# A Study on Path Planning Algorithm of a Mobile Robot for Obstacle Avoidance using Optimal Design Method

Anh-Kim Tran*, Jin-Ho Suh, Kwang-Ju Kim, and Sang-Bong Kim

* Department of Mechanical Engineering, Pukyong National University, Pusan, Korea
(Tel : +82-51-620-1606; E-mail: tanhkim@yahoo.com)

**Abstract**: In this paper, we will present a deeper look on optimal design methods that are related to path-planning for a mobile robot. To control the motion of a mobile robot in a clustered environment, it's necessary to know a suitable trajectory assuming certain start and goal point. Up to now, there are many literatures that concern optimal path planning for an obstacle avoided mobile robot. Among those literatures, we have chosen 2 novel methods for our further analysis. The first approach [4] is based on HJB(Hamilton-Jacobi-Bellman) equation whose solution is the return-function that helps to generate a shortest path to the goal. The later [5] is called polynomial-path-planning approach, in this method, a shortest polynomial-shape path would become a solution if it was a collision-free path. The camera network plays the role as sensors to generate updated map which locates the static and dynamic objects in the space. Therefore, the exhibition of both path planning and dynamic obstacle avoidance by the updated map would be accomplished simultaneously. As we mentioned before, our research will include the motion control of a true mobile robot on those optimal planned paths which were generated by above algorithms. Base on the kinematic and dynamic simulation results, we can realize the affection of moving speed to the stable of motion on each generated path. Also, we can verify the time-optimal trajectory through velocity tuning. To simplify for our analysis, we assumed the obstacles are cylindrical circular objects with the same size.

**Keywords:** Obstacle Avoidance, Mobile Robot, Optimal Control, Hamilton-Jacobi-Bellman(HJB), Return Function.

## 1. INTRODUCTION

Up to now, there are many methods on theme of path planning for obstacle avoidance mobile robot. One of the most efficient of those, say, based mainly on potential fields [1][2]. Representing the free space as an attractive potential, which pull the robot toward the goal configuration, and the occupied space as a repulsive potential, pushing the robot away from obstacles. This potentials, however, address only the obstacle avoidance problem with no concern for path optimality.

One novel approach to the on-line shortest path problem, it was motivated by HJB theory [4][6]. The *pseudo*return function is solved in given environment and helps to generate an incrementally trajectory permitting robot motion before the entire path to the goal has been completely computed. Hence, this method is very powerful that the path is modified instantly in response to updated information of environment. It would become more practical if it's required that path should be as *smooth* as possible and it should not give any anxious motion to mobile robot. The polynomial curves are well-known as having a number of advantages in planar robot path planning such as smooth, easy to calculate, the curvature can be traded off against the curve length... In the Polynomial Path Planning Approach [5], a path planner that utilizes polynomial curve which would also be applicable and suitable for common indoor floor-plans.

Our goal in this paper is to analyse the effect of velocity when controlling a real mobile robot to move on those planned paths to the optimality and stability. Both the kinematics and dynamics of a mobile robot would be considered in our controller. As a result, we can decide a best path that reflects stability and time-optimality without obstruct to the mobile robot's capability.

We demonstrated our research by using the cylindrical circular obstacle throughout the calculations and simulations.

This paper is organized as follows: some methods for obstacles avoided path-planning are briefly introduced in section 2. Motion controller on a planned path for a real mobile robot is introduced in section 3. The simulations' results are compared and analysed in section 4 and the final section are our conclusions and future work to be mentioned

The construction of experimentation is under proceeding that will serve as a mean to realize this applicable optimal design method.

## 2. OPTIMAL PATH PLANNING ALGORITHMS FOR AN OBSTACLE AVOIDED MOBILE ROBOT

In this section, we present some of advanced methods that relate optimal path planning with obstacle avoidance. The circular obstacles are chosen for the formulation.

### 2.1 Optimal Obstacle Avoidance Based on the Hamilton-Jacobi-Bellman Equation

Return function introduced by Hamilton-Jacobi-Bellman (HJB) equation shows cost of moving to target point or the shortest length path from a certain point to target point.

