Practical high curvature path planning algorithm in joint space

Raimarius Delgado and Byoung Wook Choi✉

A practical high curvature path planning algorithm that considers the velocity limit of a mobile robot in joint space is proposed. The existence of obstacles that constrict the smooth movement of mobile robots is inevitable. To avoid incoming collision, the robot is redirected to a new path with respect to the dimensions of the obstacle and a safety factor. The redirected path consists of geometric constraints such as high curvatures on its turning points which cause difficulty in controlling the mobile robot. Thus, a central velocity generation method that considers both velocity limits through a convolution operator and high curvatures at turning points is suggested. The central velocity is able to be within the configured velocity limit, otherwise actual velocity commands in joint space could result. Therefore, interpolation to obtain uniform sampling and velocity downscaling is performed to generate a feasible joint space trajectory. The result shows short travelling time compared with previous methods while satisfying velocity limits in the joint space.

Introduction: A common methodology for a mobile robot to avoid a known obstacle is through the generation of continuous curves. Jolly *et al.* addressed a path planning technique using Bézier curves and the corresponding obstacle avoidance algorithm in [1]. The original path is redirected to avoid collision by generating two consecutive Bézier curves defined by ABCD and DEFS as shown in Fig. 1. The central velocity profile was generated considering only the acceleration limits and its relationship with the path's curvature. The generated maximum allowable velocity exhibits discontinuity at the junctions between redirections, velocity limit of the mobile robot was not considered and shows non-uniform sampling time. Feasible trajectories should be in uniform sampling to avoid difficulties in practical control, especially in real-time operation [2].

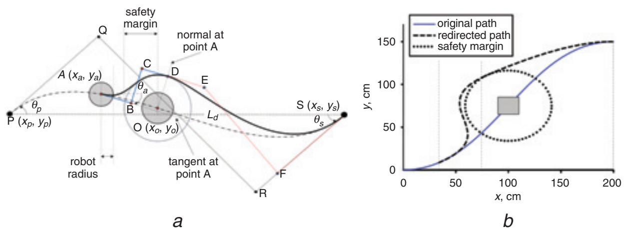

Fig. 1 *Reference path*

a Obstacle avoidance algorithm
b Redirected path to avoid obstacle

Convolution-based trajectory generation methods were suggested to consider the physical system limits in [3]. The convolution operator only considers the linear distance between two points, meaning the actual travelling distance along a Bézier curve is disregarded. Hence, Yang *et al.* in [4] suggested a heading angle that considered path planning based on convolution. The original parameter of the curve was transformed to the time domain by integrating the convolution-based central velocity to consider the actual distance of the curved path. However, another practical factor, the geometrical constraint of a high curvature path, should be taken into consideration [5–7]. For example, the first redirected path of ABCD which is focused for it possesses turning points that contemplate high curvature values.

Proposed high curvature path planning: The objective of this Letter is to solve the above drawbacks and formulate a feasible velocity profile method that considers both the physical limits of the robot and the curvature constraints of a given path. First, we suggest a method to obtain robot's optimal central velocity for the high curvature path while considering the velocity limits and the continuity of velocities between the paths. Interpolation is applied to the obtained central velocity to attain uniform sampling time, and finally a velocity downscaling scheme is presented to keep the velocity limit in the joint space to be actuated in real-time control.

In a path with a tolerable curvature, the velocities of turning points do not affect the velocities at the initial and terminal points. However, in Fig. 1*b*, the first part of the redirected path displays a high curvature, thus consideration of the turning point velocities is required.

The central velocity profile for the path is generated through convolution in the discrete time domain, defined as the iterated function [4]

$$v_a(t) = y_n[k] = \frac{y_{n-1}[k] - y_{n-1}[k - m_n]}{m_n} + y_n[k-1] \qquad (1)$$
$$k = t/T_s, \; m_n = t_n/T_s$$

where t denotes the time duration with respect to the actual travel distance along the curved path; T_s represents the sampling time; and n is the number of repetitions the convolution operator is executed–determined from the degree of the curved path.

For the high curvature path, the radius of the curvature, $\rho(u(t))$, is computed as the reciprocal of its curvature [1]. The velocity at the turning points is determined through the radial acceleration limit, $a_{r\max}$

$$v(u(t)) = \sqrt{a_{r\max}\rho(u(t))} \qquad (2)$$

where $u(t)$ is the optimised parameter to consider the heading angle along a Bézier curve in [3].

