24029602

同济大学
TONGJI UNIVERSITY

硕士学位论文

(学术学位)

障碍物密集环境中的编队导航

姓　　名：吴泽铭

学　　号：2130729

学　　院：电子与信息工程学院

学科门类：工学

一级学科：控制科学与工程

二级学科：控制理论与控制工程

研究方向：控制理论与控制工程

指导教师：王祝萍 教授

联合培养单位：

二〇二四年三月


24029602

24029602

同济大学
TONGJI UNIVERSITY

A thesis submitted to

Tongji University in partial fulfillment of the requirements for

the degree of Master of Engineering

Formation Navigation in Cluttered Environments

Candidate: Zeming Wu

Student Number: 2130729

School/Department: College of Electronics and Information Engineering

Categories: Engineering

First-level Discipline: Control Science and Engineering

Second-level Discipline: Control Theory and Control Engineering

Research Fields: Control Theory and Control Engineering

Supervisor: Prof. Zhuping Wang

March 2024


24029602

24029602

摘要

近年来，随着传感器和通讯技术的飞速发展，多机器人系统的应用场景逐渐从开阔水面、高空等障碍物稀疏的场景转向森林、废墟和城市建筑群等障碍物密集的环境。由于场景的转变，现有多机器人编队研究面临着以下三方面挑战：首先，在单机器人层面，障碍物密集场景中的机器人需要精确考虑其几何外形以实现灵活的避障，但现有文献仍偏向于基于搜索和采样的规划，难以实时精确避障；其次，在多机器人层面，当前编队算法侧重于底层控制，缺乏对环境信息的充分考虑，使得机器人集群在障碍物密集环境中难以实现安全编队；再者，障碍物密集场景中的编队需要综合考虑环境、自身和多机协同等约束，这些约束通常存在冲突，而如何引入高层次协同来消解冲突尚待研究。

为了实现障碍物密集环境中计算效率高、求解质量优的多机器人编队导航方案，本文按照从单机器人避障到多机器人编队，从低层次优化到高层次协同的思路设计了以下三个算法：

(1) 提出了一种单机器人在密集障碍物环境中的全尺寸实时避障算法。该算法利用凸多胞体拟合机器人的几何外形，并将避障问题转化为优化问题，实现了非保守的避障。为解决高维度、非线性优化难以实时求解的挑战，引入交替方向乘子法和序列二次规划，提出了适用于嵌入式设备的实时优化算法。通过无人机案例验证了算法的实际应用和高实时性。

(2) 提出了多机器人在障碍物密集环境中的协同编队轨迹优化算法。该算法利用方位约束的平移、放缩不变性为编队避障提供更多自由度，使机器人集群能够灵活地避障。通过变量消解等技术将问题转换为无约束轨迹优化问题，实现了高质量轨迹的实时求解。通过编队保持、形成和导航三方面的仿真验证了算法的有效性和优越性。

(3) 提出了一种有效消除机器人集群死锁的编队任务分配算法。该算法设计了一种完全分布式的协调方法，为集群引入高层次协同，通过精确势博弈方法寻找稳定任务分配方案。本文证明了算法的收敛性，并从理论上保证了算法能够提升编队的相似度并消解死锁。通过基准比较和联合仿真验证了算法的有效性，并在实物实验中证明了引入高层次协同能够提高编队导航的成功率。

关键词：全尺寸避障，多机器人系统，编队导航，任务分配

24029602

Abstract

In recent years, with the rapid development of sensor and communication technologies, the application scenarios of multi-robot systems have shifted from open waters and high altitudes, where obstacles are sparse, to environments with dense obstacles such as forests, ruins, and urban structures. However, the current research on multi-robot formation encounters three main challenges: firstly, in dense obstacle scenarios, robots need to precisely consider their geometric shapes for flexible obstacle avoidance. Existing literature tends to rely on search and sampling-based planning, which struggles to achieve real-time and accurate obstacle avoidance. Secondly, existing formation algorithms focus on low-level control, lacking sufficient consideration of environmental information, making it difficult to achieve safe formation in cluttered environments. Lastly, in cluttered environments, formation navigation has to deal with conflicting constraints related to the environment, self, and inter-robot collaboration. How to introduce high-level coordination to resolve these conflicts remains a topic of investigation.

To achieve a computationally efficient and high-quality multi-robot formation navigation solution in cluttered environments, three algorithms is proposed in this thesis, following a progression from single-robot obstacle avoidance to multi-robot formation, from low-level optimization to high-level coordination:

Firstly, a full-dimensional real-time obstacle avoidance algorithm is proposed for single robot. This algorithm approximates the robot's geometry with convex polyhedron, and convert the collision avoidance problem to optimization problem, enabling non-conservative obstacle avoidance. Based on this algorithm, a real-time optimization scheme that suitable for embedded devices is proposed, which solve high-dimensional and nonlinear optimization problems by alternating direction method of multipliers (ADMM) and sequential quadratic programming (SQP). The practicality and effectiveness of the proposed algorithm are validated through a case study with unmanned aerial vehicles (UAVs).

Secondly, a formation trajectory optimization algorithm for multi-robots in cluttered environments is proposed. The algorithm utilizes translational and scaling invariance from bearing constraints to provide additional freedom for formation obstacle avoidance, allowing the robot swarm to flexibly use scaling and translational maneuvers for obstacle avoidance. By transforming the problem into an unconstrained tra-

24029602

jectory optimization problem using techniques like variable elimination, the real-time high-quality trajectory solutions is achieved. The effectiveness and superiority of the proposed algorithm is validated through simulations covering formation maintenance, formation creation, and navigation.

Thirdly, a formation task assignment algorithm for multi-robot systems is proposed to resolve deadlocks. This algorithm introduces high-level coordination by proposing a fully distributed task allocation algorithm for the swarm. By transforming the task allocation problem into an exact potential game, the stable task allocation is achieved. We also proof the convergence of algorithm and guarantee the improvement of initial swarm similarity and deadlock resolution theoretically. The algorithm's effectiveness is validated through benchmark comparisons, comprehensive simulations. The real-world experiments also demonstrate that introducing high-level coordination improve the success rate of formation navigation.

Key Words: Full-dimensional Collision Avoidance, Multi-robot Systems, Formation Navigation, Task Assignment

24029602

目录

主要符号对照表 VI
插图索引 VII
表格索引 IX
算法索引 X
第 1 章 绪论 1
1.1 研究背景与意义 1
1.2 机器人编队导航研究现状 4
1.2.1 单机器人的运动规划 4
1.2.2 多机器人编队 7
1.2.3 多机器人任务分配 9
1.3 本文研究内容及主要贡献 10
第 2 章 基于优化的单机全尺寸避障方法 13
2.1 引言 13
2.2 最优化理论预备知识 14
2.3 单机避障优化问题的建模与求解 15
2.3.1 基于模型预测控制的优化问题建模 15
2.3.2 精确无碰撞约束的建模，解析表示与几何解释 17
2.3.3 GPU 辅助的优化问题实时求解 21
2.4 四旋翼无人机全尺寸避障案例研究 27
2.4.1 四旋翼无人机的微分平坦模型 27
2.4.2 AirSim 高保真仿真 32
2.5 小结 36
第 3 章 方位角约束下的编队导航方法 37
3.1 引言 37
3.2 方位角度刚性理论预备知识 37
3.3 编队导航优化问题建模与求解 40
3.3.1 编队性能指标 40
3.3.2 智能体轨迹表示方法 42
3.3.3 轨迹优化问题建模及在线求解方法 45
3.4 数值仿真验证 49
3.4.1 编队队形保持 50
3.4.2 编队队形形成 52
3.4.3 综合场景中的编队导航 54
3.5 小结 56

24029602

第 4 章 基于势博弈的编队任务分配方法 57
4.1 引言 57
4.2 博弈论预备知识 58
4.3 编队任务分配问题建模与求解 59
4.3.1 任务分配问题建模 59
4.3.2 任务分配问题求解 60
4.3.3 任务分配问题收敛性分析 62
4.4 仿真实验 64
4.4.1 任务分配仿真 64
4.4.2 任务分配与轨迹优化综合验证 65
4.4.3 CrazyFlies 实物验证 68
4.5 小结 70
第 5 章 总结与展望 72
5.1 本文工作总结 72
5.2 未来研究方向展望 73
致谢 74
参考文献 76
个人简历、在学期间发表的学术论文与研究成果 81

24029602

主要符号对照表

$a, b, c, \ldots$	使用英文小写字母表示标量
$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \ldots$	使用粗体英文小写字母表示列向量
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$	使用粗体英文大写字母表示矩阵
$(a_i), (b_i), (c_i), \ldots$	以 a_i, b_i, c_i 为元素的列向量
$\mathbb{Q}, \mathbb{R}, \mathbb{C}, \ldots$	使用黑板粗体英文大写字母表示集合
$f(x), g(x), h(x)$	函数
$\|\mathbf{a}\|_2$	向量的二范数
$\|\mathbf{a}\|_{\mathbf{W}}$	以正定矩阵为权重的向量的二范数
$\|\mathbf{A}\|_F$	矩阵的弗罗比尼乌斯范数 (frobenius norm)
$\mathbf{A}^{\mathrm{T}}$	矩阵的转置
$\mathbf{A}^{\dagger}$	矩阵的广义逆，即 Moore-Penrose 伪逆
$\mathbf{A} \otimes \mathbf{B}$	矩阵的克罗内克积
$\prec$	向量的逐元素小于
$\preceq$	向量的逐元素小于等于
$\succ$	向量的逐元素大于
$\succeq$	向量的逐元素大于等于

24029602

插图索引

图 1.1 自然界中的群体行为及多机器人在工程领域的应用。 1
图 1.2 多机器人编队的发展趋势：由开阔环境到障碍物密集环境。 2
图 1.3 基于图搜索的规划算法。 5
图 1.4 基于采样的规划算法。 6
图 1.5 本文的研究内容及技术路线图。 10
图 2.1 无人机与障碍物的几何表示。 16
图 2.2 放缩法碰撞检测的几何示意图。 18
图 2.3 无碰撞约束 (2.16) 的几何原理解释。 21
图 2.4 全尺寸避障问题的交替方向乘子法的整体流程图。 25
图 2.5 无人机的凸多胞体几何外形近似。 31
图 2.6 一百次蒙特卡洛仿真中的一个例子。 33
图 2.7 三种算法在预测时域 $T = 16$ 时，单个控制循环所需的平均计算时间。 34
图 2.8 RDA 算法在嵌入式平台 Xavier NX 上进行一次对偶更新所需要的平均计算时间。 35
图 2.9 本文算法在嵌入式平台 Xavier NX 上进行一次对偶更新所需要的平均计算时间。 35
图 3.1 投影矩阵、方位误差的示意图。 40
图 3.2 当无人机 1-3 位置确定时，无人机 4 的局部编队性能指标及其梯度示意图。 43
图 3.3 智能体的多项式轨迹示意图。 43
图 3.4 数值仿真中使用的七架无人机组成的球形编队队形。 51
图 3.5 七架无人机编队数值仿真中使用的通讯图。 51
图 3.6 一百次编队保持蒙特卡洛仿真中的一个例子。 53
图 3.7 文献[48] 中提出算法在编队形成上的结果。 54
图 3.8 本文所提出算法在编队形成上的结果。 54
图 3.9 文献[48] 所提出方法在综合场景中的编队导航误差图，无人机相对距离图和轨迹图。 55
图 3.10 本文所提出方法在综合场景中的编队导航误差图，无人机相对距离图和轨迹图。 55

24029602

图 4.1 死锁和分配的示意图。蓝色的箭头代表着协同避碰约束的梯度 $\partial\psi_{i,r}/\partial\boldsymbol{p}_i$，红色和绿色的箭头为局部编队性能指标的梯度 $\partial\psi_{i,f}/\partial\boldsymbol{p}_i$。 57
图 4.2 编队任务分配势博弈的参与者。 63
图 4.3 $N=30, R=1$ 的球形编队队形。 65
图 4.4 任务分配数值仿真：一百次蒙特卡洛仿真中的一个例子。 66
图 4.5 有任务分配时，本文所提出算法在编队形成上的结果。 67
图 4.6 实验平台的各个部件及信息流向图。 68
图 4.7 实物实验中使用的五架无人机组成的箭头编队队形。 68
图 4.8 无任务分配时，本文所提出方法在实验中的编队导航误差图和快照图。 69
图 4.9 有任务分配时，本文所提出方法在实验中的编队导航误差图和快照图。 70

24029602

表格索引

表 2.1 四旋翼无人机全尺寸避障轨迹跟踪算法基准比较 34
表 3.1 算法性能对比：编队队形保持 52
表 3.2 算法性能对比：编队队形形成 53
表 4.1 分布式任务分配算法性能对比 65
表 4.2 算法性能对比：编队队形保持 67
表 4.3 算法性能对比：编队队形形成 67

24029602

算法索引

算法 2.1 全尺寸避障问题的交替方向乘子法。 24
算法 4.1 在第 $k+1$ 次迭代中第 i 个智能体的贪婪编队任务分配算法 61

24029602

第 1 章　绪论

1.1　研究背景与意义

18 世纪，人类进入机器时代。发展至今，信息化与智能化已经深度融合，它在提高生产力水平、丰富物质供给的同时，也在重塑人力与机器力结合的劳动形式和要求，为此各国均在积极布局机器人领域。在我国，机器人已成为国民经济发展和军事力量增强的重要助力。海陆空天各类机器人系统被广泛应用于国防和社会各领域，所执行任务从侦察打击、抢险救灾到安全巡检、智能制造等。随着应用场景的多样化、复杂性和任务综合性，机器人技术的发展由个体向多体演进。多机器人系统相较于个体机器人表现出的强大的环境适应性和鲁棒性，吸引了国内外学者的广泛研究。



(a) 鸟群通过协同飞行保留体力。　(b) 鱼群通过形成编队干扰天敌。

(c) 水面舰艇协同围捕。　(d) 移动机械臂协同运输。

图 1.1　自然界中的群体行为及多机器人在工程领域的应用。

早期人们对于多机器人协同的关注来自于自然界中群体行为，如在迁徙的过程中，鸟群通过协作编队飞行来实现保留体力、鱼群通过编队共同抵御捕食者等等。这些群体行为往往能够产生“一加一大于二”的效果，因此科学家们也希望将这种群体行为引入到实际的工程应用中去。在军事领域，无人舰艇和无人机组成的异构机器人系统，能够完成协同作战、侦察和补给等任务，多无人机系统

24029602

通过蜂群算法可以实现大规模蜂群作战，提高对地攻击效率。在民用领域，多机器人系统被广泛运用于野外位置环境建图[1-2]、灾后救援[3]、协同侦察[4]和环境检测[5]等任务，如麻省理工学院完成了多机器人以几何队形在野外环境构建地图的任务[6]；在卡耐基-梅隆大学研究的搜救机器人团队中，每个机器人能够按照能力上的互补情况时刻保持局部位置上的某种关系，提升了机器人团队在野外搜救任务中的鲁棒性[7-8]。在上述实际应用中，编队控制是多机器人系统完成任务的基石。多机器人编队的目标是通过系统内部机器人之间局部的信息交互，使机器人能够形成并且维持一个特定的相对位置以及相对速度关系[9]，实现队形保持、队形变化和避障等多机器人系统的基本功能。先进的多机器人编队算法不仅能够充分发挥单个机器人的感知、计算和行动能力，而且可以不断提高多机器人系统的应用上限。因此，赋予多机器人编队系统更丰富的任务能力、更稳健的系统鲁棒性和更高等级的安全性，不仅成为了理论研究热点，而且是促进其应用升级改造的关键技术。



(a) 水面舰艇集群在开阔水域的编队。



(b) 固定翼在空中无障碍物环境的编队。



(c) 四旋翼无人机集群在森林中的建图。



(d) 卡车车队在城市道路环境中的队列控制。

图 1.2　多机器人编队的发展趋势：由开阔环境到障碍物密集环境。

传统机器人编队的应用场景有微小卫星在近地轨道的编队、固定翼无人机的高空侦察和水面舰艇在开阔水域的围捕等。在这些场景中，由于可行空间不受约束或仅受到简单约束，机器人一般不会遇到避碰和避障问题。然而随着近些年来传感器和通讯技术的发展，智能多机器人系统应用场景的不断拓展，科学界及工业界对编队技术提出了更高需求。在这种趋势下，多机器人编队的应用场景正逐渐从上述可行区域无约束或仅受简单约束的场景向森林、废墟、城市建筑群等可行空间受到极大约束的复杂场景转换。例如，在森林和废墟中的同时定位与建

24029602

图技术、城市道路场景中的车辆排编队等等。在这些场景中，躲避环境中的障碍物，保证机器人自身的安全性是编队导航的一项重要需求。现有主流的多机器人编队研究侧重于从底层控制的角度出发，根据多机器人的系统数学模型设计控制器实现编队的形成、保持和轨迹跟踪，缺少对环境信息的考虑，因而难以在可行空间受到极大约束的复杂场景中应用；对机器人进行运动规划是处理环境信息的有效方法，然而，适用于个体机器人的规划导航方法因没有考虑相互协同而无法直接应用于多机器人编队中。据作者尽可能全面的调研，将多机器人编队问题与规划导航进行有机融合的文献尚不多见。因此，相对于底层控制，从更高的层次利用规划与优化的思想显式地考虑场景信息，面向多机器人编队建立协同导航框架，在可行空间受到极大约束的场景中快速求解出可行路径，是实现复杂场景下编队的创新思路。

在复杂的环境中的多机器人系统应用，如战场打击、灾后搜救等任务都具有严格的时效性约束。现有规划算法侧重于通过对任务空间的不断地采样或将机器人的构型空间 (Configuration Space) 离散化后进行图搜索来寻找可行路径。这类算法在障碍物较为稀疏的场景中通常可以快速的找到可行解。然而，当需要考虑机器人的具体形状或可行空间受到极大约束时，这类算法往往会因为构型空间维度的升高而导致求解变慢，甚至因为构型空间中的狭窄通道而导致求解失败；再者，多机器人编队的规划本身就属于高维规划，传统规划算法会遇到维数灾难而难以应用。与上述方法不同，基于优化的运动规划算法将机器人的规划问题建模为带约束的优化问题，随后放到成熟的求解框架下进行求解。基于优化的算法通常能够利用性能指标和优化约束的梯度等高层次信息引导搜索的方向，从而加速求解。与传统基于采样和搜索的规划算法相比，基于优化的规划算法具有更快的求解速度，是保证编队导航时效性的有效手段。因此，如何将复杂环境中的规划问题建模为优化问题，并利用优化问题的内在结构及高层次信息进行快速求解，是具有重要现实意义的科学问题。

在复杂场景下的编队导航任务中，需要同时考虑机器人自身的约束、机器人之间的协作约束以及来自环境中的安全约束等等。这些约束之间难免存在冲突，这些冲突往往会使得优化问题陷入到局部最优中，甚至导致优化问题无法求解，从而降低了机器人集群协同的效率。具体到多机器人编队导航任务中，机器人的编队约束和安全约束之间往往会产生冲突，导致机器人集群陷入到死锁当中。引入更高层次的协同方法，如资源调度、任务分配等是消解不同约束之间冲突的有效手段。因此，如何针对多机器人编队任务的特点，在更高的层次对机器人进行协调，设计高效的编队协同算法是亟需解决的实际难题。

基于前述讨论中，本文针对障碍物密集环境设计了一套基于优化的编队导航框架。该框架遵循从单机规划到多机协同，从低层次优化到高层次协调逐步递

24029602

进的原则，提出了单机的精确避障算法，多机编队轨迹优化算法和分布式编队任务分配算法。本文提出的算法旨在推动多智能体编队导航算法在实际场景中的落地，因此从数值仿真、高保真物理引擎仿真和实物实验三方面对所提出算法进行了验证，并给出了与目前学术界前沿编队导航算法的对比结果。

1.2 机器人编队导航研究现状

尽管已有文献中存在着各种各样的单机器人规划算法，但在作者尽可能详尽的文献调研中，尚且缺乏在障碍物密集环境中的实时高精度避障算法。另一方面，对于多机器人的协同编队算法而言，将规划的思想与多机器人编队有机结合的文献尚且不多见。近年来提出的多机器人编队导航指的是利用运动规划的思想实现多机器人的编队，同时在更高的层次对不同机器人进行协调。因此，对于多机器人编队导航的文献调研应从单机器人规划、多机器人编队和多机器人任务分配三方面开展。

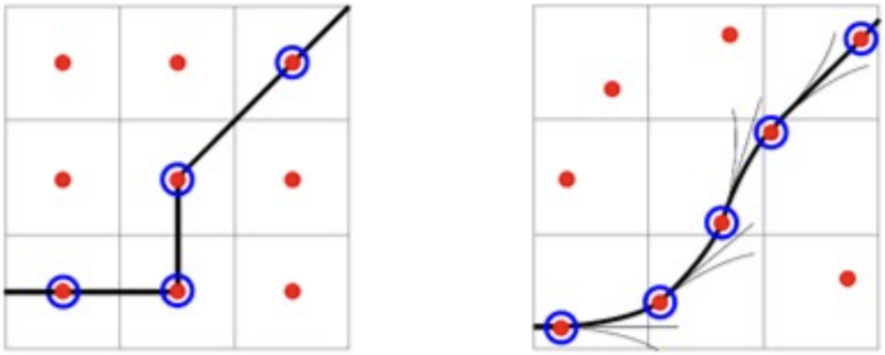
1.2.1 单机器人的运动规划

机器人运动规划的目的为对环境信息进行不断地探索和利用以寻求运动路径的可行解或最优解。常见的运动规划算法可以分为基于图搜索的算法、基于采样的算法和基于优化的算法。

基于图搜索的规划算法: 基于图搜索的算法先将构型空间均匀离散为不同的网格点，并根据这些网格点在空间上的相对位置构建图，随后通过对图的不断搜索来寻求可行路径。例如最著名的图搜索算法，迪克斯特拉算法[10](Dijkstra Algorithm) 解决了初始构型和目标构型之间连接总长度最短的问题。A 星算法[11]在迪克斯特拉算法的基础上加入了启发式策略，通过启发效用函数与累积效用函数对需要扩展的节点进行排序，有限搜索代价函数较低的节点，因而大大减少了需要遍历的节点而提高了搜索效率。一些 A 星算法的变体如 D 星算法[12]，通过在图中添加目标构型返回初始构型路径的思路提供了一种应对动态环境的方法。上述图搜索方法直接在构型空间进行离散，由于没有考虑机器人的运动学方程，生成的状态轨迹通常是不连续的因此无法直接交给到机器人执行，通常还需要进一步对离散的点进行插值。为了解决这个问题，混合 A 星算法[13]将机器人的运动学方程引入到图搜索的过程中。与常见的图搜索方法直接在构型空间进行搜索不同，混合 A 星算法对机器人的控制空间进行离散，并利用控制量对状态空间进行扩展，因此能够保证运动轨迹的连续性。类似的思想被运用到了无人机的 $SE(3)$ 轨迹规划中[14]，通过对无人机的加加速度 (Jerk) 进行离散、搜索，进而

24029602

可以实现无人机在状态空间上的扩展，因此能够找到满足无人机动力学的激进轨迹。基于图搜索的规划算法被广泛应用于自动驾驶汽车、无人机和其他机器人的路径规划中。尽管如此，基于图搜索的规划算法面临着两方面的挑战：首先，当遇到高维构型/控制空间时，搜索节点的数量将会呈指数增加，引发维数灾难；其次，为了保证最终的运动轨迹不过于保守，需要保证足够的构型/控制空间分辨率，然而分辨率的提升同样会引发节点数量的增加。因此，单纯基于图搜索的规划方法不适用于对实时性要求高的场景。



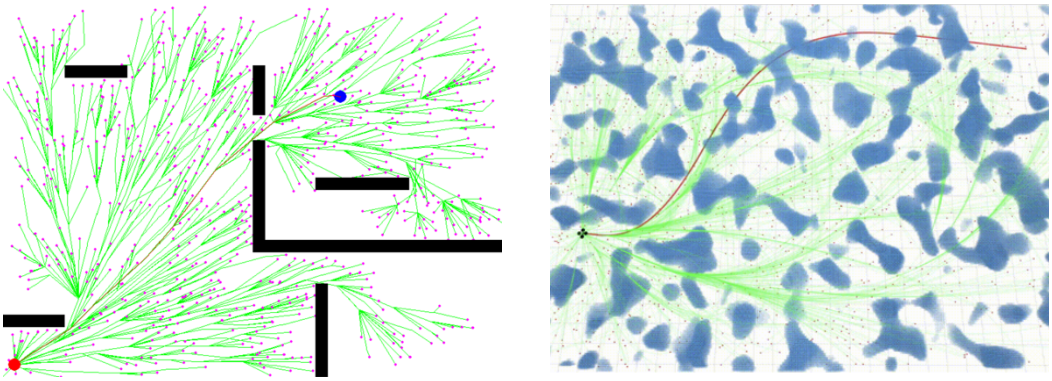
(a) A 星算法。　　(b) 混合 A 星算法。

图 1.3　基于图搜索的规划算法。

基于采样的规划算法: 与基于图搜索的规划算法不同，基于采样的规划算法不对构型空间进行离散，而是直接在构型空间中不断地采样来寻找可行路径。概率路图算法[15](Probabilistic Roadmap, RPM) 利用随机采样得到的可行顶点构建静态环境中的无碰撞轨迹图，随后采用图搜索算法找到可行路径。快速搜索随机树算法[16](Rapidly Exploring Random Tree, RRT) 则通过在线采样可行顶点和可行边来构建搜索树，通过对树进行反向搜索来获取可行路径。上述两种算法均具有概率完备特性，即随着采样个数趋于无限时，这些算法总是能够找到可行的解[17]。文献[18] 探讨了基于采样算法实现渐进最优性的条件，并提出了渐进最优版本的概率路图算法和快速搜索随机树算法，即 PRM* 算法和 RRT* 算法。这两种渐进最优的变体在采样次数趋近无穷时，能够保证当前最优解逐渐收敛至全局最优解。针对基于采样的最优运动规划问题，文献[19] 提出了一种具有智能探索和开发能力的采样框架。该采样框架将智能探索和开发结合，充分利用了当前的路径成本和位置信息，能够减少探索空间，提升了最优运动规划问题的实时性。上述基于采样的规划算法没有考虑机器人的运动学方程，因此不能直接交给机器人执行。为了将机器人的动力学方程纳入到规划中，文献[20] 基于快速搜索随机树算法提出了针对于运动-动力学 (Kino-dynamic) 的增量式采样规划方法。该方法将两个采样点之间的轨迹插值问题建模为两点边值问题 (Two Point Boundary Value Problem, TPBVP)，并用变分法对该最优控制问题进行求解。类似的思想被运用到了具有线性运动学方程的机器人系统上[21]。尽管利用线性系

24029602

统的性质可以加速两点边值问题的求解，这类算法存在难以利用启发信息进行高效采样的缺点，因而通常被视作离线规划算法。此外，由于随机采样的本质，基于采样的规划方法在遇到构型空间中的狭窄可行通道时通常难以发现可行解，直接导致算法失效。然而，在密集的障碍物通常存在大量的狭窄的可行通道，因此，单纯基于采样的规划算法亦不适用本文讨论的场景。



(a) 快速搜索随机树算法。　　(b) 运动-动力学快速搜索随机树算法。

图 1.4　基于采样的规划算法。

基于优化的规划算法: 随着学术界及工业界对于运动轨迹要求的进一步提高，基于优化的规划方法在近些年来受到越来越多的重视。相对基于搜索和采样的算法而言，基于优化的算法在流程上更类似于运动轨迹的后处理方法。该类方法算法更专注于在搜索出粗糙可行解后对机器人的运动轨迹进行更进一步的提升，从而得到高质量局部最优解。基于优化的方法将机器人的运动规划问题建模为最优化问题，然后通过成熟的最优化算法对该最优化问题进行求解。在基于优化的方法中，通常需要给出机器人在执行任务过程中需要优化的指标，如最小化距离、最小化消耗能量等；还需要给定机器人在执行任务的过程中需要满足的约束条件，如运动学方程、执行器的饱和约束和机器人的安全约束等。在多机器人的编队导航中，最重要的需求为保证机器人的安全，即优化问题中的避障约束。因此，在该小节中的文献调研将专注于基于优化的避障技术。

广而言之，基于优化的避障算法可以根据对于被控物体的几何外形建模方法大致分为两类，第一类为质点模型 (Point-mass Model) 的避障算法，第二类为全尺寸模型 (Full-dimensional Model) 的避障算法。由于方法论上的简单性，绝大多数文献[22-24] 侧重于质点模型的避障算法，这类算法将被控物体建模为一个点，随后通过引入膨胀半径的方法来考虑被控物体的几何外形。在这些方法中，障碍物通常被假定为多面体或椭球。对于多面体障碍物，通常可以使用分段优化进行处理，例如文献[25-27]将分段优化问题重构为混合整数规划问题。对于椭球障碍物，避碰约束可以建模为光滑的非凸约束[28-29]，然后通过使用通用非凸优化求解器来解决这些问题。尽管基于质点模型的避障算法得到了广泛的应用，但随着

24029602

机器人应用场景的扩展，该类算法已经难以满足需求。例如，在自动驾驶领域，质点模型的避障算法会导致密集车流中的超车行为过于保守，其解决办法为运用全尺寸模型的避障算法。据作者尽可能详尽的文献调研，除了文献[30-34] 外，在基于优化的方法的背景下对于全尺寸的受控对象做避障的研究尚且不多。文献[31] 通过受控对象的顶点对其进行建模，并在所有涉及的对象都是矩形的假设下，通过确保受控对象的所有顶点都在障碍物外部来实现避碰。文献[30] 提出了一种更一般的处理完整维度受控对象碰撞回避的方法。该方法通过约束被控对象和障碍物之间的符号距离 (Signed Distance) 来实现避障，文章作者还提出了一种序列线性化技术来处理符号距离函数的非凸性。文献[32] 基于符号距离和凸优化问题的强对偶性提出了一种光滑的无碰撞约束。尽管如此，该方法会引入大量的对偶变量，使得该约束难以应用于机器人的实时规划中。针对引入过多对偶变量的问题，现有文献中存在两种方法。文献[33] 提出了一种基于放缩法碰撞检测的隐式无碰撞约束，该方法无需引入对偶变量，但该方法会引入不必要的非光滑性，且隐式约束不利于优化问题的求解。文献[34] 则充分探索引入对偶变量后优化问题的结构特点，利用交替方向乘子法 (Alternating Direction Method of Multiplier, ADMM) 将优化问题分解为一系列子问题从而提高规划问题的求解速度，然而，求解大量的子问题对于嵌入式设备而言依然是困难的。

1.2.2　多机器人编队

目前主要有控制领域和机器人领域的学者在研究多机器人的编队问题。控制领域的学者侧重于研究多机器人编队的编队形成与保持[35-38]、编队整体机动[39-41]和编队轨迹跟踪[42-45]问题。编队形成与保持主要研究的是如何驱动机器人从任意位置出发最终运动到指定的几何队形上，并在未来的时间始终保持该几何队形。文献[35] 通过机器人间的相对位置设计代价函数，随后利用滚动时域控制的框架设计编队镇定控制器，使得机器人能够收敛到给定的相对位置，从而实现编队形成与保持。文献[36] 基于方位刚性理论设计了一种新颖的分布式控制器，该控制器能够使得机器人收敛到指定的相对方位上，从而形成期望的几何形状。由于方位信息与编队的整体位移和放缩无关，机器人集群最后形成的几何形状的中心和放缩比例也是不确定的。编队机动控制则主要研究在形成编队后，如何实现整体平移、整体放缩、整体旋转和整体仿射变换等机动动作。文献[39] 基于领航-跟随者框架，依据方位约束具有平移、放缩不变性的原理为跟随者设计跟踪控制算法。通过设计领航者的轨迹，实现了基于方位信息的编队整体平移和放缩机动，基于这些机动动作可以进一步实现编队的避障任务。文献[40] 则基于复数矩阵的特殊性质实现了二维平面内的放缩、旋转、平移和仿射编队机动控

制算法，然而，由于复数矩阵的局限性该方法仅限于二维平面的情况，难以拓展到三维空间中。与编队机动控制不同，编队轨迹跟踪则更为灵活，其目标为驱动机器人收敛到一个时变的编队上。文献[42] 基于一致性原理为无人飞行器设计了指定时间收敛的时变编队的跟踪控制器。文献[46] 则实现了单纯使用方位信息的编队跟踪控制，由于方位信息可以通过摄像头等传感器获取，该方法具有很好的落地应用潜力。尽管现有文献中存在各种各样的编队控制方法，但这些方法都局限于针对特定的动力学系统、特定的感知信息设计底层控制器使之收敛到给定的目标点或参考轨迹上。然而如何根据外界环境信息，如障碍物信息，地图信息等合理地给定目标点、合理地为领航者给出参考轨迹依然是一个悬而未决的问题，因此这些算法难以直接运用于障碍物密集的环境。

机器人领域的学者主要研究多机器人编队的轨迹生成[47-48]、障碍物躲避[49-54]等内容。编队轨迹生成的主要研究内容为如何在满足特定约束，如机器人运动学方程、机器人自身执行器约束和安全性约束等条件下生成高质量的运动轨迹。作为编队轨迹生成的里程碑式工作，文献[47] 提出了针对一般机器人系统的编队轨迹生成框架。该框架继承了传统轨迹生成方法的结构，分为前端和后端两个部分，前端的主要任务为通过图搜索、采样等方法生成粗糙的编队路径点，后端的主要任务则为通过构建带约束的优化问题为每个机器人生成连续的、光滑的、满足特定任务指标的高质量轨迹。该算法被成功应用于多移动机械臂编队协同搬运物体和多车协同穿过狭窄通道的任务中。文献[48] 同样基于优化的思想为多机器人编队生成高质量轨迹。该文献利用归一化的距离拉普拉斯矩阵提出了一种新颖的编队性能指标—编队相似性误差，该指标相对编队的整体放缩、平移和旋转不敏感，这为编队避障提供了额外的自由度。基于此特点，文献将编队相似性指标、安全指标、机器人执行器约束指标等加权后同时进行优化，实现了障碍物密集场景中的编队导航。然而，由于编队相似性误差的计算需要所有机器人的状态信息，这对集群网络的通信能力提出了极高要求，因而难以拓展到大规模集群上。另一方面，编队相似性误差存在极强的非线性，并没有理论可以保证优化该指标能够保证集群一定收敛到期望的轨迹上。编队障碍物躲避主要的研究内容为如何设计策略使得在机器人与机器人之间不发生碰撞、机器人与障碍物之间不发生碰撞的前提下形成编队。文献[50] 分布式模型预测控制设计编队控制器，通过向优化问题中添加相对位置约束来实现编队，通过约束无人机之间的距离、无人机与障碍物之间的距离实现避障。文献[52] 基于虚拟结构设计了一种分层的编队避障框架，在上层通过控制虚拟结构的轨迹来实现编队的整体机动，在下层通过人工势场来实现编队队形的保持和局部的障碍物躲避。文献[53] 为无人机设计了一种基于规则的编队协同避障方案，通过为可能发生碰撞的机器人重新规划目标点实现了无碰撞的编队轨迹生成。然而，该方法基于

