



Université Blaise Pascal
U.F.R. Sciences Exactes et Naturelles



LABORATOIRE DES SCIENCES ET MATERIAUX POUR
L'ELECTRONIQUE,
ET D'AUTOMATIQUE (U.M.R.6602 du CNRS)

MEMOIRE

Présenté par

Mouloud IDER

En vue de l'obtention du

MASTER STIC : Informatique MSIR
« RPM »

Contrôle/commande par logique floue d'un groupe de
robots mobiles pour une navigation en convoi

le 29/06/2010

Encadré par : Lounis ADOUANE

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre 1 : La robotique coopérative

1.1 Les robots.....	4
1.2 Autonomie d'un robot.....	4
1.3 Contrôle des robots mobiles.....	5
1.4 Les architectures de contrôle.....	6
1.4.1 Architecture hiérarchique (cognitive).....	7
1.4.2 Architecture réactive.....	8
1.5 Coordination de comportements.....	9
1.6 La robotique collective.....	10
1.7 La coopération de robots.....	12
1.8 Les applications de la robotique coopérative.....	12
1.9 Problème de navigation en convoi d'un groupe de robots mobiles.....	13
1.10 Conclusion.....	13

Chapitre 2 : La logique floue

2.1 Principes et définition.....	15
2.2 Opérateurs et normes.....	16
2.3 Logique floue.....	16
2.4 Modèles de contrôleurs flous.....	18
2.4.1 Modèle flou linguistique.....	18
2.4.2 Modèle flou type <i>Takagi-Sugeno-Kang (TSK)</i>	21
2.5 Concept des systèmes flous <i>type-2</i>	21
2.6 Systèmes et contrôleurs <i>flous type-2</i>	22
2.7 Traitement de la sortie d'un système flou <i>type-2</i>	23

2.8 Réduction de type des ensembles flous <i>type-2 intervalle</i>	23
2.8.1 Réduction de type par la méthode du centroïde.....	26
2.8.2 Réduction de type par la méthode des centres d'ensembles.....	26
2.9 Conclusion.....	27

Chapitre 3 : Navigation floue d'un seul robot mobile

3.1 Introduction.....	30
3.2 Modélisation.....	30
3.3 Attraction vers une cible dans un environnement sans obstacles.....	31
3.3.1 Conception du contrôleur d'attraction vers la cible.....	32
3.3.2 Résultats et commentaires.....	33
3.4 Comportement d'évitement d'obstacles.....	35
3.5 Attraction vers la cible en présence d'obstacles (coordination de comportements).....	37
3.5.1 Coordination de comportements par un switch (hard switch).....	37
3.5.2 Coordination de comportements avec un gain flou.....	38
3.5.3 Coordination de contrôleurs avec une sigmoïde versus un gain flou	42
3.6 Incorporation des incertitudes dans les contrôleurs flous.....	43
3.6.1 Simulation des incertitudes aléatoires avec un contrôleur flou <i>type-2</i>	44
3.6.2 Etude comparative des contrôleurs <i>type-1</i> et <i>type-2</i>	46
3.7 Introduction au mouvement en formation : attraction vers une cible dynamique.....	48
3.7.1 Cas d'une cible en mouvement rectiligne.....	48
3.7.2 Cas d'une cible en mouvement sinusoïdale.....	49
3.7.3 Simulation des incertitudes de mesure.....	51
3.8 Conclusion.....	52

Chapitre 4 : Navigation en convoi d'un groupe de robots mobiles

4.1 Introduction.....	54
4.2 Objectifs	54
4.3 Stratégie adoptée pour la réalisation d'un convoi autonome.....	54
4.4 Quel robot pour quelle cible dynamique ?	55
4.5 Structure de contrôle/commande du convoi.....	56
4.6 Conclusion.....	58
Conclusion générale.....	59
Bibliographie	
Annexes	
Annexe 1: Ensemble flou <i>type-1</i>	
Annexe 2: Principe d'extention de Zadeh	
Annexe 3: Terminologie de la logique floue <i>type-2</i>	
Annexe 4: Calcul des vitesses maximales admissibles par le <i>Khepera III</i>	

Introduction générale

Contrôler la coopération d'un groupe de robots mobiles est une thématique de recherche en plein essor. En effet, après l'intérêt des laboratoires de recherche pour le contrôle de robots mobiles pris individuellement, plusieurs travaux de recherche se sont orientés vers le contrôle d'une flottille de robots mobiles. C'est notamment le cas au *LASMEA* au sein de l'équipe *ROSACE*. Le travail effectué dans ce stage, qui consiste à proposer une architecture de contrôle réactive pour la navigation en formation en présence d'obstacles, s'inscrit dans cette optique.

Cependant, la maîtrise du contrôle d'un groupe de robots mobiles passe inévitablement par la maîtrise des actions engendrées par les entités robotique élémentaires, et, la coordination entre ces actions. Notre objectif consiste de proposer une architecture de contrôle qui exhibe des caractéristiques individuelles adaptées à des systèmes *multi-robots*. Cela nécessite néanmoins de s'éloigner davantage des architectures *centralisées* et *cognitives* pour tendre vers des structures *distribuées* et *réactives* telle que les *architectures de contrôle comportementales* investiguées durant ce travail.

D'autre part, devant le délicat problème de modélisation des systèmes complexes, les méthodes de commande nécessitant une bonne connaissance du modèle mathématique s'avèrent incapables de répondre aux performances exigées. Afin de répondre à ces exigences toujours croissantes, les algorithmes de commande développés dans la littérature deviennent de plus en plus complexes. Alors, la *commande floue* s'est présentée, ces dernières années, comme l'une des alternatives.

Une des contraintes les plus importantes du fonctionnement des architectures comportementales est le choix des mécanismes de coordination des actions engendrées par les différents comportements élémentaires. En effet, ceci constitue un domaine important de recherche qui a conduit au développement de différents *mécanismes de coordination* entre comportements. Parmi les plus représentatifs, nous pouvons citer la subsomption de *Rodney Brooks* [2] et les schémas moteurs de *Ronald Arkin*. Dans ce stage, une étude comparative entre trois méthodes de coordination de comportements a été effectuée à savoir : la sélection d'actions par un switch entre comportements (*hard switch*), la fusion par un *gain adaptatif flou* et la fusion par une *sigmoïde figée*.

Pour une meilleure présentation de notre travail, le *manuscrit* est organisé en quatre chapitres :

Le premier chapitre porte sur des généralités sur la robotique mobile, la robotique collective et la coopération de robots mobiles. Les différentes architectures de contrôle dédiées à la robotique sont aussi exposées dans ce chapitre.

Le deuxième chapitre est consacré à l'exploration de la technique -logique floue- à laquelle on a eu recours pour contrôler un/ou un groupe de robots mobiles. Les notions d'ensembles flous, normes et opérateurs, et les modèles de contrôleurs flous sont exposés.

Le troisième chapitre, traite le problème de navigation floue d'un robot mobile dans des environnement contenant des obstacles. Les *comportements d'attraction vers la cible* et *d'évitement d'obstacles*, et de *fusion de comportements* sont simulés. Une autre contrainte caractérisant un environnement réel, présence *d'incertitudes de mesure*, a été étudiée et les performances des systèmes flous *type-2* sont comparées à celles du *type-1*. A la fin du chapitre, le *mouvement de formation* est introduit à travers l'attraction du robot mobile vers une *cible dynamique*.

Le quatrième et dernier chapitre, aborde le problème de navigation en formation d'un groupe de robots mobiles. Une architecture de contrôle réactive exploitant une structure à cibles virtuelles y est proposée.

Chapitre 1

La robotique coopérative

***Résumé :** Ce premier chapitre se veut une introduction à la problématique de base qui consiste à faire coopérer un groupe de robots mobiles autonomes, en vue d'une navigation en convoi, objectif principal de ce stage. Après avoir introduit brièvement les caractéristiques et les contraintes inhérentes au fonctionnement d'une seule entité robotique, nous exposerons l'intérêt d'introduire une forme d'intelligence collective afin de réaliser une tâche où plusieurs entités robotiques participent. Une partie importante de ce chapitre sera consacrée aux différentes architectures de contrôle pour la robotique. Les avantages et les limites de chacune de ces architectures seront mis en avant. Ceci, nous permettra de justifier et de motiver notre choix. En fin du chapitre, on citera les approches de commande qui permettent de réaliser une telle tâche.*

1.1 Les robots

Le mot *robot*, qui veut dire - travail forcé – en tchèque, est introduit dans le vocabulaire humain pour la première fois, dans une pièce de théâtre *RUR (Rossums Universal Robots)*, par l'écrivain tchèque Karel Capek en 1920.

Un robot est un dispositif susceptible d'accomplir une ou plusieurs fonctions. Il se compose principalement d'un socle, dans le cas d'un manipulateur ce dernier est muni d'un ou de plusieurs bras articulés, équipés de systèmes de préhension (pinces, ventouses, etc.), d'une source d'énergie, d'actionneurs, capteurs internes et capteurs externes et d'un système de traitement d'information.

Deux sortes de robots existent :

- robots *manipulateurs* : robots ancrés physiquement à leur place de travail et, généralement, mis en place pour réaliser une tâche précise ou répétitive.
- robots *mobiles* : robots capables de se déplacer dans un environnement et peuvent supporter des manipulateurs suivant leur utilisation.

En industrie une grande part est réservée aux robots manipulateurs ; vu leur intérêt économique. Ces derniers sont utilisés dans les chaînes d'assemblage et permettent d'accélérer la cadence de production. Les tâches réalisées sont programmées de sorte que les manipulateurs reproduisent des mouvements séquentiels et répétitifs [1].

Un intérêt particulier est attribué ces dernières décennies aux robots autonomes. Contrairement aux premiers, ils sont dotés d'une certaine intelligence leurs permettant de prendre des décisions ou de choisir les actions à exécuter pour accomplir les tâches qui leurs sont confiées. Dans cette perspective, les robots mobiles ont suscité un intérêt particulier chez les chercheurs, puisque ces derniers ne souffrent pas de l'handicape des manipulateurs (incapacité à se déplacer).

1.2 Autonomie d'un robot

L'obtention de l'autonomie complète d'un robot mobile est conditionnée par l'adjonction de plusieurs types d'autonomie, à savoir : *énergétique, sensorielle, et décisionnelle* [1] [2].

L'autonomie énergétique est fortement souhaitée, cependant, elle n'est pas obligatoire pour toutes les tâches à accomplir par le robot. En fait, elle dépend essentiellement des technologies de production et de stockage de l'énergie nécessaire au robot.

D'autre part, l'autonomie *sensorielle* et *décisionnelle* sont toujours liées et le robot doit suivre le schéma correspondant au paradigme *percevoir-décider-agir* [2].

- *percevoir* : le robot doit acquérir des informations de son état interne et de l'état de l'environnement dans lequel il évolue, et ce, via ces capteurs. Dans le cas où le robot évolue dans un environnement inconnu, ces informations sont, pour lui, la seule source sur ce milieu. Ces dernières peuvent être utilisées pour avoir une représentation de l'environnement lui-même. Si l'environnement est statique, cette dernière peut être exploitée pour de prochaines missions.

- *décider* : selon l'architecture de contrôle adoptée, le robot définit l'action que doit entamer. Pour une architecture hiérarchique ou délibérative (cf. figure 1.2.a), il s'agit d'une séquence d'actions ; puisque le robot dispose d'un modèle de l'environnement. Dans le cas d'une architecture réactive (cf. figure 1.2.b), le robot réagit d'une manière réflexe aux stimuli des capteurs.

- *agir* : enfin, le robot exécute les séquences d'actions en envoyant des consignes aux actionneurs par l'intermédiaire des asservissements.

La manière dont un robot gère ces différents éléments est définie par son architecture de contrôle.

1.3 Contrôle des robots mobiles

Le contrôle des robots mobiles nécessite la juxtaposition des trois tâches suscitées (la *perception*, la *décision*, et l'*action*). En outre, il est possible de faire l'analogie entre la tâche que doit réaliser un robot mobile et la représentation habituelle en automatique d'un système à asservir (cf. figure 1.1). Dans ce cas, le bloc '*Contrôleur*' correspond au contrôleur du robot mobile, quant au bloc '*Système*' il correspond au bloc immergé dans son environnement via ses capteurs. Après un traitement approprié de ses perceptions au niveau de son contrôleur, une commande est produite afin de satisfaire au mieux la consigne qui est dictée entre autre par la tâche à réaliser. L'exécution de la commande par les effecteurs du robot permettra à ce

dernier de changer son état ainsi que l'état du système global (modification de la sortie), ce qui referme la boucle de retour [3].

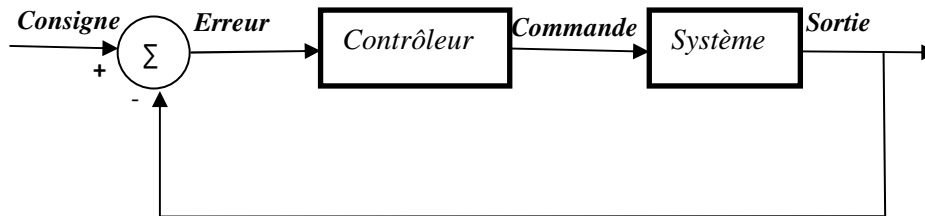


Fig.1.1 - Structure d'asservissement conventionnel

1.4 Les architectures de contrôle

Les architectures de contrôle sont généralement divisées en trois groupes [2], dont les deux premières se sont longtemps opposées [1] [4] [5]:

-*approche délibérative* : utilise une modélisation de l'environnement connu *a priori* ou obtenue à partir des données capteurs, pour planifier à l'avance les commandes que le robot doit exécuter.

-*approche réactive* : ne suppose aucun modèle *a priori* de l'environnement. Cette architecture s'appuie sur le couplage étroit entre les capteurs et les actionneurs, pour générer en continu les commandes que le robot doit exécuter.

-*approche hybride* : tente de combiner les deux approches précédentes afin de tirer partie des avantages respectifs de ces deux premières approches.

Ces trois approches (cf. figure 1.2) ne diffèrent pas dans leurs détails de contrôle, mais plutôt, par leur agencement et leurs relations.

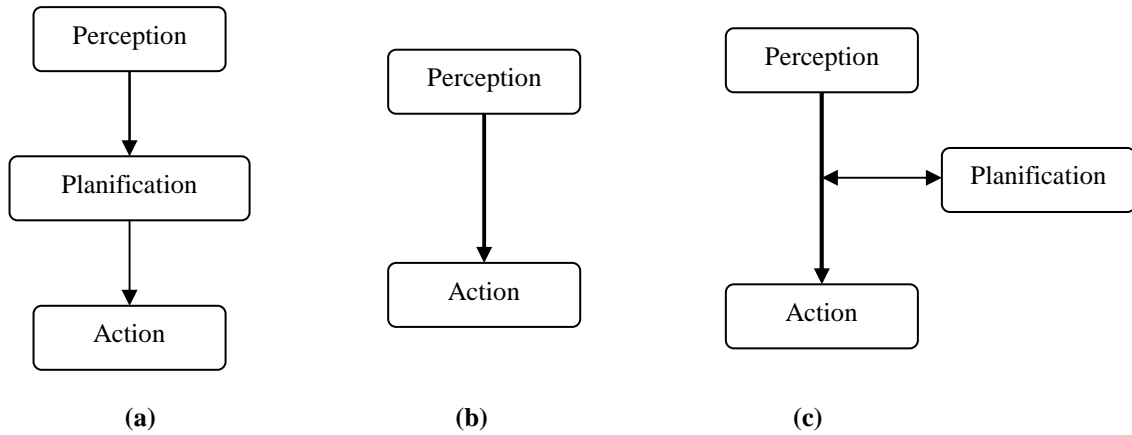


Fig.1.2 - Architectures de contrôle pour les robots mobiles : (a) hiérarchique, (b) réactive, (c) hybride

1.4.1 Architecture hiérarchique (cognitive)

Les premiers robots mobiles, obtenus des recherches en intelligence artificielle, utilisaient des contrôleurs hiérarchiques dont le fonctionnement repose essentiellement sur la capacité de décision travaillant sur un modèle de l’environnement supposé parfait. Ces architectures ont l’avantage de prouver l’existence d’une solution optimale permettant au robot d’atteindre le but assigné. Elles fonctionnent selon un cycle rigide de modélisation de l’environnement, planification des actions au sein de cette représentation, puis exécution du plan. Cette vision de la robotique autonome conduit à une décomposition séquentielle du traitement réalisé et à des systèmes fortement hiérarchiques.

Le traitement est décomposé en une série d’opérations successives décrites par la figure ci-dessous :

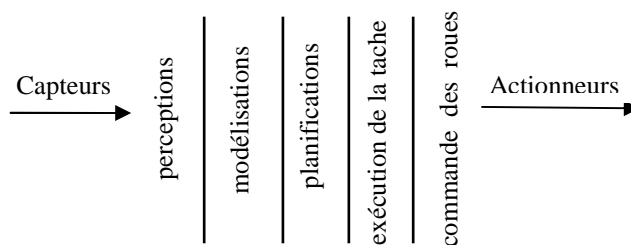


Fig.1.2.a - Architecture de contrôle hiérarchique

Cette première architecture ne garantit pas la réussite de la tâche que dans le cas des environnements connus et statiques ; puisque la planification est faite au début. Dans le cas de changement de la configuration de l'environnement, le plan établi n'est plus valable. De plus, les temps de calcul pour les déplacements des robots sont très grands.

C'est dans ce contexte que de nouvelles méthodes de contrôle s'imposèrent dans le domaine de la navigation autonome de robots. Ces méthodes doivent permettre de surmonter ce problème. Ceci est possible en établissant un couplage plus étroit entre la perception et l'action et ce en faisant des traitements locaux.

1.4.2 Architecture réactive

En 1986, Brooks propose une approche concurrentielle, sous la forme d'une architecture *réactive* qui se distingue par l'abandon des phases de modélisation et de planification. Sans aucun modèle de l'environnement, cette architecture se compose d'un ensemble de comportements réactifs fonctionnant en parallèle pour le contrôle du robot (cf. figure.1.2.b).

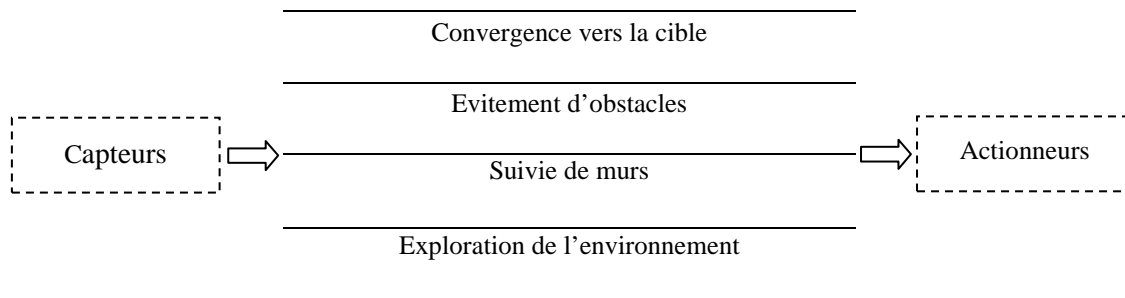


Fig.1.2.b - Architecture réactive

Cette approche, d'inspiration éthologique ou physiologique, tente de reproduire les principales propriétés du comportement des êtres vivants. Elle est basée sur plusieurs comportements (*convergence vers la cible, évitement d'obstacle, suivi de murs, exploration de l'environnement,...*etc). Pour guider le robot, il faut choisir à chaque instant lequel de ces comportements activer ou comment les combiner. Ce problème est connu dans la littérature sous le nom de *sélection d'action*. La solution proposée par Brooks [2], *l'architecture de*

subsumption, est devenue un classique. Elle utilise une hiérarchie des comportements qui s'exécutent selon un ordre de priorité en fonction des *perceptions* du robot.

Les tâches sont parallélisées. Dans cette hiérarchie, le concept de base est le suivant : un problème (comportement) donné, si complexe soit-il, peut être décomposé en sous problèmes (comportements) élémentaires, représentant chacun un niveau de compétence. Ces différents niveaux sont placés en couches (hiérarchie de couches) (cf. figure 1.2.c).

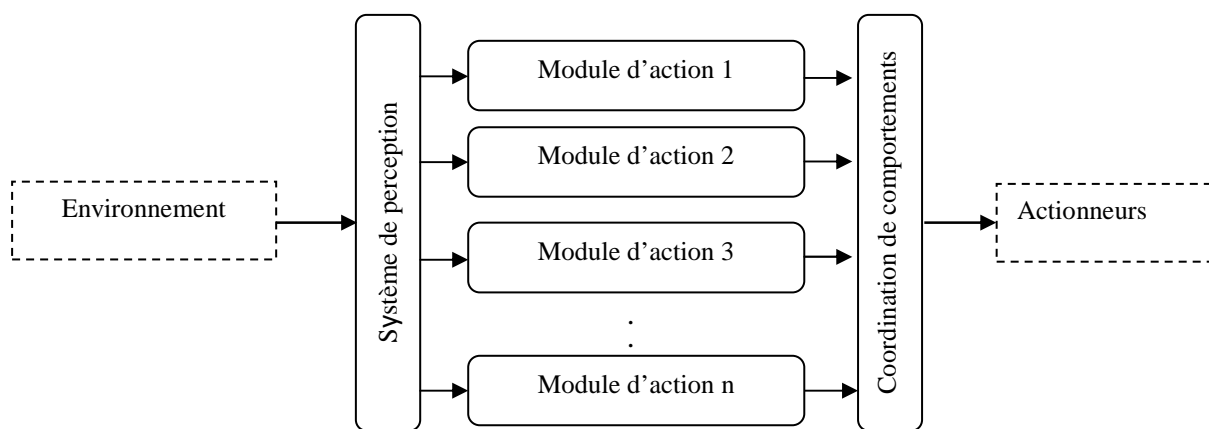


Fig.1.2.c - Structure d'une architecture comportementale

Cette architecture, adoptée dans ce travail, présente au moins quatre avantages :

- elle peut réagir aux éventualités en temps réel, et ceci grâce au parallélisme.
- chaque comportement a une tâche qui lui est confiée. Ce qui facilite leur implantation, ainsi que l'insertion ou la suppression de modules.
- elle présente une bonne robustesse ; le système peut rester fonctionnel, même si il y a échec d'un ou plusieurs comportements.
- cette approche est plus adaptative à l'environnement. Même pour un environnement dynamique, on a plus de chance pour la réussite de la tâche.

