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

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ DES MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Génie Informatique

Par :

MEKKI Habib et TAIBI Kada

Sur le thème

Simulation de la Navigation d'un robot mobile dans un environnement inconnu en utilisant les réseaux neuro-flous

---

Soutenu publiquement le 25 /06/ 2023 à Tiaret devant le jury composé de :

Mr BENGHENI Abdelmalek	MCB	Université Ibn Khaldoun Tiaret	Président
Mr MEBAREK Bendaoud	Prof	Université Ibn Khaldoun Tiaret	Encadreur
Mr MOSTEFAOUI Kadda	MAA	Université Ibn Khaldoun Tiaret	Examinateur

2022-2023

Remerciment

Tout d'abord, nous voudrions remercier **« ALLAH »** Tout-Puissant et Miséricordieux. Créateur de tout l'univers qui nous a donné la santé, la force, le courage et la volonté de mener à bien cet humble travail de recherche.

Nous voudrions remercier toutes les personnes ayant soutenu et encadré ce travail :

En premier lieu, nous remercions sincèrement et très chaleureusement, nos encadreurs, Mr. **MEBAREK Bendaoud**, Professeur à l'université Ibn Khaldoune de Tiaret, pour tout, que ce soit en raison de leur disponibilité ou de leur principale contribution à la direction des travaux de recherche et de leurs idées originales et éclairantes qui ont influencé le contenu de ce mémoire de Master et ouvert la voie à son achèvement.

Nous tenons également à remercier Mr. **BENGHENI Abdelmalek**, Maitre de Conférences classe B à l'université Ibn Khaldoune de Tiaret, pour l'intérêt qu'elle a porté à ce travail en nous honorant par sa présence dans le jury.

Nous tenons également à remercier Mr. **MOSTEFAOUI Kadda**, Maitre d'Assistant classe A à l'université Ibn Khaldoune de Tiaret, pour l'intérêt qu'elle a porté à ce travail en nous honorant par sa présence dans le jury.

Enfin, nos sincères remerciements vont à tous ceux qui ont contribué à la réalisation de ce travail, de près ou de loin.

Dedicase

Je dédie ce Travail à mes parents et à mes frères et sœurs.

Je voudrais aussi remercier ma chère épouse et mes

Enfants Tinhinane, Mokhtar Abdelhamid et Mohamed

Et à tous les personnels de

CFPA Tourak Mohamed de Sougueur.

Taibi Kadda

Dedicase

Je dédie ce Travail

à mes parents et à mes frères et sœurs.

Je voudrais aussi remercier

ma chère épouse et mon

Enfant Mossa

MEKKI Habib

Sommaire

Liste des Tableaux

Liste des Figures

Dans le cadre de la navigation d'un robot mobile dans son environnement la capacité d'aller d'une position initiale A à une position finale B de manière autonome, le problème général de navigation consiste à déterminer pour le robot un mouvement lui permettant de se déplacer entre ces données tout en respectant un certain nombre de contraintes et de critères. Ceux-ci découlent de plusieurs facteurs de natures diverses et dépendent généralement des caractéristiques du robot, de l'environnement et du type de tâche à exécuter. En l'occurrence, les contraintes relatives au robot concernant sa géométrie, sa cinématique et sa dynamique, les contraintes émanant de l'environnement concernant essentiellement la non-collision avec les obstacles encombrant l'environnement et la prise en compte d'interactions de contact avec le robot. L'évitement d'obstacles dépend de la géométrie de l'environnement et est commun à toutes les tâches robotiques

Notre mémoire est organisée de la manière suivante:

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

Dans le deuxième chapitre, on présente l intelligence artificielle et machine learning et des définitions global.

Dans le troisième chapitre, on présente les systèmes flous. Nous commençons par énoncer les fondements théoriques des sous-ensembles flous et de la logique floue,

Dans le quatrième chapitre, nous décrivons la modélisation de la commande et l'architecture de robot mobile, présenter comment le robot déplacer dansl'environnement et éviter les obstacles

Dans la cinquième chapitre on présenter la simulation et l'implémentation, l'environnement de développement de notre application et les différents tests.

1. Aperçu historique:

Les premiers robots mobiles, c'est-à-dire des robots capables de se déplacer dans un environnement donné, bien qu'à distance, n'ont été développés au MIT que quelques années plus tard, en 1968.

Entre 1966 et 1972, l'Institut de recherche de Stanford aux États-Unis a développé le robot « Shakey », qui exécute une commande fixe (pousser un bloc d'une plate-forme) grâce à sa propre planification des tâches, et est ainsi considéré comme le premier robot mobile autonome au monde. Les robots mobiles autonomes se caractérisent par le fait qu'ils peuvent se déplacer et agir de manière autonome dans leur environnement, et il peut y avoir différentes gradations dans le degré d'autonomie. Par exemple, un sous-domaine distinct de la robotique mobile autonome est la conduite autonome, c'est-à-dire la participation sans conducteur de véhicules automobiles à la circulation routière. Les tâches que la conduite autonome doit maîtriser en tant que cas d'utilisation sont très spéciales et complexes et ne peuvent donc pas être comparées à d'autres scénarios de robotique mobile autonome [1]**.**

2. Définition de robot mobile

Un appareil automatique qui peut effectuer des fonctions normalement effectués par des humains. [2]

Appareil effectuant, grâce à un système de commande automatique à base de micro-processeur, une tâche précise pour laquelle il a été conçu dans le domaine industriel, scientifique ou domestique [3].

Machine automatique dotée d'une mémoire et d'un programme, capable de se substituer à l'homme pour effectuer certains travaux [4].

3. Différents domaines d'application de la robotique actuelle

La robotique est un domaine en plein essor depuis quelques années. Les évolutions technologiques, dépassant sans cesse nos espérances, permettent maintenant de réaliser des solutions technologiques s'adaptant au moindre problème.

La robotique actuelle trouve des applications dans différents domaines (liste non exhaustive):

- la robotique industrielle,
- la robotique domestique,
- la robotique médicale,
- la robotique militaire,

Les travaux en robotique ont pour but de concevoir et de construire des machines capables d'évoluer et d'interagir avec un environnement physique.

4. Types des robots de robot mobile

La classification des robots est faite de nos jours dans un objectif de donner une idée de la présence actuelle de la robotique et de la portée de ce domaine dans les années à venir, ainsi nous pouvons convenir qu'il existe trois grandes catégories de robots.

4.1 Les robots humanoïdes :

C'est sans doute la catégorie de robot la plus connue, ceci est dû en grande partie à la promotion faite pas la science-fiction, elle regroupe ainsi tous les robots anthropomorphes, ceux dont la forme rappelle la morphologie humaine. La ressemblance avec des êtres humains est tellement forte parfois qu'au Japon un robot humanoïde a présenté le journal télévisé du 18 juin 2014. Ce type de robot est communément appelé Androïde.

4.2 Les robots industriels (manipulateurs) :

La majorité de ces robots sont à base fixe. Quand ce n'est pas le cas, ils sont généralement montés sur rail. On trouve dans cette catégorie les robots de manipulation, type « Pick And Place », les robots soudeurs ou encore les robots peintres utilisés dans l'industrie automobile. Ils représentent actuellement la majorité des robots.

4.3 Les robots explorateurs (mobiles) :

De façon générale, cette catégorie englobe aussi bien les robots humanoïdes ayant la faculté de se mouvoir dans leur environnement, que les robots mobiles à roues, ainsi que tous les autres robots capable de déplacement. Les robots mobiles à roues UGV, en anglais (Unmmaned Ground Vehicles), regroupe les robots à base actionnée par des roues ou par des chenilles.

Ils sont généralement utilisés pour l'exploration, c'est pour cette raison qu'ils sont aussi appelés des rovers (vagabonds). Les plus connus sont les robots de la

NASA (Curiosity, Mars Rovers, etc.) présente le robot Curiosity qui a été envoyé sur Mars à des fins d'exploration et d'identification du terrain martien [5].

5. L'autonomie d'un robot mobile:

Un robot autonome peut être défini comme étant l'association des capacités de perception, de modélisation de son environnement et de son propre état, et de capacités d'actions sur son propre état et sur son environnement. Pour cela, pour que le robot mobile exécute correctement ses tâches, il doit fournir à son système de contrôle un modèle interne de l'environnement dans lequel il interagit.

Perception: c'est la capacité du robot à obtenir des informations sur l'environnement dans lequel il interagit, grâce à différents capteurs.

- **Décision:** est de définir les procédures que le robot applique après avoir reçu des informations de l'environnement qui l'entoure.
- **Action:** il s'agit de l'application de procédures en envoyant des informations aux les actionneurs par l'intermédiaire des boucles d'asservissements.

6. Navigation autonome des robots mobiles

La navigation d'un robot mobile est la tache de déterminer une trajectoire optimale permettant au robot de se déplacer d'un point de départ (un point initial) *pi* à un point d'arrivée (un point final) *pf* 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[6].

Il y a deux approches pour les techniques de navigation [7]:

a- 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 panification – exécution) ;

b- 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.

7. Les capteurs de robot mobile:

La robotisation des tâches s'oriente d'une manière générale vers une plus grande prise d'initiatives de la part du robot par rapport à un système automatisé. Le robot choisit ainsi lui-même sa stratégie d'action en fonction de la tâche à réaliser et en fonction de son environnement. Pour accomplir cette opération, le robot doit être doté de capteurs lui permettant d'appréhender en temps réel son environnement [8].

Le rôle du capteur est d'envoyer au robot des informations sur son environnement de réaction, car il convertit l'état physique en un signal électrique.

On peut distinguer deux catégories de capteurs utilisés dans le robot mobile: le premier est appelé capteurs extéroceptifs, car il fournit des informations sur l'environnement externe du robot mobile, et le second est appelé capteurs proprioceptifs, car il fournit des informations sur l'état interne du robot mobile [9].

7.1 Capteurs proprioceptifs :

Fournissent des données sur l'état interne du robot (vitesse, position, orientation,…). Ces informations renseignent le robot en cas de mouvement, sur son déplacement dans l'espace (la localisation). Ce sont des capteurs que l'on peut utilizer directement, mais ils souffrent d'une dérive au cours du temps qui rend leur utilisation seul inefficace ou avec limitation. Nous citons par exemple: l'odomètre, radar doppler, systems inertiels,…[10]

7.2 Capteurs extéroceptifs:

Ont pour objectif d'acquérir des informations sur l'environnement proche du véhicule. Ils fournissent des mesures caractéristiques de la position que le robot peut acquérir dans son environnement par la détection des objets qui contourne. Ces informations peuvent être de natures très variée. Nous citons comme exemple les télémètres à ultrason, infrarouge, laser, les caméras,…etc.

Pour la navigation autonome d'un robot mobile et selon la mission visée (accomplir), on peut utiliser aussi les capteurs suivants [11]:

- Les capteurs taciles
- Les boussoles
- Les balises

- Le GPS (Global Positioning System).

La localisation est le problème d'estimation de l'emplacement du robot mobile par rapport à un plan.

8. Outils nécessaires à la localisation :

On peut classer ces outils en trois grandes catégories : ensembles des outils relatifs à la perception, formalisme de la théorie de l'estimation et technique relative à la planification et au contrôle.

8.1 Perception :

- Problème de segmentation et de structuration des données nécessaires à la construction de la représentation de l'environnement.
- Problème de fusion de données

8.2 Théories de l'estimation :

Les outils qui permettent d'estimer la position et une erreur associée à celle-ci sont nécessaires et essentielles à de nombreuses méthodes de localisation.

Parmi ces outils [12]:

- Les techniques de filtrage stochastiques sont les plus répandus
- Les approches possibilistes Markoviennes
- Les approches ensemblistes ou les erreurs sont représentés par des distributions uniformes à l'intérieur d'intervalle

Ces techniques sont généralement connues et maitrisées mais leur mise en œuvre est parfois délicate. Ainsi les non linéarités des équations du mouvement nécessitent l'utilisation du filtre de Kalman. L'application de ces techniques requièrent donc une bonne expertise et la connaissance d'un modèle d'erreur des algorithmes.

8.3 Contrôle et planification :

Les tâches de recalage et de localisation absolue nécessitent le contrôle de l'acquisition de données, voire même influencer les déplacements à effectuer et donc intervenir dans la manière dont le robot mène ses missions.

