

Comportement d'évitement d'obstacles basé sur les réseaux de neurones et la logique floue pour un robot mobile autonome

M. Kadri, O. Azouaoui, R. Kezerli et A-H Hariz

Division Productive et Robotique
Centre de Développement des Technologies Avancées
Cité 20 Août 1956 Baba Hassen Alger.
mkadri@cdta.dz azouaoui@hotmail.com

Résumé- Pour garantir une autonomie et une intelligence pour le comportement d'évitement d'obstacles, l'utilisation d'outils tels que les Réseaux de Neurones (RN) ou la Logique Floue (LF) sont nécessaires. Dans cet article, nous présentons une approche d'évitement d'obstacles utilisant ces deux outils. L'approche est validée par une implémentation expérimentale sur le robot mobile Robuter.

I. INTRODUCTION

De nos jours, les chercheurs ont à leur disposition les moyens matériels et logiciels nécessaires pour construire un système dynamique intelligent tel qu'un robot mobile autonome. D'un autre côté, ils possèdent des outils de calcul qui dérivent des technologies émergentes du "soft computing" qui sont beaucoup plus efficaces dans le développement des systèmes intelligents que les méthodes classiques. Parmi ces outils [1-3], on peut citer les Réseaux de Neurones (RN) qui offrent une capacité d'apprentissage permettant une représentation de la connaissance, et la Logique Floue (LF) qui permet de traduire l'expérience humaine en un ensemble de règles.

Par ailleurs, un robot mobile, pour être autonome et intelligent, doit être doté d'un comportement d'évitement d'obstacles qui lui permette d'acquérir la capacité de se déplacer dans son environnement. Pour atteindre cet objectif, les approches classiques ont été remplacées par des approches basées sur le soft computing utilisant en particulier les RN et la LF [4-10].

L'objectif de cet article est donc d'étudier le problème du comportement d'évitement d'obstacles dans un environnement statique et partiellement structuré qui permettrait au robot d'acquérir ce comportement intelligent en utilisant les deux outils cités plus haut. Cette approche est validée par une implémentation expérimentale sur le robot Robuter du Centre de Développement des Technologies Avancées (CDTA). Dans le paragraphe II, le comportement d'évitement d'obstacles est présenté où deux systèmes correspondant aux deux outils considérés sont introduits. Le premier utilise l'apprentissage

par la rétro-propagation du gradient pour réaliser la tâche d'évitement et le second se base sur l'adaptation en utilisant la LF. Ces deux systèmes sont ensuite implémentés sur la plateforme mobile dans le paragraphe III. Les résultats de cette implémentation sont présentés et discutés.

II. COMPORTEMENT D'ÉVITEMENT D'OBSTACLES

Dans ce travail nous considérons uniquement des environnements partiellement structurés tels que les sites industriels avec des obstacles statiques. En effet, l'être humain perçoit les situations spatiales dans de tels environnements comme des situations topologiques : pièces, couloirs, virages à droite et à gauche, etc.

Pour des raisons de simplicité, trois mouvements possibles pour le robot mobile selon trois directions sont considérés dans cet article. A chaque direction est associée une action. Par conséquent, trois actions sont définies, à savoir, " tourner à droite ", " aller tout droit " et " tourner à gauche ". Ces actions sont représentées par le vecteur d'action $A=[A_g, A_a, A_d]$. Pour détecter d'éventuels obstacles, un capteur LMS (Laser Measurement System) [11] est utilisé à l'avant du robot avec une couverture de 180° . De ce fait, dans notre approche le champ du capteur LMS est subdivisé en trois régions qui correspondent aux trois mouvements possibles comme indiqué dans la figure 1.

