Optimal Spacecraft Formation Reconfiguration with Collision Avoidance Using Particle Swarm Optimization

Haibin Huang[1], Yufei Zhuang[1], Guangfu Ma[2], Yueyong Lv[2]

[1]Harbin Institute of Technology at Weihai, Weihai 264209, People's Republic of China

[2]Harbin Institute of Technology, Harbin 150001, People's Republic of China
e-mail: hhb833@gmail.com

crossref http://dx.doi.org/10.5755/j01.itc.41.2.854

Abstract. This paper presents an energy-optimal trajectory planning method for spacecraft formation reconfiguration in deep space environment using continuous low-thrust propulsion system. First, we employ the Legendre pseudospectral method (LPM) to transform the optimal reconfiguration problem to a parameter optimization nonlinear programming (NLP) problem. Then, to avoid the computational complexity for calculating the gradient information caused by traditional optimization methods, we use particle swarm optimization (PSO) algorithm to solve the NLP problem. Meanwhile, in order to avoid the collision between any pair of Legendre-Gauss-Lobatto (LGL) points, we insert some test points in the region where collision may happen most likely. What's more, the collision avoidance constraints are also checked at these test points. Finally, numerical simulation shows that the energy-optimal trajectories for spacecraft reconfiguration could be generated by the method we proposed in a relative short time, so that it could be adopted on-board for practical spacecraft formation problems.

Keywords: formation reconfiguration; path planning; collision avoidance; Legendre pseudo spectral method; particle swarm optimization.

1. Introduction

The problem of spacecraft formation has been extensively addressed recently because of the potential benefits of formation flying missions. One of these benefits lies in that the formation could be re-assigned to establish new science configurations that we need. The purpose of formation reconfiguration is to plan a set of optimal translational trajectories, along which each spacecraft of the formation is able to transfer from its current states to the desired final states, respectively, with a performance index (such as fuel, energy, time, etc.) in a given time interval [1]. Additionally, the problem of collision avoidance and control input limits should also be considered during the optimization.

The literature on formation reconfiguration can be categorized as deep space missions (the gravity free environment) and planetary orbital environment (POE) missions [2]. In deep space missions, the spacecraft dynamics can be reduced to double integrator form, and varieties of formation reconfiguration algorithms have been proposed in the literature. Richards et al. [3] proposed a Mixed Integer Linear Programming (MILP) method, which could find a global optimized solution. However, the computation time would increase dramatically with the increase of the number of spacecraft or computation steps. Additionally, it also needs to simplify the constraints formulation to a linear form, which makes the collision avoidance constraint conservative. Singh and Hadaegh [4] used polynomials of a variable order in time to parameterize the trajectories, but the algorithm is too complex. Cetin et al. [5] combined these two methods. The trajectories were first discretized in time using a cubic spline and then a feasible MILP method was used to calculate the variables at discretized points. These methods also take a long time to solve the problem when the number of computation steps increases and could not get a high accuracy neither. Many other approaches have also been used in formation reconfiguration, such as the RRT-based method [6] and the multiple-shooting method [7].

The pseudospectral method is a newly developed class of methods for solving optimal control problems. In the pseudospectral method, the state and control vectors are discretized at specified time points using a structure of global orthogonal polynomials. This makes the optimal control problem easy to solve with high accuracy. This method has been used in some nonlinear spacecraft trajectory optimization problems. Huntington [8] used Gauss pseudospectral method for tetrahedral formation reconfiguration, but collision avoidance was not considered. Wu et al. [9] used LPM to design fuel-optimal trajectories for spacecraft reconfiguration in near-earth orbit with an exact nonlinear relative spacecraft dynamic model.

Autonomous formation flying is important for deep space missions, so the reconfiguration algorithm for deep space missions should be simple enough to run on-board and plan the trajectories fast even real-time. However, the collision avoidance constraints usually result in a non-convex feasible solution space. The reconfiguration problem with collision avoidance constraints is NP-complete [1] which makes the problem hard to solve. These problems make the aforementioned methods suffer from an accelerated increase in computational complexity when the number of spacecraft or the collocation points increases.

In this paper we present a novel method for trajectory planning of reconfiguration maneuvers of multi-spacecraft formation in deep space environment with continuous low-thrust control input. The basic problem discussed here is to find energy-optimal trajectories for the formation spacecraft in a relative short time. The spacecraft is modeled as points of constant mass. Normally, the maneuver time is short and the propulsion systems used for maneuver are quite efficient, so the mass of each spacecraft is assumed to be constant during the whole reconfiguration.

