16th Mediterranean Conference on Control and Automation
Congress Centre, Ajaccio, France
June 25-27, 2008

On-line moving-obstacle avoidance using piecewise Bezier curves with unknown obstacle trajectory

El-Hadi GUECHI, Jimmy LAUBER, Michel DAMBRINE

Abstract—**This paper presents a new technique for moving-obstacle avoidance generating on-line piecewise Bezier curves. Briefly, this technique consists in calculating a new trajectory for the robot if a safety distance is not respected; otherwise the robot follows a straight line to its objective. The piecewise Bezier curve is generated on line with significant control points, one of which is used to avoid the detected obstacle. This avoidance point is calculated automatically according to the instantaneous position of the robot, the moving obstacle and their directions. The tracking control of the trajectory is ensured by a flatness-based feedback control associated to a Kalman-Luenberger observer. In order to show the efficiency of the proposed method, some simulation results are given.**

I. Introduction

THE on-line mobile robot moving-obstacle avoidance is a well-known challenge in autonomous robot control community [1]. In the last years an increasing interest in mobile robots has appeared, notably in the aeronautical space exploration, automatized agriculture and collective mobile-robot games such as robot soccer [2]. These applications ask the mobile robot to circulate in partially-known environments with a high amount of uncertainty. Moving-obstacle avoidance with unknown obstacle trajectory remains a remarkable challenge and has opened a research area in the control of mobile robots, which is the subject of this paper.

Many approaches are available to study this research area. In [3] the authors used a potential field for obstacle avoidance. Nevertheless, in [4] it is shown that this technique produces trapping situations due to local minima, halts between closely spaced obstacles and oscillations in the presence of narrow passages. To solve the problem of trapping several authors use interpolation methods for path planning. In [5] a cubic spline is employed for static obstacle avoidance. However, this interpolation method causes oscillations at the robot trajectory. Authors in [6] use a B-spline trajectory design for the robot, though this approach has a drawback since the generated trajectory does not pass through the exact waypoints [7]. In [8] an AR prediction model is used for moving-obstacle avoidance, though it works only when the robot acceleration is weak and the model a first order system, which is not always the case. A V-obstacle approach for moving-obstacle avoidance has been developed in [9] and [10]. It is based on the knowledge of the obstacle velocity, but in this case the robot rolls are supposed at a constant speed and always following straight-line trajectories. Moreover, the obstacle velocity must be known.

In this paper the avoidance procedure is calculated from a moving obstacle direction with an unknown trajectory. Two on-line C^3 piecewise Bezier curves are employed to avoid it. In [7] the authors use a touchpad interface to determine waypoints, while in our method these points are calculated automatically according to the instantaneous position of the involved robots and the direction of the obstacle. With this method the moving-obstacle avoidance is effectuated on-line and the trajectory obtained is not oscillating thanks to the Bezier curve's properties. Moreover, avoidance trajectories pass through the exact waypoints and do not produce trapping problems. In order to test the efficiency of the algorithm, it is tried in presence of noise on the measurement of the robot position.

This paper is organized as follows: Section 2 describes the system on which this work is based on, developing a kinematics model of the unicycle mobile robot; Section 3 constitutes the core of this work: moving obstacle avoidance strategy; Section 4 introduces an algorithm for flatness-based feedback control; and at the last section simulation results for an on-line moving-obstacle avoidance are shown with and without noise on the position measurement.

II. Description of MIABOT mobile robot

The MIABOT robot [11] is a two-wheeled mobile robot where each wheel is activated independently by an electrical motor (see Fig. 1). This allows the robot to turn around itself and to move forwards and backwards. The latest MIABOT version features bidirectional Bluetooth communications, which provides a robust frequency hopping wireless-communications protocol at 2.4 GHz.

In Fig. 2 a scheme of the unicycle mobile robot is shown, taking into account the non slipping condition, the kinematical model of the unicycle mobile robot in the X-Y plane can be easily written as follows

Manuscript received February 3, 2008. The 16th Mediterranean Conference on Control and Automation – MED'08, June 25-27- Ajaccio – Corsica- France.

