

PLANIFICATEUR DE TRAJECTOIRE AVEC EVITEMENT D'OBSTACLE BASE SUR LA METHODE DES CONTRAINTES POUR ROBOT MOBILE

GUESBAYA TAHAR¹, BENMAHAMMED.KHIER² & BENALI.ABDERRAOUF³

¹Laboratoire MSE, Université de Biskra - Algérie

²Département d'électronique, Université de setif - Algérie

³Laboratoire LVR, ENSIB, Bourges - France

RESUME

Dans ce travail nous apportant une contribution à la planification locale de trajectoires pour les robots mobiles non-holonomes avec évitement d'obstacles qui est basé sur la méthode des contraintes et certaines règles heuristiques. La méthode pose des contraintes sur les vitesses permises au robot mobile l'obligeons à s'éloigner des obstacles. Le robot mobile est doté de capteurs ultrason pour mesurer la distance minimale aux obstacles. La modélisation des obstacles et des capteurs par des formes polygonales convexes facilite l'utilisation de procédures de calcul de distance minimales. Les résultats de simulation ont montré la souplesse avec laquelle le robot mobile atteint la cible en traversant des parcours difficiles.

Mots clés : Robot mobile, non holonome, méthode des contraintes, règles heuristiques.

1 INTRODUCTION

Vu le nombre de tâches limitées qu'un bras manipulateur peut réaliser c'est apparu l'intérêt croissant, dans ces dernières décennies, aux plates formes mobiles.

Les intérêts portés à la robotique mobile concernent la planification de trajectoires dans des milieux libres ou contenant des obstacles, la modélisation cinématique et dynamique, les interactions dynamiques entre plate-forme et manipulateurs.

Des recherches concernent aussi les différentes techniques de commande, la planification de trajectoires locales ou globales.

Dans ce travail la planification de trajectoire est locale, elle est réalisée pour un robot mobile différentiel où les deux roues arrière sont commandées indépendamment et leur différence de vitesse agit sur l'orientation. Le modèle utilisé est le modèle cinématique avec considération de la non-holonomie.

La génération de trajectoire dans le cas de parcours libres d'obstacles est basée sur une loi de commande qui optimise le parcours du robot mobile en ligne droite.

Dans le cas de présence d'obstacles la méthode des contraintes [5] est appliquée. Elle limite l'ensemble des vitesses que peut prendre le robot empêchant la collision avec des obstacles. Si un blocage derrière un obstacle survient, un ensemble de règles heuristiques est appliqué pour contourner l'obstacle en se basant toujours sur la méthode des contraintes. Les obstacles et les capteurs sont

modélisés géométriquement comme des polygones convexes [8], [10], [3], [13] et le calcul de la distance minimale entre eux devient plus facile.

2 MODÉLISATION CINÉMATIQUE

Le robot mobile différentiel est une plate forme avec deux roues arrière motorisées, montées sur le même axe et commandées indépendamment l'une de l'autre possédant en plus une roue avant libre.

La configuration du robot mobile est indiquée ci-dessous (figure.1).

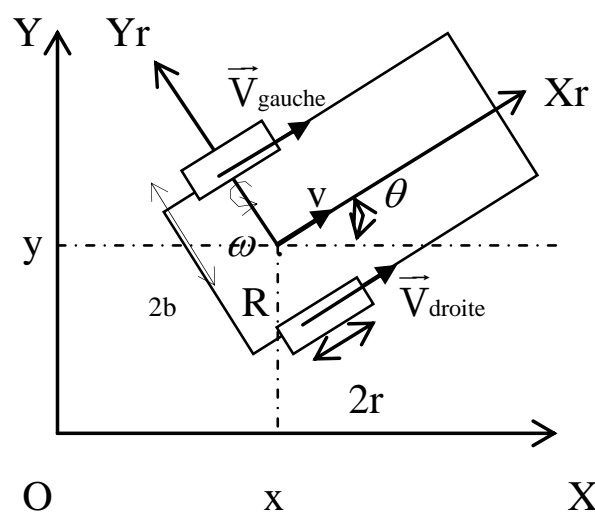


Figure 1 : Représentation cinématique du robot mobile

Les hypothèses simplificatrices considérées pour la modélisation sont :

- Le contact roue sol est un contact ponctuel.
- Le roulement de chaque roue s'effectue sans glissement.
- Le système évolue sur un sol horizontal.
- La configuration du robot mobile est décrite par cinq coordonnées généralisées :

$$q = [x \ y \ \theta \ \phi_g \ \phi_d]^T.$$

ϕ_d : L'angle de rotation de la roue droite.

ϕ_g : L'angle de rotation de la roue gauche.

Nous pouvons écrire pour les deux roues arrières les équations de vitesses suivantes [6],[12]:

$$\begin{aligned} \vec{V}_{gauche} = & (\dot{x} \cos(\theta) + \dot{y} \sin(\theta) - b\dot{\theta})\bar{x}_r \\ & + (-\dot{x} \sin(\theta) + \dot{y} \cos(\theta))\bar{y}_r \end{aligned} \quad (II-1)$$

$$\begin{aligned} \vec{V}_{droite} = & (\dot{x} \cos(\theta) + \dot{y} \sin(\theta) + b\dot{\theta})\bar{x}_r \\ & + (-\dot{x} \sin(\theta) + \dot{y} \cos(\theta))\bar{y}_r \end{aligned} \quad (II-2)$$

Il est évident que les roues ne se déplacent pas selon l'axe Y_r , donc la composante correspondante pour \vec{V}_{gauche} et \vec{V}_{droite} est nulle.

