

同行专家业内评价意见书编号：20250854373

附件1

浙江工程师学院（浙江大学工程师学院） 同行专家业内评价意见书

姓名：袁嘉喆

学号：22260011

申报工程师职称专业类别（领域）：电子信息

浙江工程师学院（浙江大学工程师学院）制

2025年03月19日

填表说明

一、本报告中相关的技术或数据如涉及知识产权保护、军工项目保密等内容，请作脱密处理。

二、请用宋体小四字号撰写本报告，可另行附页或增加页数，A4纸双面打印。

三、表中所涉及的签名都必须用蓝、黑色墨水笔，亲笔签名或签字章，不可以打印代替。

四、同行专家业内评价意见书编号由工程师学院填写，编号规则为：年份4位+申报工程师职称专业类别(领域)4位+流水号3位，共11位。

一、个人申报

(一) 基本情况【围绕《浙江工程师学院（浙江大学工程师学院）工程类专业学位研究生工程师职称评审参考指标》，结合该专业类别(领域)工程师职称评审相关标准，举例说明】

1. 对本专业基础理论知识和专业技术知识掌握情况(不少于200字)

在知识掌握层面，首先作为一个在企业实践的学生，知识的掌握对于科研工作至关重要。在企业的培养环境下，我得以接触到更为系统和深入的专业知识，同时也能通过实际的科研项目将这些知识运用到实践中。

在企业的培养过程中，我首先巩固了大学期间学到的基础知识。例如，在计算机科学领域，我通过参与研究所的基础课程复习了数据结构与算法、操作系统原理、计算机网络等基础知识。这些知识是进行高层次科研工作的基石。

同时，在掌握了基础知识后，我深入学习了与我研究方向相关的专业知识。作为学习无人机的学生，项目初期，我参与了无人机项目的需求分析和可行性研究。这一阶段让我学会了如何从实际需求出发，制定详细的项目计划和实施方案。在企业的指导下，我了解了无人机应用的广泛领域，如农业监测、环境保护、物流配送等，这些都为后续的研究提供了明确的方向。为我后续进行相关的研究提供了相应的方向。

2. 工程实践的经历(不少于200字)

在杭州兵智科技有限公司的实习期间，我深度参与了无人机自主规划控制和自主探索项目的开发。我的主要职责包括算法设计和实物部署实施，这让我有机会将理论知识应用到实际工程问题中。

在算法设计阶段，我研究并实现了适用于无人机的路径规划和环境感知算法，主要是基于三维动力学方程建模，建立优化问题进行求解。我们先使用A*算法搜索出当前的部分路径，然后基于优化求解器算法求解出最优的局部避障轨迹。

对于无人机自主探索，我采用了基于前沿点的方案，使用旅行商问题进行相关求解，得到探索的优化顺序。在多无人机协同探索方面，我实现了多旅行商问题算法，通过这一算法进行多机任务分配，成功实现了多无人机的协同自主探索任务。

实物部署环节充满挑战，我负责将算法集成到无人机的机载处理器系统中，解决了通信延迟、传感器噪声等实际问题。

3. 在实际工作中综合运用所学知识解决复杂工程问题的案例(不少于1000字)

在杭州兵智科技有限公司实习期间，我参与了一个复杂的无人机集群自主探索与控制项目。该项目旨在开发一套系统，使多架无人机能够在未知环境中协同工作，自主规划路径，避开障碍物，并高效地完成区域探索任务。这个项目涉及多个工程学科的知识，包括控制理论、任务分配、多智能体协同等领域的综合应用。

项目中我们面临几个关键挑战：无人机需要在三维空间中实时规划安全路径，同时考虑动力学约束；如何高效分配探索任务，避免重复工作，最大化探索效率；算法必须在计算资源有限的嵌入式系统上运行，并满足实时性要求；以及如何处理传感器噪声、通信延迟等实际问题带来的系统稳定性挑战。

针对三维环境下的路径规划问题，我综合应用了多门课程的理论知识。从高等数学和线性代数中，我运用向量分析和矩阵变换构建了精确的三维空间模型，并基于经典力学原理建立了无人机的完整动力学约束方程，包括速度、加速度和角速度限制。在全局路径规划层面，我应用了计算机科学中的启发式搜索算法，选择了A*算法作为基础框架，通过设计适合三维环境的启发函数，在可接受的计算复杂度下生成了全局路径骨架。为优化飞行轨迹的平滑性和可执行性，我构建了一个非线性优化模型，其目标函数以时间最小化为主，同时纳入了避障

安全距离、动力学可行性等多重约束条件。我选择内点法作为求解工具，通过迭代优化将A*生成的分段路径转化为满足动力学约束的平滑轨迹，实现了理论与实践的有效结合。这种分层规划方法既保证了全局路径的最优性，又确保了局部轨迹的可执行性，为无人机在复杂环境中的自主导航提供了可靠保障。

在未知环境中的多机自主探索是本项目的核心难点。我综合运用了运筹学、图论等学科理论知识来解决这一复杂问题。首先是将未知环境表示为概率占据栅格地图，并设计了一种高效的前沿点识别算法。该算法能够在三维空间中快速识别出探索边界上的高信息增益区域，作为潜在的探索目标点。然后，针对这些目标点，使用TSP算法确定到达这些目标点的顺序，从而得到了高效的单机探索方法。针对多机协同探索问题，我将问题进一步扩展为多旅行商问题(MTSP)，并结合任务分配的思想，对于每架无人机的任务进行分配，进而完成了高效的集群自主探索方法。

在系统测试阶段，我面临了传感器噪声和环境不确定性带来的挑战。为解决无人机实时定位问题，我采用了视觉惯性里程计(VIO)技术，具体实现上选择了英特尔RealSense T265跟踪相机作为定位方案的核心。T265集成了双目鱼眼相机和IMU传感器，内置ASIC处理芯片可直接输出六自由度姿态估计，大大简化了系统集成复杂度。我负责将T265模块集成到无人机平台上，优化了安装位置以减少振动干扰，并开发了程序将T265的位姿数据转换为无人机控制系统所需的坐标系格式。

项目实施过程中，我也遇到了一些意料之外的问题。例如，在多机实验中，我们发现了算法理论上的正确性与实际系统中的可靠性之间存在差距。针对这个问题，我采用了系统化的测试方法，包括单元测试、集成测试和场景模拟，识别了关键算法模块在实际环境中的边界情况并针对性地进行了优化。

通过这个项目，我深刻体会到了理论知识与工程实践的紧密联系。对于每个技术难点，我都需要回顾并整合多门课程的知识，有时甚至需要查阅最新的研究论文来寻找解决方案。这种综合应用知识解决复杂工程问题的能力，是我在学校课程中难以获得的宝贵经验。

(二) 取得的业绩(代表作)【限填3项, 须提交证明原件(包括发表的论文、出版的著作、专利证书、获奖证书、科技项目立项文件或合同、企业证明等)供核实, 并提供复印件一份】

