

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITÉ FERHAT ABBAS –SETIF 1-  
UFAS (ALGERIE)

# THÈSE

Présentée à la faculté de technologie

Département d'Electronique

Pour l'obtention du diplôme de

**Doctorat en Sciences**

Par :

***HAMANI MILOUD***

**Thème**

***Navigation des robots mobiles non-holonomes  
sous contrôle flou***

Soutenue le 11 / 12 / 2016 devant le jury composé de :

Pr F. KRIM	Professeur à l'université de Sétif 1	Président
Pr A. HASSAM	Professeur à l'université de Sétif 1	Rapporteur
Pr A. BARTIL	Professeur à l'université de Sétif 1	Examineur
Pr N. BENOUDJIT	Professeur à l'université de Batna	Examineur
Pr B. MENDIL	Professeur à l'université de Bejaia	Examineur
Pr T. REKIOUA	Professeur à l'université de Bejaia	Examineur

# *Dédicaces*

*Je dédie cette thèse  
à mes parents,  
à ma femme et mes filles,  
à mes frères et sœurs,  
et à tous ceux que j'aime.*

## *Remerciements*

Après louange à Allah,

Je tiens, tout d'abord, à exprimer mes vifs remerciements à mon encadreur le Professeur HASSAM ABDELOUAHAB pour m'avoir inspiré le sujet de cette thèse et pour m'avoir guidé et conseillé tout au long de l'élaboration de ce travail.

Je suis profondément reconnaissant à Monsieur F. KRIM (Professeur à l'Université de Sétif 1) pour m'avoir fait l'honneur de présider le jury de soutenance.

Je tiens aussi à exprimer mes remerciements et mes sentiments les plus respectueux à Monsieur A. BARTIL (Professeur à l'Université de Sétif 1) pour avoir accepté d'examiner ce travail.

Mes remerciements les plus sincères vont également à l'égard de Monsieur N. BENOUDJIT (Professeur à l'Université de Batna), Monsieur B. MENDIL (Professeur à l'Université de Bejaia) et Monsieur T. REKIOUA (Professeur à l'Université de Bejaia) pour l'intérêt qu'ils ont bien voulu porter à ce travail en acceptant de l'examiner.

Enfin, je n'oublie pas d'adresser mes vifs remerciements à toute ma famille, qui m'a accompagné tout au long de mes études par son soutien permanent.

*Hamani Miloud*

*Table des matières.*

Dédicaces .....	i
Remerciements .....	ii
Table des matières.....	iii
Listes des figures .....	vi
Liste des tableaux.....	ix

**Chapitre I : Introduction générale et état de l’art**

I.1 Introduction générale.....	01
I.2 Etat de l’art .....	03
I.3 Structure de la thèse .....	06

**Chapitre II : la robotique mobile**

II.1 Introduction .....	08
II.2 Définition d'un robot mobile .....	08
II.3 Autonomie d’un robot mobile .....	09
II.4 Les différents types de terrains .....	10
II.5 Modélisation des robots mobiles .....	13
II.5.1 Classification des types des roues .....	13
II.5.2 Types des plates formes mobiles .....	14
II.5.2.1 Plates formes non holonomes .....	15
II.5.2.2 Plates formes holonomes .....	17
II.5.3 Roulement avec ou sans glissement .....	17
II.6 Détection des obstacles et localisation .....	18
II.6.1 Détection des obstacles et cartographie .....	19
II.6.2 Fusion de données multi-capteurs et cartographie .....	21
II.6.3 Localisation d’un robot mobile .....	23
II.6.4 Localisation et cartographie simultanées .....	27
II.7 Les thématiques scientifiques de la robotique mobile .....	28
II.8. Conclusion .....	29

**Chapitre III : Approches de navigation en environnement inconnu**

III.1 Introduction .....	30
III.2 Approches délibératives .....	31
III.2.1 Méthodes par graphes .....	33
III.2.1.1 Graphe de visibilité .....	34
III.2.1.2 Diagramme de voronoi.....	35
III.2.1.3 Décomposition en cellules .....	36
III.2.1.4 Echantillonnage par grilles régulières .....	37
III.2.1.5 Cartes de routes probabilistes .....	38
III.2.1.6 Méthodes de replanification .....	40

III.2.2 Méthodes par arbres .....	41
III.2.2.1 Exploration rapide des arbres aléatoires (RRT).....	41
III.2.2.2 Fil d'Ariane .....	42
III.3 Approches réactives .....	44
III.3.1 Approches par champs de potentiel .....	44
III.3.2 Histogramme de champs de vecteurs .....	45
III.3.3 Navigation par diagrammes de proximité .....	47
III.3.4 Méthode de navigation courbure-vitesse .....	48
III.3.5 Fenêtre dynamique .....	49
III.3.6 Représentation des obstacles dans l'espace des vitesses .....	51
III.3.7 Le concept d'état de collision inévitable .....	52
III.3.8 Planification de mouvement partiel .....	53
III.3.9 défaut de convergence vers le but .....	54
III.4 Méthode de déformation de mouvement .....	55
III.5 Conclusion .....	56

### **Chapitre IV : Navigation autonome sous contrôle flou**

IV.1 Introduction .....	57
IV.2 Description de l'architecture de navigation autonome proposée .....	58
IV.2.1 Modèle du robot mobile .....	58
IV.2.2 Structure de l'architecture de contrôle proposée .....	60
IV.2.2.1 Module de perception .....	62
IV.2.2.2 Module de décision .....	63
IV.2.2.3 Module d'action .....	68
IV.2.3 Génération automatique des règles floues par les algorithmes génétiques .....	70
IV.2.3.1 Approche connexionniste .....	71
IV.2.3.2 Approche directe .....	71
IV.2.3.3 Conception d'un contrôleur flou par les algorithmes génétiques....	71
IV.2.3.4 Définition des paramètres à optimiser .....	73
IV.2.3.5 Codage des paramètres .....	73
IV.2.3.6 Fonction d'objectif .....	74
IV.2.3.7 Exécution de l'algorithme d'optimisation .....	76
IV.3 Résultats de simulation .....	79
IV.3.1 Comportement de navigation vers une cible .....	79
IV.3.2 Comportement d'évitement d'obstacles .....	81
IV.3.3 Comportement de suivi de murs .....	82
IV.4 Conclusion .....	84

### **Chapitre V : Concepts des obstacles virtuels pour la résolution du problème des minima locaux**

V.1 Introduction .....	86
V.2 Architecture de l'approche de navigation autonome proposée .....	87
V.2.1 Module de décision .....	88
V.2.2 Module d'action .....	89
V.2.3 Conception de l'algorithme des obstacles virtuels .....	92

V.2.4 Génération des bases de règles par l’algorithme génétique .....	93
V.3 Résultats de simulation .....	96
V.3.1 Critères de comparaison .....	96
V.3.2 Résultats et comparaisons .....	97
V.3.2.1 Résultats de simulation dans l’environnement type1 .....	97
V.3.2.2 Résultats de simulation dans l’environnement type2 .....	100
V.4 Conclusion .....	103
<b>Conclusion générale et perspectives.....</b>	<b>105</b>
<b>Références bibliographiques .....</b>	<b>109</b>

## Liste des figures.

### Chapitre II

<b>Figure II.1</b> Le robot taxi .....	10
<b>Figure II.2</b> Le robot martien Rover.....	11
<b>Figure II.3</b> (a) Robot sous marin, (b) Drone de surveillance.....	11
<b>Figure II.4</b> Modélisation du robot dans le repère absolu .....	12
<b>Figure II.5</b> Roue centrée et roue décentrée .....	13
<b>Figure II.6</b> Roue troncosphérique et roue suédoise .....	14
<b>Figure II.7</b> Equivalence modèle voiture et tricycle .....	15
<b>Figure II.8</b> Modèle de type tricycle à roues différentielles .....	16
<b>Figure II.9</b> Le robot omnidirectionnel Nomadic XR4000 .....	17
<b>Figure II.10</b> Roulement sur le sol .....	17
<b>Figure II.11</b> Localisation à l'estime .....	24
<b>Figure II.12</b> Localisation absolue (méthode par triangulation) .....	25

### Chapitre III

<b>Figure III.1</b> Bras manipulateur dans une industrie automobile .....	32
<b>Figure III.2</b> Chemin déterminé entre deux configurations à partir d'un graphe de visibilité .....	34
<b>Figure III.3</b> Illustration de la définition d'un diagramme de Voronoï.....	35
<b>Figure III.4</b> Exemples des décompositions en cellules .....	36
<b>Figure III.5</b> Chemin planifié pour un robot de type différentiel à partir d'une roadmap calculée dans l'espace de configuration du robot .....	39
<b>Figure III.6</b> Exploration rapide des arbres aléatoires : Etapes successives de construction de l'arbre de recherche dans l'espace des configurations du système robotique .....	42
<b>Figure III.7</b> Fil d'Ariane : Méthode de planification alternant deux algorithmes. ....	43
<b>Figure III.8</b> Calcul d'un chemin entre deux configurations $q_0$ et $q_f$ par la méthode de champs de potentiels .....	44
<b>Figure III.9</b> Représentation de l'environnement sous forme de grille d'occupation centrée autour du robot .....	46
<b>Figure III.10</b> Histogramme polaire calculé à partir de la grille d'occupation..	46
<b>Figure III.11</b> Navigation par Diagrammes de Proximité (ND) .....	47
<b>Figure III.12</b> Méthode de navigation courbure-vitesse (CVM) .....	48
<b>Figure III.13</b> Fenêtre Dynamique (DW) .....	50
<b>Figure III.14</b> Représentation des obstacles dans l'espace des vitesses (VO) ....	51
<b>Figure III.15</b> Concept de collision inévitable dans l'espace de travail $W$ : (a) la cinématique du robot, (b) la trajectoire de l'obstacle mobile	52
<b>Figure III.16</b> Concept de collision inévitable dans l'espace de travail $W$ avec extension de l'horizon temporel .....	53
<b>Figure III.17</b> Planification de mouvement partiel .....	54
<b>Figure III.18</b> Déformation de chemin .....	55

## Chapitre IV

<b>Figure IV.1</b> La configuration cartésienne du robot .....	59
<b>Figure IV.2</b> Architecture basée sur les comportements .....	60
<b>Figure IV.3</b> Architecture de contrôle proposée .....	61
<b>Figure IV.4</b> Les variables utilisées dans la méthode proposée .....	63
<b>Figure IV.5</b> Les fonctions d'appartenance du coordinateur des comportements: (a, b, c) Les variables d'entrée $d_L, d_R, d_F$ .....	65
<b>Figure IV.6</b> Les fonctions d'appartenance du coordinateur des comportements: La variable de sortie $B_v$ .....	65
<b>Figure IV.7</b> Exemple d'une situation de navigation vers la cible .....	66
<b>Figure IV.8</b> Exemple d'une situation d'évitement d'obstacles .....	67
<b>Figure IV.9</b> Exemple d'une situation de suivi de murs .....	68
<b>Figure IV.10</b> Les fonctions d'appartenance du contrôleur de mouvement: (a,b) Les variables d'entrée $d_F, d_t$ .....	69
<b>Figure IV.11</b> Les fonctions d'appartenance du contrôleur de mouvement: (a,b) Les variables de sortie $w_l, w_r$ .....	70
<b>Figure IV.12</b> Structure du système de navigation à base de la logique floue hybride .....	72
<b>Figure IV.13</b> Etapes de conception d'un contrôleur flou .....	72
<b>Figure IV.14</b> Navigation rectiligne vers la cible .....	79
<b>Figure IV.15</b> Rotation à droite et navigation rectiligne vers la cible .....	80
<b>Figure IV.16</b> Rotation à gauche et navigation rectiligne vers la cible .....	80
<b>Figure IV.17</b> Evitement d'un seul obstacle puis navigation vers la cible .....	81
<b>Figure IV.18</b> Evitement de plusieurs obstacles puis navigation vers la cible ...	82
<b>Figure IV.19</b> Evitement d'un mur et de plusieurs obstacles puis navigation vers la cible .....	82
<b>Figure IV.20</b> Suivi de murs, évitement d'un obstacle puis navigation vers la cible .....	83
<b>Figure IV.21</b> Suivi de murs, évitement de plusieurs obstacles puis navigation vers la cible .....	83
<b>Figure IV.22</b> Situation de blocage en boucle infini en présence d'un obstacle concave .....	84

## Chapitre V

<b>Figure V.1</b> Architecture du système de navigation autonome proposée.....	87
<b>Figure V.2</b> Les fonctions d'appartenance du coordinateur des comportements: (a, b, c) Les variables d'entrée $d_L, d_R, d_F$ .....	89
<b>Figure V.3</b> Les fonctions d'appartenance du coordinateur des comportements: La variable de sortie $B_v$ .....	90
<b>Figure V.4</b> Les fonctions d'appartenance du contrôleur de mouvement: (a,b) Les variables d'entrée $d_F, d_t$ .....	91
<b>Figure V.5</b> Les fonctions d'appartenance du contrôleur de mouvement: (a,b) Les variables de sortie $w_l, w_r$ .....	91
<b>Figure V.6</b> Principe de l'algorithme des obstacles virtuels .....	93

<b>Figure V.7</b>	Navigation par la méthode des cibles virtuelles en absence de minimum local : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite .....	98
<b>Figure V.8</b>	Navigation par la méthode des cibles virtuelles en présence de minimum local : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite .....	98
<b>Figure V.9</b>	Navigation par la méthode comportementale proposée en absence de minimum local : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite .....	99
<b>Figure V.10</b>	Navigation par la méthode comportementale proposée en présence de minimum local : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite .....	100
<b>Figure V.11</b>	Navigation par la méthode des cibles virtuelles en absence de minimum local profond : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite .....	101
<b>Figure V.12</b>	Navigation par la méthode des cibles virtuelles en présence de minimum local profond: (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite .....	101
<b>Figure V.13</b>	Navigation par la méthode proposée en absence de minimum local profond: (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite .....	102
<b>Figure V.14</b>	Navigation par la méthode proposée en présence de minimum local profond: (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite .....	102

*Liste des tableaux.*

**Chapitre IV**

<b>Tableau IV.1:</b> Codage binaire des règles floues .....	74
<b>Tableau IV.2</b> La base des règles floues du module de décision .....	77
<b>Tableau IV.3</b> La base des règles floues du module d'action .....	78

**Chapitre V**

<b>Tableau V.1</b> La base des règles floues du module de décision .....	94
<b>Tableau V.2</b> La base des règles floues du module d'action .....	95
<b>Tableau V.3</b> Comparaison des performances entre les deux types de contrôle....	103

## **Chapitre I :**

### **INTRODUCTION GENERALE ET ETAT DE L'ART**

#### **I.1 Introduction générale**

Les robots mobiles feront bientôt une partie intégrante de notre vie quotidienne. Le progrès en robotique mobile s'est avancé en un point où il devient possible de construire des robots qui peuvent agir dans nos bureaux, maisons, écoles, hôpitaux et laboratoires de recherches. Les robots rempliront bientôt plus de rôles que leurs applications courantes en tant qu'assembleurs industriels et jouets d'enfants. Nous avons maintenant atteint un point où nous pouvons concevoir des robots qui peuvent agir en tant qu'associés avec nous (enseignants, assistants de santé, assistants domestiques, chirurgiens, acteurs et collaborateur scientifiques) au lieu d'être simplement un outil ou un jouet ! Cela signifie que nous pouvons nous attendre à ce que le robot mobile agisse et réagisse d'une manière autonome comme un être humain, par exemple, comprendre nos directions pour accomplir une tâche, nous guider à apprendre quelques informations et nous aider quand nous aurions besoin d'un coup de main.

Dans le cadre de cette problématique, Des recherches intensives ont été réalisées au cours de ces dernières années dont le but est d'améliorer l'autonomie d'un robot mobile face à son environnement pour qu'il puisse, sans intervention humaine, accomplir les missions qui lui ont été confiées. Le spectre de ces missions est immense, il couvre des domaines aussi variés que l'industrie manufacturière, le spatial, l'automobile et plus récemment les loisirs et le secteur médical, ce qui démontre l'intérêt croissant porté aujourd'hui au sein de la communauté de la robotique mobile au développement d'un système intelligent autonome capable à la fois de percevoir correctement son environnement et également de savoir comment réagir en

conséquence suivant le niveau d'autonomie. C'est à lui de planifier son parcours et de déterminer avec quels mouvements va-t-il atteindre son objectif ?

La perception d'une part et la commande d'autre part sont donc les deux thèmes majeurs de recherche pour obtenir un robot mobile parfaitement autonome. Parmi les problématiques liées à la commande, celle de la navigation qui tient un rôle important, elle consiste à déterminer les trajectoires à suivre par le robot pour évoluer correctement dans des milieux encombrés d'obstacles.

La navigation autonome est un problème fondamental dans la robotique mobile, elle consiste à doter le robot réel de capacités décisionnelles afin qu'il puisse exécuter d'une manière autonome (sans aucune intervention humaine) des déplacements à travers son environnement de travail en utilisant seulement l'information perçue par ses capteurs. La complexité de la méthode de navigation mise en œuvre sur un robot mobile dépend de l'environnement dans lequel il doit évoluer (milieu intérieur ou environnement naturel, sol plan ou irrégulier, ...). Elle dépend également de la connaissance de cet environnement qui peut être figé ou évolutif et du mode de définition de la trajectoire (apprentissage préalable, planification en ligne, ...).

La détermination de la trajectoire pour un robot mobile autonome a été largement abordée au cours de ces dernières années. On distinguait habituellement deux grandes catégories d'approches : les approches délibératives et les approches réactives. Le principe des approches délibératives est de déterminer un mouvement complet (un chemin ou une trajectoire) du robot entre une position initiale et une position finale à partir du modèle de l'environnement dans lequel évolue le système. Toutefois, la complexité inhérente à ces approches peut empêcher le calcul du mouvement en temps réel. Les approches réactives quant à elles calculent uniquement le mouvement à appliquer au prochain pas de temps à partir de données capteurs récupérées par le système robotique à chaque instant. Une représentation de l'environnement est ainsi construite au fur et à mesure du déplacement; la navigation est donc possible en environnement incertain comme en environnement dynamique. La convergence vers le but du robot est néanmoins difficile à garantir. En effet ces approches sont sujettes à

des minima locaux dont il est difficile de sortir sans connaître le chemin global permettant de rejoindre le but.

Au regard des progrès réalisés dans le domaine de la navigation durant ces dernières années, la nécessité de disposer à la fois des capacités des approches réactives et délibératives s'est confirmée. De nombreux travaux actuels combinent désormais une planification globale et un évitement d'obstacle local afin de palier les défauts de chacune de ces deux catégories d'approches.

Afin de pouvoir introduire l'approche de navigation par raisonnement flou constituant le sujet de cette thèse, les efforts opérés ces dernières années pour combler le manque de convergence des approches réactives et l'efficacité des approches délibératives ont été soulevés. Cet état de l'art est constitué des travaux ayant pu inspirer les nôtres ou ayant servi d'outils nécessaires à la mise en place de notre approche.

## **I.2 Etat de l'art**

L'objectif principal de la robotique mobile actuelle consiste à développer des systèmes intelligents complètement autonomes. La navigation autonome constitue un thème de recherche majeur pour atteindre cet objectif, elle vise à doter le robot mobile de capacités intelligentes pour qu'il soit en mesure de prendre des décisions et de s'adapter aux variations des conditions de fonctionnement sans intervention humaine lors de l'exécution des déplacements à travers son environnement de travail en utilisant seulement l'information perçue par ses capteurs.

Plusieurs approches ont été proposées pour contribuer à la résolution du problème de la navigation autonome d'un robot mobile, les plus utilisées sont basées sur les champs de potentiels artificiels, les réseaux de neurones artificiels, les algorithmes génétiques, la logique floue et les systèmes hybrides. L'idée fondamentale de la méthode de champ de potentiel, proposée par Khatib [1], consiste à attribuer à l'environnement du robot un champ de potentiel virtuel où le robot est attiré par la

cible et poussé par les obstacles. Bien que cette méthode soit rapide et efficace, elle a les limitations et les inconvénients suivants [2] :

- Situations de piège dues aux minima locaux.
- Oscillations dans les passages étroits.

Afin de résoudre ces inconvénients, des recherches intensives ont été élaborées. Certains chercheurs ont préféré introduire des modifications au niveau des fonctions des champs de potentiel de manière à améliorer la qualité de la navigation du robot en minimisant les oscillations [3-13]. D'autres ont choisi de combiner la méthode du champ de potentiel avec un contrôleur flou dans le but de construire un navigateur fiable et robuste [14-17]. Malheureusement, ces améliorations restent incapables de protéger le robot contre le problème des minima locaux. Sun Ling et ses collègues [18] ont proposé une méthode des cibles virtuelles pour la planification en temps réel de la trajectoire d'un robot mobile, le contrôle flou a été utilisé pour générer les cibles virtuelles appropriées tandis que la méthode de champ de forces virtuelles a été utilisée pour calculer les forces attractives produites par les cibles virtuelles. Pour surmonter le problème des minima locaux dont cette méthode souffre, les auteurs ont proposé un algorithme simple qui consiste à ajouter une distance et un angle aléatoires aux positions des cibles virtuelles. Cet algorithme fonctionne efficacement dans le cas où le robot est emprisonné dans un point d'équilibre local, mais il échoue en présence des obstacles concaves où le robot entre dans une boucle infinie. Dans cette situation, ajouter une distance et un angle aléatoires aux cibles virtuelles n'est pas efficace ; il pourrait aider le robot pour s'en sortir mais après avoir traversé plusieurs boucles.

Les architectures à base de réseaux de neurones ont été introduites avec succès depuis plusieurs années aux applications robotiques notamment le suivi, la prédiction et la planification des trajectoires des objets mobiles [19-29].

Les algorithmes génétiques sont des outils d'optimisation très puissants, ils sont fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de la survie des structures les mieux adaptées, et les échanges d'information pseudo-aléatoires, pour former un algorithme d'exploration qui possède

certaines caractéristiques de l'exploration humaine [30,31]. Les algorithmes génétiques ont été introduits avec succès aux applications robotiques notamment l'optimisation et la planification de trajectoires hors ligne [32-36].

Le contrôle flou a montré son efficacité pour les applications robotiques [37-43], mais la construction d'un contrôleur flou performant n'est pas toujours facile. L'inconvénient majeur est le manque d'une méthodologie systématique pour la conception, due au nombre important de paramètres à régler (les paramètres des fonctions d'appartenances, les paramètres de la partie conclusion et les règles d'inférences). On trouve dans la littérature plusieurs méthodes de réglage des contrôleurs flous par l'intégration des propriétés des systèmes flous avec d'autres algorithmes de l'intelligence artificielle, telles que les réseaux de neurones, l'apprentissage supervisé, les algorithmes génétiques ou colonie de fourmis [44-89]. Ces méthodes dites hybrides combinent les propriétés de chaque approche afin d'optimiser les paramètres des systèmes d'inférence flous, elles sont capables de générer une solution optimale. L'apprentissage supervisé nécessite la disponibilité d'un nombre convenable de paires situation-action. Malgré sa convergence rapide, il est difficile d'obtenir un nombre suffisant de données pour effectuer l'apprentissage [90-92].

Les systèmes de navigation basés sur les comportements consistent à décomposer la tâche de navigation globale en un ensemble de comportements élémentaires d'action simple à concevoir et à gérer [93-96]. Ces comportements sont arrangés dans des niveaux de priorité où le comportement de plus haut niveau supprime tout les comportements de niveau inférieur. Le problème majeur rencontré lors de la conception d'un tel système est comment effectuer une coordination efficace entre les différents comportements [97] ?

### **I.3 Structure de la thèse**

L'objectif de ce travail est d'étudier et d'appliquer des techniques floues pour la navigation autonome d'un robot mobile dans un environnement inconnu, afin de permettre au robot de se mouvoir d'une position initiale à une autre finale en évitant les obstacles. On utilise d'une part l'approche comportementale à base de la logique floue, et d'autre part le paradigme de l'optimisation. Les techniques employées pour aborder ce problème sont basées sur les systèmes d'inférence flous et les algorithmes génétiques. Cette thèse est organisée de la manière suivante:

Le premier chapitre expose l'introduction générale, l'état de l'art et la structure de la thèse.

Dans le deuxième chapitre, un aperçu général sur le domaine de la robotique mobile est abordé pour examiner les types des robots mobiles, leurs principales structures cinématiques et les différents types de terrains de navigation.

Un état de l'art des approches de navigation existantes est présenté dans le troisième chapitre. On a commencé par énoncer les fondements théoriques de chaque type d'approches à savoir les approches délibératives et les approches réactives. Les avantages et les inconvénients de chaque approche sont exposés par la suite.

L'architecture de contrôle proposée pour la navigation autonome d'un robot mobile à base de la logique floue est détaillée dans le quatrième chapitre. Elle est établie sur des systèmes de commande réactifs basés sur les comportements, en décomposant la tâche globale de navigation en un ensemble de sous tâches secondaires: navigation vers la cible, évitement d'obstacles et suivi de murs.

Dans le dernier chapitre, une nouvelle approche basée sur la combinaison du concept des obstacles virtuels avec le contrôle flou est présentée pour apporter une solution optimale au problème des minima locaux. La méthode proposée est établie sur un algorithme des obstacles virtuels pour la construction du comportement de prévention des situations d'urgence et le raisonnement flou pour la coordination efficace des priorités entre les quatre différents comportements réactifs : navigation

vers la cible, évitement d'obstacles, suivi de murs et prévention des situations d'urgence.

Le long des deux derniers chapitres, des exemples de simulations sont fournis afin de montrer les performances des méthodes proposées pour la navigation autonome d'un robot mobile.

Nous terminerons cette thèse par une conclusion générale récapitulant ce qui a été fait et expose les perspectives de ce travail.

## **Chapitre II:**

### **LA ROBOTIQUE MOBILE**

#### **II.1 Introduction**

L'objectif de ce chapitre est de donner un aperçu général sur la robotique mobile. Après la présentation des différents types des robots mobiles, et des contraintes de terrain sur lequel ils sont conçus pour évoluer, nous aborderons la classification des plateformes mobiles et leurs principales structures cinématiques. Enfin nous étudions les outils permettant aux robots de percevoir leur environnement, étape primordiale nécessaire à l'autonomie totale des robots mobiles.

A la fin de ce chapitre, les thématiques de recherche actuelles associées au domaine de la robotique mobile sont présentées pour aboutir au développement d'un robot totalement autonome. Parmi celles-ci, nous nous focalisons alors sur celle de la navigation d'un robot mobile.

#### **II.2 Définition d'un robot mobile**

En fait, il existe deux principaux types de robots mobiles à savoir [98-102] : les robots manipulateurs et les robots mobiles. Les robots manipulateurs ont une base fixe contrairement aux robots mobiles qui peuvent se déplacer. Les robots mobiles à roues sont en effet les systèmes les plus étudiés, parce qu'ils sont plus simples à réaliser que les autres types de robots mobiles, ce qui permet d'en venir plus rapidement à l'étude de leur navigation [103-104]. Ce type de robots est notamment très souvent utilisé pour l'étude des systèmes autonomes. Vient ensuite la robotique mobile à pattes, avec notamment la robotique humanoïde, mais également des robots avec un nombre de pattes plus élevés qui offrent de bonnes propriétés pour la locomotion en milieu difficile (milieux forestiers et agricoles). La stabilité des mouvements de ce type de robots est en particulier un thème de recherche important [105]. Enfin il existe

également de nombreux autres types de robots mobiles (robots marins, sous marins [54], drones volants, micro et nano robots), généralement l'étude de ce type de robots se fait dans des thématiques spécifiques avec des problèmes particuliers à l'application visée.

### **II.3 Autonomie d'un robot mobile**

Il existe 2 principaux modes de fonctionnement pour un robot mobile [108]: télé-opéré et autonome. En mode télé-opéré, une personne pilote le robot à distance. Elle donne ses ordres via une interface de commande (joystick, clavier/souris...), et ceux-ci sont envoyés au robot via un lien de communication (internet, satellite ...). D'ailleurs, suivant le niveau de télé-opération, le terme "robotique" est plus ou moins justifié. Le robot doit donc obéir aux ordres de l'opérateur qui perçoit l'environnement autour du robot, par différents moyens (retour d'image, retour d'effort, ...), de manière à donner des ordres adaptés au robot. Dans ce domaine, les efforts de recherche sont beaucoup portés sur les problèmes liés au réseau de télécommunication (retards dans le réseau de communication, problèmes de commande, pertes de données) et sur l'amélioration de la perception de l'environnement par l'opérateur (interfaces images, retours d'efforts). A l'inverse, en mode autonome le robot doit prendre ses propres décisions. Cela signifie qu'il doit être capable à la fois de percevoir correctement son environnement, mais également de savoir comment réagir en conséquence, suivant le niveau d'autonomie. C'est à lui de planifier son parcours et de déterminer avec quels mouvements il va atteindre son objectif. Les recherches dans ce domaine portent principalement d'une part sur la localisation du véhicule autonome et la cartographie de son environnement, d'autre part sur le contrôle de tels véhicules (structure de contrôle, stratégies de commande, planification).

Cette notion d'autonomie prise en exemple ci-dessus, que nous pourrions qualifier de décisionnelle, ne doit pas être confondue avec celle d'autonomie énergétique (capacité du robot à gérer efficacement son énergie, à la préserver, voire à se ravitailler), même si ces deux notions sont étroitement liées : idéalement une des

préoccupations principales d'un robot mobile totalement autonome (du point de vue décisionnel), serait en effet de pouvoir gérer de lui-même ses réserves d'énergie.

Voyons maintenant les différents types d'environnement dans lesquels les robots mobiles sont amenés à se mouvoir.

#### II.4 Les différents types de terrains

Nous rencontrons principalement 3 types d'espaces de navigation : les terrains plats, les terrains accidentés et les espaces 3D. Les terrains plats sont généralement utilisés pour modéliser les milieux urbains et les intérieurs de bâtiments. Le robot évolue sur un plan 2D considéré sans pentes, et tout objet qui sort de cet espace 2D est considéré comme un obstacle (Figure II.1).

Cette représentation est la plus simple à étudier et la plus répandue pour les robots mobiles à roues. En première approche, elle permet de se concentrer sur les problèmes de contrôle et de navigation autonome du robot.



**Figure II.1** Le robot taxi

Les terrains accidentés correspondent généralement aux milieux en extérieur, comme des forêts, des champs en robotique agricole, ou encore des terrains rocheux (Figure II.2). La différence avec les terrains plats est la présence de pentes, de bosses et de creux sur le terrain d'évolution du robot. Cela interdit d'utiliser une métrique standard 2D et cela complique pour beaucoup la détection d'obstacles et la

modélisation des déplacements du robot. De plus il devient également important de vérifier que le robot ne bascule pas quand il escalade une pente ou enjambe un obstacle. Le système de locomotion du robot doit dans ce cas être adapté à la topologie du terrain.



**Figure II.2** Le robot martien Rover

Enfin les espaces d'évolution 3D sont par exemple utilisés pour modéliser la navigation des robots sous-marins (Figure II.3-a) et des drones volants (Figure II.3-b). Les problèmes rencontrés sont spécifiques à l'application visée.



**(a)**



**(b)**

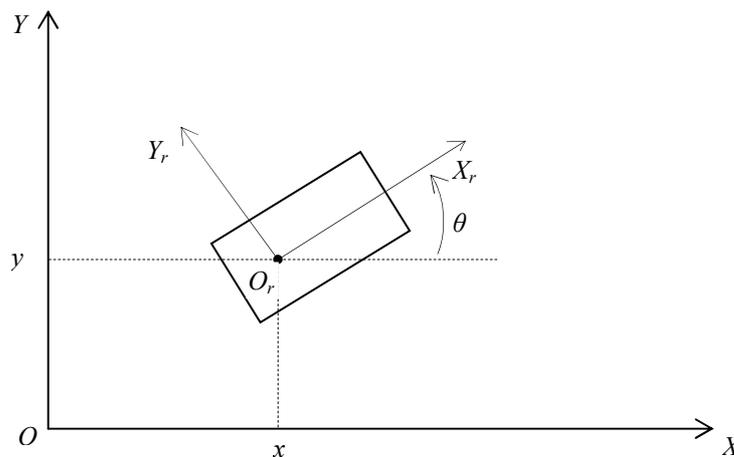
**Figure II.3** (a) Robot sous marin. (b) Drone de surveillance

Chaque type de terrain correspond à des problématiques bien spécifiques. Le type de robot étudié dans cette thèse est destiné à circuler en environnement urbain, la modélisation terrain plat sera utilisée. Cela signifie que l'on considère que tous les mouvements sont contenus dans un plan de navigation, parallèle au sol.

Pour la modélisation terrain plat, nous définissons un repère absolu (fixé dans l'environnement)  $R = (O, X, Y)$ , donc l'axe  $z$  est perpendiculaire au sol. Nous définissons un repère mobile lié au robot  $R^r = (O_r, X_r, Y_r)$  dit égocentrique. Le point  $O_r$  est le point de contrôle du robot. Généralement, sur un robot type voiture, le point de contrôle est fixé au centre de l'essieu non directeur. Ce repère égocentrique se déplace avec le robot. Pour réaliser une navigation, l'état du robot est totalement défini par le vecteur :

$$X = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (\text{II.1})$$

Dans lequel  $(x, y)$  désignent la position du robot dans le plan  $(o, \vec{x}, \vec{y})$  et  $\theta$  son orientation (Figure II.4).



**Figure II.4** Modélisation du robot dans le repère absolu

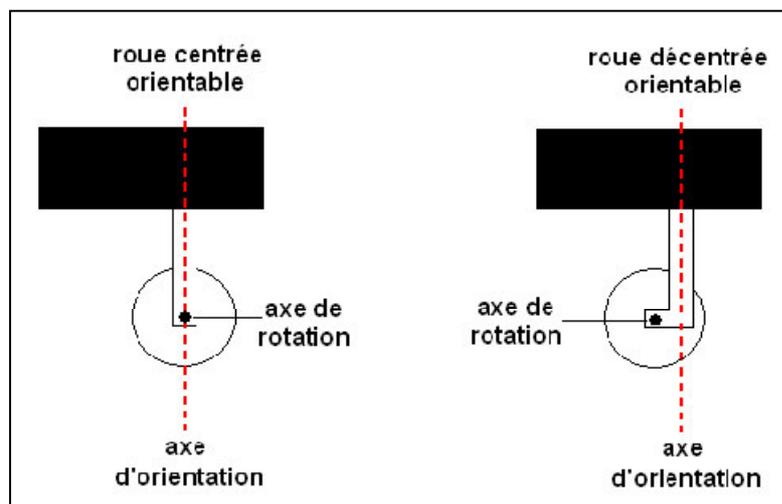
La section suivante présente les différents aspects de la modélisation sur sol plat : la classification des différents types de roues, les configurations non holonomes et holonomes, et la gestion des glissements dans le modèle.

## II.5 Modélisation des robots mobiles

### II.5.1 Classification des types des roues

La mobilité d'un robot mobile dépend grandement du type de roues utilisées. Une classification des différents types de roues rencontrées en robotique mobile est illustrée ci-dessous:

- la roue fixe : cette roue n'autorise qu'un déplacement dans la direction de son plan médian, l'orientation n'est pas modifiable,
- la roue centrée orientable : elle possède un axe d'orientation en plus de l'axe de rotation, et cet axe d'orientation passe par le centre de la roue,
- la roue décentrée orientable ou roue folle : son axe d'orientation ne passe pas par le centre de la roue (c'est le cas par exemple des roues des chaises de bureau)



**Figure II.5** Roue centrée et roue décentrée

En plus de ces roues classiques (Figure II.5), d'autres roues ont été développées pour accroître la mobilité du robot. Elles permettent d'augmenter les capacités de

déplacement dans toutes les directions du plan (Figure II.6). Cependant, elles ne sont commandables que dans certaines de ces directions. Dans cette catégorie nous trouvons notamment :

- les roues suédoises : ces roues autorisent les glissements latéraux grâce à un système de galets remplaçant la bande de roulement classique, montés en inclinaison par rapport au plan de la roue. La combinaison de la rotation de la roue avec la rotation libre du galet en contact avec le sol permet le déplacement sans glissement sur le sol dans toutes les directions. Cependant le couple moteur que l'on peut transmettre à ces roues est très limité, ce qui réduit son utilisation en pratique.
- les roues tronco-sphériques (ou orthogonal wheels) : cette structure utilisant deux roues libres en quadrature présente l'avantage de pouvoir transmettre un couple intéressant par rapport aux roues suédoises, mais souffre de petits problèmes de sauts au moment de la transition d'une roue support à l'autre.

