



Point cloud based tool path generation for corrective machining in ultra-precision diamond turning

Marco Buhmann^{1,2} · Erich Carelli³ · Christian Egger³ · Klaus Frick³

Received: 5 October 2021 / Accepted: 7 March 2022 / Published online: 20 April 2022
© The Author(s) 2022

Abstract

The increasing demand for machining non-rotational optical surfaces requires capable and flexible cutting tool path generation methods for ultra-precision diamond turning. Furthermore, the recent interest in on-machine metrology and corrective machining requires efficient as well as accurate algorithms capable to handle point cloud based surface data. In the present work, a new computation method for the tool path generation is proposed that focuses on three-axes corrective machining. It is based on the principle of defining the surface to be machined by a point cloud of given density, since surface measurement data is usually available as point cloud. Numeric approximation techniques are used to compute the surface normal vectors and calculate the resulting positions of the cutting tool path preserving a uniform radial axis motion for face turning. Investigations are performed in order to quantify the error between the calculated tool path and the exact analytical solution. The error dependencies are analyzed regarding the local surface slope and numerical parameters. Error values below 1 nm are achieved. In addition, form deviation results prove the method's capability for corrective diamond turn machining.

Keywords Ultra-precision machining · Freeform machining · Tool path generation · Diamond turning · Corrective machining

Mathematics Subject Classification MSC code1 · MSC code2 · More

1 Introduction

Diamond turning and ultra-precision machining technologies today in general offer the capabilities to machine surfaces with form deviations from the nominal surface in the range of 100 nm and roughness values below 5 nm. Besides the use of ultra-precision machine tools, whose structure and components offer high level movements in terms of both precision and accuracy, cutting tools made of single crystal diamonds are employed to achieve the mentioned surface quality. Hence, ultra-precision machining is often referred to as diamond machining [1].

Recently, the popularity of three-axes diamond turning increased in machining of non-rotational symmetric geometries including complex aspheres, lens arrays and freeforms in general. Freeform optics enable new possibilities for designers of optical systems for example in terms of a reduced number of optical elements and an increased optical performance [2]. This leads to a growing demand for precision freeform surfaces and thus forces diamond machining technologies to comply with this request. Corrective machining in combination with on-machine surface metrology is a promising approach for increasing diamond turning capabilities [3]. In view of these developments effective tool path generation is of high importance, especially for corrective machining.

Within the tool path generation the tool geometry, consisting of the nose radius and the rake angle, as well as the machining parameters like feed rates and NC point resolution need to be taken into consideration. Figure 1 shows the generalized steps of machining an optical surface and the accuracy losses throughout the process. Regarding the tool path generation, the aim is to minimize its related

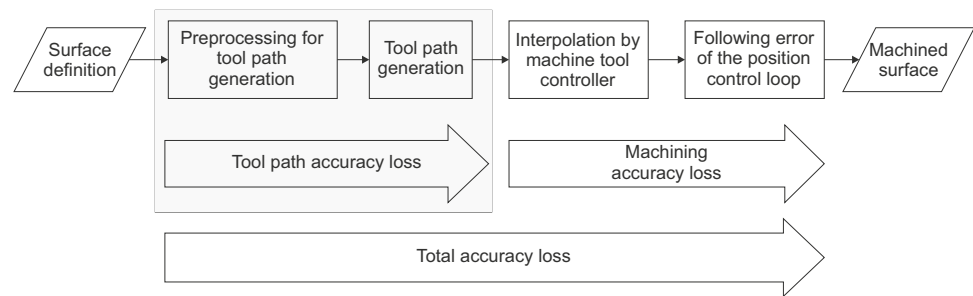
✉ Marco Buhmann
marco.buhmann@rhysearch.ch

¹ RhySearch, Buchs SG, Switzerland

² Institute of Machine Tools and Manufacturing (IWF), Swiss Federal Institute of Technology, Zurich, Switzerland

³ Institute for Computational Engineering (ICE), Eastern Switzerland University of Applied Sciences, Buchs SG, Switzerland

Fig. 1 Processing steps of diamond machining a surface and the related accuracy loss divided into two parts. Part one: Tool path accuracy loss arising out of the calculation of the NC points. Part two: Machining accuracy loss represented by the difference between the programmed path and the finally executed path of the axes



accuracy loss in order to decrease the overall accuracy loss along all processing steps. For different surface definition methods or surface data available as measured point cloud in case of corrective machining additional format transformations, e.g., fitting operations, may be necessary.

For the common use case of machining surfaces for optical applications by face turning, the tool path topography mainly follows a spiral. Regarding the tool nose radius compensation, Gong et al. [4] distinguish and discuss two different ways of projecting the spiral tool path onto the surface to be machined. The first is called oscillating-X method, since it can cause oscillating movements in X-direction having the potential to negatively affect the machining process due to high dynamic movements. The second leads to a constant feed in the radial direction and is named as steady-X method. Since constant or slowly changing feed rates prevent the process from any high dynamic movements, the steady-X method is generally preferred, but also more complex in terms of the tool-surface contact points calculation based on a predefined spiral tool path. Several publications discuss the issues in this context and present diverse approaches for the tool path generation for freeform machining. Gong et al. [4] give a comprehensive review of tool path generation techniques in ultra-precision diamond turning. Besides, Brinksmeier et al. [5] focus on tool path generation strategies for freeform surfaces.

A widely used method to describe freeforms is using non-uniform rational B-splines (NURBS), as described in [6, 7]. NURBS offer a framework to represent freeforms in a standardized way and to get an analytical description between given sampling points. However, Gong et al. [4] point out that there is no obvious correspondence between the parameter of the NURBS and the coordinate parameters and that *analytic parameter conversion is often impossible*. Moreover, also stated in [8], the NURBS approximation may not reach the desired accuracy. To overcome the problem it is possible to sample the surface and use a function fitting to obtain an analytical formula, see Scheiding et al. [9]. This process may suffer from accuracy and time consumption, especially in case of sampling noise or uneven point distribution, see [10] and references therein. In [4] it is noted that a similar algorithm is used in NANOCAM 2.0, but which may lead to accuracy losses.

To overcome the drawback of accuracy losses when approximating an analytic surface description by NURBS, Gong et al. [8] present a tool path generation method using symbolic computation for the calculation of the surface normals. However, the finding of the tool-surface contact point still requires numerical computation, since this usually cannot be done analytically for a given spiral path [8].

Moreover, there are machining applications which are forced to handle the surface to be machined defined by (non-uniform or uniform) point clouds. An example is machining related to reverse engineering, where surface information typically consists of coordinate points gained by 3D scanning (as stated in [11, 12]). Masood et al. [11] present a direct machining concept for three-axes milling using ball-end tools, which avoids the traditional surface fitting process in reverse engineering applications. The point cloud is divided into fields which correspond to one of the parallel oriented milling paths. Each path is then obtained by fitting a curve to the tool center points of the respective field, which are obtained using the normals of the triangulated surface data. The accuracy of the employed 3D scanning equipment is stated to be ± 0.018 mm. However, no analysis is presented regarding the tool path accuracy compared to the point cloud accuracy and to the method using surface fitting processes.

Chui et al. [12] also present a direct machining approach tailored to milling using ball-end tools. It focuses on the applicability to both regularly and irregularly distributed point clouds. After subdividing the points into fields, whose width depends on the step-over of two adjacent milling paths, the tool radius center points are obtained by geometric calculations determining the contact points between a surface point and the tool. A discussion of the resulting tool path accuracy is not part of their work.

Corrective machining of surfaces is an application, where the actual measured surface form is mainly available as point cloud. In this context, Xiao et al. [13] present an algorithm providing direct tool path generation for the repair machining of aero-engine blades based on a measured point cloud. The approach avoids surface fitting operations as well as the usage of CAM software and is designed to feature five-axes milling. Hence, it is

concluded to be more efficient than traditional methods. The accuracy of the employed scanning measurement is able to reach a value of 30μ . Besides, algorithms are presented for the region separation of the blades and recognition of damaged areas. The tool path accuracy is not explicitly analyzed, but is stated to be mainly influenced by the point cloud's accuracy and density.

As mentioned before, corrective machining is also becoming more popular in ultra-precision diamond turning and is strongly encouraged by the recent interest in on-machine surface metrology [3]. Although the use of a point cloud for the tool path generation is not further analyzed in [5], it is listed in the given overview as a possible strategy for ultra-precision machining. The named fact that numeric computations cannot be avoided entirely for a given spiral path (see [8]) and point cloud form measurement data motivate a point cloud based surface representation throughout the tool path generation process, which, in contrast to a hybrid process according to [8] using both analytic and numeric computations, for example also allows to handle freeform point clouds exported from a CAD design software. These aspects together with the characteristics and benefits of the above-described applications of direct machining presented in [11–13] lead to the demand for a tool path generation method combining efficiency in processing point clouds and enhanced accuracy to the needs of diamond turning.

Therefore, this work presents a novel simplified method that uses a point cloud and numerical computations to calculate the tool path for 3D diamond turning. It follows the explained and preferred steady-X method to maintain a uniform feed rate in radial direction, is capable to handle various tool rake angles and offers the option to compensate the point cloud by measured form errors in case of corrective machining. Since this strategy implies a certain accuracy loss due to no gapless analytical surface description and the employment of numeric computation methods, a comprehensive accuracy loss analysis is presented as well. The analysis and the method in general follow the aim of calculating tool path points of highest accuracy. This is required for any tool path point in diamond machining, regardless of the point density with which a tool path is programmed.

As part of the analysis every calculation step of the presented approach accompanied by a numerical approximation is investigated regarding its accuracy. Thus, the present work supplements the important analysis, which compares the point cloud approach with the exact solution and hence mitigates the shortcomings in the analysis of [11–13]. In contrast to these, the selection of the computational methods and its parameters is deduced and justified by analyzing various conditions regarding the accuracy demands in diamond turning.

Beyond, exemplary form error results prove the usability for freeform corrective machining in combination with the use of on-machine metrology.

