

# Graphical Abstract

## Orientation control via $C^3$ parametric quaternion interpolation spline curves

### Hot issues

With the rapid development of information technology, the design of rigid body rotational motion has become increasingly important. The unit quaternion provides a natural and efficient way to represent rotation in three-dimensional space, avoiding the universal joint locking phenomenon that may occur in rotation representation and making the rotation operation more stable. Spline curves with local parameters and high continuity can fit the orientation changes of rigid bodies more accurately, thereby achieving more accurate and stable orientation control.

The design method of quaternion spline curves has long been a hot topic of discussion. Some existing works have problems such as low continuity, no local parameters, and poor orientation efficiency. Furthermore, it is impossible to better fit the real rigid body orientation motion.

### Method

In this paper, a parametric polynomial curve is first constructed in the  $\mathbb{R}^3$  space, and then extended to the  $\mathbb{S}^3$  space.

- Given 4 points  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3 \in \mathbb{R}^d$  ( $d = 2, 3$ ) and tension parameters  $\beta_1, \beta_2 \in \mathbb{R}$ , the 6th-degree polynomial curve  $t \mapsto \mathbf{p}(t; \beta_1, \beta_2)$  is constructed as follows

$$\mathbf{p}(t; \beta_1, \beta_2) = \sum_{j=0}^3 C_j(t; \beta_1, \beta_2) \mathbf{P}_j, \quad t \in [0, 1],$$

where

$$\begin{cases} C_0(t; \beta_1, \beta_2) = -\beta_1 B_1^6(t) + \left(-2\beta_1 + \frac{1}{12}\right) B_2^6(t), \\ C_1(t; \beta_1, \beta_2) = B_0^6(t) + B_1^6(t) + \frac{5}{6} B_2^6(t) + \frac{1}{2} B_3^6(t) + \left(2\beta_2 + \frac{1}{12}\right) B_4^6(t) + \beta_2 B_5^6(t), \\ C_2(t; \beta_1, \beta_2) = \beta_1 B_1^6(t) + \left(2\beta_1 + \frac{1}{12}\right) B_2^6(t) + \frac{1}{2} B_3^6(t) + \frac{5}{6} B_4^6(t) + B_5^6(t) + B_6^6(t), \\ C_3(t; \beta_1, \beta_2) = \left(-2\beta_2 + \frac{1}{12}\right) B_4^6(t) - \beta_2 B_5^6(t). \end{cases}$$

are called blending functions.

- Given 4 unit quaternion points  $\mathbf{Q}_0, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3 \in \mathbb{S}^3$  and tension parameters  $\beta_1, \beta_2 \in \mathbb{R}$ , the 6th-degree quaternion curve  $t \mapsto \mathbf{q}(t; \beta_1, \beta_2)$  is constructed as follows

$$\mathbf{q}(t; \beta_1, \beta_2) = \mathbf{Q}_0^{\tilde{C}_0(t; \beta_1, \beta_2)} \prod_{k=1}^3 (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)}, \quad t \in [0, 1],$$

where

$$\tilde{C}_k(t; \beta_1, \beta_2) = \sum_{j=0}^3 C_j(t; \beta_1, \beta_2), \quad \text{for } k = 0, 1, 2, 3,$$

is the summation of the blending functions.

- Let  $\mathbf{Q}_0 = \mathbf{Q}_1, \mathbf{Q}_{n+1} = \mathbf{Q}_n$ , and  $\Delta u_i = u_{i+1} - u_i$ . For all  $i = 1, \dots, n-1$ , the quaternion spline curve  $q : [u_i, u_n] \rightarrow \mathbb{S}^3$  is defined by

$$q(u; \beta) = q\left(\frac{u - u_i}{u_{i+1} - u_i}; \beta_i, \beta_{i+1}\right), \quad u \in [u_i, u_{i+1}],$$

where  $\beta = (\beta_1, \dots, \beta_n)$  is parameter vector.

The spline curve is  $C^3$  continuous and interpolates the given points  $\mathbf{Q}_i$ .

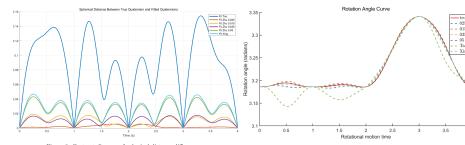


Table 1: Comparison of Different Methods

Method	Degree of the spline curve	Continuity	Local parameter value	MSE	Maximum error (rad)	Time consumed to generate the rigid body motion orientation in Fig. 1 (s)
Xing [12]	4	$C^3$	No local parameters	0.012979	0.256878	0.540 154
Tan [17]	5	$C^2$	$[\alpha_1, \alpha_2, \alpha_3, \alpha_4] = [0.45 \times [1, 1, 1, 1]]$	0.011337	0.188917	0.738 309
Ours	6	$C^3$	$[\alpha_1, \alpha_2, \alpha_3, \alpha_4] = 0.025 \times [1, 1, 1, 1]$	0.001130	0.079421	0.715 989

### Experimental result

Based on the spherical distance error, rotational motion speed, angle and angular speed, the influence of parameter adjustment is qualitatively analyzed, and the regulation effect of the whole and local shape of the spline curve is demonstrated in this paper. Compared with other construction methods, it has the advantages of simple calculation, flexible regulation and higher continuity. It can be widely used in path planning and posture design in simulation and robotics according to actual requirements.

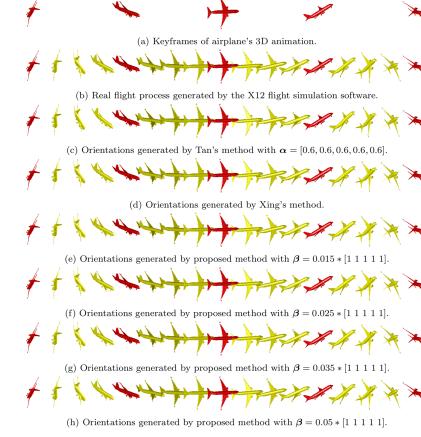


Figure 1: Orientations generated by different interpolation methods.

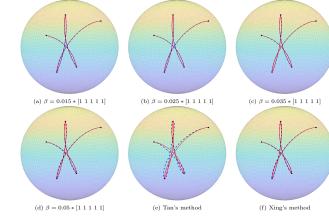


Figure 2: Comparison of the rotational motion trajectory of the right wing.

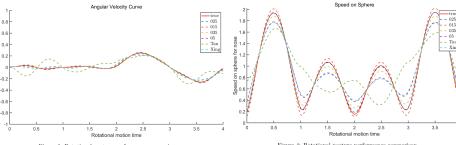


Figure 3: Rotational posture performance comparison.

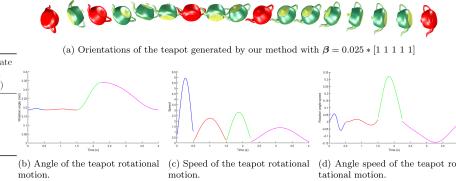


Figure 4: Orientations of the teapot generated by our method with  $\beta = 0.025 \times [1, 1, 1, 1]$ .



Figure 5: Our quaternion spline with non-uniform knot spacing.

# Orientation control via $G^3$ parametric quaternion interpolation spline curves

---

## Abstract

In order to interpolate the real rigid body motion orientation, this paper proposes a unit quaternion spline curve based on the parametric 6th-degree polynomial in Euclidean space. The quaternion spline curve preserves  $G^3$  continuity and interpolates a given sequence of points. Meanwhile, it possesses the local shape adjustability. Through numerical experiments, the influence of parameter variation on motion orientation is specifically analyzed and the performance of the method in the case of sudden rotation is also considered.

