Multiple tool path planning for NC machining of convex pockets without islands

G. Vosniakos*, P. Papapanagiotou

Department of Mechanical Engineering, National Technical University of Athens, Manufacturing Division, 157 80 Zografou, Athens, Greece

Abstract

This work focuses on machining of pockets with convex angles and containing no islands. The approach is a hybrid contouring – staircasing one. Contouring aims at clearing enough space around the boundary to allow efficient staircasing in the interior. Three tools are foreseen: two for contouring and one for staircasing. The algorithm for contouring is based on creating offsets of the pocket boundary and checking them for self-crossing. Staircasing is based on parametrisation of the tool path with respect to its orientation to the pocket, so as to minimise the tool path length. Machining strategy is complemented by choosing the best combination of diameters of the three tools used. This is simply done by first enumerating all available combinations and then excluding the non-feasible ones, in order to compare the rest according to tool path length (or to machining time when individual feed values are known). The output of the program, which was implemented in Fortran, is the tool path, the CNC part program, which is created automatically, and a numerical comparison of all the tool-angle combinations tried out. A machining simulation based on the CNC program output is conducted on commercial CAM software to demonstrate validity of the result.

Keywords: CNC; Pocketing algorithm; Tool path; Staircasing; Contouring

1. Introduction

Pocketing is a very common machining process referring to the creation of an inner empty volume starting from a component face. In some cases, the empty volume contains protrusions known as “islands”. Pocketing has similarities to face milling, but also differs from it in at least two important respects:

- To begin machining the tool has to move and cut in a direction normal to the pocket entry. This cutting action is commonly called plunging and consequently such capability is required of the tool.
- The pocket walls constrain the tool path all around causing an additional clearance problem at the corners, which always have a finite radius.

Generic pockets consisting of non-planar faces and continuous variation of their depth may be termed “3D”, while those with planar entry and bottom faces and walls being normal to those faces are termed “2-1/2-D”.

Although the pocket walls can be planar, cylindrical or, in general, any part of a tabulated cylinder surface, for the sake of simplicity in this work they are assumed to be only planar connected by cylindrical radii. In addition, the radius connecting the bottom surface of the pocket to its walls is ignored. Such pockets may be thought of as the extrusion of a polygon (with radii connecting adjacent edges) along a straight line normal to it. The polygon in this work is considered to be convex and no islands are allowed. Under those circumstances, it is attempted to find the combination of tools that can clear the pocket volume as quickly as possible which is equivalent to tool paths with minimum length. The main strategies for pocket machining are [1]:

- “staircasing”, where the volume is cleared with a series of parallel passes successively taken in opposite directions;
- “offsetting”, where the volume is cleared with a series of closed contour passes which are produced as inner offsets of the original pocket wall contour.

In this work, a combination of these strategies is employed. Other research work has also been based on those strategies as follows.
Jamil [2] tackles both convex and concave pockets and finds a sub-optimal continuous tool path of the staircasing type. The reference direction to which the staircase is parallel is that of a suitable edge of the polygon. There are cases in concave polygons where there is no suitable edge and as a result the polygon is subdivided into convex ones. Suitability of an edge consists in the distance of just one vertex from that edge being larger than the distances of its neighbouring vertices from the same edge.

Lakkaraju [3] deals with convex polygons and only with facing operations. He computes the total tool path length with varying direction angle of the tool path (staircasing); for small enough step of this angle an optimum is approached with good enough precision. Note that in that work a triangle is employed as the only field of experimentation but offered a good basis for others to expand upon. Sun and Tsai [4] improved Lakkaraju’s approach in that they provided an additional tool path at the end of each pass where otherwise material would have been left unmachined. In addition, a final pass is foreseen, which is usually short compared to the previous ones and aims to remove a remaining “cusp” region. Such improvement resulted in a tool path 17% shorter than Lakkaraju’s. Prabhu [5] generalised from a triangle to a convex polygon and divides the toolpath into sections along the perimeter and sections in the interior of the polygon. The sections along the perimeter are discriminated into those exploiting the full tool diameter and those exploiting just part of it. Analytical calculations are implemented in a computer program. Note that all three approaches concern face milling, where the boundaries of the polygon present no physical constraint to the movement of the tool.

