JCAD759

ELSEVIER

COMPUTER-AIDED DESIGN

Computer-Aided Design 00 (2002) 000–000

www.elsevier.com/locate/cad

Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation

B. Lauwers*, P. Dejonghe, J.P. Kruth

K.U. Leuven—Division PMA, Celestijnenlaan 300B, B-3001 Leuven, Belgium

Received 14 June 2001; revised 25 October 2001; accepted 26 November 2001

Abstract

The generation of collision free NC-programs for multi-axis milling operations is a critical task, which leads to multi-axis milling machines being exploited below their full capacities. Today, CAM systems, generating the tool path, do not take the multi-axis machine movements into account. They generate a multi-axis tool path, described by a sequence of tool postures (tool tip + tool orientation), which is then converted by a NC-postprocessor to a machine specific NC-program. As the postprocessing is normally done in batch mode, the NC-programmer does not know how the machine will move and the chance for having collisions between (moving) machine components is often very high. The execution of a machine test run or the application of a machine simulation system (NC-simulation) is the only solution to inform the NC-programmer about possible machine collisions during operation.

This paper describes a multi-axis tool path generation algorithm where the tool orientation is optimised to avoid machine collisions and at the same time to maximise the material removal rate along the tool track. To perform efficient collision avoidance, the tool path generation module (traditional CAM), the postprocessing (axes transformation) and machine simulation has been integrated into one system. Cutting tests have been carried out to define the allowable tool orientation changes for optimisation and collision avoidance without disturbing the surface quality.

The developed multi-axis tool path generation algorithm is applicable for the machining of several part surfaces within one operation. This, together with tool path generation functionality to adapt the tool orientation for both, maximal material removal and avoidance of collisions between (moving) machine components, are the innovative aspects of the presented research work.

Keywords: Computer aided manufacturing; Machine simulation; 5-Axis milling; Tool path optimisation

1. Introduction

The off-line generation of NC-programs for multi-axis milling operations mostly proceeds in two sequential steps (Fig. 1). In the first step, a CAM module (tool path generation) calculates the trajectory of the milling cutter. Each tool posture of the trajectory is described by its tool tip (x, y, z) and tool orientation (i, j, k), both expressed in a workpiece co-ordinate system.

In the second step, the tool path, output as a machine controller independent Cutter Location DATA file (CLDATA), is converted by a NC-postprocessor to a machine specific NC-program. An important task of a multi-axis postprocessor is the transformation from CLDATA to machine co-ordinates. Often, the axes transformation is not unique because multi-axis milling machines can have two possible configurations for a given CLDATA tool posture. Therefore, the postprocessor must select an appropriate configuration.

After postprocessing, the NC-program is not guaranteed to be free of collision. Collisions can occur between part and machine, tool and part or between moving machine components. Fig. 2 shows the high risk for collision between the machine head and the clamping table, during the 5-axis machining of a larger propeller blade.

Collisions between workpiece and machine components (e.g. between machine head and part) can only be checked by executing some test runs on the machine or by using a NC-simulation program. The output of a NC-simulation system is a report with a list of NC-statements where collision has occurred. The NC-programmer itself must solve the collision problems by changing milling strategy and regeneration of the tool path. Often, a number of iterations (tool path generation, NC-postprocessing, NC-simulation or test runs) are necessary until the complete NC-program is 100% free of collision.

* Corresponding author. Tel.: +32-16-322-480; fax: +32-16-322-987.
E-mail address: bert.lauwers@mech.kuleuven.ac.be (B. Lauwers).

0010-4485/02/$ - see front matter
PII: S0010-4485(02)00045-3

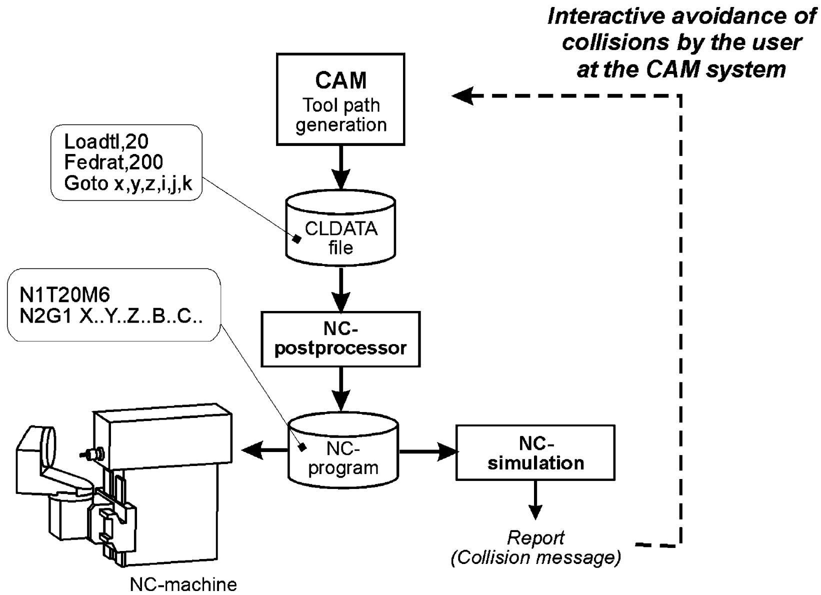


Fig. 1. Generation of NC-programs (standard method).

2. Related research and development: state-of-the-art

In spite of the technological benefits of multi-axis machining [1–3], the use of it remains limited due to the complexity and difficulty in the generation of collision free NC-programs. Commercial CAM systems mostly have limited functionality for the generation of multi-axis tool paths. The user is often forced to apply a constant tool orientation, set mostly, based on the highest local curvature of the part surface to be machined. For all other areas on the surface, this orientation will result in sub-optimal surface quality. More advanced CAM systems have functionality to linearly interpolate the tool orientation between a number of pre-defined points, each with a user defined tool orientation.

Some research and development is going on in the domain of tool orientation optimisation for multi-axis machining of complex shaped surfaces. So far, most of the published works present methods to optimise the tool orientation to maximise the machining strip width (~maximal material removal, minimal scallop, etc.), and to avoid collisions between tool and part (e.g. gouging, collision between tool holder and part, etc.).

Different kinds of tool orientation optimisation algorithms have been developed for maximal material removal. Pure analytical methods described in Refs. [2,4,5] optimise the tool orientation based on the curvature information in the cutter/contact point. These methods do not take the surface anomalies in the neighbourhood for the cutter contact point into account. Especially, when milling detailed surfaces with large cutters, details on the surface can be cut away unintentionally. Other tool optimisation methods [2,6,7] fit the tool as close as possible to the part surface. These optimisation techniques, often called 'cutting shape fitting' algorithms, use the entire surface definition in order to avoid gouging. In contradiction to the pure analytical methods, most of these algorithms determine the optimal tool posture iteratively.

