On the Design and Analysis of Freeform Motions Using Subdivision Schemes

A Dissertation Presented

by

Carlos Andrés Trujillo Suárez

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Mechanical Engineering

Stony Brook University

May 2009

Stony Brook University

The Graduate School

Carlos Andrés Trujillo Suárez

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Qiaode Jeffrey Ge, Dissertation Advisor
Professor, Department of Mechanical Engineering

Yu Zhou, Chairperson of Defense
Assistant Professor, Department of Mechanical Engineering

Peisen Huang
Professor, Department of Mechanical Engineering

Hong Qin
Professor, Computer Science Department

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

On the Design and Analysis of Freeform Motions Using Subdivision Schemes

by

Carlos Andrés Trujillo Suárez

Doctor of Philosophy

in

Mechanical Engineering

Stony Brook University

2009

This dissertation aims to develop new approaches for motion design and analysis suitable for CAD-CAM integration. Therefore, not only shape but also kinematic information should be conveyed in an efficient manner that can be easily implemented in CAD-CAM systems and eventually interpreted by CNC manufacturing equipment.

Dual quaternions, quaternions, and planar quaternions are used to represent spatial, spherical, and planar displacements, respectively. In this way, displacements of a rigid body in Cartesian space are mapped into points in the image space of displacements, transforming the kinematic problem of motion interpolation into a geometric problem where the powerful techniques for curve and surface interpolation from Computer Aided Geometric Design

(CAGD) can be readily applied.

Motivated by this consideration, initially in this dissertation a set of orientations with angular velocity constraints are interpolated by means of quaternion biarcs. The resulting quaternion curve represents a piecewise spherical line-symmetric rational motion with C^1 continuity which is used for tool path generation in 5-axis machining. Next, the four point interpolatory subdivision scheme for curve generation is adapted to the interpolation of a given set of positions of a cylindrical tool represented by dual quaternions. It is shown that the resulting discrete model of the tool path lends itself naturally to an algorithm for computing the characteristic curve belonging to the boundary surface of the swept volume at each of the discrete positions as well as the contribution from the top and bottom planes, and circular edges of the cylinder. Then, the dual tensor-product extension of such subdivision scheme is also used for two-parameter motion generation and swept surface computation.

A preliminary approach for mechanism simulation using subdivision schemes is also attempted by analyzing the motion of the coupler link of a planar 4R closed kinematic chain from the viewpoint of constrained motion interpolation. In the early stage of this refinement process, each new in-between position must be made to satisfy the 4R kinematic constraints exactly to ensure the correct motion. When there is sufficient number of coupler positions, one can use the unconstrained four-point interpolatory scheme to generate the in-between positions to allow for fast animation of the coupler motion.

The results obtained have applications in CNC tool path generation, robot path planning, and computer animation.

To My Wife, My Daughter, My Mother, and the Memory of My Father.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor Qiaode Jeffrey Ge, my research advisor, for his knowledgable guidance and encouragement throughout the work on this dissertation. His commitment to high academic standards and innovative thinking have set the example for my future professional life.

I am very thankful to the members of my defense committee, Professor Yu Zhou, Professor Peisen Huang, and Professor Hong Qin for their review of this manuscript.

My appreciation also go to my research mates Jun Wu, Dr. Zhe Jin, and Professor Anurag Purwar for their useful suggestions and friendship.

The diligent work of the staff at LASPAU especially Ms. Lisa Mallozi Tapiero solving my inquiries and processing my immigration paperwork is highly recognized.

I must not forget the pertinent and prompt assistance of the Graduate Program Secretary Ms. Diane Van Tronk on academic and administrative procedures and her kindness and sense of humor whenever I stopped by her office just to chat.

The financial support of this work under the grant DMI-0500064 of the National Science Foundation is gratefully acknowledged as well as the support of the Fulbright-Colciencias program, Colfuturo, and the leave of absence from the Universidad de Antioquia that allowed me to pursue doctoral studies and return to my teaching and research activities in Colombia.

Last but not least, all my love to my wife Margarita Rosa for her unselfish endurance, understanding, and devotion to our lovely daughter Isabela. She

has done an outstanding job at providing a warm family while I concentrated in my research.

The text of this manuscript in part is a reprint of the materials as it appears in [1], [2], and [3]. The co-author listed in the publications directed and supervised the research that forms the basis for this dissertation.

Chapter 1

Introduction

Current computer-aided manufacturing (CAM) processes such as Computer Numerical Controlled (CNC) machining and robotic welding require efficient methods not only for shape representation but also for motion generation. Traditionally, Computer Aided Geometric Design (CAGD) research has focused on the development of tools for shape modeling. CAGD is a very developed and mature field that encompasses a broad variety of well proved and established techniques for shape representation compiled in a wealth of reference textbooks, see for instance Farin [4, 5] and Piegl [6] among others. Due to their relative easiness for computational implementation and numerical robustness, CAGD techniques have even become industrial standards for shape representation in Computer Aided Design (CAD) software packages. Because of their flexibility, geometric intuitiveness, and ability to represent analytic and free form shapes, the so-called Non-Uniform Rational B-Splines (NURBS) [7, 8] are the widespread method for shape design.

However, when it comes to manufacturing and motion planning, the parametric representation of the shape, either by means of NURBS or any other method, is discretized and then locally described by a first order approximation [9]. This first order approximation of the shape is commonly used for collision avoidance in CNC milling either by lifting the cutter in the direction of the tool axis or by changing the orientation of the tool by tilting it about the cutter contact (CC) point as described by Klass and Schramm [10]. The latter approach is only realizable in 5-axis milling or in a restricted fashion in 4-axis milling. Based on the collision check a satisfactory set of CC points and tool orientations is generated for all of the discrete positions of the cutter. In order to manufacture the actual piece, a tool path that passes through these discrete positions is created. The most common approach to achieve the translational motion of the tool is by linear or circular interpolation from one position to the next one although parabolic and cubic interpolation are available in special controllers as mentioned by Thyer [11]. Generally, the orientation of the tool is kept fixed when the machine is moving the cutter from one position to the next. More sophisticated approaches assume a linear change of orientation between two consecutive positions as in Sarma and Rao [12], or even combine linear and circular interpolation as described by Liang et al. [13], or aim to keep the tool axis perpendicular to the surface as proposed by Koren and Lin [14]. In any case, the geometric information pertaining to the piece is decoupled from the actual motion of the tool, i. e. the geometry and the kinematics required to manufacture the piece are two different entities which are usually dealt with in different software applications. Therefore, the link between CAD and CAM remains weak.

Motivated by this last issue, researchers started to look into the blend of CAGD techniques and motion design. To this end, it was realized that the elegant representation of spatial displacements by means of dual quaternions was suitable in order to extend the powerful CAGD methods to the realm of motion generation. Shoemake was arguably the first in using quaternions for animation of rotation in Computer Graphics; he introduced the spherical linear interpolation (Slerp) in order to generate interpolating orientations in between two quaternions. Then, he used spherical Bézier curves to obtain piecewise Bézier rotational (spherical) motions [15]. However, Slerp only generates C^0 continuity, i.e. the motions lack velocity and higher order continuity. Wang and Joe [16] interpolated orientations by spherical biarcs invoking the fact that unit quaternions lie on a hypersphere. Barr et al. [17] developed a method for orientation interpolation with unit quaternion curves that minimize tangential acceleration. Ramamoorthi and Barr [18] implemented cubic splines and the minimization of the Euler-Lagrange error functional for fast construction of unit quaternion splines.

In order to fit the current standard of shape representation by rational curves and surfaces employed in CAD an important effort has been devoted to the extension of these methods to motion generation. Jüttler extended Bézier curves to handle dual quaternions; however, he interpolated the real and the dual part of the dual quaternions separately [19]. Jüttler and Wagner [20] also explored the rational motion generation problem using NURBS and matrix representation of spatial displacements but this approach is cumbersome from a computational point of view when compared to the use of dual quaternions. Ge and Ravani studied the application of the de Castelajau algorithm to handle dual quaternions and offered an alternative version of it that preserves the subdivision properties of regular Bernstein-Bézier curves [21]. They also introduced the concept of *orientable image space* in order to avoid ambiguities in the interpolation of dual quaternions and developed methods for designing spline curves in the image space with tangent, curvature and, torsion continuities [22]. The interested reader is referred to the survey prepared by Röschel [23] for a comprehensive account on rational motion design and specific results.

Even though rational curves and surfaces are the most common tool for shape representation in current CAD systems, there is a growing trend among the computer graphics community and solid modeling industry towards replacing rational representation by subdivision schemes. Actually, large companies in animation production have opted for subdivision schemes as their preferred method for shape representation [24]. This fact stems from their ability to represent shapes of arbitrary topology, more localized control than rational representations, computational efficiency, and numerical robustness given their simple linear refinement rules. Chaikin [25] introduced subdivision as an efficient technique for curve generation; he utilized fixed ratios on cutting off the corners of a polygon. The application of subdivision in Computer Aided Geometric Design (CAGD) became a reality when the extension of curve generation schemes to tensor-product rules was achieved. Catmull and Clark [26] generalized bi-cubic uniform B-splines. Doo and Sabin [27] used the analytical expression of the bi-quadratic uniform B-spline surface to generate a subdivision scheme that produces C^1 limit surfaces with arbitrary topology out of arbitrary initial meshes. These subdivision procedures are of approximating nature. Dyn et al. [28] developed an interpolating butterfly scheme which is the tensor-product extension of the four-point interpolatory scheme for curve generation [29].

Subdivision has reached a mature level in CAGD literature and many other schemes along with their convergence analysis and characteristic features have been explored, see for instance Warren and Waimer [30]. Some researchers have turned their attention to such schemes for the design of rigid body motions. Hofer et al. [31, 32] devised algorithms based on homologous points. In such approach, a set of key points on a rigid body are chosen to be interpolated along a set of discrete positions of the body by variational subdivision; registration techniques from computer vision need to be applied on the interpolating points to correct deformations of the body during the motion. Wallner and Pottmann [33] apply subdivision on the 3×3 orthogonal rotation matrix and the translation vector of a set of spatial displacements considering the terms of the matrix and the vector for each position as a 12-dimensional vector. Again, since this process does not guarantee that the interpolating components corresponding to the interpolating matrix form an orthogonal matrix, a correction procedure based on singular value decomposition of the matrix is required. In this work, Wallner and Pottmann also claim that small perturbations to subdivision schemes do not alter their smoothness.

This dissertation aims to develop new approaches for motion design and analysis suitable for CAD-CAM integration. Therefore, not only shape but also kinematic information should be conveyed in an efficient manner that can be easily implemented in CAD-CAM systems and eventually interpreted by CNC manufacturing equipment. The results presented in this manuscript are part of the kinematics-driven geometric modeling framework introduced by Ge [34] in which techniques to integrate sculptured-surface design with tool-path generation for 5-axis CNC machining are studied.

It should be pointed out that even though the aforementioned methods and the ones here discussed can be readily implemented in CAD-CAM systems, their actual use in motion control of CNC machines would only be achieved in open architecture controllers since current CNC controllers are vendor-specific standardized and hardware centered. Thus, our approaches are more suitable for PC-based control where the PC-based system directly generates the reference commands for the axes and the spindle. A well documented overview on open architecture controllers was presented by Pritschow et al. [35]. Also, Rober and Shin [36] demonstrated the feasibility of PC-based open architecture computer numerical control of milling machines. Nonetheless, the techniques hereby developed are also useful alternatives to the traditional display of CNC machining using Boolean substraction of solid objects introduced by Van Hook [37] or to the G-buffer method used by Saito and Takahashi [38].

The organization of this document is as follows. The representation of spatial and spherical displacements by means of dual quaternions and quaternions is reviewed in chapter 2. In chapter 3 a method for orientation interpolation with angular velocity constraints using piecewise quaternion biarcs is developed. Such a technique can be readily used for tool path generation in 5-axis milling of sculptured surfaces. Chapter 4 elaborates on the extension of the four-point interpolatory subdivision scheme for curve generation to the realm of motion interpolation and in the exploitation of its geometric characteristics for swept volume approximation of a one parameter motion of a cylindrical cutter. Chapter 5 uses the tensor-product version of the four-point scheme to generate the swept surface of the two-parameter motion of the cylinder. A preliminary exploration of mechanism simulation and animation by means of subdivision schemes is presented in chapter 6. Finally, concluding remarks on the main contributions of this research and an account of possible future work is listed in chapter 7.

Chapter 2

Kinematics Background

In the following sections, a description of quaternions and dual quaternions as displacement operators is presented in the extent that pertains to this dissertation. Then the motion of points, planes, and lines affected by those displacement operators is reviewed. Finally, one- and two-parameter rational Bézier motions are explained along with the computation of the swept surface of a cylinder undergoing such motions.

2.1 Representation of Spatial Displacements

In 1843, Irish mathematician and astronomer William R. Hamilton discovered quaternions as an algebraic extension of complex numbers to four dimensional space and realized they were useful to represent the orientation of a rigid body [39, 40, 41]. The four components of this special type of quaternions are the Euler-Rodrigues parameters of rotations, commonly referred to as Euler parameters (see Bottema and Roth [42] and McCarthy [43]). In 1873, William K. Clifford devised biquaternions, now called dual quaternions, to represent a general spatial displacement of a rigid body encompassing orientation and translation [44]. The primary part of the dual quaternion is the quaternion made up by the Euler-Rodrigues parameters; the dual part is the quaternion product of the translation vector and the primary part of the dual quaternion. A delightful account of the historical development of quaternions, dual quaternions, and their intrinsic relationship with other representations of rigid body displacements can be found in Dai [45].

In this dissertation, dual quaternions and quaternions, as a particular case of dual quaternions, are adopted for the representation of spatial and spherical displacements, respectively. This choice is made based on their concise description of displacements and adequacy for computational implementation when compared to the traditional matrix representation of displacements.

2.1.1 Quaternions

A quaternion $\mathbf{q} = q_1\mathbf{i}+q_2\mathbf{j}+q_3\mathbf{k}+q_4$ representing a rigid body rotation by angle θ about the unit axis vector $\mathbf{s} = (s_1, s_2, s_3)$ has as components the so-called Euler-Rodrigues parameters of the rotation

$$q_1 = s_1 \sin\frac{\theta}{2}, \;\; q_2 = s_2 \sin\frac{\theta}{2}, \;\; q_3 = s_3 \sin\frac{\theta}{2}, \;\; q_4 = \cos\frac{\theta}{2} \tag{2.1}$$

which satisfy the relation

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{2.2}$$

Thus, this type of quaternions are commonly referred to as unit quaternions.

The quaternion basis units $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ satisfy the following fundamental multiplication rules

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{2.3}$$

The corresponding rotation matrix $[R]$ can be recovered from the Euler-Rodrigues parameters by using (see [42])

$$[R] = \frac{1}{S^2}\begin{bmatrix} q_4^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_4q_3) & 2(q_1q_3+q_4q_2) \\ 2(q_1q_2+q_4q_3) & q_4^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_4q_1) \\ 2(q_1q_3-q_4q_2) & 2(q_2q_3+q_4q_1) & q_4^2+q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{2.4}$$

where $S^2 = q_1^2+q_2^2+q_3^2+q_4^2$.

From the above, it is clear that the rotation matrix $[R]$ remains the same after multiplying each one of the Euler-Rodrigues parameters by a scalar $w \neq 0$. Therefore, a unit quaternion $\mathbf{q}$ and a multiple of it $\mathbf{Q} = w\mathbf{q} = (Q_1, Q_2, Q_3, Q_4)$, $w > 0$, represent one and the same rotation. Hence, quaternions can be thought of as points on a projective three-space for the representation of rotations. Such a space has been termed as the image space of spherical displacements by Ravani and Roth [46].

The homogeneous coordinates $\mathbf{X} = (X_1, X_2, X_3, X_4)$ of a cartesian point $\mathbf{x} = (x_1, x_2, x_3, 1)$ after a rotation represented by a nonunit quaternion $\mathbf{Q}$ are related by

$$\mathbf{X} = \mathbf{QxQ^{-1}} \tag{2.5}$$

being $\mathbf{Q^{-1}}$ the inverse of $\mathbf{Q}$ which is given by

$$\mathbf{Q^{-1}} = \frac{\mathbf{Q}^*}{N(\mathbf{Q})} \tag{2.6}$$

where $\mathbf{Q}^* = (-Q_1, -Q_2, -Q_3, Q_4)$ is the conjugate of $\mathbf{Q}$ and $N(\mathbf{Q}) = \mathbf{QQ}^*$ is the norm of $\mathbf{Q}$. The cartesian coordinates of the point after the rotation can be obtained by dividing $\mathbf{X}$ by the homogenizing component X_4.

Obviously, if $\mathbf{Q}(t)$ is a function of time t in the image space then Eq. 2.5 describes the spherical motion of a point. The time derivative $\dot{\mathbf{q}}$ of a unit

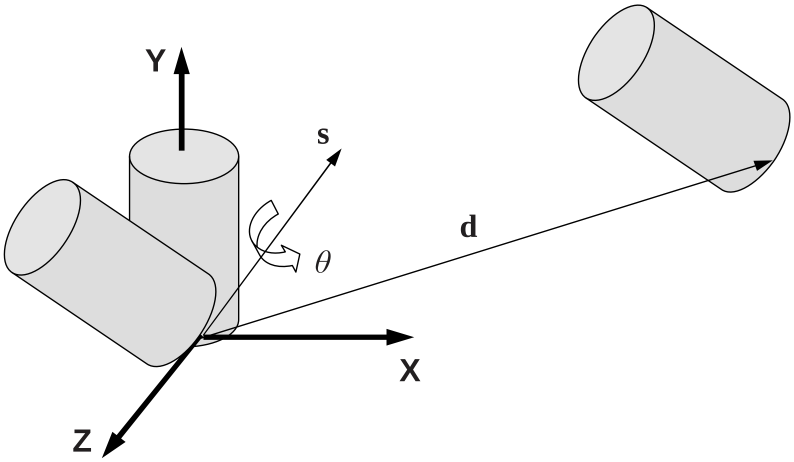

Figure 2.1: A spatial displacement consisting of a rotation by angle θ about axis $\mathbf{s}$ followed by a translation $\mathbf{d}$

quaternion $\mathbf{q}$ is related to the instantaneous angular velocity vector ω of the motion as follows

$$\dot{\mathbf{q}} = (1/2)\omega\mathbf{q}. \tag{2.7}$$

2.1.2 Dual Quaternions

A spatial displacement, as shown in Fig. 2.1, consisting of a rotation by angle θ about the unit axis vector $\mathbf{s} = (s_1, s_2, s_3)$ followed by the translation $\mathbf{d}$ can be conveniently represented by a unit dual quaternion $\hat{\mathbf{q}}$. This unit dual quaternion is made up by two quaternions $\mathbf{q}$ and $\mathbf{q^0}$, i.e. $\hat{\mathbf{q}} = \mathbf{q} + \varepsilon\mathbf{q^0}$ where ε is the dual unit with the property $\varepsilon^2 = 0$. The real part $\mathbf{q}$ is the unit quaternion whose components are the Euler-Rodrigues parameters of the rotation previously discussed. The dual part $\mathbf{q}^0 = (q_1^0, q_2^0, q_3^0, q_4^0)$ is obtained from the quaternion multiplication of the translation vector written as a vector quaternion, i.e. $\mathbf{d} = (d_1, d_2, d_3, 0)$, and the unit quaternion $\mathbf{q}$ as follows

$$\mathbf{q}^0 = \frac{1}{2}\mathbf{d}\mathbf{q}. \tag{2.8}$$

The translation vector $\mathbf{d}$ can be recovered in terms of the components of the dual quaternion by using the following expression

$$\mathbf{d} = \frac{\mathbf{q}^0\mathbf{q}^* - \mathbf{q}(\mathbf{q}^0)^*}{\mathbf{q}\mathbf{q}^*} \tag{2.9}$$

which in matrix form can be recast as

$$\mathbf{d} = \frac{2}{S^2}\begin{bmatrix} q_4^0 q_1 - q_1^0 q_4 + q_2^0 q_3 - q_3^0 q_2 \\ q_4^0 q_2 - q_2^0 q_4 + q_3^0 q_1 - q_1^0 q_3 \\ q_4^0 q_3 - q_3^0 q_4 + q_1^0 q_2 - q_2^0 q_1 \end{bmatrix} \tag{2.10}$$

where $S^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2$. It is instructive to note that the rotation matrix, described in the previous subsection, and the translation vector yielded from a unit dual quaternion, $\hat{\mathbf{q}} = \mathbf{q} + \varepsilon\mathbf{q^0}$, and a dual multiple of it, $\hat{\mathbf{Q}} = \hat{w}\hat{\mathbf{q}} = \mathbf{Q} + \varepsilon\mathbf{Q}^0$, are preserved. Hence, a dual quaternion $\hat{\mathbf{Q}} = \hat{w}\hat{\mathbf{q}} = \mathbf{Q} + \varepsilon\mathbf{Q}^0$, where $\hat{w} = w + \varepsilon w^0$ is a nonpure dual number such that w is a nonzero scalar, is a valid representation of a spatial displacement. In other words, a unit dual quaternion $\hat{\mathbf{q}}$ and a general dual quaternion $\hat{\mathbf{Q}} = \hat{w}\hat{\mathbf{q}}$ represent the same spatial displacement. Dual quaternions have been used to define a dual projective three-space, also known as the image space of spatial displacements, by Ravani and Roth [47]. It has been widely accepted that in order to represent a valid rigid body displacement a dual quaternion must be of unit magnitude, i.e. $N(\hat{\mathbf{Q}}) = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^{-1} = 1$, and that the components of the real part $\mathbf{Q} = (Q_1, Q_2, Q_3, Q_4)$ and the components of the dual part $\mathbf{Q}^0 = (Q_1^0, Q_2^0, Q_3^0, Q_4^0)$ must satisfy the Plücker, or normal, condition:

$$Q_1Q_1^0 + Q_2Q_2^0 + Q_3Q_3^0 + Q_4Q_4^0 = 0 \tag{2.11}$$

However, the results stated in Eq. 2.4 and in Eq. 2.10 clearly express that a normalization constraint and the Plücker condition are not needed for the design of motions in the image space. With these restrictions removed, CAGD techniques for curve and surface design such as NURBS, Bézier curves, and subdivision schemes can be readily extended to the generation of freeform rigid body motions. See for instance [22], [19], and [20].

Another interpretation of a unit dual quaternion $\hat{\mathbf{q}} = (\hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4)$ is its function as a screw operator (see Fig. 2.2). Let $\mathbf{s}$ be the unit direction vector of a line and $\mathbf{s}^0$ the moment of the line with respect to the origin of coordinates, which can be obtained as the cross product of a vector from the origin to any point on the line and $\mathbf{s}$. These are called the Plücker vectors of the line. Denoting the unit dual vector $\hat{\mathbf{s}} = \mathbf{s} + \varepsilon \mathbf{s}^0 = (\hat{s}_1, \hat{s}_2, \hat{s}_3)$, where each component is a dual number, e.g. $\hat{s}_1 = s_1 + \varepsilon s_1^0$, and the dual angle $\hat{\theta} = \theta + \varepsilon h$, the four dual components of the dual quaternion representing the screw displacement by angle θ and translation h about and along the screw axis $\hat{\mathbf{s}}$ are given by the dual Euler parameters as follows

$$\hat{q}_1 = \hat{s}_1 \sin\hat{\theta}, \ \ \hat{q}_2 = \hat{s}_2 \sin\hat{\theta}, \ \ \hat{q}_3 = \hat{s}_3 \sin\hat{\theta}, \ \ \hat{q}_4 = \cos\hat{\theta} \tag{2.12}$$

where $\hat{q}_1^2 + \hat{q}_2^2 + \hat{q}_3^2 + \hat{q}_4^2 = 1$.

