Int J Adv Manuf Technol (2017) 93:4089–4098
DOI 10.1007/s00170-017-0849-1

ORIGINAL ARTICLE

Sharp corner transitional trajectory planning based on arc splines in glass edge grinding

Kun Ren[1] · **Kai Xu**[1] · **Wenhua Chen**[1] · **Jun Pan**[1] · **Bin Yao**[1]

Received: 13 March 2017 / Accepted: 20 July 2017 / Published online: 1 August 2017

Abstract The sharp corners on final contours in flat glass grinding can cause broken wheel center trajectories. A corner transitional trajectory planning algorithm based on arc splines is proposed in this paper to control the wheel center trajectory G^1 continuity and the grinding depth at a corner. The radii of arc splines are derived first to meet design requirements based on the geometrical features of a sharp corner. A three-phase feedrate planning scheme is introduced to generate the arc splines. Corner classification and treatment are presented to make the proposed algorithm flexible, and steps for algorithm application are summarized. Experimental results demonstrate that the proposed algorithm can achieve satisfactory shape accuracy at corners without any breakage, rounding, or flattening.

Keywords Flat glass · Edge grinding · Sharp corner transition · Arc splines · Grinding depth control

1 Introduction

Flat glass is widely used in the building, automobile, and household furniture industries. The demand for personalized glass products with sharp corners is increasing significantly because of their unique appearance. A piece of regular glass is usually cut first to obtain a specific shape but with sharp edges and inaccurate contours. Grinding with diamond wheels is a necessary procedure to remove redundant material along edges and create a final shape that qualifies for assembly.

As shown in Fig. 1, the blue area denotes the glass with a continuous final contour and the red is the redundant material to be removed. The final contour is radius compensated to obtain the wheel center trajectory shown with the black dashed lines. Obviously, at the sharp corners, the wheel center trajectory is broken and transitional trajectories are needed for wheel movement.

"Corner rounding" and "velocity blending" are two main techniques for sharp corner machining of metallic materials. Corner rounding usually uses various parametric curves to correct sharp corners and impose G^1, G^2, or higher-order continuity on the tool path. Sencer and Shamoto [1] and Pateloup et al. [2] use B-spline curves to interpolate discrete segments and thereby eliminate abnormal motion vibrations induced by sharp corners. Combined with acceleration/deceleration (acc/dec) control algorithms, analytical contour error control equations were presented to realize high-precision corner machining. Polynomial curves were adopted to smooth sharp corners [3–6]. Considering the contour error limitations and capacities of servo system, the velocities along the tool path were optimized. Bezier curves were also used to smooth a discrete tool path composed of linear and circular segments [7–9]. Corner velocities were optimized with curvature constraints and acc/dec algorithms. Nonuniform rational basis spline (NURBS) curves were usually proposed to interpolate tool paths which were corrected to higher-order continuity to avoid harmful mechanical vibrations at sharp corners [10–14]. In addition, Jahanpour and Imani [15, 16] and Farouki and Nittler [17] used Pythagorean hodograph (P-H) curves to blend sharp corner trajectories and achieved constant feedrate machining at corners.

Velocity blending [18] is another method of sharp corner machining. Tajima and Sencer [19] and Jahanpour and

✉ Kun Ren
renkun_2008@sina.com

[1] College of Mechanical Engineering and Automation, Zhejiang Sci-Tech University, Hangzhou 30018, China

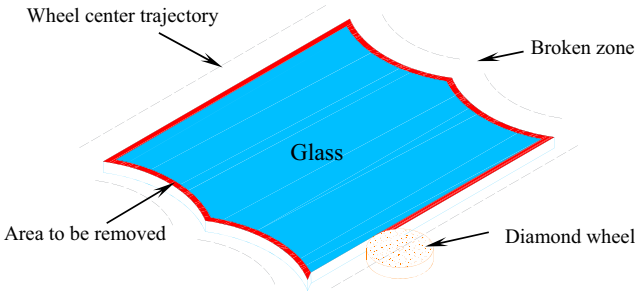


Fig. 1 Glass edge grinding

Alizadeh [20] adopted S-type acc/dec algorithms and calculated optimum velocities at sharp corners taking into account the constraints, such as jerk limitation and servo capacities. Luo et al. [21] and Li et al. [22] proposed velocity look-ahead algorithms under constraints, such as maximum velocity and corner geometrical angle.

Both "corner rounding" and "velocity blending" can guarantee that the tool traverses a sharp corner smoothly, but cannot be directly used in glass edge grinding for the following reasons:

1. There is always more redundant material at a sharp corner than normal. The abrupt increase in grinding depth easily results in glass breakage, rounding, or flattening. How to properly control the grinding depth at a corner has rarely been considered.
2. There is neither an interpolating point in broken zones for curve fit nor enough acc/dec distance for velocity blending.

To realize sharp corner grinding, a transitional trajectory planning algorithm based on arc splines is proposed in this article. The article is structured as follows: Section 2 describes the transitional trajectory design. Section 3 presents the classification and treatment of corners and summarizes the application steps. In Section 4, a typical contour is ground to test the performance of the proposed algorithm. Section 5 presents our conclusions.

2 Trajectory design

2.1 Transitional process planning

As shown in Fig. 2a, the black dashed lines denote the broken wheel center trajectory, and the blue solid line is the desired transitional trajectory, which is composed of four circular arcs denoted as *AB*, *BC*, *CD*, and *DE*, respectively. When the wheel center moves from point *A* to point *E* along the arcs, the process is scheduled as follows:

1. Circular arc *AB* is tangent to the wheel center trajectory at point *A*. From point *A* to point *B*, the grinding depth decreases, gradually.
2. Circular arc *AB* is tangent to the *BC* at point *B*. As shown in Fig. 2b, from point *B* to point *C*, the grinding depth continues to decrease, and to be zero at point *C*.
3. Circular arc *BC* is tangent to the *CD* at point *C*. From point *C* to point *D*, the grinding depth tends to increase, gradually.
4. Circular arc *CD* is tangent to the *DE* at point *D*. As shown in Fig. 2c, from point *D* to point *E*, the grinding depth continues to increase and returns to normal at point *E*, finally.
5. Circular arc *DE* is tangent to the wheel center trajectory at point *E*.

