

Path and Trajectory Generation for Robotic Polishing of Sheet Metal Parts

Yuezhi (Sean) Liu¹, Zhiyong Chen², Fengfeng (Jeff) Xi^{1 a *} and Kenny Lu¹,

¹Department of Aerospace Engineering, Ryerson University, Toronto, Canada

²School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou, P.R. China

^aemail: fengxi@ryerson.ca

Keywords: Polishing, Sheet Metal Parts, Path Generation, Trajectory Generation.

Abstract. Presented in this paper is a method to generate path and trajectory for robotic polishing of sheet metal parts that are in general large and placed in an open space. The proposed method includes three elements. The first is to generate a tool path either based on a CAD model or a scanned data set. The said tool path should cover the entire polishing area according to the size of the polishing tool and truly guide the tool to follow the surface closely with the tool axis always along the surface normal. The second element is to generate a motion profile on the top of the planned path that can create a smooth motion transition among the turning points following a zigzag path pattern. The third topic is force planning to relate the contact stress with the applied force. Several contact force planning method are explored. The proposed method is demonstrated through an in-house software for polishing of automobile sheet metal parts.

1. Introduction

Robotic polishing is a relatively new research topic. A working method of it is highly desired by the industries due to its low cost and time efficacy. However, the research of robotic polishing has proven to be challenging when it comes to curved part surfaces. There are several difficulties unique to curves part surfaces. First, Unlike the flat or near flat part surfaces, each curved part surface has a unique geometry that need to be analyzed. The raw geometry data is usually obtained with either scanning or CAD model from conceptual design. However, that data cannot be used directly, and a mathematical model need to be built base on the data that describes the part surface.

Second difficulty is the tool path and trajectory planning. While this is easy to do on a flat surface, many strategies for tool path planning such as raster path and spiral path described in previous research [1, 2] are difficult to do on a curved surface. This is because the distance between the waypoints, normal and tangent of each way point along the path, as well as the step-over distance of each path need to be calculated according to the local surface curvature to ensure the tool can follow the surface closely with the tool axis always along the surface normal.

The third difficulty is the changing in contact area between tool and part surface. This is very largely affected by the geometry of the part surface. Because of this, applying constant force can create either unpolished envelops or unnecessary overlapping [6] at certain areas. To achieve a finishing result without any unpolished envelop, a special force planning is needed.

In this paper, a method is proposed in response to the those afore mentioned difficulties.

2. Mathematical Model of the Part Surface

The part surface model can be based on CAD drawing of concept design, or scanned data of a real working parts piece. If the part surface is represented by a CAD model, it is necessary to discretizing the geometry in order to obtain a set of points and their coordinates. The discretization of the CAD model is fairly straight forward. With the software Catia for example, the procedure usually involves creating 2 sets of parallel planes and use them to intersect the CAD model. The result of this is 2 sets of curves on the CAD model surface as shown in Fig. 1 below.

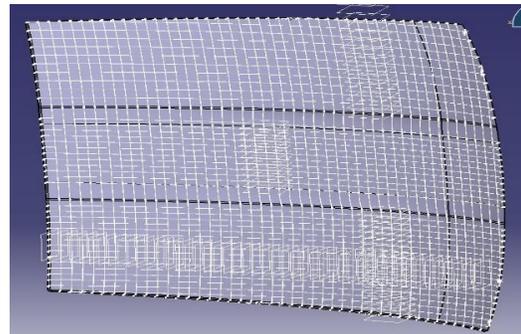


Fig. 1 Discretization of the CAD model

The intersection points of those 2 sets of curves are uniformly distributed across the surface. The coordinate data of those points are collected and used for curve and surface fitting. As for the scanned data, if the scanning directly measures the coordinates of a set points in the frame with respect to the measuring device, there is no need for discretizing, as the data can be directly used for the curve and surface fitting. Matlab has a curve fitting tool that gives quick and accurate result. During the fitting, one need to pay

special attention to the orientation of the coordinate system to avoid serious error.