Recent developments [8,9] show that by using several

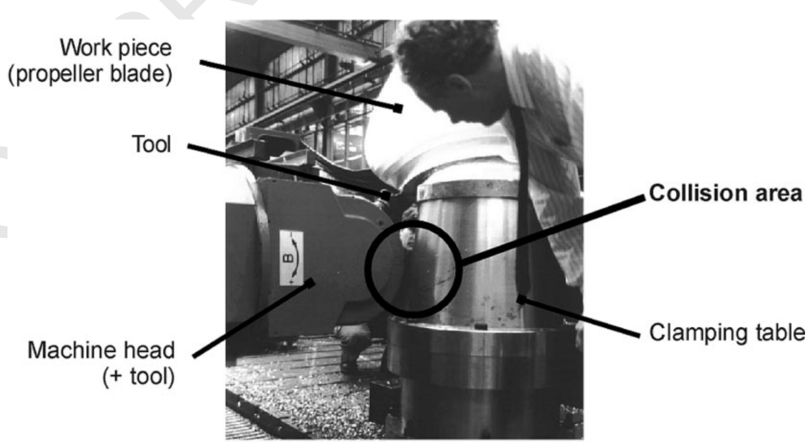


Fig. 2. High risk for collision between machine head and clamping table [John Crane-Lips].

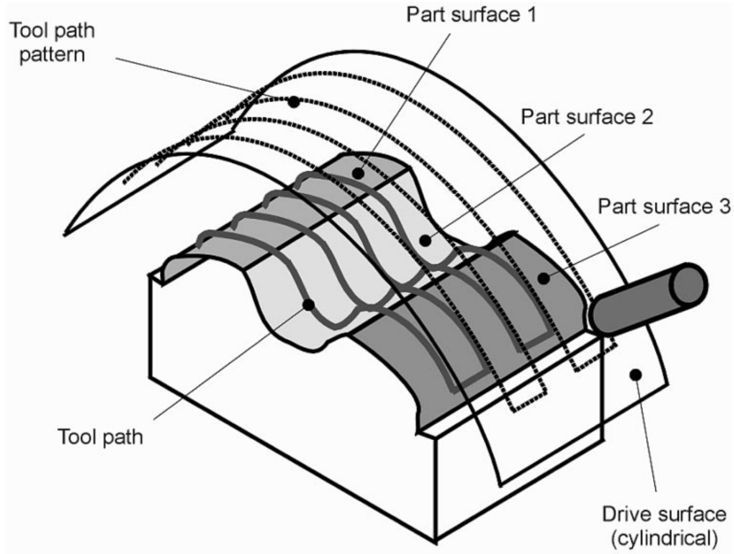


Fig. 3. Part/drive surface mechanism for tool path generation.

contact points, a better geometric match between the tool and the surface can be realised. Tool orientation optimisation is not the only parameter to optimise material removal. The ‘principal-axis method’, presented in Ref. [7], optimises the tool orientation and aligns the machining direction along the principal curvature directions of the surface to be machined. It is known that a higher material removal can be obtained in the direction of smallest absolute curvature. Further, an integration of tool positioning and tool path planning (the way the tool moves over the workpiece) is presented in Ref. [10].

Collision avoidance by changing the tool orientation has been studied by different researchers. Methods using a *C*-space, adapted from robot motion planning, are described in Refs. [11,12]. The underlying idea of *C*-space is to represent the tool orientation in an appropriate space in which the obstacles are mapped. A practical approach of the *C*-space is described in Ref. [13]. In this case, it has been used for the roughing of small impellers, where the possible colliding elements are tool, tool holder and part. Another approach to collision avoidance can be found in Ref. [14]. Before tool path generation, the part surface is dichotomised in two different sets of regions: regions accessible without colliding the check surfaces and regions causing collisions with the check surfaces. A point is accessible when the part can be reached in the direction of the part surface normal. This supposition is the main drawback of the method because it cannot be combined with, for example, tool orientation optimisation. The latter always introduces small changes in the tool orientation, making the dichotomy for collision avoidance incorrect. Other authors also focussed on this accessibility analysis and are generating Product Visibility Cones (PVCs) in a number of points on the workpiece [15,16]. A global tool interference checking for 5-axis machining is presented in Ref. [17]. The tool position is checked for possible interference with the convex hull of the check surface. If interference with the convex hull is detected, the tool interference is calculated and the tool orientation is corrected if needed. The advantage of this method over other similar methods lies in the computational efficiency. The first conservative checking phase (with the convex hull) is extremely fast, while the second phase delivers a solution to the collision problem.

All described approaches to collision avoidance have a common drawback: possible collisions between machine and part, machine and tool or between moving machine components are not taken into account. It may be clear from Fig. 2 that there is a real need to take the machine information (machine kinematics) into account during tool path generation. If only collisions are detected between part, tool and tool holder, then the example in Fig. 2 would not pose any problem.

Research and development carried out by the authors on tool orientation optimisation for maximal material removal [18] and on collision avoidance in 5-axis machining [19], has lead to the development of a new multi-axis tool path generation algorithm to adapt the tool orientation to avoid machine collisions (e.g. collisions between machine and part, machine and tool, or between moving machine components) and to maximise the material removal. The developed tool path generation algorithm uses the ‘drive/part surface paradigm’ which allows the generation of a single tool path for different part surfaces (multi-patch) easy and logic. The use of a drive surface is quite new compared to many other research works on tool orientation optimisation, where developed algorithms are only implemented for one single parametric based part surface (e.g. NURBS).

Within the ‘drive/part surface paradigm’, a drive surface is used to define the global tool path pattern for the

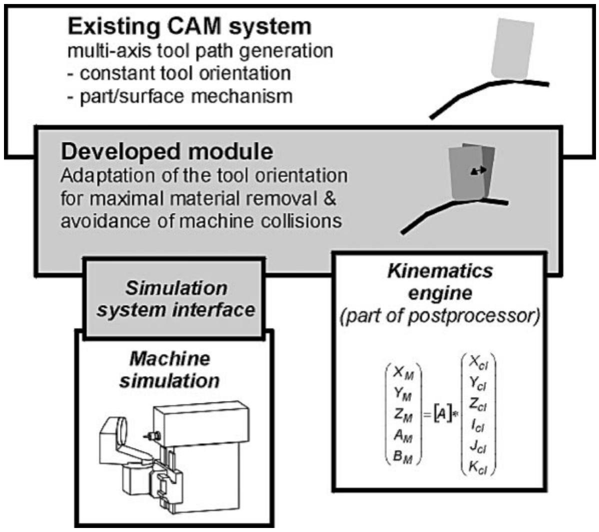


Fig. 4. General overview of the developed tool path generation system.

machining of all part surfaces (Fig. 3). To obtain the tool path, the tool geometry is projected from each drive point position parallel to the drive surface normal to the part surface. A good drive surface (also NURBS) for multi-axis tool path generation should have more or less the same shape as all the part surfaces to be machined. However in practice, the drive surface is often flat or cylindrical (as in Fig. 3) because the CAM-user has to model it interactively. Recent developments focus on the automatic construction of optimal drive surfaces [20]. In this case, the drive surface is modelled such that the iso-parametric curves (curves of constant u or v), used to define the tool path pattern (also cutting direction), are defined based on the curvature behaviour of the surfaces. This is based on the fact that higher material removal rates and increased surface quality can be obtained when the cutting direction is in the direction of absolute minimum curvature [2,7].

