Trajectory Planning for Omni-Directional Mobile Robot Based on Bezier Curve, Trigonometric Function and Polynomial

Chaobin Chen, Haichen Qin, and Zhouping Yin

State Key Laboratory of Digital Manufacturing Equipment and Technology,
Huazhong University of Science and Technology,
Luoyu Road 1037, 430074 Wuhan, China
ccbhahao@163.com

Abstract. This paper proposes a systematic trajectory planning algorithm for omni-directional mobile robot, which consists of three parts: path planning, line velocity planning and posture planning. Third-order Bezier curve is applied to the path planning algorithm and proved to be feasible by analysis of radius of curvature. Further more, the method that constructing a complex path by splicing multi-segment Bezier curves turns out to be effective. Then trigonometric function is utilized in the line velocity planning algorithm. The results demonstrate effectiveness and rationality of the algorithm. Lastly, based on trigonometric function and polynomial, a posture planning algorithm for omni-directional mobile robot is designed to plan the posture, angular velocity and angular acceleration. Results show that it can guarantee the continuity of the angular acceleration with excellent effect. On the basis of the case study on a complex path, it is concluded that the trajectory planning algorithm has good adaptability.

Keywords: Trajectory Planning, Omni-Directional Mobile Robot, Bezier Curve, Path Planning.

1 Introduction

In recent years, mobile robots have been widely applied in various fields, and relevant researches have made lots of achievements. Under holonomic constraints [1], omni-directional mobile robot has three degrees of freedom in the horizontal motion plane [2]. So it is much more flexible than ordinary non-omni-directional mobile robot, which makes it especially suitable for working in tight space [3].

In the research on omni-directional mobile robot, trajectory planning is a critical field. For most mobile robots, trajectory planning aims to find out the optimized trajectory with respect to a given objective function [4]. Up till now, there are many research methods about this issue. In the research on three-wheeled omni-directional vehicle, Kuo-Yang Tu advanced a trajectory planning method based on kinematic constraints and linear dynamics. According to the method, velocity, acceleration, angular velocity and angular acceleration were planned, via 3 cases: two intersecting lines, a straight line and a circle [5]. In Hongjun Kim and Byung-Kook Kim's

C.-Y. Su, S. Rakheja, H. Liu (Eds.): ICIRA 2012, Part III, LNAI 7508, pp. 352–364, 2012.

research on three-wheeled omni-directional mobile robot, the energy consumption is taken into account. They established the minimum-energy translational trajectory planning algorithm to achieve an optimal trajectory [6]. By designing the velocity and acceleration filters, Hashemi, E., Jadidi, M.G. and Babarsad, O.B. built up a dynamic model for four-wheeled omni-directional mobile robot, and utilized artificial intelligence and machine vision package in positioning and path planning [7]. Hongxia Zhang and Kyung Seok Byun applied trapezoidal acceleration profile to velocity planning and acceleration planning. Thus they achieved the real time path planning for three-wheeled omni-directional mobile robot [8].

Since proposed, Bezier curve has been widely used in computer graphics and animation [9]. With advantages such as symmetry and first-order continuous, in recent years it has been applied to trajectory planning for mobile robots. Gregor Klančar and Igor Škrjanc worked on collision-avoidance problem based on Bernstein-Bezier path tracking for multiple robots with known constraints [10], using a two-wheeled differentially driven robot. Khatib, M., Jaouni, H., Chatila, R. and Laumond, J.P. applied Bezier polynomials to a smoothing method, and made the original path more reasonable to meet the kinematic constraints [11]. Jung-Hoon Hwang, Arkin, R.C. and Dong-Soo Kwon utilized Bezier curve in creating a smooth trajectory, which was applied to online trajectory generation for supervisory control [12]. As for path planning for the automatic guided vehicle (AGV), Petrinec, K. and Kovačić, Z. proposed a method that combined Bezier curve and spline functions [13]. To sum up, most studies on the application of Bezier curve in trajectory planning are in connection with nonholonomic mobile robots.

This paper is organized as follows: Section 2 begins by describing the Bezier curve and its mathematical expression. In section 3, a path planning algorithm for the omni-directional mobile robot based on third-order Bezier curve is proposed. Section 4 discusses a trigonometric velocity planning algorithm. Section 5 focuses on the posture planning based on polynomial and trigonometric function. Using the above-mentioned theories, a complicated trajectory is planned for an omni-directional mobile robot in section 6. Finally, section 7 provides a brief conclusion of this paper.