1. 公开成果代表作【论文发表、专利成果、软件著作权、标准规范与行业工法制定、著作编写、科技成果获奖、学位论文等】

成果名称	成果类别 [含论文、授权专利(含发明专利申请)、软件著作权、标准、工法、著作、获奖、学位论文等]	发表时间/授权或申请时间等	刊物名称/专利授权或申请号等	本人排名/总人数	备注
基于未知环境下四旋翼飞行器时间最优避障在线规划方法	发明专利申请	2024年07月01日	申请号: 202410870930.6	1/2	
Safety-critical Online Quadrotor Trajectory Planner for Agile Flights in Unknown Environments	会议论文	2025年01月28日	International Conference on Robotics and Automation	1/5	已接受, 未收录

2. 其他代表作【主持或参与的课题研究项目、科技成果应用转化推广、企业技术难题解决方案、自主研发设计的产品或样机、技术报告、设计图纸、软课题研究报告、可行性研究报告、规划设计方案、施工或调试报告、工程实验、技术培训教材、推动行业发展中发挥的作用及取得的经济社会效益等】

(三) 在校期间课程、专业实践训练及学位论文相关情况	
课程成绩情况	按课程学分核算的平均成绩： 86 分
专业实践训练时间及考核情况(具有三年及以上工作经历的不作要求)	累计时间： 1 年(要求1年及以上) 考核成绩： 85 分
本人承诺	
<p>个人声明：本人上述所填资料均为真实有效，如有虚假，愿承担一切责任，特此声明！</p> <p style="text-align: right;">申报人签名：袁嘉喆</p>	

浙江大学研究生院
攻读硕士学位研究生成绩单

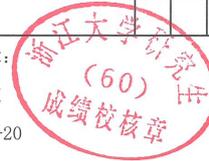
学号: 22260011	姓名: 袁嘉喆	性别: 男	学院: 工程师学院	专业: 电子信息	学制: 2.5年						
毕业时最低应获: 26.0学分	已获得: 29.0学分	入学年月: 2022-09	毕业年月:								
学位证书号:	毕业证书号:	授予学位:									
学习时间	课程名称	备注	学分	成绩	课程性质	学习时间	课程名称	备注	学分	成绩	课程性质
2022-2023学年秋季学期	数值计算方法		2.0	96	专业选修课	2022-2023学年春季学期	自然辩证法概论		1.0	87	专业学位课
2022-2023学年秋季学期	工程技术创新前沿		1.5	89	专业学位课	2022-2023学年春季学期	研究生英语基础技能		1.0	70	公共学位课
2022-2023学年秋季学期	新时代中国特色社会主义思想理论与实践		2.0	92	专业学位课	2022-2023学年春季学期	人工智能制造技术		3.0	79	专业学位课
2022-2023学年秋冬学期	工程伦理		2.0	86	专业学位课	2022-2023学年春季学期	制造物联网技术		2.0	76	专业学位课
2022-2023学年秋冬学期	智能工业机器人及其应用		3.0	89	专业选修课	2022-2023学年春季学期	研究生英语		2.0	89	专业学位课
2022-2023学年冬季学期	模式识别与人工智能		2.0	91	跨专业课	2022-2023学年春季学期	高阶工程认知实践		3.0	82	专业学位课
2022-2023学年秋冬学期	研究生论文写作指导		1.0	90	专业选修课		硕士生读书报告		2.0	通过	
2022-2023学年冬季学期	产业技术发展前沿		1.5	92	专业学位课						

说明: 1. 研究生课程按三种方法计分: 百分制, 两级制 (通过、不通过), 五级制 (优、良、中、及格、不及格)。
2. 备注中 "*" 表示重修课程。

学院成绩校核章:

成绩校核人: 张梦依

打印日期: 2025-03-20





310013

浙江省杭州市西湖区古墩路 701 号紫金广场 B 座 1103 室 杭州求是
专利事务所有限公司
刘静(0571-87911726-809)

发文日:

2024 年 07 月 01 日



申请号: 202410870930.6

发文序号: 2024070101672840

专利申请受理通知书

根据专利法第 28 条及其实施细则第 43 条、第 44 条的规定, 申请人提出的专利申请已由国家知识产权局受理。现将确定的申请号、申请日等信息通知如下:

申请号: 2024108709306

申请日: 2024 年 07 月 01 日

申请人: 浙江大学

发明人: 袁嘉喆, 李硕

发明创造名称: 基于未知环境下四旋翼飞行器时间最优避障在线规划方法

经核实, 国家知识产权局确认收到文件如下:

权利要求书 1 份 4 页, 权利要求项数: 10 项

说明书 1 份 13 页

说明书附图 1 份 2 页

说明书摘要 1 份 1 页

专利代理委托书 1 份 2 页

发明专利请求书 1 份 4 页

实质审查请求书 文件份数: 1 份

申请方案卷号: 刘-241-155-政

提示:

1. 申请人收到专利申请受理通知书之后, 认为其记载的内容与申请人所提交的相应内容不一致时, 可以向国家知识产权局请求更正。

2. 申请人收到专利申请受理通知书之后, 再向国家知识产权局办理各种手续时, 均应当准确、清晰地写明申请号。

审查员: 自动受理

联系电话: 010-62356655

审查部门: 初审及流程管理部





国家知识产权局

310013

浙江省杭州市西湖区古墩路 701 号紫金广场 B 座 1103 室 杭州求是
专利事务所有限公司
刘静(0571-87911726-809)

发文日:

2024 年 09 月 10 日



申请号或专利号: 202410870930.6

发文序号: 2024091000248020

申请人或专利权人: 浙江大学

发明创造名称: 基于未知环境下四旋翼飞行器时间最优避障在线规划方法

发明专利申请公布通知书

上述专利申请, 经初步审查, 符合专利法实施细则第 50 条的规定。根据专利法第 34 条的规定, 该申请在 40 卷 3701 期 2024 年 09 月 10 日专利公报上予以公布。

提示:

1. 发明专利申请人可以自申请日起 3 年内提交实质审查请求书、缴纳实质审查费, 申请人期满未提交实质审查请求书或期满未足额缴纳实质审查费的, 该申请被视为撤回。

2. 专利费用可以通过网上缴费、银行/邮局汇款、直接向代办处或国家知识产权局专利局缴纳。缴费时应当写明正确的申请号/专利号、费用名称及分项金额, 未提供上述信息的视为未办理缴费手续。了解缴费更多详细信息及办理缴费业务, 请登录国家知识产权局官方网站。

3. 申请人可以访问国家知识产权局政府网站 (www.cnipa.gov.cn), 在专利检索栏目中查询公布文本。如果申请人需要纸件申请公布单行本的纸件, 可向国家知识产权局请求获取。

4. 申请文件修改格式要求:

对权利要求修改的应当提交相应的权利要求替换项, 涉及权利要求引用关系时, 则需要将相应权项一起替换补正。如果申请人需要删除部分权项, 申请人应该提交整理后连续编号的部分权利要求书。

对说明书修改的应当提交相应的说明书替换段, 不得增加和删除段号, 仅只能对有修改部分段进行整段替换。如果要增加内容, 则只能增加在某一段中; 如果需要删除一个整段内容, 应该保留该段号, 并在此段号后注明: “此段删除” 字样。段号以国家知识产权局回传的或公布/授权公告的说明书段号为准。

