

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

*Option: Système d'Information et de Connaissances (S.I.C)*

*Thème*

Planification du trajectoire et évitement d'obstacle par les réseaux de neurones pour les robots mobile autonome

**Réalisé par :**

- Belkhouche Ismail

*Présenté le 26 Septembre 2011 devant le jury composé de MM.*

- Benetaallah M.A (Président)
- Benzian Y (Encadreur)
- Benmouna M (Examineur)
- Belabed A (Examineur)

Année universitaire: 2010-2011

## Remerciements

Je tiens à remercier Monsieur BENZIANE Mohammed Yaghmorasan, d'avoir accepté d'encadrer cette thèse. Je le remercie vivement pour son aide, son soutien, ses conseils et sa générosité. Son ouverture d'esprit, sa disponibilité et ses analyses pertinentes ont contribué à rendre cette recherche agréable et enrichissante.

Cela fut un grand privilège d'avoir monsieur BENTAALLAH M. A. en présidence du jury qu'il trouve ici mes remerciements les plus sincères.

Je tiens à remercier Monsieur BENMOUNA M. Pour le vif intérêt manifesté à ce travail et pour avoir accepté d'évaluer cette thèse en faisant partie du jury.

Je voudrais remercier très spécialement monsieur BELABED A. d'avoir accepté de participer à ce jury de mémoire.

Enfin, je tiens à remercier toutes les personnes qui m'ont communiqué leur savoir, leur savoir-faire et leur savoir-être et qui m'ont permis de me construire à travers ce travail de recherche.

Merci aux copains et amis pour m'avoir accompagné et soutenu durant ces années.

## Dédicace

Je dédie ce modeste travail :

A la mémoire de ma grande mère

A mes parents Fatima et Hachemi BELKHOUCHE

A mes oncles Boumediene et Hocine

A mes chères sœurs et frères

A tout mes amis

## Sommaire

DES TABLEAUX.....	12
des abréviations : .....	13
.1 Introduction : .....	19
.2 Définitions : .....	20
.2.1 Robot : .....	20
.2.2 Robotique: .....	20
.2.3 Bionique : .....	20
.2.4 Cybernétique : .....	20
.3 Le concept de tâche : .....	22
.4 L'évolution de la robotique : .....	24
.5 Types des robots : .....	27
.5.1 Les robots manipulateurs : .....	27
.5.2 Les robots mobiles : .....	28
.6 Conception et composants des robots : .....	30
.6.1 Conception et composants des robots mobiles .....	30
.7 Les capteurs, les actionneurs et le traitement de l'information : .....	32
.7.1 Les capteurs : .....	32
.7.1.1 Définition 1: .....	33
.7.1.2 Principe des capteurs: .....	33
.7.2 Les actionneurs : .....	34
.7.3 Le système informatique : .....	35
.7.3.1 Traitement des données: .....	35
.7.3.2 Les plans de mouvements : .....	36
.8 Applications de la robotique mobile : .....	36
.8.1 Exploration : .....	36

.8.2 Industries :	36
.8.3 Travail dans des environnements dangereux :	36
.8.4 Maintenances :	37
.8.5 Les pompiers :	37
.8.6 Les robots médicaux :	37
.8.7 Nanotechnologies :	37
.8.8 Les robots de la guerre :	37
.8.9 L'utilisation civile :	37
.9 Conclusion :	38
.1 Introduction :	40
.2 Concept de planification :	41
.2.1 Les ingrédients de planification : [11]	41
.2.2 Définitions :	41
.3 Classe et techniques de planification :	42
.3.1 Espace de configuration :	42
.3.2 Planification discrète :	43
.3.3 Planification basée sur l'échantillonnage (sampling based planning) :	43
.3.3.1 Présentation :	43
.3.4 Planification combinatoire :	44
.4. Exemples d'algorithme de planification :	45
.4.1 Carte routière (road map) :	45
.4.2 Décompositions en cellules :	46
.4.3 Champ potentiel (Potentiel Field) :	47
.4.3.1 Amélioration de la méthode :	48
.5 Les algorithmes d'évitement d'obstacle :	48
.5.1 Algorithme du BUG :	49
.5.2 Algorithme du BUG 2 :	51

.5.3 Histogramme du champ de vecteur : .....	54
.6 Conclusion : .....	55
.1 Introduction : .....	57
.2 Définitions et caractéristiques : .....	57
.2.1 Définitions : .....	57
.2.1.1 Définition heuristique: .....	58
.2.1.2 Définition intelligence 1: .....	58
.2.1.3 Définition intelligence 2: .....	58
.2.1.4 Définition raisonnement: .....	58
.2.2 Caractéristiques des méthodes heuristiques : .....	58
.3 Les méthodes heuristiques : .....	59
.3.1 Système à base de connaissance : .....	59
.3.1.1 Présentation : .....	59
.3.1.2 Avantage : .....	61
.3.1.3 Inconvénient du système à base de connaissance : .....	61
.3.1.4 Planification et évitement d'obstacle par les systèmes à base de connaissances : .....	61
.3.1.5 Pour notre problème : .....	62
.3.2 Les algorithmes génétiques : .....	62
.3.2.1 Présentation : .....	62
.3.2.2 Avantage: .....	63
.3.2.3 Inconvénient: .....	63
.3.2.4 les algorithmes génétiques pour la planification de la trajectoire et évitement d'obstacles: .....	63
.3.3 La logique floue: .....	65
.3.3.1 Ensemble Flou : .....	66
.3.3.2 Composant d'un system de logique floue : .....	66
.3.3.3 La logique floue pour la planification de trajectoire et évitement d'obstacles : .....	67

.3.3.4 Discussion des travaux: .....	72
.4 Les autres méthodes intelligent:.....	73
.5 La fusion de plusieurs méthodes intelligentes:.....	74
.6 Conclusion:.....	74
.1 Introduction :.....	77
.2 Définitions et concepts de base :.....	77
.2.1 Le neurone biologique :.....	78
.2.2 Définitions :.....	80
.3 Propriété des réseaux de neurone: .....	82
.4 Architecture des réseaux de neurones : .....	82
.4.1 Réseau feedforward mono couche : .....	83
.4.3 Réseaux récurrents : .....	84
.5 Paradigme d'apprentissage :.....	85
.5.1 Définitions d'apprentissage de réseaux de neurone: .....	85
.5.2 Propriétés d'un algorithme d'apprentissage:.....	85
.5.3 Quelques algorithmes d'apprentissage:.....	86
.3.1 Apprentissage par correction d'erreur:.....	86
.5.3.2 Apprentissage basé sur la mémorisation: .....	87
.5.3.3 Apprentissage de Hebb :.....	87
.5.3.4 Apprentissage compétitif:.....	88
.5.4 Types d'apprentissage:.....	89
.5.4.1 Apprentissage supervisé:.....	89
.5.4.2 Apprentissage non supervisé:.....	89
.5.4.3 Apprentissage semi supervisé: .....	90
.4.4 Apprentissage par renforcement:.....	90
.6 Représentation de connaissance dans un réseau de neurones : .....	90
.6.1 Définitions de connaissances : .....	90

.6.2 Conception des données pour un réseau de neurones: .....	91
.7 Utilisation des réseaux de neurones pour la planification et l'évitement d'obstacles dans la robotique mobile : .....	92
.7.1 Implémentation d'approche classique par les réseaux de neurones: .....	93
.7.2 Création d'une méthodologie neuronale: .....	93
.7.3 Modélisation cognitive par les réseaux de neurones: .....	93
.7.4 Discussion : .....	94
.8 Conclusion : .....	95
.1 Introduction : .....	97
.2 Une méthode pour la planification de la trajectoire et évitement d'obstacle : .....	97
.2.1 Description détaillée des objectifs : .....	98
.2.2 L'approche : .....	99
.2.3 La méthode : .....	100
.2.3.1 Distribution de la tache cognitive : .....	101
.2.3.2 L'algorithme : .....	101
.3 Conceptions des agents : .....	102
.3.1 Reconnaissance de la nature de l'environnement : .....	102
.3.2 La détection de la région de la navigation : .....	104
.3.3 La représentation de l'environnement de la navigation : .....	109
.3.4 La navigation : .....	110
: dans certain cas où le robot existe dans les bornes sud, nord, est ou ouest les voisins sont restreints.	
.....	112
.4 Simulation et résultats : .....	112
.4.1 La reconnaissance de la nature de l'environnement : .....	112
.4.1 La reconnaissance de la région de navigation : .....	113
.4.2 le calcul de la position : .....	113
transformation ponctuelle est effectuée pour faciliter les calculs des x et des y, l'origine de repère sera dirigée du milieu de l'environnement vers le point de l'abscisse et l'ordonnée les plus petites, de cette manière tous les points auront des abscisses et des ordonnées positives.....	114



.4.3 la navigation : .....	114
.5 Conclusion : .....	115

## LISTE DES FIGURES

### Chapitre 1

Figure I.1: Aibo le robot sous forme d'un chien

Figure I.2: Asimo le premier robot humanoïde

Figure I.3: L'évolution de la robotique avec le domaine d'application (à droite) et les requis de l'environnement (à gauche)

Figure I.4: Des Exemples de robot manipulateurs actuels

Figure I.5: HELPMATE est un robot utilisé dans les hôpitaux pour des tâches de transportation

Figure I.6: Principe de capteur

### Chapitre 2

Figure II.1: Graphe de visibilité (à gauche) et diagramme de Voronoi (à droite)

Figure II.2: Exemple de la méthode de décomposition en cellules (décomposition en cellules en haut, et graphe correspondant en bas)

Figure II.3: INTROS TO AUT MOB ROB SIGWART

Figure II.4: algorithme bug 1 ( $h_1$   $h_2$  point de contact  $L_1$   $L_2$  point de départ  $s$ ,  $t$  point de départ est d'arrivée)

Figure II.5: Algorithme BUG 2

Figure II.6: Exploration de la même région plusieurs fois par un robot qui utilise BUG 2

Figure II.7: Un cas où un robot fait un tour complet sur l'obstacle sous BUG 2

Figure II.8: (à gauche) Environnement d'expérience  
(à droite) L'histogramme correspondant

### Chapitre 3

Figure III.1: Diagramme expliquant le processus de construction d'un système à base de connaissances

Figure III.2: Système de logique floue

Figure III.3: schème de contrôle basé sur le comportement

### Chapitre 4

Figure IV.1: Neurone biologique

Figure IV.2: Le neurone artificiel générique

Figure IV.3: Un réseau de neurone feedforward monocouche ( $X_{ij}$   $W_{ij}$  le vecteur d'entrée,  $f$  la fonction d'activation est  $Y_j$  le vecteur de sortie)

Figure IV.4: Un réseau de neurones feedforward multicouche

Figure IV.5: Un réseau de neurone récurrent monocouche

## **Chapitre 5**

Figure V.1: Méthode de calcul de diagonale

Figure V.2 : Sous ensembles flous d'un environnement de navigation

Figure V.3: Sous ensembles flous pour  $x$  dans le cas ou l'environnement petit

Figure V.4: Sous ensembles flous pour  $x$  dans le cas ou l'environnement moyen

Figure V.5: Sous ensembles flous pour  $x$  dans le cas ou l'environnement grand

Figure V.6: Sous ensembles flous pour  $y$  dans le cas ou l'environnement petit

Figure V.7: Sous ensembles flous pour  $y$  dans le cas ou l'environnement moyen

Figure V.8: Sous ensembles flous pour  $y$  dans le cas ou l'environnement grand

Figure V.9: Exemple de représentation d'un environnement

Figure V.10: Voisinage de la position courante ( $x, y$ )

## **LISTE DES TABLEAUX**

### **Chapitre 3 :**

Tableau III.1: Environnement 10×10 sans présence d'obstacles

Tableau III.2: Environnement 100×100 sans présence d'obstacles

Tableau III.3: Environnement avec présence d'obstacles (complexité moyenne)

Tableau III.4: Environnement avec présence d'obstacles (complexité élevée)

Tableaux III.5 : Base de règles floue pour la classification du terrain

Tableau III.6: La base de règles floues pour le contrôleur FLC1

Tableau III.7: La base de règles floues pour le contrôleur FLC2

Tableau III.8: La base de règles floues pour le contrôleur FLC3

### **Chapitre 4 :**

Tableau IV 1: Quelques exemples de fonction d'activation

### **Chapitre 5 :**

Tableau V.1: Règles floues pour la région de navigation (petite environnement)

Tableau V.2: Ensemble des cas possibles avec l'action correspondant.

## Liste des abréviations :

**IA:** Intelligence artificielle

**W:** Espace de travail

**R:** l'ensemble des nombre réels

**q:** configuration

**C:** Espace de toutes les configurations possibles

**O:** Région d'obstacles dans l'espace de travail

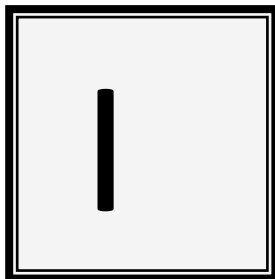
**F<sub>a</sub>(x):** Fonction d'appartenance (logique floue)

**X<sub>i,j</sub>:** Vecteur des poids synaptique pour un réseau de neurone

**F(x<sub>i,j</sub>, w<sub>i,j</sub>):** Fonction d'activation utilisé dans un RN

**RN:** Réseau de Neurones

## Chapitre



**Introduction générale**

# Introduction générale

---

## Introduction générale

Depuis l'antiquité, l'être humain essaye de découvrir et inventer des objets qui peuvent rendre sa vie plus facile. L'expérience qu'il a cumulé de la réalisation de différentes expérimentations, essais et observations constitue un ensemble de règles, de lois, et de méthodes qui est décrit par des définitions sous différentes formes, généralement appelée connaissance. Un ensemble de connaissances qui traite un domaine spécifique est appelé science.

Une comparaison préliminaire entre les sciences anciennes et modernes nous conduit à déduire que l'évolution des sciences modernes est exponentielle par rapport aux anciennes sciences, et ceci grâce aux différentes méthodes et outils dédiés à la recherche scientifique moderne.

Différentes méthodes qui servent à rendre la recherche scientifique plus facile en assurant la qualité des résultats obtenus sont développées. Ces méthodes génériques peuvent être appliquées à la résolution de différentes problématiques dans tous les domaines de la science. Les plus modernes de ces techniques sont les techniques issues de l'intelligence artificielle. Les différentes techniques de l'intelligence artificielle ont pour objectif de simuler l'intelligence humaine, d'où la machine devient capable de résoudre des problèmes plus complexes de la science et de l'ingénierie d'une manière intelligente.

Aujourd'hui, il existe plusieurs méthodes d'intelligence artificielle. Ces méthodes sont développées pour atteindre un objectif commun: rendre la machine capable de résoudre des problèmes d'une manière intelligente.

L'autre objectif consiste à rendre la machine intelligente afin qu'elle gère elle-même ses décisions, ce qui était le rêve de l'être humain depuis très longtemps. C'est avec l'apparition du domaine de la robotique que ce rêve commence à devenir une réalité.

Un robot peut être considéré comme étant un environnement idéal pour tester la validité et l'efficacité des différentes techniques de l'intelligence artificielle. Il existe divers travaux qui ont pour objectif d'implémenter une solution pour les robots. Dans le cas le plus général une telle solution ne consiste pas à être implémentée une seule fois, il s'agit de découper la problématique en sous problèmes plus simples et implémenter une solution à chaque problème séparément. L'être humain opte généralement pour ce plan de travail afin d'arriver à des solutions de quelques problèmes de la robotique. Remarquons qu'il n'existe pas encore une solution définitive d'un problème spécifique dans le domaine de la robotique.

# Introduction générale

---

Les méthodes intelligentes sont considérées parmi les meilleures techniques qui permettent la résolution de différents types de problèmes. Leur domaine d'application est assez vaste puisqu'elles peuvent modéliser fortement l'intelligence humaine grâce à leurs propriétés.

## **Motivation :**

Les méthodes intelligentes sont largement utilisées pour résoudre différents types de problèmes. En ce qui concerne la robotique, l'utilisation de ces méthodes permet de définir une nouvelle compréhension d'un point de vue différent ; cela par :

- l'apprentissage est effectué grâce aux actions du robot, cela signifie que l'apprentissage sera plus naturel, donc plus proche de l'apprentissage chez l'être humain. C'est une nouvelle approche d'analyser l'apprentissage [1].
- le placement d'une ou plusieurs techniques à l'intérieur d'un robot peut être vu comme une distribution de tâche cognitive du robot. Cette technique va se charger de la résolution de problèmes du bas niveau, ce qui nous éclaire sur la perspective d'étudier la capacité des réseaux de neurone à résoudre des problèmes de cognition de plus haut niveau.

## **Problématique et objectifs :**

La robotique mobile est confrontée à plusieurs problèmes qu'on peut résoudre en utilisant des approches différentes. Un des problèmes majeurs de la robotique mobile est le problème de planification de trajectoire et l'évitement d'obstacles.

Le problème d'évitement d'obstacles et la planification de la trajectoire pour les robots mobiles autonomes est un problème classique. Il peut être défini comme suit:

Si nous connaissons la position initiale du robot mobile dans un environnement connu ou inconnu qui peut certes contenir des obstacles statiques ou dynamiques, et si on nous a bien collecté l'ensemble d'informations nécessaires sur les caractéristiques de cet environnement par le système sensoriel du robot sous une forme spécifiée, alors comment peut-on procéder si on veut déplacer ce robot mobile vers une nouvelle position en évitant les obstacles existants ?

Dans notre problématique on va traiter seulement le cas d'obstacles statiques en supposant que les informations sont déjà collectées et enregistrées sous une certaine forme spécifique.



# Introduction générale

---

L'objectif de cette étude est d'étudier les techniques de l'intelligence artificielle comme solution pour cette problématique et de découvrir les bénéfices de ces méthodes par rapport aux autres méthodes déjà développées pour la résolution de cette problématique.

## **Approche :**

Afin de connaître les bénéfices d'utilisation des techniques de l'intelligence artificielle par rapport aux autres méthodes une comparaison est effectuée. Cette comparaison consiste à étudier les différentes méthodes d'un côté, et à montrer les limites de ces approches d'un autre côté afin de pouvoir proposer une solution efficace pour surpasser ces limites. Cependant pour pouvoir enfin présenter la solution proposée, les approches utilisant les réseaux de neurones pour la planification de la trajectoire et l'évitement d'obstacles sont aussi étudiés. Enfin une simulation sera implémentée pour montrer à la fois les résultats théoriques de l'étude et affirmer l'efficacité de la méthode.

## **Organisation du mémoire :**

Dans ce travail on va commencer par étudier le domaine de la robotique. L'objectif est de comprendre ce domaine de recherche afin de mieux définir et identifier notre problématique. Cette partie sera présentée dans le premier chapitre. Nous allons citer quelques définitions, les concepts de base, l'évolution de la robotique, les composants d'un robot et les deux types de la robotique.

Afin d'effectuer une comparaison judicieuse, les différentes méthodes doivent être présentées. Ces méthodes sont réparties dans deux classes: les approches classiques, et les approches heuristiques [2]. Nous allons alors présenter chacune de ces deux classes respectivement dans les chapitres deux et trois ; notre objectif consiste alors de dresser l'état de l'art tout en essayant à comprendre d'une manière générale ces approches (classique ou heuristique). Connaître leurs avantages et leurs inconvénients est aussi indispensable pour arriver à des résultats pouvant nous rapprocher de plus en plus à trouver une meilleure solution.

Le quatrième chapitre est dédié totalement à la présentation des réseaux de neurones, les bases des méthodes neuronales telle que les fonctions d'activation, représentation des connaissances (les poids synaptique, la conception des données d'entrée) ainsi que l'apprentissage. L'objectif consiste à étudier la capacité des réseaux de neurones à la résolution de notre problème. Nous allons présenter quelques approches neuronales pour la planification dans le domaine de la robotique.

# Introduction générale

---

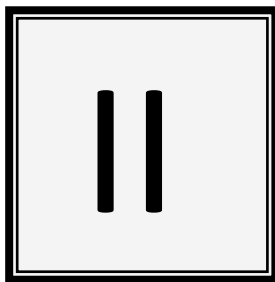
Dans le dernier chapitre, nous présenterons la simulation et les résultats. Ce chapitre a plusieurs objectifs : proposer une implémentation orientée objet en langage Java d'une ou plusieurs techniques de l'intelligence artificielle, utiliser cette implémentation pour implémenter la solution proposée, simuler la méthode proposée, et discuter les résultats de la simulation.

Afin de connaître les perspectives possibles de ce travail, les résultats théoriques et pratiques seront discutés dans la conclusion générale.

Le mémoire est organisé comme suit :

- état de l'art de la robotique : définitions, concepts de base, type des robots, évolution des robots, composants d'un robot.
- les méthodes classiques de planification et évitement d'obstacle pour les robots mobiles : définition, concept de base, présentation générale de quelques algorithmes.
- les méthodes heuristiques de planification et évitement d'obstacle pour les robots mobiles: définition, concept de base, présentation générale de quelques algorithmes.
- les réseaux de neurones et la planification et évitement d'obstacle pour les robots mobiles : définitions, neurone biologique, neurone artificiel, fonction d'activation, apprentissage, les réseaux de neurones et la robotique.
- simulation et résultats : une nouvelle méthode pour la planification de la trajectoire et évitement d'obstacle, conception et programmation, simulation et résultats.
- conclusion : discussions des résultats et perspectives.

**Chapitre**



**Etat de l'art de la robotique mobile**

## **I.1 Introduction :**

La robotique mobile est un nouveau champ d'ingénierie. Ses racines incluent plusieurs disciplines de la science et de l'ingénierie : génie mécanique, électrique et électronique, jusqu'à l'informatique, les sciences cognitives et sociales, la bionique et la cybernétique. Chacun de ces champs participe à sa manière à l'évolution de la robotique.

Le génie mécanique traite le côté mouvement et dynamique des robots. Cette discipline étudie aussi l'aspect énergétique. Son objectif est la mise en place d'un système mécanique complet.

Le génie électrique et électronique s'occupe par la conception et la mise en place d'un système électronique responsable de plusieurs tâches telle que la réception et l'émission des données grâce aux capteurs ou actionneurs, etc.

Le rôle de l'informatique est de développer une solution logicielle qui donne aux robots la capacité de traiter les informations, de les représenter et de les communiquer avec leur environnement (humain et autres robots). Cela consiste aussi à accomplir d'autres tâches telles que la planification de la trajectoire et l'aide à la décision.

Les sciences cognitives étudient l'aspect cognitif du robot. Ces sciences ont pour objectif d'évoluer les machines en fonction des connaissances.

Les sciences sociales s'intéressent à l'étude du comportement social du robot. Ces sciences interviennent dans la robotique appelée «humanoïde», la forme la plus évoluée de la robotique jusqu'à aujourd'hui.

C'est le besoin de l'être humain qui a fait apparaître les robots mobiles. Ces machines n'ont pas été inventées par hasard. Il y avait toute une évolution du plus simple au plus complexe, jusqu'à la création des robots intelligents. Ces robots sont équipés de matériels et logiciels. Le matériel est responsable de plusieurs tâches telle que la collecte d'information (capteur), le changement de la position (moteur, roues...) et participe aussi à l'exécution d'autres tâches.

La collecte d'informations est effectuée par les capteurs. Quant à l'exécution des tâches telles que le changement de la position, un actionneur intervient pour l'accomplir. Les actions d'un robot dépendent des informations collectées. Le système informatique est responsable du traitement d'information et de la prise de décision. Le rôle des actionneurs est de convertir la décision en action.

Dans ce chapitre on va définir quelques concepts de base de la robotique, en expliquant son évolution. On va entamer ensuite le sujet de la structure d'un robot pour expliquer son fonctionnement d'une manière générale.

## **I.2 Définitions :**

### ***I.2.1 Robot :***

C'est une machine capable de réagir à des entrées et de provoquer des modifications dans son environnement avec un certain niveau d'intelligence, idéalement sans intervention humaine. Les robots sont souvent conçus pour imiter des mouvements humains. On utilise souvent les robots dans l'industrie, dans la production automobile par exemple [3].

### ***I.2.2 Robotique:***

C'est une branche de l'ingénierie qui traite la création et la programmation des robots. Les roboticiens couvrent un champ très large de disciplines telles que la mécanique, l'électronique, la cybernétique, la bionique, et l'intelligence artificielle, dans le but de doter leurs créations d'un maximum de perception sensorielle, de dextérité physique, d'indépendance et de flexibilité [3].

Dans l'introduction nous avons cité brièvement comment les différentes branches de la science et de l'ingénierie participent à la création et au développement des robots et de la robotique. Nous avons déjà invoqué comment la mécanique, l'informatique et l'électronique participent à la conception des robots, mais pour la cybernétique et la bionique, l'image reste floue. Afin de mieux comprendre comment ces deux domaines participent au développement de la robotique, il convient de citer leurs définitions:

### ***I.2.3 Bionique :***

Elle englobe l'étude des organismes vivants, de leur caractéristiques, de leur fonctionnement en vue de créer du matériel qui simule ou duplique les activités d'un système biologique. La bionique recouvre également l'implémentation de prothèse ou de dispositifs électroniques dans les organismes vivants [3].

### ***I.2.4 Cybernétique :***

C'est l'étude des systèmes de contrôle comme le système nerveux, dans les organismes vivants, et développement des systèmes équivalents dans des appareils électroniques et

## Etat de l'art de la robotique

---

mécaniques. La cybernétique compare les similitudes et les différences entre les systèmes vivants et non vivants (que ces systèmes vivants et non vivants comprennent des individus, des groupes ou des sociétés). Elle est fondée sur les théories de communication et de contrôle qui peuvent s'appliquer dans les deux cas. Le terme cybernétique fut introduit en 1984 par Norbert Wiener dans son livre *Cybernetics Intelligence Artificielle* [3].

D'après les deux définitions de la bionique et de la cybernétique, on trouve que la bionique a pour objectif la mise en place des systèmes artificiels qui simulent un comportement naturel. Grâce à cette branche de la science, on peut développer des robots sous forme de chat, chien ou même sous forme d'un être humain. Des exemples de ces robots existent déjà (voire figure 1 et 2). La question qui se pose : est-ce qu'on peut développer un robot sans faire référence à une étude bionique ? La réponse sera oui, mais est-ce que ce robot va être capable de simuler un comportement naturel, la réponse sera non, d'où on trouve que la bionique intervient sous des formes les plus évoluées de la robotique, son vrai objectif sera l'amélioration du comportement du robot.



**Figure I.1:** Aibo le robot sous forme d'un chien [4]



**Figure I.2:** Asimo le premier robot humanoïde [5]

Concernant la cybernétique qui fait des études sur les différents systèmes de contrôle naturel, alors cette science s'occupe du développement d'un système de contrôle pour le robot qui est inspiré des systèmes naturels ; par conséquent, un tel système va être suffisamment «intelligent » pour contrôler un robot.

Les deux domaines, la bionique et la cybernétique participent fortement à l'évolution de la robotique, pour développer un robot qui répond à des exigences spécifiques de haut niveau. Il est nécessaire d'exploiter les connaissances offertes par ces deux domaines.

