



## Sample-and-Cover – a Brute-Force Approach to Continuous B-Axis Toolpath Generation

Tathagata Chakraborty<sup>1</sup> , Hariharan Krishnamurthy<sup>1</sup>  and Manoj Bhonge<sup>1</sup> 

<sup>1</sup>HCL Technologies Ltd, [tathagata.chakr@hcl.com](mailto:tathagata.chakr@hcl.com)

<sup>1</sup>HCL Technologies Ltd, [hariharan\\_k@hcl.com](mailto:hariharan_k@hcl.com)

<sup>1</sup>HCL Technologies Ltd, [manoj.bhonge@hcl.com](mailto:manoj.bhonge@hcl.com)

Corresponding author: Tathagata Chakraborty, [tathagata.chakr@hcl.com](mailto:tathagata.chakr@hcl.com)

**Abstract.** This paper presents a method to generate toolpaths for continuous B-Axis turning operations. Modern CNC turning centres allow continuous tool rotation about the B-Axis during operation; a suitably programmed tool orientation allows it to remove more material in a single setup. Given the tool and part profile shapes as chains of arcs and lines in 2D, the method facets the nose-radius offset of the part profile taken as the initial toolpath, and computes a gouge- and collision-free tool angle in each facet taking an initial angle. This initial angle is suggested by a strategy of tool movement, such as being normal to the toolpath. Contiguous facets with infeasible tool angles are identified. A process of “covering” them with newly constructed convex segments is applied to modify the toolpath. The steps of tool-angle computation and convex-covering are repeatedly applied till all infeasible segments are eliminated. Parameters including rotation limits, clearance angles and distances are incorporated when determining angles. The method is demonstrated with holder, insert and part shapes drawn from constructed and real-world examples. The method is simple to implement and draws on increased speeds available in current multi-core processors. Future work includes exploration of other tool movement strategies for optimized angles, checking properties, e.g., surface finish, and extension to roughing operations.

**Keywords:** Continuous B-Axis turning, Toolpath generation, Interference-check, Segment chains, Convex-covering, Tool movement strategy.

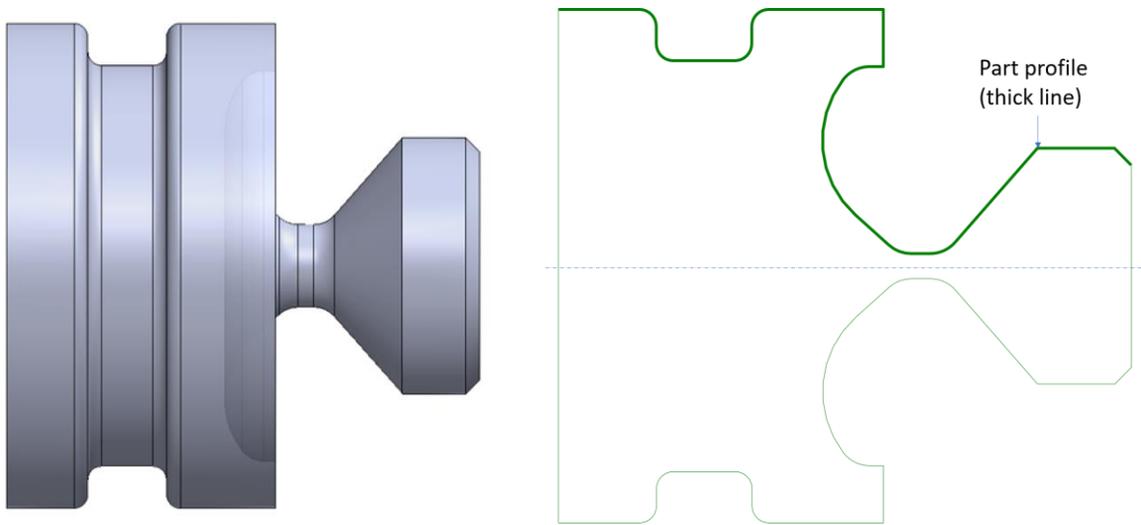
**DOI:** <https://doi.org/10.14733/cadaps.2021.714-737>

### 1 INTRODUCTION

Turning is a widely used and versatile method of manufacturing solids of revolution. In a conventional turning machine or lathe a rotating spindle holds the stock while a tool engages with the stock and removes material. The orientation of the mounted tool remains unchanged during the operation. Modern turning centres have the capability to change tool orientation during the operation. The tool rotation occurs in a plane defined by the spindle axis and the Y-axis of machine and is called

continuous or live B-Axis rotation. Continuous B-Axis rotation is useful to rotate the tool to a convenient angle to remove material from regions otherwise inaccessible by a fixed and static tool orientation. So the tool's path can be programmed to use the continuous rotation to remove a larger amount of material from the stock in a single setup. The generation of a continuous B-Axis toolpath for a part shape is the problem dealt with in the present paper.

The rest of the paper is organized as follows. The preliminary definitions, terms and problem statement are described in section 2. Prior art is reviewed in section 3. Then, the presented method is elaborated in section 4 where sampling-based safe-angle computation is detailed, followed by explanation of the process of convex covering for obtaining a continuous toolpath in section 5. The results with the method are exemplified with real-world parts and tools, after which a discussion of the method and its results is given, all in section 7. The paper is concluded in section 8, with a mention of topics of future work.



**Figure 1:** Example of a part profile – solid model (left), and section (right) showing the input profile.

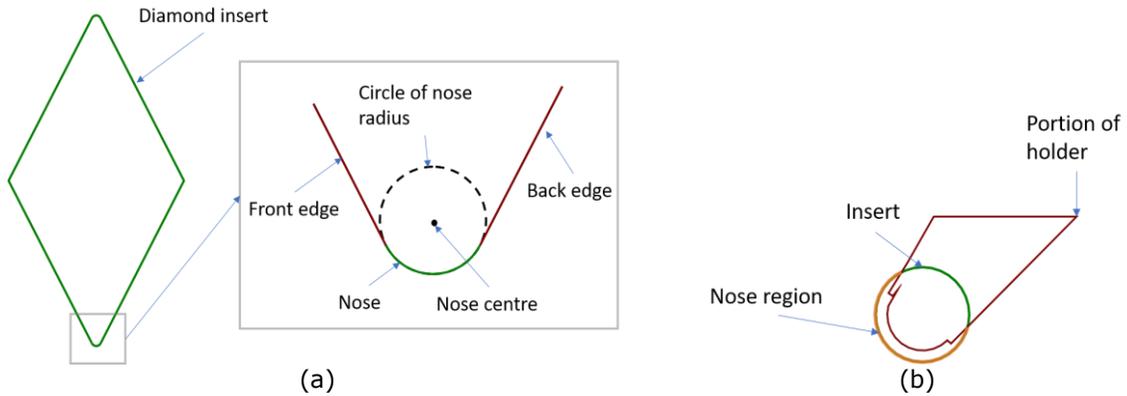
## 2 PRELIMINARIES AND PROBLEM STATEMENT

Turning toolpaths can be generated in a plane as turned parts are solids of revolution. The curve that is revolved to model the part is called the part profile. The part profile is required as an input shape for toolpath generation. The solid model and part profile of an example part are shown in Figure 1. Presently, all input shapes are assumed to be a chain of segments that are line-segments or circular arcs. Input shapes are also called *segment chains* hence.

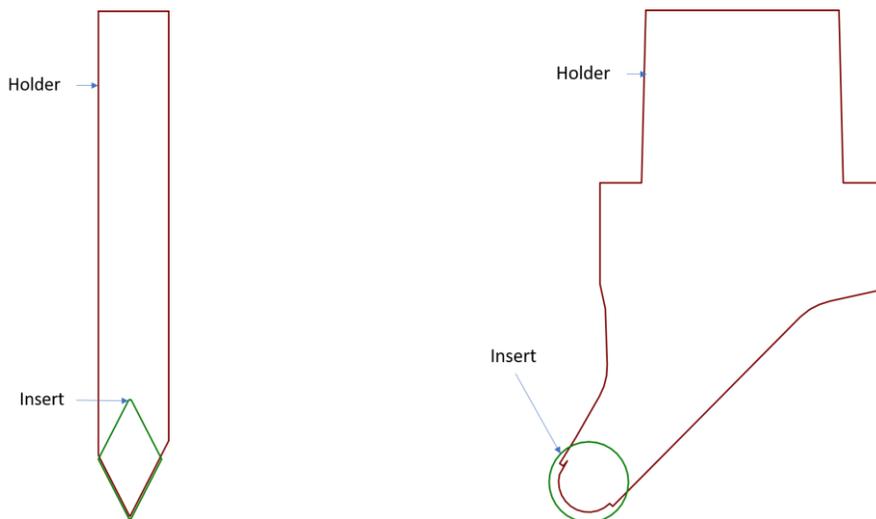
### 2.1 Tool Shape as Input

The shape of the turning tool is an additional input required for computing continuous B-Axis toolpaths. In its most basic form, a turning tool is an insert mounted on a holder; the holder and insert are usually made as a pair. An insert's shape can be of two types, namely of a polygonal or a circular cross-section. These are called polygonal and circular inserts in short. Examples of polygonal inserts include diamond, trigonal and hexagonal inserts. A circular fillet called the nose is present in place of one or more vertices of the polygon in polygonal inserts. The nose region for a diamond insert is shown in Figure 2(a), where the adjacent edges usually called the front and back edges are shown. It is the insert's nose that mainly engages with the stock and removes material. In circular inserts, the portion of the circumference of the circle which is exposed outside the holder is the nose,

as shown in Figure 2(b). The turning toolpath is the locus of the centre of the nose, or nose-centre, of the insert. The cross-section shapes of the holder and insert in the assembled state make up the tool shape required for computation. The tool shapes for tools with a diamond and circular insert are shown in Figure 3.



**Figure 2:** Details of nose-region in inserts – (a) for a diamond insert, (b) for a circular insert. A circular insert doesn't have front, back edges unlike (a). Nose region highlighted in (b).

