1

Kinematics and Animation

B. Jüttler[a] and M.G. Wagner[b]

[a]Institut für Analysis und Numerik, Johannes Kepler Universität Linz
Altenberger Str. 69, 4040 Linz, Austria

[b]Dept. of Computer Science and Engineering, Arizona State University
Tempe, AZ 85287-5406, USA

This chapter demonstrates that the techniques of Computer Aided Geometric Design can be generalized to Kinematics, Computer Animation, and Robotics. Our approach relies on spatial rational spline motions which can be seen as the kinematical analogue of rational spline curves. The potential applications include keyframe interpolation in Computer Graphics, motion planning in Robotics, and sweep surface modelling in Geometric Design.

1. INTRODUCTION

The idea to use the powerful tools of Computer Aided Geometric Design in spatial kinematics originated in Computer Graphics, where rigid body motions are needed for visualizing moving objects in Computer Animation (keyframe interpolation), and for generating smooth camera motions, e.g., in Virtual Reality. Initially, the Bézier technique was generalized to the unit quaternion sphere via 'slerping' (see Section 4.1), following ideas by Shoemake and others [33,36,44]. This technique generates unit quaternion curves which can be identified with spherical motions, thus representing the rotational part of a rigid body motion. Although this and similar generalizations seem to work well, and are apparently still in use in Computer Graphics [14], it was soon realized that these spherical generalizations of the standard algorithms lead to major difficulties, such as the absence of a subdivision property, non–linear interpolation conditions, the difficult parametric representation of the resulting point trajectories, and problems with the construction of C^2 (acceleration continuous) motions, see Section 4.2 for more.

Independently, a similar approach was developed by Ge and Ravani in Robotics, for designing robot motions via Bézier type curves in dual quaternion space [10]. Unlike the slerp techniques, this approach produces motions with rational point trajectories, the so–called *rational motions.* In kinematical geometry, these motions had been studied since the end of the 19th century [5,41,50].

Another source of the theory of rational motions can be identified in the discussion of sweeping (kinematical) surfaces. Sweeping is a very intuitive technique for generating free–form surfaces, by moving a (rigid or possibly evolving) profile curve through space, see Figure 12 of Bézier's preface to [8]. One of the first publications on rational sweeping surfaces is due to Röschel [31,42].

Computational techniques for rational spline motions have been further explored in the simultaneous Ph.D. theses of the two authors [19,45]. These motions have now been developed into a useful tool for geometric motion design and for applications in Robotics, Computer Graphics, and Geometric Modelling. Some of the results have been gathered in this chapter, which is organized as follows. The next two sections summarize fundamentals from spatial kinematics and on quaternions. Section 4 is devoted to the various non–rational techniques for motion design, using curves on the unit quaternion sphere. After introducing spherical rational motions (Section 5), we give an outline of available algorithms for spatial rational motions, along with a brief discussion of several applications. Finally we conclude this chapter and suggest some directions for further research.

2. THE KINEMATICAL MAPPING

This section collects some facts about the description of rigid body motions by homogeneous 4 × 4 matrices. We use Euler parameters to represent rotation matrices, leading directly to the kinematical mapping of spherical kinematics.

2.1. Coordinates

In the sequel we describe the points $\mathbf{p}$ in 3–space with the help of homogeneous coordinates $\mathbf{p} = (p_0, p_1, p_2, p_3)^\top \in \mathbb{R}^4 \setminus \{(0,0,0,0)^\top\}$. If the 0-th component satisfies $p_0 \neq 0$, we may obtain the corresponding Cartesian coordinates $\underline{\mathbf{p}} = (\underline{p}_1, \underline{p}_2, \underline{p}_3)^\top \in \mathbb{R}^3$ of the very point $\mathbf{p}$ from $\underline{p}_i = p_i/p_0$, where $i = 1, 2, 3$. The homogeneous coordinate vectors $\mathbf{p}$ and $\lambda\mathbf{p}$ describe the same point for any constant factor $\lambda \neq 0$. Consequently, the set of points in 3–space, which is projectively closed by adding points at infinity, is identified with the set of all one–dimensional subspaces in $\mathbb{R}^4$.

The coordinate p_0 of $\mathbf{p}$ is commonly referred to as the homogenizing coordinate or the *weight* of $\mathbf{p}$. Points with $p_0 = 0$ correspond to points at infinity; they can be identified with the ∞^2 equivalence classes of parallel lines. For further information on homogeneous coordinates see Chapter **?**.

2.2. Motions of a rigid body

Let us consider two coordinate systems in three dimensional Euclidean space, the *fixed* coordinate system E^3 ("world coordinates") and the *moving* coordinate system $\hat{E}^3$. Points can be described in either coordinate system. We denote the fixed coordinates of a point by $\mathbf{p}$ or $\underline{\mathbf{p}}$, and the moving coordinates by $\hat{\mathbf{p}}$ or $\underline{\hat{\mathbf{p}}}$, respectively. In order to convert moving coordinates into fixed coordinates we have to apply the coordinate transformation that maps $\hat{E}^3$ onto E^3. Using homogeneous coordinates, this coordinate transformation can be represented by a 4 × 4 matrix of the form

$$M = \left[\begin{array}{c|ccc} m_{0,0} & 0 & 0 & 0 \\ \hline m_{1,0} & m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,0} & m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,0} & m_{3,1} & m_{3,2} & m_{3,3} \end{array}\right], \text{ with } m_{0,0} \neq 0, \tag{1}$$

such that $\hat{\mathbf{p}} \mapsto \mathbf{p} = M\hat{\mathbf{p}}$. The homogeneous resp. Cartesian coordinate vectors

$$\mathbf{v} = M\,(1,0,0,0)^\top = (m_{0,0}, m_{1,0}, m_{2,0}, m_{3,0})^\top \quad \text{resp.} \quad \underline{\mathbf{v}} = (\frac{m_{1,0}}{m_{0,0}}, \frac{m_{2,0}}{m_{0,0}}, \frac{m_{3,0}}{m_{0,0}})^\top \tag{2}$$

describe the *position of the origin* in $\hat{E}^3$ with respect to the fixed coordinate system E^3. The 3×3 matrix $\underline{R}$,

$$\underline{R} = \frac{1}{m_{0,0}} \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{bmatrix}, \tag{3}$$

describes the *orientation* of the moving coordinate system $\hat{E}^3$. It is a *special orthogonal matrix.*. That is, it satisfies the orthogonality condition $\underline{R}\,\underline{R}^\top = I$ where I denotes the 3×3 identity matrix, and $\det(\underline{R}) = 1$.

If the matrix $M = M(t)$ depends on the time t, where t varies in some interval $[t_0, t_1]$, then we speak of a *rigid body motion*, cf. Figure 1, left. For any point $\underline{\mathbf{p}} \in \hat{E}^3$ of the moving system $\hat{E}^3$ we obtain its trajectory from

$$\hat{E}^3 \times [t_0, t_1] \;\rightarrow\; E^3 : \quad (\hat{\mathbf{p}}, t) \;\mapsto\; \mathbf{p}(t) = M(t)\,\hat{\mathbf{p}} \tag{4}$$

where $M(t)$ is of the form (1) with time–dependent components $m_{i,j}(t)$.

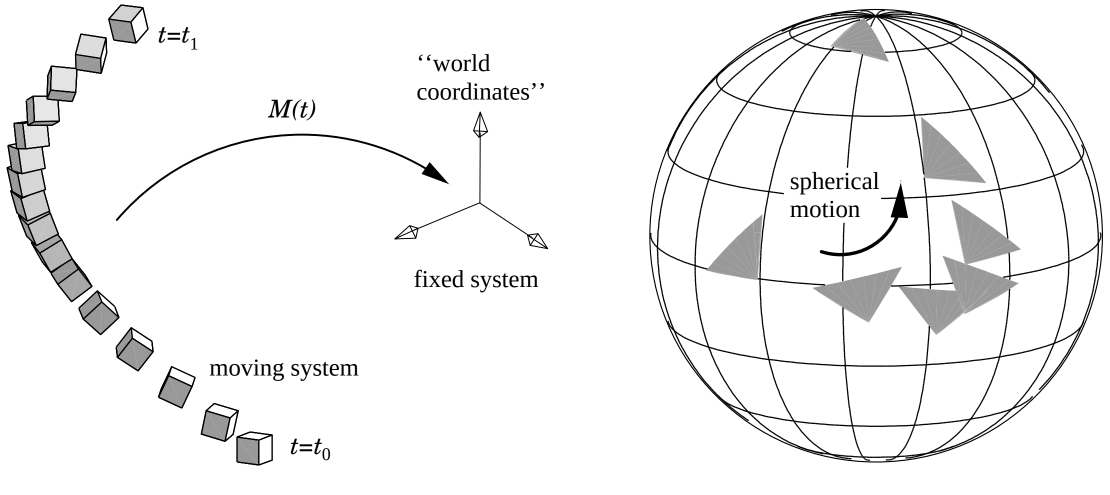


Figure 1. Spatial motion of a rigid body (left) and the associated spherical motion (right).

Obviously, the trajectory of the origin in homogeneous and Cartesian coordinates is given by $\mathbf{v}(t)$ and $\underline{\mathbf{v}}(t)$, see (2).

One may also describe the motion directly in Cartesian coordinates, which leads to

$$\hat{E}^3 \times [t_0, t_1] \;\rightarrow\; E^3 : \quad (\underline{\hat{\mathbf{p}}}, t) \;\mapsto\; \underline{\mathbf{v}}(t) + \underline{R}(t)\,\underline{\hat{\mathbf{p}}} \tag{5}$$

where $\underline{\mathbf{v}}(t)$ are the Cartesian coordinates of the trajectory of the origin (2).

The associated rotational (or spherical) part of the motion is described by the special orthogonal matrix $\underline{R} = \underline{R}(t)$, see (3). If the trajectory of the origin is replaced with the null vector, $\underline{\mathbf{v}}(t) \equiv (0,0,0)^\top$, then the trajectory $\mathbf{p}(t)$ of any point $\hat{\mathbf{p}}$ lies on a sphere of radius $\|\underline{\hat{\mathbf{p}}}\|$, centered at the origin. Consequently, the rotational part $\underline{R}(t)$ of $M(t)$ describes an intrinsic motion of the unit sphere, see Figure 1, right. It will be called the *associated spherical motion* of the rigid body motion $M(t)$. In the figure, it is visualized by several positions of a moving triangle on the unit sphere.

