OPTIMISATION DES CONTROLEURS FLOUS PAR RENFORCEMENT POUR LA NAVIGATION D'UN ROBOT MOBILE

L. Cherroun, M. Boumehraz

Département de Génie Electrique, Université de Biskra - Algérie
E-mail : cherroun_lakh@yahoo.fr medboumehraz@netcourrier.com

Résumé -- La programmation d'un robot mobile est une tâche importante nécessitant une modélisation complète de l'environnement. Dans ce travail, on présente une technique intelligente pour la navigation d'un robot mobile, ne nécessitant qu'un signal scalaire comme information de retour. Au lieu de programmer un robot pour qu'il effectue une mission, on va le laisser apprendre seul sa propre stratégie. Premièrement, l'algorithme de Q-learning de l'apprentissage par renforcement est utilisé pour la navigation d'un robot mobile en discrétisant les espaces d'état et d'action. Afin d'améliorer les performances, une approche d'optimisation des contrôleurs flous par Q-learning et son application pour la tâche de navigation d'un robot mobile est proposée. L'approche consiste à introduire des connaissances à priori dans un système d'inférence flou pour que le comportement initial soit acceptable et d'améliorer les performances en utilisant le Q-learning .

***Mots clés*-- robot mobile, navigation intelligente, apprentissage par renforcement, Q-learning, contrôleur flou.**

I. INTRODUCTION

La navigation d'un robot mobile peut être définie comme la tache de déterminer une trajectoire permettant au robot de se déplacer d'une position initiale vers une autre finale désirée en évitant les obstacles, tout en respectant les contraintes cinématiques du robot et sans intervention humaine. Le problème de déterminer une telle trajectoire est connu aussi sous le nom du problème de planification de trajectoire [1].

L'évitement d'obstacles est l'une des missions de base d'un robot mobile. C'est une tâche importante que doivent posséder tous les robots, puisque ceci permet au robot de se déplacer sans collisions dans un environnement inconnu [2].

Une stratégie de commande possédant la capacité d'apprentissage en ligne peut être réalisée en utilisant l'apprentissage par renforcement. Dans ce cas, le robot reçoit seulement un signal scalaire comme information de retour. Le signal de renforcement permet au navigateur d'ajuster sa stratégie pour améliorer ses performances. C'est une modification automatique du comportement du robot dans son environnement de navigation comme réalisé dans [3][4].

L'apprentissage par renforcement est une méthode de commande optimale, parce qu'on part d'une solution inefficace que l'on améliore progressivement en fonction de l'expérience acquise pour résoudre un problème de décision séquentielle [5][6].

Pour utiliser l'apprentissage par renforcement, plusieurs approches sont possibles. La première consiste à discrétiser manuellement le problème afin de fournir des espaces d'état et d'action finis qui pourront être utilisés directement par des algorithmes d'optimisation pour la construction de la fonction qualité Q représentée sous forme de tableau. Il faut cependant faire attention au choix des discrétisations afin qu'elles permettent un apprentissage correct en fournissant des états et des actions qui contiennent notamment des récompenses cohérentes [6].

La seconde méthode consiste à travailler directement sur les espaces d'état et d'action continus en utilisant des méthodes d'approximation de fonctions. En effet, pour utiliser l'apprentissage par renforcement, il est nécessaire d'estimer correctement la fonction de qualité Q [7]. Cette estimation peut se faire directement par un approximateur universel de fonctions continues comme les réseaux de neurones ou les systèmes d'inférence floue [7]-[10]. L'utilisation de ces approximateurs permet de travailler directement dans l'espace continu et de limiter les effets de parasites qui pourraient apparaitre suite à un mauvais choix des discrétisations [11]-[13].

L'objectif de ce travail est d'appliquer un algorithme d'apprentissage par renforcement, le Q-learning, pour l'optimisation des contrôleurs flous utilisés pour réaliser différentes tâches de navigation pour un robot mobile (recherche d'une cible, suivi de murs,…).