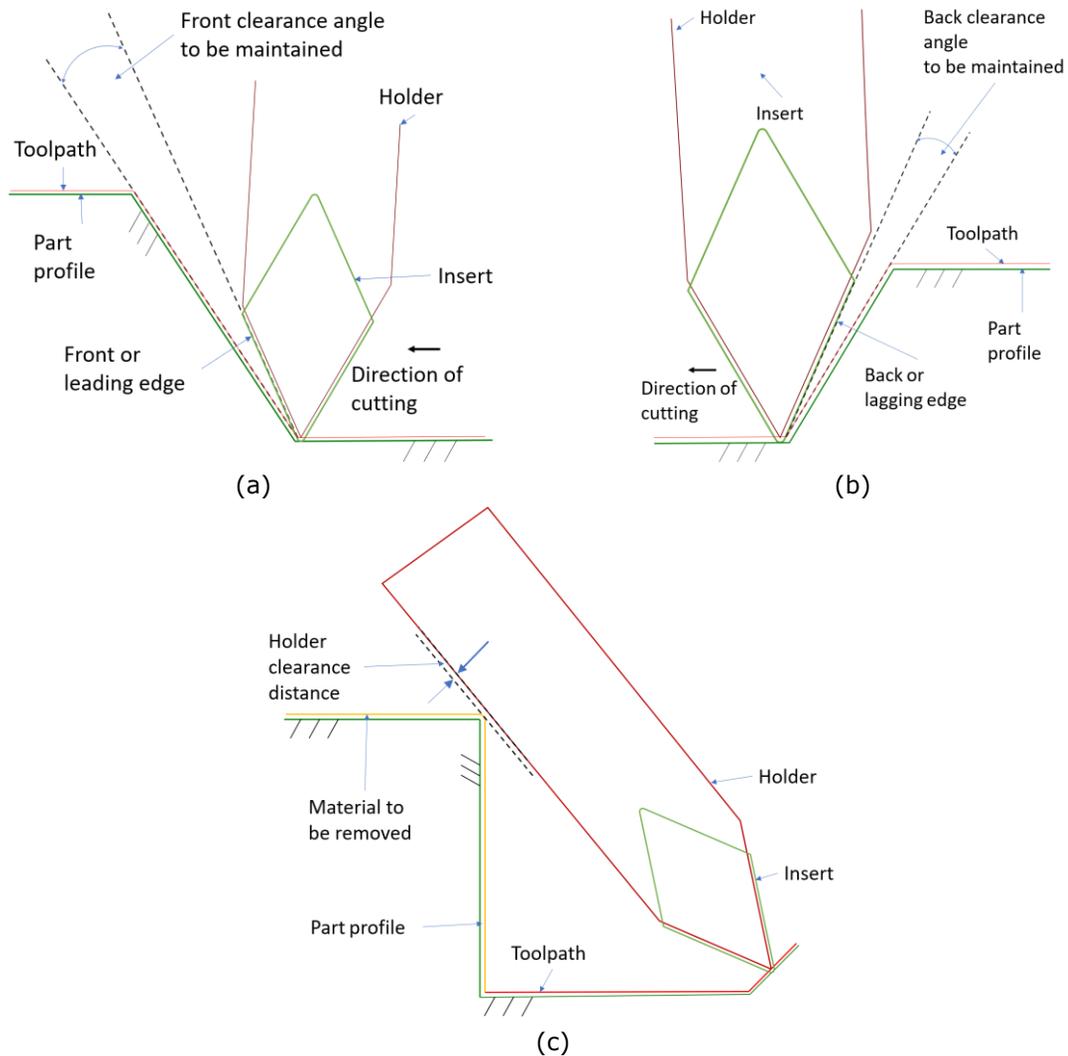


**Figure 3:** Examples of input tool shapes – (left) tool with diamond insert, and (right) tool with circular insert. Holder shapes shown in both examples.

## 2.2 Input Parameters

A tool can be rotated within a certain range of angles about the B-Axis. The limits of the range of rotation are a part of the machine's specifications. However, the limits can be restricted further for specific operations. An angular clearance is specified to be maintained between the front (leading) edge of the insert and the stock and is called the *front clearance angle*. Similarly, the back (lagging) edge of the insert is to be kept away from the stock by a *back clearance angle*. The front and back clearance angles are depicted in Figure 4(a) and (b) respectively. Front and back clearance angles

are applicable only for polygonal inserts. The holder can be required to maintain a distance from the stock during the operation called the *holder clearance*, as shown in Figure 4(c). The clearance angles and distance are recommended as a safeguard against collision of objects. The three parameters, namely the rotation limits, the front/back clearance angles, and the holder clearance distance are numerical parameters taken as input for toolpath computation.

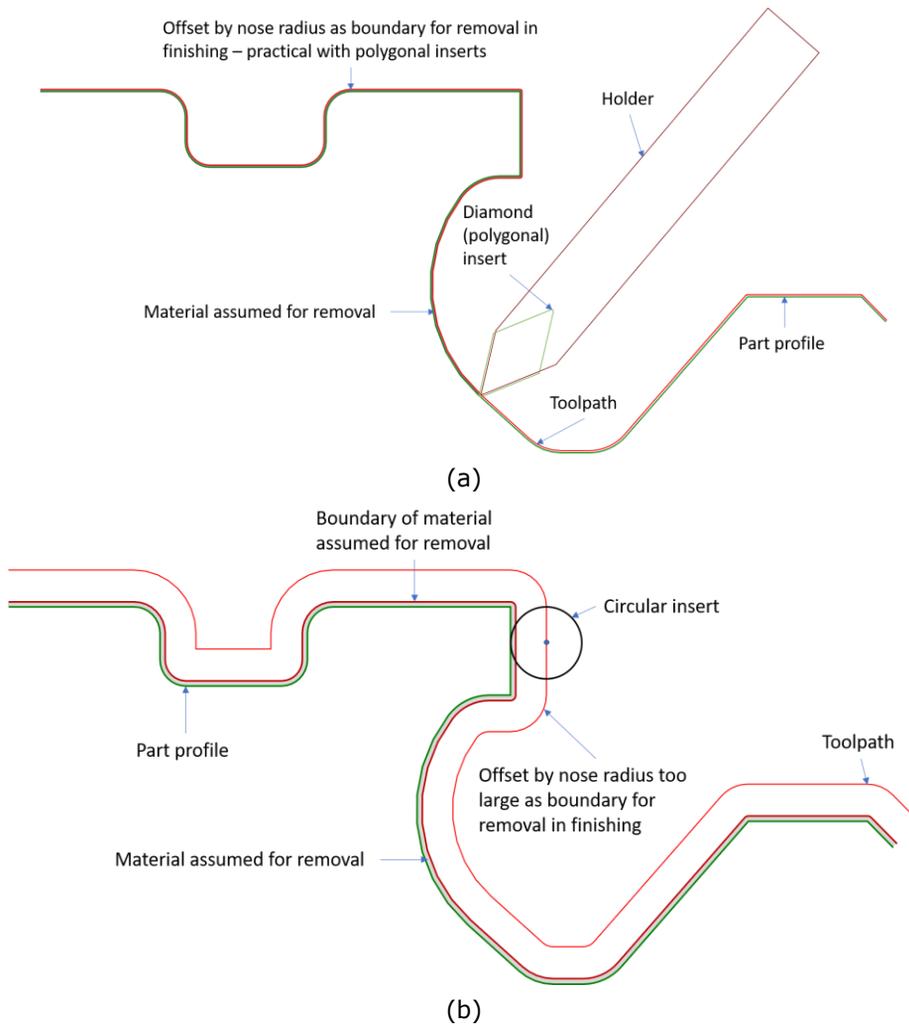


**Figure 4:** Depiction of input parameters – (a) front-clearance, (b) back-clearance angles, and (c) holder-clearance distance. (a) and (b) apply only to polygonal inserts. (c) applies to any tool holder.

### 2.3 Boundary of Material Removed

The present paper concerns with finishing operations in turning. A finishing operation removes a small amount of material left in a final stock that is close to the shape of the part profile. The bulk of material would have been removed from the original stock in earlier roughing operations leaving behind the final stock for finishing. When using polygonal inserts, the boundary of material removed

can be assumed to be the same as the toolpath itself as polygonal inserts have small nose radii typically; this is illustrated in Figure 5(a). However, it is impractical to assume that the toolpath and material boundary are same with circular inserts. Circular inserts have radii that are quite large compared to the material removed in a typical finishing operation. So, the part profile offset by a preset distance is considered as the material to be removed for a circular insert. This is shown in Figure 5(b).



**Figure 5:** Offset distance of part profile considered for material boundary – (a) nose-radius used for polygonal inserts, (b) small amount used for circular inserts, as nose-radius is impractical in finishing.

## 2.4 Problem Statement

The orientation of the tool about the B-Axis is referred to as the tool angle hence. Given the input shapes and parameters mentioned above, the aim in this paper is to develop a method to compute the toolpath and tool angle along the path, for finishing operations with live B-Axis turning, while

satisfying two conditions. One condition is that the tool angle should not cause gouge and collision, and the other is that it should be within the rotation limits. Gouge is the removal of excess material due to the tool penetrating the part profile. Collision refers to contact or penetration of the holder and the part profile. A tool angle free of gouge and collision is referred to as a *safe angle* for brevity. It is noted that honouring the parameters of clearance angles and holder clearance distance is included in the computation of safe angles.

### 3 PRIOR ART

The problem of generation of toolpaths with safe tool angles is dealt with extensively in literature on machining and 5-axis CNC toolpath generation. The main focus in such works falls into two categories broadly. One is to obtain a toolpath with desirable characteristics on configuration, such as optimized rotations, and motion, such as smooth tool velocity and acceleration. The other is the verification of a given toolpath or set of cutter contact locations for gouge. Gouge is classified into local, i.e., in the vicinity of the contact between the tool and stock, and global referring to collision between the tool, stock and other machine elements. It is stated that calculating a toolpath to avoid both local and global gouge is difficult in for a general tool and surface.

The authors in [11] present a method for computing a gouge-free toolpath from a given set of tool angles at discrete locations on the surface to be machined. The method uses a "visibility-map", a grid-based classification of the region around a location for tool accessibility, and explores paths by expanding the visible region with restrictions on tool angle change expressed as an "angular velocity" constraint. The authors in [5] develop solutions for tool orientation along a curve on a machined surface. They analyze the solution for two types of setups namely the part fixed and the tool changing configuration and vice-versa; the former requires numerically solving a differential equation while the latter admits closed-form solutions.

A method for detecting collisions on execution of an NC-program in 5 axis machining is presented in [10], where triangulated models of all machine elements and the stock are sectioned in appropriate parallel planes. These are updated with discretized steps of the motion as per the NC program, and checked for interference in terms of 2D intersections of the planar sections; the change in shape of the stock is accounted for when updating the sections. The work in [2] concerns a method to compute the 5-axis toolpath with triangulated parts by constructing "visibility cones" and pruning them for getting feasible tool access directions at a point on the part. These are verified by collision detection with the tool shape and then interpolated suitably for constructing a toolpath. The method in [1] analyzes tool-part contact to get optimal scallop width in 5-axis milling. Then points on a grid made from isoparametric toolpaths on a freeform surface are connected using so-called space filling curves obtained by a graph-based technique; the method corrects for sharp turns and angle changes in the toolpath. This and another method of toolpath generation based on clustering with vector-fields is described in [8].

In [3], a hardware-implementation of a method for calculation of toolpath for turning surfaces is presented, where the part surface around a given point is discretized and reoriented to the tool to find the closest point in the region, and this is used to find the optimal tool orientation. The patent [9] presents a method to generate toolpaths for roughing and finishing 5-axis operations by choosing accessible directions from a discretized region around a point or voxel, checking and correcting the accessible direction by interference checks and interpolating points and orientations between sampled points and voxels. The invention in [7] creates milling toolpaths by segregating interfering and contained segments of a planar part shape using the cutting direction and the tool's cutting edge, and using the interfering segments and tool shape to create a closed shape.

Sampling-based techniques have been used in literature on path planning in robotic contexts mainly. The authors of patent [4] present a method to determine a travelling (not cutting) path of a tool between terminal configurations by sampling and evaluating directions of tool positioning at two terminal configurations and using graph-based searching for planning the path; validity of the path is checked using simulation, collision-checking capabilities of the machine. Two classes of

sampling-based methods for path-planning are analyzed in [6], namely rapidly-exploring random trees and random graphs, both of which sample and expand a region around valid point in a planning space based on distance and other constraints.

It is seen that many methods described above use iterative methods for collision detection and path exploration. Some of the methods are limited to verification and interpolation of given tool configurations, while others consider local or global collisions when generating toolpaths. In the spirit of usage of sampling, a method for generation of toolpaths for determining a path with feasible tool position and orientations in a 2D-setting, such as in the continuous B-Axis setting is presented in this paper. The method generates the toolpath while preventing both local and global gouging of the tool and the part or stock.

#### 4 SAMPLING FOR SAFE ANGLES

In finishing operations the amount of material to be removed is small, and the stock is close to the shape of the part profile, as mentioned in section 2.3. So, the best toolpath in a finishing operation would be a path close to the part profile itself. The best toolpath for a continuous B-axis operation would be the offset of the part profile by the insert's nose radius. This is because the toolpath is the path of the nose-centre. The offset is a valid toolpath if all points of the path admit a safe angle and are accessible by the tool in sequence. But this is not true in general. So, the offset is taken as an initial toolpath for the present method.

