

# Gradient-based attitude planning for rigid spacecraft on SO(3)

Tao HUO<sup>1</sup>, Jin YANG<sup>2</sup>, Chengfei YUE<sup>3,4\*</sup>, Xueqin CHEN<sup>1,3</sup> & Xibin CAO<sup>1,3</sup>

<sup>1</sup> School of Astronautics, Harbin Institute of Technology, Harbin 150001, China

<sup>2</sup> School of Management, Harbin Institute of Technology, Harbin 150001, China

<sup>3</sup> State Key Laboratory of Micro-Spacecraft Rapid Design and Intelligent Cluster, Harbin Institute of Technology, Harbin 150001, China

<sup>4</sup> School of Aerospace Science, Harbin Institute of Technology Shenzhen, Shenzhen 518055, China

Received August 18, 2024; accepted January 14, 2025; published online April 7, 2025

**Abstract** Attitude planning of rigid bodies has many applications in robotics and aerospace. However, because the attitude configuration space is non-Euclidean and the constraints are complex and non-linear, the design of the attitude curve has always been a tricky problem. In this paper, a gradient-based attitude planning method is proposed to simultaneously handle attitude pointing, angular velocity, torque, and time constraints on Lie group SO(3). Firstly, the attitude interpolation algorithm on SO(3) gives an attitude curve connecting the initial and target attitudes. The shape of the curve is determined by the fitting coefficients and maneuvering time. Secondly, to match the curve with suitable angular velocity and control torque, a nonlinear planning model with fitting coefficients and maneuver time as decision variables is proposed. Solving the problem gives a smooth attitude curve that satisfies both kinematic and dynamic constraints and also avoids complicated time allocation. Then, to apply the gradient-based solver, analytical formulas for the derivatives of each order of the attitude curve with respect to the decision variables are given in this paper. Finally, the effectiveness of the proposed algorithm is verified by a series of numerical simulations.

**Keywords** attitude planning, attitude constraints, gradient optimization, three-dimensional special orthogonal group (SO(3))

**Citation:** Huo T, Yang J, Yue C F, et al. Gradient-based attitude planning for rigid spacecraft on SO(3). *Sci China Tech Sci*, 2025, 68(5): 1520401, <https://doi.org/10.1007/s11431-024-2883-1>

## 1 Introduction

Motion planning is the basis of rigid body control, and such planning is generally performed under complex constraints [1–3]. Taking the attitude motion of a spacecraft as an example, kinematic constraints, dynamics constraints, and time constraints often need to be considered in attitude planning [4, 5]. Kinematic constraints include two types. The first one is forbidden/mandatory constraints for attitude, such as infrared telescopes needing to avoid bright celestial objects and antennas needing to be always aligned with ground stations in communication relay missions [6]. The second type requires the angular velocity not to exceed the upper limit value during the motion [7]. The dynamics constraint means an upper limit on the control torque considering that the attitude actuator cannot provide an infinite control torque [8–10]. The

time constraint means that the spacecraft should complete the attitude maneuver within a given time [11].

To perform motion planning under complex constraints, geometric methods [12, 13], potential function methods [7, 14, 15], search-based methods [16–18] and optimization-based methods [19–22] have been proposed. Since attitude constraints and pointing trajectories are represented on the unit sphere, geometric methods design attitude paths based on intuitive representation. The basic idea of the potential function approach is to construct an artificial potential function, which takes a larger value when the attitude violates the constraints and the lowest value at the desired attitude, and then apply control along the inverse of the gradient of the potential function to achieve attitude maneuver [4]. The common advantages of the above two methods are simple algorithm structure and high computational efficiency. However, the performance of these two methods in handling multi-

\* Corresponding author (email: [yuechengfei@hit.edu.cn](mailto:yuechengfei@hit.edu.cn))

ple constraints such as attitude, angular velocity, and control torque simultaneously is unsatisfactory. The potential function method may suffer from the problem of unpredictable paths, which are prone to produce large control torque when the attitude is close to the constraint zones, exceeding the upper limit of the output of the actuators.

Search-based approaches attempt to build graph structures in the attitude configuration space and then use search algorithms such as A\* to give an attitude path at the kinematic level. For example, ref. [17] used a hexagon to discretize the unit sphere directly, search for a feasible path satisfying the attitude constraints, and invert the attitude curve of the spacecraft. Ref. [16] started directly in the attitude configuration space SO(3) and divided all the attitudes into a finite number of subsets of SO(3) by constructing sampling cells. The feasible path is then searched on these subsets. However, search-based algorithms split kinematics and dynamics, and an interpolation algorithm is additionally needed, for instance in ref. [16], to generate an attitude-time curve for control.

Optimization-based methods formulate the attitude curve design problem as an optimization problem with constraints for a specific metric and solve the problem numerically to obtain a solution that satisfies the attitude constraints. The square of the coefficients is used as the objective function in ref. [20], and the second-order derivatives of the quaternions are used as the objective function in ref. [22]. These objective functions have no obvious physical significance. Besides, it cannot guarantee a reduction in control consumption or maneuver time. Furthermore, the planning schemes proposed in refs. [16, 20, 22] also inevitably involve inverse dynamics, since they deal with kinematic and dynamical constraints separately. To satisfy angular velocity, torque, and maneuver time constraints, interpolation of the attitude nodes obtained from planning and careful design of time allocation algorithms are required in these approaches [23]. In addition, with the wide application of intelligent algorithms in a variety of industrial fields [24–26], there have been studies applying such intelligent algorithms to solve attitude planning problems [27, 28]. For example, ref. [27] used particle swarm optimization to find a time-optimal attitude maneuver path under pointing constraints. However, the weaknesses are excessive computational resource requirements and difficulty in constructing samples [4].

Based on the above observations, continuous attitude curve design methods that satisfy all constraints still need to be developed. Among those attitude planning methods that are also optimization-based, there are limitations such as the physical meaning of the optimization objective function is unclear and the gradient formula is not given analytically. Therefore, in this paper, a gradient-based attitude planning algorithm is proposed to handle kinematics, dynamics, and maneuver time constraints simultaneously. First, we design

a smooth curve connecting the initial attitude and the target attitude directly on the attitude configuration space SO(3) based on the attitude interpolation algorithm in refs. [29, 30]. Then, using the fitting coefficients of the curve and the maneuvering time as decision variables, we establish a nonlinear planning problem and compute the derivatives of each order state (i.e., attitude, angular velocity, and torque) analytically of the designed curve. Finally, the gradient optimization algorithm is applied to perform simulations under different fitting basis functions, different objective functions, and different constraint types to verify the effectiveness of the proposed method. The main contributions of this paper are summarized as follows.

(i) Mathematical formulas to exactly calculate the derivative of first and second states relative to the fitting coefficients and the maneuvering time are established, which matches the inherent property of the attitude dynamics where the spacecraft's attitude is driven by the second-order states, i.e., the control torque.

(ii) The proposed attitude planning method considers both kinematic and dynamical constraints on real-time intervals, which differs from refs. [20, 22] in that the objective function of our method has a clear physical meaning and avoids designing additional time allocation algorithms.

## 2 Problem statement

### 2.1 Preliminary

$\mathbb{R}$  denotes the set of real numbers.  $\mathbf{I} \in \mathbb{R}^{3 \times 3}$  denotes identity matrix.  $\mathbf{e}_i$  and  $S^2$  denote the standard orthogonal basis and the unit sphere in  $\mathbb{R}^3$ .  $F_I$  and  $F_B$  denote the inertial frame and the body frame, respectively. The attitude of a spacecraft is the rotation matrix of  $F_I$  concerning  $F_B$ .  $\|\mathbf{x}\|$  and  $X_i$  denote the 2-norm and the  $i$ -th component of the vector  $\mathbf{x}$ , respectively. Both  $\widehat{\mathbf{x}}$  and  $(\mathbf{x})^\wedge$  denote the cross product matrix of the vector  $\mathbf{x} \in \mathbb{R}^3$ .  $(\cdot)^\vee$  is inverse map of  $(\cdot)^\wedge$ , that is,  $(\widehat{\mathbf{x}})^\vee = \mathbf{x}$ .  $(\mathbf{A})^T$  is transpose of the matrix  $\mathbf{A}$ . All rotation matrices form the Lie group  $\text{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^T \mathbf{R} = \mathbf{R} \mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1\}$  in the sense of matrix multiplication. Given the axis of rotation  $\mathbf{w} \in S^2$  and the angle of rotation  $\theta$ , the corresponding rotation matrix  $\mathbf{R}(\theta, \mathbf{w})$  can be computed from the exponential map, i.e.,  $\mathbf{R}(\theta, \mathbf{w}) := \exp(\theta \widehat{\mathbf{w}}) = \mathbf{I} + \sin \theta \widehat{\mathbf{w}} + (1 - \cos \theta) \widehat{\mathbf{w}}^2$ , where  $\widehat{\mathbf{w}}$  is the cross product matrix of  $\mathbf{w}$ . Any given rotation matrix  $\mathbf{R}$  can be represented in the form of  $\exp(\theta \widehat{\mathbf{w}})$ . The inverse of the exponential map  $\exp(\cdot)$  is denoted by  $\log(\cdot)$ , i.e.,  $\log(\mathbf{R}(\theta, \mathbf{w})) = \theta \widehat{\mathbf{w}}$ , angular distance  $\theta = \arccos((\text{tr}(\mathbf{R}) - 1)/2)$  and unit vector  $\widehat{\mathbf{w}} = (\mathbf{R} - \mathbf{R}^T)/(2 \sin \theta)$ . Given  $\mathbf{R}(\theta, \mathbf{w}) = \exp(\theta \widehat{\mathbf{w}}) \in \text{SO}(3)$  and  $\mathbf{y} \in \mathbb{R}^3$ , then  $\exp(\theta \widehat{\mathbf{w}}) \mathbf{w} = \mathbf{w}$  and  $\mathbf{R} \mathbf{y} \mathbf{R}^T = \widehat{\mathbf{R} \mathbf{y}}$  [31]. Some lemmas used in this paper are listed below.

