路径规划和轨迹优化实践报告

周克涵 蒋益斌 贾文骏

**摘 要** 在修习课程《智能移动机器人技术》期间，我们以小组为单位开展避障路径生成及轨迹优化仿真实验，并形成实践报告。本文首先介绍了路径规划和轨迹优化的定义内涵与背景意义，并对现有研究现状进行调研、综述。其后，本文重点介绍了小组借助ROS框架在Rviz平台中实现路径规划和轨迹优化的方法、改进思路与过程，并将实验结果进行展示、对比与分析。

**关 键 词** 智能移动技术, 路径规划, 轨迹优化, ROS平台, A\*算法, 多项式规划方法

**代码网址** https://github.com/Khansakura/ZJU\_TrajectoryHomework

# 背景意义

随着人工智能、自动控制和计算机技术的快速发展，机器人和自动化系统的应用逐渐深入到各个领域，如自动驾驶汽车、无人机物流配送、工业机器人自动化生产线、无人探测器任务执行等。在这些场景中，路径规划和轨迹优化是智能系统实现高效、安全运行的关键技术。

## 路径规划

路径规划是指在已知环境中，从起点到终点寻找一条最优路径的过程。在复杂的现实环境中，路径规划需要综合考虑以下几个因素[1]：

1.障碍物回避：路径需要避开所有静态或动态障碍物，确保机器人能够安全到达目标点。

2.路径优化：生成的路径应尽可能短，以提高任务执行效率，减少时间和资源的消耗。

3.实时性：在动态环境中，路径规划需要快速响应环境变化，例如应对新出现的障碍物或目标点位置的变化。

4.复杂环境的规划：现代路径规划技术已从最初的简单场景逐渐扩展到复杂的多约束、多目标环境。例如，自动驾驶车辆在城市交通中，需要在动态交通流中实时避开行人、车辆等障碍，同时遵守交通规则；无人机在执行侦察或物流任务时，需要在复杂的三维环境中规划飞行路径，避开地形和其他飞行物体。

## 轨迹优化

在完成路径规划后，轨迹优化则需要进一步提升路径的质量，同时进行运动规划，以保证目标物可以按照规划好的路径进行移动。轨迹优化主要关注路径的平滑性、满足目标物的动态约束以及能量效率的提高等[2]。

平滑性：轨迹平滑性是指路径中速度和加速度的变化应尽量平缓，以避免运动的不稳定性，保证机器人能够按照路径平稳移动至目标点，同时减少对机器人的磨损。

动态约束：轨迹优化需要考虑目标物的姿态、速度、加速度、惯性等动态约束，考虑实际移动场景。轨迹优化需要确保机器人能够在这些约束条件下按照规划路径进行移动。

能量效率：在移动机器人如无人机等设备的运行中，能量消耗直接影响其续航能力。对规划好的路径进行轨迹优化可以有效降低能耗，提高系统的经济性和实用性。

轨迹优化的核心目标是保证机器人或车辆在按照规划好的路径进行移动，在此目标的基础上，进一步提高运动的流畅度和高效性。

## 挑战与意义

路径规划与轨迹优化面临的挑战在于环境的不确定性、动态变化和高维复杂性。例如，自动驾驶汽车需要实时处理大量传感器数据，同时快速计算出避障路径并优化驾驶轨迹；而机器人在仓库或工厂环境中，常常需要应对密集的动态障碍物，同时考虑路径的灵活性和轨迹的精确性。

理论意义：深入研究避障路径生成与轨迹优化，有助于推动机器人学、自动控制理论以及人工智能算法的进一步发展。路径规划与优化问题本质上是多目标、多约束的复杂优化问题，其研究成果可以应用于更广泛的学术领域，如数学优化、控制理论、计算几何等。

实际意义：高效的路径规划和轨迹优化算法在实际应用中具有重要价值。例如，在物流配送中，无人机能够更加快速且精确地完成包裹投递；在工业自动化中，机器人可以在复杂的动态生产线中更加高效地工作。此外，这些技术在国防、救援、医疗等领域也有着广阔的应用前景。例如，灾难救援机器人需要在复杂地形中执行任务，而精准的路径规划与轨迹优化能够显著提高其救援效率和安全性。

