Hermite Interpolation of Solid Orientations with Circular Blending Quaternion Curves*

Myung-Soo Kim and Kee-Won Nam
Department of Computer Science
POSTECH
Pohang 790-784, Korea

Abstract

Construction methods are presented that generate hermite interpolation quaternion curves on $SO(3)$. Two circular curves $C_1(t)$ and $C_2(t)$, $0 \leq t \leq 1$, are generated that interpolate two orientations $q_1$ and $q_2$, and have boundary angular velocities: $C'_1(0) = \omega_1$ and $C'_2(1) = \omega_2$, respectively. They are smoothly blended together on $SO(3)$ to generate a hermite quaternion curve $Q(t) \in SO(3)$, $0 \leq t \leq 1$, which satisfies the boundary conditions: $Q(0) = q_1, Q(1) = q_2, Q'(0) = \omega_1$, and $Q'(1) = \omega_2$.

Keywords: Quaternion, orientation, rotation, angular velocity, hermite interpolation, animation

1 Introduction

Given a 3D solid with a fixed reference point (e.g., its center of mass), a rigid motion of the solid can be uniquely specified by a path $(p(t), q(t)) \in \mathbb{R}^3 \times SO(3)$, $0 \leq t \leq 1$, where $\mathbb{R}^3$ is the 3D Euclidean space and $SO(3)$ is the rotation group of $\mathbb{R}^3$. That is, the first component $p(t) \in \mathbb{R}^3$ represents the translation of the reference point in $\mathbb{R}^3$, and the second component $q(t) \in SO(3)$ represents the rotation of the 3D solid around the fixed reference point. There are numerous well-known techniques to construct smooth curves in $\mathbb{R}^3$; however, it is not straightforward to extend them to $SO(3)$ while preserving all desirable

---

*A preliminary version of this paper was presented at *CG International '94*, Melbourne, Australia, June 27–July 1, 1994. This research was supported in part by KOSEF.
geometric properties of the original curve. For example, the de Casteljau type construction of spline curves in $\mathbb{R}^3$ can be extended for the construction of similar quaternion curves in $SO(3)$ [10, 11]. However, it does not preserve the $C^2$-continuity of a cubic B-spline curve in $SO(3)$ (see [4] for more details). To remedy this drawback, Kim et al. [5] suggested a general framework for transforming any spline curve in $\mathbb{R}^3$ (defined as a weighted sum of basis functions) into its unit quaternion analogue in $SO(3)$, while preserving all the differential properties of a spline curve. However, the curve construction symmetry is not preserved under this transformation. That is, the Bézier quaternion curve of [5] with control points $q_1, \ldots, q_n \in SO(3)$ has a different curve shape from the one with $q_n, \ldots, q_1$ as its control points.

The above limitations of general curve conversion schemes necessitate further investigation on the quaternion curve construction methods which are based on the intrinsic spherical structure of $SO(3)$. The local geometry of $SO(3)$ is identical to that of 3-sphere $S^3$, where $S^3 = \{ p \in \mathbb{R}^4 \mid \|p\| = 1 \}$. This is because the rotation group $SO(3)$ is a projective space of $S^3$ under the identification of two antipodal points $q, -q \in S^3$ mapping to the same 3D rotation $R_q \equiv R_{-q} \in SO(3)$ (see [2] and Appendix A). The quaternion curves are usually constructed in a more intuitive space $S^3$ instead of $SO(3)$. For the construction of a hermite quaternion curve in $S^3$, we also need to map an angular velocity $\omega \in \mathbb{R}^3$ into a tangent vector $v_q$ of $S^3$ (embedded in $\mathbb{R}^4$) as follows:

$$v_q = \frac{1}{2} \omega \cdot q \in T_q(S^3),$$

where $\cdot$ is the quaternion multiplication and $T_q(S^3)$ is the tangent space of $S^3$ also embedded in $T_q(\mathbb{R}^4) \equiv \mathbb{R}^4$. (See Appendix A for more details of the above equation.) Similarly, given a tangent vector $v_q \in T_q(S^3)$, the corresponding angular velocity $\omega \in \mathbb{R}^3$ is obtained as follows:

$$\omega = 2v_q \cdot q^{-1} \in \mathbb{R}^3.$$

Under this correspondence, we use both $\omega$ and $v_q$ interchangeably as angular velocities depending on the context.

There are some quaternion curve construction methods based on circular arcs in $S^3$ [7, 9, 14]. Wang and Joe [14] constructed a hermite interpolation curve on $S^3$ by using two circular arcs connected with $C^1$-continuity. At the junction of two circular arcs, however, large acceleration/torque is generated that gives undesirable effect on the smooth animation of a moving solid [1]. This is inevitable as long as circular arcs are used as basic components (see also the Nielson/Shieh circle spline of [9]). In this paper, we overcome such a drawback by blending two circular curves.

Kim and Nam [7] presented a circular blending method to construct a $C^k$-continuous quaternion path (with a blending function of degree $2k - 1$) which smoothly interpolates
a given sequence of unit quaternions in $S^3$. In this paper, we show that this method can be extended to generate a hermite interpolation curve on $S^3$. However, it is inefficient to transform the hermite interpolation problem on $S^3$ to an interpolation problem of four unit quaternions on $S^3$; i.e., to transform two boundary velocities into two additional unit quaternions to interpolate. The hermite interpolation problem on $S^3$ has some additional geometric features which make the algorithm more efficient. The tangent vector $v_q$ to $S^3$ at a unit quaternion $q \in S^3$ is always orthogonal to $q$ in $R^4$. This additional feature makes the transformation matrix simpler than that of Kim and Nam [7]. Hence, the overall algorithm becomes more efficient.

There are many hermite quaternion curves on $S^3$ which satisfy the same boundary conditions (see [3, 5, 6, 8, 11, 12]); however, they generate different curves on $S^3$. This is a major distinction from the hermite curve construction in $R^3$. Given $k + 1$ boundary conditions, there is a unique polynomial spline curve of degree $k$ in $R^3$. Different curve construction schemes generate the same curve of degree $k$ as long as they satisfy the same $k + 1$ boundary conditions. However, the hermite quaternion curves in $S^3$ have no polynomial degree. Even the simple circles in $S^3$ are non-polynomial curves. Therefore, it is worthwhile to develop another way of constructing a hermite quaternion curve in $S^3$ as long as the new method has relative advantages over the previous methods. The advantages of circular blending have been demonstrated in Kim and Nam [7].