**Lemma 1** ([32]) The partial derivative of the rotation matrix  $\exp(\widehat{\mathbf{x}})$  with respect to the exponential coordinate  $X_i$

is

$$\frac{\partial}{\partial x_i} \exp(\bar{\mathbf{x}}) = \frac{x_i \bar{\mathbf{x}} + (\mathbf{x} \times (\mathbf{I} - \exp(\bar{\mathbf{x}})) \mathbf{e}_i)^\wedge}{\|\mathbf{x}\|^2} \exp(\bar{\mathbf{x}}). \quad (1)$$

**Lemma 2** ([33]) Let  $\varepsilon$  denote a small positive constant,  $t$  denote time and  $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]^T \in \mathbb{R}^3$  denote a differentiable curve. Let  $\alpha(\mathbf{x}) = \sin(\|\mathbf{x}\|) / \|\mathbf{x}\|$ ,  $\beta(\mathbf{x}) = 2(1 - \cos(\|\mathbf{x}\|)) / \|\mathbf{x}\|^2$ , and  $\dot{\mathbf{x}}(t) = \frac{d}{dt} \mathbf{x}(t)$ . Then the derivative of  $\mathbf{R}(t) = \exp(\bar{\mathbf{x}}(t))$  with respect to time  $t$  is

$$\begin{aligned} \dot{\mathbf{R}}(t) &= \mathbf{R}(t)(\mathbf{D}(\mathbf{x}) \dot{\mathbf{x}})^\wedge \\ &= \sum_{i=1}^3 \frac{\partial}{\partial x_i} \exp(\bar{\mathbf{x}}) \dot{x}_i, \end{aligned} \quad (2)$$

where tangent operator  $\mathbf{D}(\mathbf{x})$  is

$$\mathbf{D}(\mathbf{x}) = \begin{cases} \mathbf{I} - \frac{\beta(\mathbf{x})}{2} \bar{\mathbf{x}} + \frac{1 - \alpha(\mathbf{x})}{\|\mathbf{x}\|^2} \bar{\mathbf{x}}^2, & \|\mathbf{x}\| \geq \varepsilon, \\ \mathbf{I}, & \|\mathbf{x}\| < \varepsilon. \end{cases} \quad (3)$$

**Remark 1** Eq. (3) is the closed-form expression of the left-trivialized differential of the exponential map on  $\text{SO}(3)$  [33, 34]. Since there is a singularity in the exponential coordinate near  $\mathbf{0}$ ,  $\mathbf{D}(\mathbf{x})$  is taken to be a segmented expression. In subsequent calculations, for convenience, the first-order derivative of  $\mathbf{R}(t)$  is calculated using eq. (1) and the higher-order derivative is calculated using eq. (2).

## 2.2 Constraints during attitude maneuver

### 2.2.1 Attitude constraints

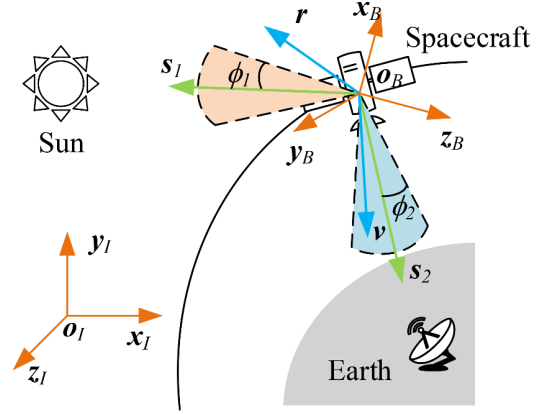
Given an initial attitude  $\mathbf{R}_0$  and a desired attitude  $\mathbf{R}_f$ , the rigid spacecraft's trajectory  $\mathbf{R}(t), t \in [0, T], T \leq t_f$  given by attitude planning is first satisfied

$$\begin{cases} \mathbf{R}(0) = \mathbf{R}_0, \\ \mathbf{R}(T) = \mathbf{R}_f. \end{cases} \quad (4)$$

As shown in Figure 1, there are forbidden pointing constraints and mandatory point constraints during attitude maneuvering. The former requires that the angle between a particular pointing on the spacecraft and some given direction is always greater than a given value. Assume that the unit vector  $\mathbf{r}$  is solidly attached to the spacecraft. The number of forbidden zones is  $q$  and the constraint angle for each forbidden pointing is  $\phi_{1,k} < 0.5\pi, k = 1, \dots, q$ . Such attitude constraints can be expressed as

$$\mathbf{s}_{1,k}^T \mathbf{R}(t) \mathbf{r} < \cos \phi_{1,k}. \quad (5)$$

Suppose that another unit vector  $\mathbf{v}$  needs to satisfy a mandatory pointing constraint, which means that the angle



**Figure 1** (Color online) Attitude pointing constraints.

between  $\mathbf{v}$  and a particular vector  $\mathbf{s}_2$  needs to always be less than a given value  $\alpha_2 < 0.5\pi$ , i.e.,

$$\mathbf{s}_2^T \mathbf{R}(t) \mathbf{v} > \cos \phi_2. \quad (6)$$

### 2.2.2 Angular velocity and torque constraints

The equations of kinematics and dynamics of the spacecraft without considering the disturbance are

$$\dot{\mathbf{R}} = \mathbf{R} \bar{\boldsymbol{\omega}}, \quad (7)$$

$$\mathbf{J} \dot{\boldsymbol{\omega}} + \bar{\boldsymbol{\omega}} \mathbf{J} \boldsymbol{\omega} = \mathbf{u}. \quad (8)$$

During the actual attitude maneuver, the upper limit to the maneuver time  $T$  is

$$0 < T \leq t_f. \quad (9)$$

Similarly, there are upper limit constraints on angular velocity and control torque:

$$\begin{cases} \|\boldsymbol{\omega}_i(t)\| \leq \omega_{\max}, \\ \|\mathbf{u}_i(t)\| \leq u_{\max}. \end{cases} \quad (10)$$

At  $t = 0$  and  $t = T$ , the angular velocity and torque are bounded by the equations:

$$\begin{cases} \boldsymbol{\omega}(0) = \boldsymbol{\omega}(T) = \mathbf{0}, \\ \mathbf{u}(0) = \mathbf{u}(T) = \mathbf{0}. \end{cases} \quad (11)$$

Attitude planning aims to generate an attitude curve  $\mathbf{R}(t)$  that satisfies the aforementioned constraints according to a specified metric  $F(\mathbf{R}(t), T)$ . Therefore, the general form of the attitude planning problem can be expressed as

$$\begin{aligned} \text{PO: } \min_{\{\mathbf{R}(t), T\}} & F(\mathbf{R}(t), T) \\ \text{s.t. } & \text{eqs. (4)–(11)}. \end{aligned} \quad (12)$$

### 3 Main results

As shown in Figure 2, in the previous section, we summarize the various constraints in the attitude maneuver and give the original attitude optimization problem P0. In this section, P0 is transformed by interpolation into P1, which uses the fitting coefficients of the attitude curve and the maneuver time as decision variables. P1 is subsequently discretized to obtain the computationally friendly P2. To solve P2, we derive the analytic derivatives for the attitude motion driven by the second-order system (i.e., eqs. (7) and (8)). Finally, the optimized solution is obtained using a gradient-based optimization method. It is worth noting that there are many gradient optimization algorithms that can be applied to solve P2, and in this paper, we use one of them, called the trust-region method.

#### 3.1 Attitude trajectory

##### 3.1.1 Attitude trajectory design

To give a smooth curve passing through a series of specific nodes in a non-Euclidean SO(3) space, ref. [29] developed a method for interpolating on SO(3) based on a rolling map. The key step in the method is to project the nodes onto the affine tangent space of a particular point and solve the corresponding interpolation problem in that affine tangent space, and then project the resulting curve back to SO(3). Since SO(3) is a smooth sub-manifold of  $\mathbb{R}^{3 \times 3}$ , the affine tangent space at any point can be regarded as an affine subspace of  $\mathbb{R}^{3 \times 3}$  [30]. Thus the interpolation problem on twisted SO(3) becomes an interpolation problem in Euclidean space.