*Keywords:* Quaternion spline, Interpolation,  $G^3$  continuity, Tension parameter

---

## 1. Introduction

In the fields of computer animation and inertial navigation, significant attention has been paid to the problem of smoothly interpolating a rigid body's motion through a given sequence of key positions and orientations. In three-dimensional Euclidean space  $\mathbb{R}^3$ , the motion of a rigid body typically consists of both translational and rotational motions. The translational motion of the rigid body can be represented by a position curve, while its rotational motion can be described by an orientation curve in the rotation matrix group  $SO(3)$ . Utilizing unit quaternions to study rotational problems helps to avoid gimbal lock[1, 2]. Moreover, compared to  $3 \times 3$  rotation matrices, quaternions are more efficient in computations, as they require only four parameters. Consequently, the study of rigid body rotational motion using unit quaternion curves has aroused widespread interest among researchers.

The concept of quaternions was proposed by Hamilton[3] as early as 1843. Later, it was gradually replaced by matrix algebra and vector analysis, and thus received little attention. In 1985, Shoemake[2] first intro-

duced quaternions to computer graphics by using the spherical linear interpolation method. Based on geodesic interpolation and the DeCasteljau algorithm, Nielson[4-6] defined a unit quaternion interpolation spline with parameters, but the calculation process of the control vertices is relatively complex. In 2006, Noakes et al.[7] extended the above-mentioned methods and constructed arbitrary dimensional spherical Bézier spline curves that satisfy  $C^1$  continuity. Further, Popiel et al.[8] studied the endpoint properties of the curves and constructed  $C^2$  spherical spline curves without shape parameters. In 2012, Luo et al.[9] constructed a cubic spherical interpolation spline curve that satisfies  $G^2$  continuity.

Inspired by Shoemake, in 1995 Kim et al.[10, 11] proposed a general algebraic method for defining unit quaternion curves in cumulative form. It overcomes the shortcomings of the recursive geometry construction method. It is not only easy to calculate the derivative vector, but also retains some geometric and differential properties. Based on Kim, Xing et al.[12] constructed a  $C^2$  unit quaternion interpolation spline curve. The following year, on the basis of Wu and Xiong[13-15], the continuity of the curve was improved, and a  $C^3$  unit quaternion spline curve was obtained in [16]. However, this curve does not have adjustable parameters. To make the curve have shape controllability, Tan et al.[17] proposed a unit quaternion interpolation spline curve with parameters based on a quintic polynomial curve. This curve has local controllability and can achieve  $G^2$  continuity at the same time. Wang et al. studied other quaternion spline curves in cumulative algebraic forms[18-20] and presented a method for selecting parameters quantitatively [21].

As the demand for higher degree smooth interpolation increases, this paper proposes a new parametric unit quaternion spline curve that automatically interpolates any given sequence of keyframe orientations. This spline curve exhibits properties similar to the corresponding curves in Euclidean space, not only ensuring  $G^3$  continuity but also allowing for local shape control. By adjusting the parameters, our interpolation curve can be closer to the real orientation curve.

## 2. Operations and properties of quaternions

Quaternions are the non-commutative extension of complex numbers. A quaternion  $\mathbf{q}$  can be expressed as  $\mathbf{q} = w + xi + yj + zk$ , where  $w, x, y, z \in \mathbb{R}$ , and  $i, j, k$  are imaginary units, satisfying the following relations:

$$i^2 = j^2 = k^2 = -1,$$

$$\mathbf{i}\mathbf{j} = \mathbf{k}, \mathbf{j}\mathbf{i} = -\mathbf{k}, \mathbf{j}\mathbf{k} = \mathbf{i}, \mathbf{k}\mathbf{j} = -\mathbf{i}, \mathbf{k}\mathbf{i} = \mathbf{j}, \mathbf{i}\mathbf{k} = -\mathbf{j}.$$

By representing the imaginary part as a three-dimensional vector, a quaternion can also be rewritten as an ordered pair  $\mathbf{q} = (w, \mathbf{v})$ , where  $\mathbf{v} = (x, y, z)$ .

Given two quaternions  $\mathbf{q}_i = (w_i, x_i, y_i, z_i)$ ,  $i = 1, 2$ , the relevant quaternion operations are defined as follows.

Addition (or subtraction):

$$\mathbf{q}_1 \pm \mathbf{q}_2 = (w_1 \pm w_2, x_1 \pm x_2, y_1 \pm y_2, z_1 \pm z_2).$$

Multiplication:

$$\mathbf{q}_1 \mathbf{q}_2 = \begin{pmatrix} w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2 \\ w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2 \\ w_1 y_2 + y_1 w_2 + z_1 x_2 - x_1 z_2 \\ w_1 z_2 + z_1 w_2 + x_1 y_2 - y_1 x_2 \end{pmatrix}.$$

Conjugate:

$$\bar{\mathbf{q}} = w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k}.$$

Norm:

$$\|\mathbf{q}\| = \sqrt{\mathbf{q}\bar{\mathbf{q}}} = \sqrt{w^2 + x^2 + y^2 + z^2}.$$

Inverse:

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|^2}.$$

It is important to note: (1) Quaternion multiplication is non-commutative. (2) For any two quaternions, their conjugates satisfy  $\overline{\mathbf{q}_1 \mathbf{q}_2} = \overline{\mathbf{q}_2} \overline{\mathbf{q}_1}$ . (3) When  $\|\mathbf{q}\| = 1$ ,  $\mathbf{q}$  is a unit quaternion. In this case,  $\mathbf{q}^{-1} = \bar{\mathbf{q}} = (w, -x, -y, -z)$ .

For a unit quaternion  $\mathbf{q} = (w, x, y, z) \in \mathbb{S}^3$ , its trigonometric form is given by

$$\mathbf{q} = \cos \theta + \mathbf{n} \sin \theta.$$

Its exponential form is

$$\mathbf{q} = \exp(\theta \mathbf{n}),$$

where  $\theta = \arccos w \in [0, \pi]$  represents the rotation angle, and  $\mathbf{n} = \frac{(x, y, z)}{\sqrt{x^2 + y^2 + z^2}}$  is a unit vector representing the axis of rotation. Any vector  $\mathbf{v}$  rotated by an angle  $2\theta$  around the axis  $\mathbf{n}$ , following the right-hand rule, can be expressed through quaternion multiplication as

$$\mathbf{q}\mathbf{v}\mathbf{q}^{-1} = (\cos \theta, \mathbf{n} \sin \theta)\mathbf{v}(\cos \theta, -\mathbf{n} \sin \theta).$$

The natural logarithm of a unit quaternion  $\mathbf{q}$  can be expressed as  $\log(\mathbf{q}) = \theta\mathbf{n}$ . Then, the exponential function with base  $\mathbf{q}$  is written as

$$\mathbf{q}^{f(t)} = \exp(f(t) \cdot \log(\mathbf{q})) = \cos(f(t)\theta) + \mathbf{n} \sin(f(t)\theta).$$

Furthermore, its first derivative with respect to  $t$  is

$$\frac{d}{dt} \mathbf{q}^{f(t)} = \mathbf{q}^{f(t)} \log(\mathbf{q}) f'(t). \quad (1)$$

Shoemake [21] provided the geodesic curve for connecting two unit quaternions  $\mathbf{q}_1$  and  $\mathbf{q}_2$  as

$$\gamma_{\mathbf{q}_1, \mathbf{q}_2}(t) = \mathbf{q}_1 (\mathbf{q}_1^{-1} \mathbf{q}_2)^t, \quad t \in [0, 1].$$

### 3. $G^3$ quaternion interpolation spline curves with tension parameters

#### 3.1. The construction of 6th-degree curve

##### 3.1.1. The construction of polynomial curve

Given 4 points  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3 \in \mathbb{R}^d$  ( $d = 2, 3$ ) and tension parameters  $\beta_1, \beta_2 \in \mathbb{R}$ , the 6th-degree polynomial curve  $t \mapsto \mathbf{p}(t; \beta_1, \beta_2)$  is constructed as follows

$$\mathbf{p}(t; \beta_1, \beta_2) = \begin{pmatrix} B_0^6(t) & B_1^6(t) & B_2^6(t) & B_3^6(t) & B_4^6(t) & B_5^6(t) & B_6^6(t) \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\beta_1 & 1 & \beta_1 & 0 \\ \frac{1}{12} - 2\beta_1 & \frac{5}{6} & 2\beta_1 + \frac{1}{12} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 2\beta_2 + \frac{1}{12} & \frac{5}{6} & \frac{1}{12} - 2\beta_2 \\ 0 & \beta_2 & 1 & -\beta_2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix},$$

where  $t \in [0, 1]$ , and  $B_i^6(t) = \binom{6}{i} t^i (1-t)^{6-i}$  ( $i = 0, 1, \dots, 6$ ) are 6th-degree Bernstein basis functions. Obviously, this curve is affected by two parameters  $\beta_1, \beta_2$ .

