

文章编号: 1001-0920(2011)11-1699-07

## 基于时间状态的敏捷自主在轨服务航天器 协同运动规划方法研究

王平, 郭继峰, 史晓宁, 崔乃刚

(哈尔滨工业大学 航天学院, 哈尔滨 150001)

**摘要:** 面对未来工业化、军事化太空对空间体系在轨服务的需求, 并兼顾敏捷自主在轨服务航天器自身和所在复杂空间环境等特点, 对多敏捷自主在轨服务航天器系统的协同运动规划问题进行研究. 利用以平动、转动和时间为状态的改进快速搜索随机树运动规划方法, 对服务系统中各航天器灵巧、精确的协同运动进行规划. 仿真算例检验了该方法的有效性, 结果表明它能够满足敏捷服务航天器系统在一定约束条件下按指定时间到达指定地点的协同运动要求, 并且面对此类高维运动规划问题具备一定的规划效率.

**关键词:** 敏捷航天器; 在轨服务; 协同运动规划; 随机树; 高维运动

中图分类号: V47

文献标识码: A

### Research on cooperative motion planning of agile and autonomous on-orbit servicing spacecraft considering time

WANG Ping, GUO Ji-feng, SHI Xiao-ning, CUI Nai-gang

(School of Astronautics, Harbin Institute of Technology, Harbin 150001, China. Correspondent: WANG Ping, E-mail: dandanping@sina.com)

**Abstract:** According to the requirements of on-orbit system servicing in the future industrialized and military space, the cooperative motion planning of multiple agile autonomous servicing spacecrafts system is studied with complicated space environment. The improved rapidly-exploring random tree method considering the state of translation, rotation and time, is used for cooperative motion planning of servicing spacecrafts' accurate motion. This planner satisfies the requirement of agile and autonomous spacecraft maneuvering with complex constraints. The simulation results show that this high-dimensional motion planner can obtain the feasible motion trajectories and the computational efficiency is high.

**Key words:** agile spacecraft; on-orbit servicing; cooperative motion planning; random tree; high-dimensional motion

## 1 引言

随着空间在轨服务技术<sup>[1]</sup>的不断发展, 空间服务任务<sup>[2]</sup>愈发复杂, 不仅需要灵活、敏捷的服务航天器, 而且对于规模化、大型化的空间服务对象而言, 还需要通过多服务航天器协同完成复杂的服务任务, 如空间装配、维修或维护<sup>[3]</sup>等. 在未来的工业化、军事化太空中, 体系在轨服务是服务大型服务对象的必由之路, 其中应用敏捷航天器和精确的协同服务是体系在轨服务的关键, 也是实现太空工业化的前提条件之一. 因此, 对于具有较高自主性的敏捷服务航天器协同运动规划方法的研究具有一定的理论和工程意义.

近年来, 人们对航天器运动规划和多航天器协同运动规划等方法进行了较为深入的研究, 如采用

人工势场<sup>[4]</sup>、混合整数线性规划<sup>[5]</sup>或快速搜索随机树(RRT)<sup>[6]</sup>等方法解决空间交会运动<sup>[7]</sup>、航天器编队队形重构运动<sup>[8-9]</sup>、多航天器近距离操作运动<sup>[10]</sup>等规划问题.

在以往的研究中, 对协同问题的考虑主要集中在几个方面, 如航天器之间的避碰问题、最短时间到达或相同时间到达等, 这样的“协同”在未来工业化太空中将无法满足不同规模在轨服务的需求. 在现代社会自动化工业生产中, 任何步骤的操作、行为都应具备准确的时间安排, 任何执行单元都必须严格按照时间安排操作, 这样才能满足大工业的要求. 所以在多服务航天器协同运动规划问题中, 也应将时间问题作为重点进行考虑, 即各航天器可在各自指定的时间到

收稿日期: 2010-04-06; 修回日期: 2010-06-12.

作者简介: 王平(1981-), 男, 博士生, 从事空间在轨服务航天器运动规划与控制的研究; 崔乃刚(1964-), 男, 教授, 博士生导师, 从事空间飞行器动力学、运动规划与控制等研究.

达指定地点,因此,将时间与平动、转动一起考虑在内的协同运动规划便成为了典型的高维运动规划问题.为实现在有限空间内各服务航天器的精确协同运动,敏捷的特点也是必不可少的.这里的敏捷主要表现在两个方面:一是通过快速反应得到运动的可行航迹,这要求在考虑应有的约束条件下,自主规划算法能够以较低的计算量和计算强度在较短的时间内得到规划结果;再者则是以较快的速度,准时完成既定机动任务.

基于随机采样原理的快速搜索随机树方法具有概率完备性、计算量小、计算难度低等特点,非常适合敏捷自主航天器.此外,它对解决高维状态规划问题具有较强的适应性<sup>[11]</sup>,这一能力是目前多数规划算法所不具备的,典型代表有对多关节机械臂的运动规划<sup>[12]</sup>,对平动加转动的六自由度运动规划<sup>[8-9]</sup>以及多智能体联合运动规划<sup>[13]</sup>等.由此可见,该规划方法非常适合本文高维运动状态(多航天器的平动、转动和时间)规划的实际情况.因此,本文将针对空间多种约束条件,利用RRT规划方法对敏捷自主在轨服务航天器协同运动规划问题进行研究.给定初末运动状态以及各航天器的服务时间安排,对航天器协同服务运动进行全局规划.同时,针对RRT规划方法易进入局部陷阱的缺点,利用随机采样原理进行改进.最终进行了数学和三维视景仿真,以检验算法的有效性.