Desired arc splines guarantee that the connected tool path is of G^1 continuity, the grinding depth is controllable, and the direction of the wheel center movement is always tangent to the final contour at points *A* and *E*.

2.2 Sharp corner geometrical analysis

As shown in Fig. 3, a typical corner is formed by two straight lines. *Q* is the corner point which is radius compensated to yield points *E* and *A*. The broken wheel center trajectories are extended, and intersect at point ***P***. $\overline{QP}$ is the bisector of corner angle 2φ, $0 \le \varphi \le 90^{\circ}$.

Let d_c be the scheduled grinding depth, and R_T be the wheel radius. According to the tool radius compensation algorithm, we have

$$\|QA\| = \|QE\| = R_T. \tag{1}$$

Obviously, around the corner *Q*, the maximum grinding depth *d* is

$$d = d_c \Big/ \sin\varphi. \tag{2}$$

$\overline{MN}$ is the common tangent of circular arcs *BC* and *CD*, and perpendicular to the bisector $\overline{QP}$. To guarantee that the grinding depth becomes zero at point ***C***, the length of $\overline{QC}$ should be

$$\|QC\| = R_T + d. \tag{3}$$

Moreover, circular arcs *AB* and *DE*, *BC* and *CD* are both symmetrical with respect to bisector $\overline{QP}$.

Points O_1 and O_2 are the center of arcs *AB* and *BC*, respectively. Obviously, points O_1, *Q*, and *A* are collinear, points O_1, O_2, and *B* are collinear, and points *Q*, O_2, and *C* are collinear. Let angle $\angle BO_1A$ be θ_1 and $\angle BO_2A$ be θ_2. $\overline{GF}$ is the common

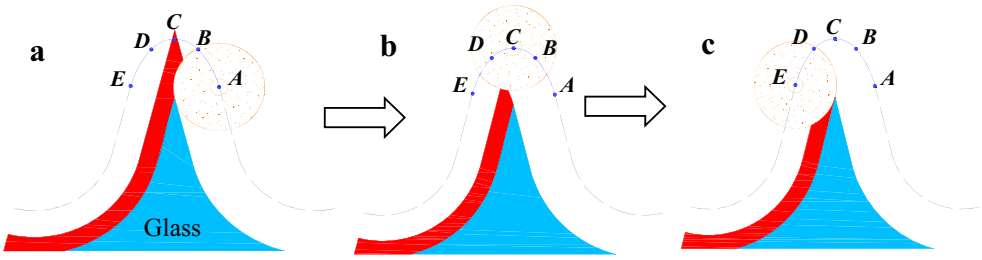


Fig. 2 Transition along desired arc splines: (*a*) transition begins, (*b*) grinding depth becomes zero, and (*c*) transition ends

tangent of circular arcs AB and BC, and perpendicular to the linear segment $\overline{O_1B}$. We have

$$\theta_1 + \theta_2 + \varphi = \frac{\pi}{2}. \tag{4}$$

2.3 Radii of arc splines

A local Cartesian coordinate system X_1QY_1 is established. Let corner point Q be the origin, R be the radius of circular arcs AB and DE, and r be the radius of circular arcs BC and CD. We have

$$\boldsymbol{O_1} = (R_T - R)\boldsymbol{e_s}, \tag{5}$$

and

$$\boldsymbol{O_1A} = R\boldsymbol{e_s}, \tag{6}$$

where the unit vector $\boldsymbol{e_s} = \frac{\boldsymbol{A}}{\|\boldsymbol{A}\|}$.

Let $\boldsymbol{O_1A}$ rotate θ_1 degrees counterclockwise to yield

$$\boldsymbol{O_1B} = R\boldsymbol{e_s}\begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix}, \tag{7}$$

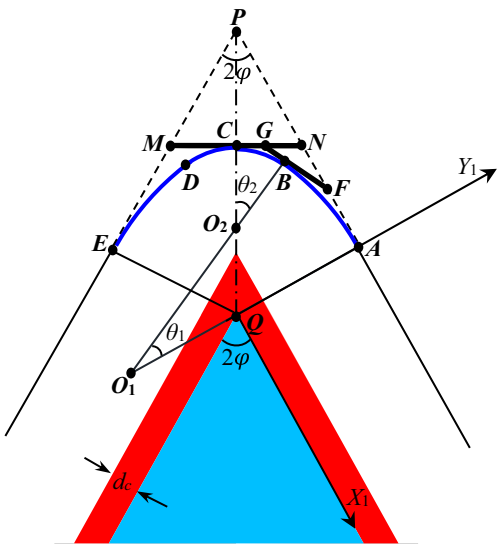


Fig. 3 A corner with desired arc splines transition

and

$$\boldsymbol{O_2B} = r\boldsymbol{e_s}\begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix}. \tag{8}$$

Let $\boldsymbol{O_1A}$ rotate $(\theta_1 + \theta_2)$ degrees counterclockwise to yield

$$\boldsymbol{O_2} = \begin{bmatrix} (r - R_T - R)\cos\varphi \\ (R_T + R - r)\sin\varphi \end{bmatrix}. \tag{9}$$

