

13/95

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

المكتبة - المكتبة  
E.N.P. ALGER  
BIBLIOTHEQUE - المكتبة  
Ecole Nationale Polytechnique

DEPARTEMENT GENIE ELECTRIQUE

Option Automatique

## *PROJET DE FIN D'ETUDES*

Pour l'obtention du diplôme  
d'Ingénieur d'Etat en Automatique

Thème

# **ETUDE ET SYNTHESE D'UN NAVIGATEUR CLASSIQUE ET FLOU D'UN ROBOT MOBILE**

Proposé et dirigé par:

**Dr. M. C. SOUAMI.**

PRESENTE PAR:

**Mlle Z. DROUCHE.**

**JUILLET 95**

10, Avenue Hassen Badi El-Harrach Alger

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE



DEPARTEMENT GENIE ELECTRIQUE

Option Automatique

## ***PROJET DE FIN D'ETUDES***

Pour l'obtention du diplôme  
d'Ingénieur d'Etat en Automatique

**Thème**

# **ETUDE ET SYNTHÈSE D'UN NAVIGATEUR CLASSIQUE ET FLOU D'UN ROBOT MOBILE**

Proposé et dirigé par:

Dr. M. C. SOUAMI.

PRESENTE PAR:

Mlle Z. DROUCHE.

**JUILLET 95**

10, Avenue Hassen Badi El-Harrach Alger

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

*à ma famille*

*à tous ceux qui me sont chers*

# REMERCIEMENTS

المدرسة الوطنية المتعددة التخصصات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

*Je tiens à exprimer mes remerciements et ma plus vive gratitude:*

*à monsieur M. C. Souami pour avoir accepté de m'encadrer durant mon sujet de fin d'étude et qui m'a fait découvrir la robotique.*

*à monsieur D. Ouelhadj pour son soutien et l'aide précieuse qu'il m'a apporté tout le long de ce travail.*

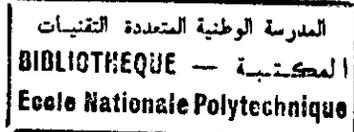
*à Madame A. SERIR pour la documentation qu'elle m'a fourni.*

*à M. HADJ NACER pour sa gentillesse*

*à ma famille qui a tout mis à ma disposition.*

*Et enfin à tous mes amis qui m'ont aidé de leur soutien et leur présence.*

## RESUME



Le travail présenté dans ce mémoire s'inscrit dans le cadre de recherches sur le problème de la navigation des robots mobiles autonome. Pour cela, il est nécessaire de doter le robot d'un contrôle intelligent capable de l'habiller à ce diriger vers une cible donnée dans son environnement, et à atteindre d'une manière autonome, sans risque de collision.

Nous avons concrétiser ceci, par la conception d'un contrôleur flou que nous avons tester en simulation et a donné des résultats plus satisfaisant.

**Mot clés:** navigation, évitement d'obstacle, robot mobile, environnement dynamique, planification de trajectoire, modélisation de l'environnement, logique floue.

# SOMMAIRE

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

## INTRODUCTION GENERALE

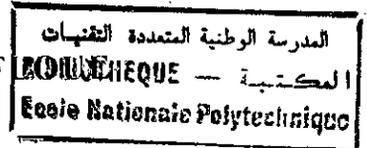
### CHAPITRE I: LES ROBOTS MOBILES

1. Définition d'un robot mobile 5
2. Architecture des robots mobiles 5
3. Définition du problème de navigation 6
  - 3.1. Navigation dans un environnement connu 6
  - 3.2. Navigation dans un environnement inconnu 6
    - 3.2.1. Navigation sans mémorisation de l'environnement 7
    - 3.2.2. Navigation avec mémorisation de l'environnement 7
4. Perception de l'environnement 7
- Conclusion 8

### CHAPITRE II: MODELISATION ET PLANIFICATION DE LA TRAJECTOIRE.

1. Position du problème 9
2. Modélisation de l'environnement (état de l'art) 10
  - 2.1. Le modèle par grille 10
  - 2.2. Le modèle polygonal 11
3. Planification globale de la trajectoire 12
  - 3.1. Génération de trajectoires par un modèle par grille 12
  - 3.2. Génération de trajectoire par un modèle polygonale 13
    - 3.2.1. Méthode de NILSSON 13
    - 3.2.2. Méthode de LOZANO - PEREZ 14
    - 3.2.3. Méthode de RAJA - CHATILA 16
    - 3.2.4. Méthode des cônes généralisé (BROOKS) 17
4. Planification locale de la trajectoire 18
5. Discussion 20

## CHAPITRE III: NOTION DE BASE SUR LA LOGIQUE FLOUE



Introduction 22

1. Ensembles flous et logique floue 23

1.1. Ensembles flous et terminologie 23

1.2. Opération théorique d'ensemble 24

1.3. Variable linguistique 24

1.4. Logique floue et le raisonnement approximatif 25

2. Organisation d'un système flou 26

2.1. Définition du système d'entrée et de sortie du contrôleur 27

2.2. Fuzzification des entrées 28

2.3. Règles d'évaluation et la logique de prise de décision 30

2.4. Défuzzification des sorties 31

Conclusion 33

## CHAPITRE IV: LA METHODE D'EVITEMENT D'OBSTACLES PAR LA LOGIQUE FLOUE

1. Principe 34

2. Algorithme de la méthode 35

2.1. Décision du degré de danger statique 38

2.2. Décision du degré de danger dynamique 43

## CHAPITRE V: SIMULATION

### CONCLUSION GENERALE

### ANNEXE

### BIBLIOGRAPHIE

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

*INTRODUCTION  
GENERALE*

Depuis longtemps l'homme a été fasciné par les contes et histoires mettant en scène des objets possédant un comportement et un raisonnement comparables à celui de l'homme.

Le mot ROBOT fût inventé en 1920 par l'écrivain tchèque KAREL CAPEK pour désigner des machines qui travaillent infatigablement, il dérive du mot tchèque ROBOTA signifiant travail forcé [LEE].

Les premiers travaux en robotique ont débuté vers les années 40 avec les manipulateurs mécaniques et ce n'est que vers les années 50 que ces derniers sont rendus plus sophistiqués c'est à dire programmables par langage symbolique à travers un ordinateur. Les premières recherches sur les robots mobiles commencèrent à partir des années 60.

Deux projets américains, SHAKEY (1968) au SRI (Stanford Research Institute) et HEXAPOD à Ohio State University, sont considérés comme les pionniers représentatifs de deux orientations majeurs: l'orientation intelligence artificielle pour le premier, et la problématique de la locomotion tout terrain avec architecture mécanique à pattes pour le second. Cet intérêt pour la mobilité des robots se justifie par les applications qu'elles peuvent apporter dans de nombreux domaines. En effet on apprécie l'utilité des robots mobiles quand il peuvent aider les handicapés moteurs (automatisation des fauteuils roulants) ou bien remplacer l'homme dans les tâches qui lui sont monotones ou fatigantes (dans l'industrie métallurgique, automobile, électrique, etc...), soit dangereuse en milieu terrestre (industrie nucléaire, chimique, des mines, lutte contre l'incendie, etc...) soit inaccessible comme dans le domaine spatial (exploration des planètes) [CAM 90].

Au départ, les robots mobiles étaient téléguidés ou encore dépendaient d'une certaine installation placée dans leur environnement (lignes électriques ou magnétiques placées sur le sol). Pour certaines applications (industrielles par exemple), ce type de mobilité était satisfaisant et apprécié, mais il n'est pas toujours possible de placer une quelconque installation guidant le déplacement du robot (exemple: les domaines marin, spatial ou militaire). C'est pourquoi aujourd'hui les chercheurs visent la conception et la réalisation de robots mobiles totalement autonomes en prise avec un environnement réel souvent complexe.

Par autonomie, on entend la capacité donnée au robots de décider et de s'adapter à des situations nouvelles dans un contexte donné. Sans cette capacité, le robot apparaît comme un automate qui ne peut exécuter que des séquences d'actions pour les quelles il était programmé.

Les robots mobiles autonomes constituent pour la recherche l'exemple parfait de robots de quatrième génération définis comme étant des machines dotées de capacité de relation intelligente entre la perception de l'environnement de travail et les actions à exécuter. Cette fonction de perception est très importante vu que grâce à elle, le robot découvre l'espace dont lequel il évolue. Il peut alors décider des actions à effectuer et réagir à des situations imprévues [TOU 91]. Cependant, la connaissance de l'environnement n'est pas suffisante. En effet, un robot mobile autonome qui doit produire des actions est amené à acquérir des informations sur son environnement pour en construire des modèles et exécuter des tâches [TOU 91]. C'est l'élaboration de planification de trajectoire qui permet de résoudre ce problème.

Lorsque les premières recherches sur la planification de trajectoire pour robots mobiles commencèrent, on s'intéressa au cas de robots pouvant se déplacer dans n'importe quelle direction à partir d'un point donné et à un instant donné. Aujourd'hui, le problème lié à la

donné et à un instant donné. Aujourd'hui, le problème lié à la planification de trajectoire pour des robots sans contraintes cinématiques est bien formulé et de nombreux résultats intéressants ont été obtenus.

Le travail présenté dans ce mémoire s'inscrit dans cette problématique de la planification de trajectoire pour un robot mobile holonome évoluant dans un environnement encombré d'obstacles en lui fixant simplement la position à atteindre.

Cette trajectoire sera générée grâce à un planificateur global. Qu'est-ce qu'un planificateur global? Il existe deux grandes familles de planificateurs: les planificateurs locaux et les planificateurs globaux.

Une planification locale est utilisée lorsque l'espace d'évolution du robot lui est totalement inconnu. Donc, vu qu'aucune information n'est disponible a priori sur l'environnement, il ne peut être toujours question de trajectoire optimale. Ceci nous a conduit à la deuxième famille de planificateurs à savoir les planificateurs globaux. Contrairement aux planificateurs locaux, l'espace d'évolution du robot est connu ou préalable ce qui permet, après une phase de traitement (modélisation) de générer des trajectoires optimales.

Le premier chapitre est consacré à la présentation des robots mobiles afin de positionner le travail à réaliser. Les chapitres qui suivent abordent les différentes parties composant ce dernier.

Dans le deuxième chapitre sont présentées les différentes techniques de planification de trajectoire sans collisions pour un robot mobile, en donnant les avantages et inconvénients de chacune d'elles.

Dans notre travail nous nous inspirons des règles qu'utilisera un conducteur de voiture, en fait celui-ci ne fait aucun calcul de trajectoire. Il ne fait qu'estimer sa position par rapport à sa cible et aux obstacles qui l'entourent ainsi que les commandes à effectuer.

Il se trouve que la théorie des ensembles floue formalise ce type de commande. Elle constitue l'outil adéquat et assez flexible pour faire face aux catégories linguistiques du protocole de contrôle. La logique floue est un pas vers un rapprochement entre la précision des mathématiques classiques et l'imprécision du monde réel.

Dans le troisième chapitre sont présentées quelques notions des ensembles flous ainsi que la logique qui manipule ces ensembles.

Dans le quatrième chapitre est décrite la méthode d'évitement d'obstacles par la logique floue, ceci sera illustré par une simulation qui validera notre travail.

Enfin, une conclusion générale termine notre mémoire donnant les éventuelles perspectives qui pourront améliorer notre travail ainsi qu'une bibliographie qui facilitera une étendue plus approfondie des différents domaines abordés.

*Chapitre I*

*LES ROBOTS MOBILES*

## INTRODUCTION:

Les robots mobiles, sont de création récente; les recherches dans ce domaine ont débuté avec le robot mobile SHAKEY développé au Etats-Unis (Standford Research institute) et jusqu'à la fin des années 70, il n'y avait environ qu'une dizaine de projets de robots mobiles à travers le monde [PLA 83a]. Certains d'entre eux n'ont pu aboutir à des réalisations par suite de difficultés technologiques (non évolution de la technologie des capteurs et de la micro-informatique). Mais, dès la fin des années 70, les études ont néanmoins repris avec le robot HILARE (Toulouse-France), VESA (Rennes/France), YAMA BIKO, MEL-DEIC (Japon), CMU-ROVER (USA) [REC 85]... et depuis les recherches dans ce domaine n'ont pas cessé de s'accroître.

### 1. DEFINITION D'UN ROBOT MOBILE.

Contrairement au véhicule guidé, un Robot mobile est un véhicule autonome doté de capacités de relation intelligente entre la perception de l'environnement de travail et les actions à exécuter. Ces actions se manifestent par la modélisation de l'environnement pour sa compréhension, la planification de trajectoire et le contrôle de mouvements [GIR 83].

Ces fonction de perception, décision et action constituent le degré d'autonomie d'un robot mobile.

### 2. ARCHITECTURE DES ROBOTS MOBILES: [JUL 84]

Leur forme la plus courante est la forme rectangulaire ou carrée, mais plusieurs présentent une forme circulaire ou intermédiaire entre un rectangle; un cercle et un triangle.

Leur déplacement se fait grâce à des roues motrices, folles et/ou directrices. Généralement ils sont constitués de châssis séparés en deux parties, l'une regroupant les organes de traction et de locomotion, l'autre

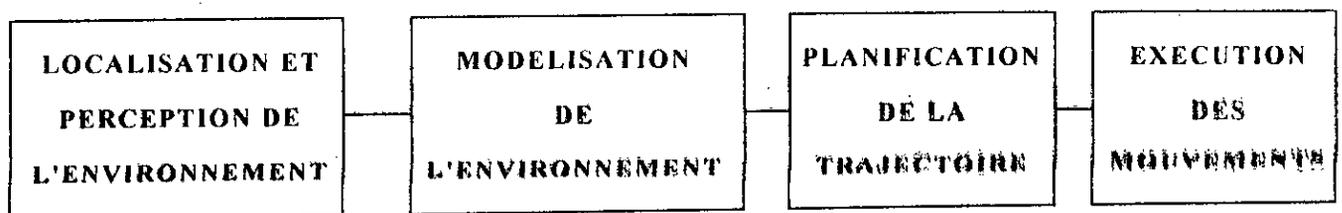
regroupant les organes de commande (système informatique et électronique). Certains disposent d'un bras manipulateur.

### 3. DEFINITION DU PROBLEME DE NAVIGATION:

La navigation autonome représente l'un des grands objectifs de la robotique mobile, elle consiste en partie en la planification de trajectoire sans collision pour un corps mobile dans un environnement donné. Ce problème est connu sous le nom évocateur du problème du "déménageur de piano" et a fait l'objet de diverses études [TOU 88].

L'objectif de cette planification est de générer une trajectoire qui relie une configuration initiale à une configuration finale du robot en lui évitant toutes collisions avec les obstacles de l'environnement.

La fonction mobilité d'un robot mobile autonome est composée de plusieurs modules.



Trois modes de navigation ont été mis au point en simulation (certains testés sur des robots) suivant que cet environnement soit connu, partiellement connu ou inconnu [PLA 83b]:

#### 3.1. NAVIGATION DANS UN ENVIRONNEMENT CONNU:

La connaissance de l'environnement évite sa perception, le système de navigation doit donc construire un modèle de celui-ci dans lequel la trajectoire sera recherchée. De telles méthodes de planification de trajectoires sont appelées "Méthodes globales".

### **3.2. NAVIGATION DANS UN ENVIRONNEMENT INCONNU:**

Le robot découvre son environnement progressivement au fur et à mesure qu'il se déplace grâce à ses différents systèmes de capteurs. Deux approches peuvent se présenter suivant la tâche confiée au robot:

#### **3.2.1. Navigation sans mémorisation de l'environnement:**

Ce mode est utilisé par un Robot qui exécute une seule mission dans l'environnement inconnu. Son chemin est défini par l'objectif qui lui est assigné, qui peut être soit d'atteindre un but donné à partir d'un point de départ (robot HILARE) [CAH 75]. Soit de balayer un atelier (le robot junior JASON nettoie le sol) [BAU 81].

La planification d'un tel chemin est guidé par ses moyens télémétriques qui le renseignent sur la présence des obstacles qu'il évite localement au fur et à mesure qu'il se déplace. Ces méthodes de planification sont appelées "méthodes locales".

#### **3.2.2. Navigation avec mémorisation de l'environnement:**

Contrairement au cas précédent, le robot à ce niveau aura à exécuter plusieurs tâches dans son environnement, ce qui nécessite une phase d'apprentissage pour son acquisition. Après cette phase, le robot aura en sa possession un environnement partiellement connu dû à une imprécision dans la représentation (une caméra ne peut pas toujours détecter les obstacles masqués par d'autres). Cette connaissance partielle de l'environnement permet de chercher la trajectoire avec des méthodes globales. En cas de détection d'un obstacle imprévu lors de sa navigation, le robot entreprend une stratégie locale pour son évitement.