2 Bezier Curve

Bezier curve was invented in 1962 by the French engineer Pierre Bezier for designing automobile bodies [9]. Different from other curves, Bezier curve doesn't go through data points that define it, which are known as control points. With fewer turning points, Bezier curve is much smoother than other spline curves on the same order [14].

Given position vectors of n+1 points in a certain space as $\mathrm{P}_i(i = 0,1, \dots, n)$, Bezier curve can be defined as

$$P(u) = \sum_{i=0}^{\mathrm{n}} P_i Bin(u), \quad u \in [0,1] \tag{1}$$

Where u is a parameter. n is the order of Bezier curve, and i is the summation index. $\mathrm{P}_i(i = 0,1, \dots, n)$ constitute the Bezier characteristic polygon. $Bin(u)$ is the Bernstein function, given by

$$Bin(u) = C_n^i u^i (1-u)^{n-i} = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}, (i = 0,1, \cdots n) \tag{2}$$

Especially, $0^0 = 1$, $0! = 1$.

Given four control points (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), a third-order Bezier curve can be expressed as

$$\begin{cases} x = x_0(1-u)^3 + 3x_1(1-u)^2 u + 3x_2(1-u)u^2 + x_3 u^3 \\ y = y_0(1-u)^3 + 3y_1(1-u)^2 u + 3y_2(1-u)u^2 + y_3 u^3 \end{cases} \quad u \in [0,1] \tag{3}$$

Bezier curve is generated as u changes from 0 to 1.

For instance, when the coordinate is defined in cm, given control points (1, 2), (2, 4), (5, 2), (6, 3), we can get the Bezier curve shown in Fig. 1.

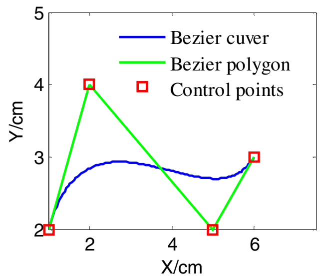


Fig. 1. A Bezier curve

3 Path Planning Based on Third-Order Bezier Curve

This section discusses how to apply third-order Bezier curve to path planning for omni-directional mobile robots.

From Eq. (3), we can see that, to construct a third-order Bezier curve, these four control points must be determined: a start point, an end point and two intermediate points. However, generally in practice, we should construct third-order Bezier curves between adjacent points when information of a series of points is given. It means that only the information of the start point and the end point is known. Thus before path planning, we need to construct two intermediate points according to some criteria.

Now, we can make use of the information of the start point and the end point, especially the directions of the velocity vectors. With the information of the start point and the end point known, the third-order Bezier curve should satisfy the following conditions as

$$\begin{cases} f(x_s) = y_s \\ df(x_s)/dx = tan(\alpha_s) \end{cases} \tag{4}$$

$$\begin{cases} f(x_g) = y_g \\ df(x_g)/dx = tan(\alpha_g) \end{cases} \tag{5}$$

Where $f(x)$ is the mathematical expression of this Bezier curve. (x_s, y_s) and (x_g, y_g) are the coordinates of the start point and the end point, whose velocity directions are α_s and α_g.

Eqs. (4) and (5) restrict the third-order Bezier curve. For Bezier curve, the tangent directions of the start point and the end point are respectively the same as the first and the last edge of the characteristic polygon [15]. Therefore, to satisfy the above constraints, the missing intermediate coordinates must meet the following conditions:

$$\begin{cases} (y_2 - y_s) \cdot cos(\alpha_s) = (x_2 - x_s) \cdot sin(\alpha_s) \\ (y_g - y_3) \cdot cos(\alpha_g) = (x_g - x_3) \cdot sin(\alpha_g) \end{cases} \tag{6}$$

Where (x_2, y_2) and (x_3, y_3) are the coordinates of the intermediate points.

According to Eq. (6), the intermediate points can be expressed as follows:

$$\begin{cases} y_2 = y_s + \lambda_1 sin(\alpha_s) \\ x_2 = x_s + \lambda_1 cos(\alpha_s) \end{cases} \tag{7}$$

$$\begin{cases} y_3 = y_g - \lambda_2 sin(\alpha_g) \\ x_3 = x_g - \lambda_2 cos(\alpha_g) \end{cases} \tag{8}$$

Where λ_1 and λ_2 are positive, which represent the influence of the velocity directions of the start point and the end point. The greater λ_1 and λ_2 are, the greater the influence is. Actually, in order to make the omni-directional mobile robots adjust the velocity direction as soon as possible, λ_1 tends to be smaller, but λ_2 tends to be greater, on the premise that the mechanical requirements are fulfilled.