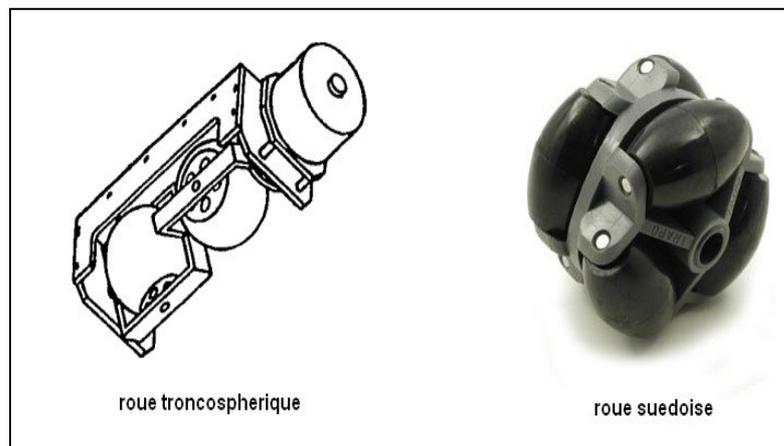


Figure II.6 Roue troncosphérique et roue suédoise

### II.5.2 Types des plates formes mobiles

En associant les différents types de roues selon une structure mécanique donnée, le robot mobile disposera de plus ou moins de mobilité. Le nombre, le type et la disposition des roues engendrera ou non la contrainte de non holonomie du robot. Si on néglige les phénomènes dynamiques tels que l'inertie, un robot holonome est un

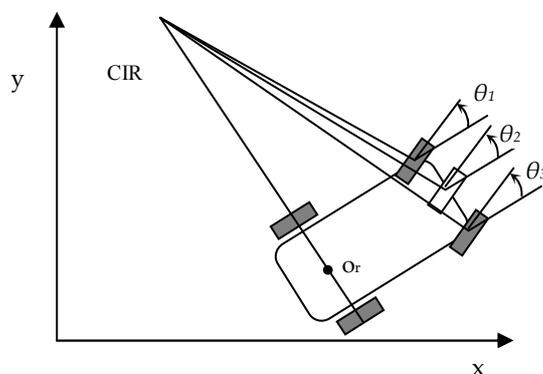
robot capable à chaque instant de se déplacer dans n'importe quelle direction du plan, sans avoir à effectuer une reconfiguration de ses roues.

Tout système évoluant dans un plan 2D possède 3 degrés de liberté : une translation selon l'axe  $x$ , une translation selon l'axe  $y$  et une rotation autour d'un axe  $z$  normal à  $(\vec{x}, \vec{y})$ . Cependant une roue classique ne possède que 2 degrés de mobilité : elle ne peut que faire une translation (avancer ou reculer), ou une rotation sur elle-même. Elle ne peut pas déraiper transversalement pour effectuer un mouvement de translation horizontal.

Cette contrainte empêche la plupart des véhicules traditionnels d'effectuer un déplacement instantané transversal (parallèlement à l'axe de rotation de la roue). Une voiture ne peut pas effectuer de créneau pour se garer, sans faire un certain nombre de manœuvres. C'est une contrainte que l'on retrouve sur tous les robots mobiles de type voiture ou à roues différentielles. De tels véhicules, possédant un nombre de degrés de mobilité inférieur au nombre de degrés de liberté, sont dits non-holonomes, et cette contrainte touche principalement les robots mobiles à roues.

### II.5.2.1 Plates formes non holonomes

Les systèmes mobiles dit non-holonomes sont ceux que l'on rencontre le plus dans la vie courante (voiture particulière, bus, camion, ...etc.). Ces systèmes ont une structure mécanique relativement simple (des roues motrices, des roues directrices et des roues libres).



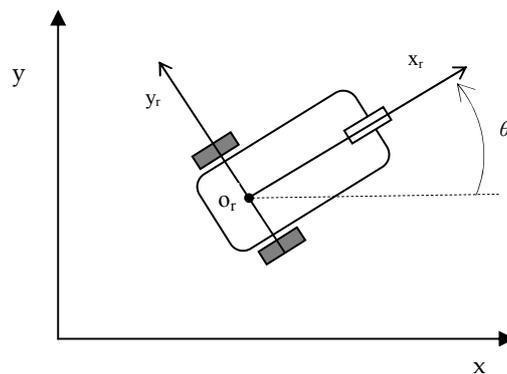
**Figure II.7** Equivalence modèle voiture et tricycle

Les configurations non holonomes les plus courantes sont :

**a) Plates formes de type tricycle et voiture :** ces deux structures sont constituées d'un axe fixe (généralement à l'arrière) et d'un axe directeur (Figure II.7). Dans le cas du tricycle, seule une roue est présente sur l'axe directeur, contrairement à la voiture qui en possède deux.

La théorie d'Ackerman-Jeantaud donne les conditions théoriques de non glissement et non dérapage pour les configurations de type voiture. Notamment les axes de rotation des 4 roues doivent s'intersecter en un point unique, le Centre Instantané de Rotation. Pour cela, la vitesse de la roue extérieure doit être légèrement supérieure à celle de la roue intérieure. La structure de type voiture peut être modélisée par une structure équivalente à 3 roues, ce qui revient au modèle du tricycle.

**b) Plates formes à roues différentielles :** cette structure est constituée de deux roues motrices placées sur le même axe (Figure II.8), et d'au moins un appui supplémentaire (généralement une ou deux roues folles). L'avantage de cette structure est qu'elle permet au véhicule de tourner sur place, suivant si les vitesses de rotation des deux roues motrices sont de signe opposées ou pas. Ainsi le robot peut pivoter rapidement, ce qui donne des capacités de déplacement intéressantes. Cependant, le déplacement latéral n'étant pas directement réalisable, cette structure n'est pas non plus holonome.



**Figure II.8** Modèle de type tricycle à roues différentielles

### II.5.2 Plates formes holonomes

En utilisant des roues telles que les roues suédoises ou tronco-sphériques sur des plateformes mobiles, des robots ayant la capacité de se mouvoir dans toutes les directions ont été créés. Ces robots, à 3 degrés de mobilité dits omnidirectionnels, permettent de s'affranchir de la contrainte de non holonomie. Leurs structures spécifiques leur permettent de se déplacer instantanément (à la dynamique près) dans toutes les directions en ayant n'importe quelle orientation, rendant possible le suivi de trajectoires de forme quelconque (Figure II.9).



Figure II.9 Le robot omnidirectionnel Nomadic XR4000

### II.5.3 Roulement avec ou sans glissement

La locomotion se fait grâce au frottement entre la roue du véhicule et le sol, et l'efficacité du mouvement dépend notamment du type de sol. Pour que l'hypothèse du roulement sans glissement soit validée, il faudrait théoriquement que le contact sol/roue ne se fasse qu'en un point, que le sol soit parfaitement plat, et que le rayon de la roue soit parfaitement constant sur toute sa périphérie (Figure II.10).

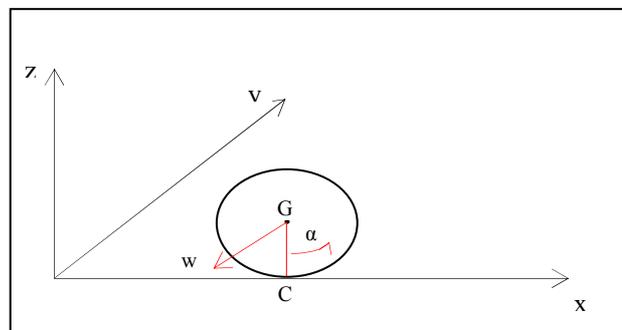


Figure II.10 Roulement sur le sol

En réalité, le contact sur le sol se fait sur une surface avec le pneu de la roue. Les glissements sur le sol sont une source d'erreur importante pour certaines méthodes de localisation. Cela est notamment le cas pour l'odométrie classique, qui s'appuie sur cette hypothèse pour déterminer la position relative d'un robot par rapport à son point de départ, à partir de la mesure du nombre de tours parcourus par chaque roue. On sait que dans ce cas l'intégration des dérives dues aux glissements, entraîne sur un parcours complet d'importantes erreurs de localisation.

Dans certaines applications robotiques, la précision du déplacement est un facteur important, il est alors nécessaire de prendre en compte les glissements dans la modélisation du robot. Il existe deux principales approches pour les intégrer : soit on passe à un modèle dynamique du robot, soit on reste sur un modèle cinématique classique (roulement sans glissement) dans lequel on introduit les effets du glissement sous forme de paramètres supplémentaires. La modélisation dynamique des phénomènes de glissements est plus complète mais elle requiert d'une part la mesure ou l'estimation, en temps réel, d'un grand nombre de paramètres tels que les coefficients de frictions, et d'autre part de paramètres supplémentaires inhérents à l'état des pneumatiques. Ces nombreux paramètres ne sont pas triviaux à obtenir en ligne, ce qui limite l'utilisation de tels modèles en pratique. Les modèles cinématiques, modifiés pour intégrer les glissements, s'avèrent plus simple à mettre en œuvre de part le nombre réduit de paramètres à estimer (on ne modélise plus les phénomènes complets de glissement, mais juste leur effet sur la cinématique du véhicule), et ils permettent tout de même de prendre en compte avec une très bonne précision les phénomènes de glissement. Par exemple, dans [106], Luka et ses collègues développent un modèle cinématique étendu qui leur permet d'améliorer la précision du suivi de trajectoire par un véhicule sur un terrain particulièrement glissant.

## **II.6 Détection des obstacles et localisation**

La perception de son environnement d'évolution est la base de tout système autonome. Sans une bonne perception et interprétation de ce qui l'entoure, un robot ne peut pas prendre de décision correcte. Cette partie vise à décrire les différents moyens

mis à disposition au robot pour localiser les obstacles qui l'entourent. Ensuite les différentes méthodes de localisation du robot lui-même sont abordées. L'idée est de permettre au final de créer un modèle, plus ou moins simplifié, des interactions entre le robot et son environnement. Cette étape est nécessaire et primordiale pour la navigation d'un robot mobile autonome.

Pour cela, un robot est équipé de capteurs proprioceptifs qui fournissent des informations sur le robot lui-même, et extéroceptifs qui fournissent des informations sur ce qu'il y a autour de lui (son environnement).

### **II.6.1 Détection des obstacles et cartographie**

Les capteurs permettant de fournir des informations sur l'environnement extérieur peuvent être classés en deux catégories, passifs et actifs [107]. Dans le premier cas, on se contente de recueillir et d'analyser une énergie fournie par l'environnement, typiquement la lumière. Dans le second cas, c'est au capteur de générer une énergie, et de récupérer cette énergie après interaction sur le milieu extérieur. C'est le principe de base des télémètres (capteurs de mesures de distances), qui sont largement utilisés pour tracer des cartes en ligne de l'environnement dans lequel évolue le robot. Les télémètres laser à balayage sont fréquemment utilisés pour la navigation de robots avec de très bonnes performances notamment en intérieur.

Le principe de ces télémètres repose sur le calcul du temps aller-retour mis par une impulsion lumineuse pour revenir sur le capteur. Une onde infrarouge de faible puissance est émise par la diode laser, et au même moment un chronomètre informatique est lancé. L'onde se réfléchit sur le premier objet rencontré en chemin, et revient sur le détecteur du capteur. Le temps mis par l'onde pour faire l'aller-retour permet de déterminer la distance de l'objet. Un miroir tournant motorisé permet de balayer toute une gamme d'angles devant le télémètre, dans le plan de balayage qui est parallèle au sol. La précision de ces appareils et leur robustesse aux variations de température en font des outils très intéressants pour les applications en robotique mobile de faible et moyenne vitesse.

Les capteurs ultrasonores utilisent des ondes sonores de fréquence non perceptible par l'oreille humaine, généralement dans la fourchette 20-200 khz. De la même manière que les télémètres laser, ils sont basés sur le principe de la mesure du temps aller-retour lors de la réflexion sur un obstacle. C'est la méthode employée par certains animaux pour percevoir leur environnement, comme les chauves-souris ou les dauphins : l'écholocalisation. Un avantage de ces capteurs est que contrairement aux télémètres, l'onde qu'ils émettent n'étant pas focalisée, ils perçoivent beaucoup plus facilement des éléments filiformes comme des pieds de chaises ou des grillages. Par contre leur portée est faible, et ils sont moins adaptés aux milieux de propagation non isotropes comme l'air.

Un des inconvénients des capteurs ultrasonores par rapport aux télémètres lasers est la divergence importante du faisceau ultrasonore, qui s'apparente plus à un cône qu'à un faisceau.

Généralement l'ouverture de l'angle est de plusieurs dizaines de degrés, ce qui rend la localisation des obstacles imprécise. Ces capteurs sont donc plutôt utilisés pour des mesures à courte distance (de quelques centimètres à quelques mètres). Ils sont relativement sensibles aux variations de température, et la fréquence de mesure dépend de la distance maximale de détection (plus cette distance est grande, moins la fréquence d'acquisition des mesures est élevée).

L'avantage de ces capteurs est qu'ils sont moins onéreux qu'un télémètre laser, et ils sont souvent utilisés dans des applications en intérieur avec des espaces de navigation assez restreints.

Les capteurs passifs se servent directement de l'énergie émise par l'environnement. C'est typiquement le cas des systèmes de vision par caméra en stéréo vision. La reconnaissance de primitives entre deux images permet d'évaluer la position/orientation d'un objet, et ainsi d'évaluer la profondeur. L'utilisation simultanée de deux caméras est cependant nécessaire pour y parvenir. Plus de deux caméras peuvent également être utilisées, de manière à améliorer la robustesse de la méthode. La vision omnidirectionnelle s'avère également très intéressante dans le cadre d'applications en robotique mobile, dans le sens où elle permet de surveiller en

même temps tout ce qui se passe autour du robot. La caméra est placée face à un miroir parabolique ou hyperbolique. L'image est complètement distordue ce qui complique la mesure de distances, mais l'aspect vision panoramique offre des avantages pour l'évitement d'obstacles dynamiques en environnement structuré. Les verticales deviennent des radiales, et les horizontales des arcs de cercle. L'utilisation de systèmes à base de vision est fortement développée, et pas seulement dans le domaine de la robotique. Mais globalement, ce type de système reste fortement tributaire de la qualité de l'énergie recueillie : influence de la luminosité ou encore du contraste.

### **II.6.2 Fusion de données multi-capteurs et cartographie**

La localisation d'un robot mobile s'effectue par la mise en correspondance de différentes sources extéroceptives et proprioceptives. Généralement il s'agira de confronter les mesures de déplacements prises par odométrie avec une méthode de localisation absolue : soit reconnaissance et calcul de distance par rapport à des balises de position connue, soit mise en correspondance avec une carte construite en ligne et/ou présente dans une base de données, soit encore localisation externe du robot par des capteurs dans l'environnement (GPS). Le principe le plus simple pour effectuer cette mise en correspondance, consiste à utiliser les mesures de localisation absolue pour recalibrer périodiquement l'état du robot, obtenu par intégration des déplacements mesurés par l'odomètre. Cette méthode, bien que simple à utiliser, présente le problème de ne pas utiliser conjointement les différents moyens de mesure, mais successivement. Ainsi on ne tient pas compte des incertitudes liées tant à l'odométrie qu'à la méthode de localisation absolue.

Cependant nous savons que quelque soit la technologie utilisée pour la prise d'informations, aucune mesure n'est parfaite, il existe toujours une part d'incertitude sur celle-ci. Ces incertitudes peuvent provenir soit du principe de mesure lui même, soit des imperfections technologiques. Typiquement pour une mesure de distance avec un télémètre laser, on trouve des erreurs systématiques d'une quinzaine de millimètres en moyenne (erreur constante intrinsèque au télémètre utilisé), et une erreur statistique

de 5 mm environ. Lorsque plusieurs méthodes de mesure sont utilisées conjointement, le principe utilisé pour mettre en concordance les informations consiste à effectuer une moyenne pondérée des différentes mesures par la confiance que l'on accorde à chacune (inversement à leur variance donc). Soit deux mesures  $z_1$  et  $z_2$  d'une même variable  $x$ , obtenues par des capteurs différents, avec des variances associées  $\sigma_1$  et  $\sigma_2$ , alors la loi de Bayes nous donne la valeur estimée de  $x$ :

$$\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \cdot z_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \cdot z_2 \quad (\text{II.2})$$

et la variance associée à l'estimée :

$$\sigma = \frac{1}{\frac{1}{\sigma_1} + \frac{1}{\sigma_2}} \quad (\text{II.3})$$

La variance associée à cette estimation est plus faible que les variances de chacune des mesures prises séparément (ce qui est logique, cela traduit simplement le fait que plus on recoupe d'informations provenant de sources différentes, plus on diminue les incertitudes). Cette méthode est intéressante si l'on dispose de ressources limitées pour le calcul, mais l'estimation peut être largement améliorée en utilisant les informations sur les mesures passées, et en filtrant avec un filtre de Kalman.

Pour des obstacles statiques, nous obtenons des mesures récurrentes selon une certaine fréquence d'acquisition. Pour des mesures récursives, la précision peut être améliorée en utilisant un filtre de Kalman. Ce type de filtrage, très utilisé notamment en automatique, est un filtre statistique qui permet de réduire les incertitudes au fur et à mesure de l'acquisition de nouvelles mesures. Cette méthode est particulièrement utilisée en robotique pour la localisation du robot relativement aux obstacles [108]. L'algorithme utilise les connaissances sur la dynamique du robot et du système de mesure et sur les incertitudes associées à chaque mesure. Le calcul s'effectue en deux phases : une phase de prédiction de la mesure et de sa variance, suivie d'une phase de

mise à jour de celle-ci par l'acquisition de nouvelles mesures. L'historique des mesures n'a pas besoin d'être gardé en mémoire, et le processus est récursif.

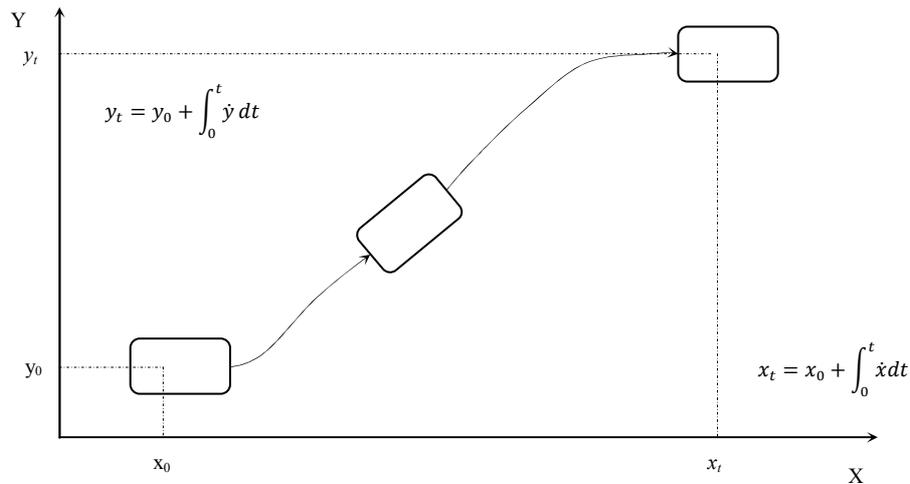
Pour pouvoir planifier les déplacements du robot, il est nécessaire d'établir une modélisation de l'environnement à partir des mesures des positions relatives des obstacles par rapport au robot. Concrètement il s'agit d'établir une cartographie locale des espaces où le robot pourra circuler, ou non, en considérant la position absolue du robot comme connue. Il existe 2 grands types de représentation pour l'environnement local : les cartes géométriques et les grilles d'occupation. Les premières peuvent être obtenues par traitement des données issues de mesures télémétriques, et en effectuant une reconnaissance des formes simples (murs, coins). A partir d'une connaissance des déplacements du robot et en comparant la carte courante avec des cartes mises en mémoire au fur et à mesure que le robot se déplace, la robustesse de la carte peut être améliorée. Les grilles d'occupation sont des cartes discrétisées, généralement sous forme de grille d'un certain nombre de lignes et de colonnes. A chaque case de la grille est soit associée une valeur booléenne pour dire si la case est accessible par le robot ou non (occupée par un obstacle ou libre), soit une probabilité d'occupation (loi de Bayes).

### **II.6.3 Localisation d'un robot mobile**

Les outils permettant la localisation d'un robot dans son environnement peuvent être classés en deux catégories : ceux par localisation à l'estime et ceux par localisation absolue [108].

Le principe de la première catégorie consiste à intégrer des informations sur les vitesses ou les accélérations fournies par des capteurs proprioceptifs (odomètres, centrales inertiels).

L'avantage de ces méthodes est qu'elles sont indépendantes de l'environnement, par contre leur souci est leur manque de précision dû à la dérive temporelle. En effet les erreurs s'intégrant elles aussi au fur et à mesure du temps, il est nécessaire d'apporter régulièrement des recalages (Figure II.11).



**Figure II.11** Localisation à l'estime

Parmi les méthodes de localisation à l'estime, le système le plus simple et le plus couramment utilisé pour la mesure de déplacement du robot est l'odométrie. L'hypothèse de roulement sans glissement que nous avons vu précédemment, nous permet de relier directement les déplacements du robot à la vitesse de rotation des roues. Par intégration des déplacements à chaque instant, on en déduit la position relative du robot par rapport à son point de départ.

L'odométrie est une méthode de localisation très courante, simple, mais également très rapidement imprécise. En effet à cause du glissement des roues sur le sol, les erreurs s'accumulent au fur et à mesure que le robot avance, ce qui implique d'importantes erreurs sur les longs parcours s'il n'y a pas de recalage régulier. Cette méthode est de ce fait fortement tributaire de la qualité du sol sur lequel le robot se déplace.

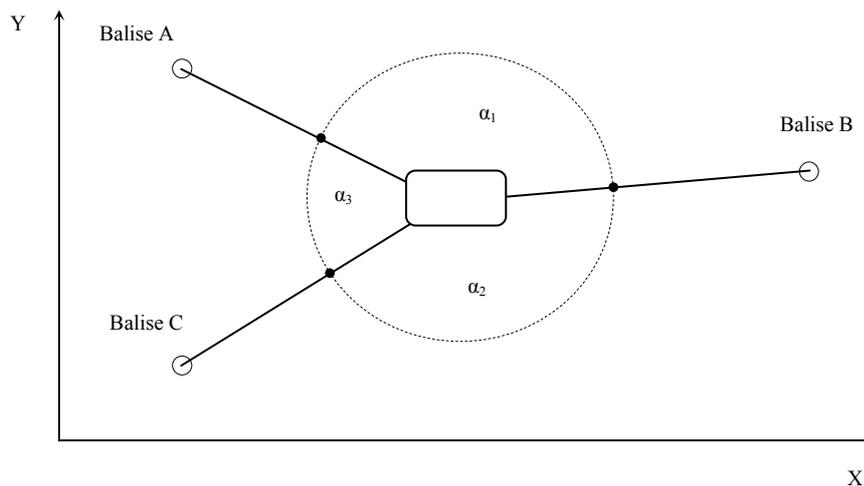
Les incertitudes sur le diamètre exact des roues, sur les paramètres géométriques du robot, sur la résolution des codeurs, génèrent des erreurs de type systématique, qui vont s'accumuler très rapidement en odométrie. Cependant, ces erreurs peuvent être identifiées et évaluées pour faire un recalibrage du système et ainsi améliorer sa précision. Les erreurs non systématiques comme les glissements ou les irrégularités du

sol, génèrent moins rapidement des erreurs, mais ne peuvent par contre pas être recalibrées puisqu'on ne peut pas les prévoir.

Pour l'exploration martienne, où le terrain est fortement accidenté, l'utilisation de système d'odométrie classique est impossible. Pour cette application, on a proposé une technique d'odométrie alternative, dite odométrie visuelle, basée sur la reconnaissance de points singuliers dans l'image vidéo du sol fournie par une caméra montée sur le robot (les points de Harris).

Connaissant le positionnement de la caméra par rapport au robot, le déplacement de ces points dans l'image permet d'évaluer les vitesses de déplacement du robot, et par intégration, de retrouver sa localisation relative par rapport à sa position initiale.

Le second type de méthode pour la localisation est la localisation absolue. Ces méthodes utilisent des éléments repérables par le robot dans l'environnement de navigation, de position connue, pour permettre au robot de se repérer relativement à ceux-ci. Ces éléments sont appelés des balises ou amers et sont dits soit réels, s'ils ont été placés spécialement pour permettre la localisation, soit virtuels s'il s'agit d'éléments présents naturellement (Figure II.12).



**Figure II.12** Localisation absolue (méthode par triangulation)

Les balises réelles sont dites passives si elles ont pour but de réfléchir un signal émis par un appareil de mesure du robot (laser ou infrarouge). Il existe deux méthodes pour utiliser ces balises pour la localisation du robot : la méthode télémétrique (calcul de la distance robot/balise), qui nécessite la présence de deux balises pour calculer la position du robot dans le plan ; et la méthode par triangulation, qui consiste à mesurer les angles entre chaque balise et le robot, et qui elle nécessite l'utilisation de 3 balises. Les balises réelles sont dites actives si elles émettent un signal captable par le robot. En milieu extérieur, le système GPS (Global Positioning System) peut être utilisé pour obtenir des positions d'une précision de l'ordre du mètre. A la base développée par l'armée américaine dans les années 80 (lancement du premier satellite GPS en 1978), il fut ouvert aux civils en 1995. Jusqu'en 2000 les mesures étaient volontairement entachées d'une erreur d'une centaine de mètres, l'armée américaine craignant que ce système soit un avantage pour leurs ennemis. Malgré le retrait de cette erreur volontaire, la précision du système restait de l'ordre du mètre, à cause des incertitudes sur l'orbite et l'horloge des satellites, ainsi que les retards engendrés par la traversée des couches atmosphériques.

Pour améliorer cette précision, nous pouvons utiliser les GPS différentiels : avec un second récepteur GPS sur une base fixe et de position connue, il devient possible de mesurer l'erreur et d'en déduire la correction à apporter pour la zone environnante. Pour que ce système fonctionne, il faut que la base mobile reste à une certaine portée de la base fixe. Cette distance varie suivant la gamme de fréquence utilisée pour l'envoi des corrections, et peut atteindre quelques dizaines kilomètres pour les besoins de la navigation maritime.

Des satellites géostationnaires permettent également de corriger certaines erreurs de position. Ils envoient des corrections sur les orbites et les horloges des satellites GPS. Ainsi pour l'Europe c'est le système EGNOS (European Geostationary Navigation Overlay System) qui se charge d'envoyer ces corrections. La précision atteinte est de l'ordre de 3m. Le système de GPS RTK (real time kinematics) permet d'améliorer la précision à quelques centimètres, en utilisant la différence de phase de l'onde porteuse du signal, sa longueur d'onde étant d'une vingtaine de centimètres. En

contrepartie la portée de la station de référence avec cette méthode n'est plus que de quelques kilomètres. Pour augmenter cette portée on a recours au RTK réseau, qui va utiliser la redondance d'informations pour communiquer les corrections à l'appareil, via un serveur de calcul à distance.

Les systèmes de localisation GPS sont très intéressants en rase campagne ou en banlieue. Cependant ils s'avèrent beaucoup moins efficaces en pleine ville ou en forêt. En effet pour fonctionner correctement un GPS a besoin de recevoir les informations de 4 satellites au moins, or la présence d'obstacles tels que les ponts et grands bâtiments en ville empêche parfois cette réception. Ainsi il n'est pas rare de perdre la localisation GPS pendant quelques minutes. C'est pour cela que beaucoup de recherches dans le domaine de la localisation sont aujourd'hui portées sur les méthodes alternatives au GPS en milieu urbain.

La vision peut être un moyen pour compléter la localisation par GPS. Ainsi on peut utiliser un modèle 3D de caméras embarquées sur le robot pour déterminer précisément la position du robot mobile.

#### **II.6.4 Localisation et cartographie simultanées**

En robotique mobile, le SLAM (simultaneous localization and mapping) consiste, pour un robot évoluant en milieu inconnu, à tracer une carte de l'environnement et localiser simultanément le robot dans celle-ci. La carte est construite de manière incrémentale au fur et à mesure que le robot évolue dans le terrain. En croisant les données perçues avec les informations géographiques dont il dispose en mémoire, le véhicule est capable de se localiser par rapport à des cartes préexistantes. L'idée du SLAM est donc de traiter conjointement les problèmes connexes que sont la navigation et la localisation d'un robot autonome.

La méthode alternative, moins gourmande en temps de calcul, consiste à utiliser la mémoire visuelle du véhicule, en repérant des éléments caractéristiques, pour se localiser par rapport à ceux-ci. Ces éléments sont extraits sous forme de primitives visuelles, que le robot cherchera à retrouver dans les images qu'il perçoit pour suivre un chemin précis [104].

## II.7 Les thématiques scientifiques de la robotique mobile

Il existe de nombreuses thématiques de recherche dans le milieu de la robotique mobile autonome, ce qui montre qu'aujourd'hui encore le problème spécifique des robots mobiles autonomes est entier. La communauté des chercheurs dans le domaine de la robotique a dégagé 4 grands axes de travail autour desquels s'articulent les recherches actuelles en robotique mobile [108]:

- Techniques de localisation et cartographie : cet axe regroupe tous les développements autour de la perception et de la localisation du robot. On y retrouve notamment les méthodes SLAM (Localisation et Cartographie Simultanées). Plus récemment l'utilisation de bases de données sous forme de cartes 2D ou 3D, mais également sous forme SIG (Système d'Informations Géographiques) a ouvert de nouvelles perspectives dans ce domaine. De manière générale la fusion de données est également un thème important, tant la nécessité de coupler diverses sources de mesures apparaît nécessaire pour améliorer la précision et garantir l'intégrité des informations,
- Contrôle et commande des véhicules : cet axe regroupe les thématiques liées à la planification de chemin, la génération de trajectoires, et la commande des robots de manière générale. Une prise en compte de plus en plus poussée des contraintes et de la dynamique des robots est nécessaire, pour adapter au mieux les robots à leur environnement. La bonne gestion des obstacles et la prise en compte des incertitudes de mesures sont également des points clés de cette thématique,
- La communication inter-véhicules : on retrouve ici tous les travaux liés à la coopération entre robots, et le contrôle de flottilles de véhicules,
- L'interprétation de scènes : les recherches dans ce domaine visent à pousser plus loin la perception de son environnement par le robot, que la simple reconnaissance des objets. En effet dans certaines applications il est nécessaire que le robot appréhende plus finement son environnement que par une simple détection et localisation des obstacles. Les travaux concernent notamment la perception multi-capteurs et la représentation dynamique des scènes.

## **II.8 Conclusion**

La perception d'une part et la commande d'autre part sont donc les deux thèmes majeurs de recherche pour obtenir un robot mobile parfaitement autonome. Parmi les problématiques liées à la commande, celle de la navigation tient un rôle important : elle consiste à déterminer les trajectoires que le robot doit suivre pour évoluer correctement au milieu d'obstacles, en considérant qu'il dispose d'une méthode de navigation.

## **Chapitre III :**

### **APPROCHES DE NAVIGATION EN ENVIRONNEMENT INCONNU**

#### **III.1 Introduction**

Notre travail s'intéresse à la navigation d'un robot mobile en environnement inconnu. Soit un système robotique équipé de capteurs lui permettant de percevoir son environnement et d'actionneurs lui permettant de se déplacer. Notre but consiste alors à déterminer un mouvement du robot qui [110]:

- évite toutes collisions avec les obstacles au cours de la navigation.
- mène à un but prédéterminé.
- respecte les contraintes cinématiques sur le mouvement du robot.

La détermination du mouvement pour un robot autonome a été largement abordée au cours de ces dernières années. On distingue habituellement deux grandes familles d'approches : les approches délibératives et les approches réactives. Le principe des approches délibératives est de calculer au préalable un chemin ou un ensemble de trajectoires entre une position initiale et une position finale à partir de la connaissance à priori de l'environnement dans lequel évolue le robot. Les approches réactives utilisent l'environnement perçu afin de générer un mouvement à exécuter sur un pas de temps, à la suite duquel un autre mouvement est généré et ainsi de suite. Ces mouvements sont appelés "mouvements partiels". Une représentation de l'environnement est ainsi construite au fur et à mesure du déplacement : la navigation est donc possible en environnement incertain comme en environnement dynamique.

L'inconvénient majeur des approches délibératives est leur incompatibilité avec des contraintes de calculs en temps-réel, rendant difficile leur utilisation en environnement dynamique. Quant aux approches réactives, leur inconvénient principal

est celui des minima locaux dont il peut être difficile de sortir sans connaître le chemin global permettant de rejoindre le but.

Ainsi, pour pallier aux problèmes inhérents à ces 2 familles, un 3ème type d'approche a vu le jour, il s'agit de la famille des méthodes hybrides. Elles sont généralement composées d'un algorithme réactif dont les décisions se font en connaissance d'un chemin planifié par une fonction délibérative.

De nombreux travaux actuels combinent désormais une planification globale et un évitement d'obstacle local afin de palier les défauts de chacune de ces deux catégories d'approches [110-113].

### **III.2 Approches délibératives**

Les approches dites délibératives consistent à résoudre un problème de planification de mouvement. La planification de mouvement est la détermination à priori d'une stratégie de mouvement entre une position initiale et une position finale du robot à partir d'une représentation de l'environnement dans lequel il évolue.

Initialement motivée par l'utilisation de bras manipulateurs dans l'industrie manufacturière (Figure III.1), la planification pour de tels systèmes robotique disposant de multiples degrés de liberté cherchait à déterminer la séquence de poses (ou configuration) du robot permettant à son élément terminal de rejoindre une position but sans entrer en collision avec les objets se trouvant à portée du robot. Formellement, une configuration  $q$  est définie par l'ensemble des paramètres indépendants définissant de manière unique la position et l'orientation de chaque partie du robot. L'espace de configurations  $C$  définit l'ensemble de toutes les configurations possibles du système robotique.



**Figure III.1** Bras manipulateurs dans une industrie automobile

Parmi toutes les configurations constituant  $C$ , certaines d'entre elles caractérisent une pose du robot telle que l'un des éléments qui le composent se trouve en collision avec les obstacles. On note généralement  $C_{obs}$  l'ensemble de toutes les configurations en collision et son complément  $C_{libre} = C \setminus C_{obs}$  l'ensemble des configurations libres. Le problème basique de planification de mouvement consiste alors à déterminer un chemin  $C$  (une séquence continue de configurations entre une configuration initiale  $q_{init}$  et une configuration finale  $q_{but}$ ) tel que toutes configurations  $q$  de  $C$  appartiennent à  $C_{libre}$ . Souhaitant déterminer un mouvement sans collisions pour un robot évoluant dans un environnement dynamique caractérisé par des obstacles mobiles, une simple planification de chemin ne suffit pas : en effet, la position de ces obstacles évoluant au cours du temps, il est nécessaire de planifier une trajectoire (un chemin paramétré par le temps) afin de s'écarter des obstacles au cours du temps. Celle-ci indique donc par quelles configurations le système passe, mais également à quels instants et à quelles vitesses. Une trajectoire est généralement exprimée dans l'espace d'états-temps où on représente à la fois la configuration prise par le robot à cet instant et également sa vitesse instantanée. Une représentation courante de l'espace d'états est l'espace des articulations donné par  $(q, \dot{q})$ .

Un robot est néanmoins soumis à un certain nombre de contraintes. Tout d'abord, les contraintes cinématiques limitent le mouvement du système robotique. On

distingue généralement deux types de contraintes cinématiques :

– les contraintes holonomes limitent l'ensemble des configurations pouvant être prises par le robot. Par exemple, un bras manipulateur tenant un verre rempli d'eau et ayant pour tâche de ne pas le renverser ne pourra pas passer par une configuration à laquelle le verre se retrouve à l'envers.

– les contraintes non-holonomes limitent le mouvement instantané du système mais pas l'ensemble des configurations qu'il peut occuper. Par exemple une voiture disposant d'un angle de braquage maximal ne pourra pas se translater sur le côté. Par contre, en effectuant une manœuvre, elle pourra néanmoins rejoindre cette position.

Ensuite les contraintes dynamiques sont des contraintes sur l'état du système robotique et limitent l'évolution de son mouvement au cours du temps. Par exemple une voiture à l'arrêt dispose d'une accélération maximale l'empêchant d'atteindre sa vitesse maximale instantanément.

Le problème étant posé, nous présentons ici un bref aperçu des principales approches délibératives ayant retenu notre attention [110].

### **III.2.1 Méthodes par graphes**

Le principe de ces méthodes est de tenter de capturer la topologie de l'espace de recherche (espace de configuration ou espace d'état du robot) dans le but de simplifier le problème à une recherche dans un graphe. Elles sont donc constituées de deux étapes :

- construction du graphe dans l'espace de recherche approprié.
- parcours du graphe dans le but de déterminer un chemin ou une trajectoire entre les configurations initiale et finale.

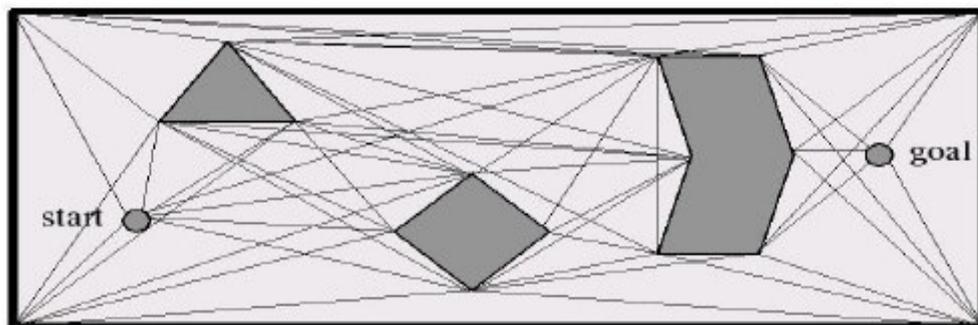
Le parcours du graphe s'effectue la plupart du temps de la même manière : un algorithme heuristique est utilisé dans le but d'éviter l'exploration complète de l'espace de recherche [116-117]. La construction du graphe peut, quant à elle,

fortement varier : alors que les premiers travaux de planification s'intéressaient à trouver un chemin pour des systèmes dépourvus de contraintes sur leur mouvement ou pouvant se déplacer dans toutes les directions (systèmes holonomes), les recherches actuelles prennent en compte les contraintes cinématiques et dynamiques des robots étudiés, et planifient des trajectoires dans l'espace-temps afin de considérer le mouvement futur des obstacles mobiles. La représentation de l'espace de recherche, s'est en conséquence adaptée à ces évolutions [118-120].

