Automatic Generation of Milling Toolpaths with Tool Engagement Control for Complex Part Geometry

Alexandru Dumitrache [*] **Theodor Borangiu** [*] **Anamaria Dogar** [*]

[*] *Centre for Research & Training in Industrial Control, Robotics and Materials Engineering, University Politehnica of Bucharest*
E-mail: alex@cimr.pub.ro, borangiu@cimr.pub.ro, dogar@cimr.pub.ro

Abstract: This paper presents a NC toolpath generation strategy with tool engagement control for arbitrarily complex discrete part geometry, which reduces machining time and tool wear and can be used in high speed machining. Simulations and comparison with existing methods are presented.

Keywords: NC Adaptive High Speed Machining, Tool Engagement Control, Tool Life Optimization

1. INTRODUCTION

Conventional tool-path generation strategies are readily available for generating geometrically feasible trajectories. Most CAM software offer the option of generating either direction-parallel or contour-parallel toolpaths; while the resulting part is correct from a geometrical point of view, the approaches seldom take into account the milling process conditions.

Handbooks define the *stepover* (1), or *radial depth of cut*, as the percentage of the tool diameter engaged in material (Fig. 1).

$$s = \frac{r}{D} \tag{1}$$

The stepover determines the material removal rate (MRR) and reflects the cutting forces, but only for straight line motions.

A parameter which better reflects the cutting force, regardless of the toolpath shape, is the *tool engagement angle* [1] (TEA), defined by Gasparraj (2005) as the amount of sweep subtended by each cutting edge as it engages and leaves the stock.

For straight line motions, there is a direct, nonlinear relationship between tool engagement angle and stepover, as illustrated in (2), (3) and Table 1, with notations from Fig. 1:

$$s = \frac{1 + \sin(\theta - 90^\circ)}{2}, \qquad 0 \le \theta \le 180^\circ \tag{2}$$

$$\theta = 90^\circ + \arcsin(2s - 1), \qquad 0 \le s \le 1 \tag{3}$$

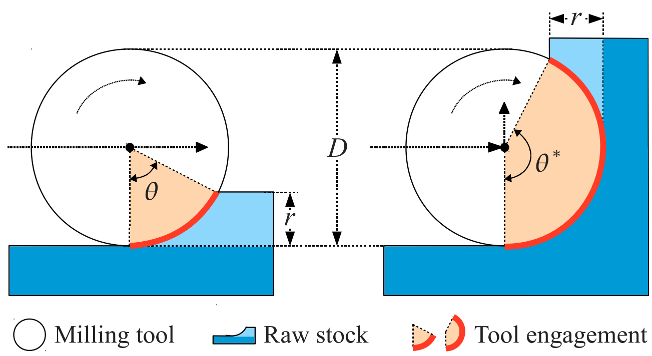


Fig. 1. Stepover and tool engagement angle, in straight line motion (left) and 90°corner (right)

Table 1. Relationship between stepover and TEA

Stepover	10%	25%	50%	75%	85%	95%
Engagement angle	37°	60°	90°	120°	135°	155°

The engagement angle reaches its maximum (360°) when plunging the tool vertically into the material. The next maximum value, 180°, is encountered during a slotting operation; this condition may lead to high thermal stress on the tool, since the chips cannot be evacuated properly. Tool engagement is also known to increase at internal corners in toolpath (Fig. 1, right).

The engagement angle also has direct influence on the chip shape, therefore keeping TEA constant ensures consistent chip size and shape throughout the milling process.

2. RELATED WORK

Coleman (2006) shows that when a tool programmed at a 50% stepover (90° TEA) encounters a 90° corner, the instantaneous engagement doubles to 180°. With small stepovers, the situation is even worse, since for a 10% stepover (37° TEA), the TEA increases inside the 90° corner to 127°, which is a 244% percent increase. Since high speed machining frequently uses small stepover values [2] (Pasko et al., 2002), contour-parallel toolpaths should be avoided.

By enforcing a minimum radius on the toolpath, one will obtain only a minor improvement of the corner TEA (Coleman, 2006).