2. Problem formulation

2.1. Statement of the problem

Consider the formation spacecraft in deep space earth-trailing formation flying, i.e. they are on an earth-trailing heliocentric orbit. When using linearized Hill equations to describe the motion of the formation spacecraft, it can be shown that the differential orbit force between two spacecraft is of the order of 10^{-23} N. Because the reconfiguration usually occurs in a relatively short time scale, ignoring the orbital forces between spacecraft in this work is well justified [10]. We assume that a total number of M spacecraft take synchronous maneuvers in the same time interval $[0,T]$. The system dynamics in deep space can be stated as follows [11]:

$$\begin{aligned}&\dot{\boldsymbol{X}}_l(t)=\boldsymbol{A}\boldsymbol{X}_l(t)+\boldsymbol{B}\boldsymbol{U}_l(t),\\&l=1,2,\cdots,M,\ \ t\in[0,T]\end{aligned}\quad(1)$$

where

$$\boldsymbol{X}_l=[x_l,y_l,z_l,\dot{x}_l,\dot{y}_l,\dot{z}_l]^T,\ \ \boldsymbol{U}_l=[u_{x_l},u_{y_l},u_{z_l}]^T$$

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{0}_{3\times3},\boldsymbol{I}_{3\times3}\\\boldsymbol{0}_{3\times3},\boldsymbol{0}_{3\times3}\end{bmatrix},\ \ \boldsymbol{B}=\frac{1}{m_l}\begin{bmatrix}\boldsymbol{0}_{3\times3}\\\boldsymbol{I}_{3\times3}\end{bmatrix}$$

$\boldsymbol{X}_l(t)$ and $\boldsymbol{U}_l(t)$ are the state and control vectors of the lth spacecraft at time t, respectively, m_l is the mass of the lth spacecraft.

The control inputs are assumed to be continuous low-thrust forces confined to lie within specified limits

$$-\boldsymbol{U}_{\max}\le\boldsymbol{U}_l(t)\le\boldsymbol{U}_{\max},\ \ l=1,2,\cdots,M\ .\quad(2)$$

The states at initial point and final point are constrained with the following conditions

$$\begin{aligned}&\boldsymbol{X}_l(0)=\boldsymbol{X}_{l0}\\&\boldsymbol{X}_l(T)=\boldsymbol{X}_{lT}\end{aligned}\quad(3)$$

where $\boldsymbol{X}_{l0}$ and $\boldsymbol{X}_{lT}$ are the initial and final state vectors of the lth spacecraft.

Since the maneuvers time for all spacecraft is the same, the objective function for the reconfiguration problem is to find $\boldsymbol{U}_l(t)$, $t\in[0,T]$, $l=1,2,\cdots,M$, so that the energy consumption

$$J=\frac{1}{2}\sum_{l=1}^{M}\int_0^T\boldsymbol{U}_l^T\boldsymbol{U}_l(t)\mathrm{d}t\quad(4)$$

is minimized.

2.2. Collision avoidance

It is obvious that, in order to avoid collisions, each spacecraft should be at least a specified distance away from others at any time step. Here each spacecraft is assumed to be a sphere with a point mass. Collision avoidance constraints can be stated as forbidden spheres associated with the spacecraft as follows [2]

$$\begin{aligned}&\left\|\boldsymbol{r}_l(t)-\boldsymbol{r}_m(t)\right\|^2\ge d_{\text{safe}}^{\ 2},\\&l,m=1,2,\cdots,M,\ l\ne m\end{aligned}\quad(5)$$

where $\boldsymbol{r}_l(t)$ is the radius vector of the lth spacecraft at time t, and d_{safe} is the minimum safety distance between the centers of any two spacecraft. These constraints change the problem into a non-convex problem, which makes the formation reconfiguration problem difficult to solve.

3. Problem discretization using pseudospectral method

3.1. Legendre pseudospectral method

Let $L_N(t)$ denote the Legendre polynomial of order N, and $\dot{L}_N(t)$ be the first-order derivative of it. Let t_h, $h=0,1,2,\cdots N$ be the zeros of $(t^2-1)\dot{L}_N(t)$, with $t_0=-1$, $t_N=1$. These points are called LGL points, which serve as collocation points of the system. Then we select the Nth order Lagrange interpolating polynomials [12]

$$\begin{aligned}&\phi_h(t)=\frac{1}{N(N+1)L_N(t_h)}\cdot\frac{(t^2-1)\dot{L}_N(t)}{t-t_h},\\&h=0,1,...,N\end{aligned}\quad(6)$$

where $\phi_h(t)$ satisfies the relationship $\phi_h(t_j)=\delta_{hj}$.

For a given continuous function $F(t)$ defined on $[-1,1]$, the Nth degree interpolation polynomial is

$$F^N(t):=\sum_{h=0}^{N}F(t_h)\phi_h(t)\,. \tag{7}$$

The integration of $F^N(t)$ is

$$\int_{-1}^{1}F^N(t)\mathrm{d}t:=\sum_{h=0}^{N}F(t_h)w_h\,, \tag{8}$$

where

$$w_h=\frac{2}{N(N+1)}\cdot\frac{1}{[L_N(t_h)]^2}\,. \tag{9}$$

The derivative of $F^N(t)$ at the hth LGL point is

$$\dot{F}^N(t_h)=\sum_{j=0}^{N}D_{hj}F(t_j) \tag{10}$$

where $\boldsymbol{D}:=(D_{hj})$ is an $(N+1)\times(N+1)$ matrix, given by

$$\boldsymbol{D}=(D_{hj}):=\begin{cases}\dfrac{L_N(t_h)}{L_N(t_j)}\cdot\dfrac{1}{t_h-t_j} & h\neq j\\ -\dfrac{N(N+1)}{4} & h=j=0\\ \dfrac{N(N+1)}{4} & h=j=N\\ 0 & \text{otherwise}\,.\end{cases} \tag{11}$$