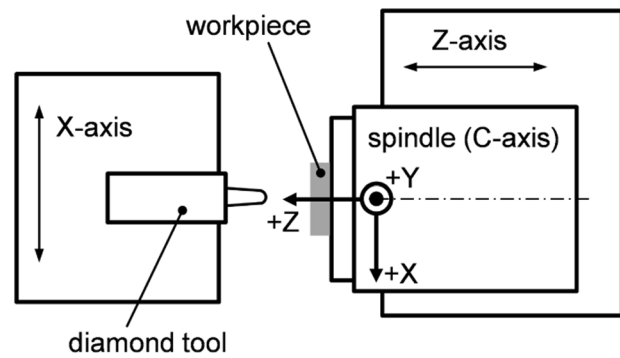


Fig. 2 Axes configuration of the used three-axes diamond turning machine consisting of two horizontal linear axes (X- and Z-axis) and one horizontal rotational axis (C-axis) with stationary right-handed Cartesian machine tool coordinate system. The configuration corresponds to a typical one used in diamond turning

2 Tool path generation

The configuration of the ultra-precision diamond turning machine used for this work is depicted in top view in Fig. 2. In contrast to the more usual configuration, the spindle is mounted on the Z-axis slide. The two horizontal linear axes are structurally separated and carry either the diamond tool or the workpiece spindle (dependent on the individual machine tool type), which operated as a controlled C-axis enables three-axes machining. Shown is also the stationary Cartesian machine tool coordinate system, in which the X-axis corresponds to the radial linear axis of the machine tool. It serves as the coordinate system used throughout the present work.

As a speciality of diamond turning machines, a movement in Z-direction can be performed by two different modes, namely either by the Z-axis itself (referred to as slow slide servo machining), or by a fast tool servo device (referred to as fast tool servo machining). In this context it is noted that the tool path generation generally is independent of the Z-axis mode used for machining (slow slide or fast tool servo), because both modes move according to the Z-coordinates calculated with respect to the machining coordinate system. The selection of the Z-axis mode takes place when programming the tool path to the machine tool control using the control-specific syntax, but which is not part of the present work.

The machining kinematics in face turning and the typical cutting tool geometry of a circular arc with certain radius cause the contact point between the workpiece surface and the tool to move along the tool nose arc when machining curved geometries. This behavior requires to compensate the nominal spiral tool path in function of the tool nose radius and the surface normal vector at each tool path point. For that, the ability to calculate the surface normal vector at

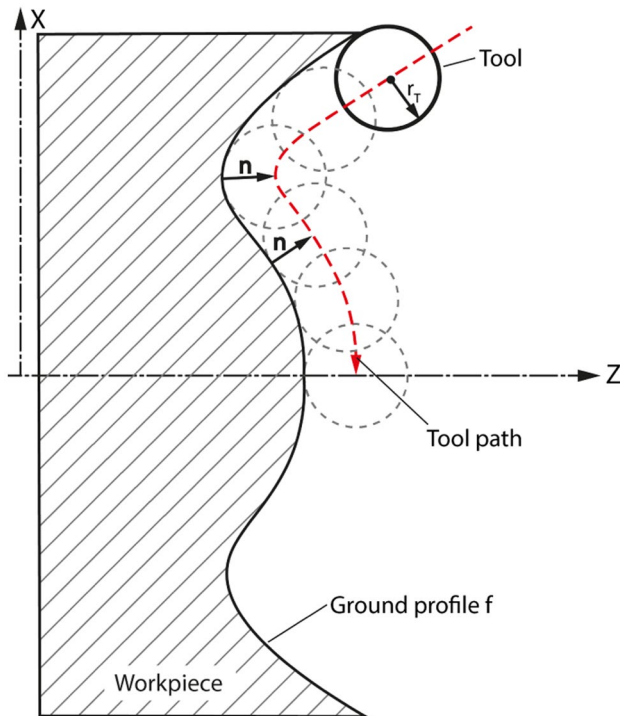


Fig. 3 Relations between a nominal surface profile f , its normal vectors \mathbf{n} , the tool nose radius r_T and the tool path

each point of the surface to be machined is crucial. Figure 3 illustrates the relation between a surface profile f and the associated tool path.

When calculating the surface normals by analytical methods, additional precomputation for each specific surface equation is necessary. This is avoided by representing the nominal surface by a point cloud as described in Sect. 1. The point cloud represents a flexible basis for the overall tool path generation and can be generated out of any nominal surface description by an initial preprocessing step. Two further steps complete the path generation, namely the calculation of the tool center surface and of the tool path points.

2.1 Calculation of surface normals

By machining a general freeform surface using a circular radius cutting tool the tool's radius center point has to move on a surface that has a constant distance to the surface to be machined. Hence, the first step of the tool path generation is the calculation of the normal vectors of the surface to be machined. The surface is considered as a point cloud P consisting of k points \mathbf{w}_i with the coordinates x_i , y_i and z_i according to the definition given by Eq. (1).

$$P = \{\mathbf{w}_i = (x_i, y_i, z_i) \in \mathbb{R}^3 \mid i = 1, 2, \dots, k\}. \quad (1)$$

As already mentioned, the point cloud can be derived by an initial preprocessing step out of any surface description method. Its properties and the associated effects are discussed in Sect. 3 subsequently.

For the common case of an analytical surface description $z = f(x, y)$, the surface normal vectors $\mathbf{n}(x, y)$ can be derived using the known analytical formulas. Although this description method is common for example for aspherical surfaces, it cannot be used for arbitrary freeform surfaces or for the correction of surfaces with measured points, which are usually available as point cloud. Using a point cloud as surface description method overcomes this drawback and represents the basis for a generalized surface normal calculation. Therefore, in a first step, the method described by Hoppe et al. [14] is utilized, where the normal vector to a point is computed using a plane fitted to its neighbors, according to the following procedure:

Finding the k_n nearest neighbor points: To accelerate the computational search of the k_n nearest neighbor points a quadtree, i.e., a tree data structure in which each internal node has exactly four children, is used. For this, the implementation by Walker [15] is used, which is based on the work of Frisken and Perry [16].

Fitting of the plane: For every point $\mathbf{w}_i = (x_i, y_i, z_i)^T \in P$ a set of nearest neighbor points $\mathbf{w}_j \in P$, $j \in K_i$, $K_i \subset \{1, 2, \dots, k\}$, where K_i is a set of neighbor points indices, are defined. By computing the centroid \mathbf{c}_i of the neighbor points \mathbf{w}_j the 3×3 matrix \mathbf{A}_i can be calculated using Eq. (2). \mathbf{A}_i is symmetric and positive semidefinite, thus the eigenvalue decomposition can be determined by Eq. (3), where $\mathbf{D}_i = \text{diag}(\lambda_i^1, \lambda_i^2, \lambda_i^3)$ and $\mathbf{V}_i = (\mathbf{v}_i^1, \mathbf{v}_i^2, \mathbf{v}_i^3)$ contains the corresponding eigenvectors.

$$\mathbf{A}_i = \sum_{j \in K_i} (\mathbf{w}_j - \mathbf{c}_i) \cdot (\mathbf{w}_j - \mathbf{c}_i)^T. \quad (2)$$

$$\mathbf{A}_i = \mathbf{V}_i \mathbf{D}_i \mathbf{V}_i^T \quad (3)$$

Assuming that the eigenvalues are arranged in decreasing order, then $\mathbf{n}_i := \pm \mathbf{v}_i^3$ represents the normal vector of the fitted plane to the nearest neighbor points of \mathbf{w}_i . The sign of the vector is then chosen according to the machining orientation within the machine tool coordinate system: Following the definition in Fig. 2 the Z-component of the normal vector is chosen to be positive.

In this way, for the point cloud P a set of normal vectors N_p , one for each point \mathbf{w}_i , is obtained according to Eq. (4).

$$N_p = \{\mathbf{n}_i \in \mathbb{R}^3 \mid i = 1, 2, \dots, k\}. \quad (4)$$

It may be noticed, that this computation of the normal vectors for a point cloud is already implemented in some commercial software, e.g., in the Computer Vision System

Toolbox of MATLAB or open3D for Python. Since the computation of the normals plays a crucial role in the computation of the trajectory, in particular for big tool radii, also a modified version of [14] and additionally the method from [17] are considered. The latter is based on a weighted average of the normals to the neighboring triangles obtained by a Delaunay triangulation. However, as it will become clear in Sect. 3, using an equally distributed point cloud leads to almost optimal accuracy¹ for a suitable point density.

$$E = \{ \mathbf{u} \in \mathbb{R}^3 \mid \langle \mathbf{u}, \mathbf{n}_T \rangle = 0 \}, \tag{5}$$

2.2 Calculation of tool center surface

Once the normal vector of each surface point is calculated, the surface on which the tool center point moves can be computed. The most common cutting tools used in diamond turning, especially for machining freeforms, have a circular cutting edge defined by the nose radius r_T and the geometrical center point coordinates $\mathbf{c}_T = (x_T, y_T, z_T)$ in the machine tool coordinate system. The tool modeling assumes an ideally sharp cutting edge, which together with the tool center point defines the rake face plane E_T with the corresponding normal vector \mathbf{n}_T according to the definition given by

where \mathbf{u} are the points on the plane and the vector \mathbf{n}_T in the surface reference system is given by

$$\mathbf{n}_T = \begin{pmatrix} -\cos \beta \sin C \\ -\cos \beta \cos C \\ \sin \beta \end{pmatrix}. \tag{6}$$

Here β and C are the rake and the C-Axis angle, respectively. For the general situation, the tool rake face is not always parallel to the XZ-plane but tilted by the tool’s rake angle.

For a given point $\mathbf{w}_i \in P$, i.e., a point on the surface to be machined, the corresponding tool center point is computed by projecting the surface normal vector \mathbf{n}_i onto the plane E_T , accordingly scaling its length and adding the resulting vector $\tilde{\mathbf{n}}_i$ to the surface point. The according mathematical procedure is described in the following and corresponds to the depiction in Fig. 4.

The surface normal vector component in direction of \mathbf{n}_T is subtracted from \mathbf{n}_i and scaled according to Eq. (7).

$$\tilde{\mathbf{n}}_i := \frac{\hat{\mathbf{n}}_i}{\|\hat{\mathbf{n}}_i\|}, \quad \hat{\mathbf{n}}_i := \mathbf{n}_i - \langle \mathbf{n}_i, \mathbf{n}_T \rangle \cdot \mathbf{n}_T. \tag{7}$$

¹ This is apparently in contrast with the results from [17], but there the goal is to reconstruct a surface from unorganized points. The present case has organized points from a smooth surface. This allows the use of less neighbor points to estimate the plane for the normal.

