Int J Adv Manuf Technol (2016) 87:765–778
DOI 10.1007/s00170-016-8474-y

ORIGINAL ARTICLE

Identification and looping tool path generation for removing residual areas left by pocket roughing

Min Zhou [1] **· Guolei Zheng** [1] **· Shulin Chen** [2]

Received: 19 July 2015 / Accepted: 1 February 2016 / Published online: 27 February 2016

Abstract For pocket with complex geometry shape, lots of residuals will be left after pocket roughing. These uncut regions always are around sharp corners, bottlenecks, and sidewalls with small areas and soft edges. Additionally, as most of the stock material has been removed, the thin wall between two adjacent pockets tends to deform when machining the residuals. Thus, certain technological requirements such as starting from the soft edge, retaining down-milling, and keeping constant feed rate are needed to machine the unmachined materials. However, little research has been carried out on this problem. In this paper, to remove the various uncut areas left by the pocket roughing, residual regions are identified first by the rolling disk motion method. Then, looping tool paths are designed and computed for corner, bottleneck, and sidewall residuals respectively. The proposed tool path satisfies down-milling, G1 continuous, and progressive radial depth of cut with consideration of the soft edge. Finally, an example is rendered to validate that the advised algorithm can identify the remained areas correctly, and the generated tool path meets the special requirements of clean-up regions machining.

Keywords NC machining · Sidewall machining · Clean-up machining · Corner machining · Tool path generation · Looping tool path · Constant feed rate · Residual area identification

✉ Min Zhou
zhoumin.buaa@139.com

[1] School of Mechanical Engineering and Automation, Beijing University of Aeronautics and Astronautics, Beijng 100191, China

[2] Shenyang Aircraft Industry (Group) Corporation Ltd, Shenyang 110034, China

1 Introduction

With the development of high-speed tool-changing mechanisms and twin-spindle machine tools, the tool change time is increasingly playing a less dominant role in the total machining time [1], and multiple cutting tools are adopted to promote pocketing efficiency. After most of the stock materials are removed by larger cutters, materials near the sharp-angle, bottleneck, and sidewall are always left uncut due to the lower accessibility of large cutter. Then, slender end mill is usually adopted to clean up them. Therefore, residuals' removal is also a significant part of pocket machining. Bala and Chang [2] suggested using two cutting-tool sizes for machining a pocket and assumed that the unmachined areas (left by the bigger size of the cutting tool) can be removed by the smaller one by contouring the pocket. However, as the shape of the uncut region may very complex, contouring is not enough for kinds of situations. Compared to pocket roughing, there are some special technological requirements for residual machining:

(1) Down-milling is recommended to avoid the unbalanced wear or breakage of a tool. A detailed explanation of this can be found in reference [3].
(2) Constant fade rate is suggested to prevent the unwanted chatter of machine tool or tool damage. Varying radial depth of cut is generally encountered by the end milling tool during the entry into and exit from the corner. This leads uneven cutting force on cutters, which may result in the unbalanced wear of a tool and the chatter of machine tool [4].
(3) Keeping G1 continuous to avoid cutter deceleration at inflexion point.
(4) The tool should engage from the soft edge of the uncut distinct as the uncut area is always small and lacks of

enough advancing space for conventional pocketing operation.

Although lots of attentions have been paid to pocket roughing, such as multi-tool selection and tool path generation, as well as uncut free pocketing tool-paths generation, few researchers have investigated the removal of uncut areas caused by poor accessibility of roughing cutter (PARC) in pocket machining.

1.1 Literature review

1.1.1 Residual area identification

The uncut material may be resulted from the CPO-pocketing (contour parallel offset (CPO)) operation when the tool path interval is larger than the cutter radius [5], or caused by PARC after pocket roughing. The type of uncut areas lead by CPO can be identified as corner uncut, center uncut, and neck uncut [5]. Mansor et al. [6] further subdivided the corner uncut area into five different types, the center uncut area into four, and the neck uncut area into two through a systematic enumeration of the topology and geometry of the bisectors and window lines. They detected the uncut regions based on a pixel 2D cut simulation method suggested by Choi and Kim [5]. Park and Choi [7, 8] detected the uncut regions based on an observation that an uncut area exists if the raw offset for the tool envelop is self-intersecting. In the work of Lin et al. [9, 10], uncut regions are detected through geometry analysis [9] and offset method based on Boolean operation [10]. The problem is, for complex pockets, the geometry analysis method may fail [10]. To machine the uncut materials caused by PARC, Chang et al. [11] identified the clean-up regions by utilizing the byproducts (LIRs and GIRs) of the PWID offset algorithm [7]. However, residual along the sidewall is not considered as the clean-up region in their work. Additionally, the method is only fit to pocket without island. Makhe and Frank [12] subdivided the polygon at the islands necks and the boundary necks first. Next, the sub-polygon boundary is first offset by the tool radius inside the polygon, and then this new polygon is offset by the tool radius in the reverse direction. Finally, the unmachined area

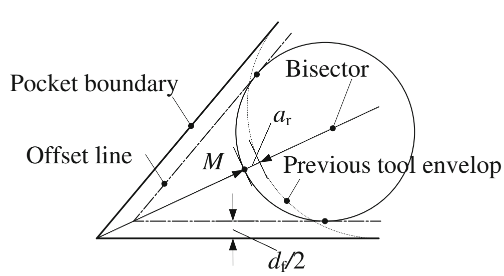


Fig. 1 Cutting arc being tangent with the offset line

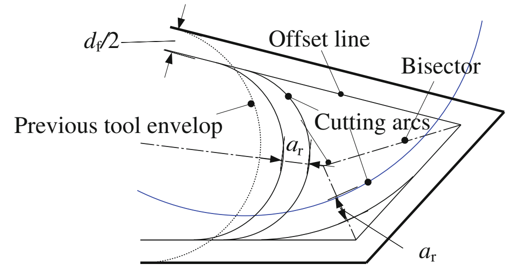


Fig. 2 Cutting arc being not tangent with the offset line

is simply equal to the difference between the original polygon boundary and the second offset. However, offset methods are prone to generate several problems [13] which need special considerations. To avoid the complexities in generating the Voronoi diagram, Veeramani and Gau [14, 15] proposed an approach for generating the Voronoi mountain of the unmachined areas. They constructed the feasible area extrusion by sweeping the feasible area of a given tool upward and outward at a 45° angle with the distance of extrusion height. Then, subtraction assures that the remaining solid possesses the 45° property to calculate the unmachined areas. However, when the boundary of the island consists of substantial arc spline segments, the Voronoi method tends to be non-effective as the 45° draft cannot be established successfully. As we know, once the machinable area is calculated, the uncut region can be computed by simple Boolean operation. Balasubramaniam [16] first generated the offsets from the Voronoi diagram for the contour in a particular slab of a pocket and obtained the limiting path of the center line of the tool. From the offset profiles, they reversed offset the path by the radius of the tool to finally get the reachable area of a tool in a slab. Chen and Fu [17] generated the tool path of a pocket by the medial axis transform (MAT) method. Then, they identified the region(s) covered by cutters by utilizing the tool paths and computed the regions with the function in the OpenGL graphic library.

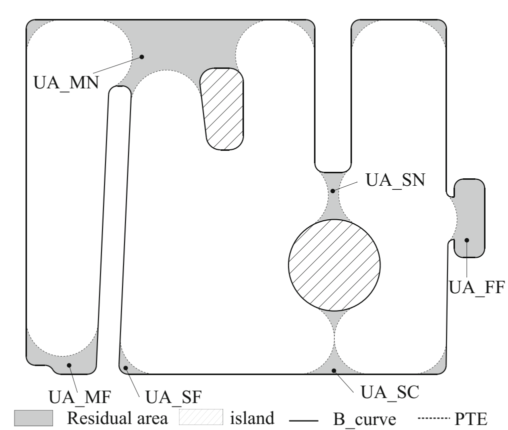


Fig. 3 Typical uncut areas in a pocket after roughing

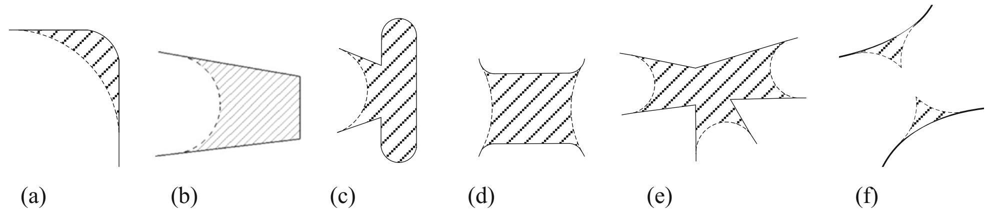


Fig. 4 Kinds of clean-up regions with different shapes. **a** UA_SF. **b** UA_MF. **c** UA_FF. **d** UA_SN. **e** UA_MN. **f** UA_SC