In contrast to the above, Kamarthi [6] computes the boundary points of each pass as defined by the polygon boundaries and the material clearance necessity, i.e. the tool needs move only up to the point where enough material has been removed, this point not necessarily lying on the polygon boundary. Optimisation is studied by rotating the polygon with respect to the tool in suitable small steps from 0 to 360°. Note that scanning the tool path from left to right gives a different length compared to scanning from right to left, the difference being due to the connecting segments of the parallel passes.

Although staircasing was the first pocketing strategy to be investigated, also named “zigzagging” [7], offsetting by definition achieves perfect results in terms of tool path continuity for pockets with or without islands and is based on sound theoretical ground [8]. The standard practice is to construct the Voronoi diagram of the polygon, i.e. a shape dividing the polygon interior into regions, each of which contains points whose distance from a polygon edge is less than their distance from any other polygon edge. The lines of the Voronoi diagram serve as “turning boundary” for the successive offsets of the polygon boundary, which make up the tool path. This approach is best illustrated in [9]. To cater for islands the so-called “contour bridges” are used to connect the outer and the inner boundaries in order to construct a unified Voronoi diagram (see [10]). If closed loops are formed after the addition of multiple contour bridges they can be detected and removed (see [11]).

In summary, it can be stated that most work on the staircasing side dealt with face milling; work dealing with pocketing employs mostly Voronoi diagrams and is not concerned with optimisation issues. Multiple diameters of cutting tools are not exploited either. In what follows, a combined offsetting-staircasing strategy is presented making use of more than one tools (diameters) and aiming at near-optimisation of the tool path length.

2. The algorithm

The problem is stated as follows:

Given

- a convex polygon with fillet radii and
- a set of \( n \) available pocket machining tools with specific diameters

with the restrictions of

- one contouring (offsetting) pass along the polygon to create the fillets with tool \( T_1 \)
- zero or more additional contouring (offsetting) passes along the polygon with one tool \( T_2 \) to create room for volume removal by staircasing
- exactly one staircasing tool \( T_3 \) and corresponding path

find

- the best combination of tool diameters \( d_1, d_2, d_3 \) and
- the orientation of the staircasing path

so as to

minimise total toolpath length (or in a variation: total machining time).

2.1. The approach

Pocketing here refers to convex shapes. If the pocket’s cross-section were concave, it would have to be divided into a number of convex polygons. This is a well-known problem in computational geometry and there exist a number of algorithms and even their computer implementation that can be readily used [12]. The strategy to machine sequentially those adjacent convex regions is not as straightforward, but it falls outside the scope of this paper.

The same applies to pockets with islands. Suitable placement of island “bridges” would essentially divide the machinable region into a number of convex polygons.
that could be individually tackled with the algorithm presented in this work. Again, the sequence and strategy of machining would need elaboration.

The approach devised makes use of multiple tools and includes two steps:

- first, a clearance of constant width is created all along the polygon boundary, a process referred to as contouring. Two tools can possibly be used, \( T_1 \) and \( T_2 \), with diameters \( d_1 \) and \( d_2 \). In general, these are distinct, but there may be cases where they are actually one and the same tool. Diameter \( d_1 \) is smaller than \( d_2 \). The corner radii of the polygonal shape dictate the former, whereas the latter is necessary in order to finish off the contouring phase fast;
- second, the reduced area is cleared with passes parallel to a certain direction, a process referred to as staircasing. This is conducted with tool \( T_3 \), with diameter \( d_3 \) which may be any tool available in the machine magazine except \( T_1 \). The diameter \( d_3 \) and angle of the passes \( \varphi \) need to be optimised.

The innovation of the approach lies in:

- Combination of staircasing and offsetting strategy rather than using a single strategy.
- Near optimisation with multiple tools rather than a single tool.
- Use of realistic conditions, i.e. within a set of available tools, and derivation of a ready to use practical result, i.e. NC pocketing code.
- Machining starts by plunging tool \( T_1 \) into the unmachined pocket tangent to the pocket boundary at the first point given. The subsequent operations are explained next.

