

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS SETIF-1

FACULTE DE TECHNOLOGIE

THESE

Présentée au Département d'Electronique

Pour l'obtention du diplôme de

DOCTORAT EN SCIENCES

Filière : Electronique

Option : Electronique

Par

KASMI BOUCETTA

THEME

**COMMANDE NEURONALE ROBUSTE
POUR LA NAVIGATION D'UN ROBOT MOBILE**

Soutenue le 01/06/2022 devant le Jury :

BOUKEZZOULA Naceur-Eddine	Professeur	Univ. F. Abbas Sétif 1	Président
HASSAM Abdelouahab	Professeur	Univ. F. Abbas Sétif 1	Directeur de thèse
HOCINI Abdeslam	Professeur	Univ. M. Boudiaf Msila	Examineur
KHEDROUCHE Djamel	Professeur	Univ. M. Boudiaf Msila	Examineur

Dédicaces

Je dédie cette thèse

À ma mère et ma femme et mes enfants,

À mes frères et sœurs,

Remerciements

Je tiens à remercier tout d'abord et avant tout ALLAH, le GRAND, le tout puissant qui m'a donné la sante, la volonté et la patience pour pouvoir terminer ce travail.

*Ensuite, je tiens à adresser mes plus vifs remerciements à mon directeur de thèse, Mr. **HASSAM. A**, Professeur à l'université de Ferhat Abbas Sétif 1 pour m'avoir encadré, suivi et encouragé, et pour ses conseils appréciables et ses critiques constructives.*

*Je suis profondément reconnaissant à Monsieur **BOUKEZZOULA Naceur-Eddine** (Professeur à l'Université de Ferhat Abbas Sétif.1) pour m'avoir fait l'honneur de présider le jury de soutenance.*

*Mes remerciements les plus sincères vont également à l'égard de Monsieur **HOCINI. A** (Professeur à l'Université de M. Boudiaf Msila), Monsieur **KHEDROUCHE. D** (Professeur à l'Université de M. Boudiaf Msila) pour l'intérêt qu'ils ont bien voulu porter à ce travail en acceptant de l'examiner.*

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

Kasmí boucetta

Symboles et Abréviations

SIF	Système d'Inférence Flou.
RNA	Réseaux de Neurones Artificiels.
ANFIS	Adaptive Network based Fuzzy Inference System.
θ	L'orientation du robot.
l	La longueur du robot.
R	La longueur du châssis.
V	La vitesse linéaire de translation du robot.
$\Delta\theta/\alpha$	L'angle de braquage du robot.
x_r, y_r	Coordonnées du robot mobile.
$\mu_{A_x}(x)$	Degré d'appartenance de la variable x .
$d_{rt/cbl}$	Distance entre le robot et le but.
$\theta_{rt/ob}$	Angle entre l'orientation actuelle du robot et celle de l'obstacle.
w_r, w_l	Les vitesses de rotation des roues droite et gauche.

Table des matières

Liste des figures.....	I
Liste des tableaux	III

Introduction Générale.....	1
----------------------------	---

Chapitre I : Navigation d'un Robot Mobile

I.1 Introduction.....	5
I.2 Définition d'un robot mobile	6
I.3 Autonomie d'un robot mobile.....	7
I.4 Les différents types de terrains	8
I.5. Modélisation cinématique d'un robot mobile à roues	9
I.5.1 Introduction	9
I.5.2 Repérage d'un robot mobile	9
I.5.3. Roulement sans glissement.....	10
I.5.4 Classification des types des roues	12
I.5.5 Types des plateformes mobiles	13
I.5.5.1 Plateformes non holonomes	14
I.5.5.2 Plateformes holonomes	15
I.6 Les capteurs comme sources d'informations.....	16
I.6.1 Capteurs intéroceptifs	16
I.6.2 Capteurs extéroceptifs	16
I.7 Navigation Autonome d'un robot mobile.....	17
I.8 Les thématiques scientifiques de la robotique mobile	18
I.8.1 Techniques de localisation et cartographie.....	18
I.8.2 Contrôle et commande des véhicules.....	19
I.8.3 La communication inter-véhicules.....	19
I.8.4 L'interprétation de scènes.....	19
I.9 Planification de trajectoire	19
I.9.1 Planification globale de trajectoire	20
I.9.2 Planification locale de trajectoire	20

I.9.2.1. Méthode par décomposition de l'environnement en cellule.....	20
I.9.2.2. La méthode des champs de potentiel	21
I.9.2.3. La méthode de la fenêtre dynamique	22
I.9.2.4. La méthode de la bande élastique	22
I.9.2.5. La planification par logique floue	22
I.9.2.6. La planification par juxtaposition de splines polynomiales.....	23
I.9.2.7. DKP (Deterministic Kinodynamic Planning)	23
I.9.2.8. La planification par Réseaux de neurones (RNA)	24
I.10 Conclusion.....	25
Références Bibliographiques	

Chapitre II : Les Systèmes d'Inférence Flous

II.1 Introduction	27
II.2 Concepts fondamentaux flous	28
II.3 Définition d'un ensemble flou	29
II.4 Fonction d'appartenance d'un ensemble flou (FA).....	29
II.4.1 Fonction d'appartenance triangulaire.....	29
II.4.2 Fonction trapézoïdale	30
II.4.3 Fonction gaussienne	30
II.4.4 Fonction sigmoïde	31
II.5 Variables floues et linguistiques.....	31
II.6 Opérations sur les ensembles flous.....	32
II.6.1 L'égalité	32
II.6.2 L'inclusion.....	32
II.6.3 L'intersection	32
II.6.4 L'union	33
II.6.5 La Complémentation	33
II.7 Commande et réglage par la logique floue.....	33
II.8 Structure des systèmes à base de la logique floue.....	34
II.8.1 Interface de fuzzification.....	35
II.8.2 La base de règles floues	35
II.8.3 La logique de prise de décision.....	35

II.8.4 Interface de défuzzification.....	35
II.9 Représentation mathématique des systèmes flous.....	36
II.9.1 Fuzzification.....	36
II.9.2 Base de règles floues.....	37
II.9.3 Moteur d'inférence floue.....	37
II.9.3.1 Méthode de Mamdani.....	37
II.9.3.2 Méthode de Takagi-Sugeno.....	39
II.9.3.3 Méthode de Tsukumoto.....	39
II.10 Approximation des fonctions par les systèmes flous.....	40
II.11 Conclusion.....	41
Références Bibliographiques	

Chapitre III : Les Réseaux de Neurones Artificiels et les Systèmes Hybrides Neuro-Flou

III.1 Introduction.....	43
III.2 Définitions.....	45
III.2.1 Intelligence Artificielle (IA).....	45
III.2.2 Agent.....	45
III.2.3 Apprentissage.....	45
III.3 Réseaux de neurones artificiels (RNA).....	47
III.3.1 Structure d'un neurone.....	47
III.3.2 Fonctions d'activation.....	49
III.3.3 Structures d'interconnexion.....	50
III.3.3.1 Réseau multicouche (Multi Layer Perceptron).....	50
III.3.3.2 Réseau à connexions récurrentes.....	51
III.3.3.3 Réseau à connexions locales.....	51
III.3.3.4 Réseaux à fonction radiale.....	51
III.3.3.5 Réseau à connexion complète.....	51
III.3.4 Apprentissage des RNA.....	52
III.3.4.1 Apprentissage supervisé.....	52
III.3.4.2 Apprentissage non supervisé.....	53
III.4 Les RNA pour la commande des processus.....	53

III.4.1	Identification par <i>RNA</i>	53
III.4.1.1	Modélisation directe	53
III.4.1.2	Modélisation inverse	54
III.4.2	Commande par RNA	54
III.5	L'approche neuro-floue	56
III.5.1	Introduction.....	56
III.5.2	Motivations pour une approche hybride	56
III.5.3	Systèmes hybrides neuro-flous	57
III.5.3.1	Le modèle <i>FALCON</i> et le modèle <i>GARIC</i>	59
III.5.3.2	Le modèle <i>NEFCLASS</i>	59
III.5.3.3	Le modèle <i>ANFIS</i>	59
III.5.4	Système d'inférence flou basé sur les réseaux de neurones adaptatifs ANFIS...	59
III.5.5	Vue d'ensemble d'ANFIS.....	60
III.5.6	Apprentissage de l'ANFIS.....	63
III.6	Conclusion.....	64

Références Bibliographiques

Chapitre IV : Navigation Autonome par des Contrôleurs Flous

IV.1	introduction	66
IV.2	Description de l'architecture de navigation autonome proposée	66
IV.3	Modèle du robot mobile.....	67
IV.4	Présentation de l'odométrie	69
IV.4.1	Application de l'odométrie pour la localisation d'un robot mobile	69
IV.5	Navigation basée sur les comportements	70
IV.6	Navigation basée sur les comportements flous	72
IV.7	Conception des comportements flous	73
IV.7.1	Comportement de navigation vers un but (recherche d'une cible)	74
IV.7.2	Comportement d'évitement d'obstacles	76
IV.8	Résultats de simulation	79
IV.8.1	Comportement de convergence vers un but	79
IV.8.2	Comportement d'évitement d'obstacles	81

IV.9 Conclusion.....	83
----------------------	----

Références Bibliographiques

Chapitre V : Navigation Autonome par des Contrôleurs Neuro-Flous

V.1 Introduction	85
V.2 La tâche de poursuite d'une trajectoire	87
V.3 Conception des contrôleurs de poursuite d'une trajectoire	88
V.3.1 Conception du Contrôleur flou	88
V.3.1.1 Résultats de poursuite de trajectoire.....	90
V.3.2 Systèmes Neuro-Flous	92
V.3.2.1 Conception du contrôleur Neuro-Flou.....	93
V.3.2.2 Résultats de poursuite de trajectoire.....	93
V.4 Poursuite d'une cible mobile	95
V.4.1 Contrôleur flou.....	95
V.4.2 Contrôleur hybride neuro-flou ANFIS.....	97
V.4.3 Comparaison	98
V.5 Conclusion.....	100

Références Bibliographiques

Conclusion Générale.....	101
---------------------------------	------------

Liste des figures

Figure I.1	Modélisation du robot dans le repère absolu.....	10
Figure I.2	Caractérisation du roulement sans glissement.....	11
Figure I.3	Roue centrée et roue décentrée.....	12
Figure I.4	Roue troncosphérique et roue suédoise.....	13
Figure I.5	Equivalence modèle voiture et tricycle.....	14
Figure I.6	Modèle de type tricycle à roues différentielles.....	15
Figure I.7	Le robot omnidirectionnel.....	16
Figure I.8	Processus de navigation autonome d'un robot mobile.....	18
Figure I.9	Modélisation d'un réseau de neurones artificiel.....	24
Figure II.1	Fonction triangulaire.....	30
Figure II.2	Fonction trapézoïdale.....	30
Figure II.3	Fonction gaussienne.....	30
Figure II.4	Fonction sigmoïde.....	31
Figure II.5	Représentation de la variable linguistique (distance).....	32
Figure II.6	Structure de base d'un système flou.....	35
Figure II.7	Méthodes de fuzzification : (a) singleton, (b) ensemble flou.....	36
Figure II.8	Différents modèles d'inférences floues.....	40
Figure III.1	Représentation de l'apprentissage automatique	46
Figure III.2	Représentation de l'apprentissage supervisé.....	46
Figure III.3	Représentation de l'apprentissage par renforcement.....	47
Figure III.4	Représentation d'un neurone biologique.....	48
Figure III.5	Modèle d'un neurone artificiel.....	48
Figure III.6	Formes usuelles de la fonction d'activation.....	50
Figure III.7	Les différentes structures d'interconnexion des RNA.....	52
Figure III.8	Principe du système neuro-flou.....	58
Figure III.9	Architecture équivalente d'ANFIS pour deux règles.....	61

Figure IV.1	Configuration du robot mobile utilisé.....	68
Figure IV.2	L'odométrie appliquée à la plateforme mobile.....	70
Figure IV.3	Architecture basée sur les comportements.....	71
Figure IV.4	Structure du système de navigation proposé.....	72
Figure IV.5	Les fonctions d'appartenances des variables d'entrées d et θ	75
Figure IV.6	Les fonctions d'appartenances des variables de sorties v et $\Delta\theta$	76
Figure IV.7	Les fonctions d'appartenances des variables d'entrées d_{ob} et φ	78
Figure IV.8	Navigation libre $x_1=2.5, y_1= 4.5, \theta=\pi$ vers $x_f=7, y_f=-1$	79
Figure IV.9	Navigation libre $x_1=8, y_1= 7, \theta=\pi/2$ vers $x_f=2, y_f=1$	80
Figure IV.10	Navigation libre à partir de trois différentes configurations initiales vers un but en utilisant un contrôleur flou MAMDANI (max min).....	80
Figure IV.11	Navigation libre avec la cible mobile.....	81
Figure IV.12	Navigation en présence d'obstacles. $x_1=2, y_1= 1, \theta=0$, vers $x_f=5, y_f=8$...	82
Figure IV.13	Navigation en présence d'obstacles. $x_1=6, y_1= 1, \theta=0$, vers $x_f=2, y_f=9$...	82
Figure IV.14	Navigation en présence d'obstacles. $x_1=8, y_1= 6.5, \theta=\pi/2$, vers $x_f=2, y_f=0.5$	83
Figure IV.15	Navigation en présence d'obstacles. $x_1=5, y_1= 7, \theta=-\pi/2$, vers $x_f=2, y_f=0.5$	83
Figure V.1	Principe de poursuite.....	88
Figure.V.2	Contrôleur flou de poursuite de trajectoire.....	89
Figure V.3	Fonctions d'appartenances de $\theta_{rb/cbl}$	89
Figure V.4	Fonctions d'appartenances de $d\theta_{rb/cbl}$	89
Figure V.5	Fonctions d'appartenances de α	89
Figure V.6	Poursuite de trajectoire pour le contrôleur flou les points initiaux (4,4)...	90
Figure V.7	Poursuite de trajectoire et l'angle de braquage pour le contrôleur flou.....	91
Figure V.8	Poursuite de trajectoire et l'angle de braquage pour le contrôleur flou.....	91
Figure V.9	Poursuite de trajectoire et l'angle de braquage pour le contrôleur flou.....	91
Figure V.10	Poursuite de trajectoire pour le contrôleur flou les points initiaux (6,6).....	92
Figure V.11	Contrôleur neuro-flou de poursuite de trajectoire.....	93
Figure V.12	Poursuite de trajectoire pour le contrôleur neuro-flou, le point initial (4,4)94	

Figure V.13	Poursuite de trajectoire à partir de différentes configurations initiales.....	94
Figure V.14	Poursuite de trajectoire pour le contrôleur neuro-flou.....	95
Figure V.15	Poursuite de trajectoire en présence d'obstacles pour le contrôleur flou, les points initiaux (2,9).....	96
Figure V.16	Poursuite d'une cible mobile en présence d'obstacles pour le contrôleur flou, les points initiaux (6,6).....	96
Figure V.17	Poursuite d'une cible mobile en présence d'obstacles pour le contrôleur flou, les points initiaux (6,6).....	97
Figure V.18	Poursuite d'une cible mobile en présence d'obstacles pour le contrôleur neuro-flou.....	97
Figure V.19	Poursuite d'une cible mobile en présence d'obstacles pour le contrôleur neuro-flou, le point initial (8,2).....	98
Figure V.20	Poursuite d'une cible mobile en présence d'obstacles en utilisant : (a) contrôleur flou, (b) contrôleur neuro-flou, Le point initial (8,2).....	99
Figure V.21	Poursuite d'une cible mobile en présence d'obstacles en utilisant : (a) contrôleur flou, (b) contrôleur neuro-flou, le point initial (2,2).....	99
Figure V.22	Poursuite d'une cible mobile en présence d'obstacles en utilisant : (a) contrôleur flou, (b) contrôleur neuro-flou, le point initial (6,4).....	100

Liste des Tableaux

Tableau III.1	Comparaison entre la logique floue et les réseaux de neurones.....	57
Tableau III.2	Méthodes utilisées pour l'apprentissage de l'ANFIS.....	64
Tableau IV.1	Les règles floues pour le comportement convergence vers un but.....	76
Tableau IV.2	Base de règles pour le comportement d'évitement d'obstacles.....	78
Tableau V.1	Règles floues de poursuite de trajectoire.....	90

Introduction Générale

Depuis quelques années, un intérêt croissant est porté, au sein de la communauté robotique, au développement des systèmes intelligents autonomes dans le cadre de la robotique mobile [1]. Un tel intérêt peut être perçu comme une conséquence logique à l'apparition des applications potentiels des machines intelligentes (industriels, services, manutention ou encore de l'aide à la mobilité des personnes âgées ou aux personnes en situation de handicapées,...). L'objectif est de mettre les robots dans des tâches variées telles que les interventions sur des sites accidentés, la manipulation sur sites sensibles ou nucléaires, ainsi pour l'exploration des sites maritimes et planétaires...etc.

L'enjeu principal de la robotique mobile actuelle consiste à développer des systèmes de navigation intelligents, où la navigation autonome est un axe de recherche qui vise à donner à une machine la capacité de se mouvoir dans un environnement sans assistance, ni intervention humaine pour accomplir un but désiré [2-3]. La tâche de navigation consiste à donner au robot la possibilité d'obtenir les informations dont il a besoin pour raisonner et le doter de capacité de locomotion adaptée à son environnement [4]. Cependant, celle-ci implique des systèmes complexes dans la réalisation, où leurs maîtrises posent d'importants problèmes non seulement technologiques mais aussi scientifiques.

Un robot mobile autonome est un système mécanique qui doit être en mesure de prendre des décisions pour effectuer des mouvements en fonction des informations sur sa position et sur l'environnement qu'il contourne. L'intérêt est d'atteindre un but désiré, tout en s'adaptant à certaines variations des conditions de fonctionnement sans intervention humaine [5].

De manière générale, on regroupe sous l'appellation "*robots mobiles*" l'ensemble des robots à base mobile [6]. Les autres robots sont distingués par le type de locomotion qu'ils soient : marcheurs, sous-marins ou aériens [7].

La recherche en robotique mobile s'intéresse à la conception des systèmes intelligents dotés par des techniques de commande efficaces pour le déplacement d'un robot mobile, où la sécurité soit prioritaire par rapport à l'optimalité [1], [5]. Au début, le système de navigation est disposé d'un modèle de l'environnement dans lequel sont représentés les principaux éléments fixes : obstacles, murs, portes, meubles...etc. Ces systèmes de navigation traditionnels présentent certaines difficultés dans les applications temps réel puisqu'ils nécessitent une grande capacité de calcul et de mémoire, exigent une planification complète de l'environnement et restent incompatibles avec les exigences des robots en temps réel en termes de rapidité et de réactivité [8-9]. Cependant, lorsque l'environnement devient plus complexe (partiellement connu ou dynamique), il apparaît indispensable que le robot mobile soit doté de capacités décisionnelles apte à le faire réagir automatiquement sans collision avec les objets imprévus, percevoir, analyser et modéliser son environnement. Ensuite, à partir de l'information disponible, le robot doit planifier sa trajectoire de mouvement et le système de contrôle doit exécuter la séquence des actions élaborées. D'autre part, les stratégies de commande réactives offrent des solutions intéressantes qui utilisent directement l'information issue des capteurs du robot pour atteindre le but avec évitement des obstacles si l'environnement est inconnu ou dynamique, en se basant sur des techniques intelligentes [10-16].

Plusieurs approches ont été proposées pour résoudre le problème de navigation autonome d'un robot mobile, les plus utilisées sont celles de l'intelligence artificielle : la logique floue, les réseaux de neurones artificiels, l'apprentissage automatique et les systèmes hybrides. Dans l'approche floue, le comportement du robot est décrit par des règles linguistiques de type *Si-Alors* [12] [15], [17-21]. Ces règles sont obtenues généralement à partir des connaissances d'un expert humain. Cependant, lorsque l'environnement est complexe et dynamique, il est difficile de construire une base de règles convenables parce que le nombre de situations à considérer est très élevé. D'autre part, le thème de l'apprentissage et l'adaptation est devenu très important dans les dernières années. Les techniques les plus utilisées sont celles qui font appel à des techniques d'auto-apprentissage [22] et les systèmes neuronaux [23]. Les capacités d'apprentissage des réseaux de neurones peuvent être utilisées pour résoudre ce

problème. L'idée de base est d'entraîner le robot mobile pour un ensemble de situations et la capacité de généralisation des réseaux de neurones lui permet de prendre les décisions adéquates dans le cas de nouvelles situations [24-26]. La commande floue a montré son efficacité pour la navigation des robots mobiles [17], [20][15], mais la construction d'un contrôleur flou performant pour un mouvement souhaité n'est pas toujours facile [27][28]. L'inconvénient majeur est le manque d'une méthodologie systématique pour la conception, due au nombre important de paramètres à régler (les paramètres des fonctions d'appartenances, les paramètres de la partie conclusion et les règles d'inférences). On trouve dans la littérature plusieurs méthodes de réglage des contrôleurs flous par l'intégration des propriétés des systèmes flous avec d'autres approches de l'intelligence artificielle, telles que : les réseaux de neurones [29-30] l'apprentissage par renforcement [[31-33], les algorithmes génétiques [34], [14], colonie de fourmis [38], ou bien les faire combinées [35-37]...etc. Ces méthodes dites hybrides combinent les propriétés de chaque approche afin d'optimiser les paramètres des systèmes d'inférence flous. Elles sont capables de générer une solution optimale ou quasi-optimale.

L'objectif de ce travail est d'étudier et d'appliquer des techniques neuro-floues pour la navigation autonome d'un robot mobile dans un environnement inconnu, afin de permettre au robot de se mouvoir d'une position initiale à une autre finale. On utilise d'une part l'approche comportementale à base de la logique floue, et d'autre part le paradigme de l'apprentissage. Les techniques employées pour aborder ce problème sont basées sur les systèmes d'inférence flous, les réseaux de neurones artificiels.

Cette thèse est organisée de la manière suivante:

Le premier chapitre est consacré à la navigation autonome d'un robot mobile. Un aperçu général sur le domaine de la robotique mobile sera abordé pour examiner en bref les types des robots mobiles, les différentes parties constitutives et les architectures de contrôle existantes pour un robot mobile.

Dans le deuxième chapitre, on présente les systèmes d'inférence flous (SIF). Nous commençons par énoncer les fondements théoriques des sous-ensembles flous et de la

logique floue, suivi par la description de la structure générale d'un contrôleur flou, en montrant leurs types.

Dans le troisième chapitre, nous présentons une étude sur les réseaux de neurones artificiels (RNA) et les systèmes hybrides neuro-floues. La première partie est consacrée à une brève présentation des RNA, leur principe de fonctionnement, les types existants et leur apprentissage. Dans la deuxième partie de ce chapitre, nous présentons les fondements mathématiques des systèmes hybrides neuro-floues, les concepts de base, ensuite nous décrivons les principaux algorithmes.

Les deux derniers chapitres présentent nos contributions :

Dans le quatrième chapitre, nous décrivons des planificateurs locaux à base de la logique floue pour la navigation autonome d'un robot mobile. On a proposé des systèmes de commande réactifs basés sur les comportements, en décomposant la tâche globale en un ensemble des sous tâches secondaires : convergence vers un but et évitement d'obstacles.

Dans le cinquième chapitre, nous proposons des contrôleurs neuro-flou de type ANFIS. Ces systèmes de neuro-flous seront utilisés pour les différents comportements d'un robot mobile : poursuite d'une trajectoire de référence et de cible mobile, suivi de murs.

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

Nous terminerons ce manuscrit par une conclusion générale récapitulant ce qui a été fait et présente les perspectives de ce travail.

REFERENCES BIBLIOGRAPHIQUES INTRODUCTION GENERALE

- [1] F. Cuesta, A. Ollero, "Intelligent Mobile Robot Navigation", Springer-Verlag, Berlin Heidelberg, 2005.
- [2] J. C. Latombe, "Robot Motion Planning", Kluwer Academic Publishers, Norwell, 1991.
- [3] J.P. Laumond, "La Robotique Mobile", Editions Hermès, 2001.
- [4] J. Borenstein, H. R. Everett, & L. Feng, "Where am I, Sensors and Methods for Mobile Robot Positioning", University of Michigan, 1996.
- [5] S. G. Shuzhi, F. L. Lewis, "Autonomous Mobile Robots, Sensing, Control, Decision, Making and Applications", Taylor and Francis Group, 2006.
- [6] D. Filliat, "Robotique Mobile", Cours à l'école Nationale Supérieure des Techniques Avancées ENSTA, Octobre 2004.
- [7] B. Bayle, "Robotique Mobile", Ecole Nationale Supérieure de Physique de Strasbourg, Université Louis Pasteur, 2007.
- [8] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", International Journal of Robotics Research, vol. 5, no. 1, pp. 90-98, 1986.
- [9] J. Borenstein, Y. Koren, "The Vector Field Histogram Fast Obstacle Avoidance for Mobile Robots", IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pp.278-288, June 1991.
- [10] P. Reignier, "Pilotage Réactif d'un Robot Mobile, Etude de Lien entre la Perception et l'Action", Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1994.
- [11] H. R. Beom, H. S. Cho, "A Sensor-based Navigation for a Mobile Robot using Fuzzy Logic and Reinforcement Learning", IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, no.3, pp. 464-477, 1995.
- [12] A. Saffiotti, "The Uses of Fuzzy Logic for Autonomous Robot Navigation", Soft Computing, vol. 1, no. 4, pp.180-197, 1997.
- [13] H. Maaref, C. Barret, "Sensor Based Navigation of an Autonomous Mobile Robot in an Indoor Environment", Control Engineering Practice, vol. 8, pp. 757-768, 2000.
- [14] F. Abdessemed, K. Benmahammed & E. Monacelli, "A Fuzzy-based Reactive Controller for Non-holonomic Mobile Robot", Robotics and Autonomous Systems, vol. 47, pp. 31-46, 2004.
- [15] A. Fatmi, A. Al Yahmadi, L. Khriji, & N. Masmoudi, "A Fuzzy Logic Based Navigation of a Mobile Robot", World Academy of Science, Engineering and Technology, vol. 22, pp. 169-174, 2006.
- [16] O. Lefebvre, "Navigation Autonome sans Collision pour Robots Mobiles non holonomes", Thèse de Doctorat de l'Institut National Polytechnique de Toulouse, 2006.
- [17] H. Seraji, A. Howard, "Behavior-based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach", IEEE Transaction on Robotic and Automation, vol. 18, no. 3, pp. 308-321, 2002.

- [18] P. G. Zavalangas, S. G. Tzafestas, "Motion Control for Mobile Robot Obstacle Avoidance and Navigation: A Fuzzy Logic-Based Approach", *Systems Analysis Modeling Simulation*, vol. 43, no. 12, pp. 1625-1637, 2003.
- [19] X. Yang, al, "An Embedded Fuzzy Controller for a Behavior-Based Mobile Robot With Guaranteed Performance", *IEEE Transaction on Fuzzy Systems*, vol. 12, no. 4, pp. 436-446, 2004.
- [20] X. Yang, al, "A Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation", *IEEE Transaction on Systems, Man, and Cybernetics, B*, vol. 35, no.6, pp.1214-1224, 2005.
- [21] M. Wang, N. K. Liu James, "Fuzzy Logic based Real-time Robot Navigation in Unknown Environment with Dead Ends", *Robotics and Autonomous Systems*, vol. 56, pp.625-643, 2008.
- [22] R. S. Sutton, A. G. Barto, "Reinforcement Learning: An Introduction", MIT Press, Cambridge, MA, 1998.
- [23] J. A. Anderson, "An Introduction to Neural Networks", Bradford - MIT Press, 1995.
- [24] S. Thrun, "An Approach to Learning Mobile Robot Navigation", *Robotics and Autonomous Systems*, vol. 15, pp.301-319, 1995.
- [25] R. Fierro, F. L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks", *IEEE Transactions On Neural Networks*, vol. 9, no. 4, pp. 589-600, 1998.
- [26] E. Gauthier, "Utilisation des Réseaux de Neurones Artificiel pour la Commande d'un Véhicule Autonome", Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1999.
- [27] J. M. Mendel, "Tutorial: Fuzzy Logic Systems for Engineering", *Proceeding of the IEEE*, vol. 83, no. 3, pp. 345-377, March 1995.
- [28] K. M. Passino, S. Yurkovich, "Fuzzy Control", Addison Wesley Longman, 1998.
- [29] J.-S.R Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference System", *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 23, pp.665-685, 1993.
- [30] J. Godjevac, "Idées Nettes sur la Logique Floue", Collection Informatique, Presses Polytechniques et Universitaires Romandes, Lausanne, 1999.
- [31] H. R. Berenji, P. Khedkar, "Learning and Tuning Fuzzy Logic Controllers Through Reinforcements", *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 724-740, 1992.
- [32] P. Y. Glorennec, "Algorithmes d'Apprentissage pour Systèmes d'inférence Floue", Edition Hermès, 1999.
- [33] L. Jouffe, "Fuzzy Inference System Learning by Reinforcement Methods", *IEEE Transactions on Systems, Man, and Cybernetics*, vol.28, no.3, pp. 338-355, 1998.
- [34] F. Herrera, M. Lozano and J. L. Verdegay, "Tuning Fuzzy Controllers by Genetic Algorithms", *International Journal of Approximate Reasoning*, vol. 12, pp. 299-315, 1995.
- [35] C. Touzet, "Neural Reinforcement Learning for Behaviour Synthesis", *Robotics and Autonomous Systems*, vol. 22, no 3, pp. 251-281, 1997.

- [36] Y. Cang, N. H. C. Yung et D. Wang, "A Fuzzy Controller with Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 33, no.1, pp.1-11, 2003.
- [37] Y. Zhou, M. Joo Er et Y. Wen, "A Hybrid Approach for Automatic Generation of Fuzzy Inference Systems without Supervised Learning", Proceedings of the 2007 American Control Conference, USA, pp. 3371-3376, 2007.
- [38] H. Boubertakh, "Contribution à l'Optimisation par Algorithmes Evolutionnaires des Contrôleurs Flous", Thèse de Doctorat en Automatique, ENP, 2009.

Chapitre I

Navigation d'un Robot Mobile

I.1 Introduction

De manière générale, on regroupe sous l'appellation robots mobile l'ensemble des robots à base mobile, par opposition notamment aux robots manipulateurs. L'usage veut néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens.

On peut estimer que les robots mobiles à roues constituent le gros des robots mobiles. Historiquement, leur étude est venue assez tôt, suivant celle des robots manipulateurs, au milieu des années 70. Leur faible complexité en a fait de bons premiers sujets d'étude pour les roboticiens intéressés par les systèmes autonomes. Cependant, malgré leur simplicité apparente (mécanismes plans, à actionneurs linéaires), ces systèmes ont soulevé un grand nombre de problèmes difficiles. Nombre de ceux-ci ne sont d'ailleurs toujours pas résolus. Ainsi, alors que les robots manipulateurs se sont aujourd'hui généralisés dans l'industrie, rares sont les applications industrielles qui utilisent des robots mobiles. Si l'on a vu depuis peu apparaître quelques produits manufacturiers (chariots guidés) ou grand public (aspirateur), l'industrialisation de ces systèmes bute sur divers problèmes délicats. Ceux-ci viennent essentiellement du fait que, 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 pas structurés.

Néanmoins, l'intérêt indéniable de la robotique mobile est d'avoir permis d'augmenter considérablement nos connaissances sur la localisation et la navigation de systèmes autonomes. La gamme des problèmes potentiellement soulevés par le plus simple des robots mobiles à roues en fait un sujet d'étude à part entière et forme une excellente base pour l'étude de systèmes mobiles plus complexes.

Le contrôle du robot mobile, en vue d'une navigation, comme la poursuite d'une trajectoire désirée ou atteindre la position d'une cible tout en évitant les obstacles, sont actuellement parmi les plus grands défis dans le domaine du contrôle. Ce besoin est né à cause des nouvelles exigences technologiques. Les robots mobiles industriels actuels possèdent une pauvre souplesse ainsi qu'une autonomie limitée. Ils travaillent dans un environnement connu et utilisent des opérations séquentielles préprogrammées. Par contre, ils sont incapables d'évoluer dans des environnements dynamiques inconnus face à des situations imprévisibles. De l'autre côté, le contrôle classique tout en cherchant à prévoir tous les paramètres susceptibles d'influencer le processus, s'avère souvent inadapté lorsqu'il s'agit de prendre les bonnes décisions en cas de perturbations aléatoires. Ces difficultés ont pour origine la nature réelle du monde, avec tout ce qu'il représente comme incertitude et non structuration. En effet, l'information sur l'environnement est en général incomplète, incertaine et approximative. Des mesures bruitées ou capteurs défaillants, mènent souvent à des données imprécises, qui auront comme conséquences des croyances incorrectes.

I.2 Définition d'un robot mobile

En fait, il existe deux principaux types de robots mobiles à savoir [1-5] : les robots manipulateurs et les robots mobiles. Les robots manipulateurs ont une base fixe contrairement aux robots mobiles qui peuvent se déplacer. Les robots mobiles à roues sont en effet les systèmes les plus étudiés, parce qu'ils sont plus simples à réaliser que les autres types de robots mobiles, ce qui permet d'en venir plus rapidement à l'étude

de leur navigation [6-7]. Ce type de robots est notamment très souvent utilisé pour l'étude des systèmes autonomes. Vient ensuite la robotique mobile à pattes, avec notamment la robotique humanoïde, mais également des robots avec un nombre de pattes plus élevés qui offrent de bonnes propriétés pour la locomotion en milieu difficile (milieux forestiers et agricoles). La stabilité des mouvements de ce type de robots est en particulier un thème de recherche important [8]. Enfin il existe également de nombreux autres types de robots mobiles (robots marins, sous-marins [9], drones volants, micro et nano robots), généralement l'étude de ce type de robots se fait dans des thématiques spécifiques avec des problèmes particuliers à l'application visée.

I.3 Autonomie d'un robot mobile

Il existe deux principaux modes de fonctionnement pour un robot mobile [10] : télé-opéré et autonome. En mode télé-opéré, une personne pilote le robot à distance. Elle donne ses ordres via une interface de commande (joystick, clavier/souris...), et ceux-ci sont envoyés au robot via un lien de communication (internet, satellite ...). D'ailleurs, suivant le niveau de télé-opération, le terme "robotique" est plus ou moins justifié. Le robot doit donc obéir aux ordres de l'opérateur qui perçoit l'environnement autour du robot, par différents moyens (retour d'image, retour d'effort, ...), de manière à donner des ordres adaptés au robot. Dans ce domaine, les efforts de recherche sont beaucoup portés sur les problèmes liés au réseau de télécommunication (retards dans le réseau de communication, problèmes de commande, pertes de données) et sur l'amélioration de la perception de l'environnement par l'opérateur (interfaces images, retours d'efforts). A l'inverse, en mode autonome, le robot doit prendre ses propres décisions. Cela signifie qu'il doit être capable à la fois de percevoir correctement son environnement, mais également de savoir comment réagir en conséquence, suivant le niveau d'autonomie. C'est à lui de planifier son parcours et de déterminer avec quels mouvements il va atteindre son objectif. Les recherches dans ce domaine portent principalement d'une part sur la localisation du véhicule autonome et la cartographie de son environnement, d'autre part sur le contrôle de tels véhicules (structure de contrôle, stratégies de commande, planification).

Cette notion d'autonomie prise en exemple ci-dessus, que nous pourrions qualifier de décisionnelle, ne doit pas être confondue avec celle d'autonomie énergétique (capacité du robot à gérer efficacement son énergie, à la préserver, voire à se ravitailler), même si ces deux notions sont étroitement liées : idéalement une des préoccupations principales d'un robot mobile totalement autonome (du point de vue décisionnel), serait en effet de pouvoir gérer de lui-même ses réserves d'énergie.