24029602

规则的设计方法过于复杂，难以推广到通用的机器人编队任务中。上述编队避障算法更多倾向于将单个机器人的避障问题与多个机器人的编队问题分层处理或分为两个子问题处理，如何充分利用编队的特点为编队避障提供更多的自由度，利用编队机动实现避障仍然是一个开放的问题。

1.2.3　多机器人任务分配

多机器人任务分配问题是一个典型的组合优化问题，其应用范围不仅限于多机器人系统领域，还包括任务调度和运营管理。匈牙利算法[55]是一种应用广泛的集中式任务分配算法，该算法需要知道所有可能的机器人执行任意任务的成本。然而，在实际的多机器人任务分配算法中，每个机器人通常只能知道自己执行任意任务的成本，即只能知道本地信息，这使得匈牙利算法难以运用。现有文献存在一些匈牙利算法的分布式变体[56-57]，但是与集中式匈牙利算法一样，这些分布式变体仅适用于具有线性目标成本函数的问题，这使得这类算法难以拓展到大多数实际的任务分配应用场景中。

另一种常见的方法是基于拍卖方法，此类方法通过设计一个市场结构来协调机器人与邻居之间的行为，并通过每个机器人最小化自身局部代价函数来实现任务的分配。通常，基于拍卖的算法需要一个中央节点来管理拍卖，每个拍卖的参与者向中央节点出价，随后中央节点按照一定的规则将任务分配给一个中标者[58-61]。因此，这些方法要求所有机器人都与中央节点进行通信。另一些方法从机器人集群中指定一个拍卖仲裁者，该拍卖仲裁者同时也是一个竞标者[62-63]，拍卖仅在该拍卖仲裁者与其邻居竞标者之间进行[64-65]。在其他方法中，每个机器人都充当拍卖仲裁者，每一个机器人都根据接收到的邻居出价迭代地更新所有任务的分配，直到所有任务都分配完毕[66-68]。尽管拍卖方法相对简单，可拓展性强，但这类方法通常不能提供所有任务的最优分配方案[69-70]。

尽管多机器人的任务分配问题已经得到了充分的研究，然而针对编队中的任务分配问题尚且不多。编队中的任务分配通过优化特定的性能指标，如编队误差、到目标的总距离等，来实现集群中死锁的消解、机器人间碰撞的避免和编队队形收敛速度的提升。文献[71] 基于匈牙利算法和粒子群优化算法来解决多无人机的编队变换问题，并找到了无人机运动总距离最小意义下的最优分配。然而，该算法是一个集中式的算法，且计算复杂度较高，难以运用在对实时性要求较高的场景。在文献[72] 中，作者将基于距离的编队控制器与分布式任务分配结合起来，实现了编队的全局渐近镇定控制。为了解决由避碰引起的死锁，文献[54] 采用基于一致性的拍卖算法[66]来解决编队控制的目标分配问题。该算法可以在一定迭代次数内可以找到无冲突分配方案，然而该算法只能找到次优解，

24029602

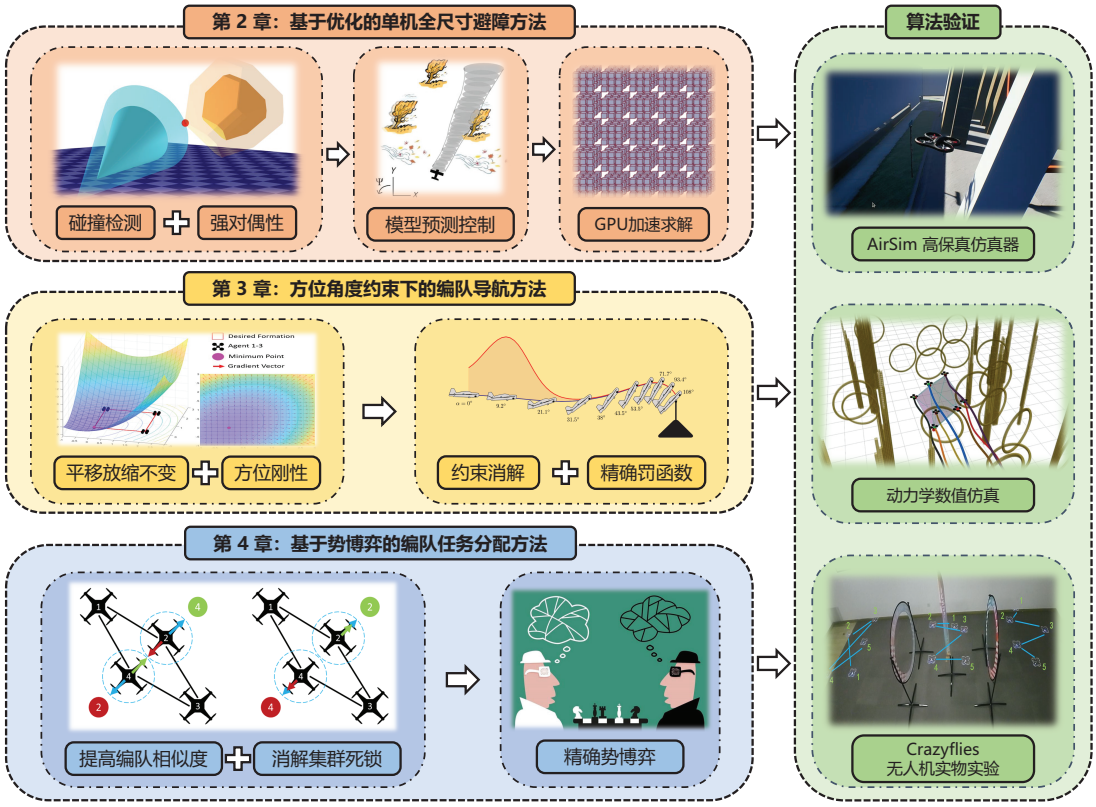


图 1.5　本文的研究内容及技术路线图。

且该解的质量最优只能达到理论值的 1/2。上述大多任务分配算法依赖同步执行，这在实际应用中可能难以实现。在文献[73] 中，作者提出了一种用于编队形成的任务交换算法，该算法具有较低的计算复杂性，可以异步实现。

1.3　本文研究内容及主要贡献

本文面向障碍物密集场景下的编队导航问题开展研究工作。本文的研究内容遵循从单机器人到多机器人，从低层次优化到高层次协同的递进顺序，将障碍物密集环境中的编队导航问题拆分为单机器人全尺寸避障、多机器人分布式协同编队轨迹优化和编队任务分配三个主要科学问题。为了验证研究结果的有效性，本文在高保真物理仿真引擎、无人机实物实验平台上对所提出算法进行了验证。本文具体研究内容及结构安排如下：

第 1 章为绪论，给出多机器人编队导航的研究背景、研究意义以及未来的发展趋势。进而详细介绍了当前学术界在单机器人规划、多机器人编队及多机器人任务分配三个领域的现有工作。最后介绍了本文的研究内容、技术路线及主要贡献。

第 2 章主要研究单机器人在障碍物密集环境中的实时避障算法。该章节以凸多胞体的碰撞检测方法为切入点，在充分考虑机器人几何外形各向异性的前

24029602

提下，基于凸优化的强对偶性为单机器人设计光滑、解析且非保守的避障约束，进一步利用模型预测控制的思想将单机器人的避障问题转换为优化问题。针对优化问题非凸、维度过高而导致无法实时求解的难题，该章节充分利用优化问题的内在结构，引入交替方向乘子法将高维问题拆分为多个低维度的二次规划问题；引入序列二次规划方法将优化问题目标函数中的非凸部分近似为凸函数，实现优化问题的实时求解。最后以无人机作为案例研究对象，介绍了该章所提出算法在具体机器人中的使用方法，并通过 AirSim 高保真仿真引擎验证了所提出算法的有效性和实时性。

第 3 章主要研究多机器人在障碍物密集环境中的编队轨迹优化算法。该章节以编队约束的描述方式为切入点，利用方位约束具有平移和放缩不变性的特点，基于方位刚性理论为机器人集群设计一种新颖的分布式编队性能指标。为了系统地考虑编队性能指标、机器人的避障约束和机器人的相互避碰约束，该章节提出了一种无约束的在线轨迹优化算法。该算法以多项式轨迹作为优化变量，通过变量消解消去受约束变量，使得求解出的轨迹始终满足轨迹的连续性要求。该算法利用精确罚函数法将约束转换为惩罚函数，使得受约束问题最终转换为无约束的优化问题，实现轨迹优化问题的在线求解。最后，通过障碍物密集环境中的编队保持、编队形成和综合场景中的编队导航三个方面的仿真实验，验证该章所提出算法的有效性和优越性。

第 4 章主要研究编队导航中的任务分配算法。针对单纯使用多机器人编队轨迹优化可能导致机器人集群陷入死锁的问题，该章节引入高层次协同的思想设计一种完全分布式的任务分配算法。针对编队任务分配问题为二次分配问题而难以求解的挑战，该章节借助博弈论的思想，将编队任务分配问题转换为精确势博弈问题，并给出了寻找精确势博弈问题纳什均衡解的方法。随后，该章节从理论上证明了算法能够在有限次迭代内收敛到纳什均衡策略，并证明了该算法能有效提升编队的初始相似度、消解死锁。最后，通过基准比较证明了该章节所提出算法的有效性；通过于轨迹优化方法的联合仿真及 Crazyflies 无人机实物验证证明了引入高层次协同能够有效提升编队导航的成功率。

第 5 章对全文的主要工作进行了总结概括，并基于现有方法框架和仿真、实验结果对编队导航的未来研究方向进行展望。本文研究内容的技术路线如图 1.5 所示，主要贡献总结如下：

首先，本文基于放缩法碰撞检测和凸优化的强对偶性提出了一种适用于光滑、显式和非保守的全尺寸无碰撞约束方法；全尺寸避障优化难以实时求解的问题，基于交替方向乘子法提出了一种适用于嵌入式设备的实时求解方法，实现了单机器人在障碍物密集环境中的实时避障。

其次，本文基于方位刚性理论提出了一种分布式的编队性能指标；创新地将

24029602

环境信息纳入到编队导航的算法中，提出了一种能够系统考虑编队性能指标、机器人避障需求和机器人间协同避碰需求的轨迹优化算法，实现了障碍物密集环境中的实时编队导航。

最后，本文通过引入机器人间的高层次协同，即任务分配，给出了消解编队导航中死锁问题的一个可行算法，并从理论上证明了该算法的有效性，从仿真和实验中证明了算法能够提升编队导航的成功率。

24029602

第 2 章　基于优化的单机全尺寸避障方法

2.1　引言

随着自动化技术的快速发展，将自主智能无人系统部署在具有障碍物的复杂环境中变得越来越普遍。例如，自动驾驶汽车在城市道路上导航[74-75]和自主四旋翼无人机在障碍物密集的森林地区自主飞行[24,76]。在这些情境中，实现高精度的避障对确保导航的安全性至关重要。

在现有文献已经提出了各种方法来实现自主智能无人系统的避障，其中包括基于搜索的方法[14]、基于采样的方法[19]和基于优化的方法[32-34,77]。基于搜索的方法将构型空间离散化为具有定义分辨率的网格，并系统地搜索这些网格以找到可行解。基于采样的方法则采用各种各样的采样算法在构型空间中搜索可行解。虽然基于搜索和基于采样的方法在实际的应用场景中得到了广泛的使用和验证，但在需要精确考虑机器人和障碍物几何形状的情况下，这些算法会面临巨大的计算开销，因而通常无法做到实时求解。这在四轴飞行器进行复杂全姿态规划以通过狭窄间隙[77]或在密集交通流中超车[33]的情况中尤为明显。

另一方面，基于优化的方法在满足动力学、运动学和安全约束的前提下，通过优化给定性能指标来实现安全导航。近期的研究[32-34,77]已经展示了基于优化的方法在高精度避障的需求下优势。此外，基于优化的方法，如模型预测控制[32,34] (Model Predictive Control, MPC) 和轨迹优化[33,77] (Trajectory Optimization, TO) 在某些性能指标下天然地能够找到更好的轨迹。尽管具有这些良好的特性，现有基于优化的方法仍然面临着以下三个挑战：首先，如何在通用的求解框架下精确地描述避障约束依然是很困难的；其次，随着障碍物数量的增加，优化问题的维度显著增加；最后，由于系统动态约束和避碰约束的非凸性，优化问题通常也是非凸的。这些特性使得基于优化的方法难以在嵌入式设备上实现实时导航，从而阻碍了其在实际场景中的使用。

为了解决这些挑战，本章节提出了一种基于优化的单体全尺寸避障框架。被控对象和障碍物的几何形状均被建模为有限个凸多胞体的并集，该描述方法能够满足绝大多数情景中对于避障精度的要求。此外，所提出的框架充探索并利用了优化问题的内在结构，并进一步利用 GPU 的计算能力加速求解过程。

24029602

2.2 最优化理论预备知识

考虑具有如下形式的一般约束优化问题，

$$\min_{\boldsymbol{x}} \quad p(\boldsymbol{x}) \tag{2.1a}$$

$$s.t. \quad g_i(\boldsymbol{x}) \leq 0, \quad i = 1, \cdots, m, \tag{2.1b}$$

$$h_j(\boldsymbol{x}) = 0, \quad j = 1, \cdots, q, \tag{2.1c}$$

其中，优化变量 $\boldsymbol{x} \in \mathbb{R}^n$。注意，标量代价函数 $p(\boldsymbol{x})$ 以及标量不等式约束函数 $g_i(\boldsymbol{x})$ 不一定为凸函数，标量等式约束函数 $h_j(\boldsymbol{x})$ 不一定为仿射函数。设优化问题的定义域是非空集合，优化问题的最优值为 $p^\star$，则有以下重要定义及定理。

定义 2.1 (拉格朗日函数)： 定义优化问题 (2.1) 的拉格朗日函数 $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}$ 为:

$$L(\boldsymbol{x}, \lambda, \mu) = p(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) + \sum_{j=1}^{q} \mu_j h_j(\boldsymbol{x}), \tag{2.2}$$

其中，称 λ_i 为第 i 个不等式约束 $g_i(\boldsymbol{x}) \leq 0$ 的拉格朗日乘子；相似地，称 μ_j 为第 j 个等式约束 $h_j(\boldsymbol{x}) = 0$ 的拉格朗日乘子。

定义 2.2 (拉格朗日对偶函数)： 定义优化问题 (2.1) 的拉格朗日对偶函数 $d : \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}$ 为其拉格朗日函数关于优化变量 $\boldsymbol{x}$ 取得的逐点最小值：

$$d(\lambda, \mu) = \inf_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda, \mu) = \inf_{\boldsymbol{x}} \left(p(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) + \sum_{j=1}^{q} \mu_j h_j(\boldsymbol{x}) \right). \tag{2.3}$$

如果拉格朗日函数 L 对于优化变量 $\boldsymbol{x}$ 无下界，则拉格朗日对偶函数取负无穷。由于对偶函数是关于拉格朗日乘子 (λ, μ) 的放射函数的逐点下确界，所以无论原优化问题是否为凸优化问题，对偶函数 $d(\lambda, \mu)$ 总为凹函数。

性质 2.1： 拉格朗日对偶函数构成了原问题最优值 $p^\star$ 的一个下界。使用更准确的数学语言描述：对于 $\forall \lambda \in \mathbb{R}^m$ 和 $\forall \mu \in \mathbb{R}^q$，且 $\lambda \succeq \mathbf{0}$ 以下不等式成立:

$$d(\lambda, \mu) \leq p^\star. \tag{2.4}$$

定义 2.3 (拉格朗日对偶问题)： 定义原问题 (2.1) 的拉格朗日对偶问题为：

$$\max_{\lambda,\, \mu} \quad d(\lambda, \mu) \tag{2.5a}$$

$$s.t. \quad \lambda \succeq \mathbf{0}. \tag{2.5b}$$

对偶问题是一个凸优化问题，因为以下原因。

24029602

- 该问题的目标函数为凹函数，最大化凹函数等同于最小化一个凸函数；
- 该问题的约束为放射不等式约束。

对偶问题的凸性与原问题的凸性无关。即无论原问题是否为凸优化问题，对偶问题始终为凸优化问题。

定理 2.1 (凸优化问题的强对偶性)： 若原问题 (2.1) 为凸优化问题，且存在一点 $\boldsymbol{x}$ 使得以下条件成立

$$g_i(\boldsymbol{x}) < 0, \quad i = 1, \cdots, m, \qquad h_j(\boldsymbol{x}) = 0, \quad j = 1, \cdots, q.$$

设对偶问题 (2.5) 的最优值为 $d^\star$。则以下等式成立，

$$d^\star = p^\star. \tag{2.6}$$

并称之为强对偶性成立。强对偶性成立说明拉格朗日对偶函数 $d(\lambda, \mu)$ 能够给出原问题的一个紧的下界。

2.3　单机避障优化问题的建模与求解

2.3.1　基于模型预测控制的优化问题建模

假设机器人受到以下运动学或动力学约束，

$$\boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{u}_t), \tag{2.7}$$

其中 $\boldsymbol{s}_t \in \mathbb{R}^{n_s}$ 为机器人在 t 时刻的状态变量，$\boldsymbol{u}_t \in \mathbb{R}^{n_u}$ 为机器人在 t 时刻的控制变量，函数 $\boldsymbol{f} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_s}$ 描述了机器人的运动学或动力学方程。注意，本文不对函数 $\boldsymbol{f}$ 做出具体的假设，$\boldsymbol{f}$ 既可以是线性的，也可以是非线性的。

除此运动学与动力学约束之外，机器人通常还会受到由自身机械结构限制而带来的状态与控制约束。例如，地面车辆的可能会受到的前轮转角限制、最大引擎力矩约束、最大刹车摩檫力约束；固定翼无人机为了防止失速而需要保证最小飞行速度；旋翼无人机则会受到最大倾斜角度、最大推力限制等约束。这些约束在本章中被统一写为：

$$\boldsymbol{g}(\boldsymbol{s}_t, \boldsymbol{u}_t) \preceq 0, \tag{2.8}$$

其中，约束函数 $\boldsymbol{g} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_g}$，而其中 n_g 是不等式约束的数量。本文同样不对约束函数 $\boldsymbol{g}$ 的凸性做出假设，但要求 $\boldsymbol{g}$ 是连续光滑的。

在机器人导航的过程中，最重要的约束为安全约束，即保证机器人与障碍物之间没有碰撞。与常规质点模型的避障直接设置避障半径不同，全尺寸避障需要

24029602

考虑机器人与障碍物的准确几何建模。在本文中，使用 $\mathbb{E}(\boldsymbol{s}_t)$ 表示机器人在状态 $\boldsymbol{s}_t$ 时所占据的几何空间。特别的，本文假设 $\mathbb{E}(\boldsymbol{s}_t)$ 能够表示为一系列实心紧致凸集合的并集，且每个实心紧致凸集都被建模为与机器人状态变量 $\boldsymbol{s}_t$ 有关的多胞体。使用更准确的数学语言描述，

$$\mathbb{E}(\boldsymbol{s}_t) = \mathbb{E}_1(\boldsymbol{s}_t) \cup \mathbb{E}_2(\boldsymbol{s}_t) \cdots \cup \mathbb{E}_N(\boldsymbol{s}_t), \tag{2.9a}$$

$$\mathbb{E}_i(\boldsymbol{s}_t) = \{\boldsymbol{y} \in \mathbb{R}^d \mid \boldsymbol{y} = \mathbf{R}_i(\boldsymbol{s}_t)\boldsymbol{x} + \rho_i(\boldsymbol{s}_t),\ \ \mathbf{A}_i\boldsymbol{x} \preceq \boldsymbol{b}_i\}, \tag{2.9b}$$

其中，$d \in \{2,\ 3\}$ 为空间的维数，$\mathbb{E}_i(\boldsymbol{s}_t)$ 为机器人的第 i 个部件在状态 $\boldsymbol{s}_t$ 所占据的集合空间。$\boldsymbol{x}$ 为部件 i 局部坐标系下的点，$\boldsymbol{y}$ 为惯性系下的点。$\mathbf{R}_i(\boldsymbol{s}_t) \in SO(d)$ 和 $\rho_i(\boldsymbol{s}_t) \in \mathbb{R}^d$ 分别为部件 i 局部坐标系到惯性系的旋转矩阵和平移向量，矩阵 $\mathbf{A}_i \in \mathbb{R}^{n_{ri} \times d}$ 和向量 $\mathbf{b}_i \in \mathbb{R}^{n_{ri}}$ 则为与部件 i 几何形状相关的参数。注意，本文假设旋转矩阵 $\mathbf{R}_i$ 与平移向量 ρ_i 为状态变量 $\boldsymbol{s}_t$ 的函数。

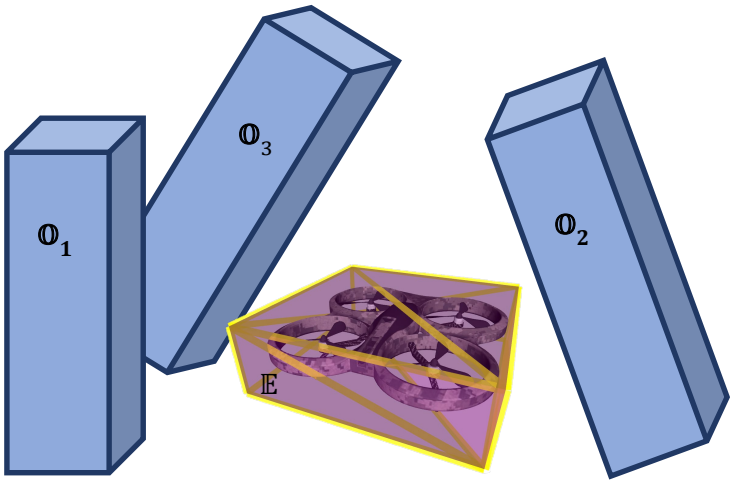


图 2.1　无人机与障碍物的几何表示。

类似的，本文使用 $\mathbb{O}$ 来表示静态障碍物所占据的几何空间，并假设其可以表示为一系列实心紧致凸集合的并集，且每个实心紧致凸集都被建模为与障碍物形状和位置有关的多胞体。具体而言，

$$\mathbb{O} = \mathbb{O}_1 \cup \mathbb{O}_2 \cdots \cup \mathbb{O}_M, \tag{2.10a}$$

$$\mathbb{O}_j = \{\boldsymbol{y} \in \mathbb{R}^d \mid \mathbf{C}_j\boldsymbol{y} \preceq \boldsymbol{d}_j + \boldsymbol{e}_j\}, \tag{2.10b}$$

其中，$\mathbb{O}_i$ 为障碍物的第 i 个部件。矩阵 $\mathbf{C}_j \in \mathbb{R}^{n_{oi} \times d}$ 和向量 $\mathbf{d}_i \in \mathbb{R}^{n_{oi}}$ 为与障碍物部件 i 几何形状相关的参数，向量 $\mathbf{e}_i \in \mathbb{R}^{n_{oi}}$ 为与障碍物部件 i 种子点有关的参数。因为任何紧致集合都能够被有限个凸集合近似，所以上述几何建模是一般的，能够描述绝大多数几何形状。如无人机与障碍物的示意图可见图 2.1。

模型预测控制是一种常用的规划方法。模型预测控制通过预测未来 T 个时刻的系统状态来生成控制序列，并同时在满足动力学约束，可行性约束和安全性

24029602

约束的前提下优化特定的指标。模型预测控制的另一个特点是其滚动优化的原则，即每次生成 T 个时刻的控制序列，但仅有 $T_u \leq T$ 个控制序列会被真正交给系统执行。当 T_u 个控制序列被执行完后，再重新求解优化问题，生成新的控制序列。模型预测控制的滚动优化原则使之能够不断地依据环境中新的信息来重新规划控制序列，这使得其在动态且具有不确定性的环境中十分有效。本文基于模型预测控制设计机器人的全尺寸避障方法，本章主要考虑如下最优控制问题。

问题 2.1 (单机器人全尺寸避障问题)： 给定机器人的初始状态 $\boldsymbol{s}_s$，终末状态 $\boldsymbol{s}_f$，以及在移动过程中需要优化的目标函数 $\mathcal{J} = \sum_{t=1}^{T} l(\boldsymbol{s}_i, \boldsymbol{u}_i)$，其中 $l : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}$ 为阶段目标函数。求解满足动力学约束，可行性约束及安全约束的最优状态序列 $\boldsymbol{s} = \{\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_T\}$ 与最优控制序列 $\boldsymbol{u} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_{T-1}\}$。即，

$$\min_{\boldsymbol{s},\, \boldsymbol{u}} \quad \sum_{t=1}^{T} l(\boldsymbol{s}_t, \boldsymbol{u}_t) \tag{2.11a}$$

$$s.t. \quad \boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{u}_t), \quad \forall t, \tag{2.11b}$$

$$\boldsymbol{g}(\boldsymbol{s}_t, \boldsymbol{u}_t) \preceq 0, \quad \forall t, \tag{2.11c}$$

$$\mathbb{E}(\boldsymbol{s}_t) \cap \mathbb{O} = \emptyset, \quad \forall t, \tag{2.11d}$$

$$\boldsymbol{s}_0 = \boldsymbol{s}_s, \qquad \boldsymbol{s}_T = \boldsymbol{s}_f. \tag{2.11e}$$

其中 (2.11d) 为无碰撞约束，(2.11e) 为终端约束。

上述问题的关键点在于如何建模无碰撞约束 (2.11d)，该部分内容将会在下一小节中进行详细描述。

2.3.2　精确无碰撞约束的建模，解析表示与几何解释

为了保证机器人与障碍物之间无碰撞，需要保证机器人的每个部件与障碍物的每个部件在每一个时刻都不发生碰撞，即

$$\mathbb{E}_i(\boldsymbol{s}_t) \cap \mathbb{O}_j = \emptyset, \quad \forall i, j, t. \tag{2.12}$$

特别的，在本文中，每一个部件都被建模为多胞体，因此需要考虑两个多胞体之间的碰撞检测问题，一个常用的多胞体检测方法为放缩法[78]。

该方法在两个多胞体之中各选取一个种子点 (seed point)，然后以种子点为中心将两个多胞体按照相同的比例进行放缩，使得两个多胞体恰好发生重叠，并记此时的放缩比例为 $\alpha^\star$。如果 $\alpha^\star > 1$, 则两个多胞体没有发生碰撞；如果 $\alpha^\star \leq 1$，则两个多胞体之间发生了碰撞。该方法的几何示意图如图 2.2 所示。

假设机器人的部件和障碍物的部件分别由 (2.9b) 和 (2.10b) 给出，则上述方法可以建模为以下线性规划问题，

24029602

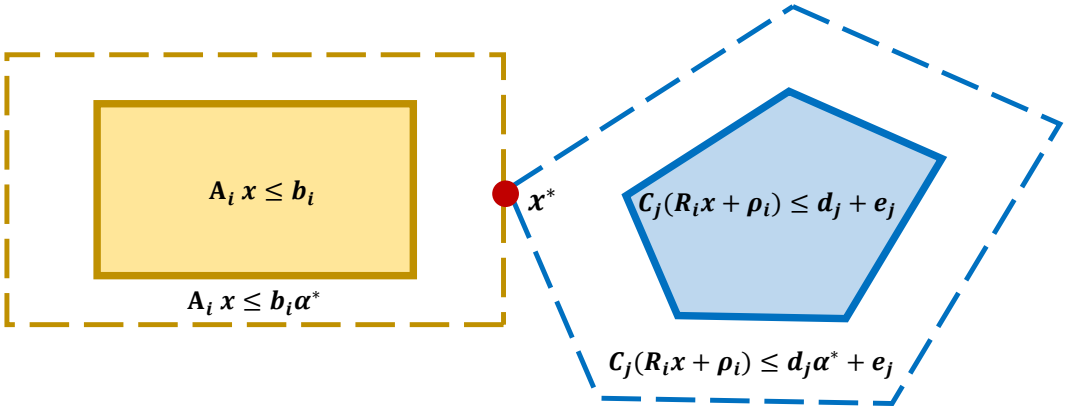


图 2.2　放缩法碰撞检测的几何示意图。

$$\min_{\alpha,\, \boldsymbol{x},\, \boldsymbol{y}} \quad \alpha \tag{2.13a}$$

$$s.t. \quad \mathbf{A}_i \boldsymbol{x} - \boldsymbol{b}_i \alpha \le 0, \tag{2.13b}$$

$$\boldsymbol{y} = \mathbf{R}_i \boldsymbol{x} + \rho_i, \tag{2.13c}$$

$$\mathbf{C}_j \boldsymbol{y} - \boldsymbol{d}_j \alpha - \boldsymbol{e}_j \le 0. \tag{2.13d}$$

将变量 $\boldsymbol{y}$ 消去，得到以下等价的线性规划问题。

$$\min_{\alpha,\, \boldsymbol{x}} \quad \alpha \tag{2.14a}$$

$$s.t. \quad \mathbf{A}_i \boldsymbol{x} - \boldsymbol{b}_i \alpha \le 0, \tag{2.14b}$$

$$\mathbf{C}_j \mathbf{R}_i \boldsymbol{x} - \boldsymbol{d}_j \alpha + \mathbf{C}_j \rho_i - \boldsymbol{e}_j \le 0. \tag{2.14c}$$

记线性规划问题 (2.14) 的最优值为 $\alpha^\star$。为了保证两个部件之间没有碰撞，则需要将约束 $\alpha^\star > 1$ 加入到优化问题 (2.11) 中。然而，由于 (2.14) 是一个约束优化问题，其最优值 $\alpha^\star$ 一般不存在解析表达式。这个特点使之难以应用于通用的优化框架中，主要的问题有两方面：

1. 为了使用基于梯度的优化方法，需要求解最优值 $\alpha^\star$ 关于参数 $(\mathbf{R}_i, \rho_i)$ 的梯度。然而，当无法得到最优值 $\alpha^\star$ 的解析表达式时，这些梯度难以求解。
2. 问题 (2.11) 通常为非凸优化问题，为了加速最优控制问题的求解，需要使用基于凸近似的方法。然而当无法得到最优值 $\alpha^\star$ 的解析表达式时，难以找到合适的方法对 $\alpha^\star$ 进行凸近似。

因此，找到一个与 $\alpha^\star > 1$ 等价的，且具有解析表达式的约束形式是十分必要的。一个基于强对偶性给出的等价显式约束由以下命题给出：

命题 2.1： 假设机器人的部件 i 的几何形状由 (2.9b) 给出，障碍物的部件 j 的几

24029602

何形状由 (2.10b) 给出。记问题 (2.14) 的最优值为 $\alpha^\star$，以及需要满足的最小安全方所比例为 $\alpha_{min} > 0$, 则约束，

$$\alpha^\star > \alpha_{min}, \tag{2.15}$$

等价于以下约束，

$$\mu^T(\mathbf{C}_j\rho_i - \boldsymbol{e}_j) > \alpha_{min}, \tag{2.16a}$$

$$\boldsymbol{b}_i^T\lambda + \boldsymbol{d}_j^T\mu - 1 = 0, \tag{2.16b}$$

$$\mathbf{A}_i^T\lambda + (\mathbf{C}_j\mathbf{R}_i)^T\mu = \mathbf{0}, \tag{2.16c}$$

$$\lambda \succeq \mathbf{0}, \qquad \mu \succeq \mathbf{0}, \tag{2.16d}$$

其中列向量 $\lambda \in \mathbb{R}^{n_{ri}}$ 和 $\mu \in \mathbb{R}^{n_{oj}}$ 分别为因不等式约束 (2.14b) 和 (2.14c) 而引入的拉格朗日乘子。

证明： 问题 (2.14) 的拉格朗日对偶函数为：

$$\begin{aligned} d(\lambda,\ \mu) &= \inf_{\alpha,\ \boldsymbol{x}} \lambda^T(\mathbf{A}_i\boldsymbol{x} - \boldsymbol{b}_i\alpha) + \mu^T(\mathbf{C}_j\mathbf{R}_i\boldsymbol{x} - \boldsymbol{d}_j\alpha + \mathbf{C}_j\rho_i - \boldsymbol{e}_j) \\ &= \inf_{\alpha}\left\{(\boldsymbol{b}_i^T\lambda + \boldsymbol{d}_j^T\mu)\alpha\right\} + \inf_{\boldsymbol{x}}\left\{\left(\mathbf{A}_i^T\lambda + (\mathbf{C}_j\mathbf{R}_i)^T\mu\right)^T\boldsymbol{x}\right\} + \\ &\quad \mu^T(\mathbf{C}_j\rho_i - \boldsymbol{e}_j). \end{aligned} \tag{2.17}$$

因此，可以得到问题 (2.14) 的拉格朗日对偶问题：

$$\max_{\lambda,\ \mu} \qquad \mu^T(\mathbf{C}_j\rho_i - \boldsymbol{e}_j) \tag{2.18a}$$

$$s.t. \qquad \boldsymbol{b}_i^T\lambda + \boldsymbol{d}_j^T\mu - 1 = 0, \tag{2.18b}$$

$$\mathbf{A}_i^T\lambda + (\mathbf{C}_j\mathbf{R}_i)^T\mu = \mathbf{0}, \tag{2.18c}$$

$$\lambda \succeq \mathbf{0}, \qquad \mu \succeq \mathbf{0}, \tag{2.18d}$$

由于线性规划问题是凸优化问题，具有强对偶性。记对偶问题的最优解为 $(\lambda^\star,\ \mu^\star)$，则根据定理 2.1，以下关系式成立：

$$\mu^T(\mathbf{C}_j\rho_i - \boldsymbol{e}_j) \ \leq\ (\mu^\star)^T(\mathbf{C}_j\rho_i - \boldsymbol{e}_j)\rho_i \ =\ \alpha^\star. \tag{2.19}$$

该关系式揭露了 $\alpha^\star$ 的一个紧的下界 $\mu^T(\mathbf{C}_j\rho_i - \boldsymbol{e}_j)$，约束 $\alpha^\star$ 可通过约束其紧下界 $\mu^T(\mathbf{C}_j\rho_i - \boldsymbol{e}_j)$ 来间接实现。证明完毕。 □

注意到与约束 (2.15) 不同，约束 (2.16) 相对于参数 $(\mathbf{R}_i, \rho_i)$ 具有解析的表达式，因此相关的梯度可以较为容易地求解。另外，当参数 $(\mathbf{R}_i, \rho_i)$ 固定时，约束 (2.16) 相对于拉格朗日乘子 (λ, μ) 是线性的；当拉格朗日乘子 (λ, μ) 固定时，约

24029602

束 (2.16) 相对于参数 $(\mathbf{R}_i, \rho_i)$ 也是线性的，该性质称为双线性。双线性对于问题的加速求解有着重要的作用，将会在下一小节中进一步讨论。

