International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 19 (2015) pp 40672-40678

Time Optimal Path Planning while Considering Physical and Geometrical Limits during Obstacle Avoidance

Raimarius Delgado

Graduate Student, Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul, South Korea raim223@seoultech.ac.kr

Byoung Wook Choi

Professor, Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul, South Korea Corresponding Author bwchoi@seoultech.ac.kr

Abstract

In this paper, an approach for a wheeled mobile robot to deal with the geometrical limits of a given path while also satisfying its physical constraints is presented. Presence of obstacles along a path is inevitable in practical mobile robot navigation. To avoid incoming collision, the robot is redirected to a new path where high-curvature turning points are addressed to be imminent. A known static obstacle is placed within a pre-determined Bézier curve-based path for the robot to exhibit obstacle avoidance accordingly. A trajectory generated through the robot's acceleration limits according to the path's curvature, expressed lengthy travel time, non-periodic time sampling, and does not consider velocity limits configured on the robot. To deal with prevailing issues, a convolution-based approach was implemented but it demonstrated inability to track the path of high curvature in uniform sampling time. Therefore, a variation of linear interpolation was proposed to generate an actual trajectory in tracking the redirected path. The proposed time optimal path planning is able to consider the robot's physical limits, cope with the path's geometrical constrictions and conserve uniform time sampling to be conducted in real-time control.

Keywords: High Curvature Path, Physical Limits, Convolution, Bézier Curve, Path Planning, Obstacle Avoidance.

Introduction

Evolutionary studies of mobile robot systems made it possible for a robot unit to display precise movement and controllability through uncomplicated user-generated velocity commands. These days, application of mobile robots has expanded not only in the industrial and commercial sectors, but also in residential areas [1]. Another great result from this development is that places inaccessible to human beings are now made reachable by proper handling and operation of mobile robot systems, such as space missions, hazardous environments, and military operations [2].

Path planning is considered to be one of the most important tasks for any robotic system. Most path planning techniques are executed by calculating a curved path for a mobile robot to follow [3]. There are two fundamental types of path planning: dynamic/adaptive path planning, on which the robot is expected to produce a new path in response to the environment it is moving on [4-5], and static path planning on which a known path can be generated beforehand [6].

A robot motion could be planned to be either constrained or not, depending on the physical limitations of the mobile robot. Typically, robots are bounded with its maximum velocity [7], maximum acceleration [8-9], maximum jerk [10]. These limits should be dealt with in order for the robot to travel along the predetermined path with higher accuracy while reducing the chance of physical damage.

While driving along a path, it is more likely for a robot to encounter obstacles, especially in many service applications where mobile robots need to share their work region with different obstacles [11]. Different researchers have studied various techniques to avoid collisions [12-15]. But the typical and mostly used algorithms can be simply classified into two—deviating the original path after detecting a known obstacle along the way, or the more sophisticated algorithm on which the robot would detect not only the existence of an obstacle but also measure its dimensions before making a specific counteraction.

A Bézier curve-based obstacle avoidance technique was suggested in [8]. If a possible collision is detected while the robot is travelling on the planned path, the robot is redirected to a different angle to generate two consecutive Bézier curves with respect to the distance of the obstruction and a safety factor. However, the produced redirections also show turning points with high curvature values. These geometric constrictions lead to known accuracy, efficiency, and reliability issues [16-18]. After implementation of the velocity generation algorithm through the relationship of the path's curvature and the robot's acceleration limits, the sampling time of the generated trajectory was observed to be non-periodic that causes difficulty in real-time control. Moreover, it does not consider the velocity limit of the mobile robot.

To deal with the issue regarding the physical limits of the mobile robot, a convolution based trajectory generation was used in [10]. However, it only considers the straight path between two points and not the rotational heading angles of the Bézier curve. Hence, a heading angle considered path planning based on convolution was introduced in [7] to compromise with this shortcoming. The velocity profile produced through the modification shows high efficiency in fulfilling the required velocity constraints.

