JST: Smart Systems and Devices
Volume 35, Issue 2, May 2025, 035-042

Orientation Motion Planning Using Cubic Spline Interpolation Based on Euler Parameters

Nguyen Quang Hoang[1*], Duong Minh Hai[2], Dinh Van Phong[1]
[1]*Hanoi University of Science and Technology, Ha Noi, Vietnam*
[2] *Hanoi University of Business and Technology, Ha Noi, Vietnam*
[*]*Corresponding author email: hoang.nguyenquang@hust.edu.vn*

Abstract

Many engineering applications require smooth orientation planning, i.e., interpolating the orientation of a rigid body so that its motion is smooth through intermediate poses. This smooth motion ensures for instance the continuity of the control torques. There are several ways to represent the orientation of a rigid body, so there are also different ways to plan motion for orientation. Each way has its advantages and disadvantages. In general, the problem of motion planning for the orientation has been less studied due to its complexity compared to motion planning for the endpoint. This paper presents the motion planning for the orientation using Euler parameters when the initial and final directions, and a set of intermediate directions are known. First, the Euler parameters are interpolated using cubic splines, and then they are normalized. Numerical simulations are carried out to validate the effectiveness of the proposed method. The proposed algorithms presented here preserve the fundamental properties of the interpolated rotation. The algorithms presented in this paper provide interpolation tools for rotation that are accurate, easy to implement.

Keywords: Euler's parameters, orientation motion planning, cubic splines, numerical simulation.

1. Introduction

In many engineering applications, such as satellite attitude control, multi-body dynamics, and robot motion control in the workspace, it is necessary not only to interpolate the trajectory of a point of the body but also to interpolate its orientation or plan a smooth rotational motion for the body. The trajectory planning of a point in space can be carried out independently for its three coordinates using interpolation methods. This problem has been thoroughly addressed in various studies [1, 3]. There are several approaches to motion planning for a point moving from an initial position to a final position, such as using cubic, quintic, or higher-order polynomials, harmonic functions, or designing trajectories based on velocity profiles in the form of triangular or trapezoidal shapes [2-8]. However, orientation motion planning is more complex and depends on the choice of parameters used to describe the orientation.

Different from point representation, there are several ways to describe the orientation of a body in space, such as the direction cosine matrix with nine elements, the Euler angles with three variables, the rotation axis and rotation angle, or the Euler parameters with 4 variables. As we know, the minimum number of parameters describing the orientation is three, so when planning a motion for an orientation with more than three parameters, we need to pay attention to the constraints between them.

Using the direction cosine matrix is an intuitive description and convenient for coordinate transformation. Using Euler angles is convenient for representing component rotations and calculating angular velocities in terms of their derivatives. With this minimal set of parameters, we can plan the motion for each Euler angle. However, it is difficult for the kinematic differential equation when the body passes through or nears singular poses.

Some authors have used the direction cosine matrix to plan the motion for the direction [9, 10] Accordingly, the cosine matrix is calculated only by the sum of the direction cosine matrices at the nodes with the weights being the shape functions of time. The interpolation matrix is guaranteed by the direction cosine matrices at the nodes thanks to the shape function. However, the orthogonality of the direction cosine matrix outside the nodes is difficult to ensure. Theoretically, it is possible to approximate the interpolation matrix to an orthogonal matrix. However, this is not easy to do and will also incur costs in terms of computation time and memory space.

In [10] authors expoilted a class of spline algorithms for generating orientation trajectories that approximately minimize angular acceleration.

ISSN 2734-9373
https://doi.org/10.51316/jst.182.ssad.2025.35.2.5
Received: Mar 18, 2025; revised: Apr 19, 2025
accepted: Apr 19, 2025

A twice-differentiable curve is constructed based on the rotation matrix that interpolates a given ordered set of rotation matrices at specified knot times. In this work, rotation matrices are parametrized by the unit quaternion, canonical co-ordinate, and Cayley-Rodrigues representations.

If Euler angles are chosen to describe orientation, functions can be used to simultaneously interpolate all three Euler angles. However, analytical singularities often occur with Euler angles when the rotation axes of two successive rotations coincide. If Euler parameters or unit quaternions are used, interpolating these parameters is relatively challenging, as it requires maintaining that their modulus is equal to one.

Some methods have been developed for quaternion interpolation [9-11, 13-15]. In [11] authors use *ν*-quaternion splines for interpolation, in which an iterative method is applied to determine the parameters appearing in the interpolation method. In [12] authors use the spherical linear interpolation for connecting two consecutive orientations described by two quaternions. However, with this method, the angular velocity of the object is not continuous at the control points.

