C^2 Piecewise Cubic Bezier Curve Based Smoothing Path for Mobile Robot

Phan Gia Luan and Nguyen Truong Thinh
Department of Mechatronics, HCMC University of Technology and Education, Vietnam
Email: pgluan95@gmail.com, thinhnt@hcmute.edu.vn

***Abstract*—Good path planning result of mobile robot can not only make better navigation system, but also optimize the travelling time of robot. Most path planning method just can extract the waypoints or line segments for guiding robot to desired target. By implementing of smoothing path, robot can easily archive target position without constantly stopping and turning many times. In this works, we construct continuous C^2 path by using Piecewise Cubic Bezier Curve that respect to initial pose and target pose of robot. The results in experiment shown the acceptable performance of our robot in the presence of proposed smoothing path algorithm.**

***Index Terms*—smooth path planning, piecewise cubic bezier curve, mobile robot**

I. Introduction

Mobile robots are increasingly playing an important role in the fields of social life so they are used more and more in different environments. Path planning becomes a backbone of robot' autonomy. There are many different path planning algorithms listed at [1] and [2]. However, most path planning algorithms only look for waypoints or line segments that are connected and towards the target point without any kinematic constraints. As a result, the robot must constantly stop and turn at sharp path segments. In this article, we propose to use a piecewise cubic Bezier curve to create curved trajectories that have continuous curvature over the entire curve based on the given waypoint set $\{\mathbf{P}(1), \mathbf{P}(2), ..., \mathbf{P}(n+1)\}$.

The problem of smoothing path is not so new [3]. The smooth path obtained by using the C^2 Piecewise cubic Bezier Curve is also introduced in [4]. However, they did not mention the pose of a robot in the start position and target position. The genetic algorithm for directly generating trajectories for robots using PCBC was introduced in [5]. However, the curvature of obtained curves in [5] is not continuous. The major disadvantage of the genetic algorithm is slowing down a system while finding an optimal path when the map has a relative level of complexity. Marzoughi proposed an incredible navigation algorithm [6] for his robot team based on the leader-follower pattern. Besides the positive simulated results, there are still some weaknesses unless the unknown environment is presented. Only by using the obstacle avoidance technique, the robot is completely unable to create an optimal path [7]. Another problem is that robots easily get stuck in the cul-de-sac scenario. Stochastic Particle Swarm Optimization for smooth path planning had been introduced in [8]. This technique can optimize both path length and robot kinematic constraints. However, generating piecewise cubic spline path for the large population can significantly slow down the processor. Premature or slow converging problems may present with non-deterministic methods. The main contribution of this work is smoothing paths by using C^2 Piecewise Cubic Bezier Curve. The waypoints obtained from A Heuristic algorithm are fed to our method. The smooth path also is reprocessed by rearrange anchor point method to reduce the oscillation and the magnitude of curvature. Produced curve shape respect to the start pose and target pose of robots in order to easily controlling is also proposed.

II. Methodology

A. Mobile Robot Kinematics

This research is based on a simple 2-wheels differential-drive mobile platform that has two independently controlled wheels and one omnidirectional wheel for balancing. The robot moves on a flat lane, localization by LiDAR and it' poses can be express by the relation between reference frame $\{W\}$ and mobile robot itself frame $\{R\}$. The robot frame $\{R\} = (\mathbf{P}_R(t), \psi_R(t))$ is defined by a position vector of robot respect to reference frame and robot heading respectively. Let $(V_R(t), \omega_R(t))$ be the instantaneous linear speed and angular speed of robot body based on $\{W\}$, $(\dot{\theta}_l(t), \dot{\theta}_r(t))$ be the left and right wheel angular speed, r_w be the radius of the driving wheel, l_w be a distance between two wheels, r_r be the half diameter of robot' circular boundary. Because each command of a processor just can accelerate one wheel independently to make a circular motion, therefore the control equation based on robot motion is given by (1).

Manuscript received May 5, 2020; revised August 20, 2020.

doi: 10.18178/ijmerr.10.9.519-525

$$\begin{bmatrix} \dot{\theta}_r(t) \\ \dot{\theta}_l(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{r_w} & -\frac{l_w}{2r_w} \\ \frac{1}{r_w} & \frac{l_w}{2r_w} \end{bmatrix} \begin{bmatrix} V_R(t) \\ \omega_R(t) \end{bmatrix} \quad (1)$$

B. A Heuristic Algorithm

A Heuristic algorithm (A Star algorithm) is the most popular grid-based heuristic searching technique that widely used in offline path planning. By the given binary map, starting position and target position, A Star algorithm can archive the optimal path in the form of waypoints. A Star is similar to Dijkstra's algorithm but it uses a heuristics approach. It adds the heuristics function to the weight of every candidate solution that can be express as (2).

$$f_n = g_n + h_n \quad (2)$$

where h_n is the distance between current checking point to the target point, g_n is the length of the route between the initial point and current checking point that contains the last checking point, f_n is the total weight of current checking point.