Bieterman (2001) replaced contour-parallel pocketing paths with a smooth spiral, which is nearly circular at the pocket center and slowly morphs into the part shape as it gets closer to the part. The author reports savings in machining time by up to 30% as compared to conventional methods, and also significantly longer tool life when hard metals are being cut, although he did not use any explicit control of the engagement angle.

Wang et al. (2005) defined a set of quantifiable metrics for evaluating a toolpath quality, which included two main criteria:

- Path curvature, which determines the acceleration and jerk required to track the trajectory;
- Instantaneous cutter engagement, which can be evaluated by geometrical process simulation.

[1] The solid angle of the tool's sector currently engaged in raw material
[2] 35-40% for roughing, 20-40% for semi-finishing, 0.1-0.2 mm for finishing

The authors applied the metric for optimizing spiral-in and spiral-out contour parallel paths for nonconvex pockets. The average tool engagement and path curvature are significantly improved from the conventional case; however, the worst case behavior is not improved at all: their toolpaths still show instantaneous cutter engagements up to 180°.

The same authors proposed a method for evaluating the instantaneous cutter engagement by discrete process simulation. Both the in-process geometry of the raw workpiece and the cutter shape are discretized, by encoding them as bitmap images. The pixels on the raw image can have two states: *empty* and *material_present*, while the pixels on the tool image can be *outside*, *inside* or on the *circumference*.

The simulation translates the tool image along the toolpath and updates the shape of the raw stock; therefore, at every toolpath point, the instantaneous raw stock shape is known. Tool engagement angle is estimated by counting the pixels on the circumference of the discretized tool which have the *unmachined* state on the raw stock image (4):

$$TEA = \frac{N_{\text{circumference \& material\_present}}}{N_{\text{circumference}}} \cdot 360^\circ \qquad (4)$$

Kim et al. (2006) also used a pixel-based simulation of the cutting process for estimating the material removal rate of a given toolpath and for altering a set of contour-parallel toolpaths in order to keep MRR constant.

Stori and Wright (2000) proposed an algorithm for convex contours, which modified an offset tool path such as the engagement is kept constant. Ibaraki et al. (2004) removed the convexity requirement and presented two strategies:

- *forward* tool path generation, where an existing toolpath is offseted along the normal to the advancing direction, in order to reach the prescribed engagement value. The computation employed one Newton step.
- *backward* tool path generation, where the algorithm starts from the innermost offset contour, and computes the precut workpiece contour, such as the engagement remains constant without changing the toolpath.

The forward algorithm works on arbitrary contour geometry, according to the authors. However, when trying to reproduce the experiment, it was found that there is no control in maintaining the milling type (climb or conventional). In certain situations involving complex geometry, the algorithm switched the milling type from climb to conventional or viceversa.

The algorithm computes the engagement angle from the geometric intersection between the tool and part contours. This idea will be extended in this paper; the geometric intersection will be used for computing the direction in which the toolpath will advance, while maintaining a constant tool engagement.

The backward algorithm is basically a non-uniform offseting scheme which enforces constant engagement when the tool cuts along the contour-parallel path. For large tool diameters, relative to the part curvature, the precut surface obtained may require a much smaller tool at the previous step, which limit the usefulness of the strategy in roughing operations.

Uddin et al. (2006) applied the backward generation approach for correcting the semi-finishing paths in order to obtain tighter tolerance on the finished part.

3. CAM SOFTWARE EXPERIMENTS

The parts from Fig. 2 were used in order to evaluate the performance of the existing and proposed milling strategies.

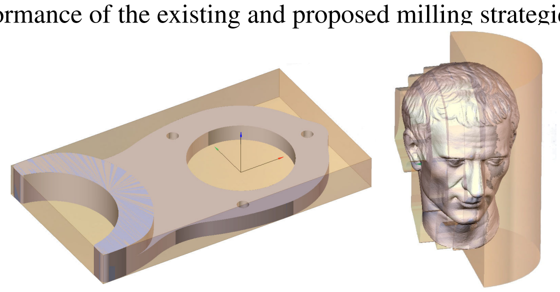


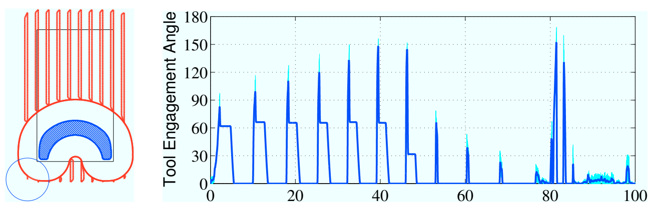
Fig. 2. Test parts for evaluating the milling strategy