To avoid singularities caused by the choice of orientation parameters for the rigid body, this paper selects Euler parameters or unit quaternions to describe the orientation of the rigid body. The interpolation method employs cubic spline functions to smoothly interpolate the Euler parameters, which are then normalized to ensure their magnitude remains unitary for describing the object's orientation. The proposed orientation planning over time ensures smooth transitions of the body from the initial direction to the final direction. Numerical simulations are conducted to verify the effectiveness of the proposed method.

The remainder of this paper is structured as follows: Section 2 presents the orientation kinematics of the rigid body, including Euler angles and Euler parameters, to clarify the singularities when using Euler angles and the advantages of Euler parameters. Section 3 discusses orientation motion interpolation using Euler parameters. Section 4 presents numerical simulation results. Finally, a conclusion is provided in the last section.

2. Orientation Kinematics of a Rigid Body

There are several approaches to representing the orientation of a rigid body in space. For a free rigid body, the minimum number of parameters required to describe its orientation is three. When the number of orientation parameters exceeds three, the number of constraint equations must be equal to the number of parameters minus three. Below are some common methods for representing the orientation of a rigid body in space:

- Euler angles, Cardan angles, Roll-Pitch-Yaw angles (three independent angles),
- Direction cosine matrix with 9 elements, defined by the special orthogonal group SO(3),
- Axis/angle (θ, $\mathbf{u}$), with the constraint $\mathbf{u}^T\mathbf{u} = 1$,
- Euler parameters or unit quaternions: $\boldsymbol{\xi} = [\eta, \boldsymbol{\varepsilon}^T]^T$, with the constraint $\boldsymbol{\xi}^T\boldsymbol{\xi} = \eta^2 + \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = 1$.

A 3×3 direction cosine matrix consists of 9 real numbers, with six constraint equations. Since the rotation matrix has six redundant parameters, this representation is computationally expensive. Moreover, using orientation errors in the form of a rotation matrix for control loops is not straightforward.

2.1 Euler Angles and Their Limitations

With three Euler angles, the orientation of a body in space is determined by three consecutive rotations around three axes in a specific sequence. However, this representation contains singularities. For example, with the three Euler angles [ψ, θ, φ] rotating sequentially around the current axes in the *z*-*x*-*z* sequence, the direction cosine matrix is given as follows:

$$\mathbf{R} = \mathbf{R}_{z0}(\psi)\mathbf{R}_{x1}(\theta)\mathbf{R}_{z2}(\varphi) = \{r_{i,j}\}$$
$$= \begin{bmatrix} c\psi c\varphi - s\psi c\theta s\varphi & -c\psi s\varphi - s\psi c\theta c\varphi & s\psi s\theta \\ s\psi c\varphi + c\psi c\theta s\varphi & -s\psi s\varphi + c\psi c\theta c\varphi & -c\psi s\theta \\ s\theta s\varphi & s\theta c\varphi & c\theta \end{bmatrix}$$

When the element r_{33} of the drection cosine matrix is different from ±1, using inverse trigonometric functions, the Euler angles can be determined. However, when $r_{33} = 1$, so $\theta = 0$ or $r_{33} = -1$, so $\theta = \pi$, the angles ψ and φ become indeterminate, and only their sum or difference can be computed.

The angular velocity of a rigid body characterizes the rate of change of its orientation in a reference system. The relationship between angular velocity and the direction cosine matrix is expressed by the following equation:

$$\tilde{\boldsymbol{\omega}}^{(0)} = \dot{\mathbf{R}}\mathbf{R}^T \tag{1}$$

where $\tilde{\boldsymbol{\omega}}^{(0)}$ is the skew-symmetric matrix corresponding to the angular velocity vector $\boldsymbol{\omega}^{(0)}$ in the fixed coordinate system.

With three *z*-*x*-*z* Euler angles $\mathbf{q} = [\psi, \theta, \varphi]^T$, we have:

$$\boldsymbol{\omega}^{(0)} = \mathbf{Q}(\mathbf{q})\dot{\mathbf{q}} \tag{2}$$

$$\dot{\mathbf{q}} = \mathbf{Q}^{-1}(\mathbf{q})\boldsymbol{\omega}^{(0)} \tag{3}$$

where

$$\mathbf{Q} = \begin{bmatrix} 0 & \cos\psi & \sin\psi\sin\theta \\ 0 & \sin\psi & -\cos\psi\sin\theta \\ 1 & 0 & \cos\theta \end{bmatrix}$$

and

$$\mathbf{Q}^{-1} = \begin{bmatrix} -\dfrac{\sin\psi\cos\theta}{\sin\theta} & \dfrac{\cos\psi\cos\theta}{\sin\theta} & 1 \\ \cos\psi & \sin\psi & 0 \\ \dfrac{\sin\psi}{\sin\theta} & \dfrac{-\cos\psi}{\sin\theta} & 0 \end{bmatrix}.$$

It is clear that when θ= 0 or θ= π, matrix $\mathbf{Q}^{-1}$ becomes indeterminate, and it is impossible to determine $\dot{\mathbf{q}}$ from the angular velocity of the object.

