Int J Adv Manuf Tech manuscript No.
(will be inserted by the editor)

Uniform scanning path generation for abrasive waterjet polishing of free-form surfaces modeled by triangulated meshes

Hamed Khakpour · Lionel Birglen · Souheil-Antoine Tahan.

Received: date / Accepted: date

Abstract In this paper, a method is proposed to generate scanning paths to be used in automated abrasive waterjet polishing of free-form surfaces. This method is able to produce trajectories with constant offset distance between curves on surfaces with holes and complex boundaries without reconfiguration of their triangular mesh model. For this, the particular requirements of this polishing technique to be kept along the path are investigated. Next, a reference curve is obtained and using geodesic distances in specific directions, the adjacent offset curves are found. Finally, if needed, the main trajectory is divided into a set of continuous sub-trajectories. By defining two indices, the effect of the shape of the surface and the configuration of the generated path on the uniformity of the distribution of the waterjet is evaluated. Through several examples, it is shown that the method can effectively generate scanning paths adapted to the requirements of this technique.

H. Khakpour
Mechanical Engineering Department
Ecole Polytechnique of Montreal
2900 Boulevard Edouard-Montpetit
Montreal, QC, CANADA, H3T 1J4
Tel.: +1(514)340-5121 ext. 2234
Fax: +1(514)340-5867
E-mail: hamed.khakpour@polymtl.ca

L. Birglen
Mechanical Engineering Department
Ecole Polytechnique of Montreal
Montreal, QC, CANADA, H3T 1J4
E-mail: lionel.birglen@polymtl.ca

S.-A. Tahan
Mechanical Engineering Department
Ecole de Technologie Superieure
Montreal, QC, CANADA, H3C 1K3
E-mail: antoine.tahan@etsmtl.ca

Keywords Abrasive waterjet polishing · trajectory planning · triangular mesh · free-form surface

1 Introduction

In high-tech manufacturing, to produce objects with free-form surfaces, different steps of machining including milling, grinding, and polishing may be performed. Amongst them, the polishing process is one of the most delicate procedures in which the method of sweeping the tool on the surface has a significant impact on its final quality.

So far, several techniques (e.g., sanding disc, electric discharge, ultrasonic, and abrasive waterjet polishing (AWJP) [1, 2]) have been developed to polish surfaces with different shapes and properties. Amongst these, AWJP is a novel non-conventional method in which the material removal from the surface takes place due to the collision of the abrasives with the surface [3]. In this method, there is no direct contact with the surface and the area affected by the jet is usually short (usually a few millimeters). Thus, it is critical to keep the material removal rate of the process constant and uniformly deliver the slurry jet to all areas of the desired surface. For this, it is required to keep the AWJP parameters at their optimal levels during the process and generate a polishing path in which the particular properties of this technique are considered. This paper is dedicated to the study of the requirements of a trajectory generated for this particular application and then, the development of a method to produce this type of paths.

Trajectory planning is a well-known field in computer-aided design (CAD) and many works have been done in this area. Depending on the application, different methods of surface-covering path generation were developed

[4–15]. These trajectories are mainly generated in two ways. In the first, the main objective is to only pass the tool over the entire surface, and thus, a constant offset distance between adjacent paths is not important. One of the common strategies used for this type of trajectories is to use parametric surfaces [10–12]. In these approaches, the trajectory is first generated on a planar surface and then, it is mapped onto the desired free-form surface. The pitch adaption method developed by Tam et al. [14], and boundary-conformed iso-parametric tool path generation methods presented by Yang et al. [11] and also Yuwen et al. [12] are examples of this strategy.

In the second approach, it is critical to uniformly cover the entire surface with the trajectory. For this, the trajectory is often obtained by considering a reference point/line/path as a starter to generate the adjacent offset curves while the constant offset distance is preserved. Depending on the requirements of the application, this reference can be a point [16, 17], a seed curve [4, 17–19], or a boundary of the surface [6, 8, 17, 18]. In this approach, the concept of geodesic distance fields is typically used to find the offset curve of a reference on the free-form surface [16–20]. This method was investigated and formulated by Patrikalakis et al. [21, 22] for free-form surfaces. For the case of surfaces modeled by triangulated faces, it was initially used by Surazhsky et al. [16] and was based on the interval propagation presented by Mitchell et al. [23]. Subsequently, Bommes et al. [18] improved the original algorithm by reconfiguring the triangular mesh of the surface to obtain a fine and smooth curve.

In this paper, the shape of the material removal profile in the area polished by the AWJP method is first investigated. Then, considering its properties, the limits for the curvature of the surface, pitch distance, and the configuration of the trajectory are determined. Afterwards, by modeling the material removal from the surface (in the planar case), it will be shown that the shape of the trajectory affects the uniformity of the material removed from a surface. Finally, a modular method is proposed to generate the surface-covering scanning path on a triangular mesh model defining the free-form surfaces to be polished. This method is developed to generate uniformly distributed trajectories on single surface with arbitrary shape (within the particular limits for the AWJP process), including holes and a complex outer boundary. In the proposed algorithm, instead of using windows (intervals) to find geodesic distances from a point or a line [16, 18, 23], the geodesic distances are calculated in specific directions. Thus, this method is simpler than the geodesic distance approach but still effective and can generate smooth trajectories on a surface for AWJP process while reconfiguration of the triangular mesh is not necessary. Lastly, by defining indices, the performance of the generated trajectories for AWJP process is investigated.

2 Estimation of material removal in AWJP

2.1 Shape of the material removal profile

In AWJP, the material removal occurs due to the collision of abrasive particles and the surface. The shape of the material removal profile depends on several polishing parameters including the nozzle diameter/angle, stand-off-distance, abrasive type/size, and speed of the jet [3]. The authors have done an experimental study on the effect of AWJP parameters on the quality on the polished surface (which is out of the scope of this paper). In these experiments, the nozzle traversed along straight lines over flat surfaces. As a part of this study, the profile of the material removal in the direction perpendicular to the nozzle movement direction was determined (assumed to be symmetrical). Although the dimension of the profile depends on the aforementioned parameters, its general shape was similar in all the tests. An example of the profile obtained by combining the results of several experiments is shown in Fig. 1(a). In this figure, the polished line is along x axis and the dimension in the direction of z axis (depth of penetration) is normalized to keep the generality of the model.