2.3. Euler parameters

When designing motions we encounter the problem that the rotational part $\underline{R}(t)$ of $M(t)$ has to satisfy the orthogonality conditions. Consequently it is not possible to simply prescribe the functions $m_{i,j}(t)$ since the resulting motion would in general not preserve the rigidity of the object; it would not be Euclidean. In order to resolve this problem, we will describe $\underline{R}$ by a set of independent parameters. There exist a number of different approaches. A well–known set of parameters is based on a classical result of Euler: any special orthogonal 3×3 matrix $\underline{R}$ can be written as

$$\underline{R} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{6}$$

where the q_i satisfy

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \tag{7}$$

The 4 parameters $\mathcal{Q}^0 = (q_0, q_1, q_2, q_3)$ are called the (normalized) *"Euler parameters"*. They should not be confused with Eulerian angles, which are also often used in spatial kinematics! Note that "antipodal" Euler parameters $\pm\mathcal{Q}^0$ are correspond to the same rotation matrix $\underline{R}$. If we further denote

$$q_0 = \cos\frac{\phi}{2} \quad \text{and} \quad \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \sin\frac{\phi}{2}\,\vec{\mathbf{r}} \tag{8}$$

with a unit vector $\vec{\mathbf{r}}$, we may give a simple geometric interpretation of these parameters; the spherical displacement described by $\underline{R}$ is a rotation with angle ϕ about the axis spanned by $\vec{\mathbf{r}}$, cf. Figure 2.

Given a rigid body motion $M = M(t)$, there are various ways to compute its normalized Euler parameters. By comparing (1) and (6) we obtain the relations

$$\begin{aligned} q_0 : q_1 : q_2 : q_3 &= m_{0,0} + m_{1,1} + m_{2,2} + m_{3,3} : m_{3,2} - m_{2,3} : m_{1,3} - m_{3,1} : m_{2,1} - m_{1,2} \\ &= m_{3,2} - m_{2,3} : m_{0,0} + m_{1,1} - m_{2,2} - m_{3,3} : m_{2,1} + m_{1,2} : m_{1,3} + m_{3,1} \\ &= m_{1,3} - m_{3,1} : m_{2,1} + m_{1,2} : m_{0,0} - m_{1,1} + m_{2,2} - m_{3,3} : m_{3,2} + m_{2,3} \\ &= m_{2,1} - m_{1,2} : m_{1,3} + m_{3,1} : m_{3,2} + m_{2,3} : m_{0,0} - m_{1,1} - m_{2,2} + m_{3,3}, \end{aligned} \tag{9}$$

see [49]. At least one of these equations gives a result different from $0 : 0 : 0 : 0$. which may then be used, along with (7), to determine the normalized Euler parameters.

2.4. The kinematical mapping

Examining equations (6) and (9) we notice that there is a birational transformation which maps each rotation matrix onto two antipodal vectors $\pm\mathcal{Q}^0 = \pm(q_0, q_1, q_2, q_3)$ of

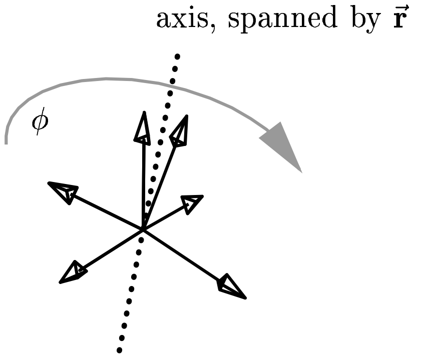


Figure 2. The geometric meaning of the Euler parameters. ϕ: angle of rotation, $\vec{\mathbf{r}}$: unit direction vector of the axis.

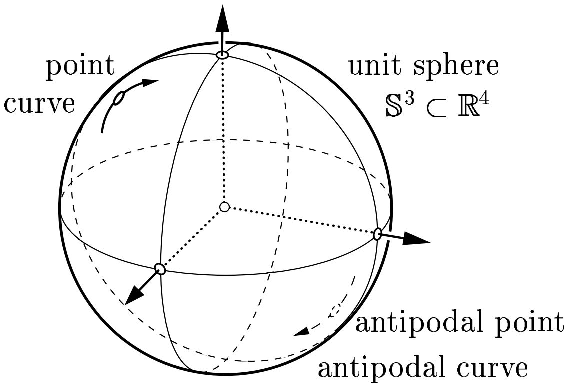


Figure 3. The kinematical mapping identifies a spherical motion with a pair of antipodal curves on the 4D unit sphere.

normalized Euler parameters, and vice versa. This transformation is called the *kinematical mapping* of spherical kinematics.

If we identify $\mathcal{Q}^0$ with a point in a four dimensional image space, the kinematical mapping defines a correspondence between the 3D rotations and the pairs of antipodal points on the 4D unit sphere $\mathbb{S}^3 \subset \mathbb{R}^4$. Furthermore, we may identify a spherical *motion* with a set of two antipodal *curves* on the unit sphere $\mathbb{S}^3$. This property is schematically illustrated in Figure 3. If no ambiguity is to be expected, then both the mapping from the set of 3D rotations to the unit sphere with identified antipodal points, and its inverse, will be referred to as the the kinematical mapping.

3. QUATERNIONS

Quaternions are a powerful tool to describe 3D rotations in spherical kinematics. In the following we will introduce the basic concepts of quaternion calculus and explain their relationship with the kinematical mapping.

3.1. Fundamentals

For any quadruple of real numbers $q_0, .., q_3$, we call the pair

$$\mathcal{Q} = [q_0, \vec{\mathbf{q}}] = [\ \underbrace{q_0}_{\text{real part}}\ , \underbrace{(q_1, q_2, q_3)^\top}_{\text{vector part}}\ ] \tag{10}$$

of the scalar q_0 and the vector $\vec{\mathbf{q}} = (q_1, q_2, q_3)^\top$ a *quaternion.* The quaternion $\overline{\mathcal{Q}} = [q_0, -\vec{\mathbf{q}}]$ is called the conjugate quaternion of $\mathcal{Q}$. Let us further define two operations that act on the set of all quaternions $\mathbb{H}$,

$$\begin{aligned} \mathcal{Q} + \mathcal{R} &= [q_0, \vec{\mathbf{q}}] + [r_0, \vec{\mathbf{r}}] = [\ q_0 + r_0,\ \vec{\mathbf{q}} + \vec{\mathbf{r}}\ ], \\ \mathcal{Q} * \mathcal{R} &= [q_0, \vec{\mathbf{q}}] * [r_0, \vec{\mathbf{r}}] = [\ \underbrace{q_0 r_0 - \vec{\mathbf{q}} \cdot \vec{\mathbf{r}}}_{\text{real part}},\ \underbrace{q_0 \vec{\mathbf{r}} + r_0 \vec{\mathbf{q}} + \vec{\mathbf{q}} \times \vec{\mathbf{r}}}_{\text{vector part}}\ ]. \end{aligned} \tag{11}$$

With these operations, the set of quaternions $\mathbb{H}$ forms a skew field (Hamilton, 1840). Quaternions with

$$\mathcal{Q} * \overline{\mathcal{Q}} = [q_0^2 + q_1^2 + q_2^2 + q_3^2, (0,0,0)] = [1, (0,0,0)] \tag{12}$$

are called *unit quaternions*, they are marked with $\mathcal{Q}^0$. (Compare with the definition of normalized Euler parameters, (7).) Quaternions with vanishing scalar part $[0, \vec{\mathbf{p}}]$ will be identified with vectors in $\mathbb{R}^3$. We may express the usual scalar and cross product of two vectors $\vec{\mathbf{q}}$, $\vec{\mathbf{r}}$ in terms of the quaternion multiplication

$$\begin{array}{ll} [\vec{\mathbf{q}} \cdot \vec{\mathbf{r}}, (0,0,0)] & = -\frac{1}{2}\left([0,\vec{\mathbf{q}}] * [0,\vec{\mathbf{r}}] + [0,\vec{\mathbf{r}}] * [0,\vec{\mathbf{q}}]\right), \\ [0, \vec{\mathbf{q}} \times \vec{\mathbf{r}}] & = \frac{1}{2}\left([0,\vec{\mathbf{q}}] * [0,\vec{\mathbf{r}}] - [0,\vec{\mathbf{r}}] * [0,\vec{\mathbf{q}}]\right). \end{array} \tag{13}$$

Now we consider the quaternion product

$$\vec{\mathbf{p}} \rightarrow= \vec{\mathbf{p}}' = \underbrace{\overline{\mathcal{Q}^0} * [0, \vec{\mathbf{p}}] * \mathcal{Q}^0}_{=\mathcal{X}}, \tag{14}$$

where Q^0 is a unit quaternion, resulting in a vector-type quaternion $[0, \vec{\mathbf{p}}'] \sim \vec{\mathbf{p}}'$. In fact, this product can be shown to satisfy the condition

$$\overline{\overline{\mathcal{Q}} * [0, \vec{\mathbf{p}}] * \mathcal{Q}} = -\overline{\mathcal{Q}} * [0, \vec{\mathbf{p}}] * \mathcal{Q}, \tag{15}$$

hence $\overline{\mathcal{X}} = -\mathcal{X}$, which characterizes the vector–type quaternions. With the help of the relationships (13) it can easily be shown that the mapping $\vec{\mathbf{p}} \mapsto \vec{\mathbf{p}}'$ preserves both products between any two vectors $\vec{\mathbf{p}}$ and $\vec{\mathbf{r}}$. Hence, the mapping (14) can equivalently be described as

$$\vec{\mathbf{p}} \rightarrow \vec{\mathbf{p}}' = \underline{U}\,\vec{\mathbf{p}}, \tag{16}$$

where U is a special orthogonal matrix, depending on the four components of the quaternion $\mathcal{Q}^0$. A short calculation indeed confirms that *the matrix U is the special orthogonal matrix with the normalized Euler parameters $\mathcal{Q}^0$*, see (6). Moreover, the *composition of rotations $\underline{U} = \underline{U}_1 \cdot \underline{U}_2$* corresponds to the *multiplication of quaternions $\mathcal{Q}^0 = \mathcal{Q}_2^0 * \mathcal{Q}_1^0$*.