Ce papier est organisé comme suit : la section 2 est un bref exposé sur la méthode d'apprentissage par renforcement, l'algorithme de Q-learning est utilisé pour une tâche de recherche d'une cible par un robot mobile. L'optimisation des contrôleurs flous par renforcement est développée dans la section 3. Cela permet d'améliorer les performances du robot mobile et de travailler avec des

espaces d'état et d'action continus. Enfin, la conclusion donne le bilan de ce travail et présente les perspectives envisagées.

II. L'APPRENTISSAGE PAR RENFORCEMENT

L'apprentissage par renforcement est une méthode d'apprentissage à partir de l'expérience afin de trouver, par un processus essais-erreurs, l'action optimale à effectuer pour chacune des situations rencontrées pour que les récompenses reçues soient maximisées. L'idée fondamentale de l'apprentissage par renforcement est d'utiliser le signal reçu pour améliorer la politique courante après chaque interaction avec l'environnement [6][14].

A l'instant t l'agent (le robot dans notre cas) perçoit la situation s_t de son environnement et agit en exécutant l'action a_t. En conséquence, il reçoit une récompense (r_t) indiquant la qualité de l'action appliquée.

Le but de l'agent, dans le cadre de l'apprentissage par renforcement, est de trouver le comportement le plus efficace pour maximiser l'espérance des gains reçus à la suite de chaque transition (s_t, a_t, r_t, s_{t+1}) [5][6][14], où s_{t+1} est la situation suivante.

Le comportement du robot est défini par une *politique* $\pi : \{S, A\} \rightarrow [0,1]$, notée $\pi(s) = a$. Le but de l'agent est de trouver la politique optimale π^* maximisant les récompenses à long terme en utilisant la fonction valeur utilisée comme mesure de performance [5]. Généralement la fonction valeur est définie pour un processus de décision markovien PDM par :

$$V_\pi(s) = E_\pi(R_t | s_t = s) = E_\pi\left(\sum_{k=1}^{\infty} \gamma^k r_{t+k} | s_t = s\right) \quad (1)$$

Ou $\gamma \in]0,1[$ est un facteur de décroissement qui permet de régler l'importance que l'on donne aux retours futurs par rapport aux retours immédiats.

La plupart des algorithmes d'apprentissage par renforcement utilisent une fonction qualité représentant la valeur de chaque paire état-action pour la détermination du comportement optimal [6][16]. Elle donne pour chaque couple état-action, le retour futur si l'agent suit cette politique π :

$$Q^\pi(s,a) = E_\pi(R_t | s_t = s, a_t = a) \quad (2)$$

La qualité optimale est :

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \quad (3)$$

On obtient alors :

$$Q^*(s,a) = E(r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a) \quad (4)$$

La plupart des algorithmes d'apprentissage par renforcement sont basés sur l'apprentissage par différences temporelles (*TD*) qui est une combinaison des méthodes de Monte Carlo et des méthodes de programmation dynamique. Cette approche permet d'apprendre le comportement optimal directement, sans l'utilisation d'un modèle de l'environnement, par l'évaluation de la qualité des actions utilisées sans avoir besoin d'arriver au but final [6].

A. Le Q-Learning

Le Q-learning est l'algorithme d'apprentissage par renforcement le plus utilisé. Il permet d'apprendre la fonction qualité en interagissant avec l'environnement [14][16], par une mise à jour itérative de la fonction courante $Q^\pi(s_t, a_t)$ à la suite de chaque transition (s_t, a_t, r_t, s_{t+1}). Cette mise à jour se fait sur la base de l'observation des transitions instantanées et de leurs récompenses associées par l'équation suivante :

$$Q(s_t,a_t) \leftarrow Q^\pi(s_t,a_t) + \alpha\left[r_t + \gamma \max_{a \in A(s)} Q(s_{t+1},a_t) - Q(s_t,a_t)\right] \quad (5)$$

Où $\alpha \in [0,1]$ est un coefficient d'apprentissage qui doit diminuer pour tendre vers 0.