## 发展趋势与研究热点

随着计算硬件性能的不断提升和人工智能算法的快速进步，避障路径生成与轨迹优化领域呈现以下发展趋势[3]：

融合人工智能技术：结合深度学习和强化学习的方法，能够在未知或部分已知环境中实现路径规划和轨迹优化的智能化。

实时性与动态适应性：研发更高效的算法，实现路径规划与轨迹优化的实时响应能力。

多机器人协作：针对多机器人系统的路径生成与轨迹优化问题，研究多智能体协作规划算法，提升群体效率。

复杂环境中的应用：探索在高维、动态、复杂环境中的规划方法，例如应用于无人驾驶的城市交通、智能农业机器人在不规则农田中的作业规划等。

综上所述，避障路径生成与轨迹优化是连接理论研究与实际应用的桥梁。通过研究高效、安全、实时的路径规划与轨迹优化方法，可以推动智能系统在更多复杂场景中的应用，从而进一步提升自动化系统的性能和价值。

# 研究现状综述

避障路径生成与轨迹优化是自动化领域中的关键技术，广泛应用于自动驾驶、机器人导航、无人机飞行等多个领域。近年来，随着智能系统的不断发展，研究者们针对避障路径生成和轨迹优化提出了多种方法，尤其是在动态复杂环境[4]下的应用，成为了研究的热点。

## 避障路径生成方法

避障路径生成的研究最早集中在静态环境中，采用基于图的搜索算法，如A\*算法和Dijkstra算法来寻找从起点到终点的最短路径。在静态环境中，路径规划问题较为简单，通常通过构建图模型并搜索路径即可实现。然而，随着应用场景的扩展，动态障碍物的出现使得路径生成问题变得更加复杂。

在动态环境中，许多研究开始关注如何在移动障碍物的影响下进行避障路径规划。基于采样的算法（如Rapidly-exploring Random Tree, RRT）成为了该领域的主流方法之一。RRT算法[5]通过随机采样扩展树状结构，能够在高维空间中有效地找到一条可行的路径，并且具备较好的计算效率。然而，传统RRT算法存在路径质量不高、计算复杂度较大等问题。为了解决这些问题，RRT算法的改进版本应运而生，其中RRT\*算法引入了路径优化机制，可以逐步改进路径质量，使得生成的路径更加平滑、更接近最优。

另外，针对动态环境中的路径生成，RRT-Connect算法[6]被提出，该算法通过同时从起点和终点两个方向进行树的扩展，能够加速路径的搜索过程，尤其在较大搜索空间中表现优异。然而，RRT系列算法普遍存在路径规划过程中的局部最优问题，且计算复杂度较高。因此，一些学者提出了基于遗传算法、蚁群算法等启发式算法的路径生成方法，这些方法通过模拟自然界的启发式搜索，能够找到更加全局最优的避障路径。

## 轨迹优化方法

在路径生成的基础上，轨迹优化需要着重提升路径的可行性，特别是在实际执行过程中，如何使路径更加平滑、符合机器人的动态约束、减少能量消耗成为轨迹优化的核心问题。轨迹优化的研究主要集中在以下几种方法：

1.基于优化的路径平滑技术：这种方法通过对路径进行数学优化，消除路径中的尖锐转折或冗余路径，使得路径更加平滑，从而使路径更适合进行实际移动。常见的平滑方法包括基于二次规划的最小化路径曲率的算法等[7]。平滑方法能够有效改善路径的平滑性，但对于非线性约束和高维动态系统的适应能力较差。

2.最优控制方法：最优控制方法考虑了动态系统的控制约束，例如速度、加速度等，常见的方法有线性二次调节器（LQR）和非线性最优控制（如Pontryagin最大值原理）[8]。通过对轨迹进行优化，最优控制可以确保在满足约束条件的前提下，使得路径的执行更加平稳且节能。不过，这些方法通常需要较高的计算资源，且在动态环境下的实时性表现较差。

3.采样与优化结合的方法：近年来，随着RRT\*算法的改进，越来越多的研究将路径生成与轨迹优化相结合。通过引入优化算法，如基于梯度下降的优化方法或基于模拟退火的全局优化方法，可以对RRT生成的路径进行进一步的优化。这些方法不仅能够提升路径质量，还能降低能量消耗，尤其适用于需要高度实时响应的应用。