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

I.4 Les différents types de terrains

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

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

Les terrains accidentés correspondent généralement aux milieux en extérieur, comme des forêts, des champs en robotique agricole, ou encore des terrains rocheux. La différence avec les terrains plats est la présence de pentes, de bosses et de creux sur le terrain d'évolution du robot. Cela interdit d'utiliser une métrique standard 2D et cela complique pour beaucoup la détection d'obstacles et la modélisation des déplacements du robot. De plus il devient également important de vérifier que le robot ne bascule pas quand il escalade une pente ou enjambe un obstacle. Le système de locomotion du robot doit dans ce cas être adapté à la topologie du terrain.

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

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

I.5. Modélisation cinématique d'un robot mobile à roues

I.5.1 Introduction

Les robots mobiles à roues sont de différents types, et suivant la disposition des roues sur la plateforme, on a le robot type unicycle, type tricycle, type voiture et type omnidirectionnel.

I.5.2 Repérage d'un robot mobile

Pour la modélisation terrain plat, nous définissons un repère absolu (fixé dans l'environnement) $R=(0, \vec{x}, \vec{y}, \vec{z})$, donc l'axe \vec{z} est perpendiculaire au sol. Nous définissons un repère mobile lié au robot $R^r = (O_r, X_r, Y_r)$ dit égoцентриque. Le point O_r est le point de contrôle du robot. Généralement, sur un robot type voiture, le point de contrôle est fixé au centre de l'essieu non directeur (figure I.1). Ce repère égoцентриque se déplace avec le robot [11]. Pour réaliser une navigation, l'état du robot est totalement défini par le vecteur [12] :

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

Dans lequel (x, y) désignent la position du robot dans le plan (o, \vec{x}, \vec{y}) et θ son orientation (Figure I.1).

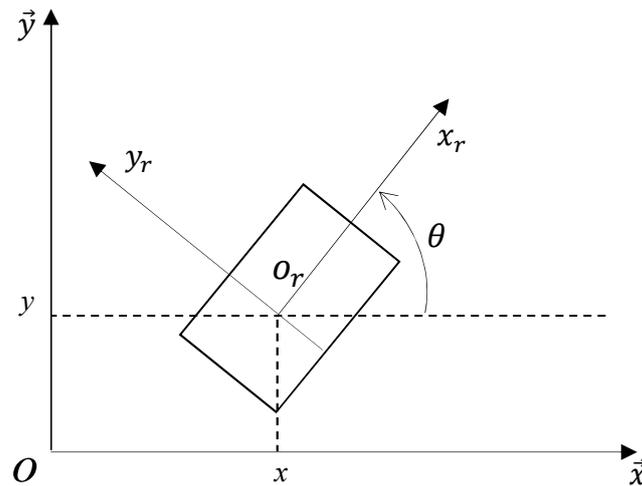


Figure I.1 Modélisation du robot dans le repère absolu.

I.5.3. Roulement sans glissement

La locomotion à l'aide des roues exploite la friction au contact entre roue et sol. Pour cela, la nature du contact (régularité, matériaux en contact) a une forte influence sur les propriétés du mouvement relatif de la roue par rapport au sol. Dans de bonnes conditions, il y'a roulement sans glissement (r. s. g.) de la roue sur le sol, c'est-à-dire que la vitesse relative de la roue par rapport au sol au point de contact est nulle. Théoriquement, pour vérifier cette condition, il faut réunir les hypothèses suivantes :

- le contact entre la roue et le sol est ponctuel.
- les roues, de rayon r , sont indéformables.

En pratique le contact se fait sur une surface, ce qui engendre bien évidemment de légers glissements. De même, alors qu'il est raisonnable de dire que des roues pleines sont indéformables, cette hypothèse est largement fautive avec des roues équipées de pneus.

Malgré cela, on supposera toujours qu'il y'a roulement sans glissement et, par ailleurs, que le sol est parfaitement plan.

Mathématiquement, on peut traduire la condition de (r. s. g.) sur une roue. Soit P le centre de la roue, Q le point de contact de la roue avec le sol, φ l'angle de rotation propre de la roue et θ l'angle entre le plan de la roue et le plan (O, \vec{x}, \vec{y}) comme indique a la (figure I.2).

La nullité de la vitesse relative \vec{v}_Q roue/sol au point de contact permet d'obtenir une relation vectorielle entre la vitesse \vec{u}_P du centre P de la roue et le vecteur vitesse de rotation $\vec{\omega}$ de la roue :

$$\vec{v}_Q = \vec{u}_P + \vec{\omega} \wedge \overrightarrow{pQ} = \vec{0} \quad (\text{I.2})$$

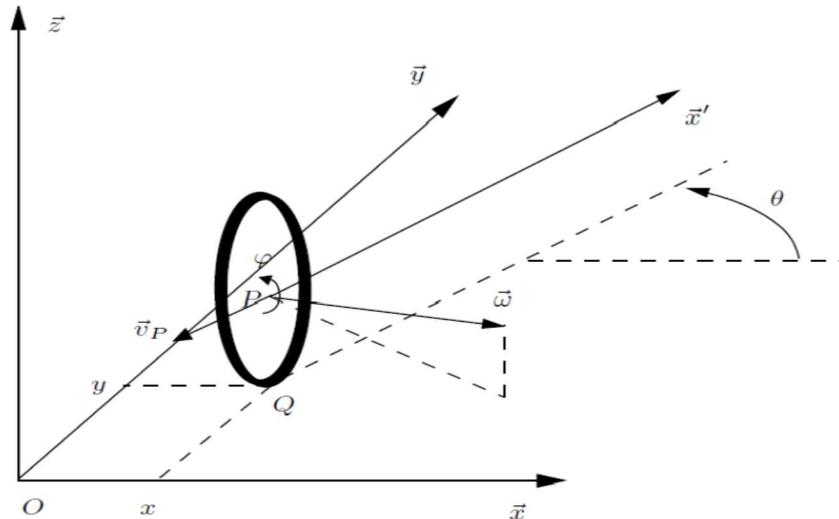


Figure I.2 Caractérisation du roulement sans glissement.

Les points P et Q ont pour coordonnées respectives $(x, y, r)^T$ et $(x, y, 0)^T$ Il vient alors :

$$\dot{x}\vec{x} + \dot{y}\vec{y} + (\dot{\theta}\vec{z} + \dot{\phi}(\sin\theta\vec{x} - \cos\theta\vec{y})) \wedge (-r\vec{z}) = \vec{0} \quad (\text{I.3})$$

$$(\dot{x} + r\dot{\phi}\cos\theta)\vec{x} + (\dot{y} + r\dot{\phi}\sin\theta)\vec{y} = \vec{0} \quad (\text{I.4})$$

Ceci nous donne le système de contraintes scalaires :

$$\dot{x} + r\dot{\phi}\cos\theta = 0 \quad (\text{I.5})$$

$$\dot{y} + r\dot{\phi}\sin\theta = 0 \quad (\text{I.6})$$

Que l'on peut transformer pour faire apparaître les composantes de vitesse dans le plan de la roue d'une part et perpendiculairement à la roue d'autre part :

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \quad (\text{I.7})$$

$$\dot{x}\cos\theta + \dot{y}\sin\theta = -r\dot{\phi} \quad (\text{I.8})$$

Les équations précédentes, caractérisant le roulement sans glissement d'une roue sur le sol, sont des contraintes non holonomes.

I.5.4 Classification des types des roues

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

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

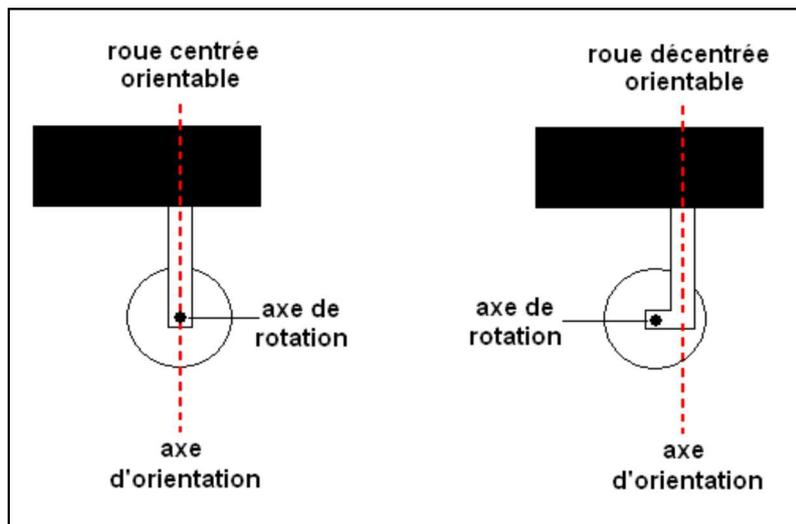


Figure I.3 Roue centrée et roue décentrée.

En plus de ces roues classiques (Figure I.3), d'autres roues ont été développées pour accroître la mobilité du robot. Elles permettent d'augmenter les capacités de déplacement dans toutes les directions du plan (Figure I.4). Cependant, elles ne sont commandables que dans certaines de ces directions. Dans cette catégorie nous trouvons notamment :

- les roues suédoises : ces roues autorisent les glissements latéraux grâce à un système de galets remplaçant la bande de roulement classique, montés en inclinaison par rapport au plan de la roue. La combinaison de la rotation de la roue avec la rotation libre du

galet en contact avec le sol permet le déplacement sans glissement sur le sol dans toutes les directions. Cependant le couple moteur que l'on peut transmettre à ces roues est très limité, ce qui réduit son utilisation en pratique.

- les roues tronco-sphériques (ou orthogonal wheels) : cette structure utilisant deux roues libres en quadrature présente l'avantage de pouvoir transmettre un couple intéressant par rapport aux roues suédoises, mais souffre de petits problèmes de sauts au moment de la transition d'une roue support à l'autre.

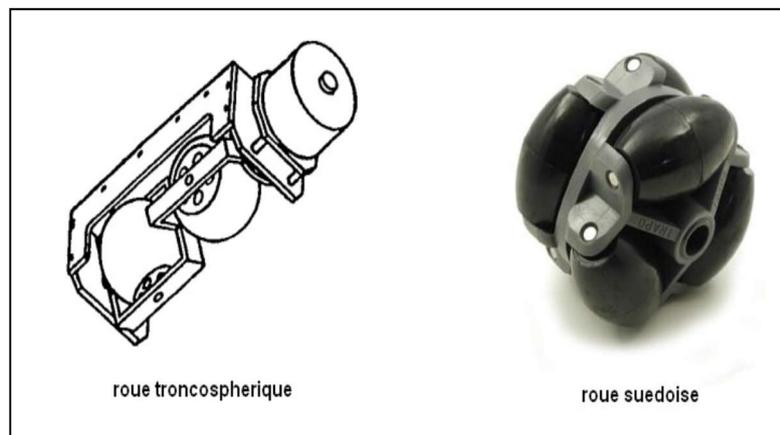


Figure I.4 Roue troncosphérique et roue suédoise.

I.5.5 Types des plates formes mobiles

En associant les différents types de roues selon une structure mécanique donnée, le robot mobile disposera de plus ou moins de mobilité. Le nombre, le type et la disposition des roues engendrera ou non la contrainte de non holonomie du robot. Si on néglige les phénomènes dynamiques tels que l'inertie, un robot holonome est un robot capable à chaque instant de se déplacer dans n'importe quelle direction du plan, sans avoir à effectuer une reconfiguration de ses roues.

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

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

I.5.5.1 Plates formes non holonomes

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

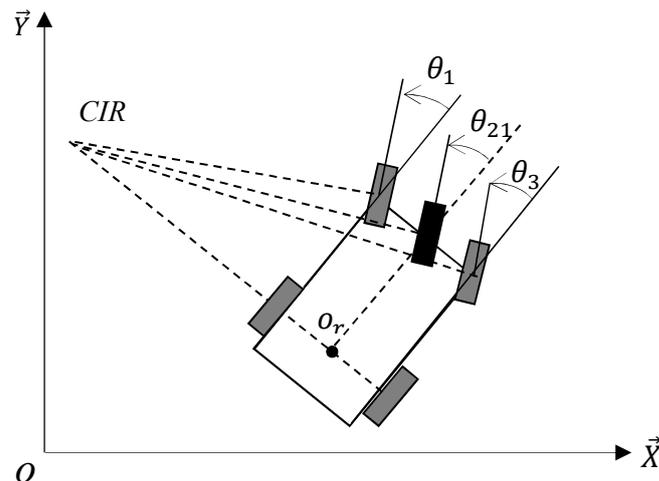


Figure I.5 Equivalence modèle voiture et tricycle.

Les configurations non holonomes les plus courantes sont :

a) Plates formes de type tricycle et voiture : ces deux structures sont constituées d'un axe fixe (généralement à l'arrière) et d'un axe directeur (Figure I.5). Dans le cas du tricycle, seule une roue est présente sur l'axe directeur, contrairement à la voiture qui en possède deux.

La théorie d'Ackerman-Jeantaud donne les conditions théoriques de non glissement et non dérapage pour les configurations de type voiture. Notamment les axes de rotation

des 4 roues doivent s'intersecter en un point unique, le Centre Instantané de Rotation. Pour cela, la vitesse de la roue extérieure doit être légèrement supérieure à celle de la roue intérieure. La structure de type voiture peut être modélisée par une structure équivalente à 3 roues, ce qui revient au modèle du tricycle.

b) Plates formes à roues différentielles : cette structure est constituée de deux roues motrices placées sur le même axe (Figure I.6), et d'au moins un appui supplémentaire (généralement une ou deux roues folles). L'avantage de cette structure est qu'elle permet au véhicule de tourner sur place, suivant si les vitesses de rotation des deux roues motrices sont de signe opposées ou pas. Ainsi le robot peut pivoter rapidement, ce qui donne des capacités de déplacement intéressantes. Cependant, le déplacement latéral n'étant pas directement réalisable, cette structure n'est pas non plus holonome.

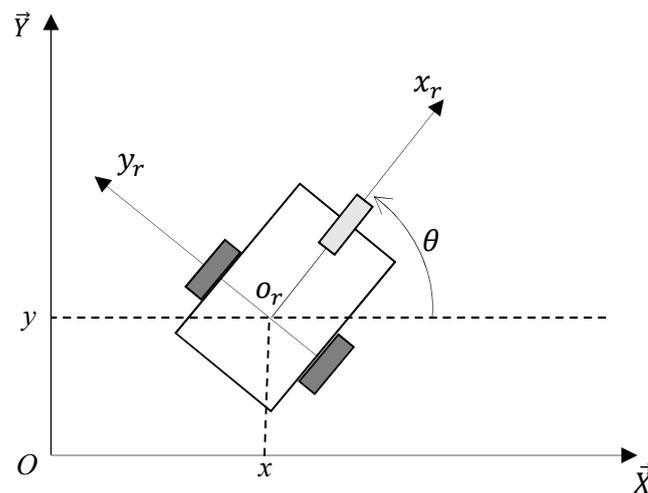


Figure I.6 Modèle de type tricycle à roues différentielles.

Dans notre travail, nous utilisons un modèle de ce type des robots pour tester les structures de commande étudiées.

I.5.5.2 Plates formes holonomes

En utilisant des roues telles que les roues suédoises ou tronco-sphériques sur des plateformes mobiles, des robots ayant la capacité de se mouvoir dans toutes les directions ont été créés. Ces robots, à 3 degrés de mobilité dits omnidirectionnels, permettent de s'affranchir de la contrainte de non holonomie. Leurs structures spécifiques leur permettent de se déplacer instantanément (à la dynamique près) dans

toutes les directions en ayant n'importe quelle orientation, rendant possible le suivi de trajectoires de forme quelconque (Figure I.7).

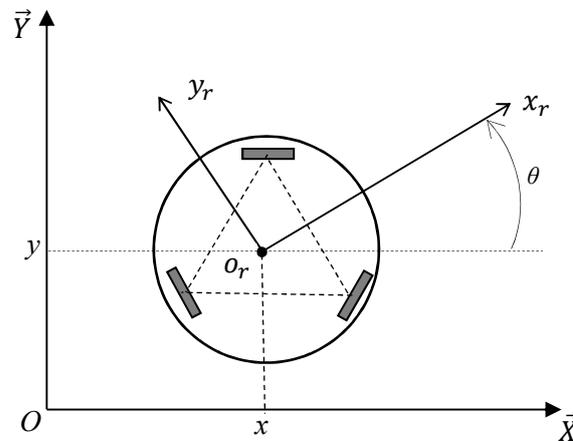


Figure I.7 Le robot omnidirectionnel.

I.6 Les capteurs comme sources d'informations

La commande des robots mobiles est basée sur deux types d'informations importantes ; les informations proprioceptives et les informations extéroceptives [35, 36]. Le système de perception est très important pour la sécurité du robot, si l'environnement est encombré d'obstacles fixes ou bien mobiles (autres robot). Pour se focaliser sur le problème de navigation, nous allons nous restreindre dans ce chapitre aux capteurs utiles pour la tâche de navigation.

I.6.1 Capteurs intéroceptifs

Fournissent des données sur l'état interne du robot (vitesse, position, orientation,...). Ces informations renseignent le robot en cas de mouvement, sur son déplacement dans l'espace (la localisation). Ce sont des capteurs que l'on peut utiliser directement, mais ils souffrent d'une dérive au cours du temps qui rend leur utilisation seule inefficace ou avec limitation. Nous citons par exemple : l'odomètre, radar doppler, systèmes inertiels,... [35, 37].

I.6.2 Capteurs extéroceptifs

Ont pour objectif d'acquérir des informations sur l'environnement proche du véhicule. Ils fournissent des mesures caractéristiques de la position que le robot peut acquérir dans son environnement par la détection des objets qui contourne. Ces informations peuvent être de natures très variées. Nous citons comme exemple les

télémètres à ultrason, infrarouge, laser, les caméras,...etc. Pour la navigation autonome d'un robot mobile et selon la mission visée (à accomplir), on peut utiliser aussi les capteurs suivants [35-37] :

- *Les capteurs tactiles* : qui sont le plus souvent utilisés pour des arrêts d'urgence lorsqu'ils rencontrent un obstacle qui n'avait pas été détecté par le reste du système de perception.
- *Les boussoles* : permettant par la mesure du champ magnétique terrestre, de déduire la direction du nord. Ces capteurs peuvent utiliser différentes technologies et ont l'avantage de fournir une direction de référence stable au cours du temps.
- *Les balises* : dans certaines applications, il est également possible d'utiliser des balises dont on connaît la position, et qui pourront être facilement détectées par le robot, afin de faciliter sa localisation. Le robot sera alors équipé d'une antenne directionnelle qui lui permettra de détecter la direction des différentes balises afin de déduire sa position.
- *Le GPS (Global Positioning System)* : est un système de balises universel dont les balises sont placées sur des satellites en orbite terrestre. Ce système permet donc d'avoir une mesure de la position dans un repère global couvrant la terre avec une précision variant de quelques dizaines de mètres à quelques centimètres suivant les équipements utilisés.

I.7 Navigation Autonome d'un robot mobile

La navigation d'un robot mobile est une tâche qui consiste, généralement, à trouver un mouvement libre dans l'espace de configuration (environnement de travail) sans collisions avec les obstacles proche du robot. L'espace de configuration est l'ensemble des paramètres caractérisant la position du robot dans son environnement (position et orientation).

Ce mouvement amène le robot d'une configuration initiale $q_0=q(t_0)=(x_0, y_0, \theta_0)$ à une autre finale désirée $q_f=q(t_f)=(x_f, y_f, \theta_f)$ [38] [39]. La figure I.8 présente une description de la tâche de navigation d'un robot mobile. Comme montré, le robot démarre d'une situation initiale s , il doit exécuter les actions de mouvement, qui sont généralement la vitesse et l'angle de braquage ($v(t)$ et $w(t)$) lui permettant de se mouvoir vers une nouvelle situation $s(t+1)$. La navigation est obtenue à travers un processus itératif comme suit :

1. À chaque instant t , avec $t = 0, 1, \dots, k, \dots$, le robot doit mesurer les distances aux obstacles de l'environnement d_i et les positions : courante et finale : $(x_r(t), y_r(t), \theta_r(t))^T, P_g(x_g, y_g)$;
2. Le système de contrôle détermine les variables de commande adéquate $v_r(t+1)$ et $\omega(t+1)$,
3. Le robot exécute ces actions en se déplaçant vers les nouvelles coordonnées,
4. Répéter le même processus (les étapes : 1, 2 et 3) de détection de la situation et génération des actions jusqu'à la destination finale appelée but [33].

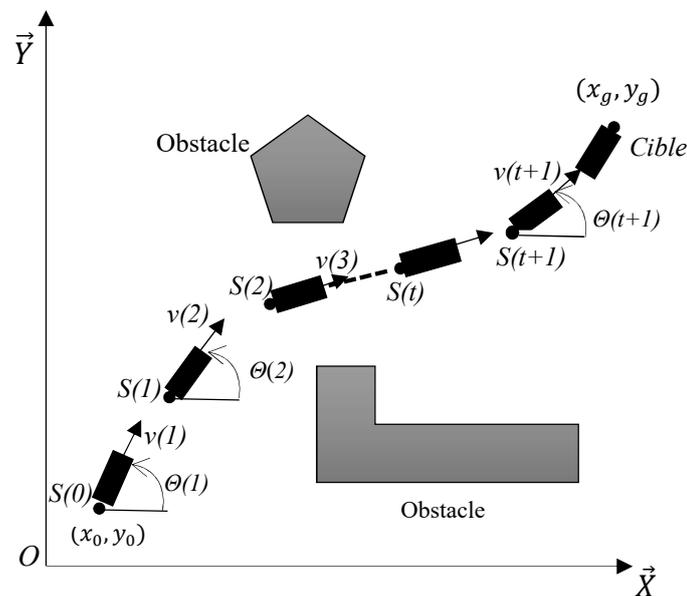


Figure I.8 Processus de navigation autonome d'un robot mobile.

I.8 Les thématiques scientifiques de la robotique mobile

Il existe de nombreuses thématiques de recherche dans le milieu de la robotique mobile autonome, ce qui montre qu'aujourd'hui encore le problème spécifique des robots mobiles autonomes est entier. La communauté des chercheurs dans le domaine de la robotique a dégagé 4 grands axes de travail autour desquels s'articulent les recherches actuelles en robotique mobile [10] :

I.8.1 Techniques de localisation et cartographie

Cet axe regroupe tous les développements autour de la perception et de la localisation du robot. On y retrouve notamment les méthodes SLAM (Localisation et Cartographie Simultanées). Plus récemment l'utilisation de bases de données sous forme

de cartes 2D ou 3D, mais également sous forme SIG (Système d'Informations Géographiques) qui a ouvert de nouvelles perspectives dans ce domaine. De manière générale la fusion de données est également un thème important, tant la nécessité de coupler diverses sources de mesures restant nécessaire pour améliorer la précision et garantir l'intégrité des informations,

I.8.2 Contrôle et commande des véhicules

Cet axe regroupe les thématiques liées à la planification de chemin, la génération de trajectoires, et la commande des robots de manière générale. Une prise en compte de plus en plus poussée des contraintes et de la dynamique des robots est nécessaire, pour adapter au mieux les robots à leur environnement. La bonne gestion des obstacles et la prise en compte des incertitudes de mesures sont également des points clés de cette thématique,

I.8.3 La communication inter-véhicules

On retrouve ici tous les travaux liés à la coopération entre robots, et le contrôle de flottilles de véhicules,

I.8.4 L'interprétation de scènes

Les recherches dans ce domaine visent à pousser plus loin la perception de son environnement par le robot, que la simple reconnaissance des objets. En effet dans certaines applications il est nécessaire que le robot appréhende plus finement son environnement que par une simple détection et localisation des obstacles. Les travaux concernent notamment la perception multi-capteurs et la représentation dynamique des scènes.

I.9 Planification de trajectoire

L'un des principaux objectifs du robot mobile est de pouvoir évoluer dans un environnement complexe encombré d'obstacles pour atteindre son but final. Il a besoin de construire une trajectoire définie comme une séquence de déplacement sans collision avec ces obstacles entre la position initiale (point de démarrage) et le point but ou cible. La planification de trajectoire dans sa formulation classique est le problème du calcul

de ce chemin, dans un modèle géométrique de l'environnement cela est fait en introduisant le concept d'espace des configurations qui permet de transformer le problème de la recherche d'un chemin pour un système à n degrés de liberté dans l'espace euclidien en celui du mouvement d'un point dans un espace à n dimensions où le robot est représenté par un point. Plusieurs approches sont proposées pour la planification de trajectoire. Cependant, les plus utilisées sont la planification globale et locale [34].

I.9.1 Planification globale de trajectoire

C'est la modélisation de l'espace de l'environnement par un graphe, ou la recherche de la trajectoire est basée sur l'utilisation des algorithmes des graphes, on peut citer : le graphe de visibilité, la décomposition cellulaire...etc.

I.9.2 Planification locale de trajectoire

Généralement, l'environnement du robot mobile est inconnu, et le robot ne dispose pas, à priori, aucune information sur l'environnement. Il est nécessaire donc de réaliser une planification locale de type réflexe. Pendant le déplacement, le robot mobile doit analyser son environnement et prendre la décision en fonction de cette analyse [34]. Les méthodes réactives de l'intelligence artificielle sont considérées comme des approches de planification locale.

Il existe donc une grande variété d'algorithmes permettant de réaliser une telle technique. On présentera certaines de ces méthodes en expliquant brièvement leur principe ainsi que leurs avantages et leurs inconvénients.

I.9.2.1. Méthode par décomposition de l'environnement en cellule

Une première méthode de planification de trajectoire consiste à décomposer l'environnement du robot en cellules [15]. Il suffit ensuite de trouver un algorithme travaillant sur la discrétisation de l'environnement. Différentes techniques existent, par exemple, la partition de Voronoï [16] ou les graphes de visibilité [17].

Dans ces cas, l'environnement discrétisé est représenté dans un graphe et trouver une trajectoire revient à chercher un chemin dans le graphe, problème qui peut être facilement traité informatiquement [18].

D'autres algorithmes de planification travaillent dans un environnement discrétisé. Parmi les plus connus, on peut citer P. Hart [19] qui planifie une trajectoire optimale connaissant l'environnement ou Stentz et Anthony [20] qui planifie une trajectoire dynamiquement, ce qui lui permet de découvrir son environnement au fur et à mesure.

Les avantages de tels algorithmes sont principalement la facilité d'implémentation et de représentation. On travaille sur une grille et chaque case est en 8-connexité. C'est-à-dire que dans une case, on a que 8 choix de déplacement possible.

Par contre, ces algorithmes possèdent trop d'inconvénients. Notamment le fait que les contraintes cinématiques du robot ne sont pas prises en compte (accélération, vitesse, vitesse angulaire maximale, encombrement etc.), ce qui permet au planificateur de trouver une trajectoire qui ne sera pas exécutable par le robot. De plus, le fait de discrétiser l'espace de recherche limite beaucoup le nombre de trajectoires possibles.

I.9.2.2. La méthode des champs de potentiel

La méthode de champs de potentiel [21] est une méthode assez originale qui assimile le robot à une particule soumise à un champ de forces répulsives et attractives.

Un obstacle génère un champ de potentiel répulsif tandis que l'objectif à atteindre génère un champ de potentiel attractif. L'algorithme calcul donc un vecteur résultant qui indiquera au robot comment effectuer son déplacement vers la cible.

Cet algorithme est totalement réactif et peut donc être très facilement implémenté en temps réel. Cependant, comme l'environnement est très peu souvent totalement convexe, cette méthode entraîne facilement le robot dans des minima locaux. De plus, les problèmes d'oscillation peuvent, dans certain cas, être constatés. Ici non plus, on ne tient pas compte des contraintes du robot.

I.9.2.3. La méthode de la fenêtre dynamique

Cette méthode [22] travaille dans l'espace de commande du robot. La méthode calcule les vitesses possibles du robot pour que celui-ci ne rentre en collision avec aucun obstacle. Une fois l'espace de recherche des vitesses calculées, l'algorithme trouvera la commande optimale à envoyer au robot en minimisant une fonction de coût sur ce domaine de recherche. (Minimisation du temps de parcours, de l'énergie dépensée, maximisation de la vitesse etc..).

Cette méthode un peu plus abstraite permet de prendre en compte les contraintes cinétiques du robot. Par contre, elle manque en flexibilité, ce qui rend difficile son implémentation dans un cadre multi-robot.

I.9.2.4. La méthode de la bande élastique

C'est une méthode [23] qui tend une "bande élastique" entre le robot et l'objectif. Cette bande étant capable de se déformer en présence d'un obstacle, celle-ci génère donc une trajectoire envisageable par le robot.

I.9.2.5. La planification par logique floue

Cette méthode de planification se base sur la logique floue [24][33], Chaque grandeur physique (par exemple une distance d'un obstacle) est convertie en une variable linguistique (petit, moyen ou grand par exemple). Alors qu'en logique booléenne classique, une distance serait soit petite, soit grande, la logique floue autorise une distance à être à la fois grande et petite (20% petite et 80% grande) suivant une fonction d'appartenance précise.

La commande floue va donc raisonner non pas sur des grandeurs physiques, mais sur des variables linguistiques. On appliquera des règles précises sur ces variables. Finalement, on obtiendra la commande sous la forme de variables linguistiques que l'on reconvertira en grandeurs physiques.

Exemple : Si la distance au feu "d'un carrefour par exemple" est petite et le feu est rouge, on freine fort. Ici, on ne raisonne pas sur la valeur de la distance directement. (On ne va pas se dire, dans la vie de tous les jours : si on est à moins de 6 mètre du feu, alors, on fait telle action. Un humain raisonnera plutôt comme ça : si on est moyennement proche, si on est assez proche, si on est très proche alors, on fait telle action).

La logique floue permet de concevoir un système de commande générant des trajectoires progressives que le robot mobile devra suivre.

I.9.2.6. La planification par juxtaposition de splines polynomiales

L'algorithme de Mickel Defoort [25][26] génère des bouts de splines polynomiales qui respectent les contraintes cinématiques du robot. On peut simplifier l'algorithme en le forçant à générer non pas des splines, mais simplement des polynômes. La trajectoire du robot est alors assimilée à la juxtaposition de plusieurs polynômes.

Un bout de trajectoire est alors représenté par un système paramétré de deux polynômes du troisième degré. Les coefficients des degrés zéro et un, fixent la position initiale et la vitesse initiale du robot (ce qui permet de garder la continuité entre les différents polynômes). Ainsi, il ne reste plus qu'à optimiser les coefficients du second et du troisième degré du polynôme à l'aide d'algorithmes d'optimisation comme CFSQP (Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems) (algorithmes de résolution des problèmes d'optimisation non linéaire) [27].

Cette méthode permet donc de respecter les contraintes cinématiques du robot. De plus, l'optimisation des paramètres des polynômes peut prendre en compte des fonctions de coût, ce qui permet de choisir selon quelle critère l'on décide qu'une trajectoire est optimale ou non.

I.9.2.7. DKP Deterministic Kinodynamic Planning

Cette méthode de planification [28] est assez complexe, mais permet de respecter toutes les contraintes du robot et d'éviter les problèmes des minimas locaux.

La méthode DKP peut être décomposée en deux parties. La première planifie localement des bouts de trajectoires polynomiales (d'ordre deux) sur différents temps de parcours. Pour trouver un bout de trajectoire optimal, il suffit de chercher une solution optimale dans l'espace des paramètres des polynômes via une résolution géométrique des contraintes. Avec ce planificateur local, on construit un arbre de solutions locales.

La seconde partie de l'algorithme est un planificateur global qui va choisir les bouts de trajectoire à adopter pour résoudre le problème.

L'avantage de cet algorithme est qu'il génère une grande diversité de bouts de polynôme, ce qui permet de trouver une solution même dans les environnements très complexes.

I.9.2.8. La planification par Réseaux de neurones (RNA)

Le neurone est l'entité de base du système de réaction animal, et les neurologues Mc Culloch et Pitts furent les premiers à étudier leur fonctionnement chez la grenouille [29]. Les réseaux de neurones désignent à la fois l'étude de ces systèmes biologiques et leur modélisation informatique, plus ou moins simplifiée, à différentes applications, comme la reconnaissance de caractères, son premier terrain d'application historique.

Dans un réseau de neurones artificiel (figure I.9), chaque neurone possède plusieurs entrées et une sortie. Chacune de ses entrées se voit affecter un poids (dit poids synaptique) différent, et si la somme ainsi pondérée des signaux des différentes entrées dépasse un seuil, la sortie prend une valeur positive (le neurone se déclenche).

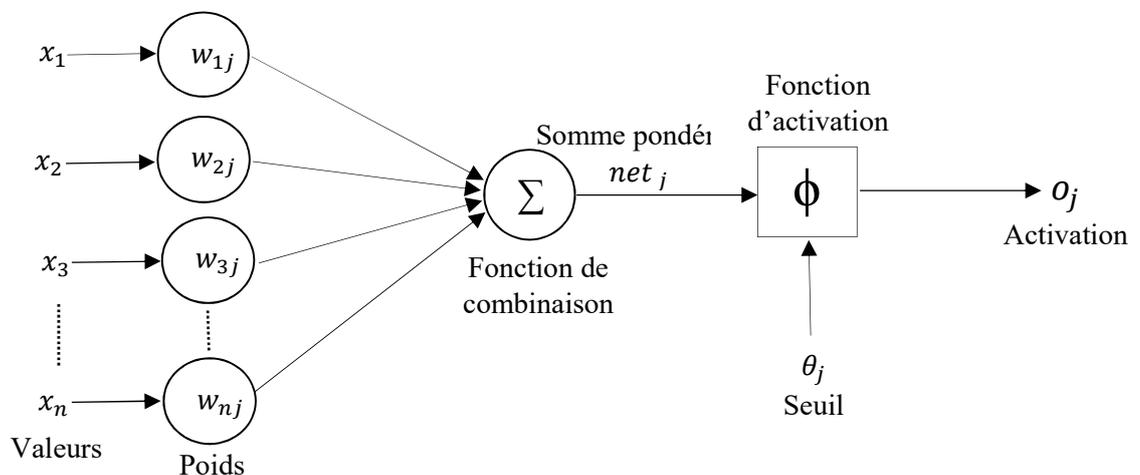


Figure I.9 Modélisation d'un réseau de neurones artificiel.

Un réseau complet est constitué de différentes couches de neurones. Les premières couches correspondent à des couches de détection, en robotique les neurones correspondants sont directement liés aux informations délivrées par les capteurs du robot. Les couches supérieures correspondent plutôt à des couches d'interprétation, et pour un robot, elles seront plutôt liées aux effecteurs. De plus il y a généralement un certain nombre de couches intermédiaires entre les couches d'entrée et de sortie.

Un apprentissage du réseau est réalisable en adaptant les poids des entrées synaptiques. Pour cela, le principe est de donner au réseau un grand nombre d'exemples, et de rétro-propager l'erreur obtenue entre la sortie du réseau et la sortie attendue. L'algorithme d'apprentissage aura alors pour tâche de régler les différents poids synaptiques pour minimiser cette erreur.

De nombreux contrôleurs pour robots mobiles ont été construits sur ces principes [30] [31]. Les dernières générations utilisent un codage en fréquence pour la transition de l'information, comme le font les neurones biologiques. Sur une fenêtre temporelle donnée, le nombre de pics émis quantifie par exemple la proximité d'un obstacle. Wang dans [32] a développé un contrôleur basé sur ce principe, pour un robot unicycle équipé de capteurs ultrasonores.