2.2. Contouring

When clearing a polygon by parallel passes (i.e. “staircasing”) there are always “cusps” left over, i.e. curved triangular shapes, which are produced because the tool has to stop at the polygon boundary. The height \( h \) of the cusp (normal to the polygon boundary) left over by a tool of diameter \( d \) when moving at an angle \( t \) to the normal to the boundary (see Fig. 1) is calculated as

\[
h = d(1 + \sin t)/2. \tag{1}
\]

When the tool moves parallel to the boundary (\( t = \pi/2 \)) a maximum height occurs: \( h = d \). These cusps would have remained, there should be no contouring before (or after) staircasing. Therefore, contouring is the natural complement of staircasing when the latter is the strategy chosen for pocket machining.

When staircasing is to be effected with tool \( T_3 \), the tool path will be at various angles to the different edges of the polygon, thus giving various cusp heights ranging, according to Eq. (1), from 0 to \( d_3 \). Bearing in mind, the need to examine a multitude of pass directions, typically covering the range 0–360° in steps of \( 1° \) or less, it can be justified that the contouring necessary to complement staircasing can be selected to have a priori a minimum width of \( d_3 \). This also facilitates entry of the staircasing tool through the clearance achieved perimetrically by the contouring operation.

In summary, the contouring width of \( d_3 \) is achieved by first using tool \( T_1 \) with diameter \( d_1 \) equal to twice the minimum radius of curvature of the polygon corner radii. Tool \( T_1 \) follows exactly the profile of the polygon boundary, including corner radii. If \( d_1 \) happens to be equal to \( d_3 \), the staircasing diameter, then no other tool is needed, otherwise a further tool, \( T_2 \), is chosen for successive offsetting so as to achieve an overall contouring width of at least \( d_3 \). Note that the choice of tool refers to a tool bank specific to a particular facility (machine, shop floor, etc.). Tool \( T_2 \) is chosen from the tool bank in increasing magnitude of diameter.

Offsetting always aims at achieving enough clearance for the next tool to fit into the “gap” and start the next operation (this being either successive offsetting with tool \( T_2 \) or staircasing with tool \( T_3 \)).

Offset calculation is simple. The offset of polygon point A to a new point B is effected along the bisector of \( \alpha \), the polygon angle at A. If the edge starting at A, with a clockwise sense of traversal of the polygon, forms an angle \( \alpha \) with the horizontal axis and the offset (contouring) width is \( h \), (see Fig. 2) then the coordinates of point B are...
calculated as

$$X_B = X_A + h \cdot (\cos(\alpha - \omega/2)/\sin(\omega/2)),$$

$$Y_B = Y_A + h \cdot (\sin(\alpha - \omega/2)/\sin(\omega/2)).$$

(2)

The contouring tool path for a given tool diameter is defined as a series of coordinate pairs as in Eq. (2). When the centre of tool $T_z$ is on an offset vertex a cusp will be left (see Fig. 3). This was pointed out also in [9], and was remedied by an auxiliary movement of the tool towards the initial vertex by

$$h' = d/2 \cdot (1/\sin(\omega/2) - 1).$$

(3)

This auxiliary movement is necessary at all offset vertices except the first offset vertex, which is reached from the direction of the cusp anyway.

Whenever a tool change is necessary the new tool is placed one fraction of its radius away from the first point of the contour to be machined (along the bisector of the respective angle) so as to avoid rubbing (burnishing) against the unmachined surface.

2.3. Staircasing

This phase refers to machining away with parallel passes the material that remained after contouring executed first. This material forms a protrusion with a polygonal base and has a clearance around it of at least one tool diameter $- d_3$.

2.3.1. Parameterisation

Assuming that the parallel passes form an angle $\phi$ with the positive $x$-axis, they can be represented as a parametric line family with equation

$$y = \tan(\phi) \cdot x + N \cdot d_3/\cos(\phi),$$

(4)

where $N$ is a real number used to parameterise the constant of the line equation with respect to the tool diameter (see Fig. 4(a)). Consequently, any polygon vertex can be parameterised by a single $N$ value denoting the line passing through it (see Fig. 4(b)). Different $N$ values correspond to different directions of the passes. The total number of passes necessary to machine the polygon is equal to the maximum difference $N_i - N_j$ where $i$ and $j$ are pointers to any possible polygon vertex. This assumes, of course, that the pass width is equal to the tool diameter.