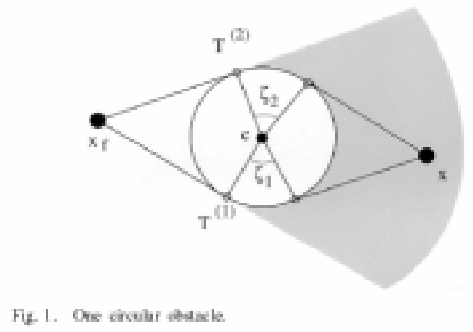



Fig. 1 One circular obstacle

Consider a circular obstacle, denoted OB, with radius r and center at $c \in R^2$, we define an area S as obstacle shadow as follows:

$$S = \{x : \angle x \in [\angle T^{(1)}, \angle T^{(2)}], \|x - c\|^2 \geq r^2, \|x - x_f\| \geq \|c - x_f\|\} \quad (1)$$

where $\angle x$ is the angle made by x with the x-axis, and $T^{(i)} \in R^2$, i=1, 2, are the points of contact on OB, of the two tangents from the goal. Return function v(x,c,r) of an arbitrary point $x \in R^2$ is equal to length of optimal path. Return function can be written as:

$$v(x,c,r) = \begin{cases} w_c(x,c,r) & \text{if } x \in S \\ \|x - x_f\|, & \text{if } x \notin S \end{cases} \quad (2)$$

where

$$w_c(x,c,r) = \min\left\{\sqrt{\|x-c\|^2 - r^2} + r\zeta_i(x) + \sqrt{\|c - x_f\|^2 - r^2}\right\} \quad (3)$$

To solve an avoidance problem with multiple circular obstacles, Sunder and Siller [4] proposed the *pseudo-return* function which helps to solve the multiple obstacle by solving one obstacle at one time. The algorithm with *pseudo-return* function can be summarized as follows:

*Step 1*: Determine the nearest obstacle for those with shadows containing original point (refer fig. 2 with x: original point, xf: goal point). If k=0, move to *step 3*.

*Step 2*: Follow the negative gradient of the *pseudo-return* function until reaching one of tangency point $T_k^{(j)}$, j=1, 2. Go to step 1. In case when the path intersect other obstacle, we can treat by define intermediate goal as one of the tangent points, then solve the sub-path using this intermediate goal to the most recently incremental point.

*Step 3*: Follow the negative gradient of the unconstrained return function (x doesn't belong to obstacles shadow), ||x-xf||, until reaching the goal. Stop.

*** *The pseudo-return function***

The nearest obstacle, that to be avoided at a given point, can be selected from the set J, defined as:

$$J = \left\{j : \|x - c_j\| = \min_{\{i: x \in S_i\}} \{\|x - c_i\|\}\right\}$$

Consider the nearest circular obstacle with radius $c_k \in R^2$, x denote current point, the *pseudo-return* function is defined as follows:

$$\omega(x,k) = \begin{cases} d_1^{(k)} + d_2^{(k)} + r_k \min[\varsigma_1^{(k)}, \varsigma_2^{(k)}] & \text{if } k \neq 0 \\ \|x - x_f\| & \text{if } k = 0 \end{cases}$$

where

$$d_1^{(k)} = \sqrt{\|x - c_k\|^2 - r_k^2},$$
$$d_2^{(k)} = \sqrt{\|c_k - x_f\|^2 - r_k^2} \quad (5)$$

the k is the index of the nearest obstacle that is selected from the se J, k=0 if J is empty.

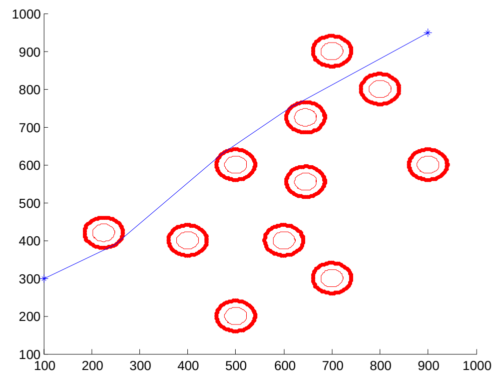



Fig. 2 Path generated by HJB *pseudoreturn* function.

This method permits to generate the path with high speed that almost doesn't depend on the number of obstacles as it solves the obstacles one by one using *pseudo-return* function.

