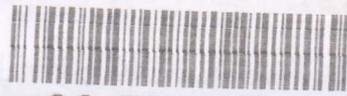


32
REPUBLICQUE ALGE



32-530-27-3

POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

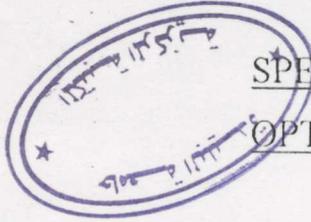
UNIVERSITE DE BLIDA

INSTITUT D'ELECTRONIQUE

MEMOIRE DE MAGISTER

SPECIALITE : ELECTRONIQUE

OPTION : COMMUNICATION



06

EXGLU DU PRAT



THEME

**CONTRIBUTION AU DEVELOPPEMENT
D'UN ROBOT MOBILE SEMI AUTONOME
ET SIMULATION D'UN PLANIFICATEUR
REACTIF DE TRAJECTOIRES**

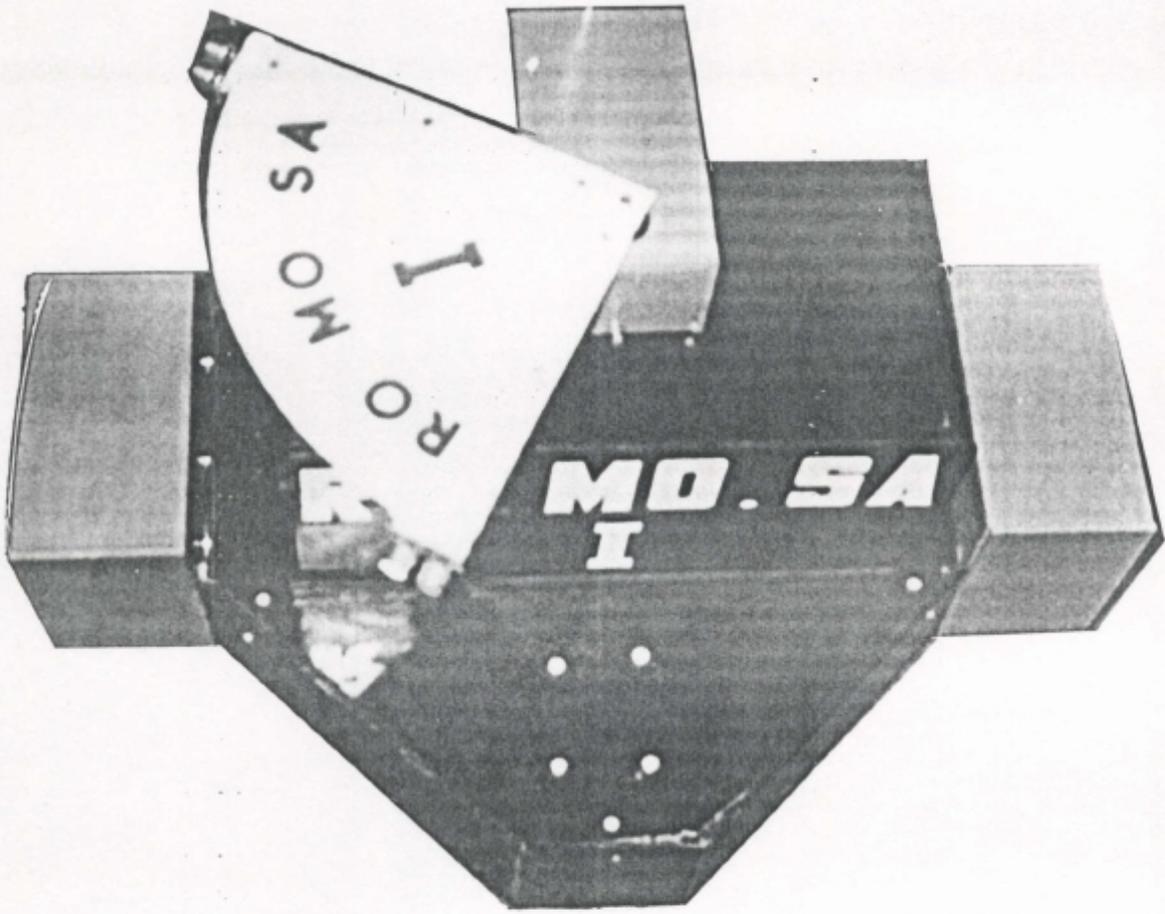
PAR
BENNILA NOUR-EDDINE

PROPOSE PAR
ARBAOUI AMAR

PRESENTEE DEVANT LE JURY:

Mr H. SALHI	PRESIDENT
MR OUGUINI	EXAMINATEUR
Mr O. NADJEMI	EXAMINATEUR
Mr A. MENACEUR	EXAMINATEUR
Mr K. AMMOUR	RAPPORTEUR

BLIDA, ALGERIE 1995



DEDICACES

A mes parents et à toute ma famille

REMERCIEMENTS

Ce mémoire présente l'essentiel des travaux que j'ai menés, avec le soutien du Docteur AMAR ARBAOUI qui a été à l'origine de ce projet:

Je tiens également à remercier le Docteur AMMOUR KAMEL pour ces suggestions et l'honneur qu'il m'a fait en acceptant la charge d'être rapporteur de cette thèse.

Merci également à:

* Monsieur SALHI HASSEN maître de conférence à l' institut d'électronique Université de BLIDA. pour sa documentation, et d'avoir accepter la charge de participer et présider le jury de cette thèse.

* Monsieur MENACEUR ABDERRAHMANE chargé de cours à l'institut d'électronique université de BLIDA. pour ces remarques constructives et enrichissantes.

Je voudrais aussi exprimer toute ma gratitude à Monsieur OMAR NEDJMI, Directeur des études à l' institut de physique Université de BLIDA, et Monsieur RACHID OUGUINI Secrétaire général du Centre de Développement des Techniques Avancées (CDTA), qui m'ont fait l'honneur de cautionner ce travail en acceptant de le juger.

CAHIER DES CHARGES

Le travail demandé dans ce projet, est le développement d'un robot mobile semi-autonome, et simulation d'un planificateur réactif des trajectoires, destiné à un robot mobile à deux roues motrices situées de part et d'autre de l'axe médian (rotation possible sur place). Les trajectoires obtenues, doivent mener le robot de la source à la destination sans collision avec les obstacles imprévus par le modèle des connaissances, dans le cas où aucun obstacle imprévu se présente, ces trajectoires doivent être optimales dans le sens "temps optimal" entre les buts partiels, et optimales dans le sens "distance euclidienne" au but final.

TABLE DES MATIERES

INTRODUCTION	1
CHAPITRE I	
LES ARCHITECTURES EN ROBOTIQUES MOBILES	
1 INTRODUCTION.....	-4-
2 LES ARCHITECTURES PUREMENTS RÉACTIVES.....	4
3 LES ARCHITECTURES POSSEDANT UNE FONCTION DE PLANIFICATION RÉACTIVE.....	5
4 CONCLUSION.....	19
CHAPITRE II	
NAVIGATION ET GUIDAGE DES ROBOTS MOBILES AUTONOMES	
1 INTRODUCTION.....	21
2 NAVIGATION SUR DES TERRAINS PARTIELLEMENTS CONNUS.....	21
2-1 INTRODUCTION.....	21
2-2 L'APPROCHE DEVELOPPÉE PAR KEISY.....	22
2-3 L'APPROCHE VISUELLE DE SUZUKI ET ARIMOTO.....	25
2-4 L'APPROCHE DEVELOPPÉE PAR BORENSTEIN ET KOREN.....	27
2-5 L'APPROCHE DEVELOPPÉE PAR UN ROBOT DE SURVEILLANCE PAR CROWLEY.....	34
3 NAVIGATION SUR DES TERRAINS INCONNUS.....	38
3-1 INTRODUCTION.....	38
3-2 NAVIGATION SUR DES TERRAINS INEXPLORÉS D'APRÉS LYENGAR ET RAO.....	38
3-3 LE GUIDAGE SONORE D'APRÉS KUC ET VIARD.....	41
3-4 EVITEMENT RAPIDE DE COLLISION D'APRÉS BORENSTEIN ET KOREN	45
4 CONCLUSION.....	50
CHAPITRE III	
ARCHITECTURE ET CARACTÉRISTIQUES MATÉRIELLES DE ROMOSA.I	
1 INTRODUCTION.....	55
2 ARCHITECTURE.....	55

2-1 GENERALITES.....	55
2-2 L'ARCHITECTURE DEVELOPPÉE.....	56
3 CARACTÉRISTIQUES MATÉRIELLES DE ROMOSA.I.....	58
3-1 PLATE FORME MÉCANIQUE.....	58
3-2 EQUIPEMENTS EMBARQUES.....	60
4 LIGNE DE COMMUNICATION HOMME-MACHINE.....	65
4-1 CODAGE DES INFORMATIONS.....	65
4-2 TRANSMISSION EN LIGNE.....	66
4-3 ETABLISSEMENT DE LA LIGNE OPERATEUR-ROMOSA.I.....	66
5 CONCLUSION.....	68

CHAPITRE IV

DEVELOPPEMENT D'UN PLANIFICATEUR RÉACTIF POUR LA NAVIGATION D'UN ROBOT MOBILE AUTONOME

1 INTRODUCTION.....	69
2 REPRESENTATION DU MODELE DE CONNAISSANCE.....	70
3 PLANIFICATION DES TRAJECTOIRES.....	72
4 PERCEPTION DE L'ENVIRONNEMENT.....	76
4-1 INTRODUCTION.....	76
4-2 LOIS PHYSIQUES DES SONARS.....	77
4-3 ACQUISITION ET INTERPRETATION DES DONNEES ISSUS DU SONAR	
5 REACTION FACE AUX EVENEMENTS IMPREVUS.....	88
6 CONTROLE D'EXECUTION ET LOCALISATION.....	93
6-1 INTRODUCTION.....	93
6-2 CONTROLE D'EXECUTION.....	93
6-2-1 INTRODUCTION.....	93
6-2-2 LES EQUATIONS CINEMATIQUE DU MOUVEMENT DU ROBOT	93
6-2-3 LE CONTROLE OPTIMAL.....	95
6-2-4 CARACTERISATION DES TRAJECTOIRES "BANG-BANG"	
a) INTRODUCTION	101
b) CALCUL DES VARIABLES AUXILIAIRES.....	102
c) LES TRAJECTOIRES "BANG-BANG" A TROIS (3) . "INSTANTS DE COMMUTATION".....	103
d) MISE EN EVIDENCE DE L'OPTIMALITE DES TRAJECTOIRES "BANG-BANG" A TROIS "INSTANTS DE COMMUTATION" TYPE+-	107

6-2-5 APPLICATION DE LA METHODE AU PLANIFICATION DES TRAJECTOIRES.....	109
6-3 MODELISATION PROBABILISTE DE L'ODOMETRIE.....	114
7 RESULTATS DE SIMULATION.....	119
8 CONCLUSION.....	121

CONCLUSION

ANNEXE A

A-1 MODELISATION DU MOTEUR.....	125
A-2 DESCRITISATION DU MODELE.....	126
A-3 IDENTIFICATION DES PARAMETRES DU SYSTEME.....	127
A-4 CONCEPTION DU CONTROLEUR.....	128
A-4-1 CONTROLEUR A RETOUR D'ETAT.....	128
A-4-2 LE CONTROLEUR P.D.....	129
A-5 CONCLUSION.....	133

introduction

le pouvoir de faire déplacer un robot mobile d'un point source à un point but en sécurité est le point focale de toutes les recherches en robotique mobile, souvent le robot doit atteindre son objectif en un temps spécifié, avec une telle contrainte, si le robot est ordonné à maintenir une vitesse raisonnable faible, il doit continuellement progresser tout droit au but tout en évitant les obstacles.

Le robot autonome doit, avant tout, être capable de décider de ses actions de façon raisonnée, puis de les exécuter dans un contexte réel. Le processus de planification s'appuie sur des modèles internes représentant le monde qui entoure le robot, la connaissance de l'environnement est parfois incomplète, il est donc nécessaire de contrôler l'applicabilité de chaque action avant et pendant son exécution, le robot doit donc être capable de détecter et de réagir aux incompatibilités entre les situations réelles et les situations planifiées.

Réagir aux événements imprévus peut conduire à modifier profondément la trajectoire ou le plan d'action établi par le planificateur; Il faut donc contrôler à tout moment l'activité du robot pour assurer une bonne réactivité.

En toute globalité, la navigation peut être résumée dans les réponses à ces trois questions: "où suis je?", "où est ce que je vais?", et "comment je l'atteint?".

Les travaux présentés dans ce mémoire portent sur, l'analyse et le développement d'une architecture d'un robot mobile autonome, l'analyse et réalisation d'un support expérimental que nous avons baptisé **RoMoSA.I** (Robot Mobile Semi-Autonome I), l'analyse et le développement d'une stratégie de scrutation des ultrasons permettant une bonne détection des situations imprévus, et le développement et la simulation d'un planificateur réactif assurant la convergence

au but quelque soit l'incompatibilité entre les situations réelles et les situations planifiées.

Pour décrire nos travaux nous avons organisé ce mémoire en quatre(4) chapitres.

Le premier chapitre présente une analyse de quelques concepts architecturaux en robotique mobile où l'autonomie est mise en avant. Ces architectures font apparaître trois sous systèmes: . Un planificateur qui prédétermine dans un univers idéalisé les tâches à accomplir. Un contrôleur d'exécution réactif qui gère l'exécution de chaque tâche en tenant compte du contexte réel et assure la mise en oeuvre des réactions du robot face aux événements non prévus par le planificateur. Une couche fonctionnelle qui procède à l'exécution des actions décidées par le contrôleur d'exécution.

Le second chapitre propose une organisation de quelques approches focalisées autour de la navigation et le guidage des robots mobiles; Ces techniques s'étendent des algorithmes où la connaissance même partielle du terrain est nécessaire, aux algorithmes les plus complexes où le terrain est complètement inconnu.

Le troisième chapitre traite le contexte de nos travaux dans le cadre de l'analyse et la réalisation du support expérimental **RoMoSA.I**, nous y proposons un concept architectural flexible et applicable à tous les robots mobiles autonomes, nous y détaillons par la suite les constituants mécaniques et électroniques de **RoMoSA.I**.

Le quatrième chapitre décrit l'analyse, le développement et la simulation du planificateur réactif proposé à la navigation et au guidage des robots mobiles autonomes, nous y détaillons tous les éléments nécessaires à sa réussite. A la fin de ce chapitre nous présentons les résultats obtenus par simulation.

A ces chapitres, nous avons ajouté une ANNEXE qui décrit les équations employées pour établir le modèle du contrôleur adaptative proposé pour le contrôle de la motricité de **RoMoSA.I.**

CHAPITRE 1

LES ARCHITECTURES EN ROBOTIQUE MOBILE

1 INTRODUCTION

Ordonné à atteindre un objectif, le robot mobile doit avant tout, être capable de décider de ces actions de façon raisonnée, puis les exécutées dans un contexte réel. Ces actions sont organisées en 3 types d'architectures:

1) les architectures où la réactivité est totalement absente, dans ce cas l'environnement est à priori connu par le robot et une fonction de planification génère les trajectoires devant être prises par le robot.

2) les architectures purement réactives ici le mécanisme de planification n'existe pas et le robot ne fait que réagir en permanence aux modifications de son environnement.

3) les architectures possédant une fonction de planification réactive; ici le planificateur génère les trajectoires et contrôle leurs exécutions. Ces trajectoires peuvent être modifiées si le robot rencontre un obstacle imprévu par le modèle.

Nous nous intéressons dans ce qui suit, uniquement aux architectures purement réactives et les architectures utilisant une fonction de planification réactive.

2 LES ARCHITECTURES PUREMENT RÉACTIVES

Les approches dites purement réactives se basent sur une hiérarchie de comportements réagissant à des excitations, internes ou externes.

BROOKS développa en 1985 [BRO 85] une architecture qui s'inspire des études menées, en éthologie, sur le comportement animal. Dans cette approche, les réactions sont organisées en une hiérarchie de niveaux, chacun correspondant à un comportement unique qui peut inhiber les entrées et les

sorties des activités du niveau immédiatement inférieur pour engendrer des réactions complexes.

La **figure I-1** illustre ces mécanismes, elle montre comment réaliser une machine qui, sans planification ni contrôle explicite, erre en évitant les obstacles.

Influencé, par l'approche de BROOKS et à partir de la méthode des potentiels développée par KHATIB en 1985 [**KHA 85**], CONNELL a conçu deux comportements qui s'adaptent à la structure de BROOKS. Le premier permet de suivre des parois, et le second de détecter et franchir des portes [**BRO 86**].

Toujours dans le cadre de cette même approche PAYTON et MITCHELL en 1986 [**PAY 86**] [**MIT 87**], ont proposé une architecture qui se démarque de celle de BROOKS par la présence d'un mécanisme qui permet de choisir explicitement les comportements à inhiber.

PAYTON et MITCHELL ont défini la notion de comportement à l'aide de capteurs virtuels qui sont associés à des comportements réflexes. Les capteurs virtuels reconnaissent les caractéristiques de l'environnement à partir des données brutes issues des capteurs; Les comportements réflexes utilisent ces résultats pour produire les commandes destinées aux effecteurs. Ces commandes sont centralisées au niveau d'une base commune, et un module particulier sélectionne la commande adéquate.

Ces dernières années BROOKS s'est intéressé au comportement d'une colonie de robots totalement autonomes et faiblement complexes, coopérant sans coordination explicite à la réalisation d'une tâche particulière, cette fois ci BROOKS s'est appuyé sur l'étude de colonies de fourmis.

3 LES ARCHITECTURES POSSEDANT UNE FONCTION DE PLANIFICATION RÉACTIVE

Les architectures qui comportent une fonction de planification indépendante disposent d'une représentation de

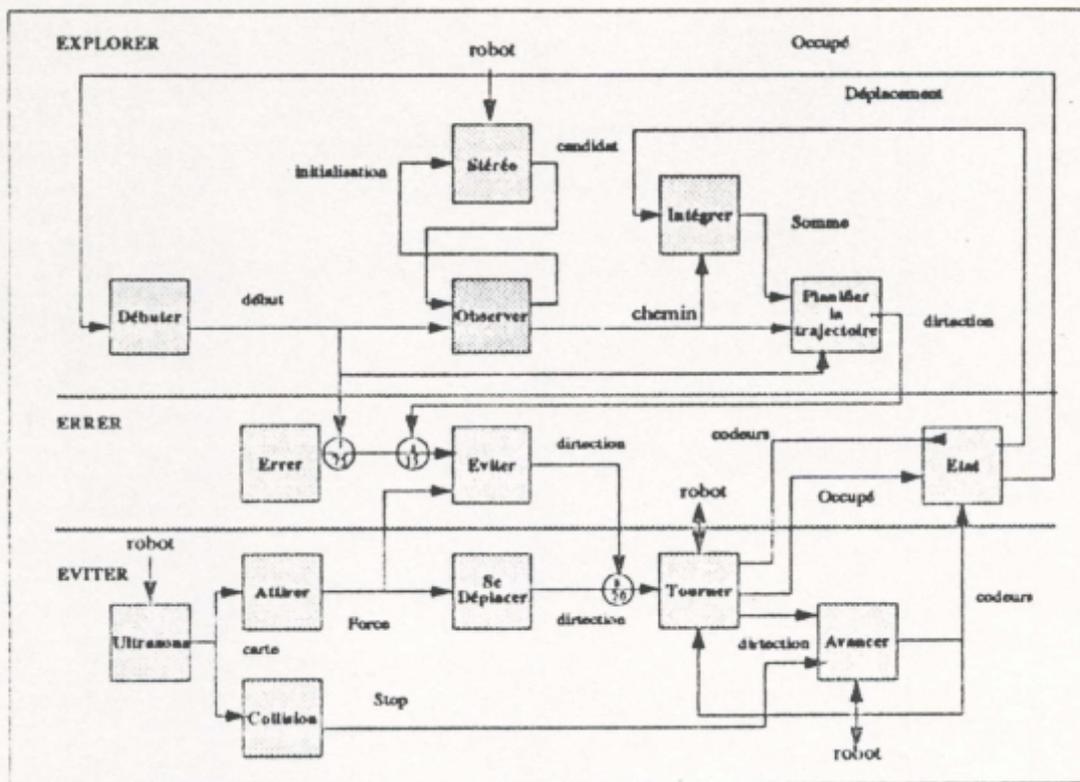


FIGURE I-1 (Prise de [PER 92])

Les trois premiers niveaux de la structure de BROOKS; chaque niveau implante un comportement unique (explorer, erreur, éviter). BROOKS n'a pas encore implémenté les niveaux supérieurs de la structure qu'il propose.

la connaissance globale du système. Cette connaissance est utilisée pour synchroniser l'activité des entités qui coopèrent à la résolution du problème posé; parmi ces architectures, nous citons les architectures issues des techniques de l'intelligence artificielle du type tableau noir /tableau blanc, où le modèle du tableau noir propose un schéma pour organiser les domaines de connaissances et les étapes du raisonnement conduisant à la construction incrémentale de la solution au problème posé, ce modèle représenté à la **figure I-2** est composé de trois éléments majeurs. (pour plus de détails sur les architectures de type tableau noir voir [**NII 86a**] et [**NII 86b**]):

* les sources de connaissances qui sont indépendantes et se repartissent toute la connaissance opératoire pour communiquer et se synchroniser, elles utilisent exclusivement la base de données centrale.

* La base de données centrale ou tableau noir, mémorise, à chaque instant l'état courant de la résolution du problème. Les éléments de solution sont mis à jour par les sources de connaissances.

* le système de contrôle qui gère l'activité des sources de connaissances.

ELFES propose en 1983 [**ELF 83**], une structure décisionnelle basée sur le concept du tableau noir pour le contrôle du robot CMU ROVER **figure I-3**. Un ensemble de modules experts communiquent par échange de messages via un tableau noir pour exécuter une mission sous la direction d'un plan de contrôle. Ces modules chacun conçu pour accomplir une tâche particulière (ex. commande d'un capteur, d'un effecteur, etc....) se composent de deux processus indépendantes:

- Un esclave qui effectue les traitements
- Un maître responsable de l'activité de l'esclave et

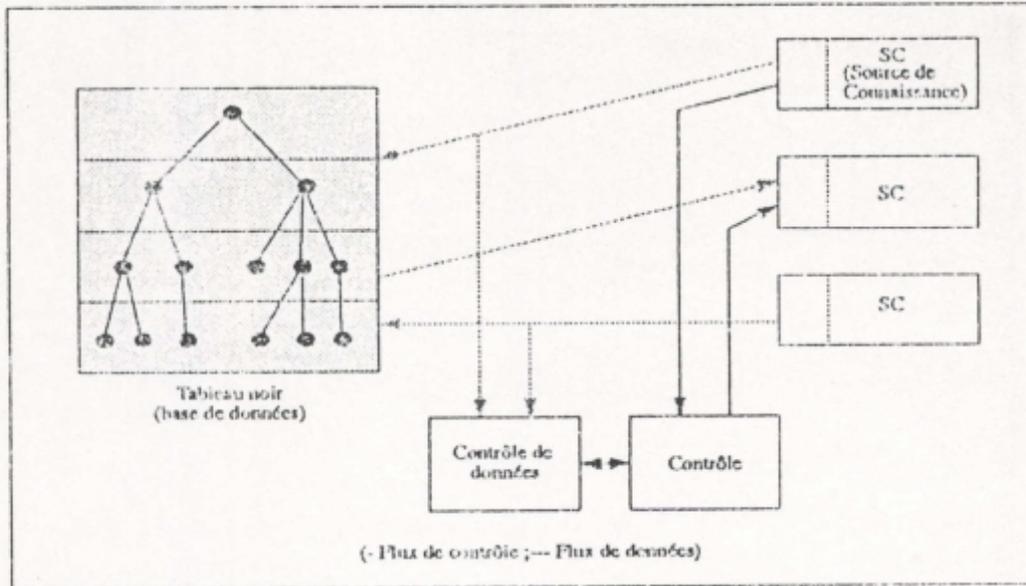


FIGURE I-2 (Prise de [PER 92])

Le modèle du tableau noir, composé de trois éléments majeurs: les sources de connaissances, la base de données centrale (tableau noir), et le système de contrôle.

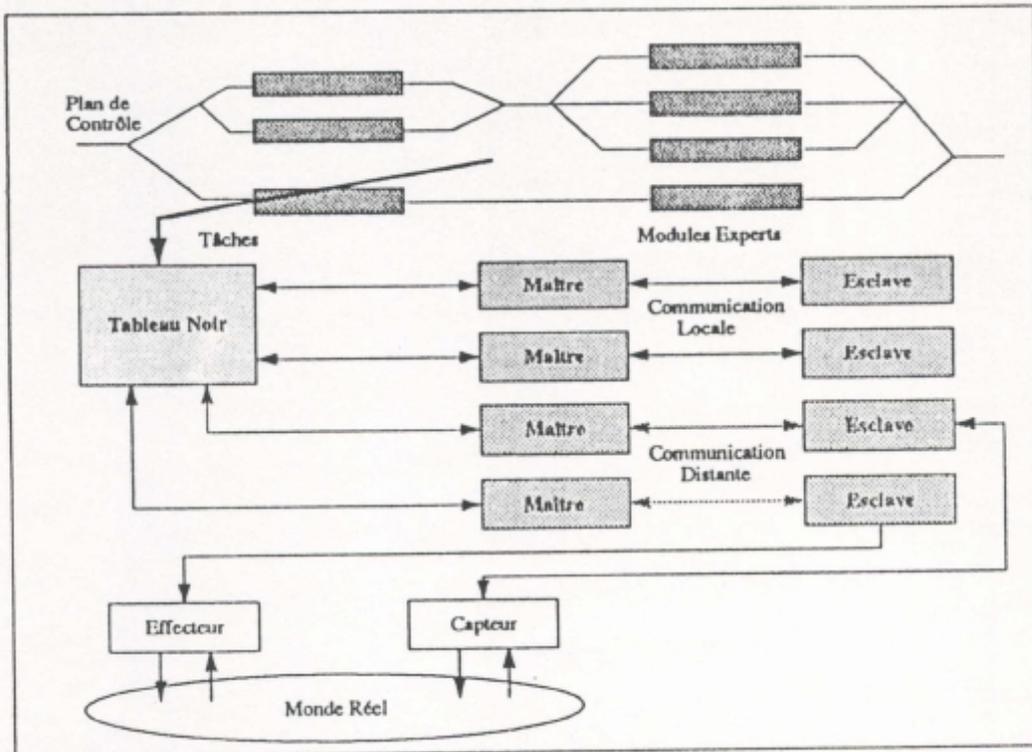


FIGURE I-3 (Prise de [PER 92])

Structure de contrôle du robot C.M.U ROVER; cette structure est du type tableau noir.

des communications avec le tableau noir, Voir **figure I-3**.

L'équipe de recherche du groupe RIA (Robotique et Intelligence Artificielle) du LAAS (Laboratoire d'Automatique et d'Analyse des Systèmes à TOULOUSE) sous la direction de GEORGE GIRALT développèrent en 1983 [**GIR 83**], une structure de contrôle pour le robot HILARE 1, composée d'un ensemble de modules spécialisés, d'un planificateur de trajectoire et d'un contrôleur d'exécution. Chaque module réalise une tâche particulière et interagit avec les autres par échange de messages voir, **figure I-4**.

Dans le cadre du projet ALV (Autonomous Land Vehicule) l'équipe de C.THORPE développa en 1986 [**THO 86**], un prototype "NAVLAB" capable d'effectuer le suivi de route en utilisant deux caméras vidéo couleurs et une caméra LASER 3D. L'architecture s'articule autour d'une base de données, nommée CODGER (COmmunication Database with GEometric Reasoning), à laquelle accèdent cinq modules (voir **figure I-5**).

- * le capitaine contrôle l'exécution d'une mission de suivi de route. Pour cela il envoie la destination et les contraintes de chaque étape au navigateur, et traite les résultats d'exécution avant d'enclencher l'étape suivante;

- * le navigateur consulte la base de données pour choisir la meilleure route, puis la décompose en segments avant de l'envoyer au pilote;

- * le pilote coordonne les activités entre la perception et la locomotion pour générer des déplacements continus;

- * la perception gère le fonctionnement des capteurs pour prendre des images de la route, détecter les objets et localiser le robot;

- * la locomotion assure les déplacements du véhicule en utilisant les trajectoires planifiées par le pilote.

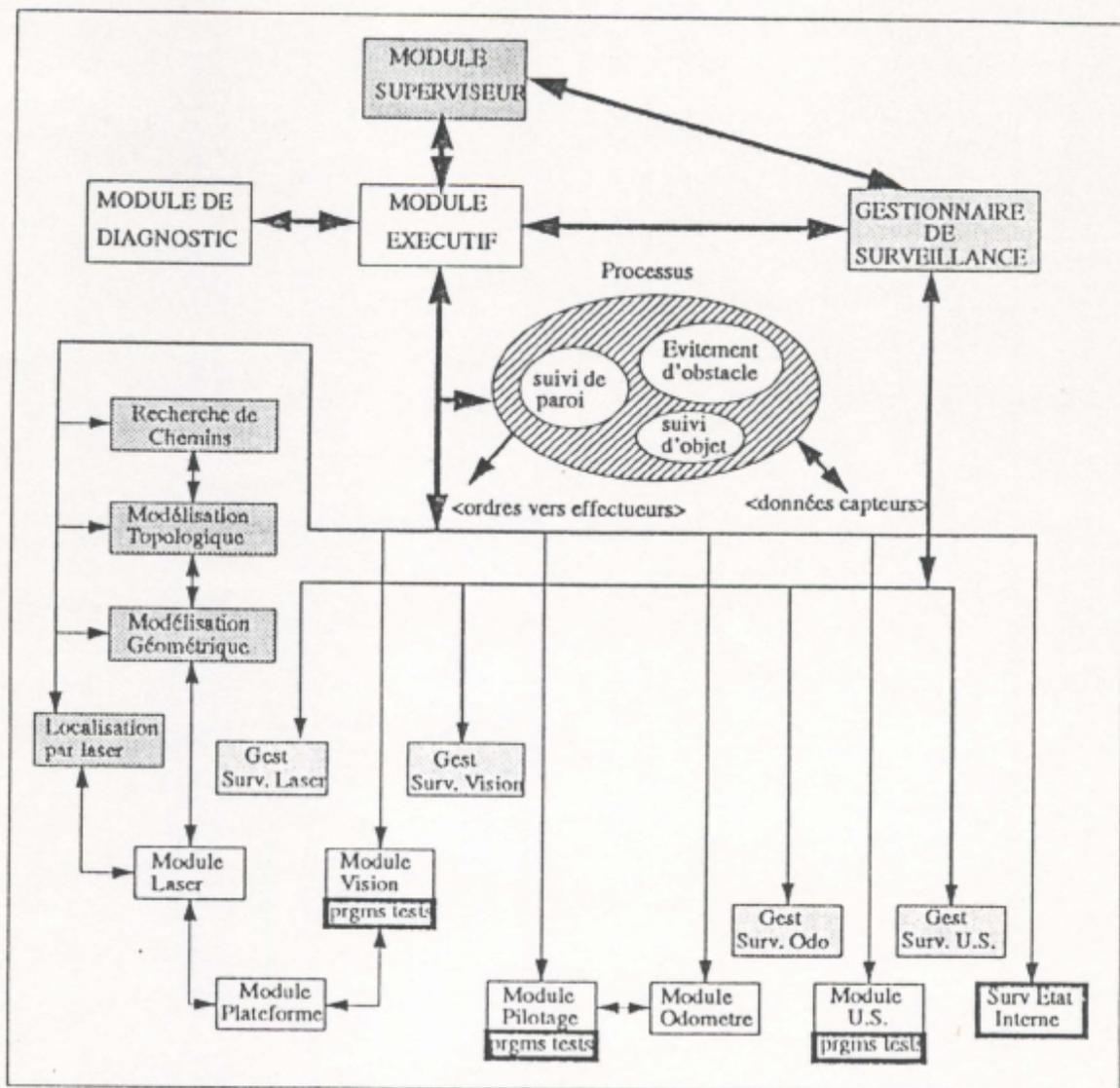


FIGURE I-4 (Prise de [PER 92])

Structure de contrôle du robot HILARE I constituée de trois niveaux:

- * Le niveau fonctionnel (niveau le plus bas)
- * Le système de contrôle (niveau le plus haut)
- * Le niveau de planification ou l'opérateur humain.

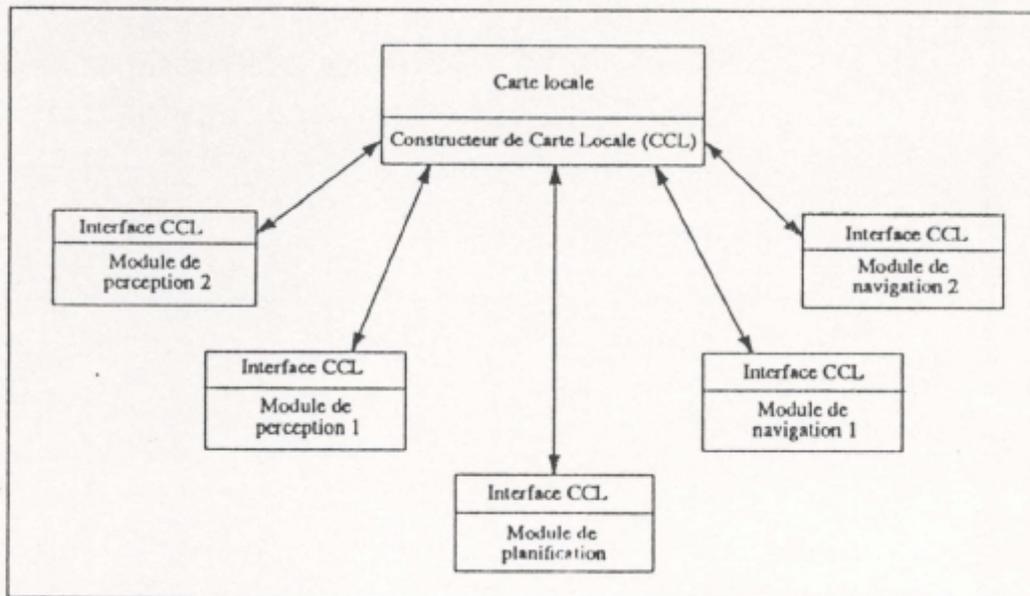


FIGURE I-5 (Prise de [STE 86])

Structure de contrôle du robot NAVLAB; elle s'articule autour d'une base de données nommée CODGER (COMMUNICATION DATABASE WITH GEOMETRIC REASONING). Elle est du type tableau blanc.

Le modèle CODGER est également appelée tableau blanc pour se démarquer du modèle tableau noir. Cette architecture permet une décomposition hétérarchique du système et un accès asynchrone à la base de données pour y déposer ou en retirer des informations.

L'architecture NASREM (NAa/nsb Standard Reference Model for telerobot control system), parue en 1987 dans [ALB 87], est une architecture en couche, conçu pour produire à la fois la structure de contrôle d'un robot, et les bancs de test des algorithmes (**figure I-6**). Elle permet une décomposition h i é r a r c h i q u e d u système et une centralisation de la connaissance, des données etc , à l'intérieur d'une structure unique pouvant être un tableau noir. La hiérarchisation du système repose sur deux niveaux de décomposition.

* Une décomposition verticale qui fait apparaître trois colonnes respectivement dédiées aux fonctions: de perception, de modélisation, et de planification et d'exécution.

* une décomposition horizontale qui définit six couches représentant les différents niveaux de conception d'un algorithme de contrôle.

Les trois les plus basses dépendent du robot qui exécute les actions et les trois autres dépendent uniquement des tâches à effectuer. Pour pouvoir observer et superviser le comportement du robot, une interface homme-machine a été conçue à cet effet, ayant un accès à tous les niveaux de la hiérarchie.

Une autre architecture en couche, décrite en 1991 dans [CHO 91], a été développée dans le cadre du projet RAMINA, destinée à un robot mobile devant évoluer à l'intérieur de bâtiments pour y effectuer des missions spécifiées de manière symbolique; c'est une architecture dite orientée objet (**figure I-7**).

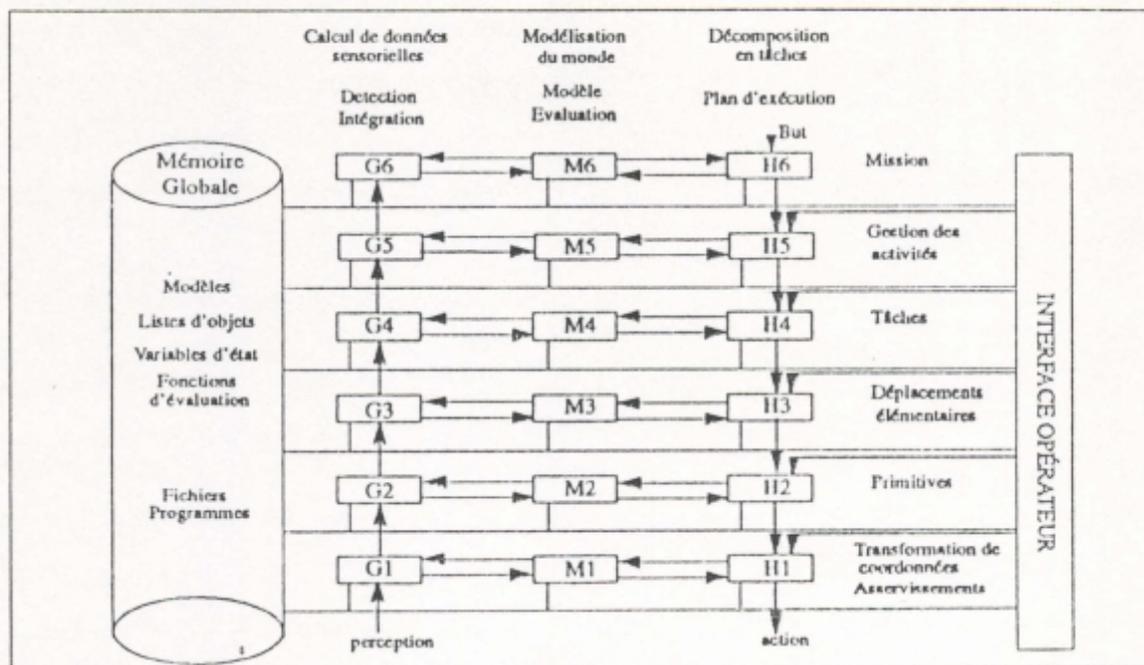


FIGURE I-6 (Prise de [PER 92])

Le modèle de l'architecture NASREM (Nae/nsb Standard Reference Model) est conçu pour produire la structure de contrôle du robot et les bancs de test des algorithmes.

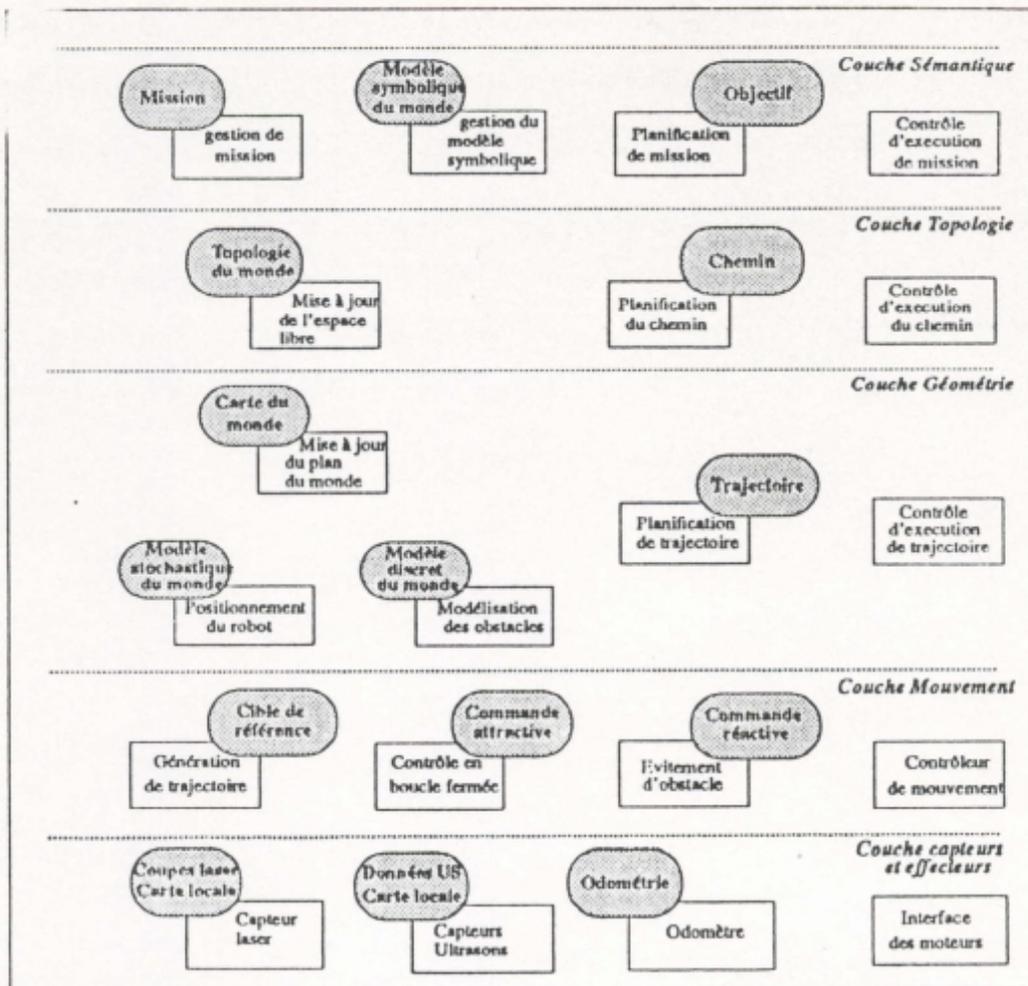


FIGURE I-7 (Prise de [PER 92])

La structure de contrôle proposée pour RAMINA; c'est une structure en couche destinée à un robot mobile devant évoluer à l'intérieur des bâtiments.

Une architecture parallèle, parue en 1991 dans [CRO 91], pour le contrôle du robot mobile de surveillance, proposée par l'équipe de JAMES.L.CROWLEY au sein du groupe LIFIA (Laboratoire d'Informatique et d'Intelligence Artificielle) (**figure I-8**). Dans cette architecture la locomotion et la perception présentent parallèlement une organisation hiérarchique de niveaux:

* Le plus bas niveau, c'est le niveau signal, il englobe la partie contrôle des actionneurs pour maintenir une vitesse spécifiée des moteurs, et traitement des signaux issus des capteurs pour estimer la position du robot.

* Le niveau immédiatement supérieur c'est le niveau effecteur; A ce niveau la partie locomotion contrôle les déplacements du véhicule alors que la partie perception s'occupe de la mise à jour du modèle locale à partir des données des capteurs.

* Le troisième niveau c'est le niveau action. Une action est une séquence coordonnée des commandes de l'effecteur; à ce niveau se fait la planification et l'exécution des tâches recommandées par le superviseur.

* Le superviseur se trouve au quatrième niveau, lié directement à l'interface homme-machine. C'est une forme d'une servoe loop symbolique composée de 3 phases:

- 1) sélection de l'action pour ramener le robot à l'état désiré.
- 2) exécution de l'action, et
- 3) évaluation des conséquences de cette action.

Le superviseur est le contrôleur des deux hiérarchies: locomotion et perception.

La **figure I-9** représente le diagramme bloc du niveau le plus bas de l'architecture du robot mobile BLANCHE développée par COX en 1991 [COX 91].