The basic construction steps of this paper are: (i) a 3-sphere $S^3$ is cut with a hyper-plane $L^3$ to generate a 2-sphere $S^2$, (ii) the 2-sphere $S^2$ is cut with 2D planes to generate two circular curves, and (iii) the two circular curves are then blended together to generate a hermite interpolation curve in $S^3$. Let $l_i$ ($i = 1, 2$) be the tangent line of $S^3$ at $q_i$ to the direction of $v_i$, then the two tangent lines $l_1$ and $l_2$ determine a hyper-plane $L^3_{l_1, l_2}$ (which may not pass through the origin of $R^4$). By cutting the 3-sphere $S^3$ by the hyper-plane $L^3_{l_1, l_2}$, we get a 2-sphere $S^2_{l_1, l_2}$ ($= S^3 \cap L^3_{l_1, l_2}$); see Figure 1(a). The four elements $q_1, q_2, v_1, v_2$ are all contained in the hyper-plane $L^3_{l_1, l_2}$; the two points $q_1, q_2$ are on $S^2_{l_1, l_2}$, and the two vectors $v_1, v_2$ are tangent to $S^2_{l_1, l_2}$. Therefore, the problem is essentially reduced to a hermite interpolation problem on $S^2_{l_1, l_2}$. Let $L^2_{l_1, v_2}$ (respectively, $L^2_{l_1, q_1}$) be the 2D plane determined by the tangent line $l_1$ (respectively, $l_2$) and the point $q_2$ (respectively, $q_1$), then the plane intersects with the 2-sphere $S^2_{l_1, l_2}$ in a circle $C_1$ (respectively, $C_2$). The circular arcs $C_1$ and $C_2$ can be parameterized so that $C_1(0) = C_2(0) = q_1, C_1(1) = C_2(1) = q_2, C_1'(0) = v_1,$ and $C_2'(1) = v_2$. By smoothly blending the two circular curves $C_1$ and $C_2$, a hermite interpolation curve $Q(t), 0 \leq t \leq 1$, is constructed that satisfies all the boundary conditions; see Figure 1(b).

The rest of this paper is organized as follows. In §2, we present how to construct a hermite interpolation quaternion curve in $S^3$. In §3, an input method is discussed outlining how to specify an angular velocity at each solid orientation. Finally, we conclude this paper.
Figure 1: (a) Construction of a 2-sphere $S^2_{l_1,l_2} (= S^3 \cap L^3_{l_1,l_2})$, and (b) Construction of a Hermite Quaternion Curve by Blending Two Circular Curves.

in §4.

2 Hermite Interpolation in $S^3$

In this section, we consider the following hermite interpolation problem: given two points $q_1$ and $q_2 \in S^3$ and two vectors $v_1$ and $v_2$ tangent to $S^3$ at $q_1$ and $q_2$, respectively, how to construct a hermite interpolation curve $Q(t) \in S^3$, for $0 \leq t \leq 1$, which satisfies the boundary conditions:

$$Q(0) = q_1, \quad Q(1) = q_2, \quad Q'(0) = v_1, \quad \text{and} \quad Q'(1) = v_2.$$ 

In §2.1, a reduction method is presented that transforms the given problem into a similar problem in $S^2$. In §2.2, two circular curves are generated in $S^2$ that satisfy partial boundary conditions. There are two ways to generate a hermite interpolation curve in $S^3$: (i) the hermite curve is constructed in $S^2$ and transformed back to $S^3$, or (ii) the two circular curves (constructed in $S^2$) are transformed back to $S^3$, and they are blended together in $S^3$. The first and second methods are discussed in §2.3 and §2.4, respectively. (The pseudo codes are given in Appendices A and B, respectively.)
2.1 Problem Reduction to $S^2$

Let $l_i$ ($i = 1, 2$) be the tangent line of $S^3$ at the point $q_i$ in the direction of the tangent vector $v_i$, then the two tangent lines $l_1$ and $l_2$ determine a hyper-plane $L^3_{l_1,l_2}$ (which may not pass through the origin of $R^4$). Let $S^2_{l_1,l_2}(= S^3 \cap L^3_{l_1,l_2})$ be the intersection of $S^3$ with the hyper-plane $L^3_{l_1,l_2}$, then (i) the two points $q_1$ and $q_2$ are contained in the 2-sphere $S^2_{l_1,l_2}$, and (ii) the two vectors $v_1$ and $v_2$ are tangent to the 2-sphere $S^2_{l_1,l_2}$ at $q_1$ and $q_2$, respectively. This is because (i) the two points $q_1$ and $q_2$ are contained in both $S^3$ and $L^3_{l_1,l_2}$, and (ii) the two vectors $v_1$ and $v_2$ are tangent to $S^3$ and also contained in $L^3_{l_1,l_2}$. Therefore, the hermite interpolation problem in $S^3$ is essentially reduced to that in a lower dimensional 2-sphere $S^2_{l_1,l_2}$. To simplify further computations on the 2-sphere $S^2_{l_1,l_2}$ (i.e., to use 3D coordinates instead of 4D coordinates), we transform the 2-sphere $S^2_{l_1,l_2}$ into the standard unit 2-sphere $S^2(\subset R^3)$. The transformation is done in two steps (see also the pseudo code given in Appendix A):

1. Rotate $R^4$ so that the hyper-plane $L^3_{l_1,l_2}$ becomes a hyper-plane $w = w_0$ which is orthogonal to the $w$-axis and parallel to the $xyz$-space, where $w_0$ is the distance of $L^3_{l_1,l_2}$ from the origin of $R^4$. Through the rotation, the 2-sphere $S^2_{l_1,l_2}$ also transforms into a 2-sphere $\bar{S}^2_{l_1,l_2}$ which is the $w$-slice of $S^3$ at $w = w_0$. The 2-sphere $\bar{S}^2_{l_1,l_2}$ has its center at the point $\bar{p}_0 = (0, 0, 0, w_0)$ on the $w$-axis and its radius as $r_0 = (1 - w_0^2)^{1/2}$.