There could be a range of possible angles which satisfies the conditions of safety and falling in rotation limits at each point of the initial toolpath. The choice of the tool angle is decided by an overall strategy of specifying a desirable tool angle. The strategy refers to the manner of change in tool angle along the toolpath; it is expressed in terms of a geometric property. Examples of strategies include the tool angle being a constant user-specified value as long as possible, or bearing an angle to the normal to each point on the toolpath among others. The former is referred to as *being at a constant angle* and the latter *being at an angle to the normal*. These two strategies are considered in the present paper. The specification of a suitable strategy depends on the objectives sought in the operation, such as minimal changes in tool angle, or an angle modelled for good surface finish, reduced tool wear and reducing unwanted forces. The tool angle at a point specified by the strategy is called the desired tool angle at that point.

The validity of the desired tool angle at each point of the toolpath needs to be evaluated by checking if the conditions are satisfied. A valid angle should also honour the parameters of clearance angles and distances. This is incorporated in the method of verifying the tool angles or calculating a valid angle. For incorporating the clearance parameters, the input shapes are pre-processed as described in the following.

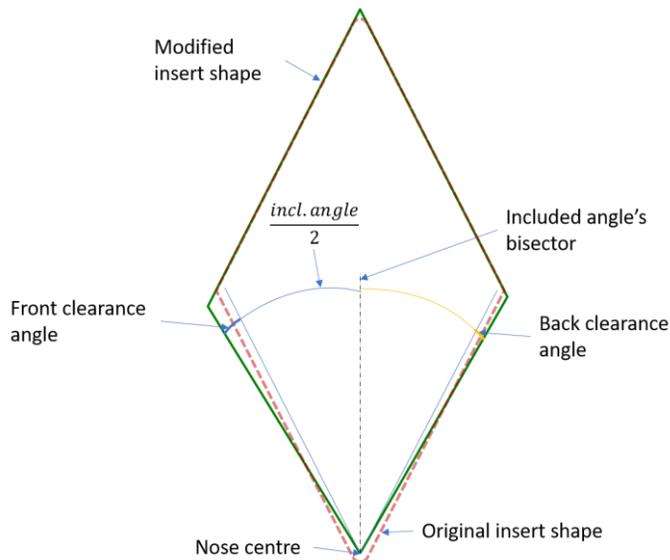
##### 4.1 Pre-processing for Obeying Parameters

The safe angle at a point should make the tool shape obey the holder clearance distance and front/back clearance angles where applicable. A possible approach for this is to calculate the tool angle that makes the tool just contact the stock, and then adjust the calculated angle for the clearance parameters. However, instead of modifying the angle post calculation, the insert and holder shapes are pre-processed to accommodate clearance parameters in the calculation of safe angle as follows.

###### 4.1.1 Modification of insert shape

Let a direction along the bisector of the included angle of the insert be considered. By construction, the front and back edges of the insert would be inclined to the bisector at half the included angle. A new edge is constructed by adding the front clearance angle to half the included angle as its inclination. The new edge is made to start or terminate at the nose-centre. Then, the original front edge is replaced with the newly constructed edge. A similar construction is performed with the back edge and the back clearance angle, and the original back edge is replaced by the constructed edge.

The nose arc of the original insert is removed. This completes the construction of a modified insert shape.



**Figure 6:** Insert shape modification to honour clearance angles – clearance angles added to angle of front, back edges of original insert (dashed line), nose removed, to get modified insert (solid line).

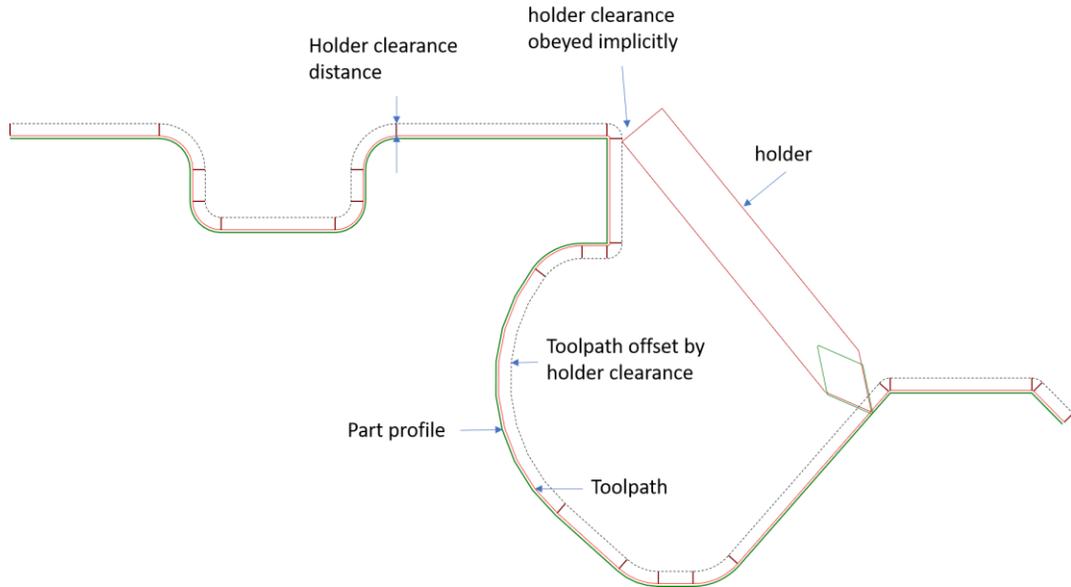
The modified shape is used when checking for gouge and collision during computation of safe angles. The modified insert obeys the clearance angles implicitly when used in toolpath computation, as the inclination of edges includes the clearance angles. The original and modified insert shapes and the modification process are shown in Figure 6.

The nose-arc is removed as the nose is unnecessary for toolpath computation. This is because the nose centre would always be on the toolpath, as the toolpath is the path of the nose-centre. The insert shape is modified only for polygonal inserts, as clearance angles are specified only for such inserts.

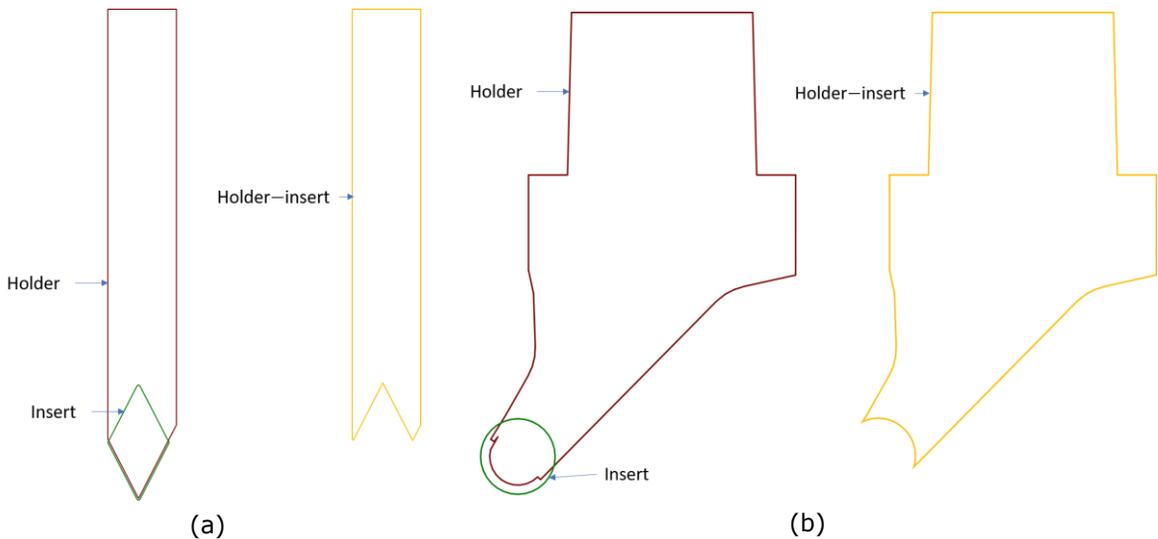
#### 4.1.2 Pre-processing for holder clearance

The holder clearance is the minimum distance to be maintained between the holder shape and the part when the tool moves along the toolpath. The minimum distance of separation would be ensured by offsetting the toolpath itself by the holder clearance. The toolpath offset by the holder clearance will be a segment chain additionally used in the calculation of safe angle. The part profile and holder-clearance offset segment chains are shown in Figure 7.

The holder clearance applies only to the holder shape. The insert shape is not bound to obey this distance. For making this distinction in computation of safe angles, the insert shape is subtracted from the holder shape as a Boolean operation. The modified holder shape will be used for collision checking with the offset by holder-clearance during the computation of the safe angle. The modified holder shape is shown in Figure 8(a) for a diamond insert and Figure 8(b) for a circular insert.



**Figure 7:** Shape used for honouring holder clearance – initial toolpath (orange) offset by holder clearance to give shape (dashed line) for intersection checking with holder shape.



**Figure 8:** Holder shape used for intersection checking – (a) for diamond, (b) circular inserts. Insert subtracted from holder for calculations as insert subsumes holder shape in common regions.

#### 4.2 Contact-based Safe Angles

One possible approach to compute the safe angle at a point of the toolpath is to calculate the range of all possible safe angles at that point. Let the tool be placed at the intended angle as per the strategy of tool rotation at a point of the toolpath. The angle that would make the tool shape contact the part, called the contact angle in short, is to be calculated. The contact angles in the clockwise and anti-clockwise senses of rotation give the range of safe angles at that point.

The contact angle is found as follows. The point under consideration separates the toolpath into forward and backward portions. Let the forward portion be considered. Let  $P_{tool}$  denote the set of end-points of segments of the tool shape and  $P_{part}$  denote the end-points of segments of the part profile. The contact angle between each  $P_i \in P_{tool}$  and the part segment chain is found. Similarly, the contact angle between each  $P_i \in P_{part}$  and the tool shape is calculated. The contact angle is found in the counter-clockwise sense by default. The end-point  $P_i \in P_{tool} \cup P_{part}$  which yields the least angle of rotation from the intended to the contact angle is the contact angle of the tool shape. Let the found contact angle be denoted  $\theta_f$ .

Arcs in both the segment chains are faceted into line-segments for the calculation of contact angle. It is mentioned that during the calculation of contact angles, the holder shape is checked for contact with the initial toolpath offset by the holder clearance distance mentioned in 4.1.2. The insert shape is checked for contact with the initial toolpath itself, as the insert is already processed for clearance angles.

The tool shape could already be penetrating the part profile when placed at the intended tool angle at the given point. Then, the end-point  $P_i \in P_{tool} \cup P_{part}$  bearing the smallest rotation in the clockwise sense would be  $\theta_f$ . The same calculations are repeated with the backward portion of the toolpath, with the default sense being clockwise for the scenario where the tool isn't already penetrating the part profile and counter-clockwise if it does so. The contact angle for the backward portion is denoted  $\theta_b$ .

When the intended angle does not mean the penetration of the tool and part profile,  $[\theta_f, \theta_b]$  is the range of safe angles at the point of concern on the toolpath. In the case of penetration in either the forward or backward portions, the magnitude of contact angles is compared. If  $\theta_f$  is clockwise and  $\theta_f \leq \theta_b$ , then the range of safe angles is definable as above. However, if  $\theta_f > \theta_b$  no safe angle is possible at that point, as the angle to escape penetration in the forward portion exceeds the angle to contact the backward portion. An analogous logic is applied for penetration in the backward portion. No safe angle is possible if both the forward and backward portions entail penetration.