However, the exact shape of this profile is not always the same and depending on the polishing parameters it may vary a little. Thus, the presented profile is used to only give an idea about how to find the desired pitch distance. Also, the scanning path is considered as the polishing trajectory to find the distribution of waterjet. This model is only used to estimate the distribution of waterjet over the surface but not the real shape of the polished area. The hypothesis is valid only if the ratio of the depth of penetration to the width of the polished area, D/W, is negligible. Otherwise, the interaction between abrasives and the surface may be affected and thus, this model cannot estimate the consequences. Taking into account these considerations, the estimation of material removal from the surface using three parallel lines are presented in Figs. 1(b)-(e).

As it can be seen in Figs. 1(b) and (c), when the pitch distance p between two adjacent lines is properly adjusted, the material removal happens evenly. On the other hand, when p is longer or shorter, some hills or valleys may respectively remain between the paths (c.f. Figs. 1(d) and (e)). In the particular case shown in Fig. 1 where $W = 10\ mm$, the total variation in the depth of the polished area in case (c) with $p = 4\ mm$ is

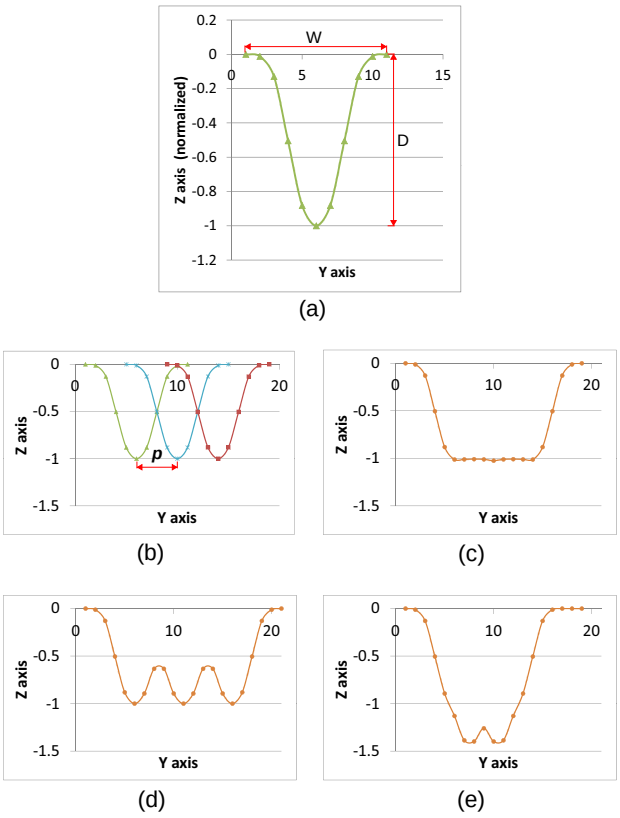

Fig. 1: (a) Profile of the material removal obtained from experiments (with normalized depth); (b) profiles of three consecutive polishing lines; (c) estimated profile of material removal using three polishing lines with p=4mm; (d) p=5mm; and (e) p=3mm.

1.6% of D, while in cases (d) and (e) it is respectively 37.4% for $p = 5\ mm$ and 13.9% for $p = 3\ mm$. Since the exact curve of the profile may slightly change with different polishing parameters, these values give an order of magnitude but can vary in practice. However, knowing the shape of the profile, the optimal pitch distance can be properly estimated by trial and error.

With free-form surfaces, the initial profile of the surface can be wavy and assuming the nozzle axis perpendicular to the surface, the interaction between profiles can be different. This difference highly depends on the ratio between the depth of the penetration and the radius of curvature of the surface around the polished area, i.e. D/r_k. If this ratio is large (e.g. $D/r_k = 0.25$) then, in concave or convex parts of the surface the deepest parts of the profiles respectively diverge from each other or converge together and consequently the overall shape of the machined surface changes. However, since in polishing process D is very small (usually less than 20 μm), this effect is often negligible. On the other hand, the angle between the nozzle axis and the surface has a

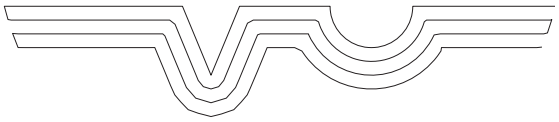

Fig. 2: Planar trajectory with sharp turns along the paths (p=4mm).

significant impact on the shape of the profile. Thus, it is necessary to minimize the effect of the radius of curvature of the surface, r_k, on this angle. Consequently, to have a constant pitch distance on the entire trajectory and maintain the uniformity of the shape of profile, the condition $r_{k.min}/W \geq 6$ is introduced to be respected for the minimal radius of curvature on the entire surface in order to keep the maximum difference in the angle between the normal of the surface and the jet nozzle axis less than 5 degrees. Thus, to polish a surface with smaller $r_{k.min}$, it is recommended to use a nozzle with smaller diameter.

2.2 Effect of configuration of the trajectory on the uniformity of polishing process

Considering the geometry of a surface, preserving the desired pitch distance does not guarantee the uniformity of the polishing process. The other parameter which can affect its uniformity is the configuration of the trajectory. Indeed, the distribution of waterjet on the areas with almost straight offset curves is not the same as that of curves with sharp turns. Therefore, an algorithm is developed to investigate the capability of the generated path in the uniform distribution of abrasive waterjet on a flat surface. As an example, the 2-D scanning path illustrated in Fig. 2 is considered. It is assumed that the nozzle axis is normal to the surface and thus, as presented in Fig. 3(a), the resultant profile of the material removal is symmetric (this can actually be done with any other profile). Then, considering a constant traversing speed along the path, the final shape of the surface is predicted and shown in Figs. 3(b) and (c).

