

# ETUDE ET SIMULATION PAR LA LOGIQUE FLOUE ET LES ALGORITHMES GENETIQUES DE LA PRISE EN CHARGE D'UNE TACHE COOPERATIVE PAR UN SYSTEME MULTI-AGENTS DE ROBOTS MOBILES

Ould Ebbat Baba<sup>1</sup>, M. Hadjila<sup>2</sup>, B. Cherki  
Département d'électronique  
Faculté des sciences de l'ingénieur  
Université Abou Bakr Belkaïd -Tlemcen  
B.P. 119

Email<sup>1</sup> : zakaria943@caramail.com, Fax : (043) 28 56 85

## Résumé

Dans cet article, nous présentons l'étude d'une tâche coopérative par un système multi-agents de robots mobiles. Le système est constitué d'un ensemble de robots, chacun peut communiquer avec les autres afin de réaliser la tâche demandée. Cette dernière réside dans la détection de la cible par les différents robots ensuite l'orientation vers cette cible et enfin, son encerclement. Pour cela, nous avons combiné les concepts de logique floue et d'algorithmes génétiques pour résoudre ce genre de problème. Le premier concept consiste à décrire le contrôleur flou alors que le second permet de trouver le contrôleur optimisé.

**Mots clés:** système multi agent, logique floue, algorithmes génétiques, robotique mobile.

## I- Introduction

Depuis longtemps, l'homme a toujours eu l'ambition de faire reproduire par des dispositifs artificiels des actions humaines. Des problèmes qui sont déjà épineux, pour un seul robot, et qui ne pourront que l'être à fortiori, s'il s'agissait d'une société - tout entière- de robots évoluant au même moment et à travers les mêmes lieux ! Et tout particulièrement si l'on s'intéressait à l'aspect collectif de leurs reflets et comportements.

Cette évolution nécessite que le robot acquiert de façon permanente, des informations sur son environnement au moyen d'un système de perception basé sur plusieurs capteurs.

Les problèmes qui se posent sont alors :

- Comment partager les actions communes, et comment déterminer les buts partagés ?
- Comment éviter ou gérer les conflits tout en assurant une solution cohérente ?
- Comment partager les connaissances ?

Dans cet article, nous présentons une étude du comportement collectif des robots dans leur environnement mais aussi de leur

collaboration pour accomplir des tâches qu'ils ne sauraient exécuter individuellement.

La nature de notre problématique nous a suggéré alors, de faire appel à un certain nombre d'approches et techniques :

- Les systèmes multi-agents comme une philosophie globale de résolution.
- La logique floue comme une technique de contrôle.
- Les algorithmes génétiques comme outil ou méthode d'optimisation.

## II- Structure du modèle multi-agents adopté

L'analyse présentée ici nous amène à proposer naturellement pour des raisons de souplesse et de modularité, une approche basée sur les systèmes multi-agents privilégiant à la fois la distribution de décision, l'autonomie et la réactivité des agents, ainsi que la coordination des actions en cas de conflits. Pour ce faire, nous avons adapté des agents robots autonomes de type multi-niveaux de structure horizontale pour la résolution du problème.

Chaque robot a une mission spécifique au sein du système robotisé et doit posséder un niveau substantiel d'autonomie à travers des capacités d'ordonnement : autonomie d'opération, perception de son environnement, communication avec les autres robots, etc. L'approche multi-agents adoptée propose d'une

part, de distribuer le contrôle et les connaissances nécessaires à la résolution d'un problème entre un ensemble d'entités appelées agents, et d'autre part, elle permet de traiter des problèmes complexes par coopération entre plusieurs agents.

### II-1 L'environnement

Même si l'environnement considéré ne contient pas d'obstacles, il reste tout de même un environnement dynamique et les seuls obstacles qui le constituent sont :

- La cible à atteindre.
- Les autres robots.

### II-2- Hypothèses du problème

Le robot est affronté aux différents événements qui nécessitent des traitements spéciaux. Pour cela, on conçoit un ensemble de robots naviguant dans un environnement inconnu vérifiant les hypothèses suivantes :

Les obstacles dynamiques sont les autres agents-robots en circulation ou en panne momentanément.