## 2 约束条件

敏捷自主在轨服务航天器的协同运动规划问题主要考虑两种约束,即运动动力学约束<sup>[14-15]</sup>和几何约束<sup>[8]</sup>.

### 2.1 运动动力学约束

航天器运动动力学约束一般源于推进装置、执行机构或有效载荷等设备.推进装置提供的平动所需的加速度和速度,执行机构提供的最大姿态控制力矩以及有效载荷对运动速度的限制等因素,分别对速度、加速度、姿态角速度和角加速度提出了约束<sup>[16]</sup>要求.根据运动动力学方程,运动动力学约束如下:

$$\begin{cases} v \leq v_{\max}(\mathbf{r}, t), \\ \dot{v} \leq \dot{v}_{\max}(\mathbf{r}, t), \\ \omega \leq \omega_{\max}(\mathbf{r}, t), \\ \dot{\omega} \leq \dot{\omega}_{\max}(\mathbf{r}, t). \end{cases} \quad (1)$$

其中:  $v_{\max}$ ,  $\dot{v}_{\max}$ ,  $\omega_{\max}$  和  $\dot{\omega}_{\max}$  为速度、加速度、姿态角速度和角加速度的最大值约束,它们分别是平动向量  $\mathbf{r}=[x \ y \ z]^T$ , 姿态四元数向量  $\mathbf{q}=[q_1 \ q_2 \ q_3 \ q_4]^T$  和运行时间  $t$  的函数,但也可根据具体情况简化为恒定不变.

### 2.2 几何约束

几何约束主要指空间环境中固定或时变的实体

障碍约束或指向约束等.这里假设所有约束均为可知,且均为实体障碍约束,即

$$\begin{cases} \text{Obs}(\mathbf{r}_i) < 0, \\ \text{Obs}(\mathbf{r}_i, \mathbf{r}_j) < 0, \\ \text{Obs}(\mathbf{r}_{i,k}) < 0, \\ \|\mathbf{r}_i - \mathbf{r}_j\| > R_{\text{Safe}}. \end{cases} \quad (2)$$

其中:  $\text{Obs}()$  为约束判断函数,小于零时为满足约束条件;第1式表示航天器质心位置  $\mathbf{r}_i$  不在实体障碍内;第2式表示位置  $\mathbf{r}_i$  与  $\mathbf{r}_j$  之间的路径不在实体障碍内;第3式表示航天器第  $i$  个质心位置对应的第  $k$  个本体特征点  $\mathbf{r}_{i,k}$  不在实体障碍内;第4式表示第  $i$  个服务航天器与第  $j$  个服务航天器质心间的距离应大于安全距离  $R_{\text{Safe}}$ .

## 3 协同运动规划

利用RRT规划方法对敏捷自主在轨服务航天器协同运动规划问题进行研究,并对该规划方法的基本内容进行描述,具体包括:多服务航天器随机运动状态的确定和度量,局部规划器,规划问题的基本描述,必然碰撞区的概念以及规划方法的详细内容.

### 3.1 多服务航天器随机运动状态与度量

#### 3.1.1 多服务航天器运动状态

第  $i$  个服务航天器运动状态应包含描述三维平动的位置矢量  $\mathbf{r}_i$ , 描述转动的姿态四元数  $\mathbf{q}_i$  和时间状态  $t_i$ . 对时间的规划,速度因素是必不可少的,因此运动状态还应包含速度和角速度的信息.利用2个相邻状态确定2个状态间平动运动方向和姿态变化的欧拉轴,因此运动状态中速度和角速度信息确定为速度和姿态角速度的大小  $v_i$  和  $\omega_i$ . 由此,第  $i$  个服务航天器的运动状态可具体表述为

$$\mathbf{p}_i = [\mathbf{r}_i \ v_i \ \mathbf{q}_i \ \omega_i \ t_i]^T. \quad (3)$$

其中:  $v_i \in [0, v_{\max}]$ ,  $\omega_i \in [0, \omega_{\max}]$ ,  $t_i \in [0, t_{i,\max}]$ ,  $t_{i,\max}$  为第  $i$  个服务航天器的运动终止时刻(各航天器可根据任务需要设置不同的终止时刻).

对于全部航天器而言,运动状态为

$$\mathbf{p} = [\cdots \ \mathbf{p}_i \ \cdots]^T. \quad (4)$$

#### 3.1.2 随机运动状态的生成

所有随机运动状态在相应的状态空间内按均匀分布进行采样,其中生成均匀分布的随机四元数为<sup>[6]</sup>

$$\mathbf{q}_{\text{rand}} = [\sqrt{1-u_1} \sin 2\pi u_2, \sqrt{1-u_1} \cos 2\pi u_2, \sqrt{u_1} \sin 2\pi u_3, \sqrt{u_1} \cos 2\pi u_3]. \quad (5)$$

其中  $u_1, u_2, u_3 \in [0, 1]$  为均匀分布采样点.

