Acta Astronautica 214 (2024) 125–136

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

Distributed safe trajectory optimization for large-scale spacecraft formation reconfiguration

Junyu Chen [a], Baolin Wu [a,*], Zhaobo Sun [a], Danwei Wang [b]

[a] Research Center of Satellite Technology, Harbin Institute of Technology, 150001, Harbin, China
[b] Nanyang Technological University, Singapore, 639798, Singapore

ARTICLE INFO

Keywords:
Large-scale spacecraft formation
Distributed optimization algorithm
Trajectory optimization
Collision avoidance
Autonomous reconfiguration

ABSTRACT

This paper presents a distributed trajectory optimization algorithm for the reconfiguration of large-scale spacecraft formation, which uses a low thrust propulsion system to safely guide the spacecraft formation to a desired formation. Formation reconfiguration is formulated as a trajectory optimization problem with complex constraints, and the relative motion is accurately described by a nonlinear relative dynamics considering J_2 perturbation. The collision avoidance constraint is convexified into a tangent plane to ensure the accuracy of the solution. The resulting nonlinear trajectory optimization problem is solved by applying the *hp*-adaptive pseudospectral method to convert it into a nonlinear programming problem. In order to overcome the disadvantage of huge computation of centralized algorithm considering collision avoidance, "predicted trajectory" is introduced to transmit information between spacecraft, and the parallel computing is implemented. Finally, a numerical simulation is given to verify the computational efficiency and collision avoidance effectiveness of the proposed distributed algorithm.

1. Introduction

With the continuous development of space technology, spacecraft formation flying (SFF) has become a hot topic in the aerospace field [1]. SSF can use a variety of sensors to form a "large virtual spacecraft" to complete high-precision and highly complex space missions that cannot be completed independently by a single spacecraft [2]. SFF offers advantages such as enhanced mission flexibility, decreased cost and risk, and so on. With the wide application of formation, small-scale SFF can no longer meets the needs of some future space missions. Large-scale SSF and related technologies of dozens or even hundreds of spacecraft are the key development direction of future space multi-satellite missions. At present, countries around the world have put forward networking plans for large-scale SSF in the fields of communication, remote sensing, technology testing, etc [3].

Formation reconfiguration refers to the formation maneuver from the current formation to another desired formation due to the need of space tasks. As we all know, reconfiguration is one of the most challenging problems in SSF. Collision avoidance of SSF is the first problem to be considered. Due to the large number of spacecraft, the trajectory optimization for formation reconfiguration needs to ensure that there is no collision between spacecraft in the formation, which makes the trajectory optimization problem complex [4]. In order to solve the spacecraft trajectory optimization problem with complex constraints, the methods proposed by researchers can be roughly divided into two categories: analytical methods and numerical optimization methods [5]. The analytical methods are generally based on the relative orbital elements, using artificial potential function and other tools to solve the reconfiguration problem relied on impulse control [6–8]. Numerical optimization methods can be divided into two categories: indirect methods and direct methods [9]. In the indirect methods, the optimal conditions are derived by using the variational method, and the nonlinear multipoint boundary value problem consisting of necessary conditions is then solved. The main disadvantage of indirect methods is that the convergence radius is usually very small, and it is difficult to analytically derive the optimal conditions for complex constraints [10]. In the direct methods, the system dynamics is discretized into a set of algebraic constraints, and the reconfiguration trajectory optimization problem is converted into a complex linear/nonlinear programming problem for solution, which is often used to solve the control problem of continuous low thrust. Low thrust engines (such as ion thrusters and Hall thrusters) can generate accurate and variable control forces, and can effectively reduce fuel consumption due to high specific impulse, so they have broad application prospects in the future [11]. Wu et al. [12] used high-precision nonlinear relative dynamics to study the trajectory optimization problem of formation maneuver by the direct methods.

* Corresponding author.
E-mail address: wuba0001@e.ntu.edu.sg (B. Wu).

https://doi.org/10.1016/j.actaastro.2023.10.012
Received 6 January 2023; Received in revised form 11 August 2023; Accepted 11 October 2023
Available online 18 October 2023

The disadvantages of the direct methods are that the number of discrete variables is large, the constraints are complex, the calculation cost is large, and it is difficult to deal with non-convex constraints such as collision avoidance. Convexification is critical for accurate solutions. In order to reduce the calculation cost, one method is to use binary variables to convexify collision avoidance constraint and describe the trajectory optimization as a mixed integer linear programming (MILP) problem [13,14]. In addition, other authors converted the complex constraints considering collision avoidance into convex sets to reduce the computation, and used convex optimization methods to solve the trajectory optimization problem [15–18].

However, the above mentioned centralized or decentralized algorithms have disadvantages. For instance, the computational load will increase sharply with the formation scale. Therefore, for large-scale formation system, it is critical to design a distributed optimization algorithm to reduce the computation. Different from centralized and decentralized algorithms, the distributed algorithm does not require a data processing center and distributes the computational burden to each agent through point-to-point communication, achieving parallel computing [19]. In the field of multi-agent, many scholars have done a lot of research on distributed algorithms, and designed distributed protocols based on different methods to ensure the convergence of control algorithms and the effectiveness of collision avoidance [20–23]. In the aerospace field, Silvestrini et al. [24] proposed reverse reinforcement learning and long short-term memory network to predict neighboring trajectories to ensure collision-free and distributed computing. Other authors employed graph theory to describe information interaction, and designed distributed algorithms for spacecraft formation reconfiguration considering collision avoidance by using artificial potential function [25,26], convex optimization [27] and other methods [28]. However, the control accuracy is not high because the perturbation term is ignored or the linearization approximation is adopted. Based on the above discussion, the main research efforts are dedicated to the development of distributed safe reconfiguration algorithms with fast computation and high control accuracy.

The contribution of this paper is to develop a distributed trajectory optimization algorithm for large-scale spacecraft formation reconfiguration, which realizes parallel computing. Specifically, the nonlinear relative dynamics considering J_2 perturbation is used to accurately describe relative motion constraint. The collision avoidance is convexified by unfolding the sphere into a tangent plane, which reduces the computational complexity and guarantees the accuracy of the solution. Then, the "predicted trajectory" is proposed to decouple the state variables of each spacecraft in optimization problem, so that each spacecraft solves the optimization problem independently. In addition, communication traffic is reduced by considering neighbor update. Finally, the trajectory optimization problem is converted into a nonlinear trajectory programming problem in discrete time by using the *hp*-adaptive pseudospectral method. Each spacecraft solves the optimization problem through several iterations to generate the collision-free optimal trajectory. The proposed algorithm is tested with a large-scale spacecraft formation reconfiguration containing more than a dozen microsatellites as a scenario.

This paper is organized as follows. In Section 2, the nonlinear relative dynamics considering J_2 perturbation and graph theory are introduced. Section 3 describes the centralized optimization algorithm of formation reconfiguration. In Section 4, a distributed trajectory optimization algorithm is designed. In Section 5, the proposed distributed algorithm is used to reconfigure a fuel-optimal and collision-free maneuver trajectory for the large-scale spacecraft formation. Finally, Section 6 gives the conclusion of this paper.

2. Preliminaries

2.1. Equation of relative motion

In order to generate accurate formation maneuver trajectory, a high precision relative dynamics is needed. In this paper, a nonlinear relative dynamics considering J_2 perturbation is used, which is briefly described below. Readers can refer to Section 2.2 of Ref. [29] written by the authors for more details of the dynamics.

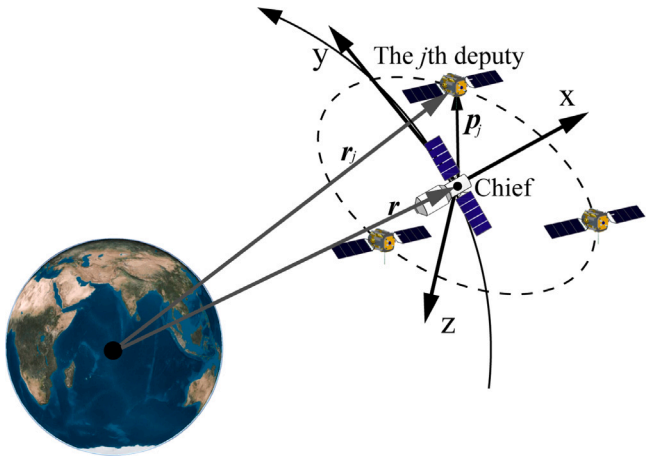


Fig. 1. LVLH frame.

First, define the local-vertical/local-horizontal (LVLH) rotating frame as shown in Fig. 1. This frame is centered at the chief spacecraft, and the unit vector $\boldsymbol{x}$ is directed from the spacecraft radially outward from the Earth center; $\boldsymbol{z}$ is perpendicular to the orbital plane in the direction of the instantaneous angular momentum vector; and $\boldsymbol{y}$ is determined by the right-hand rule.