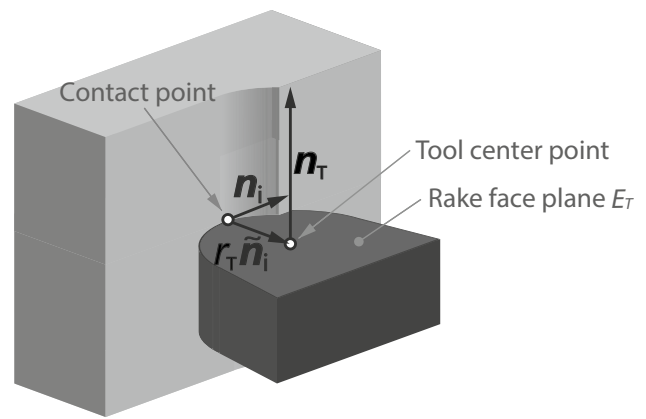


Fig. 4 Vectors arising from the tool definition and corresponding projections

Finally, the surface (or the point cloud) P_T on which the tool center point moves is computed by extending the surface points \mathbf{w}_i along the obtained vectors according to the definition given in Eq. (8).

$$P_T = \{ \mathbf{v}_i \in \mathbb{R}^3 \mid \mathbf{v}_i = \mathbf{w}_i + r_T \tilde{\mathbf{n}}_i, i = 1, 2, \dots, k \} \tag{8}$$

2.3 Calculation of the trajectory

Usually, the basis tool trajectory in face turning of optics is supposed to be an Archimedean spiral defined in the XY-plane. Equation (9) defines the spiral T_{2D} in the Cartesian machine tool coordinate system in function of the time t using the cylindrical coordinates consisting of a parametrization ϕ of the C-axis angle and the radial coordinate r . The radial coordinate r corresponds to the distance to the spindle axis C.

$$T_{2D}(t) = \begin{pmatrix} r(t) \cos \phi(t) \\ r(t) \sin \phi(t) \end{pmatrix}. \tag{9}$$

Once P_T is determined, the 2D trajectory is projected onto it along the Z-direction. By interpolating the Z-values within P_T the tool path points are generated. The position of the trajectory points in the XY-plane remains unchanged during this operation. As a byproduct, using the mapping $\mathbf{v}_i \mapsto \mathbf{w}_i$ the cutter contact points are obtained.

By choosing the type of interpolation it is possible to improve the accuracy, see e.g., Theorem 4.4.20 in [18], where estimations of the interpolation error depending on the differentiability of the surface, the degree of the interpolation polynomial and the size of the mesh are given. Since for every point the neighbor points were already determined in order to compute the normals (see Sect. 2.1), it is further possible to interpolate only locally to improve the memory management during the computations.

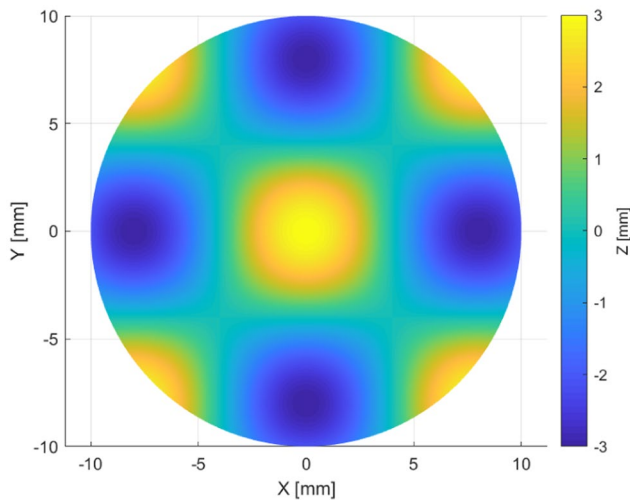


Fig. 5 Freeform surface from definition (10), with $b = 3$

3 Analysis of accuracy loss

In this section, the different error sources that affect the tool path accuracy loss as defined in Fig. 1 are analyzed for various surfaces. In a first study, only the calculation of the normal vectors for a point cloud, see Sect. 2.1, is considered and the results are compared with the exact normal vectors. The influence of the definition of the point cloud P , the number of nearest neighbors k_n and the used methods are discussed. In a second study, the interpolation error when interpolating the Z -values of the tool path points onto the tool center surface P_T is analyzed and the trajectories resulting from a point cloud and from the corresponding analytical calculation are compared.

3.1 Accuracy of the normals calculation

First, a freeform surface defined by Eq. (10) is considered, see also Fig. 5.

$$z_1(x, y) = b \cos \frac{\pi x}{8} \cos \frac{\pi y}{8}, \quad b \in \mathbb{R} \quad (10)$$

The study investigates the accuracy of the surface normals calculation, is conducted using $b = 3$ and is independent of the machine tool and the cutting tool. It is assumed that the unit of the coordinates is mm. The normal vectors are calculated for points (x_i, y_i) lying within a circle of radius 10 mm, for a number of points $k \in \{1E6, 4E6\}$ and different numbers of neighbor points k_n between 3 and 20, for both a Cartesian mesh and quasi-random points from the Halton sequence. The first grid distribution is motivated by the fact that some measuring devices, e.g., a Fizeau interferometer, return values on a Cartesian mesh, while the second

distribution because of its low discrepancy, i.e., the points are random but evenly distributed.

Then, motivated by on-machine metrology two additional methods are considered, both relying on surface data from a spiral distribution, as if it were generated by a spiral scanning measurement. Taking an exemplary but usual configuration, the spiral points have a maximal arc distance ds of $8.7 \mu\text{m}$ or 10° angular distance, while the distance dr between every C -axis turn is set to be $8.7 \mu\text{m}$, resulting approximately $4E6$ points lying within a circle of radius 10mm. Once the data is interpolated on a Cartesian grid the new grid points are used to compute the normal vectors. In the second case the method from [17] is directly applied to the spiral points to obtain the normal vectors. In this case, the number of neighbors is determined automatically for each point (the user cannot control it) using a Delaunay triangulation and each normal vector is computed using a weighted average of the normals to the faces surrounding the point.

To calculate the error of the used methods for computing the surface normals, the angle e_i^{ang} between the estimated normal vector \mathbf{n}_i and the analytical formula $\mathbf{n}(x_i, y_i)$ is calculated according to Eq. (11).

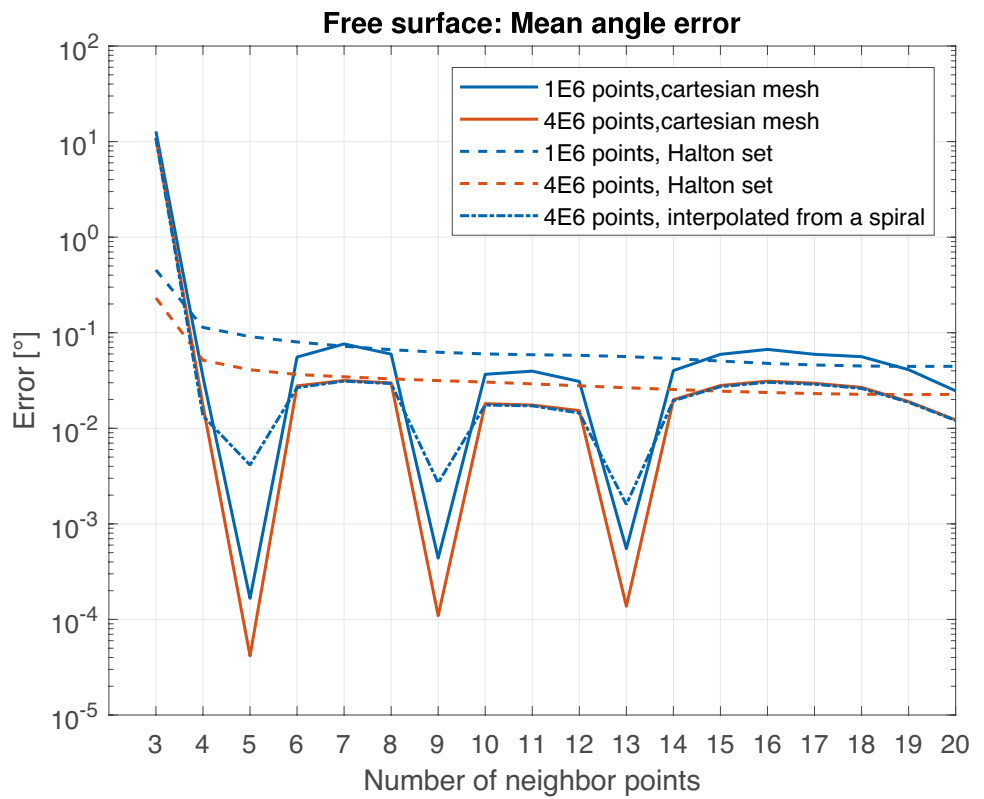
$$e_i^{ang} = \arccos(\mathbf{n}_i \cdot \mathbf{n}(x_i, y_i)) \quad (11)$$

The resulting mean values of the error angles for $k \in \{1E6, 4E6\}$ in function of the number of neighbor points k_n are shown in Fig. 6. The main observation is that in the case of a Cartesian mesh the mean error has local minima at $k_n = 5, 9$ and 13 , which show an improved error behavior in comparison to all other configurations. In the way the search for the neighbor points is defined, the actual point is considered as its own neighbor, meaning that the real numbers of neighbors for the above-mentioned amounts are $4, 8$ and 12 . The resulting neighbor points configurations within the regular grid are depicted in Fig. 7. For these special configurations, independently of the surface, the minima are a consequence of the approximation property of the central finite difference, which is of second order with respect to the mesh width, see Appendix 1.

The approximation of a function's derivative depends on the distance of the points used to compute it. Thus, by increasing the distance of the neighbor points, the accuracy of the proposed method may decrease, as it can be seen by comparing the curves with $1E6$ and $4E6$ points in Fig. 6. In contrast to the Cartesian grid, the quasi-random mesh (Halton set) does not show these local minima because the second-order approximation property is not valid for quasi-random points.

In case of the interpolated spiral data, the local minima are observed again, with the difference that the best approximation is for $k_n = 13$ points. This level of accuracy is reached also by the method from [17] applied directly to the spiral points, as shown in Table 1.

Fig. 6 Mean angle error as a function of the number of neighbor points for the freeform surface z_1 calculated using a Cartesian mesh and a Halton point set as well as two different numbers of surface points. The error improvement for the symmetric stencils is clear



In this first attempt to study the approximation of normal vectors it is noted that the choice of the grid points play an essential role in the determination of the normal vectors. First, the best results are obtained on a Cartesian mesh with suitable value k_n . Second, in the case of the spiral it turned out during the computations that the spiral points should be chosen to be well distributed, namely the arc length between the points should be approximatively equal to the distance of the spiral turns, as it can be observed in Table 1. The case of the table’s last row is computed to show how the form of the triangles affects the precision of the method from [17].

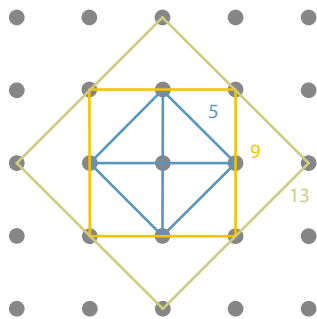


Fig. 7 Configurations to calculate the surface normals by fitting a plane to various chosen numbers of neighbor points k_n

3.2 Accuracy of the tool path points

Since not only the angle error resulting from the surface normals calculation contributes to the tool path accuracy loss, also the contribution of the interpolation of the tool trajectory points needs to be taken into account. Therefore, this section considers the error sum due to the interplay of approximation of the normal vectors and the interpolation in order to quantify the tool path accuracy loss according to Fig. 1.

