

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider - Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique



N° d'ordre :.....

Série :.....

THÈSE

Présentée pour obtenir le diplôme de **Doctorat en Sciences**

Spécialité: **AUTOMATIQUE**

Par:

NADOUR Mohamed

Thème :

Navigation Visuelle d'un Robot Mobile

Soutenu publiquement le : 14 / 10 / 2020 devant les membres de Jury :

OUAFI Abdelkarim	Président	<i>Professeur</i>	Université de Biskra
BOUMEHRAZ Mohamed	Rapporteur	<i>Professeur</i>	Université de Biskra
CHERROUN Lakhmissi	Co-Encadreur	<i>Maitre de Conférences</i>	Université de Djelfa
HAFIFA Ahmed	Examineur	<i>Professeur</i>	Université de Djelfa
TOUBA Med Mustapha	Examineur	<i>Maitre de Conférences</i>	Université de Biskra
KOUZOU Abdellah	Examineur	<i>Professeur</i>	Université de Djelfa

Remerciements

Tout d'abord, je remercie le **Dieu** de m'avoir donné la volonté et le courage qui m'ont permis de réaliser ce travail.

Je remercie vivement mon encadreur **Pr. Mohamed BOUMEHRAZ**, Professeur à l'université de Biskra, d'avoir accepté de diriger ce travail. Je le remercie pour sa disponibilité totale et ses encouragements dans toutes les phases du travail, qu'il trouve ici ma grande reconnaissance.

Je me ferais un agréable devoir de remercier mon co-encadreur **Dr. Lakhmissi CHERROUN**, Maître de conférences à l'université de Djelfa pour m'avoir dirigé ce travail de recherche et m'avoir fait bénéficier de son expérience et de ses précieux conseils.

Je remercie toutes les personnes qui m'ont l'honneur d'accepter de participer au jury de thèse.

Monsieur **OUAFI Abdelkarim**, Professeur à l'université de Biskra, pour l'honneur qu'il me fait en présidant ce jury.

Monsieur **HAFIFA Ahmed**, Professeur à l'université de Djelfa, pour avoir accepté de faire partie de mon jury et d'examiner ce travail.

Monsieur **TOUBA Med Mustapha**, Maître de conférences à l'université de Biskra, pour avoir voulu juger ce travail.

Monsieur **KOUZOU Abdellah**, Professeur à l'université de Djelfa, pour l'intérêt qu'il a accordé à mon travail en acceptant de l'examiner.

Que tous mes amis et collègues trouvent ici le témoignage de mon amitié et de ma reconnaissance.

Je remercie sincèrement toutes les personnes qui, de près, ou de loin, m'ont permis de terminer ce travail dans bonnes conditions.

Dédicace

Je dédie ce Travail:

A mes Parents, que le Dieu me les gardes,

A ma Femme et ma Fille Alaa El rahman,

A mes Frères et mes Sœurs,

A tout les membres de ma Grande Famille

....

Table des matières

Liste des Figures

Liste des Tableaux

Introduction Générale..... 01

Chapitre I: Etat de l'art sur la Navigation Visuelle des Robots Mobiles

I.1 Introduction	05
I.2 Navigation autonome des robots mobiles.....	05
I.2.1 Planification de mouvement.....	06
I.2.2 Localisation	06
I.2.3 Suivi de trajectoire	06
I.2.4 Évitement d'obstacles.....	07
I.2.5 Parking.....	07
I.3 Les capteurs en Robotique.....	07
I.3.1 Les capteurs proprioceptifs	08
I.3.2 Les capteurs extéroceptifs	08
I.4 Navigation visuelle d'un robot mobile.....	10
I.4.1 Classification des approches de navigation visuelle.....	11
I.4.1.1 Navigation visuelle en milieu intérieur.....	12
I.4.1.2 Navigation visuelle en milieu extérieur.....	13
I.5 Flux Optique (FO).....	14
I.5.1 Définition du Flux Optique (FO)	15
I.5.2 Exemples d'applications.....	16
I.5.3 Propriétés du flux optique.....	17
I.5.3.1 Formation du flux optique.....	17
I.5.3.2 Foyer d'expansion (FOE)	19
I.5.3.3 Estimation du temps avant collision	20
I.5.4 Calcul du flux optique.....	21
I.5.4.1 Les méthodes différentielles.....	21
I.5.4.3 Équations du flux optique	22
I.7 Conclusion	25

Chapitre II: Systèmes Flous Type-1 et Type-2

II.1 Introduction	26
II.2 Systèmes Flous type-1 (SF1)	26
II.2.1 Ensembles flous type-1 (EF1).....	26
II.2.2 Type des ensembles flous	27
II.2.3 Variable Floue.....	28

II.2.4 Opérations sur les ensembles flous.....	29
II.2.5 Raisonnement Flou	30
II.2.6 Structure des Systèmes Flous (SF1).....	31
II.2.7 Types des Systèmes Flous type-1.....	32
II.2.7.1 Méthode de Mamdani.....	32
II.2.7.2 Méthode de Takagi-Sugeno.....	32
II.2.8 Propriété d'Approximation Universelle.....	32
II.3 Systèmes Flous type-2 (SF2).....	33
II.3.1 Ensembles flous type-2 (EF2).....	33
II.3.2 Type des ensembles flous type-2.....	35
II.3.3 Principe d'extension monodimensionnel.....	35
II.3.4 Structure des Systèmes flous type-2.....	36
II.3.4.1 Fuzzification.....	37
II.3.4.2 Base de règles.....	37
II.3.4.3 Mécanisme d'inférence.....	38
II.3.4.4 Traitement de la sortie.....	39
II.4 Conclusion	43

Chapitre III: Navigation Visuelle par des Contrôleurs Flous type-1 et le Flux Optique

III.1 Introduction.....	44
III.2 Travaux liés à la contribution.....	45
III.3 Structures de Navigation basée sur les comportements.....	46
III.4 Modélisation par VRML.....	48
III.4.1 Modélisation de l'environnement (Virtual Reality 3D Model).....	48
III.4.2 Modèle du robot mobile.....	50
III.5 Système de navigation proposé.....	52
III.5.1 Contrôleur d'évitement des obstacles.....	53
III.5.2 Contrôleur de convergence vers un but.....	57
III.6 Résultats et Discussions.....	60
III.6.1 Tache d'évitement des obstacles.....	60
III.6. 2 Tache de Recherche de but (convergence vers un but).....	67
III.6. 3 Tache de convergence vers un but avec évitement d'obstacles.....	70
III.6. 4 Tache de poursuite d'une cible mobile avec évitement d'obstacles.....	72
III.7 Conclusion.....	74

Chapitre IV: Navigation Visuelle par des Contrôleurs Flous type-2 et le Flux Optique

IV.1 Introduction	76
IV.2 Travaux liés à la contribution	76
IV.3 Aperçu sur les contrôleurs flous type-2	76
IV.4 Aperçu sur l'algorithme Horn-Schunck.....	78
IV.5 Contrôleur flou type-2 pour l'évitement des obstacles.....	79
IV.6 Résultats de simulation	82
IV.6.1 Tâche d'évitement des obstacles.....	82
IV.7 Comparaison entre FLC de type-1 et FLC de type-2.....	88
IV.8 Conclusion	90
Conclusion Générale	91

Références Bibliographiques

Liste des figures

Figure I.1	les capteurs en robotique.....	07
Figure I.2	Schéma synoptique d'un télémètre à ultrasons.....	09
Figure I.3	Caméra panoramique.....	09
Figure I.4	Approches visuelle pour la navigation autonome	12
Figure I.5	Carte topologique.....	13
Figure I.6	Exemple de carte reconstruction créée par corrélation de scans.....	13
Figure I.7	Exemples des environnements non structurés.....	14
Figure I.8	Exemple de flux optique pour un mouvement de translation (X, Y et Z).....	16
Figure I.9	Exemple de flux optique.....	17
Figure I.10	Illustration du plan image de caméra.....	18
Figure I.11	Représentation du foyer d'expansion.....	19
Figure I.12	Temps Avant Collision.....	20
Figure II.1	Ensemble classique et ensemble flou.....	27
Figure II.2	Types des ensembles flous.....	28
Figure II.3	Représentation de l'erreur par cinq termes linguistiques.....	28
Figure II.4	Structure d'un système flou type-1.....	31
Figure II.5	Ensemble flou gaussien type-2 et son appartenance secondaire.....	34
Figure II.6	Ensemble flou type-2 obtenue à partir d'un ensemble flou type-1 : (a) variance incertaine, (b) moyenne incertaine.....	34
Figure II.7	Structure de base d'un système flou de type-2.....	37
Figure III.1	Structure subsumption.....	47
Figure III.2	Architecture basée sur les comportements.....	47
Figure III.3	L'interface de l'éditeur VRML.....	48
Figure III.4	L'environnement obtenu en utilisant VRML.....	49
Figure III.5	Robot mobile à roues simulé.....	50
Figure III.6	Modèle du robot mobile.....	52
Figure III.7	Navigateur d'évitement des obstacles.....	53
Figure III.8	Image captée.....	54
Figure III.9	Flux optique et FOE.....	54
Figure III.10	Division de l'image.....	54
Figure III.11	Fonctions d'appartenance de Er	56
Figure III.12	Fonctions d'appartenance de la variation de l'erreur ΔEr	56
Figure III.13	Valeurs de la commande de sortie.....	56
Figure III.14	Structure de contrôle globale.....	58
Figure III.15	Fonctions d'appartenance de la distance d_{rg}	58

Figure III.16	Fonctions d'appartenance de l'angle θ_{rg}	58
Figure III.17	Valeurs de la direction alpha.....	59
Figure III.18	Valeurs de la vitesse du robot.....	59
Figure III.19	Environnement simulé en 3D.....	60
Figure III.20	Positions au cours de navigation visuelle du robot mobile.....	64
Figure III.21	Trajectoire parcourue en 2D.....	65
Figure III.25	Exemple 2 de navigation du robot.....	65
Figure III.22	Flux optiques F_{gauche} et F_{droite}	66
Figure III.23	Erreur de flux.....	66
Figure III.24	Temps Avant Collision.....	66
Figure III.26	Scène du robot au point initial, au milieu et au point final.....	68
Figure III.27	Trajectoire du robot pour la convergence vers un but.....	69
Figure III.28	(a) Angle de braquage, (b) vitesse de translation et (b) la distance.....	69
Figure III.29	Images de la navigation du robot (a-b-c-d).....	71
Figure III.30	Trajectoire du robot pour la recherche d'objectifs avec évitement d'obstacles.....	72
Figure III.31	Scènes du robot pour la poursuite en mouvement avec évitement des obstacles	73
Figure III.32	Chemin du robot en 2D.....	74
Figure IV.1	Navigateur type-2 visuel pour l'évitement des obstacles	80
Figure IV.2	Fonctions d'appartenance de l'erreur Er	80
Figure IV.3	Fonctions d'appartenance de la variation d'erreur ΔEr	80
Figure IV.4	Fonction d'appartenance de α	81
Figure IV.5	Positions du robot en 3D et en 2D	86
Figure IV.6	Différentes positions du Robot en utilisant le CF-1.....	89
Figure IV.7	Différentes positions du Robot en utilisant le CF-2.....	89
Figure IV.8	Représentation des deux trajectoires du robot en utilisant CF-1 (rouge) et CF-2 (noire).....	90

Liste des Tableaux

Tableau III.1	Base de règle floue pour l'évitement d'obstacle	57
Tableau III.2	Règles floues du controleur de la convergence vers un but.....	59
Tableau IV.1	Règles floues type-2 pour l'évitement des obstacles	81

Symboles et Abréviations:

<i>FO</i>	Flux Optique
<i>2D et 3 D</i>	Deux et trios Dimensions
<i>LF</i>	Logique Floue
<i>SF1 et SF2</i>	Systèmes Flous type-1 et type-2
<i>EFT-1 et EFT-2</i>	Ensembles Flous type-1 et type-2
<i>TS</i>	Takagi-Sugeno
<i>FOE</i>	Foyer Of Expansion (foyer d'expansion)
<i>TTC</i>	Time To Contact (Temps Avant Collision)
<i>HS</i>	Algorithme de Horn et Schunck
<i>VRML</i>	Virtual Reality Modeling Language
(x_r, y_r)	Coordonnées du robot mobile
θ_r	Angle d'orientation du robot mobile
d_{rg}	Distance entre le robot et le but
θ_{rg}	Angle entre l'orientation actuelle du robot et celle de la cible
<i>D</i>	Largeur du robot
α	Angle de braquage
V_r	Vitesse linéaire de déplacement
$\mu_{A_x}(x)$	Degré d'appartenance de la variable x
(u, v)	Vecteur de vitesse du pixel
$I(x, y, t)$	Intensité lumineuse de l'image
v	Vitesse linéaire de translation du robot
<i>R</i>	Rayon de la plate-forme du robot
<i>CIR</i>	Centre Instantané de Rotation
$v_r, v_l :$	Vitesses des roues Droite et Gauche
w	Vitesse angulaire

Liste des Travaux:

Publications Internationales:

1. **M. Nadour**, M. Boumehraz, L. Cherroun and V. Puig, "Mobile Robot Visual Navigation based on Fuzzy Logic and Optical Flow Approaches", *International Journal of Systems Assurance Engineering and Management, (IJSA)*, Springer-Verlag, vol. 10, no. 6, pp. 1654–1667, 2019.
2. **M. Nadour**, M. Boumehraz, L. Cherroun and V. Puig, "Hybrid Type-2 Fuzzy Logic Obstacle Avoidance System based on Horn-Schunck Method", *Electrotehnică, Electronică, Automatică (EEA)*, vol. 67, no. 3, pp. 45-51, 2019.
3. L. Cherroun, **M. Nadour**, M. Boudiaf and A. Kouzou, "Comparison between Type-1 and Type-2 Takagi-Sugeno Fuzzy Logic Controllers for Robot Design", *Electrotehnică, Electronică, Automatică (EEA)*, vol. 66, no. 2, pp. 94-103, 2018.

Communications Internationales:

1. **M. Nadour**, L. Cherroun and M. Boumehraz, "Using Optical Flow and VRML to Visual Navigation of a Mobile Robot", *The Second International Conference on Applied Automatic and Industrial Diagnostics (ICAAID'17)*, September 2017, Djelfa, Algeria.
2. L. Cherroun, **M. Nadour** and A. Kouzou, "Type-1 and Type-2 Fuzzy Logic Controllers to Autonomous Robotic Motion", *IEEE 3rd International Conference on Applied Automatic and Industrial Diagnostics (ICAAID'19)*, September 2019, Turkey.

Introduction Générale

La navigation autonome des robots avec évitement d'obstacles est une tâche importante pour l'utilisation correcte des robots mobiles dans de nombreuses applications telles que les systèmes de surveillance, la robotique industrielle, assistée et de service [01, 02, 03]. La navigation du robot est définie comme la tâche de calcul des séquences de mouvements permettant au robot de se déplacer dans l'espace du travail sans intervention humaine [04]. Dans des environnements complexes et encombrés, le système de control a besoin d'un certain degré d'intelligence pour contrôler le robot mobile sans collision avec les objets existant dans l'environnement [28].

Dans les applications robotiques, différents capteurs ont été utilisés pour accomplir les tâches de navigation du robot (*localisation, interception, évitement d'obstacles ou navigation*) [05, 06]. La caméra est un capteur de vision très puissant utilisé en robotique. Il imite la vision des êtres vivants pour définir les propriétés visuelles de l'environnement comme les couleurs et les formes. La vision par ordinateur est un outil intéressant permettant des mesures et une détection de l'environnement sans contact avec les objets. Les applications les plus connues sont l'enregistrement et l'amélioration des images, l'appariement pour la vision stéréoscopique, la reconnaissance de formes, l'estimation et l'analyse du mouvement [07]. L'avantage de la vision dans les applications robotiques est de faire fonctionner et interagir les robots dans leurs environnements avec plus de précision et d'intelligence.

Différentes méthodes ont été proposées pour résoudre le problème de navigation du robot. La méthode de navigation visuelle est l'une des approches les plus importantes et les plus utilisées dans le domaine de la robotique [8-10]. Les stratégies de vision peuvent augmenter l'utilité des robots mobiles dans différents domaines [02, 11, 12]. Pour utiliser la vision en robotique, deux méthodes principales peuvent être employées ; la première basée sur le calcul du mouvement apparent des objets, tandis que la seconde repose sur l'apparence de chaque pixel dans l'image [02].

Les méthodes de navigation visuelle ont été appliquées dans des environnements robotisés intérieurs et extérieurs [07, 13, 14, 30, 61], pour les robots mobiles, les véhicules aériens et les véhicules sous-marins autonomes.

Le flux optique est un outil important pour la navigation basée sur la vision des robots mobiles [18, 20]. Cette approche a été utilisée principalement dans l'estimation du mouvement, la détection et l'évitement des obstacles [02, 17, 18, 20].

Cependant, l'utilisation de l'approche de flux optique pour la navigation des robots mobiles consiste à générer des actions de contrôle discrètes. La logique floue est l'une des approches les plus courantes pour résoudre cette limitation [67]. La logique floue permet d'atteindre de meilleures performances avec des actions continues. Le principal avantage du contrôle flou visuel est l'introduction de l'expérience humaine sous la forme de règles " *Si - Alors*" sur la méthode de contrôle du système en fonction des variables utilisées.

Deux raisons principales amènent à utiliser les systèmes flous comme contrôleurs des systèmes dynamiques. Premièrement, ce type de systèmes flous est caractérisé par 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 [92].

Différents systèmes de navigation visuelle, utilisant les systèmes flous, ont été proposés. Par exemple : dans le papier [22], une implémentation de deux contrôleurs à logique floue parallèles est proposée pour une vision panoramique et inclinée montée sur un véhicule aérien sans pilote (UAV, Unmanned Aerial Vehicle, en anglais). Les contrôleurs conçus présentent de bonnes performances en vol réel pour les objets statiques. Alors que dans [09], avec la même idée, les auteurs ont proposé une approche de contrôle flou basé sur la vision pour la navigation autonome des drones. Le système utilise une caméra RGB (Red, Green and Blue, en anglais) avec un logiciel spécifique pour accomplir les différentes tâches dans des expériences réelles (atterrissage autonome, suivi/inspection d'objets et éviter les collisions).

Dans le travail présenté dans [23], des comportements primitifs basés sur des méthodes de navigation visuelle floue sont proposés pour les robots humanoïdes (marche autonome, poursuite d'une cible, évitement d'obstacles et reconnaissance de marqueurs).

Pour les applications de robots mobiles, une approche bio-inspirée pour l'interprétation et l'intégration des données de flux optique basé sur un système d'inférence floue est proposée pour navigation visuelle des robots [24]. Les variables sont basées sur le traitement des modèles locaux de flux optique. Une étape de segmentation est utilisée pour la détection de l'environnement topologique et topographique du robot.

Les auteurs dans [02] ont considéré l'évitement des obstacles basé sur la vision en utilisant une approche logique floue. Le suivi visuel est également un sujet important pour la navigation mobile [01, 06, 10, 15, 26].

L'objectif de ce travail est d'étudier la navigation visuelle d'un robot mobile dans un environnement intérieur inconnu en utilisant l'association de la technique du flux optique utilisant l'algorithme (Horn & Schunck) avec les méthodes floues de type-1 et du type2, afin de permettre au robot de se mouvoir de façon autonome d'une position initiale à une autre finale en évitant les obstacles. La simulation est réalisée dans un environnement 3D en utilisant le toolbox VRML de Matlab (Virtual Reality Modeling Language).

Cette thèse est organisée de la manière suivante:

Le **premier chapitre** est consacré à l'état de l'art sur la navigation visuelle des robots mobiles. Un aperçu général sur le domaine de la robotique mobile sera abordé. Une étude détaillée sur la navigation visuelle autonome des robots mobiles à roues, et la méthode du flux optique (algorithme de Horn & Schunck) sont présentées.

Dans le **deuxième chapitre**, nous présentons les systèmes d'inférence flous (SIF) avec ses deux versions type-1 et type-2. Nous commençons par énoncer les fondements théoriques des sous-ensembles flous et de la logique floue, suivi par la description de la structure générale des contrôleurs flous de type-1 et de type-2.

Les deux derniers chapitres présentent nos applications:

Le **troisième chapitre** aborde l'application de l'hybridation de la méthode de flux optique (Horn & Schunck) et des systèmes à logique floue de type-1 pour contrôler et commander le mouvement du robot mobile dans un environnement intérieur, en faisant les tâches suivantes : l'évitement des obstacles, convergence vers un but, convergence vers un but avec évitement d'obstacles, et la poursuite d'une cible en mouvement avec évitement d'obstacles.

Dans le **dernier chapitre**, on applique des contrôleurs flous Takagi-Sugeno de type-2 avec l'algorithme Horn-Schunck de flux optique pour la commande d'un robot mobile dans un environnement inconnu en évitant les obstacles.

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

Nous terminerons ce manuscrit par une conclusion générale en présentant ce qui a été fait et exposant les perspectives de ce travail.

Chapitre I

Etat de l'art sur la Navigation Visuelle des Robots Mobiles

I.1 Introduction

Le domaine de la robotique autonome est aujourd'hui très populaire en raison de la grande diversité des applications dans plusieurs domaines. Au début, la principale considération des chercheurs dans ce domaine était de rendre possibles les manipulations dans des domaines où la présence humaine est impossible, très dangereuse ou autrement irréalisable.

Plusieurs tâches faciles qui prennent aujourd'hui un temps précieux de l'homme peuvent être réalisées par un robot, par exemple certaines tâches de maintenance, la collecte des déchets, la livraison au bureau, etc. [27].

Dans le domaine de la robotique mobile, il y a plusieurs axes de recherche tels que la perception de l'environnement, la modélisation de l'espace de travail, la planification de trajectoires sans collision, la conception mécanique et la synthèse des lois de contrôle non linéaires [04, 92]. L'évitement des obstacles est une tâche importante pour l'utilisation correcte des robots mobiles.

La tâche de navigation est définie comme le processus de détermination des séquences de mouvements permettant au robot contrôlé de se déplacer vers la destination finale sans intervention humaine [04]. Dans le domaine de la robotique, différents capteurs ont été utilisés à cet effet, la caméra étant le capteur le plus utilisé et le plus attractif. La vision par ordinateur est un outil important, qui se trouve dans plusieurs d'applications, comme l'appariement pour la vision stéréoscopique, la reconnaissance de formes, l'estimation et l'analyse du mouvement [07].

Dans ce chapitre, nous allons donner un bref exposé sur le domaine de la robotique mobile, en particulier la navigation visuelle autonome des robots mobiles à roues. Nous présentons les différents types des capteurs utilisés. On s'intéresse sur la commande de ces machines autonomes en utilisant la méthode de (Horn-Schunck) du flux optique.

I.2 Navigation autonome des robots mobiles

La navigation d'un robot mobile est la tâche de déterminer une trajectoire optimale permettant au robot de se déplacer d'un point de départ (un point initial) p_i à un point d'arrivée (un point final) p_f pour rejoindre un but désiré, et qui consiste également à

rechercher un mouvement libre dans l'espace de configuration sans collision avec les obstacles proches du robot de manière autonome [35-37].

Il y a deux approches pour les techniques de navigation [38] :

1- La planification du mouvement dans l'environnement (espace de travail) et l'exécution par asservissement du mouvement du robot pour suivre les consignes désirées (schéma planification – exécution) ;

2- La navigation par la décomposition en un ensemble de primitives plus réactives. Elle correspond alors à l'exécution d'une série de sous tâches. (suivre un mur, éviter un obstacle) dont le rôle est la division de la tâche globale en une séquence de primitives.

Afin de réaliser une navigation autonome, un robot mobile doit mettre en œuvre un certain nombre de missions [04, 37, 38], les principales fonctionnalités sont :

I.2.1 Planification de mouvement

La planification de mouvement dans sa formulation classique est le problème de construire un chemin (une trajectoire) définie comme une séquence de déplacement sans collision avec des obstacles pour un système mécanique, entre une configuration de départ (position initiale) et une configuration d'arrivée donnée (position finale), dans un modèle géométrique de l'environnement [37].

I.2.2 Localisation :

La localisation est l'estimation de la position du robot mobile dans l'environnement de navigation. Plusieurs approches de localisation sont utilisées en robotique mobile. Elles peuvent être classées en trois catégories : localisation basée sur une carte, localisation par rapport à des balises et localisation par rapport à d'autres robots [39].

I.2.3 Suivi de trajectoire

Le suivi de trajectoire est une tâche primordiale pour un robot mobile. Il consiste à contrôler le robot pour qu'il poursuive une trajectoire précise en envoyant aux actionneurs les commandes permettant de réaliser le mouvement planifié. Dans ce système, on calcule la distance du robot à une trajectoire de référence, la vitesse de déplacement longitudinale étant donnée. Un robot est un système dynamique commandé par bouclage pour la poursuite de sa trajectoire de référence [29, 40].

I.2.4 Évitement d'obstacles

La tâche de suivi de la trajectoire planifiée ne garantit pas au robot mobile l'absence de collision avec des obstacles. L'évitement des obstacles est indispensable pour permettre au robot de fonctionner dans son environnement (statique ou dynamique). Le robot mobile autonome doit avoir une capacité d'évitement d'obstacle efficace [35, 41, 42, 92].

I.2.5 Parking

Le parking est la tâche finale d'un robot mobile dans la navigation autonome, car l'objectif d'une mission de navigation est souvent d'atteindre une configuration finale (position finale) spécifiée avec une grande précision. Le succès de la navigation dépend de la réalisation de cet objectif de stationnement [92].

I.3. Les Capteurs en Robotique

Il y a deux grandes familles de capteurs qui peuvent être distinguées : les capteurs proprioceptifs, qui donnent des informations sur l'état interne du système, et les capteurs extéroceptifs qui donnent des informations sur l'environnement externe du robot mobile [29].

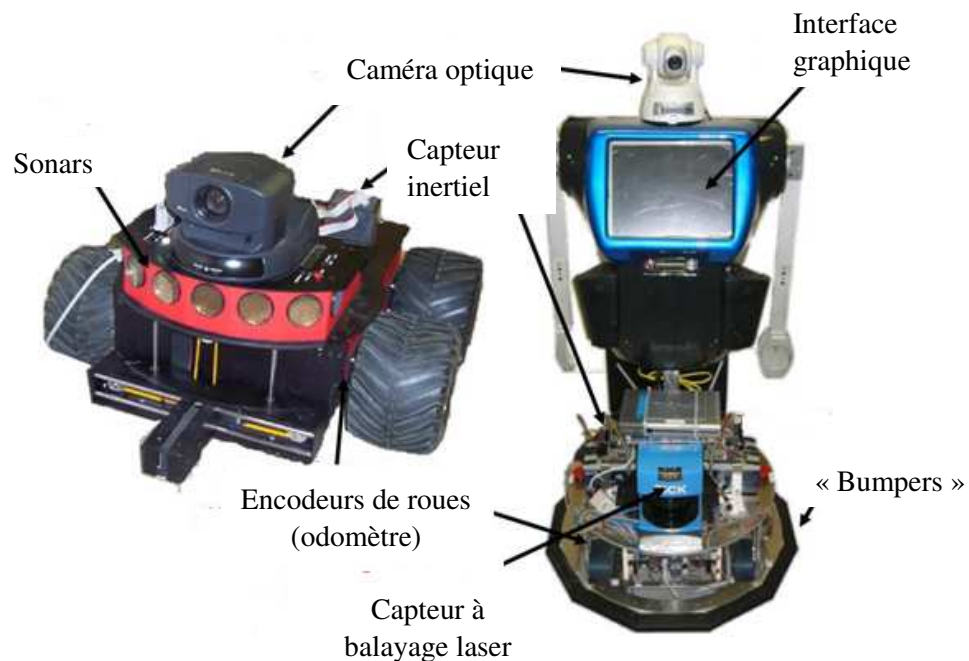


Figure I.1 Les capteurs en robotique

I.3.1 Les capteurs proprioceptifs :

Permettent une mesure de certains paramètres internes au système lui-même. Dans le cas de la robotique mobile, ces capteurs sont souvent utilisés pour estimer l'état interne du robot mobile au cours du mouvement et pour la localisation. Les capteurs proprioceptifs les plus utilisés sont [29]:

- **L'odométrie :** Elle permet d'estimer le déplacement à partir de la mesure de rotation des roues (ou du déplacement des pattes).
- **Les systèmes radar doppler :** ils permettent de mesurer la vitesse du véhicule par effet doppler en utilisant un petit radar pointé vers le sol au lieu de mesurer le déplacement par des mesures des vitesses des roues.
- **Les systèmes inertiels :** ils permettent de calculer la variation de la position en utilisant des accéléromètres pour calculer l'accélération subie par la base mobile et des gyroscopes pour calculer la variation de sa rotation.

I.3.2 Les capteurs extéroceptifs :

Les capteurs extéroceptifs ont pour objectif d'acquérir des informations sur l'environnement proche du robot mobile en mesurant la situation absolue due par rapport à des points de repère naturels (amers visuels) ou artificiels (balises, satellites...) dont la situation est connue dans un référentiel attaché à l'environnement. Les capteurs extéroceptifs peuvent être classés en deux familles [29] :

A) Les capteurs télémétriques

a. Les capteurs à ultrasons :

Les télémètres à ultrasons sont les premiers à avoir été utilisés. Ils utilisent l'air comme milieu de propagation et sont basés sur la mesure temps de retour d'une onde sonore réfléchi par les obstacles pour estimer la distance (Figure I.2). Ils sont utilisés en particulier, sur les robots mobiles pour assurer les suivis de parois ou constituer une ceinture de sécurité [29].

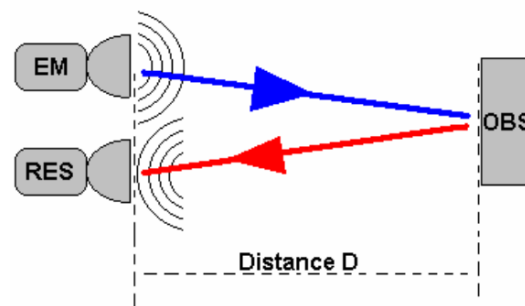


Figure I.2 Schéma synoptique d'un télémètre à ultrasons

b. Télémètres à infrarouge : Au lieu d'utiliser les ondes sonores, les télémètres à infrarouge utilisent une lumière infrarouge pour la détection des obstacles.

c. Télémètres laser : Le capteur laser est basé sur l'émission d'une onde optique. Ce capteur permet donc d'obtenir un faisceau d'ondes monochromatiques de grandes directivités et très concentré. Il est possible de localiser des objets situés à quelques dizaines de mètres avec précision de moins d'un centimètre.

La source laser est généralement associée à une caméra et fournira des informations sur la profondeur.

B) Les systèmes de vision :

La vision en robotique est similaire à la vision humaine, elle fournit des informations précieuses que le robot peut utiliser pour interagir avec le monde qui l'entoure. Les robots équipés de la vision peuvent identifier des couleurs, trouver des pièces, détecter des personnes, traiter des informations sur son environnement.

Les systèmes de vision en robotique sont basés sur l'utilisation d'une caméra CMOS (*complementary metal-oxide-semiconductor*) comme des capteurs de vision très puissants utilisés en système de navigation d'un robot mobile, qu'ils sont très performants en termes de précision, portée, et quantité d'informations à exploiter.



Figure I.3 Caméra panoramique

Les caméras sont des sources d'information très riches sur l'environnement du robot. Elles imitent la vision des êtres naturels pour percevoir l'environnement et pour se localiser et naviguer dans leur environnement [87].

On peut distinguer plusieurs types des caméras parmi lesquelles :

a- Caméras simples

Dans la navigation des robots mobiles, la caméra simple est utilisée pour détecter des amers visuels à partir desquels il sera possible de calculer la position du robot.

b- Caméras stéréoscopiques

Dans la robotique mobile, la caméra stéréoscopie est utilisée pour l'évitement d'obstacles ou la cartographie en estimant la distance des objets et pour calculer l'information de profondeur et reproduire une perception de relief qui se rapproche beaucoup de la vision spatiale humaine. Elle utilise deux images représentant la même scène à partir de deux points de vue légèrement distants.

c- Caméras panoramiques

Les caméras panoramiques nous offrent une large couverture avec une seule caméra. L'image recueillie permet d'avoir une vision de l'environnement sur 360 degrés autour de la camera. Et, parce qu'une seule caméra joue le rôle de plusieurs caméras, les coûts d'installation et d'exploitation du système sont faibles.

Dans ce qui suit, on va donner un aperçu général sur la navigation visuelle. Nous présenterons les approches utilisées pour cette tâche autonome en environnement intérieur et extérieur. La méthode la plus utilisée et adoptée dans ce travail, basée sur le flux optique, sera discutée et présentée, et son algorithme sera exposé en détails.

I.4 Navigation Visuelle d'un robot mobile

La navigation visuelle consiste à utiliser les informations fournies par une ou plusieurs caméras afin de contrôler les mouvements d'un système robotique autonome dans un environnement quelconque, inconnu a priori.

L'utilisation des informations visuelles va permettre d'optimiser la commande des robots, élaborer une loi de commande afin que ces mesures atteignent une valeur désirée ou suivent une trajectoire spécifiée et ainsi de leur garantir plus de flexibilité.

Dans la navigation visuelle, le robot mobile doit exécuter quelques fonctions pour planifier son parcours dans son environnement, parmi lesquelles selon [30, 61] :

- 1) Acquisition d'une image et extraction d'une description 2D de la scène courante ; il s'agit d'une fonction d'interprétation, qui identifie dans l'image, les entités de l'environnement utiles pour la navigation d'un robot terrestre : zones de terrain navigable, obstacles, chemins, amers...
- 2) Construction d'un modèle de l'environnement par fusion des descriptions acquises depuis les positions successives du robot mobile.
- 3) Planification d'une trajectoire requise par une tâche que doit exécuter le robot mobile, typiquement dans le contexte «aller vers la cible tout en évitant les obstacles»... cette trajectoire est générée à partir du modèle de l'environnement, acquis au préalable par le robot mobile, ou introduit par un opérateur dans la base de connaissance du système.
- 4) Exécution de trajectoires par séquençage de mouvements élémentaires contrôlés à partir des images.

Dans ce chapitre, pour choisir l'approche à adopter dans notre travail, nous présentons une classification des systèmes de vision utilisés pour la navigation des robots mobiles.

I.4.1 Classification des approches de navigation visuelle :

La classification des systèmes visuelle pour la navigation autonome d'un robot mobile est basée généralement sur trois critères : le type d'environnement de travail, les hypothèses à proposer, les objectifs de navigation, et les techniques utilisées pour la conception des approches de navigation.

