IAA-AAS-DyCoSS2-14-06-08

RESEARCH ON COOPERATIVE MOTION PLANNING OF AGILE AND AUTONOMOUS ON-ORBIT SERVICING SPACECRAFT WITH COMPLICATED SPACE ENVIRONMENT

Jifeng Guo,[*] Jinjun Shan,[†] and Xiaobu Xue[‡]

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

INTRODUCTION

Many applications and studies of current interest, including on-orbit servicing of large space structures, space stations or astronomical telescope, require maneuvering technique in close proximity of them with on-orbit servicing spacecraft. [1, 2] In these servicing missions, spacecraft must fly along a feasible trajectory generated by an adapted motion planning algorithm. This trajectory needs to satisfy the operation requirements, such as complex environment, minimizing the cost of fuel or flying time, considering constraints of collision or bound of impulse. Because of the differences of spacecraft thruster (impulse and continuous thruster), spacecraft maneuvering control in close proximity of servicing target are divided into discrete and continuous control. The motions of continuous control mode are used to complicated continuous operation and the motions of discrete control mode are used to move in a relatively large space. So the servicing spacecraft motion planning is inspired mainly from this and divided into two major directions.

In recent years, a lot of literatures have studied on motion planning of continuous control mode.[3, 4, 5] For example, Colin R. McInnes used artificial potential functions to generate safe trajectories around space station to dock,[6] Jonathan P. How solved this kind of problem based on a transcription of the motion planning problem into a Mixed-Integer Linear Program (MILP).[7]

[*] Associate Professor, Department of Astronautics Engineering, Harbin Institute of Technology, Post Box 345, 92 West Dazhi Street, Harbin, Heilongjiang, 150001, China.
[†] Associate Professor, Department of Earth and Space Science and Engineering, York University, Room 255, Petrie Science & Engineering Building, 4700 Keele Street, Toronto, ON M3J 1P3, Canada.
[‡] Engineer, Shanghai Institute of Satellite Engineering, 251 Huaning Road, Shanghai 200240, China.

Rapidly-exploring Random Trees (RRTs) algorithm based on randomized motion planning not only adapts higher dimensional space planning problem but also considers differential constraints in the generation process of the search trees. [8] And this planner can effectively avoid existence of local minima. Moreover, RRT planner shows good performance in fast and uniform exploration of the configuration space. Emilio Frazzoli used Quasi-Random Algorithms based on RRT to solve real-time servicing spacecraft motion planning and coordination in close proximity of one another. [9, 10]

According to the requirements of on-orbit system servicing in the future industrialized and military space, a cooperative motion planning method of assembling and servicing of multiple agile autonomous servicing spacecraft systems with continuous thrust in complicated space environment is studied in this paper. The improved high-dimensional motion planning method considering state of translation, rotation and time, is used for cooperative motion planning of servicing spacecraft accurate and agile motion with complicated space environment. This planner satisfies the requirement of agile and autonomous spacecraft maneuvering with complex constraints. Section II provides the relative motion described by new state transition matrix for relative motion on an arbitrary elliptical orbit. Then new approach based on RRT is added independent variable like time or true anomaly to adapt to non-static dynamical environment on elliptical orbit. Furthermore the basic planner is split into two phases. The first phase is that simpler constraints are used in this planner. And the full high-dimensional constraints are used in the second one to plan every edge in the trajectory planned in the first phase. Finally, some simulation examples are presented and discussed for the proposed planning algorithm.

RELATIVE MOTION EQUATIONS

The method of propagating an initial state to a final state without numerical integration is particularly meaningful for relative motion problems. It is possible to reduce computational complexity and increase the calculation speed of motion planning. Many attempts to solve the differential equations of relative motion for the elliptical orbit of arbitrary eccentricity can be seen in many literatures. Furthermore, Koji Yamanaka and Finn Ankersen's solution to differential equations of relative motion on an arbitrary elliptical orbit is described in a convenient state transition matrix form. And this state transition matrix has no singularity at $e = 0$ and is valid for an arbitrary elliptical orbit ($0 \le e < 1$).[10] The formulation details are as follows.

The relative motion equations, that describe on-orbit relative motion of spacecrafts without external forces imposed on the orbital coordinate system are:

$$\begin{cases} \ddot{x} = 2\omega\dot{z} + \omega^2 x + \dot{\omega} z - \mu x / r^3 \\ \ddot{y} = -\mu y / r^3 \\ \ddot{z} = -2\omega\dot{x} + \omega^2 z - \dot{\omega} x + 2\mu z / r^3 \end{cases} \tag{1}$$

where ω is the orbital angular rate and r is the distance measured from the centre of the Earth to the origin of the coordinate system.