The 6th-degree polynomial curve can also be rewritten by

$$\mathbf{p}(t; \beta_1, \beta_2) = \sum_{j=0}^3 C_j(t; \beta_1, \beta_2) \mathbf{P}_j, \quad t \in [0, 1],$$

where

$$\begin{cases} C_0(t; \beta_1, \beta_2) = -\beta_1 B_1^6(t) + \left(-2\beta_1 + \frac{1}{12}\right) B_2^6(t), \\ C_1(t; \beta_1, \beta_2) = B_0^6(t) + B_1^6(t) + \frac{5}{6} B_2^6(t) + \frac{1}{2} B_3^6(t) + \left(2\beta_2 + \frac{1}{12}\right) B_4^6(t) + \beta_2 B_5^6(t), \\ C_2(t; \beta_1, \beta_2) = \beta_1 B_1^6(t) + \left(2\beta_1 + \frac{1}{12}\right) B_2^6(t) + \frac{1}{2} B_3^6(t) + \frac{5}{6} B_4^6(t) + B_5^6(t) + B_6^6(t), \\ C_3(t; \beta_1, \beta_2) = \left(-2\beta_2 + \frac{1}{12}\right) B_4^6(t) - \beta_2 B_5^6(t). \end{cases}$$

are called blending functions.

Clearly, the blending functions are the linear combination of the tension parameters and the Bernstein basis functions. Moreover, it is easy to verify that the blending functions possess the following properties.

(i) Normalization:  $\sum_{j=0}^3 C_j(t; \beta_1, \beta_2) = 1$ .

(ii) Symmetry:  $C_j(t; \beta_1, \beta_2) = C_{3-j}(1-t; \beta_1, \beta_2)$ , when  $\beta_1 = \beta_2$ .

Based on the properties of Bernstein basis functions, it is not difficult to know

$$\begin{cases} \mathbf{p}(0; \beta_1, \beta_2) = \mathbf{P}_1, & \mathbf{p}(1; \beta_1, \beta_2) = \mathbf{P}_2, \\ \mathbf{p}'(0; \beta_1, \beta_2) = 6\beta_1(\mathbf{P}_2 - \mathbf{P}_0), & \mathbf{p}'(1; \beta_1, \beta_2) = 6\beta_2(\mathbf{P}_3 - \mathbf{P}_1), \\ \mathbf{p}''(0; \beta_1, \beta_2) = \frac{5}{2}(\mathbf{P}_2 - 2\mathbf{P}_1 + \mathbf{P}_0), & \mathbf{p}''(1; \beta_1, \beta_2) = \frac{5}{2}(\mathbf{P}_3 - 2\mathbf{P}_2 + \mathbf{P}_1), \\ \mathbf{p}'''(0; \beta_1, \beta_2) = \left(\frac{1}{4} - 3\beta_1\right)(\mathbf{P}_2 - \mathbf{P}_0), & \mathbf{p}'''(1; \beta_1, \beta_2) = \left(\frac{1}{4} - 3\beta_2\right)(\mathbf{P}_3 - \mathbf{P}_1). \end{cases}$$

### 3.1.2. The construction of quaternion curve

Given 4 unit quaternion points  $\mathbf{Q}_0, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3 \in \mathbb{S}^3$  and tension parameters  $\beta_1, \beta_2 \in \mathbb{R}$ , the 6th-degree quaternion curve  $t \mapsto \mathbf{q}(t; \beta_1, \beta_2)$  is constructed as follows

$$\mathbf{q}(t; \beta_1, \beta_2) = \mathbf{Q}_0^{\tilde{C}_0(t; \beta_1, \beta_2)} \prod_{k=1}^3 (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)}, \quad t \in [0, 1],$$

where

$$\tilde{C}_k(t; \beta_1, \beta_2) = \sum_{j=k}^3 C_j(t; \beta_1, \beta_2), \quad \text{for } k = 0, 1, 2, 3,$$

is the summation of the blending functions. And the following properties are easily verified.

**Lemma 1.** *For tension parameters  $\beta_1, \beta_2 \in \mathbb{R}$ , we have*

- (i)  $\tilde{C}_1(0; \beta_1, \beta_2) = 1, \quad \tilde{C}_2(0; \beta_1, \beta_2) = 0, \quad \tilde{C}_3(0; \beta_1, \beta_2) = 0;$   
 $\tilde{C}_1(1; \beta_1, \beta_2) = 1, \quad \tilde{C}_2(1; \beta_1, \beta_2) = 1, \quad \tilde{C}_3(1; \beta_1, \beta_2) = 0.$
- (ii)  $\tilde{C}'_1(0; \beta_1, \beta_2) = 6\beta_1, \quad \tilde{C}'_2(0; \beta_1, \beta_2) = 6\beta_1, \quad \tilde{C}'_3(0; \beta_1, \beta_2) = 0;$   
 $\tilde{C}'_1(1; \beta_1, \beta_2) = 0, \quad \tilde{C}'_2(1; \beta_1, \beta_2) = 6\beta_2, \quad \tilde{C}'_3(1; \beta_1, \beta_2) = 6\beta_2.$
- (iii)  $\tilde{C}''_1(0; \beta_1, \beta_2) = -\frac{5}{2}, \quad \tilde{C}''_2(0; \beta_1, \beta_2) = -\frac{5}{2}, \quad \tilde{C}''_3(0; \beta_1, \beta_2) = 0;$   
 $\tilde{C}''_1(1; \beta_1, \beta_2) = 0, \quad \tilde{C}''_2(1; \beta_1, \beta_2) = \frac{5}{2}, \quad \tilde{C}''_3(1; \beta_1, \beta_2) = \frac{5}{2}.$
- (iv)  $\tilde{C}'''_1(0; \beta_1, \beta_2) = -(360\beta_1 + 30), \quad \tilde{C}'''_2(0; \beta_1, \beta_2) = -(360\beta_1 + 30),$   
 $\tilde{C}'''_3(0; \beta_1, \beta_2) = 0; \quad \tilde{C}'''_1(1; \beta_1, \beta_2) = 0,$   
 $\tilde{C}'''_2(1; \beta_1, \beta_2) = -(360\beta_2 + 30), \quad \tilde{C}'''_3(1; \beta_1, \beta_2) = -(360\beta_2 + 30).$

Based on Lemma 1 (i), it is easy to know

$$\mathbf{q}(0; \beta_1, \beta_2) = \mathbf{Q}_1, \quad \mathbf{q}(1; \beta_1, \beta_2) = \mathbf{Q}_2, \quad (2)$$

which means the curve interpolates the given points  $\mathbf{Q}_1, \mathbf{Q}_2$ . Furthermore,