Note that the lines binding the tool diameter correspond to two parameters $N_0$ (upper limit line of pass) and $N_U$ (lower limit line of pass) whose difference is 1 (see Fig. 4(c)). The line passing through the tool centre corresponds, then, to $N_K = (N_0 + N_U)/2$. Therefore, $N_0 = N_K + 0.5$ and $N_U = N_K - 0.5$. Also, note that for any pair of consecutive passes the $N$ values of the lines passing through the respective tool centres differ by 1.

In the case $\phi = 90^\circ$ the parametric line family is defined as

$$x = N \cdot d_3.$$  

(4)

2.3.2. Critical points

This term refers to characteristic points of the polygon that are used to calculate the passes. These are:
the intersection points of the upper and lower pass limit lines with the polygon boundaries. A polygon edge AB intersects the upper limit line only if $N_A < N_0 < N_B$ or if $N_A < N_0 < N_B$. The intersections are found by simply solving the system of intersecting line equations. The same applies to the lower limit line intersections.

- the vertices of the polygon that lie between the upper and lower limit lines of a pass. A vertex A qualifies as a critical point if: $N_{U} < N_{A} < N_{0}$.

### 2.3.3. Pass delimiting points

These should be calculated so as to avoid any cusps being left over when full use of the tool diameter is made (i.e. there is no overlap between consecutive passes). There is always a left delimiting point $(x_L, y_L)$ and a right delimiting point $(x_R, y_R)$.

For a critical point $P_I$ with line parameter $N_I$ which falls within the pass whose tool centre line has parameter $N_K$, any possible delimiting point on the line $N_K$ will be one tool radius (i.e. $r_3$) away from $P_I$. There are three cases, in each of which calculation of the coordinates of the right and left delimiting points are slightly different (see Fig. 5). The calculated expressions are shown in Table 1. One of the coordinates is always calculated, the other one being derived from the line equations (3) and (4) as required.

For any pass there may be a number of critical points each of which gives a pair of delimiting points. Two sets are formed like this, one for potential left delimiting points and another one for potential right delimiting points. The ultimate two delimiting points are chosen from the respective sets as the one with minimum $x_L$ coordinate and maximum $x_R$ coordinate when $\phi \neq \pm 90^\circ$, otherwise they are chosen as the ones with minimum $y_L$ and maximum $y_R$ coordinate, respectively.

### 2.4. Tool path calculation

In the case of contouring, the tool path is derived straightforwardly as the polygon line connecting the consecutive offset points for tool $T_1$. For tool $T_2$ the path is calculated as for tool $T_1$ plus the auxiliary segments introduced to clear the cusps at corners. This is repeated as many times as there are offsets.

In subsequent staircasing, the tool path is a kind of meander defined by the pass delimiting points. There are two cases, then the meander can be formed starting from...
Table 1
Calculation of delimiting point coordinates for critical point \((x_c, y_c)\) as a function of pass angle \(\varphi\) (see Fig. 5)