La figure I.4 présente un schéma simple de classification des approches visuelles pour la navigation autonome, la première classification selon le type d'environnement. L'auteur dans son travail [07] a divisé la navigation visuelle du robot en deux catégories principales : Navigation en milieu intérieur et en milieux extérieur [30, 61].

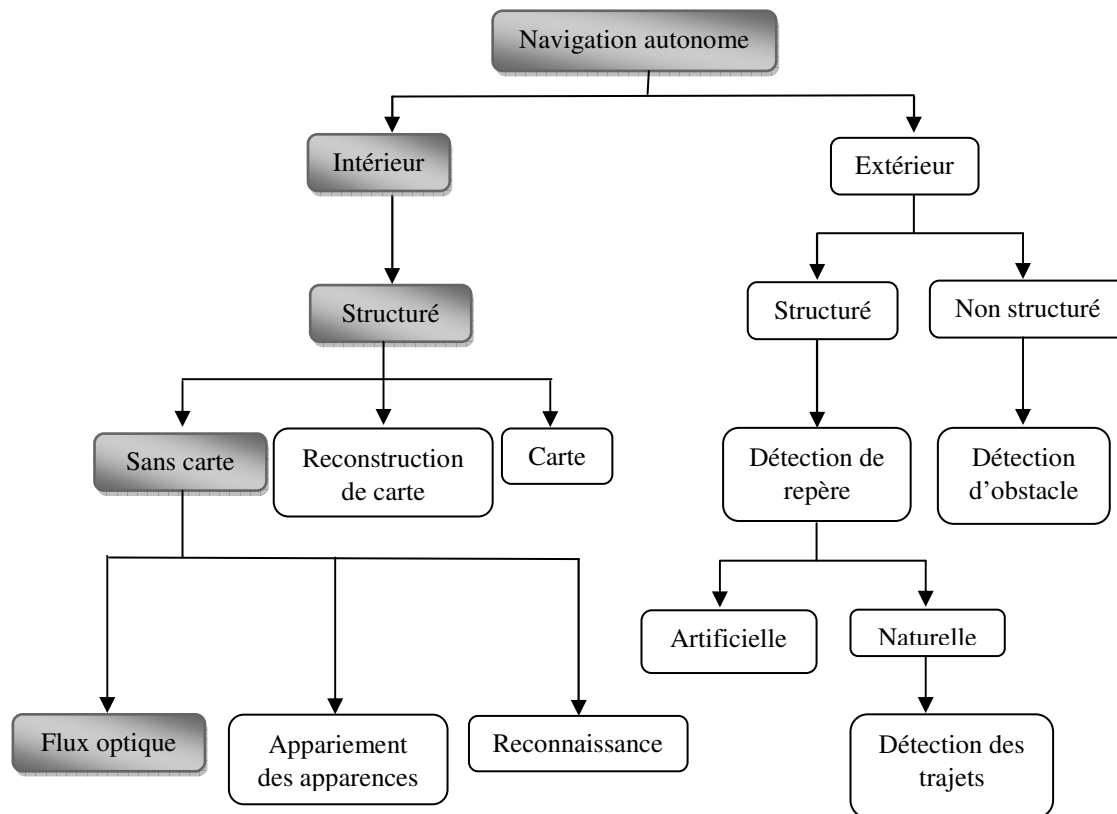


Figure I.4 Approches visuelles pour la navigation autonome

I.4.1.1 Navigation visuelle en milieu intérieur

À partir des développements du premier robot réalisés en 1979 par G. Giralt et al. [31], de nombreux systèmes de contrôle ont incorporé des informations sur l'environnement dans lequel le robot devait naviguer. Les systèmes de navigation et de localisation appartiennent principalement à l'un des trois groupes suivants [30]:

- **Systèmes de navigation basés sur des cartes connues de l'environnement** : le robot mobile utilise des cartes (modèles) géométriques ou topologiques de son environnement, fournies par l'utilisateur pour se localiser et se déplacer en détectant certains amers de l'environnement [43]. La figure I.5 représente un exemple d'une carte topologique, où chaque nœud contient la description d'un lieu distinctif et les arêtes décrivent la trajectoire reliant deux nœuds.

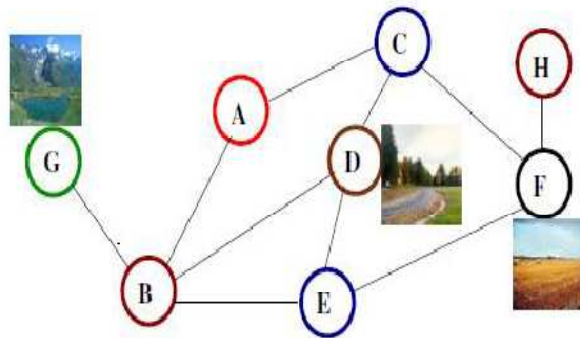


Figure I.5 Carte topologique

- **Systèmes de navigation basés sur la construction de cartes :** A partir des informations provenant des différents capteurs (ultrasons, caméras, lasers, etc.), pendant une phase d'exploration, le robot peut construire son propre modèle de l'environnement [32].

Un exemple de ce type de systèmes est représenté dans la figure I.6, exemple de carte créée par corrélation de scans. Chacun des cercles reliés par une ligne indique le centre d'un scan ajouté à la carte.



Figure I.6 Exemple de carte créée par corrélation de scans

- **Systèmes de navigation sans cartes :** Dans cette catégorie, on inclut la navigation basée sur le flux optique [44, 45, 63], la navigation basée sur l'appariement des apparences [46, 47], et la navigation basée sur la reconnaissance d'objet [48, 62].

I.4.1.2 Navigation visuelle en milieu extérieur

La navigation extérieure est subdivisée en environnements *structurés* qui peuvent être représentés par des primitives géométriques simples (i.e., détection de lignes droites, surfaces planes, couloirs, portes...) et *non structurés*, qui sont en essence des environnements naturels

qui ont subi une modification partielle de l'homme, typiquement les sentiers ou chemins laissés par des passages fréquents de l'homme ou des animaux, par exemple dans le cadre d'activités agricoles [30, 61]. La figure I.7 représente des exemples des environnements non structurés, surface d'une forêt et d'une planète.



Figure I.7 Exemples des environnements non structurés

Tous ces modèles contiennent, implicitement, des actions pour l'évitement d'obstacles, la détection d'amers, la construction ou l'actualisation de cartes et l'estimation de la position du robot.

L'approche proposée dans ce travail est basée sur l'utilisation la technique du flux optique avec un système de navigation sans carte, dans un milieu intérieur structuré, pour éviter les obstacles et obtenir un guidage efficace d'un robot mobile par rapport à son univers.

I.5 Flux Optique (FO) :

La méthode de navigation visuelle est une approche intéressante utilisée dans les applications robotiques [8, 12, 19]. Le système de vision peut améliorer l'efficacité des robots mobiles dans différents domaines [7, 8, 14].

Il existe principalement deux approches utilisées pour la navigation visuelle :

L'approche naturelle : cette approche est inspirée du système de vision de l'être humain ou des insectes pour produire un système de vision.

L'approche artificielle : cette approche est basée sur les images et des informations images extrait d'une caméra du robot.

L'inspiration du vivant et le choix du flux optique pour la navigation ont été motivés par des travaux réalisés sur la mouche et l'abeille mettant en évidence le rôle du flux optique dans la navigation de manière générale [34, 63].

Le flux optique est la méthode la plus utilisée pour la navigation des robots mobiles basée sur la vision [7, 14, 17-20]. Cette approche est une stratégie de calcul de vitesse utilisée principalement dans :

- l'estimation de mouvement ;
- la compression-reconstruction vidéo ;
- la segmentation d'image [18, 20] ;
- la détection et l'évitement d'obstacles [2, 17, 18].

I.5.1 Définition du Flux Optique (FO) :

On peut définir le flux optique comme un mouvement apparent des intensités de l'image causé par la projection 2D sur une rétine (humaine ou artificielle), du mouvement relatif 3D des points de la scène (un concept géométrique). C'est la vitesse de défilement des objets sur le plan-image d'une caméra embarquée. Il dépend de plusieurs facteurs, comme par exemple la forme des objets, leur vitesse propre, la distance ou la vitesse relative entre l'objet et la caméra (ou l'œil) [33]. Le calcul du flux optique est lié au capteur optique utilisé.

Il existe un lien entre le déplacement de la caméra et la perception de l'environnement qu'il engendre. Par exemple, si nous mettons un robot, doté d'une camera, disposé face à un mur texturé, si le robot se déplace vers la gauche parallèlement au mur, les mouvements des pixels s'effectueront vers la droite avec une vitesse égale à la vitesse du robot.

À partir de l'estimation du flux optique, il est possible de déduire la vitesse de la caméra et la direction de son mouvement [34].

La Figure suivante I.8 donne un exemple de flux optique. Il s'agit du flux optique obtenu dans le cas de translations selon les axes (X , Y et Z).

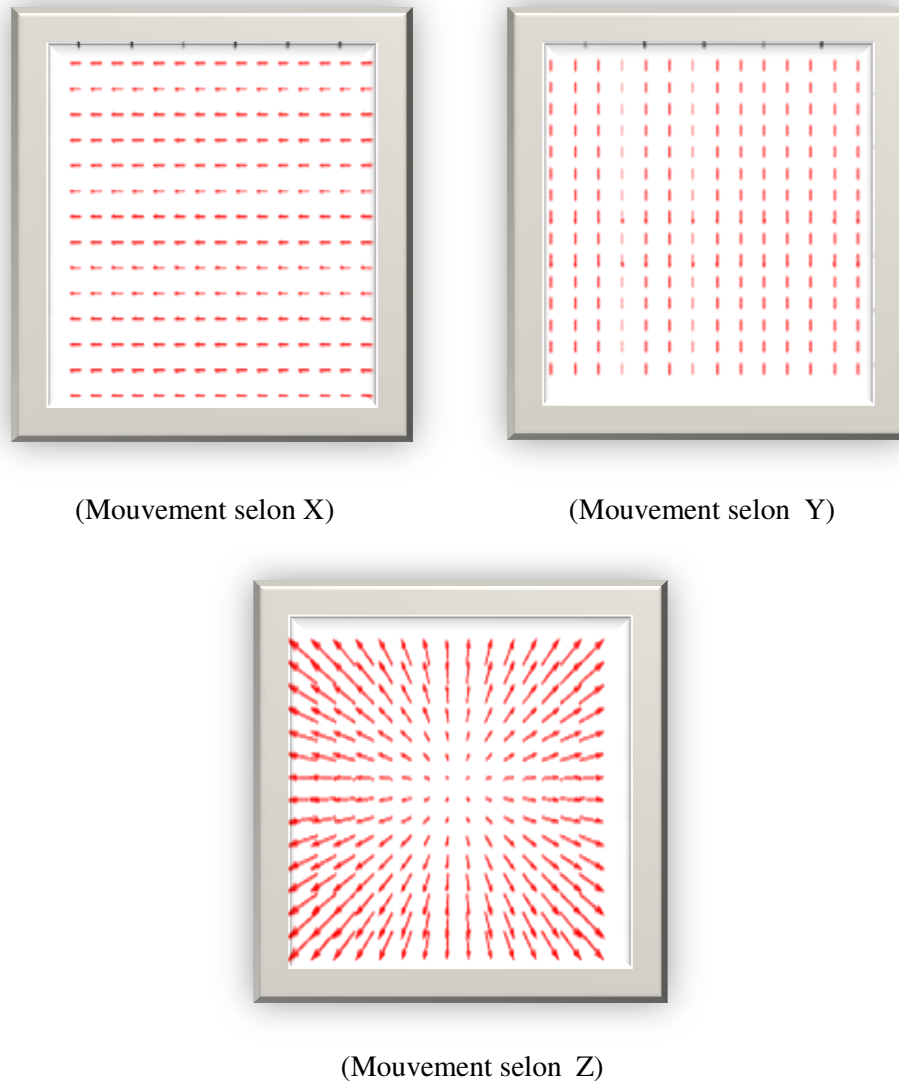


Figure I.8 Flux optique pour un mouvement en translation (X, Y et Z)

I.5.2 Exemples d'applications

Le flux optique est utilisé dans plusieurs applications [99]:

- Poursuite d'objets (militaire, vidéo surveillance, robotique);
- Analyse de comportement complexe par le mouvement : modélisation (biologie, météorologie, ...);
- Codage vidéo pour la télévision (ou Internet) : compression des données;
- En médecine: analyse des contractions cardiaques (détection des infarctus), modélisation des mouvements du corps humain ;
- Infographie : donner du réalisme au mouvement et déformation d'objets complexes.

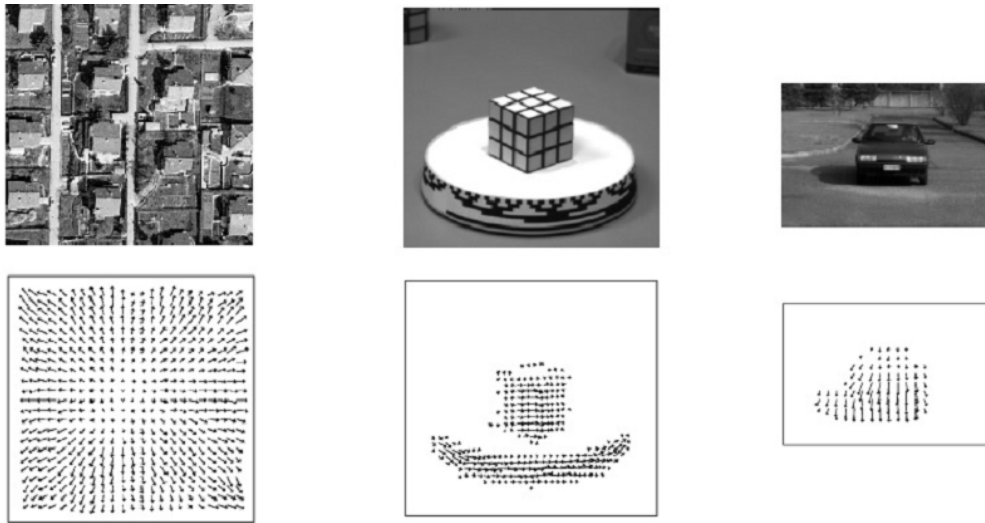


Figure I.9 Exemples du flux optique.

I.5.3 Propriétés du flux optique

Dans notre travail, nous avons utilisé l'approche du flux optique pour détecter les objets et les obstacles dans l'environnement de navigation du robot mobile. Cette méthode est utilisée pour calculer le mouvement des pixels [49, 89]. Il est basé sur les vecteurs de vitesse dans l'image.

I.5.3.1 Composition du flux optique

Nous utilisons le modèle 'trou d'épingle' (illustré dans la figure I.10) pour déterminer et calculer les coordonnées d'un point dans l'image $(x, y)^T$, à partir d'un point de l'espace réel exprimé par sa représentation cartésienne $(X, Y, Z)^T$, (les propriétés géométriques liées à la projection de l'environnement en 3D sur le plan d'image à 2D) [33].

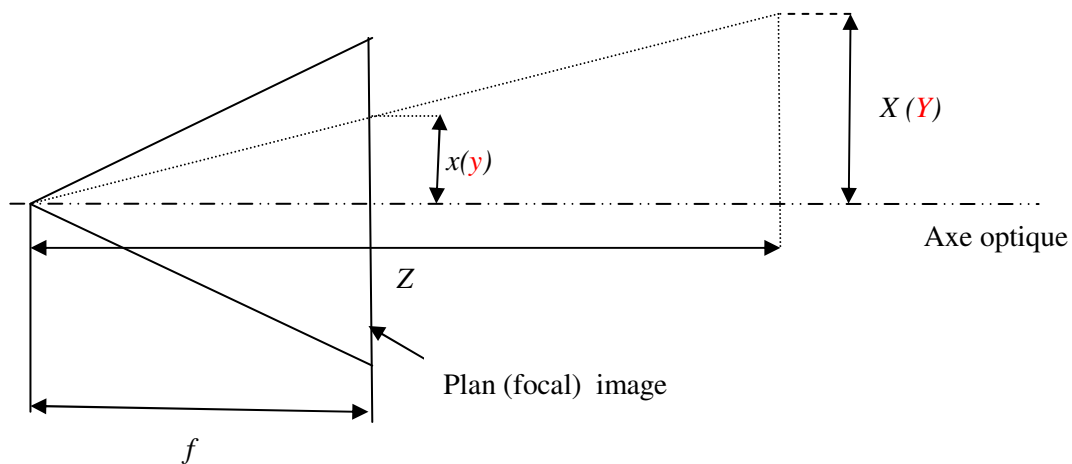


Figure I.10 Illustration du plan image de caméra.

De la figure I.10, on peut extraire la relation entre $(x,y)^T$ et $(X,Y,Z)^T$ sous la forme :

$$x = f \frac{X}{Z} \quad (\text{I.1})$$

$$y = f \frac{Y}{Z} \quad (\text{I.2})$$

On note :

- (X, Y, Z) , les coordonnées réelles 3D d'un point de la scène ;
- (x, y) , les coordonnées rétine 2D du même point projeté dans l'image ;
- f , la distance focale de la caméra.

Si on dérive (x, y) par rapport au temps (dérivée temporel du flux optique), nous obtenons :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \frac{f}{Z^2} \begin{pmatrix} \dot{X}Z - X\dot{Z} \\ \dot{Y}Z - Y\dot{Z} \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} - \frac{f}{Z} \frac{\dot{Z}}{Z} \begin{pmatrix} X \\ Y \end{pmatrix} \quad (\text{I.3})$$

I.5.3.2 Foyer d'expansion (FOE) :

La projection du vecteur de translation dans le plan de l'image définit le point appelé foyer d'expansion (FOE). Les coordonnées de ce point donnent des informations sur la direction de la translation 3D de la caméra [03].

De plus, le FOE représente le point de projection dans l'image permettant d'obtenir des informations sur la profondeur de certains pixels. Dans une situation où la caméra se déplace vers l'avant, le point de mise au point du point d'expansion est indiqué sur la Figure I.11 (a) (cercle rouge).

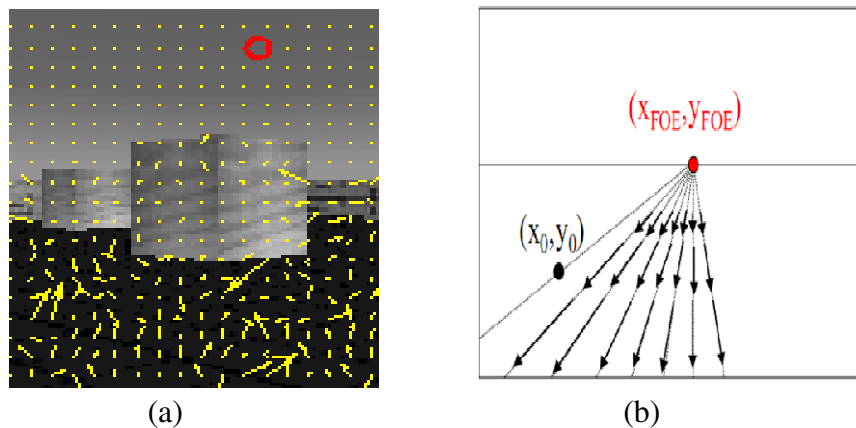


Figure I.11 Représentation du foyer d'expansion

La figure I.11 (b), est une représentation d'un flux optique simulant un déplacement en translation uniforme vers l'avant dans une scène statique. Les directions de vitesse des points projetés sur le plan d'image convergentes vers le point qui se trouve au centre (en rouge) le foyer d'expansion (FOE). Chaque vecteur de mouvement (flèches noirs), représente le mouvement visuel provoqué par le déplacement de l'observateur [33,85].

Si le mouvement de translation sur l'axe optique OZ ($\dot{X} = 0, \dot{Y} = 0, \dot{Z} = c^{te}$), l'équation (I.3) de vecteur de déplacement $(\dot{x}, \dot{y})^T$ est comme suit:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = -\frac{f}{Z} \frac{\dot{Z}}{Z} \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} -x \frac{\dot{Z}}{Z} \\ -y \frac{\dot{Z}}{Z} \end{pmatrix} \quad (\text{I.4})$$

Le champ de déplacement forme un zoom sur l'image avec foyer d'expansion (FOE). Soit T le vecteur de translation :

$$T = (-\dot{X}, -\dot{Y}, -\dot{Z}) \quad (\text{I.5})$$

Soit (X_0, Y_0, Z_0) un point de la scène, le point projeté correspondant dans l'image est (x_t, y_t) , avec :

$$(x_t, y_t) = \left(f \frac{X_0 + t\dot{X}}{Z_0 + t\dot{Z}}, f \frac{Y_0 + t\dot{Y}}{Z_0 + t\dot{Z}} \right) \quad (\text{I.6})$$

$$(x_{FOE}, y_{FOE}) = \lim_{t \rightarrow -\infty} (x_t, y_t) = \left(f \frac{\dot{X}}{\dot{Z}}, f \frac{\dot{Y}}{\dot{Z}} \right) \quad (\text{I.7})$$

I.5.3.3 Estimation du temps avant collision

Le concept du temps avant collision, souvent appelé Time-To-Contact en anglais et noté TTC a été introduit pour la première fois par Lee [50]. Le but du TTC, dans la navigation du robot, est d'estimer l'espace libre devant le robot et donc de décider, si le robot doit tourner (droite-gauche) ou s'arrêter quand un obstacle est détecté. Ce problème a été étudié par de nombreux auteurs tels que [11, 50, 86]. Dans le cas d'un mouvement de caméra le long de l'axe optique Z , le temps de contact est exprimé par le rapport entre la vitesse de rapprochement et la distance de l'obstacle détecté (la distance objet-observateur) [03, 13, 85].

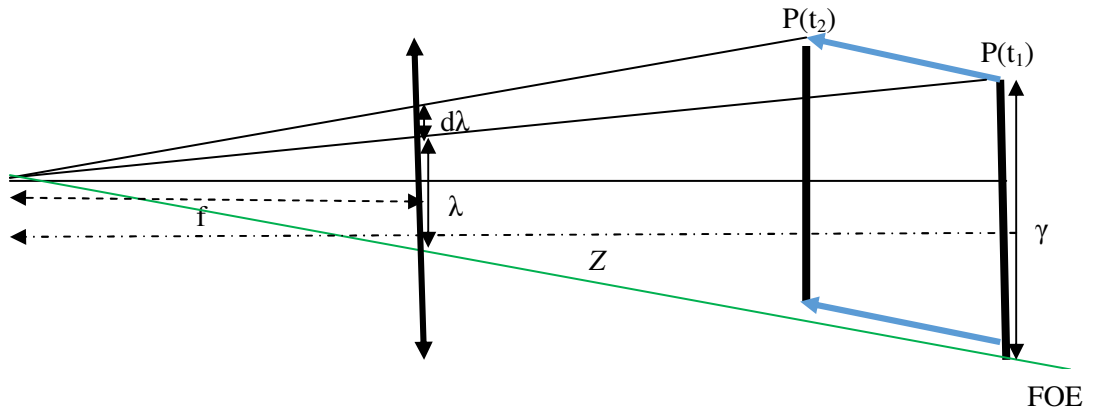


Figure I.12 Temps Avant Collision (TTC)

Mouvement d'une scène statique :

$$\lambda = f \frac{\gamma}{Z} \quad (\text{I.8})$$

$$\dot{\lambda} = -\frac{f\gamma}{Z^2} \dot{Z} \quad (\text{I.9})$$

Et donc, le TTC est calculé comme suit :

$$TTC = \frac{\lambda}{\dot{\lambda}} = -\frac{Z}{\dot{Z}} \quad (\text{I.10})$$

où Z : est la distance caméra-obstacle ;

et $\dot{Z} = dZ / dt$: est la vitesse de la caméra du robot.

I.5.4 Calcul du flux optique :

Barron, et al [54] ont fait une étude détaillée sur les différentes méthodes de calcul du flux optique. Il existe trois catégories de méthodes pour le calcul du flux optique :

- Les méthodes différentielles.
- Les méthodes de corrélation.
- Les méthodes fréquentielles.

Les deux premières méthodes correspondent à la même forme d'estimation [55] mais elles diffèrent par leur mise en œuvre algorithmique. La dernière famille de méthode est plus moins utilisée en raison de la complexité des traitements nécessaires [34, 54, 55].

I.5.4.1 Les méthodes différentielles:

Les méthodes différentielles sont les premières méthodes basées sur le calcul du flux optique à partir des dérivées spatiotemporelles d'intensité lumineuse de l'image (spatiale en x , y et temporel en t). Certaines de ces méthodes utilisent les dérivées premières de l'intensité lumineuse, cependant d'autres utilisent des dérivées secondes. Ce sont des méthodes qui sont très sensibles au bruit.

Deux types de méthodes différentielles sont utilisées pour la résolution du problème de flux optique : locale et globale [56].

a. Méthodes différentielles locales

La méthode différentielle locale consiste à régulariser le calcul du mouvement apparent sur un voisinage local. Cette méthode, présentée pour la première fois par Lucas & Kanade [89, 90] est fondée sur une optimisation locale par la méthode des moindres carrés (qui utilise une méthode des moindres carrés sur la contrainte d'intensité). La méthode consiste à minimiser la contrainte du gradient localement.

b. Méthodes différentielles globales

La méthode différentielle globale consiste à minimiser l'énergie globale dans l'image, une contrainte dite d'attache aux données basée sur l'équation du flux optique ainsi qu'une contrainte supplémentaire de lissage [57, 58]. Elle est plus sensible au bruit que la stratégie locale.

La première méthode globale proposée dans la littérature est celle de Horn et Schunck qui impose une contrainte globale de lissage sur le champ des vitesses.

Dans la suite de cette partie, nous allons expliquer la méthode d'Horn & Schunck que nous avons choisi dans notre travail pour détecter les obstacles. Nous travaillerons sur des images en niveaux de gris. La seule information alors disponible sera l'intensité lumineuse de chaque pixel.

I.5.4.2 Équations du flux optique

Une séquence d'images est une fonction réelle $I(x, y, t)$ (qui représente l'intensité lumineuse). La plupart des méthodes d'estimation du flux optique reposent sur une hypothèse fondamentale : l'intensité lumineuse se conserve entre deux images successives.

Cela s'écrit sous la forme générale :

$$I(p, t) = I(p + V(p)dt, t + dt) \quad (\text{I.11})$$

Avec :

$P(x, y)^T$: C'est le point, qui est défini par ses coordonnées x et y .

$V(p) = (u, v)^T$: Le vecteur de vitesse associé au point p à l'instant t .

Où (u, v) : Représentent les vitesses selon les directions x et y , ce que l'on cherche.

Comme cette équation n'a pas une solution unique, nous devons faire l'hypothèse que l'intensité varie lentement avec x , y et t . Si cette hypothèse est respectée, nous pouvons développer le premier membre de l'équation (I.11) en séries de Taylor à l'ordre 1 autour de (x, y, t) .

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + e \quad (\text{I.12})$$

La variable e contient les termes d'ordres plus élevés. Puisque $I(x + dx, y + dy, t + dt) = I + dI$, en divisant par dt , et en négligeant les termes d'ordre supérieurs, l'équation (I.12) peut-être réduite à :

$$\frac{dI}{dt} = \frac{dx}{dt} \frac{\partial I}{\partial x} + \frac{dy}{dt} \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \quad (\text{I.13})$$

Par l'hypothèse que l'intensité lumineuse ne varie pas avec le temps, qui conduit à $\frac{dI}{dt} = 0$. L'équation du flux optique se résume alors à la formulation suivante :

$$\frac{dx}{dt} \frac{\partial I}{\partial x} + \frac{dy}{dt} \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \quad (\text{I.14})$$

Si on note :

- $u = \frac{dx}{dt}$: La vitesse apparente suivant la direction x ;
- $v = \frac{dy}{dt}$: La vitesse apparente suivant la direction y ;
- $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = (I_x, I_y)$: gradient de l'image au pixel (x, y) à l'instant t ;
- $I_t = \frac{\partial I}{\partial t}$: est la dérivée de l'intensité d'un pixel par rapport au du temps.

L'équation (I.14) devient :

$$I_x u + I_y v + I_t = 0 \quad (\text{I.15})$$

Cette équation est appelée « équation de contrainte du flux optique ».

A. Méthode de Horn et Schunck

L'algorithme de **Horn et Schunck (HS) 1981** est l'un des algorithmes classiques du flux optique en raison de ses performances acceptables et de sa simplicité [49]. C'est une méthode différentielle utilisée dans de nombreuses applications. Cet algorithme est utilisé pour ajuster le modèle du noyau en utilisant la densité de la vitesse [20].

Pour la technique de Horn-Schunck [49], pour estimer le champ de vecteurs de vitesses, on doit ajouter un terme global de régularisation à l'équation du flux optique (I.15). Puis, on minimise le critère selon la contrainte du flux et du terme de régularisation.

La méthode différentielle de Horn-Schunck calcule l'estimation du champ de vitesse, c'est une méthode basée sur la minimisation d'une fonction d'énergie sur le domaine Ω [60] :

$$E = \iint (I_x u + I_y v + I_t)^2 dx dy + \beta \iint \left\{ \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right\} dx dy \quad (\text{I.19})$$

$$E = \int_{\Omega} (I_x u + I_y v + I_t)^2 + \beta^2 (\|\nabla u\|^2 + \|\nabla v\|^2) dx dy \quad (\text{I.20})$$

$$J_{HS} = \iint \left[((\nabla I)^T w + I_t)^2 + \beta ((\nabla v_x)^2 + (\nabla v_y)^2) \right] dx dy \quad (\text{I.21})$$

Le terme de régularisation est défini comme une contrainte de lissage de l'information, qui est donné par un carré des dérivées partielles de la vitesse $\|\nabla u\|^2$ et $\|\nabla v\|^2$. où β est l'influence donnée au terme de régularisation. Un lissage des vitesses est effectué dans un voisinage donné. Afin d'estimer la vitesse, on peut utiliser une version itérative de la minimisation de l'équation (I.21) :

$$u_{x,y}^{k+1} = \bar{u}_{x,y}^k - \frac{I_x [I_x \bar{u}_{x,y}^k + I_y \bar{v}_{x,y}^k + I_t]}{\beta^2 + I_x^2 + I_y^2} \quad (\text{I.22})$$

$$v_{x,y}^{k+1} = \bar{v}_{x,y}^k - \frac{I_y [I_x \bar{u}_{x,y}^k + I_y \bar{v}_{x,y}^k + I_t]}{\beta^2 + I_x^2 + I_y^2} \quad (I.23)$$

Dans ces équations, $[u_{x,y}^k, v_{x,y}^k]$ est l'estimation de la vitesse pour le pixel à (x, y) dans l'image et I_x , I_y et I_t ses dérivées partielles, et $[\bar{u}_{x,y}^k, \bar{v}_{x,y}^k]$ est la moyenne du voisinage de $[u_{x,y}^k, v_{x,y}^k]$. K est le nombre d'itérations et pour $k = 0$, la vitesse initiale est 0 [60].

Pour résoudre u et v en utilisant la méthode de Horn-Schunck, nous pouvons utiliser les étapes suivantes [02] :

1. Calculer I_x et I_y en utilisant le noyau de convolution de Sobel, $[-1 \ -2 \ -1 ; 0 \ 0 \ 0 ; 1 \ 2 \ 1]$, et sa forme transposée pour chaque pixel de la première image.
2. Calculer I_t entre les deux images 1 et 2 en utilisant le noyau $[-1 \ 1]$.
3. Supposons que la vitesse précédente soit 0, et calculons la vitesse moyenne pour chaque pixel en utilisant $[0 \ 1 \ 0 ; 1 \ 0 \ 1 ; 0 \ 1 \ 0]$ comme noyau de convolution.
4. Résoudre itérativement pour u et v .

I.6 Conclusion

Dans ce chapitre nous avons présenté un exposé sur le domaine de la robotique mobile, en définissant la navigation autonome des robots mobiles, les types des capteurs utilisés pour la localisation. Nous avons mentionné en détaille la navigation visuelle d'un robot mobile dans l'environnement intérieur et extérieur avec une explication de la méthode du flux optique en concentrant sur la méthode de Horn-Schunck qui est utilisée dans notre travail.

Après cet aperçu, puisque notre travail est basé sur la théorie de la logique floue pour commander le robot mobile, nous présenterons dans le chapitre suivant une étude générale sur les systèmes d'inférences flous et les fondements théoriques des ensembles flous de deux types, type-1 et type-2.

Chapitre II

Systemes Flous Type-1 et Type-2

II.1 Introduction

La logique floue (LF) a été utilisée dans plusieurs domaines d'application, notamment contrôleur d'un robot mobile. Cette approche a été établie la première fois en 1965 par Zadeh [68], comme une description mathématique basée les ensembles flous. La LF est une méthode de traitement des connaissances imprécises et incertaines. Elle est inespérée de la capacité de l'homme à décider et agir de façon pertinente malgré l'incertitude des connaissances disponibles. En effet, la logique floue a été introduite pour approcher du raisonnement humain à l'aide d'une représentation adéquate des connaissances sous forme des règles floues [67].

Dans certaines applications, la logique floue type-1 sera incapable de prendre en charge les incertitudes dans les différentes parties du système flou, la deuxième version proposée comme une extension des systèmes flous type-1 est le système flou type-2, qui peut résoudre ce problème [72].

Dans ce chapitre, on va présenter en bref une description des systèmes flous type-1 et type-2. Des définitions des ensembles flous type-1 et type-2 avec les structures des systèmes d'inférence flous pour les deux types seront présentés.

II.2 Systèmes Flous type-1 (SF1)

II.2.1 Ensembles flous type-1 (EF1)

La logique floue est une approche considérée comme une extension de la logique booléenne. Elle permet de traiter des variables linguistiques imprécises. La logique floue est basée sur la notion des ensembles flous établis en 1965 par L. Zadeh dans sa publication intitulée (*Fuzzy Sets*). L'objectif principal est la modélisation des notions (connaissances) vagues et incertaines du langage naturel [18, 64- 67, 92].

Soit U une collection d'objets (par exemple $U = R^n$) appelée univers de discours. Un ensemble flou type-1 est une fonction mathématique en 2D défini par sa fonction d'appartenance notée, $\mu_A(x)$, ($\mu_A : U \rightarrow [0,1]$). Il est représenté comme un ensemble de paires ordonnées de x et sa fonction d'appartenance partielle à l'ensemble A sous la forme:

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

Dans le cas discret (U est discret), l'ensemble flou A est représenté par :

$$A = \sum_{x_i} \mu_A(x_i) / x_i \tag{II.2}$$

Sinon (Si U est continu), l'ensemble flou A est donné par :

$$A = \int_x \mu_A(x_i) / x_i \tag{II.3}$$

La Figure II.1 présente les deux types des ensembles : classique et flou.