La fonction qualité est stockée sous forme d'un tableau: chaque ligne correspond aux qualités des différentes actions pour un état donné. Au début, lorsque la table ne contient pas suffisamment de données, une composante aléatoire est ajoutée de façon à ne pas restreindre les actions éligibles au petit nombre des actions déjà essayées. Au fur et à mesure que la table se remplit, cette composante aléatoire est réduite afin de permettre l'exploitation des informations reçues et d'obtenir une bonne performance.

B. Navigation d'un robot mobile en utilisant le Q-learning

A chaque étape le robot doit définir l'état dans lequel il se trouve, et à partir de cet état, il doit prendre une décision sur l'action à exécuter. En fonction du résultat obtenu lors de l'exécution de cette action, il est soit puni, pour diminuer la probabilité d'exécution de la même action dans le futur, soit récompensé, pour favoriser ce comportement dans les situations pareilles.

Pour une tache de recherche d'une cible par un robot mobile, l'espace autour du robot est divisé en secteurs selon l'angle entre l'orientation du robot et celle de la cible notée *E_ang*, et la distance entre le robot et la cible notée *E_pos* ou l'erreur de position. Les actions délivrées sont : avancer, tourner à droite et tourner à gauche. Ces actions sont choisies par la politique d'exploration-exploitation (*PEE)*. Ce qui permet au robot d'explorer l'espace d'état.

Pendant le déplacement, le robot reçoit les valeurs suivantes comme signal de renforcement:

$$r = \begin{cases} 4, & \text{Si le robot atteint la cible.} \\ 3, & \text{Si } E\_pos \text{ déminue et } E\_ang = 0. \\ 2, & \text{Si } E\_pos \text{ et } E\_ang \text{ déminuent.} \\ -1, & \text{Si } E\_pos \text{ déminue et } E\_ang \text{ augmente.} \\ -2, & \text{Si } E\_pos \text{ augmente.} \\ -3, & \text{Si le robot percute les murs de son environnement.} \end{cases}$$

Afin de généraliser la navigation du robot mobile pour toutes les situations possible, l'apprentissage se fait avec une position initiale aléatoire du robot et de la cible dans chaque épisode.

La figure 1 présente les trajectoires du robot obtenues après la phase d'apprentissage. On remarque que le robot se dirige vers la cible quelque soit sa posotion initiale en exécutant des actions discrètes.

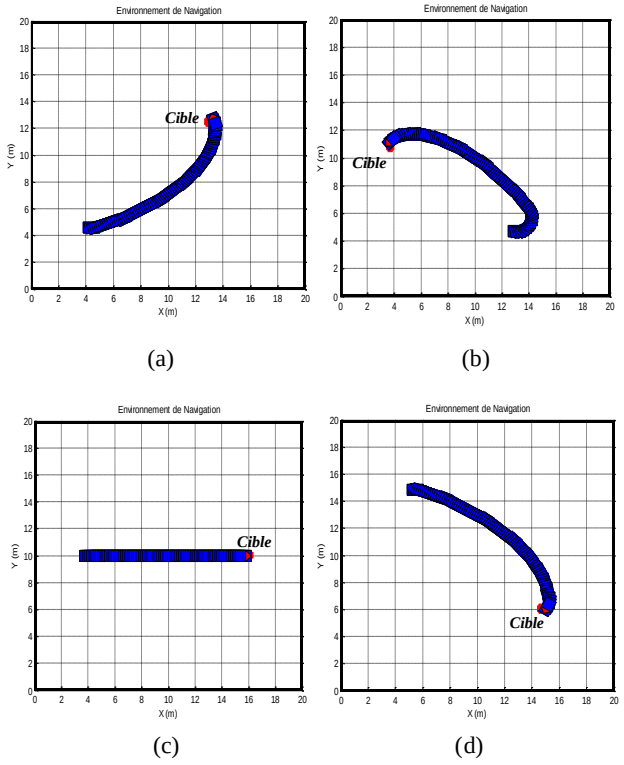


Fig.1. Recherche d'une cible en utilisant le Q-learning

Comme indicateur d'apprentissage, la figure 2 présente la moyenne des récompenses obtenues par le robot à chaque épisode de la phase d'apprentissage. On observe que le comportement s'améliore au cours du temps.

