HIGH TECHNOLOGY LETTERS | Vol. 23 No. 2 | June 2017 | pp. 141 ~ 148

doi:10.3772/j. issn. 1006-6748. 2017. 02. 004

A transition method based on Bezier curve for trajectory planning in cartesian space①

Zhang Shaolin (张少林)* **, Jing Fengshui* **, Wang Shuo②* **
(* The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China)
(** University of Chinese Academy of Sciences, Beijing 100190, P. R. China)

Abstract

In order to smooth the trajectory of a robot and reduce dwell time, a transition curve is introduced between two adjacent curves in three-dimensional space. G2 continuity is guaranteed to transit smoothly. To minimize the amount of calculation, cubic and quartic Bezier curves are both analyzed. Furthermore, the contour curve is characterized by a transition parameter which defines the distance to the corner of the deviation. How to define the transition points for different curves is presented. A general move command interface is defined for receiving the curve limitations and transition parameters. Then, how to calculate the control points of the cubic and quartic Bezier curves is analyzed and given. Different situations are discussed separately, including transition between two lines, transition between a line and a circle, and transition between two circles. Finally, the experiments are carried out on a six degree of freedom (DOF) industrial robot to validate the proposed method. Results of single transition and multiple transitions are presented. The trajectories in the joint space are also analyzed. The results indicate that the method achieves G2 continuity within the transition constraint and has good efficiency and adaptability.

Key words: transition method, Bezier curve, G2 continuity, transition constraint

0 Introduction

A robot program consists of several motion commands and each command defines a curve. The most common curves are lines and circles, which are connected head-to-tail in sequence. However, two adjacent curves may be not smooth at the intersection. The robot has to halt at the terminal of a curve to avoid velocity fluctuation. So, it is necessary to introduce a transition part between two curves to smooth the trajectory and reduce the dwell time.

In PLCopen Motion Control Specifications[1,2], the way of connecting two curves without halt is called blending mode. In this mode, a transition curve is inserted between two adjacent curves. In order to transit smoothly, the transition curve needs to satisfy some smoothness criteria. G2 continuity[3] is usually adopted as the criterion. Furthermore, the transition curve should be characterized to adapt to different applications. For example, smoothness is more important for a transportation robot, and accuracy is more important for a welding robot. So, the smoothness and accuracy should be able to be adjusted for different applications.

There are already many investigations about transition curves, especially in the field of transition between lines[4-10]. Sencer, et al.[4] proposed a method to transit between adjacent lines with quintic B-splines. G2 continuity was guaranteed and the cornering tolerance could be set by the user. Bi, et al.[5] utilized cubic Bezier curve to transit between adjacent lines. Also, G2 continuity was guaranteed, and the curvature of the transition curve was analyzed. Zhao, et al.[6] utilized a curvature-continuous B-spline with five control points for transition. Hota, et al.[7] proposed a path named γ-trajectory for transition. Their studies showed that various methods could be used to transit between lines. But they did not illustrate how to determine the order of the transition curve.

The transition including circles has also been investigated in some fields[11-14]. Habib, et al.[11] described a method based on a single cubic Bezier curve

① Supported by the National Natural Science Foundation of China (No. 61573358) and Research and Development of Large Multi-function Demolition Equipment in Disaster Site (No. 2015BAK06B00).

② To whom correspondence should be addressed. E-mail: shuo. wang@ ia. ac. cn

Received on May 17, 2016

to join two circles. The transition curve was S-shaped or C-shaped, and of G2 continuity. This method was applied to highway and railway route design. A similar work was done by Ahmad A et al.[12], but quartic Bezier spiral was used for transition instead. Rashid, et al.[13] proposed an S-shaped transition curve to join two tangent circles of the same diameter, which was used to design a Spur Gear Tooth. Most of the studies focused on planar transition in different applications. However, few studies have been done on transition between adjacent lines and circles with shape control, especially in three-dimensional space.

In this paper, a transition method is developed based on Bezier curve to achieve G2 continuity. For efficiency of the algorithm, a single curve is adopted for transition. Cubic Bezier curve is tried first because it is of low degree and easily calculated. If cubic Bezier curve does not meet the smoothness constraint, quartic Bezier curve will be used. Different situations are discussed separately, including transition between two lines, transition between a line and a circle and transition between two circles. The lines and circles are supposed to be in three-dimensional space without any limitations for the circle radii, and the length of line and circle. Furthermore, to characterize the contour curve, a transition parameter (TP) which defines the distance to the corner of the deviation is adopted. If a robot is moving along a curve, the transition will start when the remaining length is shorter than TP.

The remaining part of this paper is organized as follows. Section 1 introduces a general transition interface and a planning procedure. How to transit between two adjacent curves is presented in Section 2. The transition method is demonstrated with experiments in Section 3. Finally, conclusions are given in Section 4.