Nous présentons ci-dessous les principales méthodes de représentation de ces espaces de recherches pour des systèmes et environnement de plus en plus contraints.

### III.2.1.1 Graphe de visibilité

C'est la première méthode de planification de chemin connue, Elle consiste à relier chaque sommet des enveloppes convexes d'obstacles polygonaux à tout autre sommet visible de cet ensemble (Figure III.2). On obtient ainsi un graphe dans lequel on peut effectuer une planification après avoir relié les positions initiale et finale aux sommets de cet ensemble les plus proches. Notez que cette technique autorise les configurations de contact entre le système mobile  $A$  et les obstacles, c'est l'une des raisons pour lesquelles elle est relativement peu utilisée [121].



**Figure III.2** Chemin déterminé entre deux configurations à partir d'un graphe de visibilité

### III.2.1.2 Diagramme de Voronoï

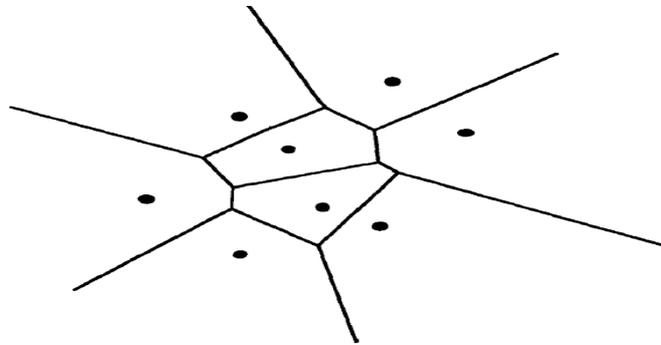
Afin d'éviter le contact avec les obstacles, une seconde option combinatoire consiste à construire un diagramme de Voronoï [122] dans l'espace de configurations dans lequel évolue le robot mobile. Le diagramme de Voronoï consiste à générer un partitionnement du plan en polygones et ceci dans un contexte de structure de données de type graphe.

Soit  $S = \{M_1, M_2, M_3, \dots, M_n\}$  un ensemble de  $n$  points du plan euclidien distincts deux à deux. Ces points sont couramment appelés germes. On appelle polygone de Voronoï associé au point  $M_i$ , l'ensemble des points les plus proches de  $M_i$  que des autres points de  $S$ . Formellement on peut écrire :

$$P(M_i) = \{M \in R^2 / d(M, M_i) < d(M, M_j) \text{ pour } j \neq i\} \quad (\text{III.1})$$

On appelle diagramme de Voronoï de l'ensemble  $S$ , noté  $V(S)$ , le graphe planaire formé par les frontières des polygones  $P(M_i)$ , notées  $Fr(P(M_i))$  (Figure III.3).

$$V(S) = \{M \in R^2; \exists i \text{ avec } i \neq j \text{ tel que } M \in Fr(P(M_i)) \text{ et } M \in Fr(P(M_j))\} \quad (\text{III.2})$$



**Figure III.3** Illustration de la définition d'un diagramme de Voronoï

Le diagramme de Voronoï (Figure III.3) étant défini, on dispose donc du graphe induit où chaque nœud correspond à un sommet des arêtes du diagramme et chaque liaison correspond aux arêtes elles-mêmes. On peut alors comme précédemment effectuer une planification par recherche dans ce graphe. Notez cependant que ce diagramme est assez facilement définissable dans un espace de dimension 2 mais qu'il devient très complexe à calculer dès que l'on passe en dimension supérieure.

### III.2.1.3 Décomposition en cellules

Un autre exemple d'approches consiste en une décomposition cellulaire de l'espace de configuration  $C$  à partir d'une représentation simple de l'espace de configuration  $C$ , cette approche consiste à diviser cet espace en un nombre fini de sous-espace convexes. Par exemple, dans un espace de configuration  $C = \mathbb{R}^2$  dont la représentation des obstacles est polygonale, une décomposition cellulaire verticale ou par triangulation peut être facilement calculée (Figure III.4). Un graphe est alors construit comme suit : les nœuds sont définis aux barycentres des cellules obtenues et au milieu de leurs cotés. Une arête relie ensuite chaque nœud défini sur un côté au barycentre des deux cellules adjacentes. Comme précédemment, en reliant les configurations initiale  $q_0$  et finale  $q_f$  aux nœuds du graphe les plus proches, une recherche heuristique dans le graphe résultant permet de trouver un chemin liant ces deux configurations.

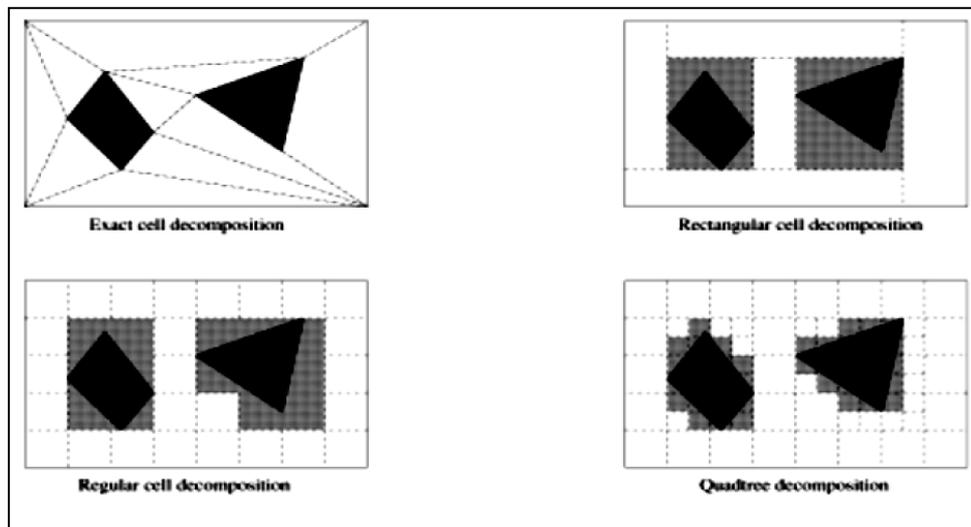


Figure III.4 Exemples des décompositions en cellules

Des méthodes de décomposition cellulaires plus complexes telles les balles connectées ont été conçues pour des environnements non structurés et des espaces de recherche (configurations ou états) de plus fortes dimensions [123-125]. En couvrant l'espace de recherche libre par des sphères se chevauchant de la dimension de cet espace, il est possible comme précédemment de construire un graphe connexe dont

une arête est définie entre chaque couple de sphères dont l'intersection est non vide.

Bien qu'attirantes par leur simplicité, les méthodes combinatoires présentées ci-dessus sont en pratique très peu souvent utilisables : pour un système robotique quelque peu complexe (forte dimensionnalité de l'espace de configuration, systèmes redondants, topologie des obstacles quelconque), une représentation des obstacles  $C_{obs}$  dans l'espace de configuration est difficilement accessible, et leur calcul explicite est bien trop complexe. De plus, la géométrie du graphe ne se soucie guère des contraintes sur le mouvement du système. Un robot disposant par exemple de contraintes non holonomes sera absolument incapable de suivre les arêtes du graphe sans une adaptation du plan. Des méthodes alternatives ont alors été apportées afin d'éviter la caractérisation de  $C_{obs}$ .

#### III.2.1.4 Echantillonnage par grilles régulières

Lorsque la complexité du système robotique  $A$  est telle qu'il est difficile de déterminer la topologie de l'espace de configurations  $C$  (et l'espace  $C_{obs}$  des configurations du robot en collision avec un obstacle de l'environnement), une méthode alternative consiste à discrétiser  $C$  et à construire une approximation conservative de l'espace libre  $C_{libre} = C \setminus C_{obs}$ . Pour ce faire, un échantillonnage de l'espace par une grille régulière de dimension  $n$  peut être effectué. Le graphe  $G$  en résultant est construit en définissant un nœud pour chaque échantillon et en le liant par une arête à chacun de ses  $2n$  voisins directs. Une planification peut alors être effectuée entre deux configurations  $q_0$  et  $q_f$  en assimilant chacune de ces configurations à un nœud du graphe  $G$ , puis en explorant itérativement à partir de  $q_0$  ses nœuds voisins jusqu'à atteindre  $q_f$ . La détermination de l'obstruction des nœuds et des arêtes du graphe par les obstacles peut ainsi être effectuée uniquement lors de leur exploration par un module de vérification de collision (souvent considéré comme une boîte noire pour ce genre d'approches). L'espace libre atteignable par une telle grille est un sous-espace de  $C_{libre}$ . La garantie de trouver une solution s'il en existe une est donc amoindrie et dépendante de la résolution de la grille.

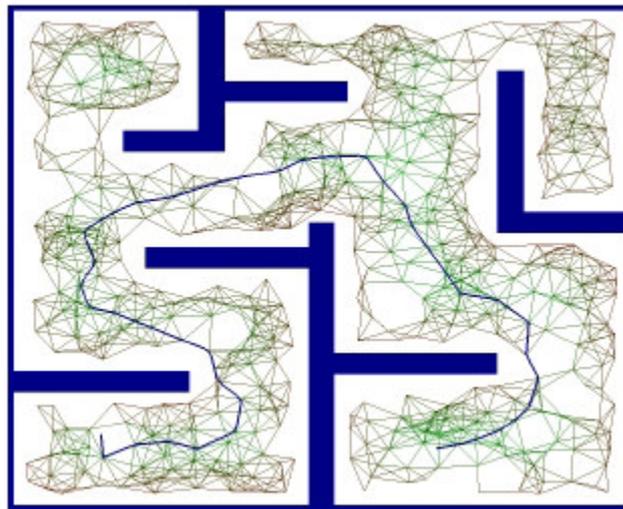
Afin d'accélérer le processus de recherche sur le graphe, une représentation de l'environnement par grilles multi résolution fut proposé par la suite [126-127]. Une recherche de chemin peut alors être effectuée en premier lieu à basse résolution. Si aucune solution n'est trouvée les résolutions supérieures sont examinées jusqu'à ce qu'une solution ait été trouvée.

Cet échantillonnage par grilles a également été la source de représentation des espaces libre  $C_{libre}$  et obstrué  $C_{obs}$  par grilles d'occupation probabilistes. Une grille d'occupation probabiliste est une grille représentant une discrétisation régulière de l'espace dans laquelle chaque cellule est caractérisée par une probabilité d'occupation évaluée lors de la navigation par les capteurs du système robotique. La vérification de collision en chaque nœud du graphe induit est alors effectuée en estimant la probabilité de collision de la cellule de la grille probabiliste associée. L'avantage de telles grilles est sa possibilité de mise à jour au cours du temps; lorsqu'un obstacle mobile se déplace ou lorsqu'un nouvel obstacle est détecté par les capteurs du système, les probabilités d'occupation des cellules de la grille associées sont réévaluées. Le graphe induit peut en conséquence être modifié, mais s'il désactive un nœud ou une arête choisis lors de la planification, un nouveau mouvement doit être replanifié.

### III.2.1.5 Cartes de routes probabilistes

Dans le cas d'espace de recherche de forte dimension, une discrétisation régulière de l'espace peut s'avérer bien trop coûteuse. Afin de diminuer la taille du graphe sur cet espace, une solution consiste à construire une carte de route probabiliste [128] (en anglais : probabilistic roadmap) sur celui-ci. La construction d'une carte de route probabiliste s'effectue en sélectionnant des configurations aléatoires dans l'espace de recherche (Figure III.5). Chaque nouvelle configuration est ajoutée au graphe si elle n'est pas en collision avec les obstacles de l'environnement. Dans ce cas un nouveau nœud est créé dans le graphe pour celle-ci. Des arêtes sont alors ajoutées à partir de cette configuration vers d'autres configurations appartenant déjà à la carte de route dans le cas où ces deux configurations sont suffisamment proches, et qu'il existe un

chemin libre de collisions entre celles-ci. La construction de la roadmap se termine généralement lorsqu'un nombre prédéterminé de nœuds constituent celle-ci. Comme dans le cas des grilles régulières sur l'espace de configuration, l'espace libre  $C_{libre}$  est caractérisé par un sous-ensemble de configurations atteignables et de chemins admissibles. La représentation de l'espace libre étant dépendante d'un nombre aléatoire d'échantillons.



**Figure III.5** Chemin planifié pour un robot de type différentiel à partir d'une roadmap calculée dans l'espace de configuration du robot

La qualité de telles approches est alors évaluée par rapport à deux critères : sa densité et sa dispersion. La dispersion reflète la taille maximale de l'espace libre  $C_{libre}$  non couverte par les nœuds composant la carte de route probabiliste. La densité s'illustre par son aptitude à s'approcher aussi près que possible de toutes les configurations de l'espace libre  $C_{libre}$ . De nombreuses extensions de ce type d'approches ont vu le jour afin d'optimiser ces deux paramètres tout en limitant le nombre de nœuds nécessaires.

Tout d'abord, [129] propose un échantillonnage aux bornes de  $C_{libre}$  afin de pouvoir déterminer simplement un chemin vers une configuration proche des obstacles de l'environnement. Pour cela, lorsqu'une configuration aléatoirement choisie se

trouve en collision, elle est déplacée aléatoirement, puis une fois libre, elle est autant que possible connectée au reste du graphe. A l'opposé, [130] essaient de maximiser l'espace libre autour de chaque nœud du graphe.

Dans [131], une probabilité de distribution est associée à chaque nœud du graphe. Lors de l'insertion d'un nouveau nœud, son positionnement est optimisé afin de maximiser la probabilité de distribution sur l'ensemble de l'espace libre  $C_{libre}$  et de pouvoir atteindre ainsi de nouvelles configurations non visibles jusqu'alors.

Une carte de route par visibilité est proposée en [132] où deux types de nœuds sont définis : les gardes et les connecteurs. Un nœud est un garde s'il n'est visible dans  $C_{libre}$  par aucun autre garde. Un connecteur est un nœud visible par au moins deux gardes. En construisant un graphe connecté tel que toute région de  $C_{libre}$  soit visible par un garde et que le nombre de nœuds total soit minimal, on obtient une carte de route de faible taille (limitant ainsi le temps de recherche) et pouvant aisément connecter n'importe quel couple de nœuds de  $C_{libre}$ . Enfin dans le cas de navigation en environnement dynamique, [133] propose une mise à jour dynamique de la carte de route probabiliste. En présence d'obstacles mobiles, certains nœuds et arêtes sont ainsi désactivés ou réactivés afin d'adapter la planification aux mouvements des obstacles sans devoir reconstruire la carte intégralement.

### III.2.1.6 Méthodes de replanification

En présence d'obstacles mobiles, certaines méthodes présentées ci-dessus permettent une adaptation du graphe sur lequel le chemin ou la trajectoire du robot est planifié, par exemple en déplaçant, activant ou désactivant des nœuds et des arêtes du graphe. Dans le cas où le plan initial passait par un nœud ou une arête n'étant plus valide, il est alors nécessaire de replanifier tout ou partie du mouvement suivi. Pour cela, diverses méthodes ont été proposées : Stentz [134] a initialement proposé une extension de l'algorithme de recherche heuristique dans le cadre de la navigation de robots mobiles. Lors de l'invalidation d'un plan, celui-ci replanifie localement la partie du plan obstruée par les obstacles. Likhachev et al. [135] ont proposé une extension "anytime" de l'algorithme : à chaque instant, le robot effectue une recherche le plus

loin possible vers le but en un temps de décision constant.

D'autres méthodes de replanification ont vu le jour modifiant une partie du plan obstrué en cas d'obstacles inattendus et améliorant le chemin suivi lorsque le temps le permet [136].

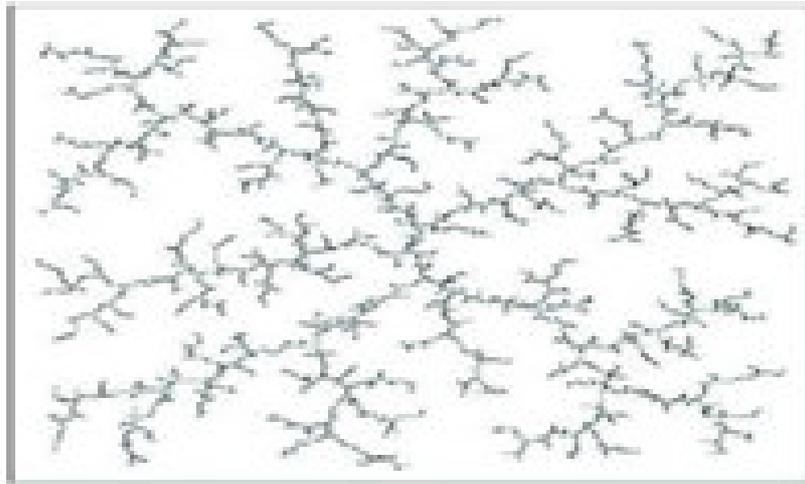
### III.2.2 Méthodes par arbres

En parallèle de la planification classique par exploration d'un graphe de recherche sont apparues les méthodes par arbres. Celles-ci consistent, à partir de la configuration initiale du système, à construire un arbre se développant dans toutes les directions autour du robot dans l'espace de recherche. Elles sont donc bien adaptées dans le cas d'espaces de recherche à forte dimensionnalité.

#### III.2.2.1 Exploration rapide des arbres aléatoires (RRT)

Elle représente certainement l'une des approches de planification de mouvement les plus répandues à l'heure actuelle [114]. A partir d'une configuration initiale  $q_0$ , l'espace de configuration du système est exploré en choisissant aléatoirement à chaque itération une nouvelle configuration  $q_{nv}$  non obstruée par les obstacles vers laquelle se diriger. La branche la plus proche de l'arbre déjà construit est alors déterminée puis étendue en direction de  $q_{nv}$ . En répétant le processus, l'espace de recherche est alors rapidement couvert, et un chemin vers toutes configurations de cet espace peut alors être facilement déterminé s'il en existe un (Figure III.6). Le but de la planification étant néanmoins d'atteindre une configuration finale  $q_f$ , le processus essaie de déterminer un chemin liant la configuration la plus proche de l'arbre à  $q_f$  après un certain nombre d'itérations de l'expansion de l'arbre.

Dans le cas où le système évolue en environnement dynamique, une extension anytime des RRTs a également été proposée [137] : l'arbre de recherche est mis à jour progressivement, et à chaque pas de temps le contrôle à appliquer est déterminé à partir de la racine de l'arbre menant vers la position se rapprochant le plus près du but.



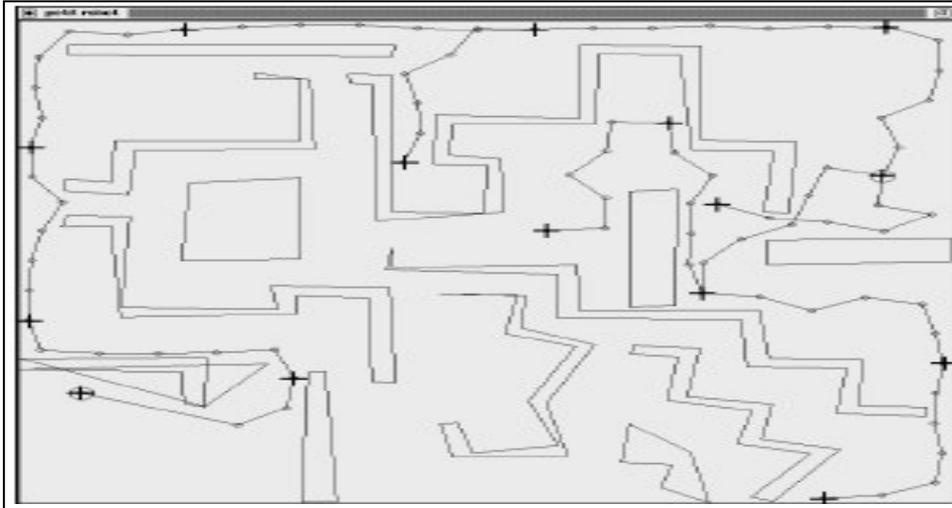
**Figure III.6** Exploration rapide des arbres aléatoires : Etapes successives de construction de l'arbre de recherche dans l'espace des configurations du système robotique

### III.2.2.2 Fil d'Ariane

Cette méthode explore l'espace de configuration du robot à partir de sa configuration initiale en construisant un arbre de recherche par l'alternance de deux algorithmes (Figure III.7) :

- **Explore** : Cet algorithme a pour but d'explorer l'espace de configuration libre en y plaçant des balises aussi loin que possible des balises existantes. A l'initialisation, la seule balise disponible est la configuration initiale du système. A chaque nouvel appel de la méthode "Explore", l'arbre est étendu à partir d'une des balises existantes choisie aléatoirement.

- **Search** : Cet algorithme recherche autour d'une balise posée s'il est possible d'accéder directement à la configuration finale par un mouvement simple (chemin de Manhattan).



**Figure III.7** Fil d'Ariane : Méthode de planification alternant deux algorithmes

En répétant successivement ces deux étapes, l'arbre de recherche va s'étendre rapidement sur tout l'espace de configuration accessible à partir de la configuration initiale, jusqu'à converger vers le but, ou s'arrêter s'il n'est plus possible de placer une balise à moins d'une distance minimale de celles déjà posées.

D'autres méthodes moins connues de planification par expansion d'un arbre peuvent être notées : Parmi celles-ci on trouve une planification expansive sur l'espace de configuration [138] consistant à étendre itérativement un arbre de recherche en sélectionnant un nœud de l'arbre  $q_{ext}$  à étendre à partir d'une probabilité inversement proportionnelle au nombre de nœuds dans la région qui l'entoure. Une nouvelle configuration  $q_{nv}$  est alors choisie dans son voisinage proche et un chemin déterminé entre ces deux configurations.

Dans la même idée, [139] propose une planification par marche aléatoire. Son principe est simple : un nœud de l'arbre déjà construit est choisi aléatoirement et étendu dans une direction aléatoire. La longueur du chemin à parcourir à chaque étape ainsi que la variation de la direction à prendre par rapport à l'étape précédente sont déterminées à partir des observations sur l'environnement, obtenues lors des extensions précédentes de l'arbre.

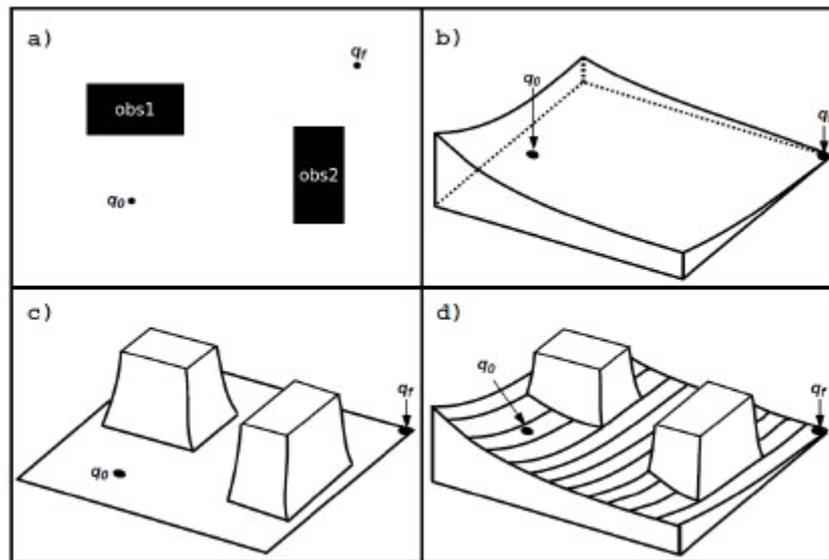
### III.3 Approches réactives

Les approches réactives consistent à calculer à chaque pas de temps (après récupération des informations sur l'environnement fournies par les capteurs du système) le contrôle instantané à appliquer sur les actionneurs du système.

#### III.3.1 Approches par champs de potentiel

Les approches dites par champs de potentiel initialement proposées par Khatib [1] consistent à considérer le robot mobile comme une particule soumise à divers champs électriques régissant son mouvement. Ses travaux considéraient les deux suivants (Figure III.8) :

- un champ de potentiel attractif provenant de la position finale à atteindre (la cible).
- un champ de potentiel répulsif provenant des obstacles statiques et mobiles de l'environnement.



**Figure III.8** Calcul d'un chemin entre deux configurations  $q_0$  et  $q_f$  par la méthode de champs de potentiels. (a) Espace de configuration du robot et représentation des obstacles obs1 et obs2 dans celui-ci. (b) Champ attractif généré par la position finale. (c) Champ répulsif exercé par les obstacles. (d) Combinaison des deux méthodes

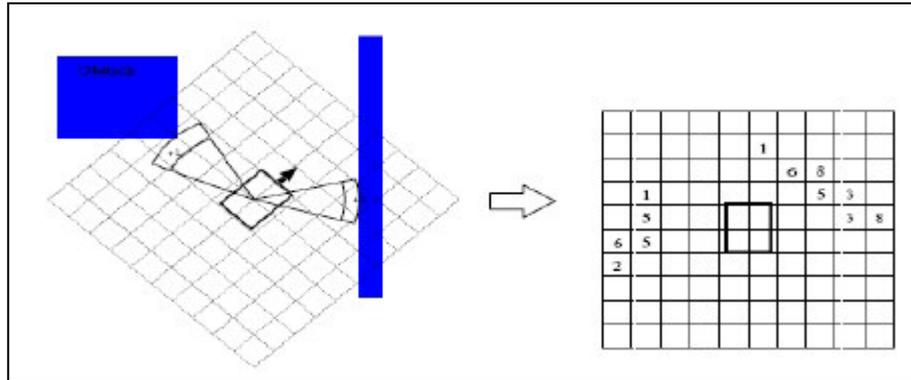
Initialement conçus pour le calcul du mouvement de bras manipulateurs, cette méthode dispose de l'avantage de calculer ces champs de potentiels dans l'espace de travail  $W$  (espace euclidien  $IR^2$  ou  $IR^3$  dans lequel une représentation des obstacles est disponible), définissant ainsi une direction privilégiée à suivre par l'élément terminal du bras dans cet espace. Une modification de la configuration du robot dans son espace articulaire est alors déduite de ce champ dans un second temps.

Bien que simple et élégante, cette méthode possède de nombreux inconvénients mis en évidence par Koren et Borenstein [2]. D'une part, cette approche est sujette à des minima locaux. Par conséquent, la convergence vers le but n'est pas assurée. D'autre part, ces potentiels peuvent donner lieu à de fortes oscillations du mouvement du robot en présence d'obstacles et principalement lorsque celui-ci navigue dans des passages étroits (couloirs, portes). Enfin, le vecteur de déplacement désigné par le champ de potentiel ne prend en aucun cas en compte la cinématique ou la dynamique du système robotique considéré. Un robot disposant de contraintes non-holonomes aura de sérieuses difficultés à suivre une telle direction.

Malgré ces limitations, de nombreuses techniques de navigation ont découlé de ces champs de potentiels. Ils ont par exemple été adaptés à la navigation au milieu d'obstacles mobiles dans [140] en prenant en compte non seulement la distance aux obstacles mais également la vitesse de ces derniers pour calculer les champs répulsifs.

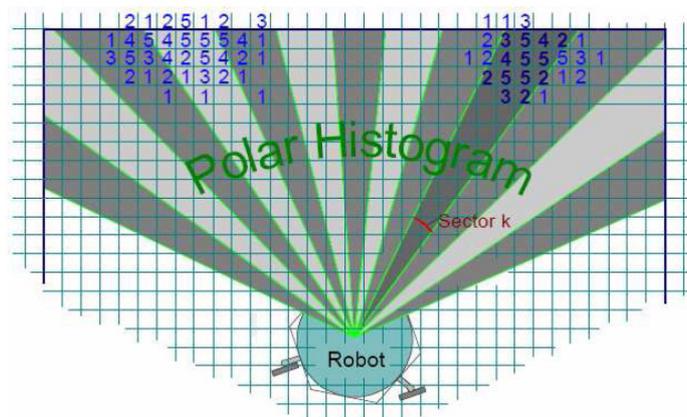
### III.3.2 Histogramme de champs de vecteurs

Dans la lignée des approches par champs de potentiels, sont apparus les histogrammes par champs de vecteurs (en anglais : Vector Field Histogram - VFH). Ceux-ci, introduits par Koren et Borenstein [2], consiste dans un premier temps à représenter l'environnement par une grille d'occupation centrée sur le robot, où chaque cellule contient une valeur entière correspondant à la probabilité d'y trouver un obstacle (Figure III.9).



**Figure III.9** Représentation de l’environnement sous forme de grille d’occupation centrée autour du robot

Afin de choisir une direction à suivre, un histogramme polaire est construit à partir de la grille d’occupation : en discrétisant les différentes directions possibles autour du robot, l’histogramme polaire est construit en pondérant pour chaque secteur de la discrétisation polaire les cellules traversées de la grille d’occupation contenant des obstacles. Une fois cet histogramme polaire construit, des “vallées candidates” sont déterminées comme les suites de secteurs contigus de l’histogramme polaire libres d’obstacles (Figure III.10). La direction à prendre par le système est alors déterminée par le milieu de la vallée menant le plus directement au but.

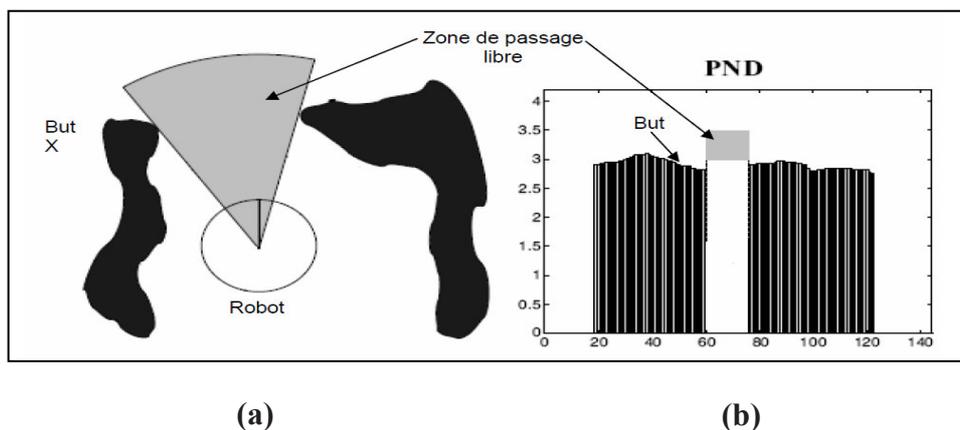


**Figure III.10** Histogramme polaire calculé à partir de la grille d’occupation

Initialement conçue pour la navigation des robots holonomes (pouvant naviguer dans toutes les directions), cette méthode a été étendue à plusieurs reprises afin de prendre en compte les dimensions du robot (par un espace de configuration implicite) et ses contraintes de vitesse. Plus tard, les VFH ont été combinés à une recherche heuristique afin de trouver un chemin menant vers le but et d'échapper ainsi aux minima locaux [141]. Les méthodes VFH disposent néanmoins encore de fortes limitations : Elles ne prennent ni en compte la dynamique du système robotique, ni l'éventuelle présence d'obstacles mobiles ; le mouvement instantané du robot est calculé uniquement à partir des informations sur la position courante des obstacles, leur vitesse n'est en aucun cas considérée.

### III.3.3 Navigation par diagrammes de proximité

L'approche de navigation par diagrammes de proximité (en anglais : Nearness Diagram Navigation - ND) proposée par Minguez et Montano [142] fortement inspirée des VFH se base sur la topologie de l'environnement pour choisir un contrôle à appliquer à chaque instant. Pour ce faire, deux diagrammes polaires sont construits : l'un représente la distance du centre du robot aux obstacles dans toutes les directions autour de celui-ci (Figure III.11) ; le second calcule cette même distance à partir des contours du robot afin d'estimer la distance de sécurité conservée.



**Figure III.11** Navigation par Diagrammes de Proximité (ND) : (a) Représentation de l'environnement autour du robot. (b) Caractérisation des vallées disponibles en repérant les discontinuités du graphe de proximité des obstacles

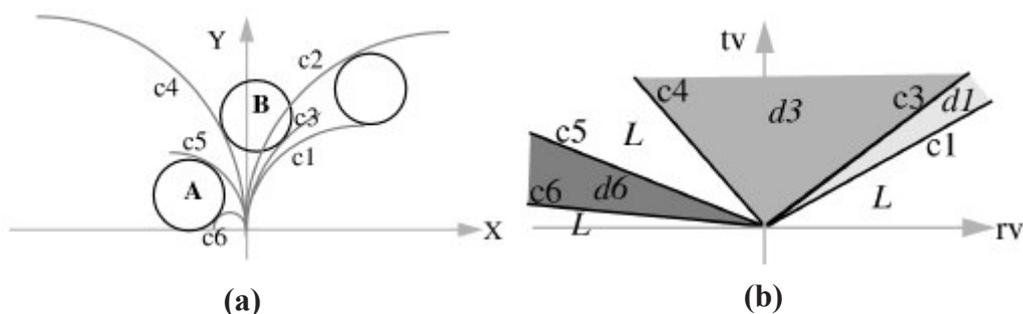
La méthode ND utilise une stratégie déterministe de choix du comportement à adopter en fonction de la distance aux obstacles les plus proches, et en fonction de la topologie des obstacles qui l'entourent. Les options possibles se résument à :

- contourner un obstacle
- passer entre deux obstacles
- aller tout droit vers le but ainsi qu'à choisir une vitesse faible ou élevée suivant la proximité des obstacles.

De la même manière que les approches présentées précédemment, cette méthode n'est pas vraiment adaptée pour naviguer au milieu d'obstacles mobiles et ne prend pas en compte la dynamique du système.

### III.3.4 Méthode de navigation courbure-vélocité

A l'opposé des méthodes dites directionnelles présentées ci-dessus (essayant de se diriger vers le but en restant hors de portée des obstacles), se trouvent les méthodes dites de l'espace des vitesses. Celles-ci ont été introduites avec la méthode de navigation courbure-vélocité (en anglais Curvature-Velocity Method - CVM) proposée par Simmons [143]. Son principe est le suivant : au lieu de choisir une direction à suivre en fonction de la position finale et de l'environnement puis de calculer une commande à appliquer dans l'espace des vitesses, la CVM évalue les différentes trajectoires possibles dans l'espace des vitesses puis sélectionne dans cet espace un mouvement à exécuter permettant d'éviter les obstacles et de se rapprocher du but.



**Figure III.12** Méthode de navigation courbure-vélocité (CVM) : (a) Trajectoires candidates représentées dans l'espace de travail  $W$ . (b) Contrôles correspondant aux trajectoires candidates dans l'espace des vitesses linéaire et angulaire

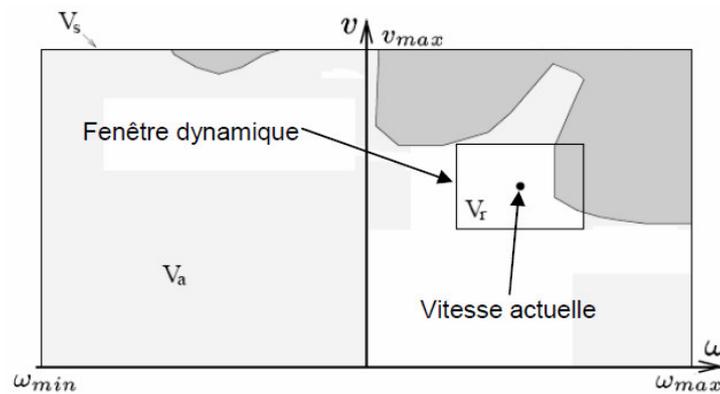
Afin de choisir quelle commande appliquer, à chaque couple vitesse linéaire / vitesse angulaire  $(\mathbf{v}, \boldsymbol{\omega})$  est associée une fonction de coût  $J$  basée sur la distance aux obstacles, la modification de l'orientation du système par rapport au but et sur le temps nécessaire pour rejoindre le but (une vitesse linéaire élevée est donc préférée). La commande maximisant cette fonction de coût est alors sélectionnée et envoyée au robot lors du prochain pas de temps.

D'après les propres auteurs de la CVM, celle-ci n'est pas toujours apte à garantir la sécurité du mouvement. La CVM peut en effet pousser le robot à passer relativement proche des obstacles et rentrer en collision avec ces derniers. Afin d'éviter ce genre de comportements, ces travaux ont été étendus. La nouvelle méthode intitulée "Lane Curvature Method" (LCM) est un algorithme en deux étapes : Tout d'abord, des "lignes" permettant de passer entre les obstacles en maximisant la distance par rapport à ceux-ci sont déterminées. Ensuite une commande permettant soit de suivre une ligne soit de rejoindre l'une d'entre elles est calculée comme précédemment par CVM.