Plusieurs implémentations de l'algorithme Q-learning, ont été utilisées en variant le nombre d'états et d'actions proposées pour obtenir un comportement acceptable. L'augmentation du nombre des couples état-action permet d'améliorer le comportement du robot mais nécessite un volume mémoire plus important et un temps d'apprentissage plus long.

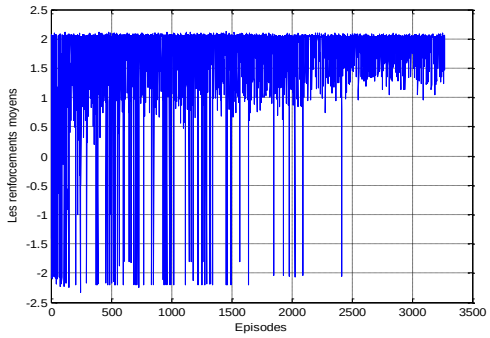


Fig.2. Les valeurs moyennes du renforcement pendant l'apprentissage

Puisque l'utilisation du Q-learning exige le stockage des valeurs de qualité pour tous les couples (état, action) on peut utiliser des tableaux dans les problèmes discrets de faible dimension. Mais dans le cas des espaces d'état et d'action continus comme la tâche de navigation d'un robot mobile, le nombre de situations est infini et la représentation de la fonction Q par des tableaux est impossible. Les approximateurs universels, comme les systèmes d'inférence floue, offrent des solutions prometteuses pour l'approximation des valeurs d'utilité [9][10][18].

Dans le but d'améliorer les performances du robot mobile, on utilise des contrôleurs flous optimisés par renforcement et caractérisés par la possibilité de l'introduction des connaissances disponibles à priori pour que le comportement initial soit acceptable.

III. OPTIMISATION DES SYSTEMES D'INFERENCE FLOUES EN UTILISANT L'APPRENTISSAGE PAR RENFORCEMENT

L'utilisation des systèmes d'inférence floue est une solution prometteuse pour la représentation des fonctions qualité avec des espaces d'état et d'action continus [15][18][19]. La tâche consiste à approcher la fonction qualité Q par la fonction :

$$s \rightarrow y = \hat{Q} = SIF(s) \qquad (6)$$

Le principe de cette optimisation consiste à proposer plusieurs conclusions pour chaque règle et à associer à chaque conclusion une valeur qualité qui sera mise à jour incrémentalement au cours du temps. Le processus d'apprentissage permet de déterminer l'ensemble des règles maximisant les renforcements futurs [10][18][19]. Cette version floue du Q-learning est appelée le Q-learning flou.

La base des règles initiales en utilisant un modèle flou de type Takagi-Sugeno d'ordre 0 est composée de m règles. Chaque règle possède N conclusions de la forme [9][17][19]:

Si s est S_i Alors $y = a[i,1]$ avec $q[i,1] = 0$
ou $y = a[i,2]$ avec $q[i,2] = 0$ (7)
…
ou $y = a[i,N]$ avec $q[i,N] = 0$

où $q(i,j)$, $i = 1..m$ et $j = 1...N$, sont des solutions potentielles dont la valeur est initialisée à 0. Durant l'apprentissage, la conclusion de chaque règle est choisie au moyen d'une politique d'exploration-exploitation notée (*PEE*) ou $PEE(i) \in \{1...N\}$. Dans ce cas, la sortie inférée est donnée par :

$$A(s) = \sum_{i=1}^{m} w_i(s).a[i, PEE(i)] \quad (8)$$

et la qualité de cette action est :

$$\hat{Q}(s, A(s)) = \sum_{i=1}^{N} w_i(s).q[i, PEE(i)] \quad (9)$$

A. Navigation d'un robot mobile en utilisant le Q-learning flou

L'application se résume en l'implémentation d'un contrôleur flou pour la navigation d'un robot mobile. La base de règles est améliorée en ligne en utilisant un signal de renforcement. Dans cet algorithme, le robot possède une base de règles initiale qui définit les situations possibles pour la tâche à accomplir.