This method of calculation of the range of safe angles may not work in the following scenarios. If the segment chains penetrate, then the sense of rotation to escape penetration may not always be the opposite to the default sense for a portion. The tool shape itself may penetrate the portion of concern at the contact angle, so an interference check would always be required to verify the computed angle. Further, this method would not give all contact angles in one portion if the angle is not unique. The implementation would also be difficult to maintain in the long-run due to the states of contact to be handled, and comparisons of  $\theta_f$  and  $\theta_b$ . So a simpler method of iteratively searching for the safe angle at a point of the toolpath is presented in the following.

### 4.3 Iterative Intersection-checking

A safe angle needs to be calculated at a point of the toolpath only if the tool at the desired angle penetrates the part profile. Thus, the tool at the desired angle is checked for intersection with the part profile at a point of the toolpath. Only if intersection is found, a safe angle is calculated. Starting with the desired angle, the tool angle is changed in steps in one sense, say the counter-clockwise direction. The segment chains are checked for intersection, for each tool angle. This is continued until a safe angle is found or the rotation limit is reached. If no safe angle is found in the search direction, then the search is performed in the opposite sense. Failure in finding the safe angle in both senses of rotation is marked.

The iterative search for a safe angle is simpler than finding contact-angles as the heuristics for states of intersection of chains and branches for comparison of angles are not required in the former. The computational expense of intersection checking is handled by increased speed of modern processors. Also, the nature of intersection checking offers great scope for parallel processing. The method is robust due to absence of 'cases', and is simple in implementation.

#### 4.4 Sampling Toolpath for Tool Angles

Discretization is used in many methods of toolpath analysis and path planning in literature as seen from the review in section 3. In this paper, a sampling-based method is used to find the safe tool angle along the toolpath. The toolpath is discretized into facets of a preset length; arcs are faceted into smaller arcs and line segments into smaller line segments. The preset length or the resolution of discretization can be chosen suitably for handling generally expected features and their size in part profiles. A smaller resolution handles most profiles, at the cost of increasing computation time.

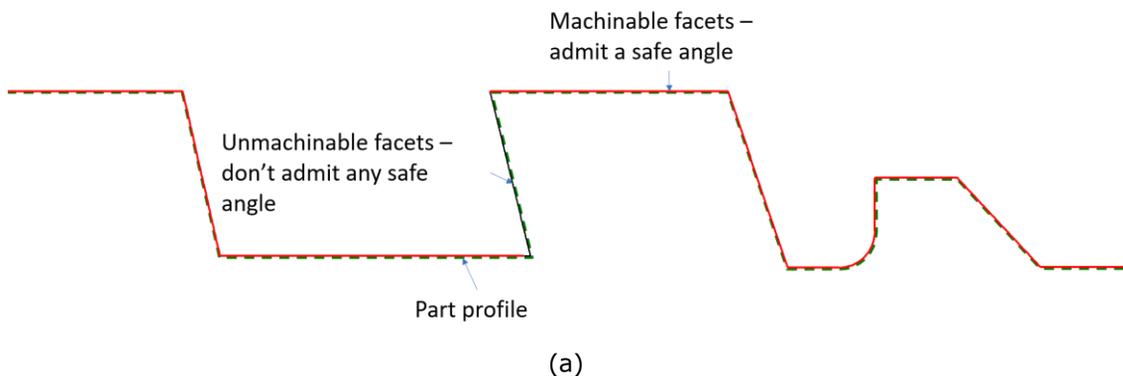
##### 4.4.1 Advantages of a sampling-based method

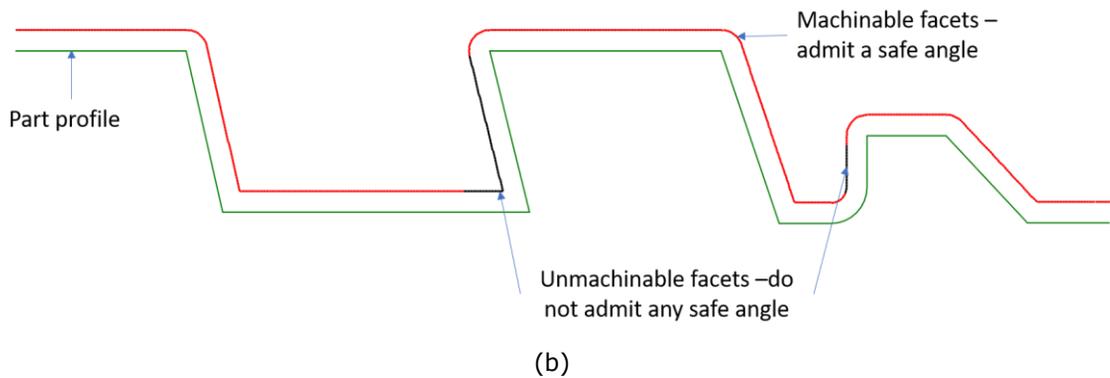
The use of discretization simplifies the process of the calculating the tool angle along the toolpath. A more efficient alternative could be to translate the tool shape along the toolpath, rotating the tool upon detection of interference. But in this process, a straight line may need to be broken up a number of times depending on the shape of the neighbouring regions of the profile to calculate safe angles. Also, translation along an arc would entail tool rotation, so arcs need to be faceted to calculate the safe angle. In effect, any advantages of efficiency are superseded by the need for discretization of segments, possibly the entire toolpath in the worst case. So a sampling-based method is adopted. As before, the method means a simpler implementation and maintenance, while utilizing increased speed in recent processors.

##### 4.4.2 Unmachinable facets

In the faceted toolpath, each pair of consecutive points defines a facet. A safe tool angle need only be found at the end-point of each facet. This would automatically specify the tool angle at the start point of the next facet due to the toolpath being continuous in position. A facet for which no safe angle can be found at either of the terminal points is deemed as unmachinable or infeasible. The initial toolpath with machinable and unmachinable facets marked in different colours is shown in Figure 9(a) for a diamond insert and Figure 9(b) for a circular insert.

Upon completion of safe-angle computation, all facets of the faceted toolpath fall into two classes namely machinable and unmachinable facets. Unmachinable facets cannot be present in the final toolpath as they cannot be physically realized. So, the toolpath must be modified to make all its facets or segments machinable. A systematic method of joining machinable points and eliminating unmachinable segments is required for this. This is presented in the next section.





**Figure 9:** Unmachinable facets marked from safe-angle finding with initial toolpath for a part profile – (a) for a diamond insert, and (b) for a circular insert, both with limits  $[-45^\circ, 45^\circ]$  w.r.t. vertical.

## 5 CONVEX-COVERING PROCESS

The sampling-based process of safe-angle computation along the toolpath identified machinable and unmachinable facets in the initial toolpath. Sets of unmachinable facets usually occur contiguously in portions of segments unreachable by the tool for example; rarely they occur as standalone facets. It is possible that one or both ends of a facet are unmachinable, but in either case the facet is deemed unmachinable. A straightforward method to remove unmachinable facets would be to replace a set of contiguous unmachinable facets by a segment joining the terminal points of the set. However, the joining segment would often intersect the part profile itself, such as if the set contains convex arcs for example. In general, a set of facets could be unmachinable due to the shape of neighbouring regions on the part profile, rotation limits, apart from the shape of the set itself. So, a systematic method is required to modify sets of unmachinable facets.

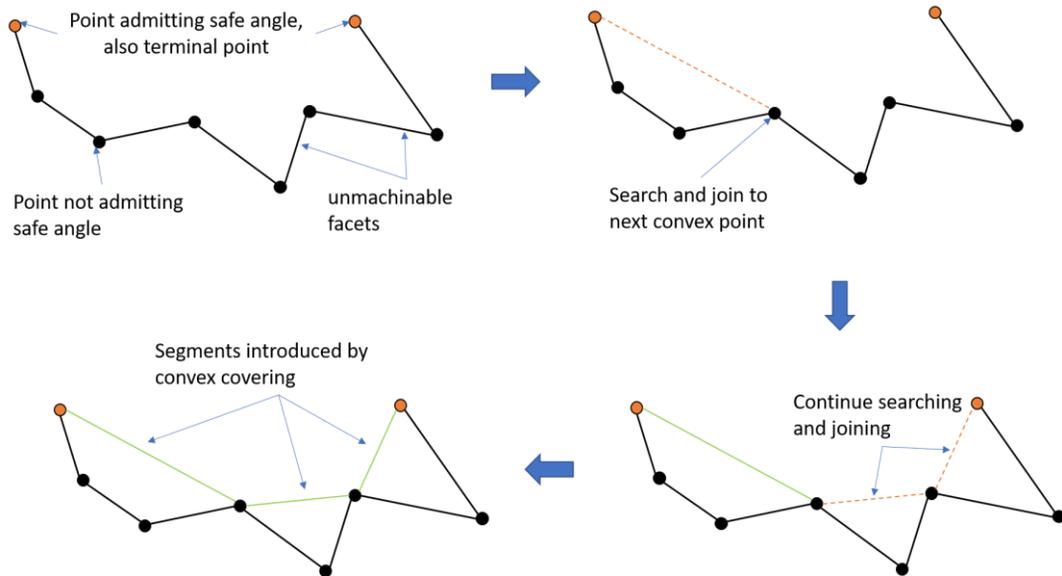
### 5.1 Convex Covering - Definition

Unmachinable facets often occur in concave regions of the part profile that are inaccessible by the tool within the given limits of rotation. Such sets can be enclosed by a segment or multiple segments that would be convex with respect to the set. The set of facets is covered by segments that are convex for the shape of that set. So this process is termed "*convex-covering*". A convex cover is a set of one or more straight-line segments connecting points of facets in a set of unmachinable facets, and is always convex with respect to the set of facets enclosed by the segment. In this sense, a convex cover can be thought of as a collection of local convex hulls of subsets of the set of unmachinable facets. It is noted that the convex-cover itself need not be convex as the relative angle of convex cover segments depends on the subsets which each segment covers.

### 5.2 Covering of Concave Sets

A set of unmachinable facets is considered concave if it contains at least one concave point. The usual definition of a concave and convex point is adopted here, i.e., in the former the angle between facets containing the point is  $\leq 180^\circ$ , and is  $> 180^\circ$  in the latter. Given a concave set of unmachinable facets, the method of convex covering begins with one of the terminal points of the set, and iterates through the facets of the set to search for a concave point. Once a concave point is found, the immediate next convex point is searched for and joined to the terminal point, forming one segment of the convex cover. The search for a concave point and construction of new segments is continued. Every new segment created is appended to the convex cover segments already present. This process ends when the other terminal point of the set is reached. The list of newly constructed segments

forms the convex cover of the set of unmachinable facets. The process is depicted schematically in Figure 10.



**Figure 10:** Covering a concave set of unmachinable facets – (top left) from one end point, interior convex points (top & bot. right) are searched and joined to ends to obtain final cover (bot. left).

