# Bézier motions with end-constraints on speed

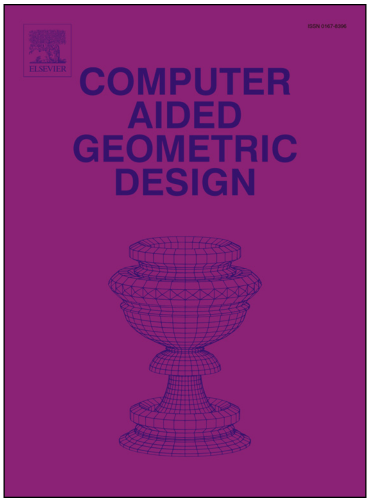
Mullineux, Glen; Cripps, Robert; Cross, Ben

*DOI:*
10.1016/j.cagd.2018.04.003

*Citation for published version (Harvard):*
Mullineux, G, Cripps, R & Cross, B 2018, 'Bézier motions with end-constraints on speed', *Computer Aided Geometric Design*, vol. 63, pp. 135-148. https://doi.org/10.1016/j.cagd.2018.04.003

Bézier motions with end-constraints on speed

Glen Mullineux, Robert J. Cripps, Ben Cross



PII: S0167-8396(18)30049-9
DOI: https://doi.org/10.1016/j.cagd.2018.04.003
Reference: COMAID 1686

To appear in: *Computer Aided Geometric Design*

Received date: 23 May 2017
Revised date: 19 April 2018
Accepted date: 20 April 2018

Please cite this article in press as: Mullineux, G., et al. Bézier motions with end-constraints on speed. *Comput. Aided Geom. Des.* (2018), https://doi.org/10.1016/j.cagd.2018.04.003

## Highlights

- Finding a motion for given precision poses and speeds is discussed
- Additive combinations of control poses are used within a geometric algebra
- Analogies with the case of curve-fitting are shown

# Bézier motions with end-constraints on speed

Glen Mullineux
Department of Mechanical Engineering
University of Bath
Bath BA2 7AY, UK

Robert J. Cripps, Ben Cross
School of Mechanical Engineering
University of Birmingham
Birmingham B15 2TT, UK

*Abstract*

A free-form motion can be considered as a smoothly varying rigid-body transformation. Motions can be created by establishing functions in an appropriate space of matrices. While a smooth motion is created, the geometry of the motion itself is not always immediately clear. In a geometric algebra environment, motions can be created using extensions of the ideas of Bézier and B-spline curves and the geometric significance of the construction is clearer. A motion passing through given precision poses can be obtained by direct analogy with the curve approach. This paper considers the more difficult problem of dealing additionally with velocity constraints at the ends of the motion: here the analogy is less obvious. A geometric construction for the end pairs of control poses is established and is demonstrated by creating motions satisfying given pose and velocity constraints.

*Keywords*: free-form motion, Bézier motion, velocity constraint, geometric algebra.

## 1. Introduction

The ideas of free-form curves and surfaces are well understood. The Bézier and B-spline forms [1] allow such objects to be defined and manipulated using control points that reflect the inherent geometry. These ideas can be extended to describing and manipulating free-form motions. These have applications in such areas as robotics [2, 3], cutter paths in manufacturing [4], mechanisms [5, 6], neuroscience [7], and motion of spacecraft [8]. A free-form motion is where a body moves smoothly through space. The points in the body move (mainly) along smooth curves. At each instant of a motion, the body has a position and orientation is space: this is called a *pose*. A rigid-body transform is one which moves a body from one pose to another without distorting it. A motion can be regarded as a smoothly varying rigid-body transform which moves a body from some reference pose to a varying pose in space.

There are a number of ways of representing rigid-body motions. Variable 4 × 4 matrices [9, 10] can have problems with robustness and clarity of geometric meaning of smoothing metrics. Quaternions [9, 11, 12] handle rotations, and their extension to dual quaternions [7, 13, 14] allows translations as well. The slerp (spherical linear interpolation) construction [11] generates free-form motions based on combining control poses multiplicatively, although this requires the use of exponentiation and logarithms. Geometric algebra approaches allow geometry and transforms to be handled in the same environment [15, 16] and have been used with a range of application areas [17, 18, 19]. In particular, transforms can be combined additively so that free-form motions can be created using the Bézier approach as an alternative to the slerp. In specifying free-form motions, one approach is to deal separately with the (translational) motion of a reference point in the moving body and the (rotational) motion of the body with respect to it [20, 21, 22]. An alternative is to handle the motion of the body as a single variable transform [5, 23, 24].

The purpose of this paper is to present a novel construction of a motion where the constraints are given not only on precision poses through which the motion must pass [23], but also on linear and angular velocities at points in the motion. This is for a variable transform of the moving body expressed as an additive combination of control poses in Bézier form within a geometric algebra. A construction is proposed and verified to establish the pairs of control poses at the ends of the motion needed to satisfy the speed constraints.

Section 2 reviews the ideas of geometric algebra and gives an overview of the form of algebra used here. This includes the use of the algebra for dealing with rigid-body transforms and section 3 discusses the extension to Bézier motions.

Obtaining smooth curves and motions often requires control of the relevant derivatives. Such derivatives have been investigated for various representations of motion [11, 20, 25, 26]. Section 4 reviews the derivative of the representation used here. This is in terms of the tangent conditions at the ends of a rational Bézier curve and these ideas are used in section 5 to investigate how the first (and last) pair of control poses influence the motion. This leads to the geometric construction of these poses. Section 6 gives some examples. These are based on published cases of free-form motions generated using matrix approaches (optimizing matrix-based metrics). In these previous cases, several motions satisfying the imposed constraints are provided (with no indication that any are better than others). The examples generated by the method presented here do not reproduce (exactly) any of these: they are not intended to. However the examples given here are certainly as good visually as those previously obtained and have the advantage that the links to the geometry of the resultant motion are more obvious.

The significance of this paper is in generating free-form Bézier motions which interpolate end-motion constraints on poses and derivatives. This will have impact in high-valued manufacturing processes that require the control of mechanisms, for example, CNC machining, robotics and human motion analysis.

## 2. Geometric Algebra and Transforms

There are a number of versions of quaternions and geometric algebra [15, 27, 28] that are available, including the commonly-used conformal geometric algebra (CGA) [16, 29]. The version used here is the algebra called $\mathcal{G}_4$ [30, 31] although, with appropriate adjustment, it is straightforward to use the dual quaternions or other versions of geometric algebra. The reasons for the choice of $\mathcal{G}_4$ are given in the appendix and the following is an overview of its construction.

It starts with a real vector space with basis vectors denoted by $e_0$, $e_1$, $e_2$, $e_3$. The space is extended to one of dimension 16 by using basis elements denoted by $e_\sigma$ where $\sigma$ is an ordered subset of the set of subscripts $\{0, 1, 2, 3\}$. This allows an anticommutative multiplication to be defined on the basis elements so that $e_i e_j = e_{ij} = -e_j e_i$ for $i \neq j$. The basis element corresponding to the empty set behaves like the real number 1 and is identified with it: $e_\phi = 1$.

The squares of the basis vectors are defined as

$$e_0^2 = \varepsilon^{-1} \; , \qquad\qquad e_1^2 = e_2^2 = e_3^2 = 1 \; ,$$

where $\varepsilon$ is a symbol representing a (vanishingly) small positive real number. This effectively allows the square of $e_0$ to represent infinity and this in turn allows points in $\mathcal{G}_4$ to correspond to points in three-dimensional space (cf. appendix). The basis element $e_{0123}$ is *the pseudoscalar* and is denoted by $\omega$; the multiplication rules show that $\varepsilon\omega^2 = 1$.

The typical element $a \in \mathcal{G}_4$ has the form

$$a \;=\; \sum_\sigma a_\sigma e_\sigma \tag{1}$$

and the coefficients $a_\sigma$ can be regarded as elements of the field of Laurent power series of the form

$$\sum_{i=m}^{\infty} \alpha_i \varepsilon^i \tag{2}$$

where $m$ is a finite (possibly negative) number and the $\alpha_i$ are real numbers. A coefficient of the form (2) is called a *scalar*.