λ_1 and λ_2 should correlate to the distance between the start point (x_s, y_s) and the end point (x_g, y_g). Assuming this correlation is proportional, λ_1 and λ_2 can be

$$\begin{cases} \lambda_1 = D_1 \sqrt{(x_s - x_g)^2 + (y_s - y_g)^2} \\ \lambda_2 = D_2 \sqrt{(x_s - x_g)^2 + (y_s - y_g)^2} \end{cases} \tag{9}$$

Where D_1 and D_2 are scale factors which are greater than 0.

According to Eq. (9), we can obtain λ_1 and λ_2, and substitute them into Eqs. (7) and (8). Then the coordinates of intermediate points are available.

Thus, the Bezier curve that connects P_s and P_g can be expressed as follows:

$$\begin{cases} x = x_s(1-u)^3 + 3x_2(1-u)^2 u + 3x_3(1-u)u^2 + x_g u^3 \\ y = y_s(1-u)^3 + 3y_2(1-u)^2 u + 3y_3(1-u)u^2 + y_g u^3 \end{cases} \quad u \in [0,1] \tag{10}$$

Where (x_2, y_2) and (x_3, y_3) are determined by Eqs. (7), (8) and (9).

For instance, when the coordinate is defined in cm, with $(x_s, y_s) = (2, 4)$, $(x_g, y_g) = (4, 6)$, $\alpha_s = 60°$, $\alpha_g = 120°$ and $D_1 = D_2 = 1/4$, we can obtain the third-order Bezier curve that links the start point and the end point, based on Eqs. (7) ~ (10). The output is clearly shown in Fig. 2.

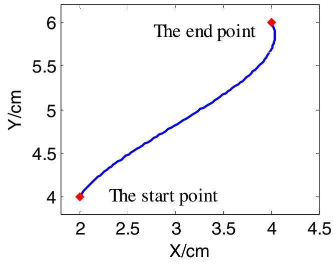


Fig. 2. Result of path planning

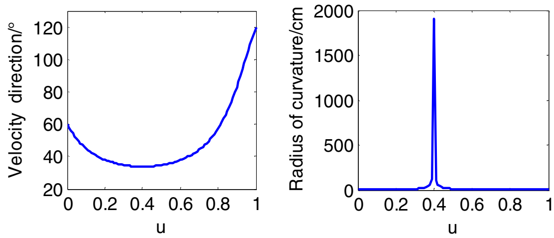
Additionally, the tangential direction and radius of curvature can be calculated as

$$\alpha = atan(dy/dx) \tag{11}$$

$$r = 1/((d^2y \cdot dx - dy \cdot d^2x) * (dx \cdot dx + dy \cdot dy)^{-1.5}) \tag{12}$$

Where α is the velocity direction, and r is the radius of curvature.

Taking the path in Fig. 2 as an example, we can obtain the velocity direction curve and the radius of curvature curve which are shown in Fig. 3.



(a) Velocity direction (b) Radius of curvature

Fig. 3. Velocity direction curve and radius of curvature curve

By the method above, we can generate n Bezier curves between n+1 control points, and splice them together. It is interesting that they are still first-order continuous. Eventually we can obtain a complete path planning algorithm for omni-directional mobile robot based on third-order Bezier curve. For example, when the coordinate is defined in cm, some information is in the following table.

Table 1. Information of three control points

Control point	Coordinates/cm	Velocity direction
point A	(2, 5)	30°
point B	(6, 9)	60°
point C	(16, 7)	45°

With $D_1 = D_2 = 1/4$, the outcome is shown below.

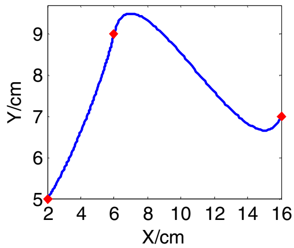


Fig. 4. A simple path

4 Line Velocity Planning Based on Trigonometric Function

Taking dynamics and smoothness running into account, the trajectory of the omni-directional mobile robots should be second-order derivative, so that it can avoid impact and improve kinematics performance.