Alors :

$$-\dot{x} \sin(\theta) + \dot{y} \cos(\theta) = 0 \quad (II-3)$$

Les composantes selon X_r pour \vec{V}_{gauche} et \vec{V}_{droite} sont égales respectivement à :

$$\dot{x} \cos(\theta) + \dot{y} \sin(\theta) + b\dot{\theta} = r\dot{\phi}_d \quad (II-4)$$

$$\dot{x} \cos(\theta) + \dot{y} \sin(\theta) - b\dot{\theta} = r\dot{\phi}_g \quad (II-5)$$

Ces contraintes cinématiques peuvent être réécrites selon la forme matricielle standard suivante :

$$A(q)^T \cdot \dot{q} = 0 ; A(q) = \begin{bmatrix} \sin(\theta) & -\cos(\theta) & 0 & 0 & 0 \\ \cos(\theta) & \sin(\theta) & b & -r & 0 \\ \cos(\theta) & \sin(\theta) & -b & 0 & -r \end{bmatrix} \quad (II-6)$$

Dans ce travail on s'intéresse à la forme ordinaire suivante [8] :

$$\dot{q} = S(q)V(t) \text{ et } V(t) = [v(t) \ \omega(t)] \quad (II-7)$$

où $q = [x \ y \ \theta]^T$

Où v est la vitesse linéaire et ω la vitesse angulaire du robot mobile autour du point R (origine du repère) alors :

$$[\dot{x} \ \dot{y} \ \dot{\theta}]^T = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} [v \ \omega]^T \quad (II-8)$$

3 MODÉLISATION DES CAPTEURS

Le robot mobile est équipé de huit capteurs ultrason installés comme indiqués en figure 2.

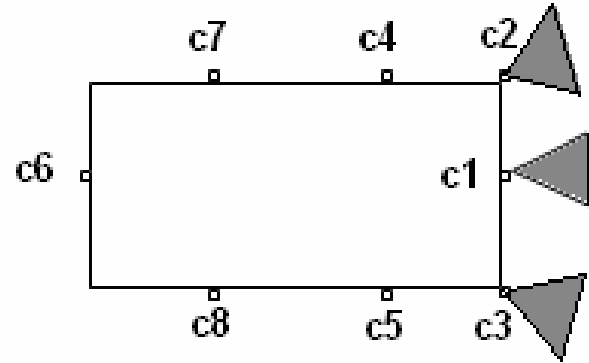


Figure 2 : Positionnement des capteurs

La plus part des capteurs ultrason possèdent un angle d'ouverture de 26° et un champs de mesure entre 16cm et 10 mètres.

Le capteur est défini par l'ensemble des paramètres suivants (figure 3):

P : Représente les coordonnées de la position du capteur en repère lié au robot.

α : est l'angle d'émission du capteur par rapport à l'axe X_r .

γ : est l'angle d'ouverture du capteur.

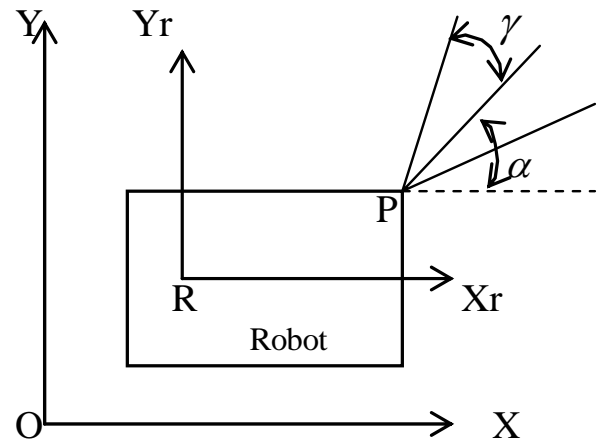


Figure 3 : Paramètres du capteur

En simulation chaque capteur est modélisé d'une manière géométrique par deux contraintes linéaires du type $(y < a_i x + b_i)$ représentant l'ensemble des points entre les deux droites c'est à dire appartenant au cône d'émission.

Donc :

Pour $i=1,2$; $(y < a_i x + b_i)$ ou $(-a_i x + y < b_i)$

Sous forme matriciel :

$$A_{sens} X \leq B_{sens} \quad (III-1)$$

avec :

$$X = [x \ y]^T$$

$$A_{sens} = \begin{bmatrix} -a_{1_{sens}} & 1 \\ -a_{2_{sens}} & 1 \end{bmatrix}, B_{sens} = \begin{bmatrix} b_{1_{sens}} \\ b_{2_{sens}} \end{bmatrix} \quad (III-2)$$

4 MODÉLISATION DES OBSTACLES

Chaque obstacle est considéré ayant une forme géométrique polygonale convexe. L'obstacle est donc l'ensemble des points appartenant au polygone modélisé par un ensemble de contraintes linéaires de type $(y < a_i x + b_i)$.

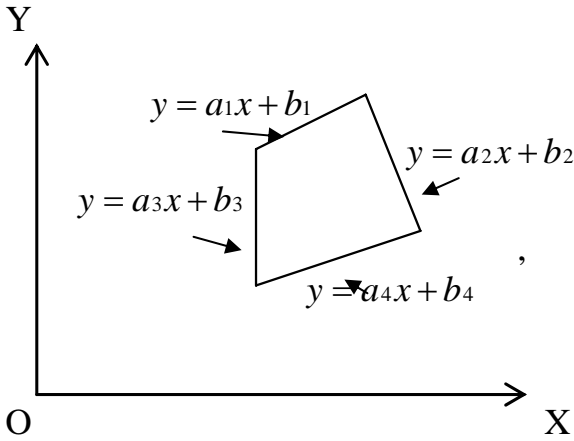


Figure 4 : Modélisation de l'obstacle par des contraintes géométriques $(y < a_i x + b_i)$

Un obstacle dans notre cas est de quatre arrêtes représenté par quatre contraintes.

Alors à chaque obstacle on a :

Pour $i=1,4$:

$$(y < a_i x + b_i) \text{ ou } (-a_i x + y < b_i)$$

ou sa forme matricielle :

$$A_{ob} X \leq B_{ob} \quad (IV-1)$$

avec :

$$X = [x \ y]^T$$

$$A_{ob} = \begin{bmatrix} -a_{1_{ob}} & 1 \\ -a_{2_{ob}} & 1 \\ -a_{3_{ob}} & 1 \\ -a_{4_{ob}} & 1 \end{bmatrix}, B_{ob} = \begin{bmatrix} b_{1_{ob}} \\ b_{2_{ob}} \\ b_{3_{ob}} \\ b_{4_{ob}} \end{bmatrix} \quad (IV-2)$$

5 CALCULE DE DISTANCE MINIMALE

Le capteur détecte la présence d'un obstacle revient à dire qu'il existe au moins un point appartenant à son cône d'émission de même appartient au polygone représentant l'obstacle.