### 2.2 Polynomial Path Planning Approach

The goal of this method [5] is to define a collection of polynomials that connect the start to the goal point, then, generate the shortest collision-free path.

Let E[xi, yi], $1 \leq i \leq X, 1 \leq j \leq Y$, i and j are integers, define a model of two-dimensional space. Let M[xi,yj] describe an occupancy map of space E, where each M[xi,yj] models a rectangle of area $\frac{1}{X} x \frac{1}{Y}$ units in size. In this occupancy map, M[xi,yj]=0 signifies a rectangle of free space while $M[x_i, y_j] \neq 0$ (1/XY) signifies a rectangle of (at least partially) occupied space. We can summarize the occupancy function M as follows:

$$M\{E(x_i, y_j)\} = \begin{cases} 0 & \text{if } (x_i, y_j) \text{ is free space of E} \\ \frac{1}{XY} & \text{if } (x_i, y_j) \text{ is occupied by obstacle} \end{cases} \quad (6)$$

Let S=(xs, ys) define a starting position, G=(xg,yg) define a goal position, 2b be the width of mobile robot. A path from S to G is defined as:

$$P = \{(p_{x_1}, p_{y_1}), (p_{x_2}, p_{y_2}), \ldots, (p_{x_N}, p_{y_N})\}, \quad (7)$$
$$S = (p_{x_1}, p_{y_1}), G = (p_{x_N}, p_{y_N})$$

such that

$$
\begin{aligned}
&-1 \le (p_{x_i} - p_{x_{i-1}}) \le 1 \text{and} \\
&-1 \le (p_{y_i} - p_{y_{i-1}}) \le 1 \forall i = 2 \ldots N, \\
&(p_{x_i}, p_{y_i}) \ne (p_{x_j}, p_{y_j}) \forall i, j = 1 \ldots N, i \ne j
\end{aligned} \qquad (8)
$$

Definition (8) describes any non-self-intersecting path that connects S to G.

A reasonable path should be smooth, implying that it should well-modeled by a continuously differentiable curve F(x(t),y(t)) where F(x(0),y(0))=S and F(x(1),y(1))=G. For such a path, we may define a function

$$
\begin{aligned}
&F^{\perp}(x(t,u), y(t,u)).F'(x(t), y(t)) = 0, \\
&\left\| F^{\perp}(x(t,0), y(t,0)) - F^{\perp}(x(t,\tfrac{1}{2}), y(t,\tfrac{1}{2})) \right\| = b, \\
&\left\| F^{\perp}(x(t,\tfrac{1}{2}), y(t,\tfrac{1}{2})) - F^{\perp}(x(t,1), y(t,1)) \right\| = b, \\
&F^{\perp}(x(t,\tfrac{1}{2}), y(t,\tfrac{1}{2})) = F(x(t), y(t))
\end{aligned} \qquad (9)
$$

Equation (9) models a line of width 2b, centered on and perpendicular to F(x,y), along the length of F. A path P is valid (traversable by the robot) if:

$$
\int_{t=0}^{1} \int_{u=0}^{1} M[F^{\perp}(x(t,u), y(t,u))] = 0 \qquad (10)
$$

Equation (10) tests a one-dimensional line of free-space at each point. Then, we can define a set of Nl second-order polynomial paths from S to G as follows:

$$
\begin{aligned}
&F_l(x(0), y(0)) = S,\ F_l(x(1), y(1)) = G \\
&F_l(x(\tfrac{1}{2}), y(\tfrac{1}{2})) = S + \tfrac{\|S-G\|}{2}(\cos l\theta, \sin l\theta), \\
&\qquad l = -\tfrac{N_l}{2} \ldots \tfrac{N_l}{2}
\end{aligned} \qquad (11)
$$