### 5.3 Covering Already Convex Sets

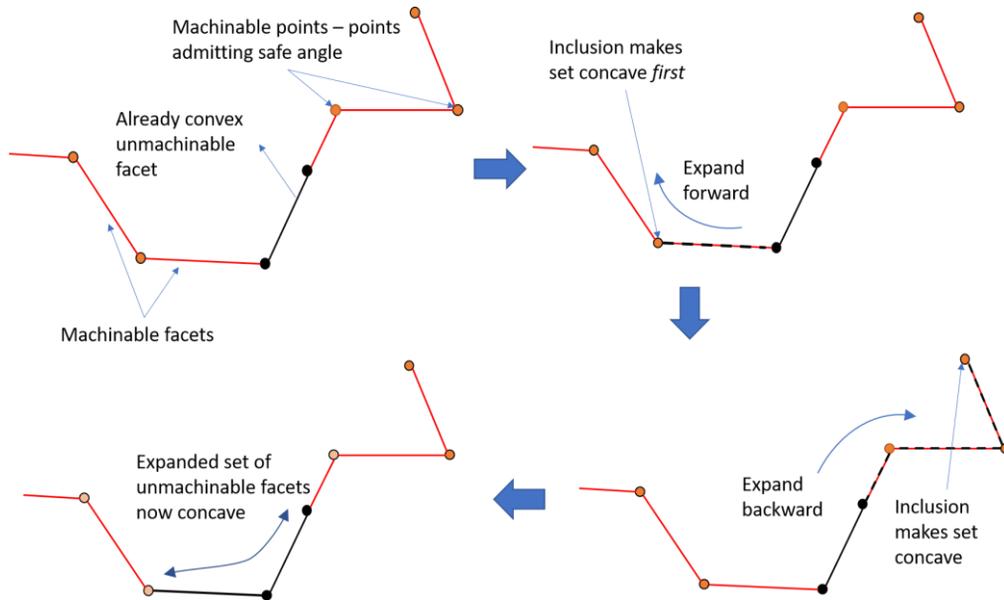
A set of unmachinable facets could already be convex, meaning it contains no concave points. Examples of convex sets of unmachinable facets include facets occurring in the interior of a straight segment or a convex circular arc. If the convex-covering process described above is applied directly then segments identical to the original set would be obtained repeatedly. So in such situations the set of unmachinable facets is first enlarged by including facets in the forward direction of the toolpath starting from one terminal point of the unmachinable set. A similar expansion is done in the backward direction from the other terminal point. This expansion is aimed at making the expanded set concave. The expansion of the set proceeds as follows. Every time a facet is added to expand the set, the convexity of the set is checked. Convexity is checked by computing the convex cover of the current expanded set and checking if it has the same total length as the expanded set itself. If the lengths do not match, a concave point from the added facet would have caused the mismatch in length. This indicates that the set has become concave now, signaling termination of expansion. Whichever of the two expansions is smaller and succeeds in rendering the expanded set concave is taken as the enlarged set of facets. The process of expansion is depicted in Figure 11. Now, the usual convex-covering process described in section 5.2 is applied again, to modify the toolpath in this region.

## 6 ALGORITHM OF TOOLPATH GENERATION

The process of safe-angle computation serves to validate a given toolpath, while that of convex-covering serves to modify the toolpath to go over unmachinable facets. Individually, the two processes cannot produce a valid toolpath. So, an algorithm of toolpath generation is developed by putting the two steps together as follows.

## 6.1 Repeated Covering of Modified Toolpath

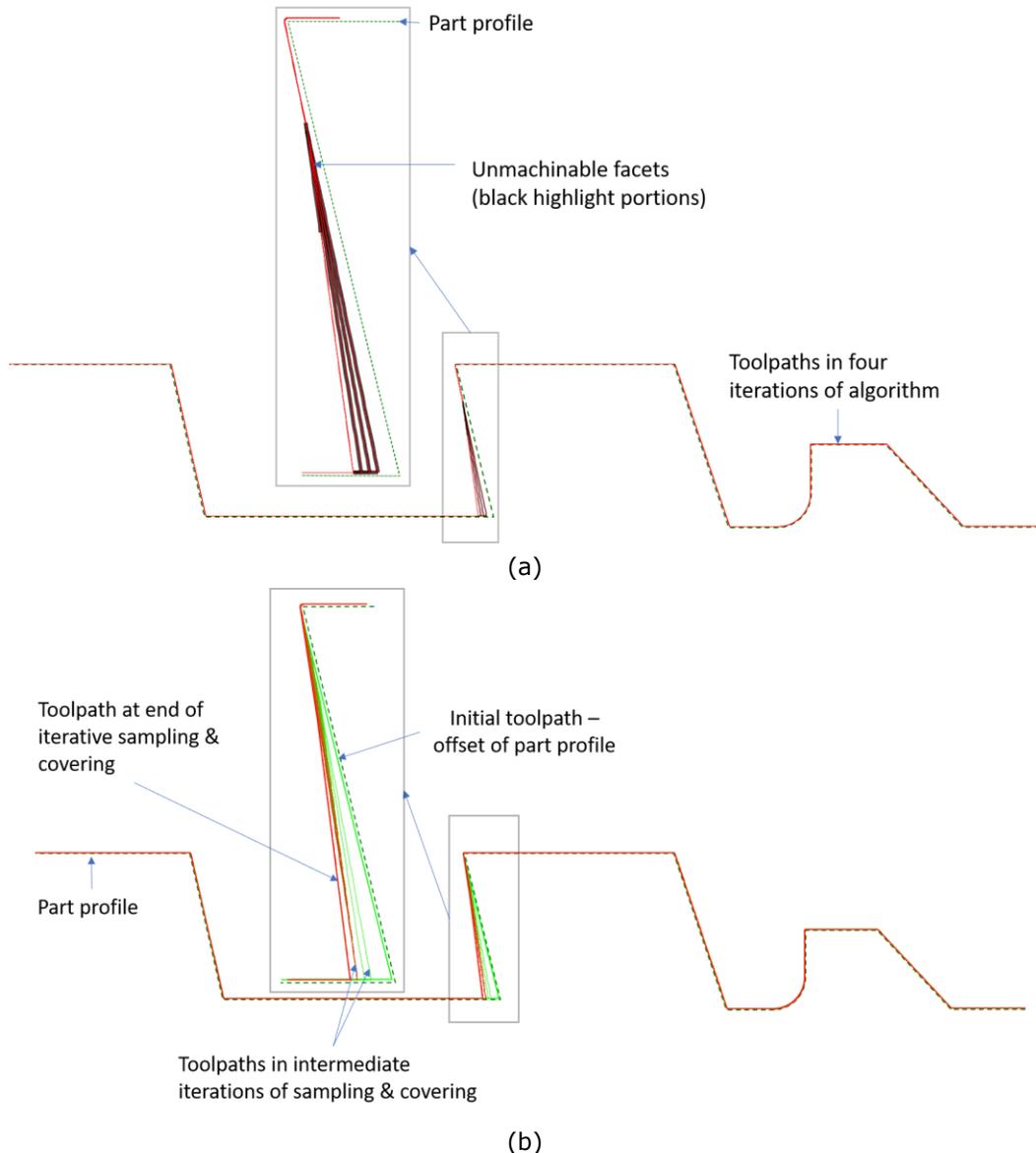
The offset of the part profile by the insert's nose radius is the initial toolpath. Starting with the initial toolpath, the following steps are repeatedly performed:



**Figure 11:** Expansion of already-convex region – (top left) convex facet(s) is expanded front (top right) and back (bot. right). Former yields a concave region with lesser facets (bot. left) for covering.

1. The toolpath is discretized into facets by a preset resolution; only non-faceted segments are faceted. The desired tool angle is calculated at the end point of each facet as per the strategy of tool angle change.
2. The desired tool angle is checked for satisfaction of conditions of safety and restriction to rotation limits. In case of violation, a safe angle is computed at that point. Facets not admitting a safe angle are marked as unmachinable.
3. The tool angle at a point is made to be the arithmetic mean of the angles computed in facets coincident at that point, for every point on the toolpath (section 6.1.1).
4. Groups of unmachinable facets are replaced by segments of the convex-cover process.
5. Steps 1, 2, 3 and 4 are repeated until the termination condition is reached. The termination condition is that no unmachinable facets are found in a toolpath, or a set of unmachinable facets is not removed by convex covering, whichever occurs first.

In practice, it is the first clause of the ending condition that's seen to occur first in almost all cases. This means that a valid toolpath will eventually be obtained by several iterations of convex covering. In other words, it is hypothesized that the iterative process of safe-angle computation and convex-covering will terminate in a toolpath that is continuous and machinable throughout. The toolpath from four iterations of the above algorithm is shown for a part profile in Figure 12(a) with a diamond insert, where unmachinable facets are highlighted in black. The toolpaths from all iterations from initial till termination are shown for the same part in Figure 12(b). The unmachinable regions in four iterations and intermediate toolpaths from the algorithm are shown for another part profile in Figure 13(a) and (b) respectively for a circular insert. It is seen that the extent of unmachinable regions is smaller in Figure 13 than in Figure 12, resulting in fewer number of iterations.

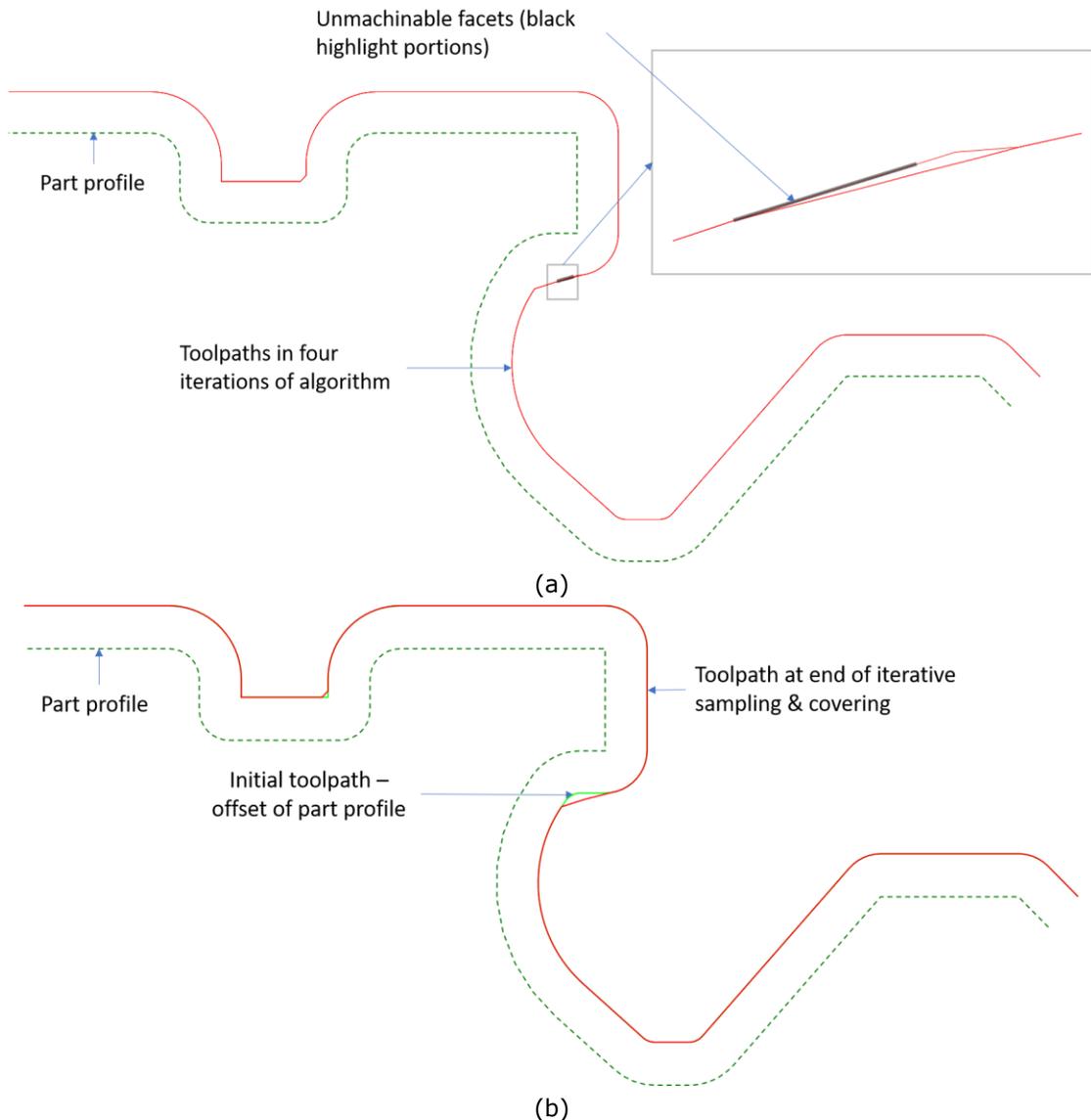


**Figure 12:** Iterations of sampling & covering with a diamond insert – (a) four iterations with unmachinable facets highlighted, (b) all iterations. Unmachinable facets seen reducing with covering.