对说明书附图修改的应当以图为单位提交相应的替换附图。

对说明书摘要文字部分修改的应当提交相应的替换页。对摘要附图修改的应当重新指定。

同时, 申请人应当在补正书或意见陈述书中标明修改涉及的权项、段号、图、页。

审查员: 自动审查

联系电话: 010-62356655

审查部门: 初审及流程管理部



210305
2023.03

纸件申请, 回函请寄: 100088 北京市海淀区蓟门桥西土城路 6 号 国家知识产权局专利局受理处收
电子申请, 应当通过专利业务办理系统以电子文件形式提交相关文件。除另有规定外, 以纸件等其他形式提交的文件视为未提交。



国家知识产权局

310013

浙江省杭州市西湖区古墩路 701 号紫金广场 B 座 1103 室 杭州求是
专利事务所有限公司
刘静(0571-87911726-809)

发文日:

2024 年 09 月 10 日



申请号或专利号: 202410870930.6

发文序号: 2024091002182900

申请人或专利权人: 浙江大学

发明创造名称: 基于未知环境下四旋翼飞行器时间最优避障在线规划方法

发明专利申请进入实质审查阶段通知书

上述专利申请, 根据申请人提出的实质审查请求, 经审查, 符合专利法第 35 条及实施细则第 113 条的规定, 该专利申请进入实质审查阶段。

提示:

1. 根据专利法实施细则第 57 条第 1 款的规定, 发明专利申请人自收到本通知书之日起 3 个月内, 可以对发明专利申请主动提出修改。

2. 申请文件修改格式要求:

对权利要求修改的应当提交相应的权利要求替换项, 涉及权利要求引用关系时, 则需要将相应权项一起替换补正。如果申请人需要删除部分权项, 申请人应该提交整理后连续编号的部分权利要求书。

对说明书修改的应当提交相应的说明书替换段, 不得增加和删除段号, 仅只能对有修改部分段进行整段替换。如果要增加内容, 则只能增加在某一段中; 如果需要删除一个整段内容, 应该保留该段号, 并在此段号后注明: “此段删除” 字样。段号以国家知识产权局回传的或公布/授权公告的说明书段号为准。

对说明书附图修改的应当以图为单位提交相应的替换附图。

对说明书摘要文字部分修改的应当提交相应的替换页。对摘要附图修改的应当重新指定。

同时, 申请人应当在补正书或意见陈述书中标明修改涉及的权项、段号、图、页。

审查员: 自动审查

联系电话: 010-62356655

审查部门: 初审及流程管理部



210307
2023.03

纸件申请, 回函请寄: 100088 北京市海淀区蓟门桥西土城路 6 号 国家知识产权局专利局受理处收
电子申请, 应当通过专利业务办理系统以电子文件形式提交相关文件。除另有规定外, 以纸件等其他形式提交的文件视为未提交。

作为第一作者的论文被 ICRA2025 接收，ICRA 为 IEEE International Conference on Robotics and Automation (ICRA)，为机器人顶会，CAAI（中国人工智能学会）A 类会议，其中邮件为发送给通讯作者的接收邮件，下图为官网上的接收结果。

Jiazhe Yuan's ICRA 2025 Submissions							
<p>Check the column 'Status' for the status of your submission, and the column 'Actions for the corresponding author' for pending actions and deadlines</p> <p>Move your mouse pointer over 'Choose an option' to open a menu with several useful options. Click anywhere within the browser window to close the menu</p>					<p>Important notice</p> <p>Links in the column 'Actions for the corresponding author' are ONLY available to the corresponding author (denoted by * in the column 'Authors or proposers')</p> <p style="text-align: center;"></p>		
Number	Type of submission	Authors or proposers *Corresponding author	Title	Profile	Status	Actions for the corresponding author - Mandatory action - Optional action Follow the link if available	Options (Submission details, files,...)
4195	Contributed paper	Jiazhe Yuan, Dongcheng Cao, Jiahao Mei, Jiming Chen, Shuo Li* (425980, 425981, 388363, 163844, 230382)	Safety-critical Online Quadrotor Trajectory Planner for Agile Flights in Unknown Environments	Planning and Simulation	Accepted	- Submit the final version until March 6, 2025	Choose an option
	Attachment to submission 4195		Video Attachment (attachment to first submission).		Accepted		

15:39

HD 5G 5G 4



The PaperPlaza Conference/Journal Management System 01/28

Message from The PaperPlaza Conference/Journal Management System

Message originated by Seth Hutchinson

To: Dr. Shuo Li

Re: ICRA 2025 Contributed paper 4195: Safety-critical Online Quadrotor Trajectory Planner for Agile Flights in Unknown Environments

Dear Shuo,

Congratulations! It is our pleasure to inform you that your paper has been accepted for presentation at the 2025 IEEE International Conference on Robotics and Automation (ICRA), May 19-23, 2025, Atlanta, USA, and for inclusion in the conference proceedings.

In this email you will find information about:

- Conference format
- Final submission instructions
- Registration
- Conference process and statistics
- Additional information

Dates to keep in mind:

Feb 6: PaperPlaza opens for Final paper submission

March 6: Final paper submissions deadline

March 6: Author registration deadline

April 12: Graphical Abstract due

May 19-May 23: ICRA!

Safety-critical Online Quadrotor Trajectory Planner for Agile Flights in Unknown Environments

Jiazhe Yuan¹, Dongcheng Cao¹, Jiahao Mei², Jiming Chen¹, *Fellow, IEEE*, and Shuo Li¹

Abstract—Autonomous high-speed flight in unknown, cluttered environments is essential for a variety of quadrotor applications, such as inspection, search, and rescue. In this study, we propose a novel trajectory planner designed to achieve efficient, high-speed, collision-free flights in such environments. The proposed approach begins by generating a safe flight corridor based on the path found by Lazy Theta*, representing the safe regions with polytopic sets. These sets are then used to define discrete-time control barrier function (DCBF), ensuring the quadrotor stays within safe bounds during flight. By selecting a single waypoint ahead of the quadrotor on the path as the next waypoint, the trajectory is optimized by considering both the total flight time and safety constraints. Extensive simulations and real-world experiments have confirmed our method’s feasibility, demonstrating its capability for high-speed performance and reliable obstacle avoidance. [video⁴]

I. INTRODUCTION

In the past decade, the application scenarios of quadrotors have been substantially broadened owing to the rapid development of agile trajectory planning methods. Among these planning methods, those for obstacle avoidance have increasingly garnered research interest due to their promising application prospects, including inspection, rescue operations and aerial videography [1], [2], [3].

Trajectory planning in cluttered environments has always been a challenging problem [4]. Planning a collision-free trajectory with limited computational resources imposes unavoidable challenges, particularly when it comes to real-time planning in unknown environments and meeting the demands of high-speed performance. Planning collision-free trajectories under these constraints requires efficient algorithms capable of rapid adaptation to unknown surroundings while ensuring safe and stable flights.