With the point O_1 and vector $\boldsymbol{O_1B}$, point $\boldsymbol{B}$ can be derived as

$$\boldsymbol{B} = \begin{bmatrix} -R\sin\theta_1 \\ R_T + R\cos\theta_1 - R \end{bmatrix}. \tag{10}$$

With the point O_2 and vector $\boldsymbol{O_2B}$, point $\boldsymbol{B}$ can be also derived as

$$\boldsymbol{B}' = \begin{bmatrix} (r - R_T - d)\cos\varphi - r\sin\theta_1 \\ r\cos\theta_1 - (r - R_T - d)\sin\varphi \end{bmatrix}. \tag{11}$$

Hence, we have

$$\boldsymbol{B} = \boldsymbol{B}'. \tag{12}$$

With Eq. ((12), the radii R and r can be derived as

$$R = \frac{R_T(\sin\theta_1 - \cos\varphi) + (R_T + d)\cos(\theta_1 + \varphi)}{\sin\theta_1 - \cos\varphi + \cos(\theta_1 + \varphi)}, \tag{13}$$

and

$$r = \frac{R\sin\theta_1}{\sin\theta_1 - \cos\varphi} - \frac{(R_T + d)\cos(\varphi)}{\sin\theta_1 - \cos\varphi}. \tag{14}$$

θ_1 is yet unknown and will be figured out in the next sections, but the relationships between θ_1 and the radii can be checked first. Typically, let half angle $\varphi = 15°$, wheel radius $R_T = 75$ mm, and grinding depth $d_c = 0.13$ mm. As shown in Fig. 4, two rules can be concluded as follows:

Rule 1. Radius R is always greater than the wheel radius R_T. R decreases and approaches R_T indefinitely with the increase of θ_1.

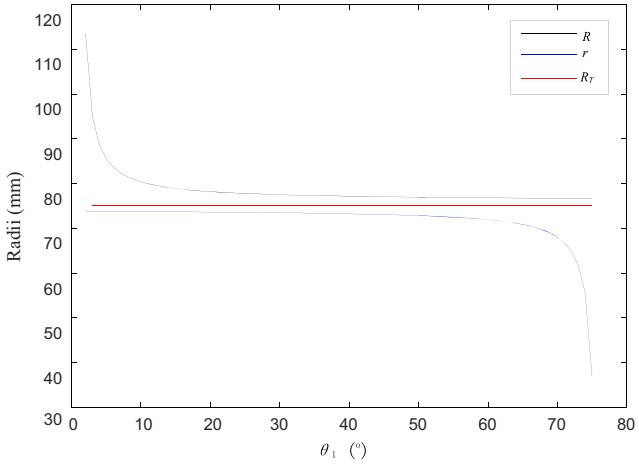


Fig. 4 The relationships of θ_1 with R, r

Rule 2. Radius r is always smaller than the wheel radius R_T. r increases and approaches R_T indefinitely with the decrease of θ_1.

2.4 Feedrate constraint

As shown in Fig. 5, when the wheel center moves along the arc splines from point A to point E, the direction of Y axis always remains unchanged, but the X axis will reverse at point C. From position C, ΔX and ΔY are the increased displacements of axes X and Y after an interpolation cycle, respectively. Two triangles, ΔCC_1C_2 and ΔCO_2C_3, are created with a same inner angle β. And $\|C_1C_2\| = \Delta X$, $\|CC_2\| = \Delta Y$, $\|CO_2\| = r$, $\|CC_3\| \approx \Delta Y/2$, we have

$$\tan(\beta) = \frac{\Delta X}{\Delta Y} \approx \frac{\Delta Y}{2r} \tag{15}$$

where ΔX and ΔY can be derived as

$$\begin{cases} \Delta X = \frac{1}{2} a_x {T_s}^2, & \text{and} \\ \Delta Y = v_{c1} T_s \end{cases}$$

where v_{c1} is a transitional feedrate, a_x is the acceleration value of axis X, and T_s is the interpolation period. With Eq. (15), we have

$$v_{c1} \approx \sqrt{a_x r}. \tag{16}$$

Taking the X and Y axes into consideration, with Eq. (16), the transitional feedrate constraint can be derived as

$$v_{c1} \le \min\left(\sqrt{a_x r}, \sqrt{a_y r}\right) \tag{17}$$

where a_y is the acceleration value of axis Y.

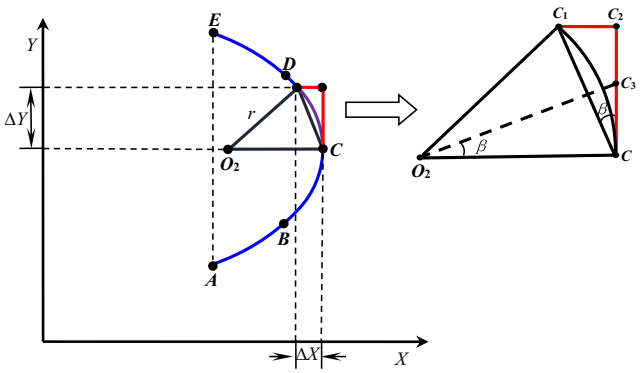


Fig. 5 Corner motion analyses

If the X or Y axis reverses on AB or DE, Eq. (17) should be corrected to

$$v_{c1} \le \min\left(\sqrt{a_x R}, \sqrt{a_y R}\right). \tag{18}$$

According to the rules 1 and 2, the radius R is always greater than the radius r. Therefore, only Eq. (17) can meet the control requirements.

2.5 Generation of arc splines

Motion through a sharp corner requires a dec/acc process. Let v_{max} be the maximum machining feedrate, and v_{trans} be the transitional feedrate. The planned arc splines are divided into three motion phases:

- Circular arc AB is a deceleration portion, where feedrate decreases from v_{max} to v_{trans}.
- Circular arcs BC and CD are both uniform speed portions, at constant feedrate v_{trans}.
- Circular arc DE is an acceleration portion, where feedrate increases from v_{trans} to v_{max}.