In [8], high curvature turning points were present when redirecting the path to avoid obstacles. In this case, we can apply the heading angle considered path planning suggested in [7] to apply velocity constraints but the generated trajectory could not follow the redirected path in uniform sampling time due to high curvature that is a geometrical constraint of the path.

Therefore, a modified convolution-based path planning method is required to consider both geometrical constraints and physical constraints in uniform sampling time. In order to produce a trajectory that follows the high curvature reference in equal sampling time, linear interpolation was applied to the trajectory generated by the heading angle considered path planning.

This approach was corroborated through mathematical simulations. Using this method, a mobile robot could avoid obstacles in a static environment while considering its physical limits and the geometrical limits of the path with periodic sampling time. The periodicity of sampling time is important in real-time control of mobile robot actuation.

Section 2 is divided into different parts that discuss the different studies related to this research. A path planning technique based on a Bezier curve with a corresponding velocity generation technique considering the acceleration limits of the robot. Obstacle avoidance was discussed afterwards. Due to occurrence of drawbacks regarding physical limitations of the mobile robot, a convolution operator was examined. The proposed algorithm and its application through numerical simulations are found in Section 3. Conclusion and discussion of future studies are included in the final section.

Related Researches and Discussion

A. Bézier Curve-based Path Planning Considering Acceleration Limits and Obstacle Avoidance Technique

i. Velocity Profile Generation

Jolly et al. in [8] defined a predetermined path with initial point, P, terminal point, S, and control points at Q and R that is based on a Bézier curve as shown in Fig. 1. The maximum velocity along the planned Bézier curve-based path is bounded by both the tangential and radial acceleration limits of a mobile robot. If the acceleration of the robot along the curve is greater than the limit, continuous slipping or skidding of the robot during its drive occurs. As a result, the actual initial and terminal velocity of the mobile robot differs depending on the relationship between the curvature of the given path and the robot's radial acceleration.

In other words, the velocity at each point along the curve is possible to be computed through the curvature of the Bezier curve and the configured acceleration, both radial and tangential, limits of the mobile robot. The velocity profile is evaluated by uniformly accelerating the velocity by the tangential acceleration from the initial point to the terminal point, and vice-versa. This would result to two velocity profiles in the forward and backward direction.

To consider the path's curvature, the velocity values of the turning points, i.e., the points along the path with the largest curvature value, were calculated using the maximum radial acceleration. If these turning points are existent on the path, the technique was altered to start from the turning point, in the forward direction to the terminal point, then backward direction to the initial point. The combination of these velocity profiles would then produce the maximum allowable velocity, which is the lowest velocity value at a certain point. If the radial acceleration of the resulting maximum allowable velocity does not exceed the limit, it is said to be realizable in practical application.

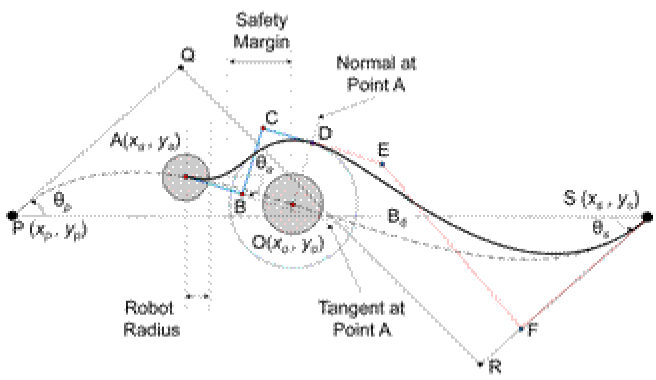

Fig. 1. Redirected Path to Avoid a Static Obstacle

ii. Obstacle Avoidance Technique

In practical application of mobile robot systems, it is inevitable for robots to encounter obstacles along its way. Before probable contact, the robot does not stick to its initially planned path. Instead, the robot is redirected to a different path to avoid colliding with the obstacle. Also show in Fig. 1, the robot represented by the circle with the center at point A, is said to collide with the obstacle at point O, when the radii of both the circles overlaps each other, in other words, the center of the robot would cross the safety margin.