The number of the points collected as well as the choice of the equation can affect the accuracy. In term of the equation choice, in general, for the same type of the equations, the equation with higher number of coefficients has higher accuracy. In term of the point density, the more points one can have the better the resulting equation describes the surface. To check whether the amount points collected is enough, first a surface function need to be generated with the coordinates set of the points collected initially, name it f_{n1} . Then reduce a certain number of points (usually by half) from that very same coordinates set while maintain the same points distribution and use it to generate another surface function, name it f_{n2} . The standard deviation of difference can be calculated with the equation below,

$$\sigma = \sqrt{\frac{\sum_{i=1}^k (f_{n1}(x_i, y_i) - f_{n2}(x_i, y_i))^2}{k-1}} \quad (1)$$

k is the number of the points (x_i, y_i) s randomly degenerated on of the part surface with a uniform distribution and should be unrelated to the coordinates set obtained from the discretization or scanned data. If 3σ is within the allowed error tolerance, the amount of the points collected is sufficient.

Nevertheless, this standard deviation of the difference can be deceiving sometime due to the fitted functions are naturally ill behaved near the edges. Fig. 2 below illustrates this issue.

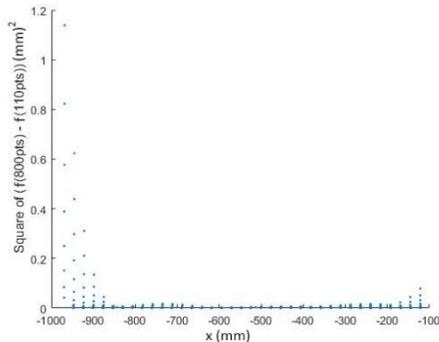


Fig. 2 Plot of $(f_{800}(x_i, y_i) - f_{110}(x_i, y_i))^2$ vs x_i over 600 uniformly distributed points throughout the area of the part surface in Fig.1. Both are polynomials of same degree.

As the result, the error over the most part of the surface is actually a lot smaller than the 3σ value. There is however a simple method to mitigate this problem. One simply need to linearly extrapolate a few extra points outside of the actual part surface based on the a few points that is closest to the edge on the part surface. The goal of this is to fix the

function value at and near the edge and shift the ill behavior part outward. For comparison, the y_{110} used in Fig. 2 is re-fitted with linear extrapolation to extend the data set along the x-direction. The result is shown in Fig. 3. The improvement to the error near the edge is very significant.

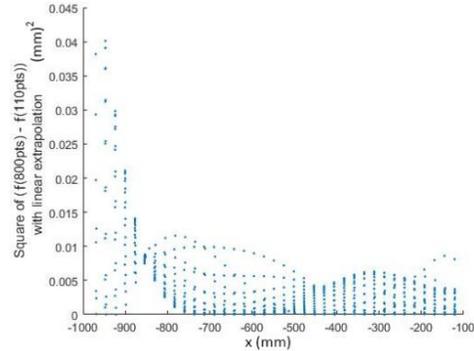


Fig. 3 Plot of $(f_{800}(x_i, z_i) - f_{110}(x_i, z_i))^2$ vs x_i over 600 uniformly distributed points throughout the area of the surface with linear extrapolation

The linear extrapolation method is extremely useful when the data collection is a challenging task. If the data collection is not an issue and can be done with automatic procedures relatively fast, linear extrapolation is not unnecessary.

3. Tool Path and Trajectory Generation

3.1 Path Generation

The tool path designed in this paper is a combination of raster path along the surface and half circular path to connect each raster curve during the turning (step-over) as shown in Fig.4.

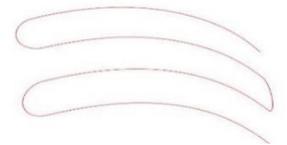


Fig. 4 Raster path with circular path during the turning.

When planning a path on a curved surface, the initial raster path is usually a straight line on x-y plane. It can also be made parallel to one of the surface edge on x-y plane for better tool coverage. The edge curve on x-y plane $y = f(x)$ can be obtained directly from the data points with curve fitting. The step-over direction should be along the direction which the part surface has the lowest average curvature and reasonable flat. Doing so, the sequential raster path on x-y plane can be approximately described by $y = f(x) + c_i$ (where c_i are constants). For most surface, the direction which it has lowest or highest curvature can be determined visually. When dealing

with more complicated geometry, the y coordinate need to be adjusted after, and the method will be described later. The complete curve equation can be written as $\vec{C}(x) = (x, y(x), z(x, y(x)))$ or in a more generalized form $\vec{C}(t) = (x(t), y(t), z(t))$.