El-Hadi GUECHI, Jimmy LAUBER and Michel DAMBRINE are with the LAMIH laboratory UMR CNRS 8530, University of Valenciennes and Hainaut-Cambrésis- Mont Houy, 59313 Valenciennes cedex, France (email: {el-hadi.guechi, jimmy.lauber, michel.dambrine}@univ-valenciennes.fr).

978-1-4244-2505-1/08/$20.00 ©2008 IEEE

$$\begin{cases} \dot{x} = v.\cos\theta \\ \dot{y} = v.\sin\theta \\ \dot{\theta} = w \end{cases} \quad (1)$$

where v, w are respectively the linear and angular speeds of the robot, constituting the control inputs of the mobile robot. The output variables (robot gravity-center position) are (x, y) and θ the angle between the speed vector and the X-axis (i.e., the robot orientation).

Fig. 1. MIABOT Pro mobile robots.

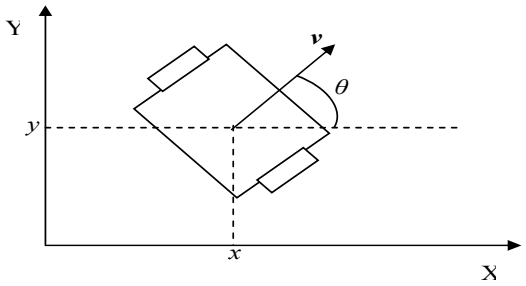

Fig. 2. Unicycle mobile robot

III. MOVING OBSTACLE AVOIDANCE

The proposed method to avoid a moving obstacle consists in four parts described in the following subsections. The information available are the instantaneous positions of the robot and of the moving obstacle.

A. General principle for moving obstacle avoidance

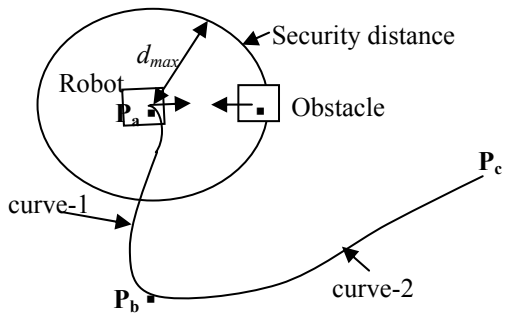

Fig.3. Avoidance strategy

The moving obstacle is avoided by generating in real-time a new path for the mobile robot. This path is entirely defined by three control points: P_a which is the position of the robot when the security distance d_{max} is violated; P_b which is an avoidance point — the manner for determine it is presented in the next subsection; and P_c is the final position of the robot (see Fig. 3). This path is composed of two C^3 Bezier curves: the first one connects P_a to P_b in a way such that the velocity vector is continuous in P_a and maximal in P_b; the second curve connects P_b to P_c and is such that the velocity is continuous in P_b and is zero in P_c. The travel time for each curve is calculated using a heuristic method presented in section *C*. The security distance d_{max} may be related to the capacities of the robot and the obstacle (maximal linear speed, maximal acceleration, size, current velocity) and to the sample period.

B. Determination of the avoidance point

The position of the point P_b has a great influence in the moving obstacle avoidance quality.

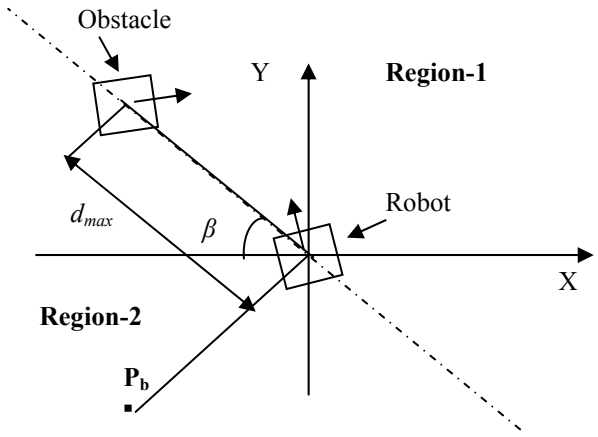

Fig.4. Direction of the obstacle in detection zone of the robot

Firstly, when the security distance is violated, the angle β is determined between the X-axis and the straight line linking the robot position to the obstacle. According to this angle, the workspace can be divided in two half planes (Fig 4). Next, the robot and obstacle directions are determined according to their present and past positions. The avoidance point P_b is fixed so it can belong to a line passing through the robot mass centre and perpendicular to the robot-obstacle axis. The distance between the avoidance point P_b and the robot mass centre is chosen equal to d_{max} and should be situated

