



**Titre:** Reconstruction tridimensionnelle et exploration autonome  
d'environnements non structurés

**Auteur:** Thomas Gerbaud

**Date:** 2004

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Gerbaud, T. (2004). Reconstruction tridimensionnelle et exploration autonome  
d'environnements non structurés [Master's thesis, École Polytechnique de  
Montréal]. PolyPublie. <https://publications.polymtl.ca/7385/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/7385/>  
PolyPublie URL:

**Directeurs de  
recherche:** Paul Cohen  
Advisors:

**Programme:** Unspecified  
Program:

UNIVERSITÉ DE MONTRÉAL

RECONSTRUCTION TRIDIMENSIONNELLE ET EXPLORATION  
AUTONOME D'ENVIRONNEMENTS NON STRUCTURÉS

THOMAS GERBAUD

DÉPARTEMENT DE GROUPE DE RECHERCHE EN PERCEPTION ET EN  
ROBOTIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE ÉLECTRIQUE)

NOVEMBRE 2004



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*

*ISBN: 0-494-01329-X*

*Our file* *Notre référence*

*ISBN: 0-494-01329-X*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

RECONSTRUCTION TRIDIMENSIONNELLE ET EXPLORATION  
AUTONOME D'ENVIRONNEMENTS NON STRUCTURÉS

présenté par: GERBAUD Thomas

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. NERGUIZIAN Chahé, Ph.D., président

M. COHEN Paul, Ph.D., membre et directeur de recherche

M. MICHAUD François, Ph.D., membre

## REMERCIEMENTS

Je tiens à remercier Paul Cohen, mon directeur de recherche, qui m'a accueilli dans son laboratoire et m'a permis d'étudier dans le domaine passionnant de la robotique mobile.

Je souhaite aussi souligner le soutien que m'a fourni Vladimir Polotski à travers ses conseils et ses suggestions.

D'autre part je souhaite saluer l'équipe du Groupe de Recherche en Perception et en Robotique de l'École Polytechnique de Montréal pour son ambiance chaleureuse, sa disponibilité et son aide notamment lors des expériences à l'extérieur.

Enfin j'ai une pensée toute particulière pour mes amis avec qui j'ai partagé de grands moments de complicité.

## RÉSUMÉ

Les robots mobiles autonomes sont souvent utilisés dans l'exploration d'environnements inconnus, non structurés et potentiellement dangereux. Un objectif important dans l'exploration est d'établir la carte tridimensionnelle du site pour informer les superviseurs humains de sa structure et de ses caractéristiques, et pour l'aider à planifier les missions à venir. Dans le cadre d'environnements extérieurs non structurés, une carte bidimensionnelle ne suffit pas et il est nécessaire de modéliser le terrain en trois dimensions. Cette carte doit aussi être suffisamment précise pour que la plate-forme l'utilise pour déterminer son espace des configurations, qui se traduit pour un robot mobile par la carte de traversabilité, et ainsi planifier sa trajectoire.

Les environnements non structurés présentent de nombreuses difficultés dues aux irrégularités du terrain, à la présence d'obstacles et à l'absence de repères facilement identifiables pour la localisation. Dans le domaine de l'exploration et de la cartographie de terrains non structurés, quatre difficultés majeures se dessinent : (1) la navigation dans ce milieu inconnu, demandant de développer une stratégie de déplacement prenant en compte les obstacles qui n'apparaissent qu'au fur et à mesure de l'exploration, (2) garantir l'exhaustivité de l'exploration, déterminer les zones inexplorées ; (3) acquérir les données, c'est-à-dire déterminer avec précision la position des points acquis ; (4) représenter les données de façon évolutive (permettre l'insertion et la suppression de nouvelles données), adaptée à la visualisation et à la nature non structurée du terrain.

Pour faire face à ces problèmes, l'originalité de notre approche est d'équiper un robot mobile d'un télémètre profilométrique incliné vers le bas. Il retourne alors des coupes de l'environnement, du terrain où il évolue. La modélisation du terrain est effectuée en rassemblant ces coupes dans une structure globale à l'aide d'un maillage représentant la surface du terrain. L'avantage d'un tel système est son faible coût

en calculs et sa grande précision. Des algorithmes de navigation permettent à la plate-forme de se déplacer de façon sécuritaire dans cet environnement en cours d'exploration.

Tout au long de sa trajectoire, la plate-forme se localise de façon tridimensionnelle grâce à son odomètre et à des capteurs inertiels. Ce positionnement de la plate-forme sert ensuite à positionner les différents profils entre eux puis à calculer la position des points les composant dans un repère global. Ces points sont ensuite placés dans un maillage basé sur une triangulation de Delaunay pour modéliser la surface du terrain. Différents filtres sont alors appliqués pour supprimer les points redondants, permettre à l'opérateur de fixer la résolution de la carte et rendre la modélisation consistante. Un module de visualisation en temps réel permet à l'opérateur de visionner, sur un ordinateur distant, cette structure au cours de sa construction. Le module de navigation réactive se base sur cette modélisation du terrain pour détecter les obstacles, construire la carte de traversabilité, planifier une trajectoire sécuritaire ainsi que pour détecter les zones à explorer.

Les résultats obtenus confirment la qualité du système. En effet la cartographie de scènes calibrées permet d'affirmer que les cartes créées ont une précision de l'ordre du centimètre sur plusieurs mètres parcourus. Les filtres appliqués sur la carte se sont révélés très efficaces, notamment le filtre de simplification de surface qui permet de diviser le nombre de points par cinq (selon le type de terrain) tout en conservant la précision de la modélisation. D'autre part, au cours des nombreux tests menés en intérieur et extérieur, les algorithmes de navigation réactive se sont révélés fiables et très sécuritaires.

Cependant certaines limitations sont à noter : (1) la précision de la carte créée dépend directement de la qualité de la localisation de la plate-forme et, comme celle-ci tend à diverger, ceci se manifeste par un manque de cohérence de la carte obtenue si la plate-forme modélise la même zone à deux moments relativement éloignés ; une amélioration serait donc de faire correspondre les zones pour corriger

la localisation de la plate-forme. (2) les algorithmes de navigation proposés sont en fait des fonctionnalités et il faudrait les améliorer et les combiner pour développer un algorithme d'exploration entièrement autonome qui n'aurait plus besoin d'un opérateur.

Enfin, les principales contributions scientifiques de notre projet sont : le développement d'un système de cartographie en haute résolution et en temps réel de terrains non structurés, ainsi que la mise en place d'algorithmes de navigation autonome à partir de la modélisation en cours de construction.

## ABSTRACT

Autonomous mobile platforms are often used to explore and map unknown, unstructured and potentially hazardous environments. An important goal in the exploration process is to produce a 3D survey map of visited sites, in order to inform human supervisors about their structures, useful features or hazards, and to help to plan subsequent missions. For outdoor and unstructured environments a 2D map is not sufficient and a 3D terrain model is required. This map should also be precise enough to help the platform determine its configuration space, the traversability map for mobile robots, and to plan its trajectory between the obstacles.

Unstructured indoor or outdoor environments present difficult challenges caused by uneven terrain, the presence of obstacles and the absence of easily distinguishable landmarks for localization. There are four major difficulties in unstructured terrain mapping : (1) to navigate in an unknown environment and to take into account the obstacles ; (2) to grant the exploration's exhaustiveness, to determine the unexplored areas ; (3) to acquire the ground data with a sufficient precision ; (4) to represent the data and to model uneven terrain in a structure well suited for insertion, deletion and visualization.

Our approach is based on a range finder mounted on a mobile robot aiming at the ground and also returning environment profiles. The terrain model is then computed by merging these slices into a global mesh representing the terrain. The advantages of such a system are low computation cost and high precision. Navigation algorithms grant safe navigation of the platform in the explored environment. Odometer and inertial sensors are used to compute the platform 3D localization. This pose is then used to place the profiles and calculate the points position in a global frame. Those points are placed into a mesh based on a Delaunay triangulation in order to model the terrain surface. Several filter are applied to remove redundant points, to allow the operator to set the map resolution, and to ensure the model

consistency. A real-time visualization module allows the operator to supervise the exploration process and see the map during its construction on a remote computer. The reactive navigation is based on this 3D map to detect obstacles, to build the traversability map, to plan a safe trajectory and to detect unexplored areas.

Results confirm that the mapping of calibrated scenes shows that our system precision is about one centimetre. The filters applied to the map are also very effective ; for example the surface simplification filter reduces the point number by a factor 5, depending on the ground type, with no significant loss of precision. Otherwise, during the numerous tests made indoor and outdoor, the reactive navigation algorithms have been shown to be very safe and reliable.

However we can note some limitations. (1) the map precision directly depends on the platform localization quality, and as this localization tends to diverge, it results in a loss in the map consistence if the platform models twice the same area at quite different time. This could be improve by matching the areas and correcting the platform localization. (2) Some functionalities of the navigation algorithms should be improved and combined to develop fully autonomous exploration algorithms that should not required the presence of an operator.

The main scientific contributions of this project are : the development of a high resolution and real-time mapping system for unstructured environments, and the setting up of autonomous navigation algorithms based on the model during its construction.

## TABLE DES MATIÈRES

REMERCIEMENTS . . . . .	iv
RÉSUMÉ . . . . .	v
ABSTRACT . . . . .	viii
TABLE DES MATIÈRES . . . . .	x
LISTE DES TABLEAUX . . . . .	xiii
LISTE DES FIGURES . . . . .	xiv
LISTE DES NOTATIONS ET DES SYMBOLES . . . . .	.xviii
LISTE DES ANNEXES . . . . .	xix
INTRODUCTION . . . . .	1
CHAPITRE 1 PROBLÉMATIQUE . . . . .	6
1.1 Revue bibliographique . . . . .	6
1.1.1 Présentation des travaux les plus représentatifs . . . . .	8
1.1.2 Critiques . . . . .	10
1.2 Notre système . . . . .	11
1.2.1 Objectifs et problèmes . . . . .	11
1.2.2 Architecture du système . . . . .	13
1.2.2.1 Architecture fonctionnelle . . . . .	13
1.2.2.2 Point de vue logiciel . . . . .	15
1.2.2.3 Architecture matérielle . . . . .	17
1.2.3 Scénario typique d'une exploration . . . . .	20

CHAPITRE 2	CARTOGRAPHIE . . . . .	22
2.1	Acquisition des données télémétriques . . . . .	22
2.1.1	Positionnement du robot/odométrie 3D . . . . .	23
2.1.2	Changements de repères . . . . .	26
2.2	Structure de triangulation . . . . .	29
2.2.1	Présentation de la triangulation de Delaunay . . . . .	30
2.2.1.1	Principe . . . . .	30
2.2.1.2	Librairies . . . . .	32
2.2.1.3	Algorithmes . . . . .	34
2.3	Filtrage des données . . . . .	36
2.3.1	Filtrage spatial de proximité . . . . .	37
2.3.2	Filtrage de simplification . . . . .	38
2.4	Visualisation . . . . .	43
CHAPITRE 3	MODULE D'EXPLORATION ET DE NAVIGATION . . . . .	47
3.1	Détection de zones traversables . . . . .	48
3.1.1	Détection des obstacles . . . . .	49
3.1.2	Création de la carte de traversabilité . . . . .	51
3.2	Modes de navigation . . . . .	52
3.2.1	Navigation purement réactive . . . . .	52
3.2.2	Navigation avec but . . . . .	53
3.2.3	Navigation vers les zones inconnues . . . . .	54
3.2.3.1	Détection des frontières . . . . .	55
3.2.3.2	Choix de la frontière à explorer . . . . .	56
3.3	Contrôleur de déplacement . . . . .	57
3.3.1	Le contrôleur d'Astolfi . . . . .	58
3.3.2	Contrôleur proposé . . . . .	59

CHAPITRE 4	RÉSULTATS . . . . .	62
4.1	Simulation unidimensionnelle . . . . .	63
4.2	Présentation des interfaces . . . . .	65
4.3	Validation de la reconstruction . . . . .	67
4.3.1	Calibration du télémètre laser . . . . .	67
4.3.2	Comparaison de scènes calibrées . . . . .	68
4.3.3	Modélisation de scènes intérieures . . . . .	70
4.3.4	Modélisation de scènes extérieures . . . . .	71
4.3.5	Régions occlusées . . . . .	76
4.3.6	Influence de l'inclinomètre . . . . .	76
4.4	Validation des modes d'exploration . . . . .	81
CONCLUSION	. . . . .	86
RÉFÉRENCES	. . . . .	89
ANNEXES	. . . . .	92

**LISTE DES TABLEAUX**

TAB. 2.1	Part de points restant après application du filtre de proximité en fonction du seuil et du type de terrain modélisé . . . . .	38
TAB. 2.2	Pourcentage de points restants après simplification de la modélisation en fonction du type de terrain (les seuils des filtres de proximité et de simplification sont fixés à 2 cm) . . . . .	40
TAB. 4.1	Mesures d'une scène calibrée . . . . .	69

## LISTE DES FIGURES

FIG. 1	Un exemple de robot mobile d'exploration extraterrestre : les robots <i>Spirit</i> et <i>Opportunity</i> de la NASA envoyés sur Mars en 2003 (©NASA Copyright Free Policy) . . . . .	2
FIG. 2	Schéma du montage de la plate-forme étudiée . . . . .	5
FIG. 1.1	Architecture générale . . . . .	14
FIG. 1.2	Architecture de type client-serveur . . . . .	16
FIG. 1.3	Architecture détaillée . . . . .	16
FIG. 1.4	Le robot ATRV utilisé avec le télémètre laser SICK (en bleu) dessus et l'inclinomètre et le gyromètre (à travers la vitre) . . . . .	18
FIG. 1.5	Schéma du montage de l'ATRV . . . . .	19
FIG. 1.6	Le gyromètre (à gauche) et l'inclinomètre (à droite) montés sur le robot ATRV . . . . .	19
FIG. 2.1	Définition de l'attitude du robot grâce aux angles d'Euler . . . . .	24
FIG. 2.2	Caractérisation du vecteur de déplacement élémentaire . . . . .	25
FIG. 2.3	Situation du plan de capture du télémètre . . . . .	26
FIG. 2.4	Définition d'un point dans le repère du télémètre . . . . .	27
FIG. 2.5	Positionnement du télémètre par rapport au robot . . . . .	27
FIG. 2.6	Définition du repère lié au robot . . . . .	28
FIG. 2.7	Grille d'occupation . . . . .	31
FIG. 2.8	Principe des <i>quad-trees</i> : approximation d'une courbe . . . . .	31
FIG. 2.9	Définition des éléments d'une triangulation . . . . .	32
FIG. 2.10	Triangulation d'un ensemble de points : en haut une triangulation quelconque et en bas la triangulation de Delaunay (les cercles permettent de vérifier le critère de Delaunay) . . . . .	33
FIG. 2.11	Triangulation de Delaunay et à gauche son dual, le diagramme de Voronoï . . . . .	33

FIG. 2.12	Principe du <i>line-walk</i> . . . . .	35
FIG. 2.13	Principe du basculement des arêtes . . . . .	36
FIG. 2.14	Insertion d'un point et rétablissement du critère de Delaunay par basculement des arêtes . . . . .	36
FIG. 2.15	Influence du filtre de proximité sur un scène extérieure : sans filtre (à gauche), avec un seuil de 2 cm (au centre) puis de 5cm (à droite) . . . . .	38
FIG. 2.16	Exemple de simplification d'une scène intérieure . . . . .	41
FIG. 2.17	Exemple de simplification d'une scène extérieure . . . . .	41
FIG. 2.18	Exemple d'effet néfaste du rajout d'un point après une sim- plification en ligne . . . . .	42
FIG. 2.19	Exemple de visualisation en trois dimensions, mode plein en haut et mode fil de fer en bas . . . . .	44
FIG. 2.20	Exemple de carte d'élévation. La ligne blanche représente la trajectoire du robot et les zones uniformes les zones incon- nues, l'œil représente le point de vue de la photographie 2.21	45
FIG. 2.21	Photographie du site de la carte d'élévation 2.20 . . . . .	46
FIG. 3.1	Photographie de la situation des cartes locales de la figure 3.2. (la photographie est prise à partir du coin inférieur droit des cartes) . . . . .	49
FIG. 3.2	Les quatre cartes locales . . . . .	50
FIG. 3.3	Principe de l'exploration purement réactive . . . . .	53
FIG. 3.4	Principe de l'exploration avec but . . . . .	54
FIG. 3.5	Carte d'élévation (à gauche) et la carte de traversabilité ré- sultante (à droite). Les frontières détectées sont représentées en blanc. . . . .	56
FIG. 3.6	Paramètres du contrôleur d'Astolfi . . . . .	58

FIG. 3.7	Comparaison des trajectoires obtenues avec le contrôleur d'As-tolfi et avec notre contrôleur . . . . .	60
FIG. 4.1	Simulation sur un sol plat . . . . .	64
FIG. 4.2	Comparaison de méthodes pour estimer la hauteur de la plate-forme . . . . .	66
FIG. 4.3	Principe de la calibration de la position du télémètre laser .	67
FIG. 4.4	Cartographie d'une scène calibrée . . . . .	69
FIG. 4.5	Photographie du laboratoire avec le système utilisé pour faire varier l'attitude de la plate-forme . . . . .	70
FIG. 4.6	Modélisation d'un environnement intérieur : le laboratoire correspondant à la photographie de la figure 4.5 . . . . .	71
FIG. 4.7	Photographie correspondant à un emplacement de type canyon (concave) . . . . .	72
FIG. 4.8	Résultat de cartographie 3D pour l'environnement de la figure 4.7 . . . . .	72
FIG. 4.9	Photographie correspondant à un emplacement de type butte (convexe) . . . . .	73
FIG. 4.10	Résultat de cartographie 3D pour l'environnement de la figure 4.9. Les troncs d'arbres ressortent bien de la surface. .	73
FIG. 4.11	Photographie d'un environnement extérieur structuré . . . .	74
FIG. 4.12	Résultat de cartographie 3D pour l'environnement de la figure 4.11 . . . . .	74
FIG. 4.13	Comparaison de l'incidence des rayons du télémètre selon le type de terrain (concave à gauche et convexe à droite). . . .	75
FIG. 4.14	Photographie du site de la figure 4.15 (la photographie est prise à partir d'un point situé en bas à droite) . . . . .	76
FIG. 4.15	Exemple de divergence du module de localisation 3D . . . .	77

FIG. 4.16	Exemple de zones d'occlusion (encerclée en blanc) suite au contournement d'un angle . . . . .	78
FIG. 4.17	Comparaison des données fournies par l'inclinomètre original (en haut) et celui avec le fluide visqueux (en bas) (les valeurs moyennes ont été décalées pour pouvoir comparer l'allure des deux signaux) . . . . .	79
FIG. 4.18	Influence des inclinomètres sur la reconstruction. La modélisation sans amortissement (à gauche) est bien plus irrégulière que celle avec amortissement (à droite). L'amplitude des variations est statistiquement deux fois plus élevée dans le cas sans amortissement. . . . .	80
FIG. 4.19	Exemple de navigation intérieure en mode réactif (partie pleine de la trajectoire) puis en mode retour au point de départ (partie de la trajectoire en tirets) . . . . .	82
FIG. 4.20	Navigation réactive en milieu extérieur. Les zones encerclées représentent les changements de direction causés par le trottoir. . . . .	83
FIG. 4.21	Exemple de navigation avec but : la plate-forme contourne un obstacle pour atteindre son objectif. . . . .	83
FIG. 4.22	Exemple de situation où la plate-forme n'arrive pas à atteindre le point-cible (le point blanc à droite). . . . .	84
FIG. 4.23	Exemple de navigation réactive puis retour au point de départ à partir du point blanc . . . . .	85
FIG. I.1	Simulation unidimensionnelle de la plate-forme sur un terrain modélisé en segments de droite. . . . .	92
FIG. I.2	Simulation de la roue sur une partie anguleuse. . . . .	93
FIG. II.1	Principe de fonctionnement de l'interféromètre de Sagnac (à gauche) et le dispositif utilisé dans l'Autogyro de KVH (à droite) . . . . .	98

**LISTE DES NOTATIONS ET DES SYMBOLES**

$h$ :	hauteur du télémètre par rapport au sol
$d$ :	avancement du télémètre par rapport au centre du robot
$\alpha$ :	angle d'inclinaison du télémètre par rapport à l'horizontale
$\beta$ :	angle du rayon du télémètre
$l$ :	longueur mesurée par le télémètre pour un rayon donné
$\theta$ :	angle de lacet
$\phi$ :	angle de tangage
$\psi$ :	angle de roulis
$dp$ :	déplacement élémentaire

**INDICES**

$t$ :	repère du télémètre
$r$ :	repère du robot
$0$ :	repère global

## LISTE DES ANNEXES

ANNEXE I	SIMULATION UNIDIMENSIONNELLE ET FILTRE DE KALMAN . . . . .	92
	I.1 Simulation unidimensionnelle . . . . .	92
	I.2 Filtre de Kalman . . . . .	93
ANNEXE II	PRÉSENTATION DU MATÉRIEL UTILISÉ . . . . .	96
	II.1 Présentation du robot ATRV . . . . .	96
	II.2 Télémètre laser . . . . .	96
	II.3 Le gyromètre . . . . .	97
	II.4 L'inclinomètre . . . . .	98
ANNEXE III	MANUEL D'UTILISATION DU PROGRAMME . . . . .	100
	III.1 Utilisation sur le terrain . . . . .	100
	III.1.1 Lancement du programme sur le robot . . . . .	100
	III.1.2 Lancement du client de visualisation . . . . .	101
	III.1.3 Exploration . . . . .	102
	III.2 Utilisation en mode hors-ligne . . . . .	103
	III.3 Conseils . . . . .	104
	III.3.1 Utilisation sur l'ordinateur Artemis . . . . .	104
	III.3.2 Utilisation en extérieur . . . . .	104
	III.3.3 Problèmes possibles . . . . .	104

## INTRODUCTION

Un robot mobile est une plate-forme, montée sur roues, chenilles ou pattes, équipée d'actionneurs et de capteurs contrôlés par ordinateur. Cette plate-forme évolue dans un environnement, intérieur ou extérieur, dynamique dans le cas général, pour accomplir des tâches aussi variées que la reconnaissance, le déminage ou encore le forage. Un environnement particulièrement intéressant est un environnement de type planétaire, qui est de plus en plus d'actualité avec les missions d'exploration extraterrestres confiées à des robots, comme celles sur la planète Mars avec les sondes *Spirit* et *Opportunity* (figure 1).

Une étape importante en robotique mobile est d'explorer, de manière plus ou moins autonome, un environnement et d'en dresser la carte. Cette dernière peut avoir de nombreuses applications comme entre autres fournir à l'opérateur une connaissance de l'environnement ou encore planifier des missions futures dans la zone explorée.

La *cartographie* robotique est le problème d'acquisition de modèles spatiaux de l'environnement physique en utilisant un ou plusieurs robots mobiles ; le résultat de ce processus est la carte, autrement appelée modélisation de l'environnement. Ce problème de cartographie est généralement considéré comme le problème le plus important en robotique mobile autonome (Latombe, 1991). Les capteurs utilisés pour cartographier l'environnement à l'aide de robots mobiles sont très variés : radar, télémètre (laser, sonar), lumière structurée, caméra ou encore des capteurs tactiles. Cependant, tous les capteurs ont une portée limitée et il est nécessaire que le robot navigue au travers de l'environnement pour le modéliser dans son intégralité.

L'exploration et la cartographie d'un milieu inconnu non structuré à l'aide d'un robot mobile présente de nombreux problèmes et difficultés :



FIG. 1 Un exemple de robot mobile d'exploration extraterrestre : les robots *Spirit* et *Opportunity* de la NASA envoyés sur Mars en 2003 (©NASA Copyright Free Policy)

1. Développer une stratégie de déplacement qui permette à la plate-forme d'évoluer dans cet environnement inconnu de façon sécuritaire. Il est nécessaire de développer des algorithmes de navigation réactifs qui prennent en compte les caractéristiques de l'environnement en cours d'exploration. Ces stratégies de navigation peuvent être locales, c'est-à-dire complètement réactives, ou bien plus globales, avec par exemple comme objectif d'atteindre un point.
2. Garantir l'exhaustivité de l'exploration, s'assurer que des zones n'ont pas été oubliées. Pour cela il faut développer des algorithmes capables de détecter les régions inexplorées et d'organiser, de planifier la visite de ces régions en privilégiant les régions potentiellement les plus intéressantes et les moins coûteuses en terme d'accès.
3. L'acquisition des données est un problème central. Il s'agit de représenter les données acquises à différents instants dans un repère commun global. Pour cela il faut prendre en compte les variations d'attitude du capteur dues aux inégalités du terrain sur lequel se déplace la plate-forme, mettre en correspondance les données pour améliorer la localisation de la plate-forme (localisation

simultanée), repérer les lacunes et les redondances au sein des données sur l'environnement. Un autre point à prendre en compte est le bruit de mesure qui oblige à filtrer les données et qui, dans le cas des capteurs de positionnement, induit une erreur s'accumulant au cours du processus.

4. La représentation de l'environnement est aussi un problème critique. Elle doit être évolutive pour permettre l'insertion et l'élimination de détails dans la carte, la condensation de données proches. Une organisation hiérarchique permettrait d'avoir une approche multi-résolutions de l'environnement modélisé. Enfin cette représentation doit être efficace en terme d'accès, c'est-à-dire qu'elle doit être facilement visualisable et interactive. Il serait aussi très utile qu'elle gère les environnements dynamiques où des objets peuvent bouger.

Sur un plan historique, la cartographie robotique a débuté dans les années 1980 selon deux approches : métriques et topologiques. L'approche métrique modélise l'environnement par des grilles d'occupation (Elfes, 1987) tandis que l'approche topologique divise l'environnement en repères reliés entre eux par des chemins (Mataric, 1990; Kuipers and Byun, 1988). Puis, depuis les années 1990, l'approche probabiliste a pris le dessus en intégrant les erreurs de mesure dans les modèles. C'est au cours de cette période qu'à été introduit le problème de résolution simultanée de la cartographie et de la localisation, autrement appelé *SLAM* (*Simultaneous Localization and Mapping* en anglais) ou encore *CML* (*Concurrent Mapping and Localization* en anglais) (Dissanayake et al., 2000). Ce dernier aspect du problème, qui occupe encore la plupart des équipes de recherche traitant de la cartographie robotique, consiste à estimer conjointement la position du robot et la position des différentes caractéristiques de l'environnement. Les modèles développés sont basés sur des filtres de Kalman étendus, des algorithmes *EM* (*Expectation Maximization*) ou encore des approches par mise en correspondance des données. Ces cartes étaient alors bidimensionnelles mais depuis cinq ans environ, l'évolution des cap-

teurs, notamment des télémètres, a permis d'aborder la cartographie robotique en trois dimensions.

Cependant ces dernières recherches se sont principalement limitées à la cartographie de milieux intérieurs, la plupart du temps structurés (Surmann et al., 2003; Martin and Thrun, 2002) mais aussi naturels tels que les mines (Thrun et al., 2003) ou extérieurs (Früh and Zakhor, 2002). Aussi la cartographie tridimensionnelle d'environnements non structurés, tels des environnements naturels, n'est abordée qu'en utilisant des caméras stéréoscopiques, qui ne permettent pas une aussi bonne précision que les télémètres.

Le but de ce projet est donc de concevoir un système d'exploration et de cartographie précise d'environnements inconnus non structurés. Le système proposé est basé sur un robot mobile équipé d'un télémètre profilométrique 2. En rassemblant des coupes successives de l'environnement, le système est alors capable de cartographier et de modéliser avec précision l'environnement en trois dimensions. Cette modélisation est en même temps utilisée pour élaborer la navigation de la plateforme et l'exploration du site. Pour rester dans le cadre général, et dans le cadre des missions d'exploration extraterrestre confiées à des robots, les environnements ciblés sont les terrains extérieurs non structurés. La surface du terrain où évolue le robot n'étant pas plane ni horizontale, le robot mobile est aussi équipé de capteurs d'attitude lui permettant de se localiser dans les trois dimensions. Cependant nous nous limitons aux environnements statiques, c'est-à-dire les scènes où la plateforme est le seul élément mobile.

Un point fondamental étant l'exploration d'un environnement inconnu, nous proposons aussi des algorithmes de navigation réactive basés sur une carte d'obstacles créée à partir de la modélisation tridimensionnelle du terrain. Un algorithme est aussi proposé pour déterminer les zones encore inexplorées.

