ étapes (B.10) est une méthode d'ordre $r = \mathbf{E}$.

B.3.3 Méthode prédicteur/correcteur

La méthode Adams–Moulton est un schéma *implicite*, c'est-à-dire qui nécessite la connaissance de $\widehat{\chi}_{n+1}$. C'est pourquoi on l'utilise souvent en relation avec le schéma d'Adams–Bashforth comme paire prédicteur/correcteur. La méthode Adams–Bashforth–Moulton (ABM) consiste donc à *prédire* la valeur $\widetilde{\chi}_{n+1}$ au moyen d'un schéma d'Adams–Bashforth (B.7), puis à *corriger* cette valeur grâce au schéma d'Adams–Moulton (B.10) pour déterminer $\widehat{\chi}_{n+1}$. L'algorithme est alors le suivant :

P : calcul de la prédiction $\widetilde{\chi}_{n+1}$ au moyen du schéma d'Adams–Bashforth (B.7)

E : évaluation de la fonction : $\widetilde{\varphi}_{n+1} = \varphi(\widetilde{\chi}_{n+1}, t_{n+1})$

C : obtention de $\widehat{\chi}_{n+1}$ par correction de $\widetilde{\chi}_{n+1}$ à l'aide du schéma d'Adams–Moulton (B.10).

E : évaluation de : $\widehat{\varphi}_{n+1} = \varphi(\widehat{\chi}_{n+1}, t_{n+1})$

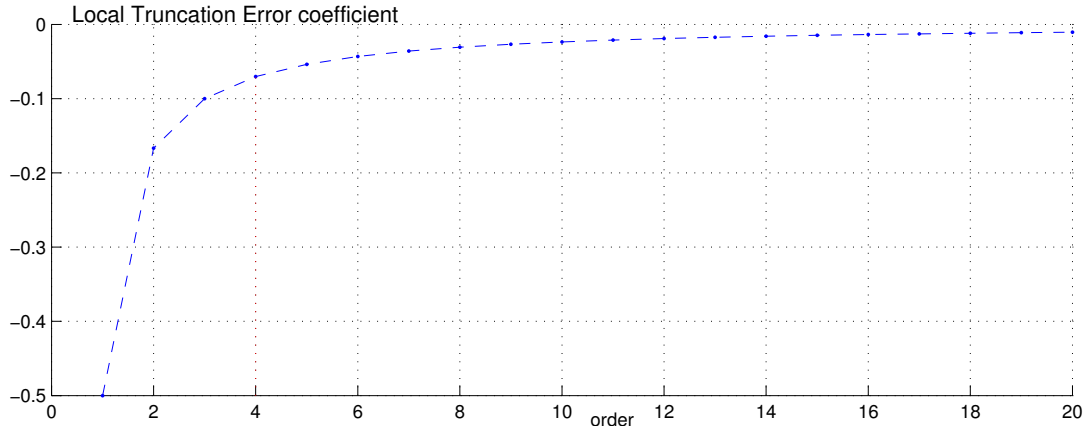
Cette procédure, que l'on dénote PECE, est la plus utilisée.

Si le prédicteur et le correcteur sont tous deux du même ordre, en supposant que le pas h est suffisamment petit, et que $\chi^{(\mathbf{E}+1)}(q, t)$ est quasi constant entre deux intervalles de temps, l'erreur de troncature locale peut être approximée par la relation suivante :

$$\text{LTE} = \chi(t_{n+1}) - \widehat{\chi}_{n+1} \approx (\widehat{\chi}_{n+1} - \widetilde{\chi}_{n+1}) \cdot \left(\frac{\mu_{\mathbf{E}}}{\beta_{\mathbf{E}} - \mu_{\mathbf{E}}} \right) \quad (\text{B.13})$$

Grâce à cette relation (B.13) on peut déterminer efficacement l'ordre le plus adéquat pour l'intégrateur numérique. En effet, la figure B.2 présente l'évolution du coefficient $\mathcal{C}_{\mathbf{E}} = \mu_{\mathbf{E}}/(\beta_{\mathbf{E}} - \mu_{\mathbf{E}})$, et on peut constater qu'au-delà de l'ordre 4 à 6 le gain en précision par rapport à l'erreur de troncature locale n'est plus conséquent.

Le tableau B.1 présente l'application de la méthode d'Adams–Bashforth–Moulton (ABM) jusqu'à l'ordre 5.

FIG. B.2 – Evolution de du coefficient $\mathcal{C}_{\mathbf{E}}$

	Prédiction & Correction	$\mathcal{C}_{\mathbf{E}}$
ordre 1 $\mathbf{E} = 0$	$\tilde{\chi}_{n+1} = \chi_n + h\varphi_n$ $\hat{\chi}_{n+1} = \chi_n + h\varphi_{n+1}$	-0.5
ordre 2 $\mathbf{E} = 1$	$\tilde{\chi}_{n+1} = \chi_n + \frac{T_N}{2}(3\varphi_n - \varphi_{n-1})$ $\hat{\chi}_{n+1} = \chi_n + \frac{T_N}{2}(\varphi_{n+1} + \varphi_n)$	$-\frac{2}{12}$
ordre 3 $\mathbf{E} = 2$	$\tilde{\chi}_{n+1} = \chi_n + \frac{T_N}{12}(23\varphi_n - 16\varphi_{n-1} + 5\varphi_{n-2})$ $\hat{\chi}_{n+1} = \chi_n + \frac{T_N}{12}(5\varphi_{n+1} + 8\varphi_n - \varphi_{n-1})$	$-\frac{1}{10}$
ordre 4 $\mathbf{E} = 3$	$\tilde{\chi}_{n+1} = \chi_n + \frac{T_N}{24}(55\varphi_n - 59\varphi_{n-1} + 37\varphi_{n-2} - 9\varphi_{n-3})$ $\hat{\chi}_{n+1} = \chi_n + \frac{T_N}{24}(9\varphi_{n+1} + 19\varphi_n - 5\varphi_{n-1} + \varphi_{n-2})$	$-\frac{19}{270}$
ordre 5 $\mathbf{E} = 4$	$\tilde{\chi}_{n+1} = \chi_n + \frac{T_N}{720}(1901\varphi_n - 2774\varphi_{n-1} + 2616\varphi_{n-2} - 1274\varphi_{n-3} + 251\varphi_{n-4})$ $\hat{\chi}_{n+1} = \chi_n + \frac{T_N}{720}(251\varphi_{n+1} + 646\varphi_n - 264\varphi_{n-1} + 106\varphi_{n-2} - 19\varphi_{n-3})$	$-\frac{27}{502}$

TAB. B.1 – Les cinq premiers ordres du schéma Adams-Bashforth-Moulton

B.4 Méthode BDF (*Backward Differentiation Formulas*)

La méthode de Gear [GEAR, 1971], aussi connue sous le nom de *Backward Differentiation Formulas* (BDF), consiste à utiliser un polynôme passant par les χ_i pour approximer les solutions du système d'équations différentielles. Considérons ainsi le polynôme $\mathcal{P}(t)$ de degré \mathbf{E} défini par :

$$\mathcal{P}(t_j) = \chi_j, \quad \text{pour } j = k+1, k, k-1, \dots, k-\mathbf{E}+1$$

qui interpole $\varphi(\chi)$ en $t_{n+1}, t_n, \dots, t_{n-\mathbf{E}+1}$. On cherche ainsi $\chi(t_{n+1})$ qui vérifie :

$$\dot{\mathcal{P}}(t_{n+1}) = \varphi(\mathcal{P}(t_{n+1}), t_{n+1}) = \varphi(\hat{\chi}_{n+1}, t_{n+1})$$

En remplaçant le polynôme $\dot{\mathcal{P}}(t)$, on obtient :

$$\hat{\chi}_{n+1} = \sum_{i=0}^{\mathbf{E}-1} \alpha_i \chi_{n-i} + h\beta_0 \varphi(\hat{\chi}_{n+1}, t_{n+1}) \quad (\text{B.14})$$

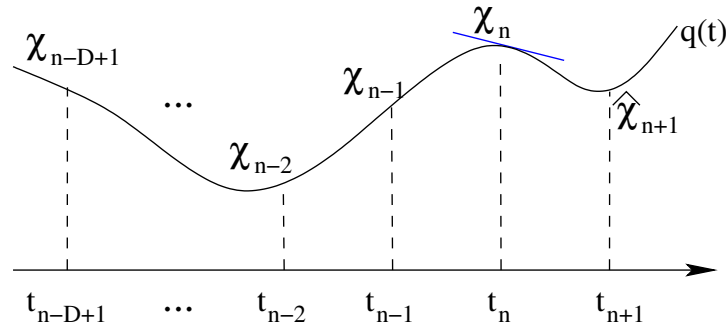


FIG. B.3 – Méthode BDF.

où β_0 est donnée par :

$$\beta_0 = \frac{1}{\sum_{j=1}^{\mathbf{E}} j^{-1}}$$

Les coefficients α_i sont donnés dans le tableau B.2 pour $1 \leq \mathbf{E} \leq 6$.