The trajectory for this investigation is an Archimedean spiral with a maximal radius $R_{out} = 10$ mm where the spiral points have a maximal arc distance of 1 mm or 1° angular distance, while the distance between every C-axis turn is set to be $10 \mu\text{m}$. The cutting tool is supposed to have a nose radius of 1 mm and a rake angle of 0° . The XY-coordinates of the point cloud lie on a regular Cartesian grid. To compute the surface normals 5 neighbor points according to Fig. 7 are used.

For this study two surfaces are considered. The first is defined by Eq. (12), see Fig. 8. The second is defined by z_1 according to Eq. (10).

$$z_0(x, y) = ax, \quad a = \tan \alpha, \quad \alpha \in \mathbb{R}, \quad (12)$$

Table 1 Comparison of the mean angle error for the freeform surface z_1 for several methods, in particular the method from [17]. In the last line the parameters in order to have very skewed triangles are chosen

Method	Grid	Number of grid points	Neighbors	dr [mm]	ds [mm]	Mean error [°]
Plane fitting	Cartesian	4E6	4	-	-	4.1599E-5
Plane fitting	Cartesian	4E6	5	-	-	2.7813E-2
Plane fitting	Halton set	4E6	4	-	-	9.1058E-2
Plane fitting	Cartesian (interp. from spiral)	4E6	13	0.0087	0.0087	1.5988E-3
Method from [17]	Spiral grid	4E6	-	0.0087	0.0087	3.2871E-3
Method from [17]	Spiral grid	2E6	-	0.15	0.001	4.7944E-2

The first surface z_0 has zero curvature and the tool center trajectory is computable by analytic methods and therefore considered to be known. Since it is a plane, the surface normals calculation does not lead to any inaccuracies, allowing to quantify the error contribution due to interpolation only for several angles α . This case needs particular attention, because since the surface z_0 is tilted by α it cannot be machined in the center region without further attention. Therefore, the center region of $r < 0.5\text{mm}$ is not considered by this investigation. The second is a general freeform surface, but since the exact solution for the tool trajectory points using the steady-X method cannot be determined by using only analytic mathematics, it is proceeded in the following way: It is supposed that the tool center point moves on the surface according to Eq. (10) and a point cloud \widetilde{P}_T is computed using Eq. (10). Then the point cloud P , defining the surface to be machined, is computed using the tool center surface (given by \widetilde{P}_T). In this way the tool center surface \widetilde{P}_T is analytically defined by the surface Eq. (10). Using the points P , the point cloud P_T can then be calculated according to the presented procedure. To evaluate the error the tool trajectory points interpolated within P_T are compared to the exact values computed using Eq. (10).

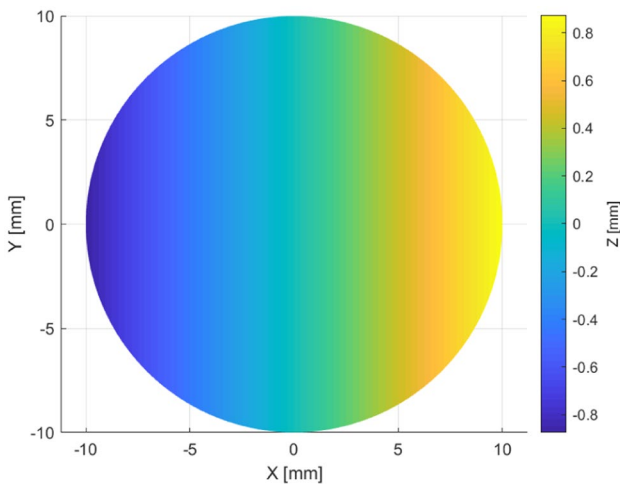


Fig. 8 Surface from definition (12) with $\alpha = \tan(5^\circ)$

To quantify the error of the trajectory, first the spiral points $(x_\ell, y_\ell), \ell = 1, \dots, N$ are computed, then the interpolation of the points in P_T is used to obtain the values of the Z-axis Z_ℓ . These are then compared with the exact values $z_\ell = z(x_\ell, y_\ell)$, where z is the surface to be machined and chosen according to the Eqs. (13) and (14).

– Planar surface z_0

$$z_{0,\text{tool}}(x_\ell, y_\ell) = a x_\ell + r_T \sqrt{1 + a^2 \frac{x_\ell^2}{x_\ell^2 + y_\ell^2}}, \tag{13}$$

– Freeform surface z_1

$$z_{1,\text{tool}}(x_\ell, y_\ell) = b \cos \frac{\pi x_\ell}{8} \cos \frac{\pi y_\ell}{8}. \tag{14}$$

By computing the two quantities

$$e_{\text{mean}} = \frac{1}{N} \sum_{\ell=1}^N e_\ell = \frac{1}{N} \sum_{\ell=1}^N |Z_\ell - z_\ell|, \tag{15}$$

$$e_{\text{max}} = \max_{1 \leq \ell \leq N} e_\ell = \max_{1 \leq \ell \leq N} |Z_\ell - z_\ell|, \tag{16}$$

the differences in Z-direction between the trajectory computed by the proposed method and the exact trajectory are obtained.

In the Figs. 9 and 10 the local errors e_i are plotted in function of the coordinate in C-direction over one rotation for different inclination angles α and two different radial positions r . A sinusoidal behavior of the error with the amplitude gradually increasing with the tool moving closer to the rotation center can be observed. This is supposed to be related to the decreasing wavelength of the sinusoidal changes of the Z-values when machining the inclined planar surface towards the center. The global error values depending on the inclination angle are given in Table 2, where, as expected, increasing error values for a more inclined planar surface are observed. The given maximal values e_{max} are throughout $< 4 \text{ nm}$ and except the maximal

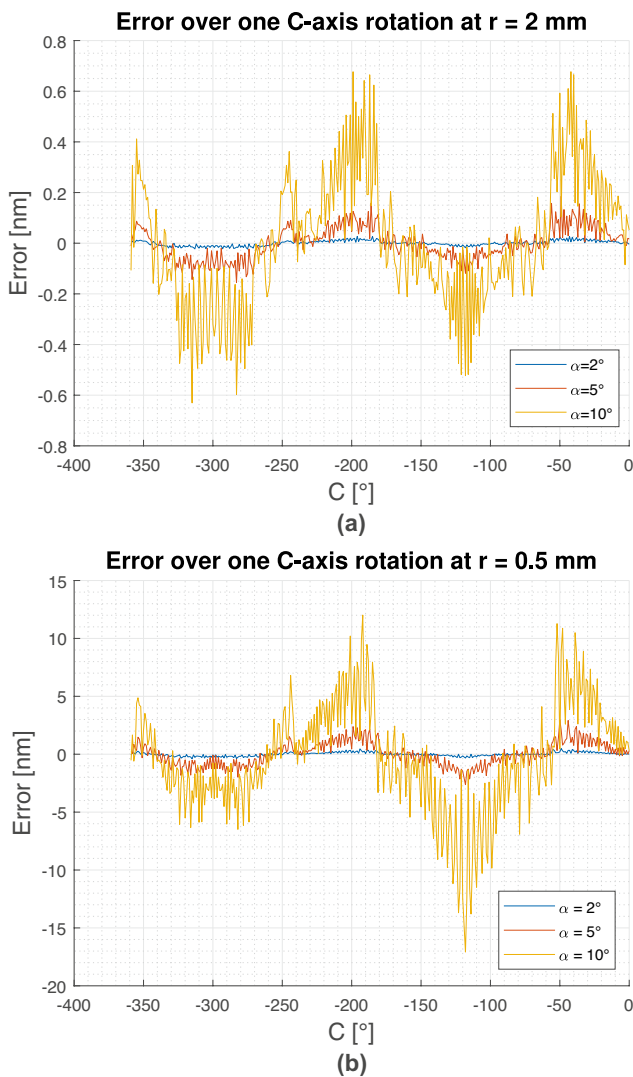


Fig. 9 Using 1E6 surface points for a surface size of $R_{out} = 10$ mm : Error between analytical and computed trajectory using the proposed method in the case of the planar surface z_0 with inclination α of 2°, 5° and 10° evaluated for a single spiral loop at the radii 2 mm (a) and 0.5 mm (b). The numerical experiment shows the dependence of the error on the slope and on the radius from the spindle axis

value of 3.9002 nm even < 1 nm. However, it should be noted that with an increasing inclination angle also other difficulties when machining the surface, for example a sufficient tool clearance angle, arise.

To study the approximation properties for the surface z_1 it is proceeded as already explained at the beginning of this section. Starting from the definition that $z_{1,tool}$ represents the tool center surface, the surface to be machined is computed by using analytical formulas to calculate the surface normals. Based on that, the tool center surface can be computed once more using the numerical method presented in the Sects. 2.1 and 2.2. This way, two point clouds for the tool center surface are derived, one by using exact mathematics

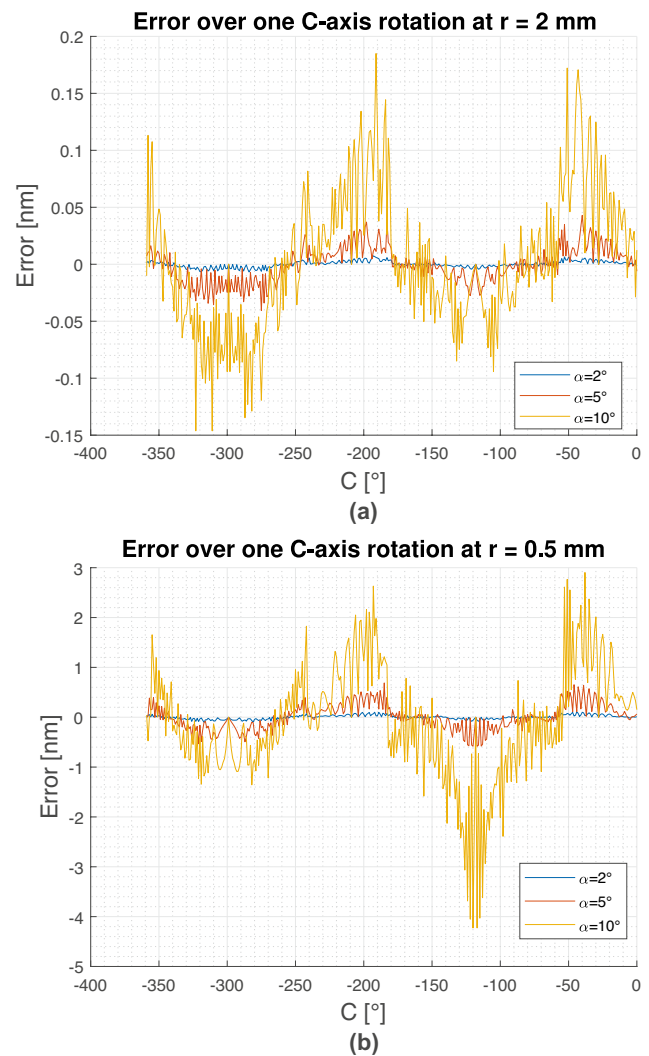


Fig. 10 Using 4E6 surface points for a surface size of $R_{out} = 10$ mm: Error between analytical and computed trajectory using the proposed method in the case of a planar surface z_0 with inclination α of 2°, 5°, and 10° evaluated for a single spiral loop at the radii 2 mm (a) and 0.5 mm (b). Again the dependence of the error on the slope and on the radius can be observed

and the other by using the presented numerical method. In addition, it should be noted that the analyzed error values may be greater than usual due to the repeated computation

