Asymptotically time-optimal smooth trajectory planning in dynamic environments

Hang Zhou
School of Aeronautics and Astronautics
Zhejiang University
Hangzhou, China
zhou_hang@zju.edu.cn

Tao Meng
School of Aeronautics and Astronautics
Zhejiang University
Hangzhou, China
mengtao@zju.edu.cn

***Abstract*—In this paper we proposed an algorithm for smooth trajectory generation in complex environments with dynamic obstacles and velocity constraints. The proposed algorithm Tube Space-Time RRT* (Tube ST-RRT*) is combined with the improved reformulation dynamic coordinate minimum snap (RDCMS) to generate smooth collision-free trajectories with asymptotic time Optimal.First, sample the space-time state space to obtain the time information for each node to complete the avoidance of moving obstacles.Then, to address the issue of non-smooth paths in ST-RRT*, generate a dynamic Tube for each node and combine it with the improved minisnap to create a smooth, collision-free trajectory Finally,Simulations in complex environments demonstrate the effectiveness of our proposed algorithm.**

***Index Terms*—Tube Space-Time RRT*; asymptotic time Optimal; smooth collision-free trajectories**

I. Introduction

Trajectory planning is a fundamental challenge in robotics [1], as obstacles in the real world often change over time. Applications such as robotics and autonomous driving typically require a smooth, collision-free trajectory. Assuming the obstacle trajectories are known a priori, the problem can be modeled as navigation in a dynamic environment. Mathematically, this is expressed as planning through a space-time state space [2].

Planning in dynamic environments has been studied for a long time and has yielded significant research results. These results can be broadly categorized into two approaches. For example, RRTX [3] and Real-time RRT* [4] require fast replanning when previously computed paths become invalid during execution. However, as the dimensionality increases, the replanning time becomes longer, making it difficult to respond to moving obstacles. Risk-RRT [5] combines predictions of obstacle movements and computes partial motion paths to keep the collision probability below a given threshold. However, since only partial paths are returned, frequent replanning is still necessary.Another approach assumes that the trajectories of moving obstacles are unknown, while another assumes that the trajectories of the obstacles are completely known. For example, Time-Based RRT [6] extends the configuration state space through the time dimension and unidirectionally plans to a set of known target states. However, it requires the assumption of knowing the specific time for each target configuration. Additionally, due to the random sampling nature of RRT, the resulting trajectories are usually not smooth.

In this paper, we proposes a Tube-based space-time sampling planning algorithm, Tube-ST-RRT*, and an improved reformulation dynamic coordinate minimum snap (RDCMS) method. Through a two-step method of planning and optimizing, it achieves asymptotic time-optimal collision-free trajectory planning in dynamic environments, while ensuring high trajectory smoothness to facilitate easier tracking control. The Tube-ST-RRT* algorithm adds a time dimension to the configuration space so that each node contains time information, allowing it to respond to time-varying obstacles. Additionally, the improved RDCMS method combines with each node's tube information and time information to generate smooth, collision-free trajectories. Finally, comparative simulations verify the effectiveness of the algorithm

The rest of this paper is organized as follows: At Section II, explains the problem of time-optimal planning in dynamic environments. In Section III,presents the main results of this paper, including the principles and steps of the Tube ST-RRT* planning algorithm and the RDCMS trajectory optimization algorithm. Section IV provides simulation results and comparative experiments to demonstrate the feasibility and superiority of the proposed method. Finally, Section V summarizes the paper.