For the first part, which has simple geometry, only the upper half will be milled. The bounding box for the part is $90 \times 50 \times 10\text{mm}$ and a $\phi30$ mm roughing tool is used. The inner concavity has a diameter equal to 38 mm, therefore it will cause a significant increase in the engagement angle if a conventional path is used. The opposite side requires a high material removal rate, which will help in evaluating the efficiency of the toolpath.

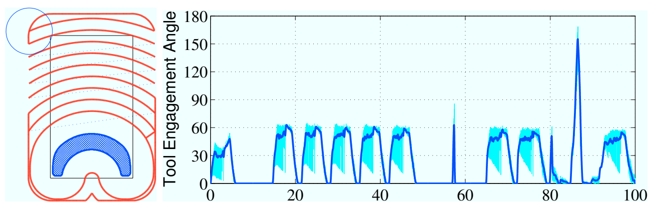
The second part is a complex shape, with many details. For the roughing process, only the bottom layer will be considered.

Current CAM software also employ trochoidal steps for controlling the tool load (Sprut Technology, 2009). The test parts from Fig. 2 were imported into a CAM package, which created contour-parallel toolpaths, with and without trochoidal steps.

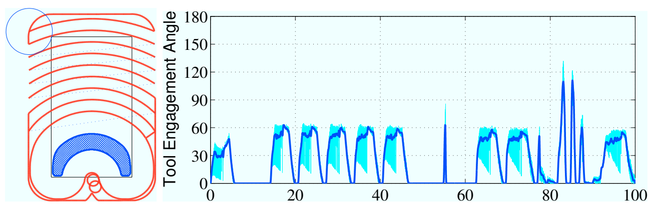
Fig. 3 and Fig. 4 show toolpaths generated with commercially available CAM software. Fig. 3(a) shows direction-parallel toolpaths, which have frequent peaks in tool engagement. Fig. 3(b) used contour-parallel paths, which are much better



(a) Parallel toolpaths, 25% stepover ⇔ 60° TEA

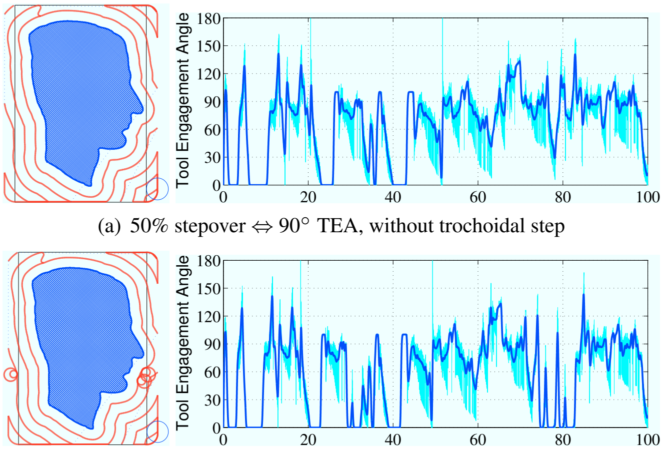


(b) Equidistant toolpaths, 25% stepover ⇔ 60° TEA



(c) Equidistant toolpaths with trochoidal step, 25% stepover ⇔ 60° TEA

Fig. 3. Direction and contour parallel toolpaths generated for the part with simple geometry. At (c), trochoidal machining was enabled in the CAM software.



(a) 50% stepover ⇔ 90° TEA, without trochoidal step

(b) 50% stepover ⇔ 90° TEA, with trochoidal step

Fig. 4. Contour-parallel paths for the complex geometry part

on average, but when the tool reaches the concave region, the engagement spikes up to 170°.

Fig. 3(c) was generated with *trochoidal step*, which according to the documentation of the CAM software, is able to generate toolpaths with constant cutter load Sprut Technology (2009). The graph shows that the evolution of the tool engagement was improved, dividing the peak into three smaller peaks, with a maximum value of 130°. Since the prescribed stepover was 25%, equivalent with a 60°engagement, the result is not impressive. At larger stepover values, the CAM software did not employ the trochoidal steps at all.

