SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2026, Vol. 69, Iss. 9, 192206:1–192206:20
https://doi.org/10.1007/s11432-025-5022-9

Hierarchical multi-terminal attitude planning for rigid-body spacecraft

Tao HUO[1], Jin YANG[2], Chengfei YUE[3,4*], Xueqin CHEN[1,3] & Xibin CAO[1,3]

[1]*School of Astronautics, Harbin Institute of Technology, Harbin 150001, China*
[2]*School of Management, Harbin Institute of Technology, Harbin 150001, China*
[3]*State Key Laboratory of Micro-Spacecraft Rapid Design and Intelligent Cluster, Harbin Institute of Technology, Harbin 150001, China*
[4]*School of Aerospace Science, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China*

Received 27 June 2025/Revised 14 September 2025/Accepted 24 October 2025/Published online 26 August 2026

Abstract This paper proposes a hierarchical attitude planning method to address the complex attitude planning problem for multi-rigid-body spacecraft equipped with multiple laser communication terminals under constraints. The proposed method comprises three key steps: attitude decomposition, constrained path search, and trajectory generation. First, attitude decomposition determines the desired spacecraft platform attitude and feasible terminal joint angles (satisfying range-of-motion limits) corresponding to the desired terminal pointing directions. Next, the multi-heuristic A* algorithm is employed to perform a search for a feasible attitude and joint angle sequence within the high-dimensional configuration space satisfying all pointing constraints. Finally, a novel Lie group SO(3)-based trajectory planning method is applied to generate smooth and dynamically feasible attitude trajectories connecting the discrete sequence, by solving a sequence of quadratic programs. Numerical simulations under different operational constraints demonstrate the effectiveness of the proposed method in generating feasible and smooth trajectories.

Keywords hierarchical attitude planning, multi-heuristic A*, three-dimensional special orthogonal group (SO(3)), attitude constraints, movable parts

Citation Huo T, Yang J, Yue C F, et al. Hierarchical multi-terminal attitude planning for rigid-body spacecraft. Sci China Inf Sci, 2026, 69(9): 192206, https://doi.org/10.1007/s11432-025-5022-9

1 Introduction

Attitude planning is a crucial component of a spacecraft's attitude control system. Spacecraft in orbit face various constraints when performing attitude maneuvers [1–4]. For example, equipment such as infrared telescopes and star trackers has attitude keep-out constraints, requiring that their sensitive axes not be exposed to bright celestial objects [5, 6]. Similarly, ground communication missions require mandatory pointing constraints, ensuring that the communication antenna remains continuously aligned with the ground station [7, 8]. Furthermore, structural strength limitations impose an upper limit constraint on the spacecraft's angular velocity during maneuvers [9, 10]. Constrained attitude maneuvers cannot be guaranteed by unconstrained control algorithms, which creates the need for attitude planning. Such planning involves designing feasible attitude trajectories under multiple constraints to serve as control inputs. This approach reduces controller complexity, improves precision, and plays a crucial role in increasingly complex space missions. As a result, generating feasible attitude paths under complex constraints has become an active research area.

In recent years, based on different approaches to dealing with pointing constraints, related research can be broadly categorized into three main approaches: the potential function approach [11–13], search-based approach [14–16], and optimization-based approach [17–19]. The potential function approach models constraints as regions of high potential and the desired attitude as a low potential minimum. Constraint avoidance is achieved by incorporating the negative gradient of this potential function into the control law. Search-based approaches discretize the attitude configuration space (typically SO(3)) and employ graph search algorithms (e.g., A*, RRT [20]) to explore feasible attitude sequences. Conversely, optimization-based approaches formulate attitude planning directly as a constrained nonlinear optimization problem, deriving the attitude trajectory as the solution to this problem.

* Corresponding author (email: yuechengfei@hit.edu.cn)

The rise of satellite internet has made attitude planning for satellites equipped with laser terminals an active research area [21, 22]. In such constellations, spacecraft must establish links with nodes in different orbital planes using laser terminals. Attitude maneuvers requiring multiple terminals are necessary both after deployment and during communication topology switches. However, existing planning algorithms struggle with these multi-terminal, multi-constraint scenarios due to several fundamental challenges: the curse of dimensionality, low computational efficiency, and inadequate dynamic feasibility.

These limitations manifest differently across various methodological approaches. Potential function methods, while effective for single rigid bodies under pointing constraints, scale poorly to multi-body systems. Search-based algorithms like A* become computationally intractable in high-dimensional spaces due to the curse of dimensionality. Nonlinear optimization approaches, though capable of handling complex constraints, often require prohibitively long computation times for practical implementation.

Furthermore, ensuring the dynamical feasibility of the planned attitude trajectory is a critical aspect that often requires careful consideration [23]. For instance, some search-based methods [14, 24] neglect the crucial step of smooth interpolation between discrete attitude nodes, resulting in non-smooth, potentially dynamically infeasible attitude profiles. While optimization-based methods [17, 21] can generate smooth trajectories, the physical interpretation of their objective functions is often unclear, making it difficult to predict or control the specific characteristics (e.g., energy consumption, maneuver time) of the resulting attitude path.

Inspired by these observations, we propose a hierarchical attitude planning method for spacecraft with three 2-degree-of-freedom (2-DOF) laser communication terminals. This method addresses key challenges in deploying low-Earth orbit satellite communication constellations: how to compute desired attitudes from topological link relationships and how to design feasible attitude trajectories under multi-terminal and multi-constraint conditions. To tackle these problems, a hierarchical structure is adopted to simplify computation by handling subsets of constraints at each step. Within this framework, the proposed method decomposes the problem into three sequential steps.

Step 1 (attitude decomposition). A simple nonlinear optimization problem is formulated for attitude decomposition, constrained by the terminals' pointing and maximum maneuverable angle, with the objective function being the minimization of the angular distance of the platform's maneuver. This decomposes the desired terminal pointing directions into corresponding desired platform attitude and feasible terminal joint angles.

Step 2 (constrained path search). The multi-heuristic A*(MHA*) algorithm is employed to address the high-dimensional path search problem. This step generates a discrete sequence of platform attitudes and terminal joint angles that satisfies all pointing constraints.

Step 3 (dynamically feasible trajectory generation). The design problems of platform attitude curves and joint angle curves are transformed into solving a series of quadratic programming (QP) problems. The variables in these QPs are the coefficients of basis functions fitting the attitude curves over segments of the path.

The main contributions of this paper are summarized as follows.

(i) A novel, comprehensive hierarchical attitude planning framework is presented. This framework integrates attitude decomposition, constrained path search, and dynamically feasible trajectory generation, enabling effective planning for high-dimensional spacecraft systems with multiple movable parts under complex constraints.

(ii) A new Lie group SO(3)-based trajectory planning methodology is introduced. This methodology efficiently transforms the constrained attitude trajectory design problem into solving a sequence of QPs. Crucially, we provide a rigorous theoretical analysis elucidating the physical significance (e.g., its relationship to minimizing angular acceleration) of the objective function within the established QP formulation.

2 Problem statement

2.1 Notation definitions and useful lemmas

The symbols used in this paper are defined as shown in Table 1. The definitions of SO(3) and so(3) are respectively $\mathrm{SO}(3)=\left\{\boldsymbol{R}\in\mathbb{R}^{3\times3}\,\middle|\,\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}=\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}}=\boldsymbol{I},\det(\boldsymbol{R})=1\right\}$ and $\mathrm{so}(3)=\left\{\boldsymbol{S}\in\mathbb{R}^{3\times3}\,\middle|\,\boldsymbol{S}^{\mathrm{T}}=-\boldsymbol{S}\right\}$. so(3) and $\mathbb{R}^3$ are isomorphic, and $\boldsymbol{x}\in\mathbb{R}^3$ and $\widehat{\boldsymbol{x}}\in\mathrm{so}(3)$ can be mapped one-to-one by the vector's cross-product operator. The lemmas that will be used in this paper are stated below.

Lemma 1 ([25]). Any rotation matrix $\boldsymbol{R}(\theta,\boldsymbol{w})$ is equivalent to a rotation about a fixed axis $\boldsymbol{w}\in S^2$ through an angle $\theta\in[0,2\pi)$.

The exponential map establishes a link between the equivalent axis representation and the rotation matrix, i.e., $\boldsymbol{R}(\theta,\boldsymbol{w}):=\exp(\theta\widehat{\boldsymbol{w}})=\boldsymbol{I}+\sin\theta\widehat{\boldsymbol{w}}+(1-\cos\theta)\widehat{\boldsymbol{w}}^2$, where angular distance $\theta=d(\boldsymbol{R},\boldsymbol{I})=\arccos\left(\left(\mathrm{tr}(\boldsymbol{R})-1\right)/2\right)$

Table 1 Notation definitions.

Notation	Meaning
$\mathbb{R}$	The set of real numbers.
$\boldsymbol{R} \in \mathbb{R}^{3\times 3}$	Rotation matrix of the satellite platform relative to the inertial frame F_I.
$\boldsymbol{I} \in \mathbb{R}^{3\times 3}$	Identity matrix.
SO(3)	Lie group formed by rotation matrices.
so(3)	Tangent space of Lie group SO(3) at identity element $\boldsymbol{I}$.
S^2	The unit sphere in $\mathbb{R}^3$.
$\boldsymbol{D}(\boldsymbol{x})$	Tangent operator of SO(3).
$\theta_i \in [-90^\circ, 90^\circ]$	Yaw angle of the laser terminal.
$\varphi_i \in [-15^\circ, 15^\circ]$	Pitch angle of the laser terminal.
$\boldsymbol{\eta}$	Exponential coordinates of a rotation matrix.
$\Omega_{\max}$	Upper limit of angular velocity for the satellite platform.
T	Time of attitude maneuver.
$\boldsymbol{r}_{1,j} \in S^2, j=1,\ldots,p$	Forbidden pointing in inertial frame F_I.
$\boldsymbol{r}_{2,j} \in S^2, j=1,\ldots,p$	Mandatory pointing in inertial frame F_I.
$\mathcal{N}_k$	The k-th node in the A* algorithm.
$\boldsymbol{\omega}_k$	Angular velocity on the k-th interval.
$\Vert\cdot\Vert$	2-norm of a vector.
$(\cdot)^{\mathrm{T}}$	Transpose of a matrix.
$(\cdot)^\wedge, \widehat{(\cdot)}$	Cross-product matrix of a vector.
$(\cdot)_0$	Initial value.
$(\cdot)_d$	Desired value.

and unit vector $\widehat{\boldsymbol{w}} = \left(\boldsymbol{R} - \boldsymbol{R}^{\mathrm{T}}\right)/\left(2\sin\theta\right)$. The inverse of the exponential map $\exp(\cdot)$ is denoted by $\log(\cdot)$, i.e., $\log\left(\boldsymbol{R}\left(\theta,\boldsymbol{w}\right)\right) = \theta\widehat{\boldsymbol{w}} \in \mathrm{so}(3)$. The bounded vector $\theta\boldsymbol{w}$ is called exponential coordinates for $\boldsymbol{R}$.

Lemma 2 ([25]). Given $\boldsymbol{R} \in \mathrm{SO}(3)$ and $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^3$, the following properties hold:

(i) $\boldsymbol{R}(\boldsymbol{p}\times\boldsymbol{q}) = (\boldsymbol{R}\boldsymbol{p})\times(\boldsymbol{R}\boldsymbol{q})$;

(ii) $\boldsymbol{R}\widehat{\boldsymbol{p}}\boldsymbol{R}^{\mathrm{T}} = (\boldsymbol{R}\boldsymbol{p})^\wedge$;

(iii) $\Vert\boldsymbol{R}\boldsymbol{p} - \boldsymbol{R}\boldsymbol{q}\Vert = \Vert\boldsymbol{p}-\boldsymbol{q}\Vert$.

Lemma 3 ([26]). Let ε denote a small positive constant, t denote time and $\boldsymbol{x}(t) = \left[x_1(t), x_2(t), x_3(t)\right]^{\mathrm{T}} \in \mathbb{R}^3$ denote a differentiable curve. Let $\alpha(\boldsymbol{x}) = \sin(\Vert\boldsymbol{x}\Vert)/\Vert\boldsymbol{x}\Vert$, $\beta(\boldsymbol{x}) = 2\left(1-\cos(\Vert\boldsymbol{x}\Vert)\right)/\Vert\boldsymbol{x}\Vert^2$, and $\dot{\boldsymbol{x}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t)$. Then the derivative of $\boldsymbol{R}(t) = \exp\left(\widehat{\boldsymbol{x}(t)}\right)$ with respect to time t is

$$\dot{\boldsymbol{R}}(t) = \boldsymbol{R}(t)\left(\boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}}\right)^\wedge, \tag{1}$$

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

$$\boldsymbol{D}(\boldsymbol{x}) = \begin{cases} \boldsymbol{I} - \frac{\beta(\boldsymbol{x})}{2}\widehat{\boldsymbol{x}} + \frac{1-\alpha(\boldsymbol{x})}{\Vert\boldsymbol{x}\Vert^2}\widehat{\boldsymbol{x}}^2, & \Vert\boldsymbol{x}\Vert \geqslant \varepsilon, \\ \boldsymbol{I}, & \Vert\boldsymbol{x}\Vert < \varepsilon. \end{cases} \tag{2}$$

Lemma 4 ([27]). The angular distance $d(\boldsymbol{R}_1, \boldsymbol{R}_2)$ on SO(3) has the following properties:

(i) The angular distance is double translation invariant, i.e., $d\left(\boldsymbol{R}\boldsymbol{R}_2, \boldsymbol{R}\boldsymbol{R}_1\right) = d\left(\boldsymbol{R}_2\boldsymbol{R}, \boldsymbol{R}_1\boldsymbol{R}\right) = d\left(\boldsymbol{R}_2, \boldsymbol{R}_1\right), \forall \boldsymbol{R} \in \mathrm{SO}(3)$;

(ii) The angular distance is physically significant as the maximum angle of one rotation.