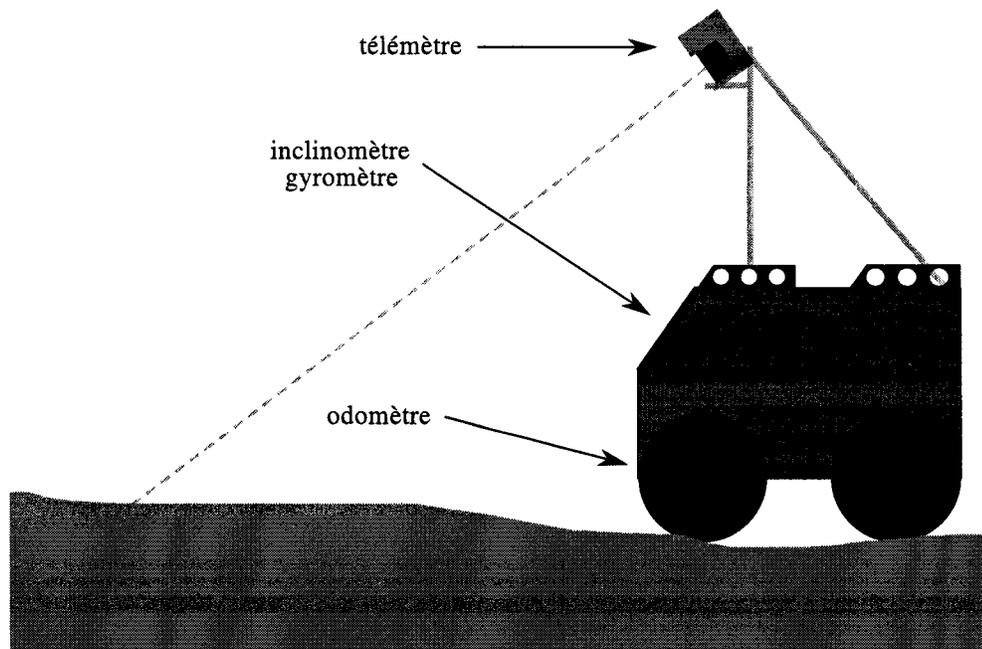


FIG. 2 Schéma du montage de la plate-forme étudiée

Dans ce mémoire nous présenterons tout d'abord dans le premier chapitre une revue littéraire sur les différentes études menées dans le domaine de la cartographie en robotique mobile, pour ensuite justifier notre approche et présenter l'architecture du système proposé. Le deuxième chapitre couvre la modélisation de l'environnement : après avoir calculé la position de la plate-forme, les données acquises par le télémètre sont insérées dans une structure de triangulation pour modéliser la surface du sol ; quelques filtres sont aussi appliqués pour améliorer la carte créée. Le troisième chapitre traite de la navigation et de l'exploration de l'environnement inconnu que la plate-forme doit cartographier ; après avoir déterminé l'espace libre autour du robot, différents algorithmes permettant une exploration sécuritaire sont présentés. Enfin le quatrième chapitre présente quelques résultats significatifs obtenus avec notre système, aussi bien sur le plan de la modélisation du terrain que sur les algorithmes de navigation.

## CHAPITRE 1

### PROBLÉMATIQUE

Nous allons tout d'abord procéder à une revue bibliographique pour évaluer les travaux des différentes équipes de recherche dans le domaine de la cartographie en robotique mobile, plus particulièrement de la cartographie tridimensionnelle.

À partir de cette analyse, nous allons redéfinir les objectifs de notre système de cartographie autonome ainsi qu'exposer les difficultés inhérentes. Nous présenterons alors l'architecture du système développé sur trois plans : fonctionnel, logiciel et matériel. Enfin nous terminerons le chapitre par la description d'un scénario typique d'une mission de cartographie.

#### 1.1 Revue bibliographique

L'exploration et la cartographie de milieux intérieurs et extérieurs est un domaine de recherche qui s'est fortement développé ces cinq dernières années et de nombreux projets ont été créés dans cette voie. Les équipes cartographiant les intérieurs en deux dimensions, créant ainsi une carte des murs, sont très nombreuses. Des sonars ont d'abord été utilisés du fait de leur faible coût et de leur simplicité (Roy and Dudek, 2001). L'apparition des télémètres lasers a permis d'obtenir des cartes plus précises du fait de sa portée, de sa précision et de la densité des données capturées. Les équipes de recherche ont alors développé des techniques de localisation et de cartographie simultanée (*SLAM*) pour optimiser ces cartes (Lemaire, 2004) : construire la carte d'un environnement inconnu et conjointement estimer la position de la plate-forme dans cette carte en cours de construction, par mise en

correspondance des profils acquis. Un autre axe de recherche concerne l'exploration et la construction de cartes à l'aide de plusieurs robots, ainsi que l'optimisation de leur parcours pour obtenir la carte la plus complète et le plus rapidement possible (Burgard et al., 2000).

Les travaux récents se sont intéressés à la cartographie 3D, reconstruisant soit le volume des environnements intérieurs, soit la surface du terrain pour pouvoir ensuite naviguer (comme dans les applications d'exploration planétaires par exemple), soit des villes comme dans le cas de l'Université de Californie (Früh and Zakhor, 2002). Les capteurs utilisés varient mais deux types de capteurs dominant : les télémètres laser (Thrun et al., 2001; Früh and Zakhor, 2002; Borenstein and Ye, 2002), modifiés par certaines équipes pour pouvoir acquérir des scènes tridimensionnelles (Surmann et al., 2003), et les caméras stéréoscopiques (Lacroix et al., 2002; Maurette, 2003).

En ce qui concerne l'exploration et la navigation, Singh et son équipe de l'institut de robotique de l'Université Carnegie Mellon (Singh et al., 2000) proposent une solution de progression basée sur la création d'une carte de traversabilité locale et à une planification globale basée sur l'algorithme D\*, modifiant localement la trajectoire globale en fonction des obstacles apparus localement. D'autres groupes de recherche (Lacroix et al., 2002; Maurette, 2003; Burgard et al., 2000) proposent aussi des algorithmes basés sur les cartes de traversabilité. L'aspect exploration associé à la représentation tridimensionnelle n'a pas été encore très bien couvert, si ce n'est l'équipe de Surmann dans le cas de scènes intérieures (Surmann et al., 2003).

### 1.1.1 Présentation des travaux les plus représentatifs

Un travail incontournable est celui de l'équipe de Thrun de l'Université Carnegie Mellon (États Unis) en collaboration avec l'Université de Freiburg (Allemagne). Ils se sont intéressés à construire des cartes volumétriques d'environnements intérieurs (Thrun et al., 2001; Martin and Thrun, 2002). Ils utilisent un robot sur lequel sont montés deux télémètres, un vertical balayant le plafond perpendiculairement à sa trajectoire et un horizontal scrutant l'espace devant lui. Le télémètre vertical sert à prendre des profils des couloirs et des pièces pour ensuite modéliser leurs volumes. Le télémètre horizontal permet d'améliorer grandement l'odométrie et donc à positionner avec une bonne précision les coupes verticales par des techniques de *SLAM*. Une caméra omnidirectionnelle est aussi montée à bord pour associer une couleur et une texture aux surfaces reconstruites. Un autre système développé utilise un télémètre fixé sur une plate-forme *tilt* leur permettant de balayer des scènes, des façades de bâtiments dans leurs exemples, et de les modéliser. Les environnements à représenter étant principalement structurés, des méthodes de modélisation de surface sont utilisées. Cette équipe s'est aussi penchée sur la cartographie bidimensionnelle d'intérieurs de bâtiments, sur la localisation au sein de ces cartes ainsi qu'à une exploration concertée avec plusieurs robots dans le but d'optimiser leur répartition (Burgard et al., 2000).

L'équipe de Surmann et Hertzberg du Fraunhofer Institute for Autonomous Intelligent Systems en Allemagne propose un système cartographiant les environnements intérieurs (Surmann et al., 2003). Ce système est basé sur un télémètre laser de type SICK monté sur un dispositif contrôlant l'inclinaison, le tout sur un robot mobile. L'opération consiste à prendre une série de profils devant le robot et de les rassembler pour obtenir le modèle tridimensionnel de l'environnement. Cette équipe a développé différents post-traitements pour en extraire les surfaces ou encore seg-

menter l'environnement en objets. Une autre partie intéressante de leurs travaux (Nüchter et al., 2003) consiste à minimiser le nombre de balayages à effectuer pour modéliser une pièce. Il s'agit de trouver la position suivante pour, lors du balayage, obtenir le maximum d'information sur les zones inconnues. Leur structure de donnée utilisée pour modéliser l'environnement est basée sur les *octree* qui consistent à subdiviser l'environnement en cubes de plus en plus petits et à déterminer s'ils sont occupés ou non.

L'équipe de l'University of Michigan (États Unis) s'est intéressé à cartographier le sol à l'aide d'un télémètre laser incliné vers le sol et se déplaçant horizontalement (Borenstein and Ye, 2002). Sa représentation est basée sur une carte d'élévation de résolution de 2cm environ et son étude porte sur les filtres à appliquer à cette carte pour améliorer la reconstruction : réduire l'erreur, enlever les artefacts et les points erratiques apparaissant lorsque le télémètre balaie les arêtes des objets. Le filtre CAS (*Certainty Assited Spatial Filter*) a été ainsi mis au point ; il est basé sur la probabilité qu'un point soit à une certaine hauteur, étant donnés les points acquis précédemment.

L'équipe du LAAS en France (Lacroix et al., 2002) utilise un robot tout-terrain à structure flexible pour cartographier le sol. Sur ce robot est montée une caméra stéréoscopique regardant le sol à l'avant. Ce système fournit une carte d'élévation (DEM : *Digital Elevation Map*) modélisant le sol d'une précision de l'ordre de la dizaine de centimètre. De nombreux aspects sont abordés : estimation du déplacement en utilisant la caméra, algorithme de localisation, simulation du comportement du robot sur un sol rocailleux (en effet les essieux du robot sont mobiles pour lui permettre de conserver une assiette constante tout en franchissant des obstacles). Les aspects de navigation autonome sont aussi abordés : leur algorithme est basé sur un découpage de l'environnement en zones traversables ou infranchissables, issu de la carte d'élévation.

L'Université de Californie Berkeley (États Unis) a développé un système de cartographie de façades de bâtiments (Früh and Zakhor, 2002). Pour cela deux télémètres laser ont été montés sur un véhicule. Les deux regardent sur le côté droit, l'un horizontalement pour positionner les profils entre eux (comme dans le montage de Thrun pour modéliser les couloirs), l'autre verticalement pour obtenir des profils verticaux sur toute la hauteur. Ils parviennent ainsi à reconstruire des rues entières, tout en éliminant les objets en avant plan (arbres, automobiles, piétons, etc.) pour ne garder que les façades des bâtiments.

### 1.1.2 Critiques

De cette revue nous pouvons constater que la cartographie en bi-dimensionnelle des environnements intérieurs est un sujet qui arrive à maturité et dont les algorithmes de localisation et de cartographie simultanée sont devenus de plus en plus fiables (Lemaire, 2004) ; ces derniers apportent aussi une solution au problème de mise en correspondance des données en deux dimensions (Thrun, 2002).

En ce qui concerne la cartographie tridimensionnelle, le problème est plus complexe et les voies de recherche sont plus variées. La modélisation des environnements intérieurs (édifices, mines) aboutit sur la modélisation des objets constituant la scène (murs, plafond, sol, portes, etc.) (Surmann et al., 2003; Thrun et al., 2001; Martin and Thrun, 2002). Aussi ces modélisations utilisent les algorithmes bidimensionnels pour résoudre les problèmes de localisation. Par contre la cartographie des environnements extérieurs par des robots mobiles, plus particulièrement des terrains, est beaucoup moins traitée. Certaines équipes se basent sur des caméras stéréoscopiques mais la résolution des cartes obtenues n'est pas très élevée. Aussi les approches proposées sont toutes basées sur un traitement hors-ligne des données (Huber and Hebert, 1999; Surmann et al., 2003).

## 1.2 Notre système

### 1.2.1 Objectifs et problèmes

Nous proposons, dans cette étude, une plate-forme mobile autonome capable d'explorer un environnement inconnu, de reconstruire et de modéliser cet environnement en utilisant un télémètre profilométrique fixe monté à l'avant du véhicule. Une contrainte du système est de traiter les données en ligne pour ainsi pouvoir utiliser la modélisation pour la navigation de la plate-forme. Ces coupes, ou profils, de l'environnement pris tout au long de l'exploration sont alors rassemblées pour modéliser le terrain. Ainsi ce système doit être capable de cartographier les environnements extérieurs et des terrains variés et non structurés (non constitués de plans) ; il doit donc pouvoir se déplacer et se localiser sur ces terrains d'inclinaison variable.

Pour permettre à l'opérateur de superviser plusieurs systèmes et pallier aux éventuels problèmes de communication<sup>1</sup>, la plate-forme doit être capable de naviguer et d'explorer l'environnement sans intervention d'un opérateur, en utilisant donc la carte qu'il est en train de construire pour baser sa planification de trajectoire et détecter et éviter les obstacles.

Bien que le système soit autonome, il est important qu'un opérateur supervise le déroulement de l'exploration et puisse intervenir. Ceci implique des fonctionnalités supplémentaires telles que la visualisation de la carte au cours de sa construction ainsi que le positionnement du robot au milieu de celle-ci, la possibilité de contrôler le déplacement du robot, soit en lui donnant un point à atteindre, soit en contrôlant directement sa vitesse.

---

<sup>1</sup>Les problèmes de communications sont inévitables lors de l'exploration planétaire ou de galeries souterraines.

Concevoir un tel système explorant et cartographiant un environnement extérieur de façon précise présente de nombreuses difficultés :

- la *localisation* : la plate-forme doit être capable de déterminer sa position de façon précise dans l'environnement inconnu où il évolue pour placer avec précision les différentes mesures de l'environnement (par exemple les mesures d'un télémètre). Ce problème, autrement appelé *SLAM*, consiste d'une part à construire une carte d'un environnement inconnu et de se positionner dans cette carte en cours d'élaboration. La précision et la fiabilité de la carte résultante dépendent directement de la connaissance de la position du robot. Ce point est d'autant plus compliqué car le terrain où évolue la plate-forme n'est pas plat et il faut donc concevoir un système de positionnement en trois dimensions fournissant aussi l'attitude de la plate-forme (lacet, tangage et roulis).
- la *représentation* interne de l'environnement est aussi un point critique. La quantité importante de données nécessaires à la modélisation de l'environnement impose de choisir une structure de données particulièrement performante : elle doit permettre une grande résolution spatiale, paramétrable par l'opérateur, éliminer les données redondantes, tout en offrant un accès rapide aux données. D'autre part, pour faciliter la compréhension de l'environnement, elle doit être visualisable aisément.
- la *notion d'obstacle* : il est important de les repérer pour la navigation de la plate-forme mais il faut avant les définir dans le cas d'un robot mobile évoluant dans un environnement extérieur *non structuré*. Ces obstacles sont en effet de nature bien plus variée que dans le cas d'environnements intérieurs et cette notion est fortement liée aux capacités de déplacement du robot mobile. En plus des obstacles *traditionnels* dressés sur le parcours de la plate-forme tels les murs et autres objets contre lesquels elle pourrait venir buter, il faut aussi considérer les trous et ravins dans lesquels elle pourrait tomber et les zones mouvementées

entraînant une navigation chaotique.

- l'*exploration* autonome : pour réduire les interventions de l'opérateur et lui permettre de superviser l'évolution de plusieurs plate-forme, il est nécessaire qu'elles aient une certaine autonomie. Cela veut dire d'une part que les plates-formes sont capables de prendre en compte leur environnement dans leur navigation, c'est la navigation réactive, et d'autre part qu'elles embarquent des fonctionnalités d'exploration plus poussées comme détecter les zones à explorer, choisir celles susceptibles d'apporter le plus d'informations sur l'environnement ou encore visiter de façon exhaustive un site.

## 1.2.2 Architecture du système

### 1.2.2.1 Architecture fonctionnelle

Nous avons opté pour une architecture modulaire où chaque module fournit une fonctionnalité : *acquisition / odométrie 3D, modélisation, interface, navigation / exploration* et *contrôleur de déplacement*. L'architecture que nous proposons organise ces modules dans le but de créer un système d'exploration et de cartographie d'environnements inconnus non structurés mais il est aussi possible de réutiliser chaque bloc séparément pour en exploiter la fonctionnalité dans un autre système.

La figure 1.1 représente l'organisation de notre système par modules ainsi que leurs relations. Il est à noter que seuls les modules d'*acquisition* et du *contrôleur de déplacement* ont des liens avec le matériel, à savoir les capteurs et les actionneurs. Ainsi seuls ces deux blocs seraient à adapter pour changer de plate-forme ou de montage.

Voici une description des rôles de chacun des modules du système organisés selon

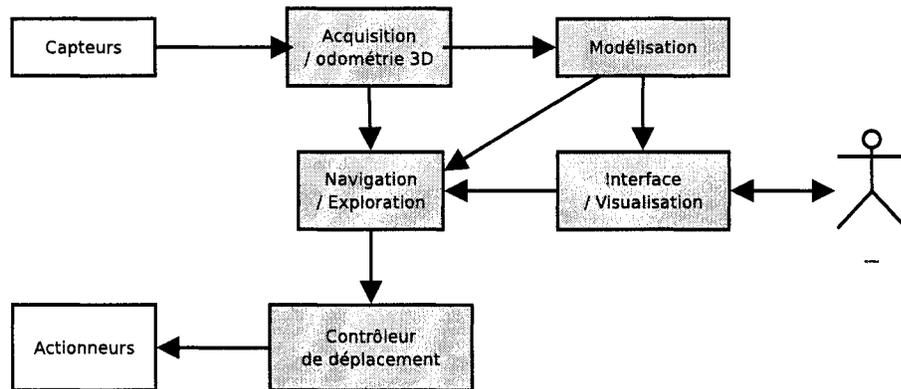


FIG. 1.1 Architecture générale

la figure 1.1 :

- Acquisition/odométrie 3D : ce module reçoit les données des différents capteurs. Grâce aux capteurs de positionnement (odométrie, gyromètre et inclinomètre dans notre cas), il déduit la position tridimensionnelle et l'attitude (tangage, roulis et lacet) du robot dans un repère global et peut ainsi positionner les points de la surface du sol acquis par le télémètre et en calculer la position dans le repère global.
- Modélisation : il comporte toute la structure de données qui sert à modéliser l'environnement. Ce module prend les points fournis par le module d'acquisition et les organise pour être facilement utilisés par les blocs de visualisation et de navigation.
- Navigation/exploration : à partir de la modélisation de l'environnement et de la position du robot dans cet environnement, ce module détermine les obstacles et en déduit la trajectoire que doit suivre le robot selon le type de navigation sélectionné par l'opérateur (navigation purement réactive, avec but ou bien détection des zones inexplorées).
- Visualisation : c'est l'interface entre le système et l'opérateur. Ce module permet d'une part à ce dernier de visionner la modélisation de l'environnement au cours de l'exploration et d'autre part de commander le robot, soit directement tel un

- véhicule téléguidé, soit lui indiquer quel mode d'exploration utiliser.
- Contrôleur de déplacement : ce dernier bloc assure que le robot se dirige bien selon la trajectoire fournie par le module de navigation.

Les principes de ces différents modules sont détaillés dans les chapitres suivants de ce mémoire : le chapitre 2 décrit la structure de données utilisée pour représenter l'environnement acquis et traite de la visualisation de l'environnement, le chapitre 3 décrit les algorithmes utilisés dans nos différents modes de navigation et d'exploration.

### 1.2.2.2 Point de vue logiciel

Une architecture de type client-serveur a été adoptée pour le système : au centre se trouve le programme roulant sur le robot qui s'occupe de rassembler les mesures des capteurs, de modéliser l'environnement à partir des points acquis, de planifier la trajectoire du robot à partir de cette modélisation et enfin de contrôler le robot. Sur ce programme peuvent venir se greffer, par l'intermédiaire de *sockets*, deux clients : l'un servant à contrôler le robot (lui indiquer comment avancer, quel mode d'exploration utiliser) et l'autre permettant de visualiser à distance et en temps réel la carte en cours de construction.

L'architecture du serveur reprend grandement celle proposée pour l'architecture fonctionnelle. La figure 1.3 montre plus en détail comment sont organisées et connectées les parties formant les modules de l'architecture générale. Nous pouvons encore remarquer que les parties bas-niveau traitant avec les capteurs et les actionneurs sont très réduites pour avoir une indépendance maximale vis-à-vis du matériel utilisé.

Nous avons aussi mis en place un mode hors-ligne qui permet de simuler les capteurs

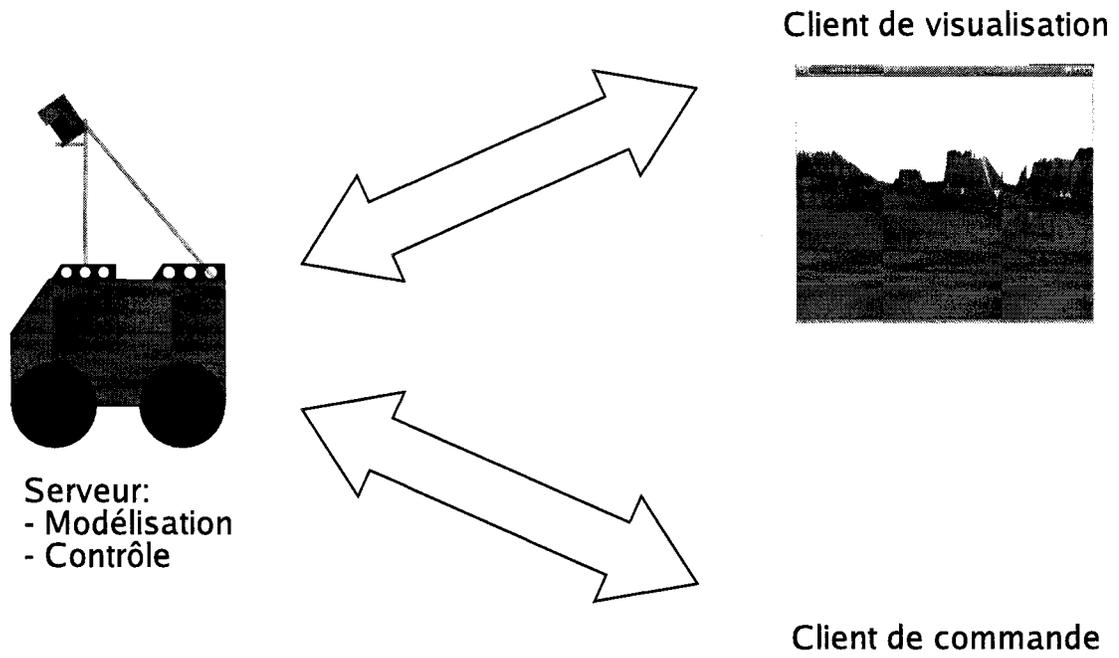


FIG. 1.2 Architecture de type client-serveur

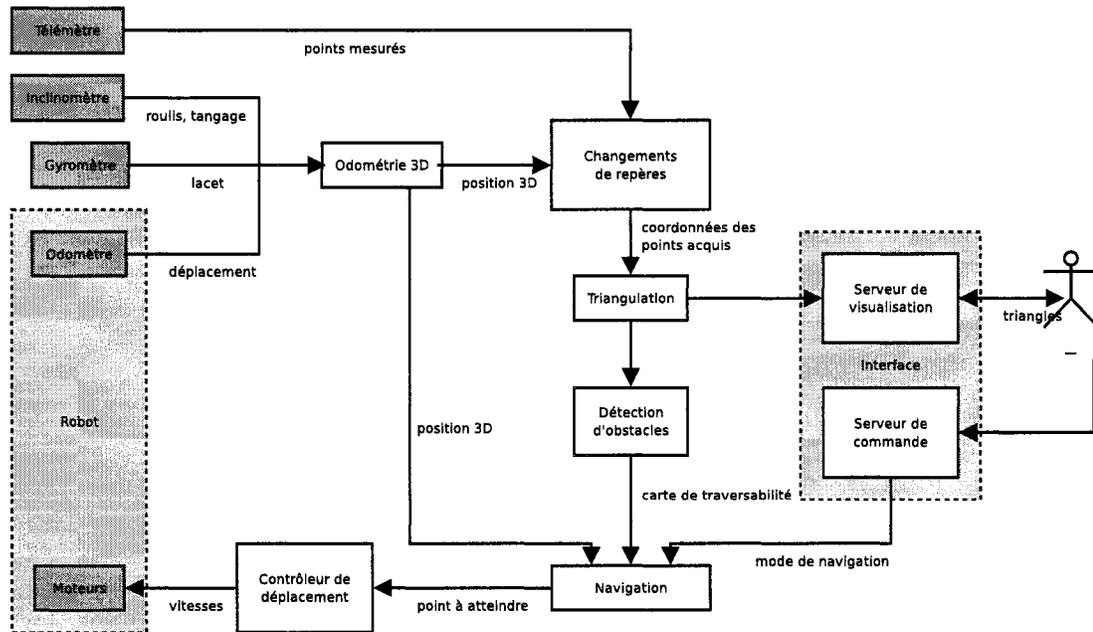


FIG. 1.3 Architecture détaillée

à partir d'un fichier de données d'un fichier prises lors d'une capture précédente. Ceci permet de tester les algorithmes sans avoir à déplacer le robot.

Enfin nous avons mis en place un système de fichiers de configuration qui permet de modifier aisément les paramètres de la plate-forme (configuration géométrique) et du système (paramètres des algorithmes).

### 1.2.2.3 Architecture matérielle

Notre montage est basé sur le robot mobile *ATRV-2* de la compagnie *Real World Interface* qui présente la particularité de pouvoir tourner sur lui-même. Nous supposons aussi qu'il se déplace sans glissement pour pouvoir se fier aux valeurs fournies par l'odomètre. Il est équipé d'un ordinateur bi-processeurs Pentium II cadencés à 330MHz sur lequel roule Linux. Une connexion sans-fil permet de le contrôler à distance et aux clients évoqués à la section précédente de se connecter.

Sur ce robot mobile est monté le télémètre laser LMS-200 de la compagnie allemande SICK (situé en haut de la plate-forme sur la figure 1.4). Tel que le montre la figure 1.5, il est orienté vers l'avant du robot mobile en faisant un angle de  $30^\circ$  avec l'horizontale et est placé à 1,15 m du sol environ. Il est utilisé dans le mode où il effectue un balayage sur  $180^\circ$  en prenant 181 point de mesure (un point par degré) et ce toutes les 100 ms.

Les capteurs de position et d'attitude utilisés sont l'odomètre fourni avec le robot pour connaître le déplacement, un gyromètre *Autogyro* de la compagnie *KVH* (figure 1.6) dont l'axe de sensibilité est placé parallèlement à la verticale du robot pour l'angle de lacet (ou l'azimut) et un inclinomètre *MD900-T* de la compagnie *Applied Geomechanics* avec un axe de sensibilité dans le sens longitudinal du robot pour obtenir son angle de roulis et l'autre perpendiculaire pour l'angle de tangage.