Chaque agent doit être capable tout seul à planifier lui-même et de suivre ses propres chemins, lui permettant de se rapprocher davantage de la cible et d'occuper une position appropriée au sein du groupe pour traquer efficacement la cible.

L'environnement dans lequel activent les agents-robots est supposé rectangulaire, plan et libre(*sans obstacles*).

### II-3 Description de l'architecture proposée : [1]

L'architecture adoptée basée sur les systèmes multi-agents pour la navigation des robots mobiles est composée des modules suivants(Figure -1) :

\* **Base de connaissance** : elle permet de stocker toutes les informations utiles à la navigation (description de l'environnement, adresses et positions des autres agents, etc..)

\* **Module de perception** : la fonction de perception permet à l'agent-robot d'une part, de percevoir l'état de son environnement (les autres agents, la cible) et d'autre part, de percevoir ses états internes.

\* **Module de communication** : Il permet l'échange d'informations utiles entre les agents-robots.

\* **Module d'action** : les capacités de raisonnement de l'agent constituent le centre de décision des différentes compétences de l'agent. En fonction du comportement déclenché, un module d'action associé à un comportement donné détermine une réponse adaptée à la situation dans laquelle se trouve l'agent-robot. La réponse de l'agent robot est fonction du savoir-faire et de la granularité du comportement. Dans notre application, ce module se compose des comportements suivants :

- Sélection chemin : il permet de trouver un chemin, plus ou moins optimal pour rattraper la cible
- Eviter obstacle : il permet d'éviter les obstacles(les autres agents) qui se trouvent dans le chemin de l'agent-robot, en suivant le trajet sélectionné.
- Arrêter : il permet d'arrêter l'agent-robot à un point donné.

\* **Module de réaction** : c'est un module superviseur qui permet de contrôler l'action globale de l'agent-robot.

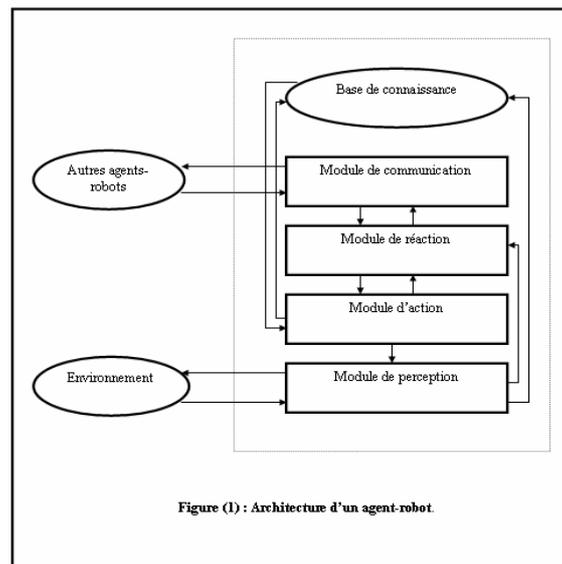


Figure (1) : Architecture d'un agent-robot.

#### II-3-1 Base de connaissance

Pour assurer la communication et l'évitement de certaines interactions. Le robot doit avoir certains informations. Pour cela, l'architecture de contrôle proposée est dotée d'une structure de stockage appelée base de connaissance, et qui contient les informations suivantes : les adresses de tous les agents, la position de

l'agent-robot lui-même, la position relative par rapport à la cible et aux autres agents et les Fonctions d'appartenances d'E/S du systèmes flous.

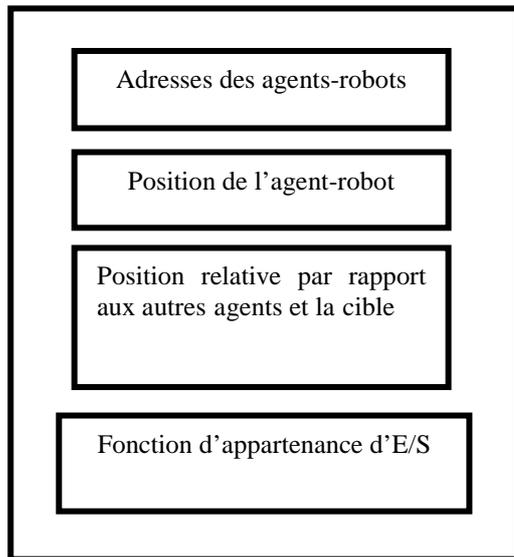


Figure.2 : La base de connaissance de l'agent-robot

### II-3-2 Module de perception :

Le module de perception assure une fonction double :

- Détecter la présence des obstacles (les autres agents, la cible, des agents tombant en panne ) a l'aide de capteurs.
- Tester la nouvelle position de l'agent-robot, qui se trouve dans la base de connaissance.