2.2. Euler Parameters and Unit Quaternion

The axis/angle parameters and quaternion have four parameters, allowing them to represent global orientation without singularities. The axis/angle parameters consist of a unit rotation axis and a corresponding rotation angle. Euler's finite rotation theorem states that any rotation around a single axis can encompass all rotations around intersecting axes.

A unit quaternion is used to describe orientation in this paper. Its parameters consist of a scalar and a three-dimensional vector. The following equation represents the relationship between the unit quaternion and the axis/angle parameters:

$$\eta = \cos(\theta/2), \qquad \boldsymbol{\varepsilon} = \mathbf{u}\sin(\theta/2), \tag{4}$$

where θ s a real number representing the rotation angle, and **u** is a unit vector indicating the direction of the rotation axis.

A quaternion $\boldsymbol{\xi}$ consists of a pair comprising a real part η and a vector part $\varepsilon = \varepsilon_1 i + \varepsilon_2 j + \varepsilon_3 k$:

$$\boldsymbol{\xi} = \eta + \boldsymbol{\varepsilon} = \eta + \varepsilon_1 i + \varepsilon_2 j + \varepsilon_3 k. \tag{5}$$

For a unit quaternion, the following constraint must hold:

$$\eta^2 + \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = \eta^2 + \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 = 1. \tag{6}$$

With the Euler parameters, $\boldsymbol{\xi} = [\eta, \ \boldsymbol{\varepsilon}^T]^T$, the rotation matrix is determined by the following equation 0:

$$\begin{aligned}\mathbf{R}(\eta,\boldsymbol{\varepsilon}) &= (\eta^2 - \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon})\mathbf{I}_3 + 2\eta S(\boldsymbol{\varepsilon}) + \boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T \\ &= \mathbf{E}(\eta,\boldsymbol{\varepsilon})\mathbf{G}^T(\eta,\boldsymbol{\varepsilon})\end{aligned} \tag{7}$$

with

$$\begin{aligned}\mathbf{E}(\boldsymbol{\xi}) &= \mathbf{E}(\eta,\boldsymbol{\varepsilon}) \\ &= \begin{bmatrix} -\boldsymbol{\varepsilon} & \eta\mathbf{I}_3 + S(\boldsymbol{\varepsilon}) \end{bmatrix} \in R^{3\times4}\end{aligned} \tag{8}$$

and

$$\begin{aligned}\mathbf{G}(\boldsymbol{\xi}) &= \mathbf{G}(\eta,\boldsymbol{\varepsilon}) \\ &= \begin{bmatrix} -\boldsymbol{\varepsilon} & \eta\mathbf{I}_3 - \mathbf{S}(\boldsymbol{\varepsilon}) \end{bmatrix} \in R^{3\times4},\end{aligned}$$

where $\mathbf{S}(\boldsymbol{\varepsilon})$ is the skew-symmetric matrix corresponding to the three-element vector $\boldsymbol{\varepsilon}$:

$$\mathbf{S}(\boldsymbol{\varepsilon}) = \tilde{\boldsymbol{\varepsilon}} = \begin{bmatrix} 0 & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & 0 & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & 0 \end{bmatrix}.$$

The detailed expression of the direction cosine matrix is written as follows:

$$\begin{aligned}&\mathbf{R}(\eta,\boldsymbol{\varepsilon}) \\ &= \begin{bmatrix} 2(\eta^2+\varepsilon_1^2)-1 & 2(\varepsilon_1\varepsilon_2-\eta\varepsilon_3) & 2(\varepsilon_1\varepsilon_3+\eta\varepsilon_2) \\ 2(\varepsilon_1\varepsilon_2+\eta\varepsilon_3) & 2(\eta^2+\varepsilon_2^2)-1 & 2(\varepsilon_2\varepsilon_3-\eta\varepsilon_1) \\ 2(\varepsilon_1\varepsilon_3-\eta\varepsilon_2) & 2(\varepsilon_2\varepsilon_3+\eta\varepsilon_1) & 2(\eta^2+\varepsilon_3^2)-1 \end{bmatrix} \\ &= \mathbf{R}(\xi)\end{aligned}$$