3.2. Discretization for Reconfiguration Problem

As LGL points lie in $[-1,1]$, the optimal problem should be first restated by the linear transformation of the independent variable τ [13]:

$$\tau=\frac{2t}{T-0}-\frac{T+0}{T-0}\,. \tag{12}$$

Then we extend LPM to multi-spacecraft case. The state and control vectors can be approximated using ϕ_h, $h=0,1,\cdots,N$

$$\boldsymbol{X}_l(\tau)\approx\boldsymbol{X}_l^{\,N}(\tau)=\sum_{h=0}^{N}\boldsymbol{X}_{l,h}\phi_h(\tau),\quad l=1,2,\cdots,M \tag{13}$$

$$\boldsymbol{U}_l(\tau)\approx\boldsymbol{U}_l^{\,N}(\tau)=\sum_{h=0}^{N}\boldsymbol{U}_{l,h}\phi_h(\tau),\quad l=1,2,\cdots,M \tag{14}$$

and

$$\dot{\boldsymbol{X}}_{l,h}(\tau)\approx\sum_{j=0}^{N}D_{hj}\boldsymbol{X}_{l,j},\quad l=1,2,\cdots,M,\ h=0,1,\cdots,N. \tag{15}$$

The integration term in J defined in the maneuvers time interval $[0,T]$ can also be approximated as

$$\int_0^T\boldsymbol{U}_l^{\,T}\boldsymbol{U}_l\ \mathrm{d}t\approx\sum_{h=0}^{N}\boldsymbol{U}_{l,h}^{\,T}\boldsymbol{U}_{l,h}w_h,\quad l=1,2,\cdots,M. \tag{16}$$

Thus, the trajectory planning problem can be translated into a nonlinear programming problem with undetermined parameters $\boldsymbol{X}_{l,h}$ and $\boldsymbol{U}_{l,h}$, and minimizeing the objective function

$$J=\frac{T-0}{4}\sum_{l=1}^{M}\sum_{h=0}^{N}\boldsymbol{U}_{l,h}^{\,T}\boldsymbol{U}_{l,h}w_h \tag{17}$$

subject to

$$\sum_{j=0}^{N}D_{hj}\boldsymbol{X}_{l,j}-\frac{T-0}{2}\boldsymbol{f}(\boldsymbol{X}_{l,h},\boldsymbol{U}_{l,h},\tau_h)=0,\quad l=1,2,\cdots,M,\ h=0,1,\cdots,N \tag{18}$$

$$d_{\text{safe}}^2-\left\|\boldsymbol{R}_{l,h}-\boldsymbol{R}_{m,h}\right\|^2\le 0,\quad l,m=1,2,\cdots,M,\ l\neq m,\ h=0,1,\cdots,N \tag{19}$$

$$\begin{cases}\boldsymbol{X}_{l,0}=\boldsymbol{X}_{l0}\\ \boldsymbol{X}_{l,T}=\boldsymbol{X}_{lT}\end{cases},\quad l=1,2,\cdots,M \tag{20}$$

$$-\boldsymbol{U}_{l\max}\le\boldsymbol{U}_l\le\boldsymbol{U}_{l\max},\quad l=1,2,\cdots,M \tag{21}$$

where $\boldsymbol{R}_{l,h}$ is the radius vector of the lth spacecraft at the hth LGL point. The number of the constraints described by Equation (18) is $6\cdot N\cdot M$, and the number of the constraints described by Equation (19) is $N\times C_M^2$.

4. Optimization using Particle Swarm Optimization

4.1. Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic optimization method which was invented by Kennedy and Eberhart in 1995 [14]. It is an evolutionary algorithm that inspired by the social behavior of bird flocking or people grouping. In PSO, each possible solution is called a particle that is analogous to a bird in the bird flocking. The objective of the particles population (called swarm) is to find the global minimum of the fitness function (cost function). In each iteration, every particle updates by its own improving velocity which is derived from the personal best solution (known as 'pbest') and the global best position (known as 'gbest') discovered so far by the whole swarm. The basic PSO algorithm can be described as

$$v_{i,h}^{k+1}=\omega v_{i,h}^{k}+c_1{r_1}^k(p_{i,h}^k-x_{i,h}^k)+c_2{r_2}^k(p_{g,h}^k-x_{i,h}^k)\,, \tag{22}$$

where $v_{i,h}^k$ and $x_{i,h}^k$ are the hth dimension velocity and position of particle i in the kth iteration; $p_{i,h}^k$ and $p_{g,h}^k$ are the hth dimension pbest and gbest of particle i in the kth iteration; ω is a weighting factor known as inertia; c_1 is the cognitive weight and c_2 is the social weight; r_1^k and r_2^k are two random numbers in the range of $[0,1]$. The new position of a particle is then calculated using

$$\bar{\boldsymbol{X}}_i^{k+1} = \bar{\boldsymbol{X}}_i^k + \bar{\boldsymbol{V}}_i^{k+1}. \tag{23}$$

Here, $\bar{\boldsymbol{X}}_i^k$ and $\bar{\boldsymbol{V}}_i^k$ are the position and velocity vector of the ith particle during the kth iteration. We use "$\bar{\ }$" to distinguish the position vector of PSO from the state vector of the reconfiguration problem.