#### 3.1.3 运动状态的度量

运动状态的度量主要用于描述运动状态空间中的距离,这里采用实数方程  $\rho(\mathbf{p}_i, \mathbf{p}_j)$  表述该运动状态

空间中某一航天器航迹 2 个节点之间的度量, 即

$$\begin{aligned} \rho(\mathbf{p}_i, \mathbf{p}_j) &= \|\mathbf{p}_i - \mathbf{p}_j\| = \\ &K_1(\|\mathbf{r}_i - \mathbf{r}_j\| + K_2|v_i - v_j|) + \\ &K_3[\rho(\mathbf{q}_i, \mathbf{q}_j) + K_4\sqrt{(\omega_i - \omega_j)^2 + K_5(t_i - t_j)^2}]. \end{aligned} \quad (6)$$

其中:  $K_1, K_2, K_3, K_4$  和  $K_5$  为权系数;  $\rho(\mathbf{q}_i, \mathbf{q}_j)$  为 2 姿态四元数间的度量, 即<sup>[6]</sup>

$$\rho(\mathbf{q}_i, \mathbf{q}_j) = \min\{\|\mathbf{q}_i - \mathbf{q}_j\|, \|\mathbf{q}_i + \mathbf{q}_j\|\}. \quad (7)$$

式 (7) 表示的 2 个姿态间度量是四元数  $\mathbf{q}_i$  分别与  $\mathbf{q}_j$  和  $-\mathbf{q}_j$  的 2 个距离中的最小值, 这是由于  $\mathbf{q}_j$  和  $-\mathbf{q}_j$  代表同一姿态, 但其旋转方式不同. 表示 2 个四元数之间的距离函数为

$$\rho_s(\mathbf{q}_i, \mathbf{q}_j) = \|\mathbf{q}_i - \mathbf{q}_j\| = \arccos(\mathbf{q}_i \cdot \mathbf{q}_j). \quad (8)$$

因此, 2 个姿态间的度量  $\rho(\mathbf{q}_i, \mathbf{q}_j)$  表示为

$$\rho(\mathbf{q}_i, \mathbf{q}_j) = \min\{\rho_s(\mathbf{q}_i, \mathbf{q}_j), \rho_s(\mathbf{q}_i, -\mathbf{q}_j)\}. \quad (9)$$

### 3.2 局部规划器

局部规划器主要用于确定协同体系中某一航天器航迹 2 个节点  $\mathbf{p}_i$  和  $\mathbf{p}_j$  间的运动. 根据规划算法的要求, 局部规划器并不用于解决全局规划问题, 而是对分段路径进行规划, 因此要求它所规划的路径相对短小, 规划方法相对简单, 这样才能满足整体算法快速性的需求<sup>[6,11]</sup>.

在规划过程中, 已知 2 个节点状态  $\mathbf{p}_i$  和  $\mathbf{p}_j$ , 根据规划算法对局部规划器的要求, 认为从  $\mathbf{p}_i$  出发至  $\mathbf{p}_j$  的平动速度方向与姿态运动旋转角速度矢量方向保持不变, 这里规划器的目的是求取从  $\mathbf{p}_i$  出发经过时间  $\Delta t$  的新节点  $\mathbf{p}_{i+1}$ , 要求 2 个节点的时间状态应满足  $t_i < t_j$ , 以保证时间不可逆的要求, 且  $t_j - t_i \geq \Delta t$ . 新节点  $\mathbf{p}_{i+1}$  表示为

$$\mathbf{p}_{i+1} = \begin{bmatrix} \mathbf{r}_{i+1} \\ v_{i+1} \\ \mathbf{q}_{i+1} \\ \omega_{i+1} \\ t_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_i + \left(v_i \Delta t + \frac{v_j - v_i}{t_j - t_i} \frac{\Delta t^2}{2}\right) \mathbf{e}_1 \\ v_i + \frac{v_j - v_i}{t_j - t_i} \Delta t \\ \mathbf{q}_i^{\oplus} \mathbf{q}_E \\ \omega_i + \frac{\omega_j - \omega_i}{t_j - t_i} \Delta t \\ t_i + \Delta t \end{bmatrix}. \quad (10)$$

其中:  $\mathbf{q}_i^{\oplus}$  为  $\mathbf{q}_i$  的四元数矩阵, 即

$$\mathbf{q}_i^{\oplus} = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix}; \quad (11)$$

$\mathbf{q}_E$  为四元数  $\mathbf{q}_i$  与  $\mathbf{q}_{i+1}$  之间的误差四元数, 即

$$\mathbf{q}_E = \begin{bmatrix} \sin\left(\omega_i \Delta t + \frac{\omega_j - \omega_i}{t_j - t_i} \frac{\Delta t^2}{2}\right) \mathbf{e}_2 \\ \cos\left(\omega_i \Delta t + \frac{\omega_j - \omega_i}{t_j - t_i} \frac{\Delta t^2}{2}\right) \end{bmatrix}. \quad (12)$$

$\mathbf{e}_1$  和  $\mathbf{e}_2$  分别为从节点  $\mathbf{p}_i$  到  $\mathbf{p}_j$  的速度和旋转角速度单位矢量方向. 利用位置矢量  $\mathbf{r}_i$  和  $\mathbf{r}_j$  求得  $\mathbf{e}_1$ , 即

$$\mathbf{e}_1 = \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|}; \quad (13)$$

利用四元数  $\mathbf{q}_i$  与  $\mathbf{q}_j$  之间的误差四元数  $\mathbf{q}_E$  对应的姿态矩阵  $\mathbf{A}(\mathbf{q}_E)$ , 通过下式求得  $\mathbf{e}_2$ :

$$\mathbf{e}_2 = \frac{1}{2 \sin \Phi} \begin{bmatrix} \mathbf{A}_{yz} - \mathbf{A}_{zy} \\ \mathbf{A}_{zx} - \mathbf{A}_{xz} \\ \mathbf{A}_{xy} - \mathbf{A}_{yx} \end{bmatrix}, \quad (14)$$

$\Phi$  为四元数  $\mathbf{q}_i$  与  $\mathbf{q}_j$  之间的欧拉转角.

$$\cos \Phi = \frac{1}{2}(\text{tr} \mathbf{A} - 1), \quad (15)$$

其中  $\text{tr} \mathbf{A} = \mathbf{A}_{xx} + \mathbf{A}_{yy} + \mathbf{A}_{zz}$  为姿态矩阵  $\mathbf{A}(\mathbf{q}_E)$  的迹.