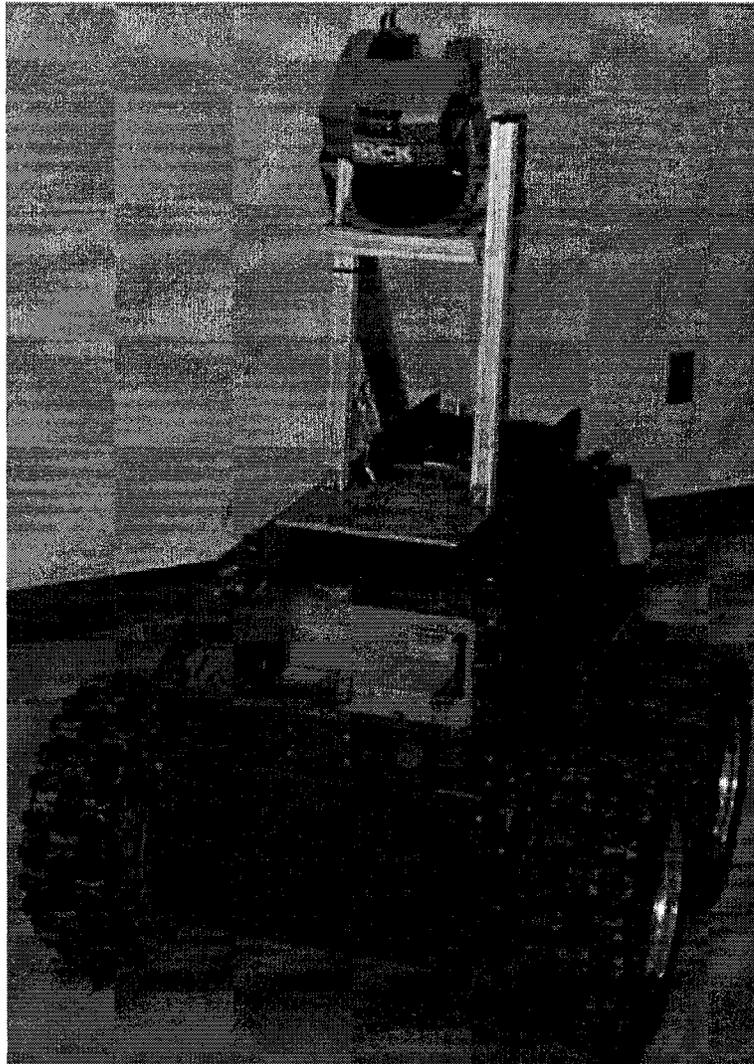


FIG. 1.4 Le robot ATRV utilisé avec le télémètre laser SICK (en bleu) dessus et l'inclinomètre et le gyromètre (à travers la vitre)

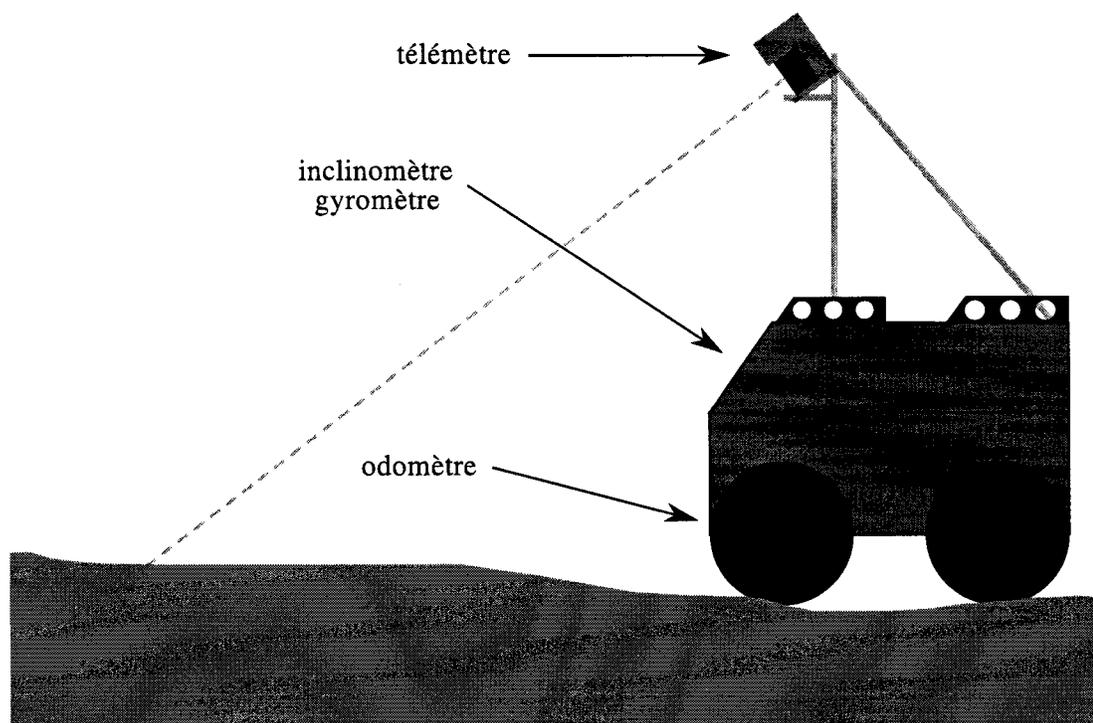


FIG. 1.5 Schéma du montage de l'ATRV

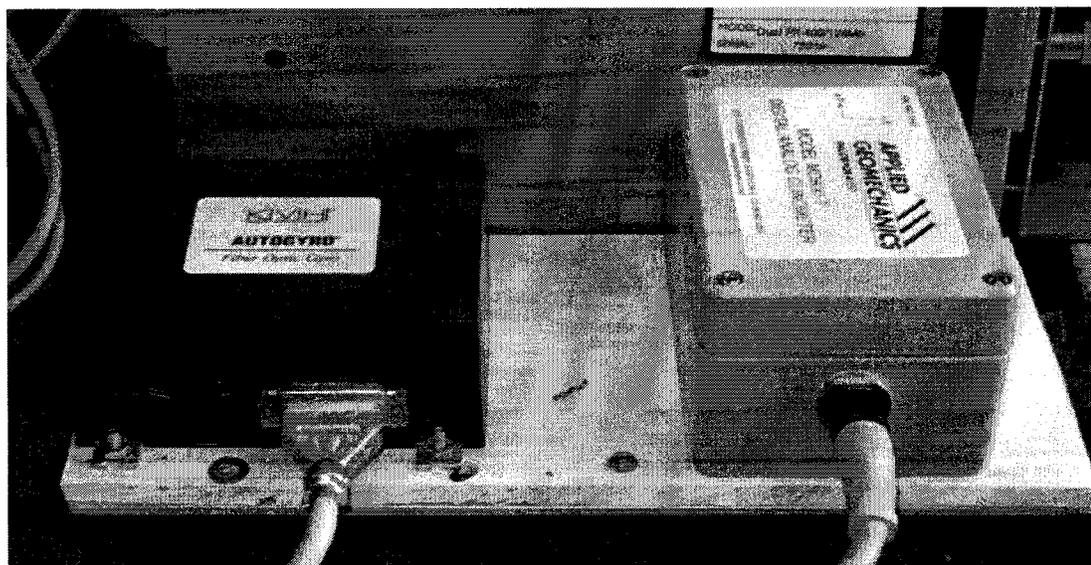


FIG. 1.6 Le gyromètre (à gauche) et l'inclinomètre (à droite) montés sur le robot ATRV

Nous pouvons remarquer que le capteur télémétrique balaie uniquement en avant du véhicule et ne peut donc pas détecter d'éventuels obstacles situés derrière lui. Nous avons donc imposé comme contrainte que le robot se déplace toujours vers l'avant.

La plate-forme ne scrute le terrain qu'à 2 m en avant d'elle ; cela implique que le contrôleur de navigation ne peut pas prévoir la trajectoire longtemps à l'avance car il ne peut connaître les obstacles situés à plus de 2 m en avant. Il faut donc replanifier la trajectoire régulièrement.

### 1.2.3 Scénario typique d'une exploration

Le scénario typique d'une mission d'exploration se présente de la manière suivante :

- l'opérateur règle tout d'abord les paramètres de la plate-forme (géométrie) et de la carte à construire (résolution, niveau de simplification) ;
- la plate-forme commence à avancer tout en acquérant des données sur le terrain pour le modéliser ; elle suppose tout d'abord qu'il n'y a pas d'obstacle devant elle car elle n'a pas assez de données pour modéliser le terrain et détecter les obstacles ;
- une fois qu'elle se situe sur une zone déjà modélisée, son module de navigation autonome se met en marche. L'opérateur a alors le choix entre différents modes d'exploration : manuel, exploration réactive ou encore exploration avec but (ce dernier est donné à l'aide de l'interface de visualisation). Bien entendu l'opérateur peut basculer d'un mode à l'autre ;
- une fois que la plate-forme a exploré une partie du site, l'opérateur peut lui donner l'ordre de se diriger vers une zone encore inconnue et de l'explorer ;
- en permanence l'opérateur peut contrôler le déroulement de l'exploration grâce à la visualisation en temps réel de la modélisation de l'environnement en cours

- et de la plate-forme au sein de celui-ci ;
- à la fin de l'exploration l'opérateur peut sauvegarder la carte construite et ainsi visionner le terrain ultérieurement.

Après une revue bibliographique présentant l'état de l'art sur le sujet, nous avons défini les objectifs et les problèmes que doit résoudre notre système. Nous avons ensuite présenté l'architecture mise œuvre, aussi bien sur le plan fonctionnel, informatique que matériel. Reste maintenant à présenter le fonctionnement de chacun des modules : la cartographie et la navigation.

## CHAPITRE 2

### CARTOGRAPHIE

La constitution de la carte constitue l'étape centrale du système. Elle consiste à rassembler et organiser les données télémétriques acquises dans une structure de données permettant d'utiliser efficacement la modélisation du terrain lors de recherche de hauteur de points ou de visualisation.

En rassemblant les profils acquis par la plate-forme au cours de son déplacement, le système peut modéliser la surface du terrain. Pour cela, il faut tout d'abord calculer les coordonnées des points de ces profils dans un repère global en utilisant la position et l'attitude de la plate-forme. Puis, nous avons choisi d'utiliser une structure de triangulation pour les rassembler et les organiser. D'autre part, des filtres ont été développés pour améliorer la modélisation du terrain. Enfin, la visualisation permet à l'opérateur de superviser le processus de modélisation.

#### 2.1 Acquisition des données télémétriques

Ce terme désigne l'étape préliminaire à la construction de la carte : obtenir les coordonnées des points des profils acquis par le télémètre dans un repère lié au terrain que l'on appellera *repère global*. Ceci se fait en deux étapes :

1. calculer la position et l'attitude du robot à chaque mesure pour connaître la position du capteur télémétrique ;
2. calculer ensuite la position des points fournis par le télémètre dans le repère global en utilisant la position du robot calculée à l'étape 1.

### 2.1.1 Positionnement du robot/odométrie 3D

Comme le système doit fonctionner dans des environnements non structurés, le robot doit être capable de déterminer sa position tridimensionnelle. Pour cela, il faut compléter les informations de base données par l'odomètre (qui permettent de connaître le déplacement sur la surface visitée) par des informations renseignant sur l'attitude du robot (tangage et roulis) et ainsi obtenir sa trajectoire tridimensionnelle.

D'autre part, les données fournies par l'odomètre se révèlent peu fiables, particulièrement en terme de lacet (ou azimuth). Le déplacement longitudinal est utilisé tel quel en négligeant le glissement, mais un gyromètre permet de remédier à ce manque de précision en orientation et de remplacer l'information angulaire fournie par l'odomètre.

La trajectoire est calculée en effectuant l'intégrale de mouvements élémentaires que nous assimilons à des vecteurs. En pratique, ces mouvements correspondent au déplacement entre deux prises de mesures.

Il est important de remarquer que nous considérons les angles de roulis et de tangage comme étant petits (en pratique ils le sont car l'inclinomètre ne fournit que des angles compris entre  $-25^\circ$  et  $25^\circ$ ) et que nous négligeons les effets des accélérations sur les mesures de l'inclinomètre. Ceci permet alors de faire l'approximation suivante : l'angle fourni par le gyromètre peut être considéré comme l'azimut du robot (en négligeant les angles de tangages et de roulis). Une étude précédente du laboratoire utilisait une odométrie basée sur les angles exacts de roulis et de tangage sans les approximations que nous venons d'effectuer (Bakambu et al., 2000).

L'attitude du robot est définie grâce à trois angles comme le montre la figure 2.1 :  
–  $\theta$  : le lacet (encore appelé azimuth ou *yaw* en anglais) qui représente l'orientation

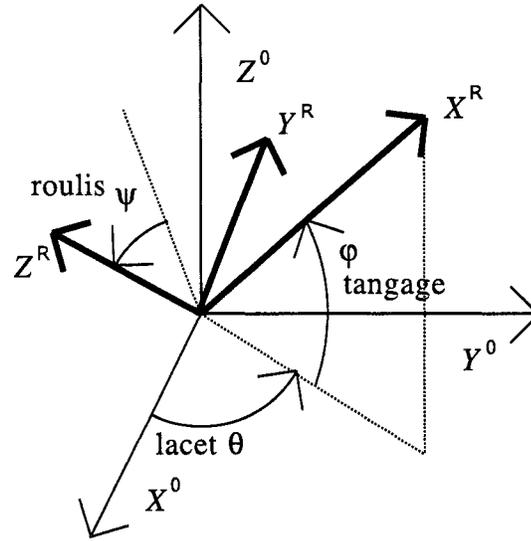


FIG. 2.1 Définition de l'attitude du robot grâce aux angles d'Euler

du robot selon l'axe vertical,

- $\phi$  : le tangage (*pitch* en anglais) qui représente l'inclinaison du robot selon l'axe transversal,
- $\psi$  : le roulis (*roll* en anglais) qui représente l'inclinaison du robot selon l'axe longitudinal.

L'odomètre donne le déplacement élémentaire  $dp$  du robot (l'indice  $i$  représente le numéro de la mesure) (voir la figure 2.2) :

$$dp_i = \sqrt{(dx_i^{odo} - dx_{i-1}^{odo})^2 + (dy_i^{odo} - dy_{i-1}^{odo})^2} \quad (2.1)$$

Nous pouvons remarquer que dans cette formule le déplacement est toujours positif ; en effet, ceci concorde bien avec notre contrainte qui impose que le robot progresse toujours vers l'avant. Ainsi, ce sont les angles de lacet et de roulis qui déterminent entièrement le sens et la direction du vecteur de déplacement. Ce vecteur est tangent à la trajectoire au point concerné.

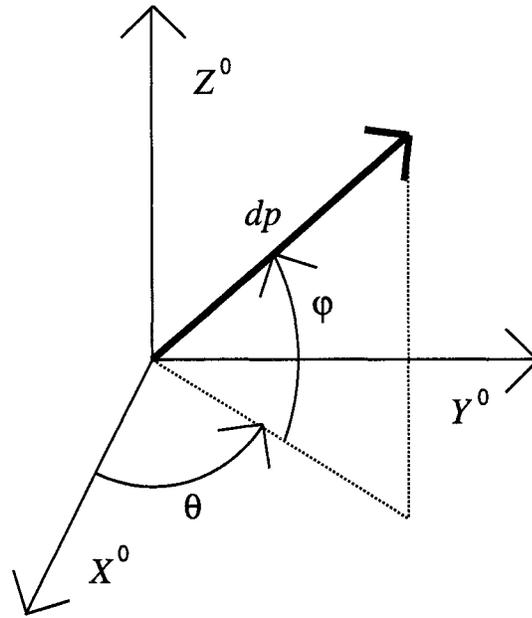


FIG. 2.2 Caractérisation du vecteur de déplacement élémentaire

Le vecteur de déplacement élémentaire s'écrit alors dans le repère global :

$$\vec{V}_i^0 = dp_i \begin{bmatrix} \cos \theta_i \cos \phi_i \\ \sin \theta_i \cos \phi_i \\ \sin \phi_i \end{bmatrix} \quad (2.2)$$

Ainsi la position du robot est donnée par la formule suivante :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_i^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{i-1}^0 + dp_i \begin{bmatrix} \cos \theta_i \cos \phi_i \\ \sin \theta_i \cos \phi_i \\ \sin \phi_i \end{bmatrix} \quad (2.3)$$

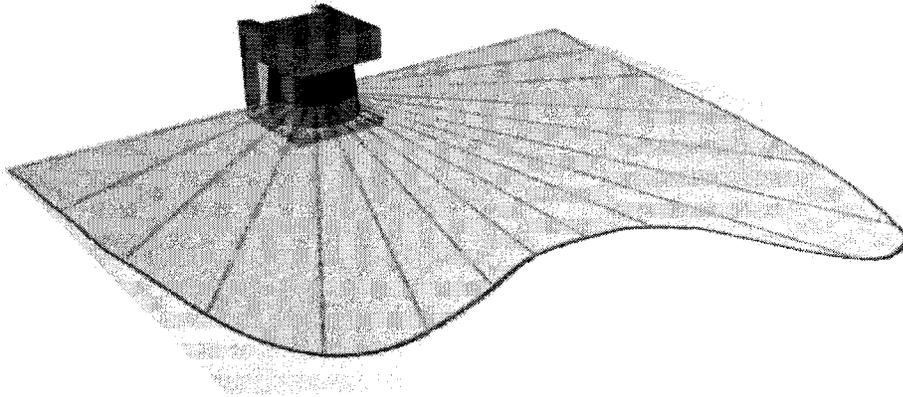


FIG. 2.3 Situation du plan de capture du télémètre

### 2.1.2 Changements de repères

Comme le montre la figure 2.3, les points acquis par le télémètre se situent tous dans un plan fixe par rapport au télémètre. Il est alors nécessaire d'effectuer quelques calculs de changement de repère pour obtenir les points dans le repère global. Pour ce, nous avons recours à deux repères intermédiaires : un repère lié au télémètre et un autre lié au robot.

La position d'un point acquis par le télémètre est définie grâce à deux valeurs comme l'illustre la figure 2.4 :

- l'angle  $\beta$  du rayon ;
- la valeur  $l$  de la distance mesurée.

Les coordonnées  $(x_t, y_t)$  du point correspondant peuvent alors être calculées dans le repère du télémètre de la manière suivante :

$$\begin{cases} x_t = l \cos \beta \\ y_t = l \sin \beta \end{cases} \quad (2.4)$$

La figure 2.5 permet de définir la position du télémètre laser par rapport au robot.

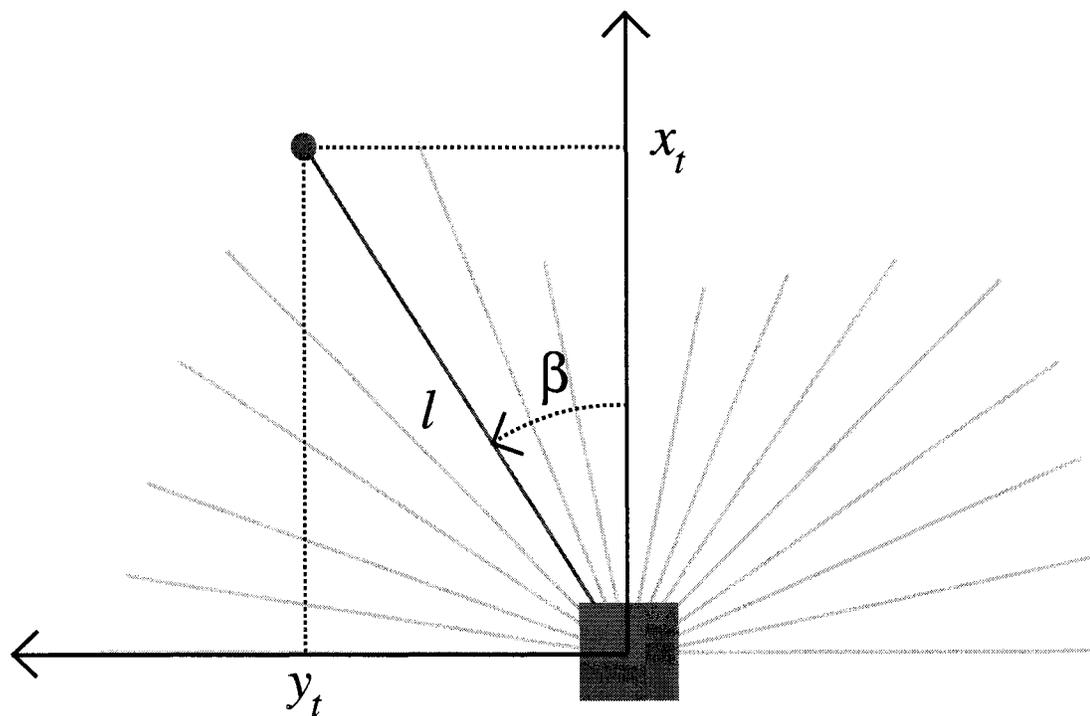


FIG. 2.4 Définition d'un point dans le repère du télémètre

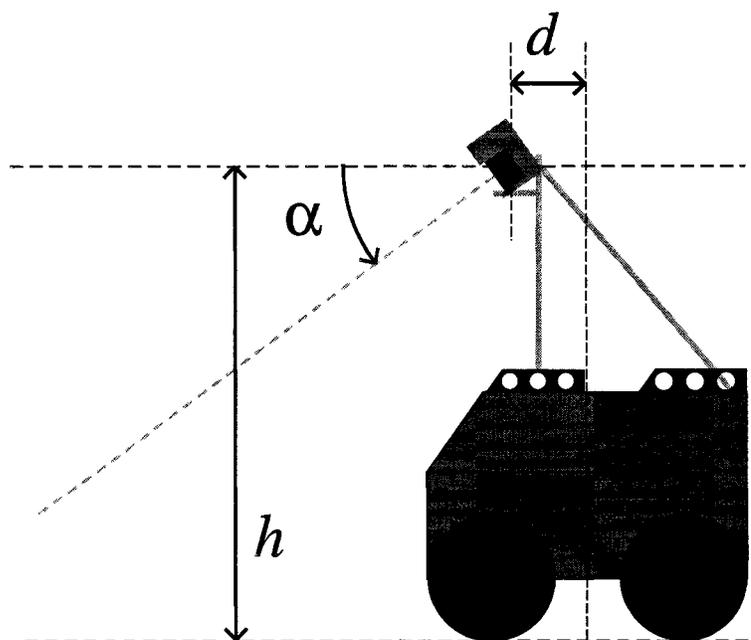


FIG. 2.5 Positionnement du télémètre par rapport au robot

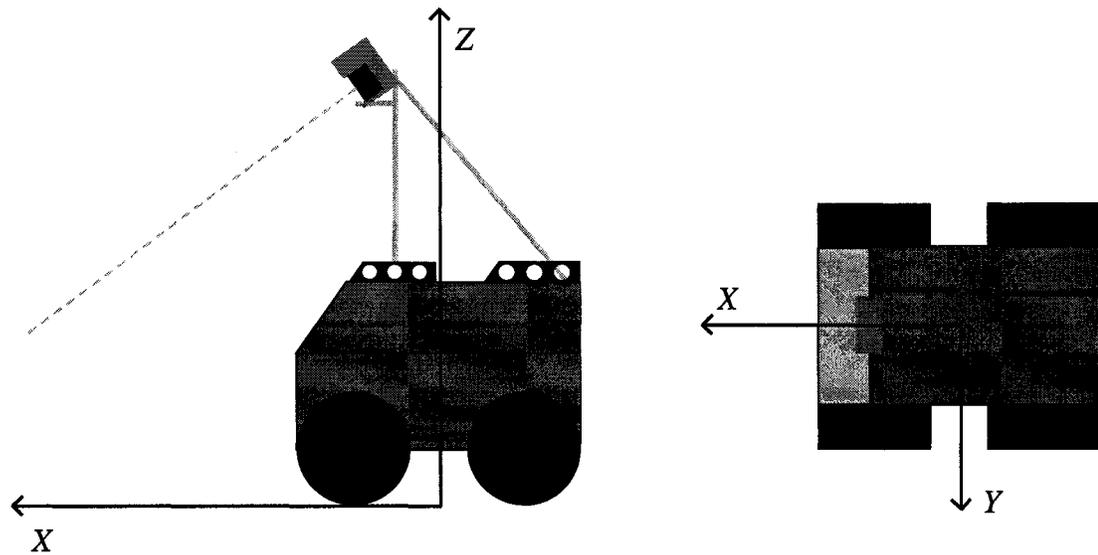


FIG. 2.6 Définition du repère lié au robot

Ainsi nous pouvons définir les paramètres suivants :

- $h$  la hauteur par rapport à la base des roues ;
- $d$  le décalage vers l'avant par rapport au centre du robot (situé au milieu du rectangle formé par les quatre roues) ;
- $\alpha$  l'angle d'inclinaison du plan de mesure par rapport à l'horizontale du robot.

Les coordonnées des points acquis dans le repère du robot s'écrivent alors (figure 2.6) :

$$\begin{cases} x_r = d + x_t \cos \alpha \\ y_r = y_t \\ z_r = h - x_t \sin \alpha \end{cases} \quad (2.5)$$

soit,

$$\begin{cases} x_r = d + l \cos \beta \cos \alpha \\ y_r = l \sin \beta \\ z_r = h - l \cos \beta \sin \alpha \end{cases} \quad (2.6)$$

Il ne reste plus qu'à utiliser la position et l'attitude du robot calculées dans la

section 2.1.1 pour obtenir les coordonnées des points dans le repère global (  $\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$  représente la position du robot dans le repère global) :

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + R \cdot \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} \quad (2.7)$$

avec la matrice de rotation  $R$  :

$$R = \begin{bmatrix} \cos \theta \cos \phi & -\sin \theta \cos \psi + \cos \theta \sin \phi \sin \psi & \sin \theta \sin \psi + \cos \theta \sin \phi \cos \psi \\ \sin \theta \cos \phi & \cos \theta \cos \psi + \sin \theta \sin \phi \sin \psi & -\cos \theta \sin \psi + \sin \theta \sin \phi \cos \psi \\ -\sin \phi & \cos \phi \sin \psi & \cos \phi \cos \psi \end{bmatrix} \quad (2.8)$$

Nous avons donc maintenant les coordonnées des données acquises par le télémètre calculées dans le repère global. Il reste maintenant à les regrouper, c'est-à-dire déterminer quelle structure utiliser pour les organiser et modéliser l'environnement.

## 2.2 Structure de triangulation

Comme l'environnement à modéliser est le terrain sur lequel évolue le robot, nous postulons qu'il peut se représenter par une surface isomorphe au plan horizontal par simple projection selon l'axe vertical. Ainsi, les points peuvent être projetés sur le plan horizontal avant d'être organisés. Cela permet d'utiliser une structure de donnée bidimensionnelle, beaucoup plus simple à mettre en œuvre et plus rapide qu'une structure tridimensionnelle. Nous associons par la suite une hauteur à chaque point de ce plan pour obtenir la troisième dimension. Ceci peut donc être

qualifié de représentation en 2D 1/2 de l'environnement.

Pour représenter les données, différentes structures de données sont possibles :

- une grille d'occupation qui consiste à quadriller le terrain avec la résolution souhaitée puis à insérer les points acquis dans les cases correspondantes. Cette solution permet d'accéder rapidement aux points du fait de sa structure tabulaire, mais est peu économe en mémoire car, dans notre cas, la majeure partie de l'espace est vide (voir la figure 2.7 pour un exemple bidimensionnel). Cette structure est aussi appelée carte d'élévation ;
- un *oct-tree* (dont le principe est connu sous le nom de *quad-tree* dans sa version bidimensionnelle) qui consiste à établir une approche hiérarchique et multi-résolution des zones occupées et libres. La figure 2.8 illustre ce principe. Cette approche est notamment utilisée par l'équipe de Surmann (Surmann et al., 2003) pour la représentation de son volume balayé ;
- une structure de *triangulation* qui découpe la surface en triangles et ainsi former un maillage triangulaire comme le montre la figure 2.9. Les données acquises sont alors représentées par les sommets des triangles.

Nous avons décidé de choisir la dernière solution qui présente l'avantage de conserver les points tels qu'ils ont été acquis sans approximer leur position comme dans les autres cas.

## 2.2.1 Présentation de la triangulation de Delaunay

### 2.2.1.1 Principe

Une triangulation est un maillage permettant de mettre en relation un ensemble de points. Différentes structures sont alors définies comme indiqué par la figure 2.9 : les *sommets* sont reliés entre eux par des *arêtes* formant ainsi des *faces* triangulaires.

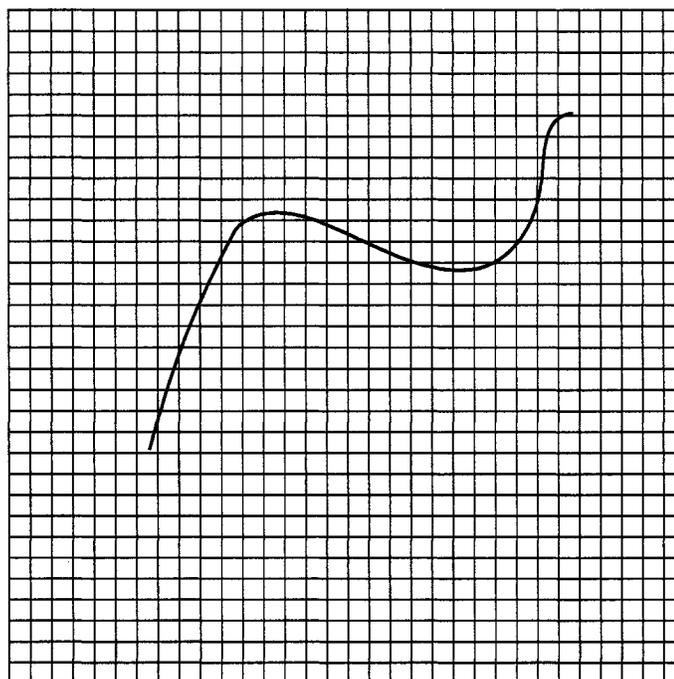
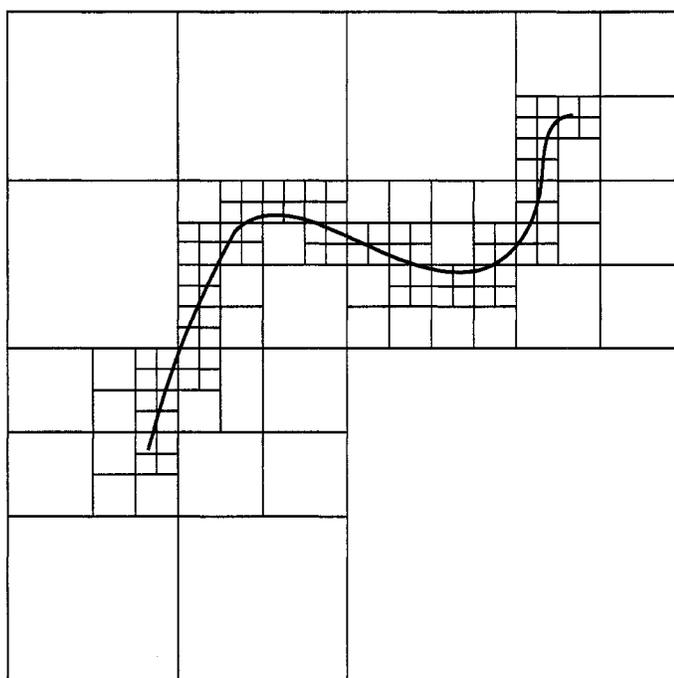


FIG. 2.7 Grille d'occupation

FIG. 2.8 Principe des *quad-trees* : approximation d'une courbe

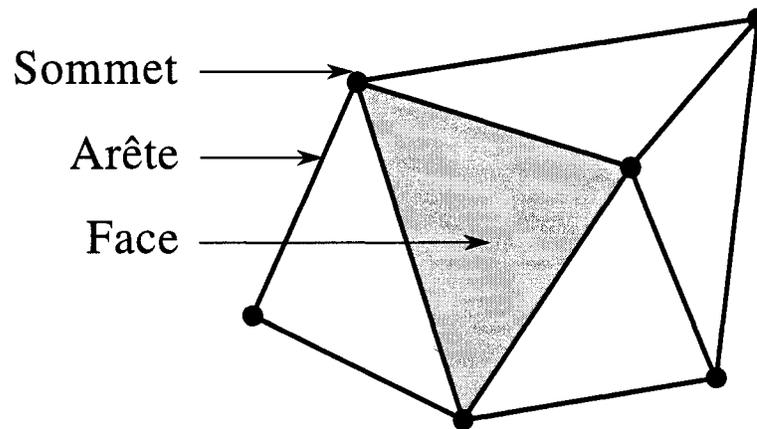


FIG. 2.9 Définition des éléments d'une triangulation

En général ces trois objets sont définis dans les structures de données et sont reliées entre eux par l'intermédiaire de pointeurs :

- les sommets sont reliés aux faces et aux arêtes dont ils font partie ;
- les faces sont reliés à leurs trois sommets ainsi qu'aux trois arêtes les bordant ;
- les arêtes sont reliées aux deux sommets les délimitant ainsi qu'aux deux faces situées de part et d'autre.