1.1.2 Tool path for residual area

To remove uncut regions caused by CPO, uncut free pocketing tool-paths generation based on PWID offset algorithm [8] and tool path compensations [6, 9, 10] based on Voronoi diagram [6], geometry analysis [9] and offset method [10] have received attention from several researchers. However, tool paths in these references do not satisfy the special requirements of residuals machining discussed above and are not fit for uncut regions caused by PARC. Additionally, neither conventional contour path pattern nor zig-zag path pattern meets the requirements. Because in both of the two ways, there is a continuous variation in radial depth of cut and frequent changes in magnitude as well as direction of the feed rate during the machining, which may lead machine tool jerk, excessive cutting force, and poor surface finish. Therefore, special tool path is asked for cleaning up the uncut areas caused by PARC. In our previous research [18], to machine the uncut bottleneck areas, new bounds are constructed by offsetting the soft edges with the radius and the hard edges with the diameter of the selected tool. However, since the residual area is expanded, the tool path length of the proposed method is increased and the radial depth of cut changes a lot during the machining. Thus, it needs to be improved. Choy and Chan [19–21] improved the conventional contour-parallel tool path pattern by appending single or double bow-like tool path segments at the corner position. By using the corner-looping-based tool path, corner material is removed progressively in several passes and cutter contact length can be controlled by adjusting the number of appended tool path loops. Based on Choy's research, Banerjee et al. [22], Rahman and Feng [23] improved the looping tool path to finish the corner of a pocket with consideration of physical constraints such as machining parameter and kinematics of the machine tool structure. The tool path loop in references [19–23] depends on the bisector line of a pocket corner. Figure 1 depicts the determination of the cutting arc of a tool path loop. The cutting segment of each loop is an arc whose center is on the bisector line of a corner. The arc passes through a given point M (M is determined by a_r, the radial depth of cut) and is tangent to the two offset lines. The two lines are obtained by offsetting the pocket boundary inwards by an amount equal to $d_f/2$, where d_f is the diameter of the finishing cutter. With this cutting loop, the radial depth is controlled under a permissible limit during machining the corner. However, when there is more than one sharp corner in an unmachined region, such tool path loop fails to work. As shown in Fig. 2, one of the corners in the remained area is an obtuse angle. In this case, it is possible that the cutting arc fails to be tangent with the offset lines. In Fig. 2, the blue cutting arc cannot be tangent to the offset lines while the distance between two adjacent cutting arcs is set as a_r. Therefore, this looping tool path based on the bisector line of corner is not applicable to irregular residuals removal. Sui et al. [24] proposed a combination strategy of corner-looping milling and clothoid curve to generate the semi-finishing tool path for pocket corner with simple shape. In the work of Chang et al. [11], the previous tool sweeping envelop, namely the tangential arc, is offset as the tool-path element to fill clean-up regions for sidewall machining. After generating the tool-path elements, one-way milling is chosen to link them to retain the down-milling. However, the tool path does not keep G1 continuous with many sharp turns and retractions, which leads a lower efficiency. Additionally, although the residual area is not big enough, the entry space for tool is not considered in their method. In references [4] and [25], trochoidal tool path (the continuous arcs) based on a MAT is adopted to machine elongated narrow regions and sharp corners. Trochoidal tool path is also a promising approach for high-speed machining of pockets due to its good continuity, high feed rate, and being smooth [25, 26]. But the trochoidal tool path increases the length of the whole tool path.

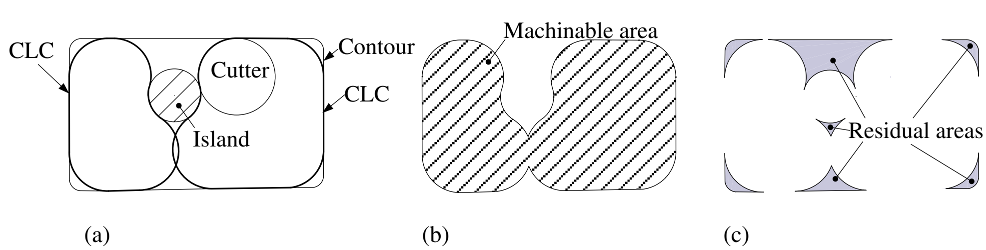


Fig. 5 The principle of rolling disk motion method for residual area identification. **a** Contact line chain (CLC). **b** Machinable area of a tool. **c** Residuals

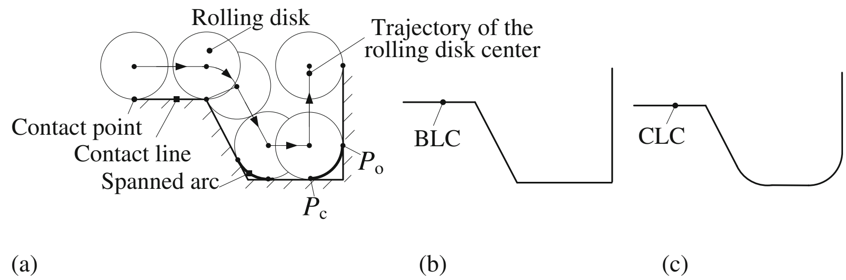


Fig. 6 Illustration of rolling disk method. **a** Rolling disk and elements of the CLC. **b** Basic line chain. **c** Contact line chain

1.2 Overview

As discussed above, both offset method and Voronoi method have some drawbacks to identify the residuals lead by PARC. And the current tool path generation measures are only applicable for simple corner residuals after pocket roughing. For various irregular residual areas caused by PARC in pocket machining, none of these studies is able to provide an effective and efficient machining strategy. To solve this problem, in this paper, rolling disk motion method is presented first to identify clean-up regions after pocket roughing. Then, with consideration of the soft edge, down-milling, G1 continuous, and progressive radial depth of cut, looping tool path is designed and generated to remove residuals around corners, bottlenecks, and sidewalls individually. In this paper, we suppose that the rough cutter can remove all the materials in its machinable area due to its accessibility and no uncut regions lead by CPO left.

The rest of this paper is organized as follows: the clean-up regions are classified and identified in Section 2; Section 3 introduces the looping tool path generation procedure in detail for various residuals. In section 4, implementation process of this approach is given, and an example is rendered to validate this presented approach; the final section is conclusion and discussion.

2 Residual areas classification and identification

2.1 Residual areas classification

With the appearance and development of rapid automatical tool change technology, pocket is usually machined by several cutters with different diameters to enhance the efficiency. Cutters with large diameters are used to remove most of the stock materials while small tools are applied to clean up the residuals left by pocket roughing. Due to the poor accessibility of large cutter, when the geometry shape of a pocket is complex, materials around sharp corners and bottlenecks tend to be left after roughing as Fig. 3 shows. According to the number of tool envelops and the geometry shape of a residual region, it can be classified six types as Fig. 4 depicts: (1) uncut area surrounded by single fillet and the previous tool envelop (PTE), UA_SF. According to the type of boundary curve, the UA_SF may be sharp corner, round corner, curvilinear corner, and combined corner. (2) Uncut area surrounded by multiple fillets and a PTE, UA_MF. If there is reflex vertex on the UA_MF's boundary, we call the UA_MF as complex. On the contrary, it'is simple. (3) Uncut area surrounded by furcate fillets and a PTE, UA_FF. The corner residual material is branched with only a sweeping envelop. (4) Uncut area surrounded by a single bottleneck and two PTEs, UA_SN. (5) Uncut area surrounded by multiple bottlenecks and PTEs, UA_MN. This kind of uncut area is bounded by more than two sweeping envelops. (6) Uncut area surrounded by side-wall curve and tool envelops, UA_SC. The boundary of UA_SC consists of sidewall boundary and at least two sweeping envelops.