From (7) it is noted that $\mathbf{R}(\eta,\boldsymbol{\varepsilon}) = \mathbf{R}(-\eta,-\boldsymbol{\varepsilon})$, meaning that each direction cosine matrix maps to two unit quaternions:

$$\mathbf{R} \quad \Leftrightarrow \quad \pm\boldsymbol{\xi}.$$

If we wants to compute the unit quaternion corresponding to a given rotation matrix $\mathbf{R} = \{r_{ij}\}$, one of the solutions to this inverse problem is given as follows:

$$\eta = \frac{1}{2}\sqrt{r_{11}+r_{22}+r_{33}+1},$$

$$\boldsymbol{\varepsilon} = \frac{1}{2}\begin{bmatrix} \operatorname{sgn}(r_{32}-r_{23})\sqrt{r_{11}-r_{22}-r_{33}+1} \\ \operatorname{sgn}(r_{13}-r_{31})\sqrt{r_{22}-r_{33}-r_{11}+1} \\ \operatorname{sgn}(r_{21}-r_{12})\sqrt{r_{33}-r_{11}-r_{22}+1} \end{bmatrix}. \tag{9}$$

Where $\operatorname{sgn}(x) = 1$ with $x \geq 0$ and $\operatorname{sgn}(x) = -1$ with $x < 0$. In (9), we set $\eta \geq 0$, so that it corresponds to the rotation angle $\theta \in [-\pi,\pi]$, ensuring that all possible rotation angles can be described. Moreover, unlike the case of computing the rotation angle and axis from the direction cosine matrix, no singularities occur in (9).

In the case where the orientation of the rigid body changes over time, we need to establish the relationship between the time derivative of the Euler parameters $\dot{\boldsymbol{\xi}} = [\dot{\eta}, \dot{\boldsymbol{\varepsilon}}^T]^T$ and the angular velocity of the rigid body in the fixed coordinate system $\boldsymbol{\omega}^{(0)} \in R^3$. This relationship is given by the quaternion propagation as follows:

$$\dot{\boldsymbol{\xi}} = \frac{1}{2}\mathbf{E}(\boldsymbol{\xi})^T \boldsymbol{\omega}^{(0)} \tag{10}$$

where $\boldsymbol{\omega}^{(0)}$ is the angular velocity vector of the object expressed in the fixed coordinate system.

It can be shown that $\mathbf{E}(\boldsymbol{\xi})\mathbf{E}(\boldsymbol{\xi})^T = \mathbf{I}_3$ [5]. The rows of the matrix $\mathbf{E}(\boldsymbol{\xi})$ are mutually orthogonal and also orthogonal to $\boldsymbol{\xi}$, hence $\mathbf{E}(\boldsymbol{\xi})\boldsymbol{\xi} = \mathbf{0}$, and can be solved from (10) as:

$$\boldsymbol{\omega}^{(0)} = 2\mathbf{E}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}}\,. \tag{11}$$

3. Orientation Motion Planning

3.1. Point-to-Point Orientation Motion Planning

The minimum requirement for an object is the ability to move from the initial pose {$\mathbf{p}_0$, $\mathbf{R}_0$} or {$\mathbf{p}_0$, ξ_0} to the specified final pose {$\mathbf{p}_f$, $\mathbf{R}_f$} or {$\mathbf{p}_f$, ξ_f}. The movement process must be smooth. Therefore, trajectory planning algorithms must be developed to generate smooth and appropriate trajectories [3]. Now, consider the possibility of finding a path between a given set of points in the workspace using interpolation curves:

$$\boldsymbol{\xi}(t) = f(t, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f)$$

to ensure its norm is equal to 1.

The simplest solution is to use spherical linear interpolation, where the interpolation function is chosen as follows [12]:

$$\boldsymbol{\xi}(t) = \frac{\boldsymbol{\xi}_0 \sin[(1-t)\theta] + \boldsymbol{\xi}_f \sin[t\theta]}{\sin\theta}, \tag{12}$$

with $0 \le t \le 1$ and the acute angle θ between two quaternion corresponding to two Euler parameters.

The above formula ensures that the Euler parameters vary continuously from the initial value to the final value, and ensures that its norm is equal to the unit. However, it is not possible to impose its derivative to match the desired angular velocity at the initial and final times.

The interpolation scheme proposed in this paper is as follows:

$$\boldsymbol{\xi}(t) = \frac{\boldsymbol{\xi}_0 + s(t)(\boldsymbol{\xi}_f - \boldsymbol{\xi}_0)}{\|\, \boldsymbol{\xi}_0 + s(t)(\boldsymbol{\xi}_f - \boldsymbol{\xi}_0) \,\|} \tag{13}$$

With $s(t)$ is a polynomial of order 3, 5,…. or any other continuously smooth function that satisfies

$$\begin{aligned} s(0) &= 0, \qquad \dot{s}(0) = 0, \\ s(t_f) &= 1, \qquad \dot{s}(t_f) = 0. \end{aligned} \tag{14}$$