1. In the half plane opposite to the obstacle direction if the obstacle velocity vector doesn't point toward the robot mass center;

2. In the region in which the robot velocity points if the obstacle velocity vector points toward the robot mass center but the two velocity vectors are not collinear;
3. In either region, randomly, in the last situation when the velocities of the robot and of the obstacle are collinear.

C. Travel time for each curve

The first section of the avoidance trajectory is a C^3 Bezier curve defined by the expressions:

$$x_1(h) = x_a(1-h)^3 + 3x_{11}h(1-h)^2 + 3x_{12}h^2(1-h) + x_b h^3 \quad (2)$$

$$y_1(h) = y_a(1-h)^3 + 3y_{11}h(1-h)^2 + 3y_{12}h^2(1-h) + y_b h^3 \quad (3)$$

where (x_a,y_a) are the coordinates of the point P_a, (x_b,y_b) are those of P_b, the parameter $h \in [0, 1]$ is defined by:

$$h = \frac{t - t_0}{\Delta t_1}, \quad (4)$$

with t being the current time, t_0 the time when the obstacle enters the security region, t_{f1} the travel time of the first curve, and $\Delta t_1 = t_{f1} - t_0$.

x_{11}, x_{12}, y_{11}, y_{12} are determined such that the velocity is continuous in P_a. These formulas are as follows:

$$x_{11} = \frac{\Delta t_1 . v_{x_a} + 3 . x_a}{3}, x_{12} = \frac{3 . x_b - \Delta t_1 . v_{x_b}}{3},$$

$$y_{11} = \frac{\Delta t_1 . v_{y_a} + 3 . y_a}{3}, \quad y_{12} = \frac{3 . y_b - \Delta t_1 . v_{y_b}}{3}$$

where v_{x_a}, v_{y_a} are the linear speeds corresponding to the X- and Y-axis at P_a, v_{x_b}, v_{y_b} are the linear speeds corresponding to the X- and Y-axis in P_b. The linear speed is chosen as a vector collinear to the segment P_aP_b with norm equals to v_{max}.

The time t_{f1} is fixed to d_{max}/v_{max}. The motivation for this is the consideration of the worst possible situation for which the avoidance must be fast: at the detection time t_0, the obstacle moves on a straight line towards the robot and, this, at a constant speed equals to the maximum velocity v_{max} of the robot (Fig 5). At the time t_{f1}, the obstacle is arrived at P_a, the position of the robot at the detection time, and the robot is near P_b. Since P_b is the fixed desired position, the theoretical linear speed has to take greater value than v_{max}, but of course, it will be saturated at its maximal value in practice.

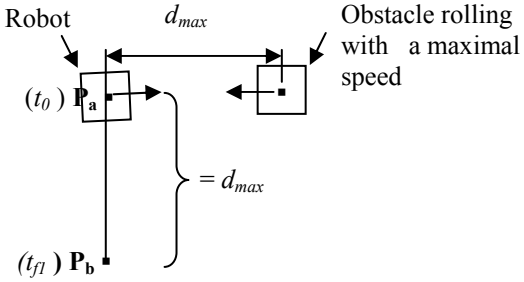

Fig.5. Case when a security distance is violated

The second piece of the trajectory is also a C^3 Bezier curve defined by the expressions

$$x_2(h) = x_b(1-h)^3 + 3x_{21}h(1-h)^2 + 3x_c h^2(1-h) + x_c h^3 \quad (5)$$

$$y_2(h) = y_b(1-h)^3 + 3y_{21}h(1-h)^2 + 3y_c h^2(1-h) + y_c h^3 \quad (6)$$

The parameter h is equal to $h = \frac{t - t_{f1}}{\Delta t_2}$ such as t_{f2} is the travel time for the second curve and $\Delta t_2 = t_{f2} - t_{f1}$.