<table>
<thead>
<tr>
<th>Condition</th>
<th>(x_L)</th>
<th>(x_R)</th>
<th>(y_L)</th>
<th>(y_R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-90^\circ &lt; \varphi &lt; 90^\circ)</td>
<td>(x_L = x_L - (d_3/2) \cos(b + \varphi))</td>
<td>(x_R = x_R + 2 \times (AB) \cos \varphi)</td>
<td>(y_L = y_L - (d_3/2) \cos(\varphi + b))</td>
<td>(y_R = y_R + 2 \times (AB) \cos \varphi)</td>
</tr>
<tr>
<td>(N_1 &gt; N_k)</td>
<td>(x_L = x_L - 2 \times (AB) \cos \varphi)</td>
<td>(x_R = x_R + (d_3/2) \cos(b + \varphi))</td>
<td>(y_L = y_L + (AB))</td>
<td>(y_R = y_R - (d_3/2) \cos(\varphi + b))</td>
</tr>
<tr>
<td>(90^\circ &lt; \varphi &lt; 270^\circ)</td>
<td>(x_L = x_L + 2 \times (AB) \cos \varphi)</td>
<td>(x_R = x_R - (d_3/2) \cos(b + \varphi))</td>
<td>(y_L = y_L + (AB))</td>
<td>(y_R = y_R - (d_3/2) \cos(\varphi + b))</td>
</tr>
<tr>
<td>(N_1 &lt; N_k)</td>
<td>(x_L = x_L + 2 \times (AB) \cos \varphi)</td>
<td>(x_R = x_R - (d_3/2) \cos(b + \varphi))</td>
<td>(y_L = y_L - (d_3/2) \cos(\varphi + b))</td>
<td>(y_R = y_R + (AB))</td>
</tr>
</tbody>
</table>

Where \((AB) = (d_3/2) \times (1 - 4 \times (N_k - N_i)^2)^{1/2}\) and \(b = \arccos((AB)/(d_3/2))\).

a left or a right delimiting point. The resulting path length is different in the two cases, because of the lateral connections of the pass lines, which, in the general case, are different left and right. Every time the path chosen is the shorter of the two.

The entry point of the tool is found by considering the distances of all polygon vertices from the starting point of the staircasing path. The vertex with shortest distance is offset by \(d_3/2\) and the offset point serves as entry of the plunging tool.

The steps of the contouring–staircasing algorithm presented above are depicted in Fig. 6.

2.5. Tool combinations

Tool \(T_1\) is always necessary to cater for the corner radii of the pocket. This accounts for the first contouring operation.

The subsequent contouring operation by tool \(T_2\) is necessary to create enough clearance for staircasing with tool \(T_3\) of diameter \(d_3\). Tool \(T_2\) may need to perform more than one contouring passes, but usually less than three because there should be a larger tool that could achieve the same clearance in one or two passes. This, of course, depends on the availability of tools with appropriate diameters; therefore, tools are chosen from a pool of available ones, rather than finding the most appropriate diameter with no reference to availability. Note that passes by tools \(T_1\) and \(T_2\) may reduce to just one pass by a single tool, if fillet radii are possible to create with the tool that also creates room for subsequent staircasing.

Tool \(T_3\) performs near-optimum staircasing when its diameter is appropriate and also the pass angle is appropriate.

Near-optimisation, then, is performed in the following stages:

- First an enumeration of the possible combinations of available tool diameters \(d_1\), \(d_2\) and \(d_3\) is done.
- Then, all non-feasible tool choices for contouring are excluded. This is a subtle point concerning the offset distance which is technically acceptable in contouring, i.e. the maximum tool diameter that can perform off-setting. This is examined by performing an offset of the original polygon. If the offset distance is very large, at least one edge of the resulting offset polygon will have a slope opposite
to that of the corresponding original polygon edge (see Fig. 7).
- Last, for each feasible combination \(d_1, d_2\) and \(d_3\) the shortest (contouring and offsetting) tool path is found and the optimum combination is the one giving the shortest path overall.

2.6. Automatic derivation of NC program

The NC program used to machine the pocket is automatically produced as ISO code by the software program written, according to the following rules:

- Block numbering is added in user-defined increments.
- The program starts with seven standard lines of ISO code naming the program and performing routine initialisation, e.g. coordinate system choice (metric, absolute), cancellation of compensations and canned cycles, etc.

- Each time a new tool is needed, a standard tool-changing line is added, followed by a rapid movement above the tool’s entry point in a safe \(Z\) plane, followed by a tool length compensation line and then by slow plunging to the correct \(Z\) height.

- For each tool all \(X-Y\) coordinate pairs are output in succession, occupying one line each. Standard Feed and Speed values are used that can be edited by the user. After reaching the exit point each tool will move vertically up to the safe \(Z\) height (one line of code) from where a new tool change statement can be effected.

- Three standard lines close the program (imaginary tool, end of program, end of character).