This simulation reveals that if there is a sharp turn along the path then the variation in the depth of penetration increases. Consequently, the uniformity of waterjet distribution is lost around that area. With the example shown in Fig. 3, the maximum variation in the areas with sharp turn (indicated by ellipses) is 39.8%. To decrease this, a limit should be taken into account for the radius of curvature along the path, r_{path}. Using this algorithm, the ratios between the variations of the depths of the polished areas in curved parts of the trajectory and D are calculated and presented in

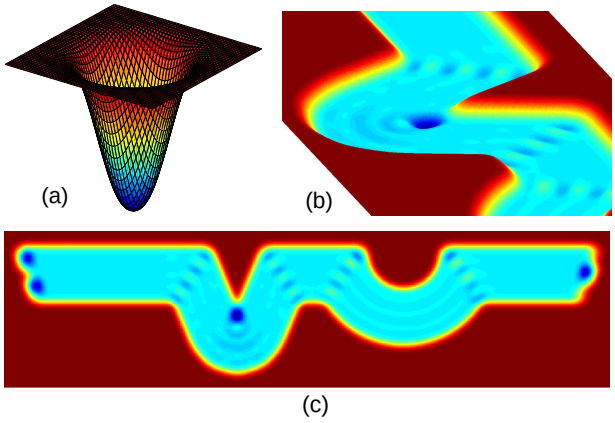

Fig. 3: (a) Profile of material removal in polishing spot by a nozzle with normal angle ($W = 10mm$); (b) 3-D view of the simulation of the shape of a surface polished through a planar trajectory ($p = 4mm$); (c) top view of this surface.

Table 1: Effect of the curve of the trajectory on the variation of depth of the polished area in a flat surface.

	length of r_{path}, ($W = 10$ & $p = 4$) all in mm				
	4	8	12	16	20
Variation %	11.5	7.1	5.4	2.52	2.44

Table 1. As expected, by increasing the r_{path}, this variation decreases. Thus, considering the particular profile of material removal illustrated in Fig. 3(a), with $r_{path}/W \geq 1.5$, a less than 5% of variation can be expected. Additionally, as computed, if a piecewise path is used to create this curvature, to keep the variation smaller than this value, the maximal angle between two consecutive line segments of the path, θ_{path}, should be smaller than $\pi/6$.

3 Scanning path generating algorithm

In the proposed method, a reference curve is first created through several options. Afterwards, for each point on this path, the tangential vector is computed as a bisector of the two consecutive line segments sharing this point, to define the normal of a plane. This plane is used to find the particular direction(s) in which the points of the adjacent offset curve located at the same geodesic distance from the point is searched. Afterwards, the procedure continues to obtain the next curve until no new one can be found. This procedure is then finished by connecting the ends of the curves to have a single scanning path. In the cases where it is not possible to generate continuous path, several methods were presented in the literature to divide an original surface into regions in which single continuous trajectories can be produced [24]. However, since in the proposed method the entire trajectory is first generated, an algorithm is developed to detect the discontinuous areas on the original trajectory and divide it into several continuous sub-trajectories.

3.1 Generation of the reference path

The algorithm starts the path generation process by collecting the data of the triangular faces of the surface. Next, it searches for the boundaries of the surface. There can be both outer and inner boundaries (i.e. the edges of either holes or obstacles inside the surface, if they exist), designated by OB and IB respectively. Next, the starting and final points (respectively P_s and P_f) of the reference curve are determined. The positions of these points on the surface have a critical impact on the general shape and pattern of the entire trajectory which significantly affects the uniformity of the distribution of the waterjet on the surface . To find them, three options are considered in this algorithm:

1. Selecting the two furthest vertices on the outer boundary of the surface;
2. If the outer boundary can be recognized as a set of edges connected together through sharp ends (e.g., boundary of a rectangular surface), then the two end points of the longest edge can be chosen, otherwise either the previous or the next option may be used (e.g., in the case of a surface with a rounded outer boundary);
3. If a particular direction or pattern is needed, two user-defined points can be used.

With the first two options, the OB is used. With option 1, the centroid of all the vertices of the OB is first found, and then, the furthest vertex from this point is selected as the starting point. The final point is the furthest vertex of the OB from the latter. With option 2, the angle θ_b, between each pair of two consecutive edges of the boundary and the average of all these angles, $\bar{\theta}$, are calculated. Then, the vertices for which this angle exceeds a predefined maximum limit are considered as sharp ends.

In one strategy, the reference curve can be obtained as an intersection of the surface and a plane passing through these two points. To find the coordinates of this plane, first, the average normal of the whole surface is obtained as $\bar{\mathbf{e}} = \frac{\sum_{g=1}^{n} \mathbf{e}_g}{\|\sum_{g=1}^{n} \mathbf{e}_g\|}$, where $\mathbf{e}_g$ is the normal vector of the gth face. Then, the normal of this plane is

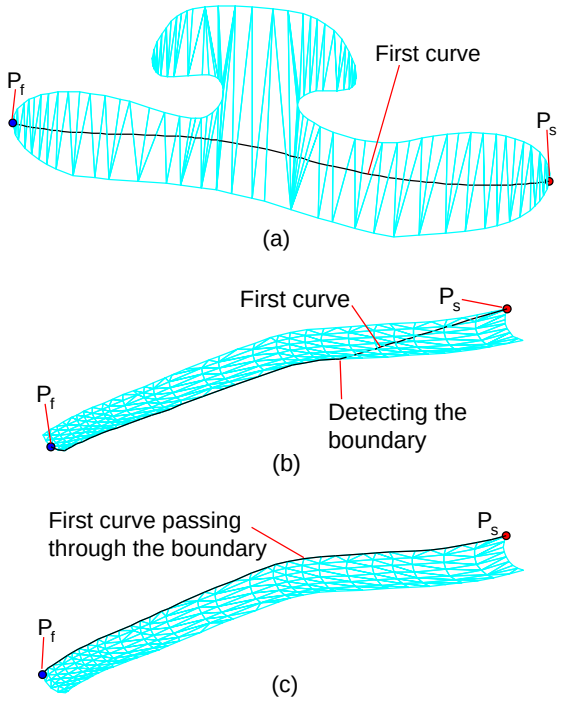

Fig. 4: Creating the reference curves using (a) the two furthest points while it completely passes though the surface; (b) the two furthest points while it partially coincides with the boundary; (c) two points on two sharp ends of the boundary.

calculated as $\mathbf{e}_p = \frac{\mathbf{v}_{if} \times \bar{\mathbf{e}}}{\|\mathbf{v}_{if} \times \bar{\mathbf{e}}\|}$ where $\mathbf{v}_{if} = \mathbf{p}_f - \mathbf{p}_i$, $\mathbf{p}_i$ and $\mathbf{p}_f$ are respectively the position vectors of the current point (initially $P_i = P_s$) and point P_f.