x_{21}, y_{21} are determined so that the total trajectory is continuously differentiable at t_{f2} and the robot velocity vanishes in P_c. These formulas are presented here:

$$x_{21} = \frac{\Delta t_2 . v_{x_b} + 3 . x_b}{3}, y_{21} = \frac{\Delta t_2 . v_{y_b} + 3 . y_b}{3}.$$

t_{f2} is calculated in order to obtain the maximal linear speed of the robot v_{max}. For this, note that the robot linear speed in the second part of the trajectory is given by

$$v(t) = \dot{h}(t)\bar{v}(h(t)), \quad (7)$$

where $\bar{v}(h) = \left((x_2'(h))^2 + (y_2'(h))^2\right)^{1/2}$

(the prime symbol denotes the derivative with respect to h). Let $\bar{v}_{max} = \max_{0 \le h \le 1} \bar{v}(h)$, then the travel time is chosen as follows:

$$t_{f2} = t_{f1} + \frac{\bar{v}_{max}}{v_{max}} \quad (8)$$

In this expression, $\bar{v}_{max}$ is computed considering that the linear speeds in the points P_b and P_c are zero, so it will never be greater than v_{max} along the trajectory. Nevertheless, in our case the linear speed in the point P_b is different from zero, so using the time t_{f2} given by equation (8) can lead to obtain a linear speed slightly greater than v_{max}. An exact solution for the time t_{f2} could be found using some numerical schemes to ensure that v_{max} won't be overtaken, but this scheme is unable to cope with real time control, since the necessary computation may turn out too long. Moreover, as the real speed of the robot is saturated and as a closed loop strategy is considered, the time given by equation (8) allows obtaining an accurate trajectory tracking.

D. Moving obstacle avoidance algorithm

The moving obstacle trajectory is unknown in general, being its position at every sample time the only information available to avoid it. The algorithm can be described as follows:

1. Initially the robot follows a given path (say a straight line) leading towards the goal point P_c.
2. If the distance between the robot and the mobile obstacle is above the security distance d_{max}, then the robot follows its planned trajectory, else it quits its previous path by following another trajectory calculated on-line to avoid the moving obstacle. As explained in section III, part *A*, the new trajectory is composed of two C^3 Bezier curves. In the first part, the detection test is inhibited, and then it goes back to the detection test (i.e., step 2).

The reference trajectory generated on-line is continuous and differentiable. To follow it, a feedback control based on flatness is applied, as explained in the following section.

IV. FLATNESS-BASED FEEDBACK CONTROL

Let us consider a nonlinear model:

$$\dot{x} = F(x,u) \tag{9}$$

System (9) is flat [12] if it exists a vector S of the same dimension as u, called the flat output, of the form

$$S = h(x,u,\dot{u},\ldots,u^{(r)}) \tag{10}$$

such that the control vector u and the state vector x can be expressed as functions of S and a finite number of its derivatives:

$$\begin{aligned} u &= f_1\left(S,\dot{S},\ddot{S},.....,S^{(q)}\right) \\ x &= f_2\left(S,\dot{S},\ddot{S},.....,S^{(q)}\right). \end{aligned} \tag{11}$$

Robot model (1) is flat with x and y as flat outputs. The open loop control is given by the expressions below:

$$v = \sqrt{\dot{x}_d^{\,2} + \dot{y}_d^{\,2}} \tag{12}$$

$$w = \frac{\ddot{y}_d \dot{x}_d - \ddot{x}_d \dot{y}_d}{\dot{x}_d^{\,2} + \dot{y}_d^{\,2}} \tag{13}$$

where v is the linear speed and w is the angular speed, and x_d and y_d define the desired robot trajectory.