The transformation is:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \left(1 + e\cos\theta\right) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2}$$

Then the relative state can be obtained by the state transition matrix in which the true anomaly θ about the origin of orbital coordinate system is the only independent variable instead of time t as follows:

$$\begin{bmatrix} \boldsymbol{r}_t \\ \boldsymbol{v}_t \end{bmatrix} = \Phi \begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{v}_0 \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{v}_0 \end{bmatrix} \tag{3}$$

where Φ , Φ_{11}, Φ_{12}, Φ_{21}, Φ_{22},

$$\boldsymbol{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

The relative state at arbitrary time can be calculated by this state transition matrix, with initial state, *e* and the value of the true anomaly given, while not performing numerical integration. This method is applied to the relative motion planning for elliptical orbit in this paper.

The constraints of kinematics and dynamics are as follows:

$$\begin{cases} v \le v_{\max}(\boldsymbol{r}, t) \\ \dot{v} \le \dot{v}_{\max}(\boldsymbol{r}, t) \\ \omega \le \omega_{\max}(\boldsymbol{q}, t) \\ \dot{\omega} \le \dot{\omega}_{\max}(\boldsymbol{q}, t) \end{cases} \tag{4}$$

where $v_{\max}$, $\dot{v}_{\max}$, $\omega_{\max}$ and $\dot{\omega}_{\max}$ are the maximal value of the velocity, acceleration, angular velocity and angular acceleration.

IMPROVED COOPERATION MOTION PLANNING ALGORITHM

In this Section, the improved randomized sampling motion planning method is described in detail. It belongs to problem descriptions, formulations of the algorithm and schemes.

Problem Descriptions

The class of problem considered in this paper can be formulated in terms of below components:

1) State Space: A topological space and its obstacles are X and X_{obs} ($X_{obs} \subset X$), and the violation-free set is X_{free} ($X_{free} = X \setminus X_{obs}$).

2) Boundary Values: $x_{init} \in X_{free}$ and $x_{goal} \in X_{free}$.

3) Constraints: Constraints Eq. (4) are satisfied for spacecraft at all times.

4) Incremental Simulator: Give the current state $x(t)$ and inputs (continuous thrust) applied at current time *t* and time interval Δt , and calculate $x(t+\Delta t)$.

5) Metric: A real-valued function, ρ , which specifies the distance between pairs of points in X_{free} .

Improved Randomized Sampling Cooperation Motion Planning Algorithm

This motion planning method for an arbitrary elliptical orbit similar to general randomized motion planning is indicated by branches, milestones and trees consisting of them. Because the time between spacecraft impulse thrust is bigger and search space is relatively limited, and autonomous assembly spacecrafts that maneuver to the final state from initial state need few impulse thrust, single-tree search mode is adopted.

The basic ideas and steps of improved randomized sampling motion planning method in Figure 1 and Table 1 are as follow,

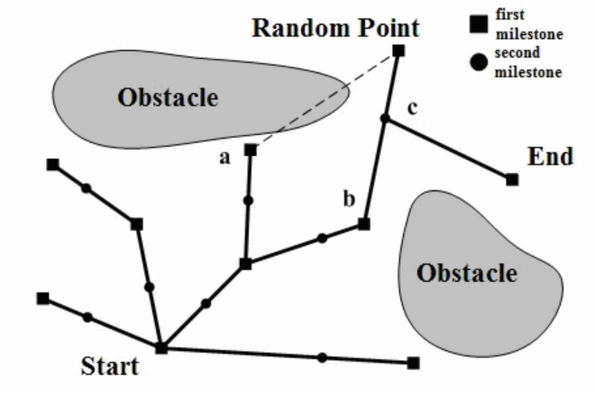


Figure 1. Improved Randomized Sampling Motion Planning Algorithm.

Table 1. Basic Steps of Improved Randomized Sampling Motion Planning Algorithm.

Mixed_Motion_Planning$(\boldsymbol{p}_{init}, \boldsymbol{p}_{end})$	
1	Tree.init$(\boldsymbol{p}_{init})$
2	for $i \leftarrow 1$ to n_I
3	First_Milestone( )
4	Second_Milestone( )
5	if Final_Milestone$\left(\boldsymbol{p}_{fms}, \boldsymbol{p}_{sms}, \ldots, \boldsymbol{p}_{end}\right) < 0$
6	return *Solution*
7	return *Failure*

1. Make the start point as initial milestone.

2. Generate random statements as candidate milestones in freedom motion state space.

3. Search the closest one between current milestone in candidate milestones and all constraints must be satisfied. If it exists, make this milestone as first milestone, save it into random search tree and make the closest milestone as its parent node. Otherwise, return to step 2 and create new candidate milestones.