2.2 Problem statement and overview of approach

This paper investigates the attitude planning problem for spacecraft in a low-Earth-orbit communication constellation. The constellation comprises satellites equipped with multiple laser terminals. Due to orbital motion, varying communication tasks, and other factors, the constellation's communication topology undergoes dynamic changes. When the communication topology shifts, spacecraft must execute corresponding attitude maneuvers to ensure laser terminals point toward different spacecraft. As illustrated in Figure 1, assuming the designed communication topology, to complete the communication task, the spacecraft needs to use three laser terminals over a future period to point toward Satellites 1–3, respectively. Specifically, Satellite 1 resides in the same orbit with a forward orbital phase, while Satellites 2 and 3 are located in adjacent orbits with backward orbital phases.

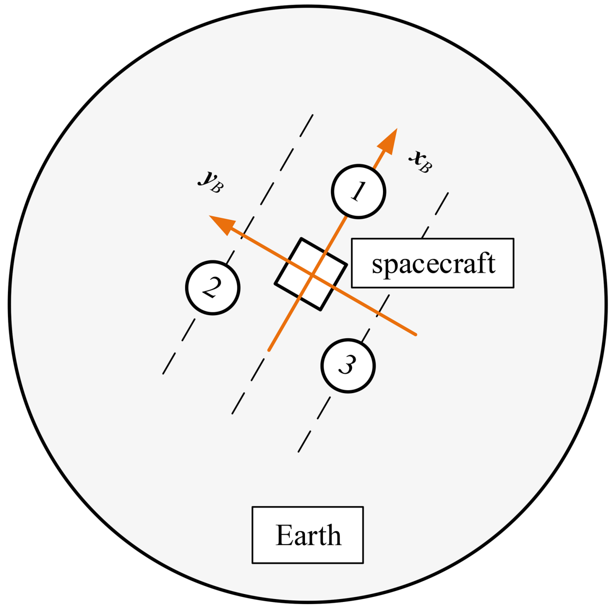

Figure 1 (Color online) Schematic diagram of the spacecraft orbit.

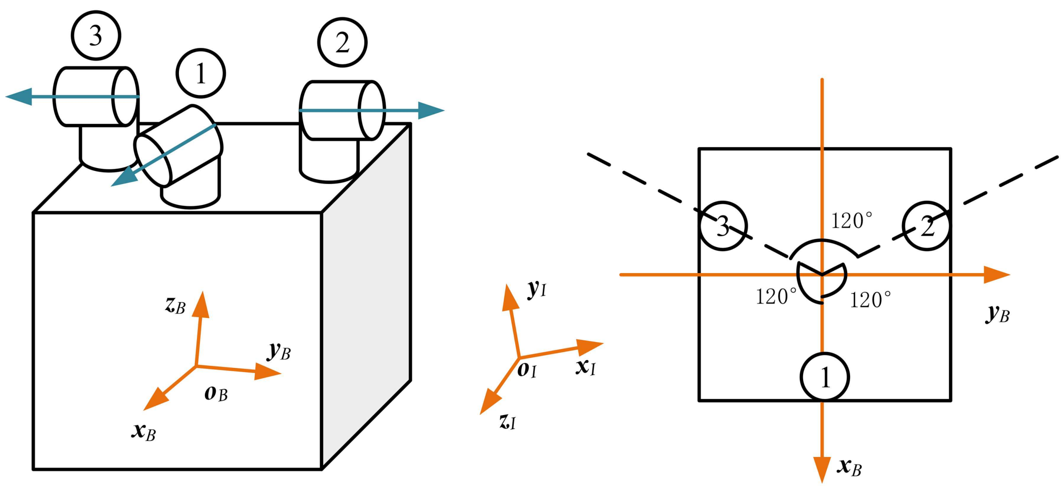

Figure 2 (Color online) Spacecraft model with three laser terminals.

The spacecraft model, illustrated in Figure 2, comprises a central platform and three laser terminals. Each terminal possesses two rotational degrees of freedom: yaw and pitch. The mounting planes of the three laser terminals are parallel to the $x_B O_B y_B$ plane of the spacecraft body frame F_B, and the angle between the terminals is 120°. The attitude of the spacecraft platform is denoted by R. The rotation angles of the three terminals are their yaw and pitch angles, denoted by $\theta_i \in [-90°, 90°]$ and $\varphi_i \in [-15°, 15°]$ for $i = 1, 2, 3$, respectively. The positive value is counterclockwise.

The objective of this work is to design smooth and dynamically feasible trajectories for the platform attitude ($\boldsymbol{R}(t)$) and the terminal joint angles ($\theta_i(t)$ and $\varphi_i(t)$) that simultaneously satisfy terminal angle constraints, terminal pointing constraints, and angular velocity constraints, given that the initial platform attitude, initial rotation angle, and terminal target pointing are known.

To achieve this objective, we propose a hierarchical attitude planning framework. As shown in Figure 3, in the first step, we design an attitude decomposition method based on terminal joint angle constraints and the maximum rotation angle of the platform to decompose the desired pointing into the platform attitude and joint angles.

The second step addresses constrained path search. We employ the MHA* algorithm to search the combined configuration space (platform attitude and joint angles) for a discrete sequence of feasible states. This sequence must satisfy all pointing constraints throughout the path and respect the discretization inherent in the search. The search is based on the geometric properties of SO(3), specifically utilizing the representation mapping rotation

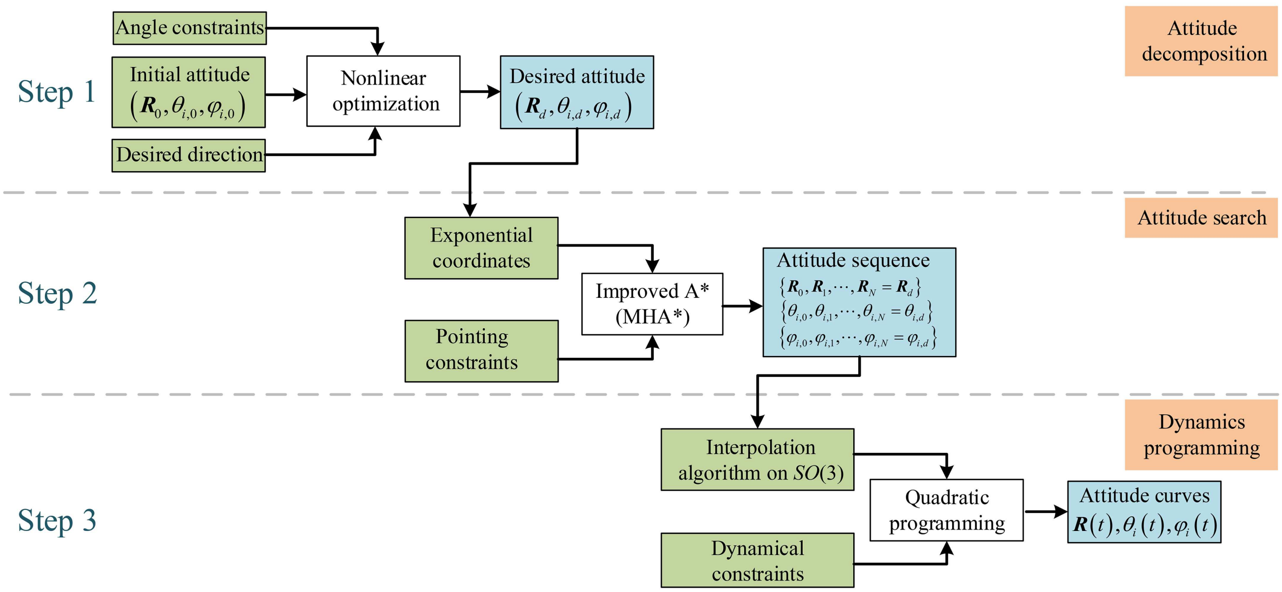

Figure 3 (Color online) Overview of multi-terminal hierarchical attitude planning algorithm.

matrices onto a 3D ball of radius π as characterized in Lemma 1.

The third step focuses on generating smooth and dynamically feasible trajectories. Given the discrete sequence of state nodes (platform attitude + terminal joint angles) from Step 2, we design continuous trajectories $R(t)$ and $\theta_i(t)$ connecting these nodes. Due to the non-Euclidean nature of the SO(3) manifold for the platform attitude, we apply a Lie group SO(3)-based trajectory planning method. This method transforms the trajectory generation problem into solving a sequence of QP problems.

3 Attitude decomposition and path search

3.1 Attitude decomposition

The functions of yaw and pitch angles taken for a unit vector $\boldsymbol{s} = [s_1, s_2, s_3]^{\mathrm{T}}$ are denoted $\mathrm{Azi}(\boldsymbol{s})$ and $\mathrm{Ele}(\boldsymbol{s})$, respectively,

$$\begin{aligned} \mathrm{Azi}(\boldsymbol{s}) &= \mathrm{atan2}(s_1, s_2), \\ \mathrm{Ele}(\boldsymbol{s}) &= \mathrm{atan2}\left(s_3, \sqrt{s_1^2 + s_2^2}\right). \end{aligned} \tag{3}$$

The known terminal desired pointing in the inertial system is $\boldsymbol{z}_j \in S^2, j = 1, 2, 3$, and the spacecraft platform attitude to be determined is $\boldsymbol{R}_d$. This desired attitude $\boldsymbol{R}_d$ should enable the terminal to satisfy the movable range of rotation (90° and 15°)

$$\begin{aligned} -\frac{1}{2}\pi \leqslant \mathrm{Azi}(\boldsymbol{R}_d \boldsymbol{z}_j) \leqslant \frac{1}{2}\pi, \\ -\frac{1}{12}\pi \leqslant \mathrm{Ele}(\boldsymbol{R}_d \boldsymbol{z}_j) \leqslant \frac{1}{12}\pi. \end{aligned} \tag{4}$$

Assuming that the spacecraft rotates with an upper limit of angular velocity $\Omega_{\max}$ and a maneuver time T, we would like the rotation angle corresponding to $\boldsymbol{R}_d$ to satisfy

$$d(\boldsymbol{R}_0, \boldsymbol{R}_d) \leqslant \Omega_{\max} T. \tag{5}$$

Using Lemma 4, we obtain $d(\boldsymbol{R}_0, \boldsymbol{R}_d) = d\big({\boldsymbol{R}_d}^{\mathrm{T}} \boldsymbol{R}_0, \boldsymbol{I}\big) = \big\|\log\big({\boldsymbol{R}_d}^{\mathrm{T}} \boldsymbol{R}_0\big)\big\|$. Based on Lemma 1, ${\boldsymbol{R}_d}^{\mathrm{T}} \boldsymbol{R}_0 \in \mathrm{SO}(3)$ can be expressed by the exponential coordinates: ${\boldsymbol{R}_d}^{\mathrm{T}} \boldsymbol{R}_0 = \boldsymbol{R}(\boldsymbol{\eta})$. Considering that the exponential coordinates $\boldsymbol{\eta}$ are bounded, Eq. (5) simplifies to

$$0 \leqslant \|\boldsymbol{\eta}\| \leqslant \min\{\pi, \Omega_{\max} T\}. \tag{6}$$

The objective of attitude decomposition is to find a minimum platform rotation while satisfying constraints (4) and (6), so $R_d(\boldsymbol{\eta})$ is obtained by solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\eta}} \quad & \|\boldsymbol{\eta}\|^2 \\ \text{s.t.} \quad & (4), (6). \end{aligned} \tag{7}$$

Thus, in attitude path planning, the starting point (known) is $\{\boldsymbol{R}_0, \theta_{i,0}, \varphi_{i,0}\}, i = 1, \ldots, 3$ and the end point is $\{\boldsymbol{R}_d = \boldsymbol{R}_0 \exp(\widehat{\boldsymbol{\eta}}), \theta_{i,d}, \varphi_{i,d}\}, i = 1, \ldots, 3$, where $\theta_{i,d} = \mathrm{Azi}(\boldsymbol{R}_d \boldsymbol{z}_j)$ and $\varphi_{i,d} = \mathrm{Ele}(\boldsymbol{R}_d \boldsymbol{z}_j)$. Attitude decomposition serves as the first step in the proposed planning framework, while dynamic-related constraints are considered in the third step. In spacecraft attitude control systems, attitude planning acts as a precursor to control. Generating smooth reference trajectories during planning helps improve control accuracy. It should be noted that attitude decomposition only provides the final desired attitude state—not the entire trajectory—and therefore does not directly affect the control process.

Remark 1. Attitude decomposition is motivated by two primary reasons. First, constellation design provides only the communication topology, which must be translated into specific platform attitudes and terminal joint angles to control the satellites during operational mode. Second, complex scenarios, such as post-deployment initialization, inter-plane link establishment, and topology switching, require coordinated platform-terminal maneuvers beyond the capability of terminal motion alone.