II. Problem Statement

Consider the time-space planning problem with a tube, where the space-time state space is defined as $\mathcal{Q} = \mathcal{X} \times \mathcal{T} \times \mathcal{B}$, where $\mathcal{X} \subset \mathbb{R}^n$ is the configuration state space, $\mathcal{T}$ is the time state space, $\mathcal{B}$ is the size of the tube, and n is the dimension of the configuration space state. Let $\mathcal{Q}_{free} \subset \mathcal{Q}$ be the collision-free subset of states, Q_{start} be the initial state, and $\mathcal{X}_{goal}$ be the goal region. In the following work, it is assumed that the trajectories of obstacles are completely known. Therefore, the planning problem can be described as finding a path solution that starts from the initial state $q_{start}(t_{start})$, reaches the goal region or target state $x_{goal} \in \mathcal{X}_{goal}$, with the tube size as large as possible and the time taken as short as possible while avoiding time-varying obstacles.

III. Main Results

A. Path Node Generation Based on Tube-ST-RRT*

The TubeST-RRT* algorithm is an improved version of RRT* Connect, modifying the sampling function, steer function, find nearest function and rewiring function, and setting the motion check function for the spacecraft's motion dynamics.Considering the time-varying information of obstacles and the flight corridor with added nodes, we ensure that the generated trajectory is safe and collision-free during subsequent trajectory optimization.

RRT* Connect algorithm is an asymptotically optimal bidirectional sampling method widely used in robot path planning with obstacle avoidance constraints. The core idea of the RRT method is to randomly sample in the state space to obtain a series of path nodes and directed edges from the parent node to the child node, forming a search tree $\mathcal{T}_{tree}$. To address planning problems in environments with time-varying obstacles, the Tube-ST-RRT* algorithm is proposed. This algorithm first adds a time dimension to the state space and samples the node times to obtain nodes that meet the time-varying constraints. Secondly, during the node sampling process, a radius variable is sampled to ensure that the node is collision-free within a spherical region of that radius. The final result is a sequence of nodes $\boldsymbol{s} = [\boldsymbol{s_0}, \boldsymbol{s}_1 \dots, \boldsymbol{s}_n]$, each containing information on position, velocity, time, and the radius of the maximum collision-free spherical region.

The algorithmic details of Tube-ST-RRT* are provided in Algorithms 1. In addition to $\mathcal{S}, q_{start}, \mathcal{X}goal, d$, the planning termination condition ptc, the probability of sampling a new goal $p_{goal} \in (0, 1]$, and the time limit $t_{\max} \in (0, \infty]$ are also required. The basic framework is similar to RRT-Connect.

Algorithm 1 outlines the overall framework of TubeST-RRT*. The general procedure of the algorithm is as follows:

Firstly, it seeks the minimum collision-free r_{init} for the starting point and then initializes parameters. In each iteration, firstly update the boundary parameters. Then, with a probability p_{goal}, decide whether to sample a new target or sample the endpoint. Sample a random state q_{rand}, find q_{near}, and expand q_{new} between q_{near} and q_{rand}. In q_{new}, find the maximum collision-free spherical region. If the path from q_{near} to q_{new} is collision-free and satisfies spacecraft motion dynamic constraints, add the new state x_{new} to the current tree T_a and attempt to connect from x_{new} to the other tree T_b. If the connection is successful, update the solution. Finally, swap T_a and T_b and start the next iteration. Repeat until the termination condition ptc is met.

The main improvements of our proposed TubeST-RRT* algorithm over RRT*-Connect are as follows:

- Generation of collision-free dynamic tubes. Tube-ST-RRT* generates a sequence of intersecting spheres δ_c with radius R_i and time information, as shown in Figure b. From δ_c, a path δ_0 is created, and then boundary points within the sphere intersections are connected to form boundary paths δ_1 and δ_2, as shown in Figure c. This results in a collision-free flight corridor, ensuring that the trajectory generated by minisnap is collision-free.
- Improved Conditional Sampling. Any state that can be part of the solution path must have a finite distance d from the starting point and at least one target state. Due to velocity constraints, only states at the intersection of the start cone and the target cone (see Figure 3) satisfy this requirement. Therefore, similar to informed RRT*, we only sample from the region where solutions can be generated. Ideally, sampling would directly occur from the union of the start velocity cone and the target velocity cone.