4. In order to make random search tree rapidly grown, make some second milestone between first milestone and its parent node.

5. Attempt to connect with first milestone, second milestone and end. If time and constraints are satisfied, feasible planning results are obtained. Otherwise, return to step 2.

Furthermore, the basic steps of first milestone are the following (reference Table 2):

Table 2. Basic Steps of First Milestone.

First_Milestone( )	
1	do $\boldsymbol{p}_{rand} \leftarrow \text{Random\_State}\left(X_{free}\right)$
2	If $\text{Constraints}\left(Tree, \boldsymbol{p}_{rand}\right) < 0$
3	then $\boldsymbol{p}_{pn} \leftarrow \text{Nearest}\left(Tree, \boldsymbol{p}_{rand}\right)$
4	$\boldsymbol{p}_{fms} \leftarrow \boldsymbol{p}_{rand}$
5	$Tree.\text{add\_vertex}\left(\boldsymbol{p}_{fms}\right)$
6	$Tree.\text{add\_edge}\left(\boldsymbol{p}_{pn}, \boldsymbol{p}_{fms}\right)$
7	return *Solution*
8	return *Failure*

1. Sample uniformly a candidate milestone x_{rand} in X_{free}.

2. Make a set of neighbours of x_{rand} chosen from V named N_n ($N_n = V$ in general), N_n is the number of neighbours.

3. Sample uniformly a flying time t_p $(p = 1 ... N_S)$ which starts from parent node and ends to candidate milestone x_{rand}. N_S is the number of sampling ($N_S = 256$ in general). Obtain v from relative motion explicit Eq. (3) for every neighbour x_j ($x_j \in N_n$). If v and the edge of x_j to x_{rand} satisfy the constraints Eq. (4), x_j is saved to the set of parent nodes V_{pn}. Otherwise, the new candidate milestone x_{rand} is sampled. Obtain t_j, the minimum of t_p.

4. If V_{pn} is not empty, find the nearest neighbour x_{pn} (parent node) from current x_{rand} by the cost J.

5. Obtain new first milestone x_{fms}, and its t_{fms}, $Tree_{ms}$, $seq\Delta V_{ms}$ and $seqTime_{ms}$. t_{fms} is the flying time that starts from initial node to the new first milestone as independent variable to calculate the next first milestone used by explicit equation.

The second milestones are added to the tree (i.e. an edge which starts from parent node to the first milestone is split into two edges, the second milestone is inserted in between). Applying of the second milestone can efficiently increase the probability of obtaining solution. Its method looks like the first milestone expressed. n_s is the number of the second milestones expected. Similarly, the t_{rand} is flying time between t_{pn} and t_{fms}.

Cooperation Strategy

The principal contradiction of cooperation motion planning is focus on the collision avoidance between every agile and autonomous servicing spacecrafts. According to these contradictions, the two kinds of cooperation strategies of motion planning algorithm are presented as follow,

1. The contradictions between every spacecrafts are reconciled by adopting the concentrating cooperation method. During generating the random search trees of every spacecrafts, the contradictions between every spacecrafts are considered and the collisions between new nodes of every search trees are avoided. So every search trees can be grown in this cooperation strategy.

2. The contradictions are also reconciled by adopting the distributed cooperation method. The trajectories of every spacecrafts are planned by themselves using randomized sampling method. Then all search trees or planning results by way of planning models are shared with every spacecrafts at real time and the collisions between new nodes of every search trees and shared models are avoided by themselves. So every search trees can be similarly generated in this cooperation strategy.

In this paper the two strategies are collaboratively adopted in the whole process of cooperation motion planning. The concentrating cooperation method is used for global planning and the distributed cooperation method is applied for local planning.

EXAMPLES

In this section, some simulation examples are presented and discussed for the proposed planning algorithm.

The planning algorithm is implemented in C++ on a 2.66 GHz Core 2 machine, with 2 GB of RAM, running Windows XP. The main loop iterates for 20000 times in this algorithm. The simulation parameters of three spacecrafts are shown in Table 3.