I.10 Conclusion

L'objectif de ce chapitre est de donner un aperçu général sur la robotique mobile, après la présentation des différents types des robots mobiles, et les contraintes de terrain sur lequel ils sont conçus pour évoluer, nous aborderons la modélisation cinématique et la classification des plateformes mobiles, leurs principales structures cinématiques, en définissant la navigation autonome des robots mobiles et les capteurs utilisés. Les thématiques de recherche actuelles associées au domaine de la robotique mobile sont présentées pour aboutir au développement d'un robot mobile. Parmi celles-ci, nous nous focalisons alors sur celle de la navigation d'un robot mobile.

La fin de ce chapitre ne fait pas un état de l'art rigoureux des méthodes de planification de trajectoire. C'est juste un aperçu non exhaustif de ce qui existe déjà. Parmi ces algorithmes, nous avons cité la méthode des champs de potentiel et la

planification par logique floue, La planification par Réseaux de neurones sur les robots mobiles. Ces algorithmes ont leurs avantages et leurs inconvénients. De plus, dans certains cas, une seule méthode de planification ne suffit pas, mais il est avantageux de tester quelques techniques de planifications différentes.

Après cet aperçu, dans le chapitre suivant nous présenterons une étude générale sur les systèmes d'inférences flous et les fondements théoriques des ensembles flous. Nous décrivons la structure générale d'un contrôleur flou avec ces composants pour l'utiliser par la suite au problème de navigation d'un robot mobile.

REFERENCES BIBLIOGRAPHIQUES CHAPITRE I

- [1] P. Coiffet, "La robotique : principes et applications", Edition Hermès, 1986.
- [2] J. J. Craig, "Introduction to Robotics, Mechanics and Control", Addison Wesley Publishing Company, 1986.
- [3] K. S. Fu, R. C. Gonzalez & C. S. G. Lee, "Robotics: Control, Sensing, Vision, and Intelligence", Mc Gram-Hill Book Company, 1987.
- [4] E. Dombre, W. Khalil, "Modélisation et commande des robots", Edition Hermès Paris, 1988.
- [5] A. Pruski, "Robotique Générale", Ellipses, 1988.
- [6] J. P. Laumond, "La robotique mobile", Hermes Sciences Europe Ltd, 2001.
- [7] B. Bayle, "Robotique mobile", Thèse de doctorat, Ecole Nationale Supérieure de Physique de Strasbourg, Université de Strasbourg, France, 2008.
- [8] S. Furuno, M. Yamamoto & A. Mohri, "Trajectory planning of mobile manipulator with stability considerations", Proc. of the 2003 IEEE Inter. Conf. on Robotics and Automation, Taipei, Taiwan, pp. 3403-3408, September 2003.
- [9] S. M. Lee, K. Y. Kwon & J. Joh, "A Fuzzy logic for autonomous navigation of marine vehicles satisfying COLREG guidelines", Inter. Journal of Control, automation and Systems, Vol. 2, No. 2, pp. 171-181, June 2004.
- [10] N. Morette, "Contribution à la navigation de robots mobiles : approche par modèle direct et commande prédictive", Thèse de doctorat de l'Institut Prisme, Equipe Systèmes Robotiques interactifs, Université d'Orléans, France, 2009.
- [11] J. Fourquet, M. Renaud. "Coordinated Control of a Non-Holonomic Mobile Manipulator". In ISER'1999, pages 115–125, Sydney, Australie, mars 1999.
- [12] G. Campion, G. Bastin & B. D'Andréa-Novel. "Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots". IEEE Transactions on Robotics and Automation, vol. 12, no. 1, pages 47–62, 1996.
- [13] D. Filliat, "Cartographie et estimation globale de la position pour un robot mobile autonome", Thèse de doctorat de l'université Paris 6 Pierre et Marie Curie, France, 2001.
- [14] H. Miloud, "navigation des robots mobiles non-holonomes sous contrôle flou ", Thèse de doctorat du Département d'Electronique de l'Université de Ferhat Abbes Sétif -1, 2016.
- [15] J. C. Latombe. "Robot Motion Planning". Kluwer Academic Publishers, Boston, 1991.
- [16] V. Choset, "Sensor Based Motion Planning": the Hierarchical Generalized Voronoi Graph (Thèse de doctorat, California Institute of Technology), 1996.
- [17] Chazelle, Guibas, " Visibiliy and ontersection problems in plane geometry", (Article, Discrete and Computational Geometry), 1989.

- [18] Shin, McKay, "A dynamic programming approach to trajectory planning of the robotic manipulators", (Article, IEEE Transactions on Automatic Control). 1986.
- [19] P. Hart, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". (Article, IEEE Transactions on Systems Science and Cybernetics), vol, SSC-4, NO. 2 July. 1968.
- [20] Stentz, Anthony, "Optimal and Efficient Path Planning for Partially-Known Environments". (Article, Proceedings of the International Conference on Robotics and Automation), 1994.
- [21] O. Khatib, "Real time obstacle avoidance for manipulators and mobile robots". IEEE, International Conference on Robotics and Automation, pages 500-505, 1985.
- [22] Fox, al. "The dynamic window approach to collision avoidance". (Article, IEEE Robotics and Automation Magazine), 1997.
- [23] Quinlan, "Real-Time Path Modifications of Collision-Free Paths", (Thèse de doctorat, Université de Stanford), 1994.
- [24] L. A. Zadeh, "Fuzzy Sets", Information and Control, vol.8, pp 29-44, 1965.
- [25] M. Defoort, "Contributions à la planification et à la commande pour les robots mobiles coopératifs". (Thèse de doctorat, Ecole centrale de Lille), 2007.
- [26] M. Defoort, "Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach". (Article Robotics and Autonomous Systems). 2009.
- [27] Lawrence, Zhou & Tits, "User's Guide for CFSQP Version 2.0: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems", Generating Iterates Satisfying All Inequality Constraints. University of Maryland, College Park, MD 20742, (Institute for Systems Research TR-94-16r1).
- [28] Gaillard, al. "Deterministic Kinodynamic Planning". (Workshop of the UK Planning and Scheduling Special Interest Group, Italy), 2010.
- [29] J.Y. Lettvin, H.R. Maturana, W.S. McCulloch, & W.H. Pitts , "What the Frog's Eye Tells the Frog's Brain", Proceedings of the IRE, Vol. 47, No. 11, pp. 1940-51, 1959.
- [30] D.V. Lebedev, J.J. Steil, & H.J. Ritter, "The dynamic wave expansion neural network model for robot motion planning in time-varying environments"; Neural Networks, Volume 18, Issue 3, April 2005, Pages 267- 285.
- [31] T. Belker, D. Schulz, "Local Action Planning for Mobile Robot Collision Avoidance", Intelligent Robots and System, 2002. IEEE/RSJ International Conference on Volume 1, 30 Sept.-5 Oct. 2002 Page(s):601 - 606 vol.1.
- [32] X. Wang, Z. Hou, & A. Zou, M. Tan, L. Cheng, "A behavior controller based on spiking neural

networks for mobile robots". Neurocomputing Volume 71, Issue 4-6 (January 2008) Pages 655-666.

- [33] **K. boucetta, A. hassam, "Comparative Study between Fuzzy Logic and Interval Type-2 Fuzzy Logic Controllers for Trajectory planning of a Mobile Robot ", Engineering, Technology & Applied Science Research, Vol. 11, No. 2, 2021, 7011-7017, 2021.**

- [34] C. Lakhmissi, "Navigation Autonome d'un Robot Mobile par des Techniques Neuro-Floues", Thèse de doctorat du Département de Génie Electrique de Université Mohamed Khider - Biskra, 2014.

- [35] J. Borenstein, H. R. Everett, and L. Feng, "Where am I, Sensors and Methods for Mobile Robot Positioning", University of Michigan, 1996.

- [36] S. G. Shuzhi, F. L. Lewis, "Autonomous Mobile Robots, Sensing, Control, Decision, *Making* and Applications", Taylor and Francis Group, 2006.

- [37] D. Filliat, "Robotique Mobile", Cours à l'école Nationale Supérieur des Techniques Avancées ENSTA, Octobre 2004.

- [38] O. Lefebvre, "Navigation Autonome sans Collision pour Robots Mobiles non holonomes", Thèse de Doctorat de l'Institut National Polytechnique de Toulouse, 2006.

- [39] **A. Hassam, B. Kasmi & M. Boubezoula, "Planning of a Trajectory Along with Avoiding the Obstacle Based on Fuzzy Logic Method for a Unicycle Mobile Robot", The 13th International Arab Conference on Information Technology ACIT'2012 Dec.10-13, ISSN: 1812-0857, 2013.**

Chapitre II

Les Systèmes d'Inférence Flous

II.1 introduction

Le nombre de travaux de recherche sur le développement des contrôleurs par la logique floue ainsi que le nombre d'applications industrielles de la commande floue a augmenté exponentiellement dans les trois dernières décennies. La théorie de la logique floue a été développée dans les années soixante à l'université de Berkeley par le professeur Lotfi Zadeh. Il travaillait à Berkeley depuis 1959 et il a été président de *l'Electrical Engineering department* de 1963 à 1968. Le texte *fuzzy sets* est apparu pour la première fois en 1965 [1] dans la revue *"Informatique and control."* Il est considéré généralement comme le début de la théorie, bien qu'une ébauche en ait été donnée plus tôt dans l'histoire, par Lukasiewicz en Pologne dans les années vingt. Depuis 1965, le professeur Zadeh n'a pas arrêté d'enrichir la théorie dans ce domaine avec ses innombrables publications dans différents journaux et conférences. C'est depuis les années quatre-vingt, que les idées formulées dans les années soixante ont été acceptées. Dans ses écrits, le professeur L. Zadeh raconte comment il a été marginalisé à cause des ses nouvelles idées qu'il véhiculait, tel n'était pas le cas au japon, où la théorie de la logique floue a reçu une très grande attention. Des sommes considérables ont fait l'objet de donations pour des fins de recherches dans ce nouveau domaine, ce qui a aboutit à de nombreux produits utilisant cette technique. Cependant, la première recherche pratique

sur cette nouvelle théorie a eu lieu au Queen Mary College de Londres. En effet, en 1974, Mamdani et ses étudiants motivés par les articles de Zadeh, ont réussi à établir le premier rapprochement entre la théorie du flou et le contrôle des systèmes. Il s'agissait de la régulation d'une machine à vapeur. Ce succès surprenant a motivé par la suite King et Mamdani [2] au contrôle de la température d'un réacteur chimique. Cependant, la première application industrielle était en 1980 destinée à la conduite d'un four à chaux de F.L.Smith & CO A/S au Danemark. Une autre application pratique de cette technique a été réalisée en 1983 au Japon ; pour la conduite d'installation d'assainissement d'eau.

Dans ce chapitre, nous exposons en bref un aperçu général sur la logique floue, les fondements mathématiques de la théorie des sous-ensembles flous et les systèmes d'inférence flous.

II.2 Concepts fondamentaux flous

Le concept d'ensemble flou a été introduit pour éviter le passage brusque d'une classe à une autre (par exemple, de la classe noire à la classe blanche) et autoriser des éléments à n'appartenir complètement ni à l'une ni à l'autre (à être gris, par exemple), ou encore à appartenir partiellement à chacune (avec un fort degré à la classe noire et un faible degré à la classe blanche, dans le cas du gris foncé). La définition d'un ensemble flou répond au besoin de représenter des connaissances imprécises, soit parce qu'elles sont exprimées en langage naturel par un observateur qui n'éprouve pas le besoin de fournir plus de précision ou n'en est pas capable, soit parce qu'elles sont obtenues avec des instruments d'observation qui produisent des erreurs de mesure [3-5].

La notion des ensembles flous nous permet de traiter :

- Des catégories aux limites mal définies.
- Des situations intermédiaires entre le tout et le rien.
- Le passage progressif d'une propriété à une autre.
- Des valeurs approximatives.

Donc, le concept des ensembles flous constitue un assouplissement de celui d'un ensemble donné.

II.3 Définition d'un ensemble flou

Soit le référentiel ou l'univers de discours U qui est le domaine de définition d'une variable x , un ensemble A dans U est défini par l'ensemble des paires ordonnées (élément générique de U , degré d'appartenance) :

$$A = \{(x, \mu_A(x)) / x \in U\} \quad (\text{II.1})$$

Où $\mu_A(x)$ est la fonction d'appartenance de x dans A , elle prend des valeurs comprises entre 0 et 1. Dans le cas particulier où μ_A ne prend que des valeurs égales à 0 ou 1, l'ensemble flou A est un ensemble classique de U . Un ensemble classique est donc un cas particulier d'ensemble flou.

L'univers de discours U peut contenir des valeurs continues ou discontinues, donc l'ensemble A de U s'écrit généralement sous la forme :

$$A = \begin{cases} \sum_{x \in U} \frac{\mu_A(x)}{x} & \text{Si } U \text{ est discret} \\ \int_U \frac{\mu_A(x)}{x} & \text{Si } U \text{ est continu} \end{cases} \quad (\text{II.2})$$

Cette notation est classiquement adoptée pour représenter l'ensemble flou A , bien qu'elle ne fasse référence à aucune idée de sommation ni d'intégration, elle indique pour tout élément x de U son degré $\mu_A(x)$ d'appartenance à A [6-8].

II.4 Fonction d'appartenance d'un ensemble flou (FA)

Soit A un ensemble flou de U , on attribue à chaque élément $x \in U$, une fonction d'appartenance $\mu_A(x)$ dont la valeur varie entre 0 et 1. La forme des FAs est arbitraire, mais il est raisonnable de prendre des fonctions convexes de façon qu'il existe au moins un point de degré d'appartenance maximal et que le degré décroît quand on s'éloigne de ce point. Les FAs les plus utilisées sont [6-7] [9] :

II.4.1 Fonction d'appartenance triangulaire

Elle est définie par trois paramètres $\{a, b, c\}$ qui déterminent les coordonnées des trois sommets (figure II.1).

$$\mu_A(x) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (\text{II.3})$$

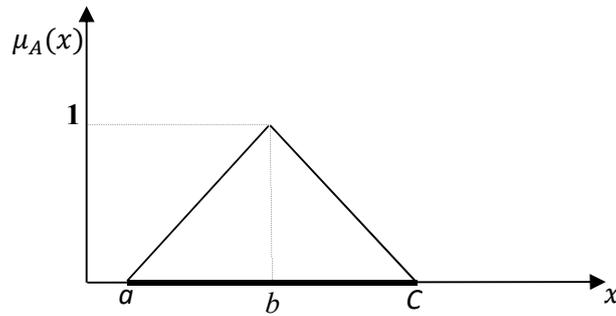


Figure II.1 Fonction triangulaire.

II.4.2 Fonction trapézoïdale

Elle est définie par quatre paramètres $\{a, b, c, d\}$:

$$\mu_A(x) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-b} \right), 0 \right) \quad (\text{II.4})$$

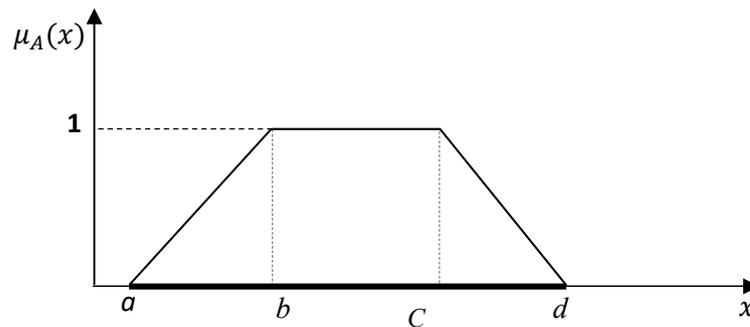


Figure II.2 Fonction trapézoïdale.

II.4.3 Fonction gaussienne

Elle est définie par deux paramètres $\{\delta, m\}$ (Figure II.3) :

$$\mu_A(x) = \text{Exp} \left\{ -\frac{(x-m)^2}{2\delta^2} \right\} \quad (\text{II.5})$$

Où, m et δ représentent respectivement, le centre et l'écart type de la fonction $\mu_A(x)$.

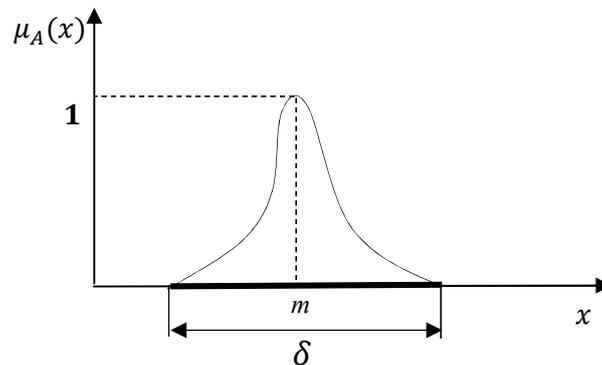


Figure II.3 Fonction gaussienne.

II.4.4 Fonction sigmoïde

Une fonction sigmoïde est définie par deux paramètres $\{a, c\}$:

$$\mu_A = 1 / (1 + \text{Exp}(-a(x - c))) \quad (\text{II.6})$$

Où a permet de contrôler la pente au point d'inflexion $x = c$.

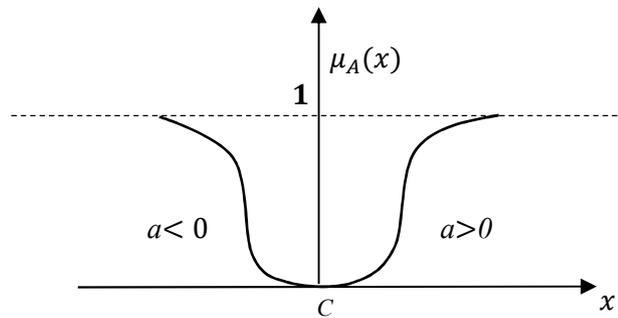


Figure II.4 Fonction sigmoïde

II.5 Variables floues et linguistiques

C'est une variable dont les valeurs ne sont pas des nombres, mais des mots ou phrases exprimés en langage naturel. La raison pour laquelle on utilise cette représentation, est que le caractère linguistique est moins spécifique que le caractère numérique. Une variable linguistique est généralement représentée par un triplet $(x, T(x), U)$ dans lequel :

- x est le nom de la variable linguistique (distance, angle, erreur,...),
- $T(x)$ est l'ensemble des valeurs linguistiques qui sont utilisées pour caractériser x ,
- U est l'univers de discours de la variable linguistique x .

Par exemple, si la distance est considérée comme une variable linguistique définie sur l'univers de discours $U = [0, D_{max}]$, ses valeurs linguistiques peuvent être définies comme suit :

$T(\text{Distance}) = \{\text{Zéros}(Z), \text{Petite}(P), \text{Moyenne}(M), \text{Grande}(G), \text{Très Grande}(TG)\}$. Les symboles linguistiques peuvent être considérés comme des ensembles flous dont les fonctions d'appartenance sont représentées sur la figure II.5.

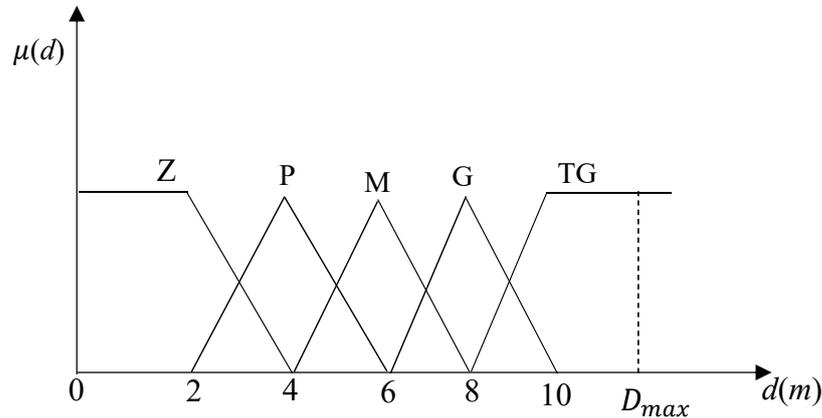


Figure II.5 Représentation de la variable linguistique (distance).

II.6 Opérations sur les ensembles flous

Soient A et B deux ensembles flous définis sur l'univers de discours U , ayant respectivement les fonctions d'appartenance μ_A et μ_B , les opérations de base effectuées sur les ensembles flous sont les suivantes :

II.6.1 L'égalité

Deux ensembles A et B de U sont égaux, si leurs fonctions d'appartenance prennent la même valeur pour tout élément x de U :

$$\forall x \in U \mu_A(x) = \mu_B(x) \quad (\text{II.7})$$

II.6.2 L'inclusion

On dit que A est inclus dans B et on note $A \subseteq B$, si leurs fonctions d'appartenance sont telles que :

$$\forall x \in U \mu_A(x) \leq \mu_B(x) \quad (\text{II.8})$$

II.6.3 L'intersection

L'intersection de deux ensembles flous A et B de U est l'ensemble flou C qui contient tous les éléments x de U appartenant à la fois à A et à B : $C = A \cap B$. Le degré avec lequel chaque élément x de U appartient à C est le plus petit des degrés avec lesquels il appartient à A et B :

$$\forall x \in U \mu_C(x) = \min(\mu_A(x), \mu_B(x)) \quad (\text{II.9})$$

Où min désignant l'opérateur de minimisation.

II.6.4 L'union

L'union de deux ensembles flous A et B , noté $A \cup B$, est l'ensemble flou C qui contient tous les éléments x de U appartenant à A ou bien à B . Le degré avec lequel chaque élément appartient à C est le plus grand des degrés avec lesquels il appartient à A ou à B c'est à dire :

$$\forall x \in U \mu_C(x) = \max(\mu_A(x), \mu_B(x)) \quad (\text{II, 10})$$

Où max désignant l'opérateur de maximisation.

Comme dans la théorie des ensembles classiques, les définitions que nous venons de donner conduisent aux propriétés suivantes pour tout A et B de U :

- Associativité de \cap et de \cup .
- Commutativité de \cap et de \cup .
- $A \cap U = A, A \cap \emptyset = \emptyset$.
- $A \cup \emptyset = A, A \cup U = U$.
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

II.6.5 La Complémentation

Etant donné un ensemble flou A de U , son complément est l'ensemble flou A contenant tous les éléments de U n'appartenant pas à A . La fonction d'appartenance de A est donnée par :

$$\forall x \in U \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (\text{II, 11})$$

II.7 Commande et réglage par la logique floue

Le principe fondamental de la logique floue est de permettre à des informations ambiguës d'être classées dans des ensembles à limites non tranchants, c'est à dire que la forme des fonctions d'appartenance n'est pas rigide et peut dépendre du contexte envisagé, à savoir professionnel, social ou autre, cette construction permet une description des contrôles avec des termes linguistiques vagues, tels que: "large",

"petit", "froid", "rapide", etc. Une des applications de la logique floue est le réglage et la commande des manipulateurs mobiles. Un de ses avantages principaux est de permettre l'incorporation des connaissances d'experts sur le *comment commander le système, sans pour autant connaître la description mathématique du problème*. C'est ainsi que la logique floue se prête particulièrement bien pour des processus mal modélisés. Nous nous sommes intéressés dans cette thèse à résoudre le problème de la commande d'un système robotique à l'aide de la logique floue.

Cet intérêt est dû principalement au succès extraordinaire qu'a connue la logique floue et qui présente les avantages suivants :

- le non nécessité d'une modélisation.
- la possibilité d'implémenter les connaissances d'experts.
- la simplicité dans la réalisation.
- la robustesse face aux perturbations.

II.8 Structure des systèmes à base de la logique floue

Un système à base de la logique floue, ou tout simplement un système flou, est décrit par un ensemble de règles de type *SI* (conditions) *ALORS* (conclusion) permettant de convertir la stratégie de contrôle linguistique acquise auprès d'un expert humain en une stratégie de contrôle automatique bien adaptée au monde réel. Il se compose principalement, comme illustré sur la figure II.6, de quatre blocs essentiels [3] [5][6][8] [10][11] :

- Une interface de fuzzification à l'entrée.
- Une base de connaissances floues.
- Un engin d'inférence ou bloc de prise de décision.
- Une interface de défuzzification à la sortie.

Nous allons rappeler dans ce qui suit une description sommaire de chaque module composant le contrôleur flou [12-13] :

II.8.1 Interface de fuzzification

Ce module traduit les données numériques caractérisant l'état du système pour fournir une caractérisation floue des variables du système flou sous forme symbolique.

II.8.2 La base de règles floues

Une base de connaissance du processus est composée de l'ensemble des renseignements que nous possédons sur le processus. Elle permet de définir les fonctions d'appartenance et les règles floues qui décrivent le comportement du système. C'est le cœur du système entier dans le sens où tous les autres composants sont utilisés pour interpréter et combiner ces règles pour former le système final.

II.8.3 La logique de prise de décision

Un moteur d'inférence floue : une action, sous forme symbolique est décidée à l'aide des techniques de raisonnement flou en fonction des variables floues précédemment calculées.

II.8.4 Interface de défuzzification

Ce module traduit l'action floue issue de l'inférence en une grandeur physique directement applicable au processus à commander.

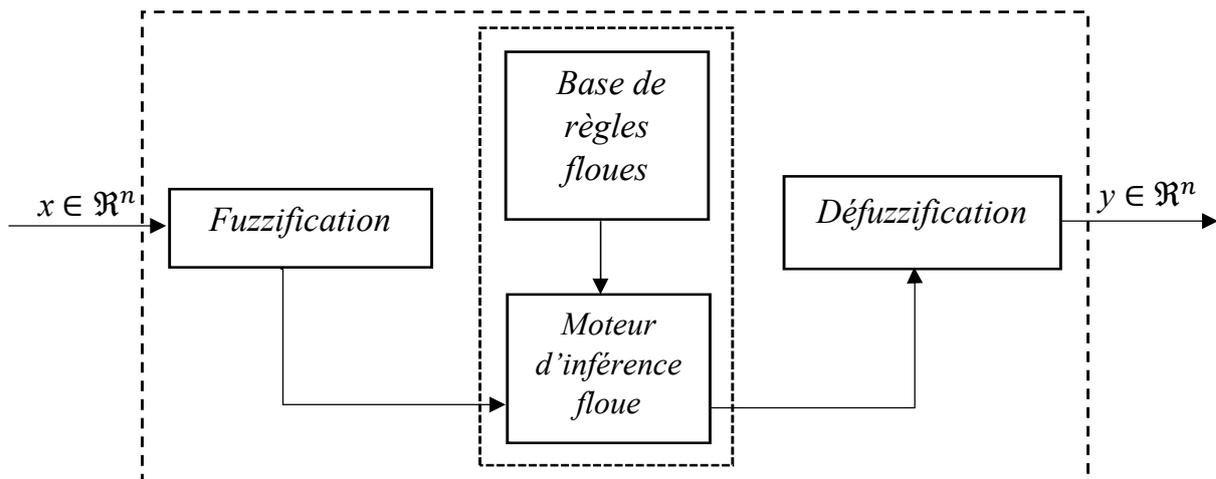


Figure II.6 Structure de base d'un système flou.

Il faut noter qu'il existe un grand nombre de possibilités de réalisation de systèmes flous. En effet, pour chaque module constitutif d'un système flou, il existe une multitude de choix différents, et chaque combinaison de ces choix engendre une classe de systèmes flous.

II.9 Représentation mathématique des systèmes flous

II.9.1 Fuzzification

La fuzzification consiste à relier le point numérique $x_0 = [x_{01}, x_{02}, x_{03}, \dots \dots x_{0n}]^T$ de U de A à l'ensemble flou $A_x = [A_{x1}, A_{x2}, A_{x3}, \dots \dots A_{xn}]^T$ dans $U = U_1 \times \dots \dots \times U_n$ où A_{xi} est un ensemble flou dans U_i . Il existe deux méthodes de fuzzification suivant la définition de A_x .

1. A_x est un singleton flou défini par :

$$\mu_{A_x}(x) = \begin{cases} 1, & x = x_0 \\ 0, & x \neq x_0 \end{cases} \quad (\text{II.12})$$

Dans ce cas, on ne considère que la valeur de x est précise et certaine (figure II.7 (a)).

2. A_x est un ensemble flou de fonction d'appartenance $\mu_{A_x}(x_0)=1$ et $\mu_{A_x}(x)$ décroît lorsque x s'éloigne de x_0 .

Dans ce cas, est pris en compte le comportement de la variable autour de la valeur x_0 . Par exemple, une variable est modélisée par une fonction d'appartenance triangulaire présentée sur la figure II.7 (b).

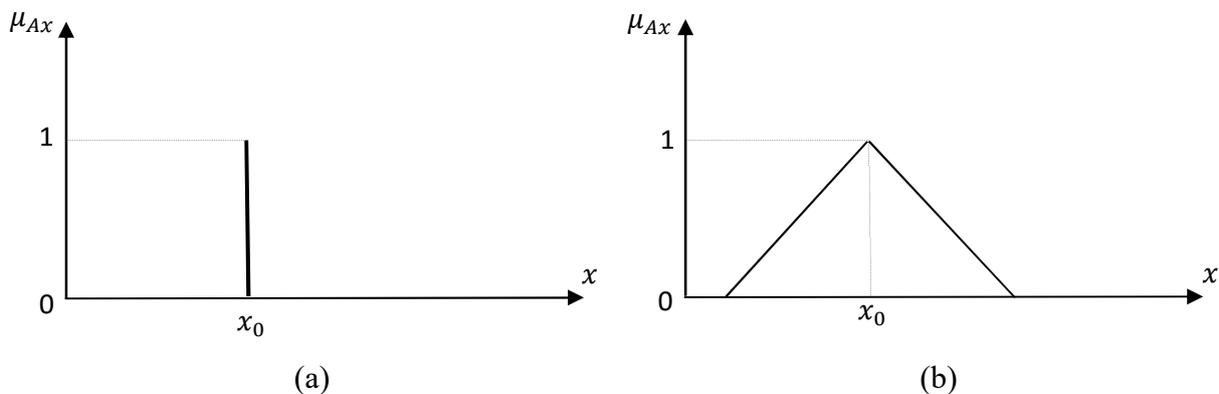


Figure II.7 Méthodes de fuzzification : (a) singleton, (b) ensemble flou.

II.9.2 Base de règles floues

Une base de règles floues R est une collection de règles floues de la forme *SI-ALORS*, $R=[R_1, R_2, R_3, \dots \dots R_m]$. Une règle floue R_i est donnée sous le modèle de «Mamdani» comme suit [14] :

$$R_i : \text{Si } x_1 \text{ est } A_{i1} \text{ et } x_2 \text{ est } A_{i2} \text{ et } \dots \text{ et } x_n \text{ est } A_{in} \text{ ALORS } y \text{ est } B_i \quad (\text{II.13})$$

Le modèle de « Takagi-Sugeno (TS) » sous la forme :

$$R_i : \text{Si } x_1 \text{ est } A_{i1} \text{ et } x_2 \text{ est } A_{i2} \text{ et } \dots \text{ et } x_n \text{ est } A_{in} \text{ ALORS } y \text{ est } f_i(x) \quad (\text{II.14})$$

$f_i(x)$ est un polynôme (hyperplan). Si le polynôme est d'ordre zéro (sous forme d'une constante), on dit que le modèle est de TS d'ordre zéro (TS0), et si le polynôme est du premier ordre, on dit que le modèle est de TS d'ordre un [15].

II.9.3 Moteur d'inférence floue

Le moteur d'inférence floue utilise la base des règles floues pour effectuer une transformation à partir des ensembles flous dans l'espace d'entrée vers les ensembles flous dans l'espace de sortie en se basant sur les opérations de la logique floue. L'antécédent de la règle R , définit un produit cartésien de $A_{i1}, A_{i2}, A_{i3}, \dots \dots A_{in}$ et la règle elle-même R_i , est vue comme une implication. Soit $B_i = A_x \circ R_i$ dans V . La fonction d'appartenance de B_i est définie par la règle compositionnelle :

$$\mu_{B_i}(y) = \sup_{x \in A_x} (\mu_A(x) * \mu_{R_i}(x, y)) \quad (\text{II.15})$$

Dans le jeu de règles du système flou interviennent les opérateurs flous "ET(AND)" et "OU(OR)". L'opérateur "ET" s'applique aux variables à l'intérieur d'une règle, tandis que l'opérateur "OU" lie les différentes règles. Plusieurs types de raisonnement flou ont été proposés dans la littérature suivant la réalisation des opérateurs flous "ET" et "OU" et le type des règles floues utilisées. Les trois moteurs d'inférence floue les plus utilisés sont : le moteur de Mamdani, de Sugeno et celui de Tsukumoto.

II.9.3.1 Méthode de Mamdani

Mamdani fut le premier à utiliser la logique floue pour la synthèse des commandes [16]. Il a utilisé le minimum comme opérateur de conjonction et d'implication. Les

règles correspondant à l'équation (II.13) où B^i est un sous ensemble flou. Généralement, les B^i forment une partition de l'espace de sortie. L'inférence floue correspond aux étapes suivantes, pour un vecteur d'entrée $x = (x_1, \dots, x_n)^t$ [17] :

1. Calcul du degré d'appartenance de chaque entrée aux différents sous-ensembles flous :

$$\mu_{A_j}^i(x_j), \text{ Pour } j = 1 \text{ à } n \text{ et } i = 1 \text{ à } N \quad (\text{II.16})$$

Où : n est la dimension d'espace d'entrée et N : Nombre des règles floues.

2. Calcul de valeur de vérité de chaque règle, pour $i = 1$ à N :

$$\alpha_i(x) = \min_j (\mu_{A_j}^i(x_j)) \quad j = 1 \text{ à } n \quad (\text{II.17})$$

3. Calcul de la contribution de chaque règle selon l'équation :

$$\mu_i(x) = \min (\alpha_i(x), \mu_{B_i}(y)) \quad (\text{II.18})$$

4. Agrégation des règles en utilisant l'opérateur max :

$$\mu(y) = \max_i (\mu_i(y)) \quad (\text{II.19})$$

Le résultat est donc un sous ensemble flou caractérisée par sa fonction d'appartenance. Pour obtenir une conclusion exacte, il faut défuzzifier. Plusieurs méthodes peuvent être utilisées : la méthode de centre de gravité donne :

$$Y = \frac{\int u \mu(u) du}{\int \mu(u) du} \quad (\text{II.20})$$

Dans le cas discret :

$$Y = \frac{\sum_k u_k \mu(u_k)}{\sum_k \mu(u_k)} \quad (\text{II.21})$$

Cette implémentation est appelée (*min, max, barycentre*). Ils existent plusieurs variantes de la méthode de Mamdani comme (min, produit, barycentre) et (produit, somme, barycentre)...etc. [7].

II.9.3.2 Méthode de Takagi-Sugeno

Les SIFs de type Takagi-Sugeno constituent un cas particulier important ; ou la conclusion n'est pas symbolique mais une fonction des entrées : $B^i = f(x_1, \dots, x_n)$, la forme la plus utilisée est la suivante :

$$B^i = \sum_{j=0}^n b_j^i x_j \quad (\text{II.22})$$

Un SIF de type Takagi-Sugeno réalise une fusion de modèles locaux par interpolation. Un SIF est dit d'ordre zéro ou de méthode de Takagi-Sugeno simplifiée si la conclusion est une constante, et SIF d'ordre 1 quand la conclusion est un hyperplan (l'équation II.22). Le modèle de Takagi-Sugeno d'ordre 0 est le plus utilisé. Dont les conclusions sont des nombres réels, ou singletons, (pouvant être considérés comme la valeur modale d'un sous ensemble flou) [15].