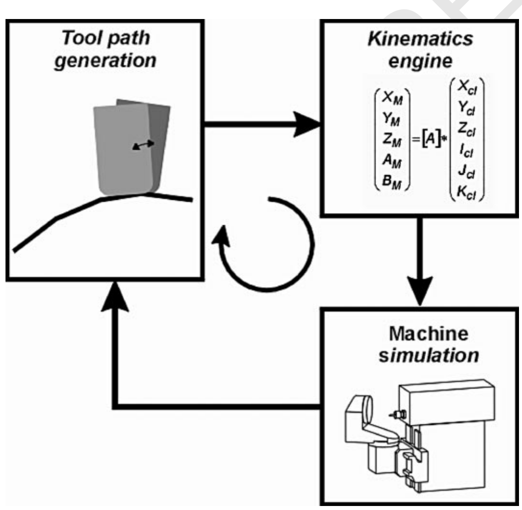


Fig. 5. Calculation of the tool path in a kind of control loop.

3. Developed concept: general overview

The developed modules, for the adaptation of the tool orientation for maximal material removal and avoidance of machine collisions, have been integrated on source code level within an existing CAM system (Fig. 4). This CAM system is capable to generate multi-axis tool paths starting from a tool path pattern defined within a drive surface. The multi-axis functionality of the existing CAM system is however limited to a fixed tool orientation and/or a linear interpolation of the tool orientation between a number of pre-defined points.

In order to perform avoidance of machine collisions, two additional modules have been integrated within the developed module for the adaptation of the tool orientation (Fig. 4). The kinematics engine, which is part of a traditional postprocessor, transforms tool posture co-ordinates to machine co-ordinates. The machine simulation system checks a given machine configuration for machine collision. The machine simulation system is based on a commercial system, but has been provided by the system developer as a shared library. In addition, a small set of simulation system dependent functions to control the simulation software has been provided. Examples of these functions are the initialisation of the simulation system, collision check of a given machine configuration, collision check while moving from one machine configuration to the next one, etc.

A simulation system interface has been developed in order to integrate machine simulation software's from different manufactures. This interface defines the relation between simulation system dependent functions and functions used by the developed module for the adaptation of the tool orientation.

Before tool path generation starts, the geometrical models of the machine, tools, part and fixtures, together with the kinematics model of the machine need to be downloaded into the simulation system. For the simulation system used, all geometrical models are input as STL-files, which are triangulated surface descriptions, initially used in the domain of rapid prototyping.

Once the tool path pattern is defined in the drive surface, the tool positioning is done within a control loop, comprising different modules: tool path generation, kinematics engine and machine simulation (Fig. 5). This concept is different from the classical concept shown in Fig. 1, where the tool path generation, postprocessing and NC-simulation are performed in a sequential way. After the generation of a single tool posture (x, y, z, i, j, k) with optimised tool orientation for maximal material removal, it is directly converted by the kinematics engine to machine axis co-ordinates and checked for collisions by the integrated machine simulation system.

After each collision check, the simulation system feeds back the type of collision (e.g. collision between tool and part, head and part, part and machine, etc.), the centre of the collision curve (this is the intersection curve calculated

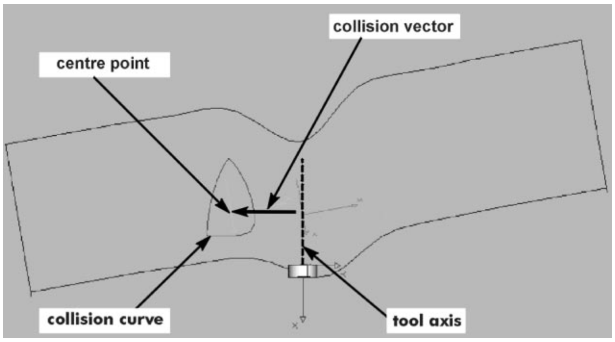


Fig. 6. Example of a collision curve and collision vector.

between two colliding elements), the length of the collision curve, and a vector (= collision vector) which is constructed from the centre point of the collision curve perpendicular to the tool axis (Fig. 6). The centre point (x_c, y_c, z_c) of the collision curve (with length L) is calculated by the following equation:

$$x_c = \frac{1}{L}\int_0^L x(s)\,\mathrm{d}s, \qquad y_c = \frac{1}{L}\int_0^L y(s)\,\mathrm{d}s, \qquad z_c = \frac{1}{L}\int_0^L z(s)\,\mathrm{d}s \tag{1}$$

If collision happens, the collision information (e.g. length of collision curve, collision vector, etc.) is used to find a new tool orientation (see details later). The new tool posture is then checked again for collision. Once a collision free tool posture is found, which may require several iterations, the tool path generation algorithm continues with the calculation of the tool posture in the next drive point. In case, no collision free tool orientation is found (e.g. after five iterations), the algorithm performs a tool retract to a safe level. The tool will engage again for one of the next tool postures that is collision free.

As the tool orientation optimisation is independent of the definition of the tool path pattern, the material removal is only maximised along the tool tracks. Further optimisation of the material removal for the whole workpiece can be obtained by choosing a proper drive surface and tool path pattern of which the feed direction is parallel to the direction of minimum curvature [20].

4. Tool orientation: inclination and screw angle

In many CAM systems, the tool orientation is defined by a lead angle and tilt angle. The lead angle is defined in a plane parallel to the feed direction, while the tilt angle is defined perpendicular to it. Within this research work, a strategy has been developed that adapts the lead angle for material removal optimisation and the lead as well as the tilt angle for collision avoidance. More details why such a strategy is chosen is described in Section 6.

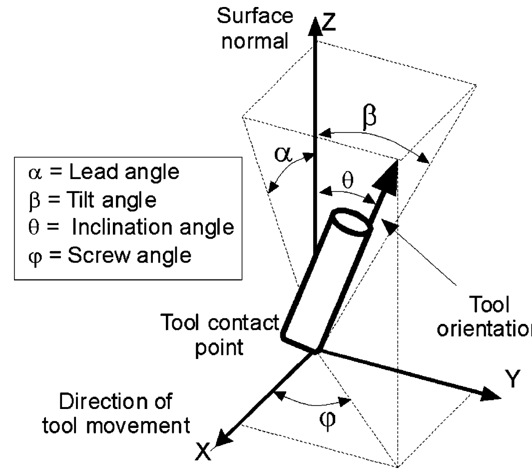


Fig. 7. Inclination and screw angle.