- The resulting code (see Appendix A), can be used for simulation in a CAM system or downloaded to a machine and executed.

3. Results and discussion

The approach was implemented in Fortran. Microsoft Fortran PowerStation v.4.0 was used. The executable code was about 220 KB and the source code about 1000 lines. The tool path results are also stored by the software in Grapher format, so as to facilitate pictorial
representation. The NC programs created were loaded into Alphacam, a commercially available CAM program, to simulate machining of the corresponding pockets. The numerical results (pass angle and tool path length for the combinations of \(d_1, d_2\) and \(d_3\) were loaded into Excel to visualise variations and trends. Various polygonal shapes were tried: triangles, rectangles, pentagon and hexagon shapes with and without symmetry. A representative case is presented here.

The shape machined is a random hexagon, (see Fig. 8). Corner radii of 3 mm dictate \(d_1 = 6\) mm. The second tool for contouring is chosen from the range \(d_2 = 6, 12, 16\) and 20 mm. The staircasing tool is chosen from the range \(d_3 = 16, 20, 30\) and 40 mm. Table 2 summarises the results of optimisation of tool diameter and staircasing pass angle.

In Table 2, it is observed that for each combination of diameters \(d_1, d_2\) and \(d_3\) there is a shortest (optimum) and a longest path, depending on the angle of the staircasing path. In some cases, for the same combination of tools, there are more than one optimum angles of staircasing, which give the same – shortest – path length, but physically different paths. On the average, the relative difference of the length of these paths is about 5%. This is the maximum error one would end up with if no attention were paid to the orientation of staircasing.

The tools available for staircasing are graphically compared as for the tool path length they achieve with a variation of path angle from 0 to 360° in diagrams of the type shown in Fig. 9.

The maximum error one would make when no attention were paid to the combination of the three tools \(T_1, T_2\) and \(T_3\) would be according to Table 2 about 53.5%. The best combination in our case is \(d_1 = 6\) mm, \(d_2 = 12\) mm and \(d_3 = 20\) mm. A typical tool path is given in Fig. 8 and the part of the CNC code automatically output by the system is shown in Appendix A. This was also simulated on Alphacam and the results are depicted in Fig. 10.

Note that from the many approaches found in international bibliography the most influential one to this work, at least as far as the staircasing path is concerned, was Kamarthi's [6]. However, due to the parametrisation technique of the passes polygon transformation for the different pass angles was avoided, thereby saving computation time. Also, in order to allow for application of such an algorithm initially devised for facing applications suitable contouring operations had to be added.

The program written has obvious applications in process planning and part programming when machining time is at a premium. The commercially available CAM systems usually calculate the tool path for the tool specified by the user. There is no guidance as to which tool diameter would give the shortest tool path. The only help possibly obtainable in the form of a warning message refers to cases where the tool put forward by the user...
### Table 2

Tool and angle results for hexagonal pocket of Fig. 8

<table>
<thead>
<tr>
<th>Tool diameter (mm)</th>
<th>Optimum tool path length (mm)</th>
<th>Largest tool path length (mm)</th>
<th>Optimum staircasing angle (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1040.847</td>
<td>1101.533</td>
<td>5.83</td>
</tr>
<tr>
<td>20</td>
<td>1121.014</td>
<td>1172.077</td>
<td>4.56</td>
</tr>
<tr>
<td>30</td>
<td>1178.344</td>
<td>1227.223</td>
<td>4.15</td>
</tr>
<tr>
<td>40</td>
<td>1358.043</td>
<td>1422.147</td>
<td>4.72</td>
</tr>
<tr>
<td>12</td>
<td>936.746</td>
<td>987.606</td>
<td>5.43</td>
</tr>
<tr>
<td>20</td>
<td>926.674</td>
<td>977.737</td>
<td>5.51</td>
</tr>
<tr>
<td>30</td>
<td>974.485</td>
<td>1031.373</td>
<td>5.84</td>
</tr>
<tr>
<td>40</td>
<td>1107.544</td>
<td>1171.205</td>
<td>5.75</td>
</tr>
<tr>
<td>16</td>
<td>1040.847</td>
<td>1101.533</td>
<td>5.83</td>
</tr>
<tr>
<td>20</td>
<td>1031.412</td>
<td>1067.303</td>
<td>3.48</td>
</tr>
<tr>
<td>30</td>
<td>1021.105</td>
<td>1072.546</td>
<td>5.04</td>
</tr>
<tr>
<td>40</td>
<td>1175.313</td>
<td>1227.321</td>
<td>4.43</td>
</tr>
<tr>
<td>20</td>
<td>1131.085</td>
<td>1181.946</td>
<td>4.50</td>
</tr>
<tr>
<td>20</td>
<td>1121.014</td>
<td>1172.077</td>
<td>4.56</td>
</tr>
<tr>
<td>30</td>
<td>1163.320</td>
<td>1213.422</td>
<td>4.31</td>
</tr>
<tr>
<td>40</td>
<td>1163.320</td>
<td>1225.306</td>
<td>5.33</td>
</tr>
</tbody>
</table>

