Probabilistic swarm guidance enables autonomous microsatellites to generate their individual trajectories independently so that the entire swarm converges to the desired distribution shape. However, it is essential to avoid crowding for reducing the possibility of collisions between microsatellites. To determine the collision-free guidance trajectory of each microsatellite from the current position to the target space, a collision avoidance algorithm is necessary. A synthesis method is proposed that generates the collision avoidance trajectories. The idea is that the trajectory planning is divided into macro-planning and micro-planning; macro-planning guides where the microsatellites move step by step from the initial cube to the target cube by probabilistic swarm guidance with Centroidal Voronoi tessellation, while the micro-planning is to generate the optimal path for each step and finally reach the specified position in the target cube by model predictive control. Simulation results are presented for the collision-free guidance trajectory of microsatellites to verify the benefits of this planning scheme.

1. Introduction

As an emerging multi-satellite cooperative flight mode, microsatellite swarm has become an important future research issue for distributed space systems due to their advantages of low cost, rapid response, and collaborative decision-making. Recently, reference [1] develops hundreds or thousands of small femtosatellites and establishes effective swarm GNC strategies for potential synthetic aperture applications. The problem of shaping a large swarm ($10^3$–$10^5$) of particles into a large reflective dish was addressed [2–3], with only a few (40) actuators in the boundary of the workspace. Due to their small size, the microsatellites have limited sensing, actuation, and computation capabilities, which require the guidance and control algorithms of the swarm to be both fuel and computationally efficient.

To address the coordination of swarms for autonomous agents, a probabilistic guidance approach was investigated [4–6], which contain sub-swarms with different mission objectives. The probabilistic guidance approach is based on designing a Markov chain such that its steady-state distribution corresponds to the desired swarm density. On the basis of the above method, a homogeneous Markov chain-based approach was presented for the probabilistic density control of swarm. The proposed approach specifies the time evolution of the probabilistic density distribution by using a Markov chain, which guides the swarm to a desired steady-state distribution, while satisfying the prescribed ergodicity, motion, and safety constraints [7–11]. Different from the above method based on homogeneous Markov chain is that a novel and generic distributed swarm guidance algorithm using inhomogeneous Markov chain that guarantees superior performance over existing homogeneous Markov chain based algorithms, when the feedback of the current swarm distribution is available [12–16]. In contrast with previous homogeneous or inhomogeneous Markov chain–based approaches, optimal transport based approach was presented which guarantees faster convergence, minimizes a given cost function, and reduces the number of transitions for achieving the desired formation [2, 3, 17]. Moreover, the optimal transport based approach was verified by the Robotarium platform [2], and Jet Propulsion Laboratory of NASA has developed a guidance algorithm for a swarm of assets deployed from the back shell of the Mars spacecraft.
for distributed science [18]. All these cases indirectly show that the probabilistic guidance method can be applied to the practical swarm satellite mission.

The solution of the probabilistic guidance algorithm establishes the target bin for each agent. Collision avoidance algorithms are necessary for generating collision-free guidance trajectories from the present location to the target. The macro-trajectories from the present location to the target are established by using a numerical example in Section 5. Some conclusions and future work are given in Section 6.

2. Problem Formulation

2.1. Probabilistic Swarm Guidance. According to the probabilistic swarm guidance method described in literature [4], the agents can move in 2D plane. In this paper, we will investigate the transfer of swarm microsatellites in 3D space. The physical space over which the swarm microsatellites are distributed is denoted as $\mathbb{R}^3$; it is assumed that the space $\mathbb{R}^3$ is divided into a union of $m \times n \times l$ disjoint subspaces $R_{\Omega}$, $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{mnl}\}, c = 1, \ldots, m \times n \times l$, that is, $\mathbb{R}^3 = \bigcup_{c=1}^{mnl} R_{\Omega}$. For $c_i \neq c_j$, $R_{\Omega_i} \cap R_{\Omega_j} = \emptyset$. The subspaces $R_{\Omega_i}$ are referred to as cubes.