Let $\mathbf{z}(t) = \boldsymbol{\xi}_0 + s(t)(\boldsymbol{\xi}_f - \boldsymbol{\xi}_0)$, we have:

$$\dot{\mathbf{z}}(t) = \dot{s}(t)(\boldsymbol{\xi}_f - \boldsymbol{\xi}_0)\,.$$

Let

$$v(t) = \left\| \boldsymbol{\xi}_0 + s(t)(\boldsymbol{\xi}_f - \boldsymbol{\xi}_0) \right\| = \sqrt{\mathbf{z}(t)^T \mathbf{z}(t)}\,,$$

we can calculate:

$$\dot{v}(t) = \frac{\mathbf{z}(t)^T \dot{\mathbf{z}}(t)}{\sqrt{\mathbf{z}(t)^T \mathbf{z}(t)}}\,.$$

From this, the Euler parameters and their time derivatives can be determined as follows:

$$\boldsymbol{\xi}(t) = \frac{\mathbf{z}(t)}{v(t)} \tag{15}$$

and

$$\dot{\boldsymbol{\xi}}(t) = \frac{\dot{\mathbf{z}}(t)v(t) - \mathbf{z}(t)\dot{v}(t)}{v(t)^2} \tag{16}$$

Thus, with a smooth function $s(t)$ satisfying (14), we can construct the orientation trajectory for the object using Euler parameters. The smooth function $s(t)$ can be cubic polynomials, higher-order polynomials, or even harmonic functions and cycloid paths,…

3.2. Orientation Motion Planning through Intermediate Poses

Problem Statement: Given a set of orientations to be passed through:

$$\{\mathbf{R}_0, \mathbf{R}_1, \cdots, \mathbf{R}_N\} \text{ or } \{\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_N\}$$

corresponding to specific time instances:

$$\{0 = t_0 < t_1 < \cdots < t_N = t_f\}$$

with initial and final angular velocities set to zero:

$$\boldsymbol{\omega}(t_0) = \boldsymbol{\omega}(t_f) = \mathbf{0}\,.$$

It is necessary to interpolate and construct continuous functions $\boldsymbol{\xi}(t)$, $0 \le t \le t_f$ such that:

$$\boldsymbol{\xi}(t_k) = \boldsymbol{\xi}_k, \quad k = 0, 1, \ldots, N\,.$$

First, we construct a spline curve passing through a set of $N+1$ control points q_i and the corresponding time sequence:

$$0 = t_0 < t_1 < ..... < t_N < t_N = t_f,$$
$$q_i = q_0, q_1, \quad ..., \quad q_{N-1}, q_N = q_f.$$

A cubic spline fuction $S(t)$ to be determined is a piecewise continuous function that sastisfies the folowing conditions:

1. $S(t) = S_k(t)$ is a cubic polynomial on each segment $t \in [t_k, t_{k+1}]$ with $k = 0,1,2,...,N-1$.
2. $S(t_k) = q_k$, $k = 0,1,2,...,N.$
3. $S(t), \dot{S}(t) \,\&\, \ddot{S}(t)$ is continuous over the inteval $t \in [t_0, t_{N+1}] = [0, t_f]$, $S(t)$ is continuous to the second order derivative.

We write the N segments of the cubic spline $S(t)$ function as follows:

$$S_k(t) = a_k + b_k(t - t_k) + c_k(t - t_k)^2 + d_k(t - t_k)^3,$$
$$t \in [t_k, t_{k+1}], \; k = 0,1,2,...,N-1,$$

where $4N$ coefficients a_k, b_k, c_k, d_k need to be determined based on the continuity conditions of the spline at the control points.

From the continuity conditions at the control points, we obtain:

$$S_k(t_k) = q_k, \qquad k = 0,1,2,...,N-1,$$
$$S_k(t_{k+1}) = q_{k+1}, \qquad k = 0,1,2,...,N-1,$$

along with the continuity conditions for the first and second derivatives:

$$\dot{S}_k(t_{k+1}) = \dot{S}_{k+1}(t_{k+1}), \qquad k = 0,1,2,...,N-2,$$
$$\ddot{S}_k(t_{k+1}) = \ddot{S}_{k+1}(t_{k+1}), \qquad k = 0,1,2,...,N-2.$$

These continuity conditions provide a total of $4N-2$ equations. However, to fully define the system, two additional equations are needed. Here, a clamped spline is used, meaning the two additional equations are:

$$\dot{S}_0(t_0) = 0, \qquad \dot{S}_{N-1}(t_N) = 0$$

The determination of the polynomial coefficients $S_k(t)$ is detailed in reference 0.

Applying the above interpolation method to the four Euler parameters, we obtain four corresponding spline functions: $[\xi_0(t), \xi_1(t), \xi_2(t), \xi_3(t)]$. These functions are continuous up to the second derivative and ensure that they pass through the control points.