Table 2 Comparison of the error values for the surface z_0 (inclined planar surface) using a regular grid with 4E6 surface points and linear interpolation. Due to a singularity, which appears near the center as a consequence of the constant surface slope, the error has not been considered for $r < 0.5$ mm

Surface	Plane angle α [°]	e_{max} [nm]	e_{mean} [nm]	σ [nm]
z_0	2	0.1051	0.0015	0.0005
z_0	5	0.6611	0.0096	0.0304
z_0	10	3.9002	0.0390	0.1302

Table 3 Comparison of the error values for the surface z_1 with different amplitude values b . The values are calculated using several numbers of surface points and linear interpolation

Surface	No. of points	e_{max} [nm]	e_{mean} [nm]	σ [nm]
$z_1, b = 3$	2.5E5	209.3377	70.3936	49.6146
	1E6	52.2725	17.7716	12.4963
	4E6	13.0668	4.3925	3.0956
	16E6	3.2657	1.0981	0.7736
$z_1, b = 1$	2.5E5	69.7992	24.2664	16.8613
	1E6	17.4218	6.1277	4.2409
	4E6	4.3554	1.5153	1.0514
	16E6	1.0885	0.3789	0.2628
$z_1, b = 0.1$	2.5E5	6.9807	2.5928	1.7739
	1E6	1.7422	0.6543	0.4452
	4E6	0.4355	0.1618	0.1106
	16E6	0.1089	0.0405	0.0277

of the normal vectors. The results are summarized in Table 3 for different values for the amplitude b . As can be seen, the error values e_{max} grow linearly as a function of the used number of surface points, which can be justified with the application of linear interpolation. Because of high slope angles and a certain variation of the Z-values depending on the amplitude b , a higher number of surface points is needed to achieve a reasonable small maximal error value along the trajectory. The surface with an amplitude of 3 mm ($b = 3$) is viewed rather as an example to prove the accuracy of the proposed method, since the maximal surface slope angle exceeds the typical available cutting tool clearance angles. Even for the surface with an amplitude of 1 mm ($b = 1$) the maximal error value can be reduced to about 1 nm by using a number of surface points of 16E6.

All of the previous calculations are performed using linear interpolation. However, it is possible to achieve greater accuracy by increasing the interpolation order. To illustrate the changes in the error values due to a higher interpolation order, the computation and analysis for the surface z_1 with $b = 3$ are performed additionally applying cubic interpolation. The results given in Table 4 show a significant improvement by means of greatly reduced error values to almost 20 % compared to the values given in Table 3 using linear interpolation. It can be emphasized that even for the chosen case of $b = 3$ the result of e_{max} is of sub-nanometer range for 16E6 surface points.

In Sect. 3.1 it is observed how much the method used to determine the normals affects the angle error defined by Eq. (11). To quantify their influence on the final trajectory the surface z_1 with $b = 3$ is considered with normal vectors obtained with different computation methods. To better show the effect of the normals cubic interpolation is used. As it can be seen in Table 5, the trajectory resulting from

Table 4 Comparison of the error values for the surface z_1 with $b = 3$ for several numbers of surface points in the case of cubic interpolation

Surface	No. of points	e_{max} [nm]	e_{mean} [nm]	σ [nm]
$z_1, b = 3$	2.5E5	47.3583	0.9630	3.3561
	1E6	11.8068	0.1399	0.6386
	4E6	2.8261	0.0206	0.1206
	16E6	0.6407	0.0032	0.0232

the quasi-random point cloud (Halton set) is at least two orders of magnitude less accurate than that computed with the Cartesian grid, while with the grid interpolated from a spiral distribution a similar maximal error is obtained. In the latter case, the mean error is increased remarkably, but remains below 1 nm. Also the tool path computed with the method from [17] shows an increased mean error, but still below 1 nm. The maximal error is slightly better than for the Cartesian grid. Investigations identified that this is caused by the double computation of the tool center points to obtain the exact solution, as explained at the beginning of this section. However, the standard deviation of the error as well as the mean error are smaller in the case of the Cartesian grid.

3.3 Analysis of machine tool interpolation

According to Fig. 1 the accuracy loss between the surface definition and the machined surface can be divided into two parts. The first is caused throughout the tool path generation and the second consists of the deviation resulting from the interpolation the machine tool controller applies on the programmed points as well as of the following error occurring in the position control loop of the axes drives. Whereas the first is analyzed and quantified in the previous Sects. 3.1 and 3.2, the second part addressing the accuracy loss related to the execution of the tool path points by the machine tool is not addressed so far. Hence, it is necessary to assess how the deviations of the tool path points compared to the exact calculation affect the execution by the machine tool. Although

Table 5 Comparison of the error values for the surface z_1 with $b = 3$. The influence of the computation of the normal vectors is investigated in the case of cubic interpolation with 4E6 surface points

Grid & normals	e_{max} [nm]	e_{mean} [nm]	σ [nm]
Cartesian grid	2.8261	0.0206	0.1206
Halton set	834.7151	2.0949	2.9579
Cartesian grid interpolated from spiral	3.0878	0.8819	0.5720
Spiral grid (method from [17])	1.8476	0.2421	0.2433

the execution quality is supposed to be also influenced by the individual employed diamond turning machine tool as well as chosen parameters (NC point resolution, cutting speed, feed rates), the following analysis is assumed to represent common conditions of diamond turning. The analysis is divided into two parts, whereas the first investigates the tool path of the surface z_0 and the second the one of z_1 . Both utilize a diamond turning machine of the type LT-Ultra MTC 650 having a POWER PMAC control.

3.3.1 Analysis of the surface z_0

For the first investigation Z-axis positions following a sine function with the C-axis positions as arguments are programmed to the machine tool and executed. Subsequently, the resulting path interpolated by the control is analyzed. A nearly sinusoidal movement in Z-direction in function of the C-axis angle corresponds to machining an inclined planar surface (see geometry z_0 according to Eq. (12)). However, for the present study the small differences to a real sine due to the radial feed rate in case of a spiral machining path and the tool nose radius compensation are neglected. The tested sine wave has a period of 4s and an amplitude of $15 \tan(5^\circ) \text{ mm}$ corresponding to a planar surface with an inclination of 5° at a radius of 15 mm .

To investigate the effect caused by the numeric inaccuracies elaborated in Sect. 3.2, the Z-values of the programmed sine are overlaid with a Gaussian noise and the resulting axes movements are recorded by reading out the position signal of the machine tool's scales. As

interpolation mode the spline mode of the POWER PMAC control is chosen, which interpolates programmed points by generating non-rational B-splines and is popular for executing curved paths represented by a multitude of points. It has the advantage of smooth axes movements, but the interpolated path does not exactly go through the programmed points. Hence, the interpolated sine wave commanded to the axes drives shows a slightly smaller amplitude compared to the initially programmed points. Figure 11 depicts one sine period consisting of points generated by the machine tool control based on programmed points without an overlaid noise and its deviations to a least squares sine fit. To exclude the effect of the reduced amplitude as far as possible the fitted sine function is supposed to represent the nominal movement for the analysis. This assumption is confirmed by the maximum deviation of just about $2E-4 \text{ nm}$ shown in Fig. 11.

The small standard deviations stated in Table 2 being in the sub-nm range are expected to result by the fact that the normals calculation for the flat surface z_0 is free of numerical inaccuracies. In this context it is expected that values below 1 nm barely affect the tool path execution. Therefore, the standard deviation σ_n of the overlaid Gaussian noises are chosen according to the values for the free-form surface z_1 given in Tables 3 and 4. With the value of 3.5 nm all values of Table 4 are covered, while the value of 5 nm covers most of the values in Table 3. Their effect is determined by the difference of the resulting movement path and the nominal sine function. Figure 12 shows the resulting movement errors over one sine period, which

Fig. 11 Commanded sine wave (interpolated points by the machine tool control) and deviation compared to a fitted sine function using the least squares method