Let $r(t)$ be the position of a microsatellite at time index $t$. Let $x(t)$ be a probability vector so that $x(\Omega_i)(t)$ is the probability of the event that the microsatellite will be in cube $R_{\Omega_i}$ at time $t$,

$$x(\Omega_i)(t) = P(r(t) \in R_{\Omega_i}).$$

Consider a swarm consisting of $N$ independently acting microsatellites so that (1) holds for $N$ independent events:

$$x(\Omega_i)(t) = P(r_k(t) \in R_{\Omega_i}), k = 1, \ldots, N. \quad (2)$$

where $r_k(t)$ denotes the position of the $k$-th microsatellite at time $t$. Here, $x(\Omega_i)(t)$ is called the swarm distribution.

The distribution guidance problem is defined as follows: Given any initial probability distribution, the swarm must be guided to a specified steady-state probability distribution $v$:

$$\lim_{t \rightarrow \infty} x(\Omega_i)(t) = v(\Omega_i). \quad (3)$$

Subject to motion constraints, specified by the adjacency matrix $A_a$ as

$$A_a^T [\Omega_{c_i}, \Omega_{c_j}] = 0 \Rightarrow r(t + 1) \notin R_{\Omega_i}, r(t) \in R_{\Omega_i}, \forall t. \quad (4)$$

The idea behind probabilistic swarm guidance is to control the propagation of probability vectors $x(\Omega_i)$ rather than the individual microsatellite positions $\{r_k(t)\}_{k=1}^N$.

The probability density vector $x(t)$ over $\mathbb{R}^3$ evolves as

$$x(t + 1) = Mx(t). \quad (5)$$

The probabilistic guidance algorithm is implemented by providing the Markov matrix $M$ to each microsatellite. The probabilistic guidance algorithm can be referenced [4, 5].

2.2. Safety Analysis of Collision Avoidance. In the method of probabilistic swarm guidance, it is essential to avoid crowdng for reducing the possibility of collisions between microsatellites. To determine the collision-free guidance...
trajectory of each microsatellite from the current position to the target space, a collision avoidance algorithm is needed. However, with high-level coordination that uses the macroscopic models, collision-free trajectories are very hard to generate. In this paper, a synthesis method was presented that the collision avoidance trajectories are generated. The idea is that the trajectory planning is divided into macro-planning and micro-planning, as shown in Figure 1, where macro-planning guides the microsatellites move step by step from the initial cube to the target cube, while the micro-planning is to generate the optimal path for each step and finally reach the specified position in the target cube.

Collision avoidance needs to be considered during trajectory generation. The minimum collision avoidance distance between all microsatellites is required, represented by $R_{col}$. Collision avoidance analysis is based on finding the lower bound $R_{col}$ of the minimum distance between all microsatellites at any time. Suppose the distance between two microsatellites is $|\eta_{ij}|$, so the difference between it and the minimum safe distance is $\gamma_{ij}(\eta_{ij}) = R_{col} - |\eta_{ij}|$, then

$$
\nabla_{\eta_{ij}} \gamma_{ij}(\eta_{ij})^T \eta_{ij} \leq 0
$$

$$
\nabla_{\eta_{ij}} (R_{col} - |\eta_{ij}|)^T \eta_{ij} \leq 0
$$

$$
\nabla_{\eta_{ij}} R_{col}^T - \nabla_{\eta_{ij}} |\eta_{ij}|^T \eta_{ij} \leq 0
$$

$$
\Rightarrow - \frac{1}{2} \frac{d}{dt} ||\eta_{ij}||^2 \eta_{ij} \leq 0
$$

therefore, $\eta_{ij}^T \eta_{ij} \geq 0, \forall (i, j)$, define

$$
\lambda_{ij}(t) = \frac{1}{2} \frac{d}{dt} \left( \eta_{ij}^T \eta_{ij} \right).
$$

The positive invariance condition can be expressed as

$$
||\eta_{ij}(t)||^2 = R_{col} \Rightarrow \lambda_{ij}(t) \geq 0, \forall (i, j).
$$

When the distance between any microsatellites is less than the lower bound of $R_{col}$, the time derivative of the distance function can maintain or increase the distance, so collision avoidance can be guaranteed.