The J_2 nonlinear differential equations describing the relative motion of spacecraft are as follows:

$$\begin{aligned}
\ddot{x}_j =\ & 2\dot{y}_j\omega_z - x_j\left(\eta_j^2 - \omega_z^2\right) + y_j\alpha_z - z_j\omega_x\omega_z - \left(\zeta_j - \zeta\right)\sin i\sin\theta \\
& - r\left(\eta_j^2 - \eta^2\right) + u_{j,x} \\
\ddot{y}_j =\ & -2\dot{x}_j\omega_z + 2\dot{z}_j\omega_x - x_j\alpha_z - y_j\left(\eta_j^2 - \omega_z^2 - \omega_x^2\right) + z_j\alpha_x \\
& - \left(\zeta_j - \zeta\right)\sin i\cos\theta + u_{j,y} \\
\ddot{z}_j =\ & -2\dot{y}_j\omega_x - x_j\omega_x\omega_z - y_j\alpha_x - z_j\left(\eta_j^2 - \omega_x^2\right) \\
& - \left(\zeta_j - \zeta\right)\cos i + u_{j,z}
\end{aligned} \tag{1}$$

where $\boldsymbol{p}_j = (x_j,\ y_j,\ z_j)$ represents the three axis components of the relative position vector of the jth deputy spacecraft in the LVLH frame. $\boldsymbol{U}_j = (u_{j,x}, u_{j,y}, u_{j,z})$ represents the control acceleration of the jth deputy spacecraft in the LVLH frame.

The five differential equations describing the changes of relevant parameters of the chief spacecraft affected by J_2 perturbation are as follows:

$$\begin{aligned}
\dot{r} &= v_x \\
\dot{v}_x &= -\mu/r^2 + h^2/r^3 - k_{J2}\left(1 - 3\sin^2 i\sin^2\theta\right)/r^4 \\
\dot{h} &= -k_{J2}\left(\sin^2 i\sin 2\theta\right)/r^3 \\
\dot{\theta} &= h/r^2 + 2k_{J2}\left(\cos^2 i\sin^2\theta\right)/\left(hr^3\right) \\
\dot{i} &= -k_{J2}(\sin 2i\sin 2\theta)/\left(2hr^3\right)
\end{aligned} \tag{2}$$

where r, v_x, h, i, θ respectively represent the distance from the chief spacecraft to the geocenter, the radial velocity, the orbital angular momentum, the orbital inclination and the latitude argument. $\mu = 398,600\ \mathrm{km}^3/\mathrm{s}^2$ represents the standard gravitational parameter.

$(\omega_x,\ \omega_z)$ and $(\alpha_x,\ \alpha_y)$ are the orbital angular velocity and acceleration of the chief spacecraft, given by the following formula:

$$\omega_x = -k_{J2}(\sin 2i\sin\theta)/\left(hr^3\right) \tag{3}$$

$$\omega_z = h/r^2 \tag{4}$$

$$\alpha_x = -k_{J2}(\sin 2i \cos\theta)/r^5 + 3v_x k_{J2}(\sin 2i \sin\theta)/\left(r^4 h\right) - 8k_{J2}^2 \sin^3 i \cos i \sin^2\theta \cos\theta/\left(r^6 h^2\right) \tag{5}$$

$$\alpha_z = -2hv_x/r^3 - k_{J2}\left(\sin^2 i \sin 2\theta\right)/r^5 \tag{6}$$

η^2, η_j^2, ζ_j and ζ are defined as follows:

$$\eta^2 = \mu/r^3 + k_{J2}/r^5 - 5k_{J2}\sin^2 i \sin^2\theta/r^5 \tag{7}$$

$$\eta_j^2 = \mu/r_j^3 + k_{J2}/r_j^5 - 5k_{J2}r_{jz}^2/r_j^7 \tag{8}$$

$$\zeta_j = 2k_{J2}r_{jz}/r_j^5, \quad \zeta = 2k_{J2}(\sin i \sin\theta)/r^4 \tag{9}$$

k_{J2}, r_j and r_{jz} are defined as follows:

$$k_{J2} = 3J_2\mu R_e^2/2 \tag{10}$$

$$r_j = \sqrt{\left(r + x_j\right)^2 + y_j^2 + z_j^2} \tag{11}$$

$$r_{jz} = \left(r + x_j\right)\sin i \sin\theta + y_j \sin i \cos\theta + z_j \cos i \tag{12}$$

where $J_2 = 1.08262\times10^{-3}$ and $R_e = 6371$ km represent the second zonal harmonic constant and the average radius of Earth, respectively.

2.2. Graph theory

In distributed algorithm, information interaction between spacecraft is required. For the convenience of description, a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{A})$ is introduced to describe the topology of information interaction [30], where $\boldsymbol{V} = \{1, 2, \ldots, N_a\}$ represents the set of N_a spacecraft in the formation system; $\boldsymbol{V}$ is the set of edges, and $e_{ij} = (i, j)$ indicates that jth spacecraft can obtain the information of the ith spacecraft; $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{N_a\times N_a}$ represents the adjacency matrix. Define $a_{ij} = 1$ when $e_{ij} = (i, j) \in \boldsymbol{E}$, otherwise $a_{ij} = 0$. Since each spacecraft does not need to communicate with itself, $a_{ij} = 0$, $\forall i \in \boldsymbol{V}$. If $\forall i, j \in \boldsymbol{V}$, e_{ij} & $e_{ji} \in \boldsymbol{E}$, that is, information interaction can be achieved between spacecraft i and j, then the topology of the system is undirected, otherwise it is directed. The undirected communication topology of formation system is considered in this paper.

3. Centralized formation reconfiguration trajectory optimization

To facilitate the reader's understanding, this paper first describes the centralized formation reconfiguration trajectory optimization algorithm. Then, based on this, the distributed formation reconfiguration trajectory optimization algorithm is proposed in the next section.

For the numerical optimization methods, the formation reconfiguration problem is regarded as a complex constrained optimization problem with two-point boundary values, and its general framework is [16]:

minimize	$\boldsymbol{J}(\boldsymbol{U})$
subject to	system dynamics
	initial & terminal state
	maximum & minimum thrust
	collision avoidance

Next, each part will be introduced in detail.

3.1. Cost function

The aim of the trajectory optimization for spacecraft formation reconfiguration is to minimize fuel consumption. For the formation system of N_a deputy spacecraft, the cost function is selected:

$$\boldsymbol{J}(\boldsymbol{U}) = \sum_{j=1}^{N_a}\frac{1}{2}\int_{t_0}^{t_f}\boldsymbol{U}_j^T(t)\boldsymbol{U}_j(t)\mathrm{d}t \tag{13}$$

where t_0 and t_f are the time corresponding to the initial and terminal state of the spacecraft respectively. The optimization problem is constrained by system dynamics, initial and terminal state, maximum and minimum thrust, and collision avoidance, which will be described in detail in the following sections.

3.2. System dynamics

For the formation system of N_a spacecraft, complete set is defined as $\mathbb{N}_a = \{1, 2, \ldots, N_a\}$. For graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{A})$, $\boldsymbol{V} = \mathbb{N}_a$. In the LVLH frame, $\boldsymbol{X}_j = \left[x_j, y_j, z_j, \dot{x}_j, \dot{y}_j, \dot{z}_j\right]^T$ is defined as the state variable describing the relative motion of spacecraft j. According to Eq. (1), system dynamics is formulated as the following nonlinear state equation:

$$\forall j \in \mathbb{N}_a, \quad \dot{\boldsymbol{X}}_j(t) = \boldsymbol{f}\left(\boldsymbol{X}_j(t), \boldsymbol{\sigma}(t)\right) + \boldsymbol{B}\boldsymbol{U}_j(t) \tag{14}$$

where $\boldsymbol{B} = \left[\begin{array}{c}\boldsymbol{0}_{3\times3}, \boldsymbol{I}_{3\times3}\end{array}\right]^T$. $\boldsymbol{f}(\cdot)$ is the nonlinear vector-valued function. $\boldsymbol{\sigma}(t) = \left[r(t), v_x(t), h(t), i(t), \theta(t)\right]^T$ represents the change of relevant parameters caused by J_2 perturbation of the chief spacecraft. It can be seen from Eq. (2) that the change of $\boldsymbol{\sigma}(t)$ is independent of $\boldsymbol{X}_j$. Therefore, Eq. (2) is simply expressed as the following closed form:

$$\dot{\boldsymbol{\sigma}}(t) = \boldsymbol{g}(\boldsymbol{\sigma}(t)) \tag{15}$$

3.3. Initial state and terminal state

The initial state constraint refers to the relative motion state of the jth spacecraft at the initial time t_0. The relevant equality constraint is given by:

$$\begin{aligned}\boldsymbol{X}_j\left(t_0\right) &= \left[x_j\left(t_0\right), y_j\left(t_0\right), z_j\left(t_0\right), \dot{x}_j\left(t_0\right), \dot{y}_j\left(t_0\right), \dot{z}_j\left(t_0\right)\right]^T \\ &= \left[x_{j,0}, y_{j,0}, z_{j,0}, \dot{x}_{j,0}, \dot{y}_{j,0}, \dot{z}_{j,0}\right]^T\end{aligned} \tag{16}$$

The initial state can be obtained by relative measurement and recursion before maneuvering. For example, the current relative motion state $\boldsymbol{X}_j\left(t_0 - \Delta t\right)$ is measured by Δt ($\Delta t > t_{compute}$, $t_{compute}$ is the computation time) before the start of the maneuver, and the state $\boldsymbol{X}_j\left(t_0\right)$ of the spacecraft at time t_0 is recursively obtained on-board (the state is considered to be accurate due to the short recursion time).