3.2 Attitude path search

In the second step, we utilize an improved A* algorithm [28] to plan a sequence of attitudes that satisfy the pointing constraints, considering the need to perform path search in a $3 + 3 \times 2 = 9$ dimensional space.

3.2.1 *Attitude node extension method and pointing constraints*

We divide the searched space S into two parts SO(3) and $S_1 \times S_2$:

$$\begin{aligned} & S = \mathrm{SO}(3) \times S_1 \times S_2, \\ & S_1 = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \\ & S_2 = \left[-\frac{\pi}{12}, \frac{\pi}{12}\right] \times \left[-\frac{\pi}{12}, \frac{\pi}{12}\right] \times \left[-\frac{\pi}{12}, \frac{\pi}{12}\right]. \end{aligned} \tag{8}$$

Since $S_1 \times S_2$ is a subset of the Euclidean space $\mathbb{R}^6$, the node extension method for angles $\theta_{i,k}$ and $\varphi_{i,k}$ are

$$\begin{aligned} \theta_{i,k+1} &= \begin{cases} \theta_{i,k} \pm \delta_h, & |\theta_{i,k} - \theta_{i,d}| > \delta_h, \\ \theta_{i,d}, & |\theta_{i,k} - \theta_{i,d}| \leqslant \delta_h, \end{cases} \\ \varphi_{i,k+1} &= \begin{cases} \varphi_{i,k} \pm \delta_h, & |\varphi_{i,k} - \varphi_{i,d}| > \delta_h, \\ \varphi_{i,d}, & |\varphi_{i,k} - \varphi_{i,d}| \leqslant \delta_h, \end{cases} \end{aligned} \tag{9}$$

where δ_S is the expansion step of the angles. Using Lemma 4, a path search from $\boldsymbol{R}_0$ to $\boldsymbol{R}_d$ is equivalent to a search from $\boldsymbol{I}$ to $\boldsymbol{R}_0{}^{\mathrm{T}} \boldsymbol{R}_d$. Further using the exponential coordinates, it can be equivalent to a search from vector $\mathbf{0} \in \mathbb{R}^3$ to vector $\boldsymbol{\eta} \in \mathbb{R}^3$, whereby we transform path search in a non-Euclidean space into one within a Euclidean space. Assuming that the current attitude node of the platform is $\boldsymbol{R}_k = \exp(\widehat{\boldsymbol{\eta}_k})$, the expansion of the node is designed as follows:

$$\boldsymbol{R}_{k+1} = \begin{cases} \exp\left(\delta_{\mathrm{SO}(3)} \widehat{\boldsymbol{\kappa}}\right) \exp\left(\widehat{\boldsymbol{\eta}}_k\right), & d\left(\boldsymbol{R}_k, \boldsymbol{R}_{k+1}\right) \geqslant \delta_{\mathrm{SO}(3)}, \\ \boldsymbol{R}_0^{\top} \boldsymbol{R}_d, & d\left(\boldsymbol{R}_k, \boldsymbol{R}_{k+1}\right) < \delta_{\mathrm{SO}(3)}, \end{cases} \tag{10}$$

where $\delta_{\mathrm{SO}(3)}$ is the expansion step size and $\boldsymbol{\kappa} \in S^2$ is the unit vector generated using uniform distribution. In the experiments of this paper, six unit vectors are generated per expansion.

In this paper, we consider the existence of forbidden pointing constraints and mandatory pointing constraints. The forbidden pointing constraints are the need to avoid the direction of bright celestial bodies, such as the Sun, for the pointing of the laser terminal during attitude maneuvers. As shown in Figure 4, these forbidden directions are known under the inertial frame F_I and denoted as $\boldsymbol{r}_{1,j} \in S^2, j = 1, \ldots, p$. The pointing of the terminal in the

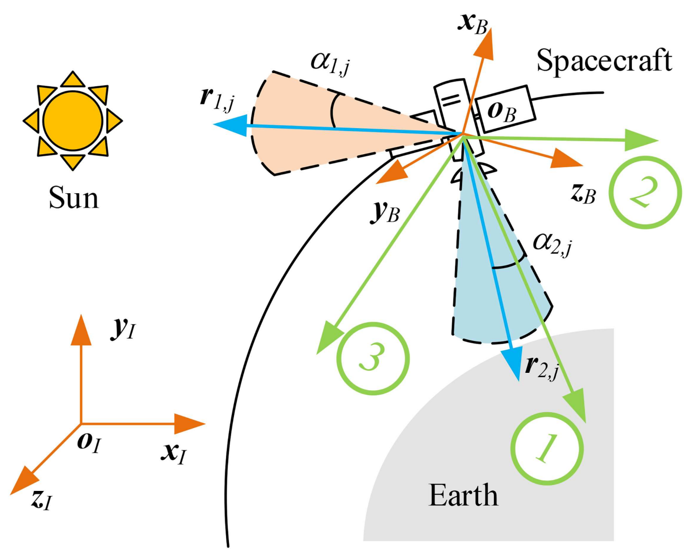

Figure 4 (Color online) Schematic diagram of spacecraft pointing constraints.

inertial frame should have an angle with $\boldsymbol{r}_{1,j} \in S^2$ greater than the corresponding constraint angle $\alpha_{1,j}$, which is therefore required to be satisfied by the extended attitude node

$$\begin{aligned}
&{\boldsymbol{r}_{1,j}}^{\mathrm{T}} \boldsymbol{R}_{k+1} \boldsymbol{v}\left(\theta_{1,k+1}, \varphi_{1,k+1}\right) \leqslant \cos \alpha_{1,j}, \\
&{\boldsymbol{r}_{1,j}}^{\mathrm{T}} \boldsymbol{R}_{k+1} \boldsymbol{v}\left(\theta_{2,k+1}+\frac{2\pi}{3}, \varphi_{2,k+1}\right) \leqslant \cos \alpha_{1,j}, \\
&{\boldsymbol{r}_{1,j}}^{\mathrm{T}} \boldsymbol{R}_{k+1} \boldsymbol{v}\left(\theta_{3,k+1}-\frac{2\pi}{3}, \varphi_{3,k+1}\right) \leqslant \cos \alpha_{1,j}.
\end{aligned} \tag{11}$$

The mandatory pointing constraint means that the direction of the laser terminal should always be maintained within a fixed angle with the given direction. Taking terminal 1 as an example, assuming that the angle between it and the given direction $\boldsymbol{r}_{2,j} \in S^2$ is always less than $\alpha_{2,j}$, the constraint on the node can be written as

$${\boldsymbol{r}_{2,j}}^{\mathrm{T}} \boldsymbol{R}_{k+1} \boldsymbol{v}\left(\theta_{1,k+1}, \varphi_{1,k+1}\right) \geqslant \cos \alpha_{2,j}, \tag{12}$$

where $\boldsymbol{v}(a,b) = [v_x(a,b), v_y(a,b), v_z(a,b)]^{\mathrm{T}} \in S^2$ is a function that calculates the direction of pointing under this system based on the yaw and pitch angles of the terminal, i.e.,

$$\begin{aligned}
v_x(a,b) &= \sin\left(\frac{\pi}{2}-b\right)\cos(a), \\
v_y(a,b) &= \sin\left(\frac{\pi}{2}-b\right)\sin(a), \\
v_z(a,b) &= \cos\left(\frac{\pi}{2}-b\right).
\end{aligned} \tag{13}$$

3.2.2 *Path search based on multiple heuristics A**

The A* algorithm is a heuristic graph search algorithm that centers on rating the priority of a node $\mathcal{N}$ based on the value of the following cost function:

$$f(\mathcal{N}) = g(\mathcal{N}) + h(\mathcal{N}), \tag{14}$$

where $g(\mathcal{N})$ is the actual cost from the starting point to the current node $\mathcal{N}$ and $h(\mathcal{N})$ is the estimated cost from the current node to the goal. Let $h^*(\mathcal{N})$ denote the actual cost from the current node to the goal. When $h(\mathcal{N}) \leqslant h^*(\mathcal{N})$, it is said that $h(\mathcal{N})$ is admissible, in which case the A* algorithm gives the shortest path. However, there are difficulties in designing a suitable heuristic function in high-dimensional, non-Euclidean spaces, and thus traditional A* algorithms do not work well on such problems. Considering the need for path planning in 9-dimensional space ($S = \mathrm{SO}(3) \times S_1 \times S_2$) in this paper, we apply a multi-heuristic A* algorithm, the pseudo-code of which is described in [28]. In MHA*, $n+1$ different heuristic functions $h_0, \ldots, h_n$ are designed, where only h_0 is required to be admissible. Using these $n+1$ heuristic functions, the $n+1$ searches run simultaneously and independently, where each search uses its own priority queue. Two parameters, $w_1 \geqslant 1$ and $w_2 \geqslant 1$, are also introduced in this algorithm to control the algorithm's sub-optimality bounds, where w_1 is the coefficient of the heuristic term in each search, and w_2 is a factor used to prioritize inadmissible searches $(h_1, \ldots, h_n)$. In this paper, we design 3 heuristic functions and choose to use the rotation angle as h_0, the Euclidean distance, and the Manhattan distance as h_1 and h_2. For the k-th node $\mathcal{N}_k = (\boldsymbol{R}_k, \theta_{i,k}, \varphi_{i,k}) = (\exp(\widehat{\boldsymbol{\eta}_k}), \theta_{i,k}, \varphi_{i,k})$, the heuristic term is computed as follows:

$$\begin{aligned}
h_0(\mathcal{N}_k) =& d(\boldsymbol{R}_k, \boldsymbol{R}_0^{\mathrm{T}}\boldsymbol{R}_d) + \sum_{i=1}^{3} |\theta_{i,k} - \theta_{i,d}| + \sum_{i=1}^{3} |\varphi_{i,k} - \varphi_{i,d}| \\
=& \left\| \log\left({\boldsymbol{R}_k}^{\mathrm{T}} {\boldsymbol{R}_0}^{\mathrm{T}} \boldsymbol{R}_d \right) \right\| + \sum_{i=1}^{3} (|\theta_{i,k} - \theta_{i,d}| + |\varphi_{i,k} - \varphi_{i,d}|), \\
h_1(\mathcal{N}_k) =& \left\| \boldsymbol{\eta}_k - \log\left({\boldsymbol{R}_0}^{\mathrm{T}} \boldsymbol{R}_d \right) \right\| + \left(\sum_{i=1}^{3} (\theta_{i,k} - \theta_{i,d})^2 + (\varphi_{i,k} - \varphi_{i,d})^2 \right)^{\frac{1}{2}}, \\
h_2(\mathcal{N}_k) =& \sum_{i=1}^{3} \left| (\boldsymbol{\eta}_k)_i - \left(\log\left({\boldsymbol{R}_0}^{\mathrm{T}} \boldsymbol{R}_d \right) \right)_i \right| + \sum_{i=1}^{3} (|\theta_{i,k} - \theta_{i,d}| + |\varphi_{i,k} - \varphi_{i,d}|).
\end{aligned} \tag{15}$$

The actual cost when a node expands is determined by the step size

$$g(\mathcal{N}_k) = g(\mathcal{N}_{k-1}) + \delta_{\mathrm{SO}(3)} + 6\delta_h. \tag{16}$$

In summary, the flowchart of attitude path planning is shown in Figure 5.

Remark 2. The MHA* algorithm improves the poor adaptability of traditional A* algorithms in high-dimensional spaces by incorporating multiple heuristic functions. Its theoretical properties are thoroughly discussed in [28], which also presents numerical experiments on 12-dimensional manipulator planning and 30-dimensional navigation problems, demonstrating its strong adaptability in high-dimensional planning. When additional terminals are installed, each terminal introduces two extra degrees of freedom. The MHA* algorithm remains effective in solving such problems.

4 Attitude dynamics planning

One difficulty in dynamical planning is that high degrees of freedom are required to fit attitude sequences when the sequences are too long. We therefore establish an optimization method for the simplest sequence containing only two nodes and then discuss some improvement techniques in the simulation section. As shown in Figure 6, after the attitude planning in Subsection 3.2, the attitude sequences $\{\theta_{i,0}, \ldots, \theta_{i,m} = \theta_{i,d}\}$, $\{\varphi_{i,0}, \ldots, \varphi_{i,m} = \varphi_{i,d}\}$, and $\left\{I, \ldots, {R_0}^{\mathrm{T}} R_d\right\} \Leftrightarrow \{R_0, \ldots, R_m = R_d\}$ are obtained, and the task of this section is to perform the dynamics planning to obtain the angle curves $\theta_i(t), t \in [0,T]$, $\varphi_i(t), t \in [0,T]$, and the attitude curve $R(t), t \in [0,T]$, which are C^2. Since a fixed step size is used to generate nodes, the total time T is divided equally in the dynamics planning.