In order to obtain a feedback control law, by differentiating the model equation (1), the following expressions arise

$$\begin{cases} \ddot{x} = u_1 \\ \ddot{y} = u_2 \end{cases} \tag{14}$$

with

$$\begin{cases} u_1 = \dot{v}\cos\theta - vw\sin\theta \\ u_2 = \dot{v}\sin\theta + vw\cos\theta \end{cases} \tag{15}$$

The control law $\dot{v}$ and w are obtained by inversion of this system and considering the linear state feedback for (14):

$$\begin{cases} u_1 = k_1\left(x_d - x\right) + k_2\left(\dot{x}_d - \dot{x}\right) \\ u_2 = k_1\left(y_d - y\right) + k_2\left(\dot{y}_d - \dot{y}\right) \end{cases} \tag{16}$$

where k_1, k_2 are stabilizing gains which are calculated by pole placement method for the linear model (14).

V. SIMULATION RESULTS

For the simulation, the maximal linear speeds of the robot and the obstacle were fixed to 0.3 m/s. The robot and the obstacle have a square shape with a 0.01 m^2 surface. The security distance has been fixed to 0.3 m.

A. Case of perfect measurement

1) Case of lateral avoidance

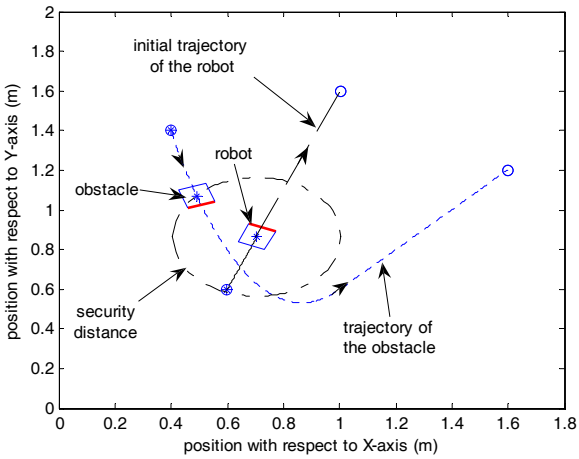

Fig.6. Situation at $t = t_0$: the obstacle enters into the security region around the robot

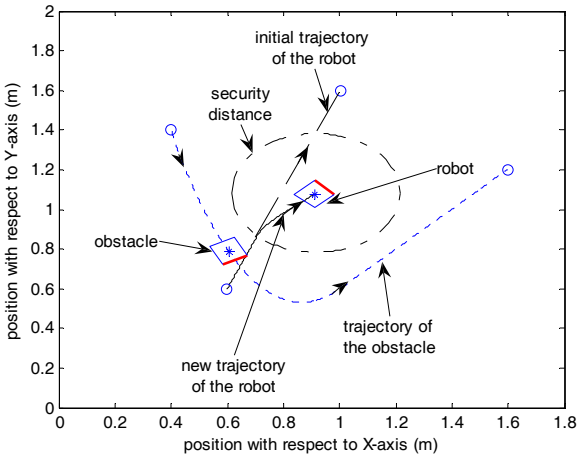

Fig.7. Situation at $t = t_{f1}$: end of the first part of the new planned trajectory

The robot position is assumed to be measured without noises. Figs. 6 – 8 represent the covered trajectory of the robot in different situations: when the obstacle enters into the security region at $t = t_0 = 1.82$ s (Fig. 6), when the robot have covered the first part of the avoidance trajectory (at $t = t_{f1} = 2.83$ s – Fig. 7), and at last, when the robot reaches its objective (Fig. 8). The linear speeds of the obstacle and the robot are depicted in (Fig. 9). In this figure a difference between the real robot speed and that of the reference trajectory can be observed. This difference results from the saturation effect, but it hasn't a remarkable influence on the followed trajectory.

In Fig. 10, the relative distance between the robot and the obstacle is plotted. It can be seen that after the time t_0, this distance goes below the security distance. This can be explained since, at t_0, the situation is almost the worst possible: the obstacle and the robot are going towards each other with the obstacle speed reaching its maximum value.