The actual path for the tool to follow is consisted of waypoints, not continues lines. In terms of the robotic control, it's much easier to move the tool at a constant speed and with a constant control input frequency. To achieve this, every waypoint need to be separated with same arc length. This requires the calculation of the total arc length of the path. When x is the parameter, the arc length is

$$L = \int_a^b \left| \left(1, y'(x), z'(x, y(x)) \right) \right| dt \quad (2)$$

When the total amount points on the path from point a to point b is decided and total arc length L is calculated, the coordinates that corresponds to equal divided arc length can be calculated according to $\Delta L = \frac{L}{n}$. If let

$$\Delta L = \frac{L}{n} = \int_{x_i}^{x_{i+1}} \left| \left(1, y'(x), z'(x, y(x)) \right) \right| dx \quad (3)$$

where $x_0 = a$, $x_n = b$. The points a , and b is usually either the points on the edges or some predetermined points according to the size of the tool, which are already known. To obtain the x_i s, one may start with point a , and use the above to solve for each x_i in sequence. Eq (3) can be solved numerically in matlab. However, this may take huge amount computing power, and make the process very slow. Sometime, when the polishing does not require an absolute precision, one can use the converge method below to approximate the result. Start from point a , first let

$$x_1^{(1)} - a = \frac{b-a}{n} \quad (4)$$

ΔL is already known, let

$$\begin{aligned} \Delta L^{(1)} &= \int_a^{x_1^{(1)}} \left| \left(1, y'(x), z'(x, y(x)) \right) \right| dx \\ \left| \Delta L^{(1)} - \Delta L \right| &= e_1 \end{aligned} \quad (5)$$

For each sequential iteration, let

$$\begin{aligned} \Delta L^{(i+1)} &= \int_a^{x_1^{(i)} \pm e_i} \left| \left(1, y'(x), z'(x, y(x)) \right) \right| dx \\ x_1^{(i+1)} &= x_1^{(i)} \pm e_i, \left| \Delta L^{(i+1)} - \Delta L \right| = e_{i+1} \end{aligned} \quad (6)$$

The \pm depends on whether $\Delta L^{(i)}$ is smaller or greater than ΔL . The iteration exits when e_i is less than the

desired error tolerance. When iteration stops at j^{th} , $x_1 = x_1^{(j)}$. Once x_1 is determined, x_2 can be determine with the above method by replacing a with x_1 and x_1 with x_2 . This method although looks messy, but it takes much less time to calculate in matlab then solve x_i s with Eq (3). After all the x_i s are obtained, the y and z can be calculated with $y(x)$, $z(x, y(x))$.

For the cases where the y coordinate need to be readjusted based on the geometry and error tolerance, the new y coordinate need be re-calculated according to equal arc length along the step-over direction with each curve uses previous curve on the raster path or a global reference curve as reference. Assume the step-over is along y direction. First, obtain y coordinates of the raster curve from x_i s with the method above. Then one need to calculate the arc length along y direction at each point of $x = x_i$ based on the previous curve or a global reference curve. Now let

$$\Delta Y_{\pm i} = \int_{y_{0\pm i}}^{y_{1\pm i}} \left| \frac{d}{dy} (x_i(y), y, z(x_i(y), y)) \right| dy \quad (7)$$