1.5 Coordination de comportements

Avant de passer au contrôle d'un groupe de robots mobiles comme nous souhaitons de le faire dans le cadre de ce stage, il est impératif de pouvoir contrôler au préalable le

fonctionnement d'une seule entité robotique. D'autres part, et dans le cas des architectures comportementales, la manière de coordonner entre les actions générées par une multitude de comportements appartenant à la même architecture de contrôle est une question cruciale est prépondérante pour la réussite de la tâche désirée. Dans ce souci, on distingue deux mécanismes principaux [3]:

-*sélection d'action* : dans cette branche, la commande à envoyer aux actionneurs à chaque pas d'échantillonnage correspond à la commande générée par un *seul* comportement élémentaire. Ce comportement est choisi d'une manière hiérarchique.

-*fusions d'actions* : comme son nom l'indique cette branche correspond aux mécanismes qui vont fusionner deux commandes ou plus (émanant de comportements distincts) pour aboutir à une seule commande. Ceci correspond donc à un certain consensus entre comportements, e.g., les schémas moteurs ou le contrôle par logique floue (l'approche adopté dans ce travail).

1.6 La robotique collective

La robotique collective est synonyme d'existence d'un groupe de robots. Elle stipule non seulement la mise en place d'un contrôle individuel pour chaque robot, mais impose aussi l'utilisation de stratégies de contrôle appropriées afin que l'assemblage de toutes ces entités robotiques engendre des configurations cohérentes et efficaces pour la réalisation des tâches désirées.

Généralement, faire appel à un groupe de robots au lieu d'en utiliser qu'un seul est motivé par deux facteurs majeurs [3]:

- 1- soit la tâche à exécuter nécessite impérativement la coopération d'un nombre minimal de robots, i.e., que la tâche en question ne peut s'accomplir sans l'intervention simultanée d'un nombre critique N_C de robots.
- 2- soit l'amélioration de certaines performances liées à l'exécution des tâches à réaliser, comme par exemple :
 - *rapidité* : on cherche à obtenir un niveau de performance élevé en tirant parti de la parallélisations des tâches. e.g., l'exploration parallèle d'un environnement inconnu par un ensemble de robots mobiles, dans le but de faire soit une cartographie de l'environnement, soit la réalisation d'une tâche de fourragement. Ces deux tâches

peuvent être réalisées par un seul robot, mais l'ajout d'autres robots va faire en sorte d'accélérer l'exécution des tâches en question,

- *robustesse – fiabilité* : les performances du contrôle peuvent être très peu affectées en cas de défaillance d'un agent,
- *flexibilité* : possibilité d'exécuter les tâches désirées, de diverse manière. Ceci est induit principalement par la redondance des entités robotiques,
- *émergence* : l'idée ici est de produire une performance collective qualitativement supérieure à celle de l'addition des unités.

Cependant, le fait de contrôler/coordonner plusieurs robots au lieu de n'en contrôler qu'un seul complexifie considérablement la commande du système, et ce vu l'augmentation :

- de la dynamique des interactions entre entités robotiques dans l'environnement. Ces interactions sont susceptibles, si elles sont mal maîtrisées, d'influer d'une manière néfaste sur l'évolution du système, car les robots peuvent se gêner, se bloquer, se désynchroniser, etc.

- du nombre de variables régissant l'évolution du système, dû directement à l'augmentation du nombre de systèmes (robots) dans l'environnement,

- de la complexité du contrôle inhérent déjà à un seul robot, qui doit non seulement agir en fonction des stimuli lui parvenant de l'environnement mais doit aussi adapter son comportement à ceux des autres individus. C'est - à-dire que le robot va essayer de tendre vers un équilibre viable voire optimal pour l'exécution de la tâche coopérative,

- des incertitudes perceptuelles des robots. Celles-ci sont dues principalement à la nature hautement dynamique qui peut être engendrée par le système multi-robots.

Ces incertitudes peuvent complexifier aussi davantage le contrôle du système si elles sont effectives au niveau d'un grand nombre de capteurs.

Les points cités ci-dessus, sont parmi ceux qui font que le contrôle d'un système multi-robots, est plus délicat à mettre en place que le contrôle d'un seul robot élémentaire. La figure (1.3) montre la structure de commande analogue à celle d'un asservissement d'un système pour réaliser une tâche multi-robots [3].

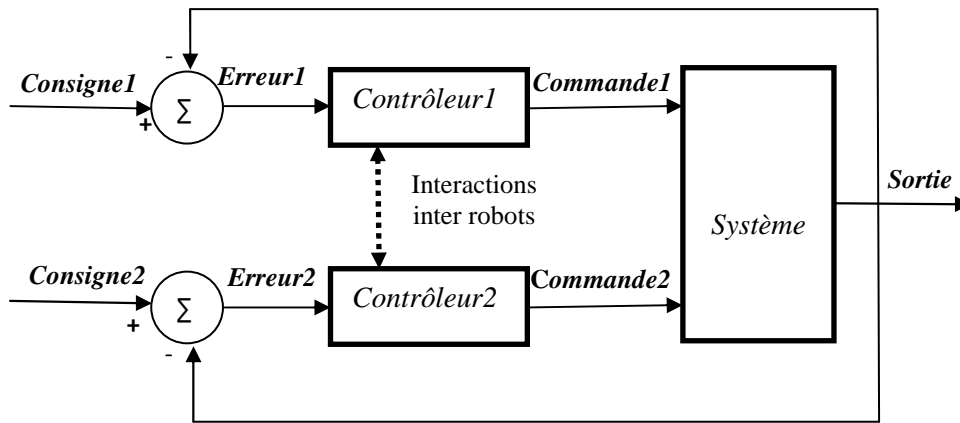


Fig.1.3 - Structure d'asservissement d'un système multi-robots

1.7 La coopération de robots

On parle de la robotique coopérative dans le cas où les robots partagent et/ou interviennent simultanément sur des ressources communes. Cela signifie que, d'une part les robots doivent évoluer dans le même environnement, et d'autre part qu'ils sont appelés à coordonner leurs actions pour réaliser la tâche coopérative. Cette coopération peut être très élaborée, et donc obtenue via des mécanismes de très haut niveau, que l'on peut qualifier de *cognitifs*. Comme elle peut être complètement émergente des interactions induites par les robots, et dans ce cas de figure l'architecture de contrôle est qualifiée de *réactive* (§1.4.2) [2].

1.8 Les applications de la robotique coopérative

Les projets applicatifs potentiels et effectifs impliquant un ensemble de robots mobiles, sont nombreux et très prometteurs. Ceci est dû principalement aux nombreuses plus-values qualitatives et quantitatives qui peuvent être apportées par rapport aux approches traditionnelles existantes. Parmi les applications majeures qui tirent et qui pourraient tirer profit des recherches menées sur la coopération d'un groupe de robots autonomes, nous pouvons citer : les applications militaires telles que le maintien de patrouille en formation, l'exploration spatiale, le transport coopératif, fourragement ...etc [3].

Les tâches génériques en robotique collective, ont comme principal objectif de fournir des tâches de références pour évaluer les architectures de contrôle proposées dans la littérature. Parmi les tâches génériques, couramment traitées en robotique coopérative on trouve le mouvement en formation ou le mouvement en convoi, objet de notre application [6].

1.9 Problème de la navigation en convoi d'un groupe de robots mobiles

Contrôler un système multi-robots est un challenge à la fois scientifique et technologique en plein essor. Dans cette optique, une des facultés premières que doit intégrer chaque entité autonome est la capacité de se mouvoir et de converger vers des lieux (cibles) qui lui sont indiqués au préalable sans aucune intervention externe, maintenir le mouvement de formation désirée, de plus, dans un environnement partiellement connu ou inconnu, des objets gênant peuvent surgir, pour cela, ces entités doivent être doté d'un module qui leur permet d'éviter ces obstacles. Pour atteindre ces objectifs, plusieurs méthodes sont proposées dans la littérature, néanmoins, deux grandes familles peuvent se distinguer : d'une part, les approches nécessitant un modèle mathématique du système, et d'autre part, on trouve les techniques issues des recherches de l'intelligence artificielle et en informatique où il est possible de dispenser d'un modèle explicite. Parmi ces techniques il existe notamment les réseaux de neurones artificiels, et la *logique floue*.

1.10 Conclusion

Ce premier chapitre est une introduction à la robotique coopérative, notamment, la problématique de navigation en convoi d'un groupe de robots mobiles (objet du stage). Pour cela, les notions de robot, de coordination d'actions (comportements), et la coopération de plusieurs entités robotiques sont décrites. Les architectures de contrôle/commande dédiées la robotique sont aussi étudiées. Dans le chapitre suivant, on explorera, en détail, l'outil (*logique floue*) qui nous permettra de résoudre un tel problème.

Chapitre 2

La logique floue

Résumé : Dans ce chapitre nous exposerons d'une manière détaillée, l'outil (logique floue) qui nous permettra de traiter la problématique de navigation en convoi d'un groupe de robots mobiles. A cet effet, les notions : ensembles flous, opérateurs et normes et celle de règles floues seront introduites. Ensuite, on donnera les structures des contrôleurs flous version classique dite de type-1 ainsi que celle du type-2. Le rôle de chaque module (bloc) constituant ces contrôleurs sera expliqué. Une partie importante de ce chapitre sera consacrée à l'étude du bloc de traitement de sortie d'un système flou de type-2 qui présente une difficulté liée essentiellement à la nécessité de la réduction de type des ensembles résultants de l'inférence floue. A cet effet, une procédure algorithmique de réduction de type des ensembles flous type-2 intervalle, relativement facile à celle de réduction de type d'un système flou type-2 général, sera présentée. Enfin, un exemple de ladite procédure complètele chapitre.

2.1 Principe et définition

Depuis quelques années déjà, on trouve sur le marché des appareils (photos, vidéos,..etc) qui sont présentés faisant intervenir un réglage par logique floue. Il est intéressant de comprendre ce concept de réglage en vue de l'appliquer sur d'autres types de systèmes comme les robots mobiles.

La logique floue exploite les ensembles flous développée par Zadeh en 1965 [7]. A côté d'un formalisme mathématique fort développé, nous préférons aborder la présentation de manière intuitive.

Les notions de température moyenne ou de courant faible sont relativement difficiles à spécifier de manière précise. En logique floue, Le concept de variable linguistique, comme son nom le suggère, joue un rôle très important. C'est une variable dont les valeurs sont des mots ou des phrases au lieu de nombres. Par exemple, ' *Température* ' est une variable linguistique et l'ensemble des valeurs qu'elle peut prendre est appelé l'ensemble de termes (nuances), soient par exemple : *Basse*, *Moyenne*, *Elevée*. D'autres termes peuvent être construits à l'aide de modificateurs linguistiques tels que (très, plus ou moins, un peu...etc.). Chaque valeur linguistique (e.g., faible) est caractérisée par sa fonction d'appartenance $\mu_{faible}(x)$ qui associe à chaque valeur numérique x de ' *Température* ', un degré qui reflète la vérité du sens faible (cf. figure 2.1) .

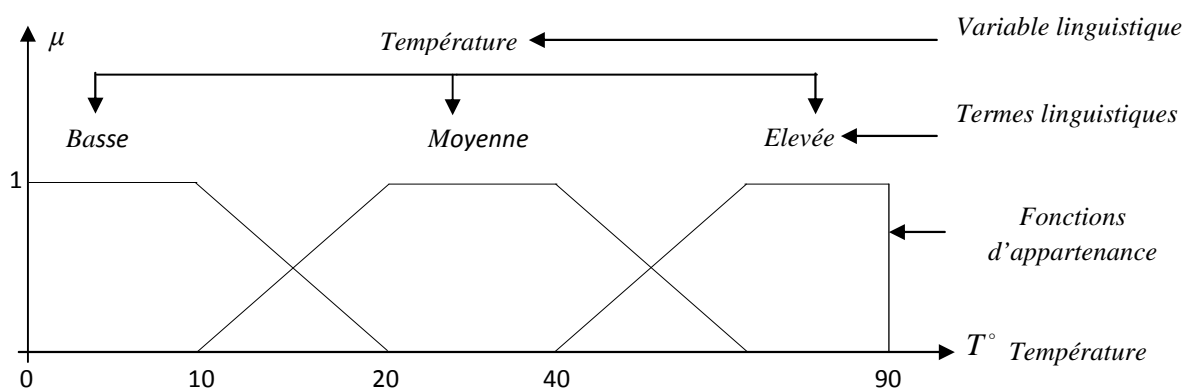


Fig.2.1 - Exemple d'une variable linguistique ' *Température* ' avec trois termes linguistiques.

2.2. Opérateurs et normes

Comme dans la théorie des ensembles classiques, on définit l'intersection (\cap), l'union (\cup) des ensembles flous ainsi que le complémentaire ($\bar{}$) d'un ensemble flou. Ces relations sont traduites par les opérateurs 'et', 'ou' et 'non'. De nouvelles fonctions d'appartenance liées à ces opérateurs sont établies [7] [8] :

- L'intersection : $x \in A \cap B \Leftrightarrow x \in \mu_{A \cap B} \mid \mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)]$ (2.1)

- L'union : $x \in A \cup B \Leftrightarrow x \in \mu_{A \cup B} \mid \mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)]$ (2.2)

- Le complément (négation) : $\forall x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$ (2.3)

L'intersection floue de deux ensembles flous peut être définie d'une façon plus générale par une opération binaire sur l'intervalle unitaire, c-à-d, une fonction de la forme $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$. Pour que la fonction T soit qualifiée pour assurer une intersection floue, elle doit vérifier certaines propriétés [7] [8]. Des fonctions connues par le nom *t-normes* (*normes triangulaires*) possèdent ces propriétés. Similairement, des fonctions appelées *t-conormes* peuvent être utilisées pour l'union floue [9].

2.3. Logique floue

La logique floue commence en empruntant des notions de la logique ordinaire, comme pour les ensembles flous qui eux aussi empruntent des notions de la théorie des ensembles ordinaires. Cependant, ce dernier passage est inadéquat pour les applications d'ingénierie. Car en ingénierie, le principe *cause-effet* est la pierre angulaire de tous les modèles. Ce qui n'est pas le cas dans la logique classique. Pour contourner ce problème, *Mamdani* a défini un opérateur d'implication (Engineering implication operator)[8].

Le principe d'extension des ensembles ordinaires vers des ensembles flous et l'extension de la logique classique vers la logique floue est réalisé en remplaçant des fonctions d'appartenance bivalentes de la logique classique par des fonctions d'appartenance floues à plusieurs degrés de vérité. Donc, la citation (règle) : *SI - ALORS* « *SI x est A, alors y*

est B », $x \in X$ et $y \in Y$, appelée en logique classique *modus ponens*, sera caractérisée, dans le cas de la logique floue, par un ensemble $\mu_{A \rightarrow B}(x, y) \in [0, 1]$ et le *modus ponens* est étendu vers le *modus ponens généralisé*.

- Modus ponens généralisé [8]

Prémisse 1 : ‘ x est A^* ’ ;

Prémisse 2 : ‘ si x est A , alors y est B ’ ;

Conséquence : y est B^* ’.

L’ensemble flou A^* n’est pas nécessairement égal à l’antécédent A de la règle, et l’ensemble flou B^* n’est pas nécessairement la même conséquence B de la règle. On dit $A \neq A^*$ mais A^* est similaire à A ; et $B \neq B^*$ mais B^* est similaire à B .

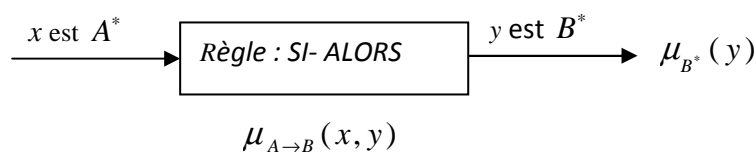


Fig.2.2 -Système interprétant le modus ponens généralisé.

L’opérateur d’implication de contrôle qui a été défini par Mamdani possède la forme suivante :

$$\mu_{A \rightarrow B}(x, y) = \min[\mu_A(x), \mu_B(y)] \quad (2.4)$$

Les raisons de ce choix étaient la simplicité de programmation. Plus tard, Larsen a proposé l’implication *produit* suivante :

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \cdot \mu_B(y) \quad (2.5)$$

Une fois de plus, la facilité dans la programmation a été la raison de ce choix. Aujourd’hui, les inférences *min* et *produit* sont largement utilisées dans les applications de la logique floue.

2.4 Modèles de contrôleurs flous

En fonction de la forme des conséquences des règles, deux types de modèles flous peuvent être distingués [10]:

- Modèle flou linguistique, où l'antécédent et la conséquence sont des propositions floues (on l'appelle aussi modèle flou de *Mamdani*).
- Modèle flou type Takagi-Sugeno-Kang (*TSK*), où la conséquence est une fonction ordinaire des variables d'entrée (antécédents).

Un système flou utilisant que des ensembles flous de *type-1* (le degré d'appartenance de chaque point de l'ensemble est une valeur singulière) (cf. figure 2.1) est dit de *type-1*. Dans la section qui suit, nous illustrons en détaille ces systèmes.

2.4.1 Modèle flou linguistique

La relation entrée/sortie du système flou est réalisée par le mécanisme d'inférence floue. Etant donné la base de règles et la valeur de l'entrée, ce dernier délivre une valeur de sortie correspondante. Comme les valeurs entrée/sortie sont numériques, alors le système flou doit être doté de deux interfaces de conversion appelées fuzzificateur et défuzzificateur, d'où la structure générale d'un contrôleur flou (cf. figure 2.3) [11] [12] [13].

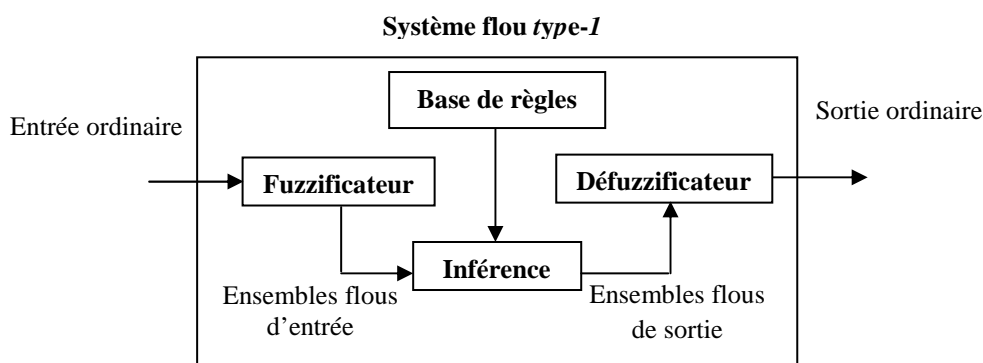


Fig.2.3 - Structure générale d'un contrôleur flou *type-1*.

-fuzzification : elle transforme toute variable numérique d'entrée en variable floue. Cependant, dans notre travail, on utilise que la fuzzification singleton (i.e, la fonction

d'appartenance est égale à 1 quant son argument correspond à l'entrée numérique et nulle ailleurs).

-*base de règles* : la forme générale d'une règle dans ce modèle est :

$$R^{(l)} : \text{Si } x_1 \text{ est } A_1^l \text{ et } x_2 \text{ est } A_2^l \text{ et } \dots \text{ et } x_p \text{ est } A_p^l \text{ alors } y \text{ est } B^l \quad (2.6)$$

où $A_i^l \subseteq X \subset \mathfrak{R}$ est la variable antécédent représentant la variable d'entrée du système flou. Similairement, $B^l \subseteq Y \subset \mathfrak{R}$ est la variable conséquence représentant la variable de sortie du système flou. A_i^l et B^l sont des termes linguistiques définis par les ensembles multidimensionnelles $\mu_{A_i^l}(x) : X \rightarrow [0 \ 1]$ et $\mu_{B^l}(y) : Y \rightarrow [0 \ 1]$, respectivement. Si tous les ensembles des conséquences possèdent seulement une seule valeur non nulle, dans ce cas, on parle d'un modèle linguistique *singleton*, dans le cas contraire, le modèle est dit *non singleton*.

-*Mécanisme d'inférence* : consiste à calculer les valeurs de vérité des différentes règles et à associer à chacune d'elle une valeur de sortie qui dépend de sa partie conclusion. L'utilisation des opérateurs *et/ou* et l'implication dans les règles floues, tels qu'ils sont définis par les fonctions élémentaires (*min, max, produit...*), et la combinaison de ces différentes fonctions conduit à plusieurs méthodes d'inférence.

-*défuzzification* : La défuzzification consiste à calculer, à partir de l'ensemble flou de sortie, la valeur numérique la plus représentative de ce dernier. Plusieurs méthodes sont proposées dans la littérature [11] [13]. Dans notre cas, on s'est limité à deux méthodes: le centre de gravité (*CG*) et celle des centres d'ensembles (*COS*).

- Méthode du centre de gravité

C'est la méthode de défuzzification la plus utilisée. Elle est exprimée par :

$$y_{CG} = \frac{\sum_{i=1}^N y_i \mu_B(y_i)}{\sum_{i=1}^N \mu_B(y_i)} \quad (2.7)$$

où B est l'ensemble de sortie discrétisé en N points (y_1, \dots, y_N) .

- Méthode des centres d'ensembles

Chaque ensemble de sortie d'une règle est remplacé par un singleton représentant son centroïde. La sortie numérique est donnée par la relation suivante :

$$y_{cos} = \frac{\sum_{l=1}^M c^l \mathfrak{S}_{i=1}^p \mu_{A_i^l}(y_i)}{\sum_{i=1}^N \mathfrak{S}_{i=1}^p \mu_{A_i^l}(y_i)} \quad (2.8)$$

où \mathfrak{S} indique la t-norme choisie, et c^l est le centroïde du $l^{i\text{ème}}$ ensemble flou de sortie.