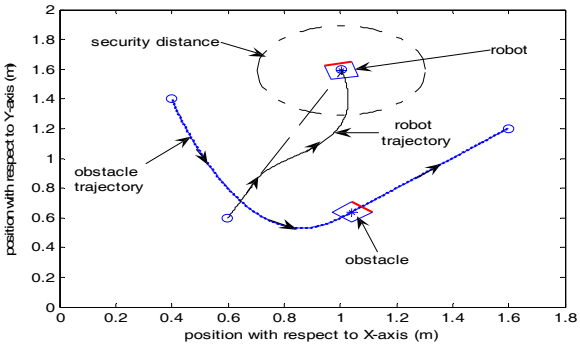

Fig.8. Final situation: the robot reaches its objective point.

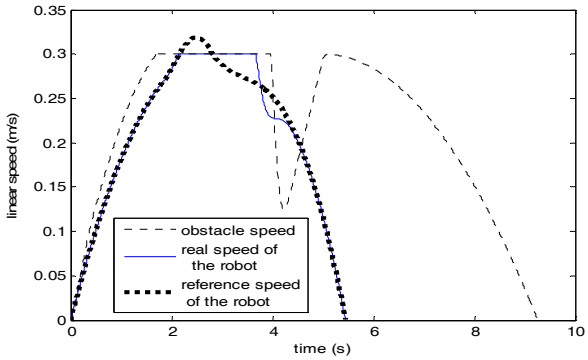

Fig.9. linear speeds of the obstacle and the robot

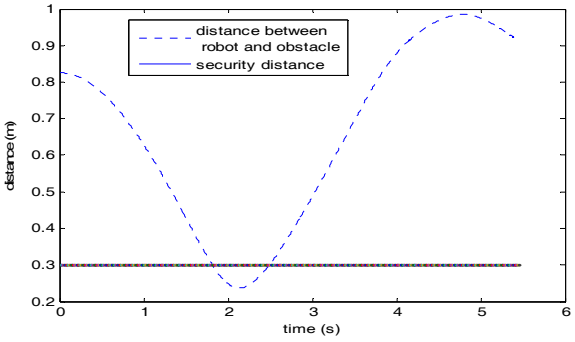

Fig.10. Distance between the obstacle and the robot along navigation this last between its initial and final position

2) *Case of frontal avoidance*

The case when the robot and the obstacle are face to face is presented here. Figs. 11-13 show the trajectory of the robot in these significant situations. The robot enters into the security region at $t = t_0 = 1.95\,\mathrm{s}$ (Fig. 11). It reaches the end of the first avoidance curve at $t = t_{f1} = 2.96\,\mathrm{s}$ (Fig. 12). Finally, Fig. 13 correspond to the situation when the robot reaches its objective. The linear speeds of the robot and the obstacle are depicted in Fig. 14.

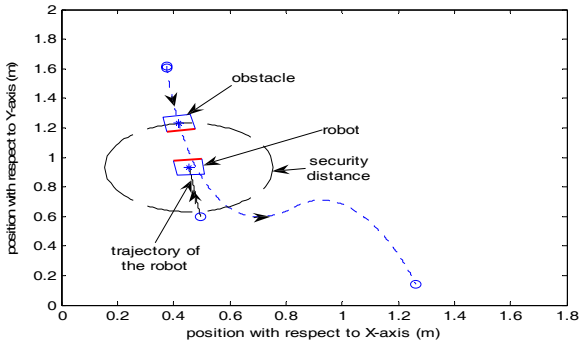

Fig.11. Situation of the robot and the obstacle at $t = t_0 = 1.95$ (s)

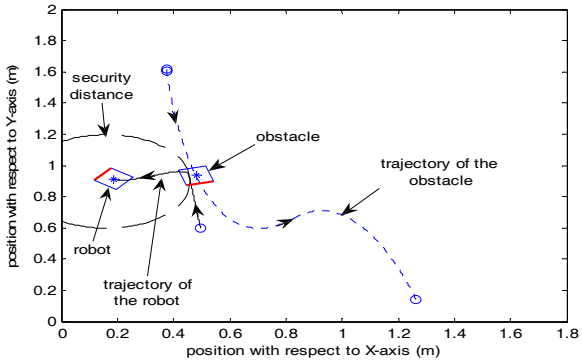

Fig.12. Situation of the robot and the obstacle at $t = t_{f1} = 2.96$ (s)