As we know, the pocket boundary includes contour boundary and island boundaries. And the boundary of uncut area is made up of hard edge and soft edge. Hard edge coincides with the pocket boundary while soft edge is an arc which is the sweeping envelop of the previous tool and has no limitation to cutters. So, when cutter interferes with a hard edge, gouging happens. A hard edge is a line chain consisting of line, arc,

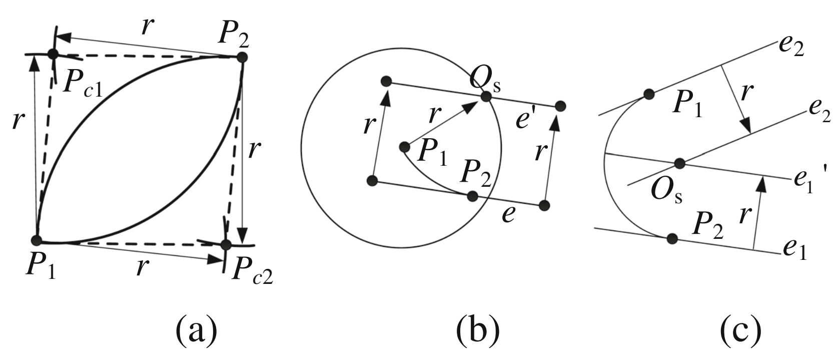


Fig. 7 Determination method of a spanned arc center: **a** Center of arc linking two basic points. **b** Center of arc linking basic point and basic line. **c** Center of arc linking two basic lines

curve, or their combination. And there are always two intersection points between a hard edge and soft edge(s). Due to the number of the hard edge and soft edge bounding a residual area, it can be classified as a corner residue, a bottleneck residue or a side residue:

Given the numbers of hard edge and soft edge which bound an uncut region are n_h and n_s, respectively, if $n_s = 1$, $n_h = 1$, the uncut region is a corner residue; if $n_s > 1$, $n_h \geq 2$, it is a bottleneck residue; if $n_s > 1$ and $n_h = 1$ are satisfied, the uncut area is a side residue. For example, uncut areas shown in Fig. 4 can be further classified as follows: a–c are corner residues; d, e are bottleneck residues; uncut areas depicted in Fig. 4f are side residues.

2.2 Residual areas identification/computation

In pocket machining, the accessible area of a tool is a region bounded by the sweeping envelop of the tool when it rolls along the pocket's boundary. The residual area can be obtained by subtracting the accessible area from the machining area. Therefore, rolling disk motion method is proposed to identify the unmachined areas.

Figure 5 illustrates the principle of rolling disk motion method for residual area identification. First, the moving cutter can be recognized as a rolling disk. When it moves along the pocket's profile, one or more closed contact line chain (CLC) can be made up of the boundaries touched by the disk and the spanned arcs. The region bounded by each CLC is a machinable area of the corresponding tool. Additionally, these machinable areas may be overlapping with each other. Thus, as Fig. 5b shows, regions bounded by each CLC should be combined to construct the accessible area of the given tool. Finally, the residual area can be obtained by subtracting the accessible area from the machining area, as depicted in Fig. 5c. Where, the machining area is bounded by the basic line chain (BLC) which is made up of a contour loop and several island loops.

The key steps of the rolling disk motion method are as follows: (1) establishment of the BLC: given the contour and island boundaries, establish the corresponding basic point list, and the basic line list; (2) computation of the spanned arc: given the basic point list and the basic line list, calculate all the spanned arcs between two basic lines, between basic point and basic line, and between basic points in order; (3) judgment of the validity of a spanned arc: identify and keep the effective spanned arcs according to the validity rules; (4) computation of the CLC: divide the basic line by the endpoints of valid spanned arcs and decide contact edges due to the parity of the points' numbers; (5) construct the CLC: construct the CLC by linking the contact edges and the spanned arcs in order; (6) reconstruct the CLC: recombine the CLCs which intersect each other; (7) calculation of the residual areas: machinable area is the union of areas bounded by the CLCs. Thus, the

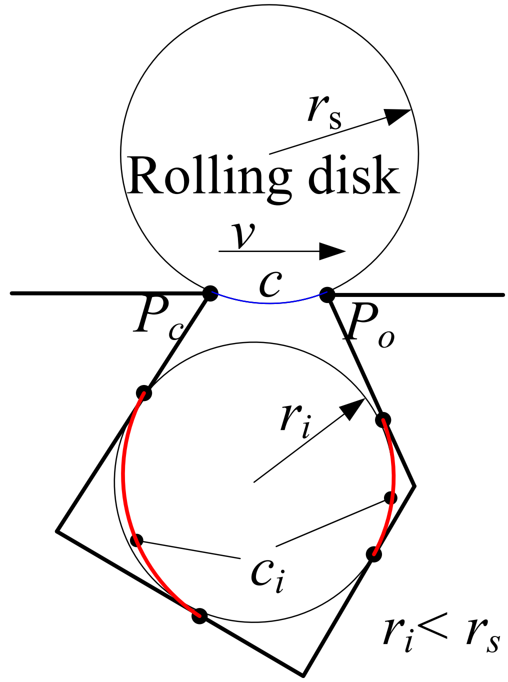


Fig. 8 Judgment by the entry rule

residual regions can be obtained by subtracting the machinable area from the machining area. The illustration of rolling disk method is shown in Fig. 6.

1. **Modeling of the BLC**

   The lines which make up of contour boundary or island boundary are called basic lines. Similarly, the points on the boundaries are basic points. Thus, a BLC can be established by putting all the basic lines and basic points of a contour loop or an island loop into a list. Furthermore, according to the points and lines on the BLC, the corresponding basic point list and basic line list can be constructed. trajectory
2. **Calculation of the spanned arc**

   In Fig. 6a, when the rolling disk is moving along the basic line, the point on the line touched by the disk is defined as contact point, represented as P_c; the point which breaks the continuity of rolling is called obstruction

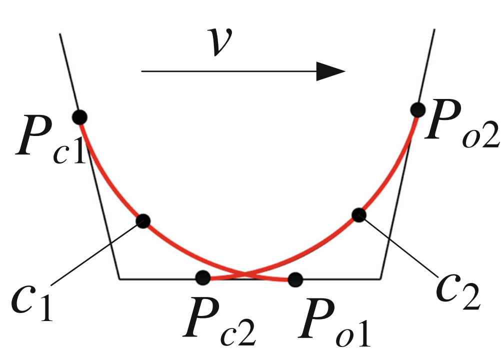


Fig. 9 Non-intersection rule

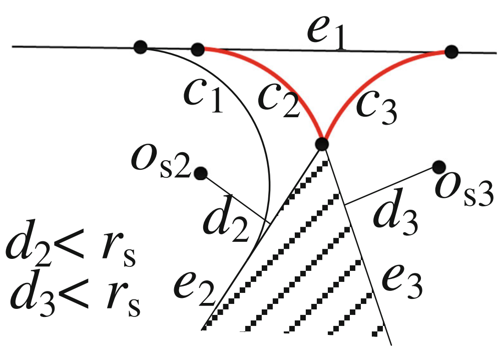


Fig. 10 Distance rule

point, represented as P_o; the arc which links the P_c and P_o with radius of $d_f/2$ is defined as spanned arc. Spanned arc is part of the CLC and represented as $c(P_c, P_o, O_s, r_s)$. Where, P_c is contact point, P_o is obstruction point, O_s is the center of the spanned arc and r_s is the radius of the spanned arc.

Due to the representation of a spanned arc, the key point to calculate a spanned arc is to find its center. Figure 7 illustrates the calculation of a spanned arc center for three different spanned arcs. The arc in Fig. 7a links two basic points P_1 and P_2. To find the center, draw two arcs centered at P_1 and P_2 with radius r_s, respectively. Then, the intersection points P_{c1} and P_{c2} of the two arcs are just the spanned arc centers. Of course, only one of them is valid. P_1 and P_2 are P_c and P_o, respectively. The arc in Fig. 7b is between basic point P_1 and basic line e. Draw a circle C centered at P_1 with radius r_s. Meanwhile, offset e toward to the rolling disk by amount r_s to get e'. Then the intersection point of e' and C is just the O_s. P_1, and the tangential point P_2 between e and the spanned arc are P_c and P_o, respectively. Figure 4c shows the determination of the O_s for spanned arc between two basic lines. Offset the basic lines e_1 and e_2 toward to the disk by amount of r_s to get e_1' and e_2'. Then, the intersection point of e_1' and e_2' is just the O_s. And the tangential points between the spanned arc and e_1, e_2 are P_c and P_o, respectively.