2. Translate the 2-sphere $\bar{S}^2_{l_1,l_2}$ by $-\bar{p}_0$ so that its center is located at the origin of $R^3$, and scale it by a factor of $1/r_0$. Then the resulting 2-sphere matches exactly with the standard unit 2-sphere $S^2$ in $R^3$.

Since the three vectors $v_1, v_2, q_2 - q_1$ are parallel to the hyper-plane $L^3_{l_1,l_2}$, the unit normal vector $n_4$ of $L^3_{l_1,l_2}$ can be computed by

$$n_4 = \frac{v_1 \times v_2 \times (q_2 - q_1)}{\|v_1 \times v_2 \times (q_2 - q_1)\|},$$

where the ternary operation $\cdot \times \cdot \times \cdot$ is defined by:

$$p_1 \times p_2 \times p_3 = \begin{vmatrix}
e_1 & e_2 & e_3 & e_4 \\
x_1 & y_1 & z_1 & w_1 \\
x_2 & y_2 & z_2 & w_2 \\
x_3 & y_3 & z_3 & w_3 \\
\end{vmatrix},$$

with $e_1 = (1, 0, 0, 0), e_2 = (0, 1, 0, 0), e_3 = (0, 0, 1, 0), e_4 = (0, 0, 0, 1)$, and $p_i = (x_i, y_i, z_i, w_i)$, for $i = 1, 2, 3$. Note that the ternary operation is a generalization of the cross product of
two vectors in $R^3$ to that of three vectors in $R^4$ [7, 13]. Moreover, let
\[ n_3 = \frac{n_4 \times v_1 \times v_2}{\|n_4 \times v_1 \times v_2\|}, \quad n_1 = \frac{v_1}{\|v_1\|}, \quad \text{and} \quad n_2 = n_3 \times n_4 \times n_1. \]

Then, the transformation
\[ T = [n_1 \quad n_2 \quad n_3 \quad n_4]^T \]
is an orthogonal transformation which maps the two points $q_1, q_2 \in S^2_{t_1, t_2}$ into two points $\bar{q}_1, \bar{q}_2 \in \bar{S}^2_{t_1, t_2}$, and the two tangent vectors $v_1, v_2$ of $S^2_{t_1, t_2}$ into two tangent vectors $\bar{v}_1, \bar{v}_2$ of $\bar{S}^2_{t_1, t_2}$, that is,
\[ \bar{q}_i = T \cdot q_i \quad \text{and} \quad \bar{v}_i = T \cdot v_i, \quad \text{for} \quad i = 1, 2. \]

Suppose the coordinates of $q_i$ and $\bar{q}_i$ are given by:
\[ q_i = (x_i, y_i, z_i, w_i) \quad \text{and} \quad \bar{q}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i, \bar{w}_i). \]
(Though all the position and velocity vectors are written as row vectors, the vectors are assumed to be column vectors when they are multiplied with the matrices $T, T^{-1}, S, S^{-1}$, etc.) Since the two points $\bar{q}_1$ and $\bar{q}_2$ are in the 2-sphere $\bar{S}^2_{t_1, t_2}$ which is contained in the hyper-plane $w = w_0$, the points have $w_0$ as their common $w$-coordinate. That is,
\[ \bar{w}_1 = n_4 \cdot q_1 \quad (= \bar{w}_2 = n_4 \cdot q_2) = w_0. \]
(This is because $n_4$ is the last row of the matrix $T$.) Similarly, the other coordinates of $\bar{q}_i$ can be obtained by
\[ (\bar{x}_i, \bar{y}_i, \bar{z}_i) = (n_1 \cdot q_i, n_2 \cdot q_i, n_3 \cdot q_i). \]
Since the unit vectors $n_3$ and $n_4$ are constructed so that they are orthogonal to $v_1$ and $v_2$, the last two coordinates of $\bar{v}_i = T \cdot v_i$ are 0's. Moreover, $n_2$ is also orthogonal to $v_1$ (= $\|v_1\| \cdot n_1$).
Thus, $\bar{v}_1$ and $\bar{v}_2$ have the following simple coordinate expressions:
\[ \bar{v}_1 = (\|v_1\|, 0, 0, 0), \quad \text{and} \quad \bar{v}_2 = (n_1 \cdot v_2, n_2 \cdot v_2, 0, 0). \]

Next, the 2-sphere $\bar{S}^2_{t_1, t_2}$ is translated by $-\bar{p}_0 = (0, 0, 0, -w_0)$ so that it can be embedded in $R^3$ with its center at the origin of $R^3$, and then the resulting sphere (with radius $r_0 = (1 - w_0^2)^{1/2}$) is scaled by
\[ S = \begin{bmatrix} 1/r_0 & 0 & 0 \\ 0 & 1/r_0 & 0 \\ 0 & 0 & 1/r_0 \end{bmatrix} \]
so that it can match exactly with the standard unit 2-sphere $S^2$. Let $\tilde{q}_i$ and $\tilde{v}_i$ ($i = 1, 2$) be the transformed points on $S^2$ and the transformed tangent vectors to $S^2$, respectively; then they have the following simple coordinate representations in $R^3$:

\[
\begin{align*}
\tilde{q}_1 &= (\tilde{x}_1, \tilde{y}_1, \tilde{z}_1) = (1/r_0) \cdot (0, n_2 \cdot q_1, n_3 \cdot q_1), \\
\tilde{q}_2 &= (\tilde{x}_2, \tilde{y}_2, \tilde{z}_2) = (1/r_0) \cdot (n_1 \cdot q_2, n_2 \cdot q_2, n_3 \cdot q_2), \\
\tilde{v}_1 &= (1/r_0) \cdot (\|v_1\|, 0, 0), \quad \text{and} \\
\tilde{v}_2 &= (1/r_0) \cdot (n_1 \cdot v_2, n_2 \cdot v_2, 0).
\end{align*}
\]