### **I.3 Le concept de tâche :**

La tâche que le robot doit accomplir est l'objectif de sa conception, c'est aussi le concept le plus important.

La caractéristique principale commune entre tous les robots mobiles est le changement de la position dans un environnement. Les différents robots effectuent des tâches différentes qui requièrent toutes un déplacement. Ce déplacement n'est pas aléatoire; il est effectué en suivant un chemin de navigation déterminé par un plan dessiné selon les informations venant de l'extérieur. L'environnement de navigation d'un robot peut être connu ou inconnu, tout dépend de la tâche principale à effectuer, dans les deux cas une condition nécessaire doit être

# Etat de l'art de la robotique

---

accomplie: elle consiste à éviter les obstacles éventuels qui se trouvent dans le chemin de navigation. Ce qui nous mène à dire que le concepteur d'une machine mobile doit prendre en considération deux choses :

- La tâche principale du robot mobile.
- La tâche d'évitement d'obstacles.

Pour réussir la deuxième tâche l'intérêt principal du robot mobile se fixe autour de l'évitement d'obstacles ce qui signifie une planification des zones d'environnement contenant des obstacles, sans prendre en considération les autres zones (vides d'obstacles)

On doit faire une planification du chemin qui doit prendre en considération la recherche d'un chemin acceptable en tenant compte des obstacles.

Ce qui conduit à conclure que les fonctions principales d'un robot mobile se résument à:

- La tâche principale du robot,
- La planification du chemin avec évitement d'obstacles.

La tâche à effectuer par le robot mobile nécessite parfois un mouvement d'un bras manipulateur, d'où l'existence du concept de la planification du mouvement. La planification de la trajectoire elle-même fait partie de la planification du mouvement, qui est le concept le plus général. Elle s'intéresse à la recherche et au développement d'une solution globale du mouvement et non seulement à la planification de la trajectoire.

Grâce à la planification du mouvement le robot mobile sera capable d'identifier et d'exécuter toutes les tâches qui requièrent un changement de la position. La mise en place d'une solution acceptable nous permettra d'atteindre nos objectifs.

La tâche de planification du mouvement elle-même peut être considérée comme une composition de plusieurs tâches qui nécessitent aussi l'exécution d'autres tâches. Pour réaliser une exécution réussie de la fonction objective du robot, ces tâches peuvent être identifiées également comme suit [15]:

- Mouvement
- Navigation



- L'interprétation d'informations sensorielles
- Localisation
- Planification
- Interaction avec les humains
- Evitement d'obstacles

Par l'exécution de ces tâches, le robot sera capable à répondre aux exigences imposées pour réussir sa mission. Les questions suivantes doivent avoir des réponses cohérentes: où suis-je? Ou devrais-je aller? Comment puis-je me déplacer de ma position pour atteindre la position finale désirée? Quelle information devrais-je échanger avec l'être humain et comment? Comment éviter les collisions? Le système informatique est responsable d'offrir une implémentation pour chacune des tâches citées ci-dessus. Nous devons bien sûr prendre en compte les limites du système physique du robot. Pour faciliter la compréhension nous citons à titre d'exemple un robot équipé d'une batterie de durée de vie d'une heure. Comme ce dernier ne peut pas fonctionner au delà d'une heure, nous ne pouvons pas alors lui attribuer une mission d'une durée supérieure à cette limite. Un autre exemple, dans le cas où on fait face à un système sensoriel faible, on doit être conscient du fait que le robot ne pourra pas accomplir une tâche qui nécessite une haute précision. Par ailleurs, si le robot est à roues, le planificateur ne doit pas commander le robot de faire des sauts.

## **I.4 L'évolution de la robotique :**

La robotique ne s'est pas faite subitement. Évoquer cette branche de la science revient à parler d'autres machines plus simples qui ont évolué au fil du temps pour donner naissance à ce qu'on appelle aujourd'hui la robotique. Les premières machines n'avaient aucune intelligence et se caractérisaient aussi par leur énorme taille. Elles étaient conçues généralement pour répondre au besoin de l'industrie. C'est l'être humain qui se chargeait de contrôler la façon avec laquelle ces machines devaient fonctionner, puisqu'elles étaient conçues pour suivre une loi mécanique simple qui ne requiert pas d'être sous le contrôle d'autres machines ou des programmes.

Avec l'apparition des premiers circuits électroniques les machines deviennent plus complexes car ces composants offrent la possibilité d'être programmés, le contrôle se fait alors d'une

## Etat de l'art de la robotique

---

nouvelle manière, généralement à l'aide des circuits intégrés à l'intérieur de la machine ou en utilisant des télécommandes.

L'évolution de ces machines et l'introduction de la notion de programmation des composants électroniques ont largement contribué à faire apparaître les premiers robots, qui ont trouvé leur application dans l'industrie, la télé robotique et d'autres applications simples. C'est en 1956 qu'une entreprise américaine a conçu le premier robot manipulateur qui était mis en service 1961 [6]. Ce type de robots fonctionne dans un environnement structuré.

Les nouveaux besoins de l'être humain ont fait évoluer la robotique.

Le nouveau type de la robotique est appelé la robotique avancée (Advanced Robotics) généralement, c'est là où on introduit la notion de la mobilité du robot.

Ce type de robotique trouve des domaines d'applications plus vastes et plus variés tels que l'exploration des planètes et des océans. Il exige alors de machines complexes et intelligentes.

Le désir de l'être humain pour profiter d'une vie meilleure a conduit à introduire un autre type de robotique appelé robotique de service (service robotics). Ces robots ont été conçus spécialement pour faciliter la vie de l'être humain. Ils peuvent accomplir les tâches qu'on fait quotidiennement; un exemple typique de ce type de robot est les robots de nettoyage des usines industrielles. Il faut noter que ces robots n'étaient pas conçus pour répondre au besoin d'une communauté précise (industrie, scientifique...), par conséquent, toutes les communautés peuvent bénéficier de leur service.

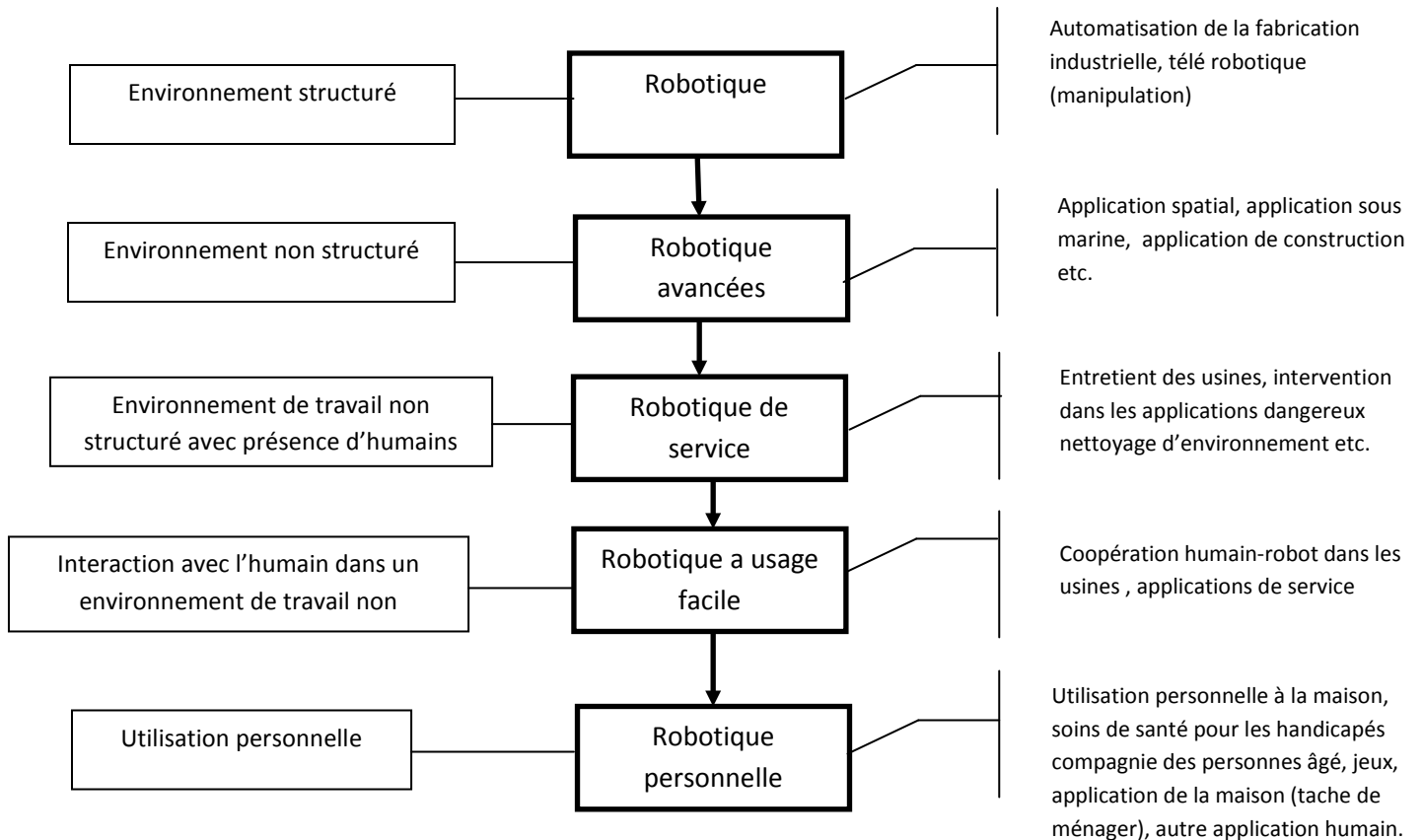
L'avant dernier type de la robotique est appelé « robotique à usage facile ». Ce type est très similaire au type précédent, mais il introduit en plus la notion de sécurité. Ces robots travaillent en collaboration avec l'être humain; ils sont donc en interaction directe avec lui; l'exemple typique qu'on peut citer illustre les robots qui servent à transporter la nourriture et les médicaments pour les patients dans un hôpital.

La forme la plus évoluée des robots concerne les robots personnels qui sont appelés aussi « humanoïde ». Ils sont conçus pour une utilisation personnelle. Leur principe consiste à simuler le comportement de l'être humain et sa façon de penser. Ce type de robot vient de faire sa naissance, d'où il rencontre plusieurs difficultés puisque les recherches dans ce domaine ne sont pas assez avancées. Son domaine d'application est très large : il peut servir à

# Etat de l'art de la robotique

aider les non voyants ou bien mis au service d'un handicapé. Les jeux consistent aussi un domaine d'application à ces robots. En effet ils peuvent aussi être utilisés à la maison pour effectuer des tâches ménagères.

Le schéma suivant explique bien l'évolution de la robotique



**Figure I.3:** L'évolution d la robotique avec le domaine d'application (à droite) et les requis de l'environnement (à gauche) [7]

## I.5 Types des robots :

On peut dire qu'il y a deux catégories des robots selon le critère de mobilité:

- Les robots manipulateurs,
- Les robots mobiles.

### *I.5.1 Les robots manipulateurs :*

Ce type de robot est conçu pour réaliser une **tâche** unique et indéfiniment **répétée** telle que l'assemblage des pièces dans une chaîne de production dans une usine, le montage de voitures par exemple. Ce type n'est pas mobile, le robot ne saura accomplir que la tâche pour laquelle il a été conçu.

La définition donnée aux robots manipulateurs industriels par la Fédération International de la Robotique sous le standard ISO/TR 8373 est la suivante :

Un robot industriel manipulateur est un contrôle automatique, reprogrammable, c'est un manipulateur multifonction programmable en trois axes ou plus, qui peut être soit fixé dans un endroit, soit mobile et il est destiné à des applications d'automatisation industrielles [8].

La définition qu'on vient de citer nous informe que ces robots étaient conçus essentiellement pour répondre aux besoins de l'industrie ; ils sont intelligents (parce que ils sont programmable) et capables d'assurer un mouvement en trois dimensions.



**Figure I.4:** Exemples de robot manipulateurs actuels [9]

Ici on veut citer quelques raisons pour utiliser les robots manipulateurs :

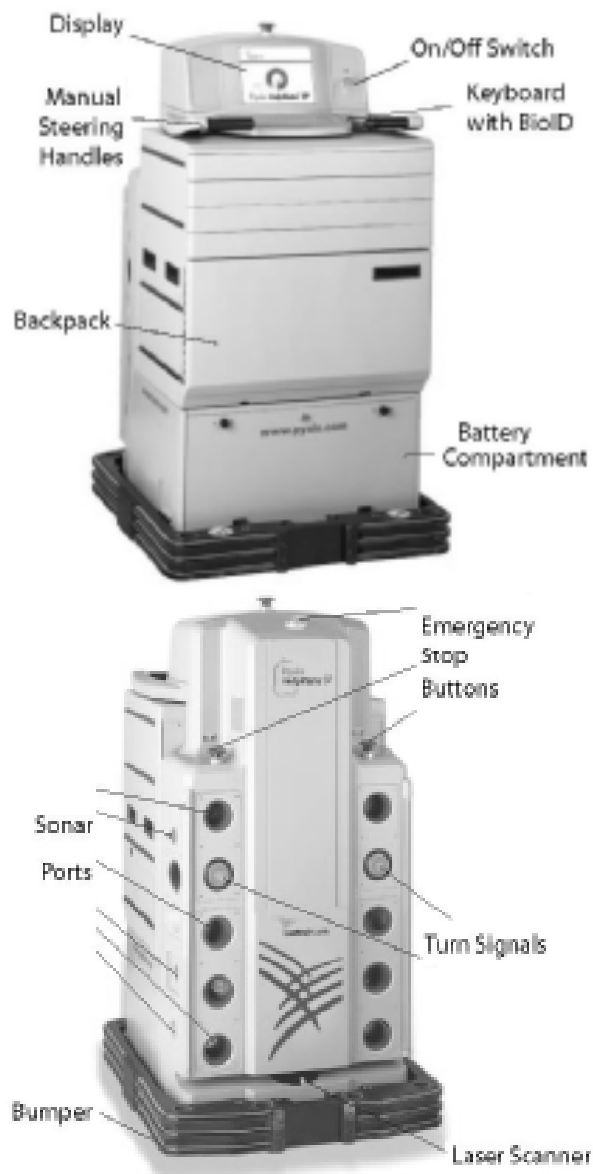
- Environnement dangereux qui présente beaucoup de risque pour la vie l'homme.
- Un travail répétitif et cyclique.
- Travail difficile à l'homme ou qui nécessite une haute précision (nano technologie).

Il existe d'autres raisons et situations dont il est préférable de se servir des robots manipulateurs: ils peuvent fonctionner sans arrêt, si nécessaire 24 h sur 24, et nécessitent un entretien minimal. Ils sont plus rapides que l'être humain et exécutent le travail en continu, d'une manière précise et constante. Ils n'entraînent jamais de dégâts et si un problème survient il sera certes d'origine humaine (défaut de conception, faute de programmation,...etc.). Seulement les robots ne sont pas encore prêts à remplacer l'humain parce que des nombreux obstacles techniques et sécuritaires se dressent encore ; par conséquent la présence de l'être humain est indispensable pour assurer le contrôle et la maintenance.

## ***1.5.2 Les robots mobiles :***

Ces robots exécutent des tâches qui nécessitent un déplacement dans un environnement. Ils sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens. Contrairement aux robots manipulateurs prévus pour travailler exclusivement dans des espaces connus et de manière répétitive, les robots mobiles sont destinés à évoluer de manière autonome dans des environnements peu ou non structurés.

Néanmoins, le concepteur doit augmenter considérablement ses connaissances sur la localisation et la navigation de systèmes autonomes et prendre en considération l'environnement du mouvement lors de la conception de la machine. On peut citer les robots explorateurs comme exemple de ce type.



**Figure I.5:** HELPMATE est un robot utilisé dans les hôpitaux pour des tâches de transportation

[10]

Classification des robots selon le critère de contrôle :

- Les robots autonomes.
- Les robots non autonomes.

## **I.6 Conception et composants des robots :**

Un robot est conçu pour répondre à des besoins bien précis de l'utilisateur. Désormais, les robots arrivent dans la maison, dans notre quotidien, c'est la diversité de nos besoins qui a fait varier les modèles de robots existants aujourd'hui.

### ***I.6.1 Conception et composants des robots mobiles***

Dans ce type de robots on se trouve face à plusieurs problématiques qui sont liées principalement au mouvement de la machine, à ses interactions avec son environnement externe et à la tâche à accomplir.

Analyser les problèmes liés au mouvement nous mène à étudier à la fois les problèmes liés à la cinématique et les problèmes liés à la dynamique.

Pour résoudre les problèmes de la cinématique, on doit tout d'abord analyser quelques contraintes telles que la vitesse, l'accélération et l'environnement de la machine pour établir ensuite un plan efficace qui sert à contrôler le mouvement de celle-ci selon l'environnement du travail où elle doit opérer.

Pour faciliter la compréhension des problèmes dynamiques, citons l'exemple des machines de transport de marchandises. Le plus souvent nous n'avons pas toujours affaire à transporter le même poids d'où il est nécessaire de prendre en considération ce changement du poids : un poids qui change implique obligatoirement le changement des variables de la loi dynamique, le concepteur de la machine mobile doit prendre en compte toutes les possibilités afin de bien gérer les situations du changement pour éviter les éventuels accidents de travail.

Une autre problématique se pose : celle de l'interaction de la machine avec son environnement externe. Le robot doit réussir sa mission quelque soit l'environnement : connu ou inconnu, statique ou dynamique ; c'est la raison pour laquelle le robot est équipé des capteurs responsables de collecter des données pour lui fournir les informations sur son environnement externe, et aussi d'un programme informatique qui lui permet de traiter ces données. La dernière tâche concerne la prise de décision : le robot prend la décision en se servant des résultats de traitement des données faites sur l'environnement. Cette décision est fortement liée à l'environnement et à la tâche à effectuer. Par conséquent le robot sera impliqué à changer sa direction du mouvement, sa vitesse, ou tout simplement, il va s'arrêter.

# Etat de l'art de la robotique

---

On va maintenant décrire les composants d'un robot mobile. Nous avons déjà dit que ces composants diffèrent d'un robot à un autre selon la tâche qu'ils doivent effectuer (c'est à dire le besoin qu'ils doivent satisfaire) mais généralement quelle que soit la fonction du robot, les composants de ce dernier appartiennent aux familles suivantes :

- Les capteurs (A)
- Les actionneurs (B)
- Une unité de traitement (microcontrôleur ou microprocesseur) (C)
- Une mémoire de stockage d'information (D)
- Des unités d'entrée/sortie (E)
- Une source d'énergie (batterie dans la plus part de cas) (F)

## *Remarque*

Les plus importantes différences entre les robots résident dans les capteurs et les actionneurs.

### **A. Les capteurs :**

Ce sont des composants qui permettent la collection des données sur l'environnement externe du robot, ils peuvent également être utilisés pour analyser et contrôler l'état interne de la machine (température, vitesse,...).

### **B. Les actionneurs:**

Le rôle des actionneurs consiste à exécuter la tâche pour laquelle la machine a été conçue.

Nous allons donner plus de détails sur les capteurs et les actionneurs dans une autre section.

### **C. L'unité de traitement :**

C'est l'emplacement physique où se font tous les traitements nécessaires au fonctionnement de la machine. Cette unité se charge d'effectuer les traitements, les calculs et les décisions.

### **D. Mémoire:**

Sert à stocker les informations venant des différents capteurs ainsi que les informations traitées.

### **E. Les unités d'entrée/sortie :**



Afin de pouvoir effectuer un échange d'information avec l'être humain ou avec d'autres machines, il est nécessaire d'équiper les robots des unités d'entrée-sortie.

## **F. Source d'énergie :**

Afin de pouvoir effectuer sa tâche, les actionneurs et les capteurs doivent être alimentés, dans le cas le plus général la source d'énergie est typiquement une batterie rechargeable.

### **Remarque**

Les unités d'entrée peuvent être aussi des capteurs et les unités de sortie peuvent aussi être des actionneurs.

## **I.7 Les capteurs, les actionneurs et le traitement de l'information :**

### ***I.7.1 Les capteurs :***

Afin de permettre au robot d'interagir librement avec son environnement, il faudra l'équiper de matériel nécessaire qui lui permet de connaître son environnement interne et externe. Ce matériel est appelé « capteur » ou « détecteur ». Les capteurs servent à collecter les données et fournir les informations nécessaires sur l'environnement du robot. Ils aident par conséquent les robots à bien connaître leur environnement. Dans le cas général on exploite une propriété physique du capteur, les capteurs ont une sensibilité vers les grandeurs physiques. Les grandeurs physiques existant dans un environnement peuvent être différentes et se mesurent par des différentes unités de mesures, d'où la nécessité de la mise en place d'un système de capteur qui permet une collecte de données plus complète sur l'environnement.

La mise en place d'un tel système nécessite une stratégie de fusion des différentes données capteurs, alors qu'on doit parfois résoudre des situations où il y a une ambiguïté entre les différentes données collectées.

Dans cette partie de travail, on va discuter les capteurs, leur principe de fonctionnement et de leurs propriétés.

## I.7.1.1 Définition 1:

Un capteur est un dispositif qui convertit un phénomène physique en un signal électrique [11]

## I.7.1.2 Principe des capteurs:

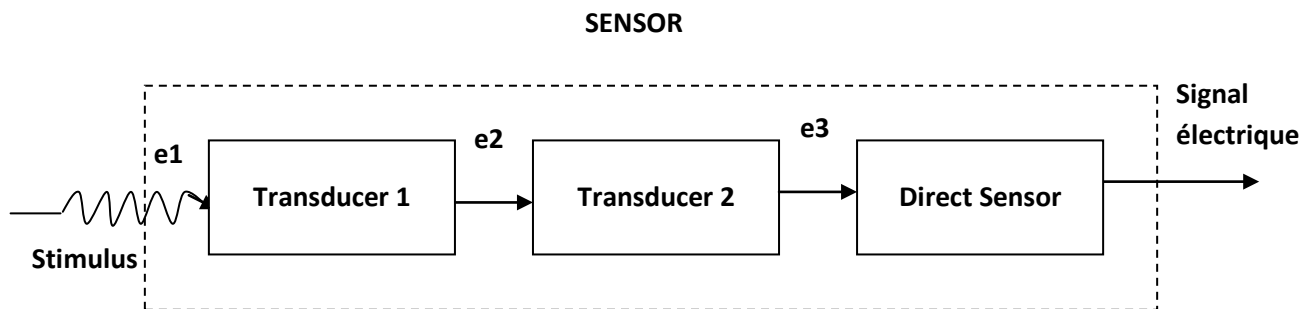


Figure I.6 : Principe de capteur [12]

Le schéma de la figure explique le principe de fonctionnement et décrit les différents composants d'un capteur. Nous commençons d'abord par définir le rôle de chaque composant, et par la suite décrire le principe de fonctionnement.

### Définitions :

Stimulus: c'est la quantité, la propriété ou la condition qui est perçue et convertie en signal électrique [12].

Transducteur (capteur): est un convertisseur d'un type d'énergie à un autre.

Capteur direct : un capteur direct sert à convertir un stimulus à un signal électrique ou de modifier un signal électrique en utilisant un effet physique approprié [12].

Nous devons distinguer la différence entre un transducteur et un capteur. Le premier est un convertisseur; il convertit l'énergie d'un type à un autre. Le capteur est aussi un convertisseur mais celui-ci convertit les différents types d'énergie à un signal électrique [12].

Pour un capteur simple il existe au moins un capteur direct. Un capteur complexe exige l'existence d'au moins un transducteur qui sert à convertir la grandeur physique (stimulus) à un autre type d'énergie, ce dernier va servir aussi d'entrée pour le prochain transducteur en capteur direct. Ce processus est répété jusqu'à obtenir le signal électrique désiré mais il faut noter que le capteur direct est toujours intégré dans la dernière position, c'est-à-dire c'est lui qui envoie le signal final de sortie en dehors du capteur.

Une question s'impose : peut-on utiliser seulement un capteur direct au lieu de plusieurs transducteurs ?

On peut répondre à cette question de deux façons [11]:

- On se sert d'un capteur direct pour des grandeurs physiques bien spécifiques,
- La mise en place d'un capteur direct serait coûteuse, alors on opte plutôt pour des capteurs complexes composés de « Transducteur » et non d'un « capteur direct » pour économiser les dépenses [11].

### ***1.7.2 Les actionneurs :***

Les actionneurs peuvent être définis comme suit: un actionneur peut être décrit comme l'opposé du capteur en général: il convertit le signal électrique à une énergie non électrique [12].

De la définition qu'on vient de citer on peut dire que les actionneurs consistent à réaliser la tâche inverse des capteurs; ils reçoivent un signal électrique et le convertissent en action.

La définition du mot action est large. Une action peut être un changement de la position, augmentation ou diminution de la vitesse, adaptation de la position de la caméra, allumage d'une lampe qui indique l'état du robot (en arrêt, en marche), un son d'alerte qui indique que la batterie est faible. Une action peut désigner aussi un envoi de données d'où on peut conclure qu'une action est un événement physique qui a un effet direct ou indirect sur le robot lui-même ou bien sur l'environnement externe du robot.

Des exemples typiques des actionneurs sont les moteurs électriques et les ventilateurs.

### ***1.7.3 Le système informatique :***

Afin de mieux traiter les données venant des capteurs et de les utiliser pour l'aide à la décision, il est nécessaire de développer un système informatique qui traite ces données. Ce système informatique va s'intéresser à la création d'un système de contrôle qui à son tour va se charger de résoudre un nombre de problèmes tels que le contrôle du mouvement, la localisation de la machine, la planification de la trajectoire et le traitement des données issues des capteurs. Dans les formes les plus évoluées de la robotique, d'autres problèmes se posent tels que la communication de données, l'interface homme-robot, etc. Ces problèmes doivent être résolus, ce qui conduit à développer des systèmes de contrôle plus complexes, et par conséquent, pour faire face à cette complexité, le processeur du robot est sensé d'effectuer plus de calculs qui peuvent lui entraîner des problèmes du bon fonctionnement. Pour éviter ce genre de complications nous devons soit améliorer les performances du processeur (une solution coûteuse) ou bien partager ces calculs sur plusieurs ordinateurs, ce qui vient de dire implémenter un système distribué [13].

Le système informatique doit résoudre les problèmes suivant :

#### **1.7.3.1 Traitement des données:**

Pour un traitement de données efficace, l'utilisation de différentes technologies informatiques de représentation, de communication et de la classification de données est nécessaire, un tel traitement de données va optimiser l'utilisation des ressources et donc va contribuer à faciliter la prise de décision.

Les chercheurs qui ont édité le document cité dans [14] ont proposé une implémentation d'un système de contrôle basé sur une architecture distribuée (CORBA).

Néanmoins, il faut noter que CORBA n'est pas la seule architecture distribuée existante sur le marché. Il existe d'autres propositions d'architectures distribuées ; on peut choisir alors une technologie parmi les différentes technologies existantes à condition qu'elle réponde le plus à nos besoins en robotiques. Sinon pour déterminer la meilleure d'entre-elles, une étude comparative basée sur plusieurs critères est indispensable.

