Asian Journal of Information Technology 6 (11): 1074-1082, 2007
ISSN: 1682-3915

Trajectory Planner for Mobile Robot with Avoidance of Obstacles Based on the Constraints Method and Heuristic Rules

[1]Guesbaya Tahar, [2]Benmahammed Khier and [3]Benali Abderraouf
[1]MSE Laboratory, University of Biskra, Algeria
[2]Department of Electronic, University of Setif, Algeria
[3]LVR Laboratory ENSIB, Bourges, France

Abstract: In this study, the aim is to present our contribution for the local planning of trajectory for nonholonomic mobile robot with avoidance of obstacles. This contribution is based on the constraintsmethod and some heuristics rules. The constraint method put some constraints on the permitted velocities of the mobile robot and obliges it to move away from the obstacles. The mobile robot possesses ultrasonic sensors to measure the minimal distance into the obstacles. Modeling the obstacles and sensors by convex polygons makes use of calculation methods of minimal distance easy. The simulation results show the suppleness with the mobile robot reach the target going across difficult path.

Key words: Mobile robot, nonholonomic, planner, avoidance, constraints method, heuristic rules

INTRODUCTION

Generally alike the manipulator robot which has restricted number of jobs, the mobile robot comes to increase its mobility. Recently, in the industrial field, a wheeled mobile robot with various sensors has been expected as sophisticated and high-performance robot. Much research has been conducted and developments have occurred in this regard.

For the mobile robot the researches concern the kinematics and dynamic modeling; the interactions between the mobile robot and the manipulator; globally and locally paths planning, avoidance static and dynamic obstacles and different techniques control.

In this study , the path planning is locally realized for differential mobile robot, where the 2 rear-wheels droved independently effect its orientation. The kinematics model was used where the nonholonomic constraints are considered. The generating of trajectory, in case of free environment from obstacles, is based law control that optimizes the path mobile robot in a straight line.

In the case of presence of obstacles the constraint method was applied (Faverjon *et al.*, 1987). This method bound the permitted velocities of the mobile robot concluding the avoidance of obstacles. If the mobile robot was blocked in, some heuristic rules must be used skirting around the obstacles and always based on the constraints method.

The obstacles and sensors were modeled geometrically by convex polygons, which facilitate calculation of minimal distance between them (Ching-Long *et al.*, 1999; Patrick, 1997; Elmer *et al.*, 1988). The minimal distance is calculated by Matlab software.

Kinematics modeling: The mobile robot is a platform with two motorized rear wheels mounted on the same axis and controlled independently.

The configuration of mobile robot was shown in Fig. 1.

The considered hypotheses for modeling are:

- The contact wheel-ground is punctual
- The rolling of each wheel must be without a sliding.
- The mobile robot moves on horizontal ground.
- The configuration of the mobile robot was described by 5 general coordinates $q = [x\ y\theta\ \phi_l\ \phi_r]$. And ϕ_l, ϕ_r are angular velocity of left and right wheels.

With those hypotheses we can write for the two wheels the linear velocity's equations (Paulo *et al.*, 2005):

$$\vec{V}_{left} = (\dot{x}\ \cos(\theta) + \dot{y}\ \sin(\theta) - b\dot{\theta})\ \vec{x}_r + (-\dot{x}\ \sin(\theta) + \dot{y}\ \cos(\theta))\ \vec{y}_r \quad (1)$$

$$\vec{V}_{right} = (\dot{x}\ \cos(\theta) + \dot{y}\ \sin(\theta) + b\dot{\theta})\ \vec{x}_r + (-\dot{x}\ \sin(\theta) + \dot{y}\ \cos(\theta))\ \vec{y}_r \quad (2)$$

Corresponding Author: Guesbaya Tahar, MSE Laboratory, University of Biskra, Algeria

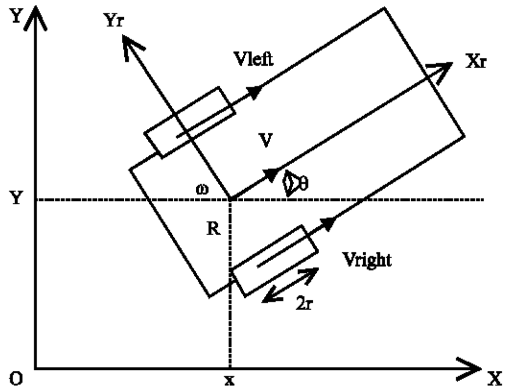


Fig. 1: Kinematics representation of mobile robot

It is evident that the wheels do not move along the r Y axis so the corresponding component for $\vec{V}_{left}$ and $\vec{V}_{right}$ is nil.

$$\text{so: } -\dot{x}\sin(\theta) + \dot{y}\cos(\theta) = 0 \tag{3}$$

The components of $\vec{V}_{left}$ and $\vec{V}_{right}$ for the Xr axis are:

$$\dot{x}\cos(\theta) + \dot{y}\sin(\theta) + b\dot{\theta} = r\dot{\phi}r \tag{4}$$

$$\dot{x}\cos(\theta) + \dot{y}\sin(\theta) - b\dot{\theta} = r\dot{\phi}l \tag{5}$$