1 Transition interface and planning procedure

Transition curve is inserted between two adjacent curves and the speed is re-planned, as shown in Fig. 1. Transition planning is a part of trajectory planning. Transition planning reads motion commands from program, and prepares the curves for interpolation. For example, there are three move line (MovL) commands in a robot program. The transition interface for program is shown in Fig. 2, the desired trajectory is shown in Fig. 3, and the general transition planning procedure is shown in Fig. 4.

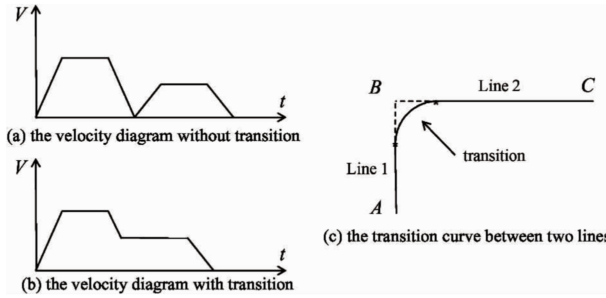

Fig. 1 Velocity diagram for transition

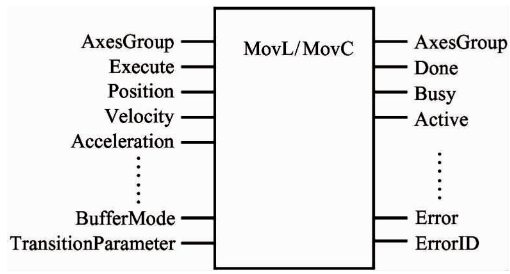

Fig. 2 the MovL/MovC command[1,2]

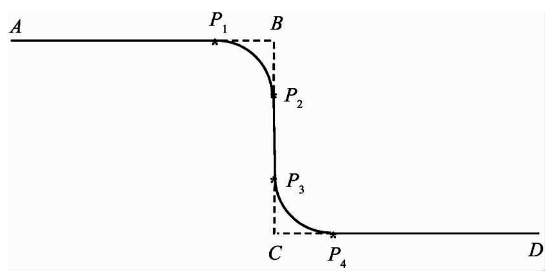

Fig. 3 An example of three lines for transition

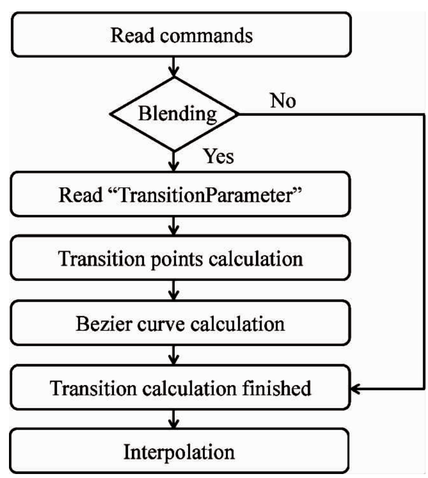

Fig. 4 A flow chart for transition planning procedure

Firstly, the transition planning task reads transition commands. If the "TransitionMode" parameter of the MovL (line AB) command is "blending", a transition curve will be inserted between line AB and line BC. Transition P_1P_2 starts at point P_1 and ends at point P_2. The length of line P_1B and line BP_2 are defined as

Algorithm 1. TP_{AB} here is short for "Transition Parameter" parameter of the MovL (line AB) command. The lengths of line P_3C and line CP_4 are calculated similarly.

If the length of line AB and line BC are both larger than $2 \cdot TP_{AB}$, the length of line P_1B and line BP_2 are equal to TP_{AB}. Otherwise, they are defined in terms of the length of line AB and line BC. So, line BC may have two different transition points or two overlapping points. More examples are shown in Fig. 5.

Algorithm 1 Calculation of *Length* (P_1B) and *Length* (BP_2)

Input: $Length(AB)$, $Length(BC)$, TP_{AB}
Output: $Length(P_1B)$, $Length(BP_2)$
1: **if**
  $Length(AB) > 2 \cdot TP_{AB}$
  $Length(BC) > 2 \cdot TP_{AB}$
 then
2: $Length(P_1B) = TP_{AB}$
3: **else**
4: $Length(P_1B) = min(Length(AB), Length(BC))/2$
5. **end if**
6. $Length(BP_2) = Length(P_1B)$

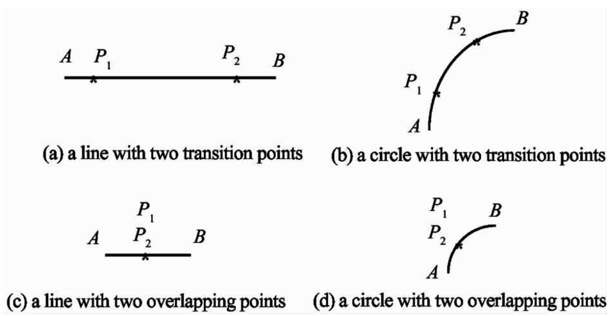

Fig. 5 The transition points of lines and circles

After getting the information of the transition points, the transition curve could be calculated, as will be introduced in Section 2. Then, it is ready for interpolation.