### 6.1.1 Averaging of angle in sharp corners

For a continuous toolpath, it would be expected that the tool angle at the common point shared by two consecutive facets is unique. However, the presence of sharp corners in the part profile would mean that the tool angle suggested at the corner by the incident facets is different. Here, a sharp corner refers to points with discontinuity in the tangent, such as common point of two perpendicular segments. The tool angle may abruptly change across sharp corners, apart from being non-unique at the corner. So, the tool angle at a sharp corner is replaced by the arithmetic mean of the tool angle contributed by the incident facets. This is done to consider the tool angle suggested by both incident facets at the corner. The averaging is done for the toolpath obtained in each iteration of the

algorithm, and the final toolpath. It is noted that usually such corners would be offset into circular arcs but can be chosen to be offset as sharp corners.

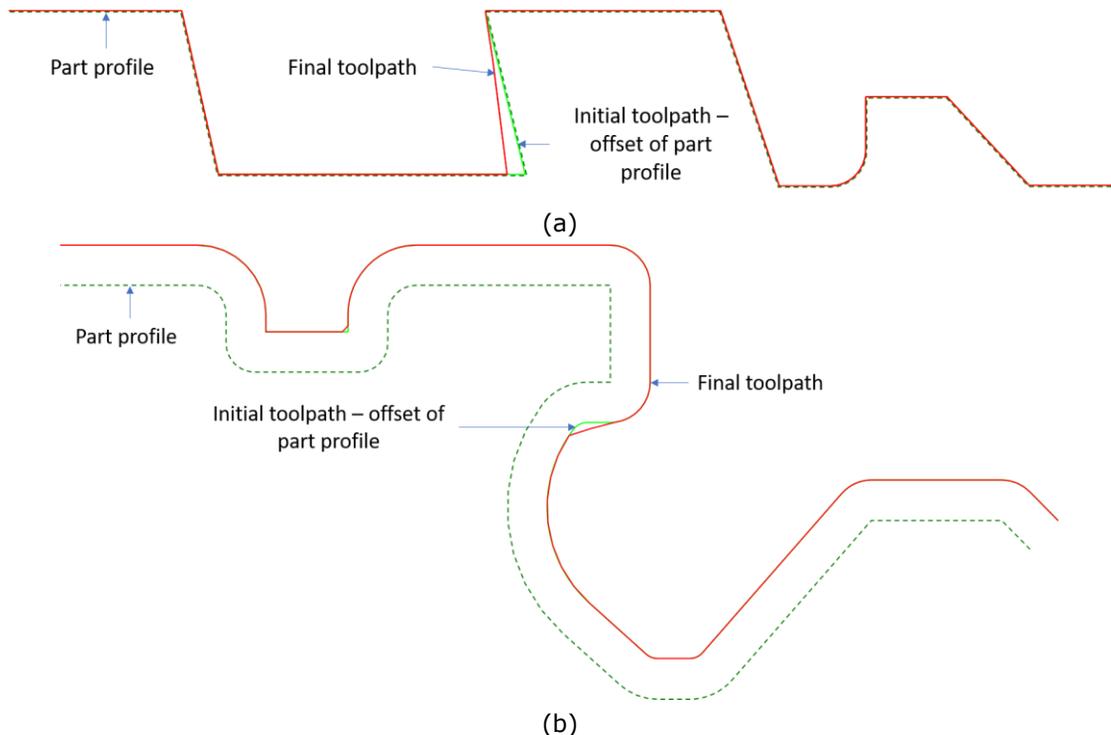


**Figure 13:** Iterations of algorithm with circular insert – (a) four iterations highlighting unmachinable facets, (b) all iterations. Toolpaths largely same, except small part made feasible quickly by covering.

## 6.2 Final Processing Steps

The iterative sampling and covering process gives a faceted segment chain as the output. This toolpath cannot be used as such to create an NC program, due to the potentially large number of moves from the discretization. Further processing may be required such as smoothing of the angle change. The desired output would be a segment chain to be post processed into a final program according to the geometric entities handled by a specific machine. Thus, the following steps are performed with the final faceted toolpath.

### 6.2.1 Facet collation

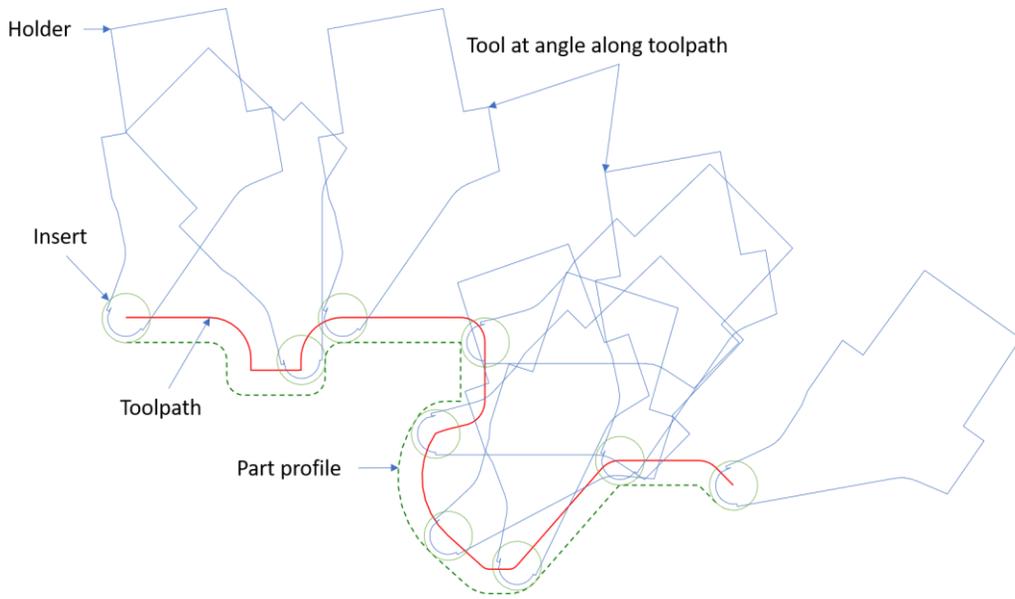


**Figure 14:** Final toolpaths got after post-processing steps – (a) with diamond insert, and (b) with circular insert. Initial and final toolpaths differ in unmachinable regions, more significantly in (a).

Many facets of the final toolpath may be consecutive and redundant, having same tool angle. So facets are merged in the output toolpath to form segments. All contiguous facets having the same “rate of angle change” are merged to form a segment. This is done for both lines and arcs, as they were distinctly faceted (section 4.4). Here, the “rate of angle change” is defined as the ratio of difference in tool angle to the facet length. For example, contiguous facets having the same tool angle or the same increase in tool angle have the same rate of change. The merging condition ensures that a merged segment has a linear change in tool angle; facets will be distinguished if the rate of change is different. If consecutive facets have a different rate of change, they would be retained in the final toolpath. The initial and final toolpath after facet collation are shown for a part profile and a diamond insert in Figure 14(a), and a different part profile and circular insert in Figure 14(b).

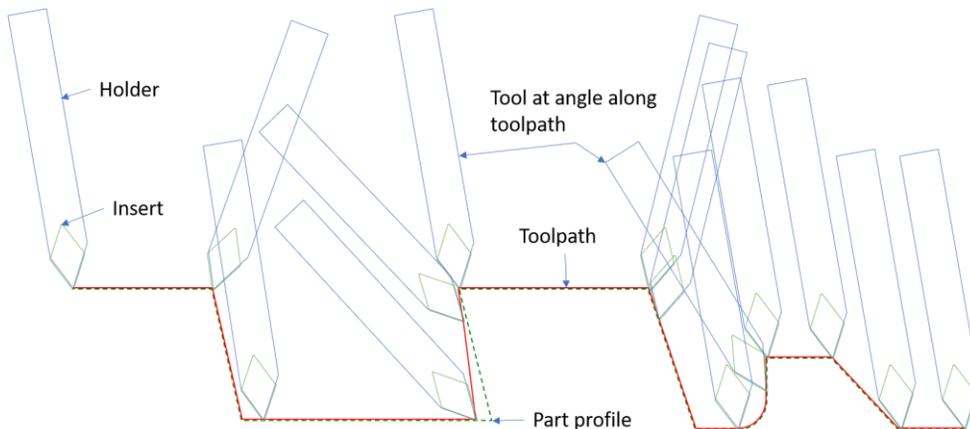
## 7 RESULTS AND DISCUSSION

The results of applying the method of toolpath generation are presented in this section. The method was tested with several different part profiles were modelled as test parts, with profiles from synthetic and real-world examples. The profiles ranged from simple ones with relatively smaller undercut regions to complex ones with the undercut regions spanning three-quarters of a circle, such as the profile in Figure 1. The part profiles in Figure 1 and Figure 9 are referred to as profile 1 and 2 respectively hence. The diamond insert used has a nose-radius of 0.4 mm, while that of the circular insert is 7 mm.



Toolpath with strategy – at angle ( $10^\circ$ ) to normal, Range  $[-45^\circ, +45^\circ]$  w.r.t. vertical

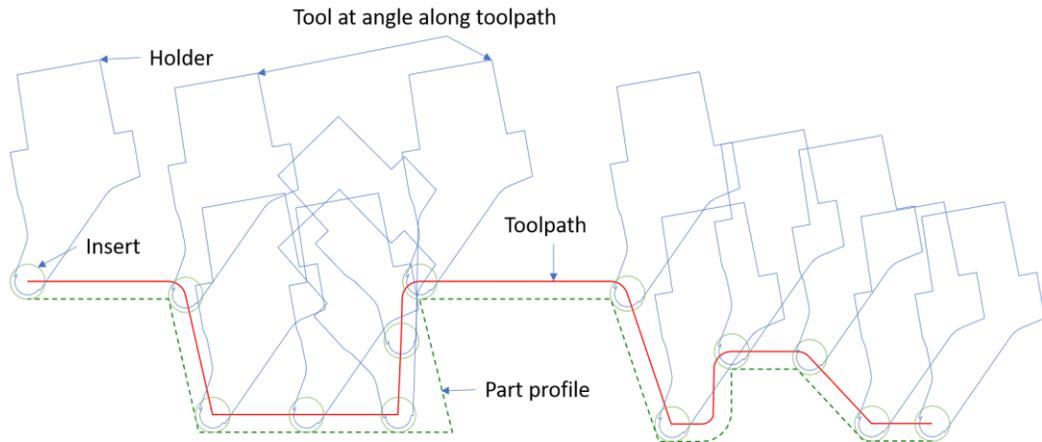
**Figure 15:** Toolpath at angle of  $10^\circ$  to normal in  $[-45^\circ, +45^\circ]$  for profile 1 with circular insert. Tool shown at fewer positions than on the toolpath for clarity.



Toolpath with strategy – constant angle ( $10^\circ$  to vertical), Range  $[-45^\circ, +45^\circ]$  w.r.t. vertical

**Figure 16:** Toolpath at constant angle ( $10^\circ$ ) strategy with diamond insert in  $[-45^\circ, +45^\circ]$  for profile 2.