Changing the tilt angle during collision avoidance has however the drawback that the tool contact point is moved, giving an irregular movement of the tool along the tool track. This is not favourable for many users and therefore, the inclination and screw angle has been introduced within this research work to represent the tool orientation (Fig. 7).

Changing the screw angle can now be seen as a rotation of the cutter around the surface normal, keeping the tool contact point at the same place. The lead angle and the inclination angle have the same value for a screw angle equal to zero. One can easily go from one definition to the other using Eq. (2)

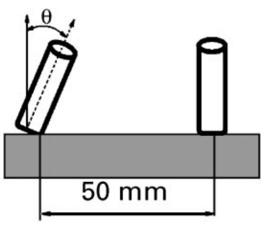
$$\cos(\varphi)\sin(\theta) = \sin(\alpha), \qquad \sin(\varphi)\sin(\theta) = -\cos(\alpha)\sin(\beta), \; \cos(\theta) = \cos(\alpha)\cos(\beta) \tag{2}$$

5. Calculation of tool orientation ranges

As mentioned earlier, the module for the adaptation of the tool orientation has been integrated within an existing CAM system. The generation of a complete tool path starts with the construction of a drive surface. Within the drive surface, a tool path pattern is defined and sampled into a number of drive points (DPs). The drive surface can cover more than one part surface, which makes the developed system industrially applicable and easy to be implemented within commercial CAM systems. In general, the tool path itself is then obtained by projecting the tool geometry from each drive point position (DP_i) onto the part surface.

During tool path generation, the change of tool orientation has to be kept under control. The influence of the change of the inclination angle per unit distance ($\mathrm{d}\theta/\mathrm{d}s$) has been investigated (Table 1). A flat surface was machined and the inclination angle was changed from an initial value down to zero over a distance of 50 mm. The workpiece

Table 1
: Error (deviation from flat) as a function of the angle change per unit distance ($d\theta/ds$)



Initial inclination, θ (°)	$d\theta/ds$ (°/mm)	Error (μm)
3	0.06	~ 10
5	0.1	~ 13
8	0.16	~ 18
10	0.2	~ 20

material was Avionol T4 and the following cutting conditions were used: tool, $\varnothing = 20$ mm, corner radius = 3.5 mm, cutting speed = 350 m/min, feed = 0.2 mm/tooth.

Table 1 shows the increase of error with increasing speed of angle change. The machine itself can influence these results, but the same tendency will remain. On the basis of these facts, two rules, related to maximum angle changes, have been defined within this research work. First, the user can set absolute and minimum values for the both angles ($\theta_{min}/\theta_{max}$); ($\varphi_{min}/\varphi_{max}$). Examples of these values are $\theta_{min} = 0°$, $\theta_{max} = 45°$, $\varphi_{min} = -30°$, $\varphi_{max} = 30°$). A minimum inclination angle of 0° is related to the fact that negative values most often give gouging problems. Next, the change of angle per unit distance ($d\theta/ds$) can be limited. This means that drastic changes of the tool orientation are not allowed. As a result of this, the maximum and minimum values for the inclination and the screw angle, to be used for the adaptation of the tool orientation, are position dependent and are further indicated as ($\theta_{i,min}$; $\theta_{i,max}$) and ($\varphi_{i,min}$; $\varphi_{i,max}$). The index (i) refers to the ith drive point position.

Fig. 8 gives a schematic overview of the evolution of the ranges for the inclination angle during tool path generation (analogue figures can be made for the screw angle). The absolute maximum and minimum values are displayed by dotted lines (e.g. $\theta_{min} = 0°$, $\theta_{max} = 45°$,). The drive point positions are indicated as DP_i ($i = 1, 2, 3, 4, \ldots$). The tool positions with optimised tool orientation are indicated as TP_i^*, while those corrected for collision are indicated as TP_i. As an example, the collision area is drafted as a grey shaded region. This means that all TPs, having their inclination angle lying in the grey shaded region, result in a collision between, for example, part and machine.

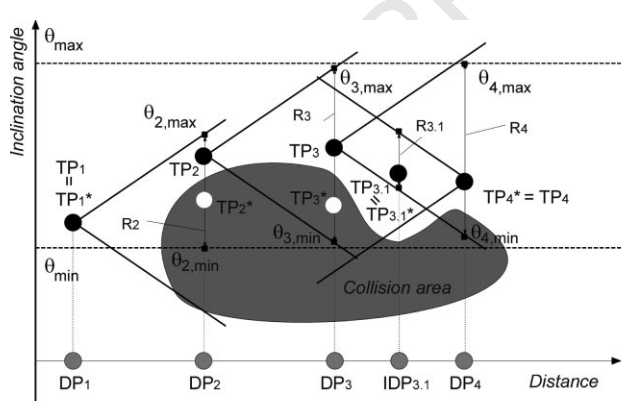


Fig. 8. Calculation of the allowable ranges for tool orientation optimisation and collision avoidance (inclination angle).

Starting from the first drive point (DP_1), the corresponding tool position with optimised inclination angle is calculated (TP_1^*). After a collision check, TP_1^* seems to be collision free. So the final tool posture will be TP_1 equals TP_1^*. In order to find the ranges for optimising and correcting tool path position 2, two lines, with a slope defined by the maximum angle change per unit distance, are constructed in point (TP_1). The valid range R_2 can easily be drawn. Notice that $\theta_{2,min}$ is equal to θ_{min}. Tool path position optimised for tool orientation (TP_2^*), is within the collision area and will be corrected by the collision avoidance module to TP_2. Similar to the points 1 and 2, the tool path positions TP_3 and TP_4 are defined. TP_3^* has to be corrected for collision, while TP_4^* is collision free ($TP_4 = TP_4*$). The ranges R_3 and R_4 are constructed similar to R_2. Note that R_4 is bounded by the maximum value for the inclination angle ($\theta_{4,max}$ is equal to θ_{max}).

Typical to the used CAM system is the automatic insertion of intermediate drive positions (IDP_i) if the tool path movement between two positions is outside the tolerance band (intol–outtol). Due to this fact, the generation of tool path positions not always proceeds in a sequential order (1,2,3,4,…). This is shown in Fig. 8 by the intermediate drive point position $IDP_{3.1}$. The range for valid inclination angles ($R_{3.1}$) has to be defined by constructing lines in the two neighbouring points TP_3 and TP_4. If a smaller inclination angle is possible (only looking to this particular point), it has to be corrected to $TP_{3.1}^*$ (and $TP_{3.1}$) in order to fulfil the requirements of smooth angle change.

Additional rules have been implemented and can be optionally set by the user for speeding up the tool path generation and to obtain an even smoother tool path (if it is a major concern). These rules are especially related to the intermediate drive point positions. For example, if the original drive positions are close to each other, optimisation for the intermediate drive points does not always lead to further improvement of the tool path. Two implemented rules, applied for intermediate drive points, are:

- For intermediate tool postures, the tool posture is not optimised for maximal material removal and the first checked tool posture for collision avoidance is the one obtained by a linear interpolation between the previous and the next tool posture. If the distance between tool postures is relative small, this will usually result in satisfactory tool paths. In Fig. 8, one can see that for the $IDP_{3.1}$, a linear interpolated tool posture would directly give a good result.