An inner and outer product are defined as follows for any two elements $x, y \in \mathcal{G}_4$. Note that these are not the same definitions as used elsewhere [27], but they are helpful in simplifying expressions later in this section

$$\begin{aligned} x \cdot y \;&=\; \tfrac{1}{2}(xy + yx) \; , \\ x \wedge y \;&=\; \tfrac{1}{2}(xy - yx) \; . \end{aligned}$$

The *grade* of a basis element is the number of its subscripts. If an element $a \in \mathcal{G}_4$ is a combination only of basis elements of a single grade, then this is also the grade of

$a$. Elements of $\mathcal{G}_4$ of grade one are called *vectors*; those of grade two are *bivectors*. The typical vector $P \in \mathcal{G}_4$ has the form

$$P \;=\; We_0 + Xe_1 + Ye_2 + Ze_3$$

and corresponds to the column vector $[X, Y, Z, W]^T$ in the projective space $\mathbb{RP}^3$, and hence to the point $(X/W, Y/W, Z/W)$ in real three-dimensional space $\mathbb{R}^3$ (assuming that $W$ is non-zero).

The following notation is introduced to identify the "homogeneous" coordinate of a vector

$$w(P) \;=\; W\;.$$

The *reverse* of a basis element $e_\sigma$ is obtained by reversing the order of its subscripts. The idea extends to the general element $a \in \mathcal{G}_4$ by taking the reverse of each term in equation (1) and it is denoted by $\overline{a}$. It has the property that $\overline{ab} = \overline{b}\,\overline{a}$ for any elements $a, b \in \mathcal{G}_4$.

**Lemma 2.1.** *If $U, V \in \mathcal{G}_4$ are even-grade elements, and $P \in \mathcal{G}_4$ is a vector, then the combinations $\overline{U}PU$ and $\frac{1}{2}(\overline{U}PV + \overline{V}PU)$ are also vectors.*

*Proof.* Each of the combinations is a term of odd grade which is equal to its own reverse. Hence each is a vector. □

The elements of even grade form a subalgebra of $\mathcal{G}_4$ which is a vector space of dimension eight with basis: 1, $e_{01}$, $e_{02}$, $e_{03}$, $e_{12}$, $e_{13}$, $e_{23}$, $\omega$.

Further, for an even-grade element $S \in \mathcal{G}_4$, the map

$$F_S : P \mapsto \overline{S}PS$$

is a linear transformation on $\mathbb{RP}^3$ and creates a rigid-body transform on $\mathbb{R}^3$ [31].

An element of the form $a + \varepsilon b\omega \in \mathcal{G}_4$ where $a$ and $b$ are scalars is called *a pseudoscalar*. Any non-zero pseudoscalar generates the identity transform. The pseudoscalars form a vector subspace of dimension two of the space of even-grade elements. Since there are six degrees of freedom determining a rigid-body transform (three for the translation components, and three for the rotation), the following result is proved.

**Lemma 2.2.** *The even-grade elements of $\mathcal{G}_4$ generate all the rigid-body transforms of $\mathbb{R}^3$.*

□

From [31], if $a = a_1e_1 + a_2e_2 + a_3e_3$ is a unit vector representing the direction of an axis through the origin, then the even-grade element generating a rotation through angle $2\theta$ about the axis is

$$R \;=\; c + sb \tag{3}$$

where $c = \cos\theta$, $s = \sin\theta$, and $b = ae_{123}$ is a unit bivector.

Further, if $u = u_1e_1 + u_2e_2 + u_3e_3$ is a vector, then the even-grade element generating a translation through $2u$ is

$$T \;=\; 1 + \varepsilon e_0 u \; . \tag{4}$$

The doubling of the angle of rotation and the distance of translation occurs because of the presence of $S$ twice in the product forming the map $F_S$.

Any even-grade element $S \in \mathcal{G}_4$ can be written as a product $S = \lambda RT$ where $R$ and $T$ have the above forms and $\lambda$ is a pseudoscalar. Further, if $S$ is normalized so that $\overline{S}S = 1$, then $\lambda = 1$.

By multiplication

$$RT \;=\; c + sb + \varepsilon ce_0u + \varepsilon se_0bu \; .$$

The product in the other order is

$$TR \;=\; (1 + \varepsilon e_0u)R \;=\; R(1 + \varepsilon e_0\overline{R}uR)$$

and hence is the product of a rotation and a translation in that order.

**Lemma 2.3.** *The product $T_1RT_2$ of a rotation $R$ about an axis through the origin and translations $T_1$ and $T_2$ is the same as the product $RT$ of the same rotation and a translation $T$.*

□

**Lemma 2.4.** *Suppose $P \in \mathcal{G}_4$ is a point and $U, V \in \mathcal{G}_4$ are even-grade elements with $\overline{U}U = 1 = \overline{V}V$. Then*

*(i)* $w(\overline{U}PU) = w(P)$

*(ii) if $Q = \frac{1}{2}(\overline{U}PV + \overline{V}PU)$, then $Q$ is a point with $w(Q) = cw(P)$ where $c$ is the cosine of half the angle of the rotation involved in $V\overline{U}$.*

*Proof.* Without loss of generality, assume that $P = e_0 + v$, so that $w(P) = 1$, where $v$ is a vector not involving $e_0$. For (i), $U = RT$ where $R$ and $T$ are given by equations (3) and (4) and it suffices to prove the result for each of these. For $U = R$, the result is immediate since

$$\overline{R}PR = e_0 + \overline{R}vR \; .$$

For $U = T = 1 + \varepsilon e_0u$, the product is

$$\overline{T}PT = [1 - \varepsilon u^2 - 2\varepsilon(u \cdot v)]e_0 + (2u + v) - \varepsilon uvu$$

and the result holds since $\varepsilon$ is (vanishingly) small.

For (ii), first note that

$$Q \;=\; \tfrac{1}{2}\overline{U}(PS+\overline{S}P)U$$

where $S = V\overline{U}$, and so it is sufficient to prove the result for $Q = \frac{1}{2}(PS+\overline{S}P)$. Taking $S = RT$ with $R$ and $T$ as in equations (3) and (4), $Q$ evaluates as follows

$$Q \;=\; [c - \varepsilon c(u\cdot v) + \varepsilon s\alpha]e_0 + cu + cv + s[(b\wedge u) - (b\wedge v)]$$

where $b$, $u$, $v$ do not involve $e_0$, and $\alpha = \frac{1}{2}(ubv - vbu)$ which is scalar as it has even grade and equals its own reverse. The result now follows. □

## 3. Bézier Motions

An even-grade element $S \in \mathcal{G}_4$ generates a rigid-body transform. This can be applied to the points in a body and hence move that body to a new pose. The term *pose* is now also used to refer to the even-grade element $S$ itself. A *motion* is a smoothly varying rigid-body transform. It can be generated by using a smoothly varying even-grade element $S(t) \in \mathcal{G}_4$ where $t$ is a real parameter.

| | | | |
|---|---|---|---|
| $S_0$ | | | |
| | $S_{01}$ | | |
| $S_1$ | | $S_{012}$ | |
| | $S_{12}$ | | $S_{0123}$ |
| $S_2$ | | $S_{123}$ | |
| | $S_{23}$ | | |
| $S_3$ | | | |

Figure 1: de Casteljau tableau of control poses

Suppose that (say) four *control poses* $S_0$, $S_1$, $S_2$, $S_3$ are given. A de Casteljau tableau can be formed as in Fig. 1. where each new entry is a combination of the two poses to its left. The combination depends upon a parameter $t$ so that $S_{0123}$ is a function of $t$.

Suppose that $U$ and $V$ are two even-grade elements. They can be combined in two ways. Following the corresponding ideas for quaternions [11], the first way is as a *slerp* (spherical linear interpolation)

$$S(t) \;=\; U\,(\overline{U}V)^t\ .$$

Clearly, $S(0) = U$ and $S(1) = (\overline{U}U)V$ which is a scalar multiple of $V$. (For convenience, it is possible to multiply $U$ by a scalar and so normalize it so that $\overline{U}U = 1$.) Hence $S(t)$ interpolates the two poses.