3. Macro-Planning of Microsatellite Swarm

3.1. Voronoi Tessellation. For convex set $Q \subset \mathbb{R}^d$, $P = \{p_1, \cdots, p_N\} \subset \mathbb{R}^{dN}$ is defined as the set of $n$ distinct points in set $Q$. The Voronoi cell $V_i$ corresponding to the generation point $p_i$ is defined as follows:

$$
V_i = \left\{ q \in Q \mid ||q - p_i|| \leq ||q - p_j||, i, j = 1, 2, 3, \cdots, k, j \neq i \right\},
$$

where $||\cdot||$ denotes the Euclidean distance in $\mathbb{R}^d$. From the above definition, it can be seen that all points in $Q$ whose Euclidean distance to the generation point $p_i$ is less than to the generation point $p_{j,i}$ and the points are included in the Voronoi cells $V_i$. For $i \neq j$, $V_i \cap V_j = \emptyset$. Except for the boundary, all Voronoi cells are disjoint, and $Q = \bigcup_{i=1}^{N} V_i$ is true, that is, the set $V$ is a partition of the set $Q$.

3.2. Centroidal Voronoi Tessellation. When the function of the target space is set, its generation point is the centroid of each Voronoi cell (the density function is used to calculate the centroid); it is the Centroidal Voronoi tessellation.

Assuming that the density function of space $Q$ is defined as $\rho(q) \geq 0$, then the centroid of each Voronoi cell is positioned as

$$
\rho_{i}^* = \frac{\int_{V_i} \rho(q) dq}{\int_{V_i} \rho(q) dq}, i = 1, \cdots, k.
$$

When the position of the generation point is the centroid, namely:

$$
p_i = p_i^*, i = 1, 2, 3, \cdots, k,
$$

is Centroidal Voronoi tessellation.

When the cost function is less than a certain threshold, it can be judged that the Centroidal Voronoi tessellation tends to be stable. Given the density function, the cost function is
The cost function estimates the error between the generation point \( p_i \) and the centroid of the Voronoi tessellation by the density function of the space.

3.3. CVT-Based Trajectory Planning for Microsatellite Swarm. The target position of each microsatellite is determined by the centroid generated through CVT algorithm, and all microsatellites move to the corresponding centroid until the algorithm converges. In the present work, a code library of CVT based on the probabilistic Lloyd’s method is incorporated to produce the CVT generators. According to the location of the centroid, the final distribution of the microsatellite swarm in the space is obtained. The main algorithm step is as follows:

1. Select the \( k \) points \( \{ p_j \}_{i=1}^k \) randomly as generated points, namely, the initial position of the microsatellites
2. The probabilistic guidance algorithm [4, 5] was executed to determine the cube where each microsatellite was located
3. Construct the corresponding Voronoi cells \( \{ V_i \}_{i=1}^k \) for each microsatellite
4. Calculate the centroid positions \( \{ p^*_i \}_{i=1}^k \) of Voronoi cells
5. These centroids form a new set of points; move the generate point to the centroid
6. If the new generation point satisfies convergence, the algorithm terminates. Otherwise return to Step 3

4. Micro-Planning of Microsatellites

4.1. The Orbital Dynamics Model Considering \( J_2 \) Perturbation. The reference frame is the local-vertical-local-horizontal (LVLH) denoted by \( F_L \). The origin of LVLH coordinate system is the centroid of the central microsatellite, the \( x \)-axis points to the centroid of the central microsatellite from the center of the earth, the \( z \)-axis is perpendicular to the orbital plane and points to the direction of the orbital angular momentum, and the \( y \)-axis is perpendicular to the \( x \) and \( z \) axes and is determined by the right hand rule.

