

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE**

**UNIVERSITE FERHAT ABBAS - SETIF
UFAS (ALGERIE)**

MEMOIRE

Présenté à la Faculté de Technologie

Département d'Electronique

Pour l'Obtention du Diplôme de

MAGISTER

Option : Contrôle

Par

Mme : Benmechri ratima

Thème

CONTROLE D'UN SYSTEME FLOU DE TYPE 2 APPLIQUE A UN ROBOT MOBILE

Soutenu le : 14/12/2011 devant la commission d'examen :

Pr : Djahli Farid	Université de Sétif	Président
Dr : Bartil Arrés	Université de Sétif	Rapporteur
Pr : Krim Fateh	Université de Sétif	Examineur
Dr : Hassam Abdelwahab	Université de Sétif	Examineur
Dr : Boukezzoula Naceur-eddine	Université de Sétif	Examineur

A mes très chers parents,
mon marié
mes garçons mohaned hecéne et housiène
mes frères et sœurs,
tous mes amis,
ceux qui m'ont aidé,
ceux qui estiment le savoir

Je dédie ce mémoire

REMERCIEMENTS

Ce travail a été effectué au département d'Electronique de l'Université Ferhat Abbas de Sétif dans le cadre des activités du laboratoire Systèmes Intelligents.

Je remercie Monsieur le Professeur K.Benmmahamad de m'avoir proposé le sujet.

J'exprime mes sincères remerciements à mon rapporteur Dr Bartil Arres, Maitre de conférences à L'Université Ferhat Abbas de Sétif. Il a fait preuve d'une grande disponibilité, efficacité et de précieux conseils.

J'exprime mes sincères remerciements aux membres de jury de soutenance de ce mémoire :

Professeur F.Djahli de l'Université Ferhat Abbas de Sétif pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Professeur F.Krim de l'Université Ferhat Abbas de Sétif pour l'honneur qu'il me fait en acceptant de participer au jury de ce mémoire.

Dr A.Hassam pour l'honneur qu'il me fait également en participant au jury de ce mémoire

Dr N.Boukezzoula pour l'honneur qu'il me fait par sa participation au jury de ce mémoire

J'associe à ces remerciements, tous les enseignants du département d'Electronique ainsi qu'à toutes les personnes ayant contribué par leur soutien , conseils et leur disponibilité durant la préparation de ce mémoire.

SOMMAIRE

INTRODUCTION GENERALE

01

CHAPITRE I

GENERALITE SUR LA ROBOTIQUE MOBILE

MOBILEE

I.1. Introduction.....	04
I.2. Les différents types de navigation.....	05
I. 2.1. Les stratégies de navigation.....	05
I.2.1.1 Approche d'un but.....	05
I.2.1.2 Guidage.....	05
I.2.1.3 Action associe à un lieu	06
I.2.1.4 Navigation topologique.....	07
I.2.1.5 Navigation métrique	07
I. 2.2. Les architectures de contrôle.....	09
I.2. 2. 1. Contrôleurs Hiérarchiques.....	10
I.2. 2. 2. Contrôleurs réactifs.....	11
I.2. 2. 3 Contrôleurs hybrides.....	13
I.3. Véhicule holonome et non holonomes.....	13
I. 3.1. Les véhicules non holonomes	13
I. 3.2. Les véhicules holonomes	14
I. 3.3. Le concept de chemin d'un robot.....	16
I. 3.4. Commandabilité d'un robot mobile.....	16
I. 3.5. Modélisation des robots mobiles	17
I. 3.5.1 Le robot à deux roues motrices	18
I. 3.5.2 Le robot type voiture.....	19
I. 3.6. Commande des robots mobiles à roues.....	20
I.3.6.1 Suivi de chemin.....	22

I.3.6.2 Stabilisation de Trajectoire.....	22
I.3.6.3 Stabilisation de Configuration Fixes.....	22
I. 3.7 Modèles de commande.....	23
I.3.7.1 Robot uni cycle.....	23
I.3.7.2 Robot de type voiture.....	24
I. 3.8. Poursuite d'un véhicule de référence.....	25
I. 3.9. Modèle de coordonnée d'un chemin.....	27
La forme enchaîner.....	28

CHAPITRE II

LOGIQUE FLOUE ET CONTROLE

II.1. INTRODUCTION.....	30
II.2. les Concepts.....	32
II.2.1 Sous ensembles flous.....	32
II.2.2 Variable linguistique.....	34
II. 3. Architecture classique d'un contrôleur flou (FLC).....	39
II. 3.1. Fuzzification.....	40
II. 3.2. Base de connaissance.....	41
II. 3.3. Inférence.....	42
II.3.4. Défuzzification.....	43
II.4. Les Modèles Floues.....	46
II. 4.1 Le Modèle flou de MAMDANI.....	46
II. 4.2 Le modèle floue de Takigi – sagino.....	47
II.5. Applications en robotique mobile.....	48
II.5.1. La voiture floue de Sugeno.....	49
II. 5.2. Le métro de Sandai.....	49
II. 5.3. Autres applications.....	50
II.6. Caractéristique Avantage et Limitation de la logique floue.....	50
II.7. CONCLUSION.....	51

CHAPITRE III

SYSTEME FLOUE TYPE -2

III.1.	Introduction.....	52
III.2.	Les ensembles flous type_2.....	53
III.2. 1.	Définitions.....	53
III.2. 2.	Exemples des ensembles flous type-2.....	55
III 2. 3.	Opération sur les ensembles flous type_2	59
III.2.4.	Type d'ensemble floue type 2.....	62
III.3.	Structure générale d'un système floue type_2.....	64
III.3. 1.	Fuzzification.....	64
III.3. 2.	Base de Règle.....	65
III.3. 3.	Le Moteur d'inférence floue.....	65
III.3. 4.	Réduction de type.....	67
III.3. 4. 1.	Calcule de centres de conséquences de règle.....	68
III.3. 4. 2.	Calcule de l'ensemble type réduit.....	71
III.3. 5.	Défuzzification.....	73
III.4.	CONCLUSION.....	78

CHAPITRE IV

APPLICATION ET RESULTATS DE SIMULATION

IV.1.	Introduction.....	79
IV.2.	Notion de sous but.....	80
IV.3.	simulation.....	81
IV.3.1	Le chemin.....	83
IV.3.2	Calcule de l'erreur actuelle.....	84
IV.3.3	calcule de l'angle Θ_p	87

IV.3.4 Le modèle de véhicule.....	87
IV.4. Les contrôleurs flous.....	90
IV.4.1. Simulation utilisant le Contrôleur floue type 1.....	91
IV.4.2. Résultats de simulation contrôleur floue type 1.....	93
IV. 4.3 Simulation utilisant un Contrôleur floue type 2.....	98
IV.4.4. Résultats de simulation contrôleur floue type 2.....	101
IV.5. Interprétation des résultats.....	104
IV.6 Conclusion	104
Conclusion Générale.....	106

Bibliographiques

Résumé

INTRODUCTION GENERALE

Les robots mobiles autonomes dirigeants dans un environnement structuré et non structuré doivent être capables d'être utilisés dans des conditions d'imprécision et d'incertitude présentes dans de tels environnements.

Le choix des méthodes adéquates pour modéliser et manipuler de telles incertitudes est crucial pour un robot mobile dirigé dans un environnement non structuré.

Le contrôleur à base de logique floue (FLC) est une meilleure méthodologie pour concevoir les contrôleurs robustes, qui sont capables de fournir une exécution satisfaisante en face de l'incertitude et l'imprécision. En conséquence, le FLC est devenu ces dernières années une approche populaire à la commande d'un robot mobile.

Il y a beaucoup de sources d'incertitude qui font face au FLC pour un robot mobile dirigé dans un environnement. Nous en citons certains d'entre eux comme suit :

- Incertitude dans les entrées du FLC qui se traduisent par des incertitudes dans les antécédents, de la fonction d'appartenance du moment que les mesures de sondes sont en général bruyantes et sont affectées par les conditions d'observation: Leurs caractéristiques sont changées par les conditions environnementales telles que le vent, soleil, humidité; pluie; etc.).
- Incertitude dans les actions de commande qui sont traduites par des incertitudes dans les sorties de la fonction d'appartenance du FLC. De telles incertitudes peuvent résulter à partir de changement de caractéristiques des déclencheurs qui peuvent être dues à la contradiction du terrain ou en raison de changements environnementaux.
- Incertitude linguistique comme étant la signification de mots utilisés dans les étiquettes linguistiques de l'antécédent et les conséquences peuvent être incertaine tandis que les mots sont interprétés différemment.

En outre, les experts ne conviennent pas toujours et ils fournissent différentes conséquences pour les mêmes antécédents. Un aperçu des experts mènera habituellement à un histogramme des possibilités pour la conséquence d'une règle, cet histogramme représente l'incertitude à la conséquence d'une règle. Ces incertitudes posent des difficultés pour déterminer l'antécédent et la conséquence exacte et précise de la fonction d'appartenance floue.

Toutes les réalisations du FLC dans la commande de robots sont basées sur le FLC type 1 traditionnel. Il y a différentes manières pour construire le FLC type 1 pour le robot mobile ; La manière la plus commune est de construire le FLC en obtenant les règles floues et les (entrées – sorties) de la fonction d'appartenance basée sur la connaissance d'expert ou par l'observation des actions d'opérateur humain commandant le robot mobile.

Pour le FLC type 1 conçu en utilisant l'expérience humaine ou les mécanismes d'apprentissage, on emploie les ensembles flous type 1 qui manipulent les incertitudes liées aux entrées et sorties utilisant la fonction d'appartenance précise et crisp que l'utilisateur prend pour la capture des incertitudes. Une fois que les fonctions d'appartenance de type 1 ont été choisies, toutes les incertitudes disparaissent, parce que la fonction d'appartenance est totalement précise.

Un ensemble flou type 2 est caractérisé par une fonction d'appartenance floue, c'est-à-dire la valeur d'appartenance (ou le degré d'appartenance) pour chaque élément de cet ensemble est un ensemble flou dans $[0, 1]$, différent de l'ensemble flou type 1 ou le degré d'appartenance est des crisp numéros dans $[0, 1]$. Les fonctions d'appartenance des ensembles flous type 2 sont tridimensionnelles et incluent une empreinte de pas (footprint) de l'incertitude. C'est la nouvelle troisième dimension d'ensemble flou type 2 et l'empreinte de pas de l'incertitude qui fournissent le degré de liberté additionnel qui permettent de modéliser et exécuter l'incertitude directement. Les ensembles flous type 2 sont utiles mais il est difficile de déterminer la fonction d'appartenance exacte et précise (c'est le cas d'un robot dans un environnement non structuré).

Le FLC type 2 utilisé dans notre travail emploie des ensembles flous type 2 pour représenter les entrées et les sorties du FLC. Employer un FLC (type1 ou type 2) pour la commande d'un robot mobile (suivi d'un chemin).

Le présent travail est composé et organisé en quatre chapitres comme suit :

Le premier chapitre est consacré à la robotique mobile d'une manière générale. Il décrit les différents types de navigation et leurs stratégies (navigation topologique et métrique) ainsi que les architectures de control.

Le second chapitre présente la théorie de la logique floue (type1) et le control flou. Les différents concepts de bases sont revus. Nous donnons dans ce chapitre l'architecture classique d'un contrôleur flou (FLC) et les parties qui le composent. Les différentes méthodes de défuzification ont été décrites.

Le troisième chapitre donne les définitions de base des ensembles flous type2 ainsi que les différents concepts du système flou type2.

Dans le dernier chapitre, nous présentons tous les résultats obtenus lors de nos différents travaux de simulation et nous mettons en évidence l'avantage de la méthode utilisée (Flou type2) par rapport au FLC de type1.

Chapitre I

Généralités sur la robotique mobile

I.1 Introduction

Depuis une vingtaine d'années, un effort particulier a été fait dans les domaines de la recherche et de l'industrie pour construire des robots mobiles évoluant avec un minimum d'intervention humaine.

Une première génération des robots a consisté en des machines capables d'évoluer dans des environnements parfaitement connus: celles-ci réalisent des missions planifiées à partir d'une modélisation complète de l'environnement (laboratoire) ou se contentent de suivre une trajectoire par un mécanisme de filo-guidage (robot de manutention). Le point commun des ces robots est qu'ils évoluent dans un environnement qui leur est totalement dédié. Cependant, lorsque l'environnement devient plus complexe (c à d : partiellement connu, dynamique, .etc.), il apparaît indispensable que le robot soit doté de capacités décisionnelles aptes à le faire réagir aux aléas qui peuvent contrarier ses mouvements (pannes partielles, obstacles imprévus). Cela peut être le cas lorsque le robot mobile évolue dans des environnements hostiles à l'homme (milieu radioactif) ou trop éloignés (exploration spatiale). Pour cela, le robot doit suivre le schéma correspondant au paradigme (Percevoir, Décider, Agir) au sien d'une architecture de contrôle. L'activité d'un tel robot se ramène aux tâches énoncées ci après:

- **Percevoir:**

Le robot doit acquérir des informations sur l'environnement dans lequel il évolue par l'intermédiaire de capteurs ; Ces informations permettent de mettre à jour un modèle de l'environnement (architecture hiérarchique), ou peuvent être directement utilisées comme entrées de comportements de bas niveau (architectures purement réactives).

- **Décider:**

Le robot doit définir des séquences d'actions résultant d'un raisonnement appliqué sur un modèle de l'environnement (architecture hiérarchique) ou répondant de manière réflexe à des stimuli étroitement liés aux capteurs (architectures purement réactives).

- **Agir:** Le robot doit enfin exécuter les séquences d'actions en envoyant des consignes aux actionneurs par l'intermédiaire des asservissements.

Au sien de la partie décision, il est nécessaire de contrôler l'exécution de ces actions afin que le robot s'adapte rapidement à des événements imprévus.

I.2 Différents types de navigation

I.2.1 Stratégies de navigation

Les stratégies de navigation permettant à un robot mobile de se déplacer pour rejoindre un but sont extrêmement diverses, de même que les classifications qui peuvent en être faites. Afin de situer ce type de navigation dans son contexte général, nous reprenons ici une classification établie par Trullier et al. [2]. Cette classification présente l'avantage de distinguer les stratégies sans modèles internes et les stratégies avec modèle interne. Cette classification comporte cinq catégories, de la plus simple à la plus complexe :

I.2.1.1 Approche d'un objet :

Cette capacité de base permet de se diriger vers un objet visible depuis la position courante du robot. Elle est en général réalisée par une remontée de gradient basée sur la perception de l'objet, comme dans l'exemple célèbre des véhicules de Valentino Braitenberg qui utilisent deux capteurs de lumière pour atteindre ou fuir une source lumineuse. Cette stratégie utilise des actions réflexes, dans lesquelles chaque perception est directement associée à une action. C'est une stratégie locale, c'est-à-dire fonctionnelle uniquement dans la zone de l'environnement pour laquelle le but est visible.

I.2.1.2 Guidage :

Cette capacité permet d'atteindre un but qui n'est pas un objet matériel directement visible, mais un point de l'espace caractérisé par la configuration spatiale d'un ensemble d'objets remarquables, qui l'entourent ou qui en sont plus proches (voisins). La stratégie de navigation, utilisant souvent une descente de gradient également, consiste alors à se diriger dans la direction qui permet de reproduire cette configuration. Cette capacité utilisée par certains insectes, comme les abeilles, a été appliquée sur divers robots. Cette stratégie utilise également des actions réflexes et

réalise une navigation locale qui requiert que les amers caractérisant le but soient visibles.

I.2.1.3 Action associée à un lieu :

Cette capacité est la première réalisant une navigation globale, c'est-à-dire qui permette de rejoindre un but depuis des positions pour lesquelles ce but ou les amers qui caractérisent son emplacement sont invisibles. Elle requiert une représentation interne de l'environnement qui consiste à définir des lieux comme des zones de l'espace dans lesquelles les perceptions restent similaires, et à associer une action à effectuer à chacun de ces lieux (figure I.1). L'enchaînement des actions associées à chacun des lieux reconnus définit une route qui permet de rejoindre le but. Ces modèles permettent donc une autonomie plus importante mais sont limités à un but fixé. Une route qui permet de rejoindre un but ne pourra en effet pas être utilisée pour rejoindre un but différent. Changer de but entraînera l'apprentissage d'une nouvelle route, indépendante des routes permettant de rejoindre les autres buts.

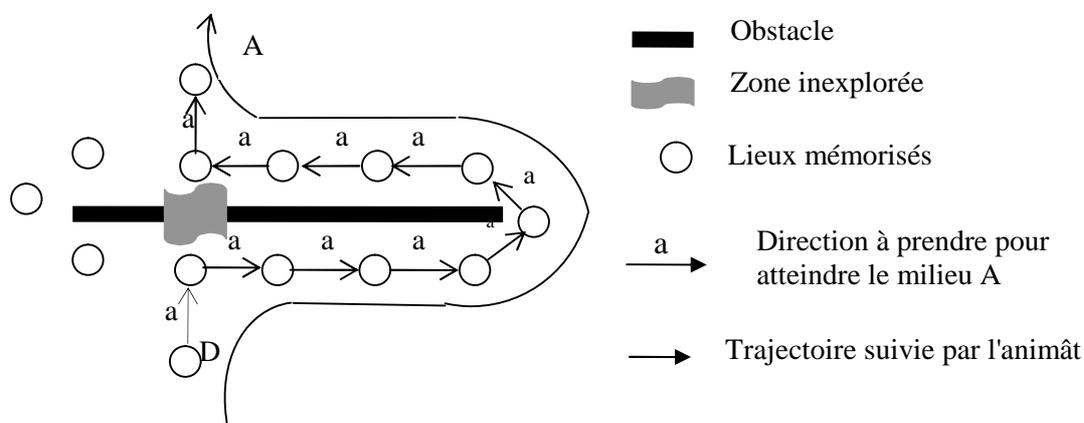


Fig.I.1: Action associée à un lieu

En chaque lieu, représenté par un cercle, l'action à accomplir pour rejoindre le but (A) est représentée par une flèche indiquant la direction à suivre à partir de ce lieu. Cette stratégie permet de rejoindre un but distant dans l'environnement mais repose sur des chemins figés. Dans cet exemple, le chemin joignant le lieu (D) au lieu (A) et passant par la droite de l'obstacle a été appris. Rejoindre le lieu (A) depuis le lieu (D) ne pourra alors

être réalisé que par ce chemin. Le raccourci empruntant le chemin de gauche, par exemple, est inutilisable

I.2.1.4 Navigation topologique :

Cette capacité est une extension de la précédente qui mémorise dans le modèle interne les relations spatiales entre les différents lieux. Ces relations indiquent la possibilité de se déplacer d'un lieu à un autre, mais ne sont plus associées à un but particulier. Ainsi le modèle interne est un graphe qui permet de calculer différents chemins entre deux lieux arbitraires. Ce modèle ne permet toutefois que la planification de déplacements parmi les lieux connus et suivant les chemins connus (figure I.2).

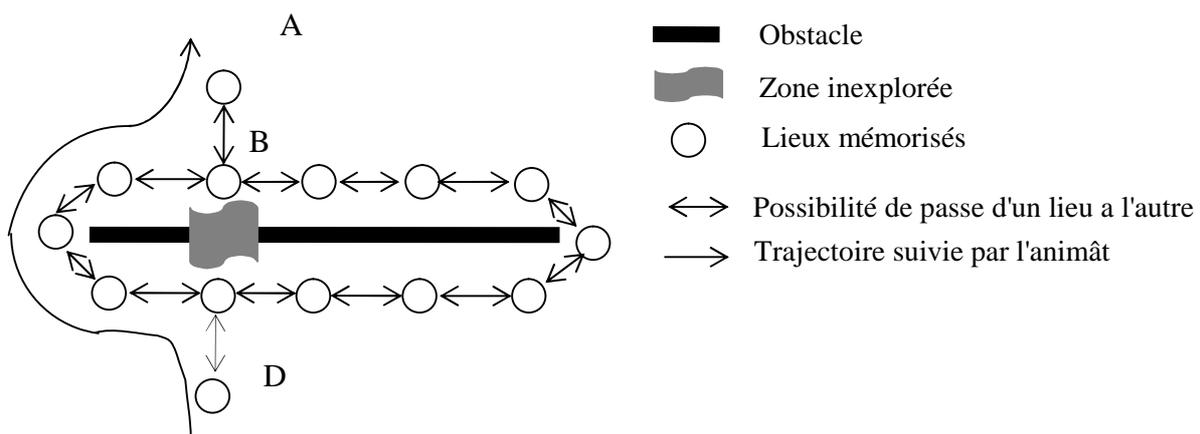


Fig. I.2 : Navigation topologique.

Cette stratégie permet de mémoriser un ensemble de lieux et les possibilités de passer de l'un à l'autre, indépendamment de tout but. Pour rejoindre un but, il faut alors une étape de planification qui permet de rechercher, parmi tous les chemins possibles, le chemin rejoignant le but. Dans notre exemple, le chemin le plus court entre (D) et (A) peut alors être calculé, mais uniquement parmi les lieux et les chemins déjà connus. Cette stratégie permet, par exemple, de contourner l'obstacle par la gauche mais ne permet pas de le traverser en ligne droite de (D) à (A)

I.2.1.5 Navigation métrique :

Cette capacité est une extension de la précédente car elle permet au robot de planifier des chemins au sein de zones inexplorées de son environnement. Elle mémorise pour cela les positions métriques relatives des différents lieux, en plus de la possibilité de passer de l'un à l'autre. Ces positions relatives permettent, par simple composition de vecteurs, de calculer une trajectoire allant d'un lieu à un autre, même si la possibilité de ce déplacement n'a pas été mémorisée sous forme d'un lien (Figure I. 3).

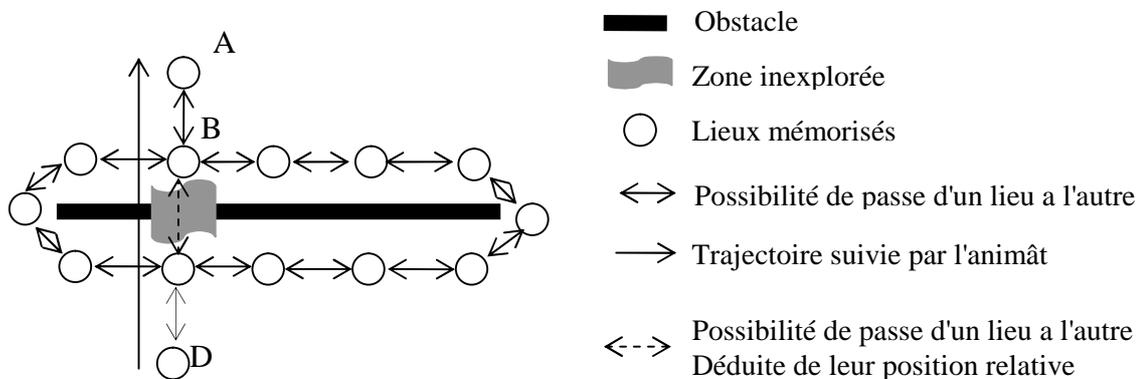


Fig. I.3 : Navigation métrique.

Cette stratégie permet de calculer le chemin le plus court entre deux lieux mémorisés, permettant même de planifier des raccourcis au sein de zones inexplorées de l'environnement. Pour cela, la carte mémorise la position métrique relative de chacun des lieux visités par le robot. Ainsi il est possible de prévoir un déplacement entre deux lieux, même si la possibilité de ce déplacement n'est pas enregistrée dans la carte. Dans cet exemple, cette stratégie permet d'aller du lieu (A) au lieu (D) en traversant la zone inexplorée.

Les modèles des trois premières catégories utilisent des actions réflexes pour guider le robot et se différencient essentiellement par le type de perceptions utilisées pour déclencher ces actions. Ils se regroupent sous le terme générique de navigation réactive dont nous

parlerons dans le deuxième chapitre. Ils peuvent être très simple, ils ne nécessitent pas de modèle global de l'environnement mais ont un domaine d'application souvent restreint. Dans le monde vivant, ces stratégies sont très répandues, notamment chez les insectes. Les comportements de ce type restent toutefois essentiels dans les robots modernes car, du fait de leur simplicité, ils sont généralement exécutés très rapidement et ils permettent de réaliser des tâches de bas niveau, comme l'évitement des obstacles imprévus, essentielles à la sécurité d'un robot.

Les modèles des deux dernières catégories autorisent pour leur part une navigation globale et permettent de rejoindre un but arbitraire au sein de l'environnement. Ils s'appuient pour cela sur un modèle interne du monde, une carte, qui supporte une planification. Ce modèle interne mémorise donc la structure spatiale de l'environnement, indépendamment d'un but précis. Chacune des positions mémorisées dans ce modèle interne peut alors être utilisée comme but par le processus de planification dont le rôle est de calculer une route vers ce but. Ce sont ces deux stratégies qui sont regroupées sous le terme de navigation par carte. Une telle représentation interne est naturelle pour les êtres humains, pour lesquels des processus cognitifs de haut niveau sont utilisés pour créer et utiliser une carte. Ces processus de haut niveau sont toutefois très difficiles à copier pour un robot réel qui ne dispose que de systèmes rudimentaires de perception et de traitement des informations en comparaison avec un homme.

I.2.2 Les architectures de contrôle

Ces architectures peuvent être classées en trois grandes catégories que nous détaillerons par la suite [1][2] : les contrôleurs hiérarchiques, les contrôleurs réactifs et les contrôleurs hybrides (Figure I.4). Toutes ces architectures ne diffèrent pas forcément par les méthodes élémentaires employées mais plutôt par leur agencement et leurs relations.

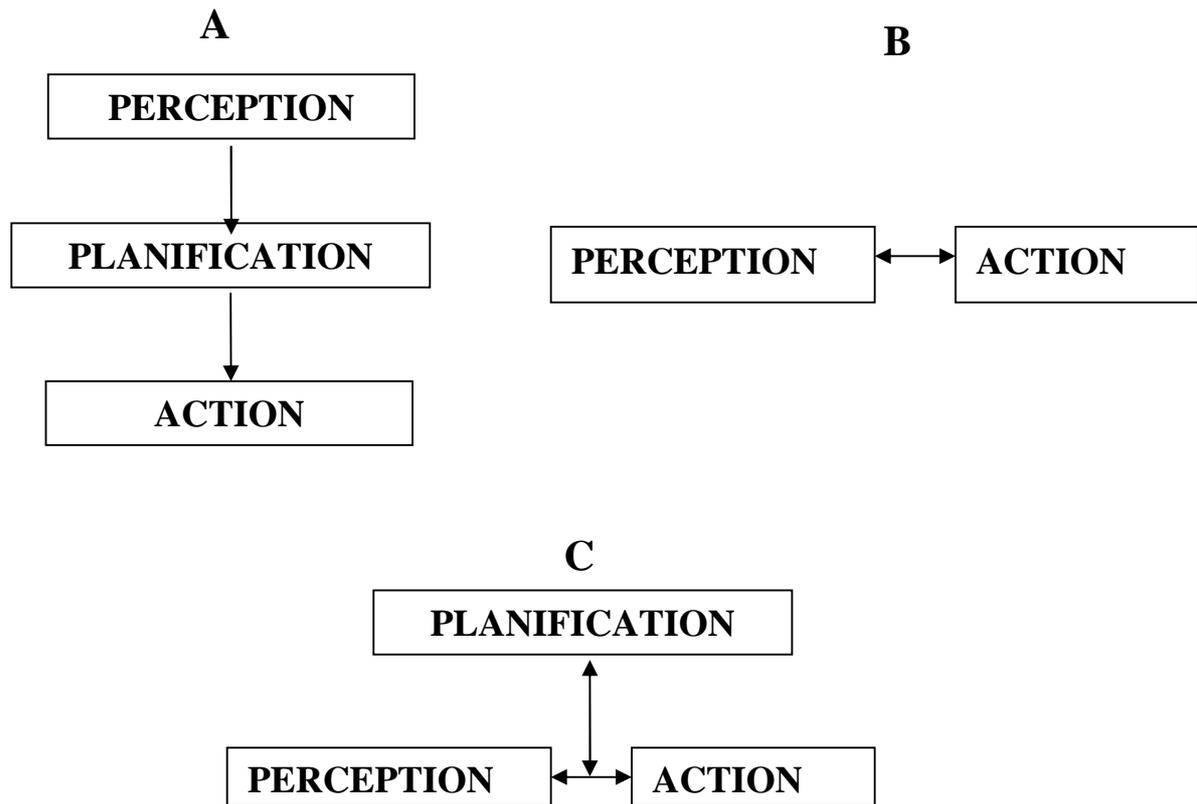


Fig.I.4: Illustration des architectures classiques des contrôleurs pour les robots mobiles : Hiérarchique (A), Réactive (B) et Hybride (C).

I.2.2.1 Contrôleurs Hiérarchiques

Un robot doit commencer par traiter toutes les données recueillies par ses capteurs concernant son environnement. Ce n'est qu'une fois ce traitement fait qu'il peut identifier les objets qui sont dans son environnement proche. Il lui faut ensuite bâtir une sorte de structure interne des données analysées, pour se représenter la scène dans son ensemble, puis utiliser cette structure pour faire une planification. Il doit calculer après au mieux une séquence de commandes vers les effecteurs pour exécuter le plan prévu.

Dans une telle architecture, on essaie de construire un modèle de l'environnement le plus complet possible et ensuite de raisonner sur la (les) représentation(s) de celui-ci. On privilégie l'aspect cognitif dans le but de reproduire au mieux l'intelligence humaine. On cite ci-dessous les avantages et inconvénients de tels contrôleurs :

I.2.2.1.a Avantages des contrôleurs hiérarchiques:

- L'atout essentiel de ces architectures réside dans la possibilité d'intégrer des raisonnements de haut niveau (niveau mission, planification) qui s'appuient sur des modèles assez complets (cartes par exemple) de l'environnement dans lequel évolue le robot. Cet aspect est, comme nous le verrons plus loin, absent des architectures purement réactives.