Remarque B.2: On montre que pour $1 \leq \mathbf{E} \leq 6$ les méthodes BDF sont zéro-stables (cf. définition B.2), et que pour $\mathbf{E} > 6$ les méthodes BDF deviennent toutes instables.

Définition B.2 (Satisfaction des conditions des racines). Une méthode à pas multiple est dite zéro-stable si aucune des racines du polynôme caractéristique a son module supérieur à 1, et si chaque racine de module 1 est simple.

$\mathbf{E} = 1$	$\hat{\chi}_{n+1} = \chi_n + T_N \varphi(\tilde{\chi}_{n+1}, t_{n+1})$
$\mathbf{E} = 2$	$\hat{\chi}_{n+1} = \frac{4}{3}\chi_n - \frac{1}{3}\chi_{n-1} + \frac{2}{3}T_N \varphi(\tilde{\chi}_{n+1}, t_{n+1})$
$\mathbf{E} = 3$	$\hat{\chi}_{n+1} = \frac{18}{11}\chi_n - \frac{9}{11}\chi_{n-1} + \frac{2}{11}\chi_{n-2} + \frac{6}{11}T_N \varphi(\tilde{\chi}_{n+1}, t_{n+1})$
$\mathbf{E} = 4$	$\hat{\chi}_{n+1} = \frac{48}{25}\chi_n - \frac{36}{25}\chi_{n-1} + \frac{16}{25}\chi_{n-2} - \frac{3}{25}\chi_{n-3} + \frac{12}{25}T_N \varphi(\tilde{\chi}_{n+1}, t_{n+1})$
$\mathbf{E} = 5$	$\hat{\chi}_{n+1} = \frac{300}{137}\chi_n - \frac{300}{137}\chi_{n-1} + \frac{200}{137}\chi_{n-2} - \frac{75}{137}\chi_{n-3} + \frac{12}{137}\chi_{n-4} + \frac{60}{137}T_N \varphi(\tilde{\chi}_{n+1}, t_{n+1})$
$\mathbf{E} = 6$	$\hat{\chi}_{n+1} = \frac{360}{147}\chi_n - \frac{450}{147}\chi_{n-1} + \frac{400}{147}\chi_{n-2} - \frac{225}{147}\chi_{n-3} + \frac{72}{147}\chi_{n-4} - \frac{10}{147}\chi_{n-5} + \frac{60}{147}T_N \varphi(\tilde{\chi}_{n+1}, t_{n+1})$

TAB. B.2 – Les six premiers ordres de la méthode BDF

Ces méthodes sont essentiellement utilisées pour résoudre des équations différentielles dont le comportement varie fortement. Dans ce cas certains algorithmes d'intégration peuvent devenir numériquement instable, à moins de choisir un pas d'intégration T_N extrêmement faible. On parle alors de problème "raide" (*stiff problem*).

Définitions

configuration *Espace des configurations* : Pour situer un système mécanique dans l'espace, il suffit de connaître un certain nombre de paramètres caractéristiques du système. L'espace de ces paramètres constitue alors l'espace des configurations du système.

holonome Les systèmes non holonomes sont caractérisés par des contraintes cinématiques prenant la forme d'équations portant sur les dérivées des paramètres de configurations : le système est non holonome lorsque ces équations différentielles ne sont pas intégrables. Cela se traduit par le fait qu'un tel système n'est pas capable de se déplacer *instantanément* dans toutes les directions.

homographie L'homographie (notée H) est une application projective bijective. Il s'agit d'une transformation linéaire entre deux plans projectifs, qui permet, entre autres, de lier des points entre différentes images.

IEEE1394 Norme d'une interface série aussi connue sous le nom de FireWire, ou également appelée interface iLink. Il s'agit d'un bus rapide (400 Mb/s pour la première version, et jusqu'à 800 Mb/s pour la seconde) véhiculant à la fois des données et des signaux de commandes des différents appareils qu'il relie.

Super Scout Robot de type char développé par la société *Nomadic Technologies*. Le robot est mis à notre disposition par l'AIP-PRIMECA de Toulouse.

situation La situation d'un corps (notée \mathbf{r}) décrit la position et l'orientation de celui-ci par rapport à un repère choisi.

Notations

- M^T Désigne la transposée de la matrice M
- D_x La longueur de l'entraxe entre le centre M du robot et le centre P de la platine.
- l Abscisse curviligne décrite par le centre M du robot.
- (x, y, z) Coordonnées 3D d'un point p dans l'environnement.
- (X, Y) Coordonnées métriques d'un point P dans l'image.
- (U, V) Coordonnées pixel d'un point P dans l'image.
- θ Orientation de la base mobile.
- ϑ Orientation de la platine par rapport à l'axe \vec{x}_M du robot.
- q Vecteur des paramètres décrivant la configuration du robot : $q = (l \ \theta \ \vartheta)^T$.
- \dot{q} Vecteur de commande, défini par les vitesses du robot : $\dot{q} = (v \ \omega \ \varpi)^T$.
- v Vitesse linéaire de la base mobile.
- ω Vitesse angulaire de la base mobile.
- ϖ Vitesse angulaire de la platine.
- r Vecteur décrivant la situation d'un système mécanique, définissant sa position et son orientation.
- s Les signaux capteurs entrant dans la synthèse d'une commande référencée capteurs. Dans le cadre de l'asservissement visuel on parle aussi d'information ou de motif visuel.
- s^* Information visuelle de référence.
- \mathcal{T} Désigne un torseur cinématique, qui se définit classiquement pour un point p par rapport au repère \mathcal{R} par les vitesses de translation $V_{p/\mathcal{R}}$ et de rotation $\Omega_{\mathcal{R}_p/\mathcal{R}}$ de la manière suivante :
- $$\mathcal{T}_{p/\mathcal{R}} = \begin{pmatrix} V_{p/\mathcal{R}} \\ \Omega_{\mathcal{R}_p/\mathcal{R}} \end{pmatrix}$$
- \mathcal{T}_r Désigne le torseur cinématique *réduit* de la caméra, qui ne contient que les degrés de liberté commandables.
- J_r Matrice jacobienne *réduite* reliant les mouvements de la caméra au vecteur de commande \dot{q} (voir aussi \mathcal{T}_r).
-

- f Distance focale de la caméra.
- \mathcal{L} Matrice d'interaction ou jacobienne de l'image, avec $\mathcal{L} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}}$.
- \mathcal{L}_r Matrice d'interaction *réduite* de la caméra.
- \mathcal{C} Matrice de combinaison.
- e_{AV} Fonction de tâche référencée vision.
- α Orientation entre la tangente à l'obstacle et la direction $\vec{\mathbf{x}}_M$ du véhicule.
- d_{coll} Distance entre le centre M du robot et le point le plus proche de l'obstacle.
- σ Définit le sens de contournement d'un obstacle.
- d_{bord} Distance entre l'objet occultant et le bord de l'image.
- d_{occ} Distance entre un objet occultant et le signal perceptuel \mathbf{s} .
- \mathcal{O}_{cc} Un objet occultant présent sur le plan image.
- $(Y_{\mathcal{O}_{\text{cc}}}^+, Y_{\mathcal{O}_{\text{cc}}}^-)$ Les ordonnées des bords droit et gauche de l'objet occultant.
- $(Y_{\text{min}}, Y_{\text{max}})$ Les ordonnées minimale et maximale du plan image.
- Y_{occ} L'ordonnée du bord de l'objet occultant le plus proche des informations visuelles.
- Y_s L'ordonnée du motif visuel le plus proche de l'objet occultant.
- $\hat{\mathbf{s}}$ Une estimation du motif visuel $\mathbf{s}(q, t)$.
- \hat{z} Une estimation de la profondeur $z(q, t)$ du motif visuel $\mathbf{s}(q, t)$.
- $\varphi = \dot{\chi}$.. Le système d'équations différentielles à résoudre.
- χ Vecteur d'état du système d'équations différentielles à résoudre.

Acronymes

ABM .. Adams–Bashforth–Moulton

La méthode Adams–Bashforth–Moulton (ABM) consiste à *prédire* la valeur $\tilde{\chi}_{n+1}$ au moyen d'un schéma d'Adams–Bashforth (B.7), puis à *corriger* cette valeur pour déterminer $\hat{\chi}_{n+1}$ grâce au schéma d'Adams–Moulton (B.10).