ΔY_0 is the reference distance based on one point on each curve that can be aligned (or the closet) with the direction of the step-over. y_{0i} is the y coordinate of the reference point on the reference curve, and y_{1i} the y coordinate of the reference point on the sequential curve. y_{0-i} and y_{1-i} are the points on the left side of the reference points y_{00} and y_{10} , and y_{0+i} and y_{1+i} are the points on the right side. The y_{1i} s can be adjusted based on the converge method described above with Eq (5, 6), with now ΔL being ΔY_0 and y is the parameter. Note that, in general, the reference point y_{00} , and y_{10} is not on the edge of the part surface, as the edges are rarely aligned with direction of the step-over. Once all the y_{1i} s have been adjusted, there is yet another problem may arises, for which the equal arc length along the raster path that is calculated previously is no longer equal. However, normally if the direction of the step-over is carefully chosen with the lowest average curvature, the change in the arc length along the raster path is very small and can be complete neglected. Only when the part surface is extremely unforgiving, then each raster curve need to be re-fitted with the new readjusted x and y, to obtain a new curve equation $y(x)$, and repeat the procedure using Eq (4-6). Additionally, step-over size should be smaller than the actual tool's diameter. Because of this, it is possible there are some corners may not be touched by the tool. To mediate this, an additional path along all the edges may be required, which can be designed with the method described above with $y(x)$ (or $x(y)$) being the edge curves.

When connecting two adjacent raster paths with circular or any other type turning path, some miss-

match often occur due to the curvature of the part surface as shown in Fig. 5.

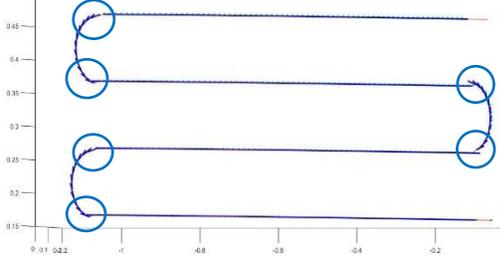


Fig. 5 Raster path with circular path during turning. (The discontinuous section is highlighted in circle.)

Using the circular path with the parameter equation $x = R \cos t + c_x$, $y = \pm R\sqrt{1 - (\cos t)^2} - c_y$ as an example, if the R is calculated to account for the curvature of the surface with Eq (7), the miss-match can be fixed by moving the center of the circle (c_x, c_y) in order to match the endings of the circle, $x(0, \pi)$, $y(0, \pi)$, with the endings of raster curve. Also note that in some cases the position of the raster curve need to be moved as well.

The advantage of using circular path for turning is that the tool can move at a constant speed along most the path without the need to perform difficult maneuvers.

3.2 Trajectory Generation

The requirements for end-effector trajectory is that it need to be at rest at the beginning of the path and reach the required constant operation speed at the end of the acceleration stage, then maintain the constant speed till the beginning of the deceleration stage and come to rest at the end of the entire path. In addition to that, the velocity profile that connects two neighbouring waypoints needs to be smooth to avoid huge acceleration at the end-effector during the transition.

The arc length of the three stages (acceleration, constant speed, and deceleration) and the corresponding speed and acceleration are $L_i(\tau_i)$, $\dot{L}_i(\tau_i)$, and $\ddot{L}_i(\tau_i)$, τ_i s are time variable, and $i = 1, 2, 3$. This in addition to the consideration of a smooth velocity profile, the boundary conditions for each stage are as follow. During acceleration (stage 1),

$$\begin{aligned} L_1(0) = 0, L_1(1) = h_1, \dot{L}_1(0) = v_0, \\ \dot{L}_1(1) = v, \ddot{L}_1(0) = \ddot{L}_1(1) = 0 \end{aligned} \quad (8)$$

The boundary conditions of the constant speed stage (stage 2) are,

$$\begin{aligned} L_2(0) = L_1(1), L_2(1) = L_1(1) + h_2, \\ \dot{L}_2(0) = \dot{L}_2(1) = v, \ddot{L}_2(0) = \ddot{L}_2(1) = 0 \end{aligned} \quad (9)$$

The boundary conditions of the deceleration stage (stage 3) are,

$$\begin{aligned} L_3(0) = L_2(1), L_3(1) = L_2(1) + h_3, \\ \dot{L}_3(0) = v, \dot{L}_3(1) = v_e, \ddot{L}_3(0) = \ddot{L}_3(1) = 0 \end{aligned} \quad (10)$$

where h_i s are the corresponding total arc length of each stage, v_0, v_e are the initial and final speed with 0 to be the default, and v is the operation speed.

