Citation for published version:
Mullineux, G, Hunt, M, Cripps, RJ & Cross, B 2016, Smooth Tool Motions Through Precision Poses. in I Horvath, J-P Pernot & Z Rusak (eds), Tools and Methods of Competitive Engineering. Delft University of Technology, pp. 551-562.

Publication date:
2016

Document Version
Peer reviewed version

Link to publication

University of Bath

Proceedings of TMCE 2016, May 9-13, 2016, Aix-en-Provence, France, edited by I. Horváth, J.-P Pernot,Z. Rusák.

SMOOTH TOOL MOTIONS THROUGH PRECISION POSES

Mat Hunt
Glen Mullineux
Department of Mechanical Engineering
University of Bath
M.Hunt@bath.ac.uk, G.Mullineux@bath.ac.uk

Robert J. Cripps
Ben Cross
School of Mechanical Engineering
University of Birmingham
R.Cripps@bham.ac.uk, B.Cross@bham.ac.uk

ABSTRACT

Multi-axis machining requires the ability to define and manipulate the free-form motion of the cutting tool. In particular, there is a need to fit a smooth motion through a number prescribed precision poses. One approach is to deal separately with the translational and rotational components. This leads to two formulations with differing parameterizations which need to be combined. This paper considers the use of geometric algebra as a means for handling translations and rotations together and so generating motions in a single form. It considers the research question: is it possible to generate a smooth tool motion in a single form to pass through a number of prescribed precision poses. The methodology is to extend the corresponding approach for free-form curves and then to compare the results obtained using a specific case study example with those given in the literature. It is found that a tool motion can be achieved in a single form and it is at least as good as that obtained by considering the aspects of the motion separately. The fitting problem requires a distance function to be established between precision poses. Problems arise if measures based on length and angle are combined since the units are incompatible. A new distance function is proposed and demonstrated which avoids these problems.

KEYWORDS

Five-axis machining, free-form motion, geometric algebra, precision pose

1. INTRODUCTION

Usually, in multi-axis machining, smooth motions are required. This is partly so that the motion of the tool with respect to the work piece is well behaved and the cutting operation is successful. It is also desirable that the motions of the various axes of the machine tool itself are smooth so that these can be better controlled. This requires the ability to represent and manipulate smooth free-form motions [27].

It is also important to be able to fit a motion through a number of precision poses, where a pose represents the position and orientation of the cutting tool. Such fitting is needed within the CAM software where the tool path is planned off-line. It is also needed within the controller of the machine tool so that the cutter is driven on a motion through the discrete poses provided by the NC instructions.

The equivalent problem for free-form curves has been well explored and has led to the widespread use of Bézier and B-spline parametric forms [25] and various techniques for curve fitting [13]. One approach to deal with the motion of a cutting tool is to treat separately its translational and rotational motions [10, 36]. Curve-based methods can deal with each aspect and this leads to separate parametric functions. However, the two parts then need to be combined into a single form. This is made difficult by the fact that their units (length and angle) are incompatible. Additionally, the two parameterizations are different, so some form of reparameterization is needed, possibly based on the arc length of the path traced out by the tool tip.

Recently, the advantages of using geometric algebra have been rediscovered. Several formulations are available [3, 4, 12] but all allow rigid-body transforms (translations and rotations) to be represented in a single form. By letting such transforms vary, free-form motions can be generated, and the standard types of motion required for machining can be generated [5]. Since the algebra allows transforms to be combined additively, the techniques of using Bézier and B-spline forms pass across from the ideas of free-form curves.

This paper investigates the use of geometric algebra representations in handling the problem of motion fitting for tool paths. The specific research question is whether it is possible to work with a single form of the required motion, rather than two parts representing separately the translational and rotational aspects. The methodology adopted is as follows: firstly to extend the approach used for free-form curves to one for fitting a free-form motion to a number of pre-defined precision poses; and secondly to apply the approach to a case study from the literature, comparing the results obtained in terms of the motions of the tool axes with those previously obtained [36]. The comparison is in terms of the jerk in the motions of the individual axes. What is found is that it is indeed possible to deal with the required motion in a single form, and the motion obtained for the case study has jerk values of the same order of magnitude as (and for some axes better than) those achieved when the translations and rotations are considered separately. The novelty of the work is that the usual approach to generating tool paths is to consider the two aspects separately with the consequent need to combine them later in the process. The significance is that the new approach is more holistic in that both aspects are dealt with concurrently. This removes the potential need for any modification in forming the subsequent combination thus making the computation more efficient and less prone to numerical errors.

Section 2 discusses the existing literature, and section 3 gives an overview of the general fitting problem. In particular it considers how B-spline functions can be used to deal with the translational and rotational aspects of a motion separately. Section 4 gives an overview of the use of geometric algebra and shows how the B-spline fitting technique for curves extends to motions where the rotations and translations are treated together. This requires a measure of the "distance" between consecutive precision poses. When the components are considered separately, measures based upon changes in distance and angle can be adopted, but it is not natural to combine these. Instead a new distance measure is proposed in section 5. This is based on the spiral motion generated by a pair of precision poses when considered in isolation to the others. Some examples are given and discussed in section 6. Included is a comparison with the results of generating a motion by treating its components separately. Finally conclusions are drawn.

2. LITERATURE SURVEY

Compared to 3-axis machining, 5-axis machining has a number of advantages including: improved smoothness of the finished surface, the ability to deal with intricate parts, and the greater opportunity to avoid problems such as gouging [16, 34]. However, a 5-axis NC program is more difficult to create since not only has the translation of the tool to be considered but also its orientation [26].

Suitable methods for dealing with free-form motions are required. Such methods need to be flexible so that the challenges of 5-axis machining can be handled. These include: the ability to solve the inverse kinematics problem [9, 30]; the need to be able to detect and avoid motions close to singularities [1]; and the need to be able to control the machine tool to follow a specified motion [9].

With 3-axis machining, there is a need to deal with translational motions along free-form curves. This has led to the wide-spread use of B-spline curves [13, 15, 20] and, more recently, NURBS curves [25, 31, 33].