Exemple 1: on considère un système flou type *Mamdani* à deux variables d'entrée (X_1, X_2) ayant deux valeurs linguistiques pour chacune d'elle : (A_1, A_2) et (B_1, B_2) respectivement ; une variable de sortie (Y) répartie en trois ensemble flous (C_1, C_2, C_3) . Le système intègre la base de règles suivantes :

- Si X_1 est A_1 et X_2 est B_1 alors Y est C_1
- Si X_1 est A_1 et X_2 est B_2 alors Y est C_2
- Si X_1 est A_2 et X_2 est B_2 alors Y est C_3

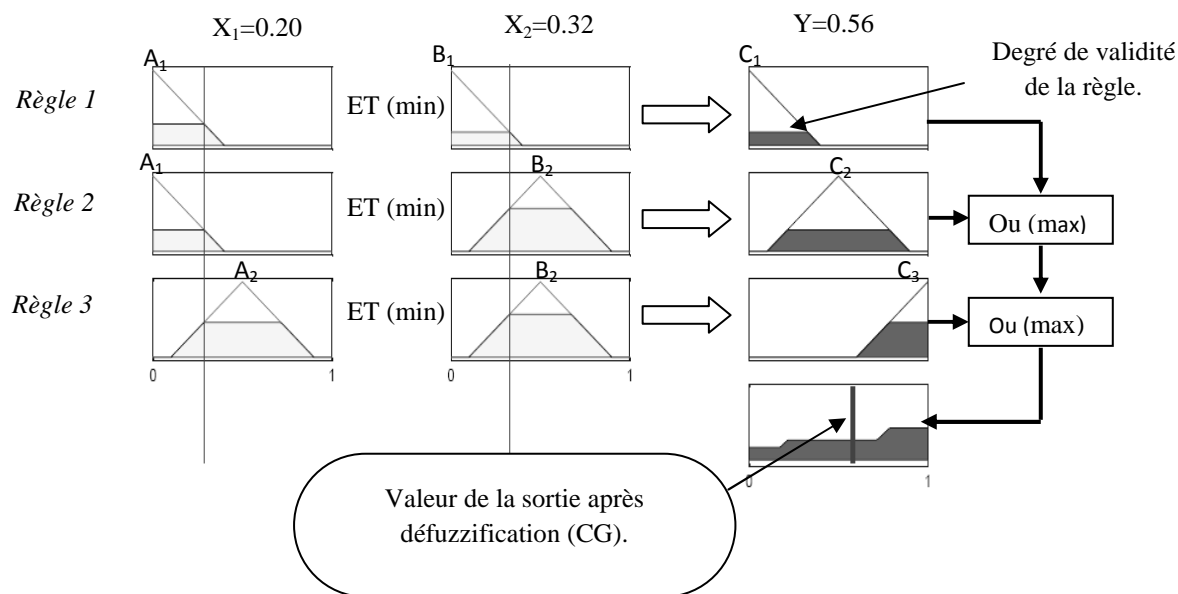


Fig.2.4 - Les étapes de calcul de la sortie d'un contrôleur flou de type *Mamdani*.

2.4.2 Modèle flou type *Takagi-Sugeno-Kang* (*TSK*)

Ce type de modèles flous a été introduit en 1985 par *Takagi-Sugeno-Kang* [14]. Les conséquences des règles sont des fonctions ordinaires des entrées du modèle [8] [14]:

$$R^{(l)} : \text{Si } x_1 \text{ est } A_1^l \text{ et } \dots \text{ et } x_p \text{ est } A_p^l \text{ alors } y^l = f(x_1, \dots, x_p) \quad (2.9)$$

Les fonctions conséquences f^l sont choisies typiquement de formes paramétriques. Si toutes les conséquences du modèle deviennent des constantes, nous obtenons alors un modèle *TSK* d'ordre zéro qui peut être vu comme un modèle flou linguistique de type singleton [18].

2.5 Concept des systèmes flous *type -2*

Le concept des systèmes flous *type-2* est basé sur les ensembles flous *type-2*. Ces derniers ont été introduits par *Zadeh* en 1975 comme extension du concept des ensembles flous ordinaires appelés ensembles flous *type-1* [7]. Un ensemble flou *type-2* est caractérisé par une fonction d'appartenance floue, c'est-à-dire, 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 sur les valeurs d'appartenance elles 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 nous ne pouvons pas déterminer le degré d'appartenance d'un élément à un ensemble par 0 ou 1, on utilise les ensembles flous *type-1*. Similairement, lorsque nous ne pouvons pas déterminer les fonctions d'appartenance floues par des nombres réels dans $[0, 1]$, on utilise alors les ensembles flous *type-2*. Ainsi, les ensembles flous *type-1* peuvent être considérés comme une approximation du premier ordre de l'incertitude. Alors que les ensembles flous *type-2* seront considérés comme approximation du deuxième ordre [15].

On distingue plusieurs types d'ensembles flous *type-2*, et ce, selon la forme des fonctions d'appartenance secondaire. Dans le cas où le degré d'appartenance de chaque point appartenant à un ensemble flou est uniforme (degré d'appartenance secondaire égal à 1) on parle d'un ensemble flou *type-2 intervalle* (cf. figure 2.5.).

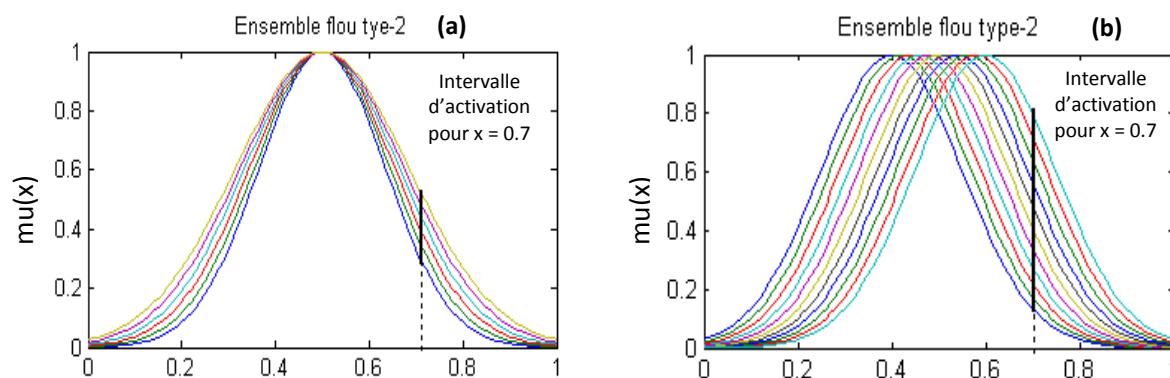


Fig.2.5 - Ensemble flou *type-2* obtenu à partir d'un ensemble flou *type-1* :

(a) variance incertaine, (b) moyenne incertaine.

2.6 Systèmes et contrôleurs flous *type-2*

Un système flou est dit de *type-2*, s'il inclut au moins un ensemble flou *type-2* dans les fonctions d'appartenance définissant ses prémisses ou ses conséquences [15]. L'inférence floue, dans ce type de système, conduit à l'obtention d'ensembles flous de sortie de *type-2*.

La structure d'un système flou *type-2* est très similaire à celle d'un système flou *type-1*. Elle comporte un bloc de fuzzification, une base de règles et un mécanisme d'inférence. La seule différence est au niveau de la sortie. Dans le cas du *type-2*, le bloc de défuzzification est précédé d'un bloc *réduction de type* (cf. figure 2.6) [15] [16][17].

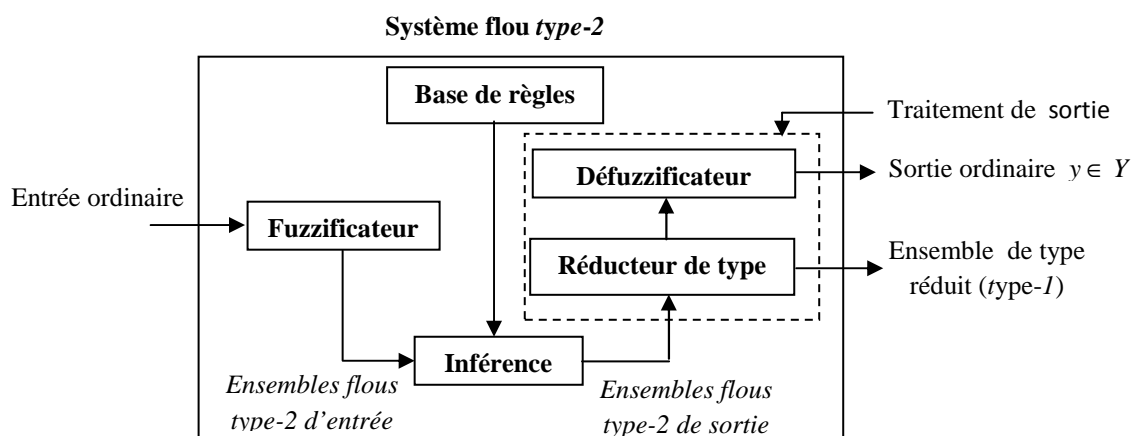


Fig.2.6 - Structure générale d'un contrôleur flou *type-2*.

2.7 Traitement de la sortie d'un système flou *type-2*

Les ensembles de sortie issus de chacune des règles d'un système flou *type-2* sont de *type-2*. Cela est dû à l'utilisation de fonctions d'appartenance de ce type dans les prémisses et/ou dans les conséquences. Donc, pour calculer la sortie defuzzifiée dans ce cas, nous devons utiliser des versions étendues, moyennant le principe d'extension de *Zadeh* (*annexe 2*), des méthodes de defuzzification *type-1*.

L'opération de defuzzification *type-1* permet d'obtenir une valeur numérique (ordinaire). Par contre, dans le cas des systèmes flous *type-2*, l'opération de defuzzification étendue permet d'obtenir, dans un premier temps, un ensemble flou de *type-1*. Cette opération qui transforme un ensemble flou *type-2* à un ensemble flou *type-1* s'appelle *réduction de type*. On appelle l'ensemble flou *type-1* résultant : *ensemble de type réduit*. Enfin, l'ensemble flou de *type réduit* doit être defuzzifié pour obtenir un nombre ordinaire [15] [18] [19].

2.8 Réduction de type des ensembles flous *type-2* intervalle [15]

Un système flou *type-2* est très compliqué à cause de la réduction de type. L'utilisation d'ensembles flous continus ou niveau des prémisses et/ou conséquences de ces systèmes conduit à l'obtention d'ensembles continus au niveau des sorties. Cependant, ces ensembles doivent être discrétisés pour des raisons d'implémentation sur des calculateurs (cf. figure 2.7). La formule générale du calcul de l'ensemble de *type réduit* (ensemble de *type-1*) est exprimée par (2.10)

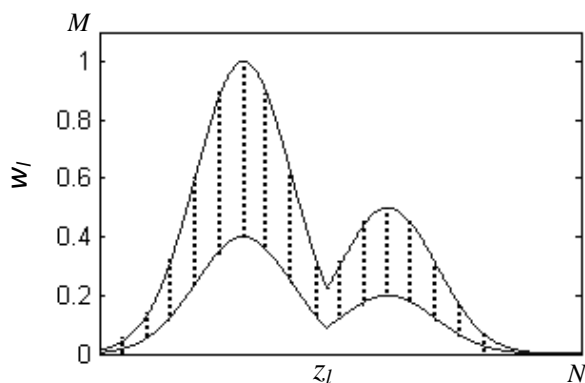


Fig. 2.7- Exemple d'un ensemble flou *type-2* intervalle discrétisé pour des raisons de réduction de type.

$$Y(Z_1, \dots, Z_M, W_1, \dots, W_M) = \int_{z_1} \dots \int_{z_M} \int_{w_1} \dots \int_{w_M} \mathfrak{S}_{l=1}^M \mu_{z_l}(z_l) \wedge \mathfrak{S}_{l=1}^M \mu_{w_l}(w_l) \left/ \frac{\sum_{l=1}^M w_l z_l}{\sum_{l=1}^M w_l} \right. \quad (2.10)$$

où \mathfrak{S} et \wedge indiquent les t -normes (*produit* ou *minimum*), $w_l \in W_l$ et $z_l \in Z_l$ pour $l=1, 2, \dots, M$. Les choses se simplifient beaucoup lorsque les fonctions d'appartenance secondaires sont de type intervalle. Dans notre travail, on s'est intéressé à la réduction de type des systèmes flous *type-2* intervalle. Dans ce qui suit, nous présentons une procédure algorithmique de calcul de l'ensemble type réduit de ce genre de systèmes où on remarquera que les combinaisons de z_l et w_l ne seront pas toutes considérées.

Pour un système *type-2* intervalle, chaque z_l et w_l ($l=1, 2, \dots, M$) dans (2.10) est un ensemble *type-1* intervalle, par conséquent nous avons $\mu_{z_l}(z_l) = \mu_{w_l}(w_l) = 1$, donc (2.10) devient :

$$Y(Z_1, \dots, Z_M, W_1, \dots, W_M) = \int_{z_1} \dots \int_{z_M} \int_{w_1} \dots \int_{w_M} 1 \left/ \frac{\sum_{l=1}^M w_l z_l}{\sum_{l=1}^M w_l} \right. = [y_l, y_r] \quad (2.11)$$

Cela signifie que le calcul de l'ensemble type réduit se limite juste au calcul du quotient :

$\frac{\sum_{l=1}^M w_l z_l}{\sum_{l=1}^M w_l}$. Comme tous les degrés d'appartenance dans un ensemble *type-1* intervalle sont

unitaires, nous allons représenter ces ensembles par leurs limites gauches et droites $[l, r]$ ou

par leurs centres et largeurs : $c = \frac{l+r}{2}$ et $s = \frac{r-l}{2}$ respectivement.

Dans un système flou *type-2*, chaque z_l dans (2.11) est un ensemble *type-1* intervalle de centre c_l et de largeur s_l ($s_l \geq 0$) et chaque w_l et un ensemble *type-1* intervalle de centre h_l ($l=1, \dots, M$). Y est aussi un ensemble *type-1* intervalle. Donc, nous avons seulement besoin de calculer les deux points extrêmes y_l et y_r . y_l dépend seulement de $(c_l - s_l)$ et sur

l'un des deux points extrêmes de W_l . y_r dépend seulement $c_l + s_l$ et sur l'un des deux points extrêmes de W_l . Dans ce qui suit, nous allons citer une procédure itérative pour calculer y_l et y_r .

$$\text{Posons : } S(W_1, \dots, W_M) \cong \frac{\sum_{l=1}^M z_l w_l}{\sum_{l=1}^M w_l} \quad (2.12)$$

où $w_l \in [h_l - \Delta_l, h_l + \Delta_l]$ et $h_l \geq \Delta_l$ pour $(l=1, 2, \dots, M)$ et $z_l \in [c_l - s_l, c_l + s_l]$. Le maximum de S, y_r est obtenu comme suit. On pose $z_l = c_l + s_l$ ($l=1, 2, \dots, M$) et en supposant que les z_l sont arrangés dans l'ordre croissant, c-a-d, $z_1 \leq z_2 \leq \dots \leq z_M$. Alors :

1. posons $w_l = h_l$ pour $(l=1, 2, \dots, M)$ et calculons $S' = S(h_1, \dots, h_M)$ en utilisant (2.12).
2. trouver K ($1 \leq K \leq (M-1)$) tel que $z_k \leq S' \leq z_{k+1}$
3. poser $w_l = h_l - \Delta_l$ pour $l \leq K$ et $w_l = h_l + \Delta_l$ pour $l \geq K+1$ et calculer $S'' = S(h_1 - \Delta_1, \dots, h_k - \Delta_k, h_{k+1} + \Delta_{k+1}, \dots, h_M + \Delta_M)$ en utilisant (2.12).
4. vérifier si $S'' = S'$; si oui, arrêter. S'' est la valeur maximale de $S(W_1, \dots, W_M)$; si non, aller à l'étape 5.
5. poser $S'' = S'$; aller à l'étape 2.

Il a été démontré dans que cette procédure itérative converge dans M itérations au maximum, où une itération est le passage de l'étape 2 à 5.

Le minimum de $S(W_1, \dots, W_M)$ peut être obtenu en utilisant la même procédure juste citée. Seulement deux changements doivent être faits :

- ✓ nous devons mettre $z_l = c_l - s_l$ pour $l=1, 2, \dots, M$.
- ✓ dans l'étape 3 nous devons poser $w_l = h_l + \Delta_l$ pour $l \leq K$ et $w_l = h_l - \Delta_l$ pour $l \geq K+1$ et calculer $S'' = S(h_1 + \Delta_1, \dots, h_k + \Delta_k, h_{k+1} - \Delta_{k+1}, \dots, h_M - \Delta_M)$.

2.8.1 Réduction de type par la méthode du centroïde

Nous discrétisons l'ensemble de sortie caractérisé par $\mu_{\tilde{B}}(y)$ en N points, y_1, \dots, y_N . Soit $[L_i, R_i]$ le domaine de $\mu_{\tilde{B}}(y_i)$. Pour utiliser la procédure itérative, notons que la somme dans (2.12) est prise de 1 à N au lieu de 1 à M ; et comme les y_i sont ordinaires; $(L_i + R_i)/2 = h_i$ et $(R_i - L_i)/2 = \Delta_i$.

2.8.2 Réduction de type par la méthode des centres d'ensembles

Dans cette méthode, la procédure itérative s'applique en deux étapes. Dans la première étape, nous calculons les centroïdes C_i des ensembles *type-2* intervalle des conséquences. Dans la deuxième, nous calculons l'ensemble de type réduit, C_i va jouer le rôle des Z_i dans (2.11). Si le domaine de C_i est l'intervalle $[L_i^c, R_i^c]$, alors $c_i = (L_i^c + R_i^c)/2$ et $s_i = (R_i^c - L_i^c)/2$. Le degré d'activation E_i joue le rôle de W_i . Si le domaine E_i est l'intervalle $[L_i, R_i]$, alors $h_i = (L_i + R_i)/2$ et $\Delta_i = (R_i - L_i)/2$.

Dans chacun des cas précédents, la sortie ordinaire du système flou est obtenue en calculant le point milieu de l'ensemble de type réduit du fait qu'il est un ensemble *type-1* intervalle [15].

Exemple 2 : Dans cet exemple, nous allons appliquer la procédure de réduction de type du (§. 2.8) sur des ensembles flous *type-2* intervalle résultants d'une inférence (*max-produit*) dans un système flou *type-2* ayant deux variables d'entrée x_1 et x_2 et une seule variable de sortie y dans l'intervalle $[0, 20]$ (figure 2.8.a, b et c). Tous les ensembles prémisses et conséquences sont de *type-2* intervalle obtenus à partir de gaussiennes à variance fixe et moyenne incertaine (cf. figure 2.5.b). Les paramètres de ces ensembles sont donnés par le tableau (2.1).

	<i>Prémises</i>						<i>Conséquences</i>		
	<i>Variable (x₁)</i>			<i>Variable (x₂)</i>					
	\tilde{A}_1^1	\tilde{A}_1^2	\tilde{A}_1^3	\tilde{A}_2^1	\tilde{A}_2^2	\tilde{A}_2^3	\tilde{B}^1	\tilde{B}^2	\tilde{B}^3
m_1	3	9.5	15	4	10	16	3	9.75	16
m_2	4.5	10	16	5	11	17	4	10.25	17
σ	1.5	2.5	1.5	2	1.5	2	2.5	1.5	2.5

Tab.2.1 - Paramètres des fonctions d'appartenance de l'exemple 2.

L'entrée appliquée au système flou ainsi définie est $X = [6 \ 2]^T$. Elle a deux appartenances non nulles \tilde{A}_1^1 et \tilde{A}_1^2 pour x_1 et une autre pour x_2 \tilde{A}_2^1 (cf. figure 2.8.a et b), par conséquent, les deux règles suivantes sont activées.

$R^{(1)}$: Si x_1 est \tilde{A}_1^1 et x_2 est \tilde{A}_2^1 alors y est \tilde{B}^1 ou

$R^{(2)}$: Si x_1 est \tilde{A}_1^2 et x_2 est \tilde{A}_2^1 alors y est \tilde{B}^1

En (d), on représente l'ensemble de type réduit ($[y_l, y_r] = [3.0008, 4.0156]$) ainsi que la sortie defuzzifiée (numérique) ($y_{cos} = 3.5082$) où nous avons pris 2000 échantillons.

2.9 Conclusion

Ce chapitre est une étude détaillée des systèmes flous *type-1* et *type-2*. Parmi les points que nous avons traité on peut citer : la notion d'ensembles flous, les opérations et les normes dans ces systèmes, la structure des systèmes et contrôleurs flous, et surtout, d'une manière plus détaillée, le traitement de la sortie dans le cas des systèmes flous *type-2* qui exige d'abord la réduction de type puis la defuzzification. Pour ne pas rester dans le cadre descriptif, nous avons fait état de deux exemples d'application.

Dans le chapitre suivant, nous allons appliquer les contrôleurs flous *type-1* et *type-2* pour contrôler le mouvement d'un robot mobile.

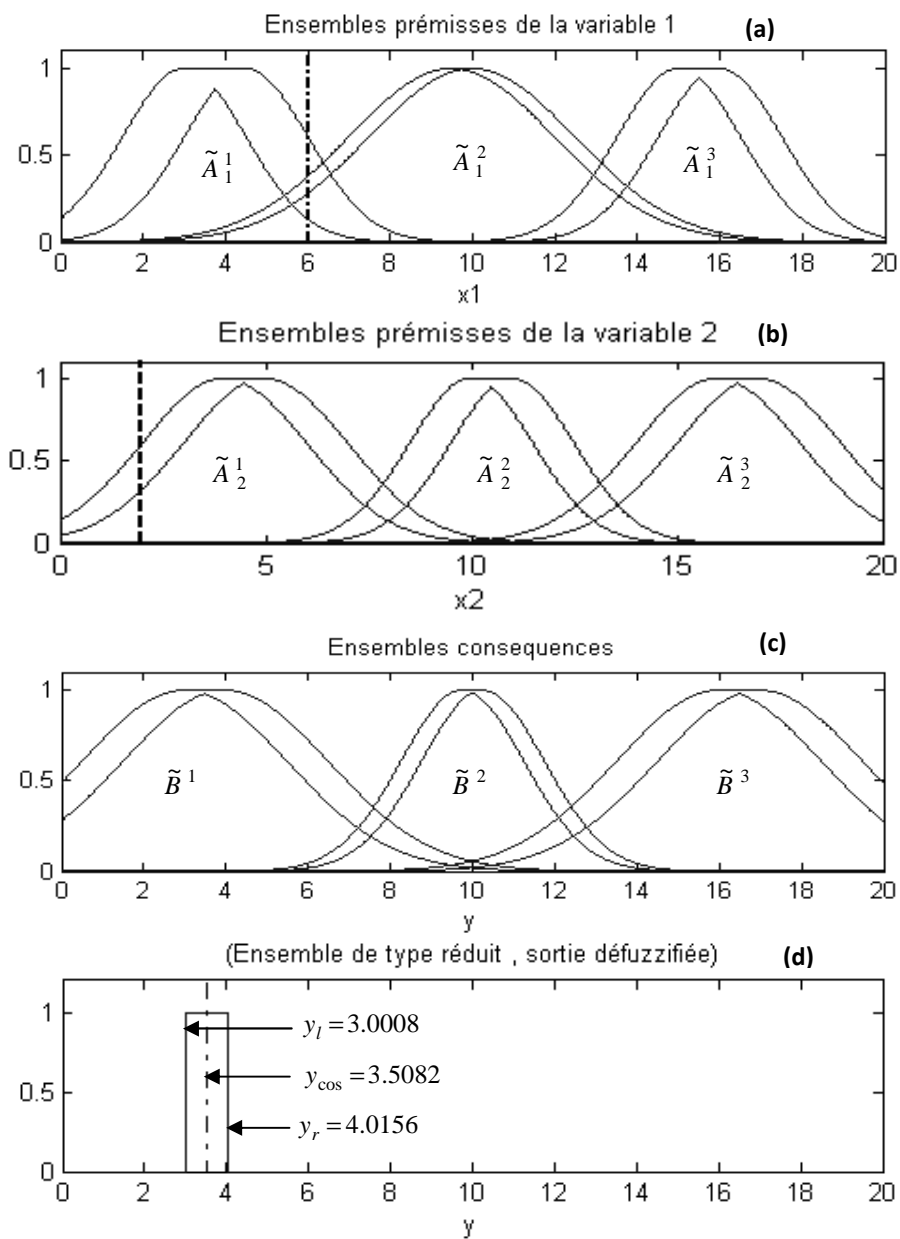


Fig.2.8- (a) et (b) : Ensembles flous antécédents, (c) : Ensembles flous conséquences, (d) : Ensemble type réduit de l'exemple 2.