The scheduled feedrate profile is shown in Fig. 6.

A linear acc/dec algorithm is used and the required deceleration distance is

$$L_{AB} = \frac{{v_{max}}^2 - {v_{c2}}^2}{2\sqrt{{a_x}^2 + {a_y}^2}} \tag{19}$$

where L_{AB} is the required deceleration distance, and v_{c2} is a transitional feedrate that can be initially set by

$$v_{c2} = \frac{\varphi}{90} v_{max}. \tag{20}$$

According to feedrate planning, L_{AB} is equal to the length of arc AB. Hence, the value of θ_1 can be derived as

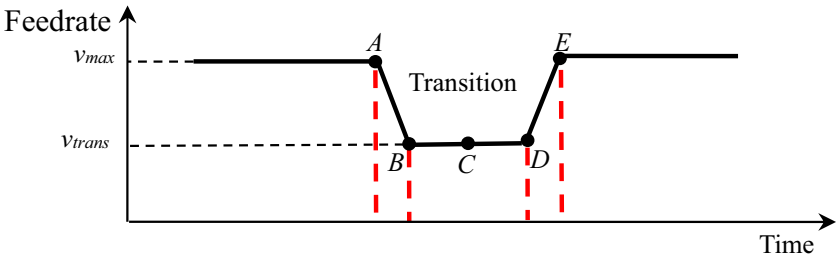


Fig. 6 Scheduled feedrate profile for corner transition

$$\theta_1 = \frac{L_{AB}}{R} = \frac{1-\left(\frac{\varphi}{90}\right)^2}{2R\sqrt{{a_x}^2 + {a_y}^2}} {v_{\max}}^2. \tag{21}$$

Because rule 1 says that radius R is always greater than the tool radius R_T, we have

$$\theta_1 R > \theta_1 R_T. \tag{22}$$

To guarantee enough deceleration distance, wheel radius R_T can be taken into Eq. (21) as a substitute to solve angle θ_1. Taking the obtained angle θ_1 into Eqs. (13) and (14), radii R and r are both determined. Then, with Eqs. (17) and (20), transitional feedrate v_{trans} can be confirmed by

$$v_{trans} = \min(v_{c1}, v_{c2}). \tag{23}$$

After confirmations of θ_1, R, and r, the circular arc AB can be derived as

$$c_{AB}(\theta) = \begin{bmatrix} -R\sin\theta \\ R\cos\theta - R + R_T \end{bmatrix}^T \tag{24}$$

where $0^0 \le \theta \le \theta_1$.

The circular arc BC is derived as

$$c_{BC}(\theta) = \begin{bmatrix} -r\sin(\theta_1 + \theta) - (R_T + d - r)\cos\varphi \\ r\cos(\theta_1 + \theta) + (R_T + d - r)\sin\varphi \end{bmatrix}^T \tag{25}$$

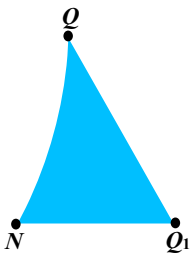
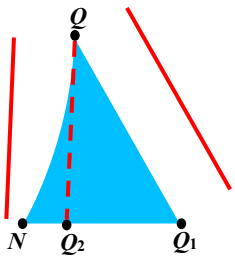
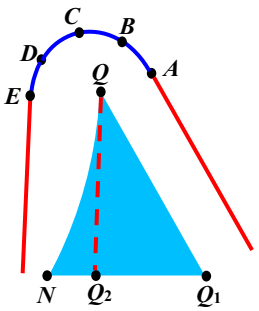
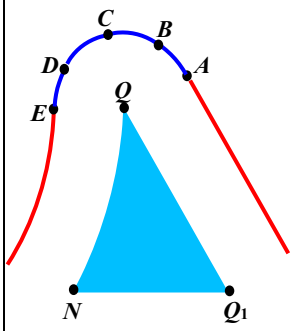
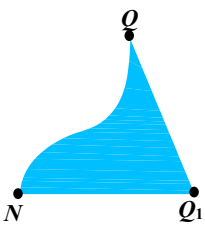
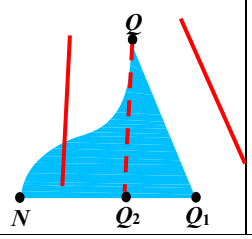
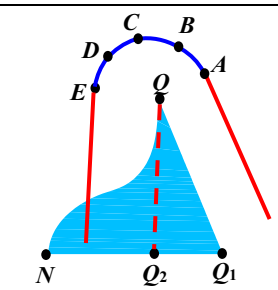
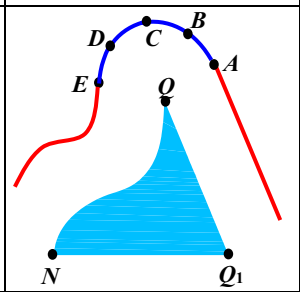
where $0^0 \le \theta \le \theta_2$.

For symmetry, arcs AB and BC are mirrored relative to bisector $\overline{QP}$, and the circular arc CD is expressed as

$$c_{CD}(\theta) = c_{BC}(\theta) \begin{bmatrix} \cos^2\varphi - \sin^2\varphi & -\sin 2\varphi \\ -\sin 2\varphi & \sin^2\varphi - \cos^2\varphi \end{bmatrix} \tag{26}$$

where $0^0 \le \theta \le \theta_2$.