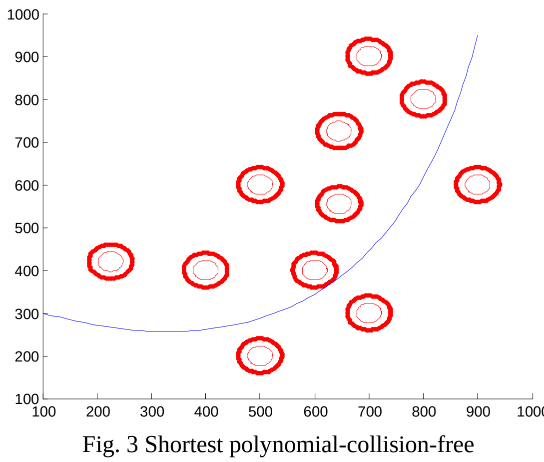



Fig. 3 Shortest polynomial-collision-free

Each polynomial is defined by three points: S, G, and a set of points equi-angular from S at a radius of $\frac{\|S-G\|}{2}$. The values for the number of polynomial Nl and the equi-angular step $\theta$ control the coverage of the set. Figure 4, 5, 6 shows an example set of polynomial paths for Nl=9 and $\theta$=60 for point S and G at a distance of 850mm, and the shortest path in the set is shown.

The optimal path is decided by choosing the shortest path among collision-free paths. The return function for our time-optimal polynomial can be described as follows:

$$
\begin{aligned}
w_c(N_l, \theta) = \{&\min\{\text{length}(F_l(S, G, N_l, \theta))\} | \\
&F_l : (7), (8), (9), (10), (11)\}
\end{aligned} \qquad (12)
$$

$$
\text{length}(F_l) = \sum_{i=1}^{n} \sqrt{\begin{matrix}(x(t_i) - x(t_{i-1}))^2 + \\ (y(t_i) - y(t_{i-1}))^2\end{matrix}},\ t_0 = 0, t_n = 1 \qquad n \in N \qquad (13)
$$

In (12), the Fl has to satisfies the equation (7), (8), (9), (10), (11) before the path-length comparison is taken. The path-generating rate is affected by the choices of Nl, $\theta$. We can decide these choices based on the density of obstacles or characteristic of obstacle arrangement...in order to optimize the rate. Therefore, the experience of the user would helps much in this case.

## 3. TRAJECTORY TRACKING FOR A REAL MOBILE ROBOT

### 3.1 A Nonholonomic Mobile Robot

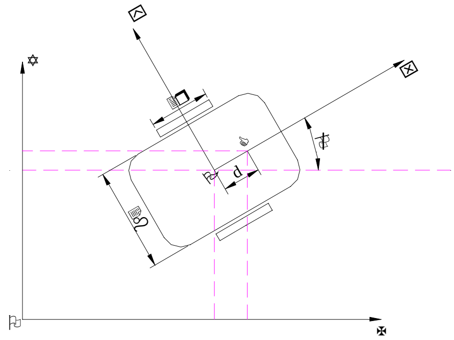



Fig. 4 Mobile robot with two actuated wheels

Our mobile robot is two actuated wheeled type as it was illustrated in fig. 4. The modelling of this nonholonomic system was mentioned in many other literatures such as [10], [11], [12]. It's briefly described here for our convenience. The configuration of the mobile robot can be described by five generalized coordinates

$$
q = [x, y, \phi, \theta_r, \theta_l]^T \qquad (14)
$$

where (x,y) are the coordinates of C, $\phi$ is the heading angle of the mobile robot, and $\theta_r, \theta_l$ are the angles of the right and left driving wheels. Formally, the kinematics of the system can be written as follows

$$
\dot{q}_1 = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -d\sin\phi \\ \sin\phi & d\cos\phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = S_1(q_1)\nu \qquad (15)
$$

$$
A_1(q_1)\dot{q}_1 = 0 \qquad (16)
$$

$$
A_1(q_1)S_1(q_1) = 0 \qquad (17)
$$

where

$$
\nu = \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad (18)
$$

$$
A_1(q_1) = [-\sin\phi \quad \cos\phi \quad -d] \qquad (19)
$$

Certainly, equation (16) results in

$$-\dot{x}\sin\phi + \dot{y}\cos\phi - d\dot{\phi} = 0 \qquad (20)$$

which is a nonholonomic constraint stating that the vehicle can not move in direction transversal to the axis of symmetry of the vehicle. Furthermore, dynamics of the system can be summarized as follows