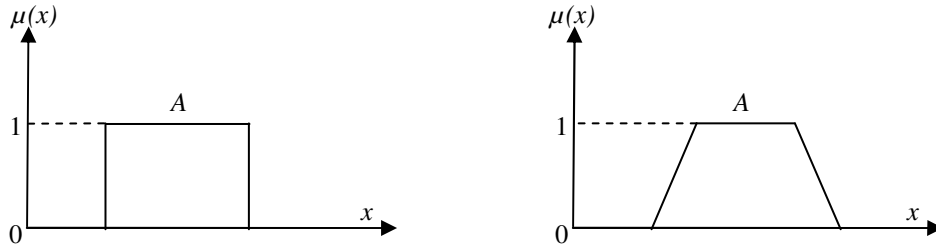
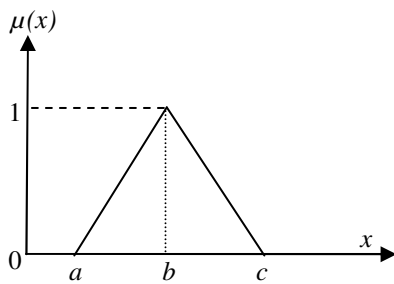


Figure II.1 Ensemble classique et ensemble flou

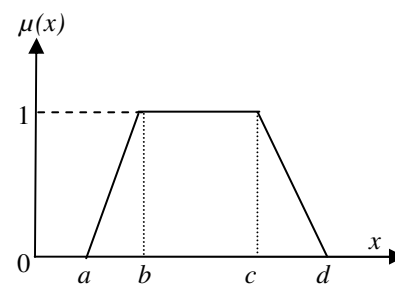
II.2.2 Type des ensembles flous :

Plusieurs formes des ensembles flous peuvent être utilisées, les plus connues sont [92]:

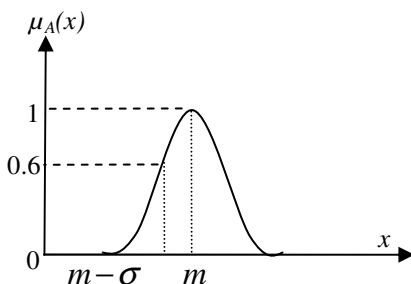
- **EF de type Triangulaire:** il est défini par trois paramètres : a, b et c (figure II.2 (a)).
- **EF de type Trapézoïdale:** il est défini par quatre paramètres : a, b, c et d (figure II.2 (b)):
- **EF de type Gaussien:** il est défini par deux paramètres : $\{\sigma, m\}$ (figure II.2 (c)):
- **EF de type Sigmoide:** il est défini par deux paramètres $\{a, c\}$ (figure II.2 (d)):
- **EF de type Constant:** il est défini par un seul paramètre : a_i (figure II.2 (e)):



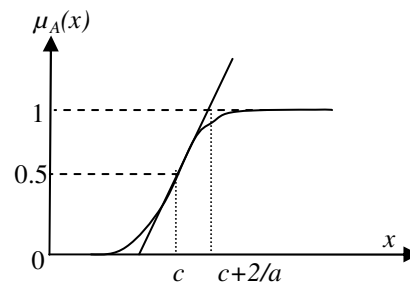
(a)



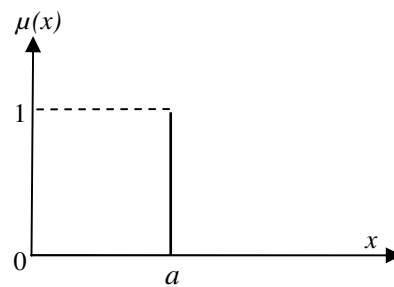
(b)



(c)



(d)



(e)

Figure II.2 Types des ensembles flous

II.2.3 Variable Floue

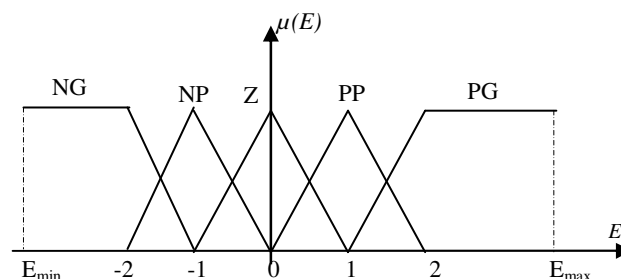
C'est une variable dont les valeurs ne sont pas des nombres, mais des mots ou phrases exprimés en langage naturel. La raison est que le caractère linguistique est moins spécifique que le caractère numérique. Une variable linguistique est généralement représentée par $(x, T(x), U)$ avec:

- x : nom de la variable linguistique (erreur, distance, vitesse, température, angle,...),
- $T(x)$: ensemble des valeurs linguistiques qui sont utilisées pour caractériser x ,
- U : univers de discours de la variable linguistique x .

Si on considère l'erreur comme une variable linguistique définie sur l'univers de discours $U = [E_{\min}, E_{\max}]$, ses valeurs linguistiques peuvent être définies comme suit:

L'ensemble $T(x=Erreur)$ {Négative Grande (NG), Négative Petite (NP), Zéros(Z), Positive Petite (PP), Positive Grande (PG)}.

Ces mots linguistiques peuvent être définis par des ensembles flous de la figure II.3.

**Figure II.3** Représentation de l'erreur par cinq termes linguistiques

II.2.4 Opérations sur les ensembles flous

Afin de manipuler les sous-ensembles flous, il est nécessaire de définir certaines opérations élémentaires. Soit A et B deux ensembles flous dans U caractérisés respectivement par les deux fonctions d'appartenance $\mu_A(x)$ et $\mu_B(x)$ [65, 66, 69, 92]:

Les opérateurs de la logique floue sont définis comme suit :

Egalité floue ($A = B$)	$\forall x \in U, \mu_A(x) = \mu_B(x)$
Sous-ensemble flou ($A \subseteq B$)	$\forall x \in U, \mu_A(x) \leq \mu_B(x)$
Complémentation floue (\bar{A})	$\forall x \in U, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$
Union floue ($A \cup B$)	$\forall x \in U : \mu_{A \cup B} = \mu_A(x) + \mu_B(x)$ Où «+» est la <i>co-norme</i> triangulaire qui est donnée par : $\mu_A(x) + \mu_B(x) = \max(\mu_A(x), \mu_B(x))$
Intersection floue: ($A \cap B$)	$\forall x \in U : \mu_{A \cap B} = \mu_A(x) * \mu_B(x)$ Où «*» est la norme triangulaire [66] avec les deux formes : $\mu_A(x) * \mu_B(x) = \min(\mu_A(x), \mu_B(x))$ $\mu_A(x) * \mu_B(x) = \mu_A(x) \times \mu_B(x)$
Produit cartésien: Si A_1, \dots, A_n sont des ensembles flous respectivement définis sur U_1, \dots, U_n	leur produit cartésien est un ensemble flou, dénoté par $A_1 \times \dots \times A_n$, défini sur le produit $U_1 \times \dots \times U_n$ $\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n) = \mu_{A_1}(x_1) * \dots * \mu_{A_n}(x_n)$
Relation floue: Une relation floue d'ordre n est un ensemble flou défini sur U_1, \dots, U_n	$R_{X_1 \times \dots \times X_n} = \left\{ ((U_1, \dots, U_n), \mu_R(x_1, \dots, x_n)) \mid (x_1, \dots, x_n) \in U_1 \times \dots \times U_n \right\}$
Composition des relations floues La composition de R et S est un ensemble flou, symbolisé par $R \circ S$,	$\mu_{R \circ S}(x, z) = \left\{ (x, z), \sup_{y \in Y} (\mu_R(x, y) * \mu(y, z)) \right\}$
Implication floue : Si x est A Alors y est B	$\mu_R(x, y) = \text{imp}(\mu_A(x), \mu_B(x))$

	Implication de Mamdani: $\mu_R(x, y) = \min(\mu_A(x), \mu_B(x))$ Implication de Larsen: $\mu_R(x, y) = \mu_A(x) \times \mu_B(x)$
--	---

II.2.5 Raisonnement Flou

Il consiste à manipuler et utiliser les propositions et les règles floues pour obtenir une décision. En logique classique, les raisonnements sont de la forme :

$$\begin{cases} \text{Si } p \text{ alors } q \\ p \text{ vrai alors } q \text{ vrai} \end{cases}$$

En logique floue, Zadeh a étendu le mode de raisonnement de la logique classique qui est le modus ponens (MP) à un modus ponens généralisé (GMP) [66, 69, 92, 95] :

$$\begin{array}{llll} \text{R\`egle floue} & : & \text{Si } x \text{ est } A & \text{ Alors } y \text{ est } B & \mu_{A'}(x) & \mu_B(x) \\ \text{Fait} & : & x \text{ est } A' & & & \\ \text{Cons\`equence} & : & y \text{ est } B' & & & \end{array}$$

En utilisant la règle «*Si x est A Alors y est B*», on d\`eduit un nouveau fait «*B'*» caract\`eris\`e par une fonction d'appartenance:

$$\mu_{B'}(y) = \text{Sup}_x(\mu_{A'}(x) * \mu_R(x, y)) \quad (\text{II.4})$$

O\`u $\mu_{A'}(x)$ et $\mu_R(x, y)$ sont les fonctions d'appartenance de A' et de la r\`egle floue.

II.2.6 Structure des Syst\`emes Flous (SF1)

Les syst\`emes flous type-1 sont des syst\`emes de traitement bas\`es sur la th\`eorie de la logique floue et ses concepts flous (ensembles flous type-1, variables floue, r\`egle floue, ...etc.). C'est un syst\`eme \`a base de connaissance comme pr\`esent\`e sur la figure II.4. Chaque SF1 se compose de quatre modules : *fuzzification*, *base de r\`egles*, *moteur d'inf\`erence* et *d\`efuzzification* [67].

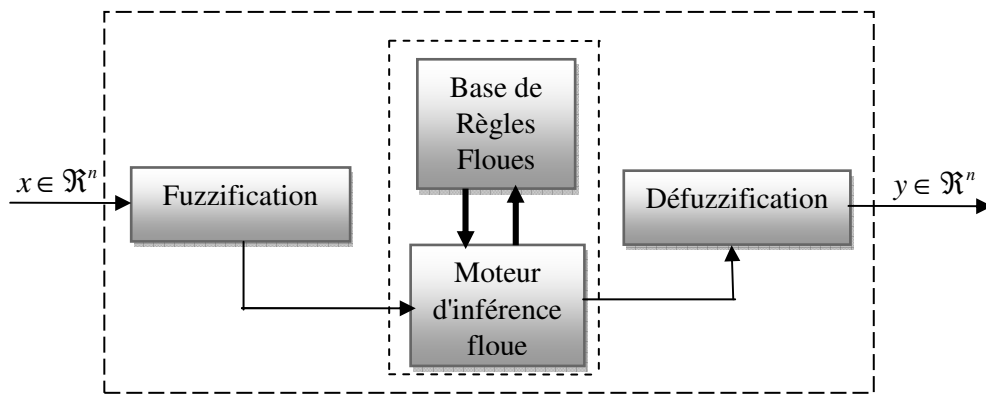


Figure II.4 Structure d'un système flou type-1

- **Fuzzification:** c'est l'étape de transformation d'une grandeur mesurée sur le processus en un ensemble flou (*Calcul des degrés d'appartenances*).
- **Base de règles floues:** Elle permet de définir les relations entre les variables d'entrées et de sorties décrivant le comportement du système.
- **Moteur d'inférence:** c'est le cœur de traitement. Il permet de calculer l'ensemble flou de sortie en utilisant les règles floues (*Calcul de degré de vérité, Implication et Agrégation*).
- **Défuzzification:** Elle permet de transformer l'ensemble flou de sortie en une valeur de commande applicable au processus.

II.2.7 Types des Systèmes Flous type-1

Dans la littérature, plusieurs types du raisonnement flou sont proposés en utilisant les opérateurs flous "ET, OU" et les règles floues. Les deux moteurs d'inférence les plus utilisés sont: de Mamdani et de Takagi-Sugeno.

II.2.7.1 Méthode de Mamdani

Mamdani est le premier à utiliser la logique floue pour la commande des systèmes [70]. Cette inférence utilise le minimum comme opérateur de conjonction et d'implication. Les règles de Mamdani sont de la forme [70]:

$$R_i : \text{Si } x_1 \text{ est } A_{i1} \text{ et } x_2 \text{ est } A_{i2} \text{ et } \dots \text{ et } x_n \text{ est } A_{in} \text{ ALORS } y \text{ est } B_i \quad (\text{II.5})$$

où B^i est un sous ensemble flou.

II.2.7.2 Méthode de Takagi-Sugeno

Le système flou de Takagi-Sugeno est considéré comme un cas particulier important; ou la conclusion n'est pas symbolique mais une fonction des entrées comme suit:

$$B^i = f(x_1, \dots, x_n).$$

Les règles floues de Takagi-Sugeno (TS) sont données par la forme suivante :

$$R_i : \text{Si } x_1 \text{ est } A_{i1} \text{ et } x_2 \text{ est } A_{i2} \text{ et } \dots \text{ et } x_n \text{ est } A_{in} \text{ ALORS } y \text{ est } f_i(x) \quad (\text{II.6})$$

où :

$f_i(x)$: qui peut prendre soit la forme d'une constante. Dans ce cas, on dit que le modèle est de TS d'ordre zéro (TS0), ou de premier ordre, on dit que le modèle est de TS d'ordre un [71, 92].

II.2.8 Propriété d'Approximation Universelle :

Parmi les notions les plus importantes des systèmes flous est la propriété d'approximation universelle [72, 74, 92], où n'importe fonction réelle et continue f définie sur un compact C de R^n , il existe un SIF qui peut approcher son comportement, tel que :

$$(\forall x \in C), (\|f(x) - \text{SIF}(x)\| < \varepsilon) \quad (\text{II.7})$$

La propriété d'approximation universelle ne donne pas une méthode de construction du système flou, mais elle garanti seulement son existence avec un degré de précision donné défini par le nombre des règles floues [92].

II.3 Systèmes Flous type-2 (SF2)

Comme les systèmes flous type-1, les systèmes d'inférence flous type-2 utilisent les mêmes notions et concepts : ensembles flous, règles floues et inférence, ...etc.

II.3.1 Ensembles flous type-2 (EF2)

L'ensemble flou type-2 (EF2) est défini comme une extension de l'ensemble flou type-1 [69]. Il est caractérisé par une fonction d'appartenance dans l'intervalle $[0,1]$, et le degré d'appartenance de chaque élément de cette fonction (de l'ensemble flou) est un ensemble flou dans l'intervalle $[0,1]$. Les ensembles flous de type-2 sont très efficaces pour modéliser les incertitudes, minimiser leurs effets dans la base de règles et qui peuvent être utilisés dans des situations où l'incertitude (linguistique et numérique) est présente soit sur la forme des fonctions d'appartenance, ou soit sur l'un de ses paramètres [76]. Donc, les ensembles flous type-1 peuvent être considérés comme une approximation du premier ordre de l'incertitude,

alors que les ensembles flous type-2 seront considérés comme approximation du deuxième ordre [72, 75].

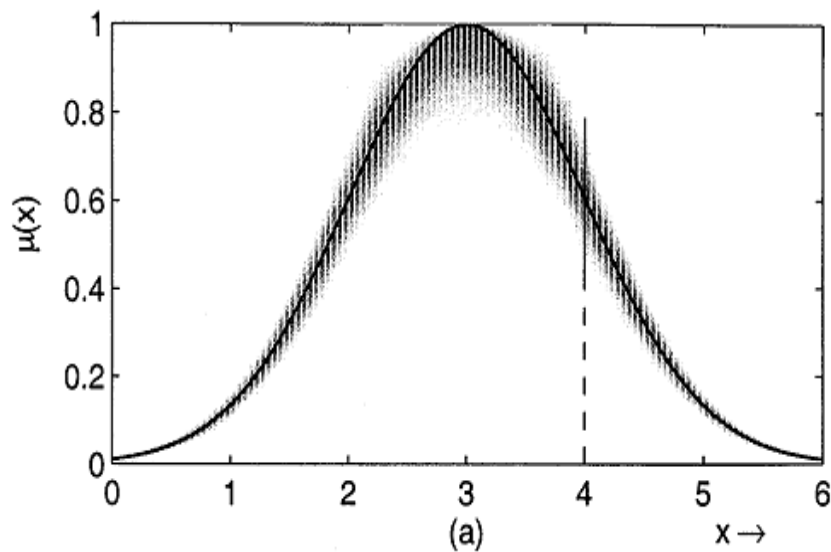
Un ensemble flou type-2 noté \tilde{A} de X (collection d'objets appelée univers de discours) est défini par sa fonction d'appartenance tridimensionnelle [69, 72], notée, $\mu_{\tilde{A}}(x, \mu)$, telle que :

$$\tilde{A} = \{(x, \mu), \mu_{\tilde{A}}(x, \mu) \mid \forall x \in X, \forall \mu \in J_x \subseteq [0, 1]\} \quad (\text{II.8})$$

(Où J_x est le domaine de la fonction d'appartenance secondaire, qui est appelé appartenance primaire de x).

Une fonction d'appartenance de type-2 est une fonction à deux variables. Pour chaque x de l'univers du discours X et de degré d'appartenance primaire μ est appelé variable primaire [76], l'appartenance secondaire $\mu_{\tilde{A}}$ est un nombre ordinaire dans l'intervalle $[0, 1]$. La fonction d'appartenance peut être définie par [75]:

$$\mu_{\tilde{A}}(x, \mu) : X \times [0, 1] \rightarrow [0, 1] \quad (\text{II.9})$$



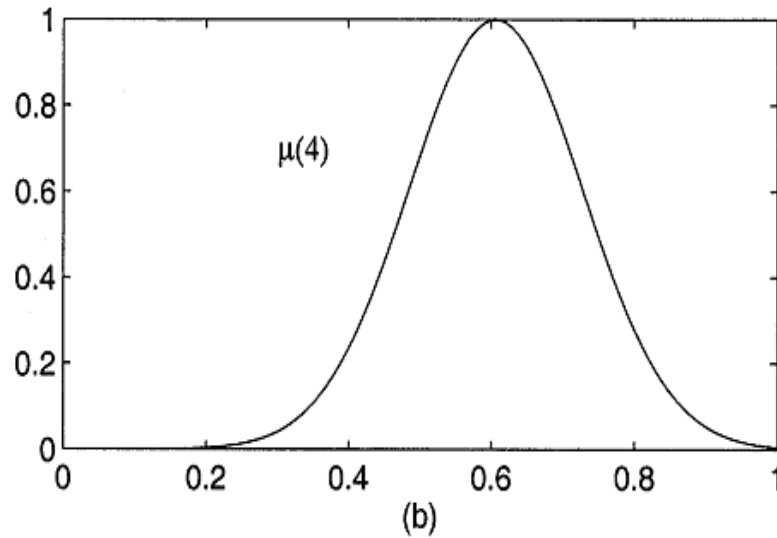


Figure II.5 (a) Représentation d'un ensemble flou Gaussien type-2
(b) fonction d'appartenance flou correspondant à $x=4$

La figure II. 5 (a) illustre l'ensemble flou gaussien de type-2, dans lequel le degré d'appartenance de chaque point de l'ensemble, est un ensemble flou gaussien (figure II.5 (b) pour $x=4$). Les appartenances secondaires gaussiennes sont caractérisées par de moyenne $m(x)$ et d'un écart type σ_m , et la fonction d'appartenance secondaire pour chaque x est défini par :

$$\mu_{\tilde{A}}(x, u) = e^{-\left[\frac{1}{2} \left(\frac{\mu - m(x)}{\sigma_m} \right)^2 \right]} \quad (\text{II.10})$$

Avec $\mu \in [0,1]$ et $m(x)$ est un ensemble flou type-1 gaussien de moyenne M et d'ecart type σ_x , tel que :

$$m(x) = e^{-\left[\frac{1}{2} \left(\frac{x-M}{\sigma_x} \right)^2 \right]} \quad (\text{II.11})$$

D'où

$$\mu_{\tilde{A}}(x, u) = e^{-\left[\frac{1}{2} \left(\frac{\mu - \exp\left(-\frac{1}{2} \left(\frac{x-M}{\sigma_x} \right)^2 \right)}{\sigma_m} \right)^2 \right]} \quad (\text{II.12})$$

II.3.2 Type des ensembles flous type-2

Les formes des ensembles flous les plus connus sont [69, 76] :

- **Ensemble flou type-2 gaussien:** Le degré d'appartenance de chaque point est un ensemble flou type-1 gaussien dont le domaine de définition est inclus dans l'intervalle $[0, 1]$.
- **Ensemble flou type-2 triangulaire:** Le degré d'appartenance de chaque point est un ensemble flou type-1 triangulaire dont le domaine de définition est inclus dans l'intervalle $[0, 1]$.
- **Ensemble flou type-2 intervalle:** Le degré d'appartenance de chaque point est un ensemble ordinaire dont le domaine de définition est inclus dans l'intervalle $[0, 1]$.

II.3.3 Principe d'extension monodimensionnel

Le degré d'appartenance de deux ensembles flous type-2 \tilde{A} dans X et \tilde{B} dans X sont des ensembles type-1 (ensembles flous type-1 dans $J \subseteq [0,1]$), qui sont représentés par :

$$\mu_{\tilde{A}}(x) = \int_{\mu} f_x(\mu) / \mu; \mu \in J \quad (\text{II.13})$$

$$\mu_{\tilde{B}}(x) = \int_w g_x(w) / w; w \in J \quad (\text{II.14})$$

Où : $\mu \in J$ et $\omega \in J \subseteq [0,1]$, indique les degrés d'appartenance primaire de x .

$f_x(u) \in [0,1]$ et $g_x(u) \in [0,1]$, indique les degrés d'appartenance secondaire de x .

Pour accomplir les opérations sur les ensembles flous type-2 \tilde{A} et \tilde{B} (l'union, l'intersection, le complément, t-conorm et t-norm), on doit utiliser le principe d'extension de Zadeh [72, 75, 76] :

<p>Union floue : $(\tilde{A} \cup \tilde{B})$</p>	<p>$\forall x \in U, \mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}} \cup \mu_{\tilde{B}} = \int_{\mu} \int_w (f_x(\mu) * g_x(w)) / (u \vee w)$</p> <p>Le symbole «*» représente la t-norme choisi et \vee représente l'opération t-conorme (le maximum).</p>
<p>Intersection floue: $(\tilde{A} \cap \tilde{B})$</p>	<p>$\forall x \in U, \mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}} \cap \mu_{\tilde{B}} = \int_{\mu} \int_w (f_x(\mu) * g_x(w)) / (u * w)$</p>
<p>Complémentation floue : $\tilde{\tilde{A}}$</p>	<p>$\forall x \in U, \mu_{\tilde{\tilde{A}}}(x) = \neg \mu_{\tilde{A}}(x) = \int_{\mu} f_x(\mu) / (1 - \mu)$</p>

II.3.4 Structure des Systèmes flous type-2

Dans cette section, nous supposons que les fonctions d'appartenance des prémisses et des conséquences sont de type-2. La figure (II. 7) représente la structure de base d'un système flou de type-2 qui est très similaire à la structure d'un système flou de type-1 [75]. Il est composé essentiellement de: *fuzzification, une base de règles, un mécanisme d'inférence et un module de traitement de la sortie.*

La seule différence avec le système flou type-1 est l'utilisation un réducteur de type pour convertir les ensembles flous type-2 à la sortie du système d'inférence en ensembles flous type-1 avant la phase de défuzzification.

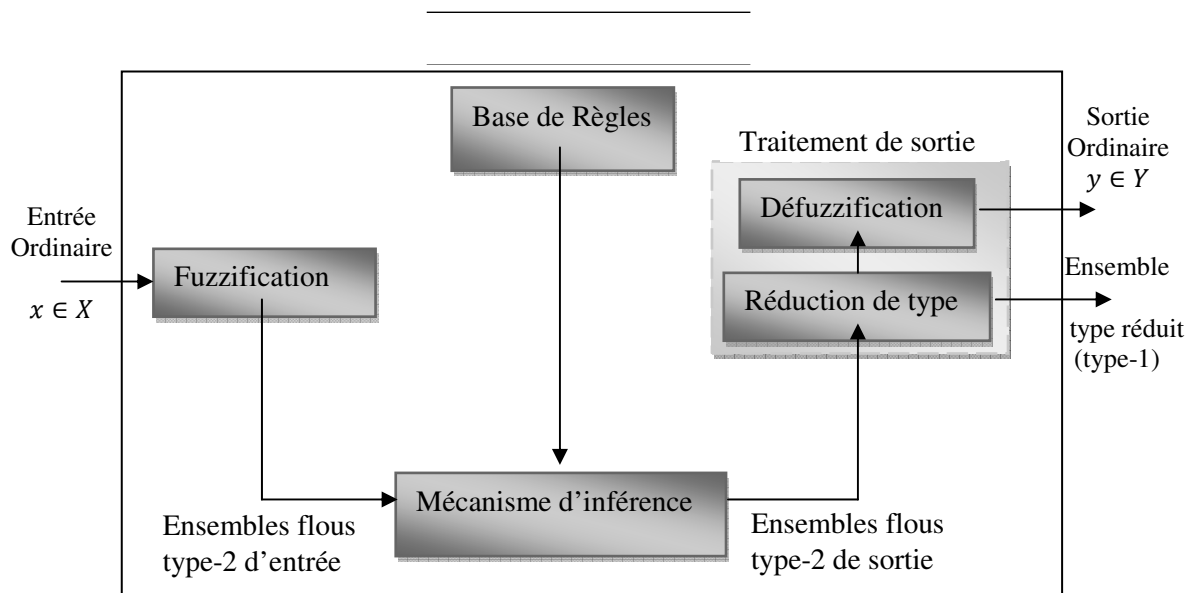


Figure II.7 Structure de base d'un système flou de type-2

II.3.4.1 Fuzzification :

L'interface de fuzzification fait correspondre à l'entrée déterministe un EFT-2 [75, 76]. Contrairement à la fonction d'appartenance type-1. La fonction d'appartenance type-2 donne plusieurs degrés d'appartenance (ou dimensions) pour chaque entrée. Par conséquent, l'incertitude sera mieux représentée. Cette représentation va nous permettre de tenir compte de ce qui a été négligé par le type-1.

II.3.4.2 Base de règles

Comme mentionné précédemment, la différence entre un système flou type-1 et celui du type-2 réside seulement dans la nature des fonctions d'appartenance des prémisses et des conséquences.

La structure des règles reste exactement la même dans le cas des systèmes flous type-1. La $i^{\text{ème}}$ règle dans un système flou type-2 prend la forme suivante :

$$R^{(i)} : SI \ x_1 \ est \ \tilde{A}_1^i \ et \ x_2 \ est \ \tilde{A}_2^i \ et \dots \ x_p \ est \ \tilde{A}_p^i \ ALORS \ y \ est \ \tilde{B}^i \quad (II.15)$$

Où les x_i sont des entrées, \tilde{A}_i^i sont des ensembles antécédents tel que $(i = 1 \dots, p)$, y est la sortie, et \tilde{B}^i est l'ensemble conséquent.

II.3.4.3 Mécanisme d'inférence :

Le mécanisme d'inférence dans un système flou type-2 calcul un ensemble flou de sortie type-2 correspondant à chaque règle. Alors, il utilise la base de règles floues (II.15) pour effectuer une relation entre un vecteur d'entrée $\underline{x} = (x_1, x_2, \dots, x_p) \in (X_1, X_2, \dots, X_p)$ et la sortie scalaire $y \in Y$.

Notons la fonction d'appartenance de cette relation floue type-2 par $\mu_{\tilde{A}_1^i \times \dots \times \tilde{A}_p^i \rightarrow \tilde{B}^i}(x, y)$ où $\tilde{A}_1^i \times \dots \times \tilde{A}_p^i$ dénote le produit cartésien de $\tilde{A}_1^i, \dots, \tilde{A}_p^i$ et $\underline{x} = (x_1, x_2, \dots, x_p)^T$. Quand une entrée x' est appliquée, la composition de l'ensemble flou \tilde{X}^i , à qui x' appartient, et la règle R^i , est calculée en utilisant une version étendue de la composition "sup-star" [76]:

$$\mu_{\tilde{X}^i \circ \tilde{A}_1^i \times \dots \times \tilde{A}_p^i \rightarrow \tilde{B}^i}(y) = \cup_{x \in \tilde{X}^i} \left[\mu_{\tilde{X}^i}(x) \cap \mu_{\tilde{A}_1^i \times \dots \times \tilde{A}_p^i \rightarrow \tilde{B}^i}(x, y) \right] \quad (II.16)$$

Nous utilisons une fuzzification par singleton, c'est à dire que l'ensemble flou \tilde{x}' a un degré d'appartenance égal à 1 pour $x = x'$ et égal a zéro pour toutes les autres entrées ($x \neq x'$) :

$$\mu_{\tilde{x}' \circ \tilde{A}_1^l \times \dots \times \tilde{A}_p^l \rightarrow \tilde{B}^l}(y) = \mu_{\tilde{A}_1^l \times \dots \times \tilde{A}_p^l \rightarrow \tilde{B}^l}(x', y) \quad (\text{II.17})$$

Notons l'ensemble de sortie correspondant à la $l^{\text{ème}}$ règle $\tilde{x}' \circ \tilde{A}_1^l \times \dots \times \tilde{A}_p^l \rightarrow \tilde{B}^l$ par \tilde{G}^l . Le second membre est calculé en utilisant l'implication minimum ou produit (qui correspond à l'opération *meet* avec t-norme min ou produit dans le cas du type-2):

$$\mu_{\tilde{G}^l}(y) = \mu_{\tilde{A}_1^l \times \dots \times \tilde{A}_p^l \rightarrow \tilde{B}^l}(x') \cap \mu_{\tilde{B}^l}(y) \quad (\text{II.18})$$

Généralement, pour cette relation, on utilise le produit ou le minimum, correspondants à l'opération *meet* sous le produit T-norme ou le minimum T-norme respectivement.

$$\mu_{\tilde{G}^l}(y) = \mu_{\tilde{A}_1^l}(x_1) \cap \mu_{\tilde{A}_2^l}(x_2) \cap \dots \cap \mu_{\tilde{A}_p^l}(x_p) \cap \mu_{\tilde{B}^l}(y) = \mu_{\tilde{B}^l}(y) \cap \left[\prod_{i=1}^p \mu_{\tilde{A}_i^l}(x_i) \right] \quad (\text{II.19})$$

où \cap dénote l'opération *meet* basée sur la t-norme choisie.

L'intervalle d'activation associé au $l^{\text{ème}}$ ensemble flou de sortie est défini par :

$$A^l(x) = [\underline{e}^l(x), \bar{e}^l(x)] \quad (\text{II.20})$$

Où :

$$\underline{e}^l(x) = \underline{\mu}_{\tilde{A}_1^l} e(x_1) * \underline{\mu}_{\tilde{A}_2^l} e(x_2) * \dots * \underline{\mu}_{\tilde{A}_p^l} e(x_p) \text{ et } \bar{e}^l(x) = \bar{\mu}_{\tilde{A}_1^l} e(x_1) * \bar{\mu}_{\tilde{A}_2^l} e(x_2) * \dots * \bar{\mu}_{\tilde{A}_p^l} e(x_p) \quad (\text{II.21})$$

Les termes $\underline{\mu}_{\tilde{A}_p^l} e(x_p)$ et $\bar{\mu}_{\tilde{A}_p^l} e(x_p)$ sont les degrés d'appartenance inférieur et supérieur relatifs à $\mu_{\tilde{A}_p^l} e(x_p)$.

II.3.4.4 Module de traitement de la sortie

Cette partie se compose de deux blocs, *le réducteur de type* qui transforme un ensemble flou type-2 de la sortie du système d'inférence, à un ensemble flou type-1, et *l'interface de défuzzification*, qui défuzzifié l'ensemble flou type-1 résultant (ensemble de type réduit) pour obtenir un nombre ordinaire [72, 75].

- **Réduction de type**

Pour transformer la sortie floue type-2 en un ensemble flou type-1, la méthode des centres d'ensembles (center of sets en anglais (cos)) est utilisée [69]. Cette méthode consiste à remplacer chaque ensemble de sortie par son centroïde qui est lui-même un ensemble flou type-1 si l'ensemble de sortie est de type-2, puis, calculer la moyenne pondérée de ces centroïdes, où le poids associé au $l^{\text{ième}}$ centroïde sera le degré d'activation de la $l^{\text{ième}}$ règle $\prod_{i=1}^p \mu_{\tilde{A}_i}(x_i)$.

L'expression de l'ensemble flou de type réduit par cette méthode est donnée sur l'équation (II. 22) obtenue par extension

$$Y_{\text{cos}}(x) = \int \dots \int \int \dots \int \int \mathfrak{S}_{d_l}^M \mu_{C_l}(d_l) \wedge \mathfrak{S}_{e_l}^M \mu_{E_l}(e_l) \left/ \frac{\sum_{l=1}^M d_l e_l}{\sum_{l=1}^M e_l} \right. \quad (\text{II.22})$$

où : \mathfrak{S} et \wedge indique les t-normes choisies (produit ou minimum) ;

$d_l \in C_l = C_{\tilde{B}^l}$, le centroïde du $l^{\text{ième}}$ ensemble de sortie;

et $e_l \in E_l = \prod_{i=1}^p \mu_{\tilde{A}_i}(x_i)$, le degré (intervalle) d'activation associé au $l^{\text{ième}}$ ensemble de sortie, pour $l=1, \dots, M$. Les choses se simplifient beaucoup lorsque les fonctions d'appartenance secondaires sont de type intervalle.

y_L, y_R sont deux points de gauche et de droite caractérisant l'ensemble réduit. Afin de calculer les points extrêmes y_L, y_R Karnik et Mendel [72] ont développé un algorithme itérative dont la procédure est donnée comme suit :

Pour calculer $Y_{\text{cos}}(x)$, nous devons suivre les étapes suivantes [72, 76]:

- 1- Discrétiser l'espace de sortie Y en un nombre convenable de points, puis calculer les centroïdes C_l de chaque ensemble de sortie. Les résultats doivent être mémorisés pour une future utilisation.
- 2- Calculer les degrés d'activation $E_l = \prod_{i=1}^p \mu_{\tilde{A}_i}(x_i)$ pour chaque règle.
- 3- Discrétiser le domaine de chaque C_l (C_l est le centroïde du $l^{\text{ième}}$ ensemble de sortie) en un nombre convenable de points, soit $M_l (l=1, \dots, M)$.