9. Méthodes de localisation :

On peut classer les méthodes de localisation en trois types [13]:

- ❖ Les méthodes de localisation relative basée sur l'utilisation des capteurs proprioceptifs
- ❖ Les méthodes de localisation absolue basée sur l'utilisation des capteurs extéroceptifs
- ❖ Les méthodes dites hybrides qui sont basées sur l'utilisation conjointe des deux types précédents

9.1 Localisation à l'estime (relative) [14]:

La navigation à l'estime consiste à évaluer la position et l'orientation (posture) et éventuellement la vitesse du robot mobile par intégration des informations fournies par des capteurs proprioceptifs. L'intégration se fait à partir du point du départ du robot. Ces données peuvent être des infos de déplacement (l'odométrie), de vitesse (vélocimétrie), ou d'accélération (accélérométrie). Ces systèmes permettent d'obtenir un flux relativement important au niveau des estimations de position.

a) L'odométrie :

Elle permet de déterminer la position et le cap (x, y, Φ) d'un véhicule par l'intégration de ces déplacements élémentaires et ce par rapport à un repaire lié à sa configuration initiale. L'algorithme de localisation est basé sur le comptage des impulsions générées par des codeurs durant une période d'échantillonnage connu. Connaissant el rayon des roues et la distance entre chaque roue, il est possible d'exprimer de manière récursive la position et le cap du robot.

Les avantages de l'odométrie :

- Simplicité de mise en œuvre et faible coût.
- Couramment utilisée en robotique mobile.
- Obtention de l'estimation de la position et du cap à une cadence relativement élevée.

Les inconvénients de l'odométrie :

- Précision médiocre sur les distances importantes à cause des erreurs cumulatives.

b) Vélocimétrie :

C’est une méthode qui consiste à mesurer directement la vitesse du véhicule et l’intégrer pour obtenir le déplacement. La vitesse de rotation instantanée dans la majorité des cas est obtenue avec des gyromètres. Quant à la vitesse linéaire, elle est généralement obtenue par un radar doppler dirigé vers le sol.

9.2 Localisation absolue :

C’est une technique qui permet à un robot de se repérer directement dans son milieu d’évolution, que ce soit un environnement extérieur ou intérieur. Elles sont basées sur l’utilisation de capteurs extéroceptifs. Elles nécessitent toujours une représentation de l’environnement. Le robot possède donc une «banque de données».
Elle utilise deux types de stratégie :

❖ Utilisation des points de repère naturels.

Elle consiste à utiliser les éléments caractéristiques de l’environnement pour estimer la position du robot. L’intérêt de ces méthodes est donc sa souplesse d’utilisation. Il s’agit d’une représentation cartographique qui intégrera la position des amers qui serviront à localiser le robot.

❖ Utilisation des points de repère artificiels.

Ce sont des balises caractéristiques qui sont ajoutées au milieu d’évolution du robot et dont les positions sont connues. L’inconvénient de ce type de technique réside dans son manque de souplesse et sa lourdeur d’utilisation (un domaine vaste nécessite un investissement lourd en équipement). En revanche, elle a le gros avantage d’être précise, robuste et surtout de satisfaire la contrainte temps réel.

10. Conclusion

La robotique est une science qui s'intéresse à l'étude et au développement de systèmes automatisés afin qu'ils puissent interagir idéalement au sein de leur environnement.

Dans ce chapitre, nous avons identifié les robots mobiles et les classer en fonction de leur environnement de travail, et nous avons également introduit la caractéristique fondamentale des robots mobiles, qui est la navigation autonome, c'est-à-dire la capacité de naviguer, localiser et cartographier grâce à un ensemble de capteurs internes et externes qui permettent la perception de cet environnement.

1. Introduction

La simulation de processus d'intelligence humaine par des systèmes informatiques est une technologie particulièrement vaste.

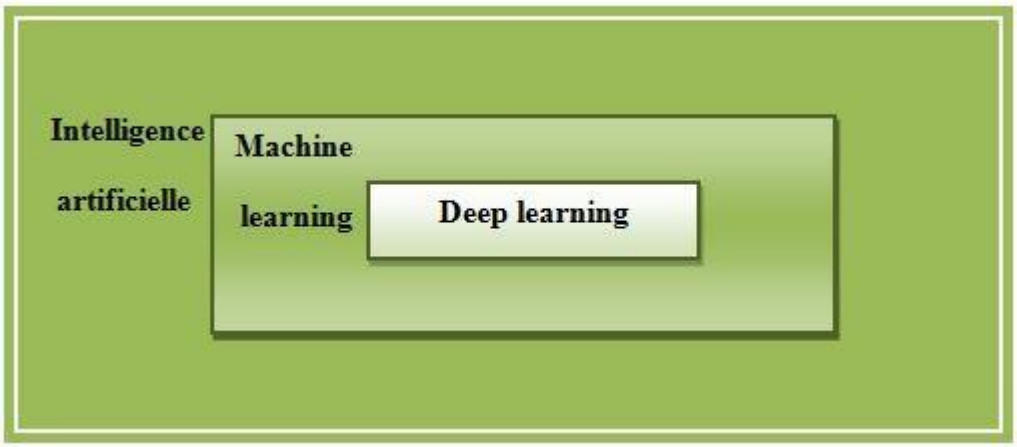


Figure.II.1 Intelligenceartificielle, machinelearninget Deeplearning

2. Intelligenceartificielle

L'intelligence artificielle est née dans les années 50, quand une poignée de pionniers du domaine naissant de l'informatique, ont commencé à se demander silesordinateurs pouvaient être amenés à «penser », une question dont nous explorons encore aujourd'hui les ramifications.

Unedéfinitionconciseduchampseraitlasuivante:l'effortd'automatiserles tâches intellectuelles normalement effectuées par les humains. En tant que tel, l'IA est un domaine général qui englobe l'apprentissage automatique et l'apprentissage en profondeur, mais qui comprend également beaucoup plus d'approches qui n'impliquent aucun apprentissage.Lesprogrammesd'échecsinitiaux,par exemple,ne concernaient que desrègles codées en dur élaborées par des programmeurs et ne se qualifiaient pas comme apprentissage automatique. Pendant un temps assez long, de nombreux experts ont estimé que l'intelligence artificielle au niveau humain, pouvait être obtenue en faisant en sorte que les programmeurs fabriquent à la main un ensemble suffisamment large de règles explicites pour manipuler les connaissances. Cette approche est connue sous le nom d'IA symbolique et elle était le paradigme dominant de l'IA des années 50 et à la fin des années 80. Elle a atteint son pic de popularité durant le boom des systèmes experts des années 80. Bien que l'IA symbolique se soit révélée appropriée pour résoudre des problèmes logiques bien définis, comme jouer aux échecs, ilétait

difficiledetrouver des règlesexplicitespour résoudredesproblèmesflous plus complexes, tels que la classification des images, la reconnaissance de la parole et latraduction. Une nouvelle approche est apparue pour prendre la place de l'IA symbolique: l'apprentissage automatique [15].

3. MachineLearning

L'apprentissage automatique(Machine learning) est undomaine de recherche en informatique qui traite des méthodes d'identification et de mise en œuvre de systèmes et algorithmes par lesquels un ordinateur peut apprendre, ce domaine a souvent été associé à l'intelligence artificielle et plus spécifiquement l'intelligence computationnelle.

L'intelligence computationnelle est une méthode d'analyse de données qui pointe vers la création automatique de modèles analytiques. Autrement dit, permettant à un ordinateur d'élaborer des concepts, d'évaluer, prendre des décisions et prévoir les options futures [16].

L'ensembleduprocessusd'apprentissagenécessiteunensemble dedonnéescommesuit:

- Ensemble de données pour l'entrainement : c'est la base de connaissance utilisée pour entrainer, notre l'algorithme d'apprentissage, pendant cette phase, les paramètres du modèle peuvent être réglés (ajustés) en fonction des performances obtenues.
- Ensemble de données pour le test : cela est utilisé juste pour évaluer les performances du modèle sur les données non-vues.

Lathéoriedel'apprentissageutilisedesoutilsmathématiquesdérivésdelathéoriedes probabilités et de la théorie de l'information.

Celavouspermetd'évaluer l'optimalité decertaines méthodesparrapportauxautres. On peut citer trois types d'algorithme d'apprentissage automatique :

- Apprentissagesupervisé.
- Apprentissagenonsupervisé.
- Apprentissageparrenforcement.

3.1 Apprentissage supervisé

L'apprentissage supervisé est la tache d'apprentissage automatique la plus simple et la plus connue. Il est basé sur un certain nombre d'exemples pré classifiés, dans lesquels est connu à priori la catégorie à laquelle appartient chacune des entrées utilisées comme exemples.

Dans ce cas, la question cruciale est le problème de généralisation, après l'analyse d'un échantillon d'exemples, le système devrait produire un modèle qui devrait fonctionner pour toutes les entrées possibles.

L'ensemble de données pour l'entrainement, est constitué de données étiquetées, c'est-à-dire d'objets et de leurs classes associées. Cet ensemble d'exemples étiquetés constitue donc l'ensemble d'apprentissage.

Afinde mieux comprendre ce concept, prenons unexemple : un utilisateur reçoit chaque jour un grand nombre d'e-mails, certains sont des e-mails d'entreprises importants et d'autres sont des e-mails indésirables non sollicités ou des spam.

Un algorithme supervisé sera présenté avec un grand nombre d'e-mails qui ont déjà été étiquetés par l'utilisateur comme spam ou non spam. L'algorithme fonctionnera sur toutes les données étiquetées, faire des prédictions sur l'e-mail et voir si c'est un spam ou non.

Cela signifie que l'algorithme examinera chaque exemple et fera une prédiction pour chacun pour savoir sil'e-mailest unspamou pas. La première fois, l'algorithme fonctionne sur toutes les données non étiquetées, la plupart des e-mails seront malétiquetés car il peut fonctionner assezmalau début.Cependant, aprèschaqueexécution, l'algorithme comparesa prédictionau résultat souhaité (l'étiquette). Au fur et à mesure, l'algorithme apprendra à améliorer ses performances et sa précision.

Dans l'exemple que nous avons utilisé, nous avons décrit un processus dans lequel un algorithme apprend à partir de données étiquetées (emails qui ont été catégorisés commespam ou non-spam).

Dans certains cas, le résultatn'est pas nécessairementdiscret etil se peutque nous n'ayons pas un nombre fini de classes dans lesquelles classer nos données. Par exemple, nousessayons peut-être de prédire l'espérance de vie d'un groupe de personnes en fonction de paramètresdesantépréétablis. Danscecas, commelerésultat

est unefonctioncontinue (nous pouvons spécifier une espérance de vie comme un nombre réelexprimant le nombre d'années que la personne devrait vivre), nous ne parlons pas d'une tâche de classification mais plutôt d'un problème de régression.
Dans un problème de régression, l'ensemble d'apprentissage est une paire formée par un objet et unevaleur numériqueassociée.
Il existe plusieursalgorithmesd'apprentissage supervisé qui ont été développés pour la classification etla régression. Parmi tous, les arbres de décision,les règles de décision, les réseaux de neurones et les réseaux bayé siens[17]

3.1 Apprentissagenonsupervisé

La deuxième classe d'algorithmes d'apprentissage automatique est appelée apprentissage non supervisé, dans ce cas, nous n'étiquetons pas les données au préalable, nous laissons plutôt l'algorithme arriver à sa conclusion.

Ce type d'apprentissage est important car il est beaucoup plus commun dans le cerveau humain que l'apprentissage supervisé.

Les algorithmes d'apprentissage non supervisé sont particulièrement utilisés dans les problèmes de clustering, dans lesquels, étant donné une collection d'objets, nous voulons être en mesure de comprendre et de montrer leurs relations. Une approche standard consiste à définir une mesure de similarité entre deux objets, puis à rechercher tout groupe d'objets plus similaires les uns aux autres, par rapport aux objets des autres clusters. Par exemple, dans le cas précédent des e-mails spam/ non spam, l'algorithme peut être capable de trouver des éléments communs à tous les spam (par exemple, la présence de mots mal orthographiés). Bien que cela puisse fournir une classification meilleure qu'aléatoire, il n'est pas clair que les spam/non spam puissent être facilement séparés.

3.2. Apprentissageparrenforcement

L'apprentissage par renforcement est une approche de l'intelligence artificielle qui metl'accent sur l'apprentissage du système à travers ses interactions avec l'environnement. Avec l'apprentissage par renforcement, le système adapte ses paramètres en fonction des réactions reçues de l'environnement, quifournit ensuite unretour d'informationsur les décisions prises. Par exemple, un système qui modélise un joueur d'échecs qui utilise le résultat des étapes précédentes pour améliorer ses performances, est un système qui apprend avec le renforcement. La recherche actuelle

sur l'apprentissage avec renforcement est hautement interdisciplinaire et comprend des chercheurs spécialisés dans les algorithmes génétiques, les réseaux de neurones, la psychologie et les techniques de contrôle. [18]

La figure suivante résume les trois types d'apprentissage avec les problèmes connexes à résoudre :

Figure.II.2 Typesd'apprentissage

4. Conclusion

Nous avons consacré ce chapitre à la présentation des notions de base comme L'IA et son évolution, l'apprentissage automatique et ses types et on conclut avec les algorithmes les plus populaires et les plus utilisés.

