

Implémentation d'un Contrôleur Flou pour la Navigation d'un Robot Mobile de type Voiture

N. Ouadah, O. Azouaoui, M. Hamerlain

Division Productique et Robotique, Centre de Développement des
Technologies Avancées, BP 17, Baba Hassen, Alger, Algérie
ouadah@yahoofr

Résumé : Ce travail traite du problème de la commande de mouvements d'un robot mobile de type voiture à double braquage. Des algorithmes permettant le déplacement du robot d'une position initiale vers une quelconque destination désirée, tout en respectant ses contraintes cinématiques, ont été développés puis implémentés. D'une part, la localisation est faite en temps réel par « Odométrie », en employant les données des capteurs incrémentaux du robot. D'une autre part, un contrôleur basé sur la logique floue envoie les consignes de vitesse et de braquage au robot pour assurer sa convergence vers le but. Une simulation de ce régulateur est présentée. Enfin, l'implémentation temps réel a été réalisée sur le robot « Robucar ». Les résultats obtenus en simulation et en expérimentation sur site réel sont exposés et comparés.

Mots Clés : Robucar, Localisation, Odométrie, Commande par Logique Floue.

1 INTRODUCTION

Suite aux problèmes d'encombrement et surtout de pollution liés à l'utilisation massive des voitures, particulièrement dans les grandes villes, des efforts considérables sont déployés dans le domaine de la recherche en robotique en vue de mettre au point un nouveau moyen de transport fiable, sécurisé et moins polluant.

La plateforme expérimentale employée dans ce travail est un fruit de ces efforts. En effet, Dans le cadre de la route automatisée, l'INRIA a imaginé un système de transport original de véhicules en libre-service pour la ville de demain. Ce système de transport public est basé sur une flotte de petits véhicules électriques spécifiquement conçus pour les zones où la circulation automobile doit être fortement restreinte. Pour tester et illustrer ce système, des prototypes, nommés « Cycab », ou « Robucar » ont été réalisés [Baille 1999].

Plusieurs travaux de recherche ont traité le problème de la commande de robot de type voiture. Différents robots d'intérieur, généralement de type holonome et ayant un découplage entre la translation et la rotation, ces robots ont une cinématique plus compliquée. Ajouté à cela leurs dimensions importantes, et évoluant dans des environnements accidentés et parfois inconnus, ils

nécessitent une stratégie de contrôle robuste et intelligente. Dans le cadre du projet « Contrôle intelligent des Robots Mobiles Autonomes (RMAs) », nous avons entamé des travaux de recherche appliqués sur le robot « Robucar », disponible à la division, en vue d'implémenter des techniques de commande intelligente. La navigation autonome de robot de type voiture se résume généralement à un déplacement d'une position initiale vers une destination finale, tout en respectant les contraintes cinématiques du robot, et sans intervention humaine.

Initialement, la destination finale est exprimée dans un environnement à 2D par les coordonnées du centre du robot (X_b , Y_b). Malheureusement, ce n'est pas toujours le cas pour les robots de type voiture, où par la nature des tâches qui leur sont assignées dans les applications d'extérieur comme le transport, l'agriculture, la surveillance, etc., l'orientation finale du robot est souvent requise avec une bonne précision. De ce fait, ces tâches se traduisent alors par une navigation autonome de la position initiale (X_i , Y_i , θ_i) vers un but orienté (X_b , Y_b , θ_b).

Il est clair que l'accomplissement d'une telle tâche, en assurant une orientation finale précise, avec une voiture conventionnelle se fait par des manœuvres non évidentes. De ce fait, la programmation d'une stratégie de navigation autonome passe par l'emploi d'une commande intelligente, employant des connaissances précises d'un expert.

La majorité des travaux traitant le problème de la navigation autonome de robots non-holonomes [Manikonda 1995], [Scheuer 1996], [Paromtchik 1997] se sont basés sur l'implémentation d'une méthode analytique de génération de trajectoire globale. Cette trajectoire peut être composée de plusieurs segments de droite reliés entre eux par des arcs de cercles, ou des paires de Clothoïdes [Scheuer 1996]. Cette classe de méthodes est réputée très peu réactive. D'autres travaux initiés par Santos-Victor et al. [Santos-Victor 2002] ont introduit dans la représentation spatiale, les contraintes cinématiques du robot, ce qui lui permet une navigation réactive dans cet espace modifié. Le problème d'atteindre une destination orientée n'a pas été traité dans ces travaux. Par contre, il a fait l'objet du travail de Fraichard et al. dans [Fraichard 2001].

En effet, au lieu de contraindre le robot à suivre une trajectoire bien précise (planification de trajectoire), Fraichard et al. ont préféré employer un algorithme basé sur la logique floue, dans le but d'atteindre une position finale orientée. La logique floue a été aussi employée pour la commande d'un petit robot non-holonyme, afin d'assurer le suivi d'une trajectoire complexe et pré-calculée [Ollero 1997].