We use this method to design a curve  $\mathbf{R}(t)$  connecting  $\mathbf{R}_0$  and  $\mathbf{R}_f$  on the time interval  $[0, T]$ . Using the left multiplication of Lie group SO(3), the starting point of the curve  $\mathbf{R}(t)$  can be mapped to the identity matrix  $\mathbf{I}$ . The problem is then equivalent to designing a curve  $\gamma(t)$  connecting  $\mathbf{I}$  and  $\tilde{\mathbf{R}} = \mathbf{R}_0^{-1}\mathbf{R}_f$ . To simplify the boundary conditions (11), the curve  $\mathbf{R}(t)$  is designed in a complex form. Specifically, the curves  $\mathbf{R}(t)$  and  $\gamma(t), t \in [0, T]$  have the following form [29, 30]:

$$\begin{cases} \mathbf{R}(t) = \mathbf{R}_0\gamma(t), \\ \gamma(t) = \mathbf{U}(t) \exp(\mathbf{K}(t) - \mathbf{X}(t) - \mathbf{I}) \mathbf{W}^T(t), \\ \mathbf{X}(t) = t\mathbf{\Omega}, \\ \mathbf{U}(t) = \exp\left(t\frac{\mathbf{\Omega}}{2}\right), \mathbf{W}(t) = \exp\left(-t\frac{\mathbf{\Omega}}{2}\right), \\ \mathbf{\Omega} = \frac{\log(\tilde{\mathbf{R}})}{T}, \end{cases} \quad (13)$$

where  $\mathbf{K}(t)$  is a smooth curve in affine tangent space at  $\mathbf{I}$ , which implies that  $\mathbf{K}(t)$  has the form  $\mathbf{K}(t) = \mathbf{I} + \hat{\mathbf{x}}$ ,  $\mathbf{x} \in \mathbb{R}^3$ . Let  $\eta$  denote the exponential coordinate of  $\exp(\mathbf{K}(t) - \mathbf{X}(t) - \mathbf{I})$ , i.e.,  $\hat{\eta}(t) = \mathbf{K}(t) - \mathbf{X}(t) - \mathbf{I}$ . Using Lemmas 1 and 2, the derivative of the curve  $\gamma(t)$  with respect to time  $t$  is calculated as follows:

$$\begin{aligned} \dot{\gamma}(t) &= \frac{d}{dt} \mathbf{U}(t) \exp(\hat{\eta}(t)) \mathbf{W}^T(t) + \mathbf{U}(t) \frac{d}{dt} \exp(\hat{\eta}(t)) \mathbf{W}^T(t) \\ &\quad + \mathbf{U}(t) \exp(\hat{\eta}(t)) \frac{d}{dt} \mathbf{W}^T(t) \\ &= \frac{\mathbf{\Omega}}{2} \gamma(t) + \gamma(t) \frac{\mathbf{\Omega}}{2} + \gamma(t) (\mathbf{W}(t) \mathbf{D}(\eta) \dot{\eta})^\wedge, \end{aligned} \quad (14)$$

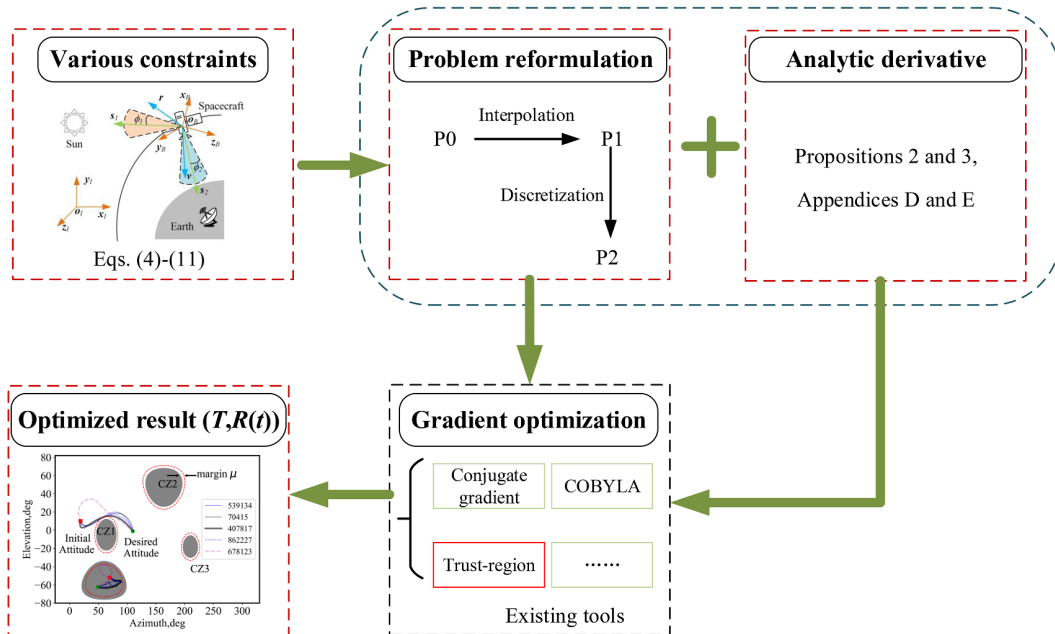


Figure 2 (Color online) Overview of the proposed method.

$$\begin{aligned}
\dot{\gamma}(t) &= \frac{\Omega}{2}\dot{\gamma}(t) + \dot{\gamma}(t)\frac{\Omega}{2} + \dot{\gamma}(t)(\mathbf{W}(t)\mathbf{D}(\eta)\dot{\eta})^\wedge \\
&\quad + \gamma(t)\left(\dot{\mathbf{W}}(t)\mathbf{D}(\eta)\dot{\eta} + \mathbf{W}(t)\frac{d}{dt}(\mathbf{D}(\eta)\dot{\eta})\right)^\wedge \\
&= \gamma(t)\left(-\frac{\Omega}{2}\mathbf{W}(t)\mathbf{D}(\eta)\dot{\eta} + \mathbf{W}(t)\dot{\mathbf{D}}(\eta)\dot{\eta} + \mathbf{W}(t)\mathbf{D}(\eta)\ddot{\eta}\right)^\wedge \\
&\quad + \frac{\Omega}{2}\dot{\gamma}(t) + \dot{\gamma}(t)\frac{\Omega}{2} + \dot{\gamma}(t)(\mathbf{W}(t)\mathbf{D}(\eta)\dot{\eta})^\wedge, \tag{15}
\end{aligned}$$

where

$$\begin{cases} \widehat{\eta}(t) = \dot{\mathbf{K}}(t) - \Omega, \\ \widehat{\eta}(t) = \ddot{\mathbf{K}}(t), \\ \dot{\mathbf{D}}(\eta) = \sum_{i=1}^3 \frac{\partial \mathbf{D}(\eta)}{\partial \eta_i} \frac{\partial \eta_i}{\partial t} = d_\eta \mathbf{D}(\eta) \cdot \dot{\eta}. \end{cases} \tag{16}$$

Detailed calculations of tangent operator  $\mathbf{D}(\eta)$  are given in Supporting Information.

**Proposition 1** Equation constraints (4) and (11) are satisfied when the exponential coordinate  $\eta(t)$  satisfies the following equations:

$$\begin{cases} \eta(0) = \eta(T) = \mathbf{0}, \\ \dot{\eta}(0) + \Omega^\vee = \mathbf{0}, \dot{\eta}(T) + \Omega^\vee = \mathbf{0}, \\ \ddot{\eta}(0) = \mathbf{0}, \ddot{\eta}(T) = \mathbf{0}. \end{cases} \tag{17}$$

*Proof.* See Appendix A of Supporting Information.

### 3.1.2 Nonlinear programming problem

Since  $\mathbf{K}(t)$  can be expressed as  $\mathbf{K}(t) = \mathbf{I} + \widehat{\mathbf{x}}, \mathbf{x} \in \mathbb{R}^3$ , we design  $\mathbf{K}(t)$  as  $(\mathbf{K}(t) - \mathbf{I})_i^\vee = \sum_{j=1}^p a_{i,j} h_j(t)$ , where  $a_{i,j}$  is the fitting coefficient and  $h_j(t)$  is some kind of basis function, such as a trigonometric or polynomial function. Based on this, the exponential coordinate and its derivatives of each order are calculated as follows:

$$\begin{cases} \eta_i(t) = (\mathbf{K}(t) - \mathbf{I})_i^\vee - \mathbf{X}_i^\vee(t) = \sum_{j=1}^p a_{i,j} h_j(t) - \mathbf{X}_i^\vee(t), \\ \dot{\eta}_i(t) = (\dot{\mathbf{K}}(t) - \dot{\mathbf{X}})_i^\vee = \sum_{j=1}^p a_{i,j} \dot{h}_j(t) - \Omega_i^\vee, \\ \ddot{\eta}_i(t) = \ddot{\mathbf{K}}_i^\vee(t) = \sum_{j=1}^p a_{i,j} \ddot{h}_j(t), \quad i = 1, \dots, 3. \end{cases} \tag{18}$$

The nonlinear optimization problem constructed with the objective of energy optimization is as follows:

$$\begin{aligned} \text{P1: } \min_{\{a_{i,j}, T\}} \quad & F = \int_0^T \mathbf{u}^T(t) \mathbf{u}(t) dt \\ \text{s.t.} \quad & \text{eqs. (4)–(11)}. \end{aligned} \tag{19}$$

Using Proposition 1, the equation constraints in eq. (19) can be transformed into constraints on the exponential coordinates:

$$\begin{cases} \mathbf{R}(0) = \mathbf{R}_0, \mathbf{R}(T) = \mathbf{R}_f, \\ \omega(0) = \omega(T) = \mathbf{0}, \\ \mathbf{u}(0) = \mathbf{u}(T) = \mathbf{0}, \\ \eta(0) = \eta(T) = \mathbf{0}, \\ \dot{\eta}(0) + \Omega^\vee = \dot{\eta}(T) + \Omega^\vee = \mathbf{0}, \\ \ddot{\eta}(0) = \ddot{\eta}(T) = \mathbf{0}. \end{cases} \tag{20}$$