AIP-PRIMECA Atelier Inter-établissement de Productique – Pôle de Ressources Informatiques pour la MECAnique

Le réseau national AIP-PRIMECA est constitué de pôles régionaux qui fédèrent les ressources technologiques et les compétences de différents sites au niveau d'une région. Ces pôles régionaux AIP-PRIMECA, inter-établissements, associent les universités et les écoles d'ingénieurs concernées par les activités pédagogiques dans tous les domaines de la conception intégrée en mécanique et de la productique.

WEB: <http://www.aip-primeca.net>

BDF ... *Backward Differentiation Formulas*

La méthode de Gear [GEAR, 1971] aussi connue sous le nom de *Backward Differentiation Formulas* (BDF).

CNRS .. Centre National de la Recherche Scientifique

Le CNRS est un organisme public de recherche fondamentale, à caractère scientifique et technologique, placé sous la tutelle du Ministre chargé de la Recherche. Il produit du savoir et met ce savoir au service de la société.

WEB: <http://www.cnrs.fr>

FSE Fonds Social Européen

Le Fonds Social Européen constitue la principale source de financement pour les actions destinées à développer la capacité d'insertion professionnelle et les ressources humaines. Il aide les États membres à lutter contre le chômage, à prévenir la sortie du marché du travail et à encourager la formation afin que la main-d'oeuvre et les entreprises européennes soient mieux armées face aux nouveaux défis mondiaux.

WEB: <http://europa.eu.int/fse>

GEPETTO GEPETTO

Le groupe GEPETTO est un groupe de recherche du LAAS. L'objectif de l'opération Gepetto créée en 2006, à partir du groupe RIA, est de contribuer à la compréhension et à la modélisation du mouvement pour des systèmes anthropomorphes, qu'ils soient artificiels (les robots humanoïdes), virtuels (les mannequins numériques), ou naturels (l'homme).

WEB: <http://www.laas.fr/laas/1-5792-Groupe-GEPETTO.php>

INRIA . Institut National de Recherche en Informatique et Automatique

L'INRIA placé sous la double tutelle des ministères de la recherche et de l'industrie, a pour vocation d'entreprendre des recherches fondamentales et appliquées dans les domaines des STIC (Sciences et Technologies de l'Information et de la Communication).

WEB: <http://www.inria.fr> .

INSA .. Institut National des Sciences Appliquées

L'INSA est une Grande Ecole d'Ingénieurs, sous tutelle du Ministère de la Jeunesse, l'Education Nationale et de la Recherche. Constitués en réseau, les INSA sont au nombre de 5, localisés dans de grandes métropoles régionales, Lyon, Rennes, Rouen, Strasbourg et Toulouse.

WEB: <http://www.insa-france.fr>

IRISA .. Institut de Recherche en Informatique et Systèmes Aléatoires

L'IRISA est un pôle de recherche public regroupant une trentaine d'équipes de recherche. L'INRIA, le CNRS, l'Université de Rennes 1 et l'INSA de Rennes sont les partenaires de cette unité mixte de recherche.

WEB: <http://www.irisa.fr>

LAAS .. Laboratoire d'Analyse et d'Architecture des Systèmes

Le LAAS est une unité propre de recherche du CNRS, rattaché au département scientifique ST2I (Sciences et Technologies de l'Information et de l'Ingénierie), il est associé à trois établissements d'enseignement supérieur de Toulouse : l'Université Paul Sabatier (UPS), l'Institut National des Sciences Appliquées (INSA) et l'Institut National Polytechnique (INP).

WEB: <http://www.laas.fr>

RAP ... Robotique Action et Perception

Le groupe RAP est un groupe de recherche du LAAS. Ce groupe de recherche étudie essentiellement la couche fonctionnelle des systèmes robotiques d'intérieur ou d'extérieur équipés généralement de plusieurs capteurs (vision, télémétrie), ou robot humanoïde, équipé de capteurs visuels et sonores.

WEB: <http://www.laas.fr/laas/1-5945-Groupe-RAP.php>

RIA Robotique et Intelligence Artificielle

Le groupe RIA est un ancien groupe de recherche du LAAS, qui fut scindé en trois groupes : GEPETTO, RAP et RIS. Il définit à présent l'un des quatre pôles de recherche du LAAS.

Le pôle RIA recouvre les thématiques de recherche du laboratoire relatives aux systèmes autonomes, du micro-système au grand système intégré ou distribué. Cette problématique est très large. Le pôle a ainsi vocation à une transdisciplinarité permettant de développer des travaux en synergie entre plusieurs groupes du laboratoire, chacun apportant son propre éclairage.

RIS Robotique et Interactions

Le groupe RIS est un groupe de recherche du LAAS.

RK4 ... Méthode de Runge-Kutta d'ordre 4

La forme, la plus classique, d'ordre 4 des méthodes de la famille de Runge-Kutta.

SfM *Structure from Motion*

En vision pour ordinateur, le *Structure from Motion* se rapporte au processus d'établissement d'un modèle 3D à partir d'une séquence vidéo d'un objet mobile rigide.

STIC .. Sciences et Technologies de l'Information et de la Communication

Les STIC sont un domaine d'application de l'informatique et des statistiques.

**U.P.S. .. Université Paul SABATIER
L'université de Toulouse III.**

WEB: <http://www.ups-tlse.fr>

Bibliographie

- [AZARBAYEJANI et PENTLAND, 1995] **AZARBAYEJANI, A.** et **PENTLAND, A.** “*Recursive estimation of motion, structure, and focal length*”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, n^o 6, pp. 562–575. Juin 1995. ISSN 0162-8828.
DOI: 10.1109/34.387503.
- [BASHFORTH, 1883] **BASHFORTH, F.** “*An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid, with an explanation of the method of integration employed in constructing the tables which give the theoretical forms of such drops, by J.C. Adams*”. Cambridge. 1883.
- [BELLOT et DANÈS, 2001A] **BELLOT, D.** et **DANÈS, P.** “*Handling visual servoing schemes through rational systems and LMIs*”. Dans *IEEE International Conference on Decision and Control (CDC’01)*. Orlando, USA, pp. 3601–3606. Décembre 2001.
- [BELLOT et DANÈS, 2001B] **BELLOT, D.** et **DANÈS, P.** “*Towards an LMI approach to multiobjective visual servoing*”. Dans *European Control Conference (ECC’01)*. Porto, Portugal, pp. 2930–2935. Septembre 2001.
- [BELLOT, 2002] **BELLOT, D.** “*Contribution à l’analyse et à la synthèse de schémas de commande référencés vision*”. Thèse de doctorat, Université Paul SABATIER, LAAS, Toulouse, France. Décembre 2002.
- [BENHIMANE et MALIS, 2003] **BENHIMANE, S.** et **MALIS, E.** “*Vision-based control with respect to planar and non-planar objects using a zooming camera*”. Dans *IEEE International Conference on Advanced Robotics (ICAR’03)*. Coimbra, Portugal, pp. 991–996. Juillet 2003.
- [BORENSTEIN et KOREN, 1991] **BORENSTEIN, J.** et **KOREN, Y.** “*The vector field histogram-fast obstacle avoidance for mobile robots*”. IEEE Transactions on Robotics and Automation, vol. 7, n^o 3, pp. 278–388. Juin 1991. DOI: 10.1109/70.88137.
- [BOUGUET, 2007] **BOUGUET, J.-Y.** “*Camera calibration toolbox for matlab*”. 2007.
Disponible sur: http://www.vision.caltech.edu/bouguetj/calib_doc/ (consulté le Septembre 2007).
- [BOUKIR, 1993] **BOUKIR, S.** “*Reconstruction 3D d’un environnement statique par vision active*”. Thèse de doctorat, Université de Rennes 1, IRISA. Octobre 1993.
- [BROCK et KHATIB, 1999] **BROCK, O.** et **KHATIB, O.** “*High-speed navigation using the global dynamic window approach*”. Dans *IEEE International Conference on Robotics and Automation (ICRA’99)*, vol. 1. Detroit, Michigan, USA, pp. 341–346. Mai 1999.
- [BROCKETT, 1983] **BROCKETT, R. W.** “*Asymptotic stability and feedback stabilization*”. Dans *Differential Geometric Control Theory*, édité par Brockett R., Millman R. et Sussman H., pp. 181–191. Birkhauser, Boston, USA. 1983.