Pour le vecteur d'entrée $x = (x_1, \dots, x_n)^t$, la sortie d'un système d'inférence flou de type Takagi-Sugeno d'ordre 0 est calculée selon les étapes suivantes [17] :

1. Calcul du degré d'appartenance de chaque entrée aux différents sous-ensembles flous :

$$\mu_{A_j}^i(x_j), \text{ Pour } j = 1 \text{ à } n \text{ et } i = 1 \text{ à } N \quad (\text{II.23})$$

2. Calcul de la valeur de vérité de chaque règle pour un vecteur d'entrée x , pour $i=1$ à N

$$\alpha_i(x) = ET(\mu_{A_1}^i(x_1), \dots, \mu_{A_n}^i(x_n)) \quad (\text{II.24})$$

3. Calcul de la sortie :

$$Y = \frac{\sum_{i=1}^N \alpha_i(x) \cdot b^i}{\sum_{i=1}^N \alpha_i(x)} \quad (\text{II.25})$$

Les SIFs de type Takagi-Sugeno permettent un passage aisé d'une expression symbolique définie par la base de règle à une valeur numérique exploitable.

II.9.3.3 Méthode de Tsukumoto

Dans ce cas, des fonctions monotoniques sont associées aux variables de sortie. La sortie totale est une moyenne pondérée des degrés de confiance des règles floues et des

valeurs des fonctions des variables de sortie [7]. La figure II.8 illustre les types du raisonnement flou pour un système flou à deux entrées et une base de connaissances de deux règles floues [18]. On constate que les différences viennent de la spécification de la partie conclusion d'une part, et de la méthode de défuzzification d'autre part. Dans cette thèse on limite à l'utilisation des systèmes flous les plus utilisés dans la commande des processus, à savoir les systèmes flous de Takagi-Sugeno à conclusion constante (d'ordre zéro).

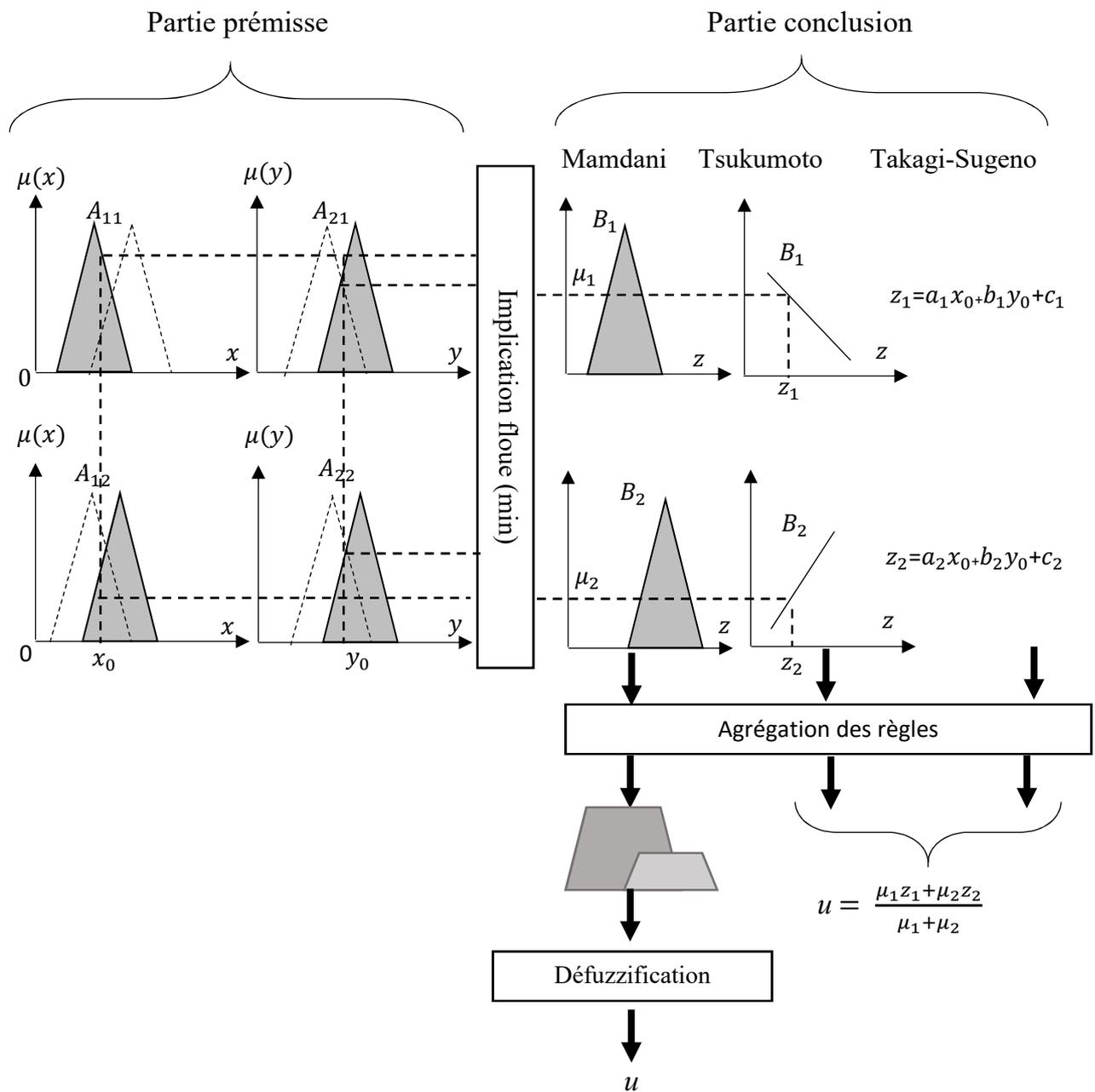


Figure II.8 Différents modèles d'inférences floues.

II.10 Approximation des fonctions par les systèmes flous

Deux raisons principales amènent à utiliser les systèmes flous comme contrôleurs des systèmes dynamiques. Premièrement, ce type de systèmes flous a la propriété d'approximateur universel de fonctions continues avec un degré de précision quelconque à condition d'utiliser un nombre suffisant de règles floues. Deuxièmement, les systèmes flous sont construits à partir de règles floues de la forme *Si-Alors*, de ce fait, les informations linguistiques ou mathématiques disponibles d'un expert peuvent éventuellement être incorporées dans le contrôleur. Dans la littérature consacrée aux systèmes flous, on dispose d'un nombre important de publications montrant que les systèmes flous sont des approximateurs universels (voir par exemple [9] [17][19], c'est-à-dire, pour toute fonction réelle continue f définie sur un compact C de R^n , et pour tout $\varepsilon > 0$, il existe un SIF tel que :

$$(\forall x \in C), (\|f(x) - SIF(x)\| < \varepsilon) \quad (II.26)$$

Notons, cependant, que la propriété d'approximation universelle ne donne pas une méthode de construction du système flou $SIF(x)$, mais elle garantit seulement son existence. De plus, pour un degré de précision quelconque, il faut utiliser un nombre important de règles floues.

II.11 Conclusion

Dans ce chapitre, nous avons introduit les principes de base des ensembles flous qui sont une généralisation du concept d'ensembles classiques. Contrairement à la fonction caractéristique d'un ensemble net qui prend la valeur 0 ou 1, la fonction d'appartenance d'un ensemble flou prend ses valeurs dans l'intervalle $[0,1]$. A partir des ensembles flous, nous pouvons construire des systèmes flous. Un système d'inférence flou est une unité de prise de décision composée de quatre parties essentielles : la fuzzification, la base de règles, le moteur d'inférence et la défuzzification. L'architecture d'un système flou est déterminée par une meilleure compréhension des ensembles flous et des opérateurs flous. Nous avons constaté qu'il n'existe pas un seul type de système flou, mais il y en a plusieurs. Un utilisateur des systèmes flous doit décider sur la méthode de fuzzification, le type des fonctions d'appartenance, le type des règles floues, la méthode

du raisonnement flou et la stratégie de défuzzification. Les systèmes flous sont des approximateurs universels. En fait, ils peuvent approcher n'importe quelle fonction à partir de données numériques. Le fonctionnement d'un contrôleur flou dépend d'un nombre important de paramètres (méthode de fuzzification, le type des fonctions d'appartenance, le type des règles floues, la méthode du raisonnement flou et la stratégie de défuzzification) qu'il faut déterminer lors de la conception. Comme ces paramètres s'influencent mutuellement, leur réglage n'est donc pas aisé. Par contre, les contrôleurs flous présentent la possibilité d'incorporer des connaissances expertes dans leurs structures, ce qui peut aider à la recherche des paramètres optimaux des contrôleurs flous en utilisant d'autres approches.

Dans le chapitre 3, nous exposerons la méthode de l'intelligence artificielle utilisée dans ce travail; les réseaux de neurones artificiels. Ces méthodes permettent l'ajustement des paramètres des règles floues et l'optimisation des contrôleurs flous d'une façon très efficace.

REFERENCES BIBLIOGRAPHIQUES CHAPITRE II

- [1] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, vol.8, pp 29-44, 1965.
- [2] P. J .King, E.H. Mamdani, "The application of fuzzy control system to industrial process", *Automatica* 13, 235-242, 1976.
- [3] H. Abdelouahab, "Calcul de trajectoire et contrôle de déplacement d'un robot ". Thèse de doctorat, université Ferhat Abbas Setif, 2007.
- [4] F. Abdessamed, E. Monacelli, K. Benmahammed: "A learning paradigm for motion control of mobile manipulators". *AMCS, Int. J. Appl. Math. Comput. Sci*, Vol. 16, No 4, pp 475-484, 2006.
- [5] B. B. Meunier, "La logique floue et ses applications", Edition Addison Wesley France, SA, 1995.
- [6] H. Bühler, "Réglage par Logique Floue", Première édition, Presses Polytechniques et Universitaires, Romandes, 1994.
- [7] K. M. Passino, S. Yurkovich, "Fuzzy Control", Addison Wesley Longman, 1998.
- [8] J. Godjevac, "Idées Nettes sur la Logique Floue", Collection Informatique, Presses Polytechniques et Universitaires Romandes, Lausanne, 1999.
- [9] L.X. Wang, "Adaptive Fuzzy Systems and Control: Design and Stability Analysis", Prentice-Hall, Englewood Clifs, NJ, 1994.
- [10] R. R. Yager, D. P. Filev, "Essential of Fuzzy Modeling and Control", John Wily & Sons Inc., 1994.
- [11] J. M. Mendel, "Tutorial: Fuzzy Logic Systems for Engineering", *Proceeding of the IEEE*, vol. 83, no. 3, pp. 345-377, March 1995.
- [12] B. Kosko, "Fuzzy systems as Universal Approximators", *IEEE transactions on computers*, 1993.
- [13] J. Jantzen, "Foundations of Fuzzy Control", West Sussex (England), John Wiley & Sons Ltd, 2007.
- [14] E. H. Mamdani, S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1- 13, 1975.
- [15] T. Takagi, M. Sugeno, "Fuzzy Identification of Systems and its Applications to Modeling and Control", *IEEE Transactions on Systems, Man, Cyber*, vol. 15, pp. 11 - 132, 1985.
- [16] E.H. Mamdani, "Application of Fuzzy Algorithms for Control of Simple Dynamic Plant", *Proc. of IEEE*, vol. 121, no. 12, pp. 1585-1588, 1974.
- [17] P. Y. Glorennec, "Algorithmes d'Apprentissage pour Systèmes d'inférence Floue", Edition Hermès, 1999.
- [18] H. Boubertakh, "Contribution à l'Optimisation par Algorithmes Evolutionnaires des Contrôleurs Flous", Thèse de Doctorat en Automatique, ENP, 2009.

- [19] B. Kosko, "Fuzzy Systems as Universal Approximators", IEEE Transactions on Computers, 1993.

Chapitre III

Les Réseaux de Neurones Artificiels et les Systèmes Hybrides Neuro-Flou

III.1 Introduction

Les recherches en réseaux de neurone remontent à la fin du 19^{ème} siècle ; et consistent en premier lieu aux travaux multidisciplinaires en physique, en psychologie et en neurophysiologie, cela par des scientifiques tels Hermann Von Helmholtz, Ernst Mach, Ivan Pavlov et bien d'autres. A cette époque, il s'agissait de théorie plutôt générale sans modèle mathématique précis d'un neurone. C'est qu'en 1943 que Warren McCulloch et Walter Pitts proposent une structure d'un neurone formel inspiré du neurone biologique du système nerveux. Ce neurone formel doté d'une représentation mathématique peut réaliser des fonctions logiques et arithmétiques.

En 1949 Donald Hebb vient renforcer le neurone en présentant dans son ouvrage (The Organization of Behavior) une règle d'apprentissage des paramètres du neurone. Frank Rosenblatt développe en 1957 le Perceptron inspiré du système visuel possédant deux couches, une de perception et une de décision. On peut dire que c'est le premier système artificiel capable à apprendre par expérience. Environ en même moment Bernard Widrow et Ted Hoff ont proposé leur modèle Adaline avec une règle d'apprentissage qui servira par la suite aux réseaux multicouches [1]. Les auteurs Marvin Minsky et Seymour Papert écrivent en 1969 une critique sur les limitations du Perceptron, et la

généralise sur les réseaux de neurones artificiels. Cette critique est venue jeter beaucoup d'ombre sur le domaine des réseaux de neurones.

Malgré ces critiques certains chercheurs n'ont pas hésités a continué de travailler sur les réseaux de neurones. Parmi eux Teuvo Kohonen et James Anderson, qui ont développé, en 1972 et en même temps et indépendamment, des réseaux qui peuvent servir de mémoire associative, de même Stephen Grossberg a investigué ce qu'on appelle les réseaux auto-organisés.

En 1982 J.J.Hopfield, physicien, propose un réseau complètement rebouclé, mais lui fixe un objectif de même une règle d'apprentissage ce qui a donné aux chercheurs un nouveau souffle. L'année 1985 a fait tout basculer au profit des réseaux de neurones suite aux travaux de David Rumelhart et James Mclelland concernant l'élaboration de la rétro-propagation des erreurs comme méthode d'apprentissage des paramètres des réseaux de neurones multicouches. A partir de ce moment, la recherche sur les réseaux de neurones a connue plusieurs applications aux cours des années.

Les réseaux de neurones servent aujourd'hui à toutes sortes d'applications dans divers domaines tel qu'autopilotage d'avion, guidage d'automobile, lecture de document bancaire et postale, traitement de signal, traitement de la parole, traitement de l'image, prévision sur les marchés monétaires, génération de trajectoire pour les robots manipulateurs et mobiles, et il y a fort à parier que leur importance ira en grandissant dans le futur.

Dans ce chapitre, nous allons introduire deux techniques importantes de l'intelligence artificielle pour l'adaptation des comportements ; les réseaux de neurones artificiels avec ces capacités d'apprentissages, et le deuxième est les systèmes hybrides neuro-flou de type (ANFIS). Grâce aux plusieurs caractéristiques, l'utilisation de ces deux approches pour l'amélioration des comportements et la commande des processus devient un domaine très important dans les dernières décennies [2-7][23]. Les définitions et les notions de base seront présentées dans ce chapitre.

III.2 Définitions

III.2.1 Intelligence Artificielle (IA)

L'intelligence artificielle est la reproduction, par des moyens artificiels de toutes les formes de l'intelligence humaine pour un objectif final, qui s'intéresse à la conception des systèmes intelligents. L'IA est la capacité à raisonner, à apprendre, et à s'adapter face aux de nouveaux changements qu'ils sont des éléments principaux que nous recherchons [8].

III.2.2 Agent

D'après Ferber [8] ; un agent est une entité autonome réelle ou abstraite qui est capable d'agir sur elle-même et sur son environnement, et peut communiquer avec d'autre agents, dont le comportement est la conséquence de ses observations, ses connaissances et de ses interactions avec son environnement. Il doit être (autonome, interactif, adaptatif, rationnel, coopératif et intelligent...). La notion d'agent se diffère selon l'utilisation.

III.2.3 Apprentissage

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques. L'objectif est d'extraire et exploiter automatiquement l'information présentée dans un jeu de données. On peut schématiser le processus d'apprentissage comme montré sur la figure III.1, l'environnement agit sur le module d'apprentissage de l'agent. Cela a pour effet de consulter et de modifier la base de connaissances pour arriver à une exécution adéquate. De cette exécution, un retour est attendu pour évaluer le résultat obtenu.

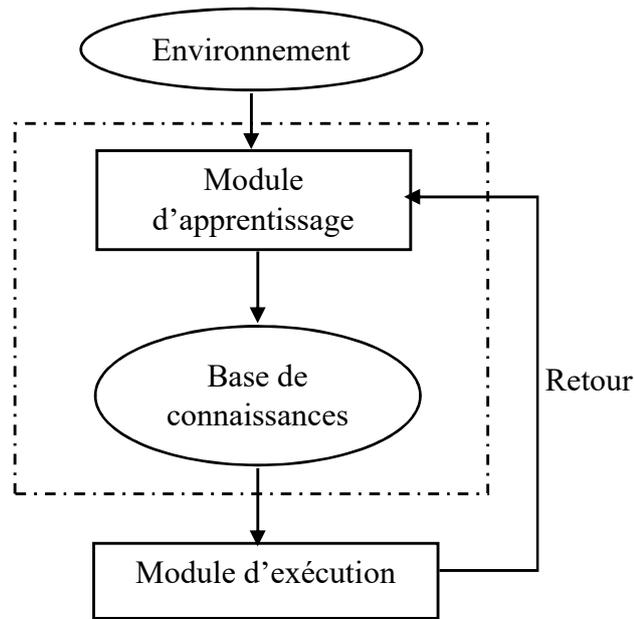


Figure III.1 Représentation de l'apprentissage automatique.

Les algorithmes d'apprentissage peuvent se catégoriser selon le type d'apprentissage, qu'ils emploient [9] :

Apprentissage supervisé : le contrôleur (ou le maître) fournit l'action qui devrait être exécutée. L'utilisation nécessite un expert capable de fournir un ensemble d'exemples formés de situations et d'actions correctes associées (figure III.2) [9]. Ces exemples doivent être représentatifs de la tâche à accomplir.

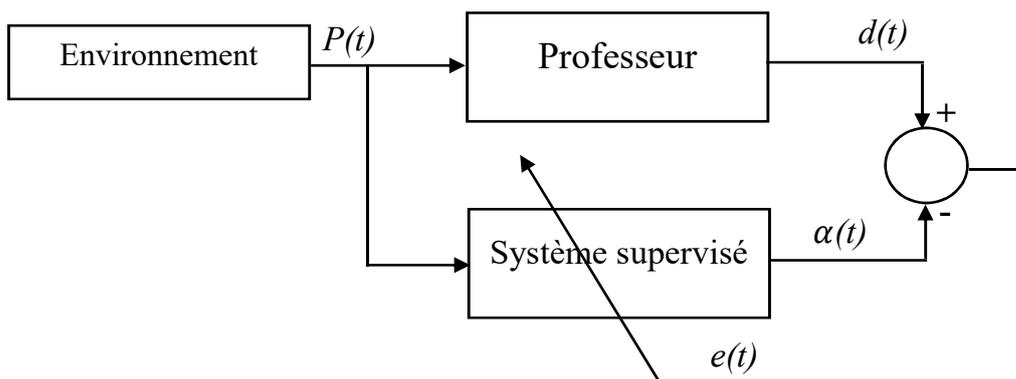


Figure III.2 Représentation de l'apprentissage supervisé.

Apprentissage non supervisé : dans lequel l'apprenant doit identifier par lui-même la meilleure réponse possible, il n'y a pas de réponse désirée. La tâche peut être par

exemple dans ce cas de créer des regroupements de données selon des propriétés communes (catégorisation) [10].

Apprentissage par renforcement (AR) : le contrôleur a un rôle d'évaluateur et non pas d'instructeur, l'information disponible est un signal de renforcement ; généralement appelé critique (figure III.3). Son rôle est de fournir une mesure indiquant si l'action générée est appropriée ou non. Le contrôleur doit déterminer et de modifier ses actions de manière à obtenir une meilleure évaluation dans le futur [5].

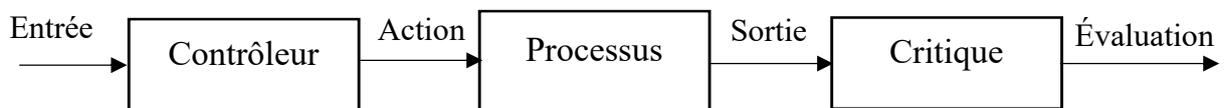


Figure III.3 Représentation de l'apprentissage par renforcement.

Dans ce qui suit, on va donner un bref exposé sur les réseaux de neurones artificiels. En présentant leurs principes et les différentes architectures existantes.

III.3 Réseaux de neurones artificiels (RNA)

Le réseau de neurones artificiels est une collection de simples processeurs mathématiques reliés ensemble. Chaque processeur ne peut effectuer qu'une seule tâche mathématique primitive. Mais un réseau complexe, possède de grandes capacités et peut faire beaucoup de choses. L'inspiration derrière les réseaux de neurones artificiels est le cerveau. Il est constitué d'environ 100 milliards d'unités de traitement reliés entre eux de manière à former un réseau. Ces unités processeurs sont appelées « cellules cérébrales » ou « Neurones », chacune d'elle est une cellule vivante.

III.3.1 Structure d'un neurone

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau (figure III.4) [9-10]. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone. L'information traitée par le neurone s'oriente ensuite le long de l'axone pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire entre l'axone du neurone afférent et

les dendrites du neurone efférent. La jonction entre deux neurones est appelée la synapse.

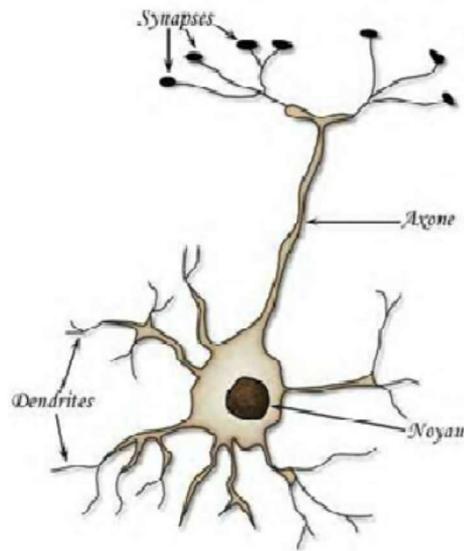


Figure III.4 Représentation d'un neurone biologique.

Par analogie avec le neurone biologique, le comportement du neurone artificiel est modélisé par deux phases comme présenté sur la figure III.5 :

- Un opérateur de sommation, qui élabore le potentiel α , Cet opérateur effectue la somme pondérée des entrées. On soustrait parfois à cette somme la valeur de seuil d'activation.
- Un opérateur non linéaire qui calcule la valeur de l'activation en utilisant une fonction de transfert (fonction d'activation).

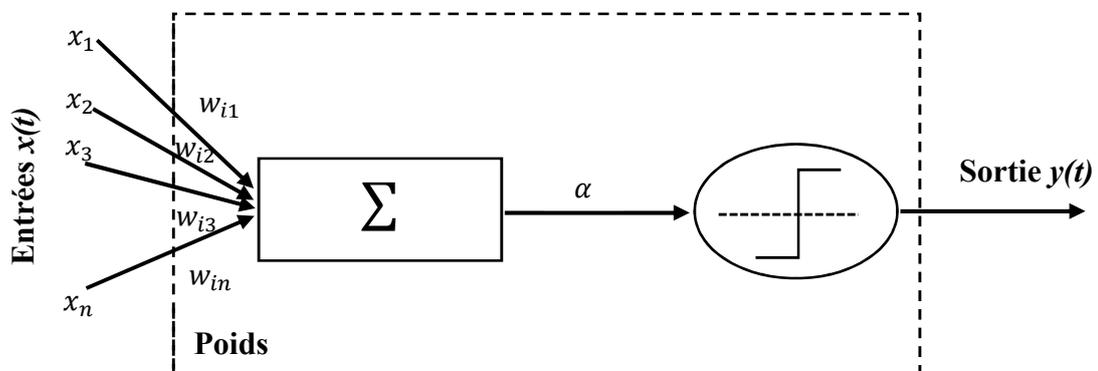


Figure III.5 Modèle d'un neurone artificiel.

Chaque neurone artificiel est un processeur élémentaire de traitement. Il reçoit un nombre variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associé un poids w_{ij} représentant la force de la connexion [10-11]. Le neurone artificiel modélisé par Mc Culloch et Pitts est représenté par la figure III.5.

x_i : Entrée du neurone i (ou sortie de neurone amont j).

w_{ij} : La valeur du poids synaptique de la connexion dirigée du neurone j vers le neurone i ,

- Un poids positif indique un effet excitateur du neurone émetteur j vers le neurone récepteur i ,
- Un poids négatif indique un effet inhibiteur.

La fonction de combinaison renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids synaptiques. Autrement dit, elle calcule la somme pondérée des entrées selon l'expression suivante :

$$\alpha_i = \sum_{j=1,n} w_{ij} \cdot x_j \quad (\text{III.1})$$

À partir de cette valeur, une fonction d'activation f , calcule la valeur de l'état du neurone. Cette valeur sera transmise aux neurones aval. Les neurones les plus fréquemment utilisés sont ceux pour lesquels la fonction f est une fonction non linéaire d'une combinaison linéaire des entrées.

$$y_i = f(\alpha_i - \theta_i) \quad (\text{III.2})$$

III.3.2 Fonctions d'activation

La fonction de transfert ou d'activation définit la valeur de sortie d'un neurone en termes des niveaux d'activité de ses entrées. Cette fonction peut prendre différentes formes possibles telles que : fonction linéaire à seuil, fonction seuil, fonction gaussienne,...etc (figure III.6).

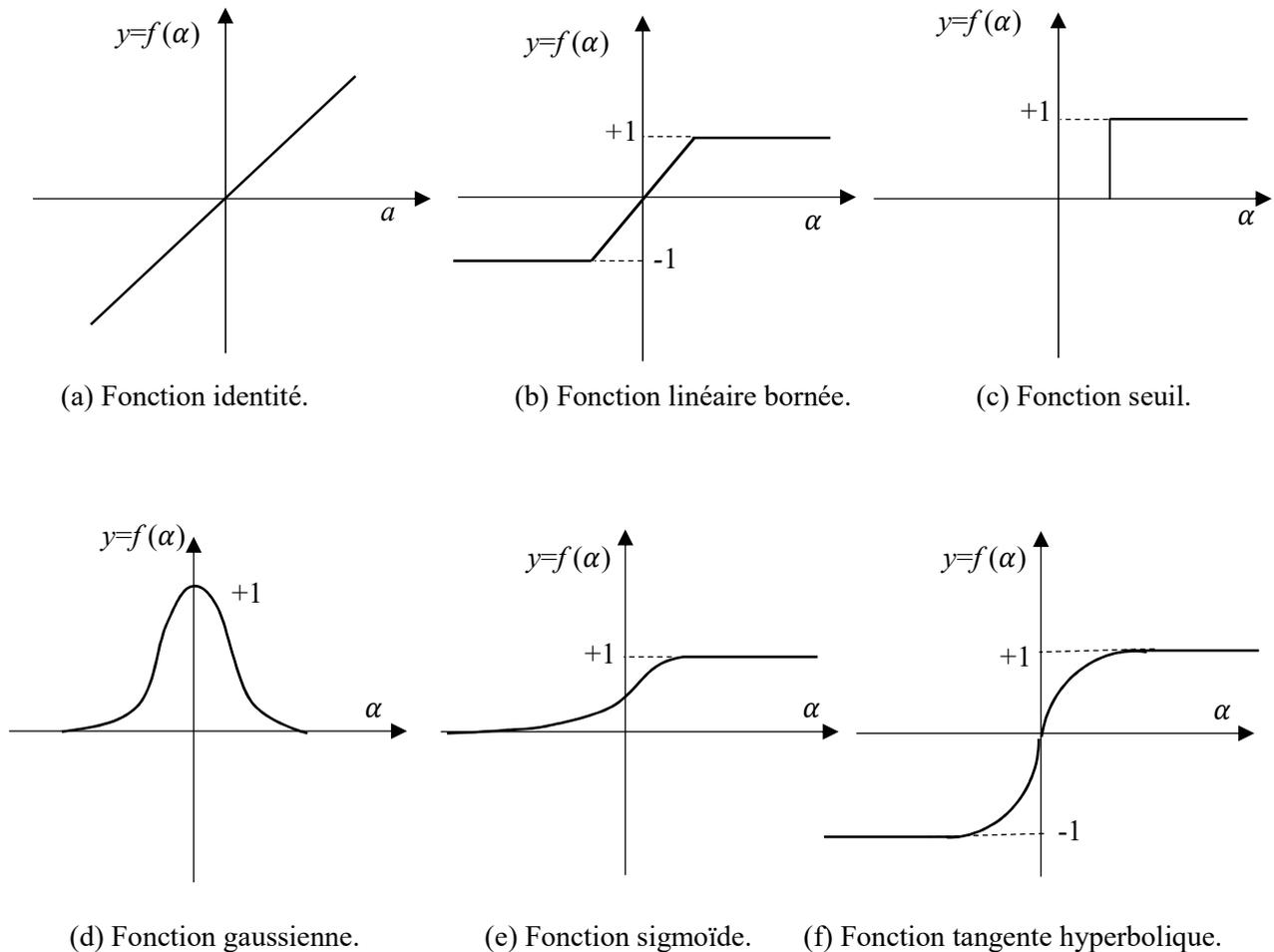


Figure III.6 Formes usuelles de la fonction d'activation.

III.3.3 Structures d'interconnexion

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité qui différencie entre ces réseaux [12-14]. On peut citer :

III.3.3.1 Réseau multicouche (Multi Layer Perceptron)

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches en aval (Figure III.7 (a)). Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble

des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées.

III.3.3.2 Réseau à connexions récurrentes

Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locales (Figure III. 7 (b)).

III.3.3.3 Réseau à connexions locales

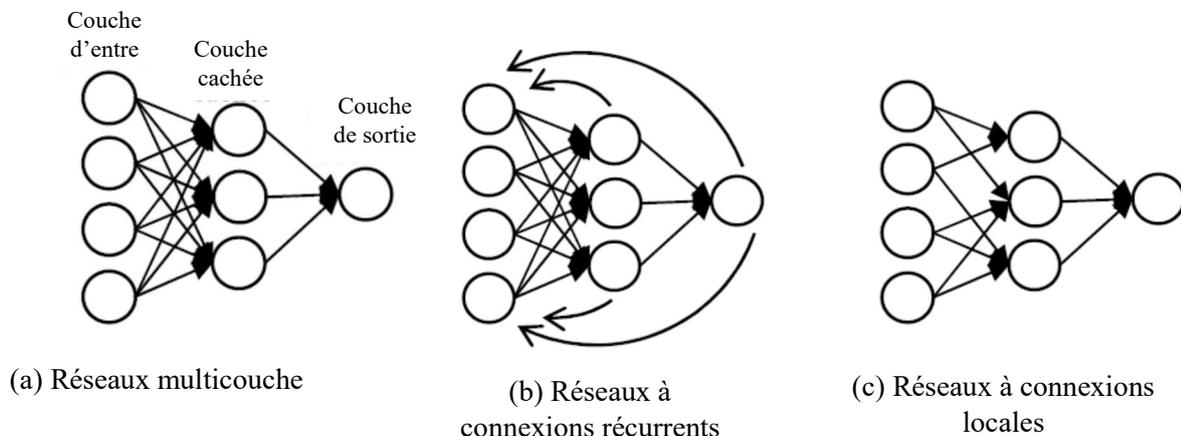
Il s'agit d'une structure multicouche, mais qui à l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale (Figure III. 7 (c)). Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

III.3.3.4 Réseaux à fonction radiale

Ce sont les réseaux que l'on nomme aussi RBF (Radial Basic Fonctions). L'architecture est la même que pour les PMC (perceptron multicouches) mais avec une seule couche cachée. Cependant, les fonctions de base utilisées ici sont des fonctions gaussiennes (Figure III.7 (d)).

III.3.3.5 Réseau à connexion complète

C'est la structure d'interconnexion la plus générale (Figure III.7 (e)). Chaque neurone est connecté à tous les neurones du réseau et à lui-même.



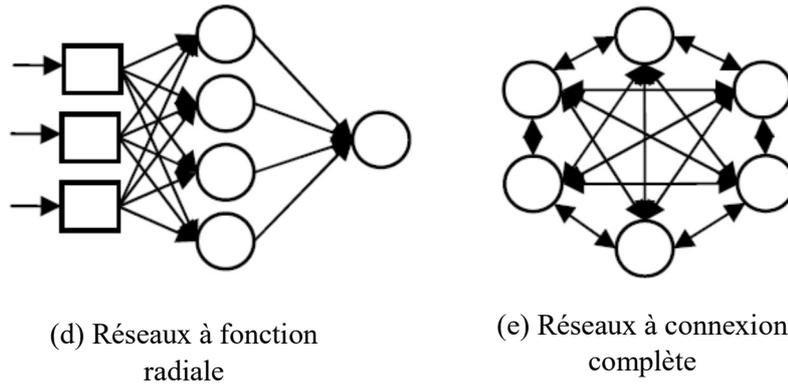


Figure III.7 Les différentes structures d'interconnexion des RNA.

III.3.4 Apprentissage des RNA

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux. L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré en modifiant les poids d'interconnexion [13] [15]. Celui-ci peut s'effectuer par présentation répétée d'une série de "patrons" ou "modèle" et peut être supervisé ou non. Dans le cas des RN multicouches, les approches les plus utilisées sont celles inspirées de la méthode de rétro-propagation du gradient. Les techniques d'apprentissage sont regroupées en deux classes ; supervisé et non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage [10] [14] :

III.3.4.1 Apprentissage supervisé

Ces algorithmes utilisent une base de données composée d'un ensemble de paires entrée-sortie. L'apprentissage dans ce cas, consiste à calculer les poids d'interconnexion de telle manière que les sorties du réseau de neurones soient, pour les exemples utilisés lors de l'apprentissage, aussi proches que possibles des sorties désirées, qui peuvent être la valeur de la fonction approchée ou de la sortie du processus à modéliser, ou encore la sortie souhaitée du processus à commander.

III.3.4.2 Apprentissage non supervisé

Le calcul des coefficients dans ce cas, se fait sur la base de la suite des vecteurs d'entrés en optimisant une fonction de cout. Les cartes autoorganisatrices de Kohonen est un modèle de référence pour l'apprentissage non supervisé. Pour l'apprentissage des RNAs multicouches, l'algorithme de la rétro propagation du gradient de l'erreur "*backpropagation*" est le plus utilisé. Le principe consiste à la modification des poids des connexions en minimisant une fonction de cout (erreur quadratique) [16] [15] [12]. Il existe plusieurs règles de modification des poids dont les principales sont : la règle de Hebb, la règle de Widrow-Hoff.

III.4 Les RNA pour la commande des processus

Les réseaux de neurones artificiels constituent des outils particulièrement efficaces pour la modélisation et la commande des systèmes complexes ou mal définis. Commander un processus, c'est déterminer l'action à appliquer, pour lui assurer un comportement donné, en dépit de perturbations. Cette action est délivrée par un organe ou une loi de commande réalisée par un RNA. Nous séparons ces approches en deux classes principales suivant qu'elles nécessitent ou non l'identification préalable d'un modèle du processus commandé. Nous parlerons de méthode de commande *indirecte* lorsqu'un tel modèle est nécessaire, par opposition aux méthodes *directes* qui n'en nécessitent pas [17-19].

III.4.1 Identification par RNA

L'idée de base est de substituer aux modèles paramétriques classiques des modèles neuroneaux dont les paramètres sont adaptés par une procédure d'apprentissage appropriée en utilisant les données entrées-sorties du système.