The slerp can be used to combine poses in the de Casteljau algorithm. However, it requires use of exponentiation and logarithms [23]. Additionally, the algorithm leads to complicated products in the control poses which are difficult to deal with since the multiplication is not commutative.

So, instead, a second way of combining poses is used here. It is additive and leads to Bézier motions. If $U$ and $V$ are two poses, the combination is

$$S(t) = (1-t)U + tV \, .$$

Again $S(0) = U$ and $S(1) = V$, so that $S(t)$ interpolates the two poses. This is the basic combination for Bézier motions. The ideas extend to more general B-spline motions. In particular, a more general additive combination is obtained, involving knot values [1]. Given $n+1$ control poses (even-grade elements), $S_0, S_1, \ldots, S_n$, the additive combination in the de Casteljau algorithm generates the following linear combination

$$S(t) \;=\; \sum_{j=0}^{n} N_j(t)\, S_j \tag{5}$$

where (in the more general case) the $N_j(t)$ are the B-spline basis functions for the appropriate degree $d$ [1]. (It is this form that is used to generate motions using dual quaternions in [13].)

The interest of this paper is in Bézier motions where $d = n$ and the basis functions are given by

$$N_j(t) \;=\; \binom{n}{j} (1-t)^{n-j} t^j \qquad \text{for } \; 0 \le j \le n, \;\; 0 \le t \le 1 \, . \tag{6}$$

Fig. 2 shows an example of a Bézier cubic motion with the four control poses shown with thicker lines.

It is straightforward to generate a motion passing through a number of prescribed precision poses [23, 32]. It is less clear how to deal with motions which are also required to meet derivative constraints (that is speeds). The angular speed is a property of the moving body itself. It is perhaps natural to specify linear speeds in terms of the speeds of specific points in the body. These move along free-form curves.

**Lemma 3.1.** *Suppose that $P \in \mathcal{G}_4$ is a point in a body undergoing a Bézier motion $S(t)$ of degree $n$ given by equations (5) and (6). Then the point traces out the parametric curve*

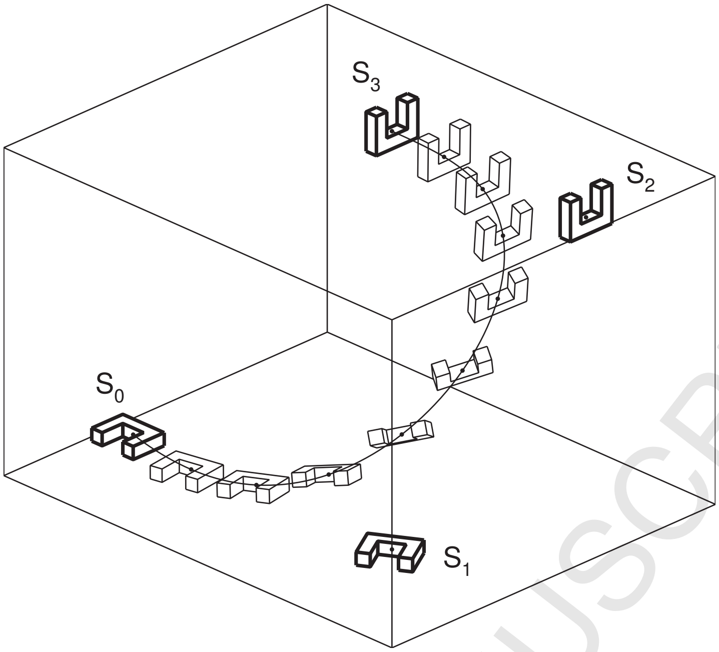


Figure 2: Bézier cubic motion for four control poses shown using thicker lines

$$Q(t) \;=\; \overline{S(t)}PS(t) \;=\; \sum_{i=0}^{2n} \binom{2n}{i} (1-t)^{2n-i} t^i Q_i \;, \tag{7}$$

*which is a Bézier curve of degree* $2n$ *whose control points have the form*

$$Q_i \;=\; \sum_{j=\max(i-n,0)}^{\min(i,n)} a_{i,j}\overline{S_j}PS_{i-j}$$

*where the coefficients*

$$a_{i,j} \;=\; \frac{\binom{n}{j}\binom{n}{i-j}}{\binom{2n}{i}}$$

*have the property that* $\sum_j a_{i,j} = 1$ *for all* $i$*.*

*Proof.* Expanding the product $\overline{S(t)}PS(t)$ gives

$$
\begin{aligned}
Q(t) &= \sum_{j=0}^{n}\sum_{k=0}^{n} N_j(t)N_k(t)\overline{S_j}PS_k \\
&= \sum_{j=0}^{n}\sum_{k=0}^{n} \binom{n}{j}\binom{n}{k}(1-t)^{2n-j-k}t^{j+k}\overline{S_j}PS_k
\end{aligned}
$$

and setting $i = j + k$ and rearranging gives the required form in terms of the $Q_i$. In the summation for $Q_i$, whenever $\overline{S_j}PS_{i-j}$, with $j \neq i - j$, appears, so too does $\overline{S_{i-j}}PS_j$, with the same coefficient $a_{i,j} = a_{i,i-j}$. Hence lemma 2.1 shows that $Q_i$ is a sum of vectors and hence represents a point.

That the sum of the coefficients is unity follows by considering the binomial expansions of both sides of the identity $[(1-t)+t]^n[(1-t)+t]^n = [(1-t)+t]^{2n}$. □

**Corollary 3.2.** *The first two control points of the Bézier curve $Q(t)$ in equation (7) are $\overline{S_0}PS_0$ and $\frac{1}{2}[\overline{S_0}PS_1 + \overline{S_1}PS_0]$; and the last two control points are $\frac{1}{2}[\overline{S_{n-1}}PS_n + \overline{S_n}PS_{n-1}]$ and $\overline{S_n}PS_n$.*

*Proof.* This follows by evaluating $Q_i$ for $i = 0, 1, 2n-1, 2n$. □

## 4. End Curve Derivatives

The interest is in applying derivative constraints at the beginning and end of a motion. This section considers the end-derivatives of a rational Bézier curve.

Such a curve, of degree $m$, can be considered as one in projective space

$$
\begin{aligned}
Q(t) &= (1-t)^m Q_0 \;+\; mt(1-t)^{m-1}Q_1 \;+\; \dots \\
&= \begin{bmatrix} X(t) & Y(t) & Z(t) & W(t) \end{bmatrix}^T
\end{aligned}
$$

with $Q_i = \begin{bmatrix} X_i & Y_i & Z_i & W_i \end{bmatrix}^T$. Let $q_i = \begin{bmatrix} X_i/W_i & Y_i/W_i & Z_i/W_i \end{bmatrix}^T$ be the corresponding cartesian points, and the corresponding cartesian curve is

$$
q(t) = \frac{Q(t)}{W(t)}
$$

Differentiation with respect to the parameter $t$ yields

$$
\dot{q}(t) = \frac{\dot{Q}W - Q\dot{W}}{W^2}
$$

At the start of the curve, when $t = 0$,

$$
\dot{Q}(0) = m(Q_1 - Q_0)
$$

and so

$$\dot{q}(0) \;=\; \frac{m(Q_1 W_0 - Q_0 W_1)}{W_0^2}$$

so that

$$\left(\frac{W_0}{mW_1}\right)\dot{q}(0) \;=\; \frac{Q_1}{W_1} \;-\; \frac{Q_0}{W_0}$$

and hence

$$\dot{q}(0) \;=\; \left(\frac{w(Q_1)}{w(Q_0)}\right) m(q_1 - q_0)\ . \tag{8}$$

This means that the initial tangent is indeed along the line joining (the cartesian version of) the first two control points. However its magnitude is changed by a factor that depends upon the ratio $w(Q_1)/w(Q_0)$.