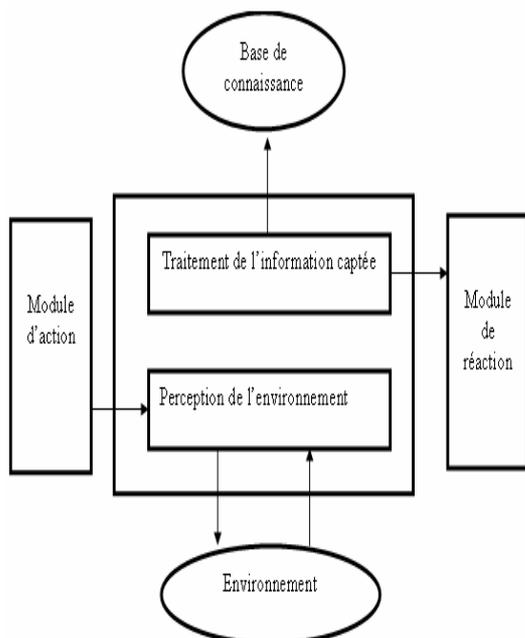


Figure.3 : Structure de module perception  
Le module de perception de l'agent extrait les informations de l'environnement, et les

stocke dans la base de connaissance. Ces informations sont utilisées ensuite, par les autres modules.

### II-3-3 Module de communication :

Le module de communication est composé (figure .4 ) de :

1. File d'attente des messages émis : elle permet de stocker les messages que l'agent veut envoyer aux autres agents.
2. File d'attente des messages reçus : elle contient les messages envoyés par les autres robots.
3. D'un processus itératif : Ce processus (appelé processus de communication ) permet de vérifier continuellement la présence des messages dans les deux files de module de communication, afin de les traiter, et envoyer le résultat aux autres agent-robots (pour les messages à émettre) ou au module de réaction (pour les messages reçus).

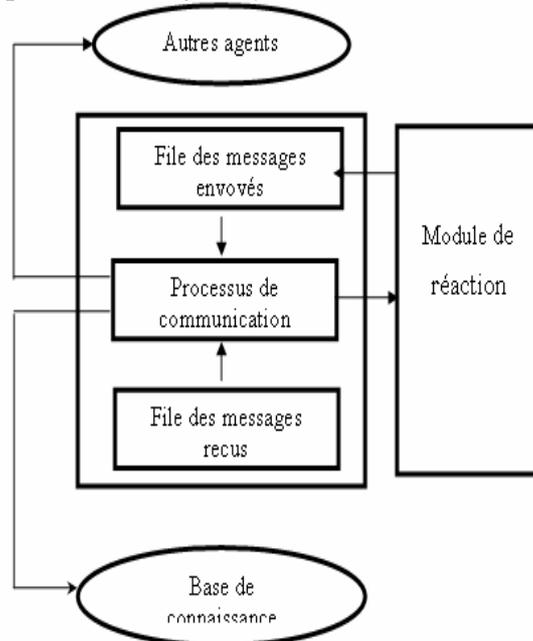


Figure .4 : module de communication

### II-3-4 Module de réaction

Ce module représente le noyau de cette architecture de contrôle. Il permet à l'agent-robot de réagir aux différents événements internes et externes en déclenchant un comportement correspondant à l'événement capté selon le mode stimulus/réponse (Figure.5).

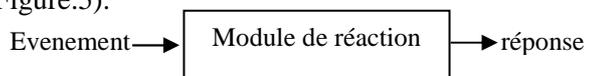


Figure .5 : Le Module stimulus/réponse du module de réaction

Le module de réaction effectue les tâches suivantes :

- Répondre aux messages qui étaient envoyés par les autres agents-robots : le Modules de réaction doit répondre aux messages des autres agent-robots, et qui sont transmis par le module de communication.