Le point le plus proche du capteur c'est à dire du robot se trouve logiquement sur le contour de l'obstacle.

La distance entre le point d'émission du capteur $P(x_p, y_p)$ et un point quelconque $Obs(x, y)$ de l'obstacle est :

$$d = [(x - x_p)^2 + (y - y_p)^2]^{1/2} \quad (V-1)$$

Le point le plus proche $Obs(x_{min}, y_{min})$ qui donne la distance minimale d_{min} est obtenu par la minimisation de d sous les contraintes de l'obstacle et du capteur $AX < B$.

avec :

$$A = \begin{bmatrix} A_{ob} \\ A_{sens} \end{bmatrix}, B = \begin{bmatrix} B_{ob} \\ B_{sens} \end{bmatrix}, X = [x \ y]^T \quad (V-2)$$

Par suite :

$$d_{min} = [(x_{min} - x_p)^2 + (y_{min} - y_p)^2]^{1/2} \quad (V-3)$$

Donc le calcul de distance minimale devient un problème d'optimisation d'un critère quadratique avec les contraintes géométriques de l'obstacle et du capteur.

L'optimisation du critère quadratique déduit de d (formule V-1) est :

$$(1/2 X^T G X + F^T X) \quad (V-4)$$

$$X = [x \ y]^T, G = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, F = [-2x_p \ -2y_p]$$

L'instruction 'quadprog' de Matlab permet la résolution du problème de minimisation sous contraintes du type $AX \leq B$. Ce qui permet de trouver le point minimal $'x_{dmin} \ y_{dmin}'$.

6 LOI DE CONTRÔLE POUR PARCOURS LIBRE

Si le parcours devant le robot mobile est considéré libre, il doit donc optimiser son chemin pour aller en ligne droite vers la cible, et d'une façon stable.

Soit $q_f = [x_f, y_f, 0]^T$ la position de la cible à atteindre et $q = [x, y, \theta]^T$ la position courante du robot mobile.

Soit $x_e = x_f - x$ et $y_e = y_f - y$; et parce que les variables de contrôle sont au nombre de deux (v, ω) , on doit définir pour le robot les coordonnées polaires suivants : a et α_1 .

avec :

$$a = [x_e^2 + y_e^2]^{1/2}$$

$$\alpha_1 = \text{atan2}(y_e, x_e) - \theta, \quad -\pi < \alpha_1 < \pi \quad (\text{VI-1})$$

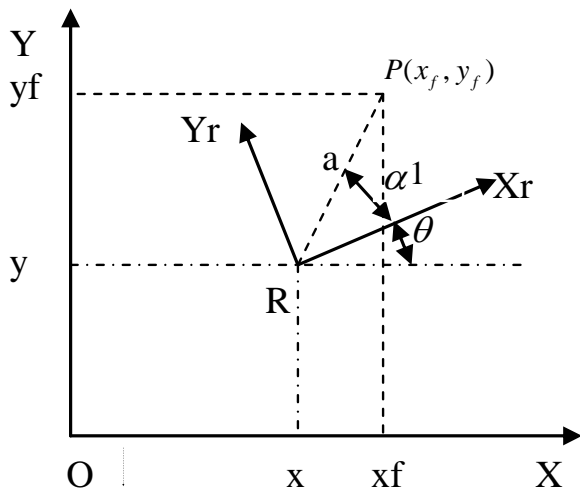


Figure 5 : Représentation cinématique en coordonnées polaires

Les variables de contrôle (v, ω) doivent assurer la réorientation du robot vers la cible par ω et sont rapprochement de la cible par v .

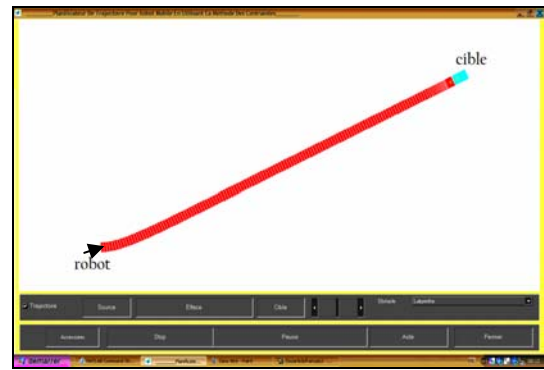
La loi de contrôle suivante [7], [11] est proposée :

$$v = k_1 \cdot a \cdot \cos(\alpha_1)$$

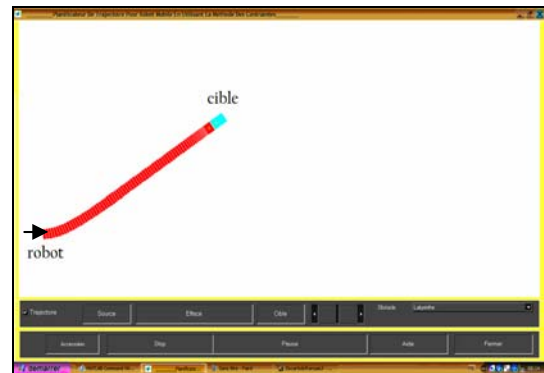
$$\omega = k_2 \cdot \alpha_1 + k_1 \cdot \sin(\alpha_1) \cos(\alpha_1) \quad (\text{VI-1})$$

avec k_1, k_2 sont des constantes.

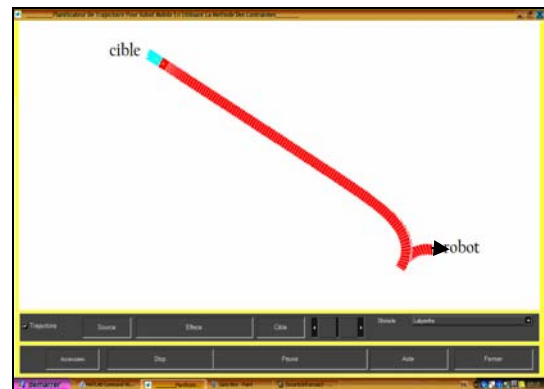
Cette loi de contrôle est appliquée tant qu'on ne détecte pas d'obstacle, et mène le robot en ligne droite vers la cible.