Thus, a function $\hat{\mathbf{Q}}(t)$ in the image space represents a rigid body motion regardless of the geometry of the rigid body, i.e. the dual quaternion representation of displacements is frame invariant. As per dual unit quaternions $\hat{\mathbf{q}}$, their time derivative is related to the dual velocity $\hat{\mathbf{V}}$ of the displacement by the following expression [22]

$$\dot{\hat{\mathbf{q}}} = (1/2)\hat{\mathbf{V}}\hat{\mathbf{q}} \tag{2.13}$$

where $\hat{\mathbf{V}} = \omega + \varepsilon \mathbf{V}$, being ω and $\mathbf{V}$ the angular and translational velocities

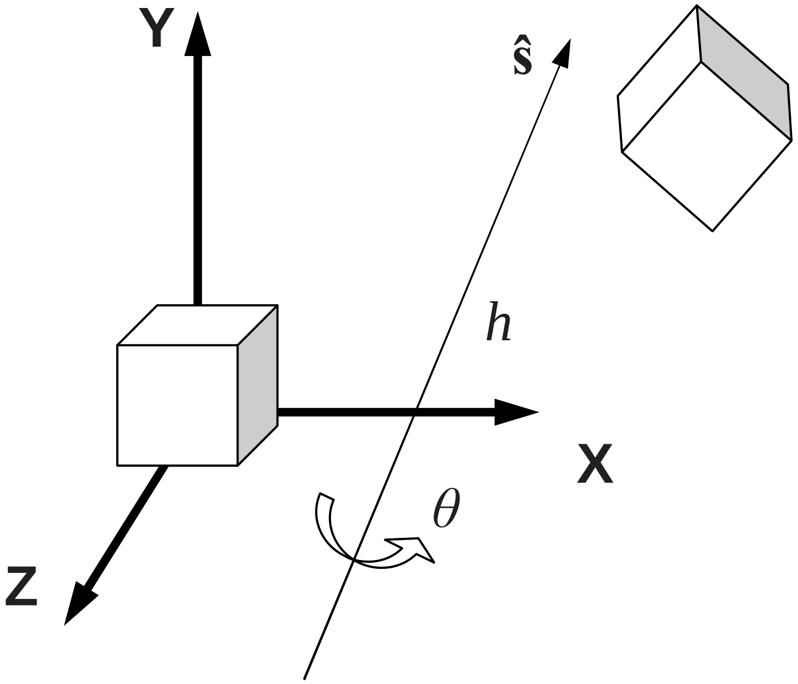

Figure 2.2: A screw displacement consisting of a rotation by angle θ about and a translation h along the screw axis $\hat{\mathbf{s}}$

about and along the instantaneous screw axis, respectively, written as vector quaternions.

It should be noted that dual quaternions encompass single rotations, translations, and planar displacements, all of them being particular cases of spatial displacements. For details on quaternions, dual quaternions and their use in kinematics, the interested reader is referred to [42] and [43].

Because of their intuitive geometric structure and appeal for computations, dual quaternions are an elegant tool to represent spatial displacements that feature computational advantages compared to the traditional matrix representation.

2.2 Point, Plane, and Line Displacements

The spatial displacement of a point $\mathbf{P}$, whose homogeneous coordinates are (P_1, P_2, P_3, P_4), acted upon by a nonunit dual quaternion, $\hat{\mathbf{Q}} = \mathbf{Q} + \varepsilon\mathbf{Q}^0$, is given by (see [48]and [49])

$$\tilde{\mathbf{P}} = \mathbf{QPQ}^* + P_4[(\mathbf{Q}^0)\mathbf{Q}^* - \mathbf{Q}(\mathbf{Q}^0)^*] \tag{2.14}$$

where $\tilde{\mathbf{P}}$ denotes the homogeneous coordinates of the point after the displacement; $\mathbf{Q}^* = (-Q_1, -Q_2, -Q_3, Q_4)$ and $(\mathbf{Q}^0)^* = (-Q^0_1, -Q^0_2, -Q^0_3, Q^0_4)$ are the conjugates of $\mathbf{Q}$ and $\mathbf{Q}^0$, respectively.

Similarly, a plane $\mathbf{M}$ represented by its homogeneous coordinates $(\mathbf{m}, m_4)$, where $\mathbf{m} = (m_1, m_2, m_3)$ is the unit normal of the plane and m_4 is the negative of the perpendicular distance from the origin of the reference frame to the plane, can be transformed to the plane $\tilde{\mathbf{M}}$ as follows (see Xia and Ge [50])

$$\tilde{\mathbf{M}} = \mathbf{QMQ}^* + \mathbf{Q}^0\mathbf{m}\mathbf{Q}^* - \mathbf{Qm}(\mathbf{Q}^0)^* \tag{2.15}$$

Finally, a general dual quaternion $\hat{\mathbf{Q}}$ acting on a vector quaternion $\hat{\mathbf{x}}$, which represents a line written in terms of the Plücker vectors, i.e. $\hat{\mathbf{x}} = \mathbf{x} + \varepsilon\mathbf{x}^0 = (\hat{x}_1, \hat{x}_2, \hat{x}_3)$, causes the displacement of $\hat{\mathbf{x}}$ onto a new line represented by the vector quaternion $\hat{\mathbf{X}}$ given by (see Li and Ge [51])

$$\hat{\mathbf{X}} = \frac{\hat{\mathbf{Q}}\hat{\mathbf{x}}\hat{\mathbf{Q}}^*}{\hat{\mathbf{Q}}\hat{\mathbf{Q}}^*} \tag{2.16}$$

where $\hat{\mathbf{Q}}^* = \mathbf{Q}^* + \varepsilon(\mathbf{Q}^0)^*$ is the conjugate of $\hat{\mathbf{Q}}$.

2.3 Rational Bézier Motions

As explained in section 2.1, a function $\hat{\mathbf{Q}}(t)$ in the image space represents a motion of a rigid body. Here, a review of one- and two-parameter rational Bézier motions developed by Ge and Sirchia [48] is presented as well as the computation of the swept surface of a cylinder undergoing such motions as discussed in Xia and Ge [50, 52] and Xia [53] which will be used in further chapters.

2.3.1 One- and Two-Parameter Motions

Suppose we are given a $(m+1)\times(n+1)$ array of dual quaternions $\hat{\mathbf{Q}}_{i,j} = \hat{w}_{i,j}\hat{\mathbf{q}}_{i,j}; (i = 0, ..., m; j = 0, ..., n)$, where $\hat{\mathbf{q}}_{i,j}$ are unit dual quaternions and $\hat{w}_{i,j}$ are dual weights. We may define the following tensor-product Bézier surface in the space of dual quaternions representing a two parameter motion of an object

$$\hat{\mathbf{Q}}^{m,n}(u,v) = \mathbf{Q}^{m,n}(u,v) + \varepsilon\mathbf{R}^{m,n}(u,v) \tag{2.17}$$

where the real part is given by

$$\mathbf{Q}^{m,n}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} B_i^m(u)B_j^n(v)\mathbf{Q}_{i,j} \tag{2.18}$$

and the dual part by

$$\mathbf{R}^{m,n}(u,v) = \sum_{k=0}^{m}\sum_{l=0}^{n} B_k^m(u)B_l^n(v)\mathbf{R}_{k,l} \tag{2.19}$$

$B_i^m(u), B_j^n(v), B_k^m(u)$ and $B_l^n(v)$ are Bernstein polynomials defined over the sequence parameters $0 \le u \le 1$ and $0 \le v \le 1$, respectively. Substituting

Eq. 2.18 and Eq. 2.19 into Eqs. 2.14, 2.15, and 2.16 yield the surface trajectory traced out by a point, a plane, and a line, respectively. It is worth noting that a tensor product surface in dual quaternion coordinates of degree (m, n) corresponds to a two-parameter rational Bézier motion with degree $(2m, 2n)$ in cartesian space. The control dual quaternions of the surface in the dual quaternion space have been termed as the kinematic control structure; depending on the geometric feature that undergoes the motion, i.e. point, plane, or line, the surface generated will have points, planes, or lines as control features which are termed the geometric control structure. The geometric control structure form a $(2m + 1) \times (2n + 1)$ array and is determined by the $(m + 1) \times (n + 1)$ array of Bézier quaternions $\mathbf{Q}_{i,j}$ and $\mathbf{R}_{i,j}$ as well as by the coordinates of the specific geometric feature.

In this dissertation, we make use of the trajectory $\tilde{\mathbf{M}}^{2n,2m}(u, v)$ of a moving plane $\mathbf{M}$ in matrix form, which can be written as follows (see Ge et al. [54])

$$\tilde{\mathbf{M}}^{2n,2m}(u, v) = \left[H^{2n,2m,*}(u, v)\right] \mathbf{M} \tag{2.20}$$

where the 4×4 matrix $[H^{2n,2m,*}(u, v)]$ represents the rigid transformations of the plane $\mathbf{M}$ under the two-parameter motion. The two-variable matrix function can be put in tensor product Bézier form as

$$\left[H^{2n,2m,*}(u, v)\right] = \sum_{g=0}^{2n} \sum_{h=0}^{2m} B_g^{2n}(u) B_h^{2m}(v) \left[H^*_{g,h}\right] \tag{2.21}$$

where $\left[H^*_{g,h}\right]$ denote the Bézier control matrices and they are given by

$$\left[H^*_{g,h}\right] = \sum_{i+k=g} \sum_{j+l=h} \frac{C_i^n C_j^m C_k^n C_l^m}{C_g^{2n} C_h^{2m}} \left[H^*_{i,j,k,l}\right] \tag{2.22}$$

where terms like C_i^n denote binomial coefficients and

$$\left[H^*_{i,j,k,l}\right] = \left[Q^+_{ij}\right]\left[Q^-_{kl}\right] + \left[Q^-_{kl}\right]\left[R^{m+}_{ij}\right] - \left[Q^+_{ij}\right]\left[R^{m-}_{kl}\right] \tag{2.23}$$

with

$$\left[Q^+_{ij}\right] = \begin{bmatrix} Q_{ij,4} & -Q_{ij,3} & Q_{ij,2} & Q_{ij,1} \\ Q_{ij,3} & -Q_{ij,4} & -Q_{ij,1} & Q_{ij,2} \\ -Q_{ij,2} & Q_{ij,1} & Q_{ij,4} & Q_{ij,3} \\ -Q_{ij,1} & -Q_{ij,2} & -Q_{ij,3} & Q_{ij,4} \end{bmatrix} \tag{2.24}$$

$$\left[Q^-_{kl}\right] = \begin{bmatrix} Q_{kl,4} & -Q_{kl,3} & Q_{kl,2} & -Q_{kl,1} \\ Q_{kl,3} & Q_{kl,4} & -Q_{kl,1} & -Q_{kl,2} \\ -Q_{kl,2} & Q_{kl,1} & Q_{kl,4} & -Q_{kl,3} \\ Q_{kl,1} & Q_{kl,2} & Q_{kl,3} & Q_{kl,4} \end{bmatrix} \tag{2.25}$$

$$\left[R^{m+}_{ij}\right] = \begin{bmatrix} R_{ij,4} & -R_{ij,3} & R_{ij,2} & 0 \\ R_{ij,3} & R_{ij,4} & -R_{ij,1} & 0 \\ -R_{ij,2} & R_{ij,1} & R_{ij,4} & 0 \\ -R_{ij,1} & -R_{ij,2} & -R_{ij,3} & 0 \end{bmatrix} \tag{2.26}$$

$$\left[R^{m-}_{kl}\right] = \begin{bmatrix} R_{kl,4} & -R_{kl,3} & R_{kl,2} & 0 \\ R_{kl,3} & R_{kl,4} & -R_{kl,1} & 0 \\ -R_{kl,2} & R_{kl,1} & R_{kl,4} & 0 \\ R_{kl,1} & R_{kl,2} & R_{kl,3} & 0 \end{bmatrix} \tag{2.27}$$

In the above, $Q_{ij,k}$ and $R_{ij,k}$; $(k = 1, ..., 4)$ denote the components of the quaternions $\mathbf{Q}_{i,j}$ and $\mathbf{R}_{i,j}$, respectively.

A one-parameter motion is a simplification of the two parameter motion when only one parameter, t, is considered. For instance, the following represents a rational Bézier curve of degree n in the space of dual quaternions

$$\hat{\mathbf{Q}}(t) = \sum_{i=0}^{n} B_i^n(t)\hat{\mathbf{Q}}_i = \sum_{i=0}^{n} B_i^n(t)\hat{w}_i\hat{\mathbf{q}}_i \tag{2.28}$$

being $\hat{\mathbf{q}}_i$ and $\hat{w}_i$; $(i = 0, ..., n)$ a set of unit dual quaternions and dual weights, respectively; $0 \leq t \leq 1$. Again, by substituting Eq. 2.28 into Eq. 2.15 the surface trajectory $\tilde{\mathbf{M}}^{2n}(t)$ of a moving plane $\mathbf{M}$ can be obtained

$$\tilde{\mathbf{M}}^{2n}(t) = \sum_{k=0}^{2n} B_k^{2n}(t)\tilde{\mathbf{M}}_k \tag{2.29}$$

where $\tilde{\mathbf{M}}_k$ denote the Bézier control planes given by

$$\tilde{\mathbf{M}}_k = \frac{1}{C_k^{2n}} \sum_{i+j=k} C_i^n C_j^n (\mathbf{Q_i M Q}_j^* + \mathbf{R}_i \mathbf{m} \mathbf{Q}_j^* - \mathbf{Q}_i \mathbf{m} \mathbf{R}_j^*) \tag{2.30}$$

Equation 2.29 can be put in matrix form as follows

$$\tilde{\mathbf{M}}^{2n}(t) = \left[H^{2n,*}(t)\right] \mathbf{M} \tag{2.31}$$

where

$$\left[H^{2n,*}(t)\right] = \sum_{k=0}^{2n} B_k^{2n}(t) \left[H_k^*\right] \tag{2.32}$$

is the rigid transformation matrix and

$$\left[H_k^*\right] = \sum_{i+j=k} \frac{C_i^n C_j^n}{C_k^{2n}} \left(\left[H_i^+\right]\left[H_j^-\right] + \left[H_j^-\right]\left[H_i^{*0+}\right] - \left[H_i^+\right]\left[H_j^{*0-}\right]\right) \tag{2.33}$$

are the control Bézier matrices with

$$\left[H_i^+\right] = \begin{bmatrix} Q_{i,4} & -Q_{i,3} & Q_{i,2} & Q_{i,1} \\ Q_{i,3} & Q_{i,4} & -Q_{i,1} & Q_{i,2} \\ -Q_{i,2} & Q_{i,1} & Q_{i,4} & Q_{i,3} \\ -Q_{i,1} & -Q_{i,2} & -Q_{i,3} & Q_{i,4} \end{bmatrix} \tag{2.34}$$

$$\left[H_j^-\right] = \begin{bmatrix} Q_{j,4} & -Q_{j,3} & Q_{j,2} & -Q_{j,1} \\ Q_{j,3} & Q_{j,4} & -Q_{j,1} & -Q_{j,2} \\ -Q_{j,2} & Q_{j,1} & Q_{j,4} & -Q_{j,3} \\ Q_{j,1} & Q_{j,2} & Q_{j,3} & Q_{j,4} \end{bmatrix} \tag{2.35}$$

$$\left[H_i^{*0+}\right] = \begin{bmatrix} R_{i,4} & -R_{i,3} & R_{i,2} & 0 \\ R_{i,3} & R_{i,4} & -R_{i,1} & 0 \\ -R_{i,2} & R_{i,1} & R_{i,4} & 0 \\ -R_{i,1} & -R_{i,2} & -R_{i,3} & 0 \end{bmatrix} \tag{2.36}$$

$$\left[H_j^{*0-}\right] = \begin{bmatrix} R_{j,4} & -R_{j,3} & R_{j,2} & 0 \\ R_{j,3} & R_{j,4} & -R_{j,1} & 0 \\ -R_{j,2} & R_{j,1} & R_{j,4} & 0 \\ R_{j,1} & R_{j,2} & R_{j,3} & 0 \end{bmatrix} \tag{2.37}$$

The dual quaternion Bézier curve given by Eq. 2.28 defines a rational Bézier motion of degree $2n$, as is obvious from Eq. 2.14. It is noted here that $\hat{\mathbf{Q}}(t)$ is required to satisfy neither the normalization nor the Plücker conditions as discussed in subsection 2.1.2. This statement also applies in the case of two-parameter motion.

2.3.2 Swept Surface of a Cylinder Undergoing Rational Bézier Motions

A half circular cylinder of radius r aligned with the X axis can be represented as a developable surface in rational Bézier form as follows

$$\mathbf{M}^2(s) = \sum_{i=0}^{2} B_i^2(s)\mathbf{M}_i \tag{2.38}$$

where $0 \le s \le 1$, $\mathbf{M}^2(s)$ are the homogeneous coordinates of the tangent plane corresponding to a parameter s, and the control planes are given by

$$\mathbf{M}_0 = (0, 1, 0, -r), \quad \mathbf{M}_1 = (0, 0, 1, 0), \quad \mathbf{M}_2 = (0, -1, 0, -r) \tag{2.39}$$

Recalling the duality between plane and point geometries (see, e.g. [55] which states that any geometric theorem or algorithm regarding points can be transformed into the equivalent regarding planes by exchanging 'points' with 'planes' and 'join' with 'intersection'; then, any ruling line of the cylinder can be evaluated as the intersection of the last two intermediate planes $\mathbf{M}_0^1(s)$ and $\mathbf{M}_1^1(s)$ in the de Casteljau algorithm associated with Eq. 2.38. The Plücker coordinates of this ruling line $\hat{\mathbf{L}}(s)$ can be conveniently computed from the wedge or exterior product "$\wedge$" of the corresponding intermediate planes, i.e. $\mathbf{M}_0^1(s) \wedge \mathbf{M}_1^1(s)$. The wedge product is an extension of the vector cross product into higher dimensions (see Flanders [56] or McCarthy [43]). The other half of the circular cylinder can be similarly represented but with the middle control plane in an opposite orientation, i.e. $\mathbf{M}_3 = -\mathbf{M}_1 = (0, 0, -1, 0)$.

If the cylinder is undergoing a one-parameter rational Bézier motion of degree $2n$, its rational Bézier form, Eq. 2.38, can be substituted into Eq. 2.29 yielding the displaced coordinates of the tangent plane with parameter s at time instant t as follows

$$\mathbf{M}(s,t) = \left[H^{2n,*}(t)\right] \mathbf{M}^2(s) = \sum_{i=0}^{2} \sum_{j=0}^{2n} B_i^2(s) B_j^{2n}(t) \mathbf{M}_{ij} \tag{2.40}$$

with Bézier control planes

$$\mathbf{M}_{ij} = \left[H_j^{2n,*}(t)\right] \mathbf{M}_i \tag{2.41}$$

Equation 2.40 defines a two-parameter set of tangent planes corresponding to the boundary surface of the swept volume of the cylinder undergoing a rational Bézier motion of degree $2n$.

In order to determine the point representation of the boundary surface, one just needs to determine the point of intersection of the ruling line of the cylinder with the ruling line of the rational Bézier motion for each tangent plane $\mathbf{M}^2(s)$ at the instant t. Since each of the ruling lines corresponds to the intersection of the two intermediate planes in the last step of the corresponding de Casteljau algorithm, the intersection point of the ruling lines is the same point of intersection of these four planes. The point of intersection $\mathbf{P}(s,t)$

can be obtained from the wedge product of any three of the four planes. For instance,

$$\mathbf{P}(s,t) = \left[H^{2n,*}(t)\right]\mathbf{M}_0^1(s) \wedge \left[H^{2n,*}(t)\right]\mathbf{M}_1^1(s) \wedge \left[H_0^{2n-1,*}(t)\right]\mathbf{M}^2(s) \quad (2.42)$$

For every time instant t', Eq. 2.42 describes a quartic curve $\mathbf{c}(s) = \mathbf{P}(s,t')$ which is the characteristic curve of the enveloping surface at instant t'. Since the entire cylinder consists of two rational quadratic cylindrical surfaces, for every time instant t', there are in general two characteristic curves which are diametrically opposite.

For a finite-height cylinder undergoing a one-parameter rational Bézier motion, the top and bottom faces of the cylinder trace out boundary surfaces as well. One is the developable surface swept by the top and bottom planes. This surface can be computed by plugging the homogeneous coordinates of the plane, with respect to the same fixed frame of the cylinder, into Eq. 2.29. An expression for the ruling of this developable surface can be obtained from the wedge product of the last two intermediate planes in the de Casteljau algorithm; one may check the perpendicular distance from the center of the circular face to the ruling line and only when this distance is smaller or equal than the radius r of the cylinder, the developable surface, i.e the ruling line at a specific time t', is part of the surface swept by the finite cylinder. Otherwise, only the circular edge of the face will contribute to the swept surface. In the case the ruling is part of the swept surface then the circle will be divided into four arcs by the ruling line of the plane and by the corresponding points on the characteristic curve of the cylindrical face. Either case, it is needed to determine the grazing points which are the contribution of the circular edge

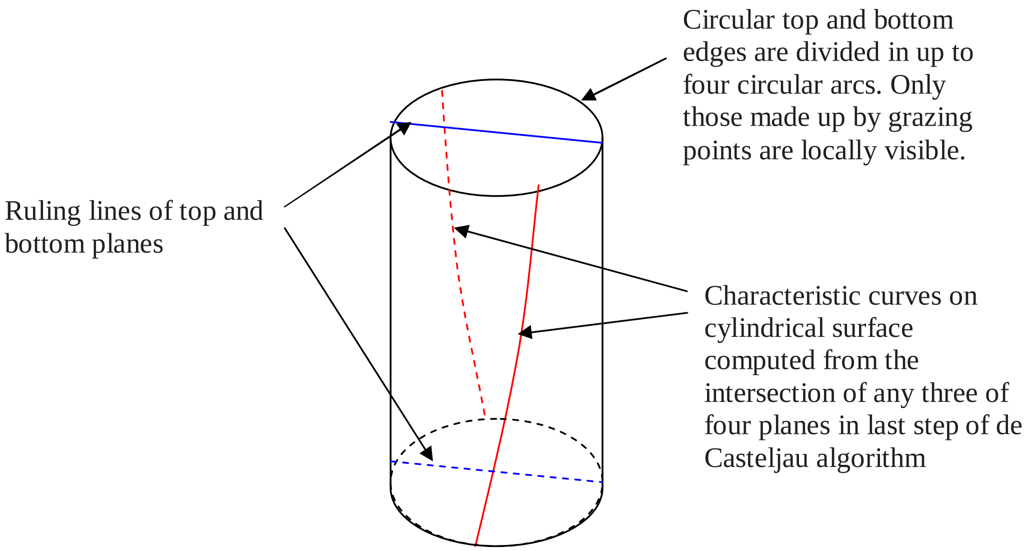
Circular top and bottom edges are divided in up to four circular arcs. Only those made up by grazing points are locally visible.
Ruling lines of top and bottom planes
Characteristic curves on cylindrical surface computed from the intersection of any three of four planes in last step of de Casteljau algorithm

Figure 2.3: The different contributions to the swept surface from the top and bottom planar faces, the circular edges, and the cylindrical surface of a cylinder undergoing one-parameter rational Bézier motion at a specific instant.

to the boundary surface. This can be achieved by evaluating the following condition on each arc [57, 58]

$$(\mathbf{N}_0 \cdot \mathbf{V})(\mathbf{N}_1 \cdot \mathbf{V}) \leq 0 \tag{2.43}$$

where $\mathbf{N}_0$ is the normal of the circular face, $\mathbf{N}_1$ is the normal of the cylinder at a specific point, and $\mathbf{V}$ is the velocity of the point under consideration. It is only necessary to evaluate one point inside each arc to determine whether the whole arc is locally visible or not, i.e. are grazing points or not. This requires the determination of the dual velocity and the velocity of points on the circular edge as described in appendix A. Figure 2.3 illustrates the contributions from the top and bottom planar faces, the circular edges, and the cylindrical surface of a cylinder undergoing one-parameter rational Bézier motion, at a specific instant t', as has just been described.