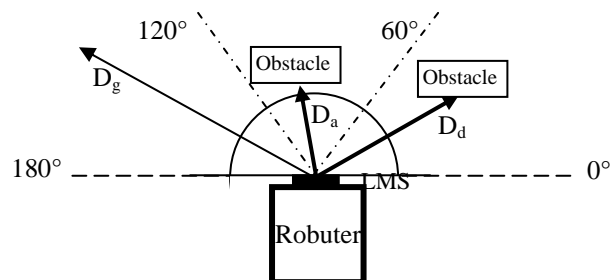


Fig. 1 Robot et Capteur

Durant la navigation, le robot doit construire une carte locale de son environnement (zones libres et obstacles) à partir des mesures fournies par le LMS. La figure (2a) représente un environnement partiellement structuré qui fait ressortir les différentes situations dont lesquelles peut se retrouver le robot mobile. Parmi ces données, la distance considérée correspond à l'obstacle le plus proche au robot dans chacune des trois zones. Ce qui permet d'obtenir un vecteur d'entrée $X = [X_1 \ X_2 \ X_3]^T$. Le vecteur de sortie $O = [O_1 \ O_2 \ O_3 \ O_4 \ O_5 \ O_6]^T$ représente les six situations possibles d'éviter les obstacles comme le montre la figure 2(b). A chaque itération, les distances robot-obstacles D_d , D_a et D_g sont donc recueillies du LMS. Ainsi pour chaque vecteur d'entrée X , le système d'évitement d'obstacles doit fournir au robot la capacité à reconnaître la situation dont laquelle il se trouve afin d'éviter ces obstacles.

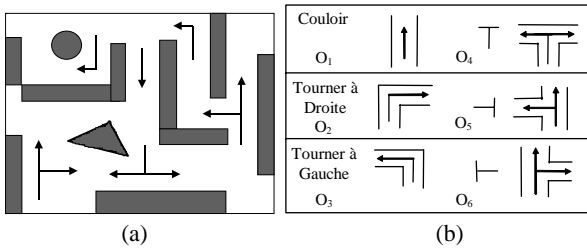


Fig. 2 (a) Environnement partiellement structuré
(b) Situations d'évitement d'obstacles

A. Classificateur Neuronal

Dans ce paragraphe, l'approche considérée est mise en œuvre en utilisant un classificateur neuronal. Le réseau de neurones choisi, comme le montre la figure 3, est un réseau multicouche à une seule couche cachée. Les distances D_d , D_a et D_g subissent un prétraitement afin de construire le vecteur d'entrée X . Ce qui permet de normaliser les données d'entrées pour permettre au classificateur de séparer entre les différentes situations :

$$X_1 = (1/\rho) \exp(-D_g/a), \quad (1)$$

$$X_2 = (1/\rho) \exp(-D_a/a), \quad (2)$$

$$X_3 = (1/\rho) \exp(-D_d/a), \quad (3)$$

Où ρ : norme du vecteur d'entrée X et a : facteur de prétraitement.

La couche d'entrée comporte trois (3) neurones recevant les éléments du vecteur d'entrée X . Les sorties de la couche cachée, avec six (6) neurones, sont obtenues en utilisant le formalisme suivant :

$$net_k = \sum_i X_i W_{2_{ki}}, \quad (4)$$

$$Y_k = f(net_k) \quad (5)$$

Où net_k représente la somme pondérée des données d'entrée par les poids $W_{2_{ki}}$ et Y_k sont les sorties de la couche cachée avec la fonction d'activation f donnée comme suit :

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (6)$$

La couche de sortie du réseau constituée de 6 neurones permet de déterminer la classe correspondant à la situation actuelle :

$$net_j = \sum_i Y_k W_{1_{jk}}, \quad (7)$$

$$O_j = f(net_j) \quad (8)$$

Où les O_j représentent les éléments du vecteur de sortie O et $W_{1_{jk}}$ représentent les poids de la couche de sortie.

Pour acquérir le comportement d'évitement d'obstacles, le réseau de neurones est entraîné en utilisant l'algorithme de la rétro-propagation du gradient sur la base de 106 exemples, un pas d'apprentissage de 0.3 et un facteur de prétraitement $a=1.1$.