3. **Judgment of the validity of a spanned arc**

Some of the spanned arcs are invalid as they may interfere with the pocket's boundary or intersect with each other as Figs. 5 and 6 show. Therefore, it'is necessary to

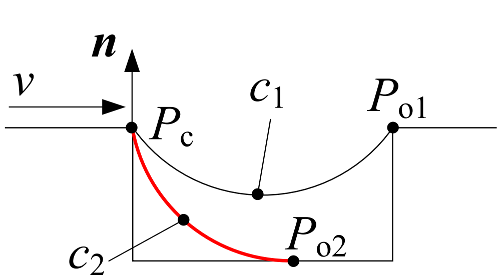


Fig. 11 Unique rule

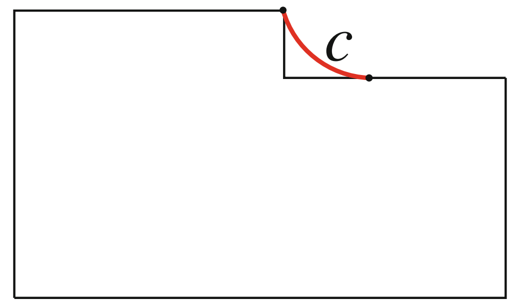


Fig. 12 Inside rule

remove those invalid arcs. The following rules are established to judge whether a spanned arc is valid or not.

Rule 1 (**Entry rule**) Given a spanned arc c and its corresponding P_c and P_o,, r_s is the radius of c,, l is the basic line chain between P_c and P_o, spanned arc c_i links lines or points of l,

(1) If the distance between centers of c_i and c satisfies $d < r_s$, then, c_i is invalid.
(2) If the distance between centers of c_i and c satisfies $d \geq r_s$, and $r_i < r_s$, where r_i is the radius of the maximum inscribed circle of l, then c_i is invalid (as shown in Fig. 8).

Rule 2 (**Major arc rule**) Valid spanned arc must be major arc.

Rule 3 (Non-intersection rule) Spanned arcs c_1 and c_2 intersect with each other, if

(1) there is only one intersection point,
(2) P_{c1}, P_{o1}, P_{c2}, P_{o2} are different from each other,
(3) the distance between the centers of c_1 and c_2 satisfy $d < r_s$,

then c_1 and c_2 are invalid (as shown in Fig. 9).

Rule 4 (Distance rule) The distance between the center of a spanned arc and an element on the boundary must be equal to or larger than r_s.

As shown in Fig. 10, there are three spanned arcs linking to e_1. But the distance between O_{s2} and e_2 is smaller than r_s. According to Rule 4, spanned arc c_2 should be excluded. Similarly, c_3 is also invalid and only c_1 is effective.

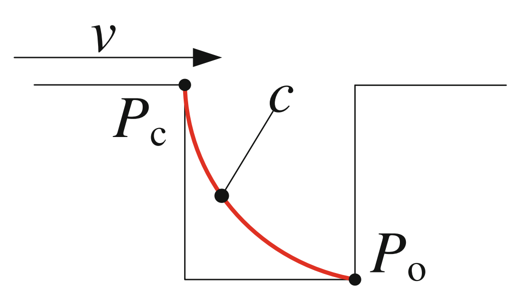


Fig. 13 Convex point rule

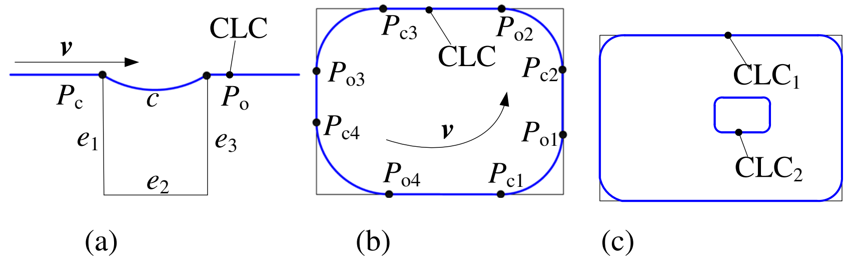


Fig. 14 Construction rules for CLC. **a** Exclude rule. **b** Parity rule. **c** Loop rule

Rule 5 (**Unique rule**) If more than one spanned arc c_i pass through contact point P_c, along the rolling direction $\boldsymbol{v}$, the leftmost c_i is valid. Similarly, if more than one spanned arc c_i pass through obstruction point P_o, the rightmost is valid.

Sometimes, there may be some small and narrow local regions in the pocket. The rolling disk cannot enter into these areas since the bottlenecks and areas are not big enough. Then, the spanned arc along the BLC belongs to these small and narrow local regions are invalid. As is show in Fig. 10, the red arc is invalid. The relative locations of the spanned arcs can be determined by contact point P_c and obstruction point P_o. Take an example of Fig. 11 as follows.

Given a vector $\boldsymbol{n}$ is perpendicular to vector $\boldsymbol{v}$, and $\boldsymbol{n}$ is consistent with $\boldsymbol{v}$ after rotating 90° clockwise. $\boldsymbol{n}_1$ is the vector with P_c as the start point and P_{o1} as the end point while $\boldsymbol{n}_2$ with P_c as the start point and P_{o2} as the end point. If the angle $<\boldsymbol{n}, \boldsymbol{n}_1>$ is smaller than the angle $<\boldsymbol{n}, \boldsymbol{n}_2>$, then the spanned arc c_1 is on the left of c_2. Where, $<\boldsymbol{n}, \boldsymbol{n}_1>$ represents the angle between the two vectors.

Rule 6 (**Inside rule**) The center of a valid spanned arc is inside of the machining area.

Because the center of a valid arc is on the trajectory of the rolling disk center, if the center of spanned arc is outside of the machining area, namely outside of the pocket, the corresponding arc must be invalid (red arc in Fig. 12).

Rule 7 (**Convex point rule**) One of the necessary conditions for two vertices linking a spanned arc is that the two vertices are both reflex.

The explanation of this rule is similar to Rule 5. As Fig. 13 shows, c is invalid.

4. Construction of the CLC

Contact line is the basic element of CLC. After determination of the valid spanned arc, contact line should be extracted and serialized. According to the definition and geometrical characteristic of the spanned arc, some of the edges on the boundary loop cannot construct the CLC. As shown in Fig. 14a, c is a valid arc; e_1, e_2, and e_3 in BLC are not contact line. Therefore, several rules are given out to determine the contact line.

Rule 8 (Exclude rule)) Along the rolling direction $\boldsymbol{v}$, the edges between P_c and P_o do not make up of the CLC if both P_c and P_o are on the contour loop l_c or both of them are on island loop l_i.

Take an example of Fig. 14a, edges e_1, e_2, and e_3 are not contact lines and should be excluded from the CLC. Besides, along the rolling direction, when the rolling disk m_g leaves an obstruction point P_o and then arrives at the next spanned arc, it must reach the P_c of the spanned arc first. Therefore, the P_c and P_o appear alternatively along $\boldsymbol{v}$ in a CLC. Furthermore, the edges between P_o and P_c are contact lines along $\boldsymbol{v}$.

Rule 9 (**Parity rule**) Along $\boldsymbol{v}$, the spanned arcs are numbered based on the parity of P_c and P_o.

As is shown in Fig. 14b, the P_c and P_o appear

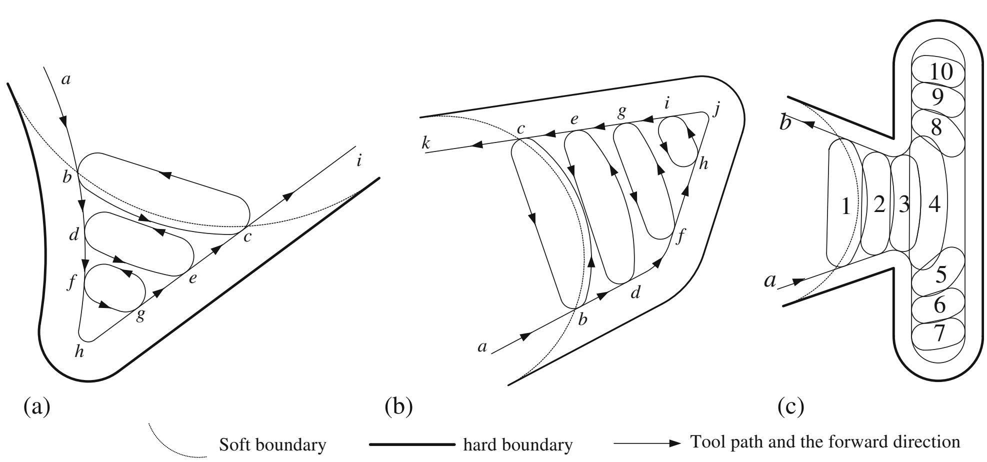


Fig. 15 Tool path for kinds of corner residuals. **a** Tool path for UA_SF. **b** Tool path for UA_MF. **c** Tool path for UA_FF

alternatively. And the numbers of P_c and P_o for each arc are the same. Therefore, when there is at least one valid spanned arc on the boundary loop or island loop, CLC can be constructed by traversing all the valid spanned arcs with Rule 8. However, when there'is no spanned arc on a loop, the loop is also able to construct a CLC.