Chapitre 3

Navigation floue d'un seul robot mobile

Résumé : Ce chapitre traite le problème de navigation autonome d'une seule entité robotique mobile en utilisant la logique floue. Dans un premier temps, l'environnement considéré est sans obstacles. Dans ce cas de figure, faire déplacer le robot d'une position initiale vers une position désirée (cible) nécessite l'introduction d'un seul module d'action (contrôleur d'attraction vers la cible). Par la suite, on abordera le même problème dans un environnement proche de la réalité, i.e : des objets gênant le mouvement du robot sont présents (environnement en présence d'obstacles). Cela nécessite l'introduction d'un autre contrôleur dit d'évitement d'obstacles. De plus, un mécanisme de coordination entre les consignes issues de ces deux contrôleurs s'impose. La robustesse de la structure de contrôle ainsi définie est testée en simulant la présence d'incertitudes (bruit) de mesure moyennant les contrôleurs flous type-2.

3.1. Introduction

Depuis quelques années, un intérêt croissant est porté au sein de la communauté robotique au développement de systèmes intelligents autonomes dans le cadre de la robotique mobile. Un tel intérêt peut être perçu comme une conséquence logique à l'apparition d'applications potentielles (nettoyage, manutention ou encore de l'aide à la mobilité de personnes âgées ou handicapées) et au désir de mettre les robots sur des tâches nouvelles telles que les opérations d'intervention sur sites accidentés, la manipulation sur sites sensibles en l'occurrence nucléaires, ou l'exploration de sites maritimes ou planétaires. Comme toute commande de systèmes, la représentation de la dynamique du robot par un modèle mathématique s'avère plus que nécessaire.

3.2 Modélisation

Nous représentons par la figure (3.1) un robot mobile en vue d'une modélisation cinématique. En effet, un robot mobile peut être considéré comme étant un véhicule à roues autonome et capable d'exécuter des missions dans des environnements connus ou inconnus [20]. Ce type de robot mobile possède deux roues différentielles montées sur un même axe (Y_m).

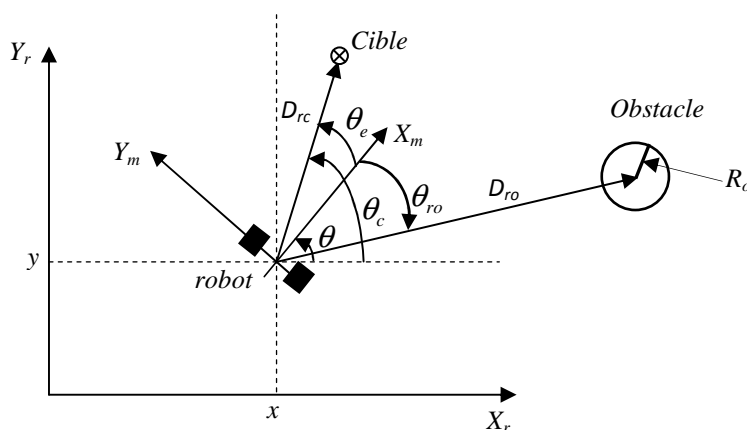


Fig.3.1 - Représentation graphique d'un robot mobile à roues différentielles

d'où le modèle cinématique suivant [1] [4] [5]:

$$\dot{q} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.1)$$

où $q = [x, y, \theta]^T$ est le vecteur des coordonnées généralisées qui décrit la position du robot, (x, y) sont les coordonnées cartésiennes du centre de l'axe des roues, θ est l'angle entre les deux axes (X_m, X_r) , $[v, \omega]^T$ est le vecteur des vitesses linéaire et angulaire, respectivement. La contrainte de la non holonomie est exprimée par l'équation (3.2) qui signifie que le robot mobile ne peut se déplacer instantanément dans la direction perpendiculaire à l'axe (X_m) .

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (3.2)$$

Le modèle (3.1) peut être réécrit comme suit :

$$\begin{cases} \dot{x} = \dot{v} \cdot \cos \theta \\ \dot{y} = \dot{v} \cdot \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (3.3)$$

où : v_x et v_y sont les projections de la vitesse linéaire sur l'axe (X_r) et (Y_r) , respectivement.

3.3 Attraction vers une cible dans un environnement sans obstacles

Ici, l'objectif recherché est d'assurer l'atteinte de la cible en empruntant la trajectoire la plus courte. A cet effet, nous allons proposer un contrôleur flou qui exploitera le modèle du robot ainsi que les informations (position) de l'objectif à atteindre. La structure de commande (cf. figure 3.2) est adoptée et son schéma sous *Matlab/Simulink* est représenté par (cf. figure 3.3) [12][13][20].

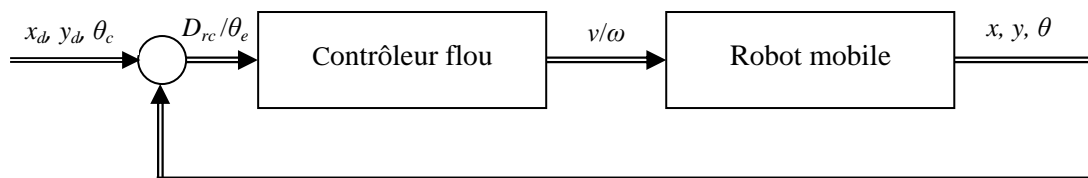


Fig.3.2 -Structure de commande : attraction vers la cible.

Avant de procéder à la conception du contrôleur de convergence vers la cible et à la simulation de l'architecture de contrôle sous *Matlab*, nous calculons d'abord les vitesses maximales admissibles par les robots *Khepera III*.

3.3.1 Conception du contrôleur d'attraction vers la cible

Le contrôleur flou possède deux entrées : distance robot-cible (D_{rc}) et l'angle robot-cible (θ_e), (cf. figure 3.1), deux sorties : la vitesse linéaire (v) et la vitesse angulaire (ω) qui serviront comme des grandeurs de commande pour le robot mobile. Les performances du système sont améliorées en ajustant, d'une manière heuristique, les paramètres des fonctions d'appartenance des prémisses et des conséquences du contrôleur flou en question.

Pour la première variable d'entrée, la distance (D_{rc}), nous définissons trois fonctions d'appartenance caractérisant les valeurs linguistiques (P : petit, M : moyenne, G : grande) et cinq ensembles flous pour la seconde variable, l'erreur d'orientation (θ_e), (NG : négative grande, NM : négative moyenne, N : nulle, PM : positive moyenne, PG : positive grande), sur les univers de discours $[0, 30]$ et $[-\pi, \pi]$, respectivement. Les valeurs linguistiques de la première variable de sortie, vitesse linéaire (v), sont (N : nulle, M : moyenne, G : grande). Pour la seconde variable de sortie, vitesse angulaire (ω), on a (GG : gauche grande, GM : gauche moyenne, D : direct, DM : droite moyenne, DG : droite grande) (cf. figure 3.3).

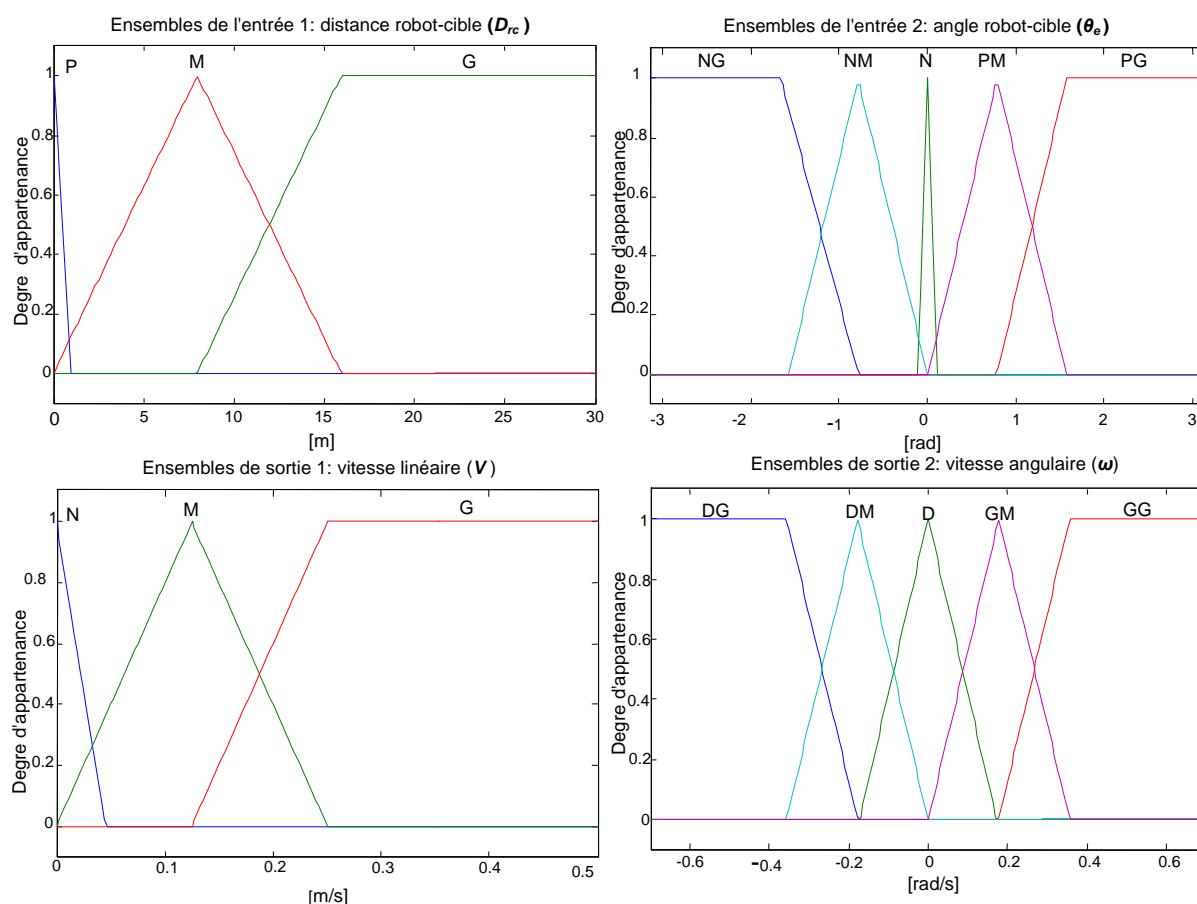


Fig.3.3 - Les ensembles prémisses et conséquences du contrôleur d'attraction vers la cible.

Le tableau (Tab 1) montre la table de décision du contrôleur flou considéré, la méthode d'inférence est le (*max-min*), à titre d'exemple, la 8^{ième} règle (*ligne 2, colonne 3*) est exprimée comme suit :

$$R^{(8)} : \text{Si } D_{rc} \text{ est } M \text{ et } \theta_e \text{ est } N \text{ alors } V \text{ est } G \text{ et } \omega \text{ est } D \tag{3.4}$$

(V, ω)

$\theta_e \backslash D_{rc}$	NG	NM	N	PM	PG
P	N/GG	N/GM	N/D	N/DM	N/DG
M	N/GG	M/GG	G/D	M/DG	N/DG
G	N/GG	M/GG	G/D	M/DG	N/DG

Tab.1. Table de décision du contrôleur de convergence vers la cible

3.3.2 Résultats et commentaires

L'architecture de la figure (3.4) ci-dessous a été simulée.

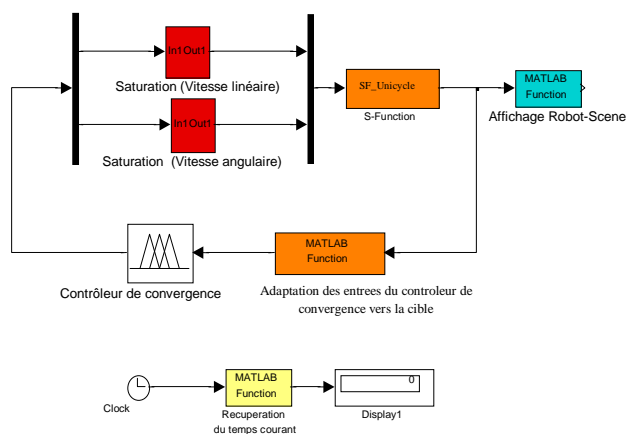


Fig 3.4 - Schéma sous matlab/simulink : attraction vers la cible

A partir de la figure (3.5.a), on constate, et pour trois positions distinctes du robot mobile : $X_{I1} = 10, Y_{I2} = 23, \theta_{I1} = -90^\circ$; $X_{I2} = 5, Y_{I2} = 20, \theta_{I2} = 0^\circ$ et $X_{I3} = 15, Y_{I3} = 17, \theta_{I3} = 180^\circ$; ce dernier a pu atteindre la cible dont les coordonnées ($X_c = 25, Y_c = 25$). Des résultats obtenus on remarque que le robot mobile adopte le comportement suivant : lorsque

l'angle robot-cible (D_{rc}) et grande, la vitesse angulaire est grande par contre la vitesse linéaire est petite. Une fois que l'angle robot-cible devient nulle la vitesse linéaire atteint sont maximum (figure c, d, e). Cette dernière diminue progressivement en s'annulant une fois la cible est atteinte (figure b,d).

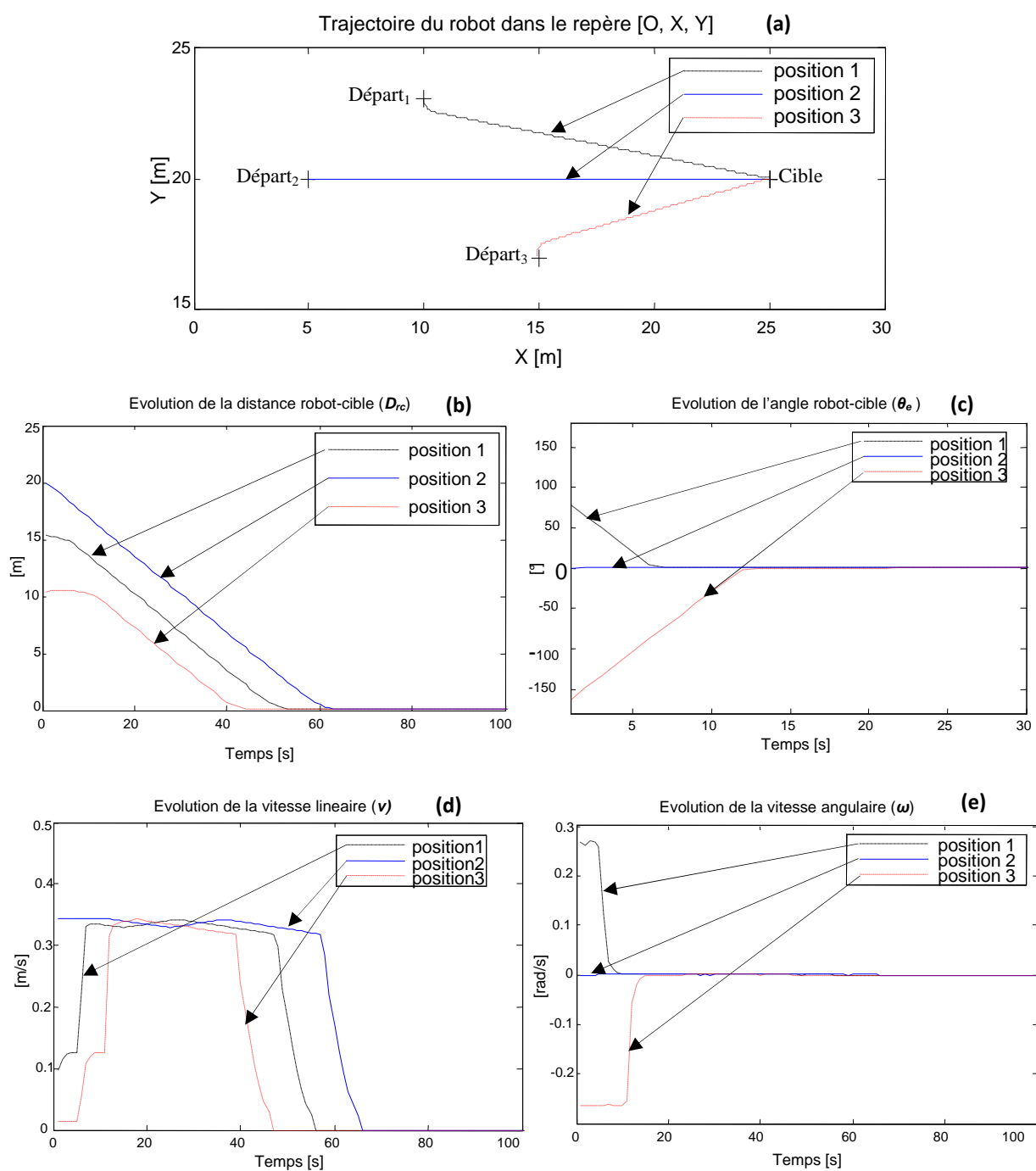


Fig.3.5- Attraction vers la cible : environnement sans obstacles

3.4. Comportement d'évitement d'obstacles

Nous allons mettre en œuvre dans ce qui suit le contrôleur d'évitement d'obstacles bien que un tel comportement employé seul n'a pas un intérêt pratique du fait que le robot n'a pas un objectif spécifique (aller vers une cible, maintenir une formation, ..etc).

Le contrôleur d'évitement d'obstacles qu'on a utilisé possède deux variables d'entrées : distance et l'angle robot-obstacle (D_{ro} et θ_{ro}) respectivement, (cf. figure 3.1), deux variables de sortie : vitesse linéaire et angulaire. La répartition des univers de discours de ces variables sont représentés ci-dessous (cf. figure 3.6).

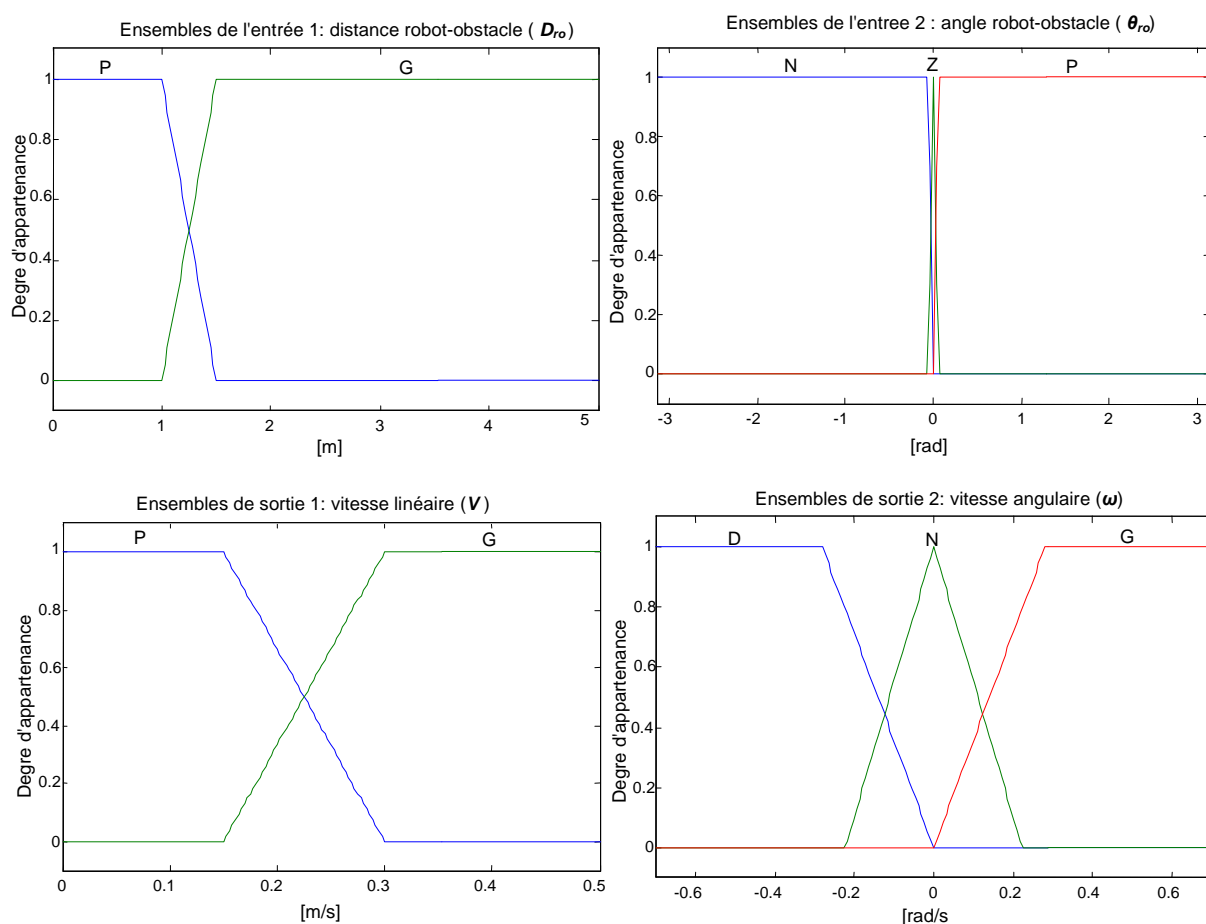


Fig.3.6- Les fonctions d'appartenance du contrôleur d'évitement d'obstacles

La base de règles de ce contrôleur est générée suivant le tableau (Tab 2).