The relative dynamics equation of the microsatellite considering the \( J_2 \) perturbation is

\[
\begin{align*}
\dot{x} & = 3n^2 x + 2ny + f_x + u_x \\
\dot{y} & = -2nx + f_y + u_y \\
\dot{z} & = -n^2 z + f_z + u_z
\end{align*}
\]

where \( [x \ y \ z]^T \) is the position vector of the microsatellite in the LVLH coordinate system of the reference satellite, \( [u_x \ u_y \ u_z]^T \) is the control torque imposed on microsatellite, \( n = \sqrt{\mu/r^3} \) is the mean orbital motion of the microsatellite, \( r \) is geocentric distance, and \( \mu \) is the Earth’s gravitational constant.

The orbital elements are described as \( (a, e, i, \Omega, \omega, \nu) \), \( a \) is semi-major axis, \( e \) is eccentricity, \( i \) is orbit inclination, \( \Omega \) is the right ascension of ascending node, \( \omega \) is argument of perigee, and \( \nu \) is true anomaly. \( [f_x \ f_y \ f_z]^T \) is the perturbing acceleration caused by \( J_2 \):

\[
\begin{align*}
f_x & = \kappa \sin^2 i \sin 2u \\
f_y & = \kappa \sin 2i \sin u \\
f_z & = \kappa \left( 1 - \frac{3}{2} \sin^2 i + \frac{3}{2} \sin^2 i \cos 2u \right)
\end{align*}
\]

where the coefficient \( \kappa = -3/2\mu J_2 (R_e^2/r^4) \), \( R_e \) is the radius of the Earth, \( J_2 \) is the second zonal harmonic coefficient of the Earth, and \( u = \omega + v \) is argument of perigee.

The relative dynamics model of microsatellite swarm is expressed in state space:

\[
\begin{align*}
\dot{x}_j(t) & = A x_j(t) + B u_j(t) + d_j, j = 1, \ldots, N 
\end{align*}
\]

where \( N \) is the number of microsatellites, and the state of the \( j \)-th microsatellite at time \( t \) is \( x_j(t) = [x_j \ y_j \ z_j \ \dot{x}_j \ \dot{y}_j \ \dot{z}_j]^T \), and \( u_j = [u_{jx} \ u_{jy} \ u_{jz}]^T \).

Meanwhile,

\[
A = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & -2n & 0 & 0 & 0 \\
0 & 0 & -n^2 & 0 & 0 & 0
\end{bmatrix},
B = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
d_j = \begin{bmatrix}
f_{xj} \\
f_{yj} \\
f_{zj}
\end{bmatrix}
\]

To satisfy the conditions of solving convex optimization problem, the dynamics model was discretized and obtained from the continuous dynamics model by zero-order equal interval sampling:

\[
u_j(t) = u_j[k], t \in [t_k, t_{k+1}), j = 1, \ldots, N, k = 0, \ldots, T - 1,
\]

where \( t_k \) is the \( k \)-th interval sampling point, and \( T \) is the total interval sampling point.
\[ t_0 = 0, t_T = t_f, t_f = T \Delta t, \Delta t = t_{k+1} - t_k, k = 0, \ldots, T - 1. \]  

(19)

Then, the dynamics model of discretization is

\[ x_j[k + 1] = \hat{A}_j x_j[k] + \hat{B}_j u_j[k] + \hat{d}_j[k], k = 0, \ldots, T - 1, j = 1, \ldots, N, \]

(20)

where

\[ \hat{A}_j = e^{A \Delta t}, \hat{B}_j = \int_0^{\Delta t} e^{A(t-\tau)} B d\tau, \hat{d}_j = d_j \Delta t. \]

(21)

4.2. Convexification of Collision Avoidance Constraints. To avoid two members of microsatellite swarm moving to the same location at the same time, collision trajectory between members should be considered in the process of swarm reconfiguration. Then, the anti-collision avoidance constraint can be expressed as

\[ \|G x_j[k]\| \geq R_{col}, \]

(22)

where \( x_{ij} = x_j - x_i \), \( G = [I_{3 \times 3}, 0_{3 \times 3}], i \in N_{||j||}, N_{||j||} = \{i\} \|x_j - x_i\|_2 \leq R_{com} \), and \( R_{com} \) is the maximum communication distance.