Chapitre 3

1. Introduction

La logique floue est le meilleur moyen de convertir les stratégies de contrôledu langage naturel qui sont utilisées par1'homme à une forme utilisable par les machines. Des expériences ont montré qu'un contrôleur classique donne parfois même de meilleurs résultats que l'opérateur humain [19]. Leprincipal avantage delalogique floue estqu'elle permet aux experts humains d'exprimer des concepts. Ce qui entraine un gain de temps et d'espace pour une recherche dans les règles pour1'exécution d'une situation donnée.

2. Caractéristiques de la logique floue

D'une manière générale l'approche floue possède les caractéristiques suivantes [20]:

a. Identification et utilisation des variables linguistiques qui subissent les contraintes, à la place des variables numériques.

b. Des variables caractérisées subjectivement (non précises) sont utilisées.

c. Des critères décrits linguistiquement dont les qualifications sont mal définis, comme la beauté d'une couleur ou le confort d'un passager, onts utilisées.

d. La connaissance mathématique du fonctionnement du processus n'est pas nécessaire

e. Caractérisation des relations simples entre les variables par des citations conditionnelles floues, vues comme une collection de contraintes.

f. Caractérisation de relations complexes par des algorithmes flous.

3. Logique booléenne, logique floue et fonction d'appartenance :

Pour illustrer très concrètement le principe fondamental de la logique floue [21], nous allons prendre l'exemple précédent. Nous souhaitons évaluer les tailles des personnes. Grande taille ou petite taille ?

Dans le cadre de la logique booléenne, il est nécessaire d'introduire un seuil correspondant à la valeur limite de la taille qui va déterminer si la taille est grande ou petite (figure III.3).

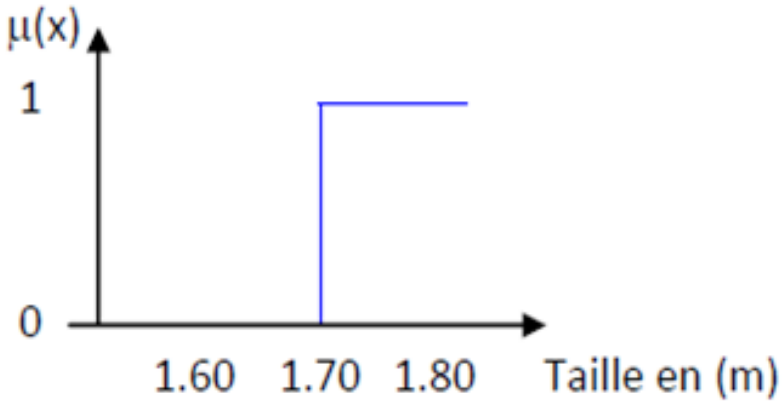


Figure.III.1 Logique classique

La logique floue, qui introduit une infinité de valeurs entre vrai et faux, permet de déterminer un degré d'appartenance à l'un ou l'autre état. La logique floue comble donc les lacunes de la logique booléenne en introduisant la notion de continuité entre les états.

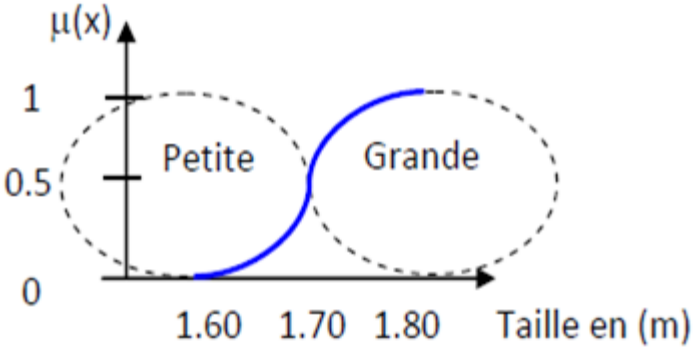


Figure.III.2 Logique floue

4. La théorie des sous-ensembles :

Soit une variable x (la taille) et un univers de référence ou de discourt U (l'intervalle de tailles entre 1.60 et 1.80 m), Un sous-ensemble flou A est défini par une fonction d'appartenance μ(x) qui décrit le degré avec lequel l'élément x appartient à A.

4.1 Théorie Floue :

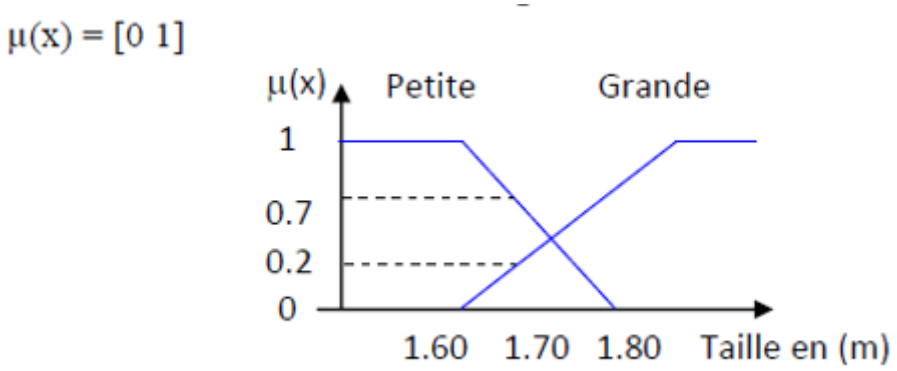


Figure.III.3 Théorie Floue.

Ici la taille de 1.65 m appartient en même temps au sous-ensemble flou (Petite) et au sous-ensemble flou (Grande), respectivement avec un degré de 0,7 et de 0,2.

5. Les Fonctions d'appartenance[21]

On définit une fonction d'appartenance à un ensemble flou *A* comme une application qui accorde à tout élément *x* de l'univers *U* ($A \subseteq U$) un degré d'appartenance entre *0* et *1* pour lequel *x* appartient à cet ensemble. Cette application généralise le concept d'appartenance classique et est notée par : $\mu_A(x)$.

Les formes les plus connues et les plus utilisées de ces fonctions sont : triangulaires, trapézoïdales et gaussiennes (figure III.4).

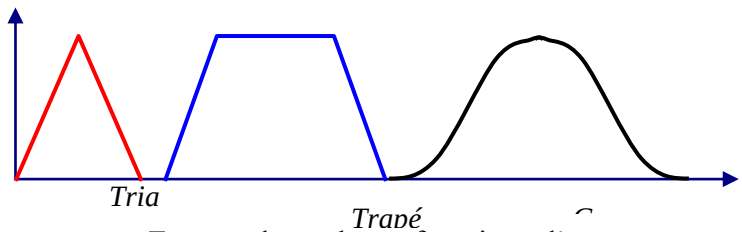


Figure.III.4 Formes de quelques fonctions d'appartenance.

6. Les opérations sur les sous-ensembles floue[22]:

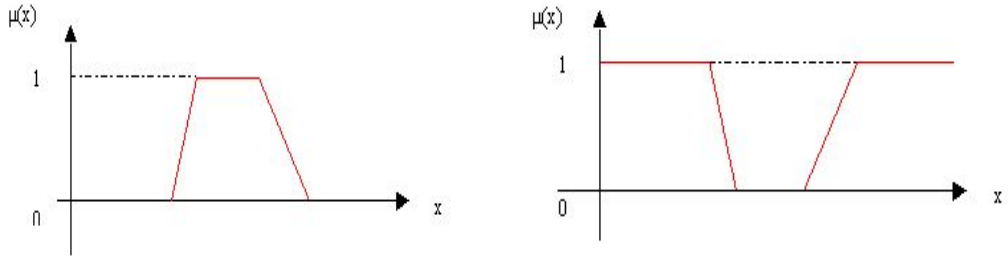
La théorie mathématique sur les sous-ensembles flous définit de nombreuses opérations sur ces sous-ensembles et sur les fonctions d'appartenances qui rendent ces notions utilisables. Les opérations possibles sur les sous-ensembles flous sont les mêmes que dans le cas de la théorie ensembliste classique, Ces opérations sont:

Si A et B sont deux sous-ensembles flous de E, et μ_A , μ_B leur fonction d'appartenance, on définit :

5.1 La Complementation

Le complément d'un ensemble flou est le « **NON logique** ». Le complémentaire de A est $\overline{\mu_A}$ donner par la fonction d'appartenance :

$$\forall x \in E, \quad \overline{\mu_A}(x) = 1 - \mu_A(x)$$



l'ensemble A l'ensemble complémentaire de A

Figure.III.5 Complément d'un ensemble flou

5.2 L'inclusion

A est inclus dans B si et seulement si : $\forall x \in E$, $\mu_B(x) < \mu_A(x)$ on μ_A on noton $B \subset A$

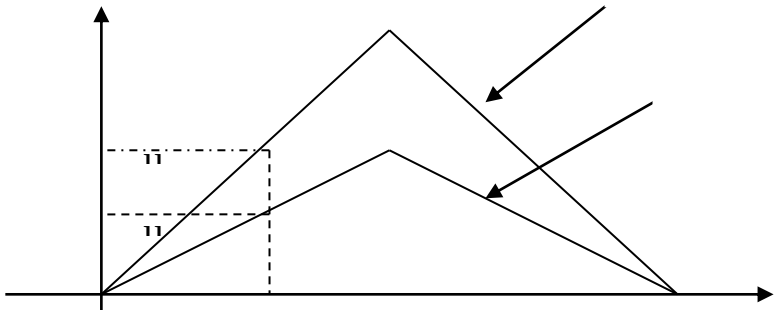


Figure III.6 L'inclusion de B dans A

5.3 L'union

C'est le « **OU logique** » de deux ensembles flous, est calculée en prenant le maximum des deux valeur d'appartenance en chaque point du domaine commun. Mathématiquement, cela donne : A∪B, donner par la fonction d'appartenance : $\forall x \in E, \quad \boldsymbol{\mu}_{A\cup B}(x) = \text{Max}(\boldsymbol{\mu}_A(x), \boldsymbol{\mu}_B(x))$

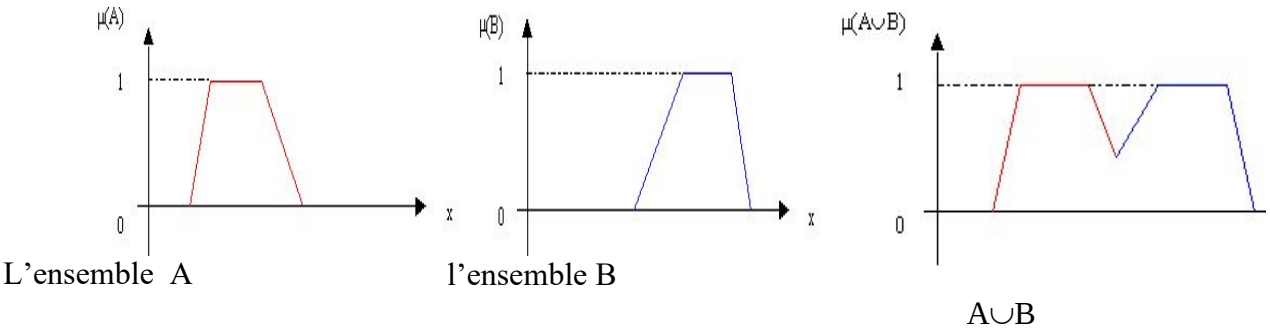


Figure III.7 L'union de deux ensembles floue

5.4 L'intersection

C'est le **ET** logique de deux ensembles flous, A∩B, donner par la fonction d'appartenance : $\forall x \in E, \ \boldsymbol{\mu}_{(A\cap B)} = \min(\boldsymbol{\mu}_A(x), \boldsymbol{\mu}_B(x))$

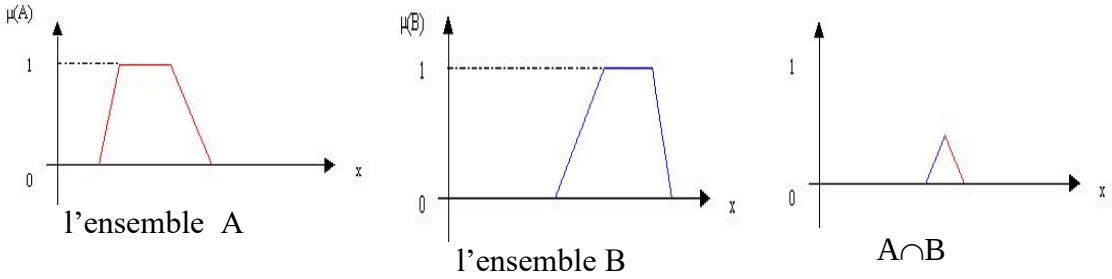


Figure III.8 L'intersection de deux ensembles floue

5.5 L'égalité

$$A=\mathbf{B} \text{ si et seulement si } :\forall x \in E \text{ , } \boldsymbol{\mu}_A(x) = \boldsymbol{\mu}_B(x)$$

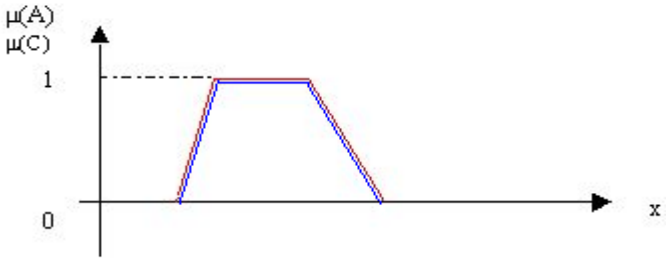


Figure.III.9 L'égalité de deux ensemble floue

7. Commande floue

Les contrôleurs flous sont des contrôleurs basés sur des règles linguistiques. La commande conventionnelle commence par un modèle mathématique du système, et les contrôleurs sont conçus pour ce modèle. La commande floue commence par l'heuristique et l'expertise humaine, et les contrôleurs sont conçus par la synthétisation de ces heuristiques et expertise humaine. Bien que les points de départ des deux approches soient différents, les produits finaux sont identiques des contrôleurs non linéaires pour les systèmes non linéaires, à cet égard, la théorie de commande floue peut être regardés comme sous ensemble de théorie de commande non linéaire dans lequel les contrôleurs non linéaires ont une structure basée sur des règles spéciales [23]

7.1 But de la commande floue

La commande floue a pour but de traiter des problèmes de commande classique de processus à partir uniquement de connaissances de comportement que les spécialistes du procédé doivent formuler sous forme linguistique (floue).