To solve this problem, a common approach is to formulate the trajectory planning task as an optimization problem. These methods often use polynomial trajectory representation and are based on the differential flatness property [5], which generates the collision-free motions in the flat output space of quadrotors. In these studies, the obstacle avoidance task is taken into account by incorporating corresponding penalty functions or constraints into the optimization problem. For example, a Euclidean Signed Distance Field (ESDF)

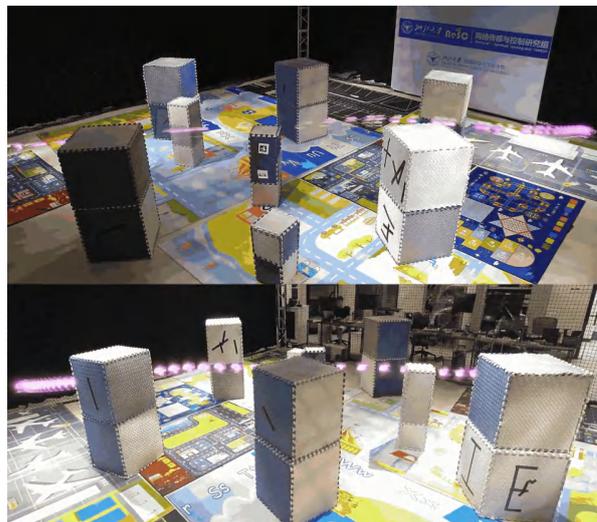


Fig. 1: The real-world experiment of our study.

[6] is often constructed to represent a neighborhood area. Based on this, the trajectory in [7], [8] is a uniform B-spline optimized with penalty functions to avoid collisions. Additionally, generating a safe flight corridor (SFC) is another common approach to represent the safe flight region [9]. The safe region is denoted by 3-dimensional polytopic sets. These polytopic sets are further used to form safety constraints in the optimization problem.

While online obstacle avoidance capabilities have been demonstrated in [10] and [11], these approaches exhibit limited velocities during flight caused by the insufficient incorporation of time-optimality criteria within their trajectory planning frameworks. Although [12] considers time-optimality to enhance trajectory performance, it still faces the fundamental limitations described in [13]: the inherent smoothness of polynomial representations constrains control inputs, preventing truly time-optimal trajectories. Another recent work [14] utilizes the framework of Model Predictive Contouring Control [15], meanwhile incorporating the Discrete-time Control Barrier Function (DCBF) technique [16] as safety constraints into the framework. However, its high-speed performance is based on the assumption that the obstacle-dense environment is already known.

Although the aforementioned works have achieved notable progress in high-speed flight capabilities, their performance remains suboptimal. Methods from autonomous drone racing research [13], [17] have shown significant potential, where multi-waypoint flight tasks have been extensively studied with demonstrated success in time optimal trajectory

¹Authors are with the College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China shuo.li@zju.edu.cn

²Jiahao Mei is with the Department of Automation, Zhejiang University of Technology, Hangzhou 310023, China.

³This work was supported in part by NSFC under Grants 62203385, 62088101.

⁴https://www.youtube.com/playlist?list=PLJFduoH7QIC0hcIX3JFsZwB4Igs4_-sPt

generation. For time-optimality considerations, [18] treats the total flight time as an optimization variable and the objective function in the optimization problem. Although time-optimal flight is achieved, the high computational cost prohibits online generation of such trajectories. A recent work [19] seeks to overcome this limitation by fixing the time when passing through the waypoints and introducing a warm-up technique to provide a suitable initial solution. These improvements significantly reduce computation time while ensuring time optimality. Both studies incorporate the quadrotor’s dynamic model as constraints during the trajectory planning process, avoiding the limitations inherent to the smoothness of polynomial trajectories, thereby significantly improving speed performance. However, they primarily focus on sequential multi-waypoint flight tasks in open terrains and neglect the challenge of cluttered environments.

In this work, we address the problem of online high-speed collision-free trajectory planning in unknown environments. To solve this problem, we propose an optimization framework that ensures both quadrotor safety and high-speed performance, while maintaining low computational cost. In the front-end processing, first we find an initial collision-free path using Lazy Theta* method and generate a safe flight corridor to provide the polytopic sets denoting the safe flight region, then we design a discrete-time control barrier function (DCBF) to serve as safety constraints in the following trajectory planning process. Furthermore, to overcome the increased computational time and performance limitations caused by using multiple waypoints along the collision-free path, we select one single waypoint ahead of the quadrotor on the path as the next waypoint. In the following trajectory planning process, we treat time as both an optimization variable and part of the objective function, while utilizing discrete-time control barrier function (DCBF) constraints to guarantee safety in cluttered environments.

The pipeline of this work is shown in Fig. 2, and the contributions of this work are listed as follows:

- 1) We propose a novel quadrotor trajectory planner that achieves online trajectory planning for high-speed obstacle avoidance flight tasks.
- 2) Our planning method achieves high-speed obstacle avoidance by leveraging full quadrotor dynamics, using DCBF constraints to maintain dynamic feasibility while ensuring safe navigation through cluttered environments.
- 3) We validate the effectiveness and robustness of our proposed method through extensive experiments both in simulation and real-world scenarios, demonstrating its superiority in achieving both high-speed and reliable obstacle avoidance.

II. FRONT-END PROCESSING

In this section, we first compute an initial collision-free path (Step 1 in Fig. 3) using Lazy Theta*[20], followed by the construction of a safe flight corridor (Step 2 in Fig. 3) based on the computed path. Subsequently, we design a discrete-time control barrier function (DCBF) (Step 3 in

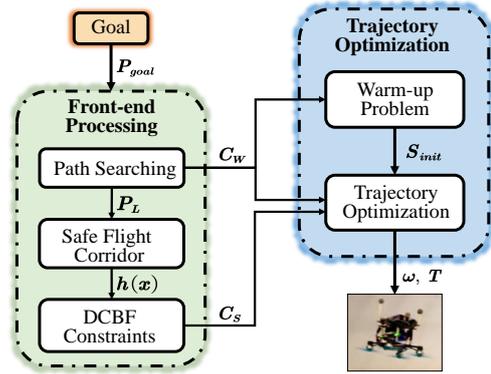


Fig. 2: The framework diagram of our study.

Fig. 3) to ensure precise obstacle avoidance by leveraging the polytopic sets derived from the safe flight corridor.

A. Path Searching and Safe Flight Corridor Generation

To generate efficient collision-free initial paths, we employ the Lazy Theta* algorithm [20]. This approach improves upon traditional A* by eliminating unnecessary waypoints through line-of-sight checks, resulting in shorter paths while maintaining comparable computational efficiency.

We construct a safe flight corridor around the waypoints generated by the Lazy Theta* algorithm [21]. This corridor consists of convex polytopic sets that explicitly define the safe flight region. These convex sets are crucial to our method, as they enable the generation of a precise obstacle avoidance trajectory, which will be further elaborated in Section III. The safe flight corridor, denoted as \mathcal{C} , is the union of $N_{\mathcal{P}}$ convex polytopic sets \mathcal{C}_i , defined as follows:

$$\mathcal{C} = \bigcup_{i=1}^{N_{\mathcal{P}}} \mathcal{C}_i, \quad (1)$$

$$\mathcal{C}_i = \{ \mathbf{p} \in \mathbb{R}^3 \mid \mathbf{A}_j^T \mathbf{p} \leq \mathbf{b}_j, j = 1, \dots, N_{fi} \},$$

where $\mathbf{A}_j \in \mathbb{R}^3$ represents the normal vector of the j^{th} facet, $\mathbf{b}_j \in \mathbb{R}$ denotes the offset of the j^{th} facet, and N_{fi} indicates the number of facets in the i^{th} polytopic set.