(a)



(b)



(c)

Figure 6 : (a),(b),(c): Robot mobile le long de parcours libre d'obstacle

7 MÉTHODE DES CONTRAINTES

La méthode est une alternative à la méthode de champs de potentiel pour la navigation sans collision de robots manipulateurs dans des environnements encombrés d'obstacles [5], [9], [4].

La méthode est adaptée aux robots mobiles de sorte que les obstacles exercent dans le plan des vitesses, des contraintes cinématiques sur le robot. Ces contraintes cinématiques obligent le robot de rester éloigné des obstacles pour éviter leur collision.

Les contraintes entrent en action lorsque le robot s'approche suffisamment des obstacles.

Soit :

d : distance minimale entre obstacle et robot mobile.

di : distance d'influence à partir de laquelle la contrainte devient active .

ds : distance de sécurité représentant la distance minimale permise entre le robot et l'obstacle.

λ : coefficient d'ajustement de la vitesse de convergence de d vers ds .

La condition d'anticollision entre le robot et un obstacle est tel que :

$$\dot{d} \geq -\lambda(d - ds)/(di - ds) \quad (\text{VII-1})$$

où à $t = t_0$ $d = d_0 > ds$

Par intégration de \dot{d} , l'évolution de d est :

$$d(t) \geq ds + (d_0 - ds) \exp(-\lambda(t - t_0)/(di - ds)) \quad (\text{VII-2})$$

Du fait que $d(t) > ds$ pour $t > t_0$ donc l'obstacle et le robot n'entrent pas en collision.

Lorsque le robot détecte par ses capteurs un ensemble d'obstacles, alors en associe à chaque obstacle une contrainte cinématique d'anticollision:

Contrainte k :

$$\dot{d}_k \geq -\lambda_k (d_k - ds_k)/(di_k - ds_k) \quad (\text{VII-3})$$

En plus on a des contraintes concernant les limitations physiques sur les vitesses :

$$(|v| < v_{\max}, |\omega| < \omega_{\max}) \quad (\text{VII-4})$$

Cet ensemble de contraintes construit un polygone convexe des vitesses permises (PVP) assurant la non collision.

Le (PVP) est évalué à chaque cycle de contrôle.

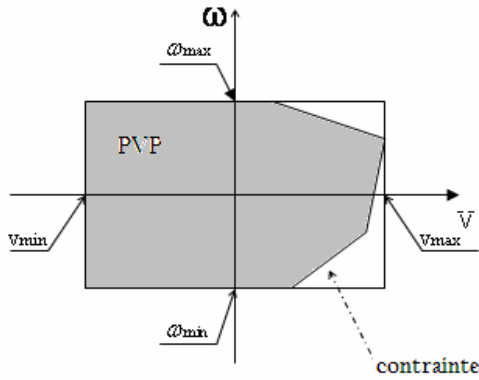


Figure 7 : Polygone des vitesses **permises (PVP)**

8 REPRÉSENTATION DES CONTRAINTES DANS LE PLAN DES VITESSES

La contrainte porte sur les vitesses de contrôle linéaire v et de rotation ω du robot mobile.

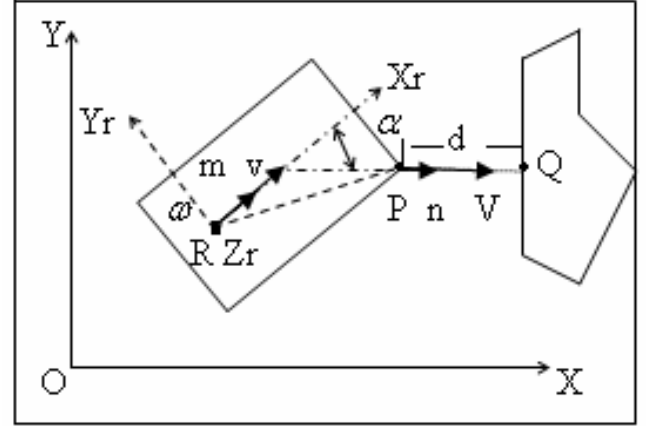


Figure 8 : Représentation des contraintes dans le plan v, ω

D'après le schéma indiqué sur la figure 8 on peut écrire :

$$\dot{d} \geq -\lambda(d - ds)/(di - ds)$$

où

$$V_p \cdot n < +\lambda(d - ds)/(di - ds) \quad (\text{VIII-1})$$

V_p : vecteur vitesse du point P.

n : vecteur unitaire le long de PQ.

α : angle que fait PQ avec X_r .

z : vecteur unitaire le long de l'axe Z_r .

avec

$$V_p = v \cdot m + \omega \cdot z \wedge \overline{RP} \quad (\text{VIII-2})$$

Soit :

$$(m \cdot n) \cdot v + (\overline{RP} \wedge n) \cdot z \cdot \omega < \lambda(d - ds)/(di - ds) \quad (\text{VIII-3})$$

Les coordonnées des vecteurs n, m, RP par rapport au repère (R, X_r, Y_r, Z_r) lié au robot sont :

$$n = [\cos(\alpha) \quad \sin(\alpha) \quad 0]^T$$

$$m = [1 \quad 0 \quad 0]^T$$

$$RP = [x_p \quad y_p \quad 0]^T$$

alors (VIII-3) s'écrit:

$$(\cos(\alpha)v) + (x_p \sin(\alpha) - y_p \cos(\alpha))\omega < \frac{\lambda(d - ds)}{(di - ds)} \quad (\text{VIII-4})$$

Soient: $A = \cos(\alpha)$

$$B = x_p \sin(\alpha) - y_p \cos(\alpha)$$

$$C = \lambda(d - ds) / (di - ds)$$

Finalement la contrainte (VIII-4) peut être écrite comme suit:

$$Av + B\omega < C \quad \text{ou aussi : } [A \ B] \begin{bmatrix} v \\ \omega \end{bmatrix} < C$$

Si on considère n contraintes actives pour n obstacles on aura l'équation suivante:

$$A_{ppv}V < C_{ppv} \quad (\text{VIII-5})$$

avec :

$$V = \begin{bmatrix} v \\ \omega \end{bmatrix}; \quad A_{ppv} = \begin{bmatrix} A_1 & B_1 \\ \vdots & \vdots \\ A_n & B_n \end{bmatrix} \quad \text{et} \quad C_{ppv} = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix}$$

n : nombre des contraintes actives.