I.2.2.1.b Inconvénients des contrôleurs hiérarchiques:

- Ces systèmes, même implantés sur des calculateurs rapides, sont relativement lents pour un observateur extérieur (notamment pour maintenir le modèle du monde). De plus, un cycle de traitement s'étend de l'acquisition des données des capteurs jusqu'à l'envoi de commandes sur les effecteurs en traversant toutes les couches de l'architecture. On observe ainsi un manque de réactivité (prise en compte et réponse en temps réel) avec l'environnement dans lequel évolue le robot.
- Ces systèmes sont peu adaptatifs dans la mesure où il leur est difficile de réagir à des situations légèrement différentes de celles qu'ils connaissent. Cela exprime une lacune dans la possibilité de généraliser des situations possibles.
- Ces systèmes sont en général peu robustes de par leur modèle centralisé. Une défaillance d'un module peut provoquer le blocage de toute l'architecture.

I.2.2.2 Contrôleurs réactifs

Dans cette architecture, un ensemble de comportements réactifs, fonctionnant en parallèle, contrôle le robot sans utiliser de modèle du monde. Cette architecture supprime évidemment les problèmes dûs aux différences entre la réalité, d'une part, et le modèle de l'environnement du robot, d'autre part, mais limite clairement les tâches que peut effectuer le robot. En effet, sans représentation interne de l'état de l'environnement, il est très difficile de planifier une suite d'actions en fonction d'un but à atteindre. Les robots utilisant cette architecture sont donc en général efficaces pour la tâche précise pour laquelle ils ont été

conçus, dans l'environnement pour lequel ils ont été prévus, mais sont souvent difficiles à adapter à une tâche différente.

Les réussites de ces architectures sont liées au couplage direct entre la perception et l'action qui permet une prise en compte très rapide des phénomènes dynamiques de l'environnement. On a donc une bonne robustesse dans des environnements complexes.

Ces architectures sont en général basées sur plusieurs comportements : évitement d'obstacles, déplacement aléatoire, déplacement vers un but, fuite d'un point... Pour guider le robot, il faut donc choisir à chaque instant lequel de ces comportements activer.

I.2.2.2.a Avantages des contrôleurs réactifs

- Robustesse

Elle est due au fait de la relative indépendance de chacun des comportements les uns par rapport aux autres. Si un venait à avoir une défaillance, le robot serait toujours capable de réaliser une tâche, ce qui est très important lorsque l'on a affaire à un robot autonome.

- Rapidité de réponse

Ceci est dû au fait, que les comportements sont directement placés entre les capteurs et les effecteurs.

- Aspect incrémental en niveau

Il suffit de mettre en place un niveau de compétence puis d'ajouter les niveaux supérieurs sans avoir, à remettre en question les niveaux précédents. Cet aspect est primordial pour la mise au point et la recherche des pannes notamment. Cette approche permet également d'accroître la complexité du comportement global du robot sans pour autant accroître celle du contrôle global puisque celui-ci est complètement distribué.

- Faible coût

Un robot conçu avec une telle architecture est d'un coût moindre à celui d'un robot doté d'une architecture traditionnelle réalisant la même tâche.

- Possibilité de miniaturisation

D'après Brooks, la relative simplicité de l'architecture de contrôle de ses robots autorise une réduction drastique de ces derniers. Ceci pourrait ouvrir la porte à de vastes domaines d'applications dont certains peuvent apparaître encore utopiques

I.2.2.3 Contrôleurs hybrides

La plupart des contrôleurs actuellement utilisés choisissent une solution intermédiaire entre deux approches sous la forme d'une architecture hybride. Cette architecture se compose de deux niveaux. Le premier est chargé des tâches de navigation de haut niveau, telles que la localisation, la cartographie et la planification. Pour cela, il s'appuie sur un second niveau réactif qui est chargé d'exécuter les commandes avec le plus de précision possible et de gérer les éléments non modélisés de l'environnement tels que les obstacles inconnus ou dynamiques. L'action conjointe de ces deux niveaux permet de réagir rapidement face aux variations imprévues de l'environnement, tout en permettant la réalisation d'actions planifiées à plus long terme.

Le bas niveau de ces architectures peut être réalisé sous forme de comportements, tels que ceux utilisés dans les architectures réactives. Ces comportements sont des boucles sensorimotrices qui relient les actions aux perceptions avec une phase de décision très courte, qui assure la réactivité. Dans le même temps, les informations sensorielles sont utilisées par le haut niveau dans une boucle sensorimotrice à une échelle de temps beaucoup plus longue. C'est la mise en parallèle de ces deux échelles de temps qui fait la force de ces architectures.

I.3 Véhicule holonome et non-holonome[3][4] :

En robotique, il existe deux grandes catégories de robots mobiles à roues. Les robots à pattes sont exclus dans notre étude.

I.3.1 Les véhicules non-holonomes

Les véhicules dit non-holonomes sont ceux que l'on rencontre plus dans la vie courante : voiture particulière, bus, camion, ... Ces véhicules ont une structure mécanique relativement simple, des roues motrices, des roues directrices et des roues porteuses. Une roue peut avoir une, deux ou les trois fonctions. Mais tous ces véhicules ont une caractéristique commune : la direction de la vitesse d'avance (ou vitesse linéaire) est imposée par la direction des roues directrices. Prenons un exemple : pour qu'une voiture aille de sa position initiale à

un mètre sur sa droite elle est obligée de faire une manœuvre : une marche avant puis une marche arrière.

I.3.2 Les véhicule holonomes :

Le deuxième type de véhicules, beaucoup plus rare dans notre vie quotidienne, s'appelle les véhicules holonomes. Ils ont une structure mécanique complexe qui leur permet de se déplacer dans toutes les directions sans manœuvre !

Il existe, néanmoins, un exemple : le chariot de magasin. Vous pouvez prendre un chariot et le tirer sur la droite, les roues s'orientent alors dans la bonne direction.

Un chariot n'a pas une structure mécanique compliquée mais ses articulations sont passives: La société Nomadie, a conçu un robot holonome :le **XR4000**. Il dispose de 4 roues motrices et directrices montées comme des roues de chariot. La synchronisation des 8 axes (2 par roue. Rotation et orientation) est assurée par une carte dédiée basée sur le microcontrôleur Motorola 68332 et des circuits **FPGAs** (Field Programmable Gate Arrays, réseau de portes programmables *in situ* ou circuits pré diffusés programmables) et la structure mécanique est composée d'engrenages coniques.

- **Contrainte non holonome**

Un robot mobile A est modélisé par un corps rigide monté sur plusieurs roues, certaines étant directrices et autres fixes. Ce robot se déplace sol - plan, le contact entre les roues et le sol étant supposé ponctuel et sans glissement. Une position de ce robot est habituellement représentée par une configuration $q = (x, y, \theta)$ de $\mathbb{R}^2 \times S^1$, où :

(x, y) sont les coordonnées d'un point de référence de A , et θ est l'angle entre l'axe orienté (Ox) et un axe de référence orienté (dit axe principal) du robot.

Pour que A puisse se déplacer et non seulement tourner autour d'un point fixe, les roues dont les axes sont fixés au châssis du robot doivent avoir des axes de rotation parallèles. Comme ces roues ne dérapent pas, la vitesse des points de A situés sur les axes de ces roues est perpendiculaire à ces roues. On prend alors l'axe principal du robot perpendiculaire à ces axes de rotation. La contrainte cinématique précédente est appelée contraintes d'orientation, la vitesse de A étant contrainte par son orientation.

Prenons le cas de la plate-forme de type voiture de la figure (Fig.I.5) ci-dessous :

La voiture est réduite volontairement à son modèle équivalent à une seule roue directrice à l'avant (roue centrée orientable, placée sur l'axe longitudinal du véhicule et deux roues fixes sur le même axe à l'arrière. Les roues sont numérotées sur la figure (Fig.I.5) (1=roue avant, 2=roue arrière droite et 3=roue arrière gauche).

La condition de roulement et pivotement sans glissement de la roue droite est décrite par deux équations qui correspondent à une projection dans le plan de la roue et dans un plan perpendiculaire à la roue:

$$\begin{cases} \dot{x} \cos \theta + \dot{y} \sin \theta + L \dot{\theta} - r \dot{\theta}_r = 0 \\ -\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \end{cases} \quad (\text{I.1})$$

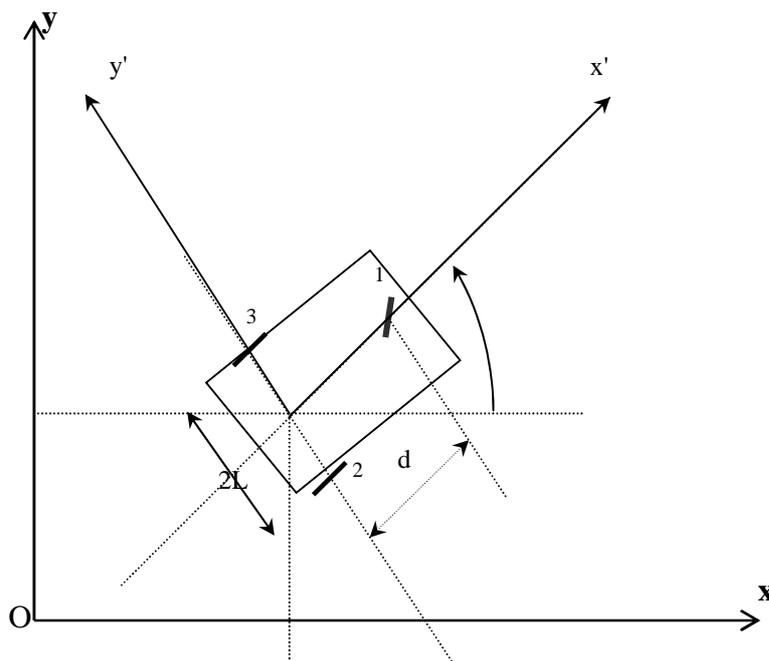


Fig. I.5 : Plate-forme mobile de robot de type voiture

De même, la condition de roulement sans glissement de la roue gauche est décrite par les deux équations:

$$\begin{cases} \dot{x} \cos \theta + \dot{y} \sin \theta - L\dot{\theta} - r\dot{\theta}_l = 0 \\ -\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \end{cases} \quad (\text{I.2})$$

On en déduit que la condition de roulement sans glissement de la plate-forme est décrite par l'équation :

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

Et que

$$\begin{cases} \dot{x} = \frac{r}{2}(\dot{\theta}_l + \dot{\theta}_r) \cos \theta \\ \dot{y} = \frac{r}{2}(\dot{\theta}_l + \dot{\theta}_r) \sin \theta \end{cases} \quad (\text{I.4})$$

r : est le rayon des deux roues arrières

$\dot{\theta}_l$ Et $\dot{\theta}_r$ sont les vitesses angulaires respectivement des roues gauche et droites.

La relation de roulement non holonomie de la plate-forme mobile s'écrit donc dans R comme suit:

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (\text{I.5})$$

I.3.3 Le concept de chemin d'un robot

Un chemin γ pour un robot R est une fonction continue :

$\gamma : [0,1] \rightarrow C : t \rightarrow q = \gamma(t)$ C représente l'espace de configuration.

Ainsi, un chemin γ associe une configuration à chaque moment t , décrivant le mouvement du robot.

Si q_i et q_f correspondent respectivement aux configurations initiales et finales telles que $q_i = \gamma(0)$ et $q_f = \gamma(1)$ on dit que γ relie q_i et q_f .

Si γ reste à l'intérieur de C_{libre} , i.e. $\forall t \in [0,1] : \gamma(t) \in C_{\text{libre}}$, alors γ est appelé un chemin libre (ou sans collision).

Lorsqu'un chemin γ décrit un mouvement sans collision exécutable par un robot, on le qualifie de chemin faisable.

I.3.4 Commandabilité d'un robot mobile [5][6]:

On a jugé utile de commencer à rappeler certaines définitions concernant la commandabilité d'un robot mobile

- **Définition 1:** (commandabilité d'un robot mobile)

Un robot A est dit commandable si, et seulement si, les contraintes cinématiques (qui limitent son déplacement) ne limitent pas l'ensemble des positions (ou des configurations) qu'il peut atteindre (en l'absence d'obstacles).

Pour obtenir des résultats en présence d'obstacles, il faut que le robot vérifie une autre propriété, dite de commandabilité en temps petit.

- **Définition 2:** (commandabilité en temps petit)

Un robot A est dit commandable en temps petit si, et seulement si, pour toute configuration q et pour tout voisinage \mathcal{U} de q , il existe un voisinage de q dont toute configuration est atteignable depuis q par un chemin admissible pour A et inclus dans \mathcal{U} .

Si A est commandable en temps petit, quelle que soit la disposition des obstacles, l'existence d'un chemin sans collision (sans contact) reliant une configuration q_i à une configuration q_f est indépendante du respect des contraintes cinématiques par le chemin: un chemin vérifiant les contraintes et sans collision existe entre ces configurations si, et seulement si, il existe un chemin sans collision.

I.3.5. Modélisation des robots mobiles [3][5][6] :

On considère généralement, deux points de vue pour modéliser un robot mobile. Le premier point de vue part de l'hypothèse de roulement sans glissement et conduit directement à un modèle cinématique. Le deuxième adopte un point de vue plus opérationnel en considérant les commandes effectives qui doivent être appliquées aux actionneurs du robot: le modèle cinématique est alors dérivé du modèle dynamique les deux points de vue sont évidemment complémentaires.

I.3.5.1 Le robot à deux roues motrices :

Ce mode classique de locomotion comprend deux roues parallèles, l'accélération de chacune d'elles étant commandée par un moteur indépendamment de l'autre (Figure I.6).

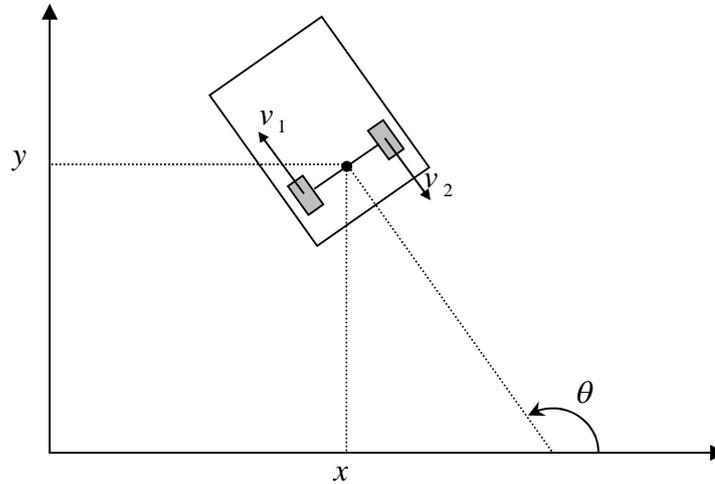


Fig.I.6 : Robot mobile à deux roues motrices

La stabilité de la plate-forme est assurée par des roues folles.

Prenons comme point de référence le point au milieu de l'axe des roues motrices. Ses coordonnées par rapport à un repère fixe de l'environnement sont notées (x, y) . La direction du véhicule est la direction θ des roues motrices. Si D représente la distance entre les deux roues motrices, le modèle cinématique du robot est un système de dimension 5 :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v}_1 \\ \dot{v}_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(v_1 + v_2) \cos \theta \\ \frac{1}{2}(v_1 + v_2) \sin \theta \\ \frac{1}{D}(v_1 - v_2) \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2 \quad (\text{I.6})$$

Où u_1 et u_2 sont les accélérations des roues. Elles sont bornées, en ce sens que l'on a $|u_1| \leq u_{1\max}$ et $|u_2| \leq u_{2\max}$. Les quantités v_1 et v_2 sont les vitesses des roues; v_1 et v_2 sont également bornées. A ce niveau de modélisation, ces bornes apparaissent comme des

"obstacles" à éviter dans l'espace de dimension 5. Le système n'est pas commandable en temps petit. Ceci est dû à la présence d'une dérive et au fait que les accélérations sont bornées.

En posant $v = \frac{1}{2}(v_1 + v_2)$ et $\omega = \frac{1}{D}(v_1 - v_2)$, on obtient le modèle cinématique du robot qui s'exprime comme un système de dimension 3:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega = f_1(x)v + f_2(x)\omega \quad (I.7)$$

Les bornes sur v_1 et v_2 induisent des bornes v_{\max} et ω_{\max} sur les nouvelles commandes v et ω .

I.3.5.2 Robot type voiture

Du point de vue de conducteur, une voiture possède deux commandes : l'accélérateur et le volant (le frein apparaît comme un accélérateur inversé). prenons comme point de référence le point de coordonnées (x, y) au milieu de l'axe des roues de l'essieu arrière (Fig.I.7)

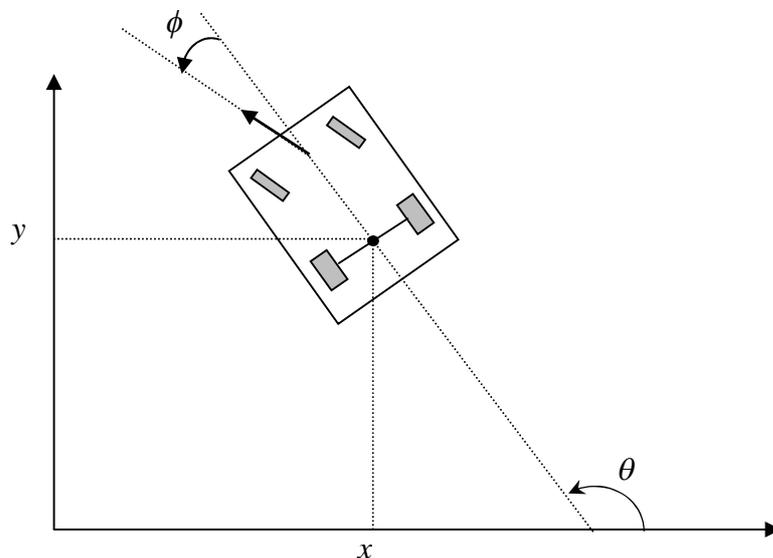


Fig. I.7 : Robot de type voiture

Supposons que la distance entre les trains avant et arrière soit 1. Notons ω la vitesse des roues avant de la voiture et ϕ l'angle formé par les roues avant et l'axe principal de la voiture,

d'orientation θ . Les contraintes mécaniques imposent $|\phi| \leq \phi_{\max}$ et par conséquent, une borne inférieure sur le rayon de braquage. Le modèle dynamique correspondant est:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\omega} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \omega \cos \phi \cos \theta \\ \omega \cos \phi \sin \theta \\ \omega \sin \phi \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2 \quad (\text{I.8})$$

avec $|u_1| \leq u_{1\max}$ et $|u_2| \leq u_{2\max}$.

La voiture apparaît comme un système de dimension 5, non commandable dans un temps petit à partir de n'importe quel point.

Une première simplification consiste à considérer ω comme une commande. On obtient alors un système de dimension 4:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos \phi \cos \theta \\ \cos \phi \sin \theta \\ \sin \phi \\ 0 \end{pmatrix} \omega + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2 = \mathbf{g}_1 \omega + \mathbf{g}_2 u_2 \quad (\text{I.9})$$

Supposons maintenant que l'on ne s'intéresse pas à l'orientation des roues avant. En posant $v = \omega \cos \phi$ et $\omega = \omega \sin \phi$, on obtient un système de dimension 3:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \quad (\text{I.10})$$

I.3.6 Commande des robots mobiles à roues [5][6]:

La commande des robots mobiles a fait l'objet de nombreuses recherches, ces dix dernières années. En particulier, les propriétés de non holonomie ont conduit à une utilisation de techniques de commande fortement non linéaires. Ces approches seront abordées, mais nous en avons volontairement limité aux cas linéaires, afin de présenter prioritairement des techniques plus classiques (c'est-à-dire faisant principalement appel aux outils de l'automatique linéaire) dont les bases, à la fois pratiques et théoriques sont mieux établies.

Les méthodes de commande seront développées principalement pour les robots de types unicycle et de type voiture. La plupart des résultats se généralisent, en fait, à d'autres robots mobiles, et en particulier, au système avec remorques.

- Le robot de type unicycle est composé, schématiquement, de deux roues motrices indépendantes à l'arrière, du corps principal du robot, et d'une roue folle à l'avant (roue non commandée) destinée uniquement à assurer la stabilité du système.
- Le robot de type voiture est composé d'un train moteur à l'arrière, du corps principal, et de roues de direction à l'avant.

Notons également, comme l'illustre la figure (Figure I.9) que le robot de type voiture peut être assimilé (au moins du point de vue cinématique) à un robot unicycle suivi d'une remorque.

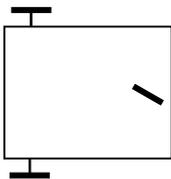


Fig. I.8.a : Robot de type uni cycle

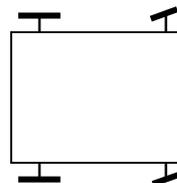


Fig. I.8.b : Robot de type voiture

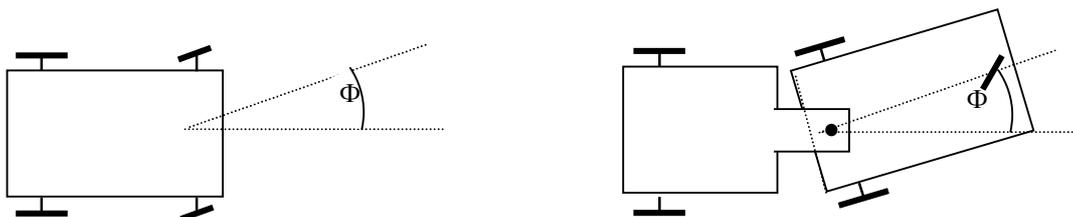


Fig. I.9 : Analogie voiture/chariot avec remorque

L'étude de la commande des robots mobiles nécessite l'étude des trois problèmes de commande évoqués ci-dessous.

I.3.6.1 Suivi de chemin :

Etant donné une courbe C du plan, une vitesse d'avancement v_0 (non nulle) pour le robot mobile et un point P fixe sur le robot, on souhaite que le point P suive la courbe C lorsque le robot roule à la vitesse v_0 . La variable que l'on doit régler à zéro est donc la distance du point P à la courbe (c'est-à-dire la plus petite distance de P à M lorsque M parcourt C). Ce type de problème correspond, typiquement, à la conduite sur route lorsque le conducteur cherche à maintenir une distance constante par rapport à des marquages au sol, par exemple.

I.3.6.2 Stabilisation de trajectoire:

Pour le problème de stabilisation de trajectoires se distingue du précédent par le fait que la vitesse d'avancement du véhicule n'est plus fixée à l'avance, du fait que l'on cherche également à régler la distance parcourue le long de la courbe C .

Cet objectif nécessite de doter la courbe C d'une loi horaire, c'est-à-dire de la paramétrer par la variable temporelle t . Cela revient à définir une trajectoire $t \rightarrow (x_r(t), y_r(t))$ par rapport à un repère de référence R_0 . On souhaite alors réguler à zéro le vecteur d'erreur $[x(t) - x_r(t), y(t) - y_r(t)]$, où $(x(t), y(t))$ sont les coordonnées du point P dans R_0 , à l'instant t . Elle peut être également interprétée comme celui consistant à asservir le véhicule à un véhicule de référence, dont la trajectoire est donnée par $t \rightarrow (x_r(t), y_r(t))$.

I.3.6.3 Stabilisation de configurations fixes :

Pour le problème, de la stabilisation de configuration fixes, on se donne un repère de référence R_0 , ainsi qu'un repère lié au véhicule R_1 . L'objectif est alors de régler la valeur de position $(x(t), y(t))$ (exprimé dans le repère R_0) d'un point P lié au véhicule, ainsi que l'orientation $\theta(t)$ de R_1 par rapport à R_0 .

Ce problème est celui pour lequel les propriétés de non holonomie interviennent de la manière la plus contraignante, et pour lequel les méthodes classiques de l'automatique linéaire se montrent insuffisantes. Les principales applications de ce problème sont liées aux manœuvres de parking (créneau, etc.) et, plus généralement, aux applications qui nécessitent une immobilisation du véhicule s'accompagnant d'un positionnement précis.

I.3.7 Modèles de commande:

Les équations dynamiques d'un système non holonome s'écrivent sous la forme:

$$J(q)\dot{u} + N(q,u)u + G(q) = B^T(q)D(q)\Gamma \quad (\text{I.11})$$

à laquelle il faut ajouter les équations cinématiques qui prennent en considération les contraintes non holonomes:

$$\dot{q} = B(q)u \quad (\text{I.12})$$

où $q \in R^n$ est le vecteur des variables de configurations $u \in R^m$ correspond aux vitesses instantanées non contraintes (sa dimension m est le nombre de degré de liberté du système, soit $m=2$ pour le robot mobile considéré ici) et $\Gamma \in R^m$ correspond aux forces des couples moteur appliqués, produits par les actionneurs dont le véhicule équipé. Pour les robot mobiles rencontrés usuellement (unicycle, voiture, chariot avec remorque, etc.) on a, de plus, les propriétés suivantes:

- $B(q) \in R^{n \times m}$ est de rang plein $m < n$;
- $J(q) \in R^{m \times m}$ est inversible;
- $D(q) \in R^{n \times m}$ est de rang plein.

Cela revient à supposer que le véhicule est équipé d'autant d'actionneurs indépendants (m) qu'il y a deux degrés de liberté, chacun produisant une force ou un couple. Si tel n'était pas le cas, le système serait sous actionné et les problèmes de commande seraient beaucoup plus difficiles (voire impossibles à résoudre dans certains cas).

Selon la terminologie consacrée en automatique, le modèle dynamique (I.11)-(I.13) est un "système de commande" de type $\dot{x} = f(x,v)$ avec $x = (q,u)$ comme vecteur d'état et

$v = \Gamma$ comme commande. le modèle cinématique (I.14) est également un système de commande avec q comme vecteur d'état et u comme vecteur de commande.

I.3.7.1 Robot unicycle:

Pour le robot unicycle, le modèle cinématique utilisé est:

$$\begin{aligned}\dot{x} &= u_1 \cos \theta \\ \dot{y} &= u_1 \sin \theta \\ \dot{\theta} &= u_2\end{aligned}\tag{I.13}$$

où (x, y) représente les coordonnées du point P_0 situé au milieu de l'axe des roues arrières et θ l'orientation du robot (Fig.II.5.a). Dans cette équation, u_1 représente l'intensité de la vitesse d'avancement du véhicule et u_2 la vitesse instantanée de rotation du corps du véhicule. Les variables u_1 et u_2 sont elles-mêmes reliées aux vitesses de rotation des roues arrières par les relations:

$$\begin{aligned}u_1 &= \frac{r}{2}(\dot{q}_1 + \dot{q}_2) \\ u_2 &= \frac{r}{2R}(\dot{q}_1 - \dot{q}_2)\end{aligned}\tag{I.14}$$

Avec r : Rayon des roues et R la distance entre ces roues.

Notons que ce modèle ne contient, en fait, qu'une partie des équations (I.13). Il est cependant habituel de ne conserver, dans le modèle, que les variables dont la régulation présente un intérêt pratique.

I.3.7.2 Robot de type voiture [3]:

Pour le robot modèle utilisé le système d'équations sera:

$$\begin{aligned}\dot{x} &= u_1 \cos \theta \\ \dot{y} &= u_1 \sin \theta \\ \dot{\theta} &= \frac{u_1}{L} \tan \phi \\ \dot{\phi} &= u_2\end{aligned}\tag{I.15}$$

où ϕ représente l'angle de braquage du véhicule et L la distance entre les axes des roues avant et arrière.

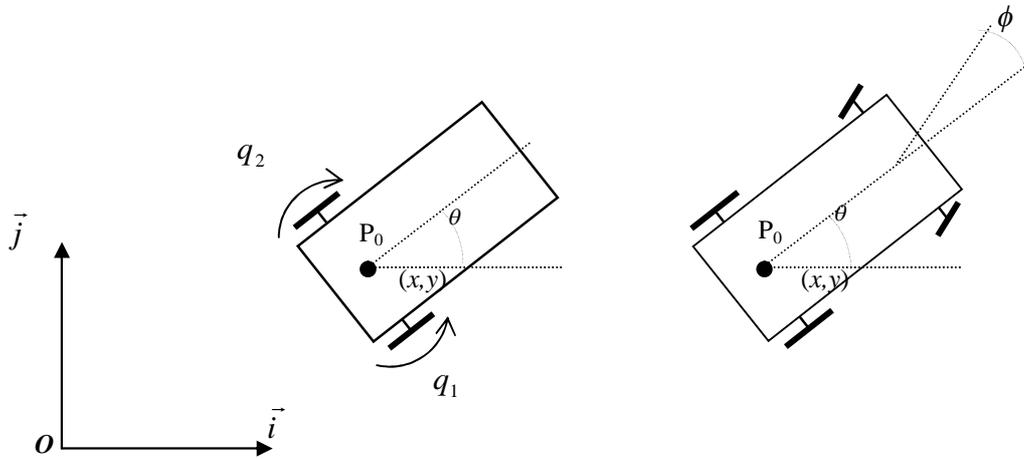


Fig.I.10: Variables de configuration

I.3.8 Poursuite d'un véhicule de référence:

Nous allons considérer le problème de stabilisation de trajectoire pour un robot unicycle, lorsque le point de référence P est situé au milieu de l'axe des roues arrière.

Contrairement à ce qui se passe lorsque P n'est pas situé sur cet axe ($d \neq 0$), le vecteur de vitesse de P est alors contraint par la non holonomie. Cela implique, en particulier, qu'une trajectoire de référence $t \rightarrow (x_r(t), y_r(t))$ n'est pas nécessairement réalisable par le robot. Aussi, pour que le problème de stabilisation ait un sens, il convient d'abord de caractériser les trajectoires réalisables. Une manière de le faire est de considérer que $t \rightarrow (x_r(t), y_r(t))$ correspond elle-même à la trajectoire d'un robot unicycle de référence, c'est-à-dire qu'il existe une "commande de référence" $t \rightarrow (u_{1,r}(t), u_{2,r}(t))$ (supposée continue) tel que:

$$\begin{aligned} \dot{x}_r &= u_{1,r} \cos \theta_r \\ \dot{y}_r &= u_{2,r} \cos \theta_r \\ \dot{\theta}_r &= u_{2,r} \end{aligned} \quad (\text{I.16})$$

Pour résoudre le problème du contrôle de la poursuite le changement global de Coordonnées suivant était proposé par *Kanayama et al.*

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (\text{I.17})$$

Ce changement global des coordonnées de $[x_r - x, y_r - y]^T$ vers $[x_e, y_e]$ rend les variables d'erreur indépendantes du choix de repère des coordonnées inertielles, les erreurs sont considérées dans un repère lié au robot mobile. Avec ces nouvelles coordonnées l'erreur dynamique de poursuite devient:

$$\begin{aligned} \dot{x}_e &= \omega y_e - v + v_r(t) \cos \theta_e \\ \dot{y}_e &= -\omega x_e + v_r(t) \sin \theta_e \\ \dot{\theta}_e &= \omega_r(t) - \omega \end{aligned} \quad (\text{I.18})$$

Le problème du contrôle consiste à trouver des lois de commande pour v et ω telle que l'erreur de poursuite $(x_e, y_e, \theta_e)^T$ converge vers zéro.