式 (10) 中的  $(v_j - v_i)/(t_j - t_i)$  和  $(\omega_j - \omega_i)/(t_j - t_i)$  分别为在每一时间增量  $\Delta t$  中的加速度和角加速度, 它们均须满足运动动力学约束 (1) 的要求.

### 3.3 规划问题基本描述

此类规划问题可具体描述为以下几部分:

1) 状态空间. 该空间及其障碍分别表示为  $X$  和  $X_{\text{obs}}$  ( $X_{\text{obs}} \subset X$ ), 因此自由运动集合可表示为  $X_{\text{free}}$ ,  $X_{\text{free}} = X \setminus X_{\text{obs}}$ .

2) 边界状态.  $\mathbf{p}_{\text{init}} \in X_{\text{free}}$ ,  $\mathbf{p}_{\text{goal}} \in X_{\text{free}}$ .

3) 约束条件. 协同体系中各敏捷自主在轨服务航天器在任何时刻都满足式 (2) 的约束.

4) 局部规划器. 给定协同体系中某航天器当前状态  $\mathbf{p}_i$ , 目标状态  $\mathbf{p}_j$  和时间增量  $\Delta t$ , 利用式 (10) 即可求得新节点状态  $\mathbf{p}_{i+1}$ .

5) 度量. 利用式 (6) 描述状态空间  $X_{\text{free}}$  中 2 个节点之间的度量.

### 3.4 必然碰撞区

对于运动物体避障问题, 必然碰撞区 (region of inevitable collision)<sup>[6]</sup> 的引入是必不可少的. 处于必然碰撞区的状态不论采取任何措施都将进入障碍区, 由这些状态所组成的集合表示为  $X_{\text{ric}}$ , 其中  $X_{\text{obs}} \subseteq X_{\text{ric}}$ . 因此, 新的自由运动集合表示为  $X_{\text{free}} = X \setminus X_{\text{ric}}$ . 此外, 必然碰撞区  $X_{\text{ric}}$  的大小直接受速度大小的决定, 速度或角速度变大, 必然碰撞发生的区域也将随之增大. 本文将采用安全飞行时间表述该区域, 即判断敏捷自主在轨服务航天器以当前速度或角速度经过安全飞行时间所得到的状态 (预测节点) 是否满足障碍约束条件.

### 3.5 规划方法

本文研究的服务航天器协同运动规划方法之

所以采用基于随机采样原理的快速搜索随机树方法 (RRT), 是因为该方法在解决形如式 (4) 的高维状态问题和可考虑多种约束等方面具备一定的优势. 基本的 RRT 规划方法对状态空间进行单向搜索, 改进的双向平衡搜索方式较之拥有更高的搜索效率, 并且对微小出口问题具有较强的解决能力<sup>[6,11]</sup> (所谓微小出口, 是指由复杂障碍所形成、相对狭窄的通道或出口, 这是很多考虑复杂障碍的规划问题经常遇到的难题), 因此本文将对采用双向平衡搜索方式的 RRT 规划方法进行研究, 并针对敏捷协同运动的特点加以改进, 得到新的搜索方法如图 1 所示.

```

IMPROVED_BALANCE_BIDIRECTIONAL( $p_{init}, p_{end}$ )
1  TreeA,init( $p_{init}$ ); TreeB,init( $p_{end}$ )
2  for  $i \leftarrow 1$  to  $K$ 
3    do  $p_{rand} \leftarrow$  Random_Unit_Vector( $X_{free}$ )
4      $p_n \leftarrow$  Nearest(TreeA,  $p_{rand}$ )
5      $p_s \leftarrow$  Stop_Configuration( $p_n, p_{rand}$ )
6     if  $p_n \neq p_s$ 
7       then TreeA,add_vertex( $p_s$ )
8         TreeA,add_edge( $p_n, p_s$ )
9          $p'_n \leftarrow$  Nearest(TreeB,  $p_s$ )
10        if Obs( $p_s, p'_n$ )  $\geq 0$ 
11          then  $p'_n \leftarrow$  Nearest(TreeB,  $p_{rand}$ )
12            $p'_s \leftarrow$  Stop_Configuration( $p'_n, p_{rand}$ )
13          else  $p'_s \leftarrow$  Stop_Configuration( $p'_n, p_s$ )
14          if  $p'_n \neq p'_s$ 
15            then TreeB,add_vertex( $p'_s$ )
16              TreeB,add_edge( $p'_n, p'_s$ )
17          if  $p'_s = p_s$ 
18            then return Solution
19          if Compare(TreeA, TreeB)  $< 0$ 
20            then Swap(TreeA, TreeB)
21  return Failure

```

图 1 改进的 RRT 规划方法

具体规划步骤如下:

Step 1: 对搜索树 Tree<sub>A</sub> 和 Tree<sub>B</sub> 进行初始化, 如图 1 第 1 行所示, 初始节点分别为起点  $p_{init}$  和终点  $p_{end}$ .