(V, ω)

θ_{ro}	N	Z	P
D_{ro}			
P	<i>P/G</i>	<i>P/D</i>	<i>P/D</i>
G	<i>G/N</i>	<i>G/N</i>	<i>G/N</i>

Tab.2. Table de décision du contrôleur d'évitement d'obstacles

Résultats et commentaires :

La figure (3.7) montre que le contrôleur d'évitement d'obstacles employé seul garantit au robot mobile une navigation sans collision avec les obstacles présents dans son environnement, néanmoins, ce dernier ne garantit pas la convergence vers la cible. Dans ce cas, le robot mobile avance tout droit, si un obstacle est rencontré, le robot fait un contournement (évitement), une fois le robot est sorti du cercle de cet obstacle (cf. figure 3.9), la trajectoire du robot devient de nouveau une droite et ainsi de suite.

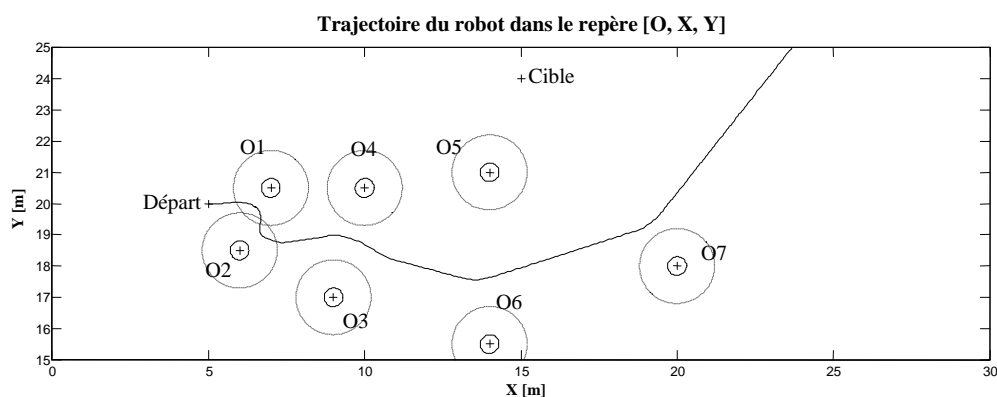


Fig 3.7- Trajectoire du robot muni du contrôleur d'évitement d'obstacle employé seul

3.5 Attraction vers une cible en présence d'obstacles (coordination de comportements) :

En section 3.3, les résultats obtenus pour un environnement sans obstacles sont assez satisfaisants. Néanmoins, pour une approche non loin de la réalité, des objets (obstacles) gênant la progression du robot peuvent être présents. Dans ce cas de figure, le contrôleur de convergence vers la cible ou le contrôleur d'évitement d'obstacles, employés chacun seul, ne peuvent garantir au robot une convergence vers la cible sans collision avec ces obstacles. D'où la nécessité de coordonner (fusionner) les actions de ces contrôleurs. (cf. figure 3.8).

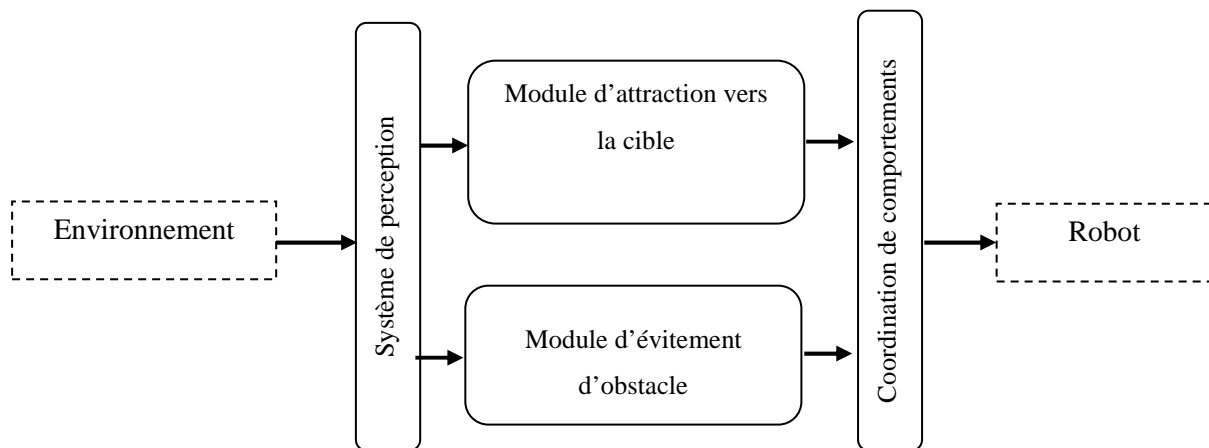


Fig.3.8 - Schéma de contrôle du robot

3.5.1 Sélection de comportements par un switch (hard switch)

Dans ce cas les deux contrôleurs flous (attraction vers la cible et évitement d'obstacles) fonctionnent d'une manière alternative, selon l'algorithme suivant [21][22][23] :

Si ($D_{ro} < D_s$) **alors**

le contrôleur d'évitement d'obstacles est activé,

Sinon

le contrôleur de convergence vers la cible est activé

fin

avec : $D_s = R_r + R_o + Marge$ (cf. figure 3.9).

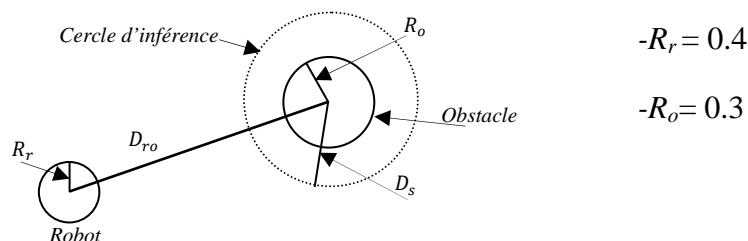


Fig.3.9 Représentation robot/obstacle

3.5.2 Fusion de comportements avec un gain flou

Dans ce cas les deux contrôleurs flous (convergence vers la cible et évitement d'obstacles) fonctionnent simultanément mais d'une manière complémentaire, et ce, en attribuant un gain « $g < 1$ » pour les sorties du contrôleur d'évitement d'obstacles et un gain de « $1 - g$ » pour celles du contrôleur d'attraction vers la cible, à chaque fois que la distance robot-obstacle D_{ro} (l'obstacle le plus proche du robot) est inférieure à la distance de sécurité D_s .

$$V = V_{eo} * g + V_{ac} * (1 - g) \quad (3.5)$$

$$\omega = \omega_{eo} * g + \omega_{ac} * (1 - g)$$

V_{eo} : la vitesse linéaire générée par le contrôleur d'évitement d'obstacles,

V_{ac} : la vitesse linéaire générée par le contrôleur d'attraction vers la cible,

ω_{eo} : la vitesse angulaire générée par le contrôleur d'évitement d'obstacles,

ω_{ac} : la vitesse angulaire générée par le contrôleur d'attraction vers la cible

g : gain généré par le contrôleur flou (cf. figure 3.10).

Les ensembles d'entrées/sortie du bloc de fusion de comportements sont représentés par la figure (3.10), et la table de décision est donnée par le tableau (Tab 3).

Gain

θ_{ro} \ / D_{ro}	N	Z	P
P	<i>M</i>	<i>G</i>	<i>M</i>
G	<i>P</i>	<i>P</i>	<i>P</i>

Tab.3.3 -Table de décision du bloc de fusion de contrôleurs

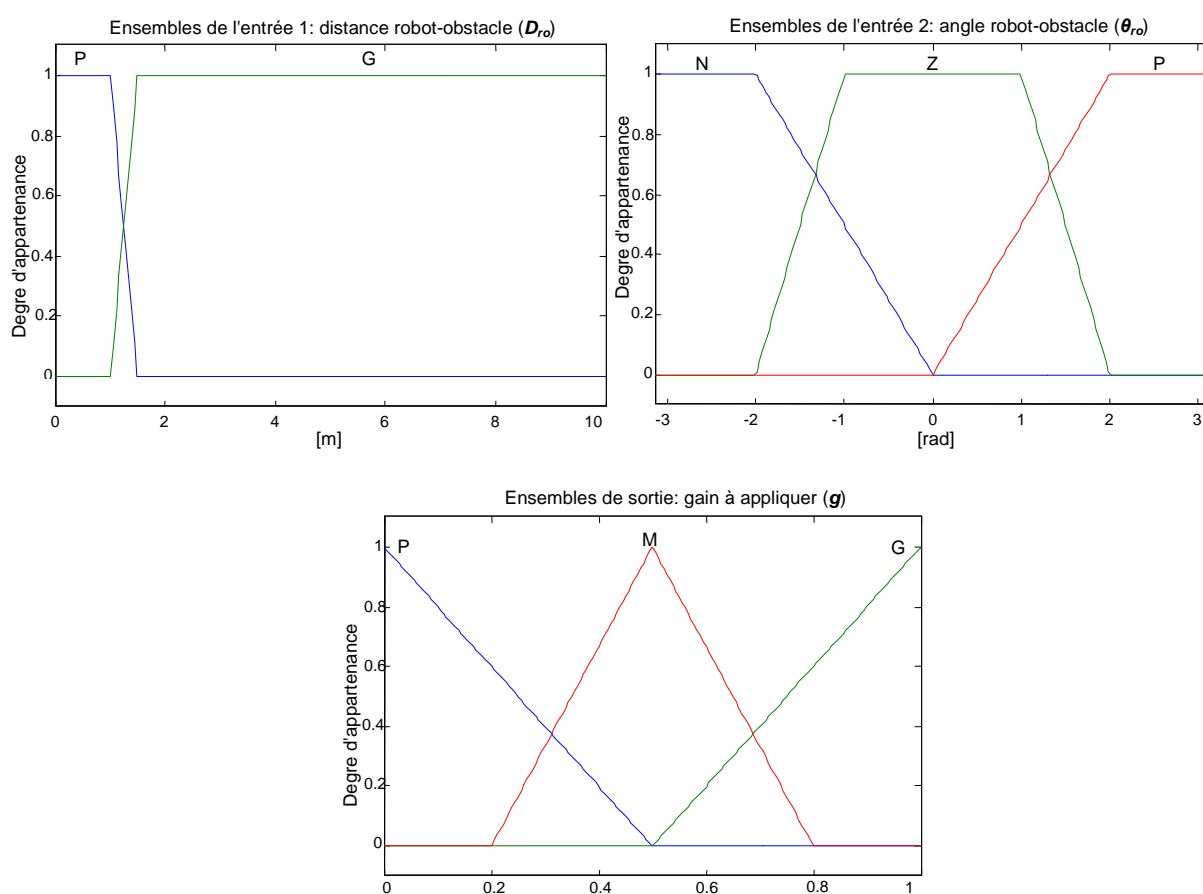


Fig.3.10- Ensembles d'entrées/sortie du bloc de fusion de comportements

Résultats et commentaires

Des résultats obtenus (cf. figure 3.11, 3.12, 3.13), on constate que les deux approches de coordination de comportements (sélection d'action par un switch et fusion par un gain flou) garanties au robot mobile d'accomplir la tâche souhaitée, à savoir : l'atteinte de la cible sans collision avec les obstacles présents dans l'environnement de navigation.

Cependant, l'approche de fusion par un gain adaptatif flou assure un contournement meilleur des obstacles par rapport au cas du switch entre les comportements ou on constate des oscillations (variations brusques) indésirables (cf. figure 3.13).

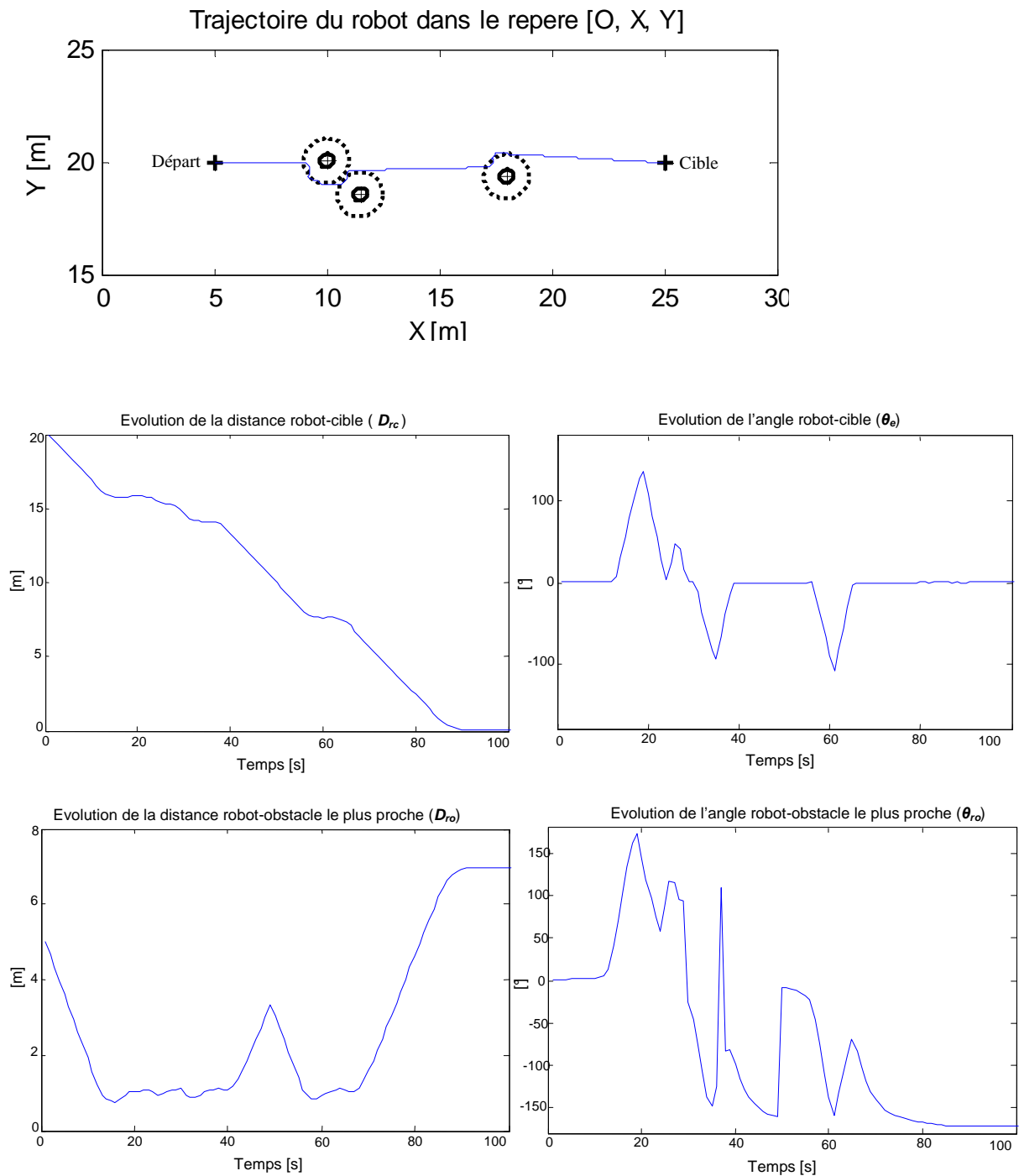


Fig.3.11- Convergence vers la cible en présence d'obstacles, coordination de comportements par un switch

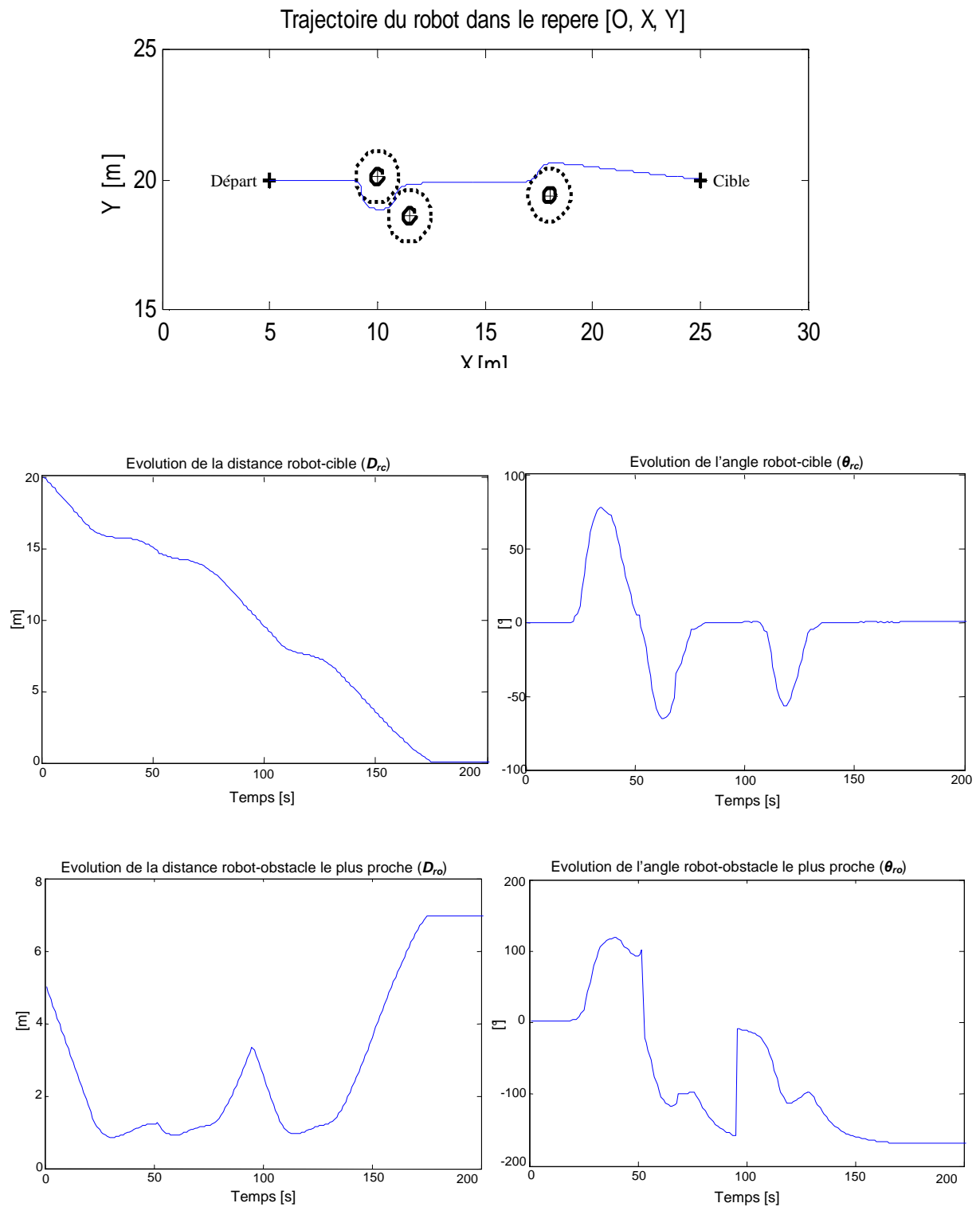


Fig.3.12-Convergence vers la cible en présence d'obstacles, coordination de comportements par fusion floue

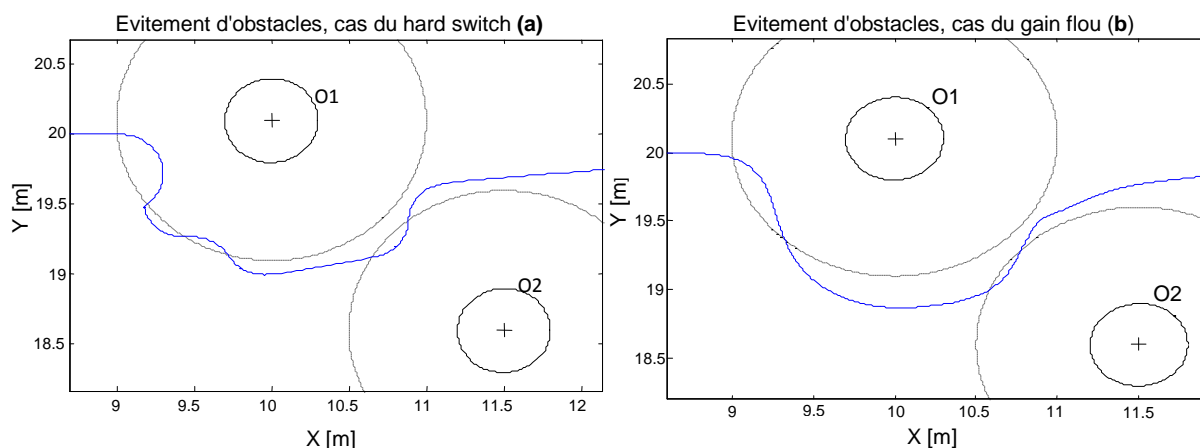


Fig.3.13 -Evitement d'obstacles : (a) cas du switch entre comportements, (b) fusion par un gain

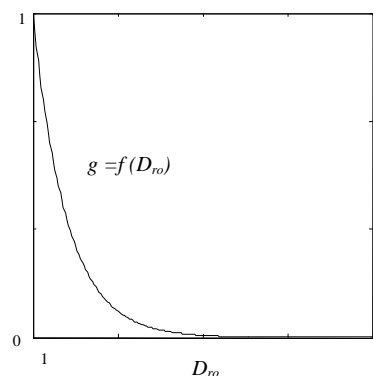
3.5.3 Coordination de comportements par une sigmoïde figée

Le but de ce test est de comparer les résultats obtenus par la fusion de comportements par un gain flou avec une fusion par une fonction sigmoïdale figée f (définie comme suit :

$$f(D_{ro}) = \exp\left(-\frac{1}{2} * \left(\frac{D_{ro}-D_s}{\sigma}\right)^2\right) ; (D_s) \leq D_{ro} \leq (2*D_s)$$

$$= 1 \quad ; \quad D_{ro} \leq D_s$$

D_{ro} : est la distance robot-obstacle et D_s : la distance de sécurité (§3.5.1).



A cet effet, on définit une fonction objectif - *temps de collision*- qui sera incrémenté à chaque fois que la distance robot-obstacles (D_{ro}) est inférieure à la distance de sécurité (D_s).

Les résultats obtenus par ce test (cf. figure 3.13) montrent, une fois de plus, que la fusion de comportements par un gain adaptatif flou est meilleure par rapport à la fusion par une fonction sigmoïdale figée (la trajectoire du robot est plus courte, contournement meilleur des obstacles, temps de collision minimal, en présence et en absence d'incertitudes de mesure). Cela est justifiable du fait que chaque sortie générée par le contrôleur flou représente

une moyenne de plusieurs valeurs, par contre dans le cas de la fusion par une sigmoïde figée, la valeur de sortie est unique.