By taking the square, the inequality constraints on angular velocity  $\omega_i$  and torque  $u_i$  are transformed into

$$\begin{cases} -\omega_{\max}^2 \leq \omega_i^2(t) - \omega_{\max}^2 \leq 0, \\ -u_{\max}^2 \leq u_i^2(t) - u_{\max}^2 \leq 0. \end{cases} \tag{21}$$

Considering the existence of various non-ideal situations (e.g., deviation during the actual attitude maneuver with respect to the target trajectory, inability to take into account all attitude pointings on the trajectory during discretization), we design an additional planning margin  $\mu$  for the attitude constraint region.

$$\begin{cases} -s_{1,k}^T \mathbf{R}(t) \mathbf{r} - \cos(\phi_{1,k} + \mu) \leq 0, \\ \cos(\phi_2 - \mu) - s_2^T \mathbf{R}(t) \mathbf{v} \leq 0. \end{cases} \tag{22}$$

## 3.2 Gradient-based trajectory planning

### 3.2.1 Discretized

To build a gradient-based optimization algorithm, we first discretize the continuous optimization problem (19). Suppose that there are  $N + 1$  discrete points and let  $\Delta t = T/N$ . Using the trapezoidal method, the discretized form of the objective function in eq. (19) is

$$\begin{aligned} F &= \sum_{n=0}^{N-1} \frac{1}{2} \left( \mathbf{u}^T((n+1)\Delta t) \mathbf{u}((n+1)\Delta t) + \mathbf{u}^T(n\Delta t) \mathbf{u}(n\Delta t) \right) \Delta t \\ &= \frac{1}{2} \Delta t \sum_{n=0}^{N-1} (f_{n+1} + f_n). \end{aligned} \tag{23}$$

Requiring constraints (21) and (22) to hold at each discrete time point  $n\Delta t$ ,  $n = 0, 1, \dots, N$ , we have

$$\begin{cases} -\omega_{\max}^2 \leq \omega_i^2(n\Delta t) - \omega_{\max}^2 \leq 0, \\ -u_{\max}^2 \leq u_i^2(n\Delta t) - u_{\max}^2 \leq 0, \\ -s_{1,k}^T \mathbf{R}(n\Delta t) \mathbf{r} - \cos(\phi_{1,k} + \mu) \leq 0, \\ \cos(\phi_2 - \mu) - s_2^T \mathbf{R}(n\Delta t) \mathbf{v} \leq 0. \end{cases} \tag{24}$$

Thus, the discretized form of the planning problem (19) is



as follows:

$$\begin{aligned} \text{P2: } \min_{\{a_{i,j}, T\}} \quad & F = \frac{1}{2} \Delta t \sum_{n=0}^{N-1} (f_{n+1} + f_n) \\ \text{s.t.} \quad & \text{eqs. (7)–(9), (20), (24),} \end{aligned} \quad (25)$$

where  $i = 1, \dots, 3, n = 1, \dots, N-1, k = 1, \dots, q$ , and  $q$  is the number of forbidden zones. The discretized inequality constraints in eq. (25), where  $n$  is not taken to 0 and  $N$ , are due to the fact that the angular velocity and torque are already limited by the equation constraints at  $t = 0$  and  $t = T$ , while the initial and target attitudes always satisfy the pointing constraints.

### 3.2.2 Calculation of derivatives

In this section, we give the calculation of the derivatives of the exponential coordinate and the curve  $\mathbf{R}(t)$  with respect to the optimization variables  $a_{i,j}$  and  $T$  for the application of the gradient optimization algorithm.

**Proposition 2** Let  $i = 1, \dots, 3, j = 1, \dots, p$  and  $n = 0, \dots, N$ . The exponential coordinate  $\boldsymbol{\eta}$  is as defined in eq. (18). Then the following equations hold:

$$(i) \begin{cases} \frac{\partial}{\partial a_{i,j}} \boldsymbol{\eta}(n\Delta t) = h_j(n\Delta t) \mathbf{e}_i, \\ \frac{\partial}{\partial T} \boldsymbol{\eta}(n\Delta t) = \frac{n}{N} (\dot{\boldsymbol{\eta}}(n\Delta t) + \boldsymbol{\Omega}^\vee), \end{cases} \quad (26)$$

$$(ii) \begin{cases} \frac{\partial}{\partial a_{i,j}} \dot{\boldsymbol{\eta}}(n\Delta t) = \dot{h}_j(n\Delta t) \mathbf{e}_i, \\ \frac{\partial}{\partial T} \dot{\boldsymbol{\eta}}(n\Delta t) = \frac{n}{N} \ddot{\boldsymbol{\eta}}(n\Delta t) + \frac{\boldsymbol{\Omega}^\vee}{T}, \end{cases} \quad (27)$$

$$(iii) \begin{cases} \frac{\partial}{\partial a_{i,j}} \ddot{\boldsymbol{\eta}}(n\Delta t) = \ddot{h}_j(n\Delta t) \mathbf{e}_i, \\ \frac{\partial}{\partial T} \ddot{\boldsymbol{\eta}}(n\Delta t) = \frac{n}{N} \dddot{\boldsymbol{\eta}}(n\Delta t). \end{cases} \quad (28)$$

*Proof.* See Appendix B of Supporting Information.

The derivatives of the curve  $\mathbf{R}(n\Delta t) = \mathbf{R}_0 \boldsymbol{\gamma}(n\Delta t)$  with respect to  $a_{i,j}$  and  $T$  are given by Proposition 3.

**Proposition 3** Let  $i = 1, \dots, 3, j = 1, \dots, p$  and  $\boldsymbol{\eta}_z = \left[ \frac{\partial \eta_1}{\partial z}, \frac{\partial \eta_2}{\partial z}, \frac{\partial \eta_3}{\partial z} \right]^T, z = a_{i,j}, T$ . Then the following equations hold:

$$(i) \frac{\partial}{\partial z} \dot{\mathbf{D}}(\boldsymbol{\eta}) = d_{\boldsymbol{\eta}}(d_{\boldsymbol{\eta}} \mathbf{D}(\boldsymbol{\eta}) \cdot \dot{\boldsymbol{\eta}}) \cdot \boldsymbol{\eta}_z + d_{\dot{\boldsymbol{\eta}}}(d_{\boldsymbol{\eta}} \mathbf{D}(\boldsymbol{\eta}) \cdot \dot{\boldsymbol{\eta}}) \cdot \dot{\boldsymbol{\eta}}_z, \quad (29)$$

$$(ii) \frac{\partial}{\partial a_{i,j}} \mathbf{R}(n\Delta t) = \frac{\partial \eta_i}{\partial a_{i,j}} \mathbf{R}_0 \mathbf{U}(n\Delta t) \frac{\partial}{\partial \eta_i} \exp(\boldsymbol{\eta}) \mathbf{W}^T(n\Delta t), \quad (30)$$

$$\frac{\partial}{\partial T} \mathbf{R}(n\Delta t) = \mathbf{R}_0 \mathbf{U}(n\Delta t) \sum_{i=1}^3 \left( \frac{\partial}{\partial \eta_i} \exp(\boldsymbol{\eta}) \frac{\partial \eta_i}{\partial T} \right) \mathbf{W}^T(n\Delta t), \quad (31)$$

where  $\frac{\partial \eta_i}{\partial a_{i,j}}$  and  $\frac{\partial \eta_i}{\partial T}$  are given by Proposition 2 and  $\frac{\partial}{\partial \eta_i} \exp(\boldsymbol{\eta})$  by Lemma 1.

*Proof.* See Appendix C of Supporting Information.

**Remark 2** The first term in eq. (C1) of Supporting Information is the second-order directional derivative of the tangent operator (D2) of Supporting Information. Since  $\dot{\boldsymbol{\eta}}$  is affected by  $z$ , the second term is derived naturally when applying the chain rule. The exact expression can be obtained by referring to the Appendix of [33] and deriving (D1) of Supporting Information term by term.

The derivative of the objective function  $F$  with respect to  $a_{i,j}$  in eq. (19) is difficult to compute, so the derivative of the discrete form eq. (23) with respect to  $a_{i,j}$  is used as an approximation, i.e.,

$$\frac{\partial}{\partial a_{i,j}} F = \frac{1}{2} \Delta t \sum_{n=0}^{N-1} \frac{\partial}{\partial a_{i,j}} (f_{n+1} + f_n), \quad (32)$$

$$\frac{\partial}{\partial a_{i,j}} f_n = 2\mathbf{u}^T(n\Delta t) \frac{\partial \mathbf{u}(n\Delta t)}{\partial a_{i,j}}. \quad (33)$$

In contrast, the derivative of  $F$  with respect to  $T$  can be calculated exactly:

$$\frac{\partial}{\partial T} F = \mathbf{u}^T(T) \mathbf{u}(T). \quad (34)$$

Based on Propositions 1–3, the calculation of the derivatives of  $\dot{\boldsymbol{\gamma}}(n\Delta t), \ddot{\boldsymbol{\gamma}}(n\Delta t), \boldsymbol{\omega}$ , and  $\mathbf{u}$  with respect to  $a_{i,j}$  and  $T$  can be accomplished. Details of the calculations are given in Appendix E of Supporting Information. The derivatives needed in the gradient optimization algorithm have all been computed analytically. In the next section, different simulation cases will be designed to verify the effectiveness of the proposed method.