9 EVITEMENT D'OBSTACLES

Dans le cas où le robot mobile détecte par ses capteurs la présence d'obstacles, l'algorithme de contrôle calcule le couple de valeurs (v^*, ω^*) qui permet d'éviter ces obstacles comme montré en figure 9.

Ce couple de commande $u^* = (v^*, \omega^*)$ est le point le plus proche de la commande $u = (v, \omega)$ (cas de parcours libre, calculé par la formule (VI-1)) et appartenant aussi au polygone des vitesses permises PVP définie à ce moment par les contraintes en vitesse imposées par les obstacles (formule VIII-5). Le point le plus proche (v^*, ω^*) donnant la distance minimale est calculé en utilisant les procédures de Matlab associées à la contrainte générale $A_{ppv}V < C_{ppv}$.

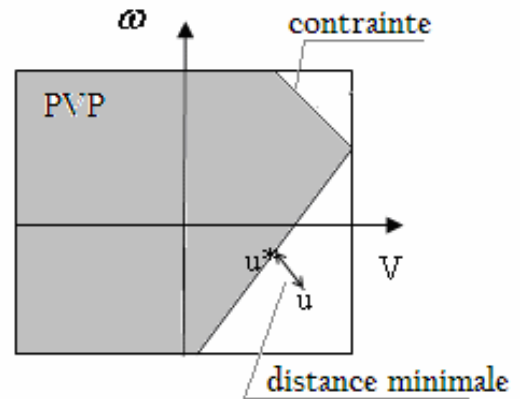
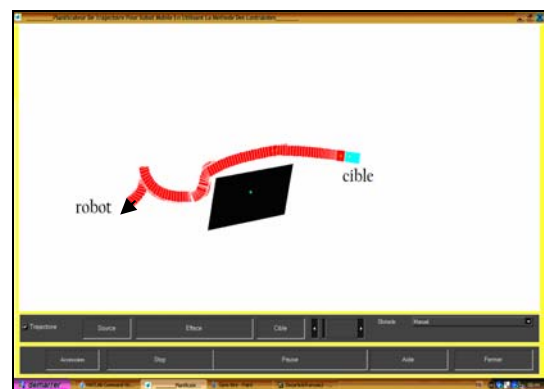
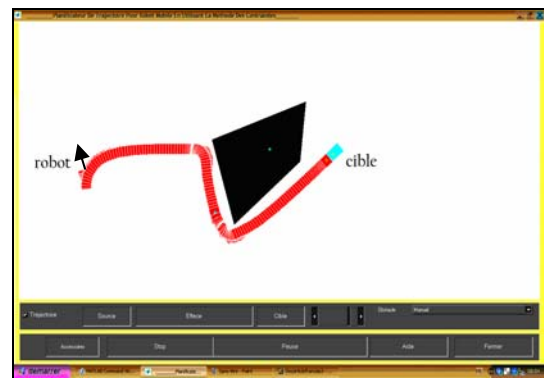


Figure 9: Calcul de $u^* = \text{dis-min}(u, \text{PVP})$



(a)



(b)

Figure 10 : (a),(b) Evitement d'un obstacle

L'application de u^* est valable pour l'évitement d'obstacles, sauf si ce couple (v^*, ω^*) est nul, ce qui provoque le blocage du robot à sa position.

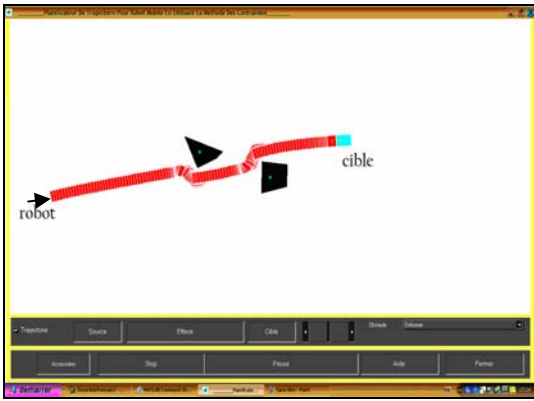


Figure 11 : Evitement de deux obstacles

On montre le long du parcours indiqué en figure 10(a),(b) et la figure 11 que l'évitement est réalisé par le calcul, de u^* permettant de s'éloigner des obstacles et se diriger vers la cible en ligne droite.

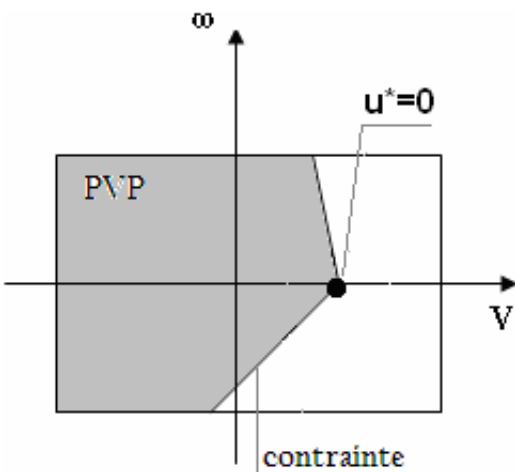


Figure 12 : Cas où $u^*=0$

L'application de la commande u^* n'est pas valable si elle est nulle, cela provoque le blocage du robot à sa position.

10 SITUATION DE BLOCAGE ET RÈGLES HEURISTIQUES POUR LE CONTOURNEMENT DE L'OBSTACLE

Le fait que le vecteur de contrôle u^* s'annule dans certaines situations, sans que cela signifie que le robot est en position finale ; cela veut dire que le robot est en situation de blocage, ce qui exige de résoudre ce problème par un autre algorithme plus approprié que celui appliqué précédemment.