### **4. PERCEPTION DE L'ENVIRONNEMENT:**

La gamme des moyens de perception pouvant être utilisé par un robot afin de découvrir son environnement est extrêmement large. Ce qui est dû à l'évolution très rapide des technologies ces dernières années.

Les moyens de perceptions peuvent être:

- Des télémètres passifs: sont utilisés pour l'acquisition de l'environnement souvent inconnu et parsemé d'obstacles (caméra, caméra couplée d'un télémètre laser ....).
- Des télémètres actifs: ce type de capteur fait appel à une source spécifique qui émet soit de la lumière (généralement en infrarouge) soit des hyperfréquences, soit des ultrason, et détermine la distance à un obstacle par des mesures effectuées sur l'écho renvoyé par l'obstacle [HEN 83].

### **CONCLUSION:**

Dans notre travail on s'intéresse à la navigation d'un Robot mobile circulaire, évoluant dans un environnement connu et parsemé d'obstacles, et la simulation graphique en deux dimensions de l'exécution de son mouvement.

*Chapitre II*

*MODELISATION ET  
PLANIFICATION DE  
LA TRAJECTOIRE*

## I. POSITION DU PROBLEME:

La planification de trajectoires sans collisions est sans conteste le problème qui a suscité le plus de travaux durant ces dernières années [LAU 88].

Ces travaux se regroupent autour de deux grandes classes de méthodes [TOU 88] [SIM 89].

- **Les méthodes locales(dynamique):**

Elles utilisent des connaissances partielles de l'environnement pour générer des incréments de déplacement du robot permettant de le rapprocher du but.

Elles ne mémorisent pas la trajectoire suivie par le robot et ne prennent en compte, à un instant donné, que des interactions du robot mobile avec les obstacles les plus proches. Leur principal avantage est la rapidité qui permet une utilisation en temps réel. Cependant, par manque de connaissance globale de l'environnement, elle ne garantissent pas toujours d'atteindre l'objectif même si une solution existe. Ceci est dû à un blocage du robot par la concavité de l'environnement (culs-de-sac) ou à la présence de boucles infinies (cas où le robot après contournement de l'obstacle retombe sur la trajectoire déjà effectuée).

Au cas où elles trouvent une solution, celle-ci n'est pas toujours optimale. Parmi ces méthodes, la plus élaborée est celle dite des "champs de potentiels" que KHATIB a introduit pour la commande dynamique des manipulateurs [KHA 86]. Elle considère que le robot mobile est plongé dans un champs de potentiels fictifs, somme d'un potentiel attractif qui l'attire vers l'objectif et de potentiels repulsifs qui l'éloigne des obstacles.

- **Les méthodes globales (statique):**

Contrairement aux méthodes locales, ces méthodes nécessitent une modélisation de l'environnement pour construire un modèle de l'espace libre, espace dans lequel le robot évite les obstacles et qu'il utilise pour rechercher sa trajectoire. Elles sont basées sur une caractérisation et une

structuration en régions de zones libres de l'espace cartésien. Cette structuration est sous forme d'un graphe décrivant l'adjacence entre ces régions. La recherche d'une trajectoire reliant les configurations (position et orientation du mobile) initiales et finales du robot se ramène à la recherche d'un chemin dans ce graphe.

Il est à noter que certaines méthodes globales ne nécessitent pas la construction d'un graphe, la recherche se fait directement dans le modèle de l'univers (méthode de LEE).

Ces méthodes sont très coûteuse en temps et en mémoire quand la dimension de l'environnement devient importante.

Par contre, elle garantissent l'obtention d'une trajectoire a coût sûr si celle ci existe.

Dans ce chapitre, sont exposés les principaux modèles de représentation de l'environnement ainsi que quelques technique de planification de trajectoire sans collisions.

## **2. MODELISATION DE L'ENVIRONNEMENT(ETAT DE L'ART):**

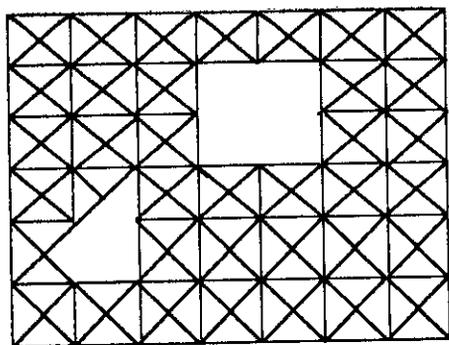
### **2.1. LE MODELE PAR GRILLE:**

Une grille est superposée à l'espace d'évolution du robot, définie par l'ensemble de noeuds et des cases qui résultent de l'intersection des lignes horizontales, verticales et parfois diagonales.

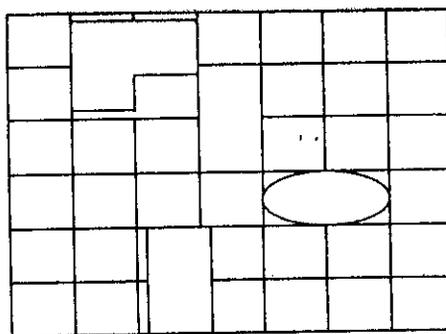
Les obstacles sont modélisés par des groupement de cases, leur sommets doivent appartenir aux noeuds de la grille.

Les unités de lieux sont des cellules polygonales (carré, losange,...) pouvant être vides ou occupées, définies par leurs centre de gravité (Fig 2.1, 2.2).

Le robot peut être ponctuel, occupe une cellule ou plus, et son déplacement peut se faire suivant quatre ou huit directions.



Modèle de grille à  
cellule losange  
(fig. 2.2)



Modèle de grille à  
cellule carrée  
(fig. 2.1)

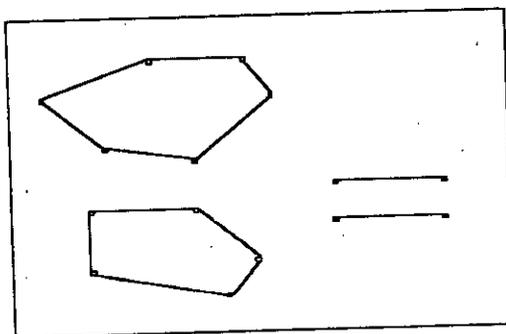
Une matrice est associée à ce modèle. On affecte à chaque élément la valeur "0" ou "1" suivant que la cellule correspondante est libre ou occupée. Ce modèle a été utilisé dans les premiers travaux sur le robot JPL (1969), robot Japonais N°1 (1971), robot de LENINGRAD (1973), ARGOS (1978) et LERM (1981).

Ce modèle est simple à mettre en oeuvre mais il présente des inconvénients tel que:

- Perte d'espace mémoire au prix d'une représentation précise de l'environnement.
- Les obstacles sont définis à partir des noeuds de la grille.

## 2.2. LE MODELE POLYGONAL:

Ce modèle remédie aux deux inconvénients du modèle de grille, d'abord il ne consomme pas autant d'espace mémoire pour la représentation de l'espace libre, et les obstacles sont définis par les positions réelles de leurs sommets dans l'espace cartésien (Fig 2.3).



Modèle polygonal Fig. 2.3

### 3. PLANIFICATIONS GLOBALE DE LA TRAJECTOIRE SANS COLLISIONS:

#### 3.1. GENERATION DE TRAJECTOIRE PAR UN MODELE PAR GRILLE:

##### Méthode de LEE:

Cet algorithme est le plus connu pour la recherche d'un chemin [RUB 74] [HOE 76]. Il utilise une grille où l'unité de lien est une cellule carrée et dans laquelle le robot peut se déplacer d'une cellule à une autre adjacente suivant quatre directions (fig. 2.4).

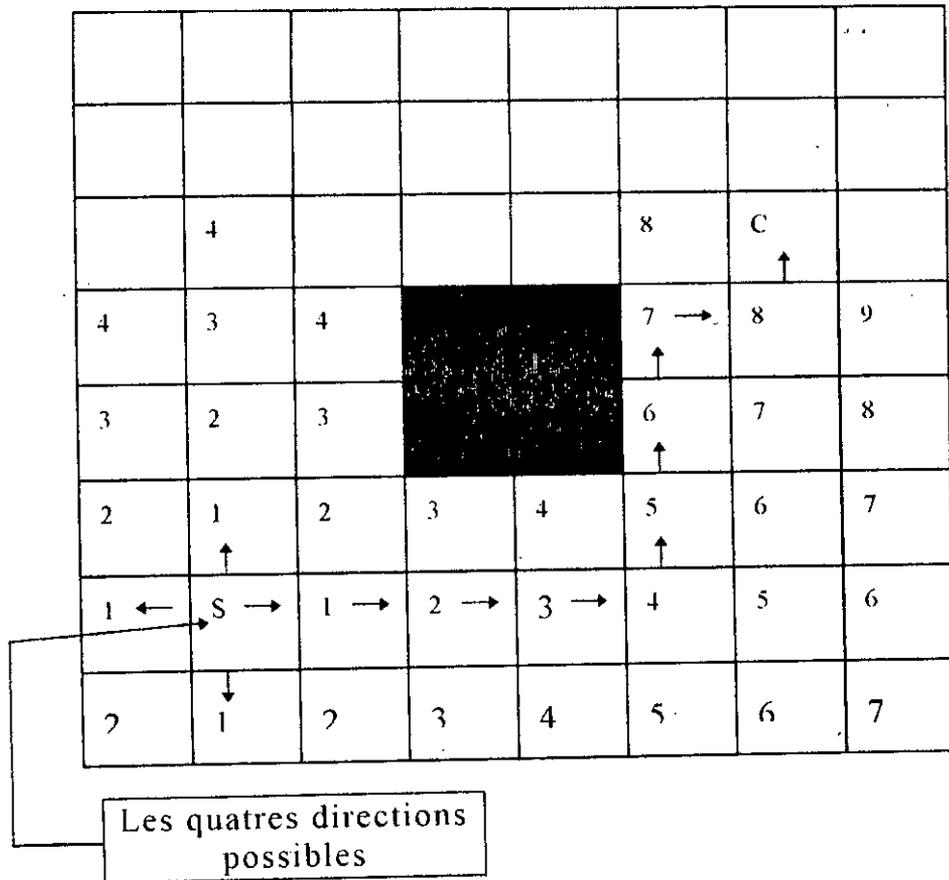
Pour déterminer un chemin entre les points source et cible, l'algorithme développe une onde à partir de la cellule centrée au point source de la manière suivante:

Pour chaque cellule entourant la cellule de départ et n'appartenant pas à un obstacle est assigné un marquage de "1". Ensuite un marquage de "2" est assigné à toute les cellules libres non marqués, voisines des' cellules de marquage "1".

Ce processus est répété jusqu'à aboutir à l'une des situations suivantes:

- on ne peut développer l'onde et le point cible n'est pas atteint, il n'y a pas de chemin reliant les points source et cible;
- Le point cible est atteint: le chemin est trouvé en remontant du point cible au point source en choisissant à chaque fois comme cellule suivante, une des cellules de marquage inférieur adjacente à la cellule courante. Cela implique l'existence de plusieurs chemin tous de même longueur, et cette longueur est optimale pour les types de déplacements considérés.

Cette méthode est très simple à mettre en oeuvre, néanmoins elle est coûteuse en temps et en espace mémoire et ce à cause du développement d'onde.

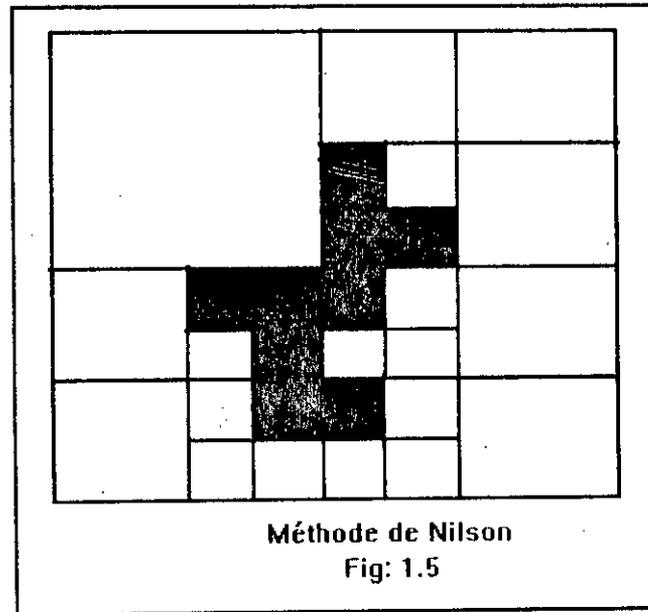


**Chemin g n r  par la m thode de LEE pour joindre S   C (fig. 2.4).**

### 3.2. GENERATION DE TRAJECTOIRE PAR UN MODELE POLYGONAL :

#### 3.2.1. M thode de NILSSON [NIL 80]:

Elle a  t  largement utilis es dans les premiers travaux (SHAKY, CART, JASON) [JUL 84]. Elle consiste   supersposer   l'environnement une grille 4\*4. On obtient ainsi des cellules carr es qui peuvent  tre vides, partiellement ou totalement occup es. Chaque cellule partiellement occup e est   son tour d compos e en une grille 4\*4. Ce processus est r p t  jusqu'  avoir une repr sentation suffisamment fine des obstacles. (Fig 2.5).



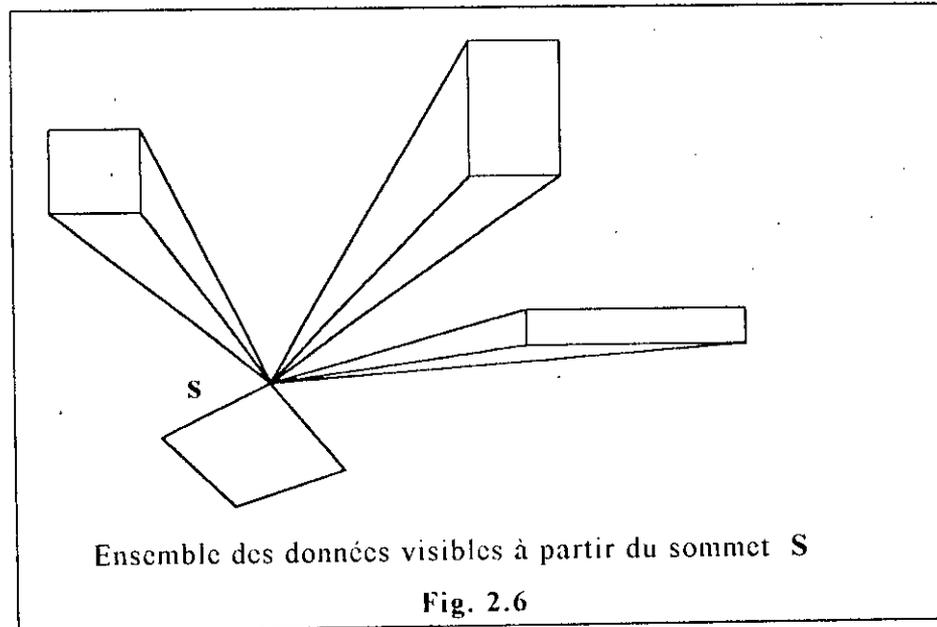
Un noeud du graphe est un sommet de l'obstacle, et une arrête reliant deux noeuds indique que les deux sommets sont atteignables l'un de l'autre en ligne droite. Le chemin optimal reliant les points source et cible est déterminé par application de l'algorithme A\* (voir annexe).

Au niveau de cette méthode aussi, on retrouve l'inconvénient de l'espace mémoire bien qu'il soit diminué par rapport aux méthodes utilisant le modèle par grille, du fait que l'espace libre n'est pas totalement discrétisé.

### 3.2.2. Méthode de LOZANO-PEREZ: [LOZ 83]

Les obstacles, dans un environnement en 2D, sont grossis selon l'encombrement du robot. Lorsqu'un robot a une forme circulaire, les obstacles sont grossis de son rayon, pour un robot rectangulaire avec une orientation de départ et une orientation d'arrivée les obstacles sont grossis d'une enveloppe, regroupant toutes les orientations intermédiaires entre les orientations de départ et d'arrivée.

La recherche du chemin optimal est effectuée dans un graphe appelé "graphe de visibilité" dont les noeuds représentent les sommets des obstacles grossis et les arrêtes une relation de visibilité. Deux sommets sont mutuellement visibles et peuvent donc être reliés par une ligne droite sans intersecter l'obstacles (voir figure 2.6).



La détermination des successeurs d'un noeud du graphe se fait par la recherche des noeuds atteignables à partir d'un noeud courant.

On appelle noeud atteignable à partir d'un noeud courant l'ensemble des noeuds représentant des sommets visibles à partir du sommet représenté par le noeud courant. Le grossissement des obstacles et le graphe de visibilité permettront de définir une trajectoire, sans avoir à prendre en considération les relations cinématique entre les mouvements de translation et de rotation, vu qu'une fois le grossissement fait on assimile le robot à un mobile ponctuel.