(Note that, since $v_1$ is orthogonal to $q_1$, $\tilde{x}_1 = n_1 \cdot q_1 = (v_1 \cdot q_1)/\|v_1\| = 0$ and $\tilde{x}_1 = (1/r_0) \cdot \tilde{x}_1 = 0$.) Thus, we have reduced the given hermite interpolation problem in $S^3$ into a similar problem in $S^2$. Once a hermite interpolation curve $\tilde{Q}(t) \in S^2$, $0 \leq t \leq 1$, is constructed that satisfies the four boundary conditions:

\[
\begin{align*}
\tilde{Q}(0) &= \tilde{q}_1, \quad \tilde{Q}(1) = \tilde{q}_2, \quad \tilde{Q}'(0) = \tilde{v}_1, \quad \text{and} \quad \tilde{Q}'(1) = \tilde{v}_2,
\end{align*}
\]

this curve can be transformed back to a hermite interpolation curve $Q(t) \in S^3$, $0 \leq t \leq 1$, which satisfies the original four boundary conditions:

\[
\begin{align*}
Q(0) &= q_1, \quad Q(1) = q_2, \quad Q'(0) = v_1, \quad \text{and} \quad Q'(1) = v_2.
\end{align*}
\]

The backward transformation is given by:

\[
Q(t) = T^{-1} \cdot (S^{-1} \cdot \tilde{Q}(t) + \tilde{p}_0) = T^{-1} \cdot (r_0 \cdot \tilde{Q}(t)) + p_0, \quad \text{for } 0 \leq t \leq 1,
\]

where

\[
T^{-1} = T^t = \begin{bmatrix} n_1 & n_2 & n_3 & n_4 \end{bmatrix}, \quad S^{-1} = \begin{bmatrix} r_0 & 0 & 0 \\
0 & r_0 & 0 \\
0 & 0 & r_0 \end{bmatrix}, \quad \text{and} \quad p_0 = T^{-1} \cdot \tilde{p}_0 = w_0 \cdot n_4.
\]

In the following subsection §2.2, we consider how to construct a hermite interpolation curve in $S^2$.

### 2.2 Hermite Interpolation in $S^2$

In this subsection, we consider the following problem: given two points $q_1$ and $q_2 \in S^2$ and two vectors $v_1$ and $v_2$ tangent to $S^2$ at $q_1$ and $q_2$, respectively, how to construct a hermite interpolation curve $Q(t) \in S^2$, $0 \leq t \leq 1$, which satisfies the boundary conditions:

\[
Q(0) = q_1, \quad Q(1) = q_2, \quad Q'(0) = v_1, \quad \text{and} \quad Q'(1) = v_2.
\]
We first construct one circular curve $C_1(t) \in S^2$, $0 \leq t \leq 1$, which satisfies the following three boundary conditions:

$$C_1(0) = q_1, \quad C_1(1) = q_2, \quad \text{and} \quad C'_1(0) = v_1.$$ 

In a similar way, the other circular curve $C_2(t) \in S^2$, $0 \leq t \leq 1$, can be constructed so that it satisfies:

$$C_2(0) = q_1, \quad C_2(1) = q_2, \quad \text{and} \quad C'_2(0) = v_2.$$ 

By smoothly blending the two circular curves $C_1$ and $C_2$ (see also [7]), we can construct a hermite curve on $S^2$ which satisfies all the four given boundary conditions.

Let $l_1$ be the tangent line of $S^2$ at $q_1$ to the direction of $v_1$, and $L_{l_1,q_2}^2$ be the 2D plane which is determined by $l_1$ and $q_2$. Moreover, let

$$n_2 = v_1/\|v_1\|, \quad n_3 = \frac{(q_2 - q_1) \times n_2}{\|(q_2 - q_1) \times n_2\|}, \quad \text{and} \quad n_1 = n_2 \times n_3,$$

then $n_3$ is the unit normal vector of $L_{l_1,q_2}^2$, and the transformation $T$ given by

$$T = [n_1 \ n_2 \ n_3]^t$$

rotates $R^3$ so that the plane $L_{l_1,q_2}^2$ transforms into a 2D plane $\bar{L}_{l_1,q_2}^2$ (which is parallel to the $xy$-plane and orthogonal to the $z$-axis). The plane $\bar{L}_{l_1,q_2}^2$ has an implicit plane equation $z = z_0$, where $z_0$ is the distance between $L_{l_1,q_2}^2$ and the origin of $R^3$. The orthogonal transformation $T$ preserves the lengths of vectors and the angle between any two vectors; thus, $T$ maps $v_1, q_1, q_2$ into $\bar{v}_1, \bar{q}_1, \bar{q}_2$ which have the following simple expressions:

$$\bar{v}_1 = T \cdot v_1 = \|v_1\| \cdot e_2 = (0, \|v_1\|, 0),$$

$$\bar{q}_1 = T \cdot q_1 = (r_0, 0, z_0), \quad \text{and}$$

$$\bar{q}_2 = T \cdot q_2 = (r_0 \cdot \cos \theta_1, \ r_0 \cdot \sin \theta_1, \ z_0),$$

for some $0 \leq \theta_1 \leq 2\pi$, where $r_0 = (1 - z_0^2)^{1/2}$ is the radius of the intersection circle $S^2 \cap \bar{L}_{l_1,q_2}^2$. This is because

1. $\bar{v}_1$ is parallel to $e_2$ and $\|\bar{v}_1\| = \|v_1\|,$

2. $\bar{q}_1$ is orthogonal to $\bar{v}_1$, and

3. the two points $\bar{q}_1$ and $\bar{q}_2$ are on the plane $z = z_0$ and have unit lengths, i.e., $\|\bar{q}_1\| = \|\bar{q}_2\| = 1$. 
The function

$$\theta(t) = (\theta_1 - \frac{1}{r_0} \cdot \|v_1\|) \cdot t^2 + \frac{1}{r_0} \cdot \|v_1\| \cdot t, \quad \text{for } 0 \leq t \leq 1,$$

satisfies the boundary conditions:

$$\theta(0) = 0, \quad \theta(1) = \theta_1, \quad \text{and} \quad \theta'(0) = \frac{1}{r_0} \cdot \|v_1\|.$$