Therefore, at point A, the robot is redirected to a different direction to avoid colliding with the obstacle by generating another Bézier curve described by the polygon ABCD. The first polygon side AB is taken as the ratio of the tangent line that passes through point A. In proportion, side CD is a segment parallel to the tangent line at A that passes through point D. Point D is then evaluated by using the slope θ of A with respect to the initial path and the normal line to the tangent at Point A which also passes through the center of the obstacle.

$$\begin{aligned} x_d &= x_o - r_o \cos\theta_a \\ y_d &= y_o - r_o \cos\theta_a \end{aligned} \quad (1)$$

Assuming the side AB is d11, side CD that corresponds to the latter side of the Bézier polygon, d12, is made equal to the computed value of the former. Thus, the points B and C are computed using these values with the following trigonometric analysis:

$$\begin{aligned} x_b &= x_a + d_{11}\cos\theta_a \\ y_b &= y_a + d_{11}\sin\theta_a \\ x_c &= x_d - d_{12}\cos\theta_a \\ y_c &= y_d - d_{12}\sin\theta_a \end{aligned} \tag{2}$$

Another Bézier curve is generated for the mobile robot to get back on track and reach the targeted terminal point. The second deviation is represented by the polygon DEFS. Point E is assumed to be lying on the same segment with the points B and C while point F lies on the same line with the second control point that defines the original Bézier curve-based path, point R.

Considering that all polygon sides of both the redirected curves are equal, the third control points are evaluated by:

$$\begin{aligned} x_e &= x_a + d_{21}\cos\theta_a \\ y_e &= y_a + d_{21}\sin\theta_a \\ x_f &= x_r - d_{22}\cos\theta_a \\ y_f &= y_r - d_{22}\sin\theta_a \end{aligned} \tag{3}$$

iii. Discussion

This paper focuses on the high-curvature that occurs when the original path is redirected to avoid an obstacle using the algorithm mentioned above. In order to verify the validity of the approach, the same path and corresponding velocity profile from [8] is shown in Figs. 2 and 3, respectively.