Similarly, at the other end of the curve, when $t = 1$,

$$\dot{q}(1) \;=\; \left(\frac{w(Q_{m-1})}{w(Q_m)}\right) m(q_m - q_{m-1})\ . \tag{9}$$

## 5. End Motion Conditions

This section considers how to deal with constraints at the end of a Bézier motion which specify the pose and speed (of the moving body). For convenience, it is assumed that these are given for the start of the motion when $t = 0$; the method for the end of the motion when $t = 1$ is the same with obvious changes in signs. These are the forms of constraint used with the examples in [33].

The “speed” constraint really refers to the derivative with respect to the parameter $t$ (rather than time). It is given in terms of the angular “velocity” vector (of the body) and the linear “velocity” of a reference point $P$ within the body. It is convenient to work with ordinary vectors in three-dimensional space and these are denoted by bold, lower case letters.

For the start of the motion, the following information is given:

- the pose $S_0$ of the body;
- the position $\mathbf{q}_0$ of the reference point $P$ and this can be derived from $S_0$;
- the linear velocity $\mathbf{v}_0$ of the reference point $P$;
- the angular velocity vector $\boldsymbol{\Omega}_0$ of the body.

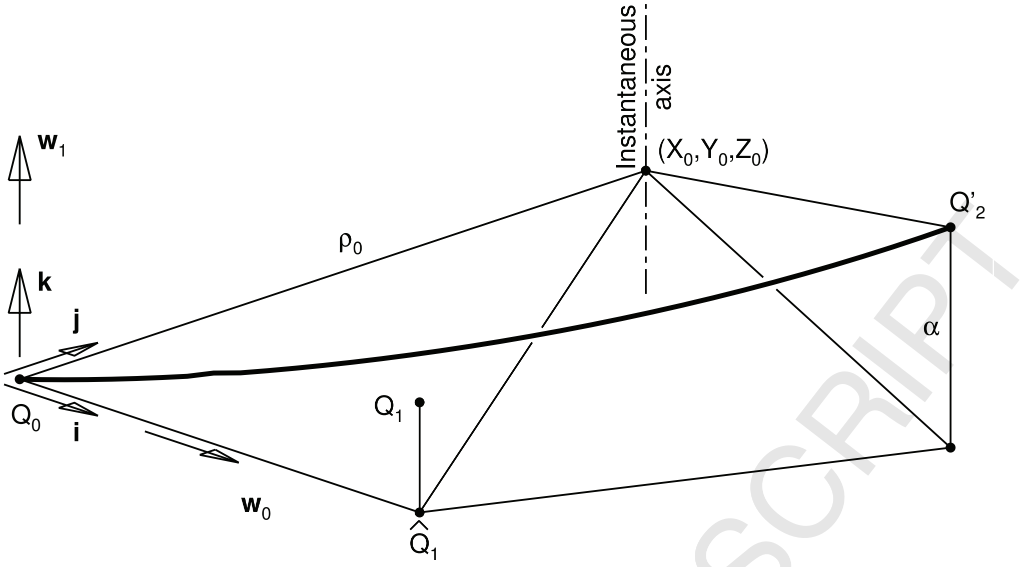


Figure 3: Start of the motion

This information is used to select an appropriate control pose $S_1$. Thus the pose and velocity conditions required at the start of the motion determine the first two control poses $S_0$ and $S_1$. A similar remark applies to the end of the motion.

Note that the curve followed by the point $P$ is given by the following (using lemma 3.1 and corollary 3.2)

$$\begin{aligned}
Q(t) &= (1-t)^{2n}\overline{S_0}PS_0 + 2n(1-t)^{2n-1}t[\tfrac{1}{2}(\overline{S_0}PS_1 + \overline{S_1}PS_0)] + O(t^2) \\
&= \overline{S_0}PS_0 + (2n)[\tfrac{1}{2}(\overline{S_0}PS_1 + \overline{S_1}PS_0) - \overline{S_0}PS_0]t + O(t^2) \\
&= Q_0 + (2n)[Q_1 - Q_0]t + O(t^2)
\end{aligned}$$

where

$$\begin{aligned}
Q_0 &= \overline{S_0}PS_0 \\
Q_1 &= \tfrac{1}{2}(\overline{S_0}PS_1 + \overline{S_1}PS_0)\ .
\end{aligned}$$

The construction for $S_1$ is now described. Fig. 3 shows the basic geometry at the start of the required motion. Here $Q'_2 = \overline{S_1}PS_1$ and the curve shown is obtained from the linear Bézier motion defined by $Q_0$ and $Q'_2$. This approximates to the curve from the required motion for small values of $t$.

Firstly define a set of orthogonal unit vectors, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$, at $\mathbf{q}_0$, such that $\mathbf{k}$ is in the direction of $\mathbf{\Omega}_0$, and $\mathbf{i}$ is in the direction of $\mathbf{w}_0 = \mathbf{v}_0 - \mathbf{w}_1$ where $\mathbf{w}_1 = (\mathbf{v_0}\cdot\mathbf{k})\mathbf{k}$ is the component of $\mathbf{v}_0$ in the direction of $\mathbf{\Omega}_0$.

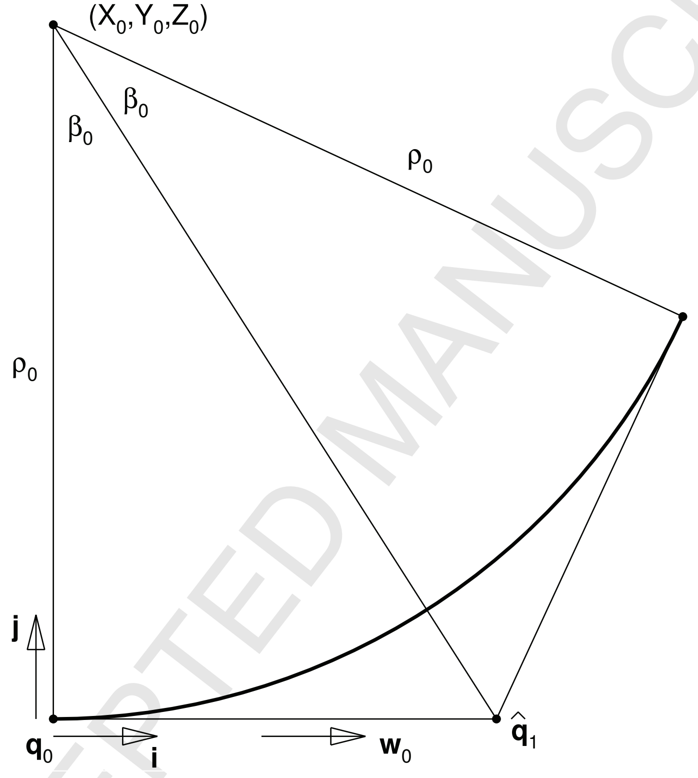


Figure 4: Start of the motion viewed along the instantaneous axis of rotation

The initial motion of $P$ is about an instantaneous axis in the direction of $\mathbf{k}$. Consider the motion restricted to the plane defined by $\mathbf{i}$ and $\mathbf{j}$. The velocity of $P$ is $\mathbf{w}_0$ and is in the direction of $\mathbf{i}$ as in Fig. 3. Fig. 4 shows the same geometry looking in the direction of the instantaneous axis. This axis cuts the plane at a point at distance $\rho_0 = |\mathbf{w}_0|/|\mathbf{\Omega}_0|$ from $\mathbf{q}_0$ in the direction of $\mathbf{j}$. The coordinates, in world space, of this point are given by

$$(X_0, Y_0, Z_0) \;=\; \mathbf{q}_0 + \rho_0 \mathbf{j} \;.$$

It is assumed that $S_0$ is chosen so that $\overline{S_0}S_0 = 1$. This means that $w(Q_0) = 1$, by lemma 2.4.