III.4.1.1 Modélisation directe

Consiste à entraîner le réseau pour représenter la dynamique directe du système où l'erreur entre les deux sorties est utilisée pour l'adaptation des paramètres du modèle. Les réseaux de neurones récurrents sont les plus adaptés pour ce type de modélisation :

- **Modélisation série-parallèle** : le but est d'entraîner le réseau pour que la transformation non linéaire f soit une approximation de la fonction du système. Dans cette structure, l'entrée du modèle comporte la sortie actuelle et les valeurs précédentes de la réponse du système réel.

- **Modélisation parallèle** : l'entrée du réseau comporte les valeurs décalées de la réponse du modèle. L'avantage est de présenter des performances meilleures dans le cas de perturbations.

III.4.1.2 Modélisation inverse

Consiste à entraîner un modèle neuronal pour qu'il génère les commandes nécessaires de déplacer le système de son état actuel $x(t)$ à un état désiré $x_d(t+1)$. Le but de la procédure d'entraînement est l'adaptation des paramètres du modèle pour qu'il approche la fonction inverse f^{-1} (évaluer l'erreur sur la commande qui sera ensuite utilisée pour l'apprentissage) :

- **Apprentissage généralisé** : en utilisant la connaissance qualitative que l'on a sur le comportement du système, des signaux compatibles avec son fonctionnement lui sont appliqués. La sortie du système sert alors d'entrée au modèle neuronal dont la sortie est comparée avec le signal d'entraînement. En utilisant cette erreur pour l'apprentissage du réseau, cette structure force le réseau à apprendre le comportement inverse du système.

- **Apprentissage spécialisé** : dans cette approche, le modèle inverse neuronal est placé en série avec le système, et reçoit à son entrée un signal d'entraînement appartenant à l'espace des sorties opérationnelles désirées du système. Le système est considéré comme une couche supplémentaire dont les poids sont fixes. Le but de la procédure d'apprentissage est de minimiser l'erreur sur les sorties du modèle inverse.

III.4.2 Commande par RNA

Deux architectures de contrôle contenant des réseaux de neurones sont possibles. Le premier type est une configuration auto-réglable, dans laquelle un réseau ajuste les paramètres d'un contrôleur conventionnel. Le second type est une configuration plus simple, dans laquelle le réseau commande directement le système. Les structures de contrôle peuvent être divisées en cinq structures de base [18] :

- **Contrôle supervisé** : dans ce cas, les entrées du réseau sont les informations que reçoit l'opérateur humain sur l'état du système, les sorties désirées sont les actions délivrées par l'opérateur.
- **Contrôle direct par modèle inverse** : ce type de contrôle utilise un modèle inverse placé en série avec le système à commander. Le modèle inverse est capable de générer les commandes nécessaires pour une tâche désirée.
- **Contrôle adaptatif neuronal** : le réseau est utilisé pour remplacer les transformations linéaires adoptées par les techniques standard de commande adaptative (à modèle de référence, contrôleurs auto-réglables,...).
- **Rétro-propagation d'utilité** : le principe est de calculer les variations par rétro-propagation à travers le temps. Le comportement désiré du système sur un horizon déterminé est spécifié par un critère de performance (fonction d'utilité) qui doit être défini en fonction des sorties du système et des sorties désirées. Le contrôleur neuronal est entraîné de façon à minimiser ce critère.
- **Contrôle à apprentissage par renforcement** : cette structure de commande comporte un réseau d'évaluation de performance du système (sous forme des récompenses et des punitions), et un réseau d'action pour générer les commandes qui optimisent le critère de performance.

Les RNA possèdent plusieurs caractéristiques intéressantes pour la réalisation des systèmes de commande parmi lesquelles [13] [15] [19] :

- la parallélisation du traitement leur confère une grande rapidité de calcul et les rend très adaptés aux applications temps réel. Ceci est d'autant plus vrai lorsque l'on s'intéresse à des réalisations réellement parallèles sur des machines multiprocesseurs,
- le caractère distribué et fortement redondant du traitement réalisé leur donne une bonne résistance aux pannes internes,
- leur capacité de généralisation garantit une bonne résistance aux bruits. Ceci est particulièrement important lorsque les capteurs permettant de mesurer l'état ou la sortie du processus commandé sont soumis à des perturbations. C'est notamment le cas

en robotique, où l'une des grandes difficultés concerne l'obtention de données sensorielles de bonne qualité.

- Faculté d'apprentissage à partir d'exemples représentatifs, par "rétro-propagation de l'erreur". L'apprentissage ou la construction du modèle est automatique,
- La simplicité d'utilisation.

Mais entre outre, les réseaux de neurones présentent quelques inconvénients :

- L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans les couches,
- Le choix des valeurs initiales des poids du réseau et le réglage des paramètres jouent un rôle important pour la convergence,
- Le problème du sur-apprentissage ou les minimums locaux qui limitent l'utilisation,
- La difficulté d'obtenir une base de données pour l'apprentissage.

III.5 L'approche neuro-floue

III.5.1 Introduction

L'utilisation conjointe des réseaux de neurones et de la logique floue permet de tirer les avantages des deux méthodes ; les capacités d'apprentissage de la première et la lisibilité et la souplesse de la seconde. Diverses combinaisons de ces deux méthodes ont été développées depuis 1988. Elles ont donné naissance aux systèmes neuro-flous, qui sont le plus souvent orientés vers la commande de systèmes complexes et les problèmes de classification [20].

III.5.2 Motivations pour une approche hybride

Les dernières années ont vu se développer de nombreux travaux sur l'étude conjointe des réseaux de neurones et de la logique floue. Les raisons de cet engouement sont sans doute à rechercher dans les nombreuses similarités qui existent entre les deux approches. Tout d'abord les deux techniques présentent des propriétés d'approximateurs universels de fonction. En effet il a été montré qu'il est possible d'approcher n'importe quelle fonction continue à l'aide d'un système flou ou d'un réseau neuronal classique à couches. On peut aussi trouver des similitudes au niveau de la structure. Il existe en

premier lieu une analogie entre les données des règles (*SI-ALORS*) de la logique floue et les couples (entrée-sortie) des réseaux de neurones. De même on peut trouver un point commun entre les fonctions réalisées par les neurones formels et les fonctions d'appartenance caractérisant les différentes variables d'un système flou. Enfin on peut établir une corrélation entre les opérations de multiplication et d'addition induites par la structure neuronale et celles de maximisation et de minimisation correspondant aux conjonctions et aux disjonctions des règles [21] [22].

Cependant la logique floue et les réseaux de neurones possèdent tous deux des points forts et des points faibles différents :

	<i>LA LOGIQUE FLOUE</i>	<i>LES RESEAUX DE NEURONE</i>
<i>AVANTAGES</i>	<ul style="list-style-type: none"> - Intégration de la connaissance à priori. - Facilité de construction. - Possibilité d'interpréter la connaissance. 	<ul style="list-style-type: none"> - Capacités d'apprentissage - Parallélisme massif - Résistance aux données bruitées.
<i>INCONVENIENTS</i>	<ul style="list-style-type: none"> - Construction manuelle des règles. - Difficultés d'optimisation des différents paramètres. 	<ul style="list-style-type: none"> - Aucune interprétation possible des résultats de l'apprentissage - Difficulté d'estimer les paramètres l'apprentissage

Tableau III.1 Comparaison entre la logique floue et les réseaux de neurones.

III.5.3 Systèmes hybrides neuro-flous

Les réseaux de neurones artificiels et la logique floue peuvent être complémentaires sur plusieurs points. La logique floue permet une spécification rapide des tâches à accomplir à partir de la connaissance symbolique disponible. Le réglage précis du système obtenu et l'optimisation de ses différents paramètres reste néanmoins beaucoup plus difficile dans de nombreux cas. Les modèles les plus courants de RNA, au contraire,

n'autorisent pas l'incorporation de connaissance à priori mais permettent de régler par apprentissage le comportement précis du système.

De nombreux auteurs ont donc tout naturellement cherché à combiner ces deux paradigmes depuis le début des années quatre-vingt et ceci de plusieurs manières.

Nous portons ici notre attention sur les approches permettant de représenter sous forme de RNA les règles d'un système d'inférence flou. Cette technique consiste à utiliser des fonctions d'activation particulières pour les unités du réseau, ainsi qu'une organisation spécifique de ces dernières et ce afin de reproduire les différents éléments constitutifs d'un système d'inférence flou.

La structure du réseau est ainsi choisie en fonction de la forme de la fonction que l'on cherche à approcher. Les dépendances entre les variables d'entrée et de sortie sont en effet spécifiées par le choix des règles floues. Les différents paramètres de ces règles (c à d forme et position des sous-ensembles flous, sortie et poids des règles), peuvent ensuite être modifiés par un algorithme d'apprentissage supervisé. Nous pouvons noter que cela ne restreint en rien l'ensemble des fonctions approchables puisqu'il a été démontré que la logique floue présente comme les RNA des propriétés d'approximation universelle de fonction.

Si l'on adopte le point de vue des systèmes flous, cette approche permet de régler de manière précise, par apprentissage, le comportement du système réalisé. Si l'on se place au contraire du côté des RNA, la connaissance sous forme de règles floues permet de choisir l'architecture du réseau en fonction de la tâche à accomplir [21-22].

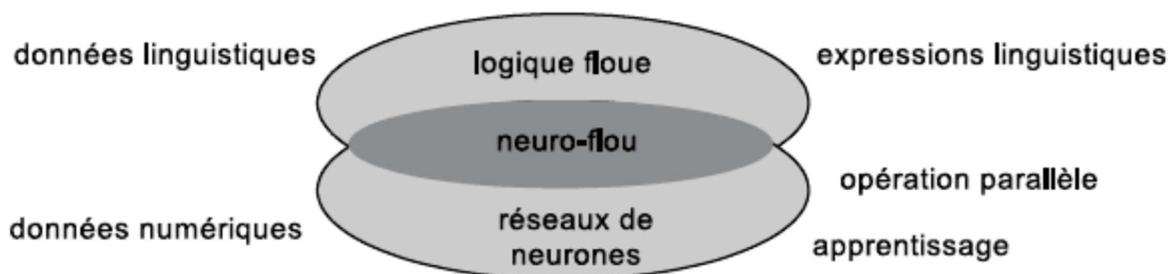


Figure III.8 Principe du système neuro-flou.

La figure III.8 résume le principe du système neuro-flou qui représente l'intersection entre la logique floue et les réseaux de neurones.

Un système neuro-flou est donc défini comme étant un réseau neuronal multicouche avec des paramètres flous, ou comme un système flou mis en application sous une forme distribuée parallèle. Il a été notamment répertorié trois familles différentes [20] :

III.5.3.1 Le modèle *FALCON* et le modèle *GARIC*

Ces deux structures à cinq couches utilisent la fuzzification en entrée et la défuzzification en sortie, correspondant à l'interprétation juste de la technique de Mamdani. Ce sont des structures très précises mais très lentes à l'exécution. Ces modèles sont généralement utilisés pour la commande.

III.5.3.2 Le modèle *NEFCLASS*

Modèle utilisé principalement en classification, il est constitué de 3 couches ; une couche d'entrée avec des fonctions d'appartenance, une couche cachée représentée par des règles et une couche de sortie définissant les classes. Modèle facile à mettre en application car il évite l'étape de défuzzification tout en étant précis.

III.5.3.3 Le modèle *ANFIS*

Modèle le plus utilisé en pratique. C'est une structure à 6 couches qui affine les règles floues déjà établies par des experts humains et réajuste le chevauchement entre les différents sous-ensembles flous. Des applications dans le traitement du signal et le filtrage adaptatif ont été réalisées avec cette architecture.

Plusieurs ouvrages et articles montrent un bon rendement du modèle ANFIS lorsqu'il est utilisé dans la commande de moteurs.

III.5.4 Système d'inférence flou basé sur les réseaux de neurones adaptatifs ANFIS

Le système ANFIS (Adaptative Network based Fuzzy Inference System) est un réseau adaptatif proposé par Jang en 1993 [2]. Ce système peut être vu comme un réseau de neurones non bouclé pour lequel chaque couche est un composant d'un système flou. Le modèle ANFIS est le modèle le plus utilisé en pratique. Des applications notamment dans le traitement du signal, le filtrage adaptatif et la commande des systèmes ont été réalisées avec cette architecture. Plusieurs ouvrages et articles présentent des meilleures

performances du modèle ANFIS lorsqu'il est utilisé dans la commande et en particulier la navigation des robots mobiles [23-27].

Cette architecture neuro-floue affine les règles floues obtenues par des experts humains pour décrire le comportement entrée-sortie d'un système complexe en utilisant une base de donnée pour l'apprentissage. La sortie globale dans ce modèle est donnée par la moyenne pondérée de chaque sortie des règles actives (le produit ou minimum des degrés d'activation) et les fonctions d'appartenance de sortie. Il s'agit d'une technique neuro-floue hybride qui apporte les capacités d'apprentissage des réseaux de neurones au système d'inférence flou de type Takagi-Sugeno. Le rôle de l'apprentissage est l'ajustement des paramètres de ce système d'inférence flou (partie prémisse et partie conclusion des règles). La force du système ANFIS est la possibilité de génération automatique des règles floues en utilisant le *subtractive clustering* ou le *grid partitioning* [2].

III.5.5 Vue d'ensemble d'ANFIS

Le système hybride neuro-flou de type ANFIS se compose de cinq couches où les nœuds adaptatifs sont situés à la première et la quatrième couche (figure III.9).

Pour simplifier la compréhension et sans perte de généralité, nous considérons un système à deux entrées x_1 et x_2 et une sortie y . Considérons aussi un modèle flou de type TSK de ce système, composé des deux règles suivantes :

$$\text{Si } x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } B_1 \text{ alors } y_1 = f_1(x_1, x_2) = \alpha_1 x_1 + b_1 x_2 + c_1 \quad (\text{III.3})$$

$$\text{Si } x_1 \text{ est } A_2 \text{ et } x_2 \text{ est } B_2 \text{ alors } y_2 = f_2(x_1, x_2) = \alpha_2 x_1 + b_2 x_2 + c_2 \quad (\text{III.4})$$

Jang [2] a proposé de représenter cette base de règles par le réseau adaptatif de la figure III.9.

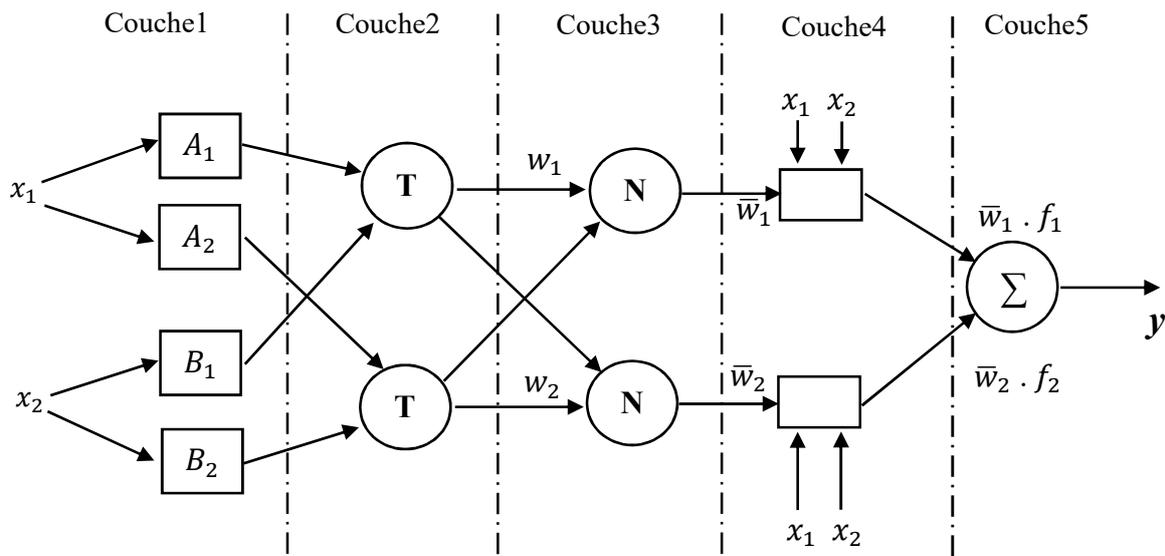


Figure III.9 Architecture équivalente d'ANFIS pour deux règles.

Le réseau adaptatif ANFIS est un réseau multicouches dont les connexions ne sont pas pondérées, ou ont toutes un poids de 1. Les nœuds sont de deux types différents selon leur fonctionnalité : les nœuds carrés (adaptatifs) contiennent des paramètres, et les nœuds circulaires (fixes) n'ont pas de paramètres. Toutefois, chaque nœud (carré ou circulaire) applique une fonction sur ses signaux d'entrées. La sortie O_i^k du nœud i de la couche k (appelée nœud (i, k)) dépend des signaux provenant de la couche $k-1$ et des paramètres du nœud (i, k) , c'est-à-dire,

$$O_i^k = f(O_1^{k-1} \dots O_{n_{k-1}}^k, a, b, c \dots) \quad (III.5)$$

Où n_{k-1} est le nombre de nœuds dans la couche $k-1$, et $a, b, c \dots$ sont les paramètres du nœud (i, k) . Pour un nœud circulaire ces paramètres n'existent pas.

Dans le réseau de la figure (III.9), les nœuds d'une même couche ont des fonctions issues d'une même famille que nous explicitons ci-dessous.

• **Première couche pour la fuzzification**

La première couche représente les fonctions d'appartenances floues. Chaque nœud de cette couche est un nœud carré avec une fonction :

$$O_i^1 = \mu_{A_i}(x) \quad (III.6)$$

Où x est l'entrée du nœud i , et A_i le terme linguistique associé à sa fonction. En d'autres termes, O_i^1 est le degré d'appartenance de x à A_i . Les paramètres d'un nœud de cette couche sont ceux de la fonction d'appartenance correspondante.

- **Deuxième couche pour les règles floues**

Chaque nœud i de cette couche est un nœud circulaire appelé Π qui engendre en sortie le produit de ses entrées. Ce produit représente le degré d'activation d'une règle :

$$w_i = \mu_{A_i}(x_1) \cdot \mu_{B_i}(x_2), i = 1, 2 \quad (\text{III.7})$$

- **Troisième couche pour la Normalisation**

Chaque nœud de cette couche est un nœud circulaire appelé N . La sortie du nœud i est le degré d'activation normalisé de la règle i :

$$v_i = \frac{w_i}{w_1 + w_2} \quad (\text{III.8})$$

La seconde et la troisième couche contiennent les nœuds qui forment la partie antécédente dans chaque règle.

- **Quatrième couche de Conséquence**

La quatrième couche calcule les coefficients de l'équation du premier ordre d'une règle de type Takagi-Sugeno et cela pour chaque règle floue. Chaque nœud de cette couche est un nœud carré avec une fonction réalisant le calcul :

$$O_i^4 = v_i \cdot f_i = v_i (a_i x_1 + b_i x_2 + c_i) \quad , i = 1...2 \quad (\text{III.9})$$

Où v_i est la sortie de la couche 3, et $\{a_i, b_i, c_i\}$ est l'ensemble des paramètres de sortie de la règle i .

- **Cinquième couche pour la Somme**

La cinquième couche qui représente la couche de sortie, calcule la sortie globale pesée (pondérée) du système. Le seul nœud de cette couche est un nœud circulaire qui effectue la somme des signaux provenant de la couche 4, c'est-à-dire,

$$O_i^5 = y = \sum_i v_i \cdot f_i \quad (\text{III.10})$$

La généralisation du réseau à un système à r entrées ne pose aucun problème particulier. Le nombre de nœuds de la couche 1 est toujours égal au nombre total de termes linguistiques définis.

L'apprentissage à partir d'un ensemble de données concerne l'identification des paramètres des prémisses et des conséquences, la structure du réseau étant fixée. L'algorithme d'apprentissage commence par construire un réseau initial, ensuite on applique une méthode d'apprentissage par rétro-propagation de l'erreur. Jang a proposé d'utiliser une règle hybride d'apprentissage qui combine un algorithme de descente de gradient avec une estimation par moindres carrées.

En ce qui concerne les paramètres des prémisses (couche 1), Jang a suggéré de n'appliquer une technique d'identification à partir des données que si l'ensemble de données est suffisamment grand. Dans ce cas, l'identification permet d'affiner les fonctions d'appartenance proposées par l'expert humain. Dans le cas contraire, il vaut mieux conserver les fonctions d'appartenance proposées par l'expert car elles reflètent les connaissances de l'expert.

Jang a montré l'équivalence entre les réseaux ANFIS et des réseaux neuronaux classiques avec des connexions pondérées. Il a également montré que sous certaines conditions, le comportement fonctionnel d'un ANFIS est équivalent à celui d'un RBFN (Radial Basis Function Network). Cette équivalence nécessite de nombreuses conditions restrictives sur les deux types de systèmes (flou ou RBF), entraînant donc leur perte d'identité propre [28].

III.5.6 Apprentissage de l'ANFIS

L'apprentissage est la phase de développement du réseau neuro-flou en optimisant les paramètres d'un système flou : les paramètres de la partie prémisses (fonctions d'appartenance) et la partie conclusion (les coefficients de sortie). Cette adaptation est basée sur la capacité d'apprentissage des réseaux de neurones artificiels multicouches à partir d'un ensemble de données. Pour l'identification des paramètres, la structure du réseau doit être fixée et les paramètres des fonctions d'appartenances et des conclusions seront ajustés en utilisant l'algorithme d'apprentissage de rétro-propagation avec une

combinaison avec l'algorithme des moindres carrés. Le système ANFIS est défini par deux ensembles de paramètres : S_1 et S_2 tels que :

S_1 : représente les paramètres des ensembles flous utilisés pour la fuzzification dans la première couche de l'ANFIS :

$$S_1 = ((a_{11}, b_{11}, c_{11}), (a_{12}, b_{12}, c_{12}), \dots, (a_{1p}, b_{1p}, c_{1p}), \dots, (a_{np}, b_{np}, c_{np})) \quad (\text{III.11})$$

p est le nombre des partitions floues de chacun des variables d'entrées et n est le nombre de variables d'entrées.

S_2 : représente les coefficients des fonctions linéaires (les paramètres conséquents) :

$$S_2 = (p_1, p_2, p_3, \dots, q_1, q_2, q_3, \dots, r_1, r_2, r_3) \quad (\text{III.12})$$

Jang [2] a proposé que la tâche d'apprentissage de L'ANFIS se faite en deux passages en utilisant un algorithme d'apprentissage hybride, comme illustré sur la table III.2.

	<i>Passage vers l'avant</i>	<i>Passage en arrière</i>
<i>Paramètre des prémisses</i>	Fixe	Rétro-propagation
<i>Paramètres des conséquents</i>	Moindres carrés	Fixe

Tableau III.2 Méthodes utilisées pour l'apprentissage de l'ANFIS.

III.6 Conclusion

Ce chapitre est divisé en deux parties ; dans la première partie, nous avons effectué une présentation générale des réseaux de neurones. Ils constituent une véritable solution pour la résolution de plusieurs problèmes, là où les méthodes classiques ont montrés leurs limites. L'utilisation des réseaux de neurones pour la commande peut se justifier par la simplicité de mise en œuvre, par leur capacité d'approximation universelle et par la possibilité de considérer le processus comme une boîte noire. Grâce à ces propriétés intéressantes, l'application des RNA pour la commande en robotique mobile présente un intérêt considérable.

Dans cette deuxième partie, nous avons présenté la méthode hybride neuro-floue (ANFIS). Nous avons vu que cette méthode combine les traitements parallèles et les capacités d'apprentissage des réseaux de neurones avec les raisonnements anthropomorphiques et les capacités d'explication des systèmes flous :

- Les *RNA* deviennent plus transparents, les systèmes flous acquièrent la capacité d'apprendre.
- Les *RNA* fonctionnellement équivalent à un modèle d'inférence floue ; on peut l'entraîner à développer des règles floues *SI-ALORS* et à trouver les fonctions d'appartenance de variables d'entrées/sorties en partant d'un ensemble de données représentatives.
 - On peut aussi y inclure les connaissances d'un expert.
 - Structure similaire à RNA multicouches : une couche d'entrée, une couche de sortie et trois couches cachées pour les fonctions d'appartenance et les règles.

REFERENCES BIBLIOGRAPHIQUES CHAPITRE III

- [1] E. Tunstel, Mo Jamshidi, "Fuzzy Logic and Behavior Control Strategy for Autonomous Mobile Robot Mapping", IEEE.1994.
- [2] J.S.R. Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference System", IEEE Transaction on Systems, Man, and Cybernetics, vol. 23, pp.665-685, 1993.
- [3] A. Guez, J. L. Elibert & M. Kam, "Neural Network Architecture for Control", IEEE Control Systems Magazine, vol. 8, no. 3, pp. 22-25, April 1988.
- [4] R. Fierro, F. L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks", IEEE Transactions On Neural Networks, vol. 9, no. 4, pp. 589-600, 1998.
- [5] R. S. Sutton, A. G. Barto, "Reinforcement Learning: An Introduction", MIT Press, Cambridge, MA, 1998.
- [6] P. Y. Glorennec, "Reinforcement Learning: an Overview", ESIT 2000, Aachen, Germany, September 2000.
- [7] D. Janglová, "Neural Networks in Mobile Robot Motion", International Journal of Advanced Robotic Systems, vol. 1, no. 1, pp.15-22, 2004.
- [8] J. Ferber, "Les Systèmes Multi Agents: Vers une Intelligence Collective", InterEdition, Université Pierre et Marie Curie Paris 6, 1995.
- [9] J. A. Anderson, "An Introduction to Neural Networks", Bradford - MIT Press, 1995.
- [10] E. Gauthier, "Utilisation des Réseaux de Neurones Artificiel pour la Commande d'un Véhicule Autonome", Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1999.
- [11] P. Y. Glorennec, L. Jauffe, "A Reinforcement Learning Method for an Autonomous Robot", Proceedings of the First Online Workshop on Soft Computing, 1996.
- [12] A. F. James, D. M. Skapura, "Neural Networks Algorithms, Applications and Programming Techniques", Addison Wesley, 1992.
- [13] J. A. Anderson, "An Introduction to Neural Networks", Bradford - MIT Press, 1995.
- [14] M. Parizeau, "Réseaux de neurones", Edition université de Laval, 2004.
- [15] C. Touzet, "Les Réseaux de Neurones Artificiels, Introduction au Connexionnisme", Paris: Masson, 1992.
- [16] J. A. Anderson, E. Rosenfeld, "Neurocomputing: Foundations of Research", MIT Press, Cambridge, 1988.
- [17] A. Guez, J. L. Elibert & M. Kam, "Neural Network Architecture for Control", IEEE Control Systems Magazine, vol. 8, no. 3, pp. 22-25, April 1988.
- [18] M. Boumehez, "Identification et Contrôle avec Réseaux de Neurones", Thèse de Magister, Département d'Electronique, Université de Sétif, 1993.

- [19] L. Personnaz, "Réseaux de Neurones Formels pour la Modélisation, la Commande et la Classification", CNRS Edition, Paris, 2003.
- [20] D. O. Abdeslam, "Techniques neuromimétiques pour la commande dans les systèmes électriques : application au filtrage actif parallèle dans les réseaux électriques basse tension", Thèse de doctorat, Université de Haute-Alsace, décembre 2005.
- [21] E. Gauthier, "Gestion d'une flotte de véhicules autonomes à l'intérieur d'un parking haute densité", Rapport d'étude, Institut National Polytechnique de Grenoble DEA Informatique Option Robotique Vision Image, juin 1995.
- [22] E. Gauthier, "Utilisation des réseaux de neurones artificiels pour la commande d'un véhicule autonome", Thèse de doctorat, Institut National Polytechnique de Grenoble, Janvier 1999.
- [23] C. N. Kim, M. M. Trivedi, "A Neuro-Fuzzy Controller for Mobile Robot Navigation and Multirobot Convoying", IEEE Transactions on Systems, Man and Cybernetics, vol. 28. no.6, pp. 829-840, 1998.
- [24] P. Y. Glorennec, "Algorithmes d'Apprentissage pour Systèmes d'inférence Floue", Edition Hermès, 1999.
- [25] P. Rusu, al, "Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation", IEEE Transactions on Instrumentation and Measurement, vol. 52, no. 4, pp.1335-1340, 2003.
- [26] B. H. Nurmal, al, "Time-Optimal Collision-Free Navigation of a Car like Mobile Robot using Neuro-Fuzzy Approaches", Fuzzy Sets and Systems, vol. 157, pp. 2171- 2204, 2006.
- [27] P. Boskoski, M. Stankovski, "Neuro-Fuzzy Controllers and Application to Autonomous Robots", Mechanics, Automatic Control and Robotics, vol. 7, no.1, pp.123-132, 2008.
- [28] Y. Nakoula, "Apprentissage des modèles linguistiques flous, par jeu de règles pondérées", Thèse de doctorat, Université de Savoie, 1997.

Chapitre IV

Navigation Autonome par des Contrôleurs Flous

IV.1 introduction

Comme on a vu dans le chapitre 2, la théorie de la logique floue est caractérisée par la capacité de modéliser et de traiter des informations incertaines et imprécises. Dans la majorité des applications de logique floue pour la navigation des robots mobiles, le modèle mathématique de la dynamique de l'environnement du robot n'est pas nécessaire dans le processus de conception du contrôleur de mouvement. Puisque la logique floue est un outil mathématique puissant qui permet de manipuler le raisonnement humain, les concepts et les termes linguistiques [1-2]. La connaissance de l'opérateur humain se présentera sous la forme d'un ensemble de règles linguistiques floues de type Si-Alors. Ces règles produisent une décision approximative de la même manière comme un expert [3-4].

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

Dans ce chapitre, on va présenter des architectures de navigation réactive d'un robot mobile autonome dans des environnements inconnus. Il s'agit de contrôler l'évolution du robot dans des environnements caractérisés par leurs complexité (ils peuvent être vastes, imprécis, dynamiques ou inconnus,...). Ce qui nous amène à la définition de certains comportements élémentaires tels que : *la convergence vers un but, suivi de*

murs, évitement d'obstacles, poursuite de trajectoire ou d'une cible mobile,...etc. Nous avons adopté un raisonnement flou pour élaborer les différents comportements ainsi que leur coordination. Le choix de la logique floue est motivé par le fait qu'elle permet d'utiliser des règles et des concepts linguistiques pour la conception des comportements, tout en permettant une fusion efficace au niveau des actions. De plus les autres techniques comme les réseaux de neurones et les champs de potentiels ne prennent pas en compte les incertitudes sur les données issues des capteurs [5].

IV.3 Modèle du robot mobile

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

La mission de base du robot mobile est d'atteindre un but désiré dans son environnement de navigation en évitant les obstacles qu'il contourne de façon autonome. Le système de navigation autonome doit générer les actions de mouvement : angle de braquage $\Delta\theta$ (ω) et la vitesse linéaire V_{rob} .

La figure IV.1 présente la configuration du robot mobile utilisé et les variables calculées dans l'espace de configuration [5][18].

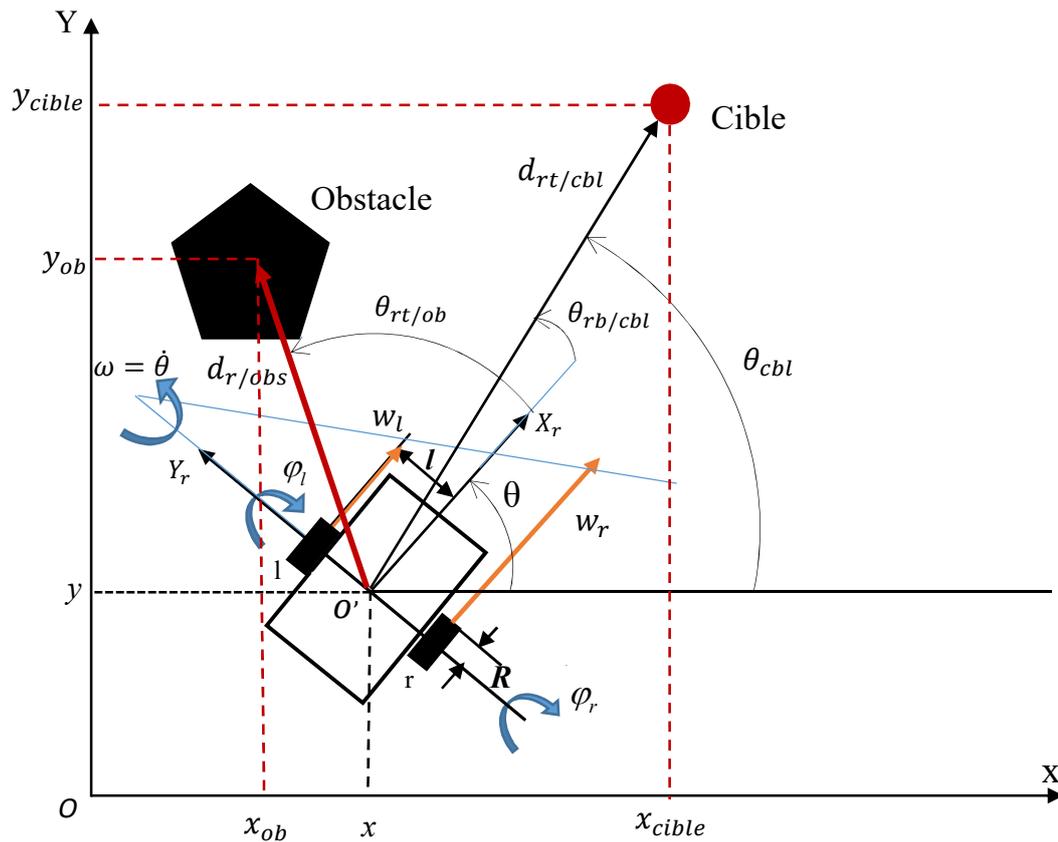


Figure IV.1 Configuration du robot mobile utilisé.

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

Le modèle cinématique du robot utilisé s'exprime dans le repère absolu (X, O, Y) par le système suivant [5][18] :

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

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

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

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

Où :

θ : L'orientation du robot,

l : la longueur du robot,

R : la longueur du châssis,

$v, \Delta\theta$: sont les commandes du robot mobile (la vitesse linéaire de translation du robot et l'angle de braquage respectivement).

IV.4 Présentation de l'odométrie

L'odométrie permet de déterminer la position (x, y) et l'orientation (θ) d'un robot mobile navigant sur un sol plan, par rapport au repère de référence qui était celui du robot dans sa configuration initiale. Cette technique est basée sur l'intégration des mouvements élémentaires des roues mesurés grâce à des codeurs incrémentaux. Connaissant le rayon de la roue et le nombre n d'impulsions délivrées par le codeur de résolution (α) durant un intervalle de temps Δt , il est possible de calculer la distance Δd parcourue par cette roue :

$$\Delta d = r n \alpha \quad (\text{IV.4})$$

IV.4.1 Application de l'odométrie pour la localisation d'un robot mobile

Le modèle d'évolution d'un robot mobile exprime son déplacement dans le repère de référence (R_0) en fonction du mouvement de ses roues. Considérons le cas d'un robot unicycle. Ce système de locomotion est très répandu pour les robots d'intérieur en raison de sa manœuvrabilité et sa simplicité de commande (Figure IV.2). Dans ce cas, le déplacement ΔD et la rotation élémentaire $\Delta\theta$ du robot dans le plan s'exprime en fonction des déplacements élémentaires des roues droite et gauche, respectivement Δd_d et, Δd_g par [17] :

$$\Delta D = \frac{\Delta d_d + \Delta d_g}{2} \quad (\text{IV.5})$$

$$\Delta\theta = \frac{\Delta d_d - \Delta d_g}{2l} \quad (\text{IV.6})$$

Désignons par (x_k, y_k, θ_k) , la configuration du robot à l'instant k , et par $(\Delta D_k, \Delta \theta_k)$ les composantes du déplacement élémentaire mesuré entre les instants k et $k+1$.