Additionally 5-axis machining requires rotations of the tool to be handled. Rotations and translations can be considered as rigid-body transforms [32]. Such transforms can be represented in a number of ways [27]. One approach has been the use of 4×4 matrices together with homogeneous coordinates [30, 37]. Motions are then treated as free-form functions of these matrices [6, 23]. By analogy with the case of curves, smooth motions can be obtained by minimizing suitable functionals of the acceleration or jerk [2]. However, this requires derivatives to be formed in the space of the matrices and it is not clear how the properties they induce relate to the tool motion itself.

Following Shoemake's seminal work [29] on the use of Bézier combinations of quaternions for rotations about the origin, extensions have been introduced including the ideas of double and dual quaternions so that translations can be handled alongside rotations

[11, 35]. More recently there has been renewed interest in the use of geometric (Clifford) algebra [12] which provides a unified environment incorporating the various form of quaternions. Initial work approximated translations as rotations about distant axes so that transforms can be modelled as 4×4 orthogonal matrices [8]. Such approximation can be avoided in a number of ways: by introducing an additional basis vector in the conformal geometric algebra (CGA) approach [3, 7, 24]; by handling infinity symbolically ($\mathcal{G}_4$) [21, 22]; and by inverting the geometric representation so that vectors in the algebra correspond to planes rather then points [28].

Given these representations, various techniques for generating smooth motions have been proposed. These include: search method based on quaternions and related representations [14, 37, 38]; and algorithms concerning the control of the tool motion [18, 19].

Most of this work applied to tool motion has been with the use of matrices and quaternions. As it has been shown that geometric algebra is capable of generating typical motions for manufacture [5], part of the interest of this paper is in how further use might be made of this approach to take advantage of the benefits it offers in considering translational and rotational motion together.

Comparison is made with a case study in a recent paper [36] where the translations and rotations of the tool are considered as separate B-spline functions and demonstrated using a specific tool motion [10]. This method is shown to produce acceptable results and, in particular, the jerk in each of the axes of the machine tool is kept under control. However, there is a need eventually to combine the two motions and this needs careful reparameterization of one or both motions and this requires additional processing and may introduce numerical uncertainties.

This paper considers the question of whether it is possible to deal with the translations and rotations concurrently without compromising the quality of the tool motion that is generated.

3. MOTION FITTING

The interest is in representing a smooth free-form motion of a machine tool which passes through a number of prescribed poses. A *pose* is the result of applying a transform to the tool (defined in some reference coordinate system) to reach a particular position and orientation (rotation) in three-dimensional space. The means whereby poses can be defined and manipulated is discussed later (section 4). At this stage, it is assumed that poses can be combined to generate other poses.

The basic fitting problem is as follows, and is an extension of the similar problem for fitting a curve through a number of prescribed points, called precision points [13]. A number of *precision poses* are given: it is through these that the motion must pass. Suppose there are N+1 precision poses denoted by P_i for $0 \le i \le N$. Following [36, 37] a B-spline is used to define the motion. This is a parametric representation and a parameter value t_i needs to be associated with each precision pose P_i. There are several ways proposed for doing this [10], but most take the following form. Starting with t_0=0, define

$$t_i = t_{i-1} + \Delta(P_{i-1}, P_i) \quad \text{for } 1 \le i \le N \tag{1}$$

where $\Delta(A, B)$ is a measure of the "distance" between poses A and B. Means for defining this function are discussed later (section 5).

For curves, a B-spline curve is a parametric function which is a piecewise linear combination of a number of control points. The values of the parameter where the function moves from one piece to the next are the *knots*. The control points define the shape of the curve. In curve fitting, the precision points help to determine the control points, but they are not the same as these points. Similarly, a B-spline motion is a piecewise linear combination of a number of *control poses*: these are different from the precision poses.

The interpolated B-spline has degree d, uses N+1 control poses, and has m+1 knots k_i where m = N+d+1 and

$$k_i = 0 \quad \text{for } 0 \le i \le d$$

$$k_i = [\, t_{i-d} + + t_{i+d-1} + \ldots + t_{i-1} \,]/\mathrm{d} \quad \text{for } d+1 \le i \le N$$

$$k_i = t_N \quad \text{for } N+1 \le i \le m \,.$$

The control poses Q_i, $0 \le i \le N$, for the motion are obtained by solving the N+1 linear equations

$$P_i = \textstyle\sum_j \; N_{j,d}(t_i)\, Q_j \tag{2}$$

where the $N_{i,d}(t)$ are the B-spline basis functions [25]. The varying pose which describes the resultant motion is

$$S(t) = \textstyle\sum_i \; N_{i,d}(t)\, Q_i \quad \text{for } 0 \le t \le t_N \,. \tag{3}$$

The representation of a precision pose used in [36] is as a pair of vectors P_i = ($\mathbf{p}_i$, $\mathbf{u}_i$) which are respectively, the position vector of the tool tip, and a unit vector in the direction of the tool axis. This means that two B-splines are used to form the motion: one for the position of the tool tip, the other for the tool orientation.

Given this split, a suitable distance function for the positional B-spline is

$$\Delta_p(P_{i-1}, P_i) = \| \mathbf{p}_{i-1} - \mathbf{p}_i \|^\alpha \quad (4)$$

which is based on the cartesian distance between consecutive precision poses. The exponent α is taken to be 0.5 in [36] which is the *centripetal* method [13] commonly used for curves to avoid "bulging".

A corresponding distance function for the orientation B-spline is

$$\Delta_u(P_{i-1}, P_i) = [\cos^{-1} (\mathbf{u}_{i-1} \cdot \mathbf{u}_i)]^\alpha \quad (5)$$

based upon the angles between consecutive precision poses. Again the centripetal form is used in [36] where α is taken to be 0.5.

The disadvantage of treating the positions and orientations separately is that it leads to two B-splines with different parameters. To reconcile these, the approach in [36] is firstly to reparameterize the positional B-spline in terms of the arc length s of the path of the tool tip. The orientational B-spline is then reparameterized to achieve the coincident parameter values at the precision poses. Between each pair, the parameter is adjusted so that a Bézier form of ninth degree is followed which attempts to minimize the jerk in the orientation motion.

4. GEOMETRIC ALGEBRA

The aim here is to obtain an approach which treats the positional and orientational motions together. This means that a single representation of the complete motion can be obtained directly, as opposed to creating two splines and then having to combine them.