Il existe de nombreuses façons de trianguler un ensemble de points. Aussi nous proposons d'utiliser la *triangulation de Delaunay* car elle permet d'obtenir des triangles plus réguliers qu'une triangulation quelconque (George and Borouchaki, 1997). Cette triangulation respecte la règle suivante : aucun point n'est situé à l'intérieur du cercle circonscrit à chaque triangle. Cette triangulation existe, est unique (sauf si plus de trois points sont co-circulaires) et son dual est le diagramme de Voronoï qui sépare le plan en cellules de proximité (figure 2.11).

### 2.2.1.2 Bibliothèques

Pour mettre en œuvre cette structure de triangulation, nous avons étudié plusieurs bibliothèques libres disponibles librement sur Internet :

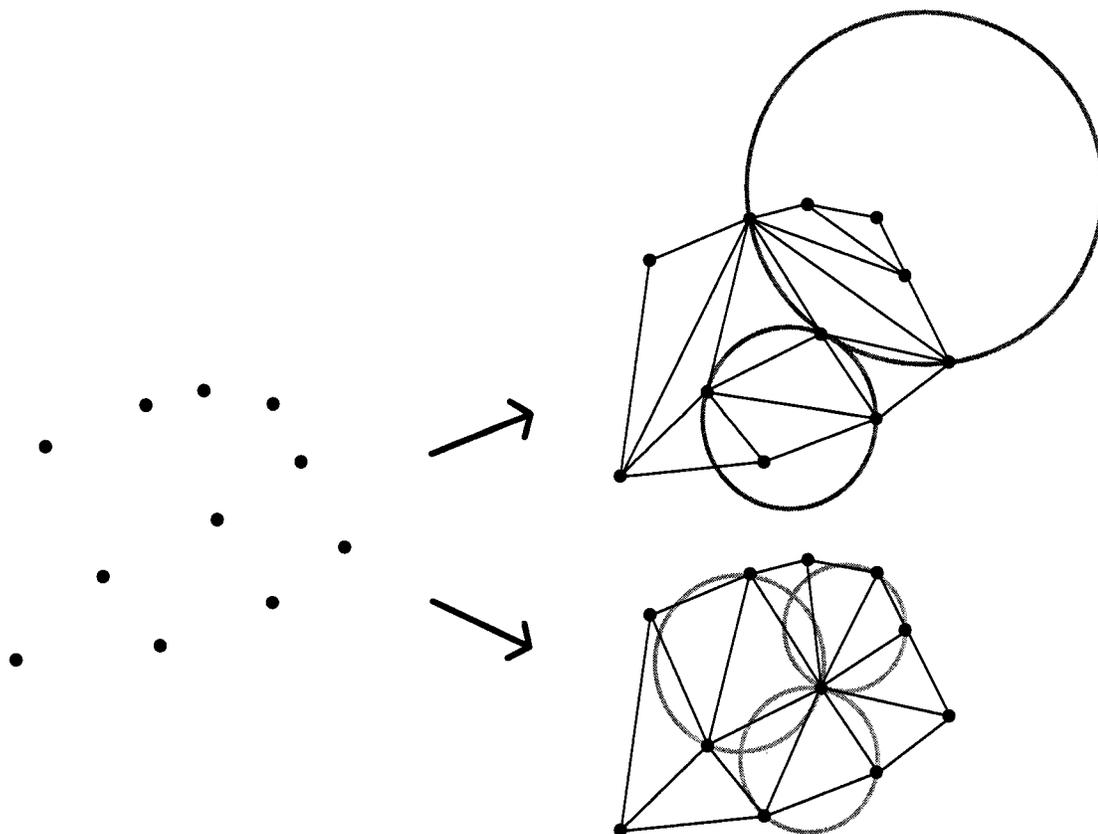


FIG. 2.10 Triangulation d'un ensemble de points : en haut une triangulation quelconque et en bas la triangulation de Delaunay (les cercles permettent de vérifier le critère de Delaunay)

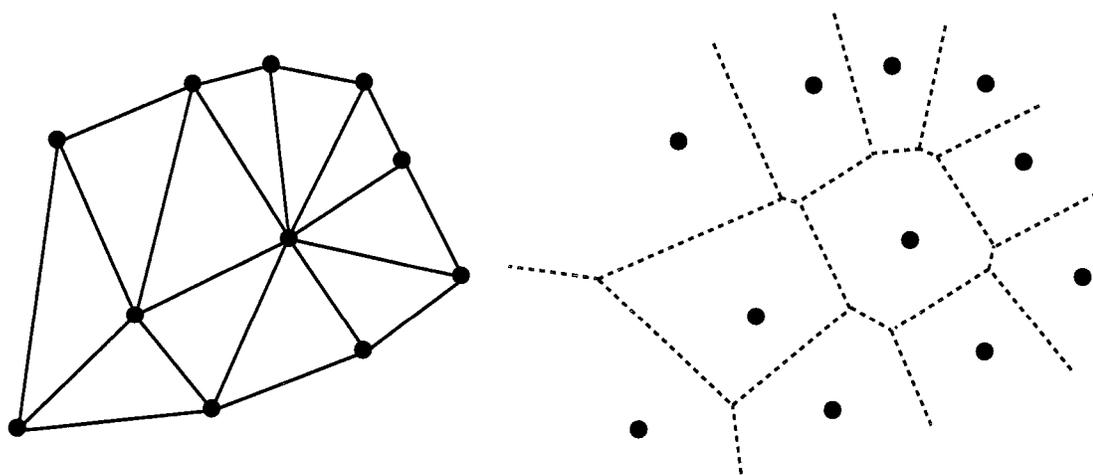


FIG. 2.11 Triangulation de Delaunay et à gauche son dual, le diagramme de Voronoï

- Qhull créée par C. Bradford Barber et Hannu Huhdanpaa de l'université du Minnesota (disponible sur <http://www.qhull.org>);
- Hull développée par les laboratoires Bell (disponible sur <http://cm.bell-labs.com/netlib/voronoi/hull.html>);
- CGAL (*Computational Geometry Algorithms Library*) développée par un ensemble d'organismes de recherche dont l'INRIA (disponible sur <http://www.cgal.org>);
- GTS (*GNU Triangulated Surface Library*) créée par Stéphane Popinet (disponible sur <http://gts.sourceforge.net>).

Les deux premières structures ne permettant pas la suppression de points ont été écartées. Nous avons choisi d'utiliser la librairie CGAL (Boissonnat et al., 2000) pour sa richesse et sa flexibilité. D'autre part, cette librairie possède un mode de triangulation projetant les points sur un plan tout en conservant les coordonnées tridimensionnelles des points, ce qui correspond exactement à notre situation.

### 2.2.1.3 Algorithmes

CGAL utilise une représentation basée sur les sommets et les faces mais n'utilise pas la notion d'arête : chaque point possède des pointeurs vers les faces auxquelles il appartient et réciproquement chaque face pointe vers ses sommets. Les opérations d'insertion et de suppression de points modifient ces liens de façon à garder la structure cohérente et selon différentes contraintes (contraintes de Delaunay dans notre cas).

L'algorithme central est celui de localisation qui consiste à retrouver le triangle contenant un point quelconque (faisant partie ou non de la triangulation) dont on donne les coordonnées. Un algorithme classique est le *line walk*. Il consiste à partir

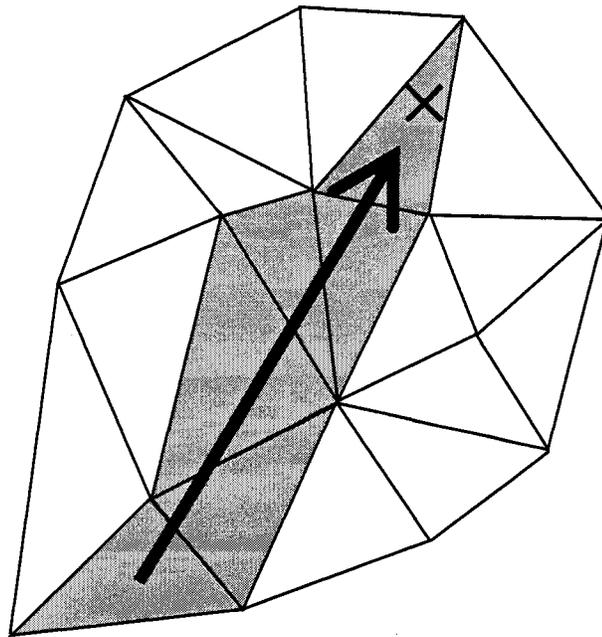


FIG. 2.12 Principe du *line-walk*

d'un triangle (quelconque ou bien supposé proche du point visé) puis à se déplacer de triangle en triangle voisin pour se rapprocher et atteindre le triangle contenant le point objectif comme le montre la figure 2.12.

L'algorithme implémenté par CGAL, le *jump and walk*, en est une version optimisée. Pour réduire le temps de parcours de la structure, il faut partir d'un triangle déjà proche du triangle final; pour ce, l'algorithme se base sur un petit échantillon aléatoire des points de la triangulation et détermine naïvement quel est le point de ce sous-ensemble le plus proche du point visé. Ensuite l'algorithme *line walk* est appliqué à partir d'un triangle adjacent à ce point pour atteindre le triangle contenant le point visé.

L'insertion d'un point se fait en localisant dans quel triangle il se trouve grâce à l'algorithme décrit précédemment, à insérer le point dans le triangle en créant les liens vers ses voisins, puis en basculant (voir la figure 2.13) les faces voisines pour

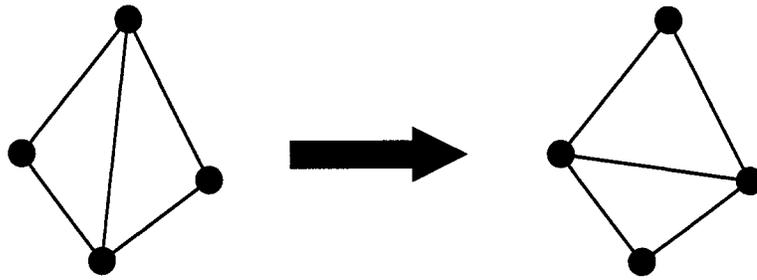


FIG. 2.13 Principe du basculement des arêtes

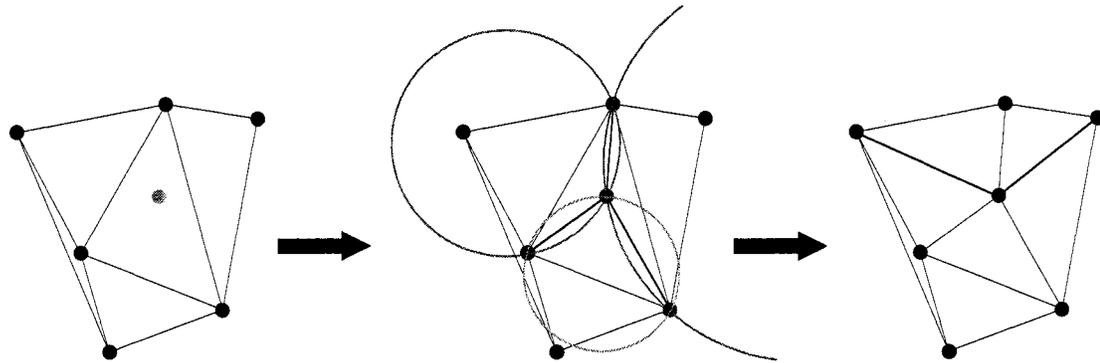


FIG. 2.14 Insertion d'un point et rétablissement du critère de Delaunay par basculement des arêtes

que la triangulation vérifie à nouveau la condition de la triangulation de Delaunay (voir la figure 2.14).

La suppression est une opération plus rapide que l'insertion car il est inutile d'effectuer l'étape de localisation puisque le sommet à supprimer est connu. Pour ce il suffit juste de supprimer le sommet et de rediriger les arêtes vers un sommet voisin de celui supprimé, puis de rétablir le critère de Delaunay par basculement des faces.

### 2.3 Filtrage des données

Si les points sont insérés tels quels dans la structure de triangulation, la structure résultante devient très grande, peut-être beaucoup trop pour la résolution requise,

ce qui rallonge les temps de manipulation de la structure. De plus, de nombreux points sont alors redondants, soit parce qu'ils sont très rapprochés, soit parce qu'ils sont coplanaires. Il est donc nécessaire de filtrer les points à insérer.

Deux méthodes de filtrage ont été mises en œuvre :

- un filtrage spatial de proximité pour supprimer les points très rapprochés ;
- un algorithme de simplification de surface pour supprimer les points coplanaires.

### 2.3.1 Filtrage spatial de proximité

Ce premier filtre a comme objectif de limiter le nombre de points de la structure de triangulation et de l'adapter à la résolution souhaitée de la carte. Pour cela, l'opérateur fixe la distance minimale entre deux points.

Avant d'insérer un point  $P$  dans la structure, le point le plus proche est déterminé, soit  $A$  avec un poids  $a$ . Si ces deux points sont trop proches, à savoir leur distance est inférieure au seuil fixé par l'utilisateur, alors ces deux points sont fusionnés pour leur barycentre pondéré  $B$  par le poids  $b$ .

$$\left\{ \begin{array}{l} x_B = \frac{a \cdot x_A + x_P}{a + 1} \\ y_B = \frac{a \cdot y_A + y_P}{a + 1} \\ z_B = \frac{a \cdot z_A + z_P}{a + 1} \\ b = a + 1 \end{array} \right. \quad (2.9)$$

Ce filtre présente l'avantage de permettre à l'utilisateur de régler la résolution (distance moyenne entre les points) du maillage en réglant la valeur du seuil de proximité. D'autre part, il joue aussi le rôle d'un filtre passe-bas en fusionnant les points.

Seuil (cm)	Scène intérieure	Scène extérieure (chemin)	Scène extérieure (butte)
0.5	43%	35%	42%
1	28%	24%	30%
2	17%	16%	20%
5	9%	9%	11%

TAB. 2.1 Part de points restant après application du filtre de proximité en fonction du seuil et du type de terrain modélisé

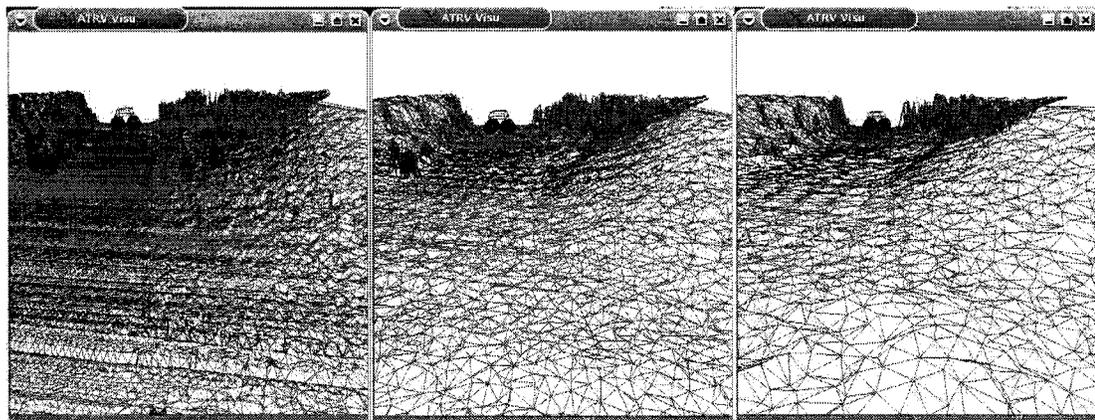


FIG. 2.15 Influence du filtre de proximité sur un scène extérieure : sans filtre (à gauche), avec un seuil de 2 cm (au centre) puis de 5cm (à droite)

Le tableau 2.1 montre l'influence du type de terrain modélisé et de la distance minimale entre deux points sur le nombre de points restant après application du filtre de proximité. Nous pouvons donc constater que le type de terrain n'a pas beaucoup d'effet sur le nombre de points supprimés. Ceci s'explique car la densité des points est relativement indépendante du type de terrain. Aussi, en réglant le seuil de proximité nous pouvons évaluer la proportion de points supprimés.

### 2.3.2 Filtrage de simplification

La simplification de surface est utilisée principalement dans l'infographie pour réduire le nombre de sommets d'un modèle 3D en l'altérant le moins possible. Ainsi

par exemple une zone lisse peut être dépourvue d'un grand nombre de points sans perte de résolution tandis que des zones anguleuses doivent conserver tous leurs points. Nous nous sommes basés sur ce principe pour supprimer des points inutiles de notre modèle. (Heckbert and Garland, 1997)

L'algorithme de base de la simplification de surface consiste à éliminer les points ayant le moins d'importance dans la description de la surface. Il faut donc aussi décider d'une mesure de l'erreur ; c'est en général la distance moyenne d'un point au plan moyen voisin ou bien une mesure qui s'en rapproche. On peut donc associer à cette erreur une notion d'*utilité* d'un point : un point est d'autant plus utile qu'il se démarque des autres, et donc qu'il engendrerait une erreur plus grande si on l'éliminait.

L'algorithme est le suivant :

1. calculer pour chaque point l'erreur induite s'il était enlevé ;
2. classer les points selon cette erreur ;
3. tant que l'erreur minimale est inférieure au seuil :
  - retirer le point de la triangulation ;
  - mettre à jour la liste des points classés selon leur erreur.

Il existe différents types d'erreur que nous pouvons utiliser :

- le maximum de la valeur absolue de la différence de hauteur entre le point inséré et ses voisins :

$$e(p) = \max_{v \in V(p)} (|z_p - z_v|) \quad (2.10)$$

- la valeur absolue de moyenne de la différence de hauteur entre le point inséré et ses voisins <sup>1</sup> :

$$e(p) = \left| \frac{\sum_{v \in V(p)} (z_p - z_v)}{\text{card}(V(p))} \right| \quad (2.11)$$

---

<sup>1</sup>La fonction  $\text{card}(E)$  retourne la cardinalité de l'ensemble  $E$ , soit son nombre d'éléments.

Type d'environnement	Terrain	% restant
Intérieur	laboratoire (figure 2.16)	32.6
Extérieur	butte	35.7
	canyon (figure 2.17)	39.7
	forêt	42.4
	forêt	55.2

TAB. 2.2 Pourcentage de points restants après simplification de la modélisation en fonction du type de terrain (les seuils des filtres de proximité et de simplification sont fixés à 2 cm)

- la distance du point au plan moyen créé pas ses voisins, c'est cette dernière mesure qui est le plus souvent utilisée dans la littérature, mais elle est plus coûteuse en calculs :

$$e(p) = d(p, P(V(p))) \quad (2.12)$$

Dans la mise en œuvre proposée, la deuxième mesure de l'erreur a été utilisée.

Cet algorithme est appliqué à notre structure de données à des moments indiqués par l'utilisateur. En effet, son déroulement prend un certain temps (quelques secondes selon la taille de la structure) et ne peut donc être exécuté en permanence sans perturber le système. Le meilleur moment pour le lancer est à la fin de la capture, une fois que tout l'environnement a été modélisé.

Cet algorithme permet de réduire considérablement le nombre de points sans nuire à la qualité de la carte reconstruite. La figure 2.16 montre comment une scène intérieure a été simplifiée : le nombre de points constituant le sol (plat) est devenu minime mais les objets et les murs gardent toute leur résolution. La figure 2.17 présente une scène extérieure simplifiée, les zones lisses y ont perdu un grand nombre de points sans en perdre leur forme. Le tableau 2.2 présente le pourcentage de points restants après simplification pour différents types de terrain modélisés. Nous pouvons remarquer que les terrains extérieurs sont moins simplifiés que les terrains intérieurs car ils sont moins structurés.

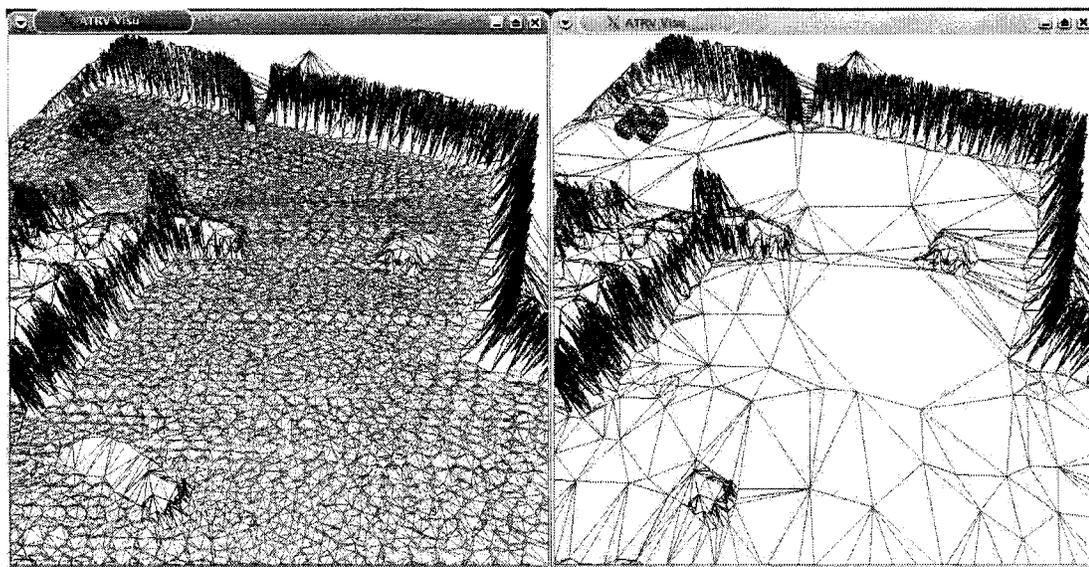


FIG. 2.16 Exemple de simplification d'une scène intérieure

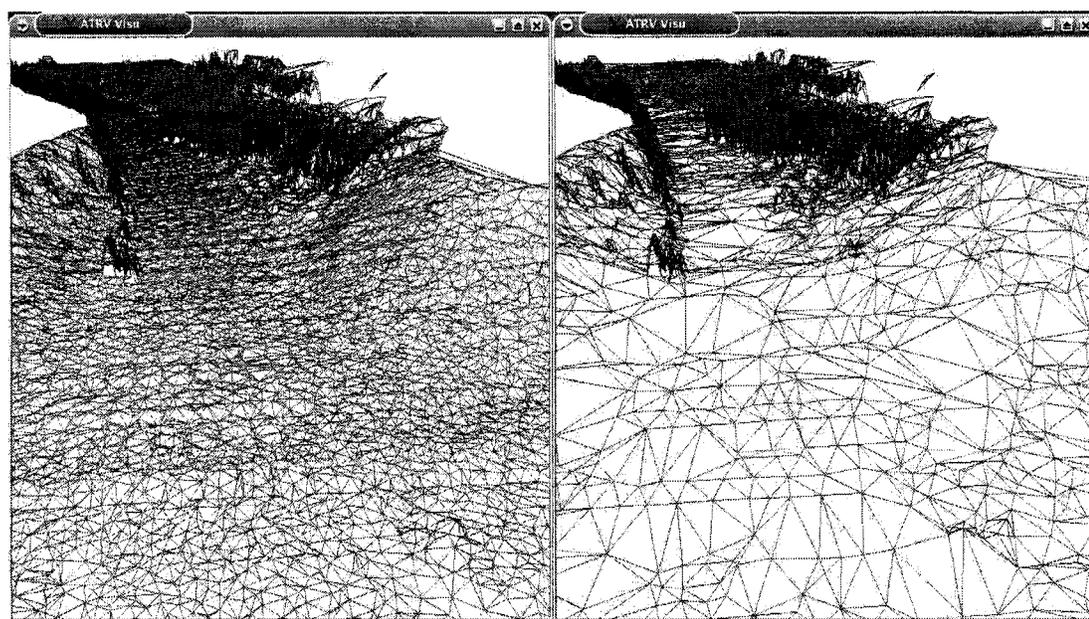


FIG. 2.17 Exemple de simplification d'une scène extérieure

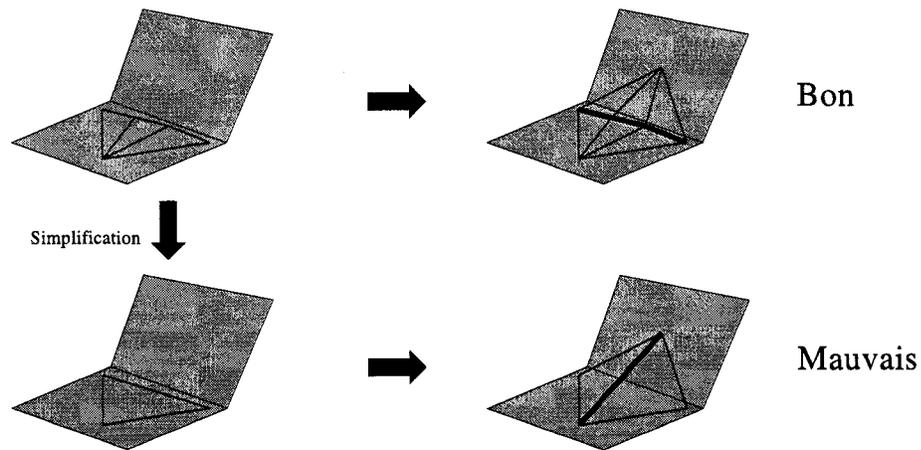


FIG. 2.18 Exemple d'effet néfaste du rajout d'un point après une simplification en ligne

Nous avons essayé d'adapter cet algorithme pour le faire opérer en ligne et ainsi ne plus attendre d'avoir acquis toute la surface pour ensuite la simplifier. Cela permet de manipuler une structure plus légère car elle contient moins de sommets.

Pour cela, nous avons testé l'algorithme suivant : lors de l'insertion d'un point, nous appliquons l'algorithme présenté pour la simplification globale mais seulement localement, sur le point ajouté et ses proches voisins.

Cependant cette méthode ne fonctionne pas bien et est source de mauvais résultats car elle dépend trop de l'ordre dans lequel sont insérés les points. Par exemple, comme le montre la figure 2.18, si un point est inséré à côté d'une zone simplifiée, il peut être lié à des points éloignés et former des arêtes ne représentant pas la réalité.

Pour ces raisons, la simplification en ligne a été abandonnée. À sa place la simplification globale est exécutée à la fin de l'acquisition.

## 2.4 Visualisation

La visualisation est une étape tout aussi importante car elle permet à l'opérateur de se rendre compte de la structure modélisée et ainsi de pouvoir superviser le déroulement de l'exploration. Aussi, la visualisation permet de valider le modèle généré en le comparant à la scène réelle.

Comme présenté à la section 1.2.2.2, l'interface de visualisation est un module distant, un client, qui se connecte à distance sur le programme central roulant sur l'ordinateur du robot. Ce module demande à intervalles réguliers, typiquement toutes les 5 secondes, la carte construite par le robot, plus précisément les coordonnées des faces résultant de la triangulation pour ensuite les afficher en trois dimensions en utilisant OpenGL. Un certain nombre de commandes permettent une certaine interactivité, comme se déplacer dans la scène reconstituée pour ainsi l'observer sous tous les angles. Pour mieux visualiser la scène, notamment avoir une idée de son échelle, le robot est aussi affiché dans sa position en temps réel.