Suppose the path is a Bezier curve, while the start velocity is v_s, and the end velocity is v_g, the robot has to meet the following constraints:

1) Continuous acceleration ;

2) The integral of velocity with respect to time (ie, distance) should equal the length of the previously planned Bezier curve.

In view of the first constraint, let acceleration and time meet the law of sine curve as follows:

$$a(t) = Asin(\varphi t) \tag{13}$$

Where $a(t)$ is the acceleration function, A is the maximum acceleration, φ is an undetermined parameter.

Integrating Eq. (13) on both sides, we can get the relationship between velocity and time as follows:

$$v(t) = -A/\varphi \cdot cos(\varphi t) + B \tag{14}$$

Where $v(t)$ is the velocity function, and B is an undetermined parameter. As a result of $v(0) = -A/\varphi + B = v_s$, there is:

$$A = \varphi(B - v_s) \tag{15}$$

When t=0, distance equals zero. Integrating Eq. (14) on both sides, we will obtain the relationship between distance $S(t)$ and time as follows:

$$S(t) = -A/\varphi^2 \cdot sin(\varphi t) + Bt \tag{16}$$

Sum up Eqs. (13), (14) and (16) as follows:

$$\begin{cases} a(t) = Asin(\varphi t) \\ v(t) = -A/\varphi \cdot cos(\varphi t) + B \\ S(t) = -A/\varphi^2 \cdot sin(\varphi t) + Bt \end{cases} \tag{17}$$

In the equations above, A, φ and B need to be determined. T stands for the time that the robot spends on the whole path. In order to make acceleration continuous at intersections of adjacent Bezier curves, the easiest way is to let the acceleration equal zero at each start point and end point, that is, $Asin(\varphi T) = 0$. Because $A \neq 0$, $sin(\varphi T) = 0$.

And $v(T) = -A/\varphi \cdot cos(\varphi T) + B = v_g$. Substituting Eq. (15) into it, there is:

$$(v_s - B) \cdot cos(\varphi T) + B = v_g \tag{18}$$

As $sin(\varphi T) = 0$, $cos(\varphi T) = \pm 1$. When $cos(\varphi T) = 1$, Eq.(18) is not permanent tenability. So $cos(\varphi T) = -1$. Substituting it into Eq. (18), the result is given by:

$$B = (v_g + v_s)/2 \tag{19}$$

Now let's come to the second constraint. The length of the Bezier curve can be obtained by calculus. Dividing the control variable u into n equal parts, we can get n +1 points on the Bezier curve. Connect adjacent points with straight-line and then calculate the length of each section. Finally, adding up the length of all sections, we can approximate get the length of the Bezier curve. The greater n is, the closer the result will approach the theoretical value. When n is great enough, the error is so small that the outcome can be regarded as approximations to the theoretical value. Length of the path (S) is obtained by calculus as follows:

$$S = -A/\varphi^2 \cdot sin(\varphi T) + BT = (v_g + v_s)T/2 \tag{20}$$

Solving the equation above, T can be defined as

$$T = 2S/(v_g + v_s) \tag{21}$$

As $sin(\varphi T) = sin(2S\varphi/(v_g + v_s)) = 0$, we can transform it into another expression: $2S\varphi/(v_g + v_s) = \pi$, which also can be written as

$$\varphi = \pi(v_g + v_s)/(2S) \tag{22}$$

Substituting Eq. (22) into Eq. (15), the maximum acceleration is defined by

$$A = \pi({v_g}^2 - {v_s}^2)/(4S) \tag{23}$$

Substituting Eqs. (19), (21), (22) and (23) into Eq. (17), we can get three equations which meet the above-mentioned constraints as follows:

$$\begin{cases} a(t) = \pi({v_g}^2 - {v_s}^2)/(4S) \cdot sin\big(\pi t(v_g + v_s)/(2S)\big) \\ v(t) = (v_s - v_g)/2 \cdot cos\big(\pi t(v_g + v_s)/(2S)\big) + (v_g + v_s)/2 \\ S(t) = S(v_s - v_g)/(\pi(v_g + v_s)) \cdot sin\big(\pi t(v_g + v_s)/(2S)\big) + (v_g + v_s)t/2 \end{cases} \tag{24}$$

Therefore, given the coordinates, velocity and velocity directions of two points, firstly we can plan the path and obtain its length S. Then substituting S into Eq. (24), we can figure out how distance, velocity and acceleration vary with time. Connecting all of adjacent Bezier curves, we can get the entire result of line velocity planning.