When updating, a high velocity will drive the particles out of bounds or divergence, so the velocity of particle needs to be constrained. Set $V_{\max,h}$ as the maximum velocity of the hth dimension, then the formulation of velocity updating can be

$$v_{i,h}^{k+1} = \begin{cases} v_{i,h}^{k+1}, & if\ \left|v_{i,h}^{k+1}\right| \le V_{\max,h} \\ V_{\max,h}, & if\ v_{i,h}^{k+1} > V_{\max,h} \\ -V_{\max,h}, & if\ v_{i,h}^{k+1} < -V_{\max,h}. \end{cases} \tag{24}$$

4.2. Optimization of Nonlinear Problem

In this section, we use PSO to solve the NLP discretized by LPM. When dealing with constraints, especially equality constraints, the PSO method needs to be modified. Several methods have been mentioned for this problem, such as eliminate the infeasible solutions method, penalty method, repair method, and so on. But for high dimensions constrained nonlinear optimization problems, it is almost impossible to find a feasible solution using these methods. Here, we make r_1^k and r_2^k the same for every dimension. Note that the number of dimensions for one particle is $M \times (N+1) \times 9$, the 'hth dimension' mentioned below contains $M \times 9$ dimensions in fact, because it contains M spacecraft and each of them has 3 position variables, 3 velocity variables and 3 control input variables. We denote the hth dimension just for convenience, i.e.:

$$\begin{aligned} \bar{\boldsymbol{X}}_{i,h}^0 = & [\boldsymbol{X}_{i,h,1}^0, \boldsymbol{U}_{i,h,1}^0, \boldsymbol{X}_{i,h,2}^0, \boldsymbol{U}_{i,h,2}^0, \\ & \cdots, \boldsymbol{X}_{i,h,M}^0, \boldsymbol{U}_{i,h,M}^0] \\ h = & 0,1,\cdots N \end{aligned} \tag{25}$$

Then using the method of linear particle swarm optimization (LPSO) [15], the following theorem is derived

Theorem 1: For each spacecraft, if all the position $\bar{\boldsymbol{X}}_i^0$ satisfy

$$\begin{aligned} &\sum_{j=0}^{N} D_{hj} \boldsymbol{X}_{i,j,l}^0 - \frac{T-0}{2} \boldsymbol{f}(\boldsymbol{X}_{i,h,l}^0, \boldsymbol{U}_{i,h,l}^0, \tau_h) = 0, \\ &h = 0,1,\cdots,N,\ \ l = 1,2,\cdots M \end{aligned} \tag{26}$$

and all the initial velocities $\bar{\boldsymbol{V}}_i^0$ satisfy

$$\bar{\boldsymbol{V}}_i^0 = \boldsymbol{0} \tag{27}$$

then for any iteration k, Equation (18) is satisfied. Here, both $\boldsymbol{X}_{i,h,l}^0$ and $\boldsymbol{U}_{i,h,l}^0$ are the variables to be determined, they all belong to $\bar{\boldsymbol{X}}_{i,h}^0$. We separate them for convenience. Accordingly, $\boldsymbol{V}_{i,h,l}^0$ and $\boldsymbol{V}_{ui,h,l}^0$ are the initial velocities of $\boldsymbol{X}_{i,h,l}^0$ and $\boldsymbol{U}_{i,h,l}^0$, and both of them are the components of $\bar{\boldsymbol{V}}_{i,h}^0$.

Proof: Since $\bar{\boldsymbol{P}}_i^0$ is the local best of particle i and $\bar{\boldsymbol{P}}_g^0$ is the global best of all the particles, the following equations can be derived

$$\begin{aligned} &\sum_{j=0}^{N} D_{hj} \boldsymbol{P}_{i,j,l}^0 - \frac{T}{2} \boldsymbol{A} \boldsymbol{P}_{i,h,l}^0 - \frac{T}{2} \boldsymbol{B} \boldsymbol{P}_{ui,h,l}^0 = 0, \\ &h = 0,1,\cdots,N,\ \ l = 1,2,\cdots,M \end{aligned} \tag{28}$$

$$\begin{aligned} &\sum_{j=0}^{N} D_{hj} \boldsymbol{P}_{g,j,l}^0 - \frac{T}{2} \boldsymbol{A} \boldsymbol{P}_{g,h,l}^0 - \frac{T}{2} \boldsymbol{B} \boldsymbol{P}_{ug,h,l}^0 = 0, \\ &h = 0,1,\cdots,N,\ \ l = 1,2,\cdots,M \end{aligned} \tag{29}$$