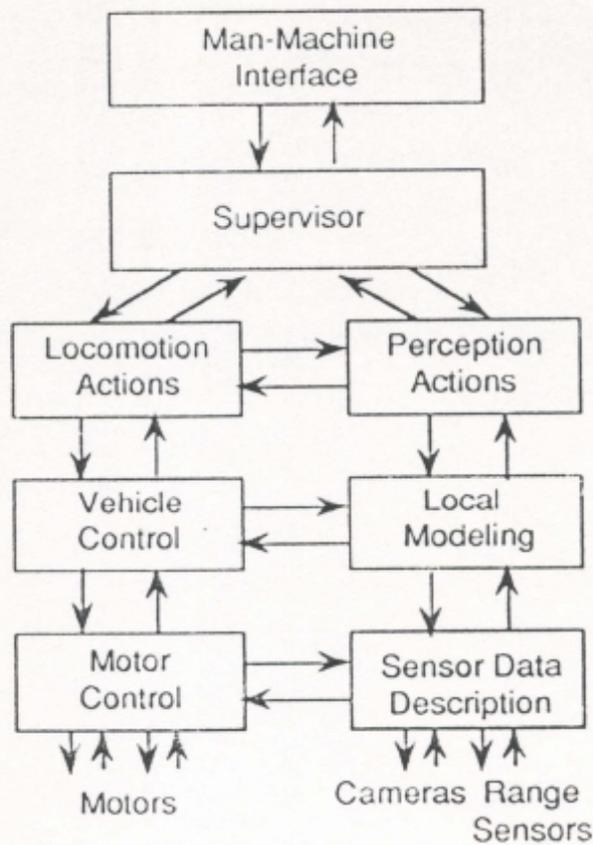


FIGURE I-8 (Prise de [CRO 91])

Architecture parallèle pour le contrôle d'un robot de surveillance proposée par JAMES.L.CROWLEY & All. La locomotion et la perception sont prises parallèlement en une hiérarchie de niveaux.

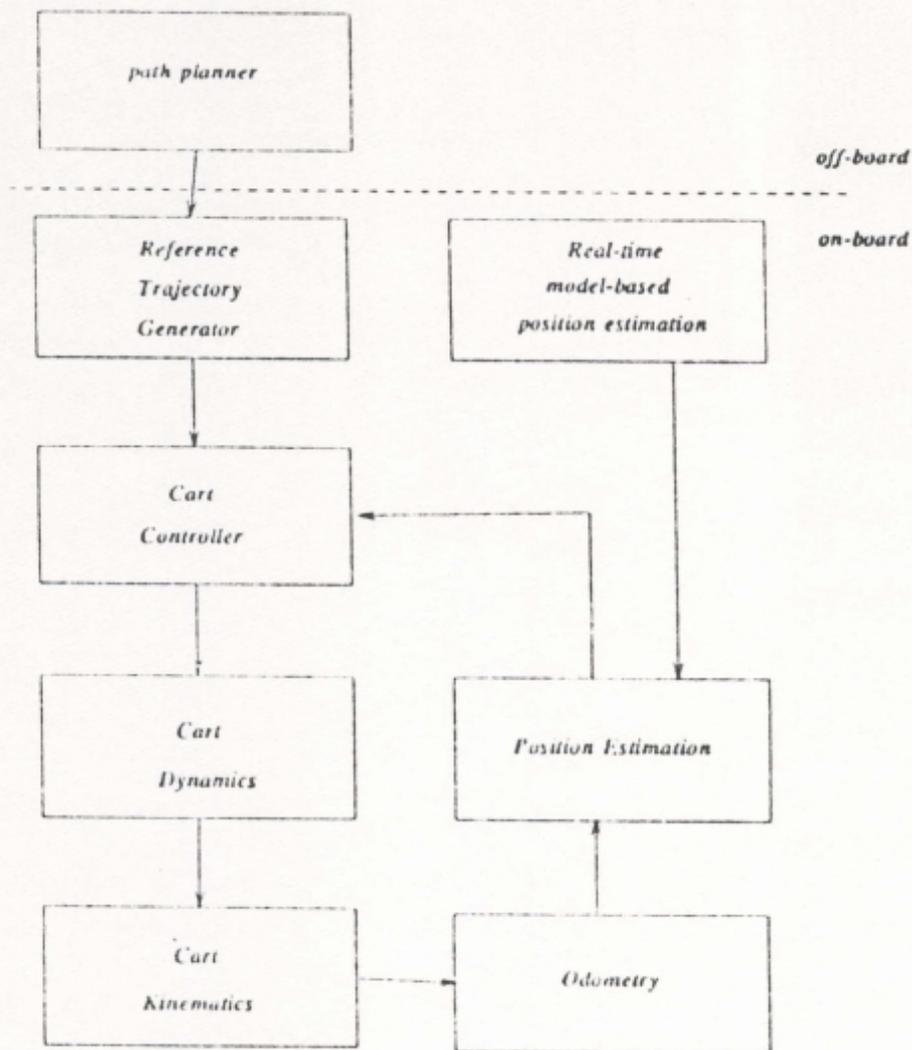


FIGURE I-9 (Prise de [COX 91])

Le diagramme bloc du niveau le plus bas de la structure de contrôle implémentée sur BLANCHE.

Une autre architecture, décrite en 1992 dans [PER 92], proche de celle de HILARE 1 a été développée au sein du même groupe RIA de LAAS, pour le contrôle du robot HILARE 2 (figure I-10), les trois niveaux ont été gardés mais chacun d'eux est composé et organisé différemment.

4 CONCLUSION

Les architectures présentées dans les approches purement réactives, mettent toutes en avant un fonctionnement du robot uniquement basé sur un ensemble de boucles perception/action; Par contre les dénominateurs communs des approches utilisant une fonction de planification sont:

- * La centralisation de la connaissance nécessaire à l'élaboration des missions, la gestion des ressources, la résolution des conflits etc....

- * La centralisation des fonctions de contrôle destinées à gérer les actions élémentaires à mettre en oeuvre pour effectuer les tâches planifiées.

La composante de planification est donc indispensable dans la structure d'un robot mobile, à associer à la réactivité.

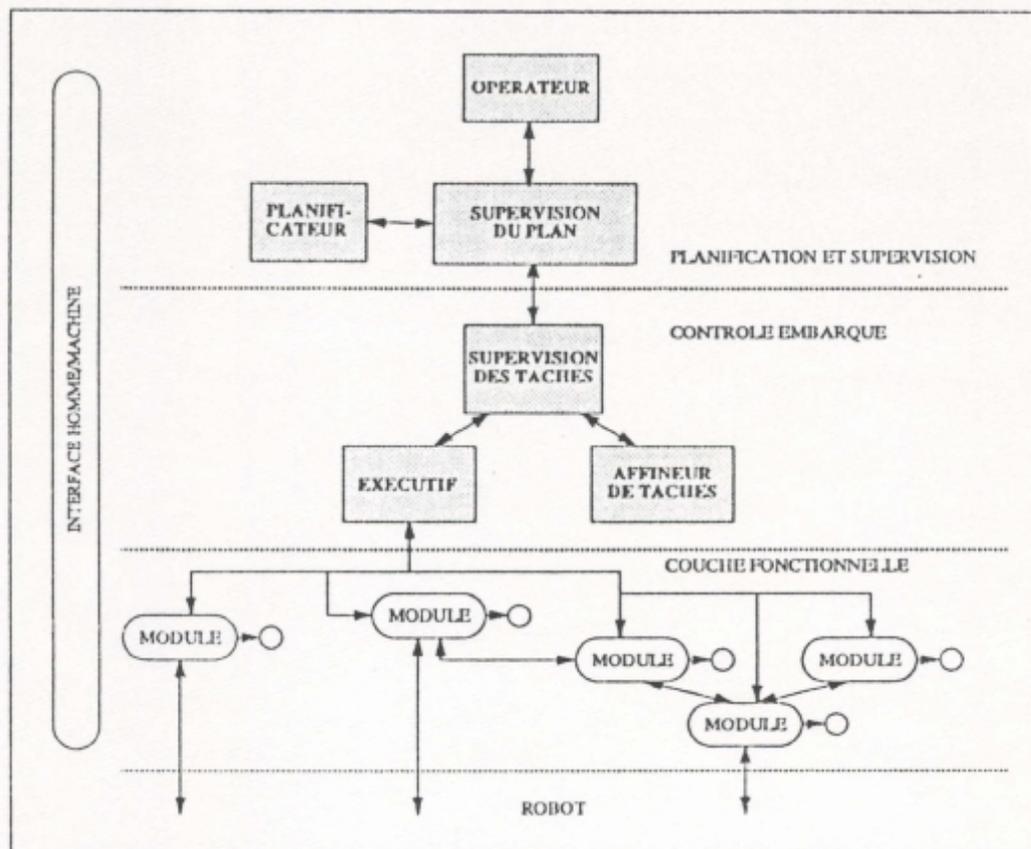


FIGURE I-10 (Prise de [PER 92])

Structure de contrôle du robot HILARE II, les trois niveaux de la structure de contrôle de HILARE I ont été maintenus et disposés autrement.

CHAPITRE II
NAVIGATION ET GUIDAGE DES ROBOTS
MOBILES AUTONOMES

1 INTRODUCTION

Le robot mobile autonome doit atteindre son objectif en sécurité malgré le manque d'information sur son environnement; A cet effet plusieurs approches ont été développées dans le but d'augmenter la capacité de décision de ces robots. Ces approches s'étendent des techniques les plus primitives où la connaissance précise de l'environnement est impérative, aux techniques les plus complexes où la connaissance de l'environnement est imprécise et le temps d'apparition des obstacles est aléatoire. Ce chapitre résume quelques approches très récentes rentrant dans le cadre de la navigation et l'évitement de collision des robots mobiles autonomes dans des environnements partiellement connus ou complètement inconnus

2 NAVIGATION SUR DES TERRAINS PARTIELLEMENT CONNUS

2 - 1 Introduction

La connaissance même partielle de la position des obstacles fixes, nécessite l'emploi des techniques de planification de trajectoires devant être prises par le robot pour atteindre un but préalablement fixé par un opérateur; ces planificateurs ont besoin d'un mécanisme de modélisation de l'environnement, et d'un système de perception destiné à la localisation du robot et l'évitement d'éventuelles collisions.

Généralement la planification et l'exécution des trajectoires sont hiérarchisées en trois niveaux de bases:

1- Le planificateur (planificateur global de trajectoires): utilise les connaissances à priori pour planifier une trajectoire globale.

2- Le navigateur (planificateur local de trajectoires): utilise le plan du planificateur global en prévoyant des routines beaucoup plus précises en accord avec des informations obtenues localement sur le terrain.

3- Le pilote: exécution des routines simples du mouvement du véhicule.

Cette section propose une classification de quelques résumés de différentes approches récentes traitant le niveau "navigateur", par ordre de leurs apparitions.

2 - 2 L'approche développée par KEISY (d'après [KEI 84])

Afin de créer une chaîne d'actions appropriés; deux algorithmes similaires ont été développés simultanément à l'université de FLORIDA et à HUGHES RESEARCH LABORATORIES, le premier est appelé W.S.A (Wandering Standpoint Algorithm) paru sous deux versions: W.S.A discrète et W.S.A continue. Le deuxième algorithme appelé FEASIBLE-PATH. Le processus de génération de trajectoires peut être défini comme suit:

* La procédure W.S.A discrète.

1) Si la ligne directe à l'objectif n'est pas interrompue par un obstacle, exécuter la trajectoire et terminer avec succès.

2) Déterminer les candidats probables pour un PWE(Passage Way Entrances) sur les deux cotés de la ligne directe et les arranger dans l'ordre croissant de préféralité, si aucun n'existe, terminer avec échec.

3) Sélectionner le PWE avec le maximum de préféralité, rendre ce point le prochain point de stop SP (Stop Point).

4) Si ce point existe déjà sur la liste des solutions, prendre le prochain point candidat comme prochain SP.

5) Exécuter le chemin dans la direction du prochain SP et mettre le SP courant dans la liste des solutions.

6) Aller à 1.

Le cas continu du WSA s'obtient en appliquant le cas discret à chaque Δt le long de la trajectoire reliant la position courante au prochain SP. Δt est un intervalle de temps compatible avec la durée de traitement pour chaque

increment de trajectoire.

*** La procédure FEASIBLE-PATH**

1) initialisation: A=IP(point intermédiaire), B=BUT. PATH-NODE-LIST(liste des noeuds du chemin)=NUL, PRED(IP)(ascendant de IP)=NUL

2) si le segment AB n'est pas interrompu par un obstacle (B est visible à partir de A), exécuter le chemin directement à B.

3) CANDIDATE-LIST(liste des noeuds candidats)={noeuds N: AN est un chemin ininterrompu; N est un point tangentiel local par rapport à A}.

* N est un point tangentiel local par rapport à A, si N est noeud de l'obstacle tel qu'au voisinage de ce noeud l'obstacle se trouve entièrement d'un seul coté de la ligne AN (**FIGURE II-1**).

* les noeuds de la liste "CANDIDATE-LIST" sont probablement candidats pour un PWE.

4) Si "CANDIDATE-LIST"=NUL, alors, "NEXT-NODE" (prochain noeud)=PRED(A); sinon "NEXT-NODE"=le noeud le plus préférable dans "CANDIDATE-LIST"; et PRED(NEXT-NODE)=A

5) si "NEXT-NODE"=NUL, terminer l'exécution avec échec (pas de chemin possible au but); sinon "PATH-NODE-LIST"=A, exécuter le chemin de A à "NEXT-NODE", et A=NEXT-NODE, aller à l'étape 2.

Dans la première étape du WSA les bordures de chaque objet sont étendues d'une valeur qui dépend des dimensions physiques du robot. Le robot étant rendu ponctuel, WSA teste les intersections du système mobile avec l'ensemble des obstacles; parmi les points d'intersections de la ligne de vision reliant le SP courant au but, avec les cotés de l'obstacle, seul, le plus proche au SP courant est retenue. Chaque sommet de l'obstacle associé au point d'intersection le plus proche est utilisé pour générer un angle de déviation à partir de la ligne directe à l'objectif. Le sommet qui donne l'angle de déviation le plus élevé est appelé E1 et celui qui donne l'angle de déviation le moins élevé est

appelé E2, les angles de déviation correspondants et la distance à ces sommets sont appelés; E1.dev, E2.dev, E1.dis, et E2.dis; ces valeurs sont utilisées pour générer le prochain SP en s'appuyant sur les contraintes d'optimisation suivantes:

Soit à minimiser:

I - $E_i.dis + d_i$ (d_i est la distance du sommet E_i à l'objectif)

II - $E_i.dev$

III - $(d_i + E_i.dis)^2 / E_i.dis$

La **FIGURE II-2** montre une trajectoire obtenue en utilisant l'algorithme WSA et les contraintes d'optimisation I, II et III.

2 - 3 L'approche visuelle de SUZUKI et ARIMOTO (d'après [SUZ 88])

H .SUZUKI et S.ARIMOTO proposèrent en 1988 dans [SUZ 88] une approche visuelle pour le contrôle de déplacement d'un robot mobile autonome équipé d'une caméra C.C.D N/B (**FIGURE II-3**); L'algorithme "P" développé dans cette approche permet une auto-organisation du modèle de l'environnement:

A chaque signal d'entrée x est liée une information y . Durant l'étape localisation et pendant l'apprentissage, la vue panoramique prise par la caméra sur le plan horizontal est prise comme l'élément x (**FIGURE II-4**), l'information y liée à cette vue, correspond au numéro du bloc où se trouve le robot (**FIGURE II-5**); Pour se localiser pendant l'exécution d'une tâche, le robot cherche dans sa base de connaissance la position liée à l'image x similaire au paysage x^- où il se trouve (le critère de décision est basé sur la minimisation d'une certaine métrique $d(x, x^-)$ définie comme étant la distance entre l'image mémorisée x et l'image vue réellement x^-).

L'algorithme "P" a été également appliqué à l'évitement de collisions; Dans cette application la caméra scrute la

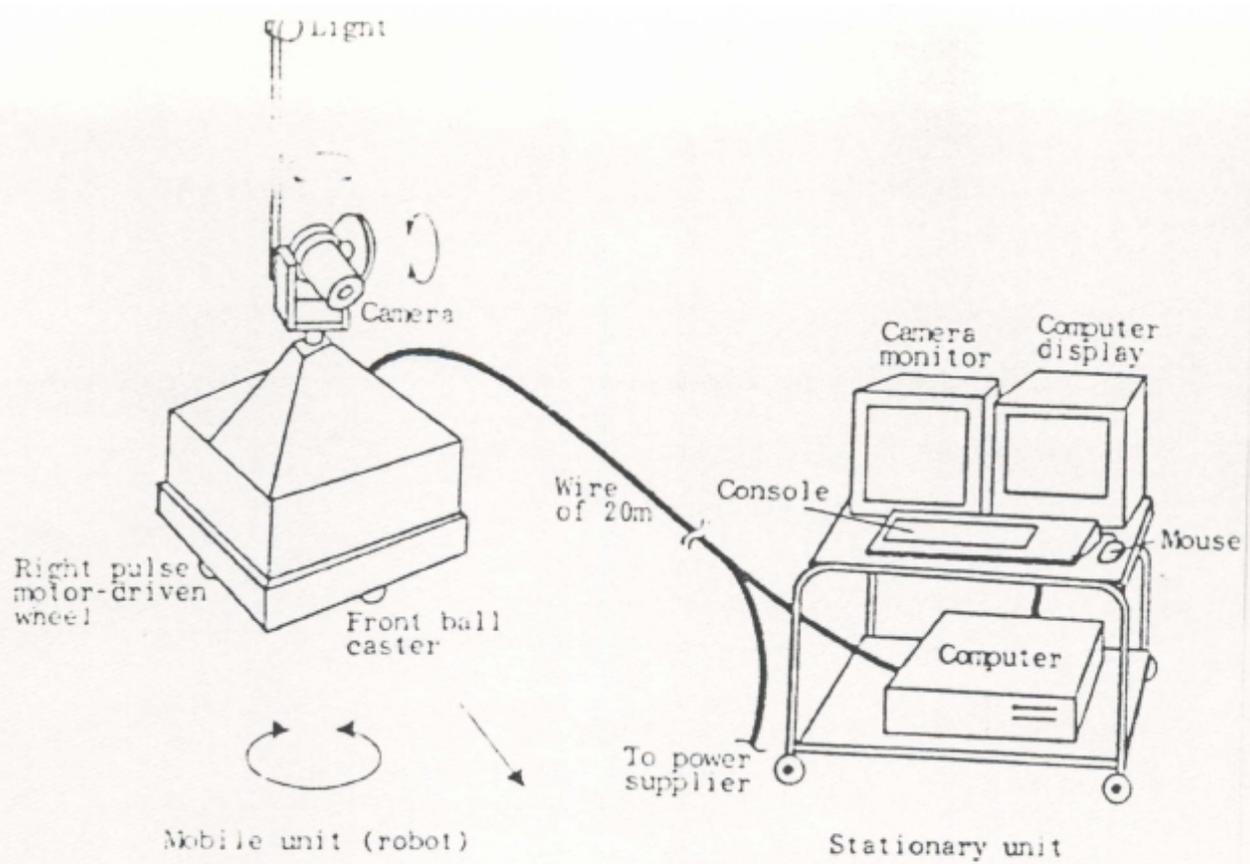


FIGURE II-3 (Prise de [SUZ 88])

Configuration du système robot mobile autonome.

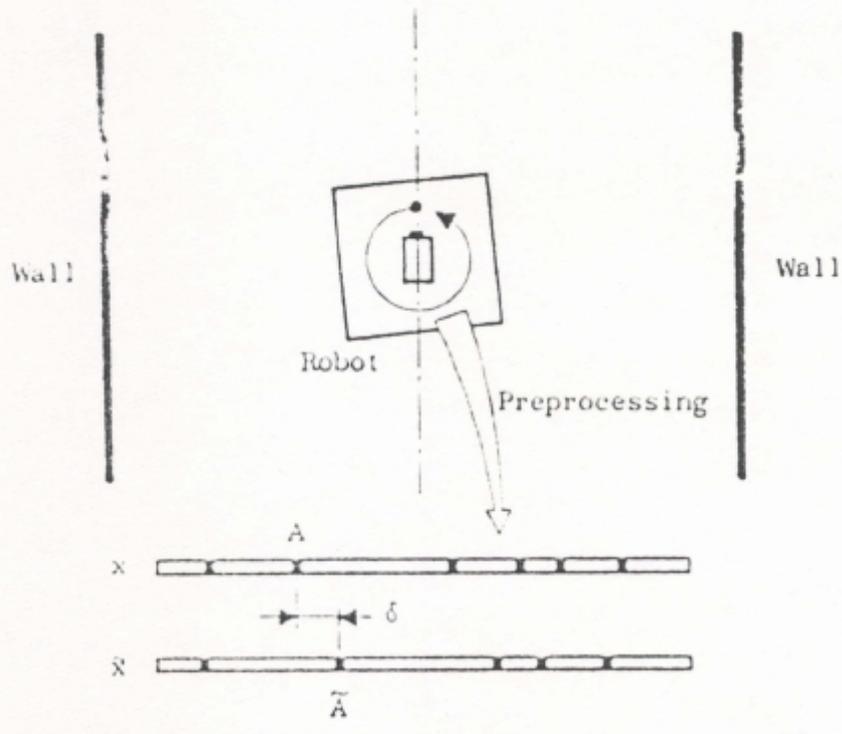


FIGURE II-4 (Prise de [SUZ 88])

Identification de la position.

face avant du haut en bas, l'image obtenue (**FIGURE II-6**) est prise comme l'élément x, l'information y liée à cette image est la trajectoire que doit suivre le robot dans l'espace libre; Comme dans l'étape localisation, durant l'exécution d'une tâche le robot cherche dans sa base de connaissance la trajectoire à suivre liée à l'image x similaire à l'image courante x^{\sim} .

2 - 4 L'approche développée par BORENSTEIN et KOREN (d'après [BOR 88])

Dans cette approche **J.BORENSTEIN** et **Y.KOREN** ont essayé de mettre en avant les performances des capteurs ultrasonores dans le guidage et la navigation des robots mobiles.

En effet, le robot infirmier de **J.BORENSTEIN** et **Y.KOREN** (**FIGURE II-7**) est capable d'atteindre son objectif dans une chambre sans collision; Une cartographie de l'environnement avec des obstacles statiques est enregistrée dans la base de donnée du robot avant toute tâche; Les obstacles imprévus doivent être détectés par le robot. Deux types de capteurs sont utilisés à cet effet:

- les capteurs tactiles (microswitchs) placés sur le pare-choc du robot.
- deux capteurs à ultrasons placés sur la partie la plus haute du robot. Ces derniers fonctionnent en deux modes distincts, mode scrutation et mode mesure.

Lorsque le robot se déplace en avant; les capteurs à ultrasons fonctionnent en mode scrutation, et les informations sont prélevées alternativement des deux capteurs à chaque 40 ms (soit un déplacement maximal de quelques centimètres); Une alerte à l'obstacle se déclenche lorsque le test suivant est "vrai":

```
IF  $R_i(j) < TD$  AND  $R_i(j) < = R_i(j-1)$  THEN ALARM
```

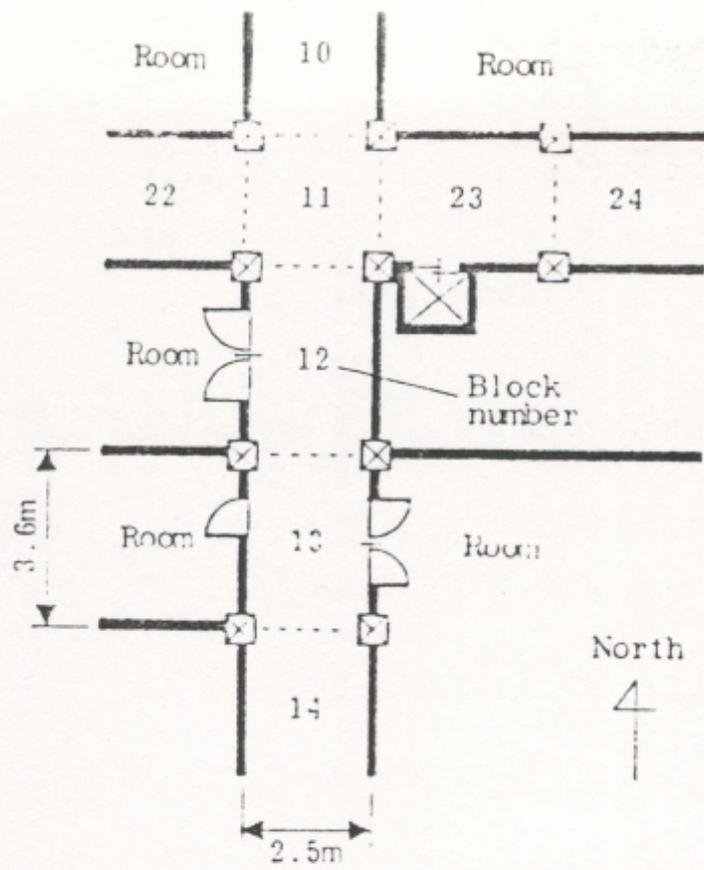


FIGURE II-5 (Prise de [SUZ 88])

Environnement expérimental.

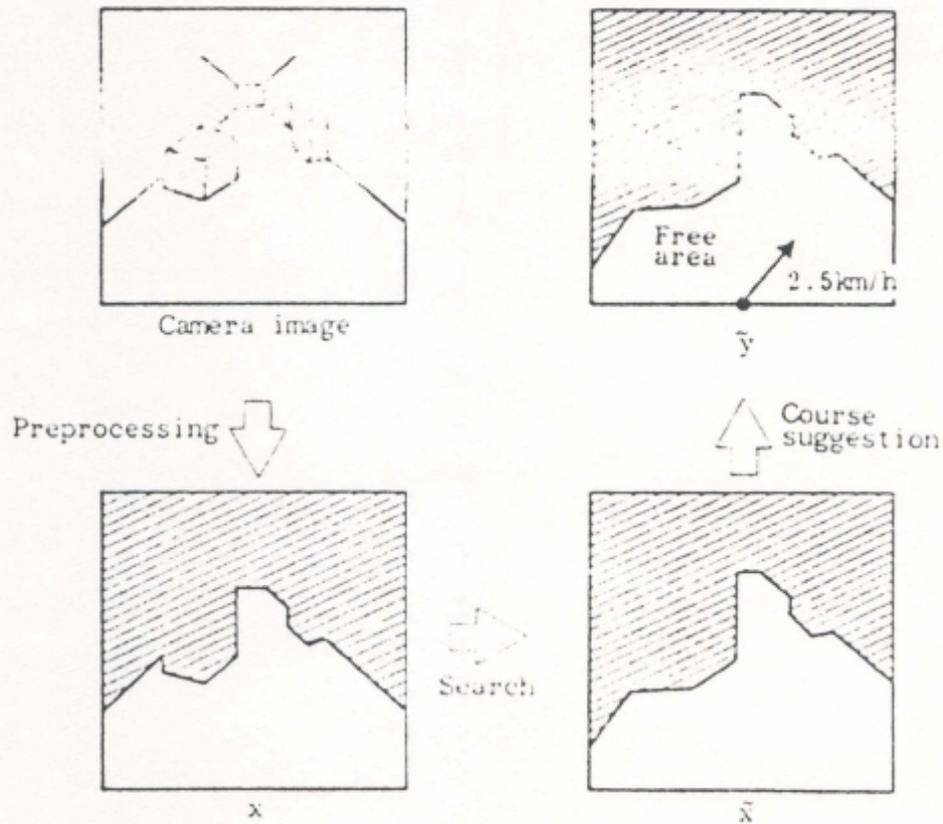
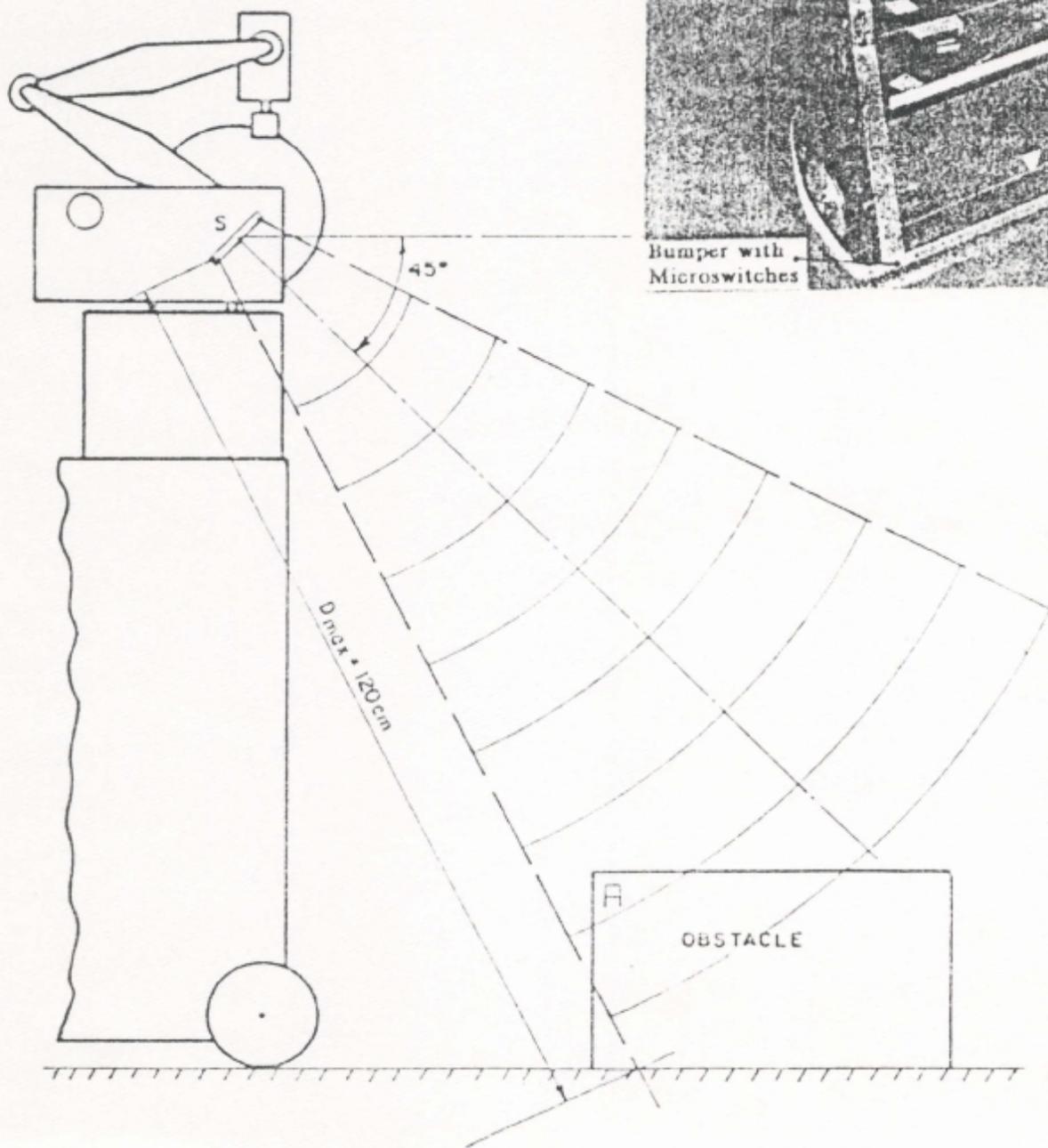


FIGURE II-6 (Prise de [SUZ 88])

Mouvement d'évitement d'obstacle.



Ultrasonic range finders

Bumper with Microswitches

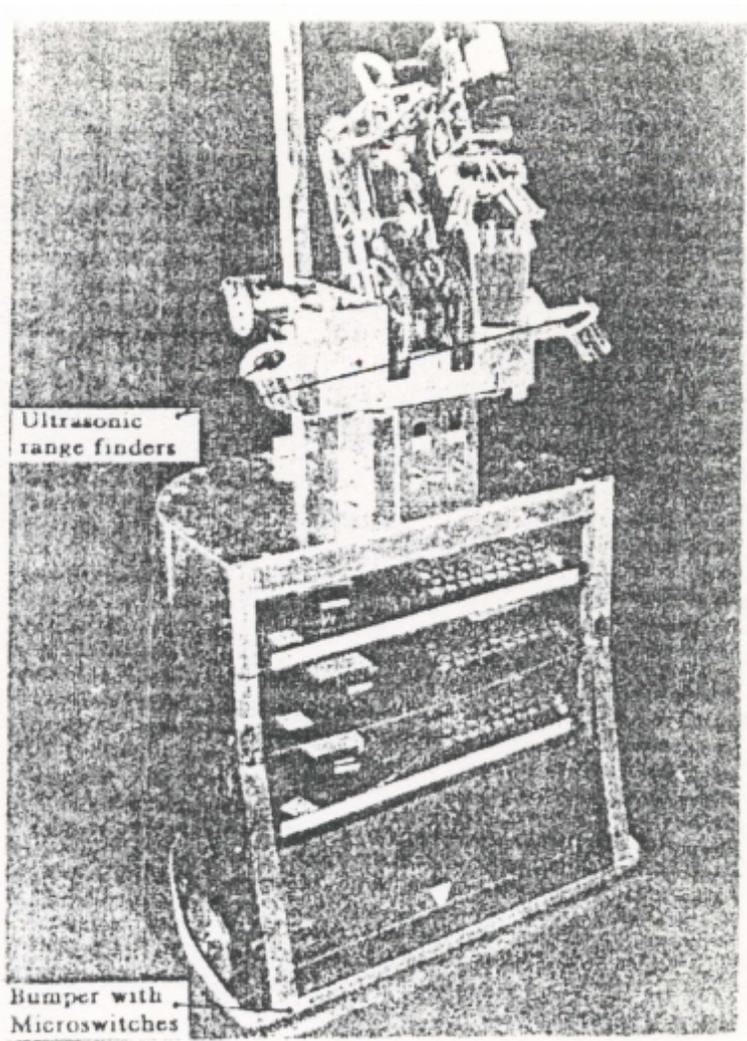


FIGURE II-7 (Prise de [BOR 88])

Le robot infirmier.

TD: seuil

$R_i(j)$: lecture de la distance prélevée de la ième sonde

$R_i(j-1)$: lecture précédente prélevée de la ième sonde

Pour répondre à une alerte, le robot s'arrête et les capteurs à ultrasons fonctionnent en mode mesure, dans ce mode le robot fait tourner le support des sondes de 70° à gauche et 70° à droite, et fait des prélèvements de la distance à chaque 2° ; Si un prélèvement dépasse subitement le seuil TD ($TD=100$ cm) un sommet de l'obstacle est alors détecté; et le robot ne garde en mémoire que ce sommet; Si aucun sommet n'est détecté, ni à gauche ni à droite (cas où le robot est en face d'un mur) un sommet est pris à l'extrême gauche ou à l'extrême droite.

La **FIGURE II-8** illustre le résultat expérimental d'une scrutation à gauche où le robot est partiellement en face d'un mur; Le transducteur S1 fait une rotation de 70° par rapport au centre C, et durant la rotation des prélèvements de la distance sont effectués à chaque pas de 3° . Les points 1 à 12 sont le résultat de cette scrutation; Les points 1 à 9 proviennent du mur, tandis que les points 10 à 12 proviennent du sol ($D=120$ cm); A partir du point 9 on a un prélèvement qui dépasse le seuil TD, le point 9 est donc pris comme sommet gauche de l'obstacle et le seul qui est retenu en mémoire, et de la même façon avec le transducteur S2 le sommet droit est détecté et mémorisé. La ligne joignant ces deux sommets constitue une bordure de l'obstacle.

L'évitement des obstacles est illustré à travers l'exemple suivant:

La **FIGURE II-9** montre le modèle de l'environnement acquis a priori par le robot; le robot est ordonné d'aller du point O au point F (**FIGURE II-10**), une chaise imprévue par le modèle est placée dans sa trajectoire, après avoir reçu la commande le robot tourne par rapport à son centre pour atteindre la direction de F, il suit la ligne droite reliant O à F, à ce moment le robot ignore la présence de la chaise;

12 11 10

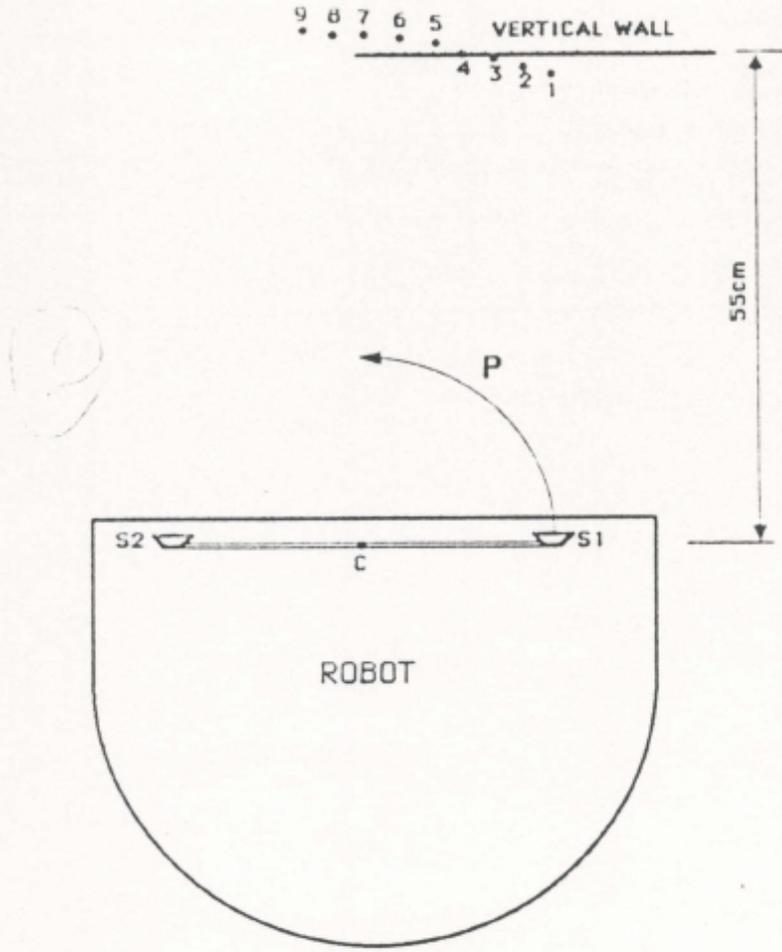


FIGURE II-8 (Prise de [BOR 88])

Scrutation à gauche où le robot est en face d'un obstacle vertical.

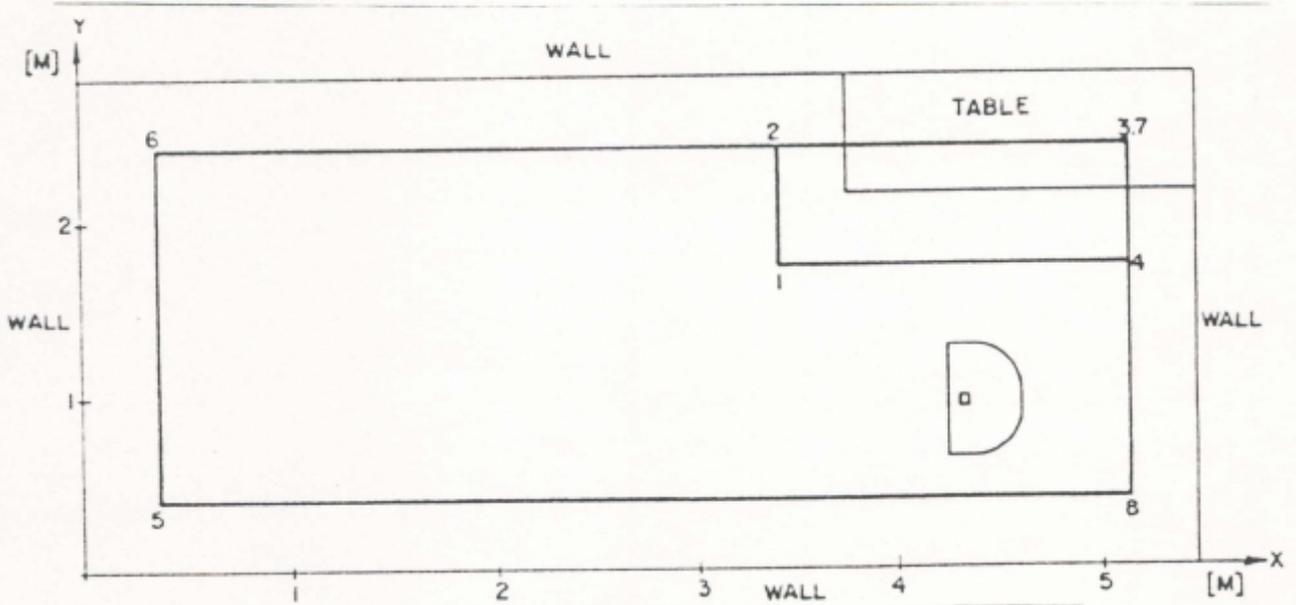


FIGURE II-9 (Prise de [BOR 88])

La cartographie des obstacles statiques.

arrivant en A les capteurs ultrasonores détectent la présence de l'obstacle, le robot s'arrête et le mode mesure est activé pour détecter les deux sommets (9-10) ces deux points sont reliés ensemble pour donner une bordure de l'obstacle; le robot à ce moment fait appel au planificateur, celui ci génère la trajectoire optimale, il propose le point 10 pour atteindre F; en essayant d'aller à F à partir du point 10, le robot détecte encore une fois la présence de l'obstacle en B, il essaye de nouveau de détecter les deux sommets (11-12), ici le planificateur propose le point 12 pour aller à F; en suivant cette trajectoire le robot arrive à F en sécurité.

Après avoir l'arrivée à destination, le robot est ordonné à revenir au point O, l'historique de se voyage est représenté à la **FIGURE II-11**.

Les bordures 9-10 et 11-12 sont maintenant connu par le planificateur, il suggère le point 9 pour atteindre le point O; le robot s'oriente vers le point 9 et il commence à se déplacer dans sa direction, avant d'atteindre le point 9, les capteurs tactiles signalent la présence d'un obstacle au robot, il s'agit d'une pied de la chaise indétectable par les capteurs ultrasonores; Le robot s'arrête, fait une marche arrière de 30cm, tourne de 30° à droite, il avance de 50cm, et il s'arrête, ce point d'arrêt est le point 14; a ce moment le robot fait appel au planificateur, celui-ci propose le point 9 pour atteindre O, en suivant cette trajectoire le robot atteint le point O en sécurité et la bordure 14-11 est ajoutée à la cartographie acquise par le robot.

Maintenant le robot est ordonné à retourner au point F; basé sur les nouvelles connaissances acquises par le robot, le planificateur suggère la trajectoire 9-14-F comme étant le plus court chemin pour atteindre F. En suivant ce chemin le robot atteint F sans interruption.

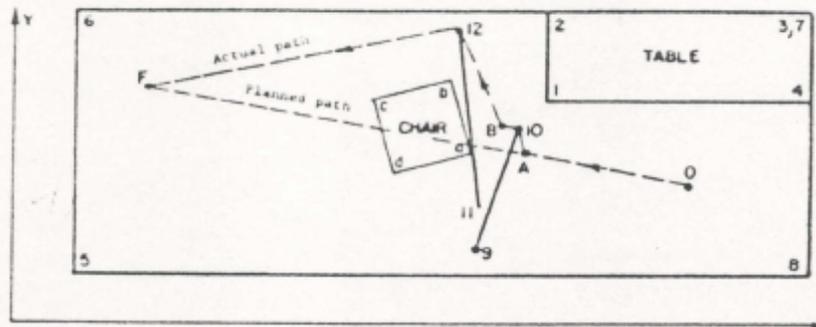


FIGURE II-10 (Prise de [BOR 88])

Test expérimental avec des obstacles inattendus.