Fig. 4(a) used contour-parallel toolpaths for the complex part, and the engagement angle presents many spikes, up to 180°. In fig. 4(b), the trochoidal step was enabled in order to control the engagement angle. The reader may see that extra circles appeared on the tool path; however, the engagement angle simulation shows that the maximum spikes are unchanged. The trochoids did lower the engagement, but in non-critical areas.

The conclusion of these simulations is that the trochoidal step algorithm used by the CAM program provides only a small improvement over conventional paths, and fails to limit the tool engagement when the design part has many contouring details.

The graphics from Fig. 3-4 were obtained by pixel-level simulation of the G-Code output from the CAM software.

On the left of each graphic, the generated XY path is displayed; the circle indicates the starting point and the tool size. The dotted lines show the tool retracts and returns, which are performed using `G0` moves. The dark area is the ideal design part. The plot on the right represents the evolution of the tool engagement angle during the milling process.

4. ALGORITHM DESIGN

The toolpath generation algorithm was designed with the following principles in mind:

- Ensure a upper bound on the tool engagement angle
- Maintain the milling type, climb or conventional
- Minimize the number of plunges
- Keep its properties in the presence of complex geometry

In order to achieve an optimal milling strategy, the toolpath generator has to know the shape and size of the raw stock. The stock shape can be either a cylinder or a box of known size, or can be acquired by 3D scanning.

In a 2D milling step, the raw stock and the part can be represented by two binary depth maps. There are four combinations:

- *white* part, *white* stock: design part
  (raw material which must not be removed)
- *white* part, *black* stock: forbidden areas
  (e.g. for modeling the clamping fixtures)
- *black* part, *white* stock: material which has to be removed
  (where the milling paths will be generated)
- *black* part, *black* stock: free space
  (no material and no restrictions; useful for performing lateral entries into the raw stock)

4.1 General Milling Strategy

The milling toolpath generator can be regarded as a state machine (Fig. 5), with the following states:

- Find Starting Point
- Constant Engagement Milling
- Contour Milling

Obviously, the operation begins in the first state, which searches for a point for begining the milling operation.

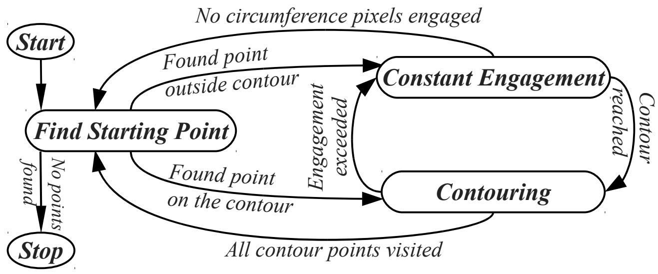


Fig. 5. State diagram for toolpath generation strategy

The second state will advance the cutter into the raw material by maintaining a constant tool engangement value. When the tool touches the design part, it is not possible to continue with the same engagement, because the toolpath would remove material which must be kept in place. When this happens, the toolpath generator will switch to the third state, *contour milling*.

In contour milling, the tool moves along the *offset contour*, being always tangent to the design part. It is necessary for the tool to move on the entire offset contour in order to obtain the correct geometry of the part; however, the maximum engagement angle can be exceeded, reaching values up to 180°.

Therefore, whenever the maximum engagement angle is exceeded by more than a given threshold, let's say 20°, the generator will switch back to *constant engagement* state.

4.2 Geometry discretization

The geometries of the design part, raw stock and tool shape are represented as bitmap images. As Z is constant, only X and Y are considered. The discretization uses a pixel-to-mm ratio, p.

The tool diameter D is discretized using (5):

$$D_D = D \cdot p \tag{5}$$

For simplification, the origin of the workpiece reference frame is mapped to the center of the image.

Therefore, the continuous X and Y are discretized using (6):

$$X_D = [X \cdot p + W/2]$$
$$Y_D = [Y \cdot p + H/2] \tag{6}$$

where $W \times H$ is the size of the depth map image, in pixels.