The terminal state constraint characterizes the relative motion state of the spacecraft at the terminal time t_f. In this paper, the equality constraint is:

$$\begin{aligned}\boldsymbol{X}_j\left(t_f\right) &= \left[x_j\left(t_f\right), y_j\left(t_f\right), z_j\left(t_f\right), \dot{x}_j\left(t_f\right), \dot{y}_j\left(t_f\right), \dot{z}_j\left(t_f\right)\right]^T \\ &= \left[x_{j,f}, y_{j,f}, z_{j,f}, \dot{x}_{j,f}, \dot{y}_{j,f}, \dot{z}_{j,f}\right]^T\end{aligned} \tag{17}$$

The terminal state is given according to the desired formation prior, which satisfies the energy matching condition to ensure the long-term passive stability of the formation. The energy matching condition for bounded relative motion of spacecraft considering nonlinearity and orbital eccentricity is shown in the formula [31]:

$$\begin{aligned}&\frac{1}{2}\left\{\left(\dot{x}_{j,f} - \omega_{z,f}y_{j,f} + \dot{r}_f\right)^2 + \left[\dot{y}_{j,f} + \omega_{z,f}\left(x_{j,f} + r_f\right)\right]^2 + \dot{z}_{j,f}^2\right\} \\ &\quad - \mu/\sqrt{\left(r_f + x_{j,f}\right)^2 + y_{j,f}^2 + z_{j,f}^2} = -\mu/2a\end{aligned} \tag{18}$$

where a is semi-major axis of the chief spacecraft. For instance, a circle fly-around formation under the two-body condition can be designed as follows [29]:

$$\begin{cases} x(t_0) = (r_{\text{cir}}/2)\cos\varphi, \quad \dot{x}(t_0) = (-r_{\text{cir}}\omega_z/2)\sin\varphi \\ y(t_0) = 2\dot{x}_0/\omega_z, \quad \dot{y}(t_0) = -\omega_z(2+e)x(t_0)/[(1+e)^{1/2}(1-e)^{3/2}] \\ z(t_0) = \pm\sqrt{3}x(t_0), \quad \dot{z}(t_0) = \pm\sqrt{3}\dot{x}(t_0) \end{cases} \tag{19}$$

where r_{cir} is the radius of the formation, φ is the phase angle of the formation, ω_z and e is the average orbital angular velocity and orbital eccentricity of the chief spacecraft, respectively. The relative motion state in Eq. (19) satisfies Eq. (18). In natural motion, due to various perturbation effects, it will drift slowly.

3.4. Maximum and minimum thrust

Because the continuous low thrust engines of micro spacecraft can only generate a certain range of variable thrust. Therefore, it is necessary to consider the maximum and minimum thrust constraints. For the jth spacecraft, the maximum and minimum thrust constraints are expressed as:

$$\left\|\boldsymbol{U}_j(t)\right\|_p \le u_{j,\max}, \quad p = 2 \text{ or } \infty \tag{20}$$

$$\left\|\boldsymbol{U}_j(t)\right\|_p \ge u_{j,\min}, \quad p = 2 \text{ or } \infty \tag{21}$$

where $u_{j,max}$ and $u_{j,min}$ represents the maximum and minimum control accelerations provided by the jth spacecraft thruster, respectively. $\|\cdot\|_p$ represents the p norm. If the spacecraft carries a single thruster, $p = 2$; if it carries multiple thrusters, $p = \infty$. Note that the Eq. (21) represents a non-convex constraint, which reduces the accuracy of the solution and increases the computation. Then, the binary variable $b_{jk}(t)$ is used to convexify it into a mixed integer linear constraint:

$$\begin{aligned} u_{j,x}(t) &\ge u_{j,\min} - Mb_{j1}(t) \\ -u_{j,x}(t) &\ge u_{j,\min} - Mb_{j2}(t) \\ u_{j,y}(t) &\ge u_{j,\min} - Mb_{j3}(t) \\ -u_{j,y}(t) &\ge u_{j,\min} - Mb_{j4}(t) \\ u_{j,z}(t) &\ge u_{j,\min} - Mb_{j5}(t) \\ -u_{j,z}(t) &\ge u_{j,\min} - Mb_{j6}(t) \\ \sum_{k=1}^{6} b_{jk}(t) &\le 5, \quad b_{jk}(t) \in \{0,1\} \end{aligned} \tag{22}$$

where M is a large number.

3.5. Collision avoidance

Define the safe distance as d_{safe}, and the relative positions of two different spacecraft j and i at time t are expressed as $\boldsymbol{p}_j(t) = [x_j(t), y_j(t), z_j(t)]^T$ and $\boldsymbol{p}_i(t) = [x_i(t), y_i(t), z_i(t)]^T$. Collision avoidance can be described as:

$$\left\|\boldsymbol{p}_j(t) - \boldsymbol{p}_i(t)\right\|_2 \ge d_{safe}, \quad \forall j, i \in \mathbb{N}_a \tag{23}$$

Eq. (23) is nonlinear and describes the keep-out-zone as a non-convex "repulsive ball" model, as shown in Fig. 2(a). Inspired by Ref. [15], the non-convex constraint in Eq. (23) is convexified by unfolding the sphere into a tangent plane, as shown in Fig. 2(b). The new collision avoidance constraint is:

$$\left(\overline{\boldsymbol{p}}_j(t) - \overline{\boldsymbol{p}}_i(t)\right)^T \left(\boldsymbol{p}_j(t) - \boldsymbol{p}_i(t)\right) \ge d_{safe}\left\|\overline{\boldsymbol{p}}_j(t) - \overline{\boldsymbol{p}}_i(t)\right\|_2, \quad \forall j, i \in \mathbb{N}_a \tag{24}$$

where $\overline{\boldsymbol{p}}_j(t)$ and $\overline{\boldsymbol{p}}_i(t)$ are the nominal trajectories, which are the initial guesses of the actual trajectories $\boldsymbol{p}_j(t)$ and $\boldsymbol{p}_i(t)$, and are used to convexify the collision avoidance constraint. The value of the nominal trajectory will be given in Section 4.2.

4. Distributed formation reconfiguration trajectory optimization

For the centralized trajectory optimization algorithm [12–18], all spacecraft are required to jointly solve the optimization problem considering collision avoidance. Because the state variables of each spacecraft are coupled, the computation will increase sharply with the scale of the formation, which is unsuitable for application in engineering. Therefore, it is necessary to design a distributed trajectory optimization algorithm that can be quickly calculated under complex constraints such as collision avoidance.

In the following section, a distributed trajectory optimization algorithm is designed by introducing "predicted trajectory". The parallel computation of the trajectory optimization problem is achieved by information interaction between neighbors.

4.1. Definition of neighbor

In order to accurately describe information interaction and implement parallel computing to reduce computation, the concept of "neighbor" is introduced. Define the neighbor distance as $d_{neighbor}$ ($d_{neighbor} > d_{safe}$). For $\forall j, i \in V$, $j \ne i$, spacecraft j and i become neighbors if the distance of their optimal trajectories is less than $d_{neighbor}$. Information can be transmitted between them, i.e. $a_{ij} = a_{ji} = 1$. Because the distance between neighbors is closer and the collision risk is higher, collision avoidance between neighbors is only considered. Define the set of j's neighbors as: $\mathbb{N}_j = \{i \mid e_{ij} \in \boldsymbol{E}\}$, with N_j neighbors.

4.2. Predicted trajectory

Since the collision avoidance constraint (24) contains the actual trajectories of other spacecraft, it is infeasible for spacecraft j to solve the trajectory optimization problem considering collision avoidance independently.

In order to make the optimization problem solvable, the actual trajectories of other spacecraft are replaced by the predicted trajectories. It is important to note that since the predicted trajectory is not equal to the actual trajectory, it takes several iterations to solve the optimization problem to obtain the optimal trajectory for effective collision avoidance [23].

Define $\boldsymbol{p}_j^{(n)^*}(t) = [x_j(t), y_j(t), z_j(t)]^{(n)^*}, \forall t \in [t_0, t_f]$ be the optimal trajectory obtained by solving the trajectory optimization problem of spacecraft j at the nth iteration. When the spacecraft j's neighbors solve their own trajectory optimization problem at the nth iteration, the predicted trajectory transmitted by spacecraft j is $\hat{\boldsymbol{p}}_j^{(n)}(t), \forall t \in [t_0, t_f]$. The relationship between the predicted trajectory and the optimal trajectory is

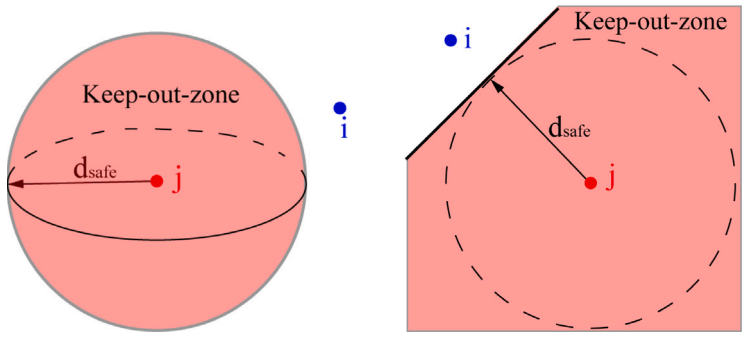
$$\hat{\boldsymbol{p}}_j^{(n)}(t) = \boldsymbol{p}_j^{(n-1)^*}(t), \quad n \ge 2 \ \&\ n \in \mathbb{Z} \quad \forall j \in \mathbb{N}_a \tag{25}$$

Since the predicted trajectory can reflect the future movement trend of the spacecraft, then the nominal trajectory in Eq. (24) is equal to the predicted trajectory:

$$\overline{\boldsymbol{p}}_j(t) = \hat{\boldsymbol{p}}_j^{(n)}(t), \quad \forall j \in \mathbb{N}_a \tag{26}$$

When spacecraft j solves its own trajectory optimization problem at the nth iteration, the set of predicted trajectories of all its neighbors is defined as:

$$\hat{\boldsymbol{P}}_j^{(n)}(t) = \left[\hat{\boldsymbol{p}}_{i_1}^{(n)}(t), \ldots, \hat{\boldsymbol{p}}_{i_{N_j}}^{(n)}(t)\right]^T, \quad \forall t \in [t_0, t_f], \ i_1, \ldots, i_{N_j} \in \mathbb{N}_j \tag{27}$$



(a) Repulsive sphere (non-convex) (b) Tangent plane (convex)

Fig. 2. Convexification of the collision-avoidance constraint.