3.2. Homogeneous quaternions and the kinematical mapping

We now rewrite equation (14) in order to allow the use of non-normalized quaternions, by switching to a homogeneous representation. Firstly we note that (14) is equivalent to

$$[1, \vec{\mathbf{p}}] \rightarrow [1, \vec{\mathbf{p}}'] = \overline{\mathcal{Q}^0} * [1, \vec{\mathbf{p}}] * \mathcal{Q}^0$$

In addition, we identify the homogeneous coordinates of a point $\mathbf{p}$ with the quaternion $[p_0, (p_1, p_2, p_3)]$. This leads us to the homogeneous quaternion representation of a rotation (or spherical displacement),

$$\hat{\mathbf{p}} \mapsto \mathbf{p} = \overline{\mathcal{Q}} * \hat{\mathbf{p}} * \mathcal{Q}, \tag{17}$$

where $\mathbf{p}$ and $\hat{\mathbf{p}}$ are the homogeneous coordinates of a point with respect to fixed and moving coordinate system, respectively, and $\mathcal{Q}$ is a quaternion. The associated unit quaternions $\pm\mathcal{Q}^0 = \|\mathcal{Q}\|^{-1}\,\mathcal{Q}$, where $\|\mathcal{Q}\| = \sqrt{\mathcal{Q} * \overline{\mathcal{Q}}}$, consist of the Euler parameters according to (8).

Similar to the homogeneous coordinates of a point $\mathbf{p}$, the quaternion $\mathcal{Q}$ in (17) can be considered a homogeneous representation of the rotation. Again, as in the point case, linearly dependent quaternions describe the same rotation. However, the normalization (12) defines a different underlying geometric structure than in the point case, where normalized coordinates were characterized by $p_0 = 1$. In the quaternion space, the geometric structure is that of a 3–dimensional elliptic space , see Chapter ?, [2]. The unit quaternion sphere $\mathbb{S}^3 \subset \mathbb{R}^4$ with identified pairs of antipodal points is the standard model of this geometry. For our applications, it is more appropriate to use homogeneous coordinates for points in 3–dimensional elliptic (i.e. quaternion) space. These coordinates will be called *homogeneous quaternion coordinates* of rotations.

3.3. Summary: homogeneous quaternion coordinates for 3D rotations

The kinematical mapping maps a point $\mathcal{Q} \neq [0,(0,0,0)]$ in 3–dimensional elliptic space, described by homogeneous quaternion coordinates, to the special orthogonal matrix

$$\underline{U}(\mathcal{Q}) = \frac{1}{u_0(\mathcal{Q})} U(\mathcal{Q}), \tag{18}$$

where

$$U(\mathcal{Q}) = (u_{i,j})_{i,j=1,2,3} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{19}$$

and $u_0(\mathcal{Q}) = q_0^2+q_1^2+q_2^2+q_3^2$. Any point in elliptic 3–space $\mathcal{Q}$ corresponds uniquely to a rotation (or spherical displacement) $\underline{U}$. Every curve in homogeneous quaternion coordinates $\mathcal{Q}(t)$ can be identified with an intrinsic spherical motion $\underline{U}(t)$. This kinematical mapping is birational. For more information on quaternions, kinematic mappings and their application we refer the reader to [2,3,9,30,49].

4. MOTION DESIGN USING CURVES ON $\mathbb{S}^3$

As we have seen in the previous sections, spherical motions are equivalent with curves in elliptic 3-space. The standard model of elliptic 3-space is the unit quaternion sphere $\mathbb{S}^3$ in $\mathbb{R}^4$ with identified pairs of antipodal points. This section discusses some methods that have been proposed in the literature for designing spherical motions with the help of curves on $\mathbb{S}^3$.

4.1. Slerping

The algorithm of de Casteljau is based on iterated linear interpolation, see Section 2.3 of Chapter ?. Though conceptually simple, it gives a very effective tool for curve design that is numerically stable and easy to implement. As a simple approach to curve design on $\mathbb{S}^3$ one may translate de Casteljau's algorithm into the geometry of a sphere, in order to produce spherical Bézier–type curves. This is achieved by replacing the line segment connecting two points in Euclidean space with the great circular arc (the geodesic) between two points on a sphere.

More precisely, consider two points$\mathcal{B}_0$ and $\mathcal{B}_1$ on the unit quaternion sphere $\mathbb{S}^3$. We define the point $\mathcal{B}_0^1(t) = \text{slerp}(\mathcal{B}_0, \mathcal{B}_1, t)$ such that $\mathcal{B}_0^1(t)$ lies on the great circular arc passing through $\mathcal{B}_0$ and $\mathcal{B}_1$ and the angles between the coordinate vectors of $\mathcal{B}_0$, $\mathcal{B}_1$ and $\mathcal{B}_0^1(t)$ satisfy

$$\angle(\mathcal{B}_0, \mathcal{B}_0^1(t)) \ : \ \angle(\mathcal{B}_0^1(t), \mathcal{B}_1) \ = \ t \ : \ (1-t). \tag{20}$$

As t varies from 0 to 1, the point $\mathcal{B}_0^1(t)$ traces a great circular arc from $\mathcal{B}_0$ to $\mathcal{B}_1$. Based on this spherical linear interpolation ('slerp') we are now able to define a spherical version of de Casteljau's algorithm as illustrated in Figure 4.

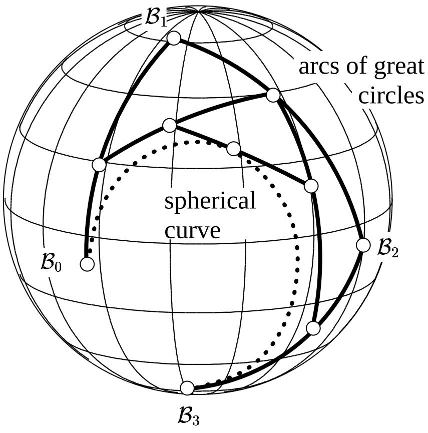


Figure 4. Spherical de Casteljau algorithm. Based on a spherical control polygon $\mathcal{B}_0, .., \mathcal{B}_n$, repeated spherical linear interpolation results in a spherical curve.

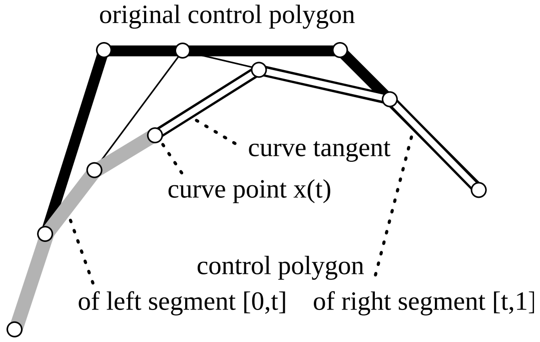


Figure 5. Subdivision and tangent property of Bézier curves, cf. Chapter **?**. Both properties are not valid for the spherical de Casteljau algorithm.

This technique is commonly referred to as 'slerping'. It has been introduced by Shoemake in [44] and has since then been subject of much research, mainly in Computer Graphics, e.g. [27,33,36].

As a first problem, the point $\mathcal{B}_0^1(t)$ is not unique. While this problem can easily be resolved, for example, by restricting $\mathcal{B}_0^1$ to the shorter arc connecting $\mathcal{B}_0$ with $\mathcal{B}_1$, there are more involved problems that are caused by the subtle differences between elliptic and Euclidean geometry.

4.2. Problems of slerping

Two fundamental properties of de Casteljau's algorithm are the subdivision property, and – closely related to it – the fact that the points obtained in last step of the algorithm span the tangent of the resulting Bézier curve. See Sections 2.3 and 2.5 of Chapter **?** for details. Unfortunately, both properties are lost by the spherical version of the algorithm. The lack of these fundamental properties has serious consequences, as none of the algorithms derived from the subdivision property can be transferred onto the sphere. In particular, C^2 joints are difficult to construct, as the standard Bézier–based construction (leading directly to B-splines, see [16, Section 4.1.2]) is closely related to extrapolation of a curve via de Casteljau's algorithm, cf. [33]. Moreover, it is not possible to use the efficient subdivision–based rendering methods for Bézier curves in the spherical situation.

In addition to these missing fundamental properties, it is very complicated to analyze the curves (and the resulting spherical motions) which result from slerping. Even for relatively low degrees, the parametric representations are rather involved, and it is therefore difficult to apply standard tools from analysis and differential geometry.

Finally, the interpolation problem for slerp Bézier curves leads to a non–linear system of equations; only approximate solutions can be found. This is clearly a serious disadvantage, as interpolation is one of the basic techniques for curve design.

4.3. Other approaches

Alternative algorithms for constructing smooth unit quaternion splines have been proposed by various authors, see [1,25,27,40,26] and elsewhere. For instance, these algorithm are based on the cumulative form of a Bézier or B–spline curve, or on blending techniques for spherical curves. In a more general setting, Park and Ravani [35] have studied Bézier curves on Riemannian manifolds.

As an alternative to generalizing the de Casteljau algorithm it is also possible to generalize the associated subdivision schemes to the spherical case. This idea has stimulated research on non-linear corner cutting algorithms on Riemannian manifolds. For instance, in the case of cubic Bézier curves (which corresponds to the Lane–Riesenfeld algorithm), a thorough analysis has been given by Noakes [34], showing that the limit curve is differentiable, and its derivative Lipschitz. Note that non-linear corner cutting, although conceptually simple, requires rather involved mathematical tools, both for generating and for analyzing motion trajectories.

A detailed study of computational techniques for motion design can be found in the survey article by Röschel [43], providing many further references.

4.4. Motion design – desired features

We conclude this section by listing a few features which should be provided by algorithms for motion design.

- The evaluation scheme should be simple and efficient, preferably without involving non-rational functions. Motion design algorithms should be easy to implement, robust and numerically stable. The resulting motion splines should at least exhibit symmetry and subdivision properties.
- Motion trajectories should have a simple representation which follows a generally accepted standard in Computer Aided Design. If possible, trajectories should be represented as NURBS curves.
- Conditions for interpolation of given data (positions, velocities etc.) and for smooth joints should be easily to formulate and computationally efficient (preferably linear conditions).
- Motion design algorithms (e.g. via interpolation techniques) should be invariant with respect to the choice of the fixed coordinate system (world coordinates), and with respect to the choice of the orientation of the moving coordinate system. That is, coordinate transformations and interpolation algorithms should commute. An additional

invariance with respect to the choice of the origin of the moving coordinate system is not really useful, as the origin will mostly have a special meaning in applications, such as the center of gravity or the 'tool center point' (TCP).

None of the approaches we have examined so far satisfies all of these properties. In particular we note that point trajectories generated by slerping algorithms are non-rational and therefore do not comply with the industrial NURBS standard (see Chapter **?**). In the remainder of this chapter we present an approach based on the kinematical mapping which produces motions generating point trajectories in NURBS form.