Table 3. Simulation Parameters

Serial Number of Spacecrafts	1	2	3
Starting Position (m)	$[0\ \ 0\ \ 0]^T$	$[5\ \ 0\ \ 0]^T$	$[10\ \ 0\ \ 0]^T$
Ending Position (m)	$[30\ \ 40\ \ 40]^T$	$[35\ \ 40\ \ 40]^T$	$[40\ \ 40\ \ 40]^T$
Starting Attitude	$\begin{bmatrix} 0.6533 \\ 0.6533 \\ -0.2706 \\ 0.2706 \end{bmatrix}$	$\begin{bmatrix} 0.6533 \\ 0.6533 \\ -0.2706 \\ 0.2706 \end{bmatrix}$	$\begin{bmatrix} 0.6533 \\ 0.6533 \\ -0.2706 \\ 0.2706 \end{bmatrix}$
Ending Attitude	$\begin{bmatrix} 0 \\ -0.3827 \\ -0.9239 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.3536 \\ -0.3536 \\ -0.8536 \\ 0.1464 \end{bmatrix}$	$\begin{bmatrix} 0.6533 \\ -0.2706 \\ -0.6533 \\ 0.2706 \end{bmatrix}$
Starting Time (s)	0	0	0
Ending Time (s)	72	74	76

The simulation step is 0.05 seconds. The 273, 292 and 290 feasible results are respectively obtained during 521 seconds. The three feasible trajectories of three spacecrafts are shown in Figure 2 and relative distances between three trajectories are shown in Figure 3.

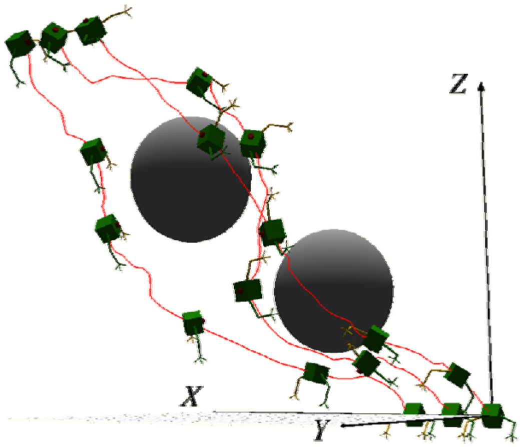


Figure 2. The Feasible Trajectories of Three Spacecrafts.

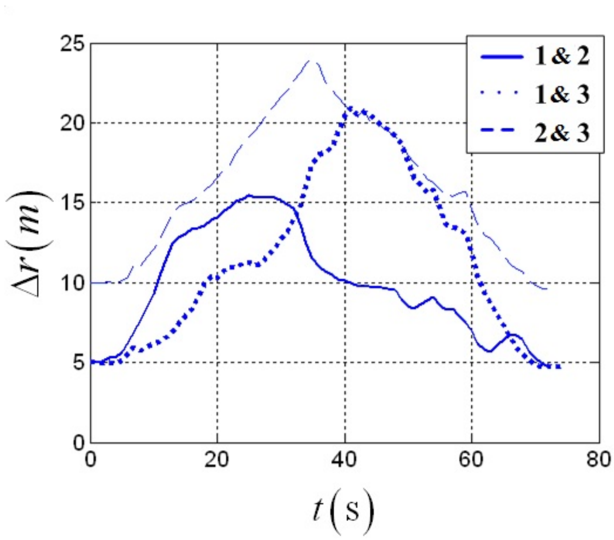


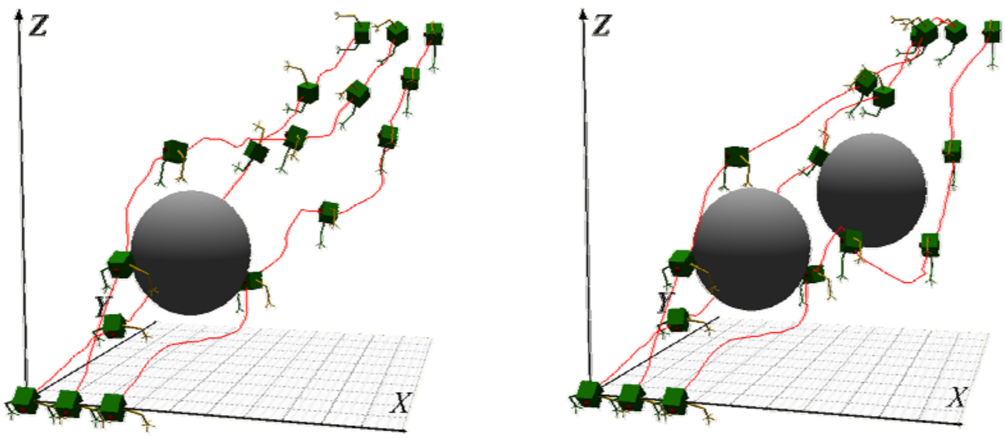
Figure 3. The Relative Distances between Three Trajectories.