Using Equation (22) and Equation (27), we can derive

$$\begin{aligned} &\boldsymbol{V}_{i,h,l}^1 = c_1 r_1^0 (\boldsymbol{P}_{i,h,l}^0 - \boldsymbol{X}_{i,h,l}^0) + c_2 r_2^0 (\boldsymbol{P}_{g,h,l}^0 - \boldsymbol{X}_{i,h,l}^0), \\ &h = 0,1,\cdots,N,\ \ l = 1,2,\cdots,M \end{aligned} \tag{30}$$

$$\begin{aligned} &\boldsymbol{V}_{ui,h,l}^1 = c_1 r_1^0 (\boldsymbol{P}_{ui,h,l}^0 - \boldsymbol{U}_{i,h,l}^0) + c_2 r_2^0 (\boldsymbol{P}_{ug,h,l}^0 - \boldsymbol{U}_{i,h,l}^0), \\ &h = 0,1,\cdots,N,\ \ l = 1,2,\cdots,M \end{aligned} \tag{31}$$

Then, from Equations (28)-(31) we can derive

$$\begin{aligned} &\sum_{j=0}^{N} D_{hj} \boldsymbol{V}_{i,j,l}^1 - \frac{T}{2} \boldsymbol{A} \boldsymbol{V}_{i,h,l}^1 - \frac{T}{2} \boldsymbol{B} \boldsymbol{V}_{ui,h,l}^1 = 0, \\ &h = 0,1,\cdots,N,\ ,\ \ l = 1,2,\cdots,M. \end{aligned} \tag{32}$$

From Equation (23), we can get

$$\begin{aligned} \sum_{j=0}^{N} D_{hj} \boldsymbol{X}_{i,j,l}^1 &= \sum_{j=0}^{N} D_{hj} \boldsymbol{X}_{i,j,l}^0 + \sum_{j=0}^{N} D_{hj} \boldsymbol{V}_{i,j,l}^1 \\ &= \frac{T}{2} \boldsymbol{A} \boldsymbol{X}_{i,l}^1 + \frac{T}{2} \boldsymbol{B} \boldsymbol{U}_{i,l}^1 \\ l = 1,2,\cdots,M. & \end{aligned} \tag{33}$$

Then we can derive

$$\begin{aligned} \sum_{j=0}^{N} D_{hj} \boldsymbol{X}_{i,j,l}^{k+1} &= \sum_{j=0}^{N} D_{hj} \boldsymbol{X}_{i,j,l}^k + \sum_{j=0}^{N} D_{hj} \boldsymbol{V}_{i,j,l}^{k+1} \\ &= \frac{T}{2} \boldsymbol{A} \boldsymbol{X}_{i,l}^{k+1} + \frac{T}{2} \boldsymbol{B} \boldsymbol{U}_{i,l}^{k+1} \\ l = 1,2,\cdots,M. & \end{aligned} \tag{34}$$

That is

$$\sum_{j=0}^{N} D_{hj} \boldsymbol{X}_{i,j,l}^{k+1} - \frac{T-0}{2} \boldsymbol{f}(\boldsymbol{X}_{i,h,l}^{k+1}, \boldsymbol{U}_{i,h,l}^{k+1}, \tau_h) = 0, \quad h = 0,1,\cdots,N, \; l = 1,2,\cdots,M, \; k \geq 0. \tag{35}$$

The above equations show that all the particles will fly through the hyperplane defined by the set of feasible solutions.

5. Solution approach

5.1. Initialization

We employ n_p particles to solve this problem, the initial positions of these n_p particles should satisfy Equation (26), and should guarantee that the formation spacecraft would not collide with each other. The operation steps are outlined as follows:

1. Optimize the reconfiguration problem without considering the collision avoidance, i.e., optimize the problem with the objective function (17) subject to Equation (18), (20) and (21). The solution of this optimization is defined as $\bar{\boldsymbol{P}}_s$. It will be used when updating the velocity. Since the optimization of this problem is a simple convex optimization, this process could be worked out quickly.
2. For any two spacecraft l and m, find out the point where the distance between them is nearest, noted as $k_{l,m}$, $k_{l,m} \in [0,1,\cdots,N]$.
3. Initialize the n_p particles using the objective function

$$J = -1 \tag{36}$$

subject to Equation (19), (21) and (22) with random initial solution guesses. So we could get a serial of feasible solutions $\bar{\boldsymbol{X}}_i^0$ ($i = 1,2,\cdots,n_p$).

5.2. Iteration

Update the positions of all the particles with Equation (23). The formulation to update the velocity is modified using

$$\begin{aligned} v_{i,h}^{k+1} = \; & \omega v_{i,h}^k + c_1 r_1^k (p_{i,h}^k - x_{i,h}^k) + \\ & c_2 r_2^k (p_{g,h}^k - x_{i,h}^k) + c_3 r_3^k (p_{s,h}^k - x_{i,h}^k) \end{aligned} \tag{37}$$

where r_3^k is a random number in $[0,1]$, c_3 is an acceleration coefficient. According to the experiments we can find that the best feasible trajectories of the reconfiguration problem always be found near the optimal trajectory obtained without collision avoidance constraints. So $c_3 r_3^k (p_{s,d}^k - x_{i,d}^k)$ will drive the particles towards the optimal trajectory, which makes convergence faster than only using Equation (22). It can be proven as with Theorem 1 that Equation (18) is also satisfied in any iteration with Equation (37). However, since $\bar{\boldsymbol{P}}_s$ is not feasible, Equation (37) will drive the particles to infeasible region after some iterations. So we would eliminate $c_3 r_3^k (p_{s,d}^k - x_{i,d}^k)$ and use Equation (22) instead after 50 iterations.