Algorithm 1 TubeST-RRT*

Input: $\mathcal{X}, q_{start}, x_{goal}, p_{goal}, range\_d, Param, ptc$
Output: $Soulution$
  $r_{start} \leftarrow FindMaxRadius(s_{start})$
  $s_{start} \leftarrow \{q_{start}, r_{start}\}$
  $T_a \leftarrow \{s_{start}\}; T_b \leftarrow \emptyset$
  $B \leftarrow InitailieBoundVariables(Param)$
  while ptc **do**
    $B \leftarrow UpdateGoalRegion(B, Param, t_{\max})$
    if $p_{goal} > rand(0, 1)$ **then**
      $B \leftarrow SampleGoal(s_{start}, x_{goal}, T_{gaol}, B)$
    end if
    $q_{rand} \leftarrow SampleConditionally(s_{start}, \mathcal{X}, B, d)$
    $r_{rand} \leftarrow FindMaxRadius(q_{rand})$
    $s_{rand} \leftarrow \{q_{rand}, r_{rand}\}$
    $s_{nearsst} \leftarrow Nearset(s_{rand}, T_a)$
    $s_{new} \leftarrow TubeSTSteer(s_{nearest}, s_{rand}, d)$
    if $MotionCheck(s_{nearsst}, s_{new})$ **then**
      $B.samplesInBatch+=1$
      $B.totalSamples+=1$
      $RewireTree(T_a, x_{new})$
      if $Connect(T_b, x_{new}, d) = Reached$ **then**
        $solution \leftarrow UpdateSolution(x_{new})$
        $t_{\max} \leftarrow CostPath(solution)$
        $B.batchProbability \leftarrow 1$
        $PruneTrees(t_{\max}, T_a, T_b)$
      end if
    end if
    $Swap(T_a, T_b)$
  end while
  return $Solution$

B. Reformulation Dynamic Coordinate Minimum Snap Trajectory Optimization

IV. Simulations

V. Conclusion

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]"

or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first . . .”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] B. Siciliano and O. Khatib, Springer handbook of robotics. Springer, 2016.

[2] F. Grothe, V. N. Hartmann, A. Orthey and M. Toussaint, “ST-RRT*: Asymptotically-Optimal Bidirectional Motion Planning through Space-Time,” 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 2022, pp. 3314-3320

[3] M. Otte and E. Frazzoli, “Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning,” International Journal of Robotics Research, vol. 35, no. 7, pp. 797–822, 2016.

[4] K. Naderi, J. Rajamäki, and P. Hämäläinen, “Rt-rrt* a real-time path planning algorithm based on rrt,” in ACM SIGGRAPH Conference on Motion in Games, 2015, pp. 113–118.

[5] C. Fulgenzi, A. Spalanzani, C. Laugier, and C. Tay, “Risk based motion planning and navigation in uncertain dynamic environment,” Research Report, Oct. 2010.

[6] A. Sintov and A. Shapiro, “Time-based rrt algorithm for rendezvous planning of two dynamic systems,” in Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2014, pp. 6745–6750.

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, arXiv:1312.6114. [Online]. Available: https://arxiv.org/abs/1312.6114

[9] S. Liu, “Wi-Fi Energy Detection Testbed (12MTC),” 2023, gitHub repository. [Online]. Available: https://github.com/liustone99/Wi-Fi-Energy-Detection-Testbed-12MTC

[10] “Treatment episode data set: discharges (TEDS-D): concatenated, 2006 to 2009.” U.S. Department of Health and Human Services, Substance Abuse and Mental Health Services Administration, Office of Applied Studies, August, 2013, DOI:10.3886/ICPSR30122.v2

[11] K. Eves and J. Valasek, “Adaptive control for singularly perturbed systems examples,” Code Ocean, Aug. 2023. [Online]. Available: https://codeocean.com/capsule/4989235/tree