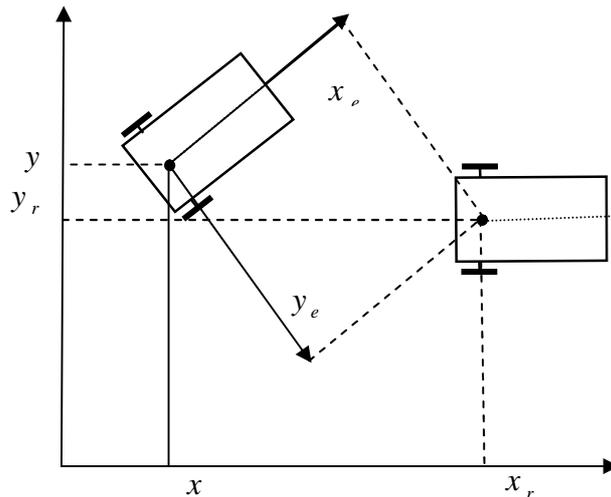


Fig. I.11 : Nouvelles coordonnées de l'erreur

I.3.9 Modèle de coordonnée d'un chemin [3][5][6]

Les coordonnées d'un chemin sont représentées dans la figure (I.12).

la distance perpendiculaire entre l'essieu arrière et le chemin est donnée par d l'angle entre véhicule et la tangente de chemin est : $\theta_p = \theta - \theta_t$; la position initiale arbitraire le long de chemin est donnée par s , longueur de l'arc.

Le modèle cinématique d'un véhicule en terme de coordonnées de chemin est donnée par :

$$\begin{bmatrix} \dot{s} \\ \dot{d} \\ \dot{\theta}_p \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{\cos \theta_p}{1 - dc(s)} \\ \sin \theta_p \\ \frac{\tan \phi}{l} - \frac{c(s) \cos \theta_p}{1 - dc(s)} \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \dots \dots \dots (I.19)$$

Ou $c(s)$ est la courbure de chemin définie par

$$c(s) = \frac{d\theta_t}{ds}$$

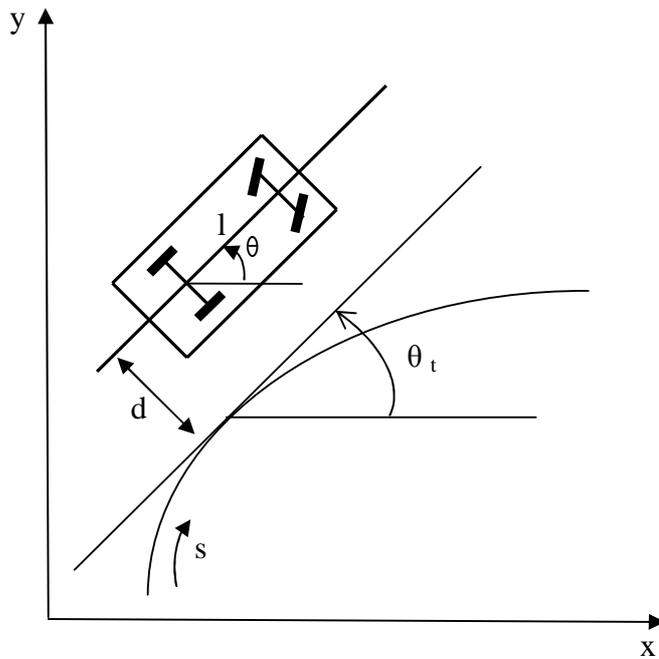


Fig. I .12: les coordonnées de chemin d'un véhicule

- **Cas forme enchaînée:**

On met le modèle (I 15) à la forme enchaînée utilisant le changement de coordonnées.

La structure de la forme enchaîner (appelée (2*n) chaîne – singulier):

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \dots \dots \dots (I . 20) \\ &\cdot \\ &\cdot \\ &\cdot \\ \dot{x}_n &= x_{n-1} u_1 \end{aligned}$$

Le système à deux entrées u_1 et u_2 , ce modèle peut être considéré à entrée unique si u_1 signalé prioritaire pour le modèle de véhicule avec 4 états, la forme enchaîner (2*4) devient:

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \dots \dots \dots (I . 21) \\ \dot{x}_4 &= x_3 u_1 \end{aligned}$$

Les états sont données par:

$$\begin{aligned} x_1 &= s \\ x_2 &= -c'(s) \tan \theta_p - c(s)(1 - dc(s)) \frac{1 + \sin^2 \theta_p}{\cos^2 \theta_p} + \frac{(1 - dc(s))^2 \tan \phi}{l \cos^3 \theta_p} \dots \dots \dots (I. 22) \\ x_3 &= (1 - dc(s)) \tan \theta_p \\ x_4 &= d \end{aligned}$$

Les variables sont définis dans la figure (I.12), $c(s)$ est la courbure de chemin, $c'(s)$ est la dérivé de c par rapport à s .

Les entrées sont définies comme suit:

$$\begin{aligned} v_1 &= \frac{1 - dc(s)}{\cos \theta_p} u_1 \dots \dots \dots (I. 23) \\ v_2 &= \alpha_1 (u_2 - \alpha_1 u_1) \end{aligned}$$

Où v_1 la vitesse linéaire de roues arrière, v_2 la vitesse angulaire de roues de braquage

$$\alpha_1 = \frac{\partial x_2}{\partial s} + \frac{\partial x_2}{\partial d}(1-dc(s))\tan\theta_p + \frac{\partial x_2}{\partial \theta_p} \left[\frac{\tan\phi \cdot (1-dc(s))}{l \cos\theta_p} - c(s) \right] \quad (\text{I.24})$$

$$\alpha_2 = \frac{l \cos^2\theta_p \cos^2\phi}{(1-dc(s))^2}$$

Chapitre II

Logique floue et contrôle

II.1. Introduction

Les connaissances de l'univers dans lequel nous évoluons sont généralement imparfaites. Nous pouvons constater que l'homme intègre naturellement ces imperfections dans la vie de tous les jours, en particulier au niveau du raisonnement et de la décision. L'idée de ZADEH, en 1965, à travers le nouveau concept ensembliste d'appartenance graduelle d'un élément à un ensemble, a été de définir une logique multivaluée permettant de modéliser ces imperfections en prenant en compte les états intermédiaires entre le tout et le rien. L'utilité de cette approche peut être illustrée de la manière suivante:

Une température de 10°C, pour un humain, est généralement considérée comme froide; une température de 40°C est, elle, qualifiée de chaude. Si chacune de ces valeurs appartient à une catégorie (ensemble) bien définie, qu'en est-il pour des valeurs intermédiaires. Une réponse intuitive consiste à affirmer qu'elles appartiennent à une ou deux des catégories précédentes avec des niveaux normalisés. On évite ainsi des transitions rigides entre différentes catégories, comme cela est le cas en logique binaire (figure II.1). Il semble en effet surprenant de considérer qu'une température de 40°C est chaude; Alors qu'une température de 39,9°C ne l'est pas. Cet exemple permet d'illustrer le fait qu'une logique binaire classique est, dans certains cas, trop restrictive. Il est nécessaire de faire appel à une logique multivaluée que sera vue comme une extension de la précédente.

En ce qui concerne le contrôle d'un processus quelconque, la logique floue permet une approche novatrice par rapport à l'automatique classique. En automatique, en général, on s'attache à modéliser le processus par la résolution d'un certain nombre d'équations différentielles. Cette modélisation est rendue difficile à mesure que la complexité des processus à contrôler augmente. D'une manière radicalement opposée, un contrôleur va d'écrire non pas le processus mais la façon de le contrôler, tout comme le ferait un expert humain à travers des règles intégrant naturellement imprécisions et incertitudes. Dans cette

approche, nous parlerons donc de systèmes experts flous basés sur des règles de production de la forme : "Si - Alors ".

Ces systèmes sont des extensions des systèmes experts classiques dans la mesure où ils intègrent des connaissances imparfaites

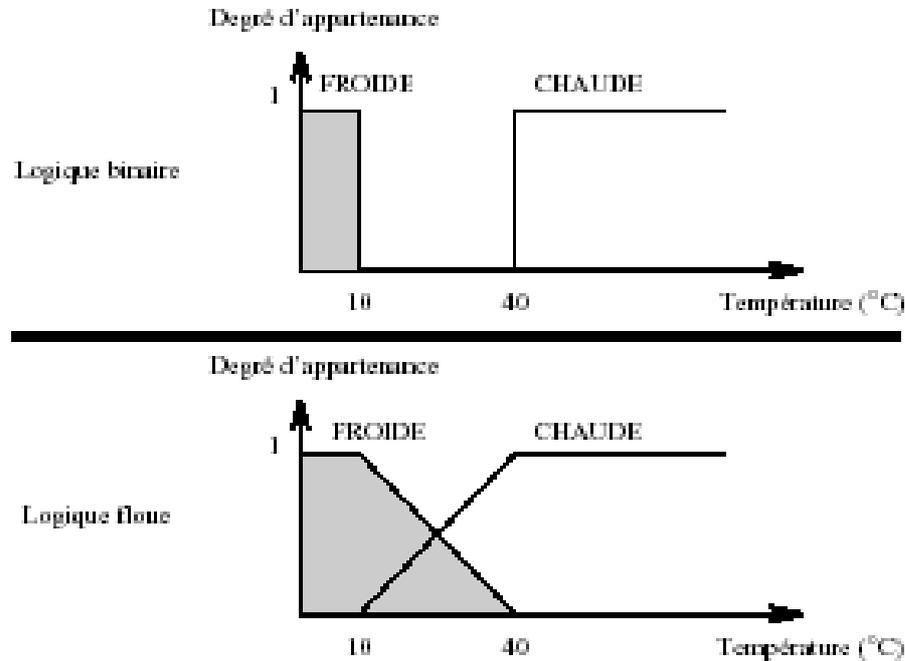


Fig. II. 1: Exemple de définition d'ensemble sur un univers de discours en logique binaire et en logique floue

Un certain nombre d'applications utilisant la logique floue ont vu le jour ces dernières années. Ces applications s'appuient sur les travaux de Mamdani, qui fut certainement le premier à voir la potentialité de la théorie des sous-ensembles flous dans ce domaine. Nous pouvons citer des biens de consommation courante dont le terme flou a même constitué un élément de marketing certain, l'automatisation du métro de Sendai en 1988, machine à laver de Panasonic , caméra vidéo de Sanyo , aspirateur d'Hitachi , télévision de Sony et le climatiseur de Mitsubishi sont des exemples pratiques. Ces exemples illustrent la percée du concept de logique floue dans le domaine industriel.

II.2. Concepts de Base [1][18][19]

Le concept de la théorie des sous-ensembles flous (et par extension, la logique floue), s'appuie sur la notion de degré d'appartenance d'un élément à un sous-ensemble flou. Tandis que les ensembles traditionnels sont caractérisés par une fonction d'appartenance, notée χ , (également appelée fonction caractéristique) définie sur $\{0,1\}$, les sous-ensembles flous sont, eux, caractérisés par une fonction d'appartenance, notée μ , définie sur $[0, 1]$. En d'autres termes, dans le langage ensembliste classique, un élément appartient ou n'appartient pas à un ensemble tandis qu'un élément appartient à un sous-ensemble flou avec un certain degré (éventuellement nul). En résumé, pour un sous-ensemble A défini sur un univers de discours U , on peut écrire :

A sous-ensemble classique : fonction caractéristique $\chi_A : U \rightarrow \{0,1\}$

A sous-ensemble flou : fonction d'appartenance $\mu_A : U \rightarrow [0,1]$

Par extension, ce nouveau concept définit une logique multivaluée qui apparaît comme une généralisation de la logique binaire. Nous allons maintenant définir un certain nombre de termes propres au domaine de la logique floue auxquels nous pourrions nous référer, si nécessaire, dans la suite de ce chapitre.

II.2.1 Sous-ensemble flou

Un sous-ensemble flou A sur un univers de discours U , est représenté selon figure II.2 à travers sa fonction caractéristique μ_A . Il peut également être décrit par un certain nombre de caractéristiques comme :

- Son support :

$$\text{sup port}(A) = \{x \in U / \mu_A(x) \neq 0\}$$

- Sa hauteur:

$$hauteur(A) = \sup_{x \in U} \{\mu_A(x)\}$$

- Son noyau :

$$noyau(A) = \{x \in U / \mu_A(x) = 0\}$$

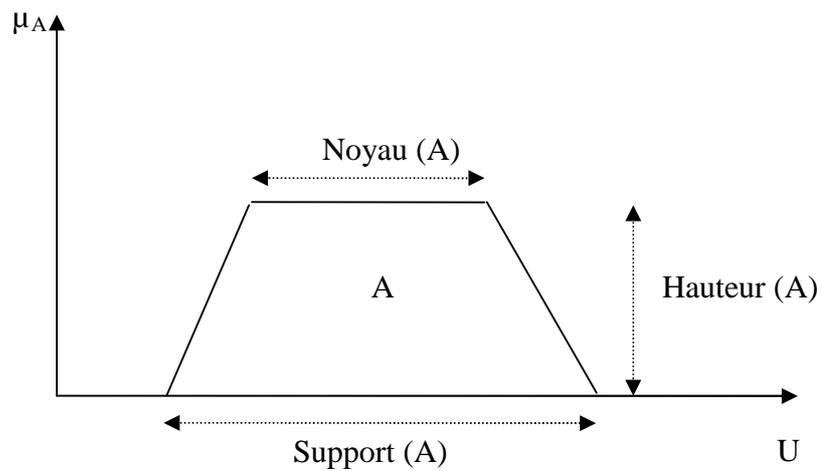


Fig. II. 2 : Représentation d'un sous-ensemble flou et principales caractéristiques

Un sous-ensemble flou permet de représenter différentes notions, en particulier la spécificité et la précision. Ces différentes notions apparaissent dans la figure II.3.

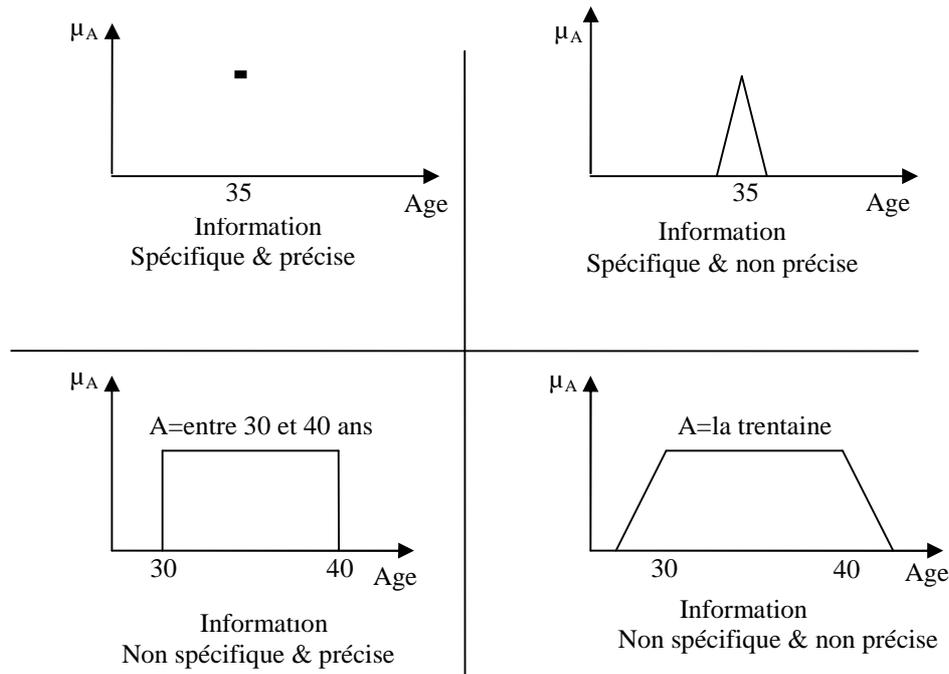


Fig. II .3 : Notions de spécificité et de précision représentée à l'aide de sous ensembles flous

II.2.2 Variable linguistique

Une variable linguistique est définie par un triple $(V ; U ; T_V)$ ou V représente une variable classique (age, température....) définie sur l'univers de discours U .

T_V est l'ensemble des instanciations possibles de la variable V : il s'agit de sous ensembles flous repérés par leur label A_i : on écrit ainsi $T_V = \{A_1; A_2; \dots \dots A_n\}$.

Graphiquement, une variable linguistique peut être représentée dans la figure II.4.

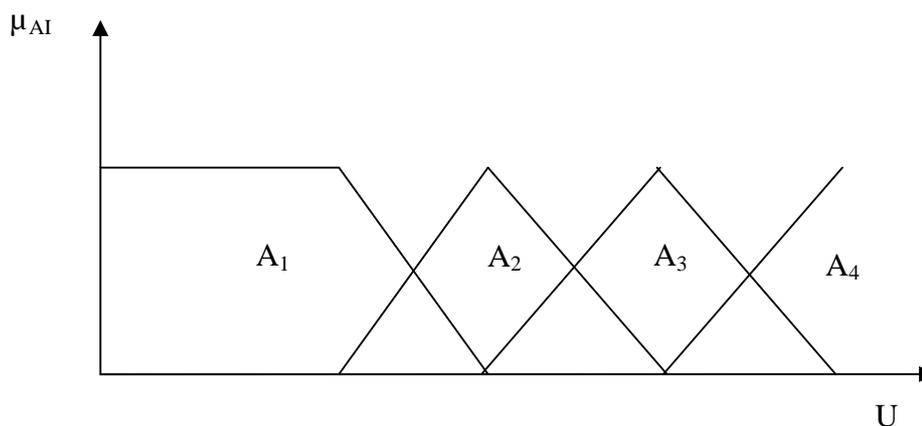


Fig. II .4 : Représentation d'une variable linguistique définie
 $\text{Come} (V; U; T_V) = \{A_1, A_2, A_3, A_4\}$

- Caractérisation floue

Une caractérisation floue d'une variable linguistique $(V; U; T_V)$ est un label A_i de sous-ensemble ou appartenant à T_V . Par la suite, cette dénomination qualifiera indifféremment un sous-ensemble flou ou son label.

- Propositions et règle floue

Une proposition floue élémentaire est définie à partir d'une variable linguistique $(V; U; T_V)$ par la qualification "V est A", avec A appartenant à T_V . Par exemple, " la taille est MOYENNE" est une proposition élémentaire définie à partir de la variable linguistique (taille est {tailles}, { . . . , MOYENNE, . . . }).

La valeur de vérité d'une proposition élémentaire "V est A" est égale à $\mu_A(v)$ ou v correspond à la valeur numérique exacte de V.

Une proposition floue générale est définie à partir de propositions élémentaires et d'opérateurs logiques binaires (et, ou, implique) ou unaire (non). Il existe plusieurs méthodes pour calculer la valeur de vérité de telles propositions. Nous ne donnons ici que les plus communément utilisées.

- **Conjonction :** $(V_1 \text{ est } A_1) \text{ et } (V_2 \text{ est } A_2)$

$$\min(\mu_{A_1}(v_1), \mu_{A_2}(v_2)) \quad (\text{Logique de Zadeh})$$

$$\max(\mu_{A_1}(v_1) + \mu_{A_2}(v_2) - 1, 0) \quad (\text{Logique de Lukasiewicz})$$

$$\mu_{A_1}(v_1) \cdot \mu_{A_2}(v_2) \quad (\text{Logique probabiliste})$$

- Disjonction : $(V_1 \text{ est } A_1) \text{ ou } (V_2 \text{ est } A_2)$

$$\max(\mu_{A_1}(v_1), \mu_{A_2}(v_2)) \quad (\text{Logique de Zadeh})$$

$$\min(\mu_{A_1}(v_1) + \mu_{A_2}(v_2), 1) \quad (\text{Logique de Lukasiewicz})$$

$$\mu_{A_1}(v_1) + \mu_{A_2}(v_2) - \mu_{A_1}(v_1) \cdot \mu_{A_2}(v_2) \quad (\text{Logique probabiliste})$$

- implication : $(V_1 \text{ est } A_1) \text{ implique } (V_2 \text{ est } A_2)$

$$\min(1 - \mu_{A_1}(v_1) + \mu_{A_2}(v_2), 1) \quad (\text{Lukasiewicz})$$

$$\min(\mu_{A_1}(v_1), \mu_{A_2}(v_2)) \quad (\text{Mamdani})$$

$$\mu_{A_1}(v_1) \cdot \mu_{A_2}(v_2) \quad (\text{Larsen})$$

- Complémentation : $\text{non}(V \text{ est } A)$

$$1 - \mu_A(v)$$

Dans ce qui précède, v , v_1 et v_2 correspondent à des instanciations numériques réelles des variables V , V_1 et V_2 .

Une règle floue est une proposition floue générale utilisant une implication entre deux propositions floues quelconques. Par exemple :

(V1 est A1) et (V2 est A2) implique (V3 est A3)

Ou sous une forme plus linguistique :

Si (V1 est A1) et (V2 est A2) alors (V3 est A3)

est une règle floue. La partie (V1 est A1) et (V2 est A2) est appelée prémisse de la règle et (V3 est A3) conclusion.

- **Activation d'une règle floue**

Une règle ne peut être activée (i.e. intervenir dans le processus d'inférence) que lorsque la valeur de vérité de la proposition floue constituant sa prémisse est non nulle.

- **Fonction d'appartenance**

Le choix de la forme des fonctions d'appartenance est subjectif, et soumis aux préférences du concepteur (Figure. II.5, II.6 et II.7). Les fonctions d'appartenance les plus répandues en pratiques sont:

La fonction triangulaire (Figure .II.5)

$$\mu_A(x) = \begin{cases} 1 - |x - c| / \rho & \text{si } x \leq \rho \\ 0 & \text{si } x > \rho \end{cases}$$

La fonction trapézoïdale (Figure .II.6)

$$\mu_A(x) = \max \left\{ \min \left\{ \frac{x - \rho_1}{c_1 - \rho_1}, 1, \frac{\rho_2 - x}{\rho_2 - c_2} \right\}, 0 \right\}$$

La fonction gaussienne (Figure .II.7)

$$\mu_A(x) = \exp\left(-\frac{(x - c)^2}{\sigma^2}\right)$$

Où σ est un paramètre caractérise la forme de μ_A .

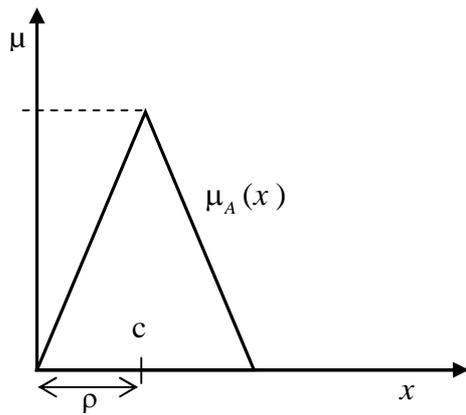


Fig. II.5 : Fonction triangulaire

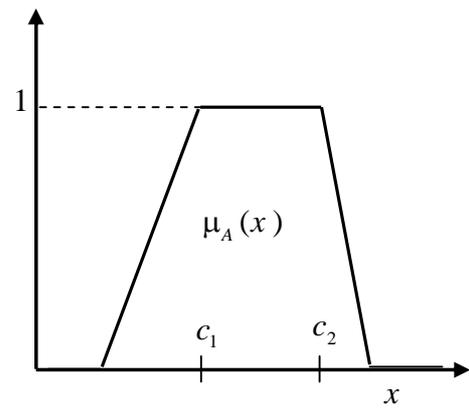


Fig. II.6 : Fonction trapézoïdale

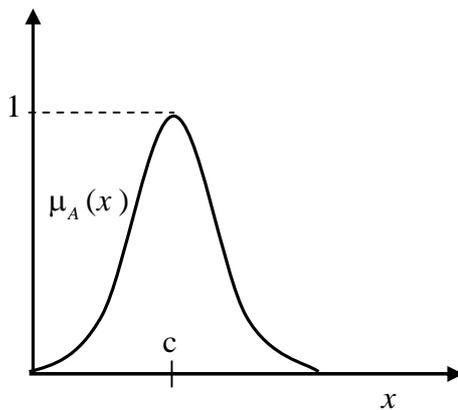


Fig. II.7 : Fonction gaussienne

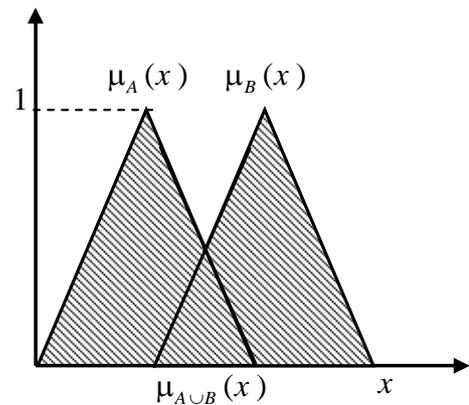


Fig. II.8 : Union floue

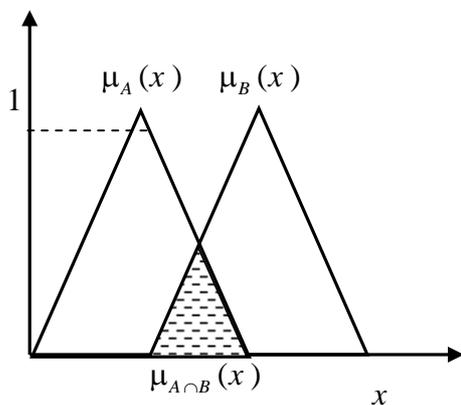


Fig. II.9 : Intersection floue.

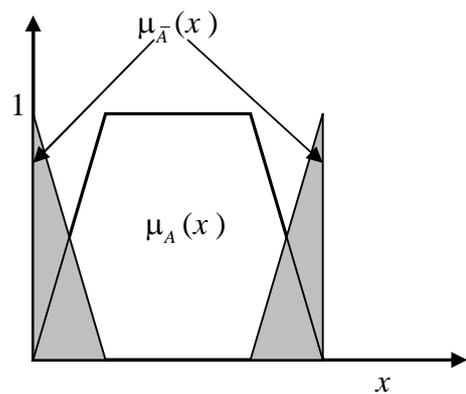


Fig. II.10 : Complément floue.

II.3. Architecture classique d'un contrôleur flou (FLC)

L'architecture classique d'un contrôleur flou (FLC), proposée par Mamdani, est illustrée dans la figure II.11 ci-dessus [1][19][20][22]

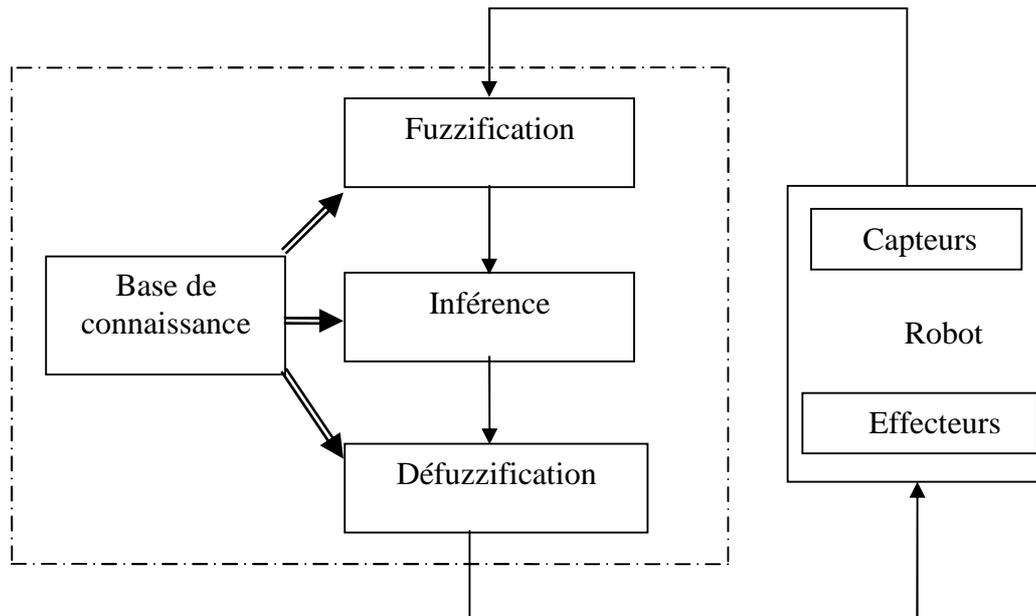


Fig. II. 11 : Architecture d'un contrôleur flou

Comme nous pouvons le voir dans la figure II.11, un contrôleur flou est composé de quatre parties :

- **La fuzzification :**

Les valeurs délivrées par les capteurs du robot, décrivant l'état du système contrôlé (en l'occurrence le robot), sont traduites en labels de sous-ensembles flous caractérisant les variables linguistiques associées.

- **La base de connaissances :**

Elle est composée de :

a/ Variables linguistiques permettant de caractériser les états associés au processus contrôlé;

b/ Règles linguistiques codant la connaissance que l'on a sur le contrôle de ce processus.

- **Inférence sur les règles** : cette étape représente le raisonnement du contrôleur.
- **Défuzzification** :

Il s'agit de la partie duale de la fuzzification, consistant à combiner les sorties des règles calculées à l'étape précédente, dans le but de déterminer les commandes effectives du robot.

Nous allons maintenant reprendre plus en détail ces différentes parties, afin de montrer le rôle de chacune dans l'architecture du contrôleur flou.