Normally, the robot is treated as the point and the obstacle' size is extended to compensate for the size of the robot [9]. This technique can help path planning algorithms ignore the collision caused by robot size but that increases the grid resolution that causes time-inefficient in searching. For this reason, a grid is generated based on robot position and robot size. However, direct searching in the grid that has cell' size equal to robot' size is not a good idea. Fig. 1 illustrates the presence of a partial unobservable route when the robot moving on the diagonal line with cells' size equal to the robot' size. Several rules have been introduced in [10] to tackle unsafe diagonal movements of robots. For the simpler and faster algorithm, this work introduces the searching window illustrated in Fig. 2 defined by four cells and it moves cell by cell.

In general, A star checks one window each time which is called the current checking window. The result has two states: feasible or not, and this result based on the results of cells in the checking window. A star algorithm stores all of reviewed windows in the set called close-set and all of potential windows in the set called open-set. Firstly, A star takes into consideration eight neighbor windows around the current checking window. If any neighbor window has location is not presented in open-set or close-set or presented in open-set but having lower g score than any presented same location window in open-set, that window will be added to open set. Any windows that have the same location with the newly added window will be removed. That means A star just takes care of the shortest route from the initial position to the current checking window. The current checking window will be considered as the parent of that neighbor window. Secondly, the current checking window will be added to the close-set and be removed from the open-set. The next checking window will be selected in the open-set that has the lowest f score.

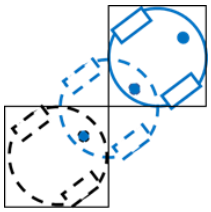

Figure 1 Robot lies in the middle of cells and move in the diagonal line.

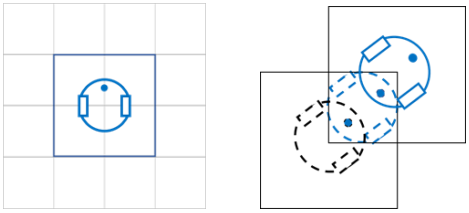

Figure 2. a) Robot lies in the middle of the searching window; b) Robot moves in the diagonal line without partial unobservable route.

That processing is repeated until a current checking window reaches to near the destination position (or reach to the destination position if the destination position aligned to the grid). The path-set is obtained by recursively taking the parent window of a current checking window. To easily construct formula in the next section, we denote (3) as the waypoint set (path-set) obtained from A Heuristic algorithm.

$$\mathbf{S} = \left\{ \mathbf{P}_A(v) \right\}_{v=1}^{n+1} \quad (3)$$

where $\mathbf{P}_A(v)$ is the v^{th} waypoint in the waypoint set that contains n+1 waypoints. $\mathbf{P}_A(1)$ and $\mathbf{P}_A(n+1)$ are the initial position and the target position of robot respectively.

C. Piecewise Cubic Bezier Curve

Named for Pierre Bezier, Bezier curves employ two endpoints called anchor points which defines the span of the line segment, which at least one additional point called control points- points used to give curviness to the original line segment. Bezier curve is a parametric curve that has an n-order polynomial function (4) based on n-1 control points.

$$\mathbf{B}(u) = \sum_{i=0}^{n} \binom{n}{i} (1-u)^{n-i} u^i \mathbf{P}_i \quad (4)$$

Directly using the original Bezier curve (4) to smooth robot trajectory turns out many disadvantages. Firstly, generating a path by using n-order Bezier curve based on n+1 waypoint causes difficultly reshaping. The shape of an n-orders Bezier curve is defined by the position of these points; it means that reshaping the curve equivalent to changing the position of these points. Secondly, the greater order of the Bezier curve is the more complex of the equation (4) lead to inefficient-time constructing the

curve. Thirdly, Bezier curve always pass through anchor point and contained in the convex hull assembled by control point set.

Therefore, the curve rarely passes through any point in the control point set. This property causes the great difference between the curve and initial path-set provided by A heuristic algorithm; and great rounding error between approximating and desired curve.