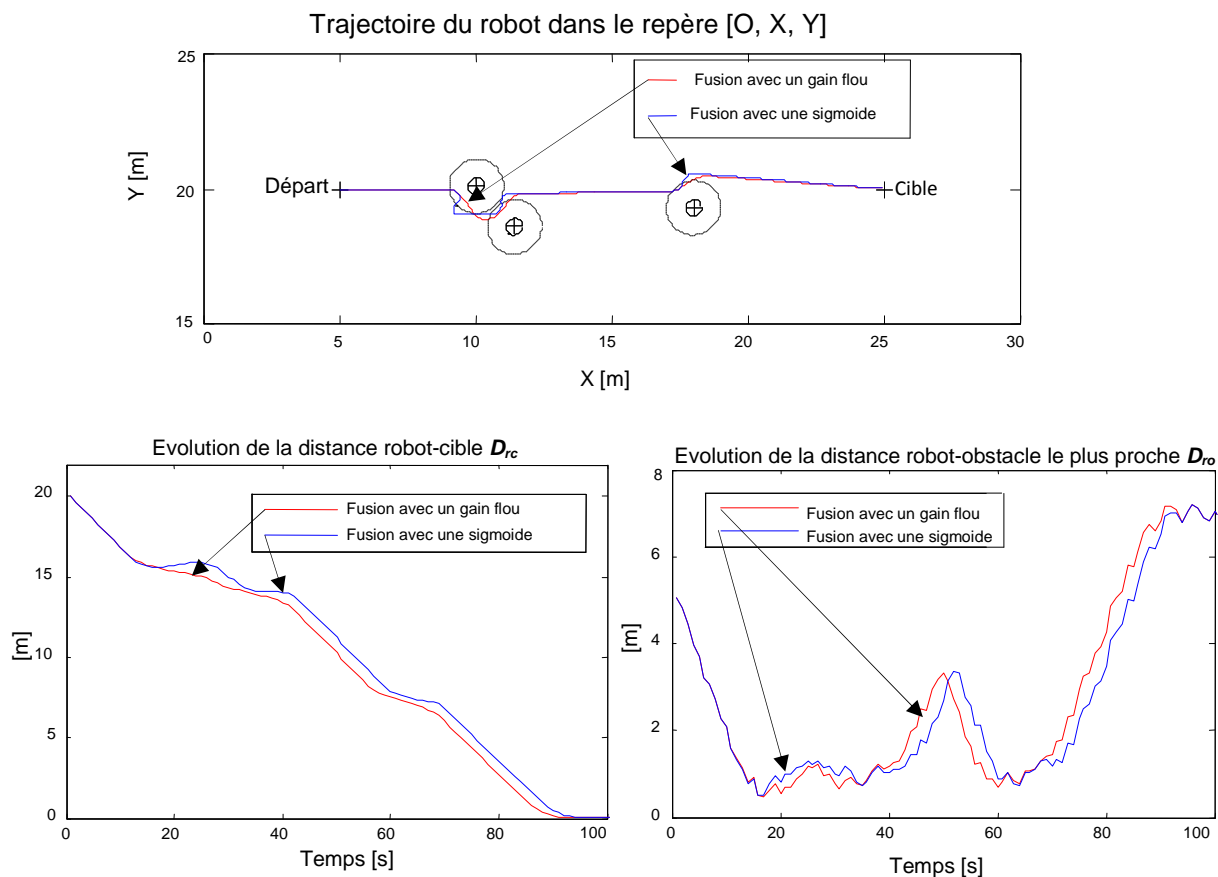


Fig.3.13. Comparaison entre la fusion de comportements par un gain flou et la fusion par une sigmoïde figée

3.6. Incorporation des incertitudes dans les contrôleurs

Une autre caractéristique d'un environnement réel est la présence d'incertitudes de mesures. Pour simuler l'effet de ces incertitudes sur l'architecture de contrôle du robot et dans le but de montrer l'efficacité des systèmes flous *type-2* pour réduire leurs effets, on considère la mesure distance robot-obstacle D_{ro} est entachée d'un bruit b gaussien, d'où :

$$D_{ro} = D_{ro} + b \quad (3.6)$$

3.6.1 Simulation des incertitudes aléatoires avec un contrôleur flou *type-2*

Un contrôleur flou de *type-2* est introduit dans l'architecture pour remplacer le module *type-1* de fusion des comportements, ces entrées/sortie sont représentées par (cf. figure 3.14) [24]. En vue de comparer les performances des deux systèmes flous *type-1* et *type-2* on a considéré une fonction objective qui calcule le temps de collision du robot avec les obstacles présents dans l'environnement de simulation. Cette fonction est évaluée à chaque itération et est incrémentée à chaque fois que la distance robot-obstacle D_{ro} est inférieure à la distance de sécurité D_s .

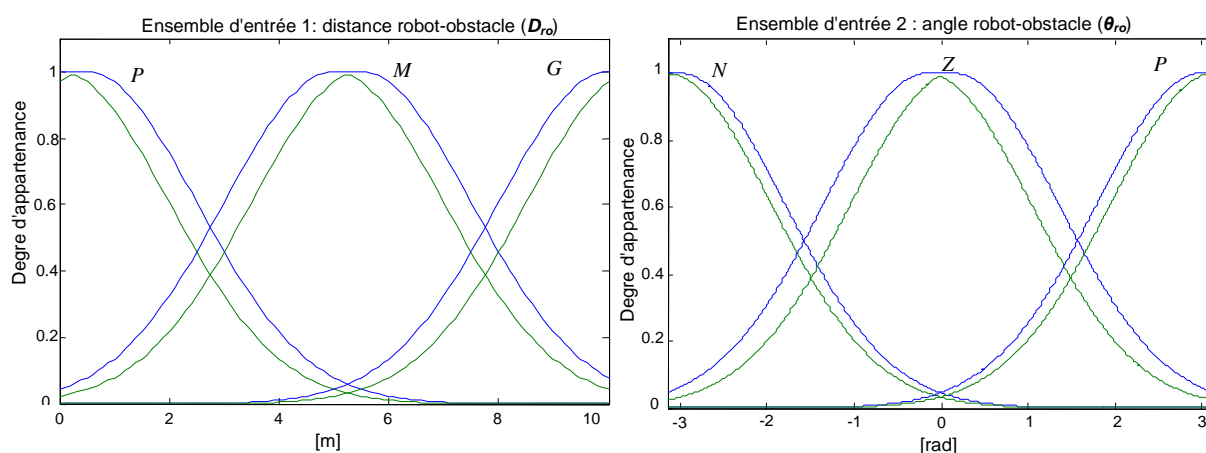


Fig.3.14- Ensembles d'entrées du contrôleur flou *type-2*

Résultats et commentaires

La figure (3.15) montre que le robot mobile munit du contrôleur de convergence vers la cible, du contrôleur d'évitement d'obstacles et du module flou de *type-2* définit ci-dessus assure une navigation sûre du robot, et ce, malgré la présence d'une source de bruit. La figure (3.16) montre, d'une part, que les performances d'une telle architecture sont supérieures à celle utilisant une fusion par un contrôleur de *type-1*, notamment, en présence d'un bruit de mesure très fort. D'autre part, elle montre que la forme des fonctions d'appartenance des contrôleurs flous n'influe pas d'une manière significative sur leurs performances. Pour assurer une navigation sans collision en présence d'un bruit fort, soit on augmente la distance de sécurité (D_s) proportionnellement avec l'écart type du bruit, où bien, en agissant sur les ensembles de la première variable d'entrée du contrôleur d'évitement d'obstacles (D_{ro}).

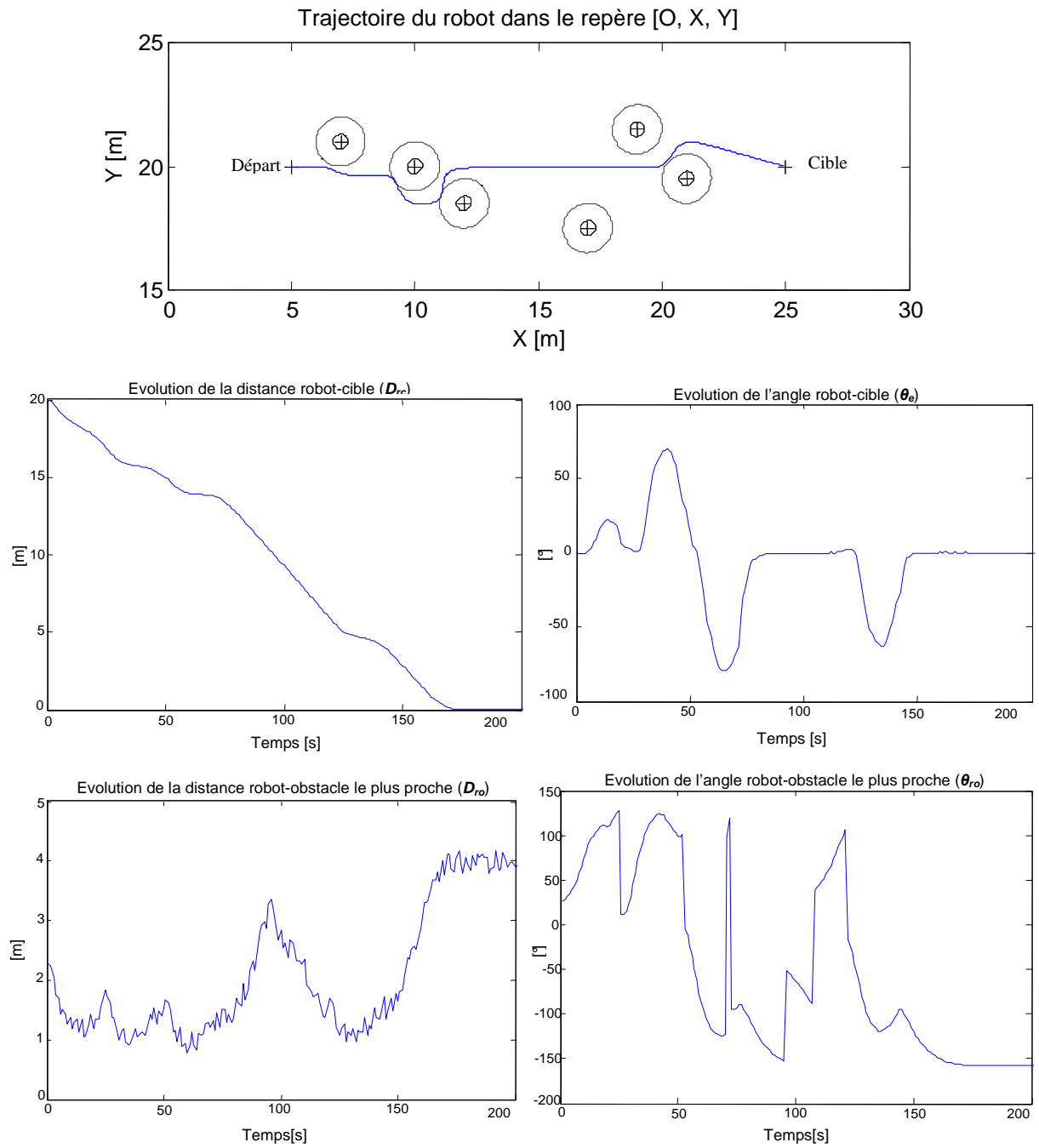


Fig.3.15- Simulation des incertitudes aléatoires en utilisant les systèmes flous *type-2*

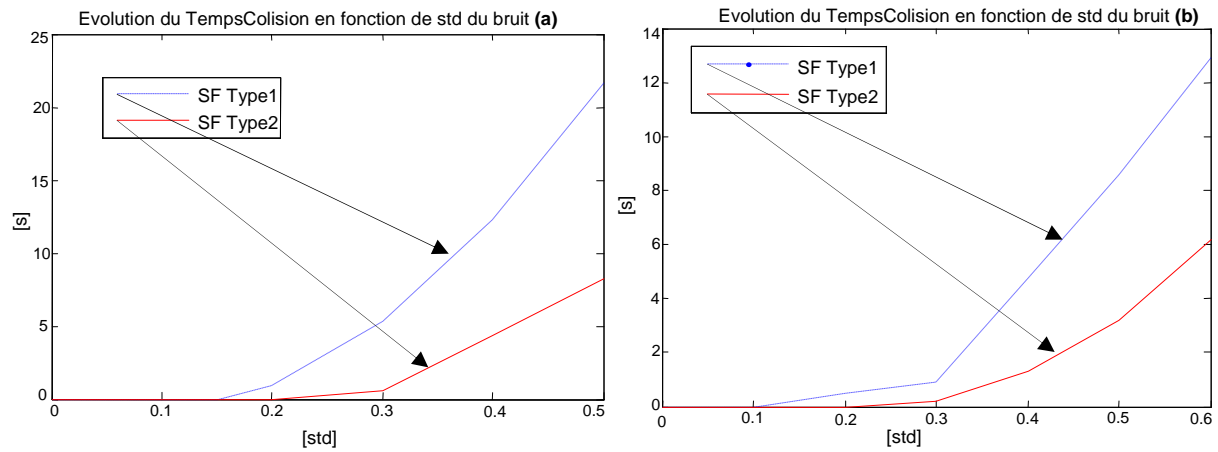


Fig.3.16 - Evolution du temps de collision en fonction de l'écart type (std) du bruit gaussien :

(a) ensembles triangulaires, (b) ensembles gaussiens

3.6.2 Etude comparative des contrôleurs flous *type-1* et *type-2*

L'évaluation de la fonction objective définie précédemment (cf. §3.5.3), à travers six environnements de navigation différents contenant six obstacles, dont les positions sont générées d'une manière aléatoire (cf. figure 3.17), et, on considérant que la mesure de distance D_{r_0} est entachée d'un bruit gaussien d'un écart type de 0.3, montre que :

-Le temps de collision moyen dans le cas du *type-1* est de : 5,25 s.

-Le temps de collision moyen dans le cas du *type-2* est de : 0,61 s.

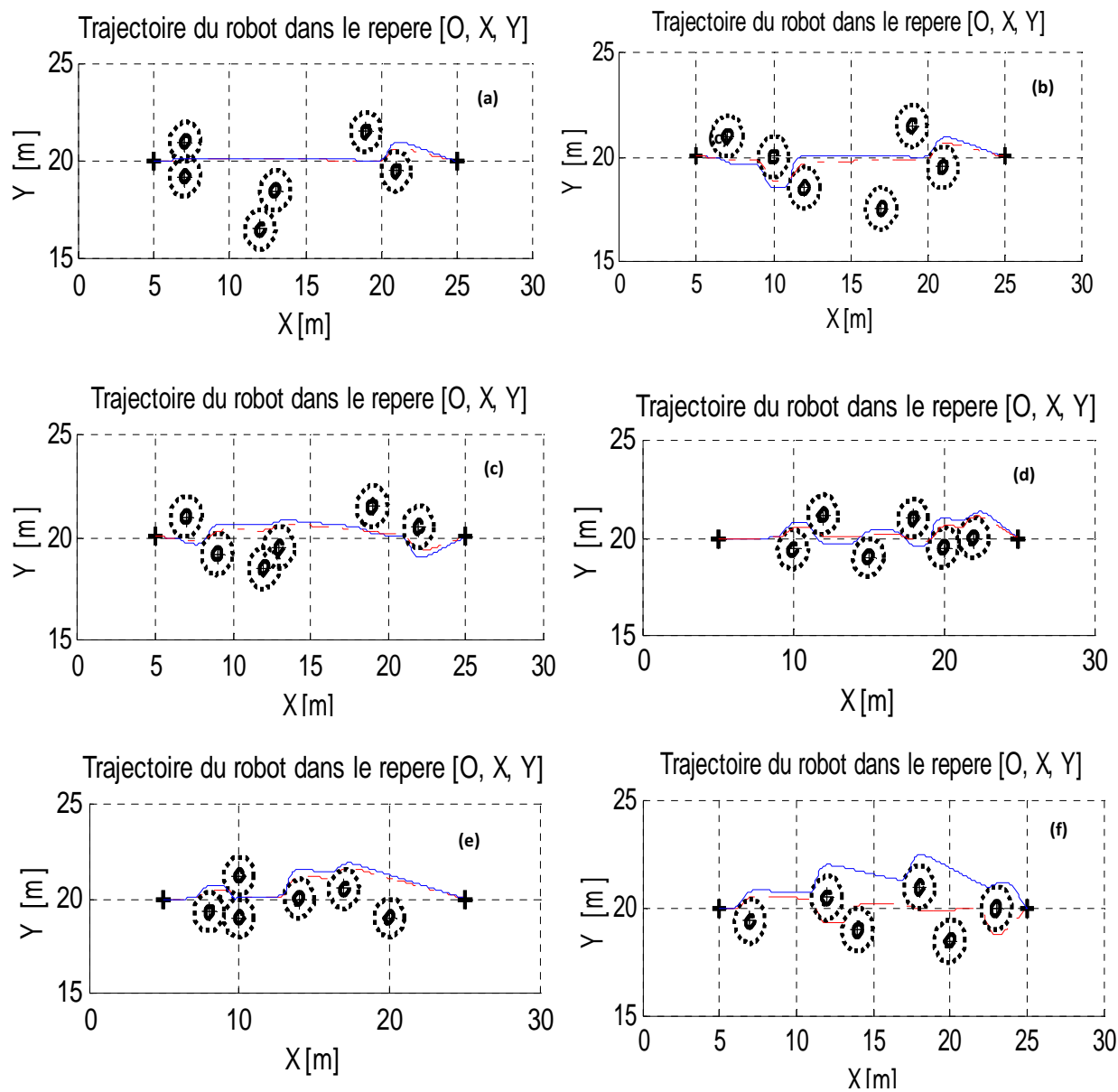


Fig 3.17- Convergence vers la cible en présence d'obstacles, en rouge:

Contrôleur flou de *type-1*, en bleu : contrôleur flou *type-2*

3.7 Introduction du mouvement en formation : attraction vers une cible dynamique

Dans le but d'aborder un mouvement en formation de plusieurs robots mobiles (navigation en convoi), objet de ce stage, on a procédé à simuler le mouvement d'attraction vers la cible vue en (§3.3 et §3.5), toute en donnant une dynamique à la cible (attraction vers une cible dynamique). Pour garantir que le robot atteigne la cible, la vitesse du robot doit être supérieure à celle de la cible.

3.7.1 Attraction vers une cible en mouvement rectiligne

Dans ce test, le mouvement de la cible est décrit par l'équation suivante :

$$x_c = x_{c0} + v * t \quad (3.7)$$

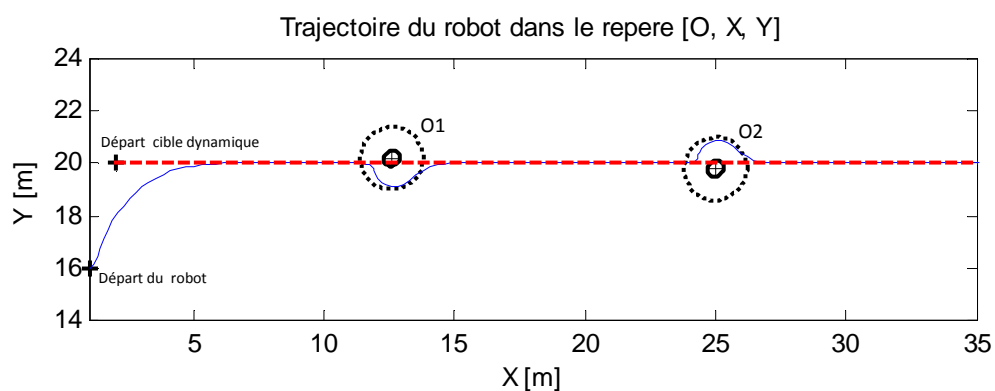
$$y_c = y_{c0}$$

avec : $v=0.2$, la vitesse de la cible ; t , est le temps.

x_c et y_c : abscisse et ordonnée courante de la cible respectivement.

$x_{c0}=2$ et $y_{c0}=20$: abscisse et ordonnée initiales de la cible respectivement.

Ici, comme dans (§3.52), la fusion de comportements est assurée par le contrôleur flou dont les entrées/sortie sont représenté par la figure (3.10).