尽管无碰撞约束 (2.16) 的代数形式是复杂且抽象的，其背后蕴含的几何原理却是简明且易于理解的。为了提高后续章节的可读性，有必要提供该约束的几何解释。

命题 2.2： 假设存在 $\alpha_{min} > 0$ 使得约束 (2.16) 成立，则必定存在至少一个超平面 $\boldsymbol{n}^T\boldsymbol{x} = b$ 严格分离放缩 α_{min} 倍后的机器人部件 i 和障碍物部件 j。即，$\forall \boldsymbol{y}_i \in \mathbb{E}_i(\alpha_{min})$ 和 $\forall \boldsymbol{y}_j \in \mathbb{O}_j(\alpha_{min})$，其中，

$$\mathbb{E}_i(\alpha_{min}) = \{\boldsymbol{y} \mid \boldsymbol{y} = \mathbf{R}_i(\boldsymbol{s}_t)\boldsymbol{x} + \rho_i(\boldsymbol{s}_t),\ \mathbf{A}_i\boldsymbol{x} \preceq \boldsymbol{b}_i\alpha_{min}\}, \tag{2.20a}$$

$$\mathbb{O}_j(\alpha_{min}) = \{\boldsymbol{y} \mid \mathbf{C}_j\boldsymbol{y} \preceq \boldsymbol{d}_j\alpha_{min} + \boldsymbol{e}_j\}, \tag{2.20b}$$

存在方向向量 $\boldsymbol{n}$ 和截距 b 以下不等式关系成立，

$$\boldsymbol{n}^T\boldsymbol{b}_i \geq b_i > b > b_j \geq \boldsymbol{n}^T\boldsymbol{y}_j. \tag{2.21}$$

超平面的方向向量 $\boldsymbol{n}$ 由下式给出：

$$\boldsymbol{n} = -\mathbf{R}_i\mathbf{A}_i^T\lambda = \mathbf{C}_j^T\mu, \tag{2.22}$$

截距 b_i 和 b_j 由下式给出：

$$b_i = -\boldsymbol{b}_i^T\lambda \cdot \alpha_{min} - \rho_i^T\mathbf{R}_i\mathbf{A}_i^T\lambda, \tag{2.23a}$$

$$b_j = \boldsymbol{d}_j^T\mu \cdot \alpha_{min} + \boldsymbol{e}_j^T\mu. \tag{2.23b}$$

而截距 b 则可以为截距 b_i 和 b_j 之间的任意值。

证明： 由于约束 $\lambda \succeq \mathbf{0}$ 和 (2.20a)，以下不等式成立，

$$\lambda^T\mathbf{R}_i\mathbf{A}_i^T\boldsymbol{y}_i \leq \lambda^T\boldsymbol{b}_i \cdot \alpha_{min} + \lambda^T\mathbf{R}_i\mathbf{A}_i^T\rho_i, \tag{2.24}$$

类似地，由约束 $\mu \succeq \mathbf{0}$ 和 (2.20b)，可以得到以下不等式，

$$\mu^T\mathbf{C}_j\boldsymbol{y}_j \leq \mu^T\boldsymbol{d}_j \cdot \alpha_{min} + \mu^T\boldsymbol{e}_j. \tag{2.25}$$

再将等式约束 (2.16b) 两边乘以 α_{min} 并消去不等式约束 (2.16a) 中的 α_{min}，即可得到以下不等式，

$$\mu^T\mathbf{C}_j\rho_i - \mu^T\boldsymbol{e}_j > (\boldsymbol{b}_i^T\lambda + \boldsymbol{d}_j^T\mu) \cdot \alpha_{min},$$

整理上式，并由等式约束 (2.16c) 可得,

$$-\lambda^T\boldsymbol{b}_i \cdot \alpha_{min} + \lambda^T\mathbf{R}_i\mathbf{A}_i^T\rho_i\ >\ \mu^T\boldsymbol{d}_j \cdot \alpha_{min} + \mu^T\boldsymbol{e}_j. \tag{2.26}$$

24029602

最后整理不等式 (2.24)，(2.25) 和 (2.26) 可得，

$$\begin{aligned} -\lambda^T \mathbf{R}_i \mathbf{A}_i^T \boldsymbol{y}_i \geq & -\lambda^T \boldsymbol{b}_i \cdot \alpha_{min} + \lambda^T \mathbf{R}_i \mathbf{A}_i^T \rho_i > \\ & \mu^T \boldsymbol{d}_j \cdot \alpha_{min} + \mu^T \boldsymbol{e}_j \geq \quad \mu^T \mathbf{C}_j^T \boldsymbol{y}_j. \end{aligned} \tag{2.27}$$

由等式约束 (2.16c) 可取 $\boldsymbol{n} = -\mathbf{R}_i \mathbf{A}_i^T \lambda = \mathbf{C}_j^T \mu$，随即可以得到结论 (2.21)，(2.22) 和 (2.23)。证明完毕。 □

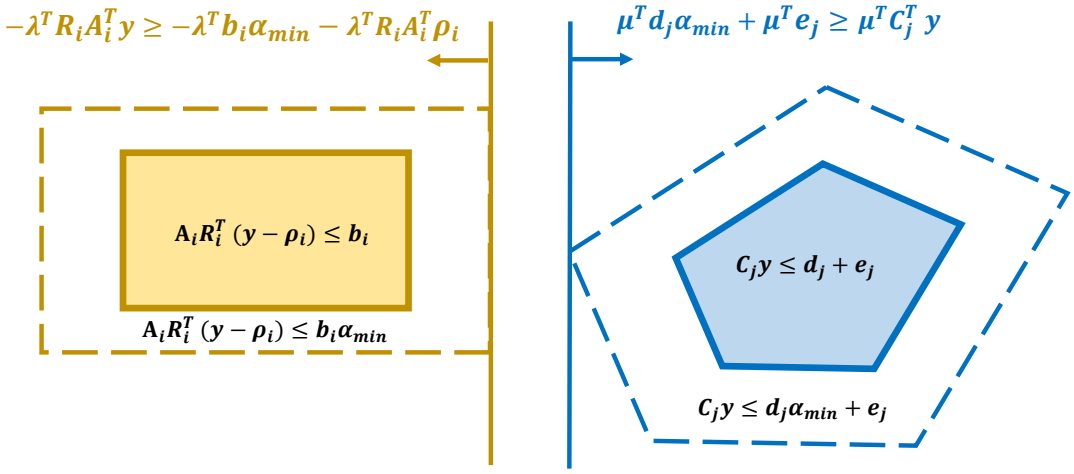


图 2.3　无碰撞约束 (2.16) 的几何原理解释。

命题 2.2 的结论为凸分析理论中的重要定理，分离超平面定理[79]，在放缩法碰撞检测视角下的特例。关系式 (2.21) 中的第一个不等式给出了放缩 α_{min} 倍后的机器人部件 i 的一个支撑超平面[80]。相似地，第四个不等式给出了放缩 α_{min} 倍后的障碍物部件 j 的一个支撑超平面。而第二和第三个严格不等式保证了两个支撑超平面之间存在间隙。命题 2.2 的几何示意如图 2.3 所示。

2.3.3　GPU 辅助的优化问题实时求解

基于上一小节中提出的无碰撞约束，最优控制问题 (2.11) 可以进一步写为，

$$\min_{\boldsymbol{s},\,\boldsymbol{u}} \quad \sum_{t=1}^{T} \Big(l(\boldsymbol{s}_t, \boldsymbol{u}_t) + \mathfrak{P}(\alpha_t) \Big) \tag{2.28a}$$

$$s.t. \quad \boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{u}_t), \quad \forall t, \tag{2.28b}$$

$$\boldsymbol{g}(\boldsymbol{s}_t, \boldsymbol{u}_t) \preceq 0, \quad \forall t, \tag{2.28c}$$

$$\mu_{ijt}^T (\mathbf{C}_j \rho_i(\boldsymbol{s}_t) - \boldsymbol{e}_j) \geq \alpha_{it}, \quad \forall i, j, t, \tag{2.28d}$$

$$\mathbf{A}_i^T \lambda_{ijt} + (\mathbf{C}_j \mathbf{R}_i(\boldsymbol{s}_t))^T \mu_{ijt} = \mathbf{0}, \quad \forall i, j, t, \tag{2.28e}$$

$$\boldsymbol{b}_i^T \lambda_{ijt} + \boldsymbol{d}_j^T \mu_{ijt} - 1 = 0, \quad \forall i, j, t, \tag{2.28f}$$

$$\lambda_{ijt} \succeq \mathbf{0}, \qquad \mu_{ijt} \succeq \mathbf{0}, \quad \forall i, j, t, \tag{2.28g}$$

24029602

$$\underline{\alpha} \leq \alpha_{it} \leq \overline{\alpha}, \quad \forall i, t, \tag{2.28h}$$

$$\boldsymbol{s}_0 = \boldsymbol{s}_s, \qquad \boldsymbol{s}_T = \boldsymbol{s}_f. \tag{2.28i}$$

其中 $\alpha_t = (\alpha_{1t}, \alpha_{2t}, \cdots, \alpha_{Nt})$ 为所有部件在 t 时刻放缩比例的堆叠向量。约束 (2.28d) - (2.28g) 为上一小节中提出的无碰撞约束。约束 (2.28h) 为对放缩比例的上下界约束，其中 $\underline{\alpha}$ 是为了保证安全性引入的放缩下界，$\overline{\alpha}$ 是为了数值稳定而引入的放缩上界。正则函数 $\mathcal{P}(\alpha)$ 定义为，

$$\mathcal{P}(\alpha_t) = -\sum_{i=1}^{N} |\alpha_{it}| = -\sum_{i=1}^{N} \alpha_{it}, \tag{2.29}$$

其作用为了使得每个部件的放缩比例尽可能大，从而提供更大的分离间隙。

直接使用通用非线性优化求解器求解优化问题 (2.28) 很难保证实时性，主要的挑战有以下两方面。

1. 优化问题 (2.28) 的优化变量维度和约束数量正比于预测时域 T，机器人部件数量 N 和障碍物部件数量 M。当 T, N 或者 M 增大时，优化问题 (2.28) 的维度会显著增加，因此在障碍物密集的环境中难以保证实时性；
2. 由于函数 $\boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{u}_t)$, $\rho_i(\boldsymbol{s}_t)$ 和 $\mathbf{R}_i(\boldsymbol{s}_t)$ 通常是非线性函数，因此约束 (2.28b)，(2.28d) 和 (2.28e) 通常是非凸约束。这些非凸约束会导致优化问题的求解时间难以得到保证。

本文通过将问题 (2.28) 分解为一系列低维度可处理的优化问题来解决第一个挑战。具体而言，注意到具有不同的下标 (i, j, t) 的变量 $(\lambda_{ijt}, \mu_{ijt})$ 是通过状态变量 $\boldsymbol{s}_t$ 间接耦合起来的。因此，当固定状态变量 $\boldsymbol{s}_t$ 时，这些变量将会被相互解耦，从而可以利用多线程技术进行并行求解，保证实时性。

为此，本文使用交替方向乘子法来处理这个耦合约束。首先，将不等式约束 (2.28d) 转化为等式约束，

$$-\mu_{ijt}^T(\mathbf{C}_j\rho_i(\boldsymbol{s}_t) - \boldsymbol{e}_j) + \alpha_{it} + \gamma_{ijt} = 0, \quad \forall i, j, t, \tag{2.30}$$

其中 $\gamma_{ijt} \geq 0$ 为引入的松弛变量。

24029602

考虑问题 (2.28) 的增广拉格朗日函数，

$$
\begin{aligned}
&\mathscr{L}_\sigma(\{\boldsymbol{s}_t,\boldsymbol{u}_t,\alpha_t\},\ \{\lambda_{ijt},\mu_{ijt},\gamma_{ijt}\},\ \{\xi_{ijt},\zeta_{ijt}\})\\
&=\sum_{t=1}^{T}\ \Big(l(\boldsymbol{s}_t,\boldsymbol{u}_t)+\mathscr{P}(\alpha_t)+\mathscr{I}_{\boldsymbol{f}}(\boldsymbol{s}_{t+1},\boldsymbol{s}_t,\boldsymbol{u}_t)+\mathscr{I}_{\boldsymbol{g}}(\boldsymbol{s}_t,\boldsymbol{u}_t)+\mathscr{I}_\alpha(\alpha_t)\Big)+\\
&\quad\frac{\sigma}{2}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M}\Big\|-(\mathbf{C}_j\rho_i(\boldsymbol{s}_t)-\boldsymbol{e}_j)^T\mu_{ijt}+\alpha_{it}+\gamma_{ijt}+\xi_{ijt}\Big\|_2^2+\\
&\quad\frac{\sigma}{2}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M}\Big\|\mathbf{A}_i^T\lambda_{ijt}+(\mathbf{C}_j\mathbf{R}_i(\boldsymbol{s}_t))^T\mu_{ijt}+\zeta_{ijt}\Big\|_2^2+\\
&\quad\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M}\mathscr{I}_{ijt}(\lambda_{ijt},\mu_{ijt}),
\end{aligned}
\tag{2.31}
$$

其中 $\sigma>0$ 为惩罚系数，$\{\xi_{ijt},\zeta_{ijt}\}$ 为放缩后的拉格朗日乘子[81]；函数 $\mathscr{I}_{\boldsymbol{f}}(\boldsymbol{s}_{t+1},\boldsymbol{s}_t,\boldsymbol{u}_t)$，$\mathscr{I}_{\boldsymbol{g}}(\boldsymbol{s}_t,\boldsymbol{u}_t)$ 和 $\mathscr{I}_\alpha(\alpha_t)$ 分别为约束 (2.28b)，(2.28c) 和 (2.28h) 的指示函数；函数 $\mathscr{I}_{ijt}(\lambda_{ijt},\mu_{ijt})$ 则为约束 (2.28f) 和 (2.28g) 的指示函数。

交替方向乘子法是一种解决优化问题的迭代算法，在每一次迭代中，通过解子问题更新部分优化变量。特别地，针对增广拉格朗日函数 (2.31)，每次迭代需要依次求解三个子问题，因此有三步更新。以下介绍在第 $k+1$ 次迭代中的更新步骤，使用上标 k 表示第 k 次迭代的，所有带上标的变量都被看作是常量。

更新步骤一：避障对偶变量更新

在第 $k+1$ 次迭代的第一步中，对避障有关的对偶变量 $\{\lambda_{ijt},\mu_{ijt},\gamma_{ijt}\}$ 进行更新。即求解以下优化问题，

$$
\{\lambda_{ijt}^{k+1},\mu_{ijt}^{k+1},\gamma_{ijt}^{k+1}\}=\underset{\{\lambda_{ijt},\mu_{ijt},\gamma_{ijt}\}}{\arg\min}\ \mathscr{L}_\sigma(\{\boldsymbol{s}_t^k,\boldsymbol{u}_t^k,\alpha_t^k\},\{\lambda_{ijt},\mu_{ijt},\gamma_{ijt}\},\{\xi_{ijt}^k,\zeta_{ijt}^k\}).
\tag{2.32}
$$

更新步骤二：状态变量与控制变量更新

在第 $k+1$ 次迭代的第二步中，对状态变量，控制变量与放缩比例 $\{\boldsymbol{s}_t,\ \boldsymbol{u}_t,\ \alpha_t\}$ 进行更新。即求解以下优化问题即求解以下优化问题，

$$
\{\boldsymbol{s}_t^{k+1},\boldsymbol{u}_t^{k+1},\alpha_t^{k+1}\}=\underset{\{\boldsymbol{s}_t,\boldsymbol{u}_t,\alpha_t\}}{\arg\min}\ \mathscr{L}_\sigma(\{\boldsymbol{s}_t,\boldsymbol{u}_t,\alpha_t\},\{\lambda_{ijt}^{k+1},\mu_{ijt}^{k+1},\gamma_{ijt}^{k+1}\},\{\xi_{ijt}^k,\zeta_{ijt}^k\}).
\tag{2.33}
$$

更新步骤三：拉格朗日乘子更新

在第 $k+1$ 次迭代的第三步中，按照以下规则对拉格朗日乘子 $\{\xi_{ijt},\zeta_{ijt}\}$ 进行更新，

$$
\xi_{ijt}^{k+1}=\xi_{ijt}^k-(\mathbf{C}_j\rho_i(\boldsymbol{s}_t^{k+1})-\boldsymbol{e}_j)^T\mu_{ijt}^{k+1}+\alpha_{it}^{k+1}+\gamma_{ijt}^{k+1},
\tag{2.34a}
$$

$$
\zeta_{ijt}^{k+1}=\zeta_{ijt}^k+\mathbf{A}_i^T\lambda_{ijt}^{k+1}+(\mathbf{C}_j\mathbf{R}_i(\boldsymbol{s}_t^{k+1}))^T\mu_{ijt}^{k+1}.
\tag{2.34b}
$$

24029602

算法 2.1　全尺寸避障问题的交替方向乘子法。

```
while k ≤ k_max do
    求解优化问题 (2.32) 更新对偶变量 {λ_ijt, μ_ijt, γ_ijt};
    求解优化问题 (2.33) 更新状态变量 {s_t, u_t, α_t};
    根据规则 (2.34) 更新拉格朗日乘子 {ξ_ijt, ζ_ijt};
    k ← k + 1;
    if 终止准则 (2.35) 满足 then
        跳出循环 ;
    end
end
```

交替方向乘子法迭代停止准则

最后，通过判断以下条件判断是否应该终止整个交替方向乘子法的迭代，

$$\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\left\|\zeta_{ijt}^{k+1}-\zeta_{ijt}^{k}\right\|_2^2+\left\|\xi_{ijt}^{k+1}-\xi_{ijt}^{k}\right\|_2^2\right)\leq\epsilon_{pri}, \tag{2.35a}$$

$$\sum_{t=1}^{T}\left(\left\|\boldsymbol{s}_t^{k+1}-\boldsymbol{s}_t^{k}\right\|_2^2+\left\|\boldsymbol{u}_t^{k+1}-\boldsymbol{u}_t^{k}\right\|_2^2+\left\|\alpha_t^{k+1}-\alpha_t^{k}\right\|_2^2\right)\leq\epsilon_{dual}, \tag{2.35b}$$

其中，常量 $\epsilon_{pri}>0$ 为控制约束 (2.28d) - (2.28e) 可行性的阈值；常量 $\epsilon_{dual}>0$ 则为控制问题最优性的阈值。在实际使用算法时，需要根据对最优性与可行性的取舍来选取合适的阈值 ϵ_{pri} 和 ϵ_{dual}。上述优化求解流程被总结于算法 2.1 中。

注意到，在子优化问题 (2.32) 中，所有拥有不同下标 (i,j,t) 的变量都是解耦的。因此这个优化问题可以进一步分解为一系列低维的问题，并且这些低维问题可以并行求解。具体而言，这些低维优化问题可以写为具有如下凸二次规划形式的优化问题：

$$\min_{\boldsymbol{y}_{ijt}}\quad\left\|\mathbf{K}_{ijt}^{T}\boldsymbol{y}_{ijt}+\beta_{ijt}\right\|_2^2 \tag{2.36a}$$

$$s.t.\quad\kappa_{ijt}^{T}\boldsymbol{y}_{ijt}=\eta_{ijt}, \tag{2.36b}$$

$$\boldsymbol{y}_{ijt}\succeq\mathbf{0}, \tag{2.36c}$$

其中参数的具体表达式为：

$$\boldsymbol{y}_{ijt}=(\lambda_{ijt},\mu_{ijt},\gamma_{ijt}), \tag{2.37a}$$

$$\mathbf{K}_{ijt}=\begin{bmatrix}\mathbf{0} & \mathbf{A}_i\\ \boldsymbol{e}_j-\mathbf{C}_j\rho_i(\boldsymbol{s}_t^k) & \mathbf{C}_j\mathbf{R}_i(\boldsymbol{s}_t^k)\\ 1 & \mathbf{0}\end{bmatrix}, \tag{2.37b}$$

24029602

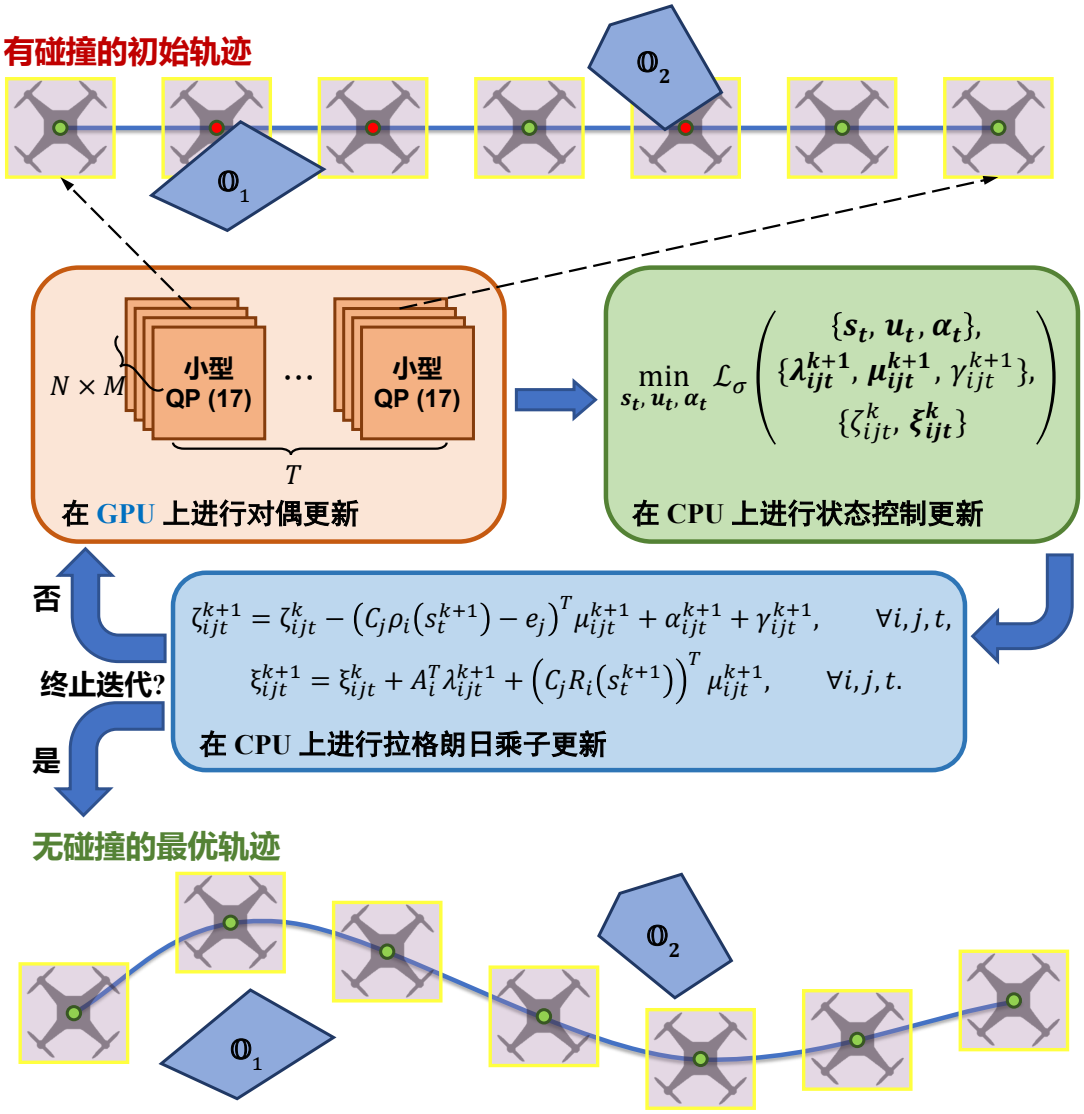


图 2.4　全尺寸避障问题的交替方向乘子法的整体流程图。

$$\beta_{ijt} = (\alpha_{it}^k + \xi_{ijt}^k,\ \zeta_{ijt}^k), \tag{2.37c}$$

$$\kappa_{ijt} = (\boldsymbol{b}_i,\ \boldsymbol{d}_j,\ 0), \qquad \eta_{ijt} = 1. \tag{2.37d}$$

这些二次规划问题总共有 $N \times M \times T$ 个，求解这些二次规划问题是整个交替方向乘子法更新的瓶颈。本文通过采用平行计算的方法消除这个计算瓶颈。具体而言，通过在图形处理器 (Graphics Processing Unit, GPU) 实现的并行版本莱姆克算法[82] (Lemke's Algorithm) 对这些二次规划问题进行求解。为了使用莱姆克算法，需要首先将这些二次规划问题转换为线性互补问题[83] (Linear Complementarity Problem, LCP)，转换总共需要两步。

为了简单起见，暂时将下标 (i, j, t) 省去。转换的第一步为将二次规划问题的线性等式约束 (2.36b) 转换为不等式约束，考虑将参数矩阵、参数向量和优化

24029602

变量按照如下方式分块，

$$\boldsymbol{y}=\begin{bmatrix}\boldsymbol{y}_e\\ \boldsymbol{y}_u\end{bmatrix},\qquad \mathbf{K}=\begin{bmatrix}\mathbf{K}_e\\ \mathbf{K}_u\end{bmatrix},\qquad \kappa=\begin{bmatrix}\kappa_e\\ \kappa_u\end{bmatrix}, \tag{2.38a}$$

其中 κ_e 为适当维度的可逆方阵。根据，等式 (2.36b) 可以得到以下等式：

$$\boldsymbol{y}_e=\kappa^{-T}(\eta-\kappa_u^T\boldsymbol{y}_u), \tag{2.39}$$

将 $\boldsymbol{y}_e$ 带回到原优化问题 (2.36) 中，可以得到以下新的二次规划问题，

$$\min_{\boldsymbol{y}_u}\qquad \left\|\mathbb{K}^T\boldsymbol{y}_u+\tilde{\boldsymbol{\beta}}\right\|_2^2 \tag{2.40a}$$

$$s.t.\qquad \tilde{\boldsymbol{\kappa}}^T\boldsymbol{y}_u=\tilde{\boldsymbol{\eta}}_{ijt}, \tag{2.40b}$$

$$\boldsymbol{y}_u\succeq\mathbf{0}, \tag{2.40c}$$

其中参数的具体表达式为：

$$\mathbb{K}=\mathbf{K}_u-\kappa_u\kappa_e^{-1}K_e,\quad \tilde{\beta}=\beta+(\kappa_e^{-1}K_e)^T\eta, \tag{2.41a}$$

$$\tilde{\kappa}=\kappa_u\kappa_e^{-1},\quad \tilde{\eta}=\kappa_e^{-T}\eta, \tag{2.41b}$$

第一步转换到此完成。

第二步转换需将上述带等式约束的二次规划问题转换为线性互补问题。考虑带等式约束的二次规划问题 (2.40) 的卡罗需-库恩-塔克 (Karush-Kuhn-Tucker, KKT) 条件，

$$\mathbb{K}\mathbb{K}^T\boldsymbol{y}_u+\mathbb{K}\tilde{\beta}+\tilde{\kappa}\phi-\psi=\mathbf{0}, \tag{2.42a}$$

$$\tilde{\kappa}^T\boldsymbol{y}_u-\tilde{\boldsymbol{\eta}}+\varepsilon=\mathbf{0}, \tag{2.42b}$$

$$\phi^T\varepsilon=0,\quad \psi^T\boldsymbol{y}_u=0, \tag{2.42c}$$

$$\phi\succeq\mathbf{0},\quad \varepsilon\succeq\mathbf{0},\quad \boldsymbol{y}_u\succeq\mathbf{0},\quad \psi\succeq\mathbf{0}, \tag{2.42d}$$

上述条件等价于以下线性互补问题：

$$\begin{aligned}\boldsymbol{w}&=M\boldsymbol{z}+\boldsymbol{q},\quad \boldsymbol{w}^T\boldsymbol{z}=0,\\ \boldsymbol{w}&\succeq\mathbf{0},\quad \boldsymbol{z}\succeq\mathbf{0},\end{aligned} \tag{2.43}$$

其中参数的具体表达式为：

$$\begin{aligned}\boldsymbol{z}&=\begin{pmatrix}\boldsymbol{y_u}\\ \phi\end{pmatrix},\quad \boldsymbol{w}=\begin{pmatrix}\psi\\ \varepsilon\end{pmatrix},\\ M&=\begin{pmatrix}\mathbb{K}\mathbb{K}^T & \tilde{\boldsymbol{\kappa}}\\ -\tilde{\boldsymbol{\kappa}}^T & \mathbf{0}\end{pmatrix},\quad \boldsymbol{q}=\begin{pmatrix}\mathbb{K}\tilde{\boldsymbol{b}}\\ \tilde{\boldsymbol{\eta}}\end{pmatrix}.\end{aligned} \tag{2.44}$$

24029602

自此，第一个挑战已被解决。整体的算法流程框架如图 2.4 所示，首先，按照每个时刻与每个障碍物构建低维度的二次规划问题以更新对偶变量；其次，根据更新后的对偶变量信息对机器人的状态、控制变量进行调整；随后，计算约束违反程度，更新拉格朗日乘子；最后，检查生成轨迹是否满足约束条件，是否可以退出迭代循环。

对于第二个挑战，本文通过对函数 $\boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{u}_t)$, $\rho_i(\boldsymbol{s}_t)$ 和 $\mathbf{R}_i(\boldsymbol{s}_t)$ 进行凸近似来加速求解。具体而言，使用序列二次规划 (Sequential Quadratic Programming, SQP) 方法对非凸问题进行求解，即，对于代价函数使用凸二次型进行近似，对于约束函数则使用仿射函数进行近似。为了保证算法的实时性本文采用实时迭代策略，即每次迭代只做一次凸二次型进行近似。这部分内容取决于机器人的具体类型，将会在下一小节的案例研究中进行详细地推导。

2.4　四旋翼无人机全尺寸避障案例研究

作为机器人导航的关键模块，本文提出的高精度避障算法框架需要在各种各样的导航任务和求解任务中进行大量的仿真、对比和实验才能验证该算法的有效性和优越性。因此，本章节将以四旋翼无人机在障碍物稠密环境中的无碰撞 **SE**(3) 轨迹跟踪作为案例研究，介绍所提出高精度避障算法在具体机器人上的具体的公式推导和算法部署流程。

与无人机在障碍物稀疏环境中的避障不同，无人机通常不能以平稳的轨迹通过障碍物稠密的环境，而是可能需要通过大机动以倾斜的姿态躲避障碍物。因此，在障碍物稠密的环境中，不能将无人机的几何外形简单粗糙地建模为一个球，而是需要显式地考虑无人机几何外形的各向异性；同时，为了保证导航算法的实时性，也不应将过多的计算资源消耗在无人机的几何外形的描述和碰撞检测上。于是，使用凸多胞体对无人机的几何外形进行较为精确的近似成为一个可行的方案。

2.4.1　四旋翼无人机的微分平坦模型

学术界提出了多种四旋翼无人机的动力学模型，如由经典力学中牛顿欧拉方程推导出的非线性动力学模型、在水平稳定姿态做线性近似得到的线性时变动力学模型以及微分平坦动力学模型。在这些动力学模型各有特点，适用于不同的工作条件和任务需求。本文为了不将额外的非线性约束引入到为优化问题 (2.28) 中而更倾向于使用线性动力学模型。然而，由于 **SE**(3) 轨迹跟踪通常需要大幅偏离水平稳定姿态，因此在水平稳定姿态附近做线性近似得到的线性时变

24029602

动力学模型通常会导致模型预测的失效，因而违背了模型预测控制的原则。另一方面，四旋翼无人机的微分平坦模型具有线性的特点，并且能够无损地表达无人机的动力学。虽然需要利用非线性变换将平坦状态及其有限阶导数转换为无人机的实际控制输入，但这种模型更适应于本文提出的全尺寸避障框架。

在正式介绍无人机的微分平坦模型之前，有必要对微分平坦的基本概念及微分平坦系统的特点进行介绍。

定义 2.4 (微分平坦)： 考虑以下一般连续非线性系统的在状态空间中的表达：

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{f}(\boldsymbol{s}(t), \boldsymbol{u}(t)), \tag{2.45}$$

其中 $\boldsymbol{s} \in \mathbb{R}^{n_s}$ 为状态变量，$\boldsymbol{u} \in \mathbb{R}^{n_u}$ 为控制变量以及 $\boldsymbol{f} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_s}$ 为一个光滑的映射。如果存在一个与控制变量维度相同的变量 $\boldsymbol{z} \in \mathbb{R}^{n_u}$ 使得状态变量 $\boldsymbol{s}$ 和控制变量 $\boldsymbol{u}$ 能够被 $\boldsymbol{z}$ 及其各阶导数表示，即，

$$\boldsymbol{s} = \Psi_s(\boldsymbol{z}, \dot{\boldsymbol{z}}, \cdots, \boldsymbol{z}^{(n_f-1)}), \tag{2.46a}$$

$$\boldsymbol{u} = \Psi_u(\boldsymbol{z}, \dot{\boldsymbol{z}}, \cdots, \boldsymbol{z}^{(n_f)}), \tag{2.46b}$$

其中，$\Psi_s : \mathbb{R}^{(n_u)\cdot(n_f-1)} \to \mathbb{R}^{n_s}$ 和 $\Psi_s : \mathbb{R}^{(n_u)\cdot(n_f)} \to \mathbb{R}^{n_u}$ 为光滑的映射，那么称系统 (2.45) 为微分平坦的，并称 $\boldsymbol{z}$ 为平坦输出。

微分平坦系统具有以下特点：一方面，如果能够给定一条足够光滑的平坦输出轨迹 $\boldsymbol{z}(t)$，那么就可以在不对原系统微风方程进行任何积分运算的情况下，直接通过式子 (2.46a) 和 (2.46b) 分别计算出原系统的状态轨迹 $\boldsymbol{s}(t)$ 和控制轨迹 $\boldsymbol{u}(t)$; 另一方面，微分平坦性质提供了一种对原系统进行精确反馈线性化的思路: 将 $\boldsymbol{z}$ 视作原系统在微分平坦空间的状态量，并定义微分平坦空间中的虚拟控制量 $\nu = \boldsymbol{z}^{(n_f)}$，则原状态空间系统可以视作以下微分平坦空间中线性系统在函数 Ψ_s 和 Ψ_u 映射下的像。

$$\begin{aligned} &\frac{d}{dt}\boldsymbol{z}^{(i)} = \boldsymbol{z}^{(i+1)}, \quad i = 0, 1, \cdots, n_f - 1, \\ &\frac{d}{dt}\boldsymbol{z}^{(n_f-1)} = \nu. \end{aligned} \tag{2.47}$$

该系统的 n_u 个分量是完全解耦的，且每一个分量都是一个 n_f 阶的积分器，因此是完全可控的。通过在微分平坦空间利用线性系统理论设计虚拟控制器 ν，再通过式子 (2.46b) 将系统的实际控制量 $\boldsymbol{u}$ 解算出来并作用到原系统，就可以实现对原系统在状态空间的控制。

接下来介绍四旋翼无人机的状态空间模型、微分平坦模型及两者之间的转换关系。四旋翼无人机的状态为:

$$\boldsymbol{s} = \{\rho, \boldsymbol{v}, \mathbf{R}, \omega\} \in \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3, \tag{2.48}$$

24029602

其中 ρ 为无人机在惯性系中的位置，$\boldsymbol{v}$ 为无人机在惯性系中的速度，$\mathbf{R}$ 为机体坐标系相对于惯性系的旋转矩阵，而 ω 则为无人机在机体坐标系中的角速度。四旋翼无人机的控制量为：

$$\boldsymbol{u} = \{f, \tau\} \in \mathbb{R}_+ \times \mathbb{R}^3, \tag{2.49}$$