Referring to Fig. 3 and Fig. 4, the control pose $S_1$ can be obtained by performing the transform $S_0$ and then using a rotation $U$ through angle $2\beta_0$ about the instantaneous axis, and a translation $T$ along the axis through distance $\alpha$; thus $S_1 = S_0UT$. If $\mathbf{k} = (k_1, k_2, k_3)$, then a unit bivector corresponding to the axis is $b = k_1e_{23} + k_2e_{31} + k_3e_{12}$. Hence, $U$ can be formed as $U = \overline{T_0}R_0T_0$ where $T_0$ translates the origin to the instantaneous centre, and $R_0$ performs a rotation about the origin through $2\beta_0$. This proves the following result.

**Theorem 5.1.** *The required second control pose $S_1$ can be formed as*

$$S_1 \;=\; S_0\overline{T_0}R_0T_0T \tag{10}$$

*where*

$$\begin{aligned} T_0 &= 1 + \tfrac{1}{2}\varepsilon e_0(X_0e_1 + Y_0e_2 + Z_0e_3) && (11)\\ R_0 &= (\cos\beta_0) + (\sin\beta_0)b && (12)\\ T &= 1 + \tfrac{1}{2}\varepsilon\alpha e_0(k_1e_1 + k_3e_2 + k_3e_3) \;. && (13) \end{aligned}$$

□

It is seen that $\overline{R_0}R_0$, $\overline{T_0}T_0$ and $\overline{T}T$ are all unity. Assume that $w(P) = 1$. Then lemma 2.4 shows that $w(Q_1) = \cos\beta_0$.

Equation (8) shows that the condition on the derivative at the start of the motion is

$$\mathbf{v}_0 \;=\; \dot{\mathbf{b}}(0) = 2n\left(\frac{W(Q_1)}{W(Q_0)}\right)(\mathbf{q}_1 - \mathbf{q}_0) = 2n\cos\beta_0(\mathbf{q}_1 - \mathbf{q}_0) \tag{14}$$

where $\mathbf{q}_1$ is the cartesian point corresponding to $Q_1$. If $\hat{\mathbf{q}}_1$ is the projection of this point onto the plane through $\mathbf{q}_0$, then the above relation becomes

$$\mathbf{w}_0 \;=\; 2n\cos\beta_0(\hat{\mathbf{q}}_1 - \mathbf{q}_0) \;.$$

From Fig. 4 it is seen that

$$||\hat{\mathbf{q}}_1 - \mathbf{q}_0|| \;=\; \rho_0\tan\beta_0$$

and hence

$$\sin\beta_0 \;=\; \frac{||\boldsymbol{\Omega}_0||}{2n}\ . \tag{15}$$

Taking the component of equation (14) perpendicular to the plane shows that the distance $\alpha$ of translation is given by

$$||\mathbf{w}_1|| \;=\; \mathbf{v}_0\cdot\mathbf{k} \;=\; (2n\cos\beta_0)\alpha\ .$$

Hence $S_1$ can now be obtained from equation (10).

## 6. Examples

Three examples are given. These are based on examples using matrix-based techniques taken from the literature. The first example [34] considers a motion passing between two specified end poses with the associated speeds also being given. A rectangular block is moved and its centre is used as the reference point. The end poses are specified in terms of: the position vector $\mathbf{r}$ of the reference point, a unit vector $\mathbf{a}$ giving the direction of the axis of rotation, and the angle $\gamma$ (in radians) of the rotation about that axis. The speeds are specified by the velocity $\mathbf{v}$ of the reference point and the angular velocity vector $\boldsymbol{\Omega}$ of the body. Subscripts 0 and 1 are used for the start and finish of the motion. The end constraints are the following

$$\begin{array}{rcl@{\qquad}rcl}
\mathbf{r}_0 &=& \mathbf{0} & \mathbf{r}_1 &=& 8\mathbf{i}+10\mathbf{j}+12\mathbf{k}\\
\mathbf{a}_0 &=& \mathbf{k} & \mathbf{a}_1 &=& 0.267\mathbf{i}+0.535\mathbf{j}+0.802\mathbf{k}\\
\gamma_0 &=& 0 & \gamma_1 &=& 1.959\\
\mathbf{v}_0 &=& \mathbf{i}+\mathbf{j}+\mathbf{k} & \mathbf{v}_1 &=& \mathbf{i}+5\mathbf{j}+3\mathbf{k}\\
\boldsymbol{\Omega}_0 &=& \mathbf{i}+2\mathbf{j}+3\mathbf{k} & \boldsymbol{\Omega}_1 &=& 2\mathbf{i}+\mathbf{j}+\mathbf{k}\ .
\end{array}$$

Applying the proposed method at each end of the motion allows the following four control poses to be found

$$\begin{array}{rcl}
S_0 &=& 1.000\\
S_1 &=& 0.782+0.167\varepsilon e_{01}+0.167\varepsilon e_{02}+0.167\varepsilon e_{03}\\
&& \quad +0.500e_{12}-0.333e_{13}+0.167e_{23}+0.213\varepsilon\omega\\
S_2 &=& 0.768+1.614\varepsilon e_{01}+1.502\varepsilon e_{02}+6.326\varepsilon e_{03}\\
&& \quad +0.404e_{12}-0.497e_{13}-0.020e_{23}+4.258\varepsilon\omega\\
S_3 &=& 0.557+2.895\varepsilon e_{01}+1.456\varepsilon e_{02}+4.010\varepsilon e_{03}\\
&& \quad +0.666e_{12}-0.444e_{13}+0.222e_{23}+7.101\varepsilon\omega
\end{array}$$

The resultant motion is shown in Fig. 5 with the control poses shown with thicker lines. The body being moved is a cube, but the scale factors along the main axes

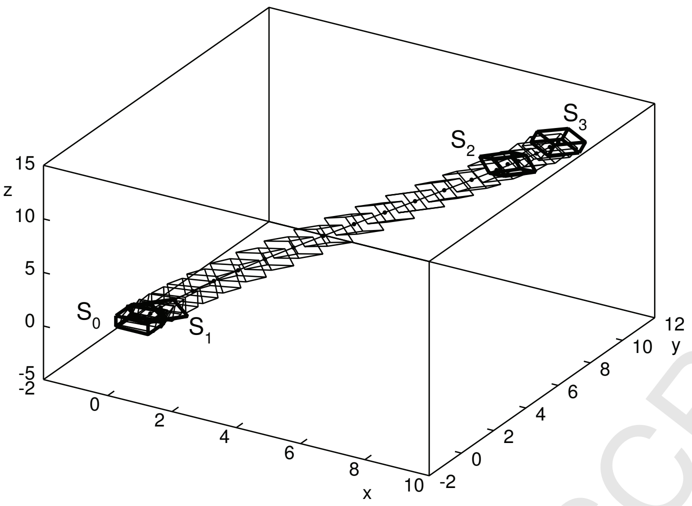


Figure 5: First example: 3D motion based on [34]

used in [34] have also been applied to ease comparison. The result is visually very similar in form to those obtained in [34].

In order to check that the imposed constraints are indeed satisfied, the paths traced by the local origin of the moving body and points unit distance along each of its main axes were considered. Using finite differences (with a step length of 0.001) the velocities of these four points were estimated and the resultant values used to derive the other properties of the motion at its start and finish. The results obtained are shown in table 1 and it is seen that the constraints have been matched exactly. The apparent error in $\mathbf{a}_1$ and $\gamma_1$ is because the constraints are given only to three decimal places meaning that the original $\mathbf{a}_1$ is not a unit vector to the accuracy used for the derived values.