The simplest equations that satisfy those six boundary conditions are

$$\begin{aligned} L_i(\tau_i) &= \sum_{n=0}^5 a_{in} \tau_i^n, \quad \dot{L}_i(\tau_i) = \frac{1}{T_i} \sum_{n=1}^5 n a_{in} \tau_i^{n-1}, \\ \ddot{L}_i(\tau_i) &= \frac{1}{T_i^2} \sum_{n=2}^5 n(n-1) a_{in} \tau_i^{n-2} \end{aligned} \quad (11)$$

T_i s are the total time interval for each stage. Substituting the Eq (8-10) into Eq (11), the six coefficients can be calculated as,

$$\begin{aligned} a_{i0} &= L_i(0), a_{i1} = \dot{L}_i(0)T_i, a_{i2} = 0, \\ a_{i3} &= 10h_i - 6\dot{L}_i(0)T_i - 4\dot{L}_i(1)T_i \\ a_{i4} &= -15h_i + 8\dot{L}_i(0)T_i + 7\dot{L}_i(1)T_i, \\ a_{i5} &= 6h_i - 3\dot{L}_i(0)T_i - 3\dot{L}_i(1)T_i \end{aligned} \quad (12)$$

To calculate the time interval T_i of the i th stage for a given h_i , the following condition

$$\dot{L}_i\left(\frac{1}{2}\right) = \frac{L_i(0) + L_i(1)}{2} \quad (13)$$

is required. Then, the time interval T_i can be written as

$$T_i = \frac{2h_i}{L_i(0) + L_i(1)} \quad (14)$$

Substituting Eq. (14) into Eq. (12), the six coefficients are

$$\begin{aligned} a_{i0} &= L_i(0), a_{i1} = \frac{2\dot{L}_i(0)h_i}{L_i(0) + L_i(1)}, a_{i2} = a_{i5} = 0 \\ a_{i3} &= \frac{2(L_i(1) - L_i(0))h_i}{L_i(0) + L_i(1)}, a_{i4} = \frac{-(L_i(1) - L_i(0))h_i}{L_i(0) + L_i(1)} \end{aligned} \quad (15)$$

The value of h_i for stage 1 and 3 can be determined according to the maximum allowed acceleration $\ddot{L}_{i,max}$, which set by the capability of the robot. When the acceleration is at maximum at time τ_i , the expression of acceleration from Eq (11) satisfying

$$\frac{d\ddot{L}_i(\tau_i)}{dt} = \frac{6a_{i3} + 24a_{i4}\tau_i + 60a_{i5}\tau_i^2}{T_i^3} = 0 \quad (16)$$

Using the results from Eq. (15) and substituting them into Eq. (16),

$$\tau_i = -\frac{a_{i3}}{4a_{i4}} = \frac{1}{2} \quad (17)$$

Therefore, the h_i can be determined with

$$h_i \geq \frac{3(L_i^2(1) - L_i^2(0))}{4\bar{L}_{i,max}} \quad (18)$$

In addition to the equations above, the unit tangent and unit normal vectors at each waypoint are also needed. The unit tangent vector is

$$\vec{T}(x) = \pm \frac{(1, y'(x), z'(x, y(x)))}{\sqrt{1 + y'(x)^2 + z'(x, y(x))^2}} \quad (19)$$

The \pm sign is due to the orientation of the coordinate system and the direction of the path, as they can affect the direction of the tangent vector which need to be adjusted according to the motion of the tool. To obtain the velocity vector at each waypoint, one simply need to multiply the speed calculated above with the unit tangent vector. Normal vector can be defined as

$$\vec{n}(x, y) = \pm \frac{\vec{s}_x \times \vec{s}_y}{|\vec{s}_x \times \vec{s}_y|} \quad (20)$$

where $\vec{s}_x = \frac{\partial \vec{s}}{\partial x}$, $\vec{s}_y = \frac{\partial \vec{s}}{\partial y}$, and $\vec{s} = (x, y, f(x, y))$.

Similar to the tangent vector, the normal vector can be affected by the orientation of the coordinate system. However, unlike the tangent vector, it cannot be affected with the direction of the path. The normal vectors are needed for the orientation of the tool.