2 A transition method based on a single Bezier curve

2.1 Preliminaries

Given spatial control points P_i ($i = 0, 1, 2, \cdots, n$), the interpolation for each point on the Bezier curve is

$$C(u) = \sum_{i=0}^{n} P_i B_{i,n}(u),\ u \in [0,1] \quad (1)$$

where

$$B_{i,n}(u) = C_n^i u^i (1-u)^{n-i},\ i = 0, 1, \cdots, n \quad (2)$$

The Bezier curve is a weighted average of each control point. It begins at P_0 and ends at P_n. The Bezier curve has the convex hull property, which means that the curve does not "undulate" more than the polygon of its control points. For cubic Bezier curve ($n = 3$), four control points are needed. For higher-order curves, the amount of computation will be larger and more intermediate points are needed.

The derivatives for a Bezier curve at $C(0)$ and $C(1)$ are

$$\begin{aligned} C'(0) &= n(P_1 - P_0) \\ C'(1) &= n(P_n - P_{n-1}) \end{aligned} \quad (3)$$

The second derivatives are

$$\begin{aligned} C''(0) &= n(n-1)(P_2 - 2P_1 + P_0) \\ C''(1) &= n(n-1)(P_n - 2P_{n-1} + P_{n-2}) \end{aligned} \quad (4)$$

For G2 continuity, the adjacent curves share a common tangent direction and a common center of curvature at the join point[3]. The curvature at $C(0)$ and $C(1)$ should be

$$\kappa = \frac{|C'(u) \times C''(u)|}{|C'(u)|^3},\ u = \{0, 1\} \quad (5)$$

Substituting Eq. (3) and Eq. (4) into Eq. (5) yields

$$\kappa = \frac{(n-1)|(P_1 - P_0) \times (P_2 - P_1)|}{n|(P_1 - P_0)|^3},\ u = 0 \quad (6)$$

$$\kappa = \frac{(n-1)|(P_n - P_{n-1}) \times (P_{n-2} - P_{n-1})|}{n|(P_n - P_{n-1})|^3},\ u = 1 \quad (7)$$

2.2 Transition between two lines

Fig. 6 shows a case of a cubic Bezier curve ($n = 3$) transiting from line AB to line BC. Point P_0 and point P_3 are the transition points set by Algorithm 1.

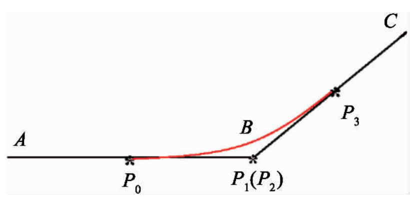

Fig. 6 The transition between two lines

1) The transition curve should be tangent with line AB and line BC.

From Eq. (3), control point P_1 should be on line AB, and control point P_2 should be on line BC.

$$\begin{aligned} \overrightarrow{P_0P_1} \times \overrightarrow{P_0B} &= 0 \\ \overrightarrow{BP_3} \times \overrightarrow{P_2P_3} &= 0 \end{aligned} \quad (8)$$

2) The transition curve should have the same curvature with line AB and line BC.

$$\begin{aligned} \kappa &= 0, \ u = 0 \\ k &= 0, \ u = 1 \end{aligned} \tag{9}$$

From Eqs(6) ~ (9), point P_1 and point P_2 should overlap at point B. Then, the transition Bezier curve is given by point P_0, point P_1, point P_2, and point P_3.

2.3 Transition between a line and a circle

Fig. 7 shows a case of a cubic Bezier curve (n = 3) transiting from line AB to circle BC. Point P_0 and point P_3 are the transition points set by Algorithm 1.

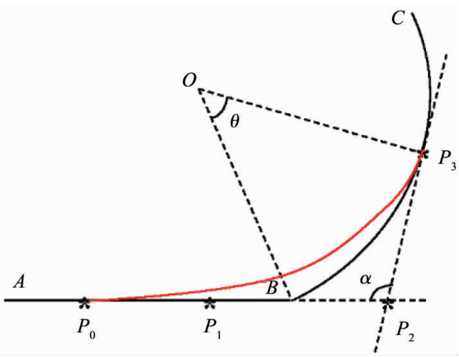

Fig. 7 The transition between a line and a circle (cubic Bezier curve)

1) The transition curve should be tangent with line AB and circle BC.

From Eq. (3), control point P_1 should be on line AB, and control point P_2 should be on the tangent line of circle BC at point P_3.

$$\overrightarrow{P_0B} \times \overrightarrow{P_0P_1} = 0 \tag{10}$$

$$\begin{aligned} &\overrightarrow{OP_3} \cdot \overrightarrow{P_2P_3} = 0 \\ &\overrightarrow{OP_2} \cdot (\overrightarrow{OB} \times \overrightarrow{OP_3}) = 0 \end{aligned} \tag{11}$$

2) The transition curve should have the same curvature with line AB and circle BC.

$$\begin{aligned} \kappa &= 0, \ u = 0 \\ k &= 1/r, \ u = 1 \end{aligned} \tag{12}$$

where r is the radius of circle BC.