- 4- Discrétiser le domaine de chaque E_l en un nombre convenable de points, soit N_l ($l=1, \dots, M$).
- 5- Enumérer toutes les combinaisons possibles $\{d_1, \dots, d_m, e_1, \dots, e_m\}$ tel que $d_l \in C_l$ et $e_l \in E_l$. Le nombre total des combinaisons sera $\prod_{j=1}^M M_j N_j$.
- 6- Calculer l'ensemble de type réduit $Y_{\cos}(x)$.

Calcul y_L :

- 1- Discrétiser l'espace de sortie Y en un nombre convenable de points, puis calculer les centroides C_l de chaque ensemble de sortie. Les résultats doivent être mémorisés pour une future utilisation.
- 2- Trier \underline{d}_l dans un ordre croissant : $\underline{d}_1 \leq \underline{d}_2 \leq \dots \leq \underline{d}_M$
- 3- Initialiser e_l en prenant comme point de départ : $e_l = \frac{e_l + \bar{e}_l}{2}, l = 1, \dots, M$

$$\text{ensuite calculer : } y_L = \frac{\sum_{l=1}^M \underline{d}_l e_l}{\sum_{l=1}^M e_l} \quad (\text{II.23})$$

et poser $y_L' = y_L$

- 4- Trouver un point de commutation N ($1 \leq N \leq M - 1$) qui satisfait : $\underline{d}_l \leq y_L' \leq \underline{d}_{l+1}$
- 5- Pose : $e_l = \begin{cases} \bar{e}_l, l < N \\ \underline{e}_l, l > N \end{cases}$

$$\text{Calculer : } y_L = \frac{\sum_{l=1}^M \underline{d}_l e_l}{\sum_{l=1}^M e_l}$$

et poser $y_L'' = y_L$

- 6- Vérifier if $y_L'' = y_L'$, si oui, arrêter et poser $y_L = y_L''$, si non, aller à l'étape 7.
- 7- Pose $y_L' = y_L''$ et aller à l'étape 4.

Calcul y_R :

- 1- Discrétiser l'espace de sortie Y en un nombre convenable de points, puis calculer les centroides C_l de chaque ensemble de sortie. Les résultats doivent être mémorisés pour une future utilisation.

2- Trier \bar{d}_l dans un ordre croissant : $\bar{d}_1 \leq \bar{d}_2 \leq \dots \leq \bar{d}_M$

3- Initialiser e_l en prenant comme point de départ : $e_l = \frac{\underline{e}_l + \bar{e}_l}{2}, l = 1, \dots, M$

$$\text{ensuite calculer : } y_R = \frac{\sum_{l=1}^M \bar{d}_l e_l}{\sum_{l=1}^M e_l} \quad (\text{II.24})$$

et poser $y_R' = y_R$

4- Trouver un point de commutation N ($1 \leq N \leq M - 1$) qui satisfait : $d_l \leq y_R' \leq d_{l+1}$

5- Pose : $e_l = \begin{cases} \bar{e}_l, l < N \\ \underline{e}_l, l > N \end{cases}$

$$\text{Calculer : } y_R = \frac{\sum_{l=1}^M \bar{d}_l e_l}{\sum_{l=1}^M e_l} \text{ et poser } y_R'' = y_R$$

6- Vérifier if $y_L'' = y_L'$, si oui, arrêter et poser $y_R = y_R''$, si non, aller à l'étape 7.

7- Pose $y_R' = y_R''$ et aller à l'étape 4.

- Défuzzification

Le type réduit sera déterminé par ses deux points extrêmes de droite et de gauche respectivement y_L et y_R [72, 76]. En appliquant le centre de gravité au type réduit, la sortie numérique sera donnée par :

$$Y = \frac{y_L + y_R}{2} \quad (\text{II.25})$$

Contrairement aux systèmes flous type-1 où la sortie est caractérisée par un seul vecteur de fonctions de bases flous (FBF), un système flou type-2 est caractérisé par deux vecteurs de FBF. Dans ce cas y_L peut être écrit comme suit :

$$y_L = \frac{\sum_{l=1}^M \underline{d}_l e_l}{\sum_{l=1}^M e_l} = \sum_{l=1}^M \underline{d}_l \xi_{Ll} = \underline{d}^T \underline{\xi}_L(x) \quad (\text{II.26})$$

où e_l désigne le degré d'activation (soit \underline{e}_l ou \bar{e}_l) contribuant au point extrême de gauche y_L ,

$\xi_{Ll} = \frac{e_{Ll}}{\sum_{l=1}^M e_{Ll}}$ sont les composants de premier vecteur de FBF, $\underline{\xi}_L^T(x) = [\xi_{L1}, \xi_{L2}, \dots, \xi_{LM}]$ et

$\underline{y}_L^T = [\underline{d}_1, \underline{d}_2, \dots, \underline{d}_M]$ est la conclusion du système flou type-2.

De même :

$$y_R = \frac{\sum_{l=1}^M \bar{d}_l e_l}{\sum_{l=1}^M e_l} = \sum_{l=1}^M \bar{d}_l \xi_{L_l} = \bar{d}^T \underline{\xi}(x) \quad (\text{II.27})$$

où e_l désigne le degré d'activation (soit e_l ou \bar{e}_l) contribuant au point extrême de gauche y_R ,

$\xi_{R_l} = \frac{e_{R_l}}{\sum_{l=1}^M e_{R_l}}$ sont les composants de premier vecteur de FBF, $\underline{\xi}_R^T(x) = [\xi_{R1}, \xi_{R2}, \dots, \xi_{RM}]$ et

$\underline{y}_R^T = [\bar{d}_1, \bar{d}_2, \dots, \bar{d}_M]$ est la conclusion du système flou type-2.

Finalement, la sortie globale peut être donnée par :

$$Y = \frac{y_L^T \underline{\xi}_L + y_R^T \underline{\xi}_R}{2} \quad (\text{II.28})$$

II.4 Conclusion

Ce chapitre est une présentation des systèmes flous type-1 et type-2. , nous avons étudié en détail les notions des ensembles flous, les opérations et les normes dans ces systèmes de traitement, les deux structures des systèmes flous type-1 et type-2 sont détaillées. Nous avons détaillés le bloc de traitement de la sortie dans le cas des systèmes flous type-2, qui exige d'abord la réduction de type, permettant de réduire des ensembles flous de type-2 à des ensembles flous de type-1 puis la défuzzification.

Dans cette thèse, on s'intéresse à l'utilisation des systèmes flous les plus couramment utilisés dans la commande des processus, à savoir les systèmes flous type-1 et type-2 de Takagi-Sugeno à conclusion constante (d'ordre zéro).

Dans le chapitre suivant, nous allons appliquer les contrôleurs flous type-1 et type-2 pour contrôler le mouvement automatique d'un robot mobile. Les structures proposées et les résultats de simulation seront discutés.

Chapitre III

Navigation Visuelle par des Contrôleurs Flous type-1 et le Flux Optique

III.1 Introduction

La navigation et l'évitement d'obstacles sont des tâches importantes pour les applications de robots mobiles dans les environnements intérieurs et extérieurs. Dans des environnements inconnus et complexes, les robots doivent disposer de systèmes de contrôle puissants et des capteurs de perception précis [04].

La navigation autonome d'un robot mobile dans des environnements statiques ou dynamiques est une tâche importante dans le domaine de la robotique mobile [36].

Pour faire la tâche de navigation autonome avec des bonnes performances, nous allons utiliser une méthode très efficace basée sur les comportements [77-79].

La méthode de navigation basée sur le comportement présente un outil efficace pour subdiviser la tâche de navigation globale en petites sous-tâches [14, 93] : évitement des obstacles, suivi des murs, recherche d'objectifs, poursuite de cibles,...., etc.

Une étude sur l'utilisation des techniques du flux optique pour les applications robotiques est présentée dans [18].

D'autre part, les contrôleurs à logique floue (FLC) ont la capacité de gérer et de traiter les incertitudes dans les systèmes robotiques [23].

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 leur complexité (ils peuvent être vastes, imprécis, dynamiques ou inconnus,...). Ce qui nous amène à la définition de certaines comportements élémentaires tels que: évitement d'obstacles, convergence vers un but, suivi de murs, et poursuite de trajectoire ou d'une cible mobile,...etc. Nous nous sommes intéressés à l'utilisation de la méthode de flux optique Horn et Schunck et des systèmes à logique floue de type-1 pour le déplacement du robot mobile d'une position initiale vers une quelconque destination désirée en évitant les obstacles, tout en respectant les contraintes cinématiques du robot et de façon autonome. La simulation est faite dans un environnement 3D en utilisant le toolbox VRML (*Virtual Reality Modeling Language*).

Dans ce travail, pour simplifier la tâche de navigation autonome, on a utilisé la structure basée sur les comportements (*structure de Subsumption*).

III.2 Travaux liés à la contribution

La tâche de navigation est l'une des missions les plus importantes utilisées dans le domaine de la robotique [8, 9, 10]. Différentes méthodes ont été proposées pour résoudre ce problème. La tâche de navigation est devenue une application importante basée sur les systèmes de vision [2, 11, 12].

Les méthodes de navigation basée sur la vision ont été proposées dans les deux types d'environnements : intérieur ou extérieur [7, 13]. Quelques applications sur le thème de navigation visuelle seront résumées comme suit :

- Les auteurs dans le papier [14], présentent une étude théorique sur les techniques utilisées pour la navigation visuelle des différents systèmes robotisés : robots mobiles, les véhicules aériens et les véhicules sous-marins autonomes. Deux approches principales sont discutées : la navigation par carte et la navigation sans carte de l'environnement.
- Dans [7], les méthodes de navigation et de localisation basées sur la vision pour les robots mobiles sont étudiées, et plus particulièrement concernant leurs applications dans des environnements intérieurs et extérieurs.
- Les auteurs dans [15] ont proposé une méthode de navigation visuelle pour un robot mobile en temps réel basée sur le suivi d'objets et la tâche d'évitement des obstacles. Un système de vision stéréoscopique est utilisé pour détecter la cible désirée tandis que le télémètre laser est utilisé pour éviter la collision avec les objets de l'environnement.
- Dans [13], les auteurs ont introduit l'utilisation de la vision monoculaire pour la navigation robotique dans un environnement dynamique et modifié. La solution proposée pour détecter les obstacles est basée sur la segmentation des couleurs et la détection des contours.
- Dans [17], les auteurs ont étudié la tâche d'évitement des obstacles en utilisant une stratégie d'équilibrage basée sur la quantité de flux latéraux gauche et droit. La profondeur est calculée à partir de deux images consécutives. Cette technique permet au robot mobile de se déplacer en toute sécurité sans collision.
- Wang et al. 2015 [03] ont proposé une méthode améliorée basée sur le flux optique pour l'évitement d'obstacles à utiliser par un quadrator équipé d'une caméra monoculaire. Il s'agit d'une stratégie équilibrée basée sur la méthode du flux optique

et d'un (temps avant collision) (en Anglais : TTC, Time-to-Collision) rapide pour extraire les informations de profondeur relatives de l'environnement. Le système contrôlé est capable d'éviter efficacement les obstacles.

- D'autres applications des techniques de flux optique pour les applications robotiques sont présentées dans [18].
- Dans [25], les auteurs ont introduit un chemin flou suivant une méthode de navigation visuelle basée sur les tâches. Pour détecter le chemin souhaité, une caméra CCD est utilisée sur le robot mobile. De chaque image captée, l'information utile est extraite, et le contrôleur flou élaboré peut guider le robot pour suivre le chemin efficacement.
- Guzel et al. 2011 [02] ont considéré l'évitement des obstacles basé sur la vision en utilisant une approche logique floue. La tâche d'évitement d'obstacles est étudiée à l'aide de la transformation des caractéristiques invariantes d'échelle (SIFT) et de la logique floue. Cette méthode s'inspire du principe de l'appariement des caractéristiques, selon lequel les descripteurs basés sur SIFT utilisent des scènes texturées et structurées pour faire correspondre les images. Le contrôleur flou génère l'action de commande de direction pour éviter les collisions.

III.3 Structures de Navigation basée sur les comportements

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 [18, 77-79, 92].

Ces comportements sont indispensables à l'exécution des sous-tâches spécifiques pour le robot mobile (éviter les obstacles, converger vers le but, suivre un mur,...). Cette architecture proposée par Brooks est appelée *subsumption* [77]. La figure III.1 présente cette structure où comportements sont arrangés dans les niveaux de priorité. En cas de déclenchement, le comportement de plus haut niveau supprime tous les comportements de niveau inférieur [92].

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 [18], et offre la possibilité de supprimer ou d'ajouter de nouveaux comportements pour

le 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. Chaque comportement peut avoir des entrées provenant des capteurs du robot (caméra, ultrasons, infrarouges, tactiles ...) et/ou d'autres comportements dans le système, et envoyer des sorties aux actionneurs du robot (moteurs, roues, pinces, bras ...) et/ou à d'autres comportements. D'autres auteurs ont utilisé la logique floue pour la représentation et la coordination des comportements comme [42, 80 - 82, 92]. La figure III.2 présente la structure d'un système de contrôle basé sur les comportements flous utilisé pour la commande des robots mobiles.

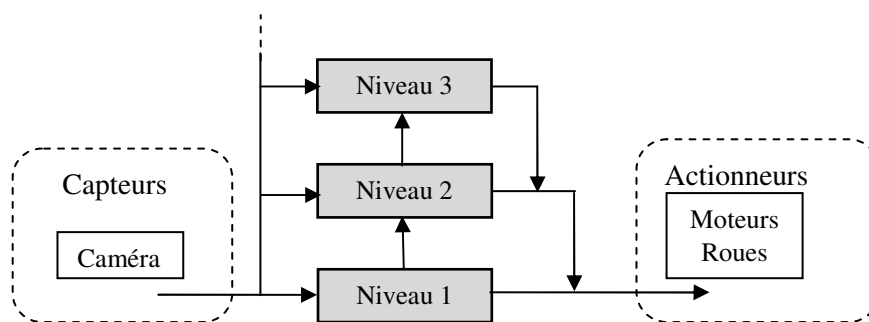


Figure III.1 Structure subsumption

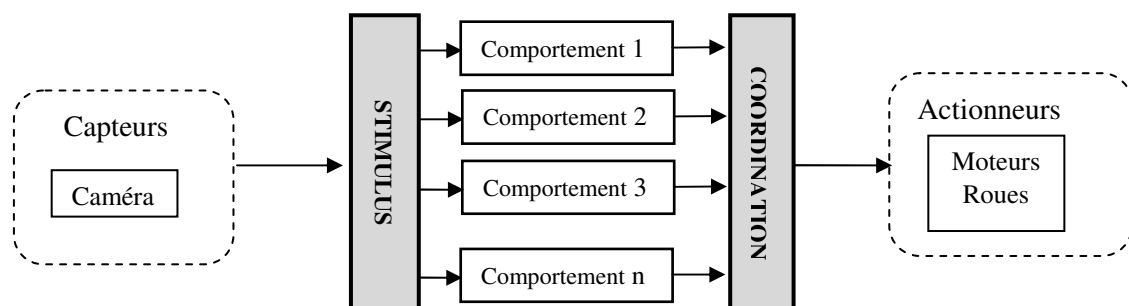


Figure III.2 Architecture basée sur les comportements

Dans la section suivante, on va présenter les étapes de conception et de modélisation des structures de commande automatique proposées pour le robot mobile en utilisant des contrôleurs flous type-1 de Takagi-Sugeno d'ordre zéro, et l'algorithme Horn-Schunck du flux optique. Une hybridation entre ces deux approches sera discutée.

III.4 Modélisation par VRML

Les étapes de modélisation du robot avec ces parties constitutives et de son environnement seront présentées. Nous avons utilisé le toolbox VRML (Virtual Reality toolbox de Matlab). Les stratégies de contrôle étudiées sont simulées en 2D et en 3D.

III.4.1 Modélisation de l'environnement (Virtual Reality 3D Model)

Les figures III.3 (a-b) montrent les interfaces de l'éditeur VRML (V-Realm Builder 2.0). En utilisant cette bibliothèque VRML, nous avons créé un environnement de navigation virtuel qui imite l'espace réel en 3D contenant des obstacles à la forme des box, le sol, les murs et la destination finale à atteindre (le but). L'environnement 3D conçu est représenté sur la figure III.4(a) en montrant ses différents éléments. L'interface de l'éditeur VRML est présentée aussi sur la figure III. 4(b) [100].

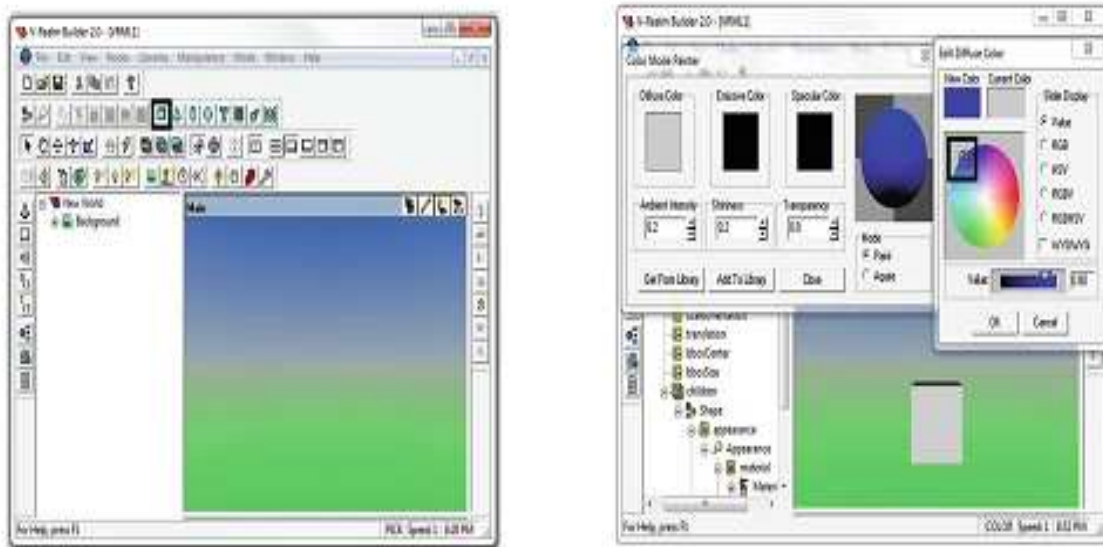
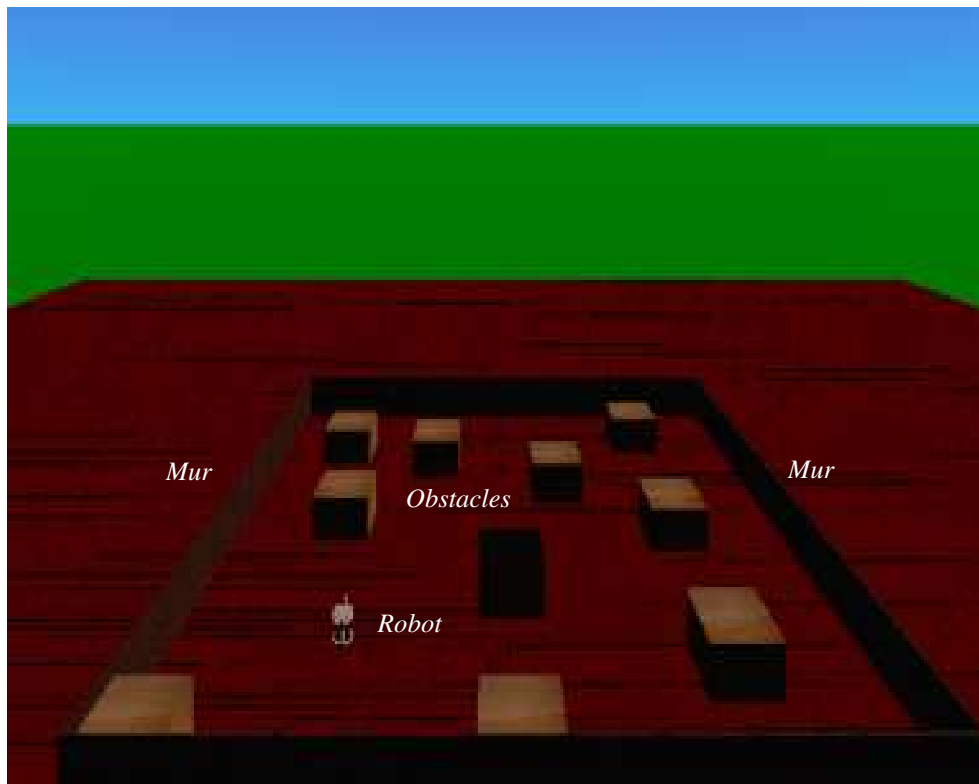
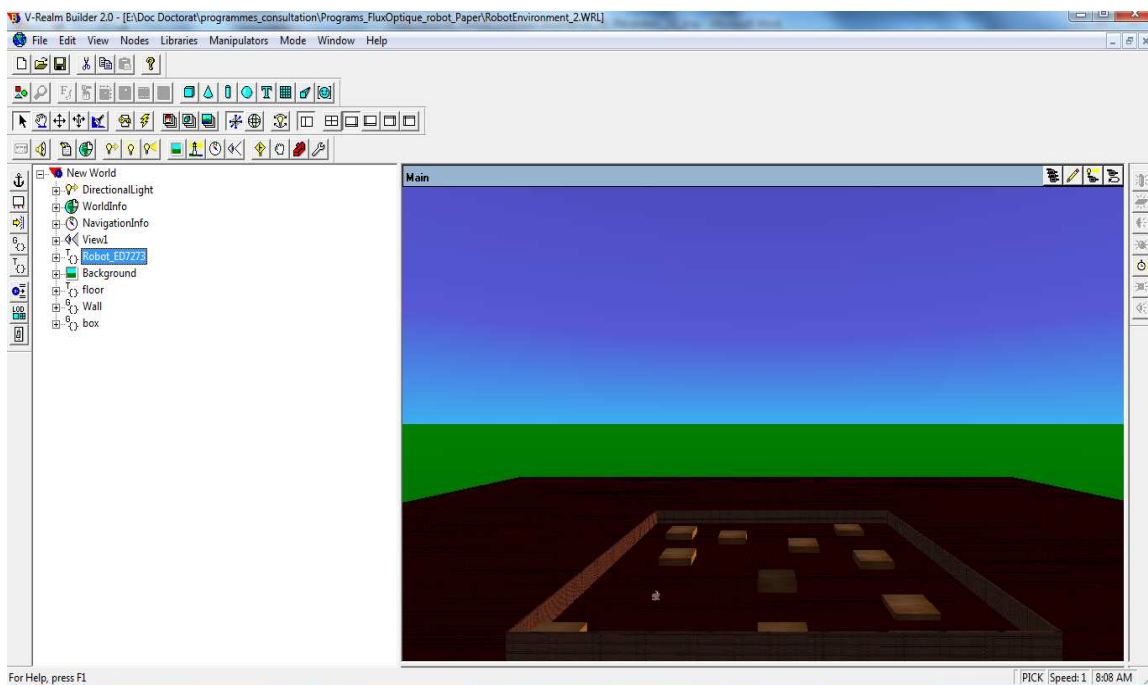


Figure III.3 L'interface de l'éditeur VRML



(a)

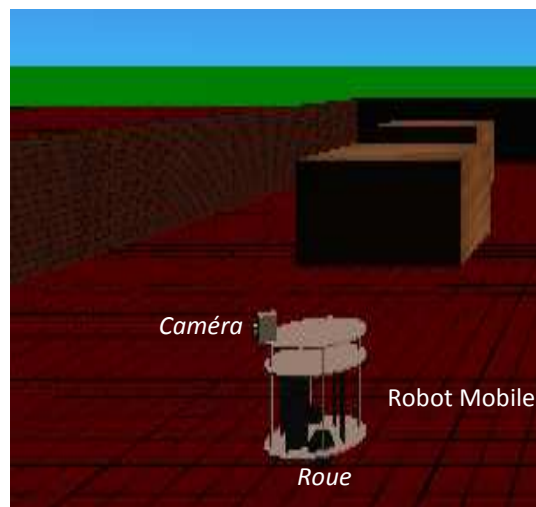


(b)

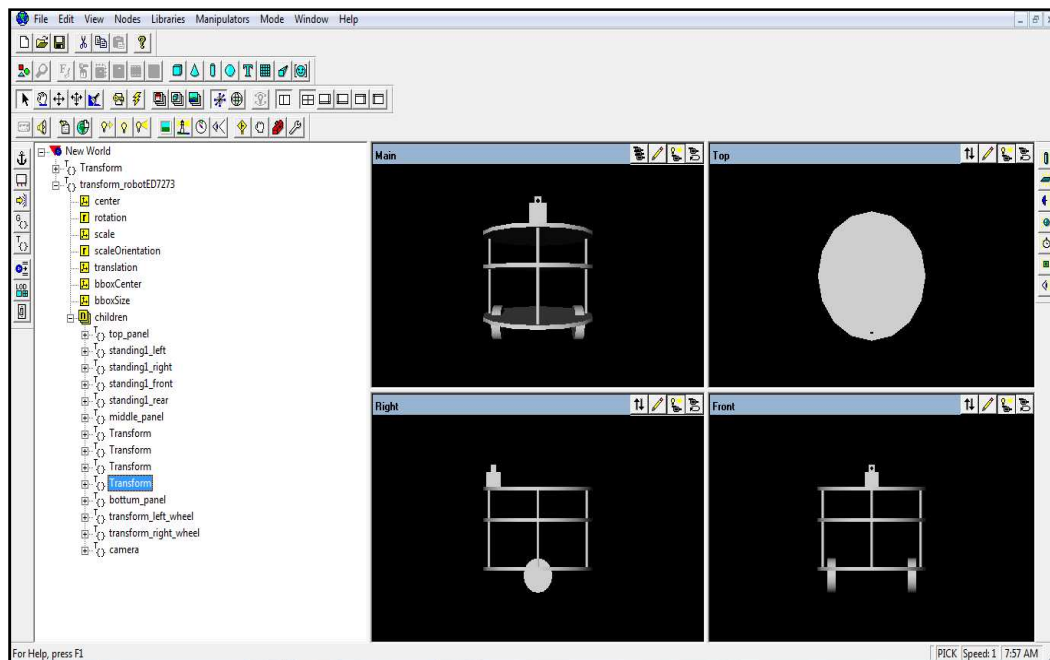
Figure III.4 L'environnement obtenu en utilisant VRML

III.4.2 Modèle du robot mobile

Le robot mobile (RM) en utilisant le VRML est une plate-forme cylindrique avec deux roues motorisées. Il est similaire au robot mobile réel (ED7273). Afin de percevoir son environnement, le robot est doté par une caméra virtuelle, où l'objectif est d'acquérir des informations sur l'environnement qui le contourne. La figure III.5(a) illustre le robot mobile simulé. La figure III.5(b) montre l'interface de conception par VRML.



(a)



(b)

Figure III.5 Robot mobile à roues simulé

En simulation, pour l'application des structures de contrôles proposées, nous utilisons un ensemble des variables comme présenté sur la figure III.6, en donnant la configuration du robot mobile et ces variables calculées dans l'espace du travail [84]. Où :

- R : est le rayon de la plate-forme du robot;
- CIR : est le centre instantané de rotation;
- v_r et v_l : sont les vitesses des roues droite et gauche;
- d_{rg} : est la distance entre le robot et le but à atteindre;
- θ_{rg} : est l'angle entre l'axe du robot et le but.

Le mouvement du robot est basé sur le modèle cinématique non holonomique décrit ci-dessous :

$$\begin{aligned}\dot{x}_r &= v \cos(\theta_r) \\ \dot{y}_r &= v \sin(\theta_r) \\ \dot{\theta}_r &= w\end{aligned}\tag{III. 6}$$

- où :
- x_r et y_r sont les coordonnées du robot ;
 - θ_r est l'angle d'orientation du robot ;
 - v est la vitesse linéaire de translation du robot ;
 - w est la vitesse angulaire calculée à partir de l'angle de braquage α_r .

La version discrète est la suivante :

$$\begin{aligned}x_r(k+1) &= x_r(k) + T \times v(k) \times \cos(\theta_r(k)) \\ y_r(k+1) &= y_r(k) + T \times v(k) \times \sin(\theta_r(k)) \\ \theta_r(k+1) &= \theta_r(k) + T \times w(k)\end{aligned}\tag{III. 7}$$

Avec : T est le pas d'échantillonnage.

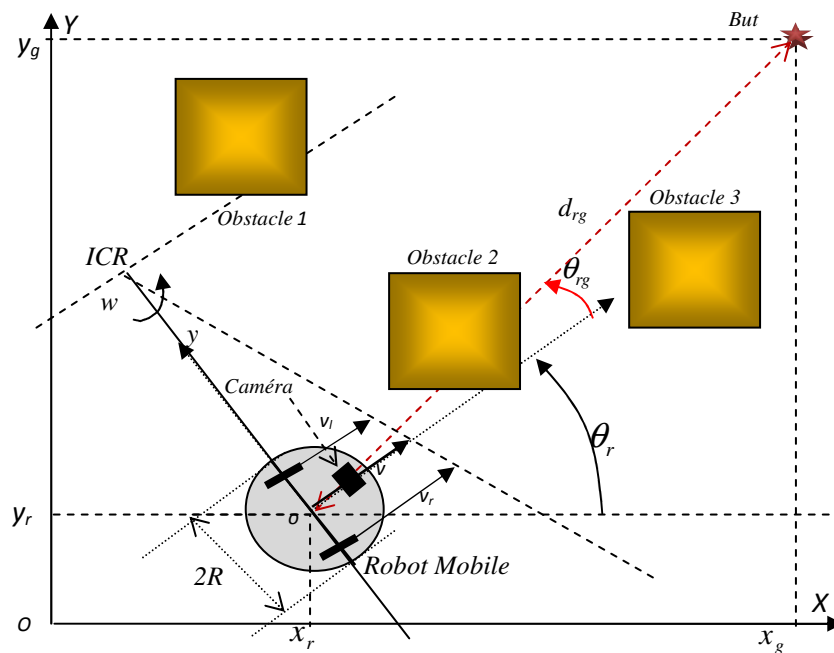


Figure III.6 Modèle du robot mobile

III.5 Système de navigation proposé

Dans ce qui suit, nous présentons le système de navigation proposé. L'objectif est de contrôler le robot mobile à roues pour qu'il se déplace de façon autonome dans son environnement en utilisant l'image acquise par la caméra [84]. Nous avons étudié les différentes tâches du robot (fonctionnalités): évitement des obstacles, convergence vers un but, interception d'une cible en mouvement, navigation vers le but avec évitement des obstacles. Dans ce travail, nous considérons les hypothèses :

- Le robot et le but (la cible) se déplacent sur le même plan,
- Les obstacles et les objets dans l'environnement simulé sont statiques.
- La caméra est montée sur le robot avec une position fixe (sans tenir compte des mouvements panoramique / inclinaison).

Le système de contrôle du robot mobile est basé sur des comportements flous type-1 sous formes des 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 obstacle, converger vers un but, suivre une cible,...) [42,82-84].

Les règles floues peuvent prendre la forme suivante:

R_1 : Si le but est Loin Alors la vitesse est Grande,

R_2 : Si l'obstacle est à droite Alors le braquage est à Gauche,

...

R_n : Si le but est Proche Alors la vitesse est Faible.

III.5.1 Contrôleur d'évitement des obstacles

Tout d'abord, nous allons étudier la tâche d'évitement des obstacles dans l'environnement de navigation. Le schéma fonctionnel de cette stratégie de contrôle est présenté sur la figure III.7 [84].

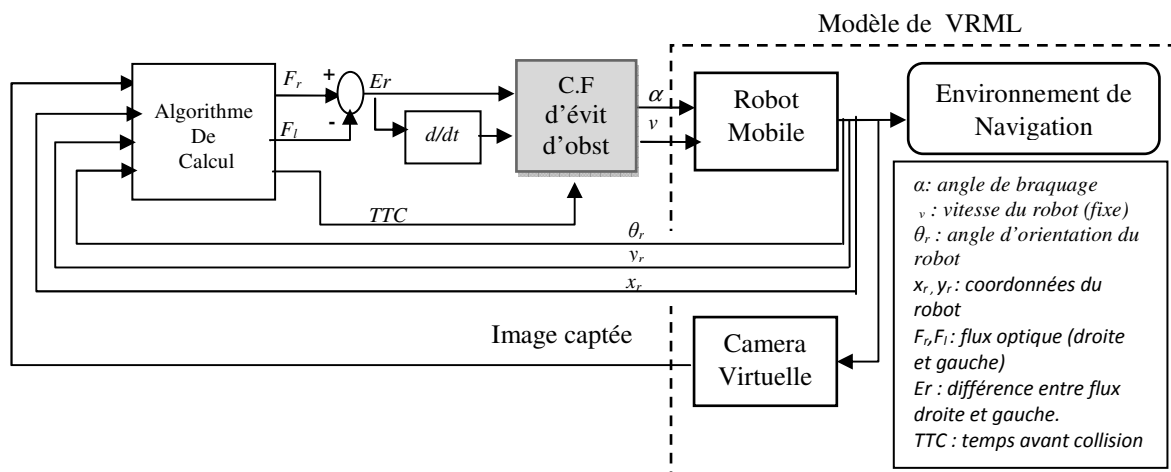


Figure III.7 Navigateur d'évitement des obstacles

Les parties constitutives de ces structures peuvent être résumées comme suit :

Comme présenté précédemment, le robot et son environnement sont modélisés en 3D par VRML.

1. Caméra Virtuelle

Pour percevoir l'environnement de navigation autour du robot mobile, il est doté par une caméra. Dans notre application, l'image comprend 320x480 pixels. L'avantage d'utiliser une seule caméra est son faible coût et son montage facile sur le robot grâce à son poids et sa taille réduite.

2. Algorithme de Calcul du flux optique

Pour le calcul des vecteurs de vitesse dans l'image captée, l'algorithme de Horn-Schunck (HS) a été utilisé. Tout d'abord, au cours du mouvement, et pour chaque pas de temps, l'image de l'environnement est captée par la caméra du robot. Ensuite, les vecteurs de flux optiques sont calculés à partir de la séquence d'images (pour une valeur d'action, deux images sont nécessaires aux moments $t-1$ et t). Les valeurs de flux optiques des pixels de l'image sont calculées en utilisant l'algorithme HS (présenté dans le chapitre 2). Les figures III.8 et III.9 montrent respectivement l'image captée par la caméra en couleur et les valeurs de flux optique générées.

Dans notre application, comme le montre la figure III.10, l'image est divisée en deux parties : droite et gauche afin de pouvoir détecter les obstacles dans les deux côtés. Pour chaque partie de l'image, on calcule les valeurs moyennes du flux optique droite et gauche (F_r et F_l). Ces deux valeurs sont calculées en utilisant les équations : 8, 9.