## 4 Numerical simulations

In this section, we design three simulation cases to compare the effects of the algorithms under different basis functions, different optimization objectives, and different numbers of pointing constraints. Among the various optimization methods available, such as gradient descent, Newton's method, and gradient-free methods [35, 36], we selected the trust-region method to solve problem P2. Trust-region methods are a class of gradient-based optimization algorithms known for their robust convergence properties and ability to handle complex objective functions [37]. These methods have been successfully applied in diverse fields, including stochastic optimization and reinforcement learning. Additionally, the trust-region method is well supported by mature software implementations, such as the “scipy.optimize.minimize” function in Python's SciPy library [38], ensuring reliable and efficient optimization. Since the development and comparison of

optimization algorithms is not the primary focus of this study, we chose the trust-region method to provide a solid foundation for our numerical simulations. The inertia matrix of the spacecraft is  $\mathbf{J} = \text{diag}([18, 20, 30]) \text{ kg m}^2$ . The spacecraft in these three simulation cases have the same initial attitude  $\mathbf{R}_0$  and target attitude  $\mathbf{R}_f$ , respectively

$$\mathbf{R}_0 = \begin{bmatrix} 0.3249 & 0.2982 & 0.8975 \\ 0.9424 & -0.0224 & -0.3337 \\ -0.0794 & 0.9542 & -0.2883 \end{bmatrix},$$

$$\mathbf{R}_f = \begin{bmatrix} -0.4888 & -0.7674 & 0.4149 \\ 0.4370 & 0.1963 & 0.8778 \\ -0.7551 & 0.6104 & 0.2394 \end{bmatrix}. \quad (35)$$

The rest of the simulation conditions are invariant across cases as  $t_f = 50 \text{ s}$ ,  $\Omega_{\max} = 5^\circ/\text{s}$ ,  $u_{\max} = 0.5 \text{ Nm}$ ,  $N = 15$ ,  $\mu = 3^\circ$  and  $p = 10$ . The maximum number of iterations in each optimization is 300. The pointing constraints used in the simulation are shown in Table 1. The basis functions (BF), objective function (OF), average maneuver time (AMT), average control effort (ACE), and success rate (SR) for all the simulations are then tabulated in Table 2.

#### 4.1 Simulations with different basis functions

In the simulations of this section, we compare the effects of the proposed algorithms with different initializations and different basis functions with the objective of minimizing the control effort (23). In these simulations, there are three forbidden zones, i.e., CZ1–CZ3, details of which are given in Table 1. Only one vector  $\mathbf{r}$  needs to remain outside the forbidden zones during attitude maneuvering. This vector  $\mathbf{r}$  is kept immobile in the body frame, and its coordinates in the body frame is  $\mathbf{r} = [0.5909, 0.4362, 0.6787]^T$ . In Figures 3

and 4, we use the Legendre and trigonometric functions on the interval  $[0, t_f]$ , respectively. And we initialize  $a_{i,j}$  and  $T$  with the following random number seeds:

$$\begin{cases} \text{seed 1} = 842431, \text{seed 2} = 305847, \text{seed 3} = 89078, \\ \text{seed 4} = 539134, \text{seed 5} = 652957, \text{seed 6} = 339035, \\ \text{seed 7} = 262054, \text{seed 8} = 70415, \text{seed 9} = 538382, \\ \text{seed 10} = 74148, \text{seed 11} = 672072, \text{seed 12} = 407817, \\ \text{seed 13} = 985535, \text{seed 14} = 545240, \text{seed 15} = 411771, \\ \text{seed 16} = 731187, \text{seed 17} = 862227, \text{seed 18} = 901009, \\ \text{seed 19} = 636102, \text{seed 20} = 678123. \end{cases} \quad (36)$$

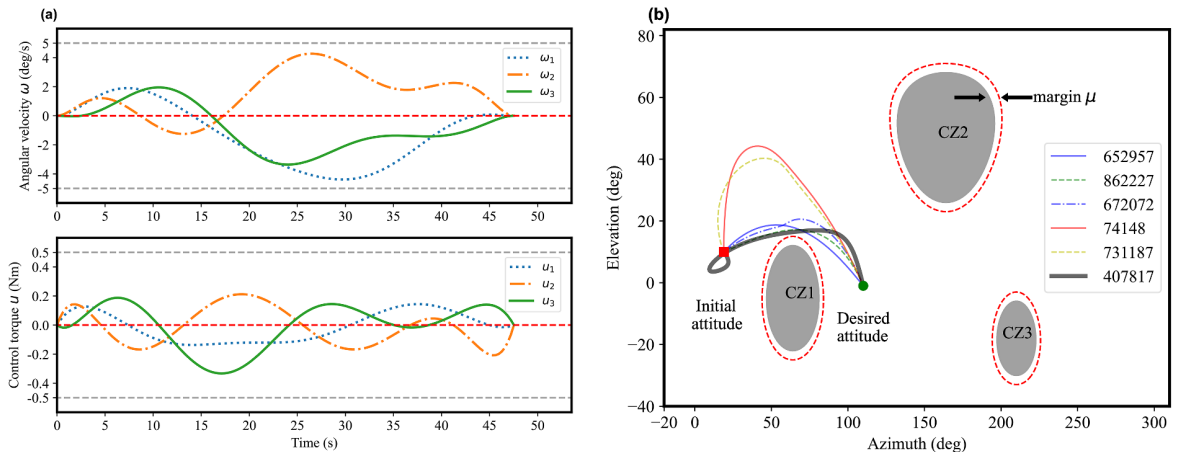
As shown in Figures 3(b) and 4(b), in both sets of simulations, the starting point ( $\mathbf{R} = \mathbf{R}_0$ ) of the vector  $\mathbf{r}$  is located in the upper left corner of the forbidden zone CZ1, and ending point ( $\mathbf{R} = \mathbf{R}_f$ ) of  $\mathbf{r}$  is between CZ1 and CZ3. In the box plot Figure 5, we count the control efforts of all feasible

**Table 1** Attitude constraints in simulations

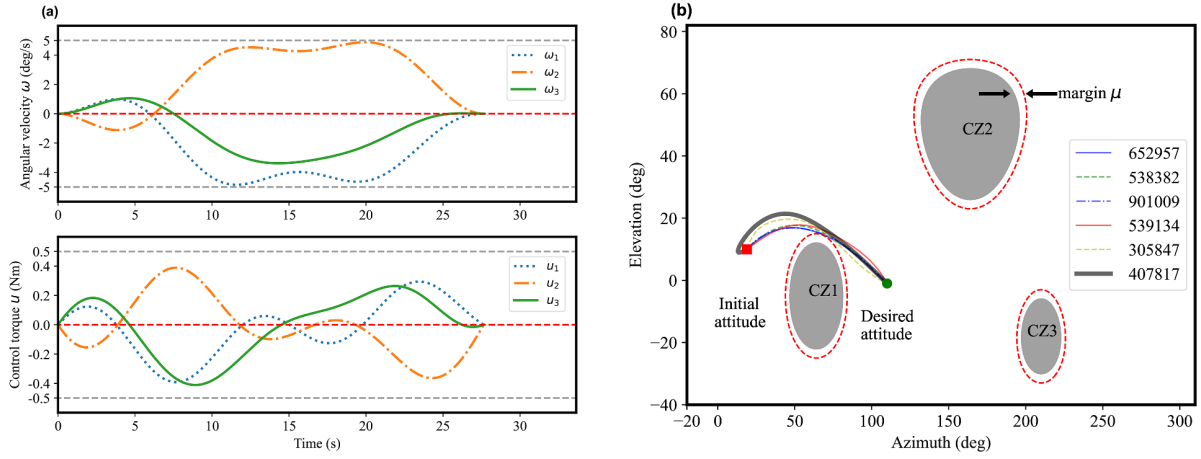
Constrained zone	Type	Constrained object	Angle
CZ1	Forbidden	$[0.4367, 0.8954, -0.0872]^T$	$17^\circ$
CZ2	Forbidden	$[-0.6556, 0.1880, 0.7314]^T$	$21^\circ$
CZ3	Forbidden	$[-0.8236, -0.4755, -0.3090]^T$	$12^\circ$
CZ4	Mandatory	$[0.2868, 0.4967, -0.8192]^T$	$21^\circ$

**Table 2** Statistics of simulation results within 300 iterations

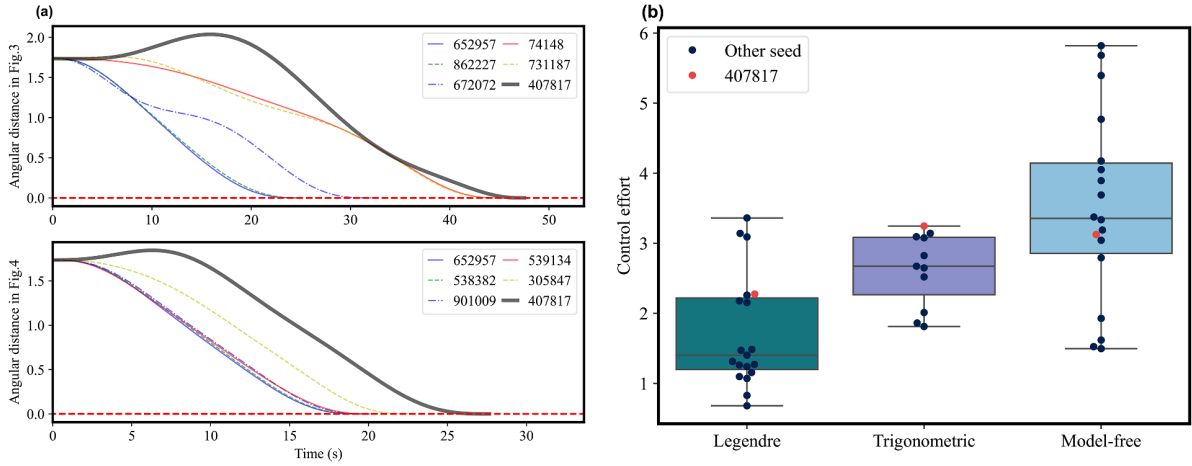
	BF	OF	AMT (s)	ACE	SR
1 (Figure 3)	Legendre	Control effort	33.7112	1.7228	19/20
2 (Figure 4)	Trigonometric	Control effort	23.7214	2.6277	11/20
3 (Figure 6)	Legendre	Maneuver time	22.9158	3.4942	19/20
4 (Figure 7)	Legendre	Model-free	33.1689	3.4942	18/20
5 (Figure 9)	Legendre	Control effort	37.9346	1.3516	18/20
6 (Figure 10)	Legendre	Control effort	35.2412	1.7907	18/20



**Figure 3** (Color online) Attitude trajectories using Legendre functions. (a) Angular velocity  $\omega$  and control torque  $\mathbf{u}$  curves (seed = 407817); (b) trajectories in 2D.