Note that, if the initial speed is very large (i.e., $\|v_1\| > r_0 \cdot \theta_1$), the function $\theta(t)$ is not a monotonically increasing function (i.e., $\theta(t) \geq \theta_1$, for some $0 \leq t \leq 1$). By taking the angle $\theta_1$ sufficiently large (i.e., $2k\pi < \theta_1 \leq 2(k + 1)\pi$, for some $k \geq 1$), the function $\theta(t)$ can be made to be a monotonically increasing function; such a large angle $\theta_1$ would generate extra spins. Let a circular curve $\bar{C}_1(t)$ be defined by

$$\bar{C}_1(t) = (r_0 \cdot \cos \theta(t), r_0 \cdot \sin \theta(t), z_0), \quad \text{for } 0 \leq t \leq 1,$$

then the curve $\bar{C}_1$ satisfies the following boundary conditions:

$$\bar{C}_1(0) = (r_0, 0, z_0) = q_1,$$

$$\bar{C}_1(1) = (r_0 \cdot \cos \theta_1, r_0 \cdot \sin \theta_1, z_0) = q_2,$$

and

$$\bar{C}_1'(0) = (0, r_0 \cdot \theta'(0), 0) = (0, \|v_1\|, 0) = \bar{v}_1.$$

The circular curve $\bar{C}_1$ can be transformed back to a circular curve $C_1$ on $S^2$ as follows:

$$C_1(t) = T^{-1} \cdot \bar{C}_1(t) = T^t \cdot \bar{C}_1(t), \quad \text{for } 0 \leq t \leq 1.$$

It is easy to show that the circular curve $C_1(t) \in S^2$, $0 \leq t \leq 1$, satisfies the given three boundary conditions:

$$C_1(0) = q_1, \quad C_1(1) = q_2, \quad \text{and} \quad C_1'(0) = v_1.$$

The other circular curve $C_2$ can be constructed in a similar way.

### 2.3 Circular Blending in $S^2$

Given two circular curves $C_1(t)$ and $C_2(t) \in S^2$, $0 \leq t \leq 1$, we consider how to blend the two curves on $S^2$ to generate a circular blending spherical curve $Q(t) \in S^2$, $0 \leq t \leq 1$. For each $0 \leq t \leq 1$, we consider the shortest path (geodesic) $\gamma_t$ on $S^2$ which connects the two points $C_1(t)$ and $C_2(t) \in S^2$. The geodesic $\gamma_t$ is a circular arc on the great circle which is the intersection of $S^2$ with the 2D plane determined by the two points $C_1(t)$ and $C_2(t)$, and the center of $S^2$. There are two circular arcs on the great circle which connect the two end points $C_1(t)$ and $C_2(t)$; the geodesic $\gamma_t$ is taken to be the shorter one.
The curve point \( Q(t) \) is defined to be the interior point of \( \gamma_t \) which subdivides the geodesic arc \( \gamma_t \) in the ratio of \( t : 1 - t \). It is trivial to prove from the definition of \( Q \) that:

\[
Q(0) = C_1(0) = q_1 \quad \text{and} \quad Q(1) = C_1(1) = q_2.
\]

However, it is non-trivial to prove the fact that:

\[
Q'(0) = C_1'(0) = v_1 \quad \text{and} \quad Q'(1) = C_2'(1) = v_2.
\]

A formal proof can be given by embedding the great 2-sphere \( S^2 \) into the standard unit 3-sphere \( S^3 \), and using the calculus of unit quaternions \([6, 7]\) on the quaternion curve \( Q(t) \in S^3 \), \( 0 \leq t \leq 1 \). Since \( S^2 \) is a great 2-sphere, the geodesic curve \( \gamma_t \) in \( S^3 \) between the two points \( C_1(t) \) and \( C_2(t) \in S^3 \) is identical with the geodesic curve constructed as a geodesic in \( S^2 \). Thus the curve \( Q(t) \in S^3 \) is identical with the curve constructed in \( S^2 \), and the derivatives of \( Q(t) \) (computed as a curve in \( S^3 \)) are also identical with those computed in \( S^2 \).

When the two end points are antipodal points each other, there are infinitely many great circles which interpolate the two antipodal points; no unique geodesic arc \( \gamma_t \) is defined for two antipodal points. Therefore, we consider only the case in which the two end points \( C_1(t) \) and \( C_2(t) \) are not antipodal points, for any \( 0 \leq t \leq 1 \). When \( C_1(t) \) and \( C_2(t) \) are antipodal points, for some \( 0 \leq t \leq 1 \), the circular blending curve \( Q(t) \) is not guaranteed to be continuous at \( t \); the curve \( Q(t) \) may break and jump into an antipodal point at \( t \). Two antipodal points \( q, -q \in S^3 \) can be identified with a single orientation \( q \in SO(3) \) \([2]\). A way to remedy the degenerate case of antipodal points might be to identify the two antipodal points as a single orientation. However, two antipodal points on the current 2-sphere \( S^2 \) are not antipodal points of the original 3-sphere \( S^3 \). The current 2-sphere \( S^2 \) is the 2-sphere \( T \cdot (r_0 \cdot S^2) + p_0 \) which is not even centered at the origin of \( R^3 \); that is, we had simply transformed the 2-sphere so that it looks like a great 2-sphere \( S^2 \) in order to make further computations simple. Thus it is dangerous to take the antipodal points on the current 2-sphere as an identical orientation. In the degenerate case of antipodal points, it is better to transform the two circular curves back to the original \( S^3 \) and blend them together in \( S^3 \).

### 2.4 Circular Blending in \( S^3 \)

Given two circular curves \( C_1(t), C_2(t) \in S^3 \), \( 0 \leq t \leq 1 \), which satisfy the boundary conditions:

\[
C_1(0) = C_2(0) = q_1, \quad C_1(1) = C_2(1) = q_2, \quad C_1'(0) = v_1, \quad \text{and} \quad C_2'(1) = v_2,
\]

we consider how to blend the two circular curves in \( S^3 \) to generate a hermite quaternion curve \( Q(t), 0 \leq t \leq 1 \), which satisfies the boundary conditions:

\[
Q(0) = q_1, \quad Q(1) = q_2, \quad Q'(0) = v_1, \quad \text{and} \quad Q'(1) = v_2.
\]
For this purpose, the unit quaternion curve \( Q(t), 0 \leq t \leq 1, \) is defined by:

\[
Q(t) = \exp(t \cdot \log(C_2(t) \cdot C_1(t)^{-1})) \cdot C_1(t) \\
= \exp((1 - t) \cdot \log(C_1(t) \cdot C_2(t)^{-1})) \cdot C_2(t),
\]

where the maps: \( \exp : \mathbb{R}^3 \to S^3 \) and \( \log : S^3 \to \mathbb{R}^3 \) are defined as follows:

\[
\exp(0, \theta(a, b, c)) = (\cos \theta, \sin \theta (a, b, c)) \in S^3 \\
\log(\cos \theta, \sin \theta (a, b, c)) = (0, \theta(a, b, c)) \in \mathbb{R}^3,
\]

for \((a, b, c) \in S^3\). (See [6, 7] for the correctness proof of Equation (1).)