However, at time instances outside the control points, the sum of the squares of these four parameters may no longer be equal to one. Therefore, normalization is required using the following formula:

$$\varepsilon_i(t) = \frac{\xi_i(t)}{\sqrt{\xi_0^2(t) + \xi_1^2(t) + \xi_2^2(t) + \xi_3^2(t)}} = \frac{\xi_i(t)}{\sqrt{Z}},$$
$$i = 0,1,2,3$$

And we obtain the derivative of Euler parameters:

$$\dot{\varepsilon}_i(t) = \frac{\dot{\xi}_i}{\sqrt{Z}} - \frac{\xi_i(t)[\xi_0\dot{\xi}_0 + \xi_1\dot{\xi}_1 + \xi_2\dot{\xi}_2 + \xi_3\dot{\xi}_3]}{Z\sqrt{Z}}.$$

4. Numerical Simulation

In this section, two numerical simulations are performed. The first case involves point-to-point interpolation, while the second case considers orientation interpolation through a set of intermediate orientations.

Interpolation of direction change from start direction to end direction

The position and orientation at the initial and final points are given by:

$\mathbf{r}_0$ = [0.50; 0.00; 0.00]; % m

$\mathbf{R}_0$ = [-0.4330 0.2746 0.8585
-0.2500 0.8785 -0.4071
-0.8660 -0.3909 -0.3117]

$\boldsymbol{\xi}_0$ =[0.5324 0.0076 0.8098 -0.2464]^T

$\mathbf{r}_1$ = [0.00; 0.50; 0.50]; % m

$\mathbf{R}_1$ = [-0.4330 0.8716 -0.2298
0.2500 0.3611 0.8984
0.8660 0.3316 -0.3743]

$\boldsymbol{\xi}_1$ =[0.3721 -0.3808 -0.7363 -0.4176]^T

Orientation motion planning is carried out in a time interval T = 2s, cubic polynomial is used:

$$s(t) = \begin{cases} \dfrac{3}{T^2}t^2 - \dfrac{2}{T^3}t^3, & 0 \le t \le T \\ 1, & T < t \end{cases}$$

The interpolation results are presented in Fig. 1 to Fig. 5. The Euler parameters and their derivatives vary smoothly over time, Fig. 1 and Fig. 2. The angular velocity increases from 0, reaches its maximum value at t = 1 s, and then decreases back to 0 at t = 2 s, Fig. 3. The object's orientation, represented by a moving frame, is shown as it transitions along a straight-line path and an arc connecting the initial and final points, Fig. 4 and Fig. 5.

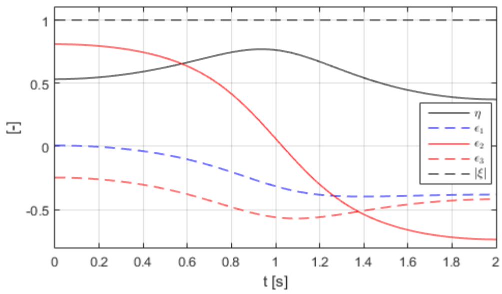

Fig. 1. Euler parameters over time

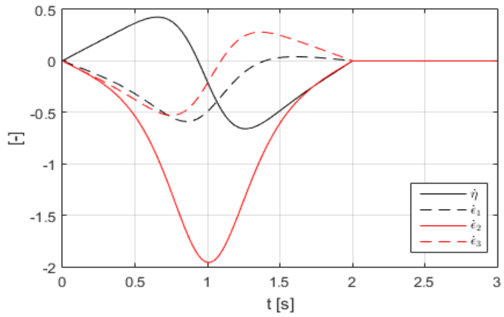

Fig. 2. Derivatives of Euler parameters over time

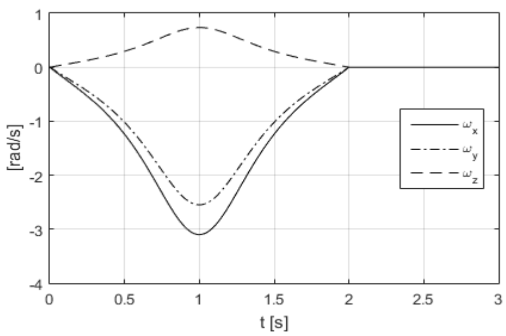

Fig. 3. Angular velocity in the fixed frame

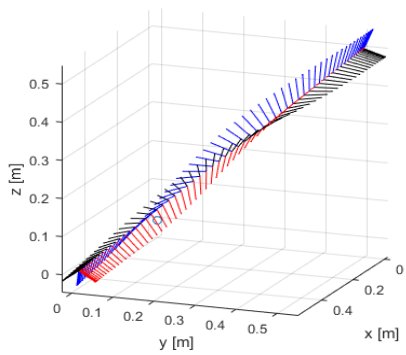

Fig. 4. Orientation change along the straight-line trajectory

From these results, it can be concluded that the proposed Euler parameter interpolation method is applicable for orientation motion planning of objects in space.