La rotation élémentaire à l'instant $k+1$ est :

$$\theta_{k+1} = \theta_k + \Delta \theta_k \quad (\text{IV.7})$$

Les formules les plus simples sont obtenues en considérant que le robot se déplace en ligne droite de ΔD_k dans la direction définie par θ_k , puis effectue une rotation sur place de $\Delta \theta_k$:

$$\begin{cases} x_{k+1} = x_k + \Delta D_k \cdot \cos(\theta_k + \frac{\Delta \theta_k}{2}) \\ y_{k+1} = y_k + \Delta D_k \cdot \sin(\theta_k + \frac{\Delta \theta_k}{2}) \end{cases} \quad (\text{IV.8})$$

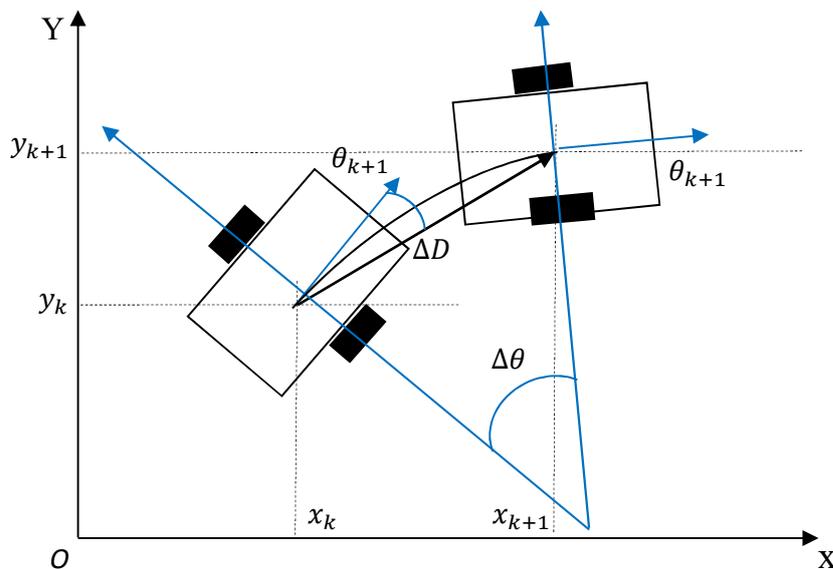


Figure IV.2 L'odométrie appliquée à la plateforme mobile.

IV.5 Navigation basée sur les comportements

L'un des aspects les plus difficiles dans le domaine de la robotique mobile est la possibilité de naviguer de manière autonome pour rejoindre un but, en évitant les obstacles modélisés et non modélisés, et en particulier dans un environnement encombré d'objets statiques ou dynamiques imprévisibles [6]. Une méthode très efficace peut être utilisée pour accomplir la tâche de navigation est la structure basée sur les comportements [7-8-9] [18]. Le principe de base du système de navigation basé sur les

comportements est de subdiviser la tâche de navigation globale en un ensemble des comportements élémentaires d'action (*comportement 1, comportement 2, ..., comportement n*) ; simple à concevoir et à gérer.

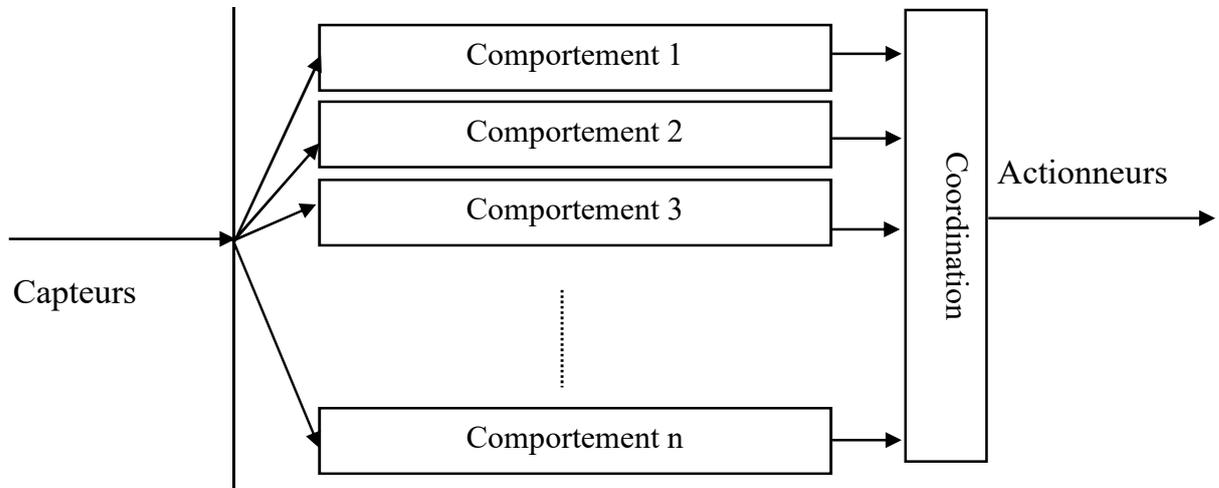


Figure IV.3 Architecture basée sur les comportements.

Arkin [10] a décrit l'utilisation de comportements réactifs appelés "*motorschemas*". Dans cette architecture, la méthode de champ de potentiel est utilisée pour définir la sortie de chaque bloc. Puis, toutes les sorties calculées sont combinées par une somme pondérée.

Rosenblatt [11] a présenté l'architecture *DAMN* (*Distributed Architecture for Mobile Navigation*). Elle consiste à utiliser une architecture distribuée multi-comportements (agents), chaque comportement représente une fonction bien particulière avec la caractéristique que ces agents ne peuvent pas discuter directement entre eux mais doivent passer par une sorte d'ordonnanceur qui gère leurs communications et leurs états.

L'architecture subsumption est très efficace puisque le système est de nature modulaire, ce qui simplifie à la fois la résolution de problème de planification de trajectoire du robot mobile, et offre la possibilité de supprimer ou d'ajouter de nouveaux comportements au système sans provoquer d'augmentation importante de la complexité. Les sorties proposées de chaque comportement actif simultanément sont ensuite rassemblées selon une règle de coordination des actions inférées. La tâche de contrôle se réduit alors à un couplage des entrées sensorielles avec les actionneurs en utilisant des blocs de comportements sous formes des contrôleurs réactifs pour le robot mobile.

Chaque comportement peut avoir des entrées provenant des capteurs du robot (caméra, ultrasons, infrarouges, ...) et/ou d'autres comportements dans le système, et envoyer des sorties aux actionneurs du robot (moteurs, roues, ...) et/ou à d'autres comportements. D'autres auteurs ont utilisés la logique floue pour la représentation et la coordination des comportements [12-13]. La figure IV.4 présente la structure du système de contrôle basé sur les comportements flous proposé dans cette thèse pour la navigation d'un robot mobile dans un environnement inconnu.

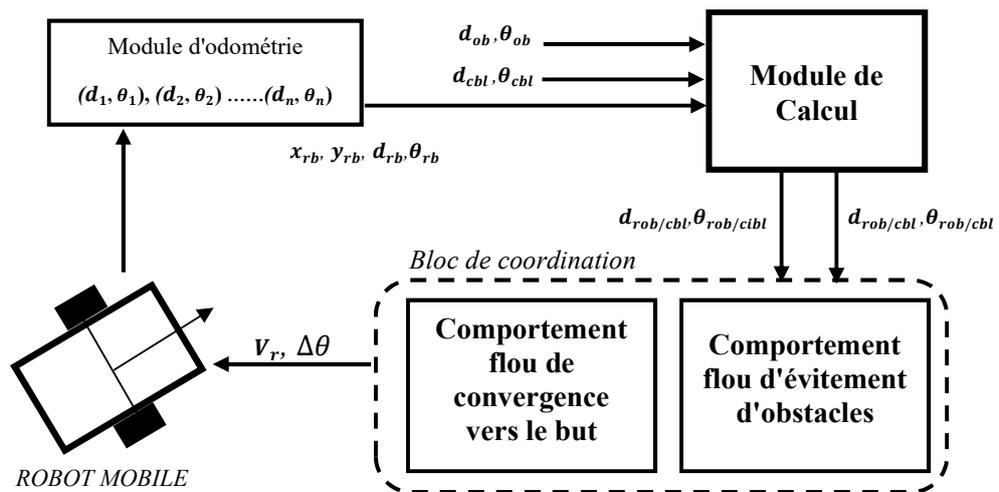


Figure IV.4 Structure du système de navigation proposé.

IV.6 Navigation basée sur les comportements flous

Dans la théorie de contrôle flou, les commandes inférées sont traités en trois étapes : fuzzification, inférence et déffuzzification. Le système de contrôle du robot mobile basé sur des comportements flous, est décomposé en sous-tâches simples (comportements indépendants), ou chaque bloc est considéré comme un contrôleur flou défini par un ensemble de règles floues visant à atteindre un objectif bien défini (éviter un obstacle, chercher la cible, suivre une trajectoire,...) [14-15-16]. Les règles linguistiques peuvent être par exemple :

R1 : Si le but est Loin Alors la vitesse est Grande et le braquage est Zéro,

R2 : Si l'obstacle est à droite Alors la vitesse est Moyenne et le braquage est à Gauche,

...

Rn : Si le but est Proche Alors la vitesse est Faible et le braquage est Zéro.

Dans ce travail, on utilise des stratégies de contrôle flou pour la navigation d'un robot mobile de type uni-cycle. On adopte des structures basées sur les comportements flous comme présentées sur la figure IV.4 illustrant la décomposition de la tâche de navigation globale en un ensemble des comportements élémentaires. Grâce aux propriétés importantes et efficaces pour la commande des robots mobiles, les avantages des comportements flous viennent notamment de leurs capacités à :

- Formaliser et simuler l'expertise d'un opérateur dans la conduite et la commande des procédés où les connaissances sont formulées sous forme d'une base de règles de type : *Si-Alors*,
- Maitriser les systèmes non linéaires et difficiles à modéliser par ces propriétés d'approximation universelles,
- Prendre en compte sans discontinuité des cas ou exceptions de natures différentes, et les intégrer au fur et à mesure dans l'expertise,
- Prendre en compte plusieurs variables et effectuer la fusion pondérée des grandeurs d'influences, et grâce à des processeurs efficaces, on peut régler des processus rapides.

IV.7 Conception des comportements flous

Pour permettre au robot mobile de se déplacer de manière autonome dans des environnements inconnus encombrés d'obstacles, on va présenter dans cette section l'approche étudiée et les structures de navigation du robot mobile à roues en utilisant les règles d'implication de MAMDANI (max-min). Cette méthode de conception est motivée par l'efficacité de commande et la simplicité pour rendre les contrôleurs plus appropriés en implémentation temps réel. Le système de commande global du robot mobile est basé sur le modèle cinématique du centre du robot (eq. IV.1) ; où les contrôleurs flous permettent de générer les deux actions de mouvement (angle de braquage du robot $\Delta\theta$ et vitesse de déplacement du robot V_r). Ces commandes sont liées par des comportements élémentaires de base pour le robot mobile : recherche de la cible et l'évitement d'obstacles. Les actions générées sont basées sur les mesures des capteurs du robot mobile pour la détection de la position de but ou pour la mesure des distances aux obstacles.

Les comportements conçus sont à structures fixes, comportent un minimum de règles floues pour être simple à interpréter, lisible et avec efficacité de commande. Chaque comportement est représenté par une base de règle avec deux entrées x_1 , x_2 et deux sorties sous la forme suivante :

$$\text{Si } x_1 \text{ est } A_1^i \text{ et } x_2 \text{ est } A_2^i \text{ alors } \alpha \text{ est } B_1^i \text{ et } V_r \text{ est } B_2^i \quad (\text{IV.9})$$

IV.7.1 Comportement de navigation vers un but (recherche d'une cible)

Ce comportement permet de réaliser l'action "*convergence vers le but*" par le robot à partir de la connaissance de sa position courante et la définition d'une position relative à atteindre dans l'environnement de navigation (faire aligner le robot vers la direction de but). La définition du point à atteindre est effectuée par l'intermédiaire de deux variables d'entrées ; la distance entre le robot et le but ($d_{rob/cible}$) et l'angle entre l'orientation actuelle du robot et celle de but θ_{cible} (figure IV.1). Le schéma fonctionnel de contrôle proposé pour le robot est donné par la (figure IV.4). Le module de calcul compare les coordonnées réelles du robot avec les coordonnées de but afin de calculer la distance robot-but et l'angle désiré pour le rejoindre. Puis cette valeur d'angle est comparée avec l'orientation actuelle du robot pour calculer l'angle entre l'axe du robot et de but (θ_{cible}) en utilisant les équations suivantes :

$$d_{rb/cible} = \sqrt{(x_{cbl} - x_{rb})^2 + (y_{cbl} - y_{rb})^2} \quad (\text{IV.10})$$

$$\theta_{cbl} = \arctg\left(\frac{y_{cbl} - y_{rob}}{x_{cbl} - x_{rob}}\right) \quad (\text{IV.11})$$

$$\theta_{rob/cible} = \theta_{cbl} - \theta_{rob} \quad (\text{IV.12})$$

Le contrôleur proposé est basé sur la règle d'implication de MAMDANI (max-min), la conjonction floue (min) et par la méthode de centre de gravité pour la défuzzification. Il a comme entrées la distance d et l'orientation θ .

Les fonctions d'appartenance utilisées pour la partition floue de l'espace des variables d'entrées et de sortie sont triangulaires, symétriques et équidistantes, figures IV.5 et IV.6.

Pour aboutir à notre commande floue, nous avons développé un programme sur plusieurs variantes de quelques univers de discours (UD), le choix de l'univers de discours retenu est obtenu après plusieurs tests.

1- les variables d'entrée :

Pour la distance $d_{rob/cible}$ (m) on a choisi quatre fonctions d'appartenance: very near (VN), near (NR) et far (F) et very far (VF) réparties sur l'univers de discours $[0, 30]$, pour l'entrée θ (rad) sept fonctions d'appartenance sont associées : Negative big (NB), Negative medium (NM), Negative small (NS), zero (Z), Positive small (PS), Positive medium (PM) et Positive big (PB), réparties sur l'univers de discours $[-3, 3]$.

Les valeurs linguistiques peuvent être décrites par rapport aux sous-ensembles flous, L_d et L_θ comme :

$$L_d = \{VN, N, F, VF\}$$

$$L_\theta = \{NB, NM, NS, Z, PS, PM, PB\}$$

2- les variables de sortie :

Pour la vitesse V (m/s) on a choisi aussi quatre fonctions d'appartenance: very slow (VS), slow (S) et fast (F) et very fast (VF) réparties sur l'univers de discours $[0, 0.2]$, pour l'angle de braquage $\Delta\theta$ (rad) sept fonctions d'appartenance sont associées : Negative big (NB), Negative medium (NM), Negative small (NS), zero (Z), Positive small (PS), Positive medium (PM) et Positive big (PB), répartis sur l'univers de discours $[-0.8, 0.8]$ [5][17].

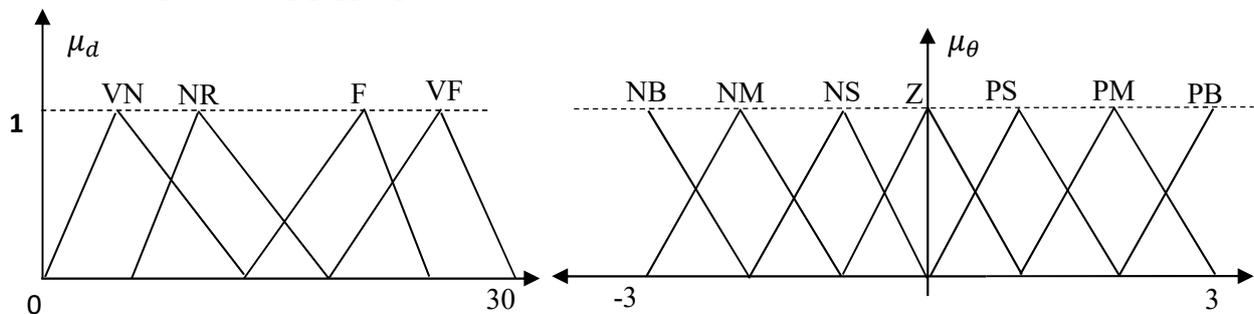


Figure IV.5 Les fonctions d'appartenances des variables d'entrées d et θ .

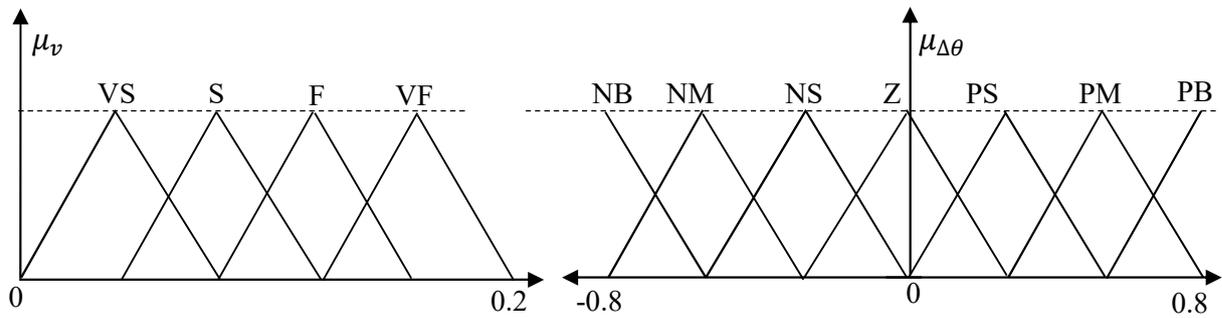


Figure IV.6 Les fonctions d'appartenance des variables de sorties V et $\Delta\theta$.

Sur la base de la description du système à commander avec des variables linguistiques et de la définition des fonctions d'appartenance pour les variables d'entrée et de sortie, on peut établir les règles d'inférence. L'étape d'inférence fait appel à 28 règles représentant le lien entre les différentes variables entrée-sortie. La table (IV.1) décrit les règles floues utilisées pour ce comportement.

$\theta \backslash d$	VN	N	F	VF
NB	VS, PB	VS, PB	S, PB	S, PM
NM	VS, PM	VS, PS	S, PS	F, PS
NS	VS, PS	S, PS	F, PS	F, PS
Z	VS, Z	S, Z	F, Z	VF, Z
PS	VS, NS	S, NS	F, NS	VF, NS
PM	VS, NM	VS, NM	S, NM	F, NS
PB	VS, NB	VS, NB	S, NB	S, NB

Tableau IV.1 Les règles floues pour le comportement convergence vers un but.

Le comportement présenté dans la section précédente permet de réaliser la navigation autonome d'un robot mobile vers un but si le robot évoluant dans un environnement non contraignant. Une telle condition est très restrictive vu la nature de l'environnement dans lequel le robot est amené à réaliser ses missions. Dans le cas de présence d'obstacles (statiques ou dynamiques) qui empêchent le mouvement du robot mobile vers son objectif, le robot doit avoir une capacité efficace d'évitement d'obstacles.

IV.7.2 Comportement d'évitement d'obstacles

Dans ce qui suit, on présente la conception du deuxième comportement de base qui s'intéresse à la génération des actions adéquates pour l'évitement des collisions avec les obstacles si un ou plusieurs objets sont observés dans un certain voisinage du robot

mobile par ses moyens de perception (en face, à droite et à gauche). Dans notre travail, on propose une structure de commande pour l'évolution du robot mobile dans un environnement encombré d'obstacles, basée sur le comportement d'évitement d'obstacles et comportement convergence vers le but en utilisant les mesures de distance et de l'angle entre le centre du robot et l'obstacle et le centre du robot et le but .

Cette stratégie de navigation réalisée est basée sur l'utilisation de contrôleur flou qui peut, à partir de la distance $D_{rb/obs}$ et l'angle θ_{obs} (figure IV.1), de générer les commandes nécessaires pour l'évitement d'obstacles. Où $D_{rb/obs}$ est la distance entre le robot et le plus proche obstacle, θ_{obs} est l'angle relatif entre l'axe du robot et l'obstacle. Ce contrôleur est de type MAMDANI (max-min), qui peut être utilisé dans les environnements moins compliqués avec des obstacles uniformes. La figure IV.4 représente l'architecture proposée pour la navigation autonome du robot mobile. On a utilisé un système de contrôle basé sur deux comportements flous (le comportement précédent pour la convergence vers le but et un comportement pour l'évitement d'obstacles) avec un bloc de coordination simple pour le choix entre les deux contrôleurs. La tâche de navigation globale se réalise par l'activation de l'un des comportements ; soit converger vers le but ou éviter les obstacles.

1- Les variables d'entrées :

Pour la distance $d_{rob/ob}$ (m) on a choisi quatre fonctions d'appartenance: very near (VN), near (NR) et far (F) et very far (VF) réparties sur l'univers de discours $[0, 1.5]$, pour l'entrée φ (rad) sept fonctions d'appartenance sont associées : Negative big (NB), Negative medium (NM), Negative small (NS), zero (Z), Positive small (PS), Positive medium (PM) et Positive big (PB), réparties sur l'univers de discours $[-3, 3]$. Les valeurs linguistiques peuvent être décrites par rapport aux sous-ensembles flous. Ld_{ob} et $L\varphi$ comme [5] (Figure IV.7) :

$$Ld_{ob} = \{VN, N, F, VF\}$$

$$L\varphi = \{NB, NM, NS, Z, PS, PM, PB\}$$

2- Les variables de sorties :

Pour la vitesse V (m/s) on a choisi aussi quatre fonctions d'appartenance: very slow (VS), slow (S) et fast (F) et very fast (VF) réparties sur l'univers de discours $[0, 0.2]$, pour la $\Delta\theta$ (rad) sept fonctions d'appartenance sont associées : Negative big (NB), Negative medium (NM), Negative small (NS), zero (Z), Positive small (PS), Positive medium (PM) et Positive big (PB), répartis sur l'univers de discours $[-0.8, 0.8]$.

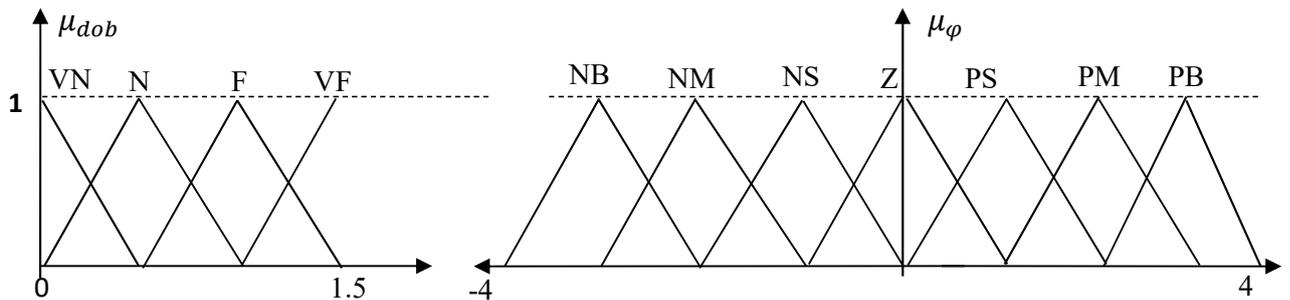


Figure IV.7 Les fonctions d'appartenances des variables d'entrées d_{ob} et φ .

Ce qui donne une base de 28 règles floues illustrées sur la table IV.2. Avec un principe de tourner à gauche si l'obstacle est à droite, et braquer à droite si l'obstacle se situe à gauche,...etc. Ces règles sont de la forme :

R1 : Si l'obstacle est en Face Alors la vitesse est Faible et le braquage est à Droite,

R2 : Si l'obstacle est à Droite Alors la vitesse est Moyenne et le braquage est à Gauche,

...

$\varphi \backslash d_{ob}$	VN	N	F	VF
NB	VS, PM	VS, PM	S, PM	S, PM
NM	VS, PM	VS, PS	S, PS	F, PS
NS	VS, PS	S, PS	F, PS	F, PS
Z	S, PM	S, NS	F, PS	VF, PS
PS	S, NM	S, NS	F, NS	VF, NS
PM	S, NM	VS, NM	S, NS	F, NS
PB	VS, NB	VS, NB	S, NB	S, NB

Tableau IV.2 Base de règles pour le comportement d'évitement d'obstacles.

IV.8 Résultats de simulation

Dans cette section, des exemples de navigation du robot mobile dans des environnements d'intérieur seront présentés pour vérifier la validité et l'efficacité des schémas de commande proposés. L'environnement utilisé prend en compte les contraintes de modélisation et de mouvement du robot utilisé dans plusieurs situations telles que : l'espace libre et l'environnement avec des obstacles statiques.

IV.8.1 Comportement de convergence vers un but

En absence d'obstacles dans l'espace de travail du robot, la tâche de navigation se réduit à une orientation et un déplacement direct vers la cible. Les Figures IV.8, IV.9 et IV.10 illustrent les trajectoires de navigation en choisissant différents points de départ et d'arrivée. Ces résultats montrent l'aptitude du contrôleur flou à générer les actions de commande les plus appropriées pour accomplir la tâche, ce qui prouve l'efficacité du contrôleur proposé.

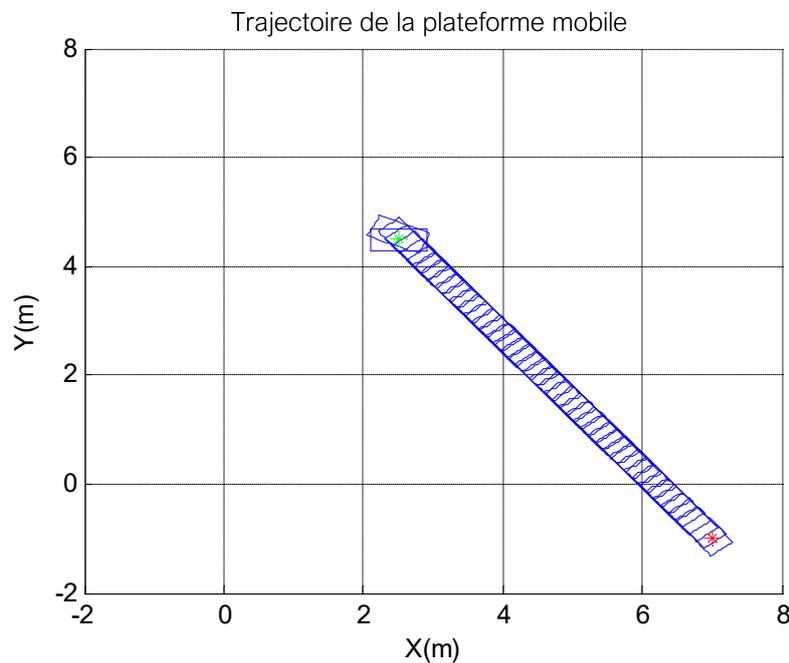


Figure IV.8 Navigation libre $x_1=2.5$, $y_1=4.5$ $\theta=\pi$ vers $x_f=7$, $y_f=-1$.

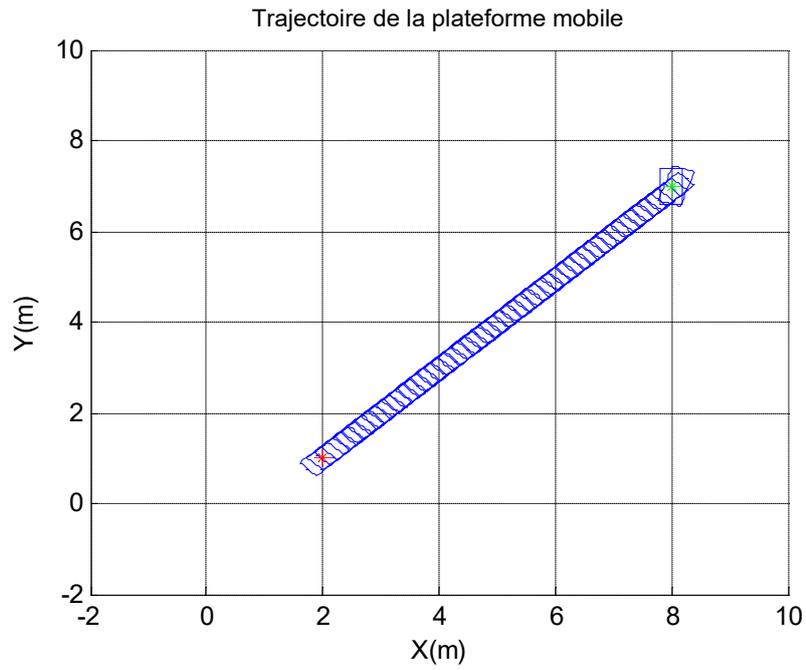


Figure IV.9 Navigation libre $x_1=8, y_1=7, \theta=\pi/2$ vers $x_f=2, y_f=1$.

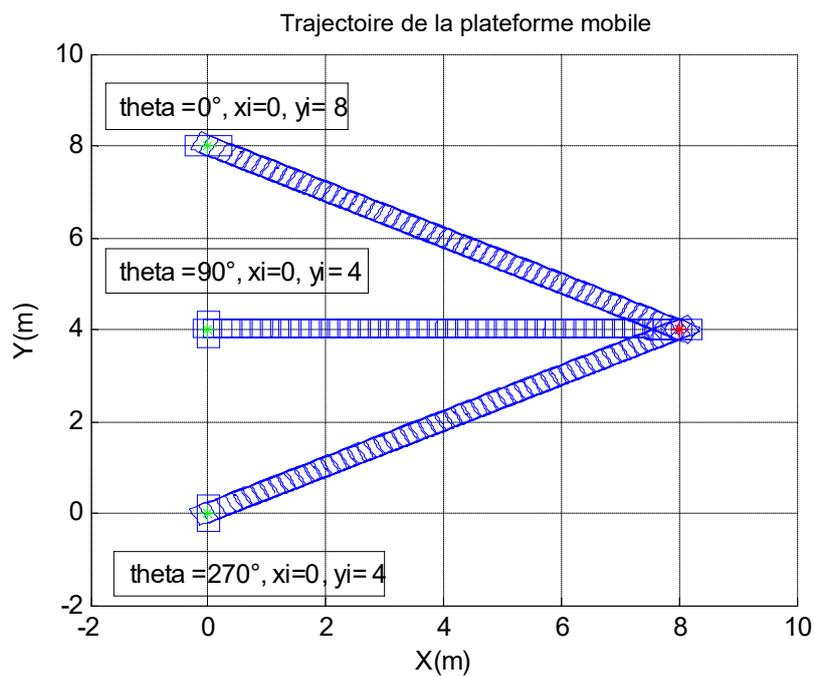


Figure IV.10 Navigation libre à partir de trois différentes configurations initiales vers un but en utilisant un contrôleur flou MAMDANI (max min).

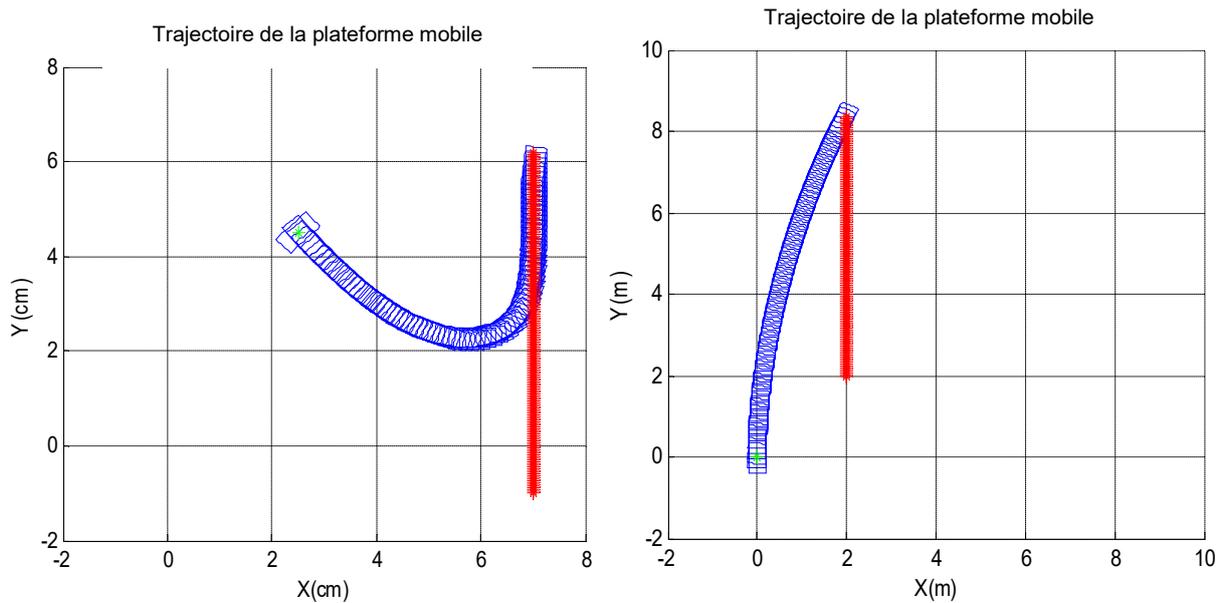


Figure IV.11 Navigation libre avec cible mobile.

IV.8.2 Comportement d'évitement d'obstacles

Si l'environnement du robot contient un ou plusieurs obstacles, le robot doit pouvoir éviter les collisions avec ces objets. Comme il est présenté précédemment, le système de navigation autonome contient deux comportements élémentaires (figure IV.4) : comportement de recherche de but et autre pour l'évitement d'obstacle. Le robot exécute l'action adéquate pour atteindre la destination finale en toute sécurité sans risque de collision avec les objets en déclenchant l'un des deux comportements selon la situation perçue.

Les figures IV.12-13-14 montrent des exemples de navigation du robot mobile en présence d'obstacles dans l'environnement. Le robot se déplace vers l'objectif, lorsqu'un obstacle est détecté dans l'un des trois côtés (en face, à droite ou à gauche), le comportement d'évitement d'obstacle est activé afin de générer les actions appropriées pour éviter ces collisions, dans tous les cas, le robot est capable de naviguer de manière autonome et peut atteindre son but efficacement en évitant les obstacles avec succès quel que soit sa position initiale. La trajectoire de mouvement obtenue et les actions générées montrent que le système de commande proposé donne des meilleures performances et d'efficacité.

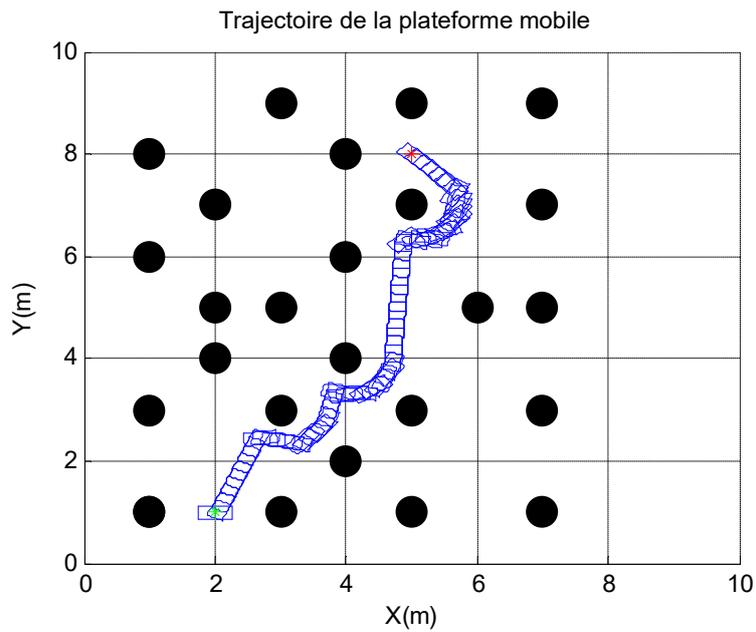


Figure IV.12 Navigation en présence d'obstacles. $x_1=2, y_1=1, \theta=0$, vers $x_f=5, y_f=8$.