B. Obstacle Avoidance with DCBF

Since the polytopic sets from the safe flight corridor alone do not ensure obstacle avoidance, we introduce the discrete-time control barrier function (DCBF) to regulate the quadrotor’s trajectory. The discrete-time control barrier function (DCBF) is a mathematical tool [22] used to ensure the safety of discrete-time dynamic systems by defining a safe set for quadrotor states. By establishing a safe region in the state space, it enforces state constraints, guiding the quadrotor’s trajectories to remain within the safe set.

The discrete-time control barrier function (DCBF) $h : \mathcal{X} \rightarrow \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^n$ denotes the state space, is designed to keep the quadrotor within a predefined safe set by regulating its movement over time. The safe set \mathcal{S} , which defines the safe range of quadrotor states within the state

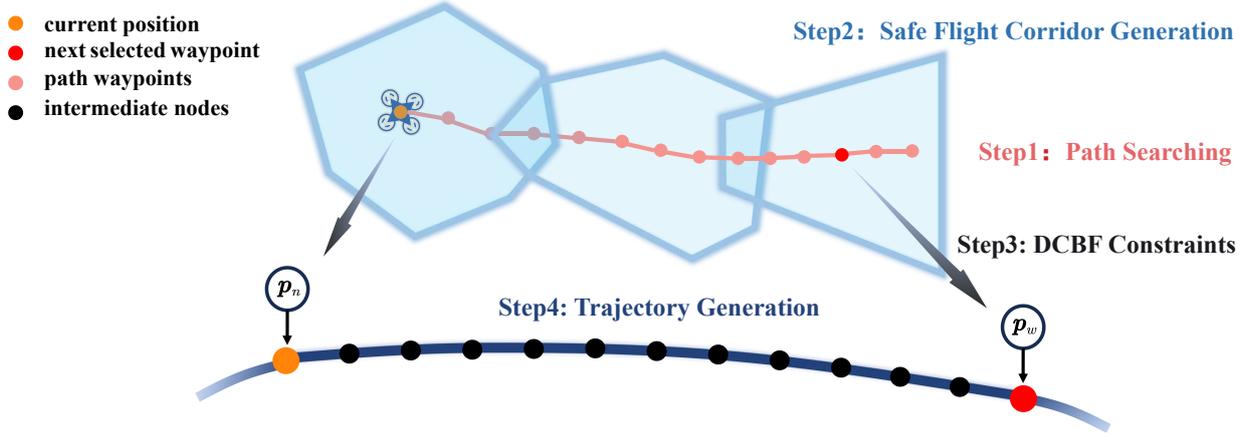


Fig. 3: Methodology Overview. The process begins with Step 1, where an initial collision-free path is computed. In Step 2, a safe flight corridor (SFC) is generated. Step 3 involves the formulation of discrete-time control barrier function (DCBF) constraints based on the SFC. Finally, in Step 4, after selecting a specific waypoint from the computed path, the trajectory is optimized to achieve high-speed motion while adhering to the DCBF safety constraints to ensure obstacle avoidance.

space under safety constraints, is defined as the 0-superlevel set of the discrete-time control barrier function (DCBF):

$$\mathcal{S} := \{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) \geq 0\}, \quad (2)$$

where \mathbf{x} represents quadrotor states and $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$. This formulation ensures that as long as $h(\mathbf{x}) \geq 0$, the quadrotor states remain within the safe region \mathcal{S} .

With the convex polytopic sets provided by safe flight corridor in Section II-A, we specifically design the discrete-time control barrier function (DCBF) using the Euclidean distance between the quadrotor and each facet of the current polytopic set differing from a safe distance constant d_{safe} . When the quadrotor reaches the i^{th} convex polytopic set \mathcal{C} , the corresponding DCBF is designed as follows:

$$h_i(\mathbf{x}) = \frac{|\mathbf{A}_j^T \mathbf{x}_p - b_j|}{\|\mathbf{A}_j\|} - d_{\text{safe}}, \quad (3)$$

$$j = 1, 2, \dots, N_{fi},$$

where $\mathbf{x}_p \in \mathbb{R}^3$ denotes the current position of the quadrotor, and d_{safe} denotes the necessary safe distance between the quadrotor and the boundary of the safe flight corridor.

Given the definition of the DCBF which ensures that the quadrotor states remain within the safe set \mathcal{S} , it becomes necessary to impose additional constraints to regulate how quickly the system approaches the boundary of this set. Similar to [23], we impose a decay condition on $h(\mathbf{x})$, ensuring that the function value decreases at a controlled rate as the quadrotor states evolve over time. This decay condition not only maintains safety by preventing the system from exiting the safe set too quickly, but also provides flexibility in system control by balancing safety with trajectory feasibility. Following this principle, the exponential decay condition can be expressed as follows:

$$h(\mathbf{x}_{k+1}) \geq \gamma_k \cdot h(\mathbf{x}_k), \quad 0 \leq \gamma_k \leq 1, \quad (4)$$

where the DCBF decreases between successive time steps at a decay rate γ_k . Therefore, given a valid DCBF $h(\mathbf{x})$,

imposing constraint (4) in an optimization problem ensures system safety by generating collision-free trajectories. However, a fixed decay rate γ_k leads to a tradeoff between system safety and feasibility of a collision-free trajectory. Reducing the decay rate enlarges the feasible domain under the constraint (4) but meanwhile results in a quick approach to the boundary of the safe set, denoted by $\partial\mathcal{S} := \{x \in \mathcal{X} : h(x) = 0\}$. This rapid approach can make the trajectory unsafe. Conversely, increasing the decay rate prioritizes safety but reduces feasibility [24]. To address this challenge, we introduce a slack variable η_k into DCBF constraints as follows:

$$h(\mathbf{x}_{k+1}) \geq \gamma_k \cdot \eta_k \cdot h(\mathbf{x}_k), \quad 0 \leq \gamma_k \leq 1, \quad (5)$$

where the relaxing variable η_k is optimized with other variables inside an optimization problem, which resolves the tradeoff between system safety and collision-free trajectory feasibility.

III. TRAJECTORY OPTIMIZATION

In this section, building upon the path and DCBF constraints introduced in Section II, we present the detailed process of trajectory planning. A waypoint on the path located at a specific distance ahead of the quadrotor is selected as the next required waypoint. The trajectory is then generated with a focus on both high-speed motion and DCBF safety constraints (Step 4 in Fig. 3). This results in a real-time planning framework that dynamically selects waypoints while optimizing for high-speed and collision-free flight.

A. Quadrotor Dynamics

We utilize the same quadrotor dynamical model as presented in [18]. The dynamics of the quadrotor are given by:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{cases} \mathbf{v} \\ \mathbf{g} + \mathbf{R}(\mathbf{q})\mathbf{z}_B\mathbf{T} \\ \frac{1}{2}\Lambda(\mathbf{q}) \begin{bmatrix} 0 & \boldsymbol{\omega}^T \end{bmatrix} \end{cases} \quad (6)$$

where $\mathbf{T} = [0, 0, T]^T$ represents the mass-normalized thrust vector, $\mathbf{R}(\mathbf{q})$ denotes the rotation matrix, \mathbf{z}_B is the unit vector along the body z-axis and $\Lambda(\mathbf{q})$ is the quaternion kinematics matrix. The variables \mathbf{p} , \mathbf{v} , \mathbf{q} and $\boldsymbol{\omega}$ represent the position, inertial velocity, attitude quaternion, and body rate, respectively. To achieve smoother thrust transitions, we define the control input as $\mathbf{u} = [\Delta T, \boldsymbol{\omega}^T]^T$, where ΔT represents the thrust rate with $\dot{T} = \Delta T$, and $\boldsymbol{\omega}$ are the body angular rates.