- If two neighbouring tool postures have collision and the distance between them is close, then there is little chance to find a collision free tool posture for the intermediate drive position. In this case, the algorithm assumes that this tool posture has collision and it will not be checked by the simulation system.

6. Adaptation of the tool orientation: developed algorithm

The developed module for the adaptation for the tool orientation makes use of some functions available within the existing CAM system. As an example, the function to project a tool from a given point onto the part surface has been used several times. According to the specification of the CAM system, this function guarantees a gouge free tool positioning.

The developed algorithm for the calculation of an optimal and collision free tool posture for a given drive point is done in different steps. For simplicity, the schematic pictures in Fig. 9, explaining the algorithm for tool orientation optimisation, use a flat plane as a drive surface.

Step 1. A given drive point DP_i is projected onto the part surface (along the drive surface normal), which will be used as the cutter contact point (CP_i). This is different to the standard CAM system (and also many other CAM systems), where the projection of a drive point gives the tool tip position. The index (i) refers to the ith drive point.

The result of this projection are the (u, v) co-ordinates of the contact point (CP_i) and a reference to the patch surface (because there can be different patches, each having their own u–v parameterisation). From this, it may be clear that multiple (eventual trimmed) part surfaces can be taken into account very easily, as far the drive surface covers all the part surfaces to be machined.

Step 2. At the projected drive point position (CP_i), a first estimation of the tool orientation for maximal material removal is calculated based on the principal curvatures, k_1 and k_2. In this step, only the inclination angle is taken into account for optimisation. The basic algorithm for the calculation of the optimal inclination angle is based on earlier research work and is fully described in Refs. [2,18]. The basic idea behind the algorithm is to incline the tool as such that the generated contact curve fits as close as possible to the local curvature of the surface. In a specific case, when the curvature k_1 is much larger then the curvature k_2, the inclination angle (θ) is mainly defined by k_1. If the cutting direction is also in the direction of the smallest curvature, an estimation of the inclination angle can be derived from Eq. (3). ϕ_{tool} and c_{r} are, respectively, the diameter and the corner radius of the end mill or a toroidal mill.

$$\sin(\theta) = \frac{\dfrac{\phi_{\text{tool}}}{2} - c_{\text{r}}}{\dfrac{1}{k_1} - c_{\text{r}}} \tag{3}$$

As a reminder, the inclination angle (θ) is defined as the angle between the tool axis and the normal on the part surface in the point CP_i. The screw angle is not optimised in this step, but is set to:

$$\begin{cases} 0 & \text{if } (\varphi_{i,\max} \le 0 \le \varphi_{i,\max}) \\ \varphi_{i,\min} & \text{if } (0 < \varphi_{i,\min} < \varphi_{i,\max}) \\ \varphi_{i,\max} & \text{if } (\varphi_{i,\min} < \varphi_{i,\max} < 0) \end{cases} \tag{4}$$

This means that during the optimisation of the material removal, the system tries to set the screw angle to zero. The screw angle can be different from zero and this is due to a collision avoidance action in a previous point.

Step 3. The estimation of the tool orientation (Step 2) is not guaranteed to be gouge-free because it is only using the curvature properties of the actual point on the part surface. Therefore, the tool orientation is further refined until a gouge-free tool posture is realised. To do this, the entire tool geometry (with estimated tool inclination) is projected from the drive surface on the part surface, using the standard CAM-systems' functionality for tool projection. The tool projection function of the existing CAM system is assumed to give a gouge free tool posture as it looks for the first contact point encountered during projection. The tool projection function returns a contact point (CCP_i) which might actually being located everywhere on the bottom of the cutter. Similar to the projection in Step 1, the contact point CCP_i is output by the existing CAMs tool projection algorithm as (u, v) co-ordinates together with a reference to the specific patch surface.

Step 4. The inclination angle is adjusted (a certain value is added for θ), based on the relationship between the desired contact point (CP_i) and the contact point returned by the tool projection method (CCP_i). The goal is to minimise the distance between CP_i and CCP_i. The tool geometry is again projected from the drive point position onto the part surface using the standard CAM-systems' functionality for tool projection. This projection gives a new CCP_i, which is normally closer to CP_i. If needed, this iteration process (calculation of a new θ) and projection is continued until CCP_i is close enough to CP_i.

If the obtained inclination angle is smaller than the minimum inclination angle ($\theta_{i,\min}$), it is set to $\theta_{i,\min}$. In case, the obtained inclination angle would be larger than $\theta_{i,\max}$, the $\theta_{i,\max}$ is set to the optimal value, because a reduction would