Rule 10 (**Loop rule**) A closed loop without any valid spanned arc on it is a CLC.

As plotted in Fig. 14c, CLC_2 is just the island loop.

After recognizing the spanned arcs and contact lines correctly, CLCs can be constructed by linking them in sequence.

5. **Reconstruction of the CLC**

   When different CLCs intersect with each other (as illustrated in Fig. 5a), it'is necessary to reconstruct them to compute the residual areas correctly.

   Suppose that a serious CLCs $C_1, C_2,\ldots,C_n$ ($n\geq 0$) are obtained by rolling disk motion method, and $C_1', C_2', \ldots, C_m'$ ($0\leq m\leq n$) are the unions of these CLCs, represented by operator f as $(C_1', C_2',\ldots,C_m')=f(C_1, C_2,\ldots,C_n)$. The approach to f is seeking the union set of $C_1, C_2,\ldots,C_n$.

6. **Residuals determination**

   Suppose the planar districts bounded by $C_1', C_2',\ldots,C_m'$ are $A_{m1}, A_{m2},\ldots,A_{mm}$, then the machinable area A_m and the residual area A_r can be determined as

$$A_m = \bigcup_{i=1}^{n} A_{mi}\ ,$$
$$A_r = A - A_m.$$

Where, the A is the machining area.

3 Tool path generation procedure

Obviously, the spanned arc is part of the uncut area's boundary. Meanwhile, the spanned arc is also the sweeping envelop of the previous tool. After the clean-up regions determination, an uncut region can be identified as a corner residue, a bottleneck residue or a side residue due to the number of spanned arcs in an uncut area. As the area of an uncut material is usually small, to retain progressive radial depth of cut, the sweeping envelop of the previous tool is defined as the soft boundary which has no limitation for cutter. Thus, the cutter can enter into the residual area from the soft boundary. Then, tool path loop is constructed to remove the uncut regions. Tool path generation procedures for corners, bottlenecks and side residuals are given respectively as follows.

3.1 Tool path for corners

Tool path loops are proposed to machine the corner residuals. As Fig. 15 shows, (a), (b) and (c) illustrate the tool paths for

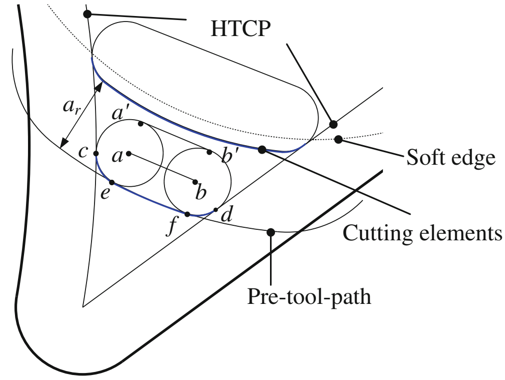


Fig. 16 Tool path loop generation for corners

UA_SF, UA_MF and UA_FF, respectively. The tool path in Fig. 15a is *ab*➔*bc*➔*cb*➔*bd*➔*de*➔*ed*➔*df*➔*fg*➔*gf*➔*fh*➔*hi*; the path in Fig. 15b is *ab*➔*bc*➔*cb*➔*bd*➔*de*➔*ed*➔*df* ➔*fg*➔*gf*➔*fh*➔*hi*➔*ih*➔*hj*➔*jk*; and the sequence of loops in Fig. 15c is from 1 to 10. As the generation methods of tool path for different types of corner residuals are the same, here,, only the procedures for UA_SF is introduced to illustrate the tool path generation for corners. Tool path for corners consists of tool path loop and hard critical tool path (HCTP). In Fig. 15a, *ah* and *hi* sections are HCTP while *bc*➔*cb* is a tool path loop (TPL). Where, HCTP is obtained by offsetting the hard boundary by a mount of r_f, the radius of the cutter. The following part will describe the detailed procedures of TPL generation.

A TPL is composite of the main cutting path, return path, and linking arcs. As shown in Fig. 16, they are *ef*, *a'b'*, *ea'* and *fb'*, respectively. Part of the linking arcs joins cutting (*ec* and *fd* in Fig. 16) while return path *a'b'* do not remove any material. The main cutting path and the cutting part of the linking path are called cutting elements (as blue lines indicated in Fig. 16).

1. **The main cutting path (The pre-tool-path)**

   As shown in Fig. 16, a pre-tool-path can be obtained by offset the cutting elements of the last TPL with amount of a_r. Then, linking arcs are added to trim the pre-tool-path to get the main cutting path. The main cutting path of the first loop can be computed by offsetting the soft edge: if $a_r<d_T/2$, offset the soft edge outward by $(d_T/2-a_r)$; if $a_r>d_T/2$, offset the soft edge inward by $(a_r-d_T/2)$; if $a_r=d_T/2$, the main cutting path of the first loop coincides

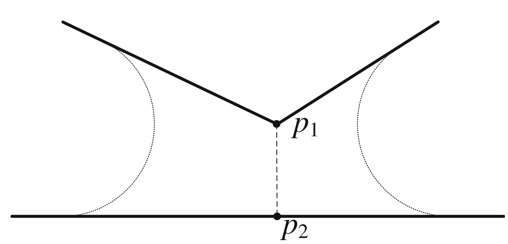


Fig. 17 A bottleneck residue area being divided into two corners

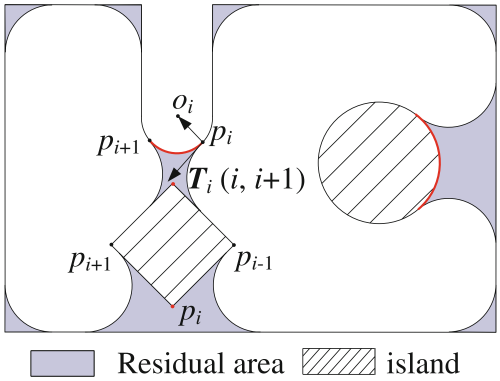


Fig. 18 Determination of reflex points and arcs on residuals' bounds

with the soft edge. In Fig. 16, the main cutting path of the first loop is just in case of $a_r > d_T/2$.

2. **The centers of linking arcs**

   Given the radius of the linking arc is r_1, obviously, the linking arc should be tangent with the HTCP and the pre-tool-path. According to the three restriction conditions, the centers of arcs can be determined. As plotted in Fig. 16, points *a* and *b* are centers of linking arcs, points *c*, *d* and *e*, *f* are the corresponding tangential points.

3. **The return path**

   As depicted in Fig. 16, the return path *a*'*b*' can be determined by offsetting the segment *ab* with r_1.

4. **Tool path loop generation**

   After the pre-tool-path, linking arcs and return path are calculated, a TPL can be obtained by the following operations.

   (1) Trim the pre-tool-path at points *e* and *f* to get the main cutting path *ef*.
   (2) Trim the linking circles at points *e*, *f* and *a*', *b*' to get linking arcs *a*'*e* and *b*'*f*.

Then, linking arcs *a*'*e*, *b*'*f*, the main cutting path *ef* and return path *a*'*b*' constitute a closed loop, i.e., a TPL.

3.2 Tool path for bottlenecks

To make the tool path down-milling and continuous, a bottleneck residual area is subdivided into several generalized corner residues based on the reflex points on the boundary. Then, the methods of tool path generation for corners can be applied for bottlenecks.

3.2.1 Generalized corners construction

The existence of reflex points on bottlenecks' boundaries always results in some local area becoming narrow, which leads the bottleneck region get inaccessible for large tools. As the tool path would change its feed direction and slow down when it meets a reflex point, to minimize the number of corners, the bottleneck area is split by the bottleneck line based on reflex points. It is shown from Fig. 17 that a bottleneck residue area is divided into two corners by bottleneck line p_1p_2, where p_1 is a reflex point on the boundary. Such a corner residual area constructed by separator is called generalized corner residue area, and the separator p_1p_2 is called virtual edge.

1. **Reflex point determination**

   A point may be reflex for a limitary region and convex for its neighboring region. To determine whether a vertex/an arc is reflex or not, cross product is applied. Suppose serial vertices $p_1, p_2, \ldots, p_i, \ldots, p_n$ are in a counter-clockwise sequence on contour loop and in clockwise direction on island loop.