4.1 Dynamics planning of laser terminals

Since yaw angles and pitch angles differ only in their symbols, we will use the yaw angle θ_i as an example to present the dynamics planning method for joint angles. In the time interval $[t_k, t_{k+1}]$, $\theta_{i,k}$ is assumed to be of the following form:

$$\widetilde{\theta}_{i,k}(t) = \sum_{k=0}^{l} a_{i,k,j} h_{j,k}(t) = (\boldsymbol{a}_{i,k})^{\mathrm{T}} \boldsymbol{h}_k(t), \tag{17}$$

where $\boldsymbol{a}_{i,k} = [a_{i,k,0}, \ldots, a_{i,k,l}]^{\mathrm{T}}$, $\boldsymbol{h}_k(t) = [h_{0,k}(t), \ldots, h_{l,k}(t)]^{\mathrm{T}}$ and $h_{j,k}(t)$ is the basis function. $\widetilde{\theta}_{i,k}(t)$ should be equal to two nodes at the left and right endpoints, and the first and second order derivatives at the left endpoint should be equal to those of the previous segment. Thus the equality constraints on $\widetilde{\theta}_{i,k}(t)$ are

$$\widetilde{\theta}_{i,k}(t_k) = \theta_{i,k},$$

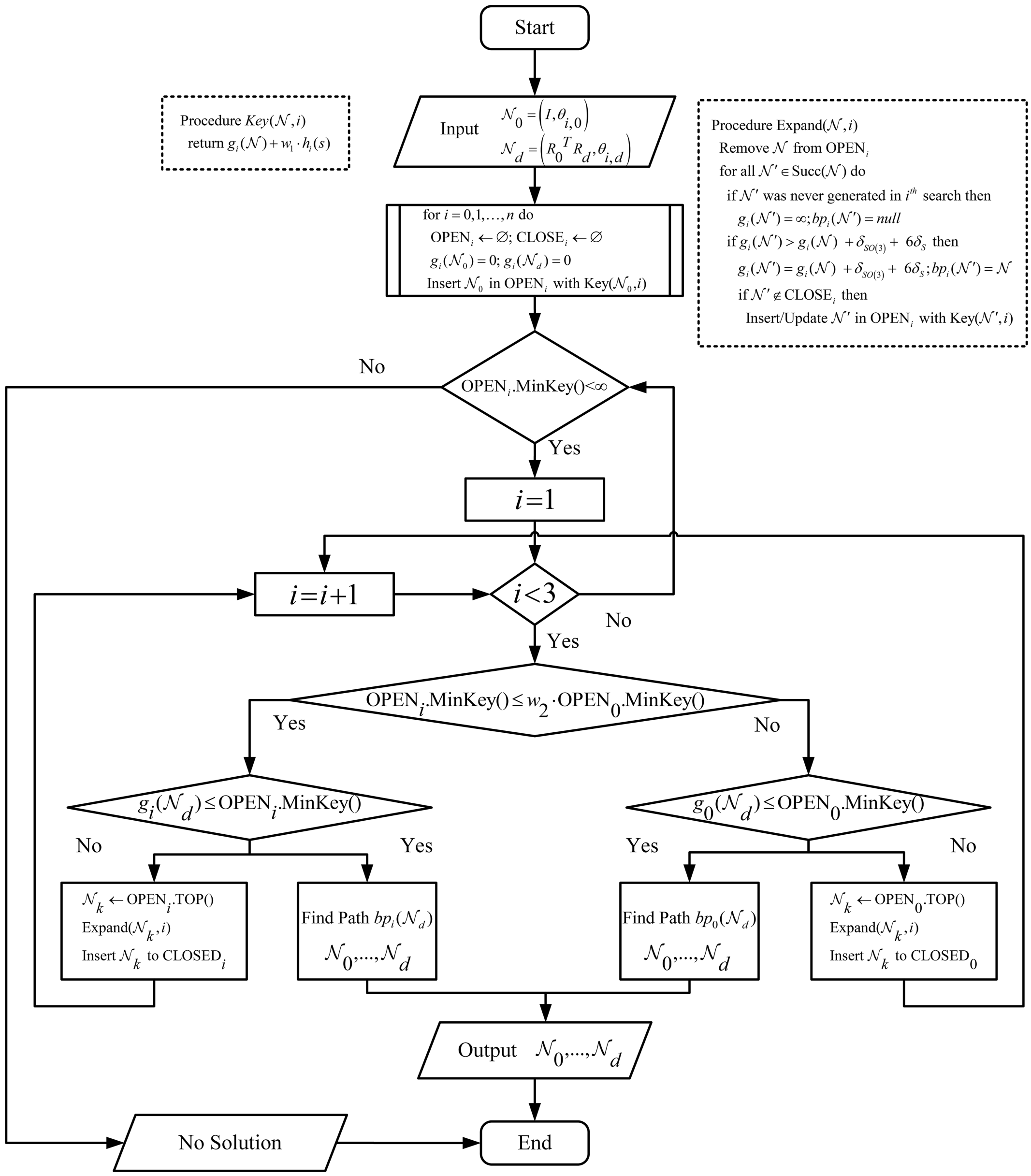

Figure 5 Flowchart of attitude path search.

$$\begin{aligned}
&\widetilde{\theta}_{i,k}(t_{k+1}) = \theta_{i,k+1},\\
&\dot{\widetilde{\theta}}_{i,k}(t_k) = \begin{cases} \dot{\widetilde{\theta}}_{i,k-1}(t_k), & k \neq 0,\\ 0, & k = 0,\end{cases}\\
&\ddot{\widetilde{\theta}}_{i,k}(t_k) = \begin{cases} \ddot{\widetilde{\theta}}_{i,k-1}(t_k), & k \neq 0,\\ 0, & k = 0.\end{cases}
\end{aligned} \tag{18}$$

Additional constraints are required for the last segment

$$\begin{aligned}
\dot{\widetilde{\theta}}_{i,k}(T) &= 0,\\
\ddot{\widetilde{\theta}}_{i,k}(T) &= 0.
\end{aligned} \tag{19}$$

Furthermore, since there is an upper limit on the angular velocity of rotation and its derivatives, the inequality constraint for $\widetilde{\theta}_{i,k}(t)$ is that the first and second order derivatives at the right endpoint $(t = t_{k+1})$ are less than the

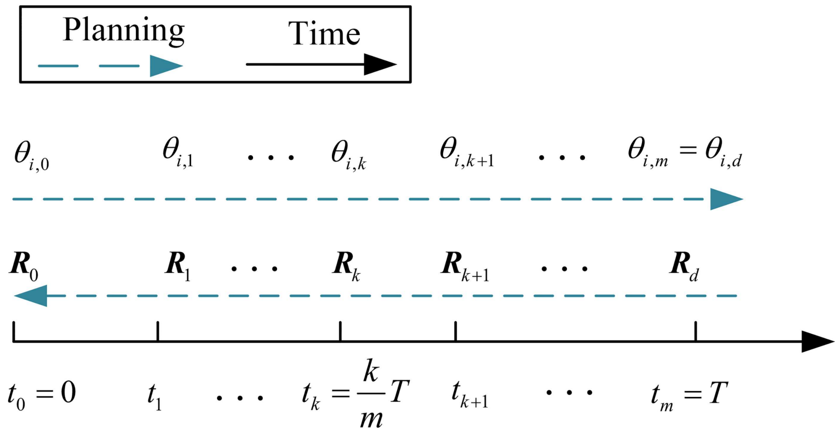

Figure 6 (Color online) Sequence of attitudes and direction of planning.

given values

$$\begin{aligned} &-\xi_1 \leqslant \dot{\widetilde{\theta}}_{i,k}(t_{k+1}) \leqslant \xi_1, \\ &-\zeta_1 \leqslant \ddot{\widetilde{\theta}}_{i,k}(t_{k+1}) \leqslant \zeta_1. \end{aligned} \tag{20}$$

The derivatives of each order at the left endpoint ($t = t_k$) remain the same as at the right endpoint in the previous segment, so the upper bound constraint is also satisfied.

Since the torque of rotation is directly related to the second-order derivative of $\widetilde{\theta}_{i,k}(t)$, $\widetilde{\theta}_{i,k}(t)$ is obtained by solving the following quadratic programming problem:

$$\begin{aligned} \min_{\boldsymbol{a}_{i,k}} \quad & \frac{1}{2}\int_{t_k}^{t_k} \ddot{\widetilde{\theta}}{}_{i,k}^{2}(t)\,\mathrm{d}t = \frac{1}{2}(\boldsymbol{a}_{i,k})^{\mathrm{T}}\boldsymbol{P}_{i,k}\boldsymbol{a}_{i,k} \\ \text{s.t.} \quad & (18), (19), (20), \end{aligned} \tag{21}$$

where

$$\begin{aligned} \boldsymbol{P}_{i,k} &= \int_{t_k}^{t_{k+1}} \boldsymbol{h}_k(t)\,\boldsymbol{h}_k{}^{\mathrm{T}}(t)\,\mathrm{d}t \\ &= \begin{bmatrix} \int_{t_k}^{t_{k+1}} h_{0,k}h_{0,k}\mathrm{d}t & \cdots & \int_{t_k}^{t_{k+1}} h_{0,k}h_{l,k}\mathrm{d}t \\ \vdots & \vdots & \vdots \\ \int_{t_k}^{t_{k+1}} h_{l,k}h_{0,k}\mathrm{d}t & \cdots & \int_{t_k}^{t_{k+1}} h_{l,k}h_{l,k}\mathrm{d}t \end{bmatrix}. \end{aligned} \tag{22}$$

Constraints (18)–(20) are linear and can be written in the following form:

$$\begin{aligned} &\boldsymbol{A}_{i,k}\boldsymbol{a}_{i,k} = \boldsymbol{b}_{i,k}, \\ &-\boldsymbol{c}_1 \leqslant \boldsymbol{G}_{i,k}\boldsymbol{a}_{i,k} \leqslant \boldsymbol{c}_1, \end{aligned} \tag{23}$$

where

$$\begin{aligned} \boldsymbol{A}_{i,k} &= \begin{cases} \left[h_k{}^{\mathrm{T}}(t_k), h_k{}^{\mathrm{T}}(t_{k+1}), \dot{h}_k^{\mathrm{T}}(t_k), \ddot{h}_k^{\mathrm{T}}(t_k)\right]^{\mathrm{T}}, & k \neq m, \\ \left[h_k{}^{\mathrm{T}}(t_k), h_k{}^{\mathrm{T}}(t_{k+1}), \dot{h}_k^{\mathrm{T}}(t_k), \ddot{h}_k^{\mathrm{T}}(t_k), \dot{h}_k^{\mathrm{T}}(t_{k+1}), \ddot{h}_k^{\mathrm{T}}(t_{k+1})\right]^{\mathrm{T}}, & k = m, \end{cases} \\ \boldsymbol{b}_{i,k} &= \begin{cases} [\theta_{i,k}, \theta_{i,k+1}, 0, 0]^{\mathrm{T}}, & k = 0, \\ \left[\theta_{i,k}, \theta_{i,k+1}, \dot{\widetilde{\theta}}_{i,k-1}(t_k), \ddot{\widetilde{\theta}}_{i,k-1}(t_k)\right]^{\mathrm{T}}, & 0 < k < m-1, \\ \left[\theta_{i,k}, \theta_{i,k+1}, \dot{\widetilde{\theta}}_{i,k-1}(t_k), \ddot{\widetilde{\theta}}_{i,k-1}(t_k), 0, 0\right]^{\mathrm{T}}, & k = m-1, \end{cases} \\ \boldsymbol{G}_{i,k} &= \left[\dot{h}_k^{\mathrm{T}}(t_{k+1}), \ddot{h}_k^{\mathrm{T}}(t_{k+1})\right]^{\mathrm{T}}, \\ \boldsymbol{c}_1 &= [\xi_1, \zeta_1]^{\mathrm{T}}. \end{aligned} \tag{24}$$

Remark 3. In the quadratic programming problem (21), the existence of a globally optimal solution is determined by whether $\boldsymbol{P}_{i,k}$ is positive definite or not. The positive definiteness of $\boldsymbol{P}_{i,k}$ is then determined by the basis functions; e.g., for common polynomial basis functions, $\boldsymbol{P}_{i,k}$ is always positive semi-definite, and hence there exists a globally optimal solution.