The results of the toolpath generation algorithm are visualized with the tool placed at the tool angle along the toolpath, for part profile 1 in Figure 15 with a circular insert and being at  $10^\circ$  to normal as the tool strategy. The toolpaths for part profile 2 with being at a constant angle of  $10^\circ$  to vertical for diamond and circular inserts is shown in Figure 16 and Figure 17 respectively.

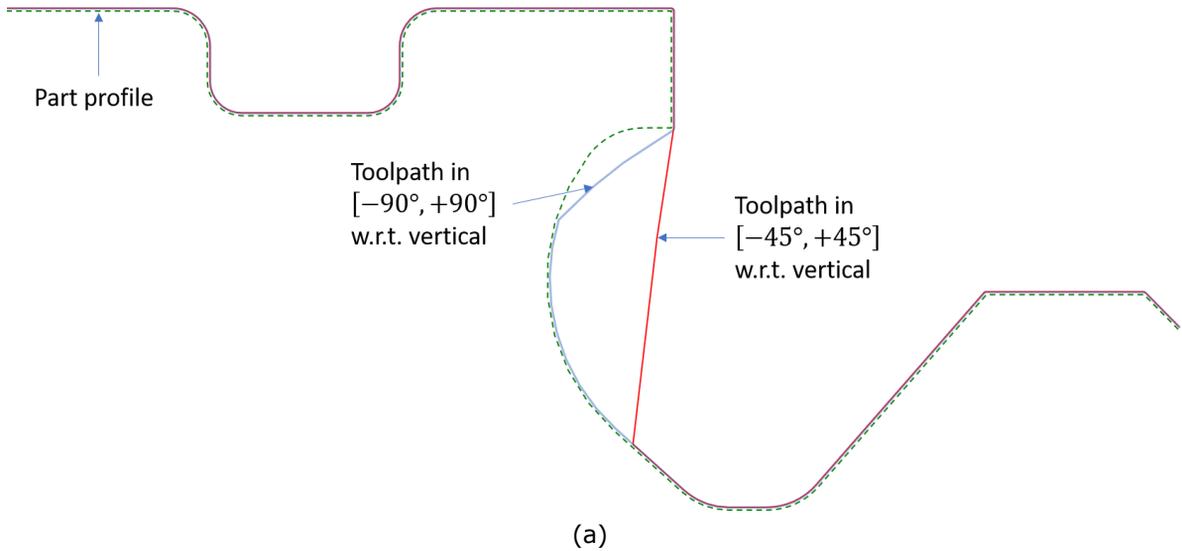


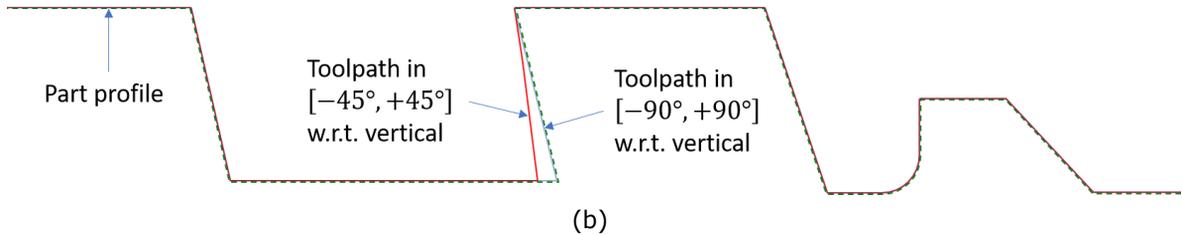
Toolpath with strategy – constant angle ( $10^\circ$  to vertical), Range  $[-45^\circ, +45^\circ]$  w.r.t. vertical

**Figure 17:** Toolpath at constant angle ( $10^\circ$ ) with circular insert in  $[-45^\circ, +45^\circ]$  for part profile 2.

**7.1 Effect of Rotation Range**

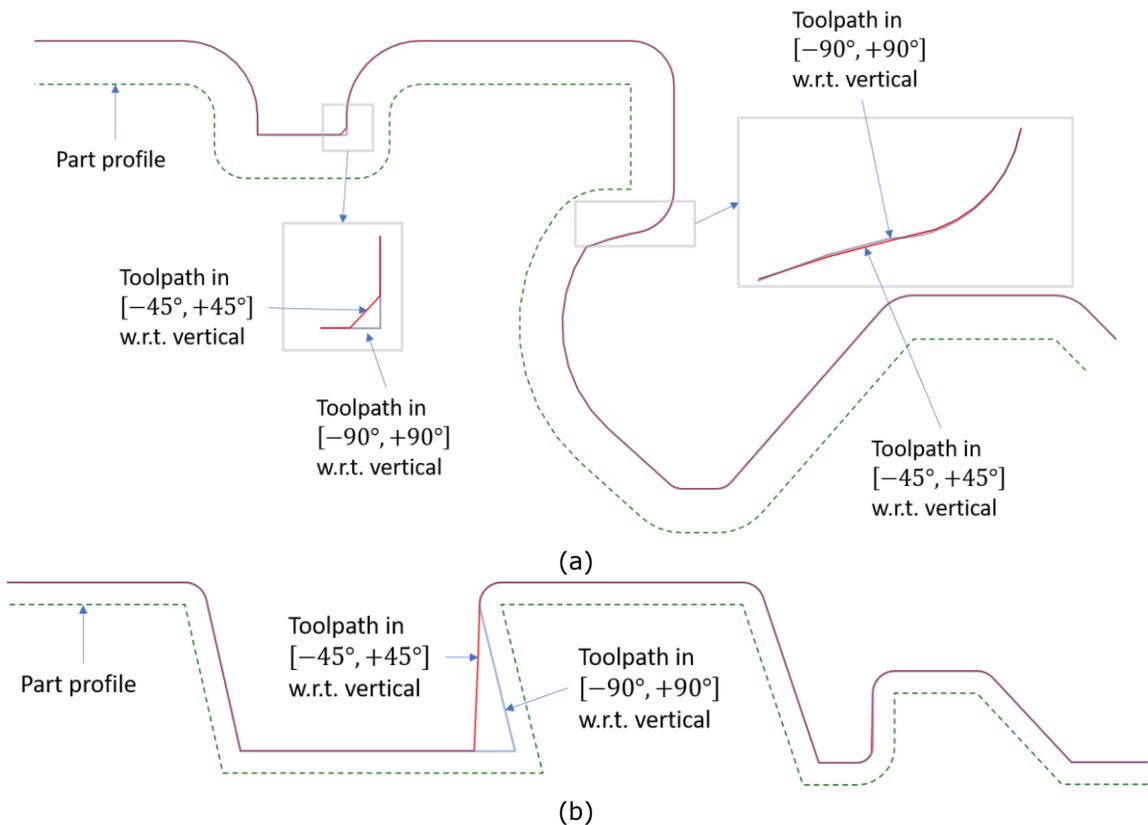
The toolpaths generated with two different ranges of rotation is shown in Figure 18(a) and (b) for part profiles 1 and 2 respectively, both with a diamond insert. The ranges of rotation used are  $[-45^\circ, +45^\circ]$  to the vertical and  $[-90^\circ, +90^\circ]$  to the vertical. The results with a circular insert for the same two-part profiles is shown in Figure 19(a) and (b) respectively.





**Figure 18:** Effect of angle limits on toolpath – (a) for profile 1, (b) for profile 2 for  $[-45^\circ, +45^\circ]$  and  $[-90^\circ, +90^\circ]$  range and diamond insert. Machined region increases with range, as intuitively expected.

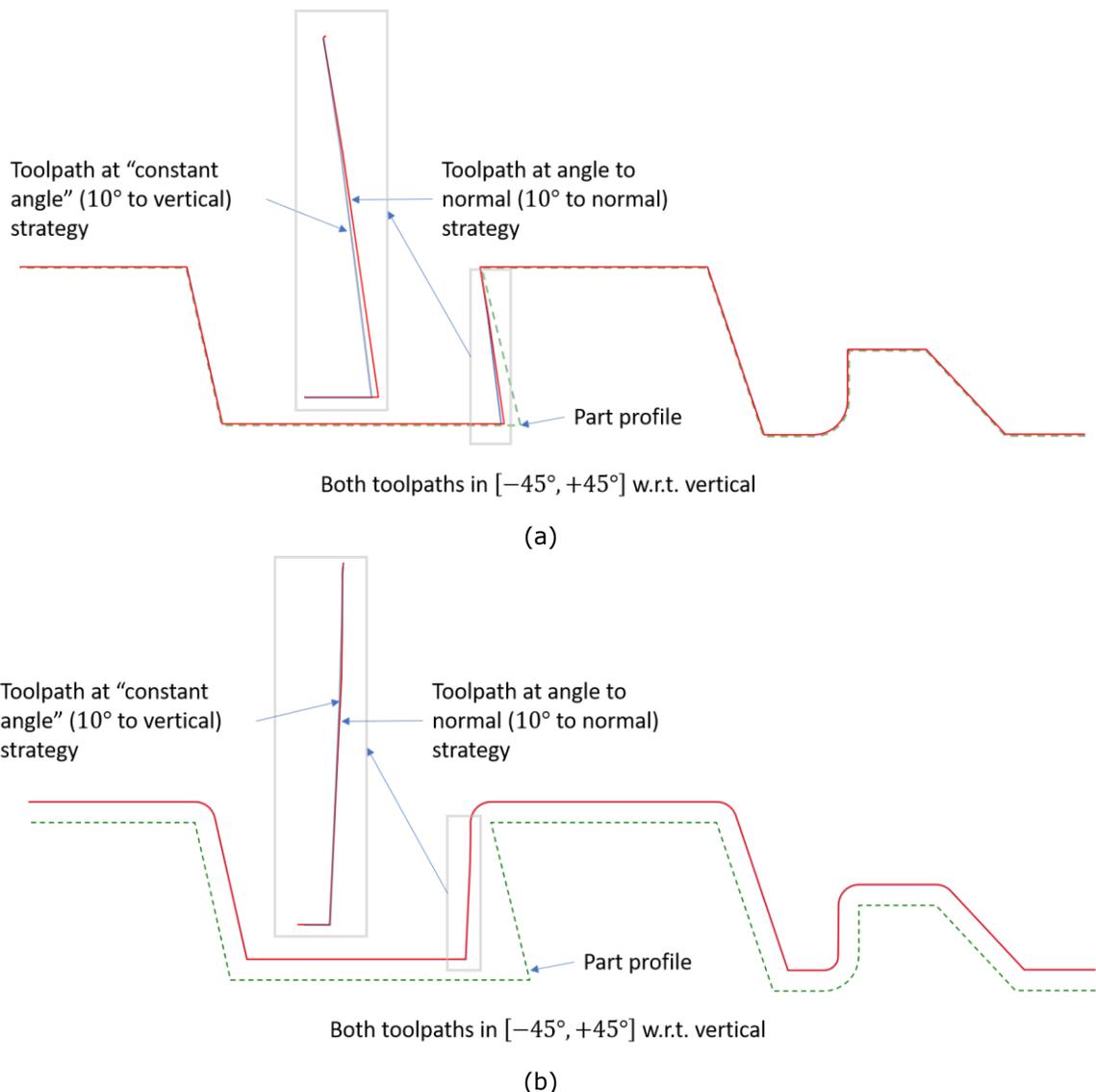
The results show that the extent of portions skipped over by the toolpath is larger if the rotation range is narrow. Further, the differences are mainly seen in the regions that were unmachinable in the initial toolpath. For the circular insert, however, the difference is miniscule, as the extent of unmachinable regions itself was small in the initial toolpath. Intuitively, it is expected that a wider range of rotation would allow the tool to assume a better angle to reach a region; this is seen in the results.