From Eqs(6),(7) and Eqs(10) ~ (12), point P_1 and point P_2 are defined. If point A, point B, point C and point O are coplanar, the solution is given as follows. Otherwise, there is no solution.

1) Point P_2 is the intersection of line P_0P_1 and line P_3P_2.

2) From Eq. (7) and Eq. (12), Eq. (13) is got. Then, point P_1 is given by Eq. (10) and Eq. (13).

$$| P1 - P2 | = \frac{3}{2r\sin(\alpha)} | P3 - P2 |^2 \tag{13}$$

where $\alpha = \angle P1P2P3$.

Quartic Bezier curve (n = 4) could meet the smoothness constraints here (see Fig. 8). Point P_0 and point P_4 are the transition points set by Algorithm 1. Point P_1, point P_2 and point P_3 are given as follows:

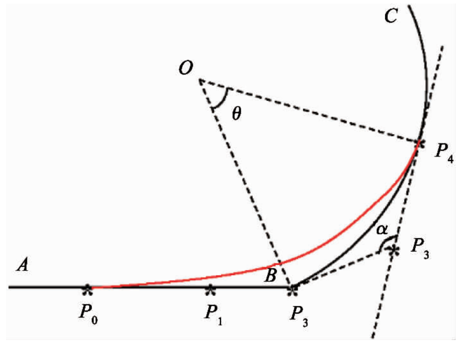

Fig. 8 The transition between a line and a circle (quartic Bezier curve)

1) Point P_2

Intuitively, in order to track the given trajectory, point P_2 should be around line AB or circle BC. For the simplicity of calculation, point P_2 is set at point B.

2) Point P_3

Similar to Eq. (11), there exists:

$$\begin{aligned} &\overrightarrow{OP_4} \cdot \overrightarrow{P_3P_4} = 0 \\ &\overrightarrow{OP_3} \cdot (\overrightarrow{OB} \times \overrightarrow{OP_4}) = 0 \end{aligned} \tag{14}$$

From Eq. (7) and Eq. (12),

$$\begin{aligned} | P_4 - P_3 |^2 &= \frac{3}{4} r | P_2 - P_3 | \sin(\alpha) \\ &= \frac{3}{4} r^2 (1 - \cos(\theta)) \end{aligned} \tag{15}$$

where $\alpha = \angle P2P3P4$, $\theta = \angle P2OP4$, $0 < \theta < 2pi$.

From Eq. (14) and Eq. (15), point P_3 is obtained.

3) Point P_1

From Eq. (6), Eq. (10) and Eq. (12), there are multiple solutions for point P_1. An optimization index can be added to obtain the optimal solution. One answer is to add a constraint as Eq. (16). Approximately, ε means a measure of curvature. Thus to get the minimum of ε makes the transition curve bend at least[15].

$$\varepsilon = \int_0^1 C''(u) \cdot C''(u) \mathrm{d}u \tag{16}$$

Let $d\varepsilon / dl_1 = 0$, $l_1 = | P_1 - P_0 |$. Substituting Eq. (1) into Eq. (16) yields

$$\begin{aligned} l_1 = \frac{1}{16(a_1^2 + b_1^2 + c_1^2)} (&4a_1^2 d + 4b_1^2 d \\ &+ 4c_1^2 d + 4a_1 a_2 l_2 + 4b_1 b_2 l_2 + 4c_1 c_2 l_2 \\ &- 3a_1 x_0 + 3a_1 x_4 - 3b_1 y_0 + 3b_1 y_4 - 3c_1 z_0 \\ &+ 3c_1 z_4) \end{aligned} \tag{17}$$

where

$$l_2 = | P_4 - P_3 |,$$
$$P_0 = (x_0, y_0, z_0),$$
$$P_4 = (x_4, y_4, z_4),$$
$$\frac{\overrightarrow{P_0P_2}}{|P_0P_2|} = (a_1, b_1, c_1),$$
$$\frac{\overrightarrow{P_4P_3}}{|P_4P_3|} = (a_2, b_2, c_2),$$
$$d = TP.$$

From Eq. (10) and Eq. (17), point P_1 is obtained.

Then, the transition Bezier curve is given by point P_0, point P_1, point P_2, point P_3 and point P_4.

2.4 Transition between two circles

Fig. 9 shows a case of a cubic Bezier curve (n = 3) transiting from circle AB to circle BC. Point P_0 and point P_3 are the transition points set by Algorithm 1.