Use is made of geometric algebra which allows translations and rotations to the defined and manipulated within a single environment. There are a number of formulations of geometric algebra, including: conformal geometric algebra (CGA) [3], homogeneous geometric algebra [28], and $\mathcal{G}_4$ [4,21]. What is presented in this paper works equally well in any of these versions: it is $\mathcal{G}_4$ that is used here.

The algebra has four basis vectors e_0, e_1, e_2, e_3. The general vector, $We_0 + Xe_1 + Ye_2 + Ze_3$, is a linear combination of these and represents projectively the three-dimensional point with cartesian coordinates $(X/W, Y/W, Z/W)$.

A multiplication is defined by extending the basis to elements of the form e_σ where σ is any subset of the set of subscripts {0,1,2,3}. Then, for example, the product e_1e_2 is defined to be the basis element e_{12}. The multiplication is anticommutative so that, for example, $e_2e_1 = -e_1e_2 = -e_{12}$. The *reverse* of a basis element is obtained by reversing the order of its subscripts. It is denoted by an asterisk, so that, for example, $e_{12}^* = e_{21} = -e_{12}$. Further details of how the multiplication is set up are given in [4, 21].

The *grade* of a basis element is the number of its subscripts. This idea passes to a combination of basis elements if they all have the same grade. Thus vectors have grade 1. Linear combinations of basis elements of even grade are important: they form a sub-algebra. If S is such an even-grade element and p is a vector, then S^*pS can be shown also to be a vector [22].

The map sending p to S^*pS is a map of projective space to itself. It preserves lengths and angles and hence is a rigid-body transform [22]. Further any rigid-body transform can be generated in this way. For example, the even-grade element

$$R = \cos(\theta/2) + \sin(\theta/2)e_{12}$$

generates a rotation through angle θ about the z-axis.

The significant point here is that the transform generated by an even-grade element can be a rotation or a translation (or a combination) and the algebra provides an environment in which such transforms can be handled in the same way.

This means that a representation of the precision poses can be created. Suppose a precision pose, $(\mathbf{p},\mathbf{u})$, is given comprising a position vector $\mathbf{p} = (p_1, p_2, p_3)$ and a unit vector $\mathbf{u} = (u_1, u_2, u_3)$. The latter represents a rotation of the tool from its vertical reference position, along the unit vector (0,0,1), to $\mathbf{u}$. This can be regarded as being formed from two rotations: the first about the y-axis through angle $\varphi = \cos^{-1}(u_3)$; the second about the z-axis through angle $\theta = \tan^{-1}(u_2,u_1)$, where the arctangent function with two arguments finds $\tan^{-1}(u_2/u_1)$ taking appropriate account of the quadrant (as with the function `atan2` in programming languages). The even-grade elements generating these rotations are

$$R_y = \cos(\varphi/2) + \sin(\varphi/2)e_{31}$$

$$R_z = \cos(\theta/2) + \sin(\theta/2)e_{12} .$$

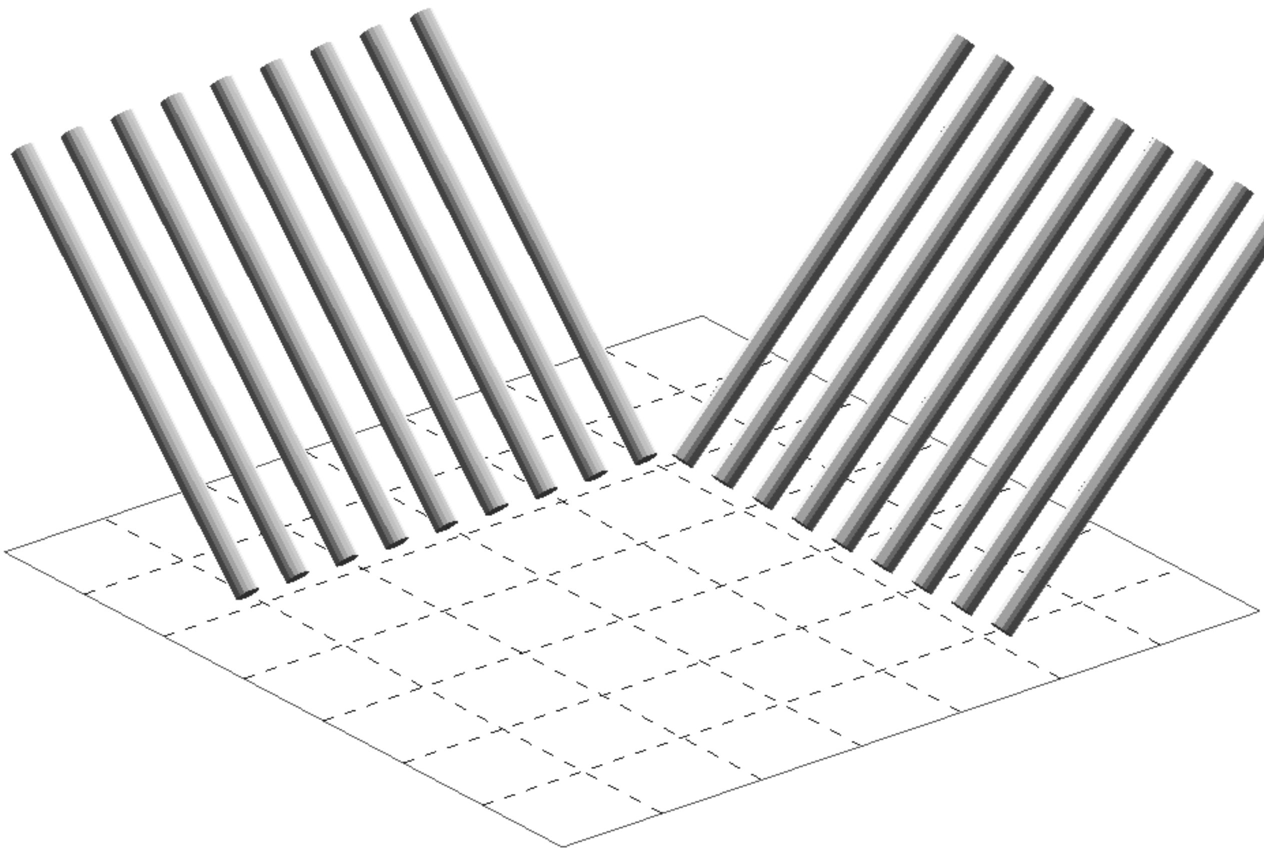

Figure 1 Precision poses for example 1

An even-grade element is required also for the translation part of the precision pose. Using the quantity ε introduced in [4,9], this element has the form

$$T = 1 + \varepsilon\, e_0\, (p_1 e_1 + p_2 e_2 + p_3 e_3)/2 .$$

Hence the precision pose is represented by the product of these even-grade elements which is

$$P = R_y\, R_z\, T .$$

In this way an even-grade element P_i, for $0 \le i \le N$, can be set up for each given precision pose. Then equation (2) is a set of linear simulation equations whose solution gives even-grade elements Q_i for $0 \le i \le N$. These can be combined, as in equation (3), to form the even-grade element $S(t)$. This generates a rigid-body transform and hence, as t varies, a smooth motion (of the tool).