Pour une tache de recherche d'une cible. Le contrôleur utilise l'angle entre l'orientation du robot et la cible (*E_ang*) et la distance entre la position du robot et celle de la cible (*E_pos*) pour générer l'angle de braquage α. Les fonctions d'appartenance pour *E_pos* et *E_ang* sont données par les figures 3 et 4.

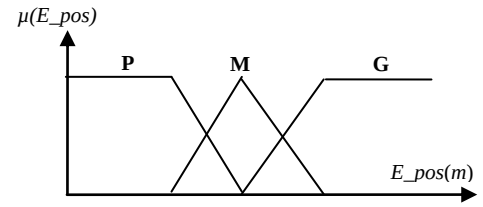


Fig.3. Les fonctions d'appartenance pour E_pos

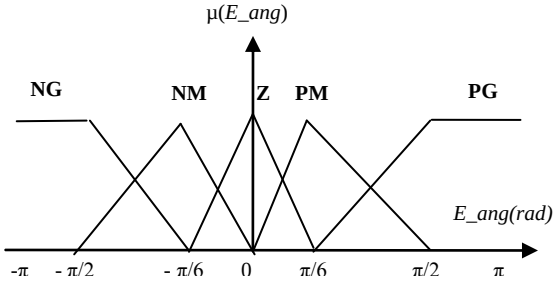


Fig.4. Les fonctions d'appartenance pour E_ang

Avec les variables linguistiques suivantes :

P : Proche **M :** Moyenne **G :** Grande
Z : Zéros **PM** : Positive Moyenne **PG** : Positive Grande
NG : Négative grande **NM** : Négative Moyenne

Pendant la navigation, le robot reçoit les mêmes valeurs de renforcement utilisées dans la section précédente.

Durant l'apprentissage et afin d'optimiser le système de navigation utilisé, les positions initiales sont choisies aléatoirement. Chaque épisode commence avec une position aléatoire et se termine lorsque le robot atteint la cible ou percute les limites de son environnement.

Pour chaque règle, 3 conclusions sont proposées. Après la phase d'apprentissage, le robot choisit pour chaque règle la conclusion correspondante à la meilleure qualité $q[i,j]_{j=1}^{N}$.

Les trajectoires du robot, après apprentissage en utilisant le Q-learning flou et les traces d'éligibilités, sont représentées sur la figure 5. Dans tous les cas, le robot s'oriente vers la cible quelque soit sa position initiale en exécutant des actions continues. La figure 6 montre la variation des valeurs moyennes de renforcement pendant la phase d'apprentissage. On observe que le comportement du robot s'améliore au cours de l'apprentissage et que l'apprentissage est plus rapide que celui en utilisant le Q-learning standard.