- Traiter les événements signalés par le module de perception: Lorsque l'agent-robots rencontre un nouvel obstacle, il doit avertir le module de réaction pour réagir aux changements de l'environnement. Cette réaction s'effectue en désactivant un comportement et activant un autre.

- Envoyer des informations : Si l'agent-robots tombe en panne, il doit informer les autres de son état en transmettant cette information au module de communication afin de constituer les messages à envoyer. L'utilisation d'un module réactif permet de donner à l'agent-robot une meilleure réaction devant les changements de l'environnement.

### II-3-4 Module d'action :

C'est une collection de comportements prédéfinis (figure.6), tel que déplacement, arrêt, évitement obstacle. Ces comportements sont nécessaires pour le déplacement d'un agent-robot.

Un comportement est défini comme évolution particulière observée d'un agent-robot dans son monde réel(environnement, autres agents). Il correspond à un mécanisme reliant perception/communication – raisonnement/décision-action de l'agent et son monde réel.

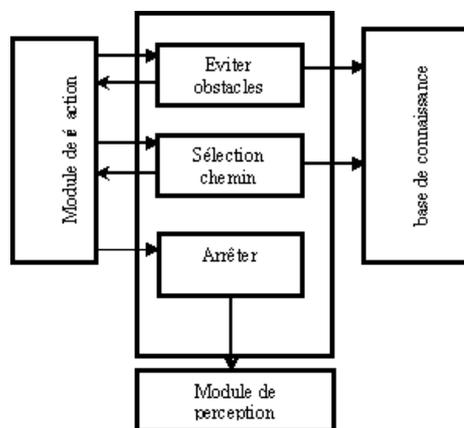


Figure .6 : Module d'action et ses Interactions

**II-3-4-1 Comportement sélection chemin :** Il joue le rôle d'un contrôleur qui surveille le déplacement de l'agent-robot. ce comportement ne s'exécute que lorsque le module de réaction veut déplacer.

Le fonctionnement de ce comportement nécessite un chemin de déplacement qui est déterminé par : Le concepteur et L'agent-robot lui-même.

#### a) Structure du contrôleur flou : [2]

Le contrôleur flou développé est une version classique ou traditionnel, utilisé pour les processus de type multi-entrées, multi-sorties. Une fois que les entrées sont connues(entrées capteurs), les sorties sont déterminées de manière unique. Les contrôleurs de ce type utilisant l'approche classique dite Mamdani[Mam.74] dans laquelle chaque règle est représentée par une conjonction(minimum) et l'agrégation des règles par une disjonction(maximum). La figure (IV.7) donne le principe de conception de notre contrôleur.

#### - Variables linguistique :

La configuration du contrôleur est donnée par :

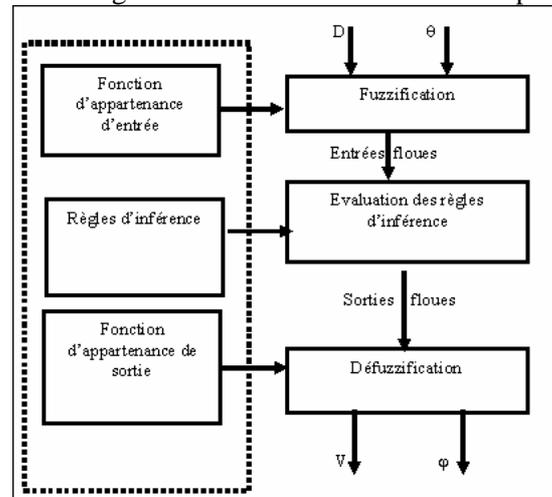


Figure .7 : configuration du contrôleur flou

Pour le contrôleur conçu, nous avons choisi comme variables d'entrées, la distance captée robot-cible et l'angle entre la verticale passant par l'extrémité de la cible et le point milieu du robot. L'univers de discours est réparti en sous ensemble flous, figure.8-a pour l'angle ( $\theta$ )et la figure.8-b pour la distance ( $D$ ). Et comme variables de sortie, la vitesse de déplacement( $V$ ),voir figure.9-a et l'angle de commande ( $\varphi$ ) figure.9-b.

Les fonctions d'appartenance de ses variables sont de forme triangulaire, où la somme des degrés d'appartenance de chaque variable aux

différents sous ensembles flous est toujours égal à l'unité.