4.3. Optimization problem formulation

Replacing the actual trajectories in the collision avoidance constraint (24) with the predicted trajectories of the neighbors. For each spacecraft $j = 1, \dots, N_a$, the initial state $\boldsymbol{X}_{j,0}$ and the terminal state $\boldsymbol{X}_{j,f}$ are known when solving its own trajectory optimization problem at the nth iteration. In particular, when $n \geq 2$, the set of predicted trajectories of neighbors is $\hat{\boldsymbol{P}}_j^{(n)}(t) = \left[\hat{\boldsymbol{p}}_{i_1}^{(n)}(t), \dots, \hat{\boldsymbol{p}}_{i_{N_j}}^{(n)}(t)\right]^T, \forall t \in \left[t_0, t_f\right],\ i_1, \dots, i_{N_j} \in \mathbb{N}_j$. Decision variables in optimization problem include $\boldsymbol{X}_j(t)$, $\boldsymbol{U}_j(t)$, and $b_{jk}(t)$, $\forall t \in \left[t_0, t_f\right]$. The optimization problem for spacecraft j (**Problem 1**) is formulated as follows:

$$min \quad J_j = \frac{1}{2}\int_{t_0}^{t_f} \boldsymbol{U}_j^T(t)\boldsymbol{U}_j(t)\mathrm{d}t \tag{28}$$

$\forall t \in \left[t_0, t_f\right]$, subject to Eqs. (14), (16), (17), (20), (22) and

$$\left(\hat{\boldsymbol{p}}_j^{(n)}(t) - \hat{\boldsymbol{p}}_i^{(n)}(t)\right)^T \left(\boldsymbol{p}_j(t) - \hat{\boldsymbol{p}}_i^{(n)}(t)\right) \geq d_{safe}\left\|\hat{\boldsymbol{p}}_j^{(n)}(t) - \hat{\boldsymbol{p}}_i^{(n)}(t)\right\|_2, \quad \forall i \in \mathbb{N}_j, n \geq 2 \tag{29}$$

For the first optimization, i.e. $n = 1$, since the predicted trajectory of each spacecraft does not exist, the problem is to solve an optimal trajectory without considering collision avoidance. When $n \geq 2$, for any spacecraft j, the "predicted trajectory" is obtained by Eq. (25). Then, the predicted trajectory is transmitted to the neighbors based on the adjacency matrix $\boldsymbol{A}$, and the predicted trajectories of neighbors are received at the same time. Then trajectory optimization problem of spacecraft j by using the predicted trajectories of its neighbors to achieve collision avoidance is then solved.

Define $\mathcal{O}(f(N))$ as the computation time complexity as a function $f(N)$ of varying formation scale N. According to Eq. (29), it can be seen that the larger the number of neighbors N_j, the higher the computation time complexity of the collision avoidance constraint, and it increases linearly, i.e., $\mathcal{O}(N_j)$. The computational complexity of the remaining constraints does not change with the formation scale. Therefore, for spacecraft formations of different scales, the computation time complexity of **Problem 1** is $\mathcal{O}(N_j)$ (only related to the number of neighbors N_j), which will be verified in Section 5.

4.4. Discretization and implementation

In the previous section, **Problem 1** is formulated on the basis of continuous system. In numerical simulation, it is discretized by using the *hp*-adaptive pseudospectral method. Taking advantages of the Gauss pseudospectral method and finite element method, the method divides the problem into several units, each of which is approximated by a polynomial with a relatively low order. An adaptive method is used to determine the number of divisions and the order of interpolation polynomials. Based on the spectral formation method used in solving differential equations, various complex constraints are allocated to the Legendre–Gauss–Lobatto (LGL) points. LGL points are applied to discretize the nonlinear optimal control problem into a nonlinear programming problem. It has been found that this method produces a more accurate solution than the global pseudospectral formation with less computation time and resources. A detailed description of the *hp*-adaptive pseudospectral method is provided in Ref. [32]. A sparse nonlinear optimization software named GPOPS is used to solve the problem. GPOPS is a state-of-the-art MATLAB toolbox that solves multistage optimal control problems using the *hp*-adaptive pseudospectral method. It can discretize multistage optimal control problems into large-scale nonlinear programming problems [33–35]. Therefore, GPOPS is well suited to solve the proposed trajectory optimization problem.

4.5. Algorithm implementation

The proposed distributed trajectory optimization algorithm is illustrated by the flow chart in Fig. 3. Specific steps are described as follows:

Step 1: In the first optimization (n=1), for each spacecraft $j = 1, 2, \dots, N_a$, the initial state $\boldsymbol{X}_{j,0}$ and terminal state $\boldsymbol{X}_{j,f}$ are known. Then the optimal trajectory without considering collision avoidance, $\boldsymbol{p}_j^{(1)^*}(t)$, $\forall t \in \left[t_0, t_f\right]$ is obtained by solving **Problem 1**.

Step 2: According to the initial adjacency matrix $\boldsymbol{A}_0$, each spacecraft j receives the optimal trajectory $\boldsymbol{p}_i^{(1)^*}(t)$, $i \in \mathbb{N}_j$, $\forall t \in \left[t_0, t_f\right]$ from its neighbors and transmits its own optimal trajectory $\boldsymbol{p}_j^{(1)^*}(t)$, $\forall t \in \left[t_0, t_f\right]$ to its neighbors. Fig. 4(a) shows the topology of information interaction between neighbors. Spacecraft j calculates the relative distance $d_{j,i}(t) = \left\|\boldsymbol{p}_j^{(n)^*}(t) - \boldsymbol{p}_i^{(n)^*}(t)\right\|_2$, $\forall t \in \left[t_0, t_f\right]$ with its neighbors to determine whether $d_{j,i}(t)$ is less than d_{safe}. If $d_{j,i}(t) > d_{safe}$, $\forall t \in \left[t_0, t_f\right]$ & $\forall i, j \in \mathbb{N}_a$, the reconfiguration is completed. Otherwise, the trajectory optimization needs to be carried out again.

Step 3: If it needs to be optimized again, first define the neighbor distance as $d_{neighbor}(d_{neighbor} > d_{safe})$. As shown in Fig. 4(b), when $d_{i,j}(t) > d_{neighbor}$, $\forall t \in \left[t_0, t_f\right]$, the trajectories are considered far apart from each other and there is no collision risk. Otherwise, the collision risk is considered, so they are neighbors to each other and continue to maintain information interaction during optimization. Update adjacency matrix $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{N_a \times N_a}$ and iteration $n = n+1$.

Step 4: For each spacecraft $j = 1, 2, \dots, N_a$, the nth($n \geq 2$) iteration considers the collision avoidance with the predicted trajectories $\hat{\boldsymbol{P}}_j^{(n)}(t) = \left[\hat{\boldsymbol{p}}_{i_1}^{(n)}(t), \dots, \hat{\boldsymbol{p}}_{i_{N_j}}^{(n)}(t)\right]^T, \forall t \in \left[t_0, t_f\right]$ of its neighbors, then obtains the optimal trajectory $\boldsymbol{p}_j^{(n)^*}(t)$, $\forall t \in \left[t_0, t_f\right]$ by solving **Problem 1**.

Step 5: According to the adjacency matrix $\boldsymbol{A}$, each spacecraft j receives the optimal trajectories $\boldsymbol{p}_i^{(n)^*}(t)$, $i \in \mathbb{N}_j$, $\forall t \in \left[t_0, t_f\right]$ from its

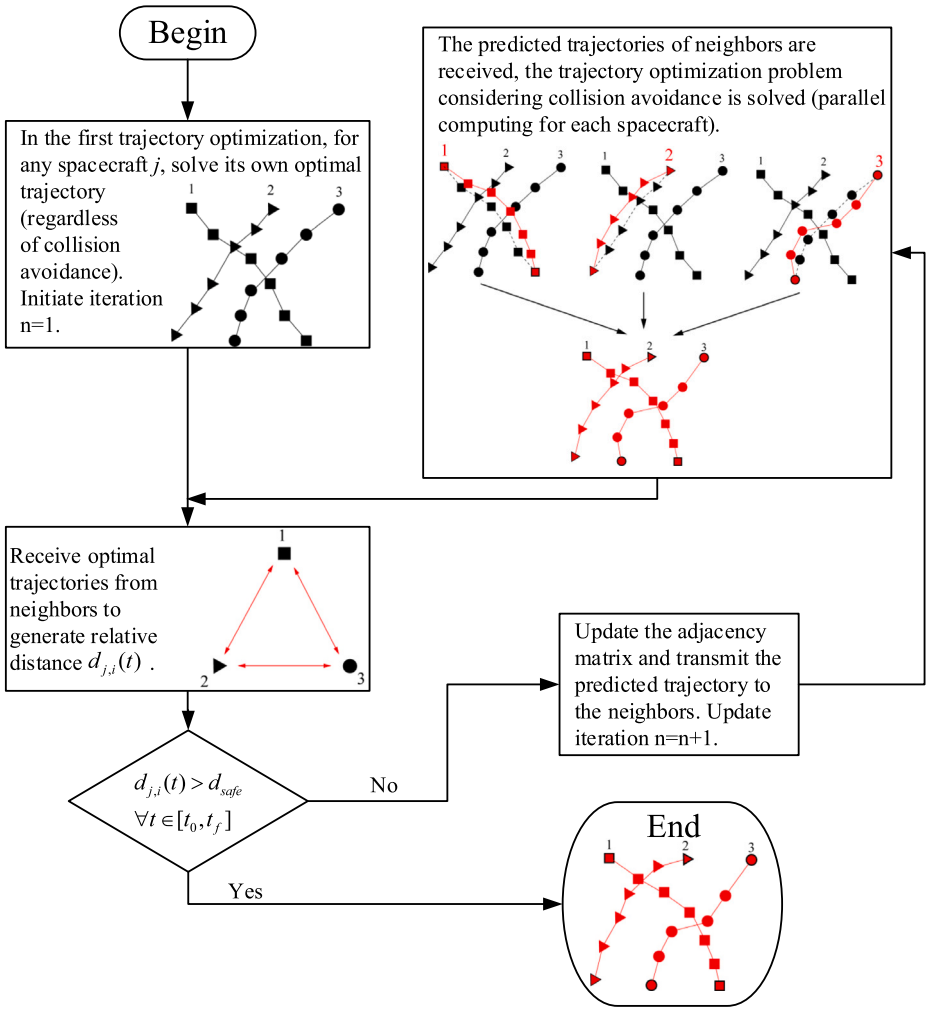


Fig. 3. Flow chart of the distributed trajectory optimization algorithm.