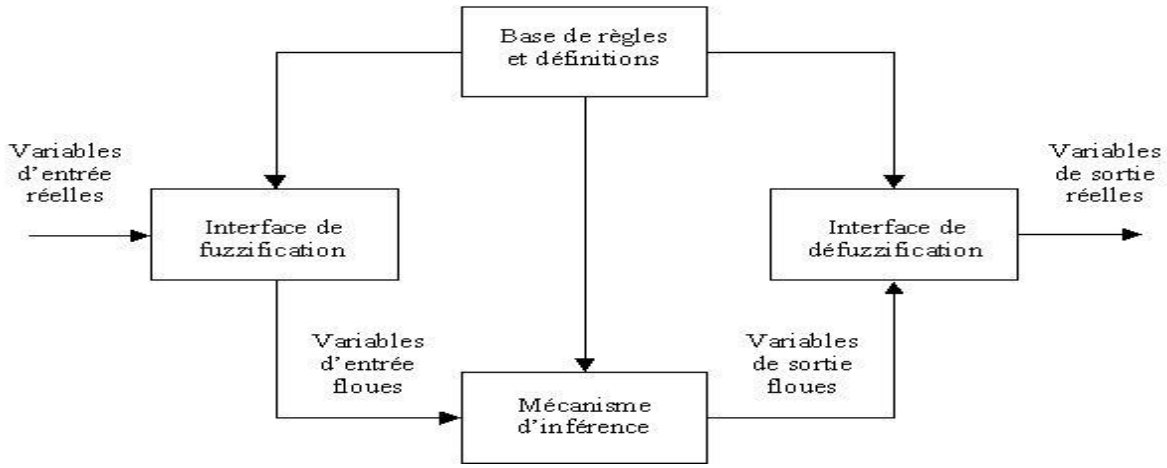


Figure.III.10 Schéma d'une commande floue

On procède tout d'abord à la partition en sous-ensembles flous des différents univers de discours (ou référentiels) que le système impose. Ensuite on détermine **la base de règles** qui va caractériser le fonctionnement désiré du système. Puis il faut transformer les variables réelles, c'est à dire celles qui ont une réalité physique, en variables floues. On appelle cette étape **la fuzzification** (de fuzzy=flou en anglais). On utilise alors ces variables floues dans **un mécanisme d'inférence** qui crée et détermine les variables floues de sortie en utilisant les opérations sur les fonctions d'appartenance.

Enfin, on opère à **la défuzzification** qui consiste à extraire une valeur réelle de sortie à partir de la fonction d'appartenance du sous-ensemble flou de sortie établi par le mécanisme d'inférence.

8. Systèmes d'inférence flous

La configuration de base d'un système d'inférence flou se compose de cinq blocks fonctionnels [24]

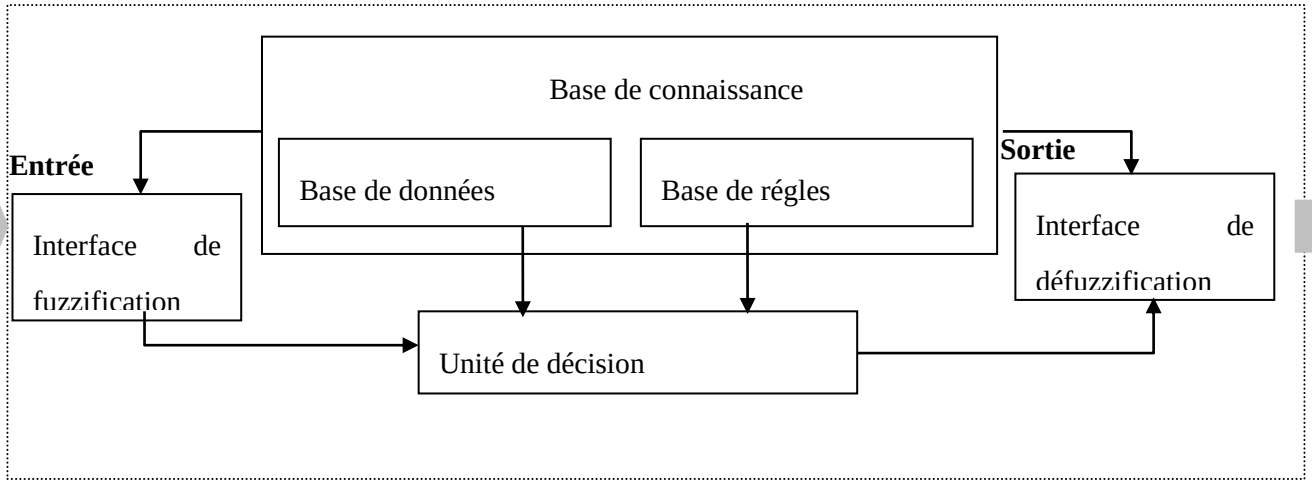


Figure.III.11Système d'inférence floue

7.1 Fuzzyfication [25]

La fuzzyfication des variables d'entrée est une phase délicate du processus, mise en œuvre par la logique floue. Elle est souvent réalisée de manière itérative et requiert de l'expérience.

7.2 Les variables floues:

Les grandeurs utilisées dans un système de réglage sont généralement générées par des capteurs. Il est nécessaire de convertir ces grandeurs en variables floues. Pour ce faire on définit les deux notions suivantes :

- Les fonctions d'appartenance qui permettent de définir le degré de vérité de la variable floue en fonction de la grandeur d'entrée.
- Les intervalles flous qui déterminent le nombre de variables floues

7.3 Règles d'inférences et operateurs:

Après avoir "fuzzyfier" les variables d'entrée et de sortie, il faut établir les règles liant les entrées aux sorties. En effet, il ne faut pas perdre le but final que nous poursuivons qui consiste à chaque instant, à analyser l'état ou la valeur des entrées du système pour déterminer l'état ou la valeur de toutes les sorties.

Ces règles permettent de relier les variables floues d'entrée aux variables floues de sortie à l'aide de différents opérateurs. Elles doivent être définies par le

concepteur du système de réglage en fonction de son expérience (rôle d'expert) et mémorisées dans l'organe de commande.

7.3.1 Combinaison des règles

Les différentes règles d'inférences produisent chacune une valeur. Ces différentes valeurs doivent être combinées afin d'obtenir la (éventuellement les) variable(s) de sortie.

L'ensemble des règles se présente sous la forme d'une énumération du type :

Si condition 1 et/ou condition 2 (et/ou...) alors action1 sur les sorties

Si condition 3 et/ou condition 4 (et/ou...) alors action2 sur les sorties

Si condition 5 et/ou condition 6 (et/ou...) alors action3 sur les sorties

...

La combinaison de ces différentes règles se fait à l'aide de l'opérateur **ou.** La justification du choix de l'opérateur se fonde sur la pratique du langage courant : en effet, une telle énumération est comprise dans le sens :

Si... alors...

ou

Si... alors...

ou

...

7.3.2 Opérateurs:

Les règles d'inférences font appel aux opérateurs **et**, **ou** et **non**, qui s'appliquent aux variables floues. Dans le cas de la logique binaire ces opérateurs sont définis de façon simple et univoque. Dans le cas de la logique floue, la définition de ces opérateurs n'est plus univoque et on utilise le plus souvent les relations présentées dans le tableau 1.

Opérateur	Opération sur le degree de vérité des variables
ET	Minimum
	Produit
OU	Maximum
	Valeur moyenne
NON	Complément à 1

Tableau III.1 : Opérateurs d'inférences flous

7.4 Défuzzyfication [26]

Qui consiste à caractériser les variables linguistiques utilisées dans le système. Il s'agit donc d'une transformation des entrées réelles en une partie floue définie sur un espace de représentation lié à l'entrée. Cet espace de représentation est normalement un sous-ensemble flou. Durant l'étape de la fuzzification, chaque variable d'entrée et de sortie est associée à des sous-ensembles flous.

8. Conclusion

La logique floue est une extension de la logique classique qui permet la modélisation des imperfections des données et se rapproche dans une certaine mesure de la flexibilité du raisonnement humain

Chapitre 04

1. Introduction

Les réseaux neuro-flous, parfois appelés réseaux neuro-flous adaptatifs, sont des modèles hybrides qui fusionnent des concepts issus des réseaux neuronaux artificiels et de la logique floue. Ils constituent une approche robuste pour la modélisation et la gestion de systèmes complexes, en intégrant la représentation et le traitement des situations incertaines et des éléments vagues.

2. Réseaux neuro-flou

Jusqu'aux années 90, les approches neuronales et floues étaient considérées comme distinctes, chacune ayant ses avantages, inconvénients et domaines d'application spécifiques. En matière de contrôle des processus, la logique floue se prêtait bien à une interaction avec un opérateur humain en raison de sa capacité à traiter des termes linguistiques. Cependant, le choix des fonctions d'appartenance et la rédaction des règles introduisaient souvent une part d'arbitraire et d'imprécision, conduisant à des solutions éloignées de l'optimum. D'autre part, les réseaux neuronaux (tels que les réseaux multicouches) étaient particulièrement adaptés lorsque seules des données numériques étaient disponibles pour élaborer le contrôle. Ces réseaux apprenaient par l'exemple en minimisant une fonction de coût, leur permettant ainsi de généraliser efficacement. Cependant, il était difficile d'incorporer des connaissances a priori dans ces réseaux, car leur structure interne était souvent difficilement interprétable (une boîte noire).

Ces deux méthodes présentent également des similitudes :

Les deux approches, lorsqu'appliquées au contrôle, démontrent une bonne robustesse, ainsi que des capacités d'extrapolation et de généralisation.

Aucune des deux méthodes ne nécessite un modèle mathématique du système à contrôler, ce qui leur permet d'être appliquées à des systèmes non linéaires sans complications majeures.

C'est ainsi qu'au début des années 90 est apparue l'idée de créer un système d'inférence floue optimisé, utilisant des méthodes d'apprentissage supervisé telles que la rétropropagation du gradient, afin d'ajuster automatiquement certains paramètres d'un système d'inférence floue.

Il existe plusieurs méthodes pour mettre en œuvre la technique de modélisation neuro-floue. En général, l'hybridation neuro-floue peut être réalisée de deux manières.

En équipant un réseau neuronal de capacités de traitement de l'information floue, ce qui donne un réseau neuronal flou (FNN).

En combinant un système flou avec des réseaux neuronaux pour améliorer certaines de ses caractéristiques telles que la flexibilité, la vitesse et la capacité d'adaptation, ce qui donne un système neuro-flou (NFS).