4.在线优化与自适应控制[9]：在动态环境下，障碍物和环境条件的变化使得轨迹优化的实时性和自适应性成为了新的挑战。近年来，研究者们提出了结合在线优化与自适应控制的策略，这些方法能够在运行过程中实时调整轨迹，以应对环境的变化。例如，采用预测模型和自适应控制技术，能够动态地调整机器人的控制策略，确保轨迹优化过程的实时性和适应性。

## 现有研究的挑战与不足

尽管目前的路径生成与轨迹优化方法已取得了一定的成果，但在实际应用中仍面临不少挑战：

计算复杂度与实时性问题：尽管RRT\*算法和轨迹优化方法在路径质量和执行效率上取得了一定进展，但在复杂环境中，尤其是动态障碍物密集的情况下，算法的计算效率仍然是一个瓶颈。大规模环境中路径生成和轨迹优化的计算复杂度较高，导致难以满足实时性要求。

动态约束与环境适应性[10]：大多数现有的轨迹优化方法主要考虑了静态约束，难以在动态环境中快速调整路径。如何在保证路径平滑性和最优性的同时，提高系统的自适应能力，依然是当前研究的难点。

路径与轨迹的平衡：路径生成与轨迹优化之间的平衡问题依然存在。尽管路径生成可以保证避障，但优化后的轨迹可能会牺牲路径的简洁性；反之，优化路径可能会导致轨迹不符合动态约束。因此，如何在两者之间找到一个合理的平衡点，成为未来研究的一个方向。

## 总结

综上所述，避障路径生成和轨迹优化技术已取得显著进展，但在动态环境下仍面临实时性、计算复杂度、轨迹平滑性和能量优化等方面的挑战。未来的研究需要在算法的实时性、动态适应性及路径与轨迹的优化平衡方面进行深入探讨，以应对更加复杂的应用场景。

# 方法介绍

## 路径规划基础功能实现方法

实现基础功能部分的代码文件名为astar\_planner\_base.cpp，该代码实现了基于A\*算法的路径规划器，使用了Robot Operating System（下简称ROS）框架。具体实现步骤如下：

1.为了进行路径规划，我们需要确定所有障碍物的位置。我们首先将世界坐标转换为网格坐标。但由于世界坐标下障碍物是圆形的，而网格是方形的，这种形状的差异可能导致误差，从而使规划的路径与障碍物冲突。因此在编写代码时，相比障碍物原有半径，我们额外增加了一定的裕度，以确保路径不会与障碍物发生冲突。

2.接下来，我们获取网格半径（grid\_radius），遍历网格坐标，期间计算当前位置到每个圆形障碍物圆心的距离。如果该距离小于障碍物的半径，则判定该网格为“占用”，即该网格为障碍物网格。通过这些步骤，我们可以确定所有存在障碍物的网格，

在完成以上步骤之后，我们获得了出发点、目标点以及地图中所有障碍物的位置范围，之后便再采用A\*算法进行路径规划。

以下是该代码中各个主要部分的结构以及功能概述：

1.Node结构体：表示路径中的一个节点，包括位置坐标（x, y），从起点到该节点的代价（g\_cost），从该节点到终点的估计代价（h\_cost），以及指向父节点的指针（parent）。

2.GridMap类：定义了一个网格地图，存储了地图的宽度、高度、分辨率以及每个网格的状态（空闲或占用），该类包含标记障碍物的方法。

3.AStarPlanner类：实现了A\*路径规划的逻辑，包含设置障碍物、打印网格地图、寻找路径等功能。该类使用了优先队列来管理开放列表，并根据启发式算法来评估路径的优劣。

4.代码主要逻辑：在findPath方法中，程序将起点和终点转换为网格坐标，并使用A\*算法搜索路径；如果找到路径，则调用reconstructPath方法回溯生成实际路径。

5.主函数部分：用于初始化ROS节点，并设置相关参数，订阅障碍物信息并通过发布路径消息来进行可视化。

实现基础功能部分的代码基于A\*算法实现了路径规划，能够在ROS环境中使用，可以根据输入的起点、终点和障碍物信息，计算出一个有效、可行的路径。

基础功能的代码虽然能够实现任务目标，但仍然存在一定不足，如：路径规划用时较长、适用场景较为单一、工作量较为繁冗等，仍有改进、提升的空间。

## 路径规划改进与优化方法

针对实现基础功能的代码中存在的一些不足之处，我们做出了相应改进，得到改进代码，其中改进之处总结如下：

1.双向搜索（Bidirectional Search）：

改进代码实现了双向A\*搜索，即从起点和终点同时进行搜索，直到它们相遇，从而可能减少搜索的时间复杂度。基础代码则只进行了单向搜索，从起点到终点。

2.启发式函数的多样化：

相比基础代码仅使用欧几里得距离作为启发式函数，改进代码引入了多种启发式距离计算方法，包括欧几里得距离、曼哈顿距离和切比雪夫距离，并通过权重组合这些启发式函数，使得路径规划可以根据不同场景进行调整和优化。