II.3.1. Fuzzification

Dans la littérature, l'étape de fuzzification est qualifiée de transformation permettant de passer :

- d'un niveau discret à un niveau continu ;
- d'un niveau numérique/quantitatif à un niveau symbolique /qualitatif.

Chaque instanciation d'une variable d'entrée réelle du contrôleur va être traduite en un label caractérisant la variable linguistique associée. Par exemple, un âge de 30 ans sera traduit en JEUNE, label de la variable linguistique (âge, {âges}, {JEUNE, . . . , AGÈ}). Ainsi une information ne sera plus caractérisée par une valeur numérique, mais par un terme linguistique comme JEUNE, si l'on considère un âge par exemple. Cependant, le degré d'appartenance $\mu_A(v)$ d'une valeur réelle v à un sous-ensemble flou A doit être mémorisé car celui-ci interviendra dans la détermination des valeurs de vérité des propositions utilisant A .

La figure II.12 illustre le mécanisme de fuzzification pour deux éléments appartenant au domaine des âges.

Ainsi deux personnes ayant respectivement 30 et 40 ans appartiennent toutes deux à la catégorie "JEUNE" avec cependant des degrés d'appartenance différents ($\mu_{JEUNE}(30) = 0.66$ et $\mu_{JEUNE}(40) = 0.33$).

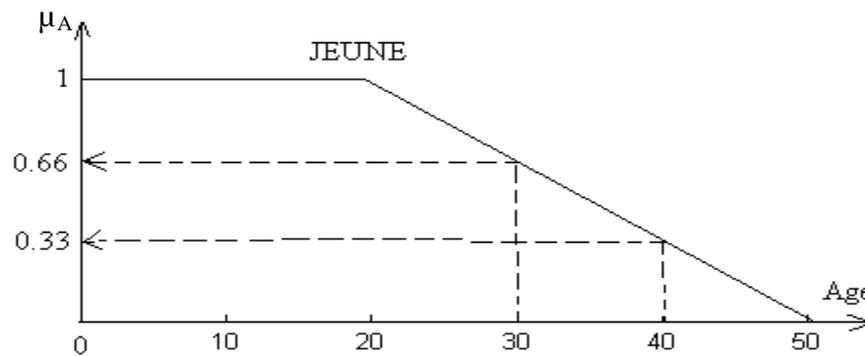


Fig. II. 12 : Mécanisme de fuzzification

II.3.2. Base de connaissances

- Variables linguistiques

Une des premières tâches qui incombent aux concepteurs de systèmes flous est, après avoir déterminé les variables liées au système à contrôler, de définir les variables linguistiques associées. Ainsi, pour chaque variable, il s'agit de discrétiser le domaine de variation en sous-ensembles flous

- Règles linguistiques

Les règles linguistiques sont des règles floues dont la forme générale est la suivante :

"**si** prémisse **alors** conclusion" ,

Où les prémisses portent sur les entrées du contrôleur flou et les conclusions sur les sorties. Comme nous l'avons vu précédemment, ces règles sont en fait des propositions floues incluant un opérateur d'implication.

Les conclusions de règles peuvent être de deux sortes :

- les règles de type Mamdani ou les conclusions, comme les prémisses, sont des propositions floues :

Si x est A1 **et** y est B1 **alors** z est C1

- les règles de type Sugeno ou dans les conclusions, une sortie est fonction des entrées :

$$\text{Si } x \text{ est } A1 \text{ et } y \text{ est } B1 \text{ alors } z = f(x; y)$$

II.3.3. Inférence

Le mécanisme d'inférence consiste à déterminer les règles floues activées (Les règles dont le degré d'activation $\mu_{\text{prémisse}}$ issu des prémisses est non nul). Au niveau de chacune de ces règles, ce degré va permettre de déterminer une valeur floue pour la (les) variable(s) de sortie apparaissant dans la partie "conclusion".

Ce degré d'activation est calculé en utilisant les formules décrites précédemment sur la détermination des valeurs de vérité des propositions floues.

Les deux méthodes d'inférence les plus couramment utilisées sont le MINIMUM et le PRODUIT (figure II.13). L'inférence MINIMUM consiste à tronquer à la valeur $\mu_{\text{prémisse}}$ la caractérisation floue associée à une variable de sortie, alors que l'inférence PRODUIT revient à affecter à celle-ci un facteur d'échelle correspondant à $\mu_{\text{prémisse}}$.

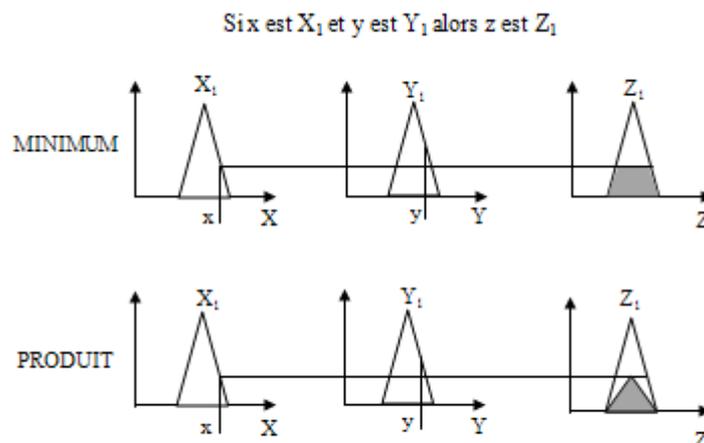


Fig. II. 13 : Inférence « MINIMUM » et « PRODUIT »

II.3.4. Défuzzification

II.3.4.1 Composition

Une fois la phase d'inférence terminée, pour chaque variable linguistique de sortie du contrôleur, il s'agit de regrouper les caractérisations floues issues de l'inférence pour en obtenir une seule par variable. Comme méthodes de composition, on peut citer en particulier les compositions MAXIMUM (en général couplée avec l'inférence MINIMUM) et SOMME (en général couplée avec l'inférence PRODUIT). La première consiste à prendre le maximum de chacune des caractérisations floues impliquées dans la phase d'inférence pour une variable linguistique déterminée. La seconde consiste à faire la somme des caractérisations floues issues de l'inférence (figure II.8). On notera que la seconde méthode ne garantit pas une normalisation de la caractérisation floue résultant de la composition.

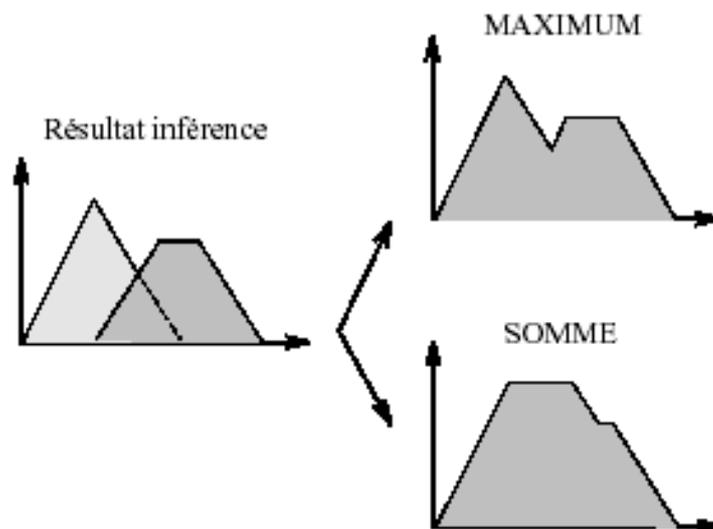


Fig. II.14 : Composition de valeurs floues issues de l'inférence

II.3.4.2 Passage symbolique → numérique

Une fois la phase de composition réalisée, il s'agit de déterminer la valeur numérique de chaque variable de sortie du système ou, à partir de la caractérisation floue résultant de cette phase. Il s'agit là de la phase de défuzzification proprement dite, permettant de générer les commandes qui peuvent être appliquées au système réel contrôlé. Il existe plusieurs méthodes de défuzzification (au moins une trentaine); les plus communément usitées sont :

- Méthode du centre de gravité

Dans le cas où la fonction d'appartenance associée à la caractérisation floue résultant de la phase de composition est de la forme $f(v)$, la valeur numérique délivrée est donnée par la formule :

$$valeur = \frac{\int_U vf(v)dv}{\int_U f(v)dv}$$

avec valeur qui est définie sur l'univers de discours U de v où $\int_U vf(v)dv$ représente le moment de $f(v)$. Cette formule est issue de celle appliquée dans le cas discret (moyenne pondérée), à savoir :

$$valeur = \frac{\sum_i v_i \mu_A(v_i)}{\sum_i \mu_A(v_i)}$$

où les v_i représentent les niveaux de discrétisation de la variable de sortie v et μ_A la fonction caractéristique de la caractérisation floue de sortie A associée à v .

Bien que les fonctions d'appartenance des caractérisations floues puissent être quelconques, force est de constater que, dans la littérature, l'on recourt souvent à des fonctions trapézoïdales, et surtout triangulaires. Cela s'explique certainement par la facilité des opérations pouvant être réalisées sur ces fonctions, notamment dans les phases d'inférence et de défuzzification. En décomposant la caractérisation floue résultant de la phase de composition en polygones convexes élémentaires (triangles par exemple), on peut déterminer

la valeur numérique en prenant le barycentre de l'ensemble des polygones pondérés par leurs aires respectives (figure II.15).

Dans ce cas, la valeur numérique sera obtenue de la manière suivante

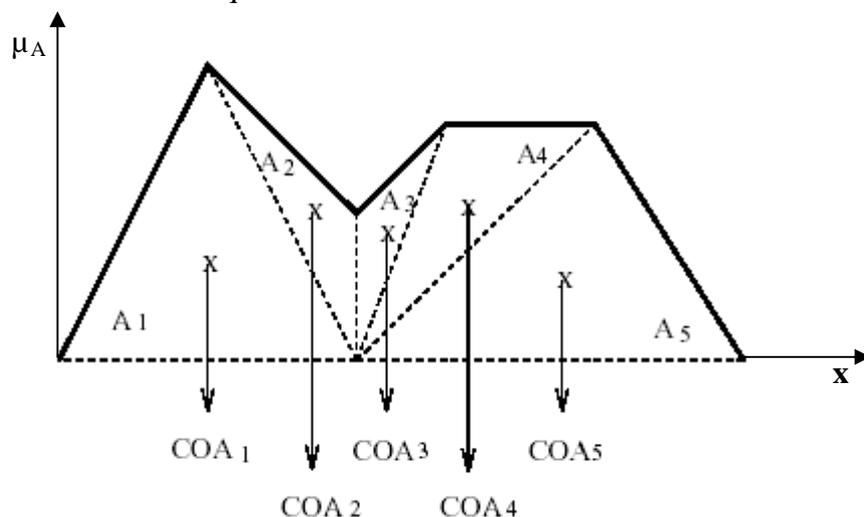


Fig. II. 15 : calcul de centre de gravité dans le cas de fonctions d'appartenances simples

$$valeur = \frac{\sum_{i=1}^n COA_i \times A_i}{\sum_{i=1}^n A_i}$$

où les COA_i et A_i représentent respectivement les barycentres et les aires des triangles composant la caractérisation floue résultant de la phase de composition.

- Méthode du maximum

Pour une variable de sortie donnée, si la fonction d'appartenance associée à la caractérisation floue de sortie ne possède qu'un maximum, la valeur numérique renvoyée est égale à ce maximum (figure II.16).

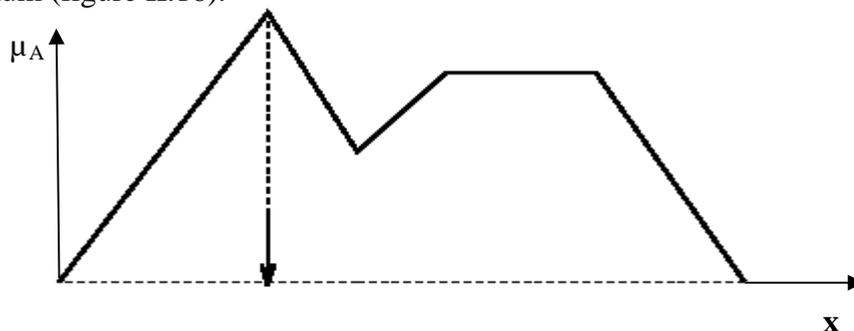


Fig. II. 16 : Défuzzification par la méthode du maximum

- Méthode de la moyenne des maxima

Pour une variable de sortie donnée, la valeur numérique délivrée est calculée comme la moyenne des valeurs appartenant à la caractérisation floue de sortie et ayant comme degré d'appartenance le maximum de la fonction caractéristique de cette caractérisation floue (figure II.17).

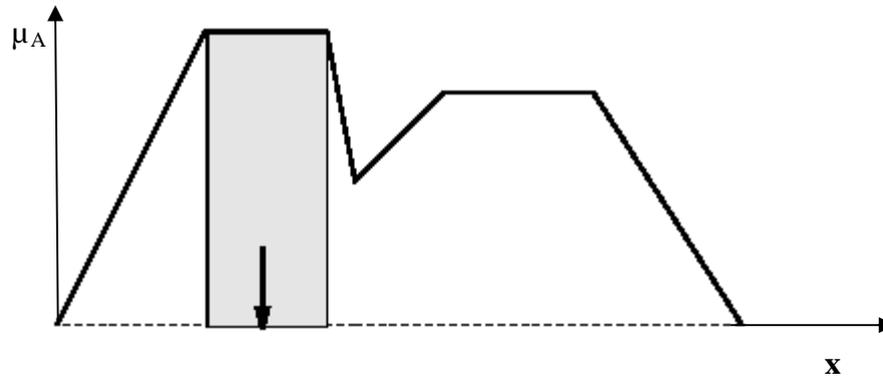


Fig. II. 17 : Défuzzification par la méthode de la moyenne des maxima

Une étude comparative de ces différentes approches peut être trouvée dans [3]. Il en ressort que la méthode de la moyenne des maxima assure de meilleures performances de transitions alors que la méthode du centre de gravité est plus régulière dans ses sorties. Les performances de la première méthode sont comparables à celles d'un système de relais à plusieurs niveaux. Les performances de la seconde méthode sont elles comparables à celles d'un contrôleur PI. Un contrôleur utilisant la méthode du centre de gravité garantit une erreur quadratique moyenne inférieure à celle d'un contrôleur utilisant la méthode de la moyenne des maxima. Cette dernière est cependant plus performante que la méthode du maximum.

II.4. LES MODELE FLOUS

Les modèles (ou systèmes) flous peuvent être classé en deux types: de MAMDANI et le modèle de TAKAGI & SAGINO.

II.4.1 Le Modèle flou de MAMDANI:

La logique floue à été appliquée pour la première fois dans la commande des système en 1974 par MAMDANI.

Le modèle proposé par MAMDANI trouve son application dans la manipulation des systèmes complexes non linéaire. Il utilise la méthodologie à base des règles. Considérons un système d'écrit par la relation $U = F(V, W)$.

U: est la variable de sortie (ou conséquence)

V, W : sont les variable d'entrée (ou antécédentes).

Dans cette modélisation floue, nous représentons cette relation par l'ensemble R des règles floues (IF- then).

IF V is A_i **AND** W is B_i **then** U is D_i

$A_i, B_i; D_i$ sont les ensembles floues sur les espaces X, Y et Z.

La méthode de la déduction floue de Mamdani est la méthodologie la plus générale du point de vu flou.

II. 4.2 La modèle floue de Takigi – sagino:

Takagi et Suger ont introduit une nouvelle approche pour modéliser un système flou.

La différence principale entre le type d'inférence flou de Mamdani et le type Sugeno est que les conséquences de chaque règle sont des valeurs numériques obtenues par combinaisons linéaire (dans le cas le plus simple) des variables des prémisses

La sortie du modèle floue est obtenue en combinant les conséquences de chaque règle.

Une règle floue typique du modèle floue de sugeno est ainsi de la forme:

règle i:

IF $x_1(k)$ is A_{Li} **AND** ...**AND** $x_n(k)$ is A_{ni} **THEN** $x_{i+1}(k) = A_i x(k) + B_i u(k)$

Pour $i = 1, 2, 3, \dots, r$

Ou: $x^T(k) = [x_1(k), x_2(k), \dots, x_n(k)]$

$$\mathbf{u}^T(k) = [\mathbf{u}_1(k), \mathbf{u}_2(k), \dots, \mathbf{u}_m(k)]$$

r est le nombre des règles flous **IF THEN**; $\mathbf{x}_i(k+1)$ est la sortie de la 1^{ère} règle if- then; A_{ij} est l'ensemble flou.

Etant donné le couple $(\mathbf{x}(k), \mathbf{u}(k))$.

La sortie finale du système flou est:

$$x(k+1) = \frac{\sum_{i=1}^r w_i(k) \{A_i x(k) + B_i u(k)\}}{\sum_{i=1}^r w_i(k)}$$

$$\text{Ou: } w_i(k) = \prod_{j=1}^r A_{ij}(x_j(k))$$

A_{ij} est le degré d'appartenance de $x_j(k)$

Le système libre est défini par:

$$x(k+1) = \frac{\sum_{i=1}^r w_i(k) A_i x(k)}{\sum_{i=1}^r w_i(k)}$$

II.5. Applications en robotique mobile

Un certain nombre d'applications utilisant la logique floue en robotique mobile existent dans la littérature. Nous donnons ci-après un modeste aperçu de ces applications.

II.5.1. Voiture floue de Sugeno

La voiture de Sugeno est capable de se garer automatiquement dans un garage [3]. Les règles du contrôleur où proviennent d'une modélisation des actions humaines dans une telle situation. Les entrées du contrôleur sont au nombre de trois : la distance au mur de devant (x), la distance au mur latéral (y) et l'angle θ que fait l'axe longitudinal du véhicule, par rapport à un repère associé à la scène.

Les sorties sont également au nombre de trois, à savoir : le braquage en marche avant, le braquage en marche arrière et la vitesse du véhicule.

Dix-huit règles permettent de contrôler le braquage dans les mouvements en avant, seize dans les mouvements en arrière. Un exemple de règle pour le contrôle du braquage dans le cas d'une marche avant est :

$$\text{If } x \text{ is } A; y \text{ is } B; \theta \text{ is } C \text{ then } f = p_0 + p_1x + p_2y + p_3.$$

L'identification des paramètres p_i résulte d'un apprentissage supervisé sur des échantillons collectés lors de manœuvres effectuées par des humains .

II.5.2. Métro de Sandai

Hitachi Ltd. a développé un système de contrôle du métro de Sandai (Japon). Ce contrôleur consiste à prédire les performances de chaque commande de contrôle candidate et sélectionne la plus appropriée en se basant sur une expérience humaine. Plus précisément, ce système est constitué de deux bases de règles prenant principalement en compte la sécurité, le confort, le suivi d'une vitesse de consigne, la consommation d'énergie et le temps. La première base de règles (CSC : constant speed control) consiste à effectuer un asservissement en vitesse assurant le démarrage du métro et le maintien à une vitesse de croisière. La seconde base de règles (TASC : Train Automatic Stop Control.) régule la vitesse du convoi afin que celui-ci s'arrête à une position donnée dans la station. Chacune des deux bases de règles contient douze règles floues.

Le contrôle est effectué tous les 100ms. Le système se comporte d'une manière analogue à un contrôle humain; il s'avère cependant supérieur à un contrôleur de type PID en termes de précision d'arrêt, consommation d'énergie et confort de condui

II.5.3. Autres Applications

Takeuchi et al. Proposent une méthode de navigation d'un robot mobile où les informations perceptives sont délivrées par une caméra CCD. Une analyse des images permet de déterminer des espaces libres entre des obstacles. La logique floue est utilisée pour déterminer, en fonction des paramètres de position, de largeur et de longueur des passages libres, le changement de direction à effectuer pour rallier un but. Les auteurs utilisent des informations ultrasonores pour la navigation d'un robot mobile. L'architecture du contrôleur s'appuie sur un ensemble de modules (Atteinte d'un point d'arrivée, Evitement des obstacles fixes, Evitement des obstacles mobiles, Demi-tour, . . .) activés par un routeur, en fonction de la carte de proximité associée aux capteurs ultrasons.

II.6. Caractéristique Avantage et Limitation de la logique floue

- **Caractéristiques:**

- la connaissance est structurée en règles linguistiques
- les traitements sont numériques.

- **Avantage:**

- Facilité de construction et d'interprétation des règles: les règles sont formulées de manière naturelle par les experts, comme en symbolique.
- Interpolation numérique entre les règles: lorsque deux règles déclenchent en même temps.
- Intégration de la connaissance à priori.
Possibilité de combiner des règles floues et règles logiques.

- **Limitations:**

- Construction manuelle des règles suivant l'intuition de l'opérateur ou connaissances de l'expert.
- Optimisation manuelle des fonctions d'appartenance: nombre de termes linguistique, choix de la forme des fonctions d'appartenance, positionnement du centre, largeur...etc.....

- Optimisation de l'importance de chaque de chaque règle: le concepteur peut donner facilement de l'importance à une règle en lui en affectant un poids, mais ce poids est choisi de manière intuitive comme les paramètres des fonctions d'appartenance.

II.7. CONCLUSION

La logique floue, de par la quantité des travaux de recherche et des réalisations existantes, présente un certain nombre de points forts. En premier lieu, elle simplifié la conception d'un contrôleur dans la mesure où l'on décrit directement le fonctionnement de celui-ci sur le système contrôlé. Dans le cas de systèmes complexes, cela représente un avantage indéniable par rapport à la plupart des autres méthodes qui s'attachent à modéliser les systèmes. De plus, un contrôleur flou est relativement ouvert à travers l'utilisation de règles linguistiques écrites en langage naturel; il est possible de gérer de manière assez souple ces règles.

Cependant, le problème majeur réside dans le fait que l'on ne puisse pas prouver la stabilité d'un système flou, contrairement aux modèles mathématiques classiques (fonctions de Lyapunov). Cela peut présenter des problèmes certains dans des domaines fortement contraints. En conséquence, au lieu de se limiter à des comparaisons stériles entre contrôleurs flous et autres plus formels, il semble intéressant de s'acheminer vers des approches hybrides intégrant différents types de contrôleurs. Au sein d'une même application, celles-ci pourront mettre en exergue les avantages d'élaboration d'un contrôleur flou et ceux de vérifications (stabilité, contraintes logiques, contraintes temporelles, . . .) de contrôleurs plus "classiques". On peut ainsi imaginer des contrôleurs flous, chargés de tâches ne souffrant pas de contraintes de sécurité, cohabitant avec des contrôleurs dont le comportement doit être prouvé de manière formelle. C'est dans cette optique que semble travailler le constructeur automobile Renault. En ce qui concerne notre travail, nous considérons que notre contrôleur d'exécution de mouvements peut être entièrement codé par un contrôleur flou.

Chapitre III

Systeme flou type -2

III.1. Introduction:

Le concept de l'ensemble flou type 2 est introduit par [Zadeh] Comme extension de concept de l'ensemble flou ordinaire appelé ensemble flou type 1. Un ensemble flou type 2 est caractérisé par une fonction d'appartenance floue. C'est à dire, la valeur d'appartenance ou le degré d'appartenance de chaque élément de l'ensemble est un ensemble flou dans $[0,1]$. De tels ensembles, peuvent être utilisés dans les situations où nous avons de l'incertitude dans les valeurs d'appartenance eux mêmes.

L'incertitude peut être soit dans la forme de la fonction d'appartenance ou dans l'un de ses paramètres.

Considérons la transition des ensembles ordinaires vers les ensembles flous lorsque ne nous pouvons pas déterminer le degré d'appartenance d'un élément à un ensemble par 0 ou 1, on utilise les ensembles flous type 1. De la même façon, lorsque nous ne pouvons pas déterminer les fonctions d'appartenance floues par des nombres réels (crisp) dans $[0,1]$ on utilise alors les ensembles flous type 2. Lorsque ne pouvons pas déterminer dans $[0,1]^{[0,1]}$ on utilise les ensembles flou type 3.....

Idéalement nous aurons besoin d'utiliser des ensembles flous type ∞ pour compléter la représentation de l'incertitude, mais cela est pratiquement impossible, parce que nous devant utiliser des ensembles flous de types finis.

Les ensembles flous type 1 peuvent être donc considérés comme une approximation du premier ordre de l'incertitude, alors que les ensembles flous type 2 seront considérés comme approximation de deuxième ordre.

III.2. Les ensembles flous type _ 2 [24][25][26]

III.2. 1. Définitions

▪ **Définition 1:**

Un ensemble floue type 2 noté \tilde{A} est caractérisé par une fonction d'appartenance $\mu_{\tilde{A}}(x, u)$, ou $x \in X$ et $J_x \subseteq [0,1]$

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) / \forall x \in X, J_x \subseteq [0,1]\} \dots\dots\dots(III. 1)$$

Ou $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. \tilde{A} Peut être écrit comme suit :

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0,1] \dots\dots\dots(III. 2)$$

Ou $\int \int$ c'est l'union de tous les éléments du produit cartésien sur x et u avec $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$.

- pour l'univers de discours discret \int est remplacé par \sum .

A chaque valeur de x de $\mu_{\tilde{A}}(x, u), \forall u \in J_x$. J_x est une fonction d'appartenance type1 appelé l'appartenance primaire de la variable primaire x.

On remarque de cette définition que la fonction d'appartenance type2 est tri dimensionnelle.

▪ **Définition 2:**

A chaque valeur de x, notée $x = x'$ le plan dimensionnel dont les axes sont : u et $\mu_{\tilde{A}}(x', u)$ est appelé tranche verticale de $\mu_{\tilde{A}}(x, u)$. Une fonction d'appartenance secondaire est une tranche verticale de $\mu_{\tilde{A}}(x, u)$.

On a : $\mu_{\tilde{A}}(x = x', u)$ pour $x' \in X$ et $\forall u \in J_{x'} \subseteq [0,1]$ c – à – d:

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u \quad J_{x'} \subseteq [0,1] \dots\dots (III.3)$$

Remarque: l'ensemble flou type2 \tilde{A} est l'union de toutes les fonctions d'appartenance secondaires.

▪ **Définition 3:**

Le domaine d'incertitude (DOU), pour l'ensemble flou type 2 \tilde{A} , est l'union de toutes les appartenances primaires de \tilde{A} .

$$DOU(\tilde{A}) = \bigcup_{x \in X} J_x \dots\dots\dots (III.4)$$

▪ **Définition 4:**

L'amplitude d'une fonction d'appartenance secondaire est appelée degré d'appartenance secondaire.

Dans (1) $\mu_{\tilde{A}}(x', u')$ ($x' \in X, u' \in J_{x'}$) est le degré secondaire

▪ **Définition 5:**

L'incertitude dans un ensemble flou type2 \tilde{A} ; est représentée par une région borné appelée l'empreinte de pas de l'incertitude (footprint of uncertainty) (Fou). c'est l'union de toutes les appartenances primaires.

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \dots\dots\dots (III.5)$$

▪ **Définition 6:**

Fonction d'appartenance supérieure et fonction d'appartenance inférieure sont des fonctions d'appartenance de type-1 et sont les frontières de FOU d'un ensemble flou \tilde{A} type2.

La fonction d'appartenance supérieure sera notée par $\bar{\mu}_{\tilde{A}}(x) \forall x \in X$, et la fonction d'appartenance inférieure sera notée par $\underline{\mu}_{\tilde{A}}(x) \forall x \in X$, alors nous avons :

$$\bar{\mu}_{\tilde{A}}(x) = \overline{FOU(\tilde{A})} \quad \forall x \in X \dots\dots\dots (III.6)$$

$$\underline{\mu}_{\tilde{A}}(x) = \underline{FOU(\tilde{A})} \quad \forall x \in X \dots\dots\dots(III.7)$$

▪ **Définition 7:**

Un ensemble flou type-2 intervalle est un ensemble flou type-2 dont toutes les fonctions d'appartenance secondaires sont des ensembles type-1 de formes intervalles, c.-à-d. $f_x(u) = 1, \forall u \in J_x \subseteq [0,1], \forall x \in X$.

$$u_{principale}(x) = \int_{x \in X} u / x \quad \text{où } f_x(u) = 1 \dots\dots\dots(III.8)$$

▪ **Définition 8:**

Supposons que chaque fonction d'appartenance secondaire d'un ensemble flou type 2 possède un seul degré d'appartenance unitaire. On définit alors une fonction d'appartenance principale comme l'union de tous ces points

$$u_{principale}(x) = \int_{x \in X} u / x \quad \text{où } f_x(u) = 1 \dots\dots\dots(III. 9)$$

III.2.2 Exemples des ensembles flous type_2

- **Exemple 1 :**

Soit un ensemble flou caractérisé par une fonction d'appartenance gaussienne de moyenne m et de déviation standard qui prend ses valeurs dans $[\sigma_1, \sigma_2]$ tel que:

$$\mu(x) = \exp\left[\frac{-1}{2}\left(\frac{x - m}{\sigma}\right)^2\right]; \quad \sigma \in [\sigma_1, \sigma_2] \dots\dots\dots (III.10)$$

A chaque valeur de σ_1 , nous allons avoir une courbe d'appartenance différente, (voir figure III 1). la valeur d'appartenance de n'importe quel x [excepte $x=m$] peut prendre plusieurs valeurs (dépendant de σ), ce que veut dire le degré d'appartenance n'est pas un nombre ordinaire, mais un ensemble flou.