Equation (21) is linearized at the discrete points corresponding to the two reference trajectories of microsatellite \( i \) and \( j \), and the affine function about the state of the two adjacent microsatellites can be obtained, so as to realize the convexity of collision avoidance constraints between microsatellites:

\[ (x_j[k] - \bar{x}_j[k])^T G^T G (x_j[k] - \bar{x}_j[k]) \geq R_{col} ||G(x_j[k] - \bar{x}_j[k])||_2, \]

\[ k = 0, \ldots, T, i > j, j = 1, \ldots, N - 1 \]

(23)

where \( \bar{x}_j \) is the nominal trajectory of each trajectory point. The accurate collision avoidance constraint requires only that the distance between two adjacent microsatellites is no less than the safe distance. However, after convexization, as shown in Figure 2, the constraint becomes that the two adjacent microsatellites cannot simultaneously be located in the belt region with width \( R_{col} \), which is perpendicular to the connection line of the discrete reference points of the two adjacent microsatellites, and the position of the space is not fixed, so it can move along the connection direction of the reference points.

4.3. MPC-Based Trajectory Optimization for Microsatellites. To implement real-time trajectory planning, model predictive control is introduced; the MPC implementation uses a receding horizon to update the optimal trajectories based on the current state information. Assuming that the prediction horizon and the control horizon are both \( p \), according to the principle of model predictive control, it can be obtained

\[ X_k = M \hat{x}_k + C U_k + N D, \]

(24)

where \( \hat{x}_k \) is initial state, and

\[ \begin{align*}
X_k &= \begin{bmatrix}
  x_k \\
x_{k+1} \\
\vdots \\
x_{k+p}
\end{bmatrix}, & U_k &= \begin{bmatrix}
  u_k \\
  u_{k+1} \\
\vdots \\
  u_{k+p-1}
\end{bmatrix}, & M_k &= \begin{bmatrix}
  I & \hat{A} & \hat{A}^2 & \cdots & \hat{A}^p
\end{bmatrix}, & D &= \begin{bmatrix}
  \hat{d} \\
\vdots \\
\hat{d}
\end{bmatrix},
\end{align*} \]

(25)

\[ C = \begin{bmatrix}
  0 & 0 & 0 & \cdots & 0 \\
\hat{B} & 0 & 0 & \cdots & 0 \\
\hat{A} \hat{B} & \hat{B} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
\hat{A}^{p-1} \hat{B} & \hat{A}^{p-2} \hat{B} & \cdots & \hat{A} \hat{B} & \hat{B}
\end{bmatrix}, \]

(26)

\[ N = \begin{bmatrix}
  0 & 0 & 0 & \cdots & 0 \\
  \hat{A} & 0 & 0 & \cdots & 0 \\
\hat{A}^2 & \hat{A} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
\hat{A}^{p-1} & \hat{A}^{p-2} & \cdots & \hat{A} & 0
\end{bmatrix}, \]

(27)

The cost function is defined as

\[ J_k = \sum_{i=0}^{p-1} \left( x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i} \right) + x_{k+p}^T F x_{k+p}, \]

(27)

where \( Q, R \), and \( F \) are the weight matrix, \( x_{k+i} \) is the predicted state, \( x_{k+p} \) is the predicted terminal state, \( Q \) and \( R \) are usually the diagonal matrix, and \( F \) is the terminal cost weight matrix obtained by solving the algebraic Riccati equation.
Substituting Equation (24) into Equation (27) yields

$$J_k = U_k^T H U_k + 2x_k^T G U_k + 2U_k^T V D + x_k^T S x_k + 2x_k^T W D + D^T K D,$$