Le problème avec les technologies distribuées appelées middleware c'est qu'elles sont conçues généralement pour les web et elles implémentent des langages spécifiques [15].

Ainsi, CORBA peut être considéré parmi les meilleures solutions possibles pour les applications en robotique [15].

L'utilisation d'autres technologies de représentation de données telles que XML et les EJB est aussi possible.

### **I.7.3.2 Les plans de mouvements :**

Le système informatique est responsable aussi de la planification du mouvement. Cette planification est établie en se servant du résultat de traitement de données effectué dans l'étape précédente. Dans cette partie le concepteur fait intervenir les différentes techniques de l'intelligence artificielle. Grâce à l'implémentation d'une stratégie de planification, le robot devient capable à prendre des décisions, il devient alors intelligent.

## **I.8 Applications de la robotique mobile :**

Les domaines d'applications des robots sont très variés et touchent fortement la vie quotidienne de l'être humain, nous pouvons citer les domaines d'application suivants [17].

### ***I.8.1 Exploration :***

NASA a utilisé les robots pour explorer les autres planètes, cela est dû à des raisons économiques, l'exemple typique de ce type de robot est Mars Pathfinder (Sojourner).

### ***I.8.2 Industries :***

Les robots sont largement utilisés dans l'industrie, l'assemblage des pièces électroniques ou des voitures, le nettoyage des usines industrielles, la peinture des surfaces, le transport des produits, etc. sont des exemples typiques d'application de robotique dans le domaine d'industrie.

### ***I.8.3 Travail dans des environnements dangereux :***

La nature du robot lui permet de travailler dans des environnements dangereux. Un exemple typique concerne les robots utilisés pour la désactivation des bombes, le nettoyage des environnements pollués par les produits chimiques ou même nucléaire. L'intervention de l'être humain dans de telles situations représente un grand risque pour sa santé et sa vie ; par

contre, l'utilisation du robot pour accomplir cette tâche n'a aucun risque sur la machine, tout en réduisant les dégâts et en sauvegardant la vie humaine.

### ***1.8. 4 Maintenances :***

Les robots peuvent être utilisés pour maintenir un système physique, un technicien peut utiliser un robot pour lui fournir des images via une caméra pour détecter et réparer le problème dans un temps réduit.

### ***1.8. 5 Les pompiers :***

Ils peuvent remplacer un humain dans la tâche de répression des feux ; ils ont la capacité de résister aux températures élevées. En plus, ils sont plus précis par rapport à un être humain.

### ***1.8. 6 Les robots médicaux :***

Dans le domaine de la médecine, on peut se servir de la présence d'un robot de plusieurs façons, on note à titre d'exemple l'aide dans des opérations médicales.

### ***1.8. 7 Nanotechnologies :***

Un exemple du robot assez minime réel n'existe pas encore, mais c'est une perspective; soit qu'il s'agisse d'intervenir dans des environnements qui le nécessitent, soit encore pour s'insérer dans des objets, des espaces ou des corps – les implants cardiaques ou rétiniens sont ou deviennent des robots un peu particuliers. À titre d'exemple on peut développer un robot qui peut être injecté dans le sang humain pour accomplir une tâche médicale telle qu'enlever les dépôts gras ou l'identification d'une tumeur et la destruction sélective de toutes les cellules cancéreuses.

### ***1.8.8 Les robots de la guerre :***

Un exemple typique concerne les missiles intelligents et les petits avions pour l'espionnage et les missions rapides.

### ***1.8.9 L'utilisation civile :***

Il y a un grand nombre d'application civile pour les robots : ils pénètrent le foyer familial en vue de prendre soin et assister des personnes âgées et les handicapés, de faire le ménage, la

cuisine, surveiller les enfants etc. Cela peut concerner même les animaux domestiques tels que le robot Aibo de Sony.

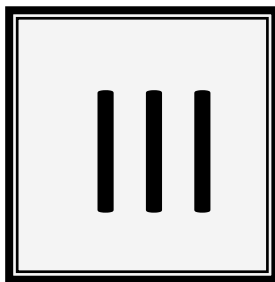
### **I.9 Conclusion :**

La robotique est un domaine qui est encore neuf mais il connaît une évolution assez importante. La robotique aujourd'hui est une industrie se chiffrant à plusieurs milliards de dollars [16]. Grâce à cette industrie, l'être humain a réussi la création des composants électroniques assez minimes, ce qui a contribué à la mise en œuvre des téléphones portables, et de l'ordinateur portable. La rapidité et la précision des robots permettent de produire une grande quantité de divers produits avec une qualité meilleure, chose que l'être humain ne peut pas réaliser dans le même laps de temps.

Les robots mobiles forment une industrie aussi importante ; grâce à eux on peut explorer les autres planètes, les océans et les utiliser pour diverses applications scientifiques, civiles et militaires.

D'après l'étude effectuée, on peut dire qu'un robot est un système combiné de hardware et de software capable de collecter, traiter les données et effectuer des tâches intelligentes dans un environnement physique.

## Chapitre



**Méthodes classique pour la planification du  
trajectoire et évitement d'obstacle**



## II.1 Introduction :

Plusieurs problèmes sont présentés par la robotique mobile moderne. Parmi ces problèmes on note la planification des mouvements et l'évitement d'obstacles. Ces deux problèmes sont liés directement car on ne peut pas implémenter une planification du mouvement sans prendre en considération l'évitement d'obstacles. Il ne peut pas y avoir de plan de navigation faisable sans évitement d'obstacles. D'un autre point de vue on trouve qu'un algorithme d'évitement d'obstacles est un petit plan de navigation proche d'une région contenant un ou plusieurs obstacles.

Un besoin fondamental en robotique est d'avoir un algorithme qui convertit la spécification haute niveau des tâches à une spécification bas niveau sur la manière du mouvement du robot ; un tel algorithme est appelé algorithme de planification. Le problème majeur d'un robot n'est qu'un ensemble de pièces électroniques et mécaniques qui composent un système ; sans logiciel un robot ne peut rien faire, nécessitant ainsi la mise en place de logiciels pour le rendre utile. Les logiciels qu'on doit installer offrent des fonctionnalités diverses et forment un ensemble d'un système logiciel complet, c'est-à-dire un système d'information pour le robot. Différents services sont fournis par ce système, à commencer par des tâches assez simples telles que le stockage de données, jusqu'aux tâches complexes et intelligentes telles que la planification sous des contraintes spécifiques.

Le thème planification peut avoir plusieurs significations selon les perspectives. La théorie de contrôle s'intéresse à la conception des entrées pour les systèmes physiques. Ces entrées sont décrites par des équations différentielles. Dans la théorie du contrôle moderne, le terme planification du mouvement se réfère à la construction des entrées pour les systèmes dynamiques non linéaires conduisant d'un état initial à un état final spécifié. Par contre, l'intelligence artificielle modélise la forme discrète d'un système. En plus de la planification des mouvements, la planification en IA peut désigner la résolution d'un puzzle ou bien la construction d'une tour (tour de Hanoï). Aussi, des problèmes modélisés avec des espaces continus peuvent être résolus en définissant une séquence d'actions. Dans la théorie moderne de planification « intelligente », la théorie de décision joue un rôle très important, par exemple, dans la modélisation de l'incertitude ou le traitement des scénarios contradictoires [11].

Dans ce chapitre on essaiera de traiter les algorithmes de planification d'un point de vue intelligence artificielle en commençant par les notions de base de planification, puis en étudiant les classes majeures de planification et enfin en couvrant divers algorithmes de la planification et d'évitement d'obstacles.

## II.2 Concept de planification :

### II.2.1 Les ingrédients de planification : [11]

**a- Etat** : un problème de planification doit capturer toutes les situations possibles. En générale un état représente la position et l'orientation d'un robot à un moment précis.

**b- Espace d'état** : l'ensemble de tous les états est appelé espace d'état.

**c -Temps** : tous les problèmes de planification définissent une séquence de décisions qui doit être appliqué avec le temps. Le temps doit être modélisé explicitement.

**d-Action** : un plan génère des actions qui manipulent les états.

**e-L'état initial et l'état final** : un problème de planification implique un démarrage d'un état initial vers un état final spécifié, appelé objectif.

**f- Critères** : Généralement deux types de critères peuvent être ajoutés pour la planification.

- 1- Un critère de faisabilité : trouver un plan qui cause une arrivée à un état objectif sans prendre en considération son efficacité.
- 2- Critère d'optimalité : trouver un plan faisable qui optimise les performances d'une manière spécifique (temps d'exécution, longueur de chemin) en plus le plan mené à l'état objectif

**g-Plan** : en général, un plan impose une stratégie, un comportement ou un décideur.

**Définition de plan** : c'est une séquence d'actions qui s'exécute à travers le temps et qui conduit d'un état initial à un état final en respectant un ou plusieurs critères.

### II.2.2 Définitions :

**a- Planificateur** : Le planificateur est le constructeur d'un plan ; il peut être un humain ou une machine. Si le planificateur est une machine, l'algorithme de planification est connu généralement comme le planificateur [11].

**b- Plan** : Une fois un plan est déterminé, on peut l'utiliser selon trois manières différentes.

- 1- Exécution: exécuter le plan soit dans une simulation soit dans un robot.
- 2- Raffinement : pour construire un plan meilleur.

- 3- Inclusion hiérarchique: intégrer le plan comme étant une action dans un algorithme d'un niveau plus haut [11].

## II.3 Classe et techniques de planification :

Des techniques de planifications différentes existent pour les différents problèmes et situations. Ces techniques peuvent être classées dans des classes selon un critère données de classification. La classification va nous aider à étudier les techniques de planification, les grouper et citer leurs caractéristiques communes.

### II.3.1 Espace de configuration :

Pour effectuer une planification réussite, il est nécessaire de trouver une bonne représentation du robot et de son environnement. Cette représentation doit être fidèle et doit refléter l'environnement du robot sans aucune perte d'information.

Une description de la géométrie du robot ( $A$ ) et de l'espace de travail ( $W$ ) est fournie..

L'espace de travail  $W = \mathfrak{R}^N$  avec  $N = 2$  ou  $N = 3$  est un environnement statique avec des obstacles. L'objectif est de trouver un chemin libre d'obstacle pour le robot  $A$ , c'est-à-dire que le robot doit être capable de se déplacer et de changer la direction afin d'éviter tous les obstacles existants dans le chemin de navigation en partant de son état initial (position et orientation initiale) et arrivant à l'état final (position et orientation finale).

Pour atteindre cet objectif, une spécification complète de la location de chaque point de la géométrie du robot doit être fournie, cette spécification est appelée configuration, et notée  $q$  [19].

L'espace de configuration ou C-espace ( $q \in C$ ) est un espace qui contient toutes les configurations possibles. Il représente l'ensemble de toutes les transformations qui peuvent être appliquées à un robot connaissant quelques informations sur sa cinématique (vitesse maximum possible, le robot est à roues ou est à pieds...).

L'avantage principal de ce concept est qu'on puisse représenter même les formes géométriques les plus complexes du robot par un seul point dans l'espace de configuration.

L'ensemble fermé  $O \subset W$  représente la région d'obstacles dans l'espace de travail défini comme étant une collection de formes géométriques diverses (l'union de plusieurs formes géométriques des différents obstacles donne l'ensemble  $O$ ).

Soit l'ensemble fermé  $A(q) \subset W$  représente l'ensemble des points occupé par le robot dans la configuration  $q \in C$ . Cet ensemble est modélisé on utilisant les mêmes primitives utilisée pour  $O$ . La région d'obstacle  $C_{obs}$  est définie par :

$$C_{obs} = \{q \in C \mid A(q) \cap O \neq \emptyset\}$$

La région d'obstacle est un ensemble fermé dans  $C$  parce que  $O$  et  $A(q)$  sont fermé l'ensemble de configuration qui évite les obstacles est  $C_{free} = C/C_{obs}$  est appelé espace libre.

### ***II.3.2 Planification discrète :***

La planification discrète est la forme la plus simple des planifications dont l'espace d'états dans la majorité des cas est fini. S'il n'est pas fini, des entiers interviennent ; dans ce cas de planification, le modèle de présentation de l'environnement n'est pas géométrique, et il n'utilise pas d'équations différentielles et on note aussi l'absence de l'incertitude ; ainsi, il n'est pas nécessaire de faire intervenir les techniques des probabilités [11].

### ***II.3.3 Planification basée sur l'échantillonnage (sampling based planning)***

#### **II.3.3.1 Présentation :**

$C$  est l'environnement du robot,  $C_{free}$  est l'espace libre de  $C$ . Le  $C_{obs}$  est l'espace occupé par les obstacles. L'idée de base est d'éviter  $C_{obs}$

L'idée majeure est d'éviter la construction explicite de  $C_{obs}$ . Ceci nous conduit à une recherche qui explore le  $C_{free}$  avec un schème d'échantillonnage. Cette exploration est effectuée par un modèle de détection de collision considéré comme étant une boîte noire par l'algorithme de planification. Ce qui permet le développement des algorithmes de planification indépendamment des modèles géométriques particuliers. Cette vision est très réussie pour la résolution des problèmes dans plusieurs domaines. Ces problèmes peuvent être pratiquement impossibles d'être résolus en utilisant les techniques qui représentent la zone d'obstacles explicitement. Dans ces algorithmes il y a des cas où on fait discrétiser (numériser) l'environnement et adapter une méthode de recherche discrète c'est dans le cas où on a un nombre réduit des points ou une petite dimension.

La détection des collisions dans ces algorithmes est considérée comme une boîte noire. Plusieurs algorithmes sont utilisés, commençant par des algorithmes de complexité assez réduite et arrivant à des méthodes heuristiques.

Vérification du segment du chemin : l'interface entre le planificateur et le mécanisme de détection de collision est usuellement associé à la validation d'un segment de chemin, une approche qui sert à segmenter le chemin est appelée le vérificateur de collision seulement sur les segments (segment par segment).

Ces algorithmes présentent une limite importante, cet ensemble d'activités logique nous met en doute, c'est-à-dire qu'il n'y a aucune garantie qu'on va trouver une solution lorsqu'on les utilise.

### ***II.3.4 Planification combinatoire :***

Tous les algorithmes de planification de ce type sont classifiés comme algorithmes complets parce qu'ils donnent une réponse définitive à la problématique. La réponse est soit la solution si elle existe, soit une réponse négative dans le cas contraire. Par contre, les algorithmes basés sur l'échantillonnage ne sont pas complets dans le sens où il se peut qu'il y ait des cas où ils ne retournent pas de réponse.

Les algorithmes combinatoires donnent une importance élevée à la représentation des données de l'environnement. Pour cette raison ils essaient de donner des réponses à quelques questions telles que :

- quelle est la représentation du robot et de l'obstacle ?
- dans quelle dimension le monde est représenté ?

Les réponses à ces questions, ainsi qu'à d'autres, donnent un ensemble d'informations suffisantes pour traiter le problème avec succès.

Ces approches sont basées principalement sur la théorie des graphes. Deux approches majeures sont la décomposition en cellules et la carte routière. Leur stratégie consiste à construire un graphe pour éviter les obstacles et pour déterminer les chemins de navigations possibles représentés par les connexions entre les différents nœuds.

## II.4. Exemples d'algorithme de planification :

### II.4.1 Carte routière (road map) :

Cette approche est basée sur l'identification d'un ensemble de routes dans l'espace libre. La base de cette approche est de capturer la connectivité de l'espace libre d'un robot dans un réseau de dimension 1 de courbes ou de lignes appelée carte routière.

Une fois la carte routière construite, elle est utilisée comme étant un réseau de segments de routes (chemin) pour la planification des mouvements des robots. L'objectif de cette approche est de construire un réseau routier qui rend le robot capable à visiter tous les points possible de l'environnement libre de navigation.

On note qu'il y a deux approches différentes pour la construction de la carte routière. Dans le premier cas les routes sont le plus proche possible de l'obstacle, d'où l'appellation de « Graphe de visibilité » les chemins qui résultent sont de longueur minimum ; le deuxième cas est appelé « diagramme de Voronoi » les routes restent le plus loin possible de l'obstacle [10].

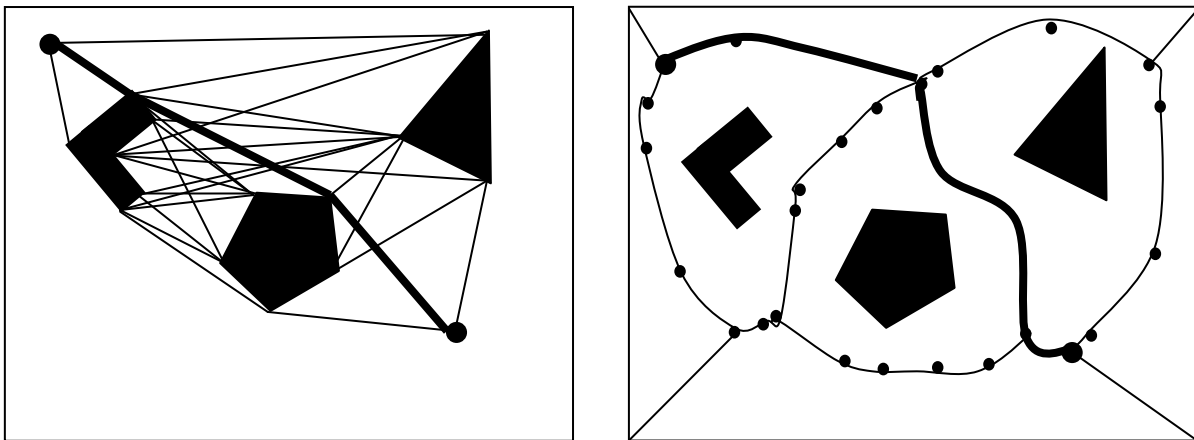
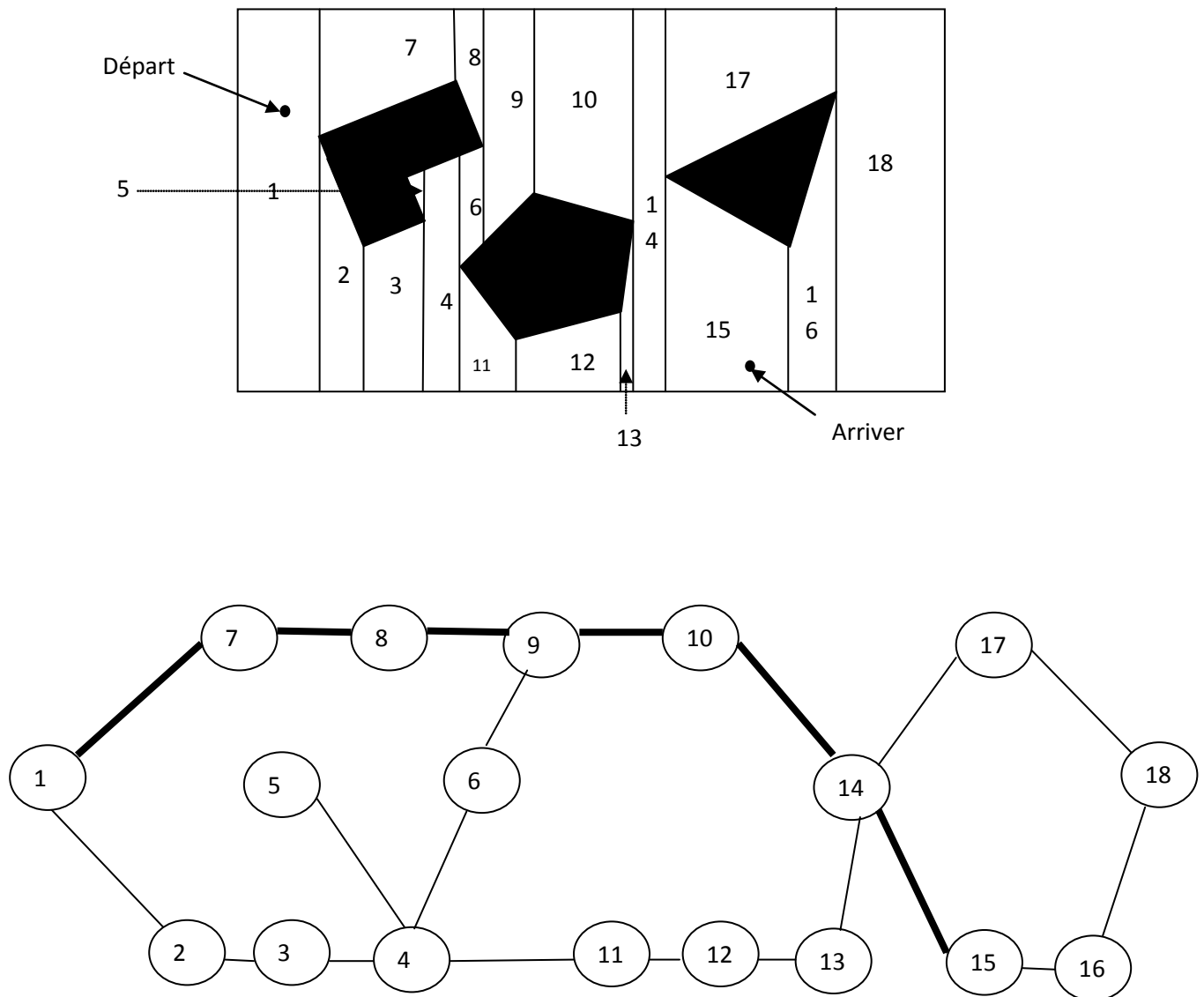


Figure II.1: Graphe de visibilité (à gauche) et diagramme de Voronoi (à droite) [19]

## ***II.4.2 Décompositions en cellules :***

L'idée de base est de faire distinguer entre les régions géométriques différentes, et décomposer en région libre et région occupé par les objets (obstacles)

- Divisé l'environnement en région simple et connectée appelée « cellule »
- Déterminer les cellules ouvertes qui sont adjacentes et construire un graphe de connectivité
- Trouver les cellules qui relient le point de départ et le point d'arrivé
- Trouver un chemin dans le graphe de connectivité qui relie le point initial aux point d'arrivé
- Partons de la séquence de cellules trouvées, et par un algorithme de recherche approprié, calculer un chemin dans chaque cellule, par exemple qui passe par le centre ou par les limites de cellules



**Figure II.2:** Exemple de la méthode de décomposition en cellules (décomposition en cellules en haut, et graphe correspondant en bas)

### II.4.3 Champ potentiel (*Potential Field*)

Cette approche considère l'environnement du robot comme étant un champ artificiel qui applique des forces différentes sur le robot. Elle traite le robot comme étant un point sous influence d'un champ potentiel artificiel. Le point d'arrivée est une force d'attraction pour le robot, les différents obstacles sont des forces répulsives. Le chemin de navigation est à calculer en faisant sommer les différentes forces [10].



## II.4.3.1 Amélioration de la méthode :

Deux améliorations sont ajoutées :

- La rotation du champ potentiel : définit la force répulsive comme étant une fonction de distance et de l'orientation de l'obstacle, [10], cela en définissant un facteur qui réduit la force répulsive si le robot et l'obstacle sont en parallèle ou il y'a une distance importante entre eux.
- Tache du champ potentiel : prendre en considération la vitesse du robot et effectuer un filtrage des obstacles qui n'influent pas le robot en court terme

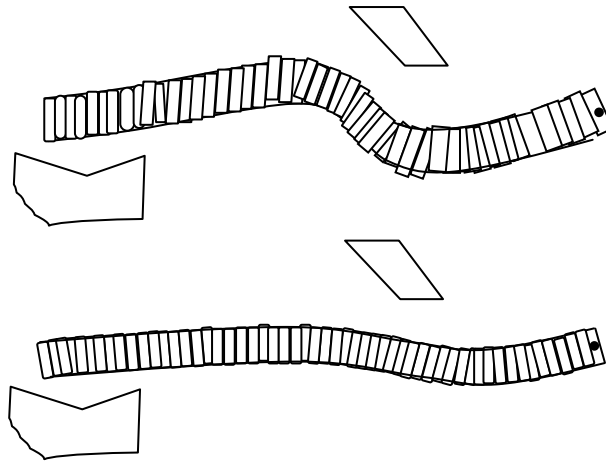


Figure II.3: Champ potentiel amélioré [10]

## II.5 Les algorithmes d'évitement d'obstacle :

Les algorithmes de planification précités traitent aussi la tâche d'évitement d'obstacle d'une manière général ;, il existe aussi des algorithmes spécifiques à l'évitement d'obstacle. Les algorithmes de planification sont plus généraux, ils ont plusieurs objectifs : établir un plan de navigation, éviter les obstacles et même optimiser. Par contre, les algorithmes d'évitement d'obstacle ont un seul objectif, c'est éviter les obstacles. Ainsi, un algorithme d'évitement d'obstacle peut être intégré dans un cas de planification ou bien un "plan" d'évitement d'obstacle peut être intégré dans un "plan" de navigation.

Il existe plusieurs approches pour l'évitement d'obstacle. On va citer les plus importantes :

### ***II.5.1 Algorithme du BUG :***

L'objectif est de générer un chemin qui relie un point de départ à un point d'arrivée. Lorsqu'un obstacle  $i$  est trouvé le robot définit un point de contact appelée  $H_i$  (Hit point). Le nombre de points de contact est égale au nombre d'obstacles. Lorsque le robot quitte la zone de l'obstacle il définit un point de départ (leave point) appelé  $L_i$ , initialement  $i = 0$  ( $L_0 = \text{départ}$ ). La procédure utilise trois registres R1, R2, R3 pour enregistrer les informations intermédiaire, tout les registres sont met à zéro si un point de contact  $H$  est définie

R1 est utilisé pour enregistrer les coordonnées du point courant  $Q_m$ , de la distance minimale entre les limites de l'obstacle et le point cible, c'est-à-dire une comparaison est effectuée à chaque point du chemin, R2 intègre la borne de l'obstacle en commençons par  $H_i$ , et R3 pour enregistrer la longueur du borne de l'obstacle calculer à partir de  $Q_m$

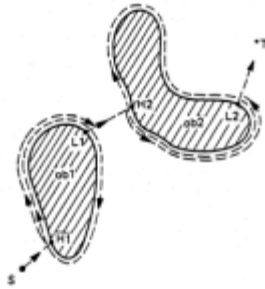
La procédure définit les étapes suivantes :

- Etape 1: du point  $L_{i-1}$  aller vers la cible au long d'une droite jusqu'à un de ces évènements suivant intervient :
  - Le robot a atteint son objectif
  - Un obstacle est trouvé et un point de contact défini; aller à l'étape 2
- Etape 2: utilisant une direction locale (gauche ou droite) naviguer aux bornes de l'obstacle (faire un tour sur l'obstacle commençons par la gauche ou la droite)
  - Si le robot a atteint son objectif alors arrêter.
  - Sinon faire un tour complet sur l'obstacle, revenir au point  $H_i$  et définir un point  $L_i = Q_m$  pour quitter la zone de l'obstacle.
  - Aller à l'étape 3.
- Etape 3 : appliquer un test sur la possibilité d'atteindre le but, s'il n'est pas possible arrêter, sinon :

# Méthodes classiques de planification et évitement d'obstacles

---

- Utiliser le contenu des registres R2, R3 pour déterminer le chemin le plus court entre tout point des bornes de l'obstacle et l'objectif
- Définir  $L_i$  et mise à jour  $i (i=i+1)$
- Aller a l'étape 1.



