



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

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : *Génie Informatique*

Par :

MEDJROUD Sarah

Sur le thème

Un système de navigation à base de la logique floue : Application sur un système multi robots

Soutenu publiquement le .. / 07/ 2018 à Tiaret devant le jury composé de :

Mr BERBER El-mehdi

MAA Université de Tiaret

Président

Mr BELARBI Mustapha

MCA Université de Tiaret

Examineur

Mr MOKHTARI Ahmed

MAA Université de Tiaret

Encadreur

RESUME

La logique floue est très présente dans notre vie quotidienne sans le constater, en particulier dans l'industrie comme, les micro-ondes, les machines à laver ..., de sa semblance de raisonnement, proche à le raisonnement humain et de sa flexibilité.

Au moyen de l'étude précédente, la logique classique appliquée à la commande de navigation d'une machine a démontré que cette dernière ne donne pas une bonne flexibilité.

Pour remédier ce problème, on a remplacé cette logique par la logique floue basée sur les techniques de l'intelligence artificielle.

Pour obtenir une machine avec un comportement approximatif, presque humain la logique floue est appliquée.

Des tests de simulation numérique ont été réalisés pour valider la faisabilité de cette technique utilisée sur un système multi-agent, plus précisément un système multi-robot lui intégrer un système de navigation flou.

Une étude comparative a démontré que la logique floue a permis d'améliorer les performances dynamiques, ainsi que la robustesse et flexibilité.

Mots clés:

Logique flou – navigation – système multi-agent – système multi robot.

ملخص

إن المنطق الغامض أو الضبابي موجود للغاية في حياتنا اليومية دون أن نلاحظه ، لا سيما في الصناعة مثل الميكروويف والغسالات ... ، وتشابهه في التفكير القريب من التفكير البشري ومرونته. باستخدام الدراسة السابقة، أظهر المنطق الكلاسيكي المطبق على التحكم في التنقل في الجهاز أنه لا يعطي مرونة جيدة. حلا لهذه المشكلة، تم استبدال هذا المنطق بالمنطق الضبابي القائم على تقنيات الذكاء الاصطناعي. للحصول على آلة ذات سلوك تقريبي، يتم تطبيق المنطق الضبابي البشري تقريبا. أجريت اختبارات المحاكاة العددية للتحقق من جدوى هذه التقنية المستخدمة في نظام متعدد الوكلاء، وعلى وجه التحديد نظام متعدد الروبوتات لدمج نظام ملاحظة غامض. أظهرت دراسة مقارنة أن المنطق الغامض أدى إلى تحسين الأداء الديناميكي وكذلك المتانة والمرونة.

كلمات البحث:

منطق ضبابي - نظام ملاحظة - نظام متعدد الوكلاء - نظام روبوت متعدد.

ABSTRACT

The fuzzy logic is very present in our daily life without noticing it, especially in the industry like, the microwaves, the washing machines ..., of its similarity of reasoning, close to the human reasoning and its flexibility.

Using the previous study, the classical logic applied to the navigation control of a machine has shown that it does not give good flexibility.

To fix this problem, this logic has been replaced by fuzzy logic based on the techniques of artificial intelligence.

To obtain a machine with approximate behavior, almost human fuzzy logic is applied.

Numerical simulation tests were carried out to validate the feasibility of this technique used on a multi-agent system, specifically a multi robot system to integrate a fuzzy navigation system.

A comparative study has shown that fuzzy logic has improved dynamic performance as well as robustness and flexibility.

Keywords:

Fuzzy logic - navigation - multi-agent system - multi robot system.

REMERCIEMENTS

Tout d'abord, je remercie ALLAH pour la santé, la volonté, le courage et la patience qui m'ont accompagnée tout au long de la préparation et l'élaboration de ce travail et qui m'ont permis d'achever ce modeste travail.

En second lieu, je tiens à remercier mon encadreur Mr MOKHTARI Ahmed, son précieux conseil et son aide durant toute la période du travail. Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à ma recherche en acceptant d'examiner mon travail Et de l'enrichir par leurs propositions.

Enfin, je tiens également à remercier toute ma famille et mes amis qui par leurs prières et leurs encouragements, j'ai pu surmonter tous les obstacles.

DEDICACE

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chères sœurs Hadjer, Lyna, Maria et mon cher frère Ahmed pour leurs encouragements permanents, et leur soutien moral,

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible,
Merci d'être toujours là pour moi.*

Table des matières

Résumé	ii
ملخص.....	iii
Abstract.....	iv
Remerciements	v
Dédicace	vi
Table des matières	vii
Liste des figures.....	x
Introduction générale.....	1
Chapitre I : Système multi-agent.....	3
I.1 Agent	3
I.1.1 Définition	3
I.1.2 Fonctionnement d'un agent.....	5
I.1.3 Architecture d'un agent	6
I.1.4 Caractéristiques d'un agent	8
I.2 Les systemes multi-agents.....	9
I.2.1 Définition	9
I.2.2 Environnement d'un SMA	11
I.2.3 Domaine d'application	12
I.3 Interaction dans un SMA.....	13
Chapitre II : Logique floue.....	15
II.1 Historique de la logique floue	15
II.2 Logique classique et logique floue.....	16
II.3 Logique floue	17
II.4 Domaine d'application	17

II.5	Concepts et définitions	18
II.5.1	Ensemble flou.....	18
II.5.2	Fonction d'appartenance	19
II.5.3	Carctéristiques d'un sous-ensemble flou.....	20
II.5.4	Opérations sur les sous-ensembles flous	22
II.6	Variable linguistique	23
II.7	Système Flou	24
II.7.1	Base de connaissance	24
II.7.2	Fuzzification	25
II.7.3	Moteur d'inférence floue.....	25
II.7.4	Défuzzification	26
Chapitre III : Conception.....		28
III.1	Domaine d'application	28
III.2	But de travail réalisé.....	28
III.3	Description de système	28
III.4	Architecture d'un robot	29
III.5	Système flou	30
III.5.1	Variables du système flou	31
III.5.2	Variables linguistiques	31
III.5.3	Fuzzification.....	35
III.5.4	Moteur d'inférence	36
III.5.4.1	Implication	37
III.5.4.2	Agrégation.....	39
III.5.5	Défuzzification	40
III.5.6	Algorithme 1	41
III.5.7	Résumé.....	43
III.5.8	Algorithme 2	44
Chapitre IV : Réalisation.....		46
IV.1	Environnement de travail	46
IV.1.1	Environnement matériel	46
IV.1.2	Environnement logiciel	46
IV.2	Développement de l'application.....	47
IV.2.1	Interface principale.....	47

IV.2.2 Interface simulateur.....	49
Conclusion générale	54
Bibliographie.....	55

Liste des figures

Figure I.1 - Agent/Environnement [2].....	4
Figure I.2 - Agent réactif [2]	7
Figure I.3 - Agent cognitif [2].....	8
Figure I.4 - Représentation d'un agent et son environnement [3].....	10
Figure II.1 - Classification de taille de personne en deux sous-ensembles.....	17
Figure II.2 - Formes usuelles des fonctions d'appartenance	19
Figure II.3 - Caractéristiques d'un ensemble flou	21
Figure II.4 - Variable linguistique « Vitesse »	23
Figure II.5 - Structure d'un système d'inférence floue	24
Figure III.1- Environnement d'un SMA	28
Figure III.2 - Architecture d'un agent	29
Figure III.3 - Contrôle en raisonnement flou d'un agent	30
Figure III.4 - Schéma représentant les étapes d'un système flou	31
Figure III.5 - Fonction d'appartenance de la variable « distance »	32
Figure III.6 - Fonctions d'appartenance des valeurs linguistiques.....	33
Figure III.7 - Fonction d'appartenance de la variable « vitesse »	34
Figure III.8 - Fonction d'appartenance de la variable « angle »	35
Figure III.9 - Exemple de fuzzification de la valeur d'entrée « distance »	35
Figure III.10 - Exemple d'implication de la règle 1	37
Figure III.11 - Exemple d'implication de la règle 2	38
Figure III.12 - Exemple d'implication de la règle 3.....	38
Figure III.13 - Exemple d'agrégation des règles floues	39
Figure III.14 - Défuzzification avec la méthode centre de gravité (COG)	40
Figure III.15 - Organigramme d'algorithme de fonctionnement d'un robot.....	42
Figure III.16 - Aperçu récapitulant le système flou	43
Figure III.17 - Schéma récapitulant un système flou	44
Figure IV.1 - Fenêtre principale	48
Figure IV.2 - fenêtre de simulateur	49

Figure IV.3 - Simulateur en fonction d'un labyrinthe.....	50
Figure IV.4 - Simulateur en cas classique et flou	50
Figure IV.5 - Traçabilité de raisonnement flou	51
Figure IV.6 - Schéma de surface pleine après l'agrégation « vitesse »	52
Figure IV.7 - Schéma de surface pleine après l'agrégation « angle »	53

Introduction générale

L'homme a toujours été intéressé par l'utilisation des systèmes d'aide à l'image de l'être humain sur lesquelles il pourrait exercer son pouvoir, exécutant ainsi ses moindres exigences. Le besoin actuel est orienté vers les logiciels qui s'inspirent de l'intelligence humaine comme : le raisonnement, la prise de décision, l'apprentissage et bien d'autres.

Le raisonnement en intelligence artificielle concerne l'ensemble des techniques permettant la manipulation des connaissances déjà acquises afin de produire de nouvelles connaissances. L'intelligence artificielle est une branche active, qui tente de rendre les machines plus intelligentes. Elle a été appliquée dans plusieurs domaines comme l'éducation, le commerce, la médecine, ... etc.

Les premiers travaux en *robotique* mobile qui est l'une des domaines *SMA (systèmes multi-agents)*, en milieu industriel ne conféraient aux robots mobiles qu'une autonomie de déplacement très limitée.

Dans de tels contextes, équiper les systèmes robotiques d'une capacité de décision et, plus particulièrement, de naviguer d'une manière autonome reste l'un des éléments clé dans la mise en œuvre de leur autonomie. En effet, le robot autonome doit être en mesure de prendre des décisions, quant aux mouvements à réaliser, en fonction des informations dont il dispose sur l'environnement ou lui parvenant de ses capteurs, et d'un *système de navigation* capable de gérer les différentes situations déjà citées.

A cet effet, le souci majeur est d'élaborer des techniques de navigation efficaces, La stratégie de navigation peut intégrer les informations, a priori, dont on dispose sur l'environnement. Cependant, lorsque l'environnement devient plus complexe (partiellement connu, dynamique,...), il apparaît indispensable que le robot soit doté de capacités décisionnelles aptes à le faire réagir aux aléas qui peuvent contrarier ses mouvements.

C'est dans cette perspective que s'inscrit notre travail. Il concerne principalement le développement d'un système de navigation à base de la *logique floue* d'un robot mobile autonome dans un environnement dont le comportement est d'éviter les obstacles, et de collecter des objets précis.

Pour une meilleure présentation de notre travail, le mémoire est organisé en quatre chapitres et structuré comme suit :

Dans le premier chapitre nous présentons les notions de base d'un système multi-agent.

Dans le deuxième chapitre nous définissons quelques concepts et les étapes de la logique floue.

Dans le troisième chapitre se trouve la conception d'un système flou, adaptatif pour la résolution de problème a été dégagé à partir de l'analyse précédemment exposée.

Le quatrième chapitre comprend la simulation d'un système de navigation comprenant deux comportements essentiels « flou » et « classique » avec une fonction de collecte.

Introduction :

L'Intelligence Artificielle (IA) est une branche très intéressante de l'informatique qui est née environ les années cinquante-six, son concept vise à permettre aux machines d'imiter une forme d'intelligence réel, à partir de modéliser ou de simuler le comportement humain tels que la perception, la prise de décision, la compréhension, l'apprentissage, etc. Seulement l'IA a vite rencontré un certain nombre de difficultés dont l'expertise, les compétences et les connaissances d'individus différents sont tous intégrés à la même base de connaissance qui, dans la réalité, ils collaborent et communiquent à la réalisation d'un but commun.

L'Intelligence Artificielle Distribuée (IAD) est née au début des années quatre-vingt, le désir de diminuer les insuffisances et d'enrichir l'approche classique de l'IA en proposant la distribution de l'expertise sur un ensemble d'agents, non soumis à un contrôle centralisé, face à être capables d'agir dans un environnement commun et de résoudre les conflits possibles. L'IAD s'intéresse à la modélisation de comportements intelligents qui sont le produit de l'activité coopérative entre plusieurs agents, dans lequel la réalisation des systèmes dits *multi-agents*.

Nous présentons par la suite de ce chapitre l'aspect de l'intelligence de la machine. Les concepts et principes fondamentaux des systèmes multi-agents (SMA) sont introduits afin de placer ce travail dans son contexte de recherche.

1- Agent :

Dans plusieurs domaines le terme « agent » est présent, il n'a pas une seule notion comme il est utilisé de façon assez vague, à l'heure actuelle il n'y a pas de véritable définition. Nous introduisons dans la suite les plus utilisées et s'adaptant à notre sujet.

1.1- Définition :

Russell et Norvig : le concept d'agent est indiqué de la façon suivante : « Un agent est tout ce qui peut être compris comme percevant son environnement à travers des senseurs et comme agissant sur cet environnement par l'intermédiaire d'effecteurs » [1].

Selon cette définition, un agent est une entité quelconque qui peut percevoir son environnement via des capteurs (senseurs) et agir sur l'environnement via des effecteurs. Comme il est présenté dans la (figure I.1) ci-dessous.

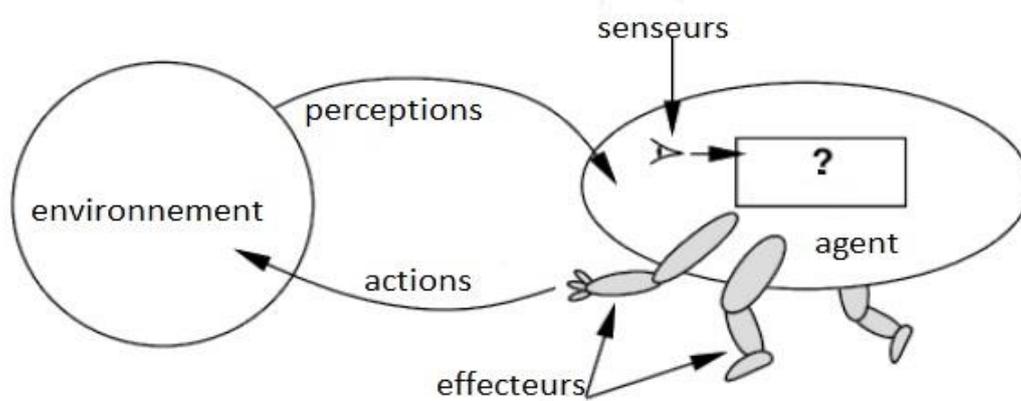


Figure I.1 : Agent/Environnement [2]

Ferber définit le terme « agent » comme un fonctionnement de l'agent et ses propriétés [3] :

On appelle agent une entité physique ou virtuelle :

- a. qui est capable d'agir dans un environnement,*
- b. qui peut communiquer directement avec d'autres agents,*
- c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),*
- d. qui possède des ressources propres,*
- e. qui est capable de percevoir (mais de manière limitée) son environnement,*
- f. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),*
- g. qui possède des compétences et offre des services,*
- h. qui peut éventuellement se reproduire,*
- i. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.*

D'après la définition on peut distinguer quelques idées :

- Une entité physique est une chose concrète qui agit dans l'univers réel comme exemple : un robot, un avion ou une voiture.
- Une entité virtuelle est un composant logiciel, un module informatique parce qu'elle n'existe pas physiquement.
- En systèmes d'IA classique y 'a seulement le raisonnement, en revanche les agents sont capables d'agir.
- Les agents peuvent aussi communiquer entre eux et c'est l'un des moyens principaux d'interaction existant entre les agents.
- Les agents n'ont qu'une représentation partielle de leur environnement, n'ont pas une vision globale de tous ce qu'ils entourent.
- L'agent est un genre « d'organisme vivant » peut communiquer, agir et se reproduire, vise à satisfaire ses besoins et objectifs à partir d'autres éléments (perceptions, actions, communications et ressources) duquel il possède.

1.2- Fonctionnement d'un agent :

Un agent navigue toujours d'après une boucle formée de plusieurs étapes [2] :

- Au tout début, l'agent se trouve dans une certaine configuration initiale ;
- Il perçoit son environnement entièrement ou partiellement grâce à ses capteurs ;
- Il décide qu'elle action à prendre en fonction de sa configuration et de ses perceptions ;
- A chaque fois qu'il reçoit de nouvelles perceptions, le procès se répète.

L'étape de décision de l'agent dépend de son architecture interne. Cette architecture représente la façon de percevoir l'agent et dépend de son processus de raisonnement.