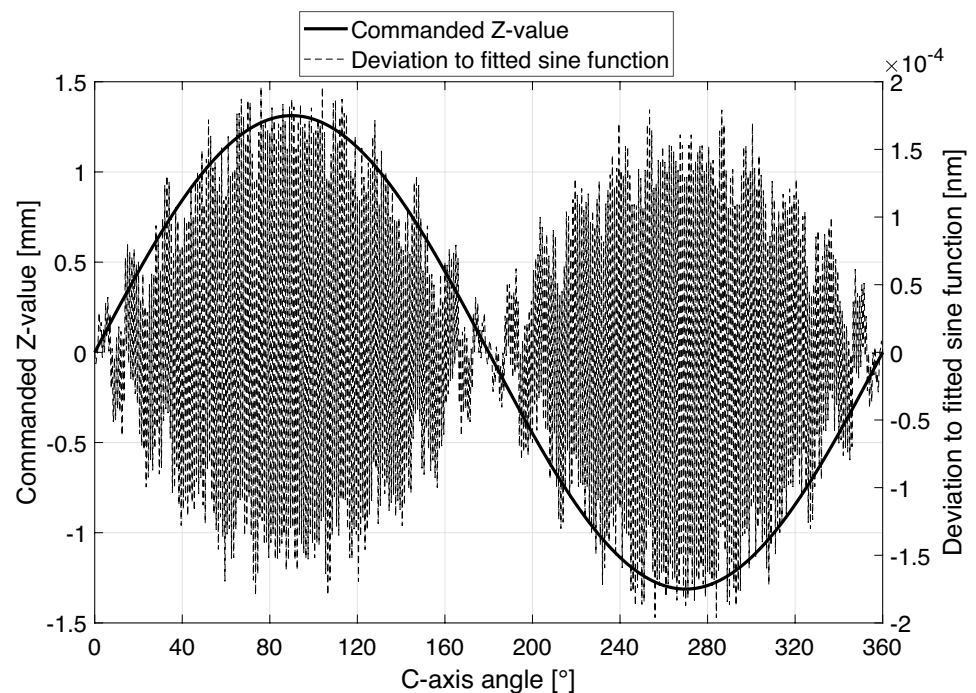
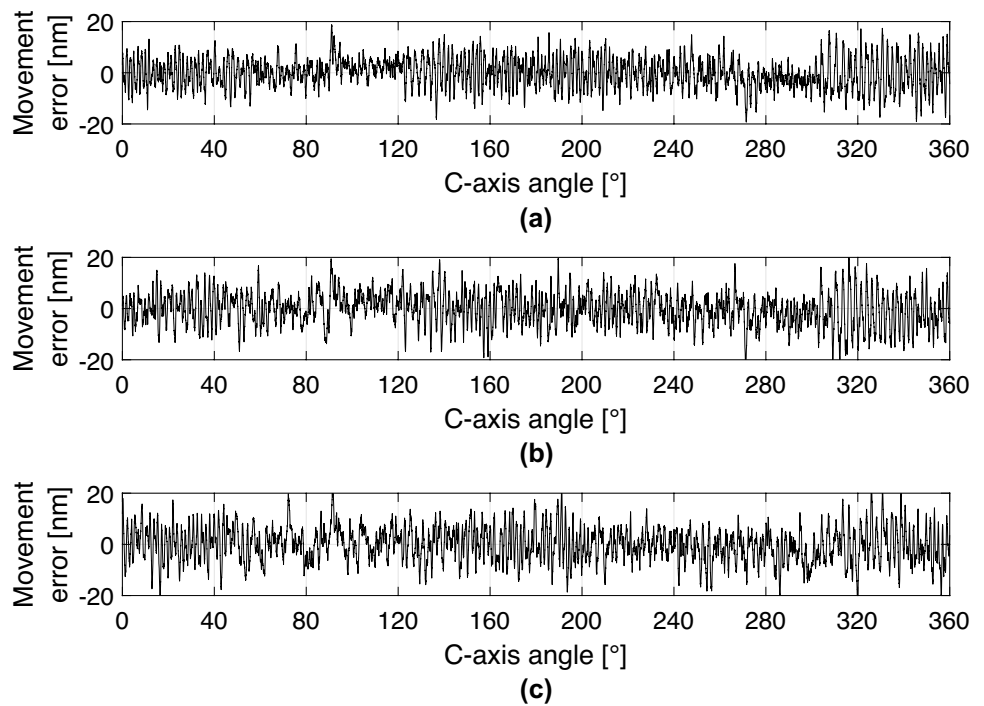


Fig. 12 Movement error calculated as the difference between the recorded Z-axis positions and the nominal sine function over one sine period. The programmed Z-coordinates are overlaid with a Gaussian noise having a standard deviation σ_n . (a) without overlaid noise; (b) $\sigma_n = 3.5nm$; (c) $\sigma_n = 5nm$. The depicted movement errors itself show a standard deviation of $5.8138nm$ (a), $6.4822nm$ (b), $6.7533nm$ (c)



standard deviations lie within a range of about $1nm$ and are $5.8138nm$ (no overlaid noise; Fig. 12a), $6.4822nm$ ($\sigma_n = 3.5nm$; Fig. 12b) and $6.7533nm$ ($\sigma_n = 5nm$; Fig. 12c).

Considering these results, it can be observed that the overlaid Gaussian noises barely affect the movement of the machine tool's Z-axis, since the standard deviations of the movement errors increase only by about $1nm$ compared to no overlaid noise. Due to the fact that the computation accuracy of the presented method can be increased even to values of $e_{max} = e_{mean} = \sigma < 1nm$ (see Tables 2, 3 and 4), the movement error and therefore also the tool path accuracy loss according to Fig. 1 resulting out of numeric inaccuracies can be assumed to be negligible.

3.3.2 Analysis of the surface z_1

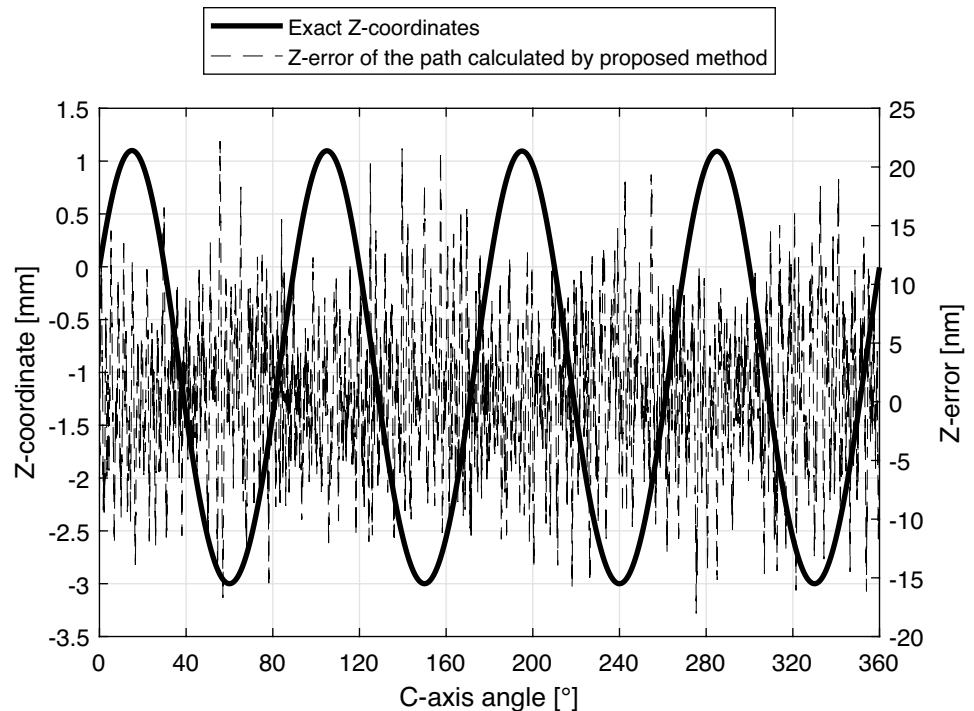
The second investigation addresses the tool path of the free-form surface z_1 by analyzing the tool path execution of one C-axis rotation at a radius of as shown by the exact Z-values in Fig. 13. Therefore, the exact solution and the tool path calculated by the proposed method are executed on the diamond turning machine while the actual axes positions are recorded from the axes' scales. By calculating the differences of the Z-coordinates between the two path executions the influence of the numeric inaccuracies is evaluated. Similar to the analysis in Sect. 3.2 the maximum error e_{Mmax} , the mean e_{Mmean} and the corresponding standard deviation σ_M over all recorded position points are calculated. The NC points programmed to the machine tool control have a spacing in

C-direction of 0.1° and a feed rate in X-direction per rotation of $10\mu m$.

The procedure is carried out for all parameter sets included in Tables 3 (tool center points determined by linear interpolation) and 4 (tool center points determined by cubic interpolation). Additionally, each parameter set is carried out with two numbers of C-axis rotations per minute (20 rpm, 40 rpm) to study the influence of different path execution velocities. Besides the exact tool path, Fig. 13 also shows the differences over one C-axis rotation for the parameter set of $b = 3, 16E6$ points for the tool path generation using linear interpolation and a C-axis rotation speed of 20 rpm.

Figure 14 shows the resulting path execution errors corresponding to the parameters listed in Table 3 for the case of linear tool path interpolation and Table 4 utilizing cubic path interpolation. The solid lines show the data for 20 rpm and the dashed lines the data for 40 rpm. There is a clear relation that higher numbers of points and smaller scaling factors b , which reduce the surface's curvature, result in decreasing motion error values. Most of the magnitudes of the motion error values agree well with the data in Tables 3 and 4. However, towards their minimum values in Fig. 14 they reach a relatively constant level, whereas the tool path errors (Tables 3 and 4) further decrease to even sub-nm values. It is supposed that this indicates the limit of the machine tool's motion accuracy, which is also confirmed by the standard deviations of the movement errors stated in Fig. 12 with about 6 nm being of the same order of magnitude as the minimum values in Fig. 14. The maximum error values e_{Mmax} do not fall below

Fig. 13 Exact Z-axis tool path coordinates over one C-axis rotation at a radius of 88 mm for the freeform surface z_1 and the motion error corresponding to the conditions: 20 rpm, $b = 3$, 16E6 points for tool path generation and linear interpolation



20mm, which is explained to be caused by the outlier data points being part of the recorded position signal from the machine tool's scale as shown in Figs. 12 and 13. Furthermore, the experiments conducted with 40 rpm C-axis rotation speed show slightly lower error values for most of the tested parameter sets. This can be interpreted in the way that the increased motion accelerations to some extent inhibit the axes to follow the inaccuracies of the tool path calculated using the proposed method. Thus, the executed motion shows a slightly better agreement with the execution of the exact tool path with 40 rpm than 20 rpm.

4 Corrective machining results

The direct advantage of the proposed method lies in the corrective machining using point cloud based measurement data. Since the tool path generation is also point cloud based, no surface fitting operation needs to be applied to the measured data and it can be easily compared to the nominal surface definition by calculating the difference in Z-direction. Prior to this, a 2D robust Gaussian regression filter is applied to remove noise and outliers.

To test this functionality a tilted flat according to Fig. 8, whose Z-values follow the function $z_F = x \tan(5^\circ)$, is machined on a diameter of 20 mm in oxygen-free copper using a spiral tool path towards the center ($X = 0$). This surface geometry is chosen, because its tilt requires three-axes

machining similar to freeform surfaces while the flat's form deviations can be verified using an external laser interferometer, which unlike a freeform metrology device was available for this work. Figure 15 depicts the machining setup for the experiment utilizing a diamond turning machine of the type LT-Ultra MTC 650, a monocrystalline diamond cutting tool with a nose radius of 0.6 mm and an interferometric 1D distance probe system. Table 6 contains the parameters used for the tool path generation, machining process and on-machine measuring process. The number of surface points 4E6 and the linear interpolation mode for the numeric tool path generation are chosen, because this setting results in small motion errors near the obtained minimum values for the surface z_1 with $b = 1$ while requiring less computation effort than with using 16E6 surface points (see Fig. 14). Since the surface z_1 with $b = 1$ has stronger curvatures than the tilted flat used for the corrective machining experiment, even smaller motion errors can be expected, which justifies the chosen parameters.