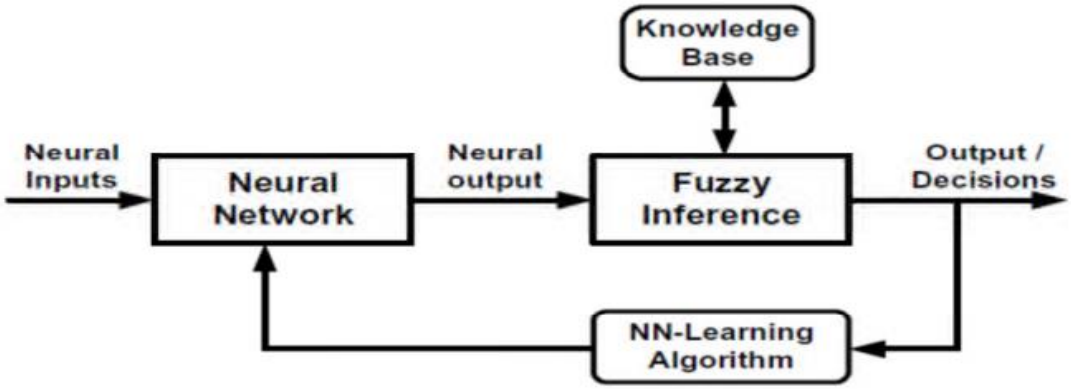


Figure IV.1 System Neuro-Flou

Dans ce qui suit, nous allons présenter l'architecture neuro-flou qui permettent d'optimiser un système d'inférence flou l'architecture ANFIS.

3. L'architecture ANFIS

Le modèle ANFIS (Adaptative Neuro-Fuzzy Inference System), proposé en 1993 par Jang [28], est un réseau de neurones flou qui comprend cinq couches, comme illustré dans la Figure IV.2. Les nœuds du réseau sont de deux types différents : des nœuds adaptatifs représentés par des carrés et des nœuds fixes représentés par des cercles. Dans le but de simplifier la compréhension, nous examinerons un modèle spécifique. Cette technique utilise la méthode des moindres carrés combinée à la rétropropagation du gradient pour l'apprentissage.

Afin de faciliter la compréhension sans perdre de généralité, considérons un système ayant deux entrées, x_1 et x_2, et une sortie, y. Nous allons également prendre en compte un modèle flou pour ce système, composé des deux règles suivantes :

Si x_1 est A_1 **et** x_2 est B_1 **alors** $y_1 = F_1(x_1, x_2) = a_1 x_1 + b_1 x_2 + c_1$ (1)

Si x_1 est A_2 **et** x_2 est B_2 **alors** $y_2 = F_2(x_1, x_2) = a_2 x_1 + b_2 x_2 + c_2$ (2)

Jang et al. [28] a proposé de représenter cette base de règles à l'aide du réseau adaptatif illustré dans la figure IV.2. La méthode ANFIS repose sur l'utilisation de réseaux multicouches (réseaux adaptatifs) où chaque cellule exécute une fonction spécifique en fonction des paramètres qui lui sont attribués.

Le réseau adaptatif ANFIS est un réseau multicouche dans lequel les connexions ne sont pas pondérées ou ont toutes un poids de 1. Les nœuds du réseau se distinguent en deux types en fonction de leur fonctionnalité : les nœuds carrés (adaptatifs) contiennent des paramètres, tandis que les nœuds circulaires (fixes) n'en possèdent pas. Cependant, chaque nœud (carré ou circulaire) applique une fonction à ses signaux d'entrée. La sortie du nœud i de la couche k (appelé nœud (i k,)) dépend à la fois des signaux provenant de la couche k-1 et des paramètres associés au nœud (i k,), c'est-à-dire :

$$O_i^k = f\left(O_1^{k-1}, \ldots, O_{n_{k-1}}^{k-}, a, b, c, \ldots\right) \quad (3)$$

Où n_{k-1} représente le nombre de nœuds dans la couche k-1, et a, b, c, ..., sont les paramètres du nœud (i k,). Il convient de noter que pour un nœud circulaire, ces paramètres n'existent pas.

Dans le réseau illustré dans la figure IV.2, les nœuds appartenant à une même couche partagent des fonctions issues d'une même famille, que nous détaillons ci-dessous :

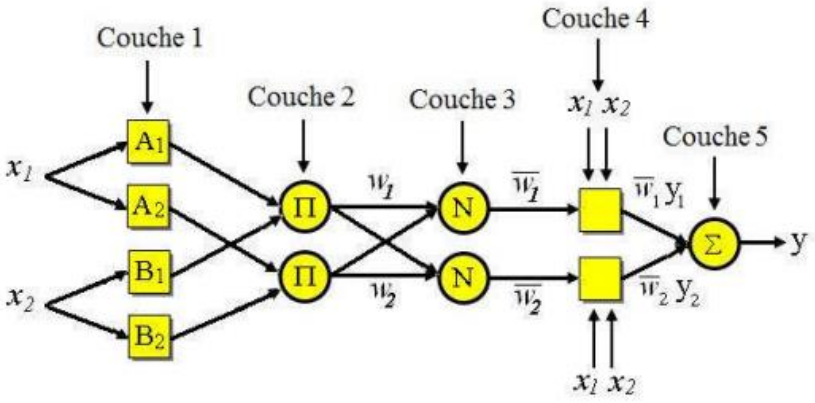


Figure IV.2 ARCHITECTURE ANFIS

Couche 1 : Fuzzyfication La première couche, appelée Fuzzyfication, a pour rôle de convertir les variables d'entrée en leur représentation floue en utilisant des fonctions d'appartenance. Chaque nœud de cette couche est représenté par un nœud carré et possède une fonction spécifique :

$$O_i^1 = \mu_{A_i}(x) \quad (4)$$

Dans cette couche, x représente l'entrée du nœud i, et Ai est le terme linguistique associé à sa fonction. Ainsi, le résultat obtenu par cette cellule représente le degré selon lequel x satisfait le qualificatif Ai. En d'autres termes, Oi1 correspond au degré d'appartenance de x à A_i. Les paramètres d'un nœud de cette couche sont ceux de la fonction d'appartenance correspondante. Les fonctions d'appartenance généralement utilisées dans la méthode ANFIS sont des gaussiennes ou des fonctions cloches.

Couche 2 : Pondération des règles floues, Chaque neurone de cette couche est associé à une règle floue et permet de déterminer le poids de cette règle en utilisant une t-norme. La t-norme la plus couramment utilisée est le produit. Chaque nœud i de cette couche est représenté par un nœud circulaire appelé Π, qui produit en sortie le produit de ses entrées. Ce produit représente le degré d'activation d'une règle :

$$w_i = \mu_{A_i}(x_1) \cdot \mu_{B_i}(x_2), i = 1..2 \quad (5)$$

Couche 3 : Normalisation, Le rôle de cette couche consiste à normaliser les différents poids attribués. Chaque nœud de cette couche est représenté par un nœud circulaire appelé N. Dans cette couche, les cellules calculent le rapport entre les valeurs de vérité de chaque règle et la somme des valeurs de vérité de toutes les règles. Elles estiment ainsi le poids "normalisé" de chaque règle.

$$v_i = \frac{w_i}{w_1 + w_2} \quad (6)$$

Couche 4 : Defuzzyfication Dans cette couche, la sortie est déterminée par une fonction du premier ordre des entrées pondérées, selon l'approche de Takagi-Sugeno. Chaque nœud de cette couche est représenté par un nœud carré, et sa fonction effectue le calcul suivant :

$$O_i^4 = v_i F_i = v_i(a_i x_1 + b_i x_2 + c_i), i = 1..2 \quad (7)$$

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

Couche 5 : Calcul de la sortie, Dans cette couche, la sortie O est calculée par un seul nœud circulaire. Ce nœud effectue la somme des signaux provenant de la couche 4, c'est-à-dire :

$$O_1^5 = y = \sum_i \, v_i F_i \qquad (8)$$

Nous pouvons observer que la sortie globale du réseau est équivalente à la sortie du modèle. La généralisation du réseau à un système avec n entrées ne présente aucun problème particulier. Le nombre de nœuds dans la couche 1 est toujours égal au nombre total de termes linguistiques définis. Le nombre de nœuds dans les couches 2, 3 et 4 est toujours égal au nombre de règles floues.

La figure IV.3 présente un exemple de réseau ANFIS correspondant à un système d'inférence floue à deux entrées avec 9 règles. Chaque entrée est associée à trois fonctions d'appartenance, ce qui signifie que l'espace d'entrée est divisé en 9 sous-espaces flous, chacun couvrant une règle floue.

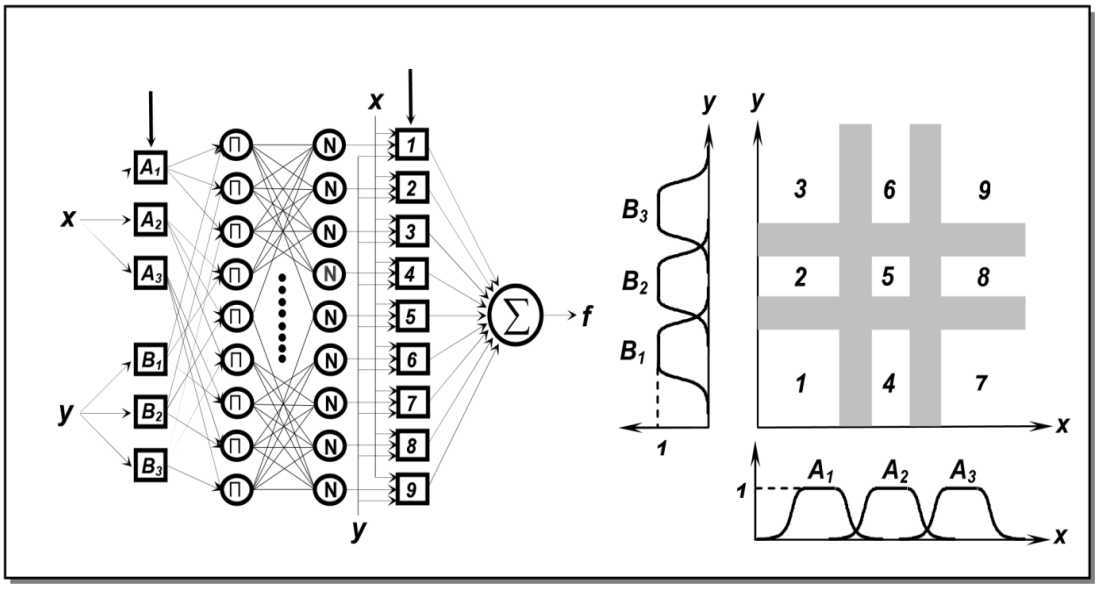


Figure IV.3 Une structure ANFIS 2-entrées avec 9 règles

Les différentes couches	Type des couches	Le nombre de neurone dans la couche
Couche 0	Les entrées	n
Couche 1	Les valeurs	(p.n)
Couche 2	Les régles	p^n
Couche 3	La normalization	p^n
Couche 4	Linéarisation des fonctions	p^n
Couche 5	Somme	1

Tableau IV.1 les différentes couches d'un système ANFIS

tel que : n : le nombre des entrées. p : le nombre des sous ensembles flous d'entrée (partition flou).

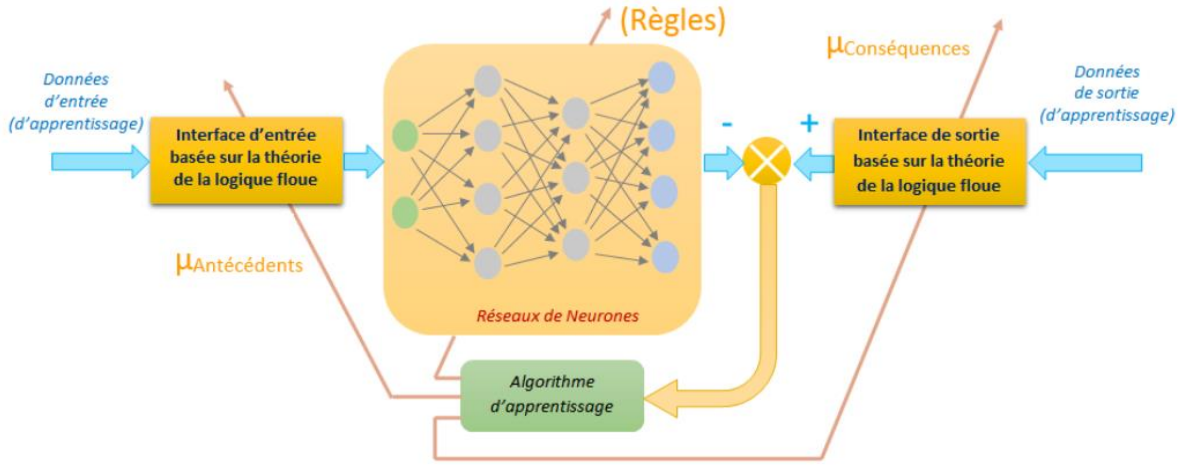
4.1 Algorithme d'apprentissage

L'apprentissage à partir d'un ensemble de données concerne l'identification des paramètres des prémisses et des conséquences, avec la structure du réseau fixée. Pour cela, L'algorithme d'apprentissage basé sur la rétropropagation de l'erreur. Jang et al. [28] a proposé une règle hybride d'apprentissage qui combine un algorithme de descente de gradient avec une estimation par moindres carrés (MC).