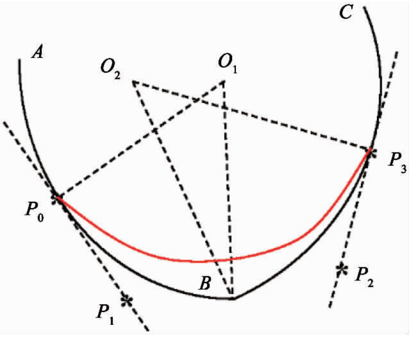

Fig. 9 The transition between two circles (cubic Bezier curve)

1) The transition curve should be tangent with circle AB and circle BC.

From Eq. (3), control point P_1 should be on the tangent line of circle AB at point P_0, and control point P_2 should be on the tangent line of circle BC at point P_3.

$$\begin{aligned} &\overrightarrow{O1P0} \cdot \overrightarrow{P0P1} = 0 \\ &\overrightarrow{O1P_1} \cdot (\overrightarrow{O1B} \times \overrightarrow{O1P0}) = 0 \end{aligned} \tag{18}$$

$$\begin{aligned} &\overrightarrow{O2P3} \cdot \overrightarrow{P3P2} = 0 \\ &\overrightarrow{O2P_2} \cdot (\overrightarrow{O2B} \times \overrightarrow{O2P3}) = 0 \end{aligned} \tag{19}$$

2) The transition curve should have the same curvature with circle AB and circle BC.

$$\begin{aligned} &\kappa 1 = 1/r1, \ u = 0 \\ &k2 = 1/r2, \ u = 1 \end{aligned} \tag{20}$$

where r_1 is the radius of circle AB, and r_2 is the radius of circle BC.

From Eqs(6), (7) and Eqs(18) ~ (20), point P_1 and point P_2 are defined. However, it is difficult to obtain the analytical solutions here. Numerical method can be used to solve these functions, but with a large amount of computation.

Quartic Bezier curve (n = 4) could meet the smoothness constraints here, see Fig. 10. Point P_0 and point P_4 are the transition points set by Algorithm 1. Point P_1, point P_2 and point P_3 are given as follows:

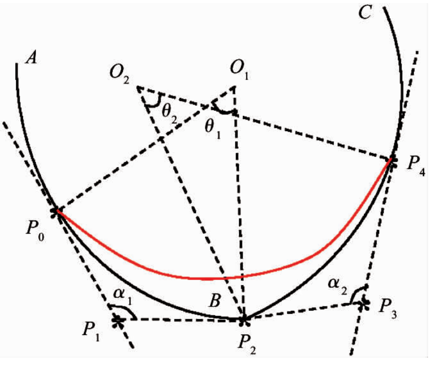

Fig. 10 The transition between two circles (quartic Bezier curve)

1) Point P_2

Similar to Section 2.3, point P_2 is set at point B.

2) Point P_3

Similar to Eq. (19), there exists:

$$\begin{aligned} &\overrightarrow{O2P4} \cdot \overrightarrow{P4P3} = 0 \\ &\overrightarrow{O2P_3} \cdot (\overrightarrow{O2B} \times \overrightarrow{O2P4}) = 0 \end{aligned} \tag{21}$$

From Eq. (7) and Eq. (20),

$$\begin{aligned} |P_4 - P_3|^2 &= \frac{3}{4} r_2 |P_2 - P_3| \sin(\alpha_2) \\ &= \frac{3}{4} r_2^2 (1 - \cos(\theta 2)) \end{aligned} \tag{22}$$

where $\alpha_2 = \angle P2P3P4$, $\theta 2 = \angle P2O2P4$, $0 < \theta 2 < 2pi$.

From Eq. (21) and Eq. (22), point P_3 is obtained.

3) Point P_1

Similar to Eq. (21) and Eq. (22), there exist:

$$\begin{aligned} &\overrightarrow{O1P0} \cdot \overrightarrow{P0P1} = 0 \\ &\overrightarrow{O1P_1} \cdot (\overrightarrow{O1B} \times \overrightarrow{O1P0}) = 0 \end{aligned} \tag{23}$$

$$\begin{aligned} |P_1 - P_0|^2 &= \frac{3}{4} r_1 |P_2 - P_1| \sin(\alpha 1) \\ &= \frac{3}{4} r_1^2 (1 - \cos(\theta 1)) \end{aligned} \tag{24}$$

where $\alpha_1 = \angle P_0P_1P_2$, $\theta_1 = \angle P_0O_1P_2$, $0 < \theta_1 < 2pi$.

From Eq. (23) and Eq. (24), point P_1 is obtained.

Then, the transition Bezier curve is given by point P_0, point P_1, point P_2, point P_3 and point P_4.

Mark 1 Although Figs6 ~ 10 illustrate conditions for planning on plane, the transition method is also feasible for spatial planning, as will be shown in Section 3.

3 Experiments

The transition method is evaluated by several experiments on a six DOF robot—ER20-C10. The control system is shown in Fig. 11. The original motion controller is replaced by an industrial computer CX5130 made by Beckhoff company. In addition to the transition method, some other components are also realized for the experiment, such as robot program interpreter, trajectory planning method for line and circle commands, and forward and inverse kinematics.