**Figure II.4:** Algorithme BUG 1 ( $h_1$   $h_2$  point de contact  $L_1$   $L_2$  point de départ  $s$ ,  $t$  point de départ est d'arrivé) [10]

## Avantages

Cette méthode présente quelques avantages :

- Ne fait pas explorer le même obstacle deux fois c'est-à-dire que le robot va éviter un nombre finis d'obstacles
- L'algorithme est indépendant de la géométrie de l'environnement
- Simplicité.

Inconvénient :

D'une autre part on trouve des limitations importantes qui peuvent rendre cet algorithme inutilisable

- Chemin très long: des tours sont effectués sur les différents obstacles trouvés
- A chaque point des bornes de l'obstacle des calculs de distance et des comparaisons sont effectués, cela va augmenter la complexité de l'algorithme

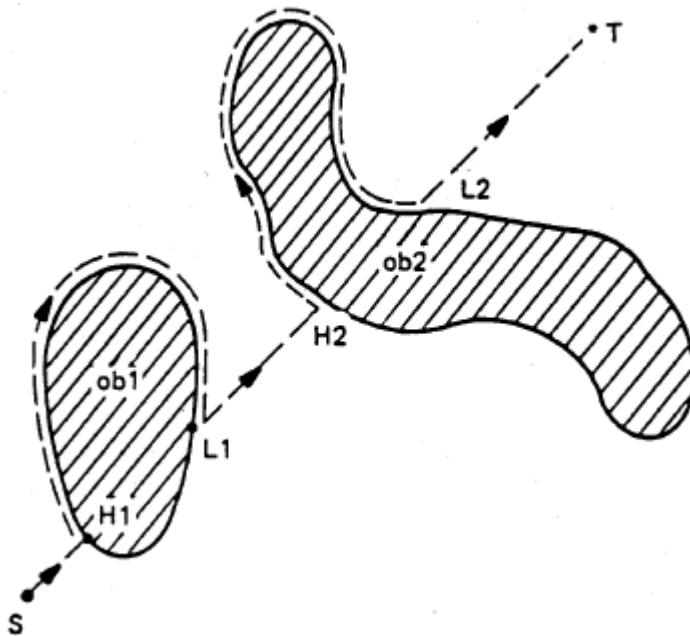
## II.5.2 Algorithme du BUG 2

On peut considérer cet algorithme comme une amélioration de l'algorithme précédent.

L'algorithme peut explorer le même obstacle  $i$  plus qu'une seule fois parce que l'algorithme n'a pas de méthode pour distinguer les différents obstacles. Pour cette raison on va utiliser l'indice  $i$  pour se référer à un d'obstacle, de plus on va utiliser  $j$  pour indiquer la  $j^{\text{ième}}$  occurrence d'un point de contact ou de départ sur le même ou sur un obstacle différent. Initialement  $j = 1$ ,  $L^0 = \text{départ}$

L'algorithme exécute les étapes suivantes :

- Etape 1 : du point  $L^{j-1}$  aller tout au long du segment de la droite (départ, arrivé) jusqu'à un des évènements suivants :
  - A- le robot a atteint l'objectif.
  - B- un obstacle est trouvée et un point de contact  $H^j$  est définie aller à l'étape 2
- Etape 2 : utilisant une direction locale acceptable (gauche, droite) suivi la borne de l'obstacle jusqu'à un des évènements suivant se présente :
  - A- le robot a atteint son objectif, la procédure s'arrête
  - B- le robot mobile et le segment droite (départ, arrivé) se mettent à un point  $Q$ , la distance  $d(Q) < d(H^j)$  et la ligne ( $Q$ , arrivé) ne se mettent pas, tandis que l'obstacle courant aux point  $Q$ , définir un point de départ  $L^j = Q$  mise a jour  $j = j+1$ , retour à l'étape 1
  - C- le robot revient aux point de contact  $H^j$  sans avoir définir le prochaine point de contact  $H^{j+1}$  le robot ne peut pas atteindre son objectif, la procédure s'arrête.



**Figure II.5:** Algorithme BUG 2

Avantages :

Cette méthode à quelques avantages par rapport au BUG 1

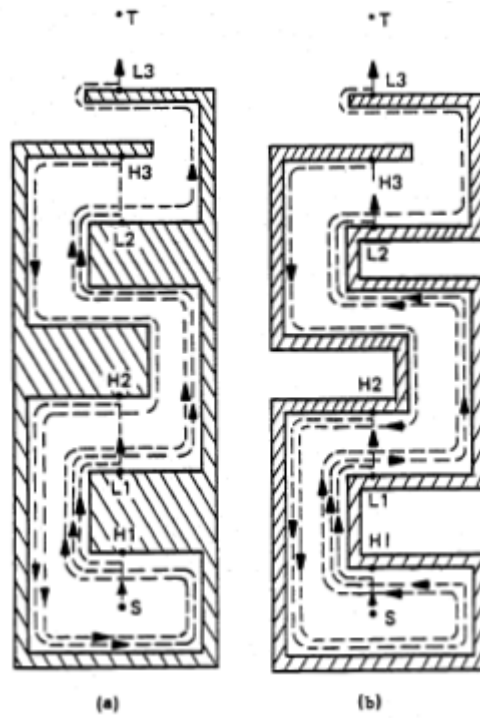
- Le chemin est court par rapport au BUG 1
- Les calculs de comparaison et de distance sont éliminés ce qui réduit la complexité de l'algorithme.

Limites :

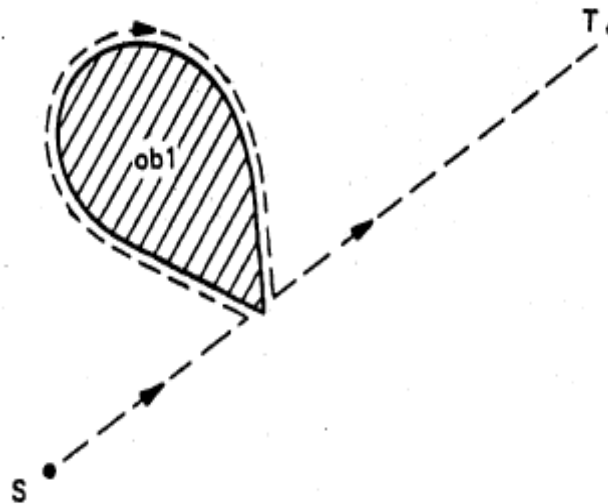
Malgré les améliorations portées sur l'algorithme il reste infaisable, on cite les limites suivantes :

- Il peut explorer le même obstacle plusieurs fois (figure. II.6)
- La stratégie de choix de direction n'est pas claire et peut causer des scénarios où le robot explore tout l'obstacle (figure. II.7)

Les deux algorithmes présentés ici sont assez simple, mais ils présentent un inconvénient majeur, ils ne trouvent jamais un chemin optimal.



**Figure II.6:** Exploration de la même région plusieurs fois par un robot qui utilise BUG 2



**Figure II.7 :** Un cas où un robot fait un tour complet sur l'obstacle sous BUG 2

## II.5.3 Histogramme du champ de vecteur :

Cette méthode construit une carte locale de l'environnement du robot. Elle génère un histogramme de deux dimensions avec l'axe x représentant l'angle  $\alpha$  et l'axe y représentant la probabilité  $p$  d'existence d'un obstacle [10].

A partir de cet histogramme une direction est calculée, premièrement, tous les espaces ouverts par où le véhicule peut passer sont identifiés. Après l'application de la fonction coût pour chaque espace, le passage minimal est choisi. Cette fonction, appelée  $G$ , est ainsi définie :

$$G = a.\text{Direction\_de\_but} + b.\text{orientations\_des\_roues} + c.\text{direction\_précédente}.$$

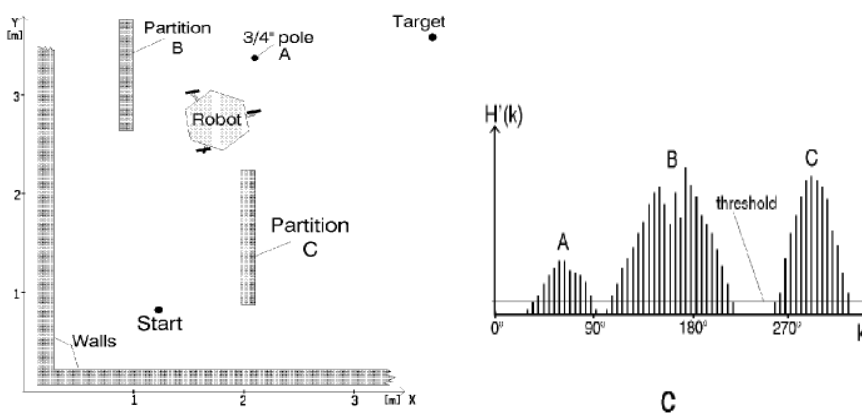
Avec :

direction\_de\_but = alignement du chemin du robot avec le but

orientation\_des\_roues = différence entre la nouvelle direction et l'orientation courante des roues

direction\_pécédente = différence entre la direction précédente sélectionnée et la nouvelle direction

Lorsque la fonction coût est calculée, on trouve que la déviation la plus grande conduit à la fonction coût la plus grande. Les paramètres  $a$ ,  $b$ ,  $c$  dans la fonction  $G$  expriment le comportement du robot.



**Figure II.8 :** (à gauche) environnement d'expérience

(à droite) l'histogramme correspondant

## II.6 Conclusion :

La planification concerne la recherche d'une solution faisable à un problème, suivant un ensemble d'étapes et en exécutant une séquence d'actions. Avec l'arrivée à l'objectif le plan suivi est dit réussi. Une telle planification est liée fortement à l'analyse correcte de la situation. Les actions effectuées peuvent être vues comme une séquence de décisions d'où un degré d'intelligence est montré.

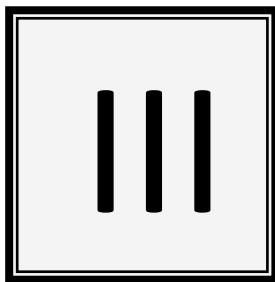
Les algorithmes présentés ici sont basés sur le traitement des cas possibles ; l'instruction « si, alors » est largement utilisée pour traiter les différents cas possibles et pour définir l'action qui convient aux cas. Le problème de ces approches de planification est que les actions sont prédéfinies selon des scénarios génériques; avec l'arrivée d'une situation (scénario) bizarre, rare ou non étudiée, alors l'efficacité de ces algorithmes sera en question, et par conséquent, un risque de ne pas avoir une solution faisable à la problématique est posé.

Pour surpasser les limites de ces approches, il est possibles de les utiliser (ou utiliser une méthode similaire) en tant que solution partielle de la problématique posée. En d'autres termes, on doit changer la façon d'utiliser ces méthodes en les injectant dans des algorithmes plus globaux, d'où ils deviennent capables de traiter des situations plus complexes.

La nature complexe des nouveaux problèmes de planification en robotique nous mène à chercher des solutions plus globales et plus efficace : ce sont les méthodes heuristiques.



## Chapitre



**Les méthodes heuristiques de la planification  
du trajectoire et évitement d'obstacle**

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

Dans le chapitre précédent on a présenté les méthodes classiques pour le problème de la planification de la trajectoire et évitement d'obstacle. Le problème majeur avec ces approches est qu'elles n'implémentent aucun comportement intelligent de haut niveau, réduisant ainsi leur capacité de traiter des situations complexes. La nécessité de trouver des solutions globales avec des aptitudes de traitement de problèmes plus difficiles a diminué les recherches dans les méthodes classiques et a introduit les approches heuristiques.

Contrairement aux anciennes méthodes, les nouvelles approches sont basées sur le caractère d'intelligence. Elles sont capables de modéliser un savoir-faire de haut niveau de l'aspect cognitif du robot ; généralement ces approches sont issues des recherches développées dans le domaine d'intelligence artificielle.

Dans ce chapitre on va définir le concept d'heuristique en citant les caractéristiques communes. On va aussi présenter certaines de ces méthodes et expliquer comment ces approches interviennent dans la résolution de notre problème.

### **III.2 Définitions et caractéristiques :**

#### ***III.2.1 Définitions :***

Le mot heuristique vient du mot grec « heuriskein » qui signifie découverte et se réfère aux théories qui décrivent comment les humains font les découvertes. L'intelligence artificielle a popularisé l'heuristique comme un moyen qui capture d'une façon quantitative les connaissances que les gens utilisent pour résoudre des problèmes quotidiens [20].

L'origine des méthodes heuristiques revient aux premières recherches en intelligence artificielle. Les chercheurs, réunis dans le symposium à Dartmouth en 1956, ont analysé l'aspect cognitif qu'un être humain utilise pour démontrer un théorème mathématique. Ils ont identifié quelques étapes communes effectuées lors de la démonstration d'un théorème. Ces étapes sont:

- Le problème est décomposé en plusieurs sous problèmes plus simples.
- Chacun de ces sous problèmes est résolu en utilisant des méthodes de substitution et de remplacement conformément aux axiomes enregistrés.

Grace à ces remarques le concept de la recherche heuristique a été introduit. Ce concept était implémenté en système de résolution de problèmes (General Problem Solver) [21].

L'objectif d'une méthode heuristique est de trouver une solution à une problématique exécutée d'une manière similaire à celle de l'être humain. Afin d'examiner ces techniques, on va citer les définitions de l'heuristique, l'intelligence et le raisonnement.

### **III.2.1.1 Définition heuristique:**

Une heuristique est une règle simple ou efficace pour la résolution d'un problème donné ou pour prendre une décision [22].

### **III.2.1.2 Définition intelligence 1:**

L'intelligence est la capacité de prendre la bonne décision basée sur un ensemble d'entrée et une variété d'actions possibles [22].

### **III.2.1.3 Définition intelligence 2:**

L'intelligence est un ensemble de propriétés du cerveau, ayant la capacité de planification, résolution de problèmes, et généralement de raisonnement [22].

### **III.2.1.4 Définition raisonnement:**

Le raisonnement est une fonction générale de l'intelligence qui permet de trouver une séquence d'actions logiques conduisant à une solution d'un problème.

Une combinaison de toutes ces informations nous conduit à déduire qu'une méthode heuristique peut être définie comme une méthode basée sur l'expérience qui guide le raisonnement afin d'arriver à une solution d'un problème donné.

## ***III.2.2 Caractéristiques des méthodes heuristiques :***

Partant des définitions présentées précédemment, on peut citer les caractéristiques suivantes :

- Des recettes génériques : les méthodes heuristiques ne sont pas spécifiques pour résoudre un problème particulier, mais elles éclairent un travail de recherche de la solution ; on simule le raisonnement humain ou on s'inspire de notre environnement. Grâce à cette propriété une méthode heuristique peut être utilisée pour résoudre des problèmes dans différents domaines
- Modéliser fortement l'intelligence : grâce aux différents algorithmes heuristiques, un savoir-faire plus évolué est défini et modélisé. Parce qu'ils présentent une approche pour raisonner, généralement ils définissent une manière pour représenter les

connaissances du domaine et appliquent une ou plusieurs règles pour arriver aux résultats souhaités. Les règles appliquées peuvent se métamorphoser d'une méthode à une autre selon la situation.

- Inspiration de l'environnement de l'être humain : La majorité des approches heuristiques sont inspirées de la biologie, l'intelligence se réside dans l'environnement naturel de l'être humain, il suffit d'effectuer quelques remarques sur un processus et développer un concept similaire pour résoudre des problèmes qui nécessitent un raisonnement.
- Qualitative ou quantitative : Une méthode est soit qualitative ou quantitative; elle est nommée qualitative lorsque elle conduit le raisonnement par un ensemble de règles logiques ; elle est désignée quantitative dans le cas où elle utilise des calculs mathématiques numériques.

### **III.3 Les méthodes heuristiques :**

Il existe plusieurs méthodes heuristiques (plus que huit méthodes). Dans la partie qui suit, on va présenter seulement trois approches, les plus connues jusqu'à aujourd'hui.

#### ***III.3.1 Système à base de connaissance :***

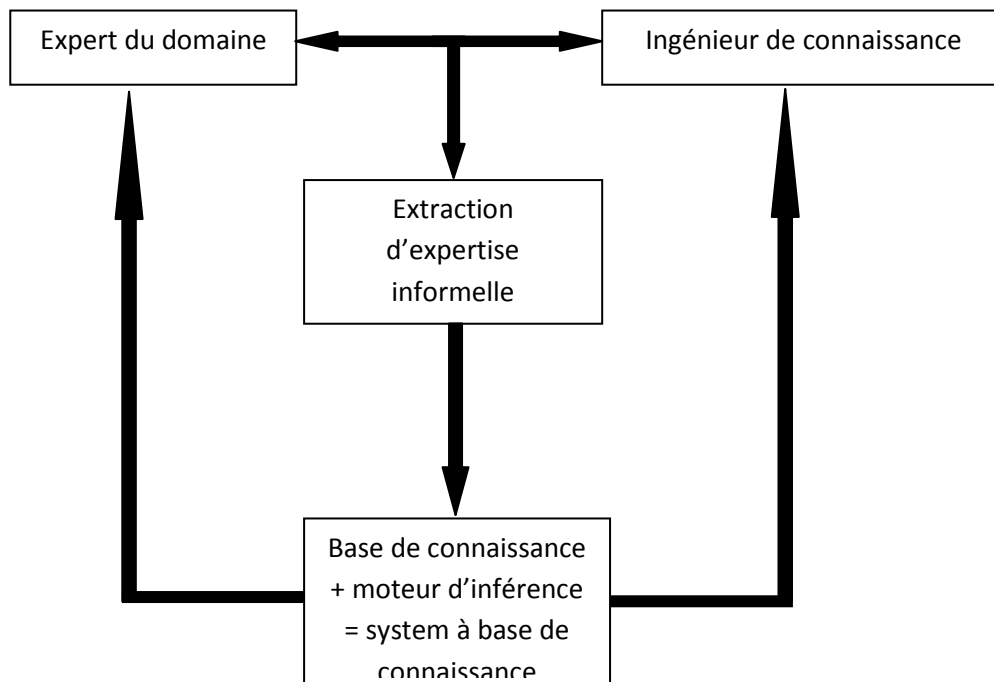
##### **III.3.1.1 Présentation :**

Les systèmes à base de connaissance sont des systèmes qui utilisent les techniques de l'intelligence artificielle pour collecter, stocker et utiliser des connaissances pour résoudre des problèmes spécifiques.

On peut citer comme exemple de systèmes :

- Les systèmes de raisonnement à base des cas,
- Les systèmes à base de règle.

Le schéma suivant explique le processus de construction d'un système à base de connaissance.



**Figure III.1:** Diagramme expliquant le processus de construction d'un système à base de connaissances

[23]

Dans le schéma ci-dessus on trouve deux personnes qui participent directement dans le processus de construction des systèmes à base de connaissance, l'expert de domaine et l'ingénieur de connaissance.

L'expert du domaine a pour objectif de donner une expertise pour modéliser afin de créer la base de connaissance.

L'ingénieur de connaissance est responsable du traitement des problèmes liés à la connaissance telle que la représentation, la structure, l'indexation des connaissances pour qu'elles deviennent aisément lisible par une machine.

Le centre d'un système à base de connaissance est la base de connaissance ; celle-ci est constituée d'un ensemble de connaissances, de relations, de règles, ainsi que l'intervention interne des données. Finalement une base de connaissance est généralement exprimée par des textes [23].

L'autre part d'un système de base de connaissance est le moteur d'inférence. En plus d'une méthodologie de raisonnement, c'est le mécanisme avec lequel la base de connaissance est interrogée. Le raisonnement à partir des cas est un exemple typique de techniques utilisées pour l'inférence [23].

### **III.3.1.2 Avantage :**

L'évitement des anciennes fautes : un système qui stock les succès et les échecs aide à fuir de nouveaux échecs.

Reproduit le raisonnement humain : généralement le mécanisme de raisonnement reflète le raisonnement humain. Il ne sera pas difficile d'expliquer un tel mécanisme aux programmeurs, les utilisateurs et mêmes les gérants [24].

### **III.3.1.3 Inconvénient du système à base de connaissance :**

Acquisition des connaissances : La tâche d'acquisition des données d'un domaine spécifique et de les convertir en représentation formelle consomme énormément de temps ; dans certains cas où on a une compréhension ambiguë du domaine, la formalisation des connaissances ne peut pas être complète [24].

### **III.3.1.4 Planification et évitement d'obstacle par les systèmes à base de connaissances :**

Les systèmes à base de connaissances ne sont pas assez utilisés ; ceci est dû à la difficulté de la modélisation des connaissances. Dans notre travail on va décrire deux travaux qui utilisent les systèmes à base de cas.

Le premier travail [25] sert à implémenter plusieurs approches traditionnelles ; chaque fois que le robot fait une navigation, l'approche utilisée va être enregistrée avec ses résultats, alors une base de cas est fondée. Avec l'arrivée d'un nouveau problème, la base de cas est consultée pour retirer le cas le plus similaire. L'approche utilisée et le chemin de navigation sont adaptés selon les besoins pour enrichir la solution de la nouvelle problématique. L'objectif de cette étude était d'assurer le résultat d'une solution. Pour cette raison, plusieurs approches ont été implémentées.

Dans le deuxième travail [26], l'approche utilise entièrement un système à base de cas. Elle a été conçue pour les véhicules autonomes sous-marins. Des informations très détaillées concernant l'environnement, les routes, les obstacles, et les données concernant une mission (navigation) sont enregistrées. Avec l'arrivée d'une commande pour une nouvelle mission, une recherche des cas similaires est effectuée ; le cas le plus similaire est retiré et adapté selon le besoin. A la fin de la navigation, la nouvelle mission est enregistrée afin d'être utilisée pour des futures navigations.

Dans l'exemple où il n'y a pas de cas similaire, la méthode effectue une composition de route sans utiliser la base de connaissances.

Le problème de ces approches est qu'elles adaptent une base de connaissances, ce qui augmente le degré de la complexité. Les solutions nécessitent des ressources importantes pour réussir leur implémentation ; pour les robots à ressources limitées, ces méthodes ne seront pas utiles.

### **III.3.1.5 Pour notre problème :**

La nature de notre problème nécessite un calcul continu de chemin de navigation. Les systèmes à base de connaissances ne sont pas conçus pour faire des calculs numériques mais plutôt pour un raisonnement (calcul logique), et par conséquent, la possibilité d'utiliser un système à base de connaissances est restreint.

Remarque : On peut utiliser une base de connaissances pour ces problèmes. Dans ce cas l'objectif sera de construire une mémoire pour le robot pour l'aider à résoudre le problème.

### **III.3.2 Les algorithmes génétiques :**

#### **III.3.2.1 Présentation :**

C'est un algorithme de recherche multiple basé sur le principe de la génétique naturelle et la sélection naturelle [27].

Les étapes peuvent être comme suit [28].

**Etape 1:** Choisir une population aléatoire de  $n$  chromosomes ( $n$  solutions acceptables de la problématique).

**Etape 2:** Mesurer la fitness  $f(x)$  de chaque chromosome  $x$  dans la population.

**Etape 3:** Créer une nouvelle population par la répétition des étapes suivantes jusqu'à la nouvelle population soit complète

**Etape 4:** La sélection: sélectionner deux chromosomes selon leur fitness (le chromosome avec la meilleure fitness est sélectionné au premier lieu).

**Etape 5:** Croisement : avec une probabilité de croisement, métrer les parents pour obtenir des nouveaux enfants. S'il n'y a pas de mélange, le résultat est une copie exacte des parents,

**Etape 6:** Mutation.

**Etape 7:** Acceptation: placer des nouveaux résultats dans la nouvelle population

**Etape 8:** Remplacer: utiliser la nouvelle population pour l'exécution d'algorithme

**Etape 9: Test:** si le critère d'arrêt est achevé, arrêter et retourner à la meilleure solution sinon aller à l'étape 2.

### **III.3.2.2 Avantage:**

L'avantage principal de ces algorithmes, c'est qu'ils contribuent à arriver aux résultats. On n'a pas besoin de définir les caractéristiques de la solution optimale, il suffit seulement de commencer par une solution quelconque.

### **III.3.2.3 Inconvénient:**

Les algorithmes génétiques sont coûteux en fonction du calcul.

### **III.3.2.4 les algorithmes génétiques pour la planification de la trajectoire et évitement d'obstacles:**

Il existe beaucoup de travaux qui utilisent les algorithmes génétiques. Le premier qu'on va citer [29] sert à représenter l'environnement par un graphe de grille (10\*10 ou 100\*100). Les chemins de la première génération sont créés en connectant les nœuds du graphe, y inclus les points de départ et d'arrivée. La fonction du fitness est mesurée par le calcul de distance entre les différents nœuds. On fait sommer la distance entre tous les nœuds qui forme un chemin et les chemins avec les distances les plus courtes sont sélectionnés pour le croisement et la mutation. Le processus est répété jusqu'à l'obtention des résultats optimaux. Les résultats obtenus sont présentés par les tableaux suivants :



## Les méthodes heuristiques de planification et évitement d'obstacles

---

Expérience	Taille de la population	Meilleure valeur de fitness	N génération
1	10	20	60
2	20	18	60
3	50	18	8

**Tableau III.1:** Environnement 10\*10 sans présence d'obstacles

Expérience	Taille de la population	Meilleure valeur de fitness	N génération
1	10	198	80
2	20	198	40
3	50	198	30

**Tableau III.2:** Environnement 100\*100 sans présence d'obstacles

Expérience	Taille de la population	Meilleure valeur de fitness	N génération
1	10	26	90
2	20	22	90
3	50	20	70

**Tableau III.3:** Environnement avec présence d'obstacles (complexité moyenne)

Expérience	Taille de la population	Meilleure valeur de fitness	N génération
1	1000	26	90
2	1200	26	90
3	2000	24	70

**Tableau III.4:** Environnement avec présence d'obstacles (complexité élevée)

**Discussion** : Les deux premiers tableaux des résultats optimaux sont obtenus après un nombre assez élevé de générations. Dans les deux premières expériences, avec l'augmentation de la taille de la population des meilleurs résultats sont obtenus avec un nombre de génération réduit.

Avec la présence d'obstacles (tableau 3 et 4) le nombre de génération est plus élevé. Alors pour obtenir des meilleures valeurs de fitness (résultats optimaux) il faut augmenter le nombre de la population de départ.

Deux problèmes se posent :

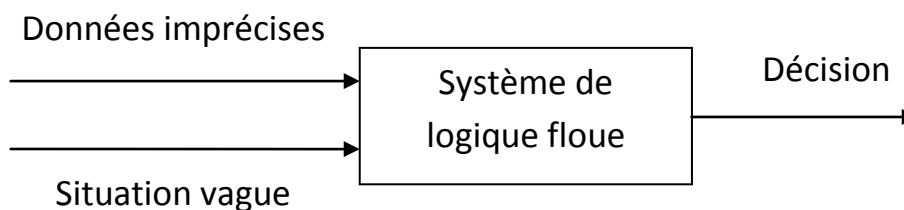
- La taille de la population et des générations (itération) est élevée, ce qui implique une croissance de la complexité.
- Avec l'augmentation de la taille de la population du démarrage on obtient les meilleurs résultats. Cependant, personne ne peut assurer qu'une taille précise de la population ne va pas améliorer les résultats de l'optimisation, c'est-à-dire que les résultats obtenus sont optimaux par rapport à la population de départ, mais on ne sait pas s'ils sont les plus optimaux.