3.路径重构的优化：

改进代码中，我们增设了reconstruct\_ Bidirectional\_Path函数，能够在找到相遇点后，分别从起点和终点回溯路径并合并。该功能相比于基础代码的单向回溯更加高效、直观。

4.搜索过程中的节点访问记录：

相比基础代码，改进代码通过visited\_from\_start和visited\_from\_goal记录从起点和终点搜索过程中访问的节点，这使得在搜索时可以更快判断是否遇到相同节点，减少重复工作，从而加快搜索过程。

5.路径长度统计：

改进代码新增了路径长度统计功能：在重构路径时，计算了路径的总长度并打印出来，从而能够提供对路径的定量评估。

6.代码结构与可读性：

改进代码的结构较为清晰，功能划分明确，尤其是在路径重构和距离计算方面。基础代码相对简单，但在处理复杂场景时的可扩展性可能不足。

7.逻辑处理的鲁棒性：

改进代码在处理节点时鲁棒性更强，确保每一步都检查是否已经在关闭列表中，从而减少不必要的计算。这一点在基础代码中也有体现，但改进代码通过双向搜索的方式进一步增强了这一点。

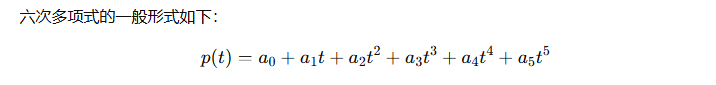
总体而言，改进代码在路径规划的效率和灵活性方面做出了显著改进，特别是在应对复杂环境和动态障碍物时表现得更为优越。

## 路径to轨迹优化方法

路径to轨迹优化的功能模块的主要工作内容是生成从一个路径点到另一个路径点的轨迹。在该功能模块，我们分别使用了多项式方法和贝塞尔曲线方法。

### 3.3.1 多项式方法

首先，我们使用了六次多项式方法。多项式方法需要通过边界条件（即起始位置、起始速度、起始加速度、结束位置、结束速度和结束加速度），计算出多项式的系数；然后，基于这些系数，可以在时间步长内计算路径上各个点的坐标。六次多项式的一般形式如下：



在名为trajectory\_generator.cpp的代码文件中，我们实现了基于六次多项式的路径to轨迹优化。

### 3.3.2 贝塞尔曲线方法

在使用多项式方法之后，我们又尝试了使用贝塞尔曲线方法。

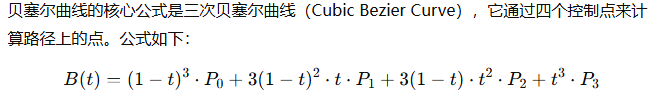
贝塞尔曲线的核心公式是三次贝塞尔曲线(Cubic Bezier Curve)，它通过四个控制点来计算路径上的点。

对于每一段路径（start到end），生成两个控制点：control1和control2。

control1 = start + 0.4 \* (end - start) 表示第一个控制点偏离起始点 40% 的距离。

control2 = end - 0.4 \* (end - start) 表示第二个控制点偏离结束点 40% 的距离。

将起点，控制点1，控制点2，终点共计四个控制点代入公式求解，最后可以在时间步长内计算路径上各个点的坐标。其中，公式的一般形式如下:



三阶贝塞尔曲线是最为常见的类型，它允许更灵活的路径形状，能够提供平滑的过渡和弯曲效果。

在名为trajectory\_generator2.cpp的代码文件中，我们实现了基于三阶贝塞尔曲线的路径to轨迹优化。

# 实验结果及分析（找了差距更明显的图）

## A\*算法的无碰撞路径图

我们修改了场景中的障碍物数量，由初始的10个上升为20个，更能体现出路径规划的效果。基础代码和改进代码的效果图及相关性能图如下：

### 4.1.1 A\*算法基础代码效果展示dff76cf345c84bf6d6e9ce500fe61cc

图4.1 A\*算法基础代码无碰撞路径图

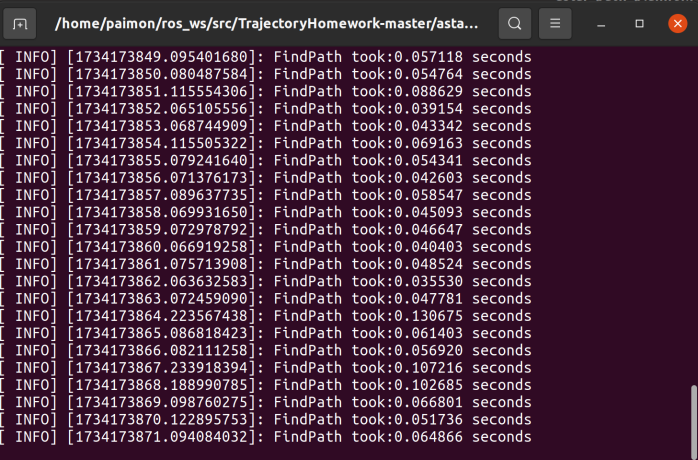


图4.2 基础代码寻路时间

### 4.1.2 A\*算法改进代码效果展示

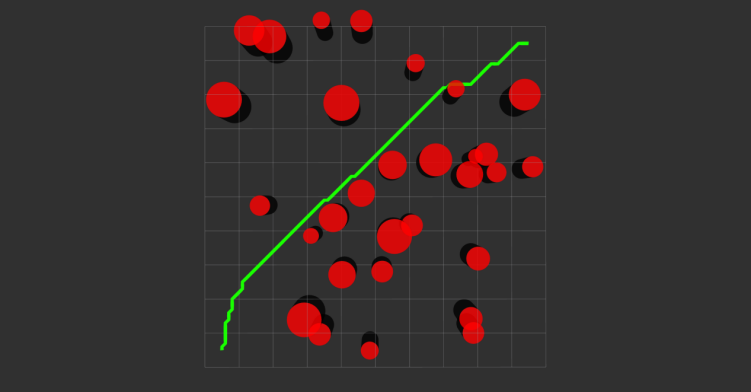


图4.3 A\*算法改进代码无碰撞路径图

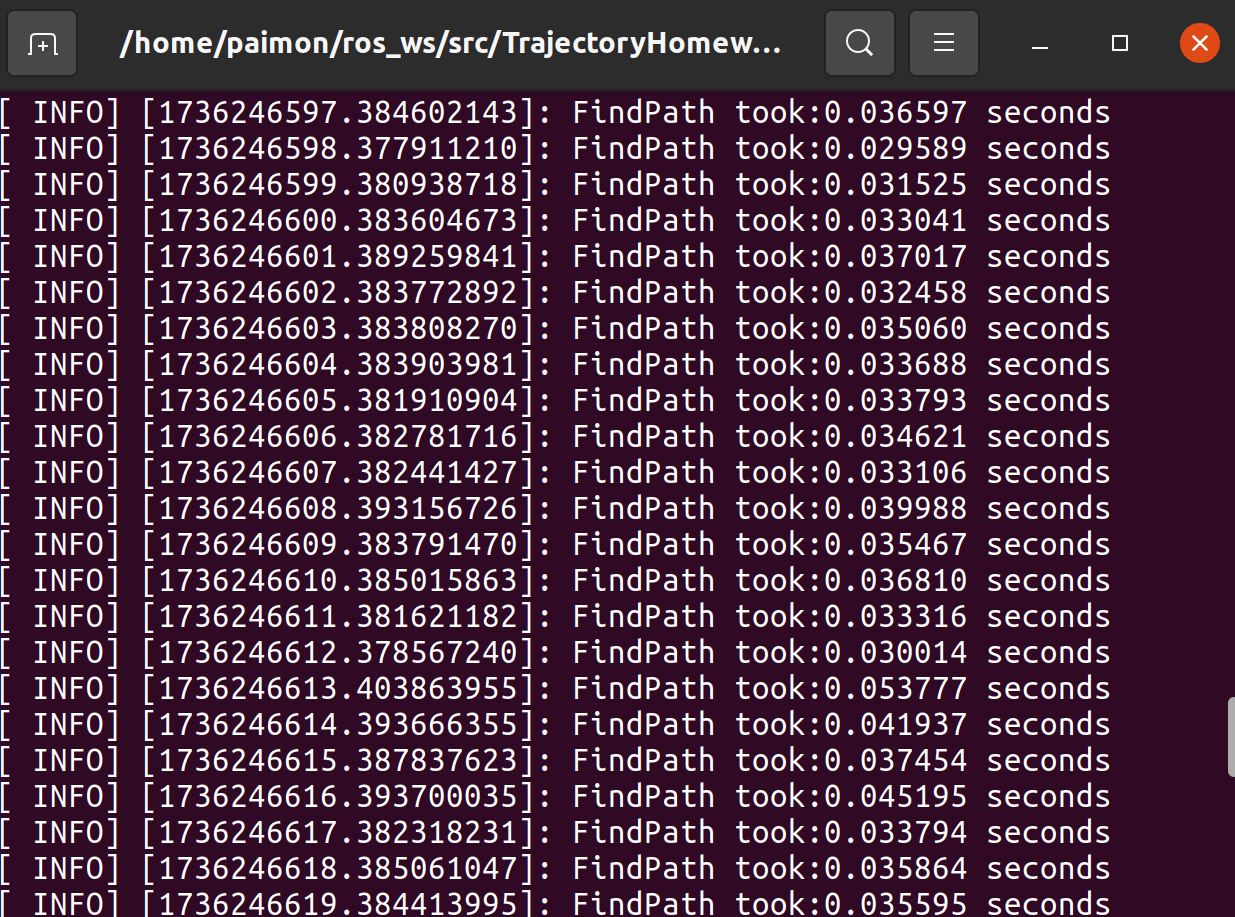


图4.4 改进代码寻路时间

### 4.1.3结果分析

通过Rviz中的效果图可以看出，A\*算法的基础代码和改进代码都能够实现任务目标，能够找到对应的无碰撞路径。相比基础代码，改进代码寻找可行路径所花费的时间更少，路径规划的平均速度更快，可见性能有所提升。