Cette méthode est l'une des plus importantes en robotique mobile. Son intérêt réside dans le fait qu'elle génère le plus court chemin, ce chemin est considéré comme optimal car il passe au plus près des obstacles. De plus, cette méthode prend en considération l'orientation du robot et son encombrement, aspect qui fut assez longtemps négligé.

L'inconvénient de cette méthode est que le grossissement des obstacles peut occlure des chemins qui existent, en plus du fait qu'elle est coûteuse en temps.

### 3.2.3. Méthode des cellules polygonales convexes

(RAJA CHATILA): [RAJ 81]

Cette méthode a été définie dans le cadre du projet "HILARE". La représentation de l'espace de recherche, ou encore la structuration de l'espace de recherche, ne se fait pas sur les obstacles comme précédemment, mais sur l'espace libre d'évolution du robot.

Le but est de décomposer l'univers en unités élémentaires (que l'on associe à des noeuds reliés par les relations d'accessibilité, ce qui permet de représenter l'espace libre par un graphe de recherche.

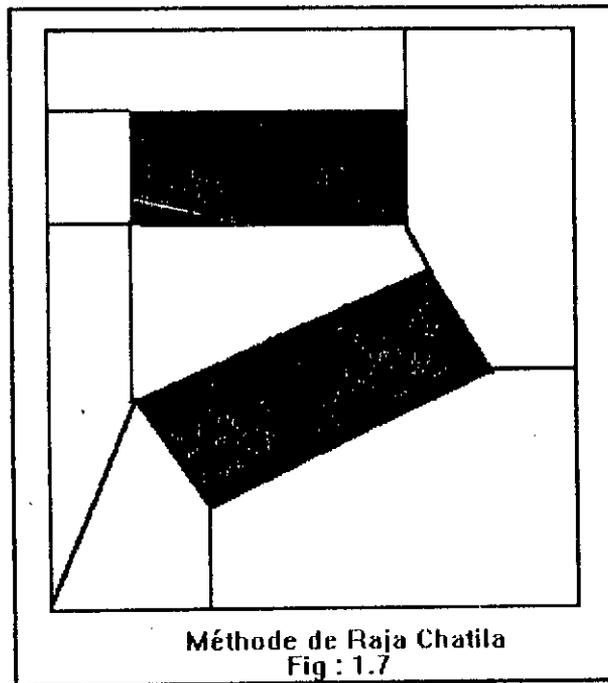
L'idée est de discrétiser l'espace en cellules polygonales convexes, dont la forme et la taille dépendent des objets de l'environnement (fig. 2.7).

Une fois ces cellules obtenues, on construit le graphe de recherche comme suit; un arc entre deux cellules n'est établi que si la longueur de leur côté commun est supérieur à l'encombrement du robot.

La recherche de chemin s'effectue dans ce graphe. Elle consiste à déterminer dans quels cellules se situent le point de départ et le point d'arrivée et à trouver un chemin optimal reliant ces deux cellules grâce à l'algorithme A\*.

L'originalité de cette méthode réside dans le fait qu'elle génère un chemin sûr éloigné des obstacles du fait qu'il passe par les centres de gravité des cellules, ce qui évite de les grossir.

De plus, elle est peu coûteuse en espace mémoire par rapport aux méthodes citées précédemment. Cependant, la construction de l'espace de recherche du chemin qui exige de nombreux calculs d'intersection de droites, est d'autant plus longue à réaliser que le nombre d'obstacles augmente.



### 3.2.4. Méthode des cônes généralisés (Brooks): [BRO 82]

Brooks partitionne l'espace libre d'évolution du robot en un ensemble de couloirs appelés "cônes généralisés" (Free ways).

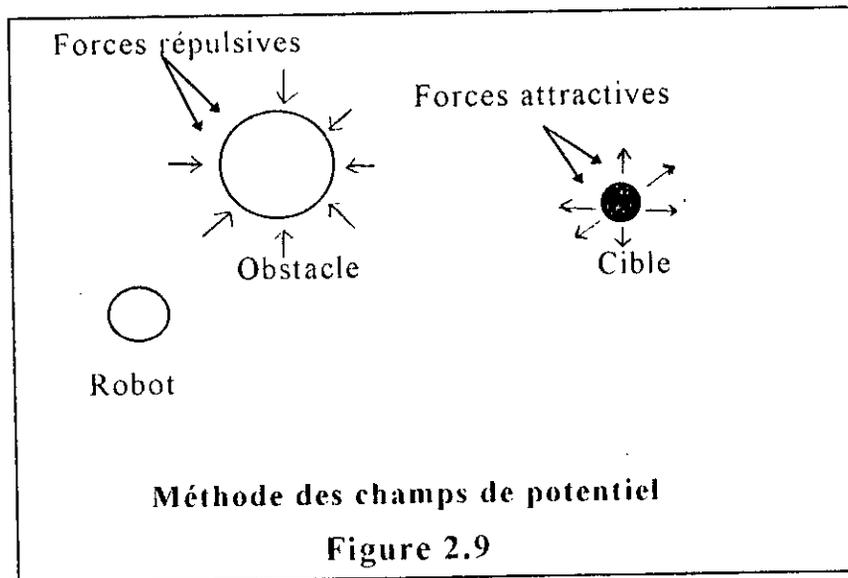
A chaque couloir, on associe un intervalle d'orientations possibles calculé en fonction de la position de référence du robot se déplaçant sur l'axe du cône.

Il structure alors l'espace libre sous la forme d'un graphe dont les noeuds représentent les points d'intersection entre les axes, tandis que les arcs représentent l'existence d'un intervalle angulaire valide commun entre les noeuds (voir figure ).

La recherche de chemin se fait dans le graphe grâce à l'algorithme A\*. Cette méthode fournit le chemin le plus court puisqu'elle ne frôle pas les obstacles.



Parmi les méthodes locales, l'approche la plus connue est certainement la méthode des champs de potentiel développée par Oussama Khatib [KHA 80].



Le principe de cette méthode consiste à introduire des forces s'exerçant à distance sur le robot. Ainsi, la cible exercerait des forces attractives sur le robot alors que les obstacles constitueraient un ensemble de forces répulsives vis à vis du robot (fig. 2.9).

Plusieurs expériences ont montré que les méthodes de potentiel comportent divers avantages, dont la rapidité qui permet une utilisation quasiment en temps réel, même lorsqu'il s'agit de système complexe.

De plus, le robot traite les informations de son environnement par le biais de système de perception, ce qui implique que le planificateur local peut, en temps réel, prendre en compte les obstacles imprévus.

Par contre, la difficulté majeure de cette méthode est l'ajustement des gains entre le terme attractif et le terme répulsif, pour éviter certains blocages dus à des minima locaux de potentiel.

Ainsi, le robot peut être pris dans une sorte de piège, notamment lorsque la cible se trouve entre deux obstacles, ou encore lorsque le contournement d'un obstacle le ramène sur sa trajectoire. De nombreuses méthodes ont été développées afin de faire sortir le robot de ces situations

de blocage. Lumelsky a montré que pour des systèmes simples, il est possible de garantir la convergence de la méthode en définissant des heuristiques appropriés à chaque cas.

D'autres méthodes ont été développées dans le contexte de la navigation locale, mais généralement le même reproche est fait à ces méthodes. Elles ne garantissent le succès que si la configuration des obstacles est suffisamment aérée pour permettre un cheminement confortable au robot. Elle échouent particulièrement lorsqu'il est question d'environnement en forme de labyrinthe. De plus, les chemins suivie ne sont pas toujours optimaux.

Néanmoins, le grand mérite de ce type de navigation reste sa faculté de s'adapter a des environnements quelconques, sans qu'aucune modélisation de l'environnement ne soit nécessaire. L'autre atout incontestable de ces méthodes est la simplicité de leurs algorithmes de navigation qui rend leurs exécutions en temps réel tout à fait réalisable.

## 5. DISCUSSION:

Les méthodes locales sont souvent peu efficaces surtout si l'espace d'évolution du robot est un espace très exigü ou en forme de Labyrinthe. Il y'a un risque de se retrouver dans des situations de blocage, par ailleurs les méthodes locales generent des trajectoires qui ne sont pas forcément optimales. En effet, on ne dispose d'aucune connaissance à priori sur l'environnement et pose des risques de prolonger inutilement le chemin, voir tourner en rond avant d'atteindre sa cible.

Ces deux aspects du problème nous ont conduit à opter pour la conception et la réalisation d'un planificateur global. Comme nous l'avons vu précédemment, un planificateur global peut être implémenté selon différentes méthodes.

Le modèle polygonale est plus intéressant que le modèle par grille. Actuellement ce modèle est le plus utilisé dans les laboratoires. Les méthodes de planification des trajectoires utilisant le modèle polygonale sont moins gourmandes en mémoire que celles utilisant le modèle par grille, mais ce dernier s'adapte bien aux robots n'ayant pas des contraintes simulatives ou qui peuvent être assimilées à un point matériel et aux robots qui occupent une seule case, en outre le modèle par grille est plus facile à mettre en oeuvre et parfois moins couteux en mémoire.

*Chapitre III*  
*NOTION*  
*DE BASE SUR*  
*LA LOGIQUE FLOUE*

## 1. INTRODUCTION:

La logique floue est un outil qui offre des avantages multiples pour la résolution des problèmes technique. Ces avantages consistent à un temps de développement plus courts, une meilleure performance, un coût moins élevé du hardware et une robustesse plus importante [VIOT 93].

Les compagnies japonaises ont été les premières à reconnaître et à exploiter la valeur commerciale de la logique floue, en l'incorporant dans une large gamme de produit tel que les caméras, les voitures, électroménagers, la médecine etc. [VIOT 93].

En général, la logique floue est utilisée dans des problèmes complexes de contrôle (Handling) qui ne peuvent pas être définis facilement par un modèle mathématique formel. Cependant, elle n'est pas limitée à ces problèmes, en effet la logique floue s'applique aussi à des domaines moins techniques (Softer sciences) tel que les statistiques, la psychologie et la sociologie.

La question que l'on pourrait supposer est: à quoi est dû ce grand succès? ceci est dû principalement au fait que les contrôleurs basés sur la logique floue tolèrent l'imprécision.

Considérons par exemple, le stationnement d'une voiture. Généralement un conducteur d'automobile stationne sa voiture dans un endroit quelconque d'un parking sans grande difficulté puisque la position finale de l'automobile ne lui est pas spécifiée exactement. En fait, il n'est pas tenu de la respecter à 100%.

C'est pour raisonner de cette manière là, que les contrôleurs basés sur la logique floue ont été développées.

Par contre si ce stationnement avait été spécifié avec une grande précision (grâce à des contrôleurs basés sur la logique classique), ça aurait pris des jours et peut être plus pour stationner une voiture.

LOTFI ZADEH, professeur à l'université Berkeley, de Californie; est la personne la plus largement associée à la logique floue. En 1965, il écrit un article originale "Fuzzy Sets" où il définit de manière formelle la théorie des ensembles flous.

Les premières recherches sur le contrôleur flou ont été faites par MANUDANI et ses collègues, motivé par les publications de ZADEH sur l'analyse des systèmes et l'approche linguistique basées sur la logique floue.

## 2. ENSEMBLES FLOUS ET LOGIQUE FLOUE [CHU 90]:

### 2.1. ENSEMBLES FLOUS ET TERMINOLOGIE:

Traditionnellement, une expression ou une condition logique, ne peut appartenir qu'à deux extrémités complètement vraie ou bien complètement fausse.

Par contre, dans le concept flou une entrée a des degrés qui varie entre 0 et 1.

Les ensembles sont définis de manière qualitative en utilisant des termes linguistiques tels que Froid, Chaud, Brulant...

Soit U une collection d'objets notée (u) qui peuvent être discrets ou continus. U est appelé l'univers de discours.

**Definition 1:** un ensemble flou F dans un univers de discours U caractérisé par une fonction d'appartenance qui prend des valeurs dans un intervalle [ 0 , 1 ]. doit être représenté comme un ensemble de paires ordonnées d'un élément  $\mu$  et de sa valeur dans la fonction d'appartenance

$$\mu_F: U \implies [ 0 , 1 ]$$

$$F = \{ (U, \mu_F(U)) / U \in U \}.$$

## 2.2. OPERATION THEORIQUE D'ENSEMBLES:

Soit A et B, deux ensembles flous dans U avec des degrés d'appartenance  $\mu_A$  et  $\mu_B$  respectivement.

### Définitions:

Union:  $\mu_{A \cup B}(U) = \max \{ \mu_A(U), \mu_B(U) \}$ .

Intersection:  $\mu_{A \cap B}(U) = \min \{ \mu_A(U), \mu_B(U) \}$ .

Complement:  $\mu_{\bar{A}}(U) = 1 - \mu_A(U)$ .

Produit cartésien:  $\mu_{A_1 \times \dots \times A_n}(U_1, \dots, U_n) = \min \{ \mu_{A_1}(U_1), \dots, \mu_{A_n}(U_n) \}$

Composition Sup-Star: si R et S sont des relations floues dans U.V et V.W respectivement, la composition de R et S est une relation floue notée RoS et définie par:

$RoS = \{ [(u,w), \sup(\mu_R(u,v) * \mu_S(v,w))], u \in U, v \in V, w \in W \}$ .

Où \* peut être un opérateur quelconque dans la classe des normes nommé, minimum, produit algébrique.

## 2.3. VARIABLE LINGUISTIQUE:

Une variable linguistique est caractérisée par un quintuplet (X, T(X), U, G, M).

X : Nom de la variable.

T(X) : l'ensemble des noms des valeurs linguistiques de X où chaque valeur étant un nombre flou défini sur U.

G : une règle syntaxique pour générer les noms des valeurs de x.

M : règle sémantique pour associer à chaque valeur son sens.

### Exemple:

Si "vitesse" est interprétée comme une variable linguistique, alors son ensemble de termes T (vitesse) = {lente, moyenne, rapide; très lent, ...}, où chaque terme de T (vitesse) est caractérisé par un ensemble flou dans l'univers de discours  $U = [0, 100]$ . On peut interpréter "lente" comme "une vitesse inférieure environs à 40m/n", "moyenne" comme "une vitesse autour de 50m/n" et "rapide" comme "une vitesse supérieure environs à 70m/n. Ces termes peuvent être caractérisés comme ensembles flou dont les fonctions d'appartenance sont données dans la figure 3.1.

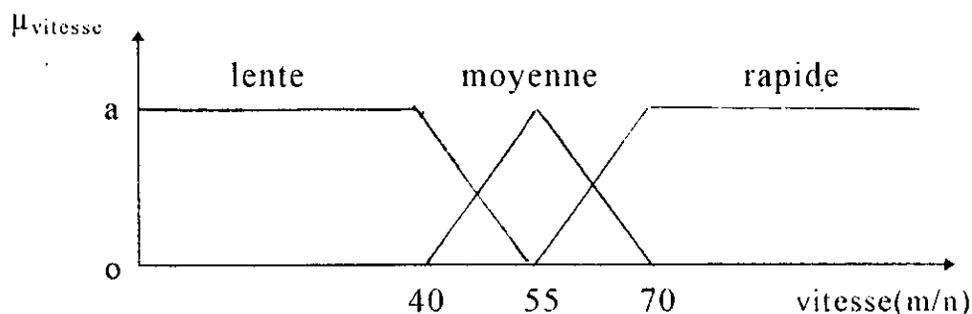


Fig. 3.1

**2.4. LOGIQUE FLOUE ET RAISONNEMENT APPROXIMATIF:**

En logique floue et raisonnement approximatif il existe deux règles de déduction d'implication floue nommées **GMP** (Generalized Modus Ponens) et **GMT** (Generalised Modus Tollens).

Prémisse 1: x est A'		
Prémisse 2: si x est A alors y est B.		<b>(GMP)</b>
Conséquence: y est B'		
Prémisse 1: y est B'		
Prémisse 2: si x est A alors y est B.		<b>(GMT)</b>
Conséquence: x est A'		

Ici, on introduit des ensembles flous A, A', B, B' via les variables linguistiques x, y, à la place des ensembles dans la logique binaire.

La **(GMP)** réduite en **(MP)** lorsque A' = A et B' = B, est proche relativement à la déduction en chainage avant, qui est particulièrement utilisé dans les FLC.

La **(GMT)** réduite en **(MT)** lorsque B' = non B et A' = non A, est proche relativement à la déduction en chainage arrière utilisé dans les systèmes experts spécialement dans les diagnostics médicaux.