When using pseudospectral method or other collocation method, the constraints are only satisfied at collocation points. So the solution may not be feasible between collocation points. More LGL points may solve this problem, but the computation time will also increase dramatically with the increasing points. To ensure that the spacecraft would not collide with each other between the LGL points, we insert some time points, called test points between $(k_{l,m} - 1)$ and $k_{l,m}$, and between $k_{l,m}$ and $(k_{l,m} + 1)$. The time at these test points should be calculated using Equation (13) before iteration. And then, in each iteration, abandon the solutions which could not avoid the collision at these test points and the LGL points. In this way, the final solution might be feasible for the entire reconfiguration. Here, we choose the quadrisection points as the test points. Note that we can choose more or less test points according to the actual situation. It have little influence on the computation time.

If the values of the fitness functions of all the swarms do not improve in the last n_{stall} generations or the generation maximum n_g is reached, stop the optimization. The optimal state and control input vectors of every spacecraft will be the last global best position of the swarm.

6. Result

In this paper, we used the NLP solver, known as KNITRO, to generate every particle's initial trajectories described by LPM. The software interfaced with Matlab, where the problem descriptions were performed. The problem was solved on a 2.1GHz personal computer with 2GB of RAM.

This example involves three spacecraft in three-dimensional space. It is assumed that they take synchronous maneuvers in 10 time units. The initial and final positions are

$$\begin{aligned} &\boldsymbol{r}_1(0) = [0\ 0\ 0]^T, \quad \boldsymbol{r}_1(T) = [15\ 15\ 15]^T, \\ &\boldsymbol{r}_2(0) = [10\ 0\ 0]^T, \quad \boldsymbol{r}_2(T) = [0\ 15\ 15]^T, \\ &\boldsymbol{r}_3(0) = [10\ 0\ 10]^T, \quad \boldsymbol{r}_3(T) = [0\ 15\ 0]^T \end{aligned} \tag{38}$$

The initial and final velocities are all zero. The mass of each spacecraft was assumed as 1 mass unit; the thrust limit was 1 unit; the safety distance was set to be 2 units; the number of LGL points was 10. For the PSO method, we took 10 particles for every spacecraft, and $V_{\max} = 2$, $n_{\text{stall}} = 50$, $n_g = 200$. The optimal trajectories of the three spacecraft are shown in Figure 1, where 'SC' means spacecraft. The distances between any two spacecraft in the given time units are shown in Figure 2, where the mark '∘' means the locations of the LGL points. The control inputs of the three spacecraft are shown in Figure 3, and the mark '∘' also means the value of control inputs on each LGL point. The total energy consumption was 11.1958 units optimized by 145 generations. The history of the global best during iteration is shown in Figure 4.