De plus les contraintes cinématiques imposées par la non holonomie, Lindgren et al. ont tenté de modéliser l'interaction dynamique qui existe entre les roues et le sol, pour assurer une meilleure navigation. Ce travail a été expérimenté sur un véhicule autonome utilisé dans l'agriculture [Lindgren 2002].

Enfin, et en vue de réaliser un positionnement orienté d'un véhicule d'extérieur appelé ROJO, une nouvelle stratégie a été employée dans [García-Pérez 2003]. Elle consiste à décomposer la tâche en trois étapes : « Orientation », « Préparation » puis « Approche finale ». Ces dernières sont assurées à l'aide de trois contrôleurs basés sur la logique floue. Bien que cette stratégie a été validée en simulation, les résultats expérimentaux restent très modestes.

Dans ce travail, un contrôleur de manoeuvres basé sur la logique floue est synthétisé, en simulation, en employant le modèle du robot, à savoir modèle de robot mobile non-holonyme à double braquage. Le but est de réaliser un positionnement rapide, précis et sécurisé. L'implémentation temps réel sur le robot « Robucar » de ce contrôleur est ensuite abordée. Enfin, Les résultats de la simulation et de l'expérimentation sont présentés et comparés.

2 COMMANDE PAR LOGIQUE FLOUE

2.1 Généralités

Depuis quelques années déjà, on trouve sur le marché des appareils de grande consommation (appareils de photos, vidéo, ...) qui sont présentés comme faisant intervenir un réglage par logique floue. Au delà de l'argument publicitaire évident, il est intéressant de comprendre ce nouveau concept de réglage et de l'appliquer à certains types de problèmes de commande [Bühler 1994].

Le principe du réglage par logique floue part du constat suivant : dans les problèmes de régulation auxquels il est confronté, l'homme ne suit pas, à l'image de ses inventions, un modèle mathématique précis fait de valeurs numériques et d'équations. Au contraire, il utilise des termes tel que « un peu, beaucoup, plus, à fond, etc... » ainsi que ses propres connaissances qu'il a dans le domaine. Ces connaissances sont, le plus souvent, acquises de façon empirique.

Le principe du réglage par la logique floue s'approche de la démarche humaine dans le sens où les variables traitées ne sont pas des variables logiques (au sens de la logique binaire par exemple) mais des variables linguistiques, proches du langage humain. De plus, ces variables linguistiques sont traitées à l'aide de règles qui font référence à une certaine connaissance du comportement du système à régler.

2.2 Principes [Bühler 1994]

La logique floue est une branche des mathématiques et, à ce titre, toute une série de notions fondamentales sont développées. Ces notions permettent de justifier et de démontrer certains principes de base. Dans ce qui suit, on ne retiendra que les éléments indispensables à la compréhension du principe du réglage par la logique floue. Ces éléments sont :

- les variables floues,
- les règles d'inférences,
- les fonctions d'appartenance.

2.2.1 Les variables floues

Contrairement aux variables binaires qui sont définies par les deux états « vrai » ou « faux », les variables floues présentent toute une graduation entre ces deux valeurs. D'une part, on préfère représenter l'état de la variable à l'aide de son degré de vérité en associant la valeur 1 (degré de vérité de 100%) à la valeur « vrai » et le degré de vérité nul à la valeur « faux ». D'autre part, on constate que cette façon de faire est très éloignée de ce que fait l'être humain lorsqu'il résout ce genre de problème. En effet, l'homme ne fait pas naturellement une distinction franche entre « petit » et « moyen » par exemple. Il utilise des expressions du genre « plutôt petit ».

En conclusion, la logique binaire présente l'avantage de la simplicité mais est assez éloignée de la logique utilisée naturellement par l'être humain. Si l'on représente le même problème à l'aide de la logique floue, les variables ne sont plus binaires mais présentent une infinité de valeurs possible entre le « vrai » et le « faux ».

On constate que cette représentation est beaucoup plus proche de la façon dont l'être humain raisonne, puisqu'elle permet de faire intervenir des notions telles que « plutôt petit », « assez grand » etc. Cet avantage se fait, évidemment, au détriment de la simplicité de la représentation.

2.2.2 Les règles d'inférence

On appelle règles d'inférence l'ensemble des différentes règles reliant les variables floues d'entrée d'un système aux variables floues de sortie de ce système. Ces règles se présentent sous la forme :

- Si (condition 1) et/ou (condition 2) et/ou ... (condition N) alors (action sur les sorties).

En termes d'intelligence artificielle, ces règles résument en fait l'expérience de l'expert et elles ne sont en général pas définissables de façon unique puisque chaque individu crée ses propres règles. Les grandeurs utilisées dans un système de réglages sont généralement générées par des capteurs.

Il est nécessaire de convertir ces grandeurs en variables floues. Pour ce faire, on définit les deux notions suivantes :

- Les fonctions d'appartenances qui permettent de définir le degré de vérité de la variable floue en fonction de la grandeur d'entrée.
- Les intervalles flous qui déterminent le nombre de variables floues.