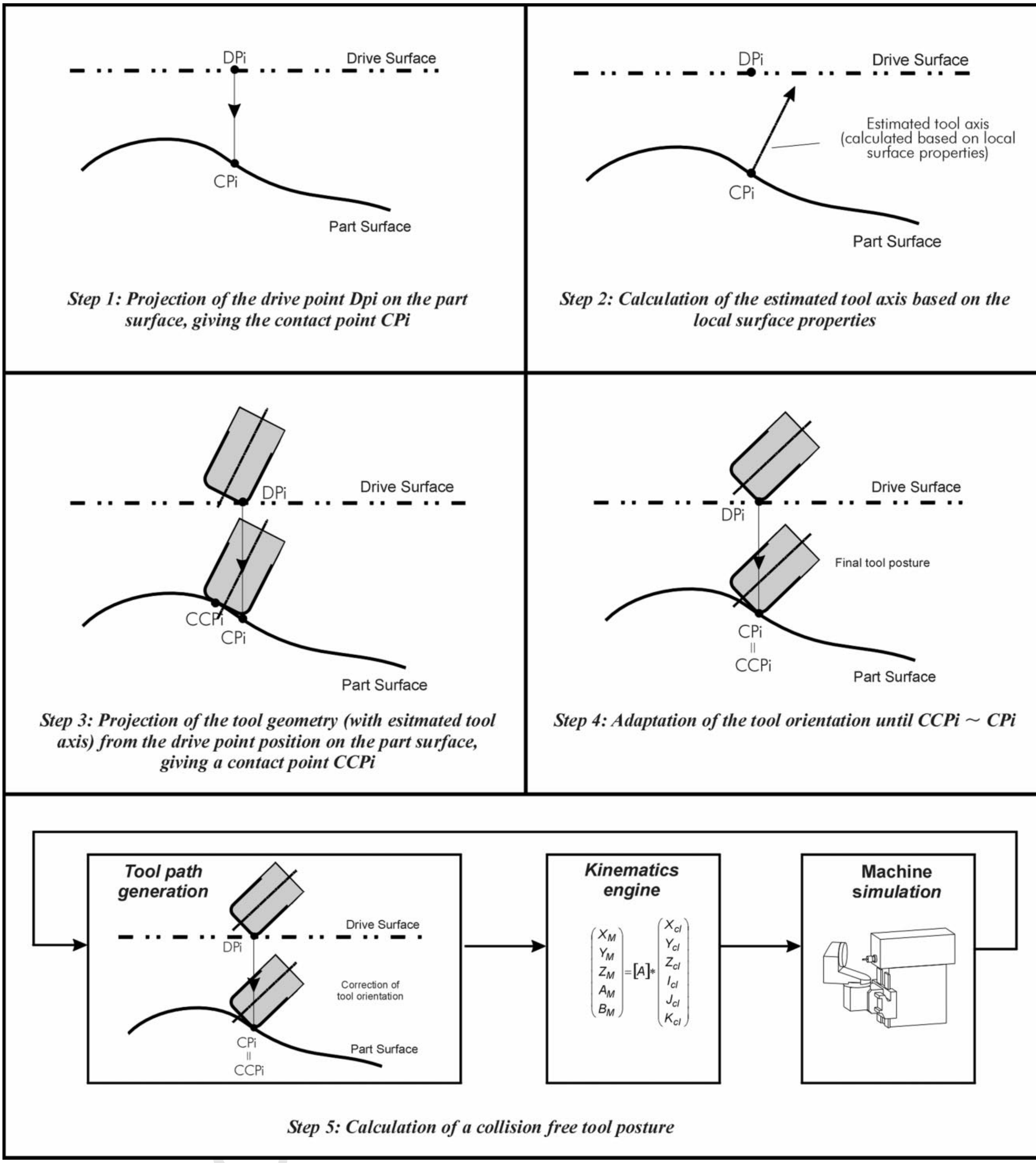


Fig. 9. Adaptation of the tool orientation for maximal material removal and avoidance of machine collisions.

give gouging. However, for realistic values of $\theta_{i,\max}$ (e.g. 45°), this situation would be exceptional.

Step 5. The tool posture $(x_{\mathrm{cl}}, y_{\mathrm{cl}}, z_{\mathrm{cl}}, i_{\mathrm{cl}}, j_{\mathrm{cl}}, k_{\mathrm{cl}})$ is converted to machine axes values $(X_{\mathrm{M}}, Y_{\mathrm{M}}, Z_{\mathrm{M}}, A_{\mathrm{M}}, B_{\mathrm{M}})$ by the kinematics engine and the movement from the previous to the current position is checked for collision by the machine simulation system. If the tool posture is collision free, it is kept as the final tool posture. If a collision occurs, a new tool orientation is found. Although tests proved that an increase of the inclination angle could solve most of the collisions, the increase of this angle should be limited, because it is in contradiction with the increase of material removal. Therefore, a collision avoidance strategy is proposed that acts on the inclination angle as well as on the screw angle. A little change of the screw angle helps to limit the increase of inclination angle. In this sense, it is

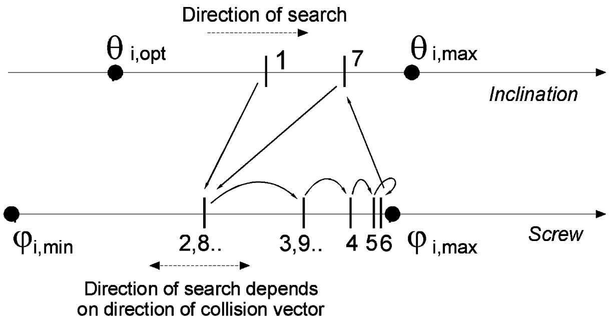


Fig. 10. Graphical representation of the aggressive algorithm for finding a collision free tool orientation.

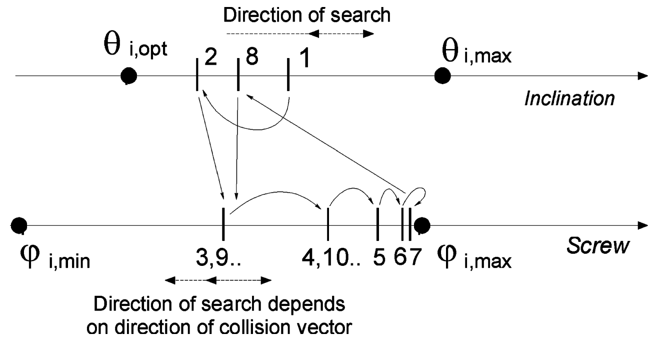


Fig. 11. Graphical representation of the smooth algorithm for finding a collision free tool orientation.

better to choose for a collision free tool posture ($\theta = 25°$, $\varphi = 10°$) instead of having ($\theta = 35°$, $\varphi = 0°$).

Two algorithms have been developed that searches for a collision free tool orientation within the allowable ranges (minimum and maximum values) for the inclination angle ($\theta_{i,\mathrm{opt}}$ and $\theta_{i,\max}$) and the screw angle ($\varphi_{i,\min}$ and $\varphi_{i,\max}$). Note that the minimum value for $\theta_{i,\min}$ is equal to $\theta_{i,\mathrm{opt}}$, which is the inclination angle obtained during the optimisation process for maximal material removal (Step 4).

The first algorithm tries to find a collision free tool position as fast as possible. Collision free tool positions are not always optimal in the sense that the change of inclination and screw angles may be larger then really needed. The second algorithm tries to find more optimal collision free tool positions. It searches for positions very close to collision (just not colliding). The latter algorithm gives a smoother tool path and is therefore called the smooth algorithm, while the first algorithm is called aggressive.

6.1. Aggressive algorithm for collision avoidance

The algorithm, schematically shown in Fig. 10, uses a kind of *Bisection Search Method* (new evaluated position is in the middle of previous bound) to find a collision free tool orientation within the given ranges for inclination and screw angle. The algorithm continuously switches between the ranges of the inclination angle and the screw angle. This means that after each check within the range for the inclination angle, a full search is done within the range for the screw angle. After a full search for the screw angle, the algorithm checks the next position for the inclination angle and if collision still occurs, the search within the range for the screw angle is repeated. The number of checks within each range can be set by the user. In the example of Fig. 10, the number of checks for the screw angle is set to 5.

The direction of search for the inclination angle is always positive (increasing angle). The direction of search for the screw angle is such that the tool moves in the direction of the collision vector. This collision vector is calculated after each collision check by the simulation system (see above). Further, after each check, the system evaluates the length of the collision curve (also a result of the collision check by the simulation system). A decreasing length means that the chosen direction is the right one. An increasing length will start a search in the opposite direction. The algorithm stops when a collision free tool position is found or when a maximum number of checks are exceeded.

It is clear that this collision avoidance strategy sets some requirements to the simulation system. The developed algorithm uses the length of the collision curve and the collision vector to control the iteration process to find a collision free tool posture. Some commercial machine simulation systems only feed back ‘collision’ or ‘no collision’, which makes it, of course, difficult to develop an efficient collision avoidance strategy.