其中 f 为四旋翼无人机四个螺旋桨产生推力的幅值，τ 则为四个螺旋桨产生在三个轴产生的扭矩。无人机在状态空间中的动力学方程由下式给出，

$$\dot{\rho} = \boldsymbol{v}, \tag{2.50a}$$

$$m\dot{\boldsymbol{v}} = \mathbf{R}f\boldsymbol{e}_3 - mg\boldsymbol{e}_3, \tag{2.50b}$$

$$\dot{\mathbf{R}} = \mathbf{R}\omega^{\wedge}, \tag{2.50c}$$

$$\mathbf{M}\dot{\omega} = \tau - \omega \times \mathbf{M}\omega, \tag{2.50d}$$

其中 $m \in \mathbb{R}_+$ 为无人机的质量，$\mathbf{M} \in \mathbb{R}_+^{3\times3}$ 为无人机的转动惯量，$g \in \mathbb{R}_+$ 为重力加速度，向量 $\boldsymbol{e}_3 = (0, 0, 1)$ 。四旋翼无人机的平坦输出为：

$$\boldsymbol{z} = \{\rho, \psi\} \in \mathbb{R}^3 \times SO(2). \tag{2.51}$$

其中 ψ 为机体坐标系相对于惯性系的偏航角。下面介绍微分平坦空间到状态空间的转换关系。对于无人机的位置和速度，显然有：

$$\rho = \rho, \quad \boldsymbol{v} = \dot{\rho}, \quad \dot{\boldsymbol{v}} = \ddot{\rho}. \tag{2.52}$$

而无人机的总推力为：

$$f = \|m\dot{\boldsymbol{v}} + mg\boldsymbol{e}_3\|, \tag{2.53}$$

进一步，定义中间变量：

$$\boldsymbol{x}_b = \mathbf{R}e_1, \quad \boldsymbol{y}_b = \mathbf{R}e_2,$$

其中向量 $\boldsymbol{e}_1 = (1, 0, 0)$ 和向量 $\boldsymbol{e}_2 = (0, 1, 0)$，而 $\boldsymbol{x}_b$ 与 $\boldsymbol{y}_b$ 的几何意义为机体坐标系的 x 轴和 y 轴。将上述中间变量与 (2.50b) 求内积可得，

$$\begin{aligned}(\mathbf{R}\boldsymbol{e}_i)^T m\dot{\boldsymbol{v}} &= (\mathbf{R}\boldsymbol{e}_i)^T(\mathbf{R}f\boldsymbol{e}_3 - mg\boldsymbol{e}_3)\\ &= f\boldsymbol{e}_i^T\boldsymbol{e}_3 - mg(\mathbf{R}\boldsymbol{e}_i)^T\boldsymbol{e}_3\\ &= -mg(\mathbf{R}\boldsymbol{e}_i)^T\boldsymbol{e}_3.\end{aligned}$$

即对于机体坐标系的 x 轴和 y 轴，有以下两个方程,

$$\boldsymbol{x}_b^T(m\dot{\boldsymbol{v}} + mg\boldsymbol{e}_3) = 0, \quad \boldsymbol{y}_b^T(m\dot{\boldsymbol{v}} + mg\boldsymbol{e}_3) = 0,$$

24029602

这两个方程表明，无人机推力与重力的合力 $(m\dot{\boldsymbol{v}}+mg\boldsymbol{e}_3)$ 与机体坐标系的 x 轴和 y 轴垂直。进而可以推出 $(m\dot{\boldsymbol{v}}+mg\boldsymbol{e}_3)$ 与机体坐标系的 z 轴平行，因此

$$\boldsymbol{z}_b=\frac{m\dot{\boldsymbol{v}}+mg\boldsymbol{e}_3}{\|m\dot{\boldsymbol{v}}+mg\boldsymbol{e}_3\|}.$$

再由无人机机体坐标系的偏航角 ψ 即可推出旋转矩阵 $\mathbf{R}$。为了方便后续对旋转矩阵 $\mathbf{R}$ 进行线性化，本文暂时使用四元数对其进行表述。偏航角对应的四元数可以表示为：

$$\boldsymbol{q}_\psi=(\cos(\frac{\psi}{2}),\ 0,\ 0,\ \cos(\frac{\psi}{2})),$$

而 z 轴对应的四元数可以表示为：

$$\boldsymbol{q}_z=\frac{1}{\sqrt{2(1+\boldsymbol{z}_b(3))}}(1+\boldsymbol{z}_b(3),\ -\boldsymbol{z}_b(2),\boldsymbol{z}_b(1),\ 0).$$

由此可推出体坐标系到惯性系的四元数为：

$$\boldsymbol{q}=\boldsymbol{q}_z\times\boldsymbol{q}_\psi=\frac{1}{\sqrt{2(1+\boldsymbol{z}_b(3))}}\begin{bmatrix}\Big(1+\boldsymbol{z}_b(3)\Big)\cdot\cos(\frac{\psi}{2})\\ \boldsymbol{z}_b(1)\sin(\frac{\psi}{2})-\boldsymbol{z}_b(2)\cos(\frac{\psi}{2})\\ \boldsymbol{z}_b(1)\cos(\frac{\psi}{2})+\boldsymbol{z}_b(2)\sin(\frac{\psi}{2})\\ \Big(1+\boldsymbol{z}_b(3)\Big)\cdot\sin(\frac{\psi}{2})\end{bmatrix}=\begin{bmatrix}q_w\\ q_x\\ q_y\\ q_z\end{bmatrix},\tag{2.54}$$

注意此处的乘法 $\times$ 为四元数的乘法。根据四元数与旋转矩阵的转换关系即可推出旋转矩阵的具体表达式：

$$\mathbf{R}=\begin{bmatrix}1-2(q_y^2+q_z^2) & 2(q_xq_y-q_wq_z) & 2(q_wq_y+q_xq_z)\\ 2(q_xq_y+q_wq_z) & 1-2(q_x^2+q_z^2) & 2(q_yq_z-q_wq_x)\\ 2(q_xq_z-q_wq_y) & 2(q_wq_x+q_yq_z) & 1-2(q_x^2+q_y^2)\end{bmatrix},\tag{2.55}$$

由公式 (2.50c) 可得，

$$\begin{aligned}\omega&=(\mathbf{R}^T\dot{\mathbf{R}})^\vee=2(\boldsymbol{q}_z\otimes\boldsymbol{q}_\psi)^{-1}\otimes(\dot{\boldsymbol{q}}_z\otimes\boldsymbol{q}_\psi+\boldsymbol{q}_z\otimes\dot{\boldsymbol{q}}_\psi)\\&=\begin{bmatrix}\dot{\boldsymbol{z}}_b(1)\sin(\psi)-\dot{\boldsymbol{z}}_b(2)\cos(\psi)-\frac{\dot{\boldsymbol{z}}_b(3)}{1+\dot{\boldsymbol{z}}_b(3)}\Big(\boldsymbol{z}_b(1)\sin(\psi)-\boldsymbol{z}_b(2)\cos(\psi)\Big)\\ \dot{\boldsymbol{z}}_b(1)\sin(\psi)+\dot{\boldsymbol{z}}_b(2)\cos(\psi)-\frac{\dot{\boldsymbol{z}}_b(3)}{1+\dot{\boldsymbol{z}}_b(3)}\Big(\boldsymbol{z}_b(1)\cos(\psi)+\boldsymbol{z}_b(2)\sin(\psi)\Big)\\ \frac{\boldsymbol{z}_b(2)\dot{\boldsymbol{z}}_b(1)-\boldsymbol{z}_b(1)\dot{\boldsymbol{z}}_b(2)}{1+\boldsymbol{z}_b(3)}+\dot{\psi}\end{bmatrix},\end{aligned}\tag{2.56}$$

进一步由 (2.50d) 可以得到，

$$\tau=\mathbf{M}\dot{\omega}+\omega\times\mathbf{M}\omega,\tag{2.57}$$

其中 $\dot{\omega}$ 可以通过对 (2.56) 求导得到，$\dot{\omega}$ 中包括的平坦输出导数的最高项为 $\rho^{(4)}$ 和 $\psi^{(2)}$。至此，四旋翼无人机状态空间中的非线性动力学方程、平坦空间中的线性动力学方程及两者之间的转换关系以及全部给出。

24029602

由以上分析可知，应该使用四阶积分器的离散时间模型表示无人机在微分平坦空间中的动力学模型，则优化问题中的动力学约束 (2.28b) 应为，

$$\boldsymbol{s}_{t+1} = \mathbf{A}\boldsymbol{s}_t + \mathbf{B}\boldsymbol{u}_t, \tag{2.58a}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & T_s\mathbf{I} & \frac{T_s^2}{2}\mathbf{I} & \frac{T_s^3}{6}\mathbf{I} \\ \mathbf{0} & \mathbf{I} & T_s\mathbf{I} & \frac{T_s^2}{2}\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & T_s\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{16\times 16}, \quad \mathbf{B} = \begin{bmatrix} \frac{T_s^4}{24}\mathbf{I} \\ \frac{T_s^3}{6}\mathbf{I} \\ \frac{T_s^2}{2}\mathbf{I} \\ T_s\mathbf{I} \end{bmatrix} \in \mathbb{R}^{16\times 4}, \tag{2.58b}$$

其中 $T_s \in \mathbb{R}_+$ 为采样时间，$\mathbf{I}$ 为适当维度的单位矩阵。状态变量为平坦输出及其一至三阶导数，即 $\boldsymbol{s}_t = (\boldsymbol{z}_t, \boldsymbol{z}_t^{(1)}, \boldsymbol{z}_t^{(2)}, \boldsymbol{z}_t^{(3)})$，而控制变量则应为平坦输出的四阶导数，即 $\boldsymbol{u}_t = \boldsymbol{z}_t^{(4)}$。

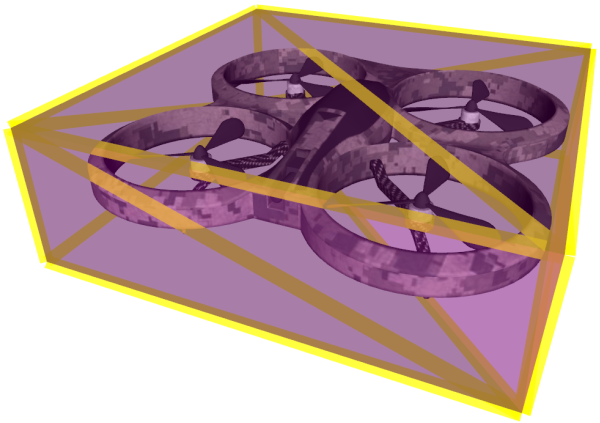


图 2.5　无人机的凸多胞体几何外形近似。

由于无人机的几何外形较为简单，本文直接使用单个多胞体对无人机的几何外形进行近似，如图 2.5 所示。该多胞体的坐标系与无人机的机体坐标系重合，即其旋转矩阵由 (2.55) 给出，其平移向量由 (2.52) 给出。仅剩的问题为如何对状态更新子问题 (2.33) 进行凸近似。

无人机的任务目标为：以尽可能小的控制量跟踪期望轨迹，因此问题目标函数 (2.28a) 应当设定为以下形式：

$$l(\boldsymbol{s}_t, \boldsymbol{u}_t) = \left\|\boldsymbol{s}_t - \boldsymbol{s}_t^r\right\|_{\mathbf{Q}_s}^2 + \left\|\boldsymbol{u}_t - \boldsymbol{u}_t^r\right\|_{\mathbf{Q}_u}^2 \tag{2.59}$$

其中 $\mathbf{Q}_s \in \mathbb{R}^{n_s\times n_s}$ 和 $\mathbf{Q}_u \in \mathbb{R}^{n_u\times n_u}$ 为正定的加权矩阵，$\boldsymbol{s}_t^r$ 为期望轨迹，$\boldsymbol{u}_t^r$ 为根据给定期望轨迹而得到的名义期望控制量。

注意到，问题中唯一的非凸部分为目标函数中的，

$$\mathfrak{R}(\boldsymbol{s}_t) = \left\|\mathbf{R}_i^T(\boldsymbol{s}_t)\kappa_{ijt}^k + \eta_{ijt}^k\right\|_2^2 \tag{2.60}$$

24029602

其中的参数由下式给出，

$$\kappa_{ijt}^k = \mathbf{C}_j^T \mu_{ijt}^k, \qquad \eta_{ijt}^k = \mathbf{A}_i^T \lambda_{ijt}^k + \zeta_{ijt}^k,$$

本文使用高斯牛顿法对该部分进行近似，即直接对 $\mathbf{R}_i^T(\boldsymbol{s}_t)\kappa_{ijt}^k + \eta_{ijt}^k$ 在 $\boldsymbol{s}_t^k$ 处进行泰勒展开，并保留一阶部分：

$$\Re(\boldsymbol{s}_t) \approx \left\| \mathbf{J}_{ijt}^k (\boldsymbol{s}_t - \boldsymbol{s}_t^k) + \eta_{ijt}^k \right\|_2^2, \tag{2.61}$$

其中的雅可比矩阵 $\mathbf{J}_{ijt}^k$ 可由链式法则给出：

$$\mathbf{J}_{ijt}^k = \left(\frac{\partial \boldsymbol{q}_t}{\partial \boldsymbol{s}_t} \frac{\partial \mathbf{R}_i^T \kappa_{ijt}^k}{\partial \boldsymbol{q}_t} \right)^T.$$

其中 $\frac{\partial \boldsymbol{q}_t}{\partial \boldsymbol{s}_t}$ 可由公式 (2.54) 算出，$\frac{\partial \mathbf{R}_i^T \kappa_{ijt}^k}{\partial \boldsymbol{q}_t}$ 可由公式 (2.55) 算出。至此，全尺寸实时避障算法在四旋翼无人机上的使用方法已经全部给出。

2.4.2 AirSim 高保真仿真

本小节在 AirSim[84] 高保真仿真器中对四旋翼无人机的高精度避障算法进行仿真验证。同时为了突出算法在实时性上的优越性，所提出算法将于以下目前文献中较为先进的算法进行对比：

1. 基于优化的避障算法[32] (Optimization-based Collision Avoidance, OBCA)。该算法采用符号距离描述的避障约束，并且直接采用通用的非线性优化器，如伪牛顿法、序列二次优化法，直接对高维问题进行求解。值得一提的是，OBCA 算法通常被视为离线规划算法。
2. 正则化对偶交替方向乘子法[34] (Regularized Dual Alternating Direction Method of Multiplier, RDA)。该算法采用符号距离描述的避障约束，通过交替方向乘子法将问题拆分为一系列二次锥规划问题以及一个相对较小的非线性状态更新优化问题。并通过 CPU 的线程池技术实现二次锥规划问题的并行求解，使用序列二次规划算法求解状态更新子问题。

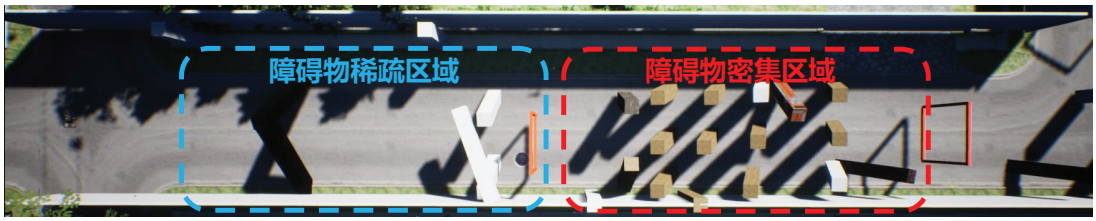
上述方法都将在英伟达的 Xavier NX 嵌入式开发平台上进行实现和测试。

AirSim 是微软针对无人机开发的仿真平台，其核心组件包括环境模型、无人机模型、物理引擎、传感器模型和飞控固件接口层等。在仿真过程中，AirSim 通过提供从仿真环境中获取的传感器信息向无人机飞控传递关键数据。飞控根据接收的信息生成执行信号，传递至无人机模型。无人机模型根据这些信号计算所需的力、扭矩，并设置电机转速。此外，阻力、摩擦力和重力等物理引擎的输入用于计算无人机在仿真环境中的下一个运动状态。这些考虑了环境模型中的

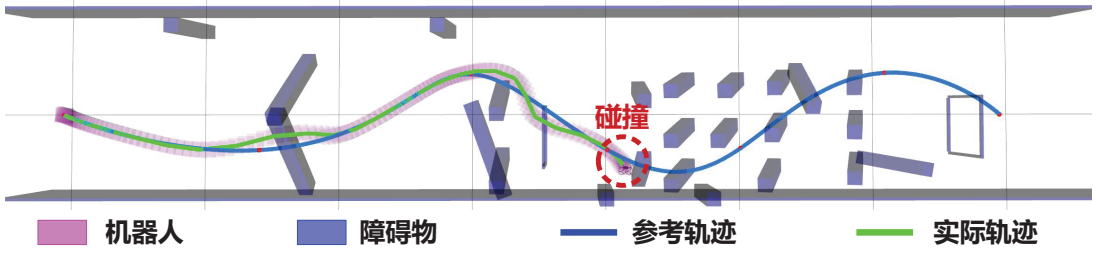
24029602

重力、空气密度、气压、磁场和 GPS 坐标的无人机动力学信息再传递给传感器模型，最终生成传感器获得的数据。

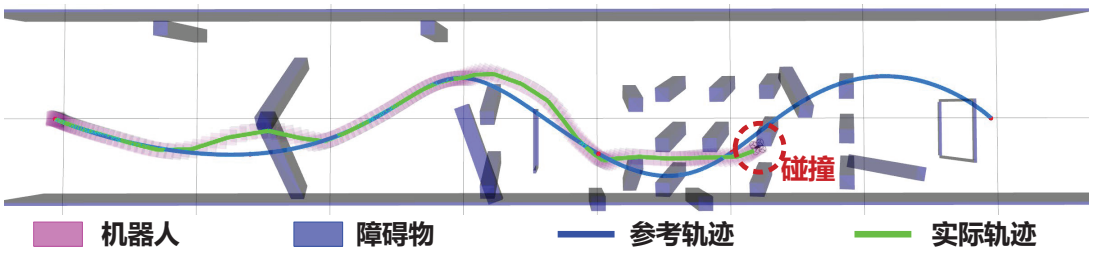
为了验证所提出算法，本文搭建了一个长 70 米，宽 20 米，高 6 米的仿真场景。在这个仿真环境中，分为障碍物稀疏区和障碍物密集区两个区域，这两个区域总共拥有 32 个柱状障碍物，如图 2.6(a) 所示。



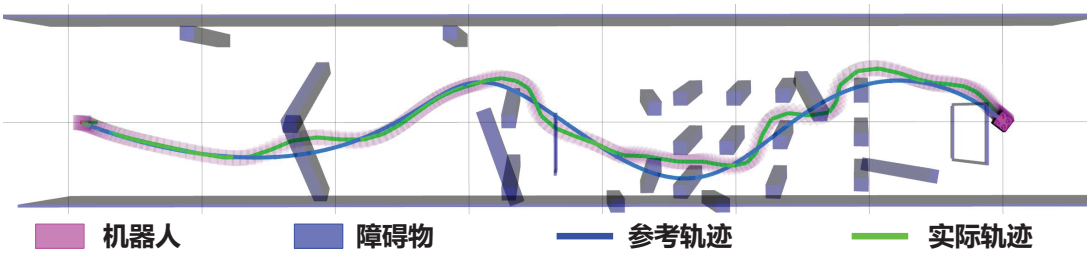
(a) AirSim 中的仿真场景，包括一个障碍物稀疏区域和一个稠密区域，总障碍物数量为 32。



(b) 使用 OBCA 算法控制无人机得到的真实轨迹。



(c) 使用 RDA 算法控制无人机得到的真实轨迹。



(d) 使用本文所提出算法控制无人机得到的真实轨迹。

图 2.6　一百次蒙特卡洛仿真中的一个例子。

对于所有方法，均设置模型预测控制的采样时间 $T_s = 0.1s$，预测时域 $T = 16$。本文所提算法及 RDA 算法的惩罚因子 $\sigma = 300$。参考轨迹的初始位置设为 $(0,\ 0,\ 1m)$，其余状态设为零；终止位置为 $(0,\ 70m,\ 1m)$，其余状态设为零；执行总时间为 $25s$。参考轨迹的生成方法为：在场景中先随机生成 5 个路点，然后使用多项式对初始位置、终止位置和路点进行插值。在导航的过程中，无人机只

24029602

能感知到长 20 米，宽 20 米，高 6 米范围内的障碍物。

其中一个对比案例如图 2.6(b)-图 2.6(d)所示。对于所有三个方法，无人机都能成功地通过障碍物稀疏区域。然而，OBCA 算法和 RDA 算法都导致无人机在障碍物密集区域发生了碰撞，只有本文提出的算法使得无人机成功避开了所有障碍物。

表 2.1　四旋翼无人机全尺寸避障轨迹跟踪算法基准比较

算法名称	成功率	导航总花费时间 *			导航总代价 *		
		最小值	最大值	平均值	最小值	最大值	平均值
OBCA 算法[32]	35%	**26.6s**	32.7s	**28.3s**	380.7	**532.6**	483.9
RDA 算法[34]	58%	27.2s	**32.1s**	28.5s	348.1	576.1	492.6
本文所提出算法	**92%**	26.8s	33.8s	28.7s	**322.4**	558.2	**458.7**

* 只有导航成功的样本会被纳入统计中。

本文将 100 次蒙特卡洛实验的结果总结于表 2.1 中。可见，三种算法在导航时间及导航代价这两项指标上表现相似，然而本文所提出方法的成功率显著超过了 OBCA 算法和 RDA 算法。

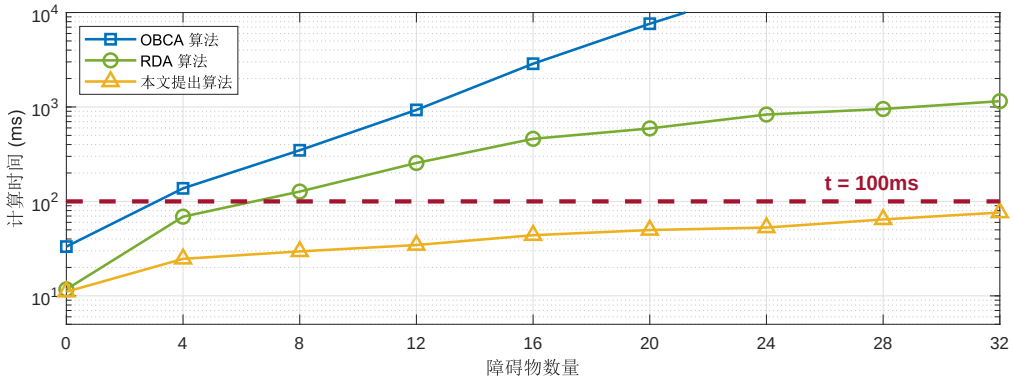


图 2.7　三种算法在预测时域 $T = 16$ 时，单个控制循环所需的平均计算时间。

为了进一步解释表中的结果，三种算法的计算时间被总结于图 2.7 中。OBCA 算法直接求解非凸优化问题，由于避障约束需要引入额外的对偶变量，所以非凸优化问题的维度与障碍物数量成正比。因此，当障碍物部件的数量 M 增加时，OBCA 算法的计算时间显著增加。如图 2.7 所示，OBCA 算法在障碍物密集环境中不能保证及时地 (在一个采样周期内，即小于 $0.1s$) 生成安全轨迹，所以会导致无人机发生碰撞。RDA 算法和本文提出的算法均使用交替方向乘子法将高维度非凸优化问题拆解为一系列低维度凸优化问题和一个相对较小的非线性状态更新优化问题。不同之处在于，RDA 算法使用的是基于符号距离的碰撞检测方法，从而导致了一系列二次锥规划。这些二次锥规划问题需要使用 CPU 的线程

24029602

池技术求解，这在嵌入式设备上依然需要消耗巨大的计算量。而本文提出的算法由于使用了基于一致放缩的碰撞检测方法，从而导致了一系列的二次规划问题，这些二次规划问题利用 GPU 的多线程算法求解，可以节省大量求解时间。

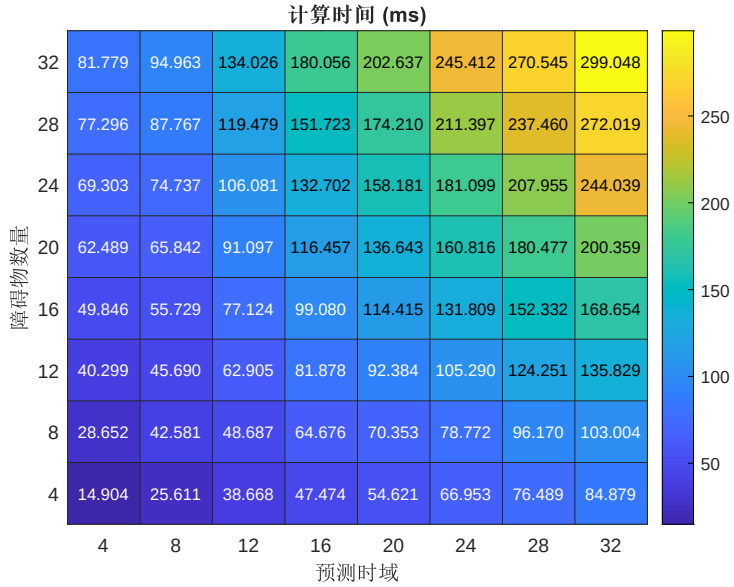


图 2.8　RDA 算法在嵌入式平台 Xavier NX 上进行一次对偶更新所需要的平均计算时间。

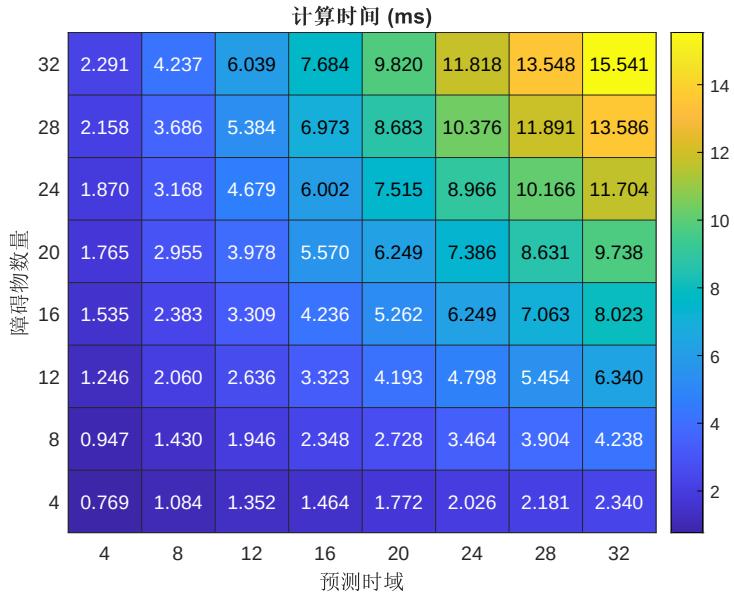


图 2.9　本文算法在嵌入式平台 Xavier NX 上进行一次对偶更新所需要的平均计算时间。

为了突出所提出算法相较于 RDA 算法的优势，本文总结了两种算法的对偶变量更新在嵌入式开发平台 Xavier NX 上所需消耗时间。值得注意的是，嵌入式开发平台 Xavier NX 的 CPU 的核心数为 6，而 GPU 的核心数量为 384。结果如图 2.8 和图 2.9 所示，可以看到本文提出算法的计算时间显著小于 RDA 算法，因此即使在障碍物稠密区域，本文所提出算法依然能够及时地生成安全轨迹。

24029602

2.5 小结

本章依据模型预测控制的思路提出了一种 GPU 加速的基于优化的单体全尺寸避障框架。为了实现全尺寸避障，在本框架中，机器人和障碍物的几何外形均被近似为多个凸多胞体组合。本章首先从凸多胞体的碰撞检测问题出发，提出了一种隐式的精确无碰撞约束，随后利用凸优化的强对偶性，给出了无碰撞约束的一种解析化表达。同时，本章也给出了该无碰撞约束背后蕴含的几何原理，给出了形象化且易于理解的解释。

针对该无碰撞约束会引入过多对偶变量使得优化问题难以求解的问题，本章按照交替方向乘子法的范式，将高纬度非凸问题转化为一系列低维度凸二次规划问题和一个可处理的状态更新优化问题。为了求解提高求解的实时性，本章给出了利用 GPU 加速求解多个低维度凸二次规划问题的方法。

为了演示算法的效果，本文以四旋翼无人机在障碍物稠密环境中的无碰撞 $\mathbf{SE}(3)$ 轨迹跟踪作为案例研究，介绍了如何针对本框架的特点挑选被控对象的系统模型、如何利用凸函数近似非凸约束和非凸目标函数的详细步骤。最后，通过与目前主流算法 (如 OBCA 算法和 RDA 算法) 的比较，证明了本文提出的算法在嵌入式设备上具有更高的避障成功率和更优越的实时性。

本章通过引入 GPU 加速和优化方法，为解决实际场景中复杂避障问题提供了新的思路和解决方案。在未来，这一框架有望在自主智能无人系统的发展中发挥重要作用，为其在复杂环境中的高效导航提供更可行的解决方案。

24029602

第 3 章　方位角约束下的编队导航方法

3.1　引言

基于优化的方法通过对机器人施加不同类型的编队约束来实现编队导航。现有文献中使用了各种各样的约束来实现多机器人的编队，这些约束包括相对位置约束[50,85-86]、相对距离距离[48-49]和相对方位[39]。使用什么样的编队约束方式决定了机器人集群在躲避障碍物时可以使用的编队机动，从而间接地影响了编队的性能。相对位置约束是最常用的编队约束方式。文献[50]使用相对位置误差的欧几里得范数来设计人工势函数，并利用人工势场法驱动机器人收敛到期望编队上。在文献[85]通过模型预测控制直接约束机器人之间的相对位置来实现期望编队。然而，由于相对位置约束只对位移具有不变性，因此上述方法只能使用编队整体平移机动来躲避障碍物，这严重限制了编队的灵活性，因此不适用于障碍物密集的环境。为了给编队避障提供更多的自由度，相对距离约束和相对方位约束被应用到了多机器人的编队导航中。在文献[49]利用距离约束对旋转和平移的不变性来实现编队机动控制。与旋转机动相比，缩放机动在实际情况中更为重要，特别是在执行诸如野外探索、废墟搜救等任务时，机器人集群通常需要使用缩放机动来通过狭窄的缝隙和通道。文献[48]提出了一种基于归一化距离拉普拉斯的代价函数—编队相似性误差，该代价函数对编队的旋转机动和缩放机动不敏感，因此能够为障碍物躲避提供更多的自由度。然而，归一化需要所有机器人之间的相对距离信息，这要求集群的通信网络拓扑必须是全联通的，这严重限制了其扩展到大规模集群的能力。与相对距离约束不同，相对方位约束具备对缩放和平移具有内在的不变性。此外，目前研究文献中已存在成熟的方位刚性理论，可以保证在不完全图下实现具有唯一几何形状的编队。但目前尚且缺少将方位约束和基于优化方法进行有机结合的文献。为了弥补这一差距，本章在权衡障碍物避障能力和编队规模扩展能力之后，采用方位约束来描述编队。

3.2　方位角度刚性理论预备知识

在正式介绍基于方位角度的多智能体协同编队导航之前，有必要先介绍与编队有关的基本概念及定理。为了保持文章的简洁，

定义 3.1 (编队构型)： 给定 N 个在 $\mathbb{R}^d$, $d = \{2, 3\}$ 空间中的智能体，记第 i 个智

24029602

能体的位置为 $\boldsymbol{p}_i \in \mathbb{R}^d$，则定义这 N 个智能体的编队构型为：

$$\boldsymbol{p} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N) \in \mathbb{R}^{Nd}. \tag{3.1}$$

定义 3.2 (编队网络)： 给定 N 个在 $\mathbb{R}^d$, $d = \{2, 3\}$ 空间中的智能体，智能体之间的通讯使用图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 来描述。其中 $\mathcal{V} = \{1, 2, \cdots, N\}$ 为节点集，在本文中，每个节点对应一个智能体；$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ 为图的边集，如果 $(i, j) \in \mathcal{E}$，则表明智能体 i 能从 j 处获取信息，本文聚焦于无向图，即，如果 $(i, j) \in \mathcal{E}$ 则可以推出 $(j, i) \in \mathcal{E}$。对于一个编队而言，其既和编队的构型 $\boldsymbol{p}$ 有关，也和编队内多智能体间的通讯网络 $\mathcal{G}$ 有关。因此，编队网络被定义为 $(\mathcal{G}, \boldsymbol{p})$。在后续章节中，为了方便会简称编队网络为编队。

定义 3.3 (邻居集合)： 智能体 i 的邻居集合定义为所有能够与之通讯的智能体，

$$\mathcal{N}_i = \{j \mid j \in \mathcal{V},\ (i, j) \in \mathcal{E}\}. \tag{3.2}$$

定义 3.4 (方位向量)： 在本文中，使用边向量 $\boldsymbol{e}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i$ 描述智能体 i 和 j 之间的相对位置。使用方位向量 (Bearing Vector) $\boldsymbol{g}_{ij} = \boldsymbol{e}_{ij}/\|\boldsymbol{e}_{ij}\|$ 描述智能体 i 和 j 之间的相对方位。需要注意的是，与边向量相比，方位向量只具备方向信息，而不具备距离信息。

定义 3.5 (正交投影矩阵)： 给定一个非零向量 $\boldsymbol{x} \in \mathbb{R}^d$, $d = \{2, 3\}$，定义其正交投影矩阵为：

$$\mathbf{P}_{\boldsymbol{x}} = \mathbf{I}_d - \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}\frac{\boldsymbol{x}^T}{\|\boldsymbol{x}\|} \in \mathbb{R}^{d\times d}, \tag{3.3}$$

其中 $\mathbf{I}_d$ 为 d 维的单位矩阵。

性质 3.1： 非零向量 $\boldsymbol{x}$ 的正交投影矩阵 $\mathbf{P}_{\boldsymbol{x}}$ 具有以下性质：

1. 正交投影矩阵是对称矩阵，$\mathbf{P}_{\boldsymbol{x}}^T = \mathbf{P}_{\boldsymbol{x}}$；
2. 相同正交投影矩阵的相乘满足 $\mathbf{P}_{\boldsymbol{x}}\mathbf{P}_{\boldsymbol{x}} = \mathbf{P}_{\boldsymbol{x}}$；
3. 正交投影矩阵的零空间为 $\boldsymbol{x}$，即 $Null(\mathbf{P}_{\boldsymbol{x}}) = span\{\boldsymbol{x}\}$；
4. 正交投影矩阵是半正定的，即 $\forall \boldsymbol{y} \in \mathbb{R}^d$，有，$\boldsymbol{y}^T\mathbf{P}_{\boldsymbol{x}}\boldsymbol{y} \geq 0$；
5. 对于任意两个非零向量 $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$，$\boldsymbol{x}, \boldsymbol{y}$ 平行当且仅当 $\mathbf{P}_{\boldsymbol{x}}\boldsymbol{y} = 0$。