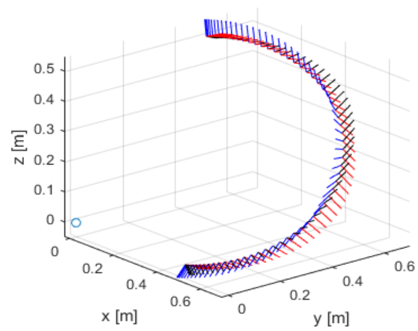

Fig. 5. Orientation change along the circular trajectory

Interpolation of orientation through a set of intermediate orientations

In this simulation, the object moves along a helical path:

$$\mathbf{r}(t) = [0.5\cos\omega t,\ 0.5\sin\omega t,\ 0.2t]^T,\ \mathrm{m}\,.$$

while its orientation must pass through the following intermediate orientations defined by Euler parameters:

$\xi_0 = [0.7071\ \ 0\ \ \ \ 0.6124\ \ -0.3536]^T$

$\xi_1 = [0.8457\ \ -0.0363\ \ 0.3141\ \ -0.4300]^T$

$\xi_2 = [0.9808\ \ 0\ \ \ \ 0.1951\ \ 0\ ]^T$

$\xi_3 = [0.8140\ \ 0.2614\ \ -0.0700\ \ 0.5139]^T$

$\xi_4 = [0.4305\ \ 0.5610\ \ 0.0923\ \ 0.7011]^T$

corresponding to the time instances:

$T = [0\ \ 0.50\ \ \ 1.00\ \ 1.50\ \ 2.00]\mathrm{s}$

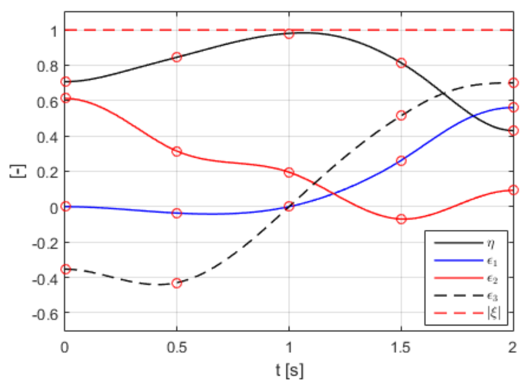

Fig. 6. Euler parameters over time passing through the control points (circular markers)

The simulation results for orientation interpolation through control points are shown in Fig. 6 to Fig. 9. Fig. 6 shows that four Euler parameters smoothly transition through the intermediate orientations while maintaining unit norm throughout the interpolation process. The time derivatives of the Euler parameters and the angular velocity of the object are also smooth functions, ensuring that the initial and final values are 0, Fig. 7 and Fig. 8. On Fig. 9, the object's orientation, represented by a moving coordinate frame, is visualized as it follows the helical trajectory. Fig. 1 and Fig. 6 also show the norm of Euler parameters being unit during the motion time.

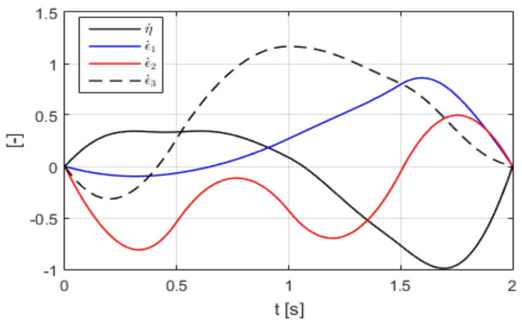

Fig. 7. Time derivatives of Euler parameters

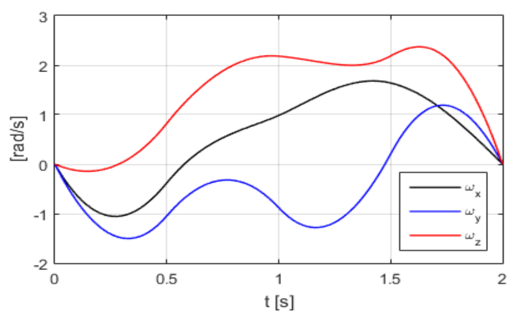

Fig. 8. Components of the object's angular velocity in the fixed coordinate system