Setting up the equations for the Q_i requires the distance function Δ to be specified. This is discussed in the next section.

5. DISTANCE FUNCTION

As noted in section 3, a great deal of attention has been paid in relation to curve fitting to how to select the function Δ to measure the distance between precision points. As a result, Δ_p, given by equation (4), with α=0.5, is commonly used and creates curves which are generally regarded as acceptable. This is the centripetal method. Similarly, a measure for distance between orientations is Δ_u, given by equation (5).

However, when dealing with translations and rotations together, the choice of function is much less clear. Simply using one of Δ_p or Δ_u can lead to difficulties. For example, Δ_p seems a poor choice if the tool motion has the tool tip becoming (roughly) stationary while the orientation changes significantly.

One compromise is a combination of the two previous functions

$\Delta(P_{i-1}, P_i) = w_p\Delta_p(P_{i-1}, P_i) + w_u\Delta_u(P_{i-1}, P_i)$

where w_p and w_u are suitably chosen real numbers. However, this involves a combination of quantities with different units (length and angle). The results it gives are unlikely to behave predictably if, say, the object to which the precision poses relate is doubled in size.

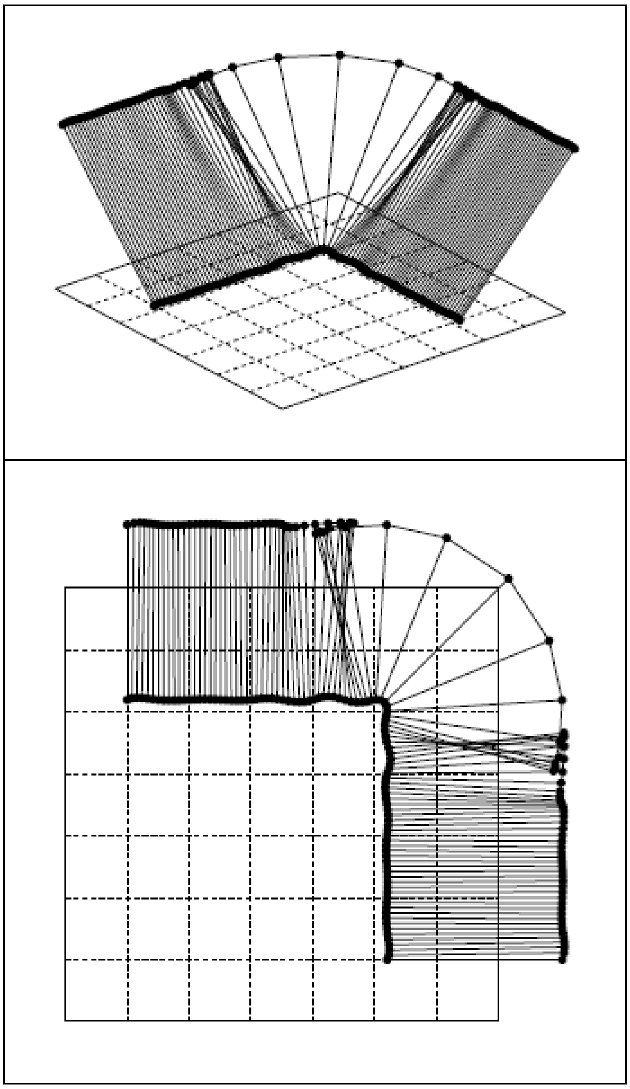

Figure 2 Motion using centripetal chord lengths

To overcome these problems, a new distance measure is proposed as follows. Suppose that A and B are two even-grade elements. These represent poses in three-dimensional space. Motions between these poses can be defined in two ways [4]. The first is a Bézier linear combination

$S(t) = (1-t)A + tB \quad \text{for} \quad 0 \leq t \leq 1$.

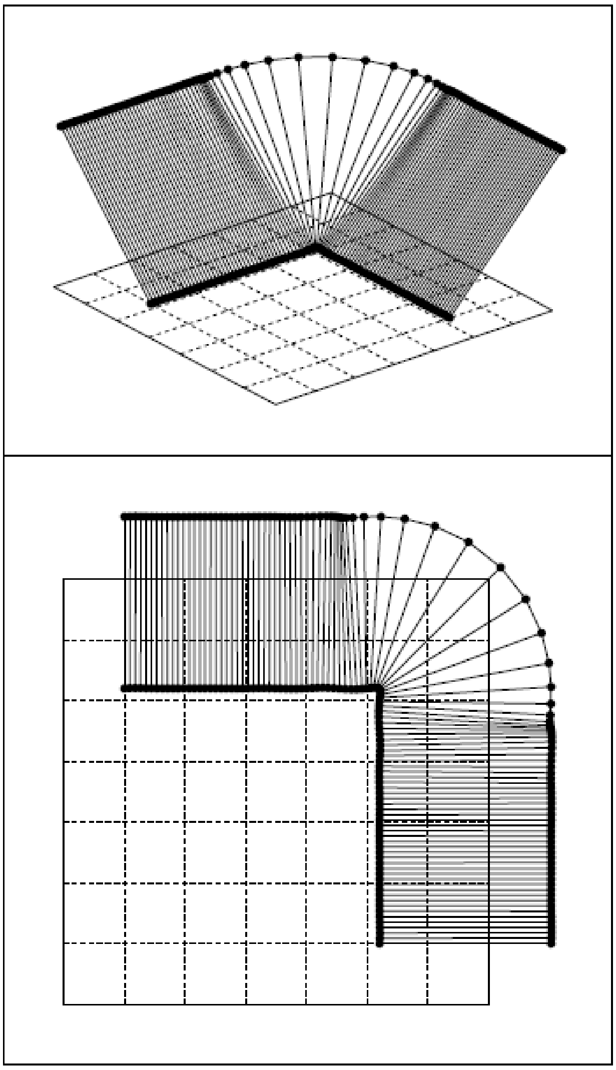

Figure 3 Motion using spiral distances

The second is a slerp (spherical linear interpolation), involving non-integer exponents [3,8] given by

$S(t) = A\,(A^* B)^t \quad \text{for} \quad 0 \leq t \leq 1$.