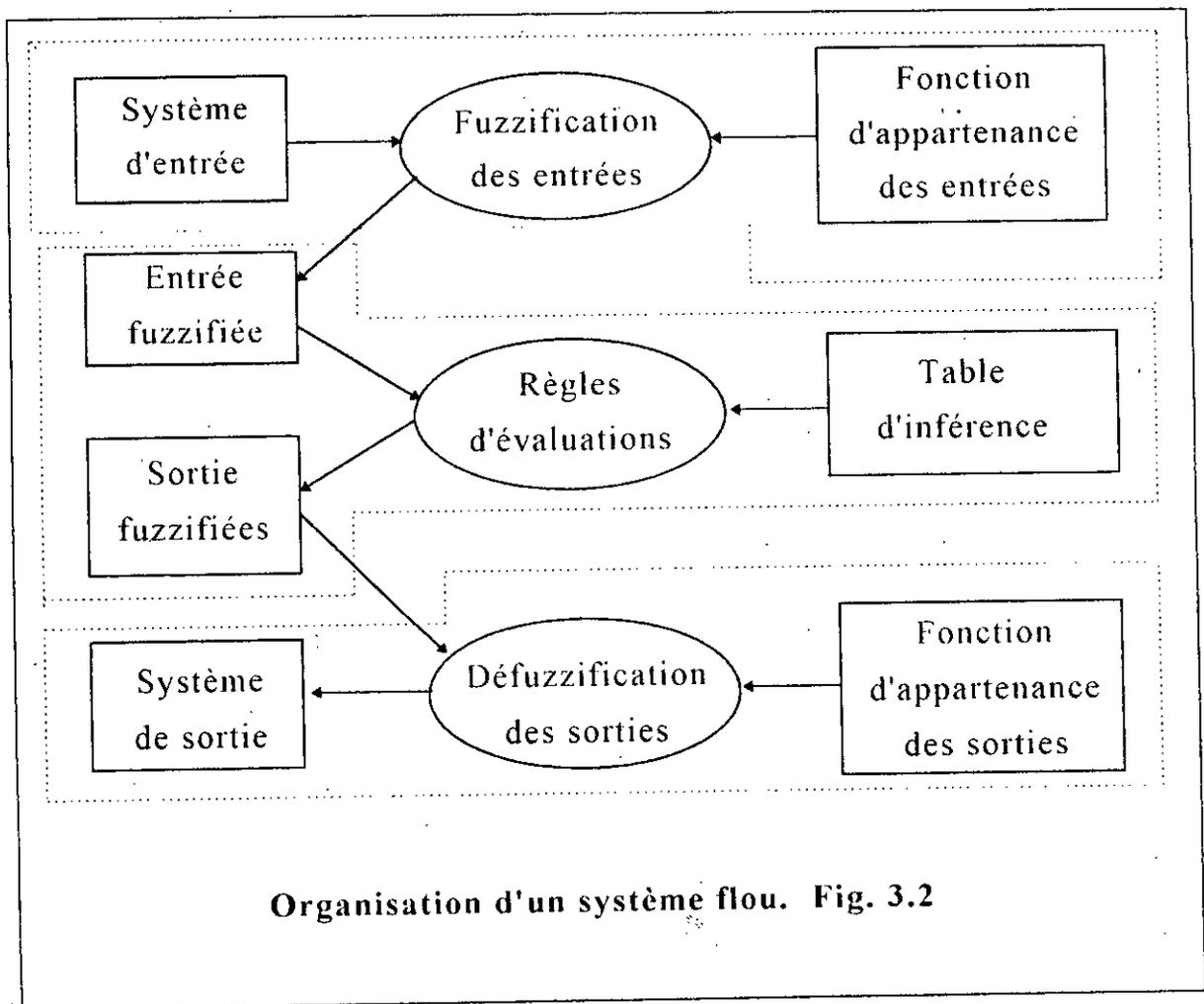
### 3. ORGANISATION D'UN SYSTEME FLOU :

Un système flou est capable d'évaluer et de contrôler un système externe. Ceci se fait grâce aux informations qu'il perçoit du système externe et celles qu'il restitue à ce dernier.

L'ensemble d'informations perçues est englobé dans un système d'entrée et l'ensemble des informations restituées est englobé dans un système de sortie.

Un traitement flou se base sur trois opérations fondamentales:

- Une opération dite de "FUZZIFICATION" qui permet de faire correspondre à une donnée réelle une donnée floue.
- L'évaluation qui se base sur un ensemble de règles grâce auquel le contrôleur décidera de la sortie à appliquer au système externe.
- Enfin une "DEFUZZIFICATION", opération symétrique à la fuzzification. C'est elle qui permet le passage d'une donnée floue à une donnée réelle [VIOT 93], Fig (3.2).



Dans ce qui suit, nous abordons plus en détail les concepts de base de l'élaboration d'un contrôleur flou.

### 3.1. Définition du système d'entrées et du système de sortie du contrôleur:

Un contrôleur est conçu pour répondre à un besoin bien précis et restitue une commande en fonction de ce qui lui a été injecté en entrée.

Ceci fait ressortir la nécessité de bien définir les variables d'entrée ainsi que les variables de sortie qui dépendent du problème à traiter.

### 3.2. Fuzzification des entrées:

Chaque système d'entrée peut contenir une ou plusieurs classifications qualitatives, appelées "label" ou ensemble flou. A chacun est associé une fonction d'appartenance.

Par exemple, si la température est une entrée d'un système flou, alors les ensembles flous seront "Froid", "Frais", "Bon", "Chaud", et "Brulant".

La valeur d'une température donnée à un degré d'appartenance à chaque ensemble flou, ce degré pouvait être nul.

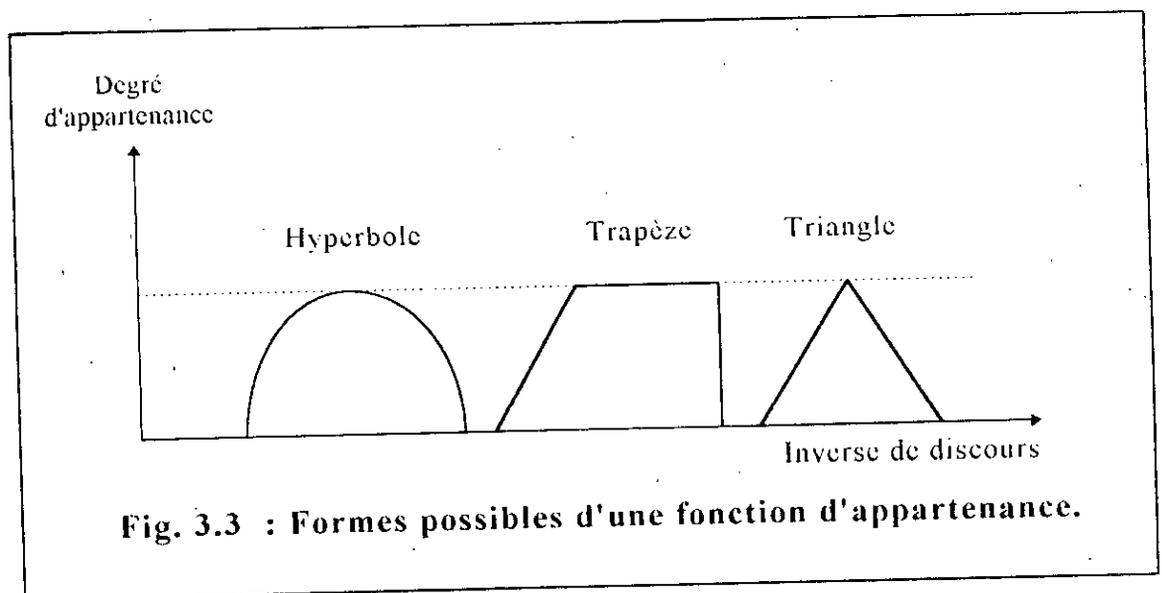
Les degrés d'appartenance à un ensemble flou sont décrits par des fonctions appelées fonction d'appartenance (membership function).

Une fuzzification consiste donc à calculer le degré d'appartenance d'une entrée à tous les ensembles flous du système.

Le plus important dans cette opération est de trouver les fonctions d'appartenance. Le concepteur du système les définit en se basant sur son intuition et/ou sur expérience.

Pour définir ces fonctions, on retient souvent des formes simples tel que les trapèzes et les triangles. (Voir fig. 3.3).

Cependant, il est à noter que la simplicité d'une fonction d'appartenance implique une facilité d'implémentation, de maintenance, de manipulation et d'exécution. Les triangles et les trapèzes sont les plus souvent utilisés et ont prouvé leur efficacité [VIOT 93].



Le recouvrement (Overlapping) entre les ensembles flous est nécessaire, si non le système aura tendance à fonctionner comme un système gouverné par la logique bivalente classique; diminuant ainsi sa souplesse et sa possibilité de traiter des situations intermédiaires (voir fig. 3.5). Des pourcentage de recouvrement de 25% à 50% sont utilisés dans la majorité des cas.

Toutefois, le recouvrement ne doit pas dépasser ces limites car trop de recouvrement diminue la possibilité du système à faire la distinction entre deux ensembles flous. [VER 92]

En conclusion, le concepteur peut modéliser le comportement du système en langage naturel, ce qui accroît la possibilité de décrire des tâches complexes de manière claire.

#### • Algorithme de Fuzzification:

Ayant établi les fonctions d'appartenances et connaissant le principe de fuzzification, toute entrée du contrôleur flou est fuzzifiée grâce à l'algorithme suivant :

$\Delta_1 := \text{entree} - p_1;$

$\Delta_2 := p_2 - \text{entree};$

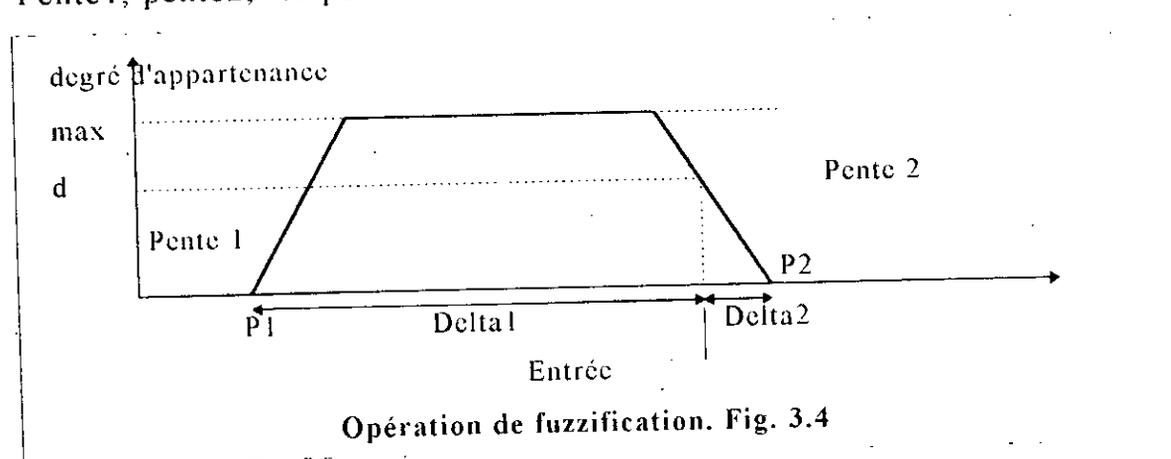
si  $(\Delta_1 \leq 0)$  ou  $(\Delta_2 \leq 0)$

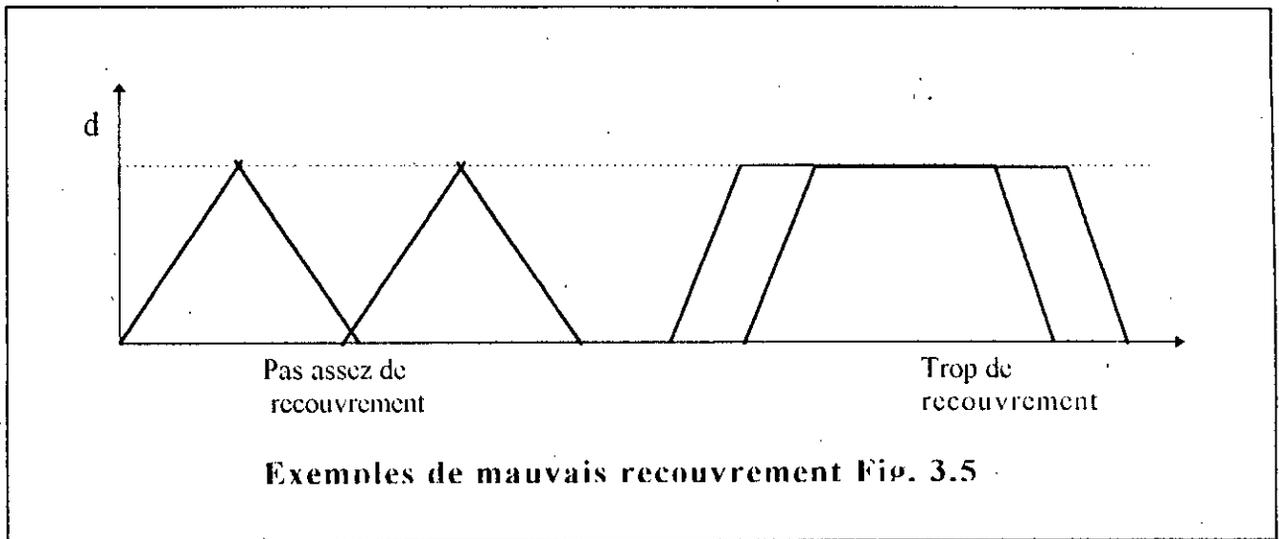
Alors  $d=0;$

sinon  $d := \min(\text{pente}_1 * \Delta_1, \text{pente}_2 * \Delta_2, \text{Max});$

On note  $p_1, p_2$  les extrémités de la fonction d'appartenance.

$\text{Pente}_1, \text{pente}_2$ , les pentes de cette dernière (voir figure 3.4)





### 3.3. REGLES D'EVALUATIONS ET LA LOGIQUE DE PRISE DE DECISION:

Pour décrire le comportement d'un système flou, le concepteur doit développer un ensemble de règles ayant la forme :

#### SI-ALORS (IF-THEN STATEMENT)

La partie "SI" de la règle contient une ou plusieurs conditions, appelées "antécédant", la portée "ALORS" contient un ou plusieurs actions, appelées "conséquence".

L'antécédent d'une règle correspond directement au degré d'appartenance (Fuzzi input) calculé durant la fuzzification.

Quant à la conséquence; elle correspond à l'action à entreprendre si l'antécédent de la règle est vérifié (Fuzzy output).

Le degré d'appartenance de la sortie (conséquence) est obtenu en appliquant la fonction minimum aux entrées fuzzifiées se trouvant dans la sortie antécédent.

$$\mu_R (A,B) = \min(\mu_A , \mu_B)$$

Il est important de signaler que si plusieurs règles accèdent au même ensemble flou de sortie, alors on retiendra le maximum de degrés d'appartenance.

### 3.4. DEFUZZIFICATION DES SORTIES:

Une fois que le processus d'évaluation des règles affecte à chaque ensemble flou de sortie son degré d'appartenance, un traitement supplémentaire est requis. Ce processus est appelé défuzzification. Ce traitement est nécessaire pour deux raisons :

- Supprimer la notion de flou au niveau des actions à accomplir; et ceci en utilisant les fonctions d'appartenance des sorties.
- Résoudre le conflit entre plusieurs actions compétitives.

A chaque entrée, on associe plusieurs notes (les degrés d'appartenance aux sous ensembles flous d'entrées). La Defuzzification consiste à calculer une valeur unique à partir des différentes notes obtenues, grâce à l'application des règles et ce pour chaque sortie.

#### \* Méthode du maximum (MN):

Elle donne le point auquel la possibilité de distribution de la commande atteint son maximum.