Next, the intersection of this plane with the surface (which is a set of points creating the reference curve) can be found. This intersection curve may not be continuous and in some cases the plane may go beyond the boundaries of the surface. To avoid this problem, the algorithm starts searching for intersection points from the point P_s and step by step proceeds toward point P_f. In each step, one point is found and is considered as point P_i to obtain the next intersection.

If the point P_i is on a inner boundary (a hole) of the surface and no face exists in the way toward the point P_f, the next intersection of this boundary with the plane in the proceeding direction is considered as a new point. If it is on the outer boundary and again no face exists in the desired direction, the algorithm follows the corresponding boundary in the proper direction until it reaches the point P_f. In Figs. 4(a) and (b), examples of creating the reference path using the two furthest points method are illustrated. As the latter figure shows, when the current point reaches the surface outer boundary, it follows the latter until point P_f.

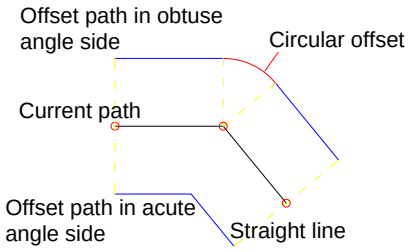

Fig. 5: Ideal offsets for a pair of line segments in a planar surface.

With the second option, the previously discussed plane is not used and as shown in Fig. 4(c), the corresponding sub-boundary between points P_s and P_f is considered as the reference curve. If the third option (user defined points) is used, the strategy described for the first option is used again. Using different options to find the reference path gives the algorithm the capability to generate scanning paths with different properties and then select the one with the best performance (based on the indices which are presented in Section 4).

With this method, the number of points found when constructing the reference curve depends on the number of faces/edges located along its way. Therefore, the distances between pairs of these points are variable. This can affect the precision of the offset curves and weakens the uniformity of the entire trajectory. Thus, this distance should be kept shorter than an appropriate limit which is defined as:

$$UL = min\left\{\frac{p}{s}\ ,\ Ed_m + \left(\frac{Ed_M - Ed_m}{s'}\right)\right\} \quad (1)$$

where Ed_M and Ed_m are respectively the lengths of the longest and shortest edges of the surface; s and s' are respectively constant values used to adjust the effect of the pitch and the shape of the faces on this limit. By trial and error it is suggested to use $s \geq 2$ and $s' \geq 20$ to obtain smooth and accurate offsets. If the distances between each pair of two consecutive points are larger than UL then, by interpolation, new point(s) are generated between them. Then, without reconfiguring the surface and increasing its total number of faces, the precision of the path can be controlled.

3.2 Generation of the offset curves

The ideal offsets for a pair of line segments of a piecewise curve in the planar case are shown in Fig. 5. As depicted in this figure, in the acute angle side of these lines, the offsets intersect each other. While in the obtuse angle side, a part of the offset has a circular shape. Since, the path has a piecewise structure, this circular

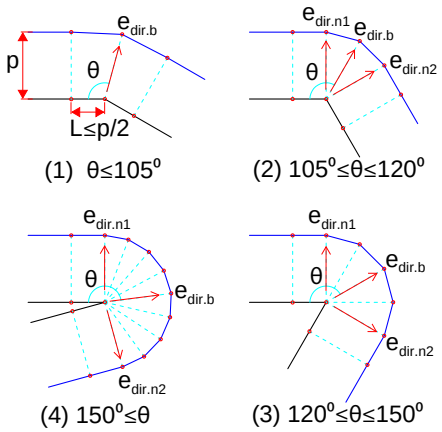

Fig. 6: Directions considered to find the points of the next offset based on the angle between the line segments and their bisector in obtuse side of the path.

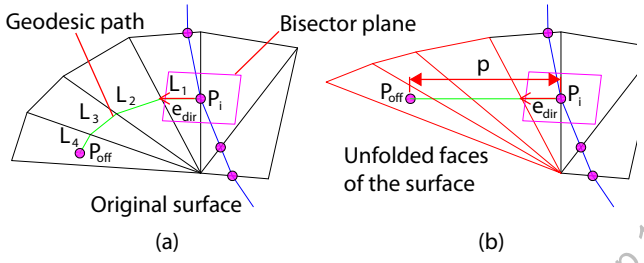

Fig. 7: (a) Geodesic distance between points P_i and P_{off} on a part of the 3-D surface modeled by triangular faces; (b) the geodesic distance coincides with the Euclidean distance when the corresponding faces are unfolded.

parts are modeled by a set of lines. For this, as illustrated in Fig. 6, some cases are defined to obtain the number of points which are representing the circular area and keep the error in the exact values of the pitch distance lower than its maximal accepted value. To calculate the angle θ between one of these two consecutive lines and the vector $\mathbf{e}_{dir.b}$ (which is along the intersection of their bisector plane with the three-dimensional (3-D) surface), they are first mapped onto the tangential plane attached to point P_i. Through this mapping, the effect of the curvature of the surface on this angle is cancelled.

As shown in Fig. 6, in case (1), $\theta < 105°$ and only the vector $\mathbf{e}_{dir.b}$ is used. In case (2), $105° \leq \theta < 120°$and the vectors $\mathbf{e}_{dir.n1}$ and $\mathbf{e}_{dir.n2}$ are also obtained. They are the intersections between the surface and the planes defined by point P_i and normals along the two consecutive lines. In case (3), $120° \leq \theta < 150°$ and the two bisector planes between these three consecutive vectors are computed and then, their intersections with the surface around point P_i are considered as the two additional directions. Finally, in case (4), $\theta \geq 150°$ and similarly, four other vectors amongst the abovementioned five consecutive vectors are obtained. Then, the maximum error would be less than 3.7% on a planar surface.