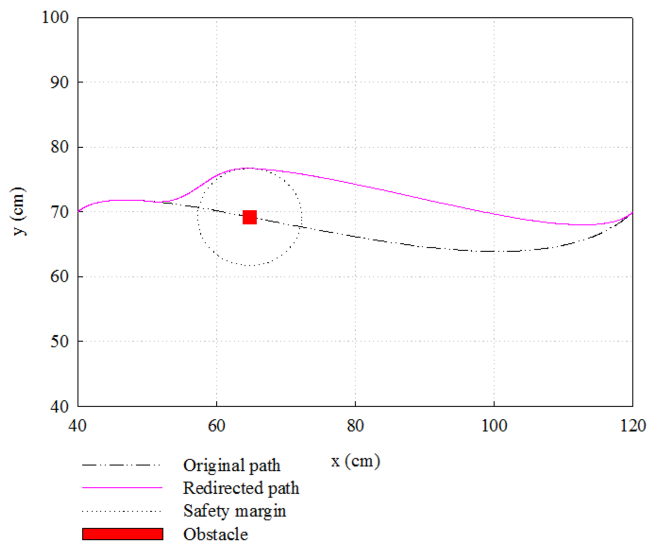

Fig. 2. Path Planning to Avoid a Known Obstacle

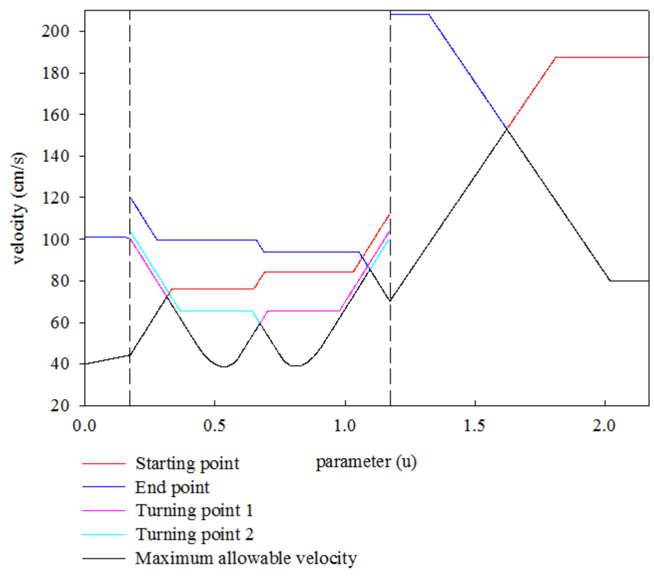

Fig. 3. Velocity Profile Considering Acceleration Limits

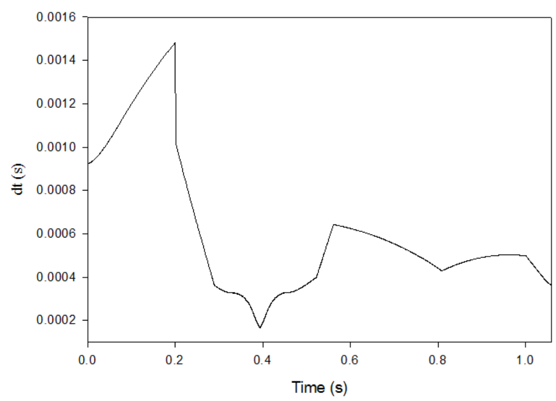

Fig. 4. Time Difference between each Point of the Velocity Considering Acceleration Limits

set *period and* ***make*** *periodic task*
Procedure *Servo_Task*
  while*(1)*
    read start time;
    grab the mutex lock;
    get velocity from trajectory;
    put velocity to actuator
    release the mutex;
    read end time;
    wait period;
  end
end

Fig. 5. Pseudo Code for a Sample Real-time Task, *Servo_Task*

In Fig. 2, the original path prior to the obstacle was a smooth Bezier curve however, on the first portion of the redirection; an S-curve with high curvature was produced as can be seen on the turning points of the velocity profile in Fig. 3. The second division of the graph expresses velocity values taken from the initial, terminal, and two turning points generated with respect to the curvature of the curve.

The generated velocity profile was able to generate a trajectory that can follow the planned path with high curvature. However, it was shows non-uniform sampling time as shown in Fig. 4. Non-periodic velocity commands are considered difficult to be realized in practical situations, for example, cyclic tasks in real time operating systems, such as Xenomai [19-20].

Xenomai is a real-time development framework that provides a pervasive, interface-agnostic, hard real-time support to user-space applications [21]. As shown by the pseudo code in Fig. 5, given tasks in real time is iterated inside a loop within a given period. An important feature of real-time systems is that tasks responses should be kept in strict time samples or constants. Hence, controlling the robot in real-time or using real-time operating systems requires the velocity commands to be in uniform sampling time.

If a system could not handle these periodic commands then the output is not expected to show proper performance. Moreover, the velocity limit of the robot was also not considered during this path planning method that is another factor when dealing with real-life navigation of robot systems. To deal with these drawbacks, the proposed method uses a convolution operator with the results shown in the next section.

B. Convolution-based Trajectory and Bezier Curve Heading angle Consideration

A convolution operator suggested by Lee et al. in [10] was utilized in order to deal with the physical limits neglected in the previous results. In order to create the velocity profile based on convolution, other physical limits are identified such as the maximum velocity and maximum jerk, according to the specification of the mobile robot. However, the convolution-based trajectory generation only computes the distance of the straight line between two points. It does not consider the heading angles at each point of the curve, in other words, the curvature of the Bézier curve.