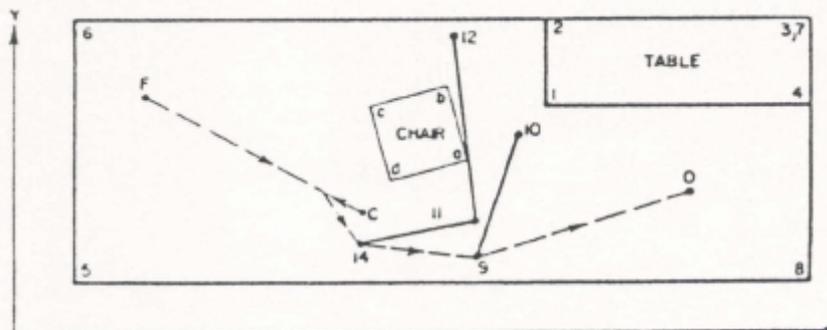


FIGURE II-11 (Prise de [BOR 88])

Test expérimental d'évitement de collision.

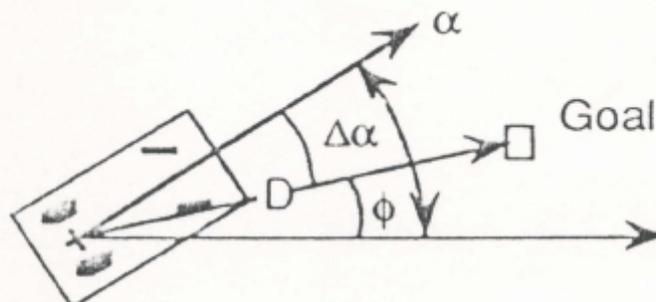


FIGURE II-12 (Prise de [CRO 91])

La procédure "TURN()" fait tourner le véhicule d'un angle $\Delta\alpha$

2 - 5 L'approche développée pour un robot de surveillance par JAMES.L.CROWLEY (d'après [CRO 91])

CROWLEY et AL ont proposé dans [CRO 91] une organisation logicielle pour le contrôle des actions d'un robot mobile de surveillance. Les tâches que doit accomplir un robot de surveillance sont:

- * être à la place <P> pendant l'intervalle de temps <T>.
- * inspecter la région <R> durant l'intervalle <T>.
- * signaler la détection de l'événement <X> à l'intérieur de la région <R> durant l'intervalle <T>.

Chaque tâche est décomposée en une liste de buts partiels; La réussite de cette décomposition n'est pas sûre puisque l'environnement n'est connu que partiellement; Dans l'architecture mise au point par CROWLEY [CRO 91]

(FIGURE I-8) le superviseur est installé à cet effet, son rôle est de surveiller l'exécution de chaque tâche et générer les actions nécessaires à l'accomplissement du but. Cette architecture propose une décomposition parallèle du système de perception et du système de locomotion. Le niveau action du système de locomotion est géré par la procédure "FOLLOW()" décomposée en 3 états:

- "FOLLOW-TURN"
- "FOLLOW-MOVE"
- "FOLLOW-HALT"

La procédure " FOLLOW-TURN " est appelée à chaque fois qu'une déviation trop large du chemin a lieu; cette procédure appelle à son tour la procédure "MOVE()" d'une distance "0" pour arrêter tout mouvement en avant, puis la procédure "TURN-PATH" pour vérifier si le robot peut tourner sans collision, et enfin la procédure "TURN()" pour faire tourner le véhicule d'un angle $\Delta\alpha = \alpha - \phi$ (FIGURE II-12), où α est l'orientation du véhicule et ϕ la direction au but courant. Dans le cas où l'angle de déviation n'est pas trop large le robot s'oriente au but en se déplaçant. Quant au système de perception, il sert les deux principaux points de la navigation:

1) la détection des limites de l'espace libre: c'est un ensemble de positions et orientations que peut prendre le robot sans qu'il y ait collision avec d'autres objets.

2) l'estimation de la position: qui permet au mobile de se localiser par rapport au but. Dans le cas du robot de surveillance de JAMES.L.CROWLEY [CRO 91], cette tâche est accomplie par un modèle pré-enregistré et une ceinture de 24 sondes ultrasonores autour du mobile (FIGURE II-13).

Ce système est géré par les procédures "FREE-PATH", "TURN-PATH", et "FIND-PATH"; Les procédures "FREE-PATH" et "TURN-PATH" vérifient si le mobile peut se déplacer ou tourner sans collision, en cherchant dans le modèle locale si un obstacle existe sur le chemin du robot, dans le cas de "FREE-PATH" ce chemin est approximé par un rectangle (FIGURE II-14). Et dans le cas de "TURN-PATH" c'est la zone balayée par le robot au court de la rotation (FIGURE II-15). La procédure "FIND-PATH" peut être considérée comme planificateur récursif, elle calcule les points d'évitements des obstacles, dans les deux sens, ces points sont déterminés par une projection perpendiculaire de chaque extrémité du segment de l'obstacle sur le chemin original (FIGURE II-16). Et chaque nouvelle trajectoire est testée par la procédure "FREE-PATH", si elle retourne un bilan négatif, la procédure "FIND-PATH" s'exécute encore une fois pour donner d'autres points d'évitements et un chemin (FIGURE II-17). La procédure "FIND-PATH" génère une liste de chemins possibles, et choisit le chemin le plus court qui possède le nombre le moins élevé de points d'évitements.

Dans un environnement très encombré, la marge de sécurité est réduite au stricte minimum, en contre partie, le risque de collision avec un obstacle imprévu augmente. Pour permettre au système mobile de naviguer en sécurité dans cet espace encombré J.L.CROWLEY et AL ont ajouté une procédure réactive à la procédure "FOLLOW()", il s'agit de la procédure

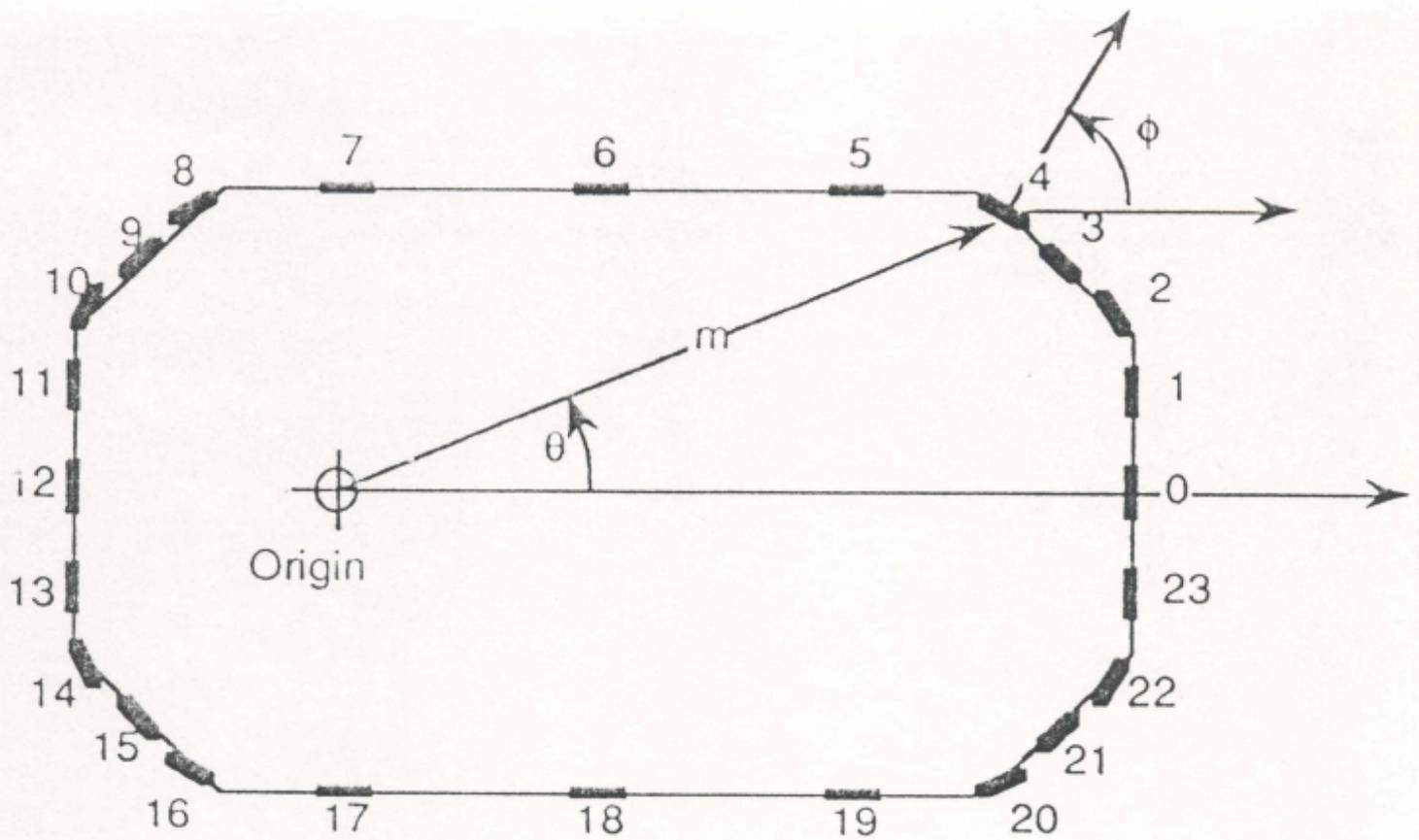


FIGURE II-13 (Prise de [CRO 91])

Plan des coordonnées du robots, et disposition des 24 sondes ultrasonores.

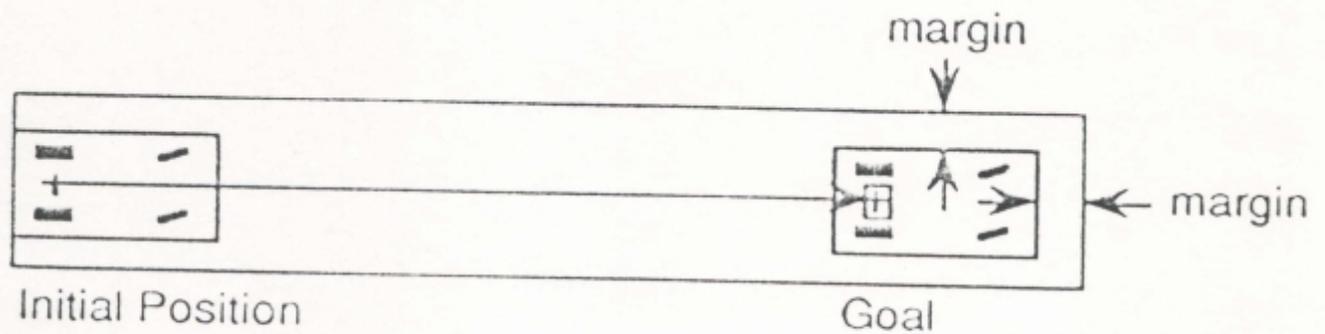


FIGURE II-14 (Prise de [CRO 91])

Dans le cas de la procédure "FREE-PATH" le chemin du robot est approximé par un rectangle.

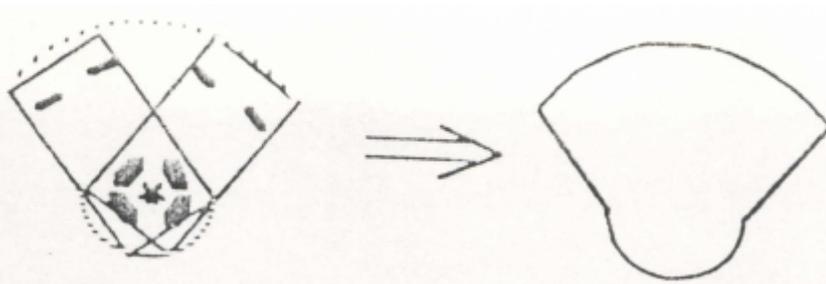


FIGURE II-15 (Prise de [CRO 91])

Zone balayé par le robot au court de la rotation prise en considération par la procédure "TURN-PATH".

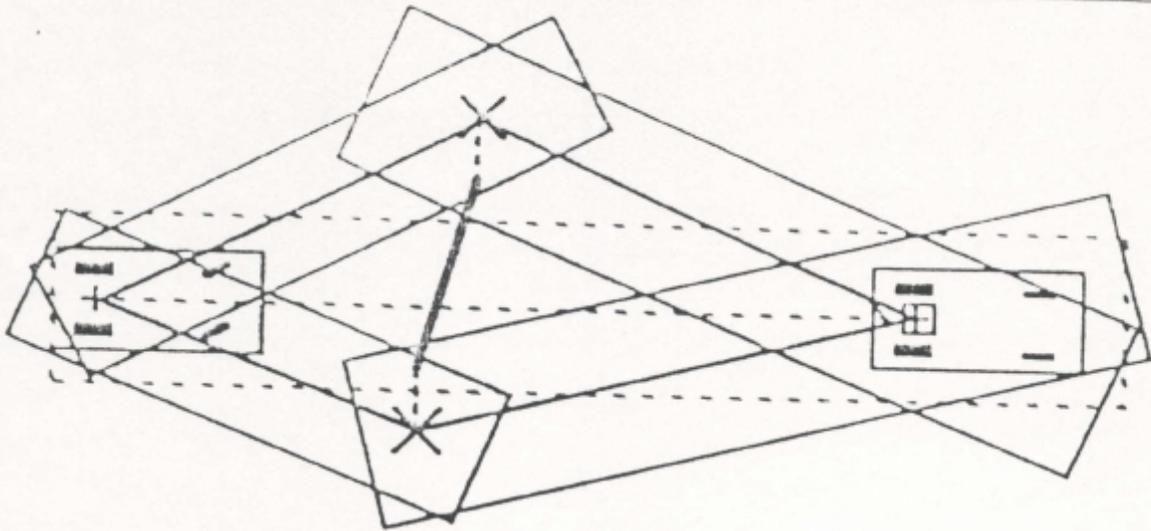


FIGURE II-16 (Prise de [CRO 91])

Les deux points d'évitement sont déterminés dans les deux sens de l'obstacle par une projection perpendiculaire de chaque terminaison du segment de l'obstacle sur le chemin originale. ce calcul se fait par la procédure "FIND-PATH".

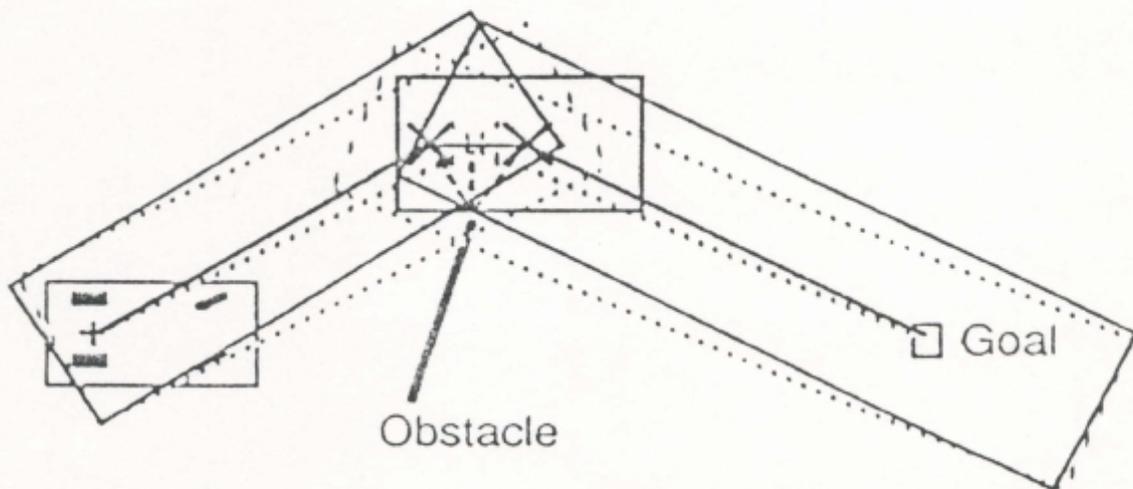


FIGURE II-17 (Prise de [CRO 91])

Chaque nouvelle trajectoire est testée par la procédure "FIND-PATH"

"MOVE-REFLEX", elle arrête temporairement le mobile à chaque fois qu'un capteur de la face avant détecte un obstacle; et "TURN-REFLEX" permet de repousser le robot d'un angle:

$$\Delta\alpha_{\text{reflex}} = \Sigma\Delta\alpha_k$$

avec $k=\{1,2,3,4,21,22,23\}$

et $\Delta\alpha_k = \text{Arctg}[(Y_{ok} - D_r)/x_{ok}]$

(x_{ok}, Y_{ok}) les coordonnées du kième capteur responsable de la détection

D_r : distance de répulsion désirée.

3 NAVIGATION SUR DES TERRAINS INCONNUS

3 - 1 Introduction

Les techniques de planification de trajectoire ne sont pas directement utilisables dans le cas où le terrain est complètement inconnu, le robot trace une cartographie de son environnement au cours de son évolution, en s'appuyant sur les informations de son système de perception; on parle dans ce cas d'une modélisation dynamique de l'environnement traitée dans les approches proposées dans cette section.

3 - 2 Navigation sur des terrains inexplorés d'après LYENGAR et RAO [LYE 86]

Récemment, LYENGAR et RAO [LYE 86] ont développé une technique sur la navigation, basée sur le concept de l'apprentissage; cette technique n'a besoin d'aucun modèle du terrain, il est par contre construit graduellement au cours des traversées du robot.

Ce processus opère en deux modes de base: la navigation locale et la navigation globale. Cette dernière consiste en deux composants: la planification de la trajectoire en utilisant le modèle partiel du terrain déjà construit, et l'apprentissage, en intégrant les informations issues des capteurs. Initialement les trajectoires sont planifiées et traversées dans le mode local, elles sont entièrement basées

sur les informations des capteurs, ces chemins partagent l'environnement en un ensemble de polygones. La **FIGURE II-18** représente un terrain inexploré et à ce moment aucune information sur ce terrain n'est disponible au robot.

Il commence alors ses traversés dans le mode de navigation locale en scrutant le terrain autour de la ligne SD reliant la source S à la destination D. Si cette ligne est coupée par un obstacle, deux points d'inflexions sont alors calculés de part et d'autre de SD et le robot transite par l'un des deux points (le choix du point est basé sur la minimisation d'une certaine métrique) et la même stratégie est appliquée à chaque point intermédiaire.

La **FIGURE II-19** illustre la navigation du mobile de S_1 à D_4 , ce chemin est parcouru en sept (7) traversées, des sources $S_1, D_1, S_2, D_2, S_3, D_3$ et S_4 aux destinations $D_1, S_2, D_2, S_3, D_3, S_4$ et D_4 , respectivement; le terrain est donc partagé en un ensemble de polygones et à ce moment le seul modèle du terrain connu par le mobile est cet ensemble de polygones.

Ces polygones sont classés comme suit :

- a) les polygones libres représentent une région dépourvue d'obstacles.
- b) des polygones inexplorés représentent une région dont l'intérieur n'est pas encore scruté.
- c) les polygones obstacles représentent une région où il existe un obstacle à l'intérieur.

Initialement, tous les polygones sont inexplorés. Considérons, la navigation de S à D dans le mode globale (**FIGURE II-20**).

Les sommets, les plus proches de S et D, sont respectivement S^* et D^* . La traversée de S à S^* se fait dans le mode local, le polygone P_2 de la **FIGURE II-19** est scruté à partir de S^* , il est décomposé en polygones libres P_{21}, P_{23} et un polygone obstacle P_{22} , le robot dans ce cas transite à S_5 et explore P_4 . P_4 est décomposé en polygones libres P_{41} ,

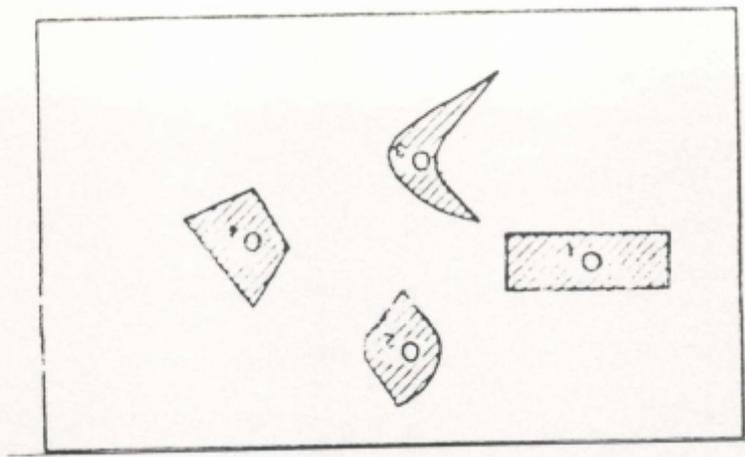


FIGURE II-18 (Prise de [LYE 86])

Terrain inexploré

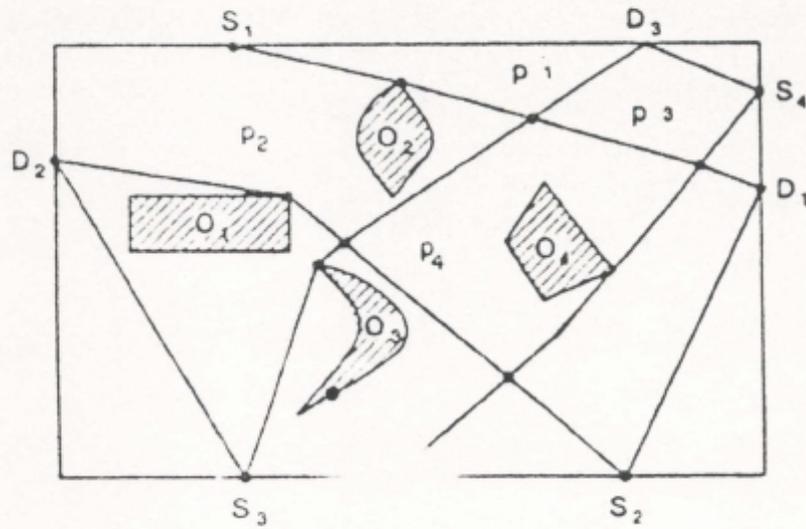


FIGURE II-19 (Prise de [LYE 86])

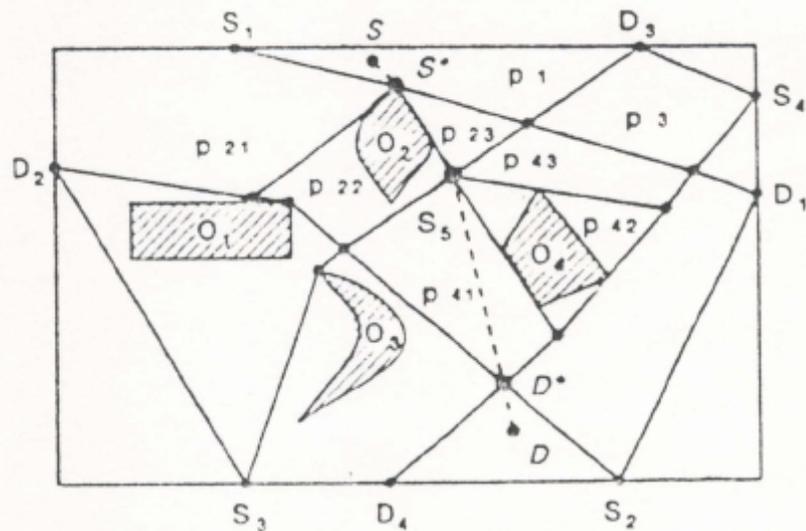


FIGURE II-20 (prise de [LYE 86])

Navigation de S a D en mode globale

P_{43} et un polygone obstacle P_{42} , le mobile voyage à D^* à travers le polygone libre P_{41} ; et de D^* à D la navigation locale est utilisée. Remarquez que les obstacles sont de plus en plus encerclés et les polygones libres adjacents sont concaténés pour former un seul et dans ce cas ils deviennent de plus en plus larges. L'implémentation de cette technique a été faite sur le robot **HERMIES-II [WEI 85]**, équipé d'un ordinateur à bord pour le système de contrôle et un autre à l'extérieur pour la planification.

Le temps que met le robot pendant son voyage de S à D est une fonction de:

- $T_r = \Sigma T_t(P_i)$ où $T_t(P_i)$ est le temps que met le robot pour aller d'un point S_i à un autre D_i à l'intérieur du polygone P_i

- $T_s = \Sigma T_s(P_i)$ où $T_s(P_i)$ est le temps mis par les capteurs pour scruter l'intérieur de P_i . si P_i est déjà scruté alors $T_s(P_i) = 0$

- $T_u = \Sigma T_u(P_i)$ où $T_u(P_i)$ le temps nécessaire pour la mise à jour des informations sur le polygone P_i

- $T_p = \Sigma T_p(P_i)$ où $T_p(P_i)$ est le temps que met le système de planification pour planifier une trajectoire d'un point S_i à un autre D_i à l'intérieur de P_i .

La durée totale que met le mobile pour aller de S à D est comprise entre $T_p(P_1) + T_t(P_1)$ (la source et le but sont à l'intérieure du polygone libre P_1), et $T_t + T_p + T_s + T_u$. Les durées T_t et T_s dépendent de la structure mécanique du robot, elles ne sont pas sous le contrôle du modèle du terrain alors que T_p et T_u peuvent être minimisée en utilisant un modèle adéquat.

3 - 3 Le guidage sonore d'après KUC et VIARD [KUC 91]

En 1991 **KUC** and **VIARD** développèrent une densité de scrutation des ultrasons par un seul transducteur, permettant un déplacement du robot par pas de Δ en sécurité (**FIGURE II-21**), et essayèrent de l'appliquer pour naviguer un robot autonome dans un environnement qui n'est pas structuré; Dans

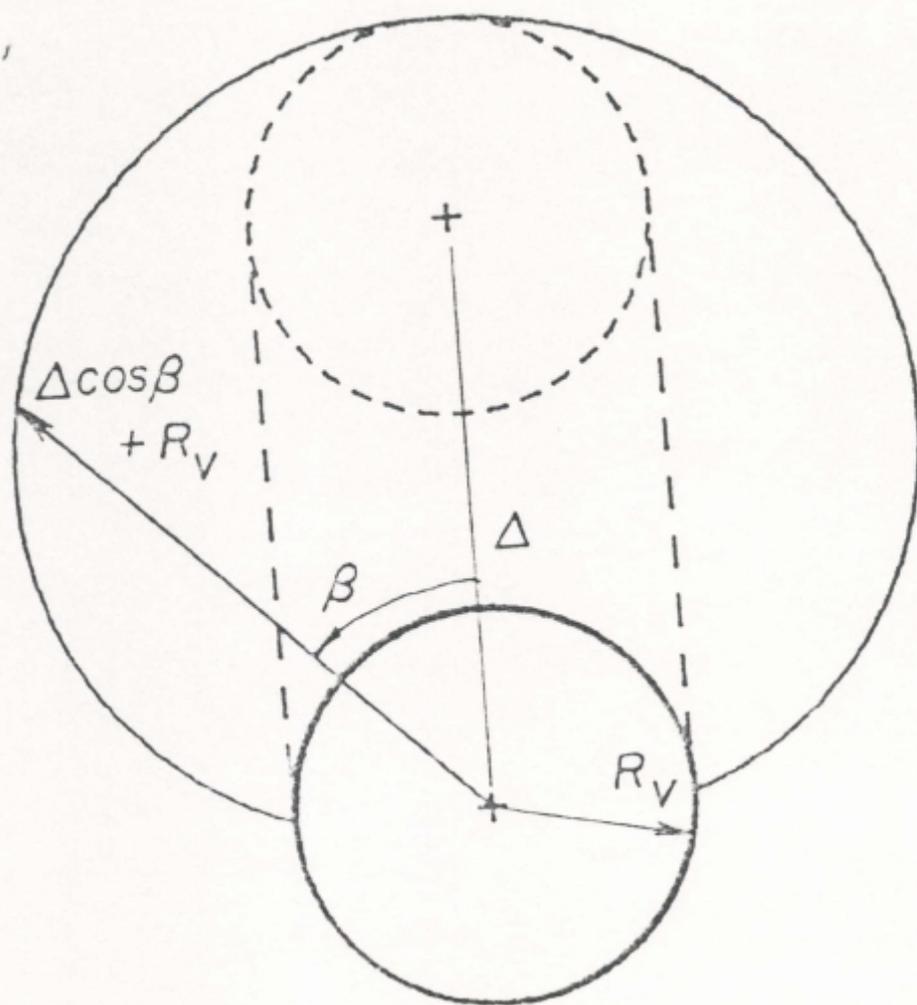


FIGURE II-21 (prise de [KUC 91])

Espace de sécurité du robot

cette approche, l'environnement est discrétisé en une grille de cellules carrées de dimension $(\Delta/\sqrt{2})$ (le pas Δ représente la diagonale d'une cellule; configurée dans un tableau à deux dimensions dans la mémoire de l'ordinateur. Le problème est de trouver un chemin à la destination sans collision; la position courante et la destination sont supposées connues.

Parmi les 8 cellules adjacentes à la cellule où il se trouve, le robot essaye toujours de prendre celle qui minimise la distance à la destination.

Pour avoir un modèle approximatif durant l'évolution du robot; les cellules occupées sont codées dans la mémoire de l'ordinateur en 2D par:

(1) position courante, (2) destination, (3) n'est pas encore visitée, (4) déjà visitée, et (5) occupée par un obstacle.

L'algorithme permet une translation amenant le véhicule de sa position courante à la cellule adjacente qui n'est pas encore visitée et qui minimise la distance à la destination, si toutes les cellules adjacentes sont déjà visitées, un chemin est pris à travers les cellules déjà visitées à la cellule la plus proche qui n'est pas encore visitée; le parcours se termine si la destination est atteinte ou si la région occupée par le véhicule est entourée par des obstacles et toutes les cellules sont déjà visitées.

Cet algorithme a été implémenté sur le robot DD (DEW-DROP) représenté à la **FIGURE II-22** équipé de trois roues motrices actionnées par six moteurs pas à pas, trois moteurs pour le déplacement du véhicule et trois pour l'orientation des roues; Le capteur à ultrason est placé au sommet du moteur avant, le balayage se fait par rotation du véhicule; les informations du capteur sont transportées à l'ordinateur (DEC PDP 11/34a, opérant sous le RT-11) à travers un long câble servant aussi à alimenter les moteurs et la carte électronique qui se trouve à bord.

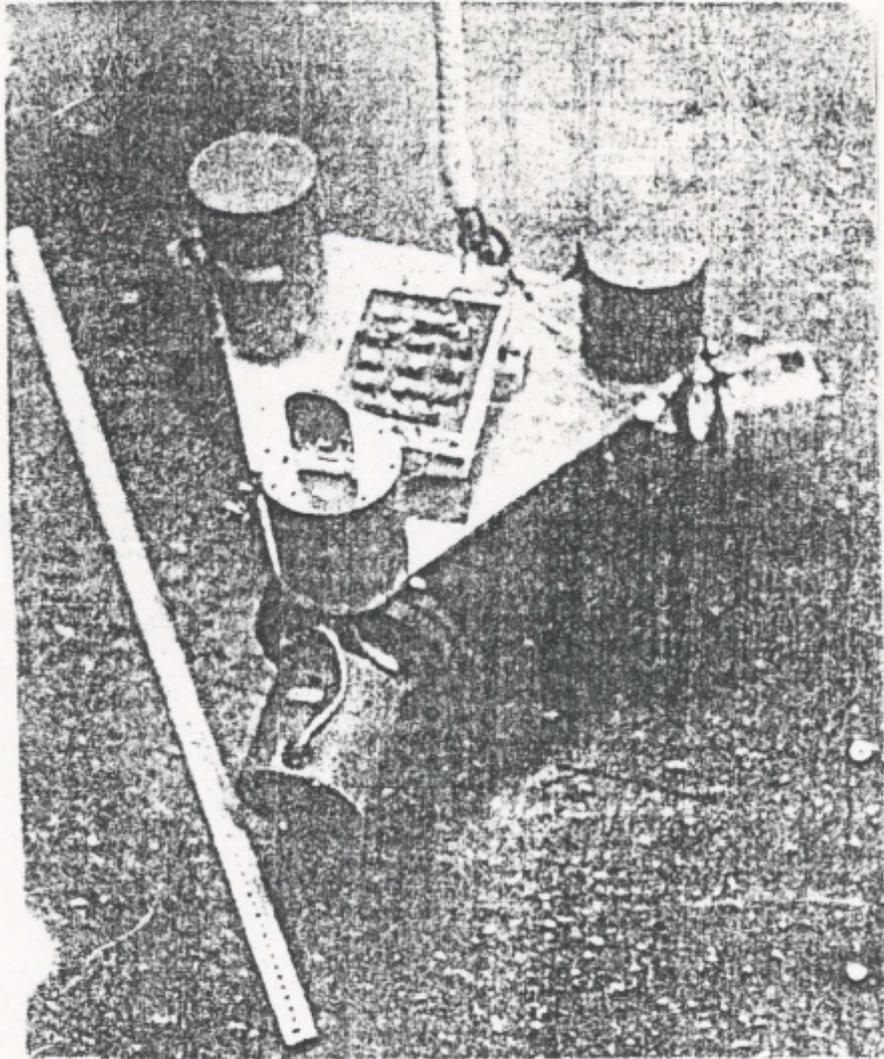


FIGURE II-22 (prise de [KUC 91])

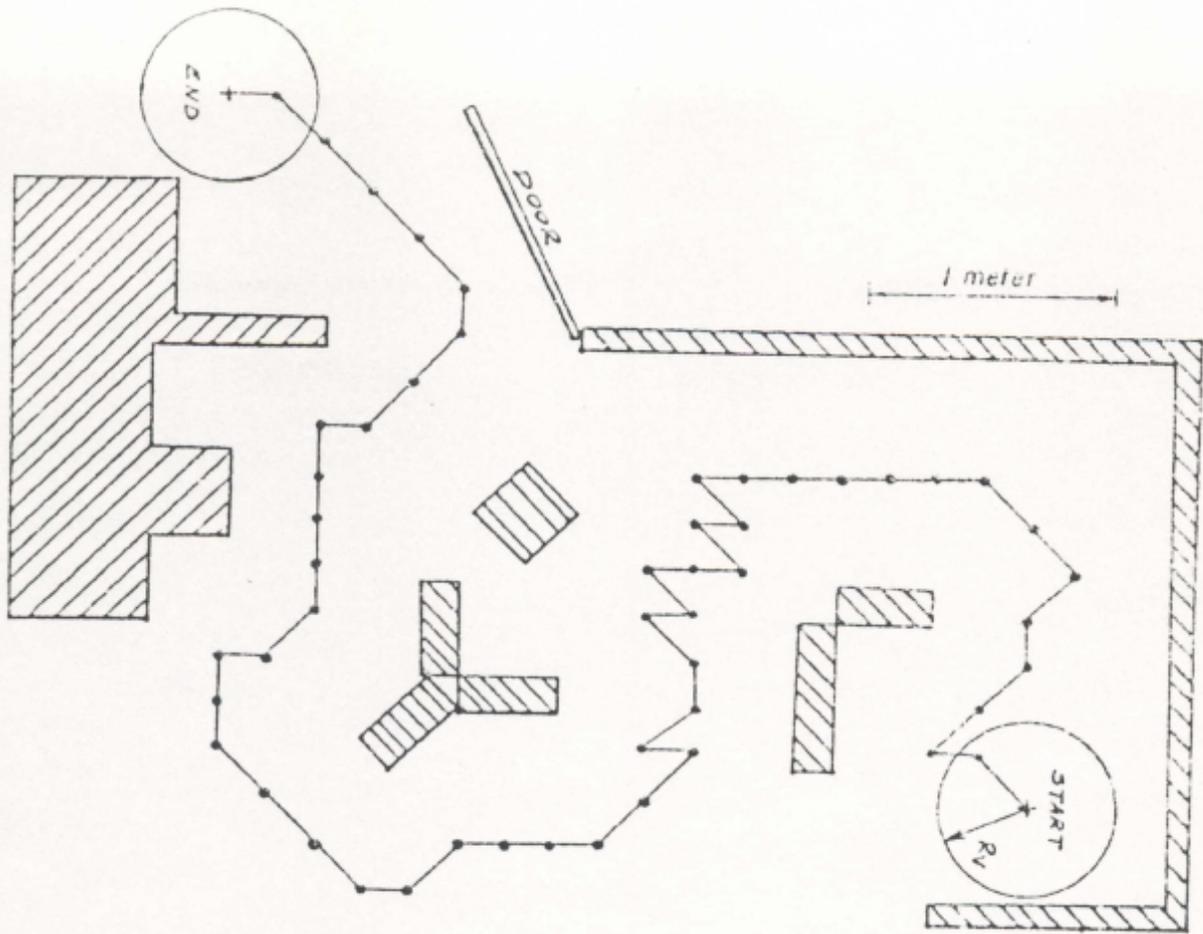
Le robot mobile DEW-DROP

A chaque position courante, la direction de translation qui minimise la distance à la destination est prise et une scrutation de 88° est initialisée dans la direction du mouvement; le balayage s'arrête si un obstacle est détecté dans la direction du mouvement, la cellule correspondante est marquée occupée dans la mémoire, et une nouvelle direction est choisie; si aucun obstacle n'est détecté le robot continue son mouvement à la position désirée. La **FIGURE II-23** montre un exemple de trajectoire empreintée par le robot DD et la façon de mémoriser l'environnement.

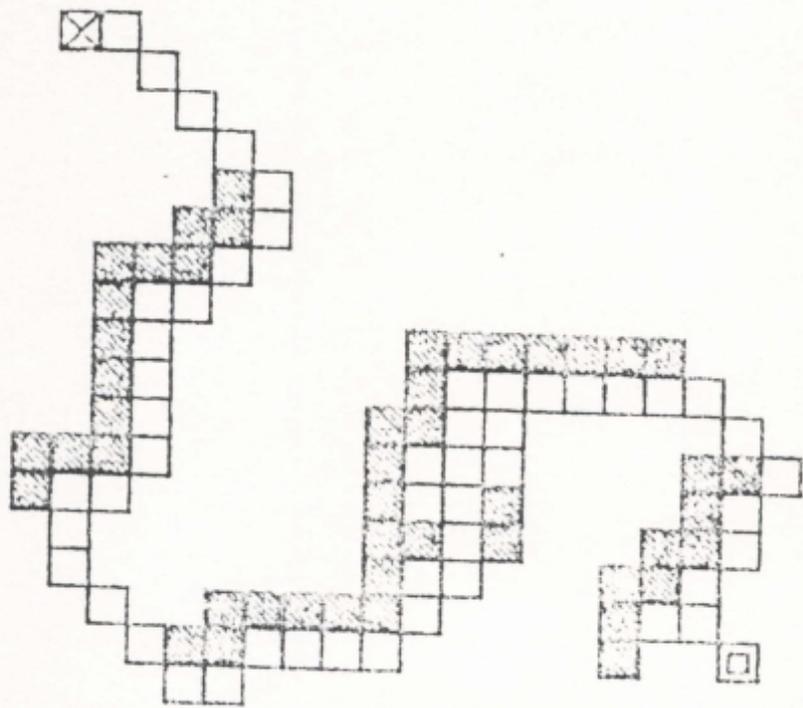
3 - 4 Evitement rapide de collision d'après BORENSTEIN et KOREN [BOR 91a]

Inspirés par la méthode des grilles de certitude développée par MORAVEC et ELFES à l'université de CARNEGIE-MELLON "C.M.U" [MOR 85] [MOR 88], et la méthode des potentiels proposée pour la première fois en 1985 par O.KHATIB [KHA 85]; J.BORENSTEIN et Y.KOREN [BOR 89] proposèrent en 1989 une nouvelle méthode d'évitement d'obstacles en temps réel, parue sous le nom de "V.F.F" (Vector Force Field). La méthode V.F.F utilise une histogramme grille cartésienne à deux dimensions (C) pour la représentation des obstacles. Comme dans le cas de la grille de certitude de C.M.U, chaque cellule (i,j) possède une valeur C_{ij} indiquant l'existence de l'obstacle dans cette cellule, appelée valeur de certitude (C.V); contrairement à la méthode de C.M.U, le C.V d'une seule cellule est incrémenté pour chaque capteur, c'est la cellule qui se trouve sur l'axe acoustique à une distance d. Une distribution probabilistes est obtenue par un échantillonnage continu et rapide de chaque cellule lorsque le robot se déplace; de cette façon la même cellule et ses voisinages sont incrémentés à plusieurs reprises (**FIGURE II-24**), cela donne une distribution histogrammique de probabilité où les grandes valeurs de certitude sont obtenues pour les cellules proches de l'obstacle.

V



END
B



START

FIGURE II-23 (prise de [KUC 91])

A - Exemple de trajectoire prise par DD
 B - Représentation de l'environnement en 2D dans la mémoire de l'ordinateur

Au cours de la seconde étape, la méthode des potentiels est appliquée à l'histogramme grille. Au fur et à mesure que le véhicule se déplace, une fenêtre de $W_s \times W_s$ cellules l'accompagne (**FIGURE II-25**); cette région est appelée "région active C^* " et les cellules appartenant à cette région appelées "cellules actives C_{ij}^* "; le centre de la fenêtre coïncide à tout instant avec la position courante du robot. Chaque cellule active exerce une force répulsive F_{ij} sur le robot, l'amplitude de cette force virtuelle est proportionnelle à la valeur de certitude C_{ij} et inversement proportionnelle à d^2 (d est la distance séparant la cellule au centre du robot). A chaque itération toutes ces forces répulsives sont totalisées pour produire une force résultante répulsive F_r . Simultanément, la cible exerce sur le véhicule une force virtuelle attractive d'une amplitude constante F_t ; le robot suit alors la direction de la résultante R de F_r et F_t (**FIGURE II-25**).

Durant l'implémentation de cette méthode deux problèmes majeurs ont été rencontrés:

- 1) le passage à travers l'ouverture des portes est parfois impossible dû aux forces répulsives créées par le cadre.

- 2) Un autre problème peut surgir lorsque le robot se déplace dans des couloirs étroits: Le mouvement du robot reste stable lorsqu'il se déplace au milieu du couloir; une déviation légère de cette trajectoire donne naissance à une très grande force répulsive créée par le mur le plus proche; elle oblige le robot d'aller de l'autre côté de la ligne du milieu et le même processus se répète avec l'autre mur, ceci donne naissance à un mouvement oscillatoire et instable.