**Figure 4** (Color online) Attitude trajectories using trigonometric functions. (a) Angular velocity  $\omega$  and control torque  $u$  curves (seed = 407817); (b) trajectories in 2D.



**Figure 5** (Color online) Angular distance curves and box plot using different basis functions. (a) Angular distances in Figures 3 and 4; (b) optimized objective values.

solutions. As can be seen from Table 2 and Figure 5(b), using the Legendre function has a higher success rate and lower control consumption. For this reason, we are using the Legendre functions in subsequent simulations.

To further demonstrate the simulation results, we plot pointing trajectories (i.e., Figures 3(b) and 4(b)) and state curves (i.e., Figures 3(a), 4(a), and 5(a)) for a selection of simulations. Our selection criteria are: the 5 simulations that are closest to the mean of the objective function values and the simulation with a random number seed 12 = 407817 (the bolded one). This criterion is also used subsequently. Regardless of the basis functions used, the paths of the vector  $r$  are from above around CZ1 to the ending point. The angular distances relative to the target attitude  $R_f$  are shown in Figure 5(a), where the maneuver times using trigonometric functions are shorter. When the random number seed is 407817 (seed 12), the angular velocity and torque curves obtained using the Legendre functions are shown in Figure 3(a), and the results obtained using the trigonometric functions are shown

in Figure 4(a). None of these curves exceed the maximum value limit. Comparing the above results, it can be seen that the choice of basis functions will directly affect the success and optimality of the optimization, and the use of the Legendre functions in the optimization leads to a higher success rate and lower control efforts.

Since general attitude maneuver durations are on the order of  $10^1$  to  $10^2$  s, the coefficients  $a_{i,j}$  can be randomly initialized when such basis functions with small maxima are used in integrated attitude planning. However, for polynomial basis functions  $\{1, t, t^2, t^3, \dots\}$ ,  $(10^2)^9$  is an excessively large value, and therefore the coefficients should be initialized to decrease as the order of the basis function increases.

## 4.2 Simulations with different objective functions

In Supporting Information, the formulas for calculating the derivatives of various states with respect to the variables  $a_{i,j}$  and  $T$  during maneuver are given. Constructing the opti-

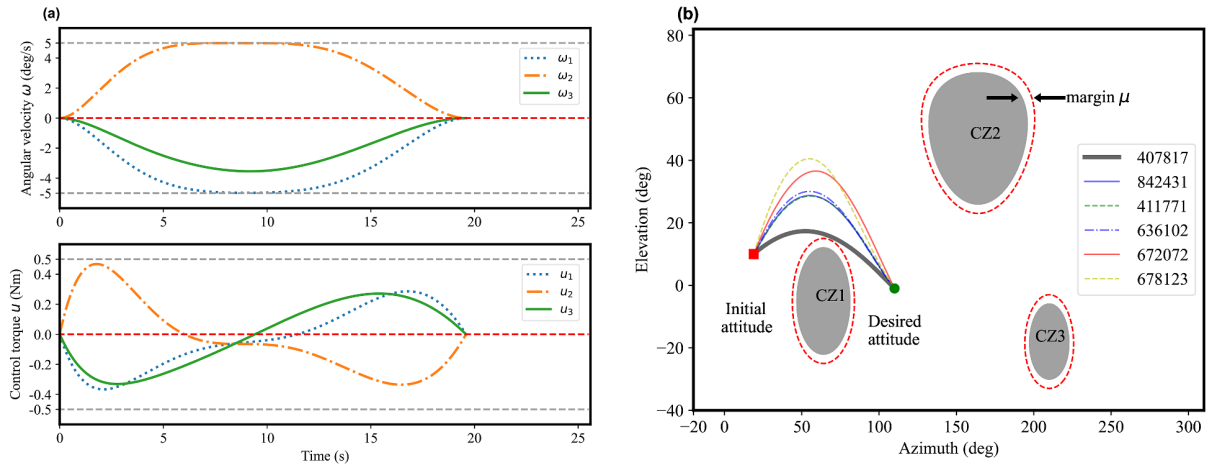


mization objective with different states can obtain attitude trajectories in different physical senses without modifying the overall structure of the algorithm. This implies that our proposed method is scalable. To illustrate this advantage, we carry out the two groups of simulations in this section. In the simulations shown in Figure 6, we aim to minimize the maneuvering time, i.e., the objective function is  $F = T$ . In the simulation shown in Figure 7, on the other hand, we perform a model-free optimization, i.e., the objective function is  $F = \sum_{n=0}^{N-1} \frac{1}{2} (\dot{\omega}^T ((n+1)\Delta t) \dot{\omega} ((n+1)\Delta t) + \dot{\omega}^T (n\Delta t) \dot{\omega} (n\Delta t)) \Delta t$  and the constraints on the control torque are replaced with constraints on the angular velocity derivatives. The rest of the unspecified settings are the same as Section 4.1.

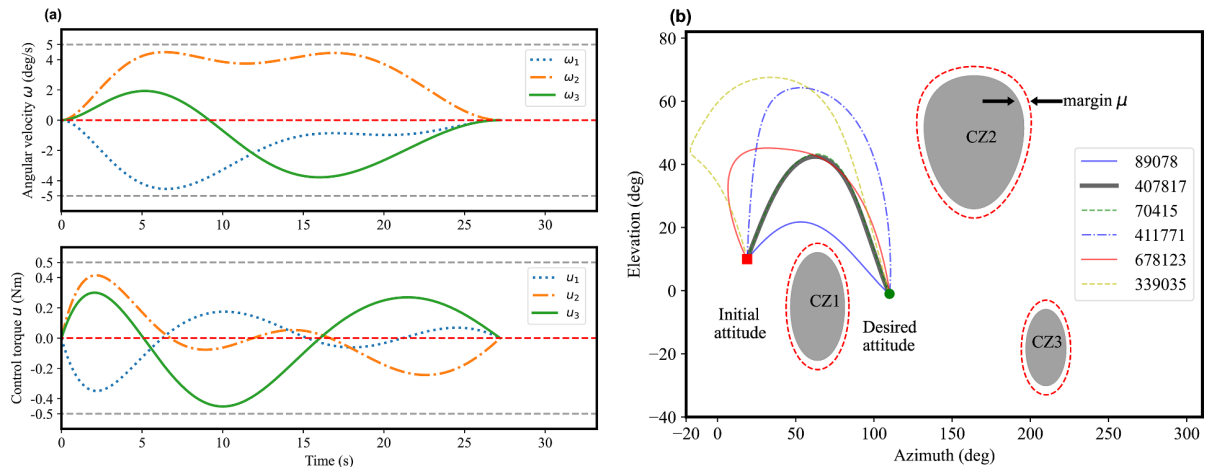
The statistics of the simulation results using the Legendre function to minimize the maneuvering time are shown in Figure 8 and Table 2. Only the simulation corresponding to seed 13 does not give a feasible solution within 300 iterations, and the average maneuver time for the rest of the simulations is 22.9158 s. Following the criteria described in Section 4.1,

we show the trajectories of six simulations in Figures 6(b) and 8(a). The vector  $\mathbf{r}$  goes around CZ1 from above to the end. When seed = 407817, the angular velocity and torque curves are shown in Figure 6(a). All the above curves satisfy the kinematic, dynamic, and time constraints.

The results of the model-free optimization can be found in Figures 7 and 8, and Table 2. This approach is similar to that of refs. [20, 22]. With appropriate constraints, this approach can give feasible solutions that satisfy attitude constraints, angular velocity constraints, and torque constraints, as illustrated in Figure 7(a) and (b). However, it must be noted that this approach has no direct effect in reducing control consumption and maneuver time. In the box plots (Figures 5(b) and 8(b)), we count the control efforts and maneuvering times in the model-free optimization, from which we can see that the mean and range of control consumption and maneuvering time of the model-free optimization are larger compared to the experiments in Figures 3 and 4 (minimizing control efforts as an objective) and Figure 6 (minimizing maneuvering



**Figure 6** (Color online) Attitude trajectories when minimizing maneuver time. (a) Angular velocity  $\omega$  and control torque  $u$  curves; (b) trajectories in 2D.