| | |
|---|---|
| $\mathbf{r}_0$ | $0.00000\mathbf{i} + 0.00000\mathbf{j} + 0.00000\mathbf{k}$ |
| $\mathbf{a}_0$ | $0.00000\mathbf{i} + 0.00000\mathbf{j} + 1.00000\mathbf{k}$ |
| $\gamma_0$ | $0.00000$ rad $=$ $0.00000$ deg |
| $\mathbf{v}_0$ | $1.00000\mathbf{i} + 1.00000\mathbf{j} + 1.00000\mathbf{k}$ |
| $\mathbf{\Omega}_0$ | $1.00000\mathbf{i} + 2.00000\mathbf{j} + 3.00000\mathbf{k}$ |
| $\mathbf{r}_1$ | $8.00000\mathbf{i} + 10.00000\mathbf{j} + 12.00000\mathbf{k}$ |
| $\mathbf{a}_1$ | $0.26726\mathbf{i} + 0.53452\mathbf{j} + 0.80178\mathbf{k}$ |
| $\gamma_1$ | $1.95913$ rad $=$ $112.24972$ deg |
| $\mathbf{v}_1$ | $1.00000\mathbf{i} + 5.00000\mathbf{j} + 3.00000\mathbf{k}$ |
| $\mathbf{\Omega}_1$ | $2.00000\mathbf{i} + 1.00000\mathbf{j} + 1.00000\mathbf{k}$ |

Table 1: Properties derived from the generated motion in the first example

The second example is a planar motion derived in [33]. As well as the constraints imposed at the ends of the motion, an additional constraint specifies the pose in the middle, when the parameter $t$ is $\frac{1}{2}$. Using the same notation as before and with the subscript $m$ for the mid-pose, the constraints are the following.

| | |
|---|---|
| $\mathbf{r}_0$ | $2.00000\mathbf{i} + 8.00000\mathbf{j} + 0.00000\mathbf{k}$ |
| $\mathbf{a}_0$ | $0.00000\mathbf{i} + 0.00000\mathbf{j} + 1.00000\mathbf{k}$ |
| $\gamma_0$ | $0.00000$ rad $=$ $0.00000$ deg |
| $\mathbf{v}_0$ | $3.00000\mathbf{i} + 10.00000\mathbf{j} + 0.00000\mathbf{k}$ |
| $\boldsymbol{\Omega}_0$ | $-0.00000\mathbf{i} + 0.00000\mathbf{j} - 1.00000\mathbf{k}$ |
| $\mathbf{r}_1$ | $23.00000\mathbf{i} + 10.00000\mathbf{j} + 0.00000\mathbf{k}$ |
| $\mathbf{a}_1$ | $0.00000\mathbf{i} + 0.00000\mathbf{j} + 1.00000\mathbf{k}$ |
| $\gamma_1$ | $1.04720$ rad $=$ $60.00000$ deg |
| $\mathbf{v}_1$ | $2.00000\mathbf{i} + 5.00000\mathbf{j} + 0.00000\mathbf{k}$ |
| $\boldsymbol{\Omega}_1$ | $0.00000\mathbf{i} + 0.00000\mathbf{j} + 2.00000\mathbf{k}$ |

Table 2: Properties derived from the generated motion in the second example

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
\mathbf{r}_0 & = & 2\mathbf{i} + 8\mathbf{j} & \mathbf{r}_m & = & 12\mathbf{i} + 4\mathbf{j} & \mathbf{r}_1 & = & 23\mathbf{i} + 10\mathbf{j} \\
\mathbf{a}_0 & = & \mathbf{k} & \mathbf{a}_m & = & \mathbf{k} & \mathbf{a}_1 & = & \mathbf{k} \\
\gamma_0 & = & 0 & \gamma_m & = & -1 & \gamma_1 & = & \pi/3 \\
\mathbf{v}_0 & = & 3\mathbf{i} + 10\mathbf{j} & & & & \mathbf{v}_1 & = & 2\mathbf{i} + 5\mathbf{j} \\
\boldsymbol{\Omega}_0 & = & -\mathbf{k} & & & & \boldsymbol{\Omega}_1 & = & 2\mathbf{k}
\end{array}$$

The five constraints allow a Bézier quartic motion to be used. The first and last pair of control poses are derived as before. The middle control pose $S_2$ is found by rearranging equation (5) when $t = \frac{1}{2}$ so that

$$S_2 \;=\; \tfrac{1}{6}[16P_m - (S_0 + 4S_1 + 4S_3 + S_4)]$$

where $P_m$ is the prescribed pose at the middle of the motion.

The control poses are found to be the following

$$\begin{array}{rcl}
S_0 & = & 1 + \varepsilon e_{01} + 4\varepsilon e_{02} \\
S_1 & = & 0.992 + 0.867\varepsilon e_{01} + 5.344\varepsilon e_{02} - 0.125e_{12} \\
S_2 & = & 0.725 + 0.737\varepsilon e_{01} + 7.476\varepsilon e_{02} - 1.457e_{12} \\
S_3 & = & 0.964 + 11.890\varepsilon e_{01} + 1.328\varepsilon e_{02} + 0.268e_{12} \\
S_4 & = & 0.866 + 12.459\varepsilon e_{01} - 1.420\varepsilon e_{02} + 0.500e_{12}
\end{array}$$

and the motion is shown in Fig. 6. Several motions satisfying the constraints are given in [33], some of which are significantly different to others. The motion generated here is similar in form to some of those previously generated in [33] and seems to be a natural motion between the given constraints.

As with the previous example, the generated motion is sampled and its properties, given in table 2, derived. Again, it is seen that the imposed constraints have been satisfied.

The final example is based on a three-dimensional motion given in [33]. Again there are five constraints: three on pose and two on speed. Unfortunately it appears that

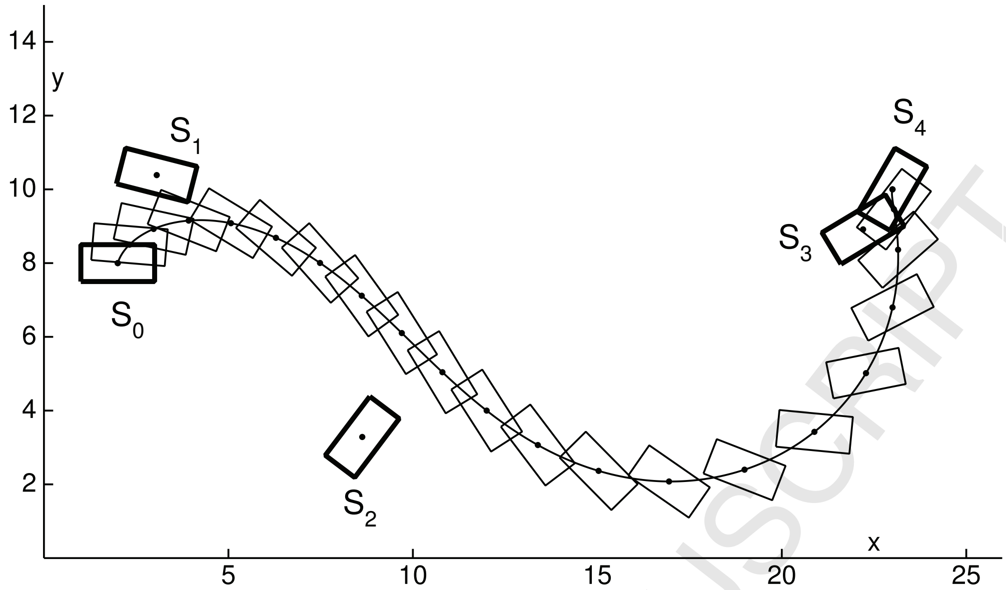


Figure 6: Second example: 2D motion based on [33]

there are a number of typographical errors in the previously published details of the constraints. The following values are used here as they generate a motion similar to those shown in [33].

$$\begin{aligned}
\mathbf{r}_0 &= \mathbf{i}+\mathbf{j}+\mathbf{k}\\
\mathbf{a}_0 &= 0.436\mathbf{i}-0.218\mathbf{j}+0.873\mathbf{k}\\
\gamma_0 &= 26.3\,\mathrm{deg}\\
\mathbf{v}_0 &= 12\mathbf{i}-12\mathbf{j}-16\mathbf{k}\\
\mathbf{\Omega}_0 &= \mathbf{i}+0.5\mathbf{j}-0.6\mathbf{k}
\end{aligned}$$

$$\begin{aligned}
\mathbf{r}_m &= 10\mathbf{i}-20\mathbf{k}\\
\mathbf{a}_m &= 0.914\mathbf{i}-0.304\mathbf{j}+0.269\mathbf{k}\\
\gamma_m &= -32\,\mathrm{deg}
\end{aligned}$$

$$\begin{aligned}
\mathbf{r}_1 &= 32.4\mathbf{i}+19\mathbf{j}-19\mathbf{k}\\
\mathbf{a}_1 &= 0.966\mathbf{i}-0.221\mathbf{j}+0.132\mathbf{k}\\
\gamma_1 &= 103.8\,\mathrm{deg}\\
\mathbf{v}_1 &= 12\mathbf{i}+16\mathbf{j}+4\mathbf{k}\\
\mathbf{\Omega}_1 &= -0.4\mathbf{i}-0.2\mathbf{j}+0.4\mathbf{k}
\end{aligned}$$