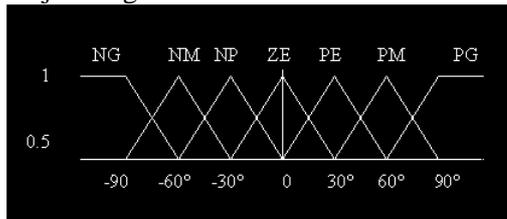


Figure.8-a : répartition floue de la variable angle  $\theta$

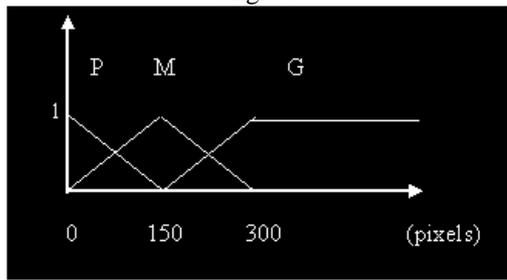


Figure .8-b : répartition de l'univers de discours de la variable Distance D

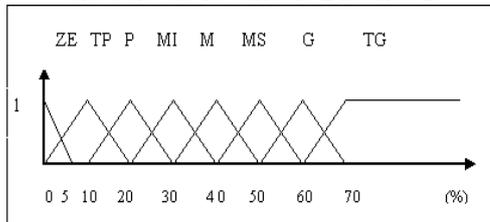


Figure .9-a : Répartition de l'univers du discours de la variable de sortie (V)

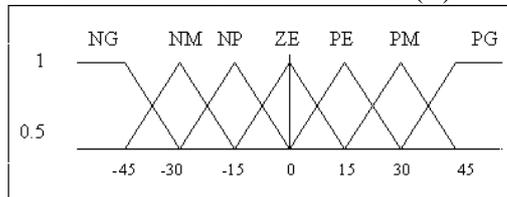


Figure .9-b : Répartition de l'univers de discours de la variable de sortie ( $\varphi$ )

### b) Contrôleur flou optimisé : [3]

Pour optimiser le contrôleur flou on a utilisé Un algorithme génétique.

\* **Codage** : Cette étape consiste à coder la table de décision suivante :

<b>D</b>	$\theta$	NG	NM	NP	ZE	PP	PM	PG
P		NM	NM	NM	ZE	PP	PM	PM
		P	TP	ZE	ZE	ZE	TP	P
M		NM	NP	PP	PM	PM	PG	PG
		G	MS	M	MI	M	MS	G
G		NM	NP	ZE	PP	PM	PM	PG
		TG	TG	G	MS	G	TG	TG

Table de décision

D'où le chromosome peut représenter comme suit :

NM	NM	NM	ZE	PP	PM	PM
P	TP	ZE	ZE	ZE	TP	P
NM	NP	PP	PM	PM	PG	PG
G	MS	M	MI	M	MS	G
NM	NP	PP	PM	PM	PG	PG
G	MS	M	MI	M	MS	G

Pour faciliter l'implémentation, on représente chaque prédicat de sortie par un indice :

La sortie angle ( $\varphi$ ) : NG=1, NM=2, NP=3, ZE=4, PP=5, PM=6, PG=7.

La sortie vitesse(V) : ZE =1, TP=2, P=3, MI=4, M=5, MS=6, G=7, TG=8.

### \* Fonction d'adéquation :

Afin d'assurer un bon fonctionnement des AGs, on ne peut que prêter une attention particulière au choix de la fonction d'évaluation (d'adéquation).

La fonction d'adéquation va prendre la forme :

$$f_i = D + V * tg(\theta + \varphi)$$

\* **La sélection** : L'AG attribue à chaque chromosome une chance de participer à la phase de reproduction.

\* **Le croisement** : Une étape du processus des AGs, qui permet de générer les descendants à partir des parents sélectionnés.

\* **La mutation** : Cette étape permet de modifier aléatoirement un gène d'un chromosome, la position du gène est choisie aléatoirement entre 1 et la longueur du chromosome.

### II-4-5-2 Comportement Eviter Obstacle :

Il permet d'éviter les obstacles apparus dans le champ de vision d'un agent, pour cela l'agent doit déterminer un autre chemin(via le comportement sélection chemin d'Obstacle).