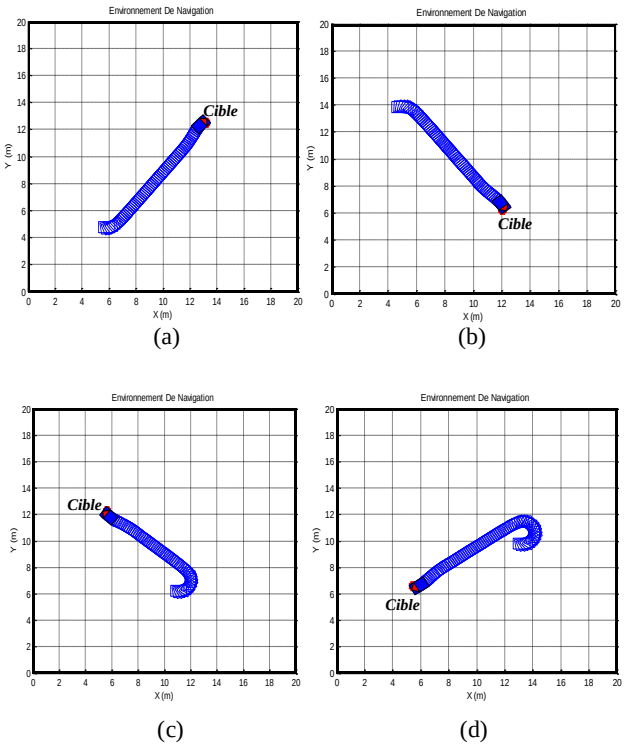


Fig.5. Recherche d'une cible en utilisant le Q-learning flou

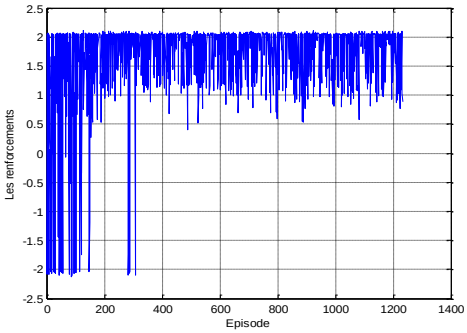


Fig.6. Evolution de la moyenne des récompenses

B. Suivi de mur avec des connaissances imprécises

Pour une mission de suivi de murs, on applique l'algorithme de Q-learning flou sur un système flou avec des connaissances imprécises en proposant des interprétations numériques pour les variables linguistiques de sortie. Le contrôleur flou utilise comme variables :

- Les entrées sont les distances de l'obstacle dans les trois directions (en face, à droite et à gauche).
- Les sorties sont l'angle de braquage et la vitesse de déplacement du robot.

Pour simplifier la stratégie de navigation, on utilise deux ensembles flous pour les distances de l'obstacle dans les trois directions du robot mobile (figure 7), où: **P** : proche et **L** : loin

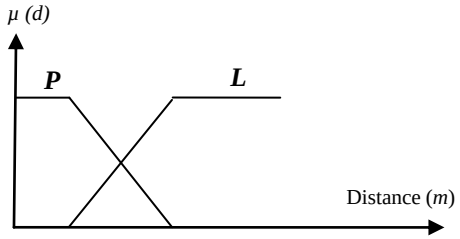


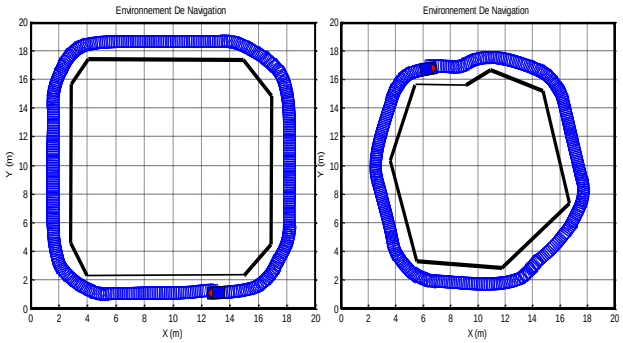
Fig.7. Les fonctions d'appartenance de la distance

Premièrement, la politique adoptée pour cette mission est exprimée symboliquement par les règles d'inférences présentées dans la table 1.

TABLE I
Les règles d'inférence du suivi de mur

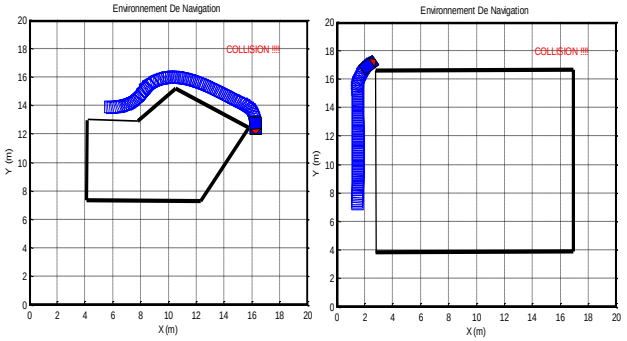
Angle de braquage / V_r			*distance* d_a			
			P		**L**	
			distance d_f			
			P	**L**	**P**	**L**
distance d_d	**L**	*A*	*NG*	*NP*	*PG*	*NM*
		V_r	*Z*	*M*	*Z*	*M*
	P	*A*	*PG*	*Z*	*PG*	*PP*
		V_r	*Z*	*M*	*P*	*P*

Les résultats obtenus en utilisant ce contrôleur flou sont donnés sur les figures 8 et 9.