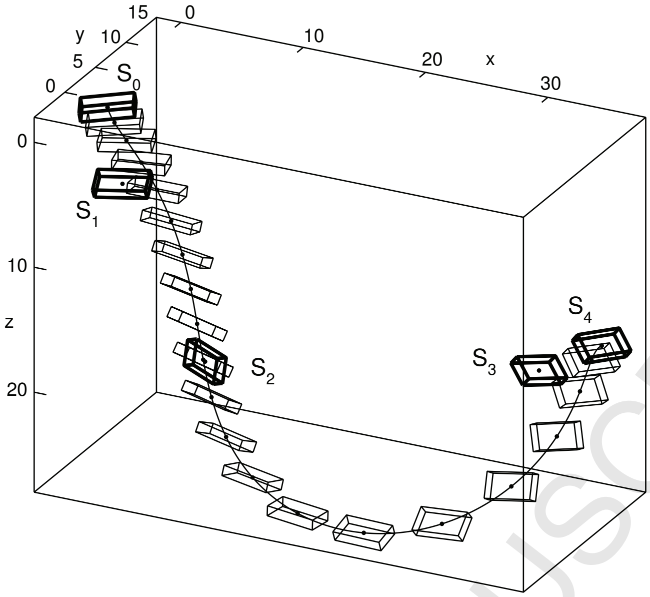


Figure 7: Third example: 3D motion based on [33]

Again a Bézier quartic motion is obtained whose control poses are the following

$$\begin{aligned}
S_0 &= 0.974 + 0.611\varepsilon e_{01} + 0.437\varepsilon e_{02} + 0.412\varepsilon e_{03}\\
&\quad +0.199e_{12} + 0.050e_{13} + 0.099e_{23} + 0.124\varepsilon\omega\\
S_1 &= 0.967 + 1.588\varepsilon e_{01} - 1.402\varepsilon e_{02} - 1.563\varepsilon e_{03}\\
&\quad +0.111e_{12} + 0.020e_{13} + 0.228e_{23} + 0.226\varepsilon\omega\\
S_2 &= 1.262 + 7.750\varepsilon e_{01} + 11.789\varepsilon e_{02} - 11.032\varepsilon e_{03}\\
&\quad -0.352e_{12} - 0.408e_{13} - 1.489e_{23} - 9.458\varepsilon\omega\\
S_3 &= 0.587 + 6.878\varepsilon e_{01} - 4.069\varepsilon e_{02} - 14.790\varepsilon e_{03}\\
&\quad +0.045e_{12} + 0.201e_{13} + 0.783e_{23} + 9.438\varepsilon\omega\\
S_4 &= 0.617 + 9.330\varepsilon e_{01} - 3.045\varepsilon e_{02} - 15.904\varepsilon e_{03}\\
&\quad +0.104e_{12} + 0.174e_{13} + 0.760e_{23} + 9.679\varepsilon\omega
\end{aligned}$$

The motion generated is shown in Fig. 7 and (because of the choices made for the constraints) this has a similar form to the motions shown in [33].

Table 3 shows properties derived from the generated motion and, once again, these agree with the imposed constraints.

## 7. Conclusions

Free-form motions can be regarded as being generated by smoothly varying rigid-body transforms. They clearly have analogies with free-form curves although they

| | |
|---|---|
| $\mathbf{r}_0$ | $1.00000\mathbf{i} + 1.00000\mathbf{j} + 1.00000\mathbf{k}$ |
| $\mathbf{a}_0$ | $0.43644\mathbf{i} - 0.21822\mathbf{j} + 0.87287\mathbf{k}$ |
| $\gamma_0$ | $0.45902$ rad $=$ $26.30000$ deg |
| $\mathbf{v}_0$ | $12.00000\mathbf{i} - 12.00000\mathbf{j} - 16.00000\mathbf{k}$ |
| $\mathbf{\Omega}_0$ | $1.00000\mathbf{i} + 0.50000\mathbf{j} - 0.60000\mathbf{k}$ |
| $\mathbf{r}_1$ | $32.40000\mathbf{i} + 19.00000\mathbf{j} - 19.00000\mathbf{k}$ |
| $\mathbf{a}_1$ | $0.96628\mathbf{i} - 0.22106\mathbf{j} + 0.13204\mathbf{k}$ |
| $\gamma_1$ | $1.81165$ rad $=$ $103.80000$ deg |
| $\mathbf{v}_1$ | $12.00000\mathbf{i} + 16.00000\mathbf{j} + 4.00000\mathbf{k}$ |
| $\mathbf{\Omega}_1$ | $-0.40000\mathbf{i} - 0.20000\mathbf{j} + 0.40000\mathbf{k}$ |

Table 3: Properties derived from the generated motion in the third example

are more difficult to construct and manipulate. Rigid-body transforms can be generated using geometric algebra (of which there are several forms). A common means of combining transformations is multiplicatively using the slerp construction. While this can be used to generate free-form motions using the appropriate extension of the de Casteljau algorithm, the lack of commutativity in the multiplication makes geometric properties of the motion intractable.

It has been seen that geometric algebra also allows transforms to be combined additively and this leads to natural extensions of the Bézier and B-spline constructions to generate free-form motions. Some properties which are well known for curves pass over to motions, but the fact that the transform appears twice when it is applied to transform any point complicates matters.

This paper has been concerned with obtaining a geometric construction for the pair of end control poses for a Bézier motion when constraints on the end pose and linear and angular velocities are given. Such a construction has been found and demonstrated on some examples. The approach draws on the idea that locally a motion is approximately on the surface of a circular cylinder. The significance here is that the construction is a natural extension of the corresponding idea for the end-tangents of a Bézier (or B-spline) curve. As such, it is a geometric property which has immediate meaning for the motion itself. Such properties are far less apparent when motions are constructed by alternative means such as metrics in a space of matrices. This suggests that geometric algebra is a more natural environment for investigating free-form motions.

Although a particular form of geometric algebra has been used here, similar results are possible in other representations provided that suitable adjustments are made. Certainly the approach goes across directly to use with the homogeneous model. Care is required with using the CGA if the idea that points are represented by null vectors is to be preserved. Care is also be required if the dual quaternions are used since these make less distinction between geometry (points) and transforms.

*Acknowledgement*

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

[1] Farin, G., 2002. Curves and Surfaces for CAGD: A Practical Guide, 5th edition. Academic Press, London.

[2] Fu, Z., 2013. Solution of inverse kinematics for 6R robot manipulators with offset wrist based on geometric algebra. Journal of Mechanisms and Robotics. 5(3), 031010:1–7.

[3] Gouasmi, M., Ouali, M., Brahim, F., 2012. Robot kinematics using dual quaternions. International Journal of Robotics and Automation. 1(1), 13–30.

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

[5] Li, Q., Chai, X., Xiang, J., 2016. Mobiltity analysis of limited-degrees-of-freedom parallel mechanisms in the framework of geometric algebra. Journal of Mechanisms and Robotics. **8**(4), 041005:1–9.

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

[7] Leclercq, G., Lefèvre, P., Blohm, G., 2013. 3D kinematics using dual quaternions: theory and applications in neuroscience. Frontiers in Behavioral Neuroscience. 7, 7:1–25.

[8] Qiao, B., Tang, S., Ma, K., Liu, Z., 2013. Relative position and attitude estimation of spacecrafts based on dual quaternion for rendezvous and docking. Acta Astronautica. 91, 237–244.