$$\overline{M}(q_1)\dot{\nu} + \overline{V}_m(q_1,\dot{q}_1)\nu + \overline{F}(\nu) + \overline{\tau}_d = \overline{B}(q_1)\tau \qquad (21)$$

where

$$\overline{V}_m = \begin{bmatrix} 0 & -2m_w d\dot{\phi} \\ 2m_w d\dot{\phi} & 0 \end{bmatrix}$$

$$\overline{M} = \begin{bmatrix} m + 2\frac{I_w}{r^2} & 0 \\ 0 & I_1 \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{b}{r} & -\frac{b}{r} \end{bmatrix}$$

$$I_1 = I_c + 2I_w\frac{b^2}{r^2} + 2m_w b^2 + 2I_m$$

$$m = m_c + 2m_w$$

where $m_w$ denotes mass of the wheel, $m_c$ denotes mass of the vehicle without the wheels, $I_m$ is the moment of inertia of the wheel about a diameter, $I_w$ is the moment of inertia of each driving wheel and the motor rotor about the wheel axis, and $I_c$ is a moment of inertia of the vehicle without the driving wheels and motor rotors about vertical axis passing through point P.

$\tau$ is a set of two moments acting at the wheels, namely $[\tau_r \quad \tau_l]^T$, $\overline{\tau}_d = S^T\tau_d$ represents bounded disturbances including unmodelled dynamics, $\overline{F} = S^T F$ represents friction vector into dynamics.

### 3.2 Trajectory tracking control design

The complete dynamics (15), (21) that consist of the kinematic steering system (15) plus some extra dynamics (21). Let u be an auxiliary input, then by applying the nonlinear feedback [10].

$$\tau = \overline{B}^{-1}(q_1)[\overline{M}(q_1)u + \overline{V}_m(q_1,\dot{q}_1)\nu + \overline{F}(\nu) + \overline{\tau}_d] \qquad (22)$$

One can convert the dynamic control problem into the kinematic control problem

$$\dot{q}_1 = S_1(q_1)\nu \qquad (23.a)$$

$$\dot{\nu} = u \qquad (23.b)$$

From the path planning, we can construct a reference model for the vehicle to follow, these are

$$q_r = [x_r \quad y_r \quad \phi_r]^T, \quad \nu_r = [v_r \quad \omega_r]^T, \quad \dot{x}_r = v_r\cos\phi_r$$

$$\dot{y}_r = v_r\sin\phi_r, \quad \dot{\phi}_r = \omega_r \qquad (24)$$

The tracking error vector is define as follow

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \phi_r - \phi \end{bmatrix} \qquad (25)$$

and the derivative of the error is

$$\dot{e} = \begin{bmatrix} \omega e_2 - v + v_r\cos e_3 \\ -\omega e_1 + v_r\sin e_3 \\ \omega_r - \omega \end{bmatrix} \qquad (20) \quad (26)$$

The auxiliary velocity control input that achieves tracking for 23.a is given by

$$\nu_c = \begin{bmatrix} v_r\cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix} \qquad (21) \quad (27)$$

The derivative of $\nu_c$ becomes

$$\dot{\nu}_c = \begin{bmatrix} \dot{v}_r\cos e_3 \\ \dot{\omega}_r + k_2\dot{v}_r e_2 + k_3\dot{v}_r\sin e_3 \end{bmatrix} + \begin{bmatrix} k_1 & 0 & -v_r\sin e_3 \\ 0 & k_2 v_r & k_3 v_r\cos e_3 \end{bmatrix}\dot{e} \qquad (28)$$

Then the nonlinear feedback acceleration control input is

$$u = \dot{\nu}_c + k_4 I(\nu_c - \nu) \qquad (29)$$

## 4. SIMULATION RESULTS

Our simulation was executed with an mobile robot that is assumed to have the following parameters

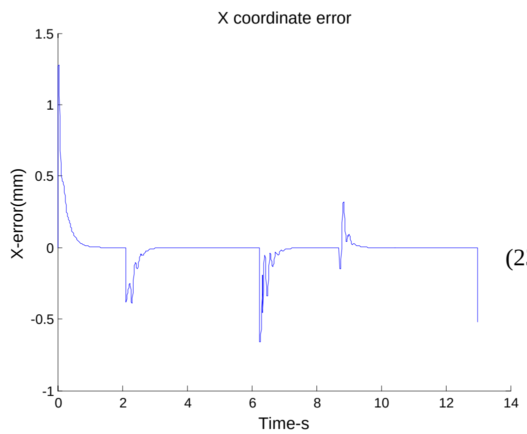
$b = 31.5\,mm$, $a = 63\,mm$, $d = 0.3\,mm$, $r = 21.5\,mm$