Thus, a heading angle considered path planning suggested by Gil Jin Yang et al in [7] that transform the parameter of the curve by integrating the convolution-based trajectory of a straight-line distance to the actual travel distance along the Bézier curve was used to consider the position of the robot while travelling along a predetermined path. Furthermore, practical computations are made easier and more realizable by transforming the original parameter, to the time domain since motion of any bodies, such as mobile robots, are expressed with respect to time.

Simulation Results

The same path in Fig. 2 was used in order to formulate the proposed method. A cubic Bézier curve starting at P (40, 70) with 45º heading angle and velocity of 40 cm/s and ending point coordinates at S (40, 70), 35º, and 80 cm/s was generated as the planned path. The Bézier polygon sides, *d*1 and *d*2, that control the curve was computed to be 12.28 cm and 24.56 cm respectively. The obstacle with a maximum radius of 7.5 cm was placed along the curve and was computed to have the coordinates of O (64.72, 69.20). The computed position of point A, where the avoidance started, was A (49.74, 71.69) from a safety factor and robot radius of 7.5 cm. The redirected Bézier curves are then generated using these initial values.

Fig. 6 shows the velocity profile generated using the heading angle considered path planning in [7] for a velocity limit of 120 cm/s. Using these velocity values, the actual trajectory was generated through the computation of the radian angle at each point by:

$$\theta(u) = \tan^{-1}\frac{dy(u)}{dx(u)} \tag{4}$$

Using (4), the velocity in each point at both the *x* and *y* axes were calculated using simple robot kinematics expressed as:

$$\begin{aligned} v_x(u) &= v(u)\cdot\cos\theta(u) \\ v_y(u) &= v(u)\cdot\sin\theta(u) \end{aligned} \tag{5}$$

In this equation, *v*(*u*) is denoted as the velocity profile to be used in generating the numerical interpretation of the actual trajectory from the initial position expressed as:

$$\begin{aligned} x(u) &= x_i + \sum_{u=0}^{n} v_x(u)dt \\ y(u) &= y_i + \sum_{u=0}^{n} v_y(u)dt \end{aligned} \tag{6}$$

where *n* is the total number of sample points and *dt* is the time that the robot would take to move from one point to another along the curve.

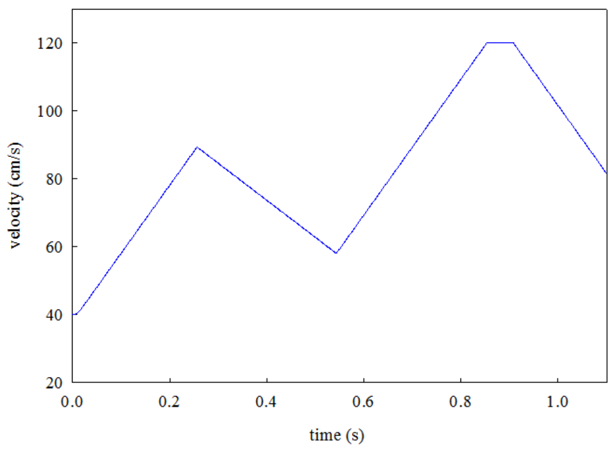

Fig. 6. Convolution-based Velocity Profile Considering the Heading Angles of a Bézier Curve