La figure 2.19 montre deux captures d'écran de ce module de visualisation. Le mode plein permet de bien visualiser la scène entière alors que le mode fil de fer présente l'avantage de n'afficher que les arêtes et donc de bien visualiser les triangles de la structure de triangulation.

Des limitations de bande passante dans la liaison sans-fil nous ont aussi obligé à ne pas transmettre toute la carte à chaque fois. Ainsi seules les faces situées dans un voisinage proche du robot sont transmises, celles qui a priori ont eu un supplément d'information par rapport à la demande précédente ; le module d'affichage remplit alors juste les faces de ce voisinage dans l'affichage.

Un autre mode de visualisation est aussi très utile, notamment pour avoir une

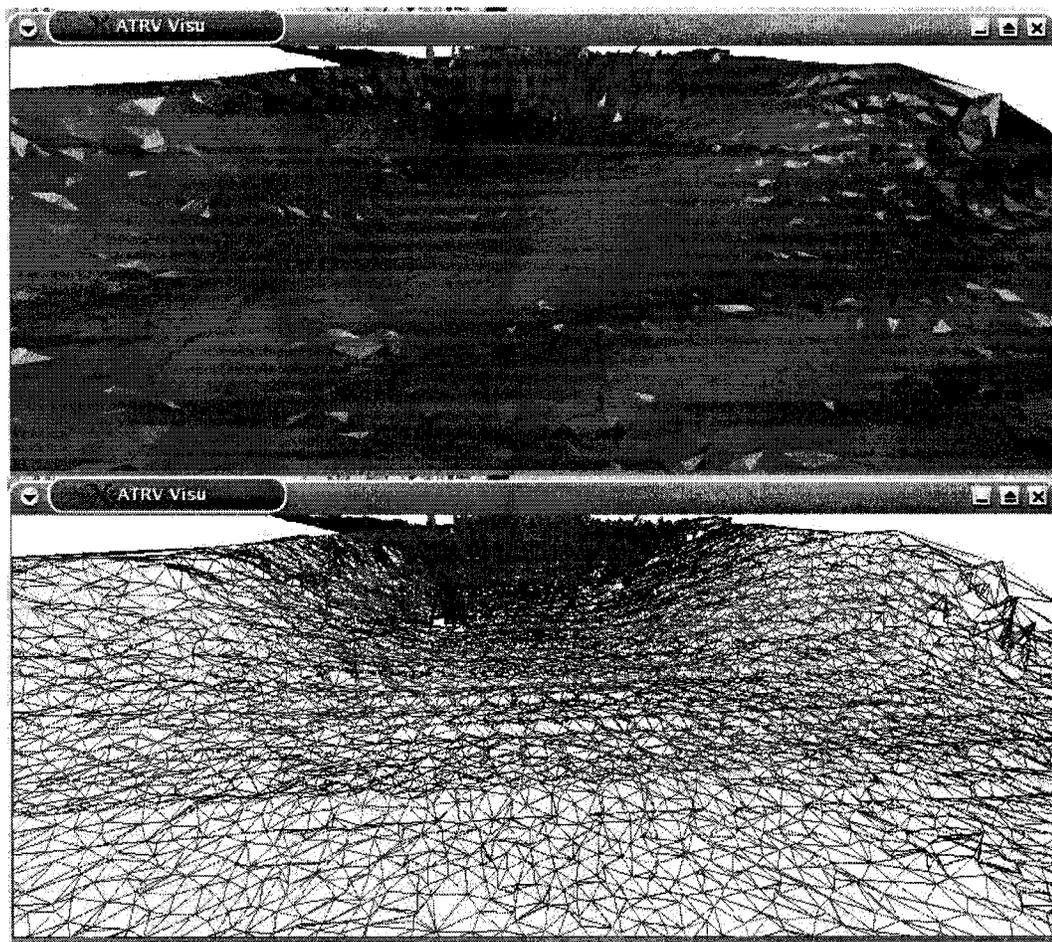


FIG. 2.19 Exemple de visualisation en trois dimensions, mode plein en haut et mode fil de fer en bas

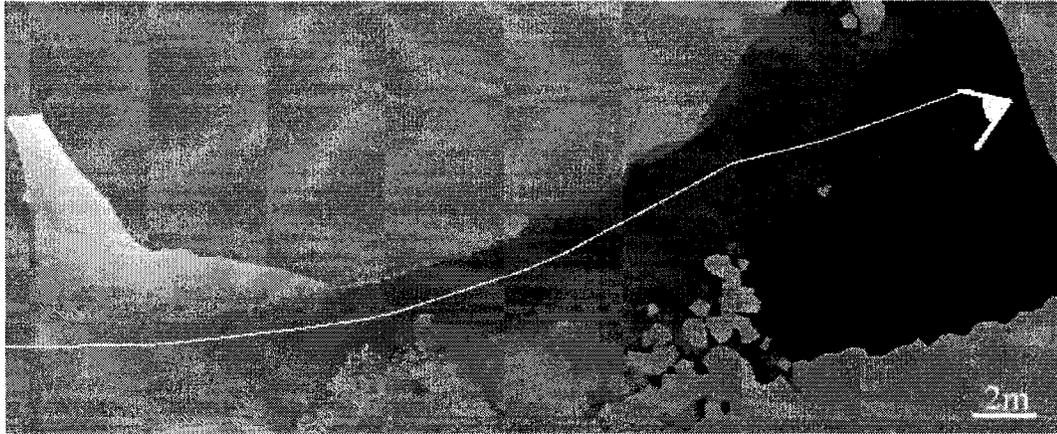


FIG. 2.20 Exemple de carte d'élévation. La ligne blanche représente la trajectoire du robot et les zones uniformes les zones inconnues, l'œil représente le point de vue de la photographie 2.21

idée plus précise de la structure du terrain et pour le mesurer. Il s'agit de la carte d'élévation (aussi appelée *DEM* pour *Digital Elevation Map* en anglais), une image représentant l'environnement projeté sur le plan horizontal et dont l'intensité des pixels est proportionnelle à la hauteur du point représenté (voir la figure 2.20 et la figure 2.21). Cette carte est créée à la fin de l'exploration et est donc utilisée a posteriori. Elle est particulièrement utile pour mesurer des distances dans l'environnement et des hauteurs; elle est d'ailleurs utilisée pour valider les différentes reconstructions.

Sur cette carte d'élévation peut aussi être tracée la trajectoire de la plate-forme (lignes en vert sur la figure 2.20). Il est ainsi possible de tester les algorithmes de navigation et d'exploration.

Nous venons de présenter un module de modélisation de la surface du terrain. À partir des points des profils acquis par le télémètre et de la trajectoire tridimensionnelle de la plate-forme, nous construisons un maillage triangulaire pour la



FIG. 2.21 Photographie du site de la carte d'élévation 2.20

modéliser. Des filtres ont été développés pour régler la résolution et pour éliminer les données redondantes. Enfin deux représentations sont créées pour permettre à l'opérateur de visualiser la modélisation.

## CHAPITRE 3

### MODULE D'EXPLORATION ET DE NAVIGATION

Ce module a pour tâche d'automatiser la navigation et l'exploration pour réduire les interventions de l'opérateur et lui permettre par exemple de superviser plusieurs plates-formes. Pour cela il est nécessaire que la plate-forme navigue de façon autonome en prenant en compte l'environnement qui l'entoure ; c'est-à-dire principalement éviter les obstacles. De plus, un objectif majeur pour la plate-forme étant l'exploration, elle doit donc être capable d'effectuer une exploration de l'environnement la plus exhaustive possible avec une intervention minimale de l'opérateur.

À noter aussi que seule la notion d'exploration nous intéresse ici, à savoir que le robot explore une zone donnée en évitant les obstacles ; nous ne cherchons donc pas à faire de suivi de trajectoire précis.

Ainsi différentes fonctionnalités d'exploration ont été développées :

- tout d'abord pour garantir une navigation sécuritaire pour le robot, il faut qu'il soit capable de déterminer les zones franchissables des zones infranchissables. C'est le rôle du module de détection d'obstacle ; il est basé sur la modélisation de l'environnement décrite dans le chapitre 2 ;
- un module de navigation purement réactive permettant au robot de naviguer sans objectif précis mais en évitant les obstacles détectés par le module précédent ;
- un module de navigation avec but dérivant du module précédent où le robot se déplace vers un point fournit par l'opérateur (tout en évitant les obstacles). Ce module dérive directement du précédent ;
- un module de recherche de zone à explorer où le robot détecte les zones inexplorées, en choisit une, puis se dirige dans sa direction pour enrichir au maximum

la carte.

Les modules de navigation fournissent un point vers lequel se rendre, il faut donc développer un contrôleur de déplacement qui garantisse que le robot se dirige bien vers le point en question.

Un scénario typique d'exploration consiste à explorer tout d'abord l'environnement soit en mode purement réactif, soit en mode réactif avec un but donné par l'opérateur. Au bout de quelques mètres, l'opérateur peut alors basculer dans le mode d'exploration de zones inconnues pour optimiser le processus.

Une contrainte importante est liée au montage proposé : la plate-forme ne balaie l'environnement que 2 m devant elle, elle ne peut donc pas effectuer de planification globale de sa trajectoire et doit la replanifier tous les mètres si l'on se donne 1 m de marge.

### 3.1 Détection de zones traversables

La première étape consiste à déterminer les zones où la plate-forme peut accéder sans encombres, c'est-à-dire les zones *traversables* regroupées dans une représentation appelée *carte de traversabilité*. Pour cela il faut déterminer les éléments pouvant empêcher ou perturber sa progression, à savoir les *obstacles*.

Pour un robot mobile les obstacles peuvent être classés en deux catégories :

- les *obstacles positifs*, les éléments contre lequel le robot mobile pourrait se heurter, par exemple un mur, un pilier, un arbre . . .
- les *obstacles négatifs*, qui représentent l'ensemble des zones où le robot pourrait chuter. En général ce sont des trous, des falaises, des escaliers descendants, etc.
- les zones difficilement traversables. Ces zones sont en théorie praticables par le robot mobile mais elles sont tellement perturbées qu'elles rendent sa progression



FIG. 3.1 Photographie de la situation des cartes locales de la figure 3.2. (la photographie est prise à partir du coin inférieur droit des cartes)

difficile et les mesures des capteurs à bord sont alors trop perturbées pour être suffisamment précises.

La solution que nous proposons est basée sur la création successive de quatre cartes locales de basse résolution centrées autour du robot, comme le montre la figure 3.2 :

1. une carte d'élévation ;
2. une carte de variations (dérivée) ;
3. une carte d'obstacles ;
4. une carte de traversabilité.

Les deux sections suivantes présentent l'élaboration de ces cartes.

### 3.1.1 Détection des obstacles

Reste maintenant à savoir comment détecter ces obstacles. Dans chaque type d'obstacle présentés précédemment, nous pouvons remarquer une caractéristique com-

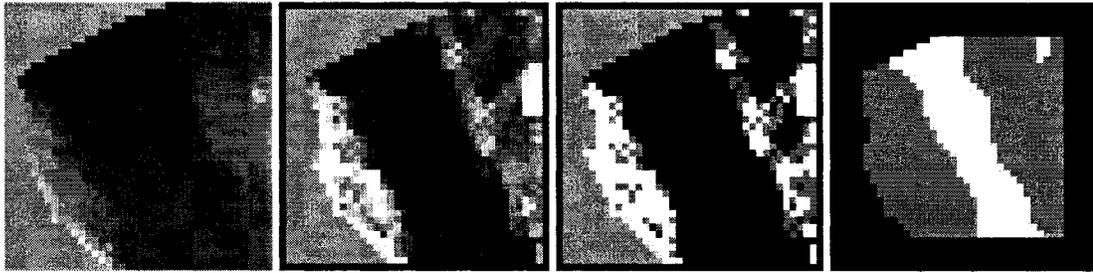


FIG. 3.2 Les quatre cartes locales créées (de gauche à droite) : carte d'élévation, carte dérivée, carte d'obstacles, carte de traversabilité locale (la photographie de situation est présentée par la figure 3.1). Dans la carte d'obstacles le blanc représente les obstacles, le gris les points perturbés et le noir les zones passables. Dans la carte de traversabilité le blanc représente les zones traversables, le gris les zones non traversables et le noir les zones inconnues.

mune : une forte variation de la surface du terrain. Ainsi pour repérer les obstacles, il faut détecter ces variations.

Bien que la structure de triangulation soit efficace pour la visualisation et l'approche multi-résolution, il est beaucoup plus simple de manipuler des cartes d'élévation pour y détecter des variations fortes. La première étape consiste donc à créer une *carte d'élévation* autour du robot avec une résolution moyenne (environ 10 pixels par mètre). Pour cela il suffit de centrer une grille autour du robot et de déterminer dans quels triangles tombent les points pour en calculer la hauteur par interpolation des trois sommets. La première image de la figure 3.2 présente un exemple d'une telle carte d'élévation.

Ensuite, il faut détecter les zones de forte variation du terrain qui sont signe d'obstacles et de zones perturbées. Le calcul consiste à prendre pour chaque point de la carte d'élévation (de coordonnées  $(i, j)$ ) le maximum de la différence de hauteur entre ce point et ses voisins dans un voisinage  $V(i, j)$  de taille proche de celle des roues comme le donne la formule 3.1. Ainsi, nous obtenons une deuxième carte représentant le gradient omnidirectionnel, une sorte de dérivée de la carte d'élévation ; nous l'appelons la *carte de variations* (voir la deuxième image de la figure

3.2).

$$d(i, j) = \max_{p \in V(i, j)} |h(i, j) - h(p)| \quad (3.1)$$

Un obstacle se caractérise alors par une variation infranchissable par la plate-forme, donc de taille comparable à celle des roues. La valeur du seuil choisi est donc fonction des roues; en pratique une valeur égale à 2/3 du rayon des roues est raisonnable.

Il reste encore à détecter les zones perturbées mais qui ne le sont pas suffisamment pour constituer des obstacles afin de déterminer les zones difficilement traversables. Ces zones bruitées correspondent aux variations comprises entre 1/3 et 2/3 du rayon des roues.

La troisième image de la figure 3.2 représente la *carte d'obstacles* dont les obstacles et les zones perturbées sont issues de la carte de variation (deuxième image).

Pour faciliter le déverminage et la mise au point de l'algorithme, toutes ces cartes sont en fait traitées comme des images où nous avons adopté un code de couleur (voir la figure 3.2).

### 3.1.2 Création de la carte de traversabilité

Une fois les obstacles et les zones de variation moyennes détectés, il faut en déduire les zones traversables par le robot. Pour cela nous créons la *carte de traversabilité* dont chaque point est déterminé à partir des points de son voisinage de la carte de variation :

- si un des points du voisinage correspond à un obstacle, le point est dit non

- franchissable ;
- si le nombre de points bruités est trop important, le point est aussi décrété non franchissable ;
- sinon le point est franchissable.

Le voisinage considéré correspond à la surface occupée par la plateforme.

La quatrième image de la figure 3.2 représente la carte de traversabilité créée à partir de la carte d'obstacles. Grâce à la photographie de la figure 3.1, nous pouvons voir que dans cet exemple la zone traversable correspond au chemin.

Ainsi, avec la carte de traversabilité obtenue, il est désormais possible de planifier, au moins localement, la trajectoire de la plate-forme.

Une *carte de traversabilité globale* est aussi construite au fur et à mesure de la navigation en fusionnant les cartes locales de traversabilité. Cette carte est ensuite utilisée dans le mode d'exploration par recherche de zones non explorées : cette partie est traitée à la section 3.2.3.

## 3.2 Modes de navigation

Les modes de navigation proposés sont tous basés sur la carte de traversabilité construite comme expliqué à la section 3.1.2. Chaque mode proposé est une évolution du mode précédent en apportant une nouvelle fonctionnalité.

### 3.2.1 Navigation purement réactive

Ce mode est le mode de base. Il indique le prochain point à atteindre en tenant compte des zones traversables.

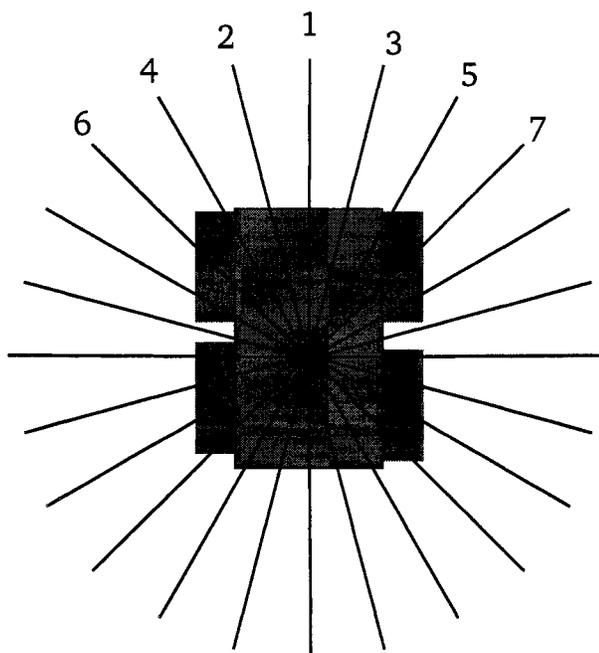


FIG. 3.3 Principe de l'exploration purement réactive

Ce mode de navigation permet au robot de se déplacer tout en modifiant le moins possible sa trajectoire, i.e. sa direction. Il privilégie donc une progression vers l'avant. La figure 3.3 illustre l'ordre de préférence des directions à suivre ; une direction est choisie si elle correspond à une zone traversable, sinon la direction suivante est envisagée.

### 3.2.2 Navigation avec but

Il est aussi intéressant que l'opérateur puisse contrôler la direction générale de l'exploration, en indiquant un point à atteindre. C'est un mode de navigation global.

Le principe est donc de modifier le module de navigation purement réactive de la section 3.2.1 pour permettre à la plate-forme de se diriger vers le point-cible. Ce dernier est indiqué par l'opérateur, par exemple au moyen de l'interface de visualisation. Au lieu de choisir la prochaine direction vers l'avant du véhicule, la

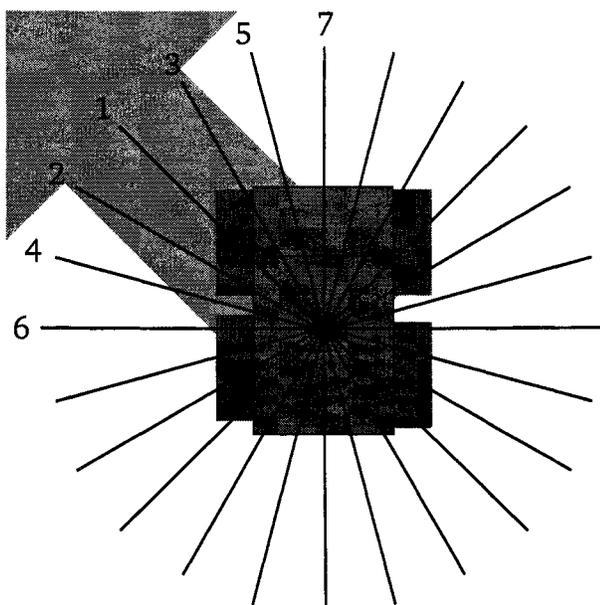


FIG. 3.4 Principe de l'exploration avec but

direction est choisie le plus possible dans l'axe du point-cible, tout en tenant compte de la carte de traversabilité. La figure 3.4 présente ce mode de navigation.

Une fonctionnalité intéressante est obtenue en choisissant comme point-cible le point de départ de la plate-forme. Ainsi cela lui permet de revenir à sa position de départ sans que l'opérateur n'ait à la guider.

### 3.2.3 Navigation vers les zones inconnues

Dans ce mode de navigation, la stratégie consiste à détecter les zones inexplorées potentiellement intéressantes, à en choisir une, puis à l'explorer.

### 3.2.3.1 Détection des frontières

Les zones inexplorées ne sont pas visibles car elles sont inconnues, mais il est possible de détecter là où elles sont situées en observant la carte de traversabilité : les frontières entre les zones traversables et les zones inconnues sont des portes vers de telles zones Ceci est représenté à la figure 3.5.

La première étape consiste donc à détecter et identifier les frontières entre les zones traversables et inconnues de la carte de traversabilité. Pour cela, il faut :

1. marquer les points voisins à la fois d'un point traversable et d'un point de zone inconnue pour en faire des *points-frontières* ;
2. identifier les points faisant partie d'une même frontière en regroupant les points-frontières et en créant un objet *frontière* par groupe ;
3. déterminer les caractéristiques de chaque frontière.

Pour regrouper les points faisant partie d'une même frontière, nous utilisons un algorithme de *flood-filling* :

1. prendre un point-frontière, le marquer et le mettre dans la pile ;
2. tant que la pile n'est pas vide ;
  - prendre un point de la pile ;
  - prendre ses points-frontière voisins non marqués, les marquer et les mettre dans la pile.

L'algorithme retourne une frontière, il suffit de le répéter jusqu'à ce qu'il n'y ait plus de points-frontière non-marqués. La figure 3.5 présente un exemple de carte globale de traversabilité sur laquelle les frontières ont été détectées.

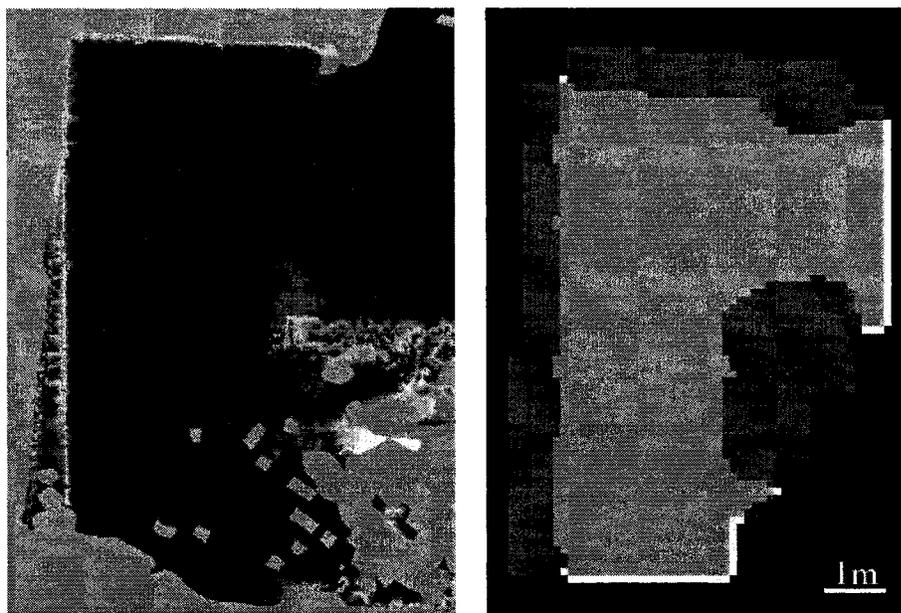


FIG. 3.5 Carte d'élévation (à gauche) et la carte de traversabilité résultante (à droite). Les frontières détectées sont représentées en blanc.

### 3.2.3.2 Choix de la frontière à explorer

Une fois les frontières identifiées, il faut déterminer des caractéristiques pour choisir laquelle sera explorée.

Une première est la taille de la frontière, c'est-à-dire le nombre de pixels qui la compose. Cette caractéristique fournit une estimation de la taille de la zone se situant derrière.

Une autre caractéristique consiste en un point de référence permettant de déterminer à quelle distance elle se trouve de la plate-forme. Ce point ne peut être le barycentre de tous les points constituant la frontière car dans le cas d'une frontière en arc de cercle, le point se trouverait dans une zone traversable. Ce point doit donc être représentatif et se situer sur la frontière. Le point médian de la frontière répond à ces caractéristiques. Une frontière est en effet linéaire, une courbe sans

bifurcation ; le point médian est donc à mi-chemin entre le point de départ et le point d'arrivée.

Le choix de la frontière à suivre se fait de la manière suivante. Les frontières courtes sont à ignorer car elles ne permettent pas à la plate-forme de passer au travers. Ensuite, l'ordre de priorité est basé sur un compromis entre la distance de la frontière à la plate-forme et de sa taille. L'équation 3.2 propose un exemple de fonction de coût  $C(f)$  à maximiser pour déterminer la frontière à explorer ( $T(f)$  est la taille de la frontière  $f$  et  $d(f)$  est sa distance à la plate-forme). L'opérateur n'a qu'à régler le paramètre  $\lambda$  pour ajuster sa préférence.

$$C(f) = \lambda \cdot T(f) - d(f) \quad (3.2)$$

L'exploration se fait ensuite en deux temps :

- la plate-forme se dirige vers le point médian de la frontière choisie et arrive de cette manière en face de la frontière ;
- elle bascule ensuite dans le mode d'exploration purement réactive pour explorer la zone située devant lui.

### 3.3 Contrôleur de déplacement

Chacun des modes de navigation décrits ci-dessus fournissent des points à atteindre ; il faut donc développer un contrôleur de déplacement garantissant que la plate-forme se dirige bien vers eux. Nous avons d'abord testé le contrôleur d'Astolfi (Astolfi, 1994), mais finalement nous l'avons modifié pour qu'il réponde plus à nos besoins.

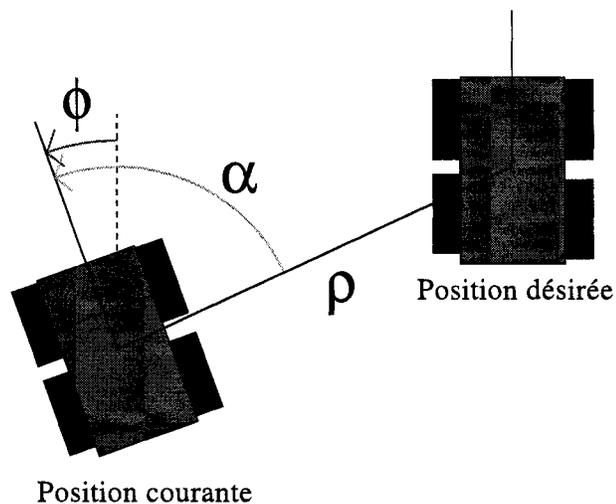


FIG. 3.6 Paramètres du contrôleur d'Astolfi

### 3.3.1 Le contrôleur d'Astolfi

Ce contrôleur permet originellement de suivre la trajectoire d'un point en commandant le robot mobile en vitesse de translation  $\Omega$  et de rotation  $V$ . Il a été choisi pour la simplicité de ses équations et pour la vitesse de translation résultante qui est toujours positive. La figure 3.6 permet de définir les paramètres de l'algorithme.

Les équations fournissant les vitesses du robot sont alors :

$$V = k_{\rho}\rho \quad (3.3)$$

$$\Omega = k_{\alpha}\alpha + k_{\phi}\phi \quad (3.4)$$

avec les paramètres  $k_{\rho}$ ,  $k_{\alpha}$  et  $k_{\phi}$  vérifiant les conditions suivantes pour assurer la

convergence du contrôleur :

$$\begin{cases} k_\rho > 0 \\ k_\phi < 0 \\ k_\alpha + k_\phi - k_\rho > 0 \\ k_\alpha + 4k_\phi - \frac{2}{\pi}k_\rho > 0 \end{cases} \quad (3.5)$$

Comme nous pouvons le voir dans l'équation 3.3, la vitesse du robot est fonction de la distance entre la plate-forme et le point objectif. Nous préférons que la plate-forme se déplace à une vitesse constante ; nous imposons donc la vitesse de translation et adaptons la vitesse de rotation pour que la trajectoire ne soit pas modifiée et pour garantir que la trajectoire converge vers le point souhaité.

Cependant ce contrôleur ne convient pas tout à fait à la situation pour les raisons suivantes :

- la trajectoire n'est pas assez directe si le point objectif n'est pas dans l'axe du robot (voir la figure 3.7). La zone traversable est donc plus étendue qu'avec un chemin direct, plus difficile à prévoir, et a plus de chance de rencontrer un obstacle ;
- l'orientation finale de la plate-forme n'est pas importante.

### 3.3.2 Contrôleur proposé

Le contrôleur que nous proposons remédie aux deux problèmes précédents.

Comme l'orientation finale n'a pas d'importance, le terme  $k_\phi\phi$  est supprimé de l'équation de contrôle.

Pour éviter que la trajectoire s'étende trop, le robot n'avance que si le point objectif

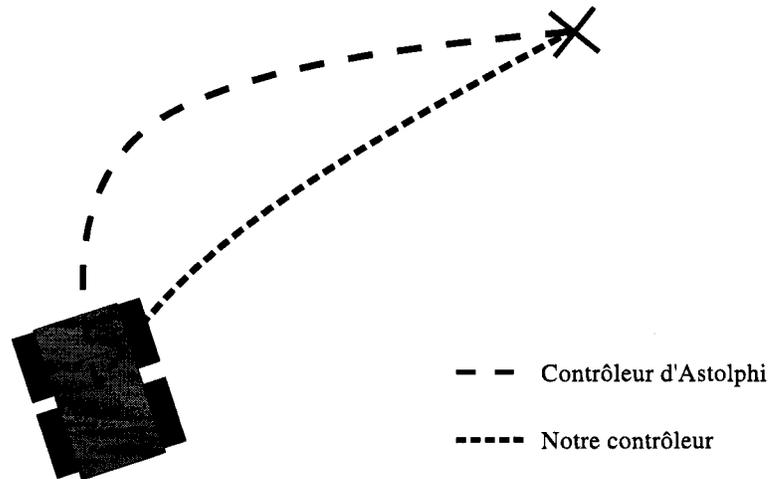


FIG. 3.7 Comparaison des trajectoires obtenues avec le contrôleur d'Astolfi et avec notre contrôleur

est compris dans un angle de  $30^\circ$  devant lui.