(28)

where

$$H = C^T Q C + \tilde{R}, G = C^T Q M,$$

$$V = N^T Q C, S = M^T Q M,$$

$$W = N^T Q M, K = N^T Q N,$$

$$Q = \begin{bmatrix} Q & 0 & \cdots & 0 & 0 \\ 0 & Q & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Q & 0 \\ 0 & 0 & \cdots & 0 & F \end{bmatrix},$$

$$R = \begin{bmatrix} R & 0 & \cdots & 0 & 0 \\ 0 & R & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & R & 0 \end{bmatrix}.$$

In the whole reconfiguration stage, it is assumed that microsatellite members of swarm have limited communication capacity, so the members can only communicate with their neighboring microsatellites, so each member microsatellite needs to consider anti-collision constraints. So the trajectory planning problem of microsatellite swarm considering collision constraints can be described as

$$\min_{U_k} \{ U_k^T H U_k + 2x_k^T G U_k + 2U_k^T V D + x_k^T S x_k + 2x_k^T W D + D^T K D \},$$

subject to

$$x_{i,k+1} = A_i x_{i,k} + B_i u_{i,k} + d_i, k = k_0, \ldots, T - 1, j = 1, \ldots, N,$$

$$G_i^T G_i (x_{j,k} - \bar{x}_{j,k}) \geq R_{col} \left\| G_i (x_{j,k} - \bar{x}_{j,k}) \right\|_2,$$

$$k = k_0, \ldots, T, i > j, j = 1, \ldots, N - 1,$$

$$\left\| u_{i,j,k} \right\|_{\infty} \leq u_{\max}, k = k_0, \ldots, T - 1, j = 1, \ldots, N.$$  

(31)

5. Results

In this section, numerical simulation is carried out to verify the proposed macro-micro trajectory planning method of microsatellite swarm. In this paper, a virtual central microsatellite is given, and a large-scale (300) microsatellite swarm is designed with an omnidirectional flight configuration. The initial orbital element of the central satellite is (7106.14 km, 0.05°, 98.3°, 270°, 0°). The earth’s gravitational constant is 3.986 × 10^5 km^3 s^-2; the radius of the earth is 6378.140 km. In LVLH coordinates, the position vectors of 300 microsatellites relative to the virtual center microsatellite are randomly generated, the azimuth angle is randomly selected, and the initial distance is randomly selected within 100 km.

In the micro-planning, we do not have a reference trajectory only a reference final position. Hence, a terminal
constraint is added to the MPC. The state error cost is kept low because we do not have a state trajectory to follow only the final destination. Meanwhile, the control cost is kept high to emphasize the optimization of fuel consumption to be minimum to the reference which is taken to be zero. The sampling time is 60s, the prediction horizon is 50. The \( Q \) is \( 1 \times 10^{-11} \cdot I_{6\times6} \), \( R \) is \( 1 \cdot I_{3\times3} \), and \( F \) is \( 1 \times 10^{-7} \cdot I_{3\times3} \).

The overall configuration transformation of microsatellite swarm is shown in Figures 3 and 4. The initial configuration generated is shown in Figures 3(a) and 3(d); the desired configuration design is of type “0”. Here, the numbers of microsatellites in each sub-swarm have been specifically chosen so that the final configuration is uniform over the shape of the “0”. With the increase of iteration times, microsatellite swarm gradually forms the desired configuration under the action of macro and micro trajectory planning. It can be seen that the desired configuration can be formed quickly. After the 10 iterations, the desired configuration is basically formed and maintained in a stable state; as shown in Figure 4, there is no significant change in distance scale. In the meanwhile, the total variation in Figure 5 indicates that convergence is achieved at roughly \( t = 10 \) steps.

In this paper, we use the CVT algorithm to divide regions, and so as to determine the position of the
Figure 6: CVT of microsatellites in a cube during the transfer of microsatellite swarm.

Figure 7: Trajectory tracking from the initial position to the target position at a certain moment.
microsatellites to be transferred at the next moment. As shown in Figure 6, we selected one of the cubes in the transfer process and perform CVT on it to determine the transfer position of the microsatellite. After 50 iterations, a stable configuration is obtained, and the position where the microsatellite moves at the next moment is determined.