When the initial direction is known, the search for next point on the offset curve, P_{off}, is started. For this, conversely to the conventional method of propagation of intervals through triangular faces, in this paper, the geodesic distance is calculated for particular initial directions. With the limits considered for the curvature of the surface with respect to the pitch distance (as a ratio of W) in the AWJP process, this method is efficient and precise while kept simple. In the proposed method, the point P_{off} is found by unfolding the triangles which are located in the direction started from P_i and defined by the assigned direction vector $\mathbf{e}_{dir.j}$ (subscript $dir.j$ indicates the jth direction vector found for P_i). Then, the 3-D surface made of triangular faces is changed to a planar surface and the geodesic distance coincides with its Euclidean counterpart. Example of this process is depicted in Fig. 7.

This procedure is started by finding the face/edge on which the geodesic curve should be followed toward the new point P_{off}. When the first reference face/edge is obtained, the algorithm obtains the location of the first temporary point $P_{temp.1}$ which is located in the proceeding direction toward point P_{off}. If this reference is a face, the algorithm finds the edge located in the side of positive direction of the vector $\mathbf{e}_{dir.j}$ with respect to point P_i and then calculates the intersection of this edge and jth plane associated with vector $\mathbf{e}_{dir.j}$ (i.e., the point $P_{temp.1}$). If the reference object in an edge, the vertex which is in the positive direction of the vector $\mathbf{e}_{dir.j}$ is considered as point $P_{temp.1}$.

If point P_i is on a boundary of the surface, then that boundary is used to search for the new offset point. For this, the algorithm recalls the vertices of that boundary and finds the edge/vertex on which point P_i is located. Then, the proper direction along an edge of that boundary is obtained either using the vector $\mathbf{e}_{dir.j}$ or the location of the previous offsets on the same boundary. Then, the first vertex of boundary located in the proper direction is considered as $P_{temp.1}$. But, if the adjacent points of P_i on the current path (which is initially the reference curve) are also located on the same boundary and the vector $\mathbf{e}_{dir.j}$ is pointed toward outside of the surface, then this calculation for point P_i is stopped.

If point $P_{temp.1}$ is found, the length of the line segment $P_iP_{temp.1}$ (i.e., L_1 in Fig. 7) is calculated. If $L_1 >$

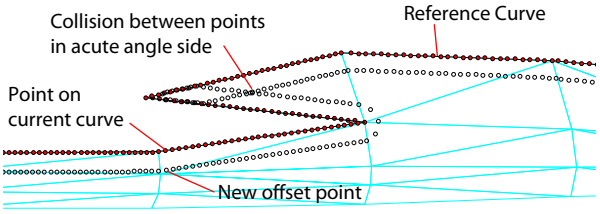

Fig. 8: Finding the offset points of a curve with two sharp turns on a free-form surface

p, the position vector of the offset point is obtained as:

$$\mathbf{p}_{off} = p\frac{\mathbf{p}_{temp.1} - \mathbf{p}_i}{\left\|\mathbf{p}_{temp.1} - \mathbf{p}_i\right\|} + \mathbf{p}_i, \tag{2}$$

If $L_1 = p$, then $\mathbf{p}_{off} = \mathbf{p}_{temp.1}$. In both situations, the search for a new point finishes.On the other hand, if $L_1 < p$, the position of the point $P_{temp.1}$ is considered as the new current point, the offset distance is changed to $p_{tm} = p - L_1$. Also, the vector $\mathbf{e}_{dir.j}$ is rotated around the intersection of the planes of the current face and the next face which is located in the proper direction (this intersection includes their shared edge/vertex) to be placed on the plane of the next face (this is equivalent to unfolding the next face with respect to the current one) as:

$$\mathbf{e}'_{dir.j} = \mathbf{R}_{\mathbf{u},\gamma}\mathbf{e}_{dir.j} \tag{3}$$

where $\mathbf{R}_{\mathbf{u},\gamma}$ is the rotation matrix around vector $\mathbf{u}$ with the angle of γ. It is defined as:

$$\mathbf{R}_{\mathbf{u},\gamma} = \mathbf{1}\cos\gamma + sin\gamma\,[\mathbf{u}]_\times + (1-\cos\gamma)\mathbf{u}\mathbf{u}^T \tag{4}$$

where $\mathbf{u} = \frac{\mathbf{n}_1\times\mathbf{n}_2}{\|\mathbf{n}_1\times\mathbf{n}_2\|}$ and $\gamma = \arccos\left(\mathbf{n}_1^T\mathbf{n}_2\right)$. Vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ are respectively normals of the current face and the next face on which the vector $\mathbf{e}_{dir.j}$ should be placed. Also, $[\mathbf{u}]_\times$ is the cross product matrix of the vector $\mathbf{u}$. Next, the same process is continued with new temporary point and direction vector until the fth step where $L_f < p$ and consequently the point P_{off} is found (e.g., $f = 4$ in Fig. 7(a)).

If the new point $P_{temp.d}$ is on a vertex or an edge of a boundary, then similarly to point P_i, instead of following the geodesic path that boundary is followed until the point P_{off} is found. Thus, if the boundary in the proceeding direction toward the point P_{off}, then this point is always found on that boundary. Consequently, the continuity of the offset path is not disrupted by that boundary and no area on the surface is lost.

In Fig. 8, an example of generating offset points for a curve on free-form surface is presented. In the parts of the reference curve where there is no sharp turn, the resultant set of new points can constitute a smooth offset curve. But, in areas with sharp turns in the current curve, as expected, the offset points may collide with each other in the acute angle side. Therefore, after obtaining the offset points for the current curve, another subroutine is needed to modify these points in the following steps:

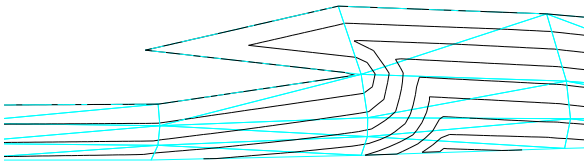

Fig. 9: Post-processed offset curves when the reference curve has sharp turns on a free-form surface

1. Replacing the points of the offset curve which are closer together than the threshold (e.g., $p/50$) with a point located in their middle;
2. Deleting the offset points which are closer than a specified minimal distance to the current curve. This usually happens in the acute angle side of the current curve (c.f. Fig. 8) or when the points are located on the boundary;
3. Checking the angle between each pair of two consecutive points on the new path and detecting sharp zigzags, turns, and loops caused by the tolerances considered in this numerical approach. Then, either modify or delete these points;
4. Comparing the distance between each pair of new points with the upper limit presented in Eq. (1). Then, if needed, new points are added to this curve similarly to the reference curve.