### 3 How to Input Angular Velocities

An angular velocity \( \omega_i \ (i = 1, 2) \) can be specified in terms of the axis of rotation \((a_i, b_i, c_i) \in S^2\) and the angle of rotation \(2\theta_i \in [0, 2\pi]\):

\[
\omega_i = 2\theta_i(a_i, b_i, c_i) \in \mathbb{R}^3.
\]

The corresponding 3D rotation with a constant angular velocity \( \omega_i \), for a unit time, is represented by the following unit quaternion:

\[
(cos \theta_i, sin \theta_i (a_i, b_i, c_i)) \in S^3.
\]

Let \( \hat{q}_i \) be defined by:

\[
\hat{q}_i = (cos \theta_i, sin \theta_i (a_i, b_i, c_i)) \cdot q_i \in S^3,
\]

then the unit quaternion \( \hat{q}_i \) represents the resulting solid orientation which is obtained by applying the rotation \((cos \theta_i, sin \theta_i (a_i, b_i, c_i))\) to the solid at orientation \(q_i\).

It is quite cumbersome to specify the two quantities: \((a_i, b_i, c_i) \in S^2\) and \(2\theta_i \in \mathbb{R}\). An easier way to input the angular velocity \( \omega_i \) is to specify the final orientation \( \hat{q}_i \) and automatically compute \( \omega_i \) as follows:

\[
\omega_i = 2\theta_i(a_i, b_i, c_i) \\
= 2 \log(cos \theta_i, sin \theta_i (a_i, b_i, c_i)) \\
= 2 \log(\hat{q}_i \cdot q_i^{-1}).
\]

The corresponding tangent vector \( v_i \in T_{q_i}(S^3) \) is computed as follows:

\[
v_i = \frac{1}{2} \omega_i \cdot q_i \in T_{q_i}(S^3).
\]
(See Appendix A for more details.) Note that the 2D circle generated by \( q_1, \hat{q}_1, q_2 \) is different from the circle generated by \( q_1, v_1, q_2 \). If they were the same, the circle should be the great circle which contains \( q_1 \) and \( \hat{q}_1 \), and is tangent to \( v_1 \); however, the other point \( q_2 \) may not be on the great circle. Therefore, they are different circles, in general.

Figure 2 shows an example of specifying two angular velocities, \( \omega_1 \) and \( \omega_2 \), at the two boundary solid orientations, \( q_1 \) and \( q_2 \), using the above method. Figure 2(c) shows the resulting smooth orientation changes of the 3D solid character “K”. The rotational motion of the solid is obtained by changing its orientation corresponding to the hermite quaternion curve \( Q(t) \), \( 0 \leq t \leq 1 \), constructed on \( SO(3) \) so that it satisfies all the four boundary conditions: \( Q(0) = q_1, Q(1) = q_2, Q'(0) = \omega_1 \), and \( Q'(1) = \omega_2 \).

4 Conclusions

We have described an algorithm to construct a hermite interpolation quaternion curve on \( SO(3) \). The basic curve construction scheme is similar to the circular blending method of Kim and Nam [7]. The hermite interpolation problem in \( S^3 \) has some additional geometric features which make the overall algorithm more efficient. Furthermore, our method preserves the relative advantages (over the previous quaternion curve methods) which have been demonstrated in Kim and Nam [7].

References


Figure 2: (a) $q_1$, $\dot{q}_1$, and $\omega_1 = 2\log(\dot{q}_1 \cdot q_1^{-1})$, (b) $q_2$, $\dot{q}_2$, and $\omega_2 = 2\log(\dot{q}_2 \cdot q_2^{-1})$, and (c) Hermite Interpolation of Solid Orientations.


A Mathematical Preliminaries

In this section, we review some mathematical preliminaries on quaternion calculus and use them to derive the following formula:

\[ q'(t) = \frac{1}{2} \omega \cdot q(t), \]

which relates the angular velocity of a 3D solid to the corresponding quaternion curve derivative.

A.1 Quaternion and Rotation

Given two 4D vectors \( q_i = (q_{i,w}, q_{i,x}, q_{i,y}, q_{i,z}) \in \mathbb{R}^4, \) for \( i = 1, 2, \) we may interpret \( q_i \) as \( q_i = [q_{i,w}, (q_{i,x}, q_{i,y}, q_{i,z})] = (q_{i,w}, q_{i,(x,y,z)}) \in \mathbb{R} \times \mathbb{R}^3. \) The quaternion multiplication \( q_1 \cdot q_2 = q_{12} = (q_{12,w}, q_{12,x}, q_{12,y}, q_{12,z}) = (q_{12,w}, q_{12,(x,y,z)}) \in \mathbb{R} \times \mathbb{R}^3 \equiv \mathbb{R}^4 \) is defined as follows:

\[
q_{12, w} = q_{1,w} q_{2,w} - <q_{1,(x,y,z)}, q_{2,(x,y,z)}> \\
q_{12,(x,y,z)} = q_{2,w} q_{1,(x,y,z)} + q_{1,w} q_{2,(x,y,z)} + q_{1,(x,y,z)} \times q_{2,(x,y,z)}
\]

where \( <, > \) denotes the inner product. The above quaternion multiplication is closed on unit quaternions: for any \( q_1, q_2 \in \mathbb{S}^3, \) we have \( q_1 \cdot q_2 \in \mathbb{S}^3, \) where \( \cdot \) denotes the quaternion multiplication. Moreover, \( (1, 0, 0, 0) = [1,(0,0,0)] \in \mathbb{S}^3 \) is the multiplicative identity, and the inverse of \( q_i \) is given by: \( q_i^{-1} = (q_{i,w}, -q_{i,x}, -q_{i,y}, -q_{i,z}) = (q_{i,w}, -q_{i,(x,y,z)}) = \overline{q}_i \in \mathbb{S}^3, \) where \( \overline{q}_i \) denotes the conjugate of \( q_i. \) Note that the relation \( \overline{q}_1 \cdot \overline{q}_2 = \overline{q}_2 \cdot \overline{q}_1 \) holds for quaternion multiplication.