Cet algorithme adopté, est basé sur le polygone des vitesses permises (PVP) et sur des règles heuristiques. L'algorithme doit assurer le contournement de l'obstacle, à droite ou à gauche, et décide de la situation de fin de contournement. Il doit aussi identifier l'obstacle à contourner par le biais de sa contrainte (passe par l'origine) parmi d'autres qui constituent le PVP. A chaque itération de calcul, la contrainte doit être retrouvée pour continuer le contournement de se même obstacle.

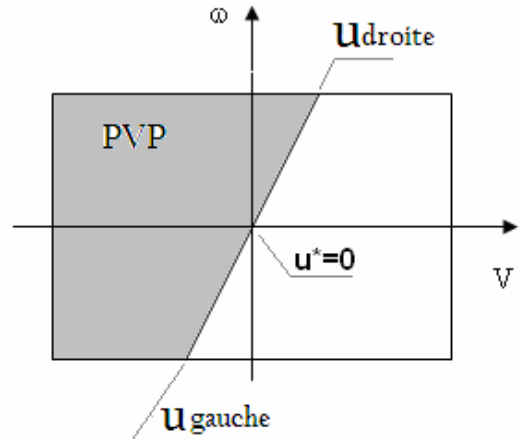


Figure 13 : Blocage du robot (à une contrainte) à cause de $u^*=0$

Le vecteur u recherché à appliquer pour le contournement est l'un des sommets u_{droite} ou u_{gauche} de la contrainte à suivre ; cela permet de suivre le contour de l'obstacle par la gauche ou par la droite [14].

Si l'obstacle bloquant se trouve à gauche le contournement se fait à droite et s'il se trouve à droite le contournement se fait à gauche. Le sens de contournement est gardé jusqu'à la décision de fin de contournement.

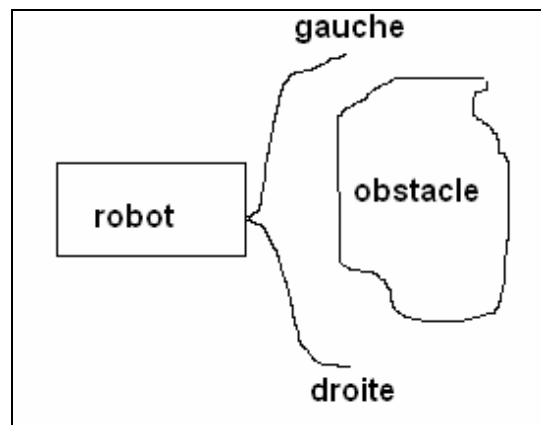
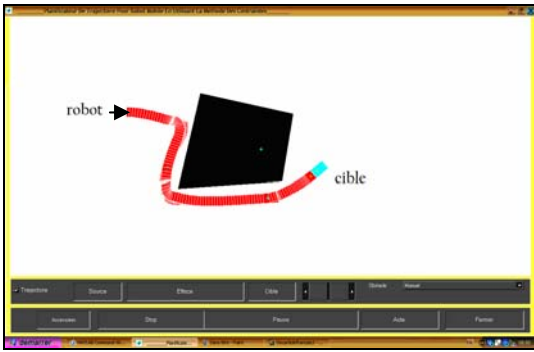
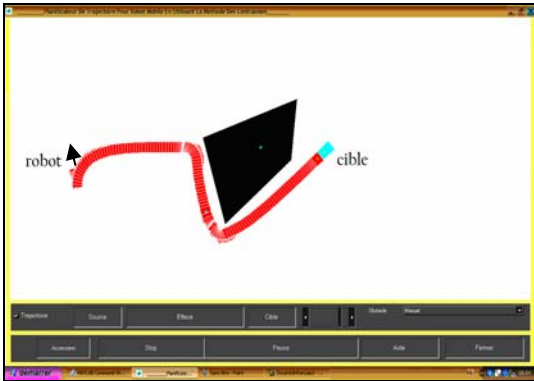


Figure 14 : Illustration du contournement par la gauche et par la droite

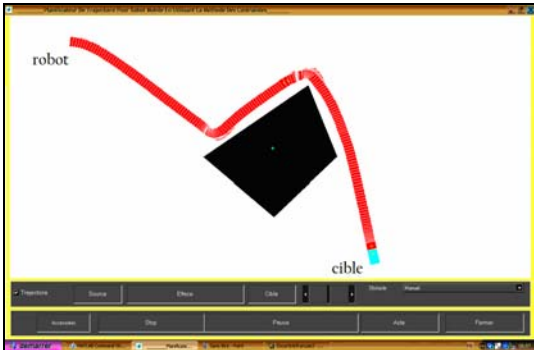


(a)

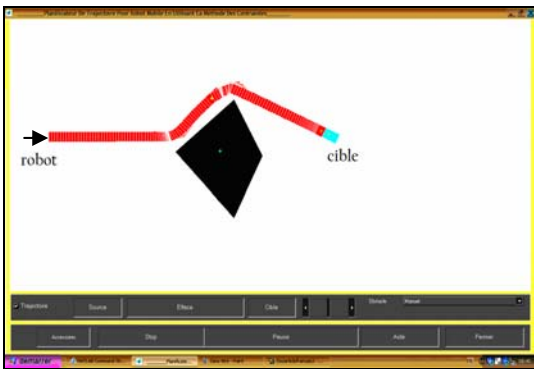


(b)

Figure 15 : Contournement d'un obstacle par la droite



(a)



(b)

Figure 16 : (a,b) Contournement d'un obstacle par la gauche.

Si au cours du contournement d'un obstacle par la gauche ou par la droite la valeur de la commande devient nulle ($u^* = 0 : v^* = 0, \omega^* = 0$) cela indique la présence d'un autre obstacle, donc on bascule vers sa contrainte en gardant toujours le même sens de contournement à droite ou à gauche pour avoir un suivi de contour.

