Volume 7 Issue 12, 2021
DOI: 10.6919/ICJE.202112_7(12).0066

RRT track planning method based on effective neighborhood

Jun Jiang, Xunli Zhang, Ming Li, Hangfei Wang, Junchao Zhao
Rocket Army Engineering University, 710038, China

Abstract

Compared to Problem 1, Problem 2 adds the constraint of turning radius, that is, the minimum turning radius is 200m. The task to be completed in this question is to plan a track that meets the constraints of the turning radius and the correction point. In addition, it is also necessary to meet the two optimization goals of the smallest track length and the smallest number of corrections through the correction area. The core of question 1 lies in the selection of correction points, while question 2 becomes a trajectory optimization problem. If the turning radius constraint and the correction point constraint are directly added to the RRT track for solution, the optimization parameters are huge and the calculation is difficult.

Keywords

NRRT, Optimal track, Monte Carlo, Bezier.

1. Introduction

As shown in Figure 1, during the flight from point A to point B, the aircraft in the three-dimensional space needs real-time positioning and error correction. The blue and red in the figure are the error correction points. After flying for a certain period of time, the aircraft needs to arrive at the correction point for error correction, otherwise the flight mission will fail due to the gradual accumulation of errors. The task to be completed in this question is to plan a trajectory under given constraints. In addition to satisfying the above constraints, the trajectory also needs to meet the two optimization goals of the smallest track length and the smallest number of corrections through the correction area. entire path.

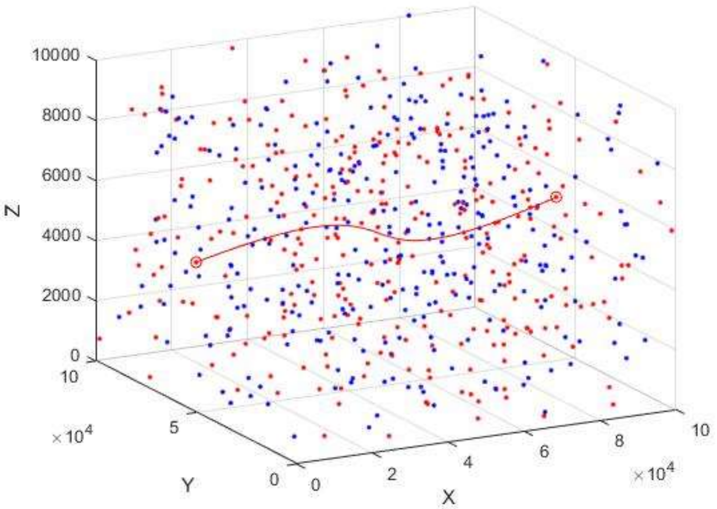

Figure 1 Schematic diagram of the trajectory

2. Construction of trajectory planning model

The previous article gave the definition of the track edge, and whether two nodes can form a track edge (that is, whether they can be adjacent) needs to meet certain conditions. The constraint condition of this article is mainly the positioning error constraint. According to the problem setting, the vertical error and horizontal error of the aircraft during flight should always be less than units before it can fly to the end according to the planned path. Therefore, there are constraints on the positioning error before point correction.

$$\begin{cases} \delta_{j0}^{\perp} = d_{ij}\delta + \delta_i^{\perp} \\ \delta_{j0}^{\parallel} = d_{ij}\delta + \delta_i^{\parallel} \end{cases}$$

$$\begin{cases} \delta_j^{\perp} = 0, \delta_j^{\parallel} = \delta_{j0}^{\parallel}, if(N_j \in U^{\perp}) \\ \delta_j^{\parallel} = 0, \delta_j^{\perp} = \delta_{j0}^{\perp}, if(N_j \in U^{\parallel}) \end{cases}$$

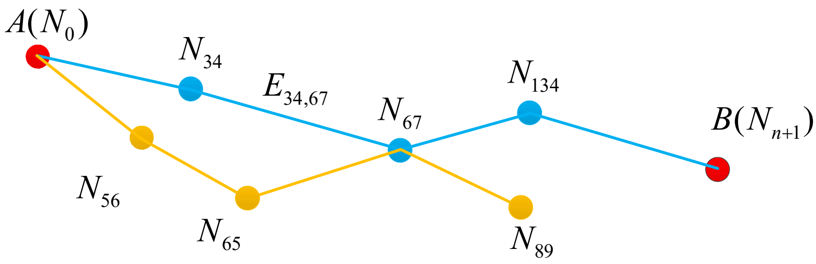

Fig. 2 Schematic diagram of adjacency relationship between nodes

Through the above analysis, it can be seen that the biggest difference between the trajectory planning of this problem and the traditional trajectory planning is the solution space. The solution space of traditional trajectory planning is usually infinite and continuous. In the search update process, the search step and search direction are less restricted, and the search can be carried out at any step length and direction. The solution space of this problem is Limited and discrete, the search step length is limited by the accumulated error information, and the search direction is limited by the prior node position information. As a result, the possibility of finding a feasible solution for this problem is lower than that of the traditional trajectory planning problem.