- [CADENAT *et al.*, 1999A] CADENAT, V., SOUÈRES, P. et COURDESS, M. “A hybrid control for avoiding obstacles during a vision-based tracking task”. Dans *European Control Conference (ECC’99)*. Karlsruhe, Allemagne. Septembre 1999.
- [CADENAT *et al.*, 1999B] CADENAT, V., SOUÈRES, P., SWAIN, R. et DEVY, M. “A controller to perform a visually guided tracking task in a cluttered environment”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’99)*, vol. 2. Kyongju, Korea, pp. 775–780. Octobre 1999. DOI: 10.1109/IROS.1999.812774.
- [CADENAT, 1999] CADENAT, V. “Commande référencée multi-capteurs pour la navigation d’un robot mobile”. Thèse de doctorat, Université Paul SABATIER, LAAS. Décembre 1999.
- [CANUDAS DE WIT *et al.*, 1993] CANUDAS DE WIT, C., KHENNOUF, H., SAMSON, C. et SORDALEN, O. “Nonlinear control design for mobile robots”. Dans *Recent Trends in Mobile Robots*, édité par Zheng Y. F., vol. 11, pp. 121–156. Singapore : World Scientific Publishing. 1993.
- [CHAUMETTE, 1990] CHAUMETTE, F. “La relation vision-commande : théorie et application à des tâches robotiques”. Thèse de doctorat, Université de Rennes 1, IRISA. Juillet 1990.
- [CHAUMETTE, 1998A] CHAUMETTE, F. “De la perception à l’action : l’asservissement visuel ; de l’action à la perception : la vision active”. Habilitation à diriger des recherches, Université de Rennes 1, IRISA. Janvier 1998.
- [CHAUMETTE, 1998B] CHAUMETTE, F. “Potential problems of stability and convergence in image-based and position-based visual servoing”. Dans *The Confluence of Vision and Control*, édité par Kriegman D., Hager G. et Morse A., vol. 237, pp. 66–78. LNCIS Series, Springer-Verlag. 1998.
- [CHAUMETTE, 2002] CHAUMETTE, F. “Asservissement visuel”. Dans *La commande des robots manipulateurs*, édité par Khalil W., Traité IC2, chapitre 3, pp. 105–150. Hermès. 2002.
- [CHAUMETTE, 2004] CHAUMETTE, F. “Image moments : a general and useful set of features for visual servoing”. *IEEE Transactions on Robotics and Automation*, vol. 20, n° 4, pp. 713–723. Août 2004. DOI: 10.1109/TR0.2004.829463.
- [CHAUMETTE et HUTCHINSON, 2006] CHAUMETTE, F. et HUTCHINSON, S. “Visual servo control, part I : Basic approaches”. *IEEE Robotics and Automation Magazine*, vol. 13, n° 4, pp. 82–90. Décembre 2006. DOI: 10.1109/MRA.2006.250573.
- [CHAUMETTE et HUTCHINSON, 2007] CHAUMETTE, F. et HUTCHINSON, S. “Visual servo control, part II : Advanced approaches”. *IEEE Robotics and Automation Magazine*, vol. 14, n° 1, pp. 109–118. Mars 2007. DOI: 10.1109/MRA.2007.339609.
- [CHAUMETTE *et al.*, 1994] CHAUMETTE, F., BOUKIR, S., BOUTHEMY, P. et JUVIN, D. “Optimal estimation of 3d structures using visual servoing”. Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’94)*. Seattle, Washington, pp. 347–354. Juin 1994. DOI: 10.1109/CVPR.1994.323850.
- [CHAUMETTE *et al.*, 1996] CHAUMETTE, F., BOUKIR, S., BOUTHEMY, P. et JUVIN, D. “Structure from controlled motion”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, n° 5, pp. 492–504. Mai 1996. DOI: 10.1109/34.494639.
- [CHESI *et al.*, 2004] CHESI, G., HASHIMOTO, K., PRATTICIZZO, D. et VICINO, A. “Keeping features in the field of view in eye-in-hand visual servoing : a switching approach”. *IEEE Transactions on Robotics and Automation*, vol. 20, n° 5, pp. 908–914. Octobre 2004. DOI: 10.1109/TR0.2004.829456.
- [COMPORT *et al.*, 2004] COMPORT, A., MARCHAND, E. et CHAUMETTE, F. “Robust model-based tracking for robot vision”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’04)*, vol. 1. Sendai, Japan, pp. 692–697. Septembre 2004. DOI: 10.1109/IROS.2004.1389433.
- [CORKE, 1996] CORKE, P. “Visual control of robots : High performance visual servoing”. *Mechatronics Research Studies Press LTD*. 1996. ISBN 0 86380 207 9.

- [CORKE et HUTCHINSON, 2001] **CORKE, P.** et **HUTCHINSON, S.** “A new partitioned approach to image-based visual servo control”. *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 507–515. Août 2001. DOI: 10.1109/70.954764.
- [CRÉTUAL, 1998A] **CRÉTUAL, A.** “Image-based visual servoing by integration of dynamic measurements”. Dans *IEEE International Conference on Robotics and Automation (ICRA’98)*, vol. 3. Leuven, Belgique, pp. 1994–2001. Mai 1998.
- [CRÉTUAL, 1998B] **CRÉTUAL, A.** “Asservissement visuel à partir d’information de mouvement dans l’image”. Thèse de doctorat, Université de Rennes 1, IRISA. Novembre 1998.
- [CRISAN et DOUCET, 2002] **CRISAN, D.** et **DOUCET, A.** “A survey of convergence results on particle filtering methods for practitioners”. *IEEE Transactions on Signal Processing*, vol. 50, n° 3, pp. 736–746. Mars 2002. DOI: 10.1109/78.984773.
- [DE LUCA et ORIOLO, 1990] **DE LUCA, A.** et **ORIOLO, G.** “Kinematic resolution of redundancy via joint-space decomposition”. Dans *8th CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators (Ro.Man.Sy. ’90)*. Krakow, Pologne, pp. 64–91. Juillet 1990.
- [DE LUCA *et al.*, 1998] **DE LUCA, A.**, **ORIOLO, G.** et **SAMSON, C.** “Feedback control of a nonholonomic car-like robot”. Dans *Planning robot motion*, édité par Laumond J.-P. Springer-Verlag, Berlin, DE. 1998.
- [DEMAILLY, 1996] **DEMAILLY, J.-P.** “Analyse numérique et équations différentielles”. EDP Sciences, Grenoble, France, presses universitaires de grenoble édition. 1996. ISBN 2-7061-0715-4.
- [DFWVL500] DFWVL500. “SONY DFW-VL 500 and SONY DFW-V 500”. Sony, ver. 1.0 édition. 2001.
- [DHOME *et al.*, 1989] **DHOME, M.**, **RICHTIN, M.**, **LAPRESTE, J.** et **RIVES, G.** “Determination of the attitude of 3d objects from a single perspective view”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, n° 12, pp. 1265–1278. Décembre 1989.
- [DORMAND et PRINCE, 1980] **DORMAND, J.** et **PRINCE, P. J.** “A family of embedded Runge-Kutta formulae”. *Journal of Computational and Applied Mathematics*, vol. 6, pp. 19–26. 1980.
- [DOYLE *et al.*, 1989] **DOYLE, J.**, **GLOVER, K.**, **KHARGONEKAR, P.** et **FRANCIS, B.** “A state space solution to the standard \mathcal{H}_2 and \mathcal{H}_∞ control problems”. *IEEE Transactions on Automatic Control*, vol. 34, n° 8, pp. 831–847. 1989. DOI: 10.1109/9.29425.
- [DRUMMOND et CIPOLLA, 2002] **DRUMMOND, T.** et **CIPOLLA, R.** “Real-time visual tracking of complex structures”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n° 7, pp. 932–946. Juillet 2002. ISSN 0162-8828. DOI: 10.1109/TPAMI.2002.1017620.
- [ESPIAU *et al.*, 1992] **ESPIAU, B.**, **CHAUMETTE, F.** et **RIVES, P.** “A new approach to visual servoing in robotics”. *IEEE Transactions on Robotics and Automation*, vol. 8, n° 3, pp. 313–326. Juin 1992. DOI: 10.1109/70.143350.
- [EULER, 1768] **EULER, L.** “*Institutionum calculi integralis volumen primum in quo methodus integrandi a primis principiis usque ad integrationem aequationum differentialium primi gradus pertractatur*”. *Opera omnia*, vol. 11. 1768.
- [FANG *et al.*, 2002] **FANG, Y.**, **DAWSON, D.**, **DIXON, W.** et **DE QUEIROZ, M.** “Homography-based visual servoing of wheeled mobile robots”. Dans *IEEE International Conference on Decision and Control (CDC’02)*, vol. 3. pp. 2866–2871. Décembre 2002.
- [FANG *et al.*, 2005] **FANG, Y.**, **DIXON, W.**, **DAWSON, D.** et **CHAWDA, P.** “Homography-based visual servo regulation of mobile robots”. *IEEE Transactions on Systems, Man, and Cybernetics—Part B : Cybernetics*, vol. 3, n° 5, pp. 1041–1050. Octobre 2005. DOI: 10.1109/TSMCB.2005.850155.
- [FAUGERAS, 1993] **FAUGERAS, O.** “*Three-Dimensional Computer Vision : A Geometric Viewpoint*”. MIT Press, Cambridge. 1993.
- [FAVARO et SOATTO, 2003] **FAVARO, P.** et **SOATTO, S.** “Seeing beyond occlusions (and other marvels of a finite lens aperture)”. Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. St. Louis, USA, pp. II : 579–586. Juin 2003.