Table 1 Corner types and treatments

Type	Sharp corner	Radius compensated	Arcs generation	Final result
Arc				
Curve				

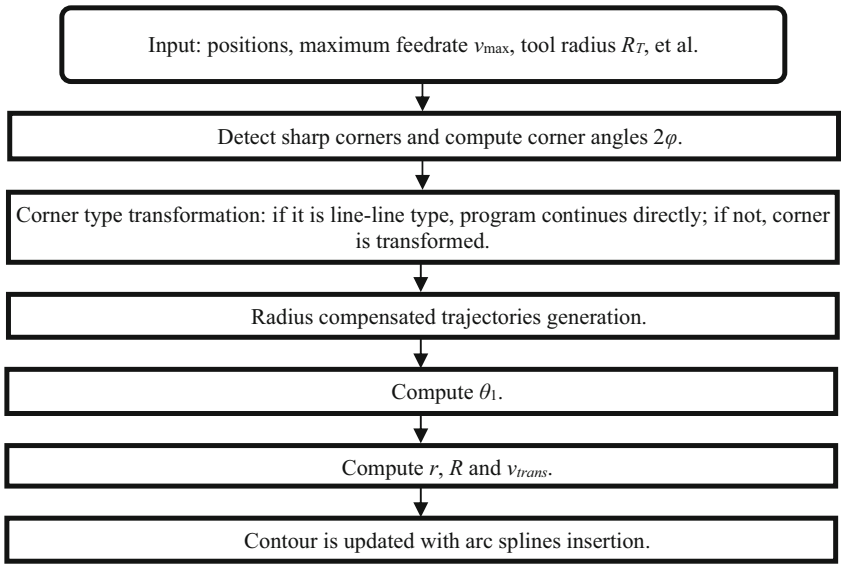


Fig. 7 Flow chart of the proposed algorithm

The circular arc DE is expressed as

$$c_{DE}(\theta) = c_{AB}(\theta)\begin{bmatrix} \cos^2\varphi - \sin^2\varphi & -\sin 2\varphi \\ -\sin 2\varphi & \sin^2\varphi - \cos^2\varphi \end{bmatrix} \quad (27)$$

where $0^0 \le \theta \le \theta_1$.

3 Application of the proposed algorithm

3.1 Corner classification and treatment

To make the wheel center trajectory continuous, generated arc splines should be properly inserted into the broken zones shown in Fig. 1. A sharp corner formed by two straight lines can be defined as a line-line type and was used as an example in Section 2 to show how arc splines are generated and inserted, but a circular arc or parametric curve is also usually used to form a corner.

As shown in the second row of Table 1, the corner is formed by a straight line $\overline{QQ_1}$ and a circular arc QN of which Q is the starting point and N is the end point. In the last row of Table 1, the corner is formed by a straight line $\overline{QQ_1}$ and a curve QN of which Q is the starting point and N is the end point. To use the proposed algorithm, an auxiliary linear segment (red dashed line) $\overline{QQ_2}$ is created and tangent to the arc or curve at point Q.

Suppose that the circular arc center is Q_c, we have

$$Q_2 = Q + M_c \frac{(Q - Q_c)^T}{\|Q - Q_c\|} L_e \quad (28)$$

where L_e is the length of the auxiliary line. If ON is clockwise, $M_c = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$; otherwise, $M_c = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

Suppose that curve $\tilde{Q}N$ is defined as $c(u)$, $0 \le u \le 1$, we have

$$Q_2 = Q - \frac{L_e}{\|c'(0)\|} c'(0) \quad (29)$$

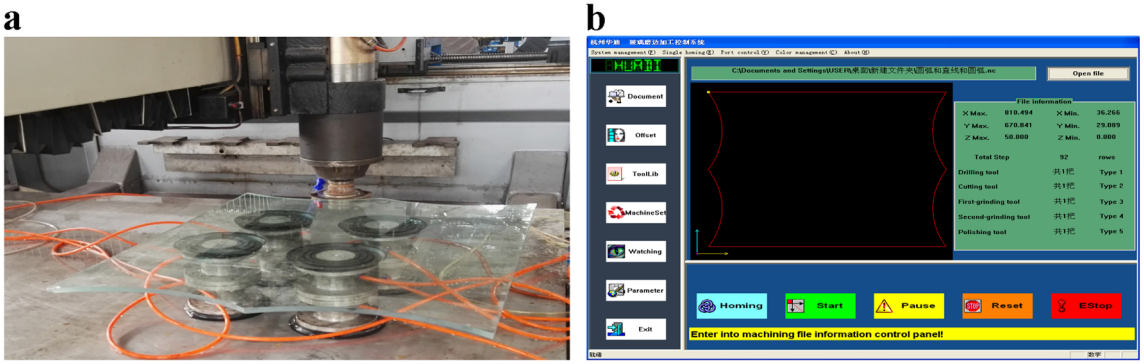


Fig. 8 Experimental setup. **a** Glass grinding center. **b** Man-machine interface