### III.3.5 Fenêtre dynamique

Découlant des CVMs, l'approche de fenêtre dynamique (en anglais Dynamic Window - DW) présentée dans [144] conserve le principe de sélection d'un mouvement à suivre dans l'espace des vitesses. A l'instar de la plupart des approches réactives présentées précédemment, les DWs ont été développées pour des robots de type différentiel, contrôlés en vitesse linéaire  $\mathbf{v}$  et vitesse angulaire  $\boldsymbol{\omega}$ . A chaque pas de temps, une nouvelle commande constante  $(\mathbf{v}, \boldsymbol{\omega})$  est sélectionnée parmi les vitesses respectant les contraintes suivantes :

- contraintes cinématiques :  $\mathbf{v} \in [0; \mathbf{vmax}]$  et  $\boldsymbol{\omega} \in [-\boldsymbol{\omegamax}; \boldsymbol{\omegamax}]$  où  $\mathbf{vmax}$  et  $\boldsymbol{\omegamax}$  sont les vitesses maximales admissibles.
- contraintes dynamiques : Les accélérations linéaire et angulaire  $a$  et  $\eta$  appliquées entre chaque pas de temps doivent être bornées.  $a \in [-amax; amax]$  et  $\eta \in [-\eta\mathbf{max}; \eta\mathbf{max}]$  où  $amax$  et  $\eta\mathbf{max}$  sont les accélérations linéaires et angulaires maximales.
- garantie de sécurité passive : Le système doit être certain de pouvoir s'arrêter avant d'entrer en collision avec un obstacle.



**Figure III.13** Fenêtre Dynamique (DW) : Calcul du mouvement à appliquer à chaque pas de temps dans l'espace des vitesses

Notons  $\tilde{v}_s$  l'ensemble des vitesses admissibles respectant les contraintes cinématiques,  $\tilde{v}_d$  les vitesses respectant les contraintes dynamiques et  $\tilde{v}_a$  les vitesses permettant de s'arrêter avant d'entrer en collision avec les obstacles de l'environnement (Figure III.13). L'ensemble des commandes candidates  $\tilde{v}_r$  est alors défini par :

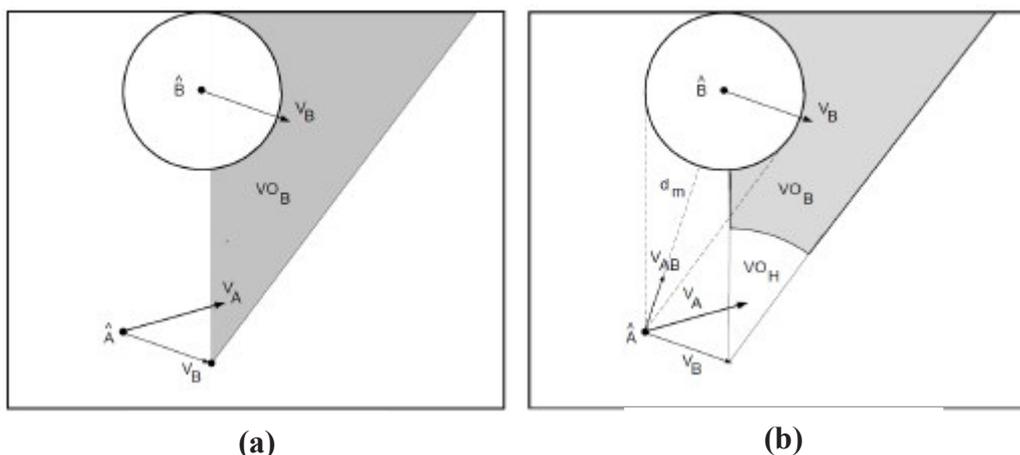
$$\tilde{v}_r = \tilde{v}_s \cap \tilde{v}_d \cap \tilde{v}_a$$

Une fois cet ensemble défini, une fonction de coût similaire aux CVMs est définie. La commande choisie est de même la commande maximisant cette fonction de coût. De par sa volonté d'assurer une sécurité passive, l'approche de DW est devenue l'une des approches de navigation réactives les plus populaires de nos jours. Cette approche initiale souffrait d'une limitation importante : seule la position courante des obstacles était prise en compte, mais pas leur mouvement. La navigation en environnement dynamique avec ce type d'approche en était donc fortement compromise. Afin de palier ce problème, les auteurs de [144] ont proposé une extension intitulée "Time Varying Dynamic Window". Celle-ci calcule à chaque instant un ensemble de trajectoires probablement suivies par les obstacles dans le futur. Une vérification de collision à court terme peut donc être opérée.

### III.3.6 Représentation des obstacles dans l'espace des vitesses

Dans le but de prendre en compte non seulement la dynamique du système robotique, mais également la dynamique de l'environnement dans lequel évolue ce système, Fiorini et Shiller ont introduit une approche intitulée "Velocity Obstacles" (VO) [145].

Supposant une connaissance à priori du mouvement des obstacles mobiles, l'approche consiste à caractériser parmi les vitesses admissibles (respectant les contraintes cinématiques et dynamiques du système), celles menant à une éventuelle collision avec les obstacles dans le futur (jusqu'à un certain horizon temporel  $th$ ). En supposant qu'un obstacle  $B$  va conserver une vitesse constante dans un futur proche (par exemple par approximation linéaire de sa vitesse courante), il est possible de déterminer les vitesses relatives du système  $A$  à cet obstacle menant à une collision dans le futur. L'ensemble de ces vitesses "interdites" s'illustre graphiquement comme présenté sur la Figure III.14(a) par un cône de vecteurs vitesse interdit. Certaines de ces vitesses ne conduiront bien sûr à une collision qu'après un temps relativement élevé.



**Figure III.14** Représentation des obstacles dans l'espace des vitesses (VO) : (a) Calcul du cône des vitesses interdites pour un horizon temporel infini. (b) Calcul du cône des vitesses interdites pour un horizon temporel limité

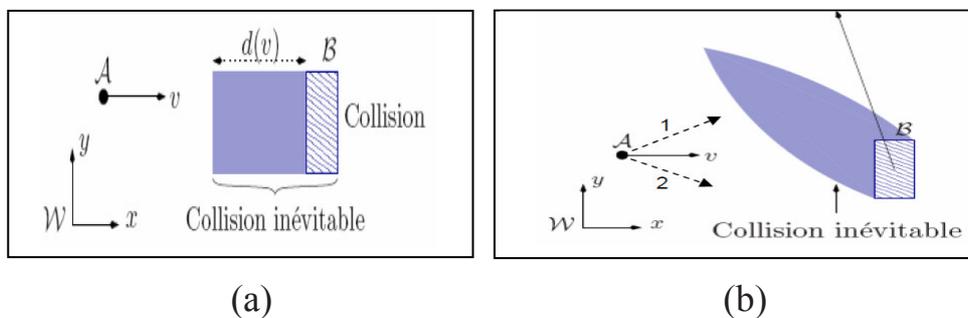
En limitant les VOs à un horizon temporel  $th$  (Figure III.14(b)), on obtient ainsi une approximation raisonnable des vitesses interdites pour le système  $A$ .

Plusieurs extensions de ces travaux ont vu le jour ces dernières années, telle la prise en compte d'approximation du modèle du futur des obstacles plus complexes ou encore une tentative de définition du "bon horizon temporel" nécessaire à la garantie de la sécurité du mouvement [146]. La détermination de cet horizon temporel est encore un sujet prêtant fortement à débattre à l'heure actuelle.

### III.3.7 Le concept d'état de collision inévitable

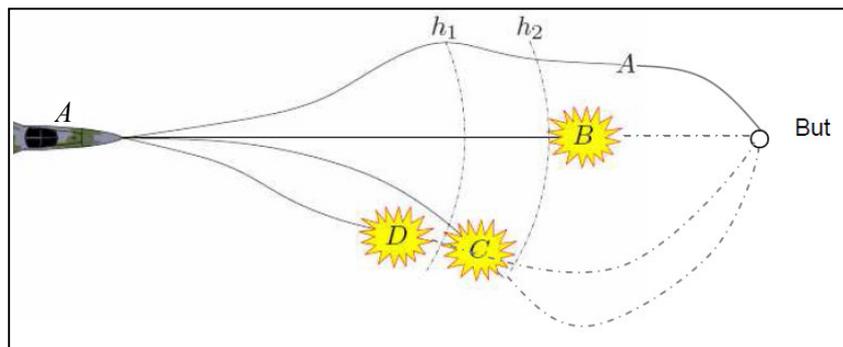
Le concept de collision inévitable définit tout état pour lequel, une collision sera inévitable quoi que soit les décisions futures [147]. Il se décompose en trois règles:

- 1) *la prise en compte des contraintes cinématiques et dynamiques du robot*: un robot en mouvement a besoin d'une distance et d'un temps pour pouvoir s'arrêter. Cette notion peut s'illustrer par l'exemple suivant (Figure III.15 (a)). Soit  $A$ , une particule se déplaçant à vitesse variable  $v$ . La dynamique de cette particule est telle qu'il lui faut une distance  $d$  pour s'arrêter. Si  $A$  se retrouve à une distance inférieure à  $d$ , une collision aura lieu inévitablement.  $A$  est alors dans un état de collision inévitable.
- 2) *la prise en compte de la trajectoire d'un obstacle mobile*: en effet, si dans notre exemple, la particule  $A$  est capable de changer de direction, la trajectoire 1 (Figure III.15 (b)) mène à une collision alors que la trajectoire 2 permet l'évitement.



**Figure III.15** Concept de collision inévitable dans l'espace de travail  $W$  : (a) la cinématique du robot, (b) la trajectoire de l'obstacle mobile

3) *l'extension de la décision sur un horizon temporel infini*: Restreindre la décision de navigation à un horizon temporel borné c'est exclure l'ensemble des situations pouvant se produire au-delà de cet horizon. En résumé, avoir un raisonnement à court terme peut avoir des conséquences néfastes à long ou moyen terme. La figure III.16 illustre ce principe; Si le système se projette à l'horizon de temps  $h_1$ , il en déduit que  $D$  est une trajectoire aboutissant à une collision. Il est alors amené à choisir  $A$ ,  $B$  ou  $C$ . Cependant,  $B$  et  $C$  sont des trajectoires menant à une collision au-delà de  $h_1$ , il se sera donc mis en danger. Cette illustration montre aussi qu'augmenter l'horizon temporel n'est pas suffisant car si le système calcule ce qui va se passer jusqu'à  $h_2$ , il aura encore le choix entre  $A$  et  $B$ . Et seul  $A$  mène au but de façon sûre. La solution pour garantir la sûreté est alors d'utiliser un horizon temporel infini ou au moins supérieur au temps requis pour atteindre le but.



**Figure III.16** Concept de collision inévitable dans l'espace de travail  $W$  avec extension de l'horizon temporel

### III.3.8 Planification de mouvement partiel

Une dernière approche réactive mérite d'être notée : il s'agit de la planification de mouvement partiel (en anglais : Partial Motion Planning (PMP) utilisée par [148]. Celle-ci consiste à calculer réactivement, en un temps de décision fixe, une trajectoire se rapprochant le plus possible du but (Figure III.17). Cette méthode consiste en un

algorithme à trois étapes répété à chaque pas de temps :

- mise à jour du modèle de l'environnement à partir des entrées capteurs du robot.
- recherche délibérative d'une trajectoire menant à l'état but. Si le but n'a pas été atteint après un temps de décision fixe, la trajectoire calculée s'en rapprochant le plus est choisie comme trajectoire à suivre.
- enfin, le mouvement planifié au pas de temps précédent est exécuté.

Cette approche permet donc d'être réactive aux diverses évolutions de l'environnement tout en étant capable de sortir d'impasses non détectées a priori.

Elle reste sujette à des minima locaux, mais y est néanmoins bien plus robuste que les approches citées précédemment.

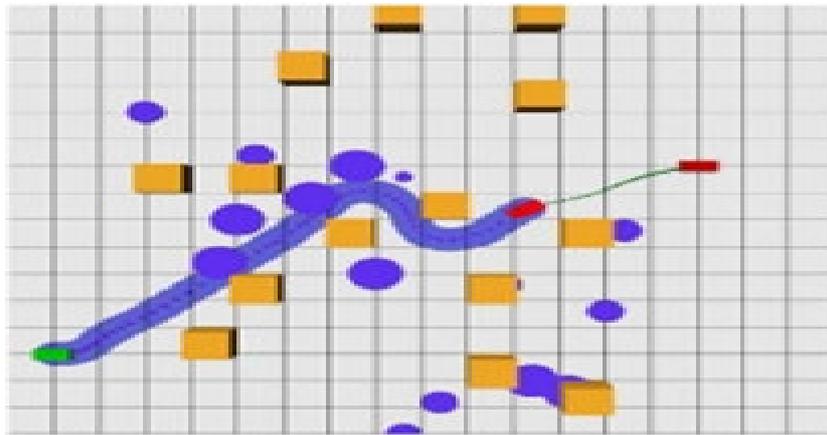


Figure III.17 Planification de mouvement partiel

### III.3.9 Défaut de convergence vers le but

Toutes les approches réactives citées ci-dessus disposent d'une complexité suffisamment faible pour être employées en temps réel au cours de la navigation. Cette caractéristique n'assure cependant en aucun cas la sécurité du système robotique (il n'est pour la plupart de ces approches ni garanti qu'il sera capable de s'arrêter s'il rencontre un obstacle et encore moins qu'il sera capable d'éviter un obstacle hostile se dirigeant vers lui). La capacité à raisonner sur le futur tend à favoriser une navigation sûre, c'est donc l'une des caractéristiques que nous souhaiterions intégrer au développement d'une nouvelle technique de navigation.

Malgré ces avantages par rapport aux approches délibératives, aucune de ces méthodes n'est capable d'assurer la convergence vers le but. Nous présentons alors dans la section suivante une méthode de déformation de mouvement essayant à la fois de s'assurer que celui-ci sera bien atteint, tout en évitant les obstacles au cours de l'exécution du mouvement.

### III.4 Méthode de déformation de mouvement

Entre approches délibératives calculant un mouvement complet jusqu'au but mais dont la complexité est trop élevée pour être utilisées en environnement dynamique, et approches réactives utilisables en temps réel mais dont la convergence vers le but est compromise, se trouve une large gamme de méthodes.



**Figure III.18** Déformation de chemin : le chemin suivi par un robot mobile est modifié itérativement au cours du mouvement afin de s'écarter de la position d'un obstacle s'en approchant

Afin de palier les inconvénients de ces deux types d'approches, Quinlan et Khatib ont proposé en 1993 une approche basée sur la déformation de mouvement [149]. Son principe est le suivant : Un chemin complet jusqu'au but est calculé à priori et fourni au système robotique. Au cours de l'exécution, la partie du mouvement restant à être exécutée est déformée continuellement en réponse aux informations sur l'environnement récupérées par les capteurs (Figure III.18). Le système peut ainsi modifier son parcours en fonction du déplacement d'obstacles ou de l'imprécision de sa connaissance de l'environnement. La déformation résulte en général de deux types

de contraintes : des contraintes externes dues à la proximité des obstacles, et des contraintes internes destinées à maintenir la faisabilité et la connectivité du mouvement. Tant que la connectivité du chemin est maintenue, la convergence vers le but est assurée.

La déformation de chemin souffre néanmoins d'une forte contrainte : les déformations sont limitées à des homotopies (déformations continues) du chemin de départ, pouvant ainsi conduire à des comportements inappropriés.

### **III.5 Conclusion**

Nous avons présenté dans cet état de l'art sur la navigation autonome d'une part des approches dites délibératives destinées à planifier un mouvement complet entre deux configurations et d'autre part des approches dites réactives qui consistent à calculer un nouveau mouvement à suivre à chaque pas de temps afin de pouvoir s'adapter au mouvement des obstacles mobiles ou inattendus.

Nos travaux se sont donc concentrés sur la résolution du problème de navigation d'un robot mobile dans un environnement inconnu. Nous proposons dans le chapitre suivant une approche par logique floue afin d'en apporter une solution.

## **Chapitre IV :**

### **NAVIGATION AUTONOME SOUS CONTRÔLE FLOU**

#### **IV.1 Introduction**

Contrôler un système complexe, fortement non-linéaire ou difficile à modéliser présente une tâche très délicate, ainsi les performances désirées deviennent de plus en plus sévères, c'est pour ces raisons qu'apparaissent de nouvelles méthodes de contrôle plus sophistiquées. L'apparition du formalisme de la logique floue [37] a donné naissance à une approche très efficace où les lois de commande conventionnelles sont remplacées par une série de règles linguistiques déterminant la stratégie et l'objectif de contrôle. Cette approche exige la disponibilité de l'expertise humaine ce qui influence négativement sur les performances des systèmes flous conçus. Ce problème a conduit au développement des systèmes hybrides permettant de réunir les capacités d'optimisation des algorithmes génétiques et celles de décision des systèmes flous [30-36]. L'utilisation conjointe de ces approches permet de réaliser un système de contrôle flou très efficace et robuste bien adapté à la complexité des systèmes réels (robots mobiles). les systèmes de navigation basés sur les comportements (*Behavior based navigation systems*) [150] ont été proposées comme une alternative à la stratégie la plus traditionnelle de commande qui est basée sur la construction d'une représentation approximative de l'environnement de navigation des robots mobiles, l'information acquise conduit à une étape de raisonnement puis agir sur le monde externe. L'idée principale de la navigation d'un robot mobile basée sur les comportements est d'identifier des réponses différentes (comportements) selon les données sensorielles. Une variété de systèmes de contrôle basés sur le comportement sont inspirés par le succès de Brooks avec son architecture [150], qui est connue par l'architecture de *subsumption*. Grace à ces propriétés intéressantes, plusieurs travaux ont utilisés la

logique floue comme approche de commande pour la navigation d'un robot mobile et pour la représentation des comportements [93, 96].

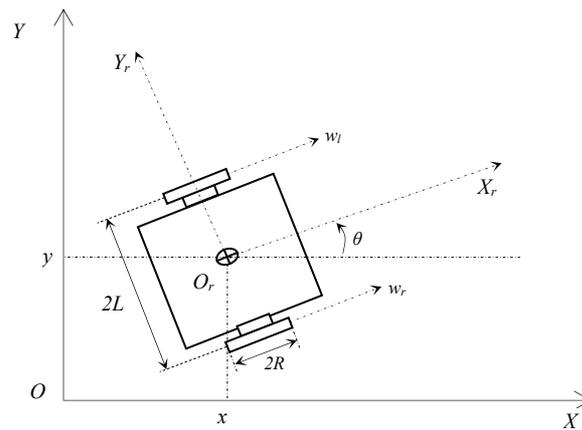
## IV.2 Description de l'architecture de navigation autonome proposée

Nous proposons dans cette étude une architecture de navigation basée sur un certain nombre de comportements élémentaires tels que: *la convergence vers un but, suivi de murs, évitement d'obstacles, poursuite de trajectoire,...*etc. Nous avons adopté un raisonnement flou pour élaborer les différents comportements ainsi que leur coordination. Le choix de la logique floue est motivé par le fait qu'elle permet d'utiliser des règles et des concepts linguistiques pour la conception des comportements, tout en permettant une fusion efficace au niveau des actions. De plus les autres techniques comme les réseaux de neurones et les champs de potentiels ne prennent pas en compte les incertitudes sur les données issues des capteurs.

Avant d'entamer la description des différents modules de notre architecture, nous allons d'abord donner une description de la mission assignée pour le robot et la plateforme du robot mobile prise en compte dans ce travail.

### IV.2.1 Modèle du robot mobile

Le robot mobile utilisé dans ce projet est de type différentiel, il possède deux roues motrices à braquage différentiel et une roue folle assurant la stabilité du robot. Pour modéliser le déplacement du robot mobile en tenant compte de ces contraintes, un système de repères convenable doit être judicieusement choisi. On commence par définir un repère fixe  $(X O Y)$  dans lequel le centre du robot a pour coordonnées  $x$  et  $y$  et on définit aussi un repère  $(X_r O_r Y_r)$  lié au robot, son centre est fixé au centre du robot. L'axe  $X_r$  est défini par l'axe (Avant/Arrière) du robot et l'axe  $Y_r$  est en quadrature directe par rapport à l'axe  $X_r$ .



**Figure IV.1** La configuration cartésienne du robot

Le robot est un objet rigide se déplaçant sur un plan horizontal avec deux degrés de liberté : le premier correspond à un mouvement de déplacement à l'avant ou à l'arrière, tandis que le deuxième est celui de la rotation autour d'un axe vertical. La position du robot est définie par le vecteur  $[x \ y \ \theta]^T$ , avec  $x$  et  $y$  sont les coordonnées cartésiennes du centre du robot et  $\theta$  est l'angle d'orientation du robot par rapport au repère fixe.

Le modèle cinématique du robot utilisé s'exprime dans le repère absolu  $(X \ O \ Y)$  par le système d'équation suivant:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (\text{IV.1})$$

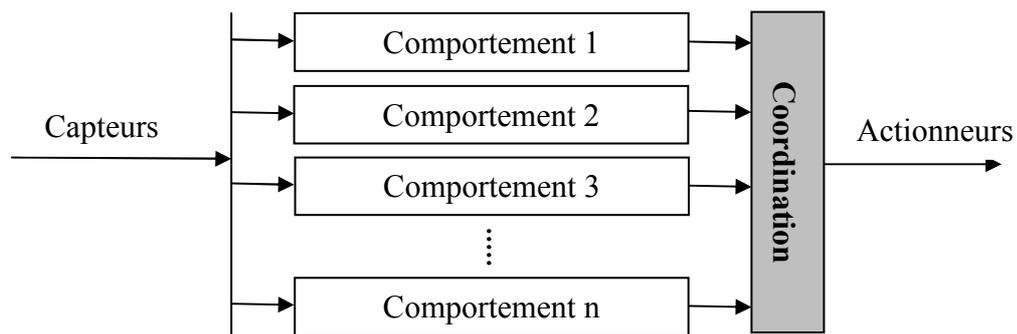
$$v = \frac{R}{2} (w_r + w_l) \quad (\text{IV.2})$$

$$w = \frac{R}{L} (w_r - w_l) \quad (\text{IV.3})$$

Avec  $v$  est la vitesse linéaire du centre du robot,  $w$  la vitesse de rotation du robot autour de son centre et  $w_r, w_l$  sont les vitesses de rotation des roues droite et gauche.

### IV.2.2 Structure de l'architecture de contrôle proposée

Le principe de base d'un système de navigation basé sur les comportements est de subdiviser la tâche de navigation globale en un ensemble de comportements élémentaires d'action (*comportement 1, comportement 2, ..., comportement n*); simple à concevoir et à gérer. Cette architecture peut régler le problème de conflit des actions dans les approches traditionnelles. Ces comportements sont indispensables à l'exécution des sous-tâches spécifiques pour le robot mobile, qui peuvent être par exemple (éviter les obstacles, converger vers le but, suivre un mur,...). Dans l'architecture *subsumption* proposée par Brooks [150], présentée sur la figure IV.2, les comportements sont arrangés dans des niveaux de priorité. En cas de déclenchement, le comportement de plus haut niveau supprime tous les comportements de niveau inférieur.



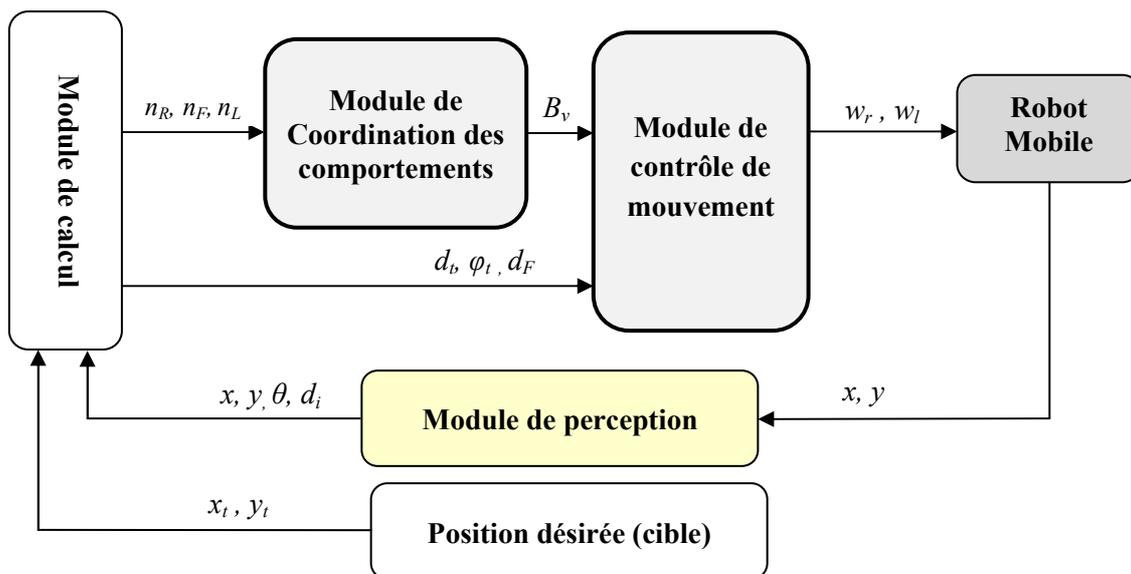
**Figure IV.2** Architecture basée sur les comportements

Arkin [151] a décrit l'utilisation de comportements réactifs appelés "*motorschemas*". Dans cette architecture, la méthode de champ de potentiel est utilisée pour définir la sortie de chaque bloc. Puis, toutes les sorties calculées sont combinées par une somme pondérée.

Rosenblatt [152] a présenté l'architecture *DAMN* (*Distributed Architecture for Mobile Navigation*). Elle consiste à utiliser une architecture distribuée multi-comportements (agents), chaque comportement représente une fonction bien particulière avec la caractéristique que ces agents ne peuvent pas discuter directement

entre eux mais doivent passer par une sorte d'ordonnanceur qui gère leurs communications et leurs états.

L'architecture subsumption est très efficace puisque le système est de nature modulaire, ce qui simplifie à la fois la résolution de problème de planification de trajectoire du robot mobile, et offre la possibilité de supprimer ou d'ajouter de nouveaux comportements au système sans provoquer d'augmentation importante de la complexité. Les sorties proposées de chaque comportement actif simultanément sont ensuite rassemblés selon une règle de coordination des actions inférées. La tâche de contrôle se réduit alors à un couplage des entrées sensorielles avec les actionneurs (c-à-d trouver un lien entre la perception et l'action) en utilisant des blocs de comportements sous formes des contrôleurs réactifs pour le robot mobile. Chaque comportement peut avoir des entrées provenant des capteurs du robot (caméra, ultrasons, infrarouges, ...) et/ou d'autres comportements dans le système, et envoyer des sorties aux actionneurs du robot (moteurs, roues, ...) et/ou à d'autres comportements. D'autres auteurs ont utilisés la logique floue pour la représentation et la coordination des comportements [93-96]. La figure IV.3 présente la structure du système de contrôle basé sur les comportements flous proposé dans cette thèse pour la navigation d'un robot mobile dans un environnement inconnu.



**Figure IV.3** Architecture de contrôle proposée

Après une description générale du principe de contrôle basé sur les comportements, les différents blocs de l'architecture proposée sont détaillés individuellement. L'approche comportementale est composée principalement d'un module de perception et deux contrôleurs flous accomplissant deux tâches élémentaires : la coordination entre les différents comportements réactifs (module de décision) et le contrôle de mouvement du robot mobile (module d'action).

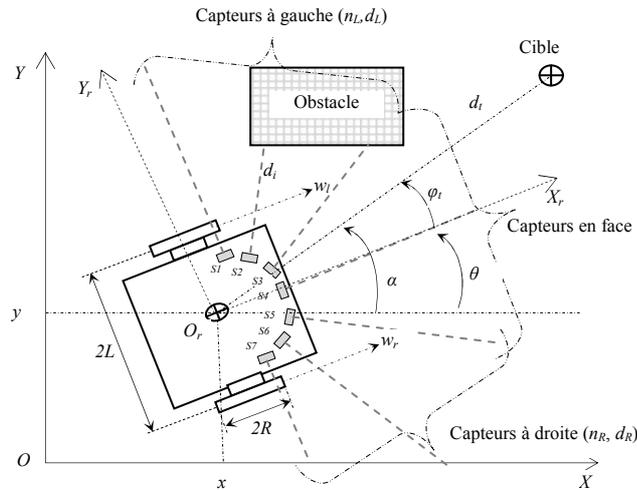
#### IV.2.2.1 Module de perception

La notion de perception en robotique mobile est relative à la capacité d'un système à recueillir, traiter et mettre en forme des informations utiles au robot pour agir dans le monde qui l'entoure. Il est aussi nécessaire que le robot dispose de nombreux capteurs mesurant aussi bien son état interne que l'environnement dans lequel il évolue pour extraire les informations utiles à l'accomplissement de sa tâche. Le module de perception a pour but d'extraire du modèle local les informations nécessaires au robot :

- pour détecter les obstacles.
- pour calculer un chemin permettant de rejoindre le but en évitant les obstacles.

Dans cette thèse, on considère le cas où l'environnement du robot est totalement inconnu ; un procédé d'acquisition est donc nécessaire pour acquérir les informations sur cet environnement. A cet effet, sept capteurs ultrasoniques ont été monté sur le robot mobile (Figure IV.4) pour mesurer les distances  $d_i$  entre le robot et les obstacles existants dans son espace de travail. Ces capteurs ultrasoniques sont regroupés en trois groupes pour détecter des obstacles vers la gauche (capteur  $i = 1, 2, 3$ ), vers l'avant (capteur  $i = 3, 4, 5$ ), et vers la droite (capteur  $i = 5, 6, 7$ ).

Dans l'approche proposée, deux types de variables sont nécessaires :  $n_L$ ,  $n_R$  et  $n_F$  qui sont les nombres de capteurs ultrasoniques qui détectent un obstacle et appartiennent au même groupe (pour l'exemple représenté sur la Figure IV.4 :  $n_L = 2$ ,  $n_R = 0$ , et  $n_F = 1$ ). Le deuxième type de variables est les distances entre le robot et les obstacles, ces variables sont définis comme suit :



**Figure IV.4** Les variables utilisées dans la méthode proposée

$$d_L = Mn\{di\} \quad i = 1, \dots, 3 \quad (IV.4)$$

$$d_F = Mn\{di\} \quad i = 3, \dots, 5 \quad (IV.5)$$

$$d_R = Mn\{di\} \quad i = 5, \dots, 7 \quad (IV.6)$$

Les valeurs de  $d_L$ ,  $d_F$  et  $d_R$  expriment les distances minimales entre le robot et les obstacles vers la gauche, vers l'avant et vers la droite, respectivement. Les variables  $\phi_t$  et  $d_t$  représentent respectivement l'angle d'orientation nécessaire pour diriger le robot vers la cible et la distance entre le robot et la cible. Elles sont données par les équations suivantes :

$$\phi_t = \alpha - \theta \quad (IV.7)$$

$$d_t = \sqrt{(x_t - x)^2 + (y_t - y)^2} \quad (IV.8)$$

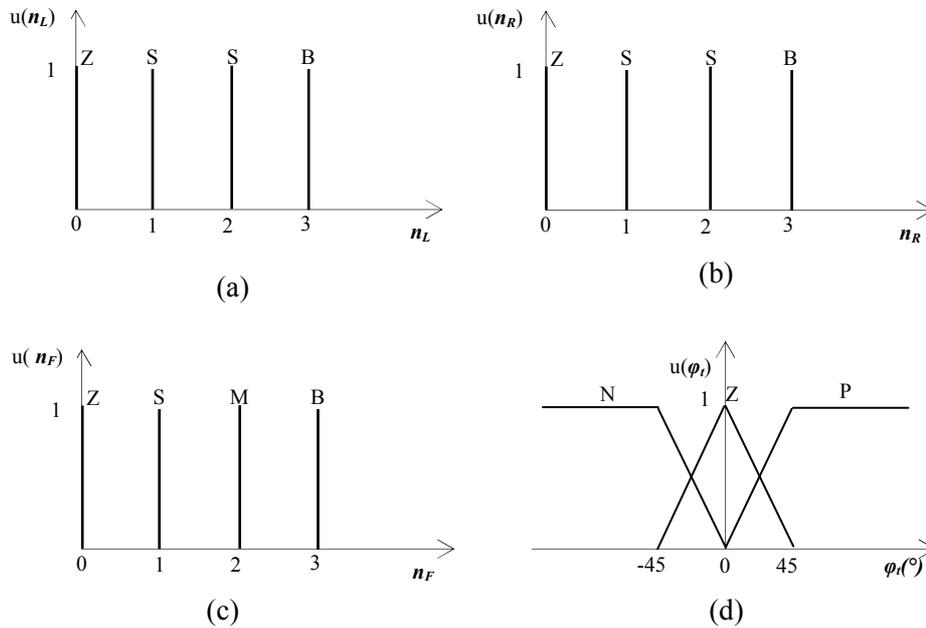
Où  $\alpha$  est l'angle de la cible,  $\theta$  est l'angle d'orientation du robot et  $(x_t, y_t)$  sont les coordonnées de la cible.

#### IV.2.2.2 Module de décision

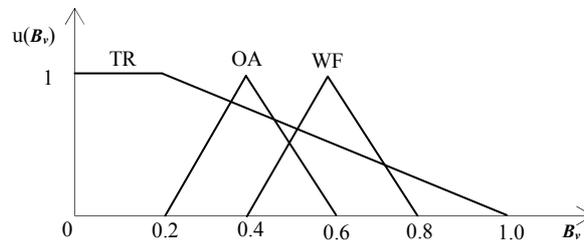
Afin d'atteindre une cible spécifique dans un environnement complexe, le robot mobile a besoin au moins des comportements réactifs suivants: navigation vers la cible (TR), évitement d'obstacles (OA) et suivi de murs (WF).

Dans la stratégie de commande proposée, ces trois comportements réactifs sont formulés par les ensembles flous dans lesquels les règles floues pour tous les comportements sont intégrées dans une seule base de règle. De cette manière, la coordination entre les différents comportements peut être facilement exécutée en utilisant seulement les données acquises par les capteurs ultrasoniques du robot. La Figure IV.3 montre le diagramme de l'architecture de contrôle proposée qui est établie sur deux systèmes flous, un pour la coordination entre les différents comportements réactifs et l'autre pour le contrôle de mouvement du robot mobile. Le comportement normal dans la navigation autonome d'un robot mobile est de guider le robot vers la cible. Sur son chemin immédiat, le robot essaye de naviguer vers la cible et quand il s'approche ou rencontre des obstacles ou des murs, le comportement concerné est exécuté avec un degré convenable. Une fois le robot est en dehors de la zone de collision, il continue sa tâche de navigation vers la cible en suivant le chemin le plus court.

Une question clé sur la commande comportementale c'est comment coordonner efficacement les priorités entre les différents comportements réactifs pour réaliser de bonnes navigations ? Pour atteindre cet objectif, dans la stratégie de commande proposée, les fonctions d'appartenance de tous les comportements réactifs sont intégrés dans un seul univers de discours normalisé. Le contrôleur de coordination des comportements flous proposé a une sortie unique qui est la valeur normale du comportement  $B_V$  et quatre variables d'entrée qui sont respectivement les nombres de capteurs ultrasoniques qui détectent un obstacle vers la gauche  $n_L$ , vers la droite  $n_R$ , vers l'avant  $n_F$  et l'angle d'orientation  $\phi t$  (voir Figure IV.4). Leurs fonctions d'appartenance sont montrées sur les Figures IV.5 et IV.6. Les règles floues utilisées dans la conception du contrôleur proposé sont générées automatiquement par un algorithme génétique (voir section IV.2.3).



**Figure IV.5** Les fonctions d'appartenance du coordinateur des comportements: (a, b, c) Les variables d'entrée  $n_L$ ,  $n_R$ ,  $n_F$  avec : Z est Zero, S est Smal, M est Medium et B est Big. (d) La variable d'entrée  $\phi_t$  avec: N est Negative, Z est Zero et P est Positive

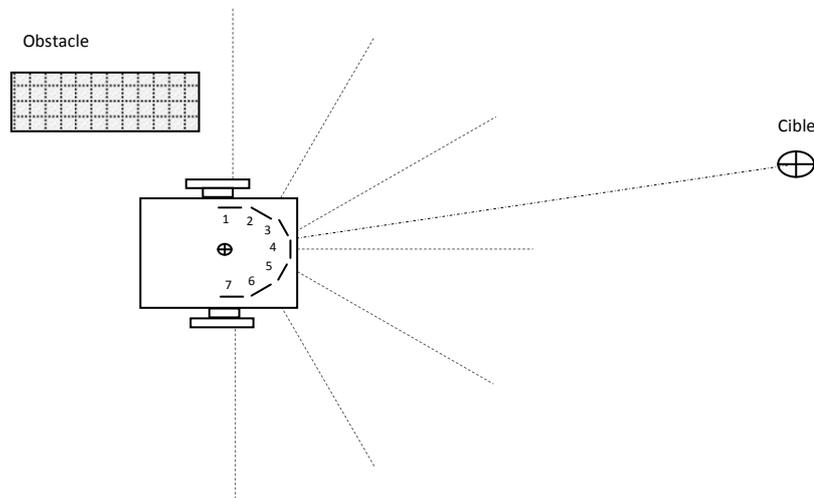


**Figure IV.6** Les fonctions d'appartenance du coordinateur des comportements: La variable de sortie  $B_v$  avec: TR est Target Reaching, OA est Obstacle Avoiding et WF est Wall Following

### a) Comportement de navigation vers la cible :

Le robot se déplace dans l'espace de recherche depuis sa position initiale vers la cible. Le but de ce comportement est d'atteindre une cible bien précise. La définition du point à atteindre est effectuée par l'intermédiaire de deux variables d'entrées; la distance entre le robot et la cible  $d_t$  et l'angle entre l'orientation actuelle du robot et

celle de la cible  $\varphi_t$  (Figure IV.4). Un exemple d'une situation de navigation vers la cible est illustré sur la Figure IV.7.