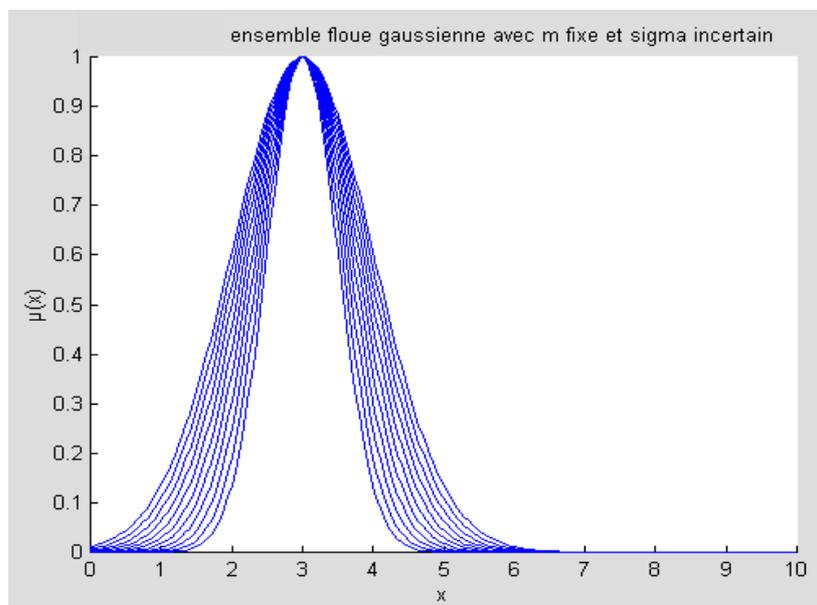


Fig. III.1: Ensemble flou type 2 représentant un ensemble flou type 1 avec déviation standard incertain

- **Exemple 2:**

Soit un ensemble flou type 1 avec une fonction d'appartenance gaussienne ayant une déviation standard fixe " σ ", mais une valeur moyenne incertaine prenant ses valeurs dans $[m1, m2]$ tel que:

$$\mu(x) = \exp\left[-\frac{1}{2} \left[\frac{x-m}{\sigma}\right]^2\right] \quad m \in [m1, m2] \dots\dots\dots (III.11)$$

$\mu(x)$ Est un ensemble flou. Voir la figure (III .2)

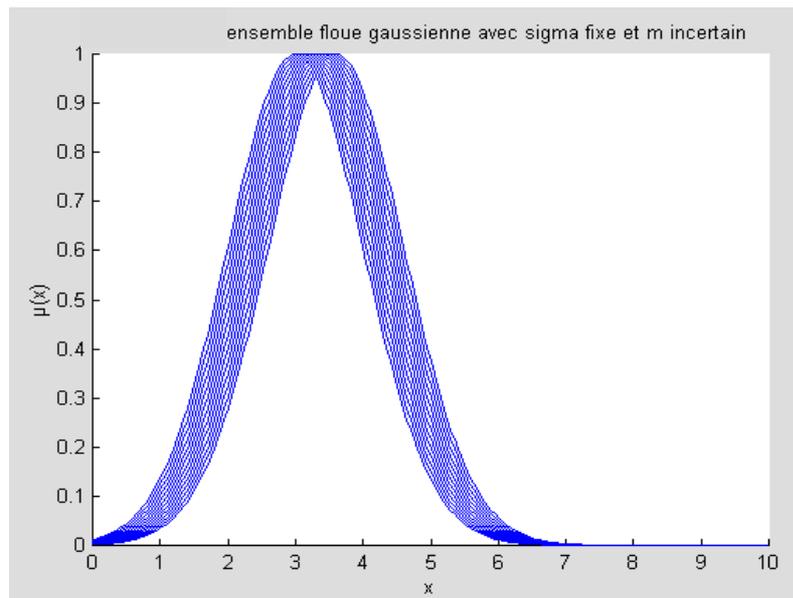


Fig. III. 2: Ensemble floue type 2 représentant l'ensemble flou type 1 avec valeur moyenne incertaine.

- **Exemple 3:**

Soit un ensemble floue type 1 caractérisé par une fonction d'appartenance gaussienne (valeur moyenne M et déviation standard σ_x donnant une seule valeur réelle d'appartenance $m(x)$ pour chaque $x \in X$ tel que:

$$m(x) = \exp \frac{-1}{2} \left[\frac{x - m}{\sigma_x} \right]^2 \dots\dots\dots \text{(III.12)}$$

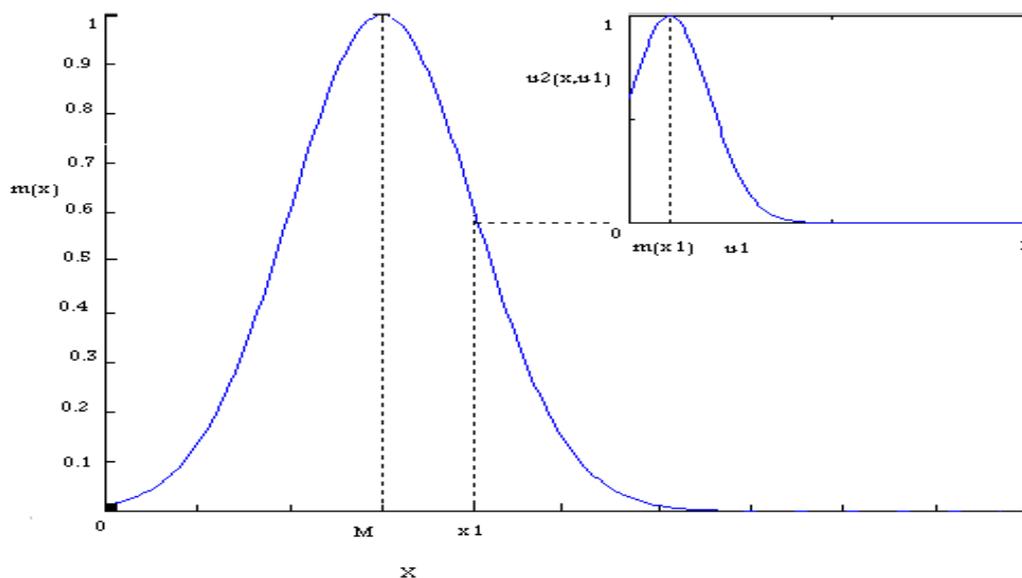


Fig. III.3: Gaussienne $m(x)$ et le degré d'appartenance de type gaussienne
Correspondant à $x = x_1$

La figure (III. 3) représente une telle faction d'appartenance .Imaginons maintenant que cette appartenance de x et un ensemble floue. Appelant les éléments du domaine de cet ensemble "appartenance primaire de x " (notées par μ_1) et les valeurs d'appartenance de ces appartenances primaires "appartenances secondaires de x " (notées $\mu_2(x, \mu_1)$).

Ainsi, pour un certain x donné, nous avons un ensemble floue type 1 dont les éléments de son domaine sont les appartenances primaires de x et les degrés d'appartenance correspondants sont les appartenances secondaires de x . Si nous supposons que les appartenances secondaires sont des gaussiennes de moyenne $m(x)$ et de déviation standard σ_m , comme le représente la figure (III 3), on peut alors représenter la fonction d'appartenance secondaire pour chaque x comme suit :

$$\mu_2(x, \mu_1) = \exp \frac{-1}{2} \left[\frac{\mu_1 - m(x)}{\sigma_m} \right]^2 \dots\dots\dots (III.13)$$

ou $\mu_1 \in [0,1]$ et m comme dans (3).

Les équations (III. 3) et (III. 4) peuvent être combinées comme suit :

$$\mu_2(x, \mu_1) = \exp \frac{-1}{2} \left[\frac{\mu_1 - \exp \frac{-1}{2} \left(\frac{x - m}{\sigma_x} \right)^2}{\sigma_m} \right]^2 \dots\dots\dots(III.14)$$

Donc de l'équation (III 14) nous pouvons considérer qu'une fonction d'appartenance secondaire comme fonction à deux variables x et μ_1 .le degré d'appartenance $\mu(x)$ de chaque x, représentant toutes les appartenances primaires ainsi que leurs appartenances secondaires correspondantes qui peuvent être écrits comme suit :

$$\mu(x) = \int_{\mu \in [0,1]} \mu_2(x, \mu_1) / \mu_2 \quad x \in X \dots\dots\dots(III.15)$$

III.2.3. Opération sur les ensembles floue type_2[26][27][28]

Soient deux ensembles flous type2 \tilde{A} et \tilde{B} dans un univers de discours X. avec les fonctions d'appartenance $\mu_{\tilde{A}}(x)$ et $\mu_{\tilde{B}}(x)$ et $J \subseteq [0,1]$

$$\mu_{\tilde{A}}(x) = \sum f_x(u_i) / u_i$$

Pour chaque x on a :(III.16)

$$\mu_{\tilde{B}}(x) = \sum g_x(w_i) / w_i$$

Respectivement ou: $u_i, w_i \in J$

- **Union floue type-2**

L'union de l'ensemble floue type2 est définie comme suit:

$$\tilde{A} \cup \tilde{B} \Leftrightarrow \mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \cup \mu_{\tilde{B}}(x)$$

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \sum_{i,j} (f_x(u_i) * g_x(w_j)) / u_i \vee w_j \dots \dots \dots (III.17)$$

- **Intersection floue type-2**

$$\tilde{A} \cap \tilde{B} \Leftrightarrow \mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \cap \mu_{\tilde{B}}(x)$$

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \sum_{i,j} (f_x(u_i) * g_x(w_j)) / u_i * w_j \dots \dots \dots (III.18)$$

- **Complément floue**

$$\bar{\tilde{A}} \Leftrightarrow \bar{\mu}_{\tilde{A}}(x) = -\bar{\mu}_{\tilde{A}}(x) = \sum_i f_x(u_i) / (1 - u_i) \dots \dots \dots (III.19)$$

Ou : \vee représente la T norme

* représente la T norme (utilisent généralement Min ou Produit)

\sum : indique l'union logique.

On utilise les définitions dans III.17, III.18 et III. 19 on examine les opérations d'union (join) et intersection (meet) et la négation.

Union et Intersection (Join et Meet) sous la T norme:

- **Théorie 1 :**

Soient F et G deux ensembles flous type1 normale, convexes caractérisés par les fonctions d'appartenances f et g respectivement.

Données : $v_0 \in \mathfrak{R}$ et $v_1 \in \mathfrak{R}$ avec $v_0 \leq v_1$ et $f(v_0) = g(v_1) = 1$.

Les fonctions d'appartenance de Join et de Meet utilisent Max T norme et Min T norme, peuvent être exprimée comme suit:

$$\mu_{(\tilde{F} \cup \tilde{G})} = \left. \begin{array}{ll} f(\theta) \wedge g(\theta) & \theta < \nu_0 \\ g(\theta) & \nu_0 \leq \theta \leq \nu_1 \\ f(\theta) \vee g(\theta) & \theta > \nu_1 \end{array} \right\} \dots\dots\dots \text{(III. 20)}$$

$$\mu_{(\tilde{F} \cap \tilde{G})} = \left. \begin{array}{ll} f(\theta) \vee g(\theta) & \theta < \nu_0 \\ f(\theta) & \nu_0 \leq \theta \leq \nu_1 \\ f(\theta) \wedge g(\theta) & \theta > \nu_1 \end{array} \right\} \dots\dots\dots \text{(III. 21)}$$

▪ **Join sous la T norme produit :**

- **Théorie 2:**

Supposons \tilde{F} et \tilde{G} deux ensembles floues réels de type1 normal, convexes, caractérisés par les fonctions d'appartenances f et g respectivement. On a : $\nu_0 \in \mathfrak{R}$ et $\nu_1 \in \mathfrak{R}$ avec $\nu_0 \leq \nu_1$ et $f(\nu_0) = g(\nu_1) = 1$. On a alors la fonction d'appartenance de Join entre \tilde{F} et \tilde{G} utilisant Max t-conorme et t- norme produit peut être exprimée par :

$$\mu_{(\tilde{F} \cup \tilde{G})} = \left. \begin{array}{ll} f(\theta)g(\theta) & \theta < \nu_0 \\ g(\theta) & \nu_0 \leq \theta \leq \nu_1 \\ f(\theta) \vee g(\theta) & \theta > \nu_1 \end{array} \right\} \dots\dots\dots \text{(III. 22)}$$

- **Meet sous la T norme produit :**

Soit n ensembles flous gaussienne $\tilde{F}_1, \tilde{F}_2, \dots, \tilde{F}_n$ avec des moyennes m_1, m_2, \dots, m_n et des déviations standards $\sigma_1, \sigma_2, \dots, \sigma_n$. Alors

$$\mu_{\tilde{F}_1 \cap \tilde{F}_2 \cap \dots \cap \tilde{F}_n}(\theta) \approx e^{-\frac{1}{2} \left(\frac{\theta - m_1 m_2 \dots m_n}{\sigma} \right)^2} \dots \dots \dots \text{(III. 23)}$$

Ou : (i=1, n)

$$\sigma = \sqrt{\sigma_1^2 \prod_{i:i \neq 1} m_i^2 + \dots + \sigma_j^2 \prod_{i:i \neq j} m_i^2 + \dots + \sigma_n^2 \prod_{i:i \neq n} m_i^2}$$

III.2.4. Type d'ensembles flous type 2

III.2.4.1 Ensemble floue type 2 Gaussien:

Dans ce type d'ensemble, le degré d'appartenance de chaque point est un ensemble type 1 gaussien dont le domaine de définition est inclus dans l'intervalle [0,1]. Notons qu'il n'est pas nécessaire que la fonction d'appartenance principale soit aussi gaussienne.

III.2.4.2 Ensemble floue type2 intervalle

Dans ce type d'ensemble le degré d'appartenance de chaque point est un ensemble ordinaire dont le domaine de définition est inclus dans l'intervalle [0,1], dans ce cas toute les appartenances secondaire sont égale à 1.

Notant que malgré que chaque degré d'un ensemble floue type 2 intervalle est un ensemble ordinaire, l'ensemble lui-même est de type 2 parce que les degrés d'appartenances sont des ensembles est pas des nombres ordinaires.

III.2.4.3 Ensemble type triangulaire:

Dans ce type d'ensemble le degré d'appartenance de chaque point est un ensemble type 1 triangulaire dont le domaine de définition est inclus dans l'intervalle [0,1].

- Exemple 4:

Soient deux ensembles flous \tilde{A}_1 et \tilde{A}_2 définis comme suit:

$$\bar{A}_1 = \exp(-0.5*(x-4).^2); \quad \underline{A}_1 = 0.75*\exp(-0.5*(x-4).^2);$$

$$\bar{A}_2 = \exp(-0.5*(x-2).^2); \quad \underline{A}_2 = 0.75*\exp(-0.5*(x-2).^2);$$

Leurs unions et intersection sont représentées dans la figure (III. 4)

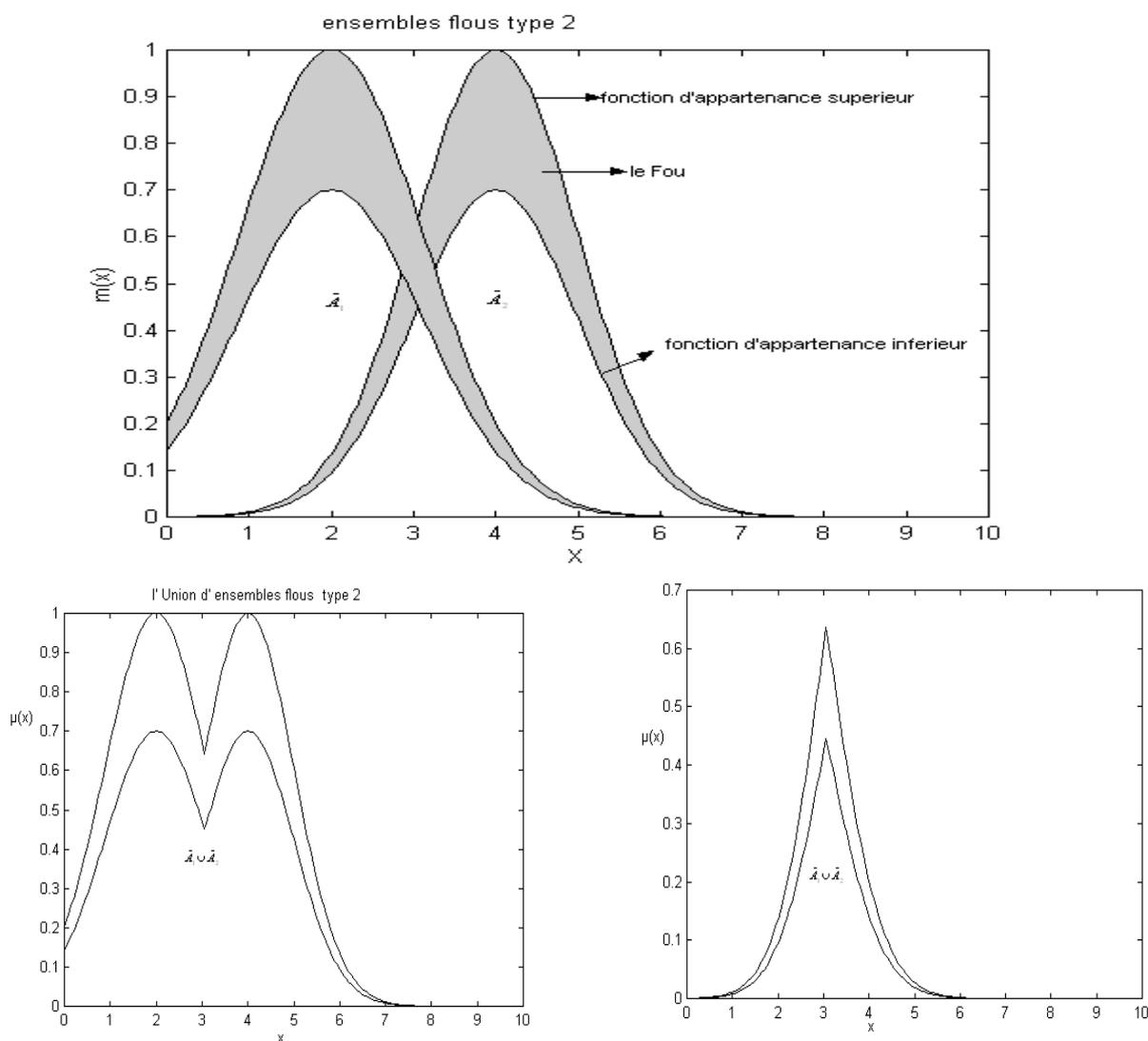


Fig. III .4 : Union et intersection des ensembles flous type 2

III.3. Structure générale d'un système flou type_ 2

La structure générale d'un système flou type 2 se compose de cinq éléments de base: la fuzzification, la base de règles, le moteur d'inférence flou, réduction de type et la défuzzification. La structure est représentée dans la figure (III 5) ci-dessous :

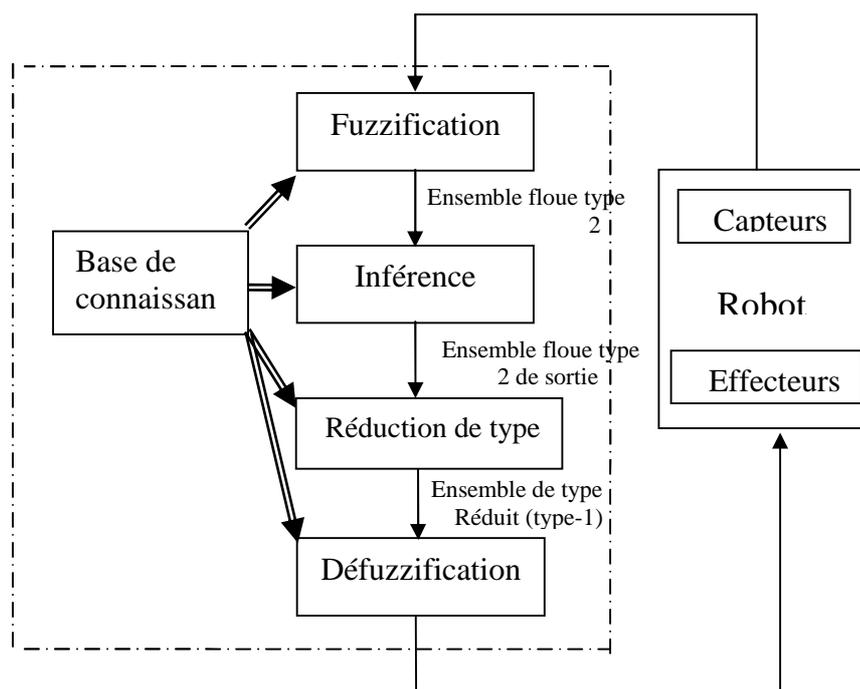


Fig. III. 5 : Architecture d'un système flou type 2

III.3. 1. Fuzzification:

Le bloc de fuzzification à un vecteur d'entrée à p entrées $x = (x_1, x_2, \dots, x_p)^T \in X_1, X_2, \dots, X_p = X$ qui fuzzifié à l'entrée des ensembles flous type 2 \tilde{A}_x [29][31].

On utilise la fuzzification singleton parce qu'elle est rapide et facile à calculer et mieux adapter pour l'opération du robot. Dans la fuzzification singleton, l'entrée floue est un point singulier possédant une valeur d'appartenance unitaire (C à d : \tilde{A}_x est un

fuzzification singleton type 2 Si $\mu_{\tilde{A}_x}(x) = 1/1$ pour $x=x'$ et $\mu_{\tilde{A}_x}(x) = 1/0$ pour $\forall x \neq x'$ [31]

III.3. 2. Base de Règles:

De [31] Les règles floues sont de la même forme que le système floue type 1 sauf que les fonctions d'appartenances des prémisses et des conséquences seront représentées par des ensembles flous type 2 intervalles.

Nous considérons un contrôleur floue type-2 « FLC type-2 » d'un robot mobile à p entrées $x_1 \in X_1, \dots, x_p \in X_p$ et c sorties $y_1 \in Y_1, \dots, y_c \in Y_c$, la i^{eme} règle du FLC multiple entrées multiple sorties (MIMO) peut être écrit comme suit:

$$R^i \text{ MIMO IF } x_1 \text{ is } \tilde{F}_1^i \text{ and } \dots \text{ and } x_p \text{ is } \tilde{F}_p^i \dots \dots \dots \text{ (III. 24)}$$

$$\text{then } y_1 \text{ is } \tilde{G}_1^i \text{ and } \dots \text{ and } y_c \text{ is } \tilde{G}_c^i \quad i = 1 \dots M$$

Où M représente le nombre des règles flous dans la base de règle.

De [31] La base de règles de type 2 a la même structure que type 1 on peut donc considérer un groupe de multiple entrées multiple sorties « MIMO » comme un (MISO) R_{KMISO}^i ou , R_{KMISO}^i est la règle qui relie les multiples entrées 'p' et la sortie singulier 'k' ou $k=1 \dots c$.

III.3. 3. Le Moteur d'inférence floue:

Le moteur d'inférence associé les règles et donne un organisme à partir des ensembles type2 d'entrées aux ensembles type 2 de sortie [31]. Dans le moteur d'inférence les multiples antécédents dont les règles sont reliés entre eux utilisant l'opération Meet.

Les degrés d'appartenance dans les ensembles d'entrées sont reliées avec celles dont les ensembles de sortie utilisent la composition sup-star, les multiples règles sont reliées et utilisent l'opération Join.

Dans la base de règle floue de MISO avec M règles. Chaque règle à p entrées $x_1 \in X_1, \dots, x_p \in X_p$ et une seule sortie $y_k \in Y_k$ peut être écrit comme suit:

$$R_{k \text{ MISO}}^i : \tilde{F}_1^i \times \dots \times \tilde{F}_p^i \rightarrow \tilde{G}_k^i = \tilde{A}^i \rightarrow \tilde{G}_k^i \quad i = 1 \dots M \dots \dots \text{(III.25)}$$

$R_{k \text{ MISO}}^i$ est décrit par la fonction d'appartenance :

$$\mu_{R^i}(x, y_k) = \mu_{R^i}(x_1, \dots, x_p, y_k) \quad [31]$$

$$\mu_{R^i}(x, y_k) = \mu_{\tilde{A}^i \rightarrow \tilde{G}_k^i}(x, y_k)$$

$$\mu_{R^i}(x, y_k) = \mu_{\tilde{F}_1^i}(x_1) \cap \dots \cap \mu_{\tilde{F}_p^i}(x_p) \cap \mu_{\tilde{G}_k^i}(y_k)$$

$$\mu_{R^i}(x, y_k) = \left[\bigcap_{\alpha=1}^p \mu_{\tilde{F}_\alpha^i}(x_\alpha) \right] \cap \mu_{\tilde{G}_k^i}(y_k) \dots \dots \dots \text{(III. 26)}$$

Comme nous avons utilisé la fuzzification singleton ; L'ensemble floue type 2 d'entrées \tilde{A}_c contient un seul élément x' et chaque $\mu_{\tilde{X}_\alpha}(x_\alpha)$ possède un degré d'appartenance unitaire à un point $x_\alpha = x'_\alpha$ est zéro ailleurs.

Dans notre FLC type-2 intervalle, nous avons utilisé l'opération Meet sous t-norme produit donc le résultat d'entrée et les opérations d'antécédent qui est contenus dans l'ensemble activée $\bigcap_{\alpha=1}^p \mu_{\tilde{F}_\alpha^i}(x'_\alpha) \equiv F^i(x')$, est un ensemble type-1 intervalle, comme suit:[31]

$$F^i(x') = \left[\underline{f}^i(x'), \bar{f}^i(x') \right] \equiv \left[\underline{f}^i, \bar{f}^i \right] \dots \dots \dots \text{(III. 27)}$$

Ou $\underline{f}^i(x')$ et $\bar{f}^i(x')$ peuvent être écrit comme suit ou * est opération produit :

$$\underline{f}^i(x') = \underline{\mu}_{\bar{F}_1^i}(x'_1) * \dots * \underline{\mu}_{\bar{F}_p^i}(x'_p) \dots \dots \dots \text{(III. 28)}$$

$$\bar{f}^i(x') = \bar{\mu}_{\bar{F}_1^i}(x'_1) * \dots * \bar{\mu}_{\bar{F}_p^i}(x'_p) \dots \dots \dots \text{(III. 29)}$$

III.3. 4. Réduction de type:

La réduction de type est proposée par Karnik et Mendel[15][11][12][38], s'appelle « réduction de type » puisque cette opération transforme un ensemble flou type-2 de la sortie du moteur d'inférence à un ensembles type 1, et on appelle l'ensemble flou type-1 obtenu « ensemble de type réduit ». L'ensemble de type réduit doit ensuite être défuzzifié pour obtenir les sorties crisp (nombre ordinaire) les quelles sont envoyées au moteurs du robot mobile.

Come dans [30] Nous emploierons le centre de type réduction parce qu'elle a une complexité informatique raisonnable. L'ensemble de type réduit utilisant l'ensemble de type réduction, s'exprime par la relation suivante:[31]

$$Y_{\cos}(x)_k = [y_{lk}, y_{rk}] = \int_{y_k^i \in [y_{lk}^i, y_{rk}^i]} \dots \int_{y_k^M \in [y_{lk}^M, y_{rk}^M]} \dots \int_{f^i \in [f_{-}^i, f_{+}^i]} \dots \int_{f^M \in [f_{-}^M, f_{+}^M]} 1 \int \frac{\sum_{i=1}^M f^i y_k^i}{\sum_{i=1}^M f^i} \dots \dots \dots \text{(II 30)}$$

Ou $y_{\cos}(x)_k$: pour la k^{ème} sortie est un ensemble d'intervalle déterminé par son point le plus à gauche et y_{lk} le point le plus à droite $y_{rk}, i=1 \dots M$. M étant le nombre de règles.

Notons : f^i la force d'activation (degré d'activation) pour la i^{ème} règle qui est un ensemble type-1 intervalle déterminé par son point à gauche f_{-}^i et à droite f_{+}^i [30].

Le calcul de l'ensemble type réduit est divisé en deux étapes. La première étape consiste au calcul centroïde des ensembles type_2, des conséquences de chaque règle avant de commencer l'opération du FLC du robot.

La seconde étape produit chaque cycle opératoire pour calculer l'ensemble type réduit déffuzifié pour produire les sorties réelles pour le déclencheur.

III.3. 4. 1. Calcul de centres de conséquences de règle :

A chaque sortie k , pour $i^{\text{ème}}$ règle \tilde{G}_k^i l'ensemble type-2 l'intervalle de conséquence sera un des ensembles flous type-2 intervalle de sortie \tilde{G}_k^t représentant cette sortie. $t=1 \dots T$ ou T est le nombre d'ensemble flou de sortie affectée à cette sortie.

Si pour chaque sortie nous calculons les centroïde de tous les ensembles type-2 d'intervalle représentant cette sortie, puis le centre d'ensemble type-2 intervalle de conséquence, pour la $i^{\text{ème}}$ règle, y_k^i sera un des centres pré calculé des ensembles type 2 de sortie y_k^t qui correspond à la conséquence de règle.

Nous devant calculer tous les centroïde de y_k^i pour $t=1 \dots T$ en avance et mettre en marche l'opérateur du FLC du robot. Comme il est nécessaire pour le calcul de $Y_{\text{cos}}(x)_k$.

Le centroïde de $i^{\text{ème}}$ sortie de l'ensemble flou y_k^t est un ensemble d'intervalle type 1, déterminé par son point le plus à gauche y_{lk}^t et son point le plus à droite y_{rk}^t . Ceci peut être obtenu à partir de l'équation suivante:[31]

$$y_k^t = [y_{lk}^t, y_{rk}^t] = \int_{\theta_1 \in j_{y_1}} \dots \int_{\theta_z \in j_{y_z}} \dots \frac{1}{\frac{\sum_{z=1}^z y_z \theta_z}{\sum_{z=1}^z \theta_z}} \dots \dots \dots \text{(III.31)}$$

Pour calculer les centroïde de chaque sortie de l'ensemble nous emploierons la procédure itérative suivante:[24][31]

- **Procédure n° 1:** Procédé itératif pour calculer $y'_{rk} \ y''_{rk}$

En général on admet que les y_z sont arrangés dans l'ordre croissant, c'est-à-dire

$y_1 \leq y_2 \leq \dots y_z$, alors;

1. Initialiser θ_z pour $z=1 \dots z$ et calculons $y' = y(h_1 \dots h_z)$ utilisant l'équation (III 32).
2. Trouver e ($1 \leq e \leq z - 1$) tel que $y_e \leq y' \leq \dots y_{e+1}$.
3. Poser $\theta_z = h_z - \Delta_z$ pour $z \leq e$ et $\theta_z = h_z + \Delta_z$ pour $z \geq e + 1$ et calculer $y'' = y(h_1 - \Delta_1, \dots, h_e - \Delta_e, h_{e+1} + \Delta_{e+1}, \dots, h_z + \Delta_z)$ utilisant l'équation (III 32).
4. Vérifier Si $y'' = y'$; Si oui stop, y'' est la valeur maximale de $y(\theta_1, \dots, \theta_z) : y'_{rk}$; Si non aller à l'étape 5.
5. Poser y'' égale y' : aller à l'étape 2.