The simulation steps are respectively set as 0.05s, 0.5s, 0.1s, 1s and 2s and the same motion planner of three spacecrafts is implemented. The results of the simulation with different steps are shown in Table 4.

Table 4. Results of the Simulation with Different Steps

Step (s)	Run Time (s)	Amount of Feasible Results for Three Spacecrafts			Time of First Result Obtained (s)
0.05	546	221	350	323	342
0.1	437	238	241	272	161
0.5	353	143	172	165	32
1	341	79	95	172	19
2	337	34	73	130	12

Furthermore, another simulation is implemented to verify the cooperativity of planning algorithm in the concentrating and distributed cooperation strategies and the results are shown in Figure 4.



(a) Advance Planning Trajectories

(b) Real-time planning trajectories when the obstacle appears at 24s

Figure 4. Advance Planning and Real-time Trajectories.

The calculating results reveal the feasibility of cooperative motion planning method based on randomized sampling upon the problem of multiple autonomous assembly spacecrafts.

CONCLUSION

A cooperative motion planning algorithms is researched to solve the agile and autonomous servicing spacecraft maneuvering in close proximity of complicated space target on elliptical orbit. This planner can use the independent variable like time to adapt non-static environment as gravitational field, and obtain satisfying result. Such algorithm can't yield an optimal solution, but it deals with very complex motion planning problem very efficiently, and yield a feasible solution extremely rapidly to satisfy the requirement of real-time planning. Finally, two-phase planner is represented and proven to adapt for the conditions of larger range of motion and more complicated obstacles. Moreover, it increases the computation speed greatly for above conditions.

ACKNOWLEDGMENTS

This work is partially supported by China Civil Aerospace Advance Research Project Funding, China National Nature Science Funding (61005060) and Key Laboratory Opening Funding of Technology of Micro-Spacecraft.

REFERENCE

[1] L. S. Homem-de-Mello, “Sequence planning for robotic assembly of tetrahedral truss structures,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, 1999, pp. 304-312.

[2] D. M. Waltz, *On-orbit servicing of space systems*, Krieger Pub. Co, 1993.

[3] S. M. LaValle, J. J. Kuffner, “Randomized kinodynamic planning”, *the 1999 IEEE International Conference on Robotics & Automation, Detroit, USA*, 1999, pp. 473-479.

[4] G. Chen, L. Zhang etc., “Repetitive motion planning of free-floating space manipulators”, *International Journal of Advanced Robotic Systems*. vol. 10, 2013, pp. 1–11.

[5] B. P. Larouche, Z. H. Zhu, “Investigation of impedance controller for autonomous on-orbit servicing robot”, *Canadian Aeronautics and Space Journal*, Vol. 59, No. 1, 2013, pp. 15–24.

[6] A. B. Roger, C. R. McInnes, “Safety constrained free flyer path planning at the international space station”, AIAA J. on Guidance, Control an Dynamics, Vol. 23, No. 6, 2000, pp. 971-979.

[7] T. Schouwenaars, B. De Moor, E. Feron, J. How, “Mixed integer programming for multi-vehicle path planning”, *European Control Conference, Porto, Portugal*, 2001, pp. 2603-2608.

[8] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, 2004.

[9] E. Frazzoli, M. Dahleh, E. Feron, R. Kornfeld, “A randomized attitude slew planning algorithm For autonomous spacecraft”, *AIAA Guidance, Navigation and Control Conference, Montreal, CA*, 2001, pp. 4155-4159.

[10] E. Frazzoli, “Quasi-random algorithms for real-time spacecraft motion planning and coordination”, *International Astronautical Congress, Houston, USA*, 2002, pp. 485-494.

[11] S. Carpin, “Randomized motion planning: a tutorial”, *International Journal of Robotics and Automation*, vol. 206, no. 2, 2006, pp. 2804-2817.

[12] G. S. Aoude, J. P. How and I. M. Garcia, “Two-stage path planning approach for designing multiple spacecraft reconfiguration maneuvers”, *the International Symposium on Space Flight Dynamics, Annapolis, USA*, 2007, pp. 513-519.