4.3 Core algorithm: Advancing with constant engagement

The heart of the toolpath generation algorithm is the second state, which generates a constant engagement toolpath, considering only the starting point and raw stock geometry. When the tool touches the part geometry, the core algorithm terminates. This prevents the cutting tool from removing more material than necessary.

A mechanism for computing the direction which maintains constant engagement angle is required. Possible approaches:

- Consider several values of α, evaluate the engagement angle for each value and choose the best among them. Requires many engagement evaluations for a single step.
- Choose α using a nonlinear minimzation approach. The result might change the milling type, from climb to conventional or viceversa.
- Compute α from the intersection point between the tool circumference and the raw part geometry. No engagement angle evaluation is required; the method is much faster and also robust, since it does not change the milling type.

The intersection point between the tool circumference and the part edge (Fig. 6) indicates the advancing direction for a 50% stepover, or 90° engagement.

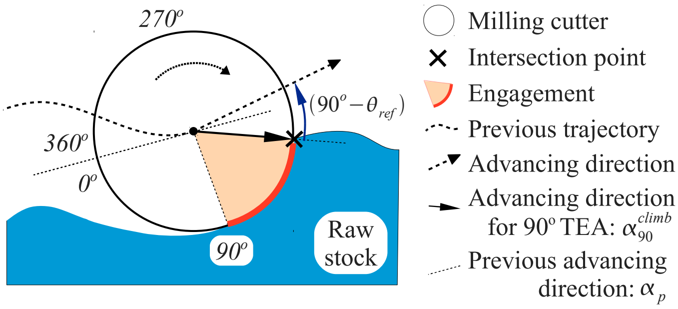


Fig. 6. Advancing direction for constant TEA, in climb milling

In the discrete domain, the intersection point can be obtained considering all the pixels from the circumference which are engaged into the material (i.e. white pixels). Each pixel is expressed in polar coordinates (ρ, ϕ) with respect to the tool center; the angular origin is the opposite of the current advancing direction (7). All pixels will have $0 \le \phi \le 360°$, considering a counter-clockwise direction. Ideally, no pixels with $0 \le \phi \le 90°$ or $270 \le \phi \le 360°$ should be engaged into material, since the tool already milled the respective location.

For *climb milling*, the *white* circumference pixel with the *largest* ϕ value will indicate the intersection between the tool circumference and the raw material edge.

Let α_p be the direction of the toolpath at the previous step. At the first step, α_p should point from the tool center towards the raw stock. Also, let (x_k^O, y_k^O) be the coordinates of pixels from the tool circumference, with respect to the tool center. Then:

$$\phi_k = \left(\operatorname{atan2}(y_k^O, x_k^O) + \alpha_p + 180°\right) \bmod 360° \tag{7}$$

Algorithm 1: Constant engagement milling

Input:
- Discrete tool diameter: D_D
- Reference engagement angle: θ_{ref}
- Raw stock: $\mathcal{R}$
- Part shape: $\mathcal{P}$
- Entry point: (x_0, y_0)
- Advancing step: Δd

Output:
- Trajectory points: $(x_1, y_1) \ldots (x_n, y_n)$
- Raw stock after the generated milling step: $\mathcal{R}$
- Reason why the algorithm finished: *stop_reason*

Initialization:
$(x, y) = (x_0, y_0)$
$i = 0$
repeat
  $i{+}{+}$
  if *engaged (white) pixels on tool circumference*
    $\alpha =$ direction for constant TEA (eq. 7-11)
    $(x, y) = (x, y) + \Delta d \cdot (\cos\alpha, \sin\alpha)$
    if *tool at* (x, y) *does not touch* $\mathcal{P}$
      $(x_i, y_i) = (x, y)$
      Update $\mathcal{R}$
    else
      stop_reason = "touched part"
  else
    stop_reason = "no material around the tool"
until *stop_reason*;

For *climb milling*, the advancing direction α^{climb} is (Fig. 6):

$$\alpha_{90}^{climb} = \max(\phi_k^{white}) - \alpha_p - 180° \tag{8}$$
$$\alpha^{climb} = \alpha_{90}^{climb} + 90° - \theta_{ref} \tag{9}$$

For *conventional milling*, the *smallest* ϕ will be used to compute the advancing direction α^{conv}. Therefore:

$$\alpha_{90}^{conv} = \min(\phi_k^{white}) - \alpha_p - 180° \tag{10}$$
$$\alpha_{conv} = \alpha_{90}^{conv} - 90° + \theta_{ref} \tag{11}$$

The algorithm can finish its loop in two ways. The stop reason will indicate the next state of the toolpath generator (Fig. 5):

- The tool touched the design part (i.e. it tried to remove white pixels from the part image)
- There are no more white pixels on the tool circumference.