Due to the large scale of the microsatellite swarm, the process of achieving the final configuration requires many transitions. To verify the proposed trajectory optimization based on model predictive control, one of the microsatellites is selected from the initial point to the next desired target point at a certain moment, as shown in Figure 7. The individual microsatellites can reach the desired point well. After the desired point is reached, the next iteration will be carried out, and due to the influence of orbital dynamics, the microsatellite may not remain the target point without control constraints. Therefore, the trajectory curve after \( T = 6000 \) s of the simulation in Figure 7 is not a reference. To make the mission of microsatellite swarm more practical, we used MPC in micro-planning to improve the performance of microsatellite swarm in terms of fuel consumption and resource utilization.

6. Conclusion

This paper developed a synthesis trajectory planning method for microsatellite swarm, which include macro-planning and micro-planning. The former used Centroid Voronoi tessellation, while the latter adopted MPC to generate the optimal trajectories. The proposed method can not only realize collision avoidance of microsatellite swarm maneuvering at the macro-level, but also provide optimal trajectories for each microsatellite of swarm individuals at the micro-level. Macro-planning provides guidance for micro-planning, and micro-planning implements the commands of macro-planning; the proposed method accords well with engineering practice. Future work will focus on the extending the presented scheme to achieve a faster, more fuel-efficient trajectory optimization method. Moreover, the validation experiments will be carried out.

Data Availability

The simulation date and the program files in this paper cannot be shared with others as they are our basis for next research.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Authors’ Contributions

Xiwei Wu contributed to the writing of the manuscript and research design. Bing Xiao participated in the research design and review. Cihang Wu conducted simulation experiment. Yiming Guo and Cihang Wu performed statistical analysis.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61873207), the National Science and Technology Major Project, China (No. J2019-1-0021-0020), the Natural Science Basic Research Program of Shaanxi, China (No. 2019JQ-344), the Science and Technology Program of Xi’an City, China (No. 2019218314GXR019CG020-GXYD19.3), and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University, China (No. CX2021084).

References

[1] F. Y. Hadaegh, S. J. Chung, and H. M. Manohara, “On development of 100-gram-class spacecraft for swarm applications,” IEEE Systems Journal, vol. 10, no. 2, pp. 673–684, 2016.
[2] C. Sinigaglia, S. Bandyopadhyay, M. Quadrelli, and F. Braghin, “Optimal-transport-based control of particle swarms for orbiting rainbows concept,” Journal of Guidance, Control, and Dynamics, vol. 44, no. 11, pp. 2108–2117, 2021.
[3] S. Bandyopadhyay and M. Quadrelli, “Optimal Transport Based Control of Granular Imaging System in Space,” in 2017 International Workshop on Satellite Constellations and Formation Flying (IWSCFF), Boulder, Colorado, USA, June 2017.
[4] B. Açıkmese and D. S. Bayard, “A markov chain approach to probabilistic swarm guidance,” in 2012 American Control Conference (ACC), pp. 6300–6307, Montreal, QC, Canada, June 2012.
[5] B. Açıkmese and D. S. Bayard, “Probabilistic swarm guidance for collaborative autonomous agents,” in 2014 American Control Conference, pp. 477–482, Portland, OR, USA, June 2014.
[6] B. Açıkmese and D. S. Bayard, “Markov chain approach to probabilistic guidance for swarms of autonomous agents,” Asian Journal of Control, vol. 17, no. 4, pp. 1105–1124, 2015.
[7] N. Demir, B. Açıkmese, and M. W. Harris, “Convex optimization formulation of density upper bound constraints in markov chain synthesis,” in 2014 American control conference, pp. 483–488, Portland, OR, USA, June 2014.
[8] N. Demir, B. Açıkmese, and C. Pehlivantürk, “Density control for decentralized autonomous agents with conflict avoidance,” IFAC Proceedings Volumes, vol. 47, no. 3, pp. 11715–11721, 2014.
[9] N. Demir and B. Açıkmese, “Probabilistic density control for swarm of decentralized on-off agents with safety constraints,” in 2015 American Control Conference (ACC), pp. 5238–5244, Chicago, IL, USA, July 2015.
[10] N. Demir, U. Eren, and B. Açıkmese, “Decentralized probabilistic density control of autonomous swarms with safety constraints,” Autonomous Robots, vol. 39, no. 4, pp. 537–554, 2015.
[11] N. Demirer, “Density control of multi-agent systems with safety constraints: a Markov chain approach,” Ph.D. dissertation, 2017.
[12] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, “Inhomogeneous Markov Chain Approach to Probabilistic Swarm Guidance Algorithm,” in 5th Int. Conf. Spacecraft Formation Flying Missions and Technologies, Munich, Germany, 2013.
[13] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, “Probabilistic swarm guidance using inhomogeneous markov chains,”
IEEE Transactions on Control of Network Systems, 2014, http://arxiv.org/abs/1403.4134.