In both cases, any point in the tool being moved follows a spiral path that lies on a circular cylinder. In the first case, the path is on a planar slice; in the second case it is a true helix. As a result, there is, in fact, not much difference between these paths [4].

The proposed distance measure $\Delta(P_{i-1}, P_i)$ is the length of the spiral path traced out by a typical point in the cutting tool when simply moving from one pose to the other. (Note this is not the same motion as when the B-spline motion is eventually fitted.)

The most natural point to choose is the tool tip. However, this suffers from one of the difficulties identified above if the tip becomes stationary while the orientation changes. So instead, take Δ to be the sum of the lengths of the paths of the tip and the top of the tool. This has the advantage of being a value with a single unit (length) and has a physical meaning. It also gives invariant results under scaling, provided the tool itself undergoes the same scaling. In the examples that follow, it is the Bézier spiral motion that is used: this is the more straightforward to implement as it does not require the evaluation of expressions with non-integer exponents. The lengths of the spiral paths are found by evaluating the positions of the points for equally spaced values of the parameter t and summing the incremental distances moved.

6. EXAMPLES

Two examples are presented. The first is artificial and is constructed simply to illustrate the advantage of using the spiral distance measure.

The second is based on the example used in [36] which in turn in based on precision poses given in [10]. The tool path has a region of high curvature but since the precision poses are roughly regularly spaced in terms of distance and orientation some of the inherent problems are ameliorated. Following [36], the jerk in the motions of the axes of a five-axis machine tool are considered.

6.1. Example 1

This example uses 18 precision poses which are shown in figure 1. Half of these trace out a straight line with the tool leaning over at 45 degrees. The other half traverse a second straight line perpendicular to the first with the tool oriented at the same angle but in a different direction. Clearly these poses suggest an extreme discontinuity in the implied motion and for this reason such an example should not occur in practice. It is used here simply to illustrate what happens.

When the distance measure Δ_p is used centripetally, the result is that shown in figure 2: the tool is now represented by a line along its axis. The upper part of the figure shows an oblique view of the motion, and the lower part a plan view.

It is clear that the tool tip "wiggles" significantly during its motion. The orientations need to move the tool away from the corner as it is approached and then suddenly rotate around it.

Figure 3 shows the result of using the spiral distance measure. The "wiggles" in the motion of the tool tip are still present but are considerably reduced. While the "overshoot" of the orientations is still present, it too is reduced.

As noted, this is an extreme example. However it does suggest that in more realistic cases, the spiral distance measure is likely to provide better motions. It also illustrates that given such a large discontinuity, methods based on fitting any sort of B-spline motion through it are likely to behave poorly. The "wiggling" of the motion of the tool tip is unavoidable. An approach based on splitting the motion (at the discontinuity) and dealing with separate parts is always going to be more effective.

6.2. Example 2

This is the main example and, as the methodology given in section 1 notes, it is the one used to assess the apporach proposed in this paper. The required tool motion is that discussed in [36] which itself uses the precision poses specified in [10]. Note that the linear dimensions are halved in [36] and it is these revised values that are used here. The precision poses are shown in figure 4. The new method is applied to these precision poses to obtain a smooth tool motion which incorporates both the translational and rotational aspects. From this, the motions of the individual axes of the 5-axis machine tool are derived so that these can be compared with the result given in [36]. The comparison is in terms of the jerk in the motion of the individual axes.

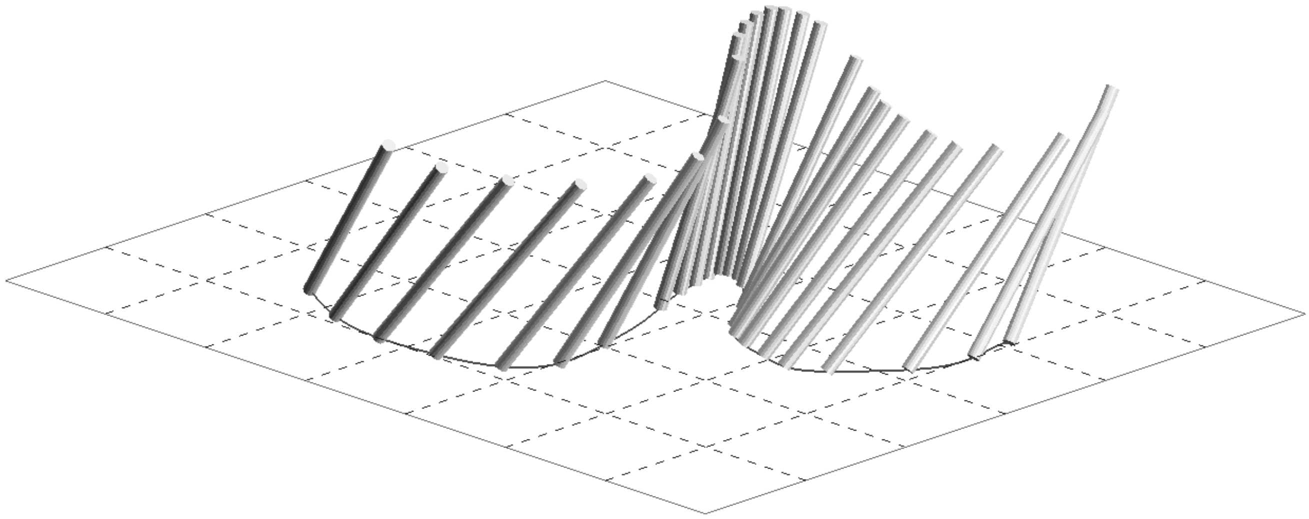

Figure 4 Precision poses for example 2 based on example from [10]