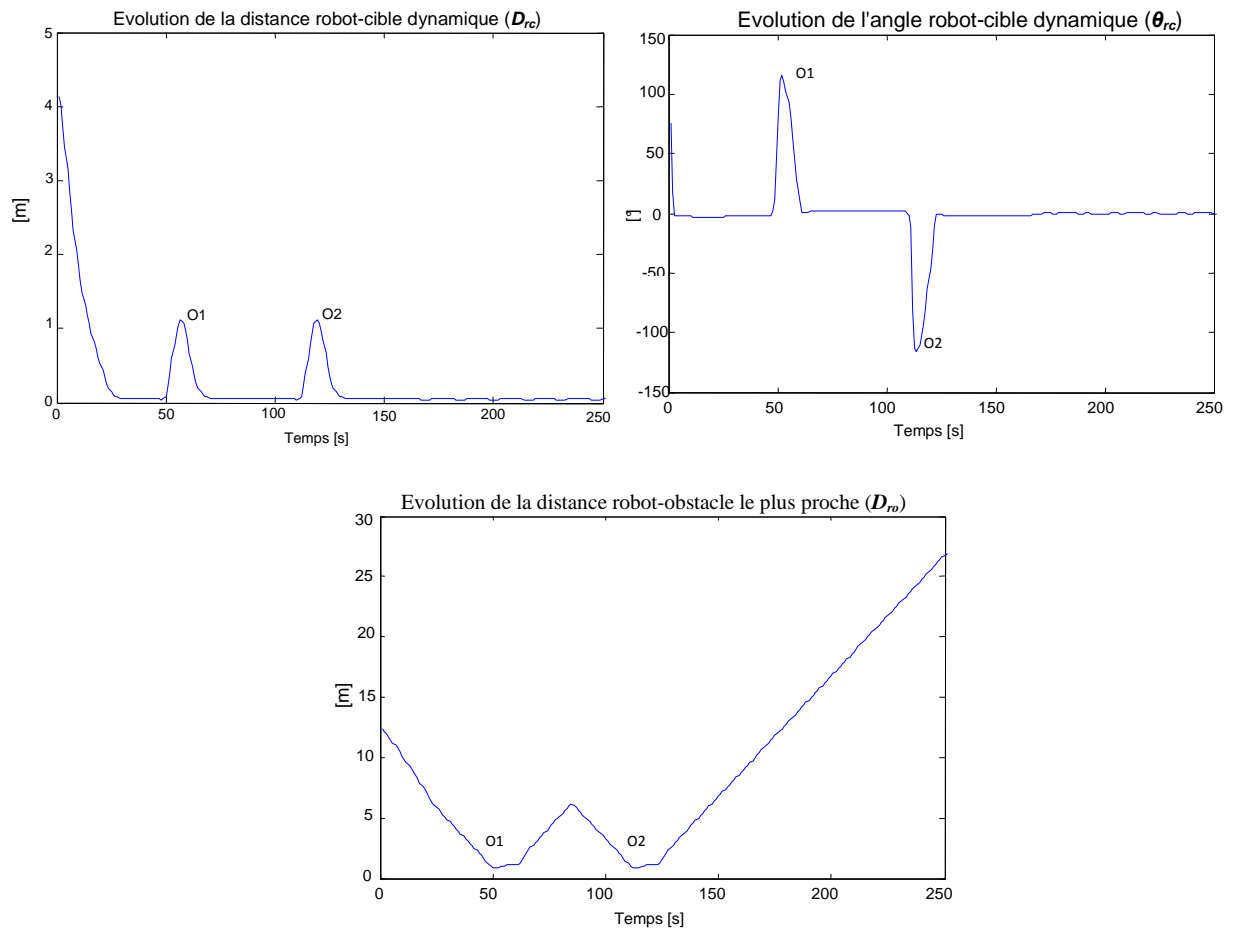


Fig 3.18 - Attraction vers une cible dynamique : mouvement rectiligne

3.7.2 Attraction vers une cible en mouvement sinusoïdale

Pour ce test, la dynamique de la cible est régie par l'équation suivante :

$$x_c = x_{c0} + v * t \tag{3.8}$$

$$y_c = y_{c0} + 2 * \sin(x_c)$$

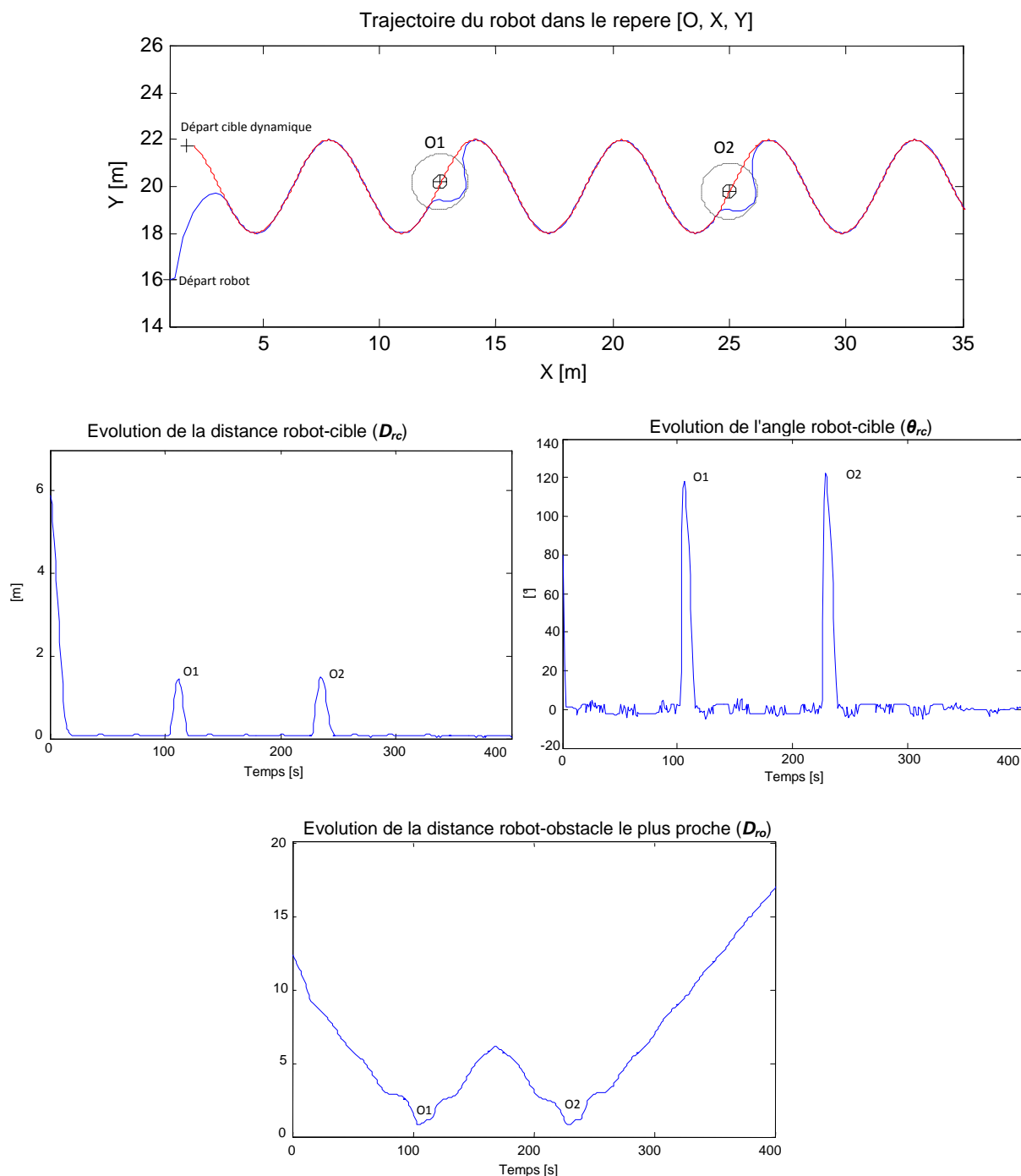


Fig 3.19 – Attraction vers une cible dynamique : mouvement sinusoïdal

Commentaires

Des résultats obtenus (cf. §3.7.1 et §3.7.2), on constate que le robot mobile doté du contrôleur flou d'attraction vers la cible et du contrôleur d'évitement d'obstacles dont les actions sont fusionnées par le contrôleur flou conçus en (§3.5.2), peut converger vers la cible dynamique toute en évitant les obstacles présents dans l'environnement. La distance séparant

le robot de la cible (D_{rc}) s'annule après peu de temps du lancement de la simulation. Cette distance devient importante une fois un obstacle est rencontré. Cela est valable aussi pour l'angle robot-cible (θ_{rc}).

3.7.3 Simulation des incertitudes

Pour tester la robustesse de l'architecture de commande dans le cas d'une cible dynamique, on considère, comme dans (§3.6) que la mesure de la distance robot obstacle est entachée d'un bruit gaussien. L'évolution du temps de collision du robot avec les obstacles en fonction de l'écart type (std) du bruit est représentée par la figure (3.20).

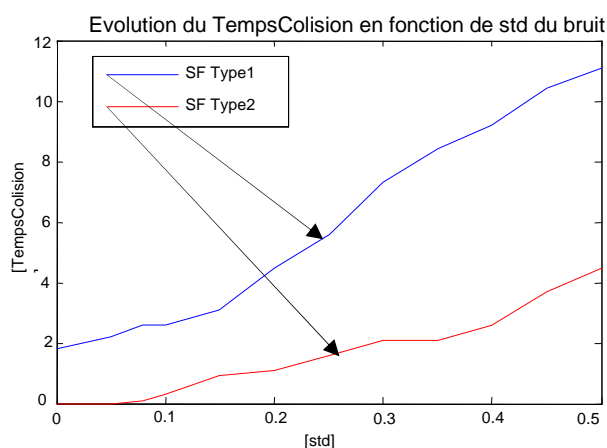


Fig 3.20 - Simulation des incertitudes aléatoires dans le cas d'une cible dynamique

Cette figure montre, une fois de plus, que l'utilisation des contrôleurs flous *type-2* donne de meilleurs temps de collision. Ce temps de collision peut être réduit davantage dans le cas des contrôleurs flous *type-2* en variant le *FOU* des ensembles flous de ces contrôleurs (annexe 3). Sinon, pour garantir une navigation sans collision (temps de collision nul), quelque soit le type du contrôleur flou utilisé, on doit choisir une distance de sécurité (D_s) suffisamment grande.

3.8 Conclusion

Dans ce chapitre, on a appliqué la logique floue pour la commande d'un robot mobile. Les comportements d'attraction vers la cible et d'évitement d'obstacles sont simulés à travers différents types d'environnements. En présence d'obstacles, les actions des contrôleurs d'attraction vers la cible et celles du contrôleur d'évitement d'obstacles doivent être coordonnées. La fusion de comportements par un gain flou adaptatif donne de meilleurs résultats par rapport, à la fois, à ceux obtenus en faisant la coordination par un switch simple entre les contrôleurs et à ceux de la fusion par une fonction sigmoïdale figée. On a montré aussi, l'apport des contrôleurs flous *type-2* par rapport à ceux du *type-1* notamment en présence des incertitudes de mesure. En fin du chapitre, une introduction à la navigation en convoi est abordée à travers l'attraction du robot vers une cible en mouvement.

Chapitre 4

Navigation en convoi d'un groupe de robots mobiles

Résumé : Ce chapitre traite le problème de navigation en convoi d'un groupe de robots mobiles. Cette tâche impose à ces derniers de maintenir une forme géométrique désirée. L'architecture de contrôle doit permettre à chaque entité robotique du convoi d'éviter d'éventuels obstacles présents dans l'environnement, puis, de rejoindre la formation. Le chapitre est organisé comme suit : d'abord, nous rappelons les objectifs envisagés, puis, on exposera la stratégie qui nous permettra d'obtenir un mouvement en formation. Nous donnerons, finalement, la structure de contrôle du convoi autonome à base de la logique floue. La robustesse de cette structure est testée en introduisant un bruit de mesure.

4.1 Introduction

Récemment, dans la littérature, plusieurs approches sont développées dans le but de réaliser une navigation en formation d'un groupe de robots mobiles [27]. Parmi eux, on peut citer l'approche de suivi du leader [28]. Dans cette approche quelque robots mobiles suivent une trajectoire prédéfinie, par contre d'autres robots ne font que suivre ces leaders. Une autre approche dite : l'approche de comportements de base. Dans cette approche, chaque robot possède certaines tâches de base à accomplir. Le comportement résultant du groupe de robots est issu des comportements de base, et ce, malgré l'inexistence d'un modèle explicite du comportement coopératif global [29] [30]. D'autres approches se basent sur des structures géométriques virtuelles. Quelque soit l'approche considérée, la coordination entre les différentes entités robotiques participant dans le convoi est la difficulté principale dans une telle tâche. Dans ce travail, la navigation d'un groupe de robots mobiles en présence d'obstacles est considérée. A cet effet, nous proposons de combiner une approche comportementale avec une structure virtuelle pour concevoir l'architecture de commande du convoi [26] [27].

4.2 Objectifs

Le travail demandé dans ce stage consiste à proposer une architecture de contrôle distribuée pour la navigation en formation (déplacement en file indienne, en triangle, ..etc) de robots *Khepera III*, et ce, en présence d'obstacles. Ces mini-robots mobiles vont nous servir pour le prototypage de nos lois de commande avant leur implantation finale sur les véhicules *CyCab*. L'architecture de contrôle/commande qui va être proposée utilisera les outils de la logique floue pour réaliser la fusion appropriée des commandes issues d'une multitude de contrôleurs élémentaires (attraction vers la cible, évitement d'obstacles, ..etc).

4.3 Stratégie adoptée pour la réalisation d'un convoi autonome

La démarche pour réaliser une navigation autonome de N robots mobiles suivant une formation donnée en présence d'obstacles, consiste à contrôler chaque robot i pour suivre une cible dynamique d'une structure géométrique virtuelle. La convergence vers une cible dynamique est largement explorée dans la littérature. La condition principale pour que le robot

mobile puisse atteindre une cible dynamique c'est que la vitesse du robot doit être supérieure à celle de la cible dynamique. La structure dynamique virtuelle que doit suivre le groupe de robots est définie comme suit [27] (cf. figure 4.1):

- définir un point qui va donner la dynamique de la structure. Ce point est considéré la cible dynamique principale.
- définir la structure virtuelle à suivre en définissant autant de cibles dynamiques virtuelles nécessaires pour obtenir la géométrie désirée. Chaque cible i est appelée cible secondaire et elle est définie par la distance D_i et un angle θ_{ric} par rapport à la cible principale. Les cibles secondaires ainsi définies possèdent la même orientation θ_c et la même vitesse v_c que la cible principale. Un exemple pour obtenir une formation triangulaire est illustré sur la figure (4.1).

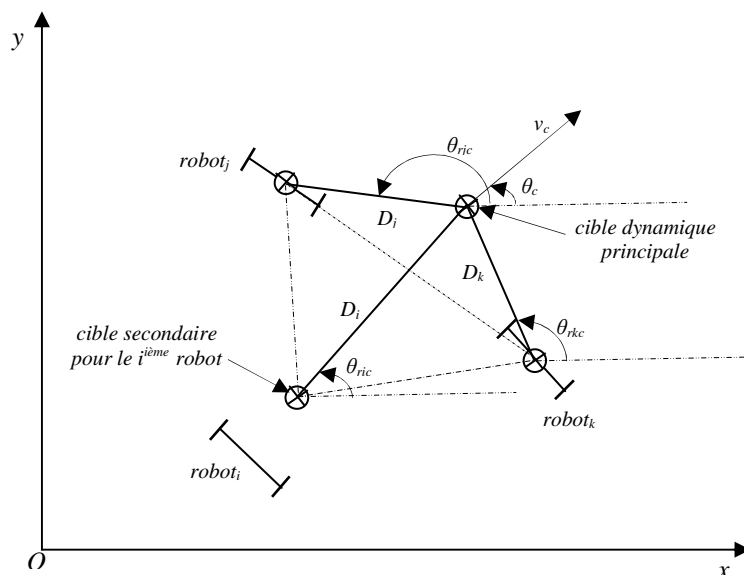


Fig 4.1- Structure virtuelle pour maintenir une formation triangulaire

4.4 Quel robot pour quelle cible dynamique ?

Comme il a été dit auparavant, chaque robot mobile doit suivre une cible secondaire de la structure géométrique désirée. Mais la question qui se pose : quelle cible secondaire pour quel robot ? Les seules informations dont on dispose au niveau de chaque robot sont sa position (x, y) et son orientation par rapport à la cible dynamique principale θ_{rc} . Par contre, les positions des cibles dynamiques secondaires sont déduites des distances et angles séparant ces cibles secondaires à la cible principale (cf. figure 4.1). Ici, et comme dans [27],

l'association robot/cible dynamique secondaire se fait de sorte que chaque robot suit la cible qui lui est très proche et qui n'est pas déjà affectée à un autre robot de rang supérieur.

4.5 Structure de commande du convoi

L'architecture de contrôle que nous proposons pour le contrôle du groupe de robots mobiles inclut les deux contrôleurs flous décrits dans le chapitre précédent (cf. §3.4 et §3.5) : contrôleur d'attraction vers une cible dynamique et un contrôleur d'évitement d'obstacles, le contrôleur de fusion de comportements (§3.6.2), ainsi que toutes les entités robotiques participant dans le convoi. De plus, l'architecture de contrôle est dotée d'un bloc (mécanisme) qui permettra d'associer chaque robot mobile à une cible dynamique secondaire (cf. figure 4.2).

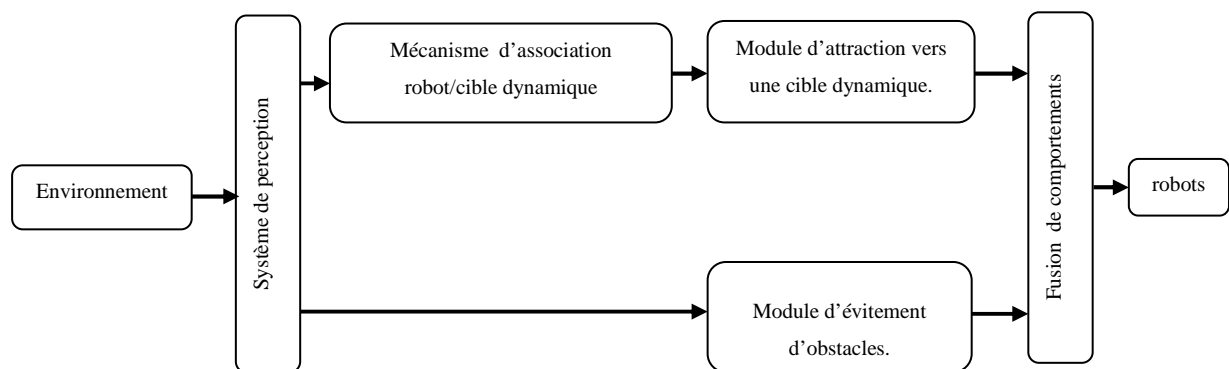


Fig.4.2 - Schéma de contrôle du convoi

Résultats et commentaires

L'architecture de la figure (4.3) a été simulée pour deux types de formation : navigation en forme triangulaire (cf. figure 4.4.a) et une navigation en file indienne (convoi) (cf. figure 4.4.b). Les résultats obtenus confirment l'efficacité de cette architecture de commande. Chaque robot participant dans le convoi est capable, une fois affecté à une cible dynamique, de prendre des décisions d'une façon autonome des autres robots, et ceci, en gardant la formation désirée. Il est clair, qu'à l'approche d'un obstacle, le mouvement de formation est suspendu jusqu'à ce que ce dernier soit complètement dépassé par le robot. La distance robot-

cible dynamique est nulle le long du mouvement sauf à l'approche d'un obstacle (cf. figure 4.5).

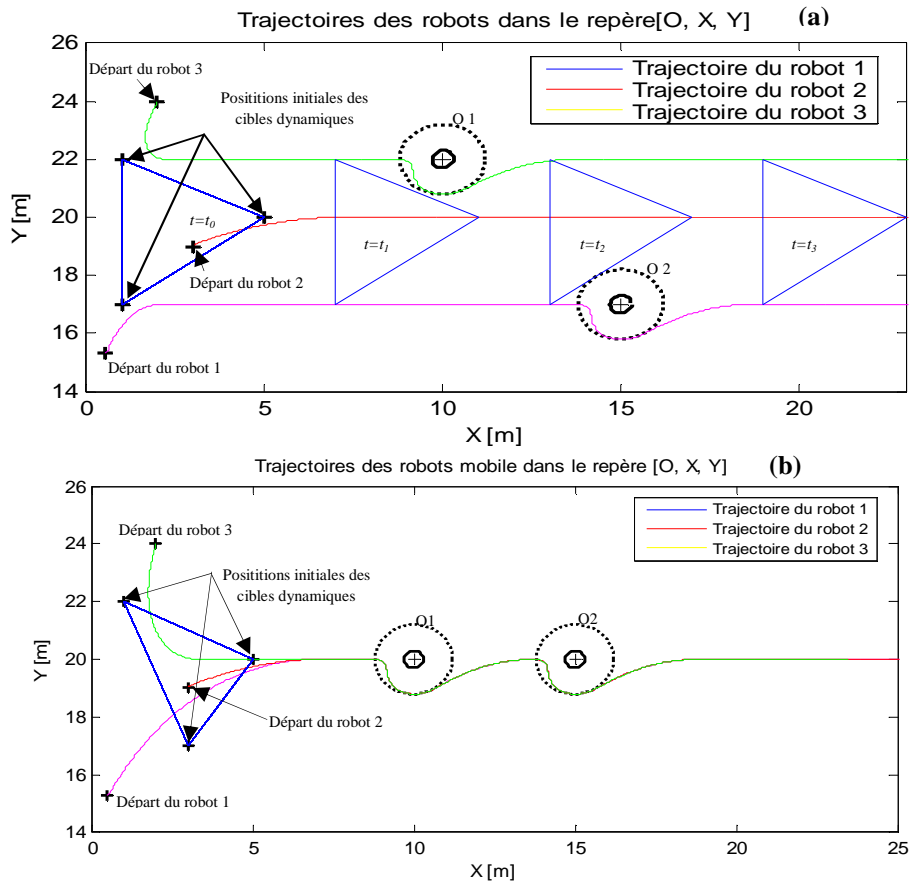


Fig 4.4- Trajectoires des robots mobiles : (a) navigation en formation triangulaire, (b) navigation en convoi (file indienne).

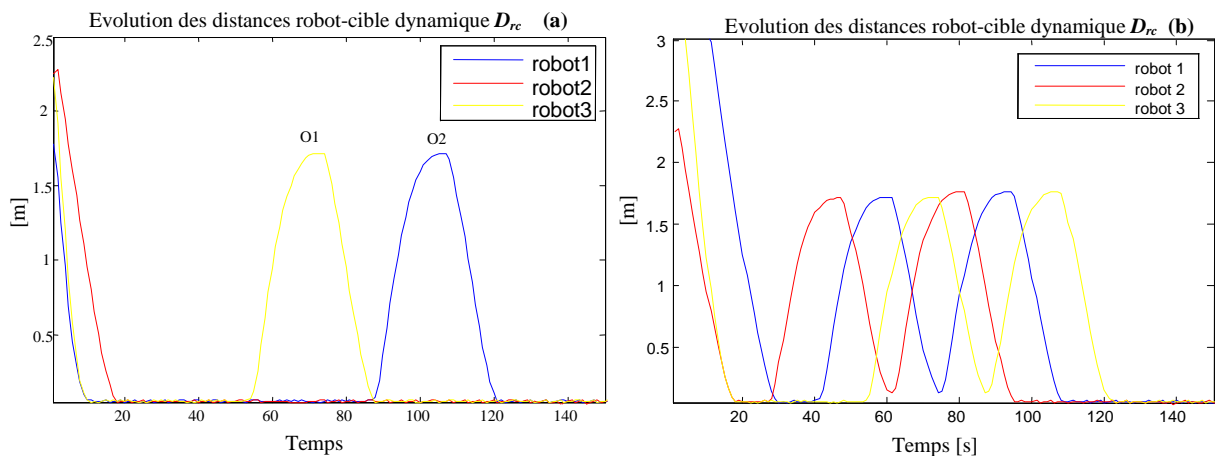


Fig 4.5- Evolution des distances robots-cibles dynamiques, (a) : triangle, (b) : file indienne

4.6 Conclusion

Dans ce chapitre, le problème de navigation en formation d'un groupe de robots mobiles (soit trois robots) a été étudié à travers une structure de contrôle/commande comportementale à base de la logique floue. La structure de commande développée permet aux robots mobiles participant au convoi de garder une certaine forme géométrique désirée (forme triangulaire, convoi), définie au travers de cibles virtuelles. Comme la navigation se fait dans un environnement contenant des obstacles, la structure de contrôle/commande permet aussi à chacun des robots d'éviter ces derniers, puis, de rejoindre la formation. La fusion des actions du module d'attraction vers la cible dynamique et celles du module d'évitement d'obstacles est assurée par un autre module dit contrôleur de fusion de comportements illustré en (§.3.5.2).