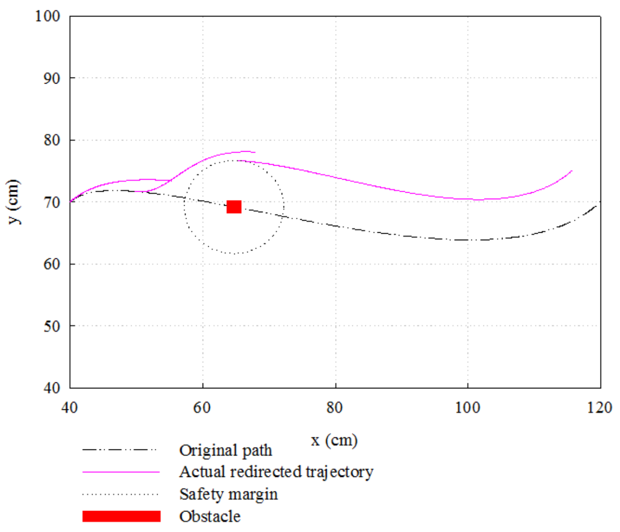

Fig. 7. Trajectory using the Convolution-based Velocity Considering the Heading Angles of a Bézier Curve

Fig. 7 shows the trajectory generated using equations (4-6) with the generated velocity from the heading angle considered path planning in Fig. 6. Inability of the generated trajectory to follow the original path is conspicuous within the figure due to the high curvature turning points.
Since the heading angle considered velocity was generated in uniform sampling time, the non-uniform time difference, *dt*, was computed using the displacement of each point in the *u*-domain, *dS*, and corresponding velocity, *v*, expressed as:

$$dt = \frac{dS(u)}{v(u)} \tag{7}$$

The computed non-uniform time difference is shown in Fig. 8. Using this result, another trajectory was generated and as expected, the result was able to follow the desired path.
Since the trajectory was non-periodic, the proposed modified convolution-based path planning velocity profile was generated by applying the following linear interpolation variation:

$$v_{uni}(t) = v(t_{n-1}) + \frac{v(t_n) - v(t_{n-1})}{t_n - t_{n-1}}(t - t_{n-1}) \tag{8}$$

where the total time from integrating the result from equation (7) is denoted by t_n and the uniformly spaced t is defined as $t_{n-1} \leq t \leq t_n$.
Application of the suggested method is shown in Fig. 9 with its comparison with the heading angle considered path-planning approach. The corresponding generated trajectory shown in Fig. 10, shows that the proposed convolution-based path planning method was able to follow the original path with high curvature. Moreover, the total travel time calculated was 0.97 s, which is shorter than the computed travel time from the two previous methods that exceeded 1 s that makes the proposed method relatively time-optimal.

Conclusion

In this paper, a path planning method was proposed to allow a mobile robot to travel along a cubic Bézier curve with high curvature turning points during obstacle avoidance. Physical constraints configured within the mobile robot were also satisfied while demonstrating uniform sampling time and time optimality.
A planned path based on a Bézier curve with the corresponding obstacle avoidance algorithm and velocity profile generation was simulated as shown in [8].
Although the previous research shows commending results, it also exhibited longer travelling time, non-uniform time difference at each sampling point intervals, and inability to consider other physical constraints, such as velocity limit, of the mobile robot.
The same path was then simulated using a heading angle considered path-planning method suggested in [7]. This approach was able to produce a velocity profile that can deal with the drawbacks of the preceding study however; it also demonstrated incapability to follow the original Bézier curve-based path in periodic time sampling.
For the general problem of generating a trajectory for the mobile robot that can be redirected when approaching a known obstacle while dealing with all possible physical limits, an approximation method by applying linear interpolation to the convolution-based velocity to generate velocity profiles that compromises all the drawbacks of the previous methods and time optimality was presented.