6.2. Smooth algorithm for collision avoidance

The smooth algorithm tries to find a collision free tool path position closer to the collision area. The algorithm is similar to the previous one, but as soon a collision free tool position is found, the Bisection Search Method is continued in the opposite direction. An example is given in Fig. 11. Check number 1 is collision free and thus the algorithm proceeds testing in the other direction (position number 2). In this example, position 2 has collision and the algorithm proceeds by evaluating the change of the screw angle. If collision still occurs, the algorithm further checks position 8 (middle of positions 1 and 2).

The smooth algorithm requires more iterations steps. It is up to the user to choose between a fast aggressive or a more CPU requiring algorithm giving a smoother and more optimal tool path.

7. Examples and machining tests

The developed multi-axis tool path generation method has been experimentally verified during several tests on industrial workpieces. A first example is the machining of a complex shape ($100 \times 100\ \mathrm{mm}^2$) with convex as well as concave areas (Fig. 12). The aim of this test is to investigate the effect of the adaptation of the tool orientation for higher material removal. The shape has been machined in aluminium (AlMgSi1) on a 5-axis milling machine, MAHO

Fig. 12. Effect of the adaptation of the tool orientation for maximal material removal rate (smaller scallops).

600C. For the experiment, a toroidal cutter of diameter 20 mm with two inserts of 8 mm has been used. To compare results, the workpiece has also been cut with a normal strategy provided by the standard CAM system. In this case, the inclination angle was set to 9°, which is the minimal inclination angle necessary to avoid gouging in the concave area of the workpiece. For both tests, 20 tool path tracks have been spread over the surface. The difference between the two strategies (fixed tool orientation and optimised tool orientation) has been evaluated by measuring the scallop height. Surface roughness and machining time are comparable. To quantify the scallop height, the waviness profile has been measured on the convex area of the machined part (perpendicular on the feed direction). The waviness depth (W_t) has been taken as a representative value for the scallop height. The strong reduction of the scallop height in the convex area is due to the fact that the inclination angle could be reduced to almost zero. For the part, machined with the standard CAM system, this was not possible, because the inclination angle was fixed to 9°.

A next example is the generation of a tool path for the machining of propeller blade on a large 5-axis milling machine. A cylindrical drive surface has been constructed in order to generate one tool path for the different part surfaces representing the foot area. Fig. 13 shows the tool path generation for the foot of the propeller blade.

During tool path generation, collisions occurred between the machine head (5-axis machine with two rotary axes in the head) and the part. The collision avoidance strategy modifies the tool orientation (moving the machine head away from the part) by keeping the contact point on the same place. Therefore, the movement of the tool tip shows an irregular behaviour. Fig. 14 shows some pictures of the simulation system running during tool path generation.

The tool path has also been generated without collision avoidance. In this case, only 5% of the foot could be machined, compared to the tool path generated with collision avoidance where more than 80% has been machined. The un-machined regions are defined as rest material that is used as input for the next operation (eventually with a longer tool).

To compare the aggressive and smooth collision avoidance algorithm, a test has been done on a HP 180C Unix workstation. The results are given in Table 2.

If no advanced collision avoidance is applied (retract only), only 13% of the checked drive points turned out to

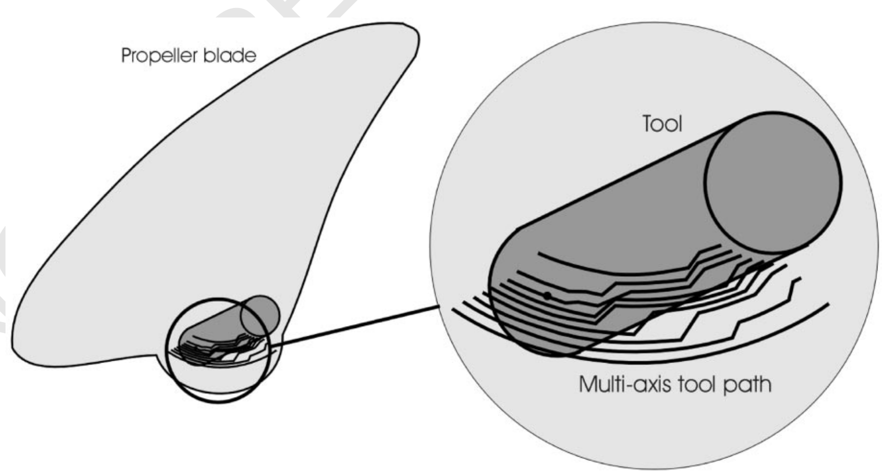


Fig. 13. Tool path generation for the machining of the foot of a propeller blade (The curves represent the tool tip. The irregular movement of the tool path is caused by a change of the tool orientation, keeping the cutter contact point on the same place.).

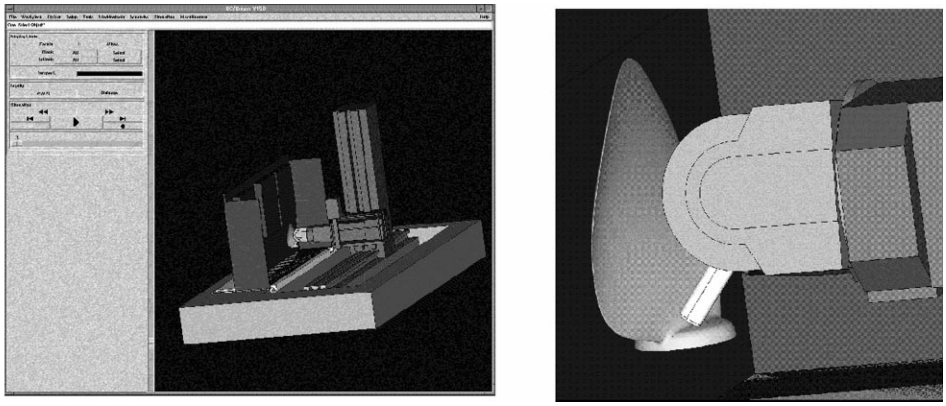


Fig. 14. Simulation (collision detection) during tool path generation.

Table 2
Performance comparisons between different collision avoidance algorithms

	Number of drive points	Number of collision free positions	Total number of collision checks	Avoidance efficiency (%)	Time (min)
Retract only	476	62	415	13	8.55
Aggressive avoidance	664	502	931	76	16.38
Smooth avoidance	582	528	1672	90	32.77

be collision free. The avoidance efficiency is the ratio of the number of collision free points to the total number of drive points (e.g. $62/476 = 0.13$). The aggressive avoidance algorithm results in 76% of the drive points having a collision free tool posture. The smooth algorithm is even performing better, with 91% of the drive points ending up collision-free. However, the smooth method is very time consuming. The increased time is due to the higher number of collision checks by the machine simulation system.