**Figure IV.7** Exemple d'une situation de navigation vers la cible

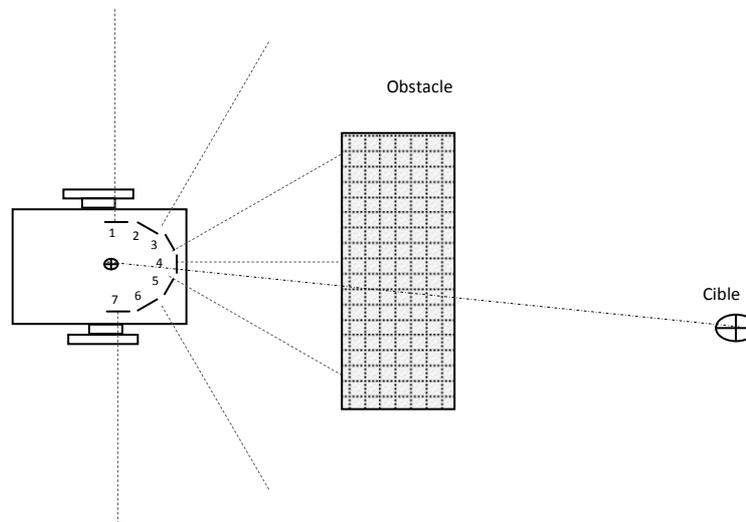
Le module de calcul compare les coordonnées réelles du robot avec les coordonnées de la cible afin de calculer la distance robot-cible et l'angle désiré  $d_t$  et  $\varphi_t$  pour la rejoindre. Le contrôleur de mouvement (module d'action) utilise ces deux variables ( $d_t$  et  $\varphi_t$ ) pour générer les deux signaux de commande des roues gauche et droite ( $w_l$  et  $w_r$ ).

Le comportement présenté dans cette section permet de réaliser la navigation autonome d'un robot mobile vers un but si le robot ne reçoit aucune information de ses capteurs de perception, où l'environnement est très peu encombré. Une telle condition est très restrictive vu la nature de l'environnement dans lequel le robot est amené à réaliser ses missions. Dans le cas de présence d'obstacles (statiques ou dynamiques) qui empêchent le mouvement du robot mobile vers la cible, le robot doit posséder d'autres capacités lui permettant d'éviter la collision avec ces obstacles.

### **b) Comportement d'évitement d'obstacles :**

L'évitement d'obstacles est un comportement de base devant être présent dans tous les systèmes de navigation autonome des robots mobiles. En effet la caractéristique de

mobilité impose que le robot doit pouvoir agir dans son environnement, c'est pourquoi tous les robots mobiles sont quasiment équipés de capteurs permettant de détecter les obstacles dont l'objectif principal est de garantir le maximum de sécurité au cours de la navigation. La Figure IV.8 illustre un exemple d'une situation d'évitement d'obstacles.



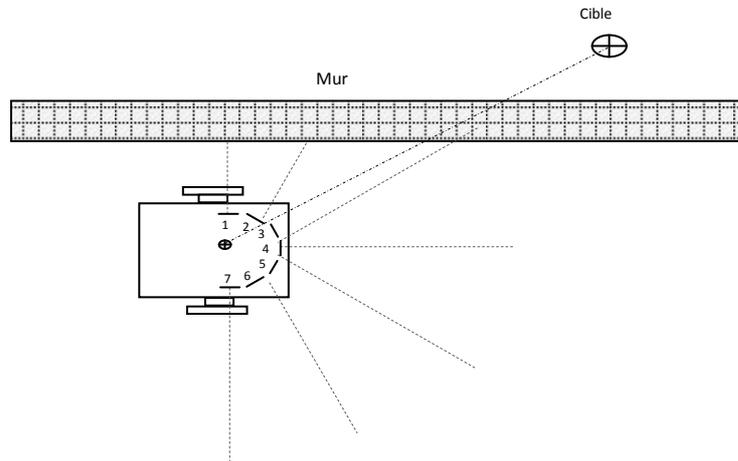
**Figure IV.8** Exemple d'une situation d'évitement d'obstacles

Dans cette application, l'environnement de navigation du robot mobile peut contenir des obstacles, le robot doit donc avoir une capacité réactive et efficace d'évitement de collisions. Il a besoin d'acquérir des informations sur l'environnement pour être capable de naviguer correctement suivant la forme de ces obstacles. Pour cela, on suppose que le robot mobile est équipé de sept capteurs de type ultrasons pour la détection des obstacles dans les trois directions (en avant, à droite et à gauche). Les mesures de ces capteurs sont utilisées pour générer l'action adéquate d'évitement des obstacles. Ces capteurs sont regroupés en trois groupes et pour chaque groupe, on utilise trois capteurs afin de générer l'action appropriée de mouvement. La figure IV.4 représente les positions des capteurs sur le robot mobile ainsi que l'arrangement sous forme de groupes.

### c) Comportement de suivi de murs :

L'objectif de ce comportement est de maintenir le robot à une distance de sécurité

loin des murs détectés sur sa droite ou sa gauche tout en minimisant les oscillations au cours de la navigation. Selon les informations fournies par les trois groupes de capteurs ultrasoniques le robot peut distinguer entre les différents comportements possibles. La Figure IV.9 illustre un exemple d'une situation de suivi de murs.

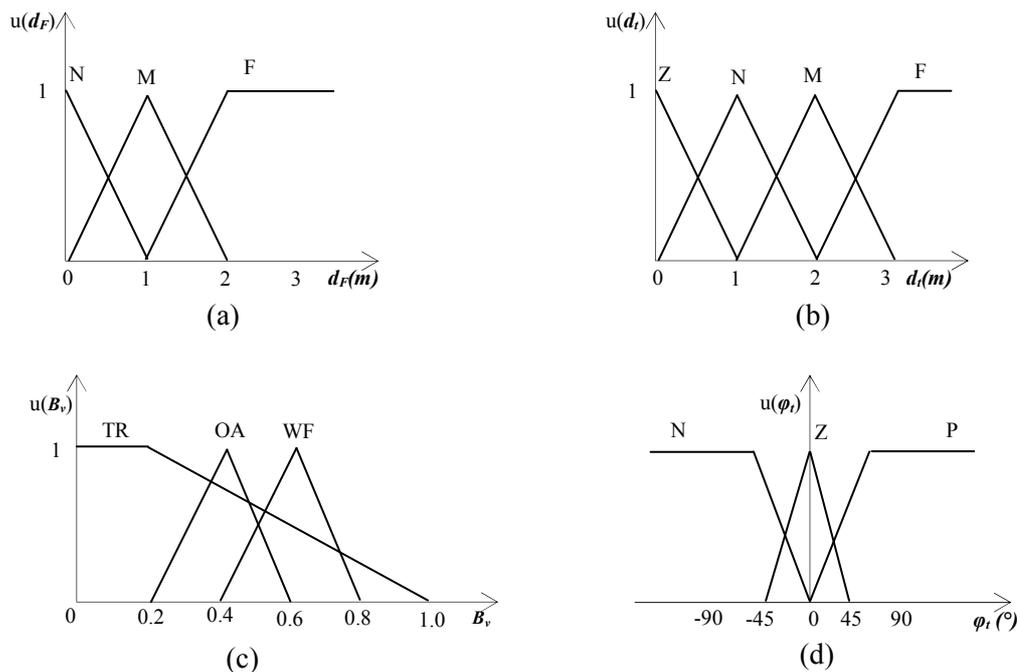


**Figure IV.9** Exemple d'une situation de suivi de murs

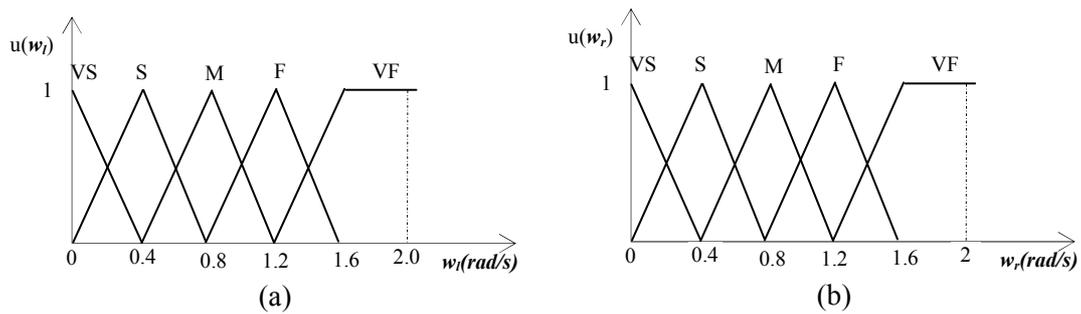
### IV.2.2.3 Module d'action

Il s'agit alors de veiller à réaliser le mouvement ou le déplacement et suivre les consignes de vitesses reçues en appliquant le type de commande adapté. Un contrôleur à base de logique floue est utilisé pour réaliser le contrôle de mouvement du robot mobile dans un environnement inconnu encombré d'obstacles. Comme décrit dans la section IV.2.1, le robot mobile possède deux roues entraînées indépendamment et sept capteurs ultrasoniques utilisés pour détecter des obstacles en avant, vers la droite et vers la gauche du robot. Selon l'information de distance fournie par ces capteurs et les positions actuelle et cible du robot, le module de calcul calcule les valeurs de  $n_L$ ,  $n_F$ ,  $n_R$  et  $\varphi_t$  (voir la section IV.2.2.1) et les transmet au coordinateur des comportements qui est responsable de décider quel comportement doit être exécuté. Parallèlement, le module de calcul calcule les valeurs de  $d_t$ ,  $\varphi_t$  et  $d_F$  et les transmet au contrôleur flou de mouvement. Les variables  $d_t$  et  $\varphi_t$  représentent la distance et l'angle entre la position actuelle du robot et la cible. La variable  $d_F$  représente la distance minimale entre le

robot et les obstacles présents en sa direction avant; elle est exploitée par le contrôleur de mouvement pour maximiser la vitesse de navigation du robot. Donc, le contrôleur de mouvement reçoit à ses entrées la valeur normalisée du comportement  $B_v$ , la valeur de la distance  $d_t$ , la valeur de l'angle  $\varphi_t$  et de la distance minimale des obstacles présents en avant du robot. Les sorties du contrôleur de mouvement sont les deux signaux de commande des vitesses des roues gauche et droite du robot  $w_l$  et  $w_r$  respectivement. Le contrôleur proposé est basé sur la règle d'implication de MAMDANI (max-min), la conjonction floue (min) et par la méthode de centre de gravité pour la défuzzification. Les fonctions d'appartenances utilisées pour fuzzifier les variables d'entrée et de sortie sont de forme triangulaire et trapézoïdale, elles sont illustrées sur les Figures IV.10 et IV.11 respectivement.



**Figure IV.10** Les fonctions d'appartenance du contrôleur de mouvement: (a,b) Les variables d'entrée  $d_F$ ,  $d_t$  avec : Z est Zero, N est Near, M est Medium et F est Far. (c) La variable d'entrée  $B_v$  avec: TR est Target Reaching, OA est Obstacle Avoiding et WF est Wall Following. (d) La variable d'entrée  $\varphi_t$  avec: N est Negative, Z est Zero et P est positive



**Figure IV.11** Les fonctions d'appartenance du contrôleur de mouvement: (a,b) Les variables de sortie  $w_l$ ,  $w_r$  avec : VS est Very Small, S est Small, M est Medium, F est Fast et VF est Very Fast

Générer les règles d'inférence est souvent l'étape la plus difficile dans le processus de conception d'un contrôleur flou. Il exige généralement la connaissance approfondie de la dynamique du système à contrôler. Dans cette thèse, les règles floues sont générées automatiquement par un algorithme génétique.

### IV.2.3 Génération automatique des règles floues par les algorithmes génétiques

Bien que les contrôleurs à logique floue aient été appliqués avec succès sur plusieurs procédés industriels complexes, leur conception reste cependant une tâche très difficile. L'approche traditionnelle pour la conception floue basée sur les connaissances acquises par des opérateurs experts est laborieuse et consomme beaucoup de temps, en plus, cette approche présente d'autres inconvénients tels que :

- les opérateurs ne peuvent pas facilement transformer leurs connaissances et expériences en une forme algorithmique ou base de règles nécessaires pour la conversion en une stratégie de contrôle automatique.
- le domaine d'expertise n'est pas toujours disponible.

Suite à ces inconvénients, des recherches approfondies ont été effectuées dans le but d'élaborer des méthodes systématiques et optimales pour la conception des contrôleurs flous, ces recherches ont conduit au développement de deux nouvelles approches :

### **IV.2.3.1 Approche connexionniste :**

Elle consiste à combiner les réseaux de neurones artificiels et les systèmes flous pour construire ce qu'on appelle les réseaux de neurones flous [26].

### **IV.2.3.2 Approche directe :**

Elle consiste à appliquer un algorithme d'optimisation pour la conception d'un système flou. A cause de la complexité de l'espace de recherche, les algorithmes génétiques sont utilisés dans la plupart des cas [31].

C'est cette dernière approche qui a été utilisée dans cette thèse pour la conception d'un système de contrôle flou robuste et efficace.

### **IV.2.3.3 Conception d'un contrôleur flou par les algorithmes génétiques :**

Le problème de conception d'un contrôleur flou (FLC) revient à la détermination des espaces de discours des variables d'entrées et de sortie, la définition des fonctions d'appartenance pour chaque terme linguistique et la dérivation des règles de contrôle [30]. Le choix d'une méthode de conception dépend étroitement de la nature du domaine de travail et de l'information disponible. Dû à la complexité de l'espace de recherche pour le problème de conception d'un contrôleur flou. Dû, entre autres, à la non dérivabilité et la discontinuité des fonctions impliquées, la meilleure alternative pour résoudre ce type de problème est l'utilisation des algorithmes génétiques.

L'incorporation de l'apprentissage génétique dans un processus de conception floue ajoute une dimension intelligente au contrôleur flou et permet de générer automatiquement ses paramètres optimaux (Figure IV.12). L'interaction des différents blocs d'un tel système de contrôle est illustrée sur l'organigramme de la Figure IV.13.

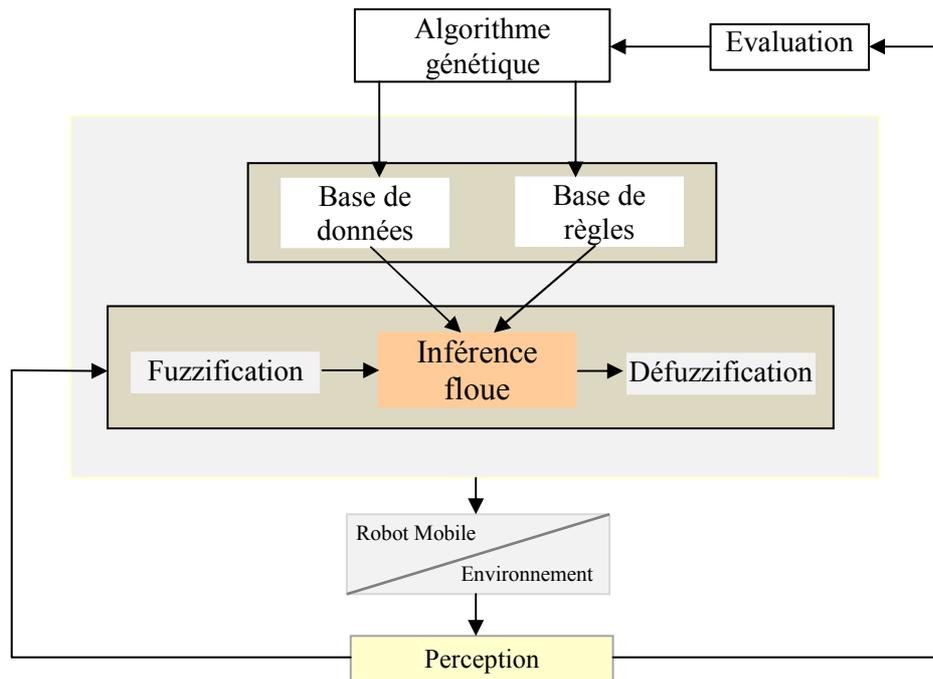


Figure IV.12 Structure du système de navigation à base de la logique floue hybride

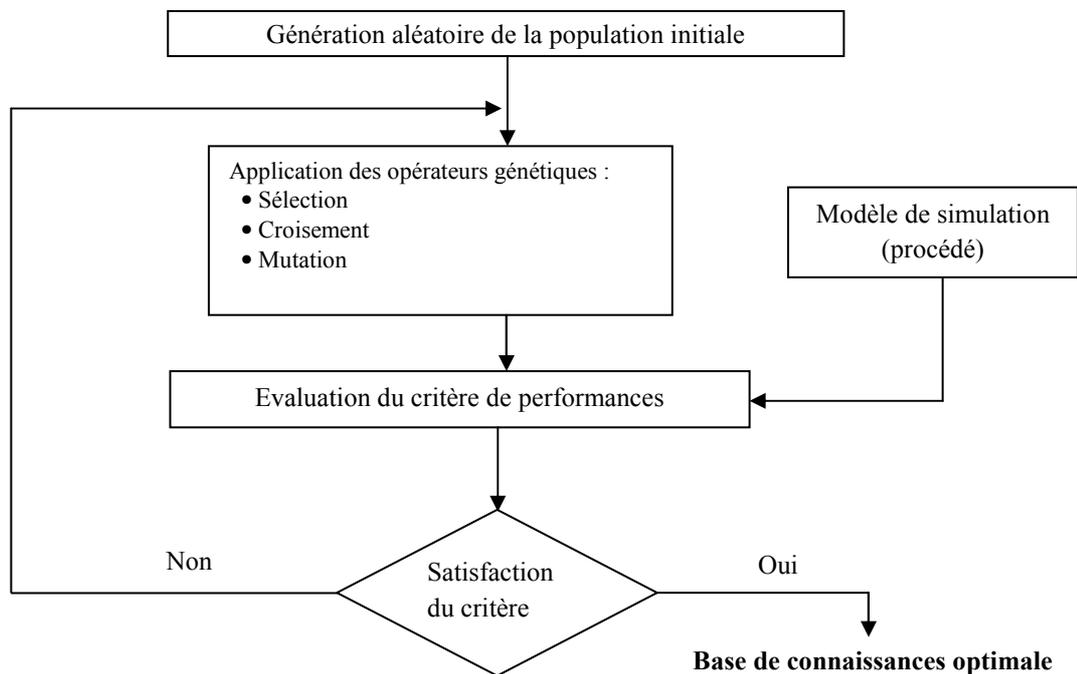


Figure IV.13 Etapes de conception d'un contrôleur flou

Concevoir un contrôleur flou consiste principalement à définir le ou les paramètres à optimiser, le type de codage des paramètres et la fonction d'objectif.

#### **IV.2.3.4 Définition des paramètres à optimiser :**

Il existe trois approches d'application des algorithmes génétiques pour la conception des contrôleurs flous. Dans la première, la base des règles (règles floues) est fixée, tandis que la base de données (définitions des fonctions d'appartenance) est optimisée par l'AG. Dans la seconde, la base de données est fixée, seulement la base des règles est optimisée par l'AG (détermination des règles floues). Dans la troisième approche toute la base de connaissances (base de données + base de règles) est optimisée par un algorithme génétique.

Les paramètres à optimiser dans cette application sont les bases des règles des deux contrôleurs flous qui contiennent  $72+108 = 180$  règles floues doivent être optimisées par l'algorithme génétique.

#### **IV.2.3.5 Codage des paramètres :**

Le codage utilisé est un codage binaire où plusieurs paramètres sont codés et juxtaposés les uns après les autres. Ces paramètres sont les 180 règles floues décrivant les bases des règles des deux contrôleurs flous. Chaque règle est codée sur 02 bits pour le module de décision et sur 03 bits pour le module d'action ce qui donne au total  $72 \times 2 + 108 \times 3 = 468$  bits. Le Tableau IV.1 représente l'idée du codage binaire des règles floues.

Module de décision	Ensemble Flou	TR		OA		TR		WF	
	Codage (2bits)	00		01		10		11	
Module d'action	Ensemble Flou	VS	VS	S	M	M	F	VF	VF
	Codage (3bits)	000	001	010	011	100	101	110	111

**Tableau IV.1:** Codage binaire des règles floues

#### IV.2.3.6 Fonction d'objectif :

Pour évaluer les performances de navigation d'un robot mobile, nous devons définir les critères essentiels qui montrent qu'une navigation soit meilleure qu'une autre.

Ces critères sont:

- Le nombre de collision que fait le robot avec les obstacles durant la navigation (**Nc**);
- La longueur de la trajectoire (**Lc**) ;
- Le temps nécessaire pour atteindre la cible (**Tc**);
- Le nombre de tour (**Nt**) effectués par le robot mobile en contournant les obstacles ;

Puisque les algorithmes génétiques sont par nature des procédures de recherche de maximum, il faut transformer le problème de minimisation en un problème de maximisation de  $f$  comme suit :

#### -Le nombre de collision

$$f_{Nc} = \frac{1}{1+Nc} \quad \text{Avec} \quad (f_{Nc} \in [0 \ 1]) \quad (\text{IV.9})$$

**-La longueur de la trajectoire**

$$f_{Lc} = \frac{1}{1+Lc} \quad \text{Avec} \quad (f_{Lc} \in [0 \ 1]) \quad (\text{IV.10})$$

**-Le nombre de tour**

$$f_{Nt} = \frac{1}{1+Nt} \quad \text{Avec} \quad (f_{Nt} \in [0 \ 1]) \quad (\text{IV.11})$$

**-Le temps nécessaire pour atteindre la cible**

$$f_{Tc} = \frac{1}{1+Tc} \quad \text{Avec} \quad (f_{Tc} \in [0 \ 1]) \quad (\text{IV.12})$$

**-Fonction d'objectif globale**

La fonction d'objectif globale du comportement du robot est la combinaison des quatre fonctions présentées précédemment :

$$f_{glob} = f_{Nc}(c_1 \cdot f_{Lc} + c_2 \cdot f_{Tc} + c_3 \cdot f_{Nt}) \quad (\text{IV.13})$$

Dans cette fonction, nous avons donné plus d'importance à la fonction  $f_{Nc}$ , car notre premier objectif est l'évitement d'obstacles. Les coefficients de pondération  $c_1$ ,  $c_2$  et  $c_3$  sont choisis empiriquement pour éviter les débordements.

**-Pénalisation des mauvais comportements**

Pour sortir des situations de blocage dont lesquelles peut tomber un robot pendant sa navigation, nous avons fixé un temps maximum de navigation et un nombre maximum de collision, la pénalisation sur la fonction d'objectif est faite comme suite :

**-Si** temps  $\geq$  temps\_max **Alors**  $f_{glob} = \mu_1 \cdot f_{glob}$

**-Si** N\_collisions  $\geq$  N\_collisions\_max **Alors**  $f_{glob} = \mu_2 \cdot f_{glob}$

**-Si** Dist tend vers 0 et temps  $\geq$  temps\_max **Alors**  $f_{glob} = \mu_3 \cdot f_{glob}$

Avec  $\mu_1$ ,  $\mu_2$  et  $\mu_3$  sont des coefficients de pénalisation, ils sont choisis empiriquement.

#### **IV.2.3.7 Exécution de l'algorithme d'optimisation:**

L'AG proposé utilise la sélection par roue de loterie avec remplacement, un opérateur de croisement en deux points et un opérateur de mutation standard. Les paramètres utilisés sont:

- la taille de la population est fixée à 100.
- le nombre maximal de génération est égal à 200.
- les probabilités de mutation et de croisement valent 0.02 et 0.95 respectivement.
- les coefficients  $c_1$ ,  $c_2$  et  $c_3$  valent 0.1, 0.05 et 0.001 respectivement.
- les coefficients  $\mu_1$ ,  $\mu_2$  et  $\mu_3$  valent 0.01, 0.3 et 0.001 respectivement.

Les Tableaux IV.2 et IV.3 présentent les bases des règles floues générées par l'algorithme génétique proposé.

Inputs				Output	Inputs				Output
$n_L$	$n_R$	$n_F$	$\varphi_t$	$B_v$	$n_L$	$n_R$	$n_F$	$\varphi_t$	$B_v$
Z	Z	Z	N	TR	S	Z	M	Z	OA
S	Z	Z	N	TR	B	Z	M	Z	WF
Z	S	Z	N	TR	Z	S	M	Z	OA
S	S	Z	N	TR	S	S	M	Z	OA
Z	Z	S	N	OA	B	S	M	Z	WF
S	Z	S	N	TR	Z	B	M	Z	WF
B	Z	S	N	TR	S	B	M	Z	WF
Z	S	S	N	OA	B	B	M	Z	OA
S	S	S	N	TR	S	S	B	Z	OA
B	S	S	N	TR	B	S	B	Z	OA
Z	B	S	N	WF	S	B	B	Z	OA
S	B	S	N	WF	B	B	B	Z	OA
S	Z	M	N	TR	Z	Z	Z	P	TR
B	Z	M	N	TR	S	Z	Z	P	TR
Z	S	M	N	OA	Z	S	Z	P	TR
S	S	M	N	OA	S	S	Z	P	TR
B	S	M	N	OA	Z	Z	S	P	OA
Z	B	M	N	WF	S	Z	S	P	OA
S	B	M	N	WF	B	Z	S	P	WF
B	B	M	N	OA	Z	S	S	P	TR
S	S	B	N	OA	S	S	S	P	TR
B	S	B	N	OA	B	S	S	P	WF
S	B	B	N	OA	Z	B	S	P	WF
B	B	B	N	OA	S	B	S	P	WF
Z	Z	Z	Z	TR	S	Z	M	P	TR
S	Z	Z	Z	TR	B	Z	M	P	WF
Z	S	Z	Z	TR	Z	S	M	P	TR
S	S	Z	Z	TR	S	S	M	P	OA
Z	Z	S	Z	TR	B	S	M	P	WF
S	Z	S	Z	TR	Z	B	M	P	TR
B	Z	S	Z	WF	S	B	M	P	TR
Z	S	S	Z	TR	B	B	M	P	OA
S	S	S	Z	TR	S	S	B	P	OA
B	S	S	Z	WF	B	S	B	P	OA
Z	B	S	Z	WF	S	B	B	P	OA
S	B	S	Z	WF	B	B	B	P	OA

Tableau IV.2 La base des règles floues du module de décision

Inputs				Outputs		Inputs				Outputs	
$d_f$	$d_t$	$\phi_t$	$B_v$	$w_l$	$w_r$	$d_f$	$d_t$	$\phi_t$	$B_v$	$w_l$	$w_r$
N	Z	N	TR	S	VS	N	M	Z	OA	M	M
M	Z	N	TR	M	VS	M	M	Z	OA	M	M
F	Z	N	TR	M	VS	F	M	Z	OA	F	F
N	N	N	TR	S	VS	N	F	Z	OA	S	S
M	N	N	TR	M	S	M	F	Z	OA	M	M
F	N	N	TR	F	S	F	F	Z	OA	VF	VF
N	M	N	TR	VS	VS	N	Z	P	OA	VS	S
M	M	N	TR	M	S	M	Z	P	OA	M	S
F	M	N	TR	F	M	F	Z	P	OA	M	S
N	F	N	TR	S	VS	N	N	P	OA	VS	S
M	F	N	TR	M	S	M	N	P	OA	VS	S
F	F	N	TR	F	M	F	N	P	OA	VS	S
N	Z	Z	TR	VS	VS	N	M	P	OA	VS	S
M	Z	Z	TR	VS	VS	M	M	P	OA	S	M
F	Z	Z	TR	VS	VS	F	M	P	OA	M	F
N	N	Z	TR	S	S	N	F	P	OA	VS	S
M	N	Z	TR	S	S	M	F	P	OA	S	M
F	N	Z	TR	S	S	F	F	P	OA	M	F
N	M	Z	TR	M	M	N	Z	N	WF	VS	VS
M	M	Z	TR	F	F	M	Z	N	WF	S	S
F	M	Z	TR	F	F	F	Z	N	WF	S	S
N	F	Z	TR	M	M	N	N	N	WF	VS	VS
M	F	Z	TR	F	F	M	N	N	WF	S	S
F	F	Z	TR	VF	VF	F	N	N	WF	M	M
N	Z	P	TR	VS	S	N	M	N	WF	S	S
M	Z	P	TR	VS	M	M	M	N	WF	M	M
F	Z	P	TR	VS	M	F	M	N	WF	F	F
N	N	P	TR	VS	S	N	F	N	WF	S	S
M	N	P	TR	S	M	M	F	N	WF	F	F
F	N	P	TR	S	F	F	F	N	WF	VF	VF
N	M	P	TR	VS	S	N	Z	Z	WF	VS	VS
M	M	P	TR	S	M	M	Z	Z	WF	S	S
F	M	P	TR	M	F	F	Z	Z	WF	S	S
N	F	P	TR	VS	S	N	N	Z	WF	S	S
M	F	P	TR	S	M	M	N	Z	WF	M	M
F	F	P	TR	M	F	F	N	Z	WF	M	M
N	Z	N	OA	S	VS	N	M	Z	WF	M	M
M	Z	N	OA	M	VS	M	M	Z	WF	F	F
F	Z	N	OA	M	VS	F	M	Z	WF	S	S
N	N	N	OA	S	VS	N	F	Z	WF	F	F
M	N	N	OA	M	S	M	F	Z	WF	VF	VF
F	N	N	OA	F	S	F	F	Z	WF	VS	VS
N	M	N	OA	S	VS	N	Z	P	WF	S	S
M	M	N	OA	M	S	M	Z	P	WF	S	S
F	M	N	OA	F	M	F	Z	P	WF	S	S
N	F	N	OA	S	VS	N	N	P	WF	S	S
M	F	N	OA	M	S	M	N	P	WF	S	S
F	F	N	OA	F	M	F	N	P	WF	M	M
N	Z	Z	OA	VS	VS	N	M	P	WF	S	S
M	Z	Z	OA	VS	VS	M	M	P	WF	M	M
F	Z	Z	OA	VS	VS	F	M	P	WF	F	F
N	N	Z	OA	S	S	N	F	P	WF	S	S
M	N	Z	OA	S	S	M	F	P	WF	F	F
F	N	Z	OA	S	S	F	F	P	WF	VF	VF

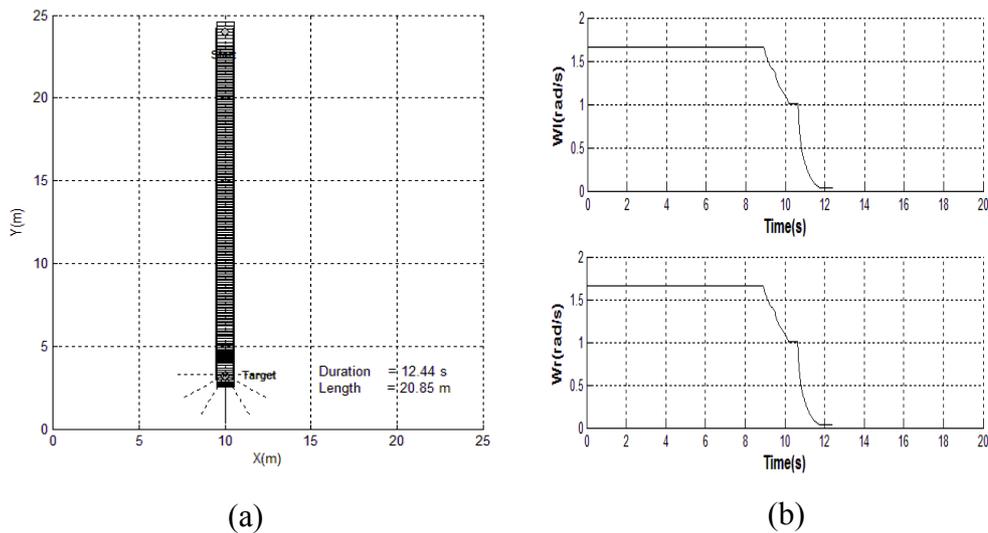
Tableau IV.3 La base des règles floues du module d'action

### IV.3 Résultats de simulation

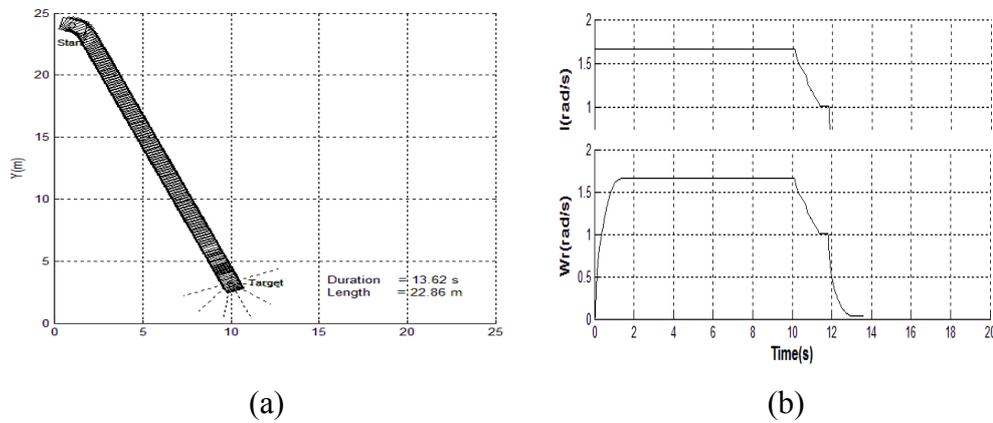
Pour montrer l'efficacité de l'approche proposée, une série de simulations a été réalisée en employant des environnements arbitrairement construits contenant des obstacles. La position de tous les obstacles est inconnue ; le robot ne connaît que les positions de départ et d'arrivée (cible). Toutes les simulations ont été faites dans l'environnement de Matlab/Simulink en utilisant le modèle cinématique du robot donné par l'Equation IV.1. La valeur numérique de la durée du pas de calcul est 0.01s dans toutes les simulations.

#### IV.3.1 Comportement de navigation vers une cible

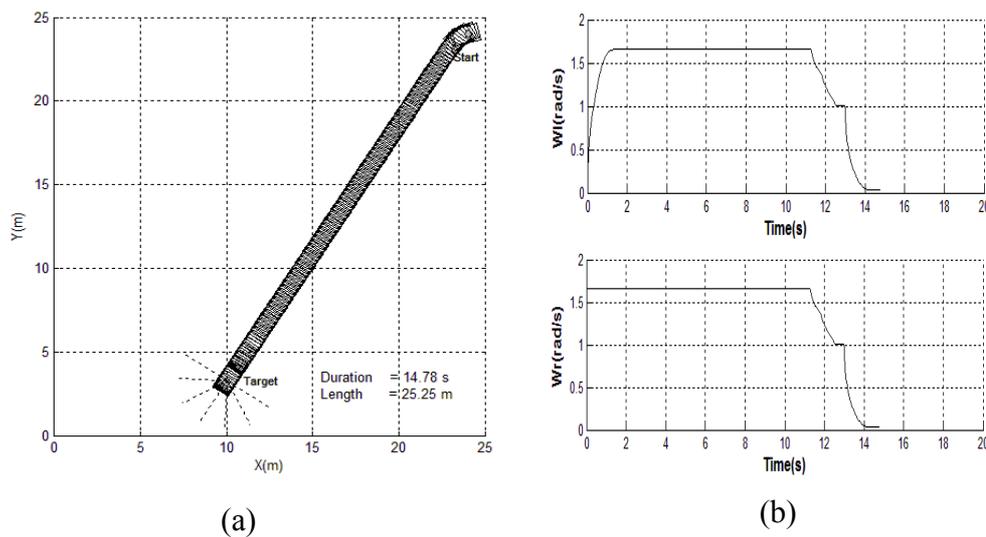
En absence d'obstacles dans l'espace de travail du robot, la tâche de navigation se réduit à une orientation et un déplacement direct vers la cible. Les Figures IV.14, IV.15 et IV.16 illustrent les trajectoires de navigation en choisissant différents points de départ et d'arrivée. Ces résultats montrent l'aptitude du contrôleur flou à générer les actions de commande les plus appropriées pour accomplir la tâche, ce qui prouve l'efficacité du contrôleur proposé.