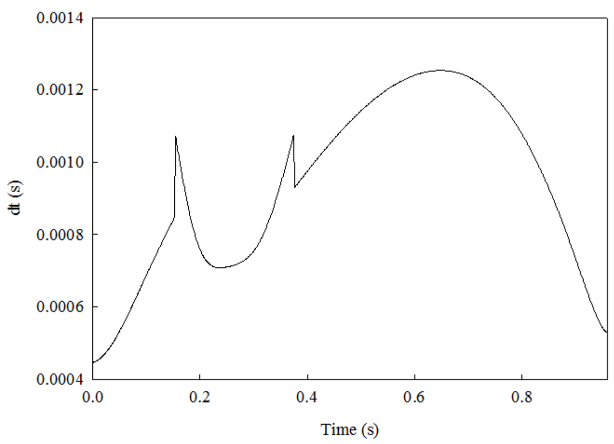

Fig. 8. Time Difference between Each Point of the Convolution-based Velocity Considering Heading Angles of a Bézier Curve

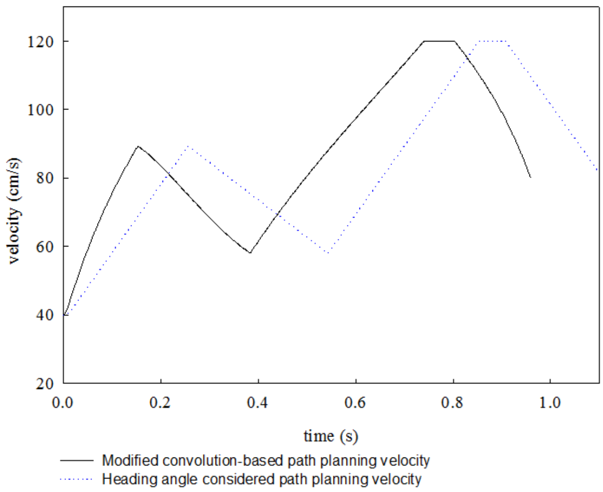

Fig. 9. Comparison between the Proposed and the Velocity Considering Heading Angles of a Bézier Curve

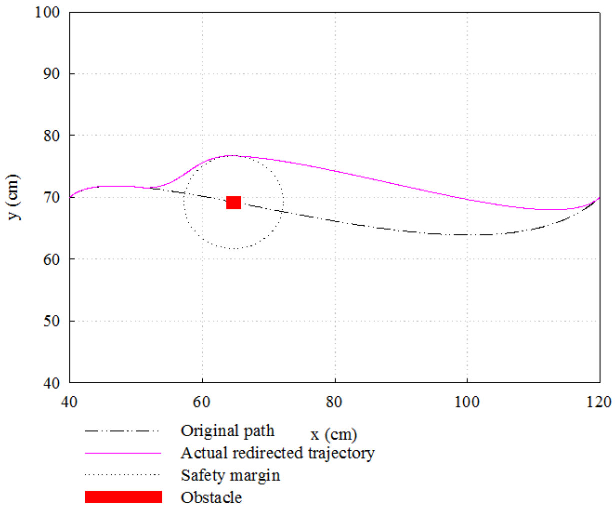

Fig. 10. Trajectory using the Proposed Velocity Profile

Through implementation of the proposed method, the generated trajectory was able to travel along a planned Bézier curve, redirect and generate a new path near a predetermined obstruction, cope with any physical limits while demonstrating the shortest travel time and periodicity.

However, these results also disclose that further studies regarding operation of this method in joint-space is needed. In order to actuate a wheeled mobile robot, the velocity commands in both its left and right wheels should be considered while also satisfying, the physical limits of the unit. Although, the central velocity of the mobile robot was able to be within the limits, if the joint space velocity does not contemplate eligible values, the robot could not reach the targeted terminal position.

Furthermore, the proposed trajectory that possesses uniform sampling time could be implemented in real-time operating systems where cyclic tasks require uniform time sampling

Acknowledgment

This study was financially supported by Seoul National University of Science and Technology.