There are many ways to overcome this problem, divide and conquer is the basic approach. Robot trajectory is disjointed to many curve segments. The number of curve segments is based on the size of the waypoint set. By adding k control points to each curve segment, it matches a k+1 order Bezier curve. Each curve segment passes through two anchor points which are adjacent points in the waypoint set. The new curve is the piecewise (k+1) order Bezier curve and pass through all points of waypoint set. The second anchor point (endpoint) of the v^{th} curve shares the first anchor point (start point) of the $(v+1)^{th}$ curve. This reason addresses any piecewise Bezier curve is a continuous function (5).

$$\begin{aligned}&\mathbf{B}(u,v)=(1-u)^{k+1}\mathbf{P}_A(v)\\&+\sum_{j=1}^{k}\binom{k+1}{1}(1-u)^{k-j+1}u^j\mathbf{C}_j(i)\\&+u^{k+1}\mathbf{P}_A(v+1)\\&u\in[0,1],v\in\{0,1,...,n-1\}\end{aligned}\tag{5}$$

And the curve must also satisfy equations (6-8).

$$\mathbf{B}(0,v)=\mathbf{P}_A(v)\tag{6}$$

$$\mathbf{B}(1,v)=\mathbf{P}_A(v+1)\tag{7}$$

$$\mathbf{B}(1,v)=\mathbf{B}(0,v+1)\tag{8}$$

D. PCBC Based Path Planning

In order to adapt the Bezier curve into a robot trajectory, the curve requires meet robot kinematic constraints. Different from car-like robots, differential-drive two wheels robots may track on any trajectory type even if the trajectory has a discontinuous curvature as long as it has a good controlling algorithm. However, these trajectories are not time-optimal. With the robot's high speed and suddenly turning curve can make robots effortlessly slip out the desired trajectory. Therefore, trajectory with small and continuous curvature meets 2 wheels differential drive mobile robot kinematic and optimize travel time of robot. Each curve segment is the distinct Bezier curve which has k+1 order polynomial function i.e. each curve segment is C^{k+1} function. The problem is the point that links 2 segments. The entire curve is undifferentiable at these points. Therefore, equations (9) and its derivations (10,11) are established to satisfy kinematic constraint.

$$\kappa(1,v)=\kappa(0,v+1)\tag{9}$$

$$\dot{\mathbf{B}}(1,v)=\dot{\mathbf{B}}(0,v+1)\tag{10}$$

$$\ddot{\mathbf{B}}(1,v)=\ddot{\mathbf{B}}(0,v+1)\tag{11}$$

According to the equation (9), the minimum condition of each curve segment is to be a C^2 function i.e. each curve formed by at least two control points. In order to simplify equation (5) and meet the above condition, the cubic Bezier curve has been employed. By the equations (10,11), we can constrain almost the control points of the cubic Bezier curve segments except the first control point of the first curve segment and the second control point of the last curve segment since the couple equation (10,11) are not interdependent to them. However, we can set their constraints (12,13) according to the start heading and target heading of the robot in order to immediately put the robot into the trajectory without rotation in place.

$$\mathbf{C}_1(v)=\mathbf{B}(0,v)+\begin{bmatrix}\cos(\psi_A(v))\\\sin(\psi_A(v))\end{bmatrix}*C_1(v)\tag{12}$$

$$\mathbf{C}_2(v)=\mathbf{B}(1,v)-\begin{bmatrix}\cos(\psi_A(v+1))\\\sin(\psi_A(v+1))\end{bmatrix}*C_2(v)\tag{13}$$

Experimentally, $C_1(0)$ and $C_1(n-1)$ defined by the couple of equations (14). They determined by one-third of the Cartesian length between the corresponding point and the neighbor point in the path-set to get curvature not too large at the beginning and the end of the entire curve. $\mathbf{C}_1(n-1)$ is not used to construct the PCBC but its insertion completes the system of equation (10-13).

$$\begin{aligned}&\mathbf{C}_1(0)=\mathbf{P}_A(0)+\begin{bmatrix}\cos(\psi_A(0))\\\sin(\psi_A(0))\end{bmatrix}\frac{\|\mathbf{P}_A(1)-\mathbf{P}_A(0)\|}{3}\\&\mathbf{C}_1(n-1)=\mathbf{P}_A(n-1)\\&+\begin{bmatrix}\cos(\psi_A(n-1))\\\sin(\psi_A(n-1))\end{bmatrix}\frac{\|\mathbf{P}_A(n-1)-\mathbf{P}_A(n-2)\|}{3}\end{aligned}\tag{14}$$

E. Reproduce Anchor Point Set

A heuristic algorithm produces the consecutive discrete waypoints by it' searching property. Directly connect and smooth these points by employing C^2 PCBC produces the oscillated path caused by the short distance between adjective anchor points. The example of this problem has been illustrated in Fig. 3 and it' curvature in Fig. 4. Controlling robot tracking on the oscillated path that has many sharp turns (zig-zag) required fine control algorithms. With a large change of curvature respect to arc length and robot' high-speed tracking still make robots easily slip out of the desired path.