Pour remédier à ces problèmes, **J.BORENSTEIN** et **Y.KOREN** développèrent en 1991 [**BOR 91b**], une nouvelle méthode d'évitement d'obstacle en temps réel appelée V.F.H (Vector Field Histogramme). Dans cette méthode, la région active C^* de la méthode V.F.F est décomposée en n secteurs d'angle α (**FIGURE II-26**), le graphe obtenu est appelé "histogramme polaire H "; le contenu de chaque cellule active est considéré

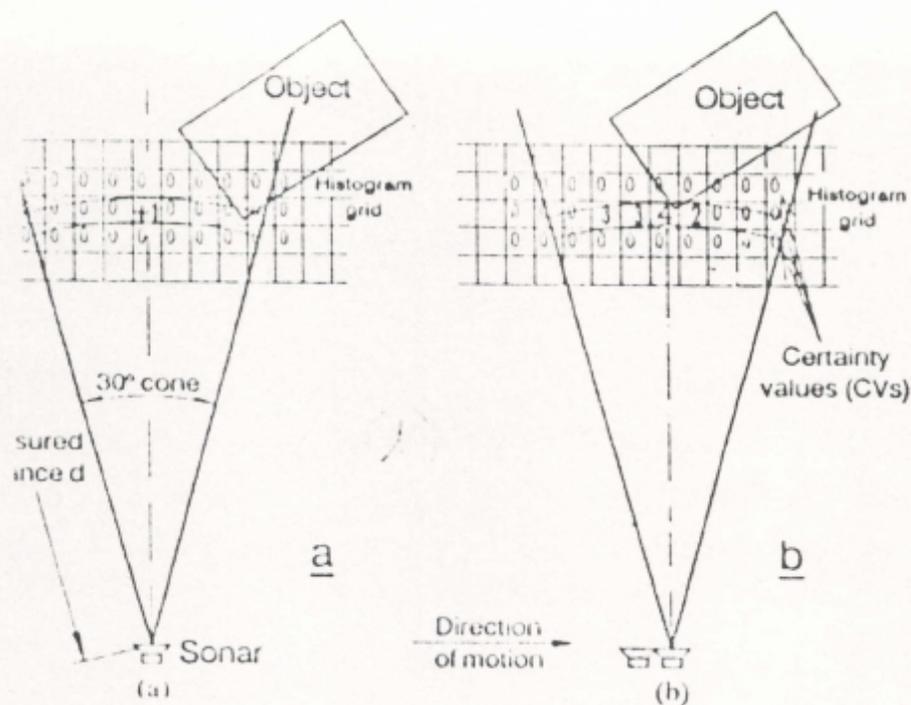


FIGURE II-24 (prise de [BOR 91a])

- a) Une seule cellule est incrémentée pour chaque capteurs
- b) Lorsque le robot se déplace le CV des cellules s'incrémente

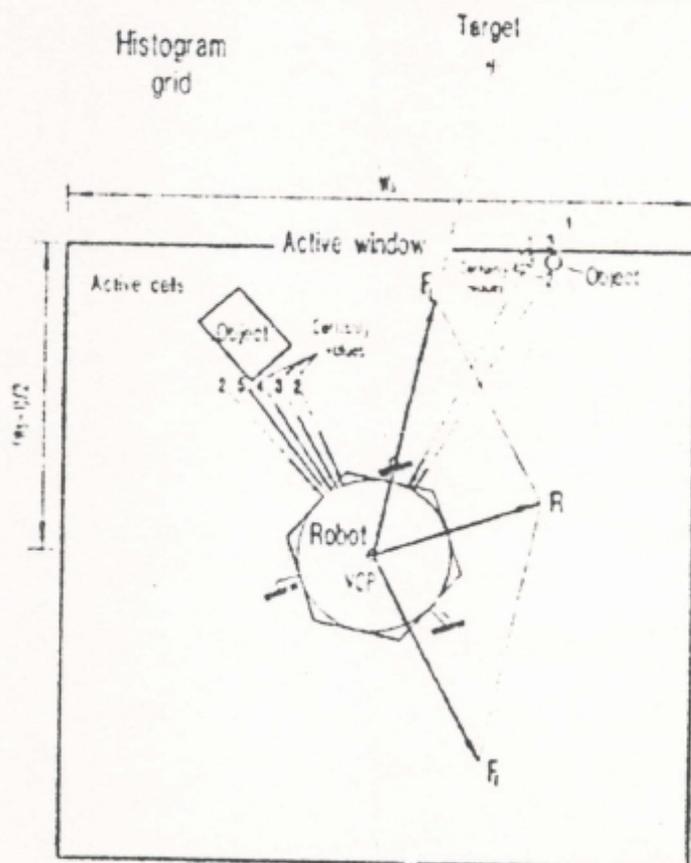


FIGURE II-25 (prise de [BOR 91a])

La fenêtre de $W_s \times W_s$ cellules accompagne le robot, le robot suit la direction de R

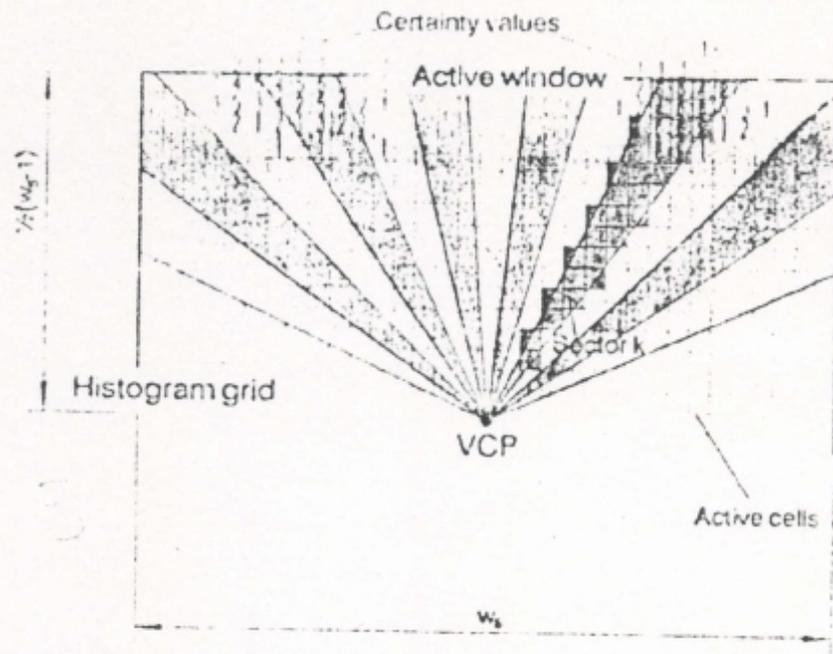


FIGURE II-26 (prise de [BOR 91a])

La fenêtre active est décomposée de n secteurs d'angle α

comme un vecteur obstacle. La méthode V.F.H a été implementée sur CARMEL (FIGURE II-27) et testée dans l'environnement représenté à la FIGURE II-28. L'histogramme polaire obtenu correspondants à la position du robot est représenté à la FIGURE II-29 où les espaces sombres correspondants aux secteurs qui ont une très grande densité polaire d'obstacle h_k ($h_k > \text{Seuil}$) et les espaces clairs correspondants aux secteurs qui ont une densité polaire d'obstacle inférieure à un certain seuil ($h_k < \text{Seuil}$); Ces espaces sont appelés "des vallées candidates" où le véhicule peut se déplacer en sécurité. Le robot prend alors la direction de la vallée candidate la plus proche à la direction de l'objectif et plus exactement la direction du secteur qui se trouve au milieu de cette vallée. Les vallées trop larges sont limitées à 18 secteurs. Pour ces vallées le robot essaye toujours à maintenir une distance ds à l'obstacle (FIGURE II-30).

A travers les FIGURES II- 30 et II-31 on voit clairement que les 2 problèmes (Passage à travers l'ouverture des portes, et dans les couloirs étroits) rencontrés dans la méthode V.F.F sont complètement éliminés.

II - 4 CONCLUSION

le point commun de toutes ces approches, est la résolution des problèmes de décision pour que le robot mobile puisse mener à bien sa mission sur des terrains parsemés d'obstacles. La complexité de ces approches, dépend du taux de connaissance de l'environnement avant la mission, et le type de capteurs utilisés pour la détection des situations imprévues; Les capteurs ultrasonores, malgré leurs imprécisions sont les plus utilisés.



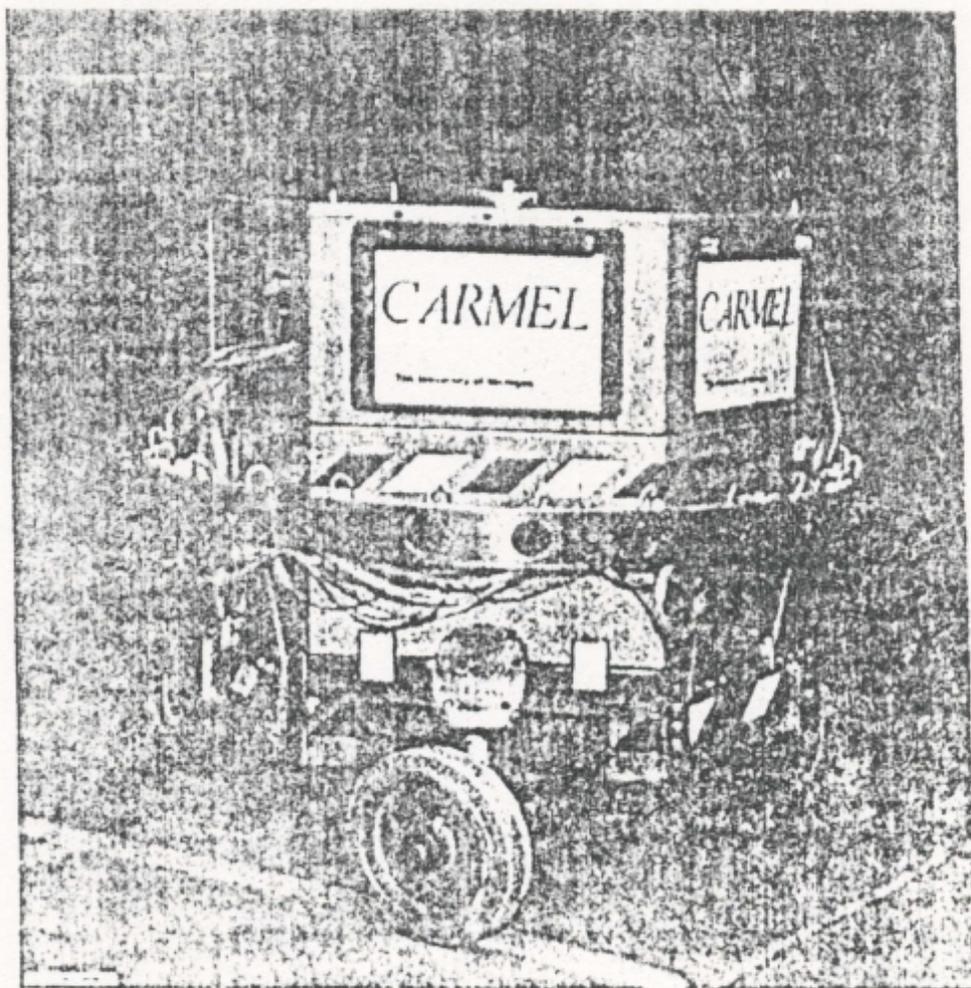
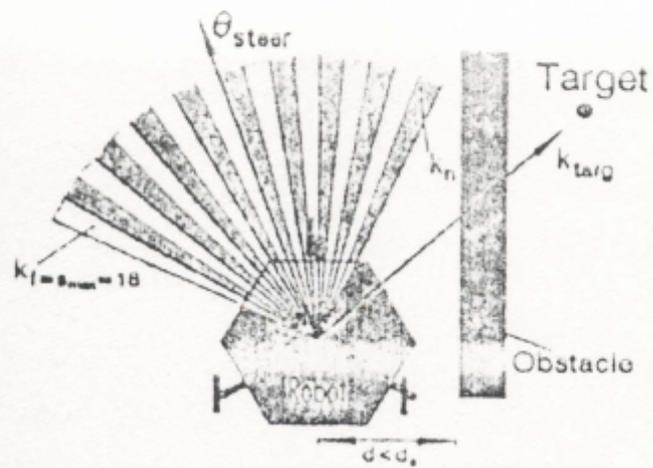
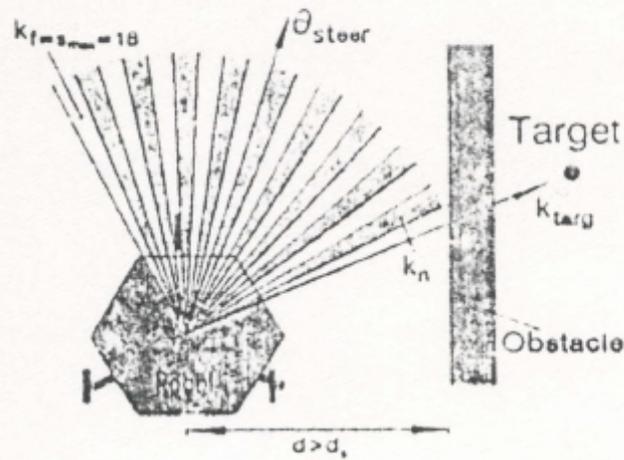


FIGURE II-27 (prise de [BOR 91a])

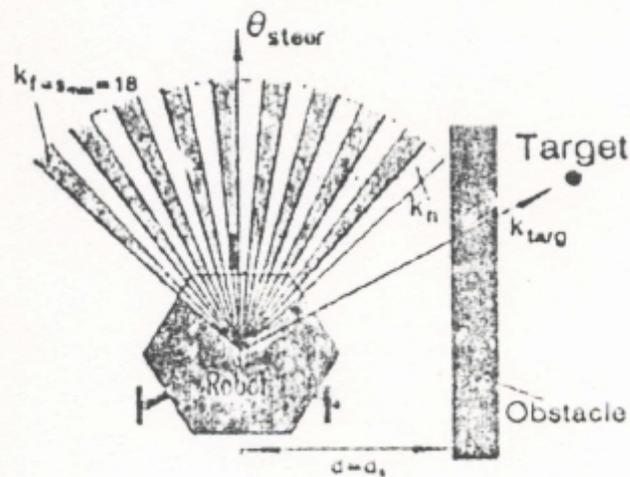
Le robot mobile de l'université de MICHIGAN



(a)



(b)



(c)

FIGURE II-30 (prise de [BOR 91a])

Le robot prend la direction du milieu de la vallée candidate la plus proche de l'objectif, en essayant de maintenir une distance d_s à l'obstacle

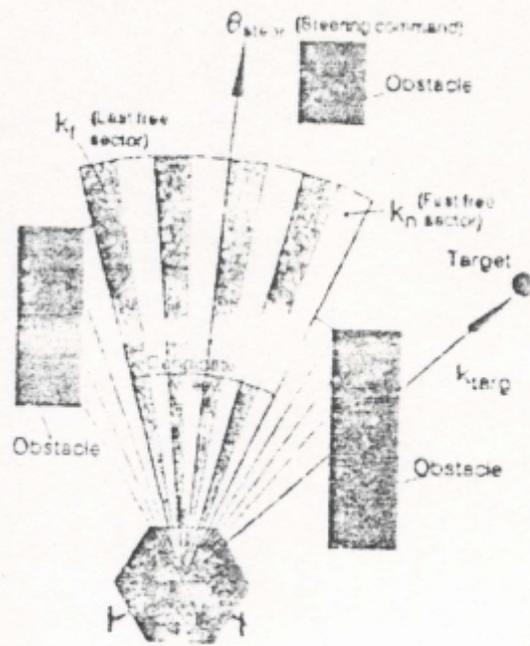


FIGURE II-31 (prise de [BOR 91a])

Le robot prend toujours
la direction du milieu
de la vallée candidate
même si le passage
est étroit

CHAPITRE III
ARCHITECTURE ET CARACTERISTIQUES
MATERIELLES DE RoMoSA.I

1 INTRODUCTION

Généralement, le test des algorithmes de planification de trajectoires et d'évitement de collisions, nécessite la présence d'un support expérimental, à cet effet nous avons conçu et mis au point le robot mobile **RoMoSA.I** (Robot Mobile Semi-Autonyme I) qui va servir de test des algorithmes que nous avons développé. Nous présentons dans ce chapitre l'architecture, la structure mécanique et les caractéristiques matérielles de **RoMoSA.I**, ainsi que la ligne de communication hertzienne établie, reliant **RoMoSA.I** à l'opérateur humain.

2 ARCHITECTURE

2-1 GENERALITES

La conception d'une architecture d'un robot mobile autonome expérimental, capable d'atteindre son objectif dans un environnement inconnu et peu ou pas structuré, est contrainte par des critères tels:

* **L'évolutivité:** Un robot mobile expérimental est en permanence en cours de développement, il est nécessaire de disposer d'une architecture tout à la fois ouverte et reconfigurable.

* **La réactivité:** Un robot mobile autonome est amené à se mouvoir dans des univers réalistes et souvent dynamiques; pour éviter toute collision, la détection des situations imprévues est donc impérative.

* **L'autonomie décisionnelle:** L'autonomie d'un robot mobile repose sur sa capacité à produire une séquence d'actions permettant d'atteindre un but fixé à l'avance, dans un contexte réel.

* **La possibilité de l'intervention humaine:** Il est indispensable de fournir à un expert humain des outils et un accès privilégié à tous les niveaux de l'architecture pour intervenir en cas d'échec total. L'autonomie dans ce cas est relativisée, on préfère utilisé le mot "semi-autonome".

2-2 L'ARCHITECTURE DEVELOPPEE

Pour prendre en compte toutes ces contraintes, nous avons proposé d'organiser l'architecture en quatre niveaux distincts (FIGURE III-1):

* **Le niveau supervision:** Contrôle l'exécution du plan dans sa globalité, cela revient à générer les actions appropriées dans le but d'accomplir une tâche. Ce niveau interagit avec un opérateur humain pour l'affinement des tâches.

* **Le niveau planification:** Le but à atteindre, défini par le générateur de tâches (niveau supervision) est envoyé au niveau immédiatement inférieur (niveau planification) où se trouve le planificateur de trajectoires, celui-ci chapeaute une base de données pour en tirer la position des obstacles afin de pouvoir générer une trajectoire optimale au but en évitant les collisions.

* **Le niveau contrôle:** Ce niveau regroupe tous les modules d'asservissements et de contrôle des fonctionnalités de base du robot (capteurs et motricités):

- **Le module pilotage:** Reçoit du planificateur une liste de "segments" de la trajectoire et assure la production des consignes (positions consignes et l'état désiré du robot à chaque fin du segment) au module locomotion.

- **Le module locomotion:** Utilise les consignes générées par le module pilotage et les informations des capteurs de déplacements pour assurer la localisation du robot et l'asservissement en vitesse et en position des moteurs.

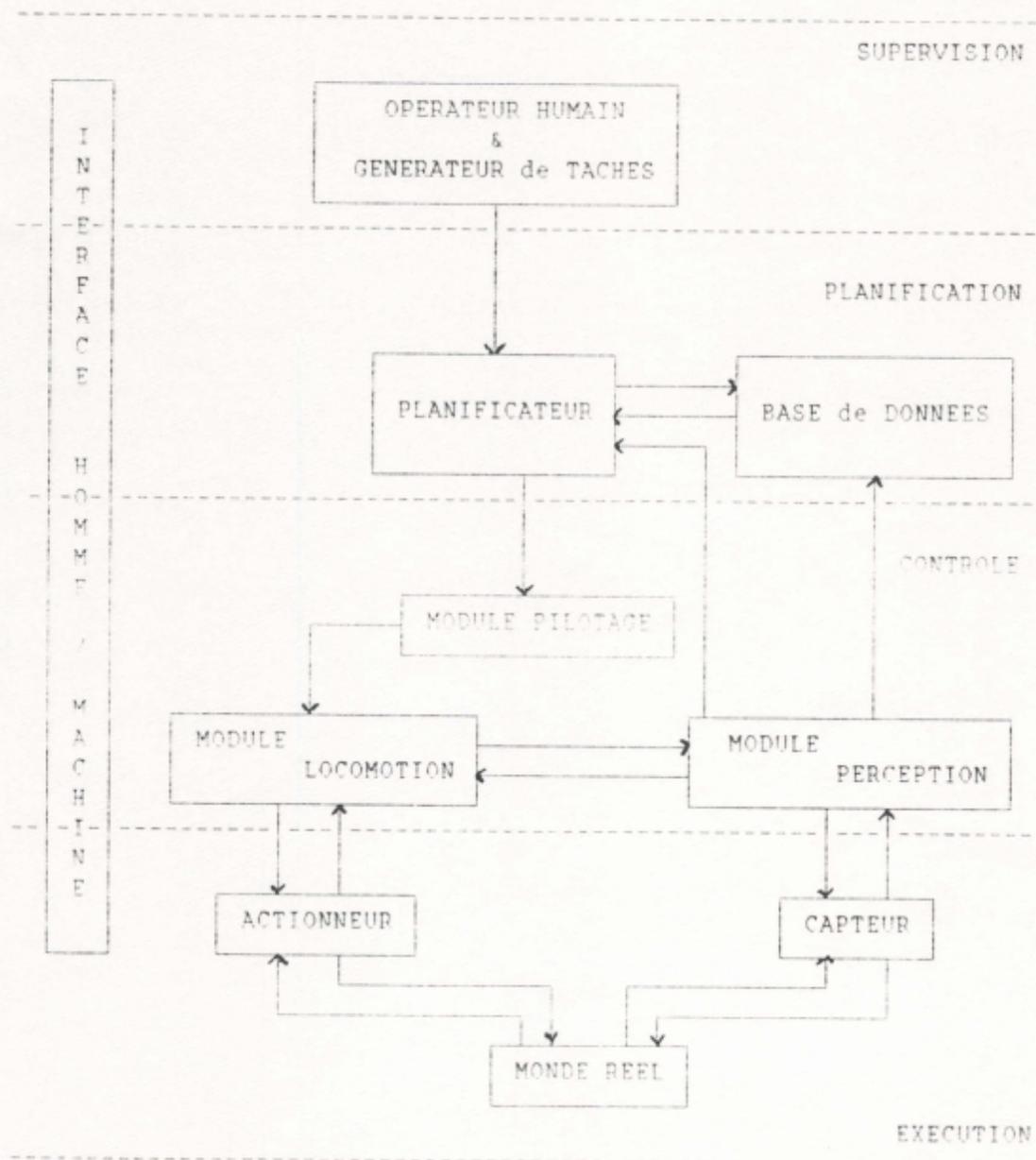


FIGURE III-1

ARCHITECTURE DEVELOPPEE POUR LE ROBOT RoMOSA I

- **Le module perception:** Assure la partie réactive du robot, en testant à tout instant l'état des capteurs (les capteurs tactiles et les capteurs de proximétrie), lorsque ces capteurs retournent un bilan négatif, le module perception demande au module locomotion la position du robot pour modifier la base de données, et il demande au planificateur de replanifier la trajectoire.

* **Le niveau exécution:** Cette couche organise toute la connaissance opératoire de base du robot, et assure une partie de sa réactivité face à l'environnement, elle contient tous les éléments nécessaires à la gestion des ressources physiques présentes (capteurs et actionneurs).

De plus, à tous les niveaux de la hiérarchie, une interface homme /machine fournit un moyen par lequel un opérateur peut observer et superviser le comportement de chaque niveau de la hiérarchie. Les chapitres suivants traitent en détail tous les niveaux de la hiérarchie.

3 CARACTERISTIQUES MATERIELLES DE RoMoSA.I

Nous présentons dans cette section la structure mécanique du robot qu'on a développée, les capteurs et l'équipement électronique embarqués sur **RoMoSA.I**, et nous détaillons par la suite la ligne de communication établie entre l'opérateur et le véhicule.

3 - 1 PLATE FORME MECANIQUE

Le châssis mécanique du robot **RoMoSA.I** (**FIGURE III-2**), construit à partir du plexiglas dur de 8mm d'épaisseur, est soutenu par deux barrettes en aluminium permettant au plateau de base de supporter une charge de 25kg sans déformation. La masse totale du robot avec tout l'équipement embarqué est d'environ 40kg; Il a une forme octogonale avec 0.7 m de longueur, 0,6 m de largeur, et 0,6 m de hauteur; à cette hauteur se trouve le support tournant des capteurs de proximétrie, surmonté par une antenne télescopique assurant une liaison hertzienne entre le véhicule et l'opérateur. De part et d'autre de ce support, deux butées de fin de course,

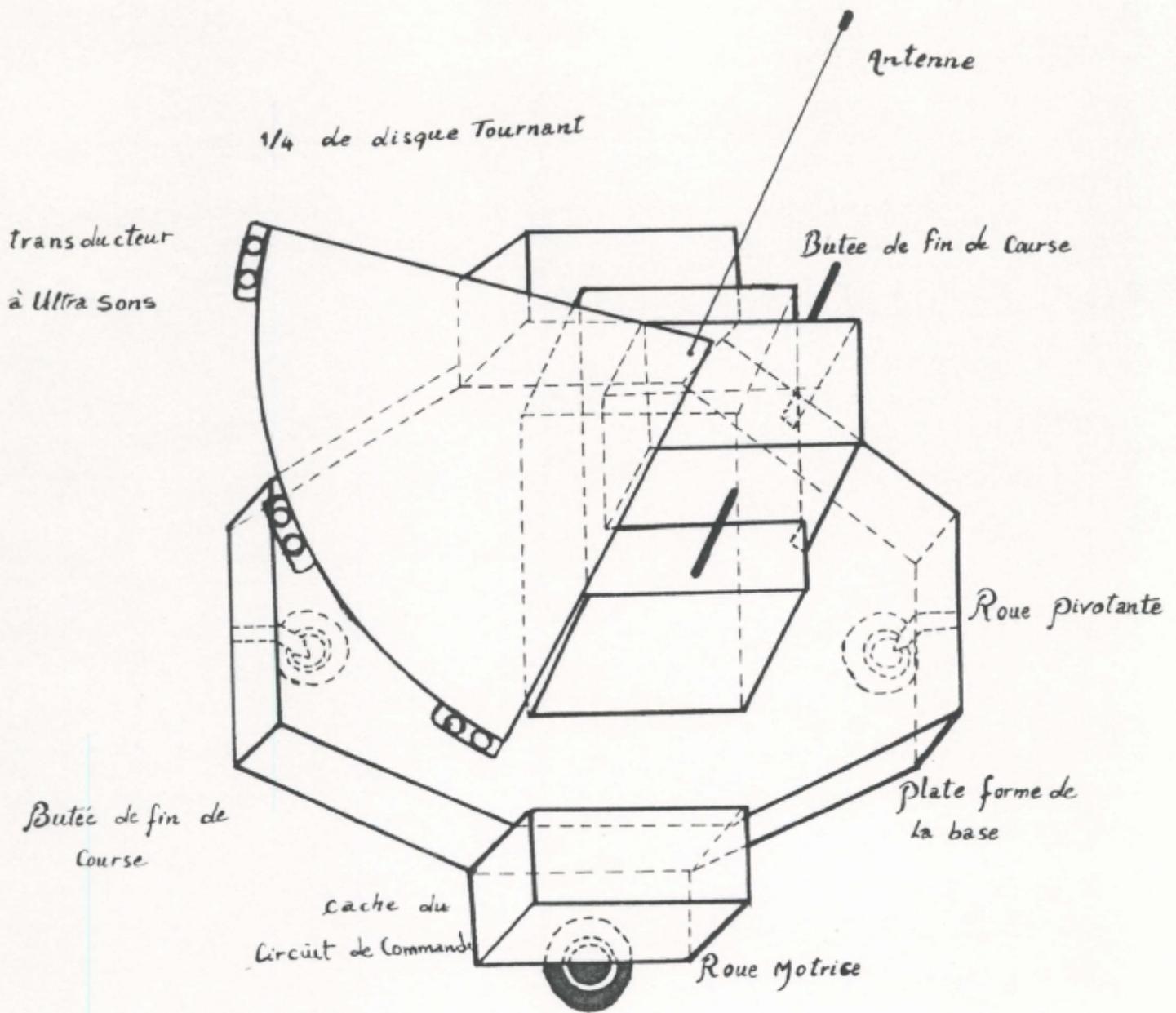


FIGURE III-2:
 STRUCTURE MECANIQUE DE ROMOSA.1

sont placées dans le but de lui permettre une rotation de $\pm 90^\circ$, par rapport à la direction du mouvement. **RoMoSA.I**, possède également des capteurs tactiles, montés sur un pare-choc, placé sur la partie la plus basse de sa face avant. Ce robot se meut à l'aide de deux roues odométriques indépendantes, situées sur l'axe transversal médian et deux autres roues libres à l'avant et à l'arrière pour sa stabilité. Les roues motrices sont actionnées par deux moteurs DC à aimants permanents, tournant à une vitesse angulaire maximale de 160 tr/min, soit une vitesse du véhicule de pratiquement 1 m/s (le diamètre des roues ≈ 0.125 m).

3 - 2 EQUIPEMENTS EMBARQUES

Nous présentons d'abord dans ce qui suit, les éléments perceptuels disposés sur **RoMoSA.I**, et nous procédons par la suite à l'analyse des équipements électroniques embarqués.

a - SYSTEME DE PERCEPTION

En effet, dans le domaine de la robotique mobile, la nécessité de percevoir l'environnement constitue un problème majeur. A cet effet, le robot **RoMoSA.I** est doté d'un certain nombre de capteurs classés en trois types:

- Les capteurs de proximétrie
- Les capteurs tactiles
- Les capteurs de vitesse et de déplacement.

*** Les capteurs de proximétrie**

On a utilisé comme capteur de proximétrie des transducteurs ultrasonores dont la fréquence de résonance est d'environ 40kHz \pm 500Hz, définie comme la fréquence d'un signal électrique d'excitation, qui provoque l'oscillation du transducteur, l'angle d'ouverture de ce type de transducteur, est de l'ordre de 30° . Ces capteurs au nombre de six (6), sont disposés sur la périphérie d'un support (1/4 de disque) tournant de $\pm 90^\circ$, par rapport à la direction du mouvement, assurant ainsi le balayage de la zone représentée à la **FIGURE III-3**.

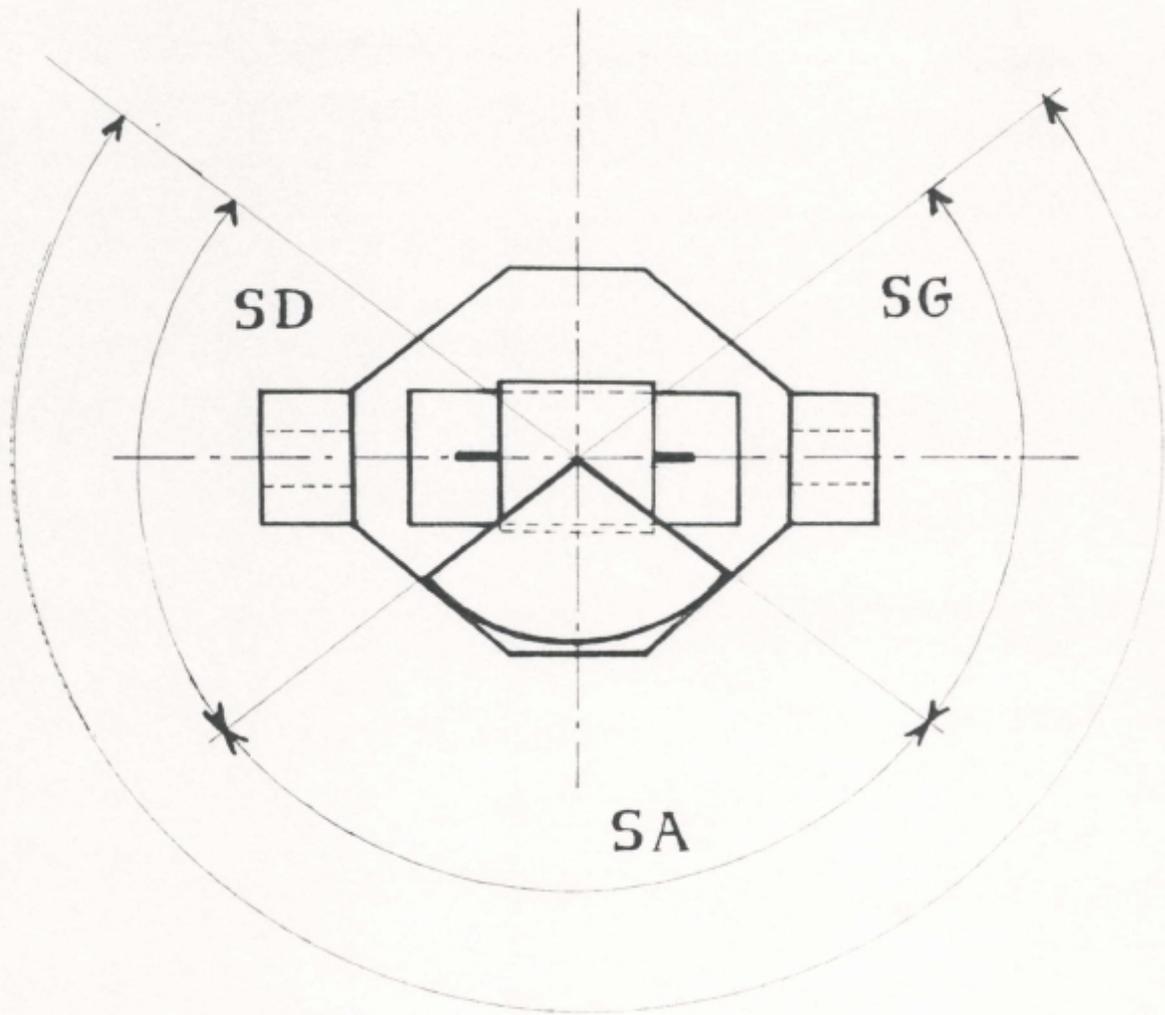


FIGURE III-3:
 TROIS ZONES DE SECURITES ASSUREES PAR LES SIX SONDES
 ULTRASONORES

* Les capteurs tactiles

Les capteurs tactiles utilisés, sont des microswitchs disposés tout au long du pare-choc avant du robot.

* Les capteurs de vitesse et de déplacement

Quant aux capteurs de vitesse et de déplacement, on a utilisé des codeurs optiques incrémentaux, placés sur l'axe des roues motrices. Les disques optiques, comportent 72 zones opaques et 72 zones transparentes, l'angle α entre deux zones transparentes ou entre deux zones opaques successives est de l'ordre de 5° (**FIGURE III-4**), ce qui correspond à un déplacement élémentaire de $dx = D\alpha/2$, avec $D = 0.125\text{cm}$ (diamètre des roues motrices) et $\alpha = 5^\circ$ ($dx = 6\text{mm}$), une telle résolution est tolérable en pratique, pour la majorité des applications civiles des robots mobiles telle que: la surveillance, la manutention, le nettoyage, les convoyeurs, etc....

b - SYSTEME ELECTRONIQUE

RoMoSA.I, comporte également, un certain nombre de modules électroniques, pour l'amplification et le traitement des informations issues de ces capteurs. La gestion de ces modules, et le contrôle des fonctionnalités de bases (motricité et capteurs), sont assurés par une carte électronique entièrement programmable, interfacée à un micro-ordinateur qui se trouve à bord du robot.

L'agencement, et la connectivité de tous ces modules est illustré dans le synoptique représenté à la **FIGURE III-5**:

Les moteurs sont pilotés par des temporisateurs programmables (P.I.T.1 et P.I.T.2), où sont également connectés les tachymètres optiques. L'ordinateur reçoit de ces temporisateurs à chaque Δt ($\Delta t = 50\mu\text{s}$), une information sur la vitesse et la position relative du robot (ces informations sont entièrement basées sur les tachymètres optiques solidaires des roues motrices). L'ordinateur, reçoit également des informations sur l'espace d'évolution du robot, à travers une interface parallèle programmable (P.P.I), où sont connectés les modules de perception constitués autour de 3 sondes ultrasonores de 40khz, et un module de traitement

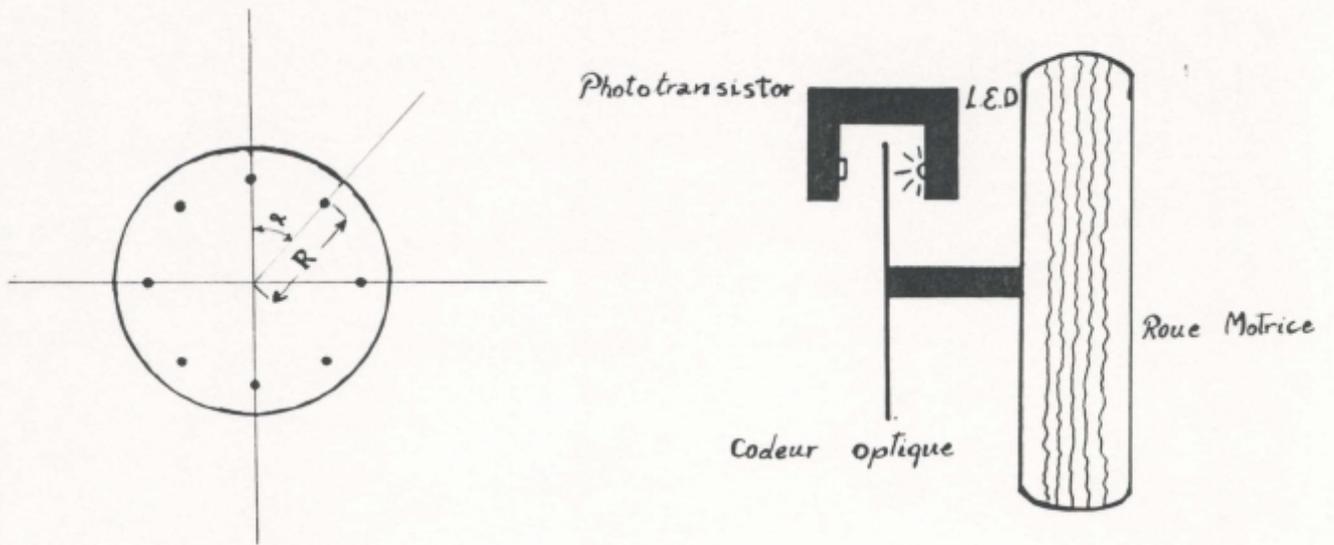


FIGURE III-4

CAPTEUR DE VITESSE ET DE DEPLACEMENT

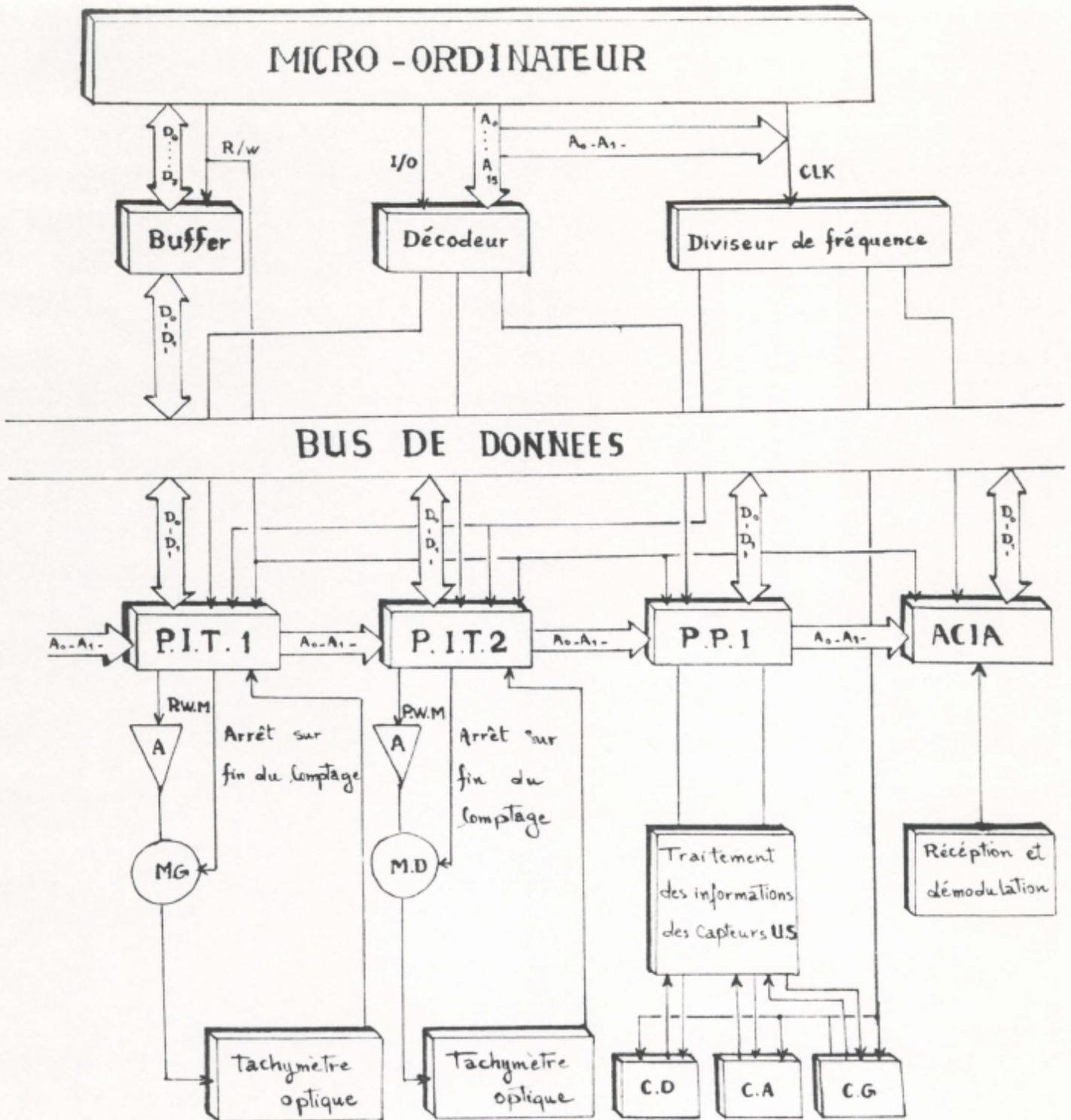


FIGURE III-5:
SYNOPTIQUE DE LA CARTE DE GESTION DE LA MOTRICITE ET DES CAPTEURS

des informations issues de ces capteurs. En fonction de toutes ces informations, l'ordinateur génère la vitesse adéquate, et le sens de rotation adéquat des moteurs aux pilotes (P.I.T.1 et P.I.T.2).

La communication avec le micro-ordinateur centrale à poste fixe, est rendue possible par l'adjonction d'un circuit de communication série programmable (A.C.I.A), interfacé au micro-ordinateur mobile. La ligne de communication hertzienne établie est unidirectionnelle; La section qui suit traite le détail.

4 LIGNE DE COMMUNICATION HOMME-MACHINE

Généralement, les robots mobiles sont équipés de deux micro-ordinateurs, l'un à bord, son rôle est d'assurer la gestion et le contrôle des actionneurs et des éléments perceptuels disposés sur le robot, en fonction du but à atteindre. L'autre à poste fixe, contient le modèle de connaissance du robot, et le planificateur de trajectoires; celui-ci génère une liste de buts partiels, à travers lesquels doit transiter le robot, pour atteindre le but final sans collision. Pour le transport de la liste au micro-ordinateur qui se trouve à bord du robot, nous avons établi une ligne de communication hertzienne fonctionnant en U.H.F canal 24.

En effet, pour être acheminée, toute information doit être mise sous forme de symboles. La signification précise de ces symboles n'est qu'une pure affaire de convention, entre l'émetteur du message et le destinataire.

4 - 1 CODAGE DES INFORMATIONS

Il s'agit de ramener les informations complexes que nous voulons échanger, à un ensemble d'informations binaires élémentaires, à l'aide de techniques de codage. Les informations à transmettre et donc à coder sont dans la pratique des éléments appelés caractères. L'ensemble $\{C_1, \dots, C_n\}$ des caractères à transmettre constitue un

alphabet. La représentation binaire $\{d_i^1, \dots, d_i^n\}$ où d_i^k prend la valeur 0 ou 1, d'un caractère C_i est appelée un mot de code, l'ensemble de ces mots de code est appelé un code. on utilise généralement pour l'échange d'informations entre micro-ordinateurs, le code ASCII.

4 - 2 TRANSMISSION EN LIGNE

La télécommunication numérique par ondes électromagnétiques, nécessite automatiquement un support d'informations série, particulièrement le port série RS232C lorsqu'il s'agit d'échange d'informations entre micro-ordinateurs. Nous utilisons un certain nombre de règles, concernant la transmission des caractères sur le support RS232C:

- La transmission des bits s'effectue par ordre de poids croissant.

- Un bit supplémentaire de parité pour la détection des erreurs.

- Les caractères sont précédés par un élément de départ "START" de même durée d'un bit, et suivis d'un élément d'arrêt "STOP" de durée double (sauf accord contraire). Ceci permet d'identifier les caractères dans le mode de transmission asynchrone, où l'instant d'apparition des caractères est aléatoire, on utilise généralement ce mode pour la communication hertzienne entre micro-ordinateurs.