2.2.3 Fonctions d'appartenance

Il s'agit d'établir une relation entre le degré de vérité de la variable floue et la grandeur d'entrée correspondante. On parle alors de « Fuzzyfication ». On peut évidemment choisir n'importe quelle forme pour les fonctions d'appartenance. Cependant, en pratique, on utilise les formes trapézoïdales et les formes en cloche (gaussiennes).

3 APPLICATION

L'application se résume en l'implémentation d'un régulateur basé sur la logique floue. La position et l'orientation actuelles du robot sont calculées, en temps réel, par le module d'odométrie. Les sorties du régulateur sont la vitesse de translation et l'angle de braquage nécessaires pour atteindre une position désirée (le but), comme illustré sur la figure 1. Par contre, l'orientation finale du robot n'est pas contrôlée.

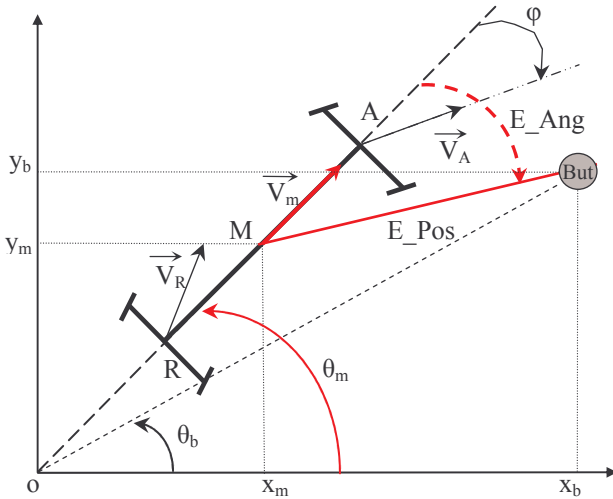


Figure 1 : Schéma de principe de la technique

Les points A, M et R représentent respectivement l'avant, le centre et l'arrière du véhicule. Le régulateur a comme entrées la distance robot - but (notée E_Pos) ou l'erreur de position, et l'erreur angulaire entre son orientation actuelle et celle du but (notée E_Ang). Les expressions de ces deux grandeurs sont données par les équations suivantes :

$$\begin{cases} E_Pos = \sqrt{D_x^2 + D_y^2} \\ E_Ang = \theta_m - \theta \end{cases} \quad (1)$$

$$D_x = x_b - x_m \quad (2)$$

$$D_y = y_b - y_m \quad (3)$$

$$\theta = \text{Atan}(D_y, D_x) \quad (4)$$

Avec :

(x_m, y_m, θ_m) : position et orientation du robot

(x_b, y_b, θ_b) : position et orientation du but

V_m : vitesse de translation du robot

ϕ : angle de braquage du robot

En se basant sur les caractéristiques techniques du robot, la conception du contrôleur flou est proposée en définissant les propriétés fonctionnelles et opérationnelles ci-dessous :

3.1 Les fonctions d'appartenance

Cela consiste à spécifier le domaine de variation des variables : l'univers de discours, qu'on divise en intervalles (sous ensembles flous ou valeurs linguistiques). Cette répartition, qui consiste à fixer le nombre de ces valeurs et les distribuer sur le domaine, est faite en se basant sur des connaissances du système et selon la précision désirée. Les fonctions d'appartenance sont explicitées dans les figures qui suivent (μ_x représente la variable floue associée à la variable physique x) :

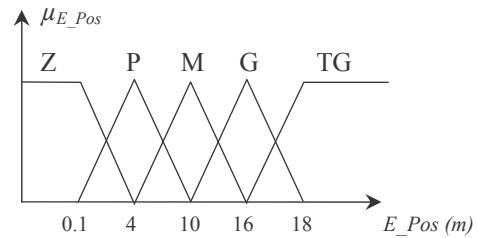


Figure 2 : Fonctions d'appartenance de l'erreur de position

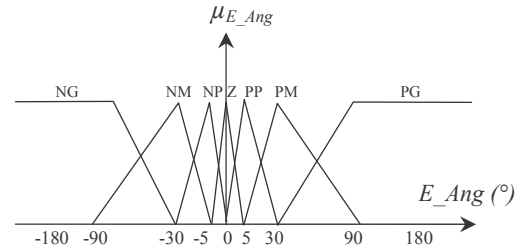


Figure 3 : Fonctions d'appartenance de l'erreur angulaire

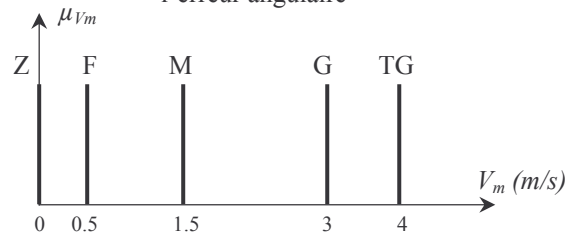


Figure 4 : Variable de sortie : Vitesse de translation du robot

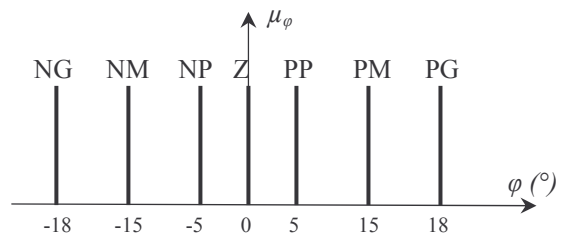


Figure 5 : Variable de sortie : Angle de braquage des roues

Les significations des labels, désignant les noms des valeurs linguistiques, sont :

Z : Zéro P : Petite F : Faible
M : Moyenne G : Grande TG : Très Grande
NG : Négative Grande NM : Négative Moyenne
NP : Négative Petite PP : Positive petite
PM : Positive Moyenne PG : Positive Grande.