Il existe également d'autres travaux qui utilisent les algorithmes génétiques. Généralement, ils sont similaires à celui-ci, ils représentent l'environnement de la même manière. Ces algorithmes sont généralement implémentés dans des environnements de simulation, donc ils ont besoin d'être installés sur des robots réels afin d'être testés et validés [30].

### ***III.3.3 La logique floue:***

C'est une théorie de l'intelligence artificielle basée sur les ensembles flous. La méthode fut introduite par LOTFI ZADEH [31]

L'objectif de la logique floue est de fournir une fondation pour le raisonnement approximatif, qui utilise des propositions imprécises basées sur la théorie des ensembles flous [31].



**Figure III.2:** Système de logique floue [32]

### **III.3.3.1 Ensemble Flou :**

Les ensembles flous sont définis par une fonction appelée fonction d'appartenance. Cette fonction associe à chaque élément de l'ensemble flou une valeur qu'elle n'appartient pas forcément à un rang. La valeur associée indique le degré d'appartenance à un ensemble et la valeur la plus élevée indique un degré d'appartenance fort [33] la définition donnée par Lotfi Zadeh est la suivante:

Soit  $X$  un espace de points (d'objets) avec  $x$  un élément générique  $X = \{x\}$

Un ensemble flou (classe)  $A$  dans  $X$  est caractérisé par une fonction d'appartenance (caractéristique)  $F_a(x)$  qui associe à chaque point de  $X$  une valeur réelle dans l'intervalle  $[0,1]$ , la valeur  $F_a(x)$  dans  $x$  représente le degré d'appartenance de  $x$  dans  $A$  [34].

### **III.3.3.2 Composant d'un system de logique floue :**

Un système de logique floue se compose principalement de deux composants [32] :

- Base de règles floue
- Moteur d'inférence floue

A- la base de règles: elle est constituée d'un ensemble de règles sous différentes formes [32] :

- Sous forme d'affectation : dans ce type de règles, une valeur est affectée à une variable (exemple, température = élevée).
- Sous forme conditionnelle : une condition spécifique est mentionnée, si la condition est juste, une action est exécutée (exemple, si la moyenne est supérieure ou égale à dix, alors l'étudiant passe à l'année suivante).
- Sous forme non conditionnelle : il n'y a pas de condition à vérifier avant d'exécuter une action, elle est sous forme d'une commande (exemple : aller à l'instruction  $c$  ; arrêter).

B - Le moteur d'inférence : le rôle du moteur d'inférence est la définition d'ensemble de la sortie. Selon la base des règles, l'ensemble de sortie est une décision et le moteur d'inférence est le

mécanisme de raisonnement approximatif ; généralement il utilise des méthodes d'inférence spécifique. [32]

Remarque :

- les systèmes de logique floue sont considérés parmi les systèmes à base de connaissances.
- il existe plusieurs méthodes d'inférence, on peut citer à titre d'exemple les méthodes Mamdani et Takagi–Sugeno–Kang. [32]

### **III.3.3.3 La logique floue pour la planification de trajectoire et évitement d'obstacles :**

La logique floue est utilisée en robotique lorsqu'on n'a pas suffisamment d'information sur le terrain de navigation [35] ou lorsque le robot est équipé par des matériaux (capteurs, actionneurs) de moindre qualité qui peuvent causer beaucoup d'erreurs [36]. Pour gérer l'imprécision et de l'incertitude, l'utilisation de la logique floue dans de telles situations permet une navigation sécurisée. Cela est assuré par une modélisation forte d'erreurs et de l'environnement. Dans ces cas et en suivant les règles prédéfinies les décisions sont bien justifiées. Quant à la complexité du calcul, la logique floue n'est pas coûteuse.

Il existe plusieurs travaux qui utilisent la logique floue pour la navigation. On va décrire seulement deux. Le premier concerne le développement d'un système de navigation d'un rover planétaire [35]. L'approche sert à analyser le terrain ; la sortie est la nature de la navigation dans ce terrain (dangereux, risqué, sécurisé). L'analyse du terrain est basée sur l'analyse de quatre facteurs : la pente, la rugosité, la discontinuité et la dureté du terrain.

Concernant la pente, elle inclut trois cas : le premier est lorsque la surface du terrain est plate, le deuxième concerne les surfaces penchées, le troisième est le cas des surfaces qui sont de pente raide.

La rugosité est aussi jugée en trois classes : un terrain doux, un terrain dur est un terrain plein de rocher.

Le troisième concerne l'existence de forme de discontinuité du terrain telle que les vallées, les falaises ou les ravins. Cette discontinuité est présentée par une mesure de distance qui est jugée petite ou grande.

## Les méthodes heuristiques de planification et évitement d'obstacles

---

Le terrain est classé en trois classes d'un point de vue dureté : il peut être souple, solide ou pas solide.

Le tableau suivant représente la classification du terrain selon ces quatre facteurs:

Pente	rugosité	Distance de séparation	Dureté	Classification du terrain
Plate	Douce	Petite	Souple	Risqué
Plate	Douce	Petite	Solide	Sécurisé
Plate	Dure	Petite	Solide	Risqué
Plate	Dure	Petite	Pas solide	Dangereux
Penchée	Douce	Petite	Souple	Risqué
Penchée	Douce	Petite	Solide	Sécurisé
Penchée	Dure	Petite	Solide	Risqué
Penchée	Dure	Petite	Pas solide	Dangereux
		Grande		Dangereux
Pente raide				Dangereux
	Pleine de rochers			Dangereux
<ul style="list-style-type: none"> <li>• les champs vides de la base indiquent que les entrées n'ont pas d'effet sur les règles de sortie</li> </ul>				

**Tableaux III.5 :** Base de règles floues pour la classification du terrain

Plus que l'analyse de l'environnement, la stratégie de navigation implémente deux autres comportements : le premier pour éviter les obstacles et le deuxième pour atteindre l'objectif. En total, le système à logique floue pour ce robot est composé de trois comportements : l'analyse de terrain, l'évitement d'obstacles et l'objectif de la mission.

Remarque:

- Les obstacles et l'objectif sont classés comme suit : très proche, proche, et loin.
- La navigation basée sur l'objectif sert à définir la vitesse et l'orientation de navigation en calculant la distance des obstacles et la distance de l'objectif pour définir la vitesse de navigation (exemple : si obstacle est loin et objectif est loin alors vitesse = élevée).

Les expériences réelles effectuées grâce à un robot mobile tout terrain ont montré que le robot mobile sélectionne l'environnement le plus facile pour compléter sa mission. Il assure aussi la tâche d'évitement d'obstacles en cherchant à atteindre l'objectif de la navigation.

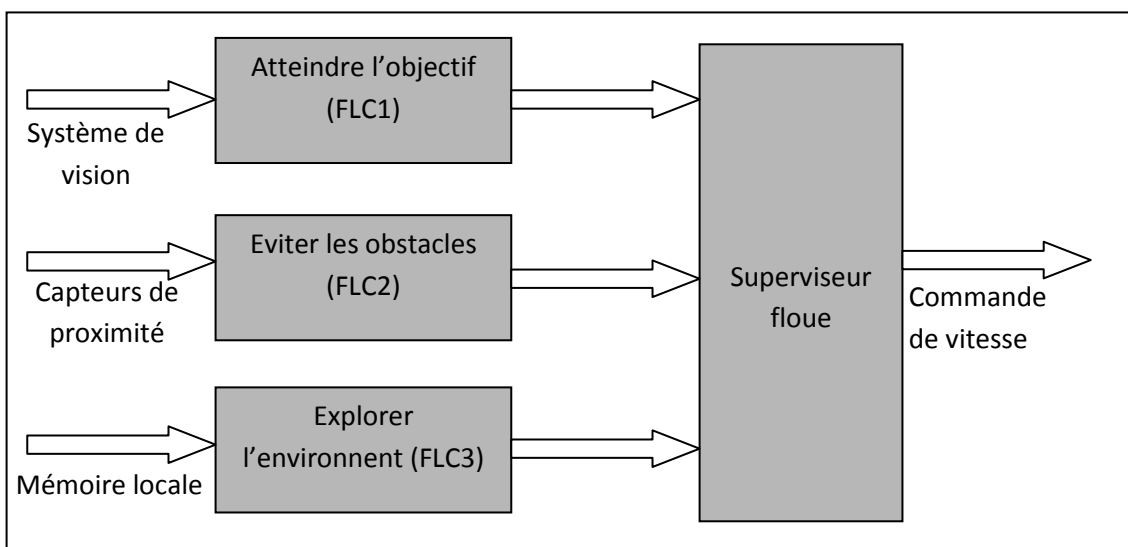
Un autre travail s'intéresse au contrôle du robot mobile par la logique floue [36] sert à présenter un concept de contrôle basé sur la définition de trois comportements différents. Ces derniers sont les comportements d'atteindre l'objectif, éviter les obstacles et explorer l'environnement. Trois contrôleurs flous sont associés à chaque comportement respectivement : ce sont FLC1, FLC2, FLC3. Ces trois contrôleurs fournissent les entrées pour un superviseur flou qui a comme sortie une commande de vitesse des roues du robot. Le rôle exact du superviseur flou est la définition de la priorité d'exécution des trois comportements élémentaires, c'est-à-dire qu'il y a trois sorties venant des trois contrôleurs. Le superviseur flou choisit une commande parmi ces trois pour l'exécuter (figure III.3).

Les expériences sont effectuées grâce à un petit robot mobile appelé khepera (ce robot est développé par l'Ecole Polytechnique Fédérale de Lausanne). Ce robot a deux roues et chaque roue est alimentée par un petit moteur électrique. Il est équipé d'une petite caméra qui assure la collection d'information visuelle, et un ensemble de huit capteurs de proximité appelé  $S_0 S_1 \dots S_7$ . Le système visuel a pour rôle l'identification de l'objectif ; les capteurs de proximité sont utilisés pour identifier les obstacles ; la fonction d'exploration est assistée par une mémoire locale.

Chaque contrôleur a une table de règles floue. Le contrôleur FLC1 a pour entrées la distance et la direction de l'objectif. La distance est exprimée par zéro, proche, loin, (parfois par quelconque pour dire que l'entrée n'est pas importante). La direction est exprimée par centre, gauche, droite, centré-gauche, centré-droite, (parfois par quelconque pour indiquer que l'entrée n'est pas importante). Les sorties sont des commandes pour identifier les vitesses des moteurs, des roues gauches et droites. La vitesse est définie par zéro, positive lent, positive rapide, négative lent, négative rapide ; le tableau III.6 exprime la base des règles floues pour le contrôleur FLC1.

Le contrôleur FLC2 est alimenté par les informations venant des capteurs de proximité. Ces informations expriment la distance entre le robot et l'obstacle, la distance appartient à un des ensembles flous suivants : loin, se rapprocher, près et en collision, la sortie est la vitesse des deux roues. (Tableau III.7)

L'activité d'exploration est contrôlée par FLC3. Le principe est de découper l'environnement en quatre régions, le nord, le sud, l'est et l'ouest. Ces régions sont ou bien moins explorées ou bien explorées ; un autre indice indique que l'entrée n'est pas importante en utilisant la variable quelconque. (Tableau III.8)



**Figure III.3:** Schème de contrôle basé sur le comportement

Entrées (ET logique)		Sorties	
Distance	Direction	Vitesse droite	Vitesse gauche
Zéro	Quelconque	Zéro	Zéro
Proche	Centre	Positive lent	Positive lent
Loin	Centre	Positive Rapide	Positive rapide
Quelconque	Gauche	Négative rapide	Positive rapide
Quelconque	Centré- gauche	Négative lent	Positive lent
Quelconque	Droite	Positive Rapide	Négative rapide
Quelconque	Centré-droite	Positive lent	Négative lent

**Tableau III.6:** La base de règles floues pour le contrôleur FLC1

Entrées (OU logique)						Sorties	
$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	Vitesse gauche	Vitesse droite
Coll.	Coll.	Coll.	Qlq	Qlq	Qlq	Rapide négative	Zéro
Qlq	Qlq	Qlq	Coll.	Coll.	Coll.	Zéro	Rapide négative
Pré	Rap	Qlq	Qlq	Qlq	Qlq	Rapide négative	Lent positive
Qlq	Qlq	Qlq	Qlq	Qlq	Rap	Lent positive	Rapide négative
Loin	Loin	Loin	Loin	Loin	Loin	Rapide positive	Rapide positive

**Tableau III.7:** La base de règles floues pour le contrôleur FLC2

Avec coll : c

ollision ; qlq : quelconque ; pré : prés de ; rap : se rapprocher



Entrées (ET LOGIQUE)				Sortie	
Delta nord	Delta sud	Delta est	Delta ouest	Vitesse gauche	Vitesse droite
Moin exp	Moin exp	Moin exp	Moin exp	Rapide positive	Rapide positive
Bien exp	Qlq	Qlq	Qlq	Rapide positive	Rapide négative
Qlq	Bien exp	Qlq	Qlq	Rapide positive	Rapide positive
Qlq	Qlq	Bien exp	Qlq	Rapide positive	Positive lent
Qlq	Qlq	Qlq	Bien exp	Lent positive	Rapide positive

**Tableau III.8:** La base de règles floues pour le contrôleur FLC3

Avec moin exp : moins explorer ; bien exp : bien explorer qlq : quelconque.

**Résultat:** les résultats ont montré que le robot exécute les comportements atteindre l'objectif et explorer l'environnement lorsque il est dans une région vide d'obstacles. Lorsque le robot se rapproche d'un obstacle, le superviseur flou fait activer le contrôleur flou associé au comportement d'évitement d'obstacles.

### III.3.3.4 Discussion des travaux:

La logique floue propose une bonne solution de la problématique ; les deux travaux donnent des bons résultats de navigation. Ils trouvent des chemins faisables pour une mission ; mais le problème qui se pose est que toutes les solutions présentées ne sont pas optimales, et cela parce que la présentation des solutions est sous forme d'une commande linguistique telle que naviguer dans la région A ou s'éloigner de l'obstacle b avec une méthode de raisonnement approximatif. Pour cette raison le chemin de navigation n'est pas optimal. Avec des ressources limitées telles que le robot utilisé dans le

deuxième travail ou même avec le premier robot si on ne cherche pas à optimiser le chemin et les ressources telles que la consommation d'énergie, on risque de ne pas avoir des bons résultats. La logique floue peut donner des meilleurs résultats pour le contrôle de tâche de haut niveau. Le calcul de chemin à chaque point de trajectoire n'est pas vraiment faisable par un système de logique floue.

Les systèmes à logique floue ont un autre problème : c'est un problème qui concerne tous les systèmes à base de connaissances, il s'agit de la sélection de la technique d'acquisition de connaissances et la disponibilité d'un expert humain [37]

### **III.4 Les autres méthodes intelligent:**

Il existe d'autres approches et méthodes pour résoudre ce problème; mentionnons les algorithmes de colonie de fourmis, et l'optimisation par essaim particulier.

Les algorithmes de colonie de fourmis sont inspirés du comportement des fourmis quand elles cherchent un chemin entre les sources de la nourriture et leurs colonies. Lorsqu'une fourmi parcourt un chemin elle laisse une trace dans ce chemin en produisant une matière appelée « phéromones ». Cette matière est attirante pour les autres fourmis, alors elles suivent les traces des phéromones. L'existence de cette matière est liée à la distance ; plus la distance est longue, plus la concentration de la matière devient faible. Si le chemin est court on remarque des traces fortes de cette matière, en conséquence, les fourmis suivent la trace la plus forte ce qui indique un chemin plus court ou optimal [38].

Les algorithmes de colonies de fourmis ont pour objectif d'optimiser un chemin pour les robots mobiles. Etant donné qu'ils nécessitent un groupe pour qu'ils soient réalisés, ces algorithmes peuvent être programmés pour une tâche qui exige un travail collaboratif de plusieurs robots mobiles. Dans notre problème on s'intéresse uniquement à un seul robot mobile.

Le principe des algorithmes d'optimisation par essaim est basé sur la simulation des comportements des animaux sociaux tels que les oiseaux migrateurs. Initialement, on choisit une population initiale aléatoirement dans un espace de recherche ; la meilleure solution est obtenue tout simplement par l'ajustement de sa trajectoire vers la meilleure particule et vers la meilleure particule dans l'essaim entier [39] ; en d'autres termes, un individu dans le groupe va mesurer sa trajectoire et la comparer aux autres trajectoires. À la fin il va choisir la meilleure.

L'optimisation par essaim est un peu similaire aux algorithmes génétiques ; elle est mieux adaptée pour les problèmes de la robotique collective. Tout comme les algorithmes de colonie de fourmis, elle s'intervient à la résolution des problèmes d'optimisation d'un chemin avec la participation de tout le groupe à la recherche de la meilleure solution.

### **III.5 La fusion de plusieurs méthodes intelligentes:**

La fusion de plusieurs méthodes intelligentes désigne l'utilisation de plusieurs approches pour résoudre un problème donné. Cela se fait par la décomposition d'un problème de complexité élevée en plusieurs sous-problèmes de complexité réduite et chaque sous problème est résolu par une méthode différente. Comme exemple de cette approche de résolution de problèmes, on peut citer [27] qui est comme suit : découper le problème en deux sous-problèmes ; trouver une manière pour représenter l'environnement ; et trouver un chemin de navigation optimale. Un réseau de neurones est utilisé pour résoudre le premier sous-problème, et un algorithme génétique intervient pour la mise en place d'un chemin de navigation optimale.

Il est également possible d'utiliser plusieurs approches. À titre d'exemple, on cite un réseau de neurones qui a pour entrées une ou plusieurs variables appartenant à un ensemble flou ou l'inverse lorsque les sorties d'un réseau de neurones seront classées dans un ou plusieurs ensembles flous qui fournissent des entrées pour un système de logique floue [21]; il est également possible de les utiliser d'une autre manière.

La fusion de plusieurs techniques intelligentes pour résoudre un problème donné est avantageuse pour des problèmes de complexité élevée plus que dans les cas des problèmes simples; dans notre cas, on ne peut pas utiliser plusieurs techniques intelligentes afin simplifier la solution de notre problème.

### **III.6 Conclusion:**

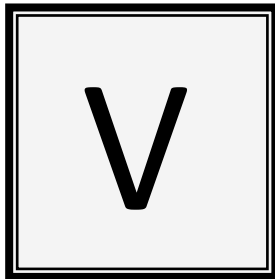
Les techniques présentées ici sont idéales non seulement pour la résolution d'un problème de navigation mais aussi pour la résolution de n'importe quel autre type de problèmes lié à la robotique. Leur avantage principal est leur capacité de trouver la solution d'une façon intelligente. Ce qui optimise et évolue le comportement de la machine mobile. Cela permet un traitement de problèmes de plus haut niveau d'une manière authentique. Il permet aussi une définition de l'aspect cognitive du robot d'une manière qui offre la perspective du développement des machines plus intelligentes qui traitent et effectuent des tâches plus proches et similaires à des tâches effectuées par un humain.

La possibilité de fusionner plusieurs techniques d'intelligence artificielle peut résoudre notamment deux problèmes : le premier est le problème pour lequel on fait fusionner plusieurs techniques, le deuxième est construire une vue plus exacte des problèmes réels en utilisant plusieurs approches de l'intelligence artificielle. Ce qui aide à construire un cadre de travail commun et mettre fin au conflit entre les différents défenseurs des approches. Ceci indique un développement en coopération de la future solution intelligente qui va mettre en place la machine intelligente désirée.

Le problème avec ces approches réside sur le fait qu'elles sont heuristiques. Personne ne peut assurer qu'une telle solution va sûrement résoudre un tel problème et ainsi l'échec de mission est toujours possible. Cela est dû au fait qu'elles sont des recettes génériques qui définissent un ensemble d'action général.

L'avantage principal des réseaux de neurones par rapport aux autres techniques intelligentes déjà présentées ici est qu'ils répondent bien à nos exigences parce qu'ils peuvent construire une solution d'une façon simple, c'est-à-dire, ils sont capable de contrôler un robot, construire une mémoire, et ils ont aussi l'aptitude de calculer un chemin exact de navigation et l'optimiser avec un minimum de temps, de ressources et de simplicité.

## Chapitre



**Les réseaux de neurones**

## IV.1 Introduction :

Le développement des réseaux de neurones est lié à la notion des neurones artificiels inspirée du fonctionnement des vrais (humains ou non). Ces neurones artificiels sont une abstraction formelle du comportement du neurone biologique [39]. Cela conduit à étudier les réseaux de neurones comme un outil de modélisation cognitive des tâches complexes. Ainsi, le domaine d'application est large, englobant la reconnaissance des formes, les antivirus, le contrôle des machines, le contrôle de trafic aérien, l'aide à la décision, les banques et les assurances comme exemples typiques d'application des réseaux de neurones.

Les réseaux de neurones forment une solution idéale pour certains problèmes qui nécessitent un raisonnement, ou qui sont de complexité élevée. Cela est dû à leur capacité d'apprentissage et à leur tendance d'être déployé dans différents environnements. La nature mathématique des réseaux de neurones permet une implémentation simple et efficace pour la mise en place de la solution.

La nature de problème à résoudre a une grande influence sur la conception d'un réseau de neurones. Cela est dû aux différences entre les entrées, les sorties et les traitements requis pour résoudre le problème données, d'où l'existence de différentes architectures neuronales dédiées à résoudre une diversité de problèmes de différents domaines.

En robotique les réseaux de neurones sont largement utilisés pour résoudre des problèmes différents dans ce domaine. Cela commence par la vision et la reconnaissance de formes jusqu'à la planification et l'aide à la décision. L'utilisation des réseaux de neurones en robotique ne se résume pas dans l'implémentation d'une solution neuronale à un problème donné, mais elle tente aussi à étudier et expliquer les capacités des réseaux de neurones en tant qu'un modèle intelligent.

Les réseaux de neurones, comme tout système réalisé par un humain, ont des avantages et des inconvénients. Leur avantage principal réside dans leur capacité d'apprentissage élevée qui permet de les déployer pour résoudre des problèmes des situations nouvelles. Sinon, leur inconvénient majeur concerne essentiellement la méthode de représentation de connaissances.

Dans cette partie de travail, on va examiner les réseaux de neurones d'une manière générale. D'abord on va citer quelques définitions ; ensuite, on va étudier leurs propriétés de bases et les différentes architectures neuronales. Nous allons expliquer aussi les concepts d'apprentissage et de représentation des connaissances. A la fin nous allons citer et examiner quelques travaux réalisés pour la résolution de la navigation.

## IV.2 Définitions et concepts de base :

En 1943 MacCulloch et Pitts avaient publié un papier très intéressant (intitulé : A logical calculus of the ideas immanent in nervous activity). Dans ce papier ils ont expliqué le principe d'un neurone et introduit les réseaux de neurones non cycliques et les réseaux de neurones cycliques.

Ils ont démontré que n'importe quelle fonction logique peut être calculée (simulée, effectuée) par un réseau de neurones avec des seuils [40].

De cette démonstration on constate qu'il est possible de résoudre une variété de problèmes de la logique. La logique qu'on connaît est un domaine de mathématique et pour mieux comprendre ce qu'elle veut

dire nous allons identifier les différents types des problèmes qu'on peut résoudre en utilisant un réseau de neurones.

Définition logique 1: une branche de la philosophie qui analyse les inférences [41].

Définition logique 2: jugement raisonné et raisonnable [41].

Définition logique 3: principes qui guident le raisonnement à un domaine ou à une situation [41].

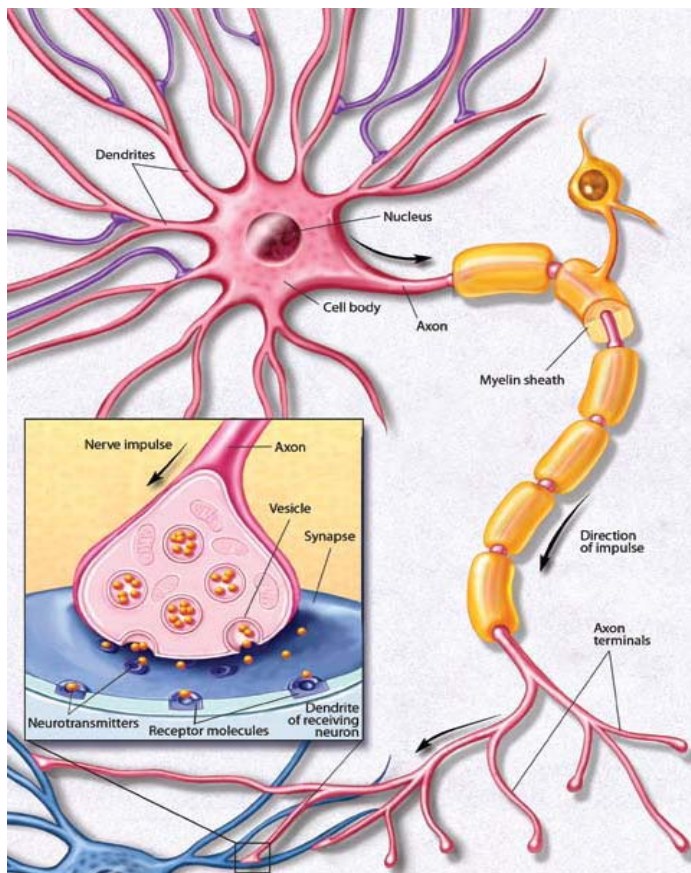
Remarquons que toutes les définitions se tournent autour de deux mots : inférence et raisonnement alors on peut déduire de ça et de l'hypothèse démontrée par MacCulloch et Pitts que n'importe quel raisonnement ou inférence peut être effectué grâce à un réseau de neurones, d'où on constate que les problèmes qui peuvent être résolus par les réseaux de neurones sont les problèmes qui requièrent des raisonnements. Autrement dit, ce sont les problèmes qui exigent de l'intelligence pour être résolus. Ainsi et selon la même hypothèse on peut modéliser une partie (ou le tout) du savoir-faire humain.

Les réseaux de neurones ont une capacité d'apprendre à travers les exemples. La question qui s'impose est la suivante : où se trouve la capacité d'apprentissage ? Une réponse à cette question était donnée par Hebb en 1949 qui a démontré que l'apprentissage dépend des changements de poids synaptique [40].

Les réseaux de neurones artificiels sont inspirés des neurones biologiques ; nous commençons d'abord par étudier celles-ci (neurones biologiques) :

### ***IV.2.1 Le neurone biologique :***

Les réseaux de neurones qu'on étudie sont inspirés des neurones biologiques (naturels). Avant de tenter à expliquer ce qu'est un neurone artificiel, on passe par une explication brève du neurone naturel.