**Figure IV.14** Navigation rectiligne vers la cible



**Figure IV.15** Rotation à droite et navigation rectiligne vers la cible



**Figure IV.16** Rotation à gauche et navigation rectiligne vers la cible

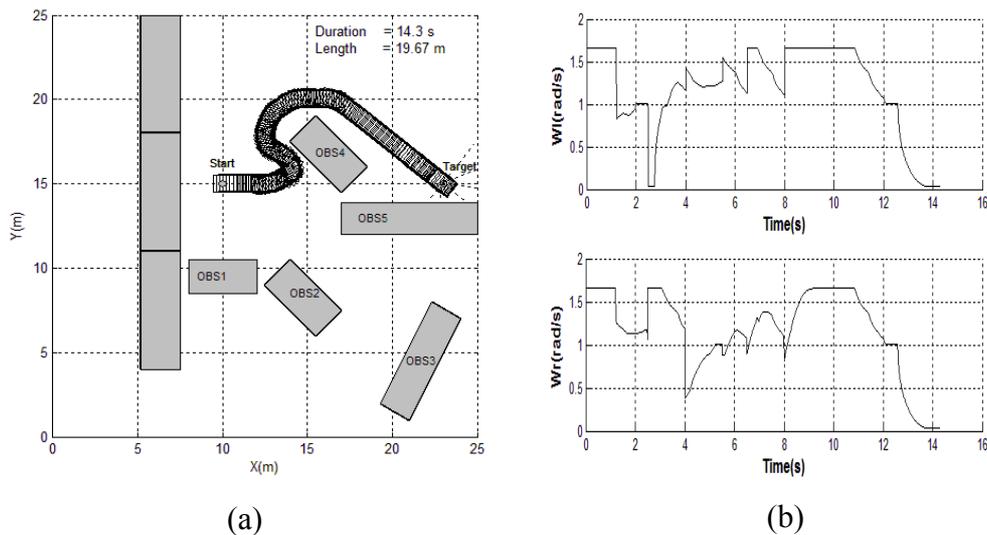
Les Figures IV.14(b), IV.15(b) et IV.16(b) présentent les signaux de commande générés par le contrôleur flou lors de la tâche de navigation du robot mobile vers la cible. Au départ, le robot se trouve loin de la cible, le contrôleur flou génère des vitesses différentes pour les roues gauche et droite afin de réaliser un braquage différentiel pour tourner le robot vers la cible. Une fois cette étape est terminée (le robot est en ligne avec la cible) le contrôleur flou génère des vitesses égales pour les

deux roues afin de permettre au robot de se déplacer directement vers la cible avec une vitesse maximale, puis cette vitesse se diminue en s'approchant de la cible.

### IV.3.2 Comportement d'évitement d'obstacles

Le comportement normal dans la navigation autonome de robot mobile est de guider le robot vers la cible. Sur son chemin immédiat, le robot essaye de naviguer vers la cible et quand il s'approche ou rencontre des obstacles, le comportement concerné est exécuté avec un degré convenable. Une fois le robot est en dehors de la zone de collision, il continue sa tâche en se dirigeant vers la cible et en suivant le chemin le plus court.

Les Figures IV.17, IV.18 et IV.19 montrent des exemples de navigation autonome du robot mobile en présence d'obstacles dans l'environnement. Au départ, le robot essaye de se déplacer en ligne droite vers la cible, mais lorsqu'il détecte la présence des obstacles dans son chemin, le comportement d'évitement d'obstacles est activé et le contrôleur flou génère des actions appropriées pour faire sortir le robot de la zone de collision en toute sécurité. Les Figures IV.17(b), IV.18(b) et IV.19(b) présentent les signaux de commande générés par le contrôleur proposé pour assurer l'évitement d'obstacles et la convergence vers la cible avec la vitesse maximale admissible.



**Figure IV.17** Evitement d'un seul obstacle puis navigation vers la cible

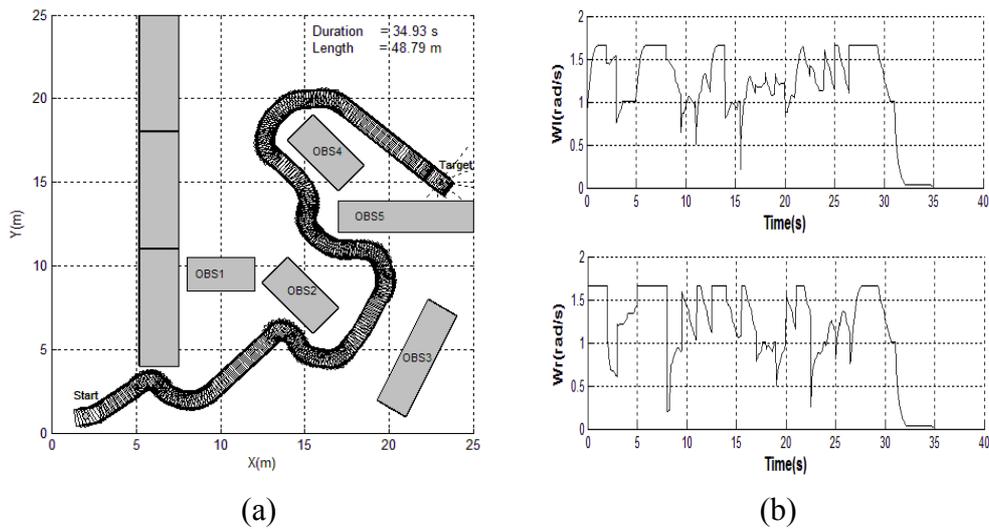


Figure IV.18 Evitement de plusieurs obstacles puis navigation vers la cible

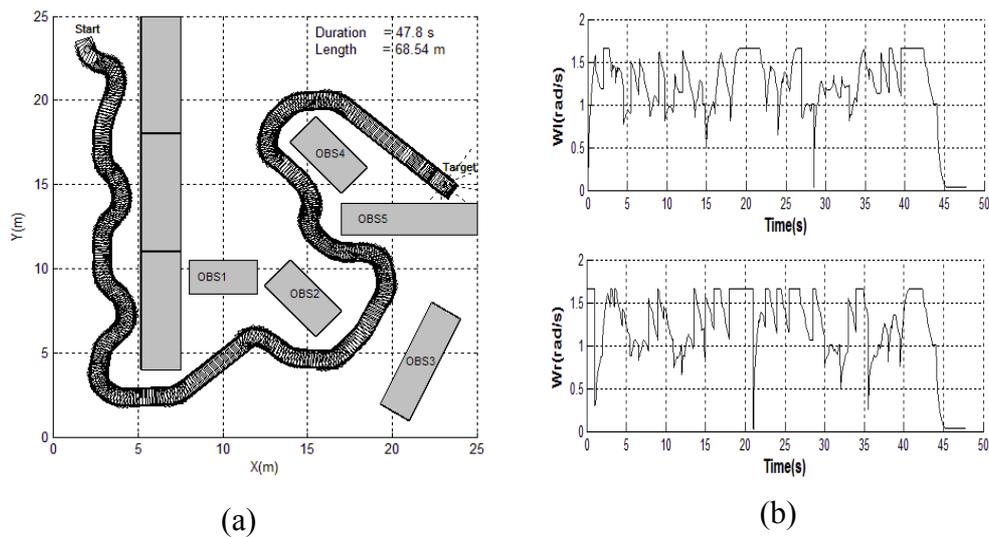


Figure IV.19 Evitement d'un mur et de plusieurs obstacles puis navigation vers la cible

### IV.3.3 Comportement de suivi de murs

L'objectif de ce comportement est de maintenir le robot à une distance de sécurité loin des murs détectés sur sa droite ou sa gauche tout en minimisant les oscillations au cours de la navigation. Les Figures IV.20 et IV.21 montrent des exemples de navigation autonome du robot mobile en présence d'obstacles et de murs dans

l'environnement. Au départ, le robot essaye de se déplacer en ligne droite vers la cible, mais lorsqu'il détecte la présence d'un obstacle ou d'un mur dans son chemin, le comportement approprié est activé et le contrôleur flou génère des actions nécessaires pour effectuer le suivi du mur sans aucune collision ni oscillation. Les Figures IV.20(b) et IV.21(b) présentent les signaux de commande générés par le contrôleur proposé pour assurer le suivi de murs, l'évitement d'obstacles et la convergence vers la cible en toute sécurité et avec la vitesse maximale admissible.

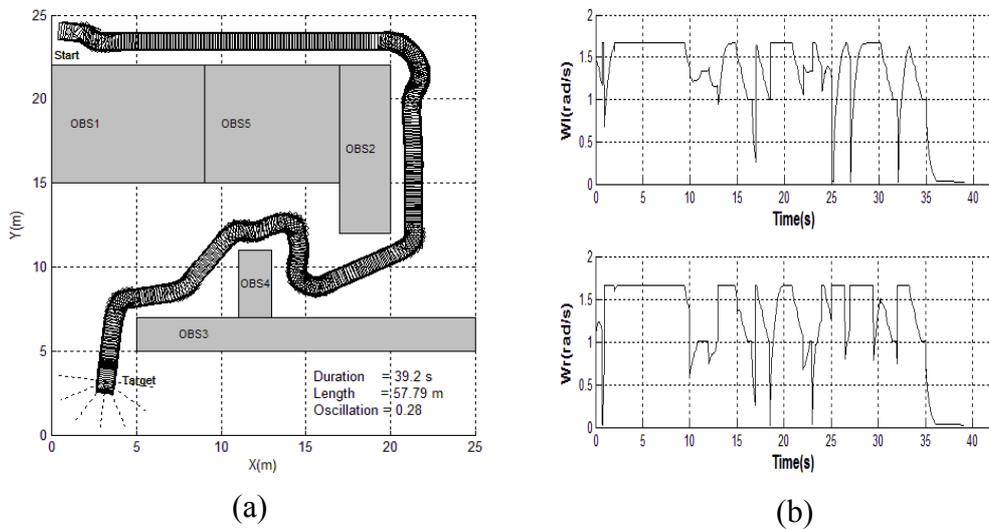


Figure IV.20 Suivi de murs, évitement d'un obstacle puis navigation vers la cible

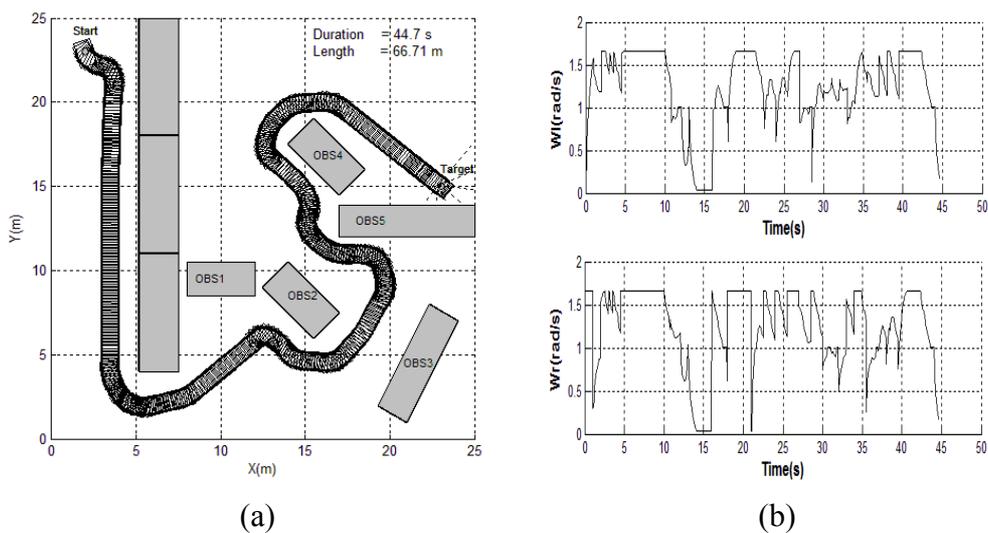
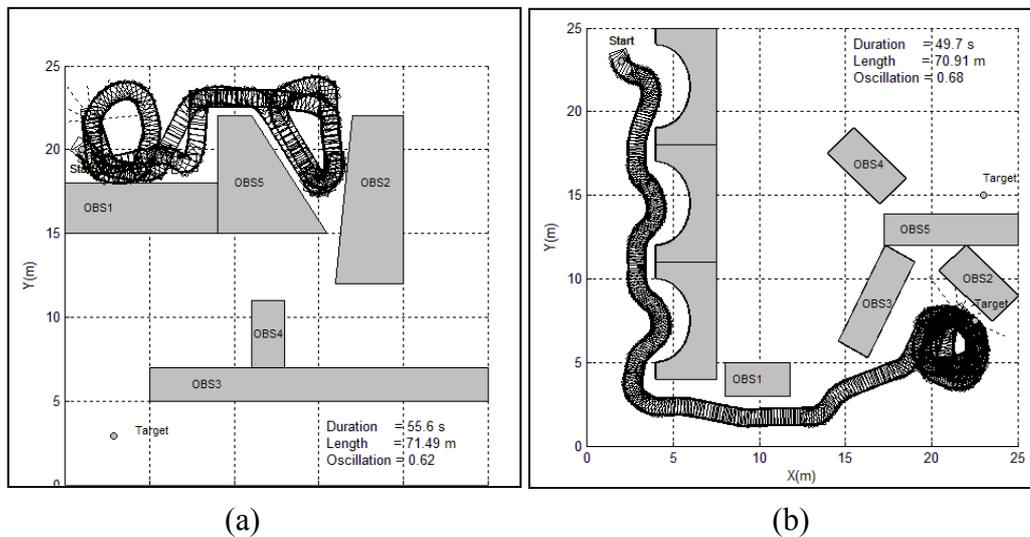


Figure IV.21 Suivi de murs, évitement de plusieurs obstacles puis navigation vers la cible

L'utilisation d'un contrôleur de navigation autonome à base de logique floue pour les robots mobiles donne des résultats acceptables dans les environnements encombrés d'obstacles de formes simples (carré, rectangle, cercle, ...). Mais, en présence des obstacles concaves (forme U), ce type de contrôleur échoue à faire naviguer le robot vers la cible, et le robot se trouve soit immobilisé dans un point appelé minimum local soit dans une boucle infinie sans atteindre sa destination (voir Figure IV.22).



**Figure IV.22** Situation de blocage en boucle infini en présence d'un obstacle concave

#### IV.4 Conclusion

Dans ce chapitre, un contrôleur comportemental à base de logique floue a été conçu pour réaliser la navigation autonome d'un robot mobile dans des environnements inconnus encombrés d'obstacles. L'idée du contrôle comportemental est basée sur la décomposition de la tâche de navigation globale en un ensemble de comportements élémentaires tels que la navigation vers la cible, l'évitement d'obstacles et le suivi de murs. Le contrôleur proposé est constitué de deux modules principaux : un module de décision chargé de la coordination des priorités entre les différents comportements réactifs et un module d'action chargé de contrôler le mouvement du robot au cours de sa navigation pour assurer la convergence vers la

cible. L'incorporation de l'apprentissage génétique dans le processus de conception floue ajoute une dimension intelligente au contrôleur ce qui a permis de générer automatiquement les bases de règles des deux modules flous.

Les résultats de simulation présentés dans ce chapitre montrent l'efficacité du contrôleur proposé surtout dans les environnements moins compliqués. L'inconvénient majeur de ce type de contrôleurs c'est la sensibilité au minimum local notamment en présence d'obstacles concaves.

Le chapitre suivant donne une solution prometteuse pour palier à cet inconvénient en utilisant le concept des obstacles virtuels.

## **Chapitre V :**

### **CONCEPTS DES OBSTACLES VIRTUELS POUR LA RESOLUTION DU PROBLEME DES MINIMA LOCAUX.**

#### **V.1 Introduction**

Le cœur de notre contribution présenté dans ce chapitre réside dans la conception d'une nouvelle approche basée sur la combinaison du concept des obstacles virtuels avec le contrôleur flou présenté dans le chapitre IV pour apporter une solution optimale au problème des minima locaux. L'idée principale de la solution proposée ici consiste à établir, en plus des trois comportements définis précédemment, un quatrième comportement appelé comportement de prévention des situations d'urgence ESP (en Anglais : Emergency Situation Preventing behavior). La méthode proposée est établie sur un algorithme des obstacles virtuels pour la construction du comportement de prévention des situations d'urgence et le raisonnement flou pour la coordination efficace des priorités entre les quatre différents comportements réactifs : navigation vers la cible, évitement d'obstacles, suivi de murs et prévention des situations d'urgence. Au cours de sa navigation vers la cible, le robot essaye d'éviter la collision avec les obstacles dans son chemin, et quand il s'approche d'un obstacle concave où il est fort possible de rencontrer un minimum local, le comportement de prévention des situations d'urgence est activé et l'algorithme des obstacles virtuels est appelé pour calculer des points intermédiaires utilisés en tant que cibles provisoires pour aider le robot à sortir de cette situation d'urgence. Une fois le robot est loin du minimum local, il continue sa tâche de navigation vers la cible finale en suivant le chemin le plus court.

### V.2 Architecture de l'approche de navigation autonome proposée

Nous proposons dans cette partie une architecture de navigation autonome basée sur quatre comportements élémentaires: navigation vers la cible, suivi de murs, évitement d'obstacles et prévention des situations d'urgence. Nous avons adopté le raisonnement flou pour coordonner les priorités entre les différents comportements et un algorithme des obstacles virtuels pour la construction du quatrième comportement.

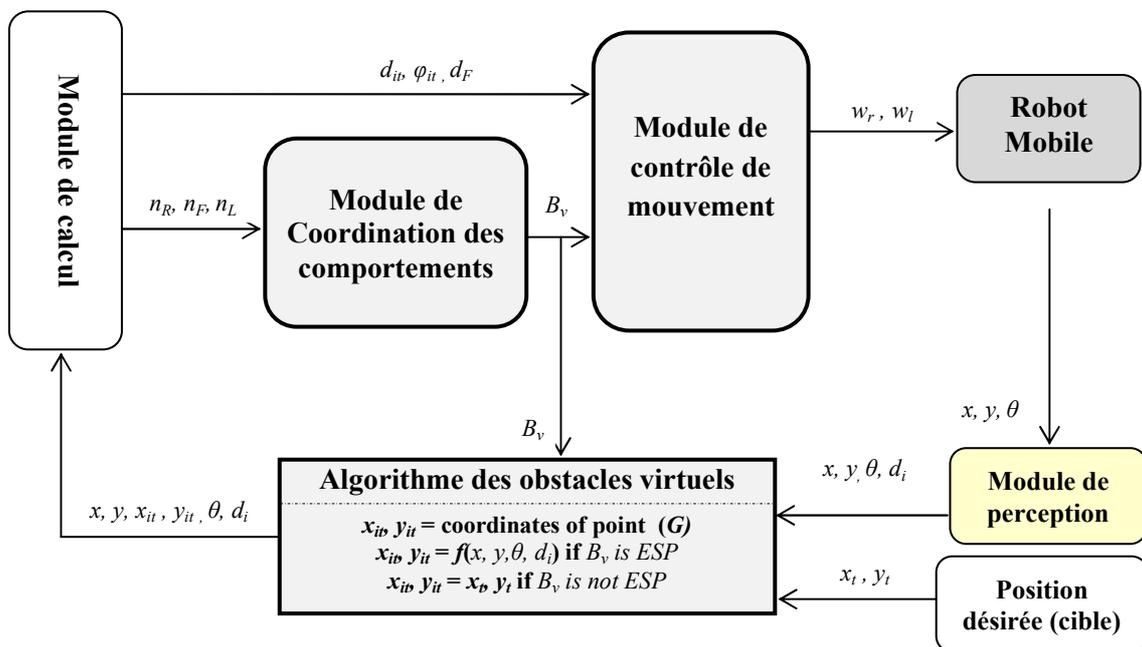


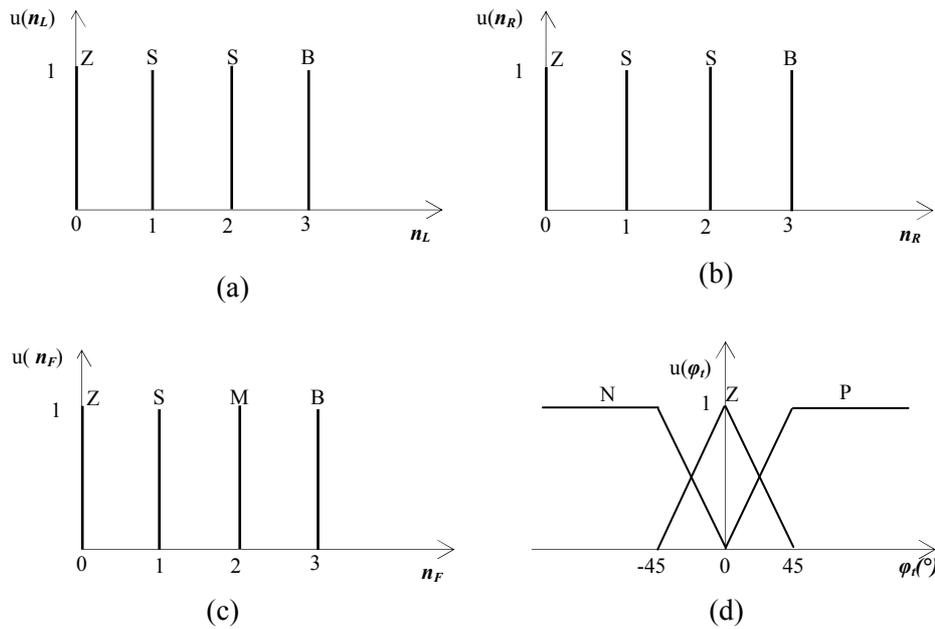
Figure V.1 Architecture du système de navigation autonome proposée

L'approche comportementale hybride proposée est composée principalement d'un module de perception, d'un algorithme des obstacles virtuels et deux contrôleurs flous accomplissant deux tâches élémentaires : la coordination entre les différents comportements réactifs (module de décision) et le contrôle de mouvement du robot mobile (module d'action).

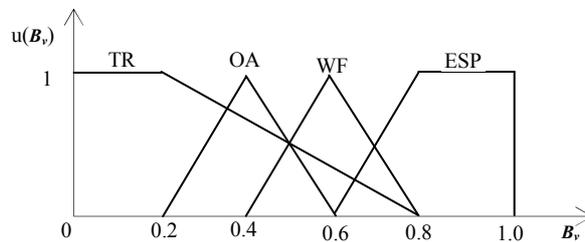
### V.2.1 Module de décision

Dans la stratégie de commande proposée ici, les quatre différents comportements réactifs sont formulés par les ensembles flous dans lesquels les règles floues pour tous les comportements sont intégrées dans une seule base de règle. De cette manière, la coordination entre les différents comportements peut facilement être réalisée en utilisant seulement les données acquises par les capteurs ultrasoniques du robot. La Figure V.1 montre le diagramme de l'architecture hybride proposée qui est établie sur deux systèmes flous, un pour la coordination entre les différents comportements réactifs et l'autre pour le contrôle de mouvement du robot mobile. Le comportement normal dans la navigation autonome de robot mobile est de guider le robot vers la cible. Sur son chemin immédiat, le robot essaye de naviguer vers la cible et quand il s'approche des obstacles ou des murs le comportement concerné est exécuté avec un degré convenable. Une fois le robot est en dehors de la zone de collision, il continue sa tâche de navigation vers la cible en suivant le chemin le plus court.

Dans la stratégie de commande proposée, tous les comportements réactifs sont intégrés dans des fonctions d'appartenance dans un seul univers de discours normalisé. Le contrôleur de coordination des comportements flous a une sortie unique qui est la valeur normale du comportement  $B_V$  et quatre variables d'entrée qui sont respectivement les nombres de capteurs ultrasoniques qui détectent un obstacle vers la gauche  $n_L$ , vers la droite  $n_R$ , vers l'avant  $n_F$  et l'angle d'orientation  $\varphi_t$  (voir Figure IV.1). Leurs fonctions d'appartenance sont montrées sur les Figures V.2 et V.3. Les règles floues utilisées dans la conception du contrôleur proposé sont générées automatiquement par un algorithme génétique (voir Tableau V.1).



**Figure V.2** Les fonctions d'appartenance du coordinateur des comportements: (a, b, c) Les variables d'entrée  $n_L$ ,  $n_R$ ,  $n_F$  avec : Z est Zero, S est Smal, M est Medium et B est Big. (d) La variable d'entrée  $\phi_t$  avec: N est Negative, Z est Zero et P est Positive

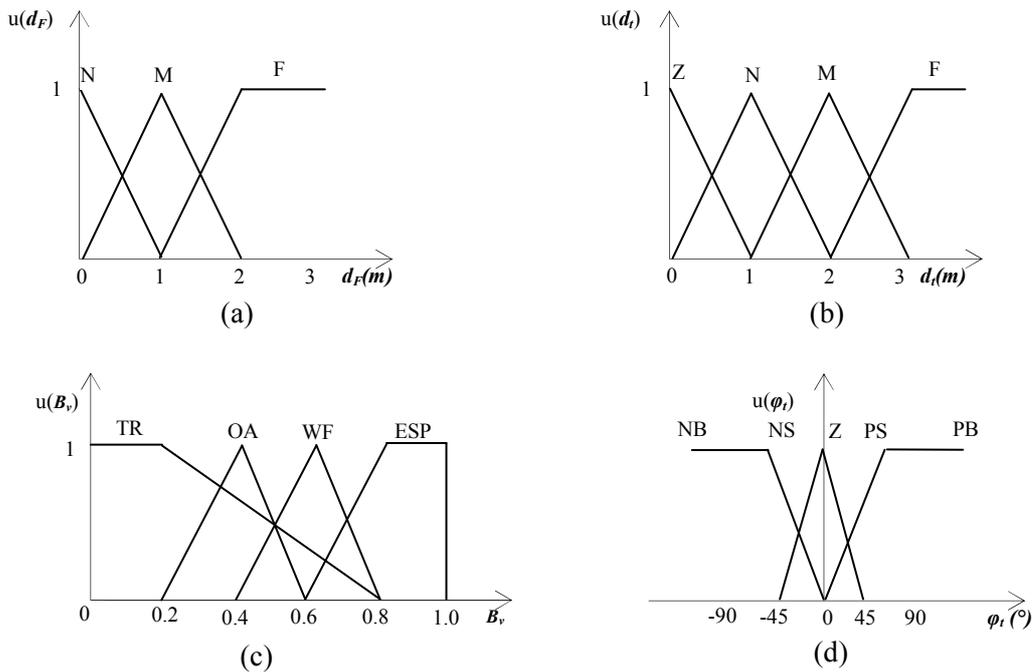


**Figure V.3** Les fonctions d'appartenance du coordinateur des comportements: La variable de sortie  $B_v$  avec: TR est Target Reaching, OA est Obstacle Avoiding , WF est Wall Following et ESP Emergency Situation Preventing

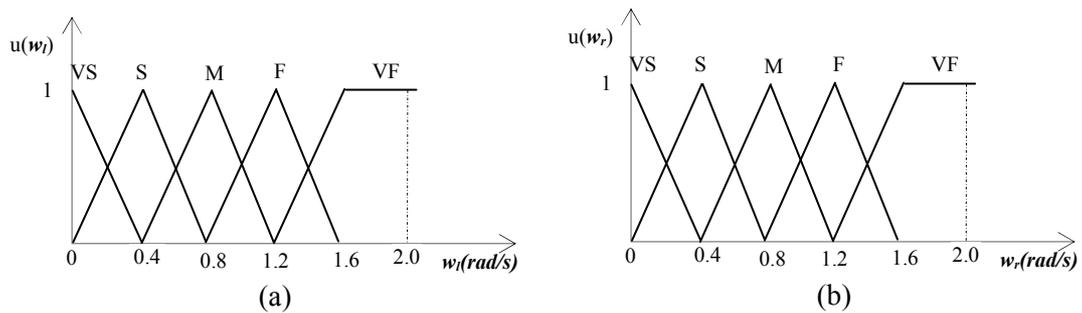
### V.2.2 Module d'action

Il s'agit alors de veiller à réaliser le mouvement ou le déplacement et suivre les consignes de vitesses reçues en appliquant le type de commande adapté. Un contrôleur à base de logique floue est utilisé pour réaliser le contrôle de mouvement du robot mobile dans un environnement inconnu encombré d'obstacles. Comme décrit dans la

section IV.2.1, le robot mobile possède deux roues entraînées indépendamment et sept capteurs ultrasoniques utilisés pour détecter des obstacles en avant, vers la droite et vers la gauche du robot. Selon l'information de distance fournie par ces capteurs et les positions actuelle et cible du robot, l'algorithme des obstacles virtuels calcule les valeurs de  $n_L$ ,  $n_F$ ,  $n_R$  et  $\varphi_t$  (voir Section IV.2.2.1) et les transmet au coordinateur des comportements qui est responsable de décider quel comportement doit être exécuté. Parallèlement, l'algorithme des obstacles virtuels calcule les valeurs de  $d_{it}$ ,  $\varphi_{it}$  et  $d_F$  et les transmet au contrôleur flou de mouvement. Les variables  $d_{it}$  et  $\varphi_{it}$  représentent la distance et l'angle entre la position actuelle du robot et la cible provisoire lorsque le comportement ESP est activé et entre la position actuelle du robot et la cible finale autrement. La variable  $d_F$  représente la distance minimale entre le robot et les obstacles présents en sa direction avant; elle est exploitée par le contrôleur de mouvement pour maximiser la vitesse de navigation du robot. Donc, le contrôleur de mouvement reçoit à ses entrées la valeur normalisée du comportement  $B_V$ , la valeur de la distance  $d_{it}$ , la valeur de l'angle  $\varphi_{it}$  et de la distance minimale des obstacles présents en avant du robot. Les sorties du contrôleur de mouvement sont les deux signaux de commande des vitesses des roues gauche et droite du robot  $w_l$  et  $w_r$ , respectivement. Le contrôleur proposé est basé sur la règle d'implication de MAMDANI (max-min), la conjonction floue (min) et par la méthode de centre de gravité pour la défuzzification. Les fonctions d'appartenances utilisées pour fuzzifier les variables d'entrée et de sortie sont de forme triangulaire et trapézoïdale, elles sont illustrées sur les Figures V.4 et V.5 respectivement. Les règles floues utilisées dans la conception du contrôleur de mouvement sont générées automatiquement par un algorithme génétique (voir Tableau V.2).



**Figure V.4** Les fonctions d'appartenance du contrôleur de mouvement: (a,b) Les variables d'entrée  $d_F$ ,  $d_t$  avec : Z est Zero, N est Near, M est Medium et F est Far. (c) La variable d'entrée  $B_v$  avec: TR est Target Reaching, OA est Obstacle Avoiding, WF est Wall Following et ESP est Emergency situation preventing. (d) La variable d'entrée  $\phi_t$  avec: N est Negative, Z est Zero et P est Positive



**Figure V.5** Les fonctions d'appartenance du contrôleur de mouvement: (a,b) Les variables de sortie  $w_l$ ,  $w_r$  avec : VS est Very Small, S est Small, M est Medium, F est Fast et VF est Very Fast

### V.2.3 Conception de l'algorithme des obstacles virtuels

Dans cette section, nous décrivons en détail une forme modifiée du concept des obstacles virtuels développé d'abord dans [15]. Selon l'information de la distance acquise par les capteurs ultrasoniques lorsque le robot s'approche d'un minimum local, le module de décision active le comportement de prévention des situations d'urgence qui, à son tour, appelle l'algorithme des obstacles virtuels pour calculer une cible intermédiaire en dehors de la zone de risque. Un exemple d'une situation d'urgence est illustré sur la figure V.6 où les valeurs des quatre variables d'entrée sont respectivement :  $n_L = 3$ ,  $n_R = 3$ ,  $n_F = 3$ , et  $\varphi_t = 6^\circ$ .

En utilisant la règle floue suivante :

- Si  $n_L$  est B et  $n_R$  est B et  $n_F$  est B et  $\varphi_t$  est Z alors  $B_V$  est ESP (Emergency Situation Preventing).

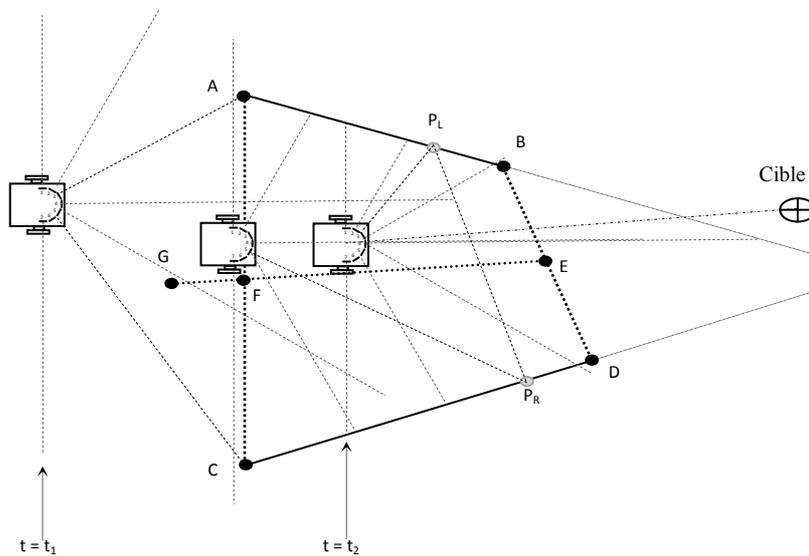
L'algorithme utilisé pour calculer la cible intermédiaire est structuré en trois étapes :

- **Etape 1** : Les deux plus proche segments de ligne détectés par le robot mobile sont construits avec les extrémités  $AB$  et  $CD$ . Les points  $A$  et  $C$  sont les deux premiers points détectés simultanément par les capteurs ultrasoniques du côté gauche et droit au moment  $t = t_1$  (voir la figure V.6). S'il y a continuité de détection de points des obstacles sur les deux côtés gauche et droit, le robot est à l'intérieur d'un couloir ou d'un obstacle concave. Puis, si la distance entre les deux points  $P_L$  et  $P_R$  est moins de deux fois le diamètre du robot, le robot est emprisonné dans un obstacle concave. A ce moment ( $t = t_2$ ), le module de décision déclare une situation d'urgence et les points  $B$  et  $D$  sont choisis comme étant les deux derniers points détectés par les capteurs ultrasoniques sur la gauche et la droite du robot respectivement.

- **Etape 2** : Les points médians  $E$  et  $F$  des segments  $BD$  et  $CA$  sont choisis et reliés avec le point  $G$  en prenant la longueur du segment  $FG$  égale à deux fois le diamètre du robot. Le robot doit naviguer vers la cible intermédiaire  $G$  pour sortir de l'obstacle

concave et éviter le minimum local. Pendant ce temps, le comportement de prévention des situations d'urgence est activé avec une haute priorité.

- **Etape 3** : Une fois le robot atteint le point  $G$ , un obstacle virtuel est construit autour des extrémités  $ABDC$  afin de prévenir le robot d'être piégé une deuxième fois dans la même situation d'urgence. Après l'atteinte du point  $G$ , d'autres comportements peuvent être activés selon l'information acquise par les capteurs ultrasoniques pour guider le robot vers la cible finale en toute sécurité.



**Figure V.6** Principe de l'algorithme des obstacles virtuels

#### V.2.4 Génération des bases de règles par l'algorithme génétique:

Les Tableaux V.1 et V.2 représentent les bases des règles floues générées par l'algorithme génétique proposé dans la section IV.2.3.7.