**Lemma 2.** *For unit quaternion points  $\mathbf{Q}_0, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3 \in \mathbb{S}^3$ , and tension parameters  $\beta_1, \beta_2 \in \mathbb{R}$ ,*

$$\begin{cases} \mathbf{q}'(0; \beta_1, \beta_2) = 6\beta_1 \mathbf{Q}_1 [\log(\mathbf{Q}_0^{-1} \mathbf{Q}_1) + \log(\mathbf{Q}_1^{-1} \mathbf{Q}_2)], \\ \mathbf{q}'(1; \beta_1, \beta_2) = 6\beta_2 \mathbf{Q}_2 [\log(\mathbf{Q}_1^{-1} \mathbf{Q}_2) + \log(\mathbf{Q}_2^{-1} \mathbf{Q}_3)]. \end{cases}$$

*Proof.* Since  $\tilde{C}_0(t; \beta_1, \beta_2) = 1$ , the unit quaternion curve  $\mathbf{q}(t; \beta_1, \beta_2)$  can be represented by

$$\mathbf{q}(t; \beta_1, \beta_2) = \mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)}.$$

By taking the first-order derivative of the above equation, we have

$$\begin{aligned} \mathbf{q}'(t; \beta_1, \beta_2) &= \mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]' (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\ &\quad + \mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]' (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\ &\quad + \mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]'. \end{aligned} \tag{3}$$

By (1),

$$\left[ (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)} \right]' = (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)} \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)' \log (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k), \tag{4}$$

for  $k = 1, 2, 3$ . Combined with Lemma 1 (i)(ii), we completes the proof.  $\square$

**Lemma 3.** For unit quaternion points  $\mathbf{Q}_0, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3 \in \mathbb{S}^3$ , and tension parameters  $\beta_1, \beta_2 \in \mathbb{R}$ ,

$$\begin{cases} \mathbf{q}''(0; \beta_1, \beta_2) = \mathbf{Q}_1 \left\{ \begin{array}{l} [6\beta_1 \log (\mathbf{Q}_0^{-1} \mathbf{Q}_1) + 6\beta_1 \log (\mathbf{Q}_1^{-1} \mathbf{Q}_2)]^2 \\ + \frac{5}{2} [\log (\mathbf{Q}_1^{-1} \mathbf{Q}_2) - \log (\mathbf{Q}_0^{-1} \mathbf{Q}_1)] \end{array} \right\}, \\ \mathbf{q}''(1; \beta_1, \beta_2) = \mathbf{Q}_2 \left\{ \begin{array}{l} [6\beta_2 \log (\mathbf{Q}_1^{-1} \mathbf{Q}_2) + 6\beta_2 \log (\mathbf{Q}_2^{-1} \mathbf{Q}_3)]^2 \\ + \frac{5}{2} [\log (\mathbf{Q}_2^{-1} \mathbf{Q}_3) - \log (\mathbf{Q}_1^{-1} \mathbf{Q}_2)] \end{array} \right\}. \end{cases}$$

*Proof.* By taking the derivative of (3), we have

$$\begin{aligned}
\mathbf{q}''(t; \beta_1, \beta_2) = & \mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]'' (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\
& + \mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]'' (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\
& + \mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]'' \\
& + 2\mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]' \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]' (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\
& + 2\mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]' (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]' \\
& + 2\mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]' \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]' . \tag{5}
\end{aligned}$$

By (4) and (1),

$$\begin{aligned}
& \left[ (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)} \right]'' \\
& = (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t)} \left[ \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)' \log (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k) \right]^2 \\
& \quad + (\mathbf{Q}_{k-2}^{-1} \mathbf{Q}_{k-1})^{\tilde{C}_k(t; \beta_1, \beta_2)} \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)'' \log (\mathbf{Q}_{k-2}^{-1} \mathbf{Q}_{k-1}) , \tag{6}
\end{aligned}$$

for  $k = 1, 2, 3$ . Combined with (4) and Lemma 1 (i)-(iii), we completes the proof.  $\square$

**Lemma 4.** For unit quaternion points  $\mathbf{Q}_0, \mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3 \in \mathbb{S}^3$ , and tension parameters  $\beta_1, \beta_2 \in \mathbb{R}$ ,

$$\begin{cases} 
\mathbf{q}'''(0; \beta_1, \beta_2) = 6\beta_1 \mathbf{Q}_1 \left\{ (6\beta_1)^2 [\log (\mathbf{Q}_0^{-1} \mathbf{Q}_1) + \log (\mathbf{Q}_1^{-1} \mathbf{Q}_2)]^3 \right. \\ 
\quad + \frac{15}{2} \left\{ [\log (\mathbf{Q}_1^{-1} \mathbf{Q}_2)]^2 - [\log (\mathbf{Q}_0^{-1} \mathbf{Q}_1)]^2 \right\} \\ 
\quad + (-360\beta_1 + 30) \mathbf{Q}_1 [\log (\mathbf{Q}_0^{-1} \mathbf{Q}_1) + \log (\mathbf{Q}_1^{-1} \mathbf{Q}_2)] \} , \\ 
\mathbf{q}'''(1; \beta_1, \beta_2) = 6\beta_2 \mathbf{Q}_2 \left\{ (6\beta_2)^2 [\log (\mathbf{Q}_1^{-1} \mathbf{Q}_2) + \log (\mathbf{Q}_2^{-1} \mathbf{Q}_3)]^3 \right. \\ 
\quad + \frac{15}{2} \left\{ [\log (\mathbf{Q}_2^{-1} \mathbf{Q}_3)]^2 - [\log (\mathbf{Q}_1^{-1} \mathbf{Q}_2)]^2 \right\} \\ 
\quad + (-360\beta_2 + 30) \mathbf{Q}_2 [\log (\mathbf{Q}_1^{-1} \mathbf{Q}_2) + \log (\mathbf{Q}_2^{-1} \mathbf{Q}_3)] \} .
\end{cases}$$

*Proof.* By taking the derivative of (5), we have

$$\begin{aligned}
q'''(t; \beta_1, \beta_2) = & \mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]''' (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\
& + \mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]''' (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\
& + \mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]''' \\
& + 3\mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]'' \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]' (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\
& + 3\mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]' \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]'' (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \\
& + 3\mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]'' (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]' \\
& + 3\mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]' (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]'' \\
& + 3\mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]'' \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]'' \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]' \\
& + 3\mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]'' \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]' \\
& + 3\mathbf{Q}_0 (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]' \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]'' \\
& + 6\mathbf{Q}_0 \left[ (\mathbf{Q}_0^{-1} \mathbf{Q}_1)^{\tilde{C}_1(t; \beta_1, \beta_2)} \right]' \left[ (\mathbf{Q}_1^{-1} \mathbf{Q}_2)^{\tilde{C}_2(t; \beta_1, \beta_2)} \right]' \left[ (\mathbf{Q}_2^{-1} \mathbf{Q}_3)^{\tilde{C}_3(t; \beta_1, \beta_2)} \right]'.
\end{aligned}$$

By (6) and (1),

$$\begin{aligned}
& \left[ (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)} \right]''' \\
= & (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)} \left[ \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)' \log (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k) \right]^3 \\
& + (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)} \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)' [\log (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)]^2 \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)'' \\
& + (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)} \left[ \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)' \log (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k) \right] \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)'' \log (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k) \\
& + (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k)^{\tilde{C}_k(t; \beta_1, \beta_2)} \left( \tilde{C}_k(t; \beta_1, \beta_2) \right)''' \log (\mathbf{Q}_{k-1}^{-1} \mathbf{Q}_k),
\end{aligned}$$

for  $k = 1, 2, 3$ . Combined with (4), (6) and Lemma 1, we completes the proof.  $\square$

### 3.2. The construction of quaternion spline curves

For all  $i = 1, \dots, n$ , given  $n$  unit quaternion points  $\mathbf{Q}_i \in \mathbb{S}^3$ , monotonically increasing knots  $\mathbf{u}_i \in \mathbb{R}$ , and tension parameters  $\beta_i \in \mathbb{R}$ , define the spline curve as follows.