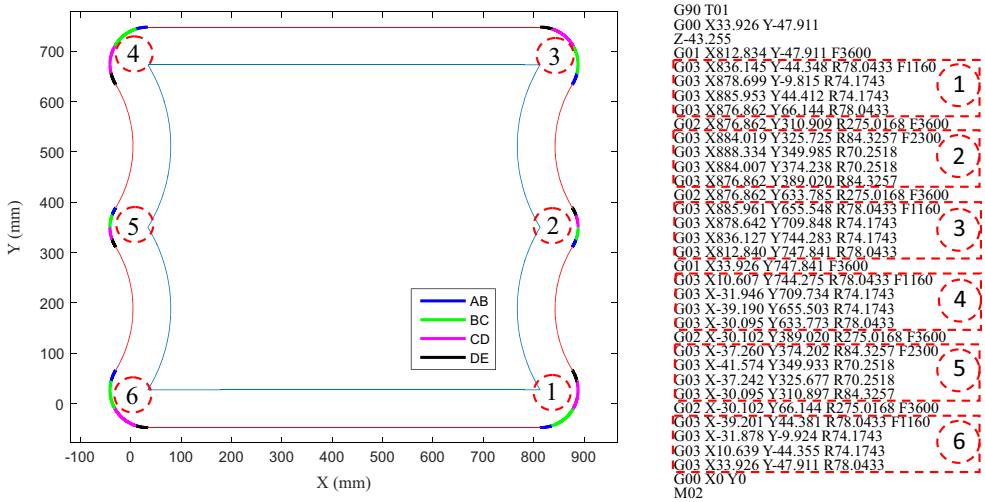


Fig. 9 Compensated trajectories with transitional arc splines

Then, with the straight lines $\overline{QQ_2}$ and $\overline{QQ_1}$, the corners of other type are transformed into the line-line type, and the proposed algorithm can be used directly to generate transitional arc splines. The final results are shown in the last column of Table 1.

3.2 Application steps

The transformation makes the proposed algorithm suitable for various corners. The steps for application are summarized in Fig. 7 and depicted briefly as follows:

Step 1

Data input Contour is represented by G-codes where the data types, positions, maximum velocities, and other useful information are stored. The necessary information should be extracted and stored first in a struct defined as follows:

typedef struct{
double xpositon;
double ypositon;
double velocity;
int type;
……
}INPUTPOINT;

Then, every *INPUTPOINT* struct object is added to an array, which is defined as follows:

typedef CArray<INPUTPOINT, INPUTPOINT> CInputPointArray;

Other necessary machining parameters, such as tool radius R_T, normal grinding depth d_c, and spindle speed S, should also be input before machining.

Step 2

Contour update The data segments in the array *CInputPointArray* are checked one by one to detect sharp corners and compute corner angles. According to the detected

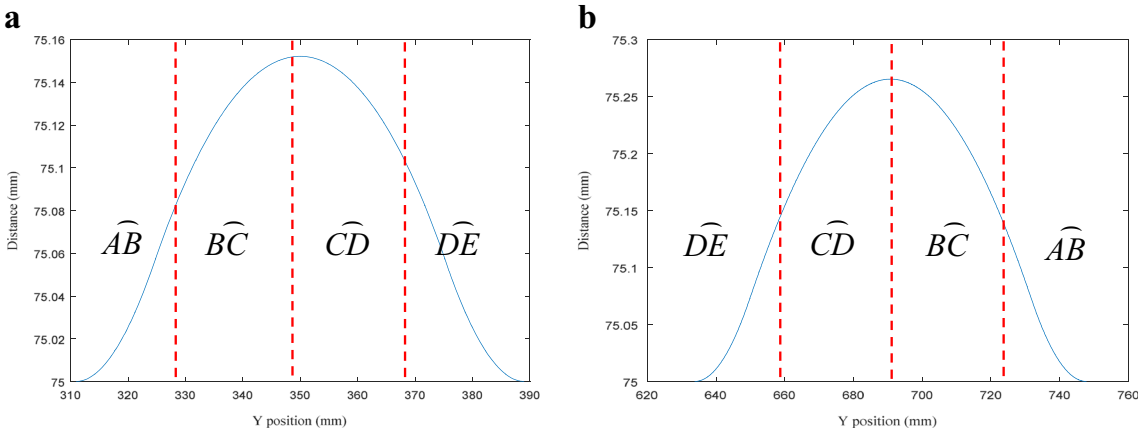
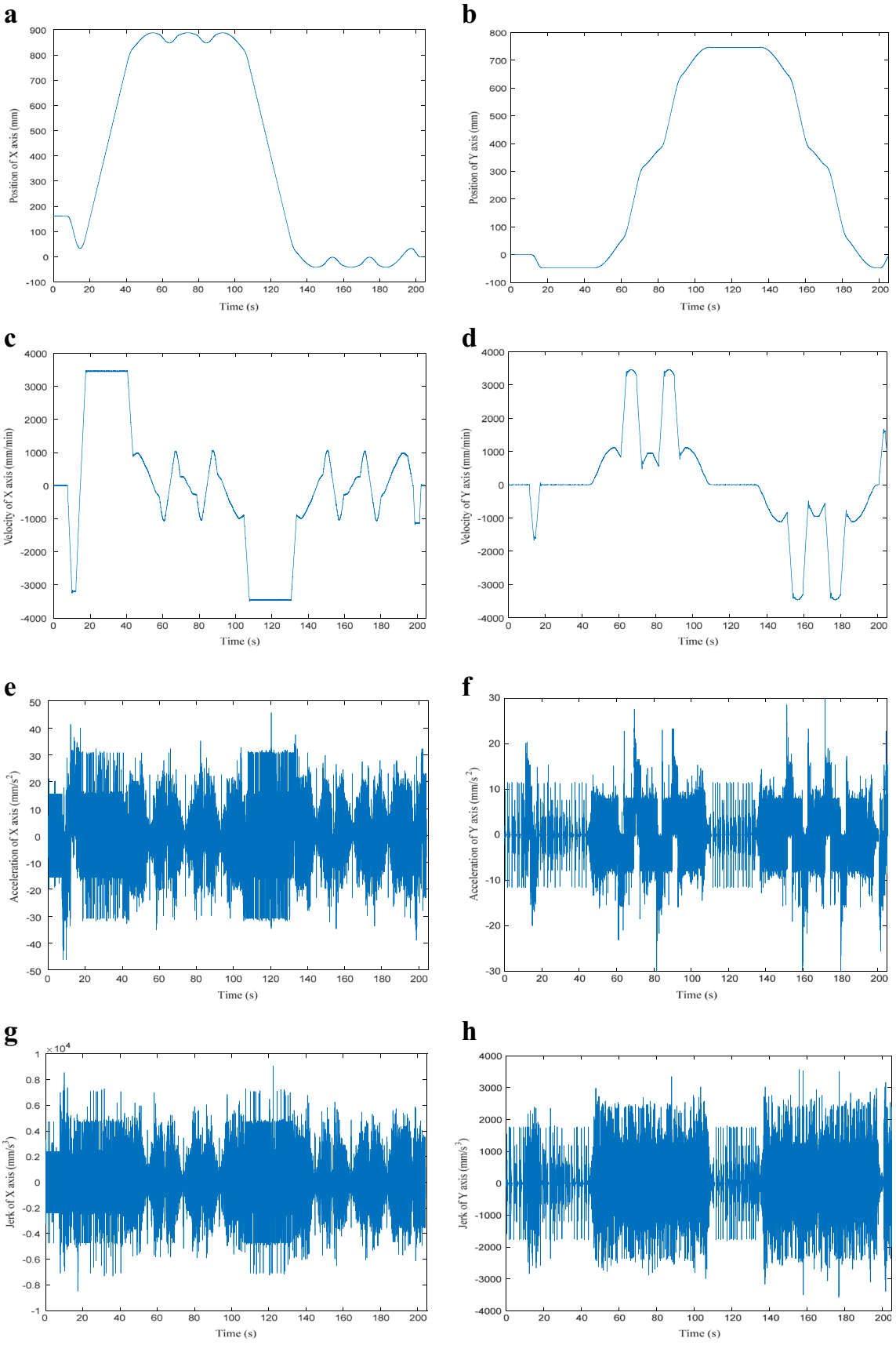