- [FEDDEMA et MITCHELL, 1989] **FEDDEMA, J.** et **MITCHELL, O.** “*Vision-guided servoing with feature-based trajectory generation*”. IEEE Transactions on Robotics and Automation, vol. 5, n° 5, pp. 691–700. Octobre 1989. DOI: 10.1109/70.88086.
- [FLEURY, 1996] **FLEURY, S.** “*Architecture de contrôle distribuée pour robot mobile autonome : principes, conception et applications*”. Thèse de doctorat, Université Paul SABATIER, LAAS, Toulouse, France. Février 1996.
- [FLEURY et HERRB, 2001] **FLEURY, S.** et **HERRB, M.** “*G^{en}M : User Manual*”. LAAS. Mai 2001.
Disponible sur: <https://softs.laas.fr/openrobots/wiki/genom> (consulté le Septembre 2007) .
- [FLEURY *et al.*, 1997] **FLEURY, S.**, **HERRB, M.** et **CHATILA, R.** “*G^{en}M : a tool for the specification and the implementation of operating modules in a distributed robot architecture*”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97)*, vol. 2. Grenoble, France, pp. 842–849. Septembre 1997. DOI: 10.1109/IROS.1997.655108.
- [FOX *et al.*, 1997] **FOX, D.**, **BURGARD, W.** et **THRUN, S.** “*The dynamic window approach to collision avoidance*”. IEEE Robotics and Automation Magazine, vol. 4, n° 1, pp. 23–33. Mars 1997.
- [FREZZA *et al.*, 1998] **FREZZA, R.**, **PICCI, G.** et **SOATTO, S.** “*A lagrangian formulation of nonholonomic path following*”. The Confluence of Vision and Control, pp. 118–133. 1998. ISSN 0170-8643.
- [GAO, 2006] **GAO, B.** “*Contribution à la synthèse de commandes référencées vision 2D multicritères*”. Thèse de doctorat, Université Paul SABATIER, LAAS, Toulouse, France. Octobre 2006.
- [GARCIA-ARACIL et MALIS, 2004] **GARCIA-ARACIL, N.** et **MALIS, E.** “*Preserving the continuity of visual servoing despite changing image features*”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, vol. 2. Sendai, Japan, pp. 1383–1388. Octobre 2004. DOI: 10.1109/IROS.2004.1389589.
- [GARCIA-ARACIL *et al.*, 2005] **GARCIA-ARACIL, N.**, **MALIS, E.**, **ARACIL-SANTONJA, R.** et **PEREZ-VIDAL, C.** “*Continuous visual servoing despite the changes of visibility in image features*”. IEEE Transactions on Robotics and Automation, vol. 21. Décembre 2005.
DOI: 10.1109/TR0.2005.855995.
- [GEAR, 1971] **GEAR, C. W.** “*Numerical initial value problems in ordinary differential equations*”. Prentice-Hall, Englewood Cliffs, NJ. 1971. ISBN 013-626606-1.
- [HAMEL, 1996] **HAMEL, T.** “*Sur la robustesse et la précision du contrôle des déplacements de robots mobile à roue*”. Thèse de doctorat, Université de Compiègne, Compiègne, France. Janvier 1996.
- [HARALICK *et al.*, 1989] **HARALICK, R.**, **JOO, H.**, **LEE, C.-N.**, **VAIDYA, V.** et **KIM, M.** “*Pose estimation from corresponding point data*”. IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, n° 6, pp. 1426–1446. Novembre 1989. DOI: 10.1109/21.44063.
- [HARRIS et STEPHENS, 1988] **HARRIS, C.** et **STEPHENS, M.** “*A combined corner and edge detector*”. Dans *Alvey Vision Conference*. Manchester, Royaume-Uni, pp. 147–151. Janvier 1988.
- [HILL et PARK, 1979] **HILL, J.** et **PARK, W. T.** “*Real time control of a robot with a mobile camera*”. Dans *9th International Symposium on Industrial Robot*. Washington, DC, pp. 233–246. Mars 1979.
- [HORAUD, 1987] **HORAUD, R.** “*New methods for matching 3d objects with single perspective views*”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, n° 3, pp. 401–412. Mai 1987.
- [HORAUD et DORNAIKA, 1995] **HORAUD, R.** et **DORNAIKA, F.** “*Hand-eye calibration*”. International Journal of Robotics Research, vol. 14, n° 3, pp. 195–210. Juin 1995.
Disponible sur: <http://perception.inrialpes.fr/Publications/1995/HD95>
(consulté le Septembre 2007).
- [HORAUD et MONGA, 1995] **HORAUD, R.** et **MONGA, O.** “*Vision par ordinateur, outils fondamentaux (deuxième édition)*”. Traité des nouvelles technologies, série Informatique. Hermès, Paris, France. 1995. ISBN 2-86601-481-2.

- [HORAUD *et al.*, 1989] **HORAUD, R.**, **CONIO, B.**, **LEBOULLEUX, O.** et **LACOLLE, B.** “An analytic solution for the perspective 4-point problem”. Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 47. pp. 33–44. Juin 1989.
- [HOSODA et ASADA, 1994] **HOSODA, K.** et **ASADA, M.** “Versatile visual servoing without knowledge of true Jacobian”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*. Munich, Allemagne, pp. 186–193. Septembre 1994.
- [HU, 1962] **HU, M.** “Visual pattern recognition by moment invariants”. *IRE Transaction on Information Theory*, vol. 8, n° 2, pp. 179–187. Février 1962.
- [HUTCHINSON *et al.*, 1996] **HUTCHINSON, S.**, **HAGER, G.** et **CORKE, P.** “A tutorial on visual servo control”. *IEEE Transactions on Robotics and Automation*, vol. 12, n° 5, pp. 651–670. Octobre 1996. DOI: 10.1109/70.538972.
- [JACKSON *et al.*, 2004] **JACKSON, J.**, **YEZZI, A.** et **SOATTO, S.** “Tracking deformable objects under severe occlusions”. Dans *IEEE International Conference on Decision and Control (CDC'04)*, vol. 3. Atlanta, USA, pp. 2990–2995. Décembre 2004.
- [JÄGERSAND *et al.*, 1997] **JÄGERSAND, M.**, **FUENTES, O.** et **NELSON, R.** “Experimental evaluation of uncalibrated visual servoing for precision manipulation”. Dans *IEEE International Conference on Robotics and Automation (ICRA'97)*, vol. 3. Albuquerque, New Mexico, USA, pp. 1874–2880. Avril 1997.
- [JERIAN et JAIN, 1991] **JERIAN, C.** et **JAIN, R.** “Structure from motion : A critical analysis of methods”. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, n° 3, pp. 572–588. Mai 1991. DOI: 10.1109/21.97478.
- [KALMAN, 1960] **KALMAN, R.** “A new approach to linear filtering and prediction problems”. *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, n° Series D, pp. 35–45. 1960.
- [KHADRAOUI *et al.*, 1998] **KHADRAOUI, D.**, **DEBAIN, C.**, **ROUVEURE, R.**, **MARTINET, P.**, **BONTON, P.** et **GALLICE, J.** “Vision based control in driving assistance of agricultural vehicules”. *International Journal of Robotics Research*, vol. 17, n° 10, pp. 1040–1054. 1998.
- [KHATIB *et al.*, 1997] **KHATIB, M.**, **JAOUNI, H.**, **CHATILA, R.** et **LAUMOND, J.-P.** “Dynamic path modification for car-like nonholonomic mobile robots”. Dans *IEEE International Conference on Robotics and Automation (ICRA'97)*. Albuquerque, New Mexico, USA, pp. 490–496. Avril 1997.
- [KHATIB, 1996] **KHATIB, M.** “Contrôle du mouvement d'un robot mobile par retour sensoriel”. Thèse de doctorat, LAAS. Décembre 1996.
- [KHATIB, 1980] **KHATIB, O.** “Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles”. Thèse de doctorat, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France. 1980.
- [KHATIB, 1986] **KHATIB, O.** “Real-time obstacle avoidance for manipulators and mobile robots”. *International Journal of Robotics Research*, vol. 5, n° 1, pp. 90–98. 1986.
- [KROGH, 1984] **KROGH, B.** “A generalised potential field approach to obstacle avoidance control”. Dans *SME World Conference on Robotics Research : The Next Five Years Beyond*. Bethlehem. Août 1984.
- [KRUPA *et al.*, 2003] **KRUPA, A.**, **GANGLOFF, J.**, **DOIGNON, C.**, **MATHELIN, M. D.**, **MOREL, G.**, **LEROY, J.** et **SOLER, L.** “Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing”. *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 842–853. Octobre 2003. DOI: 10.1109/TRA.2003.817086.
- [KYRKI *et al.*, 2004] **KYRKI, V.**, **KRAGIC, D.** et **CHRISTENSEN, H.** “New shortest-path approaches to visual servoing”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, vol. 1. Sendai, Japan, pp. 349–355. Septembre 2004. DOI: 10.1109/IROS.2004.1389376.