After all these modifications, the final shape of the new path is obtained. Then, it is connected to the previously generated paths to create the total trajectory. Following this step, the algorithm uses the new path as a current curve to repeat the same process and obtain the next curve. This procedure is continued until no offset curve can be found for the current path. As an example, the offset paths generated considering the reference curve with sharp turns shown in Fig. 8 are illustrated in Fig. 9. Next, the algorithm returns to the reference curve and with the same procedure obtains all offset curves on the other side of the reference path.

The final single trajectory generated on the free-form surfaces of a turbine blade, a channel, and a fillet (previously illustrated in Fig. 4c) are presented in Figs. 10a-c. As it can be seen in these figures, the offset distance is kept constant between all the offset curves, except for the two distal ones on each side of the trajec-

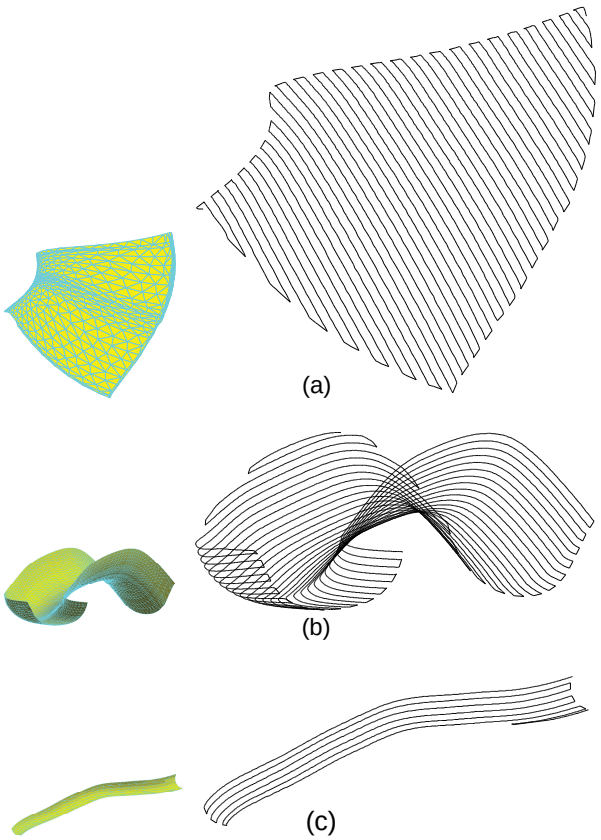

Fig. 10: Single scanning path generated on the free-form surface (a) of a turbine blade; (b) inside a channel with irregular shape; (c) inside a fillet.

tory. The algorithm does this alteration intentionally to adapt the configuration of these two end paths to the shape of the boundary.

3.3 Dividing the original trajectory

In the path generation process, if the surface has either concave parts in the outer boundary or internal holes, the generated curves may not be continuous on the surface and thus, they have to be divided into several pieces. Examples of these situations are shown in Fig. 11. These examples reveal that it is not always possible to have a single continuous scanning path on any arbitrary surface while a constant offset distance is maintained.

Thus, another supplementary module is used to detect these areas and divide the main trajectory into a set of sub-trajectories which are properly adapted to the surface. This module is made of two subroutines. The first one checks for the discontinuous areas of the tra-

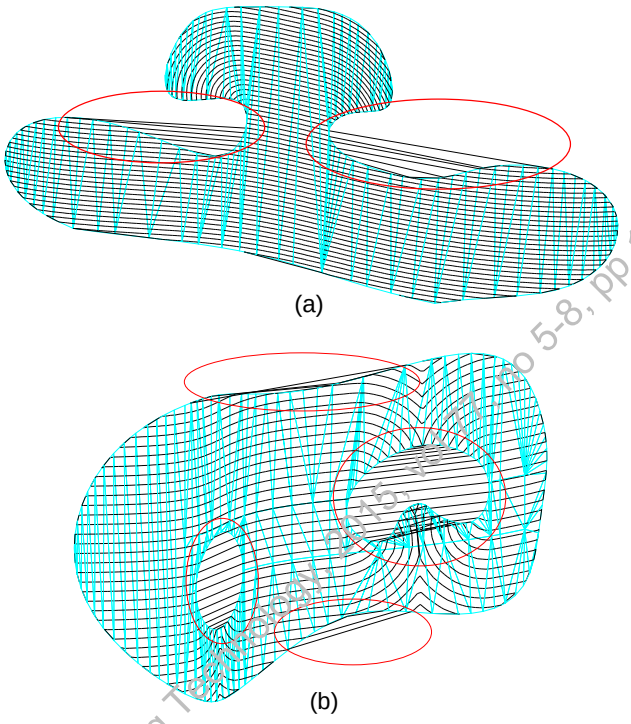

Fig. 11: Discontinuous scanning paths (highlighted by red ellipses) generated on a surface with (a) a concave boundary; (b) a concave boundary and two holes.

jectory. The second reconfigures the trajectory in these areas to produce a new set of the trajectories.

The first subroutine takes the first point of the trajectory and follows the path toward its other end and searches for lines which are not located on the surface. To find whether a line is on the surface or not, it evaluates the length of that line and the location of its midpoint. If this length is larger than a limit defined based on the maximum distance of two consecutive points on the generated path, the current line is eliminated from the trajectory. Otherwise, the algorithm checks the location of its midpoint. If this point is on the surface or if it is not on the surface but its distance with the boundary of the surface is less than a specified limit, then the current line remains valid and the corresponding point is considered in the current sub-trajectory. The subroutine then proceeds with the next line segment. If it is not accepted, the second subroutine is used to search for the new connecting line. It takes this last point and searches for the remaining points of the main trajectory which are located within a specified distance. When these points are found, they are sorted as either located on the boundaries or inside the surface.