4.2 Dynamics planning of the platform

4.2.1 *Interpolation on SO(3)*

Interpolating on SO(3) is not as straightforward as interpolating in Euclidean space; hence, in this paper, we use the algorithm in [29, 30] to obtain the attitude curve $\boldsymbol{R}(t)$. Also, unlike the planning of angles $\theta_{i,k}(t)$ which is done backwards from the interval $[t_0, t_1]$, in optimizing the curve $\boldsymbol{R}(t)$ it is done forwards from the last interval $[t_{m-1}, t_m = T]$ as shown in Figure 6. On the interval $[t_k, t_{k+1}]$, the attitude curve $\boldsymbol{R}_{k,k+1}$ is designed as follows:

$$\begin{aligned}
&\boldsymbol{R}_{k,k+1}(t) = \boldsymbol{R}_k \gamma_k(t), \\
&\gamma_k(t) = \boldsymbol{U}_k(t) \exp\left(\boldsymbol{\beta}_k(t) - \boldsymbol{X}_k(t) - \boldsymbol{I}\right) \boldsymbol{W}_k^{\mathrm{T}}(t), \\
&\boldsymbol{X}_k(t) = (t - t_k)\,\boldsymbol{\Omega}_k, \\
&\boldsymbol{U}_k(t) = \exp\left((t - t_k)\frac{\boldsymbol{\Omega}_k}{2}\right), \\
&\boldsymbol{W}_k(t) = \exp\left(-(t - t_k)\frac{\boldsymbol{\Omega}_k}{2}\right), \\
&\boldsymbol{\Omega}_k = \frac{\log\left(\boldsymbol{R}_k^{-1}\boldsymbol{R}_{k+1}\right)}{t_{k+1} - t_k},
\end{aligned} \tag{25}$$

where $\boldsymbol{\beta}_k(t) - \boldsymbol{I} \in \mathrm{so}(3)$ is the interpolation curve to be optimized. Since so(3) and $\mathbb{R}^3$ are isomorphic, the above interpolation algorithm both transforms the complex interpolation problem on SO(3) into a simple interpolation problem on $\mathbb{R}^3$ and has many elegant properties. With slight symbol abuse, we fit the vector $(\boldsymbol{\beta}_k(t) - \boldsymbol{I})^{\vee} \in \mathbb{R}^3$ in the same way as in (17), i.e.,

$$(\boldsymbol{\beta}_k(t) - \boldsymbol{I})_i^{\vee} = \sum_{j=0}^{l'} a_{i,k,j} h_{j,k}(t) = (\boldsymbol{a}_{i,k})^{\mathrm{T}} \boldsymbol{h}_k(t), i = 1, \ldots, 3. \tag{26}$$

According to [18], the first and second order derivatives of the curve $\boldsymbol{\gamma}_k(t)$ are

$$\begin{aligned}
\dot{\gamma}_k(t) =& \frac{\boldsymbol{\Omega}_k}{2}\gamma_k(t) + \gamma_k(t)\frac{\boldsymbol{\Omega}_k}{2} + \gamma_k(t)\left(\boldsymbol{W}_k(t)\boldsymbol{D}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k\right)^{\wedge}, \\
\ddot{\gamma}_k(t) =& \gamma_k(t)\left(-\frac{\boldsymbol{\Omega}_k}{2}\boldsymbol{W}_k(t)\boldsymbol{D}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k + \boldsymbol{W}_k(t)\dot{\boldsymbol{D}}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k + \boldsymbol{W}_k(t)\boldsymbol{D}(\boldsymbol{\eta}_k)\ddot{\boldsymbol{\eta}}_k\right)^{\wedge} \\
&+ \frac{\boldsymbol{\Omega}_k}{2}\dot{\gamma}_k(t) + \dot{\gamma}_k(t)\frac{\boldsymbol{\Omega}_k}{2} + \dot{\gamma}_k(t)\left(\boldsymbol{W}_k(t)\boldsymbol{D}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k\right)^{\wedge},
\end{aligned} \tag{27}$$

where $\widehat{\boldsymbol{\eta}_k(t)} = \boldsymbol{\beta}_k(t) - \boldsymbol{X}_k(t) - \boldsymbol{I}, t \in [t_k, t_{k+1}]$. The tangent operator $\boldsymbol{D}(\boldsymbol{\eta}_k)$ is defined as described in Lemma 3. The angular velocity and its derivative are

$$\begin{aligned}
&\widehat{\boldsymbol{\omega}_k}(t) = \boldsymbol{\gamma}_k{}^{\mathrm{T}}(t)\dot{\boldsymbol{\gamma}}_k(t), \\
&\widehat{\dot{\boldsymbol{\omega}}_k}(t) = \boldsymbol{\gamma}_k^{\mathrm{T}}(t)\ddot{\boldsymbol{\gamma}}_k(t) - \left(\widehat{\boldsymbol{\omega}_k}(t)\right)^2.
\end{aligned} \tag{28}$$

It follows that the angular velocity in vector form and its derivatives are

$$\begin{aligned}
\boldsymbol{\omega}_k(t) =& \boldsymbol{\gamma}_k^{\mathrm{T}}(t)\frac{\boldsymbol{\Omega}_k^{\vee}}{2} + \frac{\boldsymbol{\Omega}_k^{\vee}}{2} + \boldsymbol{W}_k(t)\boldsymbol{D}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k, \\
\dot{\boldsymbol{\omega}}_k(t) =& -\widehat{\boldsymbol{\omega}_k}(t)\boldsymbol{\gamma}_k^{\mathrm{T}}(t)\frac{\boldsymbol{\Omega}_k{}^{\vee}}{2} - \frac{\boldsymbol{\Omega}_k}{2}\boldsymbol{W}_k(t)\boldsymbol{D}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}} + \boldsymbol{W}_k(t)\dot{\boldsymbol{D}}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k + \boldsymbol{W}_k(t)\boldsymbol{D}(\boldsymbol{\eta}_k)\ddot{\boldsymbol{\eta}}_k,
\end{aligned} \tag{29}$$

where the derivative of the tangent operator $\boldsymbol{D}(\boldsymbol{\eta}_k)$ is computed in [18, 26].

4.2.2 *Constraints in platform planning*

After giving some properties of the interpolating curve (25), we then give some constraints that should be satisfied if we want the curve to be C^2. At the left and right endpoints of the interval $[t_k, t_{k+1}]$, the curve $\boldsymbol{R}_{k,k+1}(t)$ should be satisfied

$$\begin{aligned}&\boldsymbol{R}_{k,k+1}(t_k) = \boldsymbol{R}_k,\\&\boldsymbol{R}_{k,k+1}(t_{k+1}) = \boldsymbol{R}_{k+1}.\end{aligned} \tag{30}$$

According to [18], the above constraint is equivalent to

$$\boldsymbol{\eta}_k(t_k) = \boldsymbol{\eta}_k(t_{k+1}) = \boldsymbol{0}. \tag{31}$$

Substituting (26) gives

$$\begin{aligned}&(\boldsymbol{a}_{i,k})^{\mathrm{T}}\boldsymbol{h}_k(t_k) = (\boldsymbol{X}_k^{\vee})_i(t_k),\\&(\boldsymbol{a}_{i,k})^{\mathrm{T}}\boldsymbol{h}_k(t_{k+1}) = (\boldsymbol{X}_k^{\vee})_i(t_{k+1}),\ i = 1,\ldots,3.\end{aligned} \tag{32}$$

Since the platform is planned from back to front, the constraints formed by the continuity of the angular velocity and its derivatives are

$$\begin{aligned}&\boldsymbol{\omega}_k(t_{k+1}) = \begin{cases}\boldsymbol{\omega}_{k+1}(t_{k+1}), & k \neq m-1,\\ 0, & k = m-1,\end{cases}\\&\dot{\boldsymbol{\omega}}_k(t) = \begin{cases}\dot{\boldsymbol{\omega}}_{k+1}(t_{k+1}), & k \neq m-1,\\ 0, & k = m-1.\end{cases}\end{aligned} \tag{33}$$

By substituting (31) into (29) and using the result in [18], we have

$$\begin{aligned}\boldsymbol{\omega}_k(t_k) =&\frac{\boldsymbol{\Omega}_k^{\vee}}{2} + \frac{\boldsymbol{\Omega}_k^{\vee}}{2} + \dot{\boldsymbol{\eta}}_k(t_k)\\=&\boldsymbol{\Omega}_k^{\vee} + \dot{\boldsymbol{\eta}}_k(t_k),\\\dot{\boldsymbol{\omega}}_k(t_k) =&\frac{\boldsymbol{\Omega}_k}{2}\boldsymbol{\omega}_k(t_k) - \frac{\boldsymbol{\Omega}_k}{2}\dot{\boldsymbol{\eta}}_k(t_k) + \ddot{\boldsymbol{\eta}}_k(t_k)\\=&\ddot{\boldsymbol{\eta}}_k(t_k).\end{aligned} \tag{34}$$

When $t = t_{k+1}$, there is $\boldsymbol{\gamma}_k(t_{k+1}) = \boldsymbol{R}_k^{\mathrm{T}}\boldsymbol{R}_{k+1} = \exp((t_{k+1} - t_k)\boldsymbol{\Omega}_k)$, and hence we have

$$\begin{aligned}\boldsymbol{\omega}_k(t_{k+1}) =&\boldsymbol{\Omega}_k^{\vee} + \exp\left((t_{k+1} - t_k)\frac{\boldsymbol{\Omega}_k}{2}\right)\dot{\boldsymbol{\eta}}_k(t_{k+1})\\=&\exp\left((t_{k+1} - t_k)\frac{\boldsymbol{\Omega}_k}{2}\right)(\boldsymbol{\Omega}_k^{\vee} + \dot{\boldsymbol{\eta}}_k(t_{k+1})),\\\dot{\boldsymbol{\omega}}_k(t_{k+1}) =&\frac{\boldsymbol{\Omega}_k}{2}\widehat{\boldsymbol{\omega}_k}(t_{k+1}) - \frac{\boldsymbol{\Omega}_k}{2}\dot{\boldsymbol{\eta}}_k(t_{k+1}) + \exp\left((t_{k+1} - t_k)\frac{\boldsymbol{\Omega}_k}{2}\right)\ddot{\boldsymbol{\eta}}_k(t_{k+1})\\=&\exp\left((t_{k+1} - t_k)\frac{\boldsymbol{\Omega}_k}{2}\right)\ddot{\boldsymbol{\eta}}_k(t_{k+1}).\end{aligned} \tag{35}$$

Substituting (26) and (35) for (33) yields

$$\begin{aligned}&(\boldsymbol{a}_{i,k})^{\mathrm{T}}\dot{\boldsymbol{h}}_k(t_{k+1}) = \begin{cases}[\boldsymbol{W}_k(t_{k+1})\boldsymbol{\omega}_{k+1}(t_{k+1})]_i, & k \neq m-1,\\ 0, & k = m-1,\end{cases}\\&(\boldsymbol{a}_{i,k})^{\mathrm{T}}\ddot{\boldsymbol{h}}_k(t_{k+1}) = \begin{cases}[\boldsymbol{W}_k(t_{k+1})\dot{\boldsymbol{\omega}}_{k+1}(t_{k+1})]_i, & k \neq m-1,\\ 0, & k = m-1,\end{cases}\ i = 1,\ldots,3.\end{aligned} \tag{36}$$

When $k = 0$, there is an additional equation constraint

$$\begin{cases}\boldsymbol{\omega}_0(t_0) = \boldsymbol{0},\\ \dot{\boldsymbol{\omega}}_0(t_0) = \boldsymbol{0},\end{cases} \Leftrightarrow \begin{cases}(\boldsymbol{a}_{i,0})^{\mathrm{T}}\dot{\boldsymbol{h}}_0(t_0) = \boldsymbol{0},\\ (\boldsymbol{a}_{i,0})^{\mathrm{T}}\ddot{\boldsymbol{h}}_0(t_0) = \boldsymbol{0}.\end{cases} \tag{37}$$

On the interval $[t_k, t_{k+1}]$, the right endpoint is used to satisfy the constraints of the equation due to the C^2 property, while the left endpoint is used for the constraints on the maximum value of the angular velocity and its derivatives

$$\begin{cases} -\xi_2 \leqslant \boldsymbol{\omega}_{i,k}(t_k) \leqslant \xi_2, \\ -\zeta_2 \leqslant \dot{\boldsymbol{\omega}}_{i,k}(t_k) \leqslant \zeta_2, \end{cases} \Leftrightarrow \begin{cases} -\xi_2 \leqslant (\boldsymbol{a}_{i,k})^{\mathrm{T}} \dot{\boldsymbol{h}}_k(t_k) \leqslant \xi_2, \\ -\zeta_2 \leqslant (\boldsymbol{a}_{i,k})^{\mathrm{T}} \ddot{\boldsymbol{h}}_k(t_k) \leqslant \zeta_2. \end{cases} \tag{38}$$

4.2.3 *Quadratic programming of the platform*

Using Lemma 2 and the triangle inequality, one obtains

$$\|\boldsymbol{\omega}_k(t)\| \leqslant \left\|{\boldsymbol{\Omega}_k}^{\vee}\right\| + \|\boldsymbol{D}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k\|. \tag{39}$$

From the forms of $\boldsymbol{D}(\boldsymbol{\eta})$ given in Lemma 3, it is clear that $\|\boldsymbol{D}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k\|$ is bounded

$$\begin{aligned} \|\boldsymbol{D}(\boldsymbol{\eta}_k)\dot{\boldsymbol{\eta}}_k\| &\leqslant \left\| \left(\boldsymbol{I} - \frac{1-\cos(\|\boldsymbol{\eta}_k\|)}{\|\boldsymbol{\eta}_k\|^2} \widehat{\boldsymbol{\eta}_k} + \frac{1-\sin(\|\boldsymbol{\eta}_k\|)/\|\boldsymbol{\eta}_k\|}{\|\boldsymbol{\eta}_k\|^2} \widehat{\boldsymbol{\eta}_k}^2 \right) \dot{\boldsymbol{\eta}}_k \right\| \\ &\leqslant \|\dot{\boldsymbol{\eta}}_k\| + \frac{1-\cos(\|\boldsymbol{\eta}_k\|)}{\|\boldsymbol{\eta}_k\|} \|\dot{\boldsymbol{\eta}}_k\| + \frac{\|\boldsymbol{\eta}_k\| - \sin(\|\boldsymbol{\eta}_k\|)}{\|\boldsymbol{\eta}_k\|} \|\dot{\boldsymbol{\eta}}_k\| \\ &\leqslant \left(2 + \frac{1-\cos(\|\boldsymbol{\eta}_k\|) - \sin(\|\boldsymbol{\eta}_k\|)}{\|\boldsymbol{\eta}_k\|} \right) \|\dot{\boldsymbol{\eta}}_k\|, \|\boldsymbol{\eta}_k\| \in (0, \pi). \end{aligned} \tag{40}$$