4. Forcing Planning

The contact area on a curved surface is very largely depended on the local geometry of the surface as well as the tool shape. The contact area can be calculated with the contact stress model. For a quick summary, the contact area between two disks each with maximum principle radiuses R_1, R_2 , and a minimum principal radius R_1', R_2' is illustrated in Fig. 6 below [3-6].

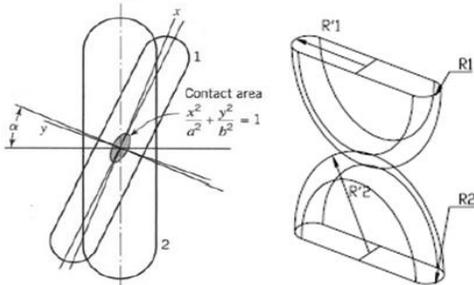


Fig. 6 Contact area between two circular discs [6]

The principal stress on the contact surface is given by [5]

$$\sigma_{zz} = -\frac{k(1-n^2)b}{k'^2 E(k') n \Delta}, n = \sqrt{\frac{k^2 + k^2 \left(\frac{Z}{b}\right)^2}{1 + k^2 \left(\frac{Z}{b}\right)^2}} \quad (21)$$

where Z is the perpendicular distance away from the contact surface, a and b are semi-major, and semi-minor axis of ellipse the and

$$\Delta = \frac{1}{A+B} \left(\frac{1-v_1^2}{E_1} + \frac{1-v_2^2}{E_2} \right), k = \frac{b}{a}, k' = \sqrt{1-k^2} \quad (22)$$

$K(k')$, and $E(k')$ are complete elliptic integral of first and second kind, A and B are function of geometrical properties only [5]. Therefore, with the principle stress given, the contact surface area is strictly determined by the geometry of the tool and part surface. For the tool, it generally has a regular shape, and most commonly flat surface. When the tool is flat, the expression for A and B reduce to

$$A = \frac{1}{R_2}, B = \frac{1}{R_2'} \quad (23)$$

R_2 and R_2' are the principle radiuses of the part surface. The applied force can be related to semi minor axis b with

$$F = \frac{2\pi b^3}{3kE(k')\Delta} \quad (24)$$

In addition to that, the relation [5]

$$\frac{B}{A} = \frac{\frac{1}{k^2} E(k') - K(k')}{K(k') - E(k')} \quad (25)$$

can be used to solve for the value of k . Additionally, with the expression of A and B for flat tool, it's easy to show from Eq (21) that the semi-minor axis of the contact surface is approximately proportional to the principle radius of the part surface under constant principle stress. When a and b being known, the contact area is easy to calculate. Reversely, if there is a specific requirement for a or b , the applied force can be calculated with Eq (24).

The calculation of principle radiuses of an arbitrary geometry requires several steps. First, the principle directions, namely $D_1(P)$ and $D_2(P)$, at a point P on surface are the directions of which the principle radiuses at point P follow. The method for calculating principle directions is described in detail in reference text [7], as well as any standard undergraduate textbook on differential geometry. The principle curvatures κ_i are the curvature of curves $\vec{C}_i(t) = (x, y_i(x), z(x, y_i(x)))$ at point P , where

$y_i = D_i(P)x + c_i, i = 1,2$. If the curvature is calculated with the method of differential geometry, the corresponding radius is exactly $R = \frac{1}{\kappa(x)}$.

For clarity, it is convenient to define major principle radius to be the greater one of the two, and the minor principle radius as the smaller one. Recall the earlier definition from contact stress model, together with the Eq (25), it is required for $R_2' \geq R_2$ so that $k \leq 1$. Therefore, R_2' need to be fixed to be the major principle radius, and R_2 to be the minor principle radius. However, in practice, the role of R_2' , R_2 can change. The plot below shows the principle directions across the upper part of the part surface in Fig. 1.

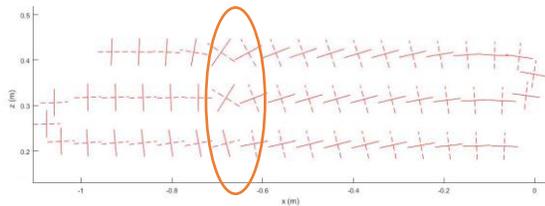


Fig. 7 Plot of principle direction pairs on the tool path generated with the method in the previous section.