Le comportement d'évitement d'obstacle est déterminé par l'agent-robot selon l'environnement dans lequel il évolue et selon sa position par rapport aux obstacles rencontrés.

### II-4-5-3 Comportement Arrêter :

Ce comportement permet d'arrêter le déplacement de tous les agents-robots, en annulant leurs vitesses, lorsque la mission est accomplie.

### III Résultats et simulation :

Les figures ci-dessous montrent les étapes à suivre par les agents-robots depuis la détection du corps étrange jusqu'à la fin de simulation (les agents entourent le corps étrange).

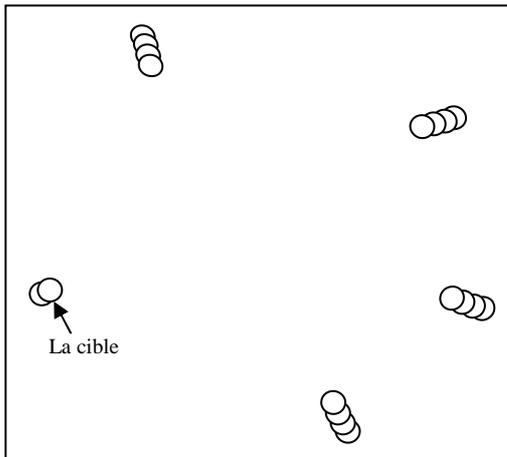


Figure .10 : positions initiales des robots

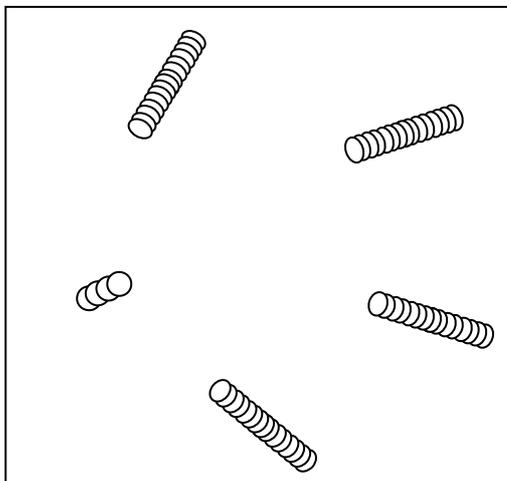


Figure.11 : les robots se dirigent vers la cible

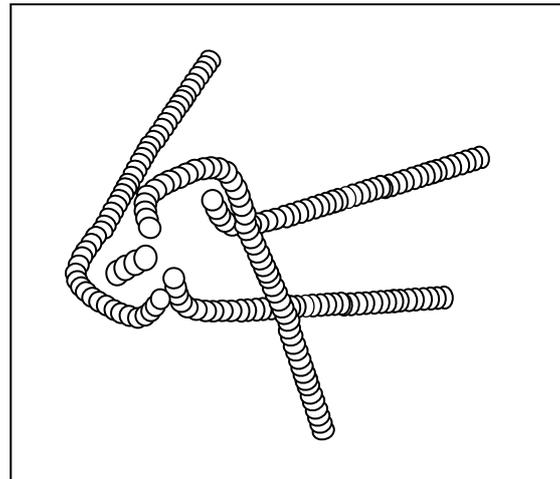


Figure.12 : les robots encerclent la cible sans lui porter atteinte.

### VI- Conclusion

Nous avons essayé de mettre en évidence un système multi-agents sur lequel repose notre étude, dont le but est de diminuer la complexité de navigation d'un ensemble de robots dans un environnement totalement inconnu. Chaque agent est doté de faculté d'autonomie et d'intelligence de réagir aux événements perçus, et de s'interagir dans le groupe.

Pour Ces raisons, nous avons choisit d'adopter pour un agent-robot une architecture multi-niveaux de structure horizontale, cette architecture est composée de quatre principaux modules : module de réaction, module de perception, module d'action et un dernier module pour la communication.

### VII- Bibliographie

- [1]: G. Weiss. "Multiagents Systemes. A Modern approach to distributed artificial intelligence" Edited by MIT press, 1999.
- [2]: C.C.Lee. « Fuzzy logic in control systems », FLC, partII, IEEE 1990 (p :419-435)
- [3]: John. Holland. « Les algorithmes génétiques », Pour la Science, Septembre 1992, n°179.