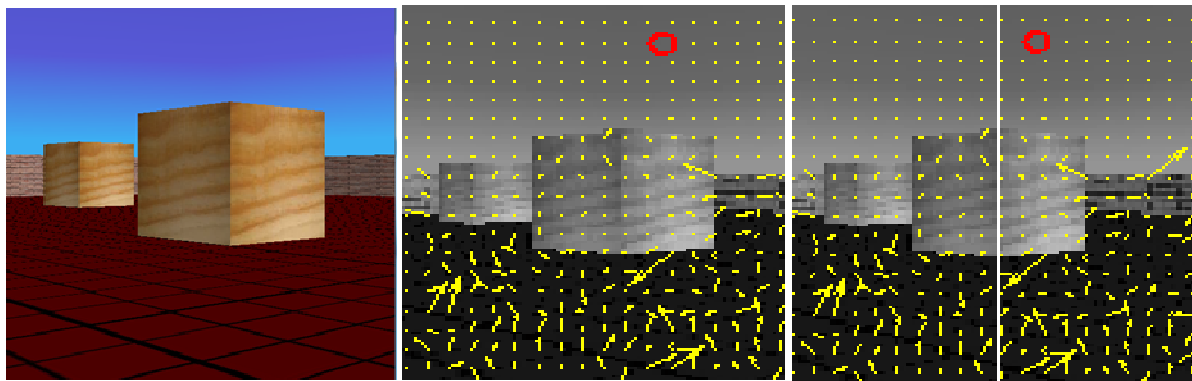


Figure III.8 Image captée **Figure III.9** Flux optique et FOE **Figure III.10** Division de l'image

$$F_r(t) = \frac{1}{N} \sum_{F \in S_r} F(t) \quad (\text{III. 8})$$

$$F_l(t) = \frac{1}{N} \sum_{F \in S_l} F(t) \quad (\text{III. 9})$$

où N : est le nombre des éléments de flux dans la moitié droite ou gauche de l'image, S_l est la moitié gauche de l'image et S_r est la moitié droite.

La troisième variable utilisée est le temps avant collision (TTC) ou (Time To Contact) qui est calculé en utilisant l'équation suivante :

$$TTC = \frac{-Z}{dZ / dt} \quad (\text{III. 10})$$

où : - Z est la distance entre le robot et l'obstacle,

- dZ/dt est la vitesse de la caméra (c'est la même du robot).

Cette variable est utilisée comme condition pour activer le système de contrôle lorsque le robot est à proximité des obstacles. L'erreur entre les flux de coté droit et gauche est notée (E_r) a été calculée avec sa variation (ΔE_r) selon les deux équations 11, 12.

$$E_r(t) = F_l(t) - F_r(t) \quad (\text{III. 11})$$

$$\Delta E_r(t) = E_r(t) - E_r(t-1) \quad (\text{III. 12})$$

3. Contrôleurs Flou type-1 de Takagi-Sugeno

Les deux variables (E_r et ΔE_r) sont utilisés par le système de contrôle basé sur la théorie de la logique floue (Contrôleur flou type-1) pour générer les actions de mouvement adéquates du robot mobile. Les commandes du robot sont : l'angle de braquage α et la vitesse de translation du robot v (comme présenté sur la figure III.7).

Le contrôleur flou (C.F) utilisé est de type Takagi-Sugeno d'ordre zéro (TS0) permettant une simplification de calcul et efficacité de contrôle pour le robot mobile. En plus, la sortie obtenue n'est pas floue, ce qui supprime une étape dans le calcul. La commande inférée est traitée en trois étapes: *fuzzification, inférence et calcul de sortie*.

Afin d'éviter la collision avec les obstacles et les objets les plus proches dans le parcours du robot, nous avons utilisé le contrôleur flou avec deux entrées et une seule sortie (en supposant que le robot se déplace avec une vitesse constante). Les variables d'entrées sont : E_r et ΔE_r sont définies par cinq fonctions d'appartenance illustrées respectivement sur les figures III.11 et III.12. Nous avons utilisé des ensembles flous triangulaires et trapézoïdaux parce qu'ils ont l'avantage de couvrir efficacement et correctement les variables utilisées [66]. L'action de sortie (angle de braquage α) est représentée par des singletons comme le montre la figure III. 13.

Les mots linguistiques utilisées pour les variables d'entrée-sortie sont: **Z** : Zéro, **PP** : Positif Petit, **PG** : Positive Grand, **NG** : Négative Grand, **NP** : Négative Petit.

La stratégie de contrôle flou est exprimée symboliquement par les règles floues résumées dans le tableau III. 1. La base de règles floues s'écrit sous la forme :

$$\text{Si } E_r \text{ est } A_1^i \text{ Et } \Delta E_r \text{ est } A_2^i \text{ Alors } \alpha_g \text{ est } B_1^i \quad (\text{III. 13})$$

où $i=1...25$, $A_1^i... A_2^i$ sont les ensembles flous d'entrée, B_1^i sont les fonctions d'appartenance de l'action de commande (sortie).

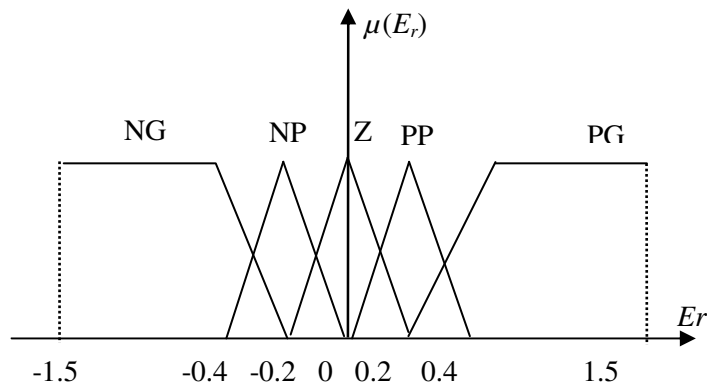


Figure III.11 Fonctions d'appartenance de Er

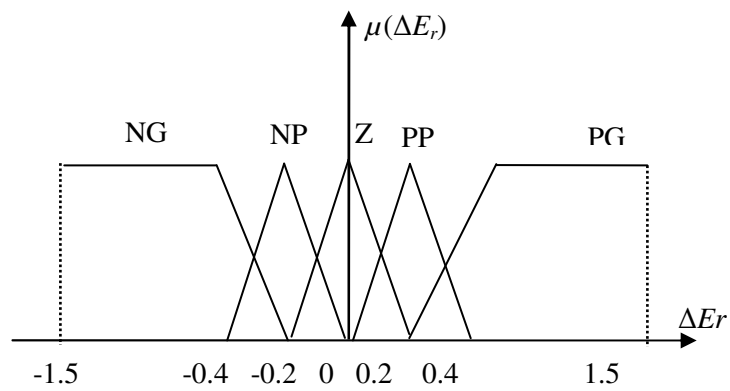


Figure III.12 Fonctions d'appartenance de la variation de l'erreur ΔEr

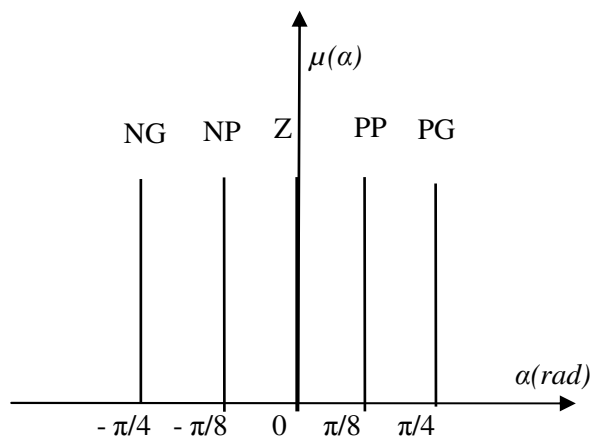


Figure III.13 Valeurs de la commande de sortie

Tableau III.1 Base de règle floue pour l'évitement d'obstacle

ΔE_r	E_r				
	NG	NP	Z	PP	PG
NG	PG	PG	PP	NP	NG
NP	PP	PP	PP	NP	NG
Z	PP	PP	NP	NP	NP
PP	PP	PP	NP	NP	NP
PG	PG	PP	NP	NG	NG

III.5.2 Contrôleur de convergence vers un but

Dans le cas d'une navigation vers une destination désirée (but), le robot doit avoir la capacité de se déplacer dans la direction du but en évitant les obstacles. Un deuxième contrôleur flou type-1 de Takagi-Sugeno (T-S) est utilisé dans un environnement 3D (VRML) pour piloter le robot jusqu'aux coordonnées du but (x_g, y_g) . La structure de contrôle globale est illustrée dans le schéma fonctionnel de la Figure. III. 14. Il se compose de deux contrôleurs flous de T-S (le premier pour l'évitement des obstacles et le deuxième pour la convergence automatique vers le but (cible)).

Ce deuxième contrôleur flou type-1 utilise les deux entrées : la distance entre le robot et le but noté (d_{rg}) et l'angle entre l'axe du robot et l'axe du but noté (θ_{rg}). Ces deux variables sont calculées en utilisant équations suivantes :

$$d_{rg} = \sqrt{(x_g - x_r)^2 + (y_g - y_r)^2} \quad (\text{III. 14})$$

$$\theta_d = \arctg\left(\frac{y_g - y_r}{x_g - x_r}\right) \quad (\text{III. 15})$$

$$\theta_{rg} = \theta_d - \theta_r \quad (\text{III. 16})$$

Les entrées d_{rg} et θ_{rg} sont définies par les fonctions d'appartenance illustrées sur les figures III. 15 et III. 16. Ce contrôleur flou doit générer l'angle de braquage α et la vitesse de translation v . Les actions de contrôle sont représentées par des singletons comme le montrent respectivement sur les Figure III. 17 et Figure III. 18, où : **TP** : Très Proche, **PR** : proche, **LO** : lointaine, **TLO** : Très Lointaine, **NG** : Négatif Grande, **NM** : Négatif Moyenne, **NP** : Négatif Petite, **ZZ** : zéro, **PP** : Positif Petite, **PM** : Positif Moyenne, **PG** : Positive Grande, **F** : Faible, **RP** : Rapide, **TR** : Très Rapide.

La base de règles utilisée pour cette tâche autonome est résumée dans le tableau. III. 2.

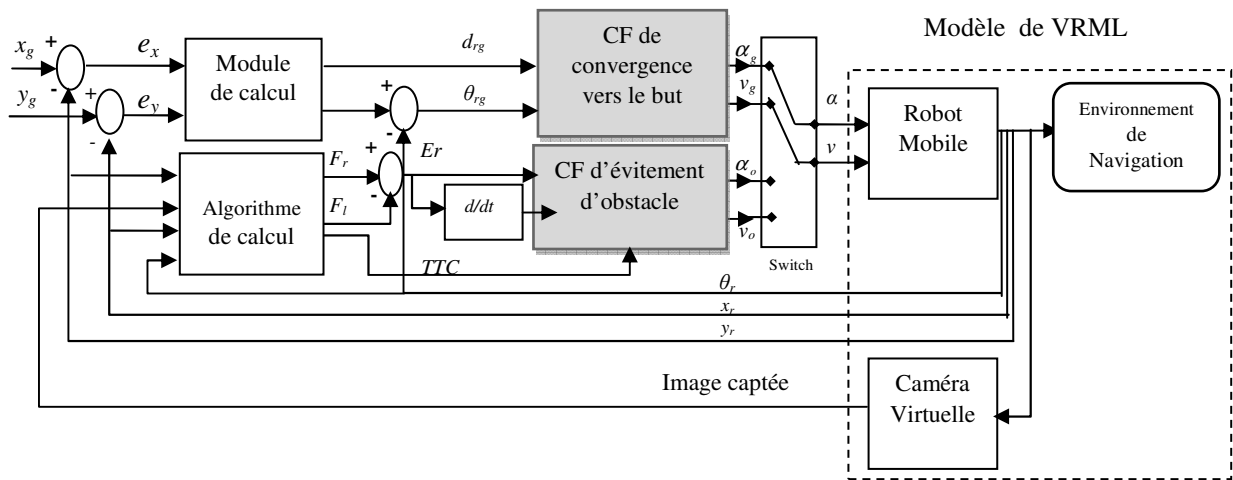


Figure III.14 Structure de contrôle globale

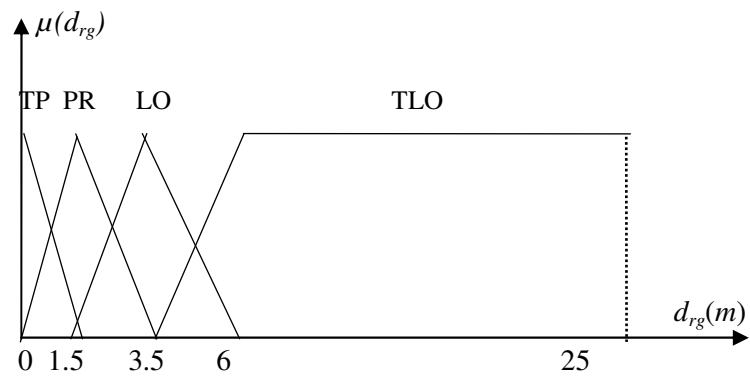


Figure III.15 Fonctions d'appartenance de la distance d_{rg}

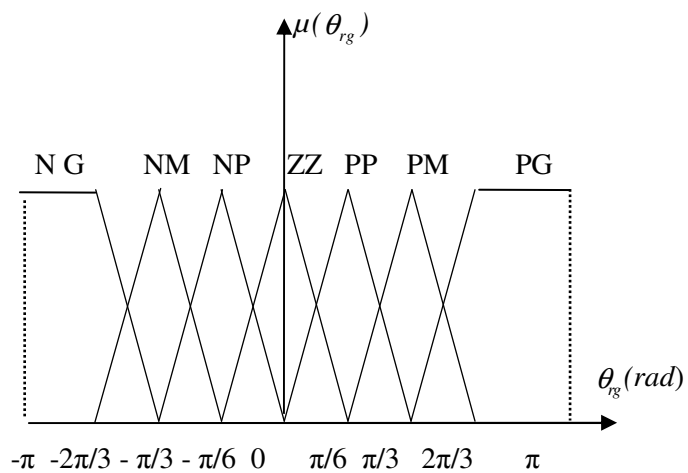


Figure III.16 Fonctions d'appartenance de l'angle θ_{rg}

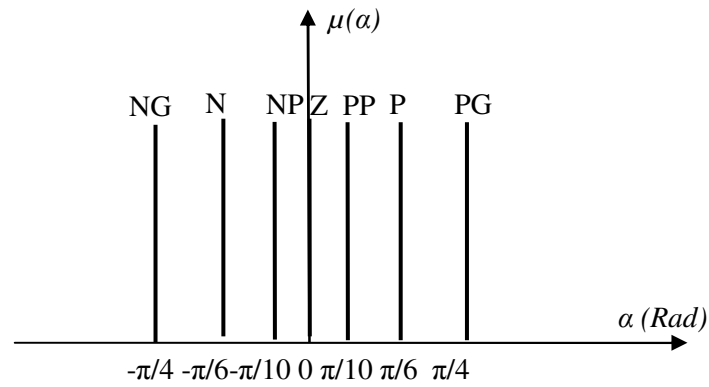


Figure III.17 Valeurs de la direction alpha

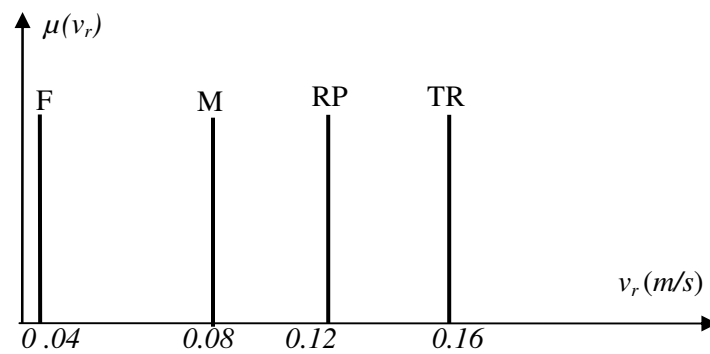


Figure III.18 Valeurs de la vitesse du robot.

Tableau III.2 Règles floues du contrôleur de convergence vers un but

Vitesse/ Braquage		d_{rg}			
		TP	PR	LO	TLO
θ_{rg}	NG	F	F	M	M
		PG	PG	PG	PG
	NM	F	F	M	FS
		PM	PM	PM	PM
	NP	F	M	RP	VF
		PP	PP	PP	PP
	ZZ	F	M	RP	VF
		ZZ	ZZ	ZZ	ZZ
	PP	F	M	RP	TR
		NF	NF	NF	NF
	PM	F	F	M	RP
		NM	NM	NM	NM
	PG	F	F	M	M
		NG	NG	NG	NG

III.6 Résultats et discussions

Dans ce qui suit, on va donner les résultats obtenus pour la navigation visuelle du robot mobile autonome. L'objectif est de vérifier l'efficacité des structures de commande proposées.

Ces exemples de simulation prennent en compte les types des environnements dans plusieurs situations telles que: l'espace libre, environnement avec des obstacles statiques, cible mobile, ...etc. Comme indiqué précédemment, la simulation a été faite en 2D et en 3D en utilisant l'environnement VRML de la réalité virtuelle. L'environnement simulé en 3D est illustré sur la figure III. 19 contenant des différents éléments de navigation : robot mobile, cible, obstacles. En plus, puisque l'environnement est du type intérieur, on a modélisé les quatre murs.

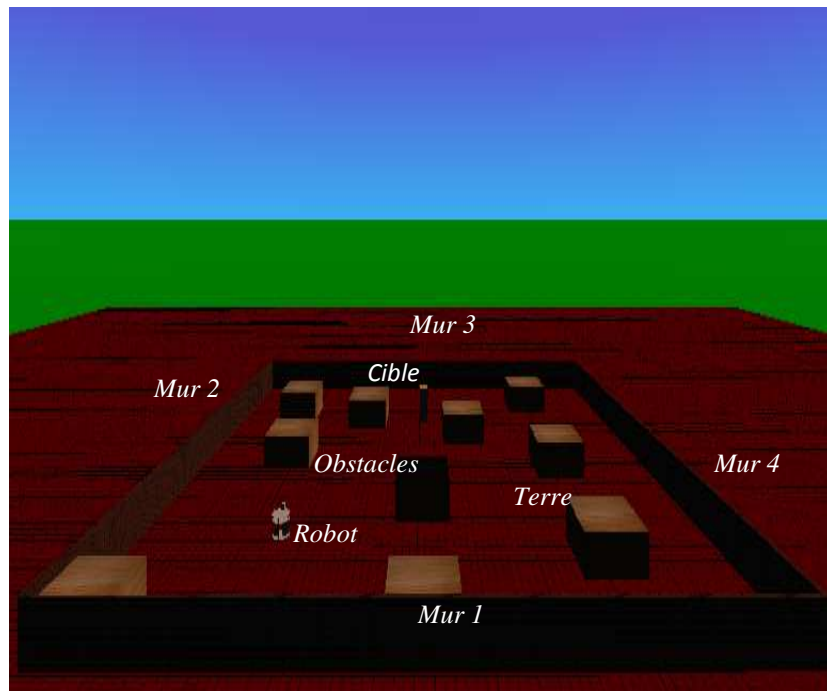


Figure.19 Environnement simulé en 3D

III.6.1 Tâche d'évitement des obstacles

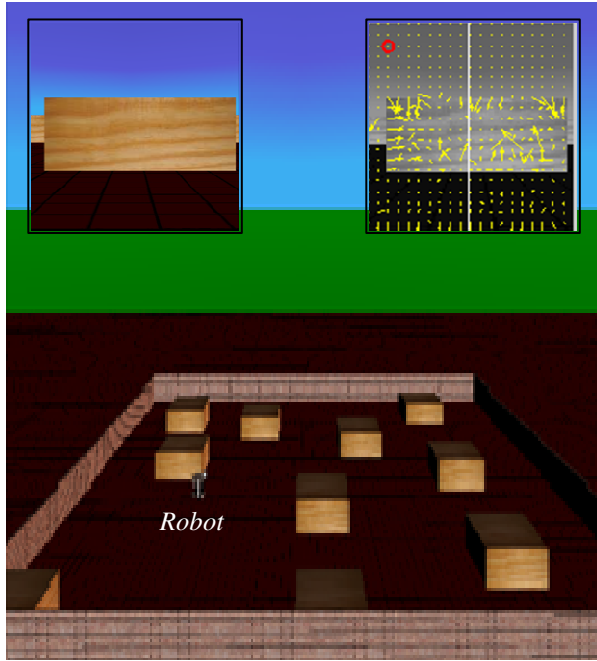
Dans le cas où l'environnement du robot contient des obstacles où des objets, le robot doit avoir une capacité d'évitement intelligente des collisions. En utilisant la structure de contrôle présentée sur la figure III. 7, les résultats de navigation autonome sont montrés sur les figures III. 20 (a...j).

Tout d'abord, dans cet essai, nous considérons le mouvement du robot sans prise en compte de la tâche de recherche du but (convergence vers un but). On a considéré les conditions suivantes :

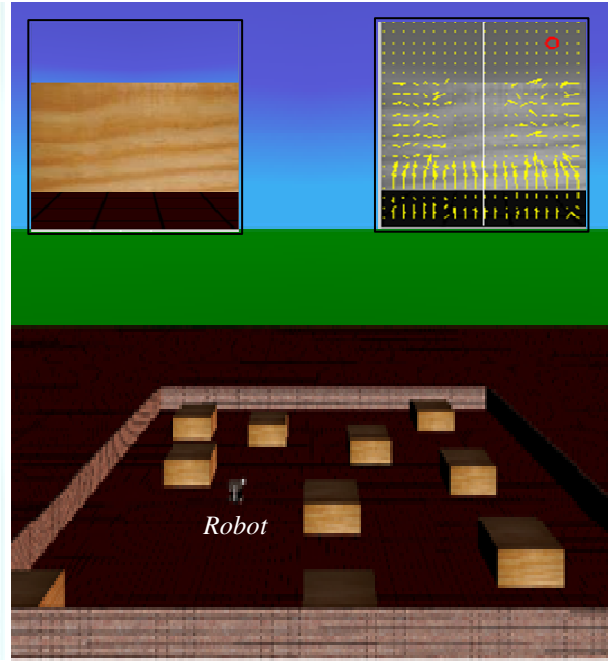
- Si l'environnement est libre d'obstacles, le robot mobile se mouvoir tout droit ($\alpha=0$) ;
- Sinon, le robot exécute l'action générée par le contrôleur flou type-1 afin d'éviter la collision avec les obstacles. Cela se fait selon les valeurs mesurées de E_r , ΔE_r .

Les résultats présentent les scènes de navigation du robot mobile à partir d'une position donnée (x_0, y_0) afin de se déplacer librement dans son environnement sans collision avec des obstacles. Pour chaque position (scène), l'image captée par la caméra et les vecteurs de flux optiques sont présentés en haut (gauche et droit, respectivement). L'image acquise indique l'état de l'environnement devant le robot mobile. Ensuite, des valeurs de flux optiques sont calculées en utilisant l'algorithme de Horn-Schunck permettant au robot de connaître la position des obstacles et leurs formes dans les deux parties droite et gauche. Où le plus proche obstacle détecté est celui que sera prise en considération par le contrôleur flou.

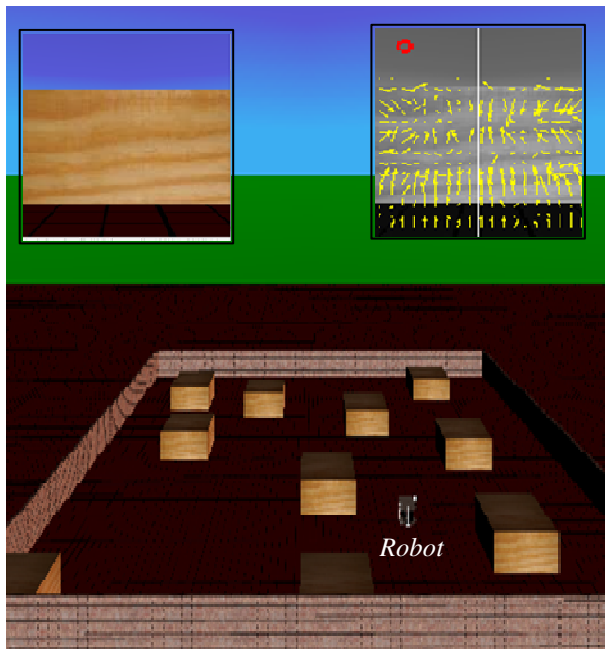
Les figures III. 20 (a...j) montrent les positions du robot mobile en présence des obstacles au cours du mouvement. Le robot se déplace tout droit, lorsqu'un obstacle est détecté devant lui par la caméra de perception, le contrôleur flou type-1 d'évitement des obstacles est activé afin de générer les actions appropriées de contournement. Il est bien indiqué que dans toutes les situations, le robot est capable de se mouvoir automatiquement sans collision.



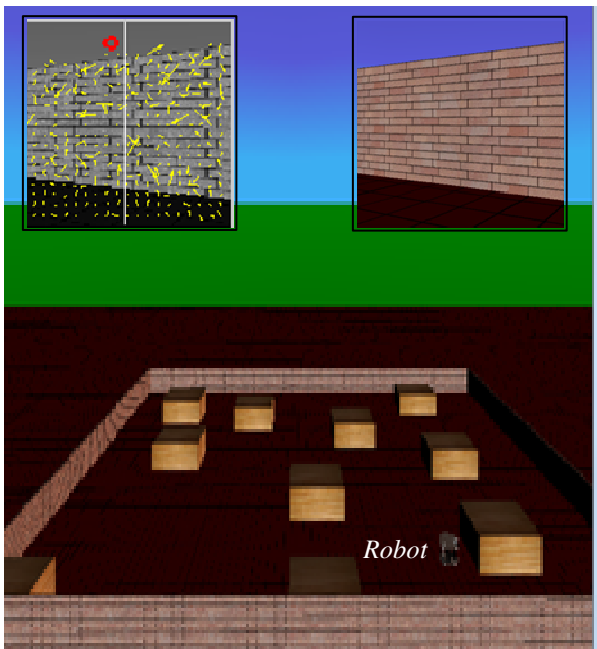
(a)



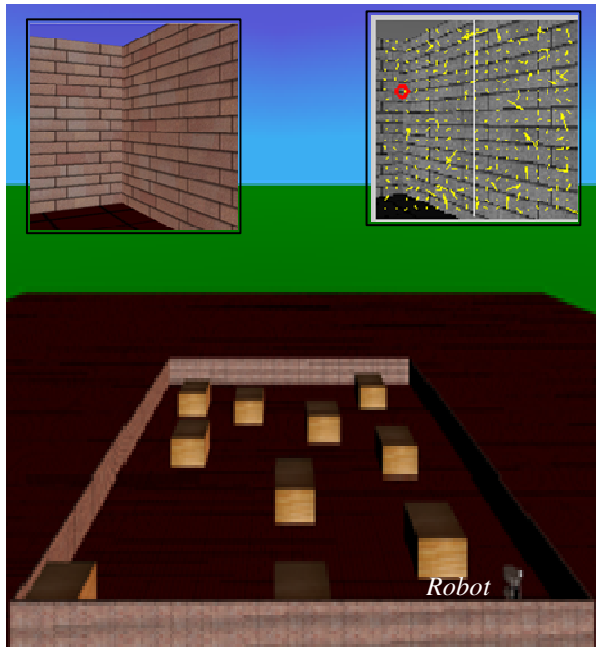
(b)



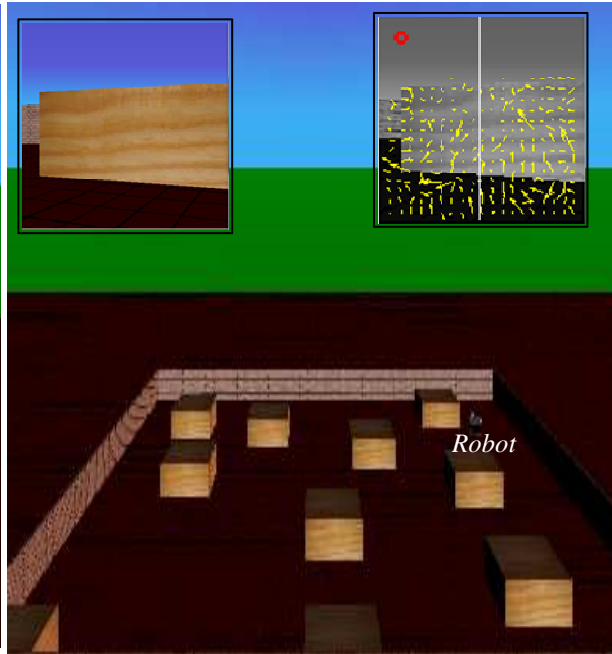
(c)



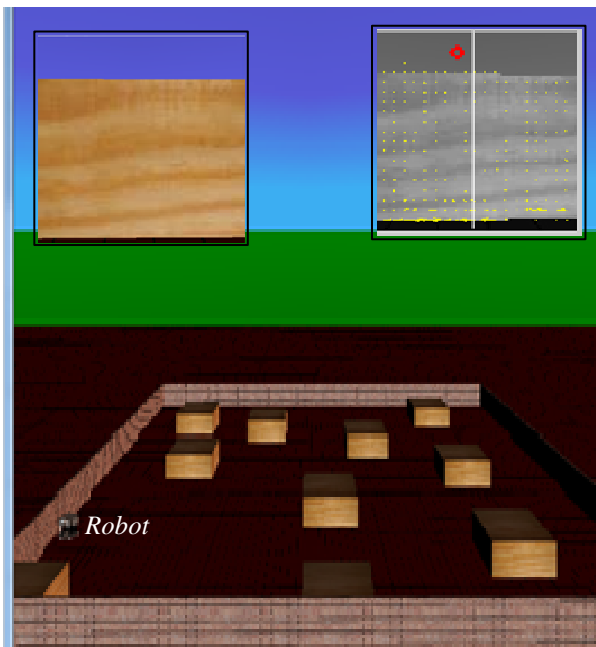
(d)



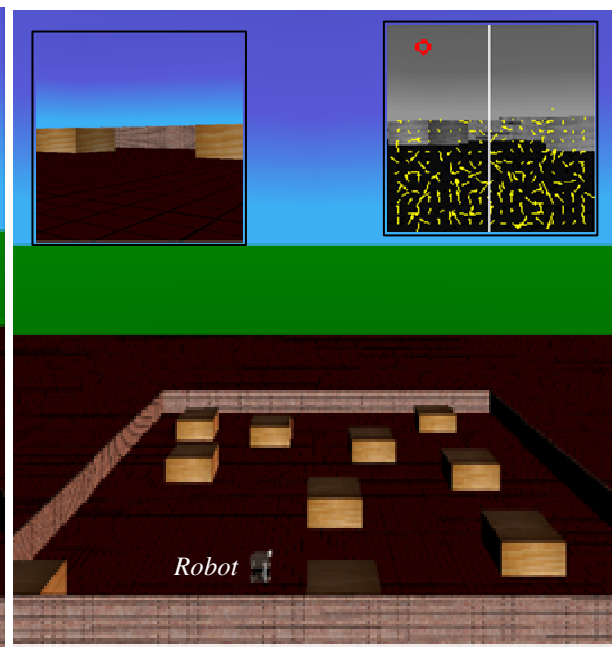
(e)



(f)



(g)



(h)

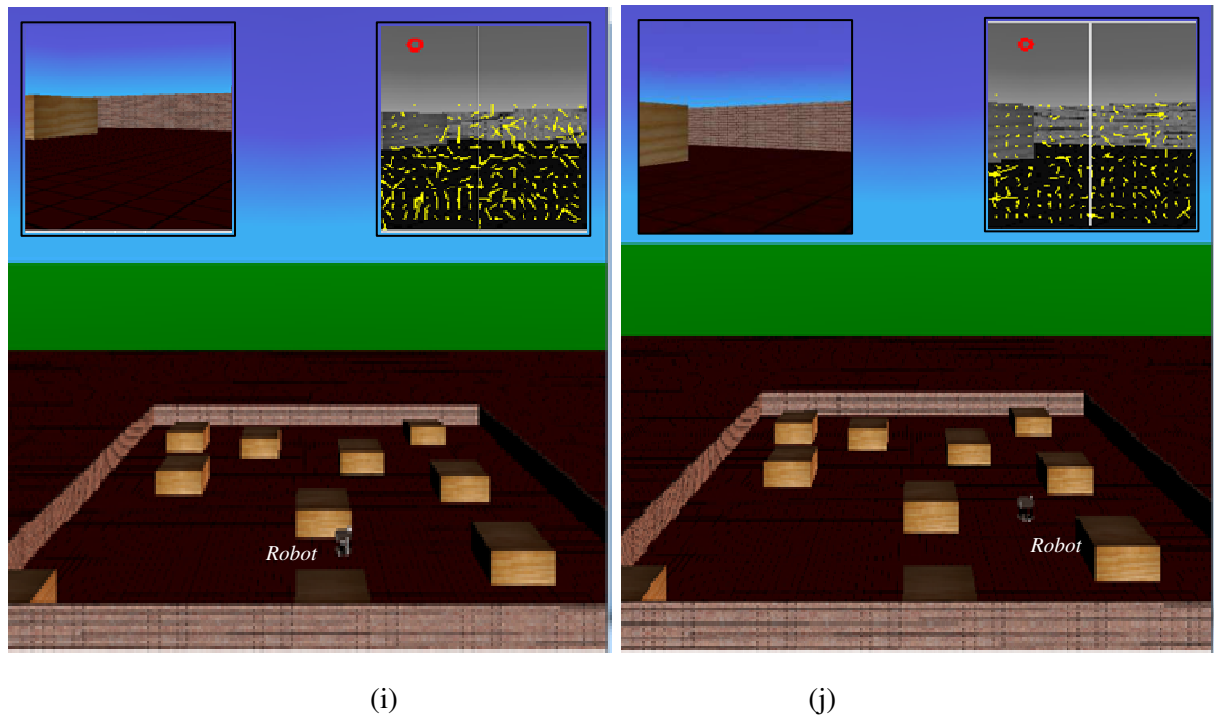


Figure III.20 Positions au cours de navigation visuelle du robot mobile

La trajectoire de mouvement obtenue en 2D est présentée sur la figure III. 21. Dans cette figure, on a montré les positions définies comme présenté en 3D de la figure III. 20 (a...j). Cette figure montre la capacité du robot à détecter et à éviter les obstacles

D'après ces scénarios, le contrôleur flou proposé est capable de générer les actions appropriées de contrôle pour accomplir cette tâche de manière efficace. Le robot se déplace librement et de manière autonome avec des bonnes performances.