Therefore, although the path can be optimized for length, it is not possible to optimize the traveling time of the robot when tracking on it. For this reason, besides trying to adapt these paths by improving the control

algorithm, we propose the method in order to reduce the oscillation of the path. Li, Guanghui, et al introduced the regression search method [11] to reduce the number of waypoints before the smooth path algorithm takes it' role. In his experiments and results, this method performs very well. However, in the hard-turning case (greater than 90 degrees), we discovered this method leaves the large differential of curvature at these points. Moreover, with the long path, the processor can take a significantly long time to reduce the size of the waypoint set.

In this work, we reduce the anchor point of the C^2 PCBC after it formed instead of reducing the path-set directly obtained from A heuristic algorithm. The main idea of this method is to reproduce the new path by taking another waypoint set in the old path. Every new waypoint is equidistance with it' neighbor waypoints. This method required the Bezier curve parameterized by arc length. To the best of our knowledge, arc length parametrization of Bezier curves cannot be directly deduced from (15). Furthermore, in this method, we do not expect too exact results from the arc length. Therefore, the lookup table for converting between arc length and curve parameters is used in our method. The benefits of using the lookup table are easy to implement and give fast results with accuracy based on the amount of data.

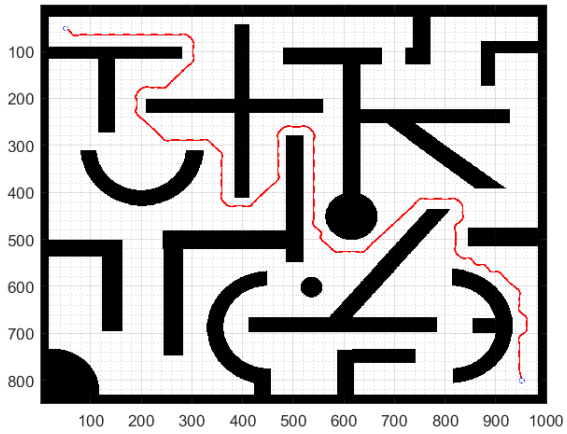

Figure 3. The smooth path formed by C^2 PCBC and raw waypoints produced by A Heuristic algorithm in the first setting.

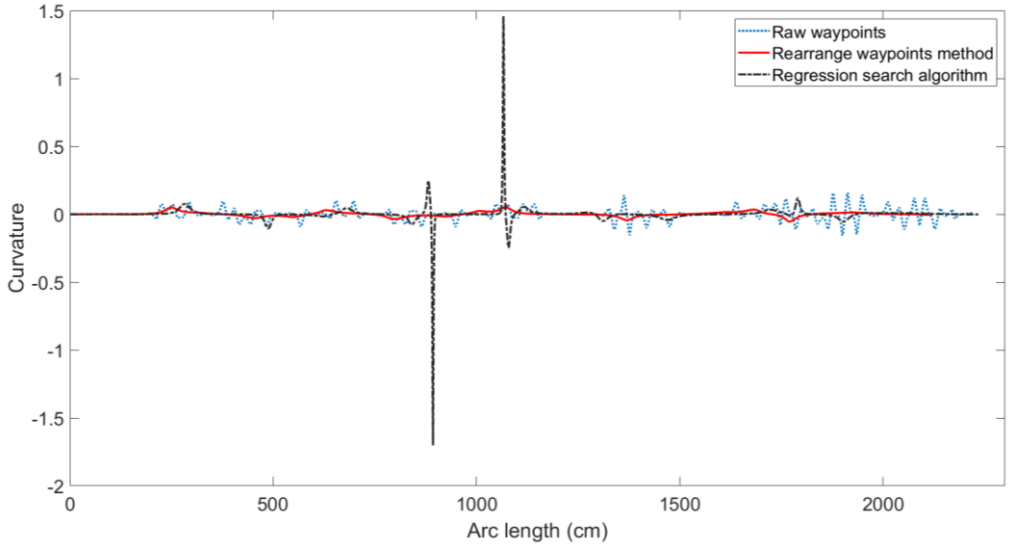

Figure 4. The curvature of the path in different cases in the first setting.