3.2 La base des règles

Cette étape concerne l'élaboration des règles, pour définir le comportement attendu du robot selon ses paramètres intrinsèques : dimensions, angle de braquage et vitesse. Pour chacune des combinaisons des valeurs des variables d'entrées, une action sur les variables de sorties lui est associée. La table des règles floues « Situation/Action », proposée dans la table 1, est construite manuellement.

			Erreur Angulaire (E_{Ang})						
			NG	NM	NP	Z	PP	PM	PG
Erreur de Positin (E_{Pos})	Z	φ	PM	PP	Z	Z	Z	NP	NM
		V_m	Z	Z	Z	Z	Z	Z	Z
	P	φ	PG	PG	PM	Z	NM	NG	NG
		V_m	F	F	F	F	F	F	F
	M	φ	PM	PM	PP	Z	NM	NG	NG
		V_m	F	F	M	M	M	F	F
	G	φ	PM	PP	PP	Z	NP	NP	NM
		V_m	F	M	G	G	G	M	F
	TG	φ	PM	PM	PP	Z	NP	NM	NM
		V_m	F	M	G	TG	G	M	F

Table 1 : Règles floues

Le mode d'utilisation de cette table est illustré par cet exemple (cases en ligne doublée) : Si E_{Pos} est petite (P) et E_{Ang} est nulle (Z), alors le but est tout droit mais à une petite distance. Dans ce cas, le robot doit avancer avec une vitesse V_m faible (F) sans changer de direction (φ est Z).

Cependant, le choix optimal des fonctions d'appartenance et des règles représente une tâche très difficile. Elle requiert du temps, de l'expérience et de l'habileté de la part de l'expérimentateur.

3.3 Méthode de défuzzification

Une fois la mise en place des fonctions d'appartenance et l'établissement des règles définissant le comportement du régulateur ont été effectués, on passe à la sélection d'une méthode de défuzzification. C'est cette étape qui permet de transformer les valeurs de commande du domaine flou vers le domaine réel (variables physiques).

Ce choix est généralement conditionné par un compromis entre facilité d'implémentation et performance de calcul [Bühler 1994]. Dans ce travail, nous avons utilisé la méthode du centre de gravité, qui peut être déterminé à l'aide de la relation générale :

$$z^* = \frac{\sum_{i=1}^m z_i \cdot \mu(z_i)}{\sum_{i=1}^m \mu(z_i)} \quad (5)$$

Avec

m : Nombre de règles

z : Variable de commande

3.4 Simulation

Avant de tester le régulateur flou sur le robot « Robucar », une implémentation sur un modèle de robot de type voiture a été simulée à l'aide de « Fuzzy Logic Toolbox » du « Simulink », sous « Matlab 6 ».

Plusieurs tests, pour différentes configurations du but, ont été effectués. L'origine du repère $O(x_m=0m, y_m=0m, \theta_m=0^\circ)$ est toujours pris comme position initiale du robot.

Les figures ci-dessous montrent la trajectoire du robot ainsi que l'évolution des variables de commande en fonction du temps, pour un but situé au point $(x_b = 30m, y_b = 20m)$:

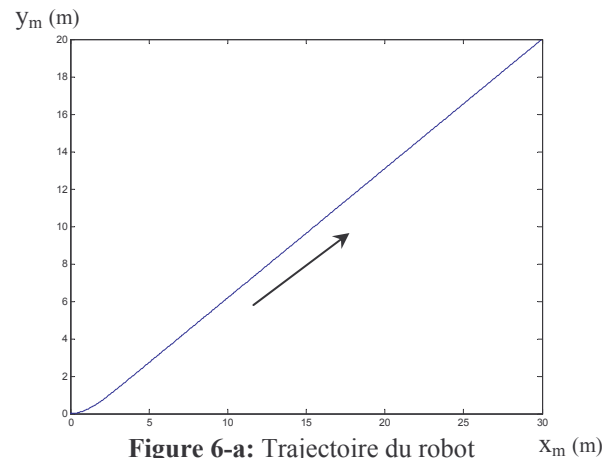


Figure 6-a: Trajectoire du robot

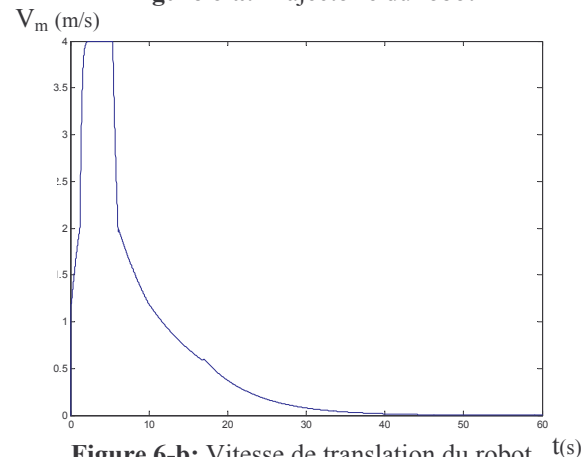


Figure 6-b: Vitesse de translation du robot

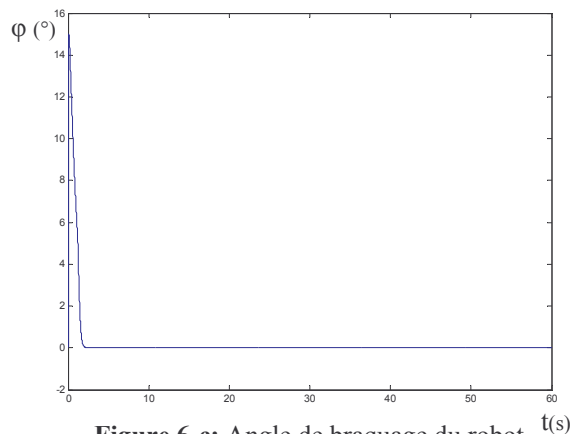


Figure 6-c: Angle de braquage du robot

- Résultats pour $(x_b = -30m, y_b = -15m)$:

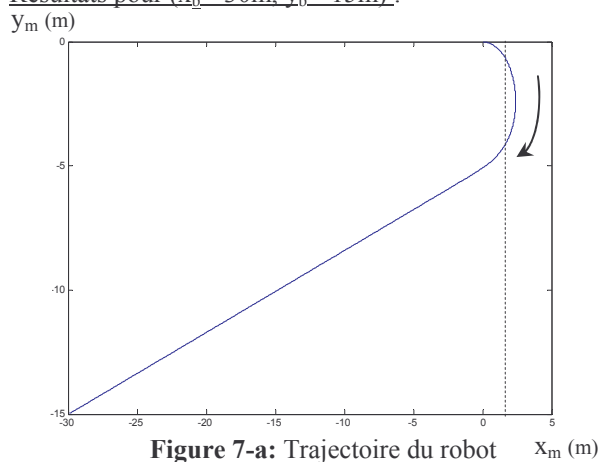


Figure 7-a: Trajectoire du robot

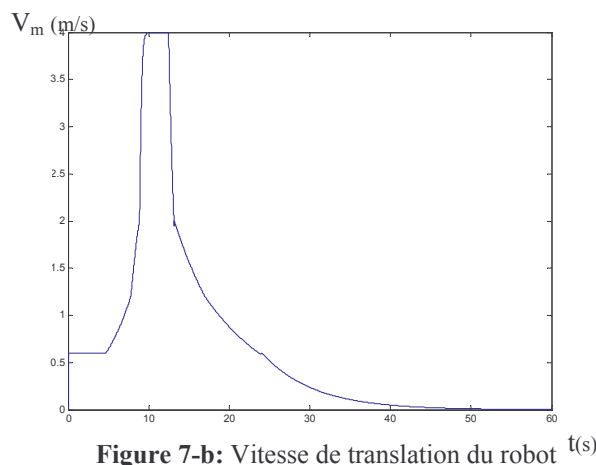


Figure 7-b: Vitesse de translation du robot

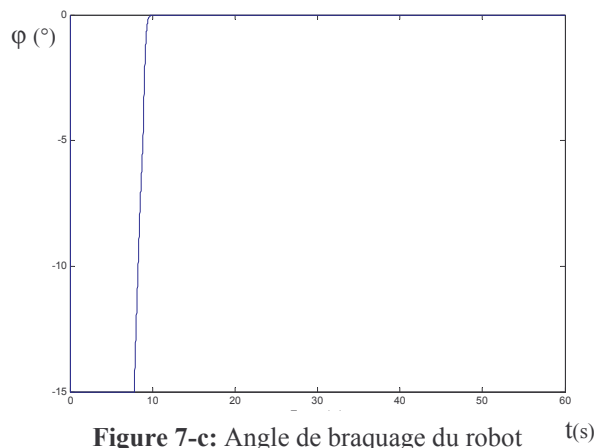


Figure 7-c: Angle de braquage du robot

Après une série de tests, on a constaté que le régulateur assure la stabilité du système (variables de commande bornées). La précision de positionnement est appréciable vu les dimensions du robot. En effet, pour tous les cas de figures, le but a été atteint avec une erreur très faible ($< 5cm$).

De plus, les variables de commande du robot progressent jusqu'à atteindre un palier (saturation), puis diminuent et s'annulent quand le but est atteint. Mais, l'évolution de ces variables est trop rapide, au démarrage, relativement à la nature du système (temps de réponse lent). Ce phénomène peut être atténué, lors de l'implémentation temps réel, grâce aux retards introduit par les actionneurs du robot.