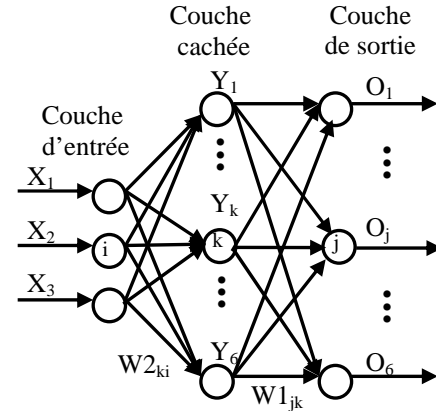


Fig.3 Classificateur Neuronal

B. Contrôleur Flou

Dans ce paragraphe, la même approche a été développée en utilisant un Contrôleur Flou (CF) qui doit permettre au robot d'acquérir le même comportement décrit précédemment.

Les distances D_d , D_a et D_g constituent le vecteur d'entrée du contrôleur (figure 4). Les éléments de ce vecteur possèdent le même degré de valeur qualitative : Près (P), Moyen (M) et Loin (L) définis par les fonctions d'appartenance de la figure 5(a). Les vitesses des roues droite et gauche, V_d et V_g respectivement, sont les sorties du CF dont les variables floues correspondantes sont : Négative Rapide (NR), Négative Moyen (NM), Négative Lente (NL), Zéro (Z), Positive Rapide (PR), Positive Moyen (PM) et Positive Lente (PL) montrées dans la figure 5(b).

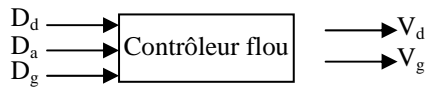


Fig. 4 Contrôleur Flou

La fuzzification est un processus qui permet d'associer les éléments du vecteur d'entrée aux ensembles flous selon un degré d'appartenance comme montré dans la figure 5.

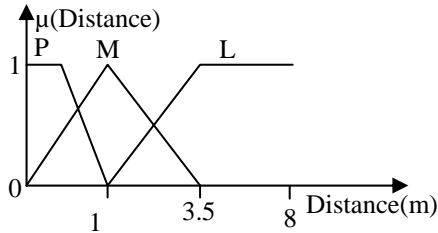


Fig. 5 Fonction d'appartenance des distances

L'inférence est basée sur des opérations Som-Prod afin d'effectuer l'évaluation des règles représentées dans le Tableau 1. La détection de la situation et l'évitement d'obstacle est donc réalisée par un ensemble de règles floues de la forme:

- Si D_d est P et D_a est L et D_g est P Alors $V_d=NR$ et $V_g=NR$ (c-à-d aller devant avec une vitesse rapide)
- Si D_d est L et D_a est P et D_g est P Alors $V_d=NR$ et $V_g=NR$ (c-à-d aller à droite avec une vitesse rapide)

Tableau.1
Matrice d'inférence

D_a	D_g	D_d		
		P	M	L
P	P	$V_r=Z$ $V_g=Z$	$V_r=NM$ $V_g=PM$	$V_r=NM$ $V_g=PM$
	M	$V_r=PM$ $V_g=NM$	$V_r=NM$ $V_g=PM$	$V_r=NM$ $V_g=PM$
	L	$V_r=PM$ $V_g=NM$	$V_r=NM$ $V_g=PM$	$V_r=NM$ $V_g=PM$
M	P	$V_r=NM$ $V_g=NM$	$V_r=NM$ $V_g=NM$	$V_r=NM$ $V_g=NM$
	M	$V_r=NM$ $V_g=NM$	$V_r=NM$ $V_g=NM$	$V_r=NM$ $V_g=NM$
	L	$V_r=NM$ $V_g=NM$	$V_r=NM$ $V_g=NM$	$V_r=NM$ $V_g=NM$
L	P	$V_r=NR$ $V_g=NR$	$V_r=NR$ $V_g=NR$	$V_r=NR$ $V_g=NR$
	M	$V_r=NR$ $V_g=NR$	$V_r=NR$ $V_g=NR$	$V_r=NR$ $V_g=NR$
	L	$V_r=NR$ $V_g=NR$	$V_r=NR$ $V_g=NR$	$V_r=NR$ $V_g=NR$