5. SPHERICAL RATIONAL MOTIONS

The kinematical mapping (18) can be used in order to apply the Bézier and B-spline techniques to spherical motions, following the ideas in [10,37]. For instance, consider a rational Bézier curve of degree n in elliptic 3–space,

$$\mathcal{Q}(t) = \sum_{i=0}^{n} B_i(t)\, \mathcal{B}_i, \quad t \in [0,1], \tag{21}$$

with the Bernstein polynomials $B_i(t) = \binom{n}{i} t^i (1-t)^{n-i}$. Its control polygon consists of the *control points* $\mathcal{B}_i = [b_{i,0}, (b_{i,1}, b_{i,2}, b_{i,3})^\top] \in \mathbb{H}$ and the *Farin points*

$$\mathcal{F}_{i,i+1} = \mathcal{B}_i + \mathcal{B}_{i+1}, \tag{22}$$

see Figure 6, left. We use homogeneous coordinates to represent these points. Consequently, the Farin points (also called *weight* or *frame points*) are located on the edges of the control polygon; they represent the weight ratio of neighbouring control points. For further information the reader should consult, e.g., [7,11], and Chapter **?**. The combination of control and Farin points provides a projectively invariant description of a rational Bézier or B-spline curve. It is also invariant in the sense of elliptic geometry, as elliptic transformations are special cases of projective mappings.

Now we apply the kinematical mapping, both to the rational Bézier curve (21) and to its control and Farin points. Firstly, consider the image of the linear Bézier curve,

$$\mathcal{Q}^{(1)}(t) = (1-t)\, \mathcal{B}_0 + t\, \mathcal{B}_1, \quad t \in [0,1]. \tag{23}$$

It turns out to be a rotation of the unit sphere with a constant axis. More precisely, the components of the rotation matrix $\underline{U}^{(1)}(t) = \underline{U}(\mathcal{Q}^{(1)}(t))$ are quadratic rational functions, cf. (18). The trajectory of any point $\underline{\hat{\mathbf{p}}}$ of the moving system is simply a circular arc, which is described as as a rational quadratic curve $\underline{U}^{(1)}(t)\, \underline{\hat{\mathbf{p}}}$.

Next we consider the image of a rational Bézier curve (21) of degree n, see Figure 6. It is a *spherical rational motion* of degree $2n$, as the components of $\underline{U}(t) = \underline{U}(\mathcal{Q}(t))$ are rational functions of degree $2n$. The trajectory of any point $\underline{\hat{\mathbf{p}}}$ of the moving system is the rational curve $\underline{U}(t)\, \underline{\hat{\mathbf{p}}}$ of degree $2n$. As an example, Fig. 6 shows a cubic rational Bézier curve and its image under the kinematical mapping.

More generally, the kinematical mapping could be applied to a rational B-spline curve of degree d in elliptic 3-space, resulting from (21) by replacing the Bernstein polynomials

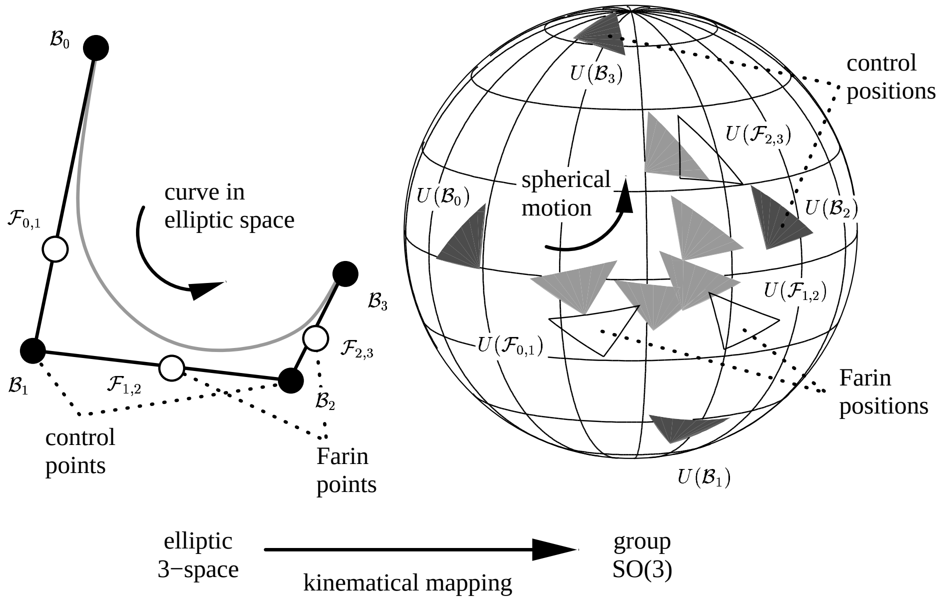


Figure 6. The image of a rational Bézier curve in elliptic 3-space under the kinematical mapping is a spherical rational motion of degree $2n$. Applying the mapping to the Bézier control polygon gives a control structure for spherical rational motions.

$B_i(t)$ with B-splines defined over a suitable knot sequence. This produces a spherical rational spline motion $\underline{U}(\mathcal{Q}(t))$ of degree $2d$. The computation of the B-spline form involves the evaluation of products of B-splines, see [32]. If the preimage curve has single interior knots, then both the preimage curve and the spherical rational spline motion are C^{d-1}. Hence, the inner knots of the spline functions $u_0(\mathcal{Q}(t))$ and $U(\mathcal{Q}(t))$ have at least multiplicity $d+1$. Computational techniques for rational spline motions in B-spline form, including a formula for their Bézier segments, have been discussed in [22].

By applying the kinematical mapping to the control and Farin points of the rational Bézier (or B-spline) curve we obtain an *intrinsic control structure* for spherical rational (spline) motions. This control structure has been introduced by Pottmann [37]. It is obtained by applying the kinematical mapping to the control and Farin points of the rational curve, leading to *control positions* and *Farin positions.* The edges of the control polygon are mapped to *rotations* of the unit sphere, joining two neighbouring control positions and the corresponding Farin position. In fact, the edges can be seen as as linear rational Bézier curves $B_0^1(t)\mathcal{B}_{i-1} + B_1^1(t)\mathcal{B}_i$ $(i = 1, \ldots, n)$; the Farin point is associated with the parameter value $t = \frac{1}{2}$.

The intrinsic control structure is suitable for interactive motion design. An example is shown in Figure 7. The two spherical rational Bézier motions are obtained from the motion of Figure 6 by changing the first control position $\underline{U}(\mathcal{B}_1)$ (left) and by modifying the weight of the second control position $\underline{U}(\mathcal{B}_2)$ (right), leading to modified Farin points $\mathcal{F}_{1,2}$ and $\mathcal{F}_{2,3}$.

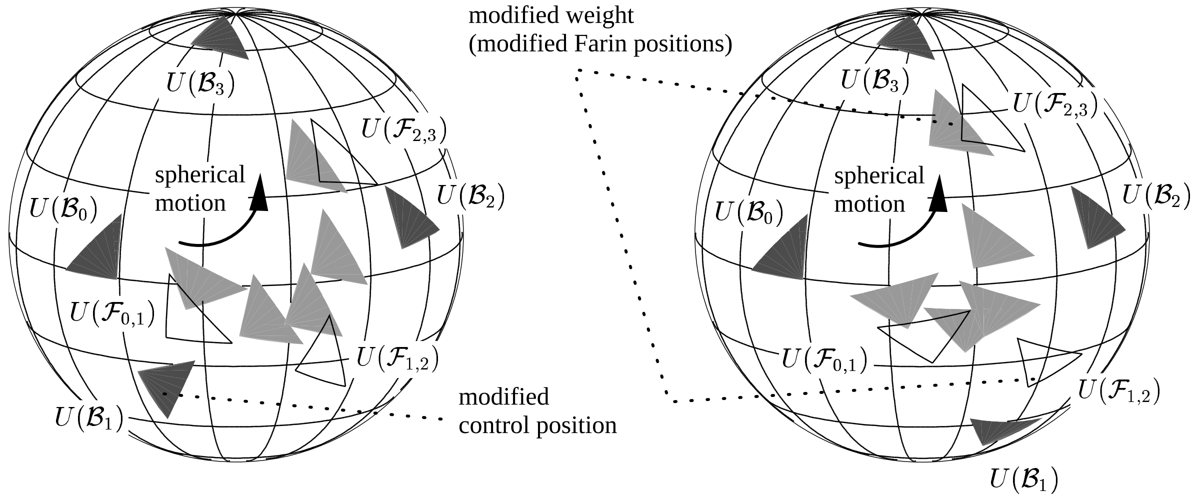


Figure 7. Interactive motion design using the intrinsic control structure of a spherical rational motion. Modification of a control position (left) and of a weight (right), compare with Figure 6.

A similar technique can be used to generate an intrinsic control structure for *spherical rational curves*, using the so–called generalized stereographic projection (see [6,37], cf. Chapter **?**. In fact, the generalized stereographic projection can be derived by restricting the kinematical mapping (18) to the trajectory of a single point.

In principle, the properties of the Bézier and B-spline control points (convex hull property etc.) can be transferred to the intrinsic control structure of a spherical rational motion, with the help of the kinematical mapping. However, it is difficult to give a useful geometric interpretation, as the image of a volume in elliptic space is somewhat difficult to visualize on the sphere.

Consider a spherical rational motion $\underline{R}(t) = \frac{1}{r_0(t)} R(t)$ of degree m. That is, the denominator $r_0(t)$ and the 9 components of $R(t)$ are polynomials of maximum degree m, and the matrices $\underline{R}(t)$ are special orthogonal matrices for all t. Then, the trajectory $\underline{R}(t)\underline{\hat{\mathbf{p}}}$ of any point $\underline{\hat{\mathbf{p}}}$ is a spherical rational curve of degree m, on the sphere (centered at the origin) with radius $||\underline{\hat{\mathbf{p}}}||$. Clearly, by applying the kinematical mapping to a rational curve of degree n we obtain a spherical rational motion $\underline{R}(t)$ of degree $m = 2n$. Conversely, one may ask whether any motion can be constructed that way.

Proposition [18]. *If the denominator $r_0(t)$ and the 9 components of $R(t)$ do not share polynomial factors, i.e.*

$$\gcd\{r_0(t), r_{1,1}(t), r_{1,2}(t), r_{1,3}(t), r_{2,1}(t), r_{2,2}(t), r_{2,3}(t), r_{3,1}(t), r_{3,2}(t), r_{3,3}(t)\} = 1 \tag{24}$$

holds in the polynomial ring $\mathbb{R}[t]$, then m is even, and the spherical rational motion $\underline{R}(t) = \frac{1}{r_0(t)} R(t)$ can be generated by applying the kinematical mapping $\underline{U}(.)$ to a rational curve $\mathcal{Q}(t)$ of degree $m/2$.