The function $f(x) = 2 + (1 - \cos x - \sin x)/x$ is bounded on the interval $[0, \pi]$, so it follows that

$$\|\omega_k(t)\| \leqslant \left\|{\boldsymbol{\Omega}_k}^{\vee}\right\| + m_f \|\dot{\boldsymbol{\eta}}_k\|, \tag{41}$$

where m_f is the maximum value of the function $f(x)$. Considering $\|\dot{\boldsymbol{\eta}}_k\| < \left\|\dot{\boldsymbol{\beta}}_k^{\vee}(t)\right\| + \left\|{\boldsymbol{\Omega}_k}^{\vee}\right\|$, we have

$$\|\boldsymbol{\omega}_k(t)\| \leqslant (1+m_f) \left\|{\boldsymbol{\Omega}_k}^{\vee}\right\| + m_f \left\|\dot{\beta}_k^{\vee}(t)\right\|. \tag{42}$$

Eq. (42) illustrates that decreasing $\|\dot{\boldsymbol{\beta}}_k^{\vee}(t)\|$ leads directly to attitude curve with smaller angular velocities. A similar analysis can be applied to the derivative of the angular velocity, i.e., decreasing $\|\dot{\boldsymbol{\beta}}_k^{\vee}(t)\|$ and $\|\ddot{\boldsymbol{\beta}}_k^{\vee}(t)\|$ results in attitude curve with smaller angular velocity derivatives, which implies lower control consumption. Therefore, we transform the attitude curve $\boldsymbol{R}_{k,k+1}(t)$ design into solving an optimization problem of the following form:

$$\begin{aligned} \min_{a_{i,k}} \quad & \frac{1}{2} \int_{t_k}^{t_{k+1}} \lambda_1 \left(\dot{\boldsymbol{\beta}}_k^{\vee}\right)^{\mathrm{T}} \dot{\boldsymbol{\beta}}_k^{\vee} + \lambda_2 \left(\ddot{\boldsymbol{\beta}}_k^{\vee}\right)^{\mathrm{T}} \ddot{\boldsymbol{\beta}}_k^{\vee} \mathrm{d}t \\ \text{s.t.} \quad & (32), (36), (37), (38). \end{aligned} \tag{43}$$

Next, we show how to rewrite the above optimization problem as a quadratic programming problem. Substituting (26) into (43) yields

$$\begin{aligned} \int_{t_k}^{t_{k+1}} \left(\dot{\boldsymbol{\beta}}_k^{\vee}\right)^{\mathrm{T}} \dot{\boldsymbol{\beta}}_k^{\vee} \mathrm{d}t &= \sum_{i=1}^{3} \int_{t_k}^{t_{k+1}} \left((\boldsymbol{a}_{i,k})^{\mathrm{T}} \dot{\boldsymbol{h}}_k(t)\right)^2 \mathrm{d}t = \sum_{i=1}^{3} {a_{i,k}}^{\mathrm{T}} \boldsymbol{P}_{k,1} a_{i,k}, \\ \int_{t_k}^{t_{k+1}} \left(\ddot{\boldsymbol{\beta}}_k^{\vee}\right)^{\mathrm{T}} \ddot{\boldsymbol{\beta}}_k^{\vee} \mathrm{d}t &= \sum_{i=1}^{3} \int_{t_k}^{t_{k+1}} \left((\boldsymbol{a}_{i,k})^{\mathrm{T}} \ddot{\boldsymbol{h}}_k(t)\right)^2 \mathrm{d}t = \sum_{i=1}^{3} {a_{i,k}}^{\mathrm{T}} \boldsymbol{P}_{k,2} a_{i,k}, \end{aligned} \tag{44}$$

where

$$\begin{aligned} \boldsymbol{P}_{k,1} &= \int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{h}_0(t) \\ \vdots \\ \dot{h}_{l'}(t) \end{bmatrix} \left[\dot{h}_0(t), \ldots, \dot{h}_{l'}(t)\right] \mathrm{d}t, \\ \boldsymbol{P}_{k,2} &= \int_{t_k}^{t_{k+1}} \begin{bmatrix} \ddot{h}_0(t) \\ \vdots \\ \ddot{h}_{l'}(t) \end{bmatrix} \left[\ddot{h}_0(t), \ldots, \ddot{h}_{l'}(t)\right] \mathrm{d}t. \end{aligned} \tag{45}$$

Thus, the objective function of the optimization is

$$\frac{1}{2}\int_{t_k}^{t_{k+1}} \lambda_1 \left(\dot{\boldsymbol{\beta}}_k^\vee\right)^{\mathrm{T}} \dot{\boldsymbol{\beta}}_k^\vee + \lambda_2 \left(\ddot{\boldsymbol{\beta}}_k^\vee\right)^{\mathrm{T}} \ddot{\boldsymbol{\beta}}_k^\vee \mathrm{d}t = \frac{1}{2}\boldsymbol{a}_k^{\mathrm{T}} \boldsymbol{P}_k \boldsymbol{a}_k, \tag{46}$$

where $\boldsymbol{a}_k = [\boldsymbol{a}_{1,k}^{\mathrm{T}}, \boldsymbol{a}_{2,k}^{\mathrm{T}}, \boldsymbol{a}_{3,k}^{\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{P}_k = \mathrm{diag}\,(\lambda_1 \boldsymbol{P}_{k,1} + \lambda_2 \boldsymbol{P}_{k,2}, \lambda_1 \boldsymbol{P}_{k,1} + \lambda_2 \boldsymbol{P}_{k,2}, \lambda_1 \boldsymbol{P}_{k,1} + \lambda_2 \boldsymbol{P}_{k,2})$. Similarly, the matrix $\boldsymbol{P}_k$ is positive definite or not depending on the basis functions. Let $\boldsymbol{H}(t) = \mathrm{diag}\left(\boldsymbol{h}_k^{\mathrm{T}}(t), \boldsymbol{h}_k^{\mathrm{T}}(t), \boldsymbol{h}_k^{\mathrm{T}}(t)\right)$. Constraints (32), (36), (37), and (38) can be rewritten as

$$\begin{aligned} &\boldsymbol{A}_k \boldsymbol{a}_k = \boldsymbol{b}_k, \\ &-\boldsymbol{c_2} \leqslant \boldsymbol{G}_k \boldsymbol{a}_k \leqslant \boldsymbol{c}_2, \end{aligned} \tag{47}$$

where

$$\begin{aligned} &\boldsymbol{A}_k = \begin{cases} \boldsymbol{H}_1, & k \neq 0, \\ \left[\boldsymbol{H}_1, \dot{\boldsymbol{H}}(t_k), \ddot{\boldsymbol{H}}(t_k)\right]^\top, & k = 0, \end{cases} \\ &\boldsymbol{H}_1 = \left[\boldsymbol{H}(t_k), \boldsymbol{H}(t_{k+1}), \dot{\boldsymbol{H}}(t_{k+1}), \ddot{\boldsymbol{H}}(t_{k+1})\right]^\top, \\ &\boldsymbol{b}_k = \begin{cases} \left[\boldsymbol{X}_k^\vee(t_k), \boldsymbol{X}_k^\vee(t_{k+1}), 0_{6\times 1}\right]^\top, & k = m-1, \\ \left[\boldsymbol{X}_k^\vee(t_k), \boldsymbol{X}_k^\vee(t_{k+1}), \boldsymbol{b}_1\right]^\top, & 0 < k < m-1, \\ \left[\boldsymbol{X}_k^\vee(t_k), \boldsymbol{X}_k^\vee(t_{k+1}), \boldsymbol{b}_1, 0_{6\times 1}\right]^\top, & k = 0, \end{cases} \\ &\boldsymbol{b}_1 = \boldsymbol{W}_k(t_{k+1})\left[\boldsymbol{\omega}_{k+1}(t_{k+1}), \dot{\boldsymbol{\omega}}_{k+1}(t_{k+1})\right]^\top, \\ &\boldsymbol{G}_k = \left[\dot{\boldsymbol{H}}(t_k), \ddot{\boldsymbol{H}}(t_k)\right]^\top, \\ &\boldsymbol{c}_2 = \left[\xi_2, \xi_2, \xi_2, \zeta_2, \zeta_2, \zeta_2\right]^\top. \end{aligned} \tag{48}$$

We have now established a complete multi-terminal hierarchical attitude planning algorithm. The proposed method is an efficient planning method, despite involving nonlinear optimization. At each step, we strive to achieve the objective simply. In the attitude decomposition step, the nonlinear optimization objective function is simple and has few constraints. In contrast, the optimization problems established in [17, 18] require simultaneous consideration of constraints such as direction, angular velocity, and torque. In the path search step, the multi-heuristic A* algorithm applied is an improved algorithm for searching in high-dimensional spaces. In dynamic planning, the problem of generating smooth curves is transformed into multiple QP problems without losing the physical context.

5 Numerical simulation

In this section, we perform three simulations with random initialization to illustrate the effectiveness of the proposed method. We assume that both the spacecraft and the target spacecraft toward which its terminal needs to be pointed are located in a circular orbit of 500 km with an inclination of 10°. The ascending nodes of the four satellites have longitudes of 10°, 10°, 5°, and 13°. The true anomalies are 20°, 25°, 15°, and 15°. In the experiment, we establish a unit sphere centered at the origin of the spacecraft's body frame and project the pointing constraints and terminal trajectory onto this unit sphere. By unfolding this unit sphere, we obtain a 2D trajectory diagram. Attitude decomposition problem (7) is a nonlinear optimization problem. We employ a solver named the trust-region method to solve this problem, which is a gradient-based method. The implementation method involves using the "scipy.optimize.minimize" function from Python's SciPy library [31]. Initialization is performed using a random approach. In the relevant calculations, the constant ε in Lemma 3 is set to $\varepsilon = 10^{-8}$.

5.1 Simulation results with multiple forbidden pointing constraints

The attitude constraints used in this set of simulations are CZ1–CZ4 in Table 2, and the pointing of these constraints remains immobile in the inertial frame F_I. CZ1–CZ4 are forbidden pointing zones. If the terminal's trajectory enters these zones, it indicates that the generated attitude trajectory does not satisfy constraint (11). The initial attitude $\boldsymbol{R}_0$ and initial angles $[\theta_0, \varphi_0]$ of the spacecraft are randomly generated, but also satisfy the pointing constraints.

$$\text{Case 1}: \ \boldsymbol{R}_0 = \exp\left(\left([-0.7005, 1.3461, 0.9415]^{\mathrm{T}}\right)^\wedge\right),$$

Table 2 Attitude constraints pointing in simulations.

Serial number	Pointing in F_I	Type	α_j (Case 1)	α_j (Case 2)	α_j (Case 3)	α_j (Case 4)
CZ1	$[0.3214, 0.5567, 0.7660]^{\mathrm{T}}$	Forbidden	21°	21°	–	–
CZ2	$[0.1330, 0.7544, -0.6428]^{\mathrm{T}}$	Forbidden	27°	25°	15°	18°
CZ3	$[-0.7544, 0.1330, 0.6428]^{\mathrm{T}}$	Forbidden	15°	20°	16°	23°
CZ4	$[-0.1504, -0.8529, -0.5000]^{\mathrm{T}}$	Forbidden	15°	20°	18°	28°
CZ5	$[-0.5349, 0.8296, 0.1604]^{\mathrm{T}}$	Mandatory	–	–	15°	15°

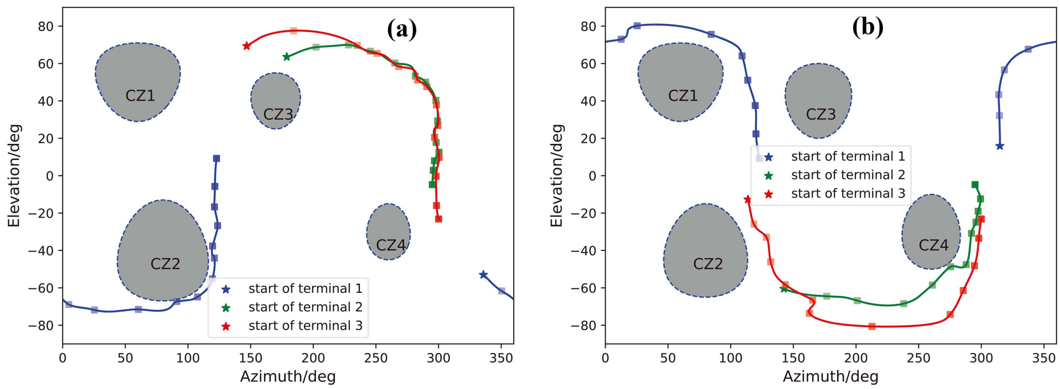

Figure 7 (Color online) Attitude trajectories. (a) Case 1; (b) Case 2.

$$[\theta_0, \varphi_0] = [-0.7435, 0.5418, -1.5104, -0.0464, -0.0977, 0.1397]^{\mathrm{T}},$$
$$\text{Case 2}: \ \boldsymbol{R}_0 = \exp\left(\left([-2.32507, -0.2407, -1.4444]^{\mathrm{T}}\right)^{\wedge}\right),$$
$$[\theta_0, \varphi_0] = [0.9272, 1.2532, 0.0784, 0.1686, 0.1439, 0.1324]^{\mathrm{T}}. \tag{49}$$

To accomplish the attitude planning task, it is first necessary to perform an attitude decomposition to obtain the desired attitudes based on the terminal desired pointing obtained from the relative positions of the spacecraft. Using the attitude decomposition method proposed in Subsection 3.1, the desired attitude and angle are obtained as follows:

$$\text{Case 1}: \ \boldsymbol{R}_d = \exp\left(\left([0.4025, 1.3486, 1.4256]^{\mathrm{T}}\right)^{\wedge}\right),$$
$$[\theta_d, \varphi_d] = [0.6032, 1.5359, -0.2236, 0.0253, -0.1317, -0.1396]^{\mathrm{T}},$$
$$\text{Case 2}: \ \boldsymbol{R}_d = \exp\left(\left([1.9598, 1.2862, 1.8662]^{\mathrm{T}}\right)^{\wedge}\right),$$
$$[\theta_d, \varphi_d] = [-0.7386, 0.2610, -1.5359, -0.1394, -0.0092, 0.1396]^{\mathrm{T}}. \tag{50}$$

The planning framework proposed in this paper includes the following parameters: expansion steps ($\delta_{\mathrm{SO}(3)}$ and δ_h), w_1, and w_2 in the attitude path search, and parameters λ_1 and λ_2 in dynamics planning. The expansion step sizes a and b determine the path length; larger step sizes result in fewer path nodes. Due to the differing motion ranges of the terminal pitch and yaw angles, step sizes for each terminal angle can be designed independently during search. For convenience, identical step sizes are adopted in this paper. Parameters w_1 and w_2 define the theoretical suboptimal boundary of MHA*. To accelerate path search, a larger value for w_1 can be adopted. Parameters λ_1 and λ_2 represent coefficients of $\left(\dot{\boldsymbol{\beta}}_k^{\vee}\right)^{\mathrm{T}}\dot{\boldsymbol{\beta}}_k^{\vee}$ and $\left(\ddot{\boldsymbol{\beta}}_k^{\vee}\right)^{\mathrm{T}}\ddot{\boldsymbol{\beta}}_k^{\vee}$, respectively. As analyzed in Subsection 4.2.3, $\ddot{\boldsymbol{\beta}}_k$ influences angular acceleration magnitude. To achieve a trajectory with smaller angular acceleration, a larger value for λ_2 can be selected.