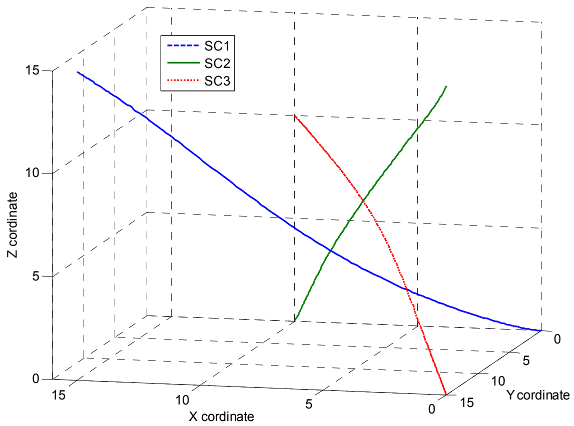

Figure 1. Optimal trajectories of spacecraft

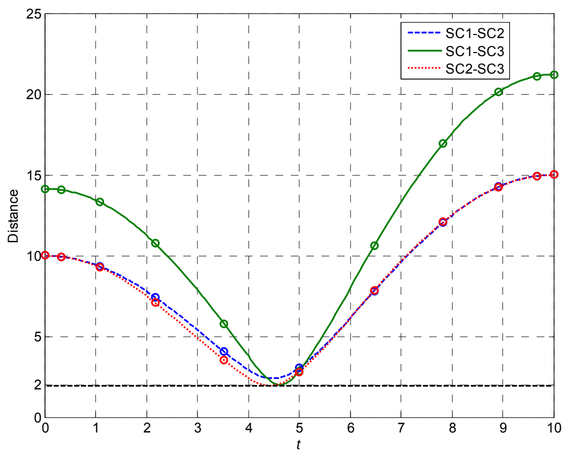

Figure 2. Distances between each pair of spacecraft

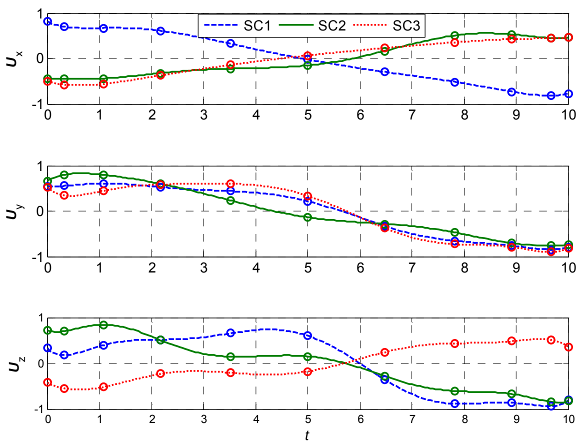

Figure 3. Control inputs of all spacecraft

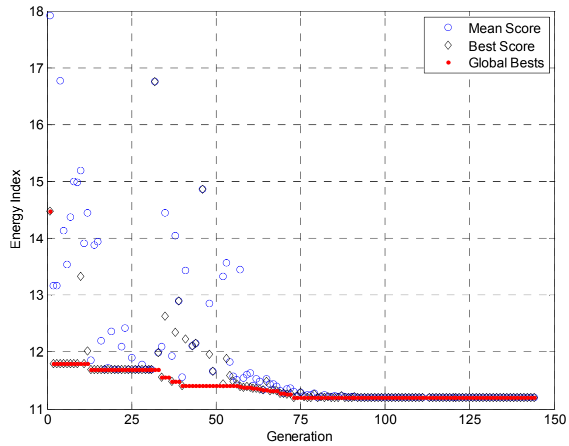

Figure 4. Changes of the global best of all spacecraft

From Figure 4 we can find that the process converges fast at beginning, and then evolves in a relative small region. After 50 iterations, all the particles converge to the global best value, which illustrates the good performance of our algorithm's convergence. Figure 5 shows the energy consumption result from 200 Monte Carlo simulations with random initial values.

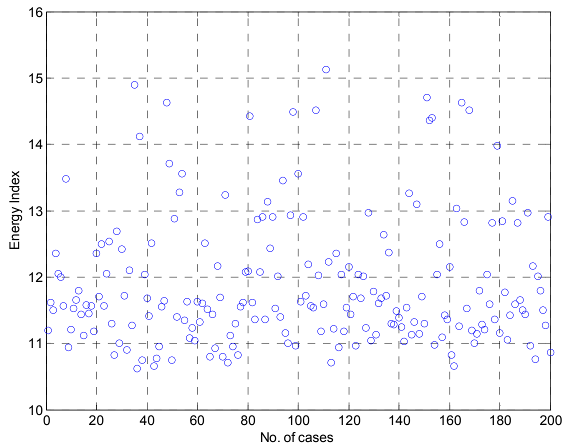

Figure 5. Evaluation through Monte Carlo simulations

From the results we can find that for a certain problem, this method could only find a near optimal solution, but not a certain optimal solution. However, this method could obtain a solution with collision avoidance in a short time which is shown in Table 1. Note that major time consumption is caused by initialization, about 18 seconds, and the iteration process only takes a very short time. The results also show that there are no great changes in the energy index, which indicates that our algorithm is robust.

Figure 6 plots the results solved only by LPM using 10 LGL points with 9.9000 units of energy consumption. The NLP solver is also KNITRO. From this figure we can find that the distances between the spacecraft are almost zero between the 4th and 5th LGL points, though they would not collide with each other at the LGL points. We also took 200 Monte

Carlo simulations with random initial values for the LPM. The comparison with PSO method is shown in Table 1. The nearest distance refers to the nearest distance between any two spacecraft. Note that the nearest distance and the computation time are the average values for 200 Monte Carlo simulations. Maximum and minimum time are the maximum and minimum computation time of 200 Monte Carlo simulations. From the results we can find that, the PSO method could avoid most collisions during the whole maneuvers. Even if some collisions might occur between LGL points, the minimum distance between any two spacecraft was only a little smaller than the safety distance. This situation could be acceptable because the safety distance is always conservative, moreover it also happens occasionally. We can also see that the LPM could not avoid the collision with 10 LGL points, and the computation time vary a lot with different initial values. Even using 30 LGL points, the nearest distance between two spacecraft was still 1.5 units. The computation time increased to about 17 minutes.