Le contrôleur devient alors :

$$V = \begin{cases} V_0 & \text{si } |\alpha| < 15^\circ \\ 0 & \text{sinon} \end{cases} \quad (3.6)$$

$$\Omega = \begin{cases} k_\alpha \alpha \frac{V_0}{k_\rho \rho} & \text{si } |\alpha| < 15^\circ \\ \Omega_{max} & \text{sinon} \end{cases} \quad (3.7)$$

Un tel contrôleur a aussi comme avantage de donner à la plate-forme une trajectoire lisse si les points objectifs sont à peu près dans l'axe du robot, ce qui est vérifié la plupart du temps.

Dans ce chapitre nous avons présenté le module d'exploration en détails. Il est constitué d'une part d'un module établissant la carte de traversabilité à partir de la modélisation du terrain du chapitre 2 et d'autre part de fonctionnalités de

navigation réactive et d'exploration. De plus un contrôleur de déplacement permet d'assurer que la plate-forme se dirige vers les points fournis par les modules de navigation.

## CHAPITRE 4

### RÉSULTATS

Notre système de modélisation de l'environnement et du terrain a été intensivement testé en intérieur et en extérieur. Différentes études ont été menées pour tester la qualité du système développé.

Tout d'abord, nous avons testé sa viabilité en effectuant une simulation unidimensionnelle de notre système, notamment pour savoir si l'estimation de la position et de l'attitude de la plate-forme à partir de la modélisation du terrain pouvait améliorer celle fournie par le module d'odométrie 3D. Ensuite, une présentation des principes de l'interface permet de prendre connaissance des principes d'utilisation du programme développé pour les expériences. Puis, après avoir expliqué la méthode de calibration de la position de l'inclinomètre, nous présentons la modélisation de scènes calibrées intérieures et extérieures pour valider la reconstruction. D'autre part une petite étude sur l'influence de l'inclinomètre révèle que ce dernier est un élément critique pour la qualité de la modélisation. Enfin, plusieurs résultats significatifs permettent de juger des algorithmes de navigation et d'exploration.

Des films sont joints sur le CD en annexe et illustrent les processus de cartographie et d'exploration dans différents environnements, intérieurs et extérieurs. La vidéo `occlusion.avi` permet d'illustrer concrètement le phénomène d'occlusion en intérieur (voir la section 4.3.5), la vidéo `labo.avi` présente un exemple de reconstruction d'un milieu intérieur avec un système de planches pour faire varier l'attitude de la plate-forme. Les vidéos `canyon1.avi`, `canyon2.avi` et `canyon3.avi` présentent des exemples de modélisation d'un environnement de type canyon, les deux dernières présentent une navigation réactive dans ce milieu. La vidéo `montee.avi` présente

la reconstruction d'un environnement où le sol n'est pas horizontal. La vidéo `triangles_auto.avi` montre une navigation réactive parmi des obstacles dans un milieu extérieur ; nous pouvons noter que le trottoir est considéré comme un obstacle. La vidéo `triangles_guide.avi` présente un exemple de navigation réactive avec objectif : à la fin de la vidéo, lorsque la plate-forme atteint son objectif, elle s'arrête.

Le manuel de l'application développée, fournit en annexe III, permet de se rendre compte du mode opératoire utilisé dans les différentes expérimentations.

#### 4.1 Simulation unidimensionnelle

Une simulation du système a été menée pour déterminer s'il était possible d'utiliser la modélisation du sol pour calculer l'attitude et l'altitude de la plateforme. En effet, en simulant la plate-forme reposant sur la surface du sol modélisé, il est possible d'obtenir sa hauteur ainsi que son attitude (tangage et roulis). Ces dernières informations pourront alors compléter et améliorer celles fournies par l'inclinomètre.

Pour cela la surface du sol est modélisée par des segments de droite. Un modèle de la plate-forme se déplaçant sur ce profil a été créé ainsi qu'un modèle du télémètre scrutant le sol. L'annexe I.1 décrit le modèle de simulation.

Dans cette simulation nous avons utilisé le profil reconstruit pour estimer l'angle de tangage et la hauteur de la plate-forme (en simulant le déplacement de la plate-forme sur le profil estimé) au lieu de prendre les données de l'inclinomètre. Cependant, comme il n'y a pas de boucle de retour pour l'estimation de l'attitude et de la hauteur, ces données ont tendance à diverger assez rapidement et à donner des résultats erronés et inexploitable comme le montre la figure 4.1.

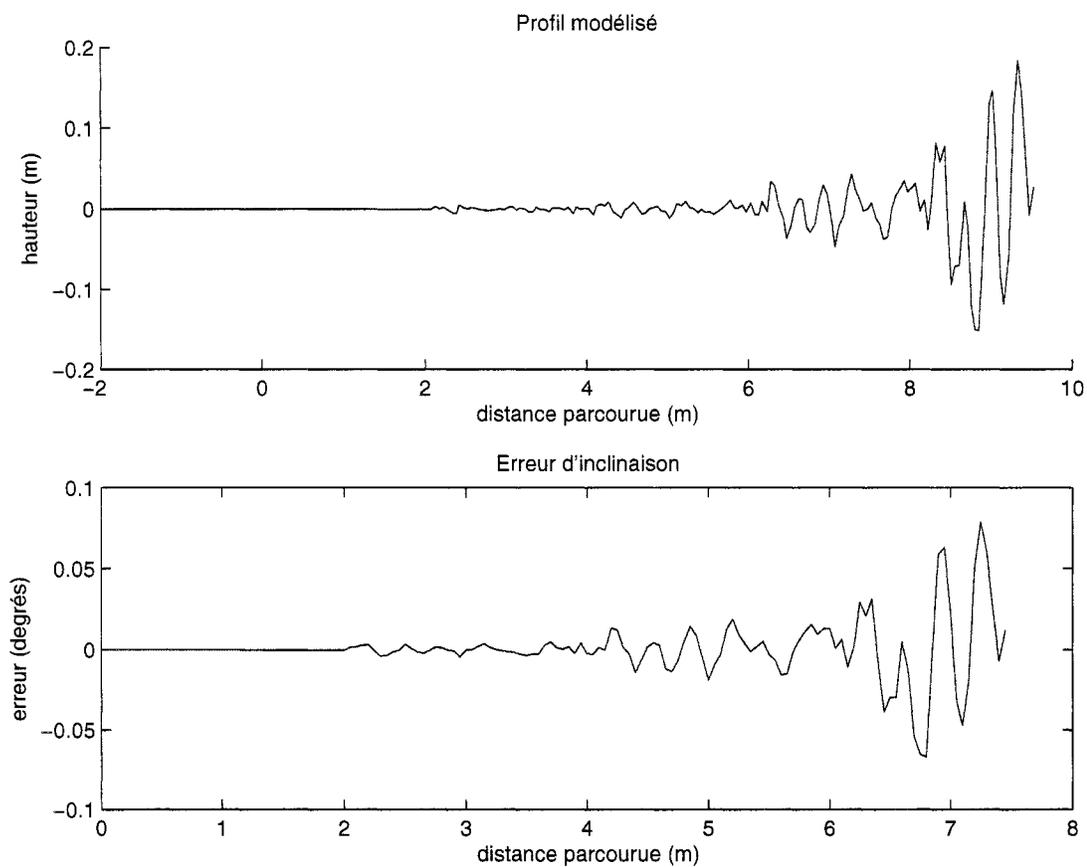


FIG. 4.1 Exemple de simulation sur un sol plat. La première courbe représente le profil du terrain modélisé par le robot tandis que la seconde représente l'erreur de l'estimation de l'inclinaison le long du parcours (bruit du laser : 0,5cm). Les oscillations de la fin des courbes sont dues à une divergence du système créées par l'accumulation des erreurs le long du chemin parcouru.

Pour corriger cette divergence, nous avons mis en place un filtre de Kalman (Welch and Bishop, 2003) pour combiner les données de l'inclinomètre et du modèle de la surface du sol et ainsi obtenir la hauteur de la plate-forme. L'annexe I.2 présente comment ce filtre a été mis en place.

Cependant les premiers résultats obtenus, montrés à la figure 4.2, indiquent que la hauteur estimée avec le filtre de Kalman n'est pas meilleure que celle obtenue par intégration des mesures de l'inclinomètre. Théoriquement, nous nous étions attendus à obtenir des résultats meilleurs en combinant les données des deux manières de calculer l'attitude. Cependant, comme le montre un calcul dans l'annexe I.2, les données de l'inclinomètre sont 40 fois plus fiables que l'inclinaison et la hauteur calculées à partir de la surface modélisée ; ce qui explique pourquoi l'amélioration attendue n'est pas visible.

Bien que l'idée soit théoriquement intéressante, nous avons donc décidé, à ce stade, de ne pas poursuivre cette piste et de n'utiliser que les données de l'inclinomètre pour calculer l'attitude et la hauteur de la plate-forme.

## 4.2 Présentation des interfaces

Comme l'a présenté la section 1.2.2.2, le programme est constitué de deux parties : une roulant sur la plate-forme et modélisant le terrain, et un autre roulant sur un ordinateur distant et permettant de visualiser la modélisation en cours.

L'interface du programme de la plate-forme est assez simple puisqu'elle est en mode console : l'opérateur dirige la plate-forme et change ses modes à l'aide des touches d'un clavier. L'annexe III présente l'utilisation détaillée du programme.

L'interface de visualisation affiche en trois dimensions le maillage généré. Comme le

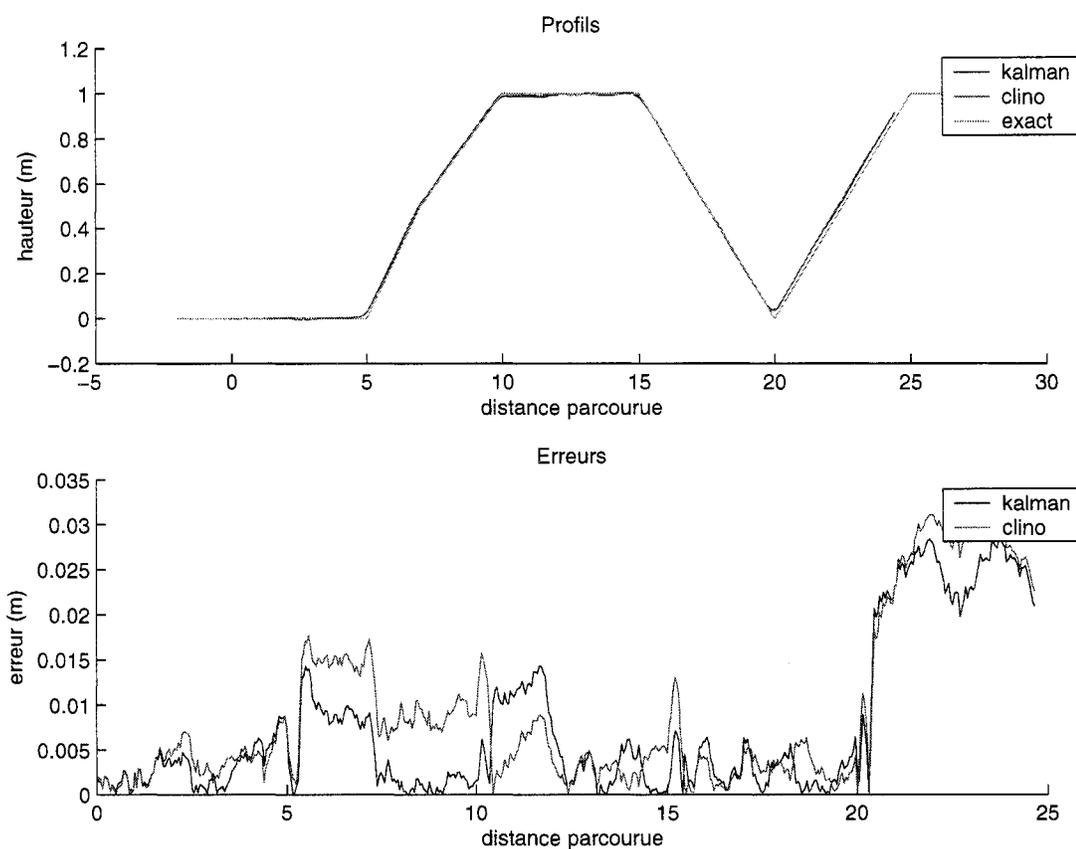


FIG. 4.2 Comparaison de deux méthodes pour calculer la hauteur de la plateforme (simulation, écart-type du bruit de l'inclinomètre :  $1^\circ$ , écart-type du bruit du télémètre : 1cm). Une méthode utilise uniquement les données de l'inclinomètre pour estimer le profil du terrain tandis qu'une autre mélange ces données avec l'attitude estimée à partir du profil modélisé à l'aide d'un filtre de Kalman. La première courbe montre que les profils obtenus sont proches de la réalité et la seconde démontre que le filtre Kalman n'apporte pas d'amélioration significative.

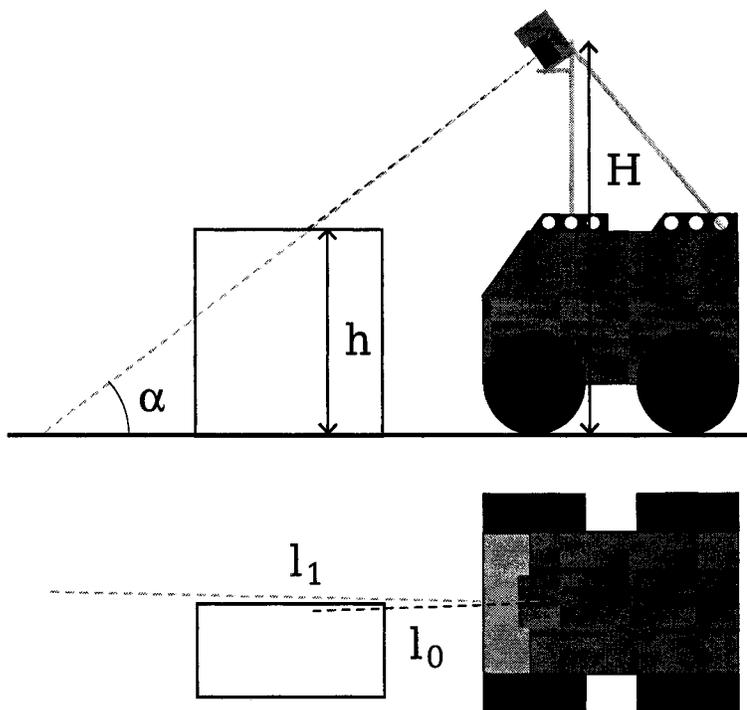


FIG. 4.3 Principe de la calibration de la position du télémètre laser

montrent les vidéos en annexe, cette interface est interactive car l'opérateur peut, avec la souris, tourner, se déplacer et zoomer dans la scène. Un mode fil-de-fer permet de mieux visualiser les triangles du maillage.

### 4.3 Validation de la reconstruction

#### 4.3.1 Calibration du télémètre laser

La calibration du télémètre laser est une étape fondamentale puisqu'elle influence directement la qualité et la précision de la modélisation.

La figure 4.3 présente la configuration de la plate-forme pour la calibration : un objet, de hauteur connue  $h$  est placé en avant de façon à ce qu'il intercepte le rayon

d'angle  $0^\circ$  du télémètre et celui d'angle  $1^\circ$  touche le sol. Ces deux rayons, d'angles  $0^\circ$  et  $1^\circ$ , sont choisis car ils correspondent aux rayons les mieux alignés avec l'axe longitudinal de la plate-forme. La position du télémètre est alors donnée par les équations suivantes (les paramètres sont définis dans la figure 4.3) :

$$\alpha = \arcsin \left( \frac{h}{l_1 - l_0} \right) \quad (4.1)$$

$$H = l_1 \cos \alpha \quad (4.2)$$

Dans notre cas nous obtenons :

$$\alpha = \arcsin \left( \frac{57}{190 - 93} \right) = 36^\circ \quad (4.3)$$

$$H = 190 \sin 36^\circ = 112 \text{ cm} \quad (4.4)$$

### 4.3.2 Comparaison de scènes calibrées

Un bon moyen de vérifier si la modélisation d'une surface est fiable est de simplement mesurer sa précision en utilisant une scène calibrée. Cependant, il est très difficile de calibrer une scène extérieure non structurée et nous avons dû nous limiter à une scène intérieure. Nous avons donc créé une scène intérieure, que nous avons ensuite mesuré pour connaître ses dimensions réelles. Nous avons comparé ensuite ces mesures avec celles obtenues avec la carte d'élévation créée à partir de la modélisation.

Cette modélisation est effectuée dans deux conditions. Dans le premier cas, la plate-forme se déplace sur un plan horizontal ; dans le second elle se déplace sur un plan incliné (pente). La figure 4.4 permet de comparer les valeurs des différentes mesures. Nous pouvons alors constater que la modélisation est précise à 1 centimètre près dans les deux cas.

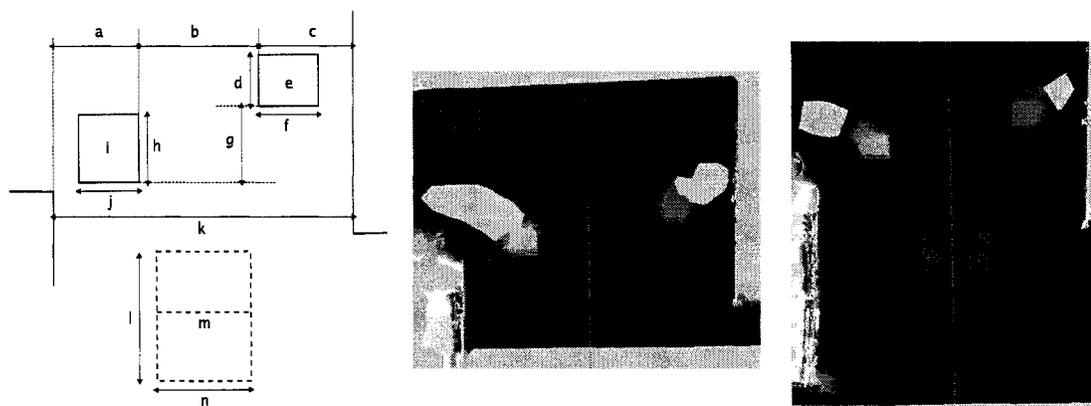


FIG. 4.4 Modélisation d'une scène calibrée : à gauche se situe le plan de la scène et à droite deux modélisation de la scène, une où la plate-forme s'est déplacée sur le sol horizontal, et une où elle a évolué sur le plan incliné. Différentes mesures ont été effectuées pour comparer la précision des modélisations avec la scène réelle (voir le tableau 4.1 pour les mesures). Les mesures  $e$ ,  $i$  et  $m$  représentent des hauteurs alors que les autres représentent des distances.

Mesures (en cm)	réalité	plan horizontal	plan incliné
a	126	127	125
b	213	214	216
c	134	133	133
d	45	45	47
e	34	34	33,5
f	42	44	44
g	61	60	63
h	40	42	42
i	57	59	56.5
j	48	48	49
k	474	474	475
l	243		245
m	16		16.5
n	122		121

TAB. 4.1 Comparaison des mesures d'une scène intérieure calibrée : mesures réelles, modélisation à partir d'un sol plat et modélisation à partir d'un sol avec un plan incliné (voir la figure 4.4)

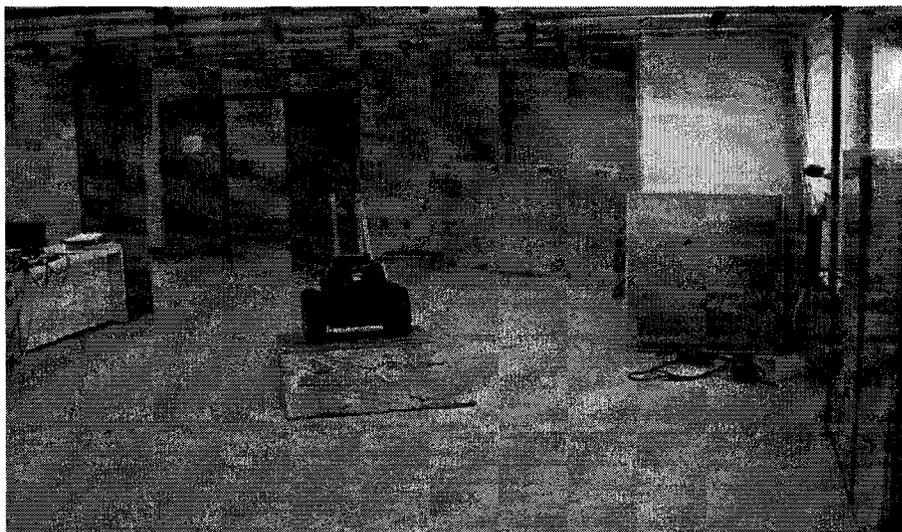


FIG. 4.5 Photographie du laboratoire avec le système utilisé pour faire varier l'attitude de la plate-forme

### 4.3.3 Modélisation de scènes intérieures

Les tests en intérieur, plus particulièrement en laboratoire, nous ont été très utiles pour tester nos algorithmes. En effet, c'est l'environnement qui est immédiatement disponible et les expériences y sont beaucoup plus faciles à mettre en œuvre qu'en extérieur.

Cependant les environnements intérieurs ne permettent pas de tester en totalité les fonctionnalités du système :

- le sol est plat et les tests de changement de l'attitude sont limités à l'utilisation de planches ;
- les environnements intérieurs sont assez limités puisqu'ils sont essentiellement composés du sol horizontal et des murs verticaux.

Les figures 4.5 et 4.6 présentent la modélisation typique du laboratoire avec le système de planches utilisées pour faire varier l'attitude de la plate-forme.

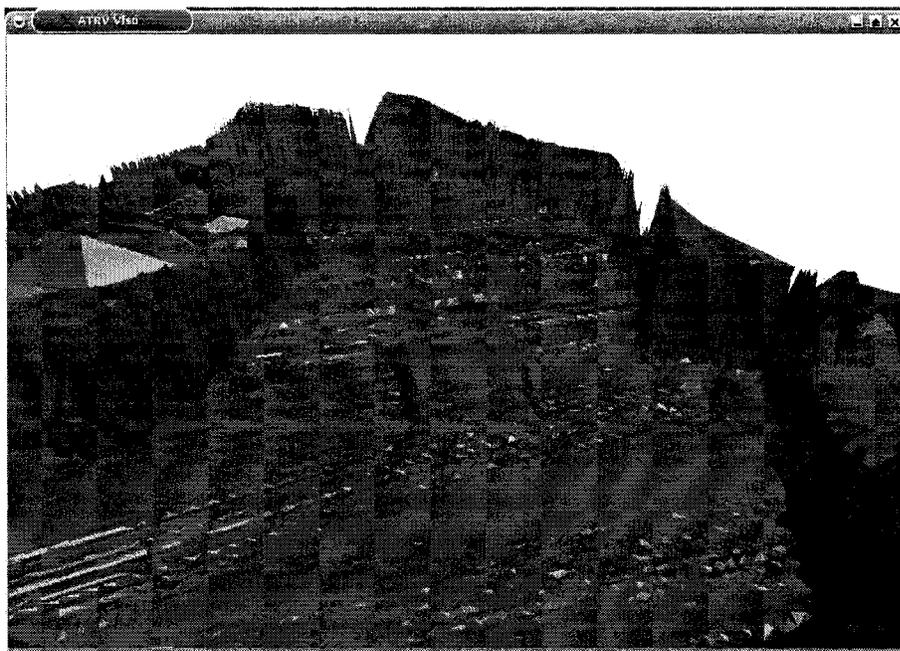


FIG. 4.6 Modélisation d'un environnement intérieur : le laboratoire correspondant à la photographie de la figure 4.5

#### 4.3.4 Modélisation de scènes extérieures

Les scènes extérieures ne sont pas structurées. Il est donc difficile de mesurer la précision de leur modélisation. Cependant, des comparaisons entre les modélisations et les photographies des sites permettent de juger de la qualité des cartes obtenues ainsi que de leur fidélité.

Nous avons testé notre système dans différents types d'environnements extérieurs :

- des environnements concaves comme le terrain en forme de canyon de la figure 4.8;
- des environnements convexes comme la butte boisée de la figure 4.10;
- des environnements plus structurés comme la figure 4.12.

Les environnements concaves se révèlent mieux modélisés que les environnements convexes. En effet, les premiers sont mieux adaptés à la géométrie du montage,

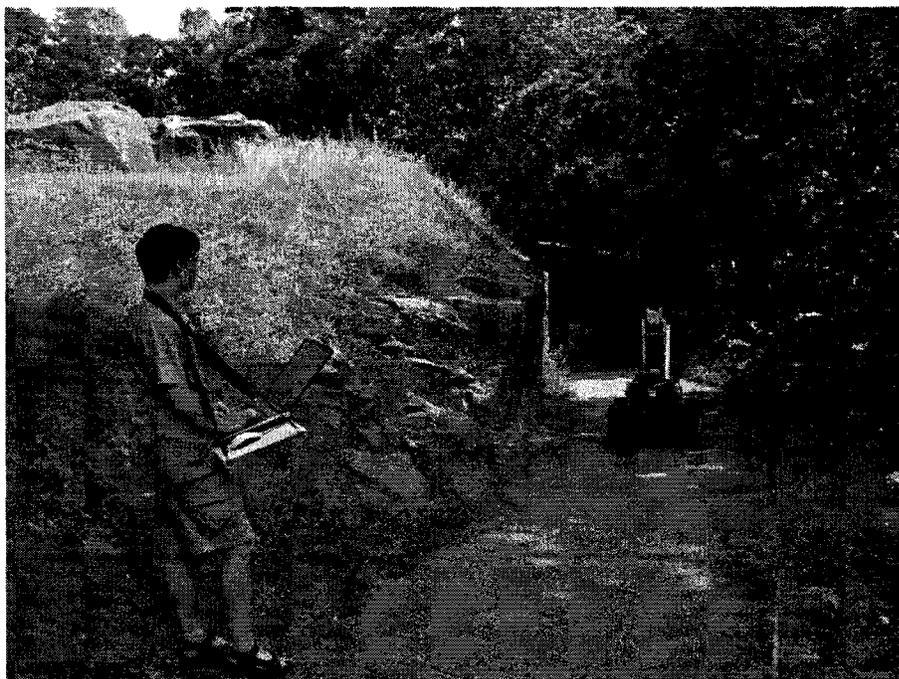


FIG. 4.7 Photographie correspondant à un emplacement de type canyon (concave)

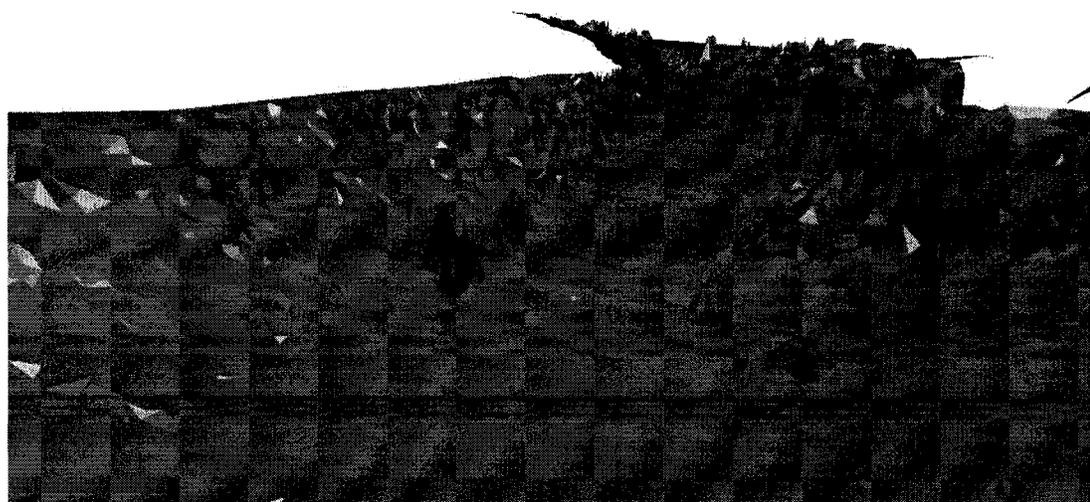


FIG. 4.8 Résultat de cartographie 3D pour l'environnement de la figure 4.7



FIG. 4.9 Photographie correspondant à un emplacement de type butte (convexe)



FIG. 4.10 Résultat de cartographie 3D pour l'environnement de la figure 4.9. Les troncs d'arbres ressortent bien de la surface.