Les figures III. 23 et III. 24 illustrent les valeurs mesurées des flux optiques dans les deux parties de l'image (droite et gauche) avec sa différence (erreur) pour cette trajectoire exécutée. Dans la figure III. 25, nous montrons l'estimation des valeurs de temps de contact (TTC). Cette variable est calculée à partir des valeurs de flux optiques des séquences d'images acquises pendant le mouvement du robot. Pour chaque courbe, nous avons indiqué les moments d'évitement des obstacles en exécutant les actions (tourner à gauche ou à droite).

Un deuxième exemple pour la navigation des robots mobiles est donné à la figure III. 22.

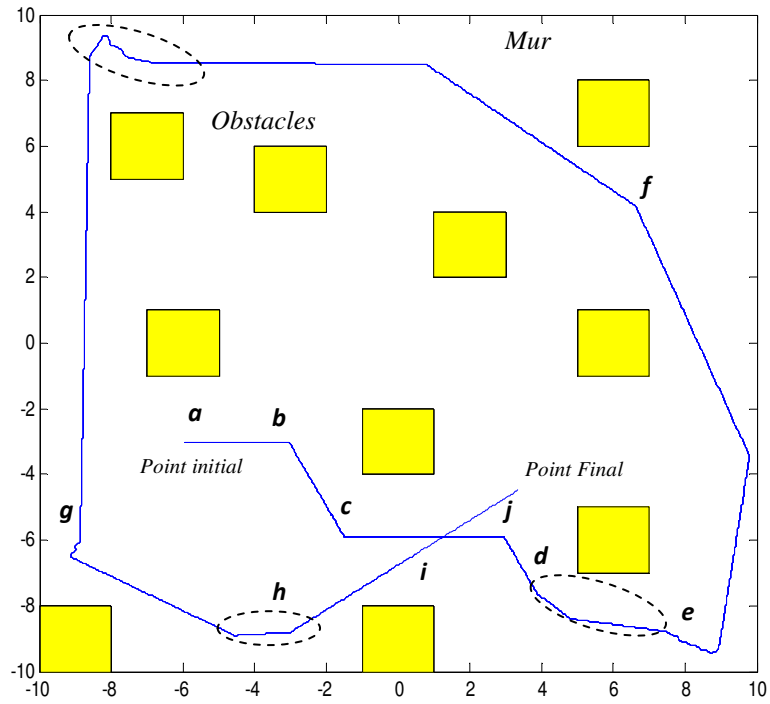


Figure III.21 Trajectoire parcourue en 2D

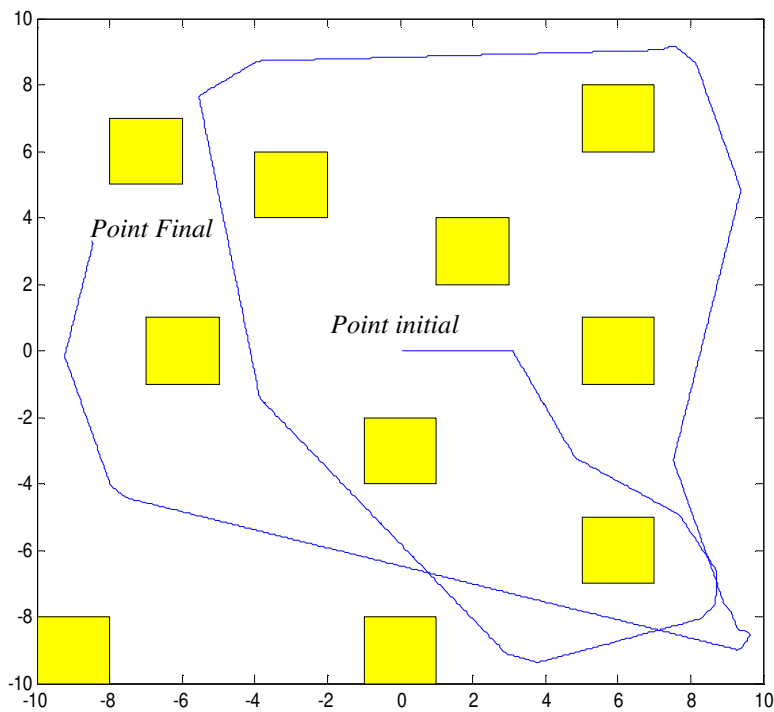


Figure III.22 Exemple 2 de navigation du robot

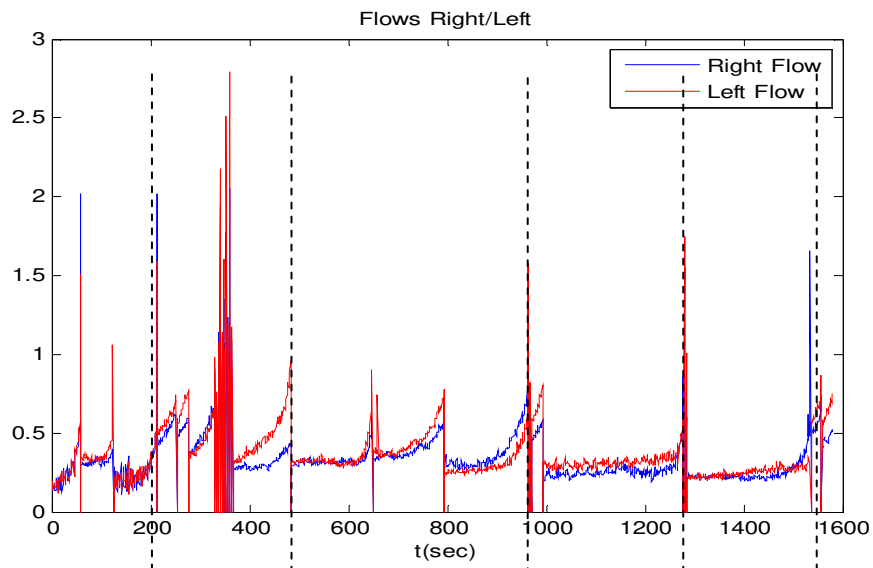


Figure III.23 Flux optiques F_{gauche} et F_{droite}

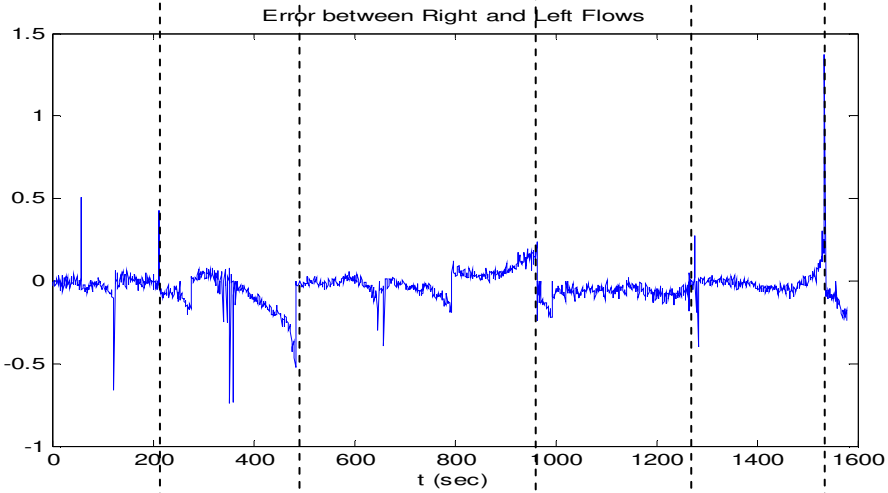


Figure III.24 Erreur de flux

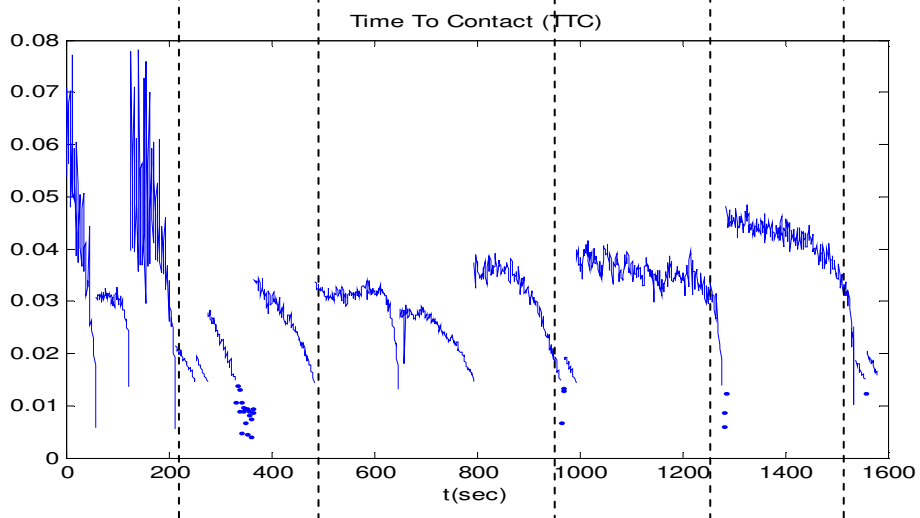


Figure III.25 Temps Avant Collision

III.6. 2 Tâche de Recherche du but (convergence vers un but)

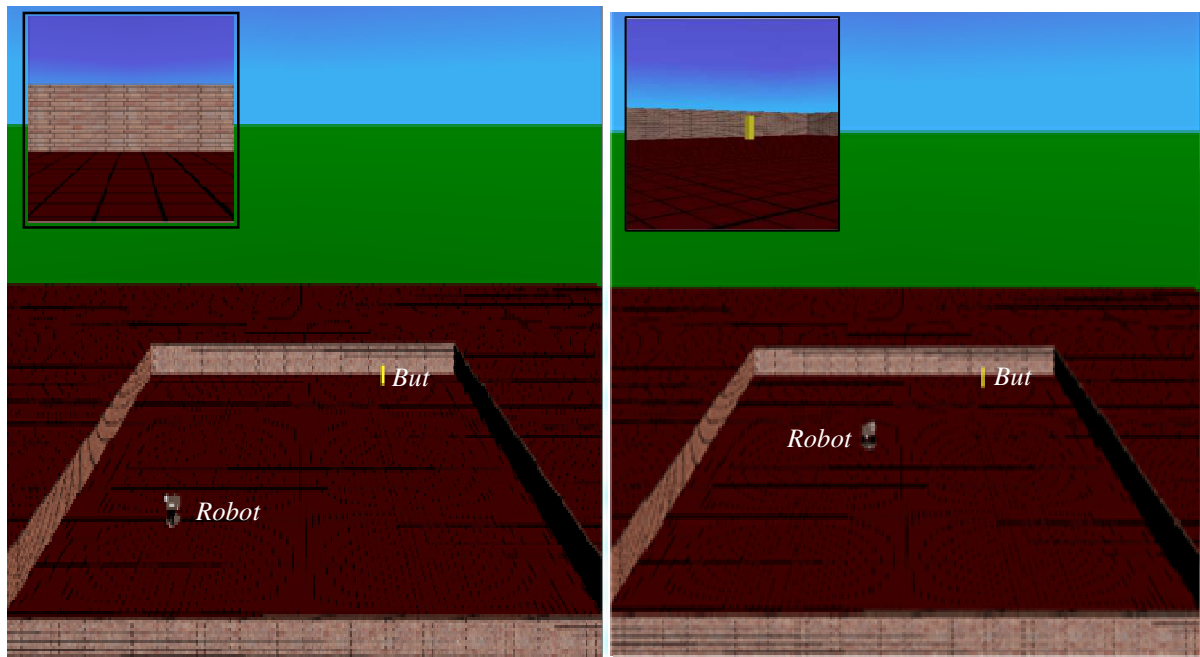
Dans le cas où aucun obstacle n'a été détecté, la tâche du robot mobile devienne alors une orientation directe vers le but pour l'atteindre (convergence vers un but).

Dans ce deuxième test, nous démontrons l'efficacité du contrôleur flou type-1 proposé pour accomplir cette tâche. Les résultats obtenus en 3D sont donnés sur les figures III. 26 (a-b-c) qui présentent les positions du robot pendant le mouvement (au début, en mouvement vers la cible et à la destination finale). Pour chaque scène, on montre l'environnement observé par la caméra. Dans cet exemple, nous avons considéré que le robot mobile se démarre à partir du point de coordonné $(x_r=5, y_r=5)$ avec une orientation $\theta_r=180^\circ$. Il doit se mouvoir vers le but défini par $(x_g=15, y_g=18)$.

Comme illustré dans ces figures, premièrement en point (a), le robot observe le mur devant lui, puis le contrôleur flou proposé génère les commandes nécessaires pour le braquage vers le but. Après un intervalle de temps comme indiqué sur la figure (b), le robot se trouve dans la même direction avec le but. Finalement, le robot peut atteindre sa destination finale de manière efficace (figure. c).

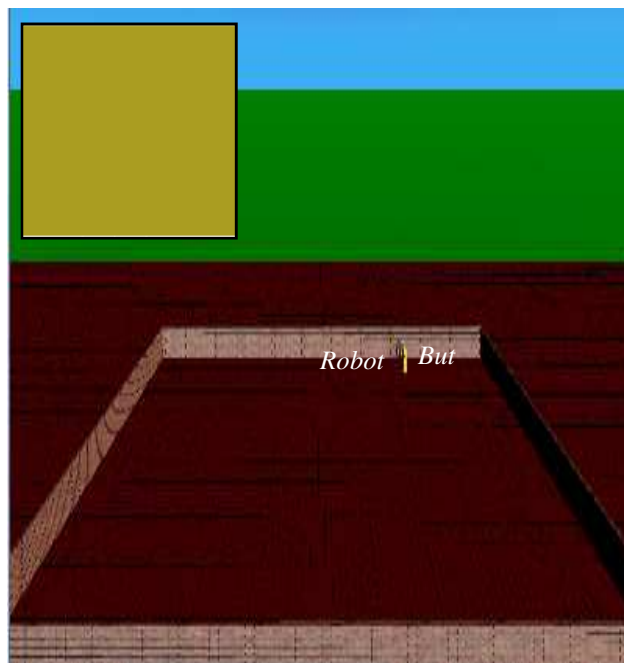
Le contrôleur flou type-1 élaboré est capable de diriger le robot vers le but final. La trajectoire obtenue en 2D est présentée sur la figure III. 27. Les actions générées par ce contrôleur : angle de braquage et vitesse sont présentés sur les figures III. 28 (a-b). Comme le montre sur la figure III. 28(a), puisque le robot se dirige vers le but, après une orientation vers la direction de la cible, l'angle de braquage est fixé à une valeur nulle. La vitesse de translation se diminue lorsqu'il s'approche au but. La figure III. 28(c) montre la variation de la distance entre le robot et le but qui converge vers zéro.

Le contrôleur flou proposé pour la convergence vers un but a démontré son efficacité pour accomplir cette tâche. Il donne au robot la capacité de naviguer de façon autonome.



(a)

(b)



(c)

Figure III.26 Scène du robot au point initial, au milieu et au point final

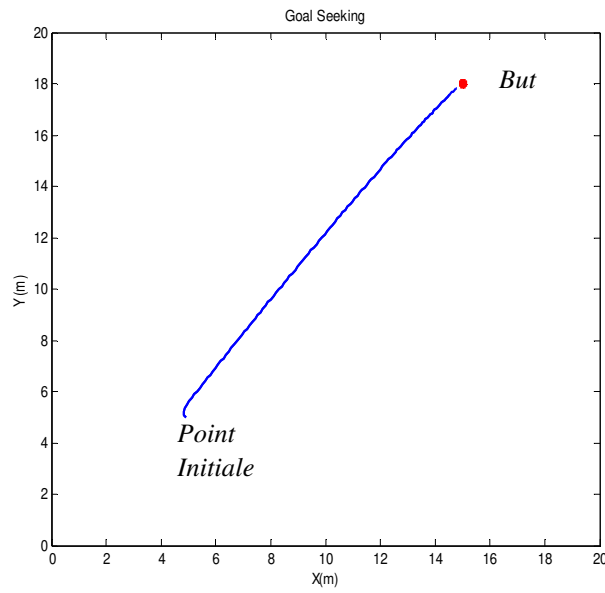
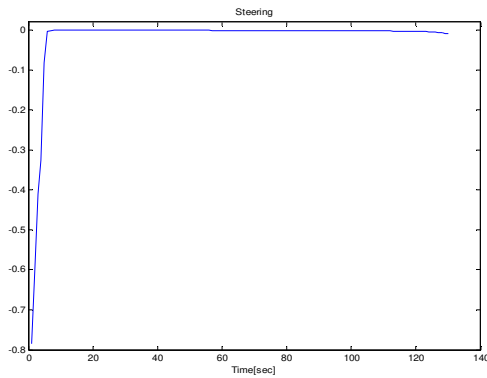
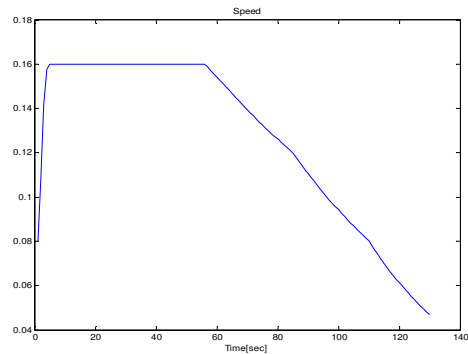


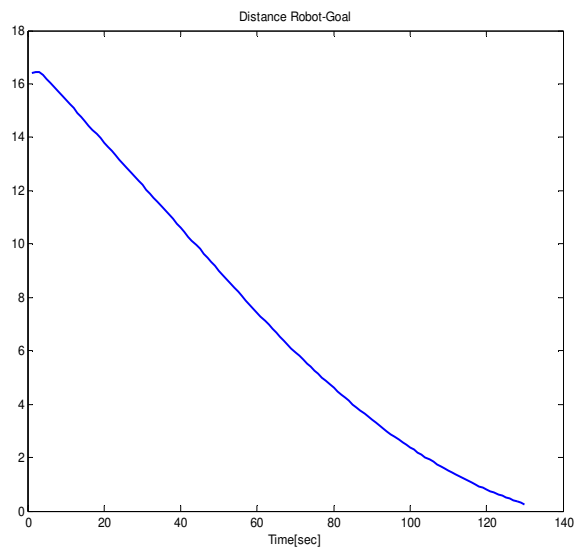
Figure III.27 Trajectoire du robot pour la convergence vers un but



(a)



(b)



(c)

Figure III.28 (a) Angle de braquage, (b) vitesse de translation et (c) la distance

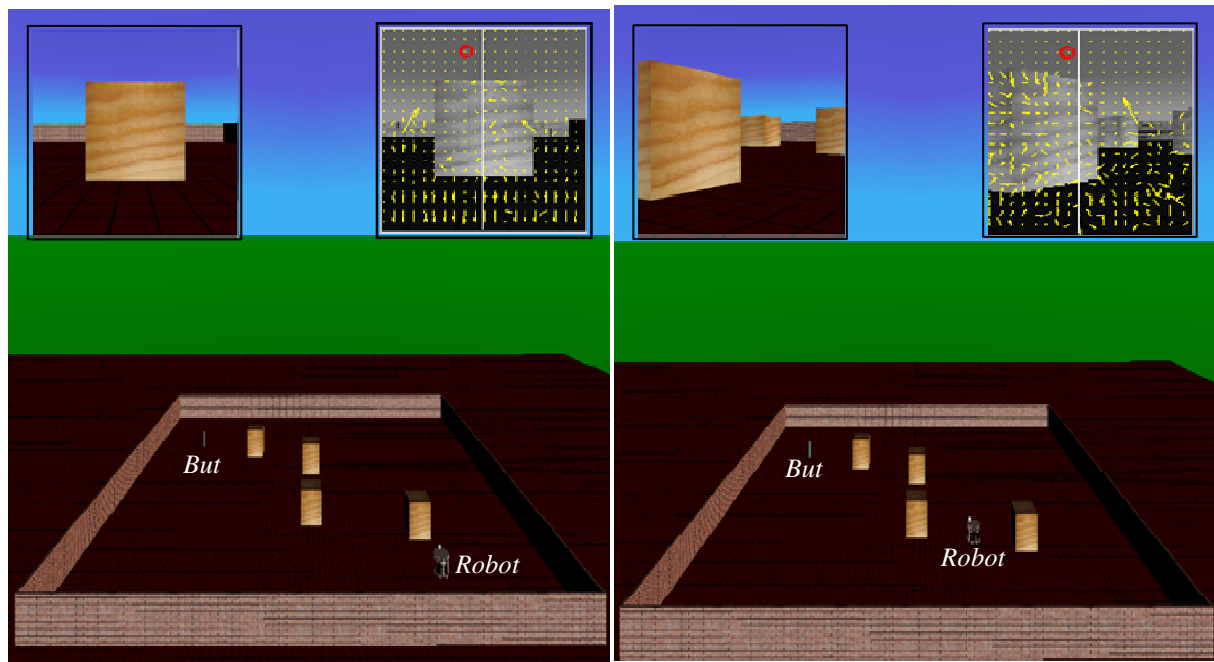
III.6. 3 Tâche de convergence vers un but avec évitement d'obstacles

Dans le cas de navigation en présence des obstacles, le robot mobile utilise la structure présentée précédemment sur la figure III.14. Le contrôle se fait par l'activation des deux contrôleurs flous type-1 de convergence vers le but et de l'évitement des obstacles. Le robot doit exécuter l'action adéquate en déclenchant l'un des deux contrôleurs selon la situation perçue.

Dans la figure III. 29 (a-b-c-d), on donne les résultats de simulation en 3D pour quatre positions du robot mobile au cours du mouvement. En utilisant le premier contrôleur flou de convergence vers un but, le robot se déplace librement dans la direction du but (la figure III. 29 (a)), mais si un obstacle est détecté par la caméra (les valeurs de flux calculées pour chaque partie de l'image), le deuxième contrôleur flou d'évitement des obstacles est activé pour générer des actions de contournement des obstacles. Pour chaque position dans les figures III. 29 (a-b-c-d), on montre l'image acquise par la caméra et les vecteurs de flux optique calculés pour les deux parties droite et gauche.

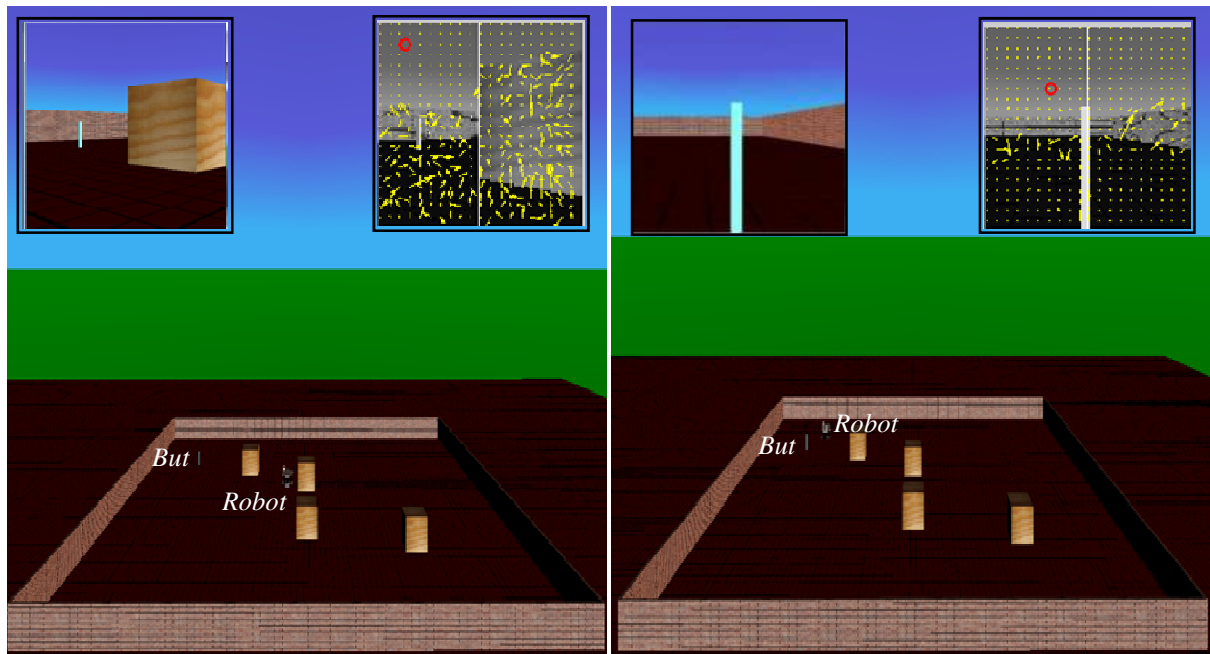
La trajectoire du robot obtenue en 2D est illustrée sur la figure III.30. Les situations des scènes (a-b-c-d) sont indiquées dans cette figure par les points (a-b-c-d).

D'après ces résultats, on constate que le robot mobile est capable de naviguer de manière autonome et avec succès pour atteindre son but en évitant les collisions avec les obstacles. Les résultats en 2D et en 3D montrent l'efficacité de la stratégie de contrôle hybride flou et optique proposée.



(a)

(b)



(c)

(d)

Figure III.29 Images de la navigation du robot (a-b-c-d)

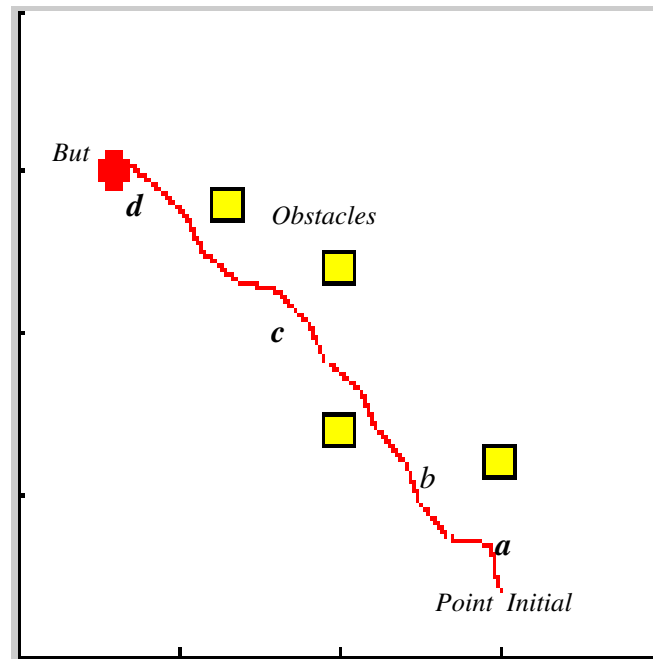
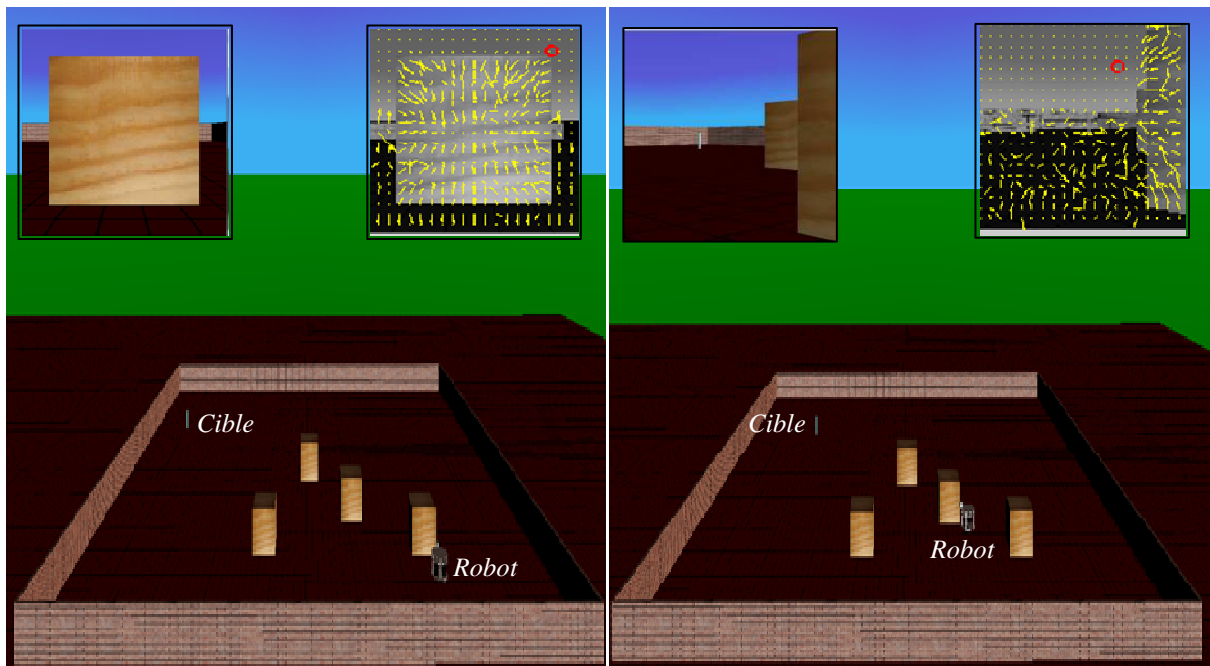


Figure III.30 Trajectoire du robot pour la navigation vers un but avec évitement des obstacles

III.6. 4 Tâche de Poursuite d'une cible mobile avec évitement d'obstacles

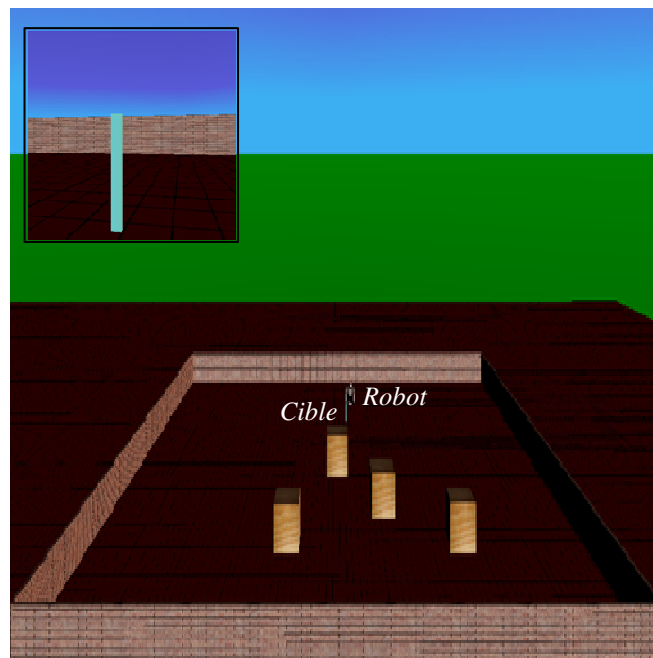
Dans cette section, la stratégie de navigation visuelle proposée a été testée pour suivre une cible en mouvement dans son environnement. L'objectif est de vérifier ses performances.

Les résultats de la simulation de cette tâche sont présentés en 3D sur les figures III. 31(a-b-c). Les contrôleurs proposés peuvent diriger le robot vers la cible afin de la poursuivre efficacement comme le montre la figure III. 32. Cette figure montre les trajectoires du robot et de la cible mobile (indiquées par les positions : a-b-c). Les contrôleurs flous type-1 élaborés peuvent garantir un bon suivi avec une fonctionnalité d'évitement d'obstacles.



(a)

(b)



(c)

Figure III.31 Scènes du robot pour la poursuite en mouvement avec évitement des obstacles

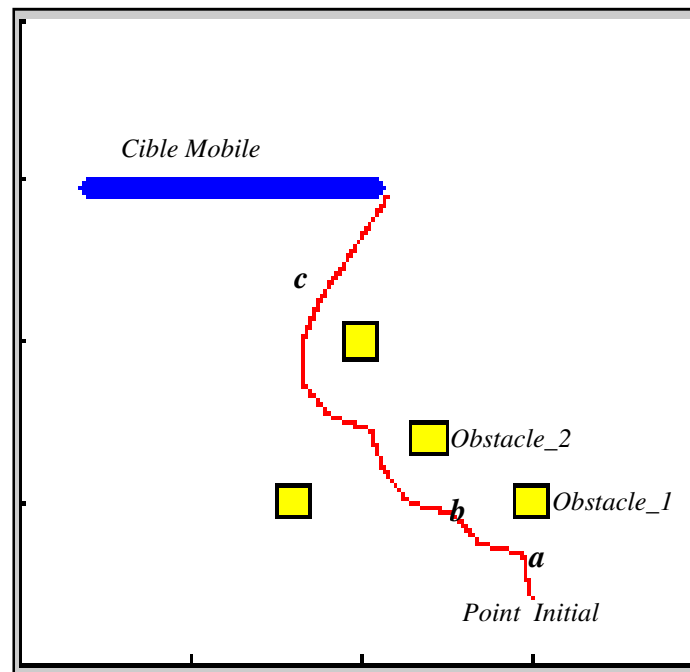


Figure III.32 Chemin du robot en 2D

III. 7 Conclusion

Dans ce chapitre, nous avons présenté un système de navigation visuelle pour un robot mobile à roues. La stratégie de contrôle proposée est basée sur l'application de l'algorithme Horn-Schunck du flux optique pour détecter les obstacles et les objets dans l'environnement de la navigation. L'architecture basée sur les comportements flous est utilisée pour régler le problème du conflit des actions dans les approches traditionnelles.

Deux contrôleurs à base de logique floue type-1 sont utilisés pour générer les actions du mouvement du robot mobile, le premier oriente le robot vers la direction de but en générant la vitesse et l'angle de braquage. Le second contrôleur flou utilisé pour diriger le robot autour des obstacles les plus proches par le calcul des valeurs de flux optique des deux parties de l'image acquise. L'image captée est traitée, redimensionnée et subdivisée en deux parties afin de générer les actions de contrôle appropriées.

L'approche proposée a été simulée en environnement 3D à l'aide de la bibliothèque VRML. L'efficacité est démontrée à l'aide de nombreux tests de simulation pour les différentes sous-tâches (évitement d'obstacles, convergence vers un but, poursuite d'une cible mobile avec évitement d'obstacles).

Les résultats obtenus confirment l'utilité et l'efficacité des contrôleurs conçus. Ils présentent un bon niveau de performances pour la navigation autonome du robot mobile.

Dans le prochain chapitre, nous allons utiliser des contrôleurs flous type-2 avec la méthode de Horn-Schunck du flux optique pour le mouvement automatique d'un robot mobile.