References

1. S. Schiffer, A. Ferrein, and G. Lakemeyer, "Caesar: an intelligent domestic service robot," Journal of Intelligent Service Robotics, vol. 5, no. 4, pp. 259-273, 2012.
2. K. Sugihara, and I. Suzuki, "Distributed motion coordination of multiple mobile robots," Proc. 5th IEEE Int'l Symp. Intelligent Control (ISIC), 1990.
3. Y.C. Chang, and Y. Yamamoto, "Path planning of wheeled mobile robot with simultaneous free space locating capability," Journal of Intelligent Service Robotics, vol. 2, no. 1, pp. 9-22, 2009.
4. C. Laugier, D. Vazquez, M. Yguel, Th. Fraichard, and O. Aycard, "Geometric and Bayesian models for safe navigation in dynamic environments," Journal of Intelligent Service Robotics, Vol. 1, No. 1, pp. 51-72, 2008.
5. Cheng, P.Y., and Chen, P.J., "Navigation of mobile robot by using D++ algorithm," Journal of Intelligent Service Robotics, vol. 5, no. 4, pp. 229-243, 2012.
6. J. Tu, and S.X. Yang, "Genetic algorithm based path planning for a mobile robot," Proc. IEEE Int'l. Conf. Robotics and Automation (ICRA), 2003.
7. G.J. Yang, and B.W. Choi, "Smooth trajectory planning along Bézier curve for mobile robots with velocity constraints," International Journal of Control and Automation, vol. 6, no. 2, pp. 225-234, 2013.
8. K.G. Jolly, R.S. Kumar, and R. Vijayakumar, "A Bézier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits," Robotics and Automation Systems, vol. 57, no. 1, pp. 23-33, 2009.
9. M. Lepetič, G. Klančar, I. Škrjanc, D. Matko, and B. Potočnik, "Time optimal path planning considering acceleration limits," Robotics and Automation Systems, vol. 45, nos. 3-4, pp. 199-210, 2003.
10. G. Lee, J. Kim, and Y. Choi, "Convolution-based trajectory generation methods using physical system limits," Journal of Dynamic Systems, Measurement, and Control, vol. 135, no. 1, pp. 1-8, 2013.
11. T.C. Liang, J.S. Liu, G.T. Hung, and Y.Z. Chang, "Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral," Robotics and Automation Systems, vol. 52, no. 4, pp. 312-335, 2005.
12. J. Bronstein, and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, no. 5, pp. 1179-1187, 1989.
13. R. Simmons, "The curvature-velocity method for local obstacle avoidance," Proc. IEEE Int'l Conf. Robotics and Automation (ICRA), 1996.

14. J. Borenstein, “Obstacle avoidance with ultrasonic sensors,” IEEE Journal of Robotics and Automation, vol. 4, no. 2, pp. 213-218, 1988.
15. A.Chakravarthy, “Obstacle avoidance in a dynamic environment: a collision cone approach,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 28, no. 5, pp. 562-574, 1998.
16. K. Yang, D. Jung, and S. Sukkarieh, “Continuous curvature path-smoothing algorithm using cubic Bézier spiral curves for non-holonomic robots,” Advanced Robotics, vol. 27, no. 4, pp. 247-258, 2013.
17. M. Tounsi, and J.F. Le Corre, “Trajectory generation for mobile robots,” Robotic Research, vol. 41, nos. 3-4, pp. 367-376, 1996.
18. L. Zeng, and G.M. Bone, “Mobile robot navigation for moving obstacles with unpredictable direction changes, including humans,” Advanced Robotics, vol. 26, no. 16, pp. 1841-1862, 2012.
19. G.J. Yang, and B.W. Choi, “Implementation of joint space trajectory planning for mobile robots with considering velocity constraints on Xenomai,” International Journal of Control and Automation, vol. 7, no. 9, pp.189-200, 2014.
20. M. Mächtel, and H. Rzehak, “Measuring the influence of real-time operating systems on performance and determinism,” Control Engineering Practice, vol. 4, no. 10, pp. 1461-1469, 1996.
21. B.W. Choi, D.G. Shin, J.H. Park, S.Y. Yi, and S. Gerald, “Real-time control architecture using Xenomai for intelligent service robots in USN environments,” Journal of Intelligent Service Robotics, vol. 2, no. 2, pp. 139-151, 2009.