**Figure IV.1:** Neurone biologique [42]

Définition de neurone biologique: Un neurone tire en transmettant des signaux électriques le long de son axone. Quand les signaux atteignent la fin de l'axone, ils libèrent les neurotransmetteurs qui sont conservés dans des poches appelées vésicules. Les neurotransmetteurs se lient aux molécules de récepteur sur les surfaces de neurones adjacents. Le point de contact virtuel est connu comme la synapse [42].

La fonction d'un neurone donc consiste à recevoir un signal et le transmettre à un autre neurone. Ce même principe de fonctionnement existe dans les réseaux de neurones artificiels : ils reçoivent un signal en entrée, effectuent une fonction mathématique et les calculs nécessaires, puis ils l'envoient ensuite au prochain neurone [42].

Un neurone est composé principalement de core, dendrites et axone. Le point de connexion entre un neurone et un autre est appelé synapse.

**IV.2.1.1 Dendrites :** Ils sont similaires à un arbre de fibres. Ils sont les portes d'entrées des ; leur densité varie entre  $10^3$  et  $10^4$  dendrites par neurone. Ils forment une interface réceptrice pour les signaux d'entrée des neurones [52].

**IV.2.1.2 Synapse :** Il s'agit d'une zone de stockage de l'expérience (base de connaissances). Il procure la mémoire long terme (Long Term Memory, LTM). Il reçoit les informations à partir des capteurs et des autres neurones et il fournit la sortie via l'axone [52].



**IV.2.1.3 Le corps cellulaire (soma) :** Il reçoit les informations synaptiques et exécute davantage un traitement sur ces informations. Presque toutes les fonctions logiques du neurone sont réalisées dans le corps cellulaire [52].

**IV.2.1.4 Axone :** la ligne de sortie du neurone est appelée axone. La sortie qui apparaît sous forme d'un potentiel d'action sera transmise aux autres neurones pour des traitements ultérieurs [52].

Dans les réseaux de neurones artificiels, le principe est le même : il y'a une entrée, un nombre fini des neurones, une sortie et une cellule qui effectue un calcul d'une fonction mathématique et qui envoie les résultats pour qu'ils servent pour une entrée d'un autre neurone ou la sortie finale du réseau. Chaque entrée est pondérée par un poids. Ces pondérations sont appelées les poids synaptiques.

### IV.2.2 Définitions :

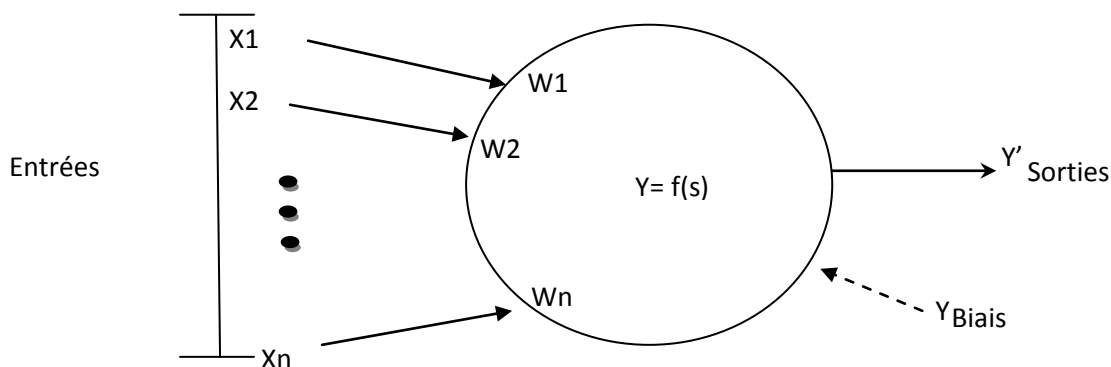
Un neurone est une fonction non linéaire, paramétrée, à valeur bornée [43].

L'élément de base d'un réseau de neurones est bien entendu, le neurone artificiel. Un neurone contient deux éléments principaux:

Un ensemble de poids associés aux connexions du neurone, et

Une fonction d'activation.

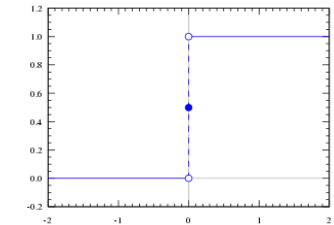
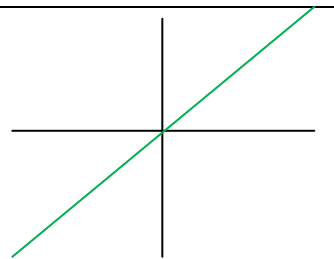

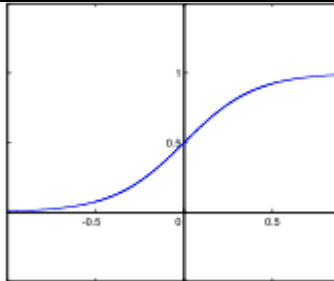
Les valeurs d'entrées sont multipliées par leur poids correspondant et additionnées pour obtenir une somme  $S$  [44].



**Figure IV.2:** Le neurone artificiel générique [44]

## Les réseaux de neurones

De cette définition on peut déduire qu'un neurone fait des calculs, donc c'est une unité qui exécute une fonction mathématique. Les fonctions mathématiques utilisées sont différentes. Le développeur de réseaux de neurones utilise pour une raison ou une autre une fonction mathématique qu'il juge mieux adaptée à son problème.

Nom du Fonction	Expression mathématique	Forme géométrique
fonction de Heaviside (échelon, étape)	$\forall x \in \mathbb{R}, H(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0. \end{cases}$	
Fonction linéaire	$\forall x \in X, f(x) = x \in X$	
Tangente hyperbolique	$\text{th}(x) = \frac{\text{sh}(x)}{\text{ch}(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
Sigmoïde	$f(x) = \frac{1}{1 + e^{-x}} \quad \forall x \in \mathbb{R},$	

**Tableau IV 1:** Quelques exemples de fonction d'activation [45]

Réseau de neurones: un réseau de neurones se compose d'un groupe d'unités de traitement qui communiquent en envoyant des signaux les unes aux autres sur un grand nombre de connexions pondérées [46].

### IV.3 Propriété des réseaux de neurone:

Dans le livre traitement parallèle distribué (parallel distributed processing), McClelland et Rumelhart [47] identifient huit caractéristiques de bases des réseaux de neurones :

- Un ensemble de neurones
- Un état d'activation pour chaque neurone (actif, inactif, ...)
- Une fonction de sortie pour chaque neurone ( $f(S)$ )
- Un modèle de connectivité entre les neurones (chaque neurone est connecté à tous les autres, par exemple)
- Une règle de propagation pour propager les valeurs d'entrée à travers le réseau vers les sorties
- Une règle d'activation pour combiner les entrées d'un neurone (très souvent une somme pondérée)
- Une règle d'apprentissage
- Un environnement d'opération (dans notre cas c'est le robot mobile)

Le comportement d'un réseau et les possibilités d'application dépendent complètement de ces huit facteurs et le changement d'un seul d'entre eux peut changer le comportement du réseau complètement [44].

### IV.4 Architecture des réseaux de neurones :

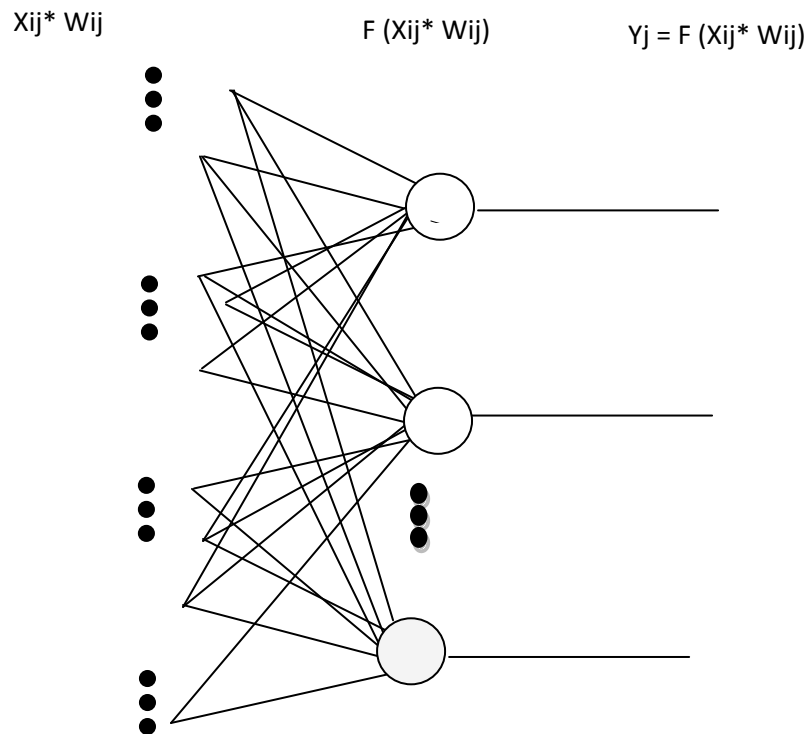
Il est possible de classer les réseaux de neurone selon le nombre de couches (mono couche, multi couche) ainsi que la méthode de propagation d'information dans le réseau. Selon ces deux critères, on cite les classes suivantes des réseaux de neurone:

**A- Nombre de couches :** un réseau de neurone peut être constitué de plusieurs couches : la première est appelée couche d'entrée, la dernière est appelée couche de sortie, et entre les deux on trouve un nombre fini de couches appelées couches cachées. Une couche est construite elle-même d'un nombre fini de neurones.

**B- Propagation de l'information :** la méthode de propagation de l'information dans un réseau de neurone diffère selon le besoin ; les données de sortie peuvent être utilisées comme étant des entrées, qu'on appelle rétro propagation de données ; dans d'autres cas on ne les utilise pas.

## IV.4.1 Réseau feedforward mono couche :

Dans ces réseaux, les informations ne sont pas « rétro-propagées » ; de plus, elles sont composées d'une seule couche qui est à la fois une couche d'entrée et de sortie [10].



**Figure IV.3:** Un réseau de neurone feedforward monocouche ( $X_{ij} * W_{ij}$  le vecteur d'entrée,  $f$  la fonction d'activation est  $Y_j$  le vecteur de sortie)

## IV.4.2 Réseaux feedforward multicouche:

Les informations ne sont pas « rétro propagées » et elles sont composées de plusieurs couches. Deux couches différentes pour l'entrée et la sortie et  $n$  couches cachées [10].

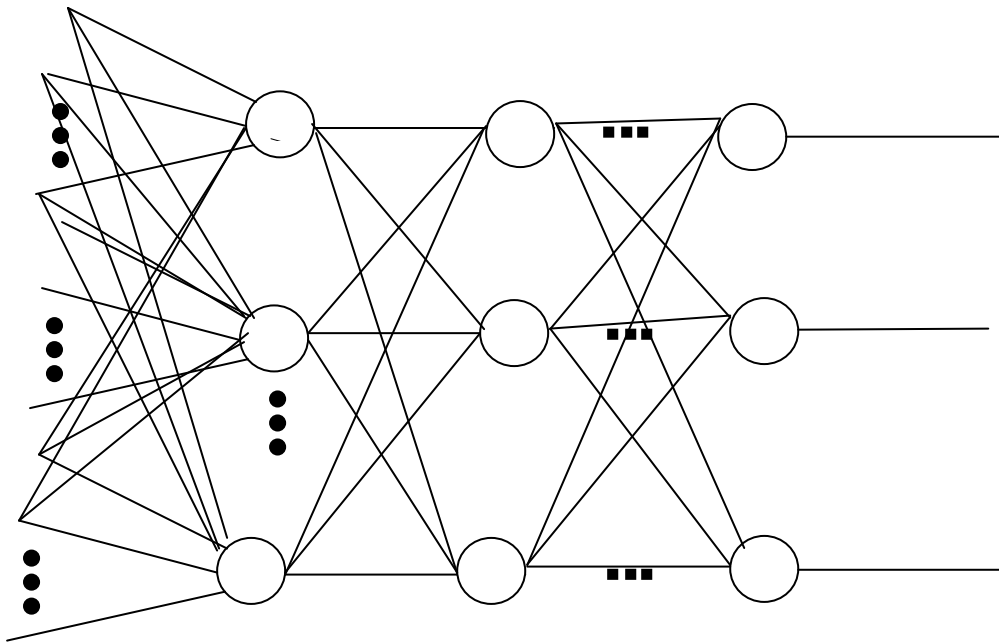


Figure IV.4: Un réseau de neurones feedforward multicouche

### IV.4.3 Réseaux récurrents :

Ce type de RN est caractérisé par la rétro propagation de l'information. Les résultats de sortie au moment  $n$  sont utilisés comme étant entrée au moment  $n+1$  ; ces réseaux peuvent être mono couche ou multicouche [10].

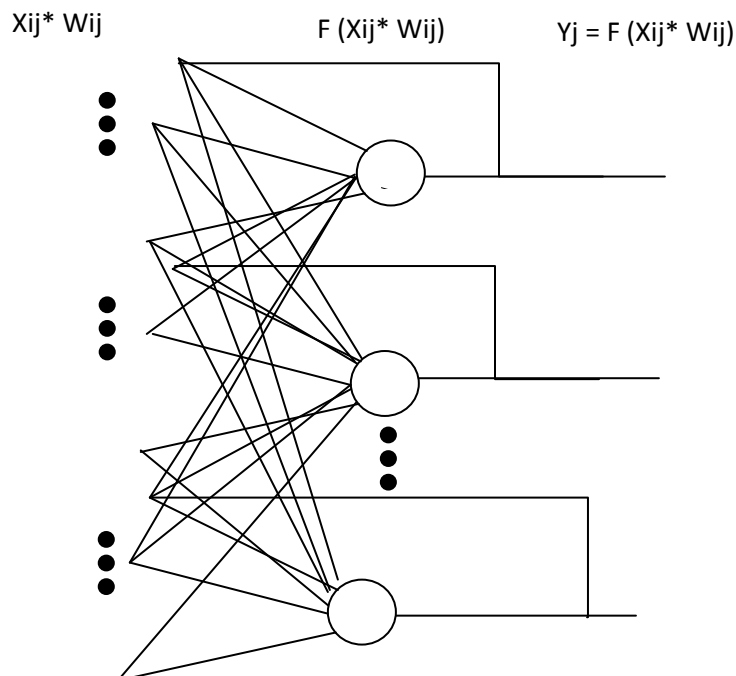


Figure IV.5: Un réseau de neurone récurrent monocouche

Remarque :

Il existe des réseaux totalement connectés et d'autres partiellement connectés. Un réseau est dit totalement connecté si toutes les unités d'une couche sont connectées à toutes les unités de la couche suivante ; dans le cas inverse, le réseau est appelé partiellement connecté.

## **IV.5 Paradigme d'apprentissage :**

Les réseaux de neurones sont caractérisés par leur capacité d'apprendre. Ceci est assuré grâce à un algorithme d'apprentissage. Dans cette section, on va définir les concepts de base de l'apprentissage et discuter brièvement quelques algorithmes d'apprentissage.

La méthode d'apprentissage peut être différente d'un réseau à un autre, selon l'architecture des réseaux de neurone et de la nature du problème. Par conséquent, il existe plusieurs algorithmes d'apprentissage.

### ***IV.5.1 Définitions d'apprentissage de réseaux de neurone:***

Définition 1 : l'apprentissage est le processus par lequel les paramètres libres d'un réseau de neurones sont adaptés via un processus de stimulation par l'environnement dont le réseau est embarqué. Le type d'apprentissage est déterminé par la manière dont laquelle le changement de paramètres est effectué [10].

Définition 2 : L'apprentissage est le processus qui a pour objectif de formuler les règles explicites qui permettent des généralisations, puis sauter d'une manière brève du cas particuliers aux cas générales [47].

Définition 3 : l'apprentissage peut être compris comme un changement dans les capacités ou le comportement d'un organisme provoqué par l'expérience [50].

Grâce à l'apprentissage, le réseau de neurones passe de quelques exemples spécifiques aux traitements des cas génériques. L'activation d'un réseau par des données venant de son environnement de fonctionnement permet de définir une première vue ; l'algorithme d'apprentissage va formuler les règles explicites qui lui permettent la généralisation ; la formulation des règles est effectuée par le changement des poids synaptiques qui mène au changement du comportement du réseau ; le changement est effectué par un ensemble d'itération ce qui figure l'expérience.

### ***IV.5.2 Propriétés d'un algorithme d'apprentissage:***

Pour conduire un apprentissage de bonne qualité et augmenter la performance du système, l'algorithme d'apprentissage doit avoir les propriétés suivantes [51]:

Généralité de tâche : améliore la performance des systèmes pour toutes les tâches du domaine.

Généralité de connaissances : toutes les connaissances disponibles sur le domaine peuvent être utilisées. Ces connaissances peuvent être représentées sous forme d'exemples, d'instructions ou d'expériences anciennes.

Généralité de l'aspect : ça peut améliorer la performance de tous les aspects du système.

Transfert de l'apprentissage : ce qui est appris dans une situation doit être utilisé dans d'autres situations.

Les trois premières propriétés permettent la généralisation. Afin de passer d'un cas élémentaire vers le cas général, il faut passer par des tâches spécifiques vers des tâches génériques, par des connaissances spécifiques vers les connaissances générales, et traiter aussi les différents aspects du domaine.

La dernière propriété permet la réutilisation des connaissances acquises et des anciennes expériences. Cela permet de se référer à des connaissances et des expériences afin de résoudre une certaine problématique.

Un algorithme qui assure ces quatre propriétés donne une bonne performance au système où il est déployé. Ceci est dû à sa capacité de généralisation et de réutilisation des ressources soit sous forme de connaissances ou bien d'expériences.

### ***IV.5.3 Quelques algorithmes d'apprentissage:***

Tous les algorithmes d'apprentissage changent les poids synaptiques de connexions entre différents neurones. La façon de changement des poids synaptiques diffère d'un algorithme à un autre.

#### **IV5.3.1 Apprentissage par correction d'erreur:**

Dans ce type d'apprentissage, l'erreur est notée par  $e_k(n)$  ; le calcul de l'erreur dépend de deux valeurs : la sortie de réseau de neurones notée  $y_k(n)$ , qui est appelée aussi sortie réelle et de la sortie désirée notée  $d_k(n)$  ; alors l'erreur  $e_k(n)$  sera calculée comme suit :

$$e_k(n) = d_k(n) - y_k(n).$$

Avec  $k$  est l'indice du neurone, on dit le neurone  $k$ .

L'objectif du calcul d'erreur est d'appliquer une séquence d'ajustements de poids synaptiques. Ces ajustements sont en fonction de l'erreur, avec comme objectif le fait de rendre  $y_k(n)$  proche de  $d_k(n)$  ou par autre terme minimiser l'erreur. Cet objectif est atteint en minimisant la fonction coût  $\xi(n)$  définie par :

$$\xi(n) = \frac{1}{2} \sum_k e_k(n)^2.$$

La minimisation de la fonction coût conduit à une règle appelée règle de delta ou règle de Widrow-Hoff [REF] ; cette règle est donnée par :

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n)$$

$W_{kj}$  = poids synaptique de neurone  $k$

Avec l'entrée  $x_j(n)$  au moment  $n$  et  $\eta$  est une constante positive appelée pas d'apprentissage. Après le calcul du delta, les nouveaux poids synaptiques sont calculés par :

$$W_{kj}(n+1) = W_{kj}(n) + \Delta w_{kj}(n)$$

### IV.5.3.2 Apprentissage basé sur la mémorisation:

Dans ce type d'apprentissage, toutes les expériences passées (ou la majorité) sont enregistrées explicitement dans une mémoire large. Les expériences enregistrées sont jugées justes et sont formées de couple entrée-sortie  $(X_i, D_i)$ , où  $X_i$  est le vecteur d'entrée et  $D_i$  les sorties désirées correspondantes.

Lorsque un nouveau vecteur d'entrée  $X_{test}$  est présenté au réseau de neurones, l'algorithme recherche et analyse les données aux voisinages de  $X_{test}$ .

Tous les algorithmes de ce type ont deux composants communs :

1. Critère de définition des voisins du nouveau vecteur  $X_{test}$
2. Règle d'apprentissage appliquée pour l'entraînement des exemples au voisinage du vecteur  $X_{test}$ .

La règle la plus connue et la plus simple dans ce type d'apprentissage définit un calcul de distance entre le vecteur  $X_i$  et le vecteur de test  $X_{test}$  ; le voisin est appelé

$$X_N = \{X_1, X_2, \dots, X_N\}$$

Alors le voisin est défini par:

$$\text{Min}_i d(X_i, X_{test}) = d(X_N, X_{test})$$

Tel que  $d(X_i, X_{test})$  est la distance euclidienne entre les vecteurs  $X_i$ , et  $X_{test}$ .

Le minimum de différentes distances calculées est le voisin. Le vecteur  $X_i$  est dit classification du vecteur  $X_{test}$ .

### IV.5.3.3 Apprentissage de Hebb :

La règle de Hebb est la plus ancienne des règles d'apprentissage. Elle est modélisée par la formule mathématique suivante:

$$\Delta W_{kj}(n) = F(Y_k(n), X_j(n))$$



Où  $W_{kj}$  est le poids du neurone  $k$  au moment  $n$ ,  $X_j(n)$  le vecteur d'entrée et  $Y_k(n)$  est la sortie de neurone qui est sujet d'adaptation de poids.

$F(Y, X)$  est la fonction de l'entrée et de sortie du neurone ; alors la forme la plus simple de la règle de Hebb est:

$$\Delta W_{kj}(n) = \eta Y_k(n) X_j(n)$$

Les nouveaux poids sont calculés à partir de cette formule par :

$$W_{kj}(n+1) = W_{kj}(n) + \Delta w_{kj}(n)$$

Les chercheurs ont considéré les différentes règles d'apprentissage comme étant des variantes de la règle d'Hebb [46].

### IV.5.3.4 Apprentissage compétitif:

Ce type d'apprentissage est comme son nom l'indique basé sur une « compétition » entre les différents neurones de réseaux. A la fin, il y a un neurone vainqueur. La correction des poids d'effectue seulement sur les différents poids qui affecte ce neurone.

Rumelhart et Zipser définissent trois éléments de base pour une règle d'apprentissage compétitive:

1. Un ensemble de neurones qui sont tous les mêmes sauf pour quelques poids synaptiques distribués aléatoirement et qui répond différemment à l'ensemble des entrées.
2. Une limite imposée sur la force de chaque neurone.
3. Un mécanisme qui permet une compétition entre les neurones, seulement un neurone par groupe est actif à un moment donné.

Les neurones individuels des réseaux vont apprendre à être spécialisés sur un ensemble de formes similaires. Ils deviennent des capteurs ou des détecteurs de particularité ou des caractéristiques pour les différents ensembles d'entrées.

Dans la forme la plus simple d'apprentissage compétitif, les réseaux de neurones contiennent une seule couche de neurones de sortie, chaque neurone est totalement connecté à chaque nœud d'entrée.

Pour un neurone  $k$  qui est considéré le vainqueur, il est « induced ». Un champ local  $v_k$  pour une forme d'entrée spécifique  $X$  doit être le plus grand à travers les neurones dans tout le réseau, la sortie des neurones vainqueur,  $Y_k$  est mis égale à un, la sortie de tous les neurones qui perdent la compétition sont mis égal à zéro :

$$\left\{ \begin{array}{l} Y_k = 1 \text{ si } v_k > v_j \\ Y_k = 0 \quad \text{sinon.} \end{array} \right.$$

$v_k$  représente les actions combinées de toutes les entrées de neurone K.

L'apprentissage est effectué seulement sur les neurones vainqueurs, les neurones qui perdent la compétition ne seront pas mis à jour.

Un neurone vainqueur est dit actif et tous les poids synaptiques des connexions entrant sont mis à jour.

Le changement des poids synaptiques est défini par la règle standard d'apprentissage compétitif:

$$\left\{ \begin{array}{ll} \Delta w_{kj}(n) = \eta (x_j - w_{kj}) & \text{si le neurone k est vainqueur.} \\ \Delta w_{kj}(n) = 0 & \text{sinon.} \end{array} \right.$$

Dans ce type d'apprentissage, à chaque itération un seul neurone est activé à la fois et les mises à jour sont effectuées localement sur les poids associés à ce neurone. A la fin du processus d'apprentissage, tous les neurones sont adaptés.

## ***IV.5.4 Types d'apprentissage:***

L'apprentissage est effectué par un algorithme spécifique. Il existe plusieurs façons d'appliquer un tel algorithme ; la raison et la diversité des problèmes qui nécessitent des procédures de résolution de problèmes différents. On peut citer les types d'apprentissage suivants:

### **IV.5.4.1 Apprentissage supervisé:**

Dans ce cas le réseau de neurone entraîné par un ensemble de couple entrée-sortie. Ces couples sont fournis par un enseignant externe ou par le système contenant le réseau de neurones [46].

Dans ce type d'apprentissage on cherche à imposer aux réseaux un fonctionnement donné en forçant à partir des entrées qui lui sont présentées, les sorties de réseaux à prendre des valeurs données en modifiant les poids synaptiques. Les réseaux se comportent alors comme un filtre dont les paramètres de transfert sont ajustés à partir des couples entées sorties [53].

L'adaptation des paramètres du réseau s'effectue à partir d'un algorithme d'optimisation, l'initialisation des poids synaptiques étant le plus souvent aléatoire [53].

### **IV.5.4.2 Apprentissage non supervisé:**

Dans ce cas, l'unité de sortie est entraînée pour répondre à des clusters. Dans ce paradigme le système est supposé capable de découvrir statiquement la population [46].

Dans ce cas, des exemples ou « prototypes » ou « patrons » sont présentés au réseau qu'on laisse s'auto-organiser au moyen de lois locales qui régissent l'évolution des poids synaptiques. Ce mode d'apprentissage est aussi appelé « apprentissage par compétition » [53].

### **IV.5.4.3 Apprentissage semi supervisé:**

L'apprentissage semi supervisé se situe au milieu des deux précédents. Il s'applique dans le cas d'existence de quelques couples d'entrées sorties ; l'entraînement se fait en utilisant ces couples, pour l'apprentissage de réseau de neurones [54].

### **IV.5.4.4 Apprentissage par renforcement:**

Ce type d'apprentissage basé sur l'expérience possède le pouvoir de rendre la machine capable de mapper les expériences aux actions. L'action à effectuer n'est pas connue pour la machine; alors, elle doit essayer le maximum d'actions possibles. Une conséquence possible et un signal cadeau, ou un signal punition, selon les conséquences de l'action, la machine va apprendre. Les cas futurs sont traités en faisant références aux anciennes expériences. Dans ce cas l'apprentissage est effectué en interaction [55].

## **IV.6 Représentation de connaissance dans un réseau de neurones :**

Les réseaux de neurones sont capables de stocker les connaissances. La question qui se pose c'est comment les connaissances circulent dans un réseau de neurone.

### ***IV.6.1 Définitions de connaissances :***

Une définition du mot connaissance est la suivante :

1. Une connaissance est obtenue lorsque les régularités, les organisations et les relations sur les objets physiques sont découvertes par le cerveau, les régularités sont produites par les activités mathématiques [49].