Chapitre IV

Navigation Visuelle par des Contrôleurs Flous type-2 et le Flux Optique

IV.1 Introduction

Les contrôleurs flous ont la capacité de gérer et de traiter les incertitudes présentes aux systèmes dynamiques [13, 14, 18]. La logique floue type-1 utilise des ensembles flous nets et précis. Cependant, La logique floue type-2 présente une autre génération des contrôleurs basés sur le principe d'extension de Zadeh [72]. Ces contrôleurs offrent l'opportunité de modéliser les incertitudes et de donner une meilleure représentation des connaissances pour la navigation des robots mobiles [16, 17, 93].

Ces dernières années, les systèmes à logique floue de type-2 ont introduit des outils importants pour améliorer les performances du robot [97]. Les ensembles à logique floue de type-2 et leurs concepts théoriques créent une nouvelle version des contrôleurs à logique floue dans le domaine de la robotique [98]. Les systèmes à logique floue de type-2 offrent un bon outil pour modéliser certains niveaux d'incertitudes que les systèmes à logique floue de type-1 ne peuvent pas atteindre. La dimension supplémentaire de fonction d'appartenance de type-2 peut donner une meilleure représentation de l'incertitude que de fonction d'appartenance de type-1 [75, 97, 98].

Comme présenté dans le chapitre précédent, on peut citer ici quelques travaux liés à notre application, concernant l'utilisation des systèmes flous type-2:

- Dans le travail de [93, 94], les auteurs ont proposé une comparaison entre des contrôleurs de Takagi-Sugeno type-1 et type-2 pour contrôler un robot mobile. Ils ont étudié et comparé les performances des trois principaux comportements : la recherche d'objectifs, l'évitement des obstacles et le suivi de murs.
- Les auteurs de [96], ont étudié la tâche de navigation dans des environnements dynamiques avec une structure hiérarchique en utilisant un contrôleur à base de la logique floue type-2.

Dans ce chapitre, l'association du contrôleur logique floue Takagi-Sugeno du type 2 avec l'algorithme de Horn-Schunck de la méthode de flux optique est utilisée pour la navigation autonome d'un robot mobile avec évitement des obstacles. Les résultats seront comparés et discutés.

IV.2 Aperçu sur les contrôleurs flous type-2

Le principal avantage de la commande par la logique floue type-2 est l'introduction de l'expérience humaine sous la forme de règles " Si -Alors " sur la façon de contrôler des

systèmes dont les incertitudes augmentent. La structure d'un contrôleur à logique floue de type 2 est la même que celle d'un contrôleur de type-1, la différence entre eux se situe dans l'étape de traitement de sortie [15, 17, 76]. Les ensembles flous de type-2 sont des fonctions tridimensionnelles.

Dans ce chapitre, nous avons utilisé un contrôleur flou de Takagi-Sugeno type-2 afin d'inférer les valeurs de commande nécessaires pour la navigation visuelle et autonome d'un robot mobile.

La structure du système flou type-2 est celle présentée dans chapitre II. Il contient les quatre blocs discutés précédemment :

- *Fuzzification* : pour calculer l'appartenance de chaque valeur réelle ;
- *Une base de règles* : de la forme : **Si** condition (s) **Alors** Conclusion (s), qui définit le comportement du robot mobile (dans notre application c'est le comportement flou d'évitement des obstacles) ;
- *Un mécanisme d'inférence* : pour traiter les valeurs d'entrées et de lire la base de règles floues ;
- *Un module de calcul de la sortie* : qui contient le réducteur de type et une étape de défuzzification.

Puisque dans ce travail, on a utilisé un contrôleur flou de Takagi-Sugeno type-2. Ce type du système flou supprime la notion de défuzzification [69, 72, 76].

La sortie est calculée selon les étapes suivantes [93, 76, 101]. :

Pour un système flou de Takagi-Sugeno avec M règles floues, vecteur d'entrée de p valeurs ($x_1 \in X_1, \dots, x_p \in X_p$), et une seule sortie ($y \in Y$).

- La règle floue r^{th} peut être exprimée par:

$$\text{Si } x_1 \text{ est } \tilde{F}_1^r \text{ et } x_2 \text{ est } \tilde{F}_2^r \text{ et...et } x_p \text{ est } \tilde{F}_i^r, \text{ ALORS } y^r = c_0^r + c_1^r x_1 + \dots + c_p^r x_p$$

Avec $c_0^r, c_1^r, \dots, c_p^r$ sont des constantes.

- Le degré de vérité de la règle est exprimé par les équations suivantes:

$$W^i(x') = [w^i(x'), \bar{w}^i(x')]$$

$$\begin{aligned} \underline{w}^i &= \underline{\mu}_{\tilde{F}_i}(x_1) * \dots * \underline{\mu}_{\tilde{F}_p}(x_p) \\ \overline{w}^i &= \overline{\mu}_{\tilde{F}_i}(x_1) * \dots * \overline{\mu}_{\tilde{F}_p}(x_p) \end{aligned} \quad (\text{IV.1})$$

- La sortie est calculée sous la forme:

$$Y(Y^1, \dots, Y^M, W^1, \dots, W^M) = [y_l, y_r] \quad (\text{IV.2})$$

$$[y_l, y_r] = \int_{y^1} \dots \int_{y^M} \int_{w^M} \dots 1 / \frac{\sum_{i=1}^M w^i y^i}{\sum_{i=1}^M w^i} \quad (\text{IV.3})$$

- Où $y_i \in Y^i$, et $Y^i = [y_l^i, y_r^i]$, ($i=1 \dots M$), y_l et y_r sont calculées en utilisant les équations 4 et 5.

$$y_l = \frac{\sum_{i=1}^M w_l^i y_l^i}{\sum_{i=1}^M w_l^i}, \quad y_r = \frac{\sum_{i=1}^M w_r^i y_r^i}{\sum_{i=1}^M w_r^i} \quad (\text{IV.4})$$

- La sortie globale du système est :

$$y = (y_l + y_r) / 2 \quad (\text{IV.5})$$

IV.3 Contrôleur Flou type-2 pour l'évitement des obstacles

Dans cette section, on va présenter la structure de navigateur proposé pour l'évitement d'obstacles [91]. La structure de contrôle est présentée sur la figure IV.1 (qui est celle utilisée dans le chapitre III, en remplaçant le contrôleur flou type-1 par un contrôleur flou type-2). Le robot contrôlé perçoit son environnement à l'aide d'une caméra virtuelle à haute résolution. L'algorithme Horn-Schunck (HS) du flux optique est utilisé pour estimer les vecteurs de vitesse de l'image (calcul du flux optique). L'image est divisée en deux parties : droite et gauche.

Le contrôleur proposé est de type Takagi-Sugeno qui utilise les deux entrées : E_r et ΔE_r et génère une seule sortie (angle de braquage) en considérant dans le cas de contournement des obstacles que la vitesse est fixe et faible. Les entrées sont fuzzifiées à l'aide des fonctions d'appartenance type-2 illustrées dans les figures IV.2 et IV.3.

L'action de sortie (angle de braquage) est représentée par des singletons comme le montre la figure IV.4. Les mots linguistiques utilisés sont : **ZZ** : Zéro, **PP** : Positive Petite, **PG** : Positive Grande, **NG** : Négative Grande, **NP** : Négative Petite.

Les valeurs utilisées pour les paramètres des ensembles flous sont :

- $a = [-0.3573 \ -0.073]$;
- $b = [-0.203 \ -0.153]$;
- $c = [-0.156 \ -0.106]$;
- $d = [-0.0368 \ 0.132]$;
- $e = [0 \ 0.05]$; $f = [0.079 \ 0.129]$;
- $g = [0.121 \ 0.171]$;
- $h = [0.2966 \ 0.3466]$.

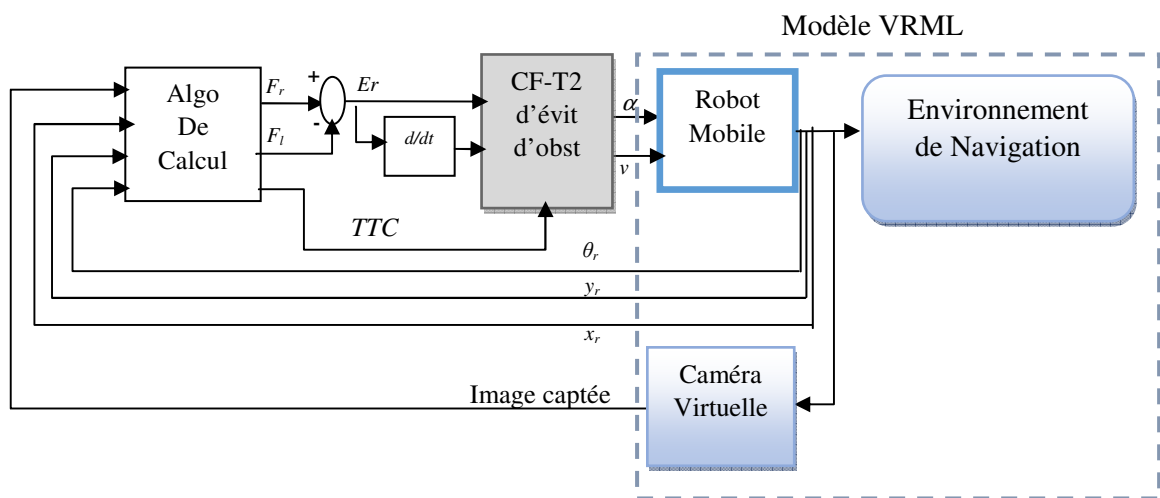


Figure IV.1 Navigateur type-2 visuel pour l'évitement des obstacles

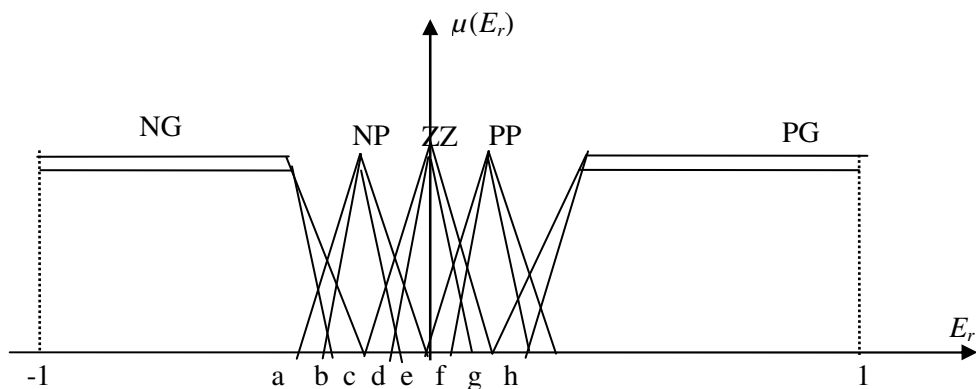


Figure IV.2 Fonctions d'appartenances de l'erreur E_r

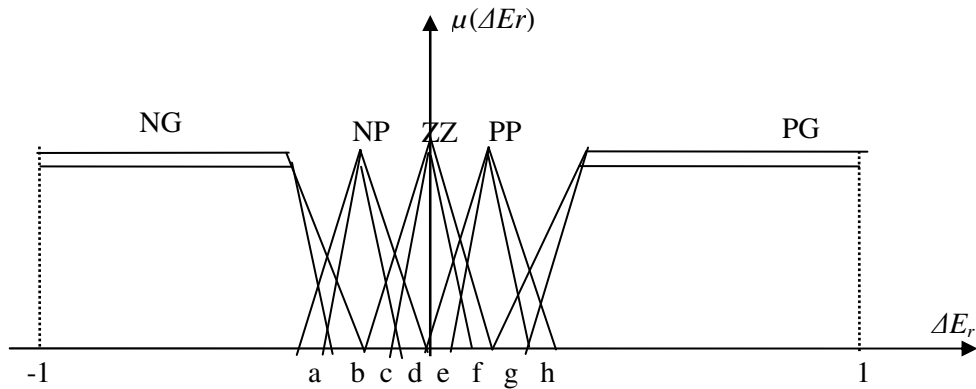


Figure IV.3 Fonction d'appartenance de la variation de l'erreur ΔE_r

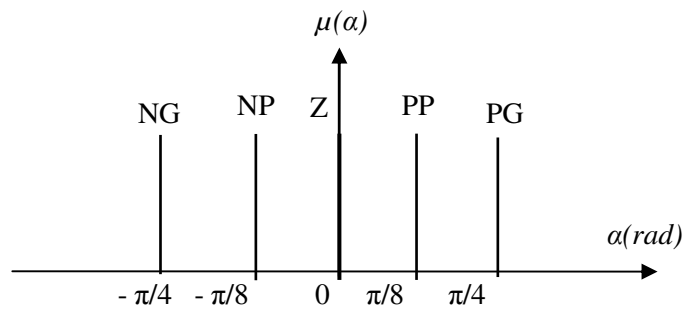


Figure IV.4 Fonctions d'appartenance de α

En utilisant ces variables, les règles floues de type-2 « Si-Alors » sont données dans le tableau IV.1. La base de règles est utilisée pour caractériser la relation entre les entrées floues de type-2 et la sortie floue de type-2 comme suit :

$$\text{Si } Er \text{ est } A_1^i \text{ et } \Delta Er \text{ est } A_2^i \text{ ALORS } \alpha = B^i \text{ avec } v = c^{ste} \quad (\text{IV.8})$$

Où $i=1\dots N$, et N est le nombre de règles floues dans une base de règles floue donnée, $A_1^i \dots A_2^i$ sont les ensembles flous de type 2, B^i est l'ensemble flou de sortie.

Tableau IV.1 Règles floues type-2 pour l'évitement des obstacles

ΔEr	Er				
	NG	NP	ZZ	PP	PG
NG	PG	PG	PP	NP	NG
NP	PP	PP	PP	NP	NG
ZZ	PP	PP	NP	NP	NP
PP	PP	PP	NP	NP	NP
PB	PG	PP	NP	NG	NG

IV.6 Résultats de simulation

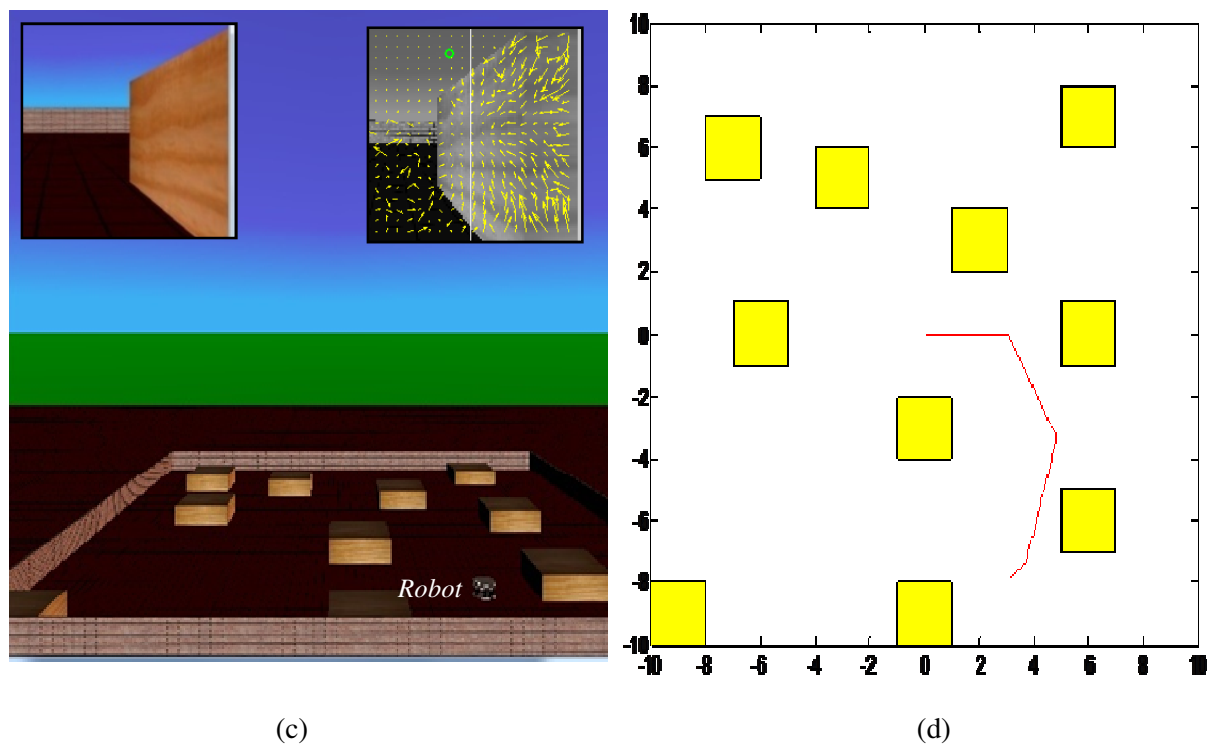
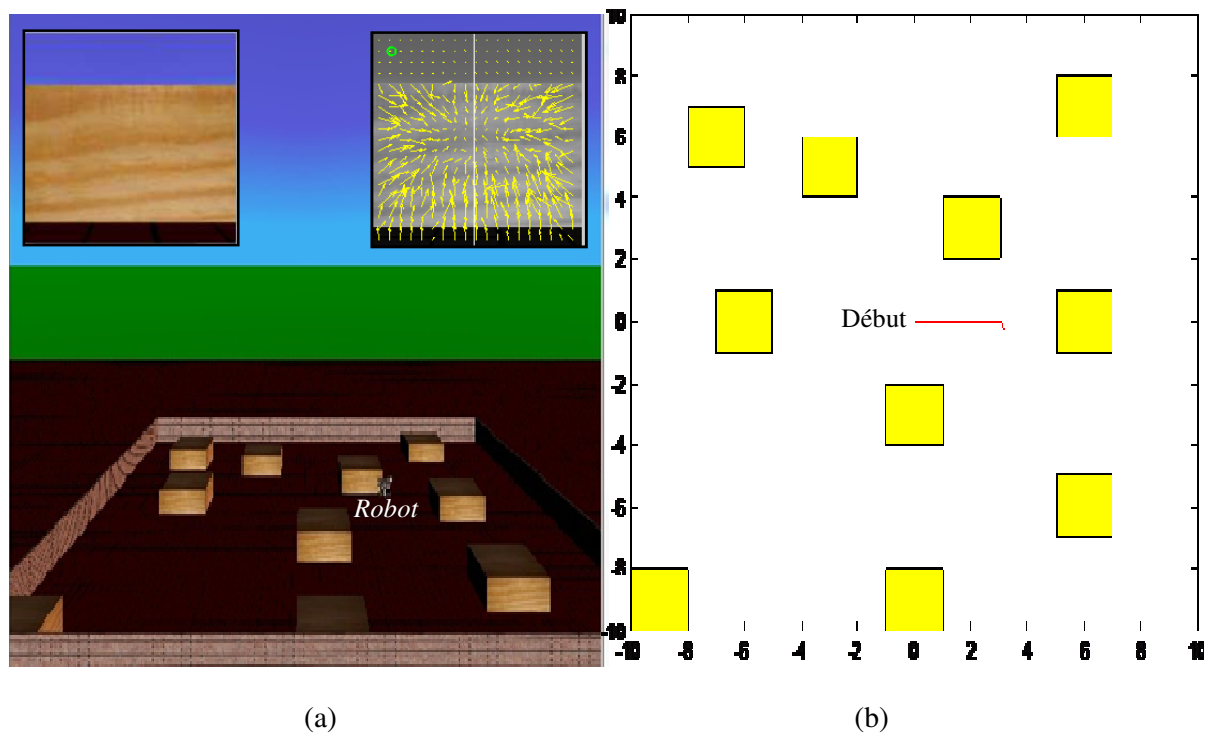
Les résultats de simulation de la navigation visuelle d'un robot mobile seront présentés dans cette partie en utilisant la structure basée sur le contrôleur flou et l'algorithme H.S proposé.

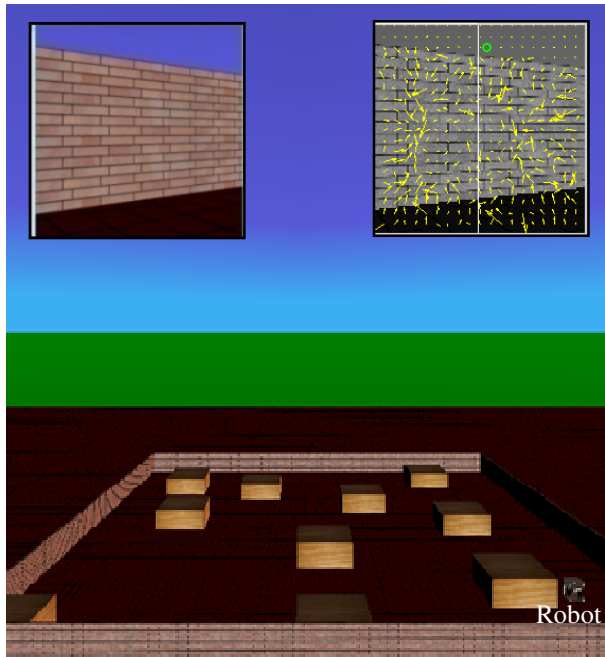
La simulation a été faite à l'aide de toolbox VRML (Virtual Reality Modeling Language) pour définir l'environnement de navigation sous forme 2D et 3D comme que nous avons utilisé dans le chapitre précédent.

IV.6.1 Tâche d'évitement des obstacles

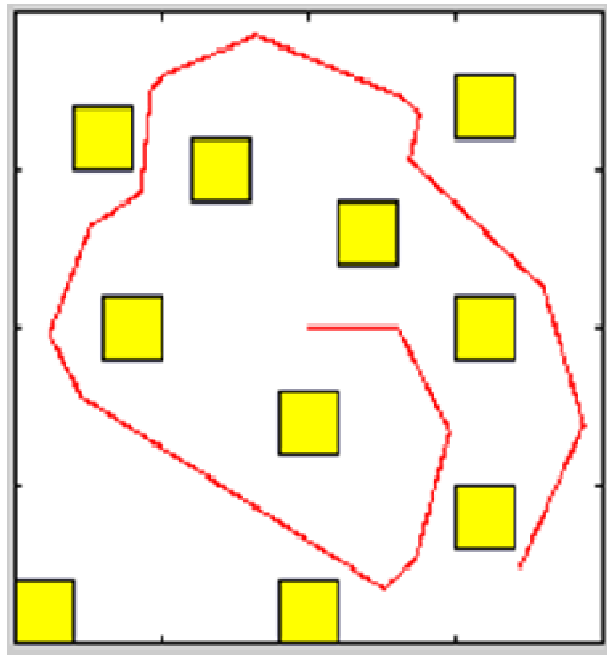
Pour tester la tâche d'évitement d'obstacles visuels, nous supposons que dans le cas où l'environnement du robot est libre (aucun obstacle proche), le robot mobile avance tout droit, si non, il doit exécuter la valeur d'action générée par le contrôleur flou type-2 pour éviter la collision.

Les résultats de simulation pour la tâche de mouvement automatique avec l'évitement d'obstacles sont présentés sur les figures IV.5 (a...n). Ils présentent les positions du robot pendant le déplacement, en évitant les obstacles à partir du point initial (début) jusqu'à la destination finale (Fin). Pour chaque position, l'image captée en couleur et les vecteurs de flux optiques calculés sont donnés dans les parties supérieures gauches et droites, respectivement. Pour chaque position, on montre la position et la trajectoire parcourue par le robot mobile en 2D.

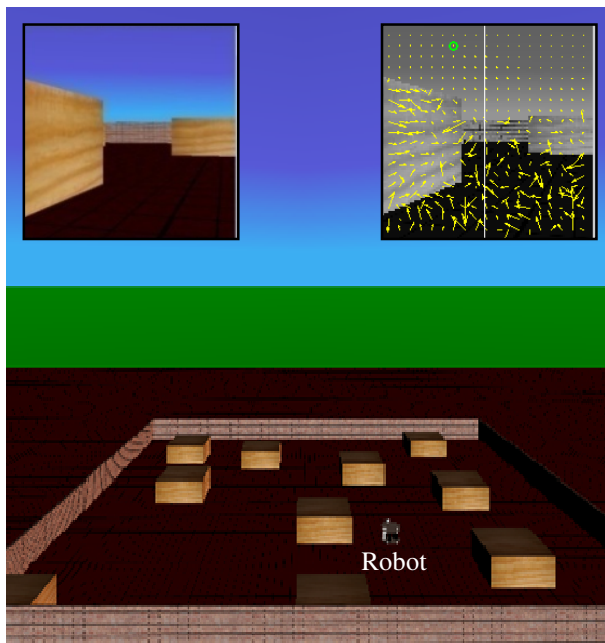




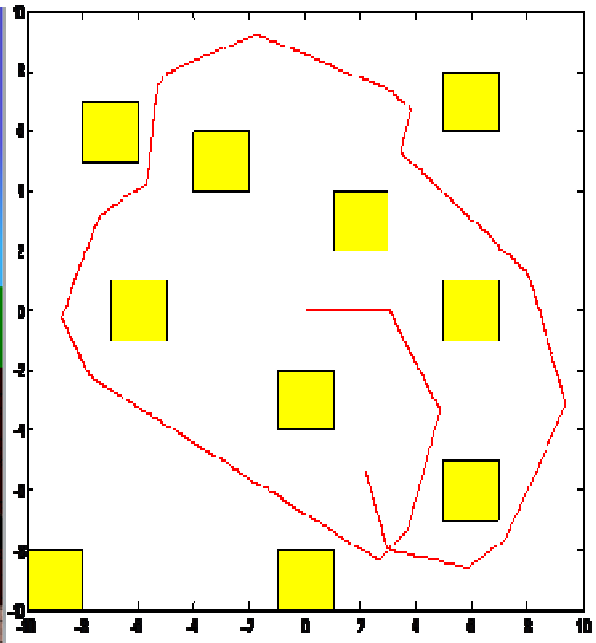
(i)



(j)



(k)



(l)

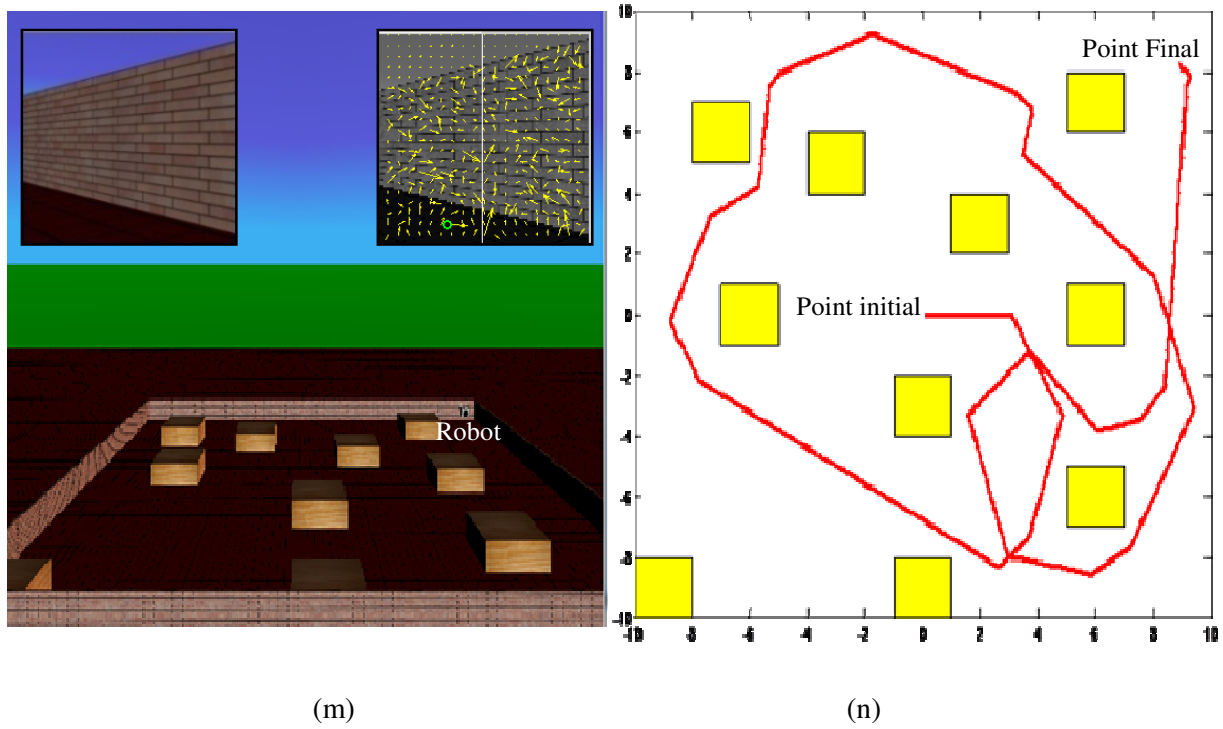


Figure IV. 5 Positions du robot en 3D et en 2D, (n) : Trajectoire finale obtenue

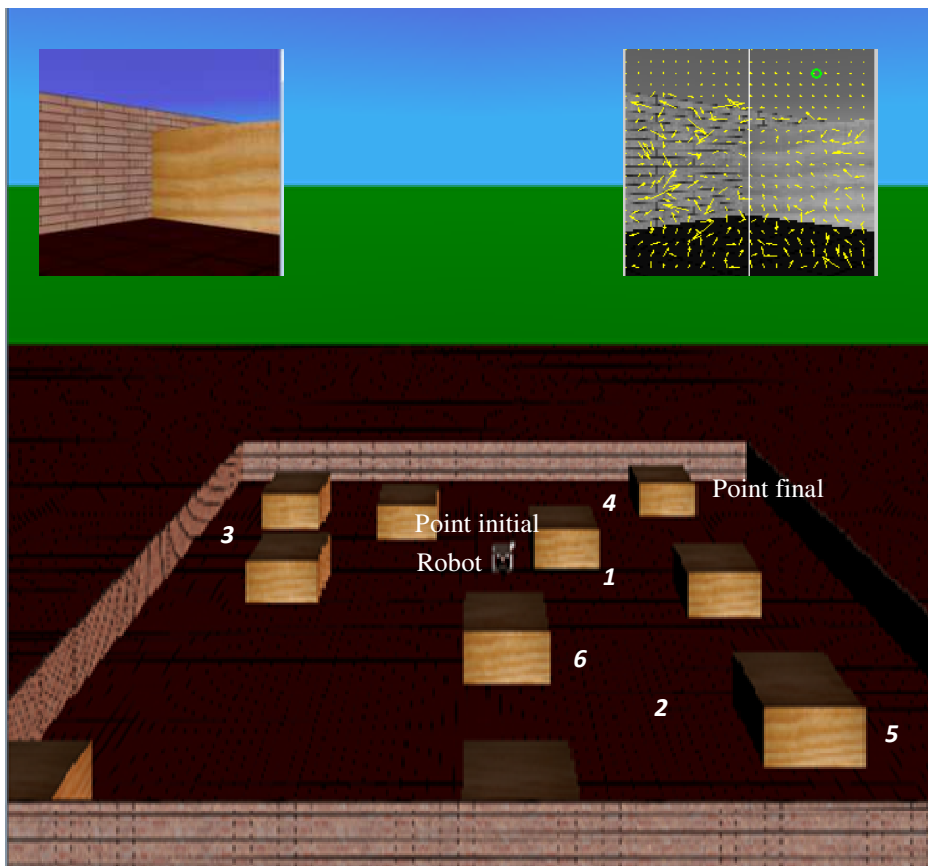


Figure IV.6 Différentes positions du Robot en mouvement.

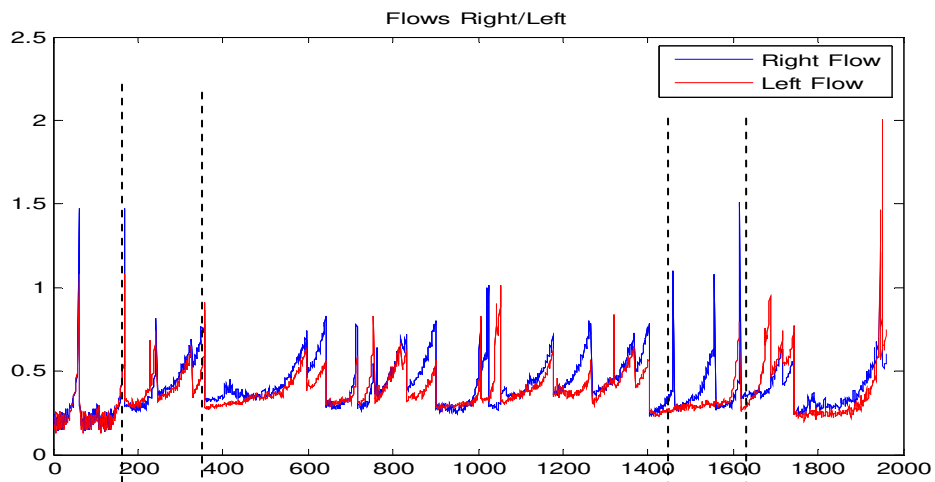


Figure IV.7 Flux optiques F_{gauche} et F_{droite}

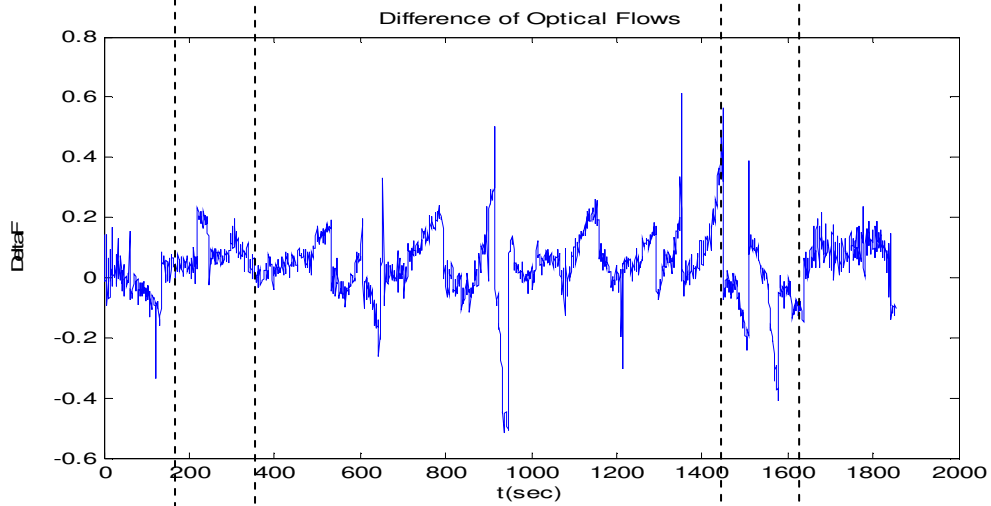


Figure IV.8 Erreur de flux

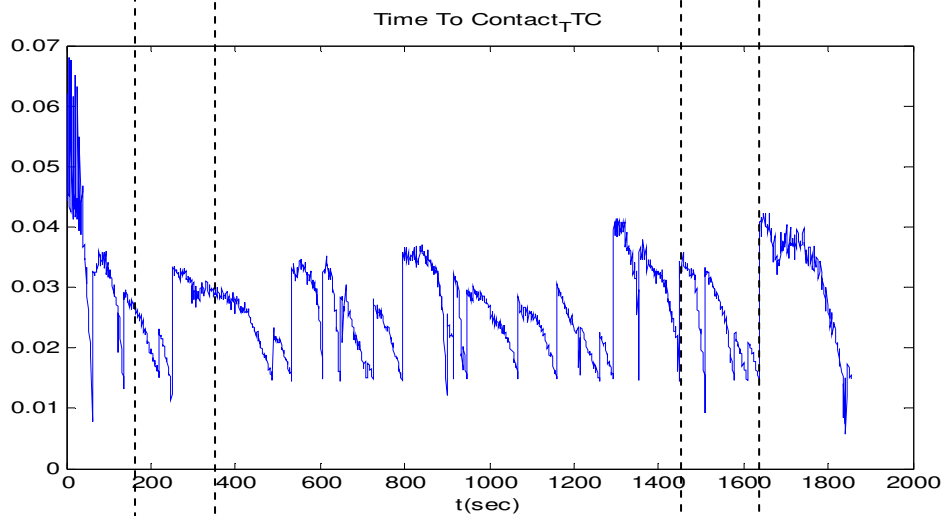


Figure IV.9 Temps Avant Collision

À chaque pas de temps, le robot perçoit l'environnement environnant à l'aide de la caméra virtuelle utilisée. L'image acquise indique l'état de l'environnement sur la face avant du robot. Ensuite, des valeurs de flux optiques sont calculées pour détecter les obstacles dans les deux parties : droite et gauche. L'obstacle détecté et l'obstacle le plus proche sont définis par les vecteurs de vitesse optique (en couleur jaune, en haut à droite).