**Figure 7** (Color online) Attitude trajectories in model-free optimization. (a) Angular velocity  $\omega$  and control torque  $u$  curves; (b) trajectories in 2D.

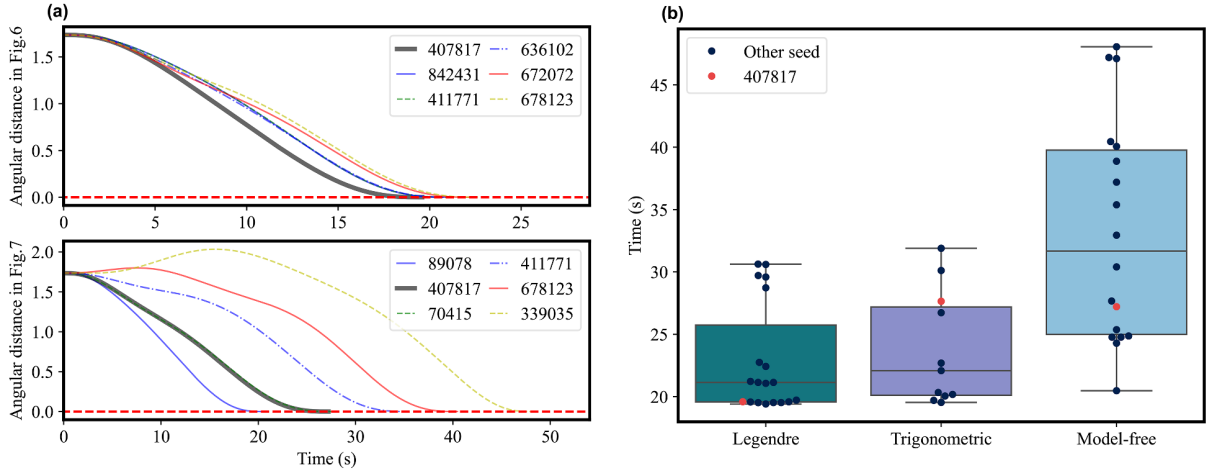
time as an objective).

### 4.3 Simulations with multi-pointing constraints

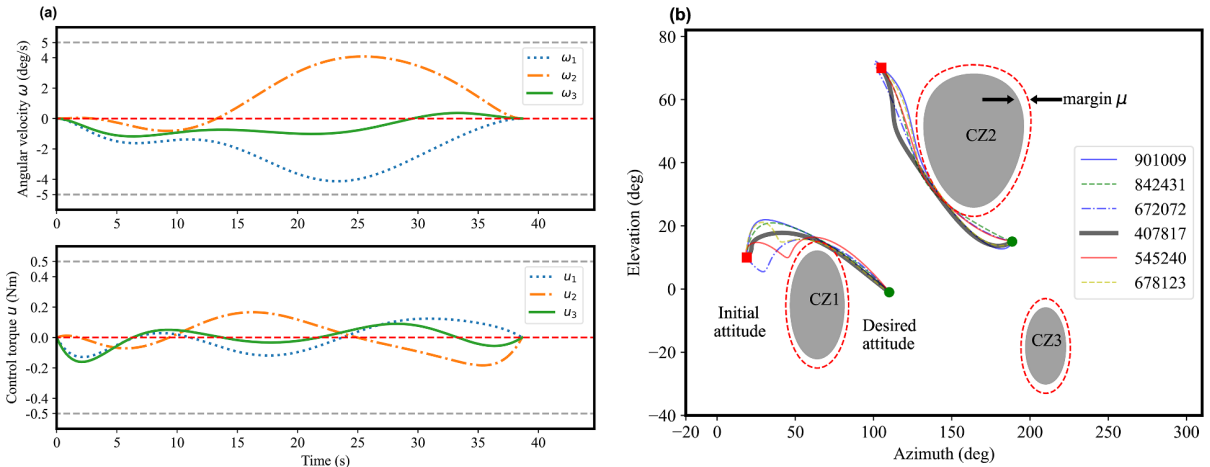
The gradient optimization method used in this paper can handle multiple attitude constraints. In this section, two sets of simulations are designed to illustrate the effectiveness of the algorithm in handling multiple attitude constraints. In Scenario 1, the number of vectors that must be kept outside the forbidden zones in the first set of simulations is 2, that is,  $\mathbf{r}_1 = [0.5909, 0.4362, 0.6787]^T$ ,  $\mathbf{r}_2 = [0.7757, -0.1397, -0.6154]^T$ . On the unit sphere,  $\mathbf{r}_1$  starts and ends at the same location as Sections 4.1 and 4.2, i.e., the starting point is located to the upper left of CZ1 and the ending point is located between CZ1 and CZ3, while the starting point of  $\mathbf{r}_2$  is to the left of CZ2 and the ending point is located between CZ2 and CZ3. In Scenario 2, the vector that needs to be kept outside the forbid-

den zones is  $\mathbf{r} = [0.5909, 0.4362, 0.6787]^T$ , and the vector that needs to be kept inside the mandatory zone is  $\mathbf{v} = [0.6762, -0.7021, 0.2231]^T$ . The rest of the unspecified settings are the same as Section 4.1.

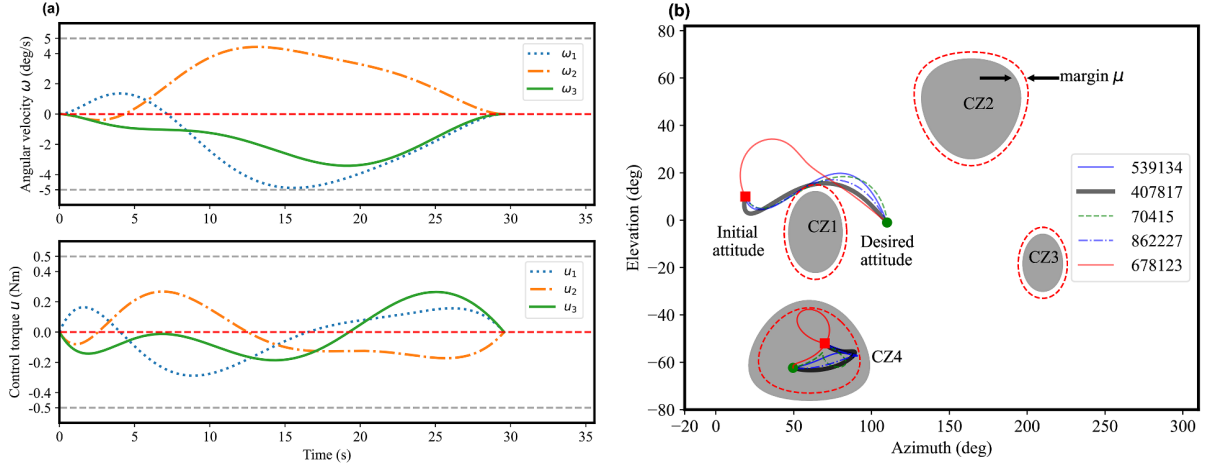
In both scenarios, 18 out of 20 simulations give feasible trajectories. The control consumption for the two scenarios is 1.3516 and 1.7907, respectively. The details of the simulation are further demonstrated in Figures 9, 10, and 11(a). The pointing trajectories in Scenario 1 are shown in Figure 9(b), where the vector  $\mathbf{r}_1$  goes around the top of CZ1 and reaches the ending point, while the vector  $\mathbf{r}_2$  goes around the bottom left of CZ2 and reaches the ending point. In Scenario 2, the trajectories of the vector  $\mathbf{v}$  remain within CZ4 at all times. In Figure 10(b), there are curves for only 5 simulations, and this is because the results corresponding to 407817 (seed 12) are close to the mean in Scenario 2, as illustrated in Figure 11(b). The corresponding angular velocity and torque curves for these two scenarios when initializing the variables



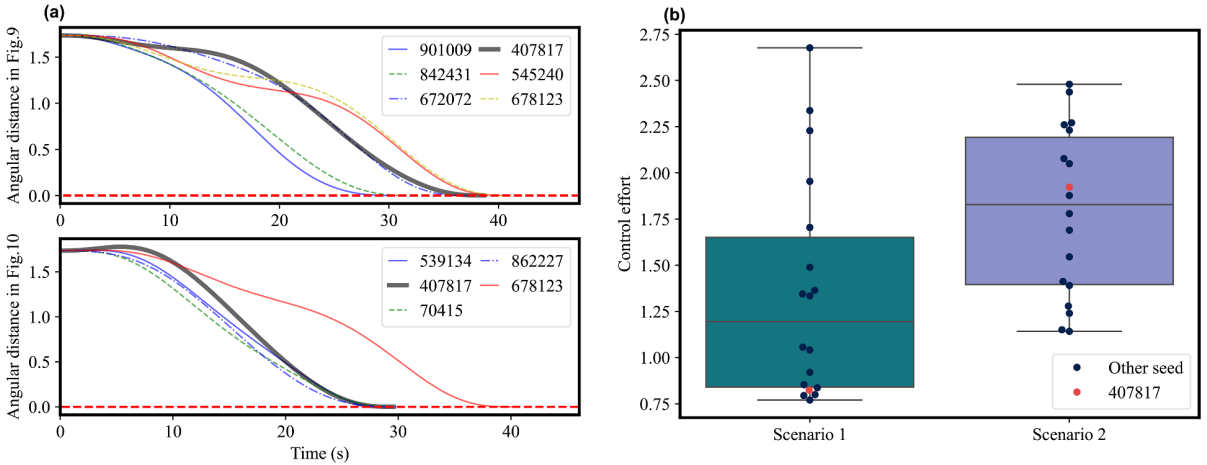
**Figure 8** (Color online) Angular distance curves and box plot using different objective functions. (a) Angular distances in Figures 6 and 7; (b) optimized objective values.