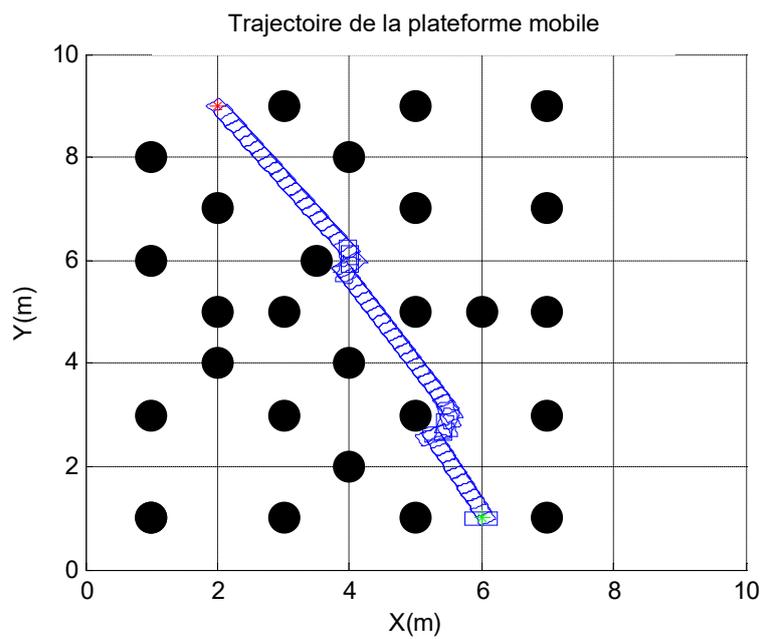


Figure IV.13 Navigation en présence d'obstacles. $x_1=6, y_1=1, \theta=0$, vers $x_f=2, y_f=9$.

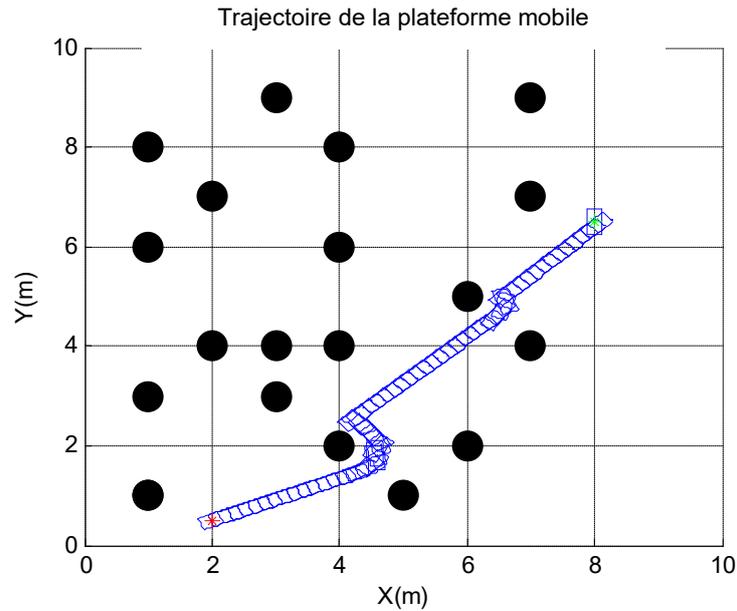


Figure IV.14 Navigation en présence d'obstacles. $x_1=8, y_1=6.5, \theta=\pi/2$, vers $x_f=2, y_f=0.5$.

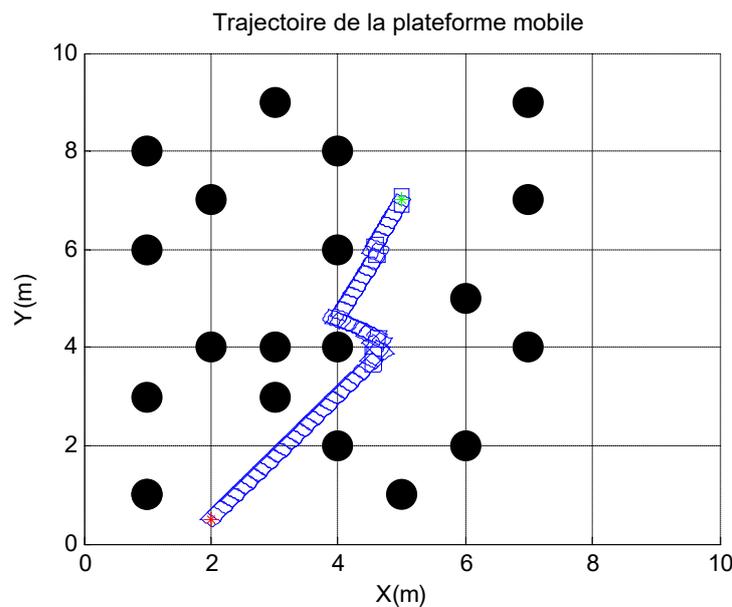


Figure IV.15 Navigation en présence d'obstacles. $x_1=5, y_1=7, \theta=-\pi/2$, vers $x_f=2, y_f=0.5$.

IV.9 Conclusion

Dans ce chapitre, un contrôleur comportemental à base de logique floue a été conçu pour réaliser la navigation autonome d'un robot mobile dans des environnements inconnus encombrés d'obstacles. L'idée du contrôle comportemental est basée sur la décomposition de la tâche de navigation globale en un ensemble de comportements élémentaires tels que la navigation vers la cible, l'évitement d'obstacles et le suivi de

murs. Les résultats de simulation présentés donnent des solutions prometteuses pour la commande autonome du robot mobile surtout dans les environnements moins compliqués. Dans tous les cas, le robot est capable d'atteindre son objectif en évitant les obstacles.

La mise en place d'un système de commande floue nécessite une expérience pour une meilleure conception de ce type des contrôleurs. La logique floue est un outil puissant et assez simple à implémenter. Mais ses contrôleurs présentent quelques inconvénients, entre autres : la nécessité d'évaluer et d'affiner les contrôleurs (les fonctions d'appartenance d'entrée et de sortie et les tables d'inférence). Le réglage des différents paramètres dans un environnement pouvait fournir de bons résultats pour cet environnement et ne pas répondre correctement face à une situation nouvelle. De manière à augmenter l'adaptabilité du système de navigation, nous nous sommes tournés vers le domaine de l'apprentissage automatique. Une solution possible est l'application des techniques d'apprentissage pour l'ajustement et la génération des paramètres du contrôleur flou.

Dans le chapitre suivant, on va présenter les possibilités offertes par les réseaux de neurones artificiels pour l'adaptation des paramètres des comportements destinés à la commande automatique d'un robot mobile.

REFERENCES BIBLIOGRAPHIQUES CHAPITRE IV

- [1] J. M. Mendel, "Tutorial: Fuzzy Logic Systems for Engineering", Proceeding of the IEEE, vol. 83, no. 3, pp. 345-377, March 1995.
- [2] J. Jantzen, "Foundations of Fuzzy Control", West Sussex (England), John Wiley & Sons Ltd, 2007.
- [3] A. Saffiotti, "The Uses of Fuzzy Logic for Autonomous Robot Navigation", Soft Computing, vol. 1, no. 4, pp.180-197, 1997.
- [4] X. Yang, al, "A Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation", IEEE Transaction on Systems, Man, and Cybernetics, B, vol. 35, no.6, pp.1214-1224, 2005.
- [5] **K. boucetta, A. hassam, "Comparative Study between Fuzzy Logic and Interval Type-2 Fuzzy Logic Controllers for Trajectory planning of a Mobile Robot ", Engineering, Technology & Applied Science Research, Vol. 11, No. 2, 2021, 7011-7017, 2021.**
- [6] F. Cuesta, A. Ollero, "Intelligent Mobile Robot Navigation", Springer-Verlag, Berlin Heidelberg, 2005.
- [7] R. Brooks, "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, vol. 2, no.1, pp.14-23, 1986.
- [8] R. C. Arkin, "Motor Schema-based Mobile Robot Navigation", International Journal of Robotic Research, vol. 8, pp. 92-112, 1989.
- [9] J. Rosenblatt, D. W. Payton, "A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control", Proceedings of the IEEE/INNS International Joint Conference on Neural Networks, Washington DC, vol. 2, pp. 317-324, 1989.
- [10] R. C. Arkin, "Motor Schema-based Mobile Robot Navigation", International Journal of Robotic Research, Vol. 8, pp. 92-112, 1989.
- [11] J. Rosenblatt, D. W. Payton, "A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control", Proceedings of the IEEE/INNS International Joint Conference on Neural Networks, Washington DC, Vol. 2, pp. 317-324, 1989.
- [12] W. li, J. Farrell, F. M. Wahl & K. R. Kozlowski, "A nonholonomic mobile robot navigation in uncertain environments based on behavior control", World Automation Congress 8th Inter. Symposium on Robotics with Applications, Maui, Hawaii, pp. 1-8, June 2000.
- [13] Y. S. Lee, S. B. Cho, "A hybrid system of hierarchical planning of behavior selection networks for mobile robot control", Inter. Journal of Advanced Robotic systems, Vol. 11, No. 57, pp. 1-3, 2014.
- [14] X. Yang, al, "A Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation", IEEE Transaction on Systems, Man, and Cybernetics, B, vol. 35, no.6, pp.1214-1224, 2005.
- [15] A. Fatmi, A. Al Yahmadi, L. Khriji, & N. Masmoudi, "A Fuzzy Logic Based Navigation of a Mobile Robot", World Academy of Science, Engineering and Technology, vol. 22, pp. 169-174, 2006.

- [16] M. Wang, N. K. Liu James, "Fuzzy Logic based Real-time Robot Navigation in Unknown Environment with Dead Ends", *Robotics and Autonomous Systems*, vol. 56, pp.625-643, 2008.
- [17] **A. Hassam, B. Kasmi & M. Boubezoula, "Planning of a Trajectory Along with Avoiding the Obstacle Based on Fuzzy Logic Method for a Unicycle Mobile Robot", *The 13th International Arab Conference on Information Technology ACIT'2012 Dec.10-13, ISSN: 1812-0857, 2013.***
- [18] H. Miloud, "navigation des robots mobiles non-holonomes sous contrôle flou ", Thèse de doctorat du Département d'Electronique de l'Université de Ferhat Abbes Sétif -1, 2016.

Chapitre V

Navigation Autonome par des Contrôleurs Neuro-Flous

V.1 introduction

La commande floue a montré son efficacité pour la navigation des robots mobiles, mais la construction d'un système flou performant n'est pas toujours facile. Son principal inconvénient est le manque d'une méthodologie systématique pour la conception, due au nombre important de paramètres à régler (les paramètres des fonctions d'appartenances, les paramètres de la partie conclusion et les règles d'inférences). Une solution possible est l'intégration des propriétés des systèmes flous avec d'autres approches de l'intelligence artificielle comme les réseaux de neurones [1-2], l'apprentissage par renforcement [3-4], les algorithmes génétiques [5-6], colonie de fourmis [7],...etc. Ces méthodes hybrides combinent les propriétés de chaque approche afin d'optimiser les paramètres des systèmes d'inférence flous [8].

La capacité d'apprentissage des réseaux de neurones peut être utilisée pour automatiser le processus d'ajustement des paramètres des systèmes d'inférence flous, en réduisant considérablement le temps de développement, et d'améliorer les performances des contrôleurs flous. Cette hybridation a permis la création des contrôleurs neuro-flous qui sont actuellement l'un des domaines de recherche les plus populaires. Les systèmes hybrides neuro-flous [9] permettent de tirer les avantages de ces deux approches, les

réseaux de neurones (RN) offrent une capacité d'apprentissage et de généralisation permettant une représentation efficace de la connaissance, et la logique floue (LF) permet de traduire l'expérience humaine en un ensemble de règles linguistiques et facilite le traitement des connaissances imprécises. Ces systèmes hybrides sont dérivés des technologies émergentes qui sont beaucoup plus efficaces dans le développement des systèmes intelligents que les méthodes classiques, et ont démontrés son efficacité dans plusieurs domaines d'applications et notamment en commande des processus.

Plusieurs combinaisons des RNA et la LF ont été développées dans la littérature [1], [10-11]. Elles ont donné naissance à des systèmes neuro-flous, qui sont le plus souvent orientés vers la commande des systèmes complexes et les problèmes de classification. Les systèmes hybrides neuro-flous sont classés généralement en deux catégories : les systèmes neuro-flous de type (ANFIS) [1], et les systèmes hybrides comportant des réseaux de neurones et des systèmes flous [9], [12]. La première catégorie est la plus utilisée, et elle est conçue pour combiner les capacités d'apprentissage des réseaux de neurones et les propriétés de raisonnement de la logique floue. La fonction principale du réseau de neurones dans ce type est d'apprendre davantage sur le comportement du système d'inférence flou, et utilise cette connaissance pour modifier de manière adaptative ces paramètres. L'adaptation du SIF est réalisée par la modification de la base de règles et/ou de fonctions d'appartenance (les paramètres de la partie condition et conclusion des règles floues). Les règles peuvent être générées, modifiées ou éliminées, tandis que les fonctions d'appartenance des variables entrée-sortie peuvent être ajustées par des mécanismes de mise à jour [1]. L'idée de base d'utiliser les systèmes de la deuxième catégorie est de remplacer toute ou une partie des systèmes flous par des réseaux de neurones ou inversement [13-14], [12]. Le système de commande global est une association des RNA et SIFs. L'objectif de ces arrangements est de combiner les avantages de ces deux approches, réduisent la vitesse de traitement et présentent une efficacité de commande.

Ces méthodes hybrides sont des solutions prometteuses pour la tâche de navigation d'un robot mobile, ou la mission présente une connaissance incomplète et incertaine en raison de l'imprécision inhérente du système sensoriel du robot, et la difficulté

rencontrée par l'utilisateur pour générer les règles floues représentant les relations entrée-sortie. Le robot mobile pour être autonome, doit être doté par une capacité de décision intelligente qui lui permet de se déplacer vers un objectif désiré en exécutant quelques fonctionnalités dans son environnement. Les robots mobiles, connaissant leurs positions, se déplacent d'un point à un autre avec diverses contraintes.

Dans ce chapitre, on va présenter l'application de contrôleur flou, pour la commande d'un robot mobile afin de réaliser une tâche de poursuite d'une trajectoire de référence. Pour améliorer les performances du robot mobile, un contrôleur hybride neuro-flou de type ANFIS est utilisé pour une mission de poursuite de trajectoire. Les deux contrôleurs proposés seront appliqués aussi pour la tâche de poursuite d'une cible mobile. Les résultats sont comparés et discutés.

V.2 La tâche de poursuite d'une trajectoire

En robotique mobile, la poursuite de trajectoire est une tâche élémentaire qui représente la base de toute autre tâche du robot. C'est une fonction importante que doit exécuter un robot mobile avec le minimum d'erreurs. La trajectoire est un ensemble de points à parcourir par le robot pour atteindre son but final [15-16]. La trajectoire à suivre est représentée par un certain nombre de points de passage entre le point de départ et le point d'arrivée. Le chemin à suivre peut être une trajectoire d'évitement d'obstacle issue d'un planificateur, ou un nombre de points de passage qui peuvent être des postes de travail dans l'environnement de navigation. Le robot exécute sa trajectoire point par point, en corrigeant son orientation en fonction de sa position par rapport au point désiré. Le rôle des systèmes de contrôle proposés est de régler l'orientation du robot sur la trajectoire en générant les actions appropriées. Généralement pour exécuter cette mission, le robot se déplace avec une vitesse fixe où la commande inférée est seulement l'angle de braquage α du robot mobile.

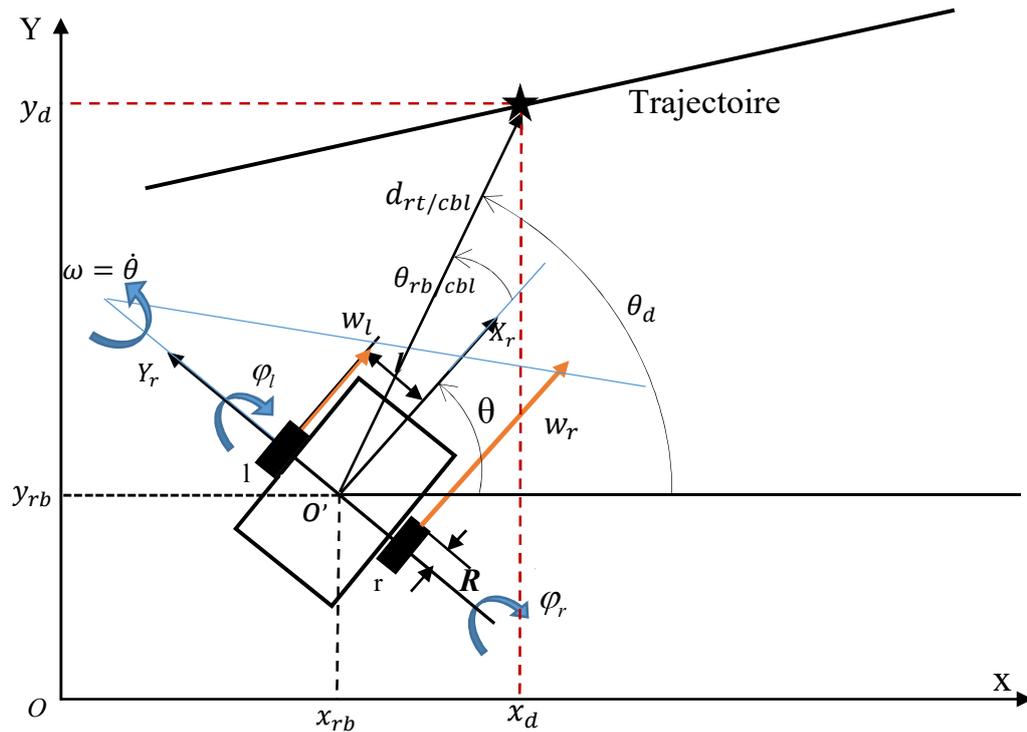


Figure V.1 Principe de poursuite.

V.3 Conception des contrôleurs de poursuite d'une trajectoire

Dans ce qui suit, on va proposer des contrôleurs intelligents pour une mission de poursuite de trajectoire par un robot mobile autonome. Ils sont basés sur des méthodes de conception simples en utilisant des contrôleurs flous de type MAMDANI max-min, neuro-flou de type ANFIS. Ces comportements seront utilisés aussi pour une tâche de poursuite d'une cible mobile par un robot. Le développement est motivé par la simplicité et l'efficacité des structures étudiées pour accomplir cette mission d'une manière autonome.

V.3.1 Conception du Contrôleur flou

Dans ce travail, nous avons choisi la structure la plus utilisée pour la commande des processus, qui est un contrôleur à deux entrées : l'erreur et la variation de l'erreur. La sortie du contrôleur est la commande du processus. Les variables d'entrées sont l'erreur angulaire $\theta_{rb/cbl}$ et sa variation $d\theta_{rb/cbl}$ (figure V.1). Ces entrées sont partitionnées sur 5 ensembles flous triangulaires représentés sur les figures V.3 et V.4. La sortie α partitionnée sur 5 ensembles flous triangulaires représentés sur la figure V.5.

Pour les ensembles flous, la forme triangulaire est la plus utilisée en commande des systèmes. Les valeurs linguistiques des variables floues entrée-sortie sont :

Z(Zéro), PM(Positive Moyenne), PG(Positive Grande), NG(Négative Grande) et NM(Négative Moyenne).

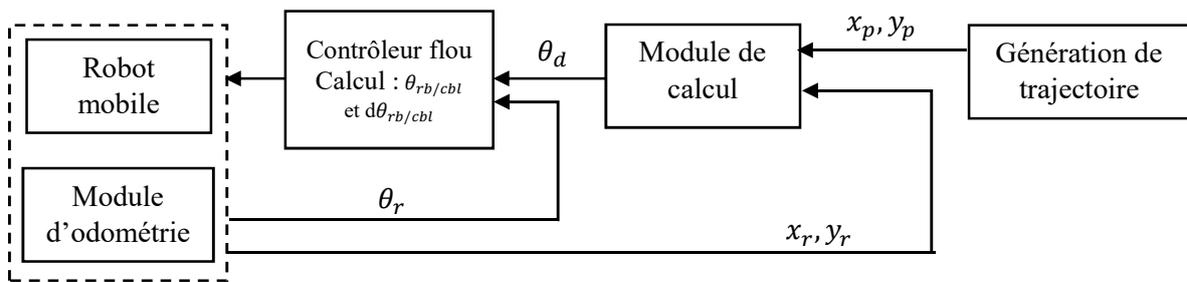


Figure.V.2 Contrôleur flou de poursuite de trajectoire.

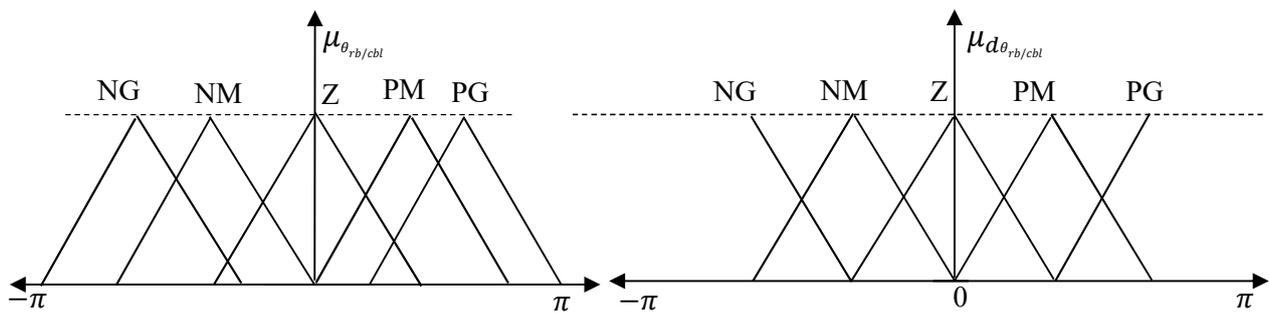


Figure V.3 Fonctions d'appartenance de $\theta_{rb/cbl}$. Figure V.4 Fonctions d'appartenance de $d\theta_{rb/cbl}$.

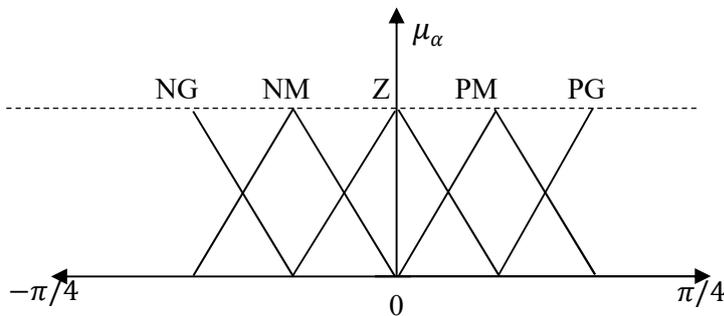


Figure V.5 Fonctions d'appartenance de α .

$\theta_{rb/cbl} \backslash d\theta_{r/c}$	NG	NM	Z	PM	PG
NB	PG	PG	PG	PM	Z
NM	PG	PG	PM	Z	NM
NS	PG	PM	Z	NM	NG
Z	PM	Z	NM	NG	NG
PS	Z	NM	NG	NG	NG

Tableau V.1 Règles floues de poursuite de trajectoire.

V.3.1.1 Résultats de poursuite de trajectoire

Les résultats de simulation pour la poursuite de trajectoire de type droite sont présentés sur les figures V.6-10. La tâche commence par un point initial de la trajectoire et se termine en un point final. D'après ces résultats, on observe que le robot peut suivre les trajectoires de référence avec minimum d'erreurs. Le contrôleur flou agit de manière correcte en générant les commandes appropriées pour cette mission. Dans les premières étapes, le robot exécute l'action appropriée, afin de rattraper le chemin (braquage à droite ou à gauche), puis, lorsque l'erreur angulaire est nulle, il se déplace sur les points de référence en générant un signal de sortie nul (aucun changement de direction selon la règle d'inférence choisie). Pour chaque figure, nous présentons les premiers mouvements exécutés. Les valeurs de commande générées par le contrôleur flou pour chaque situation sont données sur les figures V.7-8.

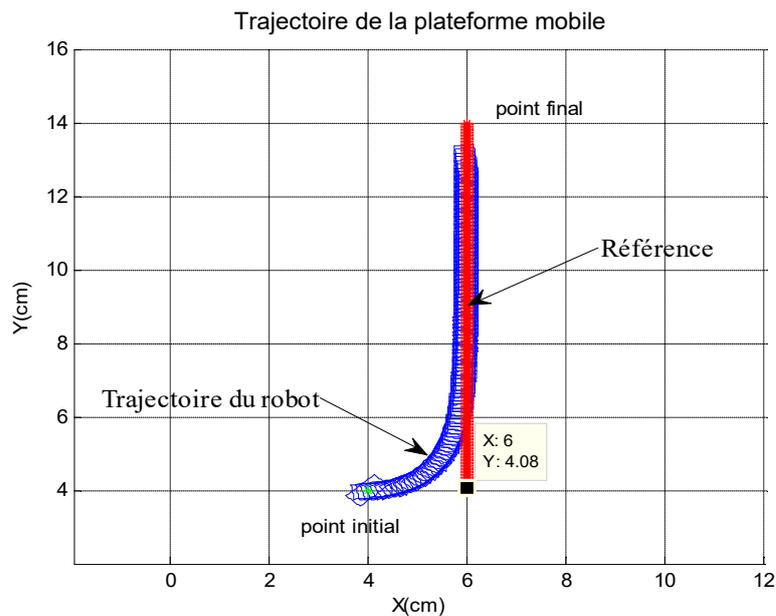


Figure V.6 Poursuite de trajectoire pour le contrôleur flou, les points initiaux (4,4).

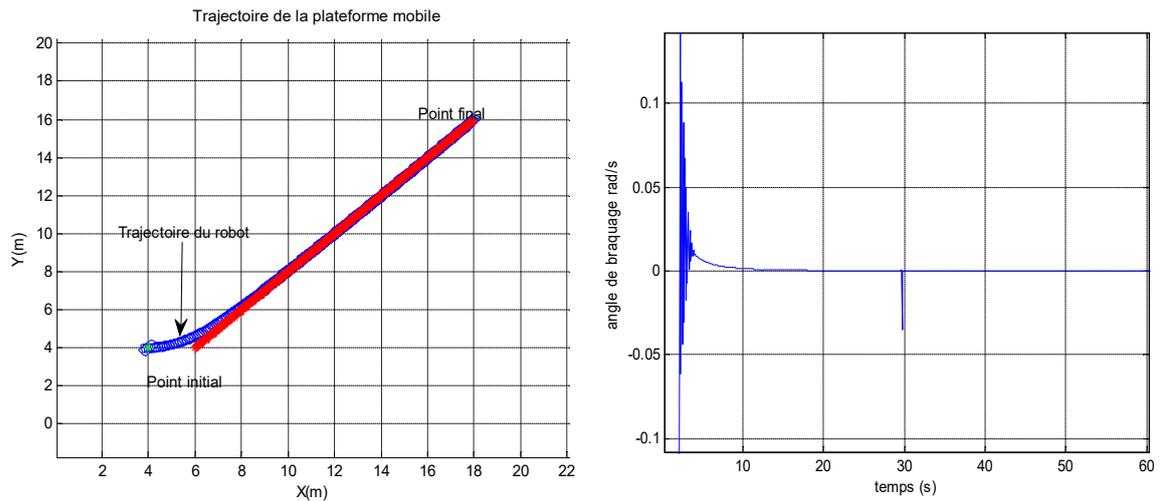


Figure V.7 Poursuite de trajectoire et l'angle de braquage pour le contrôleur flou.

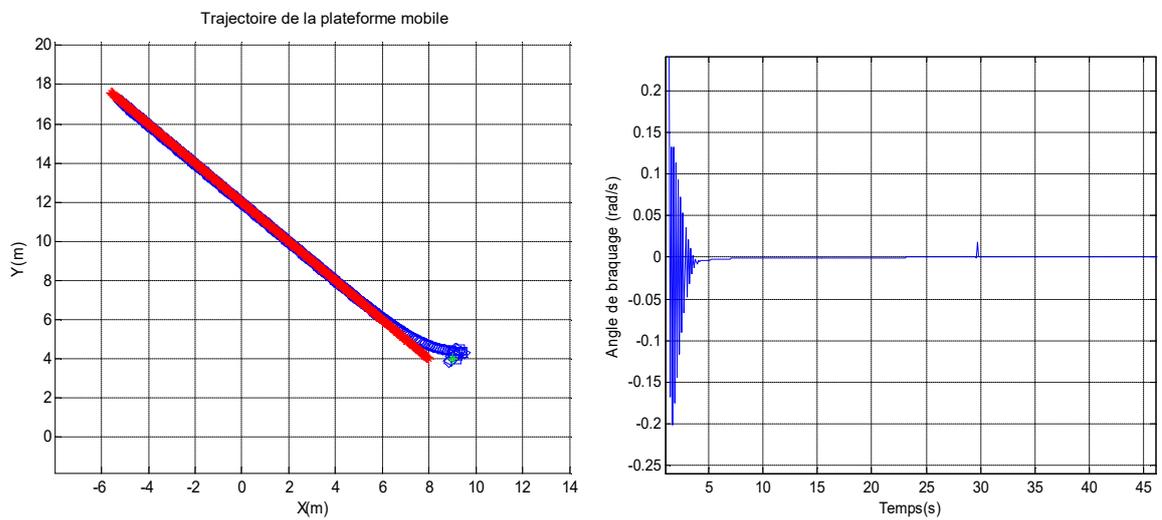


Figure V.8 Poursuite de trajectoire et l'angle de braquage pour le contrôleur flou.

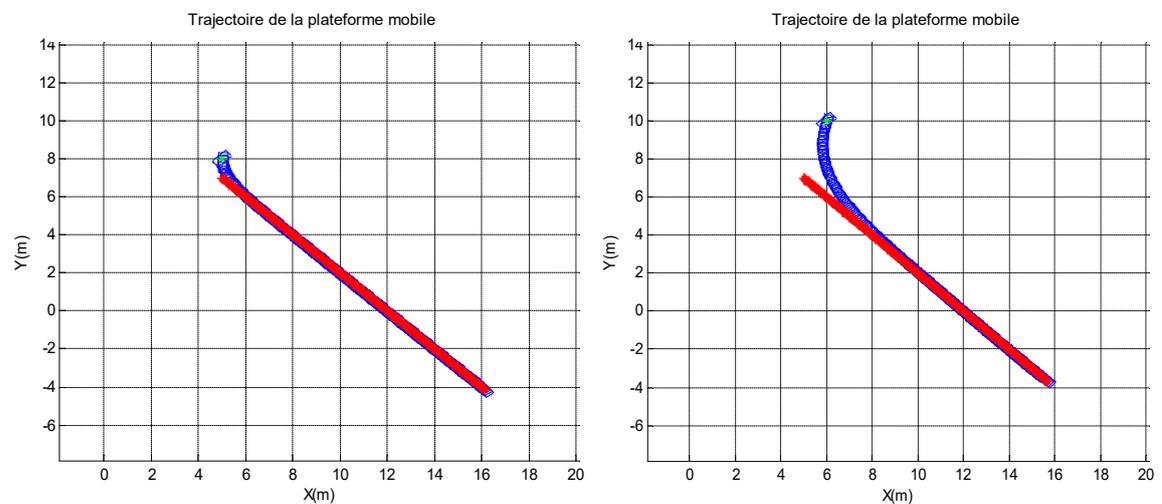


Figure V.9 Poursuite de trajectoire à partir de différentes configurations initiales.

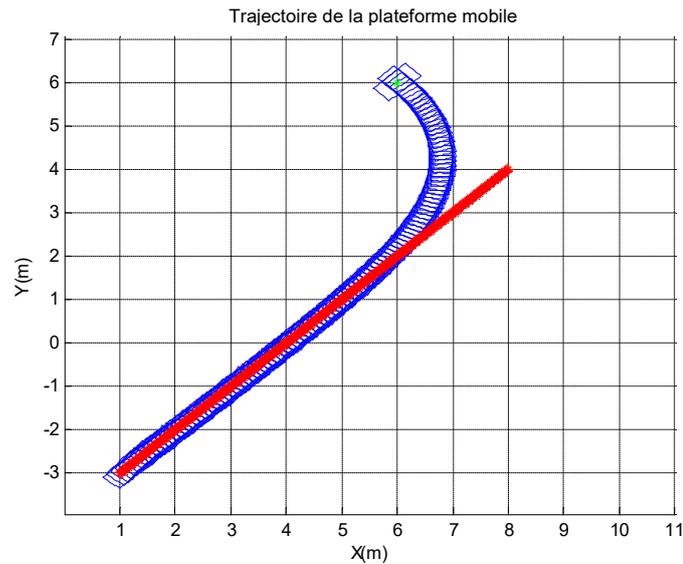


Figure V.10 Poursuite de trajectoire pour le contrôleur flou, les points initiaux (6,6).

V.3.2 Systèmes Neuro-Flous

Les systèmes d'inférences flous (SIFs) et les réseaux de neurones artificiels (RNA) ont chacun des avantages particuliers. Les méthodes hybrides neuronales et floues permettent de tirer les avantages de ces deux approches, principalement la capacité d'apprentissage des RNA et de la lisibilité et la souplesse des SIFs. Plusieurs combinaisons de ces deux méthodes ont été développées dans la littérature [1] [10] [17]. Elles ont données naissance aux systèmes neuro-flous, qui sont le plus souvent orientés vers la commande des systèmes complexes et les problèmes de classification. Il existe généralement quatre grandes catégories de combinaisons des réseaux de neurones avec la logique floue :

- **Réseau flou neuronal** : dans ces réseaux, les techniques floues sont employées pour augmenter les capacités d'apprentissage et d'application des réseaux de neurones.
- **Système neuronal-flou simultanément** : le réseau de neurone et le système flou fonctionnent ensemble pour la même tâche, mais indépendamment, c-à-d ni l'un ni l'autre n'est employé pour déterminer les paramètres de l'autre. Habituellement le réseau neuronal traite les entrées, ou les sorties du système flou.

- **Modèles neuro-flous coopératifs** : le réseau de neurone est employé pour déterminer les paramètres (les règles et les ensembles flous) d'un système flou. Après la phase d'apprentissage, le système flou fonctionne sans le réseau de neurone. C'est une forme simple des systèmes neuro-flous.

- **Modèles neuro-flous hybrides** : les approches neuro-floues modernes sont de cette forme. Un réseau neuronal et un système flou sont combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous, ou comme un système flou sous une forme distribuée parallèle.

V.3.2.1 Conception du contrôleur Neuro-Flou

Il s'agit d'un modèle ANFIS de Takagi-Sugeno d'ordre un, formé par une base de données regroupées pour cette mission [18]. Le contrôleur utilise l'erreur angulaire comme variable d'entrée est entraîné pour générer l'angle de braquage α du robot. Cette commande va permettre de diriger le robot vers le point de la trajectoire de référence. Pour la variable d'entrée, on a utilisé 5 fonctions d'appartenances gaussiennes puisque ce type est le plus adapté au système ANFIS. La structure du contrôleur neuro-flou proposé est donnée sur la figure V.11.