3. Improved RRT track planning algorithm based on effective neighborhood

3.1 Basic Principles of RRT

The key of the classic RRT algorithm [2] for trajectory planning is the construction of a random tree. First, the starting point A is selected in the task area as the root node (Root Point), and then a leaf node (Leaf Point) is grown from the root node. The leaf node grows out of the leaf node again until it grows to the end point, that is, a random tree is obtained, and finally the complete track can be obtained by looking for the leaf node in the reverse direction from the end point. Figure 3 shows the expansion principle of the RRT algorithm in a two-dimensional space. Suppose it is all the leaf nodes of the current random tree. Next, a target point will be randomly selected in the solution space except the leaf nodes, and the final target point will be selected. The probability of B is that the probability of not choosing B is; from all the current leaf nodes of the random tree, select the nearest one, which is called the adjacent node; then grow a distance from the direction to get a new leaf node. During the growth process, it is necessary to determine whether the relevant constraints are met. If they are met, the new node is accepted and added as a leaf node of the random tree; if the constraints are not met,

ISSN: 2414-1895 DOI: 10.6919/ICJE.202112_7(12).0066

the new node expanded this time does not meet the requirements, then Abandon the new node, and re-select random target points. Through such continuous extension and expansion, when the leaf nodes in the random tree can grow to point B or close enough to point B, the construction task of the random tree is considered to be completed. At this time, the leaf node closest to the target position is started. , Search the parent node upwards in turn, you can get a feasible path from the starting position to the target position.

3.2 Effective neighborhood construction

After establishing the node's neighborhood reachable set, combined with the basic principles of RRT, it can be known that the generation of neighboring nodes only needs to be selected from the neighboring reachability set of the current leaf node, and the generation of new leaf nodes only needs to be selected from the neighboring nodes. The neighborhood can be generated centrally, and the search range is greatly reduced. At the same time, in order to establish the state vector of the leaf node to describe the state information of each leaf node, and then arrange the above state vectors into a list according to the order in which the leaf nodes are generated. When the leaf node grows to point B, one can be obtained. The leaf node status list containing the complete trajectory, and the trajectory can be obtained by tracing the source backward from point B according to the list. Specific steps are as follows:

(1) Construction and initialization of the leaf node state table

First, take A as the root node, and randomly select a correction point from the reachable set of A's neighborhood as the initial leaf node. Here, A is regarded as a leaf node, as shown in Figure 4.

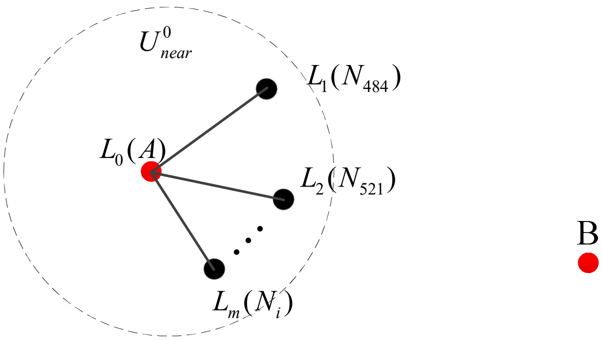

Fig. 3 initial leaf node

4. Conclusion

Aiming at the characteristics of the problem, this paper designs a fast-expanding random tree method based on effective neighborhoods, verifies the effectiveness of the algorithm through numerical simulation, and further analyzes the effectiveness, computational complexity and parameter sensitivity of the algorithm, mainly including The following conclusions.

References

[1] Chen Qiulian, Jiang Huanyu, Zheng Yijun. A review of fast expanding random tree algorithm for robot path planning[J]. Computer Engineering and Applications, 2019, 55(16): 10-17.

[2] Tian Jiang. Three-dimensional constrained multi-target trajectory planning for UAV[D]. Lanzhou University of Technology, 2018.

[3] Zheng Changbo, Li Xiaoyi. Spherical curve in three-dimensional Euclidean space[J]. Journal of University of Shanghai for Science and Technology, 2012, 34(1):56-58.

[4] Qiu Mengluan, Bai Genzhu. Use a single-leaf hyperbolic spiral to smoothly join two straight lines of different planes [J]. Journal of Hubei University for Nationalities (Self Science Edition), 2015, 33(3): 256-259.

[5] Xiao Lin, Bai Genzhu, etc. Smoothly splicing two straight lines from different planes with wavy spirals[J]. Journal of Inner Mongolia University for Nationalities (Natural Chinese Edition), 2016, 31(5):381-384.

[6] Bézier curve, https://en.wikipedia.org/wiki/Bézier_curve, 2019.09.21

[7] Zheng Zongzhun, Wu Hao, Wang Yongji. Research on reentry trajectory optimization based on sequential quadratic programming algorithm[J]. Aerospace Control, 2009, 27(6): 8-13.

[8] Gao Song, Zhang Jinwei, Rong Hui, Wang Wenyang, Guo Peng, He Jia. Application of local obstacle avoidance for unmanned vehicles based on Bezier curve[J]. Modern Electronic Technology, 2019, 42(09): 163-166.