**Definition 1.** Let  $\mathbf{Q}_0 = \mathbf{Q}_1$ ,  $\mathbf{Q}_{n+1} = \mathbf{Q}_n$ , and  $\Delta u_i = u_{i+1} - u_i$ . The quaternion spline curve  $\mathbf{q} : [u_1, u_n] \rightarrow \mathbb{S}^3$  is defined by

$$\mathbf{q}(u; \boldsymbol{\beta}) = \mathbf{q}\left(\frac{u - u_i}{u_{i+1} - u_i}; \beta_i, \beta_{i+1}\right), \quad u \in [u_i, u_{i+1}], \quad (7)$$

where  $i = 1, \dots, n-1$ , and  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_n)$  is parameter vector.

By (2), the spline curve  $\mathbf{q}(u; \boldsymbol{\beta})$  interpolates all quaternion points  $\{\mathbf{Q}_i\}_0^{n+1}$ . And obviously, it is  $C^\infty$  when  $u \in (u_i, u_{i+1})$ . For  $m = 1, 2, 3$ , and all  $i = 2, \dots, n-1$ , the left and right derivative vectors of the spline curve at the knots  $u_i$  are as follows

$$\begin{cases} \mathbf{q}_-^{(m)}(u_i; \boldsymbol{\beta}) = \mathbf{q}^{(m)}\left(\frac{u - u_{i-1}}{u_i - u_{i-1}}; \beta_{i-1}, \beta_i\right)|_{u=u_i}, & u \in [u_{i-1}, u_i], \\ \mathbf{q}_+^{(m)}(u_i; \boldsymbol{\beta}) = \mathbf{q}^{(m)}\left(\frac{u - u_i}{u_{i+1} - u_i}; \beta_i, \beta_{i+1}\right)|_{u=u_i}, & u \in [u_i, u_{i+1}]. \end{cases} \quad (8)$$

**Theorem 1.** The spline curve  $\mathbf{q} : [u_1, u_n] \rightarrow \mathbb{S}^3$  defined by (7) is  $G^3$ .

*Proof.* Following from (8) and Lemma 2-4, we have, for all  $i = 2, \dots, n-1$ ,

$$\begin{cases} \mathbf{q}'_-(u_i; \boldsymbol{\beta}) = \frac{6\beta_i}{\Delta u_i} \mathbf{Q}_i [\log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i) + \log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})], \\ \mathbf{q}'_+(u_i; \boldsymbol{\beta}) = \frac{6\beta_i}{\Delta u_{i+1}} \mathbf{Q}_i [\log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i) + \log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})]. \end{cases}$$
  

$$\begin{cases} \mathbf{q}''_-(u_i; \boldsymbol{\beta}) = \frac{1}{(\Delta u_i)^2} \mathbf{Q}_i \left\{ [6\beta_i \log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i) + 6\beta_i \log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})]^2 \right. \\ \quad \left. + \frac{5}{2} [\log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1}) - \log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i)] \right\}, \\ \mathbf{q}''_+(u_i; \boldsymbol{\beta}) = \frac{1}{(\Delta u_{i+1})^2} \mathbf{Q}_i \left\{ [6\beta_i \log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i) + 6\beta_i \log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})]^2 \right. \\ \quad \left. + \frac{5}{2} [\log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1}) - \log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i)] \right\}. \end{cases}$$

$$\left\{ \begin{array}{l} \mathbf{q}_-'''(u_i; \boldsymbol{\beta}) = \frac{6\beta_i}{(\Delta u_i)^3} \mathbf{Q}_i \left\{ (6\beta_i)^2 [\log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i) + \log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})]^3 \right. \\ \quad + \frac{15}{2} \left\{ [\log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})]^2 - [\log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i)]^2 \right\} \\ \quad \left. + (-360\beta_i + 30) \mathbf{Q}_i [\log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i) + \log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})] \right\}, \\ \mathbf{q}_+'''(u_i; \boldsymbol{\beta}) = \frac{6\beta_i}{(\Delta u_{i+1})^3} \mathbf{Q}_i \left\{ (6\beta_i)^2 [\log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i) + \log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})]^3 \right. \\ \quad + \frac{15}{2} \left\{ [\log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})]^2 - [\log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i)]^2 \right\} \\ \quad \left. + (-360\beta_i + 30) \mathbf{Q}_i [\log(\mathbf{Q}_{i-1}^{-1} \mathbf{Q}_i) + \log(\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})] \right\}. \end{array} \right.$$

Then, by the interpolation of the spline curve, for  $m = 0, 1, 2, 3$ ,

$$\mathbf{q}_-^{(m)}(u_i; \boldsymbol{\beta}) = \left( \frac{\Delta u_{i+1}}{\Delta u_i} \right)^m \mathbf{q}_+^{(m)}(u_i; \boldsymbol{\beta}), \quad (9)$$

which complete the proof.  $\square$

It is easy to know from Eqs.(9) that the 6th-degree quaternion interpolation spline curve can achieve  $C^3$  continuity when the knot spacing is uniform.

## 4. Experiments

This section presents experimental results that validate the effectiveness and versatility of the proposed scheme.

### 4.1. The case of uniform knot spacing

First, produced by different spline curves with uniform knot spacing, 3D animations of aircrafts with the same keyframe orientations are showed in Fig. 1. For clarity in presentation, all positional trajectories are represented as linear segments. Fig. 1(a) and (b) show the five keyframes of aircraft and the real flight process respectively, which are derived from real flight orientation data generated by the X12 flight simulation software. Fig. 1(c) shows flight process of which orientations are produced by Tan's quaternion spline curve [17] with tension vector  $\boldsymbol{\alpha} = 0.6*[1 1 1 1]$ . Fig. 1(d) is by Xing's quaternion spline curve [12]. Fig. 1(e)-(h) are produced by our  $C^3$  quaternion interpolation spline curves with uniform knot spacing and different tension parameter vectors. The five given keyframes of aircraft are red, and the new generated frames are yellow.