Let l be the arc length of υ^{th} segment given μ, can be found:

$$l(\mu,\upsilon)=\int_0^{\mu}\left\|\dot{\mathbf{B}}(u,\upsilon)\right\|du \tag{15}$$

The arc length of entire PCBC respect to curve parameters μ, υ (16) can be obtain by sum up total arc length of $\{v\}_0^{\upsilon-1\,\text{th}}$ curve segments and arc length of υ^{th} curve segment given μ.

$$L(\mu,\upsilon)=\sum_{v=0}^{\upsilon-1}l(1,v)+l(\mu,\upsilon) \tag{16}$$

For the convenience, we denote (17) as the new waypoint set obtained from reproduce anchor point set method.

$$\mathbf{S}^*=\left\{\mathbf{P}_A^*(\lambda)\right\}_{\lambda=1}^{m+1} \tag{17}$$

where $\mathbf{P}_A^*(\lambda)$ is the λ^{th} waypoint in the new waypoint set that contains m+1 waypoints. By feeding this set into (5) and (10-14), we have the new path. Let *h* be the arc length between arbitrary two adjacent new anchor points. By the lookup table, we can obtain the $\mathbf{P}_A^*(\lambda)$ that can be express as (18).

$$\mathbf{P}_A^*(\lambda)=\mathbf{B}\left(u\left((\lambda-1).h\right),v\left((\lambda-1).h\right)\right) \tag{18}$$

Since the start position and the target position of robot are constant, the new path must satisfy equations (19) and (20).

$$\mathbf{P}_A^*(1) = \mathbf{P}_A(1) \tag{19}$$

$$\mathbf{P}_A^*(m+1) = \mathbf{P}_A(n+1) \tag{20}$$

If h $\approx$ r_r , the path does not have a significant change. If h >> r_r , the path may have undesired behaviors such as encountering obstacles due to it' change. There is a trade-off between oscillation reduction and the difference between the curve and initial path-set. In our particular experiments' context, we have a fine-tuned parameter for our case. h is approximated about 4 times r_r . By deploying reproduce anchor point set method, the curvature of the new smooth path has significant oscillating reduction compared to the old path and the new path is still feasible.

III. Results and Experiments

In this section, we introduce two scenarios: simulation and real experiment in our robot. In both scenarios, the diameter of the circular boundary of robot is 0.4m. In the simulation scenario, C^2 PCBC and A Heuristic algorithm are tested in the maze with two different settings. These settings are summarized in Table I. The maze has many sharp turns, cul-de-sac cases and the size is 10x8.5m^2. Results of the first setting are divided into three cases. In the first case, Fig. 3 illustrates the path formed based on the raw waypoint set that directly obtained from A Heuristic algorithm. With the naked eyes, we almost cannot see the oscillation in the curvature of the path. However, in Fig. 4 that illustrates the curvature of the path produced by different reprocessing waypoints methods in the first setting, the oscillation in the curvature of the path when using raw waypoints clearly presents. Fig. 5 shows the result of the second case when the regression search method has been deployed. In this case, we also note the large curvature points that match with the curvature graph in Fig. 4. And the result of the last case improved from the first case by the reproduce anchor points method is presented in Fig. 6. In the second setting, the smooth path provided by A Heuristic algorithm and reproduce waypoints method is illustrated in the Fig. 7 and the curvature of the path provided by A Heuristic algorithm and different reprocess waypoints methods are showed in Fig. 8. In both settings, the reproduce anchor point set method generated the smoothest path compare to other methods. Both the regression search method and the reproduce anchor points method provide non-oscillation paths. However, the regression search method provides the path that contains many couples of points that have short arc length of curve segment and this property creates the large curvature between these points. While the reproduce anchor point set method provides a new path that contains equidistance anchor points respect to arc length, which gives result in smaller curvature and does not make sudden change of curvature.