The swept surface of the two-parameter motion of a finite cylinder can be considered as the enveloping surface of a three-parameter (s, u, v) family of planes. For a two parameter motion, if we fix v and let u vary, we have a one-parameter motion, then a characteristic curve $\mathbf{P}^u(s, u, v)$ can be generated from this one-parameter motion; by a similar procedure, a characteristic curve $\mathbf{P}^v(s, u, v)$ can be obtained by fixing u and varying v. Because normals of all the points on $\mathbf{P}^u(s, u, v)$ are perpendicular to $\mathbf{V}_u$ and normals of all the points on $\mathbf{P}^v(s, u, v)$ are perpendicular to $\mathbf{V}_v$, where $\mathbf{V}(\mathbf{V}_u, \mathbf{V}_v)$ is the velocity of the point under consideration, so the normal of the intersection point of above two curves must be perpendicular to $\mathbf{V}$. Then for a given set of parameter values (u_0, v_0), the intersection of the two-characteristic curves, $\mathbf{P}^u(s, u, v_0)$ and $\mathbf{P}^u(s, u_0, v)$, on the same cylindrical surface yields the corresponding characteristic point of the cylinder for the two-parameter motion. The intersection point can be obtained from the condition that the point of intersection is the concurrent point of the planes that generate the characteristic points on each one of the one-parameter motions, i.e. Eq. 2.42. This may lead to four, two, or no points at all that belong to the swept surface of the cylindrical surface at the specific parameters (u_0, v_0) (see Xia and Ge [52] and Xia [53]).

As for the top and bottom planar faces of the cylinder undergoing a two-parameter motion, the point that belongs to the swept surface of the plane undergoing a two-parameter motion can be computed as the intersection of any three of the four planes in the last step of the two-parameter de Casteljau algorithm when applied to Eq. 2.20. The distance from this point to the center of the circular face may be checked in order to determine if it lies on the finite cylinder. Finally, in order to determine the points on the circular edge that are grazing points, it is required to determine the points that are locally visible both in u and v directions. Thus, the points that satisfy the following two conditions

$$\begin{aligned}(\mathbf{N}_0 \cdot \mathbf{V}_u)(\mathbf{N}_1 \cdot \mathbf{V}_u) &\leq 0 \\ (\mathbf{N}_0 \cdot \mathbf{V}_v)(\mathbf{N}_1 \cdot \mathbf{V}_v) &\leq 0 \qquad (2.44)\end{aligned}$$

are the grazing points that belong to the swept surface of the two-parameter motion of the circular edge. The velocity $\mathbf{V}(\mathbf{V}_u, \mathbf{V}_v)$ can be obtained from the instantaneous dual velocity and instantaneous screw axis. Figure 2.4 sketches the different contributions to the swept surface from the top and bottom planar faces, the circular edges, and the cylindrical surface of a cylinder undergoing two-parameter rational Bézier motion, at a specific set of parameters u and v, as explained.

The dual velocity is calculated by normalizing the function $\hat{\mathbf{Q}}^{m,n}(u, v)$, finding the partial derivatives $\frac{\partial \hat{\mathbf{Q}}}{\partial u}$ and $\frac{\partial \hat{\mathbf{Q}}}{\partial v}$ after the normalization, and solving for the dual velocities $\hat{\mathbf{V}}_u$ and $\hat{\mathbf{V}}_v$ from Eq. 2.13. The dual velocities $\hat{\mathbf{V}}_u$ and $\hat{\mathbf{V}}_v$ are not to be confused with the actual velocities $\mathbf{V}_u$ and $\mathbf{V}_v$ of the point of interest since the dual velocities are the velocities with respect to the instantaneous screw axes of the motion in the u and v directions, respectively. The dual velocity $\hat{\mathbf{V}}$ is related to the angular velocity Ω, the linear velocity Ω^0, and the unit dual vector $\mathbf{v}$ representing the instantaneous screw axis as follows [22]

$$\hat{\mathbf{V}} = 2\hat{\Omega}\hat{\mathbf{v}} \qquad (2.45)$$

where $\hat{\Omega} = (\Omega + \varepsilon\Omega^0)$ is the magnitude of the dual velocity with $\Omega \geq 0$ and

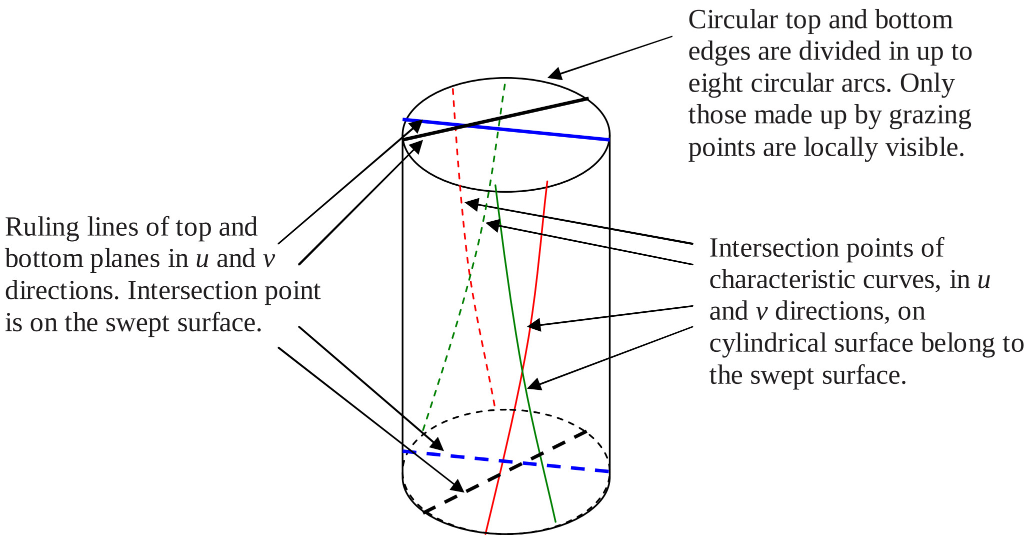
Circular top and bottom edges are divided in up to eight circular arcs. Only those made up by grazing points are locally visible.
Ruling lines of top and bottom planes in u and v directions. Intersection point is on the swept surface.
Intersection points of characteristic curves, in u and v directions, on cylindrical surface belong to the swept surface.

Figure 2.4: The different contributions to the swept surface from the top and bottom planar faces, the circular edges, and the cylindrical surface of a cylinder undergoing two-parameter rational Bézier motion at a specific set of parameters u and v.

$\Omega^0 \geq 0$. Obviously, after knowing the instantaneous screw axes and dual velocities in the u and v directions it is possible to find the velocity of any point on the cylinder as the velocity of any point on the screw axis plus the relative velocity of the point on the cylinder with respect to the point on the screw axis.

Chapter 3

Piecewise Line-Symmetric Spherical Motions for 5-Axis CNC Tool Path Planning

3.1 Introduction

It is common practice in Computer Graphics and Computer Aided Geometric Design (CAGD) to approximate a space curve such as a cubic B-spline with C^0 piecewise line segments, or better, with G^1 continuous biarcs [59]. Such techniques are commonly employed in robot motion planning to generate joint trajectories in what is called the configuration space (C-space) [60, 61]. There is abundant literature on the subject (e.g. [62, 63]) but the general approach consists of optimizing some parameter or objective function while avoiding obstacles and kinematic singularities which are mapped onto the C-space. A similar approach is followed in CNC motion generation for 5-axis machining. In this context, a discrete set of tool positions, or cutter location (CL) data, is generated out of the geometry of the surface to be machined; then, the inverse kinematics is performed for each CL data, the corresponding joint parameters are obtained, and the C-space techniques are applied (see, e.g. [64, 65, 66]).

However, in most commercial machines a simple piecewise linear interpolation of the joint parameters is commonly used which requires a huge amount of data in order to achieve the desired tolerance and does not offer velocity continuity since the cutter has to abruptly change its direction of motion and orientation between line segments compromising surface accuracy and machining time [14, 67]. Furthermore, a drawback of the C-space for motion design is that it does not seamlessly reflect the traits of the actual rigid body motion and does not allow for intuitive manipulation and fine-tuning of the motion.

This is why the space defined by the four components of a quaternion is recognized as an elegant tool for handling spherical motions and it is referred to as the *image space* of spherical displacements because it permits a direct description of the spherical motion of a rigid body [47, 42]. Typically, unit quaternions are used and the resulting image space is a unit hypersphere [43]. Alternatively, when four components of a quaternion are considered as homogeneous coordinates, the resulting image space is a projective three-space where no normalization of the quaternions is required in order to be valid representations of spherical displacements [48]. Thus, the study of a spherical motion corresponds to that of a curve (called image curve) in the image space. For example, a great circle of the unit hypersphere corresponds to a pure rotational motion about a fixed axis. This property has formed the basis for the so-called spherical linear interpolation (Slerp) by Shoemake [15] and has been used for spherical motion planning in robotics and 5-axis machining, yet the angular velocity is not continuous [68, 69]. Although regular (or small) circular arcs on the unit hypersphere such as spherical biarcs have been used to generate interpolating spherical motions (see, for example, [16]) and even C^1 quaternion biarcs on the hypersphere have been used to approximate cubic B-spline quaternion curves [19], it is far less well known that a planar curve in the image space corresponds to a line-symmetric motion [51]. Line-symmetric motions offer more flexibility to the motion designer since they are not restricted to be circular arcs and can be designed in the projective three-space decreasing the computational burden inasmuch as no constraint in the magnitude of the quaternions is imposed.

The purpose of the current chapter is to put line-symmetric motions into use from the viewpoint of motion approximation and show their potential applications in Computerized Numerical Control (CNC) machining simulation and tool path planning. To this end quaternion biarcs are used for motion design. Kinematically, this means that we can use a velocity-continuous piecewise line-symmetric spherical motion to approximate the B-spline rational spherical motion. The advantage of using quaternion biarcs is that they can be generated much more efficiently than cubic b-spline quaternion curves.

The content hereby presented is organized as follows. Firstly, some kinematics fundamentals are briefly reviewed including quaternions and line-symmetric motions. Secondly, the kinematic constraint manifold of a positioning head for 5-axis machining is obtained and its differential properties are discussed which will be used in the application discussed next. Thirdly, the biarc technique is extended to handle quaternions resulting in an image curve describing a piecewise line-symmetric spherical motion. Then, the quaternion biarcs are used for 5-axis tool path generation; the algorithm is sketched and an example is discussed. Concluding remarks are drawn at the end of the chapter.

3.2 Kinematics Fundamentals

In this section the representation of spherical displacements using quaternions and line-symmetric motions are reviewed in the extent that pertains to this chapter.

3.2.1 Spherical Displacements

A unit quaternion $\mathbf{q} = q_1\mathbf{i}+q_2\mathbf{j}+q_3\mathbf{k}+q_4$ representing a spherical displacement is made up by the so-called Euler-Rodrigues parameters as follows

$$q_1 = s_1 \sin\frac{\theta}{2}, q_2 = s_2 \sin\frac{\theta}{2}, q_3 = s_3 \sin\frac{\theta}{2}, q_4 = \cos\frac{\theta}{2} \tag{3.1}$$

where θ and the unit vector $\mathbf{s} = (s_1, s_2, s_3)$ represent the angle and the axis of the rotation, respectively. The quaternion basis units $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ satisfy the fundamental multiplication rules

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{3.2}$$

The quaternion $\mathbf{q}$ is said to be a unit quaternion since its magnitude is one, i.e. $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$. Although it has been widely accepted that only unit quaternions represent spherical displacements, it has been shown that this restriction is not necessary in case of rational motion synthesis (see Purwar and Ge [70]). Therefore, a unit quaternion $\mathbf{q}$ and a multiple of it $\mathbf{Q} = w\mathbf{q} = (Q_1, Q_2, Q_3, Q_4)$, $w > 0$, represent the same rotation since the components of $\mathbf{Q}$ are homogeneous coordinates of $\mathbf{q}$ and the quaternion space is referred to as the image space of spherical kinematics.

The spherical displacement of a point $\mathbf{P}$, whose homogeneous coordinates are (P_1, P_2, P_3, P_4), is given by

$$\tilde{\mathbf{P}} = \mathbf{Q}\mathbf{P}\mathbf{Q}^* \tag{3.3}$$

where $\tilde{\mathbf{P}}$ denotes the homogeneous coordinates of the point after the displacement; $\mathbf{Q}^* = (-Q_1, -Q_2, -Q_3, Q_4)$ is the conjugate of $\mathbf{Q}$. Thus, a curve $\mathbf{Q}(t)$ in the image space describes the spherical motion of a rigid body being t a parameter usually associated with time.

The time derivative $\dot{\mathbf{q}}$ of a unit quaternion $\mathbf{q}$ is related to the instantaneous angular velocity vector $\dot{\theta}$ of the rotation as follows

$$\dot{\mathbf{q}} = (1/2)\dot{\theta}\mathbf{q} \tag{3.4}$$

$\dot{\theta}$ is a vector quaternion whose scalar part is zero. For details on quaternions and the image space, the interested reader is referred to [42], [43], and, [47].

3.2.2 Line-Symmetric Spherical Motion

A general line-symmetric motion is a geometric construction of motion in which an arbitrary position in the space is rotated half-turn about a continuous set of lines. The set of positions obtained is called a line-symmetric motion. The arbitrary position is called *polar position* and the ruled surface formed by the set of lines is the *basic surface* of the line-symmetric motion. If the polar position is described as a quaternion $\mathbf{P}$ relative to a frame F; and the rotation of the i-th line-symmetric position with respect to the polar position is described by the quaternion $\mathbf{s}_i$; then, the line-symmetric rotation $\mathbf{Q}_i$ of each one of the positions relative to F is

$$\mathbf{Q}_i = \mathbf{P}\mathbf{s}_i \tag{3.5}$$

Since every one of the quaternions $\mathbf{Q}_i$ is displaced half-turn with respect to the unique polar position $\mathbf{P}$ in the quaternion space then $\mathbf{P}$ is analogous to the normal of a plane. Thus, $\mathbf{P}$ can be obtained from the wedge product "$\wedge$" of any three of the quaternions $\mathbf{Q}_i$ $(i = 0, 1, \ldots, n)$, e.g.

$$\mathbf{P} = \mathbf{Q_0} \wedge \mathbf{Q_1} \wedge \mathbf{Q_2} \tag{3.6}$$

This means that all the quaternions representing a line-symmetric spherical motion lie on a hyperplane in the image space. Hence, a quadratic curve on the image space indeed represents a line-symmetric spherical motion. Therefore, the curve can be defined by a set of three Bézier control positions and written as a rational Bézier conic, for instance. For a thorough discussion on line-symmetric motions see [42] and [51].

3.3 Kinematic Constraint Manifold of a CNC Positioning Head

In this chapter, the problem of spherical motion planning in 5-axis CNC machining will be addressed assuming that the tool is attached to a positioning head as shown in Fig. 3.1. Nevertheless, a similar analysis applies for a tilting rotary-table type of machine setting up the appropriate coordinate frames. In our case, we attach the moving frame **xyz** to the tool and the fixed frame **XYZ** to the machine. The designation of angles A and C follows the traditional G-code convention as standardized by ISO 6983 [71].

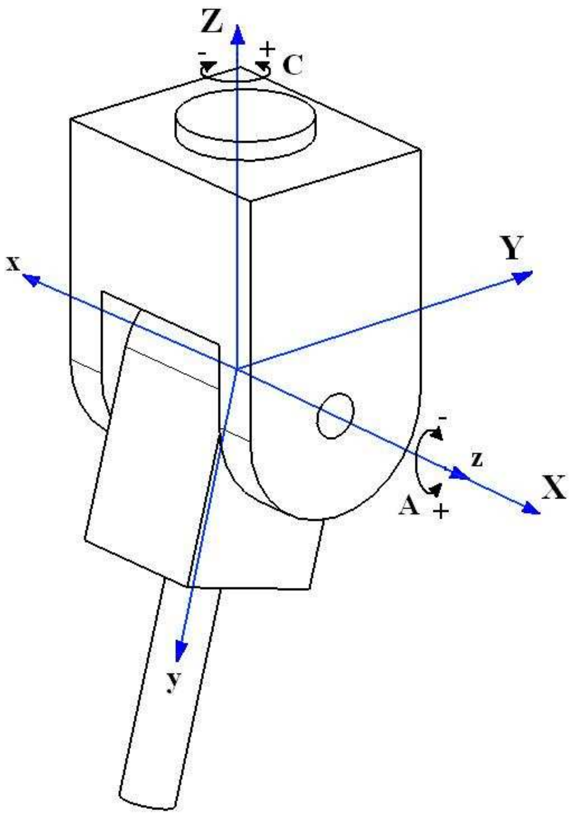

Figure 3.1: Positioning head for 5-axis machining

With these conventions the structure equation of the head is defined by a rotation of the tool about the **z** axis on the moving frame by angle A, a rotation about **x** by an angle of 90^o, and a rotation of the head about **Z** on the fixed frame by angle C. In terms of quaternions we have

$$\mathbf{D}(A,C) = \left(0,0,\sin\frac{C}{2},\cos\frac{C}{2}\right)\left(\sin\frac{90^o}{2},0,0,\cos\frac{90^o}{2}\right)\left(0,0,\sin\frac{A}{2},\cos\frac{A}{2}\right)$$

Expanding this product, the parameterized constraint manifold in $\mathbf{R}^4$ is

$$\mathbf{D}(A,C) = \frac{\sqrt{2}}{2}(D_1(A,C), D_2(A,C), D_3(A,C), D_4(A,C)) \qquad (3.7)$$

where

$$\begin{aligned}
D_1(A,C) &= \cos\left(\frac{C+A}{2}\right) \\
D_2(A,C) &= \cos\left(\frac{C-A}{2}\right) \\
D_3(A,C) &= \sin\left(\frac{C-A}{2}\right) \\
D_4(A,C) &= \sin\left(\frac{C+A}{2}\right)
\end{aligned}$$

Eliminating A and C in Eq. 3.7, the algebraic equation of this surface is

$$D_1^2 - D_2^2 - D_3^2 + D_4^2 = 0 \tag{3.8}$$

The constraint manifold Eq. 3.7 represents, geometrically, the constraint imposed on the positions of the tool by the configuration of the head. In other words, the positions $\mathbf{D} = (D_1, D_2, D_3, D_4)$ that satisfy this equation are reachable by the tool.

Let the coordinates of points in $\mathbf{R}^4$ be represented by the vector $\mathbf{x} = (w, x, y, z)$, so Eq. 3.8 can be viewed as the canonical form of a quadric hypercone:

$$\mathbf{x}^T\,[Q]\,\mathbf{x} = 0 \tag{3.9}$$

with coefficient matrix

$$[Q] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

This may be considered to be the equation of a quadric written in the homogeneous coordinates of three dimensional projective space. The projection of this quadric onto the hyperplane $w = 1$ results in

$$x^2 + y^2 - z^2 = 1 \tag{3.11}$$

which is a right circular hyperboloid of one sheet. The axis of the hyperboloid is the z coordinate axis, and the radius of its center circle is 1. The canonical form of 3.10 is a result of the choice of the fixed and moving reference frames.

Therefore, from a quaternion representing a given position of the tool based on the reference frames described above, the inverse kinematics of the head can be carried out by solving for A and C from the constraint manifold equation.

$$A = \arctan(D_4/D_1) - \arctan(D_3/D_2) \tag{3.12}$$

$$C = \arctan(D_4/D_1) + \arctan(D_3/D_2) \tag{3.13}$$

with $-\pi/2 \leq A \leq \pi/2$ and $-\pi < C \leq \pi$.

In traditional CNC machining, the angles A and C are linearly parameterized with respect to the time t , i.e. in between two consecutive positions, say (A_0, C_0) and (A_1, C_1), the control system performs a linear interpolation of the angular values [14, 58, 72]. Thus, the intermediate angles can be written as

$$A(t) = A_0 + t(A_1 - A_0) \tag{3.14}$$

$$C(t) = C_0 + t(C_1 - C_0) \tag{3.15}$$

Plugging these expressions into Eq. 3.7, one obtains a curve $\mathbf{X}(t)$ that lies on the constraint manifold of the positioning head. It is instructive to analyze the differential properties of such a curve as it will be explained next.

The differential properties of a curve $\mathbf{X}(t)$ in the image space of spherical kinematics can be derived from the following expressions (see McCarthy [73])

$$\kappa^2 = \frac{(\mathrm{d}\mathbf{X}/\mathrm{d}t \wedge \mathrm{d}^2\mathbf{X}/\mathrm{d}t^2 \wedge \mathbf{X}) \cdot (\mathrm{d}\mathbf{X}/\mathrm{d}t \wedge \mathrm{d}^2\mathbf{X}/\mathrm{d}t^2 \wedge \mathbf{X})}{(\mathrm{d}\mathbf{X}/\mathrm{d}t \cdot \mathrm{d}\mathbf{X}/\mathrm{d}t)^3} \tag{3.16}$$

$$\tau = \frac{{}^*(\mathrm{d}\mathbf{X}/\mathrm{d}t \wedge \mathrm{d}^2\mathbf{X}/\mathrm{d}t^2 \wedge \mathrm{d}^3\mathbf{X}/\mathrm{d}t^3 \wedge \mathbf{X})}{(\mathrm{d}\mathbf{X}/\mathrm{d}t \wedge \mathrm{d}^2\mathbf{X}/\mathrm{d}t^2 \wedge \mathbf{X}) \cdot (\mathrm{d}\mathbf{X}/\mathrm{d}t \wedge \mathrm{d}^2\mathbf{X}/\mathrm{d}t^2 \wedge \mathbf{X})} \tag{3.17}$$

where κ is the geodesic curvature and τ is the geodesic torsion. The wedge product, symbolized by $\wedge$, is a generalization of the vector cross product to four dimensional vectors (see Flanders [56]). The term $\mathrm{d}\mathbf{X}/\mathrm{d}t \wedge \mathrm{d}^2\mathbf{X}/\mathrm{d}t^2 \wedge \mathbf{X}$ in Eq. 3.16 is the vector formed from the four 3×3 minors in the 3×4 matrix with rows $\mathrm{d}\mathbf{X}/\mathrm{d}t$, $\mathrm{d}^2\mathbf{X}/\mathrm{d}t^2$, and $\mathbf{X}$. The numerator of Eq. 3.17 is the determinant of the 4×4 matrix formed by the four vectors $\mathrm{d}\mathbf{X}/\mathrm{d}t$, $\mathrm{d}^2\mathbf{X}/\mathrm{d}t^2$, $\mathrm{d}^3\mathbf{X}/\mathrm{d}t^3$, and $\mathbf{X}$.

Substituting the image curve $\mathbf{X}(t)$, which is obtained after plugging Eq. 3.14 and Eq. 3.15 in the constraint manifold Eq. 3.7, into Eq. 3.16 we obtain, after some algebraic manipulation using the symbolic package MAPLE®, the formula for the curvature of $\mathbf{X}(t)$

$$\kappa = \frac{2\,|(C_1 - C_0)(A_1 - A_0)|}{(C_1 - C_0)^2 + (A_1 - A_0)^2} \tag{3.18}$$

The absolute value results from computing the square root of κ^2 from Eq. 3.16. Notice that κ is independent of the parameter t which means that the curvature of the image curve $\mathbf{X}(t)$ is constant.

The torsion of $\mathbf{X}(t)$ is obtained by substituting the image curve into Eq. 3.17. This yields after some algebraic manipulation with MAPLE®

$$\tau = \frac{|(C_1 - C_0)^2 - (A_1 - A_0)^2|}{(C_1 - C_0)^2 + (A_1 - A_0)^2} \tag{3.19}$$

The torsion τ is also constant. Therefore, the image curve $\mathbf{X}(t)$ is a helix in the image space when linear interpolation is performed by the controller. Since in traditional CNC machining, the discrete positions of the tool are linearly interpolated, the image curve is a piecewise helical curve with only C^0 continuity. This is because no smoothing conditions are imposed at the junction of contiguous pieces of the image curve. This means that there is room for improvement in the path generation of the tool since a smoother motion will enhance the appearance of the machined surface and may convey significant increase in the efficiency of the process.