If the last point of the current sub-trajectory is located inside the surface, then both of these groups of points are considered. But if it is on a boundary, then

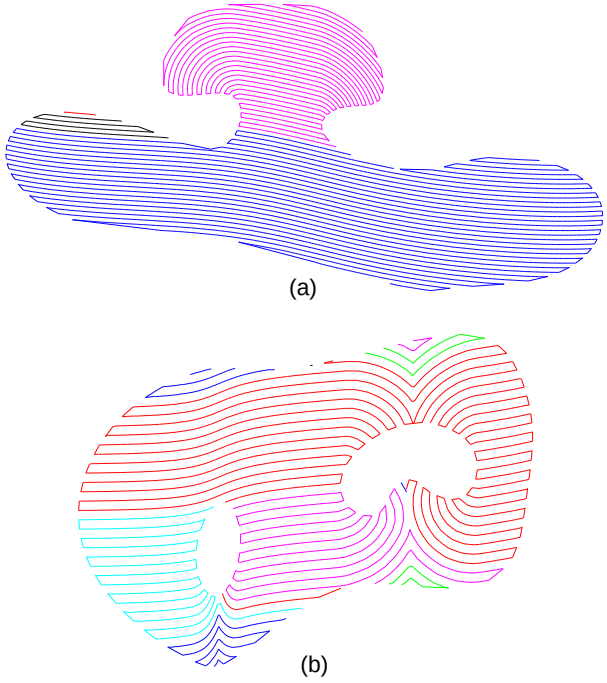

Fig. 12: Dividing the original scanning path into continuous sub-trajectories for a surface with (a) a concave boundary; (b) a concave boundary and two holes.

only the first group of points is taken into account. Next, the subroutine does the same test on the new line segments created by this last point and all those points (i.e., checking their length and midpoint position). Amongst valid line segments, the one with the shortest distance is considered. When a new point is accepted, the subroutine checks the main trajectory from this point to its other end. Finding no connection point means the end of the current sub-trajectory. Thus, again the same subroutines are used to obtain other sub-trajectories from the remaining points of the trajectory.

This modification is done for the surfaces previously shown in Fig. 11 and the resultant sub-trajectories are presented in Fig. 12. However, as it can be seen in these figures, some sub-trajectories are quite short. The reason for this is that the subroutine tries to use all the generated points and does not discard them until no more points remained around. Depending on the application, these short sub-trajectories may be manually ignored by the user.

4 Performance of the generated path in AWJP

The discused method is developed to generate trajectories for AWJP process. As mentioned in Section 2, for the AWJP of a free-form surface, some particular requirements are to be preserved. Since the shape of the reference curve has a critical impact on the configuration of the final trajectory, several options are considered to obtain this first curve. Also, in this algorithm two indices are defined to evaluate the properties of the surface and the generated path for the AWJP process. They are the curvature index, I_{curv}, and the path index I_{path}. The first one evaluates the curvature of the surface and is defined as:

$$I_{curv} = \frac{a_m}{a_t}\left(\frac{BL1}{\frac{1}{m}\sum_{j=1}^{m} r_{k.min.BL1}(j)}\right) \tag{5}$$

where a_t is the total area of the surface; a_m is approximation of the area around the vertices on which the minimum radius of curvature is less than its limit $BL1$ (e.g., $BL1 = 6W$); $r_{k.min.BL1}(j) \in \{r_{k.min}(j) < BL1\}$ for $j = 1, \cdots, m$ and $r_{k.min}(j)$ is the minimum radius of curvature obtained using an osculating circle [25, 26] around vertex v_j considering its one-ring of neighborhood and is defined as:

$$r_{k.min}(j) = \min\left\{\frac{\|\mathbf{v}_i - \mathbf{v}_j\|^2}{\mathbf{n}_j^T(\mathbf{v}_i - \mathbf{v}_j)}\right\} \tag{6}$$

where $\mathbf{v}_i$ is the position vector of the ith vertex in the one-ring neighborhood and $\mathbf{n}_j$ is the estimated surface normal at jth vertex. With an ideal surface for AWJP process, $I_{curv} = 0$, otherwise it has a value which is to be minimized.

The second index evaluates the smoothness of the generated path and is defined as:

$$I_{path} = \frac{L_{p.q}}{L_{p.t}}\left(\frac{q.BL2}{\sum_{d=1}^{q} r_{path.BL2}(d)} + \frac{\sum_{d=1}^{q}\theta_{path.BL3}(d)}{q.BL3}\right) \tag{7}$$

where $L_{p.t}$ is the total length of the trajectory; $L_{p.q}$ is approximation of the length of trajectory where either r_{path} or θ_{path} exceed the desired limits, respectively $BL2$ and $BL3$ (e.g., $BL2 = 1.5W$ and $BL3 = \pi/6$). Also, $r_{path.BL2}(d) \in \{r_{path}(d) < BL2\}$ for $d = 1, \cdots, q$ and $\theta_{path.BL3}(d) \in \{r_{path}(d) > BL3\}$ for $d = 1, \cdots, q$ which are calculated after projection of the two consecutive lines of the trajectory onto the tangential plane attached to the free-form surface at their intersection to eliminate the effect curvature of surface in this investigation. Again in ideal condition one has $I_{path} = 0$. The points which are located either on the boundary or on the U turn parts of the scanning path are excluded from this evaluation. Indeed, in these locations, the shape of the path depends on the shape of the boundary and sharp turns might mandatorily exist in these areas (e.g. U turns in the left and right sides of

Table 2: Indices I_{curv} and I_{path} measured for different surfaces and trajectories.

	Option	I_{curv}	I_{path}
2-D path (Fig. 2)	Boundary curve	0	0.3995
Turbine blade (Fig. 10)	Furthest vertices	0.0239	0.0026
Channel (Fig. 10)	Boundary curve	0.0116	0.0042
Fillet (Fig. 10)	Boundary curve	0.9147	0
Surface (a) in Fig. 11	Furthest vertices	0	0.0183
Surface (b) in Fig. 11	Furthest vertices	0.0293	0.1892

the trajectory presented in Fig. 2). To solve this problem, either the polishing process can be stopped during the U turns or, if possible, an augmented surface can be used to generate the trajectory for the original surface by extrapolating this surface beyond its current boundaries.