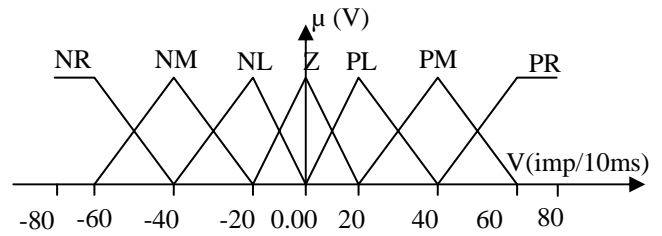


Fig.6 Fonction d'appartenance des vitesses

La défuzzification est utilisée pour fournir au robot une consigne en vitesse des roues droite et gauche, dont la fonction d'appartenance est montrée dans la figure 6, en se basant sur la méthode du centre de gravité.

III. IMPLEMENTATION ET RESULTATS

Pour valider notre approche d'évitement d'obstacles, nous avons effectué sa mise en œuvre expérimentale sur le manipulateur mobile Robuter-ULM du CDTA, montré dans la figure 7, en considérant uniquement la plateforme mobile.



Fig.7 Robuter-ULM

Le robuter est un robot mobile d'intérieur de forme rectangulaire possédant deux roues motrices indépendamment commandables et deux roues folles assurant la stabilité de la plateforme [12]. Le robot est doté d'une ceinture de capteurs à ultrasons et d'un capteur LMS. La base mobile est commandée via une carte microcontrôleur MPC555 de Motorola et un PC embarqué MMX 233 sous environnement Linux redhat 6.1.

Pour chaque système (neural et flou), nous avons effectué deux tests pour vérifier l'efficacité de l'approche. Tous les tests ont été exécutés dans un environnement partiellement structuré

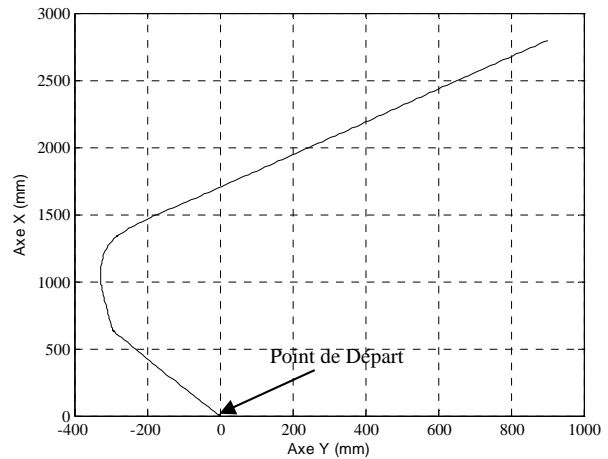
d'intérieur où nous avons dressé des obstacles fixes sur le chemin du robot. Le Robuter durant ces tests n'a aucune destination à atteindre et le seul objectif c'est de se mouvoir dans son environnement tout en évitant les obstacles.

A chaque itération le Robuter récupère les mesures recueillies du LMS et après traitement il détecte la situation dans laquelle il se trouve. Une distance de sécurité de 1 mètre a été établie et c'est sur la base de cette distance que le Robuter peut statuer sur l'absence ou la présence d'obstacle. En absence d'obstacles, le Robuter continue son chemin et dès qu'un obstacle entre dans la zone de sécurité le système d'évitement d'obstacles détecte la nouvelle situation et envoie une autre consigne pour éviter l'obstacle.

Les figures 8 et 9 montrent les résultats obtenus par RN et les figures 10 et 11 montrent ceux obtenus par LF. Le Robuter est confronté à diverses situations et à chacune d'elles il détecte la situation et il effectue les manœuvres pour éviter les obstacles. Dans la situation représentée dans la figure 8 (a) le Robuter rencontre deux obstacles, l'un devant et l'autre à droite. Le RN identifie qu'il s'agit d'un virage à gauche. Le Robuter comme le montre la figure 8(b) effectue un angle de braquage et dès que l'obstacle sort du champ du capteur LMS il se déplace en ligne droite. Ensuite le Robuter rencontre le deuxième obstacle qui occupe un espace entre deux zones. Ceci se traduit par un virage à droite.