Step 2: 主循环进行  $K$  次迭代.

Step 3: 利用函数 Random\_Unit\_Vector() 在空间  $X_{free}$  中以均匀分布采样, 获得随机节点  $p_{rand}$ .

Step 4: 利用最近点函数 Nearest() 求取搜索树 Tree<sub>A</sub> 中到随机节点  $p_{rand}$  的最近点 (量度最小)  $p_n$ , 最近点函数并不搜索 Tree<sub>A</sub> 树中的所有点, 而只是对最新生成的  $K_{new}$  个点进行搜索, 这样不仅能够有效提高计算效率, 防止计算量呈指数增长造成“爆炸”, 而且能使搜索树的新节点在新节点集的基础上“健康”生长<sup>[6,11]</sup>.

Step 5: 利用函数 Stop\_Configuration(), 求取随机

节点  $p_{rand}$  与最近点  $p_n$  间的新节点  $p_s$ , 且新节点  $p_s$  满足约束 (2). 新节点计算方法如式 (10) 所示.

Step 6: 如果最近点  $p_n$  与新节点  $p_s$  不重合, 则分别将最近点  $p_n$  到新节点  $p_s$  的线段和新节点  $p_s$  存入搜索树 Tree<sub>A</sub>.

Step 7: 利用最近点函数 Nearest() 求取搜索树 Tree<sub>B</sub> 中到新节点  $p_s$  的最近点 (量度最小)  $p'_n$ , 求取方法如前所述.

Step 8: 如图 1 中第 10 行所示, 将障碍约束判断加入双向平衡搜索随机树方法中, 在此判断节点  $p_s$  与刚求出的最近点  $p'_n$  之间是否存在障碍, 如果存在障碍, 则如图 1 的第 11 行所示, 利用随机节点  $p_{rand}$  重新计算最近点  $p'_n$ .

Step 9: 利用函数 Stop\_Configuration() 求取新节点  $p'_s$ , 方法与 Step 5 相同 (由第 10 行判断条件决定所需最近点).

Step 10: 如果最近点  $p'_n$  与新节点  $p'_s$  不重合, 则分别将最近点  $p'_n$  到新节点  $p'_s$  的线段和新节点  $p'_s$  存入搜索树 Tree<sub>B</sub>.

Step 11: 如果 2 新节点  $p_s$  与  $p'_s$  重合, 则返回规划结果.

Step 12: 如图 1 中第 19 行所示, 引入搜索树判断函数 Compare(), 用于判断何时执行两搜索树互换函数 Swap(). 执行互换函数后, 算法将两搜索树新节点计算顺序颠倒, 即如果之前先计算搜索树 Tree<sub>A</sub> 的新节点, 则运行互换函数后, 将先计算搜索树 Tree<sub>B</sub> 的新节点.

在以上规划步骤中, Step 8 和 Step 12 中的判断函数 Compare() 是本文对采用双向平衡搜索方式的 RRT 规划方法的改进. 其中引入 Step 8 是为了利用随机采样<sup>[6]</sup>的方法解决近距障碍“陷阱”问题 (如图 2 所示), 两搜索树分别从起点和终点出发, 分别按规划方法求出新节点  $a$  和  $b$ , 但由于规划空间较小, 障碍又相对较大, 使得新节点  $b$  无法满足障碍约束. 因为新节点  $a$  只为进一步仿真步长求出的节点, 因此即使更换随机点, 节点  $a$  的位置变化依然很小, 从而使得新节点  $b$  仍无法躲避面前的障碍, 从而造成迭代无法进行的结果. 本文采用 Step 8 所述改进的方法则有效地避免了这一问题. 随机点是按照均匀分布采样得到的, 因此其较大的变化范围使新节点很容易避开面前的障碍, 如图 2 所示, 新节点  $c$  便为该方法所求.

对于 Step 12, 基本的 RRT 规划方法搜索树互换的判断条件为  $|Tree_A| > |Tree_B|$ , 即当搜索树 B 大于搜索树 A 时执行互换函数 Swap(), 这样处理主要是为了达到两搜索树生长平衡的目的, 两树平衡可以有效回避障碍, 并提高计算效率, 因此本文所采用的

判断函数 Compare() 也包括了这一点. 考虑在较小活动空间与复杂障碍情况时的运动规划如图 2 所示, 随机点的随机性决定了新节点  $c$  并不一定满足障碍约束条件. 因此, 为有效回避障碍和提高计算效率<sup>[6]</sup>, 在判断函数 Compare() 中增加条件, 即哪棵随机树所求新节点的预测节点不满足障碍约束条件, 则执行互换函数, 换为先计算哪棵随机树. 在图 2 中, 如果新节点  $c$  的预测节点不满足障碍约束条件, 则重新生成随机点, 并先计算新节点  $c$ , 再计算新节点  $a$ .

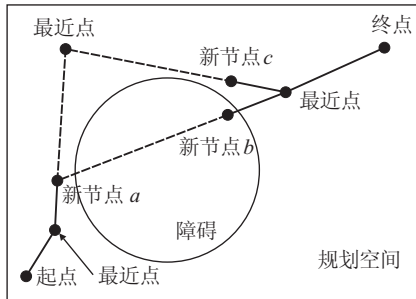


图 2 较小活动空间与复杂障碍情况时的运动规划

### 4 仿真算例

下面对 3 个服务航天器的协同规划进行仿真, 仿真基本参数如表 1 所示. 3 个航天器分别从不同地点以相同初始姿态按照各自的运动时间要求飞向不同的目标, 且终止姿态不同.