neighbors and transmits its own optimal trajectory to its neighbors. Spacecraft j calculates the relative distance $d_{j,i}(t)$, $\forall t \in [t_0, t_f]$ with its neighbors to determine whether $d_{j,i}(t)$ is less than d_{safe}. If $d_{j,i}(t) > d_{safe}$, $\forall t \in [t_0, t_f]$ & $\forall i, j \in \mathbb{N}_a$, the optimization is completed. Otherwise, return to **Step 3** until all $d_{j,i}(t) > d_{safe}$, $\forall t \in [t_0, t_f]$.

Finally, at the starting maneuver time t_0, the optimal control input sequence is applied to the spacecraft for open-loop guidance. The information to be transmitted in the algorithm is the optimal trajectory obtained by each spacecraft after each iteration to solve the optimization problem, i.e. $\boldsymbol{p}_j^{(n)^*}(t)$, $\forall t \in [t_0, t_f]$, which is only a series of discrete points. $\boldsymbol{p}_j^{(n)^*}(t)$ is transmitted to N_j neighbors of the spacecraft j. After over 1000 times of simulation calculation, it is found that even for the formation reconfiguration problem composed of hundreds of spacecraft, it can be solved after 2–3 iterations. With the increase of the number of iterations, the number of neighbors is decreased, and the amount of information to be transmitted is gradually reduced. Therefore, the information interaction required for the algorithm is feasible in engineering.

5. Numerical simulation

In this section, the numerical simulation of large-scale spacecraft formation reconfiguration is given. The effectiveness of the proposed distributed trajectory optimization algorithm for collision avoidance as well as the computational efficiency are mainly verified. The laptop computer performance used for simulation is: 16 GB of memory; processor 13th Gen Intel Core i9-13900HX GPU @2.20 GHz.

As shown in Fig. 5, the formation system consists of thirteen spacecraft, including one chief spacecraft and twelve deputy spacecraft. Among them, four spacecraft in the inner circle are evenly distributed on the $r_{01} = 500$ m space circle, and eight spacecraft in the outer circle are evenly distributed on the $r_{02} = 750$ m space circle. The twelve spacecraft are reconfigured to a in-plane formation. The initial and terminal states of each spacecraft in the LVLH frame are shown in Table 1. Table 2 gives the parameters needed for simulation. Table 3 gives the initial orbital elements of the chief spacecraft.

5.1. Initial guess of adjacency matrix

For the proposed distributed optimization algorithm, collision avoidance between neighbors is considered, and the updated adjacency matrix is based on the initial adjacency matrix $\boldsymbol{A}_0$. Therefore, to ensure the effectiveness of collision avoidance, the initial guess of the adjacency matrix is very important. For initial adjacency matrix $\boldsymbol{A}_0$, spacecraft with high collision probability should be neighbors as far as possible to ensure the effectiveness of collision avoidance. At the same time, spacecraft with low probability of collision should not be allowed to become neighbors, which can reduce computation. For this simulation, spacecraft with initial distance and terminal distance less than $2d_{safe} = 300$ m are neighbors, and the information interaction

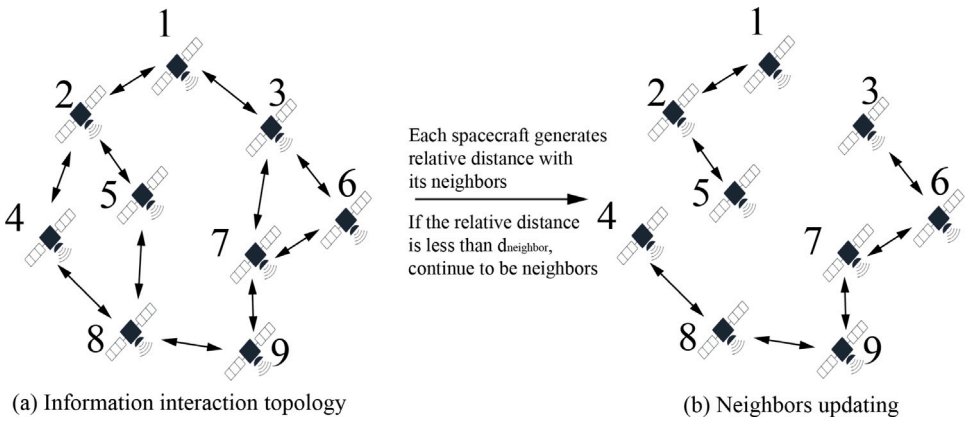


Fig. 4. Information interaction topology and neighbors updating.

Table 1
Initial and terminal states of the spacecraft formation (S.I. unit).

Spacecraft	Initial states	Initial states
1	[250;0;433; 0;−0.551;0]	[0;−900;0;0;0;0]
2	[0;−500;0; −0.275;0;−0.477]	[0;100;0;0;0;0]
3	[−250;0;−433; 0;0.551;0]	[0;900;0;0;0;0]
4	[0;500;0; 0.275;0;0.477]	[0;−100;0;0;0;0]
5	[375;0;649.5; 0;−0.826;0]	[0;−700;0;0;0;0]
6	[265.2;−530.3;459.3; −0.29;−0.58;−0.51]	[0;−1100;0;0;0;0]
7	[0;−750;0; −0.413;0;−0.76]	[0;−300;0;0;0;0]
8	[−265.2;−530.3;−459.3; −0.29;0.58;−0.51]	[0;500;0;0;0;0]
9	[−375;0;−649.5; 0;0.826;0]	[0;700;0;0;0;0]
10	[−265.2;530.3;−459.3; 0.29;0.58;0.51]	[0;1100;0;0;0;0]
11	[0;750;0; 0.413;0;0.716]	[0;300;0;0;0;0]
12	[265.2;530.3;459.3; 0.29;0.58;0.51]	[0;−500;0;0;0;0]

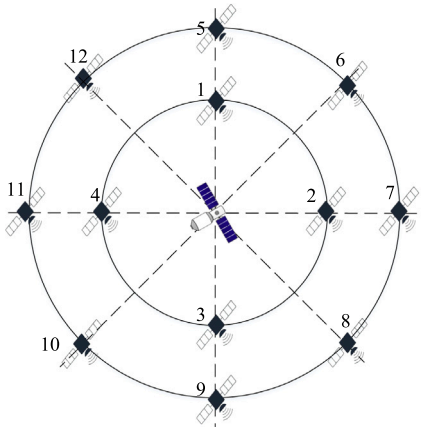


Fig. 5. Initial spacecraft formation topology network model.

Table 2
Parameters of simulation.

Parameters	Values
Reconfiguration time: t_f	3000 s
LGL points	59
Safe distance: d_{safe}	150 m
Neighbor distance: $d_{neighbor}$	180 m
Mass of the deputy spacecraft: m_0	100 kg
Maximum thrust: F_{max}	50 mN
Minimum thrust: F_{min}	2 mN
Thruster specific impulse: I_s	2000 s

Table 3
Initial orbital elements of chief spacecraft.

Orbit elements	Values
Semi-major axis	6900 km
Eccentricity	1.2e−03
Inclination	53 deg
Argument of periapsis	45 deg
Right ascension of the ascending node	50.2 deg
True anomaly	0 deg

topology between initial neighbors is shown in Fig. 6(a). The values of each element of $\boldsymbol{A}_0$ are as follows:

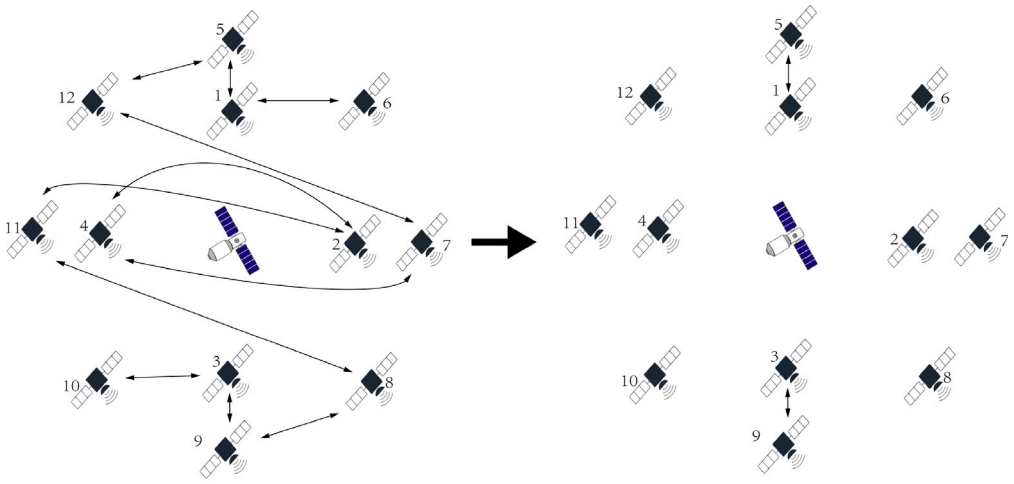
$$a_{15} = a_{51} = 1, a_{16} = a_{61} = 1, a_{24} = a_{42} = 1, a_{2,11} = a_{11,2} = 1,$$
$$a_{39} = a_{93} = 1, a_{3,10} = a_{10,3} = 1$$
$$a_{47} = a_{74} = 1, a_{5,12} = a_{12,5} = 1, a_{7,12} = a_{12,7} = 1,$$
$$a_{89} = a_{98} = 1, a_{8,11} = a_{11,8} = 1$$

all other elements are zero.