The proof consists of two parts. Firstly it is shown that any spherical rational motion corresponds to a rational curve in elliptic 3-space, as it has rational Euler parameters $\mathcal{Q}(t)$. This can be concluded from (9), expressing the Euler parameters as rational functions of the matrix components. Secondly, it can be shown that any common factor of the matrix components is a common factor of the corresponding Euler parameters. This observation leads to the degree bound of the proposition. For the details of the proof the reader is referred to [18,20].

Summing up, spherical rational motions can be generated by applying the kinematical mapping to rational curves in elliptic 3–space. According to the proposition, this construction produces motions having the minimum possible degree. In addition, the Bézier resp. B-spline control structure of rational spline curves can be translated to the kinematical setting.

6. SPATIAL RATIONAL MOTIONS

The results on spherical rational spline motions can be extended to spatial ones, by combining them with rational trajectories of the origin. This section discusses the construction and classification of rational spline motions, and the use of control polygons and control structures.

6.1. Construction

Recall from Section 2.2 that the motion of a rigid body is described by a time-dependent transformation $\hat{\mathbf{p}} \mapsto \mathbf{p}(t) = M(t)\,\hat{\mathbf{p}}$ of the form (1), mapping any point $\hat{\mathbf{p}}$ of the moving system to a point on its trajectory $\hat{\mathbf{p}}(t)$. We use homogeneous coordinates to represent both the points and the transformation.

If the components of the matrix $M(t)$ are rational (spline) functions of degree m, then the corresponding rigid body motion is called a *rational (spline) motion of degree n.* All trajectories are rational (spline) curves of degree m, see Figure 8 for an example. Consequently, as the trajectories can be described as NURBS curves, rational motions comply with industrial CAD standards.

Consider a transformation matrix of the form

$$M(t) = \left[\begin{array}{c|ccc} v_0^*(t)\;u_0(t) & 0 & 0 & 0 \\ \hline v_1(t) & & & \\ v_2(t) & & v_0^*(t)\;U(t) & \\ v_3(t) & & & \end{array}\right]. \tag{25}$$

The associated spherical motion $\underline{U}(t) = [1/u_0(t)]\;U(t)$ has been constructed by applying the kinematical mapping (18) to a rational spline curve $\mathcal{Q}(t)$ of degree k in elliptic 3–space; the piecewise polynomials $v_0^*(t)$ of degree p and $v_1(t), v_2(t), v_3(t)$ of degree q are arbitrary. Then, the motion (25) describes a spatial rational spline motion of degree $m = \max(q, p + 2k)$.

The origin of the moving space generates the trajectory

$$M(t)\;(1,0,0,0)^\top = (v_0^*(t)\,u_0(t), v_1(t), v_2(t), v_3(t))^\top \tag{26}$$

Given a spherical rational spline motion, the spatial rational spline motion (25) may combine any trajectory $\mathbf{p}(t) = (p_0(t), p_1(t), p_2(t), p_3(t))^\top$ of the origin with it, by choosing

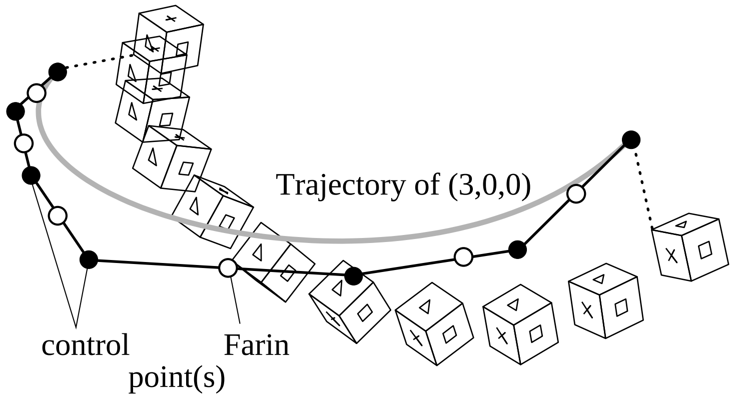


Figure 8. Example - a rational motion of degree 6. The motion is visualized by the moving unit cube. In addition, the trajectory of a point, along with its rational Bézier control polygon consisting of control points $\mathbf{b}_i$ and Farin points $\mathbf{f}_{i,i+1}$ is shown.

$v_0^*(t) = p_0(t)$ and $v_i(t) = u_0(t)\,p_i(t)$, $i = 1, 2, 3$. Generally, by combining a spherical rational spline motion of degree $n = 2k$ with a degree q rational spline curve one obtains a spatial rational spline motion of degree $m = 2k+q$. In applications, however, the degree should often be kept as small as possible. This can be achieved by choosing trajectories $\mathbf{v}(t)$ whose denominator $v_0(t) = v_0^*(t)u_0(t)$ equals the denominator u_0 of the associated spherical rational spline motion, i.e., by choosing $u_0 \equiv 1$.

Similar to the previous section, it is a natural question to ask whether any spatial rational (Bézier) motion can be generated with the help of formula (25).

Theorem [18]. *Any spatial rational motion of degree m is obtained from (25) by applying the kinematical mapping to a rational (Bézier) curve $\mathcal{Q}(t)$ of degree k in elliptic 3–space, where the degree k satisfies $0 \le k \le \lfloor m/2 \rfloor$, leading to the associated spherical rational motion $\underline{U}(\mathcal{Q}(t)) = [1/u_0(\mathcal{Q}(t))]\; U(\mathcal{Q}(t))$ of degree $2k$, and choosing polynomials $v_0^*(t)$ of degree $m - 2k$, and $v_1(t), v_2(t), v_3(t)$ of degree m.*

This result follows immediately from the previous proposition. Consequently, there are $\lfloor m/2 \rfloor + 1$ different classes of rational motions of degree m, corresponding to the degree $2k$ of the associated spherical rational motion.

6.2. Special cases

Rational motions of degree $m \le 4$ have thoroughly been studied in the theory of space kinematics. The simplest non–trivial example, given by quadratic rational motions with $k = 1$, can be traced back to Darboux [3,5]. In the general situation, these motions are obtained by composing a planar elliptic motion with a harmonic oscillation. The elliptic motion is a special trochoidal motion: a small circle (radius r) rolls within a in big circle (radius R), where $r : R = 1 : 2$, see Figure 9. All trajectories are ellipses, except for the points on the rim of the small circle, which trace diameters of the big circle. This motion is extended into 3–space, where it becomes the rolling of two circular cylinders. By adding a synchronized harmonic oscillation in the direction of the cylinders' axes we obtain a

Darboux motion, see again Figure 9. As the elliptic motion and the harmonic oscillation have equal frequencies, all trajectories are still ellipses. Darboux motions can be shown to be the most general truly spatial motions which generate planar point trajectories for all points of the moving system.

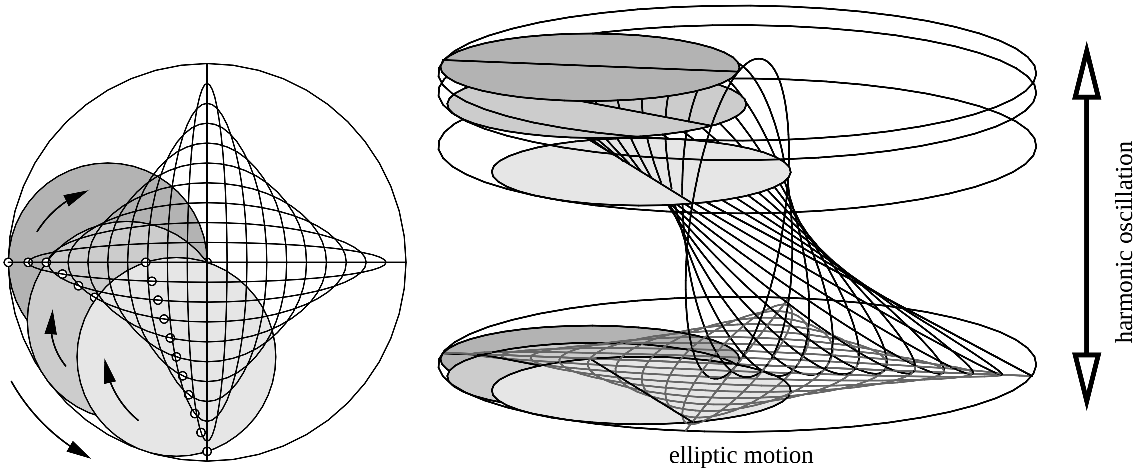


Figure 9. Elliptic motion (left) and Darboux motion (right).

More recently, a thorough geometric analysis of rational motions of degree 3 and 4 has been given by Wunderlich and Röschel [41,50].

6.3. Affine control structure

Consider again a spatial rational spline motion of degree n. It is described by a transformation matrix $M(t)$, see (1) and (25), whose components are piecewise polynomial functions (splines) of degree n. Consequently, one may represent the transformation matrix in B-spline form (or even in Bézier form, in the case of a spatial rational motion),

$$M(t) = \sum_i C_i \, B_i(t), \quad t \in [a, b], \tag{27}$$

with 4×4 control matrices C_i and B-splines $B_i(t)$, defined with over a suitable associated knot sequence. Similar to the transformation matrices $M(t)$, the coefficient matrices $C_i = (c^{(i)}_{j,k})_{j,k=0,\ldots,4}$ satisfy

$$c^{(i)}_{0,1} = c^{(i)}_{0,2} = c^{(i)}_{0,3} = 0, \quad i = 0, \ldots, N. \tag{28}$$

The orthogonality condition (3), however, is generally not satisfied (except for control matrices describing positions, e.g., at the boundaries – if the boundary knots have sufficient multiplicity).

Any point $\hat{\mathbf{p}}$ of the moving system traces a rational B-spline curve,

$$M(t)\,\hat{\mathbf{p}} = \sum_{i=0}^{N} [C_i\,\hat{\mathbf{p}}]\; B_i(t), \tag{29}$$

with control points $C_i\,\mathbf{p}$ and weights $p_0\,c_{0,0}^{(i)}$ $(i = 0, \ldots, N)$. Alternatively, one may again use Farin points to specify the weight ratios, $(C_{i-1} + C_i)\mathbf{p}$ $(i = 1, \ldots, N)$.