3.4 Interpolation of Orientations with Specified Angular Velocities Using Quaternion Biarcs

Biarcs are a technique of interpolation of data based on the fact that two points and their corresponding tangent vectors can be fitted not by one circular arc but by two, though some special cases need four arcs [59]. In the traditional biarc technique, unit tangent vectors are used so that the fitting curve obtained is G^1 continuous [59, 74]. In this section the goal is to extend this approach in order to interpolate a set of orientations expressed in terms of unit quaternions, which can be regarded as points on a unit hypersphere in $\mathbf{R}^4$, with angular velocity constraints. Moreover, we are not willing to limit our approach to circular arcs but to allow for elliptical, parabolical,

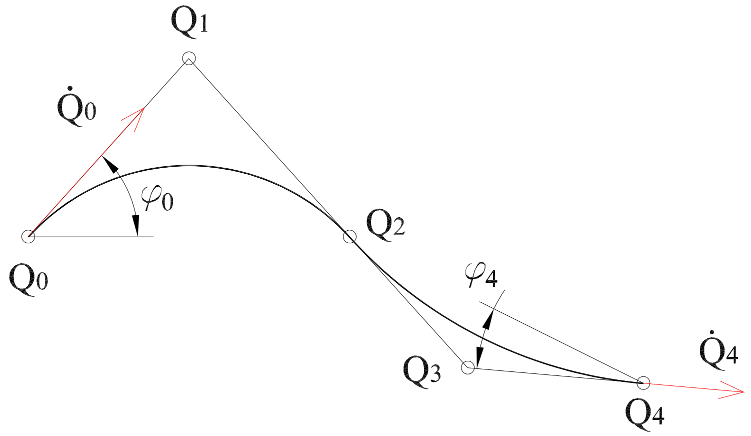

Figure 3.2: Biarcs scheme

or hyperbolical arcs, i.e. quadric arcs, achieving a flexible technique for the motion design task. Therefore, for two given orientations $\mathbf{Q}_s$ and $\mathbf{Q}_e$ with respective angular velocities $\dot{\theta}_s$ and $\dot{\theta}_e$, and consequently with corresponding non-unit tangent vectors $\dot{\mathbf{Q}}_s$ and $\dot{\mathbf{Q}}_e$, an interpolating piecewise quadric arc, i.e. a biarc, must meet the following conditions: it passes through $\mathbf{Q}_s$ and $\mathbf{Q}_e$; the tangent vectors at those points are $\dot{\mathbf{Q}}_s$ and $\dot{\mathbf{Q}}_e$ respectively; and the continuity of the arcs is C^1 at the junction point. Although Eq. 3.4 is only valid for unit quaternions, there is no need to impose normalization constraints on the interpolating quaternions if homogeneous coordinates are used to compute the rotations of points of a rigid body since a unit quaternion and a multiple of it represent the same rotation as explained before.

Let $\mathbf{Q}_s = \mathbf{Q}_0$ and $\mathbf{Q}_e = \mathbf{Q}_4$ represent two desired orientations of a rigid body with respect to an arbitrary fixed frame as illustrated in Fig. 3.2.

It should be noticed that Fig. 3.2 sketches a biarc for regular planar points although the current chapter deals with quaternions which are impossible to visualize in $\mathbf{R}^4$. However, the considerations on biarcs can be extended to the quaternion space since a quadratic arc is also a planar curve in $\mathbf{R}^4$.

The unit quaternions $\mathbf{Q_0}$ and $\mathbf{Q_4}$ and their respective derivatives is all what is needed to apply the biarc method. It is required to find the Bézier control quaternions $\mathbf{Q_1}$, $\mathbf{Q_2}$, $\mathbf{Q_3}$ such that the arcs $\mathbf{C}_1$ defined by $\mathbf{Q_0}$, $\mathbf{Q_1}$, $\mathbf{Q_2}$, and $\mathbf{C}_2$ defined by $\mathbf{Q_2}$, $\mathbf{Q_3}$, $\mathbf{Q_4}$ can be written as rational Bézier curves

$$\mathbf{C}_1(t) = \frac{w_0\mathbf{Q_0}\mathrm{B}_0^2(t) + w_1\mathbf{Q_1}\mathrm{B}_1^2(t) + w_2\mathbf{Q_2}\mathrm{B}_2^2(t)}{w_0\mathrm{B}_0^2(t) + w_1\mathrm{B}_1^2(t) + w_2\mathrm{B}_2^2(t)} \tag{3.20}$$

$$\mathbf{C}_2(t) = \frac{w_2\mathbf{Q_2}\mathrm{B}_0^2(t) + w_3\mathbf{Q_3}\mathrm{B}_1^2(t) + w_4\mathbf{Q_4}\mathrm{B}_2^2(t)}{w_2\mathrm{B}_0^2(t) + w_3\mathrm{B}_1^2(t) + w_4\mathrm{B}_2^2(t)} \tag{3.21}$$

where w_i are the weights and $\mathrm{B}_0^2(t)$, $\mathrm{B}_1^2(t)$, and $\mathrm{B}_2^2(t)$ are quadratic rational Bernstein polynomials. When the weights of the end points are equal to one, the arcs can be written in standard form [4], as follows

$$\mathbf{C}_1(t) = \frac{\mathbf{Q_0}\mathrm{B}_0^2(t) + w_1\mathbf{Q_1}\mathrm{B}_1^2(t) + \mathbf{Q_2}\mathrm{B}_2^2(t)}{\mathrm{B}_0^2(t) + w_1\mathrm{B}_1^2(t) + \mathrm{B}_2^2(t)} \tag{3.22}$$

$$\mathbf{C}_2(t) = \frac{\mathbf{Q_2}\mathrm{B}_0^2(t) + w_3\mathbf{Q_3}\mathrm{B}_1^2(t) + \mathbf{Q_4}\mathrm{B}_2^2(t)}{\mathrm{B}_0^2(t) + w_3\mathrm{B}_1^2(t) + \mathrm{B}_2^2(t)} \tag{3.23}$$

Similar to the traditional CAGD theory, it can be said that if the weight is smaller than one an elliptical arc is obtained, if the weight is equal to one the arc is parabolic, and if the weight is larger than one the arc is a hyperbola. Recalling the derivative for a rational Bézier B-spline conic in standard form and denoting the parameter intervals $\Delta_1 = u_1 - u_0$ and $\Delta_2 = u_2 - u_1$ for each arc, the unknown control quaternions $\mathbf{Q_1}$ and $\mathbf{Q_3}$ are correlated to the interpolated quaternions by

$$\mathbf{Q_1} = \mathbf{Q_0} + \frac{\Delta_1}{2w_1}\dot{\mathbf{Q}}_\mathbf{0} \tag{3.24}$$

$$\mathbf{Q_3} = \mathbf{Q_4} - \frac{\Delta_2}{2w_3}\dot{\mathbf{Q}}_\mathbf{4} \tag{3.25}$$

Let

$$\alpha = \frac{\Delta_1}{2w_1} \tag{3.26}$$

and

$$\beta = \frac{\Delta_2}{2w_3} \tag{3.27}$$

Then, Eq. 3.24 and Eq. 3.25 can be written as follows

$$\mathbf{Q_1} = \mathbf{Q_0} + \alpha\dot{\mathbf{Q}}_\mathbf{0} \tag{3.28}$$

$$\mathbf{Q_3} = \mathbf{Q_4} - \beta\dot{\mathbf{Q}}_\mathbf{4} \tag{3.29}$$

$\mathbf{Q_2}$ is the end point of one arc and the start point of the next one; thus, the derivatives at this quaternion must be equal for both arcs in order to guarantee C^1 continuity of the biarc segment. From this

$$\mathbf{Q_2} = \frac{\beta\mathbf{Q_1} + \alpha\mathbf{Q_3}}{\alpha + \beta} \tag{3.30}$$

α and β are degrees of freedom that allow the designer to fine-tune the motion which also depend on the parameter intervals and weights. For instance, after choosing α and β one can pick the parameter interval and solve for the weight of each arc segment. This course of action makes sense in the context of CNC machining since the parameter interval is related to the sampling time of the controller which is a fixed feature of the machine hardware. Of course, one can instead pick the weights and use the derived values of α and β for computation of the control quaternions of the arc segments. It should be recalled that a change in the weights affects the shape of the actual motion of the rigid body and that the rational arc segments can always be reparameterized such that the path of the motion does not change but only the speed of the motion does as explored in [70]. The choice of $\alpha = \beta$ facilitates computations, avoids awkward-looking curves and, as pointed out in [74], optimized values of α and β do not offer significant advantages and this is the alternative used in this work. In such a case Eq. 3.30 becomes

$$\mathbf{Q_2} = \frac{\mathbf{Q_1} + \mathbf{Q_3}}{2} \tag{3.31}$$

It must also be noticed that the local parameter t defined as follows must be used when evaluating each of the arcs Eqs. 3.22 and 3.23

$$t = \frac{u - u_i}{\Delta_i}, (i = 0, 1) \tag{3.32}$$

This procedure can be carried out for every two consecutive quaternions representing a set of specified key orientations such that the b-spline curve obtained, which is composed by biarc segments, interpolates the series of input orientations and is C^1 continuous. Finally, since each of the segments is a quadratic arc, the rigid body motion obtained is a continuous-velocity piecewise line-symmetric spherical motion.

3.5 Tool Path Generation by Means of Piecewise Line-Symmetric Spherical Motion

This section describes the application of the quaternion biarcs, or piecewise line-symmetric spherical motion, in path generation of a cutter mounted on a positioning head as that described in section 3.3. Let us assume that a set of n key orientations of a cutter is specified in quaternion form $\mathbf{q}_i, (i = 1, 2, \ldots, n)$ along with their respective angular velocities $\dot{\theta}_i$. Such a CL data set can be obtained from discretization procedures and local gouging analyses as those found elsewhere [10, 75].

It should be noticed that the angular velocity can be established from velocity tracking control if this type of control is available in the machine [64] and then the quaternion derivative can be computed. Otherwise, local estimation methods borrowed from CAGD may be implemented [7, 76, 77] in order to compute the tangent at each quaternion. This is the type of approach that we propose in this dissertation. Basically, one can use the unit direction vector of the line that joins two quaternions $\mathbf{q}_{i-1}$ and $\mathbf{q}_{i+1}$ as the direction of the tangent vector $\mathbf{t}_i$ at $\mathbf{q}_i$. Kinematically this means that the tangent vector points along the direction of the screw displacement between the positions represented by $\mathbf{q}_{i-1}$ and $\mathbf{q}_{i+1}$ [21, 22]. For the first and last quaternions, $\mathbf{q}_0$ and $\mathbf{q}_n$, the tangent vector can be computed from the line that joins $\mathbf{q}_0$ and $\mathbf{q}_1$, and $\mathbf{q}_{n-1}$ and $\mathbf{q}_n$, respectively, as depicted in Fig 3.3. As far as the magnitude goes one can use the maximum angular velocity allowed by the machine motors as an upper bound and eventually fine tune it after the interpolation if it is required.

The quaternions are assembled from the angles A and C of the CL data

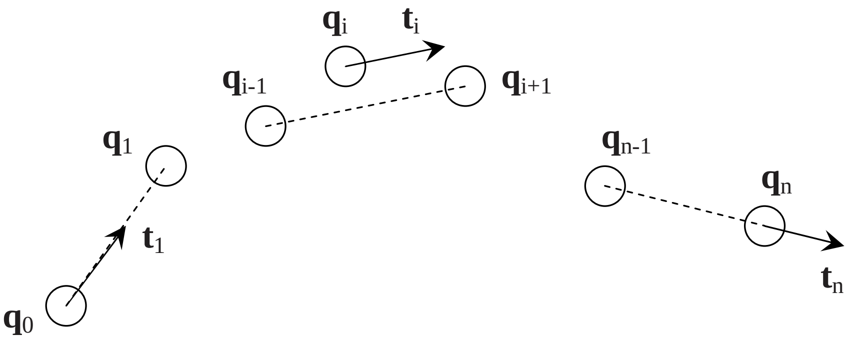

Figure 3.3: Local estimation technique of the tangent vector at the junction points of the biarcs

using Eq. 3.7. The time derivatives are computed by means of Eq. 3.4, if the angular velocity is known at each tool position, or by the local estimation method aforementioned. The parameter interval for each arc may be attached to the sampling period of the controller. Here, we assume a uniform parameterization of the piecewise biarc curve. The value of α is found by solving Eq. 3.26. The weights of each quaternion biarc can be interactively adjusted by the motion designer in order to fine-tune the motion of the tool and this way improve the accuracy of the machining if required. The control quaternions of each biarc segment are evaluated using Eqs. 3.28, 3.29, and 3.30. After the control quaternions of the piecewise biarc curve have been obtained, several intermediate positions on each biarc segment can be evaluated by using Eqs. 3.22 and 3.23; i.e. a piecewise line-symmetric motion interpolates the key orientations and approximates the actual motion of the machine. Equations 3.12 and 3.13 provide reference values of joint displacement to the machine controller.

After applying the quaternion biarc interpolation method to the orientation data in Tab. 3.1, several intermediate orientations are obtained, see Fig. 3.4. In this case a weight of 0.5 and an angular velocity of 0.2 rad/s were assumed

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10
$A(^o)$	0	5.737	18.2106	35.4977	45.4198	46.9018	54.8255	60.8735	73.8073	83.3593
$C(^o)$	0	-5.946	-16.6741	-20.4189	-30.4049	-47.4576	-58.5542	-74.7468	-81.813	-86.7452

Table 3.1: Orientation data used in example

for all of the biarc segments. The plots of the joint displacements and C-space are shown in Fig. 3.5 which are smoother compared to the traditional piecewise linear interpolation used by CNC interpolators even though the curves obtained are very close to the line segments. Figure 3.6 shows a closeup on a segment of the joint trajectories where the smoothness claimed for the curves obtained from the biarc technique is evidenced.

Figure 3.7 shows the motion generated when the weight was changed to 2. As expected, a slight change in the path of the motion is evidenced. Also, the plots of the displacements of the joints suffer changes as seen in Fig. 3.8. Nonetheless, the joint displacement curve is smoother than the piecewise linear one. The algorithm was implemented on a 1.50 GHz laptop using MATLAB. The computational time to generate the motion and the joint displacement curves of these examples was less than 2 s.

3.6 Conclusions

Quaternion biarcs permit to interpolate a set of orientations by means of a piecewise quadratic curve constituted by arcs. Quadratic curves in the image space of spatial motions describe line-symmetric motions so that the b-spline obtained describes a piecewise line-symmetric motion. The spherical line-symmetric motion achieved with this b-spline curve is C^1 continuous which is a desirable feature for orientation interpolation in 5-axis machining of sculp-

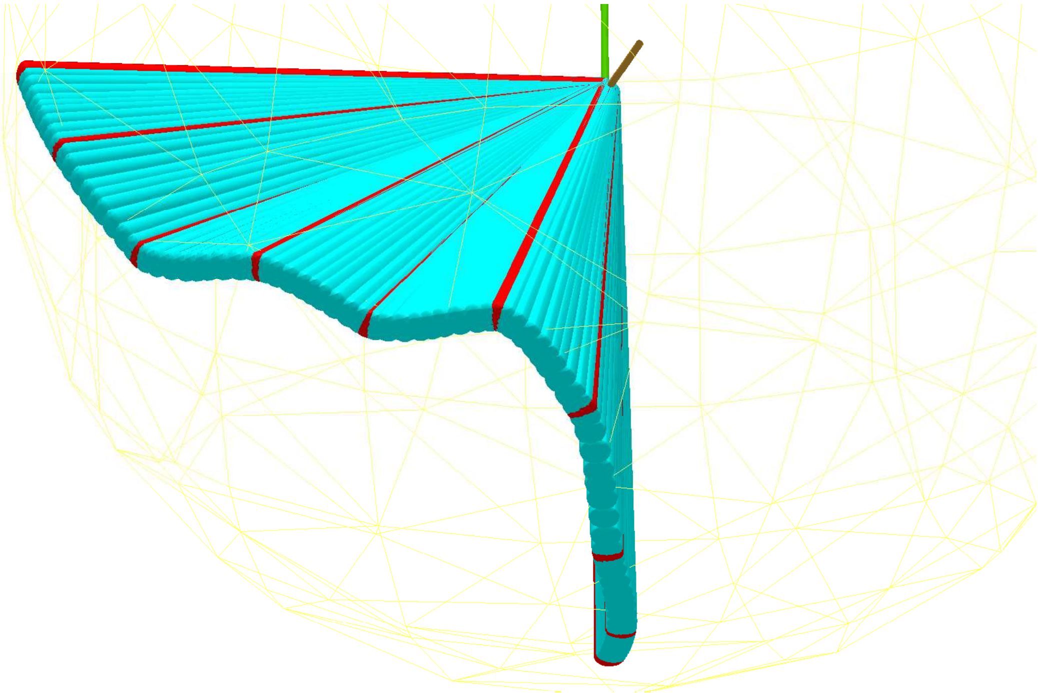

Figure 3.4: Line-symmetric positions interpolating ten key orientations; all weights equal to 0.5

tured surfaces in order to avoid poor surface finish as in the traditional method of linear interpolation of tool orientation. Moreover, the biarc technique is a superior alternative since the discrete orientations are exactly interpolated. The computational efficiency and flexibility offered for fine-tuning of the piecewise line-symmetric motion makes it an appealing choice for motion interpolation. Therefore, the method described in this chapter is useful in CAD/CAM, robot path planning, computer graphics, and simulation of some types of mechanisms.

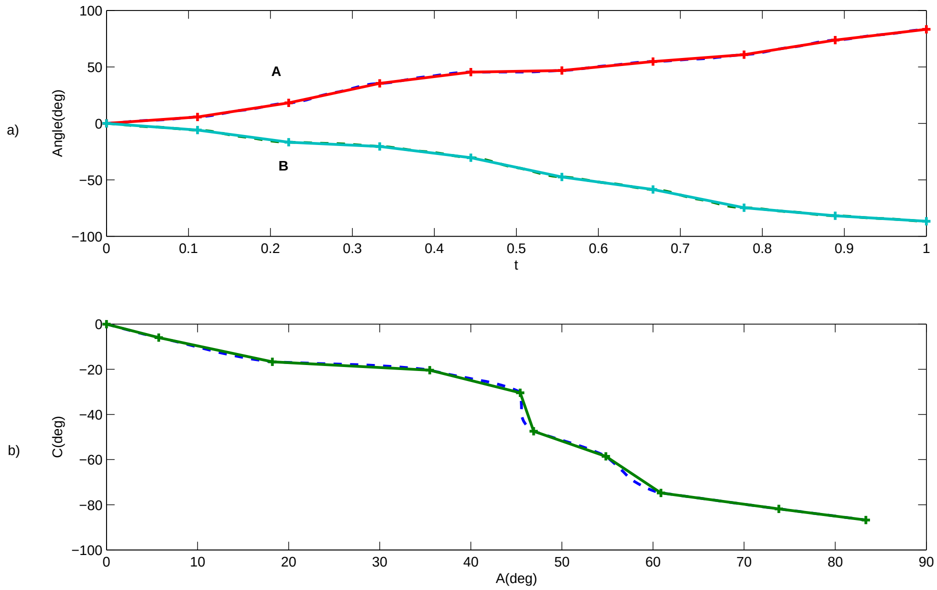

Figure 3.5: a) Joint displacements b)C-space. The solid lines correspond to the piecewise linear interpolation and the dashed curves are obtained by the biarc technique. All weights equal to 0.5.

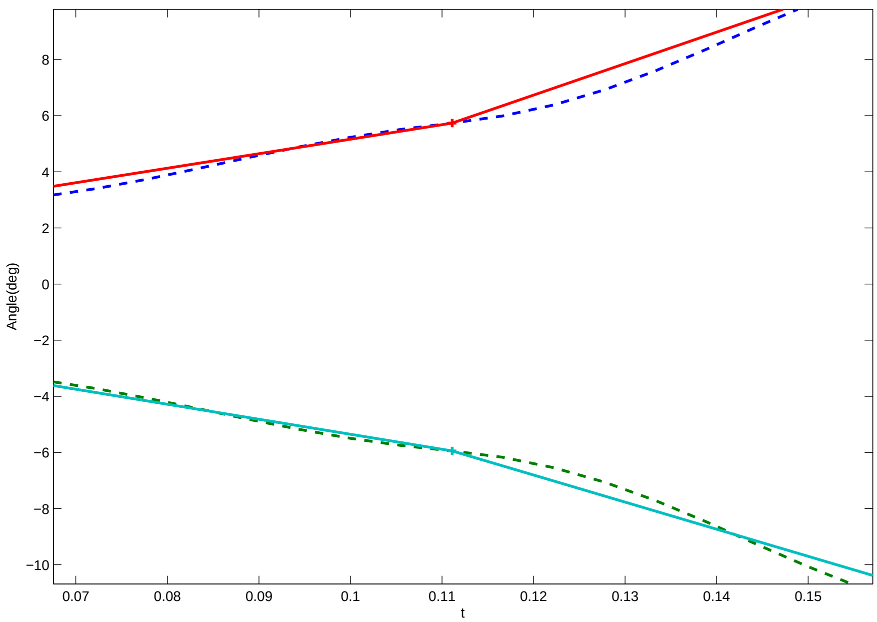

Figure 3.6: Closeup on joint trajectories where the smoothness of the curves obtained by the biarc technique, dashed curves, is evidenced compared to the solid line segments.

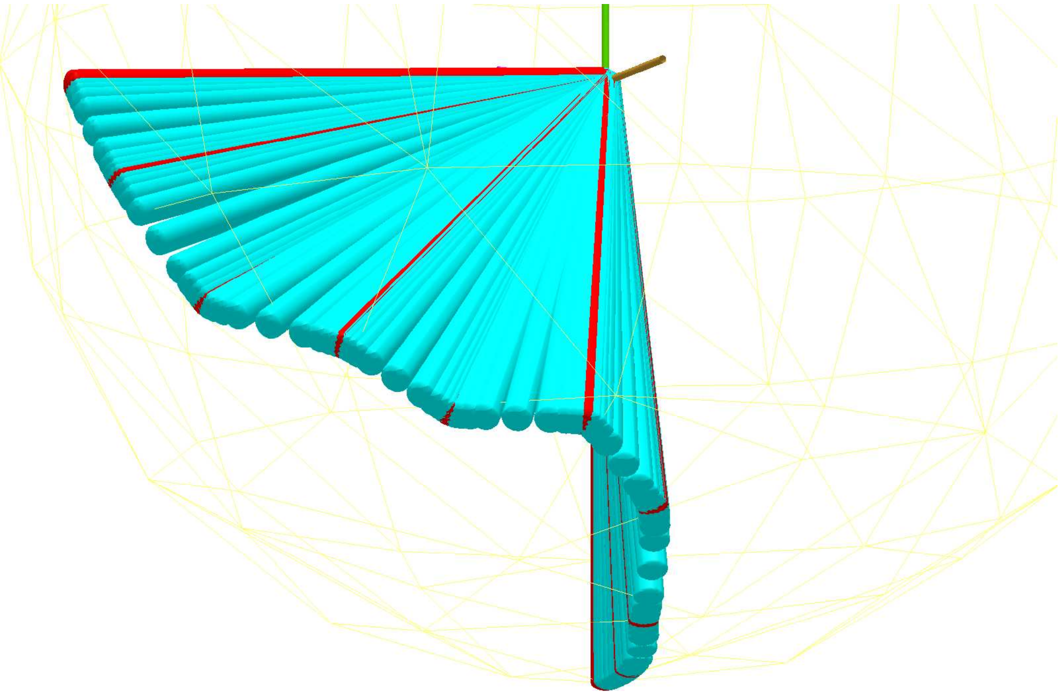

Figure 3.7: Line-symmetric positions interpolating ten key orientations; all weights equal to 2

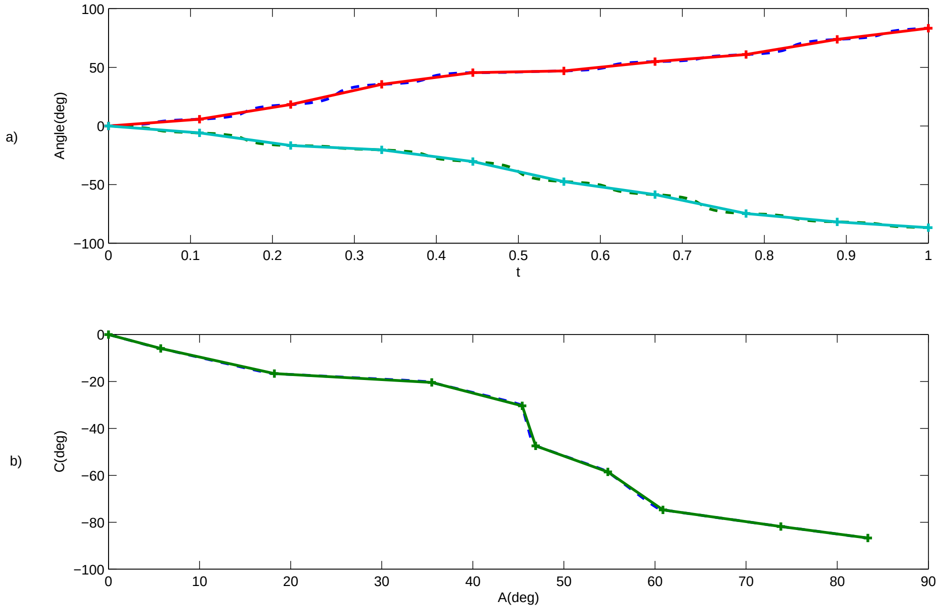

Figure 3.8: a)Joint displacements b)C-space. All weights equal to 2 and compared to linear interpolation

Chapter 4

Motion Generation And Swept Volume Analysis of a Cylinder Using a Curve Subdivision Scheme

4.1 Introduction

Much of the existing work on the generation of freeform motions such as Bézier and b-spline motions as well as the analyses of their swept volumes are based on basis functions and control nets of tool positions (see, for example, Jüttler and Wagner [20], and Xia and Ge [50]). This chapter explores an alternative approach to freeform motion synthesis and swept volume analysis based on subdivision schemes. While the kinematic structure of a continuous model of a freeform motion is defined by the choice of the basis functions, the kinematic structure of a discrete model of a freeform motion is defined by the choice of the subdivision scheme. This is in contrast to the unstructured set of discrete tool positions generated from the conventional NC tool path planning process that converts a kinematic problem of tool path planing into a geometric one at each of the tool positions.