Those kinematics constraints can be rewritten as follows:

$$A(q).q = 0 \text{ where } A(q) = \begin{bmatrix} \sin(\theta) & -\cos(\theta) & 0 & 0 & 0 \\ \cos(\theta) & \sin(\theta) & b & -r & 0 \\ \cos(\theta) & \sin(\theta) & -b & 0 & -r \end{bmatrix} \tag{6}$$

In this study, we are interested in the following form, that you can see in Shirong *et al*. (2004):

$$\dot{q} = S(q)V(t) \text{ wher } q = [\,x\ y\ \theta]^T \text{ and } V(t) = [v(t)\ \omega(t)] \tag{7}$$

Where, v is linear velocity and ω angular velocity of the mobile robot around R (origin of marker) then:

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

The sensors modelling: The robot mobile was equipped by eight ultrasonic sensors, installed like shown in Fig. 2 (Ellepola *et al*., 1997) for labmat mobile robot sensors configuration.

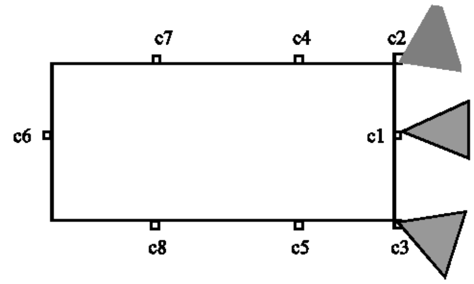


Fig. 2: Illustration of the position of sensors on the mobile robot

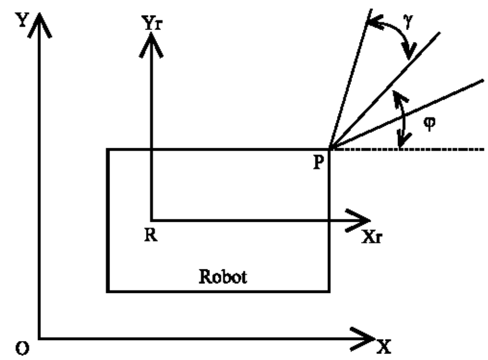


Fig. 3: Sensor parameter

The sensors possess an openness angle of 26°C and field of measure between 16 cm and 10 m.

The sensor was defined by the following parameters:

P = Position of the sensor on the mobile robot.
j = Sensor emission angle regard to Xr axis.
g = Sensor openness angle.

In simulation each sensor was modelled geometrically by two geometric constraints ($y \le a_i\, x + b$) representing the space of cone emission then (Fig. 3):

$$\text{For } i = 1, 2;\ y \le a_i\, x + b_i \text{ or } -a_i\, x + y \le b_i$$

$$\text{Can be written as: } A_{sens}X \le B_{sens} \tag{9}$$

With:

$$A_{sens} = \begin{bmatrix} -a_{1_{sens}} & 1 \\ -a_{2_{sens}} & 1 \end{bmatrix},\ X = [x\ y]^T,\ B_{sens} = \begin{bmatrix} -b_{1_{sens}} \\ -b_{1_{sens}} \end{bmatrix} \tag{10}$$

Obstacles modelling: Each obstacle is considered geometrically like convex polygon. The obstacle is all the area inside the polygon modelled by constraints type $y \le a_i\, x + b_i$.

In this study, obstacle is represented by four geometric constraints then (Fig. 4):

$$\text{For } i = 1, 4;\ y \le a_i\, x + b_i \text{ or } -a_i\, x + y \le b_i$$

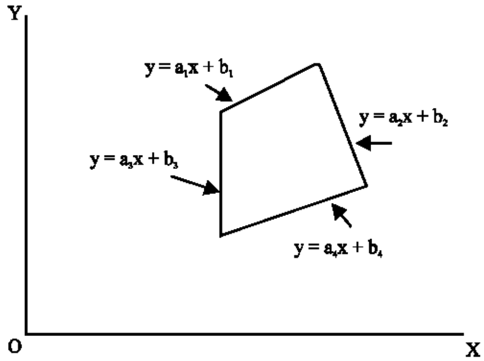


Fig. 4: Modeling the obstacle by geometric constraints ($y \le a_i\, x + b_i$)

Can be written as: $A_{ob}X \le B_{ob}$ (11)

$$A_{ob} = \begin{bmatrix} -a_{1_{ob}} & 1 \\ -a_{2_{ob}} & 1 \\ -a_{3_{ob}} & 1 \\ -a_{4_{ob}} & 1 \end{bmatrix}, X = [x\ y]^T, B_{ob} = \begin{bmatrix} -a_{1_{ob}} \\ -a_{2_{ob}} \\ -a_{3_{ob}} \\ -a_{4_{ob}} \end{bmatrix} \quad (12)$$