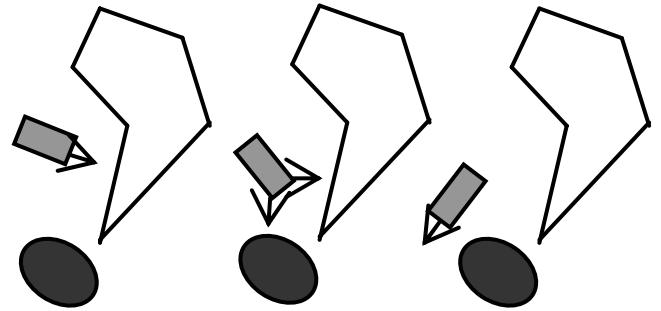
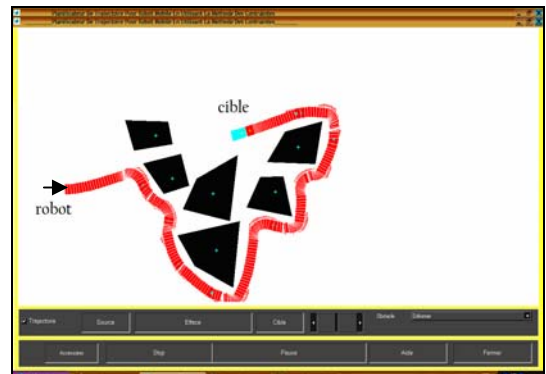
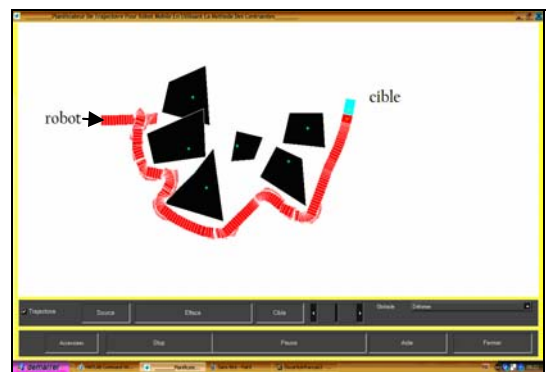


Figure 17 : Illustration du changement d'obstacle à contourner



(a)

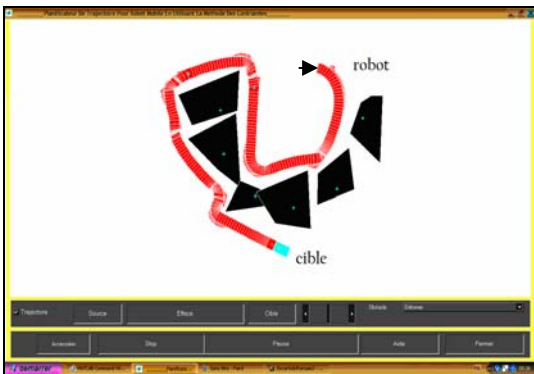


(b)

Figure 18 : (a,b) Contournement de plusieurs obstacles par la droite



(a)



(b)

Figure 19 : (a,b) Contournement de plusieurs obstacles par la gauche

11 DECISION DE FIN DE CONTOURNEMENT

La fin de contournement d'un obstacle est détecté lorsque le robot mobile se dirige vers la cible avec un angle $\pm 0.3rd$ et pas d'obstacles proches ni en face (détecter par capteur1) ni sur le coté gauche (détecter par capteur2) ni sur le coté droite (détecter par capteur3).

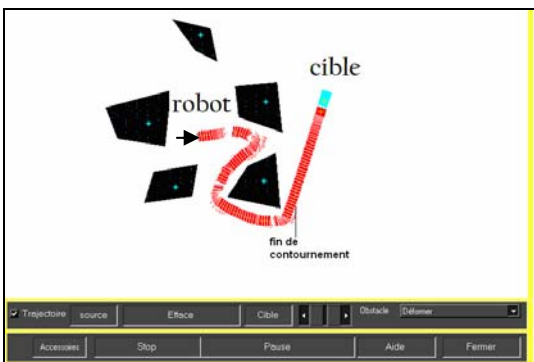


Figure 20 : Décision de fin de contournement

12 LE ROBOT EN LABYRINTHE

Le type des obstacles comme vu précédemment sont des polygones convexes de quatre arrêtes générés d'une façon volontaire ou aléatoire. Un labyrinthe est une succession d'obstacles polygonaux convexes juxtaposés.

Le labyrinthe donne un bon exemple pour confirmer l'efficacité du planificateur.

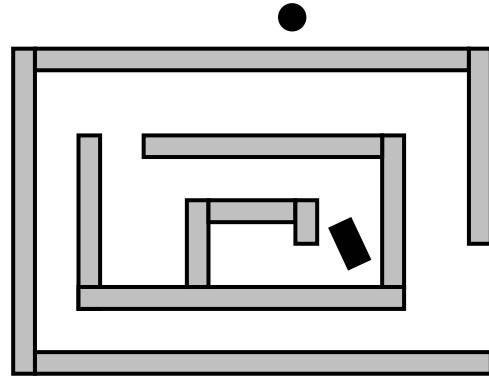


Figure 21 : Illustration de la navigation du robot mobile en labyrinthe

La figure 22 et la figure 23 montrent les résultats de simulation concernant la recherche d'une cible via un labyrinthe.

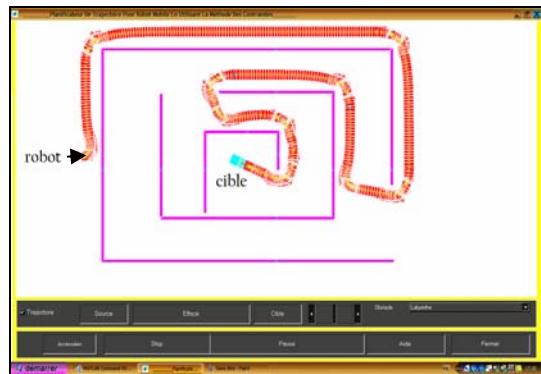


Figure 22 : Le robot à l'extérieur joint la cible à l'intérieur du labyrinthe

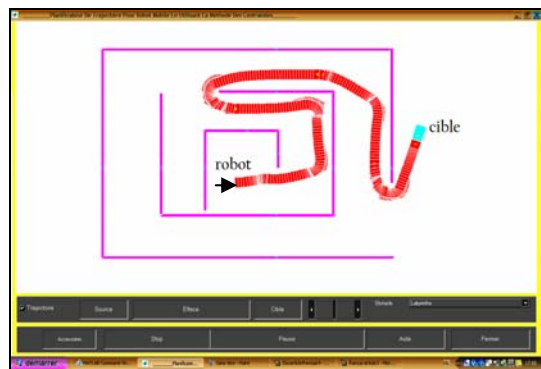


Figure 23 : Le robot à l'intérieur joint la cible à l'extérieur du labyrinthe

13 CAS DE CIBLE MOBILE

Le mouvement de la cible est identifié par le changement de ses coordonnées. L'exigence dans ce cas est que la vitesse de la cible soit plus faible que celle du robot mobile. Les résultats obtenus par simulation sont montrés en figure 24 et en figure 25.