Given a unit quaternion \( q \in \mathbb{S}^3, \) a 3D rotation \( R_q \in SO(3) \) is defined as follows:

\[
R_q(p) = q \cdot p \cdot \overline{q}, \quad \text{for} \quad p \in \mathbb{R}^3, \tag{2}
\]

where \( \cdot \) is the quaternion multiplication, \( \overline{q} \) is the quaternion conjugate of \( q, \) and \( p = (x, y, z) \) is interpreted as a quaternion \( (0, x, y, z). \) Let \( q = (\cos \theta, \sin \theta (a, b, c)) \in \mathbb{S}^3, \) for some angle \( \theta \) and unit vector \( (a, b, c) \in \mathbb{S}^2, \) then \( R_q \) is the rotation by angle \( 2\theta \) about the axis \( (a, b, c). \) The multiplicative constant, \( 2, \) in the angle of rotation, \( 2\theta, \) is due to the fact that \( q \) appears twice in Equation (2). Also note that \( R_q = R_{-q}; \) that is, two antipodal points, \( q \) and \( -q \) in \( \mathbb{S}^3, \) represent the same rotation in \( SO(3) \) (see [2]). Furthermore, two consecutive rotations, \( R_{q_2} \) applied after \( R_{q_1}, \) produce a composite rotation \( R_{q_2 \cdot q_1}; \) that is, one can easily check that:

\[
R_{q_2}(R_{q_1}(p)) = R_{q_2 \cdot q_1}(p), \quad \text{for} \quad p \in \mathbb{R}^3.
\]
A.2 Quaternion Calculus

The first derivative of a unit quaternion curve \( q(t) \in S^3 \) is always given in the following form:

\[
q'(t) = v_1(t) \cdot q(t), \quad \text{for some } v_1(t) \in R^3.
\] (3)

This is due to the Lie group structure of \( S^3 \) (see [2]). For any differentiable curve \( q(t) \in S^3 \), we may consider a curve \( q(t + s) \) parametrized by \( s \in R \):

\[
q(t + s) = q(t + s) \cdot q(t)^{-1} \cdot q(t).
\]

Then, we have

\[
q'(t) = \left. \frac{d}{ds} \right|_{s=0} q(t + s) = \left. \frac{d}{ds} \right|_{s=0} (q(t + s) \cdot q(t)^{-1}) \cdot q(t) = (q'(t) \cdot q(t)^{-1}) \cdot q(t).
\]

Since the curve \( q_1(s) = q(t + s) \cdot q(t)^{-1} \) passes through the identity element \( 1 \in S^3 \) when \( s = 0 \), we have \( q'_1(0) = q'(t) \cdot q(t)^{-1} \in T_1(S^3) \equiv R^3 \). Therefore, we have \( q'(t) = v_1(t) \cdot q(t) \) for some \( v_1(t) = q'(t) \cdot q(t)^{-1} \in R^3 \).

Let \( q(t) = (q_w(t), q_{x,y,z}(t)) \in S^3 \) be a unit quaternion curve. When the fixed point \( p \in R^3 \) is rotated by \( R_{q(t)} \in SO(3) \), it generates a path \( \hat{p}(t) \in R^3 \):

\[
\hat{p}(t) = R_{q(t)}(p) = q(t) \cdot p \cdot \overline{q(t)}.
\]

The derivative \( \hat{p}'(t) \) is given by:

\[
\hat{p}'(t) = q'(t) \cdot p \cdot \overline{q(t)} + q(t) \cdot p \cdot q'(t) = v_1(t) \cdot q(t) \cdot p \cdot q(t) + q(t) \cdot p \cdot v_1(t) \cdot q(t) = v_1(t) \cdot q(t) \cdot p \cdot q(t) + q(t) \cdot p \cdot v_1(t) \cdot q(t) = v_1(t) \cdot q(t) \cdot p \cdot q(t) - q(t) \cdot p \cdot q(t) \cdot v_1(t) = v_1(t) \cdot q(t) \cdot p \cdot q(t) - v_1(t) \cdot q(t) \cdot (-p) \cdot \overline{q(t)} = v_1(t) \cdot q(t) \cdot p \cdot q(t) + v_1(t) \cdot q(t) \cdot p \cdot \overline{q(t)} = 2v_1(t) \cdot q(t) \cdot p \cdot q(t) = 2v_1(t) \cdot \hat{p}(t) = 2v_1(t) \times \hat{p}(t).
\]
A.3 Angular Velocity

The 3D vector $\omega(t) = 2v_1(t) \in R^3$ can be interpreted as an angular velocity in the following sense:

\[
\left( \frac{d}{dt} R_{q(t)} \right)(p) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left[ R_{q(t+\Delta t)}(p) - R_{q(t)}(p) \right] = \lim_{\Delta t \to 0} \frac{\dot{\omega}(t + \Delta t) - \dot{\omega}(t)}{\Delta t} = \dot{\omega}(t) = \omega(t) \times \dot{p}(t) = \omega(t)(\dot{p}(t)) = \omega(t) \left( R_{q(t)}(p) \right) = (\omega(t)R_{q(t)})(p),
\]

where $\omega(t)$ is a skew-symmetric $3 \times 3$ matrix defined as follows:

\[
\omega(t) = \begin{bmatrix}
0 & -\omega_z(t) & \omega_y(t) \\
\omega_z(t) & 0 & -\omega_x(t) \\
-\omega_y(t) & \omega_x(t) & 0
\end{bmatrix}.
\]

with

$\omega(t) = (\omega_x(t), \omega_y(t), \omega_z(t)) \in R^3$.

Given a smooth rotation of a 3D solid which is represented by a unit quaternion curve $q(t) \in S^3$, $0 \leq t \leq 1$, its angular velocity $\omega(t) \in R^3$ is given by:

$\omega(t) = 2v_1(t) = 2q'(t) \cdot q(t)^{-1} \in R^3$.