MINIMAL DISTANCE CALCULATION

We can say that the sensors detect an obstacle when there is a point that is part of sensor cone emission and obstacle. The distance between the point of emission of sensor $P(x_p, y_p)$ and any point Obs(x,y) of obstacle is:

$$d = [(x - x_p)^2 + (y - y_p)^2]^{1/2} \quad (13)$$

The nearest point Obs(xmin,ymin) that gives the minimal distance min d is found by minimising d under the constraints of the obstacle and the sensor $AX \le B$.
Where:

$$A = \begin{bmatrix} A_{ob} \\ A_{sens} \end{bmatrix} B = \begin{bmatrix} B_{ob} \\ B_{sens} \end{bmatrix} X = [x\ y]^T \quad (14)$$

And:

$$d_{min} = [(x_{min} - x_p)^2 + (y_{min} - y_p)^2]^{1/2} \quad (15)$$

Then the calculation of minimal distance becomes minimisation problem of quadratic criterion under the geometric constraints of the sensor and the obstacle.

The criterion deduced from the distance d (V-1) is:

$$(1/2X^T GX + F^T X) \quad (16)$$

Where:

$$X = [x\ y]^T, G = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, F = [-2x_p\ -2y_p] \quad (17)$$

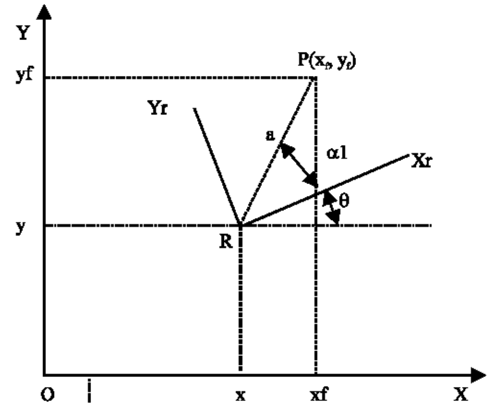


Fig. 5: Kinematics representation of target in polar coordinate

The Matlab software use "quadprog" procedure to resolve the above minimisation problem under the constraints $AX \le B$ and calculate x_{min} and y_{min}.

Control law for navigation in free environment: When the environment is free, the mobile robot path must be optimised in a straight line into the target showed in Fig. 5.

Let $q^f = [x_f, y_f, 0]^T$ the target position and $q = [x, y, \theta]^T$ the current position of mobile robot.

Let $x_e = x^f - x$, $y_e = y^f - y$ and since the number of variables control (v, ω) are two the target must be represented in polar coordinates by a and 1 with:

$$a = (x_e^2 + y_e^2)]^{1/2} \quad (18)$$

$$\alpha 1 = \mathrm{atan2}(y_e, x_e) - \theta, -\pi < \alpha 1 < \pi$$

The mobile robot is reoriented by the control variable w and moved by the variable v to the target. So the following control law was proposed (Ramirez *et al.*, 2000):

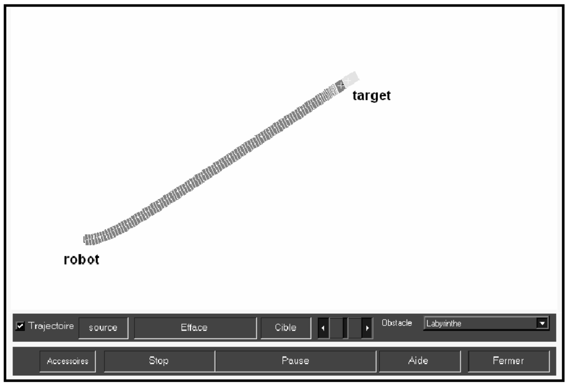
$$v = k_1\, a.\cos(\alpha 1)$$

$$\omega = k_2\, \alpha 1 + k_1 \sin(\alpha 1)\cos(\alpha 1) \quad (19)$$

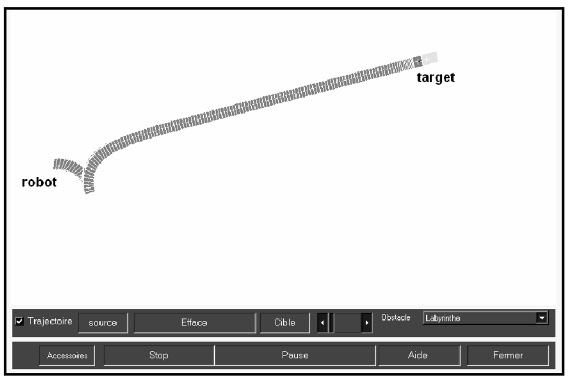
With k_1, k_2 positive constants.

This law control was applied when there is not an obstacle detected in the environment of the mobile robot like showed in Fig. 6a and b.

Constraints method: This method is an alternative of the potential fields method for navigation of manipulator robot with no collision in cluttered environment (Faverjon *et al.*, 1987; Tournassoud *et al.*, 1988; Faverjon, 1989). This method was adapted to the



(a)



(b)

Fig. 6a, b: Navigation of mobile robot in free environment

mobile robot, where the obstacles exercise kinematics constraint on the robot in the velocity's plane. Those constraints oblige the robot to go far of the obstacles. The constraints enter in action when the obstacle is so near to the mobile robot.