Consider a moving object $\hat{\mathcal{O}}$, which is described as a bounded set of points in the moving system. Collecting the control and Farin points of the trajectories we obtain the *control positions* $C_i\,\hat{\mathcal{O}}$ and *Farin positions* $(C_{i-1} + C_i)\,\hat{\mathcal{O}}$. Generally, the transformations C_i and $(C_{i-1} + C_i)$ do not preserve the rigidity (i.e., distances and angles) of the object, as the orthogonality condition (3) is not satisfied. The control and Farin positions are affine images of the moving object, as the matrices describe affine mappings (preserving ratios and parallelism), due to (28). The combination of control and Farin positions is called the *affine control structure*, see [22,45].

An example is shown in Figure 10a, b. The spatial rational motion (degree 6) of the moving unit cube (a) has been generated by composing the spherical rational motion of Figure 6 with a suitable trajectory of the origin (a rational Bézier curve of degree 6). We chose $v_0^* \equiv 1$ in (25), hence the resulting spatial motion has still degree 6. The affine control and Farin positions (b) are obtained by collecting the control and Farin points of the trajectories generated by the points of the moving unit cube.

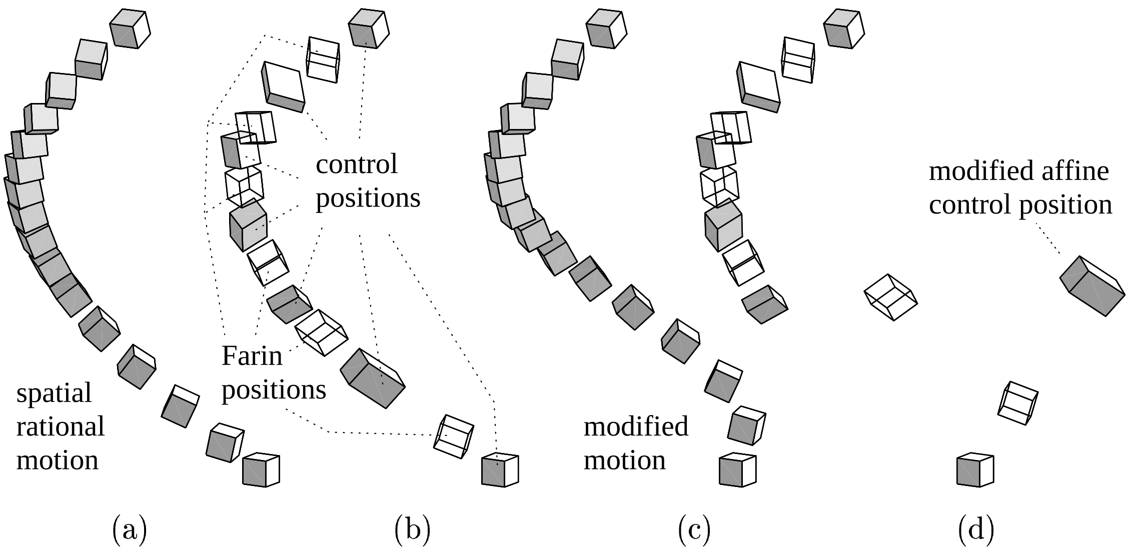


Figure 10. Two spatial rational motions (a,c) of degree 6 of a moving unit cube, and their affine control structure (b,d), consisting of control positions (solid) and Farin positions (wireframe). For both motions, the associated spherical motion is that of Figure 6.

The affine control structure is not suitable for designing the spherical part of the spatial rational motion. In particular, any change of the shape of the affine control positions, and/or of the associated weights (Farin positions) may entail a violation of the orthogonality condition (3). The spherical part should be designed with the intrinsic control structure of spherical rational motions. The affine control structure can be used for designing the *translational part* of the motion, as it is possible to apply arbitrary translations to the affine control positions. This is demonstrated in Figure 10c,d, where a translation has been applied to the second control position.

The affine control structure can be used for efficiently generating a bounding volume for the moving object. This fact has potential application to collision detection and avoidance; it can also be used for approximate computation of envelopes. If the weights are positive, then any intermediate position of the moving object is contained within the convex hull of the affine control positions. If $v_0^* \equiv 1$ has been chosen (which will mostly be the case in applications), then the denominator of the spatial rational spline motion is a sum of four squares, see (19). Consequently, the weights will mostly be positive; they can always made non–negative by splitting the rational spline motion into suitable smaller segments.

As an example, we demonstrate the convex hull property of a planar rational motion, see Figure 11. Any *planar rational spline motion* of the x_1x_2–plane can be obtained from (25) by applying the kinematical mapping to preimage curves $\mathcal{Q}(t)$ in elliptic 3–space with $q_1(t) \equiv q_2(t) \equiv 0$, and choosing $v_3(t) \equiv 0$. For a more detailed geometric discussion of planar rational motions the reader should consult [46]. As observed there, the associated affine control structure consists of *equiform images* of the moving object.

A planar rational motion, along with its (equiform) control and Farin positions is shown in Figure 11a. The resulting convex hull gives a bound on the motion of the object. This result can be made more accurate by splitting the motion into smaller segments and generating the convex hull of the resulting control structures, see Figure 11b.

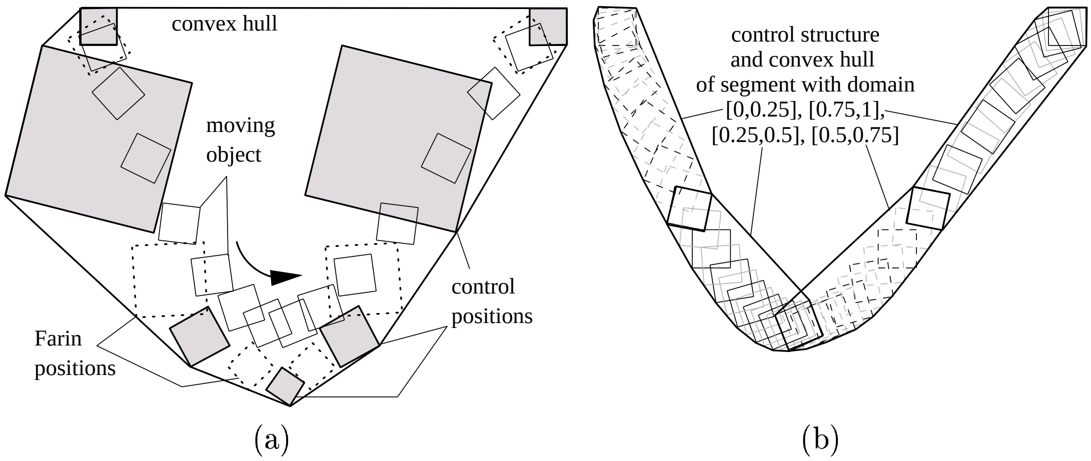


Figure 11. Computing the convex hull of a planar rational motion. The motion and its control structure (a), and the convex hulls obtained after splitting the curve into four segments (b).

The same idea can be applied to spatial rational spline motions. However, the computation of the convex hull in 3D becomes more expensive, and alternative techniques (such as bounding boxes) will be preferred.

6.4. Some properties

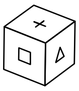
Spatial rational spline motions provide various desirable features, making them a useful tool in applications.

- The standard rational Bézier and B-spline techniques for CAGD can be applied, providing simple and efficient algorithms for evaluating these motions. Spatial rational spline motions generate point trajectories which can easily be represented in B-spline form, complying with industrial CAD standards. They can be equipped with control structures which are suitable for interactive motion design and for generating convex hulls of moving objects.
- Spatial rational spline motions have a subdivision property, as any subsegment can again be described in rational spline form. Consequently, standard CAGD techniques like hierarchical editing can be used. Moreover, it is possible to generate rational spline motions with arbitrarily high order of differentiability. This was not the case for 'slerp' Bézier spline motions [33].
- The class of rational spline motions is invariant with respect to the choice of the coordinate system in both the fixed and the moving system. Moreover, efficient interpolation techniques, generalizing the standard interpolation algorithms for spline curves, can easily be derived, see next section for details. They produce results which are independent of the choice of the fixed coordinate system, and of the choice of the orientation of the moving system [22].

  The results of the interpolation algorithm described in the next section depend on the choice of the origin of the moving system. This dependency is often desired in applications, as the origin may have a special geometric meaning (e.g., the center of gravity, tool center point). However, even this dependency can be avoided, using the more sophisticated techniques of [20].

In addition, it is possible to represent trajectories of *moving planes* in rational spline form, and to introduce a *dual control structure* [23,45]. This leads to explicit formulas for envelopes of moving planes [22,45] and, more generally, for envelopes of moving rational developable surfaces (including quadratic cylinders and cones) see [23,45,51]. As an application one may compute the envelope of moving polyhedra, without numerical approximation.

With the help of a kinematical mapping for spatial displacements, a slightly different approach to the design of spatial rational motions has been developed by Ge and Ravani [10]. It is based on the use of dual quaternions (numbers from the ring $\mathbb{H} + \epsilon\mathbb{H}$, where $\epsilon^2 = 0$), see [3]). Following this approach, the motion is described by a sequence of control positions with associated dual weights. One obtains an intrinsic control structure for spatial rational motions, whose "legs" are special Darboux motions (Darboux motions with constant axis; the two cylinders degenerate into a fixed line). The control structure is suitable for interactive motion design. However, this approach is closer to *line trajectories* (i.e., ruled surfaces) than to point trajectories; consequently, it is more difficult to control the trajectory of a specific point, such as the origin of the moving space. Also, the influence of the dual weights is sometimes not very intuitive, and the control structure does not provide a convex hull property. By restricting that approach to spherical control



positions one arrives again at the intrinsic control structure for spherical rational motions, see Section 5.

6.5. Interpolation schemes and applications

Interpolation and approximation of given point data are fundamental techniques for generating curves and surfaces in Computer Aided Geometric Design. This section demonstrates that the standard interpolation schemes can be generalized to the kinematical setting, with the help of spatial rational motions.

Let a sequence of positions $(\mathrm{Pos}_i)_{i=1,\dots,N}$ of a moving object be given. Each position is described by a coordinate transformation of the form (1) between fixed and moving system. We assume that the origin of the moving coordinate system has a special meaning, such as the center of gravity, tool center point (TCP), etc. The data are to be interpolated with a spatial rational spline motion, see Figure 12.