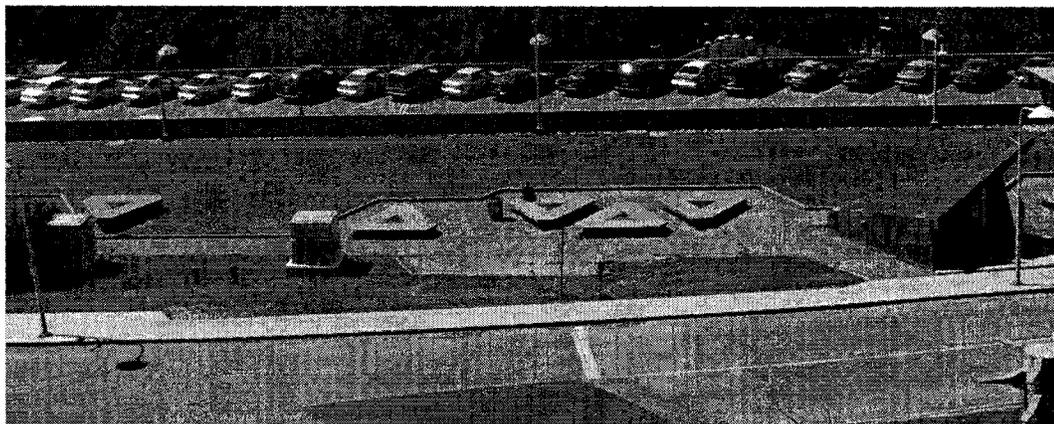


FIG. 4.11 Photographie d'un environnement extérieur structuré

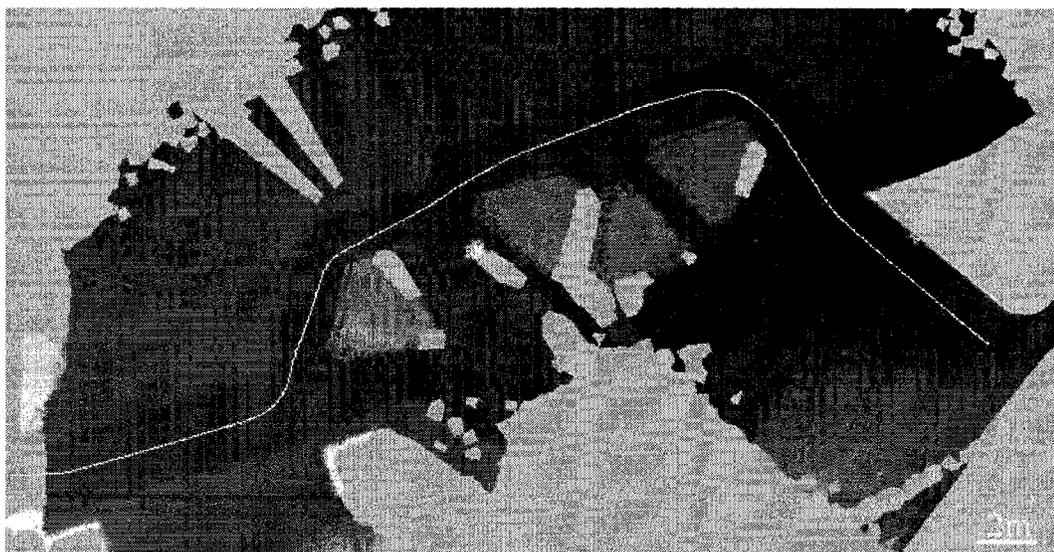


FIG. 4.12 Résultat de cartographie 3D pour l'environnement de la figure 4.11

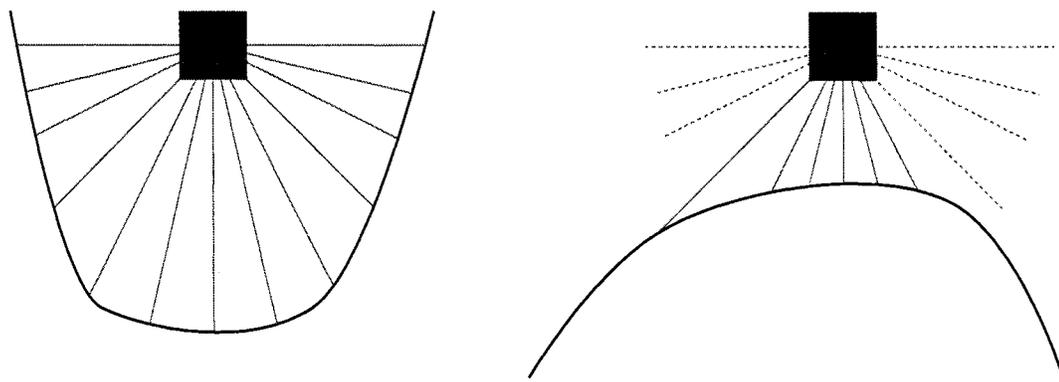


FIG. 4.13 Comparaison de l'incidence des rayons du télémètre selon le type de terrain (concave à gauche et convexe à droite).

tandis que pour les seconds, seul le dessus est modélisé ; la figure 4.13 illustre cette particularité.

La limite principale de notre système vient du module de localisation. C'est en effet lui qui garantit la consistance de la carte notamment lorsqu'une zone est balayée à plusieurs reprises. La figure 4.15 présente un exemple où cette erreur de positionnement s'est fortement accumulée et rend une partie de la carte inexploitable. Ainsi le bruit dans le quart inférieur droit de la carte d'élévation atteste du manque de boucle de retour dans l'algorithme de localisation. Dans ce cas c'est l'erreur d'estimation de la hauteur de la plate-forme qui provoque ces erreurs : au bout du parcours, d'une distance d'une cinquantaine de mètres, la plate-forme considère qu'elle est 40 cm plus haut qu'au départ ; la représentation est donc bruitée car notre système tente de représenter deux surfaces superposées. Étant données la nature mouvementée du terrain et la longueur du parcours, une telle erreur est difficilement évitable.

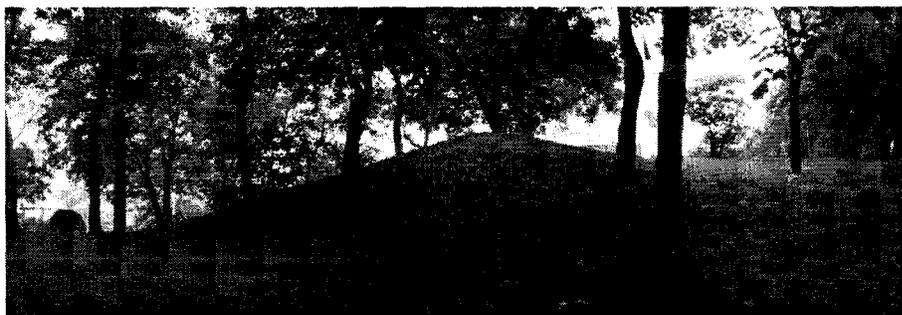


FIG. 4.14 Photographie du site de la figure 4.15 (la photographie est prise à partir d'un point situé en bas à droite)

#### 4.3.5 Régions occlusées

Une particularité du montage est de balayer le terrain 2 m à l'avant avec le télémètre laser. Ainsi lorsque la plate-forme contourne un coin et son rayon de courbure est trop faible, une zone située au pied de cet angle se retrouve non balayée et est considérée inconnue. La figure 4.16 illustre ce phénomène.

La vidéo `occlusion.avi` fournie en annexe illustre aussi ce phénomène.

#### 4.3.6 Influence de l'inclinomètre

L'inclinomètre constitue un capteur sensible de notre système. Du fait de notre montage, les erreurs d'inclinaisons se retrouvent amplifiées car la zone balayée est située à 2 m de la plateforme. De plus, comme le système balaie la surface du terrain selon des lignes perpendiculaires à la trajectoire, ce phénomène est encore plus accentué.

D'autre part, nous nous sommes aperçu que l'inclinomètre utilisé a une fréquence propre de 10Hz, fréquence proche de celle générée par les oscillations dues aux crampons lors du déplacement de la plate-forme. Ainsi l'inclinomètre donne des

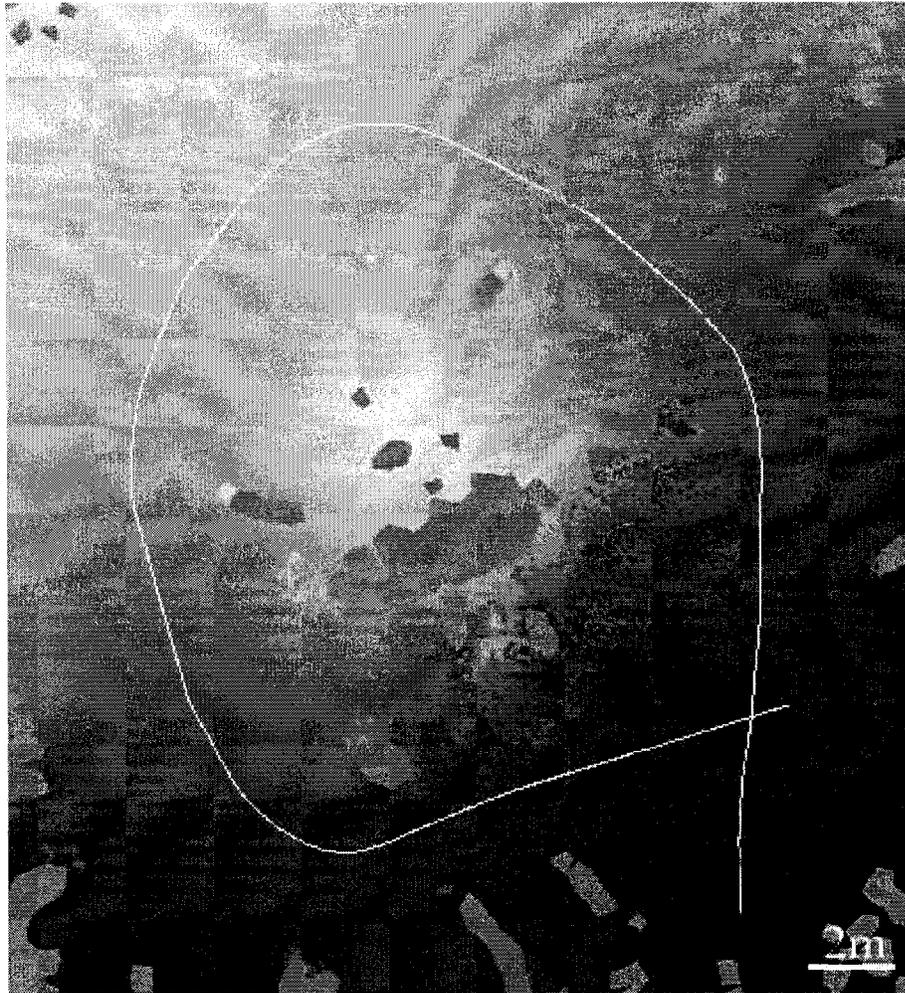


FIG. 4.15 Exemple de divergence du module de localisation 3D suite à une boucle de grande dimension (50m) sur un terrain accidenté. Le bruit du quart inférieur droit de l'image est la manifestation de cette erreur de positionnement (voir le texte).

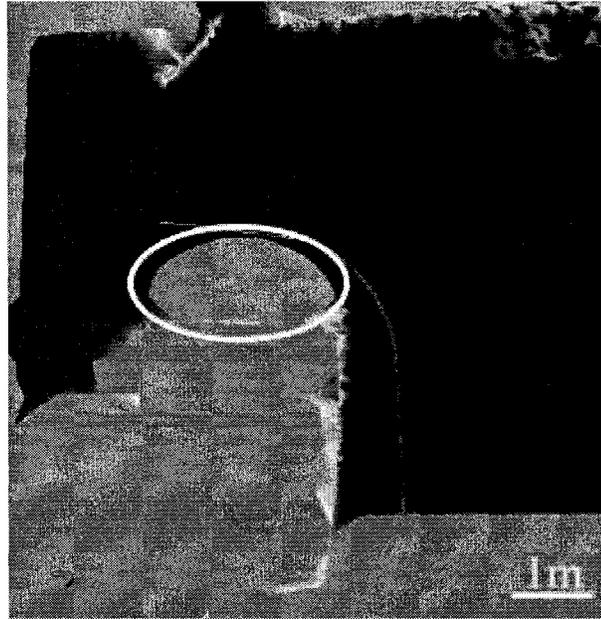


FIG. 4.16 Exemple de zones d'occlusion (encerclée en blanc) suite au contournement d'un angle

oscillations de  $1,5^\circ$  d'amplitude lorsque la plate-forme se déplace sur une surface plate (en laboratoire).

Pour atténuer cet effet, nous avons utilisé un filtre passe-bas numérique classique du premier ordre :

$$s_n = a e_n + (1 - a) s_{n-1} \quad (4.5)$$

avec  $e_n$  l'entrée du filtre,  $s_n$  sa sortie,  $0 \leq a \leq 1$  et la constante de temps vallant  $\tau = T_e \left( \frac{1-a}{a} \right)$  où  $T_e$  est la période d'échantillonnage.

On doit aussi prendre en compte que si la plate-forme évolue sur un terrain perturbé, ces perturbations peuvent très bien être de fréquence spatiale proche de celle des crampons du robot. De tels détails du terrain peuvent alors être amoindris par le filtre.

Une autre solution consiste à prendre le problème à la base, c'est-à-dire au niveau du

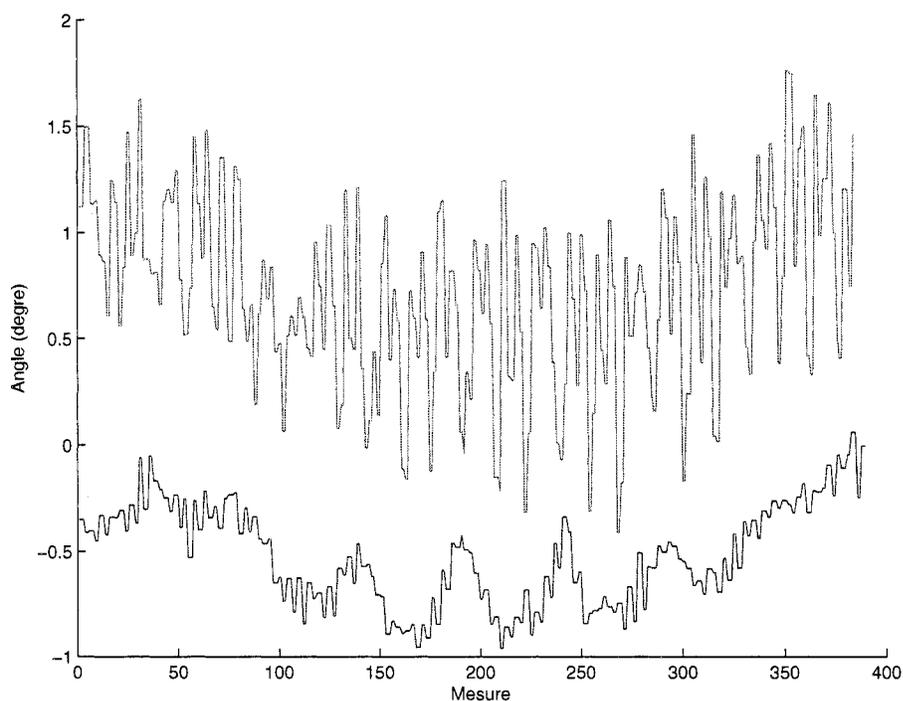


FIG. 4.17 Comparaison des données fournies par l'inclinomètre original (en haut) et celui avec le fluide visqueux (en bas) (les valeurs moyennes ont été décalées pour pouvoir comparer l'allure des deux signaux)

capteur. En effet, la compagnie fabriquant l'inclinomètre propose une amélioration du capteur en remplaçant le fluide par un autre plus visqueux et limitant ainsi la résonance à 10Hz. Nous l'avons donc fait mettre à jour et comparé les résultats comme l'illustre la figure 4.17. L'amplitude du bruit de mesure est ainsi bien plus faible avec l'inclinomètre à fluide visqueux amortissant les oscillations parasites.

Au niveau de la modélisation, le passage à un inclinomètre amortissant les oscillations se manifeste par une réduction de moitié des oscillations apparaissant sur la carte d'élévation lors de modélisation de sols plats. Ainsi, les lignes de mesure sont moins visibles. La figure 4.18 met en évidence cet effet. Quantitativement, l'écart type des variations de la modélisation avec l'inclinomètre non-amorti est 2.2 fois plus élevé qu'avec l'inclinomètre amorti.

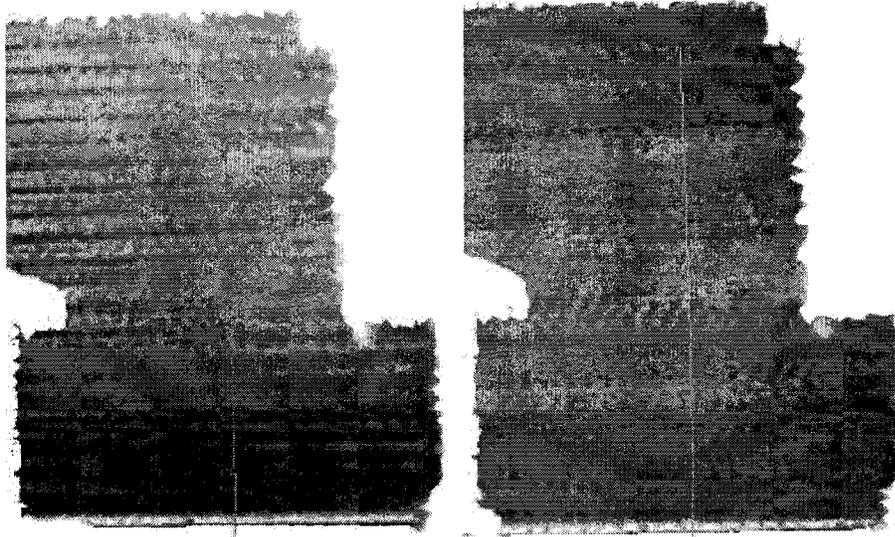


FIG. 4.18 Influence des inclinomètres sur la reconstruction. La modélisation sans amortissement (à gauche) est bien plus irrégulière que celle avec amortissement (à droite). L'amplitude des variations est statistiquement deux fois plus élevée dans le cas sans amortissement.

Ces filtre induisent aussi un retard du fait de leur constante de temps par rapport aux autres capteurs. Ce décalage est de l'ordre de 100 ms. Nous avons donc dû devancer les données de l'inclinomètre d'une mesure (le pas de mesure est de 0.1 s).

Ainsi, ces caractéristiques nous obligent à limiter l'inclinaison de notre robot (la plage d'opération de l'inclinomètre est située entre  $-25^\circ$  et  $25^\circ$ ) lors des expériences, mais surtout à ne pas se déplacer trop vite (5 cm/s au lieu de 20 cm/s) dans les terrains accidentés pour que les oscillations ne dépassent pas quelques hertz. D'autre part, le temps de réponse du capteur étant assez élevé, nous devons devancer les mesures de l'inclinomètre de 100 ms pour qu'elles soient en phase avec les autres capteurs.

#### 4.4 Validation des modes d'exploration

Les algorithmes de navigation et d'exploration ont été intensivement testés dans des environnements intérieurs et extérieurs. Cependant, les notions de réussite pour de tels algorithmes sont et reposent plus sur des constatations que des mesures. D'autre part, les algorithmes reposent sur de nombreux paramètres dont la valeur est le plus souvent issue d'un compromis entre ses différentes conséquences. Ainsi, les notions d'obstacle et de zones perturbées sont subjectives, leur frontière est floue et selon la situation un obstacle peut être ou pas considéré comme tel. Ainsi, un trottoir, sur lequel la plate-forme peut en pratique monter, est la plupart du temps considéré comme un obstacle comme dans le cas de la figure 4.20.

Les tests du mode de navigation réactive ont été très concluants, notamment au niveau de la sécurité. En effet, lors de nos expérimentations la plate-forme a évolué pendant de nombreuses minutes sans heurter d'obstacles. La figure 4.19 présente une trajectoire suivie par la plate-forme en mode réactif.

Le mode de navigation avec but s'est révélé tout aussi sécuritaire que le précédent, ce qui est tout à fait normal puisqu'il en dérive. La figure 4.21 présente un exemple typique : le robot doit se rendre à un point mais un obstacle est situé au milieu ; la plate-forme le contourne en l'évitant grâce au mode de navigation réactive et arrive au point-objectif.

Cet algorithme présente cependant quelques limites et il est possible de le mettre en défaut. Par exemple comme le montre la figure 4.22, la plate-forme se retrouve bloquée dans une cuvette et ne peut s'en dégager car l'algorithme développé évite les obstacles tout en essayant de s'approcher de l'objectif mais ne les contourne pas. Il s'agit du problème des minimums locaux des méthodes de navigation basées sur les potentiels (Latombe, 1991).

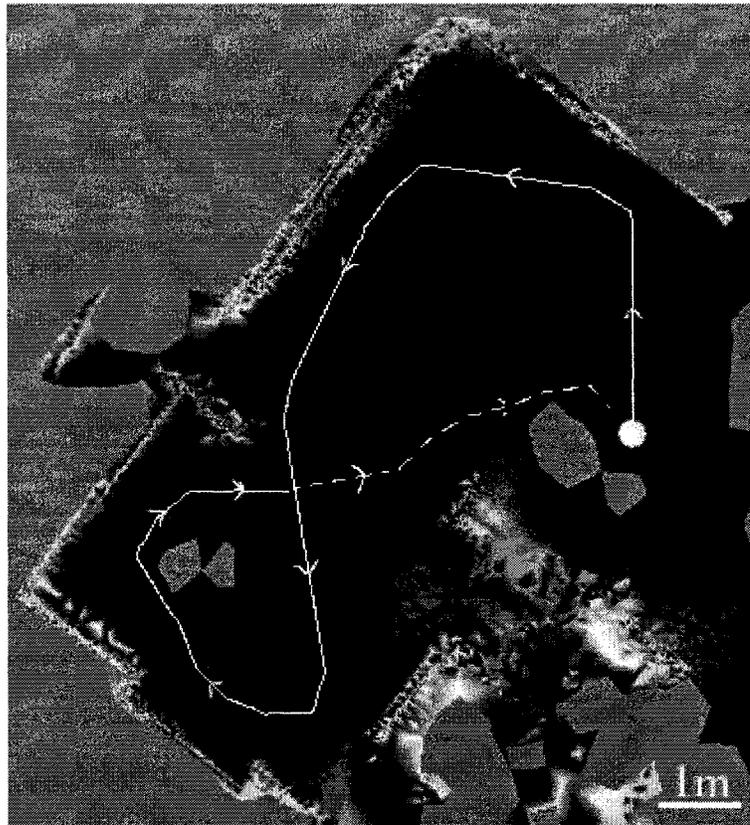


FIG. 4.19 Exemple de navigation intérieure en mode réactif (partie pleine de la trajectoire) puis en mode retour au point de départ (partie de la trajectoire en tirets)

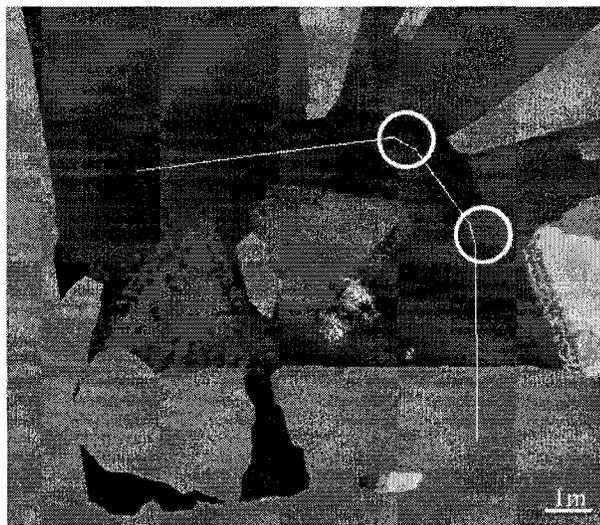


FIG. 4.20 Navigation réactive en milieu extérieur. Les zones encadrées représentent les changements de direction causés par le trottoir.



FIG. 4.21 Exemple de navigation avec but : la plate-forme contourne un obstacle pour atteindre son objectif.

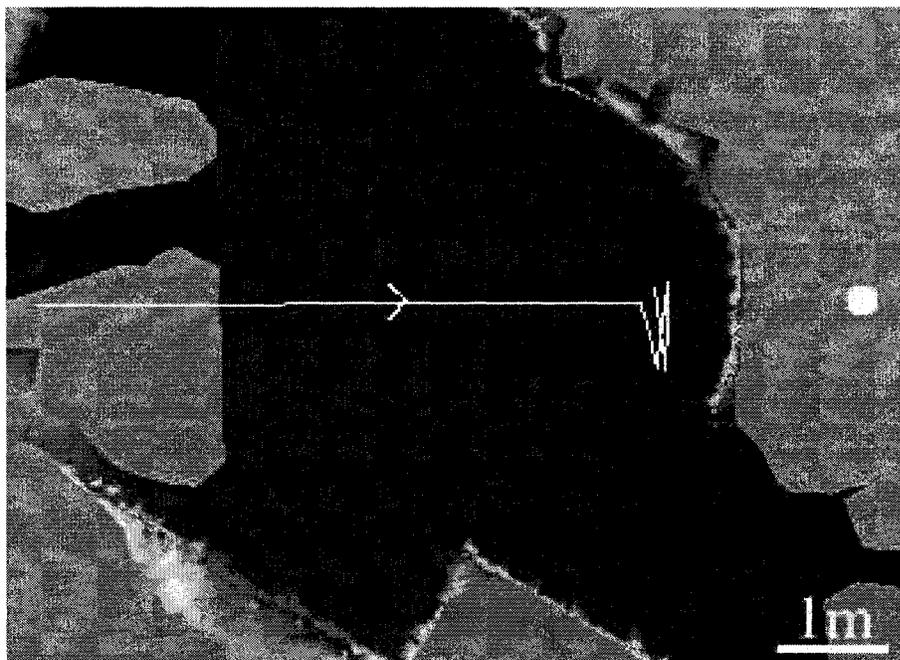


FIG. 4.22 Exemple de situation où la plate-forme n'arrive pas à atteindre le point-cible (le point blanc à droite).

Dans le cas de l'algorithme de navigation réactive avec but, il est possible de faire revenir la plate-forme à son point de départ en faisant correspondre l'objectif avec sa position initiale. Ainsi comme le montre la figure 4.23, la plate-forme revient à sa position initiale; cette position diffère toutefois sensiblement de la position d'origine car la plate-forme considère qu'elle a atteint son objectif lorsqu'elle est située à moins de 30 cm de celui-ci. C'est donc une bonne méthode d'estimation de la précision du module d'*odométrie 3D*; dans le cas de la figure 4.23, la plate-forme est arrivée à 40 cm de son point de départ (distance mesurée sur le terrain), l'odométrie a donc dévié de 10 cm au cours du parcours de 20 m environ, ce qui correspond à 0.5% d'erreur.

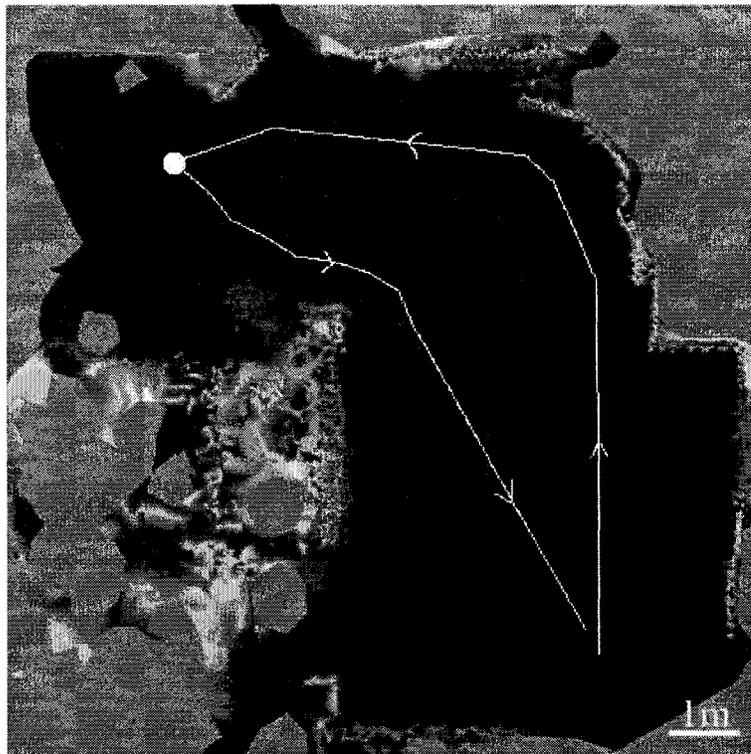


FIG. 4.23 Exemple de navigation réactive puis retour au point de départ à partir du point blanc

## CONCLUSION

Une étape majeure de la planification de trajectoire en robotique mobile est la définition de l'espace d'opération de la plate-forme, c'est-à-dire la cartographie ou encore la modélisation de l'environnement où elle évolue. Dans le cas d'environnements extérieurs non structurés une simple modélisation en deux dimensions n'est pas suffisante et il faut constituer un modèle tridimensionnel du terrain où évolue la plate-forme. Différentes raisons telles que des problèmes de communication entre la plate-forme et le poste de contrôle nous ont aussi incités à développer un système qui puisse cartographier ce terrain de la manière la plus autonome possible.