One corrective pass is carried out based on an on-machine surface form measurement using the optical distance probe. Figure 16 shows the form error results of the first (Fig. 16a; without correction) and second (Fig. 16b; with correction) machining pass. The peak-to-valley form error has been reduced from 1.195 mm to 0.467 mm, which corresponds to a reduction of 61%. An analysis of the initial form error (Fig. 16a) results in a cutting tool offset in X-direction superimposed by a temperature deflection in Z-direction being the main reasons for the pattern of the form error. The correction

Fig. 14 Resulting path execution motion errors (maximum value e_{Mmax} , mean value e_{Mmean} , standard deviation σ_M) calculated over one C-axis rotation at a radius of 88 mm of the machining path for the freeform surface z_1 . Errors depicted in function of the parameter conditions (No. of points for tool path calculation, scaling factor b , interpolation mode) used in Tables 3 and 4 and for two C-axis rotation velocities (20 rpm, 40 rpm)

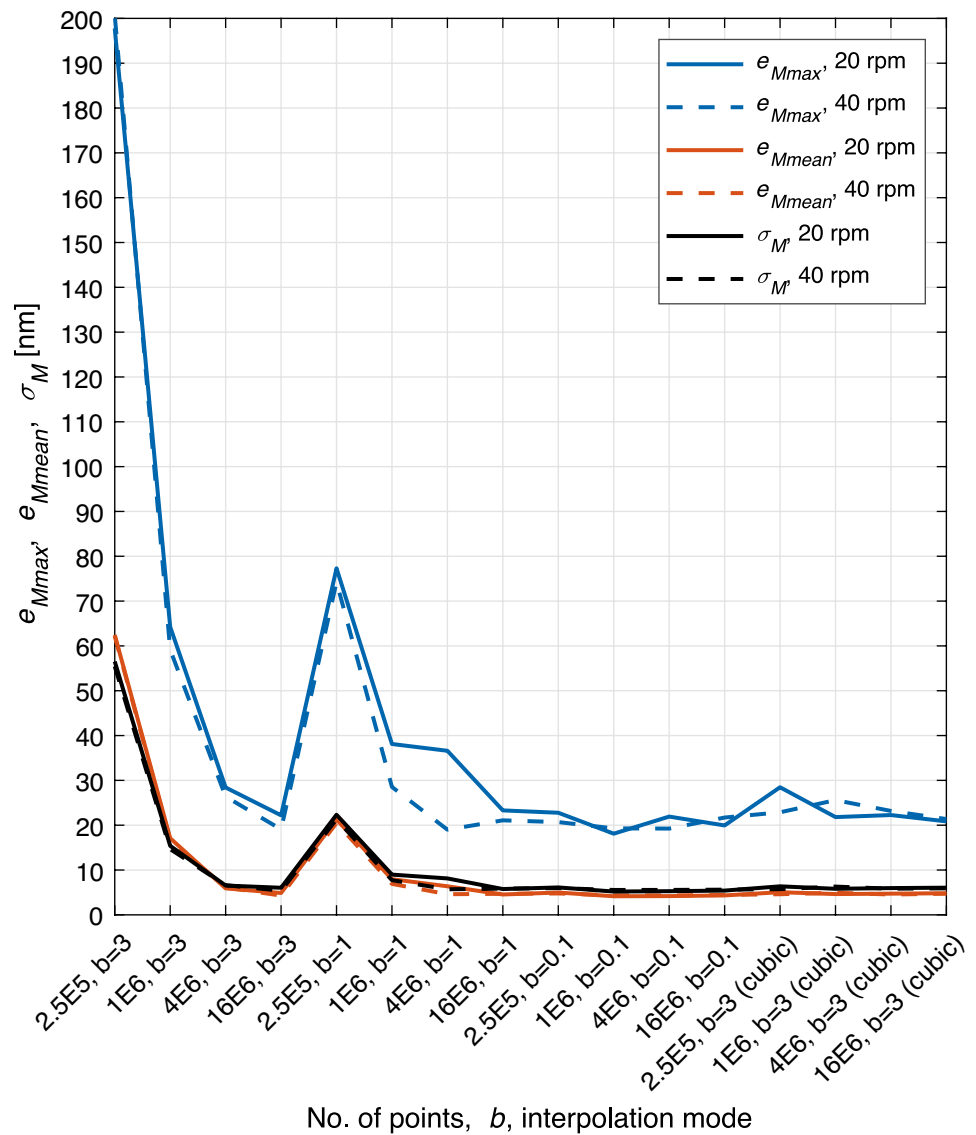


Table 6 Corrective machining experiment: used parameters for tool path generation, machining and on-machine measuring processes

Parameter	Value
Tool path generation	
No. of surface points	4E6
No. of neighbor points (for normals calculation)	5
Interpolation mode	Linear
Machining process	
Cutting depth	3 μm
Radial feed rate (stepover)	2.5 μm
C-axis rotations per minute	60 μm
Measuring process	
Radial feed rate (stepover)	20 μm
C-axis rotations per minute	19 rpm

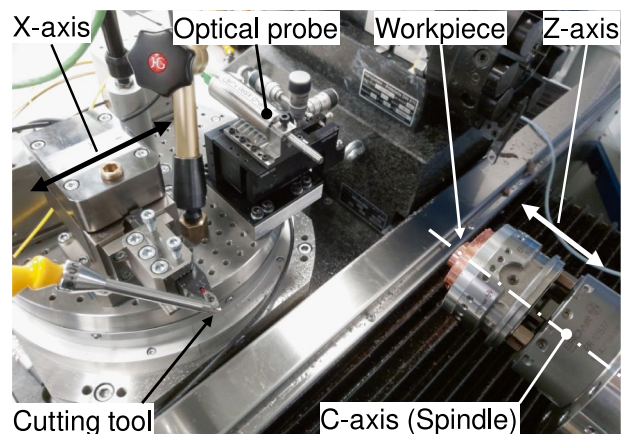


Fig. 15 Experimental setup for the corrective machining of the tilted flat

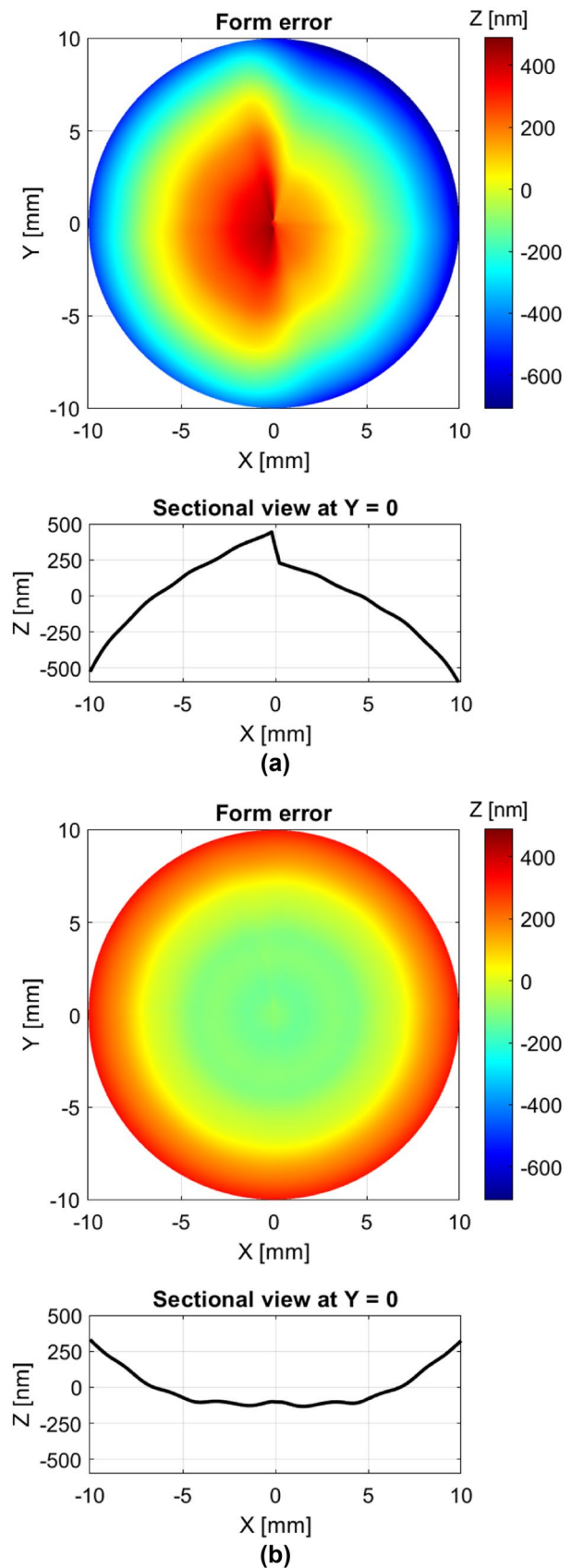
Fig. 16 Form error results of corrective machining a flat tilted by 5° in oxygen-free copper. (a) Form error for the initial machining pass; (b) form error after one corrective pass. Depicted are the form error in XY-view and the corresponding sectional view at $Y = 0$

of the characteristic step at the center (at $X = 0$; shown in the sectional view of Fig. 16a), which, as also stated by Preuss [19], is the result of a tool offset when machining a tilted flat, is impeccable. However, there seems to be a residual impact of a temperature deflection, which has not been part of the systematic errors influencing both machining passes in the same way. Besides, the surface’s roughness after the corrective pass of $S_a = 3.0nm$ (according to ISO 25178 and applying a Gaussian filter with cut-off wavelength of $80 \mu m$ is of typical order of magnitude for diamond turning.

Carried out investigations indicate that the outlet temperature of the cooling mist nozzle causes the temperature deflections during the machining passes by cooling down the workpiece when streaming against it during machining. Figure 17 depicts recorded data showing that the mist outlet temperature is about $0.3^\circ C$ lower than the ambient temperature. The dashed line part shows the temperature measured $20mm$ in front of the mist nozzle while the mist is turned off, whereas the solid line part while the mist is turned on like it is during machining.

Figure 18 shows the relations between a decreasing temperature during the machining process from the circumference to the center and the resulting workpiece surface profile. For the sake of simplicity both profiles are depicted as linear function. The initial machining result is shown in Fig. 18a, where a temperature decrease ΔT_1 leads to a positive surface form error ΔZ_{W1a} at the center, which is consistent with Fig. 16a. Supposing the same temperature profile ΔT_1 would occur during the corrective machining the surface form error profile would be compensated to the horizontal profile ΔZ_{W1b} . Because of slow temperature assimilations a more moderate decrease ΔT_2 during the corrective machining can be assumed. This together with the correction based on the profile ΔZ_{W1a} leads to the final surface profile ΔZ_2 , which corresponds to an over-correction.