Le comportement du régulateur flou semble très correct en simulation. Pour confirmer cela, il a été implémenté en temps réel sur une plateforme expérimentale, présentée sur la figure 8.

3.5 Expérimentation



Figure 8: Le Robucar

L'expérimentation a été réalisée sur le prototype « Robucar » montré ci-dessus. Pour la commande en temps réel, deux contrôleurs ont été implémentés sur le PC embarqué du robot :

- Un contrôle bas niveau : C'est la commande des actionneurs du robot. Cette dernière est assurée par un graphe sous « Syndex 5.2 ». Les actionneurs sont contrôlés en position ou en vitesse, suivant leur nature et leur rôle sur le robot. Les contrôleurs employés sont de type classique (PI, PID).
- Un contrôle Supérieur (haut niveau) : Ce contrôleur représente le régulateur flou de la position du robot. Il se traduit par une commande de la vitesse et de l'angle de braquage, en tenant compte de la position actuelle du robot et celle du but. Ce contrôleur est implémenté sous le langage C.

Les deux contrôleurs fonctionnent simultanément et en temps réel. L'échange de données nécessaires pour le calcul et l'envoi des paramètres de commande (communications entre les deux contrôleurs) est assuré

par l'intermédiaire de RTAI (Real Time Application Interface) sous « Linux Redhat 9 ».

Pour les tests, et comme dans la simulation, plusieurs essais pour différentes configurations de la cible ont été effectués. Par contre, vu l'espace restreint disponible au laboratoire, les coordonnées de la cible (x_b et y_b) ont été adaptées de telle sorte à permettre au robot de l'atteindre sans risque de collision. Enfin, la vitesse maximale du robot a été diminuée pour des raisons de sécurité.

Les données des capteurs (position, orientation, vitesse et angle de braquage) ont été enregistrées en temps réel lors du déplacement du robot. Les figures suivantes montrent quelques résultats de l'expérimentation sur le robot :

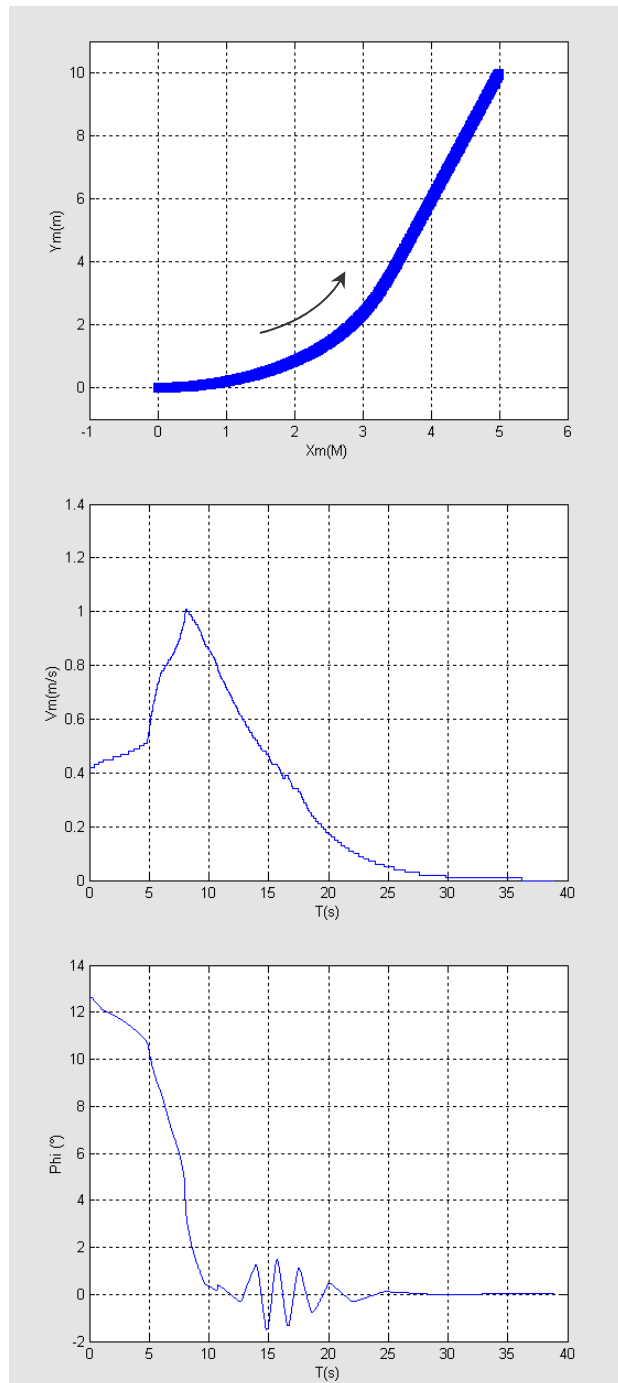


Figure 9: Résultats expérimentaux ($x_b=5$, $y_b=10$)m

Comme pour la simulation, la position initiale du robot est l'origine du repère terrestre, O ($x_m=0m$, $y_m=0m$, $\theta_m=0^{\circ}$). Après observation des réponses du robot face aux commandes injectées par le régulateur flou, nous avons été obligés de modifier les valeurs des amplitudes des singletons pour les variables de sortie, afin de garantir la stabilité du robot, et d'améliorer la précision du réglage. Les résultats montrés sur les figures 9, 10 ont été obtenus après ces modifications.