Dans l'architecture ANFIS, l'apprentissage repose sur une méthode hybride qui combine la méthode des moindres carrés avec la rétropropagation du gradient tout au long du processus d'apprentissage. Les paramètres des conclusions des règles, qui sont linéaires par rapport à la sortie du système d'inférence floue, sont ajustés par la méthode des moindres carrés dans la direction "forward" du réseau. Ensuite, les paramètres des prémisses des règles sont ajustés par la méthode du gradient dans la direction "backward" du réseau. Cette procédure est répétée à chaque itération de l'algorithme.

Pour implémenter un réseau neuro-flou, il existe plusieurs types en fonction de la fonction d'appartenance utilisée (triangulaire, gaussienne, etc.) et de l'opérateur ET utilisé (opérateur min, opérateur produit, ET de Lukasiewicz, etc.). En raison de ces différentes variantes, la formule d'apprentissage peut varier d'un type à un autre.

Le schéma de l'apprentissage des systèmes neuro-flous est illustré dans la figure IV.4.



FigureIV.4 Synoptique de l'apprentissage des systèmes neuro-flous.

En général, le processus d'apprentissage d'un réseau neuro-flou se divise en deux phases distinctes :

Le réglage des règles floues : Cette étape consiste à ajuster les intervalles des ensembles flous pour obtenir la conclusion désirée. Cela implique de déterminer les valeurs appropriées des paramètres des règles afin de représenter au mieux le comportement souhaité du système.

Le réglage des paramètres des fonctions d'appartenance : Dans cette phase, on cherche à modifier les paramètres des fonctions d'appartenance afin de les adapter aux données d'entrée. Cela peut impliquer de modifier les dimensions, les formes ou les positions des fonctions d'appartenance pour obtenir une meilleure adéquation avec les données observées.

L'identification des paramètres non-linéaires de la partie prémisse des règles floues se fait généralement en utilisant des fonctions d'appartenance de type gaussienne. Ces fonctions sont définies par la relation**(8)** , où les paramètres sont ajustés afin de capturer au mieux les caractéristiques des données et d'optimiser les performances du système d'inférence.

$$\mu_{Ajj}(x_i) = \exp\left\{-0.5\left(v_i^j\left(x_i - c_i^j\right)\right)^2\right\} \text{ Avec } v_i^j = \frac{1}{\sigma_i^j} \qquad (9)$$

La relation **(8)** décrit la modélisation des paramètres non-linéaires de la partie prémisse des règles floues à l'aide de fonctions d'appartenance de type gaussienne. Ces

fonctions sont paramétrisées par la moyenne c et l'inverse de la variance v, ce dernier étant adapté pour éviter la division par zéro.

Pour trouver les valeurs optimales du vecteur θ, qui représente l'ensemble des paramètres des prémisses, nous utilisons un ensemble de données d'entrée-sortie (x(k), $y_d(k)$). Notre objectif est de minimiser le critère afin que la sortie du système flou se rapproche au mieux de la sortie désirée $y_d(k)$.

$$J = \frac{1}{2}\sum\left(f(\underline{x}(k), \underline{\theta}) - y_d(k)\right)^2 \quad (10)$$

L'algorithme de rétro-propagation est donné par:

$$\underline{\theta}(k+1) = \underline{\theta}(k) - \lambda\frac{\partial J}{\partial \theta} \quad (11)$$

Avec λ est le gain d'apprentissage.

L'algorithme utilisé, également connu sous le nom d'algorithme de rétropropagation du gradient, est souvent associé aux réseaux de neurones. Toutefois, il présente des inconvénients, tels qu'une vitesse d'apprentissage souvent lente et la possibilité de se retrouver coincé dans des minima locaux.

Dans un système flou de type Sugeno θ=[c v]T , où [c, v] sont les paramètres des prémisses, nous avons donc:

$$\frac{\partial J}{\partial \theta} = \left[\frac{\partial J}{\partial c} \quad \frac{\partial J}{\partial v}\right]^T \quad (12)$$

l'expression de la sortie du système flou, il vient:

$$\frac{\partial J}{\partial c_i^j} = \sum_{k \in I_{c_i^j}} \frac{\partial J}{\partial \mu_k}\frac{\partial \mu_k}{\partial \mu_{A_i^j}}\frac{\partial \mu_{A_i^J}}{\partial c_i^j} \quad (13)$$

Où I j : est l'ensemble des indices (l) des règles floues (Rl) dont lesquelles apparaît l'ensemble flou A^j_i

$$\frac{\partial J}{\partial \mu_k} = \frac{\mu_k(y_k - y_d)}{\sum_{l=1}^{M}\mu_l} \quad (14)$$

$$\frac{\partial \mu_k}{\partial \mu_{-A_i^j}} = \prod_{\substack{l=1 \\ l \neq i}}^{n} \mu_{-A_l^{kj}}(x_l) \quad (15)$$

Conclusion

Les réseaux neuro-flous représentent une approche hybride prometteuse dans le domaine de la modélisation et du contrôle des systèmes complexes. Leur capacité à gérer l'incertitude et la vagueness les rend bien adaptés à des domaines tels que la robotique, la prise de décision et la reconnaissance de formes. Cependant, il est essentiel de prendre en compte leurs limites et de poursuivre les recherches afin d'améliorer leur performance et leur applicabilité

Chapitre 05

Partie I. Modélisation et architecture du robot

1. Identification et optimisation du Navigateur flou du robot :

Notre Robot doit suivre une cible Intelligente c'est-à-dire cette cible raisonne aussi par la logique floue ou par les réseaux de neurones ou par les réseaux neuro-flous pour atteindre sont but. Cette cible peut être aussi non intelligente c'est-à-dire elle suivre un trajectoire.

2. fonctionnement du robot

Notre robot se déplace en utilisant la règle générale de la cinématique donnée par la formule mathématique suivante :

$$X(n) = X(n-1) + V.T.\cos(\theta);$$

$$Y(n) = Y(n-1) + V.T.\sin(\theta);$$

Où V est la vitesse de déplacement , T est une période de temps et θ est l'angle de déplacement.

Premièrement le Robot demande des informations sur l'environnement via ses capteurs pour cela notre robot porte trois capteurs :

1. Capteur droit.
2. capteur frontal.
3. Capteur gauche.

Ces capteur on une limite de vision au delà de laquelle il ne peuvent pas détecter d'obstacles.

Ensuite son système de raisonnement peut déterminer et ordonner au système de commande l'action et le mouvement à entreprendre afin d'atteindre l'objectif.

Remarque :

- La distance entre le robot et l'obstacle est donnée par :

$$\text{Distance} = \sqrt{(X_{obstacle} - X_{robot})^2 + (Y_{obstacle} - Y_{robot})^2}$$

- La distance entre l'objectif et l'obstacle est donnée par :

$$\text{Distance } \sqrt{(X_{obstacle} - X_{cible})^2 + (Y_{obstacle} - Y_{cible})^2}$$

- L'angle de vision des capteurs latéraux par rapport au frontal est 45^o

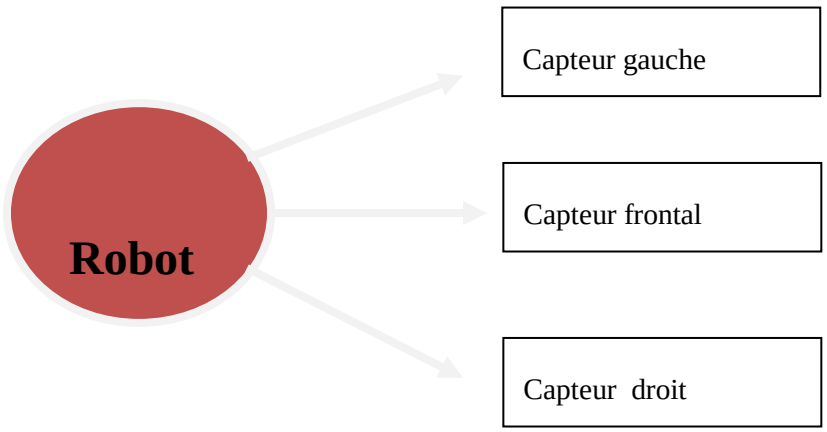


Figure V.1 Architecture du Robot

3. Navigateur neuro-flou réactif du robot

Le navigateur du robot est de type neuro-flou, il a pour tâche de contrôler le déplacement linéaire et angulaire, il possède trois entrées et deux sorties, les entrées sont trois capteurs qui permettent de relever les distances des obstacles situés dans la zone droite, gauche et frontale. Les sorties sont la vitesse et l'angle de braquage du robot (figure V.2).

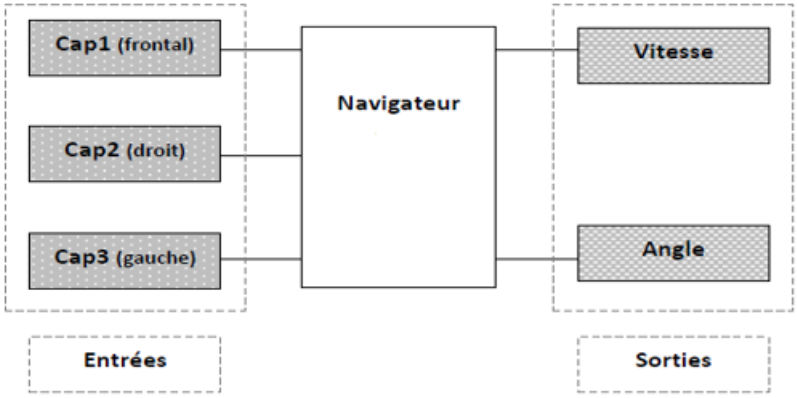


Figure V.2 Navigateur Floue de Robot

Les entrées : les fonctions d'appartenance des entrées de notre navigateur sont des gaussiennes et chaque entrée capteurs est représentée en deux sous-ensembles flous distribuées de manière équidistante sur un intervalle pour toutes les entrées. Ce dernier représente l'intervalle de distance captée par des capteurs.

Ces fonctions permettent d'exprimer des propositions logiques floues du genre « D-Distance captée -Prés », « Distance captée- Loin »

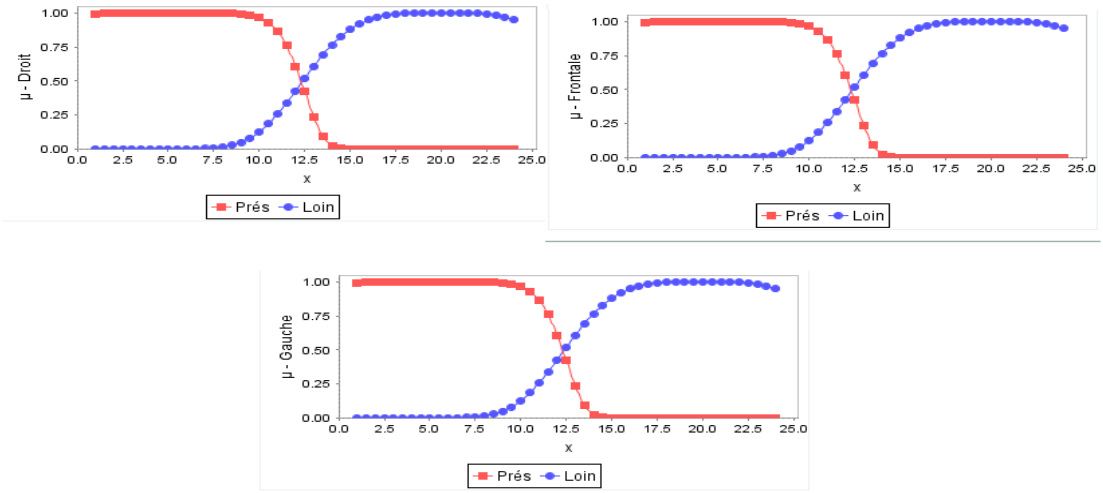


Figure V.3 Les fonctions d'appartenance

Les sorties : Comme nous l'avons déjà vu précédemment, la sortie du navigateur (agrégation et défuzzification des règles floues se fait numériquement (pas de surfaces au niveau des fonctions d'appartenances),

L'angle: La fonction d'appartenance est donnée sous la forme de constante appartenant à l'intervalle [-90 90] °.

Les différents ensembles flous sont caractérisés par des désignations standards, la signification des symboles est indiquée au tableau 1 suivant :

Tableau V.1 :signification des symbols angle

Symbol angle	Signification
N	Négatif
Z	Environ zéro
P	Positif

Négatif pour tourner à gauche, **positive** pour tourner à droite et angle **zéro** pour avancer tout droit.

La vitesse : La fonction d'appartenance est donnée sous la forme de constante.