表 1 仿真基本参数

参数名称	数值
起始位置/m	$[0 \ 0 \ 0]^T, [5 \ 0 \ 0]^T, [10 \ 0 \ 0]^T$
目标位置/m	$[30 \ 40 \ 40]^T, [35 \ 40 \ 40]^T, [40 \ 40 \ 40]^T$
起始姿态	$[0.6533 \ 0.6533 \ -0.2706 \ 0.2706]^T$
	$[0 \ -0.3827 \ -0.9239 \ 0]^T$
目标姿态	$[0.3536 \ -0.3536 \ -0.8536 \ 0.1464]^T$
	$[0.6533 \ -0.2706 \ -0.6533 \ 0.2706]^T$
起始时刻/s	0, 0, 0
目标时刻/s	72, 74, 76
飞行空域/m	$[40 \ 40 \ 40]^T$

敏捷自主在轨服务航天器以本体上特征点作为几何约束, 各特征点具体位置参数见表 2.

表 2 特征点位置

标号	位置/m	标号	位置/m
1	$[1 \ 1 \ 1]^T$	7	$[-1 \ -1 \ -1]^T$
2	$[1 \ 1 \ -1]^T$	8	$[-1 \ -1 \ 1]^T$
3	$[-1 \ 1 \ -1]^T$	9	$[3 \ 3 \ 0]^T$
4	$[-1 \ 1 \ 1]^T$	10	$[1 \ 3 \ 0]^T$
5	$[1 \ -1 \ 1]^T$	11	$[3 \ -3 \ 0]^T$
6	$[1 \ -1 \ -1]^T$	12	$[1 \ -3 \ 0]^T$

利用改进的 RRT 规划算法对敏捷自主在轨服务航天器空间运动进行规划, 规划算法相关参数如表 3 所示. 其中权系数的选择, 其主要目的是将时间等各状态调整到相同或近似的数量级, 以便在规划过

程中各状态树均匀“生长”和相交, 尽量避免其中的某些状态树过早相交而影响到其他状态树的“生长”.

表 3 规划算法相关参数

参数名称	数值	参数名称	数值
权系数 $K_1$	0.5	权系数 $K_5$	0.2
权系数 $K_2$	1.0	备选最近点数量 $K_{new}$	500
权系数 $K_3$	0.1	安全飞行时间/s	2
权系数 $K_4$	0.1	规划迭代次数 $K$	100 000

几何障碍为飞行空域内 2 个球形障碍, 半径为 8 m, 球心分别位于  $[14 \ 14 \ 14]^T$  m 和  $[28 \ 28 \ 28]^T$  m. 安全距离  $R_{Safe}$  为 4.5 m.

为简化考虑, 认为各敏捷自主在轨服务航天器在执行一次服务任务期间, 速度、加速度、姿态角速度和角加速度的最大值约束恒定不变. 根据现有研究的敏捷航天器基本能力和飞行空域与运动时间等条件的限制, 确定速度、加速度、姿态角速度和角加速度的最大值约束分别为 3 m/s, 0.5 m/s<sup>2</sup>, 0.2 rad/s 和 0.05 rad/s<sup>2</sup>.

仿真采用主频 2.66 GHz, 内存 2 GB 的计算机, 并在 C++ 环境下编写仿真程序, 利用 OpenGL 制作三维视景.

设置仿真步长为 0.05 s, 利用文中改进的 RRT 规划方法对敏捷自主在轨服务航天器协同运动进行规划, 经过 521 s 仿真结束, 3 个航天器分别获得 273, 292 和 290 个可行结果, 取其中 1 条航迹, 相关仿真结果如图 3~图 5 所示.

图 3 和图 4 分别为位置、姿态四元数、速度、加速度、角速度、角加速度、3 条航迹之间的相对距离与时间的仿真曲线, 可以看出, 所有状态都在规定时间到达预期目标, 并满足碰撞约束. 图 5 为敏捷自主在轨