定义 3.6 (方位拉普拉斯矩阵[87])： 编队网络的拉普拉斯矩阵 $\mathfrak{B}$ 定义为以下 $dN \times dN$ 维矩阵：

$$[\mathfrak{B}]_{ij} = \begin{cases} 0_{3\times 3}, & i \neq j, (i, j) \notin \mathcal{E} \\ -\mathbf{P}_{\boldsymbol{g}_{ij}}, & i \neq j, (i, j) \in \mathcal{E} \\ \sum\limits_{k\in\mathcal{N}_i} \mathbf{P}_{\boldsymbol{g}_{ik}}, & i = j, i \in \mathcal{V} \end{cases}, \tag{3.4}$$

24029602

其中 $[\mathfrak{B}]_{ij} \in \mathbb{R}^{d\times d}$ 是矩阵 $\mathfrak{B}$ 的第 i 行，第 j 列的子矩阵。

拉普拉斯矩阵 $\mathfrak{B}$ 不仅描述了编队的通讯网络信息，还描述了沿着所有边的方位信息，因此包含了描述编队网络的所有信息。

方位刚性理论为分析编队网络的几何形状提供了强有力的数学工具。通过方位刚性理论，可以设计条件使得编队网络始终具备一致的几何形状，因此有必要介绍方位刚性理论。

定义 3.7 (方位相等)： 两个编队网络 $(\mathfrak{G}, \boldsymbol{p})$ 和 $(\mathfrak{G}, \boldsymbol{p}')$ 是方位相等的如果对于通讯网络 $\mathfrak{G}$ 内所有的边 $\forall(i,j) \in \mathfrak{E}$ 都存在 $\mathbf{P}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}(\boldsymbol{p}'_i - \boldsymbol{p}'_j) = 0$。

定义 3.8 (方位全等)： 两个编队网络 $(\mathfrak{G}, \boldsymbol{p})$ 和 $(\mathfrak{G}, \boldsymbol{p}')$ 是方位全等的如果对于编队网络内任意两个节点 $i, j \in \mathcal{V}$ 都存在 $\mathbf{P}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}(\boldsymbol{p}'_i - \boldsymbol{p}'_j) = 0$。

定义 3.9 (全局方位刚性[44])： 一个编队网络 $(\mathfrak{G}, \boldsymbol{p})$ 被称为全局方位刚性，如果任意与之方位相等的编队网络都与之方位全等。

定义 3.10 (无穷小方位刚性[87])： 如果只有整个构型的放缩和整个构型的平移能够保证编队网络 $(\mathfrak{G}, \boldsymbol{p})$ 中所有的方位信息不变，则称该编队网络具有无穷小方位刚性 (Infinitesimally Bearing Rigid, IBR)。

无穷小方位刚性与全局方位刚性具有密切的关联：

引理 3.1： 无穷小方位刚性网络是全局方位刚性的。

在实际的应用中，通常希望运用无穷小方位刚性设计编队的网络，因为无穷小方位刚性编队网络的几何形状是被唯一确定的。该结论由以下引理给出。

引理 3.2 (无穷小方位刚性编队网络具有唯一的几何形状)： 一个无穷小方位刚性网络的构型 $\boldsymbol{p}'$ 可以被一个放缩比例和一个平移向量唯一地确定。更准确的数学表达为，给定一个编队网络 $(\mathfrak{G}, \boldsymbol{p})$，则任意具有相同通讯网络且与之方位相等的编队网络 $(\mathfrak{G}, \boldsymbol{p}')$ 都具有以下的构型：

$$\boldsymbol{p}' = \alpha \cdot \boldsymbol{p} + \mathbf{1}_N \otimes \rho. \tag{3.5}$$

其中 $\alpha \in \mathbb{R}$ 为放缩比例，$\rho \in \mathbb{R}^d$ 为平移向量，$\mathbf{1}_N \in \mathbb{R}^N$ 为元素全为 1 的 N 维向量。同时，称该通信网络 $\mathfrak{G}$ 是广泛无穷小方位刚性的。

尽管编队网络是由构型 $\boldsymbol{p}$ 和通信网络 $\mathfrak{G}$ 共同决定的，但是编队网络的无穷小方位刚性的性质主要是由通信网络 $\mathfrak{G}$ 决定的，该结论由以下引理给出。

24029602

引理 3.3 (无穷小方位刚性网络的密度)： 给定一个通信图 $\mathcal{G}$，如果至少存在一个构型 $\boldsymbol{p}'$ 使得编队网络 $(\mathcal{G}, \boldsymbol{p}')$ 是无穷小方位刚性的。那么，对于几乎所有的构型 $\boldsymbol{p}$，编队网络 $(\mathcal{G}, \boldsymbol{p})$ 都是无穷小方位刚性的。或者说，使得编队网络 $(\mathcal{G}, \boldsymbol{p})$ 不是无穷小方位刚性的构型 $\boldsymbol{p}$ 的测度为零。

3.3 编队导航优化问题建模与求解

本章节将基于优化的思想提出一种适用于障碍物稠密环境中的编队导航方法。在该方法中，智能体集群通过协同优化一个编队性能指标来形成所期望的编队。为了给避障提供更多的自由度，本文将在小节 3.3.1 中提出一种新颖的编队性能指标，该性能指标是对于整体编队平移和整体编队放缩是不变的。并且，本文证明了在广义方位刚性网络的条件下，可以保证所形成编队的形状是唯一的。随后在小节 3.3.2 讨论智能体的轨迹表示方法并在小节 3.3.3 中讨论基于该表示方法的在线轨迹优化方法。最后，在小节 3.4 中对所提出算法进行仿真和实物实验验证。

3.3.1 编队性能指标

给定一个期望的方位 $\boldsymbol{g}^*$ 和一个实际的边向量 $\boldsymbol{e}$，则可以定义方位误差 δ 为：

$$\delta = \|\mathbf{P}_{\boldsymbol{g}^*} \cdot \boldsymbol{e}\|_2, \tag{3.6}$$

方位误差的几何意义可以解释为实际边向量 $\boldsymbol{e}$ 垂直于期望方位 $\boldsymbol{g}^*$ 分量的模长，如图 3.1 所示。方位误差 $\delta = 0$ 当且仅当边向量 $\boldsymbol{e}$ 平行于期望方位 $\boldsymbol{g}^*$，或者 $\boldsymbol{e} = \mathbf{0}$。

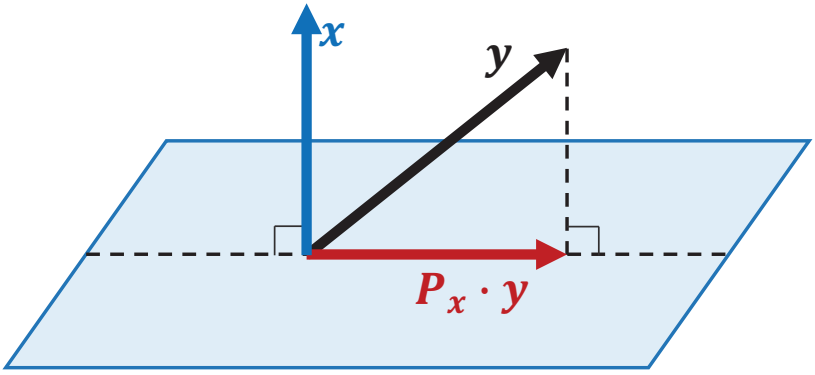


图 3.1 投影矩阵、方位误差的示意图。

24029602

给定一个期望的编队构型 $\boldsymbol{p}^* = (\boldsymbol{p}_1^*, \boldsymbol{p}_2^*, \cdots, \boldsymbol{p}_N^*)$，可以根据计算出期望的方位向量 $\boldsymbol{g}^* = (\boldsymbol{g}_{12}^*, \cdots, \boldsymbol{g}_{ij}^*, \cdots)$，进一步可以计算出期望编队网络 $(\mathcal{G}, \boldsymbol{p}^*)$ 的拉普拉斯矩阵 $\mathcal{B}$。考虑以下半正定二次型的编队性能指标：

$$\begin{aligned}\psi_f(\boldsymbol{p}) &= \boldsymbol{p}^T \mathcal{B} \boldsymbol{p} \\ &= \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} (\boldsymbol{p}_i - \boldsymbol{p}_j)^T \mathbf{P}_{\boldsymbol{g}_{ij}^*} (\boldsymbol{p}_i - \boldsymbol{p}_j) \\ &= \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \left\| \mathbf{P}_{\boldsymbol{g}_{ij}^*} (\boldsymbol{p}_i - \boldsymbol{p}_j) \right\|_2^2 \\ &= \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \delta_{ij}^2 \geq 0,\end{aligned} \tag{3.7}$$

其中 $\boldsymbol{p} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$ 为集群实际的构型。该编队性能指标对于编队的整体放缩和整体平移具有不敏感性，由以下命题给出。

命题 3.1 (性能指标对于放缩和平移的不敏感性)： 如果期望编队网络 $(\mathcal{G}, \boldsymbol{p}^*)$ 的通信图 $\mathcal{G}$ 是广泛无穷小方位刚性的，那么编队性能指标 $\psi_f(\boldsymbol{p})$ 的最小值为 0，且对应的极小值点为：

$$\{\boldsymbol{x} \in \mathbb{R}^{dN} \mid \boldsymbol{x} = \alpha \cdot \boldsymbol{p}^* + \mathbf{1}_N \otimes \rho,\ \alpha \in \mathbb{R},\ \rho \in \mathbb{R}^d\}, \tag{3.8}$$

其中 α 为放缩比例，ρ 为平移向量。

证明： 首先证明编队性能指标可以取到极小值 0。由编队性能指标的定义 (3.7) 可知，$\psi_f(\boldsymbol{p}) = 0$ 当且仅当对于所有的边 $\forall (i,j) \in \mathcal{G}$ 都有 $\delta_{ij} = 0$，即 $(\boldsymbol{p}_i - \boldsymbol{p}_j)$ 平行于 $\boldsymbol{g}_{ij}^*$。接下来证明编队性能指标的极小值点为 (3.8)。当对于所有的边 $\forall (i,j) \in \mathcal{G}$ 都有 $(\boldsymbol{p}_i - \boldsymbol{p}_j)$ 平行于 $\boldsymbol{g}_{ij}^*$ 时，由定义可知实际编队网络 $(\mathcal{G}, \boldsymbol{p})$ 于期望编队网络方位相等。再由引理 3.2 可知，实际网络编队的构型 $\boldsymbol{p}$ 具有以下性质：

$$\boldsymbol{p} = \alpha \cdot \boldsymbol{p}^* + \mathbf{1}_N \otimes \rho, \tag{3.9}$$

证明完毕。 □

该命题证明了，编队性能指标 $\psi_f(\boldsymbol{p})$ 的极小值点所对应的构型与期望编队的构型构型只相差一个放缩比例 α 和一个平移向量 ρ。重申在本章引言部分对于多智能体编队的期望：能够利用放缩、平移等方式对障碍物进行躲避。而编队性能指标 $\psi_f(\boldsymbol{p})$ 的对于放缩、平移不敏感的特性正好符合这一期望。

这个编队性能指标 (3.7) 是整个编队的性能指标，因为它是所有边的方位误差的累加。然而，在实际的应用中更期望能够仅以局部信息，如邻居的信息来评价编队的性能，因此有必要考虑对整体编队性能指标 (3.7) 进行拆分。考虑以下

24029602

拆分方法，使得局部性能指标仅依赖于邻居信息：

$$\psi_f(\boldsymbol{p}) = \sum_{i\in\mathcal{V}} \psi_{i,f}(\boldsymbol{p}), \tag{3.10a}$$

$$\psi_{i,f}(\boldsymbol{p}) = \frac{1}{2}\sum_{j\in\mathcal{N}_i} (\boldsymbol{p}_i - \boldsymbol{p}_j)^T \mathbf{P}_{\boldsymbol{g}_{ij}^*} (\boldsymbol{p}_i - \boldsymbol{p}_j) = \frac{1}{2}\sum_{j\in\mathcal{N}_i} \delta_{ij}^2. \tag{3.10b}$$

局部性能指标 (3.10b) 具有以下特点：

1. 给定邻居的位置信息 $\{\boldsymbol{p}_j\}_{j\in\mathcal{N}_i}$，局部性能指标对于智能体 i 的位置 $\boldsymbol{p}_i$ 来说是凸的。相较于文献[48] 中提出的编队相似性指标而言，局部性能指标 (3.10b) 为集群形成编队提供了更大的收敛域。这个特点为集群中的智能体能够从任何初始位置出发形成编队提供了可能性。
2. 局部编队性能指标 (3.10b) 为智能体 i 提供了全局性能指标 (3.7) 中的全部信息。在实际的使用过程中，智能体 i 通常沿着全局性能指标 (3.7) 的负梯度方向移动来对指标进行优化。局部编队性能指标 (3.10b) 对于智能体 i 的位置的梯度与全局编队性能指标 (3.7) 的梯度之间只在幅值上相差两倍，而两个梯度方向是一致的，即：

$$\frac{\partial \psi_f}{\partial \boldsymbol{p}_i} = 2\sum_{j\in\mathcal{N}_i} \mathbf{P}_{\boldsymbol{g}_{ij}^*} (\boldsymbol{p}_i - \boldsymbol{p}_j) = 2\frac{\partial \psi_{i,f}}{\partial \boldsymbol{p}_i}, \tag{3.11}$$

这使得智能体能够以分布式的方法协同优化性能指标来实现编队。

该性能指标可以如图 3.2 所解释，当固定其他智能体位置时，局部性能指标函数的极小值点就是使得方位误差最小的位置。有了这一局部性能指标，下一步即为将其放入到轨迹优化的框架中进行优化。

3.3.2 智能体轨迹表示方法

在给出轨迹优化的方法之前，需要先讨论智能体轨迹的表示方法。由于绝大多数自主智能无人系统都据有微分平坦的特性，如水面无人舰艇、自主地面移动机器人、旋翼无人机和固定翼无人机等。由上一章的解释可知，这些系统的状态都可以使用解耦的积分链来表示，因此在本章节中，将对每维度状态都采用分段多项式进行表示。特别的，由于编队只和智能体系统的位置状态有关，本文只对与位置有关的状态进行轨迹优化。

本文使用 M 段分段多项式来表示智能体的位置轨迹，其中第 i 段轨迹表示为：

$$\boldsymbol{p}_i(t) = \boldsymbol{c}_i^T \beta(t), \qquad \forall t \in [0, T_i), \tag{3.12}$$

其中 $\boldsymbol{p}_i(t) : \mathbb{R} \to \mathbb{R}^d$ 表示轨迹是以时间 t 为参数的向量函数；$\boldsymbol{c}_i \in \mathbb{R}^{(n_d+1)\times d}$ 为多

24029602

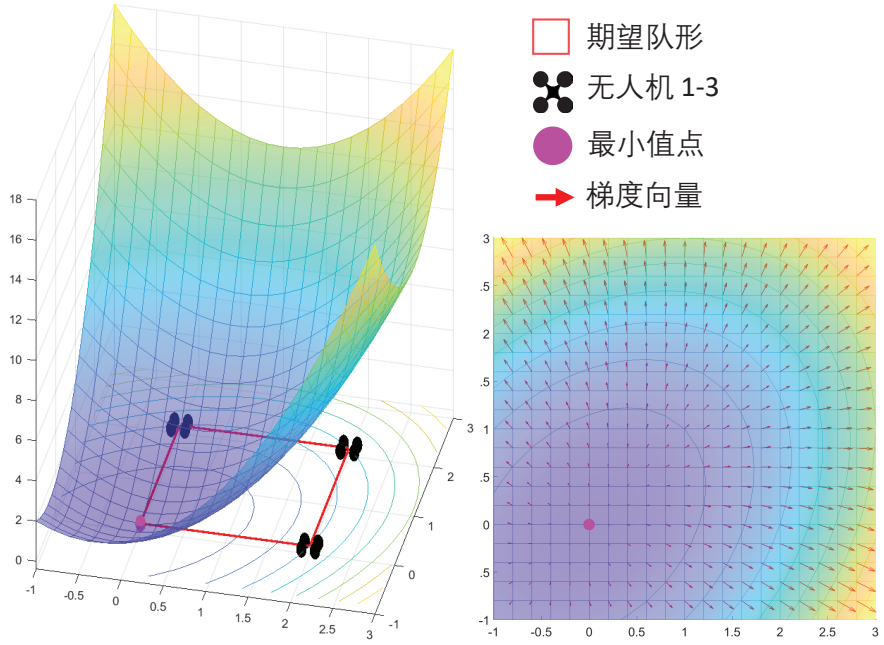


图 3.2　当无人机 1-3 位置确定时，无人机 4 的局部编队性能指标及其梯度示意图。

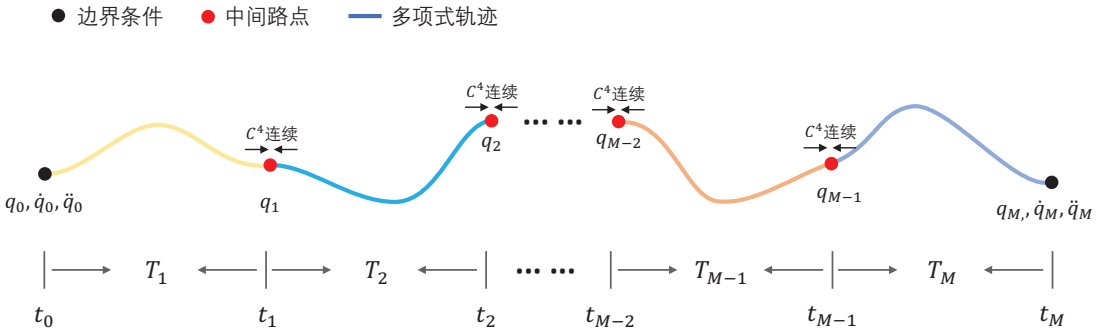


图 3.3　智能体的多项式轨迹示意图。

项式的系数；$\beta(t) = (1, t, \cdots, t^{n_d}) \in \mathbb{R}^{n_d+1}$ 为多项式的自然基；n_d 为多项式的次数，需要根据智能体具体的类型做选择，本章节将聚焦于四旋翼无人机，为了保证轨迹的四阶连续可导，选用 $n_d = 5$。

整条轨迹最终被表示为：

$$\boldsymbol{p}(t) = \boldsymbol{p}_m(t - t_{m-1}), \quad \forall t \in [t_{m-1}, t_m), \quad m = 1, \cdots, M, \tag{3.13}$$

其中 $t_i = \sum_{j=1}^{i} T_j$ 为第 i 个时间节点。为了方便表示，使用简记法，$\boldsymbol{c} = (\boldsymbol{c}_1, \cdots, \boldsymbol{c}_M)$ 和 $\boldsymbol{T} = (T_1, \cdots, T_M)$。那么，当给定参数 $\boldsymbol{c}$ 和 $\boldsymbol{T}$ 后，就可以唯一的确定一条轨迹。整条轨迹的示意图如图 3.3 所示。

对于导航而言，轨迹通常需要满足两个要求。首先，轨迹需要在特定时间经过一系列路点 (waypoint)。其次，为了保证所表示的轨迹在实际中是可以执行的，有必要保证轨迹 (3.13) 是连续可导的。如对于四旋翼无人机，位置的四阶导数与无人机的电机总推力、扭矩是直接相关的，为了保证控制量的连续性，需要保证

24029602

该轨迹是四阶连续可导的，即保证轨迹是 C^4 连续的。

对于轨迹表达方式 (3.13) 而言，只可能在时间节点 t_m, $m=1,\cdots,M-1$ 处出现不连续。因此只需要保证在每一个时间节点处的 C^4 连续即可。本文通过以下约束保证轨迹在节点 t_m 处经过路点 $\boldsymbol{q}_m$ 及轨迹的 4 阶连续可导性。

$$\begin{pmatrix}\mathbf{E}_m & \mathbf{F}_m\end{pmatrix}\begin{pmatrix}c_m\\ c_{m+1}\end{pmatrix}=\begin{pmatrix}q_m^T\\ \mathbf{0}_{5\times d}\end{pmatrix},\ m=1,\cdots,M, \tag{3.14}$$

其中 $\mathbf{E}_m$ 和 $\mathbf{F}_m$ 由以下公式给出，

$$\mathbf{E}_m=\begin{pmatrix}1 & T_m & T_m^2 & T_m^3 & T_m^4 & T_m^5\\ 1 & T_m & T_m^2 & T_m^3 & T_m^4 & T_m^5\\ 0 & 1 & 2T_m & 3T_m^2 & 4T_m^3 & 5T_m^4\\ 0 & 0 & 2 & 6T_m & 12T_m^2 & 20T_m^3\\ 0 & 0 & 0 & 6 & 24T_m & 60T_m^2\\ 0 & 0 & 0 & 0 & 24 & 120T_m\end{pmatrix}, \tag{3.15}$$

$$\mathbf{F}_m=\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\\ -1 & 0 & 0 & 0 & 0 & 0\\ 0 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & -2 & 0 & 0 & 0\\ 0 & 0 & 0 & -6 & 0 & 0\\ 0 & 0 & 0 & 0 & -24 & 0\end{pmatrix}. \tag{3.16}$$

矩阵 $\mathbf{E}_m$ 的第一行约束了轨迹在时间节点 t_m 必须经过路点 $\boldsymbol{q}_m$，而矩阵 $\mathbf{E}_m$ 和矩阵 $\mathbf{F}_m$ 的其余五行约束了轨迹必须是四阶连续可导的。

特殊的，对于轨迹的起点和终点，通常还分别需要满足给定的位置、速度和加速度初始条件和终末条件。该条件由以下约束给出，

$$\mathbf{F}_0c_0=\begin{pmatrix}q_0^T\\ \dot{q}_0^T\\ \ddot{q}_0^T\end{pmatrix},\quad \mathbf{E}_Mc_M=\begin{pmatrix}q_M^T\\ \dot{q}_M^T\\ \ddot{q}_M^T\end{pmatrix}, \tag{3.17}$$

其中 $q_0^T,\dot{q}_0^T,\ddot{q}_0^T$ 分别为初始位置、速度和加速度；$q_M^T,\dot{q}_M^T,\ddot{q}_M^T$ 分别为终止位置、速度和加速度；矩阵 $\mathbf{F}_0$ 和 $\mathbf{E}_M$ 由以下公式给出，

$$\mathbf{F}_0=\begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 2 & 0 & 0 & 0\end{pmatrix}, \tag{3.18}$$

24029602

$$\mathbf{E}_M = \begin{pmatrix} 1 & T_M & T_M^2 & T_M^3 & T_M^4 & T_M^5 \\ 0 & 1 & 2T_M & 3T_M^2 & 4T_M^3 & 5T_M^4 \\ 0 & 0 & 2 & 6T_M & 12T_M^2 & 20T_M^3 \end{pmatrix}. \tag{3.19}$$

约束 (3.14) 和 (3.17) 最终被写为以下约束形式:

$$\mathscr{M}\mathbf{c} = \mathbb{Q}, \tag{3.20a}$$

$$\mathscr{M} = \begin{pmatrix} \mathbf{F}_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{E}_1 & \mathbf{F}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 & \mathbf{F}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{F}_{M-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{E}_M \end{pmatrix} \tag{3.20b}$$

$$\mathbb{Q} = (q_0, \dot{q}_0, \ddot{q}_0, q_1, \mathbf{0}_{3\times5}, \cdots, q_{M-1}, \mathbf{0}_{3\times5}, q_M, \dot{q}_M, \ddot{q}_M)^T. \tag{3.20c}$$

在文献[77] 中已经证明，对于任意 $\boldsymbol{T} \succeq \mathbf{0}$，矩阵 $\mathscr{M}$ 都是可逆的。因此，一旦给定路点 $\mathbf{q} = [q_1, \cdots, q_{M-1}] \in \mathbb{R}^{3\times(M-1)}$ 和时间向量 $\mathbf{T} = [T_1, \cdots, T_M]$，那么就可以唯一地确定一条 C^4 的轨迹。因此，通过将在轨迹的参数将从 $(\boldsymbol{c}, \boldsymbol{T})$ 转换为 $(\boldsymbol{q}, \boldsymbol{T})$，轨迹将自然地保证轨迹能够穿过路点及 C^4 连续性。在下一小节中，将基于参数 $(\boldsymbol{q}, \boldsymbol{T})$ 对轨迹进行优化。

3.3.3　轨迹优化问题建模及在线求解方法

考虑以下多智能体编队导航的轨迹优化问题：

$$\min_{\boldsymbol{q},\,\boldsymbol{T}} \qquad \lambda^T \cdot (F_c, F_T, F_v, F_a, F_o, F_f, F_r) \tag{3.21}$$

其中 λ 为根据各项目标的代价函数进行取舍而设定的权重参数。函数 F_x，$x \in \{c, T, a, v, o, f, r\}$ 为根据不同性能指标和约束条件而设计的代价函数。

给定一个需要在整个轨迹上满足的约束函数 $\psi_x \le 0$，则其对应的代价函数 F_x 为以下通过在整条轨迹的时域上进行均匀采样而构建的惩罚函数：

$$F_x = \sum_{m=1}^{M} F_{x,m}, \tag{3.22a}$$

$$F_{x,m} = \frac{T_m}{\kappa_m} \sum_{k=0}^{\kappa_m} \varpi_k \max\{\psi_x(t_{m-1} + \frac{k}{\kappa_m} T_m), 0\}^3 \tag{3.22b}$$

24029602

其中，$\max\{\psi_x(t_{m-1}+\frac{k}{\kappa_m}T_m),0\}$ 可以理解为约束违反的程度；ϖ_k 为由梯形法则而确定的积分系数[88]，其取值为 $\varpi_k=1$，$k=1,\cdots,\kappa_m-1$ 及 $\varpi_0=\varpi_{\kappa_m}=\frac{1}{2}$。

通常而言，函数 $F_{x,m}$ 是以轨迹参数 $(\boldsymbol{c},\boldsymbol{T})$ 为变量的函数。然而，保证轨迹的 C^4 连续性，需要计算函数 $F_{x,m}$ 相对于轨迹参数 $(\boldsymbol{q},\boldsymbol{T})$ 的梯度。该梯度信息可以通过隐函数求导法则及链式法则求取。记 $q_{i,j}$ 为矩阵 $\boldsymbol{q}$ 第 i 行，第 j 列的元素，则通过对等式约束 (3.20a) 的两边进行求导可得：

$$\mathcal{M}\frac{\partial \boldsymbol{c}}{\partial q_{i,j}}=\frac{\partial \mathbb{Q}}{\partial q_{i,j}}. \tag{3.23}$$

那么可以进一步求得函数 $F_{x,m}$ 相对于 $q_{i,j}$ 的梯度为：

$$\begin{aligned}\frac{dF_{x,m}}{dq_{i,j}}&=Tr\{(\frac{\partial \boldsymbol{c}}{\partial q_{i,j}})^T\frac{\partial F_{x,m}}{\partial \boldsymbol{c}}\}\\&=Tr\{(\mathcal{M}^{-1}\frac{\partial \mathbb{Q}}{\partial q_{i,j}})^T\frac{\partial F_{x,m}}{\partial \boldsymbol{c}}\}.\end{aligned} \tag{3.24}$$

通过等式约束 (3.20a) 的两边对 T_l 进行求导可得：

$$\frac{\partial \mathcal{M}}{\partial T_l}\mathbf{c}+\mathcal{M}\frac{\partial \mathbf{c}}{\partial T_l}=0. \tag{3.25}$$

进而可以得到函数 $F_{x,m}$ 相对于 T_l 的梯度为：

$$\begin{aligned}\frac{dF_{x,m}}{dT_l}&=\frac{\partial F_{x,m}}{\partial T_l}+Tr\{(\frac{\partial \mathbf{c}}{\partial T_l})^T\frac{\partial F_{x,m}}{\partial \mathbf{c}}\}\\&=\frac{\partial F_{x,m}}{\partial T_l}-Tr\{(\mathcal{M}^{-1}\frac{\partial \mathcal{M}}{\partial T_l}\mathbf{c})^T\frac{\partial F_{x,m}}{\partial \mathbf{c}}\}.\end{aligned} \tag{3.26}$$

在公式 (3.24) 和 (3.26) 中，梯度 $\partial\mathcal{M}/\partial T_m$ 可以根据公式 (3.15) 和 (3.19) 计算出来；而梯度 $\partial\mathbb{Q}/\partial q_{i,j}$ 可以根据公式 (3.20c) 计算出来。只有梯度 $\partial F_{x,m}/\partial\boldsymbol{c}$ 和 $\partial F_{x,m}/\partial\boldsymbol{T}$ 需要进一步讨论。

记 $\boldsymbol{c}_l$ 和 T_l 为多项式系数 $\boldsymbol{c}$ 和时间向量 T 的第 l 个元素。当 $\psi_x(t_{m-1}+\frac{k}{\kappa_m}T_m)\geq 0$ 时，跟据 (3.22b) 有，

$$\frac{\partial F_{x,m}}{\partial c_l}=3\frac{T_m}{\kappa_m}\sum_{k=0}^{\kappa_m}\varpi_k[\psi_x(t_{m-1}+\frac{k}{\kappa_m}T_m)]^2\cdot\frac{\partial \psi_x}{\partial c_l}. \tag{3.27}$$

关于时间的梯度则有一些不一样，当 $l\neq m$ 时有，

$$\frac{\partial F_{x,m}}{\partial T_l}=3\frac{T_m}{\kappa_m}\sum_{k=0}^{\kappa_m}\varpi_k[\psi_x(t_{m-1}+\frac{k}{\kappa_m}T_m)]^2\cdot\frac{\partial \psi_x}{\partial T_l}, \tag{3.28}$$

当 $l=m$ 时有，

$$\frac{\partial F_{x,m}}{\partial T_m}=\frac{F_{x,m}}{T_m}+3\frac{T_m}{\kappa_m}\sum_{k=0}^{\kappa_m}\varpi_k[\psi_x(t_{m-1}+\frac{k}{\kappa_m}T_m)]^2\cdot\frac{\partial \psi_x}{\partial T_m}. \tag{3.29}$$

24029602

到目前为止，仅剩 $\partial\psi_x/\partial c_l$ 和 $\partial\psi_x/\partial T_l$ 需要根据不同的约束和性能指标进一步讨论，其余所有梯度计算方法已经给出。紧接着将介绍无人机编队导航中需要用到的约束及性能指标，及其梯度的推导。

1) 控制效用 F_c：为了使得无人机花费尽可能少的控制效用，同时使得无人机的轨迹尽可能光滑，通过惩罚无人机的加加速度的幅值设计以下控制效用指标：

$$F_c = \sum_{m=1}^{M} \int_0^{T_m} \|\boldsymbol{p}_i^{(3)}(t)\|^2 dt, \tag{3.30a}$$

$$\frac{\partial F_c}{\partial \boldsymbol{c}_l} = 2\left(\int_0^{T_m} \beta^{(3)}(t)\beta^{(3)}(t)^T dt\right)\boldsymbol{c}_l, \tag{3.30b}$$

$$\frac{\partial F_c}{\partial T_l} = \boldsymbol{c}_l^T \beta^{(3)}(T_l)\beta^{(3)}(T_l)^T \boldsymbol{c}_l. \tag{3.30c}$$

2) 轨迹总执行时间 F_T：为了使得无人机能够花费尽可能少的时间，通过惩罚无人机轨迹的总时间来设计以下时间指标：

$$F_T = \sum_{m=1}^{M} T_m, \qquad \frac{\partial F_T}{\partial \boldsymbol{c}_l} = \boldsymbol{0}, \qquad \frac{\partial F_c}{\partial T_l} = 1. \tag{3.31}$$

3) 速度约束 ψ_v 和加速度约束 ψ_a：在实际的场景中，无人机由于执行器饱和而不能超过最大速度 $\boldsymbol{v}_{max}$ 和最大加速度 $\boldsymbol{a}_{max}$，因此需要加入以下约束：

$$\psi_v = \|\dot{\boldsymbol{p}}\|^2 - \boldsymbol{v}_{max}^2 \le 0, \tag{3.32a}$$

$$\psi_a = \|\ddot{\boldsymbol{p}}\|^2 - \boldsymbol{a}_{max}^2 \le 0, \tag{3.32b}$$

它们对于多项式系数 $\boldsymbol{c}_l$ 的梯度为：

$$\frac{\partial \psi_v}{\partial \boldsymbol{c}_l} = \beta^{(1)}(t)\beta^{(1)}(t)^T \boldsymbol{c}_l, \tag{3.33a}$$

$$\frac{\partial \psi_a}{\partial \boldsymbol{c}_l} = \beta^{(2)}(t)\beta^{(2)}(t)^T \boldsymbol{c}_l, \tag{3.33b}$$

它们对于多项式时间间隔 T_l 的梯度则为：

$$\frac{\partial \psi_v}{\partial T_l} = \left(\boldsymbol{c}_l^T \beta^{(2)}(t)\beta^{(1)}(t)^T \boldsymbol{c}_l\right) \cdot \frac{\partial t}{\partial T_l}, \tag{3.34a}$$

$$\frac{\partial \psi_a}{\partial T_l} = \left(\boldsymbol{c}_l^T \beta^{(3)}(t)\beta^{(2)}(t)^T \boldsymbol{c}_l\right) \cdot \frac{\partial t}{\partial T_l}, \tag{3.34b}$$

其中的 t 为局部时间，即在第 m 段轨迹上，第 k 个采样点的局部时间为 $t = \frac{k}{\kappa_m} T_m$，偏导数 $\frac{\partial t}{\partial T_l}$ 由以下公式给出：

$$\frac{\partial t}{\partial T_l} = \begin{cases} 0, & m \ne l, \\ \dfrac{k}{\kappa_l}, & m = l. \end{cases} \tag{3.35}$$

24029602

4) 静态避障约束 ψ_o：为了使得无人机能够躲避障碍物，无人机到最近障碍物之间的距离应当大于安全阈值 d_{thr}，因此设计以下约束：

$$\psi_o = d_{thr} - d(\boldsymbol{p}) \leq 0, \tag{3.36}$$

其中 $d(\boldsymbol{p})$ 为位置 $\boldsymbol{p}$ 到最近障碍物的距离，该数值可以通过符号欧式距离场[89] (Euclidean Signed Distance Fields, ESDF) 获取。而安全阈值 d_{thr} 应该根据无人机的实际半径设置。该约束关于轨迹参数 $\boldsymbol{c}_m$ 和 T_m 的梯度可以由以下公式求出，

$$\frac{\partial \psi_o}{\partial \boldsymbol{c}_m} = -\nabla d(\boldsymbol{p})\beta(t)^T, \tag{3.37a}$$

$$\frac{\partial \psi_o}{\partial T_m} = -\dot{\boldsymbol{p}}^T \nabla d(\boldsymbol{p}), \tag{3.37b}$$