Let:

- d = The minimal distance between the obstacle and mobile robot.
- di = The influence distance from which the constraints become active.
- ds = The security distance, representing the smallest distance permitted between the obstacle and mobile robot.
- λ = The adjustment coefficient speed convergence of d to ds.

The anti-collision condition between mobile robot and obstacle is (Fig. 7):

$$\dot{d} \geq -\lambda\ (d\text{-}ds)/(di\text{-}ds) \qquad (20)$$

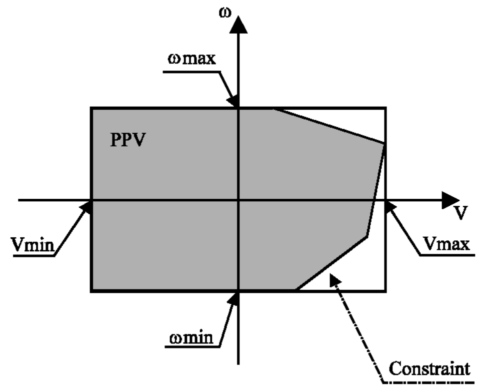


Fig. 7: Polygon of permitted velocities

Where at $t = t_0$ $d = d_0 > ds$

This gives by integrating $\dot{d}$ the evolution of d:

$$d(t) \geq ds + (d_0 - ds)\exp(-\lambda\ (t - t_0)/(di\text{-}ds)) \qquad (21)$$

Sins, d $(t) \geq ds$ for all $t > t_0$ conclude the no collision between obstacle and mobile robot.

When the mobile robot detects through its sensors some obstacles, so we associate for each obstacle a kinematics constraint:

Constraint k:

$$\dot{d}_k \geq -\lambda_k\ (d_k - d_{s_k})/(d_{i_k} - d_{s_k}) \qquad (22)$$

Plus the limitation constraints on v and ω ($|v| < v_{max}$, $|\omega| < \omega_{max}$).

All those constraints constitute the convex Polygon of Permitted Velocities (PPV) that assure the no collision. The (PPV) is always evaluated for each cycle of control.

Representation of constraints in the velocities plane: We must represent the constraints by the mobile robot velocities control, linear v and rotation ω.

From the Fig. 8 we can write:

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

or

$$V_p . n < +\lambda\ (d\text{-}ds)/(di\text{-}ds) \qquad (23)$$

- Vp = Velocity's vector of point P.
- n = Unitary vector along PQ.
- a = Angle between PQ and Xr.
- z = knitary vector along Zr axis.

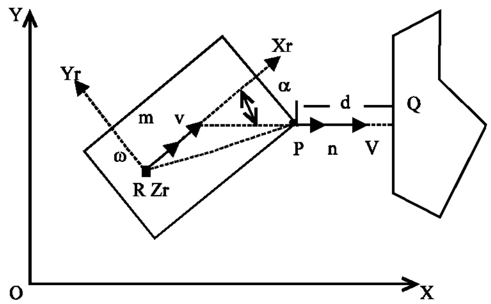


Fig. 8: Representing the constraints by v and ω

With

$$V_p = v.m + \omega.z \wedge \overline{RP} \quad (24)$$

So:

$$(m.n).v+(\overline{RP} \wedge n).z)\ \omega < \lambda\ (d\text{-}ds)/(di\text{-}ds) \quad (25)$$

The coordinate of the vectors n, m, RP in the marker (R, Xr, Yr, Zr) are:

$$n = [\cos(\alpha)\ \sin(\alpha)\ 0]^T \quad m = [1\ 0\ 0]^T \quad RP = [x_p\ y_p\ 0]^T$$

Then:

$$(\cos(\alpha).v)+(x_p \sin(\alpha)\text{-}\ y_p \cos(\alpha))\ \omega < \lambda\ (d\text{-}ds)/(di\text{-}ds) \quad (26)$$

Let:

$A = \cos(\alpha)$

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

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

So finally the constraint can be written like this:

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

So:

$$A_{ppv}\ V < C_{ppv} \quad (27)$$

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

n = Number of active constraints.

OBSTACLES AVOIDANCE

When the mobile robot detects by its sensors the presence of obstacles, the control algorithm calculates the couple (v*, ω*) to avoid them. The couple (v*, ω*) is calculated like the nearest point to the couple(v, w)(in case of free environment determined by the formula 18) and the Polygon of Permitted Velocities (PPV) composed in the same time by the constraints of those detected obstacles (formula 27).The nearest point (v*, ω*) giving the minimal distance is calculated by " Matlab procedure" considering the global constraints A_{ppv} V< C_{ppv} shown in Fig. 9.