4 - 3 ETABLISSEMENT DE LA LIGNE OPERATEUR-ROMoSA.I

La ligne de communication que nous avons établi, permet de relier le robot RoMoSA.I, à un opérateur humain par ondes électromagnétiques, sur un rayon d'environ 100 m. Pour plus de protection, le signal a été doublement modulé à la sortie du port série RS232C du micro-ordinateur émetteur de l'opérateur. Une modulation discrète par sauts de fréquence F.S.K ($f_0 = 10\text{kHz}$, $f_1 = 500\text{kHz}$) après la mise à niveau du signal à la sortie du port, suivie d'une modulation d'amplitude A.M, dans la bande U.H.F canal 24, soit une fréquence 494MHz. Le récepteur est équipé, d'un tuner U.H.F calé sur le canal 24, d'un détecteur d'enveloppe pour la

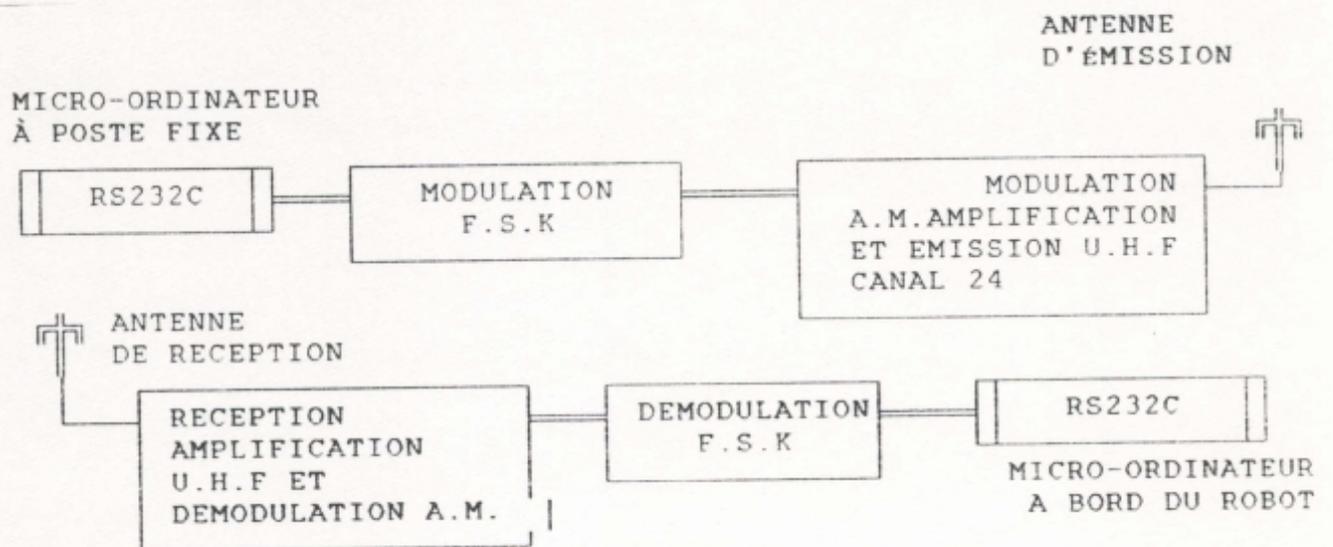


FIGURE III-6:
PRINCIPE DE LALIAISON HOMME-MACHINE

démodulation A.M, et d'un démodulateur F.S.K avant la remise à niveau du signal pour être reconnu par le port série RS232C du micro-ordinateur récepteur à bord du robot.

5 CONCLUSION

Nous avons situé dans ce chapitre, le contexte de nos travaux dans le cadre de la mise au point du support expérimental **RoMoSA.I**, qui va servir de test des algorithmes de planification de trajectoires, et d'évitement de collisions, développés dans le chapitre qui suit.

L'architecture présentée, se structure en quatre niveaux hiérarchiques; La gestion et le traitement des niveaux supervision et planification, sont assurés par un opérateur humain (générateur de tâches), et un micro-ordinateur à poste fixe. Les niveaux contrôle et exécution, englobent toutes les fonctionnalités d'interaction avec le monde réel, la gestion et le traitement de ces niveaux sont assurés par un micro-ordinateur à bord du robot.

Nous avons présenté également dans ce chapitre, les caractéristiques mécaniques et électroniques du robot **RoMoSA.I**, ainsi que la ligne de communication hertzienne dans la bande U.H.F, reliant le niveau planification au niveau contrôle.

CHAPITRE IV

DÉVELOPPEMENT D'UN PLANIFICATEUR RÉACTIF POUR LA NAVIGATION D'UN ROBOT MOBILE AUTONOME

1 INTRODUCTION

Ces dernières années, supportées par l'état d'avancement de la technologie des ordinateurs, plusieurs recherches ont été faites dans le cadre des robots mobiles autonomes, articulées autour de deux axes principaux, à savoir la navigation et le guidage. La navigation est l'action de se diriger vers un point donné, elle résout en général les problèmes de décision, afin que le mobile puisse suivre le bon chemin ou la bonne route à travers une population d'obstacles, dans un terrain qui peut être complètement connu, et ceci nécessite l'utilisation des techniques de planification de trajectoires [BRO 83] [LOZ 79]. Ces techniques ne sont pas directement utilisables dans le cas où l'espace est partiellement connu ou totalement inconnu, l'environnement dans ces cas peut contenir des obstacles de différents types; Le problème est de trouver un chemin, qui mène en sécurité le robot de la source à la destination. Des recherches intensives ont traité les problèmes d'évitement de collisions des robots mobiles avec des obstacles imprévus fixes [KHA 85] [BOR 89] [BOR 91a], ou en mouvements [GRI 90] [BUR 92]. Ceci implique généralement l'existence d'un système de mise à jour du réseau d'emplacement, pour servir à la planification globale de l'itinéraire durant les tâches qui suivent. L'exploration de l'espace, et les problèmes de décision augmentent la complexité de ce système; CROWLEY [CRO 85] [CRO 89], BORENSTEIN et KOREN [BOR 91b] [BOR 88], KUC et VIARD [KUC 91] ont essayé de résoudre les problèmes d'exploration, et la mise à jour du modèle de connaissances, en traitant des informations incertaines des capteurs ultrasonores. LYENGAR et RAO [LYE 86], et récemment FOUX et AL [FOU 93], ont développé des approches géométriques pour le traitement de ce système, basées sur des capteurs beaucoup plus précis (LASER). quant à l'approche de ZELINSKY [ZEL 92] développée autour de la méthode de partitionnement de l'espace par "QUADTREE", est entièrement basée sur les

informations des capteurs tactiles, la mise à jour du modèle se fait au fur et à mesure de l'exécution des tâches.

On a traité dans ce chapitre le cas où l'environnement est partiellement connu, et le robot est ordonné à atteindre son objectif sans collision. Ce traitement est développé autour de deux approches, l'une géométrique permettant la planification des trajectoires au but final, le chemin global obtenu est optimal dans le sens "distance euclidienne" (MINIMUM LENGTH PATHS). L'autre analytique basée sur le principe du maximum de **PONTRYAGAIN (P.M.P) [PON 86]**, et les instants de passage de l'accélération, de la limite supérieure à la limite inférieure et inversement; permettant la planification des trajectoires optimales dans le sens "temps optimal" entre les buts partiels. Ces approches, permettent une réduction considérable du temps de traitement et d'exécution, et l'évitement de collision avec les obstacles imprévus tout en guidant le robot vers l'objectif, la trajectoire dans ce cas perd son optimalité. Le système de perception a été développé autour d'une stratégie de scrutation des ultrasons, garantissant la détection de l'intrusion des obstacles imprévus, dans la zone dite de sécurité du robot.

2 REPRÉSENTATION DU MODELE DE CONNAISSANCE

Modéliser un domaine d'évolution d'un Robot mobile, revient à définir l'espace libre et l'espace occupé dans la mémoire du système du mobile, cette opération est effectuée, soit avant toute tâche (mode statique), et dans ce mode l'environnement est à priori connu ou partiellement connu par le Robot, soit au cours de la tâche (mode dynamique), et dans ce mode le Robot découvre son environnement au fur et à mesure de son évolution; et c'est dans le mode statique que nous visons le développement d'un modèle de l'environnement en s'inspirant de la méthode d'expansion des obstacles [LOZ 79].

Afin de réduire la complexité des calculs du planificateur, nous avons approximé tous les obstacles de n'importe quelle forme, à la forme rectangulaire, pour être plus clair l'environnement représenté à la **FIGURE IV-1-a** est pris comme exemple, où les obstacles considérés sont supposés fixes, parmi ces obstacles ceux qui ont une forme non rectangulaire sont ramenés à la forme rectangulaire. La **FIGURE IV-1-b** représente l'effet de cette première approximation, pour une deuxième approximation, la forme ainsi obtenue des obstacles, est étendue d'une distance dépendant directement des dimensions du Robot $R + R/4$, où R est le rayon du Robot considéré comme étant la portée du système de perception à partir du centre du Robot, représentée à la **FIGURE IV-3-b**; La **FIGURE IV-1-c** montre clairement l'effet de cette deuxième approximation, enfin, le planificateur retient de l'environnement le résultat de ces approximations représenté à la **FIGURE IV-1-d**.

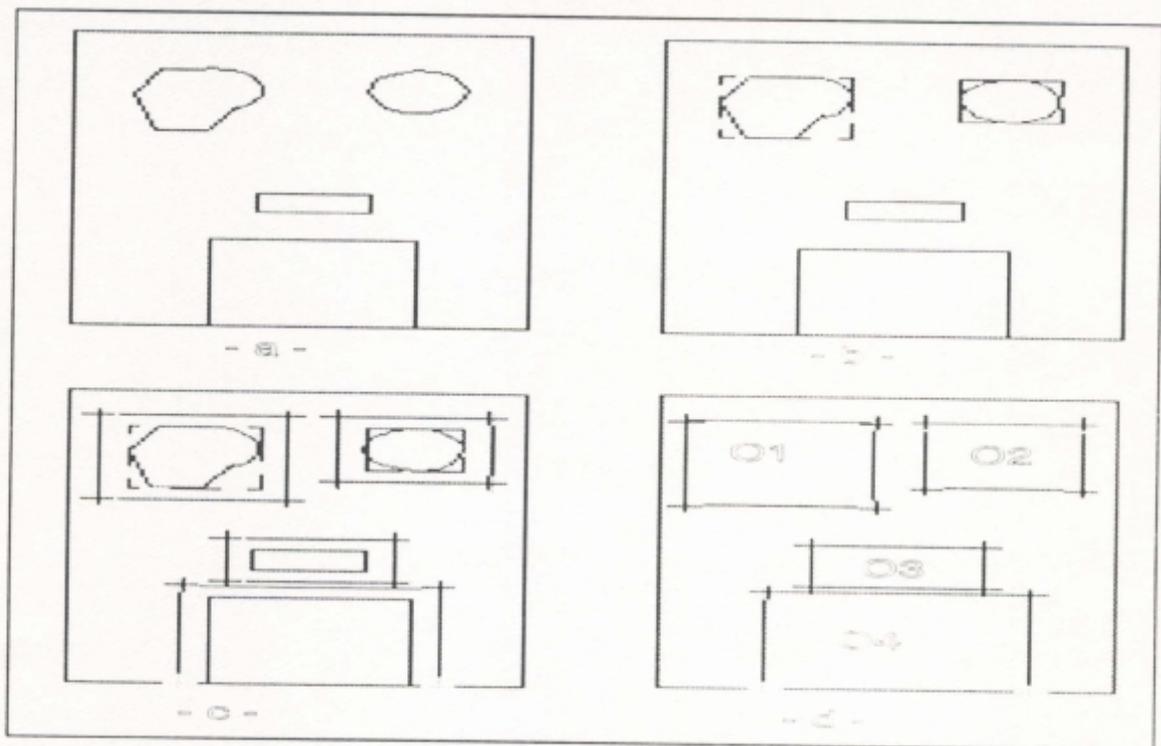


FIGURE IV-1: Modélisation de l'environnement
a: Environnement test, b: Effet de la première approximation,
c: Effet de l'expansion, d: Environnement retenu par le
p l a n i f i c a t e u r

3 PLANIFICATION DES TRAJECTOIRES

Un itinéraire global est planifié sous la forme d'une suite d'emplacements, qui conduit le Robot, de sa position actuelle à la destination donnée, cet itinéraire peut être le chemin le plus court du réseau pour atteindre le but final, l'algorithme de planification de trajectoire que nous avons développé, se place dans le cadre des procédures de recherche exhaustive, où l'optimalité de l'itinéraire est assurée au détriment du temps d'exécution du planificateur; Le réseau d'emplacements considéré est le modèle final acquis après les deux approximations **FIGURE IV-1-d**; Le Robot est réduit à un point.

Nous avons organisé le planificateur en quatre (04) procédures:

- * procédure "**VERTEX_SELECTION**" limitation des sommets des obstacles fixes.

- * procédure "**NEAREST_OBSTACLE**" calcul du sommet le plus proche.

- * procédure "**LINE SD OBSTACLE_TEST**" test d'existence des obstacles sur la ligne source-destination.

- * procédure "**STOP POINT_DETECTION**" calcul du but partiel.

1°) Procédure de limitation des sommets des obstacles fixes "VERTEX_SELECTION":

Une fois les coordonnées du Robot et de l'objectif connues, le planificateur procède à la sélection des sommets des obstacles fixes, se trouvant au voisinage immédiat de la ligne source-destination, et ceci dans le seul but de réduire le temps de calcul.

En effet, soit le repère orthonormé R_r (Repère-Robot), obtenu par changement d'origine du repère R_e (Repère-Environnement) à la position actuelle du Robot (Source), et:

X_r, Y_r : coordonnées du Robot dans le repère R_e

X_s, Y_s : coordonnées des sommets des rectangles dans

le repère Rr

Xd, Yd: coordonnées de la destination dans le repère Rr

Les sommets se trouvant dans le même cadran que l'objectif (Destination), où la condition ci-dessous est vrai, sont retenus, les autres sont rejetés.

IF (SIGN(Xs) = SIGN(Xd)) AND (SIGN(Ys) = SIGN(Yd)) THEN keep the vertex

La **FIGURE IV-2-a** montre clairement les étapes de cette procédure, les sommets de l'espace hachuré sont rejetés.

2°) Procédure de calcul du sommet le plus proche "NEAREST OBSTACLE"

Dans cette procédure, le planificateur calcule la distance séparant le centre du Robot (l'origine du repère Rr) à chaque sommet S_{ij} retenu par la procédure "VERTEX_SELECTION", cette distance est évaluée par l'équation ci-dessous:

$$D_{ij} = (X_{S_{ij}}^2 + Y_{S_{ij}}^2)^{1/2}$$

avec: $(X_{S_{ij}}, Y_{S_{ij}})$ les coordonnées du jème sommet appartenant au ième obstacle.

et D_{ij} : distance entre le centre du Robot et le sommet S_{ij} .

L'obstacle le plus proche au Robot est déduit à partir du sommet ayant la distance la plus courte:

$$D_{min} = \text{Min}(D_{ij})$$

3°) Procédure test: d'existence des obstacles sur la ligne Source-Destination "LINE SD OBSTACLE_TEST":

A ce niveau, le planificateur connaît éventuellement la position de l'obstacle le plus proche au Robot, il procède ensuite à la vérification de l'existence de cet obstacle sur la ligne Source-Destination. Pour ce faire, nous avons organisé cette procédure comme suit:

Soient (X_{si}, Y_{si}) : coordonnées du i ème sommet de l'obstacle le proche ($i=1,2,3,4$) dans le repère R_r .

α : L'angle que fait l'axe des ordonnées du repère R_r avec la droite Source-Destination.

β_i : L'angle que fait l'axe des ordonnées du repère R_r avec la droite Source- i ème sommet.

Le sens contraire des aiguilles d'une montre est pris comme sens positif des angles.

$$\alpha = \text{Arctg}(X_d/Y_d)$$

$$\beta_i = \text{Arctg}(X_{si}/Y_{si})$$

La ligne Source-Destination est interrompue par l'obstacle si la condition ci-dessous est vraie:

IF $(\beta_1 < \alpha < \beta_3)$ **OR** $(\beta_2 < \alpha < \beta_4)$ **THEN Obstacle Intersect line SD**

β_1 et β_2 sont respectivement les angles correspondant aux coins supérieurs gauche et droit.

β_3 et β_4 les angles correspondant aux coins inférieurs droit et gauche respectivement.

La **FIGURE IV-2-b** illustre le principe de cette procédure.

4°) Procédure de calcul du but partiel "STOP POINT_DETECTION"

Si l'obstacle interromp la ligne Source-Destination, il est évident que le chemin le plus court à l'objectif, passe par l'un des quatre sommets de l'obstacle, pour le déterminer, nous avons subdivisé la procédure en deux étapes:

- * Etape classification des sommets.
- * Etape détermination du sommet optimal.

a) Classification des sommets:

Dans cette étape les quatre sommets de l'obstacle sont ordonnés en deux classes:

* classe1 les sommets se trouvent au-dessus de la ligne Source-Destination.

* classe2 les sommets se trouvent au-dessous de la ligne Source-Destination.

Soit: $a_i = \alpha - \beta_i$

Si $a_i > 0$ alors le i ème sommet appartient à la classe1, sinon il appartiendra à la classe2. ($i=1,2,3,4$)

b) Détermination du sommet optimal:

Soit L_i : distance du i ème sommet à la droite Source-Destination obtenue par projection orthogonale du sommet sur cette droite.

D_i : distance du i ème sommet au centre du Robot.

L_{ij} : distance L_i du sommet de la classe j . ($j=1,2$)

$$D_i = (X^2s_i + Y^2s_i)^{1/2}$$

$$L_i = D_i \cdot \sin(a_i)$$

Le sommet recherché est celui dont la distance L_i vérifie l'équation:

$$L_i = L_{\min}$$

où $L_{\min} = \text{MIN}(\text{MAX}(L_{i1}), \text{MAX}(L_{i2}))$

La **FIGURE IV-2-c** représente les étapes de cette procédure.

Le sommet ainsi obtenu est mis dans la liste des buts partiels où doit transiter le Robot pour atteindre le but final. Chaque but partiel est pris comme source pour le calcul du prochain but partiel, et les quatre procédures se répètent jusqu'au but final. La trajectoire finale obtenue est représentée à la **FIGURE IV-2-d**.

REMARQUES:

* Si l'obstacle le plus proche ne se trouve pas sur la ligne Source-Destination ces quatre sommets sont rejetés.

* Si l'objectif se trouve à l'intérieur de la zone étendue le Robot transite à l'un des sommets le plus proche et essaye de l'atteindre à partir de ce sommet.

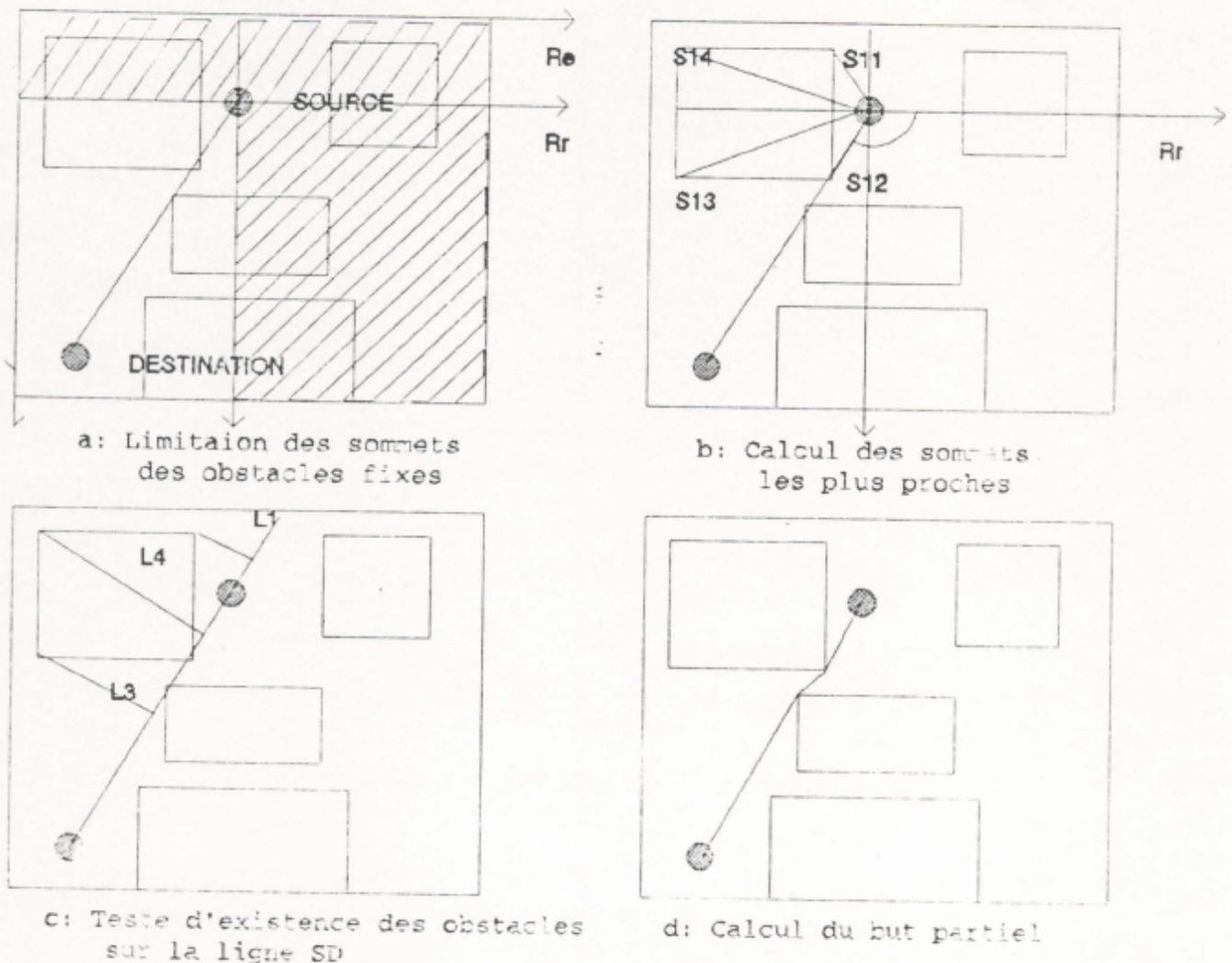


FIGURE IV-2: Planification des trajectoires

a: Limitation des sommets des obstacles, b: Calcul des sommets les plus proches, c: Test d'existence des obstacles sur la ligne S-D, d: Calcul du but partiel.

4 PERCEPTION DE L'ENVIRONNEMENT

4 - 1 INTRODUCTION

Les capteurs acoustiques ont fait preuve de leurs performances dans de nombreuses applications, notamment le guidage et la navigation des robots mobiles, où le système de mesure de temps de vol (T.O.F, time of flight) informe le mobile sur l'existence ou l'absence des obstacles dans sa trajectoire;

Parfois, ces informations sont erronées, et des objets

peuvent passer inaperçus, pour cette raison, beaucoup de chercheurs ont inclus à ce système des capteurs additionnels telles que les capteurs tactiles et la vision.

Nous décrivons maintenant, un modèle physique pour les SONARS, et nous essayons de l'appliquer à la navigation du véhicule autonome dans un environnement qui n'est pas structuré.

le problème de base est illustré à la **FIGURE IV-3**, où le véhicule autonome, contenu à l'intérieur d' un disque de rayon R essaye de faire une translation de D sans collision.

Voyons maintenant lorsque la même situation se présente a un aveugle, il scrute d'abord son environnement avec sa cane de droite à gauche et de gauche à droite, pour voir si le chemin est libre, puis il fait un pas, la grandeur de ce pas dépend de la longueur de sa cane. Dans notre cas l'environnement est interrogé par un SONAR, et le mouvement du véhicule autonome se fait par pas de D ; contrairement à la cane, les SONARS ont des difficultés pour détecter des surfaces obliques, et les bords des obstacles.

Pour résoudre ces problèmes, nous avons développé une stratégie de scrutation, en agissant sur la densité de scrutation $\delta\beta$ et l'étendu maximal β_{max} qui peut assurer une bonne détection de tous les obstacles se trouvant sur le chemin.

La direction du mouvement est définie par $\beta = 0$. pour les SONARS non rotatifs, le paramètre $\delta\beta$ détermine le nombre des capteurs de l'anneau ultrasonore, et β_{max} l'étendu de l'anneau.

IV - 4 - 2 LOIS PHYSIQUE DES SONARS

Le principe de mesure de temps de vol, l'ouverture du faisceau ultrasonore, la réflexion et la diffraction des ultrasons, tels sont les principaux points que nous allons résumer dans ce qui suit.

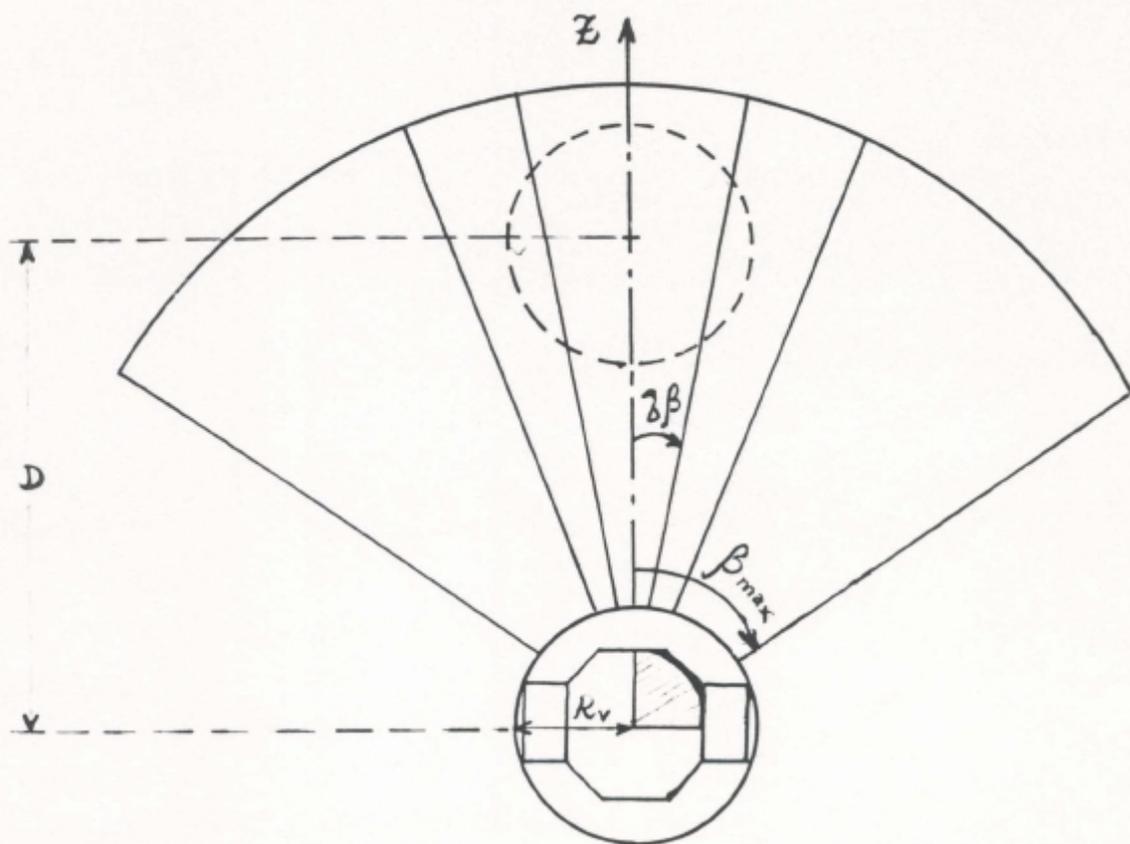


FIGURE IV-3:
 LE ROBOT APPROXIME PAR UN DISQUE DE RAYON R ESSAYE DE FAIRE
 UNE TRANSLATION DE D

a) Mesure du temps de vol (T.O.F)

Généralement, les SONARS utilise un seul transducteur émetteur-récepteur, fonctionnant alternativement dans un mode puis l'autre. La zone aveugle (distance minimal de détection Z_{min}) est réduite au stricte minimum si un transducteur est réservé à chaque mode (émetteur ou récepteur) séparément.

La **FIGURE IV-4** montre l'allure du signal reçu juste après parle transducteur récepteur, où le seuil de détection τ est inclu, pour supprimer les échos indésirables d'origine électronique ou acoustique.

Une fois que le premier écho dépassant le seuil apparait, le système de mesure de temps de vol évalue la distance Z_0 à partir de la formule:

$$Z_0 = v \cdot t_0 / 2$$

où t_0 est le temps séparant l'apparition de l'écho de l'instant de son émission.

et v la vitesse du sons dans l'air soit environ 343 m/s .

Dans notre cas, peu importe la connaissance exacte de la distance aux obstacles, ce qui importe c'est la détection de ces obstacles.

Le signal une fois amplifié, est appliqué à un détecteur de seuil, ce qui nous ramène donc à ne s'intéresser qu'à l'amplitude maximale du signal reçu (A).

Nous montrerons par la suite que A est fonction de θ , l'angle que fait le rayon ultrasonore avec la normale au transducteur, et Z la distance parcourue par ce rayon.

b) Analyse de l'amplitude de l'écho

Pour développer une stratégie de scrutation, il est très important d'avoir une description analytique des échos en fonction de θ et Z .

La fonction reliant l'amplitude de l'écho à θ et Z est donnée par:

$$A(\theta) = A_0 \cdot \text{EXP}(-4 \cdot \theta^2 / \theta_0^2)$$

$$A_0 = 2 \cdot \Psi \cdot \text{EXP}(-2 \cdot \alpha \cdot z)$$

Ψ : l'amplitude du signal à l'émission

$2 \cdot \theta_0$: l'ouverture du faisceau avec:

z : la distance parcourue par le rayon ultrasonore

α : le coefficient d'atténuation

$$\theta_0 = \arcsin(0,61 \cdot \lambda / a)$$

λ : étant la longueur d'onde du signal acoustique

a : est le diamètre du transducteur

Les transducteurs qu'on a utilisé ont une fréquence de 40 khz, une longueur d'onde $\lambda = 8,6$ mm, et une ouverture $\theta_0 = 15^\circ$. Les SONARS conventionnels à un seul transducteur émetteur-récepteur, sont efficaces au delà d'une certaine distance Z_{\min} appelée zone aveugle, déterminée à partir du temps de passage du mode émetteur au mode récepteur; dans le cas où l'émetteur et le récepteur sont séparés, cette zone est réduite au stricte minimum et c'est ce dernier cas qui nous a intéressé.

La distance maximale Z_{\max} est obtenue dans le cas où $A_0 = \tau$ (τ est le seuil de détection):

$$G \cdot \Psi \cdot \text{EXP}(-2 \cdot \alpha \cdot (Z_{\max} \cdot 2)) = \tau$$

donc on a:

$$Z_{\max} = \text{LOG}((G \cdot \Psi) / \tau) / (4 \cdot \alpha)$$

En pratique on a pris $\Psi = 5\text{v}$, $\tau = 1\text{v}$ et nous avons ajusté le gain G de telle façon à obtenir $Z_{\max} = 60$ cm.

c) L'environnement

Le modèle de l'environnement que nous avons considéré constitue une bonne approximation de la plupart des environnements réels.

Il est formé de surfaces lisses peu accidentées, et le balayage des SONARS se fait dans le plan horizontal.

A la rencontre d'un changement d'impédance acoustique¹ le signal se disperse et deux types d'échos peuvent apparaître:

Les échos réfléchis provoqués par des surfaces lisses, telles que les murs, les portes, et l'angle droit des coins; et les échos diffractés produits par les discontinuités des surfaces, et dans cette catégorie nous incluons les bords des surfaces et l'ouverture des portes.

Nous analysons dans ce qui suit chaque type d'écho séparément.

* détection des échos réfléchis

La **FIGURE IV-5** montre comment le signal acoustique est réfléchi à la rencontre d'une surface oblique; ce signal produit un écho détectable que lorsque $|\theta| < \theta_0$, sachant que l'angle d'incidence égal à l'angle de réflexion.

Soit $A_s(\theta)$ l'amplitude de l'écho reçu;

$$A_s(\theta) = A_o \cdot \text{EXP}(-4 \cdot \theta^2 / \theta_0^2)$$

Il faut que $A_s(\theta) \geq \tau$ pour $|\theta| \leq \theta_0$, pour que l'écho soit exploitable.

Les coins droits internes obéissent aussi à ces lois.

* détection des échos diffractés

Contrairement aux échos réfléchis par une surface, les échos diffractés par les bords se propagent dans toutes les directions, formant des ondes cylindriques émanant de ce bord voir **FIGURE IV-6**.

LIZUKA [LIZ 85] a montré en 1985 que l'amplitude des

¹ l'impédance acoustique est le produit de la densité du milieu par la vitesse du son dans ce milieu, dans le cas de notre robot mobile, le milieu de propagation est l'air, ayant une impédance acoustique très faible et tous les corps solides ont une impédance beaucoup plus importante

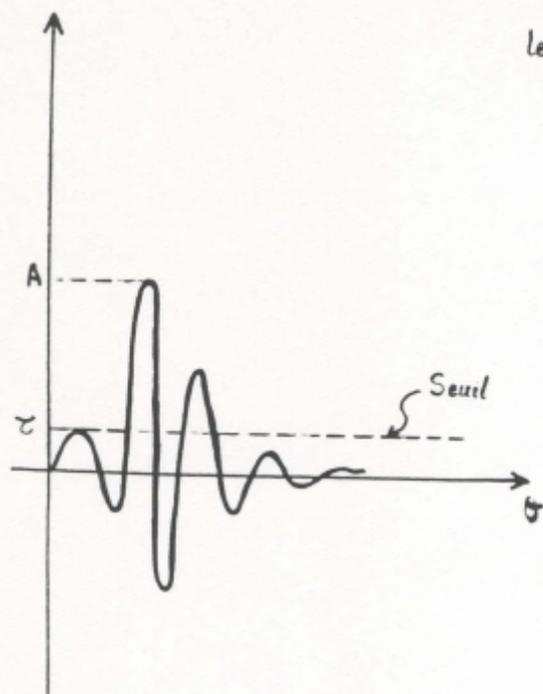


FIGURE IV-4:
ALLURE DU SIGNAL RECU PAR
LE TRANSDUCTEUR RECEPTEUR

On suppose que l'émetteur et
le récepteur sont confondus

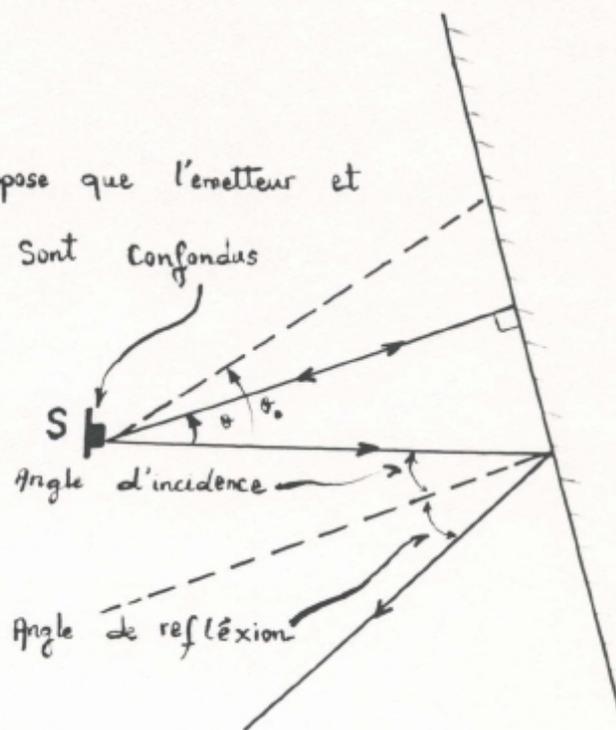
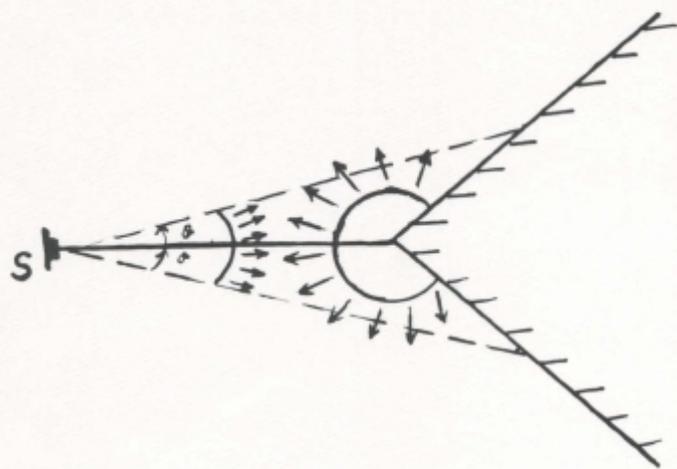
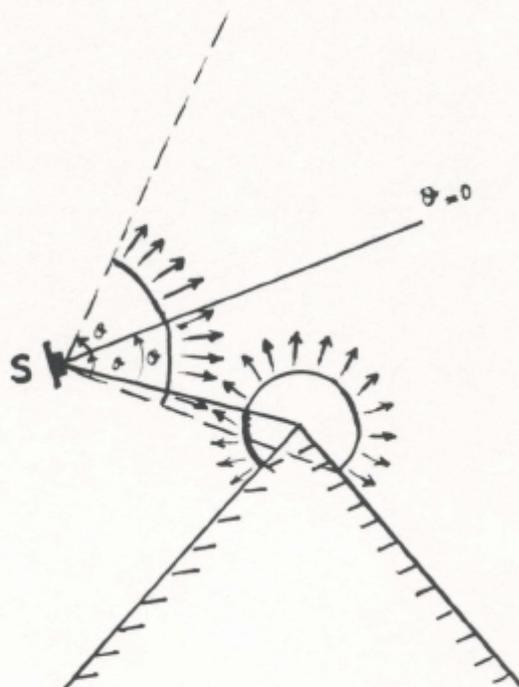


FIGURE IV-5:
REFLEXION DU SIGNAL ACOUSTIQUE
PAR UNE SURFACE OBLIQUE



Les deux surfaces sont visibles



Une seule surface est visible

FIGURE IV-6:
DIFFRACTION
DES ECHOS PAR LES BORDS

échos décroît de $Z^{-\frac{1}{2}}$ à partir du bord.

Soit $Ae(\theta, Z)$ l'amplitude d'un écho diffracté par un bord en fonction de θ et la distance au bord Z .

KUC et VIARD [KUC 91] ont défini en 1991, un coefficient ϵ_0 de dispersion d'un bord localisé à Z_0 comme suit:

$$\epsilon_0 = Ae(0, Z_0) / As(0) = Ae(0, Z_0) / A_0$$

est mesuré pour une distance $Z = Z_0$ et $Ae(0, Z)$ peut être déduite en prenant les pertes $Z^{-\frac{1}{2}}$ en considération:

$$Ae(\theta, Z) = A_0 \cdot \epsilon_0 \cdot (Z_0/Z)^{\frac{1}{2}} \cdot \text{EXP}(-4 \cdot \theta^2 / \theta_0^2)$$

$$Ae(0, Z) = Ae(0, Z_0) \cdot (Z_0/Z)^{\frac{1}{2}}$$

Expérimentalement on a obtenu pour un bord de 90° localisé à 10 cm, une valeur de $\epsilon_0 = 0,05$, en générale ϵ_0 est fonction de la fréquence du transducteur et de l'angle du bord, et durant notre expérience on a constaté, que ϵ_0 ne dépend pas de l'orientation du bord. Et le bord où une surface est visible, produit un écho dont l'amplitude est approximativement la moitié de celle produite par l'écho d'un bord où les deux surfaces sont visibles.

La **FIGURE IV-7** montre l'allure de l'écho diffracté en fonction de θ , en pratique cette courbe est échantillonnée par pas de $\delta\beta$; il est évident que pour que l'écho soit détectable, il faut que $\delta\beta$ soit assez petite pour qu'au moins un écho dépasse le seuil; pour réduire $\delta\beta$, au lieu de faire tourner le support contenant les sondes par un moteur pas à pas, comme dans le cas de KUC et VIARD; on a utilisé un moteur DC asservie en position ce qui permet de réduire $\delta\beta$.

Avec cette densité de scrutation on peut détecter un bord à une distance $Z < Z_{\text{libre}}$.

Z_{libre} est la distance à partir de laquelle un bord peut passer inaperçu, autrement dit:

$$Ae(\theta, Z_{\text{libre}}) = \tau$$

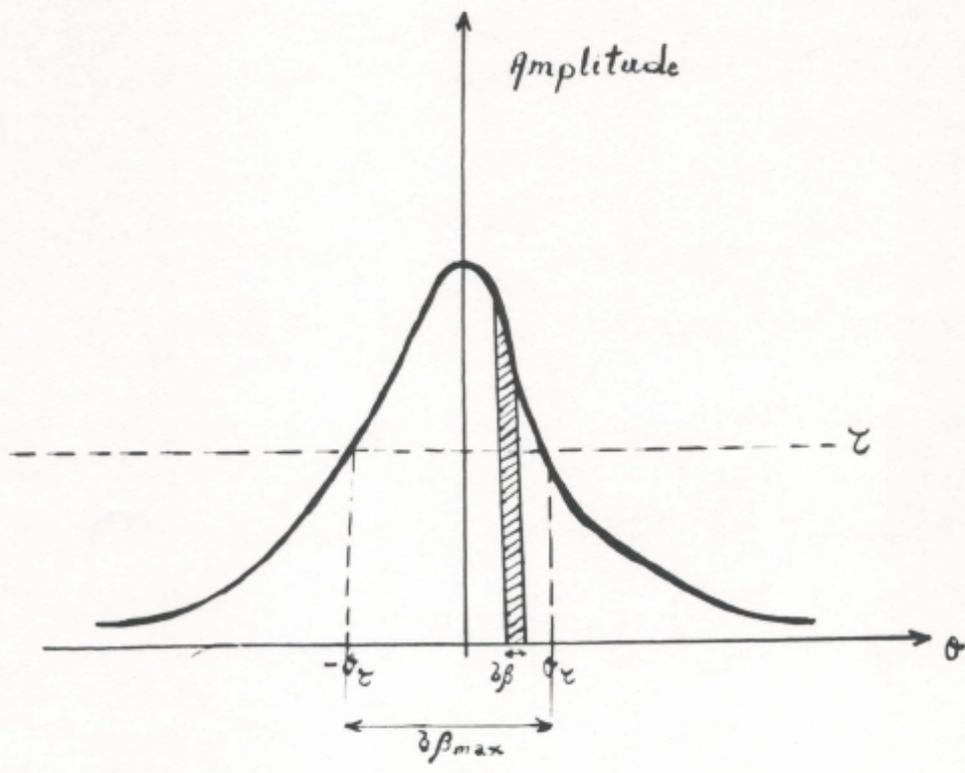


FIGURE IV-7:
 ALLURE DE L'AMPLITUDE DE L'ECHO DIFFRACTE EN FONCTION DE θ

$$Ae(\theta, Z_{\text{libre}}) = A_0 \cdot \epsilon_0 \cdot (Z_0 / Z_{\text{libre}})^2 \cdot \text{EXP}(-4 \cdot \theta^2 / \theta_0^2) = \tau$$

$Z_{\text{libre max}}$ correspond à $\theta = 0$

donc on a: $Z_{\text{libre max}} = (A_0 \cdot \epsilon_0 / \tau)^2 \cdot Z_0$

C'est la distance maximale où un bord peut être détecté

4 - 3 ACQUISITION ET INTERPRETATION DES DONNEES ISSUES DU SONAR

Ayant vu les conditions sous lesquelles une surface ou un bord peut être détecté, nous essayons maintenant de développer la stratégie de scrutation.

a) Choix du pas D

Le pas D est limité par la distance maximale de détection d'un bord. si nous choisissons $D > Z_{\text{libre}} - R$ il est évident qu'une collision peut avoir lieu avec un bord indétectable, il faut que D soit: $D < Z_{\text{libre}} - R$

b) Interprétation des données

Soit $Z_{\text{tof}}(\beta)$ la distance mesurée dans la direction β ; donc si $Z_{\text{tof}}(0) < D + R$ un obstacle est détecté dans la direction du mouvement et le robot ne se déplace pas.

Considérons une surface oblique qui peut donner naissance à une collision durant la translation, le cas limite est représenté à la **FIGURE IV-8**, où la surface est localisée à la tangente du disque après un déplacement de D .