其中 $\nabla d(\boldsymbol{p})$ 为符号欧式距离场在 $\boldsymbol{p}$ 处的梯度。

5) 编队性能指标 ψ_f：与之前的推导方法不同，由于涉及到多个无人机，因此需要定义全局时间 τ。在第 m 段轨迹上，第 k 个采样点的全局时间为：

$$\tau = T_0 + T_1 + \cdots + T_{m-1} + \frac{k}{\kappa_m} T_m, \tag{3.38}$$

其中 T_0 为智能体 i 轨迹的起始时间。为了让集群形成编队，应该使得局部编队性能指标尽可能小，因此惩罚编队性能指标的大小，

$$\psi_f = \frac{1}{2} \sum_{j \in \mathcal{N}_i} \Big(p_i(\tau) - p_j(\tau)\Big)^T P_{g_{ij}^*} \Big(p_i(\tau) - p_j(\tau)\Big) \leq 0. \tag{3.39}$$

编队性能指标相对于多项式系数 $\boldsymbol{c}_m$ 的梯度为，

$$\frac{\partial \psi_f}{\partial \boldsymbol{c}_m} = \beta(\frac{k}{\kappa_m} T_m) \sum_{j \in \mathcal{N}_i} (p_i - p_j)^T P_{g_{ij}^*}^T, \tag{3.40}$$

编队性能指标相对于多项式时长 T_l 的梯度为，

$$\frac{\partial \psi_f}{\partial T_l} = \begin{cases} \dfrac{k}{\kappa_m} \displaystyle\sum_{j \in \mathcal{N}_i} \Big(p_i - p_j\Big)^T P_{g_{ij}^*} \Big(\dot{p}_i - \dot{p}_j\Big), & l = m, \\ -\displaystyle\sum_{j \in \mathcal{N}_i} \Big(p_i - p_j\Big)^T P_{g_{ij}^*} \dot{p}_j, & l < m, \end{cases} \tag{3.41}$$

6) 协同避碰约束 ψ_r：为了保证无人机之间不发生碰撞，无人机之间的距离应该大于两倍安全阈值，因此设计以下约束：

$$\psi_r = \frac{1}{2} \sum_{j \in \mathcal{N}_i} \Big(4d_{thr}^2 - \|p_i(\tau) - p_j(\tau)\|^2\Big) \leq 0, \tag{3.42}$$

该协同避碰约束对于多项式系数 $\boldsymbol{c}_m$ 的梯度为，

$$\frac{\partial \psi_r}{\partial c_m} = -\beta(\frac{k}{\kappa_m} T_m) \sum_{j \in \mathcal{N}_i} (p_i - p_j)^T, \tag{3.43}$$

24029602

该协同避碰约束对于多项式时长 T_l 的梯度为，

$$\frac{\partial \psi_r}{\partial T_l} = \begin{cases} -\dfrac{k}{\kappa_m} \displaystyle\sum_{j \in \mathcal{N}_i} \Big(p_i - p_j\Big)^T \Big(\dot{p}_i - \dot{p}_j\Big), & l = m, \\ \displaystyle\sum_{j \in \mathcal{N}_i} \Big(p_i - p_j\Big)^T \dot{p}_j. & l < m, \end{cases} \tag{3.44}$$

至此，所有性能指标、约束条件及其梯度都已给出。

轨迹优化问题 (3.21) 是一个无约束的非线性优化问题，其中总是存在局部最小值。可以通过基于梯度的优化算法来找到这些局部最小值，例如最速梯度下降算法、L-BFGS 算法[90]等。

由于这些解是局部极小值，因此会存在轻微不满足安全约束 ψ_o 和 ψ_r 的情况。然而，在实际的编队导航中，这些约束存在轻微违反是可以接受的。并且，通过设置安全惩罚 ψ_o 和 ψ_r 的权重大于编队性能指标 ψ_f 的权重，安全约束将被优先考虑，集群将会偏离期望的编队以确保安全，这也是本文希望看到的结果。此外，在实际场景的编队导航过程中，设置相对较大的安全阈值 d_{thr} 以提供更大的安全余量也是一种可接受的解决方案。

通常而言，由于编队性能指标（公式3.7）相对于平移和缩放的不变性，问题（公式3.21）的可行解并不唯一。例如，如果在轻微平移和缩放后仍然没有与障碍物发生碰撞的编队，对应的解也是可行的。解的不唯一性不会影响本文所提出算法的有效性，因为所有的可行解都是可以接受的。

3.4　数值仿真验证

在本节中，将设计数值仿真验证本文算法的有效性。同时为了突出算法在编队队形形成上的有效性，所提出算法将与目前文献中较为先进的算法进行仿真上的对比。

为了比较不同算法的性能，需要使用量化的指标对算法进行评价。目前文献中较为先进的算法如[48] 等，不仅可以利用编队的整体放缩、平移进行避障，还可以对编队进行整体旋转来避障。本文提出的编队性能指标 (3.7) 对于编队的整体放缩、平移机动不敏感，但却对于编队的整体旋转机动是敏感的，因此不适用于对算法进行验证。

为了消除放缩、平移和旋转机动对于编队性能的影响，本文采用编队偏移误差 $\boldsymbol{e}_{dev}$ 和编队相似性误差 e_{sim} 来评价编队导航的性能。给定一个编队构型 $\boldsymbol{p}^*$ 和一个实际的编队构型 $\boldsymbol{p}$，则第 i 个智能体的编队偏移误差为：

$$\boldsymbol{e}_{i,dev} = \boldsymbol{p}_i - (s^* \mathbf{R}^* \boldsymbol{p}_i^* + \rho^*), \tag{3.45a}$$

24029602

$$s^*, \mathbf{R}^*, \rho^* = \arg\min_{s,\mathbf{R},\rho} \sum_{i \in \mathcal{V}} \|\boldsymbol{p}_i - (s\mathbf{R}\boldsymbol{p}_i^* + \rho)\|^2, \tag{3.45b}$$

其中 $\mathbf{R}^* \in SO(d)$, $\rho^* \in \mathbb{R}^d$ 和 $s^* \in \mathbb{R}$ 分别描述了实际编队构型与其最接近的期望编队构型之间的旋转、平移和放缩比例。注意到，对于非线性优化问题 (3.45b) 的最优解是唯一的，且可以由文献[91] 中的方法算出闭式解。编队整体偏移误差 $\boldsymbol{e}_{dist}$ 由以下公式给出：

$$e_{dist} = \sum_{i \in \mathcal{V}} \|\boldsymbol{e}_{i,dev}\|_2. \tag{3.46}$$

编队偏移误差的上界 $\overline{e_{dev}}$ 和下界 $\underline{e_{dev}}$ 则由以下公式给出：

$$\overline{e_{dev}} = \max\left(\boldsymbol{e}_{1,dev}, \cdots, \boldsymbol{e}_{N,dev}\right), \tag{3.47a}$$

$$\underline{e_{dev}} = \min\left(\boldsymbol{e}_{1,dev}, \cdots, \boldsymbol{e}_{N,dev}\right). \tag{3.47b}$$

而编队的相似性误差 e_{sim} 则由以下公式给出：

$$e_{sim} = \left\|\hat{\mathbf{L}}^* - \hat{\mathbf{L}}\right\|_F^2, \tag{3.48}$$

其中 $\|\cdot\|_F$ 为弗罗比尼乌斯范数 (Frobenius Norm)。而 $\hat{\mathbf{L}}$ 为以下归一化后的平方距离拉普拉斯矩阵：

$$\hat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}, \tag{3.49}$$

其中 $\mathbf{A} = \{a_{ij}\}$ 和 $\mathbf{D} = \{d_{ij}\}$ 为对应的邻接矩阵和度矩阵，其定义为：

$$a_{ij} = \begin{cases} \|p_i - p_j\|^2, & (i,j) \in \mathcal{E} \\ 0, & (i,j) \notin \mathcal{E} \end{cases}, \tag{3.50a}$$

$$d_{ij} = \begin{cases} \sum_{j=0}^{n} a_{ij}, & i = j \\ 0, & i \neq j \end{cases}. \tag{3.50b}$$

本小节将在编队保持、编队形成和综合场景中的编队导航三方面验证本文所提出算法的有效性，并与文献[48] 中的算法进行对比。

3.4.1 编队队形保持

本文采用蒙特卡洛模拟测试的方法统计 100 次仿真中的指标来获取具有统计学意义的测试结果，从而消除随机放置障碍物对于编队整体偏移误差 $\boldsymbol{e}_{dist}$ 和编队相似性误差 e_{sim} 这两个指标造成的影响。

24029602

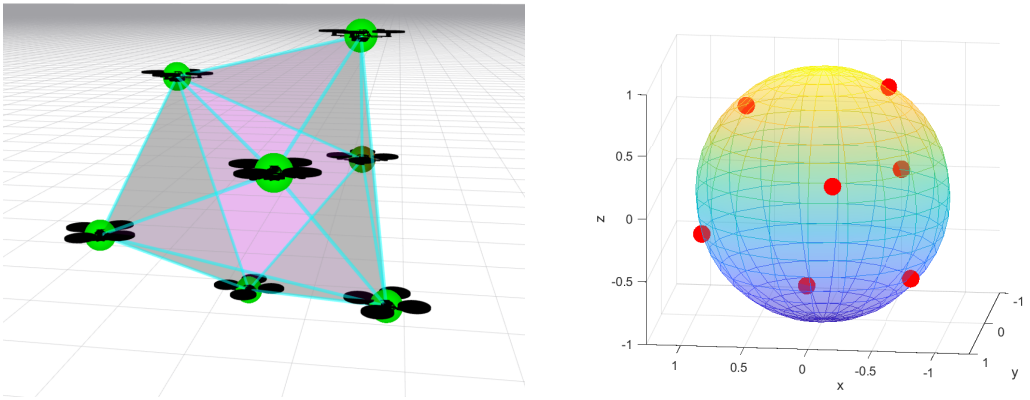
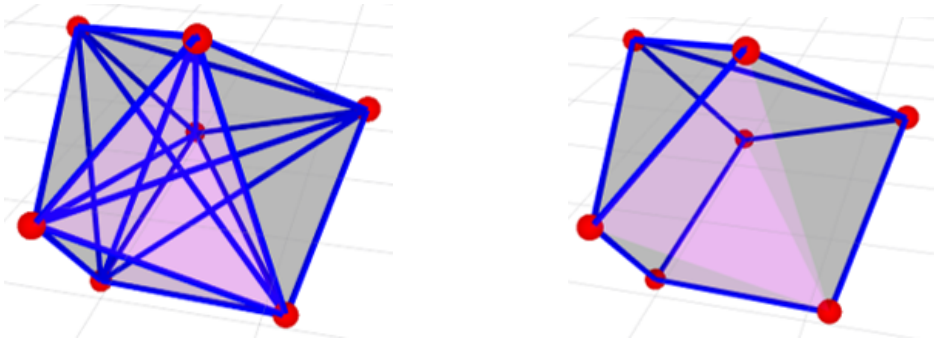


图 3.4　数值仿真中使用的七架无人机组成的球形编队队形。

在每一次仿真中，都会在长 50 米，宽 15 米，高 5 米的环境中放置 40 个直径为 0.5 米的圆柱形障碍物、30 个内径为 2 米的圆环形障碍物。七个无人机将从障碍物密集区域的一边以 0.5 米每秒的最高速度限制向障碍物密集区域的另一边飞行。七个无人机在初始位置处已经形成了期望的球形编队队形 (队形如图 3.4 所示)，并将在导航的过程中尽可能地尝试保持该球形编队队形直到到达目标位置。如果在整个导航的过程中，所有无人机都不和障碍物发生碰撞，且编队的相似性误差始终小于 0.2，则认为该次导航是成功的。



(a) 七架无人机编队的完全图。

(b) 七架无人机编队的一种拉曼图。

图 3.5　七架无人机编队数值仿真中使用的通讯图。

为了体现本文提出算法在分布式上的优越性，测试将分别采用以下两种通讯图来进行测试：

1. 完全图，即所有智能体两两之间均能够通讯的图，如图 3.5(a) 所示；
2. 拉曼图[92] (Laman Graph)，一种能够保证方位刚性但需要更少通讯的图，如图 3.5(a) 所示。

测试的结果被总结于表 3.1 中。

可从表 3.1 中总结出本文算法的表现。当通讯图为完全图时，本文所提出算法和文献[48]中的算法在成功率、编队整体偏移误差 e_{dist} 和编队相似性误差 e_{sim} 三个指标上都有着相似的表现。

24029602

表 3.1 算法性能对比：编队队形保持

算法	通讯图	成功率	编队整体偏移误差 e_{dist}	编队相似性误差 e_{sim}
文献[48] 中的算法	完全图	94%	0.2792 米	0.025
	拉曼图	76%	0.7326 米	0.162
本文所提出算法	完全图	95%	**0.2073** 米	**0.023**
	拉曼图	94%	0.2345 米	0.028

然而，当通讯图为非完整图时，如拉曼图，文献[48]中的算法出现了明显的性能下降。文献[48]中的算法通过直接优化编队相似性误差 e_{sim} 来实现编队。一方面，编队相似性误差 e_{sim} 相对于实际编队构型 $\boldsymbol{p}$ 不是凸的，因此存在许多局部极小值点，这些局部极小值点对应的构型并不是期望的编队构型；另一方面，目前并无理论保证编队相似性误差 e_{sim} 在非完整图下是能够进行有效优化的。因此，直接优化编队相似性误差 e_{sim} 会导致集群收敛到非期望的编队构型上，因此导致导航失败，并导致更高的编队整体偏移误差 e_{dist} 和编队相似性误差 e_{sim}。

本文算法优化的编队性能指标 ψ_f (3.7) 具有分布式和几何形状唯一的特点，且这些特点均有理论上的保证。因此，当通讯图为拉曼图时，本文所提出的算法依然保持着良好的性能指标。

一百次蒙特卡洛仿真中的一个例子如图 3.6(a)-3.6(d)所示。

3.4.2 编队队形形成

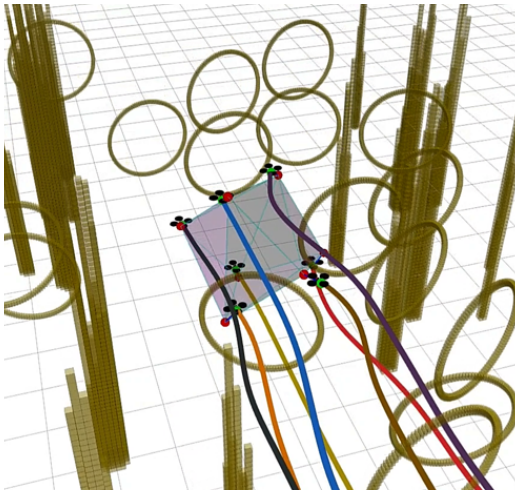
对于编队形成，本文同样使用蒙特卡洛测试的方法统计 100 次仿真中的指标来获取具有统计学意义的测试结果，从而消除无人机初始位置对于编队形成成功率造成的影响。为了突出本文提出算法能使集群从任意初始位置出发，最终收敛到期望队形，本文算法将与文献[48]中方法进行比较。

在每次仿真中，七个无人机被随机放置到无障碍物的环境中并向前飞行以形成编队。编队队形中的位置在开始导航前被随机地分配给每个无人机，且本文假设在仿真的过程中被设置为完整图。并且，本文定义一次测试被称为：

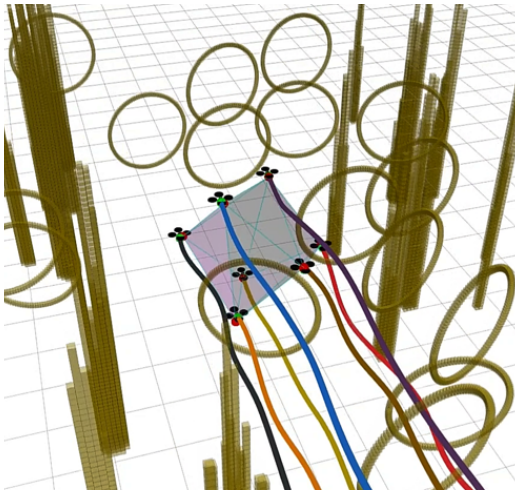
- 死锁：如果 $e_{sim} > 0.2$，且不再增加；同时对应的梯度 $\partial e_{sim}/\partial p \neq \mathbf{0}$ 或者 $\partial \psi_f/\partial p \neq \mathbf{0}$。
- 变形：如果 $e_{sim} > 0.2$，且不再增加；同时对应的梯度 $\partial e_{sim}/\partial p \neq \mathbf{0}$ 或者 $\partial \psi_f/\partial p \neq \mathbf{0}$。
- 成功：如果在 $10s$ 内，e_{sim} 始终小于 0.2，且无人机集群没有陷入到死锁中。

注释 3.1： 判断依据 $e_{sim} > 0.2$，是通过对所有仿真数据使用最大最小间距算法

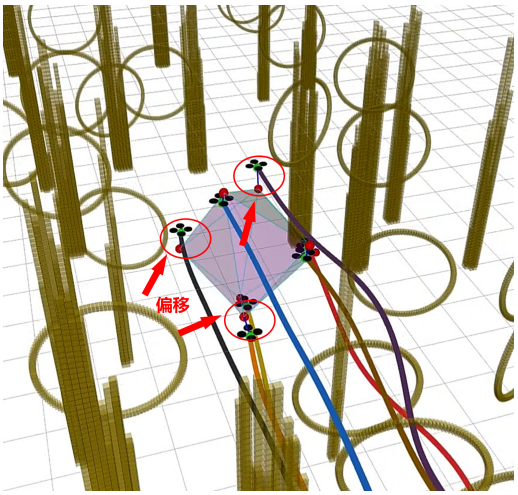
24029602



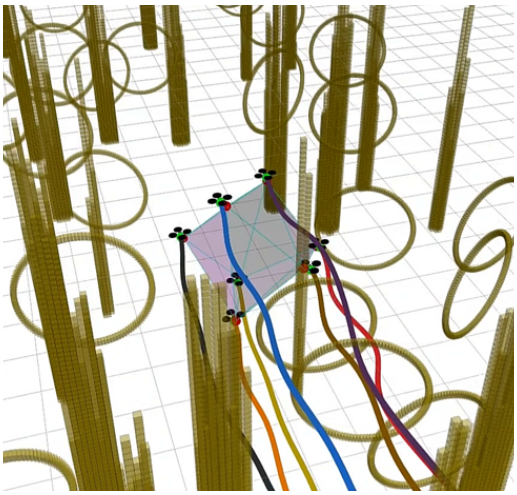
(a) 文献[48] 中提出算法使用完全图通信时编队保持的结果，编队在导航中未发生过多偏移。



(b) 本文所提出算法使用完全图通信时编队保持的结果，编队在导航中未发生过多偏移。



(c) 文献[48] 中提出算法使用拉曼图通信时编队保持的结果，编队在导航中发生了较大的偏移量。



(d) 本文所提出算法使用拉曼图通信时编队保持的结果，编队在导航中未发生过多偏移。

图 3.6　一百次编队保持蒙特卡洛仿真中的一个例子。

进行二分类得到的。

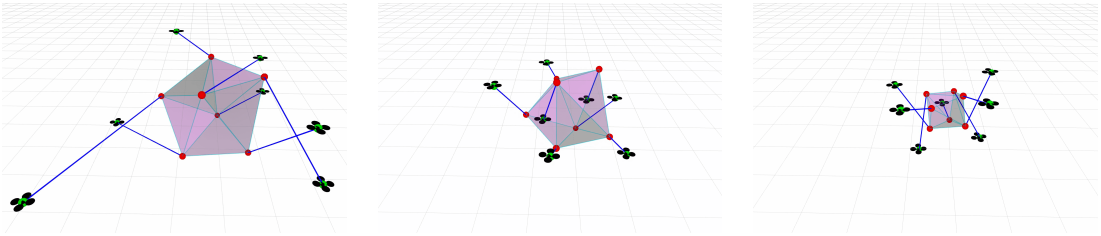
表 3.2　算法性能对比：编队队形形成

算法	成功率	变形率	死锁率
文献[48]中的算法	62%	3%	35%
本文提出的算法	**88%**	**0%**	**12%**

本文将一百次蒙特卡洛仿真的结果总结于表 3.2 中。文献[48]中的算法在编队队形形成仿真中的成功率为 62%，且存在少部分情况收敛到变形的队形上，死锁

24029602

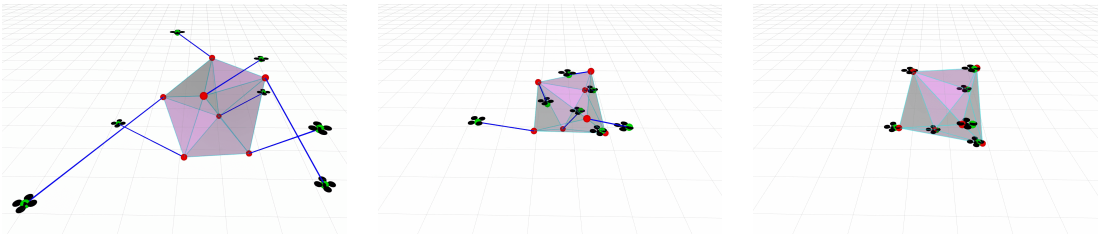
率为 35% 本文所提出算法在编队队形形成仿真中实现了 88% 的成功率，同时死锁率为 12%。因此本文算法在队形形成上优于文献[48]中的算法。



(a) $t = 0s$，集群处于初始位置。 (b) $t = 10s$，集群向期望编队收敛。 (c) $t = 20s$，集群陷入到死锁当中。

图 3.7　文献[48] 中提出算法在编队形成上的结果。

图 3.7(a)-3.7(c) 展示了文献[48] 中提出算法在一百次蒙特卡洛仿真中的一个结果，集群最终陷入到了死锁当中。在该案例中，编队相似性误差 e_{sim} 在值 0.32 处停止，且偏导数 $\partial e_{sim}/\partial \boldsymbol{p} \neq \mathbf{0}$，这意味着编队需求与避障需求相矛盾。



(a) $t = 0s$，集群处于初始位置。 (b) $t = 10s$，集群向期望编队收敛。 (c) $t = 20s$，集群收敛到期望队形。

图 3.8　本文所提出算法在编队形成上的结果。

图 3.8(a)-3.8(c) 展示了本文所提出算法在一百次蒙特卡洛仿真中的一个结果，在该案例中，集群最终形成了期望球形编队队形。

3.4.3　综合场景中的编队导航

为了验证本文算法存在编队队形形成、编队队形保持和避障的能力，本文设计了复杂障碍物环境中的综合测试。在仿真中七台无人机被放置在一个中心位置 $x = 0$ 米的方形阵列中，它们将以 0.5 米每秒的最高速度限制向障碍物密集区域的另一边飞行。在飞行的过程中，无人机集群需要保持球形编队队形，通讯网络为拉曼图。在综合场景，总共有三面带有缺口的墙，它们分别位于 $x = 8$ 米，$x = 18$ 米和 $x = 28$ 米。这些墙意在验证本文所提出算法运用放缩机动和平移机动躲避障碍物的能力。综合场景的 $x = 38$ 米到 $x = 44$ 米和 $x = 55$ 米到 $x = 61$ 米位置存在两个障碍物密集区域，每个区域中存在 9 个直径为 0.5 米的圆柱形障

24029602

碍物和 4 个内径为 2 米的圆环形障碍物。这些障碍物密集区域意在验证本文算法的队形保持能力。综合场景的示意图如图 3.9 和 3.10 所示。

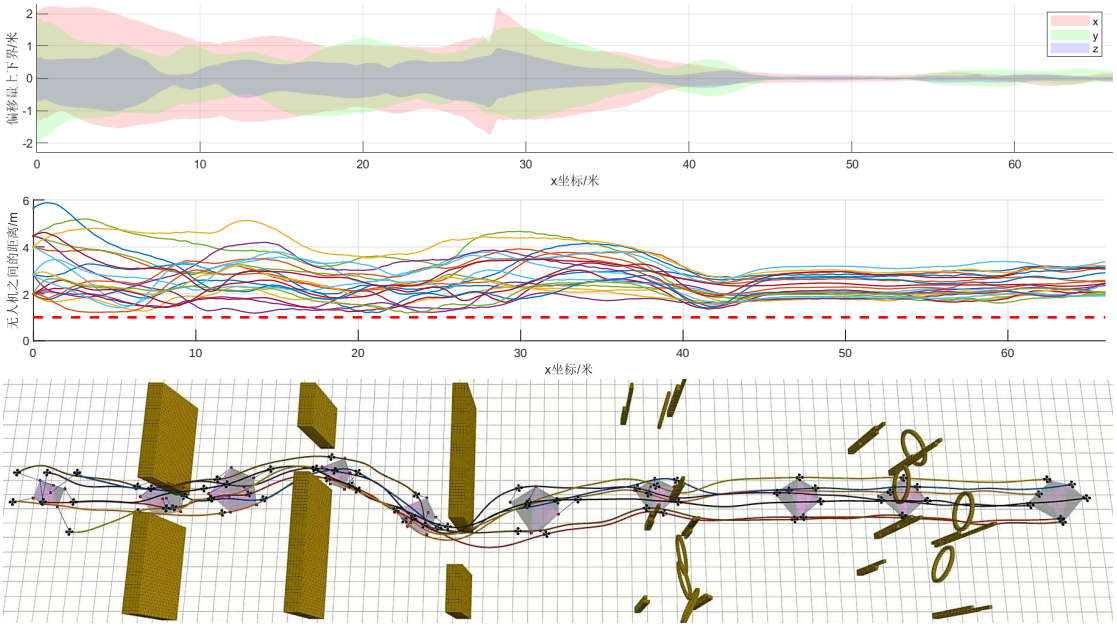


图 3.9　文献[48] 所提出方法在综合场景中的编队导航误差图，无人机相对距离图和轨迹图。

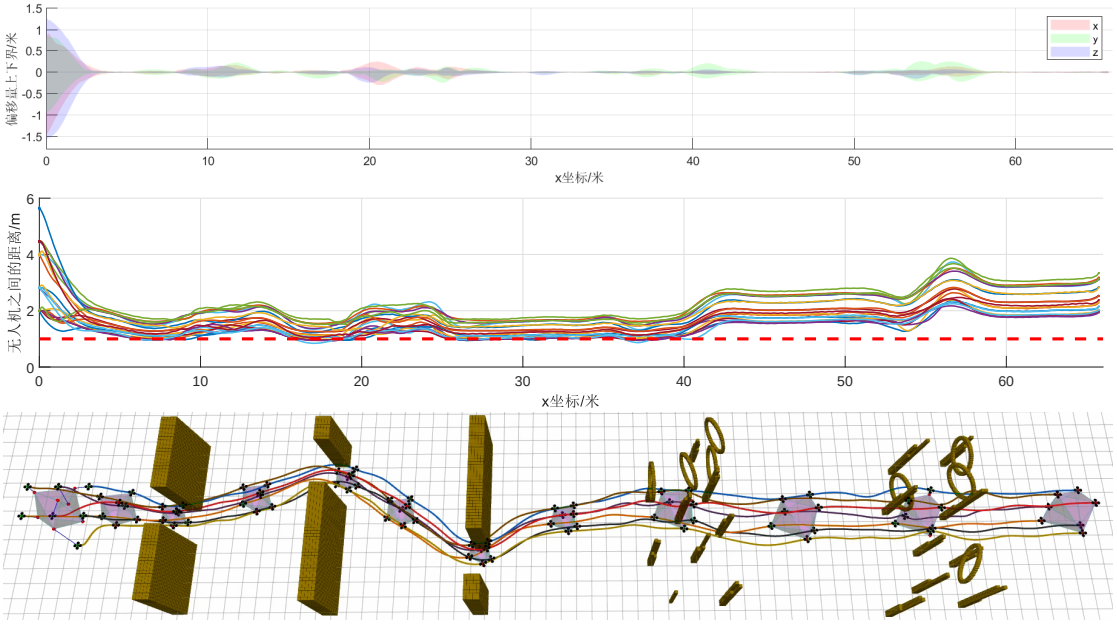


图 3.10　本文所提出方法在综合场景中的编队导航误差图，无人机相对距离图和轨迹图。

文献[48] 中提出算法的仿真结果如图 3.9 所示，由于受到环境中障碍物的干扰，集群未能快速地形成球形编队，因此只能以散乱的集群形式穿过墙上的缺口；当到达较为空旷的场景中后，集群才成功形成编队，后利用轻微的变形来躲避障碍物密集区域中的障碍物并到达目标位置。在整个导航的过程中，集群的编队偏移误差的上界 $\overline{\boldsymbol{e}_{dev}}$ 和下界 $\underline{\boldsymbol{e}_{dev}}$ 在各个维度上的分量如图 3.9 的第一个子图所示，由于一开始无人机集群无法快速形成编队，集群的编队偏移误差较大。在

24029602

整个导航的过程中，无人机之间的相对距离如图 3.10 的第一个子图所示，其中红色的虚线为 $2d_{thr} = 1.0m$，可见，在整个过程中所有距离都在 $2d_{thr}$ 以上，因此没有发生碰撞。

本文所提出算法的仿真结果如图 3.10 所示，集群首先成功地形成了球形编队，紧接着动态地使用队形整体放缩机动和队形整体平移机动来穿过墙上的缺口，最后利用轻微的变形来躲避障碍物密集区域中的障碍物并到达目标位置。在整个导航的过程中，集群的编队偏移误差的上界 $\overline{\boldsymbol{e}_{dev}}$ 和下界 $\underline{\boldsymbol{e}_{dev}}$ 在各个维度上的分量如图 3.10 的第一个子图所示，尽管无人机集群无法在整个过程中都保持零偏移，但偏移的范围能够满足绝大多数场景中的需求。在整个导航的过程中，无人机之间的相对距离如图 3.10 的第一个子图所示，其中红色的虚线为 $2d_{thr} = 1.0m$，可见，在整个过程中所有距离都在 $2d_{thr}$ 以上，因此没有发生碰撞。

3.5 小结

本章研究了障碍物密集环境中的多机器人编队轨迹优化问题。首先利用相对方位约束相对于编队整体平移和编队整体放缩的不变性，设计了一种分布式的编队性能指标。随后依据方位刚性理论从理论上证明了该分布式编队性能指标的所有极小值点均为期望的编队构型，即这些编队构型之间仅相差了一个平移量和放缩因子。

为了系统地考虑机器人集群的编队性能需求、避障需求和协同避碰需求，本章提出了一种新颖的在线轨迹优化算法。通过变量转化消解了轨迹的连续性约束，使得在新变量下描述的轨迹始终满足一定的连续性要求；通过精确罚函数法将约束函数转化为惩罚函数，将原有含约束的轨迹优化问题转变为无约束的轨迹优化问题；本章详细描述了一般惩罚函数对于轨迹参数的求导法则，给出了无约束轨迹优化问题的梯度求解方法。

为了演示本章所提出编队轨迹优化算法的有效性，本章以四旋翼无人机在障碍物稠密环境中的编队导航为案例进行仿真验证，并于同期学术前沿编队导航算法在编队形成、编队保持和综合场景验证三方面进行对比，证明了本文提出的算法的有效性和优越性。

本章通过引入相对方位约束来实现编队导航，充分利用了编队机动能力进行避障，为实际场景中的编队导航提供了新的思路和解决方案。

24029602

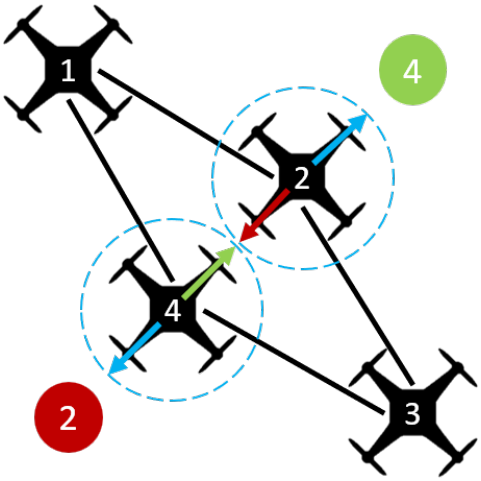
第 4 章　基于势博弈的编队任务分配方法

4.1　引言

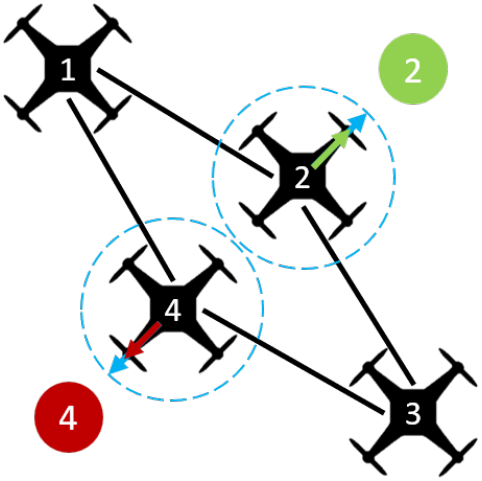
从上一章中的数值仿真中可以看出，仅利用轨迹优化算法依然会存在让集群陷入到死锁的风险。同时观察到，当集群严重偏离期望编队队形时，发生死锁的概率会变大。在本文所提出的编队轨迹优化框架中，导致死锁的主要原因为局部编队性能指标 ψ_f 与协同避障约束之间 ψ_r 之间发生了冲突。一方面，集群中的智能体希望最小化局部编队性能指标 ψ_f 而倾向于沿着 $-\partial\psi_f/\partial\boldsymbol{p}$ 的方向移动；另一方面，当集群中智能体的距离小于安全阈值时，智能体为了保证安全，即最小化编队惩罚项 ψ_r，而倾向于沿着 $-\partial\psi_r/\partial\boldsymbol{p}$ 的方向移动。因为编队性能指标和安全约束对于编队导航而言都是重要的，函数 ψ_f 和 ψ_r 被加权后放到轨迹优化问题 (3.21) 中进行同时优化。因此，有一定的概率会发生以下情况：

$$\lambda_f\frac{\partial\psi_f}{\partial\boldsymbol{p}}+\lambda_r\frac{\partial\psi_r}{\partial\boldsymbol{p}}=0, \tag{4.1}$$

其中，λ_f 和 λ_r 分别为 ψ_f 和 ψ_r 的权重。在式子 (4.1) 中，ψ_f 的梯度将 ψ_r 的梯度抵消掉了，因此集群会保持不动发生死锁。一个具体的例子由图 4.1 给出。



(a) 任务分配前发生死锁：智能体 2 正在向着红色的目标前进，然而 $\partial\psi_{2,f}/\partial\boldsymbol{p}_2$ 将 $\partial\psi_{2,r}/\partial\boldsymbol{p}_2$ 抵消掉了，因此智能体 2 无法前进；智能体 4 也存在相似的状况，因此发生了死锁。



(b) 任务分配后死锁消解：智能体 2 的目标被重新分配为绿色的目标，$\partial\psi_{2,f}/\partial\boldsymbol{p}_2$ 与 $\partial\psi_{2,r}/\partial\boldsymbol{p}_2$ 之间不再有冲突；智能体 4 也存在相似的状况，因此死锁被消解了。

图 4.1　死锁和分配的示意图。蓝色的箭头代表着协同避碰约束的梯度 $\partial\psi_{i,r}/\partial\boldsymbol{p}_i$，红色和绿色的箭头为局部编队性能指标的梯度 $\partial\psi_{i,f}/\partial\boldsymbol{p}_i$。