This approach follows our previous work of continuous kinematic geometry that uses dual quaternions for the presentation of spatial displacements. Once a set of displacements are represented as points in the space of dual quaternions, a subdivision scheme for curve generation such as the four-point interpolatory subdivision scheme introduced by Dyn et al. [29] can be readily adapted to discrete motion generation. The focus of this chapter is to develop a fast and robust algorithm to compute the swept volume of the resulting discrete motion that takes advantage of the geometric structure of the subdivision algorithm . A swept volume is defined as the volume traced out by a rigid body when moving along a path. Central to the problem of swept volume analysis is the computation of the boundary surfaces of the swept volume. Swept surface analysis is a well studied subject. There are at least three major approaches to the computation of the swept surface. Wang and Wang [78] introduced the envelope theory which basically makes use of the fact that the points on the swept surface fulfill the envelope condition

$$\mathbf{N}(\mathbf{p}) \cdot \mathbf{V}(\mathbf{p}) = 0 \tag{4.1}$$

where $\mathbf{N}$ is the normal at a point $\mathbf{p}$ on the surface of the rigid body and $\mathbf{V}$ is the instantaneous velocity of the point. This condition is derived from the evaluation of the singularities of the Jacobian of the sweep. Abdel-Malek and Yeh [79] studied the rank-deficiency conditions of the Jacobian and used this result to compute the swept surface of multi-parameter sweep. Blackmore and Leu [80] formulated the sweep differential equation and correlated it to the Lie group structure of the Euclidean motion which led them to establish criteria regarding the geometry and topology of swept volumes. Several other developments and specific approaches can be found in related literature. For a thorough state-of-the-art review on swept volumes see Abdel-Malek et al. [81].

In this chapter a new approach is proposed to compute the swept surface of a cylindrical cutter when a set of discrete or key positions of the cutter is specified. Every position may be a combination of a translation and orientation as well. To this end, the convenient representation of spatial displacements by means of dual quaternions is used, then the four-point interpolatory subdivision scheme [29] is adapted to the interpolation of dual quaternions and used for the generation of in-between positions of a rigid body. Because of the geometric features of the subdivision scheme, the new generated positions can be considered as part of a screw displacement. Here, we take advantage of this feature and make use of the the work presented by Xia [53] and Xia and Ge [50] to calculate the characteristic curve of a swept surface at a specific instant. This approach is numerically robust and computationally efficient since the procedure is based on linear combinations. In section 4.2 the four-point interpolatory subdivision scheme is extended to dual quaternions and its geometric traits are exploited to compute a discrete representation of the swept surface of a cylinder. The algorithm and an example are described in section 4.3. Finally, conclusions are drawn.

4.2 Discrete Computation of the Swept Surface

In this section, a summary of the four-point interpolatory subdivision scheme developed by Dyn, Levin and Gregory [29] is presented. Then, an adaptation of the scheme to handle dual quaternions is introduced along with a kinematic interpretation that is used to generate a discrete representation of the swept surface of a cylindrical cutter.

4.2.1 Four-Point Interpolatory Subdivision Scheme

The four-point interpolatory subdivision scheme has the useful feature that the resulting curve in the limit interpolates the vertices of all intermediate polygons obtained during the subdivision process. The algorithm finds the point $\mathbf{p}_{i+1/2}$ from four consecutive points $\mathbf{p}_{i-1}$, $\mathbf{p}_i$, $\mathbf{p}_{i+1}$, $\mathbf{p}_{i+2}$ by the following formula

$$\mathbf{p}_{i+1/2} = (1/2 + w)(\mathbf{p}_i + \mathbf{p}_{i+1}) - w(\mathbf{p}_{i-1} + \mathbf{p}_{i+2}) \tag{4.2}$$

The new polygon has vertices $\mathbf{p}_{i-1}$, $\mathbf{p}_i$, $\mathbf{p}_{i+1/2}$, $\mathbf{p}_{i+1}$, $\mathbf{p}_{i+2}$, i.e. after each round of subdivision the old points are reinserted along with the newly generated points as to obtain a refined polygon. This procedure is consecutively repeated regarding the augmented set of points as new control points up to any desired level of refinement. Hence, starting from a set of points $\mathbf{p}_i$, $-2 \leq i \leq n + 2$, a new set of points $\mathbf{p}_i^k$, $-2 \leq i \leq 2n + 2$, is generated.

Let $k > 0$ be the level of refinement or subdivision round. Then, the subdivision scheme defines the control points at level $k+1$ by

$$\begin{cases} \mathbf{p}_{2i}^{k+1} = \mathbf{p}_i^k, & -1 \le i \le 2^k n + 1, \\ \mathbf{p}_{2i+1}^{k+1} = (1/2 + w)(\mathbf{p}_i^k + \mathbf{p}_{i+1}^k) - w(\mathbf{p}_{i-1}^k + \mathbf{p}_{i+2}^k), & -1 \le i \le 2^k n \end{cases} \tag{4.3}$$

where $\mathbf{p}_i^0 = \mathbf{p}_i$, $-2 \le i \le n+2$.

The geometric meaning of the subdivision scheme can be interpreted by reorganizing Eqn. (4.2) as follows

$$\mathbf{p}_{i+1/2} = (1/2)(\mathbf{p}_i + \mathbf{p}_{i+1}) + 2w[(1/2)(\mathbf{p}_i + \mathbf{p}_{i+1}) - (1/2)(\mathbf{p}_{i-1} + \mathbf{p}_{i+2})] \tag{4.4}$$

From this expression, it can be seen that the effect of the scheme is to translate the midpoint between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ by a vector $2w\mathbf{e}$, $\mathbf{e} = (1/2)(\mathbf{p}_i + \mathbf{p}_{i+1}) - (1/2)(\mathbf{p}_{i-1} + \mathbf{p}_{i+2})$, along the line through the midpoints of the upper and the lower sides of the polygon as depicted in Fig. 4.1. The parameter w acts as a shape factor by controlling the magnitude of the translation vector. The smaller the value of w the closer the generated point is to the midpoint of the upper side of the polygon. Indeed, when $w = 0$ the new point coincides with this very same midpoint and the scheme reproduces a refined version of the original polygon.

The four point interpolatory subdivision scheme converges to a limit curve $\mathbf{p}(t)$ which is defined over a uniform parametrization in the range $t \in [0, n]$ attaching the point $\mathbf{p}_i^k$ to the parameter value $2^{-k}i, k \ge 0$, $0 \le i \le 2^k n$. When $0 < w < 1/8$ the limit curve of the subdivision process is C^1 continuous and for $w = 1/16$ polynomials up to degree 3 can be reproduced.

Another interesting feature is the local nature of the scheme since changing a specific point only perturbs the neighboring points computed from it; i.e.,

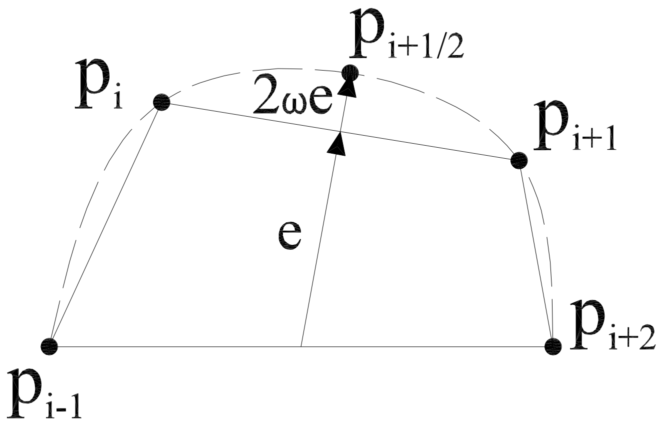

Figure 4.1: 4-Point interpolatory subdivision scheme

the curve segment $\mathbf{p}(t)$, $t \in (i, i+1)$ depends only on $\mathbf{p}_{i-2} \dots \mathbf{p}_{i+3}$.

It is worth to note that when applying the subdivision scheme cyclicly to n points, a closed curve is obtained. However, if an open curve is desired then two additional points on each end of the set of points are required. These additional points can be chosen by controlling the tangent vector of the limit curve at the end points. The tangent vector at a point i obtained at the subdivision stage k can be calculated in terms of its neighboring points, at the same subdivision stage, as follows

$$\mathbf{p}'_i = \frac{2^k}{1-4w}[(1/2)(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}) - w(\mathbf{p}_{i+2} - \mathbf{p}_{i-2})] \tag{4.5}$$

Thus, when $k = 0$, Eq. 4.5 can be used to determine the additional points by manipulating the tangent vector at the endpoints. A simpler solution is to take the endpoints twice and perform the subdivision with this choice of additional points.

Because of these four ingredients: subdivision, locality, interpolation, and shape control; the four-point subdivision scheme is a simple and convenient technique for curve design that has been applied for the generation of ruled surfaces [82] and is also appealing for the interpolation of dual quaternions and the design of motion and computation of the the swept surface as will be discussed next.

4.2.2 Discrete Swept Surface

Suppose a set of discrete positions of a cylindrical cutter is specified along a toolpath. Each of these positions can be represented by a dual quaternion $\hat{\mathbf{Q}}_i$ by attaching the moving reference frame to the center of the cylinder such that its z-axis is along the axis of the cylinder (see [75]). The aforementioned subdivision scheme may be applied directly to the set of dual quaternions in order to generate in-between positions. The new positions obtained by the scheme may need to be checked for gouging avoidance using a similar procedure as described in [75]. If, after certain level of refinement by means of the subdivision scheme, a manufacturing tolerance is not expected to be exceeded, the unconstrained scheme can be continuously applied in order to obtain a refined set of positions of the cutter. It is worth to note that since the converging dual quaternion curve is C^1 as long as the shape parameter $0 < w < 1/8$, then, the converging rigid body motion is also C^1. Shape parameters out of this range cause non-smooth motions which are not useful for numerical control machining simulation [2].

Should the cutter follow an open trajectory, Eq. 4.5 may be used to adjust the tangent of the limit curve at the end dual quaternions. In such a case the relationship between the dual-quaternion derivative $\dot{\hat{\mathbf{Q}}}$ and the dual velocity $\hat{\mathbf{V}}$ can be used in order to take into account the angular velocity $\mathbf{v}$ about and linear velocity $\mathbf{v}^0$ along the instantaneous screw axis of the displacement. Such relationship can be found in [22]:

$$\dot{\hat{\mathbf{Q}}} = (1/2)\hat{\mathbf{V}}\hat{\mathbf{Q}} \tag{4.6}$$

where the dual velocity $\hat{\mathbf{V}} = \mathbf{v}+\epsilon\mathbf{v}^0$. The derivative $\dot{\hat{\mathbf{Q}}}$ defines a tangent vector of the limit curve at the dual quaternion $\hat{\mathbf{Q}}$. Another option to generate the two additional dual quaternions is to input the dual quaternions representing the end positions twice in the corresponding subdivision step as explained in the previous subsection.

After the desired level of refinement has been reached, one might make use of the dual quaternion version of Eq. 4.5 and then by means of Eq. 4.6 solve for the dual velocity at each position, from which the angular and translational velocities could be obtained. Finally, the envelope condition, Eq. 4.1, could be invoked to find the points that belong to the swept surface at each specific position. These points, at each discrete position, which satisfy the envelope condition make up what is known as the *characteristic curve* [78].

However, recalling the geometric meaning of the four-point subdivision scheme, Fig. 4.1, it is realized that the dual quaternion generated lies on the same line passing through the dual quaternions located at the "*midpoints*" of the upper and lower sides of the polygon. Given that a line in the image space represents a screw motion [22], one can take advantage of this geometric fact to calculate the characteristic curve by applying what was described in section 2.3 specifically to the case of a screw motion since a line in the image space is the simplest case of a rational Bézier curve. Before continuing, it is instructive to notice that a line in the image space generates a Darboux's

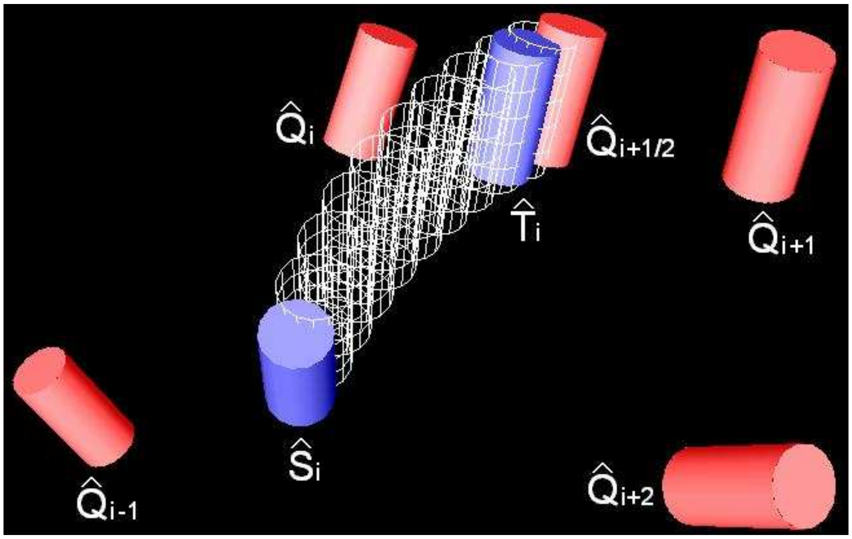

Figure 4.2: Dual quaternion version of the 4-point subdivision scheme. $\hat{\mathbf{Q}}_{i+1/2}$ belongs to the screw motion that contains $\hat{\mathbf{S}}_i$ and $\hat{\mathbf{T}}_i$

motion where the path of any point of a rigid body is a planar curve [42].

Referring to Fig. 4.2, let $\hat{\mathbf{Q}}_{i+1/2}$ denote a dual quaternion obtained by the subdivision scheme, and, $\hat{\mathbf{S}}_i$ and $\hat{\mathbf{T}}_i$ denote the dual quaternions at the lower and upper midpoints of the polygon, respectively. Then, a line segment $\hat{\mathbf{Q}}_i(t)$ representing the screw motion that contains $\hat{\mathbf{Q}}_{i+1/2}$ can be given by

$$\hat{\mathbf{Q}}_i(t) = (1-t)\hat{\mathbf{S}}_i + t\hat{\mathbf{Q}}_{i+1/2}, \quad 0 \le t \le 1 \tag{4.7}$$

Obviously, when $t = 1$ we obtain the dual quaternion $\hat{\mathbf{Q}}_{i+1/2}$ that belongs to the subdivision process.

Now, the trajectory $\tilde{\mathbf{M}}^2(t)$ of any tangent plane $\mathbf{M}$ of the cylinder undergoing the screw motion can be found by plugging $\hat{\mathbf{Q}}_i(t)$ into Eq. 2.15

$$\tilde{\mathbf{M}}^2(t) = \sum_{l=0}^{2} B_l^2(t)\tilde{\mathbf{M}}_l \tag{4.8}$$

where $B_l^2(t)$ are Bernstein basis polynomials of degree 2 and $\tilde{\mathbf{M}}_l$ denote the control Bézier planes given by

$$\tilde{\mathbf{M}}_l = \frac{1}{C_l^2} \sum_{f+g=l} C_f^1 C_g^1 (\mathbf{Q}_f \mathbf{M} (\mathbf{Q}_g^0)^* + \mathbf{Q}_f^0 \mathbf{m} \mathbf{Q}_g^* - \mathbf{Q}_f \mathbf{m} (\mathbf{Q}_g^0)^*) \tag{4.9}$$

and terms such as C_l^2 denote binomial coefficients.

The plane trajectory $\tilde{\mathbf{M}}^2(t)$ can be written in matrix form as follows

$$\tilde{\mathbf{M}}^2(t) = \left[H^{2,*}(t)\right] \mathbf{M} \tag{4.10}$$

where

$$\left[H^{2,*}(t)\right] = \sum_{l=0}^{2} B_l^2(t) \left[H_l^*\right] \tag{4.11}$$

with

$$[H_l^*] = \frac{1}{C_l^2} \sum_{f+g=l} C_f^1 C_g^1 ([H_f^+][H_g^-] + [H_g^-][H_f^{*0+}] - [H_f^+][H_g^{*0-}]) \tag{4.12}$$

and

$$[H_f^+] = \begin{bmatrix} Q_{f,4} & -Q_{f,3} & Q_{f,2} & Q_{f,1} \\ Q_{f,3} & Q_{f,4} & -Q_{f,1} & Q_{f,2} \\ -Q_{f,2} & Q_{f,1} & Q_{f,4} & Q_{f,3} \\ -Q_{f,1} & -Q_{f,2} & -Q_{f,3} & Q_{f,4} \end{bmatrix} \tag{4.13}$$

$$[H_g^-] = \begin{bmatrix} Q_{g,4} & -Q_{g,3} & Q_{g,2} & -Q_{g,1} \\ Q_{g,3} & Q_{g,4} & -Q_{g,1} & -Q_{g,2} \\ -Q_{g,2} & Q_{g,1} & Q_{g,4} & -Q_{g,3} \\ Q_{g,1} & Q_{g,2} & Q_{g,3} & Q_{g,4} \end{bmatrix} \tag{4.14}$$

$$[H_f^{*0+}] = \begin{bmatrix} Q_{f,4}^0 & -Q_{f,3}^0 & Q_{f,2}^0 & 0 \\ Q_{f,3}^0 & Q_{f,4}^0 & -Q_{f,1}^0 & 0 \\ -Q_{f,2}^0 & Q_{f,1}^0 & Q_{f,4}^0 & 0 \\ -Q_{f,1}^0 & -Q_{f,2}^0 & -Q_{f,3}^0 & 0 \end{bmatrix} \tag{4.15}$$

$$[H_g^{*0-}] = \begin{bmatrix} Q_{g,4}^0 & -Q_{g,3}^0 & Q_{g,2}^0 & 0 \\ Q_{g,3}^0 & Q_{g,4}^0 & -Q_{g,1}^0 & 0 \\ -Q_{g,2}^0 & Q_{g,1}^0 & Q_{g,4}^0 & 0 \\ Q_{g,1}^0 & Q_{g,2}^0 & Q_{g,3}^0 & 0 \end{bmatrix} \tag{4.16}$$

Equation 4.8 can be evaluated by means of the de Casteljau algorithm. Recalling the duality between the geometry of points and the geometry of planes [55], it can be said that any ruling line of the developable surface traced out by the tangent plane during the screw motion can be obtained from the intersection of the two intermediate planes $[H_0^{1,*}]\mathbf{M}$ and $[H_1^{1,*}]\mathbf{M}$ of the de Casteljau algorithm. Since we are only interested in the ruling when $t = 1$, we can compute that line which lies on the swept surface of the screw motion by finding the intersection of the two control planes $\tilde{\mathbf{M}}_1$ and $\tilde{\mathbf{M}}_2$. The Plücker coordinates of the intersection line are given by the wedge product of the two planes $\tilde{\mathbf{M}}_1 \wedge \tilde{\mathbf{M}}_2$. The wedge product is a generalization of the cross product into higher dimensions, see [43]. Finally, the intersection between the ruling line on the screw motion and the corresponding ruling line of the cylinder intersect at one point. It must be noticed here that the ruling line on the screw motion and the tangent line parallel to the cylinder axis lie on the same tangent plane. The intersection point can be obtained from the wedge product of any three of the four intermediate planes from the screw motion and from the rational Bézier representation of the cylinder as described by Xia and Ge [50] which leads to an analytical expression for the characteristic curve as explained in subsection 2.3.2. This way, the characteristic curve at a discrete position can be constructed. Figure 4.3 sketches the computation of the point on the characteristic curve as the intersection of the ruling line of the cylinder and the ruling line of the screw motion of the corresponding tangent plane.

The contribution to the swept surface from the upper and bottom planes of the finite cylinder and from the circular edge are computed as described in

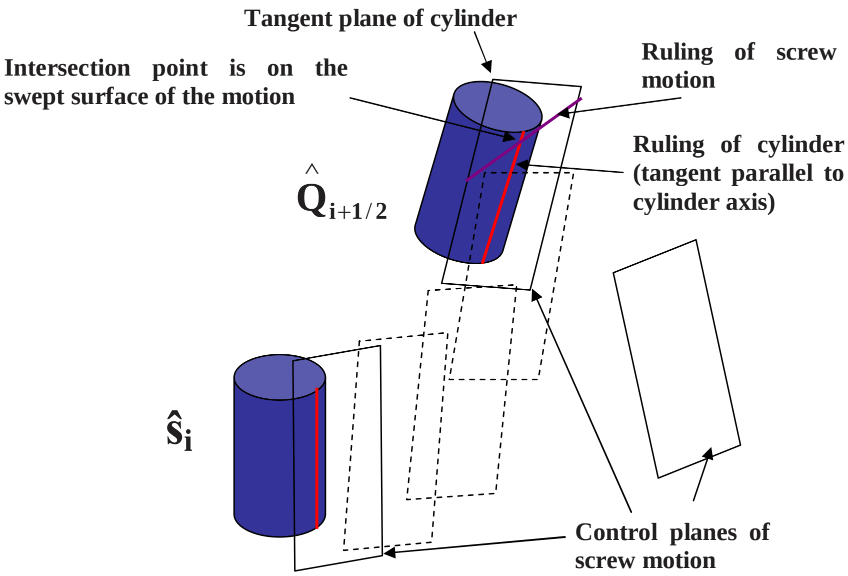
Tangent plane of cylinder
Intersection point is on the swept surface of the motion
Ruling of screw motion
Ruling of cylinder (tangent parallel to cylinder axis)
$\hat{Q}_{i+1/2}$
$\hat{S}_i$
Control planes of screw motion

Figure 4.3: Computation of point on swept surface. The intersection point of the ruling line on the tangent plane of the cylinder and the ruling line of that tangent plane undergoing the screw motion is the point on the characteristic curve

subsection 2.3.2.

4.3 Computer Implementation And Example

The algorithm to generate the swept surface out of a set of discrete positions of a cylindrical cutter can be summarized as follows.