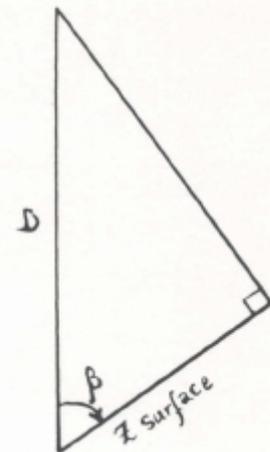
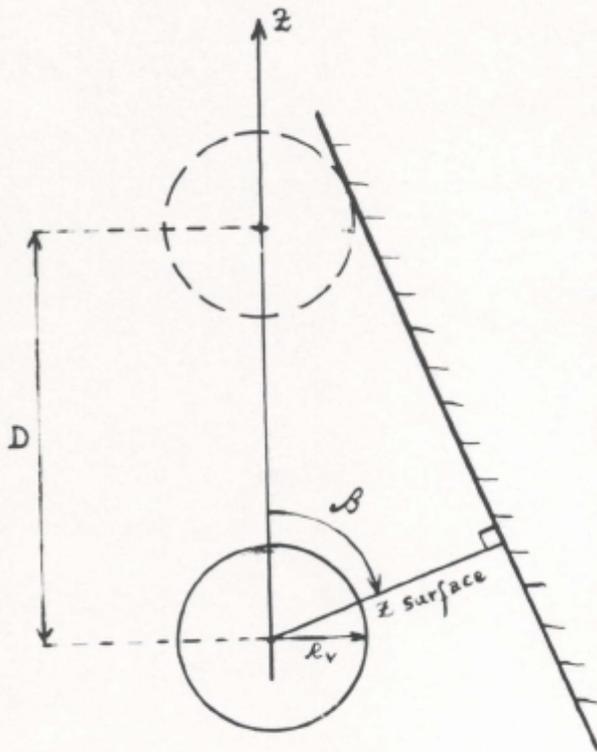
Soit Z_{surface} la distance mesurée à une incidence normale sur la surface; pour simplifier l'analyse on a réduit R à 0

FIGURE IV-9

$$Z_{\text{surface}} = D \cdot \text{COS}(\beta)$$

En reinjectant le rayon R :

$$Z_{\text{tof}}(\beta) \geq D \cdot \text{COS}(\beta) + R \quad (*)$$



$$Z_{\text{surface}} = D \cos \beta$$

FIGURE IV-9:
LE RAYON R EST APPROXIME A ZER

FIGURE IV-8:
SURFACE OBLIQUE QUI PEUT DONNER NAISSANCE A UNE COLLISION
DURANT LA TRANSLATION

Jusqu'à présent, on n'a pas pris en considération l'ouverture du faisceau $2.\theta_0$. En introduisant l'effet de cette ouverture la formule (*) devient:

$$\begin{aligned}
 Z_{\text{tof}}(\beta, \theta_0) &\geq D.\text{COS}(\beta - \theta_0) + R && \text{si } \theta_0 < \beta < \beta_{\text{max}} \\
 &\geq D.\text{COS}(\beta) + R && \text{si } -\theta_0 < \beta < \theta_0 \\
 &\geq D.\text{COS}(\beta + \theta_0) + R && \text{si } -\beta_{\text{max}} < \beta < -\theta_0
 \end{aligned}$$

Une collision peut avoir lieu; si un écho provient de cette zone; ce qui empêche donc tout déplacement du robot.

c) Choix de β_{max}

La valeur de β_{max} est déduite, à partir de la possibilité du capteur de détecter tous les obstacles se trouvant sur le chemin, une surface inclinée légèrement sur le chemin du robot doit être détectée; dans ce cas il faut prendre: $\beta_{\text{max}} > 90^\circ - \theta_0$; or à la limite aux voisinages de θ_0 un bord peut passer inaperçu; pour le détecter il faut que: $\beta_{\text{max}} > 90^\circ - \delta\beta^*$

$$\text{ou } \delta\beta^* \leq 0,5.\theta_0.[-\text{LOG}(\tau.(R/Z_0)^2 / (A_0.\epsilon_0))]^{1/2}.$$

On a pris $\beta_{\text{max}} = 85^\circ$.

Avec $\delta\beta$, β_{max} et D ainsi obtenus le véhicule autonome peut se déplacer sans collision.

Dans cette étude nous avons considéré la hauteur des obstacles supérieure au plan de scrutation.

5 REACTION FACE AUX EVENEMENTS IMPREVUS

Dans cette section, nous allons développer la partie réactive du Robot face aux obstacles imprévus par le modèle; Pour faire face à ces événements, nous avons développé une stratégie de scrutation des capteurs ultrasonores, permettant au système de perception de détecter l'intrusion des obstacles dans la zone dite de sécurité **FIGURE IV-10-c**. Cette zone est partagée en trois secteurs, où chaque secteur est

Jusqu'à présent, on n'a pas pris en considération l'ouverture du faisceau $2.\theta_0$. En introduisant l'effet de cette ouverture la formule (*) devient:

$$\begin{aligned} Z_{\text{tof}}(\beta, \theta_0) &\geq D.\text{COS}(\beta - \theta_0) + R && \text{si } \theta_0 < \beta < \beta_{\text{max}} \\ &\geq D.\text{COS}(\beta) + R && \text{si } -\theta_0 < \beta < \theta_0 \\ &\geq D.\text{COS}(\beta + \theta_0) + R && \text{si } -\beta_{\text{max}} < \beta < -\theta_0 \end{aligned}$$

Une collision peut avoir lieu; si un écho provient de cette zone; ce qui empêche donc tout déplacement du robot.

c) Choix de β_{max}

La valeur de β_{max} est déduite, à partir de la possibilité du capteur de détecter tous les obstacles se trouvant sur le chemin, une surface inclinée légèrement sur le chemin du robot doit être détectée; dans ce cas il faut prendre: $\beta_{\text{max}} > 90^\circ - \theta_0$; or à la limite aux voisinages de θ_0 un bord peut passer inaperçu; pour le détecter il faut que: $\beta_{\text{max}} > 90^\circ - \delta\beta^*$

$$\text{ou } \delta\beta^* \leq 0,5.\theta_0.[-\text{LOG}(\tau.(R/Z_0)^2 / (A_0.\epsilon_0))]^{1/2}$$

On a pris $\beta_{\text{max}} = 85^\circ$.

Avec $\delta\beta$, β_{max} et D ainsi obtenus le véhicule autonome peut se déplacer sans collision.

Dans cette étude nous avons considéré la hauteur des obstacles supérieure au plan de scrutation.

5 REACTION FACE AUX EVENEMENTS IMPREVUS

Dans cette section, nous allons développer la partie réactive du Robot face aux obstacles imprévus par le modèle; Pour faire face à ces événements, nous avons développé une stratégie de scrutation des capteurs ultrasonores, permettant au système de perception de détecter l'intrusion des obstacles dans la zone dite de sécurité **FIGURE IV-10-c**. Cette zone est partagée en trois secteurs, où chaque secteur est

balayé par deux (02) sondes (une pour l'émission et l'autre pour la réception):

- secteur de la face avant SA
- secteur du coté droit SD
- et secteur du coté gauche SG

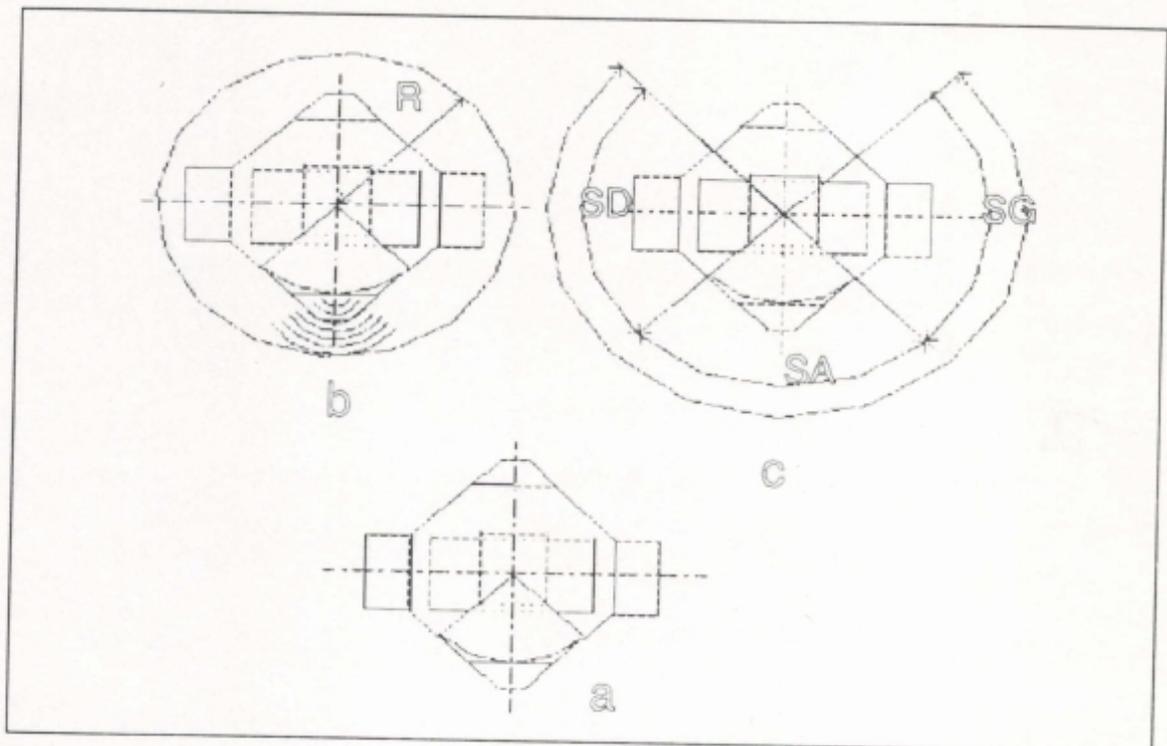
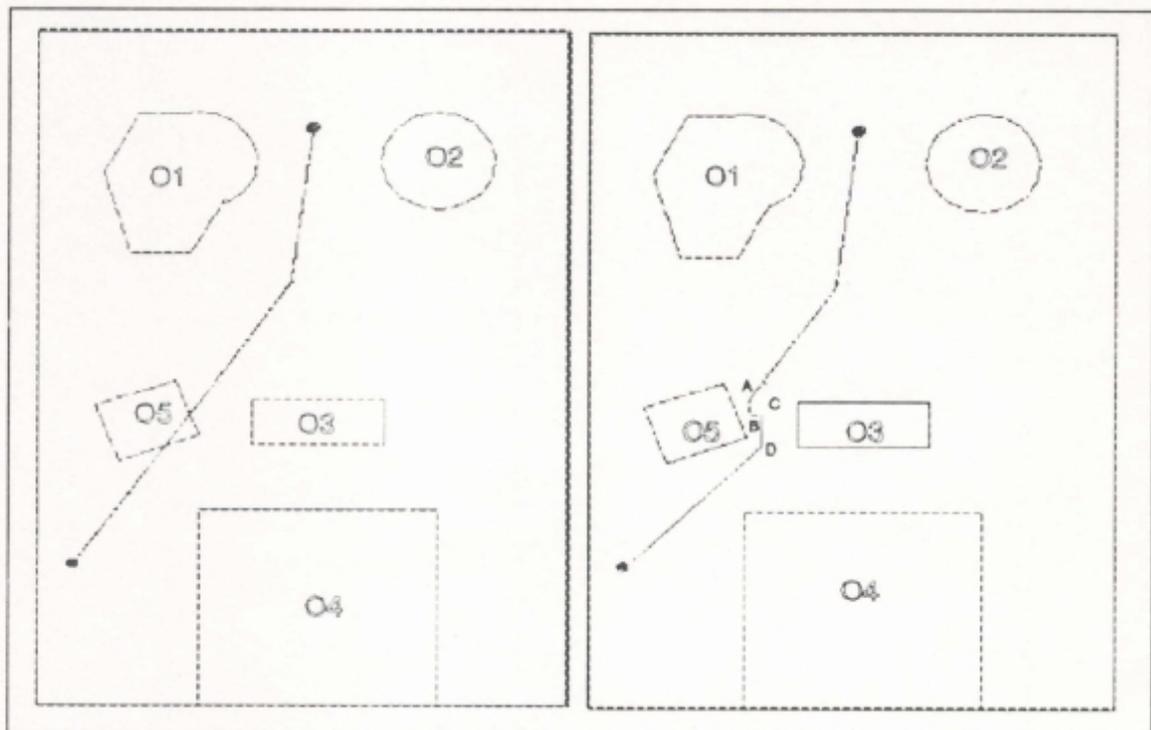


FIGURE IV-10: Représentation des secteurs de sécurité du Robot

Si les capteurs du secteur SA sont responsables de la détection de l'obstacle imprévu, une collision est possible, pour l'éviter le robot tourne à gauche ou à droite en prenant la direction de l'axe du repère R_r le plus proche à la destination. Dans le cas de la **FIGURE IV-11-a** le robot ignore la présence de l'obstacle O5; L'historique du voyage du robot de la source à la destination, est représenté à la **figure IV-11-b**; Arrivant en A le robot détecte la présence de l'obstacle O5, il essaye de prendre la direction de l'axe le plus proche à la destination (l'axe des abscisses) et incrémente le compteur d'obstacle CP ($CP=1$), en tournant à gauche dans la direction de l'axe des abscisses, le robot redétecte en B l'obstacle O5, il incrémente encore une fois

le compteur CP ($CP=2$), et tourne à gauche dans la direction de l'axe des ordonnées, les capteurs du secteur SD scrutent le côté droit pour s'assurer de la disparition de l'obstacle pendant le déplacement du robot, arrivant en C l'obstacle O5 disparaît du secteur SD, le compteur CP est décrémenté ($CP=1$), et le robot tourne à droite pour reprendre la direction de l'axe des abscisses; Les capteurs du secteur SD scrutent encore une fois le côté droit pendant le déplacement du robot, arrivant en D l'obstacle O5 disparaît du secteur SD, le compteur CP est décrémenté ($CP=0$), une fois le contenu de ce compteur est nul le planificateur replanifie la trajectoire à la destination, ainsi le robot atteint son objectif en sécurité.



a b
 FIGURE: IV-11: Evitement de l'obstacle imprévu (O5)

REMARQUES :

- * lorsque le CP est différent de zéro "0" le robot garde toujours le même sens de rotation à chaque incrémentation du CP, à la rencontre du mur ou d'un obstacle connu le robot tourne dans le sens contraire pour prendre le sens opposé de la direction en cours.

- * A chaque décrémentation du CP le robot tourne dans le sens contraire au sens pris pendant l'incrémentation du CP, lorsque le CP=0 le planificateur replanifie la trajectoire au but final.

6 CONTROLE D'EXECUTION ET LOCALISATION

6 - 1 INTRODUCTION

Les déplacements du robot **RoMoSA.I** sont assurés par deux roues motrices, contrôlées indépendamment; Ces moteurs sont gérés au moyen de deux modules:

1-Le module de locomotion: assure la localisation odométrique du Robot et l'asservissement de celui-ci sur une position consigne.

2-Le module Pilotage: assure la production des consignes qui sont utilisées par la locomotion pour asservir la position du Robot sur les trajectoires demandées; Ce module doit inclure les fonctions nécessaires pour effectuer des mouvements asservis du type évitement d'obstacles.

Dans ce chapitre, nous allons tout d'abord développer, d'une part une méthode d'asservissement en temps optimal des moteurs sur une position consigne θ_c , et d'autre part une méthode de modélisation de l'odométrie.

6 - 2 CONTROLE D'EXECUTION

6 - 2 - 1 Introduction

La mission entière du robot est exprimée en terme de séquences de positions, générées par le module pilotage, par où le robot doit transiter pour atteindre le but final; Notre objectif est de planifier des trajectoires en temps optimal entre ces positions, et contrôler l'exécution de ces trajectoires, en s'appuyant sur le principe du maximum de PONTRYAGAIN (P.M.P); Dans cette étude on ne tient pas compte des glissements des roues, et nous montrons formellement, que tous les mouvements en temps optimal du robot, nécessite un contrôle "bang-bang" des roues, l'accélération des roues est à chaque instant soit au maximum, soit au minimum de ces limites.

6 - 2 - 2 Les équations cinématiques des mouvements du robot.

Dans cette section, nous allons développer un modèle cinématique du robot, en supposant que les glissements des

roues sont négligeables, et que l'accélération des roues, peut instantanément passer de sa limite supérieure, à sa limite inférieure, et vice versa. La configuration du robot est décrite par; les coordonnées cartésiennes x et y du point milieu de l'axe reliant les deux roues motrices, et l'orientation θ par rapport à l'axe des abscisse x , d'un repère absolue relié à l'environnement (FIGURE IV-12).

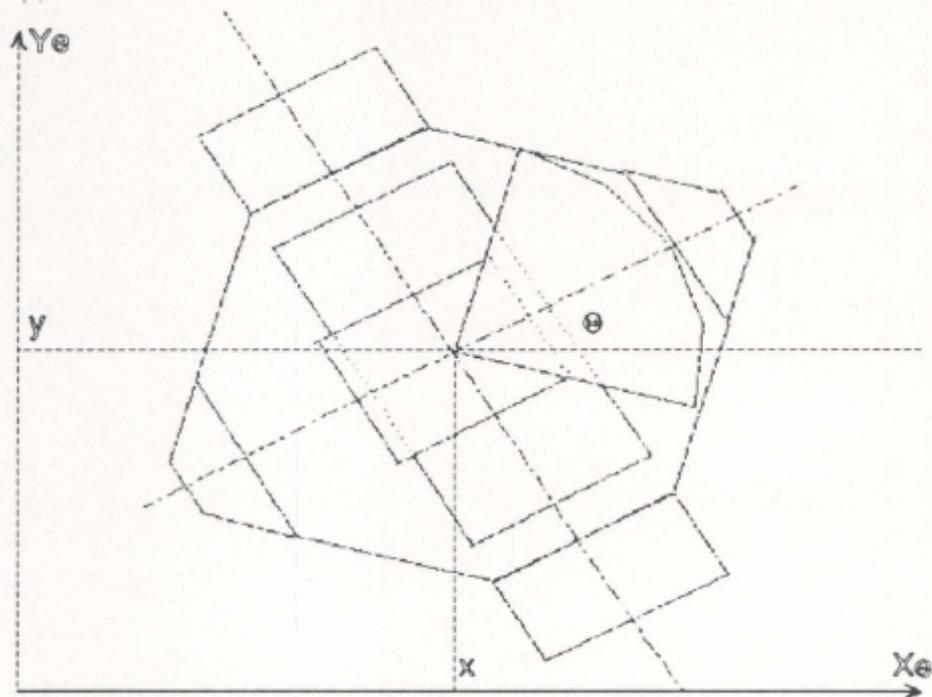


FIGURE IV-12: La position et l'orientation du robot sont définies par les coordonnées cartésiennes (x,y) et l'angle θ .

Les autres variables utilisées sont:

d_d et d_g : Les déplacements exprimés en mètres, des roues, respectivement droite et gauche.

V_d et V_g : Les vitesses linéaire exprimées en mètres par seconde, des roues, respectivement droite et gauche.

U_d et U_g : Les variables de commandes, représentant l'accélération des roues, respectivement droite et gauche.

Le modèle cinématique développé, relie les coordonnées cartésiennes x et y , et l'orientation θ aux variables de

contrôle U_d et U_g , à travers les vitesses linéaire V_d et V_g :

$$\text{I} \begin{cases} \dot{\theta} = (V_d - V_g) / D \\ \dot{x} = ((V_d + V_g) / 2) \cdot \cos(\theta) \\ \dot{y} = ((V_d + V_g) / 2) \cdot \sin(\theta) \\ \dot{V}_d = U_d \\ \dot{V}_g = U_g \end{cases}$$

$$\text{II} \begin{cases} \dot{d}_d = V_d \\ \dot{d}_g = V_g \end{cases}$$

où D est la distance séparant les deux roues motrices.

6 - 2 - 3 Le contrôle optimale.

Pour déduire les conditions de la commande en temps optimal, du mouvement du robot entre la position initiale et la position finale, nous avons utilisé, le principe du maximum de PONTRYAGAIN (P.M.P), où les variables d'états sont:

$$x_1 = x, \quad x_2 = y, \quad x_3 = \theta, \quad x_4 = V_d, \quad x_5 = V_g.$$

Les équations du système I deviennent:

$$\text{III} \begin{cases} \dot{x}_1 = f_1(x, u) = ((x_4 + x_5) / 2) \cdot \cos(x_3) \\ \dot{x}_2 = f_2(x, u) = ((x_4 + x_5) / 2) \cdot \sin(x_3) \\ \dot{x}_3 = f_3(x, u) = (x_4 - x_5) / D \\ \dot{x}_4 = f_4(x, u) = u_1 \\ \dot{x}_5 = f_5(x, u) = u_2 \end{cases}$$

Les deux variables de contrôle sont: $u_1 = U_d$ et $u_2 = U_g$. Le problème d'optimisation consiste à déterminer les deux variables de contrôle u_1 et u_2 , qui mène le système de l'état initiale X_0 à l'état final X_1 en un temps de transition minimal; On considère que les accélérations sont bornées ($|u_1| \leq a_{\max}$ et $|u_2| \leq a_{\max}$).

PONTRYAGAIN à introduit un système d'équation par rapport aux variables auxiliaires Ψ_i définis comme suit:

$$\dot{\Psi}_i = - \sum_{j=1}^n \frac{\partial f_j(x, u)}{\partial x_i} \cdot \Psi_j$$

$$i = 1, 2, \dots, n$$

avec conditions initiales $\Psi_i(t_0) = \lambda_i, \quad i = 1, 2, \dots, n.$

Dans notre cas $n=5$. Ce système étant linéaire et homogène,

admet quelque soient les conditions initiales pour Ψ_i , une solution unique $\Psi=(\Psi_1, \dots, \Psi_5)$.

Nous allons maintenant grouper les équations du mouvement et les variables auxiliaires en un seul HAMILTONIEN H:

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial \Psi_i}$$

$$\frac{d\Psi_i}{dt} = -\frac{\partial H}{\partial x_i}$$

$i=1, \dots, 5$

$$H(\Psi, x, u) = \sum_{j=1}^5 \Psi_j f_j(x, u)$$

$$H(\Psi, x, u) = \Psi_1 \cdot f_1 + \Psi_2 \cdot f_2 + \Psi_3 \cdot f_3 + \Psi_4 \cdot u_1 + \Psi_5 \cdot u_2$$

Alors le P.M.P stipule (Théorème 2 [PON 86]): Si $u(t)$ est une commande admissible (continue par morceaux), transférant le point représentatif de l'état initial $x(t_0)=x_0$ à l'état $x(t_1)=x_1$; alors pour que la commande $u(t)$ et la trajectoire $x(t)$ soient optimales (au sens du temps optimal), il est nécessaire qu'existe un vecteur fonction $\Psi(t)=(\Psi_1(t), \dots, \Psi_n(t))$ continu et non nul, défini par (1) correspondant aux fonctions $u(t)$ et $x(t)$, et tel que: $u(t)$ maximise la fonction $H(\Psi, x, u)$, $\forall t, t \in [t_0, t_1]$.

Essayant de déterminer les variables auxiliaires; d'après l'équation (1) on a:

$$\text{IV} \left\{ \begin{array}{l} \dot{\Psi}_1 = 0 \\ \dot{\Psi}_2 = 0 \\ \dot{\Psi}_3 = \frac{(\Psi_1 \sin(x_3) - \Psi_2 \cos(x_3)) (x_4 + x_5)}{2} \\ \dot{\Psi}_4 = -\frac{(\Psi_1 \cos(x_3) + \Psi_2 \sin(x_3)) - \Psi_3}{D} \\ \dot{\Psi}_5 = -\frac{(\Psi_1 \cos(x_3) + \Psi_2 \sin(x_3)) + \Psi_3}{D} \end{array} \right.$$

Le robot est amené à se mouvoir d'une configuration initiale $x_0 = x(t_0) = (x_i, y_i, \theta_i, v_{di}, v_{gi})$ à une configuration finale $x_1 = x(t_1) = (x_f, y_f, \theta_f, v_{df}, v_{gf})$. Afin de simplifier l'étude, on a pris comme: Position initiale l'origine des coordonnées ($x_i = 0, y_i = 0$), l'orientation initiale $\theta_i = 0$, et les vitesses initiales $v_{di} = v_{gi} = 0$. On suppose que le robot atteint la configuration finale avec une vitesse nulle ($v_{df} = v_{gf} = 0$); Les états initiales et finals du système deviennent:

$$x_0 = x(t_0) = (0, 0, 0, 0, 0)$$

$$x_1 = x(t_1) = (x_f, y_f, \theta_f, 0, 0).$$

Le robot est ordonnés à atteindre une position, l'orientation finale θ_f n'est pas spécifiée, le problème d'optimisation est devenu donc à extrémité droite ($x(t_1)$) libre, la valeur finale des variables auxiliaires $\Psi_i(t)$, est alors déterminée par la condition de transversalité du P.M.P (Théorème 3 [PON 86]). Le vecteur $\Psi(t)$ satisfait à la condition de transversalité à l'extrémité droite $x_1 = x(t_1)$, si le vecteur $\Psi(t_1) = (\Psi_1(t_1), \Psi_2(t_1), \Psi_3(t_1), \Psi_4(t_1), \Psi_5(t_1))$ est orthogonal au plan contenant toutes les positions possibles de x_1 . Un vecteur T contenu dans ce plan est défini ainsi: $T = (0, 0, T_3, 0, 0)$.

$$\Psi(t_1) \perp T \rightarrow \text{le produit scalaire } (\Psi(t_1), T) = 0 \rightarrow \Psi_3(t_1) \cdot T_3 = 0 \rightarrow \Psi_3(t_1) = 0.$$

Cela veut dire que la valeur finale de la troisième variable auxiliaire Ψ_3 doit s'annuler, les autres valeurs sont arbitraires.

Par ailleurs, on sait d'après le P.M.P, que les valeurs optimales des variables de contrôles u_1 et u_2 sont celles qui maximise l'HAMILTONIEN $H(\Psi, x, u)$. Il est évident d'après l'équation (2) que ces valeurs sont:

soit $u_1 = a_{\max}$ si $\Psi_4 > 0$ ($u_2 = a_{\max}$ si $\Psi_5 > 0$),

soit $u_1 = -a_{\max}$ si $\Psi_4 < 0$ ($u_2 = -a_{\max}$ si $\Psi_5 < 0$).

Les variables de contrôle (u_1 et u_2) prennent alors les bornes de leurs intervalles, le contrôle optimale dans ce cas est "BANG-BANG";

Voyons maintenant le cas où l'une des deux variables auxiliaires Ψ_4 ou Ψ_5 s'annule.

Supposons que Ψ_4 s'annule pendant un intervalle de temps fini $[t_2, t_3]$

$$\begin{aligned} \Psi_4 = 0 \rightarrow \dot{\Psi}_4 = 0 &\rightarrow -\frac{\Psi_1 \cos(x_3) + \Psi_2 \sin(x_3)}{2} - \frac{\Psi_3}{D} = 0 \\ &\rightarrow \Psi_3 = -\frac{D}{2} (\Psi_1 \cos(x_3) + \Psi_2 \sin(x_3)) \end{aligned}$$

On pose:

$$\begin{aligned} f(x_3) &= -\frac{(\Psi_1 \cos(x_3) + \Psi_2 \sin(x_3))}{2} \\ \frac{df}{dx_3} = g(x_3) &= \frac{\Psi_1 \sin(x_3) - \Psi_2 \cos(x_3)}{2} \end{aligned}$$

Alors on a:

$$\begin{aligned} \dot{\Psi}_3 &= D \cdot f(x_3) \rightarrow \dot{\Psi}_3 = D \cdot \frac{df}{dx_3} \cdot \dot{x}_3 \\ \text{et du système III on a: } \dot{x}_3 &= \frac{x_4 x_5}{D} \\ &\rightarrow \dot{\Psi}_3 = g(x_3) \cdot (x_4 x_5) \end{aligned}$$

Par ailleurs du système IV on a:

$$\begin{aligned} \dot{\Psi}_3 &= \frac{\Psi_1 \sin(x_3) - \Psi_2 \cos(x_3)}{2} \cdot (x_4 x_5) = g(x_3) \cdot (x_4 x_5) \\ \text{On peut dire que: } g(x_3) \cdot (x_4 x_5) &= g(x_3) \cdot (x_4 x_5) \\ &\rightarrow g(x_3) \cdot x_5 = 0 \end{aligned}$$

Deux cas se présentent:

1^{er} Cas:

$$g(x_3)=0 \Rightarrow \Psi_1 \sin(x_3) = \Psi_2 \cos(x_3)$$

\rightarrow Quatres situations se présentent:

a) $\Psi_1 \neq 0$ et $\Psi_2 \neq 0$

b) $\Psi_1 = 0$ et $\Psi_2 \neq 0$

c) $\Psi_1 \neq 0$ et $\Psi_2 = 0$

d) $\Psi_1 = 0$ et $\Psi_2 = 0$

Pour les situations a), b), et c) on a:

(a) $\rightarrow \operatorname{tg}(x_3) = (\Psi_2 / \Psi_1)$

(b) $\rightarrow \cos(x_3) = 0$

(c) $\rightarrow \sin(x_3) = 0$

De ces trois situations on peut dire que:

$$x_3 = \text{cte} \rightarrow \dot{x}_3 = 0 \rightarrow (\text{du système III}) x_4 = x_5 \rightarrow$$
$$\rightarrow u_1 = \dot{x}_4 = \dot{x}_5 = u_2$$

Les deux accélérations sont identiques sur les deux roues, le mouvement est donc rectiligne pendant l'intervalle $[t_2, t_3]$.

Est ce que le contrôle optimal dans ce cas est "BANG-BANG"?

Pour répondre à cette question, il faut voir la situation qui se présente pour la variable auxiliaire Ψ_5 .

Du système IV on a:

$$\dot{\Psi}_5 = - \frac{(\Psi_1 \cos(x_3) + \Psi_2 \sin(x_3))}{2} + \frac{\Psi_3}{D}$$
$$= f(x_3) + \frac{\Psi_3}{D}$$

On est dans le cas où $g(x_3)=0 \rightarrow$ (du système IV) $\dot{\Psi}_3=0 \rightarrow \Psi_3=\text{cte}$.
Par ailleurs on a: $x_3=\text{cte}$, $\Psi_1=\text{cte}$, et $\Psi_2=\text{cte}$.

On peut dire alors que $\dot{\Psi}_5=\text{cte}$.

Si $\dot{\Psi}_5=\text{cte} \neq 0 \rightarrow \Psi_5$ varie linéairement en fonction du temps, et le contrôle optimal dans ce cas est "BANG-BANG".

Si $\dot{\Psi}_5=0 \rightarrow f(x_3) + \Psi_3/D = 0$,

et d'autre part on a: $\dot{\Psi}_4 = f(x_3) - \Psi_3/D = 0$

alors, $f(x_3) = g(x_3) = \Psi_3 = 0$.

Cela veut dire que $\Psi_1 = \Psi_2 = 0$, ce qui n'est pas conforme aux situations (a), (b), et (c). $\dot{\Psi}_5 = 0$ n'est pas une solution possibles.

Voyons maintenant la situation d). Si $\Psi_1 = \Psi_2 = 0$ durant l'intervalle $[t_2, t_3]$, alors d'après le système IV ces variables s'annulent durant tout l'intervalle $[t_0, t_1]$.

$\Psi_1 = \Psi_2 = 0 \rightarrow f(x_3) = g(x_3) = 0 \rightarrow \dot{\Psi}_3 = 0 \rightarrow \Psi_3 = \text{cte}$, durant tout l'intervalle $[t_0, t_1]$.

Par ailleurs on a: $\Psi_4 = 0$ durant l'intervalle $[t_2, t_3] \rightarrow \dot{\Psi}_4 = f(x_3) - \Psi_3/D = 0 \rightarrow \Psi_3 = 0$ durant tout l'intervalle $[t_0, t_1]$, ceci aboutit donc a:

$\dot{\Psi}_5 = f(x_3) + \Psi_3/D = 0 \rightarrow \Psi_5 = \text{cte}$, durant tout l'intervalle $[t_0, t_1]$.

L'hamiltonien $H(\Psi, x, u)$ devient: $H(\Psi, x, u) = \Psi_5 \cdot u_2 = \text{cte} \cdot u_2$

La variable de contrôle u_2 ne change pas de signe durant tout l'intervalle $[t_0, t_1]$, ce qui entraine donc une monotonie croissante ou décroissante de la vitesse, impossible alors d'atteindre l'état final x_1 .

$$x_1 = x(t_1) = (x_f, y_f, \theta_f, 0, 0)$$

là aussi $\Psi_1 = \Psi_2 = 0$ n'est pas une solution possible.

2^{eme} Cas

$x_5 = 0$ dant l'intervalle $[t_2, t_3]$, alors la variable de contrôle u_2 s'annule ($u_2 = \dot{x}_5$) dans cet intervalle, et la variable auxiliaire Ψ_5 doit s'annuler dans l'intervalle $[t_2, t_3]$.

$$\dot{\Psi}_5 = f(x_3) + \Psi_3/D = 0$$

$$\dot{\Psi}_4 = f(x_3) - \Psi_3/D = 0$$

Ceci implique que $\Psi_3 = 0$ et $f(x_3) = 0$

$$\rightarrow \dot{\Psi}_3 = g(x_3)(x_4 + x_5) = 0, \text{ alors:}$$

Soit $(x_4 + x_5) = 0 \rightarrow x_4 = x_5 = 0 \rightarrow u_1 = u_2 = 0$, le robot dans ce cas reste immobile dans l'intervalle $[t_2, t_3]$, ce qui n'est pas optimal (dans le sens temps optimal).

$$\text{Soit } g(x_3) = 0 \rightarrow f(x_3) = g(x_3) = 0 \rightarrow \Psi_1 = \Psi_2 = 0.$$

On sait d'après le système IV que $\Psi_1 = \Psi_2 = \text{cte}$ dans l'intervalle $[t_0, t_1]$, alors on peut dire dans ce cas que:

$$\Psi_1 = \Psi_2 = \Psi_3 = 0 \text{ dans tout l'intervalle } [t_0, t_1].$$



Par ailleurs on a :

$f(x_3) = \Psi_3 = 0$ dans tout l'intervalle $[t_0, t_1]$, cela veut dire que $\dot{\Psi}_4 = \dot{\Psi}_5 = 0$ dans tout l'intervalle $[t_0, t_1]$, et $\Psi_4 = 0$ et $\Psi_5 = \text{cte}$

dans cet intervalle; D'autre part, comme il a été dit précédemment, on est dans le cas où Ψ_5 doit s'annuler dans l'intervalle $[t_2, t_3]$, alors on peut dire que $\Psi_5 = 0$ dans l'intervalle $[t_0, t_1]$; Le vecteur $\Psi = (\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5)$ s'annule durant toute la trajectoire, ce qui n'est pas admissible d'après le P.M.P.

En conclusion, on peut dire, que dans le cas où l'une des variables auxiliaires Ψ_4 ou Ψ_5 s'annule, le seul contrôle optimal admissible est "BANG-BANG". A travers cette étude, nous avons montré que toutes les solutions possibles du système III, permettant au robot d'atteindre la position finale en un temps optimal, nécessite un contrôle optimal "BANG-BANG".

6 - 2 - 4 Caractérisation des trajectoires "BANG-BANG".

a) Introduction.

Dans la section précédente, on a montré que toutes les commandes en temps optimal, permettant au robot de se mouvoir d'un point (x_i, y_i) à un point (x_f, y_f) sont "BANG-BANG"; Les trajectoires établies sous ce contrôle on les appelle trajectoires "BANG-BANG"; Ces trajectoires sont caractérisées par le nombre de commutations des variables de contrôle u_1 et u_2 (représentant les accélérations des deux roues), entre les bornes de leurs intervalles (a_{\max} et $-a_{\max}$). La vitesse des deux roues consiste donc en une succession de segments linéaire croissants et décroissants. Dans cette section, nous allons d'abord montrer que ce type de trajectoires vérifie la condition nécessaire du P.M.P, et nous sélectionnons par la suite, celles qui sont optimales dans le sens "temps optimal"; Pour se faire, nous avons utilisés la notion des "instants de commutation", où les accélérations (u_1 et u_2) passent d'une borne à une autre de leurs intervalles.

b) calcul des variables auxiliaires.

Pour vérifier qu'une trajectoire "BANG-BANG" satisfait la condition nécessaire du P.M.P, il faut déterminer les variables auxiliaires Ψ_i , $i=1,2,\dots,5$, et montrer que les signes des variables Ψ_4 et Ψ_5 suivent, respectivement les signes des variables de contrôle u_1 et u_2 .

Les équations du système IV étant linéaire et homogènes, le système admet donc une solution unique quelque soit les conditions initiales λ_i .

$$\dot{\Psi}_1 = 0 \rightarrow \Psi_1 = \lambda_1$$

$$\dot{\Psi}_2 = 0 \rightarrow \Psi_2 = \lambda_2$$

Pour simplifier le problème, on prend comme position initiale l'origine des coordonnées (0,0), on obtient:

$$\Psi_3 = \lambda_1 x_2 - \lambda_2 x_1 + \lambda_3$$

$$\dot{\Psi}_4 = -\lambda_1 \left(\frac{\cos x_3}{2} + \frac{x_3}{D} \right) + \lambda_2 \left(\frac{x_1}{D} - \frac{\sin x_3}{2} \right) - \frac{\lambda_3}{D}$$

$$\dot{\Psi}_5 = \lambda_1 \left(\frac{x_1}{D} - \frac{\cos x_3}{2} \right) - \lambda_2 \left(\frac{\sin x_3}{2} + \frac{x_1}{D} \right) + \frac{\lambda_3}{D}$$

On pose:

$$\dot{z}_1 = \cos(x_3), \text{ et } z_1(0) = 0$$

$$\dot{z}_2 = \sin(x_3), \text{ et } z_2(0) = 0$$

$$\dot{z}_3 = x_1, \text{ et } z_3(0) = 0$$

$$\dot{z}_4 = x_2, \text{ et } z_4(0) = 0$$

$$\Psi_4 = -\lambda_1 \left(\frac{z_1 + z_4}{2} \right) + \lambda_2 \left(\frac{z_3 - z_2}{D} - \frac{\lambda_3 t}{D} \right) + \lambda_4$$

$$\Psi_5 = \lambda_1 \left(\frac{z_1 - z_4}{2} \right) - \lambda_2 \left(\frac{z_2 + z_3}{D} \right) + \frac{\lambda_3 t}{D} + \lambda_5$$

Aux "instants de commutation", les variables de contrôle changent de signes, alors à ces moments, les variables auxiliaires Ψ_4 et Ψ_5 doivent s'annuler, ce qui nous permet de déterminer les conditions initiales λ_i en résolvant l'équation matricielle $A \cdot \lambda = 0$, où $\lambda^T = (\lambda_1, \lambda_2, \dots, \lambda_5)$, et les lignes de la matrice A sont définies par l'équation de Ψ_4 ou Ψ_5 appliquée aux "instants de commutation"; Si par exemple, à l'instant τ_1 une commutation se produit au niveau de la roue de droite, alors on a $\Psi_4(\tau_1) = 0$, ce qui définit une ligne de la matrice A. D'une manière générale, une trajectoire de

n "instant de commutation" répartis sur les deux roues, produit une matrice A de n lignes. Une autre ligne peut être déterminée en appliquant la condition de transversalité, à l'équation de Ψ_3 à l'instant t_1 . Pour que le vecteur λ admette une solution non nulle, le rang de la matrice A doit être strictement inférieur à cinq (5), autrement dit, le nombre total des "instants de commutation", doit être strictement inférieur à quatre (4). Une trajectoire "BANG-BANG" à trois "instants de commutation" répartis sur les deux roues, vérifie alors la condition nécessaire du P.M.P. Reste donc à sélectionner parmi ces trajectoires, celle qui peuvent atteindre n'importe quel point du plan x.o.y et montrer quelles sont optimales dans le sens "temps optimal".

c) Les trajectoires "BANG-BANG" à trois (3) "instants de commutation".

Dans cette étude, le repère lié à l'environnement est défini de telle manière, que tous les points possibles à atteindre se trouvent dans le quadrant positif ($x > 0$ et $y > 0$), et on répartit les trois "instants de commutation" sur les deux roues comme suit:

* Deux "instants de commutation" sur la roue de gauche, ceci entraîne une succession de trois segments linéaire croissants et décroissants, représentant la vitesse de la roue de gauche en fonction du temps, dont les durées sont: tg_1 , tg_2 , et tg_3 .

* Et un seul "instant de commutation" sur la roue de droite, ceci entraîne une succession de deux segments linéaire croissants et décroissants, représentant la vitesse de la roue droite en fonction du temps, dont les durées sont td_1 et td_2 .

Les valeurs finales et initiales de la vitesse d'une roue étant nulles, alors chaque une des deux roues, accélère pendant la moitié de la durée totale de la trajectoire, et décélère pendant la deuxième moitié.

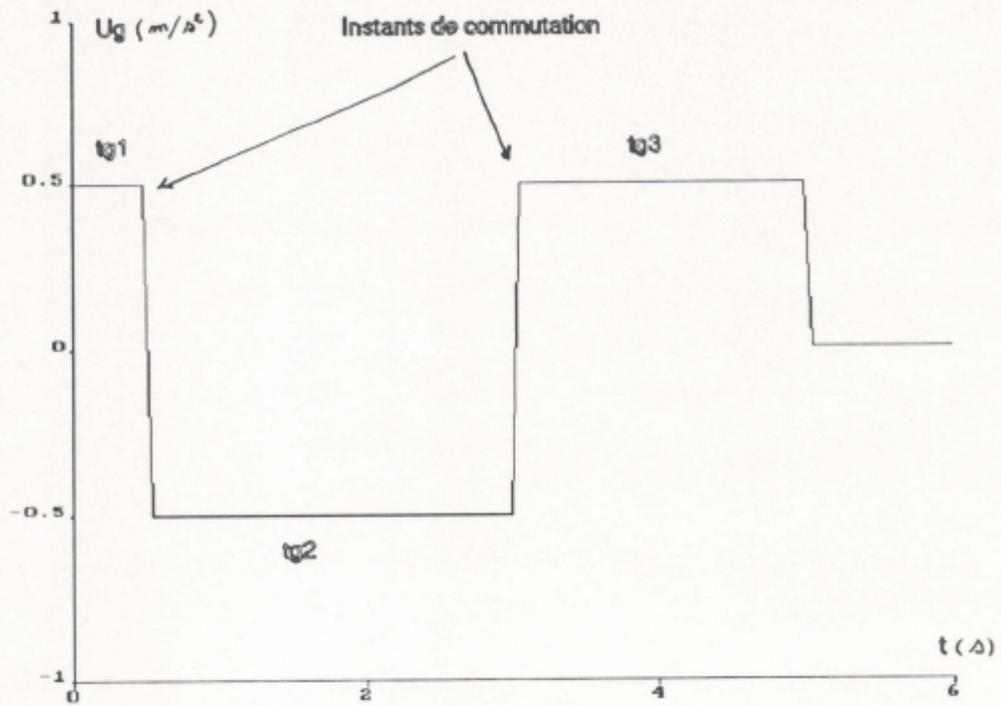
Soit T la durée totale de la trajectoire, dans ce cas on

a:

$$td_1 = td_2 = T/2$$

$$tg_2 = T/2$$

$$\text{et } tg_1 + tg_3 = T/2 \quad (\text{voir FIGURE IV-13})$$



Une trajectoire "BANG-BANG" à trois (3) "instants de commutation", est alors complètement définie par le couple (tg_1, T) , et selon les accélérations initiales des roues, deux types de trajectoires "BANG-BANG" à trois (3) "instants de commutation" peuvent se présenter:

* Le type++, les accélérations initiales des roues sont positives ($U_d = U_g = +a_{\max}$), ce type de trajectoires commence par une translation et se termine par une rotation.

* Le type+-, l'une des accélération est positive ($U_d = +a_{\max}$) et l'autre est négative ($U_g = -a_{\max}$), ce type de trajectoires commence par une rotation et se termine par une translation.