死锁是编队导航面临的一个具有挑战性的问题，尤其是在障碍物密集的环

24029602

境中。文献[54]设计了一种基于规则的避碰机制，当两个机器人距离小于规定的安全距离时，该机制会驱使它们向相反的方向。然而，采用这种机制进行避障可能会使得机器人集群陷入第二类死锁[93]。为了生成安全的编队轨迹，文献[48]将碰撞惩罚和编队相似性误差加权后同时优化。然而，当机器人相互阻塞时，碰撞惩罚和编队相似性误差的梯度可能相互抵消，导致死锁。为了消解死锁，引入诸如任务分配之类的高层次协调是十分必要的。基于以上讨论，本文将采用任务分配的思想来解决死锁问题，并提出了一种直接优化编队性能指标的分布式贪婪任务分配算法。

4.2 博弈论预备知识

在给出具体任务分配算法之前，有必要给出与博弈论有关的基础定义及基础理论知识。

定义 4.1 (纯策略博弈)： 一个纯策略博弈被定义为三元组 $(\mathcal{R}, \mathcal{S}, \mathcal{U})$，其中：

1. $\mathcal{R} = \{r_1, r_2, \cdots, r_N\}$ 为拥有有限个元素的参与者 (Player) 集合，其中符号 r_i 表示第 i 个参与者。
2. $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_N\}$ 为拥有有限个元素的策略 (Strategy) 集合，其中集合 $\mathcal{S}_i$ 表示第 i 个参与者的可行策略集合，注意 $\mathcal{S}_i$ 也为有限集合。
3. $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$ 为拥有有限个元素的效用函数 (Utility Function) 集合，其中函数 $u_i : \mathcal{S}_i \to \mathbb{R}$ 为第 i 个参与者的效用函数。

在博弈论中，策略组合 (Strategy Profile) 是指所有参与者在博弈中选择的策略构成的组合。记策略组合为 $\boldsymbol{s} = (s_1, s_2, \cdots, s_N)$, 其中 $s_i \in \mathcal{S}_i$ 为第 i 个参与者所选择的策略。特别的，记 $\boldsymbol{s}_{-i}$ 为除了第 i 个参与者外所有参与者选择策略的组合，因此策略组合 $\boldsymbol{s}$ 可以被写为 $\boldsymbol{s} = (s_i, \boldsymbol{s}_{-i})$。按照以上记法，可给出纳什均衡的定义。

定义 4.2 (纯策略纳什均衡)： 一个策略组合 $\boldsymbol{s}^* = (s_1^*, s_2^*, \cdots, s_N^*)$ 被称为纯策略纳什均衡，当且仅当没有任何一个参与者可以通过仅改变自身策略来提升自身的效用函数。使用更准确的数学语言描述：

$$u_i(s_i^*, \boldsymbol{s}_{-i}^*) \geq u_i(s_i, \boldsymbol{s}_{-i}^*), \quad \forall r_i \in \mathcal{R}, \quad \forall s_i \in \mathcal{S}_i, \quad s_i^* \neq s_i. \tag{4.2}$$

在博弈论中，有一类具有特殊结构的博弈类型被称为严格势博弈。通过利用其特殊的结构可以设计算法快速找到纯策略纳什均衡。

24029602

定义 4.3 (严格势博弈)： 一个博弈被称为严格势博弈 (Exact Potential Game)，如果存在一个函数 $\Phi : \mathcal{S} \to \mathbb{R}$ 满足以下性质:

$$\begin{aligned} u_i(s_i', \boldsymbol{s}_{-i}) - u_i(s_i, \boldsymbol{s}_{-i}) = \Phi(s_i', \boldsymbol{s}_{-i}) - \Phi(s_i, \boldsymbol{s}_{-i}), \\ \forall r_i \in \mathcal{R}, \qquad \forall s_i',\ s_i \in \mathcal{S}_i. \end{aligned} \tag{4.3}$$

函数 Φ 被称为势函数 (Potential Function)。

根据以上定义可知，对于势博弈而言，虽然势函数具有全局性 (即势函数是相对于所有参与者定义的)，但策略改变造成的局部效用变化等于势函数的变化。这揭示了势博弈具有局部变化反应全局的特点，该特点为分布式优化算法的设计提供了一个良好的思路。除此之外，势博弈还具有很多良好的性质。

引理 4.1 (纯策略纳什均衡的存在性)： 严格势博弈存在至少一个纯策略纳什均衡策略。

引理 4.2 (势博弈有限下降)： 如果严格势博弈的策略集合 $\mathcal{S}$ 是有限的，那么势函数 Φ 只能被最小化有限次。且每个参与者都按照最小化效用 u_i 的方法给出策略，则最终得到的策略组合为该博弈的纳什均衡策略。

4.3　编队任务分配问题建模与求解

任务分配的目的为找到一个任务分配规则 α 将每一个智能体唯一地分配到一个编队网络节点上，同时最小化某些就特定的指标。在本文的框架中，为了提高编队的成功率，所设计的任务分配规则应该满足以下两个需求。

1. 提高性能。更准确地说，任务分配需要使得初始队形尽可能地和期望队形相似，即，尽可能地最小化编队性能指标 (3.7)；
2. 消解死锁。更准确地说，在任务分配之后形成的新的编队网络应该改满足以下不等式：

$$\lambda_f \frac{\partial \psi_f}{\partial \boldsymbol{p}} + \lambda_r \frac{\partial \psi_r}{\partial \boldsymbol{p}} \neq \mathbf{0}. \tag{4.4}$$

为了满足以上需求，本章将提出一种基于博弈论的分布式贪婪任务分配算法。

4.3.1　任务分配问题建模

给定一个期望的编队网络 $(\boldsymbol{p}^*, \mathcal{G}^*)$ 以及智能体集群的实际构型 $\boldsymbol{p}$，任务分配的目标为最小化以下整体编队性能指标 ψ_f，

$$\min_{\alpha \in \mathbb{S}_n} \quad \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \delta_{ij}^2, \tag{4.5a}$$

24029602

$$\delta_{ij} = \left\| \mathbf{P}_{\boldsymbol{g}_{ij}^*}(\boldsymbol{p}_i - \boldsymbol{p}_j) \right\|_2, \tag{4.5b}$$

因为编队中节点的数量等于智能体的数量，因此任务分配规则 $\alpha \in \mathbb{S}_n$，其中 $\mathbb{S}_n$ 为从集合 $\{1, 2, \cdots, N\}$ 到其自身的对称群映射。

问题 (4.5a) 实际上为二次分配问题 (Quadratic Assignment Problem, QAP)，二次分配问题是 NP 难问题中的最难以解决的一类[94]。在现有文献中，解决二次分配问题的方法主要有精确算法和启发式算法两类。精确算法通过在整个解空间中进行搜索来找到全局最优解。然而，问题的解空间通常随着问题的维度迅速增大，因此精确算法在求解大规模问题时常常会面临难以负担的计算开销，所以不适用于问题的实时求解。启发式算法则通过启发函数来引导算法的搜索方向，从而加速问题的求解。启发式算法通常无法保证全局最优性，但具有更好的算法时效性。为了保证算法的实时性本文将用最优性来换取计算效率，因此本章节提出的算法属于启发式算法。

4.3.2 任务分配问题求解

本小节提出解决任务分配问题 (4.5a) 的一种分布式贪婪算法。使用上标 k 表示任务分配算法的第 k 次迭代，例如 α^k 和 ψ_f^k 分别表示在第 k 次迭代时的任务分配规则和编队性能指标。考虑在第 k 次迭代时，编队中的节点 a 和节点 b 交换被分配给它们的智能体，那么该次交换造成的整体性能指标变化为：

$$\begin{aligned}
&\Delta\psi_f(p) = \psi_f^k(p) - \psi_f^{k-1}(p) \\
=&\frac{1}{2}\Big(\sum_{i\in n_a}\Delta\delta_{ia}^2 + \sum_{i\in n_b}\Delta\delta_{ib}^2 + \sum_{j\in n_a}\Delta\delta_{aj}^2 + \sum_{j\in n_b}\Delta\delta_{bj}^2\Big) \\
=&\sum_{j\in n_a}\Delta\delta_{aj}^2 + \sum_{j\in n_b}\Delta\delta_{bj}^2 \\
=&2\Delta\psi_{a,f}(p) + 2\Delta\psi_{b,f}(p)
\end{aligned}, \tag{4.6}$$

其中 $\Delta\delta_{ij}^2$ 为节点 i 和节点 j 之间的方位误差的变化，即，

$$\Delta\delta_{ij}^2 = \|P_{g_{ij}^*}(p_{\alpha^k(i)} - p_{\alpha^k(j)})\|_2^2 - \|P_{g_{ij}^*}(p_{\alpha^{k-1}(i)} - p_{\alpha^{k-1}(j)})\|_2^2. \tag{4.7}$$

由 (4.6) 可以看出，尽管每个节点的局部编队性能指标是耦合的，但局部编队性能指标的下降直接反映了全局编队性能指标的下降，该特点可以被运用到任务分配算法的设计中。

因此，本文提出了一个分布式贪婪算法来解决任务分配问题 (4.5a)，该算法的伪代码被总结在算法 4.1 中。为了方便起见，定义 γ 为任务分配规则 α 的逆映射，即, $i = \gamma(\alpha(i))$。同时定义智能体 i 的邻居集合定义为所有能够与之通讯的

24029602

智能体，

$$\tilde{\mathcal{N}}_i = \left\{ j \mid j = \alpha(v),\ v \in \mathcal{N}_{\gamma(i)} \right\}. \tag{4.8}$$

请注意，该集合与节点 i 的邻居集合 $\mathcal{N}_i$ 是有区别的。

算法 4.1　在第 $k+1$ 次迭代中第 i 个智能体的贪婪编队任务分配算法

1 向所有邻居 $j \in \tilde{\mathcal{N}}_i^k$ 发送自己的轨迹 $\boldsymbol{p}_i^k(t)$ 和自己所分配的节点 $\gamma^k(i)$;
2 从所有邻居 $j \in \tilde{\mathcal{N}}_i^k$ 接收邻居的轨迹 $\boldsymbol{p}_j^k(t)$ 和邻居所分配的节点 $\gamma^k(j)$;
3 for $a \in \tilde{\mathcal{N}}_i^k$ do
4 　初始化局部方位角度的变化 $\Delta\psi_{i,a} = 0$ 并更新当前时间 $t = t_{now}$;
5 　for $b \in \tilde{\mathcal{N}}_i^k / \{a\}$ do
6 　　根据 (4.7) 计算出局部编队性能指标的变化，
　　$\Delta\psi_{i,a} = \|P_{g^*_{\gamma^k(i)\gamma^k(b)}}(p_a(t) - p_b(t))\|^2 - \|P_{g^*_{\gamma^k(i)\gamma^k(b)}}(p_i(t) - p_b(t))\|^2$;
7 　end
8 end
9 向所有邻居 $j \in \tilde{\mathcal{N}}_i^k$ 发送局部编队性能指标的变化 $\Delta\psi_{i,j}$;
10 从所有邻居 $j \in \tilde{\mathcal{N}}_i^k$ 接收局部编队性能指标的变化 $\Delta\psi_{j,i}$;
11 记录造成最大的性能指标减少的邻居智能体 $z = \arg\min_j \Delta\psi_{i,j} + \Delta\psi_{j,i}$;
12 if $\Delta\psi_{i,z} + \Delta\psi_{z,i} < 0$ then
13 　与 z 交换分配的节点，$\gamma^{k+1}(i) = \gamma^k(z)$;
14 　与 z 交换分配的节点的邻居，$\tilde{\mathcal{N}}_i^{k+1} = \tilde{\mathcal{N}}_z^k$;
15 end

在该算法中，第 i 个智能体首先与其所有邻居智能体 $j \in \tilde{\mathcal{N}}_i^k$ 交换自己的轨迹和所分配节点 (算法 4.1，第 1-2 行)。随后，计算与邻居智能体 j 交换节点后造成的局部编队性能指标变化 $\Delta\psi_{i,j}$ (第 3-8 行)。紧接智能体 i 再次与其所有邻居智能体 $j \in \tilde{\mathcal{N}}_i^k$ 交换自己计算出的局部变化 (第 9-10 行)。最后，根据 (4.6) 检查全局性能指标是否发生了下降，若发生了下降，则选取发生最大下降的智能体进行节点交换 (第 11-15 行)。算法 4.1 的特点总结如下：

1. 算法 4.1 可以完全分布式地实现，因为算法中只用到了邻居智能体的信息；
2. 算法 4.1 的复杂度为 $O(\tilde{\mathcal{N}}_i^2)$。此外，通过该算法的复杂度可以通过使用一些稀疏图，如拉曼图，来进一步降低。

注释 4.1： 在任务分配的过程中，一个智能体有可能同时受到多个邻居智能体的交换节点请求。所以在实际的算法实现中，本文使用了文献[73] 中的双向握手机制来解决该情况。即，当一个智能体受到多个节点交换请求时，该智能体只选

24029602

择其中的一个请求，并向其期望的交换对象发送回复指令。同时，该机制的运用使得本算法是异步的。

注释 4.2 (轨迹优化与任务分配的关系)： 对于单个智能体而言，编队轨迹优化和编队任务分配被重复地执行在两个不同的线程中。其中编队任务分配以大概 $1Hz$ 的频率运行，而轨迹优化算法则以尽可能快的频率运行。一旦发生了任务交换，则相对应的智能体马上以新的任务分配规则初始化轨迹优化问题并求解新的轨迹。

4.3.3 任务分配问题收敛性分析

在本小节将在博弈论视角下对算法 4.1 的收敛性、解的稳定性进行分析。首先注意到公式 (4.6) 反映了局部编队性能指标下降会造成全局性能指标下降的特点，该特点与严格势博弈中效用函数和势函数所具备的性质 (4.3) 十分相似。实际上，任务分配问题 (4.5a) 可以被建模为一个严格势博弈问题，该结论由以下命题给出。

命题 4.1： 任务分配问题 (4.5a) 等价于一个严格势博弈问题。且依据最小化局部性能指标的原则确定每个智能体的节点分配方法，则该问题至少存在一个稳定的解，即，严格势博弈的纳什均衡策略。

证明： 将任务分配问题的边集合 $\mathfrak{E}$ 当作博弈的参与者集合 $\mathfrak{R}$；将是否交换节点 i 和节点 j 的智能体作为博弈参与者 (i,j) 的策略集合 $\mathfrak{S}_{(i,j)}$；定义以下博弈参与者 (i,j) 的效用函数：

$$\begin{aligned} u_{(i,j)} &= \psi_{i,f} + \psi_{j,f} \\ &= \frac{1}{2}\sum_{a\in\mathcal{N}_i}\delta_{ia}^2 + \frac{1}{2}\sum_{b\in\mathcal{N}_j}\delta_{jb}^2; \end{aligned} \tag{4.9}$$

最后定义以下势函数：

$$\Phi = \frac{1}{2}\psi_f = \frac{1}{4}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\delta_{ij}^2. \tag{4.10}$$

根据公式 (4.6)，博弈参与者的效用函数 $u_{(i,j)}$ 和势函数 Φ 存在以下关系，

$$\begin{aligned} &u_{(i,j)}(s'_{(i,j)}, s_{-(i,j)}) - u_{(i,j)}(s_{(i,j)}, s_{-(i,j)}) \\ &= \Delta\psi_{i,f}(\boldsymbol{p}) + \Delta\psi_{j,f}(\boldsymbol{p}) = \frac{1}{2}\Delta\psi_f \\ &= \Phi(s'_{(i,j)}, s_{-(i,j)}) - \Phi(s_{(i,j)}, s_{-(i,j)}). \end{aligned} \tag{4.11}$$

24029602

根据定义 4.3 可知，编队任务分配问题 (4.5a) 等价于一个严格势博弈问题。再由引理 4.1 可知，该编队任务分配问题存在至少一个稳定解，即其对应严格势博弈问题的纳什均衡策略。证明完毕。 □

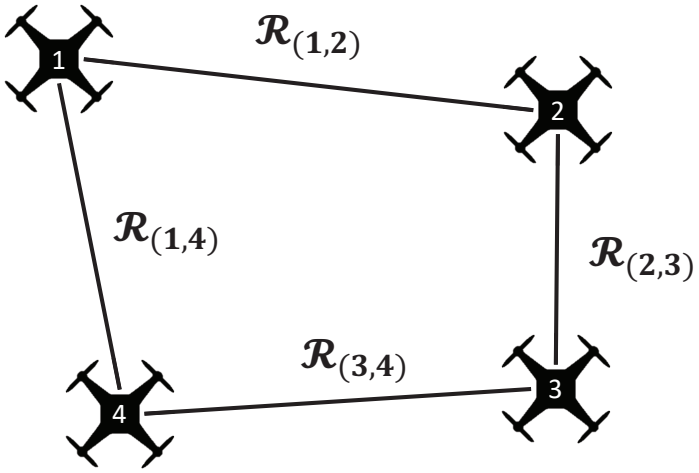


图 4.2　编队任务分配势博弈的参与者。

在证明编队任务分配问题 (4.5a) 是一个严格势博弈问题，且存在稳定解之后，依旧需要证明算法 4.5a 能够在有限次迭代内达到稳定解，即纳什均衡解，该问题由以下命题讨论。

命题 4.2： 算法 4.1 会在有限次迭代后终止，且终止时的任务分配规则为一个稳定的分配，即，其对应严格势博弈的纳什均衡策略。

证明： 对于有限个智能体，则任务分配的方案数为排列数，因为排列数是有限的，所以该严格势博弈的策略集合是有限的。在算法 4.1 的每次迭代中，智能体 i 和智能体 j 只有在博弈参与者的效用 $u_{(i,j)}$ 下降的时候才会交换节点。由公式 (4.11) 可知，博弈参与者效用 $u_{(i,j)}$ 的下降意味着势函数 Φ 的下降。再根据引理 4.2 可知，势函数 Φ 只能下降有限次，这意味着算法 4.1 会在有限次迭代后停止。同样，由于每次选择的都是造成编队性能指标下降最大的分配方案，即博弈参与者效用下降最大的策略，因此算法 4.1 会最终停止在稳定解，即纳什均衡策略。证明完毕。 □

重申本文对于任务分配算法的要求：既要尽可能地最优化编队性能指标 (3.7)，也要消解死锁。前述结论只能保证优化了编队性能指标，但并不能证明原有的死锁也被消解了，因此需要进一步地讨论。

命题 4.3： 如果编队网络发生了死锁，那么在经过节点交换后，死锁会被消解。换言之，编队网络发生了死锁，此时的编队性能指标为 ψ_f，则有以下等式关系：

$$\lambda_f \frac{\partial \psi_f}{\partial \boldsymbol{p}} + \lambda_r \frac{\partial \psi_r}{\partial \boldsymbol{p}} = \mathbf{0}. \tag{4.12}$$

24029602

则在发生节点交换后，编队的性能指标变为 $\psi_f' < \psi_f$，并且存在以下不等式：

$$\lambda_f \frac{\partial \psi_f'}{\partial \boldsymbol{p}} + \lambda_r \frac{\partial \psi_r}{\partial \boldsymbol{p}} \neq \mathbf{0}. \tag{4.13}$$

证明： 注意到任务分配规则发生改变后，只有方位拉普拉斯矩阵 $\mathcal{B}$ 发生了改变，记任务分配规则发生变化后的拉普拉斯矩阵为 $\mathcal{B}'$。由于任务分配规则发生改变后，整体编队性能指标发生了下降，因此有 $\psi_f' < \psi_f$，根据整体编队性能指标的定义 (3.7)，有以下不等式：

$$\boldsymbol{p}^T \mathcal{B}' \boldsymbol{p} < \boldsymbol{p}^T \mathcal{B} \boldsymbol{p} \quad \Longrightarrow \quad \boldsymbol{p}^T (\mathcal{B}' - \mathcal{B}) \boldsymbol{p} < 0. \tag{4.14}$$

由于 $\boldsymbol{p} \neq \mathbf{0}$ (否则所有智能体聚成一个点，该构型无意义)，因此有 $(\mathcal{B}' - \mathcal{B})\boldsymbol{p} \neq \mathbf{0}$，进而可以得到以下结论：

$$(\mathcal{B}' - \mathcal{B})\boldsymbol{p} \neq \mathbf{0}, \quad \frac{\partial \psi_f'}{\partial \boldsymbol{p}} = \mathcal{B}'\boldsymbol{p}, \quad \frac{\partial \psi_f}{\partial \boldsymbol{p}} = \mathcal{B}\boldsymbol{p} \Longrightarrow \frac{\partial \psi_f'}{\partial \boldsymbol{p}} \neq \frac{\partial \psi_f}{\partial \boldsymbol{p}}. \tag{4.15}$$

再由死锁的定义可得，

$$\lambda_f \frac{\partial \psi_f}{\partial \boldsymbol{p}} + \lambda_r \frac{\partial \psi_r}{\partial \boldsymbol{p}} = \mathbf{0}, \quad \frac{\partial \psi_f'}{\partial \boldsymbol{p}} \neq \frac{\partial \psi_f}{\partial \boldsymbol{p}} \Longrightarrow \lambda_f \frac{\partial \psi_f'}{\partial \boldsymbol{p}} + \lambda_r \frac{\partial \psi_r}{\partial \boldsymbol{p}} \neq \mathbf{0}. \tag{4.16}$$

证明完毕。 □

至此，已给出关于算法 4.1 所有要求性能的分析和证明。

4.4 仿真实验

在本小节中将对本章节提出的任务分配算法进行仿真验证，并与基于一致性的分布式拍卖算法[54]进行比较。同时，为了验证本章节提出任务分配算法对于编队导航成功率的提升，将给出任务分配后与轨迹优化的综合仿真结果。

4.4.1 任务分配仿真

考虑以下由 N 个节点组成的球形编队队形，所有节点均匀地分布在半径为 R 的球体上，第 i 个节点的坐标由以下公式给出，

$$\phi_i = \arccos(1 - \frac{2i}{N}), \quad \theta_i = \pi(1 + \sqrt{5})i$$

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} R\cos(\theta)\sin(\phi) \\ R\sin(\theta)\sin(\phi) \\ R\cos(\phi) \end{bmatrix}.$$

24029602

随后在 $[-R, R]$ 的范围内随机生成 N 个坐标，将这些坐标使用任务分配算法分配到前述 N 个编队节点上。其中，$N = 30$ 的编队队形如图 4.3 所示。任务分配的评价指标由编队偏移误差 e_{dev} 和算法执行时间 T 给出。

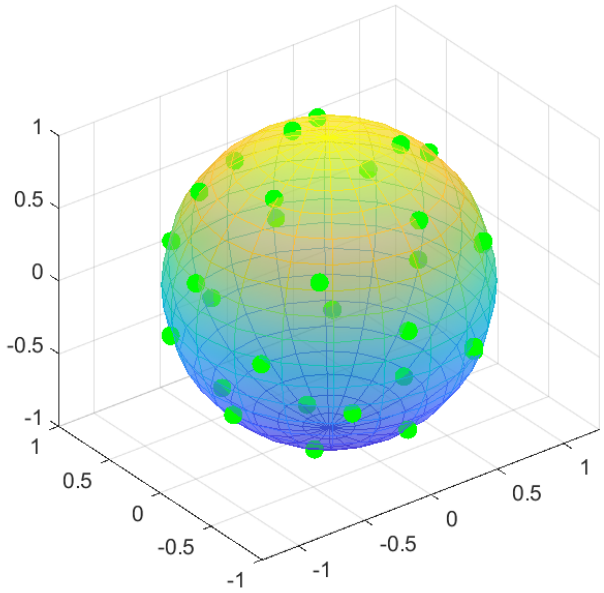


图 4.3　$N = 30, R = 1$ 的球形编队队形。

设置 $N = 30, R = 3$ 将一百次蒙特卡洛仿真的结果总结于表中，一方面，本文算法得到的编队偏移误差 e_{dev} 优于基于一致性的分布式拍卖算法，因为该算法只能达到理论最优值的 50 %[66]。本文算法的执行时间显著优于基于一致性的分布式拍卖算法，因此更适用于实时的编队导航。

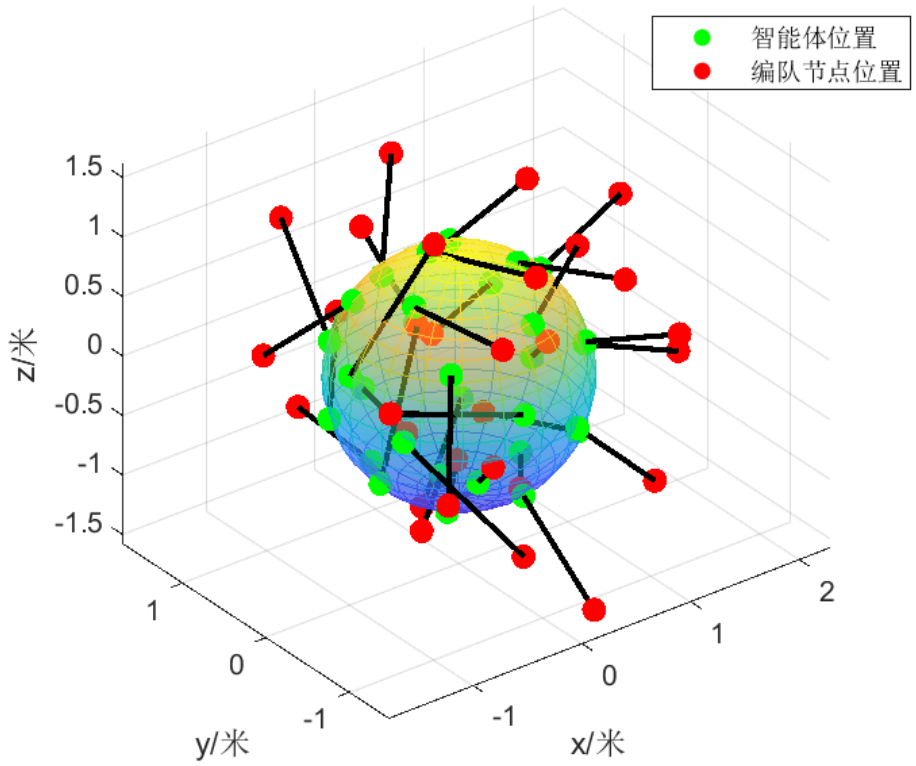
表 4.1　分布式任务分配算法性能对比

算法	分配后的平均编队偏移误差 e_{dev}	算法平均运行时间 T
基于一致性的分布式拍卖算法[66]	42.53 米	1.57 秒
本文提出的算法	**34.83** 米	**0.34** 秒

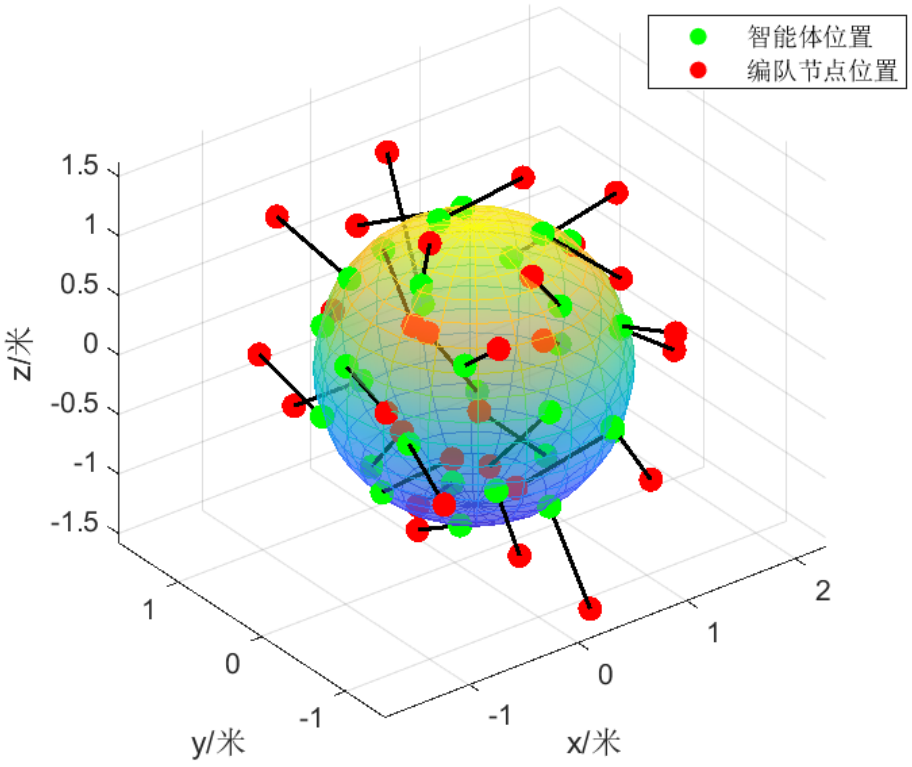
一百次蒙特卡洛仿真中的一个例子如图 4.4 所示。

4.4.2　任务分配与轨迹优化综合验证

本文提出的编队分配算法目的为提高编队导航的成功率，因此有必要结合上一章提出的轨迹优化算法进行仿真和实验验证，仿真将在编队保持和编队形成两方面展开，并在实物仿真平台上进一步验证算法有效性。



(a) 基于一致性的分布式拍卖算法[54]。



(b) 本文所提出的任务分配算法。

图 4.4　任务分配数值仿真：一百次蒙特卡洛仿真中的一个例子。

24029602

本文使用与上一章同样的仿真场景做一百次蒙特卡洛仿真实验，将加入任务分配后的结果与无任务分配前进行对比，对比结果总结于表 4.2 中。可以看出，在加入任务分配后，由于智能体会被动态的分配到不同的节点上来适应外界扰动，因此表现更好。

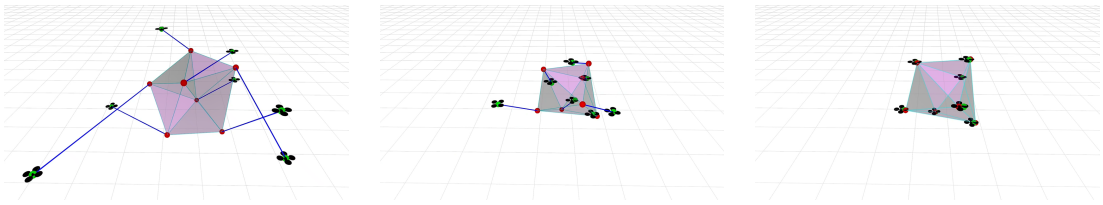
表 4.2　算法性能对比：编队队形保持

算法	通讯图	成功率	编队整体偏移误差 e_{dist}	编队相似性误差 e_{sim}
无任务分配	完全图	95%	0.2073 米	0.023
	拉曼图	94%	0.2345 米	0.028
有任务分配	完全图	98%	**0.1927** 米	**0.020**
	拉曼图	98%	0.2121 米	0.022

同样，使用与上一章相同的仿真环境对编队形成任务做一百次蒙特卡洛仿真实验，将加入任务分配后的结果与无任务分配前进行对比，对比结果总结于表 4.3 中。可见加入任务分配算法后，编队的死锁率有了显著的降低，因而提升了编队队形形成的成功率。另一方面，由于任务分配能够提升初始编队相较于期望编队队形的相似性，因此能够降低编队收敛到期望编队所花费的时间。

表 4.3　算法性能对比：编队队形形成

算法	成功率	变形率	死锁率	平均编队形成时间
无任务分配	88%	0%	12%	23.36 秒
有任务分配	**100**%	**0**%	**0**%	**9.73** 秒



(a) $t = 0s$，集群处于初始位置。　(b) $t = 5s$，集群向期望编队收敛。　(c) $t = 10s$，集群收敛到期望队形。

图 4.5　有任务分配时，本文所提出算法在编队形成上的结果。

一百次蒙特卡洛仿真中编队队形形成的一个例子如图 4.5 所示。可见，与无任务分配相比，有任务分配能使得无人机仅需 10 秒即可形成期望队形，原因是任务分配能使得集群快速找到与当前构型最相似的期望队形。

24029602

4.4.3 CrazyFlies 实物验证

为了验证本文所提出算法在现实场景中的可行性，本文搭建了一个无人机实验验证平台。该平台由一套动作捕捉系统、一台 Linux 电脑主机和 5 台无人机组成。在仿真的过程中，无人机的位置和障碍物的位置通过动作捕捉系统获取，随后这些信息将会被动作捕捉系统通过局域网传输到 Linux 电脑主机上进行处理，如图 4.6 所示。不同无人机的轨迹将会被 Linux 电脑主机上不同的进程计算出来，随后发送给无人机执行。

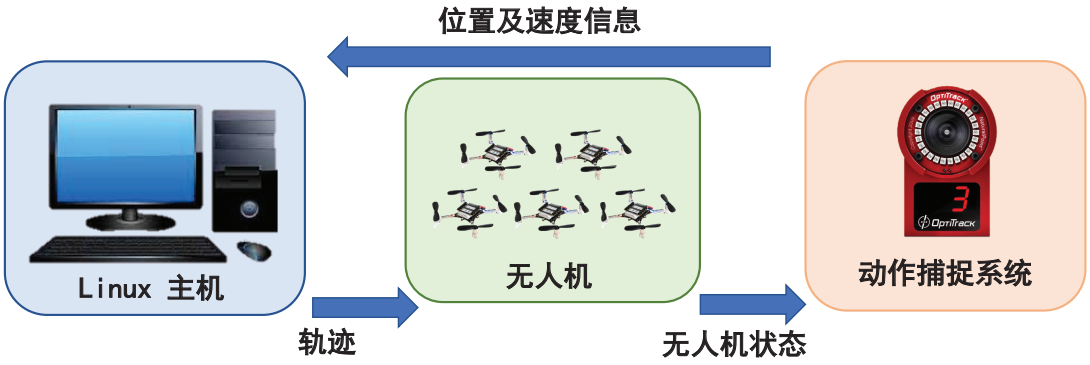


图 4.6　实验平台的各个部件及信息流向图。

在实验中，五架无人机被随机的放在 $x = 0$ 米之处。这些无人机将以 0.2 米每秒的最高速度限制飞行穿过障碍物区域，在飞行过程中无人机集群的期望队形为箭头队形，每个节点的标号如图 4.7 所示。测试场地中存在两个内径 0.9 米的圆环形障碍物和一个高为 2 米，宽 0.2 米的旗子障碍物，如图 4.8(b) 所示。