- [LAMIRAUX *et al.*, 2004] **LAMIRAUX, F.**, **BONNAFOUS, D.** et **LEFEBVRE, O.** “*Reactive path deformation for non-holonomic mobile robots*”. IEEE Transactions on Robotics and Automation, vol. 20, n° 6, pp. 967–977. Décembre 2004. DOI: 10.1109/TR0.2004.829459.
Disponible sur: <http://www.laas.fr/~florent/publi/04ieee-tra.html> (consulté le Septembre 2007).
King-Sun Fu Memorial Best Transactions on Robotics Paper Award 2004
- [LAUMOND, 2001] **LAUMOND, J.-P.**, rédacteur. “*La robotique mobile*”. Hermès Sciences Publication. 2001. ISBN 2-7462-0246-86.
- [LEPETIT et FUA, 2006] **LEPETIT, V.** et **FUA, P.** “*Monocular model-based 3d tracking of rigid objects*”. Found. Trends. Comput. Graph. Vis., vol. 1, n° 1, pp. 1–89. 2006. ISSN 1572-2740.
DOI: 10.1561/0600000001.
- [MALIS et CHAUMETTE, 2000] **MALIS, E.** et **CHAUMETTE, F.** “*2d/2 visual servoing with respect to unknown objects through a new estimation scheme of camera displacement*”. International Journal of Computer Vision, vol. 37, n° 1, pp. 79–97. Juin 2000.
- [MALIS *et al.*, 1999] **MALIS, E.**, **CHAUMETTE, F.** et **BOUDET, S.** “*2 1/2d visual servoing*”. IEEE Transactions on Robotics and Automation, vol. 15, n° 2, pp. 238–250. Avril 1999.
DOI: 10.1109/70.760345.
- [MALIS, 1998] **MALIS, E.** “*Contribution à la modélisation et à la commande en asservissement visuel*”. Thèse de doctorat, Université de Rennes 1, IRISA. Novembre 1998.
- [MANSARD, 2006] **MANSARD, N.** “*Enchaînement de tâches robotiques*”. Thèse de doctorat, Université de Rennes 1, mention informatique, Rennes, France. Décembre 2006.
- [MANSARD et CHAUMETTE, 2004] **MANSARD, N.** et **CHAUMETTE, F.** “*Tasks sequencing for visual servoing*”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, vol. 1. Sendai, Japan, pp. 992–997. Septembre 2004.
DOI: 10.1109/IROS.2004.1389482.
- [MANSARD et CHAUMETTE, 2005A] **MANSARD, N.** et **CHAUMETTE, F.** “*Visual servoing sequencing able to avoid obstacles*”. Dans *IEEE International Conference on Robotics and Automation (ICRA'05)*. Barcelona, Spain, pp. 3143–3148. Avril 2005.
- [MANSARD et CHAUMETTE, 2005B] **MANSARD, N.** et **CHAUMETTE, F.** “*A new redundancy formalism for avoidance in visual servoing*”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, vol. 2. Edmonton, Canada, pp. 1694–1700. August 2005.
DOI: 10.1109/IROS.2005.1545222.
- [MARCHAND et HAGER, 1998] **MARCHAND, E.** et **HAGER, G.** “*Dynamic sensor planning in visual servoing*”. Dans *IEEE International Conference on Robotics and Automation (ICRA'98)*, vol. 3. Leuven, Belgium, pp. 1988–1993. Mai 1998. DOI: 10.1109/ROBOT.1998.680607.
- [MARTINET *et al.*, 1998] **MARTINET, P.**, **THIBAUD, C.**, **THUILOT, B.** et **GALLICE, J.** “*Robust controller synthesis in automatic guided vehicules applications*”. Dans *Advances in Vehicles Control and Safety (AVCS'98)*. Amiens, France, pp. 395–401. Juillet 1998.
- [MARTINET et THIBAUD, 2000] **MARTINET, P.** et **THIBAUD, C.** “*Automatic guided vehicules : Robust controller design in image space*”. Autonomous Robots, vol. 8, n° 1, pp. 25–42. Janvier 2000. ISSN 0929-5593. DOI: 10.1023/A:1008936817917.
- [MATTHIES *et al.*, 1989] **MATTHIES, L.**, **KANADE, T.** et **SZELISKI, R.** “*Kalman filter-based algorithms for estimating depth from image sequences*”. International Journal of Computer Vision, vol. 3, n° 3, pp. 209–238. 1989.
- [MAYBECK, 1979] **MAYBECK, P. S.** “*Stochastic Models, Estimation, and Control : Volume I*”, vol. 141-1 de Mathematics in Science and Engineering. Academic Press, San Diego. 1979.
- [MEDIO et ORIOLO, 1991] **MEDIO, C. D.** et **ORIOLO, G.** “*Robot obstacle avoidance using vortex field*”. Advances in robotic kinematics, pp. 227–235,. 1991.
- [MEZOUAR et CHAUMETTE, 2002] **MEZOUAR, Y.** et **CHAUMETTE, F.** “*Avoiding self-occlusions and preserving visibility by path planning in the image*”. Robotics and Autonomous Systems, vol. 41, n° 2, pp. 77–87. Novembre 2002.
-

- [MORIN et SAMSON, 2003] **MORIN, P.** et **SAMSON, C.** “*Practical stabilization of driftless systems on lie groups : the transverse function approach*”. IEEE Transactions on Automatic Control, vol. 48, n° 9, pp. 1496–1508. Septembre 2003. DOI: 10.1109/TAC.2003.816963.
- [MURRAY et al., 1994] **MURRAY, R., LI, Z.** et **SASTRY, S.** “*A mathematical introduction to robotic manipulation*”. CRC Press, Boca Raton, FL, 1^{ère} édition. 1994. ISBN 0849379814.
- [OBERKAMPF et al., 1996] **OBERKAMPF, D., DEMENTHON, D.** et **DAVIS, L.** “*Iterative pose estimation using coplanar feature points*”. Computer Vision Image Understanding, vol. 63, n° 3, pp. 495–511. Mai 1996. ISSN 1077-3142. DOI: 10.1006/cviu.1996.0037.
- [OLIENSIS, 2002] **OLIENSIS, J.** “*Exact two-image structure from motion*”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, n° 12, pp. 1618–1633. Décembre 2002. DOI: 10.1109/TPAMI.2002.1114853.
- [PISSARD-GIBOLLET, 1993] **PISSARD-GIBOLLET, R.** “*Conception et commande par asservissement visuel d’un robot mobile*”. Thèse de doctorat, École des Mines de Paris, Paris, France. Décembre 1993.
- [PISSARD-GIBOLLET et RIVES, 1991] **PISSARD-GIBOLLET, R.** et **RIVES, P.** “*Asservissement visuel appliqué à un robot mobile : état de l’art et modélisation cinématique*”. Rapport technique 1577, INRIA, France. 1991.
- [PISSARD-GIBOLLET et RIVES, 1995] **PISSARD-GIBOLLET, R.** et **RIVES, P.** “*Applying visual servoing techniques to control a mobile hand-eye system*”. Dans *IEEE International Conference on Robotics and Automation (ICRA’95)*. Nagoya, Japan, pp. 166–171. Mai 1995. DOI: 10.1109/ROBOT.1995.525280.
- [PRESS et al., 1992] **PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T.** et **FLANNERY, B. P.** “*Numerical recipes in C*”. Cambridge University Press, Cambridge, 2^e édition. 1992. The art of scientific computing.
- [QUINLAN, 1994] **QUINLAN, S.** “*Real-Time modification of collision-free paths*”. Thèse de doctorat, Stanford University, Computer Science Department. Décembre 1994.
- [REMAZEILLES et al., 2006] **REMAZEILLES, A., MANSARD, N.** et **CHAUMETTE, F.** “*A qualitative visual servoing to ensure the visibility constraint*”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’06)*. Beijing, China, pp. 4297–4303. Octobre 2006. DOI: 10.1109/IROS.2006.281960.
- [RIVES et BORRELLY, 1997A] **RIVES, P.** et **BORRELLY, J.** “*Visual servoing techniques applied to an underwater vehicle*”. Dans *IEEE International Conference on Robotics and Automation (ICRA’97)*, vol. 3. Albuquerque, New Mexico, USA, pp. 1851–1856. Avril 1997. DOI: 10.1109/IROS.1997.656798.
- [RIVES et BORRELLY, 1997B] **RIVES, P.** et **BORRELLY, J.-J.** “*Underwater pipe inspection task using visual servoing techniques*”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’97)*, vol. 1. Grenoble, France, pp. 63–68. Septembre 1997. DOI: 10.1109/IROS.1997.648999.
- [RIVES et DEVY, 2001] **RIVES, P.** et **DEVY, M.** “*Perception pour la localisation*”. Dans *La robotique mobile*, édité par Laumond J.-P., pp. 139–193. Hermès Sciences Publication. ISBN 2-7462-0246-86. 2001.
- [SAMPEI et al., 1991] **SAMPEI, M., TAMURA, T., ITOH, T.** et **NAKAMICHI, M.** “*Path tracking control of trailer-like mobile robot*”. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’91)*. Osaka, Japon, pp. 193–198. Septembre 1991.
- [SAMPEI et al., 1995] **SAMPEI, M., TAMURA, T., KOBAYASHI, T.** et **SHIBUI, N.** “*Arbitrary path tracking control of articulated vehicles using nonlinear control theory*”. IEEE Transactions on Automatic Control, vol. 3, n° 1, pp. 125–131. Mars 1995. DOI: 10.1109/87.370718.
- [SAMSON, 1992] **SAMSON, C.** “*Path following and time-varying feedback stabilization of a wheeled mobile robot*”. Dans *2nd International Conference on Automation, Robotics and Computer Vision (ICARCV’92)*. Singapore, pp. RO-13.1.1–13.1.5. Septembre 1992.