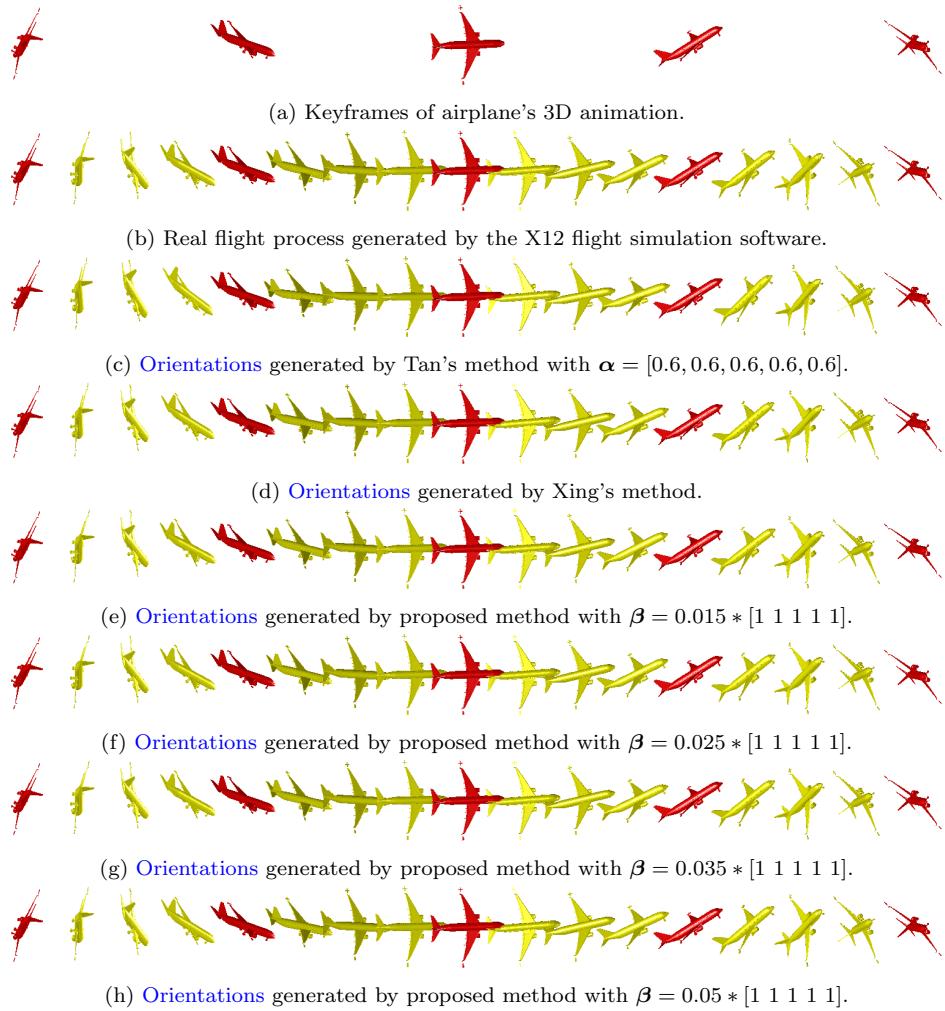


Figure 1: Orientations generated by different interpolation methods.

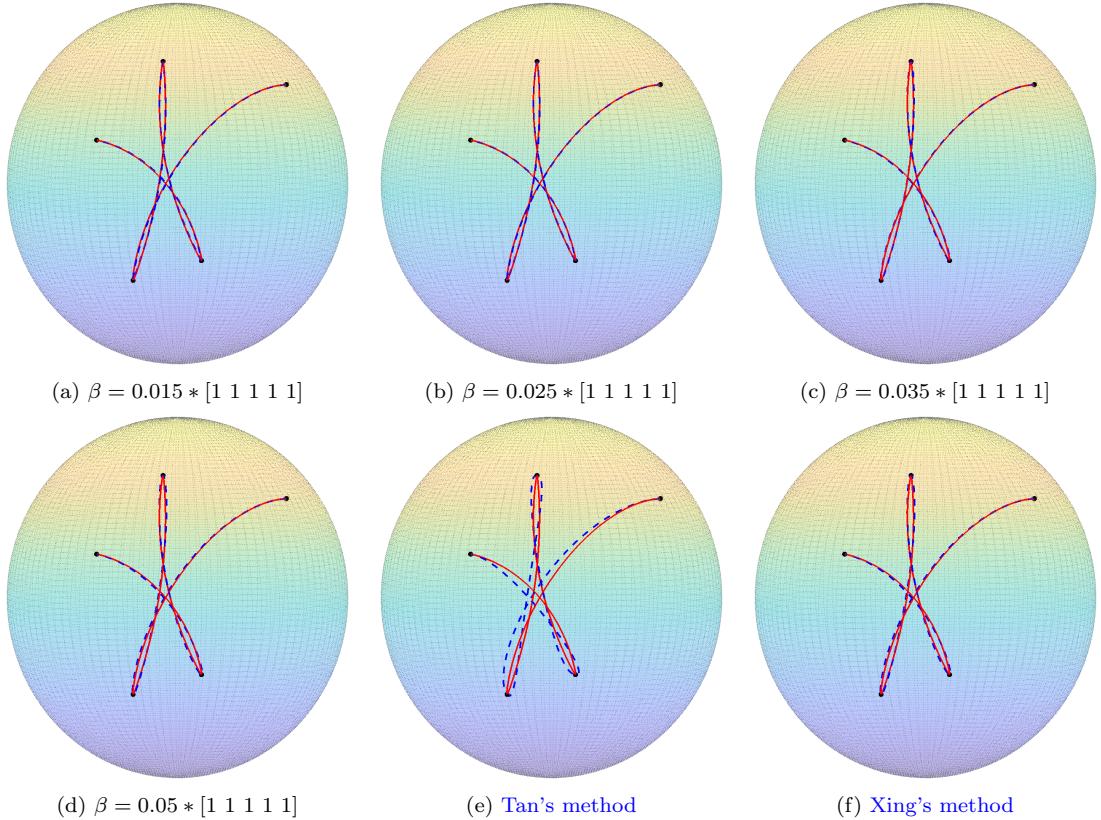


Figure 2: Comparison of the rotational motion trajectory of the right wing.

The flight processes generated by different methods are quite similar. We further consider the rotational motion trajectories. Fig. 2 compares rotational trajectory of the right wingtip in real flight process in Fig. 1(b) with the others produced by different methods in Fig. 1(c)-(h). The red line shows the real trajectory. The blue dotted lines are interpolation curves from different methods. As can be seen from the Fig. 2, our method and [Xing's](#) [12] are better than [Tan's](#) [17].

In order to further compare ours and Xing's method, we conduct an analysis based on spherical distance. Fig. 3 illustrates the spherical distance errors between real trajectories and interpolation curves in Fig. 2. Notably, Tan's method exhibits the largest errors, whereas the minimal error occurs at  $\beta = 0.025 * [1 \ 1 \ 1 \ 1 \ 1]$ . Xing's method demonstrates similar performance to ours at  $\beta = 0.05 * [1 \ 1 \ 1 \ 1 \ 1]$ , as Xing's result is a special case of our spline curve with  $\beta = \frac{5}{96} * [1 \ 1 \ 1 \ 1 \ 1]$ .

Extract the speed, angle and angular speed of the rotational motion and continue to analyze the effects of the parameters. Fig. 4-6 show that the peaks and troughs of velocity, angle, and angular velocity curves align most closely with real curve where  $\beta = 0.025 * [1 \ 1 \ 1 \ 1 \ 1]$ . At the same time, the disadvantages of Tan's method can be seen that the peaks and troughs positions of the speed, angle, and angular speed do not match the real position.

A comprehensive comparison of polynomial degree, continuity, parameter values, mean squared error (MSE) between interpolation and real trajectories, maximum error, and computational time is presented in Tab. 1. The first advantage of the proposed method is that it has controllable parameters. Changing one parameter will affect the spline curves of the two adjacent sections, so that the real motion orientation of the rigid body can be better restructured. Secondly, the proposed quaternion interpolation spline curve with uniform knot spacing is  $C^3$ , which is more advantageous in theory. When applied to the real motion orientation interpolation of rigid body, the change of speed, angle and angular speed is more controllable (see Fig. 4-6). The third is the operation efficiency. Our method runs faster than [Tan's \[17\]](#) on the basis of higher continuity, and is slightly slower than [Xing's \[12\]](#), but adds local adjustable parameters.

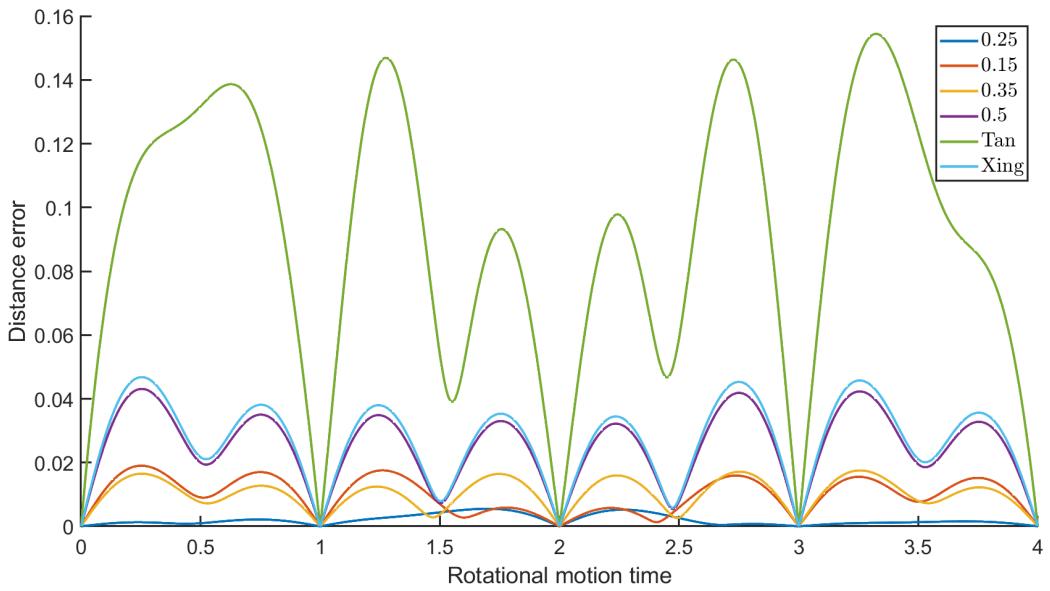


Figure 3: Comparison of the spherical distance error.