(a)



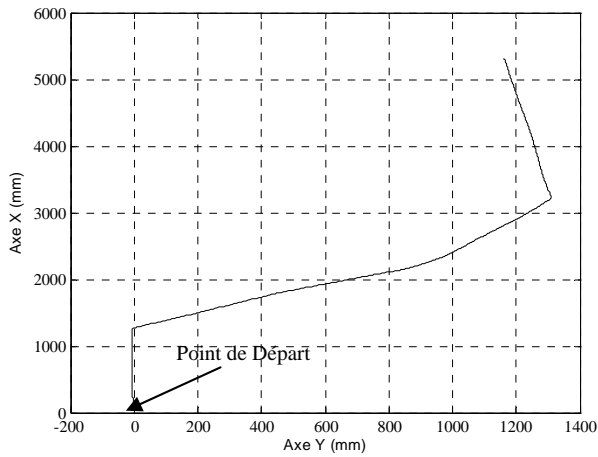
(b)

Fig. 8. (a) : Environnement expérimental 1
(b) : Trajectoire effectuée par le Robuter en utilisant les RN

Dans la situation de la figure 9(a) le robot démarre dans un environnement libre ce qui se traduit par une ligne droite comme le montre la figure 9(b) et dès que l'obstacle entre dans la zone de sécurité le robot entame un virage à droite. Le robot continue à se déplacer en ligne droite jusqu'à ce qu'il rencontre sur sa droite un obstacle qu'il a pu éviter avec un léger changement de direction. Enfin en rencontrant le mur, le robot dévie à gauche et longe le mur tout en gardant une distance de sécurité.

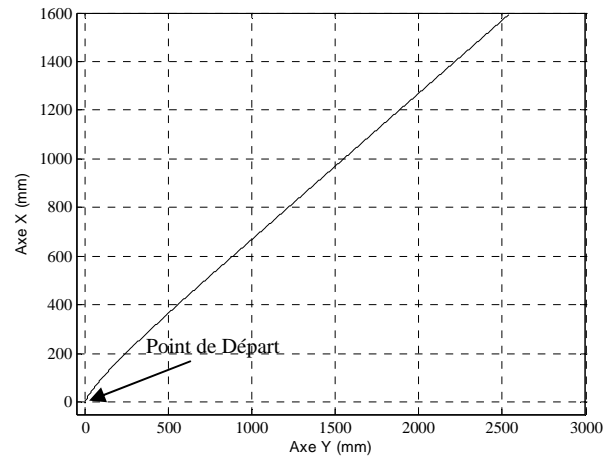


(a)



(b)

Fig. 9. (a) : Environnement expérimental 2
(b) : Trajectoire effectuée par le Robuter en utilisant les RN



(b)

Fig. 10. (a) : Environnement expérimental 3
(b) : Trajectoire effectuée par le Robuter en utilisant la LF

Afin de valider l'approche par LF, nous avons effectué deux tests. Le premier est représenté par la figure 10(a). Dès le départ, le robot rencontre un obstacle sur son chemin. Comme le montre la figure 10(b) le robot effectue une déviation sur la droite puis il continue à se déplacer tout droit sans que le deuxième obstacle le perturbe puisqu'il n'entre pas dans la zone de sécurité.

Pour la deuxième situation comme montrée dans la figure 11(a) et 11(b), le robot démarre dans un environnement libre ce qui lui permet d'aller tout droit. A chaque fois qu'il rencontre un obstacle, il l'évite tout en gardant comme dans les précédents tests une distance de sécurité qui garantit un déplacement sans danger au robot.