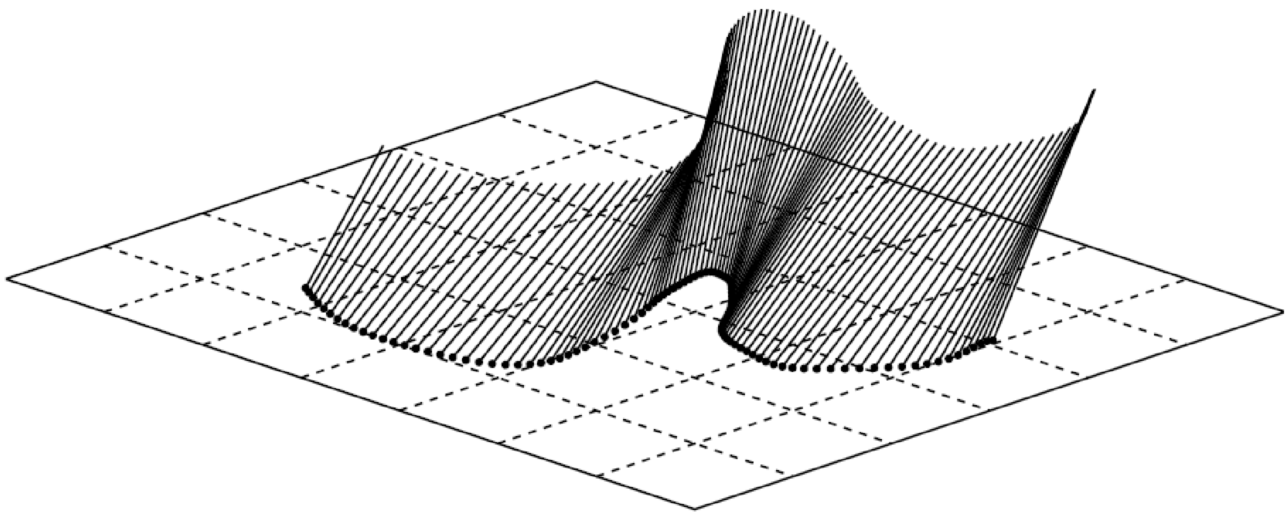

Figure 5 Motion obtained for example 2

The approach used in [36] is reviewed in section 3. It is an intricate four-stage process, using two fitting processes and two reparameterizations. During the latter, an attempt is made to reduce, via a minimization process, the jerk in the orientation motion. However, it is not clear that there are sufficient degrees of freedom for this to have a significant effect.

The approach used here is simply to fit a quintic B-spline through the precision poses. These are represented as even-grade elements from the geometric algebra so that the translational and rotational motion are handled concurrently. The spiral distance measure is used (centripetally) to establish the values of the parameter at each of the precision poses. Figure 5 shows the motion obtained with the tool represented by a line along its axis.

If the typical pose during the motion is represented by the position vector $\mathbf{p} = (p_1, p_2, p_3)$ and the unit vector $\mathbf{u} = (u_1, u_2, u_3)$ along the tools axis, then the positions of the axes of a five-axis machine tool can be determined. The relations used in [36] are as follows

$$A = \cos^{-1}(u_3)$$

$$C = \tan^{-1}(u_1, u_2)$$

$$X = -\cos(C)p_1 - \sin(C)p_2$$

$$Y = \cos(A)\sin(C)p_1 - \cos(A)\cos(C)p_2 - \sin(A)p_3 - \sin(A)a$$

$$Z = \sin(A)\sin(C)p_1 - \sin(A)\cos(C)p_2 + \cos(A)p_3 + \cos(A)a + b$$

where the machine dependent offsets are a = 70mm, and b = 150mm.

Use of these relations means that, at any point on the motion generated here, the values of the five machine parameters can be obtained. Figure 6 shows graphs of these and they compare well with those presented in [36]. These graphs show individually the motions of the five axes of the machine tool when creating the tool motion shown in figure 5.

The graphs of the motions of the axes are plotted against time and this is achieved as follows. The tool pose is found at a number of equally spaced values of the parameter (201 values are used for the graphs here). The position of the tool tip is found for each pose and the arc length s at each pose is determined (estimated) by adding chord lengths along the path. The tool tip is assumed to move at a constant speed of u = 50mm/s, and hence a time value at each pose is determined. This allows the graphs to be plotted and also acts as a simple means for reparameterizing the motion in terms of arc length.

Knowledge of the arc length allows the third derivatives, with respect to s, of the five machine parameters to be found numerically. These are then multiplied by u^3 to give the jerk with respect to time.

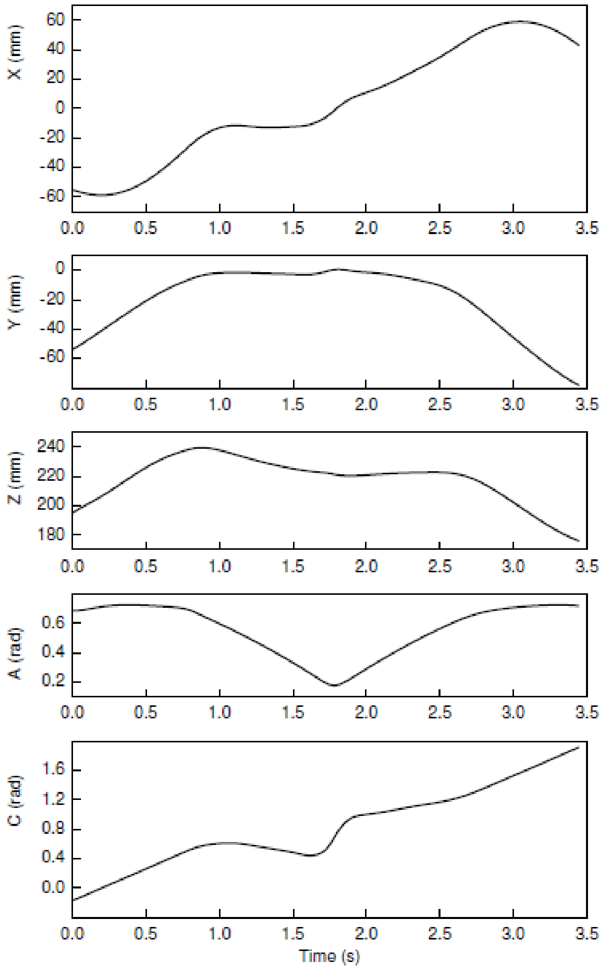

Figure 6 Five axes parameters for example 2

The numerical differentiation scheme combines five values: those at the current pose and those at the two on either side. (Of course, this is adjusted at the start and end of the motion.)