#### \* Méthode de la moyenne du maximum (MON):

Donne la valeur moyenne de toutes les valeurs de la commande de chaque règle pour lesquelles leurs fonctions d'appartenances atteignent le maximum.

$$G = \frac{1}{L} \sum_{j=1}^L (S_j / L)$$

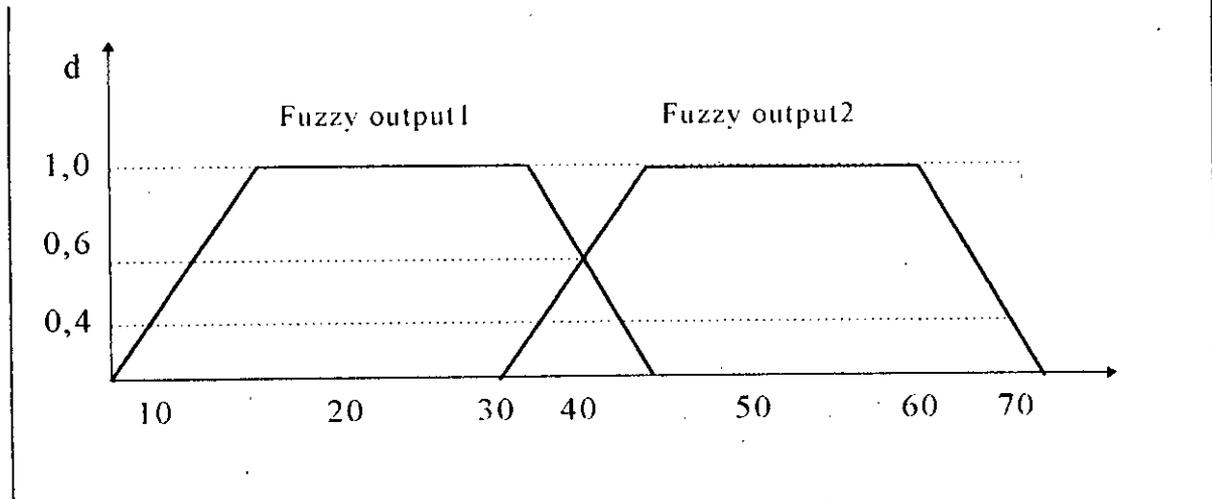
où  $S_j$ : la valeur pour laquelle la fonction d'appartenance atteint son maximum.

$L$ : le nombre de fonction d'appartenance.

**\*Méthode de centre de gravité:**

La méthode de defuzzification la plus souvent utilisée est la "Fuzzy centroïd defuzzification" qui revient à calculer le centre de gravité de plusieurs figures simples.

Nous illustrons l'implémentation de la defuzzification par l'exemple suivant (fig. 3.6).



Opération de défuzzification Fig. 3.6

1. Le centre selon l'axe des "x" (x-axis centried) de Fuzzy out put 1 = 20.  
Le centre selon l'axe des "x" (x-axis centried) de Fuzzy out put 2 = 50.
2. Le degré d'appartenance de la sortie Fuzzy output = 0.6  
Le degré d'appartenance de la sortie a fuzzy output 2 = 0.4
3. La surface hachurée correspond à Fuzzy output 1 = 20.4  
La surface hachurée correspond à fuzzy output 2 = 14.4  
Le centre de gravité est donné par la relation suivante :

$$G = \frac{\sum_{i=1}^n S_i * X_i}{\sum_{i=1}^n S_i}$$

On notera:  $S_i$ : surface du  $i^{\text{ème}}$  ensemble flou de sortie.

$X_i$ : Centre selon l'axe des X du  $i^{\text{ème}}$  ensemble flou de sortie.

$$\text{Donc: } G = (20.4+20+14.4*50) / (20.4+14.4)$$

$$G = 32.4$$

#### 4. CONCLUSION :

En conclusion, la logique floue et par conséquent, les contrôleurs basés sur la logique floue sont très appréciés, puis qu'ils permettent d'approcher le raisonnement humain (sans pour autant prétendre l'atteindre).

Ainsi, ces contrôleurs ont été utilisés pour plusieurs applications telles que:

- Le contrôle des qualités des eaux,
- Les systèmes de trains automatiques,
- le contrôle des ascenseurs de charges,
- le contrôle de la transmission automobile,
- le contrôle de robots mobiles.

*Chapitre IV*

*METHODE D'EVITEMENT  
D'OBSTACLES PAR  
LA LOGIQUE FLOUE*

## **METHODE D'EVITEMENT D'OBSTACLES BASEE SUR UN RAISONNEMENT DE LOGIQUE FLOUE**

### **1. PRINCIPE:**

L'être humain peut instantanément estimer sa décision approximative par sa capacité potentielle de percevoir le danger.

De ce point de vue, la méthode est déduite come suit :

- En premier, le robot estime le degré du danger basé sur sa reconnaissance intuitive approximée par les règles, de logique floues.
- En second, il décide de la prochaine action en utilisant la table de décision écrite avec les règles de production.

## 2. ALGORITHME DE LA METHODE:

### \* Le problème d'évitement d'obstacles:

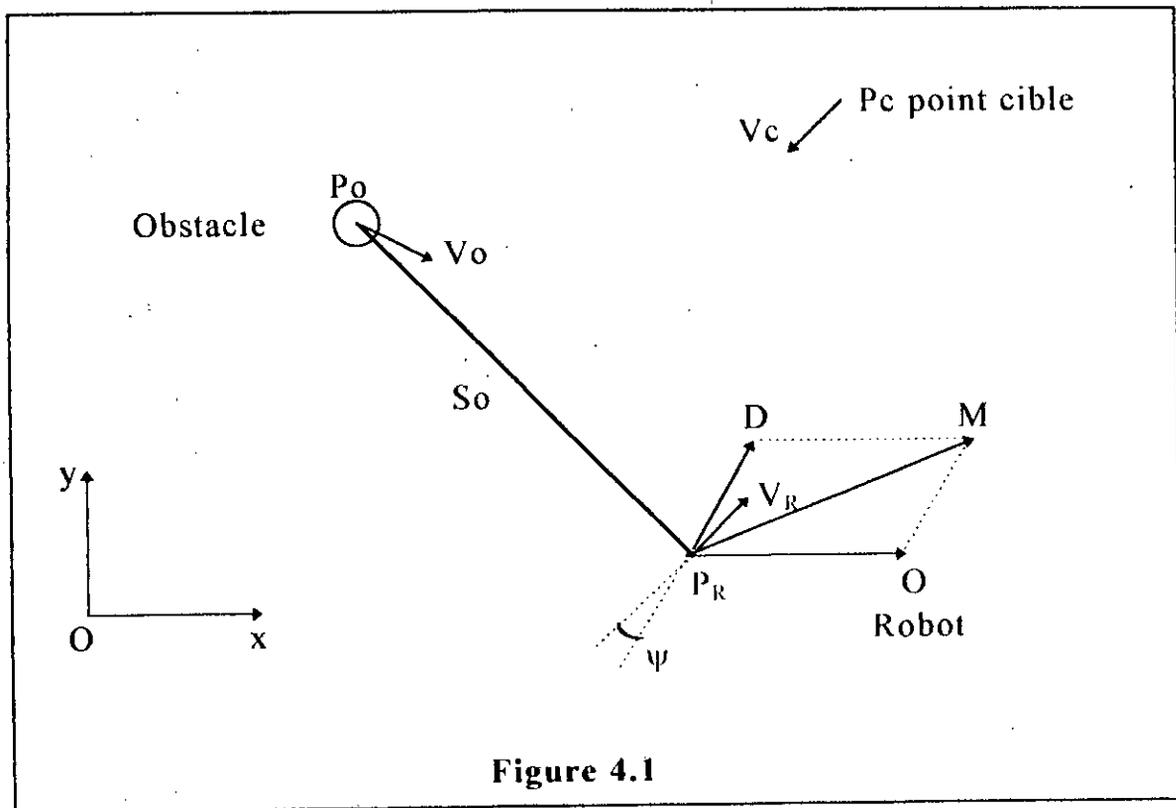


Figure 4.1

La figure 4.1 décrit l'état d'un robot mobile se dirigeant vers un point cible en évitant les obstacles statiques et dynamiques où:

- $P_r$ : vecteur de position absolue du robot,
- $P_o$ : vecteur de position absolue de l'obstacle,
- $P_c$ : vecteur de position absolue de la cible,
- $V_r$ : vecteur de vitesse absolue du robot,
- $V_o$ : vecteur de vitesse absolue de l'obstacle,
- $V_c$ : vecteur de vitesse absolue de la cible,
- $S_o$ : vecteur de position relative de l'obstacle,
- $U_o$ : vecteur de vitesse relative de l'obstacle,
- $O$ : vecteur d'évitement d'obstacles,
- $D$ : vecteur cible,
- $M$ : vecteur d'orientation,
- $U$ : la direction du robot,

basée sur les coordonnées cartésiennes absolues.

**NB:**  $S_0$  et  $U_0$  sont basés sur les coordonnées cartésiennes du robot.

**\* hypotheses:**

La méthode suppose que le robot mobile et les obstacles dynamiques vérifient les conditions suivantes:

- 1) Le robot mobile ne doit pas rencontrer des obstacles multiples en même temps.
- 2) Le robot mobile est contrôlé uniquement par son orientation.
- 3) Le robot mobile peut percevoir une distance, et une direction pour un obstacle à l'intérieur de la limite de 180 degrés en avant.
- 4) Le robot mobile se déplace exactement selon la commande.
- 5) Le robot mobile et l'obstacle sont concentrés en leurs centres de masses.
- 6) Le point cible est fixe.
- 7) Les obstacles ont un mouvement continu.

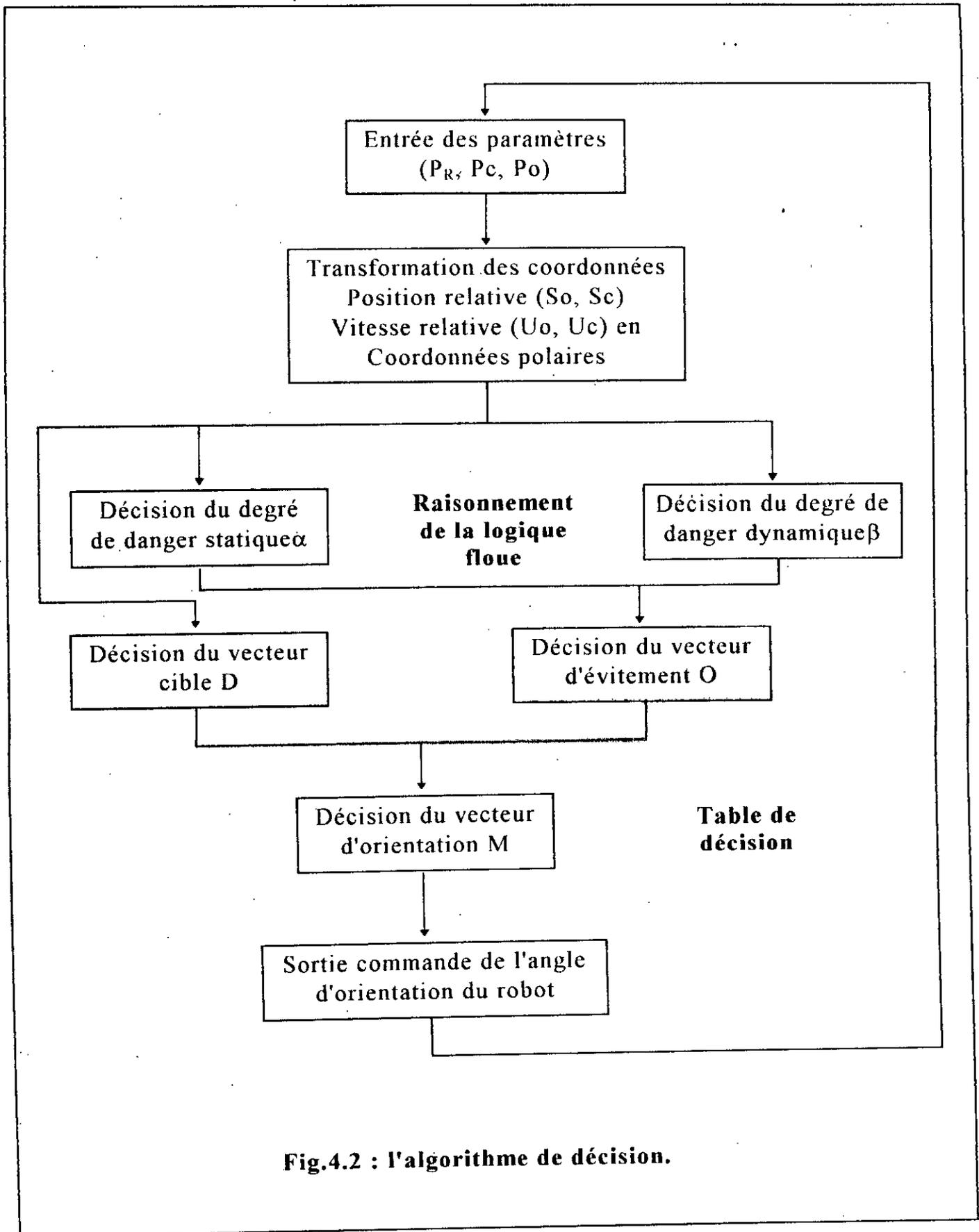


Fig.4.2 : l'algorithme de décision.

Grâce à un raisonnement de logique floue qu'on verra plus tard, le robot estime le degré du danger statique  $\alpha$  (basé sur les positions relatives) et le degré du danger dynamique  $\beta$  (basé sur les vitesses relatives).

Ayant déterminé les deux degrés du danger, et à partir d'une table de décision établie par une expertise, le vecteur d'évitement  $O$  est estimé, puis le vecteur cible  $D$  est calculé à partir des paramètres  $P_r$  et  $P_c$ .

A la dernière étape de l'algorithme, le robot détermine l'angle d'orientation  $M$  qui est égale à la somme vectorielle des vecteurs  $O$  et  $D$ . Enfin, la sortie de l'algorithme est l'angle d'orientation, grâce auquel le robot dévie pour qu'il évite l'obstacle (voir fig.4.2).

### 2.1. DECISION DU DEGRE DU DANGER STATIQUE:

C'est la relation liant la position statique du robot et de l'obstacle qui exprime ce degré du danger.

Le vecteur de position relative à l'obstacle est donné comme suit :

$$\delta = \pi/2 - \arctg(V_{ry}/V_{rx})$$

$$S_o = R(P_o - P_r) = [S_{ox} - S_{oy}]^T \quad (2.1)$$

$$\begin{pmatrix} \cos(\delta) & -\sin(\delta) \\ \sin(\delta) & \cos(\delta) \end{pmatrix} \quad (2.2)$$

où  $\delta$ : direction du robot mobile basé sur les coordonnées cartésiennes absolues.

$V_r = [V_{rx}, V_{ry}]^T$  : est le vecteur vitesse absolue du robot.

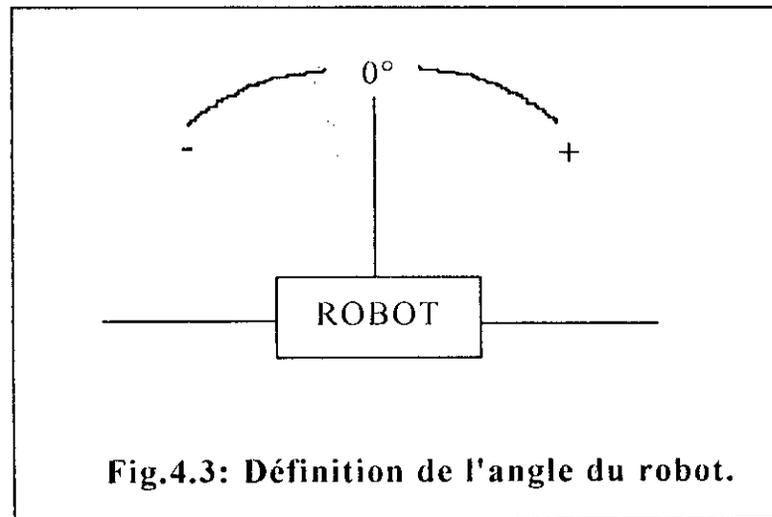
La distance et la direction d'un obstacle basées sur les coordonnées polaires du robot, dont l'angle est défini comme sur la figure 4.3, sont représentés par:

$$L_s = \sqrt{S_{ox}^2 + S_{oy}^2} \quad (2,3)$$

$$\theta_s = \frac{\pi}{2} - \arctg(S_{oy}/S_{ox}) \quad (2,4)$$

où  $L_s$ : distance d'un obstacle exprimée en fonction des coordonnées polaires du robot.

$\theta_s$ : direction d'un obstacle exprimé en fonction des coordonnées polaires du robot.



\* On suppose que le robot peut percevoir un obstacle dans une direction qui appartienne au champ  $-90^\circ$  à  $90^\circ$ , ce dernier est divisé en quatre régions.

La fonction d'appartenance (Membership function) [Mae 90] de direction  $\theta_s$  peut être représentée par quatre labels de logique floue comme le montre la figure 4.4:

LL: large à gauche.

LS: petite à gauche.

RS: petite à droite.

RL: large à droite.