Inputs				Output	Inputs				Output
$n_L$	$n_R$	$n_F$	$\varphi_t$	$B_v$	$n_L$	$n_R$	$n_F$	$\varphi_t$	$B_v$
Z	Z	Z	N	TR	S	Z	M	Z	OA
S	Z	Z	N	TR	B	Z	M	Z	WF
Z	S	Z	N	TR	Z	S	M	Z	OA
S	S	Z	N	TR	S	S	M	Z	OA
Z	Z	S	N	OA	B	S	M	Z	WF
S	Z	S	N	TR	Z	B	M	Z	WF
B	Z	S	N	TR	S	B	M	Z	WF
Z	S	S	N	OA	B	B	M	Z	ESP
S	S	S	N	TR	S	S	B	Z	OA
B	S	S	N	TR	B	S	B	Z	ESP
Z	B	S	N	WF	S	B	B	Z	ESP
S	B	S	N	WF	B	B	B	Z	ESP
S	Z	M	N	TR	Z	Z	Z	P	TR
B	Z	M	N	TR	S	Z	Z	P	TR
Z	S	M	N	OA	Z	S	Z	P	TR
S	S	M	N	OA	S	S	Z	P	TR
B	S	M	N	OA	Z	Z	S	P	OA
Z	B	M	N	WF	S	Z	S	P	TR
S	B	M	N	WF	B	Z	S	P	WF
B	B	M	N	ESP	Z	S	S	P	TR
S	S	B	N	OA	S	S	S	P	TR
B	S	B	N	OA	B	S	S	P	WF
S	B	B	N	ESP	Z	B	S	P	WF
B	B	B	N	ESP	S	B	S	P	WF
Z	Z	Z	Z	TR	S	Z	M	P	TR
S	Z	Z	Z	TR	B	Z	M	P	WF
Z	S	Z	Z	TR	Z	S	M	P	TR
S	S	Z	Z	TR	S	S	M	P	OA
Z	Z	S	Z	TR	B	S	M	P	WF
S	Z	S	Z	TR	Z	B	M	P	TR
B	Z	S	Z	WF	S	B	M	P	TR
Z	S	S	Z	TR	B	B	M	P	ESP
S	S	S	Z	TR	S	S	B	P	OA
B	S	S	Z	WF	B	S	B	P	OA
Z	B	S	Z	WF	S	B	B	P	OA
S	B	S	Z	WF	B	B	B	P	ESP

Tableau V.1 La base des règles floues du module de décision

Inputs				Outputs		Inputs				Outputs	
$d_F$	$d_t$	$\varphi_t$	$B_v$	$w_l$	$w_r$	$d_F$	$d_t$	$\varphi_t$	$B_v$	$w_l$	$w_r$
N	Z	N	TR	S	VS	N	Z	P	OA	VS	S
M	Z	N	TR	M	VS	M	Z	P	OA	M	S
F	Z	N	TR	M	VS	F	Z	P	OA	M	S
N	N	N	TR	S	VS	N	N	P	OA	VS	S
M	N	N	TR	M	S	M	N	P	OA	VS	S
F	N	N	TR	F	S	F	N	P	OA	VS	S
N	M	N	TR	VS	VS	N	M	P	OA	VS	S
M	M	N	TR	M	S	M	M	P	OA	S	M
F	M	N	TR	F	M	F	M	P	OA	M	F
N	F	N	TR	S	VS	N	F	P	OA	VS	S
M	F	N	TR	M	S	M	F	P	OA	S	M
F	F	N	TR	F	M	F	F	P	OA	M	F
N	Z	Z	TR	VS	VS	N	Z	N	WF	VS	VS
M	Z	Z	TR	VS	VS	M	Z	N	WF	S	S
F	Z	Z	TR	VS	VS	F	Z	N	WF	S	S
N	N	Z	TR	S	S	N	N	N	WF	VS	VS
M	N	Z	TR	S	S	M	N	N	WF	S	S
F	N	Z	TR	S	S	F	N	N	WF	M	M
N	M	Z	TR	M	M	N	M	N	WF	S	S
M	M	Z	TR	F	F	M	M	N	WF	M	M
F	M	Z	TR	F	F	F	M	N	WF	F	F
N	F	Z	TR	M	M	N	F	N	WF	S	S
M	F	Z	TR	F	F	M	F	N	WF	F	F
F	F	Z	TR	VF	VF	F	F	N	WF	VF	VF
N	Z	P	TR	VS	S	N	Z	Z	WF	VS	VS
M	Z	P	TR	VS	M	M	Z	Z	WF	S	S
F	Z	P	TR	VS	M	F	Z	Z	WF	S	S
N	N	P	TR	VS	S	N	N	Z	WF	S	S
M	N	P	TR	S	M	M	N	Z	WF	M	M
F	N	P	TR	S	F	F	N	Z	WF	M	M
N	M	P	TR	VS	S	N	M	Z	WF	M	M
M	M	P	TR	S	M	M	M	Z	WF	F	F
F	M	P	TR	M	F	F	M	Z	WF	S	S
N	F	P	TR	VS	S	N	F	Z	WF	F	F
M	F	P	TR	S	M	M	F	Z	WF	VF	VF
F	F	P	TR	M	F	F	F	Z	WF	VS	VS
N	Z	N	OA	S	VS	N	Z	P	WF	S	S
M	Z	N	OA	M	VS	M	Z	P	WF	S	S
F	Z	N	OA	M	VS	F	Z	P	WF	S	S
N	N	N	OA	S	VS	N	N	P	WF	S	S
M	N	N	OA	M	S	M	N	P	WF	S	S
F	N	N	OA	F	S	F	N	P	WF	M	M
N	M	N	OA	S	VS	N	M	P	WF	S	S
M	M	N	OA	M	S	M	M	P	WF	M	M
F	M	N	OA	F	M	F	M	P	WF	F	F
N	F	N	OA	S	VS	N	F	P	WF	S	S
M	F	N	OA	M	S	M	F	P	WF	F	F
F	F	N	OA	F	M	F	F	P	WF	VF	VF
N	Z	Z	OA	VS	VS	N	Z	Z	ESP	VS	VS
M	Z	Z	OA	VS	VS	M	Z	Z	ESP	S	S
F	Z	Z	OA	VS	VS	F	Z	Z	ESP	S	S
N	N	Z	OA	S	S	N	N	Z	ESP	S	S
M	N	Z	OA	S	S	M	N	Z	ESP	M	M
F	N	Z	OA	S	S	F	N	Z	ESP	M	M
N	M	Z	OA	M	M	N	M	P	ESP	VS	S
M	M	Z	OA	M	M	M	M	P	ESP	S	M
F	M	Z	OA	F	F	F	M	P	ESP	M	F
N	F	Z	OA	S	S	N	F	P	ESP	S	M
M	F	Z	OA	M	M	M	F	P	ESP	M	F
F	F	Z	OA	VF	VF	F	F	P	ESP	F	VF

Tableau V.2 La base des règles floues du module d'action

### V.3 Résultats de simulation

Dans cette section, on examine la théorie développée dans les sections précédentes. On commence par comparer les performances de navigation obtenues par la méthode des cibles virtuelles proposée dans [18] avec celles obtenues par le contrôleur comportemental flou proposé dans ce chapitre en utilisant deux types différents d'environnement contenant des situations de piège réelles. Ensuite, on évalue les améliorations obtenues par l'intégration de l'algorithme des obstacles virtuels avec le contrôleur comportemental flou.

#### V.3.1 Critères de comparaison

Afin de mesurer objectivement les performances de la méthode proposée dans diverses conditions, on considère les trois paramètres suivants :

**-La longueur totale du parcours  $L_t$**  : elle représente la distance totale parcourue par le robot entre les positions de départ et d'arrivée. Elle est donnée par l'équation suivante :

$$L_t = \sum_{k=0}^N v_k \tau \quad (\text{V.1})$$

**-La durée totale du parcours  $D_c$**  : elle représente le temps total pris par le robot pour se déplacer entre les positions de départ et d'arrivée. Elle est donnée par l'équation suivante :

$$D_c = N\tau \quad (\text{V.2})$$

**-Le taux d'oscillation  $R_o$**  : en effet, le robot oscille quand il change sa direction de navigation ; la surveillance de la vitesse de rotation instantanée ( $w_k$ ) constitue un critère important pour évaluer le taux d'oscillation du robot au cours de la navigation. Il est donné par l'équation suivante :

$$R_o = \frac{1}{N} \sqrt{\sum_{k=0}^N w_k^2} \quad (\text{V.3})$$

Où  $v_k$  et  $w_k$  représentent les vitesses linéaire et de rotationnelle du robot mobile à l'instant  $k$ ,  $\tau$  représente la durée du pas de calcul et  $N$  est le nombre de pas de calcul entre le moment de départ du robot et le moment de son arrivée à la cible.

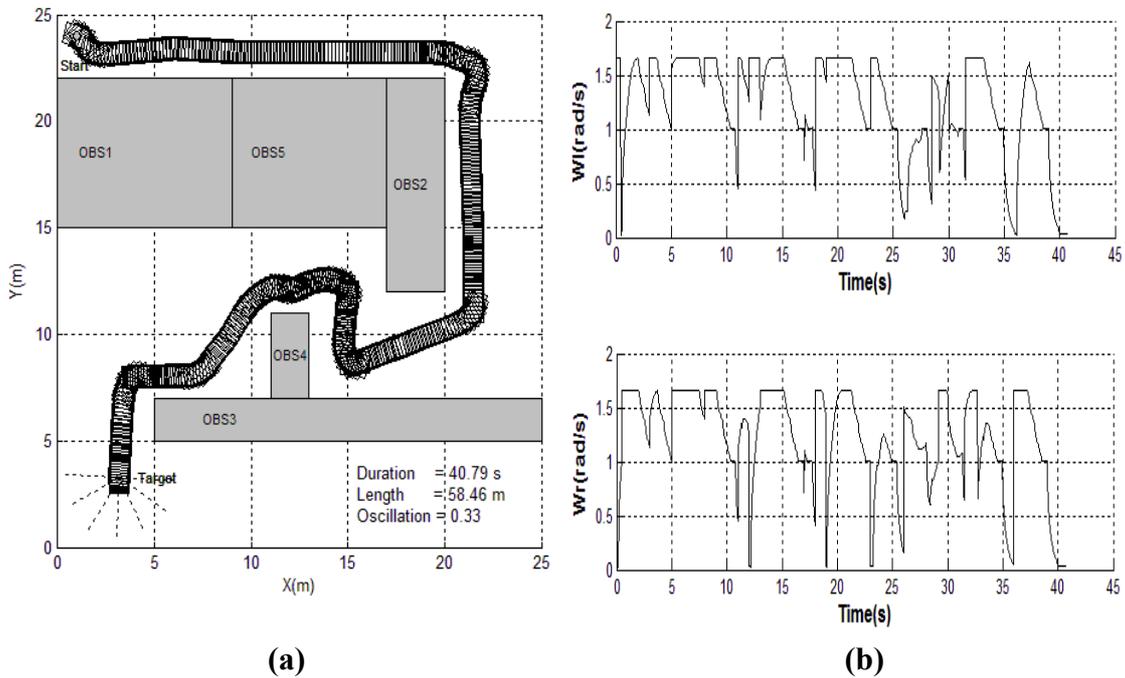
### V.3.2 Résultats et comparaisons

Pour montrer l'efficacité de l'approche proposée, une série de simulations a été conduite en utilisant deux types d'environnements arbitrairement construits comprenant des obstacles. La position de tous les obstacles est inconnue ; le robot ne connaît que les positions de départ et d'arrivée (cible). Toutes les simulations ont été faites dans l'environnement de Matlab/Simulink en utilisant le modèle cinématique du robot donné par l'Equation IV.1. La valeur numérique de la durée du pas de calcul est 0.01s dans toutes les simulations. Les valeurs de  $L_t$ ,  $D_c$ ,  $R_o$  et  $N$  sont variables et dépendent de la longueur et de la forme de la trajectoire parcourue par le robot. Leurs valeurs numériques sont indiquées dans le Tableau 5.

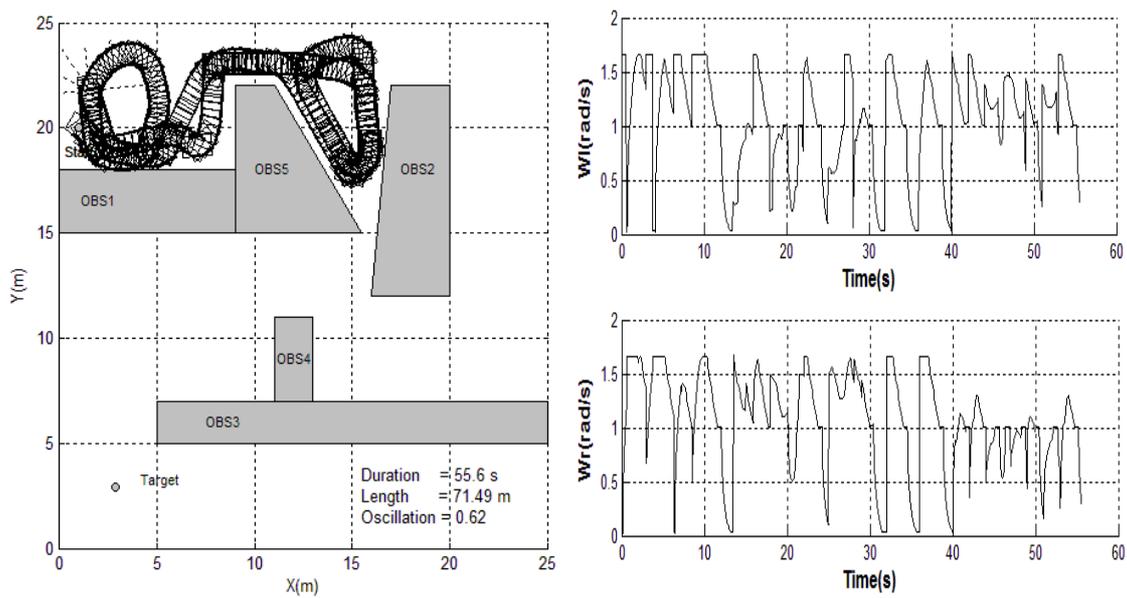
#### V.3.2.1 Résultats de simulation dans l'environnement type1

On commence la simulation par employer la méthode des cibles virtuelles proposée dans [18]. Cette méthode consiste à générer une série de cibles virtuelles par lesquelles le robot doit passer pour éviter la collision avec les obstacles existant dans son espace de travail (pour plus de détail sur cette méthode il possible de consulter la référence [18]). Les graphes des Figures V.7 et V.8 montrent les trajectoires de navigation et les signaux de commande des roues gauche et droite du robot. Sur la Figure V.7(a), le robot réussit à éviter les obstacles et à atteindre la cible finale avec succès. Pour tester la sensibilité de cette méthode, un obstacle concave a été introduit dans l'espace de travail robot pour créer un minimum local. La Figure V.8 (a) montre clairement que le robot est facilement tombé dans le piège; il se trouve emprisonné dans une boucle infinie et ne pourra jamais atteindre la cible finale. Un autre inconvénient important de cette méthode c'est la nécessité d'accélérer et de ralentir le robot (voir Figures V.7(b) et V.8 (b)) pour atteindre le nombre élevé de cibles

virtuelles qui sont souvent très proches du robot. Ce phénomène contribue à diminuer la vitesse de navigation et à augmenter le taux d'oscillation du robot.

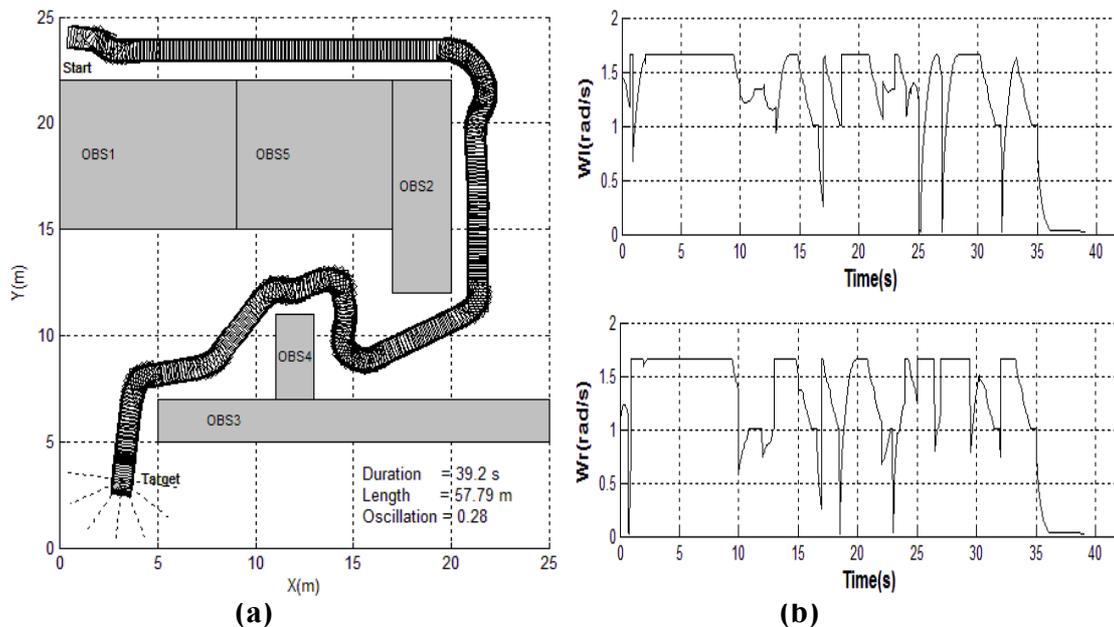


**Figure V.7** Navigation par la méthode des cibles virtuelles en absence de minimum local : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite

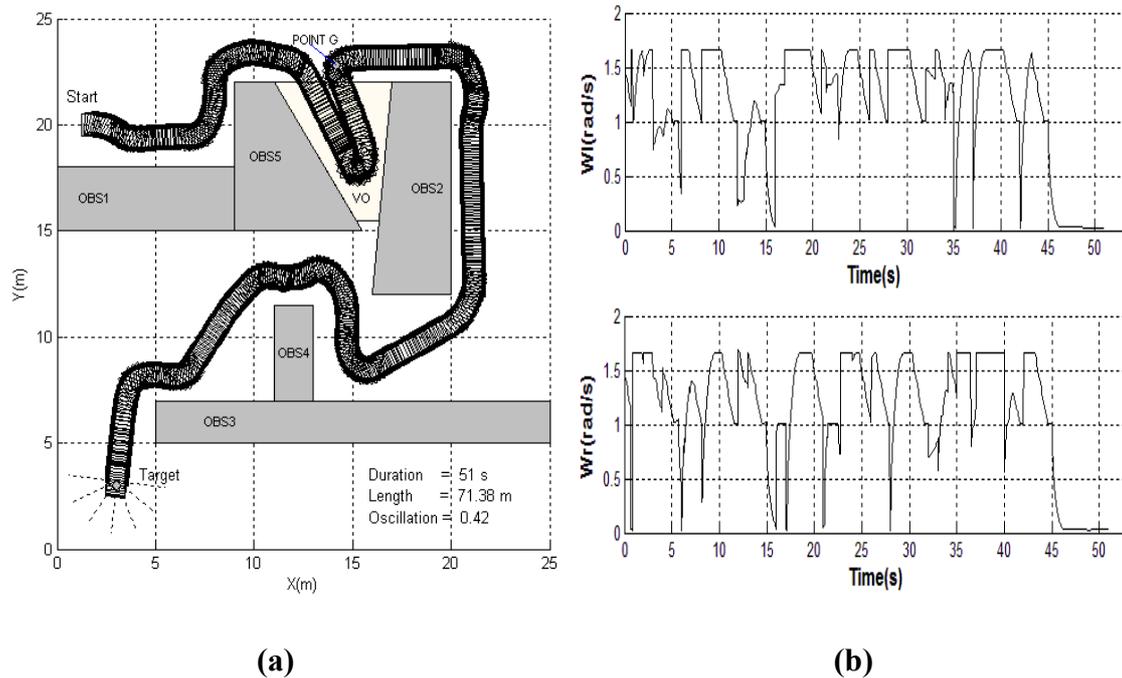


**Figure V.8** Navigation par la méthode des cibles virtuelles en présence de minimum local : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite

Par l'intégration de l'algorithme des obstacles virtuels avec le contrôleur comportemental, on réussit à améliorer de manière significative la qualité de la trajectoire de navigation et les signaux de commande des roues gauche et droite du robot. Les graphes des Figures V.9 et V.10 exposent clairement l'avantage obtenu par cette combinaison. Sur la Figure V.9(a), le robot réussit à éviter les obstacles et à atteindre la cible finale en toute sécurité. Pour tester la sensibilité de la méthode proposée, un obstacle concave a été introduit dans l'espace de travail robot pour créer un piège. En effet, lorsque le robot est emprisonné dans un minimum local, le comportement de prévention des situations d'urgence est activé et l'algorithme des obstacles virtuels est invoqué pour calculer les coordonnées de la cible intermédiaire (point G sur la Figure V.10 (a)) vers laquelle le robot doit se déplacer pour sortir du minimum local. Quand le robot est en dehors de la zone de risque, un obstacle virtuel est construit autour des extrémités de l'obstacle concave pour empêcher le robot d'être piégé une deuxième fois dans le même minimum local. De cette manière, le robot a pu facilement sortir de la situation d'urgence et a atteint la cible finale en toute sécurité. Les Figures V.9(b) et V.10(b) illustrent les signaux de commande des roues gauche et droite du robot mobile.



**Figure V.9** Navigation par la méthode comportementale proposée en absence de minimum local : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite.

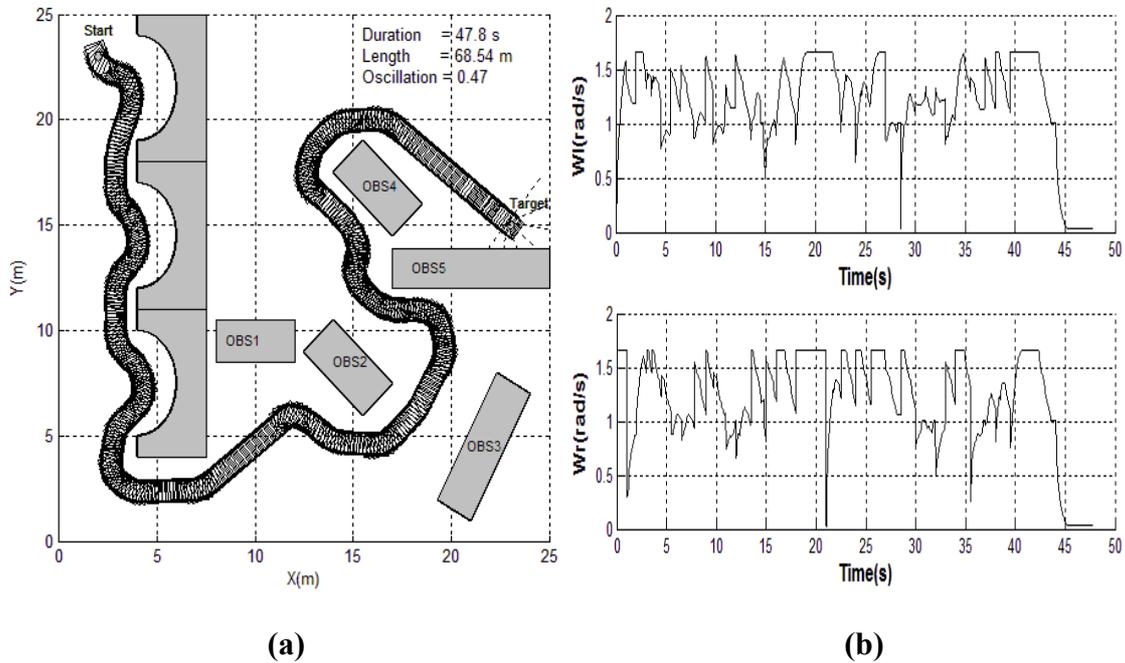


**Figure V.10** Navigation par la méthode comportementale proposée en présence de minimum local : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite.

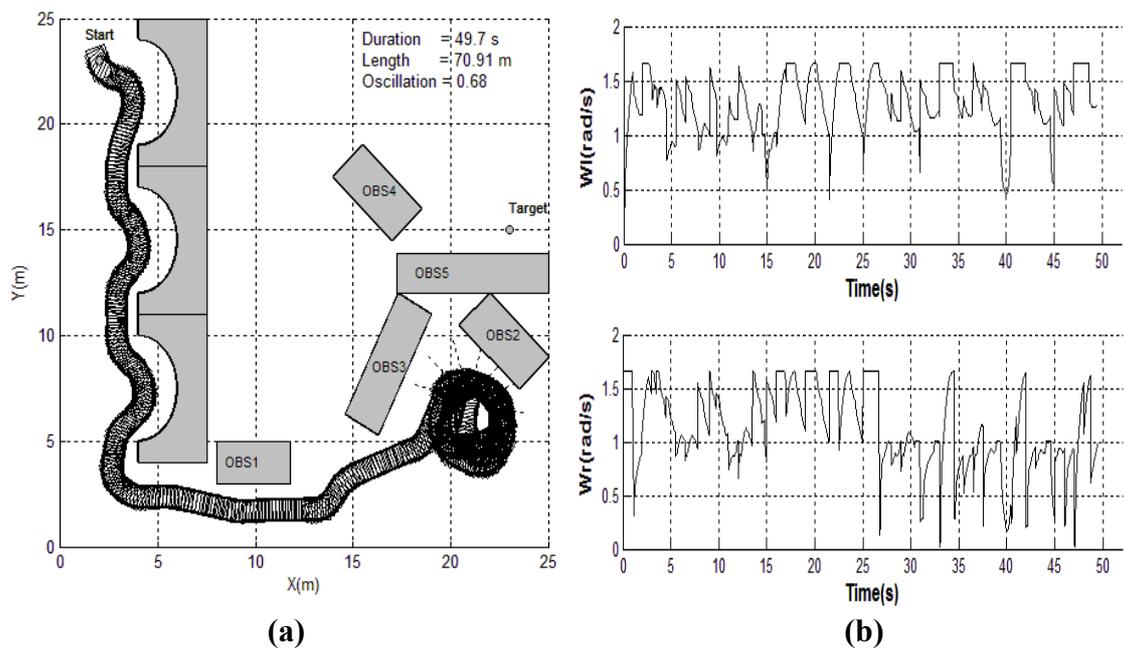
### V.3.2.2 Résultats de simulation dans l'environnement type2

Pour démontrer plus les capacités de la méthode proposée, un environnement plus compliqué a été choisi, il est semblable à celui utilisé dans [18]. Nous commençons la simulation par utiliser la méthode des cibles virtuelles. Sur la Figure V.11(a), le robot réussit à éviter les obstacles et à atteindre la cible finale. Mais sur la Figure V.12(a), où les positions des obstacles OBS1, OBS2 et OBS3 ont été arrangées pour créer une situation de piège profond, le robot est facilement emprisonné dans un minimum local. Ajouter une distance et un angle aléatoires à la cible virtuelle, comme donnée dans [18], pour sortir le robot de cette situation n'est pas efficace et le robot se trouve dans une boucle infinie et ne pourra plus atteindre la cible finale. En utilisant la méthode proposée, une navigation douce est obtenue comme il est clairement illustré par les Figures V.13 et V.14. Sur le graphe V.14(a), le robot a réussi de sortir de la situation du piège profond et a atteint la cible finale avec un temps et oscillation minimales, ce

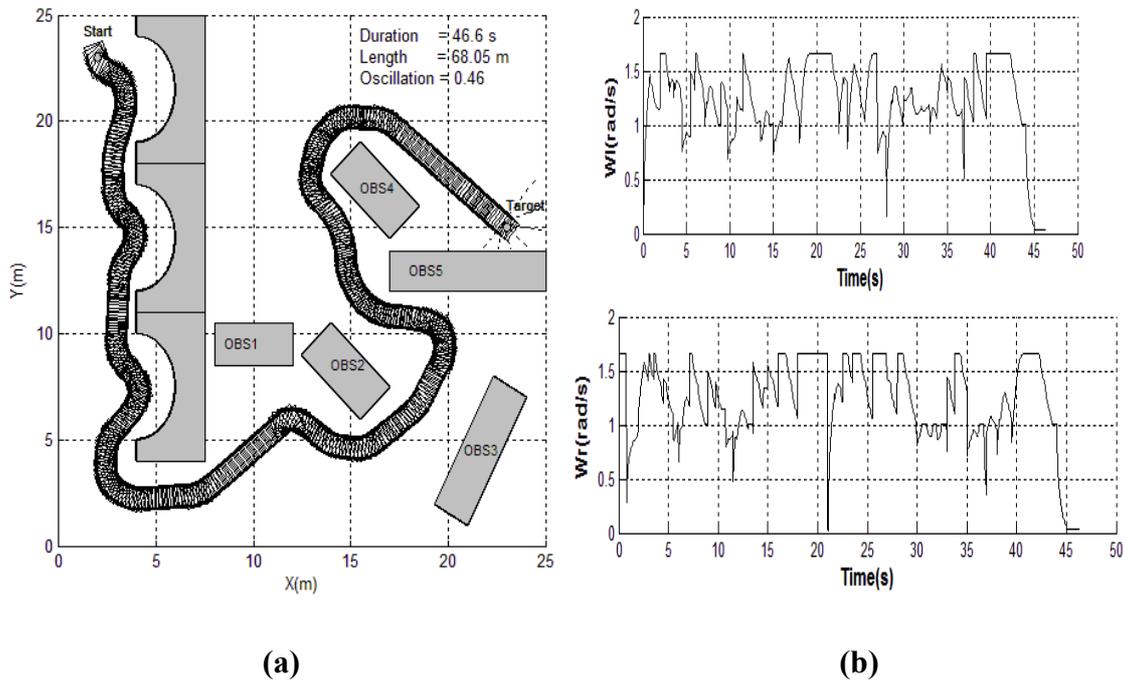
qui prouve l'efficacité de la méthode proposée même dans les environnements très compliqués.



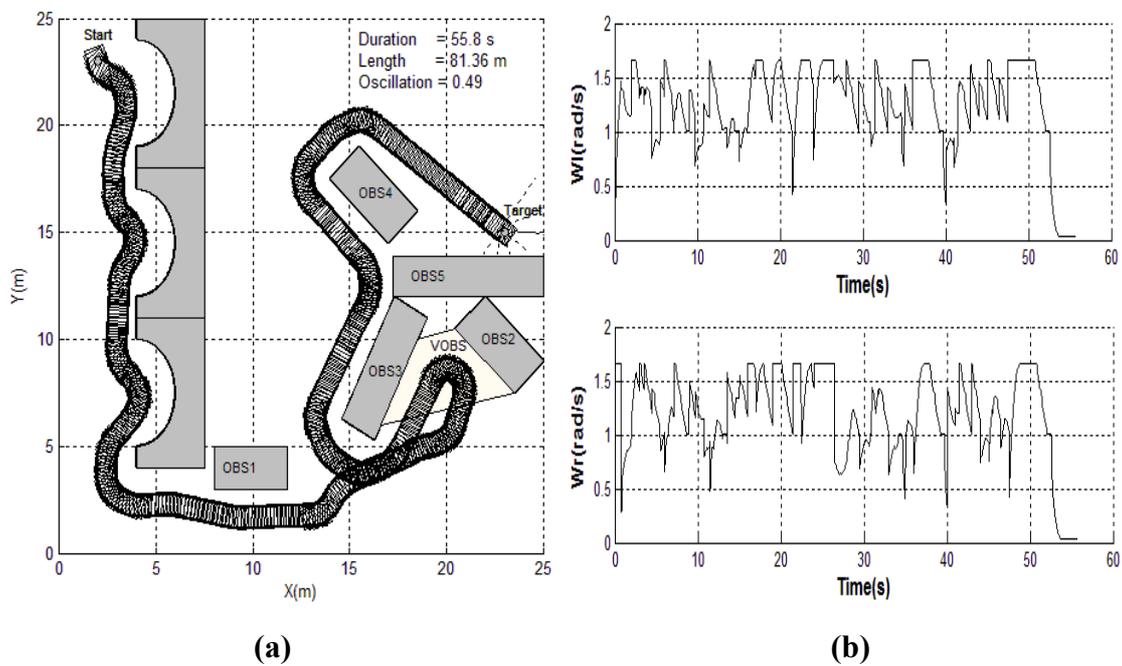
**Figure V.11** Navigation par la méthode des cibles virtuelles en absence de minimum local profond : (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite



**Figure V.12** Navigation par la méthode des cibles virtuelles en présence de minimum local profond: (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite.



**Figure V.13** Navigation par la méthode proposée en absence de minimum local profond: (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite



**Figure V.14** Navigation par la méthode proposée en présence de minimum local profond: (a) Trajectoire parcourue par le robot. (b) Signaux de commandes des roues gauche et droite

Méthodes		Paramètres	Durée	Longueur	Oscillation	Nombre de pas $N$	Cible
			$D_c(s)$	$L_i(m)$	$R_o$		
Méthode des cibles virtuelles	Environnement type1	Sans minimum local	40.79	58.46	0.33	4079	Atteinte
		Avec minimum local	55.6	71.49	0.62	5560	Non Atteinte
	Environnement type2	Sans minimum local	47.8	68.54	0.47	4780	Atteinte
		Avec minimum local	49.7	70.91	0.68	4970	Non Atteinte
Méthode proposée	Environnement type1	Sans minimum local	39.2	57.79	0.28	3920	Atteinte
		Avec minimum local	51	71.38	0.42	5100	Atteinte
	Environnement type2	Sans minimum local	46.6	68.05	0.46	4660	Atteinte
		Avec minimum local	55.8	81.36	0.49	5580	Atteinte

**Tableau V.3** Comparaison des performances entre les deux types de contrôle

#### V.4 Conclusion

En comparant les deux méthodes, chacune est prise sous ses meilleures conditions; on peut conclure que l'approche proposée améliore les performances de la navigation à plusieurs niveaux : l'oscillation et la longueur de la trajectoire sont minimales. La durée du parcours est minimale et proportionnelle à la distance entre les positions de départ et d'arrivée (voir Tableau V.3). L'intégration de l'algorithme des obstacles virtuels contribue à prévenir le robot d'être emprisonné dans des minima locaux créés par des obstacles concaves. La méthode des cibles virtuelles présente deux inconvénients principaux ; le premier est sa vulnérabilité au problème des minima locaux particulièrement en présence des obstacles profondément concaves où le robot se trouve emprisonné dans une boucle infinie et ne pourra pas atteindre la cible finale.

Le second est le phénomène d'accélération et de décélération au niveau des signaux de commande dus au nombre élevé de cibles virtuelles que le robot devrait atteindre. Ce phénomène augmente le taux d'oscillation et diminue la vitesse de la navigation du robot.

## CONCLUSION GENERALE

Dans cette thèse, nous avons contribué à la résolution du problème de la navigation autonome d'un robot mobile dans des environnements encombrés et inconnus. Nos contributions portent sur le développement de nouvelles méthodes inspirées de l'intelligence artificielle en combinant le raisonnement flou et le concept des obstacles virtuels pour améliorer les performances de navigation en dotant le robot mobile de capacités décisionnelles lui permettant de prendre ses propres décisions. Cela signifie qu'il doit être capable à la fois de percevoir correctement son environnement et également de savoir comment réagir en conséquence suivant le niveau d'autonomie. C'est à lui de planifier son parcours et de déterminer avec quels mouvements va-t-il atteindre son objectif ?

Dans le cadre de cette problématique, notre travail de thèse se résume à travers les points suivants :

- Nous avons développé un contrôleur flou basé sur une nouvelle approche comportementale où la tâche globale de navigation est subdivisée en un ensemble de comportements élémentaires d'action simple à concevoir et à gérer. Le cœur de notre contribution dans cette partie réside dans l'intégration des trois comportements : navigation vers la cible, évitement d'obstacles et suivi de murs, dans une seule base de règles et leurs fonctions d'appartenance dans un seul univers de discours normalisé. De cette manière, une transition douce entre les trois différents comportements est obtenue ce qui minimise les oscillations et améliore les performances de navigation du robot mobile lorsqu'il se déplace près des murs et des obstacles. La génération automatique des bases de règles floues est obtenue grâce à un algorithme génétique afin d'éliminer le besoin de l'expertise humaine au cours de la conception d'un tel contrôleur.

Les résultats de simulation présentés dans cette partie affirment l'efficacité de l'approche proposée.

- Nous avons combiné le raisonnement flou avec le concept des obstacles virtuels pour résoudre le problème des minima locaux notamment en présence des obstacles profondément concaves. L'idée principale de la solution proposée ici consiste à établir, en plus des trois comportements définis précédemment, un quatrième comportement appelé comportement de prévention des situations d'urgence ESP. La méthode proposée est établie sur un algorithme des obstacles virtuels pour la construction du comportement de prévention des situations d'urgence et le raisonnement flou pour la coordination efficace des priorités entre les quatre différents comportements réactifs. Selon l'information de la distance acquise par les capteurs ultrasoniques, lorsque le robot s'approche d'un minimum local, le module de décision active le comportement de prévention des situations d'urgence qui, à son tour, appelle l'algorithme des obstacles virtuels pour calculer des points intermédiaires utilisés en tant que cibles provisoires pour aider le robot à sortir de cette situation d'urgence. Une fois le robot est loin du minimum local, il continue sa tâche de navigation vers la cible finale en suivant le chemin le plus court.
- Une comparaison entre la méthode des cibles virtuelles et celle des obstacles virtuels, proposée dans cette thèse, a été conduite pour illustrer la différence entre les deux méthodes particulièrement en présence des obstacles profondément concaves.

Nous avons synthétisé dans ce travail de thèse une méthode complète pour l'amélioration des performances de navigation d'un robot mobile dans un environnement encombré, complexe et inconnu, les limitations des méthodes classiques, spécifiquement le problème des oscillations et celui des minima locaux ont été significativement minimisées et dans certains cas complètement éliminées.

Tous ces contributions ont été valorisé par une publication dans un journal international à comité de lecture et des communications internationales. La liste suivante résume l'ensemble de ces travaux.

**Publication:**

- "Improving mobile robot navigation by combining fuzzy reasoning and virtual obstacle algorithm", Journal of Intelligent & Fuzzy Systems, IOS Press, Vol. 30, No. 3, pp. 1499–1509, 2016.

**Communications Internationales:**

- "Combining planned and reactive controls to improve mobile robot navigation in unknown environments", The 1st International Electrical and computer Engineering Conference IECEC 2015, May 23-25, 2015, Setif, Algeria.
- "A new method for mobile robot navigation in unstructured environments", The 15th International Arab Conference on Information Technology ACIT'2014, December 09-11, 2014, University of Nizwa Oman, [www.acit2k.org](http://www.acit2k.org).
- "Mobile robot navigation in unknown environment using improved APF method", The 13th International Arab Conference on Information Technology ACIT'2012, December 10-13, 2012, Saudi Arabia, [www.acit2k.org](http://www.acit2k.org).
- "Motion control of non-holonomic mobile manipulator using fuzzy logic", The International Arab Conference on Information Technology ACIT'2011, December 11-14, 2011, Saudi Arabia, [www.acit2k.org](http://www.acit2k.org).
- "Combined control for non-holonomic mobile manipulator using fuzzy logic and genetic algorithms", 1<sup>st</sup> International Symposium on Computing in Science & Engineering ISCSE, June 3-5, 2010, Kusadasi, Turkey, <http://iscse2010.gediz.edu.tr>.

## PERSPECTIVES

L'approche proposée dans cette thèse traite le problème de la navigation des robots mobiles dans un espace encombré d'obstacles non-humains. Quand les robots partagent cet espace avec les hommes, ils doivent prendre en compte leur présence et se comporter de manière socialement acceptable. Leurs trajectoires doivent être non seulement sûres mais également lisibles et prévisibles. Leurs comportements doivent suivre des conventions sociales respectant les contraintes de proxémie, évitant les personnes qui partagent le même environnement. Ce qui étend le problème de la navigation à celui de la navigation centrée humaine (human aware navigation).