5.2. Collision avoidance effectiveness

In the simulation, the optimal trajectory satisfying the collision avoidance constraints is obtained by solving the optimization problem through two iterations.

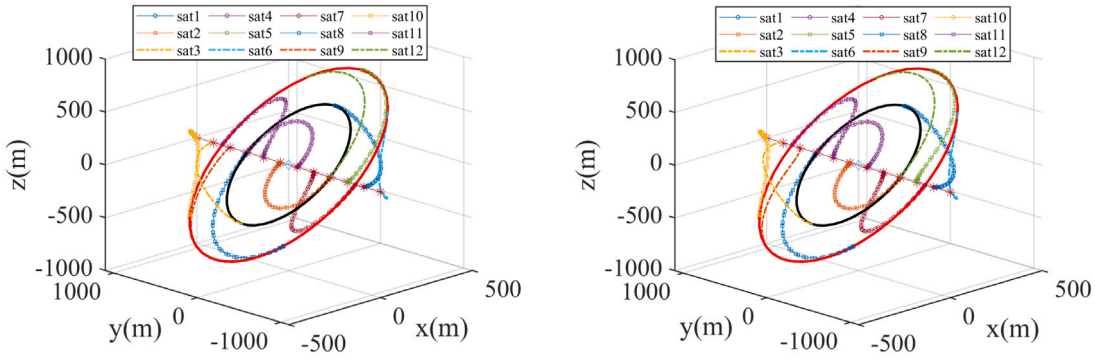
The reconfiguration trajectories of the first and second optimization are shown in Fig. 7. After the first optimization, each spacecraft transmits the optimal trajectory to the neighbors according to the initial adjacency matrix $\boldsymbol{A}_0$, and receives the optimal trajectories from the neighbors at the same time, thus generating the relative distances with the neighbors. Because of the limited space, only the result that the distances with the neighbors are less than d_{safe} is given. Fig. 8 show the relative distances of spacecraft 1 and 3 with their neighbors after the



(a) Information interaction topology of initial neighbors

(b) Information interaction topology after updating neighbors

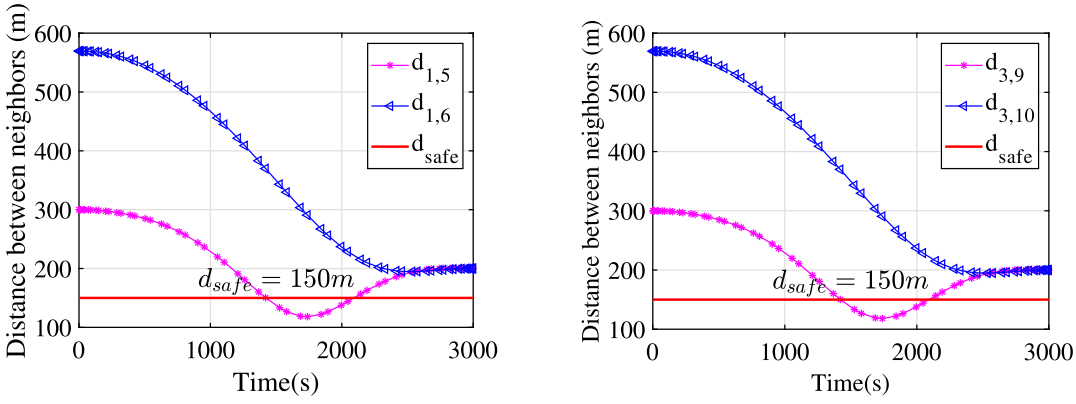
Fig. 6. Initial neighbors information interaction topology and neighbors update.



(a) First optimization

(b) Second optimization

Fig. 7. Spacecraft trajectories for formation reconfiguration.

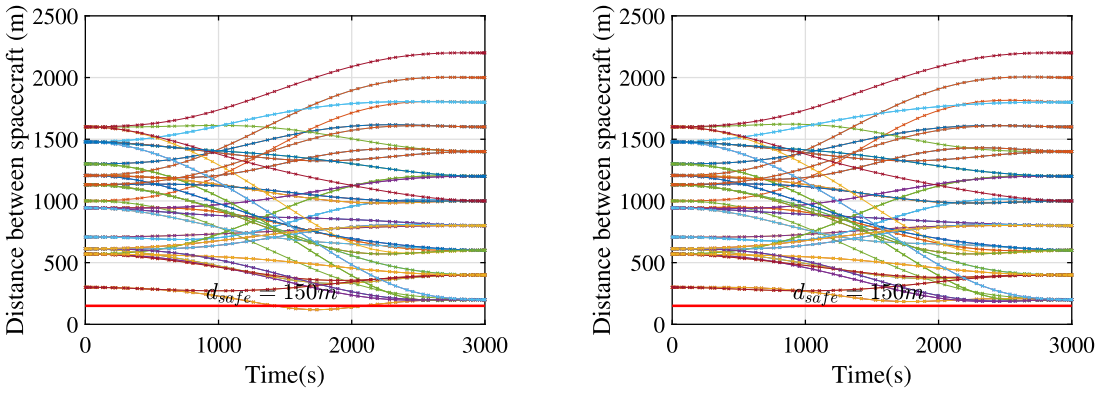


(a) Distances between spacecraft 1 and its neighbors

(b) Distances between spacecraft 3 and its neighbors

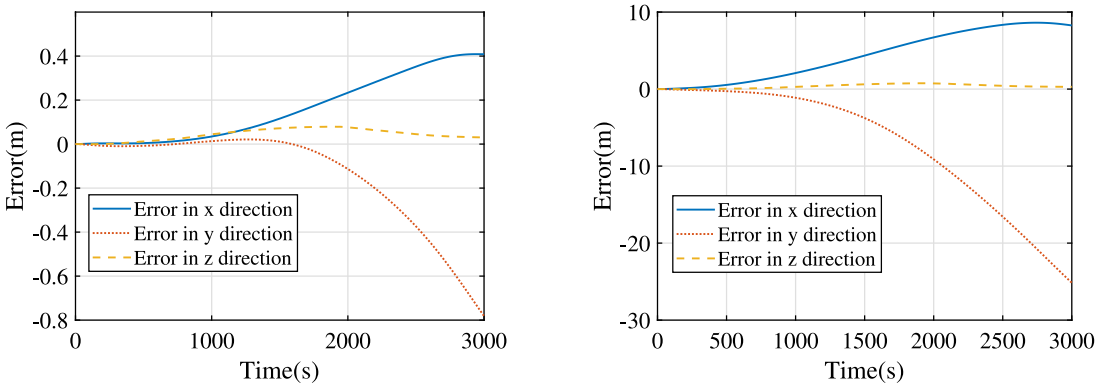
Fig. 8. Relative distances between some spacecraft and their neighbors after first optimization.

first optimization. It can be seen that the distance between spacecraft 1 and 5 is less than d_{safe} and that between spacecraft 3 and 9 is less than d_{safe}. In the simulation, only the distances between neighbors are used to determine whether there is collision risk in the formation system. For actual trajectory, if the distances between non-neighbors are less than d_{safe}, this situation will not be discovered, resulting in no guarantee for the effectiveness of collision avoidance. Fortunately, at the beginning, spacecraft with high collision probability became neighbors, making the collision risk between non-neighbors very low. The statistical result of distances between all spacecraft is called global statistics, and the



(a) First optimization (b) Second optimization

Fig. 9. Relative distances between all spacecraft during formation reconfiguration.



(a) the case with J_2 perturbation (b) the case without J_2 perturbation

Fig. 10. Comparison of relative position error for the cases with or without considering J_2 perturbation.

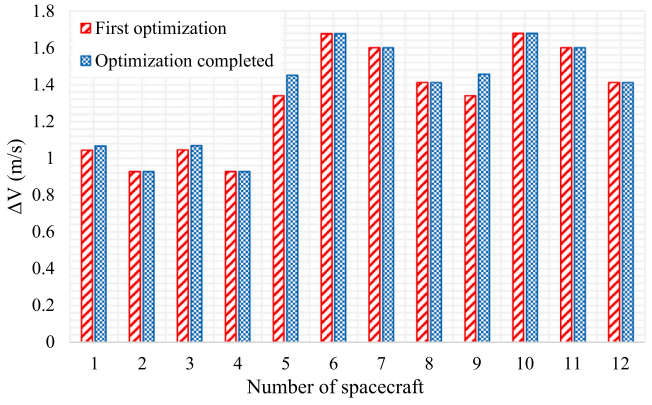


Fig. 11. ΔV comparison of two optimizations.

statistical result of distances between spacecraft and their neighbors is called local statistics. To prove that the result of local statistics is the same as the global statistics, the global statistical result after the first optimization is shown in Fig. 9(a), totaling $(12 \times 11)/2 = 66$ groups of distances. Statistically, only spacecraft 1 and 5, as well as spacecraft 3 and 9, have distances less than d_{safe}, which are the same as the result of local statistics.