Nous montrerons dans ce qui suit, que les trajectoires de type++ ne peuvent être optimales, pour ce faire, nous avons fixé l'un des paramètre (T) à 5s, et varié l'autre (tg_1) de 0 à 0,7s par pas de 0,05s pour les trajectoires du

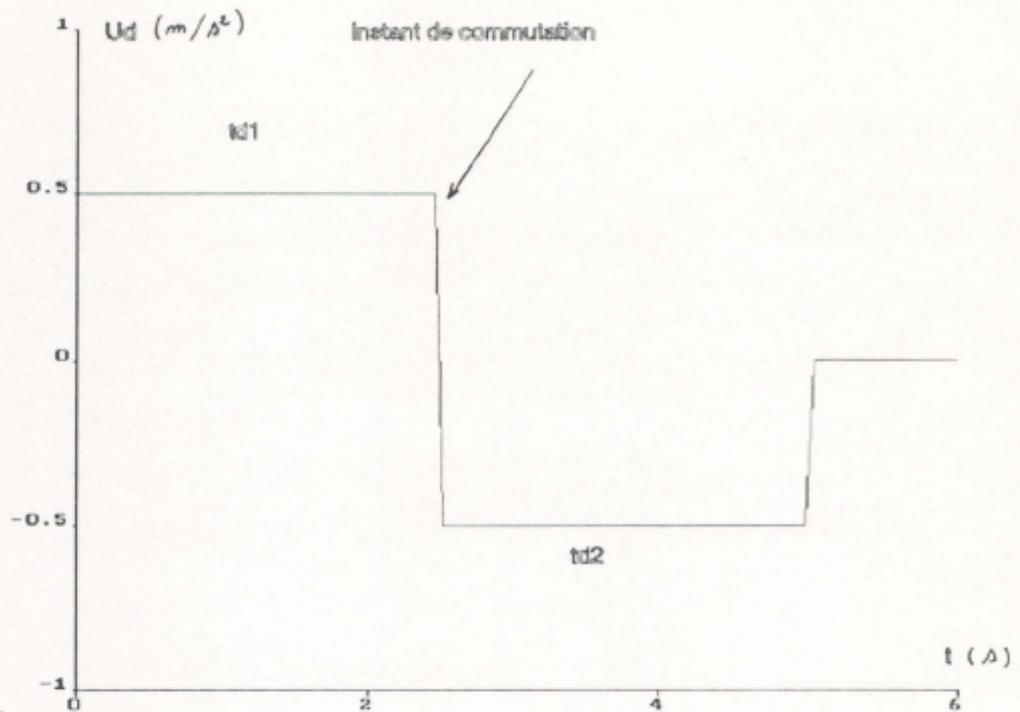


FIGURE IV-13: Représentation des accélérations U_g et U_d , avec deux "instants de commutation" sur la roue de gauche et un "instant de commutation" sur la roue de droite.

type+-, et de 0 à 2,4s par pas de 0,3s pour les trajectoires du type++. L'étendue des positions possibles que peuvent atteindre les trajectoires de type++ et de type+- sont représentées par la **FIGURE IV-14**; Ces courbes sont obtenues par le logiciel "SIMNON", spécialisé dans la simulation des processus de contrôle, régis par des systèmes d'équations différentielles. Nous avons pris $a_{max}=0,4m/s^2$, la durée totale d'exécution des trajectoires $T=5s$, et nous avons fait varier le paramètre (tg_1). Ces courbes montre clairement que les trajectoires de type+-, atteignent des positions beaucoup plus éloignées que les trajectoires de type++, et ceci pour une même durée totale d'exécution de 5s ($T=5s$), cela veut dire que les trajectoires de type++ ne peuvent être optimales dans le sens "temps optimal".

Dans le cas de la **FIGURE IV-15**, nous avons pris quatre (4) valeurs différentes de la durée totale d'exécution des trajectoires type+- ($T=5s, T=8s, T=10s, T=12s$), et pour chaque

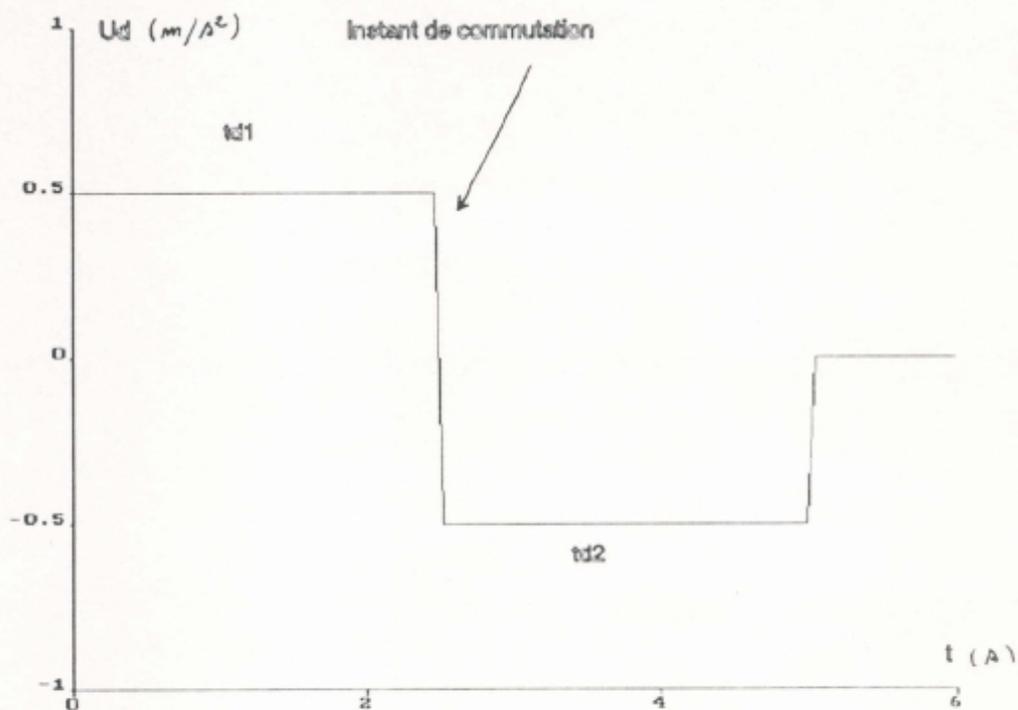
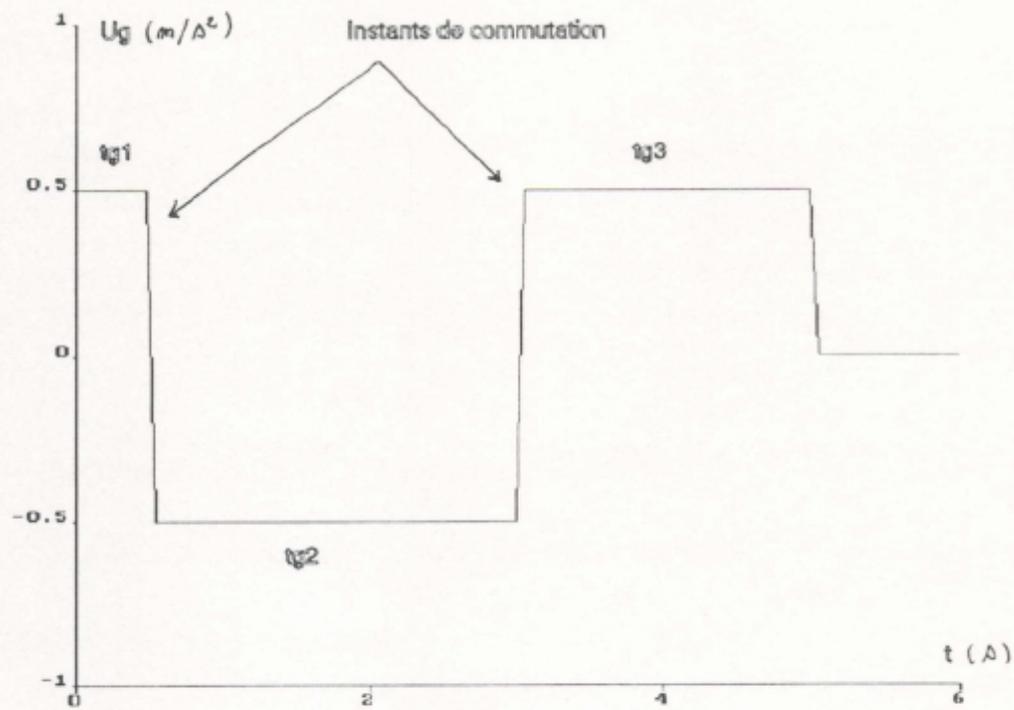


FIGURE IV-13: Représentation des accélérations U_g et U_d , avec deux "instants de commutation" sur la roue de gauche et un "instant de commutation" sur la roue de droite.

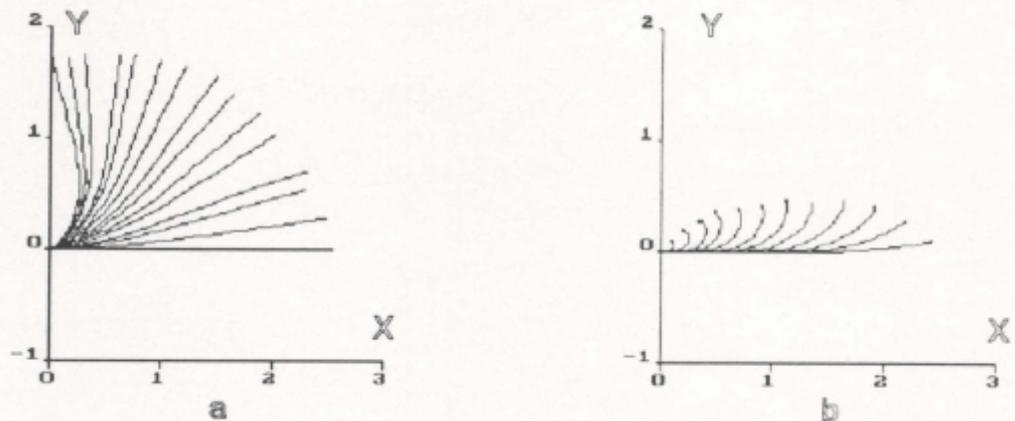


FIGURE IV-14: positions atteintes par les trajectoires à 3 "instants de commutation" pour une durée $T=5s$. a) type+-; tg_1 varie de 0 à 0,7s par pas de 0,05s. b) type++; tg_1 varie de 0 à 2,4s par pas de 0,3s

valeur de T , nous avons fait varier le paramètre tg_1 de 0 à 0,3s par pas de 0,1s, les courbes obtenues montrent que les positions atteintes sont de plus en plus éloignées, et ceci avec l'augmentation de la durée d'exécution T ; ce qui veut dire que les trajectoires "BANG-BANG" à trois "instants de commutation", peuvent atteindre n'importe quel point de l'environnement, et à chaque position (x,y) correspond un couple de paramètres (T, tg_1) .

d) Mise en évidence de l'optimalité des trajectoires "BANG-BANG" à trois (3) "instants de commutation" type+-.

Dans cette section, nous montrerons par simulation, l'optimalité des trajectoires "BANG-BANG" à trois "instants de commutation" type+-, permettant au robot d'atteindre une position prédéterminée, en un temps optimal.

Soient, $n=n_d+n_g$ le nombre totale des "instants de commutation" réparti sur les deux roues (n_d sur la roue de droite et n_g sur la roue de gauche), et on dit que les

Les courbes de la **FIGURE IV-16** et **FIGURE IV-17**, montre les positions atteintes par, respectivement les trajectoires de la classe 13 et la classe 22; Dans le cas de ces courbes, nous avons fixé la durée totale d'exécution T à 5s ($T=5s$) et le paramètre tg_1 à 0,3s ($tg_1=0,5s$), et nous avons pris quatre valeurs différentes de la durée supplémentaire t_{sup} de 0 à 0,6s par pas de 0,2s; Les positions atteintes sont de moins en moins éloignées, et ceci avec l'augmentation de la durée t_{sup} , cela veut dire que les trajectoires "BANG-BANG" à trois (3) "instants de commutation", sont plus rapide que les trajectoires "BANG-BANG" avec un nombre d'"instants de commutation" supérieur à trois; Ceci peut aussi dire que les trajectoires "BANG-BANG" à deux (2) "instants de commutation", sont plus rapide que les trajectoires "BANG-BANG" à trois (3) "instants de commutation", néanmoins les trajectoires à deux (2) "instants de commutation" de la classe 11 type+-, ne peut donner naissance à une translation, ce qui est impossible d'atteindre la position prédéterminée; alors que le type++ de la classe 11, ne peut donner naissance à une rotation, impossible d'atteindre la position prédéterminée dans le cas où l'orientation initiale θ_1 n'est pas dans la direction de la position à atteindre. En conclusion on peut dire que les trajectoires "BANG-BANG" à trois (3) "instants de commutation" sont les plus rapides.

6 - 2 - 5 Application de la méthode au planification de trajectoires

Le planificateur développé dans la section 3 de ce chapitre, génère l'itinéraire globale au but final, sous forme d'une suite d'emplacements, la trajectoire finale obtenue est optimale dans le sens " distance euclidienne"; Le planificateur développé dans cette section, permet de planifier des trajectoires optimales dans le sens "temps-optimal" entre ces emplacements. L'implémentation de cette méthode, nécessite l'emploi d'un contrôleur de vitesse et de position. Le contrôleur proposé est développé dans l'annexe A.

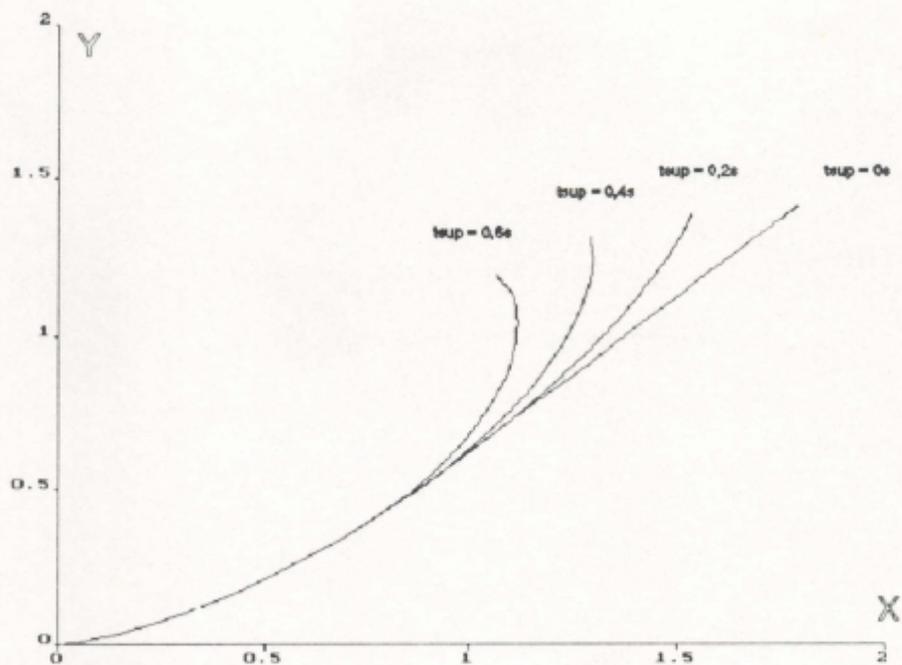


FIGURE IV-16: Positions atteintes par les trajectoires à 4 "instants de commutation" classe 13, pour une durée $T=5s$, avec $tg1=0,3s$, et t_{sup} varie de 0 (3 "instants de commutations") à 0,6s par pas de 0,2s.

REMARQUE

La méthode numérique de POWEL [PRE 88] permet de déterminer le couple $(T, tg1)$ pour une position (x, y) donnée.

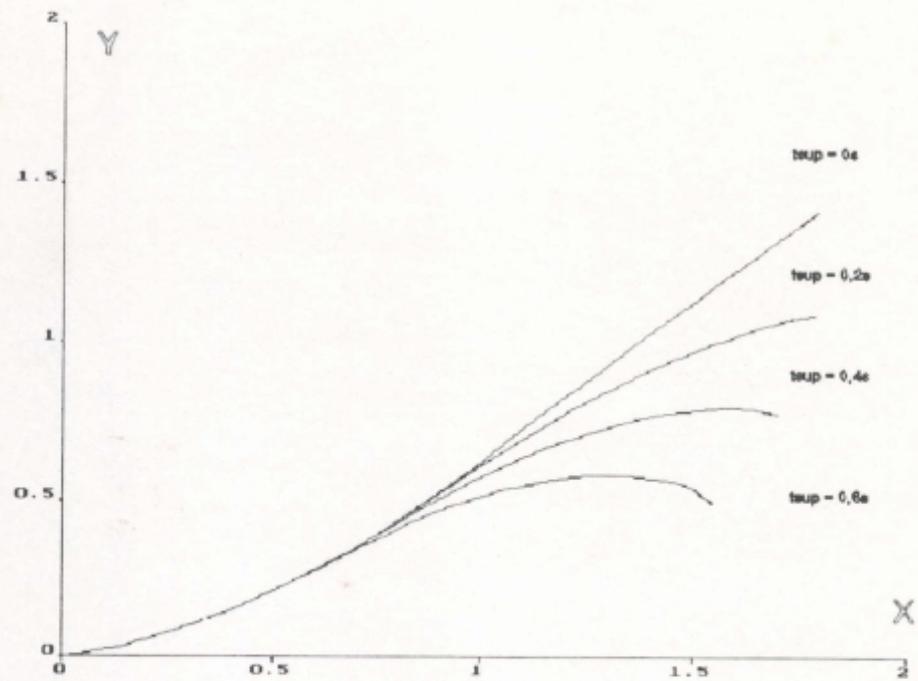


FIGURE IV-17: Positions atteintes par les trajectoires à 4 "instants de commutation" classe 22, pour une durée $T=5s$, avec $t_{gl}=0,3s$, et t_{sup} varie de 0 (3 "instants de commutation") à 0,6s par pas de 0,2s.

dans cette étude, nous avons découplé les deux moteurs, et les résultats obtenus par simulation sont représentés aux **FIGURE IV-18 et IV-19**. La **FIGURE IV-18** représente l'effet du contrôleur où aucune perturbation n'est appliquée au système, le moteur suit fidèlement la trajectoire consigne; Dans le cas de la **FIGURE IV-19**, une perturbation de 100% a été appliquée aux paramètres du système (k et τ) à l'instant $t = 0.6$ s, les **FIGURES IV-19-a et IV-19-b** montre qu'au bout de 6 itérations, l'estimateur atteint la bonne valeur des paramètres du moteur (k et τ), le contrôleur compense l'effet de cette perturbation, et la trajectoire consigne est toujours maintenue par le moteur (**FIGURE IV-19-c**).

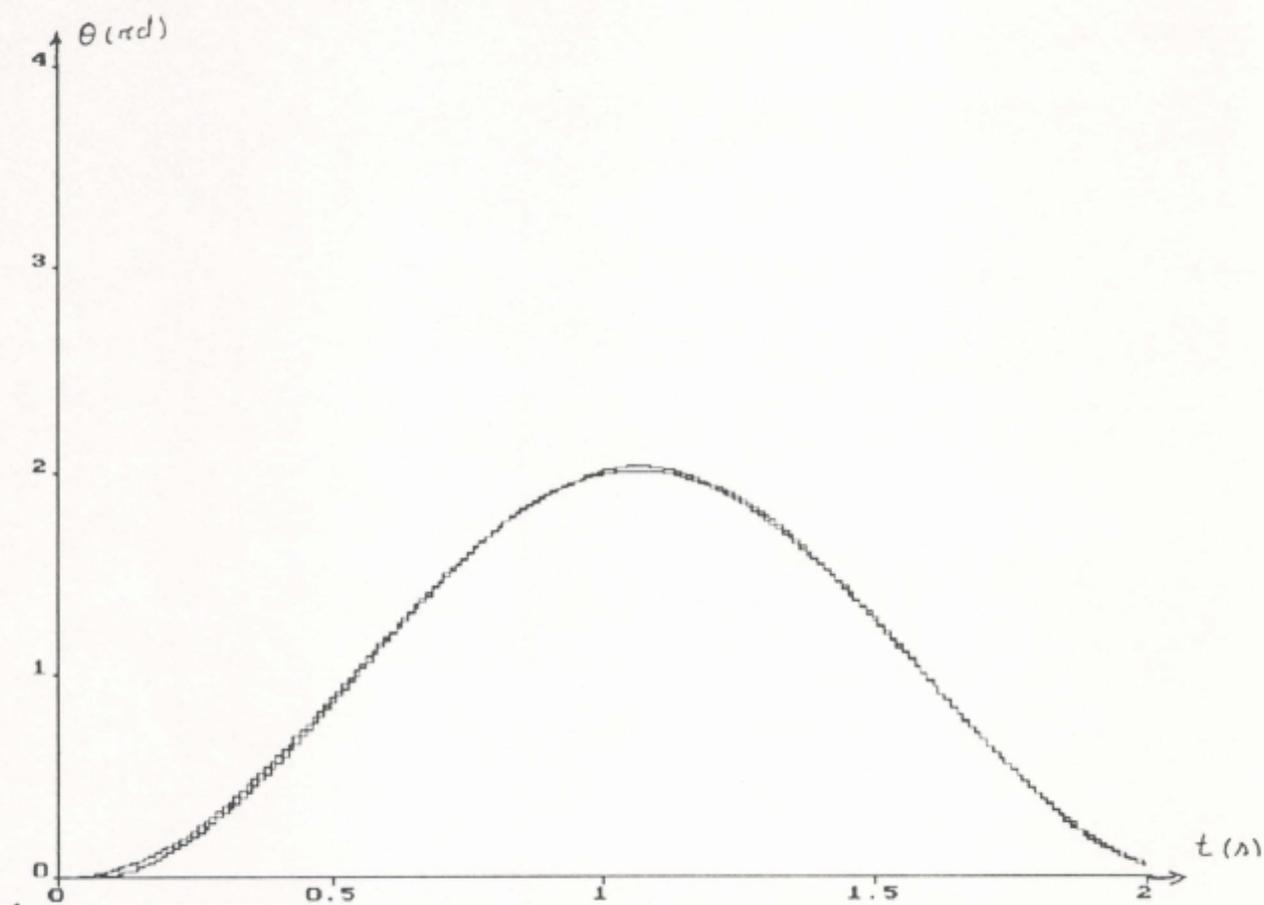


FIGURE IV-18: Sans les perturbations le moteur suit fidèlement la position consigne.

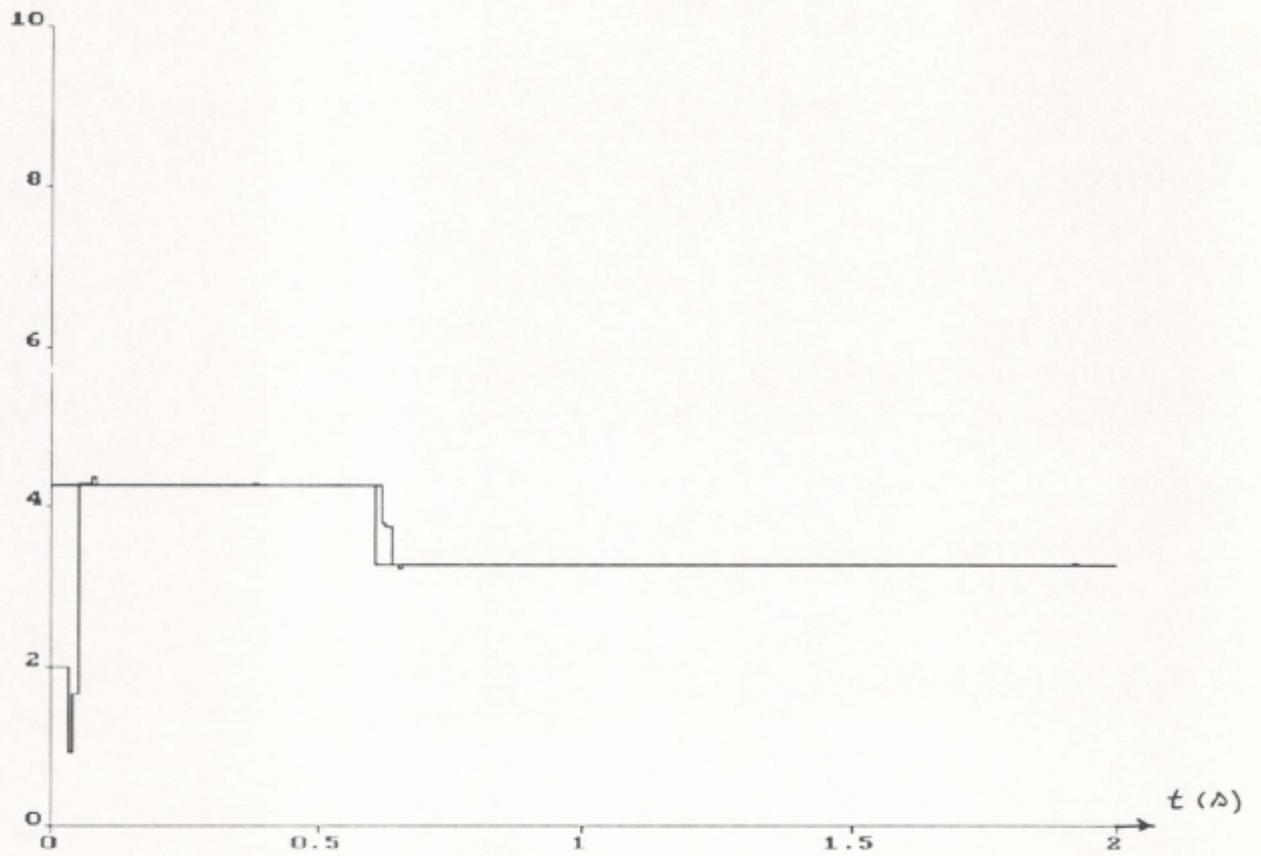


FIGURE IV-19-a: l'effet de la perturbation sur le paramètre K, l'estimateur atteint la bonne valeur au bout de pratiquement 6 itérations

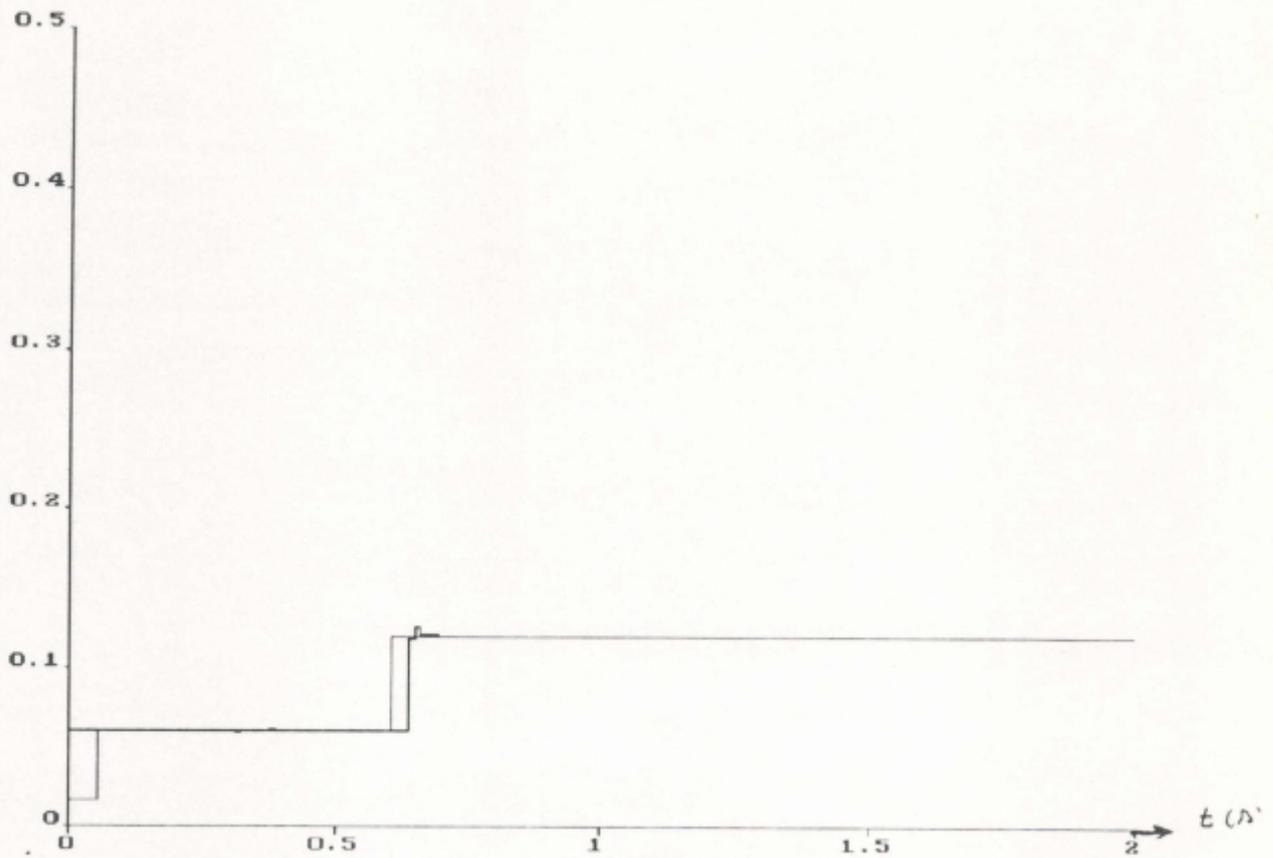
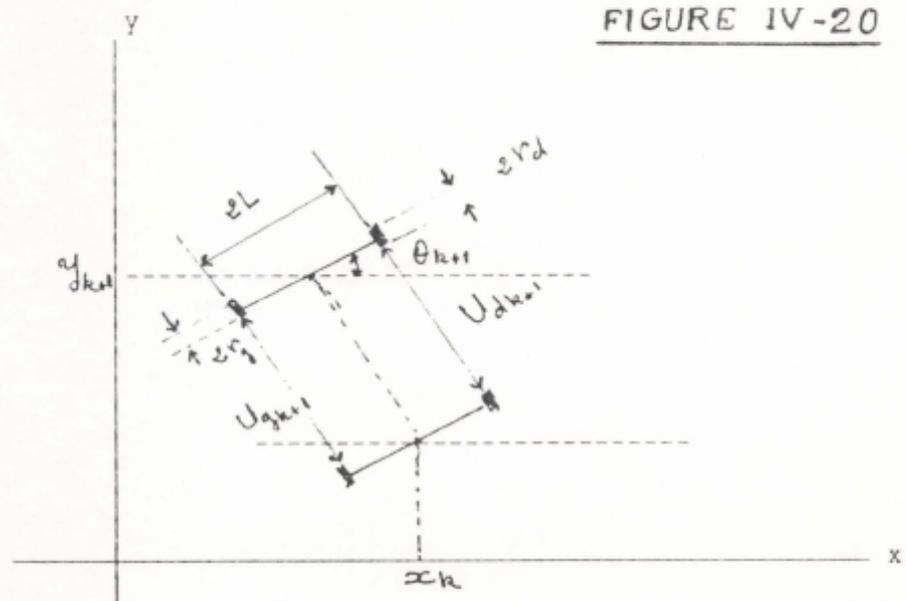


FIGURE IV-19-b: l'effet de la perturbation sur le paramètre τ , l'estimateur atteint la bonne valeur au bout de pratiquement 6 itérations

6 - 3 MODELISATION PROBABILISTE DE L'ODOMETRIE

a) Developpement du modèle



Le Robot possède deux roues motrices équipées de codeurs

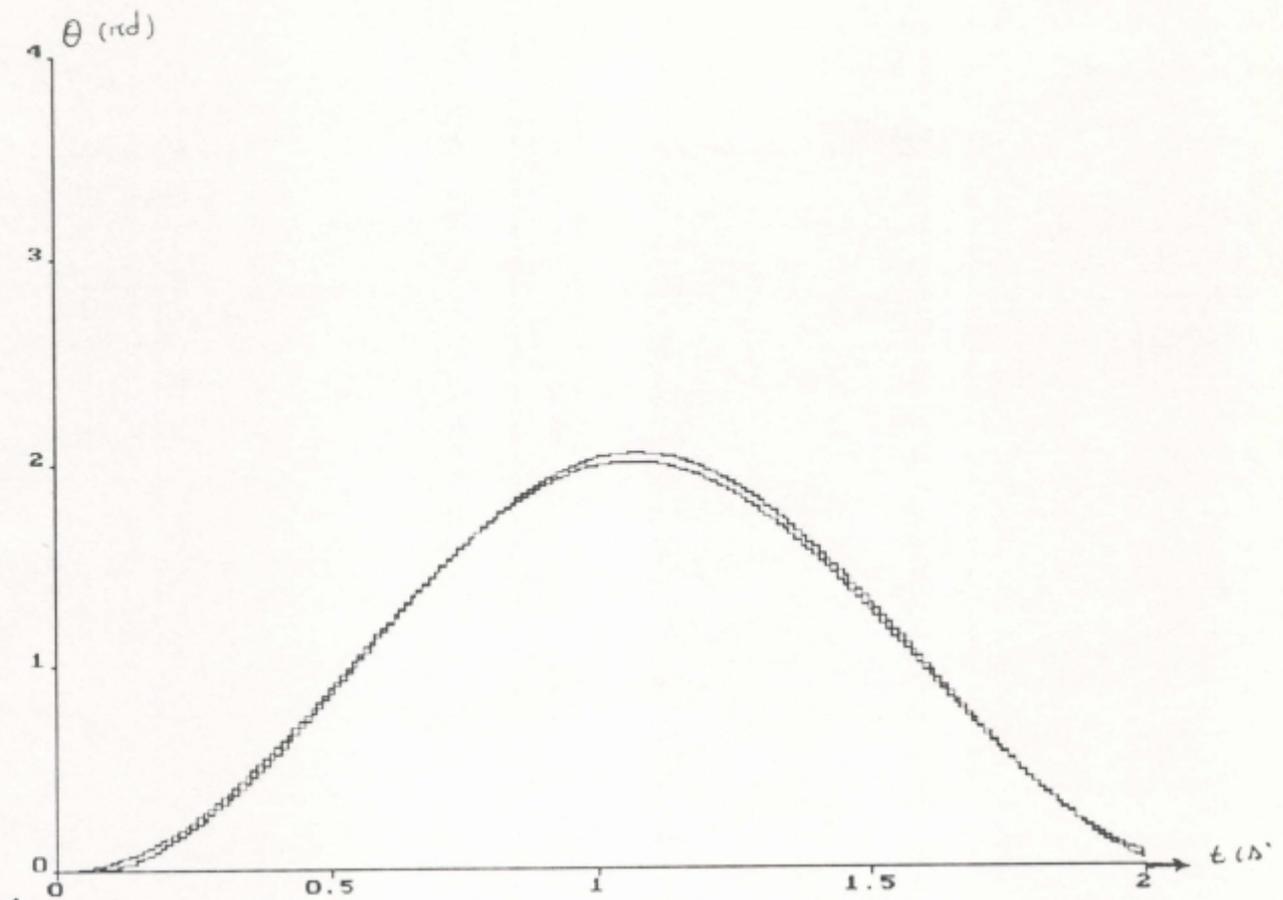


FIGURE IV-19-c: Trajectoire prise par le moteur en appliquant une perturbation sur les paramètres du moteur de 100%.

mesurant leur déplacement angulaire.

La position du Robot à l'instant $(k+1)$ est donnée par:

$$X_{k+1} = X_k + (1/2) * (U_{gk+1} + U_{dk+1}) * \cos \theta_k$$

$$Y_{k+1} = Y_k + (1/2) * (U_{gk+1} + U_{dk+1}) * \sin \theta_k$$

$$\theta_{k+1} = \theta_k + (1/L) * (U_{dk+1} - U_{gk+1})$$

U_{dk+1} et U_{gk+1} étant les déplacements des deux roues entre les deux instants de mesure k et $k+1$.

$$U_{dk+1} = r_d * \Delta \Psi_{dk+1}$$

$$U_{gk+1} = r_g * \Delta \Psi_{gk+1}$$

avec:

r_d et r_g les rayons des deux roues.

$\Delta \Psi_{dk+1}$ et $\Delta \Psi_{gk+1}$ la variation angulaire de chacune d'elles.

Soit sous la forme matricielle:

$$P_{k+1} = P_k + M_k * d_{k+1} \quad (1)$$

ou :

$$P_k = \begin{bmatrix} X_k \\ Y_k \\ \Theta_k \end{bmatrix} ; \quad M_k = \begin{bmatrix} \frac{\cos \Theta_k}{2} & \frac{\cos \Theta_k}{2} \\ \frac{\sin \Theta_k}{2} & \frac{\sin \Theta_k}{2} \\ -1/L & 1/L \end{bmatrix} \quad \text{et} \quad d_{k+1} = \begin{bmatrix} U_{gk+1} \\ U_{dk+1} \end{bmatrix}$$

Pour obtenir la variance sur la position du Robot, nous en exprimons d'abord les variances sur les paramètres constants de l'équation (1): (le rayon des roues et la longueur de l'axe). Cette équation non-linéaire n'est malheureusement valide que pour de petits déplacements, ce qui rend la calibration statistique de ces paramètres difficile.

Toutefois, si nous supposons que les erreurs sur les paramètres constants conduisent à une erreur globale proportionnelle à la distance parcourue par chacune des roues, nous pouvons donner une évaluation de la variance.

Soit: $P_{k+1} = F(P_k, d_{k+1})$

Nous supposons que l'erreur sur le déplacement relatif de chacune des roues est indépendante de la position de départ, nous obtenons:

$$P_{k+1} = F(P_k, d_{k+1}) + \Delta P_{k+1}$$

avec: $P_{k+1} = F(P_k, d_{k+1})$ position estimée du Robot à l'instant k+1.

$$\text{et} \quad \Delta P_{k+1} = J_{P_k} * \Delta P_k + J_{d_{k+1}} * \Delta d_{k+1}$$

L'erreur sur la position du Robot à l'instant k+1 s'exprime alors:

$$V(P_{k+1}) = E(\Delta P_{k+1} * (\Delta P_{k+1})^T) \quad (2)$$

Comme on peut considérer que l'erreur sur le déplacement du Robot est indépendante de l'erreur sur la mesure, le produit:

$\Delta P_k * (\Delta d_{k+1})^T = 0$; ainsi en développant l'équation (2) on trouve:

J_{pk} et J_{dk+1} sont les Jacobiennes de F autour de la position et du déplacement estimés pour le Robot exprimées comme suit:

$$J_{pk} = \begin{bmatrix} \frac{\partial f_1}{\partial X_k} & \frac{\partial f_1}{\partial Y_k} & \frac{\partial f_1}{\partial \theta_k} \\ \frac{\partial f_2}{\partial X_k} & \frac{\partial f_2}{\partial Y_k} & \frac{\partial f_2}{\partial \theta_k} \\ \frac{\partial f_3}{\partial X_k} & \frac{\partial f_3}{\partial Y_k} & \frac{\partial f_3}{\partial \theta_k} \end{bmatrix}$$

$$f_1 = X_k + \delta_d * \cos \theta_k$$

$$f_2 = Y_k + \delta_d * \sin \theta_k$$

$$f_3 = \theta_k + (1/L) * (U_{dk+1} - U_{gk+1})$$

avec:

$$\delta_d = (1/2) * (U_{gk+1} + U_{dk+1})$$

$$J_{pk} = \begin{bmatrix} 1 & 0 & -\delta_d * \sin \theta_k \\ 0 & 1 & \delta_d * \cos \theta_k \\ 0 & 0 & 1 \end{bmatrix}$$

de même:

$$J_{dk+1} = \begin{bmatrix} \frac{\partial f_1}{\partial U_{gk+1}} & \frac{\partial f_1}{\partial U_{dk+1}} \\ \frac{\partial f_2}{\partial U_{gk+1}} & \frac{\partial f_2}{\partial U_{dk+1}} \\ \frac{\partial f_3}{\partial U_{gk+1}} & \frac{\partial f_3}{\partial U_{dk+1}} \end{bmatrix} = \begin{bmatrix} \frac{\cos \theta_k}{2} & \frac{\cos \theta_k}{2} \\ \frac{\sin \theta_k}{2} & \frac{\sin \theta_k}{2} \\ -1/L & 1/L \end{bmatrix}$$

donc on a:

$$J_{dk+1} = M_k$$

$$\begin{aligned}
\text{D'où:} \quad & \Delta P_{k+1} * (\Delta P_{k+1})^T = \\
& = (J_{pk} * \Delta P_k + J_{dk+1} * \Delta d_{k+1}) * (J_{pk} * \Delta P_k + J_{dk+1} * \Delta d_{k+1})^T \\
& = (J_{pk} * \Delta P_k + J_{dk+1} * \Delta d_{k+1}) * [(\Delta P_k)^T * J_{pk}^T + (\Delta d_{k+1})^T * J_{dk+1}^T] \\
& = J_{pk} * \Delta P_k * (\Delta P_k)^T * J_{pk}^T + J_{dk+1} * \Delta d_{k+1} * (\Delta d_{k+1})^T * J_{dk+1}^T
\end{aligned}$$

Ce qui donne:

$$\begin{aligned}
V(P_{k+1}) & = E(\Delta P_{k+1} * (\Delta P_{k+1})^T) \\
& = J_{pk} * E(\Delta P_k * (\Delta P_k)^T) * J_{pk}^T + J_{dk+1} * E(\Delta d_{k+1} * (\Delta d_{k+1})^T) * J_{dk+1}^T \\
& = J_{pk} * V(P_k) * J_{pk}^T + J_{dk+1} * V(d_{k+1}) * J_{dk+1}^T \\
V(P_{k+1}) & = J_{pk} * V(P_k) * J_{pk}^T + J_{dk+1} * V(d_{k+1}) * J_{dk+1}^T
\end{aligned}$$

Pour modéliser l'erreur sur les paramètres constants et la mesure effectuée à

l'instant k+1, nous avons choisi:

$$V(d_{k+1}) = \alpha^2 * \begin{bmatrix} (U_{gk+1})^2 & 0 \\ 0 & (U_{dk+1})^2 \end{bmatrix}$$

avec: $\alpha \approx 0.03$.

Quoiqu'il en soit, le glissement des roues et les dérapages transversaux qui ne peuvent être identifiés statistiquement introduisent des biais imprévisibles. Nous ne pouvons donc pas produire une variance fiable sur la position du Robot. Par contre, il est possible, en supposant que l'erreur sur le déplacement des roues est indépendante de l'erreur sur la position absolue de départ, de calculer la corrélation entre deux positions successives du Robot comme suit:

$$\begin{aligned}
\text{Cov}(P_k, P_{k+1}) & = E(\Delta P_k * \Delta P_{k+1}^T) \\
& = E[\Delta P_k * (J_{pk} * \Delta P_k + J_{dk+1} * \Delta d_{k+1})^T] \\
& = E[\Delta P_k * (\Delta P_k)^T * J_{pk}^T] \\
& = E[\Delta P_k * (\Delta P_k)^T] * J_{pk}^T = V(P_k) * J_{pk}^T
\end{aligned}$$

donc on a :

$$\text{Cov}(P_k, P_{k+1}) = V(P_k) * J_{pk}^T$$

Il est alors possible d'en déduire la corrélation existant entre la position initiale du Robot P_0 et sa position à l'instant $k+1$:

$$\text{Cov}(P_0, P_{k+1}) = E[\Delta P_0 * (\Delta P_{k+1})^T] = V(P_0) * \prod_{i=0}^k J_{pi}^T$$

b) Conclusion

Ce modèle produit une estimation de la position du Robot et de l'erreur sur cette position, il fournit également une mesure de la corrélation existant entre la position courante du Robot et sa position du moment du dernier recalage.

Ce modèle de l'odométrie est utilisé par le système de modélisation de l'environnement développé par P.MOUTARLIER [MOU 91] et qui fonctionne sur le Robot "JUNIOR" et sur le Robot "HILARE 2" avec l'adjonction d'un Gyroscope donnant le cap du Robot [PER 92].