1.3- Architecture d'un agent :

Nous différencions deux grandes catégories d'agents : les agents réactifs et les agents cognitifs. Cette différence se base typiquement sur le procès décisionnel de l'agent et la représentation de l'environnement duquel il dispose. Si l'agent est fourni d'une représentation symbolique de l'environnement dont il est capable de formuler des raisonnements, nous disons qu'il est cognitif, au moment où il dispose que d'une représentation limitée à ses perceptions, donc il est réactif.

a- Agent réactif :

Comme son nom l'indique, un agent réactif réagisse par réflexe aux changements de ses perceptions de l'environnement, ainsi il a un agissement simple (pas intelligent) [4].

Donc l'agent réactif ne fait que réagir aux changements qui se produisent dans l'environnement. C'est-à-dire, un tel agent ne planifier pas son comportement, tout simplement il capte les perceptions et réagit à celles-ci. Du fait qu'il n'y a pas de raisonnement, l'agent réactif peut agir très rapidement [2].

Ferber détermine ce type d'agent comme suit : « Les agents réactifs, définis par le fait même qu'ils n'ont pas de représentation de leur environnement et des autres agents, sont incapables de prévoir ce qui va se passer et donc d'anticiper en planifiant les actions à accomplir» [3].

L'agent réactif agit seulement à partir de la perception actuelle, n'a pas une représentation explicite de son environnement et ne prend pas compte de ses actions passées (pas de mémoire). Comme il ne dispose que d'un protocole et d'un langage de communication réduit.

Le processus de décision d'un agent est effectué sous la forme de la règle ci-dessous :
situation → action [5].

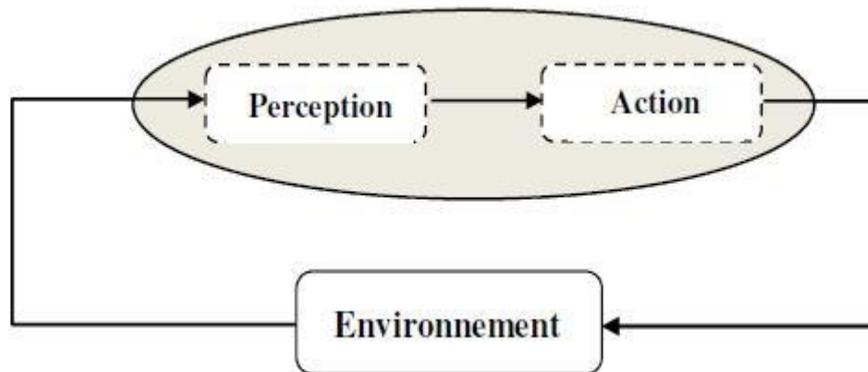


Figure I.2. Agent réactif [2]

b- Agent cognitif :

L'agent cognitif a l'avantage de raisonner, de communiquer par envoi de messages ou via des langages d'interaction élaborés, de négocier, de planifier son comportement et de mémoriser ses actions passées, il possède une représentation explicite de son environnement, des autres agents et de lui-même [2,6].

L'une des architectures cognitives les plus courantes est certainement l'architecture BDI : Belief (Croyance), Desire (Désir), Intention (Intention). Est fondée sur trois concepts de comportements mentaux qui définissent la rationalité d'un agent intelligent [6] :

- Croyance : ce que l'agent dispose comme informations sur son environnement et sur les autres agents agissant sur le même environnement [2].
- Désir : sont les états possibles de l'environnement et parfois, de l'agent lui-même que l'agent souhaiterait voir réalisés. Ce sont les buts que se fixe un agent [2, 5,6].
- Intention : sont les actions qu'un agent a choisi pour atteindre ses désirs. Sont un ensemble de plans qui sont effectués autant que le but relatif n'est pas atteint [2].

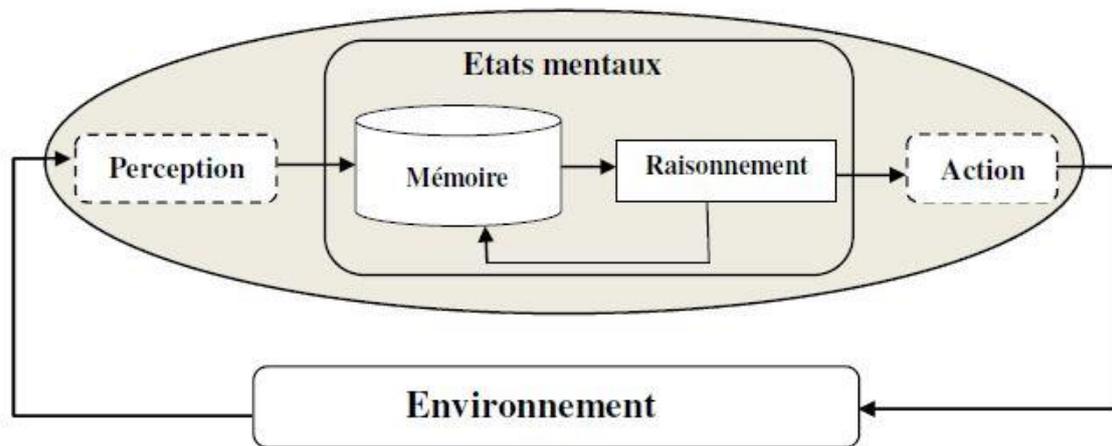


Figure I.3. Agent cognitif [2]

La combinaison de ces deux architectures, se produit la troisième architecture, donc, on introduira dans suit les agents hybrides.

c- Agent hybride :

L'agent hybride est formé pour fusionner entre les capacités réactives et les capacités cognitives. En effet la rapidité de réponse des agents réactifs ainsi que le raisonnement des agents cognitifs. D'où l'agent hybride utilise les deux comportements.

Le choix du ou des types d'agents à utiliser dépend du système multi-agent le plus convenable pour résoudre le problème. Alors, certains SMA n'utilisent qu'un seul type d'agents, d'autres utilisent plusieurs types.

1.4- Caractéristiques d'un agent :

Nous pouvons introduire quelques caractéristiques fondamentales qu'un agent doit avoir pour qu'il puisse être qualifié comme agent intelligent. Ce n'est pas forcément qu'un agent avoir tous ces caractéristiques en même temps.

a- Autonomie :

Les agents ne sont pas dirigés par des commandes venant de l'utilisateur (ou d'un autre agent), on peut dire que le moteur d'un agent c'est lui-même. Il peut accepter ou refuser des requêtes venant d'autres agents. Il dispose d'une liberté de comportement, ce qui le différencie de tous les concepts semblables « objets », « modules logiciels » ou « processus » [3].

b- Réactivité :

Les agents perçoivent leur environnement et réagissent aux changements dans le temps requis [2].

c- Initiative :

Le comportement des agents est déterminé à partir des buts qu'ils suivent pour atteindre, comme ils peuvent développer des actions qui ne sont pas seulement des réponses à leur environnement [2].

d- Habilité sociale :

Pour atteindre le but un agent peut demander l'aide d'autres agents dont le but est de partager la réalisation de tâches [2].

e- Mobilité :

Dans des applications déterminées il peut être intéressant qu'un agent a la possibilité de déplacer d'un nœud à un autre dans un réseau (ou d'un serveur à un autre), le code de l'agent est transporté et ses données aussi, lors de sauts de nœud à nœud [2, 5].

2- Les Systèmes multi-agents :

Un système multi-agent peut être introduit comme un ensemble d'agents partageant un environnement commun et capable d'interagir entre eux et sur l'environnement qui les entoure. Il existe plusieurs définitions des systèmes multi-agents, on présente dans ce qui suit les plus simples et convenant à notre sujet.

2.1- Définition :

Ferber définit un système multi-agent comme suit [3] :

On appelle système multi-agent (ou SMA), un système composé des éléments suivants:

1. Un environnement E , c'est-à-dire un espace disposant généralement d'une métrique.
2. Un ensemble d'objets O . Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E . Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
3. Un ensemble A d'agents, qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système.
4. Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.
5. Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O .
6. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification.

La figure I.4 donne une illustration d'un système multi-agent. D'après Ferber.

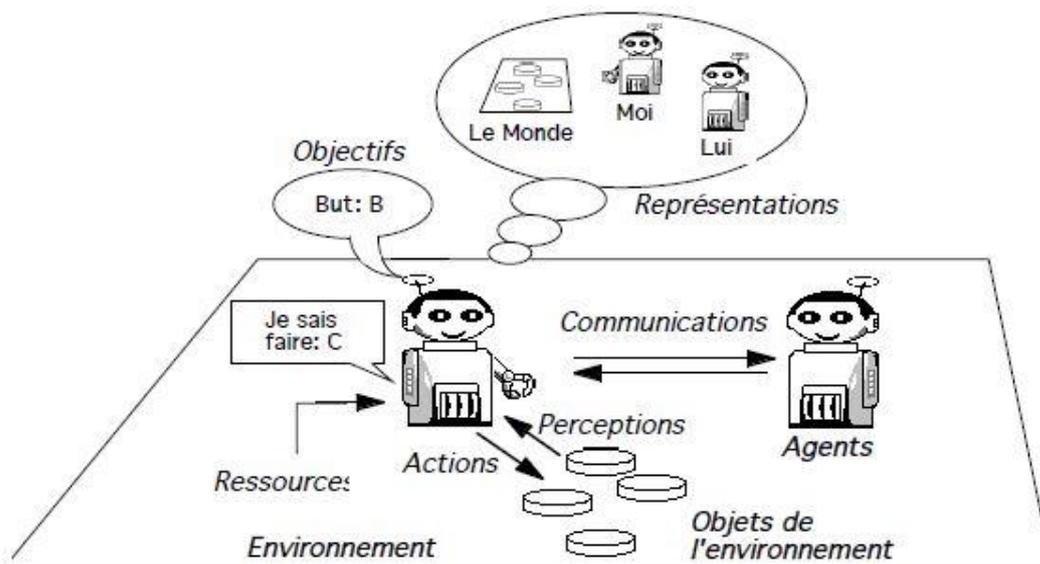


Figure 1.4. Représentation d'un agent et son environnement [3]

On voit que Ferber a décomposé un système multi-agent (SMA) en cinq dimensions, E (environnement), O (objet), A (agent), R (relation) et Op (opération).

Une autre définition, présentée par Wooldridge et Jennings, détermine un système multi-agent comme étant : « Un SMA est un ensemble d'agents ayant des buts ou des tâches, et qui interagissent pour les accomplir » [7].

Wooldridge et Jennings ont défini un SMA par une très simple façon, pour dire qu' un SMA est juste un ensemble d'agents coopérants pour atteindre un but.

2.2- Environnement d'un SMA:

L'environnement décrit l'entourage dans lequel l'agent perçoit et sur lequel il peut agir, aussi les différents événements qui s'y introduisent [4]. La nature de l'environnement et ses propriétés qui influencent de façon très importante les actions de l'agent, nous exposons dans ce qui suit les caractéristiques les plus importantes [6] :

- ***Complètement observable (vs. Partiellement observable) :***

Si l'agent a accès à l'état total de l'environnement à chaque instant, ce dernier est dit complètement observable, dans ce cas l'agent peut acquérir une information complète et correcte sur l'état actuel de son environnement, grâce à ses capteurs. Dans un environnement partiellement observable uniquement une partie de l'information est accessible [2].

- Jeu d'échecs est complètement observable.
- Jeu de cartes est partiellement observable.

- ***Déterministe (vs. non Déterministe ou Stochastique) :***

Si l'état suivant est entièrement déterminé par l'état courant et l'action exécutée par l'agent ou les agents, donc l'environnement est déterministe, autrement il est stochastique. Dans un environnement non déterministe, une action n'a pas un seul effet assuré [2].

- Jeu d'échecs est déterministe.
- Jeu de cartes est stochastique.

- ***Dynamique (vs. Statique) :***

Si l'environnement change lorsqu'un agent ou les agents agissent, donc ce dernier est dynamique sinon il est statique. L'état de l'environnement dynamique repose sur des actions du système qui se trouve dans cet environnement et aussi des actions d'autres processus. Comme les changements ne peuvent pas être devinés par le système. Un environnement statique ne se modifie pas sans que le système agisse [2].

- ***Continu (vs. Discret) :***

Le nombre de perceptions et d'actions possibles dans un environnement continu est illimité et indéfini. Dans l'environnement discret, un nombre fini et nettement précis d'états [2].

- ***Mono-agent (vs. Multi-agent):***

Dans un environnement mono-agent, un agent résout seul le problème, il ne doit pas tenir compte les évènements et les actions exécutés par d'autres agents [2].

- Les jeux sans et avec adversaire.

2.3- Domaine d'application :

Les domaines d'applications des systèmes multi-agents sont tellement nombreux. On mentionnera que les essentielles directions. On peut prendre en compte qu'il existe cinq grandes classes d'applications des systèmes multi-agents : la résolution de problèmes, la robotique distribuée, la simulation multi-agent, la construction de mondes hypothétiques et la conception génétique de programmes [3].

2.3.1- La résolution de problèmes :

La résolution de problèmes, dans tous les sens du terme concerne en fait tous les cas dans lesquels des agents logiciels réalisent des tâches importantes aux êtres humains. Cette classe s'oppose aux applications robotique distribuée en ce sens que les agents sont uniquement informatiques et n'ont pas de structure physique visible.

2.3.2- La simulation multi-agent :

La simulation est une discipline importante de l'informatique qui repose à examiner les propriétés de modèles théoriques du monde entourant. La physique, la chimie, la biologie, l'écologie, la géographie et les sciences sociales spécialement font un énorme emploi des simulations pour mieux expliquer et de prédire les phénomènes naturels.

2.3.3- La construction de mondes synthétiques :

Quoiqu'il ne s'agisse pas d'application, puisqu'elle ne permet pas de résoudre un problème objectif, n'utilise pas d'agents physiques et ne simule aucun monde réel. La construction de mondes synthétiques, tient une vaste contribution dans les recherches montant sur les systèmes multi-agents puisqu'elle permet d'étudier quelques mécanismes d'interaction de façon plus fine que ne pourrait le faire une vraie application.

2.3.4- La robotique distribuée :

La robotique distribuée c'est la réalisation non pas d'un seul robot, mais d'un ensemble de robots qui coopèrent pour atteindre un objectif. À l'opposition du domaine d'application précédant (la construction de mondes synthétiques), La robotique distribuée emploie des agents physiques qui circulent dans un environnement réel.

2.3.5- La conception kénétique de programmes :

La kénétique introduit un nouveau mode de conception de systèmes informatiques, cherchant à dépasser les techniques actuelles pour réaliser des logiciels distribués aptes d'évoluer par interaction, adaptation et reproduction d'agents relativement autonomes et fonctionnant dans des univers physiquement distribués.

3- Interaction dans un SMA :

Le concept d'interaction représente l'essence d'un système multi-agent puisque c'est grâce à elle que les agents vont pouvoir produire des comportements collectifs complexes. La fonction d'interaction repose sur un ensemble des mécanismes lui permettant de faire le lien avec ce qui l'entoure (son environnement et d'autres agents) [6].

D'une autre vue, L'interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Etant, un agent interagit avec un autre produira à la fois une puissance comme un problème. Interagir permet à un agent de partager des informations et des services pour accomplir ses buts et d'éviter les conflits. Une interaction, si elle est commencée, elle doit se déroulé correctement et se terminé correctement. Pour ceci les interactions sont structurées selon des schémas typiques appelés protocoles. Les protocoles d'interactions permettent aux agents d'échanger des messages structurés et de contrôler l'échange de ces messages et faciliter leur coordination. Un protocole d'interaction définit des règles qui doivent être respectées par les agents en cours la conversation [5].

En générale, on distingue plusieurs situations d'interactions entre les agents : la communication, la coopération, la négociation et la coordination. Leur classification se fait selon plusieurs critères [2]:

- La présence d'objectifs communs ou compatibles.
- L'accès à des ressources communes.
- La répartition des compétences au sein des agents.

Conclusion :

Naissance de l'Intelligence Artificiel et la conception des systèmes multi-agents ont un grand effet sur le développement et la réalisation de systèmes complexes. D'abord ce chapitre a abordé les notions d'agent et de systèmes multi-agents. Ces systèmes sont composés d'entités intelligentes qui peuvent interagir d'une manière comportementale comme les humains. Ensuite, les caractéristiques de l'agent et de l'environnement, les domaines d'applications où peut trouver les systèmes multi-agents. Finalement les interactions entre les agents.

Dans le chapitre suivant, nous détaillerons comment un agent peut raisonner juste comme nous les humains.

Introduction :

Dans le fonctionnement du cerveau humain les imperfections sont marquantes d'une manière intéressante. L'homme est capable de classer les éléments de la nature nativement, il sait estimer l'âge d'un individu en l'observant. Il sait aussi exprimer des données vagues «*grand de taille*», imprécises «*la plage est à 100 m*», mal définies «*il fait frais*» et incertitude «*très probable qu'il arrive aujourd'hui*», dont la validité n'est pas exacte.

La préoccupation d'automatiser les actions humaines, naturelles et imprécises dans le domaine d'aide à la décision ou du contrôle, c'est la capacité de gérer l'incertitude et l'imprécision que nous nous sommes occupés à la logique floue.