**Figure 19:** Effect of angle limits on toolpath – (a) results for profile 1, (b) profile 2 in  $[-45^\circ, +45^\circ]$  and  $[-90^\circ, +90^\circ]$  ranges for a circular insert. Differences similar to Figure 18, with lesser change in (a).

## 7.2 Effect of Strategy of Rotation

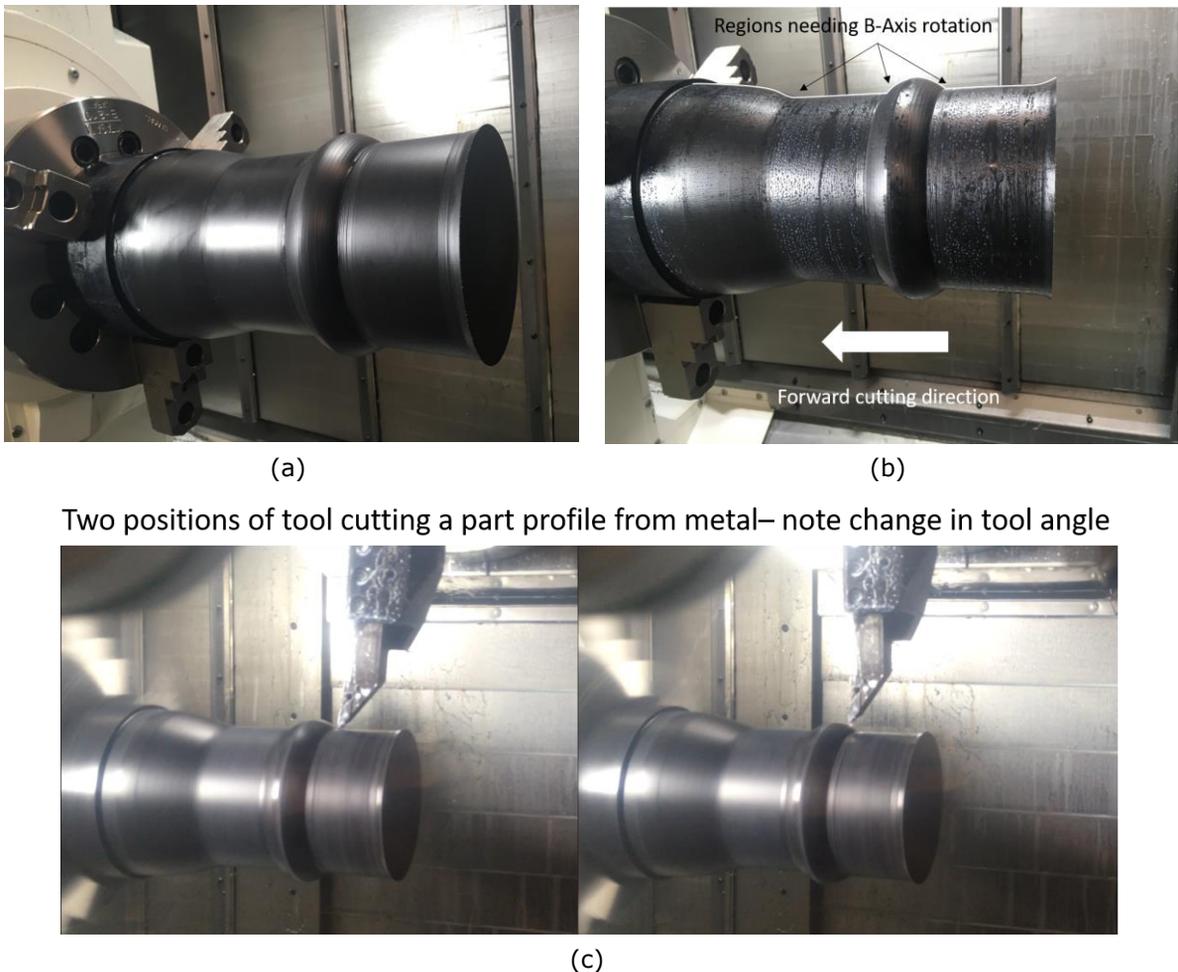
The toolpaths obtained under the two strategies of rotation, namely being at constant angle and being at an angle to the normal to the toolpath are now compared. Figure 20(a) compares the toolpaths under being at  $10^\circ$  to normal and being at a constant angle of  $10^\circ$  for part profile 2. Both the toolpaths are for angle limits of  $[-45^\circ, +45^\circ]$ . Figure 20(b) compares the toolpaths for the same parameters with a circular insert for part profile 2. There is not much difference seen in the toolpaths under the two strategies with either type of insert. This could be due to the fact that both the strategies do not prescribe a particular tool angle at the unmachinable regions but only differ in the desirable angle specified in the machinable regions. The lack of difference in toolpaths could also be specific to the input shapes and parameters used; further experimentation is necessary to determine if the strategies can produce different toolpaths.



**Figure 20:** Comparison of results with at constant and at angle to normal strategies for profile 2 – (a) diamond, and (b) circular insert. No significant differences seen; similar results seen in profile 1.

### 7.3 Cutting of Actual Parts

The toolpaths generated with the presented method were used to generate the NC-files for cutting actual part shapes on machines with continuous B-Axis rotation capabilities. The part profile of a view-scope was cut from a stock that had been subjected to roughing operations. The result is shown in Figure 21(a). In Figure 21(b), the part-profile is highlighted and portions requiring tool rotation are indicated; these are typically regions “occluded” by other regions of the part shape in the direction of cutting, or those having a difference in inclination from that of the tool’s leading edge. The change in tool angle is seen in Figure 21(c) where two nearby positions of the tool are shown. As expected, the tool tilts when climbing the curved region in the middle. Another example of a profile cut from wax is shown in Figure 22, again showing the difference in tool angle between nearby positions on the profile.



**Figure 21:** Example of part cut from metal using toolpath from the presented method – (a) final part, (b) profile with regions needing tool rotation, and (c) two positions showing change in tool angle.

Two positions of tool cutting a part profile in wax – note change in tool angle



**Figure 22:** Example of part cutting from wax stock. Two positions of tool are shown with clear change in tool angle during operation.

#### 7.4 Discussion on Computational Complexity

The number of iterations of sampling and covering required depend on the extent of the unmachinable region, as fewer number of unmachinable facets can be covered in lesser number of iterations. This is seen in the example of profile 1 (Figure 12) vs. profile 2 (Figure 13). The computational cost of the method is mainly from the computation of safe angles through iterative interference check for each facet sampled toolpath in any iteration. Broadly, the computational cost may be expressed as  $t_n * \theta_{iter} * n_{part} * n_{tool}$ , where  $t_n$  is the number of facets in the toolpath, and  $\theta_{iter}$  is the angle change used for the iterative interference check and  $n_{part} * n_{tool}$  is the product of the number of segments in the tool and part segment chains respectively. This cost can certainly be reduced by improving the method of interference check to avoid checking each increment in all cases, performing the interference check parallelly across multiple threads or cores of a processor as each check is independent of the other, and more adaptively sampling the toolpath for safe-angle computation. The authors have implemented some of these improvements and working on the others.

## 8 CONCLUSIONS

The method of toolpath generation presented is simple as the sampling avoids specifying multiple conditions for handling special cases in the geometric computation of safe angles and seems to be robust. The range of allowed tool rotation decides the nature of the toolpath shape and extent of material removed in continuous B-Axis operations more than the strategy of tool angle change, at least with the two strategies considered here. The computational performance of the method can be improved using parallelization of the computation for safe angles, as the interference-check at each tool angle at a point is independent of the same check at other tool angles.

As mentioned in the discussion section 7.4, one ready avenue for future work is the improvement of the computational performance of the method by various techniques including use of parallelization. The evaluation of alternate methods for safe angle computation is another possible direction of future work. The exploration of various other strategies of tool angle change can be done to enumerate the type of toolpath obtained with different strategies; this would of course involve considerations other than of a geometric nature such as the surface finish, smoothness of tool angle change and the dynamical behaviour of the tool and stock under various machining forces. Finally, the authors would also like to extend the method to other types of turning operations such as roughing, layered toolpath generation and so on.

## 9 ACKNOWLEDGEMENTS

The authors would like to thank the following colleagues at HCL Technologies Ltd. for suggesting the problem and improvements to the developed method: Vivek Govekar (General Manager), Nitin Umap (Associate General Manager), Prashant Pawar (Support Specialist), Anand Mutalik (Product Manager) and Baburaj Iyer (Program Director). The actual part-cutting exercises were carried out by Prashant Pawar. The authors also thank the anonymous reviewers for their comments that helped improve the content and its presentation.

*Tathagata Chakraborty*, <http://orcid.org/0000-0002-2752-2533>  
*Hariharan Krishnamurthy*, <http://orcid.org/0000-0002-3059-4132>  
*Manoj Bhonge*, <http://orcid.org/0000-0001-7638-4600>

## REFERENCES

- [1] Anotaipaiboon, W.; Makhanov, S.S.: Tool path generation for five-axis NC machining using adaptive space-filling curves, *International Journal of Production Research*, 43(8), 2005, 1643-65. <https://doi.org/10.1080/00207540412331322948>
- [2] Balasubramaniam, M.; Sarma, S.E.; Marciniak K.: Collision-free finishing toolpaths from visibility data, *Computer-Aided Design*, 35(4), 2003, 359-74. [https://doi.org/10.1016/S0010-4485\(02\)00057-X](https://doi.org/10.1016/S0010-4485(02)00057-X)
- [3] Cuenca, S.; Martínez, A.; Jimeno, A.; Sánchez, J.L.: A hardware/software architecture for tool path computation. An application to turning lathe machining. In *Proc. of 2007 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation, GREECE, 2007*, 96-102. IEEE. <https://doi.org/10.1109/ICSAMOS.2007.4285739>
- [4] Fard, M.J.; Nowak, A.; Francis, B.R.: Device, system and methods for automatic development and optimization of positioning paths for multi-axis numerically controlled machining, United States patent US 9,465,380, 2016. <https://patents.google.com/patent/US9465380B2/en>
- [5] Farouki R.T.; Han C.Y.; Li, S.: Inverse kinematics for optimal tool orientation control in 5-axis CNC machining, *Computer Aided Geometric Design*, 31(1), 2014, 13-26. <https://doi.org/10.1016/j.cagd.2013.11.002>
- [6] Karaman, S.; Frazzoli, E.: Incremental Sampling-based Algorithms for Optimal Motion Planning, arXiv preprint, arXiv:1005.0416. 2010. <https://arxiv.org/pdf/1005.0416>
- [7] Lee S.G.; Hyun Y.T.: Tool path forming method in milling processing system, United States patent US 8,942,843, 2015. <https://patents.google.com/patent/US8942843B2/en>
- [8] Makhanov S.: Optimization and correction of the tool path of the five-axis milling machine: Part 1. Spatial optimization, *Mathematics and Computers in Simulation*, 75(5-6), 2007, 210-30. <https://doi.org/10.1016/j.matcom.2006.12.009>
- [9] Sarma, S.E.; Laxmiprasad, P.; Balasubramaniam, M.: Tool path generator for computer aided manufacturing, United States patent US 6,311,100, 2001. <https://patents.google.com/patent/US6311100B1/en>
- [10] Tang, T.D.; Bohez, E.L.; Koomsap P.: The sweep plane algorithm for global collision detection with workpiece geometry update for five-axis NC machining, *Computer-Aided Design*, 39(11), 2007, 1012-24. <https://doi.org/10.1016/j.cad.2007.06.004>
- [11] Wang, N.; Tang, K.: Automatic generation of gouge-free and angular-velocity-compliant five-axis toolpath, *Computer-Aided Design*, 39(10), 2007, 841-52. <https://doi.org/10.1016/j.cad.2007.04.003>