Using I_{curv}, the algorithm evaluates the eligibility of the desired surface to be polished with AWJP technique. Then, using I_{path}, it estimates the performance of the paths generated using several options in AWJP process and finally selects the best one with minimum I_{path}. In Table 2 the values of these indices for the surfaces illustrated in Figs. 2, 10 and 11 are presented. As can be seen in this table, with the examples shown in Figs. 10 and 11, the values of I_{path} are far smaller than the one of the 2-D trajectory presented in Figs. 2. However, for Surface (b) in Fig. 11, it is larger than the others. This means that the existence of inner holes and concave boundaries can lead to curved paths along the trajectory which may decrease its efficiency in AWJP. On the other hand, with the path generated on the fillet, all the limits are properly preserved and $I_{path} = 0$ but the index I_{curv} is large and thus, a smaller nozzle with shorter W would preferably be used.

5 Conclusions

In this paper, a modular method was presented to generate scanning paths for automated abrasive waterjet polishing of free-form surfaces which are modeled by triangular mesh. The method was developed to produce trajectories with a constant offset distance preserved between adjacent curves. For this, the particular requirements of this polishing technique to be preserved by the path generation technique were determined. Then, to generate these paths, through several options, a reference curve was obtained. Afterwards, using geodesic distances in specific directions, the location of the points of the adjacent offset paths were calculated. Finally, the algorithm checked the continuity of the generated trajectory. If it passed beyond the boundaries of the surface, it was then divided into a set of continuous sub-trajectories. Finally, two indices were defined to quantify the effect of the shape of the free-form surface and the generated path on the uniformity of the distribution of the abrasive waterjet on the surface.

Acknowledgements The authors gratefully acknowledge the support from Alstom Energie et Transport Canada, Hydro-Quebec, and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Mao TF, Yang SC, Tsai FC, Hung JC, Yan BH (2010) Experimental investigation of abrasive jet polishing on the free-form machined surfaces of skd61 mold steel using sic particles. Mater Manuf Process 25(9): 965–973
2. Shafiei N, Getu H, Sadeghian A, Papini M (2009) Computer simulation of developing abrasive jet machined profiles including particle interference. J Mater Process Technol 209(9): 4366–4378
3. Li Z, Li S, Dai Y, Peng X (2010) Optimization and application of influence function in abrasive jet polishing. Appl Opt 49(15): 2947–2953
4. Atkar PN, Conner DC, Greenfield A, Choset H, Rizzi AA (2009) Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage. IEEE Trans Autom Sci Eng (USA) 6: 107–120
5. Feng-yun L, Tian-sheng L (2005) Development of a robot system for complex surfaces polishing based on cl data. Int J Adv Manuf Technol 26(9-10): 1132–1137
6. Hauth S, Richterich C, Glasmacher L, Linsen L (2011) Constant cusp toolpath generation in configuration space based on offset curves. Int J Adv-Manuf Technol 53(1-4): 325–338
7. Nagata F, Watanabe K, Izumi K (2001) Furniture polishing robot using a trajectory generator based on cutter location data. In: Proceedings ICRA. IEEE International Conference on Roboticsand Automation 1: 319–324
8. Park SC, Chung YC, Choi BK (2003) Contour-parallel offset machining without tool-retractions. Comput Aided Des 35(9): 841–849
9. Ryuh B, Park SM, Pennock GR (2006) An automatic tool changer and integrated software for a robotic die polishing station. Mech Mach Theory 41(4): 415–432
10. Sheng W, Chen H, Xi N, Chen Y (2005) Tool path planning for compound surfaces in spray forming processes. IEEE Trans Autom Sci Eng 2(3): 240–249

11. Yang DCH, Chuang JJ, Han Z, Ding S (2003) Boundary-conformed toolpath generation for trimmed free-form surfaces via coons reparametrization. J Mater Process Technol 138: 138–144
12. Yuwen S, Dongming G, Zhenyuan J, Haixia W (2006) Iso-parametric tool path generation from triangular meshes for free-form surface machining. Int J Adv Manuf Technol 28(7-8): 721–726
13. Tam HY (1999) Toward the uniform coverage of surfaces by scanning curves. Comput Aided Des 31(9): 585–596
14. Tam HY, Lui OCH, Mok ACK (1999) Robotic polishing of free-form surfaces using scanning paths. J Mater Process Technol 95(1-3): 191–200
15. Tsai MJ, Chang JL, Haung JF (2005) Development of an automatic mold polishing system. IEEE Trans Autom Sci Eng 2(4): 393–397
16. Surazhsky V, Surazhsky T, Kirsanov D, Gortler SJ, Hoppe H (2005) Fast exact and approximate geodesics on meshes. ACM Trans Graph 24(3): 553–560
17. Kout A, Muller H (2010) A framework for the generation of distance field based curves on triangular mesh surfaces for nc manufacturing. Tech rep, Department of Computer Science, Technische Universitt Dortmund
18. Bommes D, Kobbelt L (2007) Accurate computation of geodesic distance fields for polygonal curves on triangle meshes. VMV: 151–160
19. Yuen P, Mokhtarian F, Khalili N, Illingworth J (2000) Curvature and torsion feature extraction from free-form 3-d meshes at multiple scales. IEE Proc Vis Image Signal Process 147(5): 454–462
20. Xin SQ, Ying X, He Y (2012) Constant-time all-pairs geodesic distance query on triangle meshes. In: Proc. - I3D: ACM SIGGRAPH Symp. Interact. 3D Graph. Games, Costa Mesa, CA, USA: 31–38
21. Patrikalakis N, Bardis L (1989) Offsets of curves on rational b-spline surfaces. Engineering with Computers 5: 39–46
22. Patrikalakis NM, Maekawa T (2002) Shape Interrogation for Computer Aided Design and Manufacturing. Springer
23. Mitchell JSB, Mount DM, Papadimitriou CH (1987) The discrete geodesic problem. SIAM J Comput 16(4): 647–668
24. Galceran E, Carreras M (2013) A survey on coverage path planning for robotics. Rob Autom Syst 61(12): 1258–1276
25. Dong C, Wang G (2005) Curvatures estimation on triangular mesh. JZUS: 128–136
26. Szilvasi-Nagy M (2006) About curvatures on triangle meshes. KoG 10: 13–18