Utilisant cette procédure pour calculer y'_{rk} . Dans cette procédure pour la k^{ème} sortie nous discrétiserons chaque ensemble floue de sortie à Z points, $y_1 \dots y_z$ ou $z=(1 \dots Z)$, mettre $J_{y_0} = [L_z, R_z]$ et $\Upsilon = \frac{L_z + R_z}{2}$; $\Delta_z = \frac{R_z - L_z}{2}$.

La figure (III 5) présente pour chaque y_z la méthode pour trouver L_z, R_z, H_z , utilisée par la procédure itérative. Nous avons besoin de maximiser et minimiser y en fonction de $(\theta_1, \dots, \theta_z)$ pour calculer les points finales optimiser :[31]

$$y(\theta_1, \dots, \theta_z) = \frac{\sum_{z=1}^z y_z \theta_z}{\sum_{z=1}^z \theta_z} \dots \dots \dots (III.32)$$

Pour calculer y'_{rk} nous emploierons, la procédure itérative N°1

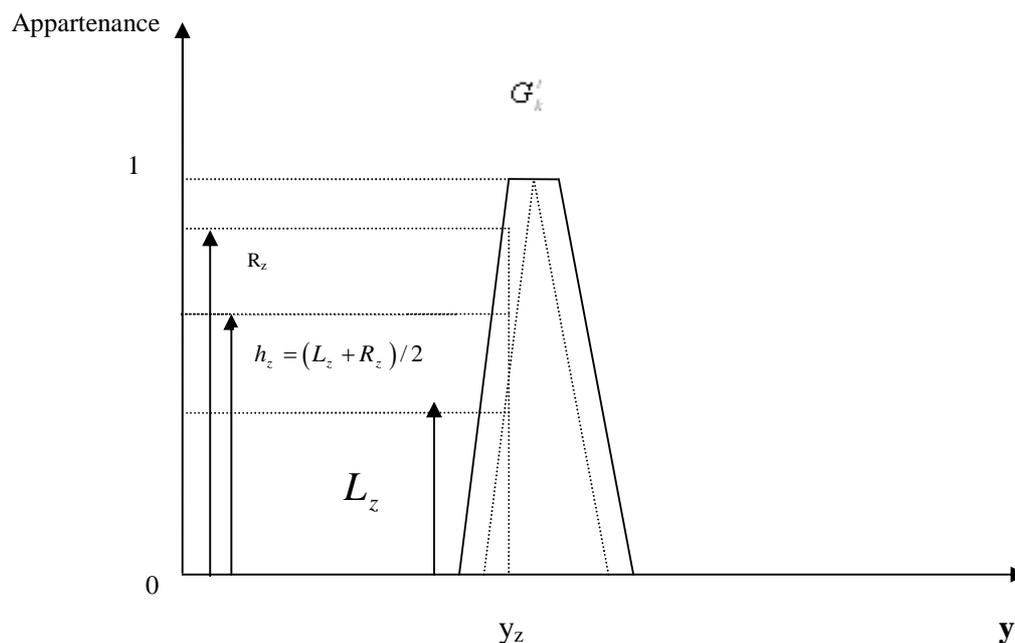


Fig. III. 6: Paramètres utilisés par chaque y_z à la procédure n° 1

La valeur y_{ik}^t peut être obtenue en utilisant la même procédure itérative n°1, mais en faisant un seul changement dans l'étape 3.

Posons $\theta_z = h_z + \Delta_z$ pour $z \leq e$ et $\theta_z = h_z - \Delta_z$ pour $z \geq e + 1$ et calculons $y'' = y(h_1 + \Delta_1, \dots, h_e + \Delta_e, h_{e+1} - \Delta_{e+1}, \dots, h_z - \Delta_z)$ utilisant l'équation (III 32).

Cette procédure itérative converge après au plus Z itérations pour trouver y_{rk}^t et le même nombre d'itérations pour trouver y_{ik}^t [31]. Une itération est le passage de l'étape 2 à l'étape 5 car l'étape 1 est une étape d'initialisation. Cependant; comme mentionné précédemment le calcul des centres de conséquences est fait seulement une fois avant que le robot commence à se déplacer et n'est pas une partie de cycle opératoire.

III.3. 4. 2. Calcul de l'ensemble type réduit:

A chaque sortie k afin de calculer $y_{\cos}(x)_k$ dans (III. 30) nous avons besoin du calcul de ces deux points finaux y_{lk} et y_{rk} .

- Ainsi pour chaque règle nous avons besoin d'attacher au degré d'activation f^i de centre de conséquence y_k^i de la $i^{\text{ème}}$ règle calculé dans l'étape précédente puis de [31] mettre les valeurs de f^i et y_k^i qui sont associés à y_{lk} notée par f_k^i et y_{lk}^i respectivement ainsi que les valeurs de f^i et y_k^i qui sont associés à y_{rk} notée par f_r^i et y_{rk}^i respectivement.

La formule (III.30) illustre ce fait

$$y_{lk} = \frac{\sum_{i=1}^M f_l^i y_{lk}^i}{\sum_{i=1}^M f_l^i} \dots\dots\dots(\text{III.33})$$

$$y_{rk} = \frac{\sum_{i=1}^M f_r^i y_{rk}^i}{\sum_{i=1}^M f_r^i} \dots\dots\dots(\text{III.34})$$

Afin de calculer y_{lk} , nous devons déterminer $\{f_l^i, i = 1 \dots M\}$ ou sont associés $\{y_{lk}^i, i = 1 \dots M\}$ et pour calculer y_{rk} nous devons déterminer $\{f_r^i, i = 1 \dots M\}$ ou sont associés $\{y_{rk}^i, i = 1 \dots M\}$ [31].

Ceci est fortement possible par l'emploi du procédé suivant qui utilise pour calculer y_{rk} une procédure itérative de quatre étapes ou la 1^{ère} étape est une étape d'initialisation [30][31].

- **Procédure n° 2:** Procédé itérative pour calculer y_{rk}

On admet en général que les prés calculés de y_{rk}^t sont arrangés dans un ordre croissant c'est-à-dire : $y_{rk}^1 \leq y_{rk}^2 \leq \dots \dots y_{rk}^M$ alors:

1. Calculer y_{rk} dans l'équation (III.34), initialiser par la mise de

$$f_r^i = (f_{-}^i + \bar{f}^i) / 2 \text{ pour } f_{-}^i \text{ et } \bar{f}^i \text{ calculés précédemment utilisant les}$$

équations (III. 28) et (III .29) respectivement puis mettre $y'_{rk} \equiv y_{rk}$.

2. Trouver R ($1 \leq R \leq M - 1$) connu que $y_{rk}^R \leq y'_{rk} \leq y_{rk}^{R+1}$.

3. Calculer y_{rk} dans l'équation (III. 32) utilisant $f_r^i = f_{-}^i$ pour $i \leq R$ et

$$f_r^i = \bar{f}^i \text{ pour } i > R \text{ et mis } y''_{rk} \equiv y_{rk} \text{ .}$$

4. Si $y''_{rk} \neq y'_{rk}$ alors aller à l'étape 5. Si $y''_{rk} \equiv y'_{rk}$ alors fin et poser

$$y''_{rk} = y_{rk} \text{ .}$$

5. Poser y'_{rk} égale y''_{rk} et retour à l'étape 2.

On peut remarquer que dans cette procédure le nombre R est très important.

Pour $i \leq R$ $f_r^i = f_{-}^i$ et quand $i > R$ $f_r^i = \bar{f}^i$.par conséquent y_{rk} dans (III 34) peut être écrit comme suit:

$$y_{rk} = \frac{\sum_{u=1}^R f_{-}^u y_{rk}^u + \sum_{v=R+1}^M \bar{f}^v y_{rk}^v}{\sum_{u=1}^R f_{-}^u + \sum_{v=R+1}^M \bar{f}^v} \dots \dots \dots \text{(III.35)}$$

La procédure pour calculer y_{lk} est la même que celle donnée pour calculer y_{rk} . On ne fait que juste remplacer y_{rk}^i par y_{lk}^i et dans l'étape 2 on trouve $L (1 \leq L \leq M - 1)$, On a

$$y_{lk}^L \leq y'_{lk} \leq y_{lk}^{L+1}$$

calculer y_{lk} dans (III.33) avec : $f_l^i = \bar{f}^i$ pour $i \leq L$ et $f_l^i = \underline{f}^i$ pour $i > L$, donc y_{lk} dans (III. 33) peut être écrit comme suit:

$$y_{lk} = \frac{\sum_{u=1}^L \bar{f}^u y_{lk}^u + \sum_{v=L+1}^M \underline{f}^v y_{lk}^v}{\sum_{u=1}^R \bar{f}^u + \sum_{v=L+1}^M \underline{f}^v} \dots\dots\dots(III.36)$$

Cette procédure sera employée chaque cycle opératoire pour calculer les ensembles type réduit, qui sont défuzzifiés pour donner les sorties crisp aux déclencheur du robot.

III. 5. Défuzzification:

A l'étape de réduction de type, nous prenons pour chaque sortie l'ensemble type réduit $y_{\cos}(x)_k$, déterminé par son point la plus gauche y_{lk} et son point la plus droite y_{rk} .

Nous défuzzifions l'ensemble d'intervalle en utilisant la moyenne de y_{lk} et y_{rk} et par conséquent, la sortie crisp défuzzifiée pour chaque sortie k est [30][31]:

$$y_k(x) = \frac{y_{lk} + y_{rk}}{2} \dots\dots\dots(III.37)$$

Exemple 5:

Dans cet exemple, nous allons calculer la réduction de type pour les trois ensembles flous type-2 intervalle, notés par \tilde{A}_1, \tilde{A}_2 et \tilde{A}_3 , représentés dans la figure(III.6) et donnés par :

$$\underline{A}_1 = 0.75 \exp\left(-0.5\left(\frac{x-2}{1.2}\right)^2\right) \qquad \bar{A}_1 = \exp\left(-0.5\left(\frac{x-2}{1.2}\right)^2\right)$$

$$\underline{A}_2 = 0.75 \exp(-0.5(x-6)^2)$$

$$\bar{A}_2 = \exp(-0.5(x-6)^2)$$

$$\underline{A}_3 = 0.75 \exp(-0.5(x-9)^2)$$

$$\bar{A}_3 = \exp(-0.5(x-9)^2)$$

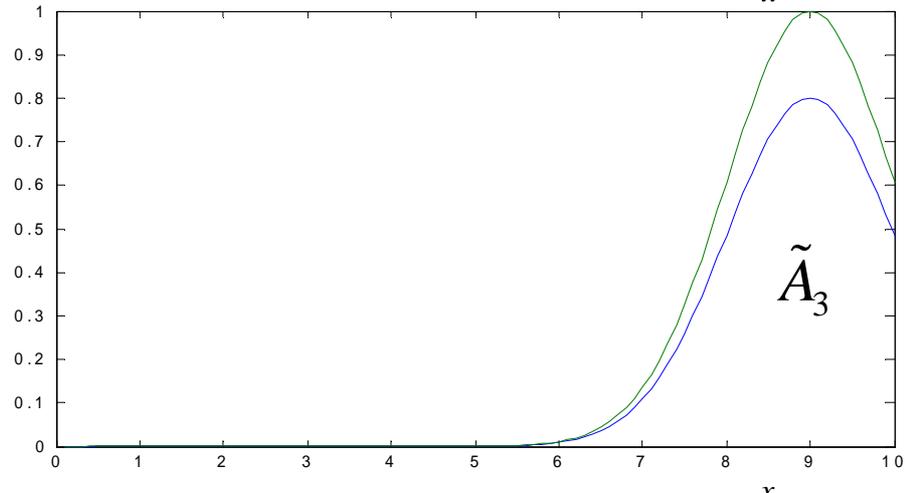
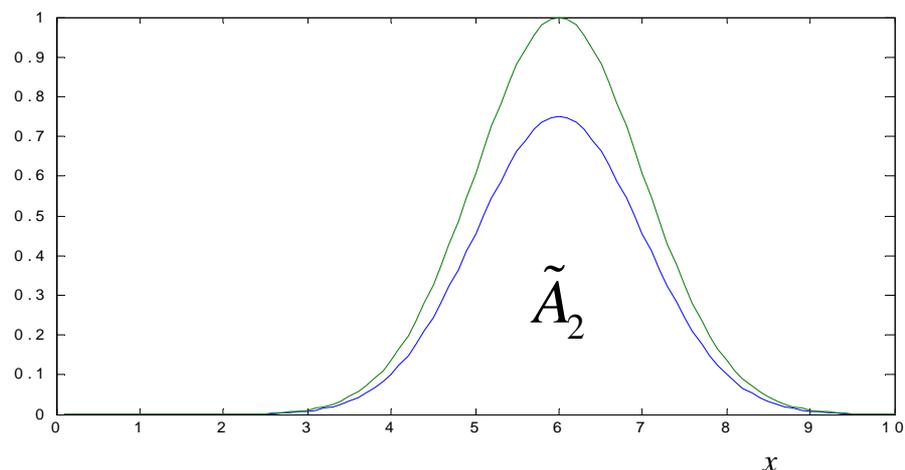
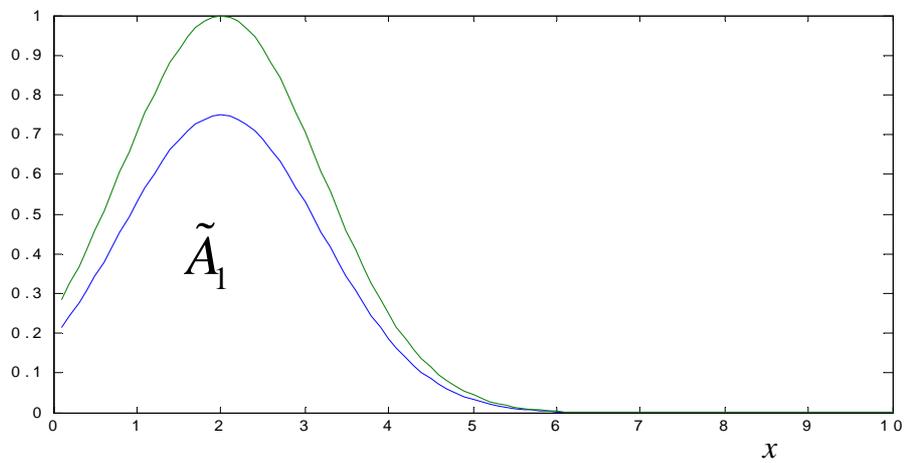


Fig. III.7: Fonctions d'appartenance floue type-2 définies dans l'exemple 5

Les résultats sont présentés dans le tableau 1 où chaque ensemble intervalle de type réduit est représenté par son centre et sa largeur

	Centre	Largeur
Ensemble \tilde{A}_1	2.1261	[1.9996 2.2525]
Ensemble \tilde{A}_2	6.0000	[5.8853 6.1147]
Ensemble \tilde{A}_3	8.7119	[8.6399 8.7839]

Tableau. 1 : Résultats de réduction de type de l'exemple 5

Exemple 6

Dans cet exemple nous allons illustrer l'utilisation de la méthode de réduction de type pour les ensembles flous type-2 intervalles. Soit un système flou de type-2 utilisant une t-norm et une inférence types produit, où les règles sont données par :

$$R^{(l)} : IF \ x \text{ is } \tilde{F}^{(l)}, \text{ THEN } y \text{ is } \tilde{G}^{(l)}$$

$$\text{où } x \text{ et } y \in [0, 20]$$

La figure 7 (a) et (b) montre l'allure des ensembles prémisses et conséquences. Chacun de ces ensembles est un ensemble type-2 intervalle qui peut être décrit par deux gaussiennes ayant la même moyenne et la même variance. Les deux gaussiennes ont des hauteurs différentes. La hauteur maximale atteinte par la plus haute gaussienne est unitaire, alors que celle atteinte par la plus basse est s . Si la moyenne de la variance d'une gaussienne est respectivement s et m , alors, le degré d'appartenance d'un point x' est

$$\text{l'intervalle} \left[s \exp\left(-0.5\left(\frac{x'-m}{\sigma}\right)^2\right) \quad \exp\left(-0.5\left(\frac{x'-m}{\sigma}\right)^2\right) \right]$$

Les valeurs de m et de σ pour tous les ensembles flous type-2 sont donnés dans le tableau 2.

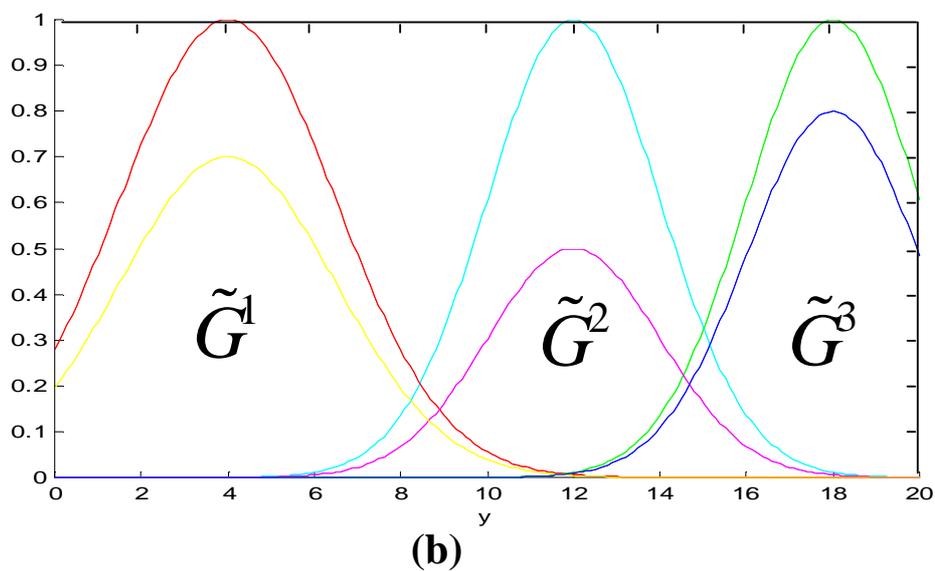
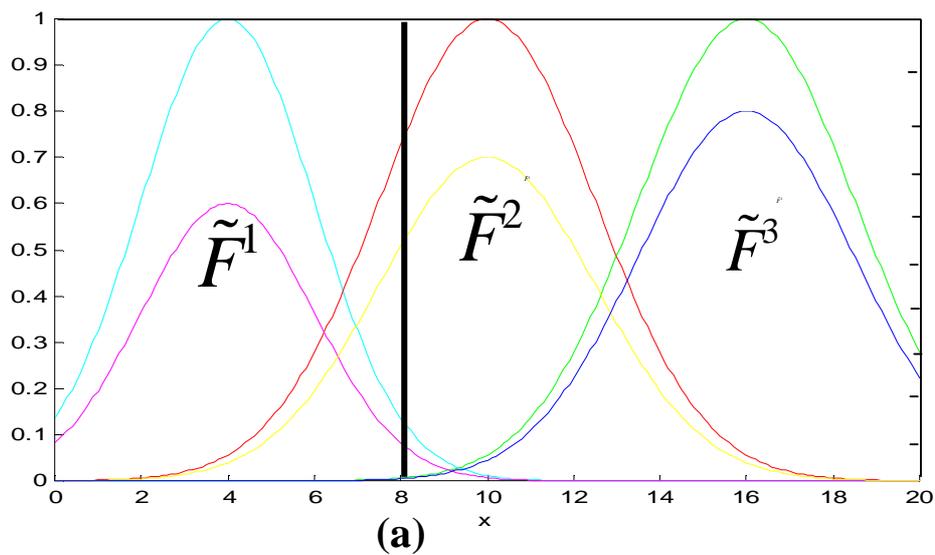


Fig. III. 8 : (a) : Ensembles flous antécédents (b) : ensembles flous Conséquences de l'exemple 6.

III.5. CONCLUSION:

Dans ce chapitre nous avons exposé la théorie des sous ensemble floue type 2 et rappeler les définitions les plus importantes et de base concernant le concept du système flous type 2.

Une comparaison entre les deux types de systèmes flous 1 et 2 à été efficacement mise en évidence.

Chapitre IV

Application et résultats de simulations

IV.1. Introduction:

La commande de mouvement des robots mobiles a trouvé une attention considérable dans les dernières années. La plupart de ces travaux se sont concentrés sur la direction ou problème de la génération de la trajectoire au niveau de la cinématique ,c'est-à-dire considérer le système des vitesses comme grandeurs de commande et ignorer la dynamique du système mécanique. Il existe peu de travaux qui ont été publiés sur la commande avec la présence d'incertitudes dans le modèle dynamique.

Il y a trois taches possibles où le véhicule peut faire la stabilisation point à point, suivi d'un chemin et suivi d'une trajectoire.

La stabilisation point à point nécessite que le véhicule se déplace d'un point A à un point B sans conditions sur son déplacement entre ces deux points. Le suivi du chemin nécessite que le véhicule doive se déplacer le long d'un chemin géométrique. Le suivi de la trajectoire est le même que le suivi du chemin sauf que la voiture suit le chemin avec une vitesse donnée.

Dans ce chapitre, on va appliquer d'une part la commande floue type-1 et d'autre part la commande floue type-2, pour commander un robot non holonome à roues (véhicule) afin que ce dernier puisse suivre un chemin de référence premièrement rectiligne puis circulaire.

Nous disposons pour cela d'un chemin de référence discrétisé, sous une forme d'une séquence de configuration d'état du véhicule c'est-à-dire à chaque instant, on a :

- La configuration de référence: $q_{ref}(t) = (x_{ref}(t), y_{ref}(t), \theta_{ref}(t))$
- L'état de référence : vitesse $\Omega_{ref}(t)$

En complément de ce chemin, nous disposons d'informations courantes, correspondant aux caractéristiques réelles du véhicule, comme :

- La configuration Courante : $q_{court}(t) = (x_{court}(t), y_{court}(t), \theta_{court}(t))$ délivrée par un module de localisation qui peut un capteur ou un odomètre.
- Une vitesse courante $\Omega_{court}(t)$

IV.2. Notion de sous but:

Notre suivi de chemin va consister à essayer de relier à chaque instant t le sous but G_{ref} caractérisé par la configuration $q_{ref}(t)$ et $\Omega_{ref}(t)$ de référence du véhicule au même instant. Le sous but évalue continuellement le suivi de chemin en minimisant un critère associé à un comportement : C'est la minimisation de l'écart entre la configuration de référence et la configuration courante notée:

$$\|q_{ref} - q_{court}\|.$$

Cette minimisation est assurée en deux étapes par:

1- La convergence en position:

Pour cela nous considérons la position de référence (x_{ref}, y_{ref}) en coordonnées polaires dans le repère de véhicule de configuration $q_{court}(t)$.

La composante radiale représente l'erreur en distance et la composante angulaire représente la direction à suivre, l'angle du cap (θ), dans le but de minimiser cette erreur.

2- La convergence en orientation:

En complément de la convergence en position, nous devons considérer également celle en orientation; celle-ci est assurée par actions sur la vitesse de braquage.

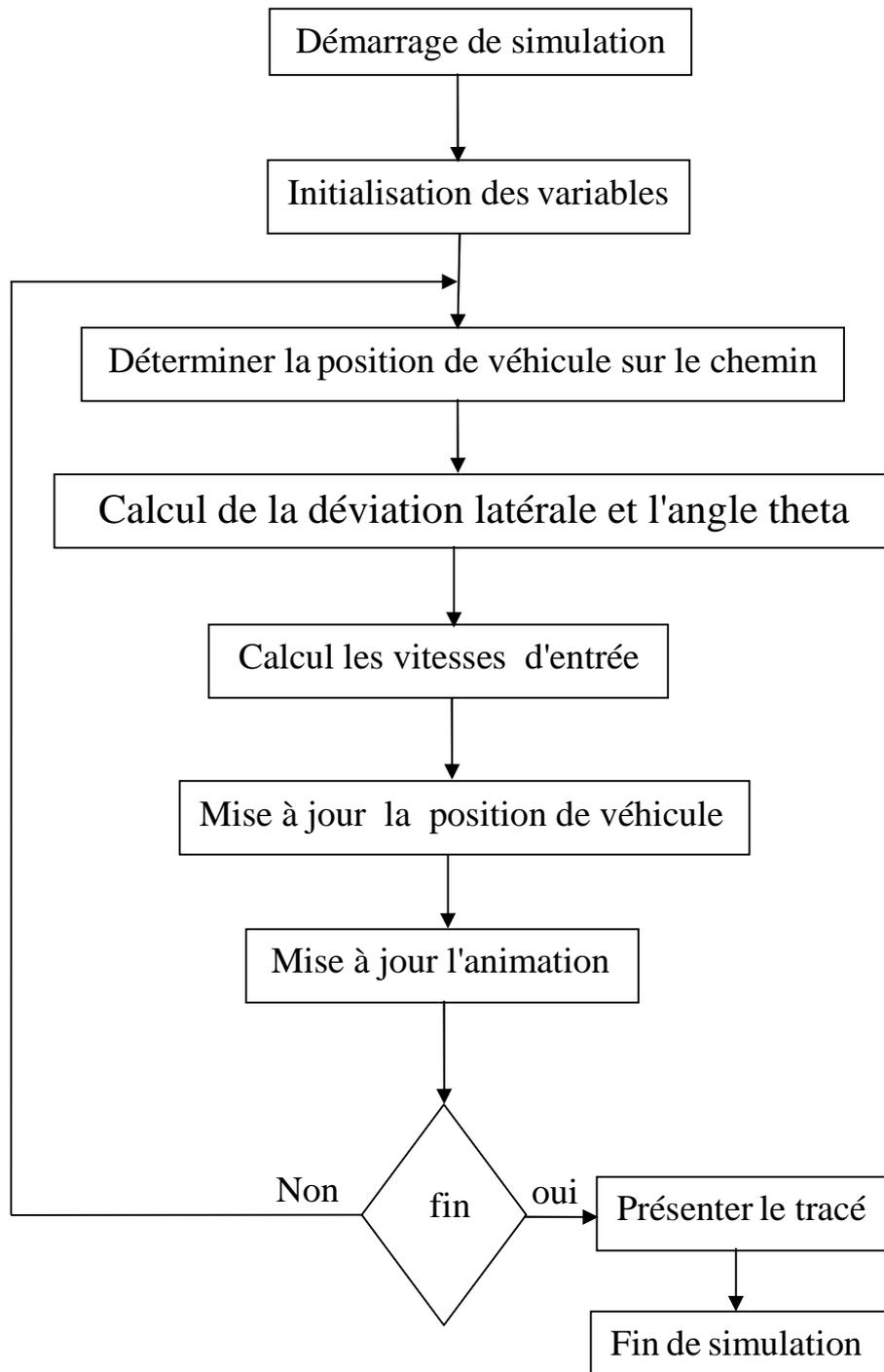
IV.3. Simulation

L'étape d'initialisation implique la création du véhicule et le chemin pour l'animation et l'emplacement de véhicule sur le chemin.

La position sur le chemin est déterminée et les valeurs désirées sont calculés : vitesse nécessaire à l'entrée, pour permettre au véhicule de suivre le chemin.

Ces entrées sont utilisées dans le modèle cinématique pour mettre le véhicule en position et mise à jour .Ces étapes sont répétées jusqu'à la fin de la simulation.

L'organigramme de simulation est représenté dans la figure (IV.1) ci-dessous :

**Fig.IV.1** : Organigramme de simulation

IV.3.1. Le chemin:

Le chemin contient un segment droit suivi d'une courbe, suivie par un autre segment droit.

Le chemin est défini par les coordonnées (x,y) , les arcs et le rayon de courbure sont définis comme suit:

$$y = \begin{cases} -x + r \left(1 - \frac{2}{\sqrt{2}}\right) & x < \frac{-r}{\sqrt{2}} \\ -\sqrt{r^2 + x^2} + r & \frac{-r}{\sqrt{2}} \leq x \leq \frac{r}{\sqrt{2}} \\ x + r \left(1 - \frac{2}{\sqrt{2}}\right) & x > \frac{r}{\sqrt{2}} \end{cases} \dots\dots\dots(IV.1)$$

Ou :

r est le rayon de la section courbée de chemin et en utilisant $r = 1$.

La figure IV.2 illustre le chemin engendré en utilisant une programmation sous Matlab

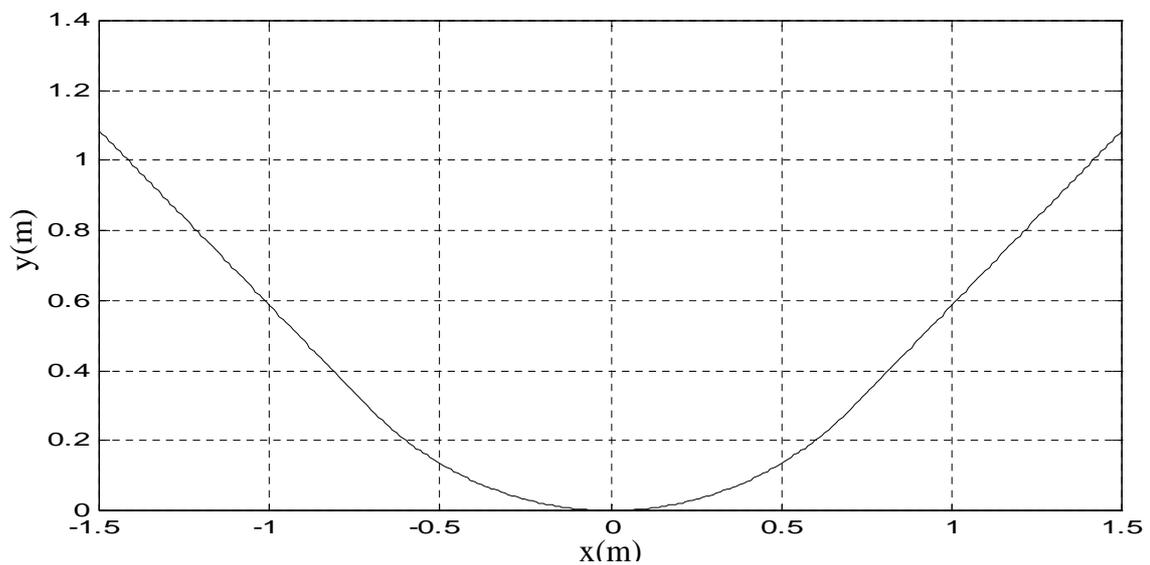


Fig .IV.2: Chemin engendré sous logiciel Matlab

IV.3.2. Calcul de l'erreur actuelle

La position de véhicule est connue par les coordonnées (x_0, y_0) et il en est de même pour l'orientation θ et l'angle représentant la directrice ϕ .