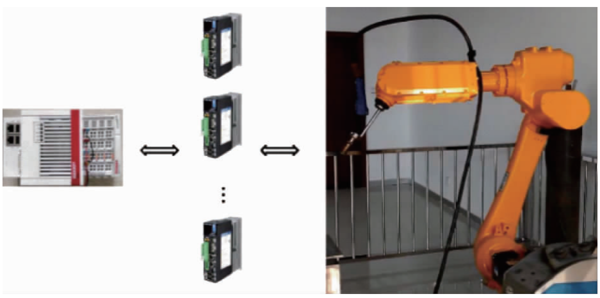

Fig. 11 The control system of a six DOF robot

To verify the feasibility of the transition method, experiments are organized as follows.

3.1 Transition between two adjacent curves

A program with two move commands is tested. Each command may be a line or a circle. The first command is set to blending mode with an appropriate TP. In order to guarantee G2 continuity and minimize the amount of calculation, a cubic Bezier curve is used for transition between two lines and a quartic Bezier curve is used for transition involving one or two circles. The sample tests are shown in Figs12 ~ 14. The TP parameters are all set to 5. The transition curve with a smaller TP stays closer to the original trajectory which leads to a smaller transition error and a limited smoothness, and vice versa. The method shows good adaptability no matter where the end point of the second command is set.

3.2 Velocity and acceleration analysis

Multiple lines and circles are tested in this experiment, as shown in Fig. 15. A transition curve is inserted between each pair of adjacent curves. The velocity and acceleration of the trajectory are shown in Fig. 16.

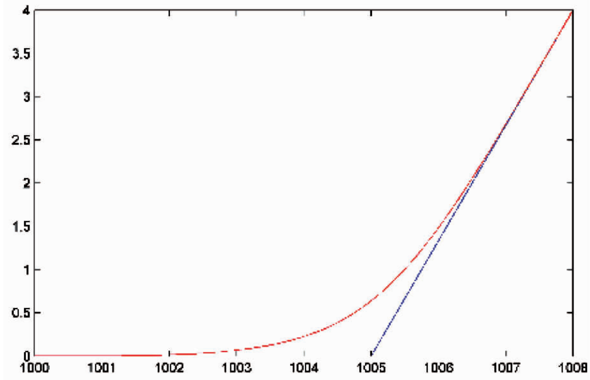

Fig. 12 The transition between two lines (cubic Bezier curve)

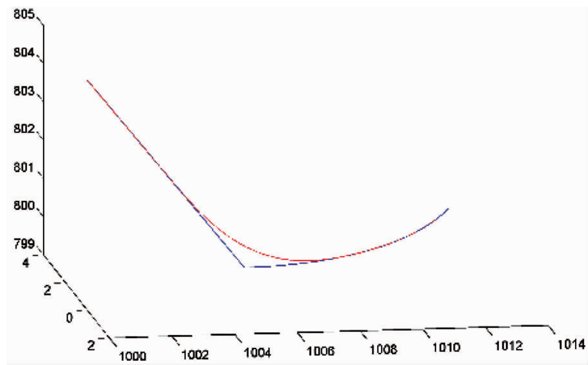

Fig. 13 The transition between a line and a circle (quartic Bezier curve)

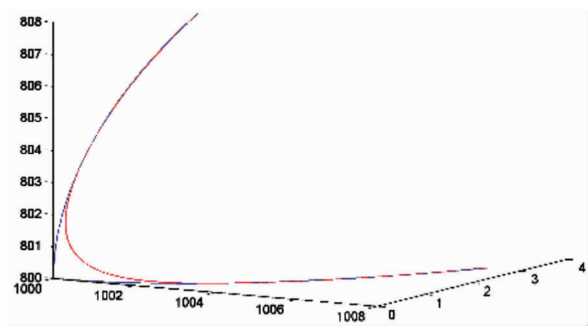

Fig. 14 The transition between two circles (quartic Bezier curve)