Fig. 9. Variation of tool path length for $d_1 = 5$ mm, $d_2 = 10$ mm and $d_3 = 10$, 20 or 40 mm.

cannot achieve the desired pocket geometry. Tool path orientation is also poorly handled, i.e. usually one of the “main” directions of the pocket (whatever that means) is taken for determining the staircasing angle. Finally, multiple tool combination suggestions are not touched upon either. Therefore, a random but feasible choice of tools by
A final point worth discussing is that the choice of tool combinations can be based not on tool path length, but on machining time. Machining time per tool can be derived from tool path length by just dividing by feed (per minute) which, at simplest, can be calculated according to manufacturer's recommendations. Adding up machining time for all tools used gives total machining time which – again – should be minimum. This criterion may give different results from those obtained on the minimum tool path length criterion.

4. Conclusion and further work

A combined contouring and staircasing approach for convex pocketing has been presented. Machining of convex pockets is complex enough to justify optimisation, which is affected by the following factors:

- The shape of the pocket (symmetry, corner angles, corner radii, number of vertices).
- Diameter of available contouring and staircasing tools.
- Overall dimensions of the pocket. Large pockets promote large tools and staircasing plays a comparatively more important role in tool path length optimisation.
- Angle of staircasing passes.

The software written can give very accurate results when comparing different combinations of the above factors. In order to avoid time-consuming runs it is best to work stepwise, i.e. first work out staircasing tool diameter with large pass angle step and then work out the exact pass angle, rather than optimising both parameters at once.

Work towards a complete pocketing algorithm is currently being carried out concerning the strategy of breaking down concave pockets with islands into large enough convex polygons, which are machinable with the algorithm presented in this paper.

Appendix A

CNC part program for hexagonal pocket of Fig. 9.

```
START
\(\text{(HEX1)}\)
%
N0001 (FILE NAME = HEX1)
  .0001
N0004 G90 G71
N0005 G78 G40 G80
N0006 M06 T01
N0007 G00 Z5 M03 S1000
N0010 X2.958 Y22.042
N0011 G01 Z-10 F100
```

Fig. 10. Simulation result of the CNC program for hexagonal pocket of Fig. 8. Application of the three tools in sequence.

the inexperienced user might result in twice as long tool paths.

The software developed here can be used in conjunction with a CAM system to help specify tools and staircasing orientation to the point allowed by the system. Otherwise, the NC code resulting after execution of this software can directly substitute the one produced by the CAM system.
N0015 X22.303 Y118.768
...
N0125 X8.873 Y26.125
N0129 Z5
N0130 M06 T02
N0132 G00 Z5 M03 S1000
N0135 X5.916 Y24.083
N0136 G01 Z-10 F100
N0140 X17.747 Y32.249
...
N0220 X17.747 Y32.249
N0224 Z5
N0225 M06 T03
N0227 G00 Z5 M03 S1000
N0230 X70.000 Y61.926
N0231 G0 Z-10 F100
N0235 X32.311 Y113.844
...
N0250 X49.963 Y117.336
N0260 Z5
N0660 T0
N0670 M30
%

References