7 RESULTAS DE SIMULATION

Comme outils de simulation nous avons utilisé le langage "turbo-pascal", le robot considéré est octogonale, le balayage des trois secteurs (SA-SD-SG) est assuré par six (06) sondes ultrasonores (deux sondes pour chaque secteur) placées sur un support tournant situé sur la partie la plus haute du robot, l'angle d'ouverture de ces sondes est de l'ordre de 30° , la vitesse maximale du robot est d'environ 1m/s.

Dans le cas de l'environnement représenté à la **FIGURE IV-21** les obstacles sont connus par le robot; Le planificateur génère l'itinéraire de S à D et le robot l'exécute en succès (**FIGURE IV-22**).

Un obstacle supplémentaire est ajouté à l'environnement où le robot est ordonné encore une fois d'aller de S à D; Le planificateur génère la trajectoire sans prendre en

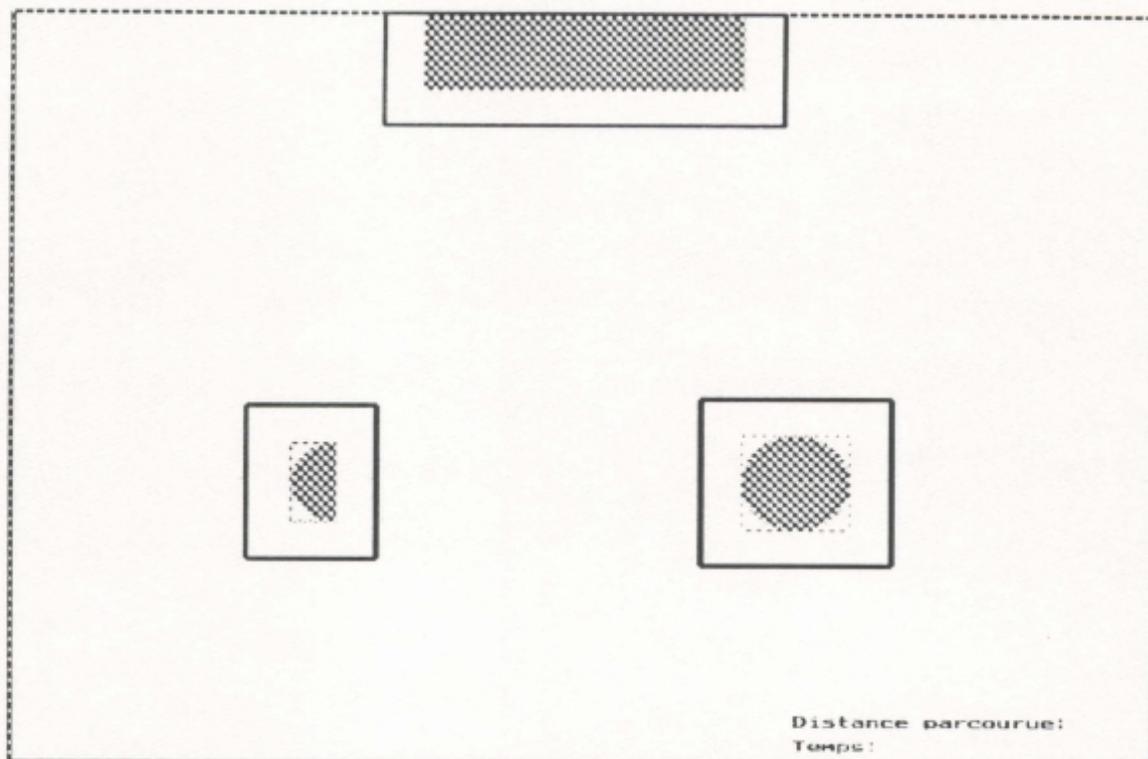


FIGURE IV-21: L'environnement test

considération la présence de l'obstacle supplémentaire, celui-ci est inconnu par le robot. Au démarrage et pendant l'exécution de sa trajectoire, le robot n'active que les sondes du secteur SA, à environ 60cm de l'obstacle supplémentaire le robot détecte sa présence, pour le contourner, il tourne à droite afin de prendre la direction de l'axe du repère Rr le plus proche à la destination et active les sondes du secteur SG pour le contrôle de la disparition de l'obstacle, le compteur d'obstacle CP est incrémenté à un "1". A l'extrême droite de l'obstacle, les sondes du secteur SG rendent un bilan positif, l'obstacle a donc disparu, le compteur est décrementé à zéro "0" et le planificateur réactualise la trajectoire à l'objectif. Le robot atteint sa destination en succès (**FIGURE IV-23**).

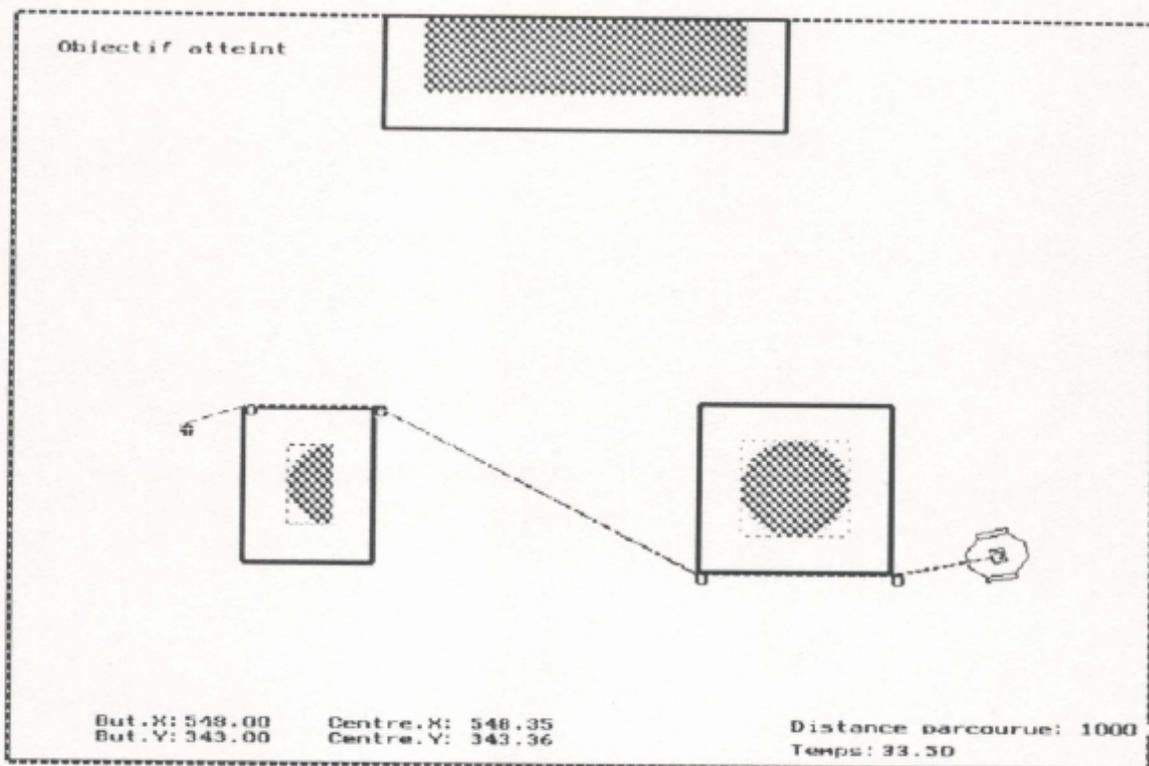


FIGURE IV-22: Trajectoire Optimale générée par le planificateur

8 CONCLUSION

L'approche géométrique, la notion des "instants de commutation", et la représentation des connaissances qu'on a développé permettent de réduire considérablement la complexité et le temps d'exécution du planificateur d'une part, et d'augmenter la sécurité du robot tout en assurant une convergence vers l'objectif d'autre part, et ceci au détriment de l'espace libre navigable diminué par les deux approximations représentant les obstacles; L'espace libre peut être optimisé par la méthode de partitionnement de l'espace par "QUATREE" développé dans [KAM 86][SAM 88] et [ZEL 92].

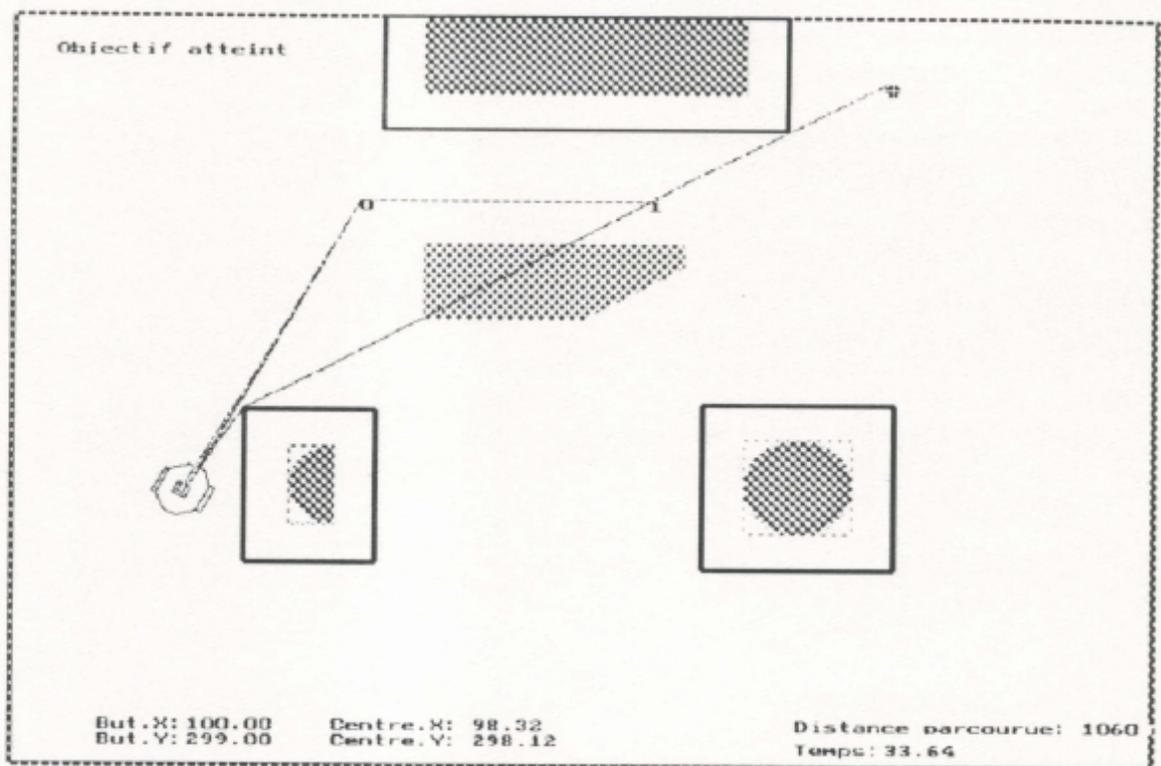


FIGURE IV-23: Evitement de l'obstacle imprevu et replanification de la trajectoire

CONCLUSION

Les travaux présentés dans ce mémoire, ont principalement porté sur la navigation et le guidage d'un robot mobile autonome; Avant d'évoquer les points forts de nos travaux, et de souligner les aspects qui mériteraient d'être approfondis, il convient de rappeler que la construction mécanique et la mise au point du support expérimental RoMoSA.I s'insère dans le cadre de nos travaux.

L'autonomie d'un robot mobile dépend beaucoup de ses capacités de perception et d'interprétation, mais elle provient surtout du système qui gère les facultés du robot. Ce système doit être apte à raisonner sur un modèle abstrait et idéalisé de son environnement, pour produire une séquence d'actions, permettant de faire aboutir la mission assignée au robot, contrôler l'exécution de cette séquence d'actions dans un contexte réel, détecter et réagir aux événements non planifiés, et assurer la reprise de la mission en cas d'échec.

Pour répondre à toutes ces exigences, nous avons mené nos travaux suivant quatre axes principaux:

- Elaboration d'une architecture hiérarchique
- Conception et mise au point d'un support expérimental
- Conception et développement du système de perception
- développement d'un planificateur réactif concrétisé par une simulation.

L'architecture hiérarchique élaborée est articulée autour de quatre sous systèmes:

Un superviseur qui génère les actions appropriées dans le but d'accomplir une tâche; Un planificateur qui prédetermine dans un univers connu l'itinéraire à suivre pour atteindre le but assigné; Un contrôleur d'exécution réactif qui tient compte du contexte réel et assure les réactions du

robot face aux événements non prévus par le planificateur, et enfin l'exécuteur qui contient tous les éléments opératoires du robot.

Le système de perception développé, essentiellement basé sur une étude physique très poussée sur les ultrasons, assure la détection de l'intrusion des obstacles dans la zone de sécurité du robot, et ceci malgré les aléas et l'imprécision des capteurs ultrasonores; Ce système a été principalement conçu, pour servir la partie réactive du planificateur développé autour d'une approche hybride géométrico-analytique.

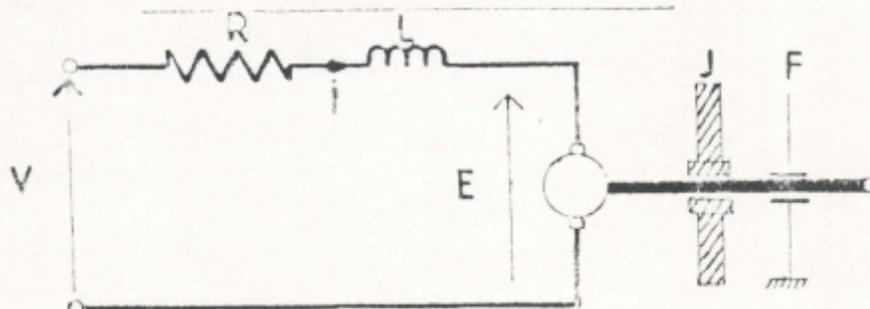
Nous voulons à travers cette étude, mettre au point un planificateur réactif, où la convergence au but est assurée, en s'appuyant sur une méthode géométrique très simplifiée de modélisation de l'environnement, combinée à une méthode analytique utilisant le critère des "instants de commutation" des accélérations, pour optimiser dans le sens "temps optimal" les trajectoires entre les buts partiels, et une simple détection d'obstacles sans avoir recours au système de mesure de temps de vol pour l'évaluation de distance aux obstacles; Ceci constitue l'originalité de nos travaux qui ont été validés par simulation. Nous allons maintenant aborder les points qui mériteraient d'être approfondis.

La perspective naturelle d'un robot mobile est de voir son autonomie s'accroître. Pour y parvenir, les directions d'extensions sont nombreuses, et les sujets qui traitent l'autonomie des robots mobiles sont loin d'être épuisés; Parmi ces sujets, nous citons en particulier le système de localisation qui n'a pas été abordé dans ce mémoire. La réalisation d'un robot réellement autonome reste donc un objectif de recherche passionnant, et nous espérons, par ce travail, avoir contribué à la réalisation d'un support expérimental ouvert et flexible.

ANNEXE A

A - 1) MODÉLISATION DU MOTEUR:

Soit le modèle électrique ci-dessous, d'un moteur à courant continu, on a:



$$\Gamma_m = K_r * i \quad (\text{couple moteur}) \quad (1)$$

$$E = K_E * \Omega \quad (\text{force contre électromotrice}) \quad (2)$$

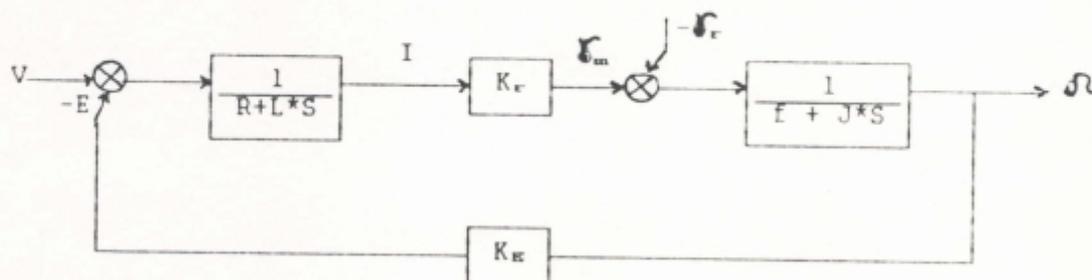
$$\Gamma_m - \Gamma_r = f * \Omega + J * \frac{d\Omega}{dt}, \quad (f + J * S) * \Omega = \Gamma_m - \Gamma_r \quad (3)$$

$$V - E = R * i + L * \frac{di}{dt}, \quad (R + L * S) I = V - E \quad (4)$$

on prend: $\Gamma_r = 0$

de 1 et 3 $K_r * i = f * \Omega + J * \frac{d\Omega}{dt}, \quad K_r * I = (f + J * S) * \Omega$

de 2 et 4 $V - K_E * \Omega = (R + L * S) * I$



$$\frac{\Omega}{V_t} = \frac{K_t}{K_t^2 + R * f * (1 + \frac{L}{R} * S) * (1 + \frac{J}{f} * S)}$$

avec: $K_t = K_E = K_r$

soit $\tau_e = L/R$ Constante de temps électrique

$\tau_m = J/f$ Constante de temps mécanique

$$\tau_m \gg \tau_e$$

$$\frac{\Omega(S)}{V_t(S)} = \frac{K_t}{K_t^2 + R*f*(\tau_e*\tau_m*S^2 + (\tau_e + \tau_m)*S + 1)}$$

or

$$\begin{aligned} \tau_e * \tau_m &\approx 0 \\ \tau_e + \tau_m &\approx \tau_m \end{aligned}$$

on pose: $K = K_t / (K_t^2 + R*f)$ on obtient: $\frac{\Omega(S)}{V_t(S)} = \frac{K}{1+r*S}$

$$r = J*R / (K_t^2 + R*f)$$

et: $\frac{\Theta(S)}{V_t(S)} = \frac{K}{S*(1+r*S)}$

A - 2) DISCRETISATION DU MODÈLE

Afin de pouvoir traiter les informations par un ordinateur, il est nécessaire de passer au domaine discret, pour cela, on a utilisé un échantillonneur bloqueur d'ordre zéro, sa fonction de transfert est donnée par:

$$B_0(S) = \frac{1 - \text{EXP}(-S*T)}{S} \quad \text{ou: } T \text{ est la période d'échantillonnage.}$$

La nouvelle fonction de transfert Y(S) est donnée par :

$$\begin{aligned} Y(S) &= B_0(S)*H(S) \\ &= \frac{K*[1 - \text{EXP}(-S*T)]}{S^2(1+r*S)} \end{aligned}$$

d'où:

$$\begin{aligned} y(t) &= -K*r*u(t) + K*r(t) + K*r*\text{EXP}(-t/r) + K*r*u(t-T) \\ &\quad - K*r(t-T) - K*r*\text{EXP}[-(t-T)/r] \end{aligned}$$

***Passage à la transformée en Z :

$$Y(Z) = -K\tau Z/(Z-1) + K^*T^*Z/(Z-1)^2 + K\tau^*Z/[Z-\text{EXP}(-T/\tau)] \\ + K\tau^*1/(Z-1) - K^*T^*1/(Z-1)^2 - K\tau^*1/[Z-\text{EXP}(-T/\tau)]$$

$$Y(Z) = \frac{b_1 * Z^{-1} + b_2 * Z^{-2}}{1 + a_1 * Z^{-1} + a_2 * Z^{-2}}$$

avec : $b_1 = K[T - \tau(1 - \text{EXP}(-T/\tau))]$

$$b_2 = K[\tau(1 - \text{EXP}(-T/\tau)) - T * \text{EXP}(-T/\tau)]$$

$$a_2 = \text{EXP}(-T/\tau)$$

$$a_1 = -(1 + a_2)$$

Le système dans son évolution subit des perturbations susceptibles d'affecter les paramètres du système (a_1, a_2, b_1 et b_2); Le signal de commande fonction de (a_1, a_2, b_1 , et b_2) doit réagir à ces perturbations pour éviter toutes instabilités intolérables, la connaissance exacte de ces paramètres à tout instant est donc nécessaire.

A - 3) IDENTIFICATION DES PARAMÈTRES DU SYSTÈME

1-L'estimateur RLS (Recursive Least Square)

Une équation Réursive peut être déterminée à partir de la fonction de transfert

$Y(Z)$:

$$\frac{\Theta(Z)}{V_t(Z)} = \frac{b_1 * Z^{-1} + b_2 * Z^{-2}}{1 + a_1 * Z^{-1} + a_2 * Z^{-2}}$$

$$\Theta(Z) + a_1 * \Theta(Z) * Z^{-1} + a_2 * \Theta(Z) * Z^{-2} = b_1 * Z^{-1} * V_t(Z) + b_2 * Z^{-2} * V_t(Z)$$

***Passage au domaine temporelle :

$$\Theta(k) + a_1 * \Theta(k-1) + a_2 * \Theta(k-2) = b_1 * V(k-1) + b_2 * V(k-2)$$

or: $a_1 = -(1 + a_2)$

$$\Theta(k) - \Theta(k-1) - a_2 * \Theta(k-1) + a_2 * \Theta(k-2) = b_1 * V(k-1) + b_2 * V(k-2)$$

$$\Theta(k) - \Theta(k-1) = a_2 * [\Theta(k-1) - \Theta(k-2)] + b_1 * V(k-1) + b_2 * V(k-2)$$

on pose: $\epsilon(k) = \Theta(k) - \Theta(k-1)$

on obtient:

$$\epsilon(k) = a_2 * \epsilon(k-1) + b_1 * V(k-1) + b_2 * V(k-2)$$

Les équations de l'estimateur RLS sont dans ce cas :

$$\Theta(k) = \Theta(k-1) + K(k-1) * [Y(k) - \Phi^T(k) * \Theta(k-1)]$$

ou $\Theta(k)$ est le vecteur paramètre à estimer

$$\Theta(k) = [a_2(k), b_1(k), b_2(k)]^T$$

$$y(k) = \epsilon(k) \quad \dots \text{c'est un scalaire.}$$

$\Psi(k)$... est un vecteur des entrées-sorties mesurées

$$\Psi(k) = [\epsilon(k-1), V(k-1), V(k-2)]^T$$

La matrice de correction K est définie comme suit :

$$K(k-1) = \frac{P(k-1) * \Psi(k)}{[\lambda + \Phi^T(k) * P(k-1) * \Psi(k)]}$$

avec:

$$P(k) = \frac{[I - K(k-1) * \Psi^T(k)] * P(k-1)}{\lambda}$$

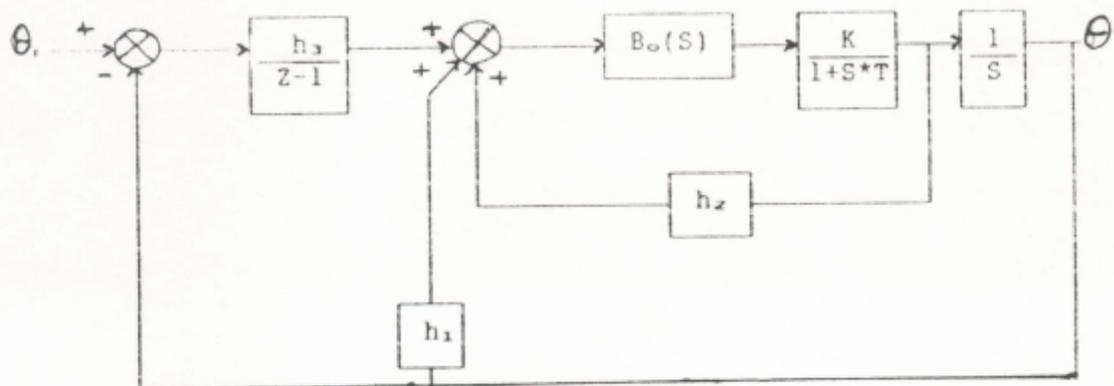
La matrice P est proportionnelle à la variance des estimations; Sa valeur initiale est $\alpha * I$ ou α est une valeur arbitraire très élevée.

A - 4) CONCEPTION DU CONTRÔLEUR

A - 4 - 1 CONTROLEUR À RETOUR D'ÉTAT

Un contrôleur PID peut être utilisé pour implémenter une servo-commande de position. Ce contrôleur peut être pris comme contrôleur à retard d'état ou les états sont: la position θ ; la vitesse angulaire w , et l'intégrale de l'erreur de position $(\theta_r - \theta)$.

La référence θ_r est appliquée uniquement à l'intégrale, cette structure est appelée "Set-Point-ON-I-ONLY".



1-Calcul de la fonction de transfert.

$$\left[\frac{(\Theta_r - \Theta) \cdot h_3}{Z-1} + w \cdot h_2 + \Theta \cdot h_1 \right] \cdot \left[\frac{b_1 \cdot Z^{-1} + b_2 \cdot Z^{-2}}{1 + a_1 \cdot Z^{-1} + a_2 \cdot Z^{-2}} \right] = \Theta$$

$$\left[\frac{(\Theta_r - \Theta) \cdot h_3}{Z-1} + h_2 \cdot w + h_1 \cdot \Theta \right] \cdot \left[\frac{b_1 \cdot Z + b_2}{Z^2 + a_1 \cdot Z + a_2} \right] = \Theta \dots\dots 1$$

d'autre part ,on a:

$$Z^2 + a_1 \cdot Z + a_2 = (Z - a_2) \cdot (Z - 1)$$

et $\frac{w(Z)}{V(Z)} = \frac{K \cdot [1 - \text{EXP}(-T/\tau)]}{[Z - \text{EXP}(-T/\tau)]}$

$$\frac{\Theta(Z)}{V(Z)} = \frac{b_1 \cdot Z + b_2}{(Z-1) \cdot (Z-a_2)}$$

$$w(Z) = \frac{K \cdot (1-a_2) \cdot (Z-1)}{b_1 \cdot Z + b_2} \cdot \Theta(Z)$$

on remplace w(Z) par sa valeur dans 1 et de là, on obtient:

$$\frac{\Theta}{\Theta_r} = \frac{h_3 \cdot b_1 \cdot (Z + b_2/b_1)}{Z^3 - C_1 \cdot Z^2 + C_2 \cdot Z - C_3}$$

avec : $C_1 = h_1 \cdot b_1 + h_2 \cdot K \cdot (1-a_2) + 2 + a_2$

$$C_2 = (b_1 - b_2) \cdot h_1 + 2 \cdot K \cdot (1-a_2) \cdot h_2 + h_3 \cdot b_1 + 1 + 2 \cdot a_2$$

$$C_3 = K \cdot (1-a_2) \cdot h_2 + a_2 - b_2 \cdot (h_3 - h_1)$$

A - 4 - 2 LE CONTRÔLEUR P.D.

L'introduction d'un intégrateur dans un contrôleur élimine l'erreur statique en régime permanent; Il a par conséquent une légère influence sur la précision; La complexité du contrôleur peut être réduite en utilisant un contrôleur P.D.

1-Représentation du moteur DC dans l'espace d'état discret:

$$H_\Theta(S) = \frac{K \cdot [1 - \text{EXP}(-S \cdot T)]}{S^2 \cdot (1 + r \cdot S)} = \frac{\Theta(S)}{V(S)}$$

$$H_w(S) = \frac{w(S)}{V(S)} = \frac{K \cdot [1 - \text{EXP}(-S \cdot T)]}{S \cdot (1 + r \cdot S)}$$

$$\xrightarrow{T \cdot Z} H_w(Z) = K \cdot \left[\frac{Z^{-1} \cdot [1 - \text{EXP}(-T/\tau)]}{1 - \text{EXP}(-T/\tau) \cdot Z^{-1}} \right] = \frac{w(Z)}{V(Z)}$$

$$w(Z) - \text{EXP}(-T/\tau) * w(Z) * Z^{-1} = K * [1 - \text{EXP}(-T/\tau)] * V(Z) * Z^{-1}$$

$$\xrightarrow{\text{T.Z.I}} w(k) - \text{EXP}(-T/\tau) * w(k-1) = K * [1 - \text{EXP}(-T/\tau)] * V(k-1)$$

$$w(k+1) = \text{EXP}(-T/\tau) * w(k) + K * [1 - \text{EXP}(-T/\tau)] * V(k) \dots 1$$

d'autre part , on a:

$$H_{\theta}(S) = \frac{K * [1 - \text{EXP}(-S * T)]}{S^2 * (1 + r * S)}$$

$$\xrightarrow{\frac{\text{T.Z}}{H_{\theta}(Z)}} H_{\theta}(z) = \frac{K * T}{z - 1} - K * r * \left[1 + \frac{1 - z}{z - \text{EXP}(-T/\tau)} \right]$$

$$H_{\theta}(Z) = \frac{K * T}{Z - 1} - r * H_w(Z)$$

$$H_{\theta}(Z) + r * H_w(Z) = \frac{K * T * Z^{-1}}{1 - Z^{-1}}$$

$$\frac{\Theta(Z) + r * w(Z)}{V(Z)} = \frac{K * T * Z^{-1}}{1 - Z^{-1}}$$

$$\xrightarrow{\text{T.Z.I}} \Theta(k) + r * w(k) - \Theta(k-1) - r * w(k-1) = K * T * V(k-1)$$

$$\Theta(k+1) + r * w(k+1) - \Theta(k) - r * w(k) = K * T * V(k)$$

$$\Theta(k+1) = \Theta(k) + r * w(k) - r * [\text{EXP}(-T/\tau) * w(k) + K * [1 - \text{EXP}(-T/\tau)] * V(k)] + K * T * V(k)$$

$$\Theta(k+1) = \Theta(k) + r * [1 - \text{EXP}(-T/\tau)] * w(k) + K * [T - r * [1 - \text{EXP}(-T/\tau)]] * V(k)$$

On obtient donc le système suivant:

$$\left[\begin{array}{l} \Theta(k+1) = \\ \Theta(k) + r * [1 - \text{EXP}(-T/\tau)] * w(k) + K * [T - r * [1 - \text{EXP}(-T/\tau)]] * V(k) \\ w(k+1) = \text{EXP}(-T/\tau) * w(k) + K * [1 - \text{EXP}(-T/\tau)] * V(k) \end{array} \right.$$

On pose: $a_1 = r * [1 - \text{EXP}(-T/\tau)]$

$$a_2 = \text{EXP}(-T/\tau)$$

$$a_3 = K * [T - r * [1 - \text{EXP}(-T/\tau)]]$$

$$a_4 = K * [1 - \text{EXP}(-T/\tau)]$$

d'où:

$$\begin{bmatrix} \theta(k+1) \\ w(k+1) \end{bmatrix} = \begin{bmatrix} 1 & a_1 \\ 0 & a_2 \end{bmatrix} * \begin{bmatrix} \theta(k) \\ w(k) \end{bmatrix} + \begin{bmatrix} a_3 \\ a_4 \end{bmatrix} * V(k)$$

2-Implémentation du régulateur P.D.:

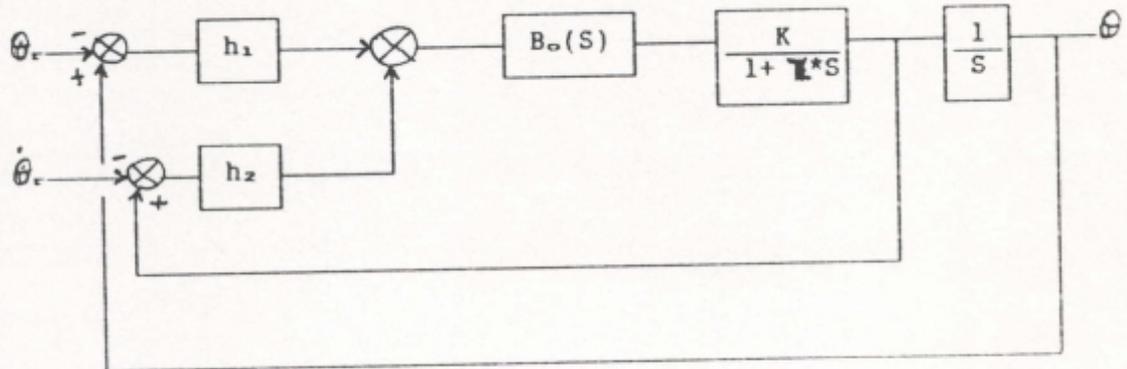


FIGURE A-4

Diagramme bloc du contrôleur P.D.

h_1 et h_2 sont les paramètres du régulateur.

Le signal de commande de la boucle fermée représentée à la FIGURE A-4 est donnée par:

$$V = h_1 * (\theta - \theta_r) + h_2 * (w - \dot{\theta}_r)$$

3-Calcul de la fonction de transfert du système en B.F:

On a:

$$[(\theta - \theta_r) * h_1 + (w - \dot{\theta}_r) * h_2] * \left[\frac{b_1 * Z + b_2}{(Z - a_2) * (Z - 1)} \right] = \theta$$

$$(\theta - \theta_r) * h_1 + (w - \dot{\theta}_r) * h_2 = \frac{(Z - a_2) * (Z - 1)}{b_1 * Z + b_2} * \theta$$

d'autre part on a vu précédemment que:

$$\frac{\theta(Z)}{w(Z)} = \frac{b_1 * Z + b_2}{K * (1 - a_2) * (Z - 1)}$$

d'où:

$$w(Z) = \frac{K(1 - a_2)(Z - 1) * \Theta(Z)}{b_1 * Z + b_2}$$

d'où:

$$\frac{\Theta}{\Theta_r} = \frac{-Z(h_2 * a_4 + h_1 * a_3) + h_1(a_3 * a_2 - a_1 * a_4) + h_2 * a_4}{Z^2 - Z(h_1 * a_3 + h_2 * a_4 + a_2 + 1) + h_1(a_3 * a_2 - a_1 * a_4) + h_2 * a_4 + a_2}$$

4-Placement des Pôles:

L'équation caractéristique du système à boucle fermée est donnée par:

$$D(Z) = Z^2 - Z(h_1 * a_3 + h_2 * a_4 + a_2 + 1) + h_1(a_3 * a_2 - a_1 * a_4) + h_2 * a_4 + a_2$$

$$\left[\begin{array}{l} a_1 = r * [1 - \text{EXP}(-T/\tau)] \\ a_2 = \text{EXP}(-T/\tau) \\ a_3 = b_1 = K * [T - r * [1 - \text{EXP}(-T/\tau)]] \\ a_4 = K(1 - a_2) = K * [1 - \text{EXP}(-T/\tau)] \\ b_2 = a_1 * a_4 - a_3 * a_2 \end{array} \right.$$

Pour placer les pôles (ξ et w_m donnés), il suffit de comparer le polynôme

caractéristique $D(Z)$ avec le polynôme caractéristique désiré donné par:

$$(Z - P_1)(Z - P_1) = Z^2 - Z(P_1 + P_1) + P_1 * P_1$$

où:

$$(P_1 + P_1) = 2 * [\text{EXP}(-\xi * w_m * T)] * [\cos(w_m * T * (1 - \xi^2)^{1/2})]$$

$$P_1 * P_1 = \text{EXP}(-2 * \xi * w_m * T)$$

on obtient:

$$P_1 + P_1 = a_2 + a_3 * h_1 + a_4 * h_2 + 1 = 2 * [\text{EXP}(-\xi * w_m * T)] * [\cos(w_m * T * (1 - \xi^2)^{1/2})]$$

$$P_1 * P_1 = a_2 * (1 + a_3 * h_1) + a_4 * (h_2 - a_1 * h_1) = \text{EXP}(-2 * \xi * w_m * T)$$

d'où:

$$\left[\begin{array}{l} h_1 = \frac{2 * [\text{EXP}(-\xi * w_m * T)] * [\cos(w_m * T * (1 - \xi^2)^{1/2})] - \text{EXP}(-2 * \xi * w_m * T) - 1}{a_3 * (1 - a_2) + a_4 * a_1} \\ h_2 = \frac{\text{EXP}(-2 * \xi * w_m * T) - a_2 - (a_2 * a_3 - a_4 * a_1) * h_1}{a_4} \end{array} \right.$$

et le signal de commande est donné par:

$$v = h_1 * (\Theta - \Theta_r) + h_2 * (w - \dot{\Theta}_r)$$

On voit clairement que si le système subit des perturbations affectant les paramètres a_i , les paramètres h_i compensent l'effet de cette perturbation pour maintenir l'emplacement des pôles.

A - 5) CONCLUSION

Une partie de ce travail a été inspirée des travaux de LOUISA.DESSAINT and AL [LOU 90].

REFERENCES

[ALB 87] J.S.ALBUS, H.G.McCAIN et R.LUMIA, "NASA/NSB Standard reference model for telerobot control system architecture (NASREM)", Rapport technique NSB, Technical note 1235, US département of commerce National bureau of standards, 1987.

[BOR 88] J.BORENSTEIN and Y.KOREN, "Obstacle Avoidance with Ultrasonic Sensors", IEEE J.Robotics Automat., Vol.RA-4,NO.2.,pp.213-218, 1988.

[BOR 89] -----., "Real Time Obstacle Avoidance for Fast Mobile Robots", IEEE Trans.Syst. Man Cybern., Vol.19.,NO.5., pp.1179-1187., Sept/Oct.1989.

[BOR 91a] -----., "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots", IEEE Trans.Robotics Automat., Vol.7, pp.278-288, June 1991.

[BOR 91b] -----., "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance", IEEE Trans.Robotics Automat.Vol.7, NO.4., pp.535-539., August 1991.

[BRO 83] R.A.BROOKS, "Solving The Find-Path Problem by Good Representation of Free Space", IEEE Trans.Syst.Man Cyber., Vol.SMC-13, NO.3.PP.190-197. Mar/Apr 1983.

[BRO 85] R.A.BROOKS, "A layered intelligent control system for mobile robot", In Robotics Research: the third international symposium, GOUVIEUX (FRANCE), 1985.[BUR 92]

[BRO 86] R.A.BROOKS et J.H.CONNELL, " Asynchronous distributed control system for a mobile robot. SPIE Mobile Robot, Vol 727, pp 77-83, 1986.

[BUR 92] P.BURLINA, D.DeMENTHON and LARRY.S.DAVIS., "Navigation with Uncertainty: Reaching a Goal in High Collision Risk Region", IEEE in proc.Int.Conf.Robotics Automat., NICE-FRANCE, May 1992, pp.2440-2445

[CHO 91] H.CHOCHON, "Object oriented design of mobile robot control system", In second international symposium on experimental robotics, juin 1991.

[COX 91] I.J.COX, "Blanche--An experiment in guidance and navigation of an autonomous robot vehicle", In IEEE Trans. on robotics and autom., Vol.7, NO.2, pp 193-204, APRIL 1991.

[CRO 85] J.L.CROWLEY, "Navigation of an Intelligent Mobile Robot," IEEE J.Robotics Automat., Vol.RA-1, NO.1, pp.31-41, Mar 1985.

[CRO 89] J.L.CROWLEY, "World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging", Tech.Report. LIFIA(IMAG) GRENOBLE-FRANCE, Feb 1989.

[CRO 91] J.L.CROWLEY, P.REIGNIER, O.CAUSSE, et F.WALLNER, "Action level behaviours for locomotion and perception", In International symposium on experimental robotics, GRENOBLE, MAY 1991.

[ELF 83] A.ELFES et S.N. TALUKDAR, "A distributed control system for the CMU ROVER", In 8th international joint conference on artificial intelligence (IJCAI), KARLSRUHE (RFA), 1983.

[FOU 93] G.FOUX, M.HEYMANN, and A.BRUCKSTEIN, "Two-Dimensional Robot Navigation Among Unknown Stationary Polygonal Obstacles", IEEE Trans.Robotics Automation., Vol.9, NO.1, Feb 1993.

- [GIR 83] G.GIRALT, R.CHATILA et M.VAISSET, "An integrated navigation and motion control system for autonomous multisensory mobile robots", In robotics research: the first international symposium, BRETON WOODS (NEW HAMPSHIRE, USA), 1983.
- [GRI 90] N.C.GRISWOLD and J.EEM, "Control for Mobile Robots in the Présence of Moving Objects", IEEE Trans.Robotics Automat., Vol.6, NO.2, pp.263-268, April 1990.
- [KAM 86] S.KAMBHAMPATI et L.S.DAVIS, "Multiresolution path planning for mobile Robot", IEEE J.Robotics. Automat., Vol.RA-2, NO.3, pp 135-145, SEPT. 1986.
- [KEI 84] D.M.KEIRSY et al., "Algorithm of navigation for a mobile Robot", In Proc. 1st Int.Conf.Robotics, ATLANTA, GA, MARS 1984, pp.574-583.
- [KHA 85] O.KHATIB, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", In Proc.IEEE Int.Conf.Robotics Automat. (St LOUIS, USA, Mar.25-28, 1985), pp.500-505.
- [LOU 90] L.A.DESSAINT, B.J.HEBERT, H.LE-HUY, et G.CAVUOTI, "A D.S.P-Based adaptative controller for a smooth positioning system", In IEEE Trans.on Indistr.Electronics, Vol.37, NO.5, pp 372-377, OCTOBER 1990.
- [LOZ 79] T.LOZANO-PEREZ and M.A.WESLEY, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles", Communication. A.C.M, Vol.22, NO.10, pp.560-570, Oct 1979.
- [LYE 86] S.S.LYENGAR, N.S.V.RAO, C.C.JORGENSEN, and C.R.WEISBIN, "Robot Navigation In an Explored Terrain", J.Robotic Syst, Vol.3, NO.4, pp.389-407, 1986.
- [MIT 87] J.S.B.MITCHELL, D.W.PAYTON et D.M.KEIRSY, "Planning and reasoning for autonomous vehicule control", International journal of intelligent systems, Vol II, pp 129-198, 1987.

[MOR 85] H.P.MORAVEC et A.ELFES, "High resolution maps from wide angle sonar", In Proc.IEEE Conf.Robotics.Automat., (WASHINGTON D.C.), 1985, pp 116-121.

[MOR 88] H.P.MORAVEC, "Sensor fusion in certainty grids for mobile Robots", Al Mag., pp 61-74, Summer 1988.

[MOU 91] P.MOUTARLIER, "Modélisation Autonome de l'environnement par un Robot Mobile", Thèse de l'université PAUL SABATIER, TOULOUSE (FRANCE) 91381, LAAS(CNRS), OCTOBRE 1991.

[Nii 86a] H.P.Nii, "Blackboard system. Blackboard application systems. Blackboard systems from a knowledge engineering perspective", artificial intelligence magazine, August 1986.

[Nii 86b] H.P.Nii, "Blackboard system: the blackboard model of problem solving and the evolution of blackboard architectures", Artificial intelligence magazine, Summer 1986.

[PAY 86] D.W.PAYTON, "An architecture for reflexive autonomous vehicle control", In IEEE International conference on robotics and automation, SAN FRANCISCO (USA), pp 1838-1845, 1986.

[PER 92] V.O.PEREBASKINE, "Une architecture modulaire pour le controle d'un robot mobile autonome", Thèse de doctorat: présentée au LAAS du CNRS, TOULOUSE, AVRIL 1992.

[PON 86] L.S.PONTRYAGAIN et al., "Theorie mathématique des processus optimaux", Edition MOSCOW 1986.

[PRE 88] W.H.PRESS et al, "Numerical Recipes in C: The Art of Scientific Computing". CAMBRIDGE.UK: Cambridge University Press.1988.

[SAM 88] H.SAMET, "An Overview of Quatrees, Octrees, and Related Hierarchical Data Structures", In NATO ASI Series, Vol.F40. Theoretical Foundations of Computer Graphics. BERLIN, Springer-Verlag, pp.51-68, 1988.

[SUZ 88] H.SUZUKI et S.ARIMOTO, "Visual Control of Autonomous Mobile Robot Based on Self-Organizing Model for Pattern learning", Journal of Robotics Systems, Vol.5, NO.5, pp 453-470, MAY 1988.

[THO 86] C.THORPE, M.HEBERT, T.KANADE et S.SHAFFER, Year end report for road following robot, "Section III: Vision and navigation for the CARNEGIE MELLON NAVLAB, pp 16-42, Rapport technique CMU-RI-TR-87-11, CMU, 1986.

[ZEL 92] A.ZELINSKY, "A Mobile Robot Exploration Algorithm", IEEE Trans. Robotics Automat., Vol.8, NO.6, pp.707-717, Dec 1992.