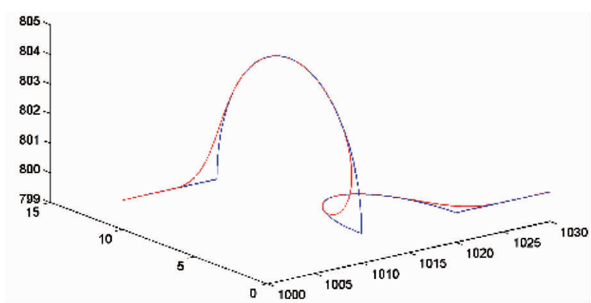

Fig. 15 The transition between multiple curves

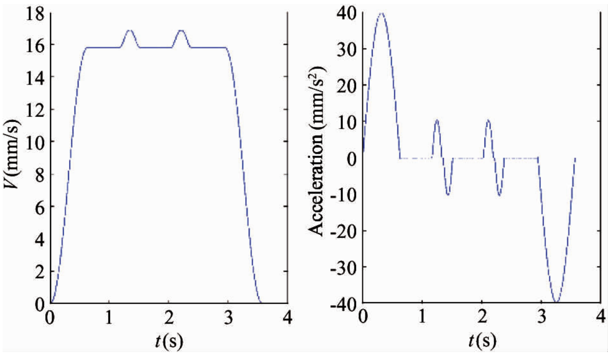

Fig. 16 The velocity and acceleration of the trajectory with transition

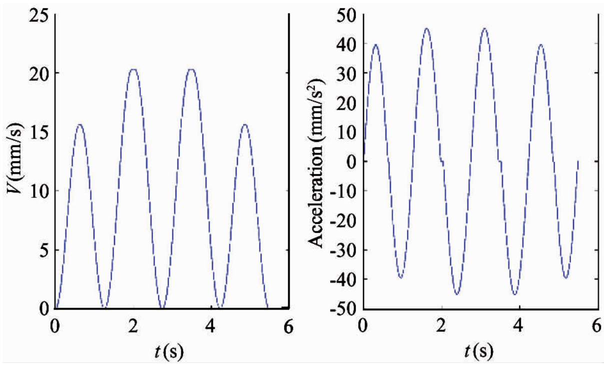

Fig. 17 The velocity and acceleration of the trajectory without transition

S-curve-type acceleration profile is adopted for the velocity and acceleration planning. The maximum velocity for each curve is 50 mm/s, max acceleration is 100 mm/s^2 and maximum jerk is set to 200 mm/s^3. The whole trajectory takes about 3.57s. The velocity and acceleration of the original trajectory without transition are shown in Fig. 17. It is tested with the same velocity, acceleration and jerk constraints, and takes about 5.48s. Obviously, the velocity of the trajectory with transition is smoother and takes less time.

3.3 Velocities in the joint space

Since the trajectory of a robot is finally realized in the joint space, the position and velocity of each joint are tested. When the robot moves along the trajectory shown in Fig. 15, the position of each joint can be got by inverse kinematics, and the velocity is the differential of position. Figs18 ~ 19 show the joint angle and velocity when transition mode is set to blending, and Figs20 ~ 21 show those without transition. The same result can be got that the trajectory with transition moves smoother and takes less time.

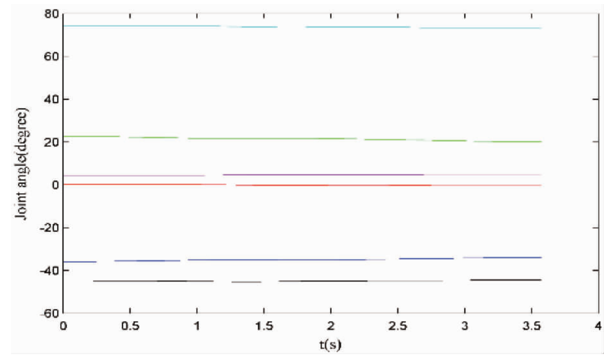

Fig. 18 The joint angle corresponding to the trajectory with transition in Fig. 15 (axis 1; axis 2; axis 3; axis 4; axis 5; axis 6)

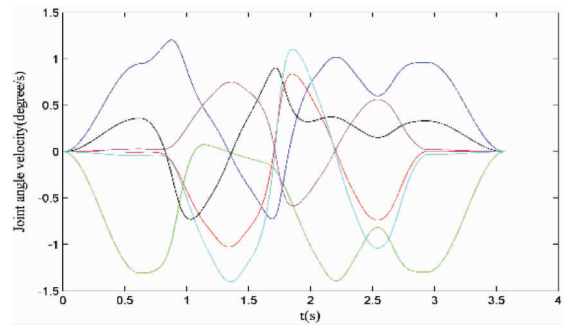

Fig. 19 The joint angle velocity corresponding to the trajectory with transition in Fig. 15 (axis 1; axis 2; axis 3; axis 4; axis 5; axis 6)

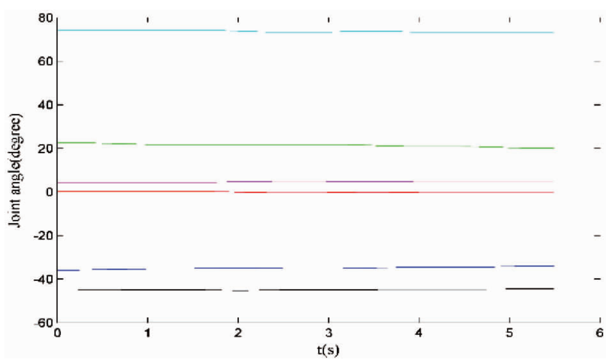

Fig. 20 The joint angle corresponding to the trajectory without transition in Fig. 15 (axis 1; axis 2; axis 3; axis 4; axis 5; axis 6)