   For vertex connecting two line segments, suppose vectors $\overrightarrow{p_{i-1}p_i} \times \overrightarrow{p_ip_{i+1}} = \boldsymbol{p_i}$, based on the right-hand rule,

   (1) if vector $\boldsymbol{p}_i$ is outside, p_i is a convex point;
   (2) if vector $\boldsymbol{p}_i$ is inside, p_i is a reflex point;
   (3) if vector $\boldsymbol{p}_i = \boldsymbol{0}$, p_i is a tangent point.

   For vertex connecting two circular arcs or connecting one line segment and a circular arc, vector $\overrightarrow{p_ip_{i+1}}$ in above equation should be replaced by $\boldsymbol{T}_i$ (i, $i+1$), the tangent vector of arc p_ip_{i+1} at point p_i. And the direction of $\boldsymbol{T}_i$ (i, $i+1$) is in line with the direction of boundary (shown in Fig. 18). If p_i connects two arcs, $\overrightarrow{p_{i-1}p_i}$ should also be replaced by $\boldsymbol{T}_i$ (i-1, i), the tangent vector of arc $p_{i-1}p_i$ at point p_i.

   For an arc, cross product $\boldsymbol{T}_i \times \overrightarrow{p_io_i}$ is employed, where o_i is the center of an arc. If the result of $\boldsymbol{T}_i \times \overrightarrow{p_io_i}$ is outside, arc p_ip_{i+1} is a convex arc. And if it is inside, p_ip_{i+1} is a reflex arc. Obviously, if the

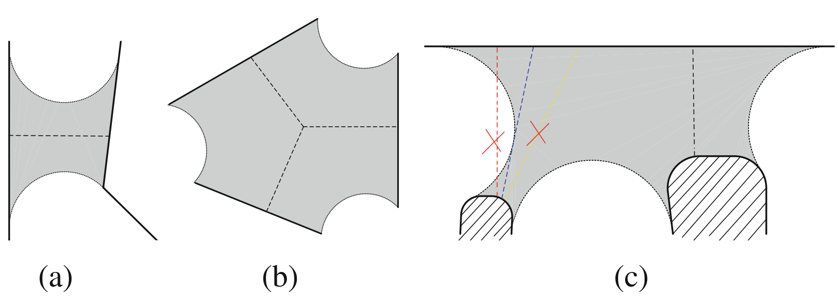
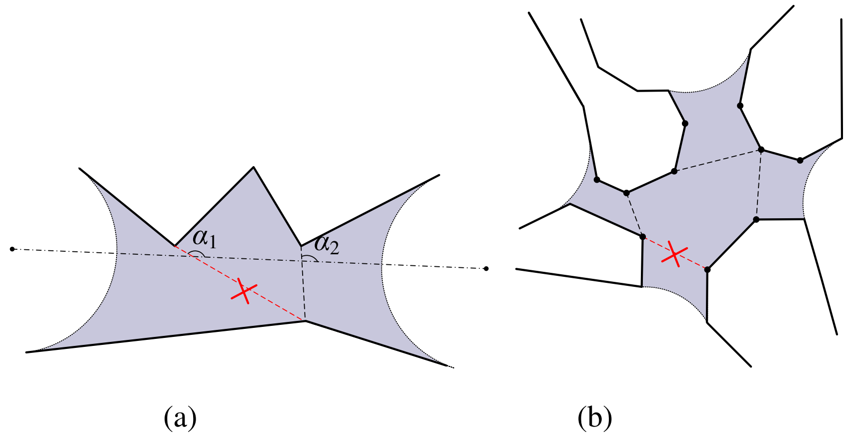


Fig. 19 Separator supplement: **a** Additive separator for bottleneck area with two soft edges. **b** Additive separators for bottleneck area with multiple soft edges. **c** Additive separator when bottleneck line intersecting with the soft edge

Fig. 20 Unwanted separator deletion when $n_v = n_s$. **a** Rule 11. **b** Rule 12



contour is a circle, the whole boundary is a convex arc. If the boundary of island is a circle, the whole boundary is a reflex arc.

Essentially, for a residual area, it'is unnecessary to judge a point which linking soft edge and hard edge is reflex or not, because such a point is unavailable to establish a separator. Figure 18 illustrates valid reflex points and reflex arcs of a pocket's boundaries in red.

2. **Bottleneck line determination**

Given hard edges E_i and E_j are linked by a soft edge, p_i^s is the reflex point on E_i while P_j^t is the reflex point on E_j, where $s = \overline{1, n_i}$, $t = \overline{1, n_j}$, $n_i \geq 0$, $n_j \geq 0$. The distance between p_i^s and P_j^t is represented as $d_r^{(s,t)}$.

1) If $n_i \neq 0$, $n_j \neq 0$, the bottleneck line corresponding with p_i^s is a line linked p_i^s and P_j^t subject to min ($d_r^{(s,t)}$), $t = \overline{1, n_j}$.
2) If $n_i \neq 0$, $n_j = 0$, the bottleneck line corresponding with p_i^s is defined as the shortest-distance line segment between p_i^s and E_j.
3) If $n_i = 0$, $n_j = 0$, extra separator should be added to split a bottleneck area as shown in the following part.

3. **Bottleneck residuals subdivision**

As the tool enters residuals from the soft edge, the number of sub-areas should be equal to n_s. Suppose the number of virtual edges is n_v after subdivision, the n_v should satisfy $n_{s-}1 = n_v$. If the equation fails to be satisfied when reflex points are applied to construct the virtual edge, separators should be deleted or supplemented.

Rule 11 (The minimum angle rule) When the residual area is a UA_MN, suppose l is the line linking the two centers of its two soft edges, and the angles between l and bottleneck lines $l_1, \ldots, l_i, \ldots, l_s$ for E_k and E_j are $\alpha_1, \ldots, \alpha_i, \ldots, \alpha_s$, if $\alpha_k = \min(|\alpha_i - 90°|)$, $i = 1, 2, \ldots, s$, $k \in \{1, 2, \ldots, s\}$, then only the bottleneck line l_k is kept.

Rule 12 (Minimum reflex points rule) When the residual area is a UA_MN, $n_r > n_s$ and $n_v = n_s$, the virtual edge of the sub-region which contains soft edge and the minimum reflex points should be removed.

As shown in Fig. 20, the red dotted lines should be deleted according to Rule 11 and Rule 12, respectively.

1) Separator supplement

Suppose the number of reflex points is n_r on the boundary of a bottleneck area while the number of corresponding bottleneck lines is n_n. Obviously, if $n_r = 0$, then $n_n = 0$. At this time, separator should be added according to other rules instead of reflex elements. As the separator is perpendicular to the hard edge, the number of the TPLs is the least. It'is best to minimize the difference between the adjacent angles whose vertex links the hard edge and the separator. For a bottleneck area with only two soft edges, the additive separator is perpendicular to one of the hard edges as shown in Fig. 19a. For a residual region with multiple soft edges and without any reflex points, the center of the region is applied as endpoint of separators (shown in Fig. 19b). The other endpoint of a separator is on a hard edge and the separator is perpendicular to the hard edge. If the bottleneck line intersects with a soft edge, a new separator is preferred to replace the bottleneck line. The new

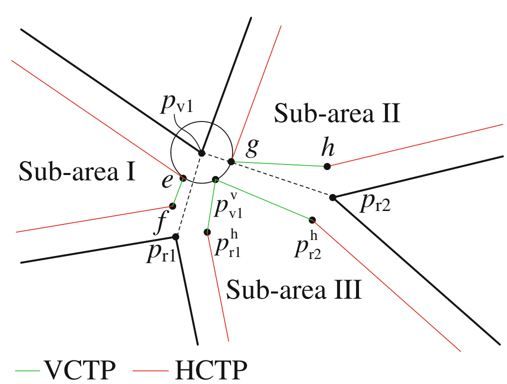


Fig. 21 Illustration of the generation for VCTP

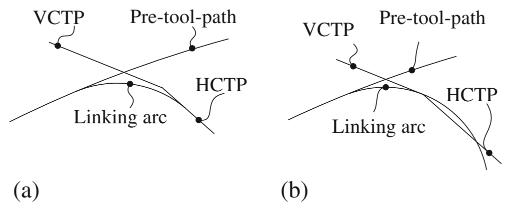


Fig. 22 Non-intersection rule for linking arc. **a** No intersection. **b** Intersection with the HCTP

separator is advised to be tangent with the soft edge to minimize the difference between the angles discussed above. As depicted in Fig. 19c, the blue separator is better than the yellow one to be added as the virtual edge instead of the red one.