Tableau V.2 signification des symbols distance

Symbol vitesse	Signification
PV	PETIT VITESSE
MV	MOYENNE VITESSE
GV	GRANDE VITESSE

Base des règles :

La base des règles consiste à déterminer le nombre de règles d'inférence, nous avons trois fonctions d'appartenance (trois capteurs de détection d'obstacle) et chaque fonction est divisée en deux sous ensemble flous (obstacle **Prés** – obstacle **Loin**), on déduit huit règles (il faut s'assurer que chaque sous ensemble de chacune des variables est utilisé au moins une fois) : avec φ=Angle, V=Vitesse, cap1=capteur frontal, cap2= capteur droit, cap3=capteur gauche.

Les règles sont regroupées dans le tableau d'inférence suivant :

Tableau V.3 :tableau d'inférence

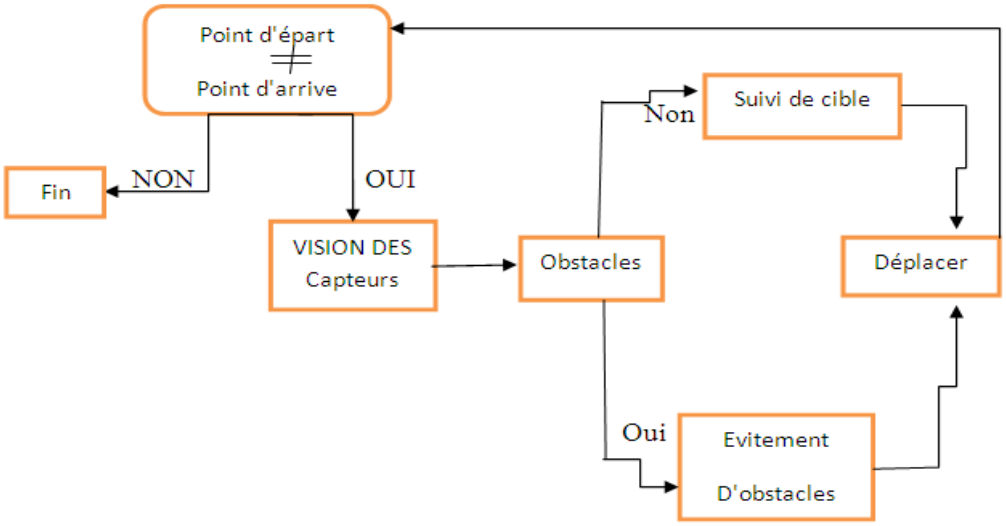
CAPTEUR 1	CAPTEUR 2	CAPTEUR3	VITESSE	ANGLE
Prés	Prés	prés	PV	N
Prés	Prés	loin	PV	P
Prés	Loin	prés	PV	N
Prés	Loin	loin	PV	N
Loin	Prés	prés	MV	Z
Loin	Loin	prés	MV	N
Loin	Prés	loin	MV	P
Loin	Loin	loin	GV	Z

3. Architecture de navigateur

La navigation d'un robot mobile et décomposer en deux actions:

- Evitement d'obstacle.
- Robot 2 Suivi robot 1
- Robot 1 Suivi de cible.

Ce qui concerne l'évitement d'obstacle il est nécessaire d'applique une commande par exemple la commande floue (c'est le cas de notre navigateur) , Cette phase et la plus difficile.



FigureV.4 les actions de robot

Partie II. Implementation et Simulation :

Nous abordons dans ce chapitre notre partie implémentation, qui mettra en pratique la conception de notre système.

Pour ce faire nous présentons d'abord le langage de programmation et l'environnement de développement, puis nous verrons les différents API, les étapes à suivre pour son importation et ensuite nous enchainons par la présentation de notre application et expliquer son fonctionnement.

2. Environnement de développement

2.1 L'environnement matériel

Notre application a été développée sur une machine ACER avec une RAM de 4Go, sous le système d'exploitation Microsoft Windows 10.

2.2 L'environnement logiciel

L'implémentation de l'application a été réalisée avec le langage de programmation java sous la plateforme NetBeans IDE 8.0, à l'aide de différente API que nous décrirons ultérieurement.

2.2.1 Le langage Java

Nous avons utilisé comme langage de programmation le langage objet « java ».Le choix de java se justifie par les avantages suivants :

✓ C'est un langage bien connu et largement répandu. Il existe de nombreuses librairies qui facilitent le développement des applications.

✓ Les applications java s'exécutent en utilisant une machine virtuelle, ce qui les rend indépendantes du système d'exploitation.

✓ Des machines virtuelles java ont été développées pour la plupart des systèmes actuels, ce qui facilite la portabilité des applications java.

✓ Les compilateurs java sont gratuits.

✓ Java permet de définir facilement des interfaces graphiques agréables à utiliser.

2.2.2Environnement Netbeans

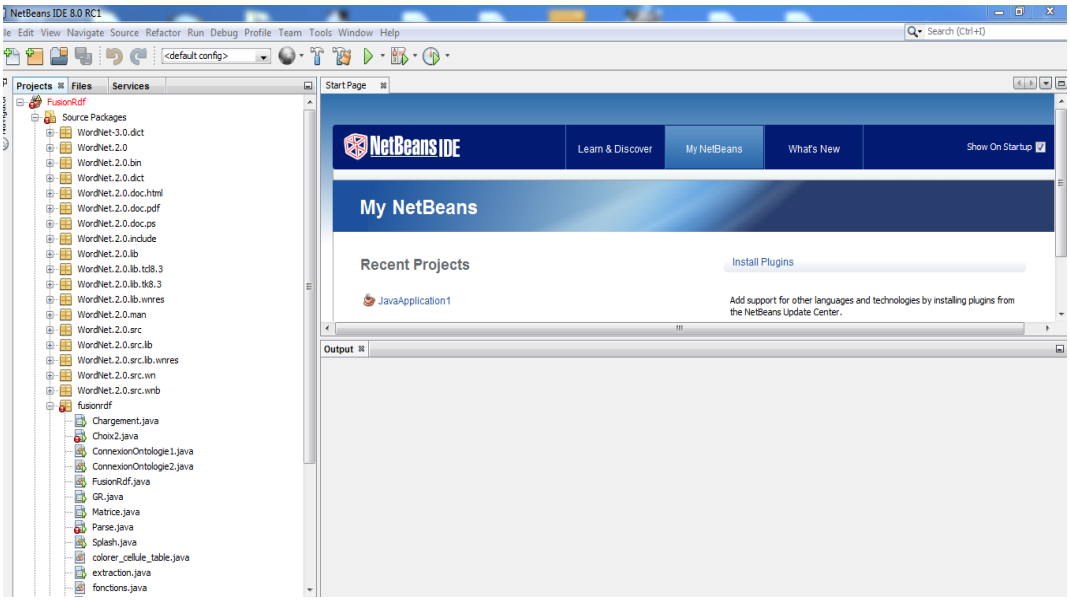


Figure V.5 : NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Développement and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP 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). Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

3. **Déroulement de notre application**

Description de la maquette:

Nous présentons ici notre maquette logicielle de la commande neuro-floue appliqué à la navigation d'un robot mobile dans le suivi de cible dynamique.

3.1 Menu principale

Le menu principal consiste a superviser l'ensemble de notre travail, paramétrage des données, déroulement des algorithme, et affichage des entrée, sorties.

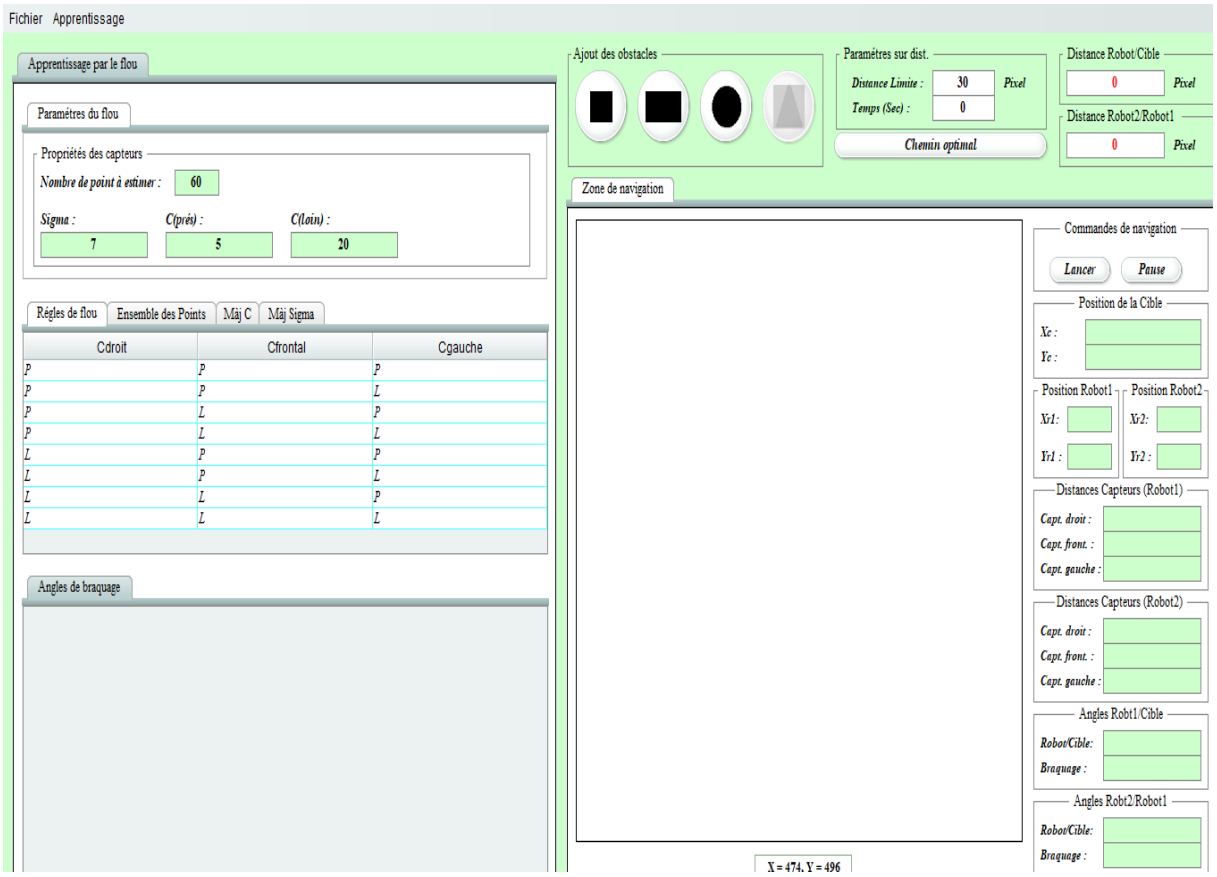


Figure V.6 : Menu principal de l'application

3. 2 Type des obstacles

On a utilisé des obstacles simple, pour ce faire on a choisir trois types d'obstacles qui sont :

1- Rectangles.

2- Cercle

On dessine les obstacles avec la souris.

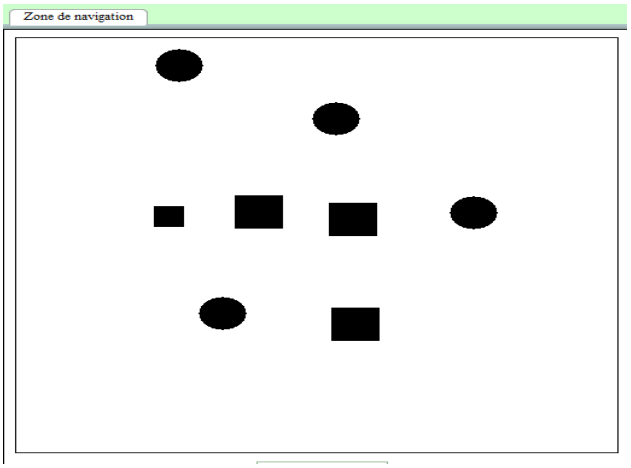


Figure V.7 : Creation des obstacles .