Fig. 10 The distance change profiles between the wheel center and corner points when diamond wheel traverses corners 2 and 4. **a** Corner 2. **b** Corner 4


a
Position of X axis (mm)
Time (s)
b
Position of Y axis (mm)
Time (s)
c
Velocity of X axis (mm/min)
Time (s)
d
Velocity of Y axis (mm/min)
Time (s)
e
Acceleration of X axis (mm/s²)
Time (s)
f
Acceleration of Y axis (mm/s²)
Time (s)
g
×10⁴
Jerk of X axis (mm/s³)
Time (s)
h
Jerk of Y axis (mm/s³)
Time (s)

◀ **Fig. 11** The kinematic profiles. **a** Displacement profile of the *X* axis. **b** Displacement profile of the *Y* axis. **c** Velocity profile of the *X* axis. **d** Velocity profile of the *Y* axis. **e** Acceleration profile of the *X* axis. **f** Acceleration profile of the *Y* axis. **g** Jerk profile of the *X* axis. **h** Jerk profile of the *Y* axis

corner type, all corners are transformed into line-line corner type. After the contour is radius compensated and broken zones are confirmed, the contour is updated with arc spline insertions.

4 Experimental results

The experiment is carried out on a three-axis NC glass grinding center shown in Fig. 8a. The *X*, *Y*, and *Z* axes travel together to span a 1900 mm × 3100 mm × 240 mm 3-D space. The experimental parameters are set as follows: spindle speed $S = 6000$ r/min, acceleration values of the *X* and *Y* axes $a_x = -a_y = 50$ mm/s^2, glass thickness $T_g = 12$ mm, maximum feedrate $v_{max} = 3600$ mm/min, wheel radius $R_T = 75$ mm, and normal grinding depth $d_c = 0.13$ mm.

As shown in Fig. 8b, the man-machine interface is coded with Visual C++ 6.0 and integrated with the proposed algorithm. When required trajectories (represented by G-codes) are loaded into the PC, the steps shown in Fig. 7 are implemented to update the contour which will be downloaded to the motion controller to achieve processing.

As shown in Fig. 9, a piece of curtain wall glass with six sharp corners is ground. The original contour is shown with blue lines, and the radius-compensated trajectories are shown with red lines. Inserted transitional arc splines at each corner are shown with different colors. The machining G-codes with inserted arc splines (enclosed in the red dashed rectangles) are also listed on the right of Fig. 9.

The corners are numbered sequentially and calculation results are expressed as follows:

1. Half angle $\varphi = 29.3°$ at corners 1, 3, 4, and 6; $\varphi = 58.6°$ at corners 2 and 5.
2. Let $L_e = 10$ mm, and corners of other type are all transformed into line-line type with Eq. (28).
3. With Eq. (21), $\theta_1 = 17.39°$ at corners 1, 3, 4, and 6; $\theta_1 = 11.20°$ at corners 2 and 5.
4. With Eqs. (13), (14), and (23), $R = 76.616$ mm, $r = 74.562$ mm, and $v_{trans} = 1160$ mm/min at corners 1, 3, 4, and 6; $R = 77.841$ mm, $r = 73.553$ mm, and $v_{trans} = 2300$ mm/min at corners 2 and 5.
5. With Eqs. (24), (25), (26), and (27), transitional arc splines for each corner are generated.

Figure 10a, b shows the distance change profiles between the wheel center and corner points when the diamond wheel traverses corners 2 and 4. When the wheel center moves along arcs *AB* and *BC* at the corners, the increasing distance indicates the gradually decreasing grinding depth. When the wheel center moves along arcs *CD* and *DE*, the decreasing distance indicates the gradually increasing grinding depth. The distance change profiles at two different corners are both in accordance with the design requirements.