Les bases théoriques de la logique floue ont été fondées en 1965 par le professeur Lotfi A. Zadeh, de l'université de Berkeley en Californie. Il a introduit la notion de sous-ensemble flou pour correspondre aux problèmes auxquels se trouvent face à nombreux systèmes complexes, qui doivent traiter des informations de nature imparfaite [8].

La logique floue (*fuzzy logic*) est en journal. Au tout début, elle a été appliquée dans des domaines non-techniques, comme : le commerce, la jurisprudence et la médecine, afin de compléter les systèmes experts. Au cours de l'année 1985, les Japonais ont commencé à utiliser la logique floue dans les industries pour résoudre les problèmes de réglage et de commande. Ces temps derniers, une extrême activité de recherche en Amérique et en Europe est remarquée introduisant le principe du réglage par logique floue [9].

Ce chapitre est dédié à la présentation des principes et éléments de base de la logique floue, auquel notre travail repose.

1- Historique de la logique floue :

- En 1965, Lotfi A. Zadeh, professeur en Electronique de l'Université de Californie à Berkeley, publie «Fuzzy Sets», qui présente la théorie des ensembles flous par extension, de la logique floue. Zadeh a observé que la logique classique est limitée, elle ne peut pas représenter des idées artificielles ou vagues, d'où il a inventé la logique floue pour permettre aux ordinateurs d'être semblables au processus du raisonnement humain [10].
- La première application de la logique floue a été effectuée par *Mamdani* en 1975 à Londres qui applique la théorie inventée par *Zadeh* sur la conduite d'un moteur à vapeur [11].

- Plusieurs applications ont alors vu en Europe, pour des systèmes parfois très complexes, telle la régulation de fours de cimenterie réalisée par la société *F. L Smidt-Fuller* en 1978 [11].
- Seiji Yasunobu et Soji Miyamoto d'Hitachi, En 1985, ils ont fait des simulations qui prouvaient la hauteur des systèmes de contrôle flou pour le chemin de fer de Sendai. Leurs idées ont été adoptées, et les systèmes flous ont été utilisés pour le contrôle d'accélération et de freinage lorsque la ligne a ouvert en 1987 [10].
- Toujours en 1987, lors d'une réunion internationale de chercheurs flous à Tokyo, Takeshi Yamakawa a montré l'emploi du contrôle flou, via un ensemble de puces logiques floues dédiées, dans une expérience de "pendule inversé". Qui a été un problème de contrôle classique. Aussi le métro de *Sendai* [10].
- Logique floue à immerger notamment au Japon Grâce au chercheur japonais *M.Sugneo*. En 1990, il y avait une généralisation de l'utilisation de cette technique.

2- Logique classique et logique floue :

Une proposition est soit vraie, soit fautive en logique classique (1 ou 0). Si on prend par exemple deux ensembles qui représentent la taille des personnes « moins de 1,70 m » et « plus de 1,70 m », alors toutes les tailles plus de 1,70 m sont considérées comme appartenant à l'ensemble « plus de 1,70 m » et ils prennent la valeur 1. Toutes les tailles moins de 1,70 m ne sont pas considérées comme appartenant à l'ensemble « plus de 1,70 m », ils prennent la valeur 0.

Néanmoins, l'humain peut différencier si une personne est grande de taille ou petite malgré, il ne connaît pas les valeurs exactes, donc le concept de la logique floue est inspiré du raisonnement humain dont il repose sur des connaissances inexacts, incertaines ou imprécises.

La logique floue permet de définir des sous-ensembles comme « grand » ou « petit », en donnant la possibilité pour une valeur d'appartenir plus ou moins à chacun de ces sous-ensembles [8].

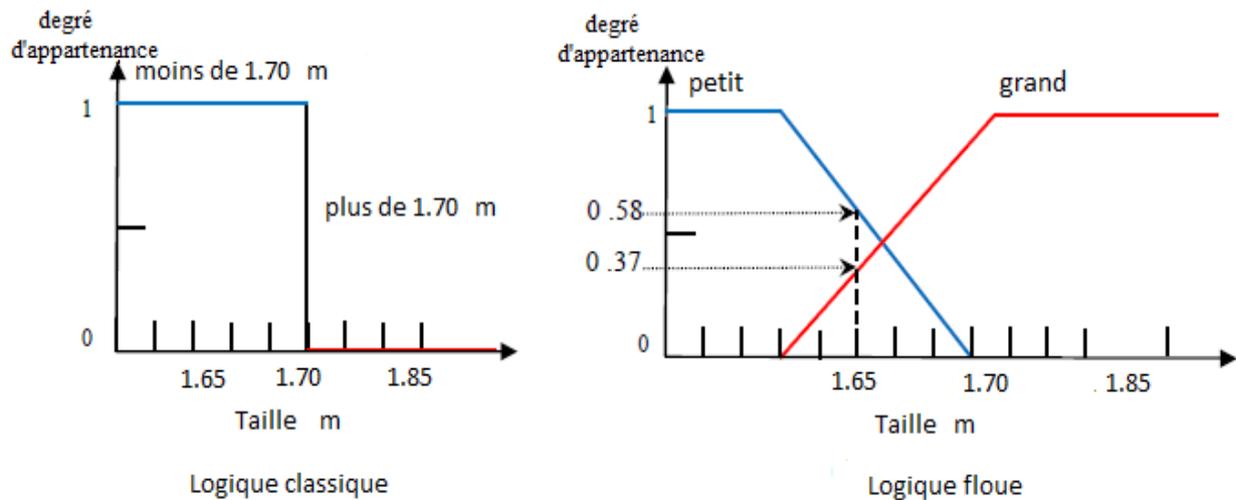


Figure II.1. Classification de taille de personne en deux sous-ensembles

3- Logique floue :

La logique floue a été développée par Lofti Zadeh à partir de sa théorie des sous-ensembles flous. Les sous-ensembles flous sont une manière mathématique d'exprimer l'imprécision de la langue naturelle. La logique floue est également appelée "logique linguistique" parce que ses valeurs sont exprimées par des mots du langage courant : « faible, gros, jeune, si fort, près de, lent, presque parfait ... ». La logique floue est considérée comme une généralisation de la logique classique [12].

Les connaissances dont nous disposons sont généralement imprécises, soit parce qu'elles sont exprimées en langage naturel par un observateur qui n'est pas précis, soit parce qu'elles sont obtenues avec des instruments d'observation qui produisent des erreurs de mesure [13].

La logique floue a comme intention de traiter la représentation des connaissances imprécises, pour pouvoir construire des systèmes artificiels qui réagissent comme les humains [8].

4- Domaine d'application :

La logique floue est présente presque dans tous les domaines, on cite quelques domaines d'applications de la logique floue :

- Robotique
- Systèmes experts
- Reconnaissance des formes
- Classification
- Traitement d'images
- Recherche documentaire
- Langage de programmation
- Commande de processus

5- Concepts et Définitions :

5.1- Ensemble flou :

Le concept d'ensemble flou produit un caractère graduel de l'appartenance d'un élément à un ensemble donné. Pour une bonne représentation des connaissances et des termes imprécises que les humains manipulent quotidiennement. Ces connaissances sont définies comme une collection d'objets de notre intuition et notre intellect. Tous simplement les ensembles flous sont inventés pour modéliser l'incertitude liée au flou, à l'imprécision et au manque d'information dans une situation quelconque [14, 15, 10].

Mathématiquement il est défini comme suit :

Définition 1:

Un sous-ensemble *classique* A de X est défini par une fonction caractéristique μ_A qui prend la valeur 0 pour les éléments de x n'appartenant pas à A et la valeur 1 pour ceux qui appartiennent à A [13] : $\mu_A : x \rightarrow \{0,1\}$

Soit x une couleur (gris) et on a deux ensembles le blanc et le noir, si on est en logique classique (booléenne) donc, x n'appartient ni au blanc ni au noir (on sait que le gris est un mélange entre le blanc et le noir).

Là où Zadeh a voulu sortir de cette logique à deux valeurs uniquement, et donner une chance à ces x et elle est devenue un sous-ensemble flou.

Donc le concept de sous-ensemble flou a été introduit pour éviter la négligence des éléments qui appartiennent partiellement à une classe, il a inventé une logique qui peut définir des connaissances imprécises ou incertaines.

Définition 2 :

Un sous-ensemble *flou* A de X est défini par une fonction d'appartenance qui associe à chaque élément x de X , le degré $\mu_A(x)$, compris entre 0 et 1, avec lequel x appartient à A [13] : $\mu_A : x \rightarrow [0,1]$

Si on prend le même exemple précédant, x la couleur gris, A un sous-ensemble (blanc) et B un sous-ensemble (noir) on a : x appartient à A avec 50 % ($\mu_A = 0.5$) et à B avec 50 %, dans le cas d'un gris foncé : x appartient à A avec 30 % ($\mu_A = 0.3$) et à B avec 70% ($\mu_B = 0.7$).

5.2- Fonction d'appartenance:

Un ensemble flou est décrit par sa fonction d'appartenance, il existe différentes formes de fonctions d'appartenance, les plus utilisées sont les formes triangulaires et trapézoïdales (figure II.2).

La forme de la fonction d'appartenance est choisie arbitrairement suivant les conseils de l'expert, ou d'après une étude statistique.

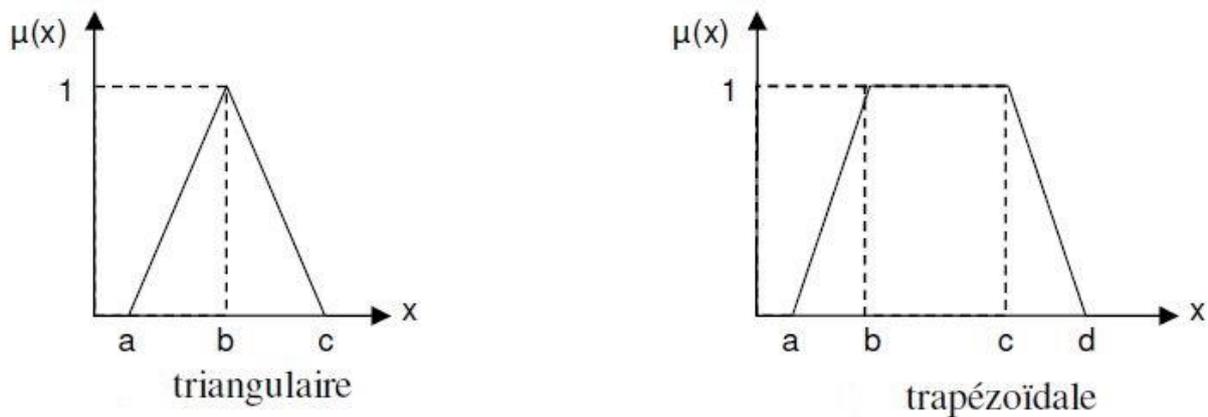


Figure II.2. Formes usuelles des fonctions d'appartenance

La fonction d'appartenance est déterminée par l'une des fonctions suivantes :

➤ Triangulaire :

$$\mu_A(x) = \begin{cases} 0 & \text{si } x < a \\ \frac{x-a}{b-a} & \text{si } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{si } b \leq x \leq c \\ 0 & \text{si } x > c. \end{cases}$$

Où a, b et c sont des nombres réels et $(a < b < c)$.

➤ Trapèze :

$$\mu_A(x) = \begin{cases} \frac{x-b}{b-a} & \text{si } a < x \leq b \\ 1 & \text{si } b \leq x \leq c \\ \frac{d-x}{d-b} & \text{si } c \leq x \leq d \\ 0 & \text{autrement.} \end{cases}$$

Où a, b, c et d sont des nombres réels et $(a \leq b \leq c \leq d)$.

5.3- Caractéristiques d'un sous-ensemble flou :

Pour bien montrer la différence entre un sous-ensemble *classique* et un sous-ensemble *flou*, c'est par ces caractéristiques suivantes d'un sous-ensemble flou A de X .

a- Support :

Le *support* de A , notée $\text{supp}(A)$, est la partie de X sur laquelle la fonction d'appartenance de A n'est pas nulle [13] :

$$\text{supp}(A) = \{x \in X \mid \mu_A(x) \neq 0\}$$

b- Hauteur :

La *hauteur*, notée $h(A)$, du sous-ensemble flou A de X est la plus grande valeur prise par sa fonction d'appartenance [13] :

$$h(A) = \sup \mu_A(x) \mid x \in X$$

sup : borne supérieure.

Le sous-ensemble flou A de X est *normalisé* si sa hauteur $h(A)$ est égale à 1.

c- Noyau :

Le *noyau* de A de X , notée $\text{noy}(A)$, est l'ensemble des éléments de X pour lesquels la fonction d'appartenance de A vaut 1 [13] :

$$\text{noy}(A) = \{x \in X \mid \mu_A(x) = 1\}$$

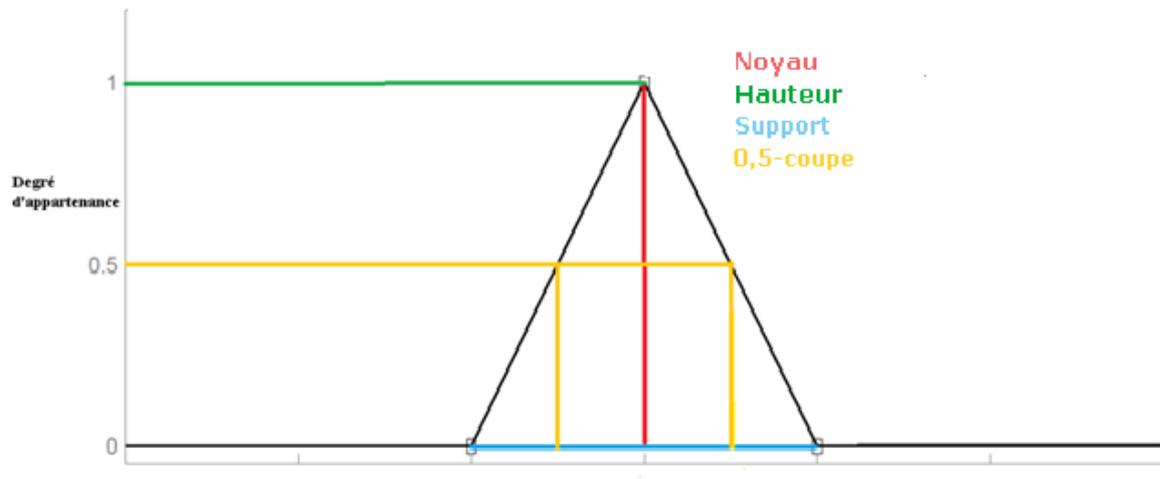


Figure II.3. Caractéristiques d'un ensemble flou

d- Cardinalité :

La cardinalité d'un sous-ensemble flou A de X , notée $|A|$, est le nombre d'éléments appartenant à A pondéré par leur degré d'appartenance. Formellement, pour A fini [16] :

$$|A| = \sum \mu_A(x) \quad : x \in X$$

e- α -coup :

Une α -coupe de A est le sous-ensemble classique des éléments ayant un degré d'appartenance supérieur ou égal à α :

$$\alpha\text{-coupe}(A) = \{x \in X \mid \mu_A(x) \geq \alpha\}$$

Remarque :

Si A est un sous-ensemble classique, devient :

$$\text{supp}(A) = \text{noy}(A) \text{ et } h(A) = 1$$

Ou $h(A) = 0$ si $A = \emptyset$.

5.4- Opérations sur les sous-ensembles flous :

Les opérations existant sur les sous-ensembles classiques d'un même univers de référence, sont généralisées pour être utiles de la même façon sur les sous-ensembles flous.

a- Egalité :

Deux sous-ensembles flous A et B de X sont dits égaux s'ils ont des fonctions d'appartenance égales en tout point de X . Formellement, $A = B$ si et seulement si [16] :

$$\forall x \in X, \mu_A(x) = \mu_B(x)$$

b- Inclusion :

Soient A et B deux sous-ensembles flous de X . Si pour n'importe quel élément x de X , x appartient toujours moins à A qu'à B , alors on dit que A est inclus dans B ($A \subseteq B$). formellement, $A \subseteq B$ si et seulement si [16] :

$$\forall x \in X, \mu_A(x) \leq \mu_B(x)$$

c- Union :

L'union de deux sous-ensembles flous A et B de X est le sous-ensemble flou constitué des éléments de X affectés du plus grand des degrés avec lesquels ils appartiennent à A et B . formellement, $A \cup B$ est donné par [16] :

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

d- Intersection :

L'intersection de deux sous-ensembles flous A et B de X est le sous-ensemble flou constitué des éléments de X affectés du plus petit des degrés avec lesquels ils appartiennent à A et B . formellement, $A \cap B$ est donné par [16] :

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

e- Complément :

Le complément d'un sous-ensemble flou A de X est noté \bar{A} . il est défini à partir de la fonction d'appartenance de A par [16] :

$$\forall x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Remarque :

- En logique floue, le principe du tiers exclu est contredit : $A \cup \bar{A} \neq X$, autrement dit $\mu_{A \cup \bar{A}}(x) \neq 1$.
- En logique floue, un élément peut appartenir à A et non A en même temps : $A \cap \bar{A} \neq 0$, autrement dit $\mu_{A \cap \bar{A}}(x) \neq 0$. Notons que ces éléments correspondent à l'ensemble $supp(A) - noy(A)$.