2) Unwanted bottleneck line deletion

When $n_n \geq n_s$, if all the bottleneck lines are used as separators, some of the sub-regions may be bounded without soft edge. Meanwhile, $n_{s-1} \neq n_v$. Therefore, unwanted bottleneck lines should be deleted when $n_n \geq n_s$. As the reflex points may increase the number of TPLs, deletion principles are set to arrange the reflex points in every sub-area evenly.

3.2.2 Tool path generation for bottlenecks

Once the bottleneck residuals are constructed as generalized corners, the tool path generation technique for corners can be used for bottleneck residuals. However, as the virtual edge of generalized corner residue has no limitation for a cutter, special measures are asked for tool path generation of bottleneck residuals.

The tool path determined by the virtual edge is defined as virtual critical tool path (VCTP). The VCTP is special for bottleneck residuals.

Suppose the numbers of virtual edges and hard edges of an uncut bottleneck area are n_{v_s} and n_{h_s}, respectively. The VCTP can be determined as follows.

1) $n_{v\_s} = 1$

If $n_{v\_s} = 1$, the VTCP is the line segment whose endpoints coincide with the endpoints of two HTCPs. The segments *ef* in sub-area I and *gh* in sub-area II shown in Fig. 21 are VTCPs.

2) $n_{v\_s} > 1$

If $n_{v\_s} > 1$ and there'is no intersection point of virtual edges, or the intersection point is not on HCTP, then, the virtual edges can be used as the VCTP. For example, the VCTPs for Fig. 19 are the separators. Otherwise, special VCTP is needed.

Suppose that the vertex p_{vi} links at least two virtual edges while vertex p_{rj} links only one virtual edge, the bisector of $\angle p_{vi}$ intersects with circle C_{vi} at point p^{v}_{vi}, where, $\angle p_{vi}$ is an angle between two virtual edges and C_{vi} whose diameter is d_f. Additionally, the corresponding point of p_{rj} on the HCTP is p^{h}_{rj}. Then, the VCTP consists of line segments $p^{h}_{rj}p^{v}_{vi}$ and $p^{v}_{vk}p^{v}_{v(k+1)}$. As indicated in Fig. 21, the VCTP of sub-area III is made up of segments $p^{h}_{r1}p^{v}_{v1}$ and $p^{v}_{v1}p^{h}_{r2}$.

Furthermore, the linking arc of a TPL is allowed to be tangent with a HCTP or VCTP (Fig. 22a) rather than intersect with any of them (Fig. 22b). In Fig. 20b, the linking arc is tangent with the VCTP while it also intersects with the HCTP. Thus, scheme plotted in Fig. 20a is available.

3.3 Tool path for side residuals

Besides the bottleneck residue, the bottleneck areas of a pocket may lead side uncut areas (as the UN_SC plotted in Fig. 3). Figure 23a shows that when the length of the bottleneck line is bigger than $2d_f$, the uncut material cannot be cleaned up by only a tool path in sidewall finishing. Thus, in order to retain down-milling and progressive radial depth of cut, it'is necessary to append some loops to the tool path for side residue. The tool path for uncut material in Fig. 4f is provided in Fig. 23b. The path is *ab*➔*bc*➔*cb*➔*bd*➔*de*➔*ed*➔*df*. Where, the crucial techniques different from the tool path generation for corner and bottleneck residuals are given as follows.

1. **The location of TPL**

The tool path for side residue consists of HCTP and TPL. The generation method of HCTP for sides is as same as it for corners. And the loops are tangent with HCTP and

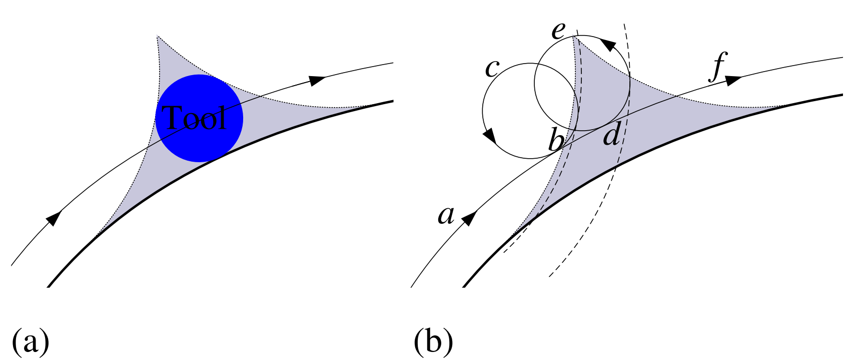


Fig. 23 Tool path for side uncut region. **a** Uncut material on side. **b** TPL for side residue

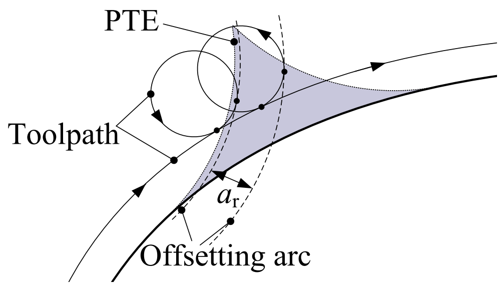


Fig. 24 Generation of tool path for side uncut region

offsetting arcs which is obtained by offsetting the soft edge. As plotted in Fig. 24, the distance between adjacent offset arcs is a_r. The method for determination of the first offsetting arc is the same as that of the pre-tool-path of the first loop for corners (see Section 3.1).

2. **The diameter of TPL**

   The TPL for side residue is a circle being tangent with HCTP and offsetting arc. Suppose its diameter is d_L, the diameter of the previous tool is d_D, the centers' distance between the two soft edges bounding the side residue is d_d, then, $d_L = \frac{1}{2}\left[\left(d_D - \sqrt{d_D^2 - d_d^2}\right) - d_f\right]$.