Taking the path in Fig. 4 as an example, Table 2 gives the information of the points.

Table 2. Information of the 3 control points

Control point	Coordinates/cm	Velocity/(cm/s)	Velocity direction
point A	(2, 5)	10	30°
point B	(6, 9)	20	60°
point C	(16, 7)	30	45°

The relationship between the acceleration and time is illustrated below.

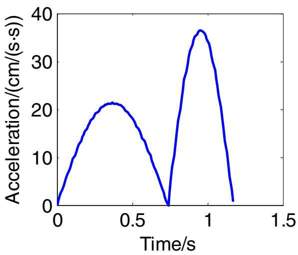
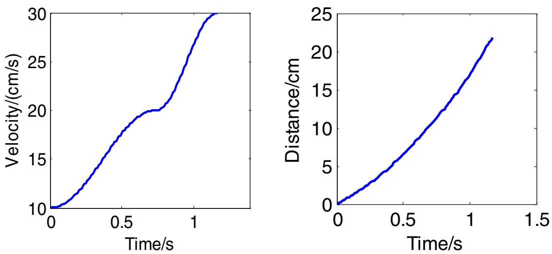


Fig. 5. The relationship between the acceleration and time

Fig. 5 clearly shows that the acceleration is continuous.

The velocity curve and the distance curve are shown in Fig. 6.



(a) Velocity (b) Distance

Fig. 6. The velocity curve and the distance curve

Fig. 5 and **Fig. 6** illustrate that the line velocity planning algorithm based on trigonometric function is entirely reasonable.

5 Posture Planning

Compared with ordinary two-wheeled mobile robot, a prominent advantage of the omni-directional mobile robot is that it can achieve flexible posture adjustment (namely rotating movement) and rapid translation at the same time. The flexibility allows it to perform complex work better.

In order to control the posture, it must be planed reasonably. Next, a posture planning algorithm for the omni-directional mobile robot will be discussed.

Generally in posture planning, the posture and expected angular velocity at the start and end points (i.e., θ_s, θ_g, ω_s, ω_g in turn) are known. We should design a reasonable law of posture variation, making the posture and angular velocity at the start and end points equal the set value, with the angular acceleration continuous. At the same time, another implicit restriction must be met, that is, the time spent during the posture planning must equal the time T that has been identified in line velocity planning.

In order to make angular acceleration continuous, let it equal zero at the start and end points. Considering trigonometric function and quadratic polynomial, the relationship between angular acceleration and time can be designed as follows:

$$\beta(t) = Msint + Nt^2 + Ct \tag{25}$$

Where $\beta(t)$ is the angular acceleration function, M, N and C are undetermined parameters, t stands for time. Integrating Eq. (25) on both sides, the expression of angular velocity is given by

$$\omega(t) = -Mcost + \frac{1}{3}Nt^3 + \frac{1}{2}Ct^2 + E \tag{26}$$

Where $\omega(t)$ is the angular velocity function and E is an undetermined parameter. Integrating Eq. (26) on both sides, angle (i.e., posture) is given by

$$\theta(t) = -Msint + \frac{1}{12}Nt^4 + \frac{1}{6}Ct^3 + Et + F \tag{27}$$

Where $\theta(t)$ is the posture function. F is an undetermined parameter.

Now let's come to the boundary conditions of Eqs. (25) ~ (27). For Eq. (25), it's obvious that $\beta(0)$=0. Similarly, the following equation is also necessary.

$$\beta(T) = MsinT + NT^2 + CT = 0 \tag{28}$$

For Eq. (26), the boundary conditions are defined as

$$\omega(0) = -M + E = \omega_s \tag{29}$$

$$\omega(T) = -McosT + \frac{1}{3}NT^3 + \frac{1}{2}CT^2 + E = \omega_g \tag{30}$$

For Eq. (27), the posture of the start and end points can be defined as

$$\theta(0) = F = \theta_s \tag{31}$$

$$\theta(T) = -MsinT + \frac{1}{12}NT^4 + \frac{1}{6}CT^3 + ET + F = \theta_g \tag{32}$$

Therefore, we can get the simultaneous equations below.