[9] Horn, B. K. P., 1987. Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A. 4(4) 629–642.

[10] Jüttler, B., Wagner, M. G., 1996. Computer-aided design with spatial rational B-spline motions. Transactions of the ASME: Journal of Mechanical Design. 118(2), 193–201.

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

[12] Goldman, R., 2011. Understanding quaternions. Graphical Models. 73(1) 21–49.

[13] Purwar, A., Ge, Q. J., 2010. Kinematic convexity of rigid body displacements. In: *Proceedings of the ASME IDETC/CIE Conferences 2010*, Montreal, 1761–1772.

[14] Wu, W., You, Z., 2010. Modelling rigid origami with quaternions and dual quaternions. Proceedings of the Royal Society, Series A. 466, 2155–2174.

[15] González Calvet, R., 2007. Treatise of Plane Geometry through Geometric Algebra. TIMSAC, Cerdanyola del Vallés.

[16] Dorst, L., 2010. Tutorial: structure-preserving representation of Euclidean motions through conformal geometric algebra. In: Bayro-Corrochano, E., Scheuermann, G. (eds). Geometric Algebra Computing. Springer-Verlag, London, pp. 35–52.

[17] Shen, C. W., Hang, L. B. , Yang, T. L., 2017. Position and orientation characteristics of robot mechanisms based on geometric algebra. Mechanism and Machine Theory. 108, 231–243.

[18] Huo, X. M., Sun, T., Song, Y. M., 2017. A geometric algebra approach to determine motion/constraint, mobility and singularity of parallel mechanism. Mechanism and Machine Theory. 116, 273–293.

[19] Muralidhar, K., 2014. Complex vector formalism of harmonic oscillator in geometric algebra: particle mass, spin and dynamics in complex vector space. Foundations of Physics. 44(3), 266–295.

[20] Jaklič, G., Jüttler, B., Krajnc, M., Vitrih, V., Žagar, E., 2013. Hermite interpolation by rational $G^k$ motions of low degree. Journal of Computational and Applied Mathematics. 240, 20–30.

[21] Počkaj, K., 2014. Hermite $G^1$ rational spline motion of degree six. Numerical Algorithms. 66(4) 721–739.

[22] Tan, J., Xing, Y., Fan, W., Hing, P., 2018. Smooth orientation interpolation using parametric quintic-polynomial-based quaternion spline curve. Journal of Computational and Applied Mathematics. 329 256–267.

[23] Wareham, R., Lasenby, J., 2008. Mesh vertex pose and position interpolation using geometric algebra. In: Perales, F. J., Fisher, R. B. (eds). Articulated Motion and Deformable Objects. LNCS 5098. Springer-Verlag, Berlin, pp. 122–131.

[24] Cripps, R. J., Mullineux, G., 2016. Using geometric algebra to represent and interpolate tool poses. International Journal of Computer Integrated Manufacturing. 29(4), 406–423.

[25] Purwar, A., Chi, X., Ge, Q. J., 2007. Automatic fairing of two-parameter rational B-spline motion. Transactions of the ASME: Journal of Mechanical Design. 130(1) 011003:1–7.

[26] Popiel, T., Noakes, L., Bézier curves and $C^2$ interpolation in Riemannian manifolds. Journal of Approximation Theory. 148(2) 111–127.

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

[28] Perwass, C., 2009. Geometric Algebra with Applications in Engineering. Springer. Berlin.

[29] Cibura, C., Dorst, L., 2011. Determining conformal transformations in $\mathbb{R}^n$ from minimal correspondence data. Mathematical Methods in the Applied Sciences. 34(16), 2031–2046.

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

[31] Mullineux, G., Simpson, L. C., 2011. Rigid-body transforms using symbolic infinitesimals. In: Dorst, L., Lasenby, J. (eds). Guide to Geometric Algebra in Practice. Springer, London, pp. 353–369.

[32] Hunt, M., Mullineux, G., Cripps, R. J., Cross, B., 2016. Smooth motions through precision poses. In: Horváth, I., Pernot, J.-P., Rusák, Z. (eds). Proceedings of Tools and Methods for Competitive Engineering (TMCE) 2016. Delft University of Technology, Delft, pp. 551–562.

[33] Žefran, M., Kumar, V., 1998. Interpolation schemes for rigid body motions. Computer-Aided Design. 30(3), 179–189.

[34] Belta, C., Kumar, V., 2002. On the computation of rigid body motion. Electronic Journal of Computational Kinematics. 1, 1–12.

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

[36] Gunn, C., 2011. On the homogeneous model of Euclidean geometry. In: Dorst, L., Lasenby, J. (eds). Guide to Geometric Algebra in Practice. Springer, London, pp. 297–327.

[37] Gunn, C., 2017. Geometric algebras for Euclidean geometry. Advances in Applied Clifford Algebras. 27(1), 185–208.

## Appendix

The purpose of this appendix is to explain the choice of using the geometric algebra $\mathcal{G}_4$ in this paper. The choice is based largely upon the simplicity of notation and this is illustrated partly in connection with lemma 2.4. This is not to say that alternative approaches are not possible. It is simply that they do not seem as "natural" for the particular application considered.

The quaternions can handle rotations but the dual quaternions are needed to handle translations as well [7, 13, 14]. Use $\gamma$ to denote the dual number (with $\gamma^2 = 0$). The typical Euclidean point $(x, y, z)$ is represented by $P = 1 + p\gamma$ where $p = xi + yj + zk$. (A more natural representation for the point might have been expected to be simply $1 + p$, without the need for the appearance of $\gamma$.) A translation in the direction $(u, v, w)$ is generated by $T = 1 + \frac{1}{2}t\gamma$ where $t = ui + vj + wk$. The translation map is then

$$P \mapsto TPT = (1 + \tfrac{1}{2}t\gamma)(1 + p\gamma)(1 + \tfrac{1}{2}t\gamma) = 1 + (p + t)\gamma$$

The fact that $T$ is normalized is because $T^\dagger T = 1$, where $T^\dagger$ is the conjugate which changes the sign of $\gamma$. In $\mathcal{G}_4$, the equivalent map is $P \mapsto \overline{T}PT$ and the fact that $\overline{T}T = 1$ says that $T$ is normalized. The reverse operation is used in both the map and the normalization. With the dual quaternions, the conjugate is used with only one of them (and there is potential confusion as to which conjugate to use). Further, in $\mathcal{G}_4$ and other algebras, points are elements of grade 1, and transforms are elements of even grade: so they are clearly distinct ideas. This distinction is absent with the dual quaternions: indeed the elements $P$ and $T$ above have essentially the same form.

The conformal geometric algebra (CGA) [16, 29] is a popular form. It represents points in projective (and Euclidean) space by null vectors. In the direct equivalent of part (ii) of lemma 2.4, the element $Q$ is not necessarily a null vector, even if $P$ is. For example, using the notation of [29], if $U = (1 + e_{12})/\sqrt{2}$, $V = (1 + e_{13})/\sqrt{2}$, and $P = e_0 + e_1 + \frac{1}{2}e_\infty$, then $P^2 = 0$, so that $P$ is null. However, $Q = \frac{1}{2}(e_0 + e_1 + e_2 + e_3 + \frac{1}{2}e_\infty)$ with $Q^2 = \frac{1}{2}$, so that $Q$ is not null. The lemma can be amended to give a CGA version but it becomes more complicated.

Finally consider the homogeneous model [35, 36, 37]. It is straightforward to use this in place of $\mathcal{G}_4$ in this paper. Its slight drawback is that vectors in the algebra correspond to planes in Euclidean geometry, and trivectors to points (which seems the wrong way round): in particular, the Euclidean point $(x, y, z)$ corresponds to the trivector $e_{123} + xe_{023} - ye_{013} + ze_{012}$. A change of notation (for example, renaming $e_{123}$ as $E_0$, $e_{023}$ as $E_1$, and so on [36]) can be used to make the correspondence seem more natural but this seems an added complication.