In the second case, the computation for the advancing direction can be repeated after artificially increasing the tool radius, up to a preset value [3]. The approach helps when the shape of the raw part exhibits external sharp corners.

4.4 Contouring

There are some more details regarding the contouring step. In this mode, the cutter advances on the offset path, which is parallel to the contour of the design part. When the engagement exceeds the reference value, the tool should stop moving on the offset path with point data storage and should try to further maintain a constant engagement.

The first problem is how to decide when the engagement exceeds the reference. The estimation of engagement based on

[3] for example, 5 pixels or 10% of the nominal tool radius

pixel simulation is imprecise, having errors due to discretization. Furthermore, even with a perfect function for evaluating the engagement, when the part has many small details, the toolpath generator will alternate frequently between the contouring and constant engagement modes very, generating many tool returns, which would increase the machining time. Therefore, a compromise is made: in contouring mode, the tool engagement angle is allowed to exceed the reference value, but not more than a preset threshold. Good results were obtained with a 20° threshold, also called *maximum allowed overshoot*.

For avoiding the situations when the TEA computation routine would underestimate the result, an extra test was made. If the direction indicated by (9) differs from the contour direction with more than the allowed overshoot, the generator state is switched to constant engagement.

These two conditions guarantee that the prescribed engagement will never be exceeded by more than the allowed overshoot.

4.5 Searching for a starting point

The milling operation can begin in three modes:

- Plunging the cutter into the raw stock
- Entering into the stock horizontally, from lateral
- Continuing a contouring operation, from the point where the generator switched from state 3 to state 2.

As plunges should be avoided, the algorithm will search the above list from backwards. The first step is to look for a point on the contour where the toolpath generator switched from *contouring* to *constant engagement*. In these points, the tool is tangent to the raw material, so no plunging is needed.

If no such point is found, the algorithm tries to find another point, with the tool being tangent to the raw stock. At the same time, the tool should not be tangent to the design part, because in these points, the material has already been removed and the part already has the correct shape.

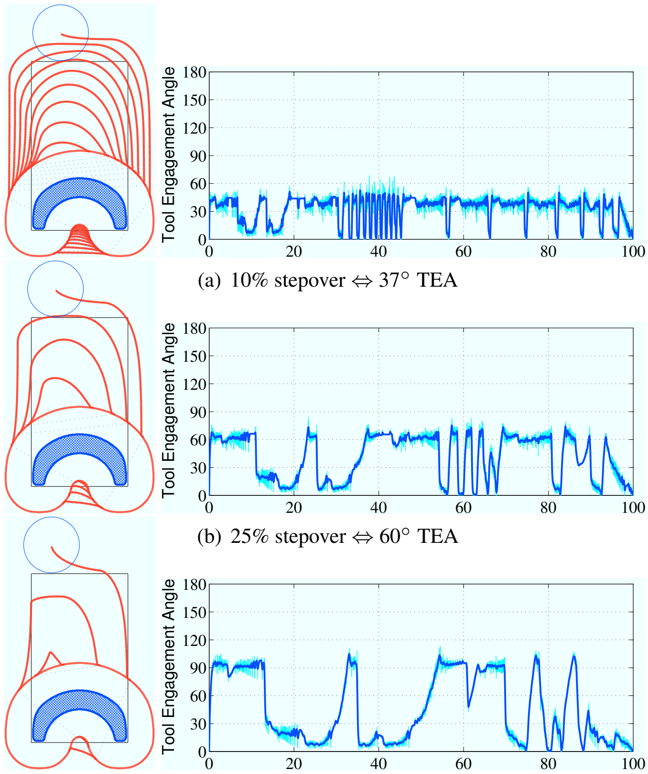
This point is found by offseting the raw and design part images and taking the contour pixels corresponding only to the raw image. For robustness, the part image can be offseted 1 pixel more than the raw image. There are many solutions, and only one is randomly selected. Experiments did not show significant influence of the starting point on the overall performance.