Dans l'exploration et la cartographie de milieux non structurés les principaux problèmes à résoudre sont les suivants : (1) définir des stratégies de navigation réactives prenant en compte les obstacles ; (2) garantir l'exhaustivité de l'exploration ; (3) acquérir les données, c'est-à-dire localiser la plate-forme, mettre les données dans un repère commun, les filtrer ; (4) représenter l'environnement grâce à une structure évolutive et facilitant l'accès aux données.

Pour résoudre ces problèmes, le système proposé est constitué d'un robot mobile équipé d'un télémètre scrutant le sol en avant. L'architecture fonctionnelle proposée est basée sur cinq modules : un module d'acquisition et de localisation qui détermine la position des points acquis à partir de la position et de l'attitude du robot, un module de modélisation basé sur une triangulation de Delaunay qui rassemble et ordonne ces points, un module de navigation et d'exploration basé sur la modélisation de l'environnement du module précédent, un module de contrôle du déplacement et un dernier d'interface et de visualisation permettant à l'opérateur de visualiser et de contrôler l'évolution de la plate-forme. Notre stratégie de division du système en modules permet de n'avoir que deux blocs dépendants du

matériel utilisé et donc de l'adapter sur une autre plate-forme avec un minimum de modifications.

Pour les expériences, notre plate-forme de test est composée d'un robot mobile sur lequel est monté un télémètre profilométrique incliné vers l'avant du robot. Ce télémètre acquiert des profils du terrain qui permettent de modéliser le terrain une fois rassemblés. Notre plate-forme est aussi équipée d'un inclinomètre et d'un gyromètre dont l'axe de sensibilité est vertical pour ainsi estimer sa trajectoire en trois dimensions.

Les résultats obtenus montrent que le système développé est capable de fournir des cartes de l'environnement précises au centimètre près. Cependant dans les cartes de grande taille (plusieurs dizaines de mètres) le facteur limitant est la localisation de la plate-forme qui a tendance à dériver. Aussi les comparaisons entre les vues offertes par le module de visualisation en trois dimensions et les photos des sites permettent d'apprécier qualitativement la modélisation. D'autre part les algorithmes de navigation se sont révélés très sécuritaires car après avoir optimisé les paramètres, la plate-forme ne heurte aucun obstacle. Les tests du mode de navigation avec objectif sont la plupart du temps positifs sauf dans certains cas où la plate-forme reste bloquée par un obstacle de forme concave. Enfin le module de recherche de nouvelles zones à explorer se révèle efficace.

Cependant notre système présente certaines limites qui peuvent servir de base à de futures recherches.

Notre modélisation du terrain n'est pas tout à fait tridimensionnelle mais plutôt en 2D 1/2 puisque l'environnement est projeté sur le plan horizontal. Ainsi lors de la modélisation de certaines scènes, notamment intérieures, certains points sont superposés et la modélisation s'en retrouve perturbée à cet endroit (comme avec

des tables et le sol). Pour régler ce problème, une piste serait d'utiliser une représentation intrinsèquement tridimensionnelle.

Il serait également intéressant d'associer aux points une couleur ou bien une texture afin de fournir à l'opérateur une représentation plus fidèle de l'environnement. Ceci pourrait en même temps permettre à l'opérateur d'avoir une visualisation plus fidèle de l'environnement exploré et permettre au système de différencier les types de terrain plus facilement et par exemple de déterminer si le sol est de l'herbe ou bien de la terre. Un point fondamental pour la modélisation serait d'améliorer la localisation de la plate-forme. Une solution consisterait à faire correspondre les profils lors de balayages de zones identiques pour corriger les erreurs de localisation relatives. Il s'agirait donc d'étendre les notions de localisation et de cartographie simultanées (SLAM) à des cartes tridimensionnelles. Sur le plan du système de navigation, il faudrait améliorer le module de navigation pour qu'il contourne les obstacles au lieu de les éviter. Aussi, nous avons développé un système de détection de nouvelles zones à explorer cependant notre système d'exploration n'est pas encore totalement autonome.

Ainsi quatre améliorations principales se profilent : (1) ajouter une information de couleur au maillage pour faciliter la visualisation et l'identification des terrains ; (2) utiliser une structure intrinsèquement tridimensionnelle ; (3) développer un algorithme de mise en correspondance des profils ; (4) développer un algorithme d'exploration entièrement autonome.

Enfin, les principales contributions scientifiques de notre projet sont le développement d'un système de cartographie en haute résolution et en temps réel de terrains non structurés ainsi que la mise en place d'algorithmes de navigation autonome à partir de la modélisation en cours de construction.

## RÉFÉRENCES

- Astolfi, A. (1994). On the stabilization of nonholonomic systems. *IEEE Conference on Decision and Control*.
- Bakambu, J. N. (2003). *Système intégré de navigation autonome dans un réseau de galeries : exploration, guidage et repérage naturel*. Thèse de doctorat, École Polytechnique de Montréal.
- Bakambu, J. N., Polotski, V., and Cohen, P. (2000). Heading-aided odometry and range-data integration for positioning of autonomous mining vehicles. *IEEE International Conference on Control Applications*.
- Boissonnat, J. D., Devillers, O., Teillaud, M., and Yvinec, M. (2000). Triangulations in CGAL. *Computational Geometry*.
- Borenstein, J. and Ye, C. (2002). A new terrain mapping method for mobile robots obstacle negotiation. Technical report, University of Michigan (USA).
- Burgard, W., Moors, M., Fox, D., Simmons, R., and Thrun, S. (2000). Collaborative multi-robot exploration. *IEEE Conference on Robotics and Automation*.
- Carrière, J.-J. (2004). Positionnement et construction de cartes 2d simultanés dans les réseaux de couloirs en utilisant les repères naturels. Mémoire de maîtrise, École Polytechnique de Montréal.
- Dissanayake, G., Durrant-Whyte, H., and Bailey, T. (2000). A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. *IEEE International Conference on Robotics and Automation (ICRA'00)*.
- Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*.
- Früh, C. and Zakhor, A. (2002). Data processing algorithms for generating textured 3D building façade meshes from laser scans and camera images. *IEEE Procee-*

*dings of the First International Symposium on 2D Data Processing Visualization and Transmission (3DPVT 02).*

George, P.-L. and Borouchaki, H. (1997). *Triangulation de Delaunay et maillage*. Hermès.

Guivant, J. and Nebot, E. (2003). Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms. *IEEE Transactions on Robotics and Automation*.

Heckbert, P. and Garland, M. (1997). Survey of polygonal surface simplification algorithms. *SIGGRAPH 97 Course Notes : Multiresolution Surface Modeling*.

Huber, D. and Hebert, M. (1999). A new approach to 3D terrain mapping. *IEEE International Conference on Intelligent Robotics and Systems (IROS '99)*.

Kuipers, B. J. and Byun, Y. T. (1988). A robust, qualitative method for robot spatial learning. *The Seventh National Conference on Artificial Intelligence (AAAI 88)*.

Lacroix, S., Mallet, A., Bonnafous, D., Bauzil, G., Fleury, S., Herrb, M., and Chatila, R. (2002). Autonomous rover navigation on unknown terrains, functions and integration. *International Journal of Robotics Research*.

Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers.

Lemaire, T. (2004). Robotique mobile : Localisation et cartographie simultanées. In *Colloque Ecole Doctorale Informatique et Télécommunication (EDIT)*.

Martin, C. and Thrun, S. (2002). Real-time acquisition of compact volumetric 3D maps with mobile robots. *IEEE Conference on Robotics and Automation*.

Mataric, M. J. (1990). *Distributed Model for Mobile Robot Environment-Learning and Navigation*. Thèse de doctorat, MIT.

Maurette, M. (2003). Mars rover autonomous navigation. *Autonomous Robots*.

- Nüchter, A., Surmann, H., and Hertzberg, J. (2003). Planning robot motion for 3D digitalization of indoor environments. Technical report, Fraunhofer Institute for Autonomous Intelligent Systems (Allemagne).
- Roy, N. and Dudek, G. (2001). Collaborative robot exploration and rendez-vous : Algorithms, performance bounds and observations. *Autonomous Robots*.
- Singh, S., Simons, R., Smith, T., Stentz, A., Verma, V., Yahja, A., and Schwehr, K. (2000). Recent progress in local and global traversability for planetary rovers. *IEEE Conference on Robotics and Automation*.
- Surmann, H., Nüchter, A., and Hertzberg, J. (2003). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Journal Robotics and Autonomous Systems*.
- Thrun, S. (2002). Robotic mapping : A survey. In Lakemeyer, G. and Nebel, B., editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.
- Thrun, S., Burgard, W., and Hähnel, D. (2001). Learning compact 3d models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*.
- Thrun, S., Hähnel, D., Ferguson, D., Montemerlo, M., Triebel, R., Burgard, W., Baker, C., Omohundro, Z., Thayer, S., and Whittaker, W. (2003). A system for volumetric robotic mapping of abandoned mines. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- Welch, G. and Bishop, G. (2003). An introduction to the Kalman filter. *Department of Computer Science TR95-041, University of North Carolina at Chapel Hill*.

## ANNEXE I

## SIMULATION UNIDIMENSIONNELLE ET FILTRE DE KALMAN

## I.1 Simulation unidimensionnelle

Au début du projet nous avons simulé le processus avec Matlab : pour simplifier le problème nous nous sommes limités à une simulation unidimensionnelle. Le terrain est modélisé par une ligne brisée ; la position et l'attitude de la plate-forme sont alors calculées à partir du rayon des roues et de leur espacement (voir la figure I.1). Un point délicat est le positionnement des roues sur les parties anguleuses où les roues peuvent rouler (voir la figure I.2). L'inclinomètre est alors simulé en calculant l'inclinaison de la plate-forme.

Les mesures effectuées par le télémètre sont aussi simulées en calculant l'intersection entre le rayon (calculé à partir de la position, de l'attitude de la plate-forme et des paramètres de positionnement du capteur) et le profil modélisé du terrain. Le processus de modélisation du terrain par la plate-forme est alors simulé dans le cas

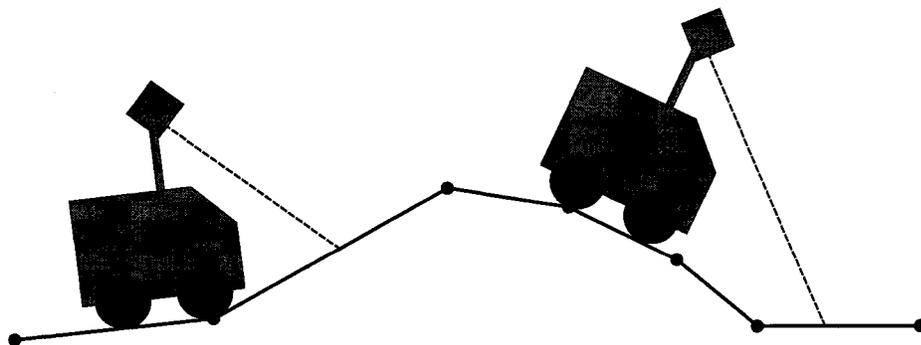


FIG. I.1 Simulation unidimensionnelle de la plate-forme sur un terrain modélisé en segments de droite.

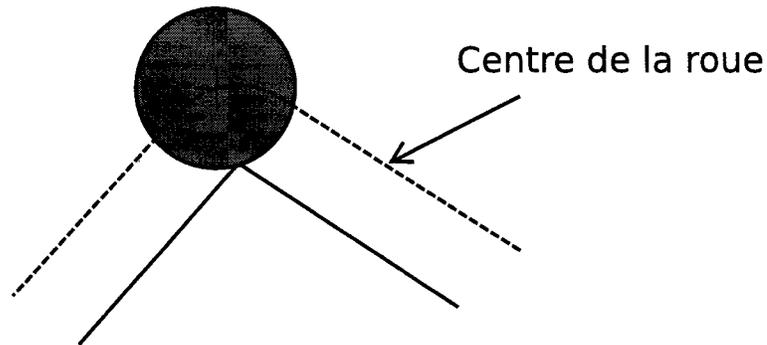


FIG. I.2 Simulation de la roue sur une partie anguleuse.

unidimensionnel.

Pour rendre la simulation plus réaliste nous avons aussi rajouté un bruit de mesure aux données des capteurs.

Ce modèle de simulation a ensuite été utilisé pour tester la viabilité du système ainsi que la possibilité d'utiliser la modélisation du terrain par la plate-forme pour calculer sa hauteur ainsi que son attitude.

## I.2 Filtre de Kalman

L'objectif est de combiner les données de l'inclinomètre et de la modélisation de la surface du sol pour obtenir une meilleure estimation de la hauteur de la plate-forme. Le modèle de simulation est celui décrit à la section précédente.

Les variables sont définies comme suit :

- $h_k$  est la hauteur de la plate-forme à l'instant  $k$ ,
- $dp_k$  est le déplacement mesuré par l'odomètre entre les instants  $k - 1$  et  $k$ ,
- $\phi_k$  est l'angle de tangage à l'instant  $k$  fourni par l'inclinomètre,
- $w$  est le bruit de mesure de l'inclinomètre de variance  $Q$ ,

–  $z_k$  est la hauteur mesurée à partir de la modélisation avec une variance  $R$ .

Le modèle d'état du système calcule la hauteur de la plate-forme en intégrant les données de l'inclinomètre :

$$h_k = f(h_{k-1}, \phi_k, w_k) = h_{k-1} + dp_k \sin(\phi_k + w_k) \quad (\text{I.1})$$

Le modèle de mesure s'écrit :

$$z_k = h(h_k, v_k) = h_k + v_k \quad (\text{I.2})$$

La phase de *prédiction* du filtre de Kalman étendu devient <sup>1</sup> :

$$h_k = h_{k-1} + dp_k \sin(\phi_k) \quad (\text{I.3})$$

$$\begin{aligned} P_k^- &= A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \\ &= P_{k-1} + (dp \cos(\phi_k - 1))^2 Q \end{aligned} \quad (\text{I.4})$$

et la phase de *correction* :

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \\ &= P_k^- / (P_k^- + R) \end{aligned} \quad (\text{I.5})$$

$$h_k = h_k^- + K_k (z_k - h_k) \quad (\text{I.6})$$

$$\begin{aligned} P_k &= (I - K_k H_k) P_k^- \\ &= (1 - K_k) P_k^- \end{aligned} \quad (\text{I.7})$$

avec

$$- A_k = \frac{\partial f}{\partial x_k} = 1$$

---

<sup>1</sup>Bien que les données soient en pratique unidimensionnelles, la notation matricielle est utilisée pour conserver la généralité et le formalisme des filtres de Kalman.

$$\begin{aligned}
- H_k &= \frac{\partial h}{\partial x_k} = 1 \\
- W &= \frac{\partial f}{\partial w} = dp \cos(\phi_k - 1) \\
- V &= \frac{\partial h}{\partial v} = 1
\end{aligned}$$

Cependant si nous considérons l'erreur de mesure de l'inclinomètre de  $0.5^\circ$  (ce qui est grand par rapport à la réalité), soit  $Q = (\pi/(2 * 180))^2 = 0,0002$ , un pas de déplacement de 5 cm, soit  $W \approx dp = 0.05$  et une erreur de mesure de la hauteur de la plate-forme à partir de la modélisation de l'ordre de 2 cm, soit  $R = 0.0004$ , nous constatons que les données venant de la structure sont quarante fois moins fiables que celles venant de l'inclinomètre. Cela explique pourquoi la combinaison de ces deux données n'apporte pas d'amélioration notable. Cependant sur le plan théorique cette étude reste très intéressante.

## ANNEXE II

### PRÉSENTATION DU MATÉRIEL UTILISÉ

#### II.1 Présentation du robot ATRV

Ce robot est produit par la société *Real World Interface* sous le nom ATRV-2. Il mesure 1 m de long, 80 cm de large et 65 cm de haut sans équipement.

Il est équipé de quatre roues fonctionnant deux par deux ce qui lui donne la possibilité de tourner sur lui-même. Ces roues sont aussi équipées d'encodeurs, ce qui permet d'obtenir le déplacement du robot ; ce module est appelé l'*odomètre*. Par moyenne entre les roues nous obtenons le déplacement longitudinal du robot et par différence son déplacement angulaire (lacet).

À bord est aussi installé un ordinateur qui permet de lire les données de tous les capteurs, d'effectuer le traitement de l'information et de commander les actionneurs (les moteurs des roues). Cet ordinateur roule sous Red Hat Linux et est constitué de deux processeurs cadencés à 350 MHz. Cet ordinateur est équipé d'un module sans-fil lui permettant d'être connecté au réseau à une vitesse de 3 Mb/s.

#### II.2 Télémètre laser

Un télémètre laser est un capteur qui mesure des distances entre lui et un objet à l'aide d'un laser. La mesure est effectuée en envoyant une impulsion laser très courte, qui est ensuite retransmise par l'objet visé. Il suffit alors de mesurer le temps séparant l'émission de la réception. Les mesures sont effectuées sur un demi-plan

en utilisant un miroir rotatif qui oriente le faisceau dans les directions souhaitées.

Le télémètre laser que nous utilisons, le LMS 200 de la compagnie allemande SICK, couvre un demi-plan (voir la figure 2.3) : nous l'utilisons dans le mode où il prend une mesure à chaque degré sur  $180^\circ$ . Nous appellerons *scan* cet ensemble de 181 points qui représente une coupe de l'environnement. Ces coupes sont acquises à un rythme de 10 scans par seconde et avec une précision de l'ordre du centimètre.

Dans notre installation (voir la figure 1.5) le télémètre est pointé vers l'avant du robot en direction du sol avec un angle d'environ  $30^\circ$  avec le plan horizontal et perché à 1,15 m du sol. Ainsi, le télémètre permet au robot de *voir* un peu plus de 2 m devant lui.

Il faut ensuite appliquer quelques transformations aux données pour obtenir les coordonnées des points mesurés dans le repère global. Ces calculs sont développés dans le chapitre 2.

### II.3 Le gyromètre

Ce capteur vient compléter l'odomètre qui souffre d'un certain manque de précision en lacet (rotation autour de l'axe vertical du robot). Comme son axe de sensibilité est orienté verticalement, le gyromètre fournit les variations de déplacement en lacet.

Son principe de fonctionnement est basé sur l'interféromètre de Sagnac (voir la figure II.1) : des fibres optiques enroulées autour de l'axe de sensibilité et une source lumineuse est dirigée de façon à ce que sa lumière traverse la fibre dans les deux sens. Une mesure entre la différence de phase entre un rayon passant dans un sens et un autre passant dans l'autre permet d'en déduire la vitesse de rotation

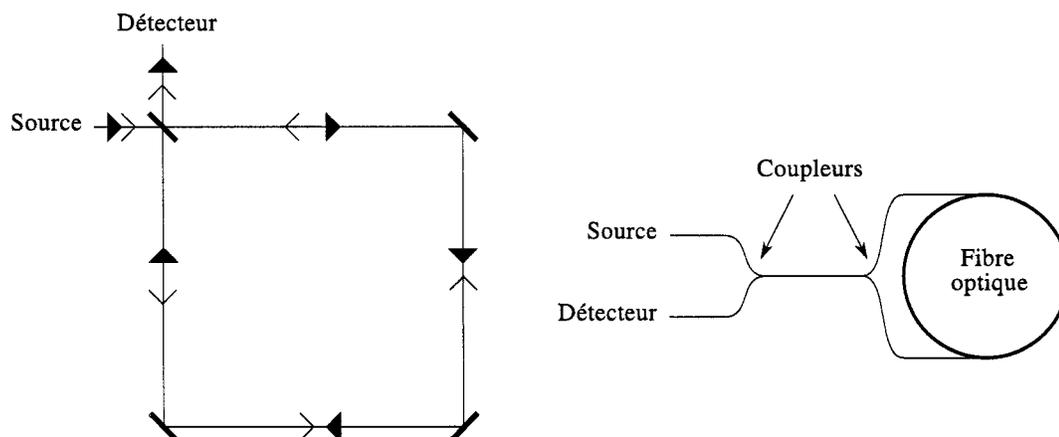


FIG. II.1 Principe de fonctionnement de l'interféromètre de Sagnac (à gauche) et le dispositif utilisé dans l'Autogyro de KVH (à droite)

autour de cet axe. Les deux rayons se rejoignent alors au niveau du détecteur pour interférer (interférences constructives et destructives) ce qui se traduit au niveau du détecteur par une simple variation d'intensité lumineuse directement proportionnelle à la différence de phase. Le modèle utilisé est l'*Autogyro* de la compagnie *KVH*.

#### II.4 L'inclinomètre

Son principe de fonctionnement repose sur un liquide électrolytique conducteur situé à l'intérieur. Pour obtenir l'inclinaison du boîtier par rapport au plan horizontal, le capteur mesure l'inclinaison de la surface du liquide par rapport au boîtier. Ainsi nous obtenons les angles de tangage et de roulis du robot en mesurant son inclinaison selon les deux axes perpendiculaires.

Au cours du projet nous avons changé l'inclinomètre pour un autre avec un fluide plus visqueux. Ce changement a permis de diminuer les oscillations parasites et d'avoir des mesures plus stables. La figure 4.18 permet de comparer les données

obtenues avec chacun des capteurs.

Le modèle utilisé est le *Digital/Analog Clinometer MD900-T* de la compagnie *Applied Geomechanics* (voir la figure 1.6). Ce modèle présente comme caractéristiques une plage de fonctionnement comprise entre  $-25^\circ$  et  $25^\circ$ , une constante de temps de 0.15 s (la sortie est proportionnelle à  $1 - e^{-\frac{t}{\tau}}$ ) et une fréquence propre de 10 Hz.

## ANNEXE III

### MANUEL D'UTILISATION DU PROGRAMME

Ce manuel décrit l'utilisation du programme d'exploration et de modélisation en trois dimensions de terrains non structurés.

#### III.1 Utilisation sur le terrain

Le programme est constitué de deux parties :

- une partie qui est exécutée à bord du robot ;
- une partie qui est exécutée à bord de l'ordinateur distant.

##### III.1.1 Lancement du programme sur le robot

Tout d'abord il faut allumer les capteurs avec les interrupteurs à l'arrière gauche du robot (les deux à l'extérieur) et tourner la clé pour permettre au robot d'avancer (elle doit être horizontale).

Se connecter sur le robot en tapant `telnet herakles` et en se connectant sous l'identifiant `mobility` et le mot de passe `grprcar5`. Aller dans le répertoire DEMO en tapant `cd Projets_Thomas\DEMO`

Répéter quatre fois ces opérations pour avoir cinq consoles ouvertes.

Lancer une fois pour toutes `name -i .`

Maintenant il faut lancer les serveurs pour les différents capteurs : dans quatre des

consoles lancer `topsick`, `clino`, `gyro` et `base`. Ces serveurs pourront être fermés à la fin de la démo en pressant "q" puis "entrée". Le serveur `topsick` est un peu long à se lancer (30 secondes) et le message "ERROR !! End Incorrect CRC !" s'affiche lorsqu'il est prêt, ce n'est pas un message d'erreur.

La console restante va servir à lancer le programme principal.

Il est tout d'abord possible de déplacer le robot en utilisant le programme `move`. Les touches de commande sont les chiffres "8", "4", "6", "2" et "5" (pour s'arrêter) et "0" pour quitter.

Le programme principal se lance par `acq nomfichier` où `nomfichier` représente le radical des fichiers où seront enregistrés les données.

### III.1.2 Lancement du client de visualisation

Le client de visualisation s'appelle `3DVisu`. Il se lance en tapant `./3dVisu herakles` (`herakles` est le nom du robot) ou `./3dVisu 132.207.24.6`.

Une fenêtre OpenGL s'ouvre alors et affiche le robot. Cette fenêtre est interactive et les commandes sont :

- le bouton gauche pour tourner autour de la scène,
- le bouton droit pour translater la scène,
- le bouton central pour zoomer.

Des messages apparaissent sur la console ainsi que sur la console du programme principal pour indiquer que le client de visualisation est bien connecté au programme principal.

### III.1.3 Exploration

Le système est alors prêt pour déplacer le robot et explorer.

Sur la console du programme principal il est alors possible de contrôler le déplacement du robot comme avec le programme `move` avec les touches du clavier. Sur l'interface de visualisation la construction de la carte apparaît en direct (la carte est rafraîchie toutes les 5 secondes) et on peut observer le robot se déplacer.

Le mode automatique peut être lancé en appuyant sur la touche "a", ce mode explore l'environnement de façon réactive en changeant sa direction lorsqu'il se trouve face à un obstacle. Avant de lancer ce mode il faut attendre que le robot ait parcouru 2m environ pour qu'il ait assez de données sur le sol pour déterminer les obstacles ; sinon il s'arrêtera.

On peut définir un point-objectif au moyen de l'interface de visualisation en cliquant sur la carte avec le bouton gauche tout en maintenant la touche "Ctrl" enfoncée. On peut alors se diriger vers ce point en pressant "h" dans la console du programme principal. Si aucun point n'a été défini, la plate-forme se dirige vers le point de départ (de coordonnées (0,0)). Le robot s'arrête lorsqu'il a atteint le point (dans une zone de 20cm autour). Dans ce mode le robot effectue une navigation réactive pour se rendre au point-objectif, c'est-à-dire qu'il prend en compte les obstacles et les contourne.

Presser la touche "f" permet de détecter les frontières entre les zones explorées et les zones inexplorées et de se diriger vers celle qui est susceptible d'apporter le maximum d'information. Une fois arrivé à cette zone, le robot repasse en mode d'exploration réactive (comme si la touche "a" avait été pressée). Ce mode est efficace s'il est lancé après un certain temps d'exploration, quand la plate-forme a déjà rencontré des obstacles. Cette partie n'est pas encore entièrement automatique.

Pour quitter le programme principal il faut presser la touche "0". Une invite apparaît alors demandant la résolution de la carte d'élévation résultant de la modélisation. Une valeur correcte est de 20 pour les environnements extérieurs et de 40 dans les environnements intérieurs.

Lorsque le programme se termine il crée plusieurs fichiers dans le répertoire `maps/` du répertoire courant. Entre autres se trouve les fichiers `nomfichier.dat` contenant les enregistrements des données des capteurs (utilisé dans le mode hors ligne comme source de données), `nomfichier.dat.pnm` qui est une image représentant la carte d'élévation et enfin `gtm.pnm` qui est une image représentant la carte de traversabilité globale.

### III.2 Utilisation en mode hors-ligne

Un mode hors-ligne permet de simuler le programme sur des données prises lors d'une capture précédente en utilisant les fichiers de données enregistrés dans le mode en ligne. Comme ce programme utilise des données déjà acquises, il n'est pas nécessaire de lancer les quatre serveurs des capteurs (topsick, clino, gyro et base).

Pour cela il suffit de lancer `acq -o donnees.dat fichiersortie` où `donnees.dat` est le fichier de données (enregistré en général dans le répertoire `maps`) et `fichier-sortie` est le radical des fichiers créés par le programme.

Il est alors possible de lancer le client de visualisation. Pour faire avancer le robot il suffit de peser sur une touche de mouvement ("8" par exemple).

### III.3 Conseils

#### III.3.1 Utilisation sur l'ordinateur Artemis

Se connecter avec l'identifiant `demo` et le mot de passe `demogrpr`.

Avec cet ordinateur il suffit de se connecter une fois en telnet sur le robot puis de lancer le script `artemis.sh` du répertoire DEMO du robot. Cinq consoles s'ouvrent alors automatiquement et il n'est plus nécessaire de lancer le service `name -i`.

#### III.3.2 Utilisation en extérieur

Lors d'une sortie à l'extérieur il faut installer le module sans fil AP sur le robot et le connecter par un câble croisé. L'ouverture des terminaux prend du temps : il faut attendre 3 minutes avant de pouvoir entrer l'identifiant et le mot de passe ; et ATTENTION il n'y a qu'une minute pour le rentrer sinon il faut recommencer et encore attendre 3 min!!!

En extérieur éviter d'utiliser le robot ATRV sur l'herbe car il a tendance à labourer le sol.

#### III.3.3 Problèmes possibles

Il est possible que le programme principal n'arrive pas à se lancer. C'est la plupart du temps à cause du programme "name" qui s'est arrêté. Il faut donc le relancer en tapant `name -i` et en relançant tous les serveurs des capteurs.