As seen in Fig. 1*a*, the proposed central velocity profile is generated by the application of the convolution operator in (1) from the initial point, A, to the terminal point, D, in the forward direction. In perspective, the velocity in the backward direction from the terminal point would produce exactly the same result as the velocity from the initial point and thus it is neglected.

In the case of the turning points, the velocity profile is divided into two parts and is generated with the same distance as the first step. For the first portion, the convolution operator is applied in the forward direction towards the velocity in the terminal point. For the second portion, velocity profile is generated in the opposite direction approaching the initial point.

Finally, the lowest velocity at each point from the various velocity profiles determines the proposed central velocity. Fig. 2*b* shows the generated velocity profiles from the proposed method and the corresponding central velocity $v_c(u(t))$.

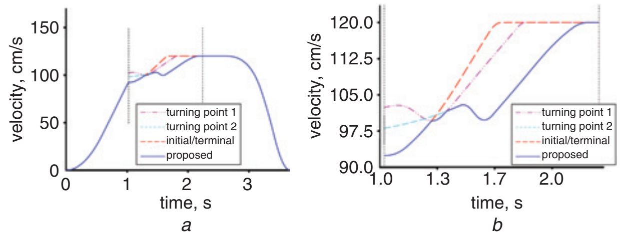

Fig. 2 *Proposed central velocity profile generation method*

a Velocity profile for whole path
b Velocity profile for high curvature region

The proposed central velocity profile is able to produce values that consider the path's curvature and the physical limits of the mobile robot. The travelling time is extended because of considering the velocity limit when generating the forwarding part of turning point 2. Hence, the time duration between sampling points is altered to conform to the proposed velocity through

$$dt = \frac{dL(u(t))}{v_c(u(t))} \qquad (3)$$

where $v_c(u(t))$ is the proposed central velocity; $dL(u(t))$ is the distance between each sampling point; and dt is the required time sampling for the actual distance along the planned path.

Consequently, the sampling time attained through this notation would express non-uniformity. Since uniform time samples are required for practical control of the mobile robot, a variation of linear interpolation is administered to acquire feasible results

$$v_{uni}(t) = v_c(t_{k-1}) + \frac{v_c(t_k) - v_c(t_{k-1})}{t_k - t_{k-1}} \times (t - t_{k-1}) \qquad (4)$$

where v_{uni} denotes the proposed velocity in uniform sampling time; t_k is the numerical integration of (2) using Simpson's rule [8]; and t is the arbitrary uniform sampling time, bounded within $t_{k-1} \leq t \leq t_k$.

Furthermore, actual velocity commands for actuating the mobile robot depend on the angular velocities; both left and right wheels. Although the central velocity could satisfy the given velocity limit, it is possible for the joint space velocity to exceed this limit, as shown in Fig. 3. To keep the joint space velocity within the limits of the mobile robot, a downscaling algorithm is presented in (4) to reduce the central velocity limit of the robot as much as the excess. After the downscale, the velocity profile generation process is repeated with the downscaled velocity, v_{dscal}, as the velocity limit

$$\begin{aligned} v_{\mathrm{over}} &= \max(\max(v_{\mathrm{r}}, v_{\mathrm{l}})) \\ \Delta v &= v_{\mathrm{over}} - v_{\mathrm{max}} \\ v_{\mathrm{dscal}} &= v_{\mathrm{max}} - \Delta v \end{aligned} \tag{5}$$

where v_{r} and v_{l} are the right and left wheel velocities, respectively.

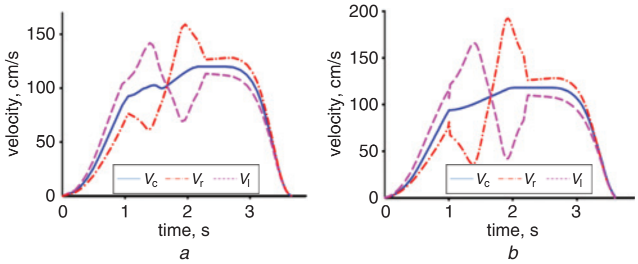

Fig. 3 *Joint space velocities*

a From proposed velocity profile
b From Yang's central velocity

Simulation results: Fig. 1*b* shows a predetermined curve with an initial position at (0, 0), heading angle of 0° and initial velocity of 0 cm/s. The terminal point (200, 150) was configured to have the same angle and velocity values as the initial point. The obstacle is located at (100.12, 75.16) and the joint velocity limit of the mobile robot was set to 120 cm/s. To be able to perform the convolution operation, the actual distance along the path was computed, and was found to be 289.95 cm. The central velocity profile was generated through the discussed approaches and was compared with the criteria of time optimality, time sampling uniformity and the efficiency of the generated velocity.