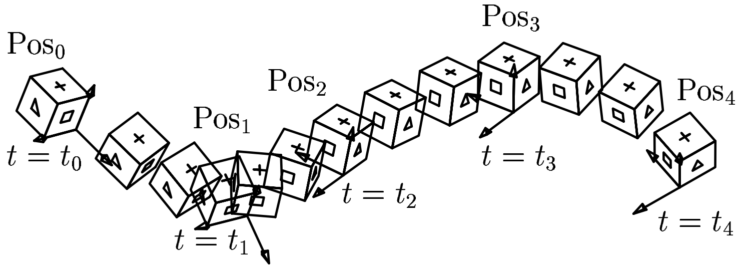


Figure 12. Interpolation of 5 given positions of the unit cube with a spatial rational rigid body motion. By browsing through this chapter, you will see an animation of the associated spherical motion in the upper right corners of the odd pages.

In the sequel we give a brief summary of a the interpolation procedure. For further details the reader should consult, e.g., [22].

1. **Preprocessing.** As the initial step of the interpolation procedure, the given data is converted into quaternion form. More precisely, each of the given positions is described by the Cartesian coordinates $\underline{\mathbf{w}}_i$ of the origins, and by the normalized Euler parameters (unit quaternions) $\mathcal{R}_i^0$ which are associated with the corresponding rotation matrices. From (7) and (9) we obtain two solutions, corresponding to a pair of antipodal points on the unit quaternion sphere $\mathbb{S}^3 \subset \mathbb{R}^4$ $(\mathbb{H})$. We pick one of those points (i.e., the sign of the normalized Euler parameters) such that neighbouring points $\mathcal{R}_i^0, \mathcal{R}_{i+1}^0$ on $\mathbb{S}^3$ belong to one hemisphere, that is the inner product of the corresponding vectors in $\mathbb{R}^4$

$$\langle \mathcal{R}_i^0, \mathcal{R}_{i+1}^0 \rangle = \frac{1}{2}(\mathcal{R}_i^0 * \overline{\mathcal{R}_{i+1}^0} + \mathcal{R}_{i+1}^0 * \overline{\mathcal{R}_i^0}) \geq 0 \tag{30}$$

   should be non–negative.

   Secondly, we need to associate parameter values t_i with the given positions. Similar to the methods for parameterizing point data (see [16, Section 4.4.1]), these parameters can be estimated from the distances between the given positions. In addition to the distance of the origins, the difference of the orientations should be taken into account. For instance, by generalizing the chordal parameterization, one may choose the differences $t_{i+1} - t_i$ proportional to

$$||\underline{\mathbf{w}}_{i+1} - \underline{\mathbf{w}}_i|| + \omega \arccos \langle \mathcal{R}_i^0 \mathcal{R}_{i+1}^0 \rangle \tag{31}$$

with some weight $\omega > 0$ controlling the influence of the spherical part. Several other possibilities are listed in [20].

2. **Interpolation of the spherical part.** As the next step, we compute the preimage curve $\mathcal{Q}(t)$ of the kinematical mapping from the interpolation conditions $\mathcal{Q}(t_i) = \mathcal{R}_i^0$. This curve is a rational spline curve of degree d in elliptic 3–space, cf. (21). Knots and degree are chosen such that the number of degrees of freedom equals the number of unknowns, where additionally the Schoenberg-Whitney conditions (see Chapter **?**) are to be satisfied. For example, one may choose a cubic spline curve with not–a–knot type boundary conditions, producing the knot vector

   $$(\underbrace{t_1, \ldots, t_1}_{\text{4-fold knot}}, \underbrace{t_3, t_4, \ldots, t_{N-2}}_{\text{single knots}}, \underbrace{t_N, \ldots, t_N}_{\text{4-fold knot}}) \tag{32}$$

   The control quaternions $\mathcal{B}_i$ of the preimage curve can be found by solving the resulting banded system of linear equations, see Chapter **?**.
   Now we generate the spherical part of the interpolating motion by applying the kinematical mapping of spherical kinematics to the preimage spline curve. This results in a spherical rational motion of degree $2d$, described by the spline functions $u_0(t)$ and $U(t)$, see (25).

3. **Translational part.** For the sake of simplicity we may choose $v_0^*(t) \equiv 1$. In order to interpolate the translational parts of the given positions, we have to find spline functions $v_1(t)$, $v_2(t)$, $v_3(t)$ satisfying the interpolation conditions $\underline{\mathbf{v}}(t_i) = \underline{\mathbf{w}}_i$, where

   $$\underline{\mathbf{v}}(t) = (\ v_1(t)/u_0(t),\ v_2(t)/u_0(t),\ v_3(t)/u_0(t)\ )^\top \tag{33}$$

   is the trajectory of the origin. It seems to be a natural choice to choose spline functions $v_1(t)$, $v_2(t)$, $v_3(t)$ of degree $2d$ whose knots are those of the spline functions $u_0(t)$ and $U(t)$. This choice, however, leads to an underdetermined system of linear equations, as the translational part of the motion has far more degrees of freedom than the spherical one. Consequently, additional constraints are needed to pick a unique solution. These may be not–a–knot type conditions at inner knots, enforcing higher order of differentiability for the trajectory of the origin. Alternatively, one may use the additional degrees of freedom for minimizing quadratic 'energy' functionals, such as

   $$\int_{t_0}^{t_N} ||\ddot{\underline{\mathbf{v}}}(t)||^2 dt \to \text{Min.} \tag{34}$$

   See Chapter **?** for further information on this technique. In either case, the resulting interpolating spline motion of degree $2d$ is found by solving a system of linear equations.

Various algorithms for interpolation with spline curves can be generalized to the kinematical setting, simply by applying them to the preimage curve of the kinematical mapping, and combining the result with a suitable trajectory of the origin. For instance, a kinematical version of *cubic Hermite splines* has been implemented as part of a commercial robot controller [15]. This leads to rational spline motions whose spherical part has degree 6. Cubic Hermite splines provide some features which are essential in this application, such as real–time capability and certain shape–preserving properties. Compared with traditional techniques, the use of spline curves leads to a substantial reduction of the data volume, and to enhanced programming of robot motions, in particular for the manufacturing of free-form shapes.

Further computational techniques for rational spline motions include the optimization ('fairing') of motion segments, to generate spherical motions that minimize (e.g.) the integral of the squared angular acceleration (which can be seen as the analogue of cubic spline curves). Moreover an algorithm for spline motion fitting has been used to reconstruct the motion of the human knee joint from measurement data. See [19,22,24] for additional information.

Although the above interpolation scheme for rational spline motions has many desirable features, it is not fully satisfying from the theoretical point of view, as it lacks what was called 'invariance with respect to parameterization' or *parameter invariance* by Röschel [12,43]. If we sample data (positions with associated parameters t_i) from a rational spline motion and apply the interpolation procedure, including the preprocessing step, then it will generally not reproduce the original motion, even if the same spline spaces are used. This is due to the fact that the normalization (7) is only valid at the original interpolation nodes, and not everywhere. In order to guarantee the reproduction property one would need to use rational curves on the unit quaternion sphere $\mathbb{S}^3 \in \mathbb{R}^4$.

Such curves can be generated with the help of stereographic projection, but then the results depend on the choice of the coordinates, as it is the case for the method described in [17].

In the case of the sphere in 3–space, the *generalized stereographic projection* can be shown to give results which are independent of the chosen system of coordinates, see Chapter **?**, [6]. Unfortunately, similar results for spheres in higher dimensions are currently not available. Recently, Gfrerrer [12] has developed a new algorithm for interpolation with rational curves on hyperspheres of arbitrary dimension, producing coordinate–independent results. However, the degree of the resulting spherical motion is about twice as high as the one which would result from th earlier algorithm, and it may happen that no solutions exist. Furthermore, the generalization of Gfrerrer's method to rational *spline* curves is still an open problem.

6.6. Rational frames and sweeping surfaces

The kinematical version of Hermite interpolation with cubic C^1 splines has also been used to generate highly accurate rational approximation of *rotation minimizing frames*, using spatial rational spline motions of degree 6 [23]. In geometric modeling, the rotation minimizing frame has been introduced by Klok as an alternative to the Frenet frame of a space curve [13,29] . It is useful for sweep surface modeling, as it provides a robust and intuitive way of moving a profile curve along a given 'spine' curve, see [48] for examples.

As an alternative to the approximation of frames, one may also study spatial curves which have an associated rational frame. A spatial rational motion is called a *rational frame* (see Figure 13) of a given space curve, if the origin of the moving system travels along that curve, and if additionally the tangent vector of the curve is always parallel to the (say) $\hat{x}_1$–axis of the moving system.

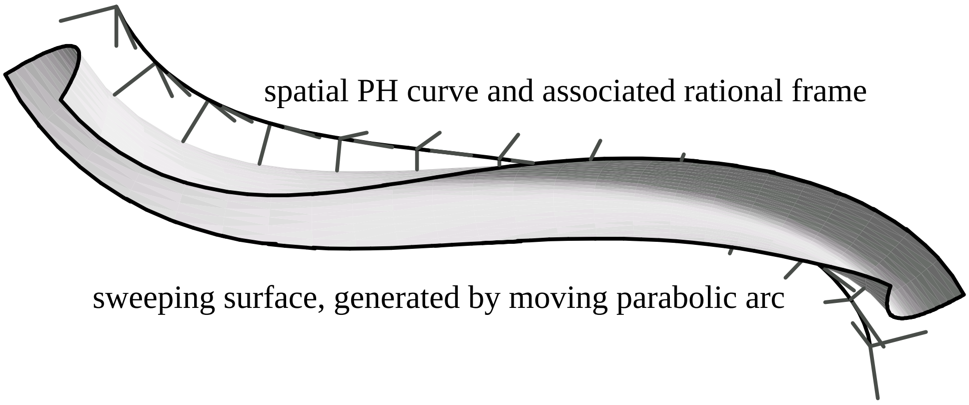

spatial PH curve and associated rational frame
sweeping surface, generated by moving parabolic arc

Figure 13. Rational frame of a PH curve of degree 7 and a rational sweeping surface.

Note that these frames are closely related to *Pythagorean hodograph (PH) space curves* , see Section 2.3 of Chapter **?**. In fact, the parametric representation of a polynomial PH curve in 3–space can be generated by integrating the hodograph

$$\dot{\underline{\mathbf{x}}}(t) = U(\mathcal{Q}(t))\,(1,0,0)^\top = (\,q_0^2{+}q_1^2{-}q_2^2{-}q_3^2,\ 2(q_1q_2{+}q_0q_3),\ 2(q_1q_3{-}q_0q_2)\,)^\top, \tag{35}$$

cf. (19) and Eq.(10) of Chapter **?**, where $\mathcal{Q}(t)$ is an arbitrary preimage curve in elliptic 3–space. For instance, the rational frame in Figure 13 has been generated from the spherical motion shown in Figure 6. The resulting curves are automatically equipped with rational frames, which can be obtained by combining the trajectory of the origin with the spherical rational motion $\underline{U}(t)$, see (18). Recently, even more sophisticated classes of rational space curves have been studied, providing a *rational Frenet frame* or a *rational rotation–minimizing frame*, see [38,39,47].