Une deuxième définition qu'on peut citer ici:

2. Une connaissance se réfère aux informations enregistrées ou aux modèles utilisés par une personne ou par une machine pour interpréter, prédire et proprement répondre à l'environnement externe [48].

Ces deux définitions présentent deux formes de représentation d'une connaissance : la forme mathématique présentée par la première définition et la forme linguistique présentée par la deuxième définition. La forme mathématique concrétise une connaissance par une loi appelée une relation, une organisation, ou régularité. Cette loi est modélisable par une formule mathématique. Cette définition est en plein accord avec le concept de présentation de connaissance à l'intérieur d'un réseau de neurones, étant donné que ces réseaux captent les régularités par les formules mathématiques (les poids synaptiques, les règles d'apprentissage, et les fonctions d'activation).

Les régularités d'un domaine spécifique sont supposées invariables pour tous les cas de ce domaine. Lorsqu'elles sont captées par un algorithme d'apprentissage et enregistrées dans les poids synaptiques, la généralisation sera possible.

La deuxième définition supporte une représentation logique de la connaissance sous forme d'un modèle qui peut être utilisé pour des situations précises.

### ***IV.6.2 Conception des données pour un réseau de neurones:***

Un réseau de neurones code les connaissances d'une manière différente par rapport aux autres méthodes. Ceci réside dans des valeurs numériques qui peuvent être non significatives pour une personne qui n'a pas suffisamment de connaissances sur le problème. Un des problèmes fondamentaux qui se pose est comment choisir un ensemble de connaissances qui reflète fidèlement la problématique et qui conduit à une bonne solution.

Probablement il n'existe pas une réponse exacte à cette question, mais on va essayer de construire une compréhension générique de la conception et la mise en place d'un réseau de neurones afin de guider d'une manière abstraite la façon dans laquelle on fait la sélection d'une population de données.

Une architecture est choisie et une population est présentée (sous forme de couple entrée-sortie dans le cas le plus général). Cette population représente des exemples pour entraîner le réseau de neurones. A la fin de cette étape, les poids sont adaptés. Cette étape s'arrête lorsqu'on obtient une erreur minimale [48].

Un ou plusieurs échantillons différents sont présentés pour le réseau de neurones afin de tester la validité des poids et que le réseau de neurones d'apprendre proprement, d'où il est capable de généraliser [48].

Après le test, l'étape suivante est la validation. Elle sert à garder les poids justes et déployer le réseau de neurones dans son environnement final pour qu'il soit opérationnel [48].

Ainsi est le processus de construction d'un réseau de neurones d'une manière générale ; les règles suivantes peuvent aider à repérer la population de données correctes à la problématique :

Règle 1 : les entrées similaires des classes similaires doivent usuellement produire des représentations similaires à l'intérieure du réseau et alors doivent être classées comme étant appartenant à la même catégorie [48].

Règle 2 : les objets catégorisés comme étant appartenant à des classes différentes doivent donner des représentations différentes dans le réseau [48].

Règle 3 : si une fonction particulière est importante, elle doit avoir un grand nombre de neurones pour sa représentation dans le réseau [48].

Règle 4 : les informations préalables et les invariances doivent être intégrées dans la conception du réseau, simplifiant ainsi la conception du réseau pour ne pas avoir à l'entraîner [48].

Remarque : il est très clair que ces quatre règles sont pour les problèmes de reconnaissance de formes.

Discussion des quatre règles :

Règle 1: pour classer une forme on doit définir et classer ces caractéristiques, les caractéristiques des objets de la même classe sont similaires à certain degré, les caractéristiques d'un objet qui subissent un traitement spécifié, donnent des résultats précis, d'autre caractéristiques similaires, s'ils passent par le même traitement vont donner des résultats similaires, des ensembles de données similaires en entrées d'un réseau de neurones vont produire des sorties similaires, ce que permet de les référer comme étant de la même catégorie. Cela est du a la structure de réseau de neurones qui ne change pas, les différents ensembles de données des objets similaires vont passer par les mêmes opérations (même poids, même fonction d'activation, et même nombre de neurones).

Règle 2: cette règle est l'inverse de règle 1. L'activation d'un réseau de neurones par des données différentes va produire des ensembles de sorties différentes, ce qui implique leur classement comme étant appartenant à des classes différentes.

Règle 3: une fonction importante à la problématique doit être représentée fortement à l'intérieur du réseau. Pour assurer cette représentation, un nombre important de neurones sera réservé à cette fonction ; les connaissances pour figurer une telle fonction doivent être représentées d'une manière qui la reflète fidèlement. Une manière pour assurer la bonne représentation est de la modéliser par un maximum de connaissances possible, ce qui indique un nombre élevé de connexions et des neurones dédiés à la représentation de cette fonction.

Règle 4 : cette règle expose l'importance de choix de la population. Les connaissances précédentes sur un domaine peuvent nous aider à éviter l'entraînement du réseau de neurones par la sélection des poids synaptiques dès le début. Théoriquement, ceci est approprié, mais dans la pratique il est assez difficile de sélectionner un ensemble de données et des poids synaptiques justes de la première fois, dû à la complexité de certains problèmes et même à la complexité de quelques architectures neuronales. En effet, il est nécessaire de passer par l'étape d'apprentissage d'un réseau de neurone.

### **IV.7 Utilisation des réseaux de neurones pour la planification et l'évitement d'obstacles dans la robotique mobile :**

Il existe plusieurs manières pour utiliser les réseaux de neurones pour résoudre le problème de planification et d'évitement d'obstacles d'un robot mobile. On peut citer ici les efforts suivants.

Une implémentation d'une méthodologie basée sur d'autres approches par les réseaux de neurones notamment [56] qui proposent une implémentation neuronale basée sur la méthode de champ potentiel.

La création d'une nouvelle méthodologie en utilisant les réseaux de neurones [19] [58], pour développer une approche neuronale simple.

La modélisation cognitive du robot par les réseaux de neurones [59] ; le principe est de décomposer le problème en sous-problèmes, la résolution se fait par l'implémentation de plusieurs populations neuronales, chacune des populations neuronales est dédiée à résoudre un des sous-problèmes.

Ces approches sont différentes dans le principe, mais elles ont certains aspects en commun, on va essayer de décrire d'une manière générale les implémentations citées ci-dessus.

### ***IV.7.1 Implémentation d'approche classique par les réseaux de neurones:***

Dans ce travail [56], les auteurs proposent une implémentation basée sur la méthode champ potentiel. Le principe sert à connecter le point de départ au point d'arrivée par une droite. Dans le cas où le segment de la droite passe par un obstacle, de nouveaux points sont calculés par des formules spécifiques. Dans le cas des autres segments, l'objectif sera de calculer des nouveaux points pour optimiser l'ancien chemin. Chaque obstacle est décrit par un réseau de neurones. Un réseau de neurones est stimulé par les coordonnées de l'obstacle.

La limitation principale de cette méthode réside dans le fait que chaque obstacle est représenté par un réseau de neurones différents ; dans une situation contenant un nombre élevé d'obstacles, la complexité des calculs, ainsi que l'implémentation, sera plus élevée, ce qui implique un coût élevé en fonction du temps et de calculs.

### ***IV.7.2 Création d'une méthodologie neuronale:***

L'architecture proposée [57] est une carte discrète organisée topographiquement. Un neurone représente une position ou une configuration dans l'espace de travail et chaque neurone est connecté seulement à ses voisins ; il répond seulement au signal venant de ses voisins.

L'activité de la destination est propagée dans tout le réseau. L'activité des obstacles, statiques ou dynamiques, est représentée localement (dans quelques neurones locaux). Partant d'une position initiale, la prochaine position est calculée à partir de la position courante ; si la prochaine position est égale à la position courante, c'est que le robot n'a pas changé la position. Le réseau de neurones détermine quelle est la prochaine position du robot.

Cette méthodologie ne requiert aucun apprentissage, ce qui réduit la complexité des calculs, et malgré qu'aucun processus d'apprentissage ne soit porté sur le réseau, les simulations ont prouvé une efficacité de cette méthode.

Probablement l'avantage de cette méthodologie est aussi l'inconvénient : le concept le plus important dans les réseaux de neurones est l'apprentissage, donc plusieurs questions se posent sur un réseau de neurones mis en place sans passer par l'étape d'apprentissage.

### ***IV.7.3 Modélisation cognitive par les réseaux de neurones:***

C'est la forme la plus évoluée d'utilisation des réseaux de neurones qui applique la théorie biologique à la résolution de la problématique ; celle-ci dernière est découpée en un ensemble fini de sous-problèmes, chacun est résolu à part par une population neuronale, coïncidant avec la distribution des tâches cognitives des robots.

Un exemple typique de cette approche [59] sert à appliquer une théorie d'apprentissage biologique appelée conditionnement opérant. Le principe de ce type d'apprentissage sert à associer des cadeaux à des comportements supportés, et associer des punitions aux comportements non souhaités. Lorsque le robot se rapproche de l'objectif, un signal de motivation (cadeau) est stimulé à l'intérieur du réseau de neurones ; dans le cas où il se rapproche d'un obstacle le robot sera puni par un autre signal. Le robot doit mémoriser les différentes situations antérieures pour les utiliser dans des cas postérieurs ; cette méthode permet la définition de deux comportements principaux qui permettent la navigation en évitant

les obstacles et en se rapprochant de la destination ; ces deux comportements sont le comportement d'éviter les obstacles, et le comportement de se rapprocher de but.

Le processus d'apprentissage ici est un apprentissage par renforcement. Il permet un apprentissage en mémorisant les différentes expériences. Quelque soit le résultat d'une action (punition, cadeau), le robot mobile va construire une compréhension qui va lui permettre de distinguer entre les différentes situations et puis sélectionner l'action la plus pertinente à une situation future.

Cette approche bénéficie d'un degré d'explication plus élevé. Elle permet de construire des populations (agents) neuronales spécialisées qui exécutent des fonctionnalités précises, ce qui explique le fonctionnement du réseau de neurones d'une manière générale. L'autre avantage de cette approche c'est qu'elle adopte une théorie d'apprentissage biologique, d'où un processus d'apprentissage plus naturel qui permet de définir les actions d'une manière plus juste grâce aux expériences. D'autre part, l'accumulation des expériences améliore la performance du système par la minimisation du taux d'erreur à travers les expériences successives.

### ***IV.7.4 Discussion :***

La diversité des travaux et des vues a enrichi le domaine de planification de la trajectoire et d'évitement d'obstacles par la proposition de différentes méthodes. Cela permet d'identifier un ensemble de critères pour assurer la qualité d'une telle solution en évitant les problématiques d'une solution antérieure. Selon notre étude on peut identifier quelques critères pour assurer une bonne méthodologie neuronale :

Le nombre des neurones doit être indépendant de l'environnement du travail : ce qui permet le déploiement du robot dans des environnements différents sans aucune difficulté.

Le nombre des neurones doit être indépendant du nombre des obstacles : ce qui permet la définition d'un nombre fini acceptable de neurones pour la représentation des différents obstacles, et réduit la complexité des calculs nécessaires.

L'algorithme d'apprentissage utilisé doit refléter au maximum possible un processus d'apprentissage naturel: alors une interaction entre le robot et son environnement maximal, qui permet de lier les actions aux situations.

Distribution de tâche cognitive : permet la définition de différents comportements intelligents, et assure un degré d'explication plus élevé.

#### **IV.7.4.1 Le nombre de neurones:**

Si le nombre de neurones d'un réseau est lié à l'environnement (surface) ou au nombre d'obstacles, l'implémentation sera plus difficile à cause des changements possibles de l'environnement de travail. Sur un terrain de travail plus large ou qui contient plus d'obstacles, le système doit définir automatiquement l'architecture neuronale après une étude effectuée sur le terrain. Si les informations sensorielles ne sont pas exactes, le risque de faire une erreur est élevé, il est aussi possible de ne pas trouver une solution faisable.

#### **IV.7.4.2 L'algorithme d'apprentissage:**

La qualité de processus d'apprentissage influence les résultats d'application finale. Pour cette raison il est nécessaire de mettre en place une solution meilleure pour l'étape d'apprentissage. Dans le cas de la robotique mobile, on peut s'inspirer de la biologie ; grâce à leur propriété de changement de la position, et de leur mise en place dans des environnements réels, il est préférable de laisser le robot apprendre des événements qui se passent dans son environnement. Cette attitude sert à rendre le processus d'apprentissage plus naturel.

### **IV.7.4.3 La distribution des tâches cognitives:**

L'objectif de distribution de tâches cognitives est la mise en place d'une solution plus intelligente qui permet une interaction de haut niveau entre le robot et l'environnement, ce qui permet la résolution de problèmes plus complexes d'une manière intelligente. Ceci est grâce à la collaboration des différentes populations neuronales. L'identification de tâches cognitives permet de traiter les différents cas possibles, tout en associant chaque tâche cognitive à une population neuronale.

### **IV.8 Conclusion :**

L'approche des réseaux de neurones est une méthode pour modéliser l'intelligence. Ces réseaux sont capables de résoudre des problèmes dans différents domaines. Leur point de fort réside dans leur capacité d'apprentissage. Les réseaux de neurones peuvent être également considérés comme une modélisation mathématique des neurones biologiques.

La compréhension neuronale des connaissances est différente ; une telle connaissance peut être sous la forme suivante :

La couleur bleu est équivalente à la valeur 0.5 ou bien le chemin optimal est retrouvé par le vecteur des poids synaptique  $W$ , d'où un codage différent des connaissances. On peut considérer que la représentation des connaissances dans les réseaux de neurones est plus proche du langage machine, pour cette raison il est difficile de l'interpréter par un être humain en comparaison avec les autres méthodes de représentation de connaissance. Pour cette raison, les réseaux des neurones sont appelés boîte noire. Un des objectifs des travaux de recherches est de comprendre comment les connaissances sont distribuées à l'intérieur de réseau de neurones et comment extraire ces connaissances d'une manière compréhensive par un être humain.

Dans notre travail comprendre la représentation interne des connaissances de notre réseau de neurones est une perspective.



**Chapitre**

**VI**

**Conception et implémentation**

## **V.1 Introduction :**

Les méthodes intelligentes représentent un modèle à capacité élevée de résolution de problèmes dans différents domaines, afin de tester cette capacité, il est nécessaire de les implémenter dans un système réel ou de les simuler grâce à un simulateur ou un programme dédié.

Pour achever notre objectif, il faudra passer par beaucoup de travail de conception et de programmation qui reflète précisément ces méthodes. Dans ce chapitre on va proposer une méthode basée sur les anciens travaux pour pouvoir aller au-delà de ces problèmes et ces limites.

Un bon programme de simulation de ces techniques qui les montrent strictement doit être basé sur une analyse et une conception orientée objet, cela permet la réutilisation des codes sources et offre une vue plus exacte et juste des différentes méthodes intelligentes.

Dans ce chapitre on va essayer de créer une simulation d'une solution proposée en utilisant des programmes conçus selon les normes de l'approche orientée objet et écrit en Java, l'objectif est multiple, en plus de la compréhension de l'approche cognitive, on va profiter aussi d'une implémentation orientée objet et une simulation de la solution, les résultats de travail sont présentés et discutés à la fin du chapitre.

## **V.2 Une méthode pour la planification de la trajectoire et évitement d'obstacle :**

Afin d'arriver à nos objectifs, il est nécessaire de passer par quelques étapes, ça concerne le choix d'une stratégie globale pour la résolution du problème, une telle stratégie sert à établir une solution complète, ceci peut être réalisé par mentionner les objectifs, décrire les processus, les méthodes, et les approches pour atteindre ces différents objectifs. Notre stratégie commence par le choix et la justification d'une approche, puis un développement d'une méthodologie générale pour la résolution du problème, à la fin, un algorithme qui implémente la méthodologie proposée est établi, ce dernier va respecter l'approche choisie au début.

### *V.2.1 Description détaillée des objectifs :*

Le problème de planification et évitement d'obstacles pour les robots mobiles peut se résumer en deux points :

- Changer la position d'une initiale vers une autre finale.
- Eviter les obstacles éventuels.

Et pour une résolution réussie de ce problème, le robot mobile doit :

- Avoir toutes les informations nécessaires
- Traiter ces informations en temps réel
- Posséder un module de décision qui va lui permettre la définition de la prochaine action.

La mise en place d'une solution qui traite les deux points précédents, doit assurer ces trois fonctionnalités, ça concerne le développement d'une structure de données qui permet d'optimiser les informations nécessaires pour le traitement, une telle structure doit représenter les différentes informations sensorielles, les distances entre le robot et les obstacles, la position de l'objectif, l'angle d'orientation du robot... etc.

Le traitement des informations en temps réel est effectué par la méthode sélectionner, la structure de données est formée de toutes les informations nécessaires pour la navigation, les informations en entrées peuvent changer en temps réel, une telle technique doit retourner les résultats dans un temps minimum, la méthode doit être simple pour être capable d'optimiser le temps de traitement des informations.

La solution doit être conçu d'une manière qui permet de retourner une commande sous certaine forme en sorties, l'exécution des sorties (commande) définit la prochaine action, l'exécution de l'ensemble de toutes les commandes donne une solution de la problématique, la méthode alors a deux rôles, traitement des informations et envoi des résultats sous forme de commande.

## Conception et implémentation

---

La planification et l'évitement d'obstacles alors peuvent être accomplis en assurant une implémentation des trois points cités en haut, alors la solution proposée va être construite de ces trois fonctions.

### *V.2.2 L'approche :*

Pour la mise en place de la solution, il est nécessaire de suivre une approche qui permet une résolution de la problématique en assurant la faisabilité, et l'efficacité de la solution, pour cette raison une telle approche doit modéliser fortement les aspects intelligents du robot.

On voit ce problème comme étant un problème de cognition, la mise en place d'une solution de ce point de vue va permettre la modélisation de l'aspect cognitif du robot, ceci permet au robot de s'inter-actionner avec son environnement avec un taux de connaissances important, et une stratégie intelligente pour l'utilisation de ces connaissances, d'où la résolution des problèmes sera plus facile, et le robot sera capable de faire face à des situations complexes avec un minimum de difficulté.

Une approche cognitive est capable de modéliser le problème d'une manière plus intelligente qui rend le robot capable à résoudre des problèmes de haut niveau, un cycle cognitive passe par Cinq étapes fondamentales : sensément, acquisitions, perception, planning, et action, [60] le traitement est effectué après la perception, selon les résultats de ce traitement les bonnes actions sont sélectionnées.

Le déploiement d'une solution cognitive peut être achevé par la mise en place d'un système multi agents, et cela est achevé grâce à la considération des agents comme étant une société d'esprit [39]. C'est-à-dire utiliser plusieurs agents dans une organisation (société) ou modèle (model cognitif) [39].

Les différents acteurs qui forment le système sont appelés « agents », chaque agent est conçu pour résoudre un problème simple de complexité réduite.

La résolution du problème commence par la décomposition de celui-ci en un ensemble de sous-problèmes plus simples, plutôt le plus simple possible, chacun de l'ensemble de ces petits problèmes est résolu appart, la solution du problème entier est obtenue à l'aide des différents résultats partiels retournés par les agents, cela est par la

composition, la sélection, ou la combinaison d'une ou plusieurs résultats. La validité de la solution est testée à la fin par un mécanisme précis.

### *V.2.3 La méthode :*

La méthode présentée ici décrit un mécanisme de raisonnement implémenté par un système multi agent qui fusion plusieurs approches intelligentes, on a distribué les différentes tâches sur les différents agents, le mécanisme permet d'activer un agent ou de désactiver un autre afin d'effectuer un raisonnement ou d'ignorer un autre, chaque action est effectuée en suivant un raisonnement qui est implémenté par un agent spécialisé.

On a décrit le problème de planification par deux points : le changement de la position et l'évitement d'obstacle, le changement de la position peut être exprimé par : aller de la position  $A(x, y)$  et l'angle de l'orientation  $\alpha$  vers la position  $B(x_1, y_1)$  avec l'angle d'orientation  $\beta$ , arrêter lorsque le robot atteint le but  $C(x_n, y_n)$  avec l'angle d'orientation  $\delta$  ; le calcul de la prochaine position se fait à partir de la position courante, les limitations physiques du robot doivent être respectées par la vérification de la convergence des points, par autre terme le point suivant est un voisin du point courant, de plus l'angle de navigation doit être définie d'une manière qui permet de se rapprocher de l'objectif, c'est dire que le robot doit être orienté vers l'objectif, ou avoir une orientation proche, il ne doit pas être orienté dans la direction opposée, cela peut mener le robot à s'éloigner de l'objectif au lieu de s'approcher.

Dans les régions d'obstacles, les points sont identifiés avec une condition, le point ne doit pas être trouvé dans la surface de l'obstacle, l'agent doit posséder un bon mécanisme qui change l'orientation du robot dans les régions proches d'obstacles, il doit permettre aussi l'identification l'orientation de l'objectif après avoir dévié de la première orientation, cela permet au robot d'éviter les obstacles et continuer son chemin jusqu'à atteindre son objectif.

Afin de détecter les régions libres et les régions d'obstacles, il est nécessaire de définir une méthode pour représenter l'environnement du travail.

### V.2.3.1 Distribution de la tâche cognitive :

On a défini les actions de base pour réussir la tâche de planification et évitement d'obstacles, on doit maintenant faire une distribution de la tâche cognitive sur les différents agents, autrement dit, chaque tâche doit être effectuée par un agent dédié.

Les différentes actions définies précédemment peuvent être considérées comme étant des tâches cognitives d'un modèle cognitif, selon l'analyse effectuée, on peut lister les différentes tâches à accomplir comme suit :

- A chaque moment  $t$  du temps de navigation, changer la position  $A(x, y)$  avec l'angle d'orientation  $\alpha$  vers  $B(x_1, y_1)$  avec l'angle d'orientation  $\beta$ . (le changement de la position et l'orientation doit mener à éviter un obstacle ou à atteindre l'objectif)
- Identifier la nature de l'environnement (grand, moyen, et petit)
- Identifier la région où le robot se trouve (dans chaque moment de navigation (centre, ou sud est, sud ouest, nord est, et nord ouest))
- Faire une représentation partielle de l'environnement

Chacune de ces tâches est effectuée par un agent spécialisé, de plus un agent de contrôle doit être déployé pour permettre l'activation d'un agent ou la désactivation d'un autre agent.

En conséquence un ensemble d'agents différents sont proposés, les résultats de sortie venant des différents agents forment une solution de notre problématique.

### V.2.3.2 L'algorithme :

Avant de donner une description du fonctionnement du système, on va donner un nom ou un identifiant à chaque agent et lui attribuer une tâche :

- Agent-Navi : pour le changement de la position et l'angle de navigation.
- Agent-NatEnv : la nature de l'environnement (grand, moyen, petit).
- Agent-RegNavi : identifier la région de navigation (SE, NE, SO, NO).

## Conception et implémentation

---

Le fonctionnement de ce système peut être décrit par l'algorithme suivant :

Algorithme Navigation

Begin

Start Agent-NatEnv // reconnaissance de la nature de l'environnement.

Position courant = A (X, Y)

Position Final = B (X', Y')

While (position courant  $\neq$  position final)

Start Agent-RegNavi // identification de la région de la navigation.

    Get Agent-RegNavi output

        Agent-Navi = générer N chemin, avec certain critère

        Agent-Navi = comparer les fonctions de fitness

        Agent-Navi = sélectionner la meilleure solution

End.

La conception de chacun de ces agents est faite apart, un agent doit être le plus simple possible pour réduire le temps d'exécution et la complexité du système.

Remarque : cet algorithme est l'agent de contrôle.

### **V.3 Conceptions des agents :**

#### ***V.3.1 Reconnaissance de la nature de l'environnement :***

Le premier agent à mise en place est un agent responsable de la reconnaissance de la nature de l'environnement, l'agent envoi en sortie une classification de l'environnement courant selon la taille de ce dernier, la taille peut être petite, moyenne, ou grande.

L'agent a pour entrées les coordonnées (x, y) du point le plus loin du sud est, et les coordonnées (x, y) du point le plus loin du nord ouest (en quelque sorte on va calculer le

## Conception et implémentation

---

diagonal), l'environnement est dit petit si ces deux points sont proches, dans le cas contraire où ces deux points sont éloignés, l'environnement est considéré comme étant grand, dans le troisième cas si cette distance est moyenne, on dit que l'environnement est de taille moyenne.

L'objectif de cet agent est de nous aider à représenter l'environnement, et de donner une approximation de l'erreur.

Afin d'arriver à classer les différents environnements où le robot peut être déployé un classificateur flou est mis en place.

Pour faciliter le travail le plan est associé à un repère orthonormé, le repère est mis exactement au milieu de l'environnement, on calcule ensuite un segment de la fonction  $f(x)=x$  ; ce segment est identifi par les bornes du repère (ou de l'environnement).

La formule suivante est utilisée pour calculer la diagonale

$$diagonale = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$

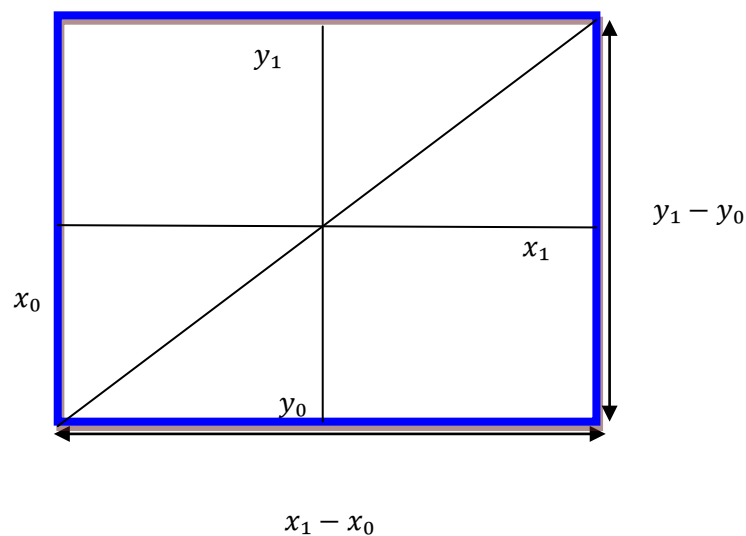
Avec :

$x_1$ : le plus grand abscisses (borne supérieure du repère, axe des x.)

$y_1$ : le plus grand ordonnées (borne supérieure du repère axe des y).

$x_0$ : le plus petit abscisses (borne inférieure du repère axe des x).