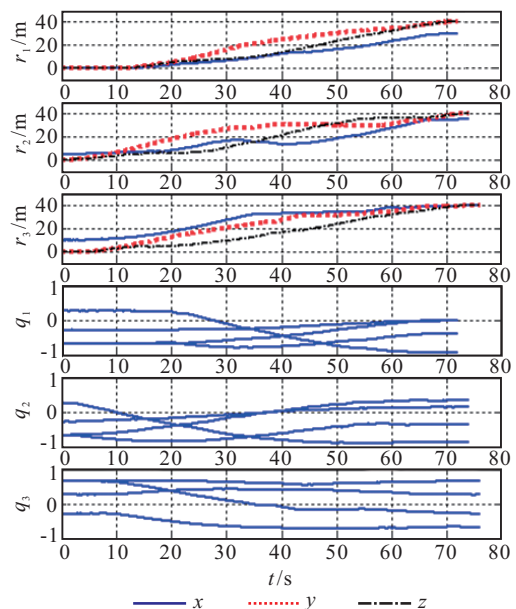


图 3 位置、姿态四元数-时间曲线

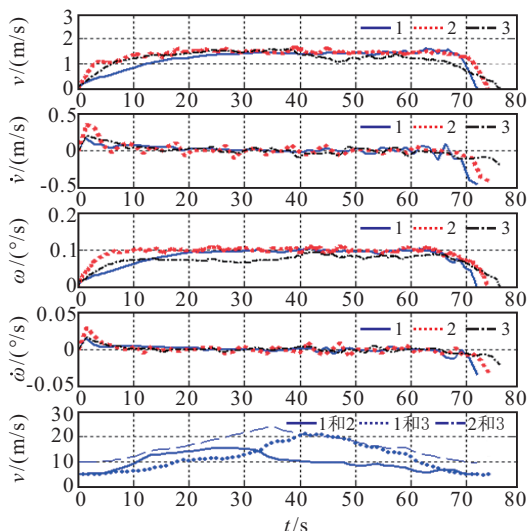


图 4 速度、加速度、角速度、角加速度、3 条航迹之间的相对距离-时间曲线

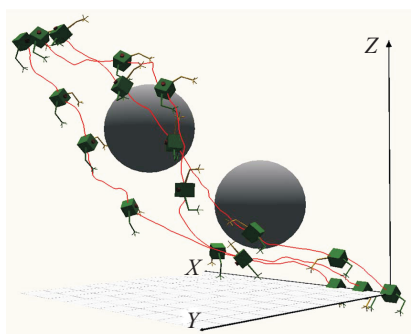


图 5 三维视景仿真图

服务航天器协同运动的三维视景仿真图, 图中所示为 3 个航天器于 24 s, 36 s, 50 s, 60 s 以及起始和终止时刻的运动状态。

分别设置仿真步长 0.05 s, 0.1 s, 0.5 s, 1 s 和 2 s, 同样利用前述规划方法对敏捷自主在轨服务航天器协同运动进行规划, 所得相关结果如表 4 所示。从表 4 可以看出, 相同迭代次数下, 随着仿真步长的不断增大, 仿真所需时间不断减少, 在计算相同两点间距离时, 步长增大造成该两点间节点数量减少, 从而降低了计算量, 亦即缩短了仿真时间。随着仿真步长的不断增大, 规划获得的可行结果数量也在不断减少, 这是因为在规划过程中, 某一时间-空间区域内, 规划所获得的节点密度变小, 因此获得一个可行结果后, 周围通过扩展得到的其他结果数量也相应地减少了。随着仿真步长的不断增大, 首次获得可行结果的时间不断减小, 而为获得结果所付出的代价(利用式(5)计算可行航迹中每两点间的度量之和, 这里为统一评价标准, 将所有相应仿真步长的规划所得航迹按 0.05 s 步长进行线性拟和, 而后再计算其代价)不断增大, 这是因为步长增大加速了起点与目标的连接速度, 但也同时造成了航迹平滑度的下降, 从而增大了航迹的长度, 亦即提高了相应的代价。

表 4 不同仿真步长对应的仿真结果

仿真步长/s	仿真时间/s	可行结果/s	首次获得结果时间/s	代价
0.05	546	221 350 323	342	188.1
0.1	437	238 241 272	161	197.7
0.5	353	143 172 165	32	237.9
1	341	79 95 172	19	283.8
2	337	34 73 130	12	366.0

利用该方法对单个敏捷航天器空间运动进行仿真, 同样采用 5 种仿真步长, 所得仿真时间分别为 268 s, 244 s, 226 s, 228 s 和 230 s。可以看出, 3 个航天器协同运动规划所需仿真时间是单个航天器的 1.5~2.0 倍, 这说明随着规划对象状态维数的增加, 仿真时间呈线性趋势增长, 而额外消耗的时间是仿真中矛盾协调所用去的时间。

在同样的条件下, 本文亦采用未改进的 RRT 规划方法, 分别设置仿真步长 0.05 s, 0.1 s, 0.5 s, 1 s 和 2 s, 执行 100 000 次仿真迭代。5 种仿真步长分别进行 10 次仿真, 无一次仿真完成, 均陷入无限循环, 也就是前述的“陷阱”。将仿真步长延长至 3 s 和 5 s, 结果也是如此。将球形几何障碍半径减小至 3 m, 进行 10 次仿真, 仿真步长为 0.05 s, 0.1 s, 0.5 s, 1 s, 2 s 和 3 s 时均无结果; 5 s 时, 6 次陷入“陷阱”, 4 次成功得到结果, 首次获得结果时间均大于 800 s。由此可见, 未改进的双向平衡搜索随机树方法对障碍的跨越能力十分有限, 当仿真步长较小时易陷入“陷阱”, 当仿真步长较大时, 可在一定程度上对“陷阱”实现跨越, 但由于搜索树互换的判断条件过于简单, 降低了相应的计算效率, 因而所需的规划时间略长。

## 5 结 论

本文利用基于采样理论快速搜索随机树方法针对高维问题具有较强适应性这一特点, 对敏捷自主在轨服务航天器协同运动规划问题进行了研究。采用随机理论等方法对采用双向平衡搜索方式的 RRT 规划方法进行改进, 使之较好地回避了规划过程中的“陷阱”, 提高了仿真效率。仿真结果证明, 该方法符合敏捷航天器的特点, 并能够在指定时间到达指定地点, 而且对规划中的“陷阱”问题具有较强的解决能力。此外, 该方法还可通过调整仿真步长减小首次获得结果的时间, 从而可以更加快速得到可行航迹, 但航迹代价将相应变大。