### 4.1 Tracking to a path generated by HJB approach

The design parameters were obtained through simulation:

$k_1 = 50$, $k_2 = 0.2$, $k_3 = 0.35$, $k_4 = 5$.

The designed straight reference velocity is assumed to be constant at value of 80mm/s, the angular velocity then be given through it's constraints (24) with coordinate data of the planned path, straight reference velocity with respect to time at sampling rate of 10msec.




(23.a)
(23.b)

Fig. 4 Error on x direction of local coordinates

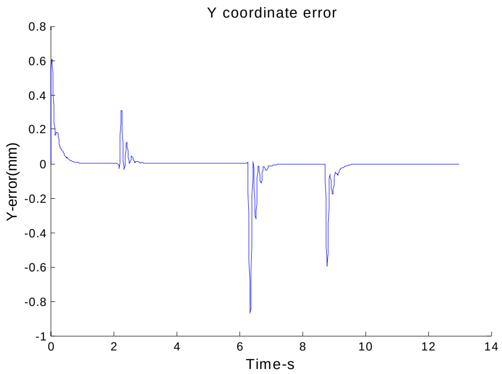



Fig. 5 Error on y direction of local coordinates

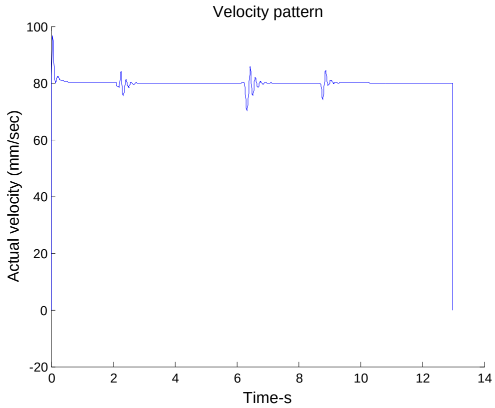



Fig. 6 Traversal velocity acquired by torque control inputs

### 4.2 Tracking to a path generated by Polynomial approach

The design parameters were obtained through simulation:

$k_1 = 105$, $k_2 = 0.09$, $k_3 = 0.25$, $k_4 = 7$.