For the second trajectory optimization, the adjacent matrix $\boldsymbol{A}$ is updated using the distance $d_{j,i}(t) > d_{neighbor}$, $\forall t \in [t_0, t_f]$, $j \neq i$ obtained by each spacecraft in the previous iteration. If $d_{j,i}(t) > d_{neighbor}$, $\forall t \in [t_0, t_f]$, consider that there is no collision risk, cancel the neighborhood, otherwise continue to be neighbors. The updated information interaction topology is shown in Fig. 6(b). It is clear that the number of information interaction has been reduced from 11 sets at the first optimization to only 2 sets at the second optimization. It can be concluded that the communication resources of formation system can be significantly reduced by updating the information interaction topology with the current optimal trajectory. The updated adjacency matrix $\boldsymbol{A}$ is a sparse matrix with only $a_{15} = a_{51} = 1$, $a_{39} = a_{93} = 1$ and all other elements are zero.

According to adjacency matrix $\boldsymbol{A}$, it can be seen that after the first optimization, only spacecraft 1 and 5, and spacecraft 3 and 9 have distances less than $d_{neighbor}$. Therefore, their collision avoidance needs to be considered in the second trajectory optimization. Taking spacecraft 1 and 5 as examples, in the second optimization, they transmit their "predicted trajectory" to each other, and then solve a trajectory optimization problem considering collision avoidance with the neighbor's predicted trajectory. The relative distances of all spacecraft after the second optimization are shown in Fig. 9(b). It is obvious that all the distances are larger than d_{safe} after the second optimization, which realizes collision avoidance. Finally, after two trajectory optimization, the fuel optimal trajectory realizing the collision avoidance is obtained for the formation system.

The control acceleration generated by the *hp*-adaptive pseudospectral method is used for open-loop guidance of the formation reconfiguration process. Since the *hp*-adaptive pseudospectral method only gives state estimation and control at LGL points, the control between LGL

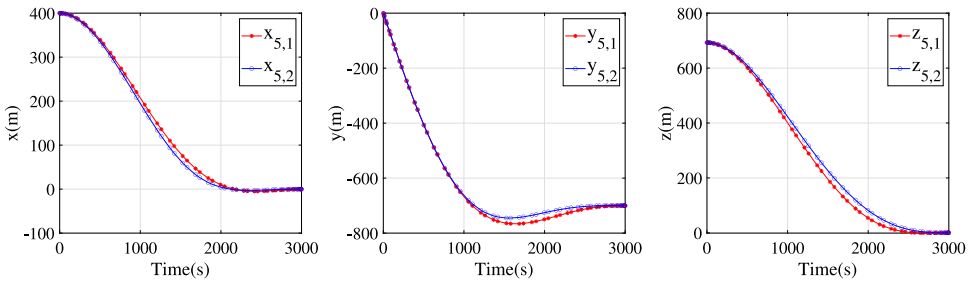


Fig. 12. Comparison of relative position trajectory of spacecraft 5 of two optimizations.

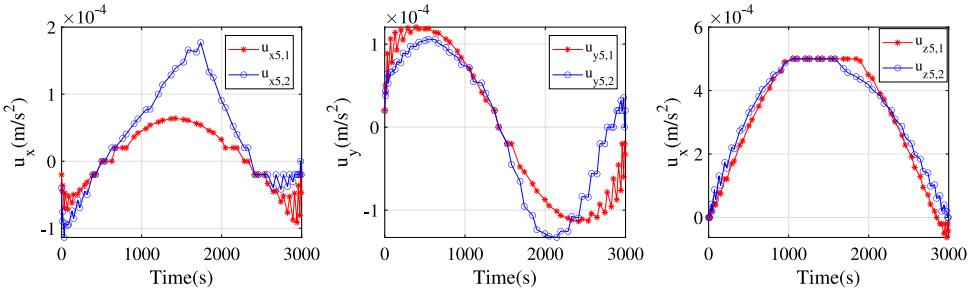


Fig. 13. Comparison of control acceleration of spacecraft 5 of two optimizations.

points is obtained by using cubic spline interpolation. In the simulation, the open loop optimal control of the spacecraft is not modified, and the actual trajectory is generated by using high-precision orbital propagator (HPOP) integration. The HPOP model includes the gravitational field model EGM2008 21*21, the atmospheric model NRLMSISE-2000, considering the solar and lunar third body force and the spherical solar radiation pressure (SRP). The error between the actual trajectory and the optimized trajectory is shown in Fig. 10(a). For comparison, another trajectory without considering J_2 perturbation is generated. The error between the actual trajectory and the optimized trajectory is shown in Fig. 10(b). It can be seen from the results that the control error considering J_2 perturbation is much smaller than that without J_2 perturbation. Therefore, it can be concluded that the nonlinear relative spacecraft dynamics considering J_2 perturbation is necessary to obtain the accurate trajectory of the spacecraft formation maneuver.

Fig. 11 shows the velocity increment ΔV comparison between the first optimization and the optimization completed for each spacecraft. It can be seen that the ΔV of spacecraft 1, 3, 5 and 9 have changed compared with the first optimization, which is also consistent with the previous analysis. In the second trajectory optimization, spacecraft 1, 3, 5 and 9 adjust the maneuvering strategy to avoid collision with neighbors by considering predicted trajectories of neighbors.

Taking spacecraft 5 as an example, the change of maneuver trajectory during the optimization process is analyzed in detail next. Figs. 12 and 13 show the comparison of the two optimization results. The red asterisk curve is the first optimization result of spacecraft 5 and the blue circle curve is the second optimization result. Wherein, the comparison of relative position trajectory of two optimizations is shown in Fig. 12, and the comparison of optimal control acceleration is shown in Fig. 13. It can be seen that in order to achieve collision avoidance, the control acceleration of spacecraft 5 in the x and y directions has changed greatly, which makes the radial and in-track relative position significantly different from those in the first optimization. At the same time, fine-tuning is made in the cross-track (z) direction, enabling spacecraft 5 to reach the terminal state at the end time, while avoiding collisions with neighbor.

Table 4 shows the comparison of fuel consumption Δm between the first optimization and the optimization completed, as well as the comparison of ΔV. For the first optimization, ΔV = 15.996 m/s, and

Table 4
Fuel consumption comparison between first optimization and optimization completed.

Cases	Δm, g	ΔV, m/s
First optimization	81.56	15.996
Optimization completed	82.77	16.233

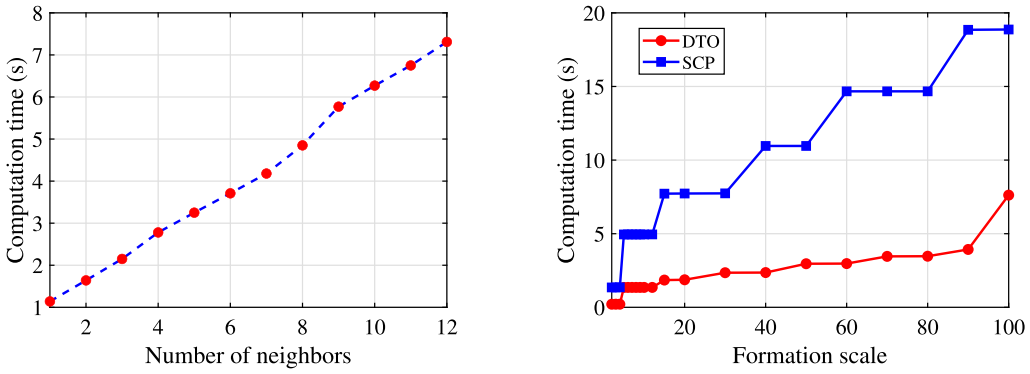
when the optimization is completed, ΔV = 16.233 m/s. ΔV increases by only 1.48% compared to the first optimization. Due to the high specific impulse of the low thrust engines, only 1.21 g of extra fuel is consumed to avoid collision. Therefore, it can be concluded that the proposed trajectory optimization algorithm can achieve safe formation reconfiguration maneuver with little additional fuel consumption.

5.3. Comparison of computation time

Due to the parallel computation of the distributed algorithm, the average computation time for each spacecraft to solve the optimization problem for the first and second iteration is statistically analyzed, as shown in Table 5. For the first optimization, the computation time is only 0.209 s, and for the subsequent optimization solution of collision avoidance by considering "predicted trajectory" with neighbors, the computation time is 1.141 s. For this numerical simulation, the optimal trajectory for collision avoidance is obtained after two optimizations, and the total computation time is only 1.35 s.

By calculating over 1000 numerical simulations, the computation time of different number of neighbors for **Problem 1** is shown in Fig. 14(a). It can be seen that the computation time increases linearly with the number of neighbors, i.e. the computation time complexity is $\mathcal{O}(N_j)$, which verifies the analysis in Section 4.3. It is easy to know that even a large-scale formation of thousands of spacecraft, each spacecraft can only collide with several or more spacecraft around it (become neighbors). Therefore, the distributed algorithm can quickly solve the trajectory optimization problem of large-scale spacecraft formation.