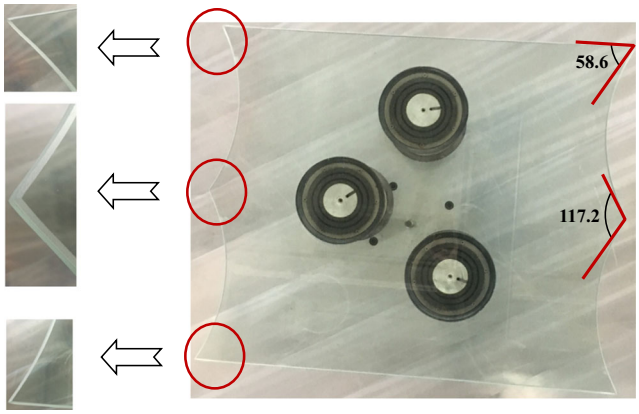


Fig. 12 Final ground glass

Figure 11 shows the kinematic profiles of the machining process. When wheel traverses a sharp corner with the proposed algorithm, there is no an abnormal jerk value, and the wheel movement is smooth without any harmful mechanical vibration.

Figure 12 shows the final ground glass. The proposed algorithm can achieve satisfactory shape accuracy without any corner breakage, rounding, or flattening. The corner angle accuracy of ±0.1° fully meets the factory requirements.

5 Conclusions

A transitional trajectory planning algorithm based on arc splines is proposed in this article. The algorithm takes grinding depth into consideration when the wheel traverses a sharp corner with arc splines. A four-portion trajectory scheduling algorithm with three-phase acc/dec arrangements is adopted to grind sharp glass corners.

The techniques now available cannot be used directly in glass edge grinding. Compared with previous related works, the proposed algorithm has the following advantages:

1. Taking into consideration the grinding depth control at a sharp corner, a four-arc trajectory is designed to meet the transitional requirements.
2. Four arcs are tangent to each other and guarantee the whole trajectory G^1 continuity. Meanwhile, a transitional feedrate constraint is introduced to guarantee smooth motion. Arcs can be easily implemented by the control system unlike parametric curves.

3. Corner types are analyzed and a transformation algorithm makes the proposed algorithm flexible and suitable for machining various glass corners.

The experimental results show that the transition scheme proposed in this paper can achieve satisfactory shape accuracy without any corner breakage, rounding, or flattening.

Acknowledgements This study is funded by the National Natural Science Foundation of China (Grant No. 51405445).

References

1. Sencer B, Shamoto E (2014) Curvature-continuous sharp corner smoothing scheme for Cartesian motion systems. 13th IEEE International Workshop on Advanced Motion Control 374-379
2. Pateloup V, Duc E, Ray P (2010) B-spline approximation of circle arc and straight line for pocket machining. Comput Aided Des 42(9):817–827
3. Zhang L, You Y, He J, Yang X (2011) The transition algorithm based on parametric spline curve for high-speed machining of continuous short line segments. Int J Adv Manuf Technol 52(1):245–254
4. Altintas Y, Erkorkmaz K (2003) Feedrate optimization for spline interpolation in high speed machine tools. CIRP ANN-Manuf Techn 52(1):297–302
5. Erkorkmaz K, Altintas Y (2001) High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation. Int J Mach Tools Manuf 41(9):1323–1345
6. Yutkowitz SJ, Chester W (2005) Apparatus and method for smooth cornering in a motion control system. United States patent US 6922606
7. Bi Q, Wang Y, Zhu L, Ding H (2011) A practical continuous-curvature Bézier transition algorithm for high-speed machining of linear toolpath. 4th International Conference on Intelligent Robotics and Applications 465-476
8. Ernesto CA, Farouki RT (2012) High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error. Int J Mach Tools Manuf 58(1):327–338
9. Beudaert X, Lavernhe S, Tournier C (2013) 5-axis local corner rounding of linear tool path discontinuities. Int J Mach Tools Manuf 73(10):9–16
10. Zhao H, Zhu L, Ding H (2013) A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of shortline segments. Int J Mach Tools Manuf 65(2):88–98
11. Pateloup V, Duc E, Ray P (2010) Bspline approximation of circle arc and straight line for pocket machining. Comput Aided Des 42(9):817–827
12. Tulsyan S, Altintas Y (2015) Local toolpath smoothing for five-axis machine tools. Int J Mach Tools Manuf 96:15–26
13. Beudaert X, Lavernhe S, Tournier C (2012) Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path. Int J Mach Tools Manuf 57(3):73–82
14. Yeh SS, Su HC (2009) Implementation of online NURBS curve fitting process on CNC machines. Int J Adv Manuf Technol 40(5): 531–540
15. Jahanpour J, Imani BM (2008) Real-time P-H curve CNC interpolators for high speed cornering. Int J Adv Manuf Technol 39(3): 302–316
16. Imani BM, Jahanpour J (2008) High-speed contouring enhanced with PH curves. Int J Adv Manuf Technol 37(7):747–759
17. Farouki RT, Nittler KM (2016) Efficient high-speed cornering motions based on continuously-variable feedrates. I. Real-time interpolator algorithms. Int J Adv Manuf Technol 87(9):3557–3568
18. Volpe R (1993) Task space velocity blending for real-time trajectory generation. IEEE Int Conf Robot Autom 1993:680–687
19. Tajima S, Sencer B (2016) Kinematic corner smoothing for high speed machine tools. Int J Mach Tools Manuf 108:27–43
20. Jahanpour J, Alizadeh MR (2015) A novel acc-jerk-limited NURBS interpolation enhanced with an optimized S-shaped quintic feedrate scheduling scheme. Int J Adv Manuf Technol 77(9): 1889–1905
21. Luo FY, Zhou YF, Yin J (2007) A universal velocity profile generation approach for high-speed machining of small line segments with look-ahead. Int J Adv Manuf Technol 35(5):505–518
22. Li H, Gao X, Zhang L, Sun R (2012) Discrete interpolation of G01 codes in 2D machining under bounded acceleration. Math Comput Sci 6(3):327–344