6- Variable linguistique :

L'idée a été de simplifier le modèle par l'incorporation de langage naturel, donc les valeurs de cette variable linguistique ne sont pas numériques, mais bien symbolique en mots ou expressions du langage naturel.

Une variable linguistique est caractérisée par un triplet $(x, T(x), U)$ où :

x : le nom de variable ;

$T(x)$: c'est l'ensemble des noms des valeurs linguistiques de x dont chaque valeur représente un sous-ensemble flou défini dans U , qui assure la correspondance avec l'univers numérique ;

U : l'univers de discours.

Comme il est présenté sur la figure ci-dessous, on prend une variable linguistique (vitesse) :

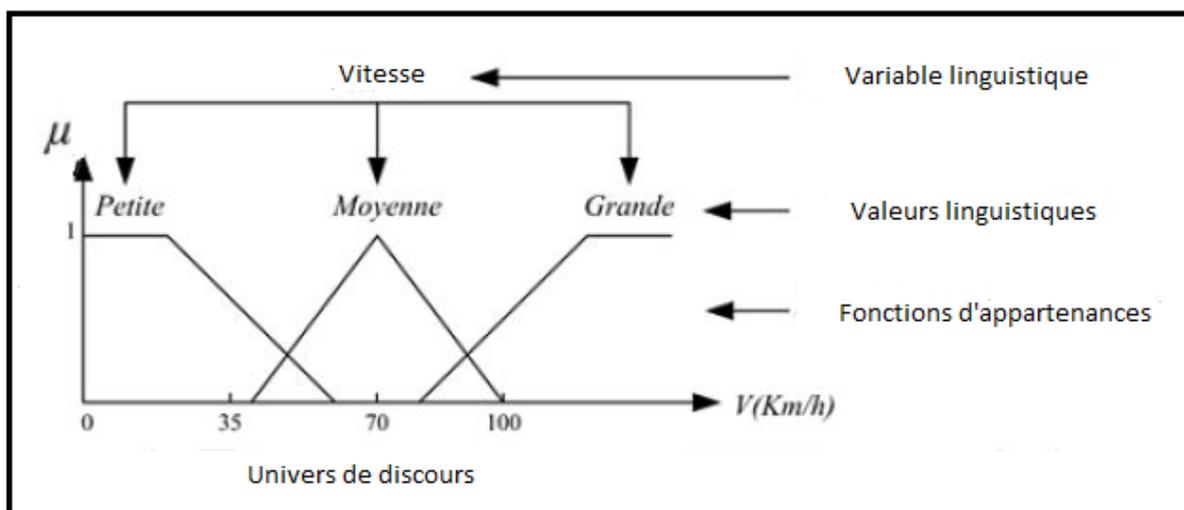


Figure II.4. Variable linguistique « Vitesse »

7- Système flou :

Un système flou est formé de quatre blocs essentiels comme indiqué sur La figure II.5 :

- *Base de connaissances.*
- *Fuzzification .*
- *Moteur d'inférence.*
- *Défuzzification.*

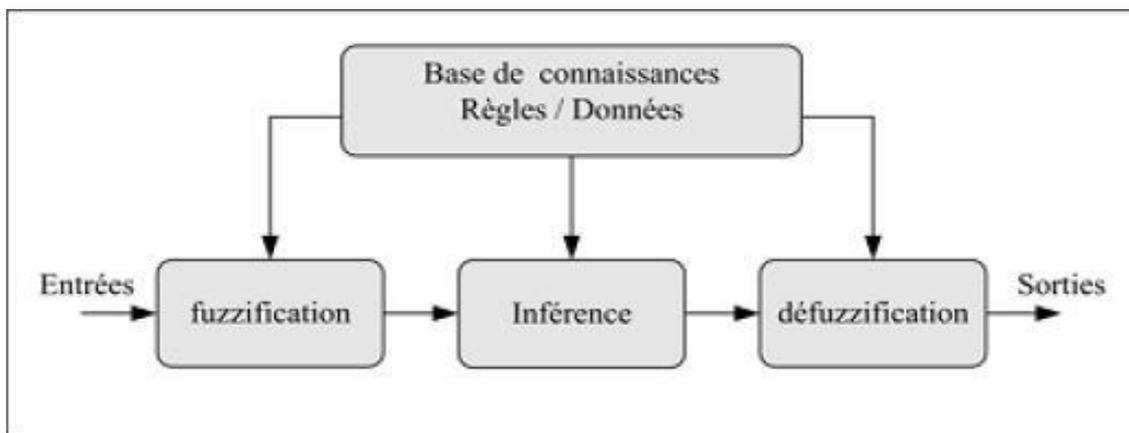


Figure II.5. Structure d'un système d'inférence floue

7.1- Base de connaissance :

La base de connaissances comporte d'une base de données et de base des règles floues [15].

➤ **La base de données :**

Elle contient les définitions des fonctions d'appartenance (formes et paramètres) associées aux variables d'entrée/sortie [14].

➤ **La base des règles floues :**

Elle réunit l'ensemble des règles floues de type « *Si-Alors* » écrites en langage naturel basées sur la connaissance d'un expert [15] :

Les règles décrivent les relations qui doivent exister entre les variables d'entrée et de sortie afin que cette dernière atteigne la consigne désirée [17].

7.2- Fuzzification :

Cette étape consiste de passer d'une valeur précise à une valeur floue (linguistique). Pour se faire, on associe à chaque variable d'entrée un degré d'appartenance à un sous-ensemble flou donné. À l'aide des fonctions d'appartenance définies dans la base de données [14, 15,8].

7.3- Moteur d'inférence floue :

C'est un processus de décision. Il permet de prendre une décision à partir de la base des règles floues en observant un fait et en exploitant le raisonnement humain [15]. Le principe pour construire un ensemble de règles floues est basé sur la connaissance ou l'expérience d'un être humain [10].

Les opérateurs ET et OU sont appliqués dans l'inférence floue comme suit : l'opérateur ET s'applique à l'intérieur d'une règle entre les variables (appelé implication) alors, que l'opérateur OU s'applique à l'intérieur d'une règle entre les variables, et à l'extérieur lie les différentes règles (appelé agrégation des règles) [15]. En générale deux ou plusieurs règles sont activées en même temps.

La règle floue est exprimée sous la forme « *si (condition) alors (conclusion)* », exemple d'un feu de circulation routière :

R1 : Si le *feu est rouge* ET la *distance et proche* ALORS la *vitesse est petite*

- Les clauses « *feu est rouge* » et « *distance et proche* » sont appelées les antécédents de la règle.
- La conclusion « *vitesse est petite* » est appelée le conséquent de la règle [17].

Amenant à une sortie basant sur des faits observés. En regroupant les règles pour obtenir un sous-ensemble flou de sortie [10], Comme méthodes d'inférences on peut citer :

➤ La méthode d'inférences max-min (méthode de Mamdani) :

La méthode max-min joue le rôle du maximum si l'opérateur était OU et du minimum si était ET, au niveau de la condition et la conclusion [15].

➤ La méthode d'inférence Max-Produit (méthode de Larsen) :

Au niveau de la condition ET et OU ont le même rôle de la méthode précédente, par contre au niveau de la conclusion, l'inférence doit être réalisé par le produit [9].

➤ **La méthode d'inférence Somme-Produit (méthode de Sugeno) :**

La méthode somme-produit joue le rôle du somme si l'opérateur était OU et du produit si était ET, au niveau de la condition et la conclusion [15].

7.4- Défuzzification :

Cette étape permet de produire une valeur numérique à partir de la fonction d'appartenance résultante de l'étape précédente (inférence), elle peut être réalisée à travers plusieurs méthodes [15, 9]. Les méthodes existantes :

➤ **Méthode de centre de gravité (COG) :**

Cette méthode fournit une valeur numérique basée sur le centre de gravité de l'ensemble flou [10].

La valeur numérique μ_r , produite de centre de gravité de l'ensemble flou inféré [15] :

Dans le cas discret :

$$u_r = \frac{\sum_{i=1}^n u_i \mu_{Res}(u_i)}{\sum_{i=1}^n \mu_{Res}(u_i)}$$

n : le pas dans l'univers de discours.

μ_i : les valeurs numériques.

$\mu_{Res}(\mu_i)$: le degré d'appartenance.

Dans le cas continu :

$$u_r = \frac{\int u \cdot \mu_{Res}(u) du}{\int \mu_{Res}(u) du}$$

➤ **Méthode moyenne pondérée (WAM) :**

Cette méthode est convenable pour les ensembles flous avec des fonctions d'appartenance de sortie symétriques et produit des résultats très proches de la méthode COG. Cette méthode nécessite moins de calculs. Chaque fonction d'appartenance est pondérée par sa valeur d'adhésion maximale [10].

➤ **Méthode de la moyenne des maxima (MOM) :**

Dans cette méthode, la valeur de sortie est évaluée par l'abscisse du point correspondant au centre de l'intervalle M pour lequel la fonction d'appartenance est maximale. Cette valeur est fournie par l'expression:

$$y^* = \left(\frac{\inf (M) + \sup (M)}{2} \right)$$

Où $\inf (M)$ et $\sup (M)$ sont les bornes inférieure et supérieure respectivement de l'intervalle M [9].

Conclusion :

Dans ce chapitre, nous avons exposé les concepts fondamentaux de la logique floue. Initialement, la raison et l'histoire d'apparition de la logique floue, la déférence entre elle et la logique classique, les domaines d'application intégrant la logique floue. Ensuite, la théorie des ensembles flous, ses caractéristiques et opérations. Enfin, le système flou et ses étapes.

Introduction :

Dans ce chapitre, nous présentons la conception d'un système de navigation à base de la logique floue intégré dans un système multi robots. Afin qu'il pourra naviguer et collecter des objets dans le comportement proche à l'humain.

Initialement nous décrivons notre environnement ensuite, les démarches du raisonnement flou. Nous concluons avec un résumé récapitulatif pour mieux fixer la logique flou et ses étapes.

1- Domaine d'application :

Notre domaine choisi est la robotique distribuée ou qu'on appelle les systèmes multi robots, qui adopte des agents physiques. Les robots ont comme tâche la collecte (des cerises par exemple) qui nécessite un système de navigation.

2- But de travail réalisé :

Le but de travail est d'intégrer un système de navigation dans le robot à base de la logique floue.

3- Description de Système :

Notre système comprend un ensemble de robots qui se déplacent en évitant les obstacles et en collectant les cerises, à l'aide de raisonnement flou.

a- Environnement :

Est l'univers où se situent les robots et les objets.

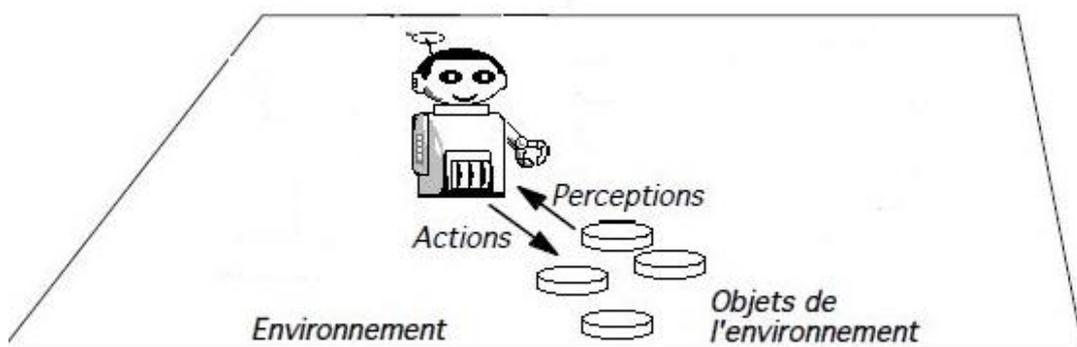


Figure III.1 : Environnement d'un SMA

b- Robot :

Est un agent physique, qui possède des senseurs ou capteurs avec lesquels il perçoit les données, dans notre cas il doit détecter tout objet et mesure combien y a-t-il de distance entre le robot et l'objet. Et des effecteurs pour collecter si l'objet est une cerise, sinon l'éviter.

c- Objets :

- **Obstacle :**
Est un objet qui se situe dans notre environnement, où le robot le considère comme un objet à éviter.

- **Cerise :**
Est un objet qui se situe dans notre environnement, où le robot le considère comme un objet à collecter.

4- Architecture d'un robot :

Comme on a vu précédemment, l'agent est une entité virtuelle ou physique (robot). Dans notre cas est un robot, il fait la perception via des senseurs en mesurant la distance entre lui et l'objet, soit pour l'éviter s'il été un obstacle soit le collecter s'il été une cerise via les effecteurs, comme il est défini dans la figure ci-dessous (Figure III.2).

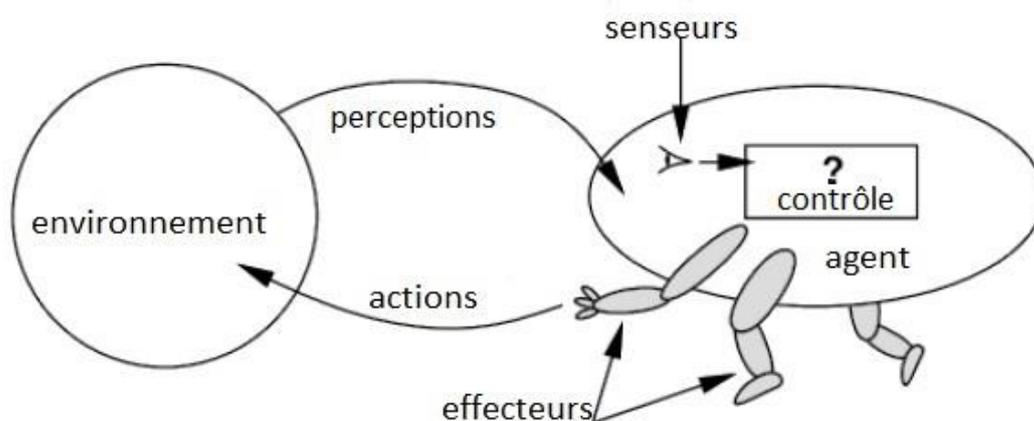


Figure III.2 : Architecture d'un agent

D'après la figure ci-dessus (Figure III.2), le robot fait la perception de son environnement, le contrôle et l'action. Comme on voit que le contrôle est presque en boîte noire.

Comment il interprète la perception en action?

Où intégrer le raisonnement flou ?

Alors, après qu'il fait la perception et il reçoit les données, il les interprète à l'aide d'un raisonnement flou aux actions (Figure III.3).

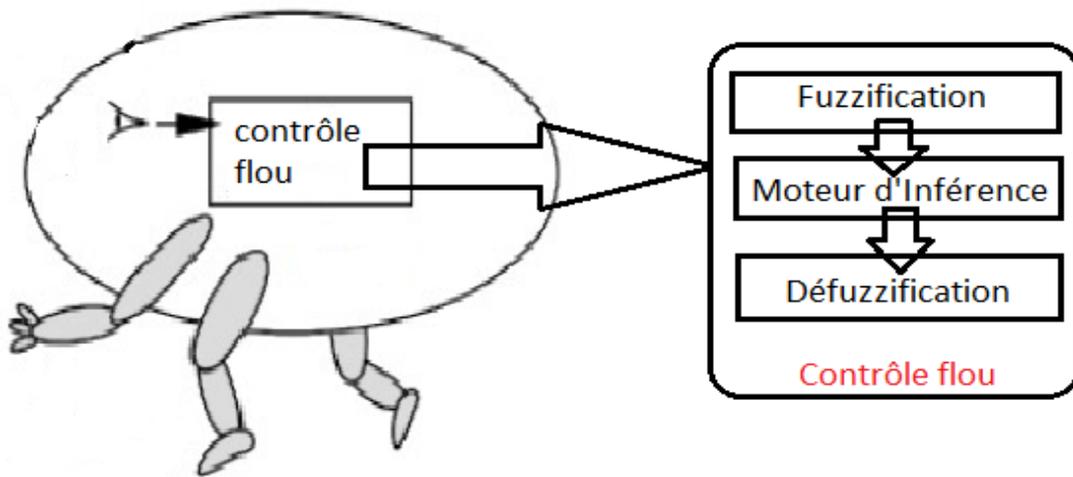


Figure III.3 : Contrôle en raisonnement flou d'un agent

Les actions d'un robot :

- naviguer
- collecter
- Arrêter

5- Système flou :

Comme il est mentionné dans le chapitre précédent, un système flou est constitué de 3 étapes : la fuzzification, moteur d'inférence et défuzzification (Figure III.4).