TABLE I. Robot Settings in the Simulation Scenario

Setting #	Initial position	Initial heading	Destination position	Destination heading
1	Top-left	0 degree	Bottom-right	-90 degree
2	Bottom-left	0degree	Top-right	0 degree

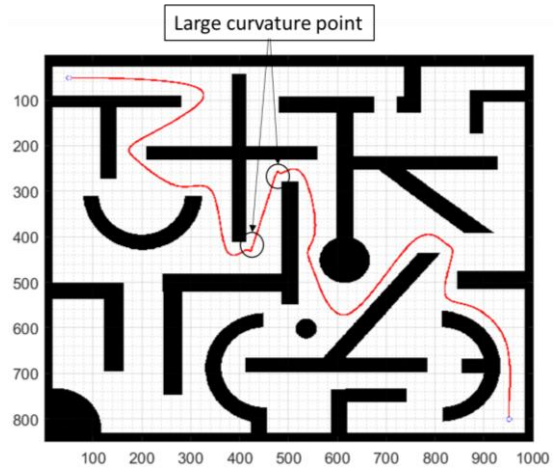

Figure 5. The smooth path formed by C^2 PCBC and the waypoint set reprocessed by regression search method.

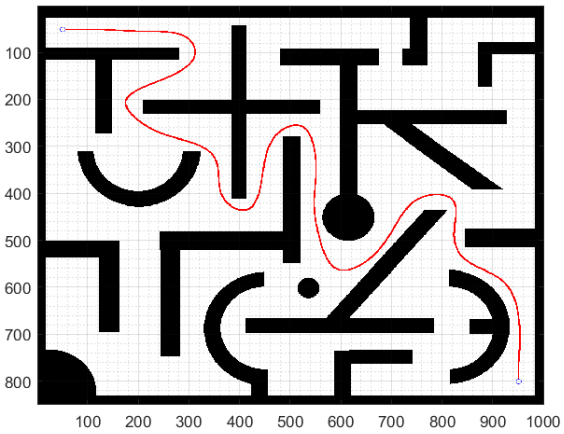

Figure 6. The smooth path formed by C^2 PCBC that defined by the new anchor point set provided by the reproduce anchor point method.

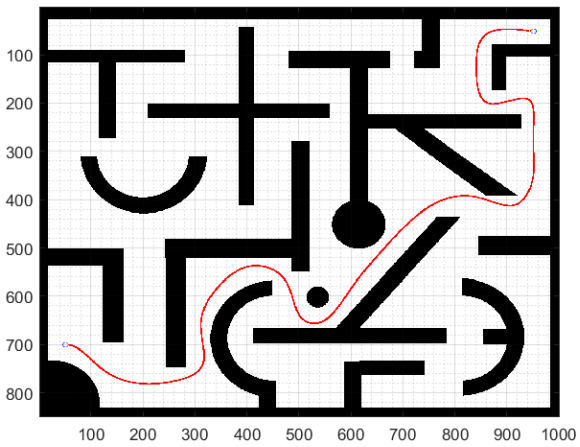

Figure 7. The smooth path formed by C^2 PCBC that has anchor point set provided by the reproduce anchor point set method in the second setting.

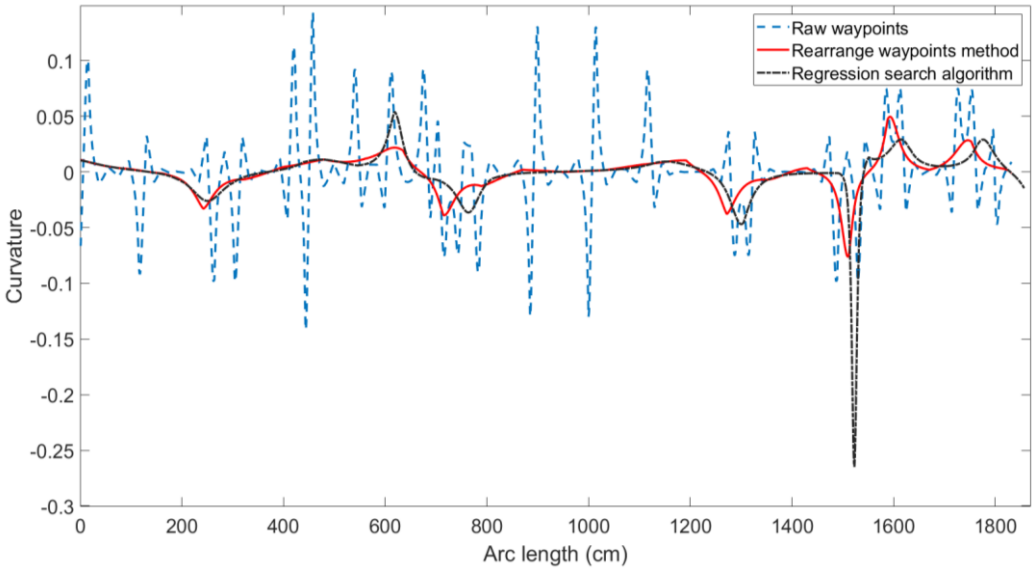

Figure 8. The curvature of the path in different cases in the second setting.