The discrete-time formulation of (6) is also derived, which is used in the following optimization process.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, dt_k) \quad (7)$$

B. Optimization Formulation with Safety Constraints

1) *Trajectory Planning*: The trajectory generation method proposed in this work accounts for both obstacle avoidance and the minimization of the total flight time. Specifically, time is treated as an optimization variable and integrated into the objective function to ensure a high-speed trajectory. Additionally, the previously introduced DCBF constraints (5) are imposed in the optimization process, thereby guaranteeing collision-free trajectories throughout the flight.

Generating trajectories based on multiple waypoints in Section II-A leads to a more heavy computational burden. Therefore, in this study, to generate optimal trajectories online, we select the waypoint which is located 2m ahead of the quadrotor, as the next waypoint \mathbf{p}_w (the red point in Fig. 3) and discard all the waypoints between the current position of the quadrotor and \mathbf{p}_w . Similar to [19], to generate the optimal trajectory, we fix the number of the intermediate nodes N_{tr} (the black points in Fig. 3). In other words, the trajectory to be generated is discretized into N_{tr} segments, each with a sampling interval $dt > 0$. Consequently, the total flight time can be obtained as follows and our optimization target is to minimize this flight time P_t :

$$P_t = N_{tr} \cdot dt. \quad (8)$$

In addition, penalties on η_k (the slack variable for DCBF constraints) and \mathbf{u}_k (the control input) are incorporated into the objective function, which prevents the solver from selecting these variables arbitrarily. The introduction of these additional penalties has been demonstrated as crucial for maintaining stability in nonlinear optimization formulations [25], [26]. Without them, the generated solutions often exhibit excessive noise, rendering the control inputs impractical for real-world flight scenarios. To ensure obstacle perception during flight, we compute the yaw angle difference $\Delta\psi$ as $\Delta\psi = \psi_{\text{desired}} - \psi_{N_{tr}}$, where the desired yaw angle ψ_{desired} is derived from the direction vector pointing to the next waypoint. Inspired by [27], to ensure quadrotor safety, we set the desired velocity v_{desire} according to the complexity of the environment. A penalty function:

$$P_v = \sum_{k=1}^{N_{tr}} \max(0, \|\mathbf{v}_k\| - v_{\text{desire}}) \quad (9)$$

is introduced to constrain velocities from exceeding the sampling threshold.

Finally, the proposed collision-free optimization problem, incorporating both the DCBF constraints (5) and dynamic constraints (7), can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{x}_k, \mathbf{u}_k, dt, \eta_k} \quad & P_t + P_v + \|\eta_k\|_{Q_\eta}^2 + \|\boldsymbol{\omega}_k\|_{Q_\omega}^2 \\ & + \|\Delta\mathbf{T}_k\|_{Q_{\Delta T}}^2 + \|\Delta\psi\|_{Q_{\Delta\psi}}^2 \\ \text{s.t.} \quad & \|\mathbf{p}_n - \mathbf{p}_w\|_2^2 \leq \delta_i^2 \\ & \mathbf{x}_{lb} \leq \mathbf{x}_k \leq \mathbf{x}_{ub}, \mathbf{x}_0 = \mathbf{x}_{init} \\ & T_{lb} \leq T_k \leq T_{ub} \\ & \mathbf{u}_{lb} \leq \mathbf{u}_k \leq \mathbf{u}_{ub} \\ & \text{the constraints in (5)} \end{aligned} \quad (10)$$

where $n = 1, 2, \dots, N_{tr}$ denotes the number of the current time step.

2) *Online Replanning with Warm-up Technique*: Due to the presence of nonlinear dynamic constraints, solving Problem (10) without a suitable initial solution leads to either an infeasible solution which violates the given constraints or an optimal but computationally expensive solution. Similar to [19], we formulate a preliminary optimization problem to serve as a warm-up technique prior to solving Problem (10), providing an initial solution. This ensures both the quality of the final solution and a manageable computation time when using the interior-point method.

In the preliminary optimization problem, first we relax the waypoint, dynamics, and DCBF constraints by integrating them into the objective function using penalty terms. Specifically, we define the penalty functions as follows: $P_w = \|\mathbf{p}_{N_{tr}} - \mathbf{p}_w\|_2^2$ for the waypoint constraint, $P_d = \sum_{k=0}^{N_{tr}} \|\mathbf{x}_{k+1} - f(\mathbf{x}_k, \mathbf{u}_k, dt_0)\|_2^2$ for the dynamics constraint, where dt_0 is pre-assigned. We design an adaptive strategy to assign dt_0 a corresponding value based on time-optimal trajectories for a point mass. The minimum time required to reach the next waypoint is determined by solving the time-optimal trajectory for a point mass, applying maximum acceleration \mathbf{a}_{max} from the current position \mathbf{p}_c and velocity \mathbf{v}_c , where the equation $\mathbf{v}_c t_{min} + \frac{1}{2} \mathbf{a}_{max} t_{min}^2 = \mathbf{p}_c - \mathbf{p}_w$ holds. Once the minimum time t_{min} is obtained, dt_0 is calculated by $dt_0 = t_{min}/N_{tr}$.

Additionally, penalties on η_k and \mathbf{u}_k are retained in the objective function to ensure the stability of the preliminary optimization problem. The final formulation of the preliminary optimization problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{x}_k, \mathbf{u}_k, \eta_k} \quad & P_w + P_d + \|\eta_k\|_{Q_\eta}^2 + \|\boldsymbol{\omega}_k\|_{Q_\omega}^2 \\ & + \|\Delta\mathbf{T}_k\|_{Q_{\Delta T}}^2 \\ \text{s.t.} \quad & \mathbf{x}_{lb} \leq \mathbf{x}_k \leq \mathbf{x}_{ub}, \mathbf{x}_0 = \mathbf{x}_{init} \\ & \mathbf{u}_{lb} \leq \mathbf{u}_k \leq \mathbf{u}_{ub} \\ & T_{lb} \leq T_k \leq T_{ub} \end{aligned} \quad (11)$$

By appropriately selecting dt_0 and utilizing the interior point method, the solution to the preliminary optimization Problem

TABLE I: Comparison between with EGO-Planner-v2[28] and our method. The comparison is conducted in two environments with different obstacle densities. The metrics displayed in the table are flight time, average velocity, peak velocity, and flight distance, all presented in the format of **mean \pm standard deviation**, as well as the success rate.