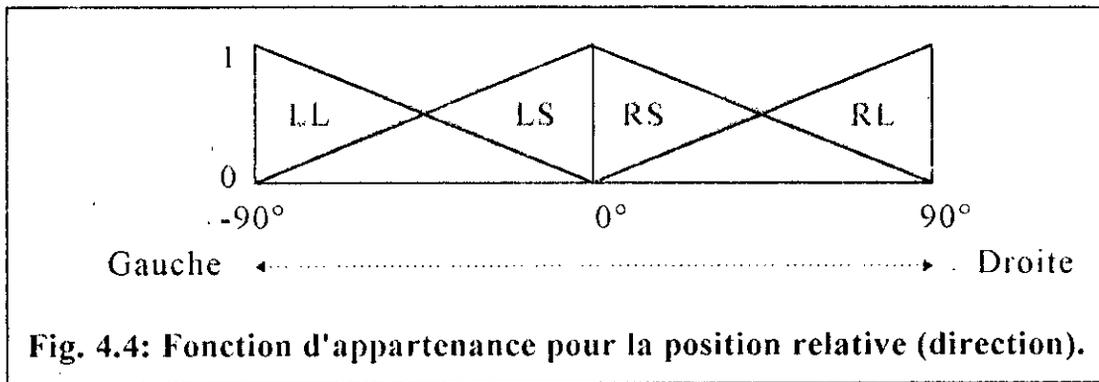


Fig. 4.4: Fonction d'appartenance pour la position relative (direction).

Les quatres labels de logique floue sont donc définis ainsi:

$$LL \implies -90^\circ \leq \theta_s < -45^\circ$$

$$LS \implies -45^\circ \leq \theta_s < 0$$

$$RS \implies 0^\circ \leq \theta_s < 45^\circ$$

$$RL \implies 45^\circ \leq \theta_s < 90^\circ$$

Chaque label montre la direction dans laquelle l'obstacle est identifié.

\*\* On suppose que le robot peut percevoir un obstacle jusqu'à 150cm.

La fonction d'appartenance de distance  $L_s$  peut être exprimée par trois labels de logique floue comme le montre la figure 4.5:

N: près.

M: moyen.

F: loin.

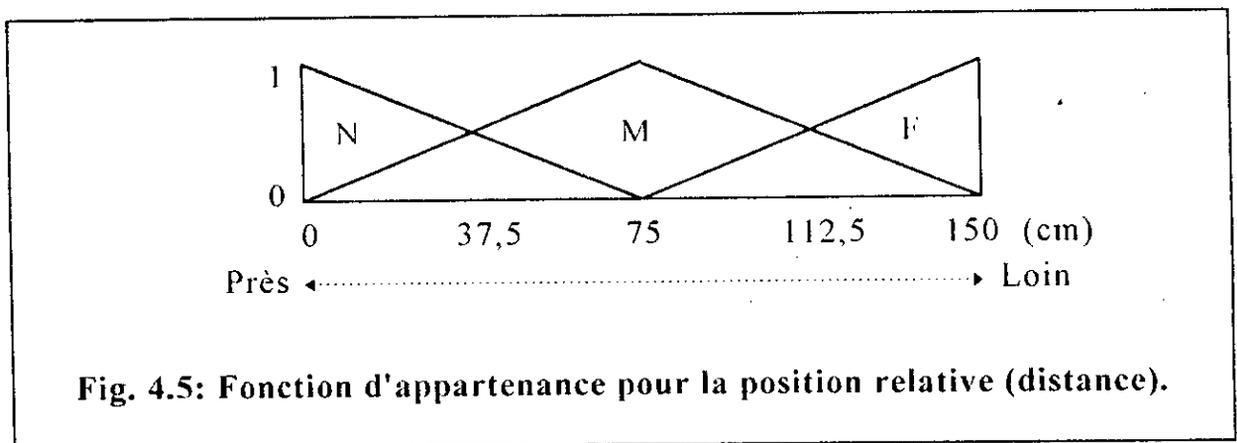


Fig. 4.5: Fonction d'appartenance pour la position relative (distance).

Les trois labels de logique floue sont donc définis ainsi:

$$N \implies 0\text{cm} \leq L_s < 37,5\text{cm}$$

$$M \implies 37,5\text{cm} \leq L_s < 112,5\text{cm}$$

$$F \implies 112,5\text{cm} \leq L_s < 150\text{cm}$$

\*\*\* Les fonctions d'appartenance du degré de danger statique  $\alpha$  sont comprises entre -1 et 1 comme le montre la figure 4.6:

LDL: large danger à gauche.

LDS: petit danger à gauche.

LSS: petite sécurité à gauche.

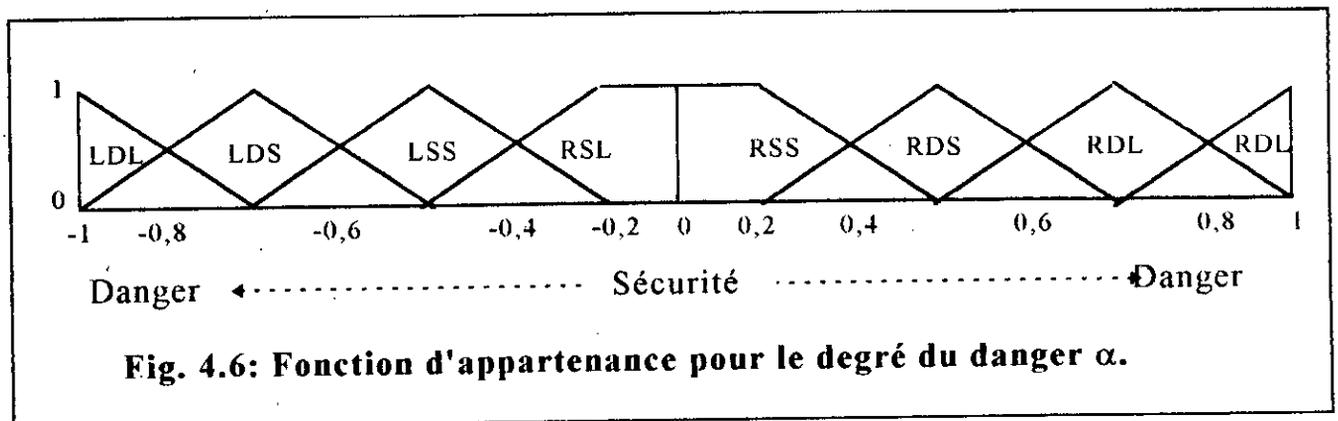
LSL: large sécurité à gauche.

RDL: large danger à droite.

RDS: petite danger à droite.

RSS: petite sécurité à droite.

RSL: large sécurité à gauche.



Les huit labels de logique floue sont donc définis ainsi:

$$LDL \implies -1 \leq \alpha < -0,8$$

$$LDS \implies -0,8 \leq \alpha < -0,6$$

$$LSS \implies -0,6 \leq \alpha < -0,4$$

$$LSL \implies -0,4 \leq \alpha < -0,2 \text{ et } -0,2 \leq \alpha < 0$$

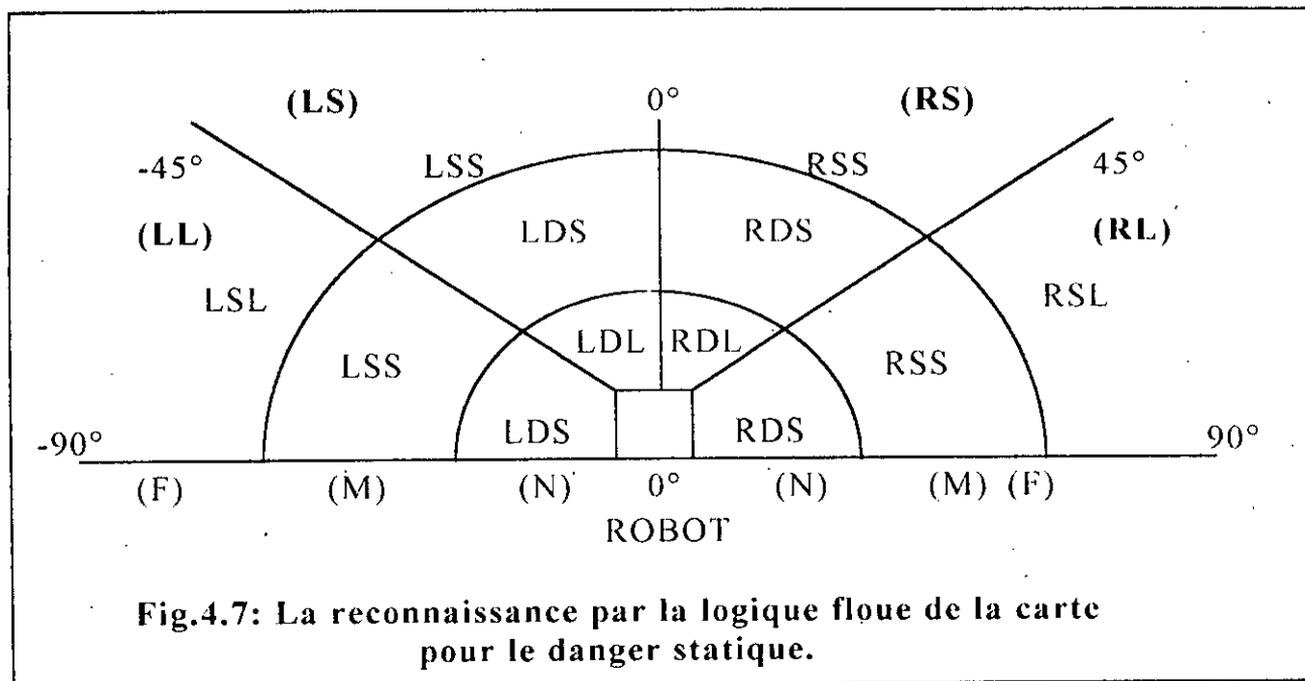
$$RDL \implies 0 \leq \alpha < 0,2 \text{ et } 0,2 \leq \alpha < 0,4$$

$$RDS \implies 0,4 \leq \alpha < 0,6$$

$$RSS \implies 0,6 \leq \alpha < 0,8$$

$$RSL \implies 0,8 \leq \alpha < 1$$

La figure 4.7 montre une reconnaissance par la logique floue du degré de danger statique.



Dans la figure 4.7, les degrés du danger statique sont décidés par douze règles de logique floue.

Ces règles sont décrites par les labels de direction et de distance dans la partie condition de la règle (SI) et les labels de degré du danger statique dans la partie action de la règle (ALORS).

Les douze règles de logique floue choisies pour la navigation sont:

- SI (direction est LL et distance est N) ALORS (danger est LDS),
- SI (direction est LL et distance est M) ALORS (danger est LSS),
- SI (direction est LL et distance est F) ALORS (danger est LSL),
- SI (direction est LS et distance est N) ALORS (danger est LDL),
- SI (direction est LS et distance est M) ALORS (danger est LDS),
- SI (direction est LS et distance est F) ALORS (danger est LSS),
- SI (direction est RS et distance est N) ALORS (danger est RDL),
- SI (direction est RS et distance est M) ALORS (danger est RDS),
- SI (direction est RS et distance est F) ALORS (danger est RSS),

SI (direction est RL et distance est N) ALORS (danger est RDS),  
 SI (direction est RL et distance est M) ALORS (danger est RSS),  
 SI (direction est RL et distance est F) ALORS (danger est RSL).

Le resultat du raisonnement de logique floue basé sur la composition min-max de ces règles conduit au degré  $\alpha$  du danger statique.

## 2.2. DECISION DU DEGRE DU DANGER DYNAMIQUE:

C'est la vitesse relative entre le robot et l'obstacle qui exprime ce degré du danger.

Le vecteur de position relative est défini par:

$$U_0 = R (V_0 - V_r) = [U_{0x}, U_{0y}]^T \quad (2.5)$$

où R est la matrice de transformation définie dans (2.2)

Ce vecteur est alors transformé en un vecteur de vitesse  $U_0'$  basé sur les coordonnées cartésiennes du robot. L'axe des coordonnées de ce repère est confondu avec la direction de l'obstacle.

$$U_0' = R_s U_0 = [U'_{0x}, U'_{0y}]^T \quad (2.6)$$

$$R_s = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix}$$

Ainsi le module et la direction de la vitesse d'un obstacle basé sur les coordonnées polaires du robot, dont l'angle de direction de l'obstacle est défini comme le montre la figure suivante sont données par :

$$W_u = U'^2_{0x} + U'^2_{0y}$$

$$\Phi_u = \pi/2 - \arctg(U'_{0y}/U'_{0x})$$

où  $W_u$ : module de la vitesse d'un obstacle,

$\Phi_u$ : direction de la vitesse d'un obstacle.

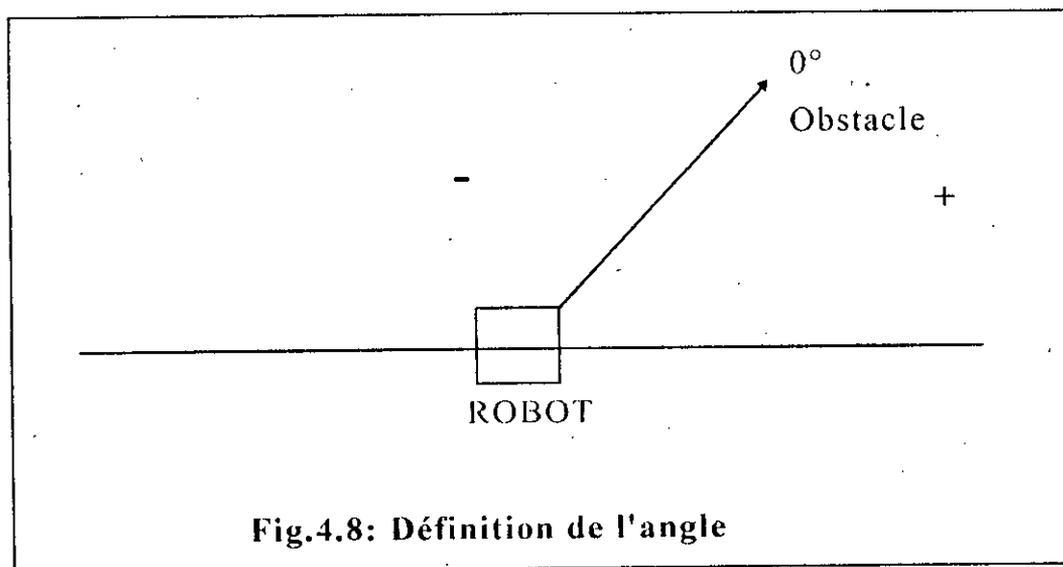


Fig.4.8: Définition de l'angle

\* La fonction d'appartenance de la direction de la vitesse  $\Phi$  basée sur les coordonnées polaires du robot peut être représentée par six labels de logique floue compris entre  $-180^\circ$  et  $180^\circ$  comme le montre la Fig.4.9:

- VLL: large à gauche.
- VLM: moyenne à gauche.
- VLS: petite à gauche.
- VRM: moyenne à droite.
- VRS: petite à droite.
- VRL : large à droite.

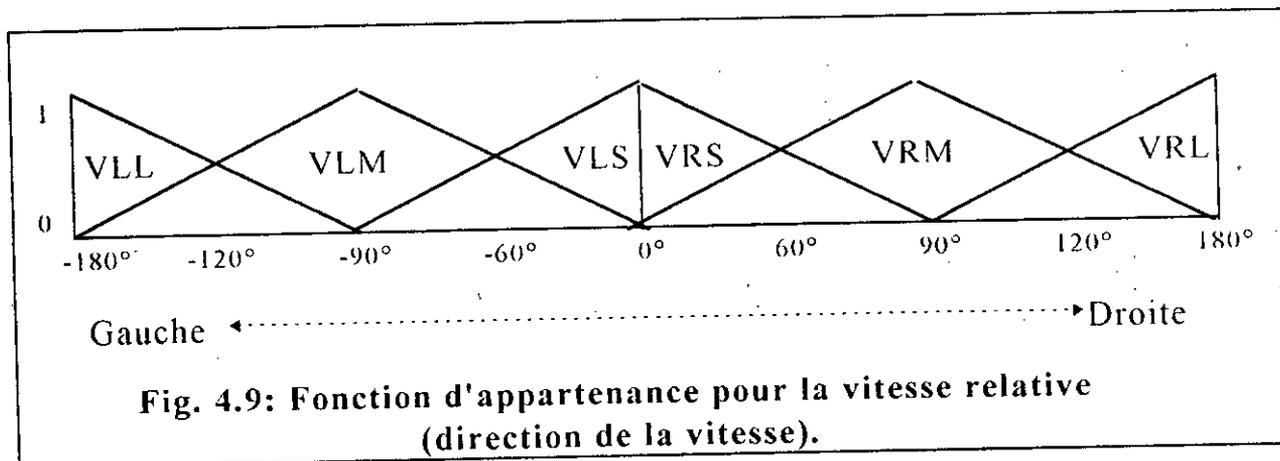


Fig. 4.9: Fonction d'appartenance pour la vitesse relative (direction de la vitesse).