Given an ordered set of dual quaternions representing the discrete positions of a cylindrical cutter, the four-point subdivision scheme is applied, Eq. 4.2. Then, based on the screw motion of the tangent planes of the cylinder, Eq. 4.10, the control structure is obtained for the screw displacement of each plane. Finally, the intersection of the three planes, the two intermediate obtained from the screw motion of the tangent plane for $t = 1$ and any of the intermediate planes that originate the tangent plane in the rational Bézier representation of the cylinder in the discrete position, permits to obtain the points on the characteristic curve, Eq. 2.42. The contributions from the top and bottom planes as well from the circular edges to the swept surface are also determined.

Figure 4.4 shows the characteristic curves of the interpolating positions obtained by the four-point interpolatory subdivision scheme for a cylindrical cutter given ten positions extracted from the NC part program presented by Chiou and Lee [58]. The shape parameter used for this example was $w = 1/16$. The ten cutter contact (CC) points are transcribed in Table 4.1. $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are the cartesian components of the translation vector of the pivot point of the tool, which in this example is the midpoint along the axis of the cylinder. θ_A and θ_C are the corresponding rotations about the $\mathbf{x}$ and $\mathbf{z}$ axes in degrees.

For implementation purposes, a cylindrical cutter of unit height was used. In order to determine which points on the characteristic curve should be displayed, the distance between the computed points and the midpoint along the axis of the cylinder was checked at each discrete position in the local frame of the cutter. If this distance was outside of the range of the unit height cylinder then the points are not displayed. As for the ruling lines traced out by the top and bottom planes, after determining whether they contribute to the swept surface by simple trigonometry taking into account the radius of the cylinder one can determine the effective length of the ruling line. For the circular arcs that contribute to the swept surface, the condition stated in Eq. 2.43 must be checked. Figure 4.5 shows the swept surface with the contributions from the cylindrical surface, the top plane, and the circular edge of the cylinder.

The application was implemented on a 1.50 GHz laptop using Microsoft®

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10
x	-0.0499	-0.0148	0.0004	0.0102	0.0136	0.0125	0.0072	-0.0012	-0.0052	-0.0017
y	1.1717	-1.1867	-2.7558	-4.1193	-5.2711	-6.2307	-6.7454	-6.4152	-5.5897	-4.6435
z	4.8600	5.2681	5.1969	4.9501	4.6330	4.3285	4.2291	4.4644	4.6232	4.4840
θ_A	11.4210	-7.3160	-16.7896	-24.0767	-28.9988	-31.4808	-28.4045	-15.4525	2.6137	22.7803
θ_C	2.7550	-1.1910	0.0809	0.3361	0.3501	0.2974	0.2008	0.0082	0.9420	0.0098

Table 4.1: NC part program for Fig. 4.4

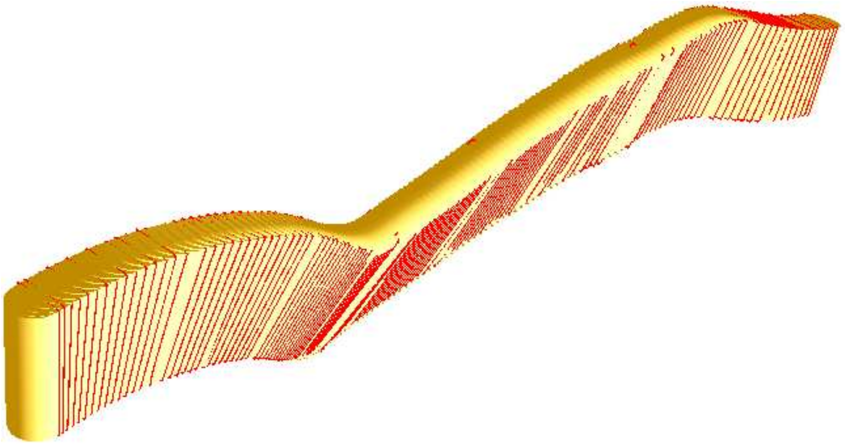

Figure 4.4: Characteristic curves on the surface swept by the cylindrical cutter after four rounds of subdivision; the shape parameter w=1/16.

Visual C++ and OpenGL®. In this example four rounds of subdivision were performed obtaining 135 in-between positions. The computational time was 6.5 min. For a lower level of refinement, e.g. three rounds of subdivision, the computational time is reduced in half.

4.4 Conclusions

The four-point interpolatory subdivision scheme was readily adapted to the interpolation of dual quaternions representing rigid body displacements. Such scheme is numerically robust and computationally simple to implement in order to generate fast interpolation of a rigid body motion. Since the limit curve of such scheme is a C^1 b-spline curve, the motion generated is C^1 as

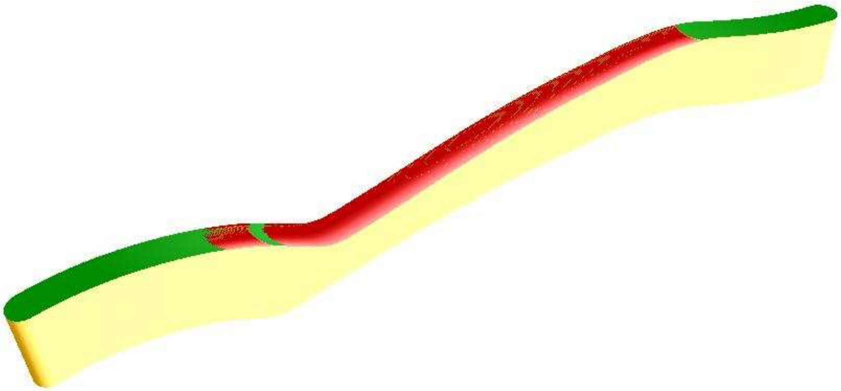

Figure 4.5: Swept volume of example with w=1/16. The yellow surface is traced out by the cylindrical surface; the green is the contribution of the top plane; and the red is the swept of the circular edge of the top face of the cylinder.

long as the shape factor is between the appropriate range from 0 to 1/8. Given the geometric meaning of the algorithm when extended to the image space, the computation of the characteristic curve for each of the discrete positions of a developable surface can efficiently be carried out. In this chapter only a cylindrical body was considered. Nonetheless, a similar approach may be implemented on any developable surface, i.e. generated by the motion of a plane. Moreover, if the convenience of such a surface is not possible, the approach still is useful since it allows to compute the dual velocity for each discrete position in the image space and to determine the velocity of any point on a rigid body and then invoke the envelope condition to calculate the characteristic curve. This last fact has not been implemented by the authors and will be explored in future work.

Overall, the discrete method for motion interpolation and swept surface generation is a useful tool for numerical control tool-path verification and simulation, robot path planning, and computer animation.

Chapter 5

Motion Generation And Swept Volume Analysis of a Cylinder Using a Surface Subdivision Scheme

5.1 Introduction

The notion of two-parameter motions has been introduced by Bottema and Roth [42] when the positions of a rigid body depend on two parameters, u and v say. In this case, the locus of a point is, in general, a surface which is called the trajectory surface; that of a plane is the set of tangent planes of a surface, enveloped by the plane; and the locus of a moving line is its trajectory congruence. The connection with time, a one dimensional phenomenon, can be recovered when arbitrary functions of one parameter t are introduced, i.e. $u(t)$ and $v(t)$; thus, a motion in the ordinary sense is defined. Ge and Sirchia [48] developed computer aided design methods for synthetizing two-parameter rational Bézier and b-spline motions. Ge et al. [54] studied the plane trajectory of a two-parameter rational Bézier motion. Xia and Ge [52] used the duality between point and plane geometry in projective geometry to exactly compute the boundary surfaces of a cylinder undergoing a two-parameter rational Bézier motion.

This chapter extends the approach employed in chapter 4 to the computation of the swept surface of a cylinder undergoing the motion prescribed by the surface (tensor-product) version of the four point interpolatory subdivision scheme which is a two-parameter dependent scheme. Here, we also take advantage of the geometric features of the subdivision scheme in the image space and make use of the results obtained by Xia and Ge [52].

5.2 Discrete Swept Surface Computation

In this section, the tensor product version of the four-point interpolatory subdivision scheme developed by Dyn, Levin and Gregory [29] is presented. Then, an adaptation of the scheme to handle dual quaternions is introduced along with a kinematic interpretation that is used to generate the swept surface of a cylindrical cutter undergoing the two-parameter motion.

5.2.1 Tensor Product Interpolatory Subdivision Scheme

The four point interpolatory subdivision scheme used in chapter 4 can be easily extended for the design of a surface which passes through a set of control points assembled as a regular squarelike grid of the form

$$\mathbf{p}_{i,j}; \quad -2 \leq i \leq n+2, \quad -2 \leq j \leq m+2 \tag{5.1}$$

First, the subdivision rule, Eq. 4.2, is applied to the index i, inserting points between $\mathbf{p}^k_{i,j}$ and $\mathbf{p}^k_{i+1,j}$. Then, the same scheme is applied to the index j

ending up with a denser net of points that can be further refined as required. Thus, the tensor-product form of the four point interpolatory scheme is the following

$$\begin{cases} \mathbf{p}^{k+1}_{2i,2j} &= \mathbf{p}^{k}_{i,j}, \qquad -1 \leq i \leq 2^k n+1, -1 \leq j \leq 2^k m+1, \\ \mathbf{p}^{k+1}_{2i+1,2j} &= (1/2+w)(\mathbf{p}^{k}_{i,j}+\mathbf{p}^{k}_{i+1,j}) - w(\mathbf{p}^{k}_{i-1,j}+\mathbf{p}^{k}_{i+2,j}), \\ & \qquad -1 \leq i \leq 2^k n, -1 \leq j \leq 2^k m+1, \\ \mathbf{p}^{k+1}_{i,2j+1} &= (1/2+w)(\mathbf{p}^{k+1}_{i,2j}+\mathbf{p}^{k+1}_{i,2j+2}) - w(\mathbf{p}^{k+1}_{i,2j-2}+\mathbf{p}^{k+1}_{i,2j+4}), \\ & \qquad -1 \leq i \leq 2^k n+1, -1 \leq j \leq 2^k m. \end{cases} \tag{5.2}$$

It should be noticed here that applying the scheme to the index j first and then to the index i yields the same set of points as sketched in Fig. 5.1.

For $0 < w < 1/8$ the points computed by the above process converge to a surface which has continuous tangent plane. This surface may be viewed as a blending surface of the $m+2$ curves $\mathbf{p}_j(t)$, $t \in [0,n]$, obtained by the application of the refinement rule 4.2 to the points $\{\mathbf{p}_{ij}; -2 \leq i \leq n+2\}$ for $j = -2, \ldots, m+2$. The blending process uses 4.2 again on the sets of control points $\{\mathbf{p}_{-2}(t), \ldots, \mathbf{p}_{m+2}(t)\}$, $t \in [0,n]$, to yield the surface $\mathbf{p}(t,s)$, $t \in [0,n]$, $s \in [0,m]$ which interpolates the curves $\mathbf{p}_j(t)$, $0 \leq j \leq n$.

An appealing feature of subdivision schemes is their ability to represent surfaces of arbitrary topology. Therefore, they can be used to interpolate closed and open surfaces; i.e. patches. In the former case, the subdivision process is applied cyclicly. In the latter case, one may use Eq. 4.5 to control the tangent in each direction i and j in order to generate the additional points required. Another option is to repeat the first and the last control points in

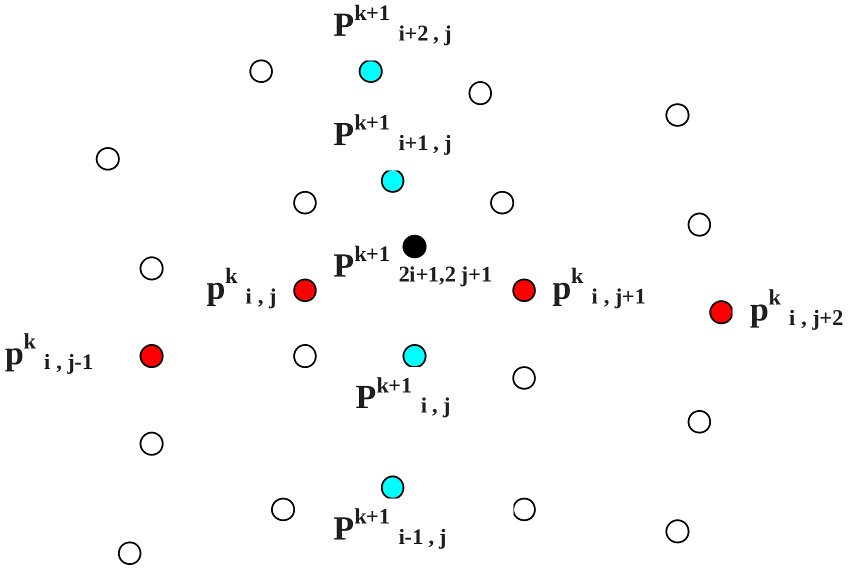

Figure 5.1: Any interior point, represented by the black dot, can be obtained from either a set of points in the i direction (red dots) or a set of points in the j direction (turquoise dots) in the tensor-product subdivision scheme.

the subdivision process as was explained for the curve case in the previous chapter. This is the approach used in this work for implementation purposes.

5.2.2 Discrete Swept Surface

The subdivision scheme 5.2 can be readily extended to interpolate a given set of positions of a rigid body represented by dual quaternions. Since the algorithm converges to a surface with continuous tangent plane the motion is C^1 continuous given the fact that the motion is as smooth as its dual quaternion surface in the image space.

In the previous subsection, it was recalled that any interior point on the converging surface can be obtained from either a set of four points in the i direction or a set of four points in the j direction. This means that any interior dual quaternion can be obtained from a set of four dual quaternions in the i or in the j direction from the immediately previous subdivision stage. Therefore, a position obtained from the subdivision scheme is the intersection of two screw motions, one in the i direction and one in the j direction. Each screw motion can be described by a line in the image space as follows

$$\hat{\mathbf{Q}}(i) = (1 - i)\hat{\mathbf{s}}_i + i\hat{\mathbf{q}}_{i+1/2}, 0 \leq i \leq 1 \tag{5.3}$$

$$\hat{\mathbf{Q}}(j) = (1 - j)\hat{\mathbf{s}}_j + j\hat{\mathbf{q}}_{j+1/2}, 0 \leq j \leq 1 \tag{5.4}$$

where $\hat{\mathbf{s}}_i$ and $\hat{\mathbf{s}}_j$ are the midpoint quaternions of the lower side of the corresponding parallelogram in the image space as explained in the previous chapter; $\hat{\mathbf{q}}_{i+1/2}$ and $\hat{\mathbf{q}}_{j+1/2}$ are the same dual quaternion representing the position obtained from the subdivision scheme which is at the last instant, i.e. $i = 1$, $j = 1$, on both screw motions.

When using the tensor product scheme to generate the two-parameter motion of a cylinder, each of the screw motions has a characteristic curve on the cylindrical surface at the last instant, i.e. $\mathbf{P}^{i=1}(s)$ and $\mathbf{P}^{j=1}(s)$, being s the parameter in the rational Bézier representation of the cylinder Eq. 2.38. The characteristic curves can be obtained by the method described in chapter 4. Because normals of all the points on $\mathbf{P}^{i=1}(s)$ are perpendicular to $\mathbf{V}_i$ and normals of all the points on $\mathbf{P}^{j=1}(s)$ are perpendicular to $\mathbf{V}_j$, where $\mathbf{V}(\mathbf{V}_i, \mathbf{V}_j)$ is the velocity, so the normal of the intersection point of above two curves must be perpendicular to $\mathbf{V}$ and, thus, belongs to the swept surface of the cylindrical surface.

Now, as it was described in the previous chapter, the analytical expression of the characteristic curve can be obtained as the intersection of the two last intermediate planes in the de Casteljau algorithm of the cylindrical surface and of the two last intermediate planes in the de Casteljau algorithm of each tangent plane of the cylinder undergoing the screw motion. This statement applies to both screw motions; therefore, in order to find the intersection point, one must find the parameter or parameters s, where the set of four planes that gives origin to $\mathbf{P}^{i=1}(s)$ and the set of four planes that gives origin to $\mathbf{P}^{j=1}(s)$ are concurrent. This condition for concurrency can be expressed such that the determinant of the 4×4 matrix formed by the four homogeneous vectors representing any three of the four planes from one set and any one of the two non-overlapping planes from the other group vanishes. As shown by Xia and Ge [52] this results in a polynomial equation of degree 6 in the variable s that can be reduced to a quartic equation in s. The quartic equation might have 4, 2, or 0 real roots; there are 2 real roots, 1 real root, or 0 real root inside [0,1), respectively. After obtaining s, the intersection point can be obtained by plugging s into any of the equations of the characteristic curves $\mathbf{P}^{i=1}(s)$ or $\mathbf{P}^{j=1}(s)$. For a finite cylinder, the distance to a reference point on the cylinder must be checked in order to determine if the point belongs to the swept surface of the finite cylinder.

As for the top and bottom circular faces, each of their planes traces out a ruling line in the i and in the j direction. Therefore, the intersection of these two ruling lines is the point on the swept surface of an infinite plane at the position obtained from the subdivision scheme, which can be solved by finding the intersection point of any three of the four last intermediate planes in the de Casteljau algorithms from the screw motions in i and j directions of the corresponding plane. In order to check if this point lies on the swept surface of a finite cylinder one needs to compare the distance from the point to the center of the circular face with the radius of the circle.

In regards to the circular edges, the grazing points need to be determined by checking the circular arcs that fulfill the conditions 2.44. The quaternion derivative at each discrete position is determined by using Eq. 4.5 in the i and j direction. Then the normalization procedure described in subsection 2.3.2 is carried out to solve for the velocities $\mathbf{V}_i$ and $\mathbf{V}_j$, respectively.

5.3 Computer Implementation And Example

The procedure just developed for the discrete representation of the swept surface of a cylindrical tool of unit length undergoing a motion as prescribed by the tensor-product subdivision scheme 5.2 has been applied to the cutter location (CL) data displayed in Fig. 5.2

The contributions of the cylindrical face, the top plane, and the top circular edge are shown in Fig. 5.3.

For implementation purposes a data structure consisting of a matrix whose elements are objects, with private members containing the dual quaternions $\hat{\mathbf{s}}_i$, $\hat{\mathbf{s}}_j$, and $\hat{\mathbf{q}}_{i+1/2}$ that give origin to the representative screw motions at each discrete position represented by $\hat{\mathbf{q}}_{i+1/2}$, was designed. Appropriate operator overloading functions were implemented that are used during the subdivision.

5.4 Conclusions

A tensor-product subdivision scheme was extended to handle dual quaternions. The tensor-product dual quaternion surface in the image space has continuous

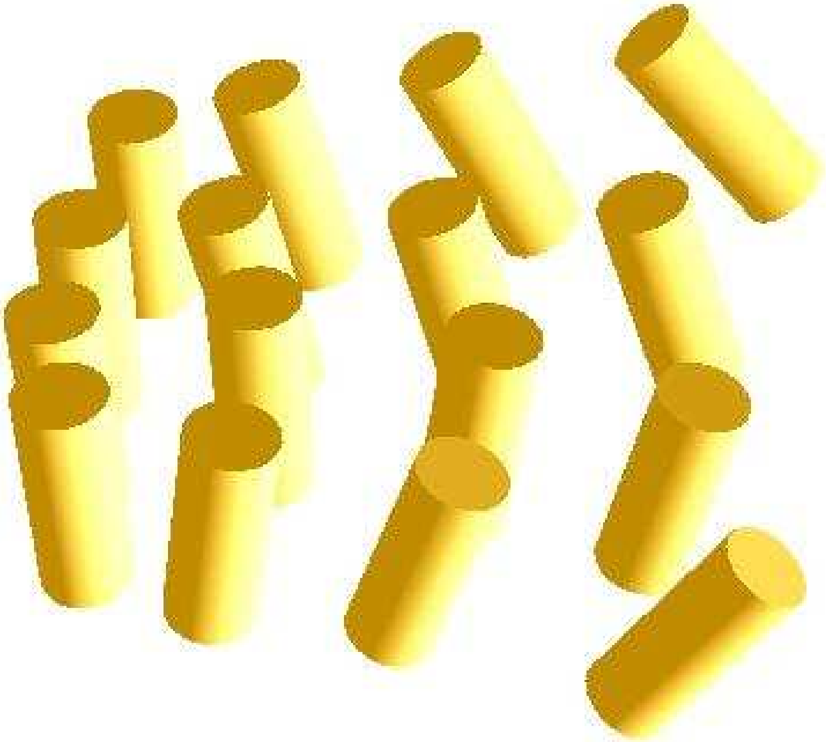

Figure 5.2: Initial positions of a cylindrical cutter

tangent plane; thus, the resulting two-parameter motion described by the rigid body is of C^1 continuity. The geometry of the subdivision scheme in the image space of spatial kinematics is exploited to generate a discrete representation of the swept surface of a cylindrical tool undergoing the motion prescribed by the scheme. The results have applications in CNC machining simulation, collision detection, and computer graphics.

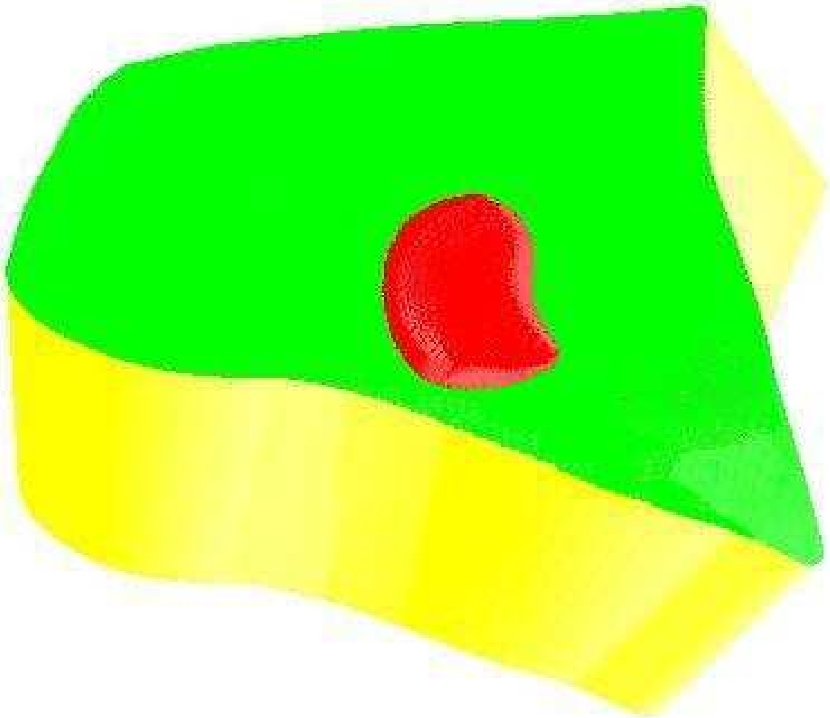

Figure 5.3: Swept surface generated by a cylindrical tool after 5 rounds of subdivision using a shape factor $\omega = 1/16$. The yellow surface is generated by the cylindrical face; the red surface is generated by the top plane; and the green surface is generated by the top circular edge.

Chapter 6

Mechanism Animation by Means of Subdivision

This chapter is concerned with the classical problem of position analysis of a planar four-bar linkage, also known as a planar 4R closed chain. The four-bar position analysis is such a well-solved problem that it is covered in every undergraduate text in the area of mechanism analysis and synthesis (see for example, [83, 84]). It involves the solution of a so-called loop closure equations, which is a pair of trigonometric equations relating angular positions of the output and coupler links to the angular position of the input link. The analytical solution can be obtained by reducing the loop closure equation to a quadratic equation.

In theoretical kinematics ([42], [43]), the motion of the coupler link of a planar 4R chain is studied as the constrained motion of a rigid body (the coupler) subject to the geometric constraints that two points (the moving pivots) of the body stay on two separate circles. Following this perspective, this chapter seeks to develop a method for generating positions of the coupler link of a planar 4R chain as a constrained subdivision problem. The main idea is to use planar quaternions to transform coupler positions into points in the space of planar quaternions. In this way, the coupler motion becomes a planar quaternion curve, which is obtained as the intersection of two constraint surfaces that correspond to the geometric constraints that two moving pivots stay on two separate circles. The four-point interpolatory subdivision scheme developed by Dyn, Levin and Gregory [29, 82] for curves in the field of Computer Aided Geometric Design (CAGD) is extended to planar quaternions for the generation of an inbetween candidate position of the coupler link from a given set of four key positions. The planar quaternion corresponding the candidate position is then checked and modified according to constraint surfaces of the planar 4R chain. The resulting new inbetween position that satisfies the kinematic constraints is then inserted into the given set of key positions. In the early stage of this refinement process, each new inbetween position must be made to satisfy the 4R kinematic constraints exactly to ensure the correct motion. When there is sufficient number of coupler positions, one can use the unconstrained four-point interpolatory scheme to generate the inbetween positions to allow for fast animation of the coupler motion.