La position des sondes (capteur) est indiqué par:

$$x_1 = x_0 + l \cos \theta \dots \dots \dots (IV .2)$$

$$y_1 = y_0 + l \sin \theta \dots \dots \dots (IV .3)$$

Deux points le long de l'axe du centre de véhicule sont connus (x_0, y_0) et (x_1, y_1) la

penste de la ligne (1) dans la figure (IV.3) est: $\frac{y_1 - y_0}{x_1 - x_0}$.

La ligne (1) et la ligne (2) étant perpendiculaires, par conséquent la penste de la ligne (2) est donnée par l'expression :

$$\frac{-(x_1 - x_0)}{y_1 - y_0}$$

La penste de la ligne (2) étant maintenant un point connu, son équation est de la forme :

$$y = -m (x - x_1) + y_1 \dots \dots \dots (IV.4)$$

Ou :

$$m = \frac{(x_1 - x_0)}{y_1 - y_0}$$

Il faudra déterminer le point (x_2, y_2) par l'intersection de la ligne (2) et le chemin.

Notons que le coté droit de la courbe (IV.1) égale au coté droit (IV. 2) ce qui donne :

pour $x_1 < \frac{-r}{\sqrt{2}}$:

$$x_2 = m x_1 + y_1 + r \left(1 - \frac{1}{\sqrt{2}}\right) \dots \dots \dots (IV.5)$$

$$y_2 = -x_2 + r \left(1 - \frac{1}{\sqrt{2}}\right) \dots \dots \dots (IV.6)$$

pour $\frac{-r}{\sqrt{2}} \leq x \leq \frac{r}{\sqrt{2}}$:

$$x_2 = -b^2 \pm \sqrt{b^2 - 4ac} \dots \dots \dots (IV.7)$$

$$y_2 = -\sqrt{r^2 - x_2^2} + r \dots \dots \dots (IV.8)$$

Le signe du carré est le même que m :

$$a = m^2 + 1$$

$$b = -2m^2 x_1 - 2m x_1 + 2m r$$

$$c = 2m x_1 y_1 - 2m x_1 r + y_1^2 - 2y_1 r$$

pour $x_1 > \frac{r}{\sqrt{2}}$:

$$x_2 = \frac{m x_1 + y_1 - r \left(1 - \frac{1}{\sqrt{2}}\right)}{m - 1} \dots \dots \dots (IV.9)$$

$$y_2 = x_2 + r \left(1 - \frac{1}{\sqrt{2}}\right) \dots \dots \dots (IV.10)$$

Connaissant les points de coordonnées : (x_1, y_1) ; (x_2, y_2) , l'erreur est par conséquent la différence :

$$e_f = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \dots\dots\dots (IV.11)$$

La racine carrée, négative ou positive utilisée dépend du centre du véhicule à droite ou à gauche du chemin.

La convention utilisée est si le chemin est à droite on prend la valeur positive.

L'erreur à l'arrière du véhicule, peut être trouvée en utilisant la méthode précédente sauf uniquement la ligne 2 passes à travers (x_0, y_0)

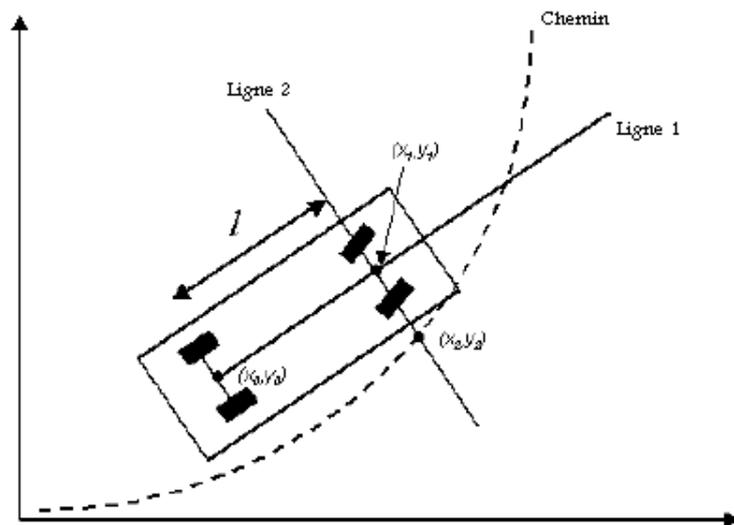


Fig.IV. 3 : Erreur de chemin suivi le véhicule

IV.3.3. Calcul de l'angle θ_p

Il faut que le contrôleur connaisse l'angle θ_p , angle entre le véhicule et le chemin.

La valeur peut être calculée en utilisant l'erreur de déplacement avant et arrière du véhicule comme déterminées précédemment e_f et e_b et la distance entre eux l .

On a respectivement:

e_f : l'erreur de déplacement avant

e_b : l'erreur de déplacement arrière

l : la distance entre e_f et e_b

On suppose que le chemin est directement sous le véhicule et tout droit : La valeur de θ_p est donnée par la formule (IV.12)

$$\theta_p = \tan^{-1} \left[\frac{e_f - e_b}{l} \right] \dots\dots\dots (IV.12)$$

IV.3.4. Modèle de véhicule:

Le mouvement de véhicule est déterminé sur une période d'échantillonnage T .

Le modèle cinématique est implémentée dans le logiciel Simulink illustré par la figure (IV.4).

Le modèle utilise la position courante du véhicule comme condition d'initialisation et doit être intégré pour déterminer la nouvelle position.

On suppose que les entrées sont des constantes sur le temps d'échantillonnage.

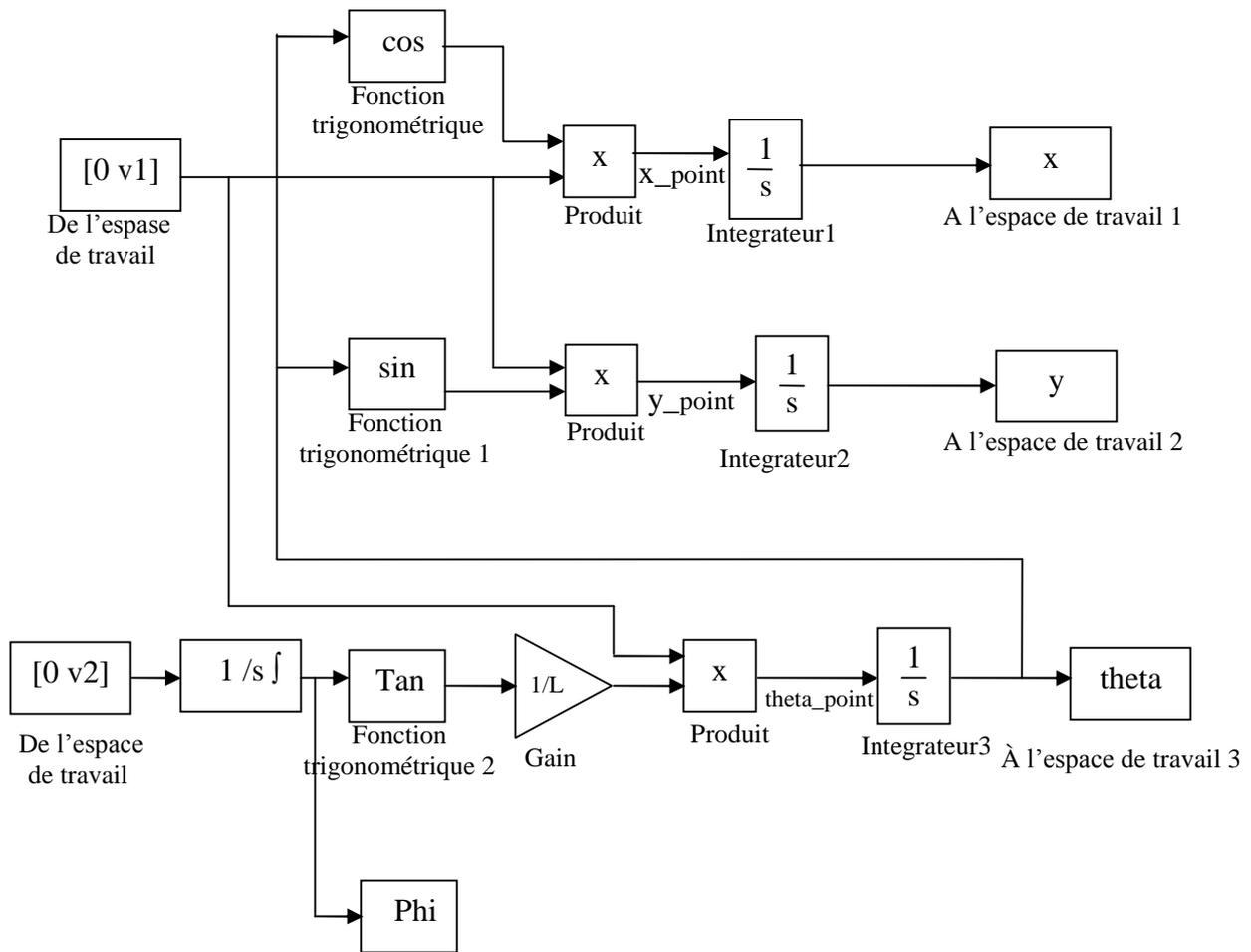


Fig. IV. 4 : Modèle cinématique de véhicule (Simulink)

Le modèle cinématique est donné par :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \frac{\tan \phi}{l} \\ 0 \end{bmatrix} \mathbf{v}_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{v}_2 \dots\dots\dots(IV.13)$$

\mathcal{V}_1 et \mathcal{V}_2 représentent respectivement :

\mathcal{V}_1 : Vitesse linéaire

\mathcal{V}_2 : Vitesse angulaire

Avant développer le contrôleur le système est transformé dans la forme enchaîné (2,n) (chaîne- singulier)

Dans le système à 2 entrées u_1 et u_2 , il peut être considéré à une seule entrée si L'évolution de u_1 est connue.

Pour le modèle de véhicule à quatre (4 états) la représentation de la forme enchaîné (2,4) devient :

$$\begin{aligned}\dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \dots\dots\dots(IV.14) \\ \dot{x}_4 &= x_3 u_1\end{aligned}$$

Pour stabiliser x_2, x_3, x_4 du système représenté par l'équation (IV .2) vers zéro, les entrées sont données par :

On a :

$$\chi = (\chi_1, \chi_2, \chi_3, \chi_4) = (x_1, x_4, x_3, x_2)$$

La forme enchaînée est donc présentée par l'équation (IV.15):

$$\begin{aligned}\dot{\chi}_1 &= u_1 \\ \dot{\chi}_2 &= \chi_3 u_1 \\ \dot{\chi}_3 &= \chi_4 u_1 \dots\dots\dots(IV.15) \\ \dot{\chi}_4 &= u_2\end{aligned}$$

Le système représenté par l'équation (IV.3) est contrôlable si $u_1(t)$ est continue, à valeurs limitées et fonction strictement positive ou négative.

En supposant que $u_1(t)$ est fixée d'une manière arbitraire, on a :

$$u_2(t) = -k_1 |u_1(t)| \chi_2 - k_2 u_1(t) \chi_3 - k_3 |u_1(t)| \chi_4 \dots \dots \dots (IV.16)$$

$$k_1 = \lambda^3, k_2 = 3\lambda^2, k_3 = 3\lambda, \text{ avec } \lambda = 8$$

Où λ et k représentent respectivement

$u_2(t)$: la loi de la commande

k, λ : les gains

IV.4. Les contrôleurs flous :

Notre travail s'articule essentiellement sur le développement d'un contrôleur d'exécution de mouvement pour un robot mobile de type voiture.

Notre contrôleur réactif se porte à la correspondance étroite entre les capteurs et les actionneurs. C.-à-d. : l'utilisation direct d'information capteur dans la génération des consignes, son comportement de base est le suivi d'un chemin.

La programmation de contrôleur repose sur la logique floue par l'utilisation des fonctions d'appartenances classiques (type1) pour le contrôleur floue type 1 et les fonctions d'appartenances de type 2 pour le contrôleur floue type2

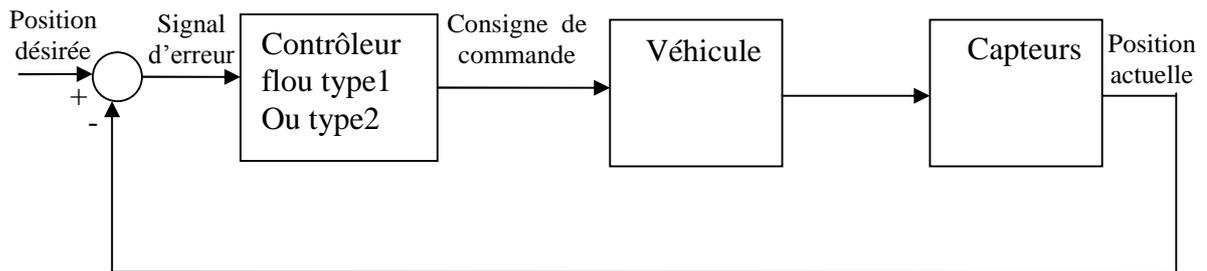


Fig .IV .5 : Bloc diagramme pour le contrôleur flou

Dans cette partie nous donnons un extrait de la base de règles utilisées pour notre contrôleur flou. Cette base a été décomposée en un certain nombre de comportements en précisant pour chacun les entrées et les sorties considérées du contrôleur.

IV.4.1 Simulation utilisant le contrôleur flou type1

Dans ce type de contrôleur on a choisi les entrées du contrôleur flou dont le type de vecteur est de la forme type :

$$[\Delta e \quad \theta_p \quad \Delta\theta_p]$$

Ou :

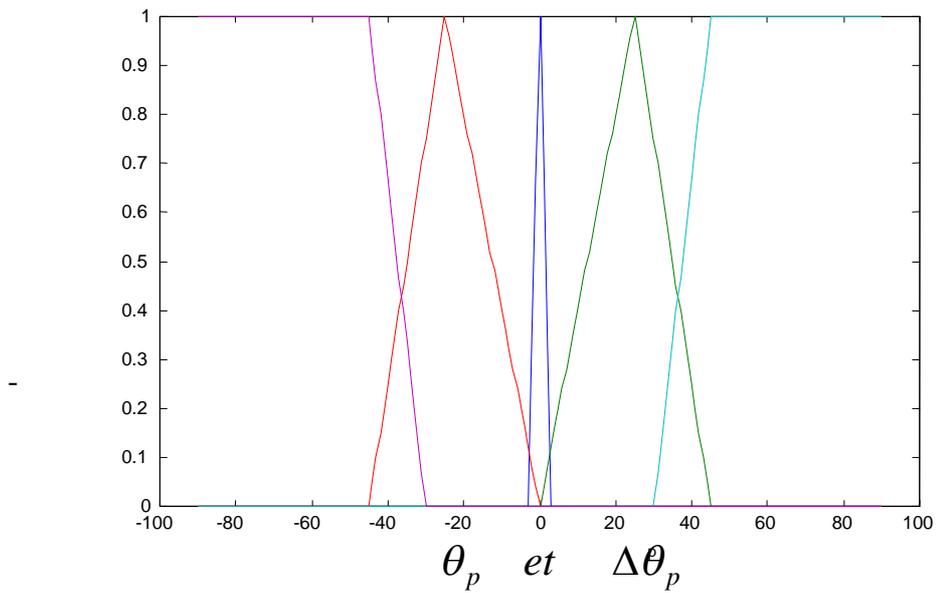
- θ_p : angle entre la véhicule et le chemin
- Δe : Erreur distance

Le vecteur de sortie :

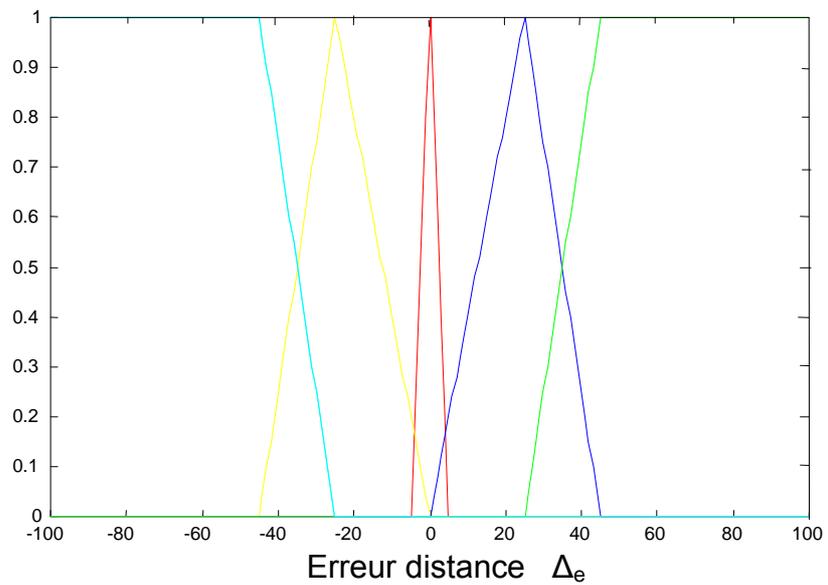
$$[v1 \quad v2]$$

- v1: Vitesse linéaire du robot
- v2 : vitesse angulaire du robot

Les fonctions d'appartenances sont choisies comme l'illustre la figure (IV.6) ci-dessous:



(a)



(b)

Fig. IV.6 : Fonctions d'appartenances du contrôleur flou type1(a) : θ_p angle entre le véhicule et le chemin(b) : Δ_e Erreur distance

La figure (IV.6) illustre les différentes fonctions d'appartenances de notre contrôleur flou ou chaque entrée est représentée par cinq (05) fonctions d'appartenances, ceci nous donnera 25 règles dans la base de règle.

L'extrait de la base de règles se déroule de la manière suivante :

Si angle_cup θ_p est NG Alors θ_p _point est NG

Si angle_cup θ_p est PG Alors θ_p _point est PG

Si erreur_distance est NG Alors vitesse est NG

Si erreur_distance est PG Alors vitesse est PG

Si delta_theta est NG Alors θ_p _point est NG

Si delta_theta est PG Alors θ_p _point est PG

▪ **Remarque:**

Dans les règles, les suffixes de labels de sous-ensembles flous PG, PP, ZR, NP, NG correspondent respectivement à Positif Grand, Positif Petit, Zéro, Négatif Petit, Négatif Grand.

IV.4.2. Résultats de Simulation Contrôleur flou type 1 :

D'abord la loi de commande va être utilisée pour la plate forme du robot mobile afin que ce dernier puisse suivre un chemin.

Prenons les paramètres du robot :

$L = 10 * 2.54 / 100$; distance entre les roues arrières et les roues avants (m)

$W = 6.5 * 2.54 / 100$; l'espace entre les roues (m)

$H = 2.5 * 2.54 / 100 / 2$; hauteur (m)

$D = 2.5 * 2.54/100$; diamètre des roues (m)

▪ **Conditions initiales**

$x_0 = -x_{\max} * \text{rayon}$;

$y_0 = -x_0 + \text{rayon} * (1 - 2/\text{racine}(2))$;

$\theta_0 = -39 * \pi/180$; (rad)

$\phi_0 = 0 * \pi/180$; (limité entre -45 et 45 degré) (rad)

$u_1 = 1.5$ (m/s)

$u_2 = 0$ (rad/s)

Les résultats de la simulation du contrôleur flou type 1 sont mis en évidence dans les figures suivantes :

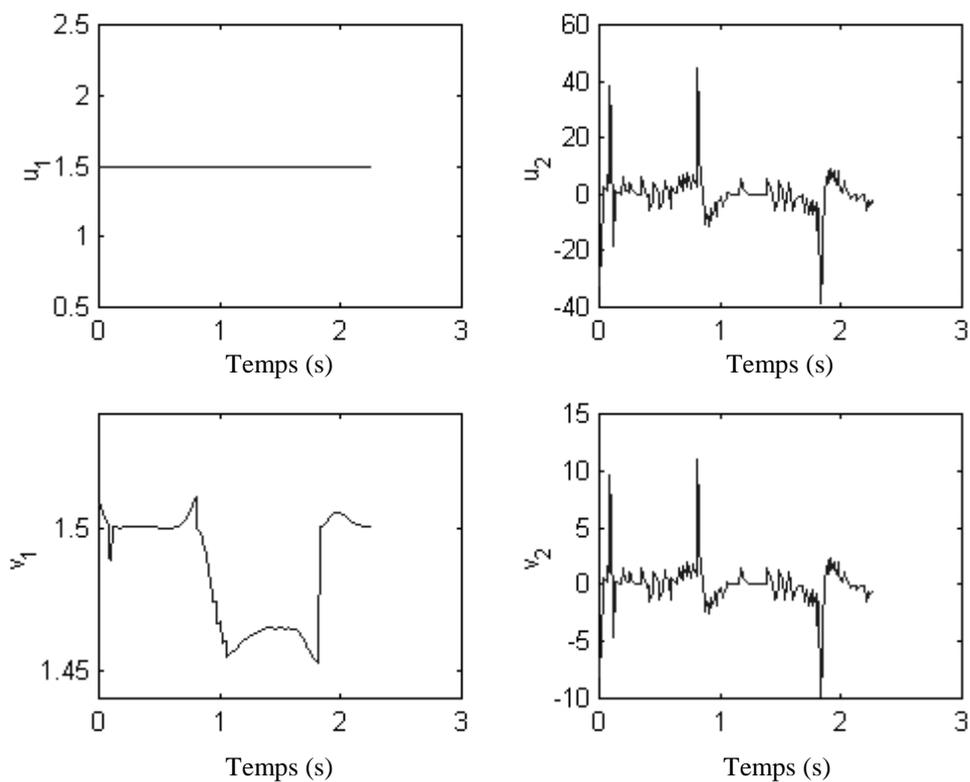


Fig. IV. 7 : Commandes résultantes de contrôleur flou type 1

Ou u_1, u_2 , commandes (entrées du système) v_1 et v_2 sont respectivement vitesse linéaire et vitesse angulaire de roue

D'après la figure(IV.7) on remarque :

- Pendant le temps de simulation u_1 est constant, ayant été pris égal à 1.5.
- u_1 constant et v_1 varie entre (1.5 m/s et 1.45 m/s) ; Cette vitesse est fixée à 1.5 m/s lorsque le chemin est tout droit puis diminue à la valeur de 1.45 m/s et enfin augmente à 1.5 m/s lorsque il y a une courbure ou un virage.
- La vitesse angulaire v_2 varie au voisinage du zéro (la valeur de θ étant petite) puis augmente à la valeur maximale et absolue absolue (+10 et -10) lorsqu'on est en présence d'un virage.
- Les deux pointes produites dans u_1 à l'initialisation sont dus à l'estimation de courbure qui est incorrect.
- La première pointe produit lors de transition de zéro à $1/R$.
- La deuxième pointe produit lors de transition de $1/R$ à zéro.

Les figures (IV.8) et (IV.9) ci-dessous représentent successivement l'angle θ_p ,

l'angle directeur ϕ résultant et les états résultants du véhicule :

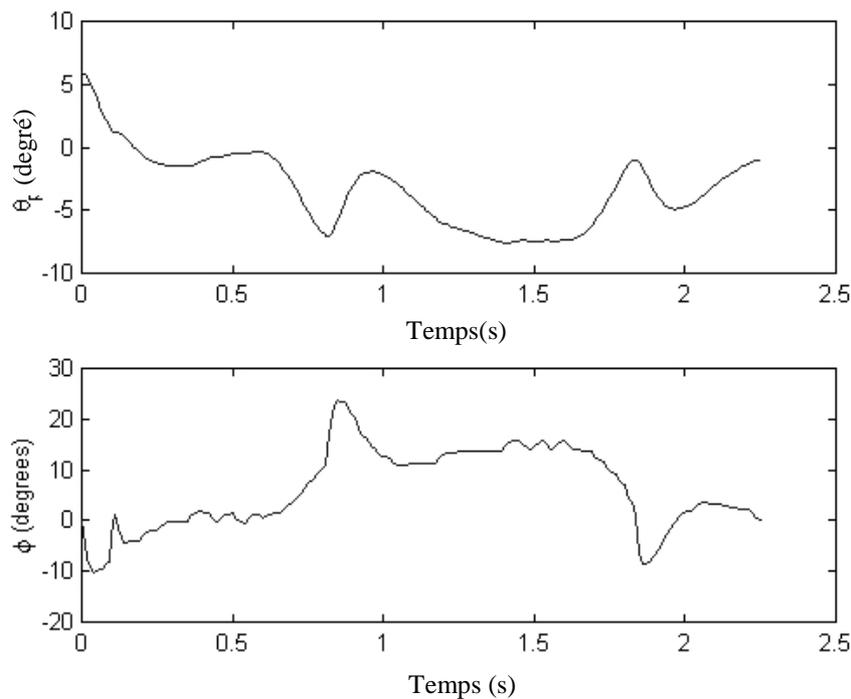


Fig .IV. 8: Angle θ_p et l'angle directrice ϕ résultant de contrôleur flou type 1

D'après la figure (IV.8) on remarque qu'il ya des petit angles iorsque le véhicule est au chemin droite ceci due a la correction des angles pour avoir $\phi=0^\circ$ de la part le contrôleur floue type1.à l'instant $t =0.9s$ $\phi = 24^\circ$ et à $t =1.9s$ $\phi = -10^\circ$ donc ϕ augmente ou diminue aux valeurs maximums lorsque le véhicule rencontre des virages.

On remarque aussi que θ_p sui ϕ à l'inverse quand ϕ positif est θ_p négative et quand ϕ négative θ_p positif

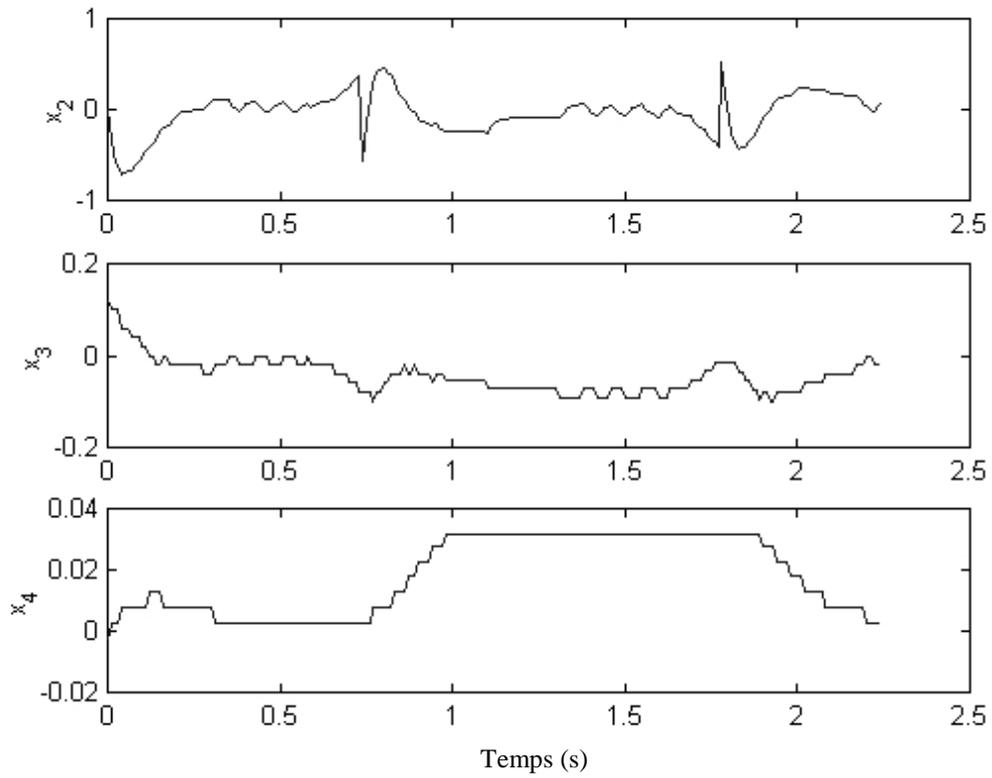


Fig.IV .9 : Etats résultants de véhicule de contrôleur flou type1

Les courbes des états de véhicule en la même forme des courbes θ_p et ϕ parce que ces dernier en fonction de u_1 et u_2

IV.4.3 Simulation utilisant un contrôleur flou type2

Pour faire une étude comparative entre le contrôleur flou type 1 et le contrôleur flou type 2, nous allons prendre le même système avec les mêmes entrées et les mêmes conditions initiales.