Les six labels de logique floue sont définis ainsi:

$$VLL \implies -180^\circ \leq \Phi_u < -120^\circ,$$

$$VLM \implies -120^\circ \leq \Phi_u < -60^\circ,$$

$$VLS \implies -60^\circ \leq \Phi_u < 0^\circ,$$

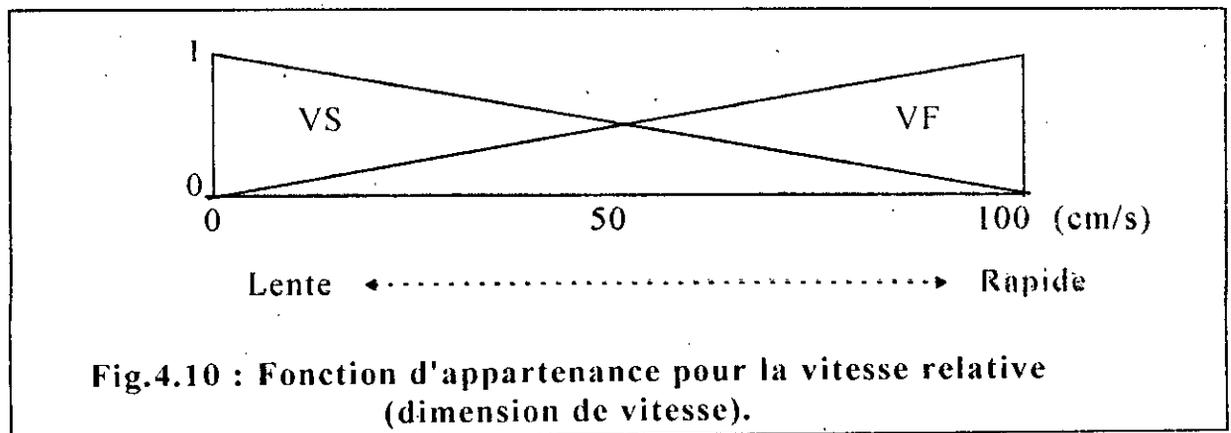
$$VRS \implies 0^\circ \leq \Phi_u < 60^\circ,$$

$$VRM \implies 60^\circ \leq \Phi_u < 120^\circ,$$

$$VRL \implies 120^\circ \leq \Phi_u < 180^\circ,$$

Chaque label montre la direction de la vitesse relative entre le robot et l'obstacle.

\*\* la fonction d'appartenance de dimension de vitesse  $W_u$  est représentée par deux labels de logique floue entre 0cm/s et 100 cm/s comme le montre la figure 4.10 les deux labels sont définis ainsi:



$$VS \implies 0\text{cm/s} \leq W_u < 50\text{cm/s},$$

$$VF \implies 50\text{cm/s} \leq W_u < 100\text{cm/s}.$$

\*\*\* la "fonction nombre" de degré du danger dynamique  $\beta$  est représentée par la même fonction comme celle présentée dans la figure 4.6 celle-ci est comprise entre -1 et 1 avec:

- LDL ==>  $-1 \leq \beta < -0,8$
- LDS ==>  $-0,8 \leq \beta < -0,6$
- LSS ==>  $-0,6 \leq \beta < -0,4$
- LSL ==>  $-0,4 \leq \beta < -0,2$  et  $-0,2 \leq \alpha < 0$
- RSL ==>  $0 \leq \beta < -0,2$  et  $0,2 \leq \alpha < 0,4$
- RSS ==>  $0,4 \leq \beta < +0,6$
- RDS ==>  $0,6 \leq \beta < +0,8$
- RDL ==>  $0,8 \leq \beta < +1$

La figure 4.11 montre une reconnaissance par la logique floue du degré du danger dynamique qui est décidé par douze règles de logique floue.

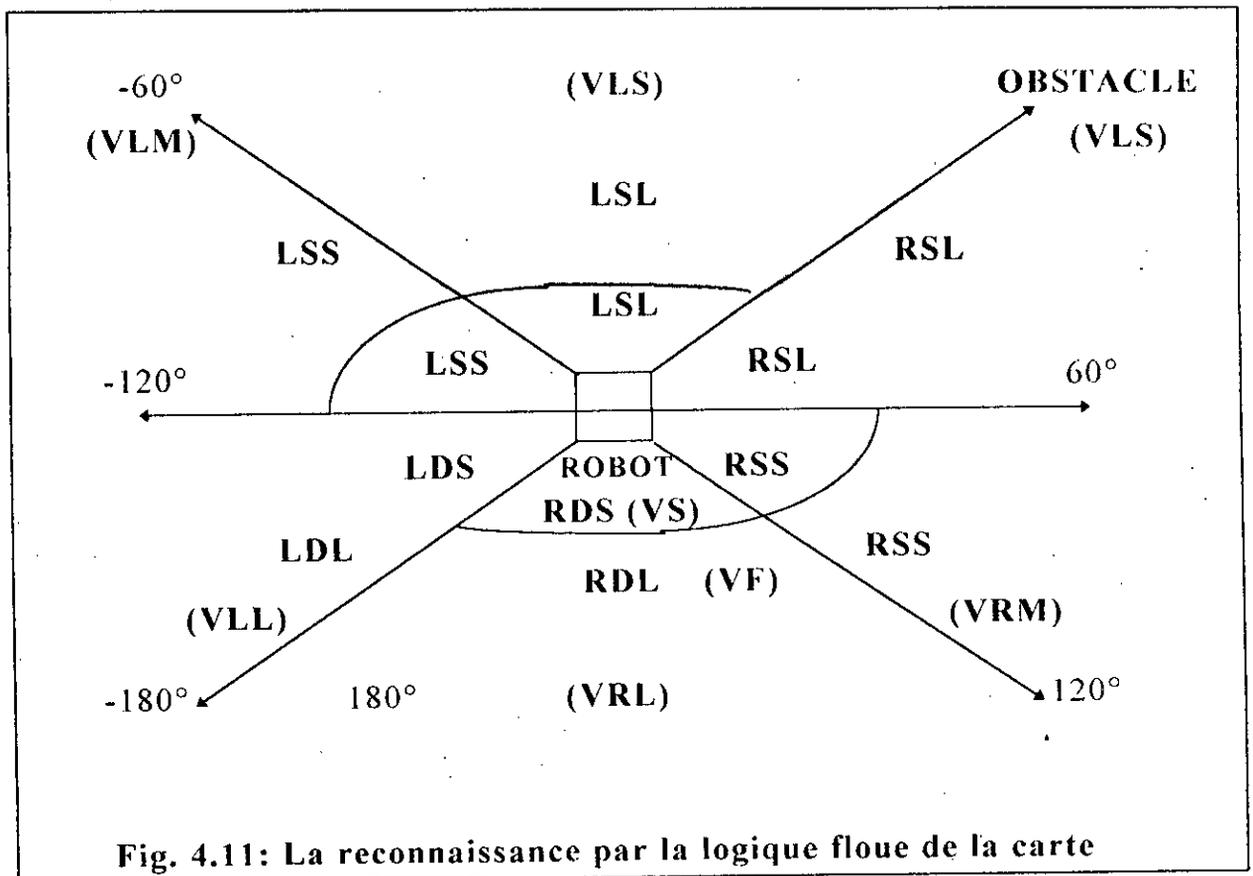


Fig. 4.11: La reconnaissance par la logique floue de la carte pour le danger dynamique

Ces règles sont décrites par des labels de logique floue de la direction et le module de la vitesse (fig. 4.9) et (fig. 4.10) dans la partie condition de la règle (SI), et des labels du degré du danger dynamique dans la partie action de la règle (ALORS).

Les douze règles de logique floue choisies sont :

SI (dir. vit est VLL et dim vit est VN) ALORS (danger est LDS),  
 SI (dir. vit est VLL et dim vit est VF) ALORS (danger est LDL),  
 SI (dir. vit est VLM et dim vit est VS) ALORS (danger est LSS),  
 SI (dir. vit est VLM et dim vit est VF) ALORS (danger est LSS),  
 SI (dir. vit est VLS et dim vit est VS) ALORS (danger est LSL),  
 SI (dir. vit est VLS et dim vit est VF) ALORS (danger est LSL),  
 SI (dir. vit est VRS et dim vit est VS) ALORS (danger est RSL),  
 SI (dir. vit est VRS et dim vit est VF) ALORS (danger est RSL),  
 SI (dir. vit est VRM et dim vit est VS) ALORS (danger est RSS),  
 SI (dir. vit est VRM et dim vit est VF) ALORS (danger est RSS),  
 SI (dir. vit est VRL et dim vit est VS) ALORS (danger est RDS),  
 SI (dir. vit est VRL et dim vit est VF) ALORS (danger est RDL).

Le résultat du raisonnement de logique floue basé sur la composition min-max de ces règles conduit au degré du danger dynamique  $\beta$ .

#### \* Décision du vecteur d'évitement:

La fig.4.12 nous donne la table de décision avec laquelle nous choisissons la direction de l'évitement de l'obstacle.

- L'axe horizontal est rapporté au degré du danger statique  $\alpha$ .
- L'axe vertical est rapporté à la direction de l'évitement de l'obstacle  $\beta$ .

Les nombres dans cette table réfèrent à la direction de l'évitement de l'obstacle. La fig.4.13 montre les onze directions d'évitement, face au robot, qui sont utilisées dans cette table.

Les directions 1+, 2+, 3+, 4+, 5+ signifient une direction à droite respectivement de 15°, 30°, 45°, 60°, 75°; 0 signifie l'orientation zéro.

$\alpha$  Danger ← ..... Sécurité ..... → Danger  
 -1.0 0.8 0.6 0.4 0.2 0 0.2 0.4 0.6 0.8 1.0

$\beta$	-1.0	+3	+2	+1	0	0	0	-2	-3	+4	+5
Danger	-0.8	+2	+1	0	0	0	0	-1	-1	+3	+4
↑	-0.6	+1	0	0	0	0	0	0	0	-1	-2
Sécurité	-0.4	0	0	0	0	0	0	0	0	0	-1
↓	-0.2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0.2	0.2	+1	0	0	0	0	0	0	0	0	0
0.4	0.4	+2	+1	0	0	0	0	0	0	0	-1
0.6	0.6	-4	-3	+1	+1	0	0	0	0	-1	-2
Danger	0.8	-5	-4	+3	+2	0	0	0	-1	-2	-3
	1.0										

Fig. 4.12: Table de décision pour le vecteur d'évitement  $O$

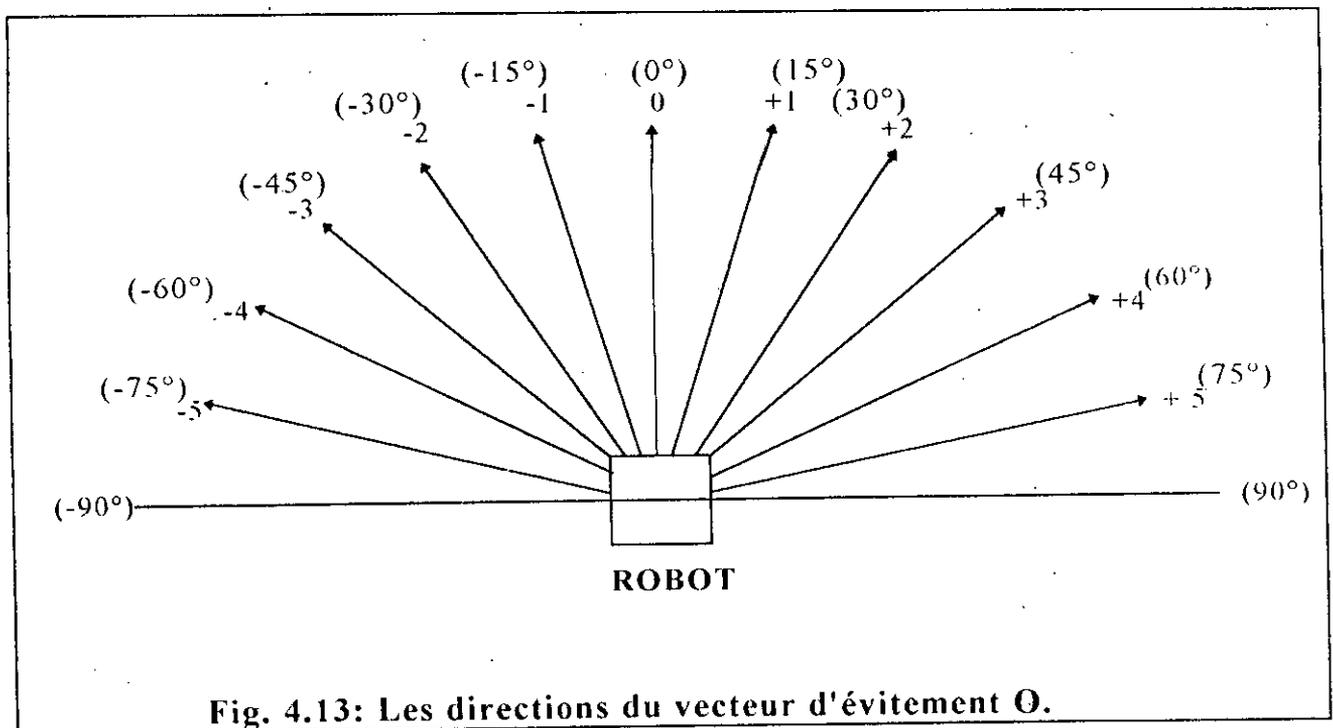


Fig. 4.13: Les directions du vecteur d'évitement  $O$ .

La stratégie est représentée avec les règles de production SI-ALORS. La direction de l'évitement décidée par cette table, est donnée comme un vecteur unité basé sur les coordonnées cartésiennes du robot. Ce vecteur unité devient le vecteur d'évitement  $O$ .

L'opérateur peut exprimer sa stratégie d'évitement avec la table de décision par l'utilisation de deux types de labels de logique floue ( $\alpha$  et  $\beta$ ) obtenus à partir des règles de logique floue; et peut facilement changer la performance de l'évitement par la réécriture de ces règles de logique floue, donc de la table de décision.

\* décision du vecteur cible et du vecteur d'orientation quand le robot fait face au point désiré, le vecteur cible  $D$  (le vecteur unité) est défini comme suit:

$$D = (P_c - P_r) / |P_c - P_r| \quad (2.9)$$

Par l'utilisation de ce vecteur cible  $D$  et du vecteur d'évitement  $O$ , le vecteur d'orientation  $M$  (vecteur unité est décidé comme suit:

$$M = (D + O) / |D + O| \quad (2.10)$$

**N.B:** La position et la vitesse de l'obstacle sont déterminées après un traitement de données venant des capteurs.

*Chapitre V*

*SIMULATION*

Nous interprétons dans ce chapitre quelques résultats obtenus par les deux méthodes:

- Méthode d'évitement d'obstacles classique (recherche de chemin est effectué par l'algorithme A\*).
- Méthode d'évitement d'obstacles par la logique floue.

Sur les figures qui suivent:

- La cible est représentée par les deux cercles concentriques.
- Le robot est représenté par le grand cercle.
- L'obstacle est représenté par le petit cercle.