Furthermore, given an angular velocity $\omega(t) \in R^3$, $0 \leq t \leq 1$, for the rotation of a given 3D solid, the corresponding quaternion derivative is given by:

$q'(t) = \frac{1}{2} \omega(t) \cdot q(t) \in T_{q(t)}(S^3)$.

Although the angular velocity $\omega(t)$ is a 3D vector, the corresponding quaternion derivative $q'(t)$ is a 4D vector which is contained in a 3-dimensional tangent space $T_{q(t)}(S^3)$ (embedded in $T_{q(t)}(R^4) \equiv R^4$).
B Pseudo Code for the First Hermite Quaternion Curve

Algorithm HermiteCurveS3 \((q1, q2, v1, v2, m)\)

begin

{ Step 1: Initial setup. }
\(n4 \leftarrow \text{compute the normal } v1 \times v2 \times (q2 - q1);\)
\(n4 \leftarrow \text{normalize } n4;\)
\(n3 \leftarrow \text{compute the normal } n4 \times v1 \times v2;\)
\(n3 \leftarrow \text{normalize } n3;\)
\(n1 \leftarrow \text{normalize } v1;\)
\(n2 \leftarrow \text{compute the normal } n3 \times n4 \times n1;\)

{ Step 2: Reduce the problem to the 2-sphere \(S^2\). }
\(T^{-1} \leftarrow \text{the matrix } [n1, n2, n3];\)
\(T \leftarrow \text{the transpose of } T^{-1};\)
\(w0 \leftarrow n4 \cdot q1;\)
\(r0 \leftarrow (1 - w0 \cdot w0)^{1/2};\)
\(p0 \leftarrow w0 \cdot n4;\)
\(\bar{q}1 \leftarrow (1/r0) \cdot (0, n2 \cdot q1, n3 \cdot q1);\)
\(\bar{q}2 \leftarrow (1/r0) \cdot (n1 \cdot q2, n2 \cdot q2, n3 \cdot q2);\)
\(\bar{v}1 \leftarrow (1/r0) \cdot (\|v1\|, 0, 0);\)
\(\bar{v}2 \leftarrow (1/r0) \cdot (n1 \cdot v2, n2 \cdot v2,0);\)

{ Step 3: Call HermiteCurveS2. }
\(\bar{Q}[0,\ldots,m] \leftarrow \text{HermiteCurveS2} (\bar{q}1, \bar{q}2, \bar{v}1, \bar{v}2, m);\)

{ Step 4: Generate points on the hermite quaternion curve. }
for \(i := 0\) to \(m\) do

\(Q[i] \leftarrow T^{-1} \cdot (r0 \cdot \bar{Q}[i]) + p0;\)

return\((Q[0,\ldots,m]);\)
end
Algorithm HermiteCurveS2 \((q1, q2, v1, v2, m)\)

begin

\{ Step 1: Call CircularCurve twice. \}
\(C1[0,\ldots,m] \leftarrow \text{CircularCurve} \ (q1, v1, q2, m);\)
\(C2[0,\ldots,m] \leftarrow \text{CircularCurve} \ (q2, -v2, q1, m);\)
\(C2[0,\ldots,m] \leftarrow \ C2[m,\ldots,0];\)

\{ Step 2: Generate points on the hermite curve. \}
for \(i := 0\) to \(m\) do

\(Q[i] \leftarrow \text{Slerp} \ (C1[i], C2[i], i/m);\)

return\((Q[0,\ldots,m])\);
end

Algorithm CircularCurve \((q1, v1, q2, m)\)

begin

\{ Step 1: Initial setup. \}
\(n2 \leftarrow \text{normalize} \ v1;\)
\(n3 \leftarrow \text{compute the normal} \ (q2 - q1) \times n2;\)
\(n3 \leftarrow \text{normalize} \ n3;\)
\(n1 \leftarrow \text{compute the normal} \ n2 \times n3;\)

\{ Step 2: Reduce the problem to a circle. \}
\(T^{-1} \leftarrow \text{the matrix} \ [n1, n2, n3];\)
\(T \leftarrow \text{the transpose of} \ T^{-1};\)
\(z0 \leftarrow n3 \cdot q1;\)
\(r0 \leftarrow (1 - z0 \cdot z0)^{1/2};\)
\(s1 \leftarrow ||v1||;\)
\(\theta 1 \leftarrow \text{arctan} \ 2(n2 \cdot q2, n1 \cdot q2);\)

\{ Step 3: Generate points on the circular curve. \}
\(b \leftarrow s1/r0;\ \ a \leftarrow \theta 1 - b;\)

for \(i := 0\) to \(m\) do

begin;
\(t \leftarrow i/m;\ \ \theta \leftarrow t \cdot (a \cdot t + b;\)
\(C[i] \leftarrow (r0 \cdot \cos(\theta), r0 \cdot \sin(\theta), z0);\)

end
\[ C[i] \leftarrow T^{-1} \cdot C[i]; \]
\[ \text{end}; \]
\[ \text{return}(C[0, \ldots, m]); \]
\[ \text{end} \]

C Pseudo Code for the Second Hermite Quaternion Curve

**Algorithm** HermiteCurveS3 \((q1, q2, v1, v2, m)\)

\[ \text{begin} \]
\[ \{ \text{Step 3: Call CircularCurve twice. } \} \]
\[ \tilde{C}1[0, \ldots, m] \leftarrow \text{CircularCurve} \,(q1, v1, q2, m); \]
\[ \tilde{C}2[0, \ldots, m] \leftarrow \text{CircularCurve} \,(q2, -v2, q1, m); \]
\[ \tilde{C}2[0, \ldots, m] \leftarrow \tilde{C}2[m, \ldots, 0]; \]
\[ \{ \text{Step 4: Generate points on the hermite quaternion curve. } \} \]
\[ \text{for} \, i := 0 \, \text{to} \, m \, \text{do} \]
\[ \text{begin} \]
\[ C1[i] \leftarrow T^{-1} \cdot (r0 \cdot \tilde{C}1[i]) + p0; \]
\[ C2[i] \leftarrow T^{-1} \cdot (r0 \cdot \tilde{C}2[i]) + p0; \]
\[ Q[i] \leftarrow \text{Slerp} \,(C1[i], C2[i], i/m); \]
\[ \text{end} \]
\[ \text{return}(Q[0, \ldots, m]); \]
\[ \text{end} \]