If no tangent point is found, it is because the algorithm has to do a *pocketing* operation, where plunging cannot be avoided. A starting point can be found by computing the distance transform Fisher et al. (2003) of the part image and taking the highest value. If the value is larger than the discrete tool radius, spiral plunging is possible.

5. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, it was tested on the two parts from Fig. 2, with various values for the prescribed engagement angle.

In Fig. 7(a)–7(c), the algorithm was tested on the part with simple geometry, with prescribed engagement values of 37°, 60° and 90°. These values correspond to stepover values of 10%, 25% and 50%.



(a) 10% stepover ⇔ 37° TEA

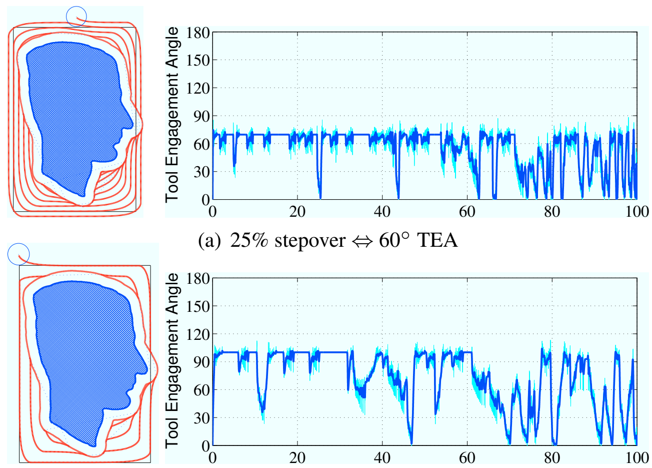
(b) 25% stepover ⇔ 60° TEA

(c) 50% stepover ⇔ 90° TEA

Fig. 7. Milling toolpaths generated with the proposed algorithm for the simple part

In Fig. 8(a)–8(b), the same algorithm was tested on the part with complex geometry, with prescribed engagement values of 60° and 90°, corresponding to stepovers of 25% and 50%.

The reader may observe that the generated toolpaths use an engagement angle close to the prescribed value, not exceeding it with more than 20°. At the beginning of the milling operation, the tool engagement angle has less variations, since most of the time is spent in constant engagement milling rather than contouring. As more material is removed, more time is spent in contouring, where tool engagement is lower than the prescribed reference and cannot be controlled.



(a) 25% stepover ⇔ 60° TEA

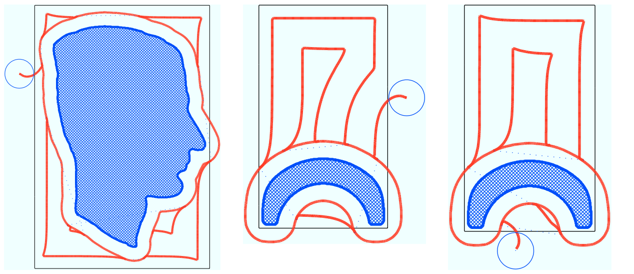
(b) 50% stepover ⇔ 90° TEA

Fig. 8. Toolpaths generated with the proposed algorithm for the part with complex geometry

For reducing the machining time, an adaptive feed rate (F) optimization scheme (Borangiu and Ivanescu, 2000) can be used, allowing higher feed rates for regions with small engagement values. The feed rate variation should be smoothed, since rapid variations will require high acceleration and jerk values.

5.1 Toolpath smoothing and simplification

While most authors try to generate a smooth toolpath, the algorithm proposed in this paper does not use smoothing. External sharp corners in the toolpath are allowed (Fig. 9), since they do not lead to engagement angle increase.



(a) 145° engagement (b) 135° engagement (c) 145° engagement

Fig. 9. Exterior sharp corners are allowed in the toolpath

The toolpaths generated with the proposed algorithm are described by a sequence of very small linear segments. On older-generation NC machine controllers, which have limitations regarding controller memory and/or processing speed, these toolpaths have to be simplified. The Douglas-Peucker algorithm gives very good results and remains the most widely used algorithm for curve simplification (Heckbert and Garland, 1997).