De ces résultats, nous remarquons que le régulateur agit de manière correcte sur le robot et le mène vers la position désirée avec une précision très appréciable. L'erreur de positionnement converge asymptotiquement vers zéro dans tous les cas de figure.

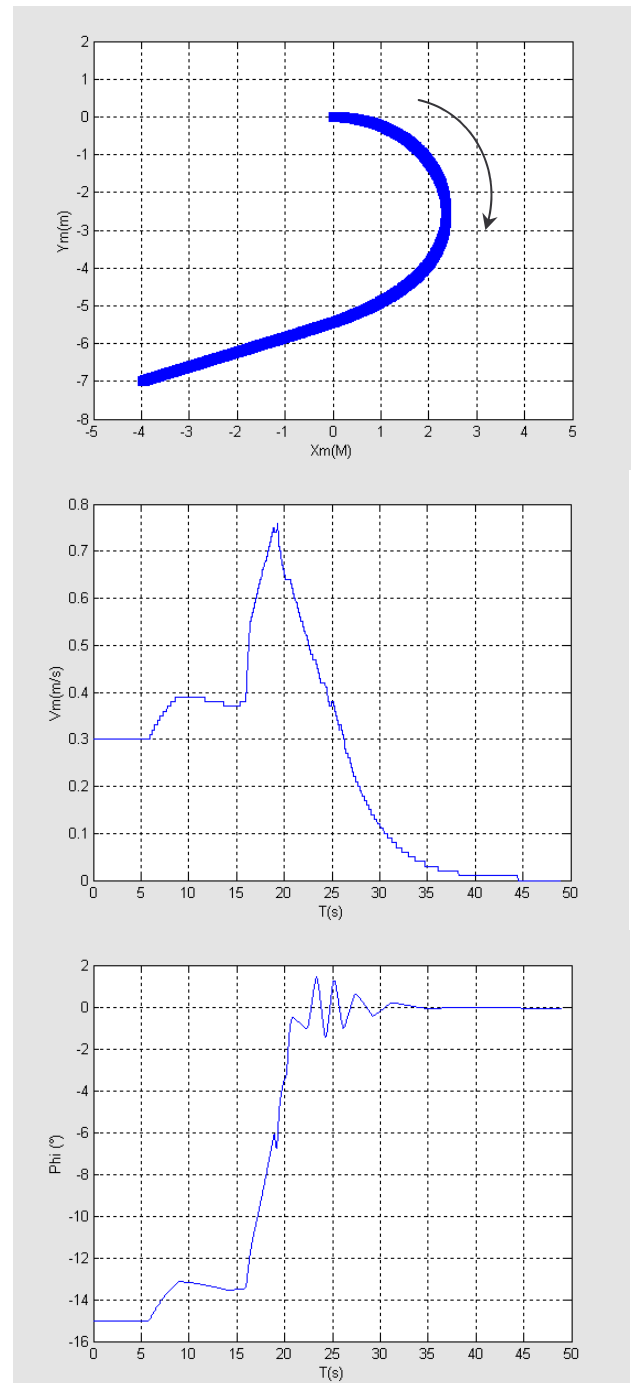


Figure 10: Résultats expérimentaux ($x_b=-4$, $y_b=-7$)m

Les commandes restent toujours bornées, donc la sollicitation des actionneurs du robot est normale. Concernant l'angle de braquage, la nouvelle distribution des sous-ensembles flous pour la commande de l'erreur angulaire a montré une bonne efficacité, sauf qu'il ne faut pas qu'elle dépasse les limites fixées $[-180^\circ, 180^\circ]$. Pour ce faire, des fonctions trigonométriques assurant une limitation lors du fonctionnement ont été employées. Par contre, nous remarquons que l'effet Echelon (variation brusque) de la commande en braquage, observé en simulation, persiste toujours. Heureusement, ce phénomène se produit uniquement au démarrage, où la vitesse du robot est faible.