Figure 7 shows the graphs of jerk obtained here. Table 1 compares the minimum and maximum values of these with those in [36] (estimated from the graph in that paper).

The extrema for the jerk in X and C obtained here are clearly poorer (numerically larger) than those of [36]. But they are still of the same order of magnitude and indeed do not exceed twice the other values. Given the problems inherent in numerical differentiation, this seems to be entirely acceptable. What is surprising is that the extrema for the values of Y, Z, A are an improvement on those in [36]. This seems all the more surprising since no attempt is made here to minimize the jerk in the motion (beyond whatever is inherent in the use of the quintic B-spline).

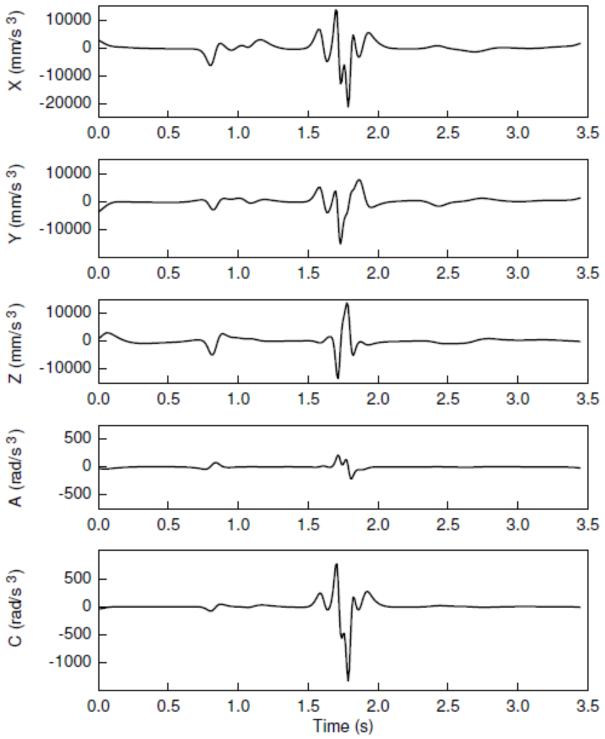

Figure 7 Jerk of parameters for example 2

It may be worth raising a further question. This is whether consideration of the jerk in the machine parameters represents a good way for assessing the quality of the motion of the cutting tool. Certainly a smooth motion of the machine tool is desirable but it is not clear that this necessarily reflects a good interaction between the tool and the work-piece. Conversely, if a smooth motion of the machine tool is what is required, then it would seem better to optimize the motion in the space of its five parameters, rather than in the space of the work-piece in the hope that this has the desired effect.

	Method presented in [14]		$\mathcal{G}_4$ and spiral distance	
X (mm/s^3)	−12000	7000	−21200	13700
Y (mm/s^3)	−30000	30000	−15300	7900
Z (mm/s^3)	−16000	14000	−13400	13800
A (rad/s^3)	−290	380	−209	215
C (rad/s^3)	−1000	500	−1330	770

Table 1 Comparison of maximum and minimum jerk values

7. CONCLUSIONS

The need to fit a free-form motion through a number of precision poses is one that arises in several application areas. These include the generation of smooth cutting tool movements in multi-axis machining. Here the fitting may be required to be undertaken off-line by the CAM software, or during machining itself by the machine tool controller.

The equivalent problem for free-form curves has been widely investigated and a number of standard techniques established. One approach to dealing with motions is to treat separately the translations and orientations and use curve-based techniques with each. This however requires the two separate components to be brought together into a single form.

This paper has investigated the use of geometric algebra as a means of representing translations and rotations in a single form. It has been seen that this allows a free-form motion (in B-spline form) to be obtained which passes through a number of prescribed precision poses. The technique has been applied to a case study example and the motion achieved for the individual axes exhibits jerk values comparable with those that appear in the literature.

Key to this approach is the ability to provide a “distance” function to allow parameter values to be assigned to the precision poses. There are inherent difficulties if linear and angular measures are simply combined. A new measure has been proposed and illustrated which considers the lengths of spiral paths traced out by points on a notional paths between consecutive pairs of poses. This provides a measure which is of a single form (length). When dealing with machining, natural points to choose are the ends of the tool axis.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Engineering and Physical Sciences Research Council(EPSRC) in funding a project entitled "Algebraic modelling of 5-axis tool path motions" (ref: EP/L006316/1 and EP/L010321/1). The collaboration, guidance and advice of Delcam International Plc is also gratefully acknowledged.

REFERENCES

[1] Affouard, A., Duc, E., Lartigue, C., Langeron, J.-M., and Bourdet, P., (2004), "Avoiding 5-axis singularities using tool path deformation", International Journal of Machine Tools & Manufacture, 44(4), 415–425.

[2] Belta, C., and Kumar, V., (2002), "Euclidean metrics for motion generation on $SE(3)$", Proceedings of the Institution of Mechanical Engineers: Part C, Journal of Mechanical Engineering Science, 216(1), 47–60.

[3] Cibura, C., and Dorst, L., (2010), "Determining conformal transformations in R^n from minimal correspondence data", Mathematical Methods in the Applied Sciences, 34(16), 2031–2046.

[4] Cripps, R. J., and Mullineux, G., (2014), "Design of free-form motions", in: Tools and Methods for Competitive Engineering, ed. Horváth, I. and Rusák, Z., University of Delft, 275–284.

[5] Cripps, R. J., and Mullineux, G., (2015), "Using geometric algebra to represent and interpolate tool poses", International Journal of Computer Integrated Manufacturing, to appear.

[6] Crouch, P., Kun, G., and Silva Leite, F., (1999), "The de Casteljau algorithm on Lie groups and spheres", Journal of Dynamical and Control Systems, 5(3), 397–429.