Modern NC controllers also have the ability to smooth the toolpaths described by small linear segments. On the CNC used for these experiments, this function is enabled with **G64**. Industrial robots also have this capability, which is usually called *procedural motion*.

CONCLUSION

The paper proposed a 2D roughing strategy with automatic tool engagement control, which has the following properties:

- The tool engagement is controlled using a prescribed reference value and a maximum overshoot (default 20°)
- Consists of small linear segments
- Is driven by the part and raw stock geometry
- Is suitable for arbitrarily complex part and stock geometry

The main benefits of the proposed algorithm for roughing tool path generation are: permanent cutting force control, tool life increase and reducing machining time.

Best results are obtained in conjunction with the approach described by (Uddin et al., 2006), which also ensures constant tool engagement at the finishing stage. More precisely, the proposed roughing algorithm should be applied on the precut semifinishing surface obtained with the backward algorithm for the finishing tool.

Therefore, modern CAM software which employ tool engagement control should define a combined roughing / finishing operation, instead of two independent operations, because roughing toolpaths depend on the finishing technological parameters.

The discussed toolpath generation algorithm can be combined with the method for reproduction of complex 3D surfaces described by depth map images obtained from laser scanning (Borangiu et al., 2007).

ACKNOWLEDGEMENTS

This work is funded by the National Council for Scientific University Research, in the framework of the National Plan for Research, Development and Innovation, grant 69/2007.

The 3D mesh model of Caesar's head was downloaded from `www.archive3d.net`.

REFERENCES

Bieterman, M. (2001). Curvilinear tool paths for pocket machining. University of Minnesota IMA.

Borangiu, T., Dogar, A., and Dumitrache, A. (2007). Automatic generation of 3d machining surfaces with tool compensation from grey level image models. In *Preprints of IFAC Wks. Intelligent Manufacturing Systems - IMS'07*. Alicante, Spain.

Borangiu, T. and Ivanescu, N. (2000). Intelligent control station for a robot-tended adaptive machine tool. In *Preprints of IFAC Sym. Robot Control SYROCO 2000*, 569 – 574. Vienna.

Coleman, G. (2006). How to exceed the limits of your toolpaths. *MoldMaking Technology Online*. URL `www.moldmakingtechnology.com/articles/100503.html`.

Fisher, R., Perkins, S., Walker, A., and Wolfart, E. (2003). Hypermedia image processing reference. URL `http://homepages.inf.ed.ac.uk/rbf/HIPR2/index.htm`.

Gasparraj, E. (2005). Constant material removal: The key to hard milling. *Modern Machine Shop*.

Heckbert, P.S. and Garland, M. (1997). Survey of polygonal surface simplification algorithms. Technical Report, Carnegie Mellon University.

Ibaraki, S., Ikeda, D., Yamaji, I., Matsubara, A., Kakino, Y., and Nishida, S. (2004). Constant engagement tool path generation for two-dimensional end milling. In *JUSFA'04: Proc. of 2004 Japan-USA Symposium on Flexible Automation*. Denver, CO, USA.

Kim, H.C., Lee, S.G., and Yang, M.Y. (2006). An optimized contour parallel tool path for 2D milling with flat endmill. *The International Journal of Advanced Manufacturing Technology*, 31(5-6), 567–573.

Pasko, R., Przybylski, L., and Slodki, B. (2002). High speed machining (HSM) – The effective way of modern cutting. *Proceedings of 7th DAAAM International Workshop CA Systems And Technologies*, 72–79.

Sprut Technology (2009). Sprutcam v7 documentation.

Stori, J.A. and Wright, P.K. (2000). Constant engagement tool path generation for convex geometries. *Journal of Manufacturing Systems*, 19(3), 172–184.

Uddin, M.S., Ibaraki, S., Matsubara, A., Nishida, S., and Kakino, Y. (2006). Constant engagement tool path generation to enhance machining accuracy in end milling. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 49(1), 43–49.

Wang, H., Jang, P., and Stori, J.A. (2005). A metric-based approach to 2D tool path optimization for high speed machining. *Journal of Manufacturing Science and Engineering (JMSE)*, 127(1), 33–48.