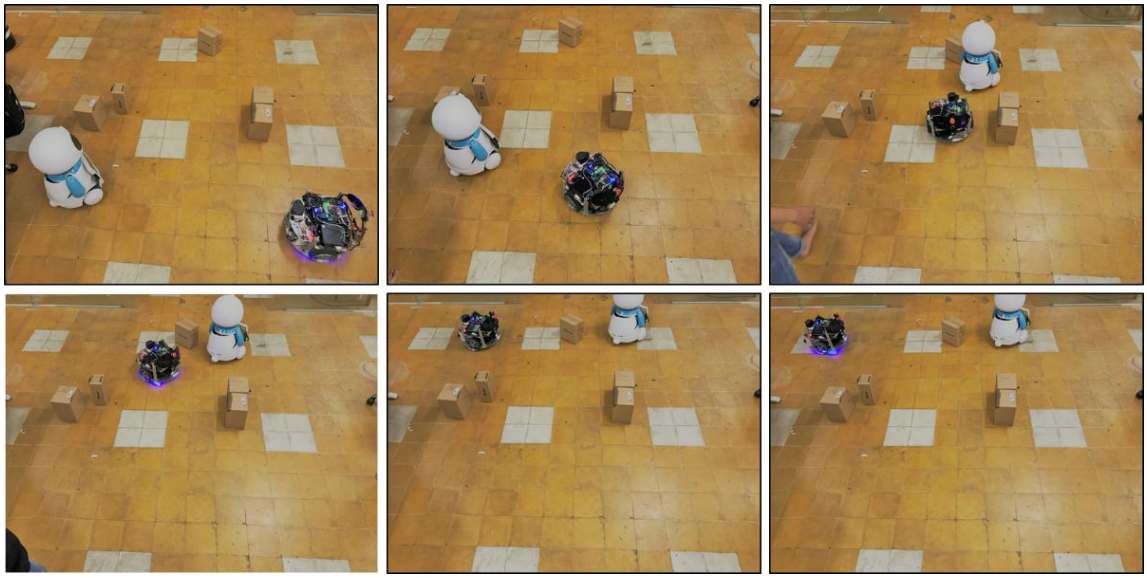

Figure 9. Smooth piecewise cubic Bezier curve is used in navigation algorithm of our robot.

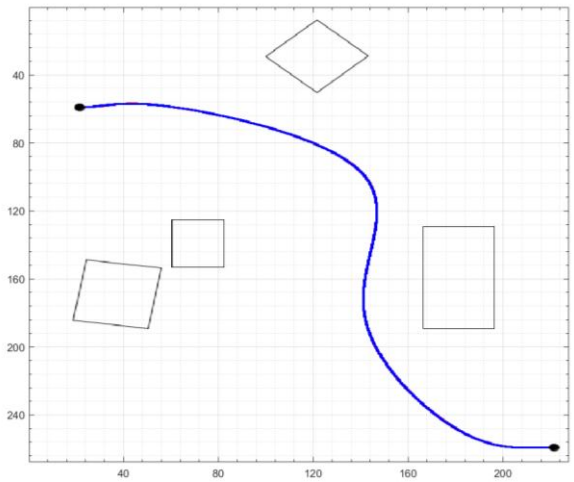

Figure 10. The trajectory of robot in real experiment is visualized by MATLAB.

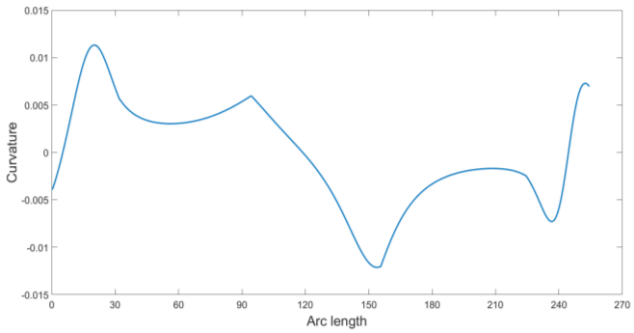

Figure 11. The curvature of trajectory of the real experiment respect to arc length is visualized by MATLAB.