The achieved form error reduction of 61% agrees with the reported values (between 57% and 61%) recently published by Tong et al. [20], who focus on closed-loop freeform machining. The peak-to-valley values presented there lie between $207nm$ and $384nm$ and are achieved over half of the diameter considered in the present work using a fast tool servo, implying a reduced machining duration compared to the present work. This, together with the identified thermal deflections, indicates potential for further improvements. Thus, a more detailed analysis of the process and the employed on-machine metrology setup including its calibration and an uncertainty analysis will be focused on in a subsequent publication.



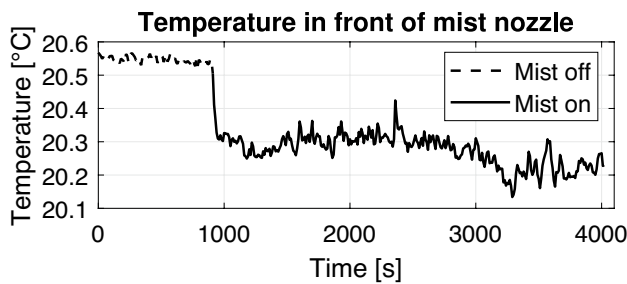


Fig. 17 Recording of ambient temperature 20 mm in front of the mist nozzle measured inside the machine tool housing. Dashed line: cooling mist off; solid line: cooling mist turned on. Total recording time is about 67 minutes or ≈, 4000s

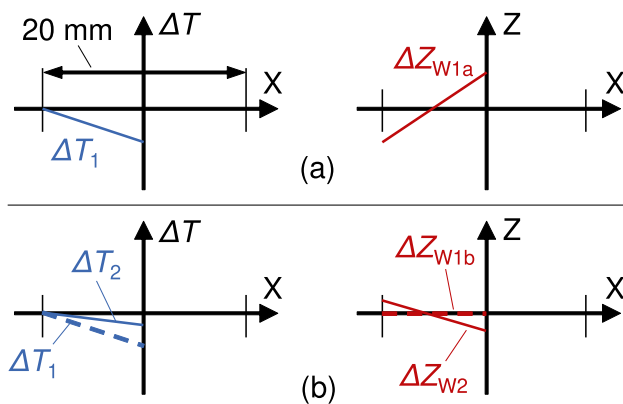


Fig. 18 Relations between a temperature change ΔT during machining from the circumference ($X = -10mm$) to the center ($X = 0mm$) and the resulting surface profile Z over X . Profiles simplified as linear functions. (a) initial machining: temperature profile ΔT_1 , surface profile ΔZ_{W1a} ; (b) corrective machining: ΔZ_{W1b} as result of ΔT_1 and over-corrected ΔZ_{W2} as result of ΔT_2

5 Conclusions and outlook

A new method for the computation of the cutting tool path in ultra-precision diamond turning, especially for three-dimensional freeform and corrective machining using form measurement data, is presented and analyzed. It allows the handling of surfaces defined by equations and by point clouds without using symbolic computation methods. Instead, numeric tools are used for the computation of the surface normal vectors as well as the surface on which the center point of the cutting tool’s nose radius moves. Together with interpolation techniques, the generation of a spiral tool path in case of the steady-X method ensuring a uniform axis motion in X-direction is provided. By a comprehensive analysis, the errors occurring throughout the computation steps are compared to the exact solutions for various geometries with diverse curvatures. Under realistic assumptions regarding

the curvature of the surface to be machined and reasonable refinement of the applied numeric methods the results show errors $< 1nm$. Combined with an investigation of the effect of non-exact tool path points on the behavior of an exemplary diamond turning machine this allows to neglect the accuracy loss compared to the exact tool path. Besides providing a tool path of sufficient accuracy, the method offers to update the point cloud representing the surface to be machined by surface form measurement data. Presented form error results of corrective machining a flat, which is tilted by 5° with regard to the XY-plane, showing a reduction of the peak-to-valley form error of 61% verify the ability of the proposed method for this application.

The results show the subtle interplay of grid, normal vectors and interpolation in the accuracy of the computed trajectory. While the role of interpolation is well known, additional points need to be addressed, e.g., the point distribution and the influence of noise in the computation of the normals. To the best of the authors’ knowledge, even in recent works, where point clouds are used, this is not investigated.

Appendix 1: Analysis of the plane fitting method

From the results depicted in Fig. 6, it is observed that for point symmetric stencils on a Cartesian grid, the angle error behaves a lot better than for other general stencils. This suggests that it is depending on the surface, but this is because the plane approximation for point symmetric stencils delivers a second-order approximation of the normal vector. The aim of this appendix is to prove this claim for a smooth surface, or at least a three times continuously differentiable surface. Without loss of generality, the following configuration for a small number $h > 0$ is considered

$$X = \begin{pmatrix} h & -h & 0 & 0 \\ 0 & 0 & h & -h \\ f(h, 0) & f(-h, 0) & f(0, h) & f(0, -h) \end{pmatrix} \tag{17}$$

under the assumption that

$$\frac{1}{4}(f(h, 0) + f(-h, 0) + f(0, h) + f(0, -h)) = 0. \tag{18}$$

Then it follows

$$A = X \cdot X^T = 2h^2 \begin{pmatrix} 1 & 0 & \Delta_{h,x}f(0, 0) \\ 0 & 1 & \Delta_{h,x}f(0, 0) \\ \Delta_{h,x}f(0, 0) & \Delta_{h,x}f(0, 0) & \frac{F}{2h^2} \end{pmatrix}, \tag{19}$$

where the missing entries are completed by symmetry. By letting

$$F := f(h, 0)^2 + f(-h, 0)^2 + f(0, h)^2 + f(0, -h)^2, \tag{20}$$

which is the variance of the function for this stencil (see Eq. (19)), the following notations for the central finite difference equations can be used

$$\Delta_{h,x}f(0,0) = \frac{f(h,0) - f(-h,0)}{2h} \quad (21)$$

$$\Delta_{h,y}f(0,0) = \frac{f(0,h) - f(0,-h)}{2h}. \quad (22)$$

Since f is differentiable and because of Eq. (19), the bound $F \leq Ch^2$ can be obtained. Thus, it is inferred that the third column of A is a linear combination of the first two columns up to an error proportional to h^2 . This implies the stated claim. Since every point symmetric stencil can be expressed as the union of several 4-points symmetric stencils, the claim is valid for all the point symmetric stencils with the error being a finite sum of second-order errors, one for every stencil.

Acknowledgements The authors wish to thank the referees for their insightful questions and constructive remarks, which helped improving the clarity of the paper.

Author contributions Marco Buhmann mainly planned and conducted the experiments. Erich Carelli and Christian Egger mainly were responsible for the programming work. Klaus Frick mainly helped in the design phase with expertise in numerical methods. All authors contributed text and figures and worked together for proofreading and structuring the submission

Funding information Open access funding provided by Swiss Federal Institute of Technology Zurich. This work is co-funded by the Swiss Innovation Agency (Grant 26200.1 PFNM-NM).

Availability of data and materials Data and materials are submitted as part of the manuscript.

Declarations

Research involving human and animal participants No human participants were involved.

Consent for publication All involved authors consent to publish this submission.

Ethical approval Ethical standards are in place (no human participants or animals involved).

Conflicts of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will

need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Brinksmeier E (2014) Diamond Machining, in: CIRP Encyclopedia of Production Engineering, Springer, Berlin, Heidelberg, pp. 390–395
2. Fang FZ, Zhang XD, Weckenmann A, Zhang GX, Evans C (2013) Manufacturing and measurement of freeform optics. CIRP Annals - Manufacturing Technology 62:823–846
3. Gao W, Haitjema H, Fang FZ, Leach RK, Cheung CF, Savio E, Linares JM (2019) On-machine and in-process surface metrology for precision manufacturing. CIRP Annals 68:843–866
4. Gong H, Ao S, Huang K, Wang Y, Yan C (2019) Tool path generation of ultra-precision diamond turning: A state-of-the-art review, Nanotechnology and Precision. Engineering 2:118–124
5. Brinksmeier E, Riemer O, Osmer J (2008) Tool path generation for ultra-precision machining of free-form surfaces. Production Engineering 2:241–246
6. Brecher C, Lange S, Merz M, Niehaus F, Wenzel C, Winterschladen M, Weck M (2006) NURBS Based Ultra-Precision Free-Form Machining. CIRP Annals 55:547–550
7. Fang FZ, D ZX, Hu XT (2008) Cylindrical coordinate machining of optical freeform surfaces, Optics express 16:7323–7329
8. Gong H, Fang FZ, Hu XT (2012) Accurate spiral tool path generation of ultraprecision three-axis turning for non-zero rake angle using symbolic computation, Springer 58:841–847
9. Scheiding S, Yi AY, Andreas G, Li L, Risse S, Eberhardt R, Tünnermann A (2011) Freeform manufacturing of a microoptical lens array on a steep curved substrate by use of a voice coil fast tool servo, Opt. Express 19:23938–23951
10. Dhanda M, Kukreja A, Pande S (2021) Adaptive spiral tool path generation for computer numerical control machining using point cloud. Proc IMechE Part C: J Mechanical Engineering Science
11. Masood A, Siddiqui R, Pinto M, Rehman H, Khan MA (2015) Tool path generation, for complex surface machining, using point cloud data. Procedia CIRP 26:397–402
12. Chui KL, Yu KM, Lee TC (2002) Direct tool-path generation from massive point input, in: Proceedings of the Institution of Mechanical Engineers, Part B: J Eng Manuf 216, IMECHE, pp. 199–206
13. Xiao W, Liu G, Zhao G (2021) Generating the tool path directly with point cloud for aero-engine blades repair. Proc IMechE Part B: J Engineering Manufacture 235:877–886
14. Hoppe H, DeRose T, Duchamp T, McDonald J, Werner S (1992) Surface reconstruction from unorganized points, pp. 71–78
15. Walker S (2014) Matlab class wrapper for a c++ implementation of a quadtree, <https://jp.mathworks.com/matlabcentral/fileexchange/45020-matlab-class-wrapper-for-a-c-implementation-of-a-quadtree>
16. Frisken SF, Perry RN (2002) Simple and efficient traversal methods for quadtrees and octrees. J Graph Tools 7:1–11
17. Ma J, Feng HY, Wang L (2013) Normal vector estimation for point clouds via local delaunay triangle mesh matching. Computer-Aided Design & Applications 10:399–411
18. Brenner SC, Scott LR (2002) The mathematical theory of finite element methods, volume 15 of Texts in applied mathematics Springer-Verlag, New York, Berlin, Heidelberg
19. Preuss W (2019) A guide to diamond machining. Shaker Verlag, Aachen, Berichte aus der Fertigungstechnik
20. Tong Z, Zhong W, Zeng W, Jiang X (2021) Closed-loop form error measurement and compensation for FTS freeform machining. CIRP Annals 70:455–458

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.