$$\begin{cases} MsinT + NT^2 + CT = 0 \\ -M + E = \omega_s \\ -McosT + \frac{1}{3}NT^3 + \frac{1}{2}CT^2 + E = \omega_g \\ F = \theta_s \\ -MsinT + \frac{1}{12}NT^4 + \frac{1}{6}CT^3 + ET + F = \theta_g \end{cases} \tag{33}$$

Obviously, we can obtain the unique solution. The outcome is shown below.

$$\begin{cases} M = (\theta_g - \theta_s - 0.5T(\omega_s + \omega_g))/(0.5T - sinT + T^2 sinT/12 + 0.5TcosT) \\ C = 6(\omega_g - \omega_s - M(1 - cosT) + MTsinT/3)/T^2 \\ N = (-MsinT - CT)/T^2 \\ E = M + \omega_s \\ F = \theta_s \end{cases} \tag{34}$$

Substituting ω_s, ω_g, θ_s, θ_g and T into Eq. (34), we can obtain M, C, N, E and F in turn. Then substituting these parameters into Eqs. (25)~(27), we can achieve the posture planning for the omni-directional mobile robot.

Still take the path in Fig. 4 as an example. With other information unchanged, posture at point A equals 0°, and its angular velocity is 0°/s; posture at point B equals 10°, and its angular velocity is 20°/ s; posture at point C equals 20°, and its angular velocity is 0°/s. We can obtain the outcome of posture planning as follows:

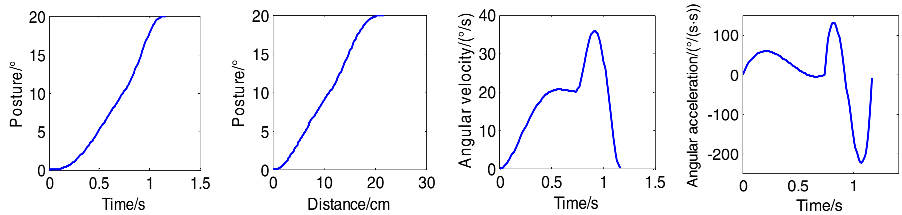


Fig. 7. Results of posture planning

From the results in Fig. 7, it's clear that the posture curve and the angular velocity curve are relatively smooth when the robot moves from point A to point C.

6 A Case Study for Omni-Directional Mobile Robot

After systematically introducing the trajectory planning algorithm for omni-directional mobile robot, in this section, we will apply it to practical operation and test its effect. The expected information of control points is given in the following table.

Table 3. Information of control points

Number	Coordinates/cm	Velocity/(cm/s)	Velocity direction	Posture	Palstance /(°/s)
1	(326, 50)	0	18°	30°	0
2	(650, 150)	200	14.9313°	30°	0
3	(987.5, 240)	165	14.9313°	60°	0
4	(1090, 350)	140	90°	60°	0
5	(1000, 425)	110	180°	60°	0
6	(900, 425)	90	180°	60°	0
7	(800, 400)	0	−143.2394°	60°	0

According to the information in Table 3, we can carry out the trajectory planning. With $D_1 = D_2 = 1/3$, the result of path planning is clearly shown in Fig. 8.

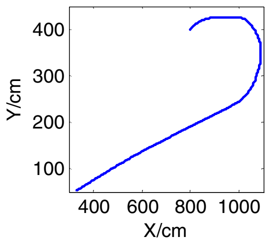


Fig. 8. The result of path planning

It can be seen from Fig. 8 that the whole path is relatively smooth. Some other results are as follows:

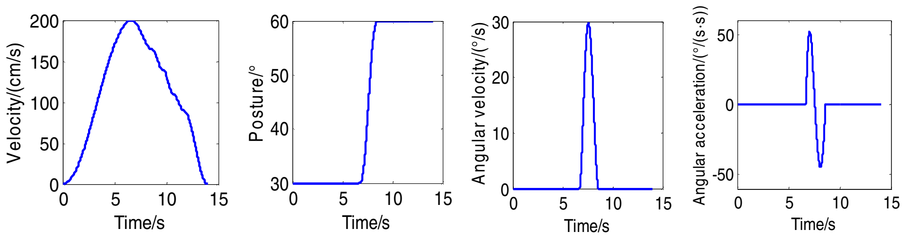


Fig. 9. Some other results

These results suggest that even though the path is very complicated, this trajectory planning algorithm still has a good effect. Therefore, the algorithm has been applied to the 10th ABU Asia-Pacific Robot Contest which was held in Bangkok in 2011.