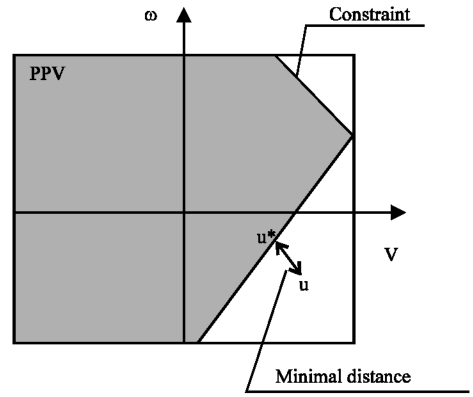


Fig. 9: u* is the command point assuring min-dist between (u and PPV)

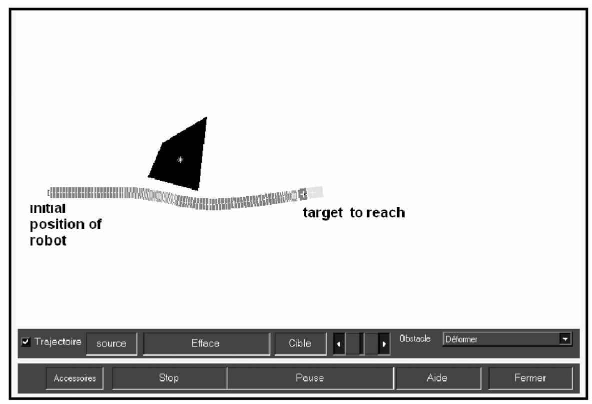


Fig. 10: Avoidance of one obstacle

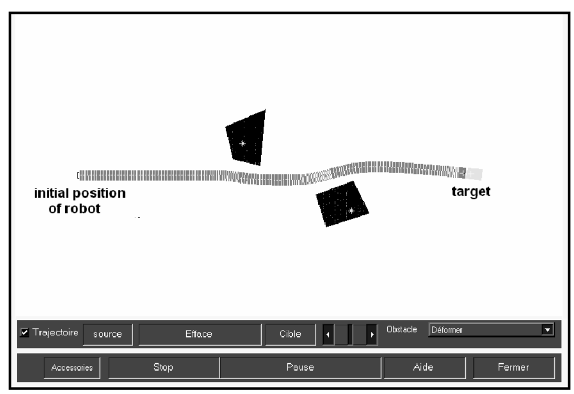


Fig. 11: Avoidance of 2 obstacles

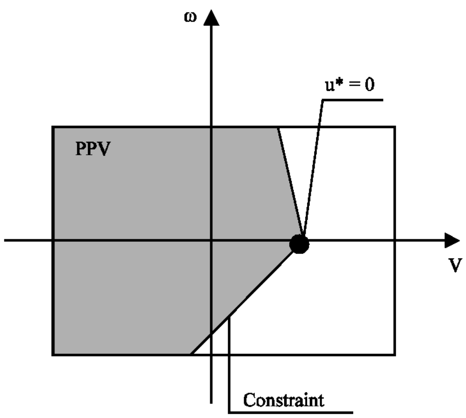


Fig. 12: Case where u* = 0

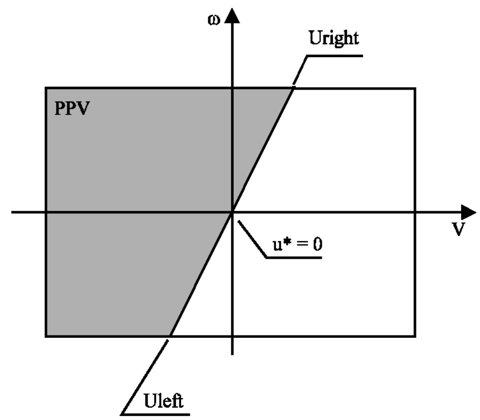


Fig. 13: Blockage of mobile robot caused by u* = 0

We show by the Fig. 10 and 11 that the avoidance is realised by the calculation of u* giving to the mobile robot the possibility to move away from the obstacles and toward the target in straight line.

The application of u* is not valid if it is nil (v* = 0, ω* = 0, Fig. 12) because it causes the blockage of the mobile robot in its position.

BLOCKING SITUATION AND THE HEURISTIC RULES TO CONTOUR THE OBSTACLES

When the mobile robot has not reached the target and the control vector (v*, ω*) is nil, this is the blocking situation and an algorithm must be used to resolve this problem.

This algorithm is based on the polygon of Permitted Velocities (PPV) and some heuristic rules. The algorithm assures that the obstacles will be contoured by the left or right and decides when this operation is finished.

The algorithm must identify the obstacle to contour by its constraint (passing by the origin of the marker of velocities v, w) among the constraints constructing the PPV. For each iteration of calculation, the constraint of the same obstacle must be found to continue to contour it.

The vector u searched to contour the obstacle by the left or the right is one of the summits uleft or uright of its constraint (Fig. 13).

If the obstacle is on the left of mobile robot it wile be contoured by the right and if theobstacle is on the right of mobile robot it will be contoured by the left. The contour sense is conserved until the decision that the obstacle is contoured shown in Fig. 14 and 15.