**Figure 9** (Color online) Attitude trajectories under multiple forbidden constraints (Scenario 1). (a) Angular velocity  $\omega$  and control torque  $u$  curves; (b) trajectories in 2D.



**Figure 10** (Color online) Attitude trajectories with both forbidden and mandatory constraints (Scenario 2). (a) Angular velocity  $\omega$  and control torque  $u$  curves; (b) trajectories in 2D.



**Figure 11** (Color online) Angular distance curves and box plot when dealing with multi-directional constraints. (a) Angular distances in Figures 9 and 10; (b) optimized objective values.

$a_{i,j}$  and  $T$  using 407817 (seed 12) are shown in Figures 9(a) and 10(a). None of the curves violates constraints. Comparison with Section 4.1 shows that under multiple attitude constraints, the effectiveness of the method proposed in this paper does not deteriorate and still maintains the same level in terms of success and optimality.

## 5 Conclusion

In this paper, we develop a gradient-based attitude planning algorithm to generate an attitude curve that simultaneously satisfies attitude pointing, angular velocity, control torque, and maneuver time constraints. Firstly, we design an attitude curve based on an attitude interpolation algorithm whose shape is directly controlled by the fitting coefficients and maneuvering time, and the equation constraints about the attitude curve are transformed into constraints about these fitting coefficients and maneuvering time. Secondly, to avoid designing a time allocation algorithm, we build a nonlinear attitude

planning problem by using both the fitting coefficients and the maneuver time as decision variables. Also, to apply the gradient optimization algorithm, we give an exact formula for the derivatives of the attitude curve with respect to the decision variables. Finally, numerical simulations are performed with different basis functions, different initializations, different objective functions and different types of pointing constraints. In all these simulations, feasible solutions can be given within a given number of iterations with a high success rate, which validates the effectiveness of the proposed algorithm.

Future work focuses on two aspects. First, how to deal with non-regular and time-varying pointing constraints is a problem that needs to be investigated, e.g., the forced pointing constraints that exist in satellite communication and observation missions are time-varying, and a certain pointing of the satellite needs to maintain a fixed angle with a time-varying target. Second, how to apply the proposed planning algorithm in the attitude planning of satellites with movable

parts is another problem to be investigated. For example, in satellite laser communication, the laser terminal has additional movable degrees of freedom.

**Acknowledgements** This work was supported by Shenzhen Science and Technology Program (Grant No. JCYJ20220818102207015) and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023B1515120018).

**Supporting Information** The supporting information is available online at [tech.scichina.com](http://tech.scichina.com) and [link.springer.com](http://link.springer.com). The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

## References

- Luo Y, Liu G, Guo L, et al. Scalable wing sailing and snowboarding enhance efficient and energy-saving mobility of polar robot. *IEEE ASME Trans Mechatron*, 2024, 29: 3833–3844
- Calaon R, Schaub H. Constrained attitude maneuvering via modified-rodrigues-parameter-based motion planning algorithms. *J Spacecraft Rockets*, 2022, 59: 1342–1356
- Abenezer G T, Roberto V, Akshay R, et al. Safe and scalable real-time trajectory planning framework for urban air mobility. *J Aerosp Inf Syst*, 2024, 21: 641–650
- Hu Q L, Xiao X D, Yang H Y, et al. Spacecraft attitude planning and control under multiple constraints: Review and prospects (in Chinese). *Acta Aeronaut Astronaut Sin*, 2022, 43: 403–431
- Xu R, Zhu Z, Li Z Y, et al. Research progress of spacecraft attitude maneuver planning technology (in Chinese). *J Astronaut*, 2023, 44: 155–1667
- Walsh A, Forbes J R. Constrained attitude control on SO(3) via semidefinite programming. *J Guidance Control Dyn*, 2018, 41: 2483–2488
- Shen Q, Yue C, Goh C H, et al. Rigid-body attitude stabilization with attitude and angular rate constraints. *Automatica*, 2018, 90: 157–163
- Kristiansen R, Hagen D. Modelling of actuator dynamics for spacecraft attitude control. *J Guidance Control Dyn*, 2009, 32: 1022–1025
- Shen Q, Yue C, Goh C H, et al. Active fault-tolerant control system design for spacecraft attitude maneuvers with actuator saturation and faults. *IEEE Trans Ind Electron*, 2019, 66: 3763–3772
- Zhou N, Cheng X, Xia Y, et al. Fully adaptive-gain-based intelligent failure-tolerant control for spacecraft attitude stabilization under actuator saturation. *IEEE Trans Cybern*, 2022, 52: 344–356
- Wang X, Wu G, Xing L, et al. Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions. *IEEE Syst J*, 2021, 15: 3881–3892
- Hablani H B. Attitude commands avoiding bright objects and maintaining communication with ground station. *J Guidance Control Dyn*, 1999, 22: 759–767
- Duan C, Hu Q, Zhang Y, et al. Constrained single-axis path planning of underactuated spacecraft. *Aerospace Sci Tech*, 2020, 107: 106345
- Yue C F, Huo T, Chen X Q, et al. Synergistic potential functions for constrained attitude control of rigid spacecraft (in Chinese). *Acta Anat Sin*, 2024, 50: 54–65
- Tian Y, Hu Q, Shao X. Adaptive fault-tolerant control for attitude re-orientation under complex attitude constraints. *Aerospace Sci Tech*, 2022, 121: 107332
- Tan X, Berkane S, Dimarogonas D V. Constrained attitude maneuvers on SO(3): Rotation space sampling, planning and low-level control. *Automatica*, 2020, 112: 108659
- Kjellberg H C, Lightsey E G. Discretized constrained attitude pathfinding and control for satellites. *J Guidance Control Dyn*, 2013, 36: 1301–1309
- Kjellberg H C, Lightsey E G. Discretized quaternion constrained attitude pathfinding. *J Guidance Control Dyn*, 2016, 39: 713–718
- Xu R, Fan Y, Li Z, et al. Time-optimal attitude planning for spacecraft with movable parts using second order cone programming. *Aerospace Sci Tech*, 2023, 141: 108589
- Celani F, Lucarelli D. Spacecraft attitude motion planning using gradient-based optimization. *J Guidance Control Dyn*, 2020, 43: 140–145
- Yue C F, Huo T, Lu M, et al. A systematic method for constrained attitude control under input saturation. *IEEE Trans Aerosp Electron Syst*, 2023, 59: 6005–6015
- Fan Y, Xu R, Li Z, et al. Polynomial attitude trajectory planning for spacecraft with movable parts using decoupled strategy. *J Guidance Control Dyn*, 2024, 47: 72–86
- He H, Shi P, Zhao Y. A pointing-based method for spacecraft attitude maneuver path planning under time-varying pointing constraints. *Adv Space Res*, 2022, 70: 1062–1077
- Feng K, Ji J C, Zhang Y, et al. Digital twin-driven intelligent assessment of gear surface degradation. *Mech Syst Signal Process*, 2023, 186: 109896
- Ni Q, Ji J C, Feng K, et al. Data-driven bearing health management using a novel multi-scale fused feature and gated recurrent unit. *Reliability Eng Syst Saf*, 2024, 242: 109753
- Feng K, Ji J C, Ni Q, et al. A novel vibration-based prognostic scheme for gear health management in surface wear progression of the intelligent manufacturing system. *Wear*, 2023, 522: 204697
- Dario S, Robert G M, Fabio C. Inverse dynamics particle swarm optimization applied to constrained minimum-time maneuvers using reaction wheels. *Aerosp Sci Technol*, 2018, 75: 1–12
- Wu C, Xu R, Zhu S, et al. Time-optimal spacecraft attitude maneuver path planning under boundary and pointing constraints. *Acta Astronaut*, 2017, 137: 128–137
- Hüper K, Leite F S. On the geometry of rolling and interpolation curves on  $S^n$ ,  $SO(n)$ , and Grassmann manifolds. *J Dyn Control Syst*, 2007, 13: 467–502
- Shen Y S, Hüper K, Leite F S. Smooth interpolation of orientation by rolling and wrapping for robot motion planning. In: *Proceedings of IEEE International Conference on Robotics and Automation (IRCA)*. Orlando, 2006. 113–118
- Richard M M, Li Z X, Sastry S S. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton: CRC Press, 1994. 25–30
- Gallego G, Yezzi A. A compact formula for the derivative of a 3-D rotation in exponential coordinates. *J Math Imag Vis*, 2015, 51: 378–384
- Todesco J, Bröls O. Highly accurate differentiation of the exponential map and its tangent operator. *Mechanism Machine Theor*, 2023, 190: 105451
- Rossmann W. *Lie Groups: An Introduction Through Linear Groups*. New York: Oxford University Press, 2002. 14–16
- Bertsekas D P. Nonlinear programming. *J Oper Res Soc*, 1997, 48: 334
- Floudas C A, Pardalos P M. *Encyclopedia of Optimization*. Berlin: Springer, 2008. 439–571
- Andrew R C, Nicholas I M G, Philippe L T. *Trust Region Methods*. Philadelphia: Society for Industrial and Applied Mathematics, 2000. 1–16
- Virtanen P, Gommers R, Oliphant T E, et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat Methods*, 2020, 17: 261–272