[7] Dorst, L., Fontijne, D., and Mann, S., (2007), Geometric Algebra for Computer Science: An Object-oriented Approach to Geometry, Morgan Kaufmann, Amsterdam.

[8] Etzel. K. R., and McCarthy, J. M., (1999), "Interpolation of spatial displacements using the Clifford algebra of $E4$", Transactions of the ASME: Journal of Mechanical Design, 121(1), 39–44.

[9] Farouki, R. T., Han, C. Y., and Li, S., (2014), "Inverse kinematics for optimal tool orientation control in 5-axis CNC machining", Computer Aided Geometric Design, 31(1), 13–26.

[10] Fleisig, R. V., and Spence, A. D., (2001), "A constant feed and reduced angular acceleration interpolation algorithm for multi-axis machining", Computer-Aided Design, 33(1), 1–15.

[11] Gan, D., Liao, Q., Wei, S., Dai, J. S., and Qiao, S., (2008), "Dual quaternion-based inverse kinematics of the general spatial 7R mechanism", Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 222(8) 1593–1598.

[12] González Calvet, R., (2007), Treatise of Plane Geometry through Geometric Algebra, TIMSAC, Cerdanyola del Vallès.

[13] Haron, H., Rehman, A., Adi, D. I. S., Lim, S. P., and Saba, T., (2012), "Parameterization method on B-spline curve", Mathematical Problems in Engineering, 2012, 640472:1–22.

[14] Ho, M. C., Hwang, Y. R., Hu, C. H., (2003), "Five-axis orientation smoothing using quaternion interpolation algorithm", International Journal of Machine Tool Manufacture, 43(12), 1259–1267.

[15] Langeron, J. M., Duc, E., Lartigue, C., and Bourdet, P., (2004), "A new format for 5-axis tool path computation, using B-spline curves", Computer-Aided Design, 36(12), 1219–1229.

[16] Lasemi, A., Xue, P., and Gu, P., (2010), "Recent developments in CNC machining of freeform surfaces: a state-of-the-art review", Computer-Aided Design, 42(7) 641–654.

[17] Leeney, M., (2009), "Fast quaternion slerp", International Journal of Computer Mathematics, 86(1), 79–84.

[18] Li, J., Zhouyang, H., and Lou, Y., (2013), "Tool path optimization in postprocessor of five-axis machine tools", International Journal of Advanced Manufacturing Technology, 68(9{12), 2683–2691.

[19] Li, J.-G., Qiu, M.-M., Zhang, T.-H., and Li., Z.-X., (2012), "A practical real-time non-uniform rational B-spline curve interpolator for computer numerical control machining", Proceeding of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 226(C4) 1068–1083.

[20] Li, M., Liu, X., Jia, D., and Liang, Q., (2015), "Interpolation using non-uniform rational B-spline for the smooth milling of ruled-surface impeller blades", Proceedings of the Institution of

Mechanical Engineers: Part B, Journal of Engineering Manufacture, 229(7), 1118–1130.

[21] Mullineux, G., (2004), “Modeling spatial displacements using Clifford algebra”, Transactions of the AMSE: Journal of Mechanical Design, 126(3), 420–424.

[22] Mullineux, G., and Simpson, L., (2011), “Rigid-body transforms using symbolic infinitesimals”, in: Guide to Geometric Algebra in Practice, ed. Dorst, L. and Lasenby, J., Springer-Verlag, London, 353–369.

[23] Nava-Yazdania, E., and Polthier, K., (2013), “De Casteljau’s algorithm on manifolds”, Computer Aided Geometric Design, 30(7), 722–732.

[24] Perwas, C., (2009), Geometric Algebra with Applications in Engineering, Springer, Berlin.

[25] Piegl, L. and Teller, W., (1997), The NURBS Book, 2nd edition, Springer-Verlag, Berlin.

[26] Rao, N., Ismali, F., and Bedi, S., (2000), “Integrated tool positioning and tool path planning for five-axis machining of sculptured surfaces”, International Journal of Production Research, 38(12), 2709–2724.

[27] Röschel, O., (1998), “Rational motion design – a survey”, Computer-Aided Design, 30(3), 169–178.

[28] Selig, J. M., (2000), “Clifford algebra of points, lines and planes”, Robotica, 18(5), 545–556.

[29] Shoemake, K., (1985), “Animating rotation with quaternion curves”, ACM SIGGRAPH, 19(3) 245–254.

[30] Sørby, K., (2007), “Inverse kinematics of five-axis machines near singular configurations”, International Journal of Machine Tools & Manufacture, 47(2), 299–306.

[31] Tikhon, M., Tae, J. K., and Lee, S. H., (2004), “NURBS interpolator for constant material removal rate in open NC machine tools", International Journal of Advanced Manufacturing Technology, 44, 237–245.

[32] Ting K.-L., and Zhang, Y., (2004), “Rigid body motion characteristics and unified instantaneous motion representation of points, lines, and planes”, Transactions of the ASME: Journal of Mechanical Design, 126(4) 593–601.

[33] Tsai, M. C., Cheng, C. W., and Cheng, M. C., (2003), “A real-time NURBS interpolator for precision three-axis CNC machining”, International Journal of Machine Tool Manufacture, 43(12), 1217–1227.

[34] Wang, Y. J., Dong, Z., and Vickers, G. W., (2007), “A 3D curvature gouge detection and elimination method for 5-axis CNC milling of curved surfaces”, International Journal of Advanced Manufacturing Technology, 33(3–4), 368–378.

[35] Wu, W., and You, Z., (2010), “Modelling rigid origami with quaternions and dual quaternions”, Proceedings of the Royal Society, A, 466(2119) 2155–2174.

[36] Yuen, A., Zhang, K., and Altintas, Y., (2013), “Smooth trajectory generation for five-axis machine tools”, International Journal of Machine Tools & Manufacture, 71, 11–19.

[37] Yang, J., and Altintas, Y., (2013), “Generalized kinematics of five-axis serial machines with non-singular tool path generation”, International Journal of Machine Tools & Manufacture, 75, 119–132.

[38] Zheng, G., Bi, Q.-Z., and Zhu, L.-M., (2012), “Smooth tool path generation for five-axis flank milling using multi-objective programming”, Proceedings of the Institution of Mechanical Engineers: Part B, Journal of Engineering Manufacture, 226(2), 247–254.