When we contour the obstacle by the left or the right we calculate u(v, w) and if it will be nil; we conclude the presence of an other obstacle, so we bascule to its constraint with guarding the same contour sense to have the wall following.

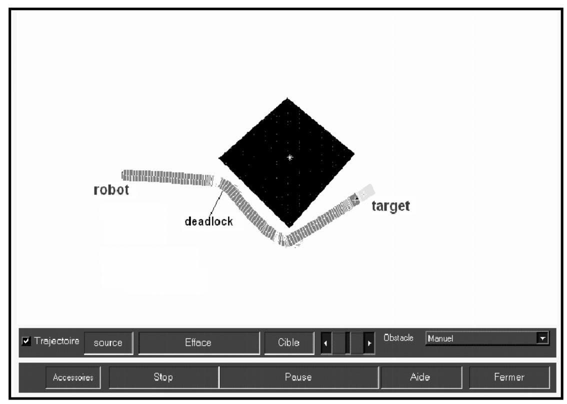


Fig. 14: The obstacle contoured by the right

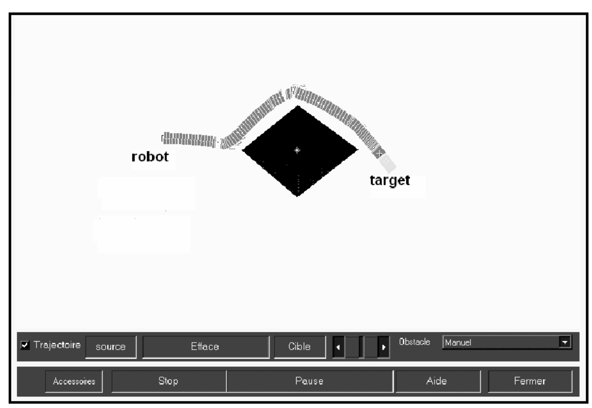


Fig. 15: The obstacle contoured by the left

The Fig. 16-18 illustrate when we change the obstacle to contour.

Decision of the end of contouring an obstacle: In Fig. 19 the decision of end of contouring the obstacle is detected when the mobile robot is in the direction ($-0.3^{rd} < \alpha 1 < 0.3^{rd}$) of target and there is not a near obstacle