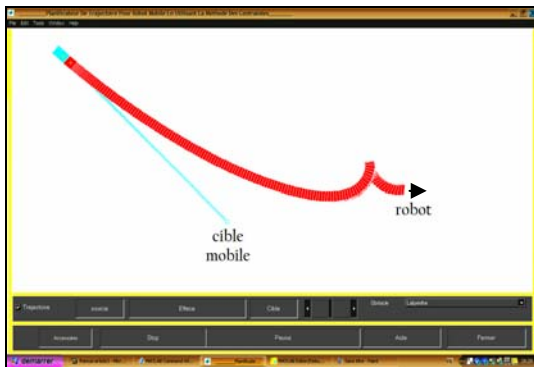


Figure 24 : Poursuite de cible mobile en environnement libre d'obstacles

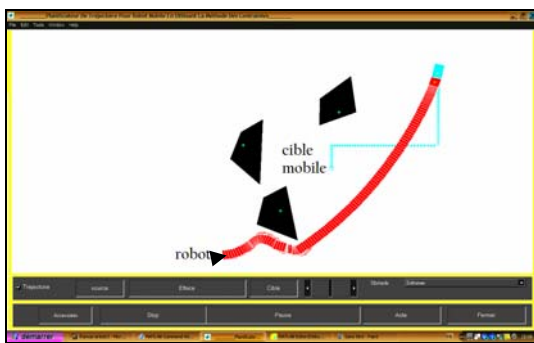


Figure 25 : Poursuite de cible mobile en environnement encombré d'obstacles

14 CONCLUSION

Dans cet article le planificateur intervient sous trois formes ; le cas où il n'y pas des obstacles on utilise une commande qui optimise le parcours pour aller en ligne droite. Dans le cas où il y'a des obstacles on utilise la méthode des contraintes pour construire le PVP et on calcul la commande sur un point du PVP qui assure une distance minimale par rapport au point de commande dans le cas de parcours libre.

Dans le cas où il y'a un blocage du robot derrière un obstacle à cause de la valeur nulle de la commande, on utilise le PVP et un ensemble de règles heuristiques qui permettent de contourner l'obstacle en suivi de mur.

Le planificateur décide de la fin du contournement de l'obstacle si une condition relative à l'orientation du robot envers la cible et la non présence d'obstacles proches est vérifiée.

Le robot mobile peut rattraper une cible mobile si sa vitesse est relativement faible.

L'ensemble des résultats de simulation montrent que se planificateur local permet de franchir des scènes libres d'obstacles, des scènes contenant des obstacles et des labyrinthes.

BIBLIOGRAPHIE

- [1] Ching-Long.S and Jane-Yu.L,'Computing the Minimum Directed Distances Between Convex polyhedra',Journal of information on science and engineering vol15,pp353-373, 1999.
- [2] Ellepola.R and Kovesi.P, 'mobile robot navigation using recursive motion control',IEEE ,pp168-174, 1997.
- [3] Elmer.G.G, W.Johnson.D and Keerthi.S.S ,'A fast procedure for computing the distance between complexes objects in three-dimensional space', IEEE, journal of robotics and automation ,vol.4, no.02,pp193-203, April,1988.
- [4] Faverjon.B, 'Hierchical object models for efficient anti-collision algorithms', IEEE, pp333-340, 1989.
- [5] Faverjon.B and Tournassoud.P, 1987, 'A local based approche for path planning of manipulators with a high number of degrees of freedom ',IEEE , 1987.
- [6] Paulo.C and Urbano.N, April,'Path-following control of mobile robot in presence of uncertainties', IEEE, Transactions on robotics, vol21 .no2,pp252-261,2005.
- [7] Ramirez.G and Zeghloul.S, ' A New local path planner for nonholonomic mobile robot navigation in cluttered environments', IEEE, ICRA, san Francisco pp2058-2063, April 2000.
- [8] Shirong.L, Simon.X.Y and Huidi.Z, 'Adaptive neurons Based control system design for mobile robot', proceedings of 2004 IEEE/RSJ, ICIRS, sendal Japan, pp2636-2641, September 28- Octobre02, 2004.
- [9] Tournassoud.P and Jehl.O, ' Motion planning for a mobile robot with a kinematics constraint', IEEE, pp1785-1790, 1988.
- [10] Patrick G. X, 'Fast Swept-Volume distance for Robust Collision Detection', Proceeding of the 1997 IEEE, ICRA, Albuquerque, New Mexico, April 1997.
- [11] Zhihua Qu, Jing Wang and Clinton.E 'A New Analytical Solution to Mobile Robot trajectory Generation in the Presence of Moving Obstacles' IEEE Transaction on Robotics Vol.20 No.6. Decembr 2004.
- [12] Yasutaka Umeda and Takahiro Yakoh 'Configuration and Readhesion Control for Mobile Robot With External Sensors 'IEEE Transaction on Idustrial Electronics, Vol.49.No.1, February 2002.
- [13] Ji Yeong Lee and Howie Choset 'Sensor-Based Exploration for Convex Bodies: A New Raodamp for a convex-Shaped Robot 'IEEE Transactions on Robotics Vol.21 .No2 April.2005.
- [14] Maaref.H and Barret.C 'Sensor-Based Navigation of a Mobile Robot in an Indoor Environment' Elsevier, Robotics and Autonomous Systems 38 (2002) 1-8.