(a) (b)

Fig.8 Suivi de murs par le contrôleur flou



(a) (b)

Fig.9 Collision avec les obstacles

Le contrôleur flou utilisé donne des résultats acceptables comme le montre la figure 8. Mais dans les cas ou l'obstacle contient des pointes (coins), le comportement est mauvais et le robot ne peut pas éviter les collisions (figure 9).

Pour résoudre ce problème, la base des règles est améliorée en ligne en utilisant un signal de renforcement *r* définit par :

$$r = \begin{cases} -2, & \text{Si le robot percute un obstacle,} \\ -1, & \text{Si } d_i < d_m,\ i = 1...3, \\ 0, & \text{ailleurs.} \end{cases}$$

Ce signal va servir pour déterminer la meilleure interprétation numérique des termes linguistiques utilisés, en proposant trois interprétations pour chaque label de sortie (angle de braquage).

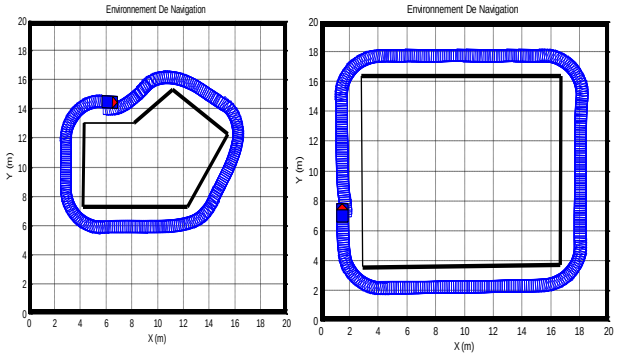
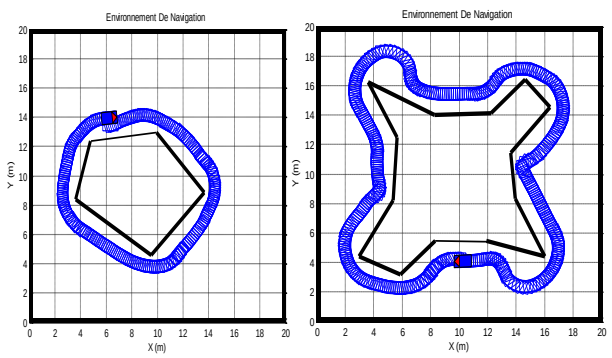


Fig.10. Les comportements précédents améliorés



(a) (b)

Fig.11. Suivi des murs de différentes formes

Les figures 10 et 11 montrent des exemples du résultat obtenu après optimisation. On observe que le robot est capable d'évoluer dans son environnement sans collision avec les obstacles

IV. CONCLUSION

Dans ce travail, on a présenté une technique intelligente pour la navigation d'un robot mobile. Cette technique est basée sur l'optimisation des contrôleurs flous en utilisant le Q-learning. C'est une stratégie de commande avec une capacité d'apprentissage en ligne, basée sur la modification automatique du comportement du robot dans son environnement de navigation pour maximiser ses récompenses. Le Q-learning est un outil performant pour obtenir un comportement optimal qui ne nécessite qu'un signal scalaire comme information de retour indiquant la qualité de l'action appliquée. Dans ce travail on a utilisé une extension du Q-learning, appelée Q-learning flou, pour la navigation d'un robot mobile.

Le principe du Q-learning flou consiste à optimiser les conclusions des règles floues en utilisant un signal de renforcement. Le Q-learning flou combine les avantages de l'apprentissage par renforcement et le contrôle flou et peut être considéré, d'une part comme une méthode d'optimisation des systèmes d'inférences floue, et d'autre part comme une extension naturelle de l'algorithme Q-learning standard aux systèmes à espaces d'état et d'action continus.