## 参考文献(References)

- [1] 崔乃刚, 王平, 郭继峰, 等. 空间在轨服务技术发展综述[J]. 宇航学报, 2007, 20(4): 33-39.  
(Cui N G, Wang P, Guo J F, et al. A review of on-orbit servicing[J]. J of Astronautics, 2007, 20(4): 33-39.)
- [2] 陈小前, 袁建平, 姚雯, 等. 航天器在轨服务技术[M]. 北京: 中国宇航出版社, 2009: 3-27.  
(Chen X Q, Yuan J P, Yao W, et al. On-orbit servicing

- technology of spacecraft[M]. Beijing: China Astronautics Publishing House, 2009: 3-27.)
- [3] Donald M Waltz. On-orbit servicing of space systems[M]. USA: Krieger Publishing Company, 1993: 1-283.
- [4] Roger A B, McInnes C R. Safety constrained free flyer path planning at the international space station[J]. AIAA J on Guidance, Control and Dynamics, 2000, 23(6): 971-979.
- [5] Richards A, Schouwenaars T, How J P, et al. Spacecraft trajectory planning with collision and plume avoidance using mixed-integer linear programming[J]. AIAA J on Guidance, Control, and Dynamics, 2002, 25(4): 755-765.
- [6] LaValle S M. Planning algorithms[M]. Illinois: Cambridge University Press, 2004: 228-238.
- [7] LaValle S M, James J, Kuffner Jr. Randomized kinodynamic planning[C]. Proc of the 1999 IEEE Int Conf on Robotics and Automation. Detroit, 1999: 473-479.
- [8] Ian Garcia, Jonathan P How. Trajectory optimization for satellite reconfiguration maneuvers with position and attitude constraints[C]. American Control Conf. Portland, 2005: 889-894.
- [9] Georges S Aoude, Jonathan P How, Ian M Garcia. Two-stage path planning approach for designing multiple spacecraft reconfiguration maneuvers[C]. Int Symposium on Space Flight Dynamics. Annapolis, 2007: 134-149.
- [10] Emilio Frazzoli. Quasi-random algorithms for real-time spacecraft motion planning and coordination[C]. Int Astronautical Congress. Houston, 2002: 485-494.
- [11] Kavraki L E, LaValle S M, Siciliano In B, et al. Springer handbook of robotics[M]. Berlin: Springer-Verlag, 2008: 109-133.
- [12] Yershova A, LaValle S M. Motion planning for highly constrained spaces[M]. Berlin: University of Illinois, 2008: 297-306.
- [13] Yu J, Liberzon D, LaValle S M. Rendezvous without coordinates[C]. Proc IEEE Conf Decision and Control. Cancun, 2008: 488-495.
- [14] Frazzoli E, Dahleh M A, Feron E, et al. A randomized attitude slew planning algorithm for autonomous spacecraft[C]. AIAA Guidance, Navigation, and Control Conf and Exhibit. Montreal, Quebec: AIAA, 2001: 1-8.
- [15] 仲维国, 崔平远, 崔枯涛. 航天器复杂约束姿态机动的自主规划[J]. 航空学报, 2007, 28(5): 1091-1097. (Zhong W G, Cui P Y, Cui K T. Autonomous attitude maneuver planning for spacecraft under complex constraints[J]. Acta Aeronautica et Astronautica Sinica, 2007, 28(5): 1091-1097.)
- [16] Creamer G, DeLaHunt P, Gates S, et al. Attitude determination and control of clementine during lunar mapping description and on-orbit performance[C]. AIAA Guidance, Navigation, and Control Conf. Baltimore, 1995: 920-930.

(上接第1698页)

- [4] Safadi H. Local path planning using virtual potential field[EB/OL]. (2007-04-18)[2010-01-13]. <http://www.cs.mcgill.ca/hsafad/robotics/index.html>.
- [5] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1999: 156-161. (Zhou M, Sun S D. Genetic algorithms: Theory and applications[M]. Beijing: National Defense Industry Press, 1999: 156-161.)
- [6] Wu X P, Feng Z P, Zhu J M. GA-based path planning for multiple AUVs[J]. Int J of Control, 2007, 7(80): 180-185.
- [7] 何娟, 涂中英, 牛玉刚. 一种遗传蚁群算法的机器人路径规划方法[J]. 计算机仿真, 2010, 27(3): 170-174. (He J, Tu Z Y, Niu Y G. A method of mobile robotic path planning based on integrating of GA and ACO[J]. Computer Simulation, 2010, 27(3): 170-174.)
- [8] Li M X, Xia X H, Zhang Y. A path planning method based on adaptive genetic algorithm for a shape-shifting robot[J]. Computer and Information Science, 2010, 3(4): 208-214.
- [9] 刘春阳, 程亿强, 柳长安. 基于改进势场法的移动机器人避障路径规划[J]. 东南大学学报(自然科学版), 2009, 39(增): 116-120. (Liu C Y, Cheng Y Q, Liu C A. Anti-collision path planning for mobile robot based on modified potential field method[M]. J of Southeast University: Natural Science, 2009, 39(S): 116-120.)
- [10] Al Taharwa I, Sheta A, Al Weshah M. A mobile robot path planning using genetic algorithm in static environment[J]. J of Computer Science, 2008, 4(4): 341-344.
- [11] 梁毓明, 徐立鸿. 基于改进模拟退火混合算法的移动机器人全局路径规划[J]. 控制与决策, 2010, 25(2): 237-240. (Liang Y M, Xu L H. Mobile robot global path planning based modified simulated annealing hybrid algorithm[M]. Control and Decision, 2010, 25(2): 237-240.)