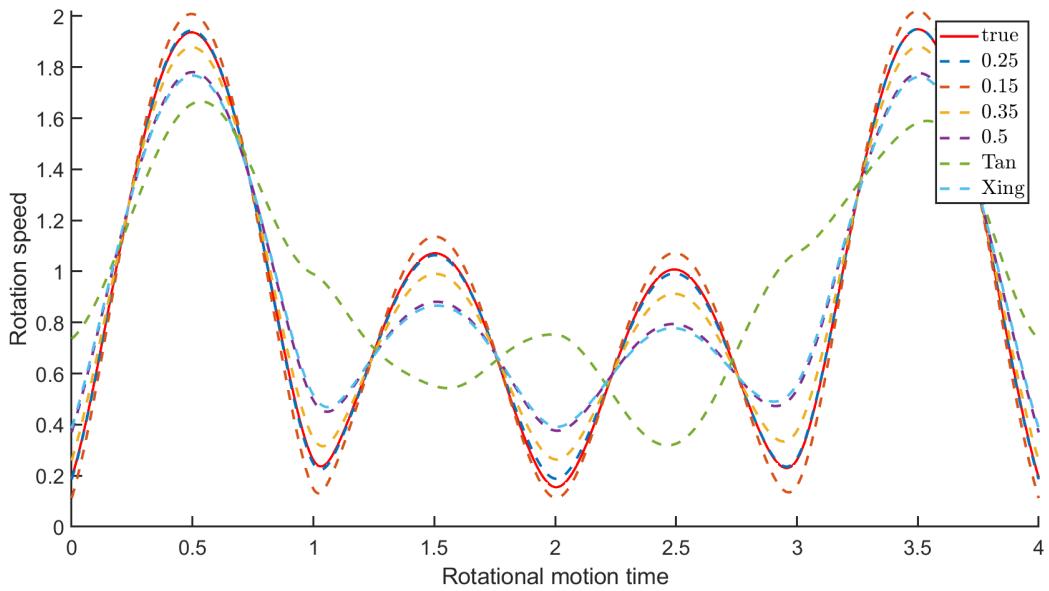


Figure 4: Comparison of the speed of the rotational motion.

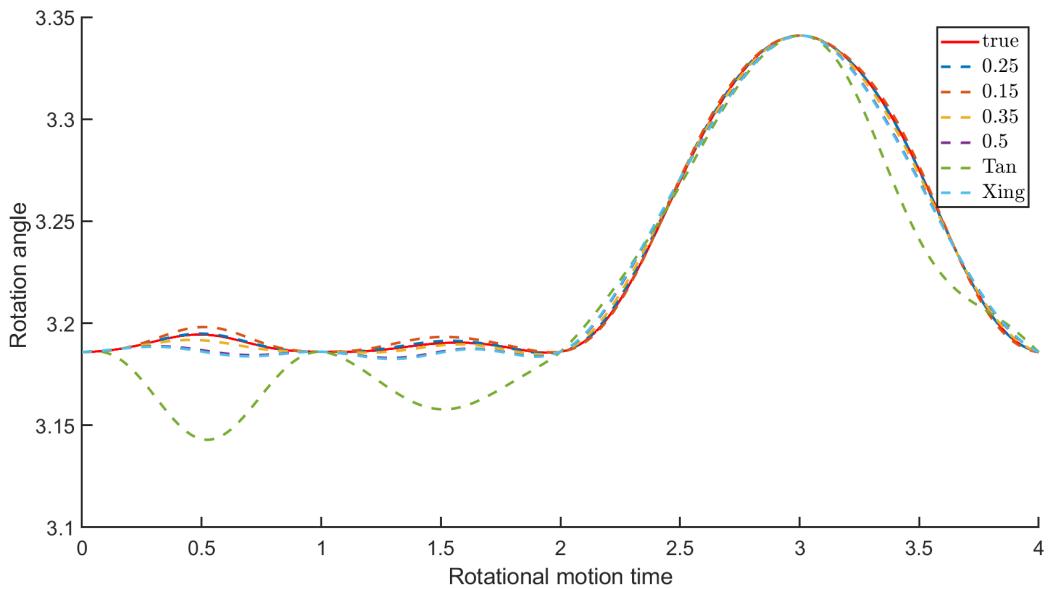


Figure 5: Comparison of the angle of the rotational motion.

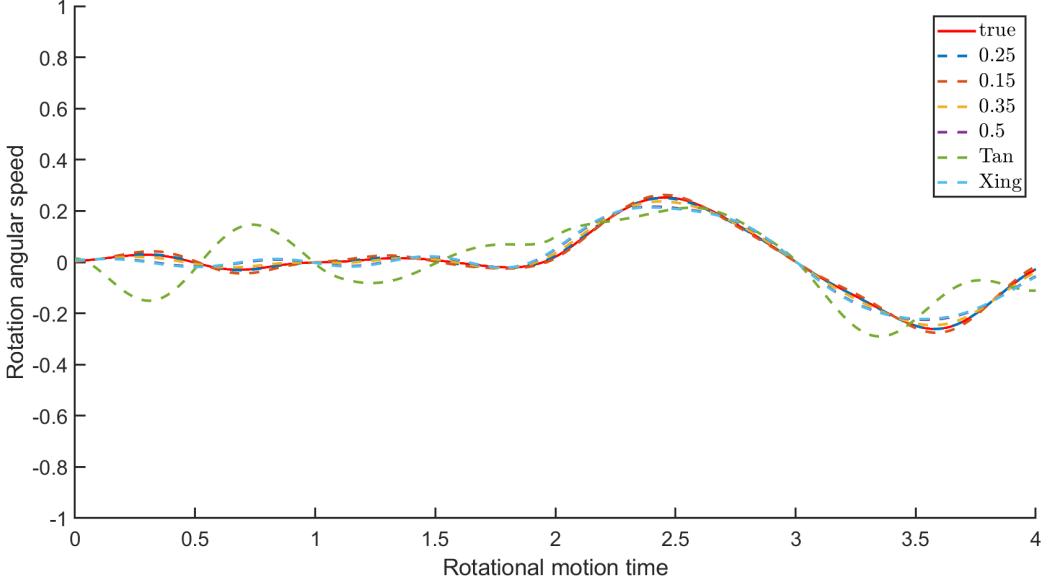


Figure 6: Comparison of the angular speed of the rotational motion.