It shows the rotation of the contact ellipse across the surface. More importantly, it shows a sudden rotation occurs at the highlighted area. That is because there is a line of umbilical points [7] situated. The umbilical points are the points on the surface where two principle curvatures (radiuses) equal to each other. Most surfaces have umbilical points. A principle radius along a tool path can go from major to minor or vice versa when pass through umbilical points. The value of the principle radius need to checked to ensure $R_2' \geq R_2$, and the major and minor axis of the ellipse need to be exchanged accordingly (the ellipse would lay down on the side). There are strategies such as constant principle stress can be used for force planning. In addition, constant major semi-axis is also a viable method.

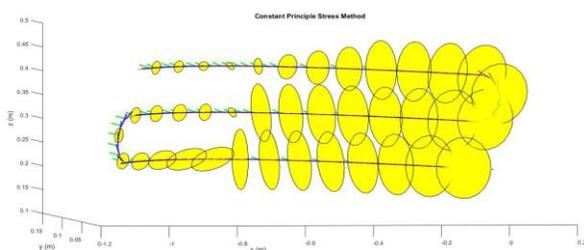


Fig. 8 Plot contact area along the tool path using constant principle stress defined in Eq (21). The z value is set to be 0.0001, materials are aluminum (part surface) and alumina (tool). To produce this plot the principle stress is set at around 10 bar.

The plot above shows the contact area varies dramatically across the upper part of the part surface in Fig. 1 according to the curvature of the part

surface. Of course, if one set the contact force big enough, the contact area will all be equal to the size of the tool. That however can cause a great deal in structure deformation, which will change the geometry of the part surface. The constant major axis method can ensure a uniform width of the polish mark when the umbilical points are absent. The umbilical points add an additional degree of difficulty in force planning. Therefore, a combination of the strategies need to be employed for part surfaces with complex geometry. Moreover, to ensure accurate result, the structure deformation due to contact force cannot be ignored. This is a topic will be closely studied in the future.

5. Summary

In this paper, a method for robotic polishing planning is purposed that consists three elements. The first element is to generate a mathematical model based on the discretization of the part surface model. The second element is to plan a tool path and trajectory on the part surface based on the mathematical model obtained in the first element. The third element is force planning. By using the contact stress method, a uniform width of the polishing track can be possibly achieved by adjusting the applied force according to the local geometry. structure deformation due to contact force will be researched in future to ensure better result.

Acknowledgement

The research was supported by Natural Sciences and Engineering Research Council of Canada, National Natural Science Foundation of China (11502052) and State Scholarship Fund of China (201706655033)

References

- [1] C.K. Toh, "A study of the effects of cutter path strategies and orientations in milling," *Journal of Materials Processing Technology*, 152 (2004) 346–356.
- [2] W. Sheng, N. Xi, M. Song, and Y. Chen, "CAD-guided robot motion planning," *Industrial Robot: An International Journal*, 28 (2) (2001) 143-151.
- [3] L. Liao and F. Xi, "A Linearized Model for Control of Automated Polishing Process," in *Proceeding of 2005 IEEE Conference on Control Applications*, Toronto, Canada, August 28-31, 2005.
- [4] Avery Roswell, Fengfeng (Jeff) Xi, Guangjun Liu; *Modelling and analysis of contact stress for automated polishing*; *International Journal of Machine Tools & Manufacture* 46 (2006) 424–435
- [5] A.P. Boresi, R.J. Schmidt, O.M. Sidebottom, *Advanced mechanics of materials*, Fifth ed., Wiley, 1993.
- [6] Michael Rososhansky, Fengfeng (Jeff) Xi, and Yuwen Li; *Coverage Based Tool Path Planning for Automated Polishing Using Contact Stress Theory*; 2010 IEEE International Conference on Automation Science and Engineering, 2010
- [7] Thomas Banchoff, Stephen Lovett; *Differential Geometry of Curves and Surfaces*; CRC Press, Taylor & Francis Group, LLC, 2010