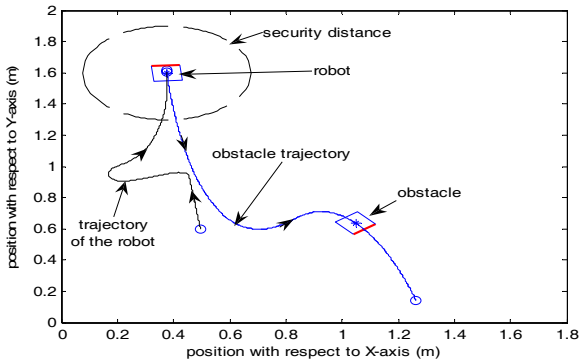

Fig.13. Final situation: the robot reaches its objective point.

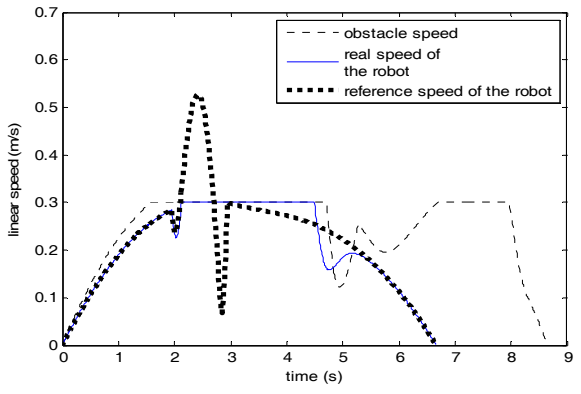

Fig.14. Linear speeds of the obstacle

B. *Simulation with noise on the measures*

In reality, the measures obtained by a camera could be noisy. Hereby it is shown that the on-line moving obstacle avoidance still works in the presence of noise at the position measurement. The states are estimated by a Kalman-Luenberger observer. A uniform noise signal in the range of ±0.03 m for the position has been applied.

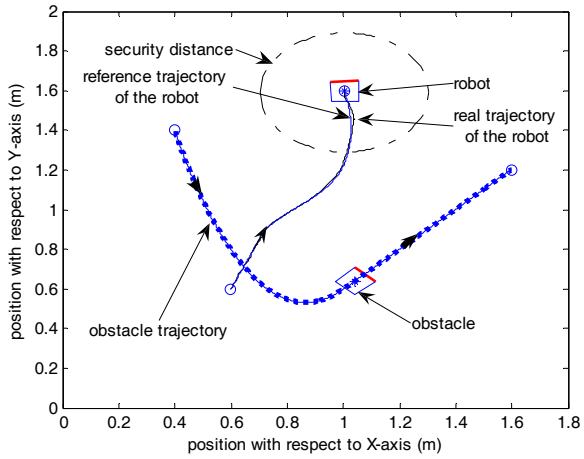

Fig.15. Moving obstacle avoidance: case of noisy measurements of the robot position

In Fig. 15 the robot follows a reference trajectory in spite of the presence of noise, which means that the control law is not very sensitive with respect to noise. In Fig. 16 the real linear velocities of the obstacle and the robot when the measurement of the robot position is noisy are shown.

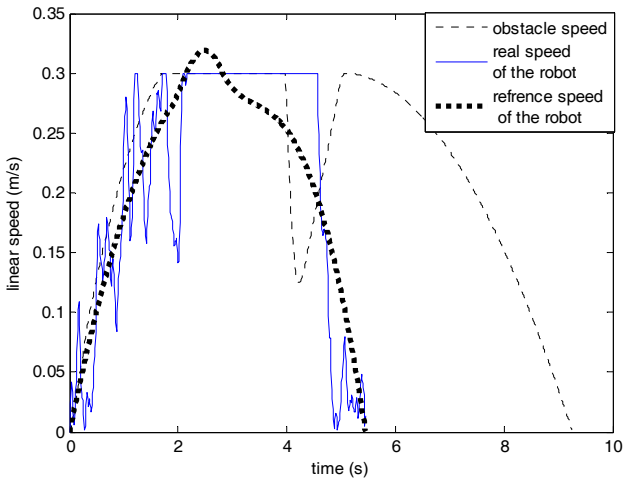

Fig.16. Linear speeds of the robot and the obstacle (m/s).