### interprétation des résultats:

Figure 1: Le résultat de la simulation dans ce cas montre le comportement du robot en présence d'un obstacle qui s'approche de lui de droite à gauche.

$$V_{rx} = 30\text{cm/s}$$

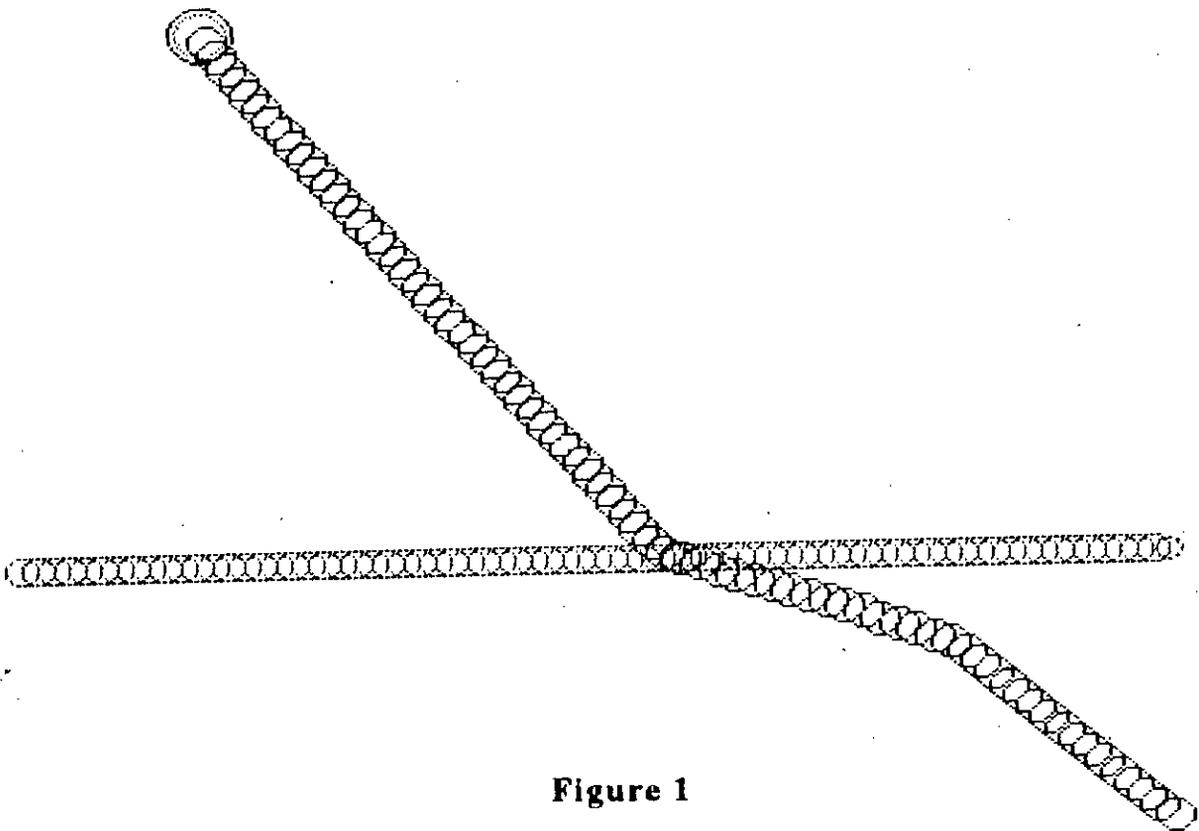
$$V_{ry} = 30\text{cm/s}$$

$$V_{ox} = 0$$

$$V_{oy} = -30\text{cm/s}$$

$$\text{Temps d'échantillonnage} = 0,01\text{s}$$

dans cette figure on constate que le robot passe au devant de l'obstacle lorsque la vitesse de ce dernier est lente par contre il serait passé derrière lui si la vitesse de l'obstacle est supérieur à celle du robot.



**Figure 1**

Figure 2: Dans ce cas les paramètres sont les suivants:

$$V_{rx} = 30\text{cm/s}$$

$$V_{ry} = 0\text{cm/s}$$

$$V_{ox} = 30$$

$$V_{oy} = 0\text{cm/s}$$

$$\text{Temps d'échantillonnage} = 0,01\text{s}$$

Sur cette figure on voit comment réagit le robot quand l'obstacle vient droit sur lui, cette méthode a pour rôle de faire réagir le robot comme aurait réagit un être humain dans pareil cas c'est à dire, lorsqu'un obstacle survient il faut que le robot l'évite souplement sans s'éloigner pour autant de lui.

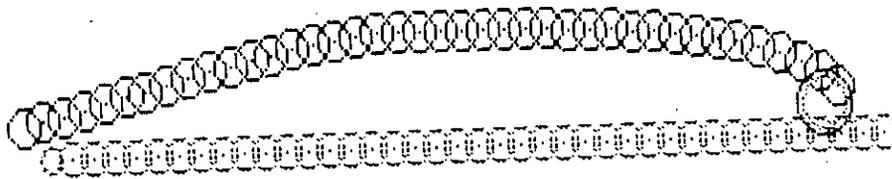
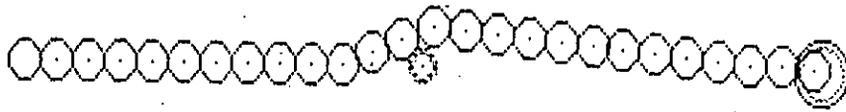
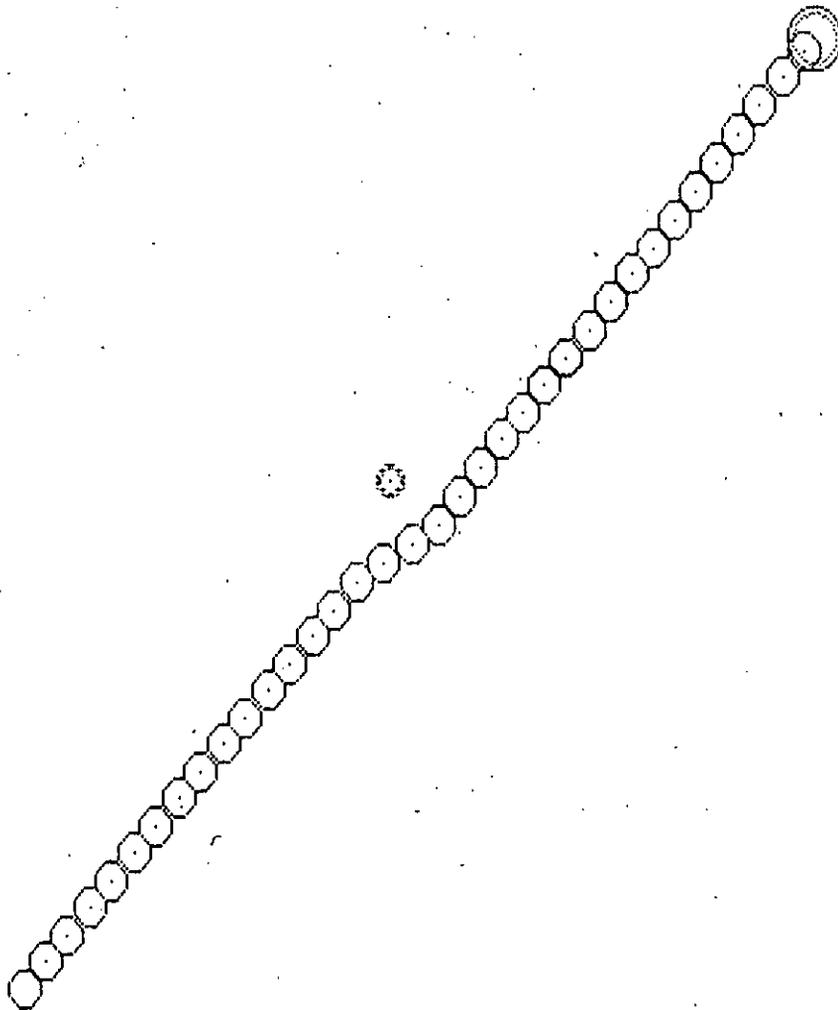


Figure 2.



**Figure 3: Cas où l'obstacle est fixe (exemple 1).**



**Figure 4: Cas où l'obstacle est fixe (exemple 2).**

PS	1								
	2	3							
		4							
		5	6						
			7	8					
				9					
				10					
				11	12	13	14	15	
								16	17
									PC

Figure 5: Chemin trouvé par l'algorithme A\*

PS: point source.

PC: point cible.

Cases vides: espace libre de recherche.

Cases pleines: représentent les obstacles.

*CONCLUSION  
GENERALE*

Nous avons présenté à travers ce travail deux méthodes de navigations:

- Méthode classique basé sur les algorithmes d'intelligence artificielle.
- Méthode par la logique floue.

Afin de résoudre un problème concret qui est "la navigation d'un robot mobile" pour ce qui concerne la méthode classique le chemin est généré par un planificateur global grâce à ce dernier nous somme passé d'un site réel à un modèle de calcul. Nous avons par ce fait ramener la recherche de chemin de ce modèle, la trajectoire ainsi obtenue donne les cellules de passages qui constituent les chemin du robot.

La méthode par la logique floue nécessite la conception à partir de la notion de floue approchant le processus traitement de l'information chez l'être humain.

L'utilisation de la logique floue évite le module de modélisation de l'environnement, celle-ci s'avère très pratiques au cas où les obstacles sont mobiles.

Nous estimons toute fois qu'il serait très intéressant de pousser encore plus loin la méthode classique en abordant les point suivant:

- nous avons émis comme hypothèse que les obstacles sont définis à partir des noeuds de la grille. Cette contrainte peut être levée en utilisant le modèle polygonale.
- Etendre la navigation pour un environnement 3D, en utilisant la projection du modèle sur le plan 2D dans ce cas le calcul se ramène à plusieurs grilles juxtaposées.

- Il est clair que l'atout majeur qui justifie le choix d'une planification globale est la génération de trajectoire optimale. Toutefois que le robot puisse réagir à une éventuelle évolution de l'environnement. Il est indispensable d'associer au planificateur global un planificateur local, ainsi grâce à ce dernier un obstacle imprévu est contourné.
- La génération du système à  $n$  obstacles par une approche fondée sur les concepts de logique floue sous forme d'un système expert, en calculant pour chaque obstacle les degrés du danger statique et dynamique et l'évitement qui lui est associé, enfin il reste à résoudre le conflit entre plusieurs évitements pour sélectionner l'action finale.

*ANNEXE*

### Algorithme A\* [RIC 87]:

Une étape de l'algorithme consiste à mettre à jour deux listes de noeuds du graphe. La liste Open dite liste « ouverte », comprend les noeuds atteints mais non encore développés, c'est à dire dont on n'a pas examiné les voisins. La liste Close, dite liste fermée, est la liste des noeuds développés.

Précisons les notations employées. Nous notons  $c(n, n')$  le coût de l'arc  $nn'$ . En un noeud  $n$  développé tel qu'il existe au moins un chemin de  $n_0$  à  $n_f$  passant par ce noeud, on écrit par  $g(n)$  la longueur du plus court chemin de  $n_0$  à  $n$ ,  $h(n)$  la longueur du plus court chemin de  $n$  à  $n_f$ , et  $f(n)$  la somme de  $g(n) + h(n)$ .

A une étape de l'algorithme, nous ne connaissons qu'une estimation de  $g(n)$ , notée  $g^*(n)$ , le plus court chemin de  $n_0$  à  $n$  parmi ceux que l'on a déjà examinés. De plus, on ne peut évaluer la distance au but que de façon heuristique. On note  $h^*(n)$  l'estimation par exemple, si  $s(n, n')$  mesure la distance euclidienne entre deux noeuds voisins, on utilisera la distance d'un noeud au noeud but comme heuristique.

Au cours de l'algorithme on maintient pour chaque noeud développé un pointeur vers son prédécesseur sur le meilleur chemin qui y conduit.

L'algorithme est donné comme suit :

**Début**

Initialiser Open à  $n_0$  et calculer  $f(n_0)$ ;

Initialiser Closed à  $\emptyset$ ;

**Tantque** (Open  $\neq \emptyset$ ) and (not succès)

Choisir un noeud  $n$  de Open pour lequel  $f$  est minimum;

**Si** ( $n=n_t$ )

Succès = vrais;

Objectif atteint;

**Sinon**

**Pour** tout successeur  $n'$  de  $n$

$f = g^*(n) + c(n, n') + h^*(n')$ ;

**Si** ( $n' \in \text{Open}$ ) et ( $n' \notin \text{Closed}$ )

Ajouter  $n'$  à open avec  $f^*(n') = f$ ;

Créer un pointeur de  $n'$  vers  $n$ ;

**Sinon**

**Si** ( $f < f^*(n)$ )

$f^*(n) = f$ ;

Remplacer le pointeur attaché à  $n'$  par un pointeur vers  $n$ .

**Si** ( $n' \in \text{Closed}$ )

Déplacer  $n'$  vers Open

**Finsi**

**Finsi**

**Finpour**

**Fintanque**

**Finsi**

**Fintanque**

**Fin**

## BIBLIOGRAPHIE

[BAU 81] G. BAUZIL, M. BRIVOT, P. RIVERS. "A navigation sub-system using ultrasonic sensors for the mobile robot HILARE".  
Int conference on Ravise C, Straft Ford Upon-Avon 1.3  
Avril 1981 p-p-47-5.

[BRO 82] Brooks R.A "Solving the Find-Path problem by good representation of free space".  
IEEE transaction on SMC, Mars 82 volume 13, n°3  
pp 190-197

[CAH 75] D.F.CAHN, S.R PHILIPS, ROBNV. "A Range based navigation and obstacle avoidance algorithm".  
IEEE transaction on systems, Mens and cybernetics,  
Septembre 75, p\_p 544-551

[CAM 90] Valerio BOSCHIAN-CAMPANER. "Modélisation de l'environnement par grille adaptative et recherche de chemins pour robot mobile" 1990 thèse de doctorat inge de l'université de METZ, option: production automatisée.

[CHU a 90] et [CHU b 90] CHUEN CHIEN LEE 1990.  
Fuzzy logic in Control systems: Fuzzy logic Controller part 1 et part 2.  
IEEE transactions on systems

[CIL 88] M. KEMAL Ciliz C. Hsik 1988  
Fuzzy rule-based motion controller for an autonomous mobile robot.  
SYRACUSE UNIVERSITY

- [GIR 83] Georges GIRALT "Mobile Robots": publication L.A.A.S  
N°2866, Juillet 83.
- [HEN 83] HENRI place, Michel Jullière et Lionnel Marcé, "Qu'en est-il  
des robots mobiles? Utilité et utilisation des robots  
expérimentaux "Nouvel Automatismes" 1983.
- [HOE 76] JH HOEL " Some variations of lee's Algorithm"  
IEEE transactions on Computer vol.c25 N°1 Janvier 1976  
p-p 19-24.
- [JUL 84] Michel Jullière "Des Vehicules guidés aux robots mobile:  
contribution à l'accroissement de leur autonomie".  
Thèse de Doctorat d'état, Institut des Sciences appliquées de  
Renne et l'université de Renne 1984.
- [KAH 86] "Real Time obstacle avoidance for manipulators and mobile  
robots". Int Journal of robotics research, 5(2): pp-90-99  
printemps 86.
- [KHA 80] O.KHATIB "commande dynamique dans l'espace operationnel  
des robots manipulateurs en présence d'obstacles". Thèse de  
docteur-ingenieur, école supérieure de l'aéronautique de  
l'espace (France) 1980.
- [LEE] K.S.FU, R.C. GONZALEZ, C.S.G. LEE. "Robotics: Control,  
sensing, vision and intelligence".
- [LOZ 83] Lozano-Perez "Spatial planning à configuration space approach".  
IEEE transaction on computer 32(2) 1983.

[LUM 86] LUM Skey et A.Stepanov, "Dynamic path planning, for a mobile automation with limited information on the environnement".  
IEEE transaction on automatic control, 31, Novembre 1986.

[MAE 90] Yoichiro Maeda 1990 ."collision avoidance control. among obstacles for a mobile robot on the fuzzy reasoning".  
Laboratory for international fuzzy engineering research

JAPAN

[MEM] M.Hadad S.Abdedaim "évitement de collision pour robot mobile en temps réel et dans un environnement dynamique" USTHB.

[NIL 80] NILSSON N.J "principales of artificiel intelligence".Tioger pub, palmo, alto, Californie USA 80.

[PLA 83a] HENRI PLACE, MICHEL JULLIERE, LIONNEL MARCE.  
"Qu'en est-il des robots mobiles: utilités, utilisation, robots experimentaux". Nouvel automatisme 1983.

[PLA 83b] HENRI PLACE, MICHEL JULLIERE, LIONNEL MARCE.  
"Qu'en est-il des robots mobiles: tactiques de navigation".  
Nouvel automatisme 1983.

[RAJ 81] Raja Chatila "Systeme de navigation pour un robot mobile autonome, modélisation et processus decisionnels" thèse de docteur-ingénieur, V.P.S. Toulouse, 8 Juillet 1981.

[REC 85] "intelligence artificielle". Recherche N°170, Octobre 1985.

[RIC 87] "intelligence artificielle".  
The university of Texas at Austin Masson 1987

[RUB 74] FRANK RUBIN "The LEE path connection algorithm".

IEEE transactions on computers, vol C.23 N°9  
Septembre 74.

[SIM 89] T.SIMEON."Génération automatique de trajectoire sans collision  
et planification de tâches de manipulation en robotique".

Thèse de doctorat de l'université de Paul-Sabatier de Toulouse  
Rapport L.A.A.S. N°89.002 1989.

[TOU 88] Pierre Tournassoud". Planification de trajectoire en Robotique:  
Complexité et approche pratique".

Thèse de doctorat de l'université de Paris-Sud, centre d'Orsay,  
spécialité informatique, INRIA.

[TOU 91] Pierre Tournassoud, "Géométrie des Robots et intelligence  
artificielle". 1991.

[VER 92] J-L.Vern. "La logique floue: Concepts et definitions".

[VIO 93] Gret Viot, "Fuzzy logic concept to construct".

I.A expert Novembre 1993.

[YAG 83] RONALD YOGER 1983

"Robot planning with fuzzy sets".

[ZAD 92] LOTFI A.ZADEH 1992

"The calculus of fuzzy IF/THEN rules".