(a)



(a)

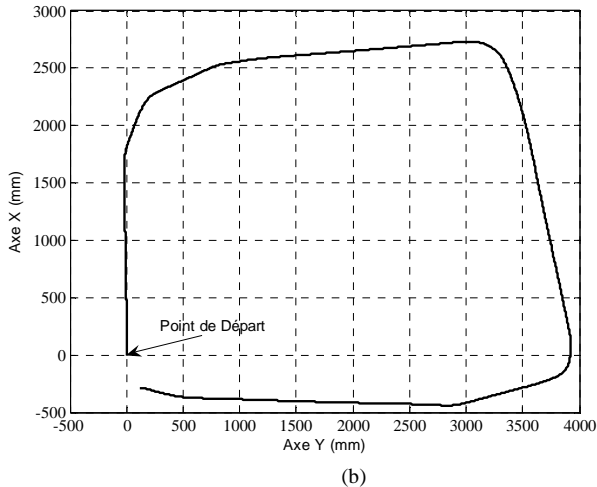


Fig. 11. (a) : Environnement expérimental 4
(b) : Trajectoire effectuée par le
Roberuter en utilisant la LF.

IV. CONCLUSION

Dans cet article, nous avons présenté une approche d'évitement d'obstacles résolue d'une part en utilisant les RN et d'autre part en utilisant la LF. Les deux systèmes implémentés sur le Robuter ont donné des résultats satisfaisants qui permettent de doter le robot d'un certain degré d'intelligence. Cependant, il reste quelques améliorations à apporter pour augmenter les performances des deux systèmes. Premièrement, la fusion entre les mesures du LMS et celles des capteurs ultrasons pourrait améliorer la perception de l'environnement. En effet, dans certains tests lorsqu'un obstacle est assez proche du robot et en dehors de la zone de couverture du LMS, le robot peut heurter cet obstacle s'il opère une manœuvre pour éviter un autre. Deuxièmement, ajouter une action de marche arrière afin de sortir des situations où le robot se retrouve bloqué est à prévoir.

BIBLIOGRAPHIE

- [1] B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice Hall, 1992.
- [2] P.J. Werbos, "Neurocontrol and fuzzy logic: Connections and designs," *Int. J. of Approx. Reaso.*, vol. 6, pp. 185-219, 1992.
- [3] C.C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, Part I and Part II," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 404-435, 1990.
- [4] O. Azouaoui and A. Chohra, "Pattern Classifiers Based on Soft Computing and Their Integration for Intelligent Behavior Control of Mobile Robots", *Proceeding of ICAR 2003, The 11th International Conference on Advanced Robotics, Coimbra, Portugal, June 30- July 3, 2003*, pp. 148-154
- [5] A. Chohra and O. Azouaoui, "Neural Networks (NN) Using Genetic Algorithms (GA) and Gradient Back-Propagation (GBP) for an Intelligent Obstacle Avoidance Behavior", *Proceeding of ICAR 2003, The 11th International Conference on Advanced Robotics, Coimbra, Portugal, June 30- July 3, 2003*, pp. 917-922.
- [6] T. Fukuda, F. Arai, and K. Shimojima, "Intelligent robotic system", *Proc. Int. Multiconf. Computational Engineering in Systems Applications*, France, 1996, pp. 01-10.
- [7] T. Tanaka, J. Ohwi, L.V. Litvintseva, K. Yamafuji, and S.V. Ulyanov, "Soft computing algorithms for intelligent control of a mobile robot for service use Part I and Part II," *Soft Computing*, vol. 1, no. 2, pp. 88-106, June 1997.
- [8] Patrick Reignier, "Pilotage Reactif d'un Robot Mobile Etude du Lien entre la Perception et L'action", *Thèse de Doctorat, Institut National Polytechnique de Grenoble*, 1994.
- [9] H. Surmann, J. Huser and L. Peters, "A Fuzzy System for Indoor Mobile Robot Navigation", *Proc. of the Fourth IEEE Int. Conf. on Fuzzy Systems*, pp. 83 - 86, 20 - 24.03.1995, Yokohama, Japan.
- [10] x. Yang M. Moallem and R.V. Pate1, "An Improved Fuzzy Logic Based Navigation System for Mobile Robots", *Proceedings of the 2003 IEEE/RSJ Int. Conference on Intelligent Robots and Systems Las Vegas, Nevada ' October 2003*, pp. 1709-1717.
- [11] *LMS User Manual*, Sick optic electronic, 1997.
- [12] *Robuter et Bras Ultra Léger, Manuel d'utilisation et de maintenance*, Robosoft 2004.