Conclusion générale

Le travail présenté dans ce manuscrit a pour objectif final la commande par logique floue d'un groupe de robots mobiles pour une navigation en convoi. Le premier chapitre résume une recherche bibliographique sur la robotique en général. Les architectures de contrôle/commande pour la robotique, la robotique collective ainsi que la coopération de robots sont abordés. En fin du chapitre, on a cité les différentes approches de commande en robotique.

Le deuxième chapitre a été consacré pour exposer en détail l'outil *-logique floue-* que nous avons adopté pour traiter la problématique en question. Cette approche présente un avantage majeur du fait qu'elle est possible sans la connaissance d'un modèle explicite du système à commander. De plus, les contrôleurs flous de *type-2*, vue leur structure intégrant des fonctions avec une dimension supplémentaire leur permettant une meilleure prise en compte de ces incertitudes, Mais, cela, au prix d'un degré plus élevé de complexité qui fait, même, la raison de leur timide utilisation.

La complexité des systèmes flous *type-2* est due essentiellement à la nécessité de la réduction du type des ensembles de sortie résultant de l'inférence floue dans ces systèmes. Dans notre cas, on a utilisé les systèmes flous *type-2* intervalle (degré d'appartenance secondaire est égal à 1) relativement facile.

Le troisième chapitre traite le problème de la navigation autonome d'un seul robot mobile. Les comportements d'attraction vers la cible, d'évitement d'obstacles et la coordination de comportements ont été développés. Les tests de simulation ont montrés que les résultats obtenus dans le cas de la fusion de comportements par un gain adaptatif flou sont, à la fois, meilleurs de ceux obtenus par la coordination par un switch entre les comportements ainsi qu'à ceux obtenus en faisant la fusion par une sigmoïde figée. D'autre part, les tests de simulation ont confirmé la supériorité des systèmes flous *type-2* par rapport à ceux du *type-1* en minimisant l'effet des incertitudes de mesure.

En fin du chapitre, le mouvement en formation est introduit à travers l'attraction d'un robot mobile vers une cible dynamique.

Le dernier chapitre est consacré à la navigation en convoi d'un groupe de robots mobiles. A cet effet, une architecture de commande comportementale réactive exploitant une structure de cibles virtuelles donnant la forme géométrique désirée a été développée et simulée.

Bibliographie

- [1] R. Siegwart & R. Nourbakhsh, Introduction to autonomous mobile robot, 'MIT Press Cambridge', Massachusetts, 2004.
- [2] R. Brooks, A layered control system for a mobile robot, 'IEEE JN of robotics and automation', vol. RA-2 n°1, p14-23, 1986.
- [3] L. Adouane, Architectures de contrôle comportementales et réactives pour la coopération d'un groupe de robots mobiles, PhD thesis, Université de Franche-Comté, UMR CNRS 6596, Avril 2005, Thèse N°1071.
- [4] S. Sam Ge & Frank L. Lewis, Autonomous mobile robots sensing, control, decision making and applications, 'CRC press taylor & francis group', 2006.
- [5] D. Han Kim, Real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer, 'JN. Robotics and Autonomous Systems', p17-30, 2003.
- [6] J. Bom, Etude et mise en œuvre d'un convoi de véhicules urbains avec accrochage immatériel, ' PhD thesis, Ecole Doctorale Sciences pour L'Ingénieur de Clermont-Ferrand, EDSPIC N° 1666.
- [7] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-1, 'Information sciences', vol. 8, pp. 199 – 249, 1975.
- [8] J. M. Mendel, Fuzzy logic systems for engineering: A tutorial, 'IEEE proceedings', vol. 83, no. 3, pp. 345-377, 1995.
- [9] G. J. Klir & B. Yuan, Fuzzy sets and fuzzy logic: theory and applications, 'Prentice Hall', 1995.
- [10] J. R. JANG & C. T. Sun, Neuro-Fuzzy modeling and control, 'Proceedings of the IEEE', vol. 83, no. 3, 1995.
- [11] L. Ruthowski, Computational intelligence: methods and techniques, 'Springer', 2008.
- [12] K. M. Passino, S. Yurkovich, Fuzzy control, 'Addison-Wesley', 1998.
- [13] L. Reznik, Fuzzy controllers, 'Biddles Ltd', 1997.

- [14] J. R. JANG & C. T. Sun, Neuro-Fuzzy modeling and control, 'Proceedings of the IEEE', vol. 83, no. 3, 1995.
- [15] N. N. Karnik & J. M. Mendel, Type-2 fuzzy logic systems, 'IEEE transactions on fuzzy systems', vol. 7, no. 6, 1999.
- [16] O. Castillo & P. Melin, Type-2 fuzzy logic: theory and applications, 'Springer', 2008.
- [17] B. Bouchon-Meunier & C. Marsala, Logique floue, principe, aide à la décision, 'Lavoisier', 2003.
- [18] Q. Liang, J. M. Mendel, 'Interval type-2 fuzzy logic systems: theory and design', IEEE transactions on fuzzy, vol. 8, no. 5, 2000.
- [19] N. N. Karnik & J. M. Mendel, Type-2 fuzzy systems: Type reduction, 'In SMC'98 Conference Proceedings', IEEE International conference on systems, Man, and Cybernetics, Part 2, pp. 2046-2051, 1998.
- [20] H. A. Hagra, Hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, 'IEEE Trans on Fuzzy Systems', Vol. 12, No. 4, pp. 524-539, 2004.
- [21] L. Adouane, Hybrid and safe control architecture for mobile robot navigation, 'In 9th conference on autonomous robot systems and competition', Portugal, 2009.
- [22] L. Adouane, Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation, 'In 9th conference on autonomous robot systems and competition', Portugal, 2009.
- [23] L. Adouane, An adaptive multi-controller architecture for mobile robot navigation, 'In 10th IAS, Intelligent Autonomous systems, Baden-Baden, Germany, July 2008, pp 342-347.
- [24] M. Ider, B. Soualmi, B. Mendil, Incorporation des incertitudes dans les contrôleurs en utilisant la logique floue type-2, '3rd international conference on electrical and engineering', Alger's, Algeria, May 2009.
- [25] E. Tunstel, Mobile robot autonomy via hierarchical fuzzy behaviour control, '6th intl symp on robotics & manuf', France, 1996.

- [26] L. Adouane, Hybrid behavioral controle architecture for the cooperation of minimalist mobile robots, 'Intl conf on robotics & automation', Orleans, USA, April 2004.
- [27] A. Benzerrouk, L. Adouane, L. Lequievre & P. Martinet, Navigation of multi-robot formation in unstructured environment using dynamical virtual structures, 'IROS'0 IEEE/RSJ intl conf on intelligent robots and systems', Taipei, Taiwan, 18-22 octobre.
- [28] H.G. Tanner, G.J. Pappas, and V. Kumar, Leader -to-formation stability, IEEE Transactions on Robotics and automation, 20(3):433U455, 2004.
- [29] G. Antonelli, F. Arrichiello, S. Chakraborti, and S. Chiaverini, Experiences of formation control of multi-robot systems with the null-space-based behavioural control, int conf on robotics and automation, page 1068-1073, 2007.
- [30] T. Balch and R. C. Arkin, Behavior-based formation control for multirobot systems, IEEE Transactions on robotics and automation, 14(12):926U939, 1998.

Annexes

Un sous-ensemble flou est complètement défini par la donnée de sa fonction d'appartenance. A partir d'une telle fonction, un certain nombre de caractéristiques du sous-ensemble flou peuvent être étudiées. Les définitions utilisées ici sont celles données dans [7] [9] [13] [17].

- *Noyau* : le noyau d'un sous-ensemble flou A de X , noté $Noy(A)$, est l'ensemble de tous les éléments qui lui appartiennent totalement. Formellement :

$$Noy(A) = \{ x \in X \mid \mu_A(x) = 1 \} \quad (\text{A.1.1})$$

- *Support* : le support d'un sous ensemble flou A de X , $Supp(A)$, est l'ensemble de tous les éléments qui lui appartiennent totalement. Formellement :

$$Supp(A) = \{ x \in X \mid \mu_A(x) > 0 \} \quad (\text{A.1.2})$$

- *Hauteur* : la hauteur d'un sous ensemble flou A de X , notée $h(A)$, est la valeur maximale atteinte sur le support de A . Formellement :

$$h(A) = \sup_{x \in X} \mu_A(x) \quad (\text{A.1.3})$$

On dira alors qu'un sous-ensemble flou est normalisé si sa hauteur $h(A)$ est égale à l'unité.

- *Cardinalité* : la cardinalité d'un sous-ensemble flou discret A de X , notée $|A|$, est définie par la somme des degrés d'appartenance, Formellement :

$$|A| = \sum_{i=1}^n \mu_A(x) \quad (\text{A.1.4})$$

- α – coupe : α – coupe A_α d'un ensemble flou A est un sous ensemble ordinaire de l'univers de discours X dont tous ses éléments possèdent un degré d'appartenance supérieur ou égal à α :

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\} \quad \alpha \in [0 \quad 1] \quad (\text{A.1.5})$$

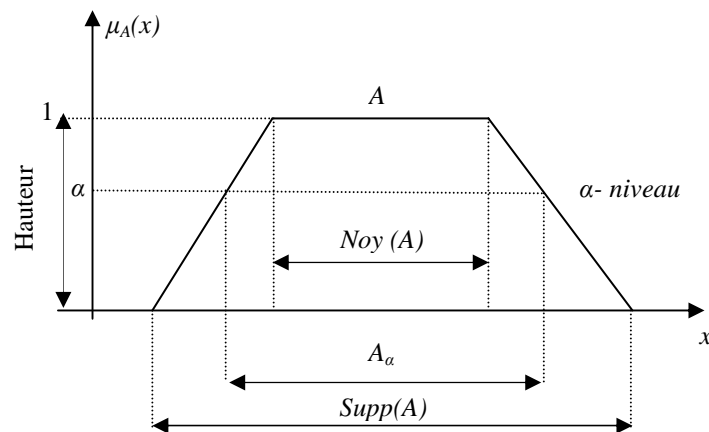


Fig.A.1. Caractéristiques d'un sous ensemble flou.

Un ensemble flou unidimensionnel A sur un univers de discours X est défini par sa fonction d'appartenance $\mu_A(x)$, où x est un élément de X , c-à-d

$$A = \int_X \mu_A(x) | x \quad \text{cas continu} \quad (\text{A.2.1})$$

$$A = \sum_i \mu_A(x) | x_i \quad \text{cas discret} \quad (\text{A.2.2})$$

Où $\mu_A(x)$ désigne le degré d'appartenance d'un élément x à l'ensemble flou A .

Si A_1, A_2, \dots, A_n sont des ensembles flous sur X_1, X_2, \dots, X_n respectivement, le produit cartésien de A_1, A_2, \dots, A_n noté par $A_1 \times A_2 \times \dots \times A_n$ est défini comme un sous ensemble flou possédant la fonction d'appartenance multidimensionnelle suivante :

$$\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n) = \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) \quad (\text{A.2.3})$$

Ce qui est équivalent à :

$$A_1 \times \dots \times A_n = \int \dots \int \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) | (x_1, \dots, x_n) \quad (\text{A.2.4})$$

Le terme $\mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n)$ représente une fonction d'appartenance multidimensionnelle représentant une relation \mathfrak{R} .

2.1. Principe d'extension monodimensionnel

Le principe d'extension des ensembles flous est essentiellement une identité de base qui permet à un domaine de définition d'une relation à être étendu de quelques points de X à des sous ensembles flous de X . Particulièrement, supposons que f est une relation de X vers Y et A un ensemble flou défini par (A.2.1) si A continu et par (A.2.2) si A est discret. Alors le principe d'extension affirme que l'image de A sous la relation f est un ensemble flou, $f(A)$ défini par :

$$f(A) = f \left[\int_X \mu_A(x) | x \right] = \int_Y \mu_A(x) | f(x) \quad \text{cas continu} \quad (\text{A.2.5})$$

$$f(A) = f \left[\sum_i \mu_A(x_i) \mid x_i \right] = \sum_i \mu_A(x_i) \mid f(x_i) \quad \text{cas discret} \quad (\text{A.2.6})$$

Par conséquent, l'image de A sous f peut être déduite en connaissant les images de u_i sous f .

- **Exemple 1**

Soit $U = 1+2+\dots+10$, et soit f l'opération « carré ». Soit *petit* un sous ensemble flou sur X définit par :

$$\text{petit} = 1/1+1/2+0.8/3 + 0.6/4 + 0.4/5 \quad (\text{A.2.7})$$

par conséquent :

$$\text{petit}^2 = f(\text{petit}) = 1/1+1/4+0.8/9+0.6/16 + 0.4/25 \quad (\text{A.2.8})$$

2.2. Principe d'extension multidimensionnel

Soit A un ensemble flou multidimensionnel sur $X_1 \times X_2 \times \dots \times X_n$ défini par sa fonction d'appartenance multidimensionnelle $\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n)$, et soit f une relation sur produit cartésien $X_1 \times X_2 \times \dots \times X_n$ vers l'espace Y .

L'application directe du principe d'extension nous donne :

$$\begin{aligned} f(A) &= f \left[\int_{X_1} \dots \int_{X_n} \mu_A(x_1, \dots, x_n) / (x_1, \dots, x_n) \right] \\ &= \int_{X_1} \dots \int_{X_n} \mu_A(x_1, \dots, x_n) / f(x_1, \dots, x_n) \\ &= \int_{X_1 \times \dots \times X_n} \mu_A(x_1, \dots, x_n) / f(x_1, \dots, x_n) \quad (\text{A.2.9}) \\ &= \int_{X_1, \dots, X_n} \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) / f(x_1, \dots, x_n) \\ &= \int_Y \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) / f(x_1, \dots, x_n) \end{aligned}$$

Les μ_{A_i} sont les projections de A sur les X_i . Parce que, généralement, on ne connaît pas A mais ces projections A_1, A_2, \dots, A_n .

- **Exemple 2**

Soit * une relation du produit cartésien $X \times V$ vers W Donc nous avons :

$$w = x * v \quad x \in X, v \in V \text{ et } w \in W \quad (\text{A.2.9})$$

Ou w est l'image de x et de v sous*.

Soit A et B deux ensembles flous sur X et V respectivement :

$$A = \int_X \mu_A(x) | x \quad \text{où} \quad A = \sum_i \mu_A(x_i) | x_i \quad (\text{A.2.10})$$

$$B = \int_V \mu_B(v) | v \quad \text{où} \quad B = \sum_i \mu_B(v_i) | v_i \quad (\text{A.2.11})$$

Alors l'image de $A \times B$ sous * est un sous ensemble flou $A * B$ sur W défini par :

$$f(A, B) = \int_X \int_V \mu_A(x) \wedge \mu_B(v) | (x * v) \quad (\text{A.2.12})$$

où :

$$f(A, B) = A * B = \sum_i \sum_j \mu_A(x_i) \wedge \mu_B(v_j) | (x_i * v_j) \quad i=1, \dots, m \quad j=1, \dots, n \quad (\text{A.2.13})$$

Les définitions de 1 à 8 sont celles données dans [11] [16]

Définition 1

Un ensemble flou *type-2* noté \tilde{A} est caractérisé par une fonction d'appartenance tridimensionnelle, $\mu_{\tilde{A}}(x, u)$, c-à-d :

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu(x, u) \mid (x, u) \quad J_x \subseteq [0, 1] \quad (\text{A.3.1})$$

où $\int \int$ dénote l'union de tous les éléments du produit cartésien sur x et u avec $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. A chaque point fixe x de X , J_x est l'appartenance primaire de x , et x est appelé variable primaire.

Définition 2

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

Donc pour $x' \in X$ et $\forall u \in J_{x'} \subseteq [0, 1]$, on a :

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) \mid u \quad J_{x'} \subseteq [0, 1] \quad (\text{A.3.2})$$

où $0 \leq f_{x'}(u) \leq 1$. Puisque $\forall x'$, cet x' appartient à X , c-a-d $x' \in X$, alors nous notons la fonction d'appartenance secondaire par $\mu_{\tilde{A}}(x)$, qui est une fonction d'appartenance flou *type-1*. En se basant sur le concept des ensembles secondaires, on peut réinterpréter un ensemble flou *type-2* comme l'union de tous les ensembles secondaire, c-à-d, en utilisant l'équation (d.2), on peut écrire \tilde{A} sous la forme suivante :

$$\tilde{A} = \left\{ (x, \mu_{\tilde{A}}(x)) \mid \forall x \in X \right\} \quad (\text{A.3.3})$$

où par :

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x) \mid x = \int_{x \in X} \left[\int_{u \in J_x} f_x(u) \mid u \right] \mid x \quad J_x \subseteq [0, 1] \quad (\text{A.3.4})$$

Définition 3

Le domaine de la fonction d'appartenance secondaire est appelée degré d'appartenance primaire de x , noté J_x telle que $J_x \subseteq [0, 1] \quad \forall x \in X$.

Définition 4

L'amplitude de la fonction d'appartenance secondaire est appelée degré d'appartenance secondaire, notée $f_x(u)$.

Définition 5

Un ensemble flou intervalle est un ensemble flou *type-2* dont toutes les fonctions d'appartenance secondaires sont des ensembles *type-1* de formes intervalles, c-a-d

$$f_x(u) = 1, \quad \forall u \in J_x \subseteq [0, 1], \quad \forall x \in X. \quad (\text{A.3.5})$$

Les ensembles flous intervalle reflètent l'uniformité de l'incertitude au niveau de la fonction d'appartenance primaire. Ce type de fonctions d'appartenance est le plus souvent utilisé dans les systèmes flous *type-2*. Notons que ce type de fonctions d'appartenance est représenté seulement par ses domaines (intervalles) qui peuvent être exprimés en fonction des bornes gauches et droites $[l, r]$ ou par leurs centres et largeurs $[c-s, c+s]$ où $c = (l+r)/2$ et $s = (r-l)/2$.

Définition 6

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 :

$$u_{\text{principale}}(x) = \int_{x \in X} u \mid x \quad \text{où } f_x(u) = 1 \quad (\text{A.3.6})$$

Pour un ensemble flou *type-2* intervalle, nous définissons la fonction d'appartenance principale comme étant l'union de toutes les valeurs d'appartenance moyennes de la fonction d'appartenance primaire. Notons que lorsque l'incertitude des fonctions d'appartenance disparaît, la fonction d'appartenance d'un ensemble flou *type-2* se réduit à la fonction d'appartenance principale.

Définition 7

L'incertitude dans un ensemble flou *type-2* \tilde{A} est représentée par une région bornée désignée par *FOU* « *Footprint Of Uncertainty* ». C'est l'union de toutes les appartenances primaires (voir figure A.2):

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \quad (\text{A.3.7})$$

Définition 8

La fonction d'appartenance supérieur et la fonction d'appartenance inférieur sont des fonctions d'appartenance de *type-1*, et elles sont les frontières du *FOU* d'un ensemble flou \tilde{A} *type-2*. La première sera notée par $\bar{\mu}_{\tilde{A}}(x) \quad \forall x \in X$. Par contre, la seconde est notée par $\underline{\mu}_{\tilde{A}}(x) \quad \forall x \in X$ alors nous avons (voir figure A.2):

$$\begin{aligned} \bar{\mu}_{\tilde{A}}(x) &= \overline{FOU(\tilde{A})} & \forall x \in X \\ \underline{\mu}_{\tilde{A}}(x) &= \underline{FOU(\tilde{A})} & \forall x \in X \end{aligned} \quad (\text{A.3.8})$$

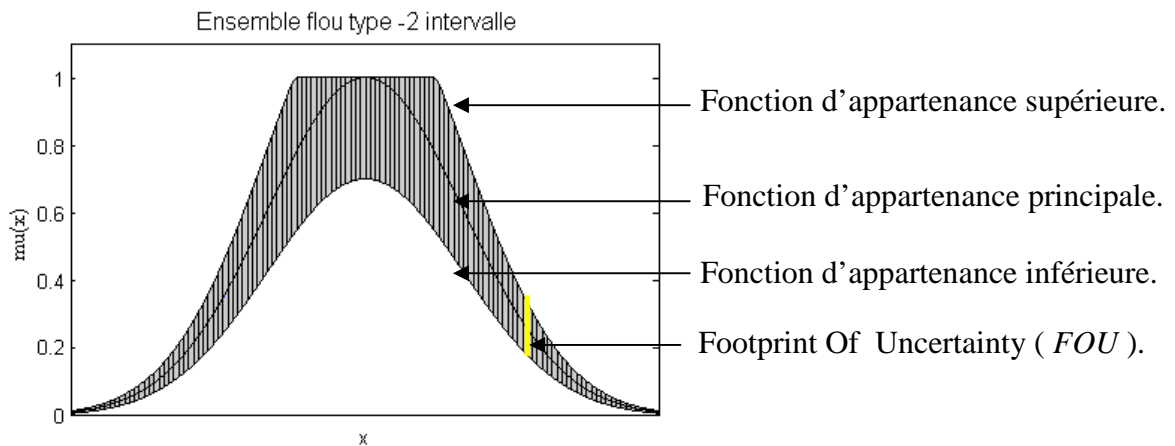


Fig. A.2. Représentation bidimensionnelle d'un ensemble flou *type -2* intervalle.

L'implémentation pratique des structures de contrôle/commande sur les robots *Khepera III* nécessite la prise en compte de certaines contraintes pratiques à savoir : le calcul des vitesses maximales admissibles par ce type de robots mobiles.

La vitesse linéaire maximale admissible par une roue dans le cas des robots mobiles *khepera III* est de $v_{max}=0.5m/s$. D'où la vitesse angulaire maximale de chacune des deux roues :

$$\omega_{max} = v_{max}/r = 24,39 \text{ rad/s.}$$

avec : $r = 20,05 \text{ mm}$: le rayon de la roue.

La vitesse maximale qui peut être appliquée au robot dans notre cas est :

$$\omega_{max} = -r * (v_{dmax} + v_{gmax}) / 2 * L = 7,6 \text{ rad/s.}$$

avec : v_{dmax} , v_{gmax} : vitesses maximales de la roue droite et gauche respectivement.

L : La moitié de la distance séparant les deux roues du robot.