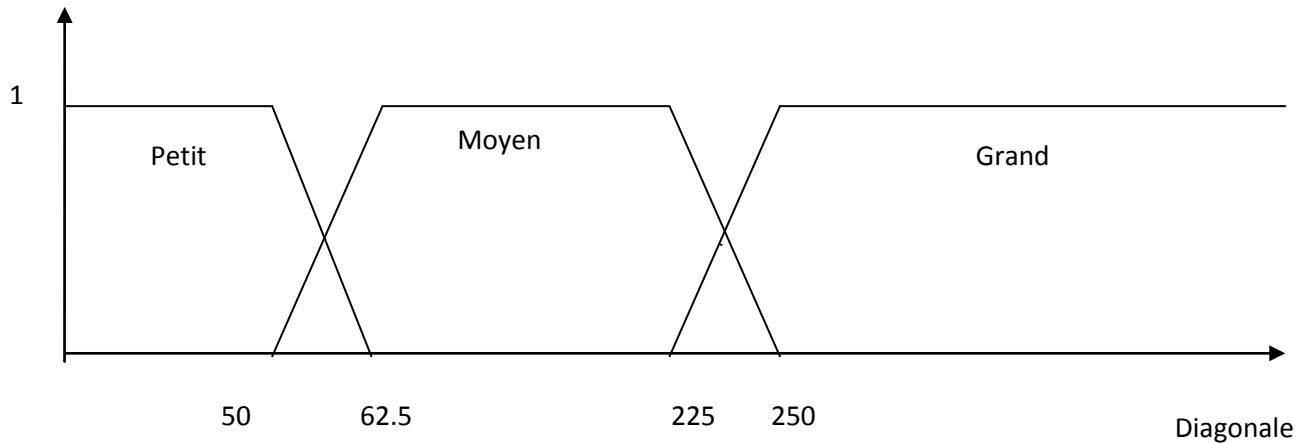
$y_0$ : le plus petit ordonnées (borne inférieure du repère axe des y).



**Figure V.1:** Méthode de calcul de diagonale



Selon la diagonale les environnements sont classés, le schéma suivant présente les sous-ensembles flous :



**Figure V.2 :** Sous ensembles flous d'un environnement de navigation

Les règles floues suivantes sont utilisées pour classer des différents environnements

- Si diagonale  $< 50$  alors environnement petit.
- Si  $50 < \text{diagonale} < 62.5$  calculer le degré d'appartenance par la projection sur l'axe des y (\*100 pour calculer le pourcentage)
- Si  $60 < \text{diagonale} < 225$  alors environnement de taille moyenne.
- Si  $225 < \text{diagonale} < 250$ , calculer le degré d'appartenance par la projection sur l'axe des y (\*100 pour calculer le pourcentage).
- Si diagonale  $> 250$  alors environnement est grand.

Cet agent va aussi aider à identifier la taille de la région centre de chaque environnement de navigation.

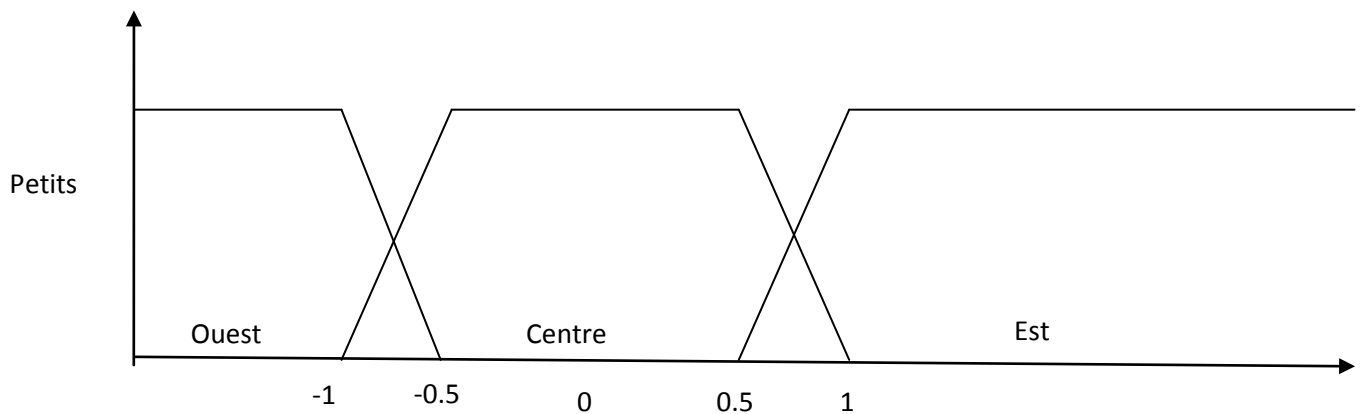
### ***V.3.2 La détection de la région de la navigation :***

Cet agent est mis en place pour détecter la région où se trouve le robot, puis il va aider à optimiser les ressources par donner la région où le robot se trouve pour permettre une représentation partielle de l'environnement (représentation de la région).

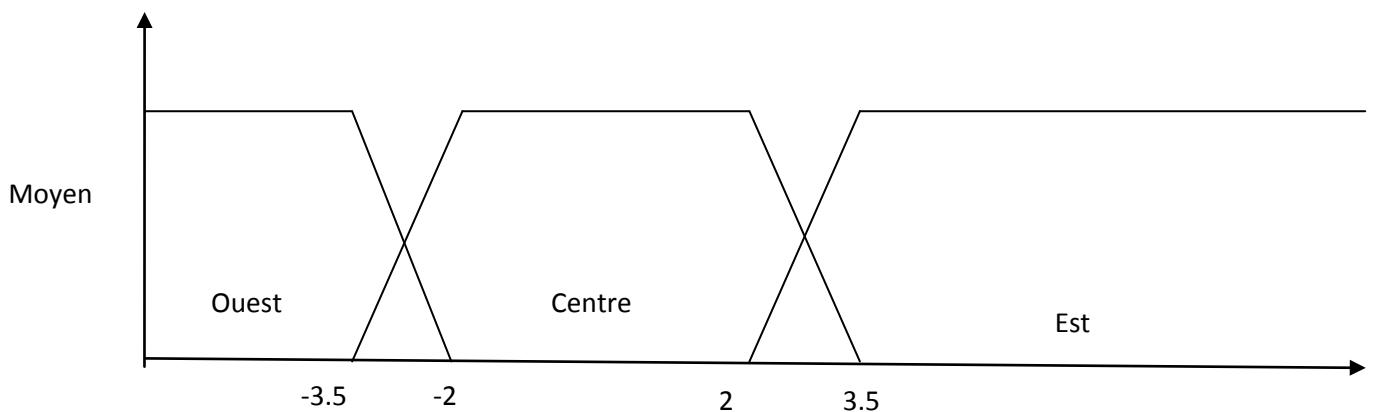
## Conception et implémentation

Il doit être activé à chaque moment de la navigation, les valeurs d'entrées changent à chaque moment  $t$ , ce sont les coordonnées  $(X, Y)$  de la position de robot au moment  $t$ , la sortie qui vient de cet agent est la région où se trouve le robot au moment  $t$ , et selon cette sortie, la représentation partielle des différentes régions sera activée. Pour cela, un classificateur flou est mis en place.

Les différentes régions à être classé en Cinq classes sud-est, sud-ouest, nord-est, nord-ouest et centre. Dans ce cas on a deux variables floues  $x$  et  $y$ , les sous ensembles flous sont différents selon la taille de l'environnement.

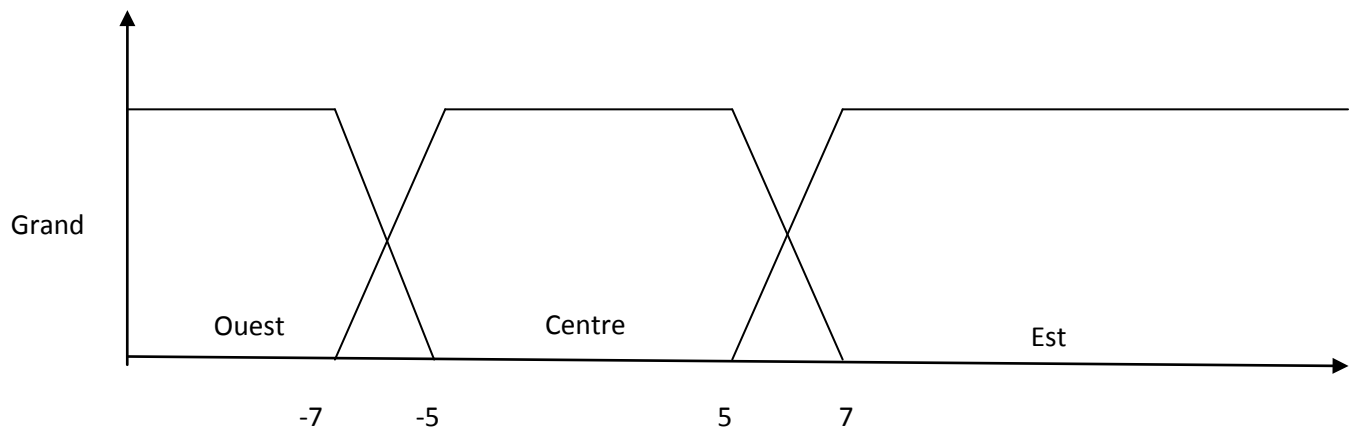


**Figure V.3:** Sous ensembles flous pour  $x$  dans le cas de petit environnement

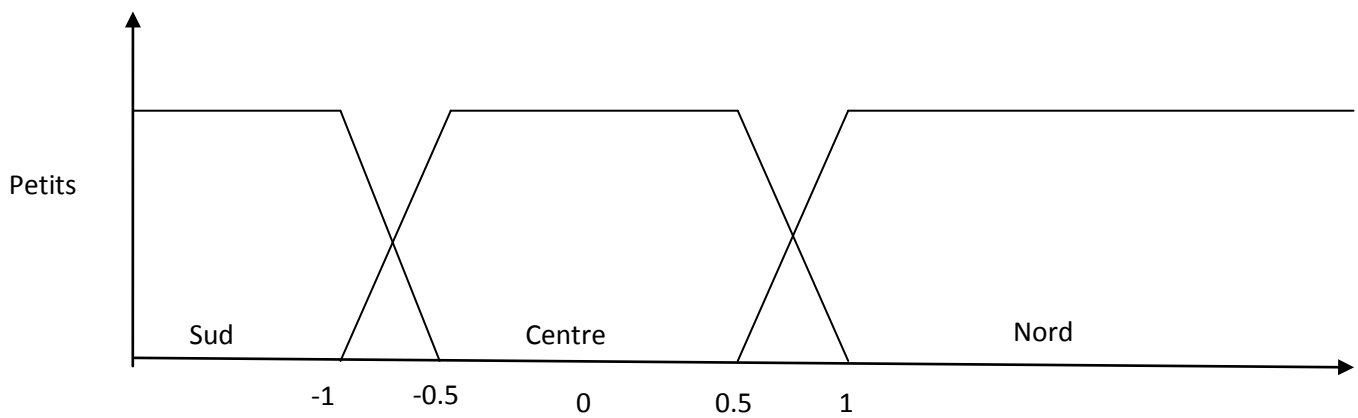


**Figure V.4:** Sous ensembles flous pour  $x$  dans le cas ou l'environnement moyen

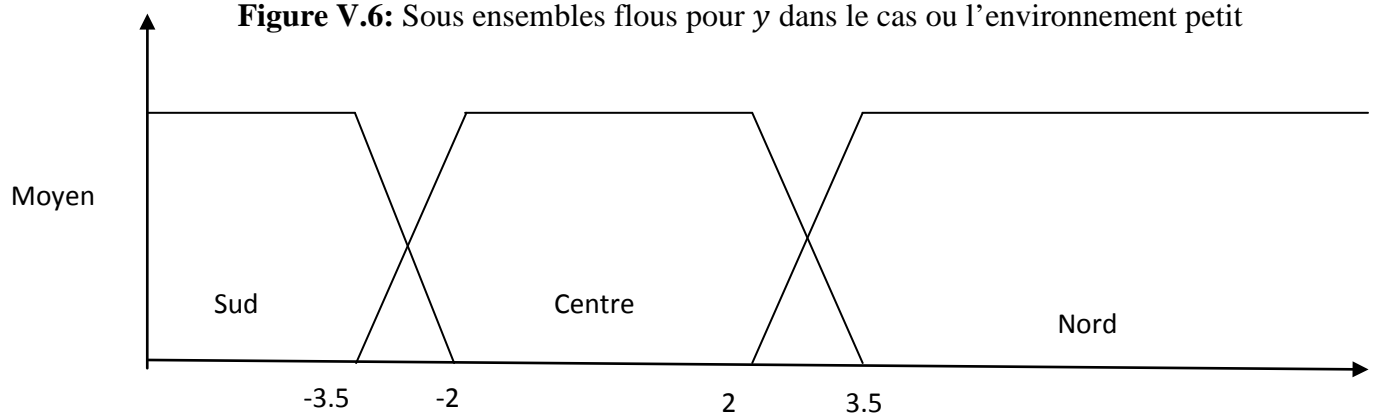
## Conception et implémentation



**Figure V.5:** Sous ensembles flous pour  $x$  dans le cas ou l'environnement grand



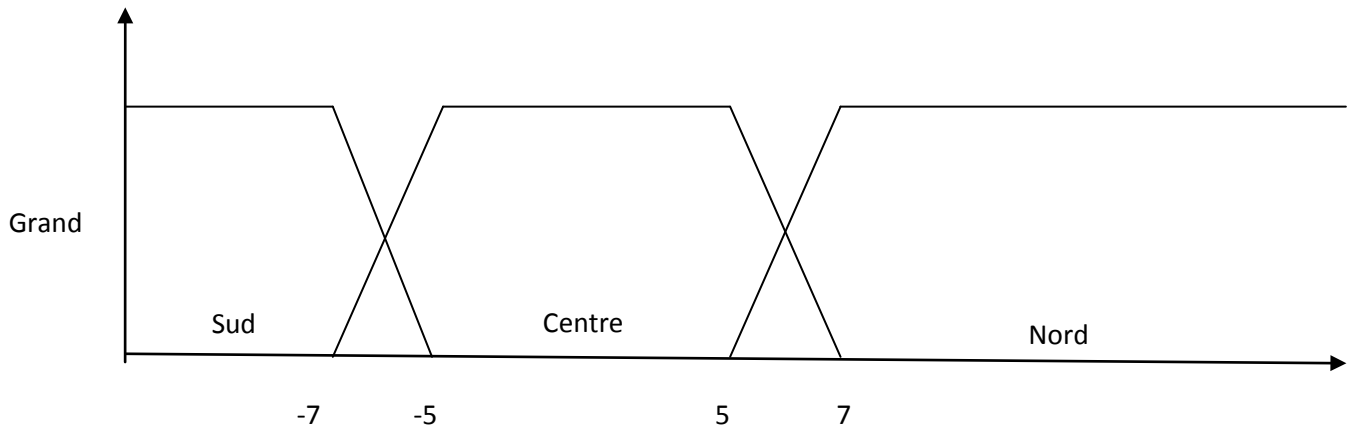
**Figure V.6:** Sous ensembles flous pour  $y$  dans le cas ou l'environnement petit



**Figure V.7:** Sous ensembles flous pour  $y$  dans le cas ou l'environnement moyen

## Conception et implémentation

---



**Figure V.8:** Sous ensembles flous pour  $y$  dans le cas où l'environnement grand

Les règles d'inférence pour détecter la région de navigation sont données par le tableau suivant :

## Conception et implémentation

---

x	y	Sortie
<-1	<0	Sud-ouest
<-1	>0	Nord-ouest
[-1, -0.5]	<-1	Sud-ouest
[-1, -0.5]	[-1, -0.5]	Calculer le degré d'appartenance
[-1, -0.5]	[-0.5, 0.5]	Centre
[-1, -0.5]	[0.5, 1]	Calculer le degré d'appartenance
[-1, -0.5]	>1	Nord-ouest
[-0.5, 0]	<-1	Sud-ouest
[-0.5, 0]	[-1,1]	Centre
[-0.5, 0]	>1	Nord-ouest
[0, 0.5]	<-1	Sud-est
[0, 0.5]	[-1,1]	Centre
[0, 0.5]	>1	Nord-est
[0.5, 1]	<-1	Sud-est
[0.5, 1]	[-1, -0.5]	Calculer le degré d'appartenance
[0.5, 1]	[-0.5, 0.5]	Centre
[0.5, 1]	[0.5, 1]	Calculer le degré d'appartenance
[0.5, 1]	>1	Nord-est
>1	<0	Sud-est
>1	>0	Nord-est

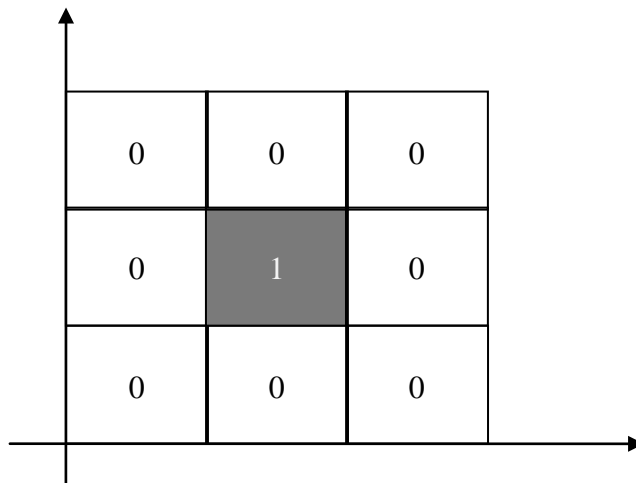
**Tableau V.1:** règles floues pour la région de navigation (petite environnement)

Remarque: Les mêmes règles s'appliquent pour le cas d'environnement de taille moyenne et de grande taille, avec le changement des valeurs de x et de y pour qu'ils conviennent à la taille de l'environnement

### ***V.3.3 La représentation de l'environnement de la navigation :***

L'algorithme est conçu pour donner une représentation très simple de l'environnement du travail, premièrement il est découpé en grille de 40×40 (1600) puis on associe une valeur à chaque petit carré, 0 si le carré ne contient pas d'obstacles et 1 s'il contient d'obstacles.

Exemple :



**Figure V.9:** Exemple de représentation d'un environnement

Le vecteur qui représente cette grille ici est : 0, 0, 0, 0, 1, 0, 0, 0,0.

Le calcul de x et y se fait de la position dans le vecteur par les formules suivantes:

$x = 0.5 ;$

`pour(int i =0; i<position dans le vecteur;i++)`

`{ si (x == 20)`

`{ x = 0.5; }`

```
        sinon {x = x+0.5;}  
    }  
    x = x-0.5;  
y = 0.5;  
pour(int i =0; i<position dans le vecteur;i++)  
    { pour(int j =0; j<40;j++)  
        { i++;}  
        y= y+0.5;  
        if(y>20)  
            { y = 0; }  
    }  
}
```

Exemple :

### ***V.3.4 La navigation :***

Lorsque l'environnement du travail du robot est bien présenté, on met en service un algorithme qui aide à trouver un chemin de navigation, le principe du fonctionnement est très simple, ça sert à calculer les valeurs abscisse de départ moins abscisse d'arrivée et ordonnées de départ moins ordonnées d'arrivée. Selon les signes des résultats une orientation est choisie, le robot ne change pas sa position jusqu'à ce qu'il vérifie que la direction choisie ne contient pas un obstacle. Si cette direction est occupée par un obstacle, une autre direction est sélectionnée on utilise un mécanisme précis qui permet de se rapprocher du but et d'éviter l'obstacle. Le tableau suivant présente l'ensemble de tous les cas possibles et l'orientation choisie avec chaque cas possible, les nombres indiquent la priorité du choix :

## Conception et implémentation

---

Action		x=x+0.5		x=x+0.5		x=x		x=x		x=x-0.5		x=x+0.5		x=x-0.5		x=x-0.5	
		y= y+0.5		y= y		y= y+0.5		y= y-0.5		y= y		y= y-0.5		y=y+0.5		y= y-0.5	
X <sub>dep</sub> -X <sub>arr</sub>	Y <sub>dep</sub> -Y <sub>arr</sub>	Obs +	Obs -	Obs +	Obs -	Obs +	Obs -	Obs +	Obs -	Obs +	Obs -	Obs +	Obs -	Obs +	Obs -	Obs +	Obs -
-	-	0	1	0	2	0	2	0	4	0	4	0	3	0	3	0	5
0	-	0	2	0	3	0	1	0	4	0	3	0	4	0	2	0	4
-	0	0	2	0	1	0	3	0	3	0	4	0	2	0	4	0	4
0	0	Objectif atteint: arrêter la navigation															
0	+	0	4	0	3	0	4	0	1	0	3	0	2	0	4	0	2
+	0	0	2	0	1	0	3	0	3	0	4	0	2	0	4	0	4
+	+	0	4	0	4	0	4	0	2	0	2	0	3	0	3	0	1
+	-	0	3	0	4	0	2	0	4	0	2	0	4	0	1	0	3
-	+	0	3	0	2	0	4	0	2	0	4	0	1	0	4	0	3

**Tableau V.2:** Ensemble des cas possibles avec l'action correspondant.

Avec :X<sub>dep</sub> : abscisse de départ.

X<sub>arr</sub> : abscisse d'arriver.

Obs - : pas d'obstacle.

Obs + : présence d'obstacle.

0 : ne pas naviguer dans cette région

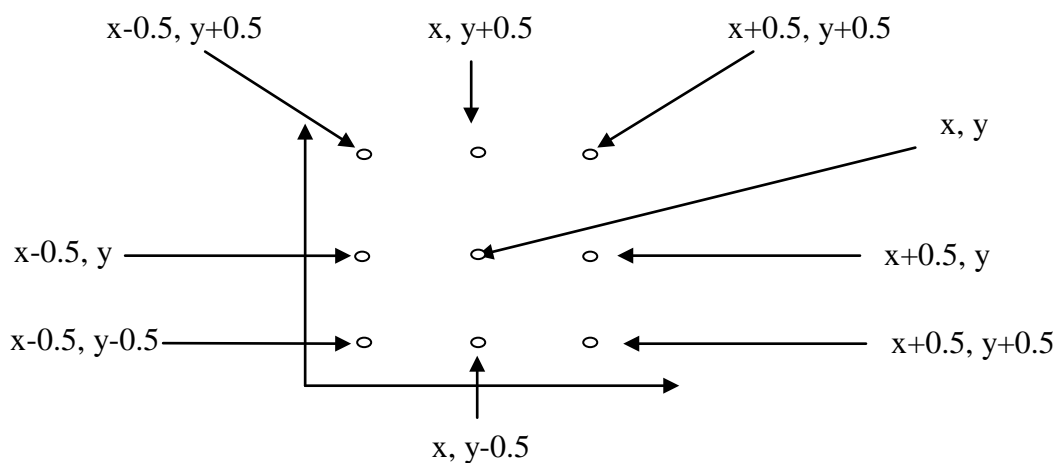


1 : choix numéro 1. (Première priorité).

2 : choix numéro 2. (Deuxième priorité).

3 : choix numéro 3. (Troisième priorité).

4 : choix numéro 4. (quatrième priorité).



**Figure V.10:** Voisinage de la position courante (x, y)

Remarques : dans certains cas où le robot existe dans les bornes sud, nord, est ou ouest les voisins sont restreints.

Le tableau présenté ici peut être considéré comme étant un tableau d'une fonction logique (par remplacement de toutes les valeurs sauf une par des zéros), d'où il est possible de créer un réseau de neurones basé sur des neurones ET, et des neurones OU, d'où avoir en sortie le choix de navigation.

## V.4 Simulation et résultats :

### V.4.1 La reconnaissance de la nature de l'environnement :

Dans notre travail, on a fait présenter 25 cas de différents environnements de taille différente, les résultats ont montré que l'agent a une capacité élevée de reconnaissance et de classification des différents cas, à titre d'exemple :

## Conception et implémentation

---

(-3.75,-5), (3.75, 5)

“This environment with diagonal = 12.5 meter is: small”

Un autre exemple:

(-18.75,-25), (18.75, 25)

This environment with diagonal = 62.5 meter is: small at 40.0 percent and medium at 60.0 percent

Dans le cas de ce dernier il est considéré plus proche à un environnement de taille moyenne, ce qui va le classer avec les environnements de taille moyenne.

### ***V.4.1 La reconnaissance de la région de navigation :***

Pour tester l'efficacité de cet agent il a comme entrées un ensemble de 1200 points aléatoires alors il a donné des résultats comme suite :

-67.0 -20.0 sud-ouest

22.0 -41.0 sud-est

-63.0 99.0 nord-ouest

69.0 38.0 nord-est

A partir de ces résultats, on fait remarquer que l'agent a fait classer les entrées dans leurs classes correspondantes.

### ***V.4.2 le calcul de la position :***

Pour calculer x et y de la position 326 dans le vecteur d'entrée on écrit :

x= 0.5;

pour (int i =0; i<326;i++)

{ si (x == 20)

{ x = 0.5;}

```
        sinon {x = +0.5;}  
    }  
  
    x = x-0.5;  
  
y = 0.5;  
  
pour (int i =0; i<326;i++)  
  
    {pour (int j =0; j<40;j++)  
  
        {i++;}  
  
        y= y+0.5;  
  
        if (y>20)  
  
            {y = 0;}  
  
    }  
}
```

Les sorties seront :

$$x = 3.0$$

$$y = 4.5$$

Remarque :

Il est aussi possible de calculer une position dans le vecteur partant de x et y correspondant.

Une transformation ponctuelle est effectuée pour faciliter les calculs des x et des y, l'origine de repère sera dirigée du milieu de l'environnement vers le point de l'abscisse et l'ordonnée les plus petites, de cette manière tous les points auront des abscisses et des ordonnées positives.

### ***V.4.3 la navigation :***

Grace a un telle réseau de neurones en peut obtenir des résultats optimale, de plus il est possible d'extraire les connaissances a partir de ce réseau et justifier la décision pris.

### **V.5 Conclusion :**

Toutes les solutions intelligentes ont des avantages et des inconvénients, la combinaison de plusieurs méthodes nous permet non seulement de surpasser les inconvénients d'une méthode ou d'une autre, mais aussi d'implémenter un savoir faire de haute niveau pour un robot (dans le cas de robots), dans le cas général les solutions basées sur cette approche permet la résolution des problèmes plus complexes, il suffit de conduire une analyse exacte de la problématique.

### **Conclusion générale:**

La robotique mobile et grâce à leurs domaines d'applications très variés, elle occupe un champ de recherche prédominant.

Dans le cas des robots mobiles, le concept de planification de la trajectoire et d'évitement d'obstacles est très important, parce qu'il contribue largement à réussir la mission pour laquelle le robot mobile a été conçu, c'est la raison pour laquelle on trouve beaucoup de contributions et de recherches qui proposent des solutions différentes à cette problématique.

Les différentes propositions sont basées sur des méthodes issues de l'intelligence artificielle, et peuvent être classées en tant que méthodes classiques ou heuristiques.

Les méthodes classiques se caractérisent par la simplicité et la facilité d'implémentation, mais elles présentent aussi beaucoup de limitations car elles sont simples et ne cherchent pas d'optimiser ou à faire face à des situations de complexité élevée; par contre les méthodes heuristiques présentent une capacité élevée dans la résolution de ces complications, le problème avec ces méthodes réside dans le débat philosophique actuel entre les différents défenseurs d'approches et aussi dans leur nature heuristique.

Pour aller au-delà et trouver une nouvelle façon de la résolution de cette problématique, on a essayé de combiner plusieurs approches pour trouver la solution, cela permet de bénéficier de la simplicité des méthodes classiques et de l'intelligence des méthodes heuristiques, donc exploiter les avantages des deux approches et éviter le conflit existant entre les défenseurs des différentes méthodes.



**Annexe**