The work presented in this paper is partly related to the recent work of Jin and Ge [85] that deals with the problem of synthesizing rational interpolating motions subject to the kinematic constraints of planar 2R and 3R open chains. Instead of formulating the problem using an analytic form of B-spline interpolation, this paper seeks to obtain inbetween positions of the coupler link based on a subdivision and refinement process. For computer aided synthesis of planar 4R chain, this paper offers a new approach for four-bar motion simulation by generating the simulation data for the coupler positions directly from a set of given positions used for rigid body guidance. The method presented in this chapter may also be used for generating additional coupler positions for task specification in computer aided synthesis of four-bar linkages.

The chapter is based on the paper by Trujillo and Ge [2] and is organized as follows. Firstly, a review of planar quaternions as well as the kinematics of planar 4R closed chains is presented. Then, the numerical geometry of the coupler motion is studied. A constrained subdivision scheme for generating inbetween positions of the coupler motion is developed by combing the four-point interpolatory subdivision scheme with the kinematic constraints of the coupler motion. Examples and discussion on the issue concerning how to adjust the shape parameter and the choice of the initial positions in the four-point interpolatory scheme are included. Finally, concluding remarks are stated.

6.1 Kinematics of Planar 4R Closed Chains

Planar quaternions are an elegant tool to represent general planar displacements involving both rotations and translations. In this section, a review of planar quaternions and their use in the kinematics analysis of the planar 4R closed chain is presented in so far as necessary for the development of this chapter. More details on the subject can be found in [43].

6.1.1 Planar Quaternions

Let d_1 and d_2 denote the coordinates of the origin **o** of the moving frame **M** with respect to the origin **O** of the fixed frame **F**. Let α be the angle between

M and **F** as depicted in Fig. 6.1. The planar displacement can be represented by a planar quaternion $\mathbf{Z} = Z_1\epsilon\mathbf{i} + Z_2\epsilon\mathbf{j} + Z_3\epsilon\mathbf{k} + Z_4$, where **i**, **j**, **k** and 1 form the quaternion basis and ϵ is the dual unit with the property $\epsilon^2 = 0$. The components of the planar quaternion $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$, are given by

$$\begin{aligned} Z_1 &= (d_1/2)\cos(\alpha/2) + (d_2/2)\sin(\alpha/2) \\ Z_2 &= -(d_1/2)\sin(\alpha/2) + (d_2/2)\cos(\alpha/2) \\ Z_3 &= \sin(\alpha/2) \\ Z_4 &= \cos(\alpha/2) \end{aligned} \tag{6.1}$$

These four components can be identified as a point in four dimensional space. The point **Z** is called the *image point* of a planar displacement. The set of image points that represent all planar displacements is called the *image space* of planar displacements. From Eqn. (6.1), the coordinates of a planar quaternion must satisfy the relationship

$$Z_3^2 + Z_4^2 = 1 \tag{6.2}$$

The composition of two planar displacements is obtained by the multiplication of the two planar quaternions, $\mathbf{G} = (G_1, G_2, G_3, G_4)$ and $\mathbf{H} = (H_1, H_2, H_3, H_4)$, representing each of the displacements and can be computed in matrix form as follows

$$\mathbf{GH} = [G^+]\mathbf{H} = [H^-]\mathbf{G} \tag{6.3}$$

where

$$[G^+] = \begin{bmatrix} G_4 & -G_3 & G_2 & G_1 \\ G_3 & G_4 & -G_1 & G_2 \\ 0 & 0 & G_4 & G_3 \\ 0 & 0 & -G_3 & G_4 \end{bmatrix} \tag{6.4}$$

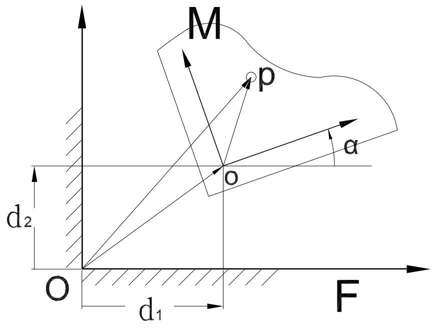

Figure 6.1: Planar displacement

and

$$[H^{-}] = \begin{bmatrix} H_4 & H_3 & -H_2 & H_1 \\ -H_3 & H_4 & H_1 & H_2 \\ 0 & 0 & H_4 & H_3 \\ 0 & 0 & -H_3 & H_4 \end{bmatrix} \tag{6.5}$$

The components of a planar quaternion are related to the homogeneous matrix of a planar displacement by

$$[A] = \frac{1}{Z_3^2 + Z_4^2} \begin{bmatrix} Z_4^2 - Z_3^2 & -2Z_3Z_4 & 2(Z_1Z_4 - Z_3Z_2) \\ 2Z_3Z_4 & Z_4^2 - Z_3^2 & 2(Z_1Z_3 + Z_4Z_2) \\ 0 & 0 & Z_4^2 + Z_3^2 \end{bmatrix} \tag{6.6}$$

Notice, as pointed out in [70], that the matrix [A] remains unaltered after multiplying each of the quaternion components by a nonzero scalar w, i.e., $\mathbf{Z}$ and $w\mathbf{Z}$ represent the same planar displacement. Therefore, Eqn. (6.2) may be transformed in the more general relationship

$$Z_3^2 + Z_4^2 = w^2 \tag{6.7}$$

This means that if a quaternion is obtained by an interpolation or approximation process, there is no need to normalize its coordinates in order to use it as a displacement operator, since any quaternion can be viewed as a homogeneous

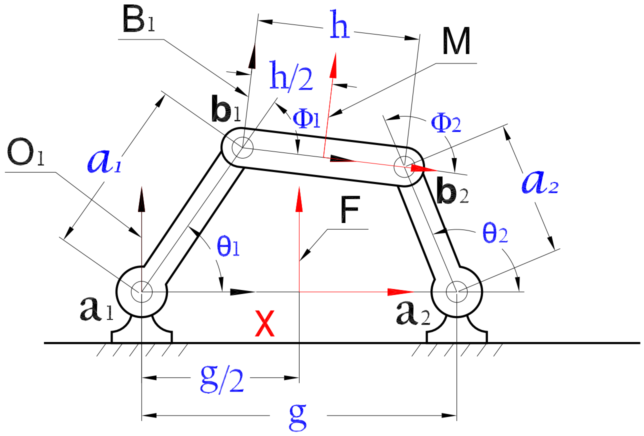

Figure 6.2: Planar 4R closed chain

equivalent of the quaternion that fulfills Eqn. (6.2) and the computations are performed using homogeneous coordinates.

6.1.2 Planar 4R Closed Kinematic Chain

A planar 4R closed chain is composed of a pair of planar 2R chains with their end links rigidly connected. For each of the 2R chains, the constraint manifold can be computed as a composition of the corresponding displacements from the moving frame **M** to the fixed frame **F**, as shown in Fig. 6.2. For instance, the transformation $\mathbf{Y}(\theta_1, \phi_1)$ of the coupler computed through the left side chain, with **M** attached to the midpoint of the coupler and **F** attached to the midpoint of the ground link, is composed of a translation by the planar quaternion $\mathbf{H}_1(h) = (h/4, 0, 0, 1)$, a rotation by $\mathbf{Z}(\phi_1) = (0, 0, \sin\phi_1/2, \cos\phi_1/2)$, a translation by $\mathbf{X}(a_1) = (a_1/2, 0, 0, 1)$, another rotation by $\mathbf{Z}(\theta_1) = (0, 0, \sin\theta_1/2, \cos\theta_1/2)$, and finally a translation by the planar quaternion $\mathbf{G}_1(g) = (-g/4, 0, 0, 1)$. Therefore,

$$\mathbf{Y}(\theta_1, \phi_1) = \mathbf{G}_1(g)\mathbf{Z}(\theta_1)\mathbf{X}(a_1)\mathbf{Z}(\phi_1)\mathbf{H}_1(h) \tag{6.8}$$

whose components are

$$
\begin{aligned}
Y_1(\theta_1, \phi_1) &= (a_1/2)\cos((\theta_1 - \phi_1)/2) - \tau\cos((\theta_1 + \phi_1)/2) \\
Y_2(\theta_1, \phi_1) &= (a_1/2)\sin((\theta_1 - \phi_1)/2) + \sigma\sin((\theta_1 + \phi_1)/2) \\
Y_3(\theta_1, \phi_1) &= \sin((\theta_1 + \phi_1)/2) \\
Y_4(\theta_1, \phi_1) &= \cos((\theta_1 + \phi_1)/2) \qquad (6.9)
\end{aligned}
$$

where

$$\tau = (g - h)/4, \quad \sigma = (g + h)/4 \qquad (6.10)$$

Similarly, the constraint surface for the 2R chain on the right side $\mathbf{Y}(\theta_2, \phi_2)$ can be obtained. Since both are transformations for the same points on $\mathbf{M}$ with respect to $\mathbf{F}$, then $\mathbf{Y}(\theta_1, \phi_1) = \mathbf{Y}(\theta_2, \phi_2) = \mathbf{Y} = (Y_1, Y_2, Y_3, Y_4)$. Eliminating θ_1 and ϕ_1 from Eqn. (6.9), and carrying out the same procedure with the components of $\mathbf{Y}(\theta_2, \phi_2)$, the equations of the constraint manifolds may be written as follows

$$\frac{(Y_1 + \tau Y_4)^2 + (Y_2 - \sigma Y_3)^2}{Y_3^2 + Y_4^2} = \frac{a_1^2}{4} \qquad (6.11)$$

$$\frac{(Y_1 - \tau Y_4)^2 + (Y_2 + \sigma Y_3)^2}{Y_3^2 + Y_4^2} = \frac{a_2^2}{4} \qquad (6.12)$$

The constraint manifolds can be viewed by projecting them onto the hyperplane $Y_4 = 1$. Therefore, the equations of the projected surfaces are

$$(Y_1 + \tau)^2 + (Y_2 - \sigma Y_3)^2 = \frac{a_1^2}{4}(1 + Y_3^2) \qquad (6.13)$$

$$(Y_1 - \tau)^2 + (Y_2 + \sigma Y_3)^2 = \frac{a_2^2}{4}(1 + Y_3^2) \qquad (6.14)$$

which are the equations of two hyperboloids. Hence, the kinematic constrain of the planar 4R chain, projected onto $Y_4 = 1$, is the intersection curve of this two hyperboloids which also recalls the fact that the planar 4R chain has one degree of freedom. Keeping Y_4 and Y_3 constant, Eqns. (6.11) and (6.12) become the equations of two circles. Therefore, the position of the coupler corresponding to that slice is any of the two possible intersections of the circles.

The constraint curve $\mathbf{Y} = (Y_1, Y_2, Y_3, Y_4)$ describing the motion of the coupler in the 4R closed chain can be parameterized by determining the angle ϕ_1 in terms of θ_1 and plugging it into Eqn. (6.9), yielding the components of the planar quaternions along the curve. The angle ϕ_1 is given by Eqn. (6.15), as follows

$$\phi_1(\theta_1) = 2\arctan\left(\frac{-B \pm \sqrt{\Delta}}{C - A}\right) \tag{6.15}$$

where

$$\begin{aligned} A &= 2ha_1 - 2gh\cos\theta_1 \\ B &= 2gh\sin\theta_1 \\ C &= g^2 + h^2 + a_1^2 - a_2^2 - 2a_1 g\cos\theta_1 \end{aligned} \tag{6.16}$$

Picking the positive or negative root in Eqn. (6.15) determines which of the two possible configurations of the linkage the curve represents.

6.2 Constrained Subdivision of Coupler Positions

The 4-point interpolatory subdivision algorithm described in chapter 4 can handle planar quaternions by considering these as regular points in $\mathbf{R}^4$ since planar quaternions are a singular case of dual quaternions. The first step is to convert a given set of ordered positions of the coupler into a set of ordered planar quaternions, $\mathbf{Z}_{2i}$ $(i = 0, \dots, n)$ using Eqns. (6.9) and (6.15). We then generate inbetween planar quaternions $\mathbf{Z}_{2i+1}$ $(i = 0, \dots, n-1)$ by applying the 4-point interpolatory subdivision using the four points, ($\mathbf{Z}_{2i-2}$, $\mathbf{Z}_{2i}$, $\mathbf{Z}_{2i+2}$, $\mathbf{Z}_{2i+4}$). In the case of a Grashof linkage, the kinematic constraint is a closed curve and the subdivision scheme can be applied continuously using the initial set of key positions. However, for non-Grashof linkages the kinematic constraint is an open curve. It should be noted here that Schrocker et al. devised a method to study the branch problem in 4R planar closed chains [86]. Basically, the method determines the tangential contact points of the circular cross sections of the constraint hyperboloids of the 2R open chains comprising the 4R closed chain. Such method is useful to identify the number and type of branches of a 4R linkage and the branch that a specific position of the coupler belongs to. As the scheme requires four points to generate a new inbetween point, special treatment is required for the generation of $\mathbf{Z}_1$ and $\mathbf{Z}_{2n-1}$. The first inbetween point $\mathbf{Z}_1$ is generated from the point set $(\mathbf{Z}_0, \mathbf{Z}_0, \mathbf{Z}_2, \mathbf{Z}_4)$, effectively counting the point $\mathbf{Z}_0$ twice. Similarly, $\mathbf{Z}_{2n-1}$ is generated from ($\mathbf{Z}_{2n-4}$, $\mathbf{Z}_{2n-2}$, $\mathbf{Z}_{2n}$, $\mathbf{Z}_{2n}$), effectively counting $\mathbf{Z}_{2n}$ twice. Another possibility of generating the missing points is by controlling the slope of the tangent of the curve at the end points as suggested by Dyn et al. Nonetheless, the first approach generates acceptable results as will be shown in the examples.

An inbetween point such as $\mathbf{Z}_{2i+1}$ in general does not satisfy the kinematic constraints of the coupler motion, i.e., it does not satisfy the equations (6.11) and (6.12) simultaneously. It has to be modified in order to meet the kinematic constraints. One way to modify $\mathbf{Z}_{2i+1}$ is to find its normal projection on the planar quaternion curve that represents the coupler motion of the planar 4R chain. The normal projection method was developed for quaternion based approach for four-bar linkage synthesis by Ravani and Roth [46] and refined by Boddulurri and McCarthy [87]. In this paper, we use a more efficient way of modifying the planar quaternion $\mathbf{Z}_{2i+1}$ by fixing the third and fourth components of the planar quaternion and then solving for the first two components from the constraint equations (6.11) and (6.12). This problem is essentially reduced to that of finding the intersection of two circles in a plane. Kinematically, this means that we are fixing the angular orientation of the coupler link while translating the link so that the kinematic constraints would be satisfied. The resulting new inbetween planar quaternion that satisfies the kinematic constraints is then inserted into the given set of planar quaternions. In the early stage of this refinement process, each new inbetween planar quaternion must be made to satisfy the 4R kinematic constraints exactly to ensure the correct motion. When there is a sufficient number of coupler positions, one can use the unconstrained four-point interpolatory scheme to generate the inbetween positions to allow for fast animation of the coupler motion.

We now turn our attention to the choice of the shape parameter w. The value of the shape parameter w can be adjusted as the one that would minimize the distance between the computed candidate position and the closest exact point on the constraint surfaces. This can be formulated as the minimization of the squared distance from the new point to the circular intersections for the kinematic constraints since the radii of these circles can be determined from Eqns. (6.11) and (6.12). However, after performing the aforementioned adjustment procedure and through experimentation, it has been observed that using four exact positions as the initial points to initiate the subdivision scheme, a value of w equal to 1/16, and after the first round of subdivision the unconstrained procedure generates acceptable and good-looking results.

6.3 Examples and Discussion

Figure 6.3 shows the intermediate positions obtained from eight initial key positions and a value of w of 1/16. The initial initial key positions were computed from the parameterized equation of the constraint curve as described in the previous subsection. This is a Grashof linkage of the type crank-rocker where its link lengths are: $a_1 = 100$, $a_2 = 200$ and $h = g = 300$. In Fig. 6.4 the quaternion curve generated by the subdivision algorithm is projected onto 3D space, dividing the vector components of the planar quaternions by their real component, and it is compared to the exact curve computed using the parameterized curve equation. In this case, the maximum deviation from the kinematic constraint is 2.69%, computed as the Euclidean distance from the interpolating quaternion to the corresponding exact quaternion divided by the magnitude of the latter. The exact quaternion representing the position on the parameterized curve is computed substituting the last two components of the interpolating quaternion in the equations of the circles for the corresponding slice, i. e., Eqns. (6.13) and (6.14), and simultaneously solving them. Figures 6.5 and 6.6 display the effect of $w = 1/8$. In this case the maximum

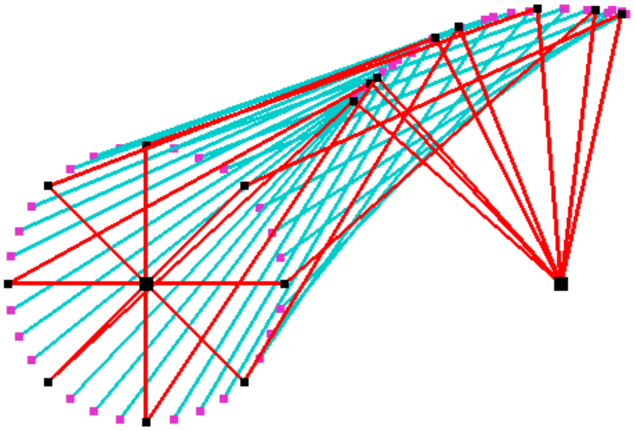

Figure 6.3: Discrete positions for planar 4R closed chain computed from 8 key frames after first round of subdivision, $w = 1/16$.

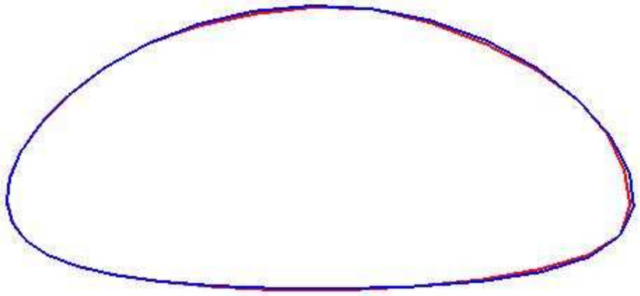

Figure 6.4: Quaternion curves projected onto 3-D space. Blue: exact curve, red: obtained by subdivision scheme, 8 key frames after first round of subdivision, $w = 1/16$.

deviation is 5.83% and the discrepancy of the motion is visually noticeable.

Figures 6.7 and 6.8 reproduce the simulation, and the comparison of the quaternion curves, based only on 4 initial key frames and $w = 1/16$. The maximum deviation is 9.56%. The simulation of a non-Grashof linkage of the type rocker-rocker is displayed in Fig. 6.9 and its corresponding quaternion curve comparison, Fig. 6.10. The simulation was achieved repeating the end-points as described before and the maximum deviation found was 1.86%.

Comparing the computational efficiency of the scheme to that of the exact

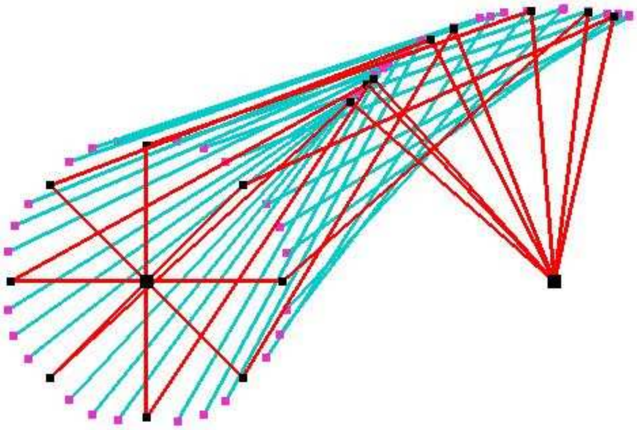

Figure 6.5: Discrete positions for planar 4R closed chain computed from 8 key frames after first round of subdivision, $w = 1/8$.

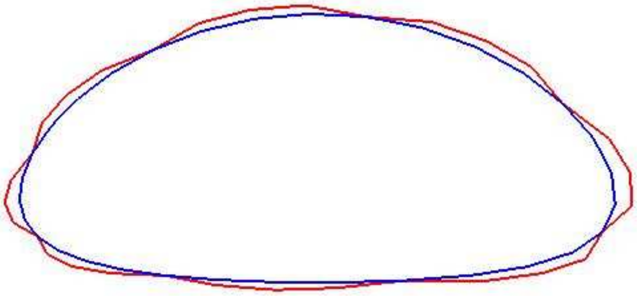

Figure 6.6: Quaternion curves projected onto 3-D space. Blue: exact curve, red: obtained by subdivision scheme, 8 key frames after first round of subdivision, $w = 1/8$.

simulation using Eqns. (6.9) and (6.15), it seems that the subdivision approach is faster since for the exact simulation trigonometric and inverse trigonometric functions are required which are computationally more expensive than the fewer multiplications in the subdivision algorithm. The difference in computing time between both methods increases as the number of required frames to display grows. For instance, if displaying 32 positions of the coupler, the subdivision scheme is three times faster than using the exact simulation and almost seven times faster when displaying 128 positions, according to experi-

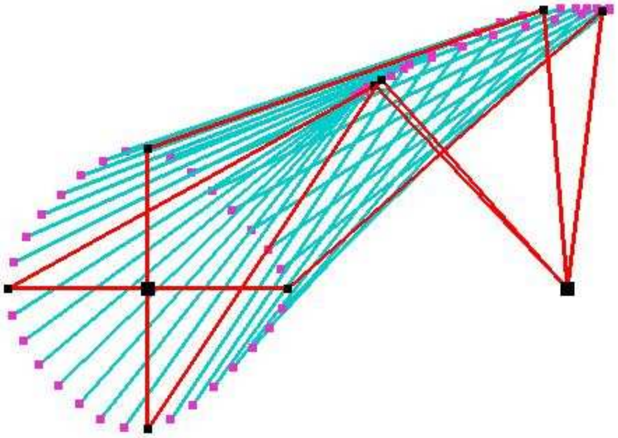

Figure 6.7: Discrete positions for planar 4R closed chain computed from 4 initial key frames without constraint check, $w = 1/16$.

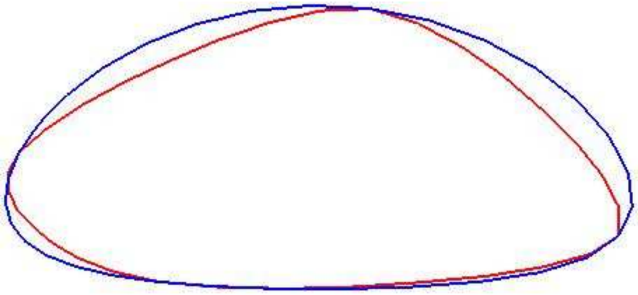

Figure 6.8: Quaternion curves projected onto 3-D space. Blue: exact curve, red: obtained by subdivision scheme, 4 key frames without constraint check, $w = 1/16$.

mentation.