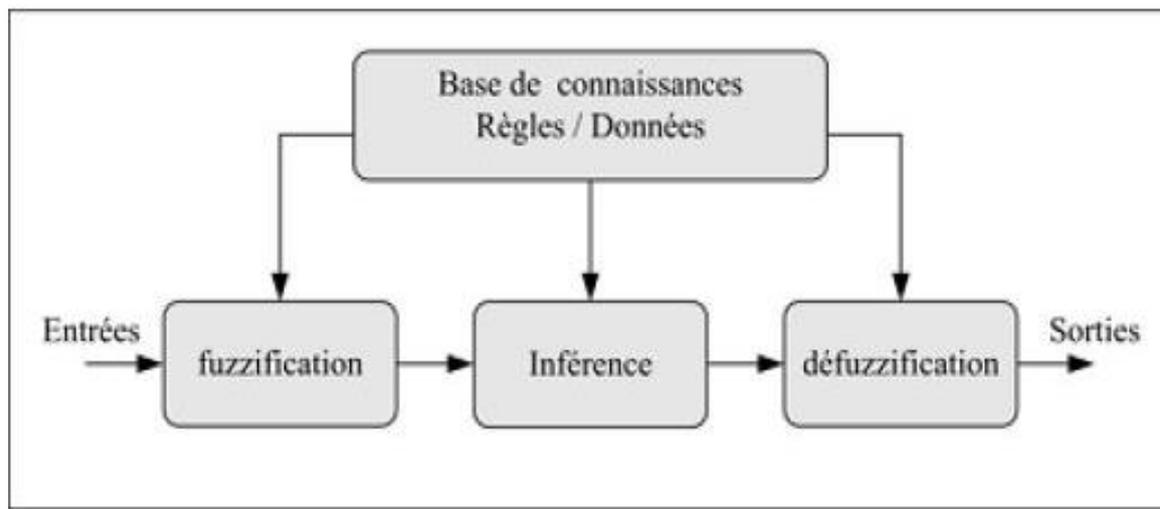


Figure III.4 : Schéma représentant les étapes d'un système flou

5.1- Variables du système flou :

Les variables du système d'entrée et sortie sont numériques et nettes, pas encore floues.

- D'entrée : distance ;
- De sortie : vitesse, angle.

5.2- Variables linguistiques :

Nous pouvons alors définir ce système par les variables linguistiques d'entrée *distance* et de sortie *vitesse, angle*.

Pour la variable floue « distance » on peut deviser l'univers de discours en trois sous-ensembles équitablement qu'on appelle *proche, moyenne* et *loin* on peut choisir des formes trapézoïdales pour ces fonctions d'appartenance.

$x = \text{distance}$

$T(x) = \{\text{proche, moyenne, loin}\}$

$U = [0\text{m}, 100\text{m}]$

Fonction d'appartenance :

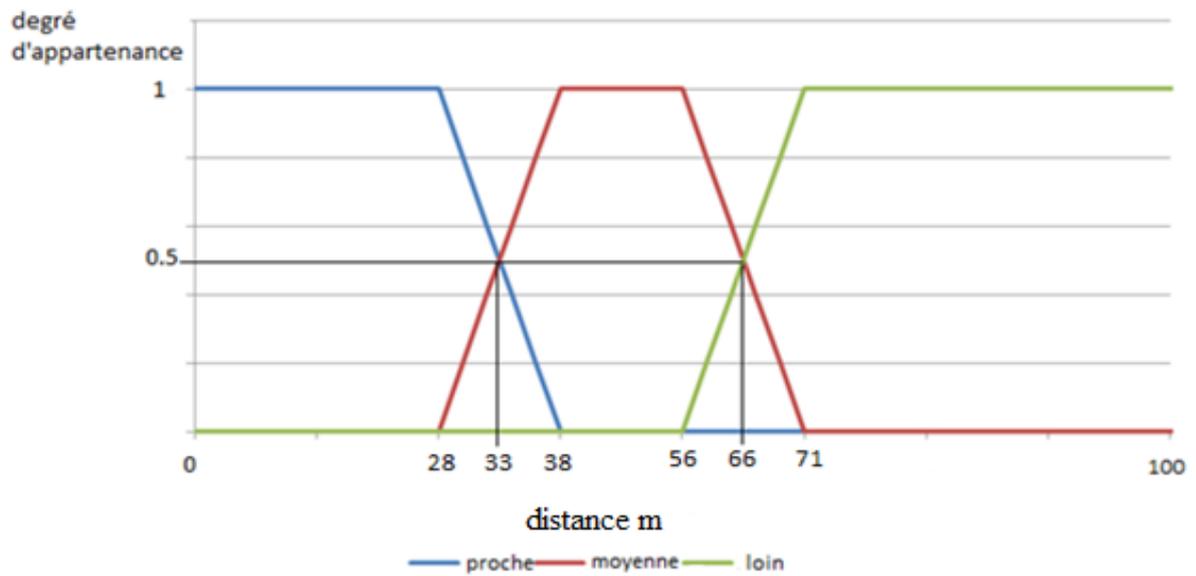


Figure III.5 : Fonction d'appartenance de la variable « distance »

En fait, cette fonction est formée de 3 fonctions pour chacune des valeurs linguistiques proche, moyenne et loin comme présenté sur la figure ci-dessous (Figure III.6).

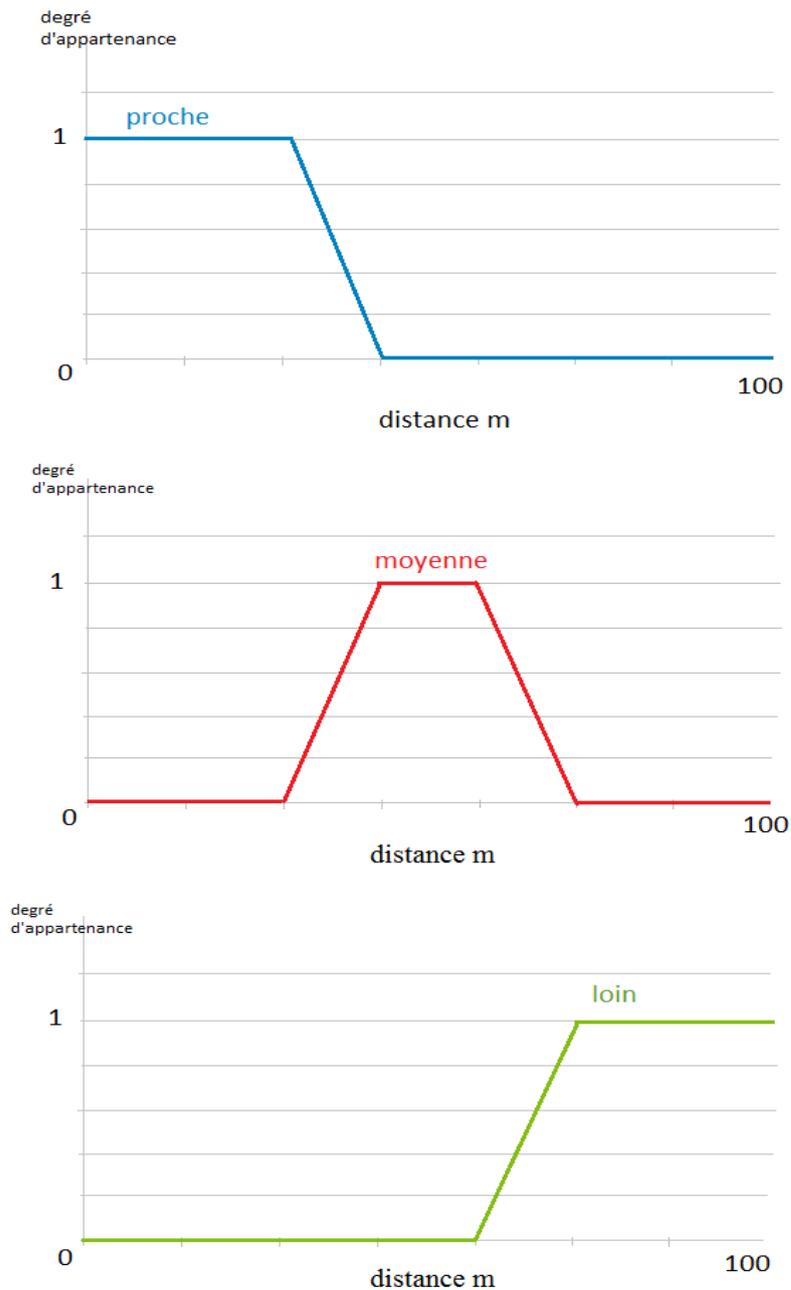


Figure III.6 : Fonctions d'appartenance des valeurs linguistiques

Aussi, pour la variable floue « vitesse » on a défini trois sous-ensembles : *lente*, *moyenne* et *rapide*.

x = vitesse

$T(x) = \{ \text{lente, moyenne, rapide} \}$

$U = [0 \text{ m/s}, 100 \text{ m/s}]$

Fonction d'appartenance :

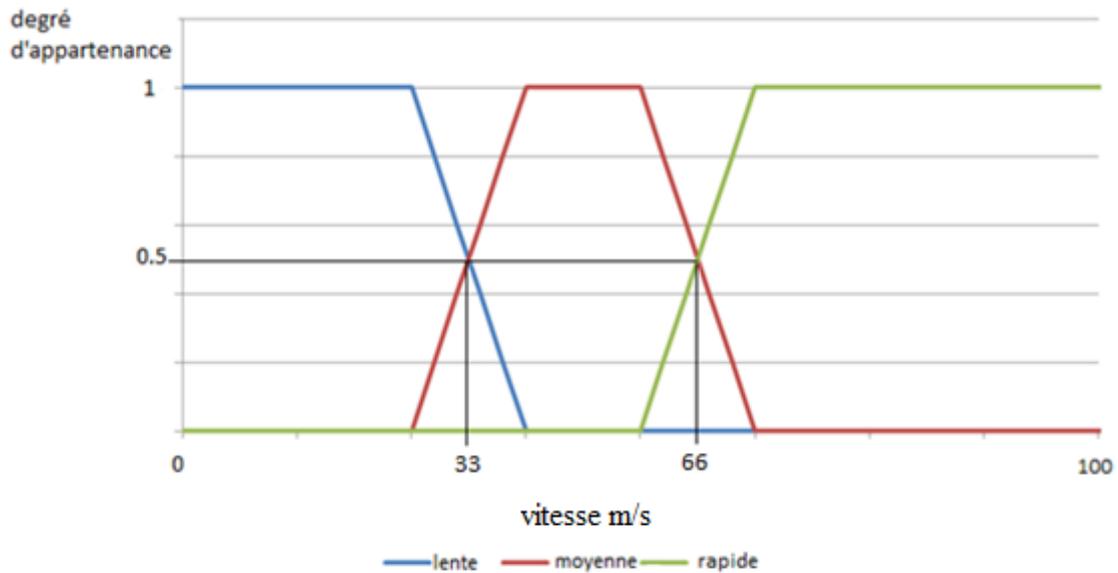


Figure III.7 : Fonction d'appartenance de la variable « vitesse »

La valeur 0 m/s de la variable vitesse appartient à 100% à l'ensemble flou *proche* et 0% aux ensembles flous *moyenne* et *rapide*.

La valeur 33 m/s de cette variable, appartient à 50% à l'ensemble flou *proche*, 50% à l'ensemble *moyenne* et 0% à l'ensemble *rapide*.

Également, nous définissons de même pour la variable floue « angle », trois fonctions d'appartenance pour *petit*, *moyen* et *grand*.

$x = \text{angle}$

$T(x) = \{\text{petit}, \text{moyen}, \text{grand}\}$

$U = [0^\circ, 90^\circ]$

Fonction d'appartenance :

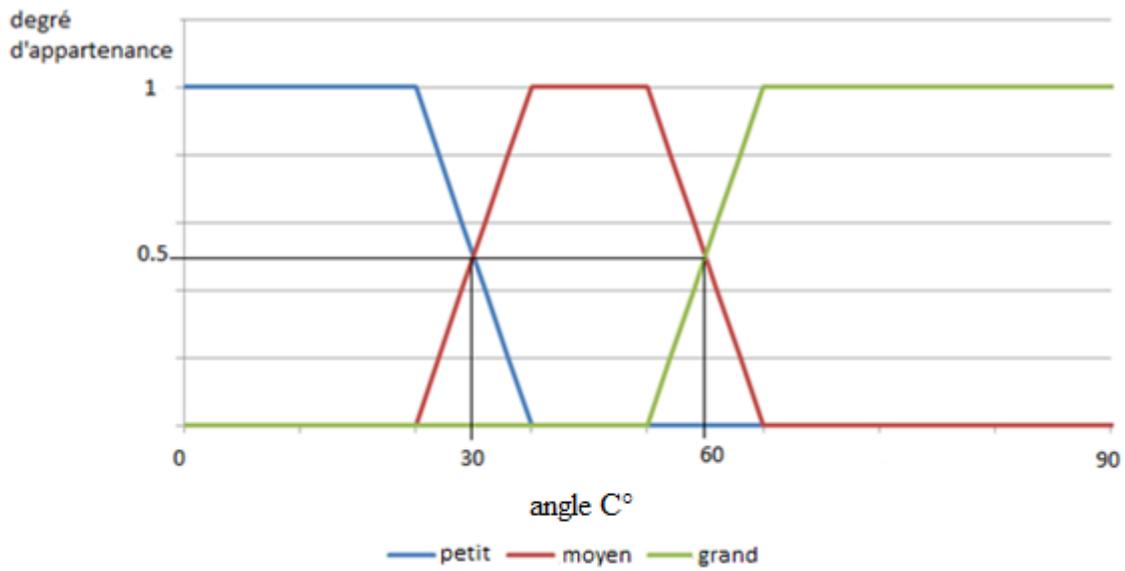


Figure III.8 : Fonction d'appartenance de la variable « angle »

5.3- Fuzzification :

Pour chaque variable d'entrée réelle, on calcule ses degrés d'appartenance aux ensembles flous qui lui sont associés. Par exemple :

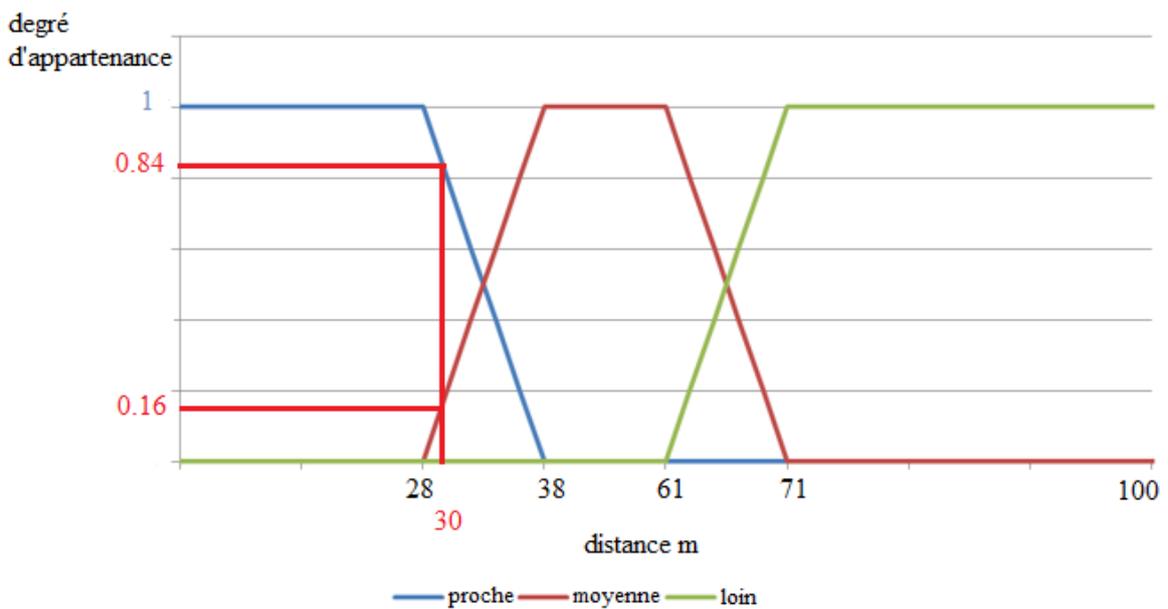


Figure III.9 : Exemple de fuzzification de la valeur d'entrée « distance »

- $\mu_{\text{proche}}(30) = 0.84$
- $\mu_{\text{moyenne}}(30) = 0.16$
- $\mu_{\text{loin}}(30) = 0.0$

Les calculs sont faits via le « Théorème de Thalès ».

5.4- Moteur d'Inférence :

Comme on a déjà mentionné que, Les règles décrivent les relations qui doivent exister entre les variables d'entrée et de sortie afin cette dernière atteigne l'instruction attendue.

Une règle floue comporte une prémisse du genre « *Si 'distance' est 'proche'* » et une déduction floue du type « *alors 'vitesse' est 'lente' et 'angle' est 'grand'* ».

Notre base de règles est composée de :

R1 : Si 'distance' est 'proche' alors 'vitesse' est 'lente' et 'angle' est 'grand'

R2 : Si 'distance' est 'moyenne' alors 'vitesse' est 'moyenne' et 'angle' est 'moyen'

R3 : Si 'distance' est 'loin' alors 'vitesse' est 'rapide' et 'angle' est 'petit'

- La clause « *distance est proche* » est l'antécédent de la règle.
- Les conclusions « *vitesse est lente* » et « *angle est grand* » sont les conséquentes de la règle.