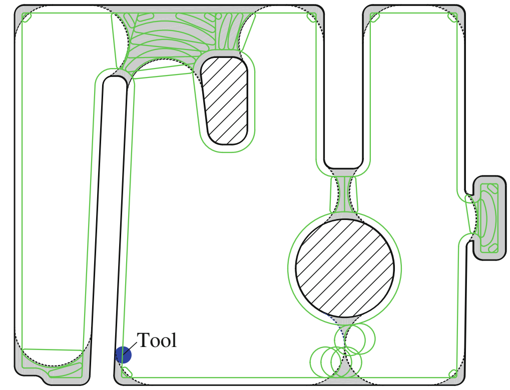


Fig. 26 Tool path for removing clean-up regions by the proposed approaches

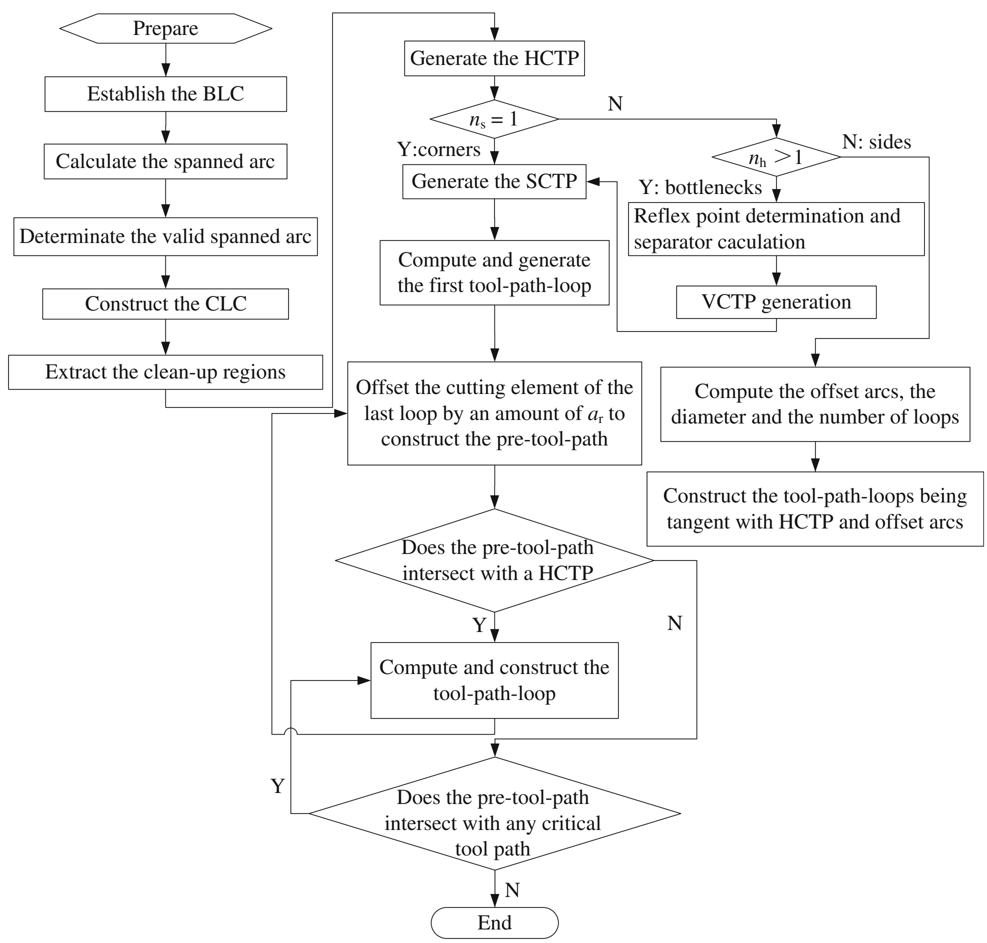
3. **The numbe of TPL**

   Suppose n_l is the number of TPL, then,

$$n_l = \mathrm{int}\left(\left[\frac{1}{2}\sqrt{d_D^2 + 4d_d^2 - 4d_d\sqrt{2d_f(d_D - d_f)}} - \frac{1}{2}d_D \pm \left(a_r - \frac{d_f}{2}\right)\right]\Big/ a_r\right) + 1$$

where, if $a_r \le d_f/2$, plus sign is selected; if $a_r > d_T/2$, minus sign is adopted. The equation above is applicable for linear hard edge. For other cases, geometry computation is enough to find the corresponding n_l.

Fig. 25 Flow chart of looping tool path generation for clean-up regions



4 Implementation

The algorithm for residual area identification and looping tool path generation is provided in Fig. 25. To validate the advantages of the proposed approach, an example shown in Fig. 3 with all kinds of residuals is given. The green lines shown in Fig. 26 reveal the tool path for removing the uncut areas of this example. In this example, the diameter of the previous tool is 40 mm, and that for the semi-finishing/finishing tool and linking arc is 8 and 3 mm, respectively. Additionally, the radial depth of cut a_r is 4.8 mm.

From Fig. 26, we can see that for UA_SN, the tool enters into the area from the two soft edges; for UA_MN plotted in the graph, it is split into three generalized corners since there are three soft edges. Besides, as one of the bottleneck lines intersect with a soft edge, a new separator which is tangent with the soft edge is added to replace the bottleneck line. For UA_FF in the example, five loops are generated to machine it by the tool entering from the soft edge. Additionally, the cutting element of each loop, instead of the soft edge, is offset to construct the next loop according to our method. In this way, the proposed tool path loop is flexible well to uncut areas with various geometry shapes. For instance in Fig. 26, the axes of loops for UA_MN and UA_FF are furcate to adapt to their shapes. Furthermore, as maximum distance between any two adjacent cutting elements is no bigger than a_r, the cutting force will not change a lot during the machining. And the TPL always retains down-milling and G1 continuous during the machining.

5 Conclusion and discussion

To remove the clean-up regions after pocket roughing, the residual areas are classified according to their geometry shapes and the number of soft edges. Then, a rolling disk method is proposed to identify the various uncut regions, and tool path loops are designed for different kinds of residuals. Compared to the previous research by others, the unique features of the approaches proposed and the main advantages of the techniques over them can be concluded as follows: (1) the presented rolling disk motion method for residual areas identification avoids adopting offset method or Voronoi diagram which have some potential problems [13] and need special considerations; (2) the advised looping tool path is available for kinds of complex corners, bottleneck residuals, and side uncut areas; (3) the rendered tool path retains down-milling and G1 continuous; (4) the practical radial depth of cut is always smaller than the theoretical depth with little variation in machining; (5) as the cutting element of each loop is offset to construct the next loop iteratively, it breakthroughs the limitation of the geometry shape of residuals to tool path generation. Thus, the presented looping tool path is able to remove all kinds of residual regions.

Acknowledgments This research was supported by the National Science and Technology Major Project (NSTMP) under Grant No. 2012ZX04010051 and the Union Aerospace Science Foundation under Grant No. 2014028026.

References

1. Zhang Y, Ge L (2009) Selecting optimal set of tool sequences for machining of multiple pockets. Int J Adv Manuf Technol 42(3–4): 233–241
2. Bala M, Chang TC (1991) Automatic cutter selection and optimal cutter path generation for prismatic parts. Int J Prod Res 29(11): 2163–2176
3. Choi YK, Banerjee A, Lee JW (2007) Tool path generation for free form surfaces using Bezier curves/surfaces. Comput Ind Eng 52(4): 486–501
4. Ibaraki S, Yamaji I, Matsubara A (2010) On the removal of critical cutting regions by trochoidal grooving. Precis Eng 34(3):467–473
5. Choi BK, Kim BH (1997) Die-cavity pocketing via cutting simulation. Comput Aided Des 29(12):837–846
6. Mansor MSA, Hinduja S, Owodunni OO (2006) Voronoi diagram-based tool path compensations for removing uncut material in 2½D pocket machining. Comput Aided Des 38(3):194–209
7. Choi BK, Park SC (1999) A pair-wise offset algorithm for 2D point-sequence. Comput Aided Des 31(12):735–745
8. Park SC, Choi BK (2001) Uncut free pocketing tool-paths generation using pair-wise offset algorithm. Comput Aided Des 33:739–746
9. Lin Z, Fu J, Shen H, Gan W (2013) Global uncut regions removal for efficient contour-parallel milling. Int J Adv Manuf Technol 68(5–8):1241–1252
10. Lin Z, Fu J, Shen H, Yao X (2015) Smooth contour-parallel tool path generation for high-speed machining through a dual offset procedure. Int J Adv Manuf Technol 81(5–8):1233–1245
11. Chang M, Kim CM, Park SC (2009) Tool-path generation for sidewall machining. Comput Ind Eng 56:1649–1656
12. Makhe A, Frank MC (2010) Polygon subdivision for pocket machining process planning. Comput Ind Eng 58(4):709–716
13. Dhanik S, Xirouchakis P (2010) Contour parallel milling tool path generation for arbitrary pocket shape using a fast marching method. Int J Adv Manuf Technol 50(9–12):1101–1111
14. Veeramani D, Gau YS (1997) Selection of an optimal set of cutting-tool sizes for 212D pocket machining. Comput Aided Des 29(12): 869–877
15. Veeramani D, Gau Y (2000) Cutter-path generation using multiple cutting-tool sizes for 2-1/2D pocket machining. IIE Trans 32:661–675
16. Balasubramaniam M, Joshi Y, Engels D, Sarma S, Shaikh K (2001) Tool selection in three-axis rough machining. Int J Prod Res 39(18): 4215–4238
17. Chen ZC, Fu Q (2011) An optimal approach to multiple tool selection and their numerical control path generation for aggressive rough machining of pockets with free-form boundaries. Comput Aided Des 43(6):651–663
18. Zhou M, Zheng G, Chen S (2015) An automated CNC programming approach to machining pocket with complex islands and boundaries by using multiple cutters in hybrid tool path patterns. Int J Adv Manuf Technol 1–14
19. Choy HS, Chan KW (2002) Enhanced strategy for milling corners. J Eng Manuf 216(8):1135–1154
20. Choy HS, Chan KW (2002) Machining tactics for interior corners of pockets. Int J Adv Manuf Technol 20(10):741–748

21. Choy HS, Chan KW (2003) A corner-looping based tool path for pocket milling. Comput Aided Des 35(2):155–166
22. Banerjee A, Feng HY, Bordatchev EV (2011) Process planning for corner machining based on a looping tool path strategy. Proc Inst Mech Eng B J Eng Manuf 225(9):1578–1590
23. Rahman AKMA, Feng HY (2013) Effective corner machining via a constant feed rate looping tool path. Int J Prod Res 51(6):1836–1851
24. Sui S, Li Y, Shao W, Feng P (2015) Tool path generation and optimization method for pocket flank milling of aircraft structural parts based on the constraints of cutting force and dynamic characteristics of machine tools. Int J Adv Manuf Technol 1–12
25. Elber G, Cohen E, Drake S (2004) MATHSM: medial axis transform toward high speed machining of pockets. Comput Aided Des 37(2):241–250
26. Elber G, Cohen E, Drake S (2006) Continuous toolpath generation toward 5-axis high speed machining. Comput Aided Des Appl 3(6): 803–810