La novation que sera apportée à ce contrôleur réside dans le fait que la fonction d'appartenance de type 2 est choisie comme le montre la figure (IV.10) ci-dessous:

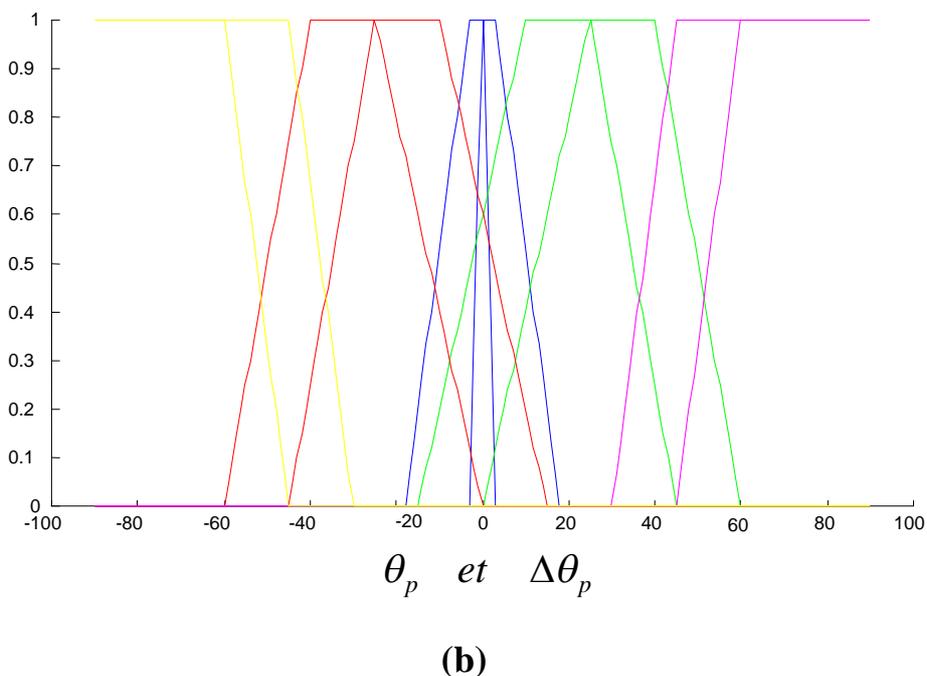
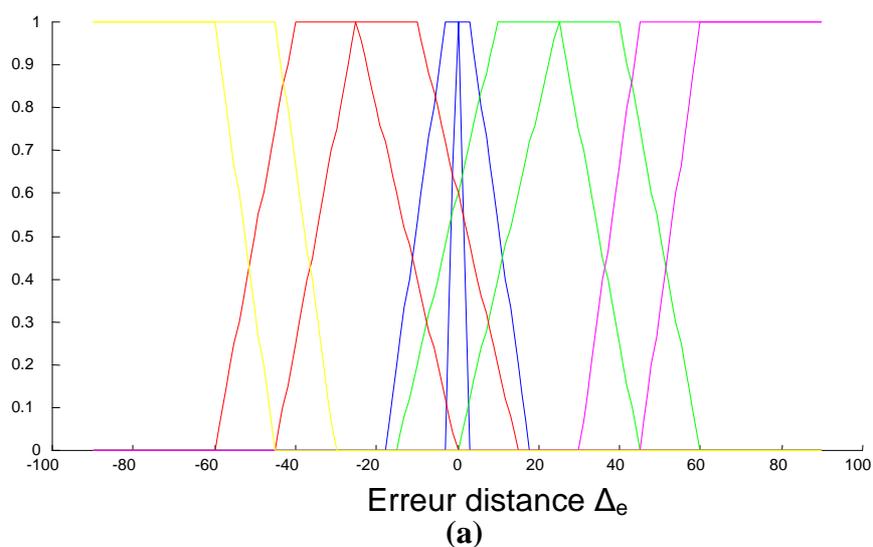


Fig .IV. 10: Fonctions d'appartenance du contrôleur flou type 2

(a) : Δe Erreur distance. **(b) :** θ_p angle entre le véhicule et le chemin

La figure IV.10 met en évidence les différentes fonctions d'appartenances du contrôleur flou type 2 ou chaque entrée est représentée par cinq(05) fonctions d'appartenances (fonction d'appartenance supérieure et la fonction d'appartenance inférieure $\underline{\mu}_{\tilde{F}_i}(x_i)$ et $\bar{\mu}_{\tilde{F}_i}(x_i)$).

- On a 3 entrées x_1, x_2, x_3 et 2 sorties Y_1, Y_2
- Notre choix est fixé pour la fuzzification singleton ; Ceci nous permet d'affirmer que l'ensemble type 2 d'entrée \tilde{A}_c contient un seul élément.
- La $i^{ème}$ règle de FLC type 2 peut être écrit comme suit :

*SI x_1 est \tilde{F}_1^i et x_2 est \tilde{F}_2^i et x_3 est \tilde{F}_3^i ALORS y_1 est \tilde{G}_1^i et y_2 est \tilde{G}_2^i
 $i=1, \dots, M$; M étant le nombre de règles.*

- Dans le moteur d'inférence les entrées multiples sont reliées entre eux et utilisent l'opération « Meet ».
- les degrés d'appartenance des ensembles d'entrées sont reliés avec ceux des ensembles de sortie en utilisant l'opération « sup-star ».
- les multiples règles sont reliées entre eux utilisant l'opération « Join ».

Le résultat d'entrée et les opérations d'antécédent qui sont contenus dans l'ensemble activé

$$\bigcap_{\alpha=1}^p \mu_{\tilde{F}_\alpha}(x'_\alpha) \equiv F^i(x'),$$

est un ensemble type-1 intervalle :

$$F^i(x') = [\underline{f}^i(x'), \bar{f}^i(x')] \equiv [\underline{f}^i, \bar{f}^i] \dots \dots \dots (IV.17)$$

(*) : Étant l'opération produit :

Ou $f^i(x')$ et $\bar{f}^i(x')$ peuvent être écrit comme suit :

$$f^i(x') = \underline{\mu}_{\bar{F}_1^i}(x'_1) * \dots * \underline{\mu}_{\bar{F}_3^i}(x'_3) \dots \dots \dots (IV.18)$$

$$\bar{f}^i(x') = \bar{\mu}_{\bar{F}_1^i}(x'_1) * \dots * \bar{\mu}_{\bar{F}_3^i}(x'_3) \dots \dots \dots (IV.19)$$

L'ensemble de type réduit utilisant l'ensemble de type réduction, s'exprime par la relation suivante:

$$Y_{\cos}(x)_k = [y_{lk}, y_{rk}] = \int_{y_k^i \in [y_{lk}^i, y_{rk}^i]} \dots \int_{y_k^M \in [y_{lk}^M, y_{rk}^M]} \dots \int_{f^i \in [f^i, \bar{f}^i]} \dots \int_{f^M \in [f^M, \bar{f}^M]} 1 \int \frac{\sum_{i=1}^M f^i y_k^i}{\sum_{i=1}^M f^i} \dots \dots \dots (IV.20)$$

Ou

- $y_{\cos}(x)_k$: pour la k^{ème} sortie est un ensemble d'intervalle
- Le calcul de l'ensemble type réduit est divisé en deux étapes :
- La première étape consiste au calcul des centres des ensembles type_2, des conséquences de chaque règle avant de commencer l'opération du FLC du robot.

$$y_k^t = [y_{lk}^t, y_{rk}^t] = \int_{\theta_1 \in j_{y1}} \dots \int_{\theta_z \in j_{yz}} \dots 1 / \frac{\sum_{z=1}^z y_z \theta_z}{\sum_{z=1}^z \theta_z} \dots \dots \dots (IV.21)$$

Pour calculer les centroïdes de chaque sortie de l'ensemble nous emploierons la procédure itérative n° :1 voir chapitre 3.

On doit maximiser et minimiser y en fonction de $(\theta_1 \dots \dots \theta_z)$ pour calculer les points finaux puis passer à l'optimisation :

$$y(\theta_1 \dots \dots \theta_z) = \frac{\sum_{z=1}^z y_z \theta_z}{\sum_{z=1}^z \theta_z} \dots \dots \dots (IV.22)100$$

- La seconde étape produit chaque cycle opératoire pour calculer l'ensemble type réduit diffuzifié afin de reproduire les sorties réelles pour le déclencheur.

A chaque sortie k afin de calculer $y_{\cos}(x)_k$ dans (IV.22) nous avons besoin du calcul de ces deux points finaux y_{lk} et y_{rk} .

$$y_{lk} = \frac{\sum_{i=1}^M f_l^i y_{lk}^i}{\sum_{i=1}^M f_l^i} \dots\dots\dots(IV.23)$$

$$y_{rk} = \frac{\sum_{i=1}^M f_r^i y_{rk}^i}{\sum_{i=1}^M f_r^i} \dots\dots\dots(IV.24)$$

IV.4.4. Résultats de Simulation Contrôleur flou type2 :

Les résultats de simulation obtenus pour le contrôleur flou type 2 sont illustrés par les graphes de la figure (IV.10) ci-dessous :

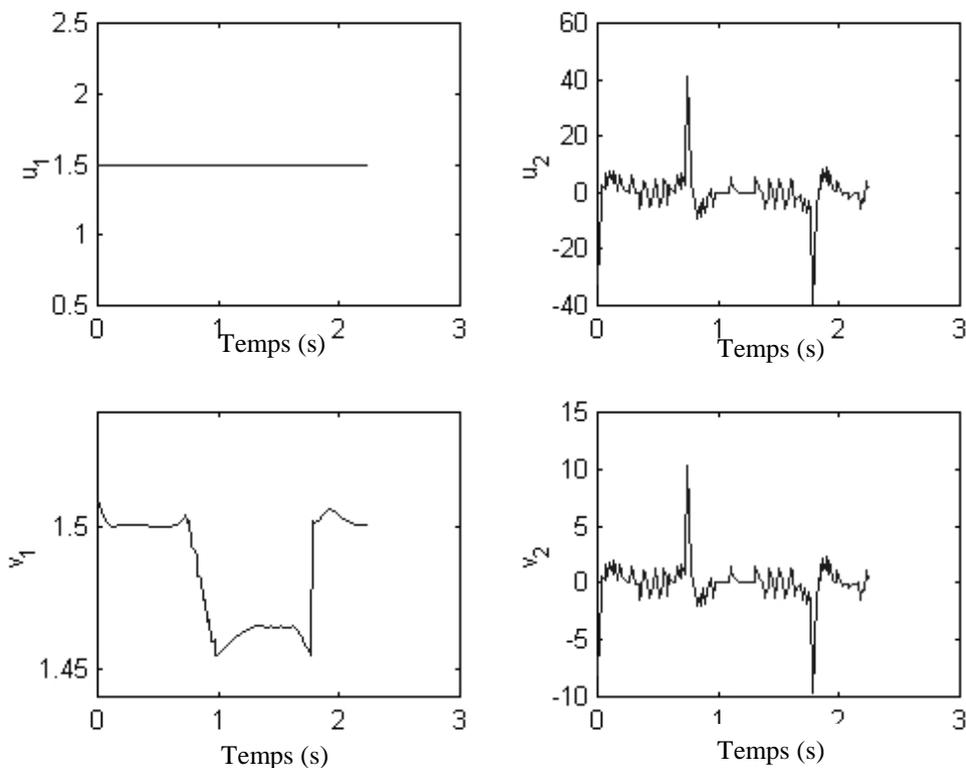


Fig.IV. 11: Commandes resultantes de controleur flou type 2

Ou u_1, u_2 , commandes (entrées du système) v_1 et v_2 sont respectivement vitesse linéaire et vitesse angulaire de roue

- U_1 est constant
- U_1 transforme en v_1 et par conséquent la voiture ralentit dans le virage.
- Il y'a trois pic dans la commande u_2 , résultants des pointes qui se produisent en x_2 .
- Ces pic sont produites exactement lorsque le chemin change de courbure.
- Dans cette point la pente de courbure est infini

La figure IV.11 représente l'angle θ_p et l'angle directeur ϕ résultant de contrôleur flou type 2

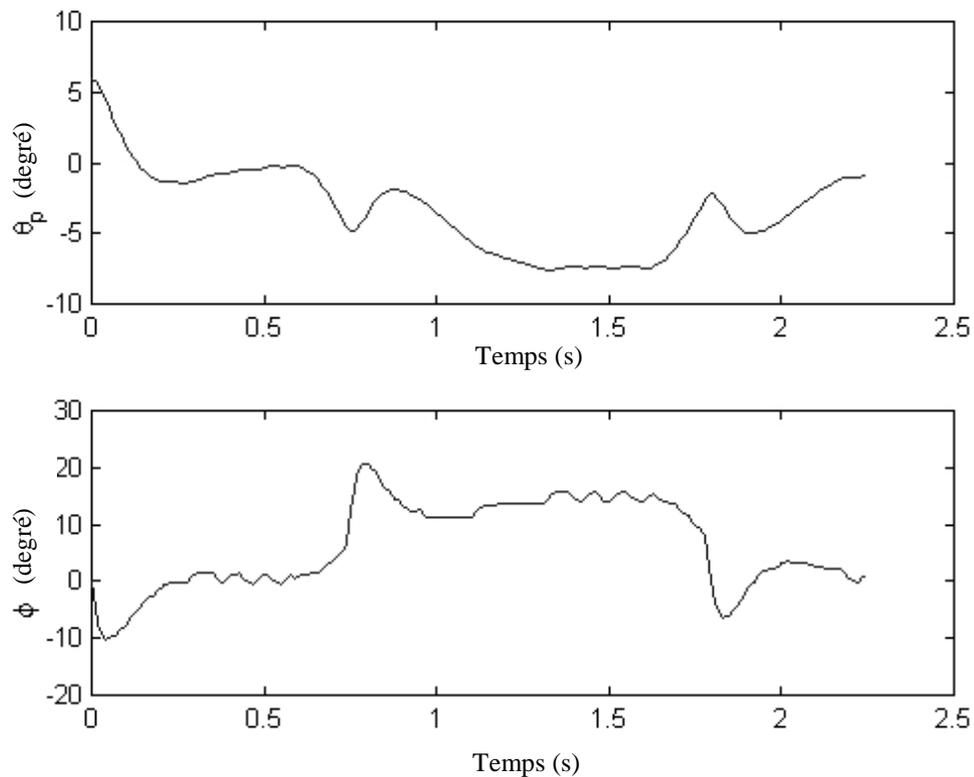


Fig .IV.12 : Angle θ_p et l'angle directrice ϕ résultant de contrôleur flou type 2

Les graphes de la figure IV.12 mettent en évidence les états résultants du véhicule :

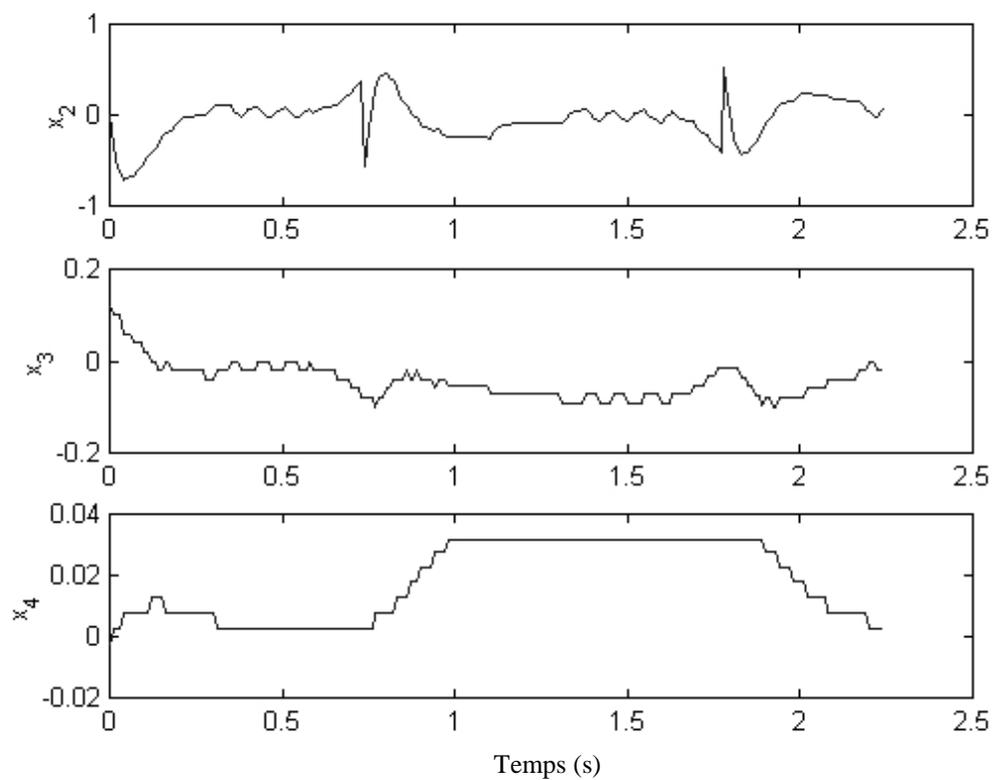


Fig .IV.13: Etats résultants du véhicule de contrôleur flou type2

IV.5. Interprétation des résultats

Les résultats obtenus et mis en évidence par les principales courbes des deux systèmes contrôleur flou type 1 et type 2 successivement nous permettent de conclure :

Lorsque U_1 est constant et V_1 variable (voir figures), on prend en considération les états de véhicule et aussi les courbures du chemin ce qui a pour conséquence (diminution de la vitesse) dans la section de courbure correspondant au virage.

On peut remarquer aussi qu'il y a des pointes dans l'entrée U_2 cela est due aux pointes présentes dans x_2 ; ces pointes se produisent exactement lorsque le chemin change de courbure.

Les courbes de simulation obtenues par l'utilisation du contrôleur type 2 sont meilleures que celles des courbes obtenues par simulation pour le contrôleur type 1. Cette affirmation est justifiée par les points suivants

- La commande représentée par le contrôleur type 2 est plus lisse,
- Les vitesses angulaires et linéaires sont plus précises en ce qui concerne le deuxième contrôleur,
- Les états du véhicule X_2 , X_3 et X_4 sont plus précis et plus lisses

IV.6. Conclusion

Pour la condition initiale $u_1=1.5\text{m/s}$, les résultats de simulation pour les deux systèmes type-1 et type-2 sont représentés dans les figures Fig.IV.7 et Fig IV. 10, respectivement. Nous remarquons dans ces figures que vitesse lineaire et vitesse angulaire obtenue par le controleur flou type-1 est meilleure que celle obtenue par le régulateur flou type-2. Par contre, l'angle ϕ du contrôleur flou type-2 est plus précise que celle obtenue avec le contrôleur flou type-1. Ces remarques sont confirmées par les figures IV.8 et IV.12 où nous illustrons les étas de véhicule pour les deux systèmes. L'avantage du contrôleur flou type-1 est confirmé par la figure V.8, tandis

que l'avantage du contrôleur flou type-2 est confirmé par la figure V.12. Les mêmes remarques sont valables pour les réponses en vitesse pour les deux systèmes. Concernant les actions de commande nous remarquons que la commande fournie par le contrôleur flou type-2 est plus lisse, ce qui représente un avantage pur ce dernier.

Pour plus de comparaisons, et pour en tirer plus de conclusions, nous allons maintenant considérer les réponses correspondantes à la condition initiale suivante $x(0)=0m$. En se reportant aux figures IV.7, IV.8, IV.11, et IV.12 nous pouvons constater la supériorité du contrôleur flou type-2, car selon ces figures, on remarque que :

Pour le contrôleur type-1 :

$$u1 = 1.5(m/s)$$

$$u2 = 0 (rad/s)$$

$$V1 = 1.5 \text{ diminue à } 1.49$$

$$V2 = -6$$

Pour le contrôleur type-2 :

$$u1 = 1.5(m/s)$$

$$u2 = 0 (rad/s)$$

$$V1 = 1.5 \text{ m/s}$$

$$V2 = -6m/s$$

Donc, les efforts des commandes générés par le contrôleur flou type-2 sont inférieurs à ceux générés par le régulateur flou type-1, ce qui représente un avantage pour le contrôleur flou type-2.

On conclut que la commande par le contrôleur flou type 2 est nettement meilleure en précision concernant la vitesse et les états du véhicule.

Conclusion Générale

Dans ce mémoire notre objectif primordial réside dans l'application de la logique floue dans la robotique mobile en se basant sur nouvelle logique floue appelé logique floue type 2.

Nous avons commencer notre travail de recherche par la présentation des concepts de base et la théorie de la logique floue type 1 : Théorie des ensembles flous, variables linguistiques, fonctions d'appartenance, caractéristiques d'un ensemble flou, opérations sur les ensembles flous, logique floue, le principe d'extension, système flous et fonctions de base floues.

On a ensuite présenté la théorie de la logique floue type-2. Les points traités sont : Principales définitions terminologie utilisée dans la logique floue type-2 ,les opérations utilisées dans la logique floue type-2, systèmes flous type-2, , la réduction de type et la défuzzification.

Nous avons axé notre travail à la comparaison entre les résultats de simulation obtenus pour les deux approches, logique floue classique et logique floue type 2

Les résultats de simulation ont montré clairement l'efficacité de la logique floue type-2 dans la commande du robot mobile (véhicule) par rapport au contrôleur type 1

Nous avons choisi un contrôleur floue pour la simplicité relative d'écriture de celui-ci à travers des règles type système expert , la prise en compte au niveau même du contrôleur des information imparfaites (imprécision notamment) et la possibilité de disposer d'un système relativement ouvert en ce qui concerne la mise à jour de la base de règle.

Le contrôleur devra suivre le chemin de référence en l'amendant d'une manière réactive pour s'adapter à la situation courante s'appuyant sur une base de règle. Ce contrôleur est introduit dans une boucle ferme "bas niveau" : l'entrée du contrôleur reliée avec les capteurs du robot et la sortie avec les actionneurs du robot .:

Les valeurs mesurées avec les capteur sont des valeurs crisp qui sont converties en variables linguistiques de l'entrée du contrôleur où ils sont traités dans la base d'inférence pour produire la sortie defuzzifiée qui donne la valeur ou la commande pour les actionneur du robot. Le choix des fonctions d'appartenance et la construction de la base de règle a été faite d'une manière naturelle.

Dans la phase de simulation nous avons proposé une nouvelle méthode de la commande du robot mobile basée sur le système flou type-2. Nous avons utilisé le même principe que celui cité dans la première partie à la différence que nous avons changé le contrôleur flou type-1 par un contrôleur flou de type-2. On a suivi les mêmes étapes pour la construction de contrôleur en utilisant les fonctions d'appartenance type 2 et on ajoutant une étape de type réduction avant l'étape de defuzzification, le reste du programme reste similaire au contrôleur flou type 1.

Les tests de simulation que nous avons effectués en appliquant les deux types de contrôleurs ont donné des résultats satisfaisants du point de vue commande du robot avec une supériorité pour le contrôleur type deux

Bibliographie

- [01] Philippe Garnier," Contrôle d'exécution Réactif de Mouvement de Véhicules en environnement dynamique structure" thèse doctorat Institut national polytechnique de Grenoble, France Décembre 1995.
- [02] David Filliat "Robotique mobile " cours C 10-2 ENSTA Octobre 2004
- [03] Patricia Mellodge "Feedback control for path following robotic car" theme Master Institute Virginia polytechnic and stat university Blacksburg April 2002
- [4] Selig. J. M «Introductory Robotics» First published 1992 by prentice Hall international (ut) ltd.
- [05] Laumond. J “ Robot motion planning and Control “ Springer 1998
- [06] Samson. C” Control of chained systems, application to path following and time varying point - stabilization of mobile robots” IEEE Trans. On Automatic control , Vol 40,N° 01 PP 64-77 ,1995
- [07] Andrew Payne, “Design and Construction of Pair of cooperating Autonomous Mobile robots” Theme Master of science in physics and Electronic Engineering at the university of Waikato 2004.
- [08] Ricardo S. Sanchez – Pena, Mario SZnaier " Robust systems- theory and applications" A wely – interscience Publication John Wily & Sons ,INC 1998
- [09] Andrew Payne, “Design and Construction of Pair of cooperating Autonomous Mobile robots” Theme Master of science in physics and Electronic Engineering at the university of Waikato 2004.
- [10] Gareth Brawn, “Absolute Beginner’s Guide to Building Robots” August, 11; 2003.
- [11] Pete Miles, Tom Carroll, “Built your Own Combat Robot” Mc Graw – Hill copyright 2002.
- [12] Ming Chieh Lin, “ Efficient Collision Detection for Animation and Robotics” Doctor of philosophy in Engineering - Electrical Engineering and Computer Science in the

Graduate Division of the university of California at Berkeley 1993.

- [13] Roland Siegwart and ILLah R Nourbakhsh « Introduction to Autonomous Mobile Robots » a Bradford book the MIT Press Cambridge Massachusetts London England copyright 2004 Massachusetts institute of technology.
- [14] John Holland, « Designing Autonomous Mobile Robots » Newness; copyright 2004, ELSevier Inc.
- [15] Thomas R. Kurfess PhD. P, E « Robotics and Automation hand book » CRC Press LLC copyright 2005.
- [16] David R. Shircliff, « Build a Remote Controlled Robot » MC Graw – Hill Companies copyright 2002.
- [17] Luis Eduardo Aguilier Melo « Commande robuste et Coordination de mouvement de robot mobile » doctorat de l’université Paul Sabatier de Toulouse1997.
- [18] Robert Babuska, "fuzzy and neural control Disc course lecture notes (2001)" delft university of technology, Delft, the Netherlands publication copyright 2001.
- [19] Hung T. Nguyen, Albert A. Walker "A first course in fuzzy logic" second edition chapma, and Hall / CRC 2000.
- [20] KAZUO Tanaka and HUA O.WANG “Fuzzy Control systems Design and Analysis A linear Matrix Inequality Approach” A Wiley interscience publication copyright 2001 John Wiley & Sons, Inc.
- [21] Olef Wolkenhaner « Data Engineering Fuzzy Mathematics in Systems theory and Data Analyses » copyright 2001, by John Wily & Sons, Inc.
- [22] Kevin M. Passino Stephen Yurkovich “Fuzzy Control” Addison Wesley Longman, Inc Copyright 1998

- [23] Laumond. J, "Robot Motion Planning and Control" by A Deluca, G oriolo, C.Samson, Springer – Verlag 1998-ISBN 3-540-762191.
- [24] Karnik.N and Mendel. M “An introduction to type 2 fuzzy logic system” Univ southern california, los angeles, CA. USA [on line] Available:<http://Sipi.Vsc.educ~mendel/report> 1998.
- [25] Nilnesh N Karnik, jerry M .Mendel, Fellow, IEEE, and Qilian liang "Type 2 Fuzzy Logic systems» IEEE Transaction on fuzzy systems, VOL. 7. No6 December 1999.
- [26] Hani A .Hagras, Member, IEEE "A Hierarchical Type 2 Fuzzy Logic Control Architecture for Autonomous Mobile robots" IEEE transactions on Fuzzy Systems, vol, 12, No,4 , August 2004.
- [27] Qilian Liang, Jerry M. Mendel “An Introduction to Type 2 TSK Fuzzy Logic Systems” university of Southern California. IEEE International Fuzzy System Conference Producing, August 22-25, 1999 Seoul Korea.
- [28] Jerry .M. Mendel, Robert I, Bob john “Type 2 Fuzzy Sets Made simple” IEEE transactions on Fuzzy System, vol 10, No 2, April 2002.
- [29] Liang. Q Karnik. N and Mendel. J “Connection admission control in ATN net work using survey- based type 2 fuzzy logic systems” IEEE trans, sys, Man, cybern. C, vol 30 pp329-339,Aug2000
- [30] Liang. Q and Mendel. J “Interval type 2 Fuzzy logic systems: theory and Design”, IEEE Trans, Fuzzy sets syst, vol 08 PP535-550, Oct 2000
- [31] Jerry Mendel “Uncertain Rule-Based Fuzzy logic Systems: Introduction and new Directions”. Upper saddle river. Nj: Prentice Hall 2001
- [32] Dongming Wang and Levent Acar "An Analysis of type 1 and type 2 Fuzzy logic Systems» Proceedings of the 1999 IEEE International symposium on Intelligent

control/ Intelligent systems and semiotics Cambridge, MA” September 15-17,1999

- [33] Jerry M. Mendel, *Life Fellow, IEEE*” Computing Derivatives in Interval Type-2 Fuzzy Logic Systems” IEEE Transaction on Fuzzy Systems, Vol 12, N°1, February 2004
- [34] Jerry Mendel, “Fuzzy logic systems for engineering: A tutorial,” IEEE proceedings, vol. 83, no. 3, pp. 345-377, March 1995.
- [35] Nilesh N, Karnik and Mendel J “ Applications of type -2 fuzzy logic systems: handling the uncertainty associated with surveys “ IEEE international fuzzy systems Conference proceedings August 22-25 ,1999, Seoul, Korea

Résumé

Le contrôleur de la logique floue (LFC) est crédité d'être une meilleure méthodologie pour concevoir les contrôleurs robustes, peuvent fournir une exécution satisfaisante de l'incertitude et de l'imprécision; en conséquence il est devenu ces dernières années une approche populaire à la commande d'un robot mobile.

Dans notre travail nous introduisons un système robuste de logique floue qui peut manipuler de n'importe quelle incertitude comme l'incertitude des règles pour pouvoir produire une meilleure exécution.

L'utilisation d'un tel système avec les ensemble floue de type 2 permet l'appariation d'un autre système appelé système de logique floue type 2 (FLS type 2).

L'implémentation de système floue type 2 implique les opérations suivantes: la fuzzification, l'inférence et la sortie du processus "defuzzification qui amène à une nouvelle opération appelé : réduction de type.

On applique ce concept à un robot mobile suivi d'un chemin.

Mots clés: système floue type 1 et type 2 contrôleur floue robot mobile

Abstract

The fuzzy logic controller (FLC) is credited with being an adequate methodology for designing robust controller that able to deliver a satisfactory performance in face uncertainty and imprecision as a result the FLC has become a popular approach to mobile robot in recent years.

In our work; we introduce a robust fuzzy logic system which can handle uncertainties as uncertainties in rules, which can produce a best execution.

The use such system allow appearance author system call type 2 fuzzy logic system (FLS).

The implementation of this type 2 FLS involves the operation of fuzzification, inference and output processing "defuzzification" that lead to novel operation call type reduced.

We apply this approach to mobile robot flow a path

Key words: type1 and type 2 fuzzy logic system, fuzzy controller, mobile robot.

مراقب المنطق الغامض كاد يكون من أحسن الطرق لتصوير و فهم المراقبات القوية التي يمكنها التنفيذ بدقة. نتيجة لذلك أصبحت في هذه السنوات من بين الطرق التي لها شعبية كبيرة خاصة في التحكم الآلي (الإنسان الآلي (الإنسان الآلي).

في هذا العمل نقدم نظام المنطق الغامض الذي يستطيع معالجة أي ترتيبات مثل الارتياح في العلاقات للتمكن من التنفيذ الجيد.

استعمال مثل هذه الأنظمة مع الوحدة الغامضة نوع 2 يسمح بظهور نوع جديد من الأنظمة تسمى نظام المنطق الغامض نوع 2.

تنفيذ النظام الغامض نوع 2 يسد تلزم العمليات التالية: إسناد القيمة العددية للمدخل إلى دالة الانتماء (fuzzification), معالجة و استدلال الانتماء (inférence), معالجة المخرج "defuzzification" نمر من خلالها إلى عملية جديدة تسمى نمط الاختزال أو التخفيض "type réduit" تطبق هذه الطريقة على آلي متحرك.