Pour la vitesse, l'allure est similaire à celle obtenue en simulation. De plus, les accélérations et les amplitudes sont plus faibles en expérimentation. Toutefois, comme lors de la simulation, la vitesse évolue très rapidement au départ, mais la valeur atteinte reste faible, la réaction des moteurs a été normale face à ces consignes. Nous pensons que si l'erreur de positionnement était très importante ($>50\text{m}$), ce comportement pourrait poser des problèmes. Une solution peut être envisagée pour limiter ce phénomène : elle consiste à ajouter une troisième entrée à ce régulateur, qui est la variation de l'erreur de position « ΔE_{Pos} ». Cela va permettre un meilleur contrôle de la vitesse et imposera une variation graduelle au démarrage. Enfin, d'autres résultats pour différentes positions du but sont représentés sur les figures suivantes :

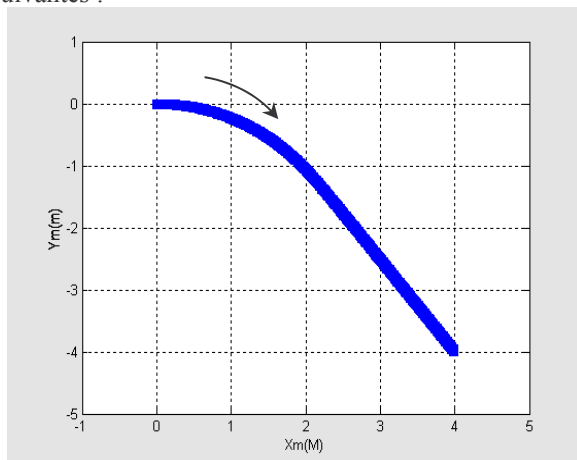


Figure 11: Trajectoire du robot ($x_b=4\text{m}$, $y_b=-4\text{m}$)

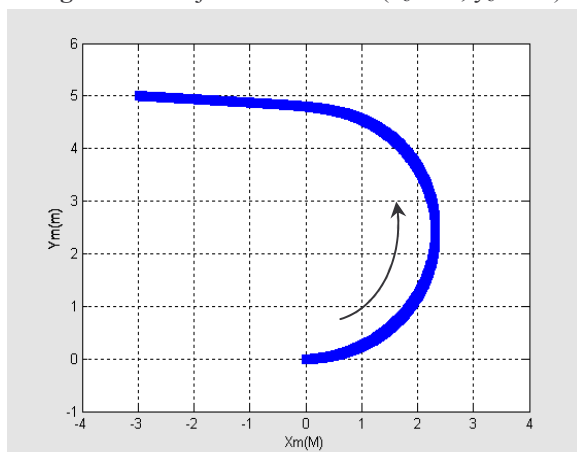


Figure 12: Trajectoire du robot ($x_b=-3\text{m}$, $y_b=5\text{m}$)

4 CONCLUSION

Dans ce travail, le problème de la navigation autonome d'un robot de type voiture a été abordé, en employant la localisation par odométrie. Nous avons synthétisé un régulateur flou qui a été d'abord testé en simulation, puis implémenté sur le robot mobile « Robucar ». Les résultats expérimentaux obtenus sont satisfaisants, et ouvrent la voix vers l'implémentation d'un autre régulateur flou en vue d'assurer un positionnement avec prise en compte de l'orientation finale du véhicule.

BIBLIOGRAPHIE

- [Baille 1999] Baille G., Garnier P., Mathieu H., Pissard-Gibollet R.: "*Le CyCab de L'INRIA Rhône Alpes*". Rapport technique, INRIA Rhône Alpes, France (1999).
- [Bühler 1994] Bühler H.: "*Réglage par la logique floue*", Presses polytechniques et universitaires romandes, (1994).
- [Fraichard 2001] Fraichard T., Garnier P.: "*Fuzzy control to drive car-like vehicles*". Robotics and Autonomous Systems, vol. 34, N° 1, p. 1-22 (2001).
- [García-Pérez 2003] García-Pérez L., García-Alegre M. C., Ribeiro A., Guinea D.: "*Fuzzy control for an approaching-orienting maneuver with a car like vehicle in outdoor environments*". IV Workshop de Agentes Físicos (WAF03), Alicante, Spain (2003).
- [Lindgren 2002] Lindgren D. R., Hague T., Probert Smith P. J., Marchant J. A.: "*Relating torque and slip in an odometric model for an autonomous agricultural vehicle*". Autonomous Robots, vol. 13, p. 73-86 (2002).
- [Manikonda 1995] Manikonda V., Hendler J. A., Krishnaprasad P. S.: "*Formalizing behavior-based planning for non-holonomic robots*". In IJCAI, p. 142-149 (1995).
- [Ollero 1997] Ollero A., García-Cerezo A., Martínez J. L., Mandow A.: "*Fuzzy tracking methods for mobile robots*". Chapter 17 in Jamshidi M., Titli A., Zadeh L., Boverie S. "*Applications of fuzzy logic: Towards high machine intelligence quotient systems*". Prentice-Hall, New Jersey (1997).
- [Paromtchik 1997] Paromtchik I., Garnier P., Laugier C.: "*Autonomous maneuvers of a non-holonomic vehicle*". In International Symposium on Experimental Robotics, Barcelona, Spain (1997).
- [Santos-Victor 2002] Santos-Victor J., Minguez J., Montano L.: "*Reactive navigation for non-holonomic robots using the ego kinematic space*". In Proceedings of the International Conference on Robotics and Automation (ICRA02), Washington, USA (2002).
- [Scheuer 1996] Scheuer A., Fraichard Th.: "*Planning continuous-curvature paths for car-like robots*". In Proceedings of the International Conference on Intelligent Robots and Systems, vol 3, p. 1304-1311, Osaka, Japan (1996).