7 Conclusion

The trajectory planning algorithm for omni-directional mobile robot which is presented in this paper consists of three parts: path planning which is based on third-order Bezier curve, line velocity planning which is based on trigonometric function, posture planning which is based on polynomial and trigonometric function. These parts are closely linked, step by step. Each part carries out detailed theoretical analysis and effect validation. Finally, using the algorithm, a more complicated trajectory is planned, and the results show that the algorithm has good adaptability to work in complex environment. The algorithm applies to not only competition robots, but also industrial and household omni-directional mobile robots. The trajectory planning under dynamic environment is not taken into account in this paper, and this will be emphasis in our future research.

References

1. Liu, Y., Wu, X., Zim Zhu, J., Lew, J.: Omni-Directional Mobile Robot Controller Design by Trajectory Linearization. In: Proceedings of the American Control Conference, pp. 3423–3428. IEEE Press, New Jersey (2003)
2. Treesatayapun, C.: A Discrete-Time Stable Controller for an Omni-Directional Mobile Robot Based on an Approximated Model. Control Engineering Practice 19, 194–203 (2011)
3. Zhao, D., Yi, J., Deng, X.: Structure and Kinematic Analysis of Omni-Directional Mobile Robots. Robot 25, 394–398 (2003)
4. Kim, K.-B., Kim, B.-K.: Minimum-Time Trajectory for Three-Wheeled Omnidirectional Mobile Robots Following a Bounded-Curvature Path with a Referenced Heading Profile. IEEE Transactions on Robotics 27, 800–808 (2011)
5. Tu, K.-Y.: A Linear Optimal Tracker Designed for Omnidirectional Vehicle Dynamics Linearized Based on Kinematic Equations. Robotica 28, 1033–1043 (2010)
6. Kim, H., Kim, B.-K.: Minimum-Energy Translational Trajectory Planning for Battery-Powered Three-Wheeled Omni-Directional Mobile Robots. In: Proceedings of the International Conference on Control, Automation, Robotics and Vision, pp. 1730–1735. IEEE Press, New Jersey (2008)
7. Hashemi, E., Jadidi, M.G., Babarsad, O.B.: Trajectory Planning Optimization with Dynamic Modeling of Four Wheeled Omni-Directional Mobile Robots. In: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 272–277. IEEE Press, New Jersey (2009)
8. Zhang, H., Byun, K.S.: Real Time Path Planning Using Trapezoidal Acceleration Profile for Omnidirectional Mobile Robot. In: Proceedings of the International Conference on Control, Automation and Systems, pp. 830–833. IEEE Press, New Jersey (2008)
9. Choi, J.-w., Elkaim, G.H.: Bezier Curve for Trajectory Guidance. In: Proceedings of the World Congress on Engineering and Computer Science, pp. 625–630. International Association of Engineers, Hong Kong (2008)
10. Klančar, G., Škrjanc, I.: A Case Study of the Collision-Avoidance Problem Based on Bernstein-Bezier Path Tracking for Multiple Robots with Known Constraints. Journal of Intelligent and Robotic Systems 60, 317–337 (2010)
11. Khatib, M., Jaouni, H., Chatila, R., Laumond, J.P.: Dynamic Path Modication for Car-Like Nonholonomic Mobile Robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2920–2925. IEEE Press, New Jersey (1997)

12. Hwang, J.-H., Arkin, R.C., Kwon, D.-S.: Mobile Robots at Your Fingertip: Bezier Curve On-line Trajectory Generation for Supervisory Control. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 1444–1449. IEEE Press, New Jersey (2003)
13. Petrinec, K., Kovačić, Z.: The Application of Spline Functions and Bezier Curves to AGV Path Planning. In: Proceedings of the IEEE ISIE Conference, pp. 1453–1458. IEEE Press, New Jersey (2005)
14. Jolly, K.G., Kumar, R.S., Vijayakumar, R.: A Bezier Curve Based Path Planning in a Multi-Agent Robot Soccer System without Violating the Acceleration Limits. Robotics and Autonomous Systems 57, 23–33 (2009)
15. Wang, X., Shang, J.: Concrete Application of Triple Bezier Curve Jointing Model in Method of Curve-Rization to Broken-Lined Contours. Science of Surveying and Mapping 36, 192–194 (2011)