Comme perspective à ce travail, on envisage d'améliorer le comportement du robot par la commande de la vitesse de déplacement et d'appliquer cet algorithme pour la commande d'un robot réel. En plus, Il serait très intéressant d'étendre cette technique pour les systèmes flous de type Takagi-Sugeno d'ordre 1 ou de Mamdani.

REFERENCES

[1] P. Reignier, *Pilotage Réactif d'un Robot Mobile, Etude du Lien entre la Perception et l'Action*, Thèse de Doctorat, Institut National Polytechnique de Grenoble (INPG), 1994.

[2] F. Cuesta, A. Ollero, *Intelligent Mobile Robot Navigation*, Springer-Verlag, Berlin Heidelberg, 2005.

[3] L. M. Zamstein, A. A. Arroyo, E. M. Schwartz, S. Keen, B. C. Sutton, and G. Gandhi , "Koolio : Path Planning using Reinforcement Learning on a Real Robot Platform, " in *Proc. 2006, FCRAR, 19th Florida Conference on Recent Advances in Robotics, , Miami, Florida, May 25-26*.

[4] W. D. Smart and L. P. Kaelbling "Effective Reinforcement Learning for Mobile Robots," in *Proc. 2002, ICRA'02, IEEE International Conference on Robotics & Automations, vol. 04, pp. 3404-3410* .

[5] R. S. Sutton, A. G. Barto and R. J. Williams, "Reinforcement Learning is Direct Adaptive Optimal Control," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 19-22, 1992.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

[7] S. T. Hagen and B. Krose , "Q-Learning for Systems with Continuous State and Action Spaces," in *Proc. 2000, BENELEARN, 10th Belgian-Dutch Conference in Machine Learning*.

[8] Claude Touzet, *L'Apprentissage par Renforcement*, Paris:Masson, 1999.

[9] P. Y. Glorennec, *Algorithmes d'Apprentissage Pour Systèmes d'Inférence Floue*, Paris: Hermes, 1999.

[10] M. Boumehraz , K. Benmahammed, M. L Hadjili and V. Wertz, " Fuzzy Inference Systems Optimization by Reinforcement Learning," *Courrier du Savoir* , no. 01, pp. 09-15, 2001.

[11] H. N. C. Yung and C. Ye, "An Intelligent Mobile Vehicle Navigation Based on Fuzzy Logic and Reinforcement Learning". *IEEE Trans. Syst., Man and Cybern. B*, vol 29. no.2, pp.314-321, 1999.

[12] C. Ye, N. H.C. Yung and D. Wang, "A Fuzzy Controller with Supervised Learning Assisted Reinforcement Learning Algorithm For Obstacle Avoidance" *IEEE Trans. Syst., Man, and Cybern. B*, vol. 33, no.1, pp.17-27, 2003.

[13] P. Rishikesh, L. Janardhanan and E. M. Joo, "Goal Seeking of Mobile Robots using Dynamic Fuzzy Q-learning " *Journal of the Institution of Engineers*, vol. 45 no. 5, pp. 62-76, 2005.

[14] L. P. Kaelbling, M. L. Littman, and A.W. Moore, "Reinforcement Learning: A Survey", *Journal of Artificial Intelligence Research*, vol. 04, pp.237-285, 1996.

[15] P.Y. Glorennec, " Reinforcement Learning: an Overview," in *Proc. 2000, ESIT'2000, European Symposium on Intelligent Techniques*, pp. 17-35.

[16] C. Watkins and P. Dayan , "Q-Learning", *Machine Learning*, vol. 8, pp. 279-292, 1992.

[17] K. M. Passino and S. Yurkovich, *Fuzzy Control*, Menlo Park:Addison Wesley, 1998.

[18] M. Sc. Mykhaylo Konyev , "Using fuzzy Inference System as a Function Approximator of a State Action Table," *Advanced Aspects of Theoretical Electrical Engineering*, Sozopol, Bulgaria, 2005.

[19] P. Y. Glorennec, and L. Jouffe, "Fuzzy Q-learning," in *Proc. 1997, FUZZ-IEEE'97, 6th IEEE International Conference on Fuzzy Systems*, pp. 659-662.