- [SAMSON *et al.*, 1991] **SAMSON, C.**, **LE BORGNE, M.** et **ESPIAU, B.** “*Robot Control : The task function approach*”. Oxford science publications, Oxford, England, Clarendon Press édition. 1991.
- [SANDERSON et WEISS, 1980] **SANDERSON, A.** et **WEISS, L.** “*Image-based visual servo control using relational graph error signals*”. Dans *IEEE International Conference on Cybernetics and Society*. Cambridge, Massachusetts, pp. 1074–1077. Octobre 1980.
- [SANTOS-VICTOR et SANDINI, 1997] **SANTOS-VICTOR, J.** et **SANDINI, G.** “*Visual behaviors for docking*”. *Computer Vision and Image Understanding*, vol. 67, n° 3, pp. 223–238. Septembre 1997. ISSN 1077-3142. DOI: 10.1006/cviu.1997.0528.
- [SCHRAMM et MOREL, 2006] **SCHRAMM, F.** et **MOREL, G.** “*Ensuring visibility in calibration-free path planning for image-based visual servoing*”. *IEEE Transactions on Robotics and Automation*, vol. 22, n° 4, pp. 848–854. Août 2006. DOI: 10.1109/TR0.2006.878955.
- [SHAMPINE et GORDON, 1975] **SHAMPINE, L. F.** et **GORDON, M. K.** “*Computer Solution of Ordinary Differential Equations*”. W. H. Freeman, San Francisco. 1975.
- [SHIRAI et INOUE, 1973] **SHIRAI, Y.** et **INOUE, H.** “*Guiding a robot by visual feedback in assembling tasks*”. *Pattern Recognition*, vol. 5, pp. 99–108. 1973.
- [SINGER, 1993] **SINGER, M.** “*A general approach to moment calculation for polygons and line segments.*” *Pattern Recognition*, vol. 26, n° 7, pp. 1019–1028. 1993. DOI: 10.1016/0031-3203(93)90003-F.
- [SOATTO et PERONA, 1998A] **SOATTO, S.** et **PERONA, P.** “*Reducing "structure from motion" : A general framework for dynamic vision part 2 : Implementation and experimental assessment*”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, n° 9, pp. 933–942. Septembre 1998. DOI: 10.1109/34.713360.
- [SOATTO et PERONA, 1998B] **SOATTO, S.** et **PERONA, P.** “*Reducing "structure from motion" : A general framework for dynamic vision part 2 : Implementation and experimental assessment*”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, n° 9, pp. 943–960. Septembre 1998. DOI: 10.1109/34.713361.
- [SOATTO et PERRONA, 1995] **SOATTO, S.** et **PERRONA, P.** “*Visual motion estimation from point features : unified view*”. Dans *IEEE International Conference on Image Processing*, vol. 3. pp. 21–24. Octobre 1995. DOI: 10.1109/ICIP.1995.537570.
- [SOATTO *et al.*, 1996] **SOATTO, S.**, **FREZZA, R.** et **PERRONA, P.** “*Motion estimation via dynamic vision*”. *IEEE Transactions on Automatic Control*, vol. 41, n° 3, pp. 393–413. Mars 1996. DOI: 10.1109/9.486640.
- [SOUÈRES et CADENAT, 2003] **SOUÈRES, P.** et **CADENAT, V.** “*Dynamical sequence of multi-sensor based tasks for mobile robots navigation*”. Dans *7th Symposium on Robot Control (SYROCO'03)*. Wrocław, Poland, pp. 423–428. Sept. 2003.
- [SOUÈRES *et al.*, 1998] **SOUÈRES, P.**, **HAMEL, T.** et **CADENAT, V.** “*A path following controller for wheeled robots which allows to avoid obstacles during transition phase*”. Dans *IEEE International Conference on Robotics and Automation (ICRA'98)*, vol. 2. Leuven, Belgique, pp. 1269–1274. Mai 1998. DOI: 10.1109/ROBOT.1998.677277.
- [STEGER, 1996] **STEGER, C.** “*On the calculation of arbitrary moments of polygons*”. Technical Report FGBV-96-05, Forschungsgruppe Bildverstehen (FG BV), Informatik IX, Technische Universität München. Octobre 1996.
- [TAHRI, 2004] **TAHRI, O.** “*Application des moments à l’asservissement visuel et au calcul de pose*”. Thèse de doctorat, Université de Rennes 1, mention informatique, Rennes. Mars 2004.
- [TAHRI et CHAUMETTE, 2005] **TAHRI, O.** et **CHAUMETTE, F.** “*Point-based and region-based image moments for visual servoing of planar objects*”. *IEEE Transactions on Robotics and Automation*, vol. 21, n° 6, pp. 1116–1127. Décembre 2005. DOI: 10.1109/TR0.2005.853500.
- [THRUN *et al.*, 2001] **THRUN, S.**, **FOX, D.**, **BURGARD, W.** et **DALLAERT, F.** “*Robust monte carlo localization for mobile robots*”. *Artificial Intelligence*, vol. 128, n° 1-2, pp. 99–141. Mai 2001. ISSN 0004-3702. DOI: 10.1016/S0004-3702(01)00069-8.

- [TSAI et LENZ, 1989] **TSAI, R.** et **LENZ, R.** “A new technique for fully autonomous and efficient 3d robotics hand/eye calibration”. *IEEE Transactions on Robotics and Automation*, vol. 5, n° 3, pp. 345–358. 1989.
- [TSAKIRIS *et al.*, 1996] **TSAKIRIS, D.**, **SAMSON, C.** et **RIVES, P.** “Vision-based time-varying stabilization of a mobile manipulator”. Dans *International Conference on Control, Automation, Robotics and Vision (ICARCV'96)*. Westin Stanford, Singapore. Décembre 1996.
- [TSAKIRIS *et al.*, 1997] **TSAKIRIS, D.**, **SAMSON, C.** et **RIVES, P.** “Applying visual servoing techniques to control a non-holonomic mobile robots”. Dans *Workshop on “New Trends in Image-based Robot Servoing” (IROS'97)*. IEEE, Grenoble, France, pp. 8–12. Septembre 1997.
- [VACCHETTI et LEPETIT, 2004] **VACCHETTI, L.** et **LEPETIT, V.** “Stable real-time 3d tracking using online and offline information”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, n° 10, pp. 1385–1391. 2004. ISSN 0162-8828. DOI: 10.1109/TPAMI.2004.92.
- [VICTORINO *et al.*, 2001] **VICTORINO, A.**, **RIVES, P.** et **BORRELLY, J.-J.** “Mobile robot navigation using a sensor-based control strategy”. Dans *IEEE International Conference on Robotics and Automation (ICRA '01)*, vol. 4. Washington, D.C., USA, pp. 3924–3930. Mai 2001.
- [VITRANI *et al.*, 2006] **VITRANI, M.-A.**, **MOREL, G.**, **BONNET, N.** et **KAROUIA, M.** “A robust ultrason-based visual servoing approach for automatic guidance of a surgical instrument with *in vivo* experiments”. Dans *First IEEE/RAS-EMBS Biomedical Robotics and Biomechatronics (BioRob'06)*. pp. 35–40. Février 2006.
- [WUNSCH et HIRZINGER, 1997] **WUNSCH, P.** et **HIRZINGER, G.** “Real-time visual tracking of 3D objects with dynamic handling of occlusion”. Dans *IEEE International Conference on Robotics and Automation (ICRA '97)*, vol. 4. Albuquerque, Mexico, pp. 2868–2873. Avril 1997.
- [ZHANG, 2000] **ZHANG, Z.** “A flexible new technique for camera calibration”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n° 11, pp. 1330–1334. 2000.
-