La navigation des robots en société doit donc s'adapter à la manière d'interagir des hommes à leur confort, leurs préférences et leurs besoins. Cela implique que ces robots soient capables de distinguer les obstacles des personnes et se comporter de manière appropriée (par exemple, en gardant une distance par rapport à une personne).

Comme suite à nos travaux, Plusieurs perspectives sont donc envisageables traitant le problème de la navigation centrée humaine. Elles peuvent aborder à titre d'exemples :

- La conception d'un algorithme de perception de l'environnement dynamique et humain.
- L'intégration de modèles de comportements d'obstacles dynamiques et humains dans la décision de navigation.
- Le développement d'un algorithme de navigation suivant des comportements socialement acceptables.
- L'étude des intentions des personnes dans la décision de navigation.

## REFERENCES BIBLIOGRAPHIQUES

- [1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *The Inter. Journal of Robotics Research*, Vol. 5, No. 1, pp. 90-98, 1986.
- [2] Y. Koren & J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation", *Proc. of the IEEE, Conf. of Robotics and Automation*, Sacramento, California, pp. 1394-1404, 1991.
- [3] E. Rimon & D. E. Koditschek, "Exact robot navigation using artificial potential functions", *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 5, pp. 501-518, October 1992.
- [4] Ge, S. S., Cui, & Y. J., "New potential functions for mobile robot path planning", *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 5, pp. 615-620, 2000.
- [5] G. C. Luh & W. W. Liu, "Dynamic mobile robot navigation using potential field based immune network", *Systemics, Cybernetics and Informatics*, Vol. 5, No. 2, pp. 43-50, 2005.
- [6] A. A. Masoud, "Solving the narrow corridor problem in potential field-guided autonomous robots", *Proc. of the IEEE, Inter. Conf. on Robotics and automation*, Barcelona, Spain, pp. 2909-2914, 2005.
- [7] J. Vascak, "Navigation of mobile robots using potential fields and computational intelligence means", *Acta Politechnica hungarica*, Vol. 4, No. 1, pp. 63-74, 2007.
- [8] V. O. S. Olunloyo & M. K. O. Ayomoh, "Autonomous mobile robot navigation using hybrid virtual force field concept", *European journal of scientific research*, Vol. 31, No 2, pp. 204-228, 2009.
- [9] H. Adeli, M. H. N. Tabrizi, A. Mazloomiam, E. Hajibour & M. Jaheb, "path planning for mobile robots using iterative artificial potential field method", *Inter. Journal of Computer Science Issues*, Vol. 8, Issue 4, No. 2, July 2011, pp. 28-32, July 2011.
- [10] M. Hamani & A. Hassam, "Mobile robot navigation in unknown environment using Improved APF method", *The 13<sup>th</sup> Inter. Arab Conf. on information Technology*, Zarqa University, Jordan, pp. 453-458, December 2012.
- [11] A. Mazur & M. Cholewinski, "Virtual force concept in steering mobile manipulators with skid-steering platform moving in unknown environment", *Journal of Robotic and Intelligent Systems*, Vol. 77, pp. 433-443, 2013.
- [12] J. Ni, W. Wu, J. Shen & X. Fan, "An improved VFF approach for robot path planning in unknown and dynamic environments", *Mathematical Problems in Engineering (Hindawi)*, Vol. 2014, ID 461237, PP. 1-10, 2014.
- [13] M. Hamani & A. Hassam, "A new method for mobile robot navigation in unstructured environments", *The 15<sup>th</sup> Inter. Arab Conf. on information Technology*, University of Nizwa, Oman, pp. 120-124, December 2014.

- [14] J. B. Mbebe, X. Huang, & M. Wang, "Fuzzy motion planning among dynamic obstacles using artificial potential fields for robot manipulators", *Robotics and Autonomous Systems*, Elsevier, Vol. 32, pp. 61-72, 2000.
- [15] L. Chengqing, M. H Ang Jr, H. Krishnan, & L. Ser Yong, "Virtual obstacle concept for local-minimum-recovery in potential- field based navigation", *Proc. of the IEEE, Inter. Conf. on Robotics and automation*, San Francisco, CA, pp. 983-988, 2000.
- [16] N. C. Tsourveloudis, K. P. Valavanis, & T. Hebert, "Autonomous vehicle navigation utilizing electrostatic potential field and fuzzy logic", *IEEE Trans. on Robotics and Automation*, Vol. 17, No. 4, pp. 490-497, 2001.
- [17] N. Achour & R. Toumi, "Stratégie de navigation par une approche potentiel-flou pour véhicules intelligents", *The 3<sup>rd</sup> Inter. Conf. on Sciences of Electronic, Technologies of Information and Telecommunications*, Tunisia, pp. 1-6, March 2005.
- [18] S. Ling, L. Rui, W. Weidong & D. Zhijiang, "Mobile robot real-time path planning based on virtual targets method", *Proc. of the IEEE, Third Inter. Conf. on Measuring Technology and Mechatronics Automation*, pp. 568-572, 2011.
- [19] S. Y. Kung & J. N. Hwang, "Neural Network Architectures for Robotic Applications, *IEEE Trans. on Robotics and Automation*", Vol. 5, No. 5, pp. 641-657, October 1989.
- [20] K. S. Narendra & K. P., "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 4-26, March 1990.
- [21] H. M. Tai, J. Wang & K. Ashenayi, "A neural Network-Based Tracking Control System, *IEEE Trans. on Robotics and Automation*", Vol. 39, No. 6, pp. 504-510, December 1992.
- [22] P. Payeur, H. Le-Huy & C. M. Gossolin, "Trajectory Prediction for Moving Objects Using Artificial Neural Networks", *IEEE Trans. on Industrial Electronics*, Vol. 42, No. 2, pp. 147-158, April 1995.
- [23] K. S., Y. I. & M. Ohtani, "Optimal State Feedback Control Law for Nonlinear Systems and its Best Approximation by a Neural Network", *Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics*, Vol. 4, pp. 1096-1101, 1999.
- [24] M. Asano, T. Yamamoto, T. Oki & M. Kaneda, "A Design of Neural-Net Based Predictive PID Controllers", *Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics*, Vol. 4, pp. 1113-1118, 1999.
- [25] Y. G. Leu, W. Y. Wang, & T. T. Lee, "H<sub>∞</sub>-Observer-Based Adaptive Fuzzy- Neural Control for a Class of Uncertain Nonlinear Systems", *Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics*, Vol. 1, pp. 449-454, 1999.
- [26] W. Chang, R. Hwang & J. Hsieh, "A Single Auto-Tuning Neural Controller for Nonlinear Dynamic Systems", *Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics*, Vol. 1, pp. 461-465, 1999.
- [27] M. Saito & J. Sugi, "A Hierarchical Bayesian Scheme for Nonlinear Dynamical System Reconstruction and Prediction with Neural Nets", *Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics*, Vol. 4, pp. 1119-1124, 1999.

- [28] S. X. Yang & M. Meng, "A neural Network Approach to Real-Time Motion Planning and Control of Robot Manipulators", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 674-679, October 1999.
- [29] S. X. Yang & M. Meng, "Real-time Collision-free Path Planning of Robot Manipulators using Neural Network Approaches", Autonomous Robots, Vol. 9, No. 1, pp. 27-33, 2000.
- [30] D. E. Goldberg, "Algorithmes Génétiques : Exploration, Optimisation et Apprentissage Automatique", Edition Addison Wesley France, 1994.
- [31] G. Winter, J. Periaux, M. Galan & P. Cuesta, "Genetic Algorithms in Engineering and Computer Science", British library Cataloguing in Publication Data, 1995.
- [32] G. C. & B. Chakraborty, "Ideal Marriage for Fine Tuning in GA", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 631-636, 1999.
- [33] T. N., K. Sugawara & I. Yoshihara, "A Motion Planning Method for a Hyper Multi-joint Manipulator using Genetic Algorithm", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 645-650, 1999.
- [34] K. T. H. Tamaki, "A Continuous Age Model of Genetic Algorithms Applicable to Optimization Problems with Uncertainties", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 637-642, 1999.
- [35] Q. C. Meng & Z. Chen, "Genetic Algorithms Encoding Study and A sufficient Convergence Condition of GAs", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 649-652, 1999.
- [36] N. A. Shiltagh & L. D. Jalal, "Path planning of intelligent mobile robot using modified genetic algorithm", Inter. Journal of Soft Computing and Engineering, Vol. 3, Iss. 2, pp. 31-36, May 2013.
- [37] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision process", IEEE Trans. on Systems, Man and Cybernetics, Vol. 3, pp. 28-44, 1973.
- [38] E.H. Mamdani & S. Assilian, "Application of fuzzy algorithms for control of simple dynamic plant", Proc. of the Institute of Electrical Engineers, pp. 1585-1588, 1974.
- [39] M. Sugeno & M. Nishida, "Fuzzy control of a model car", Fuzzy Sets and Systems, Elsevier, North Holland, pp. 103-113, 1985.
- [40] B. B. Meunier, "La logique floue et ses applications", Edition Addison Wesley France, SA, 1995.
- [41] Lo & Kuo, "Decoupled Fuzzy Sliding-Mode Control", IEEE Trans. on Fuzzy Systems, Vol. 6, No. 3, pp. 217-222, August 1998.
- [42] C. Kung & T. H. Chen, "Fuzzy Sliding Mode Controller Design Based on Fuzzy Union Reasoning", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 3, pp. 320-325, 1999.

- [43] A. Hassam & M. Hamani, "Motion control of non-holonomic mobile manipulator using fuzzy logic", The 12<sup>th</sup> Inter. Arab Conf. on information Technology, Naif Arab University for security Science, Saudi Arabia, pp. 1-6, December 2011.
- [44] W. Li, "Fuzzy logic based robot navigation in uncertain environments by multisensor integration", Proc. of the 1994 IEEE Inter. Conf. on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, pp. 259-265, October 1994.
- [45] E. Tsang & D. S. Yeung, "Optimizing Fuzzy Knowledge Base by Genetic Algorithms and Neural Networks", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 3, pp. 367-371, 1999.
- [46] Y. T. & Y. Dote, "Novel Fuzzy-Neural Network with General Parameter Learning Applied to Sliding Mode Control Systems", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 376-379, 1999.
- [47] T. Morimoto, K. Tu & Y. Hachimoto, "A Decision and Control Technique Based on Fuzzy Control, Neural Networks, and Genetic Algorithms For Optimization of a Fruit-Storage Process", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 6, pp. 440-445, 1999.
- [48] A. Thammano, "A New Forecasting Approach with Neuro-Fuzzy Architecture", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 386-389, 1999.
- [49] K. Kiguchi, H. H. Jang & T. Fukuda, "Identification of Robot Manipulators Using Neural Networks and Genetic Programming", Proc. IEEE Inter. conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 802-806, 1999.
- [50] A. Okatan & G. M. Dimirovski, "Fuzzy logic navigation and control of a non-holonomic vacuum cleaner", Proc.s of the 10<sup>th</sup> Mediterranean Conf. on Control and Automation, Lisbon, Portugal, July 2002.
- [51] D. P. Garg & M. Kumar, "Genetic algorithm based PD control and fuzzy logic control of a two link robot", Proc. of IMECE'02 Inter. Mechanical Engineering Congress & Exposition, New Orleans, Louisiana, pp. 1-7, November 2002.
- [52] F. Cuesta, A. Ollero, B. C. Arrue & R. Brauningl, "Intelligent control of nonholonomic mobile robots with fuzzy perception", Fuzzy Sets and Systems, Elsevier, Vol. 13, pp. 47-64, 2003.
- [53] Z. Hendzel, "Fuzzy reactive control of wheeled mobile robot", Journal of Theoretical and Applied mechanics, Vol. 42, No. 3, pp. 503-517, Warsaw 2004.
- [54] S. M. Lee, K. Y. Kwon & J. Joh, "A Fuzzy logic for autonomous navigation of marine vehicles satisfying COLREG guidelines", Inter. Journal of Control, automation and Systems, Vol. 2, No. 2, pp. 171-181, June 2004.
- [55] F. Abdessemed, K. Benmahammed, & E. Monacelli, "A fuzzy-based reactive controller for a non-holonomic mobile robot", Robotics and Autonomous Systems, Elsevier, 47, pp. 31-46, 2004.
- [56] A. Fatmi, A. Alyahmadi, L. Khriji & N. Masmoudi, "A fuzzy logic based navigation of a mobile robot", World Academy of Science, Eng. and technology, Vol. 22, 169-174, 2006.

- [57] F. Le Menn, P. Bidaud & F. Ben Amar, "Genetic differential kinematic modeling of articulated multi-monocycle mobile robots", Proc. of the 2006 IEEE Inter. Conf. on Robotics and Automation, Orlando, Florida, pp. 1505-1510, May 2006.
- [58] O. Castillo, L. T. Aguilar & S. Cardenas, "Fuzzy logic Tracking control for unicycle mobile robot", Engineering Letters, Vol. 13, No. 2, pp. 1-5, August 2006.
- [59] A. Hassam, K. Benmahammed & M. Hamani, "Motion Planning and Control of Mobile Manipulators", Asian Journal of Information Technology, Medwell Online, Vol. 5, No. 9, pp. 938-943, 2006.
- [60] L. Astudillo, O. Castillo, P. Melin, A. Alanis, J. Soria & L. T. Aguilar, "Intelligent control of an autonomous mobile robot using Type-2 Fuzzy logic", Engineering Letters, Vol. 13, No. 2, pp. 1-5, August 2006.
- [61] F. Cupertino, V. Giordano, D. Naso & L. Delfine, "Fuzzy control of a mobile robot: Implementation using a MATLAB-Based rapid prototyping system", IEEE robotics and Automation Magazine, pp. 74-81, December 2006.
- [62] Y. Lu, B. L. LU & Z. F. Wu, "Hierarchical fuzzy filter method for unsupervised feature selection". Journal of Intelligent and Fuzzy Systems, Vol. 18, pp. 157-169, 2007.
- [63] C. C. Wong, H. Y. Wang & S. A. Li, "PSO-based motion fuzzy controller design for mobile robots, Inter. Journal of Fuzzy Systems", Vol. 10, No 1, pp. 284-292, March 2008.
- [64] V. K. Banga, R. Kumar & Y. Singh, "Fuzzy-Genetic optimal control for four degree of freedom robotic arm movement", World Academy of Science, Engineering and technology, Vol. 60, 489-492, 2009.
- [65] O. Hachour, "The proposed fuzzy logic navigation approach of autonomous mobile robots in unknown environments", Inter. journal of Mathematical Models and methods in Applied Sciences, Vol. 3, Iss. 3, pp. 204-218, 2009.
- [66] A. R. Jasim, "A fuzzy logic control for autonomous mobile robot", Master thesis of Science in Mechronics Engineering, University of Baghdad, Republic of Iraq, 2009.
- [67] K. K. Tahboub & M. S. N. AL-Din, "A neuro-fuzzy reasoning system for mobile robot navigation", Jordan Journal of Mechanical and industrial, Vol. 3, No. 1, pp. 77-88, March 2009.
- [68] O. Obe & I. Dumitrache, "Fuzzy control of autonomous mobile robot", U.P.B. Sci. Bull., Series C, Vol. 72, Iss. 3, pp.173-186, 2010.
- [69] R. Rashid, I. Elamvazuthi, M. Begam & M. Arrofiq, "Fuzzy- based navigation and control of a non-holonomic mobile robot", Journal of computing, Vol. 2, Issue 3, pp. 130-137, March 2010.
- [70] A. Hassam & M. Hamani, "Combined control for non-holonomic mobile manipulator using fuzzy logic and genetic algorithms", 1<sup>st</sup> Inter. Symposium on Computing in Science & Engineering, Gediz University, Izmir Turkey, pp. 205, June 2010.
- [71] Z. M. Ma & L. Yan, "A literature overview of fuzzy conceptual data modeling", Journal of Information Science and engineering, Vol. 26, pp. 427-441, 2011.

- [72] N. Daei, H. M. Shirazi, R. Askari, & M. Ghanavati, "Service robot navigation based on Q-learning and fuzzy logic", *Inter. Journal of Robots, Education and Arts*, Vol. 1, No. 2, pp. 1-9, 2011.
- [73] L. Cherroun, R. Mechgoug & M. Boumehrez, "Path following behavior for an autonomous mobile robot using fuzzy logic and neural networks", *Courrier du savoir*, No. 12, pp. 63-70, October 2011.
- [74] S. Munaf & N. Al-Din, "Decomposed fuzzy controller for reactive mobile robot navigation", *Inter. Journal of Soft Computing and Engineering*, Vol. 2, No.4, pp. 140-149, 2012.
- [75] P. Nattharith, "Behavior modulation using fuzzy logic control for mobile robot navigation", *Trans. on Fuzzy Logic and Mobile Robot, Global Journal of Technology and Optimization*, Vol. 3, pp. 1-6, 2012.
- [76] N. Beklouti, R. John & A. M. Alimi, "Interval type-2 fuzzy logic control of mobile robots", *Journal of Intelligent Learning Systems and Applications*, Vol. 4, pp. 291-302, 2012.
- [77] N. Hassan S. & Deepika S., "Fuzzy based mobile robot navigation in an unknown environment", *Inter. Conf. on computing and Control engineering*, April 2012.
- [78] O. Obe & I. Dumitrache, "Adaptive neuro-fuzzy controller with genetic training for mobile robot control", *Inter. Journal of computers, Communications and control*, Vol. 7, No. 1, pp. 135-146, 2012.
- [79] M. I. Hamzah, & T. Y. Abdalla, "Mobile robot navigation using wavelet network and fuzzy logic", *Inter. Journal of Computer Applications*, 2013, 79(10), pp. 4-10.
- [80] H. Ramdane, M. Faisal, M. Algabri & Khalid Al-Mutib, "Mobile robot navigation with obstacle avoidance in unknown indoor environment using MATLAB", *Inter. Journal of Computer Science and network*, Vol. 2, Iss. 6, pp. 25-31, December 2013.
- [81] M. Algabri, H. Mathkour & H. Randane, "Mobile robot navigation and obstacle avoidance using ANFIS in Unknown environments", *Inter. Journal of Computer Applications*, Vol. 91, No. 14, pp. 36-41, April 2014.
- [82] L. Ting, C. J. B. Macnab, & S. Magierowski, "Reactive robot navigation utilizing nonlinear control", *Inter. Journal of Advanced Robotic Systems*, Vol. 100, No. 11, pp. 1-10, 2014.
- [83] K. B. Cho & S. Y. Cho, "The concept of collision-free motion planning using a dynamic collision map", *Inter. Journal of Advanced Robotic Systems*, Vol. 145, No. 11, pp. 1-9, 2014.
- [84] A. Pandey & D. R. Parhi, "Matlab simulation for mobile robot navigation with hurdles in cluttered Environment using minimum rule based fuzzy logic controller", *2<sup>nd</sup> Inter. Conf. on Innovations in Automation and Mechatronics Engineering, Procedia Technology, Elsevier*, Vol. 14, pp. 28-34, 2014.
- [85] B. Fatma, "Contribution des outils de l'intelligence artificielle dans la robotique mobile", *Thèse de doctorat du Département d'Informatique de l'Université d'Oran 1*, 2014.
- [86] F. Abdessemed, M. Faisal, M. Emmadeddine, R. Hedjar, K. Al-mutib, M. Alsulaiman & H. Mathkour, "A Hierarchical fuzzy control design for indoor mobile robot", *Inter. Journal of Advanced Robotic systems*, Vol. 11, No. 33, pp. 1-16, 2014.

- [87] H. Kang, C. Hyun & S. Kim, "Robust tracking control using fuzzy disturbance observer for wheeled Mobile robots with skidding and slipping", *Inter. Journal of Advanced Robotic systems*, Vol. 11, No. 75, pp. 1-11, 2014.
- [88] I. Chaari, A. Koubaa & S. Trigui, "SmartPath: An efficient hybrid ACO-GA Algorithm for solving the global path planning problem of mobile robots", *Inter. Journal of Advanced Robotic systems*, Vol. 11, No. 94, pp. 1-15, 2014.
- [89] M. Hamani & A. Hassam, "Combining planned and reactive controls to improve mobile robot navigation in unknown environments", *Inter. Electrical and computer Engineering Conf.*, University of Ferhat Abbes Setif1, Algeria, May 2015.
- [90] P. Lucidarme, "Apprentissage et adaptation pour des ensembles de robots réactifs coopérants", *Thèse de doctorat Montpellier II, France*, 2003.
- [91] F. Abdessemed, E. Monacelli & K. Benmahammed, "A learning paradigm for motion control of mobile manipulators", *Inter. Journal of Application Math. Computer Science*, Vol. 16, No. 4, pp. 475-484, 2006.
- [92] L. Khriji, F. Touati, K. Benhmed & A. Al-Yahmedi, "Mobile robot navigation based on Q-learning technique", *Inter. Journal of Advanced Robotic systems*, Vol. 8, No. 1, pp. 45-51, 2011.
- [93] W. li, J. Farrell, F. M. Wahl & K. R. Kozlowski, "A nonholonomic mobile robot navigation in uncertain environments based on behavior control", *World Automation Congress 8<sup>th</sup> Inter. Symposium on Robotics with Applications*, Maui, Hawaii, pp. 1-8, June 2000.
- [94] M. S. Guzel & R. Bicker, "A behavior-based architecture for maples navigation using vision", *Inter. Journal of Advanced Robotic systems*, Vol. 9, No. 18, pp. 1-13, 2012.
- [95] H. Mo, Q. Tang & L. Meng, "Behavior-based fuzzy control for mobile robot navigation", *Mathematical Problems in Engineering (Hindawi Publishing Corporation)*, Vol. 2013, ID 561451, pp. 1-10, 2013.
- [96] Y. S. Lee & S. B. Cho, "A hybrid system of hierarchical planning of behavior selection networks for mobile robot control", *Inter. Journal of Advanced Robotic systems*, Vol. 11, No. 57, pp. 1-3, 2014.
- [97] M. Hamani & A. Hassam, "Improving mobile robot navigation by combining fuzzy reasoning and virtual obstacle algorithm", *Journal of Intelligent and fuzzy systems*, Vol. 30, No. 3, pp. 1499-1509, 2016.
- [98] P. Coiffet, "La robotique: principes et applications", *Edition Hermès*, 1986.
- [99] J. J. Craig, "Introduction to Robotics, Mechanics and Control", *Addison Wesley Publishing Company*, 1986.
- [100] K. S. Fu R. C. Gonzalez & C. S. G. Lee, "Robotics : Control, Sensing, Vision, and Intelligence", *Mc Gram-Hill Book Company*, 1987.
- [101] E. Dombre et W. Khalil, "Modélisation et commande des robots", *Edition Hermès Paris*, 1988.

- [102] A. Pruski, "Robotique Générale", Ellipses, 1988.
- [103] J. P. Laumond, "La robotique mobile", Hermes Sciences Europe Ltd, 2001.
- [104] Bernard BAYLE, "Robotique mobile", Thèse de doctorat, Ecole Nationale Supérieure de Physique de Strasbourg, Université de Strasbourg, France, 2008.
- [105] S. Furuno, M. Yamamoto & A. Mohri, "Trajectory planning of mobile manipulator with stability considerations", Proc. of the 2003 IEEE Inter. Conf. on Robotics and Automation, Taipei, Taiwan, pp. 3403-3408, September 2003.
- [106] A. De Luca, G. Oriolo & P. R. Giordano, "Kinematic modeling and redundancy resolution for nonholonomic Mobile manipulators", Proc.s of the 2006 IEEE Inter. Conf. on Robotics and Automation, Orlando, Florida, pp. 1867-1873, May 2006.
- [107] D. Filliat, "Cartographie et estimation globale de la position pour un robot mobile autonome", Thèse de doctorat de l'université Paris 6 Pierre et Marie Curie, France, 2001.
- [108] N. Morette, "Contribution à la navigation de robots mobiles : approche par modèle direct et commande prédictive", Thèse de doctorat de l'Institut Prisme, Equipe Systèmes Robotiques interactifs, Université d'Orléans, France, 2009.
- [109] O. Lefebvre, "Navigation autonome sans collision pour robots mobiles nonholonomes", Thèse de doctorat de l'Institut National Polytechnique de Toulouse, France, 2006.
- [110] V. Delsart, "Navigation autonome en environnement dynamique : Une approche par déformation de trajectoire", Thèse de doctorat Université de Grenoble, 2010.
- [111] P. J. Antsaklis & K. M. Passino, "Towards intelligent autonomous control systems : architecture and fundamental issues", Journal of Intelligent and Systems, Netherland, Vol. 1, pp. 315-342, 1989.
- [112] C. Ye & E. Tunstel, "Advances in field robotics and intelligent systems", Inter. Journal of Intelligent Control and systems, Vol. 13, No. 1, pp. 1-2, 2008.
- [113] S. Nurmaini & B. Tutuko, "A new classification technique in mobile robot navigation", Telkomnika, Vol. 9, No. 3, pp. 453-464, 2011.
- [114] E. Swere & D. Mulvaney, "Robot navigation using decision trees", Electronic Systems and Control Division Research, UK, pp. 15-17, 2003.
- [115] M. C. Hoy, "Methods for collision-free navigation of multiple mobile robots in cluttered environments", 2014.
- [116] E. Papadopoulos & J. Poulakakist, "Trajectory planning and control for mobile manipulator systems", 8<sup>th</sup> IEEE Proc. Mediterranean Conf. on Control & Automation, Patras, Greece, pp. 01-06, 2000.
- [117] T. Guesbaya, K. Benmahammed & A. Benali, "Planificateur de trajectoire avec évitement d'obstacle basé sur la méthode des contraintes pour robot mobile", Courrier du Savoir, No. 9, pp. 71-81, Mars 2009.

- [118] A. De Luca, G. Oriolo & P. R. Giordano, "Kinematic control of nonholonomic Mobile manipulators in the presence of steering wheels", Inter. IEEE Conf. on Robotics and Automation, Anchorage, Alaska, USA, pp. 1792-1798, 2010.
- [119] D. C. Coner, H. Chset & A. A. Rizzi, "Towards provable navigation and control of nonholonomically constrained convex-bodied systems", Proc. of the 2006 IEEE Inter. Conf. on Robotics and Automation, Orlando, Florida, pp. 4416- 4418, May 2006.
- [120] J. Plaskonka, "Different kinematic path following controllers for a wheeled mobile robot of (2,0) Type", Journal of Intelligent robot Systems, Vol. 77, pp. 481-498, 2013.
- [121] M. Ghanavati, "A new method of mobile robot navigation: Shortness null space", Journal of Advances in Computer Research, Vol. 3, No. 3, pp. 65-74, August 2012.
- [122] F. Aurenhammer, "Voronoi diagrams- a survey of a fundamental geometric data structure", ACM Computing Surveys, Vol. 23, No. 3, pp. 345-405, 1991.
- [123] N. Vandapel, J. Kuffner & O. Amidi, "Planning 3-d path network in unstructured environments", IEEE Int. Conf. on Robotics and Automation, pp. 4624–4629, 2005.
- [124] M. Erdmann & T. Lozano-Pérez, "On multiple moving objects", Algorithmica, Vol. 2, No. 4, pp. 477–521, 1987.
- [125] T. Fraichard, "Trajectory planning in a dynamic workspace: a state-time space approach", Advanced Robotics, Vol. 13, No. 1, pp. 75–94, 1998.
- [126] C.-T. Kim & J.-J. Lee, "Mobile robot navigation using multi-resolution electrostatic potential field", 2005.
- [127] M. Pivtoraiko & A. Kelly, "Generating near minimal spanning control sets for constrained motion planning in discrete state spaces", IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, 2005.
- [128] L. Kravaki, P. Svestka, J-C. Latombe & M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration space", IEEE Trans. on Robotic and Automation, Vol. 12, No. 4, pp. 566–580, 1996.
- [129] N. M. Amato & Y. Wu, "A randomized roadmap method for path and manipulation planning", Proc. IEEE Int. Conf. on Robotics and Automation, pp. 113–120, 1996.
- [130] S. A. Wilmarth, N. M. Amato & P. F. Stiller, Maprm: "A probabilistic roadmap planner with sampling on the medial axis of the free space", Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1024–1031, 1999.
- [131] R. Hedjar & M. Bounkhel, "Real-time obstacle avoidance for a swarm of autonomous mobile robots", Inter. Journal of Advanced Robotic Systems, Vol. 11, No. 93, pp. 1-12, 2014.
- [132] L. Jaillet & T. Simeon, "Path deformation roadmaps: Compact graphs with useful cycles for motion planning", Int. Journal of Robotics Research, Vol. 27, No. 11-12, 2008.

- [133] Jur P. van den Berg & Mark H. Overmars, "Roadmap-based motion planning in dynamic environment", IEEE Transactions on Robotics, Vol. 21, No. 5, 2005.
- [134] Anthony Stentz, "Optimal and efficient path planning for unknown and dynamic environments", International Journal of Robotics and Automation, Vol. 10, pp. 89–100, 1993.
- [135] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz & S. Thrun, "Anytime dynamic a\* : An anytime replanning algorithm", 2005.
- [136] F. Kuhne, W. F. Lages & J. M. G. Da Silva Jr., "Mobile robot Trajectory tracking using model predictive control", IEEE Lars, Sao Luis, pp. 1-7, September 2005.
- [137] Dave Ferguson & Anthony Stentz, "Anytime rrt", IEEE- RSJ Int. Conf. on Intelligent Robots and Systems, Beijing, China, Oct. 2006.
- [138] D. Hsu, J.-C. Latombe & R. Motwani, "Path planning in expansive configuration spaces", Int. Journal Computational Geometry and Applications, Vol. 9, pp. 495–512, 1999.
- [139] S. Carpin & G. Pillonetto, "Robot motion planning using adaptive random walks", IEEE Trans. on Robotics and Automation, Vol. 21, No. 1, pp. 129–136, 2005.
- [140] M. Dakulović, B. Dimitrijević & I. Petrović, "Avoiding deadlocks of mobile robots in narrow passages of environments populated with moving obstacles", 2011 IEEE/ASME Inter. Conf. on Advanced Intelligent Mechatronics, Budapest, Hungary, pp. 936-941, July 2011.
- [141] J. Velagic, B. Lacevic & N. Osmic, "Efficient path planning algorithm for mobile robot navigation with a local problem solving", IEEE 2006, pp. 2325-2330, 2006.
- [142] J. Minguez & L. Montano, "Nearness diagram navigation (nd): A new real time collision avoidance approach", In IEEE RSJ Int. Conf. on Intelligent Robots and Systems, 2000.
- [143] R.G. Simmons, "The curvature-velocity method for local obstacle avoidance", In IEEE Int. Conf. on Robotics and Automation, pp. 3375–3382, 1996.
- [144] D. Fox, Burgard W. & S. Thrun, "The dynamic window approach to collision avoidance", IEEE Journal on Robotics and Automation, Vol. 4, No. 1, 1997.
- [145] P. Fiorini & Z. Shiller, "Motion planning in dynamic environments using the relative velocity paradigm", IEEE Int. Conf. on Automation & Robotics, Vol. 1, pp. 560–566, 1993.
- [146] Oren Gal, Zvi Shiller & Elon Rimon, "Efficient and safe online motion planning in dynamic environments", Proc. of the Int. Conf. on Robotics and Automation, Kobe, Japan, May 2009.
- [147] Luis Martinez-Gomez & Thierry Fraichard, "Collision avoidance in dynamic environments : an ics-based solution and its comparative evaluation", Proc. of the Intl Conf. on Robotics and Automation, Kobe, Japan, May 2009.
- [148] Stephane Petti & Thierry Fraichard, "Safe Navigation of a Car-Like Robot in a Dynamic Environment", Proc. of the European Conf. on Mobile Robots, Ancona (IT), 2005.
- [149] S. Quinlan & O. Khatib, "Elastic bands : Connecting path planning and control", IEEE Int. Conf. on Robotics and Automation, Atlanta, GA (US), May 1993.

- 
- [150] R. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol. 2, No.1, pp.14-23, 1986.
- [151] R. C. Arkin, "Motor Schema-based Mobile Robot Navigation", *International Journal of Robotic Research*, Vol. 8, pp. 92-112, 1989.
- [152] J. Rosenblatt and D. W. Payton, "A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control", *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks, Washington DC*, Vol. 2, pp. 317-324, 1989.
- [153] A. Tuncer, M. Yildirim & K. Erkan, "A motion planning system for mobile robots", *Advances in Electrical and Computer Engineering*, Vol. 12, No. 1, pp. 57-62, 2012.
- [154] S. Ahmadzadeh & M. Ghanavati, "Navigation of mobile robot using the pso particle swarm optimization", *Journal of Academic and Applied Studies*, Vol. 2, No. 1, pp. 32-38, 2012.
- [155] U. A. Syed & F. Kunwar, "Cellular automata based real-time path-planning for mobile robots", *Inter. Journal of Advanced Robotic systems*, Vol. 11, No. 93, pp. 1-15, 2014.
- [156] S. Zhang, B. He, L. Ying, M. Li & G. Yuan, "Autonomous navigation with constrained consistency for C-ranger", *Inter. Journal of Advanced Robotic systems*, Vol. 11, No. 84, pp. 1-13, 2014.

**ملخص:** قمنا في هذه الأطروحة بتطوير تقنية جديدة للمساهمة في حل مشكل الإبحار الذاتي للإنسان الآلي في محيط غير معروف. اعتمدنا في تصميم هذه التقنية على دمج المنطق الغامض مع خوارزمي الحواجز الافتراضية لمعالجة الحدود الدنيا الناتجة عن وجود حواجز مقعرة وذلك بالتنسيق الفعال للأولوية بين الوظائف التفاعلية المتعددة مثل: الإبحار نحو الهدف، تجنب الاصطدام بالحواجز، تتبع الجدار و تجنب الوقوع في وضعيات الطوارئ. من أجل تحقيق هذا الهدف، قمنا بتركيب صف من المجسات فوق الصوتية لقياس المسافات بين الإنسان الآلي و الحواجز، استعملت هذه المسافات من طرف خوارزمي الحواجز الافتراضية من أجل حساب أهداف وسيطة لتعيين اتجاه الحركة المناسب لتجنب وقوع الإنسان الآلي في الحدود الدنيا (وضعيات الطوارئ)، كما استخدمنا المنطق الغامض في المراقبة الوظيفية للإنسان الآلي فقمنا بدمج دالات الانتماء لجميع الوظائف التفاعلية في مجال واحد لضمان انتقال ناعم بين مختلف هذه الوظائف و تناقص ملحوظ في الاهتزازات خاصة عندما ينتقل الإنسان الآلي عبر حواجز متقاربة. لإثبات فعالية التقنية المقترحة، قمنا بانجاز سلسلة من المحاكاة على إنسان آلي في محيطات غير معروفة تحتوي حواجز مقعرة.

**كلمات مفتاحية:** المنطق الغامض، التحكم الوظيفي، وضعية الطوارئ، خوارزمي الحواجز الافتراضية، الحدود الدنيا.

**Résumé:** Dans cette thèse, une nouvelle approche est développée pour contribuer à la résolution du problème de la navigation autonome d'un robot mobile dans des environnements inconnus. Cette approche est établie en combinant le raisonnement flou et l'algorithme des obstacles virtuels pour surmonter le problème des minima locaux créé par les obstacles concaves en coordonnant efficacement les priorités entre de multiples comportements réactifs tels que : la navigation vers la cible, évitement des obstacles, le suivi des murs et la prévention des situations d'urgence. Pour atteindre cet objectif, une rangée de capteurs ultrasoniques est montée sur le robot mobile fournissant les distances entre le robot et les obstacles. Ces distances sont employées par l'algorithme des obstacles virtuels pour calculer des cibles intermédiaires aidant à déterminer la bonne direction de mouvement pour éviter que le robot soit emprisonné dans un minimum local (situation d'urgence). Tandis que, le raisonnement flou est employé pour le contrôle comportemental du robot mobile. Tous les comportements réactifs sont intégrés dans un seul univers de discours pour garantir une transition douce entre les différents comportements particulièrement quand le robot se déplace à travers des obstacles étroitement alignés. De cette manière, les oscillations du robot sont significativement réduites. Pour montrer l'efficacité de l'approche proposée, une série de simulations a été conduite en employant des environnements arbitrairement construits comprenant des obstacles concaves.

**Mots-clés:** Logique flou, contrôle comportemental, situation d'urgence, algorithme des obstacles virtuels, minima locaux.

**Abstract:** In this thesis, a new approach is developed for contributing in solving the problem of autonomous mobile robot navigation in unknown environments. This approach is built upon combining fuzzy reasoning and virtual obstacle algorithm to overcome the local minimum problem encountered in presence of concave obstacles by efficiently coordinating priorities between multiple reactive behaviors such as goal reaching, obstacles avoiding, wall following and emergency situations preventing. To achieve this objective, an array of ultrasonic sensors is mounted on the mobile robot providing the distance information between the robot and obstacles. This distance information is used by the virtual obstacle algorithm to calculate some sub-goals for determining the good motion direction to avoid robot trap in local region (emergency situation), since the fuzzy reasoning is used for behavior control of the mobile robot. All the reactive behaviors are mapped into one universe of discourse to guarantee a smooth transition between them especially when the robot moves through closely spaced obstacles. In this manner, the robot oscillations are significantly reduced. To show the effectiveness of the proposed approach, series of simulations have been conducted by using arbitrarily constructed environments including concave obstacles.

**Keywords:** Fuzzy logic, Behavior Control, Emergency Situation, Virtual Obstacle Algorithm, Local Minima.