Next, we illustrate the rapidity of the proposed algorithm by comparison. The decentralized sequential convex programming (SCP) algorithm proposed by Ref. [15] and the distributed trajectory optimization



(a) Computation time of different number of neighbors for Problem 1

(b) Comparison of computation time between both algorithms

Fig. 14. Computation time statistics (calculated over 1000 numerical simulations).

Table 5
Average computation time of trajectory optimization.

Cases	Computation time (s)
First optimization	0.209
Second optimization	1.141
Optimization completed	1.350

(DTO) algorithm are used to solve the formation reconfiguration problems of different scales respectively. The simulation parameters are the same as Table 2, and the collision-free optimal trajectory is obtained by solving the respective optimization problems. The average computation time is shown in Fig. 14(b). It can be clearly seen that regardless of the scale of the formation, the computation time of DTO is less than that of SCP. This is because DTO can be calculated in parallel, while SCP requires adjacent spacecraft to solve the optimization problem according to the sequential. With the increase of the formation scale, the advantage of DTO rapidity is more obvious and more suitable for engineering application. In addition, the computation time of the DTO algorithm is also significantly less than the centralized algorithm of Ref. [12] and the convex optimization method of Ref. [18] through verification.

6. Conclusions

In this paper, we present a distributed trajectory optimization algorithm considering collision avoidance for large-scale spacecraft formation reconfiguration. The nonlinear relative spacecraft dynamics considering J_2 perturbation is used to obtain the high-precision maneuver trajectory. The "predicted trajectory" is introduced to decouple the state variables of each spacecraft, so that each spacecraft solve its own trajectory optimization problem independently, which is the main reason for the improvement of computational efficiency. In addition, the collision avoidance convexification ensures the accuracy of the solution and improves the computational efficiency. Finally, the nonlinear optimal control problem is discretized by the *hp*-adaptive pseudospectral method. Numerical simulation results validate that the proposed algorithm has higher computational efficiency than the centralized and decentralized algorithms, and collision avoidance is achieved.

The distributed algorithm provides an optimal open-loop solution for spacecraft formation maneuver. However, further research is needed to make the proposed distributed algorithm adapt to onboard real-time guidance and control. It must be combined with the closed-loop control design to ensure that the optimal trajectory can be obtained in the case of system uncertainty and external disturbances. Our future research work will focus on using the proposed distributed algorithm to solve this closed-loop control problem.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 62188101).

References

[1] Z. Zhang, L. Deng, J. Feng, L. Chang, D. Li, Y. Qin, A survey of precision formation relative state measurement technology for distributed spacecraft, Aerospace 9 (7) (2022) 362.
[2] D. Gu, B. Ju, J. Liu, J. Tu, Enhanced GPS-based GRACE baseline determination by using a new strategy for ambiguity resolution and relative phase center variation corrections, Acta Astronaut. 138 (2017) 176–184.
[3] Y. Ding, J. Yang, Y. Hao, A review on coordinated control of formation configuration of space solar power station energy transmission system, J. Aeronaut., Astronaut. Aviat. 54 (1) (2022) 49–65.
[4] G. Ma, H. Dong, Q. Hu, Fault-tolerant translational control for spacecraft formation flying with collision avoidarce requirement, Acta Aeronaut. et Astronaut. Sinica 38 (10) (2017) 206–216.
[5] J. Wang, J. Zhang, X. Cao, F. Wang, Optimal satellite formation reconfiguration strategy based on relative orbital elements, Acta Astronaut. 76 (2012) 99–114.
[6] Z. Wang, Y. Xu, C. Jiang, Y. Zhang, Self-organizing control for satellite clusters using artificial potential function in terms of relative orbital elements, Aerosp. Sci. Technol. 84 (2019) 799–811.
[7] J. Wang, C. Zhang, J. Zhang, Analytical solution of satellite formation impulsive reconfiguration considering passive safety constraints, Aerosp. Sci. Technol. 119 (2021) 107108.
[8] C. Lippe, S. D'Amico, Safe, delta-v-efficient spacecraft swarm reconfiguration using Lyapunov stability and artificial potentials, J. Guid. Control Dyn. 45 (2) (2022) 213–231.
[9] J.T. Betts, Practical methods for optimal control and estimation using nonlinear programming, Soc. Ind. Appl. Math., SIAM Rev. 53 (2010) 183–185.
[10] H.K. Sipowa, J. McMahon, Fuel-optimal geometric path planning algorithm for spacecraft formation flying, J. Guid. Control Dyn. 45 (10) (2022) 1862–1872.
[11] N. Pillet, P. Bousquet, E. Chesta, R. Cledassou, M. Delpech, E. Hinglais, Propulsion options for preliminary formation flying missions studies at CNES, in: 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2006, p. 5221.
[12] B. Wu, D. Wang, E.K. Poh, G. Xu, Nonlinear optimization of low-thrust trajectory for satellite formation: Legendre pseudospectral approach, J. Guid. Control Dyn. 32 (4) (2009) 1371–1381.

[13] A. Richards, T. Schouwenaars, J.P. How, E. Feren, Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming, J. Guid. Control Dyn. 25 (4) (2002) 755–764.
[14] G.D. Mauro, D. Spiller, S.F.R. Carnà, R. Bevilacqua, Minimum-fuel control strategy for spacecraft formation reconfiguration via finite-time maneuvers, J. Guid. Control Dyn. 42 (4) (2019) 752–768.
[15] D. Morgan, S.J. Chung, F.Y. Hadaegh, Model predictive control of swarms of spacecraft using sequential convex programming, J. Guid. Control Dyn. 37 (6) (2014) 1725–1740.
[16] S. Sarno, M. D'Errico, J. Guo, E. Gill, Path planning and guidance algorithms for SAR formation reconfiguration: Comparison between centralized and decentralized approaches, Acta Astronaut. 167 (2020) 404–417.
[17] B. Li, H. Zhang, W. Zheng, L. Wang, Spacecraft close-range trajectory planning via convex optimization and multi-resolution technique, Acta Astronaut. 175 (2020) 421–437.
[18] F. Scala, G. Gaias, C. Colombo, M. Martín-Neira, Design of optimal low-thrust manoeuvres for remote sensing multi-satellite formation flying in low Earth orbit, Adv. Space Res. 68 (11) (2021) 4359–4378.
[19] T. Timmons, J. Beeley, G. Bailet, C.R. Mclnnes, Range-based relative navigation for a swarm of centimeter-scale femto-spacecraft, J. Guid. Control Dyn. 45 (9) (2022) 1583–1597.
[20] L. An, G. Yang, Collisions-free distributed optimal coordination for multiple Euler-Lagrangian systems, IEEE Trans. Automat. Control 67 (1) (2021) 460–467.
[21] P. Wang, B. Ding, Distributed RHC for tracking and formation of nonholonomic multi-vehicle systems, IEEE Trans. Automat. Control 59 (6) (2014) 1439–1453.
[22] L. Chen, H. Duan, Collision-free formation-containment control for a group of UAVs with unknown disturbances, Aerosp. Sci. Technol. 126 (2022) 107618.
[23] Y. Lyu, J. Hu, B. Chen, C. Zhao, Q. Pan, Multivehicle flocking with collision avoidance via distributed model predictive control, IEEE Trans. Cybern. 51 (5) (2019) 2651–2662.
[24] S. Silvestrini, M. Lavagna, Neural-based predictive control for safe autonomous spacecraft relative maneuvers, J. Guid. Control Dyn. 44 (12) (2021) 2303–2310.
[25] Y. Guo, J. Zhou, Y. Liu, Distributed RISE control for spacecraft formation reconfiguration with collision avoidance, J. Franklin Inst. B 356 (10) (2019) 5332–5352.
[26] X. Xue, X. Yue, J. Yuan, Connectivity preservation and collision avoidance control for spacecraft formation flying in the presence of multiple obstacles, Adv. Space Res. 67 (11) (2021) 3504–3514.
[27] S. Sarno, J. Guo, M. D'Errico, E. Gill, A guidance approach to satellite formation reconfiguration based on convex optimization and genetic algorithms, Adv. Space Res. 65 (8) (2020) 2003–2017.
[28] C. Vela, R. Opromolla, G. Fasano, A low-thrust finite state machine based controller for N-satellites formations in distributed synthetic aperture radar applications, Acta Astronaut. 202 (2023) 686–704.
[29] D. Wang, B. Wu, E.K. Poh, Satellite Formation Flying: Relative Dynamics, Formation Design, Fuel Optimal Maneuvers and Formation Maintenance, Springer, Singapore, 2017.
[30] F. Di, A. Li, Y. Guo, C. Xie, C. Wang, Event-triggered sliding mode attitude coordinated control for spacecraft formation flying system with disturbances, Acta Astronaut. 188 (2021) 121–129.
[31] P. Gurfil, Relative motion between elliptic orbits: Generalized boundedness conditions and optimal formation keeping, J. Guid. Control Dyn. 28 (4) (2005) 761–767.
[32] C.L. Darby, W.W. Hager, A.V. Rao, An hp-adaptive pseudospectral method for solving optimal control problems, Optim. Control Appl. Methods 32 (4) (2011) 476–502.
[33] D.A. Benson, G.T. Huntington, T.P. Thorvaldsen, A.V. Rao, Direct trajectory optimization and costate estimation via an orthogonal collocation method, J. Guid. Control Dyn. 29 (6) (2006) 1435–1440.
[34] D. Garg, W.W. Hager, A.V. Rao, Pseudospectral methods for solving infinite-horizon optimal control problems, Automatica 47 (4) (2011) 829–837.
[35] M.A. Patterson, A.V. Rao, GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming, ACM Trans. Math. Softw. 41 (1) (2014) 1–37.