The proposed velocity values show a total travel time of 3.73 s. Relatively, Yang's central velocity presented by the initial/terminal graph in Fig. 2*b* travelled the same path in a shorter duration at 3.59 s. However, the velocity at the turning points is not considered which results in higher joint space velocities.

Fig. 3 shows a comparison between Yang's central velocity and the proposed joint space velocity profiles. As expected, both methods produced joint space velocities that exceeded the velocity limit. Hence, the downscaling scheme in (4) was implemented. The proposed joint space velocity transcended the limit lower than Yang's velocity, meaning it requires lesser downscaling. The overall travel time computed for Yang's method was 5 s, whereas the proposed method shows a faster travelling time of 4.59 s, making it relatively time optimal while satisfying the physical limits of the mobile robot and the geometrical constraints of the path shown in Fig. 4*a*.

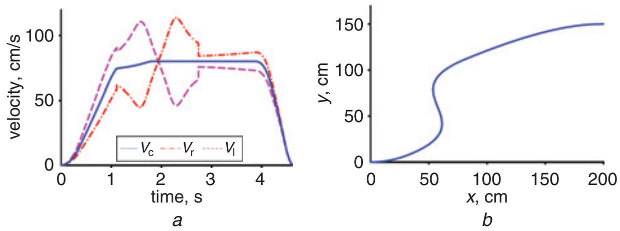

Fig. 4 *Simulation results*

a Joint space velocity after scaling down velocity limit
b Actual robot trajectory driven by joint space velocity

The proposed algorithm was applied in a designed mobile robot through anyKode Marilou simulator neglecting environmental factors, such as friction and slip [9]. The trajectory generated, shown in Fig. 4*b*, shows high accuracy in tracking the reference path to avoid a known obstacle.

Conclusion: This Letter introduces a practical path planning method to generate a feasible velocity profile that considers both the physical limits of the robot and the geometrical constraints of the path. The accurate results obtained through different numerical and actual simulations indicate that the proposed velocity profile generation for a practical path planning algorithm is highly capable of tracking a high curvature path while observing its geometric constraints and the robot's physical limits. In comparison with previous methods, the presented approach has shown time optimality, uniform time sampling and accurate velocity values. This approach opens up possibilities for application in practical operations such as in real-time systems.

Acknowledgments: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. 2012-006057).

6 November 2014
doi: 10.1049/el.2014.3926
One or more of the Figures in this Letter are available in colour online.

Raimarius Delgado and Byoung Wook Choi (*Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul, Republic of Korea*)

✉ E-mail: bwchoi@seoultech.ac.kr

References

1 Jolly, K.G., Kumar, R.S., and Vijayakumar, R.: 'A Bézier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits', *Robot. Auton. Syst.*, 2009, **57**, (1), pp. 23–33
2 Jeon, J.D., and Lee, B.H.: 'Ellipse-based velocity obstacles for local navigation of holonomic mobile robot', *Electron. Lett.*, 2014, **50**, (18), pp. 1279–1281
3 Lee, G., and Choi, Y.: 'Convolution-based trajectory generation methods using physical system limits', *J. Dyn. Syst. Meas. Control*, 2013, **135**, (1), pp. 1–8
4 Yang, G.J., and Choi, B.W.: 'Smooth trajectory planning along Bézier curve for mobile robots with velocity constraints', *Int. J. Control Autom.*, 2013, **6**, (2), pp. 225–234
5 Yang, K., Jung, D., and Sukkarieh, S.: 'Continuous curvature path-smoothing algorithm using cubic Bézier spiral curves for non-holonomic robots', *Adv. Robot.*, 2013, **27**, (4), pp. 247–258
6 Tounsi, M., and Le Corre, J.F.: 'Trajectory generation for mobile robots', *Robot. Res.*, 1996, **41**, (3–4), pp. 367–376
7 Cowlagi, R.V.: 'Curvature-bounded transversability analysis in motion planning for mobile robots', *IEEE Proc.*, 2014, **30**, (4), pp. 1011–1019
8 Tseng, C.C.: 'Digital integrator design using Simpson rule and fractional delay filter', *IEE Proc., Vis. Image Signal Process.*, 2006, **153**, (1), pp. 79–85
9 anyKode, Marilou Robotics Studio. Available at http://www.anykode.com