Liste de mes publications

▷ Conférences internationales (avec actes et comités de lecture)

- [FOLIO et CADENAT, ECC2007] **FOLIO, DAVID** et **CADENAT, VIVIANE** “A new controller to perform safe vision-based navigation tasks amidst possibly occluding obstacles”. Dans *European Control Conference (ECC’07)*. Kos, Grèce, Juillet 2007.
- [FOLIO et CADENAT, ICINCO2007] **FOLIO, DAVID** et **CADENAT, VIVIANE** “Using simple numerical schemes to compute visual features whenever unavailable”. Dans *IFAC International Conference on Informatics in Control, Automation and Robotics (ICINCO’07)*. Angers, France, Mai 2007.
- [FOLIO et CADENAT, ROBIO2006] **FOLIO, DAVID** et **CADENAT, VIVIANE** “A redundancy-based scheme to perform safe vision-based tasks amidst obstacles”. Dans *IEEE/RSJ International Conference on Robotics and Biomimetics (ROBIO’06)*. Kunming, Yunnan, Chine. Décembre 2006.
Finaliste aux prix “T.J. Tarn Best Paper in Robotics”. (Taux d’acceptation : 45%)
- [FOLIO et CADENAT, ECC-CDC2005] **FOLIO, DAVID** et **CADENAT, VIVIANE** “A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment”. Dans *European Control Conference (ECC-CDC’05)*. Seville, Espagne. Décembre 2005.
(Taux d’acceptation : 48.2%)
- [FOLIO et CADENAT, ECMR2005] **FOLIO, DAVID** et **CADENAT, VIVIANE** “Using redundancy to avoid simultaneously occlusions and collisions while performing a vision-based task amidst obstacles”. Dans *European Conference on Mobile Robots (ECMR’05)*. Ancône, Italie. Septembre 2005. (Taux d’acceptation : 60.6%)

▷ International Workshop

- [FOLIO et CADENAT, ECMS2007] **FOLIO, DAVID** et **CADENAT, VIVIANE** “Dealing with visual features occlusions and collisions during a vision-based navigation task in cluttered environments”. Dans *8th International Workshop on Electronics, Control, Modelling, Measurement and Signals (ECMS’2007)*. Liberec, Czech Republic. Mai 2007 .
- [FOLIO et CADENAT, ECMS2005] **FOLIO, DAVID** et **CADENAT, VIVIANE** “A method to safely perform a visually guided navigation task amidst occluding obstacles”. Dans *7th International Workshop on Electronics, Control, Modelling, Measurement and Signals (ECMS’2005)*. Toulouse, France. Mai 2005 .

▷ Congrès Nationaux

[FOLIO et CADENAT, JNRR2005] **FOLIO, DAVID** et **CADENAT, VIVIANE** “*Gestion des problèmes d’occultation et de collision durant une tâche référencée vision*”. Poster présenté lors des Journées Nationales de la Recherche en Robotique (JNRR’05). Guidel, France. Octobre 2005.

[FOLIO, EDSYS2005] **FOLIO, DAVID** “*Évitement d’obstacles et gestion des occultations durant une tâche de navigation référencée vision*”. Dans *6ème Congrès des doctorants EDSYS (EDSYS’05)*. Toulouse, France. Mai 2005 .

▷ Rapport Techniques

[FOLIO et CADENAT, LAAS2006] **FOLIO, DAVID** et **CADENAT, VIVIANE** “*Using simple numerical schemes to compute visual features whenever unavailable*”. Rapport technique 06594, LAAS, Toulouse, France. Septembre 2006.

▷ Mémoires

[FOLIO, 2003] **FOLIO, DAVID** “*Navigation par enchaînement de tâches référencées multi-capteurs*”. Rapport de stage de DEA, Université Paul SABATIER, LAAS, Toulouse, France. Juin 2003.

Stratégies de commande référencées multi-capteurs et gestion de la perte du signal visuel pour la navigation d'un robot mobile

La synthèse de lois de commande efficaces apparaît comme un enjeu important dans la réalisation autonome de tâches de navigation robotiques. Ce problème peut être abordé par différentes approches. L'une d'entre elles, la commande référencée capteur, permet de définir les boucles de commande directement à partir des mesures sensorielles au lieu de les exprimer en fonction de l'état du robot. Dans ce contexte, du fait de la richesse du signal vidéo, la vision apparaît comme un capteur privilégié pour la réalisation de tâches très variées de manière précise. Cependant, la commande référencée vision reposant sur la seule régulation des indices visuels dans l'image s'avère mal adaptée pour réaliser des tâches de navigation complexes dans des environnements encombrés d'obstacles. En effet, dans ce cas, il est nécessaire de garantir d'une part la sécurité du robot, et d'autre part la disponibilité permanente des indices visuels dans l'image. Ce sont précisément ces problèmes que nous avons voulu étudier dans le cadre de cette thèse. Notre contribution a consisté à définir des stratégies de commande référencées multi-capteurs pour un robot mobile réalisant une tâche référencée vision dans un environnement encombré d'obstacles susceptibles d'occulter le motif visuel. Nous avons tout d'abord proposé des lois de commandes permettant d'éviter à la fois les occultations et les collisions. Toutefois, les résultats obtenus ont montré que chercher à éviter simultanément ces deux phénomènes sur-contrainait le mouvement du robot, limitant la gamme des missions réalisables. Nous avons alors développé une seconde approche consistant à tolérer temporairement la perte du signal visuel. Celle-ci repose sur l'exploitation de la réversibilité du lien vision/mouvement exprimé par le torseur d'interaction. Nous avons ainsi proposé dans un cadre général plusieurs méthodes (analytiques et numériques) de reconstruction du signal visuel lorsqu'il devient indisponible. Nous avons ensuite validé chacune de ces méthodes dans le cas de la réalisation d'une tâche de navigation référencée vision dans un milieu encombré d'obstacles. Nous avons également montré l'intérêt de nos approches lorsque la caméra présente un défaut de fonctionnement pendant l'exécution d'une mission.

Mots-clés : Asservissement visuel, Commande référencée capteurs, Évitement de collision et d'occultation, Estimation des indices visuels.

Multi-sensor-based control strategies and visual signal loss management for mobile robots navigation

The literature provides many techniques to design efficient control laws to realize robotic navigation tasks. In recent years, the sensors improvement gave rise to the sensor-based control which allows to define the robotic task in the sensor space rather than in the configuration space. In this context, as cameras provide high-rate meaningful data, visual servoing has been particularly investigated, and can be used to perform various and accurate navigation tasks. This method, which relies on the interaction between the camera and the visual features motions, consists in regulating an error in the image plane. Nonetheless, vision-based navigation tasks in cluttered environment cannot be expressed as a sole regulation of visual data. Indeed, in this case, it is necessary to preserve not only the robot safety (i.e. non-collision) but also the visual features visibility. This thesis addresses this issue and aims at developing sensor-based control laws allowing a mobile robot to perform vision-based tasks amidst possibly occluding obstacles. We have first proposed techniques able to fulfill simultaneously the two previously mentioned objectives. However, avoiding both collisions and occlusions often over-strained the robotic navigation task, reducing the range of realizable missions. This is the reason why we have developed a second approach which lets the visual features loss occurs if it is necessary for the task realization. Using the link between vision and motion, we have proposed different methods (analytical and numerical) to compute the visual signal as soon it becomes unavailable. We have then applied them to perform vision-based tasks in cluttered environments, before highlighting their interest to deal with a camera failure during the mission.

Keywords: Visual servoing, Sensor-based control, Collision and Occlusion avoidance, Visual Features estimation.