6.4 Conclusions

The 4-point interpolatory subdivision scheme was readily extended to planar quaternions. This procedure can be used to efficiently generate accurate discrete positions of the coupler in a planar 4R closed chain. Computing four exact positions using the exact simulation by means of the parameterized curve and using a shape factor of 1/16 offer a very precise simulation

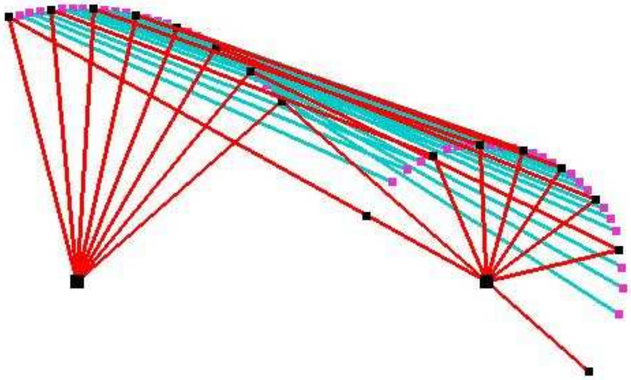

Figure 6.9: Discrete positions for non-grashof linkage, 8 key frames after first round of subdivision, $w = 1/16$.

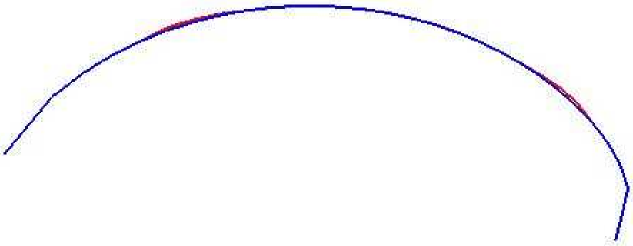

Figure 6.10: Quaternion curves projected onto 3-D space for non-grashof linkage. Blue: exact curve, red: obtained by subdivision scheme, 8 key frames after first round of subdivision, $w = 1/16$.

tool for Grashof and non-Grashof linkages. Furthermore, after only one round of subdivision the unconstrained scheme can be continuously applied since it generates very accurate simulations. The procedure utilizing the 4-point interpolatory subdivision algorithm possesses the interesting feature of generating fast simulations and animations within acceptable precision. Therefore, this type of motion simulation procedure can be implemented in applications where fast animation of the coupler and the whole linkage is desired; for instance, on-line four-bar linkage position analysis [88]. Moreover, the technique may have potential applications in robot path planning and CNC machining.

The extension of the approach to more complex kinematic chains where the constraint check becomes more challenging is an open issue.

Chapter 7

Concluding Remarks and Future Work

The work carried out along the course of this research has aimed to develop new approaches for design and analysis of motion suitable for CAD-CAM integration. In the hope that not only shape but also kinematic information should be conveyed in an efficient manner that can be easily implemented in CAD-CAM systems and eventually interpreted by CNC manufacturing equipment.

Towards this goal, a technique for orientation interpolation with angular velocity constraints based on quaternion biarcs was developed. The resulting b-spline quaternion curve represents a piecewise line-symmetric rational spherical motion with C^1 continuity. This attractive feature makes this approach for orientation interpolation more advantageous than current approaches formally used in 5-axis milling of sculptured surfaces. However, the actual deployment of the quaternion biarc technique into CNC machining can only be achieved through open architecture controllers.

The utilization of subdivision schemes for motion design seems to be a promising contribution. Their computational ease and numerical robustness together with their interesting geometric traits make subdivision schemes an attractive option for CAD-CAM applications. In this research the four point interpolatory subdivision scheme for curve generation was adapted to the interpolation of a given set of positions of a cylindrical tool represented by dual quaternions. It was shown that the resulting discrete model of the tool path lends itself naturally to an algorithm for computing the characteristic curve belonging to the boundary surface of the swept volume at each of the discrete positions as well as a discrete representation of the swept surface traced out by the cylinder. The tensor-product version of the four-point scheme was also utilized for the generation of the swept surface of a cylindrical cutter undergoing a two parameter motion. This is a very useful resource for tool path verification, machining simulation, and collision avoidance.

The application of subdivision schemes in mechanism simulation has also been explored. A preliminary attempt for the motion simulation of the coupler link of a planar 4R closed kinematic chain was successfully achieved. However, the use of subdivision schemes for simulation, task specification, and motion planning in more complex kinematic chains remains unexplored. The main challenge being the increased complexity of the kinematic constraints. This problem can be stated in a more general way as constrained subdivision. To this end first or even second order approximations of the kinematic constrains can be used in order to estimate the deviation of the positions generated by the subdivision scheme from the kinematic constrain. Should this approach prove successful it would pave the path for generalized utilization of subdivision in mechanism simulation applications.

Bibliography

[1] Trujillo, C. A., and Ge, Q. J., 2006. “Geometric construction of piecewise line-symmetric spherical motions using quaternion biarcs”. In Proceedings ASME-IDETC/CIE. CD-ROM.

[2] Trujillo, C. A., and Ge, Q. J., 2007. “Numerical geometry of the coupler motion of a planar 4R closed kinematic chain”. In Proceedings ASME-IDETC/CIE. CD-ROM.

[3] Trujillo, C. A., and Ge, Q. J., 2008. “A subdivision scheme for motion generation and swept volume analysis”. In Proceedings ASME-IDETC/CIE. CD-ROM.

[4] Farin, G., 1996. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 4^{th} ed. Academic Press, San Diego, CA, USA.

[5] Farin, G., Hoschek, J., and Kim, M.-S., eds., 2002. *Handbook of Computer Aided Geometric Design*. Elsevier Science, North Holland.

[6] Piegl, L., 1993. *Fundamental Developments of Computer Aided Geometric Modeling*. Academic Press, San Diego, CA, USA.

[7] Piegl, L., and Tiller, W., 1995. *The NURBS Book*. Springer, Berlin.

[8] Piegl, L., 1991. “On nurbs - a survey”. *IEEE Computer Graphics and Applications,* **11**(1), pp. 55–71.

[9] Marciniak, K., 1991. *Geometric Modelling for Numerically Controlled Machining*. Oxford University Press, New York.

[10] Klass, R., and Schramm, P., 1991. “Numerically-controlled milling of CAD surface data”. In *Geometric Modeling: Methods and Applications*, H. Hagen and D. Roller, eds. Springer-Verlag, pp. 213–225.

[11] Thyer, G. E., 1993. *Computer numerical control of machine tools*, 2^{nd} ed. Industrial Press, New York.

[12] Sarma, R., and Rao, A., 2000. “Discretizors and interpolators for five-axis CNC machines”. *ASME Journal of Manufacturing Science and Engineering,* **122**(1), pp. 191–197.

[13] Liang, H., Hong, H., and Svoboda, J., 2002. “A combined 3-d linear and circular interpolation technique for multi-axis CNC machining”. *ASME Journal of Manufacturing Science and Engineering,* **124**(2), pp. 305–312.

[14] Koren, Y., and Lin, R.-S., 1995. “Five-axis surface interpolators”. *Annals of the CIRP,* **44**(1), pp. 379–382.

[15] Shoemake, K., 1985. “Animating rotations with quaternion curves”. *ACM SIGGRAPH Computer Graphics,* **19**(3), pp. 245–254.

[16] Wang, W., and Joe, B., 1993. “Orientation interpolation in quaternion space using spherical biarcs”. *Proceedings Graphics Interface’93,* pp. 24–31.

[17] Barr, A. H., Currin, B., Gabriel, S., and Hughes, J. F., 1992. “Smooth interpolation of orientations with angular velocity constraints using quaternions”. *Computer Graphics,* **26**(2), pp. 313–320.

[18] Ramamoorthi, R., and Barr, A. H., 1997. “Fast construction of accurate quaternion splines”. In SIGGRAPH ’97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 287–292.

[19] Jüttler, B., 1994. “Visualization of moving objects using dual quaternion curves”. *Computers and Graphics,* **18**(3), pp. 315–326.

[20] Jüttler, B., and Wagner, M. G., 1996. “Computer-aided design with spatial rational b-spline motions”. *ASME Journal of Mechanical Design,* **118**(2), pp. 193–201.

[21] Ge, Q. J., and Ravani, B., 1994. “Geometric construction of Bézier motions”. *ASME Journal of Mechanical Design,* **116**(3), pp. 749–755.

[22] Ge, Q. J., and Ravani, B., 1994. “Computer aided geometric design of motion interpolants”. *ASME Journal of Mechanical Design,* **116**(3), pp. 756–762.

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

[24] DeRose, T., Kass, M., and Truong, T., 1998. “Subdivision surfaces in character animation”. In SIGGRAPH ’98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM, pp. 85–94.

[25] Chaikin, G. M., 1974. "An algorithm for high speed curve generation". *Computer Graphics and Image Processing,* **3**, pp. 346–349.

[26] Catmull, E., and Clark, J., 1978. "Recursively generated b-spline surfaces on arbitrary topological meshes". *Computer-Aided Design,* **10**(6), pp. 350– 355.

[27] Doo, D., and Sabin, M., 1978. "Behaviour of recursive division surfaces near extraordinary points". *Computer-Aided Design,* **10**(6), pp. 356–360.

[28] Dyn, N., Levin, D., and Gregory, J. A., 1990. "A butterfly subdivision scheme for surface interpolation with tension control". *ACM Transactions on Graphics,* **9**(2), pp. 160–169.

[29] Dyn, N., Levin, D., and Gregory, J. A., 1987. "A 4-point interpolatory subdivision scheme for curve design". *Computer Aided Geometric Design,* **4**(4), pp. 257–268.

[30] Warren, J., and Weimer, H., 2002. *Subdivision Methods for Geometric Design: A Constructive Approach.* Morgan Kaufmann, San Francisco.

[31] Hofer, M., Pottmann, H., and Ravani, B., 2002. "Subdivision algorithms for motion design based on homologous points". In *Advances in Robot Kinematics*, J. Lenarčič and F. Thomas, eds. Kluwer Academic Publ., pp. 235–244.

[32] Hofer, M., Pottmann, H., and Ravani, B., 2004. "From curve design algorithms to the design of rigid body motions". *The Visual Computer,* **20**(5), pp. 279–297.

[33] Wallner, J., and Pottmann, H., 2006. "Intrinsic subdivision with smooth limits for graphics and animation". *ACM Trans. Graphics,* **25**(2), pp. 356–374.

[34] Ge, Q. J., 1996. "Kinematics-driven geometric modeling: a framework for simultaneous nc tool-path generation and sculptured surface design". In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, Vol. 2, pp. 1819–1824.

[35] Pritschow, G., Altintas, Y., Jovane, F., Koren, Y., Mitsuishi, M., Takata, S., van Brussel, H., Weck, M., and Yamazaki, K., 2001. "Open controller architecture past, present and future". *CIRP Annals - Manufacturing Technology,* **50**(2), pp. 463–470.

[36] Rober, S., and Shin, Y., 1995. "Modeling and control of cnc machines using a pc-based open architecture controller". *Mechatronics,* **5**(4), pp. 401–420.

[37] Van Hook, T., 1986. "Real-time shaded nc milling display". *SIGGRAPH Comput. Graph.,* **20**(4), pp. 15–20.

[38] Saito, T., and Takahashi, T., 1991. "Nc machining with g-buffer method". In SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques, ACM, pp. 207–216.

[39] Hamilton, W. R., 1847. "On quaternions". *Proceedings of the Royal Irish Academy,* **3**, pp. 1–16.

[40] Hamilton, W. R., 1899. *Elements of Quaternions, Volume I*, 2^{nd} ed. Longmans, Green, and Co., London.

[41] Hamilton, W. R., 1901. *Elements of Quaternions, Volume II*, 2^{nd} ed. Longmans, Green, and Co., London.

[42] Bottema, O., and Roth, B., 1990. *Theoretical Kinematics.* Dover Publications, New York.

[43] McCarthy, J. M., 1990. *An Introduction to Theoretical Kinematics.* The MIT Press, Cambridge, MA.

[44] Clifford, W. K., 1873. "Preliminary sketch of biquaternions". *Proceedings of the London Mathematical Society,* **4**, pp. 381–395.

[45] Dai, J. S., 2006. "An historical review of the theoretical development of rigid body displacements from rodrigues parameters to the finite twist". *Mechanism and Machine Theory,* **41**(1), pp. 41–528.

[46] Ravani, B., and Roth, B., 1983. "Motion synthesis using kinematic mappings". *ASME Journal of Mechanisms, Transmissions and Automation in Design,* **105**(3), pp. 460–467.

[47] Ravani, B., and Roth, B., 1984. "Mappings of spatial kinematics". *ASME Journal of Mechanisms, Transmissions, and Automation in Design,* **106**(3), pp. 341–347.

[48] Ge, Q. J., and Sirchia, M., 1999. "Computer aided geometric design of two-parameter freeform motions". *ASME Journal of Mechanical Design,* **121**(4), pp. 502–506.

[49] Ravani, B., and Ge, Q. J., 1993. “Computation of spatial displacements from geometric features”. *ASME Journal of Mechanical Design,* **115**(1), pp. 95–102.

[50] Xia, J., and Ge, Q. J., 2001. “On the exact representation of the boundary surfaces of the swept volume of a cylinder undergoing rational Bézier and b-spline motions”. *ASME Journal of Mechanical Design,* **123**(2), pp. 261–265.

[51] Li, S., and Ge, Q. J., 2005. “Rational Bézier line-symmmetric motions”. *ASME Journal of Mechanical Design,* **127**(2), pp. 222–226.

[52] Xia, J., and Ge, Q. J., 2000. “On the exact computation of the swept surface of a cylindrical surface undergoing two-parameter rational Bézier motions”. In Proceedings ASME Design Automation Conference. DETC2000/DAC-14039.

[53] Xia, J., 2001. “Motion Based Geometric Modeling”. PhD Thesis, Stony Brook University, Stony Brook, NY, Dec.

[54] Ge, Q. J., Kang, D., and Sirchia, M., 1998. “Kinematically generated dual tensor-product Bézier surfaces”. In Proceedings 1998 ASME Mechanisms Conference, DETC98/MECH-5977.

[55] Bodduluri, R. M. C., and Ravani, B., 1993. “Design of developable surfaces using duality between plane and point geometries”. *Computer-Aided Design,* **25**(10), pp. 621–632.

[56] Flanders, H., 1989. *Differential Forms with Applications to the Physical Sciences.* Dover Publications, Inc, New York.

[57] Hu, Z.-J., and Ling, Z.-K., 1996. "Swept volumes generated by the natural quadric surfaces". *Computers & Graphics,* **20**(2), pp. 263–274.

[58] Chiou, C.-J., and Lee, Y.-S., 2002. "Swept surface determination for five-axis numerical control machining". *International Journal of Machine Tools & Manufacture,* **42**(14), pp. 1497–1507.

[59] Piegl, L., and Tiller, W., 2002. "Data approximation using biarcs". *Engineering with Computers,* **18**, pp. 59–65.

[60] Latombe, J. C., 1991. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA.

[61] Lozano-Perez, T., 1983. "Spatial planning: A configuration space approach". *IEEE Transactions on Computers,* **32**(2), pp. 108–120.

[62] Schwarzer, F., Saha, M., and Latombe, J., 2004. "Exact collision checking of robot paths". In *Algorithmic Foundations of Robotics V*, J. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, eds. Springer, pp. 25–41.

[63] Gasparetto, A., and Zanotto, V., 2007. "A new method for smooth trajectory planning of robot manipulators". *Mechanism and Machine Theory,* **42**(4), pp. 455 – 471.

[64] Chou, J.-J., and Yang, D. C. H., 1992. "On the generation of coordinated motion of five-axis cnc/cmm machines". *Journal of Engineering for Industry,* **114**(1), pp. 15–22.

[65] Morishige, K., Kase, K., and Takeuchi, Y., 1997. "Collision-free tool path generation using 2-dimensional c-space for 5-axis control machining". *The International Journal of Advanced Manufacturing Technology,* **13**(6), pp. 393–400.

[66] Jun, C.-S., Cha, K., and Lee, Y.-S., 2003. "Optimizing tool orientations for 5-axis machining by configuration-space search method". *Computer-Aided Design,* **35**(6), pp. 549–566.

[67] Tsai, M.-C., Cheng, M.-Y., Lin, K.-F., and Tsai, N.-C., 2005. "On acceleration/deceleration before interpolation for cnc motion control". *Mechatronics, 2005. ICM '05. IEEE International Conference on*, July, pp. 382–387.

[68] Feng-yun, L., and Tian-sheng, L., 2005. "Development of a robot system for complex surfaces polishing based on cl data". *The International Journal of Advanced Manufacturing Technology,* **26**(9), pp. 1132–1137.

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

[70] Purwar, A., and Ge, Q. J., 2005. "On the effect of dual weights in computer aided design of rational motions". *ASME Journal of Mechanical Design,* **127**(5), pp. 967–972.

[71] ISO, 1982. ISO 6983-1:1982 - Numerical control of machines – Program format and definition of address words – Part 1: Data format for positioning, line motion and contouring control systems.

[72] Altintas, Y., 2000. *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. Cambridge University Press.

[73] McCarthy, J. M., 1987. "The differential geometry of curves in an image space of spherical kinematics". *Mechanism and Machine Theory,* **22**(3), pp. 205–211.

[74] Rossignac, J., and Requicha, A., 1987. "Piecewise circular curves for geometric modeling". *IBM Journal of Research and Development,* **31**(3), pp. 296–313.

[75] Zhang, W., Zhang, Y. F., and Ge, Q. J., 2005. "Interference-free tool path generation for 5-axis sculputured surface machining using rational Bézier motions of a flat-end cutter". *International Journal of Production Research,* **43**(19), pp. 4103–4124.

[76] Akima, H., 1970. "A new method of interpolation and smooth curve fitting based on local procedures". *Journal of the ACM,* **17**(4), pp. 589–602.

[77] Arney, T., 2007. "Dynamic path planning and execution using b-splines". *Information and Automation for Sustainability, 2007. ICIAFS 2007. Third International Conference on*, Dec., pp. 1–6.

[78] Wang, K. K., and Wang, W. P., 1986. "Geometric modeling for swept volume of moving solids". *IEEE Computer Graphics and Applications,* **6**(12), pp. 8–17.

[79] Abdel-Malek, K., and Yeh, H.-J., 1997. "Geometric representation of the swept volume using jacobian rank-deficiency conditions". *Computer-Aided Design,* **29**(6), pp. 457–468.

[80] Blackmore, D., and Leu, M. C., 1992. "Analysis of swept volume via Lie groups and differential equations". *The International Journal of Robotics Research,* **11**(6), pp. 516–537.

[81] Abdel-Malek, K., Yang, J., Blackmore, D., and Joy, K., 2006. "Swept volumes: Fundation, perspectives, and applications". *International Journal of Shape Modeling,* **12**(1), pp. 87–127.

[82] Pottmann, H., and Wallner, J., 2001. *Computational Line Geometry.* Springer, Berlin.

[83] Erdman, A., and Sandor, G., 1996. *Mechanism Design: Analysis and Synthesis, Volume 1*, 3^{rd} ed. Prentice-Hall, Upper Saddle River, NJ.

[84] Sandor, G., and Erdman, A., 1984. *Advanced Mechanism Design: Analysis and Synthesis, Volume 2.* Prentice-Hall International, London.

[85] Jin, Z., and Ge, Q., 2006. "Computer aided synthesis of piecewise rational motions for planar 2R and 3R robot arms". In Proceedings ASME-IDETC/CIE. CD-ROM.

[86] Schröcker, H. P., Manfred, H., and McCarthy, J. M., 2005. “Kinematic mapping based evaluation of assembly modes for planar four-bar synthesis”. In Proceedings 2005 ASME-DETC, MECH-85037. CD-ROM.

[87] Bodduluri, R. M. C., and McCarthy, J. M., 1992. “Finite position synthesis using image curve of a spherical four-bar motion”. *ASME Journal of Mechanical Design,* **114**(1), pp. 55–60.

[88] Softintegration, 2007. Web-based mechanism design and analysis. See URL `http://www.softintegration.com`, Feb.

Appendices

Appendix A

Dual Velocity

The dual velocity $\hat{\mathbf{V}} = \mathbf{v} + \varepsilon \mathbf{v}^0$ is a vector dual quaternion representing the velocity distribution of a rigid body motion with respect to the instantaneous screw axis of the motion. Its real part $\mathbf{v}$ is the angular velocity about the instantaneous screw axis of the motion and the dual part $\mathbf{v}^0$ is the translational velocity of any point on and along the screw axis. In this appendix we explain how to compute the instantaneous dual velocity of a motion prescribed by a generalized homogeneous dual-quaternion function $\hat{\mathbf{Q}}(t)$, i.e. $\hat{\mathbf{Q}}(t)$ is not necessarily a unit dual-quaternion valued function, in the image space and how to compute the velocity of a point on a rigid body which is undergoing the motion described by $\hat{\mathbf{Q}}(t)$.

A.1 Computation of the Dual Velocity

Let $\hat{\mathbf{Q}}(t) = \hat{w}(t)\hat{\mathbf{q}}(t)$ be a generalized homogenous dual-quaternion valued function of t in the image space where $\hat{w}(t) = (\hat{\mathbf{Q}}(t)\hat{\mathbf{Q}}^*(t))^{1/2} = w + \varepsilon w^0$ is the normalizing factor and $\hat{\mathbf{q}}(t)$ are the equivalent unit dual-quaternion coordinates.Then

$$\hat{\mathbf{q}} = \frac{\hat{\mathbf{Q}}}{\hat{w}} \tag{A.1}$$

and the unit dual quaternion derivative $\dot{\hat{\mathbf{q}}}$ can be obtained as

$$\dot{\hat{\mathbf{q}}} = \frac{\hat{w}\dot{\hat{\mathbf{Q}}} - \dot{\hat{w}}\hat{\mathbf{Q}}}{\hat{w}^2} \tag{A.2}$$

where

$$\dot{\hat{w}} = (1/2)\frac{\dot{\hat{\mathbf{Q}}}\hat{\mathbf{Q}}^* + \hat{\mathbf{Q}}\dot{\hat{\mathbf{Q}}}^*}{(\hat{\mathbf{Q}}\hat{\mathbf{Q}}^*)^{1/2}} = (1/2)\frac{\dot{\hat{\mathbf{Q}}}\hat{\mathbf{Q}}^* + \hat{\mathbf{Q}}\dot{\hat{\mathbf{Q}}}^*}{\hat{w}} \tag{A.3}$$

The dual velocity $\hat{\mathbf{V}}$ is [42]

$$\hat{\mathbf{V}} = 2\dot{\hat{\mathbf{q}}}\hat{\mathbf{q}}^* \tag{A.4}$$

Plugging Eq. A.2 and $\hat{\mathbf{q}}^* = \hat{\mathbf{Q}}^*/\hat{w}$ into Eq. A.4, we obtain

$$\hat{\mathbf{V}} = 2\left[\frac{\dot{\hat{\mathbf{Q}}}\hat{\mathbf{Q}}^*}{\hat{w}^2} - \frac{\dot{\hat{w}}}{w}\right] \tag{A.5}$$

Therefore, the dual velocity can be obtained in terms of the homogeneous coordinates $\hat{\mathbf{Q}}$, $\hat{\mathbf{Q}}^*$, and their corresponding derivatives $\dot{\hat{\mathbf{Q}}}$ and $\dot{\hat{\mathbf{Q}}}^*$.

The dual velocity can be normalized as any dual quaternion in order to obtain the instantaneous screw axis of the motion and the dual magnitude of the dual velocity. Then $\hat{\mathbf{V}}$ can be written as follows

$$\hat{\mathbf{V}} = (v + \varepsilon v^0)(\mathbf{l} + \varepsilon\mathbf{l}^0) \tag{A.6}$$

where $\hat{v} = v + \varepsilon v^0$ is the magnitude of the dual velocity and $\hat{\mathbf{l}} = \mathbf{l} + \varepsilon\mathbf{l}^0$ is the instantaneous screw axis of the motion.

A.2 Computation of the Velocity of a Point

After obtaining the dual velocity at a specific instant for a motion, the velocity $\mathbf{V}_p$ of any point $\mathbf{p}$ on a rigid body undergoing such a motion can be computed as follows

$$\mathbf{V}_p = \mathbf{v}^0 + \mathbf{v} \times (\mathbf{p} - \mathbf{l} \times \mathbf{l}^0) \tag{A.7}$$

where $\mathbf{l}$ and $\mathbf{l}^0$ are the Plücker vector of the screw axis that can be obtained from the dual velocity vector as previously explained.