Density (obs/m ²)	Method	Flight Time (s) (Mean \pm Std. Dev.)	Velocity (m/s, Mean \pm Std. Dev.)		Flight Distance (m) (Mean \pm Std. Dev.)	Success Rate (%)
			Avg. Vel.	Peak Vel.		
0.15	EGO-Planner-v2($v_{max} = 10$)	14.011 \pm 2.404	3.883 \pm 0.608	9.440 \pm 0.313	53.001 \pm 0.926	100
	Our Method ($v_{max} = 10$)	7.972 \pm 0.093	6.302 \pm 0.074	9.991 \pm 0.141	50.293 \pm 0.096	100
0.3	EGO-Planner-v2($v_{max} = 8$)	17.625 \pm 1.576	3.146 \pm 0.287	7.972 \pm 0.260	54.175 \pm 1.099	90
	EGO-Planner-v2($v_{max} = 10$)	14.333 \pm 2.357	3.886 \pm 0.414	9.572 \pm 0.438	55.467 \pm 3.294	60
	Our Method ($v_{max} = 10$)	8.692 \pm 0.190	5.825 \pm 0.126	9.080 \pm 0.027	50.658 \pm 0.078	100

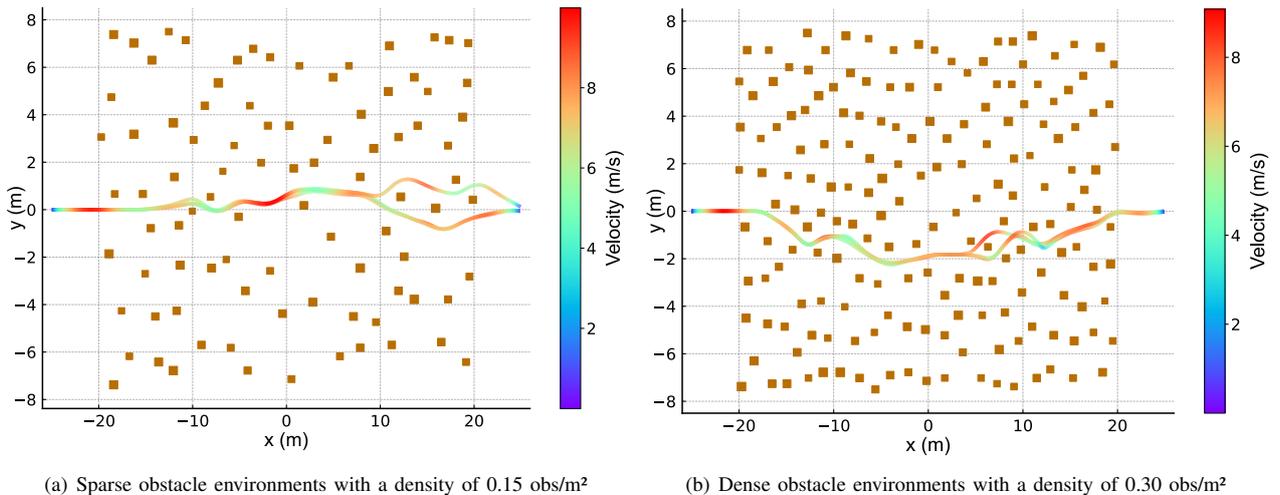


Fig. 4: Simulation results: Trajectory and Velocity Performance in Sparse and Dense Obstacle Environments. We present 2 representative trajectories from 10 repeated experiments, which exhibit slight variations but consistently demonstrate similar overall speed and time performance, shown in Table I. This consistency highlights the stability of our method in performing real-time trajectory planning tasks.

(11) can be obtained, which serves as the initialization for Problem (10).

Furthermore, to facilitate online collision-free trajectory generation, both Problem (10) and Problem (11) are solved in a receding horizon fashion. Specifically, the quadrotor computes trajectories within a limited horizon and continuously updates them at a high frequency, ensuring both system safety and low computation time.

IV. SIMULATION EXPERIMENT

In this section, we conduct a series of simulation experiments to evaluate the advantages of our method in comparison with existing approaches. The simulations are implemented in C++ using the ROS communication framework. The initial collision-free path searching, detailed in Section II-A, is conducted in simulation with a frequency of 50Hz. As detailed in Section III, during each iteration of trajectory generation, the number of intermediate nodes is set to $N_{tr} = 10$. The optimization problem (10) is solved using CASADI [29] as the code generation tool and IPOPT [30] as the solver. The solving time is about 20 ms on a laptop with an Intel I7-11800H CPU with a base frequency of 2.5GHz and RAM of 32G.

As an online trajectory generation method, MINCO [12] has shown excellent performance in obstacle avoidance, ensuring trajectory safety in both sparse and dense environments. Therefore, we selected the MINCO-based online obstacle avoidance planner, EGO-Planner-v2[28], as the baseline for our comparative experiments. We evaluate the performance of our method against the EGO-Planner-v2 with single drone deployment, by comparing key metrics such as success rate, average velocity, maximum velocity, and flight time. The comparison is conducted in two environments with different obstacle densities. One environment features a sparse distribution of cylindrical obstacles with a density of 0.15 obs/m², while the other features a denser distribution with a density of 0.3 obs/m². In both scenarios, the flight tests take place within a confined space measuring 40m \times 15m \times 5m. The minimum distance between any two obstacles is maintained at 1m. The dynamical limits for EGO-Planner-v2 are set to maximum acceleration and maximum jerk, which correspond to the same maximum thrust and thrust change rate in our method. To compare the stability performance of our method against EGO-Planner-v2, the simulation experiments under each configuration are repeated 10 times, and all reported results represent the

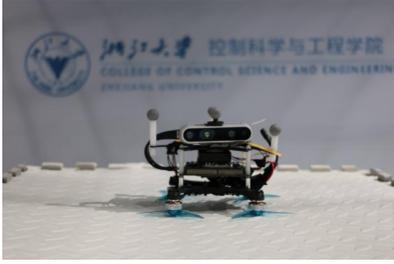


Fig. 5: The snapshot of our quadrotor.

average of these repeated trials.

As shown in Table I and Fig. 4, our method consistently outperforms EGO-Planner-v2 in both sparse and dense environments across various metrics, including flight time, velocity, and success rate. In the sparse environment with a density of 0.15 obs/m², EGO-Planner-v2 achieves a 100% success rate, but its average velocity is limited to 3.88 m/s, resulting in a longer flight time of 14.01 s. In contrast, our method, maintains a 100% success rate while significantly increasing the average velocity to 6.30 m/s, reducing the flight time to 7.97 s.

In the dense environment with a density of 0.3 obs/m², EGO-Planner-v2 shows a notable decline in performance. When its maximum velocity is limited to $v_{\max} = 10$ m/s, its success rate drops dramatically to 60%, with an average velocity of 3.89 m/s and a longer flight time of 14.33 s due to trajectory tracking errors increasing during sharp turns. Even with a reduced limitation $v_{\max} = 8$ m/s, EGO-Planner-v2 only achieves a success rate of 90%, while its average velocity remains lower at 3.15 m/s, leading to an increased flight time of 17.63 s. On the other hand, our method, illustrated in Fig. 4(b), consistently maintains a 100% success rate in both configurations, with an average velocity of 5.83 m/s in the dense environment and a significantly shorter flight time of 8.69 s. Moreover, our method demonstrates more efficient trajectories, covering shorter distances than the EGO-Planner-v2, which traverses more ground due to less optimal path planning. In both environments, our method achieves superior stability, as evidenced by consistently higher success rates, faster velocities, and shorter flight times, outperforming EGO-Planner-v2 across all metrics.