Table 1: Comparison of Different Methods

Method	Degree of the spline curve	Continuity	Local parameter value	MSE	Maximum error (rad)	Time consumed to generate the rigid body motion orientation in Fig. 1 (s)
Xing [12]	4	$C^3$	No local parameters	0.012979	0.256878	0.540 154
Tan [17]	5	$C^2$	$[\alpha_1, \alpha_2, \alpha_3, \alpha_4] = 0.45 * [1, 1, 1, 1]$	0.011337	0.188917	0.738 309
Ours	6	$C^3$	$[\alpha_1, \alpha_2, \alpha_3, \alpha_4] = 0.025 * [1, 1, 1, 1]$	0.001130	0.079421	0.715 989

#### 4.2. The case of non-uniform knot spacing

To evaluate the generalization ability of our method with non-uniform knot spacing, a teacup rotation model is introduced in this subsection as a new test scenario (see Fig. 7). Given the non-uniform time knot vector  $[0, 0.5, 1.5, 2.25, 4]$ , relatively short length of the interval between adjacent knots means that the teacup suddenly rotates during this period, i.e., the orientation of the teacup changes suddenly. At this time, applying the proposed method can also achieve keyframe orientations interpolation, and the corresponding rotation angular curve is still continuous, as shown in Fig. 7(a)

and (b). Furthermore, Fig. 7(c) and (d) show that the teacup rotational motion has relatively uniform speed and angular speed. It should be noted that the quaternion spline curve with non-uniform knot spacing is  $G^3$  continuous, so the speed and angular speed curves are discontinuous at the interpolated knots.

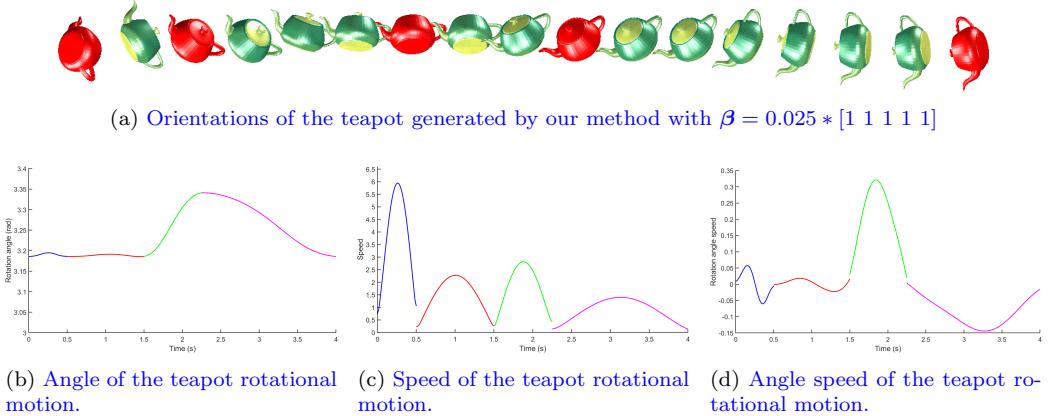


Figure 7: Our quaternion spline with non-uniform knot spacing

## 5. Conclusion

In this paper, we present a new method for constructing quaternion spline curves with parameters. Firstly, the 6th-degree polynomial and quaternion curves with parameters are constructed by blending functions. The properties of the curves are further studied. Secondly, the quaternion spline curve is constructed and proved to achieves  $G^3$  continuity with non-uniform knot spacing and  $C^3$  continuity with uniform knot spacing. Based on the spherical distance error, rotational motion speed, angle and angular speed, the influence of parameter adjustment on rigid rotational motion is demonstrated. Compared with other construction methods, the proposed spline curve has the advantages of simple calculation, higher continuity and flexible regulation. The method also performs very well under highly discontinuous orientations (e.g., sudden rotations). So it can be widely used in path planning and orientation design in simulation and robotics according to actual requirements.

At present, this paper only carries out qualitative analysis on the parameter influence of the constructed quaternion spline curve. In the following

research, we will focus on the quantitative analysis, so as to propose the optimal parameter value scheme.

## References

- [1] Hamilton, W. R., *Elements of Quaternions*, Longmans, Green, & Company, London, 1st ed., vol. 1-2, 1866.
- [2] Shoemake, K., *Animating Rotation with Quaternion Curves*, ACM SIGGRAPH Comput. Graph., 19(3): 245-254, 1985.
- [3] Hamilton, W. R., *On a New Species of Imaginary Quantities Connected with the Theory of Quaternions*, Proc. R. Ir. Acad., 2: 424-434, 1840.
- [4] Nielson, G. M., Heiland, R. W., *Animated Rotations Using Quaternions and Splines on a 4D Sphere*, Program. Comput. Softw., 18(4): 145-154, 1992.
- [5] Nielson, G. M., *Smooth Interpolation of Orientations*, in *Designing Fair Curves and Surfaces* (N. Sapidis, Ed.), SIAM, Philadelphia, 1993, pp. 57-76.
- [6] Nielson, G. M.,  *$\nu$ -Quaternion Splines for the Smooth Interpolation of Orientations*, IEEE Trans. Vis. Comput. Graph., 10(2): 224-229, 2004.
- [7] Noakes L. *Spherical splines* // Geometric Properties for Incomplete Data. Dordrecht: Springer Netherlands, 2006: 77-101.
- [8] Popiel T, Noakes L.  *$C^2$  spherical Bézier splines*. Computer aided geometric design, 2006, 23(3): 261-275.
- [9] Luo, Z., Wang, Q., Fan, X., Gao, Y., Shui, P., *Generalized Rational Bézier Curves for Rigid Body Motion Design*, Vis. Comput., 32(8): 1071-1084, 2016.
- [10] Kim, M. J., Kim, M. S., Shin, S. Y., *A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives*, Comput.-Aided Des., 27(12): 887-894, 1995.
- [11] Kim, M. S., Nam, K. W., *Interpolating Solid Orientations with Circular Blending Quaternion Curves*, Comput. Aided Geom. Des., 12(5): 455-476, 1995.
- [12] Xing, Y., Fan, W., Tan, J., Xu, R., *A  $C^2$ -Continuous Unit Quaternion Interpolatory Spline Curve*, Adv. Mech. Eng., 9(7): 1-10, 2017.

- [13] Wu, H.,  *$C^3$ -Continuous 4-5-5-4 Alternating B-type Parametric Spline Curves*, J. Anhui Inst. Technol., 17(2): 1-7, 1997.
- [14] Xiong, J., Guo, Q., Zhu, G.,  *$C^3/C^4$ -Continuous Interpolating Curves with Global/Local Shape Control*, Numer. Math. Theor. Meth. Appl., 32(3): 165-173, 2011.
- [15] Wu, H., Du, W., *Shape-Parameterized Interpolating Curves*, J. Comput.-Aided Des. Comput. Graph., 18(3): 352-357, 2006.
- [16] Xing, Y., Xu, R., Tan, J., Fan, W., Hong, L., *A Class of Generalized B-spline Quaternion Curves*, J. Comput. Anal. Appl., 19(5): 873-883, 2015.
- [17] Tan, J., Xing, Y., Fan, W., Hong, P., *Smooth Orientation Interpolation Using Parametric Quintic-Polynomial-Based Quaternion Spline Curve*, J. Comput. Appl. Math., 329: 144-153, 2018.
- [18] Wang, Q., Liu, M., Zhou, T.,  *$G^2$ -Continuous Unit Quaternion Spline Curves*, J. Liaoning Norm. Univ. (Nat. Sci. Ed.), 46(4): 438-445, 2023.
- [19] Wang, Q., Xu, Y., Zhou, T., *A Class of  $C^3$ -Continuous Singular Hybrid Quaternion Spline Curves*, J. Liaoning Norm. Univ. (Nat. Sci. Ed.), 47(4): 449-457, 2024.
- [20] Wang, Q., Zhou, Y., Xu, Y., *A Class of Parametric  $C^3$ -Continuous Unit Quaternion Interpolation Spline Curves*, J. Gannan Norm. Univ., 44(6): 31-38, 2023.
- [21] Wang, Q., Zhou, T., Zhu, J., et al., *Energy Optimization of Quaternion Spline Curves*, J. Comput.-Aided Des. Comput. Graph., 36(1): 1-11, 2024. (ISSN 1003-9775)