La trajectoire finale parcourue par le robot mobile est illustrée sur la Figure IV. 5.

La figure IV.6 résume les positions du robot (1, 2, ..., 6) pendant le mouvement avec évitement des obstacles, en partant du point initial (début) jusqu'à la destination finale (fin).

Des exemples de la vue de la caméra en couleur et des vecteurs de flux optique calculés sont donnés dans les coins supérieurs gauche et droit, respectivement.

Les résultats présentés dans ces figures montrent que la structure de commande proposée et le contrôleur flou type-2 élaboré est capable de générer l'action de contrôle appropriée pour la tâche d'évitement des obstacles. Le robot se déplace librement et de manière autonome, sans collision. Ce système de navigation est efficace pour accomplir cette tâche avec de bonnes performances.

Les figures IV.7, IV.8 et IV.9 donnent la variation des valeurs de flux optique pour les deux parties de l'image droite-gauche, l'erreur et le temps de contact (TTC) respectivement. Pour chaque figure, le moment où l'obstacle a été évité est indiqué en ligne pointillés (l'action peut être : *tourner à gauche* ou *tourner à droite*).

IV.7 Comparaison entre FLC de Type-1 et FLC de type-2

Dans cette section, nous présentons une comparaison entre les deux structures de commande proposées pour l'évitement des obstacles : la structure présentée sur la figure III. 7 du chapitre III et celle de la figure IV. 1.

Pour les mêmes conditions, et les valeurs initiales concernant la position de démarrage du robot et les positions des obstacles, les deux figures IV.10 et IV.11 représentent les trajectoires du robot en 2D en utilisant les deux contrôleurs flous type-1 (CF-1) et type-2 (CF-2) respectivement. La figure IV.12 montre les deux trajectoires en même temps. On a montré que les systèmes de commande proposés donnent des meilleures performances pour cette fonctionnalité. Le robot peut éviter la collision avec les objets proches en toute efficacité.

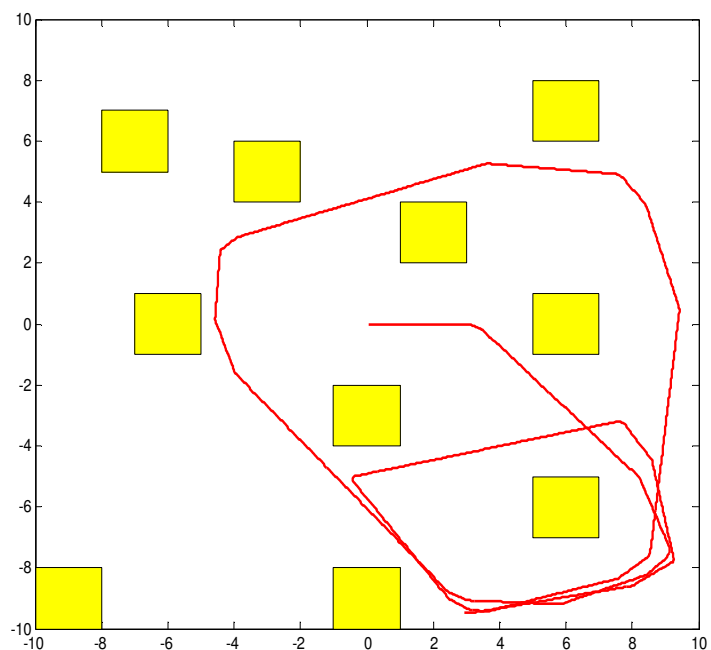


Figure IV.10 Trajectoire du Robot en utilisant le CF-1

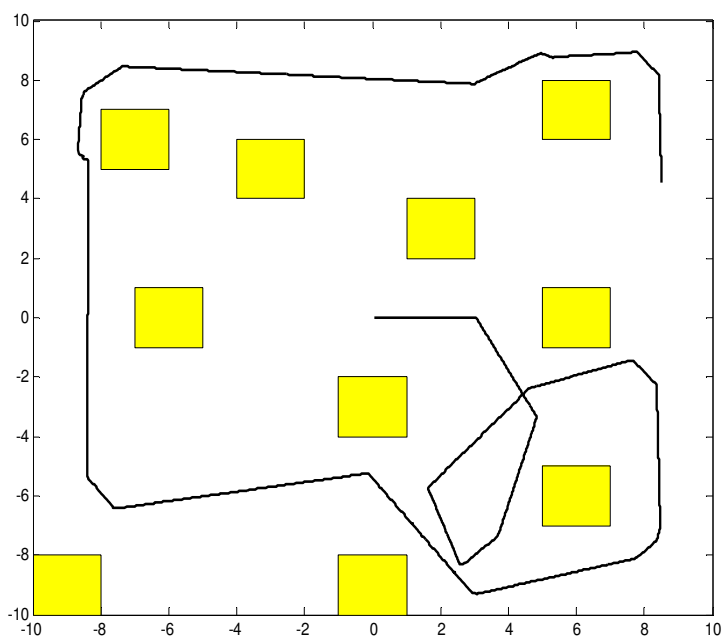


Figure IV.11 Trajectoire du Robot en utilisant le CF-2

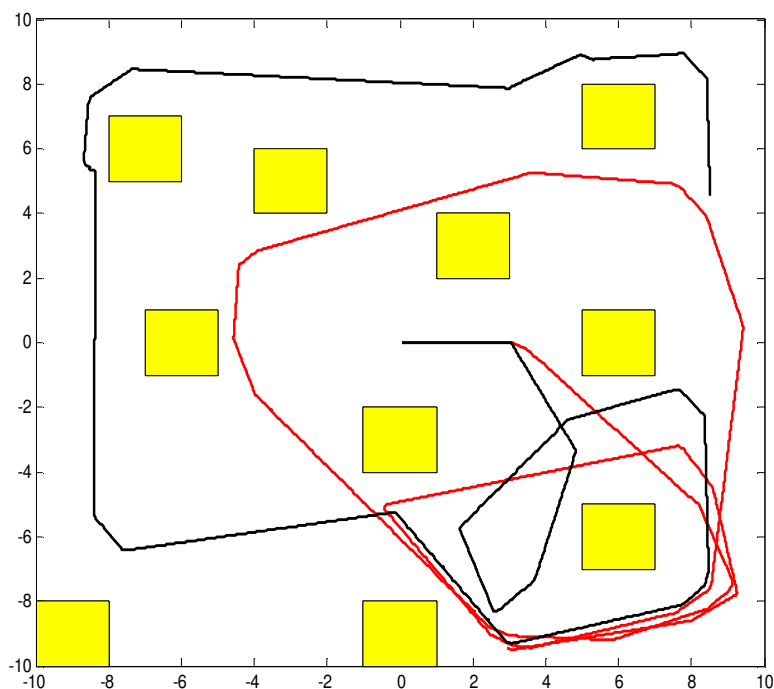


Figure IV.12 Représentation des deux trajectoires du robot en utilisant CF-1(rouge) et CF-2 (noire)

IV.8 Conclusion

Dans ce chapitre, un contrôleur visuel à base de la logique floue type-2 pour l'évitement d'obstacles est proposé pour les robots mobiles. Le navigateur proposé est un contrôleur flou de Takagi-Sugeno type-2 combiné avec l'algorithme Horn-Schunck de la méthode du flux optique. Les résultats de simulation obtenus démontrent l'efficacité de la stratégie de contrôle proposée pour la navigation des robots mobiles et l'évitement des obstacles.

Conclusion Générale

Dans ce travail, le problème de la navigation visuelle autonome d'un robot mobile à roues a été abordé en utilisant la méthode de Horn & Schunck du flux optique et la logique floue de type-1 et type-2. L'objectif principal de notre travail est de faire commander le robot mobile pour atteindre la configuration d'arrivée à partir d'une configuration de départ sans collision avec les obstacles et sans intervention humaine avec un minimum d'erreur.

Le travail présenté dans cette thèse est divisé en deux parties:

La première partie a présenté un aperçu sur le domaine de la navigation autonome de robot mobile en détaillant la navigation basée sur la vision. Dans cette partie nous avons donné les concepts de base des approches utilisées à savoir; la commande floue et la méthode de flux optique.

Par la suite, nous avons exposé la théorie de la commande par logique floue de type-1 et de type-2 et l'architecture de base d'un contrôleur flou pour les deux types. 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.

Dans la deuxième partie, nous avons présenté les deux applications:

Dans un premier lieu, nous nous sommes intéressés par l'utilisation de l'algorithme de Horn & Schunck pour détecter les obstacles et des contrôleurs flous pour la navigation autonome d'un robot mobile. Nous avons utilisé des structures basées comportements avec des modèles de type Takagi-Sugeno d'ordre zéro. La commande à base de la logique floue a démontré son efficacité pour les différentes missions du robot mobile (navigation visuelle autonome avec évitements d'obstacle, convergence vers un but, convergence vers un but en détectant les obstacles à éviter, et poursuite d'une cible mobile avec évitement d'obstacles).

Nous avons utilisé deux contrôleurs flous pour commander et contrôler le robot mobile afin de donner la capacité de se mouvoir dans l'environnement simulé en 3D à l'aide de la toolbox VRML. Le premier contrôleur oriente le robot vers la cible (le but) en générant la vitesse et l'angle de braquage. Le second est utilisé pour éviter les obstacles autour de robot mobile.

L'approche proposée a été simulée en environnement 3D par le VRML. L'efficacité des structures proposées est illustrée à l'aide de nombreux tests de simulation pour les différentes sous-tâches (évitement d'obstacles, convergence vers un but, poursuite d'une cible mobile avec évitement d'obstacles).

Les résultats obtenus montrent que les structures étudiées en utilisant les contrôleurs flous type-1 et type-2 donnent des meilleures performances pour la navigation visuelle autonome d'un robot mobile dans un environnement intérieur.

Enfin, nous proposons les perspectives suivantes :

- La mise en œuvre en temps réel de la stratégie de contrôle étudiée;
- l'utilisation d'un système visuel pour suivre et intercepter des objets en mouvement dans des environnements encombrés;
- L'application des architectures présentées pour des environnements avec des obstacles dynamiques.

Références Bibliographiques:

- [01] L. Freda, G. Oriolo, " Vision based Interception of Moving Target with Non holonomic Mobile Robot ", *Robotics and Automation Systems*, vol. 55, no. 5, pp. 419-432, 2007.
- [02] M. S. Guzel , Bicker R, " Vision Based Obstacle Avoidance Techniques ", *Book chapter in Recent Advances in Mobile Robotics (InTech)*, pp. 83-108, 2011.
- [03] C. W. Wang, Q. H, Meng, " Obstacle Avoidance for Quadrotor Using Improved Method Based on Optical Flow ", *IEEE International Conference on Information and Automation*, pp. 1674-1679, Lijiang, China, 2015.
- [04] S.G. Shuzhi, F. L. Lewis, " Autonomous Mobile Robots, Sensing, Control, Decision Making and Applications ", *Book in CRC Press, Taylor and Francis Group*, 2006.
- [05] N. Ayache and P.T. Sander, " Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception ", *MIT Press, Mass., Hard cover*, 342 pp, 1991.
- [06] M. Boumehraz, Z. Habba, R. Hassani, " Vision Based Tracking and Interception of Moving Target by Mobile Robot Using Fuzzy Control " , *Journal of Applied Engineering Science Technology*, vol. 4, no. 2, pp. 159-165, 2018.
- [07] G. N. Desouza, A. C. Kak, " Vision for Mobile Robot Navigation ", *A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237-267, 2002.
- [08] M. Gupta, B. Uggirala and L. Behera, " Visual Navigation of a Mobile Robot in a Cluttered Environment ", *17th World Congress of IFAC*, Seoul, Korea, 2008.
- [09] A. Miguel, C. Pascual, " Vision Based Fuzzy Control Approaches for Unmanned Aerial Vehicles ", *9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)*, 2015.
- [10] X. Zhang, Y. Wang and Y. Fang, " Vision-based moving target Interception with a mobile robot based on motion prediction and online planning ", *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Angkor Wat, Cambodia, pp. 17-21, 2016.
- [11] G. Alenya and J. L. Crowley, " Time to Contact for obstacle avoidance, in Proceedings. European Conference on Mobile Robotics ", *ECMR, KoREMA*, pp. 19–24. 2009.
- [12] H. M. Becerra, C. Sagüés, Y. Mezouar, and J. Hayet, " Visual Navigation of Wheeled Mobile Robots using Direct Feedback of a Geometric Constraint ", *Autonomous Robots*, vol. 37, no. 2, pp. 137-156, 2014.
- [13] W. Benn, S. Lauria, " Robot Navigation Control Based on Monocular Images: An Image Processing Algorithm for Obstacle Avoidance Decisions ", *Mathematical Problems in Engineering*, pp. 1-14, 2012.
- [14] F. B. Font, A. Ortiz, G. Oliver, " Visual Navigation for Mobile Robots ", *A Survey, Journal of Intel Robot Syst*, vol. 53, pp. 263-296, 2008.

- [15] A. Tasalatsanis, K. Valavanis, Y. Yalcin, " Vision based target and collision avoidance for mobile robots ", *Journal of Intelligent and Robotic Systems*, vol. 48, no. 2, pp. 285-304, 2007.
- [16] J. Serres, F. Ruffier, S. Viollet, N. Franceschini, " Toward Optic Flow Regulation for Wall-Following and Centring Behaviours ", *International Journal of Advanced Robotic Systems*, vol. 3, no. 2, pp. 147-154, 2006.
- [17] S. Kahlouch and K. Achour, " Optical Flow based robot obstacle Avoidance ", *International Journal of Advanced Robotic Systems*, vol. 4, pp. 13-16, 2007.
- [18] H. Chao, Y. Gu, and M. Napolitano, " A Survey of Optical Flow Techniques for Robotics Navigation Applications ", *Journal of Intelligent and Robotic Systems*, vol. 73, pp. 361-372, 2014.
- [19] F. Tajti et al., " Optical Flow based Odometry for Mobile Robots Supported by Multiple Sensors and Sensor Fusion ", *Automatica*, vol. 57, no. 1, pp. 201-211, 2016.
- [20] J. Corso, " Motion and Optical Flow ", *College of Engineering, University of Michigan*, 2014.
- [21] J. R. Serres, F. Ruffier, " Optic flow-based collision-free strategies: From insects to robots ", *Arthropod Structure & Development*, vol. 46, pp. 703-717, 2017.
- [22] A. Miguel et al., "A Pan-Tilt, Camera Fuzzy Vision Controller on an Unmanned Aerial Vehicle ", *The International Conference on Intelligent Robots and Systems, IEEE/RSJ, USA*, 2009.
- [23] Y. Kim and S. Noh, " Fuzzy Visual Navigation Using Behavior Primitives for Small Humanoid Robot", *Robot Intelligence Technol. & Appl, AISC 208*, pp. 823–834, 2012.
- [24] N. A. Mai, and K. Janschek, " Bio-inspired Optical Flow Interpretation with Fuzzy Logic for Behavior-Based Robot Control", *18th International Workshop on Robotics in Alpe-Adria-Danube Region*, Brasov, Romania, 2009.
- [25] A. Hamissi, A. Bazoula, " Fuzzy Visual Path Following by a Mobile Robot ", *1st Mediterranean Conference on Intelligent Systems and Automation (CISA08)*, Annaba, Algeria, 2008.
- [26] L. Huang, " Velocity Planning for a Mobile Robot to Track a Moving Target - A Potential Field Approach ", *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 55-63, 2009.
- [27] L. Kiss., R. Annamária, " A Universal Vision-based Navigation System for Autonomous Indoor Robots ", *Integrated Intelligent Systems Japanese-Hungarian Laboratory Department of Measurement and Information Systems*, Budapest University of Technology and Economics, 2003.
- [28] J. Latombe, " Robot Motion Planning ", *Kluwer Academic Publishers*, 1991.

- [29] D. Filliat, " Robotique Mobile ", *Cours, Ecole Nationale de technique avancés (ENSTA)*, 2011.
- [30] H.Ghazouani, " Navigation visuelle de robots mobiles dans un environnement d'intérieur ", *Thèse Doctorat, Génie Informatique, Automatique et Traitement de Signal, Université de Montpellier*, 2012.
- [31] G. Giralt, R. Sobek, R. Chatila, " A multi-level planning and navigation system for a mobile robot, a first approach to Hilare ", *In:Proc. Of Six th Intl Joint Conf .on Artificial Intelligence*, pp.335 337, 1979.
- [32] S. Thrun, " Learning Metric-Topological Map for Indoor Mobile Robot Navigation ", *Articial Intelligencen*, February 1998.
- [33] C. Braillon, " Détection d'obstacles par fusion de flux optique et stéréovision dans des grilles d'occupation ", *Thèse de Doctorat, institut National Polytechnique de Grenoble*, 2007.
- [34] S. Doncieux et A. Angeli, " Navigation des drones par flux optique ", *Objets volants miniatures: modélisation et commande embarquée*, Hermes-Lavoisier, publisher. Pages 305-328, 2007.
- [35] Y. Cang, N. H. C. Yung, 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.
- [36] F. Cuesta, A. Ollero, " Intelligent Mobile Robot Navigation ", *Springer-Verlag, Berlin Heidelberg*, 2005.
- [37] O. Lefebvre, " Navigation Autonome sans Collision pour Robots Mobiles non holonomes ", *Thèse de Doctorat, Institut National Polytechnique de Toulouse*, 2006.
- [38] B. Bayle, "Robotique Mobile", *Ecole Nationale Supérieure de Physique de Strasbourg, Université Louis Pasteur*, 2006.
- [39] J. Borenstein, H. R. Everett, and L. Feng, " Where am I, Sensors and Methods for Mobile Robot Positioning ", *Technical Report, University of Michigan*, 1996.
- [40] L. Cherroun, R. Mechgoug and M. Boumehraz, " Path Following Behavior for an Autonomous Mobile Robot using Fuzzy Logic and Neural Networks ", *Revue Courrier du Savoir Scientifique et Technique, Université de Biskra*, vol. 12, pp.63-70, 2011.
- [41] J. Borenstein and 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.
- [42] 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.

- [43] J.B. Hayet, L. Frédéric, D. Michel, " A visual landmark framework for indoor mobile robot navigation ", *In IEEE International Conference on Robotics and Automation*, volume 4, pages 3942-3947. Washington (USA), May 11-15 2002.
- [44] J. S. Franco, E. Boyer, " Fusion of multi-view silhouette cues using a space occupancy grid ", *In International Conference on Computer Vision*, pp.1747-1753, 2005.
- [45] M. R. Schmid, M. Maehlich, J. Dickmann, H.J. Wuensche, " Dynamic level of detail 3D occupancy grids for automotive use ", *IEEE Intelligent Vehicles Symposium (IV)*, pp. 269-274, San Diego, CA, 21-24 June 2010.
- [46] K. Kohara, N. Suganuma, T. Negishi, T. Nanri, " Obstacle Detection Based on Occupancy Grid Maps Using Stereo vision System ", *International Journal of Intelligent Transportation Systems Research*, volume 8, pages 85-95, 2010.
- [47] F. Oniga, S. Nedevschi, R. Danescu, M. Meinecke, " Global map building based on occupancy grids detected from dense stereo in urban environments ", *IEEE 5th International Conference on Intelligent Computer Communication and Processing, 2009. ICCP 2009*. 111-117 Cluj Napoca, 27-29 Aug, 2009.
- [48] D. Kimand, R. Nevatia, " Symbolic Navigation with a Generic Map ", *In Proceedings of IEEE Workshop Vision for Robots*, pp.136-145, August 1995.
- [49] K. Horn, B. G. Schunck, " Determining Optical Flow ", *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.
- [50] D. N. Lee, " A theory of visual control of braking based on information about time-to-collision ", *Perception*, 5(4):437-459, 1976.
- [51] D. N. Lee, " Plummeting gannets: a paradigm of ecological optics ", *Nature*, 293, 1981.
- [52] D.N. Lee, D. S. Young and D. Rewt, " How do some resaulters land on their feet ", *Journal of Experimental Psychology, Human Perception and Performance*, 18(4), 1195-1202, 1992.
- [53] A. Negre, " Navigation visuelle par variation des échelles intrinsèques ", *Master thesis*, INPG, Grenoble, FR, 2005.
- [54] J. L. Barron, D. J. Fleet, S. S., Beauchemin, " Performance of optical flow techniques ", *In International Journal of Computer Vision*, volume 12, pp. 43-77, 1994.
- [55] E. Simoncelli, " Distributed Representation and Analysis of Visual Motion ", *PhD thesis*, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1993.
- [56] F. Cherif, " Estimation hiérarchique du mouvement par ondelettes géométriques ", *Thèse doctorat*, Électronique, Université Mohamed Khider de Biskra, 2015.
- [57] C. Grava, " Compensation de mouvement par réseaux neuronaux cellulaires. Application en imagerie médicale ", *Thèse de Doctorat*, Images et Systèmes, l'université INSA de Lyon, 2003.

- [58] A. Betka, " Estimation de Mouvement par les Techniques Méthaheuristique ", *Thèse doctorat*, Électronique, Université Mohamed Khider de Biskra, 2019.
- [59] E. Bruno, " De l'estimation locale à l'estimation globale de mouvement dans les séquences d'images ", *Thèse de doctorat*, Université Joseph Fourier, 2001.
- [60] M.Xavier, " Étude quantitative du mouvement de la paroi du cœur à partir d'images ciné-IRM par des méthodes fréquentielles de flux optique ", *Thèse de Doctorat*, Université de Bourgogne, 2010.
- [61] J. Gabriel, " Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-structuré ", *Thèse de Doctorat*, Institut National Polytechnique de Toulouse, 2005.
- [62] H. Lategahn, W. Derendarz, T. Graf, B. Kitt, J.E. ffertz, "Occupancy grid computation from dense stereo and sparse structure and motion points for automotive applications", *IEEE Intelligent Vehicles Symposium(IV)*, 1931-0587, San Diego,CA21-24, June 2010.
- [63] J. Santos-Victor, G.Sandini, F.Curotto, S.Garibaldi, " Divergent stereo for robot Navigation: learning frombees ", *In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, NewYork, 1993.
- [64] A. Idri, " un modèle intelligent d'estimation de couts de développement de logiciels ", *Thèse de doctorat*, Université du Québec à Montréal, septembre 2003.
- [65] H. Bühler, " Réglage par Logique Floue ", *Première édition, Presses Polytechniques et Universitaires*, Romandes, 1994.
- [66] K. M. Passino and S. Yurkovich, " Fuzzy Control ", *Reading, MA*, Addison Wesley Longman, 1998.
- [67] J. Godjevac, " Idées Nettes sur la Logique Floue ", *Collection Informatique, Presses Polytechniques et Universitaires Romandes*, Lausanne, 1999.
- [68] L. A. Zadeh, " Fuzzy Sets ", *Information and Control*, vol. 8, pp. 338-353, 1965.
- [69] J. M. Mendel, " Tutorial, Fuzzy Logic Systems for Engineering ", *Proceeding of the IEEE*, vol. 83, no. 3, pp. 345-377, March 1995.
- [70] E. H. Mamdani and 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.
- [71] T. Takagi and M. Sugeno, " Fuzzy Identification of Systems and its Applications to Modeling and Control ", *IEEE Transactions on Systems, Man, Cyber*, vol. 15, pp. 116-132, 1985.
- [72] N.N. Karnik, J.M. Mendel, " Operations on type-2 fuzzy sets ", *Fuzzy Systems*, vol. 122, pp. 327-348, 2001.
- [73] L.X. Wang, " Adaptive Fuzzy Systems and Control: Design and Stability Analysis ", *Prentice-Hall, Englewood Clifs, NJ*, 1994.

- [74] P. Y. Glorennec, "Algorithmes d'Apprentissage pour Systèmes d'inférence Floue", *Livre, Edition Hermès Science Publications*, 1999.
- [75] J. M. Mendel, R. I. John, and F. Liu, " Interval type-2 fuzzy logic systems made simple ", *IEEE Transactions, Fuzzy Systems*, vol.14, issue.6, pp.808–821, 2006.
- [76] K. Chafaa, " Structures d'identification et de commande des systèmes non linéaires bases sur les techniques floues ", *Thèse de doctorat*, université de Batna, Algérie, 2006.
- [77] 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.
- [78] R. C. Arkin, " Motor Schema-based Mobile Robot Navigation ", *International Journal of Robotic Research*, vol. 8, pp. 92-112, 1989.
- [79] J. Rosenblatt and D. W. Payton, " A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control ", *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*, Washington DC, vol. 2, pp. 317-324, 1989.
- [80] Y. Cang and D. Wang, " A Novel Behavior Fusion Method for the Navigation of Mobile Robots ", *IEEE International Conference on Systems, Man, And Cybernetics*, pp.3526-3531, Nashville, 2001.
- [81] M. Botros, " Application of Fuzzy Logic in Mobile Robots Control ", *Springer-Verlag Berlin Heidelberg, StudFuzz*, pp.163-193, 2005.
- [82] A. Fatmi, A. Al Yahmadi, L. Khriji, and N. Masmoudi, " A Fuzzy Logic Based Navigation of a Mobile Robot ", *World Academy of Science, Engineering and Technology*, vol. 22, pp. 169-174, 2006.
- [83] X. Yang et 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.
- [84] **M. Nadour**, M. Boumehraz, L. Cherroun and V. Puig, " Mobile Robot Visual Navigation based on Fuzzy Logic and Optical Flow Approaches ", *International Journal of Systems Assurance Engineering and Management, (IJS)*, Springer-Verlag, vol. 10, Issue 6, pp. 1654-1667, 2019.
- [85] A. Manzanera, " VISION ", *Cours M2 IMA* , UPMC – Paris 6, 2006.
- [86] J. Arnspang, K. Henriksen, R. Stahr, " Estimating Time to Contact with Curves, Avoiding Calibration and Aperture Problem ", *6th International Conference on Computer Analysis of Images and Patterns, CAIP '95*, pp. 856-861, London, UK, 1995.
- [87] C. Drocourt, " Localisation et Modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels ". *Thèse de Doctorat*, Université de technologie de Compiègne, Robotique, 2002.
- [88] M. J. Aldon, " Capteurs et méthodes pour la localisation des robots mobiles ", *Rapport technique, Techniques de l'Ingénieur*, Université de Montpellier II, S, vol. 7, pp. 1–19, 2001.

- [89] B. D. Lucas et T. Kanade, " An iterative image registration technique with an application to stereo vision ", *In Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.
- [90] B. D Lucas, " Generalized image matching by the method of differences ", *PhD thesis, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA*, 1984.
- [91] **M. Nadour**, M. Boumehraz, L. Cherroun and V. Puig, " Hybrid Type-2 Fuzzy Logic Obstacle Avoidance System based on Horn-Schunck Method ", *In Journal of Electrotehnică, Electronică, Automatică (EEA)*, vol. 67, no. 3, pp. 45-51, 2019.
- [92] L. Cherroun, " Navigation Autonome d'un Robot Mobile par des Techniques Neuro-Floues", *Thèse de Doctorat, en automatique, Université Mohamed Khider de Biskra*, 2013.
- [93] L. Cherroun, **M. Nadour**, M. Boudiaf and A. Kouzou, " Comparison between Type-1 and Type-2 Takagi-Sugeno Fuzzy Logic Controllers for Robot Design ", *In Journal of Electrotehnică, Electronică, Automatică (EEA)*, vol. 66, no. 2, pp. 94-103, 2018.
- [94] L. Cherroun, " Type-2 Fuzzy Behavior Based Navigation Method for Mobile Robot ", *In Journal of Electrical Engineering*, vol. 17, no. 4, pp. 428-434, 2017.
- [95] J.-S.R Jang, " ANFIS: Adaptive-Network-based Fuzzy Inference System ", *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 23, pp.665-685, 1993.
- [96] H. A. Hagra, " A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots ", *IEEE Transactions on Fuzzy Systems*, vol. 12, pp. 524-539, 2004.
- [97] N.Karnik, Mendel. J. and Q. Liang, " Type-2 Fuzzy Logic Systems ", *IEEE Transactions on Fuzzy Systems*, 7, 643-658(6), 1999.
- [98] H. Hagra, "Type-2 FLCs: A New Generation of Fuzzy Controllers, IEEE Computational Intelligence Magazine", 2, 30-43(1), 2007.
- [99] D. Béréziat, " Traitement des images médicales détection, estimation et analyse du mouvement", *Thèse de Doctorat, Université Pierre et Marie Curie Laboratoire d'Informatique de Paris 6*, 2012.
- [100] **M. Nadour**, L. Cherroun and M. Boumehraz, " Using Optical Flow and VRML to Visual Navigation of a Mobile Robot ", *The Second International Conference on Applied Automatic and Industrial Diagnostics (ICAAID'17)*, Djelfa, Algeria, September 2017.
- [101] L. Cherroun, **M. Nadour** and A. Kouzou, " Type-1 and Type-2 Fuzzy Logic Controllers to Autonomous Robotic Motion ", *IEEE 3rd International Conference on Applied Automatic and Industrial Diagnostics (ICAAID'19)*, Turkey, September 2019.

ملخص:

هذه الأطروحة تعالج تصميم نظام الملاحة البصرية الذاتية للآلي المتحرك ذو العجلات المحمول في بيئة داخلية تعتمد على وحدات التحكم المنطقية الغامضة النوع-1 والنوع-2 ومنهج التدفق البصري. يحتوي نظام التحكم المقترح على وحدتي تحكم المنطق الغامض تكاجي – سوجينو لتجنب العقبات وتقارب الأهداف استنادًا إلى معالجة الفيديو المحصل عليه عن طريق خوارزمية معالجة الصورة . تستخدم وحدة التحكم الأولى في الاتجاه، قيم التدفق البصري المحسوبة بواسطة خوارزمية هورن – شانك لاكتشاف وتقدير مواقع العقبات. لاستخراج المعلومات البيئية، يتم تقسيم الصورة إلى قسمين. يتم استخدام وحدة التحكم الثانية لتوجيه الروبوت إلى الوجهة النهائية. يتم التحقق من كفاءة النهج المقترح في المحاكاة باستخدام صندوق الأدوات (VRML) ثنائية وثلاثية الأبعاد (لغة نمذجة الواقع الافتراضي). تُظهر نتائج المحاكاة التي تم الحصول عليها فعالية نظام الملاحة المرئي المستقل دون تصادم مع العقبات.

كلمات مفتاحية: آلي متحرك ; التجنب البصري للحواجز ; الملاحة البصرية ; التدفق البصري ; الاتجاه نحو الهدف ; التحكم المنطقية الغامضة من نوعين (النوع-1 والنوع-2)

Résumé:

Cette thèse traite la conception d'un système de navigation visuelle pour un robot mobile à roues dans un environnement intérieur basé sur l'association des contrôleurs flous type-1 et type-2 et l'approche du flux optique (OF). Le système de contrôle proposé contient deux contrôleurs flous de Takagi-Sugeno pour l'évitement d'obstacles et la convergence vers un but basé sur l'acquisition vidéo et l'algorithme de traitement d'image. Le premier contrôleur flou de direction utilise des valeurs de flux optiques calculées par l'algorithme Horn-Schunck pour détecter et estimer les positions des obstacles. Pour extraire des informations sur l'environnement, l'image est divisée en deux parties. Le second contrôleur flou est utilisé pour guider le robot vers la destination finale. L'efficacité de l'approche proposée est vérifiée en simulation à l'aide de la bibliothèque VRML en 2D et 3D (Virtual Reality Modeling Language). Les résultats de simulation obtenus montrent l'efficacité de ce système autonome de navigation visuelle sans collision avec des obstacles.

Mots-clés : Robot Mobile ; Navigation visuelle ; Évitement visuel des obstacles ; convergence vers un but ; Cible mobile ; VRML ; Horn-Schunck ; Flux optique ; TTC ; Contrôleur flou de type 1 et type 2.

Abstract:

This thesis deals with the design of a visual navigation system for a wheeled mobile robot in an indoor environment based on the hybridization of fuzzy logic controllers (FLCs) of two types (type-1 and type-2) and the optical flow approach (OF). The proposed control system contains two Takagi-Sugeno fuzzy logic controllers for obstacle avoidance and goal convergence based on video acquisition and image processing algorithm. The first direction controller uses optical flow values calculated by the Horn-Schunck algorithm to detect and estimate obstacle positions. To extract environmental information, the image is divided into two parts. The second FLC is used to guide the robot to the final destination. The efficiency of the proposed approach is verified in simulation using the VRML library in 2D and 3D (Virtual Reality Modeling Language). The simulation results obtained demonstrate the effectiveness of this autonomous visual navigation system without collision with obstacles.

Key words: Mobile Robot; Visual Navigation; Visual Obstacle Avoidance ; Goal Seeking; Moving Target; VRML; Horn-Schunck; Optical Flow.; TTC; Type-1 and Type-2 Fuzzy Controller.