4 Conclusions

It has been demonstrated that a single Bezier curve can be utilized to transit between lines and circles in three-dimensional space. In the transition between two lines, a cubic Bezier curve could satisfy the G2 continuity. In the transition between a line and a circle, if the line is coplanar with the circle, a cubic Bezier curve is able to transit smoothly. Otherwise, a quartic Bezier curve is needed. A curvature constraint

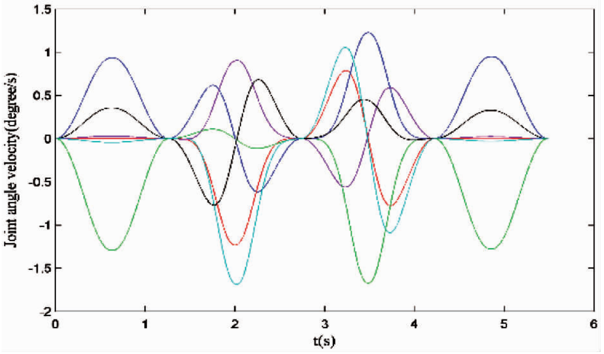

Fig. 21 The joint angle velocity corresponding to the trajectory without transition in Fig. 15 (axis 1; axis 2; axis 3; axis 4; axis 5; axis 6)

is added to obtain the optimal solution in this case. In the transition between two circles, a cubic curve is hard to get the analytical solutions. So, a quartic Bezier curve is used instead. All the three situations guarantee G2 continuity with transition curve adjustable. The amount of calculation is taken into consideration in the algorithms. This method is applicable to different situations of transition between lines and circles. The velocity and acceleration of the trajectory with transition is smoother and takes less time. The same result can be got from the experiments of joint angle and joint angle velocity. Finally, future work will take more factors into consideration to get the optimal solutions for the three transition cases.

References

[1] PLCopen T C. Safety Software Technical Specification, Version 2.0, Part 1: Basics and Part 2-Extensions. PLCopen, Germany. 2011

[2] POLCopen. PLCopen Motion Control. http://www.plcopen.org/pages/tc2_motion_control: PLCopen, 2011

[3] Barsky B A, Derose T D. Geometric continuity of parametric curves: three equivalent characterizations. *IEEE Computer Graphics & Applications*, 1989, 9(6):60-69

[4] Sencer B, Ishizaki K, Shamoto E. A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths. *International Journal of Advanced Manufacturing Technology*, 2014, 76(9):1977-1992

[5] Bi Q, Wang Y, Zhu L, et al. A Practical continuous-curvature Bézier transition algorithm for high-speed machining of linear tool path. In: Proceedings of the 2011 International Conference on Intelligent Robotics and Applications, Aachen, Germany, 2011. 465-476

[6] Zhao H, Zhu L M, Ding H. A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments. *International Journal of Machine Tools & Manufacture*, 2013, 65(2):88-98

[7] Hota S, Ghose D. Optimal transition trajectory for waypoint following. In: Proceedings of the 2013 IEEE International Conference on Control Applications, Hyderabad, India, 2013. 1030-1035

[8] Fan W, Lee C H, Chen J H. A realtime curvature-smooth interpolation scheme and motion planning for CNC machining of short line segments. *International Journal of Machine Tools & Manufacture*, 2015, 96:27-46

[9] Bi Q Z, Shi J, Wang Y H, et al. Analytical curvature-continuous dual-Bézier corner transition for five-axis linear tool path. *International Journal of Machine Tools & Manufacture*, 2015, 130: 96-108

[10] Ziatdinov R, Yoshida N, Kim T W. Fitting G2 multispiral transition curve joining two straight lines. *Computer-Aided Design*, 2012, 44(6):591-596

[11] Habib Z, Sakai M. G2, cubic transition between two circles with shape control. *Journal of Computational & Applied Mathematics*, 2009, 223(1):133-144

[12] Ahmad A, Gobithasan R, Ali J M. G2 Transition curve using Quartic Bézier curve. In: Proceedings of the 2007 International Conference on Computer Graphics, Imaging and Visualization, Bangkok, Thailand, 2007. 223-228

[13] Rashid A, Habib Z. Gear tooth designing with cubic Bézier transition curve. In: Proceedings of the Graduate Colloquium on Computer Sciences, Lahore, Pakistan, 2010. 4:17

[14] Cai H H, Wang G J. A new method in highway route design: joining circular arcs by a single C-Bézier curve with shape parameter. *Journal of Zhejiang University-Science A: Applied Physics & Engineering*, 2009, 10(4):562-569

[15] Barr A H, Currin B, Gabriel S, et al. Smooth interpolation of orientations with angular velocity constraints using quaternions. *Acm Siggraph Computer Graphics*, 1997, 26(2):873-877

Zhang Shaolin, born in 1988. He is a Ph. D student in Institute of Automation, Chinese Academy of Sciences. He received his B. S and M. S degrees in Huazhong University of Science and Technology in 2010 and 2013 respectively. His research interests include the intelligent robotics and motion control technology.