In the second scenario, the simple context has been built with some rectangular obstacles and its screenshots are shown in Fig. 9. The start position and target position of the robot located in the bottom-right and top-left of the map (Fig. 10) respectively. In this scenario, the smooth

curvature piecewise cubic Bezier curve reprocess by the reproduce anchor point set method is used in the navigation algorithm of our robot. The initial waypoint set was taken from A heuristic algorithm. The result is showed in Fig. 10 and it' curvature is showed in Fig. 11. Based on the performance of our navigation algorithm, the robot easily tracks the given path.

IV. Conclusion

The article has shown the method of smooth curvature for PCBC respect to start pose and target pose of the robot. However, each curve segment is constructed by a cubic Bezier curve that causes algorithm does not have the ability to constrain the curvature of trajectory. To improve the algorithm, we propose some future works:

- Instead of using the cubic Bezier curve as the curve segment of the entire path, a higher-order Bezier curve will be deployed to fully take control of the shape (curvature and it' derivative) of the path. Having good shaping control, the method is not only optimal in length but also in robot travel time.

- Compacting the process of generating waypoints and smooth path planning by using the memetic algorithm will be implemented. Memetic algorithm is the upgraded version of the genetic algorithm. The main difference between them is the local search technique used in memetic algorithm. That may speed up to find out the most optimal path without premature convergence while our method cannot produce it.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Phan Gia Luan and Nguyen Truong Thinh contributed to the analysis and implementation of the research, to the analysis of the results and to the writing of the manuscript. All authors discussed the results and contributed to the final manuscript. Besides, Nguyen Truong Thinh conceived the study and were in charge of overall direction and planning. Nguyen Truong Thinh is a corresponding author.

Acknowledgment

The authors wish to thank Ho Chi Minh City University of Technology and Education, Vietnam. This study was supported financially by HCMUTE Open Lab and Ho Chi Minh City University of Technology and Education, Vietnam.

References

[1] Z. Han-ye, W. M. Lin, and A. X. Chen. "Path planning for the mobile robot: A review," *Symmetry* , vol. 10, no. 10, p. 450, 2018.
[2] Kunchev, Voemir, et al. "Path planning and obstacle avoidance for autonomous mobile robots: A review," *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Springer, Berlin, Heidelberg, 2006.
[3] Ravankar, Abhijeet, et al. "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, 2018, 3170.
[4] Z. Fengyu, B. Y. Song, and G. H. Tian, "B'ezier curve based smooth path planning for mobile robot," *Journal of Information & Computational Science*, vol. 8, no. 12, pp. 2441-2450, 2011.
[5] S. Baoye, Z. D. Wang, and L. Sheng. "A new genetic algorithm approach to smooth path planning for mobile robots," *Assembly Automation*, 2016.
[6] A. Marzoughi, "Switching navigation for a fleet of mobile robots in multi-obstacle regions," *International Journal of Mechanical Engineering and Robotics Research*, vol. 8, no. 1, pp. 1-5, January 2019.
[7] Kunchev, Voemir, et al. "Path planning and obstacle avoidance for autonomous mobile robots: A review," *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Springer, Berlin, Heidelberg, 2006.
[8] C. Xin and Y. M. Li, "Smooth path planning of a mobile robot using stochastic particle swarm optimization," in *Proc. IEEE 2006 International Conference on Mechatronics and Automation*, 2006.
[9] Guruji, A. Kumar, H. Agarwal, and D. K. Parsediya, "Time-efficient A* algorithm for robot path planning," *Procedia Technology*, vol. 23, pp. 144-149, 2016.
[10] ElHalawany, M. Basem, et al., "Modified a* algorithm for safer mobile robot navigation," in *Proc. 2013 5th International Conference on Modelling, Identification and Control (ICMIC). IEEE*, 2013.
[11] L. Guanghui, et al. "An efficient improved artificial potential field based regression search method for robot path planning," in *Proc. 2012 IEEE International Conference on Mechatronics and Automation*, 2012.

Phan Gia Luan is Graduate student of Ho Chi Minh City University of Technology and Education (Vietnam) with major is mechatronics. He has many years of experience researching in mobile robot, Motion capture, mechatronics, intelligent control. In addition, He also got many scientific research awards.

Nguyen Truong Thinh is Associate Professor of Mechatronics at Ho Chi Minh City University of Technology and Education (HCMUTE). He received his Ph.D in Mechanical Engineering at Chonnnam National University (Korea) in 2010 and obtained a positive evaluation as Associate Professor in 2012. His main research interests are Industrial Robotics, Service robotics, Mechatronics, Industrial Automation.