Le raisonnement est claire si on est proche de l'obstacle on réduit la vitesse et on augmente l'angle de rotation, pour éviter la collision.

5.4.1- Implication :

R1 : Si 'distance' est 'proche' (0.84) alors

'vitesse' est 'lente' (0.84)

et

'angle' est 'grand' (0.84)

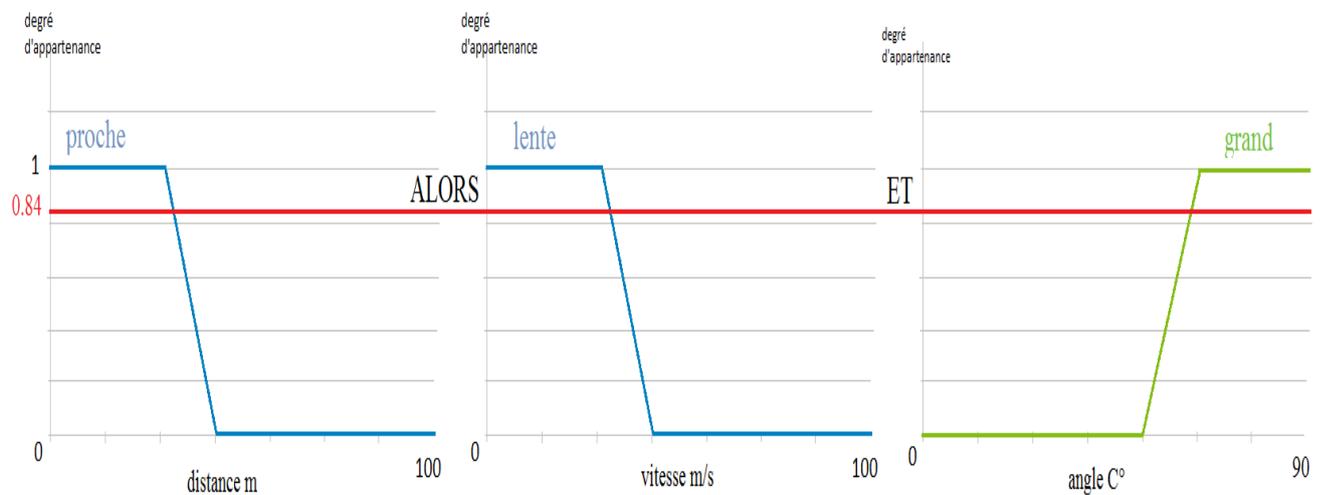


Figure III.10 : Exemple d'implication de la règle 1

R2 : Si 'distance' est 'moyenne' (0.16) alors

'vitesse' est 'moyenne' (0.16)

et

'angle' est 'moyen' (0.16)

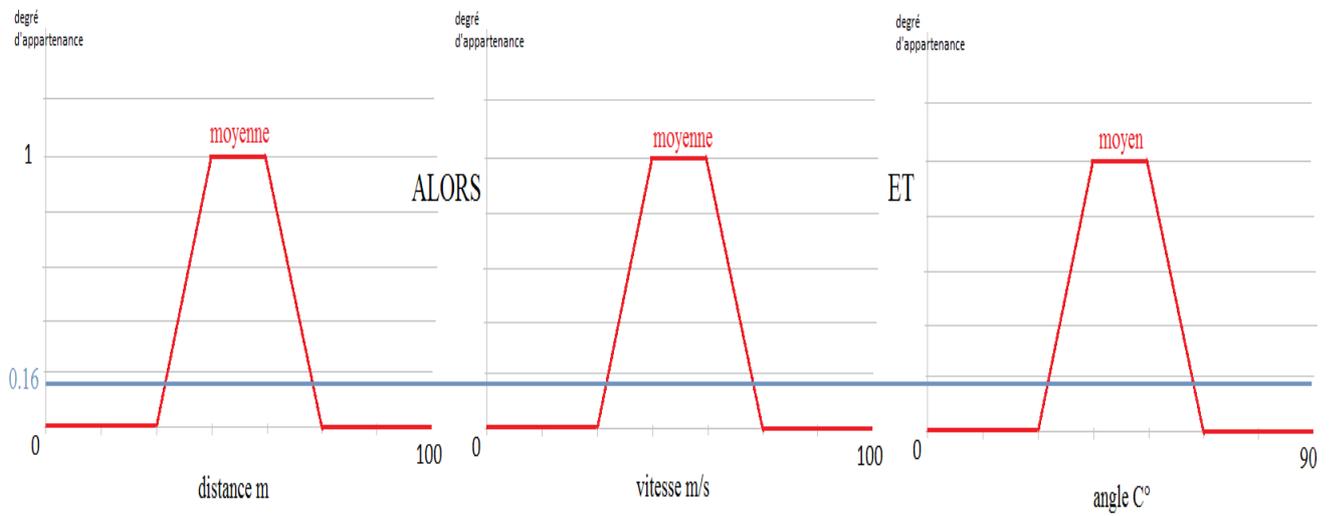


Figure III.11 : Exemple d'implication de la règle 2

R3 : Si 'distance ' est 'loin' (0.0) alors

'vitesse ' est 'rapide' (0.0)

et

'angle ' est 'petit' (0.0)

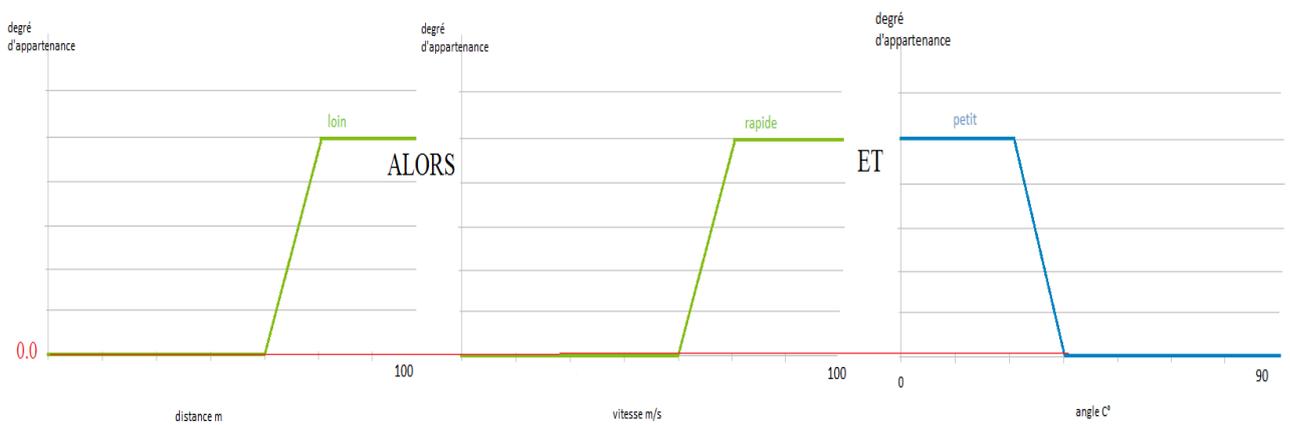


Figure III.12 : Exemple d'implication de la règle 3

Plusieurs règles peuvent être activées en même temps, veut dire que chacune de leurs prémisses possède un degré d'appartenance non nul.

5.4.2- Agrégation :

L'agrégation de ces règles se fait par l'opérateur max, qui doit mener à une seule valeur de la variable de sortie, comme si les règles étaient liées par l'opérateur OU.

Si les règles 1 et 2 sont activées en même temps, la fonction d'appartenance de la variable de sortie est caractérisée par la surface pleine ci-dessous.

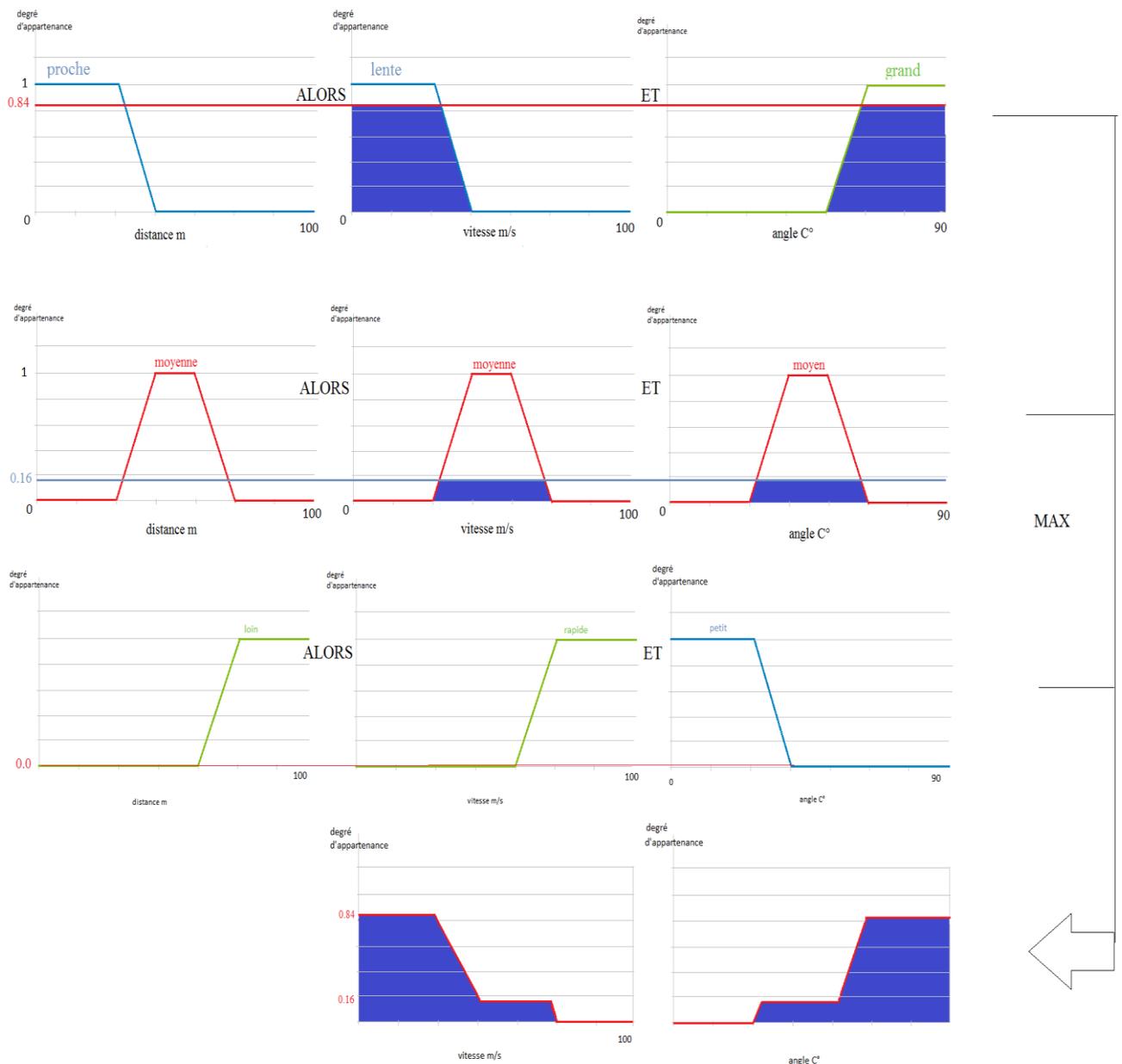


Figure III.13 : Exemple d'agrégation des règles floues

5.5- Défuzzification :

Au cours de la fuzzification, une variable réelle devienne floue par contre dans l'étape de défuzzification, on réalise l'opération inverse, pour obtenir une valeur réelle de la sortie à partir des surfaces obtenues dans l'étape d'inférence.

Il existe plusieurs méthodes de défuzzification, dont la plus utilisée est celle du centre de gravité (COG), défini comme suit :

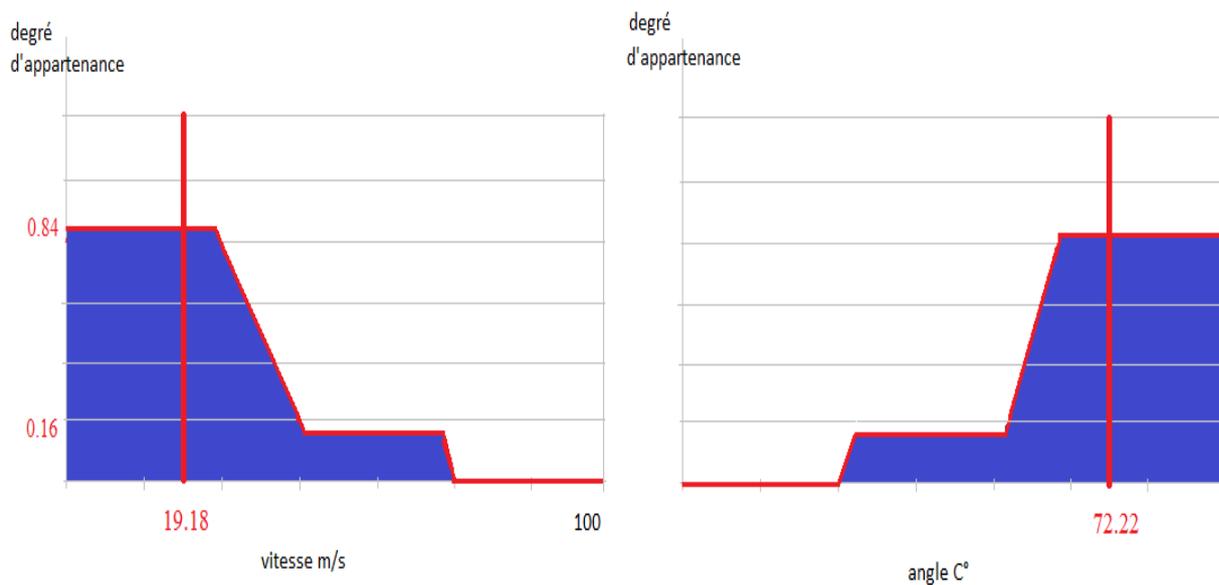


Figure III.14 : Défuzzification avec la méthode centre de gravité (COG)

Décision : vitesse = 19.18 m/s

angle = 72.22 C°

6- Algorithme 1 :

Voici un algorithme en pseudo code d'un fonctionnement de robot :

Variables Entrées :

Variable distance : réel

Variables Sorties :

Variables vitesse, angle : réels *//vitesse de déplacement et angle de rotation*

DEBUT

 TANT QUE (vrai) FAIRE

 distance ← perception ()

 fuzzification (distance)

 inference ()

 (vitesse,angle) ← defuzzification ()

 move (vitesse, angle)

 SI (detecte_cerise ()) ALORS collecter_cerise ()

 FINSI

FIN TANT QUE

FIN

Organigramme :

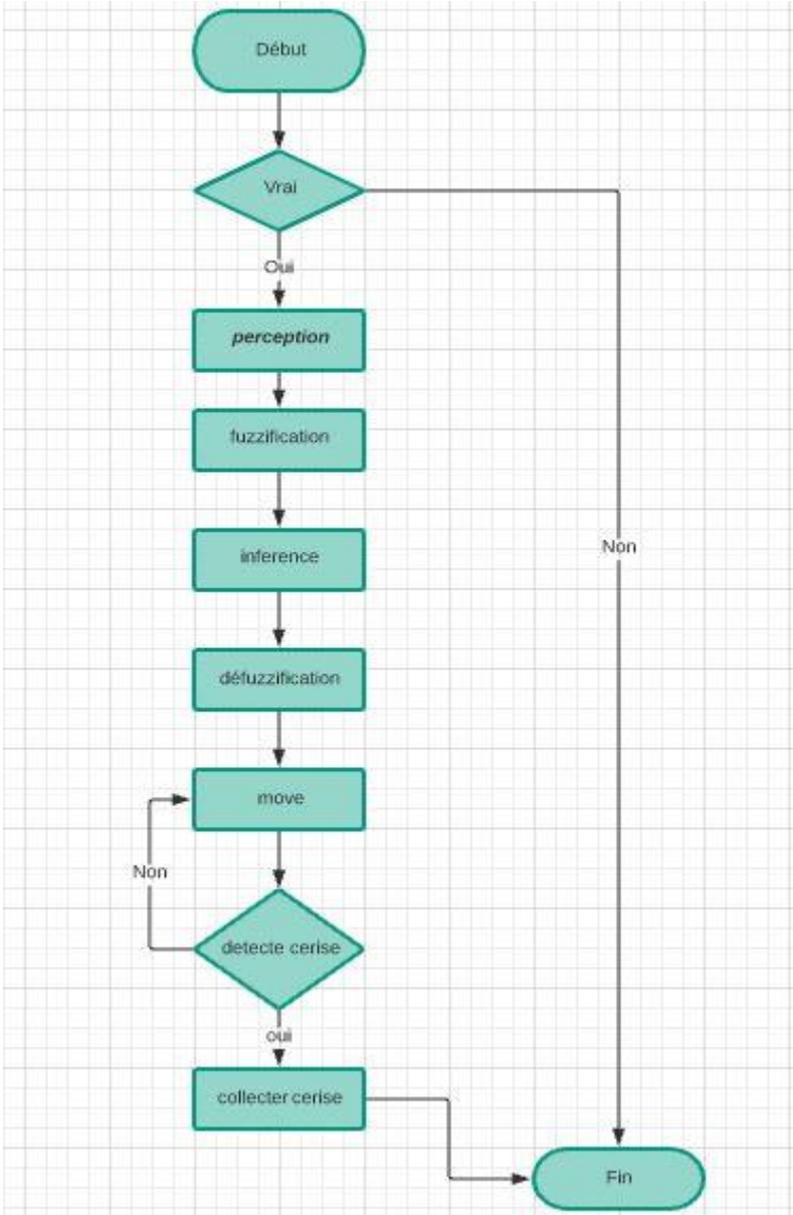


Figure III.15 : Organigramme d’algorithme de fonctionnement d’un robot

7- Résumé:

Voici un aperçu récapitulant le système flou :

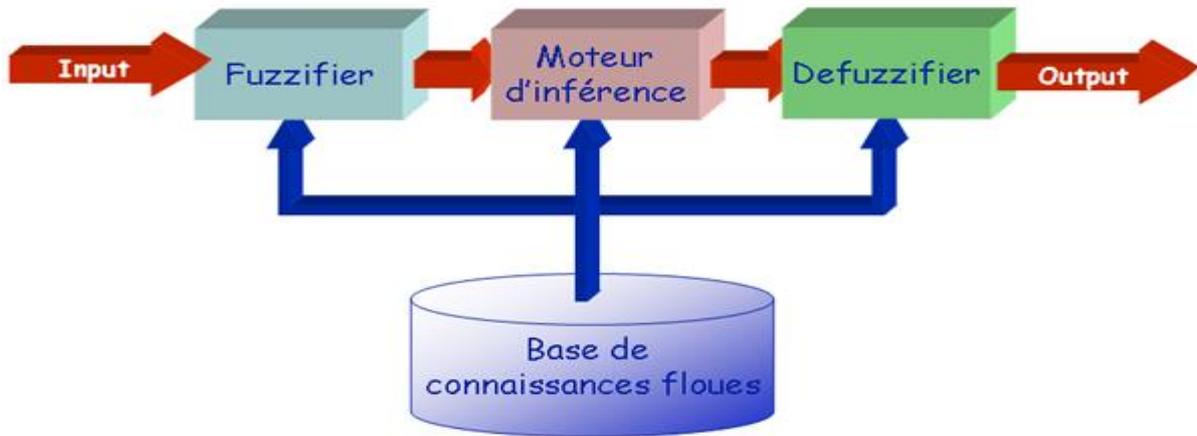


Figure III.16 : Aperçu récapitulant le système flou

Comme illustré sur la figure ci-dessus (Figure III.16) :

- L'**input** est la « distance » notée 30 ;
- Le **fuzzifier** correspond aux trois valeurs linguistiques « proche », « moyenne » et « loin » ;
- Le **Moteur d'inférence** est formé du choix des opérateurs flous ;
- La **base de connaissances floues** est l'ensemble des règles floues ;
- Le **défuzzifier** est la phase où entre le choix de la méthode de déffuzification ;
- L'**output** représente la décision finale : « la vitesse est 19.18 m/s » ;
« l'angle est 72.22 C° ».

On peut aussi mieux comprendre de la figure ci-dessous (Figure III.17), qui donne une vue globale sur la logique floue.

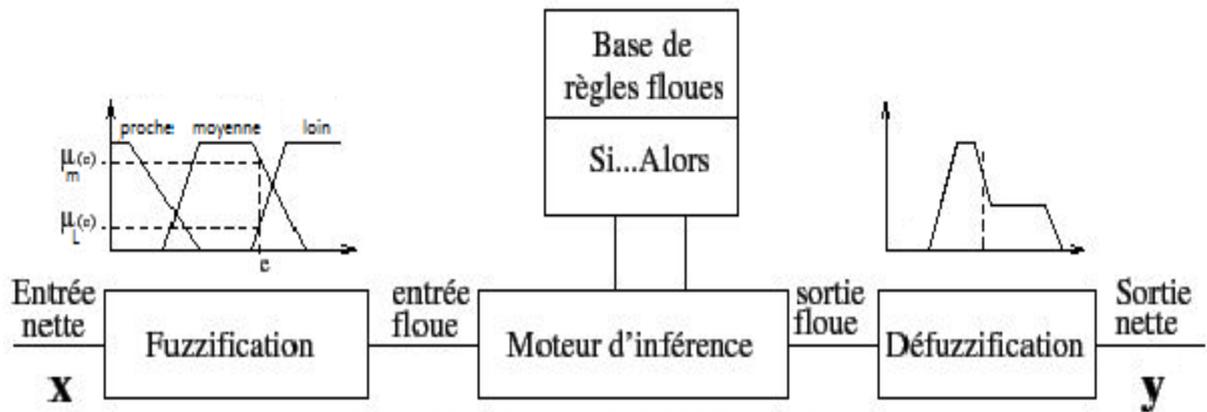


Figure III.17 : Schéma récapitulant un système flou

Afin de comparer la logique classique et la logique floue, un système de navigation à base de la logique classique est développé pour voir l'effet de la logique floue sur le comportement de robot.

8- Algorithme 2 :

Voici un algorithme en pseudo code à base de logique classique:

Variables Entrée :

Variable distance : réel

Variables Sortie :

Variables vitesse, angle : réels //vitesse de déplacement et angle de rotation

DEBUT

vitesse \leftarrow 10

angle \leftarrow 90

TANT QUE (vrai) FAIRE

distance \leftarrow perception()

SI (! obstacle_detecte (distance)) ALORS avancer (vitesse)

SINON tourner (angle)

FINSI

SI (detecte_cerise()) ALORS collecter_cerise ()

FINSI

FIN TANT QUE

FIN

Conclusion :

Ce chapitre a donné une vision sur notre travail, et a donné l'aspect conceptuel de l'application à travers les différentes étapes de logique floue comme a facilité la compréhension de notre système, qui entame vers l'activité d'implémentation de notre application qui sera détaillée dans le prochain chapitre.

Introduction :

Ce chapitre représente la dernière partie de ce rapport, il traite la phase de l'implémentation de notre application. Comprenant la description des outils technologiques utilisés et un aperçu sur le travail réalisé.

1- Environnement de travail :

L'environnement de travail est constitué par deux parties nommées environnement matériel et environnement logiciel.

1.1- Environnement matériel :

Le travail a été réalisé dans un environnement matériel comme suit :

- Windows 7;
- Machine TOSHIBA;
- Processeurs Intel® Core™ i3-4005U (3M Cache, 1.70 GHz) ;
- Mémoire RAM 4,00 Go ;
- 500 Go Capacité de Disque dure.

1.2- Environnement logiciel :

L'environnement logiciel repose sur les éléments suivants :

- **NetBeans IDE:**

NetBeans est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun Microsystems en juin 2000. En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web).

NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS. NetBeans est lui-même développé en Java.

- **Java :**

Java est un langage de programmation orienté objet et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications.

- **Simbad :**

Le projet Simbad a démarré en 2005 et a été initialement développé par le Dr Louis Hugues. Est un simulateur de robot Java 3D pour des buts scientifiques et éducatifs. Il est principalement destiné aux chercheurs / programmeurs qui souhaitent une base simple pour étudier l'intelligence artificielle située, l'apprentissage automatique et, plus généralement, les algorithmes d'IA, dans le contexte de la robotique autonome et des agents autonomes. Simbad est un logiciel multiplateforme open source. Il est écrit en langage Java.

- **Java3D :**

Java 3D est une interface de programmation (Application Programming Interface ou API) pour la plateforme Java visant la synthèse d'image 3D basée sur les graphes de scène. Aussi, est une bibliothèque de classes d'extension Java destinée à créer des scènes 3D en réalité virtuelle (avec utilisation de formes complexes, d'éclairages, de textures, d'animations, de sons,...).

- **JavaFX :**

JavaFX est un Framework et une bibliothèque d'interface utilisateur créé à l'origine par Sun Microsystems puis développé par Oracle, qui permet aux développeurs Java de créer une interface graphique pour des applications de bureau, des applications internet riches et des applications Smartphones et tablettes tactiles.

2- Développement de l'application :

Dans cette partie, nous allons exposer les différentes phases de notre application en mentionnant des captures d'écrans.

2.1- Interface principale :

C'est la première interface (Figure IV.1) qui apparaît à l'utilisateur, et lui permet de saisir la configuration de la simulation :

- Type d'environnement, il peut être en fonction de nombre d'obstacles qui sont situés aléatoirement ou en fonction d'un labyrinthe (*mase en anglais*, Figure IV.3) ;
- Nombre de cerises, objets à collecter sont aussi situés dans l'environnement aléatoirement ;
- Nombre de robots placés aléatoirement ;
- Type de raisonnement, flou ou classique.



Figure IV.1 : Fenêtre principale

Si on clique sur le bouton **Lancer**, apparaîtra la simulation avec l'un des types : flou ou classique (Figure IV.2).

Le bouton **Lancer les deux**, lance une simulation pour un raisonnement flou et une pour un raisonnement classique, avec le même environnement (Figure IV.4) pour une comparaison visuelle.

2.2- Interface simulateur :

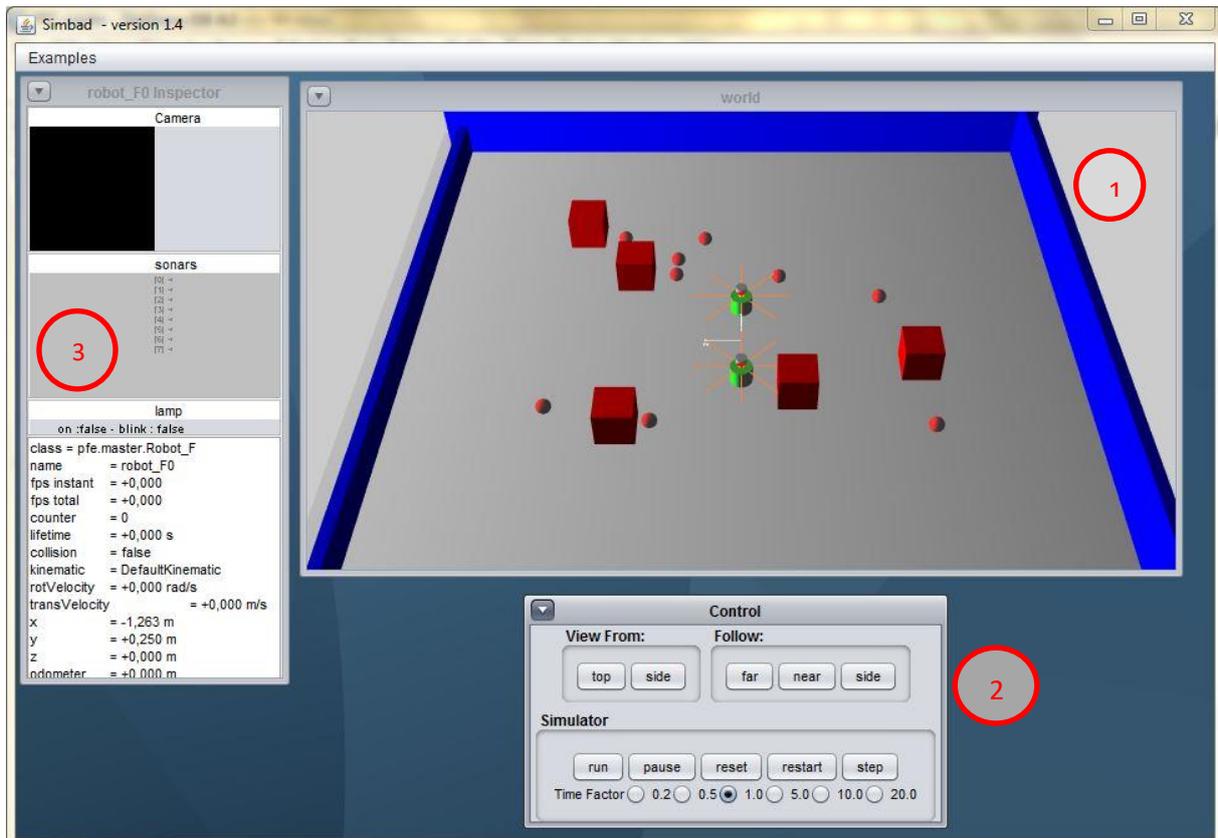


Figure IV.2 : fenêtre de simulateur

Cette interface contient trois composantes :

- ① la présentation de l'environnement, robots et objets.
- ② Un panneau de contrôle permettant de choisir l'angle de vue, vitesse de simulation et des différentes fonction (demarrer **:run**, suspendre **:pause**, réinitialiser **:reset ...**),
- ③ Des informations concernant le robot (camera, sensor, position, nom ...).