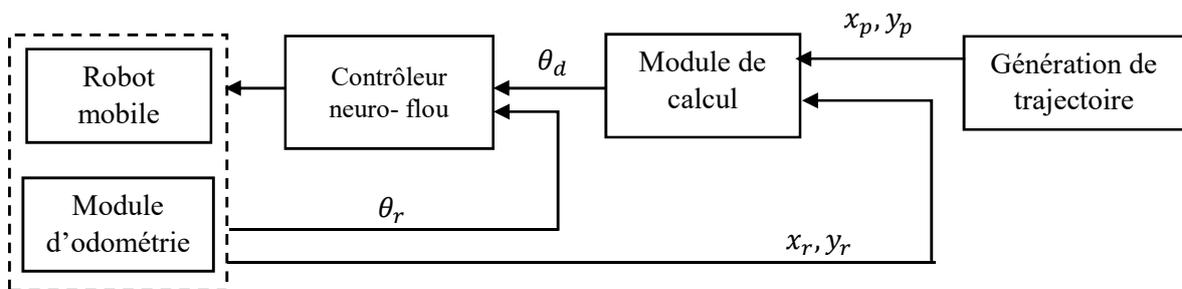


Figure V.11 Contrôleur neuro-flou de poursuite de trajectoire.

V.3.2.2 Résultats de poursuite de trajectoire

Le robot s'oriente dans les premiers pas puis il garde son orientation sur la trajectoire qui est définie par une action de braquage nulle. L'efficacité de ce contrôleur est testée aussi pour différentes trajectoires (figures V.12-13-14). Le contrôleur proposé définit un comportement correct et effectif en exécutant des actions les plus adaptées à la mission

visée. Le robot a tendance à s'éloigner de la trajectoire de référence dans les premières étapes de mouvement.

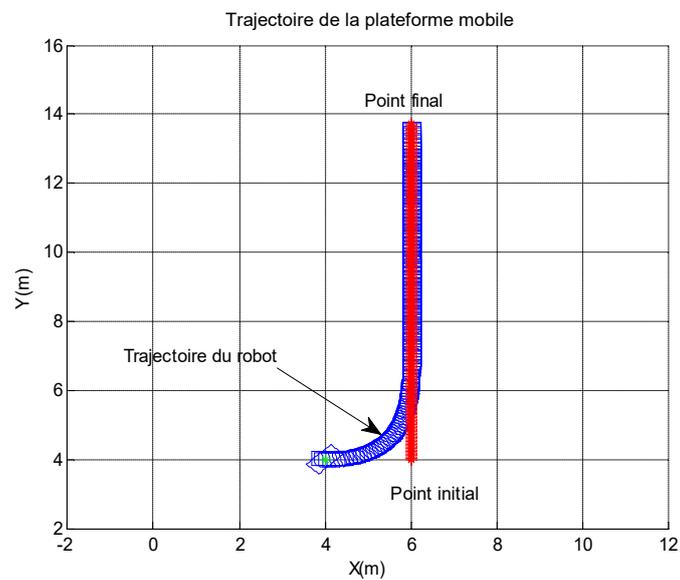


Figure V.12 Poursuite de trajectoire pour le contrôleur neuro-flou, le point initial (4,4).

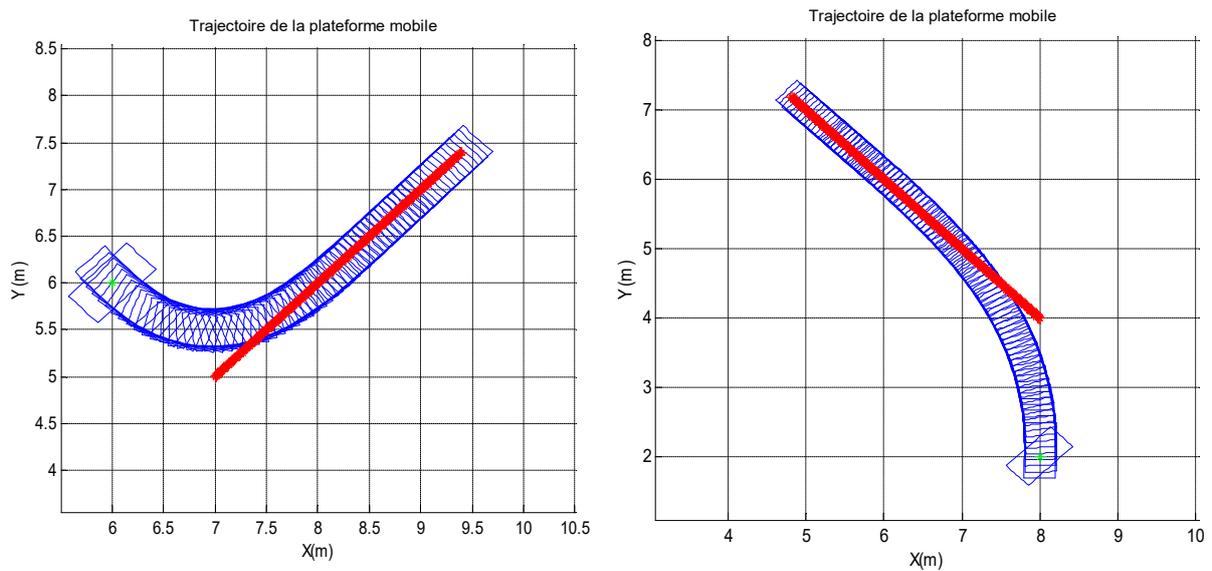


Figure V.13 Poursuite de trajectoire à partir de différentes configurations initiales.

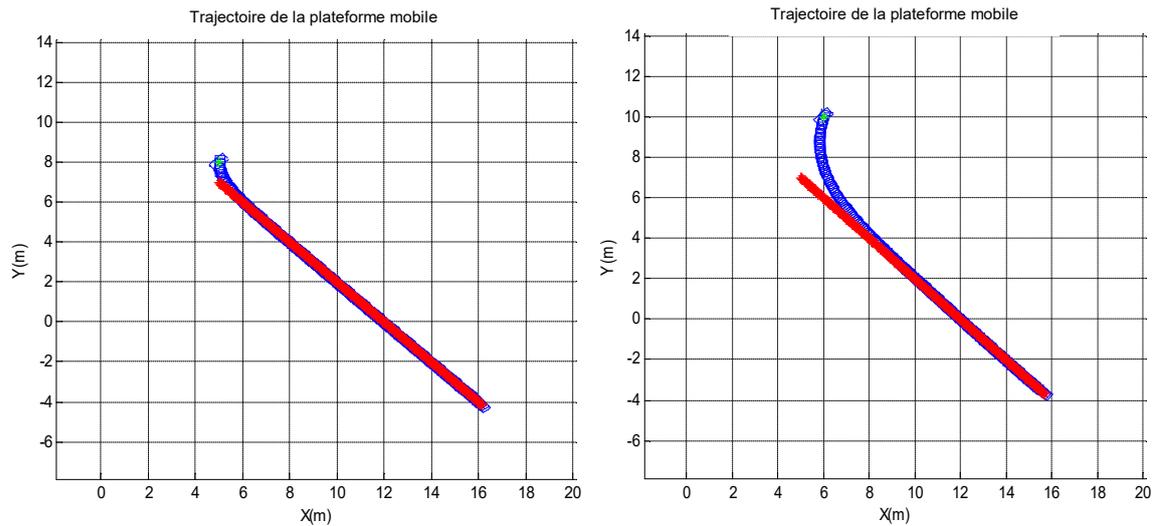


Figure V.14 Poursuite de trajectoire pour le contrôleur neuro-flou.

V.4 Poursuite d'une cible mobile

Dans cette section, nous avons appliqué les contrôleurs précédents (flou, neuro-flou) pour une tâche de poursuite d'une cible mobile. Si on considère une cible en mouvement (objet ou autre robot) dans l'environnement de navigation, la mission du robot est de poursuivre cet objet. La tâche commence avec des positions initiales différentes de la cible et du robot et se termine lorsque le robot atteint cet objet sans entrer en collision avec les obstacles, Alors ses coordonnées sont connues à n'importe quel moment du mouvement. Les systèmes de contrôle proposés doivent guider le robot vers la position de cet objet mobile en générant la commande de correction nécessaire.

V.4.1 Contrôleur flou

Les trajectoires du robot pour une tâche de poursuite d'une cible mobile avec l'évitement d'obstacles de différentes configurations initiales sont représentées dans les figures suivantes :

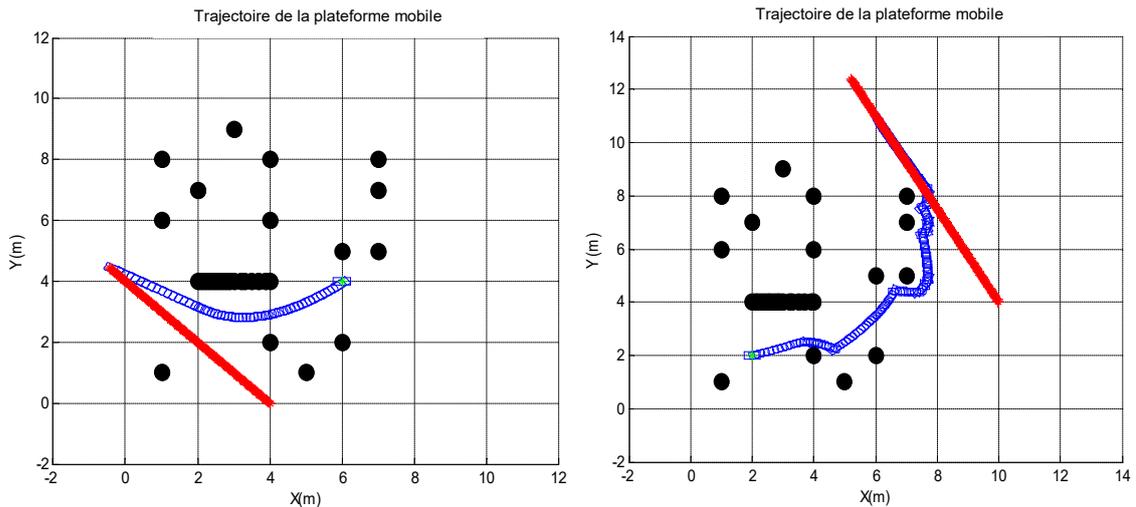


Figure V.17 Poursuite d'une cible mobile en présence d'obstacles pour le contrôleur flou, les points initiaux (6,6).

V.4.2 Contrôleur hybride neuro-flou ANFIS

Dans ce qui suit, nous utiliserons un contrôleur hybride neuro-flou de type ANFIS pour la poursuite d'une cible mobile avec évitement d'obstacles par un robot mobile autonome. Le robot peut suivre les trajectoires sans entrer en collision avec l'obstacle, le robot doit garder une distance de sécurité avec les obstacles. Les trajectoires du robot avec évitement d'obstacles de différentes configurations initiales sont représentées dans les figures suivantes :

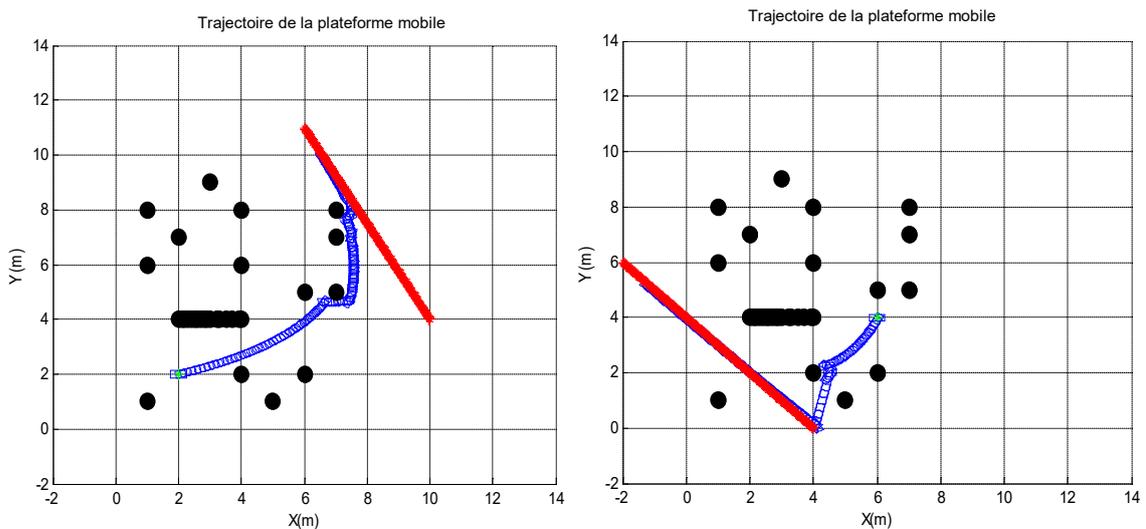


Figure V.18 Poursuite d'une cible mobile en présence d'obstacles pour le contrôleur neuro-flou.

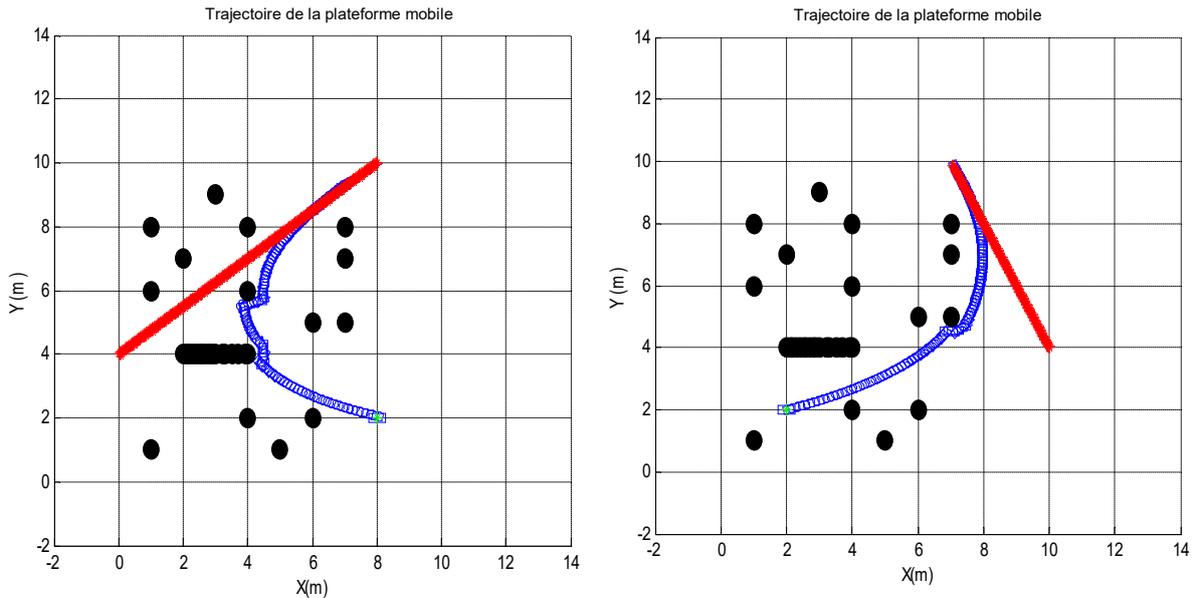
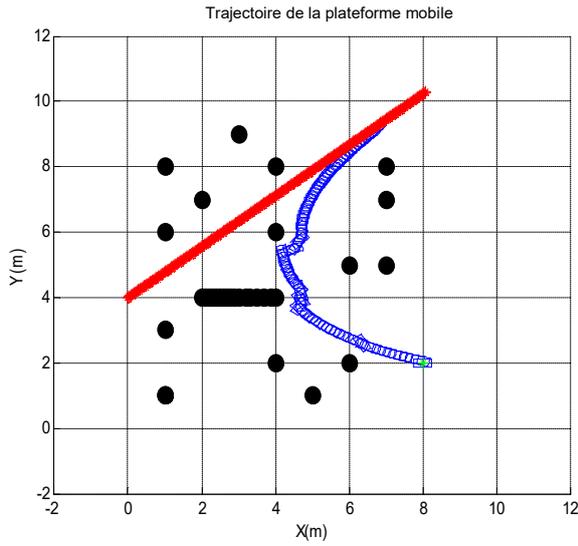


Figure V.19 Poursuite d'une cible mobile en présence d'obstacles pour le contrôleur neuro-flou, le point initial (8,2).

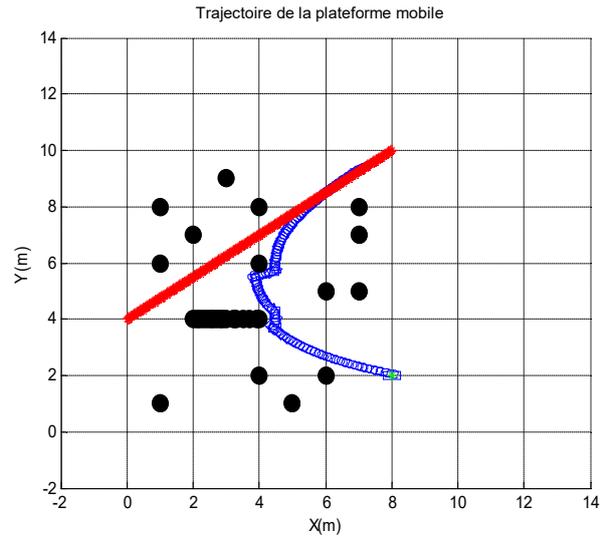
D'après ces résultats, on peut montrer que les contrôleurs proposés sont des solutions acceptables pour cette tâche. Ces contrôleurs peuvent guider toujours le robot vers la direction de la cible mobile.

V.4.3 Comparaison

Dans ce qui suit, on présente une comparaison des résultats de poursuite d'une cible mobile avec évitement d'obstacles en utilisant les deux contrôleurs (flou et neuro-flou) pour différents types des trajectoires. Les figures V.20-21-22 (a-b) illustrent les trajectoires exécutées par le robot en utilisant : (a) le contrôleur flou, (b) le contrôleur de type ANFIS. On a montré que les systèmes de commande utilisés donnent des meilleures performances pour cette fonctionnalité.



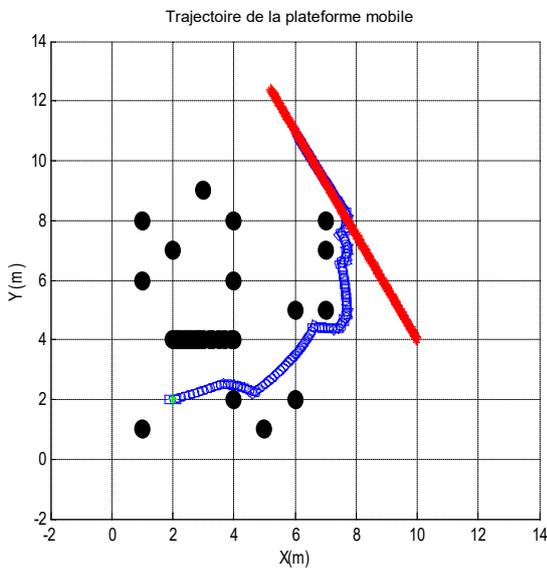
(a)



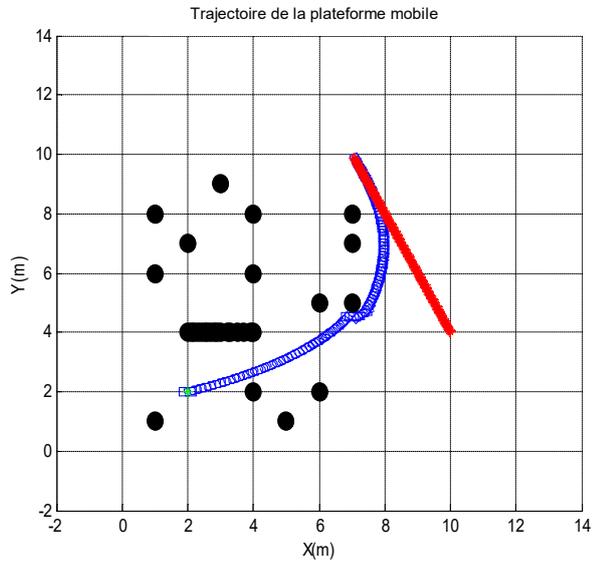
(b)

Figure V.20 Poursuite d'une cible mobile en présence d'obstacles en utilisant :

(a) contrôleur flou, (b) contrôleur neuro-flou le point initial (8,2).



(a)



(b)

Figure V.21 Poursuite d'une cible mobile en présence d'obstacles en utilisant :

(a) contrôleur flou, (b) contrôleur neuro-flou le point initial (2,2).

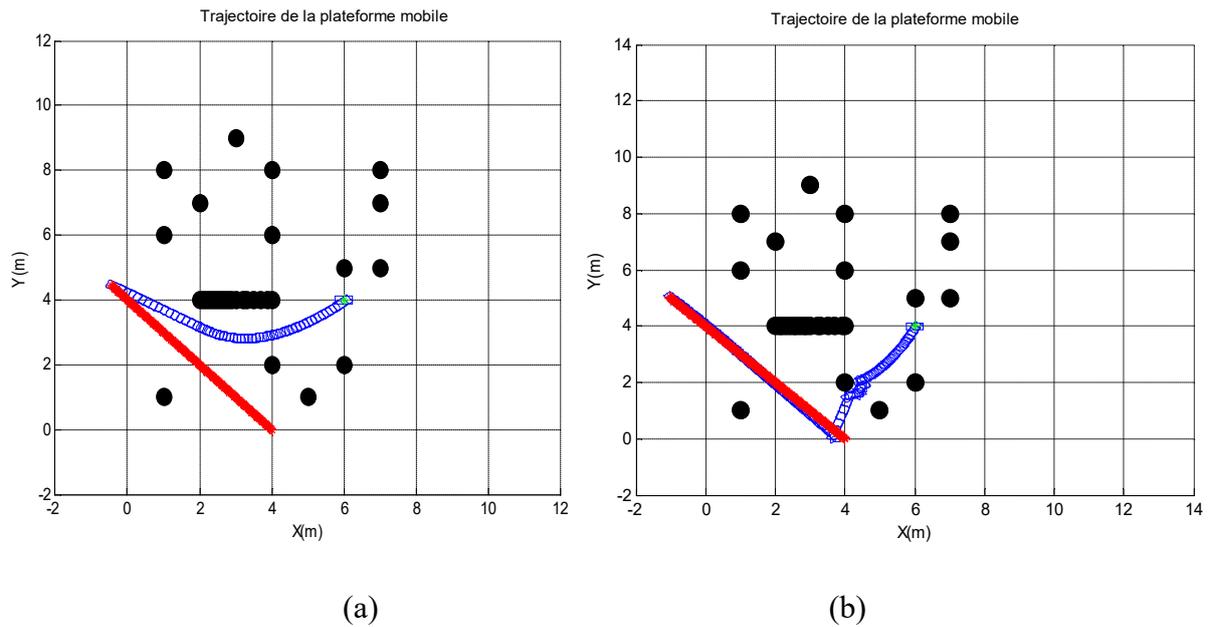


Figure V.22 Poursuite d'une cible mobile en présence d'obstacles en utilisant :
 (a) contrôleur flou, (b) contrôleur neuro-flou le point initial (6,4).

V.5 Conclusion

Dans ce chapitre, nous avons présenté des systèmes de contrôle intelligents basés sur les techniques de l'intelligence artificielle les plus populaires pour la navigation autonome d'un robot mobile. La première technique utilisée est basée sur l'élaboration d'un contrôleur flou de type MAMDANI caractérisé par la capacité d'utiliser les connaissances d'un expert humain, et ne nécessite pas un modèle analytique du processus. Le deuxième est un système hybride neuro-flou de type ANFIS qui intègre les deux propriétés. L'avantage de ce système hybride ANFIS est la génération des règles floues de manière automatique et l'ajustement des paramètres des contrôleurs flous en utilisant une base de données pour l'apprentissage. Ces contrôleurs sont utilisés pour réaliser la tâche de poursuite de trajectoire de référence et d'une cible mobile. Différentes types de trajectoires sont simulées. Les résultats de simulation montrent l'efficacité des contrôleurs proposés pour le contrôle du robot. Ces résultats obtenus présentent une grande similarité entre les approches étudiées. L'avantage des contrôleurs proposés est la simplicité et l'efficacité pour le contrôle du robot mobile.

REFERENCES BIBLIOGRAPHIQUES CHAPITRE V

- [1] J.-S.R Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference System", IEEE Transaction on Systems, Man, and Cybernetics, vol. 23, pp.665-685, 1993.
- [2] J. Godjevac, "Idées Nettes sur la Logique Floue", Collection Informatique, Presses Polytechniques et Universitaires Romandes, Lausanne, 1999.
- [3] H. R. Berenji, P. Khedkar, "Learning and Tuning Fuzzy Logic Controllers Through Reinforcements", IEEE Transactions on Neural Networks, vol. 3, no. 5, pp. 724-740, 1992.
- [4] L. Jouffe, "Fuzzy Inference System Learning by Reinforcement Methods", IEEE Transactions on Systems, Man, and Cybernetics, vol.28, no.3, pp. 338-355, 1998.
- [5] F. Herrera, M. Lozano, J. L. Verdegay, "Tuning Fuzzy Controllers by Genetic Algorithms", International Journal of Approximate Reasoning, vol. 12, pp. 299-315, 1995.
- [6] F. Abdessemed, K. Benmahammed and E. Monacelli, "A Fuzzy-based Reactive Controller for Non holonomic Mobile Robot", Robotics and Autonomous Systems, vol. 47, pp. 31-46, 2004.
- [7] H. Boubertakh, "Contribution à l'Optimisation par Algorithmes Evolutionnaires des Contrôleurs Flous", Thèse de Doctorat en Automatique, ENP, 2009.
- [8] F. Hoffman, "Soft Computing Techniques for the Design of Mobile Robot Behaviors", Information Sciences, vol. 122, pp. 241-258, 2000.
- [9] R. Fuller, "Neural Fuzzy Systems", Abo Academy University, 1995.
- [10] J. Godjevac, "Neuro-Fuzzy Controllers, Design and Application", Collection Meta, Presses Polytechniques et Universitaires Romandes, Lausanne, 1997.
- [11] P. Y. Glorennec, "Algorithmes d'Apprentissage pour Systèmes d'inférence Floue", Edition Hermès, 1999.
- [12] K. Khaldoun, S. N. Al-Din Munaf, "A Neuro-Fuzzy Reasoning System for Mobile Robot Navigation", Jordan Journal of Mechanical and Industrial Engineering, vol. 3, no. 1, pp.77-88, 2009.
- [13] G. Dongbing, H. Huosheng, S. Liber, "Learning Fuzzy Logic Controller For Reactive Robot Behaviours", IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1-6, Japan, 2003.
- [14] Ayari, A. Chatti, "Reactive Control using Behavior Modeling of a Mobile Robot", International Journal of Computers, Communications & Control, vol. 2, no. 3, pp. 217-228, 2007.
- [15] E. Gauthier, "Utilisation des Réseaux de Neurones Artificiel pour la Commande d'un Véhicule Autonome", Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1999.
- [16] A. Benmakhlouf, D. Djarah and A. Louchene, "A Novel Approach for intelligent Control Systems: Application to a path following By a Mobile Robot", International Conference on Electrical Engineering Design and Technology (ICEEDT'07), Hammamet, Tunisia, 2007.

- [17] S. G. Tzafestas, K. D. Blekas, "Hybrid Soft Computing Systems: A Critical Survey with Engineering Applications", National Technical University of Athens, Greece, 1997.
- [18] L. Cherroun, M. Boumehraz, "Path Following Behavior for an Autonomous Mobile Robot using Neuro-Fuzzy Controller", Accepted in International Journal of Systems Assurance Engineering and Management, (IJSA), Springer.

Conclusion Générale

Le travail de cette thèse avait pour but principal la navigation d'un robot mobile dans un environnement quelconque. Après avoir étudié les différentes méthodes utilisées dans le domaine, nous avons axé notre travail sur les points suivants :

Au début, des définitions et des concepts généraux sur l'autonomie du robot mobile ont été présentés. Les types des robots mobiles, les capteurs utilisés et les architectures de contrôles ont été exposés.

Nous avons présenté la théorie de la commande par la logique floue et l'architecture de base d'un contrôleur flou. Le fonctionnement d'un contrôleur flou dépend d'un nombre important de paramètres qu'il faut déterminer afin d'optimiser ses performances. Les contrôleurs flous présentent la possibilité d'incorporer des connaissances expertes dans leurs structures. L'intérêt majeur de la logique floue en commande réside dans sa capacité à traduire une stratégie de contrôle d'un opérateur qualifié en un ensemble de règles linguistiques facilement interprétables.

Par la suite, on a exposé les principes des réseaux de neurones et les systèmes hybride neuro-flou (ANFIS), ces deux méthodes permettent d'explorer de façon très efficace l'espace des solutions possibles pour la commande. En faisant l'état de l'art de ces méthodes, on a constaté que plusieurs variantes de ces algorithmes ont été proposées et appliqués pour la commande des robots mobiles.

Nous avons élaboré une méthode de contrôle de déplacement d'une plateforme mobile à roue basée sur la commande floue qui nous a permis de concevoir un contrôleur flou, pour simuler le déplacement du robot mobile d'une position initiale vers une position désirée dans des différentes configurations d'un environnement avec et sans obstacles. L'efficacité de notre contrôleur a été testée sur une plateforme mobile de type unicycle. La commande floue proposée exploite les quatre variables interactives entre le robot mobile et l'environnement inconnu pour générer la vitesse et le braquage du

robot qui permet d'amener le robot mobile vers la cible en évitant les éventuels obstacles qui peuvent occuper cet environnement. Le contrôleur flou est conçu de telle manière que lorsque la plateforme mobile se déplace en ligne droite vers la cible et un obstacle surgit, une stratégie d'évitement d'obstacles est vite introduite durant un laps de temps et le robot mobile exécute une immédiate manœuvrabilité ordonnée par le contrôleur flou autour de cet obstacle.

Nous avons également utilisé des contrôleurs flous, hybride neuro-flou pour accomplir une fonctionnalité particulière du robot mobile, qui est la poursuite de trajectoire de référence et de cible mobile. Les comportements obtenus sont très satisfaisants à partir des structures proposées et des résultats présentés pour cette tâche. Ces contrôleurs agissent d'une manière efficace, mais l'inconvénient majeur des réseaux de neurones est la nécessité des paires état-action définissant la base de donnée pour l'apprentissage en particulier pour les autres comportements (convergence vers un but, évitement d'obstacles, suivi de murs,..).

Des exemples de simulation sont fournis pour montrer les performances des systèmes de commande proposés pour les différents comportements du robot mobile. Une comparaison entre les navigateurs, flous et hybrides neuro-flous a été présentée. Les résultats obtenus montrent que les structures étudiées avec des meilleures performances pour la navigation autonome d'un robot mobile.

Enfin, nous pensons que le travail présenté dans cette thèse ouvre des nouvelles perspectives selon les principales directions suivantes :

- La mise en œuvre pratique des méthodes étudiées sur un robot mobile réel,
- L'application des architectures présentées pour des environnements avec des obstacles dynamiques,
- La comparaison des méthodes développées avec d'autres algorithmes stochastiques tels que les algorithmes génétiques,
- L'optimisation des contrôleurs flous par les algorithmes de colonies de fourmis.

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

ملخص: في هذه الأطروحة، يتم التعامل مع مشكلة الملاحة للروبوت المتحرك من خلال أنظمة الاستدلال الضبابي، والشبكات العصبية الاصطناعية. الهدف من هذا العمل هو دراسة وتطوير بني تحكم فعالة للملاحة التفاعلية لروبوت متحرك مستقل في بيئة غير معروفة، باستخدام النهج السلوكي. في البداية، استخدمنا، أنظمة قائمة على السلوكيات الغامضة للتنقل المستقل للروبوت المحمول، لتحقيق هذا الهدف قمنا بتصميم وحدة تحكم غامضة، لمحاكاة حركة الروبوت المحمول من الموضع الأولي إلى الموضع المطلوب في تكوينات مختلفة بيئة مع وبدون عقبات. تم تصميم وحدة التحكم الضبابية بطريقة أنه عندما تتحرك المنصة المتنقلة نحو الهدف وتنشأ عقبة، يتم تقديم سلوك تجنب العوائق بسرعة خلال فترة زمنية ويقوم الروبوت المتحرك بأداء قدرة فورية على المناورة أمرت بها وحدة التحكم في التموه حول هذه العقبة. ثم لتحسين أداء الروبوت المحمول، سيتم استخدام وحدة تحكم هجينة عصبية ضبابية من نوع ANFIS في مهمة متابعة المسار. سيتم أيضاً تطبيق وحدتي التحكم المقترحتين لمهمة تتبع هدف متحرك. يتم توضيح فعالية البنى المقترحة والمدرسة من خلال العديد من تطبيقات الملاحة المستقلة للروبوت المتحرك.

كلمات مفتاحية: روبوت متحرك، نظام ضبابي عصبي، أنظمة استدلال ضبابية، ملاحة، تجنب لعقبات، بيئة غير معروفة.

Résumé : Dans cette thèse, le problème de navigation d'un robot mobile est traité par les systèmes d'inférence flous, les réseaux de neurones artificiels. L'objectif de ce travail est d'étudier et développer des architectures de commande efficaces pour une navigation réactive d'un robot mobile autonome dans un environnement inconnu, en utilisant l'approche comportementale. Au début, On a utilisé, les systèmes basés sur les comportements flous pour la navigation autonome d'un robot mobile. Nous avons conçu un contrôleur flou afin de simuler le déplacement du robot mobile d'une position initiale vers une position désirée dans des différentes configurations d'un environnement avec et sans obstacles. Le contrôleur flou est conçu de telle manière que lorsque la plateforme mobile se déplace vers la cible et un obstacle surgit, un comportement d'évitement d'obstacles est vite introduit durant un laps de temps et le robot mobile exécute une immédiate manœuvrabilité ordonnée par le contrôleur flou autour de cet obstacle. Pour améliorer les performances de la commande du robot mobile, un contrôleur hybride neuro-flou de type ANFIS est aussi utilisé pour une mission de poursuite de trajectoire. Les deux contrôleurs proposés seront également appliqués pour la tâche de poursuite d'une cible. L'efficacité des architectures proposées et étudiées sont démontrées par diverses applications de navigation autonome d'un robot mobile.

Mots-clés : Robot mobile, Systèmes Neuro-Floue, Systèmes d'Inférence Flous, Navigation, Evitement d'Obstacle, Environnement Inconnu.

Abstract: In this thesis, the navigation problem of a mobile robot is dealt with by fuzzy inference systems, artificial neural networks. The objective of this work is to study and develop efficient control architectures for reactive navigation of an autonomous mobile robot in an unknown environment, using the behavioral approach. At the beginning, we used, systems based on fuzzy behaviors for the autonomous navigation of a mobile robot, to meet this objective we designed a fuzzy controller, to simulate the movement of the mobile robot from an initial position to a desired position in different configurations of an environment with and without obstacles. The fuzzy controller is designed in such a way that when the mobile platform moves towards the target and an obstacle arises, obstacle avoidance behavior is quickly introduced during a period of time and the mobile robot performs immediate maneuverability ordered by the blur controller around this obstacle. Then To improve the performance of the mobile robot, an ANFIS type neuro-fuzzy hybrid controller will be used for a trajectory pursuit mission. The two proposed controllers will also be applied for the task of tracking a target. The effectiveness of the proposed and studied architectures are demonstrated by various autonomous navigation applications of a mobile robot.

Keywords: Mobile Robot, Neuro-Fuzzy Systems, Fuzzy Inference Systems, Navigation, Obstacle Avoidance, Unknown Environment.