With the parameters shown in Table 3, using (49) and (50) as inputs to the attitude planning algorithm, the resulting path lengths are 14 and 13, respectively. The terminal pointing discrete trajectories formed from the platform attitude and terminal angles are shown in Figure 7. The starting point of pointing at each terminal is marked with "⋆". The desired pointing of the terminal is fixed in the inertial system, but the initial attitude is given randomly. Thus, in Figure 7, Cases 1 and 2 have different starting points and the same end point. Since nodes that do not satisfy the pointing constraints are removed at generation time, discrete sequences of attitudes can avoid the four constraint zones to reach the endpoints.

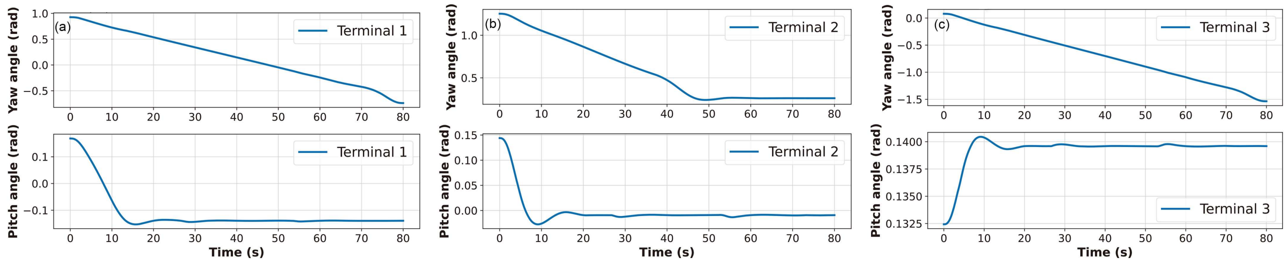

Figure 8 (Color online) Yaw angle and pitch angle curves for each terminal in Case 2. (a) Terminal 1; (b) Terminal 2; (c) Terminal 3.

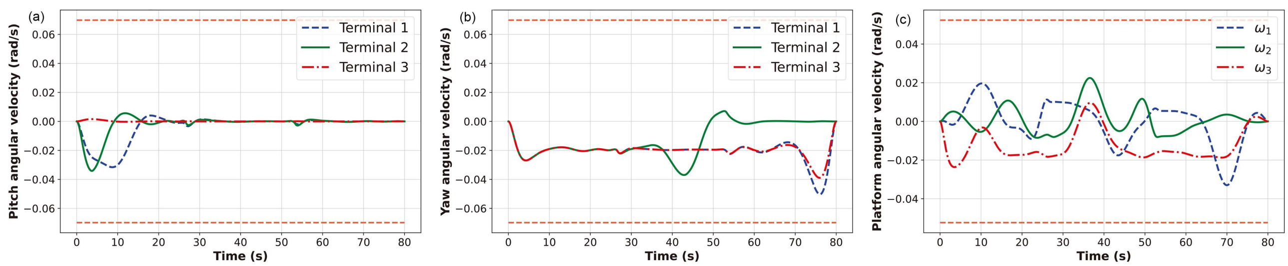

Figure 9 (Color online) Terminals and platform angular velocity curves in Case 2. (a) Pitch angular velocity curves of the terminals; (b) yaw angular velocity curves of the terminals; (c) platform angular velocity curves.

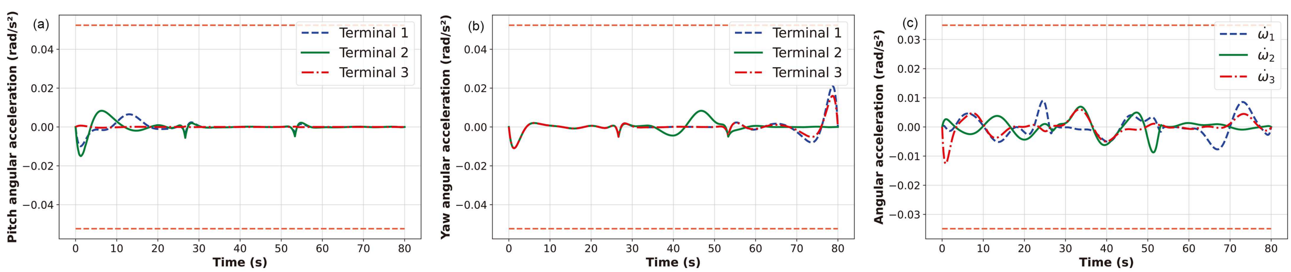

Figure 10 (Color online) Terminals and platform angular acceleration curves in Case 2. (a) Pitch angle angular acceleration curves of the terminals; (b) yaw angle angular acceleration curves of the terminals; (c) platform angular acceleration curves.

The next step after obtaining the discrete attitude sequences is to perform dynamics planning to obtain continuous curves. In the dynamics planning, the Legendre basis functions are chosen with order 12. The remaining parameters are set as follows:

$$\begin{aligned}
&T = 80\ \text{s},\ \xi_1 = 0.0698\ \text{rad/s} = 4^\circ/\text{s},\ \zeta_1 = 0.0524\ \text{rad/s}^2 = 3^\circ/\text{s}^2,\\
&\xi_2 = 0.0524\ \text{rad/s} = 3^\circ/\text{s},\ \zeta_2 = 0.0349\ \text{rad/s}^2 = 2^\circ/\text{s}^2.
\end{aligned} \tag{51}$$

We show the dynamics planning results for Case 2 in detail in Figures 8–10, and the results for Case 2 are abbreviated to give the trajectories in Figure 7. None of the trajectories entered the attitude constraint zones. The angular velocity and angular acceleration curves we have plotted satisfy the upper bound constraints.

Our proposed dynamics planning method is based on a single interval. If the interval-by-interval optimization is performed, the second-order derivatives of the resulting curves (derivatives of angular velocities) have significant abrupt changes at the endpoints of the intervals. Therefore, to reduce the abrupt changes, we combine the interval-by-interval optimization into a multiple-interval optimization. Specifically, we divide the attitude sequence approximately equally into 3 segments and obtain the final curve by solving three optimization problems. The planning results of Case 2 for the platform angular acceleration under different segmentation numbers are shown in Figure 11. As the segmentation number increases, the number of abrupt angular velocity changes caused by segmentation also increases. When such curves serve as control inputs, the adverse effect is that they can lead to multiple instances of instantaneously large control torque requirements.

5.2 Simulation results with mandatory pointing constraints

In this subsection, we perform attitude path planning in a scenario where both forbidden pointing constraints and mandatory pointing constraints exist. We assume that during the attitude maneuvering of the spacecraft, the

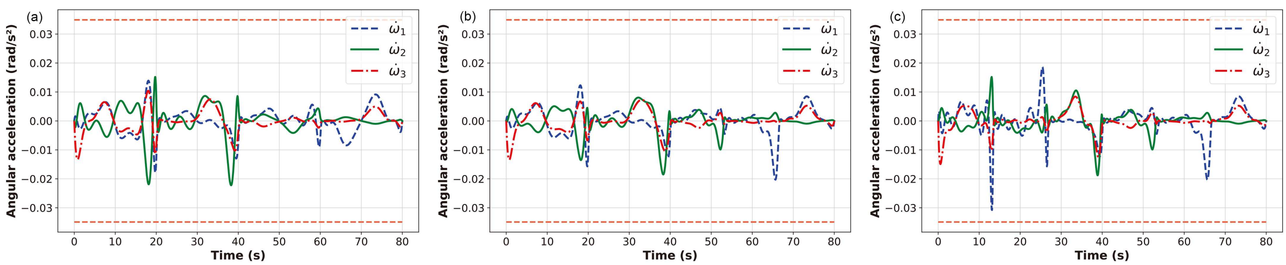

Figure 11 (Color online) Platform angular velocity curves of Case 2 at different numbers of segments. (a) 4 segments; (b) 5 segments; (c) 6 segments.

Table 3 Parameters in each case.

Serial number	δ_h	$\delta_{\mathrm{SO}(3)}$	w_1	w_2	λ_1	λ_2
Case 1	0.1	0.1	8.0	2.0	1	3
Case 2	0.12	0.13	6.0	2.0	1	3
Case 3	0.09	0.1	8.0	3.0	1	3
Case 4	0.14	0.12	8.0	2.0	1	3

trajectory of terminal 1 needs to remain within CZ5 at all times to satisfy constraint (12), while terminals 2 and 3 need to avoid the constraints CZ2–CZ4. The randomly generated initial attitude and initial angles that satisfy the pointing constraints are as follows:

$$\begin{aligned}
\text{Case 3}: \ \boldsymbol{R}_0 &= \exp\left(\left([0.1429, -0.1621, 0.5117]^{\mathrm{T}}\right)^{\wedge}\right), \\
[\theta_0, \varphi_0] &= [1.3677, 0.1354, -1.0721, 0.1211, -0.0937, 0.1837]^{\mathrm{T}}, \\
\text{Case 4}: \ \boldsymbol{R}_0 &= \exp\left(\left([-0.3050, 2.2751, 0.9203]^{\mathrm{T}}\right)^{\wedge}\right), \\
[\theta_0, \varphi_0] &= [1.1601, -0.0852, 1.3184, 0.1425, 0.2259, 0.1617]^{\mathrm{T}}.
\end{aligned} \tag{52}$$

The desired attitude and desired angles obtained after performing the attitude decomposition are as follows:

$$\begin{aligned}
\text{Case 3}: \ \boldsymbol{R}_d &= \exp\left(\left([0.1547, -0.1803, 1.5131]^{\mathrm{T}}\right)^{\wedge}\right), \\
[\theta_d, \varphi_d] &= [0.6137, 1.5359, -0.5119, -0.0080, 0.1017, -0.2269]^{\mathrm{T}}, \\
\text{Case 4}: \ \boldsymbol{R}_d &= \exp\left(\left([0.5993, 2.1165, 1.2377]^{\mathrm{T}}\right)^{\wedge}\right), \\
[\theta_d, \varphi_d] &= [0.5054, 1.5359, -0.3006, 0.0717, -0.2269, -0.0151]^{\mathrm{T}}.
\end{aligned} \tag{53}$$

The path search steps in each case are shown in Table 3. Considering that the pointing of terminal 1 is restricted to a smaller region CZ5, the nodes of terminal 1 are generated with a step size of $0.5\delta_h$. Eventually, in Figure 12, we plot the resulting path nodes, where the paths of terminals 2 and 3 always avoid the forbidden pointing constraints CZ2–CZ4, while the path of terminal 1 remains within CZ5.

In the dynamics planning, the parameter selection is exactly the same as in the previous subsection, and again the resulting path is divided into 3 segments for planning. In Case 3, the trajectories of terminals and 3 go around CZ4 and then reach the end. In Case 4, the trajectory of terminal 2 goes around CZ4 and then reaches the end point, while terminal 3 is always away from the attitude constraint zones. The trajectories obtained by dynamics planning pass sequentially through the attitude sequence obtained by path search and always satisfy the attitude constraints.

In Figures 13–15, we give further detailed results of the dynamics planning for Case 4. The angular profile of each terminal is shown in Figure 13, and the angular change conforms to the movable range constraint of the terminals. The angular velocity curves of the terminal and the platform are shown in Figure 14, while the angular acceleration is demonstrated in Figure 15, and both sets of curves comply with the maximum value constraint and the terminal constraint. The simulations from Case 1 to Case 4 illustrate the effectiveness of the proposed method in this paper to cope with the multi-pointing and multi-constraint attitude planning problem.