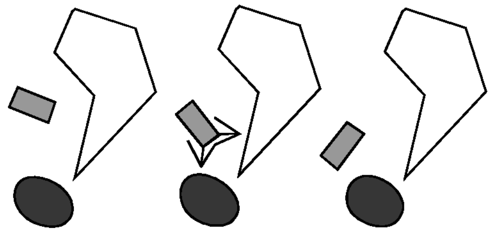


Fig. 16: Illustration of changing the obstacle to contour

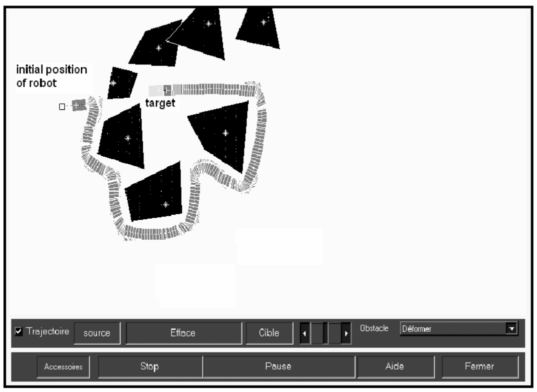


Fig. 17: Contour of some obstacles by the right

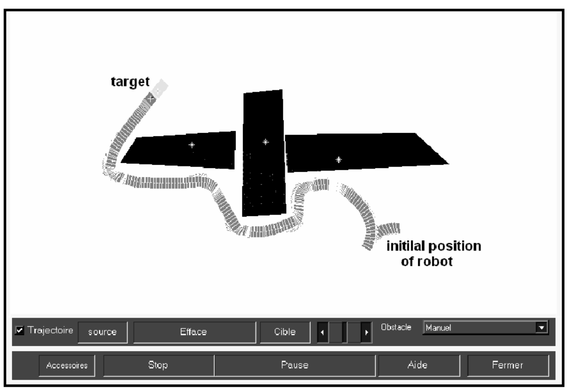


Fig. 18: Contour of some obstacles by the left

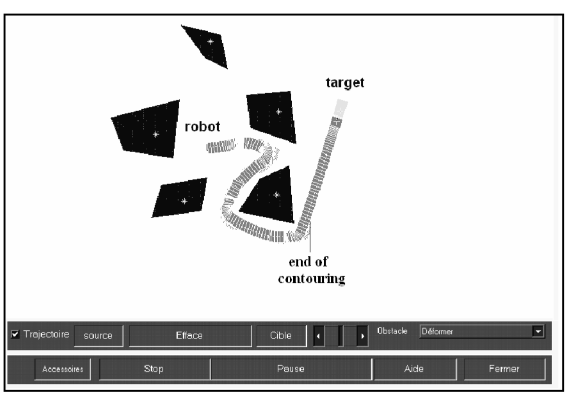


Fig. 19: Situation of the end of contouring the obstacles

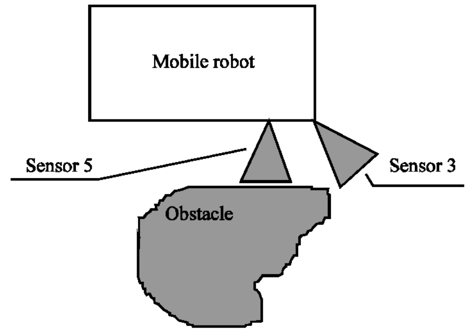


Fig. 20: The obstacle lost by sensor 3 captured by sensor5

in face(detected by the sensor 1) or near the left side (detected by the sensor 2) or near the right side (detected by the sensor 3).

THE SITUATION OF LOSING THE OBSTACLE BY THE SENSOR

An obstacle that exists can be lost bymobile robot if is not in the field of vision of one or more sensors. The solution concerning the active sensor consists on mobile robot to do an appropriate manoeuvre (v+, w+) to find again this obstacle. If the obstacle is not captured by the same active sensor we look to the juxtaposed sensors. This operation is repeated until one sensor captures the obstacle.

SITUATION OF PENETRATION IN TUNNEL

If the mobile robot finds itself in bottom of tunnel where the possibilities to leave it become not evident by using the previousrules; the rules "going out of tunnel" are activated.

Figure 20 shows the blockage in the tunnel is identified when sensor 1, sensor 2, sensor 3, sensor 4 and sensor 5 detect very nearest obstacles. This situation does not permit to move forward or to manoeuvre to turn and go back. The proposed solution is to move back until the obstacles became far toward sensors listed above and after this the mobile robot does rotation for an angle b determined experimentally to not return to the tunnel also shown in Fig. 21 and 22.

The mobile robot in labyrinth: The juxtaposition of obstacles in form of labyrinth gives us a good example to prove the efficiency of the planner.

The simulation results (Fig. 23 and 24) show how the mobile robot joins easily the target inside or outside the labyrinth.

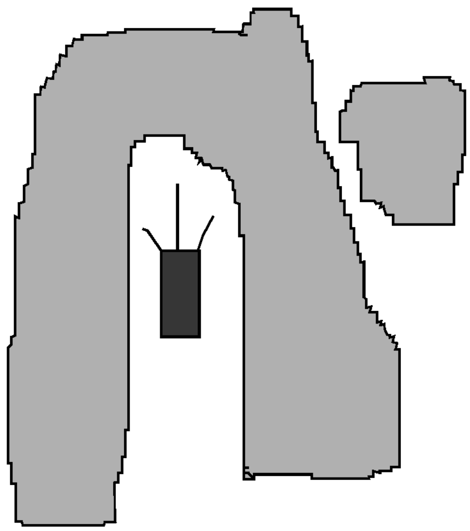


Fig. 21: Illustration of the mobile robot in the bottom of the tunnel

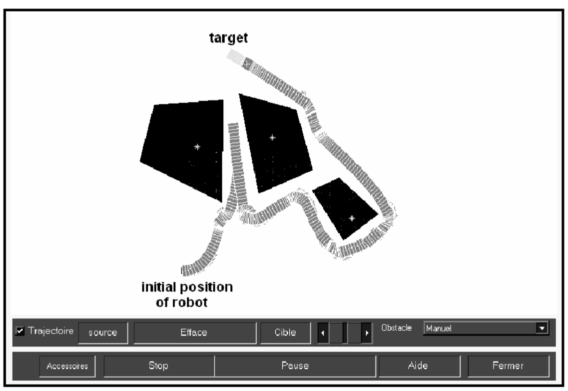


Fig. 22: Escapement of mobile robot from tunnel

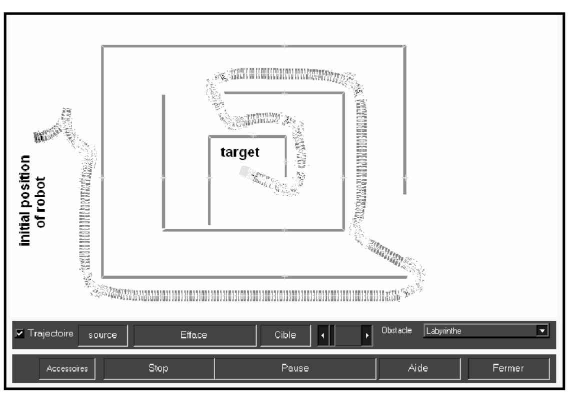


Fig. 23: The mobile robot outside joins the target inside the labyrinth

Case of mobile target: The planner assures to the mobile robot to track the mobile target in free or cluttered environment.