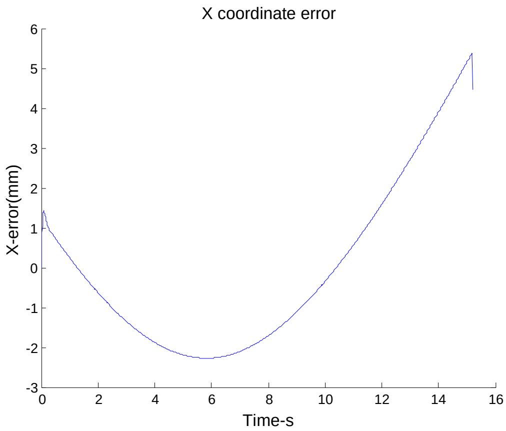



Fig. 7 Error on x direction of local coordinates

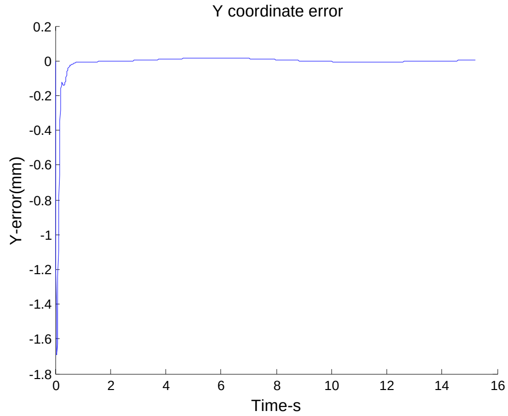



Fig. 8 Error on y direction of local coordinates

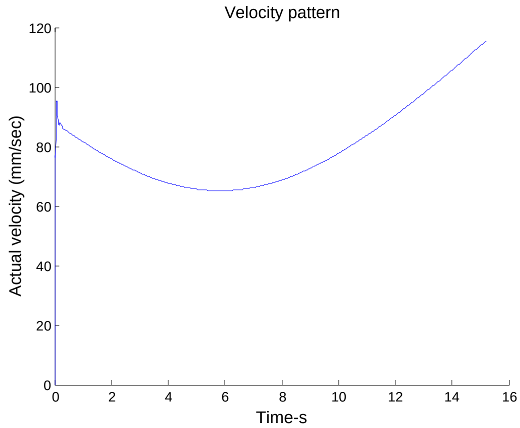



Fig. 9 Traversal velocity acquired by torque control inputs

## 5. CONCLUSION

Simulation results show us that the HJB based approach is still the optimal in both path length and traversal time although the tracking performance deteriorates along vehicle's passage about the obstacles. However, the error fluctuations are not as that bad and can be considered as acceptable error in practical meaning.

The simulation results of Polynomial approach appears that it failed to track the trajectory in the traversal direction (Fig. 7), otherwise it sucessfully follows the path since the lateral error simultaneously converges to 0 (Fig. 8) in a very short time.

One more problem in the later path planning method is that the trend of abnomal raising of the velocity (Fig. 9) control input. In this meaning, it's possible to have chance of insufficient torque inputs supplied by the actuators or defficiency in term of energy usage. The drawbacks lay much on the control law while the mobile robot has to deal with the curved trajectories

Therefore, the polynomial path planning approach can be considered as an alternative optimal method and it's necessary to improve it's performance through further researchs.

## REFERENCES


[1] C. I. Connoly, J. B. Burns, and R. Weiss, “Path planning using Laplaces equation”, *in Proc. IEEE Int. Conf. Robot. Automat.*, Vol. 1, pp. 2102-2106, 1991.

[2] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions”, *IEEE Trans. Robot. Automat*., vol. 8, pp. 501-518, 1992.

[3] C. Alexopolous and P. M. Griffin, “Path planning for a mobile robot”, *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 318-322,1992.

[4] S. Sundar, Z. Shiller, "Optimal Obstacle Avoidance Based on the Hamilton-Jacobi-Bellman Equation", *IEEE Trans. Robot. Automat.*, vol 13, No.2, pp. 305-310, 1997.

[5] Adam Hoover, Bent David Olsen, “Path Planning for Mobile Robots Using a Video Camera Network”, *in Proc. IEEE/ASME Int. Conf. Adv. Intell. Mecha.*, Atlanta, USA, 1999, pp. 890-895.

[6] A. I. Moskalenko, "Bellman equations for optimal processes with constraints on the phase coordinates", *Autom. Remote. Cont., A Translation Avtomatika I Telemekhanika*, vol. 4, pp. 1853-1864, 1967.

[7] B. Herman, R. Dominiek, “Path Planning for Mobile and Hyper-Redundant Robots using Pythagorean Hodograph Curves”, *ICAR/IEEE, Monterey, CA*, July 7-9, 1997.

[8] R. T. Farouki and C. A. Neff. Hermite, “Interpolation by Pythagorean hodograph quintics”. *Mathematics of Computation*, 64(212): 1589-1609, 1995.

[9] Kim D. K., Kim H. K., Kim S. B. (2002), "Image Processed Tracking System of Multiple Moving Objects based on Kalman Filter", *KSME International Journal*. Vol. 16, No. 4, pp. 427-435.

[10] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics", *Proc. 34th Conf. Deci. Ctrl.*, New Orleans, LA, Dec. 1995, p.p 3805-3810.

[11] Krzysztof Kozlowski, Jaroslaw Majchrzak, "A backstepping approach to control a nonholonomic mobile robot", *Proc. 2002 IEEE Int. Conf. on Robo. & Auto.*, Washington DC., May 2002, p.p 3972-3977.

[12] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive Tracking Control of a Nonholonomic Mobile Robot", *IEEE Trans. on Robot. and Auto.*, Vol. 16, No.5, October 2000.