V. REAL-WORLD EXPERIMENT

Our experimental setup features a custom-developed quadrotor equipped with an Intel RealSense D435 depth camera for real-time mapping and a Jetson Orin NX for onboard processing, as shown in Fig. 5. The quadrotor weighs 338 g and features a thrust-to-weight ratio of 3.5, providing stable flight performance. For precise state estimation, we utilize a motion capture system that tracks the quadrotor’s position and velocity. All experiments are conducted within a 5 m × 5 m × 2.5 m free-flight space, as shown in Fig. 1. The key parameters for trajectory generation, including waypoint selection and intermediate point configuration, are set identical to those used in the simulation experiments described in Section IV.

TABLE II: Flight Results of Experiments

V_max Limit (m/s)	Time (s)	Max Speed (m/s)	Avg Speed (m/s)
4	2.97	4.04	3.0
6	2.80	5.95	3.5

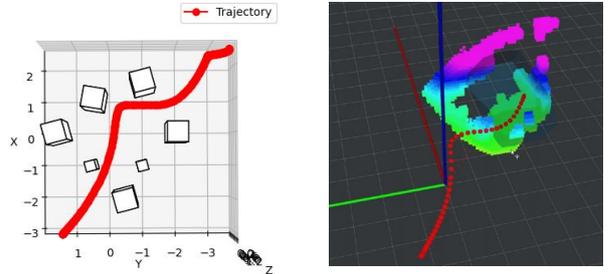


Fig. 6: Trajectory results and rviz visualization in the real-world experiments with the maximum velocity limit of 6 m/s.

In real-world experiments, the quadrotor departs from a designated hover point, autonomously generates a collision-free high-speed trajectory, and completes the flight by reaching the designated endpoint, where it hovers again. Each trajectory’s flight time is determined by the time interval between the two hover points. Specifically, the starting and ending points are set as $[-2.5, 1, 0.7]$ m and $[2.2, -3.5, 0.7]$ m, respectively. To further evaluate the stability of our method, we conduct a series of experiments under different maximum velocity limits set to 4 m/s and 6 m/s respectively, aiming to assess its performance across various velocity scenarios.

Fig. 6 and Table II illustrate the quadrotor’s trajectory results under these varying maximum velocity limits. The data shows that increasing the velocity limit leads to a shorter flight time and higher overall speeds. The drone’s ability to nearly reach the specified velocity limits in both cases suggests effective control and robust performance in high-speed scenarios. This experiment demonstrates the feasibility of the proposed method in the real world and also its adaptability to different speed constraints.

VI. CONCLUSION

This paper presents a novel trajectory planning framework tailored for high-speed, obstacle-avoidance flights in unknown environments. By utilizing Lazy Theta* to generate a safe flight corridor and incorporating discrete-time control barrier functions (DCBFs) for safety constraints. Our method ensures collision-free navigation while maintaining real-time performance. Through comprehensive simulations and real-world experiments, our method consistently demonstrates superior performance compared to existing approaches.

Future work will focus on conducting physical experiments in environments with higher obstacle densities and conducting comprehensive field trials in natural settings. Additionally, we plan to adapt the method for multi-agent trajectory planning to enable efficient coordination in cluttered airspace. These efforts aim to enhance quadrotors’ capabilities in more complex real-world environments.

REFERENCES

- [1] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *nature*, vol. 521, no. 7553, pp. 460–466, 2015.
- [2] G. Loianno and D. Scaramuzza, "Special issue on future challenges and opportunities in vision-based drone navigation," *Journal of Field Robotics*, vol. 37, no. 4, 2020.
- [3] X. Guo, Y. He, L. Shanguan, Y. Chen, C. Gu, Y. Shu, K. Jamieson, and J. Chen, "Mighty: Towards Long-Range and High-Throughput Backscatter for Drones," *IEEE Transactions on Mobile Computing*, vol. 24, pp. 1833–1845, Mar. 2025.
- [4] G. Chen, D. Sun, W. Dong, X. Sheng, X. Zhu, and H. Ding, "Computationally efficient trajectory planning for high speed obstacle avoidance of a quadrotor with active sensing," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3365–3372, 2021.
- [5] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*, pp. 2520–2525, IEEE, 2011.
- [6] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.
- [7] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1992–2009, 2021.
- [8] W. Ding, W. Gao, K. Wang, and S. Shen, "An efficient b-spline-based kinodynamic replanning framework for quadrotors," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1287–1306, 2019.
- [9] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.
- [10] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.
- [11] B. Zhou, F. Gao, J. Pan, and S. Shen, "Robust real-time uav replanning using guided gradient-based optimization and topological paths," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1208–1214, IEEE, 2020.
- [12] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.
- [13] D. Hanover, A. Loquercio, L. Bauersfeld, A. Romero, R. Penicka, Y. Song, G. Cioffi, E. Kaufmann, and D. Scaramuzza, "Autonomous drone racing: A survey," *IEEE Transactions on Robotics*, 2024.
- [14] J. Qiu, Q. Liu, J. Qin, D. Cheng, Y. Tian, and Q. Ma, "Pe-planner: A performance-enhanced quadrotor motion planner for autonomous flight in complex and dynamic environments," *IEEE Robotics and Automation Letters*, 2024.
- [15] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3340–3356, 2022.
- [16] Z. Jian, Z. Yan, X. Lei, Z. Lu, B. Lan, X. Wang, and B. Liang, "Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3679–3685, IEEE, 2023.
- [17] H. Moon, J. Martinez-Carranza, T. Cieslewski, M. Faessler, D. Falanga, A. Simovic, D. Scaramuzza, S. Li, M. Ozo, C. De Wagter, et al., "Challenges and implemented technologies used in autonomous drone racing," *Intelligent Service Robotics*, vol. 12, pp. 137–148, 2019.
- [18] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," *Science Robotics*, vol. 6, 2021.
- [19] Z. Zhou, G. Wang, J. Sun, J. Wang, and J. Chen, "Efficient and robust time-optimal trajectory planning and control for agile quadrotor flight," *IEEE Robotics and Automation Letters*, 2023.
- [20] A. Nash, S. Koenig, and C. Tovey, "Lazy theta*: Any-angle path planning and path length analysis in 3d," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, pp. 147–154, 2010.
- [21] J. Ji, N. Pan, C. Xu, and F. Gao, "Elastic tracker: A spatio-temporal trajectory planner for flexible aerial tracking," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 47–53, IEEE, 2022.
- [22] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, pp. 3420–3431, IEEE, 2019.
- [23] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 286–292, 2021.
- [24] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," *2021 American Control Conference (ACC)*, pp. 3882–3889, 2020.
- [25] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, pp. 628 – 647, 2015.
- [26] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, pp. 2994–3008, 2018.
- [27] C. Toumih and D. Floreano, "High-speed motion planning for aerial swarms in unknown and cluttered environments," *IEEE Transactions on Robotics*, vol. 40, pp. 3642–3656, 2024.
- [28] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, et al., "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.
- [29] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, p. 1–36, Mar 2019.
- [30] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.