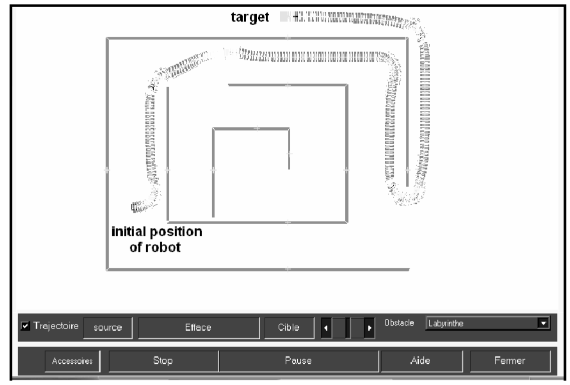


Fig. 24: The mobile robot inside joins the target outside the labyrinth

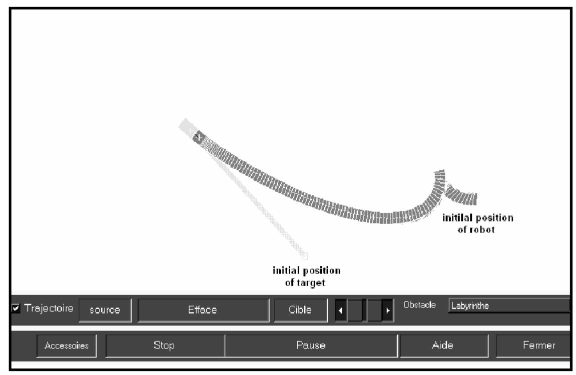


Fig. 25: Tracking the mobile target in free environment

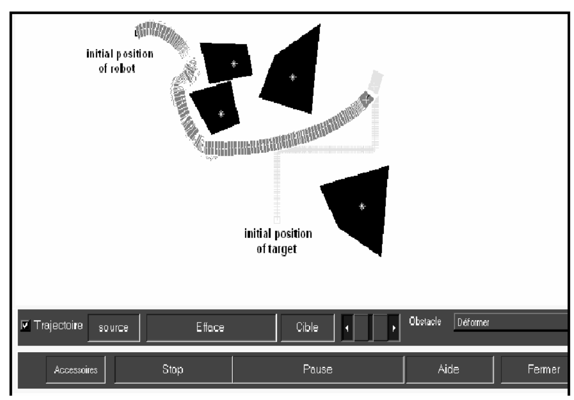


Fig. 26: Tracking the mobile target in cluttered environment

The Fig. 25 and 26 show the mobile robot track and joins the mobile target. The condition is that the target must be slower than the mobile robot.

CONCLUSION

In this study, the planner intervenes on three forms. The case where the environment is free, an optimal control law is used to move the mobile robot in straight line to the target. When the environment is cluttered by obstacles we use the constraints method to construct the PPV and then the control velocities are calculated like assuring the minimal distance between the PPV and the control velocities like if the path is free.

If the mobile robot is blocked behind obstacle because the value of control vector u is nil, we use the PPV deduced from the constraints method and some heuristic rules to contour the obstacle on wall following. The planner decides the end of contouring when some condition, about the mobile robot direction to the target and the no presence of nearest obstacles, is confirmed.
The planner with other rule resolves the problem of escapement from tunnel. The planner permits to the mobile robot to join moving target in free or cluttered environment.

The results of simulation show that this local planner offers to the mobile robot the possibility to go through free path or cluttered path, to go into or out of the labyrinth, to escape from tunnel and finally join the target.

REFERENCES

Ching-Long, S. and L. Jane-Yu, 1999. Computing the Minimum Directed Distances Between Convex polyhedra. J. Inform. Sci. Eng., 15: 353-373.

Ellepola, R. and P. Kovesi, 1997. Mobile robot navigation using recursive motion control. IEEE., pp: 168-174.

Elmer, G.G, W.D. Johnson and S.S. Keerthi, 1988. A fast procedure for computing the distance between complexes objects in three-dimensional space. IEEE. J. Robotics and Automation, 4: 193-203.

Faverjon, B. and P. Tournassoud, 1987. A local Patrick G.X., April 1997. Fast Swept-Volume distance for Robust Collision Detection. Proceeding of the IEEE, ICRA, Albuquerque, New Mexico.

Faverjon, B., 1989. Hierchical object models for efficientanti-collision algorithms. IEEE., pp: 333-340.

Paulo, C. and N. Urbano, 2005. Path-following control of mobile robot in presence of uncertainties. IEEE. Trans. Robotics, 21: 252-261.

Ramirez, G. and S. Zeghloul, 2000. A New local path planner for nonholonomic mobile robot navigation in cluttered environments. IEEE. ICRA. San Francisco, pp: 2058-2063.

Shirong, L., X.Y. Simon and Z. Huidi, 2004. Adaptive neurons Based control system design for mobile robot. Proc. IEEE/RSJ, ICIRS, Sendal Jap., pp: 2636-2641.

Tournassoud, P. and O. Jehl, 1988. Motion planning for a mobile robot with a kinematics constraint. IEEE., pp: 1785-1790.