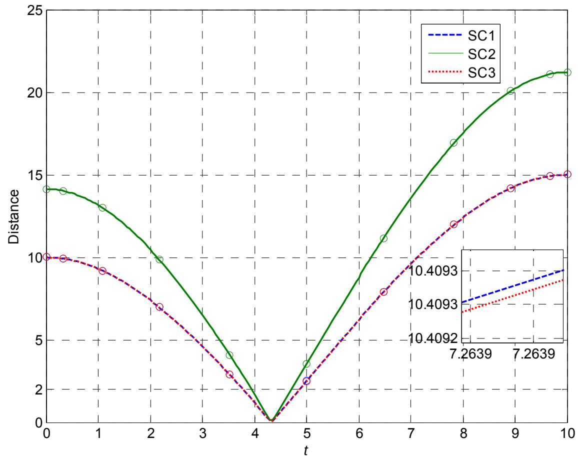

Figure 6. Evaluation through Monte Carlo simulations with only LPM

Table 1. Contrast between the PSO method and the LPM

Method	Nearest Distance	Computation Time/s	Maximum Time/s	Minimum Time/s
PSO	1.9138	19.9936	24.2913	16.7734
LPM	0.0862	17.8736	44.2292	5.5370

7. Conclusions

An efficient method for optimal reconfiguration of deep space spacecraft formation with collision avoidance is proposed in this paper. Competitive computational efficiency is obtained by combining Legendre pseudospectral method and particle swarm optimization algorithm. Compared to typical collocation methods, more potential collisions occurring between spacecraft can be avoided by using this algorithm, which considers the collision constraints between any pair of Legendre-Gauss-Lobatto points. Simulation results illustrate that this method could solve the reconfiguration problem quickly so that it could be used on-board as a general approach for spacecraft formation reconfiguration problems.

Acknowledgements

This work was supported in part by the National Science Foundation of China under the grant 61004072.

References

[1] **A. B. Acikmese, D. P. Schar, E.A. Murray, F. Y. Hadaegh.** A convex guidance algorithm for formation reconfiguration. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[2] **C. Sultan, S. Seereram, R. K. Mehra.** Deep space formation flying spacecraft path planning. The International *Journal of Robotics Research*, *Vol.* 26, *No.* 4, 2007, 405-430. http://dx.doi.org/10.1177/0278364907076709.

[3] **A. Richards, T. Schouwenaars, J. P. How, E. Feron.** Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance Control and Dynamics, Vol.* 25, *No.* 4, 2002, 755-764. http://dx.doi.org/10.2514/2.4943.

[4] **G. Singh, F. Y. Hadaegh.** Collision avoidance guidance for formation-flying applications. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2001.

[5] **B. Cetin, M. Bikdash, F. Y. Hadaegh.** Hybrid mixed-logical linear programming algorithm for collision-free optimal path planning. *IET Control Theory & Applications*, *Vol.* 1, *No.* 2, 2007, 522-531.

[6] **G. Aoude, J. How.** Two-stage path planning approach for designing multiple spacecraft reconfiguration maneuvers. *The 20th International Symposium on Space Flight Dynamics*, 2007.

[7] **M. Massari, F. B. Zazzera.** Optimization of Low-thrust Reconfiguration Maneuvers for Spacecraft Flying in Formation. *Journal of Guidance, Control, and Dynamics*, *Vol.* 32, *No.* 5, 2009, 1629-1638. http://dx.doi.org/10.2514/1.37335.

[8] **G. T. Huntington, A. V. Rao.** Optimal reconfiguration of spacecraft formations using the gauss pseudospectral method. *Journal of Guidance, Control, and Dynamics*, *Vol.* 31, *No.* 3, 2008, 689-698. http://dx.doi.org/10.2514/1.31083.

[9] **B. Wu, D. Wang, E. Poh, G. Xu.** Nonlinear Optimization of Low-Thrust Trajectory for Satellite Formation: Legendre Pseudospectral Approach. *Journal of Guidance, Control, and Dynamics*, *Vol.* 32, *No.* 4, 2009, 1371-1381. http://dx.doi.org/10.2514/1.37675.

[10] **Y. Kim, M. Mesbahi, F. Y. Hadaegh.** Dual-spacecraft formation flying in deep space: optimal collision-free reconfigurations. *Journal of Guidance, Control, and Dynamics*, *Vol.* 26, *No.* 2, 2003, 375-379. http://dx.doi.org/10.2514/2.5059.

[11] D. P. Scharf, F. Y. Hadaegh, B. H. Kang. On the validity of the double integrator approximation in deep space formation flying. *Proceedings of the international symposium Formation Flying Missions & Technologies, Toulouse, France,* 2002.

[12] G. Elnagar, M. A. Kazemi, M. Razzaghi. The pseudospectral legendre method for discretizing optimal control problems. *IEEE Transactions on Automatic Control*, *Vol.* 40, *No.* 10, 1995, 1793-1796. http://dx.doi.org/10.1109/9.467672.

[13] Q. Gong, F. Fahroo, I. M. Ross. Spectral algorithm for pseudospectral methods in optimal control. *Journal of Guidance, Control, and Dynamics, Vol.* 31, *No.* 3, 2008, 460-471. http://dx.doi.org/10.2514/1.32908.

[14] J. Kennedy, R. C. Eberhart. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ,* 1995, 1942-1948. http://dx.doi.org/10.1109/ICNN.1995.488968.

[15] U. Paquet, A. P. Engelbrecht. A new particle swarm optimizer for linearly constrained optimization. *Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, NJ*, 2003, 227-233.

Received November 2010.