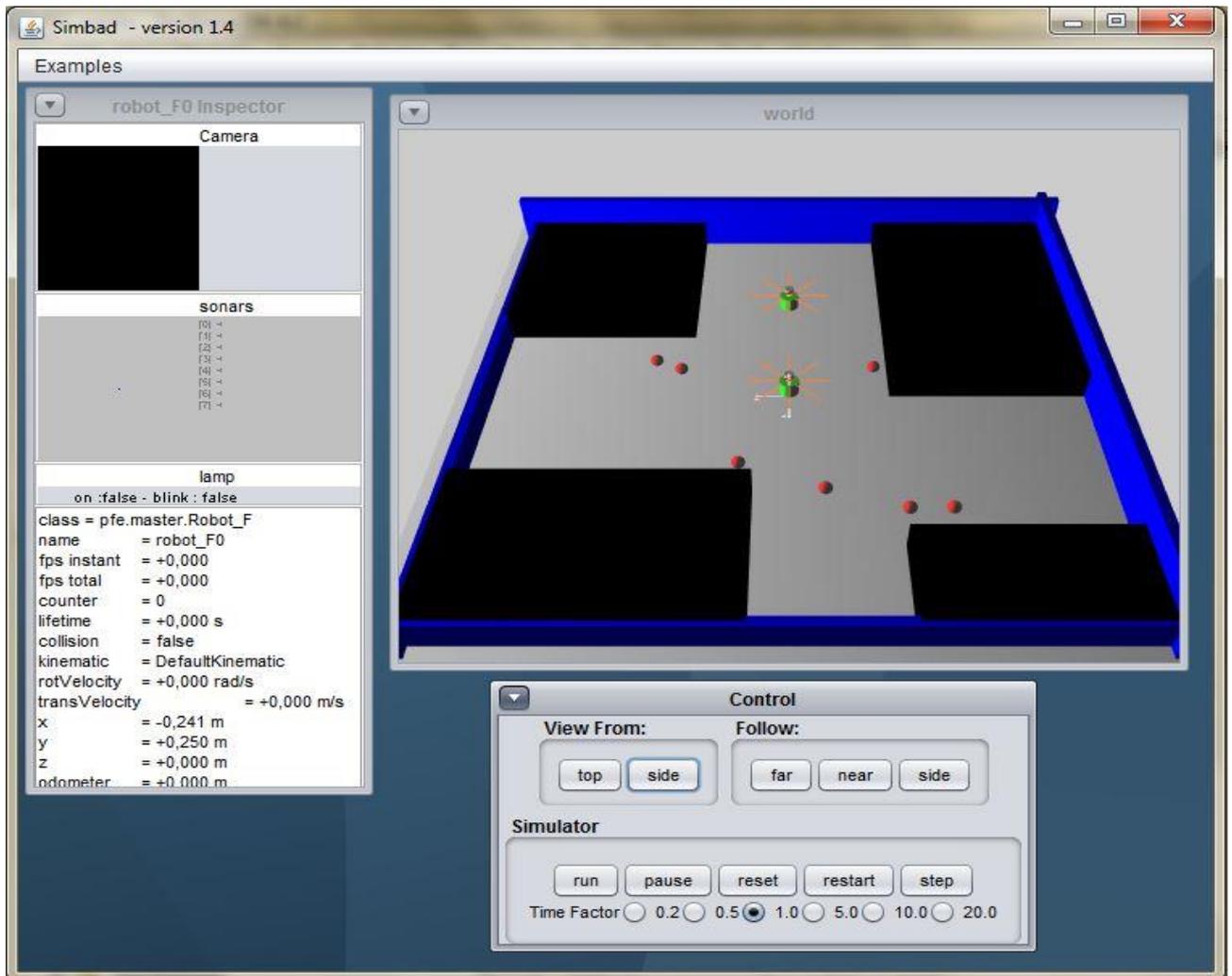


Figure IV.3 : Simulateur en fonction d'un labyrinthe

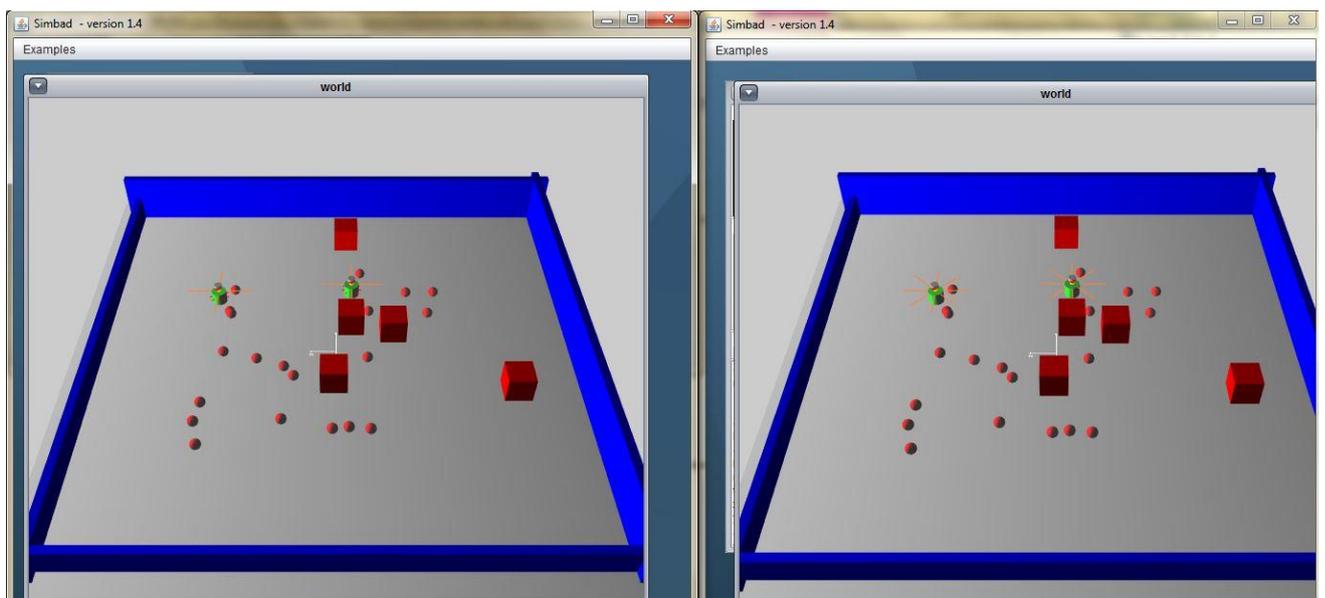


Figure IV.4 : Simulateur en cas classique et flou

La figure ci-dessus (Figure IV.4) représente une simulation de deux raisonnements classique et flou successivement, d'angle de vue « side », avec la même configuration et la même position d'objets et robots, qui permettra d'avoir une comparaison visuel de comportement en niveau du système de navigation de robot.

Comme, c'est une simulation on a pris la plage de distance en (10 m), vitesse en (30 m/s) et angle reste toujours (90 C °).

Ci-dessous (Figure IV.5), La traçabilité de déroulement de notre programme lors de la navigation flou.

La perception, fuzzification, inférence et défuzzification qui donne finalement la vitesse de déplacement et l'angle de rotation de robot.

```
Mesure de perceprion 6.4207300165218

fuzzification : Distance :
  proche : 0.0
  moyen  : 0.7459366501448672
  loin   : 0.2540633498551328

Base de regles :

Si la distance est proche ( 0.0 )
  Alors la vitesse est lente ( 0.0 )
  et l'angle est grand ( 0.0 )
  |
Si la distance est moyenne ( 0.7459366501448672 )
  Alors la vitesse est moyenne ( 0.7459366501448672 )
  et l'angle est moyen ( 0.7459366501448672 )

Si la distance est loin ( 0.2540633498551328 )
  Alors la vitesse est rapide ( 0.2540633498551328 )
  et l'angle est petit ( 0.2540633498551328 )

defuzzification : vitesse 17.150724736422475 angle : 37.42870426679153
```

Figure IV.5 : Traçabilité de raisonnement flou

On peut schématiser la fonction d'appartenance de sortie après l'agrégation des règles activées comme suite :

distance = 6.42 m :

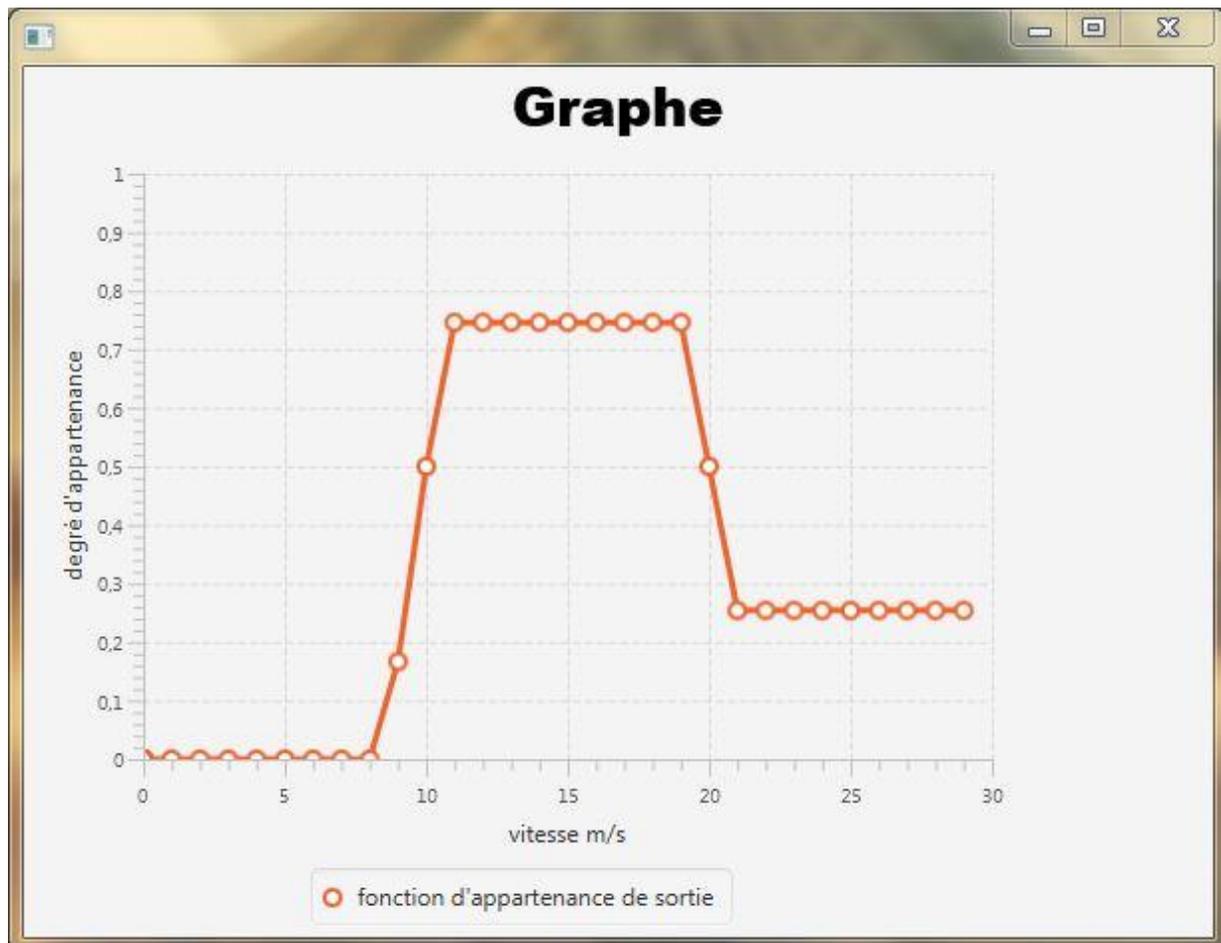


Figure IV.6 : schéma de surface pleine après l'agrégation « vitesse »

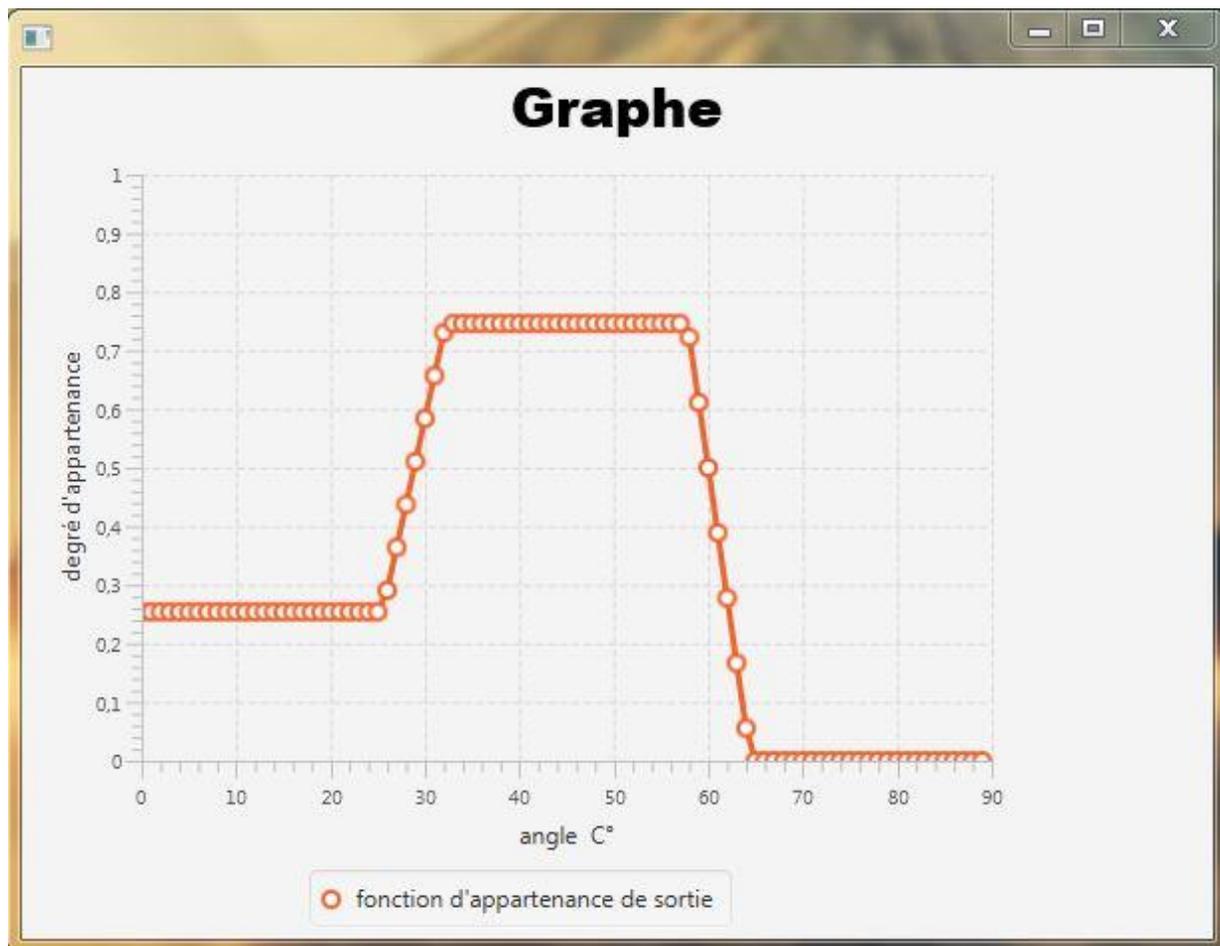


Figure IV.7 : schéma de surface pleine après l'agrégation « angle »

Après la défuzzification en COG :

vitesse = 17.15 m/s ; angle = 37.42 C°

Conclusion :

A travers ce chapitre, nous avons présenté la réalisation de l'application en représentant quelques interfaces graphiques que nous avons trouvées les plus importantes avec une brève explication comment nous avons planifié notre projet.

Conclusion générale

Le travail présenté dans ce mémoire porte sur la navigation floue, basée sur le comportement d'un robot mobile dans un environnement dynamique et complexe.

Dans ce dernier, l'environnement considéré comprend des obstacles. La détection de ces obstacles est assurée par un module de perception. Ce dernier transmet les situations des capteurs au système flou. Celui-ci génère les deux sorties (vitesse de déplacement et angle de rotation) permettant de dévier le robot des obstacles rencontrés, plus une collecte en cas d'objets prédéfinis (cerises).

Le rapport mentionne toutes les étapes traversées pour arriver au résultat attendu. La logique floue qui a été le moyen pour atteindre ce résultat, elle permet de faire un lien entre données numériques et linguistiques. C'est une théorie qui permet la formalisation des imperfections dues aux connaissances inexacts d'un système et à la description du comportement du système par des mots.

Ce travail nous a été une formation très enrichissante, puisqu'il a permis de découvrir des techniques intéressantes, et nous a permis également de se confronter à plusieurs contraintes à la fois : contraintes de temps, contraintes d'expérience et de technologie. De plus, ce projet nous a permis d'approfondir nos connaissances dans les bonnes pratiques de l'ingénierie génie informatique.

BIBLIOGRAPHIE:

- [1]: *Russell S., Norvig P. Intelligence artificielle. Pearson Education 2^e édition. 2006.*
- [2]: *Benhamza Karima, “ Conception d’un système multi-agents adaptatif pour la résolution de problème”, Thèse de doctorat en sciences, sous la direction de Seridi Hamid, Annaba, université Badji Mokhtar, 2016, 145 p.*
- [3]: *Ferber, Jacques, 1995, Les Systèmes Multi Agents: vers une intelligence collective, InterEditions, 499 p.*
- [4]: *Sameh Triki, “ Système Multi-Agent Ambient pour faciliter l’autonomie et l’accessibilité aux espaces publics des personnes ayant des déficiences cognitives”, Thèse de doctorat en sciences, sous la direction de Chihab Hanachi, Toulouse, Université Toulouse 3 Paul Sabatier, 2018, 138 p.*
- [5]: *Nachet Bakhta, “ Modèle Multi-Agent pour la conception de systèmes d’aide à la décision collective ”, Thèse de doctorat en informatique, sous la direction d’Aalaa A., Oran, Université d’Oran, 2014, 155 p.*
- [6]: *Imane Boussebough, “ Les systèmes Multi-Agent dynamiquement adaptables ”, Thèse de doctorat en sciences, sous la direction de Zaidi Sahnoun, Constantine, Université Mentouri Constantine, 2011, 134 p.*
- [7]: *M. Wooldridge, N. R. Jennings., D.Kinny, The Gaia Methodology for Agent- Oriented Analysis and Design, Journal of Autonomous Agents and Multi-Agent Systems, vol. 3, n° 3, p. 285-312, Kluwer Academic Publishers, 2000.*
- [8]: *Aid Lahcene, “ Modélisation et simulation du confort dans un bâtiment intelligent par le formalisme DEVS”, Thèse de doctorat en sciences, sous la direction de Zaoui Lynda, Oran, Université des Sciences et de la Technologie d’Oran Mohamed Boudiaf, 2016, 116 p.*
- [9]: *Khammar Fatma, “ Contribution à l’Etude de la Commande d’une Machine Asynchrone : Approches Intelligentes”, Thèse de doctorat en sciences, sous la direction de Debbache Nasr Eddine, Annaba, Université Badji Mokhtar, 2017, 100 p.*
- [10]: *Houcine Belouaar, “ Modélisation d’une approche basée agent et logique floue pour la qualité des services Web ”, Thèse de doctorat en sciences, sous la direction de Okba Kazar, Biskra, Université Mohamed Khider, 2019, 128 p.*
- [11]: *Boumertit Sofiane, “ Commande par logique floue d’un circuit de combustion d’une chaudière de type compact ”, mémoire de master en électronique, sous la direction de Lehouche Hocine & Bennai Laabid, Bejaia, Université Abderrahmane Mira, 2012, 119 p.*
- [12]: *Bonilla Huerta, Edmuno, “ Logique floue et algorithmes génétiques pour le pré-traitement de données de biopuces et la sélection de gènes”, Thèse de doctorat en Informatique, sous la direction de Jin-Kao Hao, France ,Université d’Angers, 2008,131p.*

- [13]: *Bouchon-Meunier, Bernadette, “ la logique floue et ses applications ”, France, Vuibert, 1995, 272 p.*
- [14]: *Beloufa Fayssal, “Conception d’un classifieur à base des règles floues ”, Thèse de doctorat en sciences, sous la direction de Mohammed Amine Chikh, Tlemcen, université Abou-Bekr Belkaid, 2016, 115 p.*
- [15]: *Talbi Nesrine, “ Conception des Systèmes d’Inférence Floue par des Approches Hybrides : Application pour la Commande et la Modélisation des Systèmes Non linéaires ”, Thèse de doctorat en sciences, sous la direction de Khaled BelarbiI, Constantine, Université de Constantine 1, 2014, 134 p.*
- [16]: *Bouchon-Meunier, Bernadette ; Marsala, Christophe, “ Logique floue, principes, aide à la la décision ”, France, HERMES Lavoisier, 2003, 252p.*
- [17]: *Martaj, Nadia ; Mokhtari, Mohand, “Apprendre et maîtriser LabVIEW par ses applications”, Paris, Springer Libri, 2014 ,898 p.*