The number of drive points increases for both aggressive and smooth avoidance methods with respect to the retract only option. This increase is due to the number of supplementary intermediate drive points, which need to be generated to keep the tool path within the required intol–outol range.

Although, the indicated times (min) are high compared to other existing classical multi-axis tool path generation algorithms, they have to be evaluated in a global sense. Quite some time is needed to perform the collision checks by the simulation system. However, the indicated times are still a fraction of the time that would be needed to generate a collision free program following the concept of Fig. 1. It takes hours of interactive NC-programming work to generate the different tool paths for the foot area of the propeller blade with the commercially available NC-programming systems.

8. Conclusions

This paper described a multi-axis tool path generation algorithm where tool orientation is adapted to avoid machine collisions and to increase material removal. The avoidance of machine collisions has been realised by the thigh integration of postprocessing (axes transformation) and machine simulation within an existing CAM system. Multi-axis tool paths are generated starting from a tool path pattern defined in a drive surface, which makes it easy to machine different part surfaces in one operation.

Acknowledgements

This research work has been carried out in the frame of the Industrial and Materials Technologies research and technological programme of the European Community (reference OPTIMACH, BE96-3032).

References

[1] Altan T, et al. Advanced techniques for die and mold manufacturing. CIRP Ann 1993;42(2):707–16.
[2] Kruth JP, Klewais P. Optimization and dynamic adaptation of the cutter inclination during five-axis milling of sculptured surfaces. CIRP Ann 1994;43(1):443–8.
[3] Li SX, Jerard RB. 5-Axis machining of sculptured surfaces with flat-end cutter. Comput Aided Des 1994;26(3):165–78.
[4] Lee YS, Ji H. Surface interrogation and machining strip evaluation for 5-axis CNC die and mold machining. Int J Prod Res 1997;35(1):225–52.
[5] Jensen CG, Mullins SH, Anderson DC. Scallop elimination based on precise 5-axis tool placement, orientation and step-over calculations. ASME—Adv Des Automat 1993;65-2:535–44.

[6] Redonnet J-M, Rubio W, Monies F, Dessein G. Optimising tool positioning for end-mill machining of free form surfaces on 5-axis machines for both semi finishing and finishing. Int J Adv Manufact Technol 1998;16:383–91.

[7] Rao N, Bedi S, Buchal R. Implementation of the principal-axis method for machining of complex surfaces. Int J Adv Manufact Technol 1996;11:249–57.

[8] Klocke F, Altmüller S. Five-axis milling: geometrical and technological benefits for high efficiency processes. Prod Engng 1998; V(1):1–4.

[9] Warkentin A, Ismail F, Bedi S. Multi-point tool positioning strategy for 5-axis machining of sculptured surfaces. Comput Aided Geometr Des 2000;17:83–100.

[10] Rao N, Ismail F, Bedi S. Integrated tool positioning and tool path planning for five-axis machining of sculptured surfaces. Int J Prod Res 2000;38(12):2709–24.

[11] Morishige K, Kase K, Takeuchi Y. Collision-free tool path generation using 2-dimensional *C*-space for 5-axis control machining. Int J Adv Manufact Technol 1997;13(6):393–400.

[12] Choi BK, Jerard RB. Sculptured surface machining—theory and applications, ISBN 0-412-78020-8. Dordrecht, The Netherlands: Kluwer, 1998.

[13] Morishge K, Takeuchi Y. 5-Axis control rough cutting of an impeller with efficiency and accuracy. Proc 1997 IEEE Int Conf Robot Automat 1997:1241–6.

[14] Elber G, Cohen E. A unified approach to accessibility in 5-axis freeform milling environments. Proc SSM Conf (Machining Impossible Shapes), Detroit, USA 1998:376–86.

[15] ElMaraghy HA, Vafeesefa A. Accessibility analysis in 5-axis machining of sculptured surfaces. Proc 1998 IEEE Int Conf Robot Automat, Leuven 1998:2464–9.

[16] Kang J, Suh S. Machinability and set-up orientation for five-axis numerically controlled machining of free surfaces. Int J Adv Manufact Technol 1997;13:311–25.

[17] Lee YS, Choi BK. 2-Phase approach to global tool interference avoidance in 5-axis machining. Comput Aided Des 1995;27(1):715–29.

[18] Kruth JP, Lauwers B, Dejonghe P, Dotremont J. Optimized NC-tool path generation for 5-axis machining of complex surfaces, Machining impossible shapes, ISBN 0-412-84680-2. Dordrecht, The Netherlands: Kluwer, 1999 chapter 9, p. 343–50.

[19] Lauwers B, Kruth JP, Dejonghe P, Vreys R. Efficient NC-programming of multi-axis milling machines through the integration of tool path generation and NC-simulation. CIRP Ann 2000;49(1):367–70.

[20] Lauwers B, Kruth JP, Dejonghe P. An operation planning system for multi-axis milling of sculptured surfaces. Int J Adv Manufact Technol 2001;17(11):799–804.

Bert Lauwers born at Ostend, Belgium on December 23, 1963, obtained the degree of mechanical engineer—option manufacturing technology and production management–in 1987 from the K.U. Leuven (Leuven, Belgium). In 1993, he earned his PhD from the same university on the subject 'Computer Aided Process Planning and Manufacturing for Electrical Discharge Machining'. He is currently a professor in the same university and does research and education in the field of manufacturing processes (multi-axis milling, electrical discharge machining, stereolithography, selective laser sintering, etc.) and computer support in manufacturing such as CAD/CAM, NC-programming, NC-simulation and NC-postprocessing. He is active in many research projects, mostly funded by the EU and Belgian government. He is an active member of Belgian Institute for Normalisation (BIN), board member of Belgian Society of Mechanical and Environmental Engineering (BSMEE), member of KVIV, member of the VPT (Association for Production Technology, the Netherlands) and secretary of one of the VPT subgroups active in the field of chemical and physical machining.

Pieter Dejonghe obtained his degree of mechanical engineer–option manufacturing technology and production management–in 1995 from the K.U. Leuven (Leuven, Belgium). From 1995 till 2000, he performed research at the same university on the multi-axis machining of complex shaped surfaces. In 2001, he earned his PhD on the subject 'An integated approach for tool path planning and generation for multi-axis milling'.

Jean-Pierre Kruth is a full time professor at the Catholic University of Leuven (K.U. Leuven, Belgium), where he is responsible for Production Engineering research and education at the division PMA of the Department of Mechanical Engineering. His activities involves: CAD, CAPP, CAM, production processes, metal cutting, non-traditional machining, rapid prototyping, dimensional metrology, quality control, reverse engineering. He is an active member of CIRP, Fellow of SME, Honorary member Romanian Society of Mechanical Engineers, Member of Belgian Royal Academy Council of Applied Sciences, Board Member BSMEE, Member of IMEKO/BEMEKO (International/Belgian Measuring Confederation), member KVIV, founding board member of companies MATERIALISE N.V. (>1997), METRIS N.V. and MIH N.V.