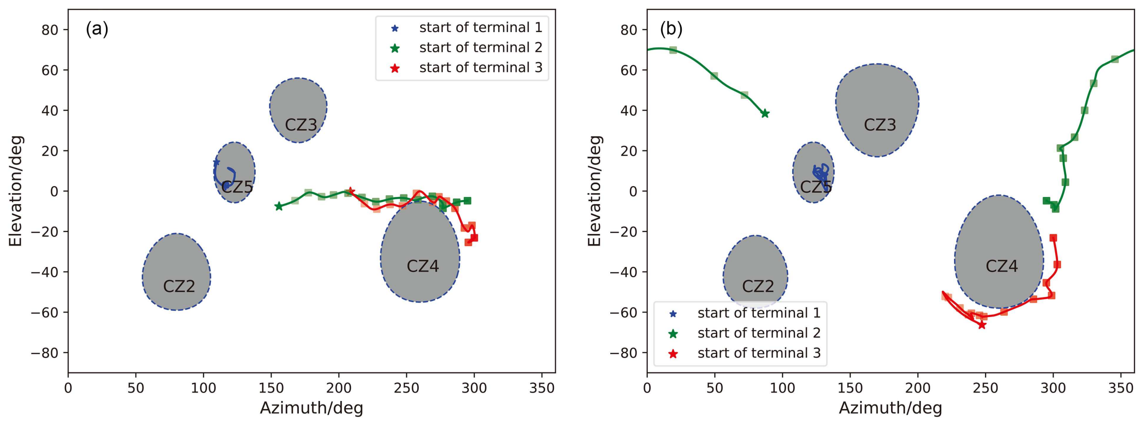

Figure 12 (Color online) Attitude trajectories. (a) Case 3; (b) Case 4.

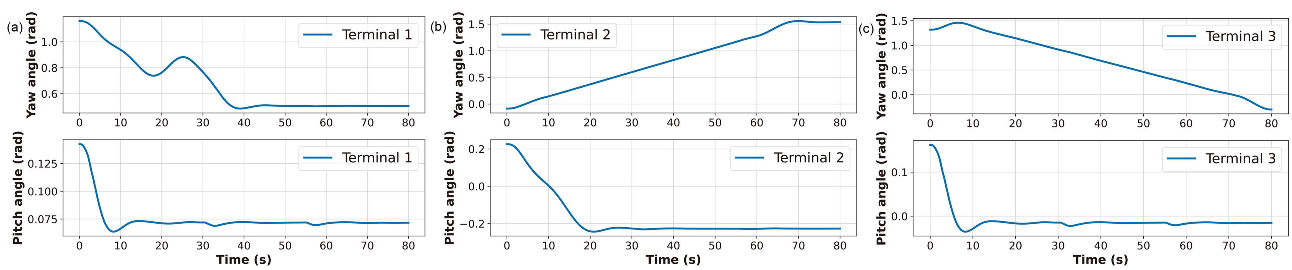

Figure 13 (Color online) Yaw angle and pitch angle curves for each terminal in Case 4. (a) Terminal 1; (b) terminal 2; (c) terminal 3.

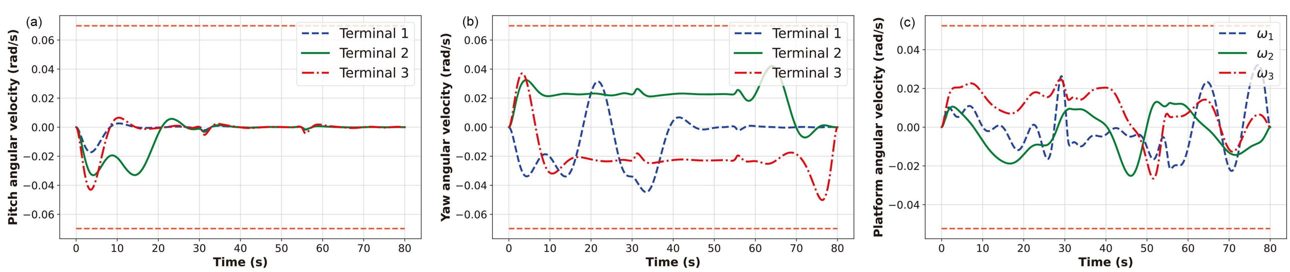

Figure 14 (Color online) Terminals and platform angular velocity curves in Case 4. (a) Pitch angular velocity curves of the terminals; (b) yaw angular velocity curves of the terminals; (c) platform angular velocity curves.

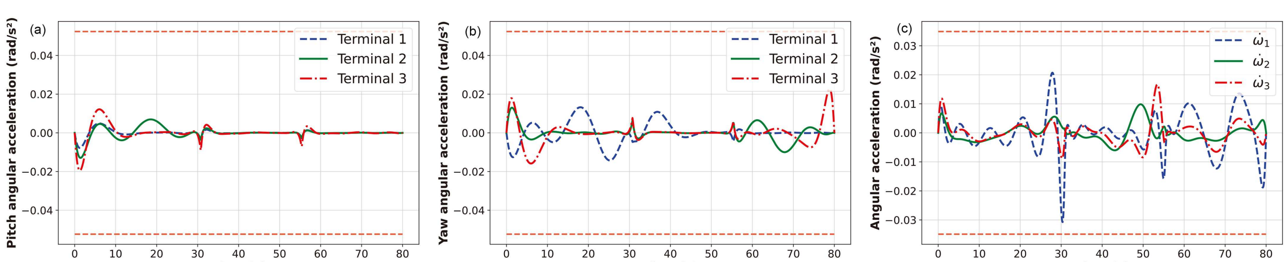

Figure 15 (Color online) Terminals and platform angular acceleration curves in Case 4. (a) Pitch angle angular acceleration curves of the terminals; (b) yaw angle angular acceleration curves of the terminals; (c) platform angular acceleration curves.

Additionally, compared to other planning algorithms, the MHA* algorithm offers advantages in computational efficiency and success rate. To illustrate this, we conducted 20 planning simulations for scenarios with all prohibitive pointing constraints and scenarios with a single mandatory pointing constraint. In these simulations, initial attitudes and constraints were randomly generated. The weighted A* (WA*) algorithm and rapidly-exploring random trees* (RRT*) algorithm are used as comparators. The performance of different algorithms in terms of success rate (SR), minimum computation time (MinT), maximum computation time (MaxT), average computation time (AT), minimum path length (MinL), maximum path length (MaxL), and average path length (AL) within

Table 4 Comparison of MHA*, WA*, and RRT*.

	SR	MinT (s)	MaxT (s)	AT (s)	MinL	MaxL	AL
MHA* (forbidden constraints)	19/20	0.1	0.7	0.3	15	49	27.6
WA* (forbidden constraints)	15/20	0.1	0.6	0.2	17	42	26.6
RRT* (forbidden constraints)	15/20	1.5	25.6	6.0	13	26	20.5
MHA* (mandatory constraint)	19/20	0.2	0.5	0.2	11	38	24.4
WA* (mandatory constraint)	18/20	0.1	1.7	0.3	13	38	24.2
RRT* (mandatory constraint)	17/20	0.7	13.1	3.2	11	28	16.8

fixed step sizes ($\delta_{\mathrm{SO}(3)} = 0.1, \delta_h = 0.08$), fixed runtime (60 s), and fixed iterations (2000) is examined. As shown in Table 4, the MHA* algorithm slightly outperforms WA* in terms of success rate and runtime, and significantly outperforms RRT*. However, RRT* has the advantage of producing planning results with shorter average paths. The algorithm's outstanding performance in success rate and runtime is the primary reason for using MHA* in this paper.

6 Conclusion

This paper proposed a hierarchical attitude planning method for spacecraft with multiple laser terminals. The method involves three key steps: (1) decomposing the inertial pointing requirements to determine the desired platform attitude and terminal angles; (2) employing multi-heuristic A* search to generate a feasible attitude sequence satisfying pointing constraints; (3) connecting this sequence via quadratic programming to form C^2 attitude curves, including an analysis of the quadratic program's physical significance. While the resulting curves are C^2, abrupt changes in angular acceleration remain. Future work will focus on generating smoother attitude nodes and developing planning methods better suited to platform dynamics. Additionally, for constellations comprising numerous satellites, surrounding obstacles impose additional constraints on attitude and orbit control. These issues will be addressed in subsequent studies.

Acknowledgements This work was supported by Guangdong Basic and Applied Basic Research Foundation of China (Grant No. 2023B1515 120018) and Science Center Program of National Natural Science Foundation of China (Grant No. 62188101).

References

1 Hu Q L, Shao 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
2 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
3 Wang D, Wu Y H, Yue C F, et al. Fully actuated flexible spacecraft attitude control with input constraint (in Chinese). Acta Autom Sin, 2024, 50: 2177–2187
4 Li D Y, Ma G F, Li C J, et al. Distributed attitude coordinated control of multiple spacecraft with attitude constraints. IEEE Trans Aerosp Electron Syst, 2018, 54: 2233–2245
5 Ayoubi M A, Hsin J. Sun-avoidance slew planning with keep-out cone and actuator constraints. J Spacecr Rockets, 2020, 57: 1175–1185
6 Alex W, James R F. Constrained attitude control on SO(3) via semidefinite programming. J Guid Control Dyn, 2018, 41: 2483–2488
7 Hari B H. Attitude commands avoiding bright objects and maintaining communication with ground station, J Guid Control Dyn, 1999, 22: 759–767
8 Xue W H, Wang B C, Huang X X, et al. Spacecraft attitude maneuver planning with multi-sensor pointing constraints using improved RRT-star algorithm. Adv Space Res, 2023, 72: 1485–1495
9 Lai B X, Tang Y L, Wang Y N, et al. Finite-time attitude tracking control for rigid spacecraft subject to angular velocity constraints and input magnitude and rate saturation. Aerosp Sci Technol, 2025, 160: 110084
10 Shao X D, Hu Q L, Shi Y, et al. Data-driven immersion and invariance adaptive attitude control for rigid bodies with double-level state constraints. IEEE Trans Contr Syst Technol, 2022, 30: 779–794
11 Shen Q, Yue C F, Cher H G, et al. Rigid-body attitude stabilization with attitude and angular rate constraints. Automatica, 2018, 90: 157–163
12 Li D Y, Tong S, Yang H Y, et al. Time-synchronized control for spacecraft reorientation with time-varying constraints. IEEE ASME Trans Mechatron, 2025, 30: 2073–2083
13 Kulumani S, Lee T. Constrained geometric attitude control on SO(3). Int J Control Autom, 2017, 15: 2796–2809
14 Tan X, Soulaimane B, Dimos V D. Constrained attitude maneuvers on SO(3): rotation space sampling, planning and low-level control. Automatica, 2020, 112: 108659
15 Kjellberg H C, Lightsey E G. Discretized constrained attitude pathfinding and control for satellites. J Guid Control Dyn, 2013, 36: 1301–1309
16 Tanygin S. Fast three-axis constrained attitude pathfinding and visualization using minimum distortion parameterizations. J Guid Control Dyn, 2015, 38: 2324–2336
17 Fabio C, Dennis L. Spacecraft attitude motion planning using gradient-based optimization. J Guid Control Dyn, 2020, 43: 140–145
18 Huo T, Yang J, Yue C F, et al. Gradient-based attitude planning for rigid spacecraft on SO(3). Sci China Tech Sci, 2025: 1520401
19 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
20 Zhong X L, Wei Z T, Chen T. Motion planning and pose control for flexible spacecraft using enhanced LQR-RRT*. IEEE Trans Aerosp Electron Syst, 2023, 59: 8743–8751
21 Fan Y Q, Xu R, Li Z Y, et al. Polynomial attitude trajectory planning for spacecraft with movable parts using decoupled strategy. J Guid Control Dyn, 2024, 47: 72–86

22 Xu R, Fan Y Q, Li Z Y, et al. Time-optimal attitude planning for spacecraft with movable parts using second order cone programming. Aerosp Sci Technol, 2023, 141: 108589

23 Calaon R, Schaub H. Constrained attitude path planning via least-squares modified-rodrigues-parameters-based smoothing curves. J Guid Control Dyn, 2023, 46: 2207–2217

24 Wang Z, Xu R. Rapid search method for a spacecraft attitude maneuver path with multiple constraints. Aerosp Sci Technol, 2021, 117: 106904

25 Richard M M, Li Z X, Sastry S S. A Mathematical Introduction to Robotic Manipulation. Boca Raton: CRC Press, 1994. 25–30

26 Juliano T, Olivier B. Highly accurate differentiation of the exponential map and its tangent operator. Mech Mach Theory, 2023, 190: 105451

27 Huynh D Q. Metrics for 3D rotations: comparison and analysis. J Math Imaging Vis, 2009, 35: 155–164

28 Aine S, Swaminathan S, Narayanan V, et al. Multi-heuristic A*. Int J Rob Res, 2016, 35: 224–243

29 Hüper K, Leite F S. On the geometry of rolling and interpolation curves on S^n, $\mathrm{SO}(n)$, and Grassmann manifolds. J Dyn Control Syst, 2007, 13: 467–502

30 Shen Y S, Hüper K, Leite F S. Smooth interpolation of orientation by rolling and wrapping for robot motion planning. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006. 113–118

31 Pauli V, Ralf G, Travis E O, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods, 2020, 17: 261–272