[14] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, “Feedback-based inhomogeneous markov chain approach to probabilistic swarm guidance,” in 2015 International Workshop on Satellite Constellations and Formation Flying (IWSCFF), Delft, Netherlands, June 2015.

[15] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, “A probabilistic Eulerian approach for motion planning of a large-scale swarm of robots,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3822–3829, Daegjeon, October 2016.

[16] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, “Probabilistic and distributed control of a large-scale swarm of autonomous agents,” IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1103–1123, 2017.

[17] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, “Probabilistic swarm guidance using optimal transport,” in 2014 IEEE Conference on Control Applications (CCA), pp. 498–505, Juan Les Antibes, France, October 2014.

[18] S. Bandyopadhyay, J.-P. de la Croix, I. Nesnas, D. Bayard, S.-J. Chung, and F. Hadaegh, “Probabilistic guidance of a swarm deployed from the back shell of a mars spacecraft,” in 2019 International Astronautical Congress (IAC), Washington, D.C., USA, October 2019.

[19] U. Eren, A. Prach, B. B. Kocer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, “Model predictive control in aerospace systems: current state and opportunities,” Journal of Guidance, Control, and Dynamics, vol. 40, no. 7, pp. 1541–1566, 2017.

[20] J. Carson and A. Acikmese, “A model-predictive control technique with guaranteed resolvability and required thruster silent times for small-body proximity operations,” in AIAA guidance, Navigation, and Control Conference and Exhibit, p. 6780, Keystone, Colorado, August 2006.

[21] S. Di Cairano, H. Park, and I. Kolmanovsky, “Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering,” International Journal of Robust and Nonlinear Control, vol. 22, no. 12, pp. 1398–1427, 2012.

[22] T. Pippia, V. Preda, S. Bennani, and T. Keviczky, “Reconfiguration of a satellite constellation in circular formation orbit with decentralized model predictive control,” 2022, https://arxiv.org/abs/2201.10399.

[23] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, “Model predictive control of swarms of spacecraft using sequential convex programming,” Journal of Guidance, Control, and Dynamics, vol. 37, no. 6, pp. 1725–1740, 2014.

[24] R. Foust, S.-J. Chung, and F. Y. Hadaegh, “Optimal guidance and control with nonlinear dynamics using sequential convex programming,” Journal of Guidance, Control, and Dynamics, vol. 43, no. 4, pp. 633–644, 2020.

[25] D. Morgan, G. P. Subramanian, S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, “Probabilistic guidance of distributed systems using sequential convex programming,” in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3850–3857, Chicago, IL, USA, September 2014.

[26] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” IEEE Transactions on Robotics and Automation, vol. 20, no. 2, pp. 243–255, 2004.

[27] T. Chevet, C. S. Maniu, C. Vlad, and Y. Zhang, “Voronoi-based uavs formation deployment and reconfiguration using mpc techniques,” in 2018 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 9–14, Dallas, Texas, USA, 2018.

[28] A. Carron and M. N. Zeilinger, “Model predictive coverage control,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 6107–6112, 2020.