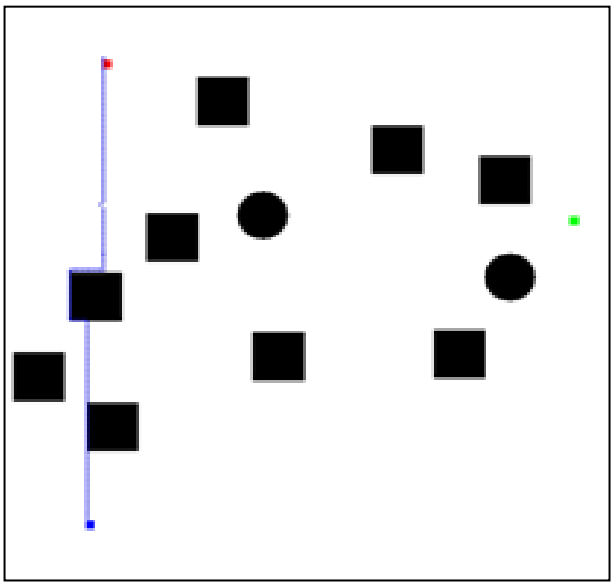
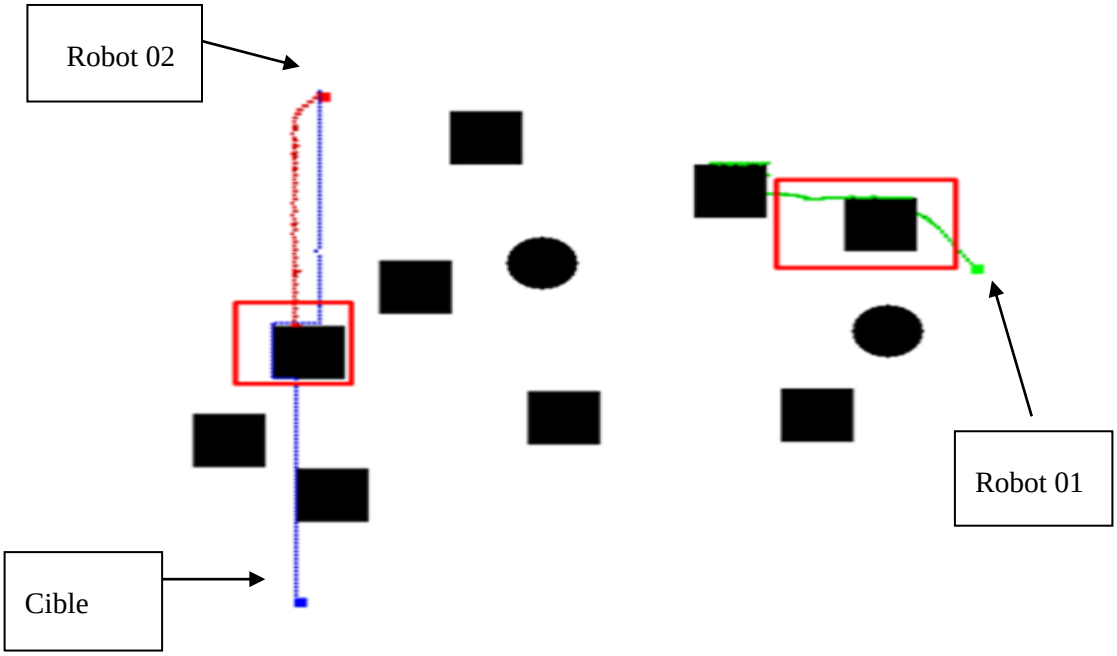
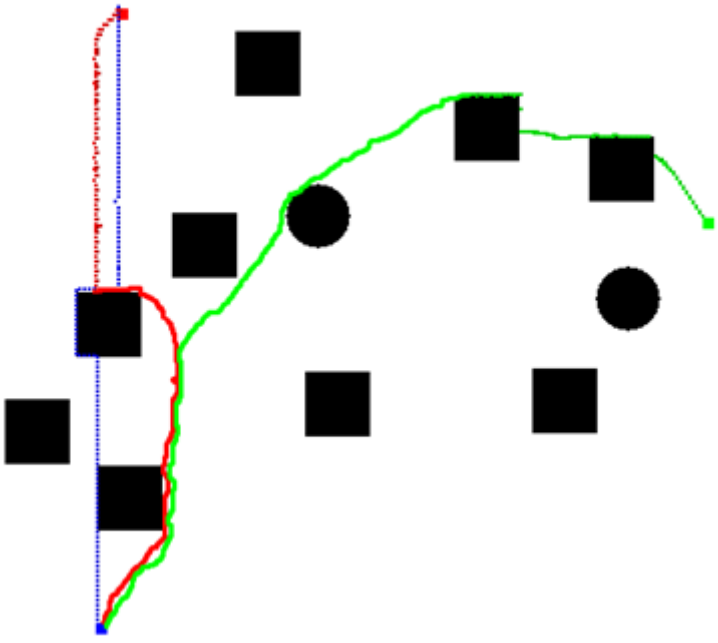


Figure V.8 : Chemin optimal avant l'application du Flou

Le robot 1 doit donc défier son environnement en évitant les obstacles et en se dirigeant vers un point précis dans son espace de navigation (Cible) et le robot 2 faire le même travaille avec robot 1



FigureV.9 Navigation des Robot 1 et 2



FigureV.10 La cible est atteinte par le robot en évitant tout les obstacles.

4. Conclusion

Le travail que nous avons réalisé dans ce mémoire propose une stratégie de navigation réactive floue d'un robot mobile dans un environnement inconnu. Le robot 1 doit donc défier son environnement en évitant les obstacles et en se dirigeant vers un point précis dans son espace de navigation et le robot 2 faire le même travaille avec robot 1

Conclusion Générale

Dans ce travail, le problème de la navigation autonome d'un robot mobile a été abordé en utilisant les techniques floues. La tâche de base que doit réaliser chaque robot mobile avec un minimum d'erreur est d'atteindre la configuration d'arrivée a partir d'une configuration de départ sans collision avec les obstacles et sans intervention humaine. On autilisé des contrôleurs neuro-flous.

Le travail présenté dans ce mémoire est divisé en deux parties:

Dans une première partie, nous avons présenté un aperçu sur le domaine de la robotique mobile et les concepts de base des approches utilisées à savoir; la commande floue.

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

Dans la deuxième partie, les contributions de ce travail sont présentées:

Nous avons exposés la théorie de la commande par le neuro-floue et l'architecture de base d'un contrôleur neuro-flou. Le fonctionnement d'un contrôleur neuro-flou dépend d'un nombre important de paramètres qu'il faut déterminer afin d'optimiser ses performances. Les contrôleurs neuro-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.

1. ARTI - Autonomous Robot Technology GmbH , Jan 15, 2021.
2. *Merriam-Webster Online Dictionary and Thesaurus*: http://www.merriamwebster.com/dictionary/robot.

3. CNRL: http://www.cnrtl.fr/. *Centre National de Ressources Textuelles et Lexicales*. http://www.cnrtl.fr/lexicographie/robot.

4. *Dictionnaire Hachette*. Hachette éducation Edition 2013. ISBN 978-2-01-281493-6
5. Robotics l'International Fédération of World Robotics, Editions 2006.

6. Y. Cang, N. H. C. Yung, D. Wang, " A Fuzzy Controller with Supervised Leaning 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.

7. B. Bayle, "Robotique Mobile", Ecole Nationale Supérieure de Physique de Strasbourg, 2006.

8. H.Joachim et R. Joachim, localization of a mobile robot by matching 3D laser-rangeimage and preicated sensor imges, October 24-26, Proceedings of the vehicules, 1994.

9. A.Oualid Djekoune , Localisation Et Guidage Du Robot Mobile Atrv2 Dans Un Environnement Naturel. Thèse de Doctorat, Université des Sciences et de la technologie Houari Boumediene, 2010.

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

11. E. Laurin, "Système Intelligent d'Assistance à la Perception dans la Conduite de Véhicule", Thèse de Doctorat, Université de Sherbrooke, 2000.

12. J.Borenstein ,"Internalcorrectionofdead-reckoningerrorswithadual-drive compliant linkage mobile robot" , Journal of Robotic systems , 1995.

13. H. Makela, K. Koskinen"Navigation of outdoor mobile robots using dead reckoning and visually detected landmarks" — 5th Int. Conf. on Advanced Robotics ICAR'9l, Pisa, Italy, June 1991.

14. G.Frappier,"Systèmeinertielsdenavigationpourrobotsmobiles"Séminaire"Lesrobotsmobiles",EC2,Paris,1990.
15. CHOLLET Francois. Deeplearning with python.2017.

16. ZACCONEGiancarlo,MDREZAULKarim,MENSHAWYAhmed.Deeplearningwithtensorflow. 2017.

17. DJOKHRAB Alaeddine.Planificationetoptimisation de trajectoire d'un robotmanipulateurà 6 ddl par des techniques neuro_floues 2015.
18. PARIZEAU Marc. Réseauxdeneurones.2004.

19. Fatmi, A.AlYahmadi, L.Khriji, and N. Masmoudi, ‹c4 *FuzzyLogic Based Navigation of a Mobile Robot*» International Journal of Applied Mathematics and Computer Sciences 1;2 , Spring 2005.

20. Kaufmann, "Introduction à la logique floue", Techniques de l'Ingénieur, traité Informatique

industrielle", Aout 2001.

21. Cirstea. M. N, A. Dinu, J.G. Khor, M. McCormick. Neural and Fuzzy Logic Control of Drives and Power Systems. 2002.

22. H. R. Beom and H. S. Cho, "A Sensor-based Navigation for a Mobile Robot using Fuzzy Logic and Reinforcement Learning", IEEE Transactions on S ystems, Man, and Cyberneiics, 1995.

23. Web, N. Douki, "Commande Floue Adaptative par Mode Glissant d'une Classe des Systèmes", mémoire de master, Univ-Ferhat Abbas Sétif, 06/2011.

24. Pierre GABRIEL, 2000-2001 / cours: « Introduction à la logique floue et à la commande floue».http://ace.montefiore.ulg.ac.be/elap/documents/ELEN032/Fuzzy_2.pdf.

25. http://theses.univbatna.dz/index.php?option=com_docman&task=doc_download&gid=180&Itemid=1.

26. Michio Sugeno and Takahiro Yasukawa. A Fuzzy-Logic-Based Approach to Qualitative Modeling. School of the Tokyo Institute of Technology, Yokohama, Japan. Tokyo Institute of Technology, Japan. (IEEE TRANSACTIONS ON FUZZY SYSTEMS,1993).

27. Tomohiro Takagi and Michio Sugeno. Fuzzy Identification of Systems and Its Applications to Modeling and Control. School of the Tokyo Institute of Technology, Yokohama, Japan. (IEEE transactions on systems, MAN, and cybernetics, 1985).

28. Jang, J.S.R. , ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Systems Man & Cybernetics,1993

Résume

La navigation autonome de robots dans des environnements inconnus présente des défis complexes en raison de l'absence de connaissances préalables sur l'environnement. Dans cette étude, un système de navigation basé sur ANFIS est développé pour permettre à un robot d'explorer et de naviguer efficacement dans un tel environnement. Le système ANFIS combine des techniques d'inférence floue et de réseaux de neurones pour modéliser les comportements de navigation du robot. Des capteurs embarqués sont utilisés pour collecter des données sur l'environnement, telles que la distance par rapport aux obstacles et la direction de déplacement. Ces données sont ensuite utilisées pour entraîner le système ANFIS à prendre des décisions de navigation appropriées.

En conclusion, cette étude démontre l'efficacité du système neuro-flou ANFIS pour la navigation autonome de robots dans des environnements inconnus. Ces résultats pourraient contribuer au développement de systèmes de navigation plus avancés pour les robots autonomes opérant dans des conditions réelles

Mots clé : anfis , robot , floue,navigue , autonome

Autonomous navigation of robots in unknown environments poses complex challenges due to the lack of prior knowledge about the environment. In this study, an ANFIS-based navigation system is developed to enable a robot to effectively explore and navigate in such an environment. The ANFIS system combines fuzzy inference techniques and neural networks to model the robot's navigation behaviors. Embedded sensors are used to collect data about the environment, such as distance to obstacles and direction of movement. This data is then utilized to train the ANFIS system to make appropriate navigation decisions.

In conclusion, this study demonstrates the effectiveness of the ANFIS neuro-fuzzy system for autonomous navigation of robots in unknown environments. These findings could contribute to the development of more advanced navigation systems for autonomous robots operating in real-world conditions

Key words : anfis , robot , floue,navigue, autonome

يمثل التنقل الذاتي للروبوت في بيئات غير مألوفة تحديات معقدة بسبب نقص المعرفة المسبقة بالبيئة. في هذه الدراسة ، تم تطوير نظام ملاحة قائم على ANFIS لتمكين الروبوت من الاستكشاف والتنقل بكفاءة في مثل هذه البيئة. يجمع نظام ANFIS بين الاستدلال الغامض وتقنيات الشبكة العصبية لنمذجة سلوكيات تنقل الروبوت. تُستخدم المستشعرات الموجودة على متن الطائرة لجمع البيانات البيئية ، مثل المسافة إلى العوائق واتجاه السفر. ثم يتم استخدام هذه البيانات لتدريب نظام ANFIS على اتخاذ قرارات التنقل المناسبة.

في الختام ، توضح هذه الدراسة فعالية النظام العصبي الغامض ANFIS للتنقل المستقل للروبوتات في بيئات غير معروفة. يمكن أن تساهم هذه النتائج في تطوير أنظمة ملاحة أكثر تقدمًا للروبوتات المستقلة التي تعمل في ظروف العالم الحقيقي.

الكلمات المفتاحية : الذاتي ، ملاحة ، الربوت ، الغامض