## 路径to轨迹优化结果与分析

### 4.2.1 多项式方法效果展示与分析

轨迹图如下图所示：

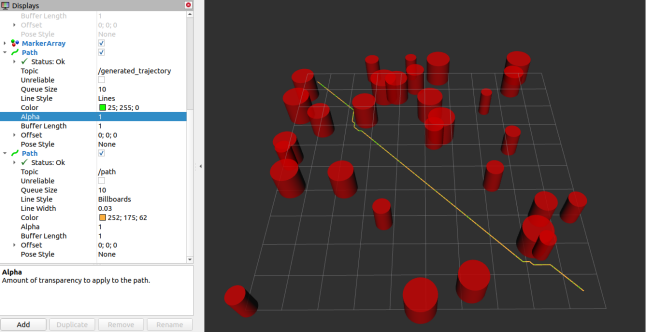


图4.5 六次多项式轨迹示意图（黄线）

其中绿线为A\*算法生成的路径，黄线为使用六次多项式轨迹函数规划得到的轨迹，两者基本吻合。

参数图如下图所示：

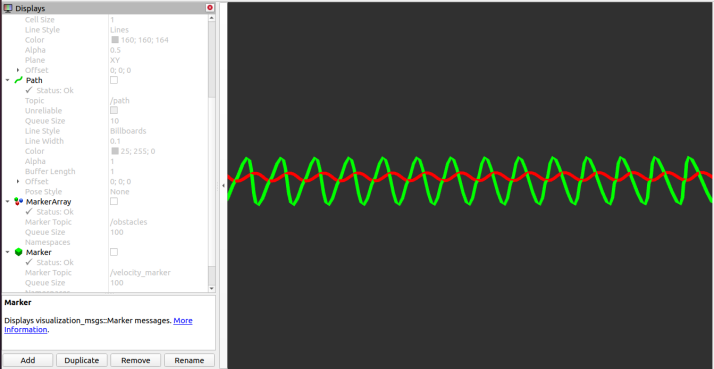


图4.6 六次多项式轨迹的速度与加速度

速度和加速度曲线，红线为速度，绿线为加速度。可以看出速度与加速度均为连续变化，因此通过优化得到的轨迹具有运动的可行性。

### 4.2.2 贝塞尔曲线方法效果展示

轨迹图如下图所示：

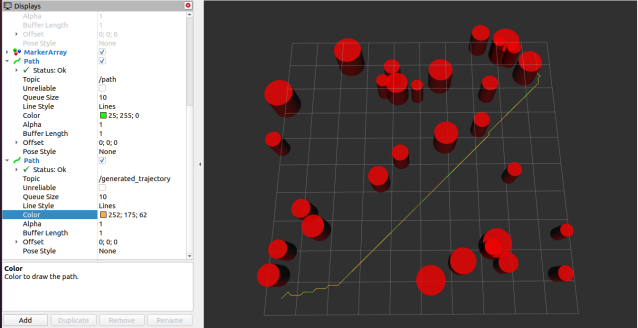


图4.7 三阶贝塞尔曲线轨迹示意图（黄线）

其中绿线为A\*算法生成的路径，黄线为使用三阶贝塞尔曲线轨迹函数规划得到的轨迹，两者同样基本吻合。

参数图如下图所示：

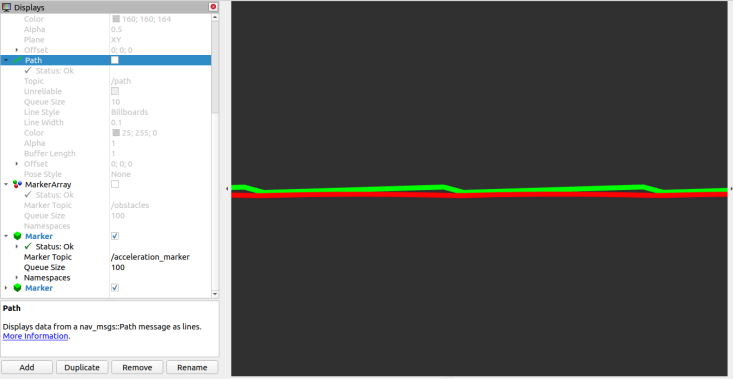


图4.8 三阶贝塞尔曲线轨迹的速度与加速度

速度和加速度曲线，红线为速度，绿线为加速度。可以看出速度与加速度均为连续变化，因此通过优化得到的轨迹具有运动的可行性。

# 参考文献列表

孙伟昌, 罗志浩, 石建迈等. 无人机覆盖路径规划方法综述[J/OL]. 控制理论与应用, 1-21 [2024-12-12].

张楠, 宝音贺西, 崔海英. 多目标探测混合整数轨迹优化方法研究进展[J/OL]. 飞控与探测, 1-16 [2024-12-12].

王小慧. 基于人工智能技术的3D校园漫游导航系统设计与应用[J]. 信息记录材料, 2024, 25(12): 77-79.

Z. Wang, X. Zhou, C. Xu and F. Gao, "Geometrically Constrained Trajectory Optimization for Multicopters," in IEEE Transactions on Robotics, vol. 38, no. 5, pp. 3259-3278, Oct. 2022.

W. Zeng, Y. Xie, S. Wang, H. Wu and T. Xiong, "Curvature-Continuous RRT-Based Path Planning with Enhanced Efficiency," 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 2023, pp. 1-6.

H. Yang, H. Li, K. Liu, W. Yu and X. Li, "Research on path planning based on improved RRT-Connect algorithm," 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 2021, pp. 5707-5712.

李炯逸, 李强, 张新闻等. 移动机器人用改进的双向A\*二次路径规划算法[J/OL]. 系统仿真学报, 1-11 [2024-12-12].

N. T. Tran, M. H. Tran, T. N. Phi Nguyen and D. T. Tran, "Modelling and Experimental Analysis for a two legged Wheel Robot with a Fuzzy LQR Control," 2024 7th International Conference on Green Technology and Sustainable Development (GTSD), Ho Chi Minh City, Vietnam, 2024, pp. 340-344.

眭耀宇. 四足机器人运动控制方法研究[D]. 南京航空航天大学, 2022.

C. Zhang and J. Fu, "Multi-objective Optimization of Path-Constrained Switched Systems Using the Modified ε-Constraint Method," 2021 China Automation Congress (CAC), Beijing, China, 2021, pp. 2898-2901.

# 附：小组分工

周克涵：路径规划功能模块。

蒋益斌：轨迹优化功能模块。

贾文骏：资料查找与报告撰写。