Spatial rational motions can be used to generate *sweeping surfaces.* These surfaces are generated by moving a fixed profile curve through 3–space, see [8, Preface (written by P. Bézier), Fig 12] for an illustration. Rational sweeping surfaces have been studied in [42,21]. The class of sweeping surfaces can be generalized by allowing simultaneous changes of the moving profile curve. This leads to 'generalized cylinders', which have been shown to be a useful tool for the interactive modelling of free–form shapes. Again, the underlying rigid body motion can efficiently be described in rational (B–) spline form, see [4].

7. CLOSURE

Based on rational spline techniques and the kinematical mapping, we have shown that computational methods for spatial rigid body motions can be obtained by generalizing the powerful techniques of Computer Aided Geometric Design. We conclude this chapter by listing a few possible topics for further research.

- *Generating optimal motions.* An interesting problem for applications in robotics and NC machining is the efficient generation of energetically or time–optimal motions, taking the robot geometry into account, and their use for robot control. This may help to reduce cycle times in manufacturing, and to increase the lifetime of the machinery. A related problem is the optimal generation of paths for NC milling, cf. [28].
- *Advanced CAD/CAM interfaces.* Currently, the sophisticated geometry models of CAD are mostly converted to piecewise linear or circular descriptions of tool paths for Numerically Controlled (NC) machining. Here, due to the advancing processor speed, it is now possible to use more advanced geometric models. First attempts in this direction include the use of Pythagorean hodograph curves in NC milling (see Chapter **?**), and spline interpolation for robot motion planning [15].
- *Simulation of machining.* This is related to a third challenging problem. Advanced methods for computer–aided simulation of manufacturing processes (e.g. milling) may help to optimize these processes, and to check the quality of the results. For instance, it would be interesting to be able to generate an accurate representation of the surface produced by the cutter of a milling machine, in order to check the quality of the results.

As a promising direction for further research, the advanced techniques for describing in Computer Aided Geometric Design should now be applied to problems from other areas, such as computer–aided manufacturing and numerical simulation in scientific computing. We are convinced that spatial rational spline motions are well suited for this forthcoming task.

REFERENCES

1. A. Barr, B. Currin, S. Gabriel, and J. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions. *Computer Graphics*, 26:313–320, 1992. (SIGGRAPH '92).
2. W. Blaschke. *Kinematik und Quaternionen.* Deutscher Verlag der Wissenschaften, Berlin, 1960.
3. O. Bottema and B. Roth. *Theoretical kinematics.* Dover Publications, New York, 1990. (Corr. reprint of the 1979 edition).
4. T.-I. Chang, J.-H. Lee, M.-S. Kim, and S. Hong. Direct manipulation of generalized cylinders based on B-spline motion. *The Visual Computer*, 14:228–239, 1998.
5. G. Darboux. Note III: Sur les mouvements algébriques. In *Leçons de cinématique (G. Kœnigs)*, pages 352–353. Hermann, Paris, 1895.
6. R. Dietz, J. Hoschek, and B. Jüttler. An algebraic approach to curves and surfaces on the sphere and on other quadrics. *Comput. Aided Geom. Design*, 10:211–229, 1993.

7. G. Farin. Algorithms for rational Bézier curves. *Comput.-Aided Design*, 15:73–77, 1983.
8. G. Farin. *Computer Aided Geometric Design – a practical guide.* Academic Press, Boston, 1993.
9. J. Gallier. *Geometric Methods and Applications.* Springer, New York, 2000.
10. Q. Ge and B. Ravani. Computer aided geometric design of motion interpolants. In *Proc. ASME Design Automation Conf., DE–Vol. 32–2*, pages 33–41. ASME, 1991.
11. G. Geise and B. Jüttler. A geometrical approach to curvature continuous joints of rational curves. *Comput. Aided Geom. Design*, 10:109–122, 1993.
12. A. Gfrerer. Rational interpolation on a hypersphere. *Comput. Aided Geom. Design*, 16:21–37, 1999.
13. H. Guggenheimer. Computing frames along a trajectory. *Comput. Aided Geom. Design*, 6:77–78, 1989.
14. A. Hanson. Visualizing quaternions. Course at SIGGRAPH 2000, New Orleans, 2000.
15. T. Horsch and B. Jüttler. Cartesian spline interpolation for industrial robots. *Comput.-Aided Design*, 30:217–224, 1998.
16. J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design.* AK Peters, Wellesley MA, 1993.
17. J. Johnstone and J. Williams. Rational control of orientation for animation. In *Graphics Interface '95 (W.A. Davis and P. Prusinkiewicz, eds.)*, pages 179–186. Canadian Human–Computer Communication Society, Toronto, 1995.
18. B. Jüttler. Über zwangläufige rationale Bewegungsvorgänge. *Sb. Österr. Akad. Wiss., Abt. II*, 202:117–132, 1993.
19. B. Jüttler. *Rationale Bézierdarstellung räumlicher Bewegungsvorgänge und ihre Anwendung zur Beschreibung bewegter Objekte.* PhD thesis, Darmstadt University of Technology, 1994. (Published at Shaker Verlag, Aachen).
20. B. Jüttler. Visualization of moving objects using dual quaternion curves. *Computers & Graphics*, 18.3:315–326, 1994.
21. B. Jüttler. Spatial rational motions and their application in computer aided geometric design. In *Mathematical Methods for Curves and Surfaces (M. Dæhlen et al., eds.)*, pages 271–280. Vanderbilt University Press, Nashville, TN, 1995.
22. B. Jüttler and M. Wagner. Computer aided design with spatial rational B-spline motions. *ASME J. Mechanical Design*, 118:193–201, 1996.
23. B. Jüttler and M. Wagner. Rational motion-based surface generation. *Comput.-Aided Design*, 31:203–213, 1999.
24. C. Keil. Approximation gemessener Kniebewegungen. Diplomarbeit, Dept. of Mechanics, Darmstadt University of Technology, 1995.
25. M.-J. Kim, M.-S. Kim, and S. Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. *Computer Graphics*, 29:369–376, 1995. (SIGGRAPH '95).
26. M.-J. Kim, M.-S. Kim, and S. Shin. A compact differential formula for the first derivative of a unit quaternion curve. *J. of Visualization and Computer Animation*, 7:43–57, 1996.
27. M.-S. Kim and K.-W. Nam. Interpolating solid orientations with circular blending quaternion curves. *Comput.-Aided Design*, 27:385–398, 1995.

28. T. Kim and S. Sarma. Time–optimal paths covering a surface. In *The Mathematics of Surfaces IX (R. Cipolla and R.R. Martin, eds.)*, pages 126–143. Springer, London, 2000.
29. F. Klok. Two moving coordinate frames for sweeping along a 3D trajectory. *Comput. Aided Geom. Design*, 3:217–229, 1986.
30. J. Kuipers. *Quaternions and rotation sequences. A primer with applications to orbits, aerospace, and virtual reality.* Princeton University Press, 1998.
31. S. Mick and O. Röschel. Interpolation of helical patches by kinematic rational Bézier patches. *Comput. & Graphics*, 14:275–280, 1990.
32. K. Mørken. Some identities for products and degree raising of splines. *Constructive Approximation*, 7:195–208, 1991.
33. G. Nielson and R. Heiland. Animated rotations using quaternions and splines on a 4D sphere. *Program. Comput. Softw.*, 18:145–154, 1992. (Translated from *Programmirovanie*, 4:17–27, 1992).
34. L. Noakes. Non–linear corner cutting. *Adv. Comput. Math.*, 8:165–177, 1998.
35. F. Park and B. Ravani. Bézier curves on Riemannian manifolds and Lie groups with kinematic applications. *ASME J. of Mechanical Design*, 115:36–40, 1995.
36. D. Pletinckx. Quaternion calculus as a basic tool in computer graphics. *The Visual Computer*, 5:2–13, 1989.
37. H. Pottmann. Studying NURBS curves and surfaces with classical geometry. In *Mathematical methods for curves and surfaces (M. Daehlen et al., eds.)*, pages 413–438. Vanderbilt University Press, Nashville TN, 1994.
38. H. Pottmann and M. Wagner. Principal surfaces. In *The Mathematics of Surfaces VII, (T.N.T. Goodman and R.R. Martin, eds.)*, pages 337–362. Information Geometers, Winchester, 1997.
39. H. Pottmann and M. Wagner. Contributions to motion based surface design. *Int. J. of Shape Modeling*, 4:183–196, 1998.
40. R. Ramamoorthi and A. Barr. Fast construction of accurate quaternion splines. *Computer Graphics*, 31:287–292, 1997. (SIGGRAPH '97).
41. O. Röschel. Rationale räumliche Zwangläufe vierter Ordnung. *Sb. Österr. Akad. Wiss., Abt. II*, 194:185–202, 1985.
42. O. Röschel. Kinematic rational Bézier patches I/II. *Rad Hrvat. Akad. Znan. Umjet. Mat. Znan.*, 456:95–108 (I), 131–138 (II), 1991.
43. O. Röschel. Rational motion design – a survey. *Comput.–Aided Design*, 30:169–178, 1998.
44. K. Shoemake. Animating rotations with quaternion curves. *Computer Graphics*, 19:245–254, 1985. (SIGGRAPH '85).
45. M. Wagner. *A Geometric Approach to Motion Design.* PhD thesis, TU Wien, Institute of Geometry, 1994.
46. M. Wagner. Planar rational B-spline motions. *Comput.–Aided Design*, 27:129–137, 1995.
47. M. Wagner and B. Ravani. Curves with rational Frenet–Serret motion. *Comput. Aided Geom. Design*, 15:79–101, 1997.
48. W. Wang and B. Joe. Robust computation of the rotation minimizing frame for sweep surface modeling. *Comput.–Aided Design*, 29:379–391, 1997.

49. E. Weiss. *Einführung in die Liniengeometrie und Kinematik.* Teubner, Leipzig, 1935.
50. W. Wunderlich. Kubische Zwangläufe. *Sb. Österr. Akad. Wiss., Abt. II*, 193:45–68, 1984.
51. J. Xia and Q. Ge. On the exact representation of the boundary surfaces of the swept volume of a cylinder undergoing rational Bézier and B-spline motions. In *ASME Design Engineering Technical Conferences.* ASME, Las Vegas, 1999. (Paper no. DETC99/DAC–8607).

Index