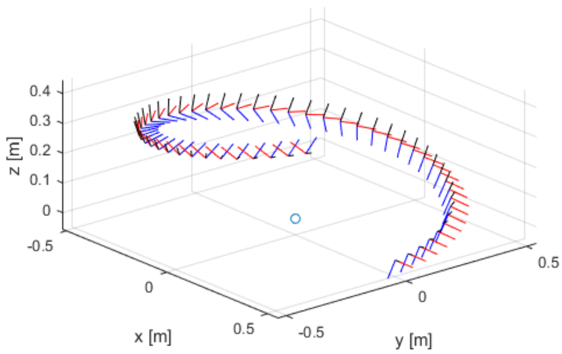

Fig. 9. Orientation change along the helical trajectory

5. Conclusion

This paper presents the orientation motion planning of a spatial rigid body using Euler parameters. The Euler parameters are interpolated using cubic spline functions, ensuring smoothness up to the second order derivative while passing through intermediate poses. The unit norm condition of the Euler parameters is maintained by normalizing the spline functions. Using Euler parameters to describe the orientation of the rigid body avoids the kinematic singularities associated with Euler angles. The effectiveness of the proposed method is demonstrated through numerical simulations. The algorithms presented in this paper provide interpolation tools for orientation planning that are accurate, easy to implement. The interpolation method proposed in this paper fully satisfies the motion planning problem in which the object is required to move through given intermediate orientations. The proposed approach will be applied to end-effector motion planning for robotic manipulators in future research.

Acknowledgment

This work was supported in part by the National Program: Support for research, development, and technology application of industry 4.0 (KC-4.0/19-25), under the grant for the project: Research, design and manufacture of Cobot applied in industry and some other fields with human-robot interaction (code: KC-4.0-35/19-35).

References

[1] Nguyen Van Khang, and Chu Anh My, Fundamentals of Industrial Robots, Ha Noi, Vietnam: Education Publishing House, 2023. (In Vietnamese: Nguyễn Văn Khang; Chu Anh Mỹ: Cơ sở robot công nghiệp, Nhà xuất bản Giáo dục Việt Nam, 2023).

[2] Nguyen Quang Hoang, Foundations of Robotics - Mechanics and Control, Bachkhoa Publishing House, 2023. (In Vietnamese: Nguyễn Quang Hoàng: Cơ sở robotics - Cơ học và Điều khiển, Nhà xuất bản Bách khoa Hà Nội, 2023).

[3] Luigi Biagiotti, and Claudio Melchiorri, Trajectory Planning for Automatic Machines and Robots, Berlin, Germany: Springer Science & Business Media, 2008. https://doi.org/10.1007/978-3-540-85629-0

[4] Ricardo Campa, Karla Camarillo, and Lina Arias, Kinematic modeling and control of robot manipulators via unit quaternions in application to a spherical wrist, Proceedings of the 45th IEEE Conference on Decision & Control, San Diego, CA, USA, Dec. 13-15, 2006. https://doi.org/10.1109/CDC.2006.377155

[5] Nguyen Van Khang, Dynamics of Multi-body Systems, Ha Noi, Vietnam: Science and Technology Publishing House, 2017. (In Vietnamese: Nguyễn Văn Khang: Động lực học hệ nhiều vật, Nhà xuất bản Khoa học và Kỹ thuật, 2017).

[6] John J Craig, Introduction to Robotics: Mechanics and Control, UK: Pearson/Prentice Hall, 2005.

[7] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo, Robotics: Modelling, Planning and Control, London, UK: Springer Science & Business Media, 2010. https://doi.org/10.1007/978-1-84628-642-1

[8] Spong M. W., Hutchinson S., and Vidyasagar M., Robot Modeling and Control, New York, USA: John Wiley & Sons, 2006.

[9] O. A. Bauchau, S. Han, Interpolation of rotation and motion, Multibody System Dynamics, vol. 31, pp. 339-370, Mar. 2014. https://doi.org/10.1007/s11044-013-9365-8

[10] G. Kang and F. C. Park, Cubic spline algorithms for orientation interpolation, International Journal Numerical Methods in Engineering, vol. 46, iss. 1, pp. 45-64, Jul. 1999.

[11] G. M. Nielson, ν-Quaternion splines for the smooth interpolation of orientations, in IEEE Transactions on Visualization and Computer Graphics, vol. 10, no. 2, pp. 224-229, Mar - Apr. 2004. https://doi.org/10.1109/TVCG.2004.1260774

[12] David Eberly: Quaternion Algebra and Calculus, Geometric Tools, Redmond WA 98052, Last Modified: Aug. 18, 2010.

[13] Matthias Neubauer and Andreas Müller, Smooth orientation path planning with quaternions using B-splines, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.

[14] A. H. Barr, B. Currin, S. Gabriel, J. F. Hughes, Smooth interpolation of orientations with angular velocity constraints using quaternions, in SIGGRAPH '92 proceedings of the 19th annual conference on computer graphics and interactive techniques, New York, pp. 313, Jul. 1992. https://doi.org/10.1145/133994.134086

[15] Andrea Tagliavini and Corrado Guarino Lo Bianco, A smooth orientation planner for trajectories in the cartesian space, in IEEE Robotics and Automation Letters, vol. 8, iss. 5, pp. 2606-2613, May 2023. https://doi.org/10.1109/LRA.2023.3256921