VI. CONCLUSION

Using trajectories composed by two C^3 piecewise Bezier curves, the present article shows an on-line moving-obstacle avoidance algorithm which employs flat control laws to follow the reference trajectories. Effective simulation results have been obtained in both noisy and noiselessly measures. The presented moving-obstacle avoidance has no problems of local minima or oscillating trajectories, hereby overcoming existing results on the same field. Future work is aimed to predict the moving-obstacle trajectory so it can be optimal for some criteria (distance, time, etc), and real-time implementation.

ACKNOWLEDGMENTS

This work is supported in part by the Region Nord Pas-de-Calais and the **FEDER** Fonds Européen de Développement Régional (European Funds of Regional Development).

REFERENCES

[1] C. Fulgenzi, A. Spalanzani, C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy gird". *2007 IEEE Int. Conf. on Robotics and Automation, ICRA 2007, Roma, Italy,* 10-14, April. 2007, pp. 1610–1616.

[2] P. Vadakkepat, X. Peng, Q. B. Kiat, L. T. Heng, "Evolution of fuzzy behaviors for multi-robotic system". *Robotics and Autonomous Systems.* vol. 55(2). February 2007, pp. 146-161.

[3] O. Khatib, "Real-time obstacle avoidance for manipulator and mobile robots". *The Int. J. of Robotics Research.* vol. 5(1), 1986, pp. 90-98.

[4] Y. Koren, J. Borenstein, "Potential field method and their inherent limitations for mobile robot navigation". *Proceedings of the IEEE Conference on Robotics and Automation,* Sacramento, California, 7-12 April 1991, pp. 1398-1404.

[5] M.Haddad, T.Chettibi, S.Hanchi, H.E.Lehtihet, "A random-profile approach for trajectory planning of wheeled mobile robots". *European journal of Mechanics-A/Solids, Vol. 26, Issue 3, May-june 2007, pp 519-540.*

[6] K. Komoriya, K. Tanie, "Trajectory design and control of wheel-type mobile using B-spline curve". *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems., Tsukuba, Japan, 4-6, September 1989, pp. 398-405.*

[7] Jung-Hoon Hwang,, R.C . Arkin, Dong-Soo Kwon "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control". ". *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2003) Las Vegas, USA,* 27-31 October 2003, vol.2, pp. 1444–1449.

[8] Z. Hui-Zhong, D. Shu-Xin, W. Tie-jun, "On-line real time path planning of mobile robots in dynamic uncertain environment". *Journal of Zhejiang University Science A, Vol. 7(4), April 2006, pp. 516-524.*

[9] P. Fiorini, Z. Shiller, "Motion planning in dynamic environments using velocity obstacle". *Int. J. of Robotics Research, Vol. 17(7), 1998, pp. 760-772.*

[10] F. Large, "Navigation autonome d'un robot mobile en environnement dynamique et incertain". *Thèse à l'Université de Savoie* ,2003.

[11] *Merlin Systems Corp. Ltd.* http://www.merlinrobotics.co.uk/.

[12] M. Fliess, J. Levine, Ph. Martin and P. Rouchon, "Flatness and defect of nonlinear systems: introductory theory and examples", *Int. J. Control*, vol. 61(6), 1995, pp. 1327-1361.

[13] R. Kindel, D. Hsu, J-C. Latombe, S. Rock, "Kinodynamic motion planning amidst moving obstacles". *IEEE Int. Conf on Robotics and Automation, ICRA 2000, San Francisco, USA,* 24-28 April 2000, pp. 537-543.