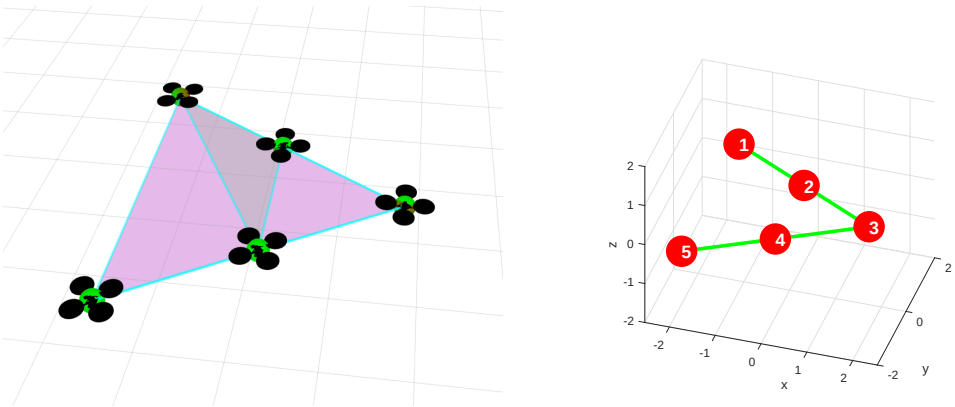
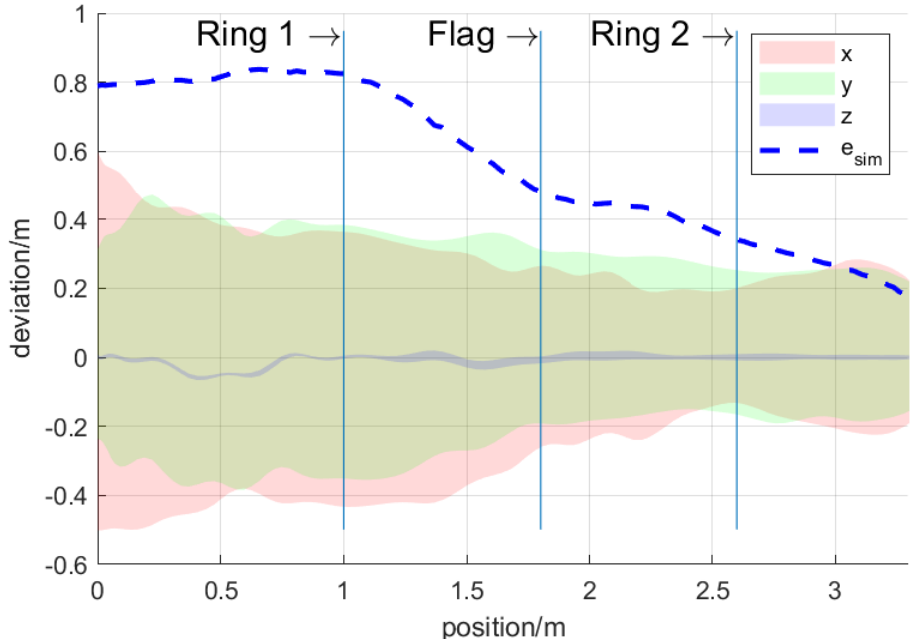
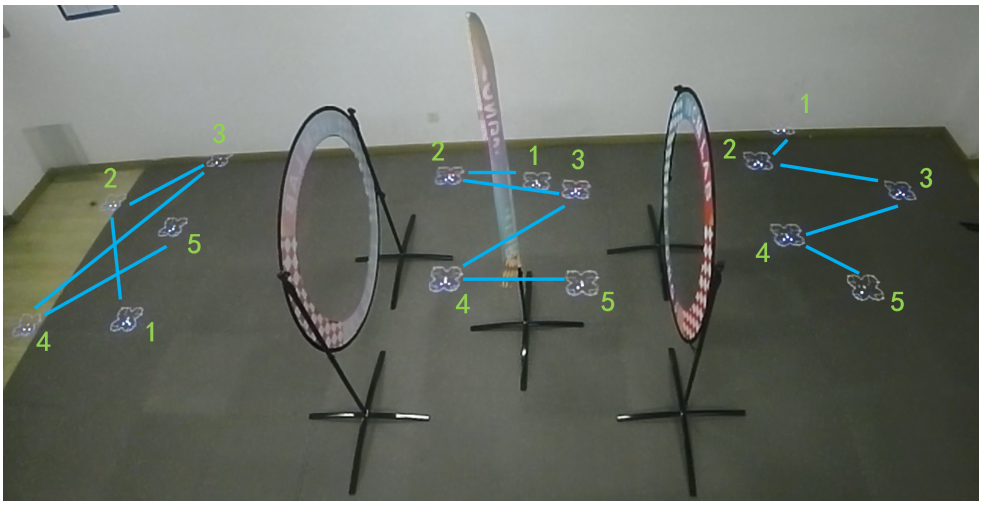


图 4.7　实物实验中使用的五架无人机组成的箭头编队队形。

24029602



(a) 实验中的编队偏移误差 e_{div} 在各个维度上的分量的上下界。

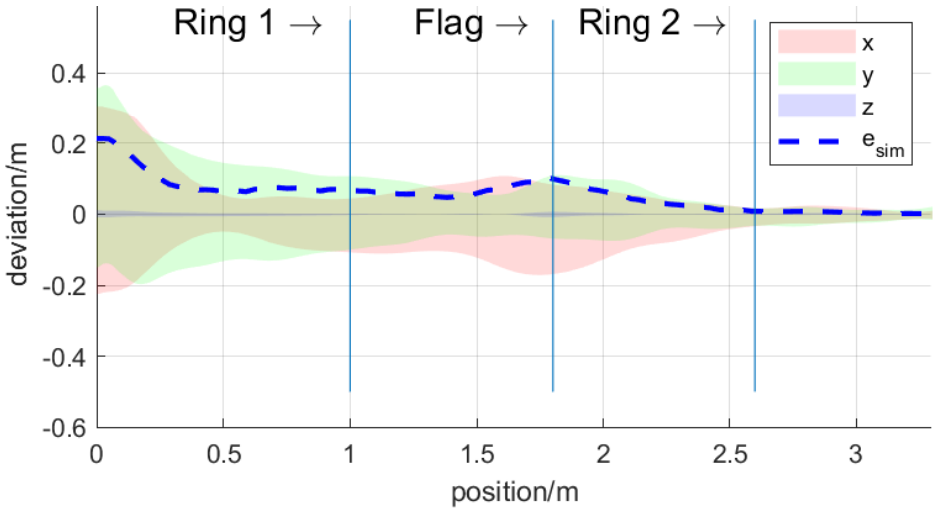


(b) 实验中无人机的快照。

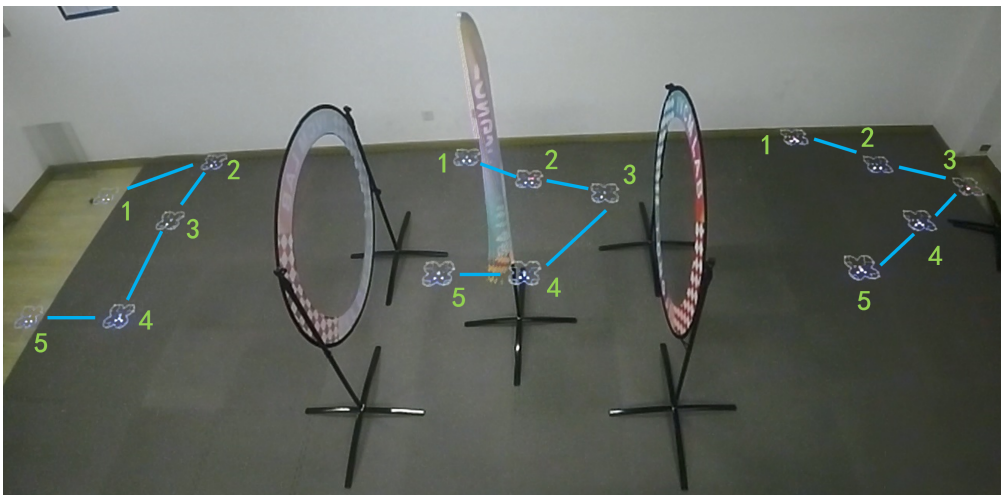
图 4.8　无任务分配时，本文所提出方法在实验中的编队导航误差图和快照图。

无任务分配时的实验结果如图 4.8 所示。由于没有任务分配，初始编队构型与期望编队构型相差巨大，因此难以在后续导航过程中快速形成队形，最终编队陷入了死锁中。在整个导航的过程中，虽然无人之机间未发生碰撞，也未与环境中的障碍物发生碰撞，但是编队偏移误差 e_{dev} 的平均值为 0.5249 米，编队相似性误差 e_{sim} 始终大于 0.2, 因而认定此次编队导航是失败的。

24029602



(a) 实验中的编队偏移误差 e_{div} 在各个维度上的分量的上下界。



(b) 实验中无人机的快照。

图 4.9　有任务分配时，本文所提出方法在实验中的编队导航误差图和快照图。

有任务分配时的实验结果如图 4.9 所示。得益于任务分配，在初始时无人机集群就能找到与期望队形最相似的任务分配方案，因此初始误差相较于无任务分配更小。在实验的过程中，无人机集群成功形成了箭头队形，并且通过在期望队形上轻微偏移来躲避障碍物。在整个飞行过程中，无人之机间未发生碰撞，也未与环境中的障碍物发生碰撞，编队偏移误差 e_{dev} 的平均值为 0.1164 米，编队相似性误差 e_{sim} 快速收敛到 0.2 以下, 因而认定此次编队导航是成功的。

4.5　小结

本章主要研究编队导航中的任务分配算法。本章首先分析了单纯使用多机器人编队轨迹优化可能导致机器人集群陷入死锁的原因，通过引入高层次协同，即引入编队任务分配的方法解决了死锁问题。

24029602

针对编队任务分配问题为二次分配问题而难以求解的挑战，本章将编队任务分配问题建模为精确势博弈问题，将编队任务分配问题放在博弈论的框架下求解。随后，本章节从理论上证明了算法能够在有限次迭代内收敛到纳什均衡策略，并证明了该算法能有效提升编队的初始相似度、消解死锁。

为了证明了所提出算法的优越性，本章将所提出算法与学术界中的基准算法 (基于一致性的拍卖算法) 进行比较，结果表明所提出算法在最优性和实时性上均优于基准算法。

最后为了验证算法对于编队导航的作用，本章将所提出算法与轨迹优化方法进行联合仿真，并搭建 Crazyflies 无人机实物平台进行实物验证，仿真和实验结果均证明了引入高层次协同能够有效提升编队导航的成功率。

24029602

第 5 章　总结与展望

5.1　本文工作总结

本文面向障碍物密集环境研究多机器人的编队导航算法，专注于单机器人全尺寸避障、多机器人分布式协同编队轨迹优化和编队任务分配三个主要研究内容。实现了高计算效率，高求解质量的编队导航算法。本文的工作总结如下：

在第 2 章中，本文深入研究了单机器人在密集障碍物环境中的实时避障算法，取得了一系列成果。利用凸多胞体的碰撞检测方法和机器人几何外形的各向异性特性，本文成功设计了光滑、解析和非保守的避障约束。通过将避障问题转化为优化问题，本文进一步引入了交替方向乘子法和序列二次规划方法，解决了高维非凸优化问题的实时求解难题。在验证过程中，本文以无人机为案例研究对象，详细介绍了算法的实际应用，并利用 AirSim 高保真仿真引擎验证了算法的有效性和实时性。这些成果为单机器人在复杂环境中的自主导航提供了有力支持。

在第 3 章中，本文研究了多机器人在障碍物密集环境中的编队轨迹优化算法。本文提出了基于方位刚性理论和方位约束的分布式编队性能指标，为机器人集群提供了有效的轨迹优化方法。同时，本文在综合考虑了编队导航中的各项需求 (如编队性能指标、机器人避障约束和相互避碰约束等) 后，提出了一种无约束的轨迹优化算法。该算法通过多项式轨迹作为优化变量，实现了轨迹的连续性和求解的实时性。利用精确罚函数法将约束条件转换为惩罚函数，成功将受约束问题转化为无约束优化问题。通过仿真实验，本文在障碍物密集环境中的编队保持、形成和导航等方面验证了算法的有效性和优越性。这些成果为多机器人协同编队导航提供了重要的理论支持和实践指导。

在第 4 章中，本文针对单纯使用多机器人编队轨迹优化可能导致机器人集群陷入死锁的问题，引入高层次协同的思想并设计了一种完全分布式的任务分配算法。通过将编队任务分配问题转换为精确势博弈问题，本文给出了寻找稳定任务分配方案，即势博弈纳什均衡解的方法。从理论上，本文证明了该算法能在有限次迭代内收敛到纳什均衡策略，并证明该算法能有效提升编队的初始相似度、消解死锁。通过基准比较和联合仿真验证了该算法的有效性，并通过使用 Crazyflies 无人机实物验证证明了引入高层次协同能够有效提升编队导航的成功率。这些成果为解决多机器人任务分配和编队导航中的关键问题提供了新的思路和方法。

24029602

5.2　未来研究方向展望

本文研究了障碍物密集环境中的编队导航，主要关注了单个机器人的全尺寸避障、编队协同轨迹优化和编队任务分配三个方面的问题。提出了一系列基于优化的避障、编队和任务分配算法。尽管这些算法在仿真和实物上得到了验证，这些算法依然有提升的空间：

对于单个机器人的全尺寸实时避障问题。首先，本文算法目前局限于障碍物和被控物体的几何外形均被建模为凸多胞体的情况。如何针对一般的几何外形，如椭球、范数球，甚至是非凸几何建立高效的碰撞检测算法和无碰撞约束仍然是一个值得深入研究的问题。其次，本文第 2 章提出的算法在使用 GPU 加速后的主要运算瓶颈在于机器人的状态变量和控制变量更新。如何运用诸如微分动态规划[95] (Differential Dynamic Programming) 等在线最优控制求解技术进一步提升运算效率亦是一个值得探索的方向。

对于多机器人的编队协同轨迹优化问题。首先，本文提出的算法需要所有机器人都共享一个共同的参考坐标系，并且不能使用旋转机动躲避障碍物。如何去掉共同参考坐标系，并且在不需要通讯拓扑为完全图的前提下实现编队旋转机动还需要进一步探究。一个可行的方案可能为引入角度刚性理论[96]来设计编队轨迹优化算法，作者将在未来的研究工作中进行尝试。其次，本文第 3 章提出算法的初始化需要用到 A 星算法和快速搜索随机树类算法。然而，这些算法可能为不同的机器人找到具有不同拓扑结构的初始路径，从而导致轨迹优化收敛到质量较差的结果。如何设计适合编队导航的前端搜索/采样算法，或者为编队导航算法设计拓扑优化算法仍然是一个悬而未决的问题。

对于编队任务分配问题。本文的算法目前仍然局限于消解集群的死锁，尚未考虑如何将外部环境信息与集群状态信息进行有机结合为集群分配更合理的编队目标点，这将是未来研究的方向。

最后，尽管本文给出了基于 Crazyflies 的实物验证实验，但该实验依赖于动作捕捉系统等实验室设备。如何仅利用机载运算资源、机载传感器实现真正具有自主智能的机器人编队导航系统将是推广多机器人技术落地应用的关键。

参考文献

[1] Han, J., Chen, Y. Multiple UAV formations for cooperative source seeking and contour mapping of a radiative signal field[J]. Journal of Intelligent & Robotic Systems, 2014, 74(1): 323-332.

[2] 原魁, 李园, 房立新. 多移动机器人系统研究发展近况[D]. 自动化学报, 2007.

[3] Acevedo, J. J., Arrue, B. C., Maza, I., et al. A decentralized algorithm for area surveillance missions using a team of aerial robots with different sensing capabilities[C]//2014 IEEE International Conference on Robotics and Automation (ICRA). 2014: 4735-4740.

[4] Balch, T., Arkin, R. C. Behavior-based formation control for multirobot teams[J]. IEEE Transactions on Robotics and Automation, 1998, 14(6): 926-939.

[5] Dunbabin, M., Marques, L. Robots for environmental monitoring: Significant advancements and applications[J]. IEEE Robotics & Automation Magazine, 2012, 19(1): 24-39.

[6] Tian, Y., Chang, Y., Arias, F. H., et al. Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems[J]. IEEE Transactions on Robotics, 2022, 38(4).

[7] Nourbakhsh, I. R., Sycara, K., Koes, M., et al. Human-robot teaming for search and rescue [J]. IEEE Pervasive Computing, 2005, 4(1): 72-79.

[8] 谭民, 王硕, 曹志强. 多机器人系统[M]. 清华大学出版社有限公司, 2005.

[9] 王祥科, 李迅, 郑志强. 多智能体系统编队控制相关问题研究综述[J]. 控制与决策, 2013, 28(11): 1601-1613.

[10] Dijkstra, E. W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269-271.

[11] Hart, P. E., Nilsson, N. J., Raphael, B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.

[12] Stentz, A. The focussed d^* algorithm for real-time replanning[C]//International Joint Conference on Artificial Intelligence: vol. 95. 1995: 1652-1659.

[13] Dolgov, D., Thrun, S., Montemerlo, M., et al. Path planning for autonomous vehicles in unknown semi-structured environments[J]. The International Journal of Robotics Research, 2010, 29(5): 485-501.

[14] Liu, S., Mohta, K., Atanasov, N., et al. Search-based motion planning for aggressive flight in SE(3)[J]. IEEE Robotics and Automation Letters, 2018, 3(3): 2439-2446.

[15] Kavraki, L., Svestka, P., Latombe, J.-C., et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces[J]. IEEE Transactions on Robotics and Automation, 1996, 12(4): 566-580.

[16] LaValle, S. M., Kuffner, J. J., Donald, B., et al. Rapidly-exploring random trees: Progress and prospects[J]. Algorithmic and Computational Robotics: New Directions, 2001, 5: 293-308.

[17] Kavraki, L. E., Kolountzakis, M. N., Latombe, J.-C. Analysis of probabilistic roadmaps for path planning[J]. IEEE Transactions on Robotics and Automation, 1998, 14(1): 166-171.

[18] Karaman, S., Frazzoli, E. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research, 2011, 30(7): 846-894.

[19] Wang, Z., Li, Y., Zhang, H., et al. Sampling-based optimal motion planning with smart exploration and exploitation[J]. IEEE/ASME Transactions on Mechatronics, 2020, 25(5): 2376-2386.

24029602

[20] Karaman, S., Frazzoli, E. Optimal kinodynamic motion planning using incremental sampling-based methods[C]//49th IEEE Conference on Decision and Control (CDC). 2010: 7681-7687.

[21] Webb, D. J., Van Den Berg, J. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics[C]//2013 IEEE International Conference on Robotics and Automation. 2013: 5054-5061.

[22] Mellinger, D., Kushleyev, A., Kumar, V. Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams[C]//2012 IEEE International Conference on Robotics and Automation. 2012: 477-483.

[23] Liu, S., Watterson, M., Mohta, K., et al. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments[J]. IEEE Robotics and Automation Letters, 2017, 2(3): 1688-1695.

[24] Zhou, X., Wen, X., Wang, Z., et al. Swarm of micro flying robots in the wild[J]. Science Robotics, 2022, 7(66): eabm5954.

[25] Richards, A., How, J. P. Aircraft trajectory planning with collision avoidance using mixed integer linear programming[C]//Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301): vol. 3. 2002: 1936-1941.

[26] Blackmore, L., Williams, B. Optimal manipulator path planning with obstacles using disjunctive programming[C]//2006 American Control Conference. 2006: 3-pp.

[27] Grossmann, I. E. Review of nonlinear mixed-integer and disjunctive programming techniques [J]. Optimization and Engineering, 2002, 3: 227-252.

[28] Rosolia, U., De Bruyne, S., Alleyne, A. G. Autonomous vehicle control: A nonconvex approach for obstacle avoidance[J]. IEEE Transactions on Control Systems Technology, 2016, 25(2): 469-484.

[29] Nägeli, T., Alonso-Mora, J., Domahidi, A., et al. Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization[J]. IEEE Robotics and Automation Letters, 2017, 2(3): 1696-1703.

[30] Schulman, J., Duan, Y., Ho, J., et al. Motion planning with sequential convex optimization and convex collision checking[J]. The International Journal of Robotics Research, 2014, 33(9): 1251-1270.

[31] Li, B., Shao, Z. A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles[J]. Knowledge-Based Systems, 2015, 86: 11-20.

[32] Zhang, X., Liniger, A., Borrelli, F. Optimization-based collision avoidance[J]. IEEE Transactions on Control Systems Technology, 2020, 29(3): 972-983.

[33] Wang, Q., Wang, Z., Pei, L., et al. A linear and exact algorithm for whole-body collision evaluation via scale optimization[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). 2023: 3621-3627.

[34] Han, R., Wang, S., Wang, S., et al. RDA: An accelerated collision free motion planner for autonomous navigation in cluttered environments[J]. IEEE Robotics and Automation Letters, 2023, 8(3): 1715-1722.

[35] Dunbar, W. B., Murray, R. M. Distributed receding horizon control for multi-vehicle formation stabilization[J]. Automatica, 2006, 42(4): 549-558.

[36] Zhao, S., Zelazo, D. Bearing rigidity and almost global bearing-only formation stabilization [J]. IEEE Transactions on Automatic Control, 2015, 61(5): 1255-1268.

[37] Sun, Z., Park, M.-C., Anderson, B. D., et al. Distributed stabilization control of rigid formations with prescribed orientation[J]. Automatica, 2017, 78: 250-257.

[38] 王祥科, 刘志宏, 丛一睿, 等. 小型固定翼无人机集群综述和未来发展[J]. 航空学报, 2020, 41(4): 15-40.

[39] Zhao, S., Zelazo, D. Translational and scaling formation maneuver control via a bearing-based approach[J]. IEEE Transactions on Control of Network Systems, 2015, 4(3): 429-438.

[40] Zhao, S. Affine formation maneuver control of multiagent systems[J]. IEEE Transactions on Automatic Control, 2018, 63(12): 4140-4155.

[41] De Marina, H. G. Distributed formation maneuver control by manipulating the complex Laplacian[J]. Automatica, 2021, 132: 109813.

[42] Dong, X., Yu, B., Shi, Z., et al. Time-varying formation control for unmanned aerial vehicles: Theories and applications[J]. IEEE Transactions on Control Systems Technology, 2014, 23(1): 340-348.

[43] Ge, S. S., Liu, X., Goh, C.-H., et al. Formation tracking control of multiagents in constrained space[J]. IEEE Transactions on Control Systems Technology, 2015, 24(3): 992-1003.

[44] Zhao, S., Zelazo, D. Bearing rigidity theory and its applications for control and estimation of network systems: Life beyond distance rigidity[J]. IEEE Control Systems Magazine, 2019, 39(2): 66-83.

[45] Lu, Y., Dong, X., Li, Q., et al. Time-varying group formation-containment tracking control for general linear multiagent systems with unknown inputs[J]. IEEE Transactions on Cybernetics, 2021, 52(10): 11055-11067.

[46] Zhao, S., Li, Z., Ding, Z. Bearing-only formation tracking control of multiagent systems[J]. IEEE Transactions on Automatic Control, 2019, 64(11): 4541-4554.

[47] Alonso-Mora, J., Baker, S., Rus, D. Multi-robot formation control and object transport in dynamic environments via constrained optimization[J]. The International Journal of Robotics Research, 2017, 36(9): 1000-1021.

[48] Quan, L., Yin, L., Xu, C., et al. Distributed swarm trajectory optimization for formation flight in dense environments[C]//2022 International Conference on Robotics and Automation (ICRA). 2022: 4979-4985.

[49] De Marina, H. G., Jayawardhana, B., Cao, M. Distributed rotational and translational maneuvering of rigid formations and their applications[J]. IEEE Transactions on Robotics, 2016, 32(3): 684-697.

[50] Van Parys, R., Pipeleers, G. Distributed model predictive formation control with inter-vehicle collision avoidance[C]//. 2017: 2399-2404.

[51] Dai, L., Cao, Q., Xia, Y., et al. Distributed MPC for formation of multi-agent systems with collision avoidance and obstacle avoidance[J]. 2017, 354(4): 2068-2085.

[52] Zhou, D., Wang, Z., Schwager, M. Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures[J]. IEEE Transactions on Robotics, 2018, 34(4): 916-923.

[53] Hu, J., Zhang, H., Liu, L., et al. Convergent multiagent formation control with collision avoidance[J]. IEEE Transactions on Robotics, 2020, 36(6): 1805-1818.

[54] Lusk, P. C., Cai, X., Wadhwania, S., et al. A distributed pipeline for scalable, deconflicted formation flying[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 5213-5220.

[55] Kuhn, H. W. The Hungarian method for the assignment problem[J]. Naval Research Logistics Quarterly, 1955, 2(1-2): 83-97.

[56] Giordani, S., Lujak, M., Martinelli, F. A distributed algorithm for the multi-robot task allocation problem[C]//Trends in Applied Intelligent Systems: 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, Cordoba, Spain, June 1-4, 2010, Proceedings, Part I 23. 2010: 721-730.

[57] Chopra, S., Notarstefano, G., Rice, M., et al. A distributed version of the hungarian method for multirobot assignment[J]. IEEE Transactions on Robotics, 2017, 33(4): 932-947.

24029602

[58] Berhault, M., Huang, H., Keskinocak, P., et al. Robot exploration with combinatorial auctions[C]//Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453): vol. 2. 2003: 1957-1962.

[59] Bertsekas, D. P. The auction algorithm: A distributed relaxation method for the assignment problem[J]. Annals of operations research, 1988, 14(1): 105-123.

[60] Zlot, R., Stentz, A. Market-based multirobot coordination for complex tasks[J]. The International Journal of Robotics Research, 2006, 25(1): 73-101.

[61] Nunes, E., Gini, M. Multi-robot auctions for allocation of tasks with temporal constraints[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 29: 1. 2015.

[62] Sariel, S., Balch, T. Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments[C]//Proceedings of the AAAI-05 Workshop on Integrating Planning into Scheduling. 2005: 27-33.

[63] Lemaire, T., Alami, R., Lacroix, S. A distributed tasks allocation scheme in multi-UAV context[C]//IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004: vol. 4. 2004: 3622-3627.

[64] Sujit, P., Beard, R. Distributed sequential auctions for multiple UAV task allocation[C]//2007 American Control Conference. 2007: 3955-3960.

[65] Smith, S. L., Bullo, F. Monotonic target assignment for robotic networks[J]. IEEE Transactions on Automatic Control, 2009, 54(9): 2042-2057.

[66] Choi, H.-L., Brunet, L., How, J. P. Consensus-based decentralized auctions for robust task allocation[J]. IEEE Transactions on Robotics, 2009, 25(4): 912-926.

[67] Michael, N., Zavlanos, M. M., Kumar, V., et al. Distributed multi-robot task assignment and formation control[C]//2008 IEEE International Conference on Robotics and Automation. 2008: 128-133.

[68] Luo, L., Chakraborty, N., Sycara, K. Distributed algorithm design for multi-robot task assignment with deadlines for tasks[C]//2013 IEEE International Conference on Robotics and Automation. 2013: 3007-3013.

[69] Otte, M., Kuhlman, M. J., Sofge, D. Auctions for multi-robot task allocation in communication limited environments[J]. Autonomous Robots, 2020, 44: 547-584.

[70] Schneider, E., Sklar, E. I., Parsons, S., et al. Auction-based task allocation for multi-robot teams in dynamic environments[C]//Towards Autonomous Robotic Systems: 16th Annual Conference, TAROS 2015, Liverpool, UK, September 8-10, 2015, Proceedings 16. 2015: 246-257.

[71] Sui, Z., Pu, Z., Yi, J. Optimal uavs formation transformation strategy based on task assignment and particle swarm optimization[C]//2017 IEEE International Conference on Mechatronics and Automation (ICMA). 2017: 1804-1809.

[72] Choi, Y. H., Kim, D. Distance-Based Formation Control With Goal Assignment for Global Asymptotic Stability of Multi-Robot Systems[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 2020-2027.

[73] Wang, H., Rubenstein, M. Shape formation in homogeneous swarms using local task swapping[J]. IEEE Transactions on Robotics, 2020, 36(3): 597-612.

[74] Wang, Z., Li, G., Jiang, H., et al. Collision-free navigation of autonomous vehicles using convex quadratic programming-based model predictive control[J]. IEEE/ASME Transactions on Mechatronics, 2018, 23(3): 1103-1113.

[75] Ding, W., Zhang, L., Chen, J., et al. Epsilon: An efficient planning system for automated vehicles in highly interactive environments[J]. IEEE Transactions on Robotics, 2021, 38(2): 1118-1138.

[76] Loquercio, A., Kaufmann, E., Ranftl, R., et al. Learning high-speed flight in the wild[J]. Science Robotics, 2021, 6(59): eabg5810.

[77] Wang, Z., Zhou, X., Xu, C., et al. Geometrically constrained trajectory optimization for multicopters[J]. IEEE Transactions on Robotics, 2022, 38(5): 3259-3278.

[78] Tracy, K., Howell, T. A., Manchester, Z. Differentiable collision detection for a set of convex primitives[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). 2023: 3663-3670.

[79] Rockafellar, R. T. Convex analysis: vol. 11[M]. Princeton University Press, 1997.

[80] Boyd, S. P., Vandenberghe, L. Convex optimization[M]. Cambridge University Press, 2004.

[81] Boyd, S., Parikh, N., Chu, E., et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends® in Machine learning, 2011, 3(1): 1-122.

[82] Kipfer, P. LCP algorithms for collision detection using CUDA[J]. GPU Gems, 2007, 3: 723-740.

[83] Cottle, R. W., Pang, J.-S., Stone, R. E. The linear complementarity problem[M]. SIAM, 2009.

[84] Shah, S., Dey, D., Lovett, C., et al. Airsim: High-fidelity visual and physical simulation for autonomous vehicles[C]//Field and Service Robotics: Results of the 11th International Conference. 2018: 621-635.

[85] Wen, G., Lam, J., Fu, J., et al. Distributed MPC-Based Robust Collision Avoidance Formation Navigation of Constrained Multiple USVs[J]. IEEE Transactions on Intelligent Vehicles, 2023.

[86] Wu, W., Tong, S. Collision-Free Finite-Time Adaptive Fuzzy Output-Feedback Formation Control for Unmanned Surface Vehicle Systems[J]. IEEE Transactions on Intelligent Vehicles, 2023: 1-9.

[87] Zhao, S., Zelazo, D. Bearing rigidity and almost global bearing-only formation stabilization [J]. IEEE Transactions on Automatic Control, 2015, 61(5): 1255-1268.

[88] Press, W. H., Teukolsky, S. A., Vetterling, W. T., et al. Numerical recipes 3rd edition: The art of scientific computing[M]. Cambridge University Press, 2007.

[89] Oleynikova, H., Taylor, Z., Fehr, M., et al. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017: 1366-1373.

[90] Liu, D. C., Nocedal, J. On the limited memory BFGS method for large scale optimization[J]. Math. Program., 1989, 45(1-3): 503-528.

[91] Horn, B. K. Closed-form solution of absolute orientation using unit quaternions[J]. Josa a, 1987, 4(4): 629-642.

[92] Zhao, S., Sun, Z., Zelazo, D., et al. Laman graphs are generically bearing rigid in arbitrary dimensions[C]//2017 IEEE 56th Annual Conference on Decision and Control (CDC). 2017: 3356-3361.

[93] Wang, L., Ames, A. D., Egerstedt, M. Safety Barrier Certificates for Collisions-Free Multirobot Systems[J]. IEEE Transactions on Robotics, 2017, 33(3): 661-674.

[94] Loiola, E. M., De Abreu, N. M. M., Boaventura-Netto, P. O., et al. A survey for the quadratic assignment problem[J]. European Journal of Operational Research, 2007, 176(2): 657-690.

[95] Xie, Z., Liu, C. K., Hauser, K. Differential dynamic programming with nonlinear constraints [C]//2017 IEEE International Conference on Robotics and Automation (ICRA). 2017: 695-702.

[96] Chen, L., Cao, M., Li, C. Angle rigidity and its usage to stabilize multiagent formations in 2-D[J]. IEEE Transactions on Automatic Control, 2021, 66(8): 3667-3681.

24029602

个人简历、在学期间发表的学术论文与研究成果

个人简历:

吴泽铭，男，1999 年 4 月生。

2021 年 7 月毕业于同济大学自动化专业，获学士学位。

2021 年 9 月入同济大学攻读硕士研究生。

录用论文:

[1] **Wu, Z.**, Wang, Z., Zhang, H. GPU-Accelerated Optimization-Based Collision Avoidance[C] 2024 IEEE International Conference on Robotics and Automation (ICRA). **Accepted**. 2024.

投稿论文:

[1] **Wu, Z.**, Wang, Z., Zhang, H., Yan, H. Quaternion-based Optimal Interpolation of Similarity Transformations for Multi-agent Formation. IEEE Robotics And Automation Letters. **Revise and Resubmit**.

[2] Lv, Y., **Wu, Z.**, Zhang H., Wang Z., Yan H. Local-bearing-based prescribed-time distributed localization of multi-agent systems with noisy measurement. IEEE Transactions on Industrial Informatics. **Major Revision**. (共同一作)

[3] **Wu, Z.**, Wang, Z., Zhang, H., Yan, H. Formation navigation in cluttered environments: A distributed optimization method. Transactions on Intelligent Vehicles. **Under Review**.

竞赛获奖:

[1] 黄定梁, **吴泽铭**, 巴志彪. 2021 第 4 届全国大学生嵌入式芯片与系统设计竞赛-全国赛区-国家级赛. 二等奖. 2021-10-30.

24029602

同济大学学位论文原创性声明

本人郑重声明：所呈交的学位论文《障碍物密集环境中的编队导航》，是本人在导师指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名： 吴泽铭

日期： 2024 年 3 月 5 日

同济大学学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；允许论文被查阅和借阅。学校有权将本学位论文的全部或部分内容授权编入有关数据库出版传播，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于（在以下方框内打“√”）：

☐ **保密，在_____年解密后适用本授权书。**

☑ **不保密。**

学位论文作者签名： 吴泽铭　　**指导教师签名：** 张[illegible]

日期： 2024 年 3 月 5 日　　**日期：** 2024 年 3 月 5 日

24029602

七、学位论文答辩委员会决议

姓名	吴泽铭	学号	2130729	所在学科/专业	控制科学与工程
指导教师	王祝萍	答辩日期	2024-03-08	答辩地点	济人楼 316
论文题目	障碍物密集环境中的编队导航				

答辩委员会共 4 人，经表决，4 人建议授予申请人硕士学位。根据《同济大学学位授予工作细则》[注]（在☐内划"√"）：

☑ 建议授予申请人硕士学位。

☐ 申请人可在一年内修改论文，申请重新答辩一次。

☐ 建议不授予申请人硕士学位。

是否推荐为同济大学优秀硕士学位论文：☑ 是 ☐ 否

吴泽铭同学撰写的学位论文《障碍物密集环境中的编队导航》探讨了障碍物密集环境中的多机器人编队导航问题，并针对四旋翼无人机设计了一套创新的编队导航框架。论文选题新颖，符合国家对自主智能系统在灾后救援、协同侦察和环境检测等方向的发展需求，具有现实意义及应用价值。论文的研究取得了下列成果：

1）研究了单机器人在障碍物密集环境中的实时避障算法。针对全尺寸避障问题高维度、非线性而难以实时求解的挑战，提出了适用于嵌入式设备的实时优化算法，通过无人机验证了算法的实际应用和高实时性。

2）研究了多机器人在障碍物密集环境中的编队轨迹优化算法。算法利用方位约束为编队避障提供更多自由度，实现障碍物密集场景中机器人集群的灵活避障。

3）研究了编队导航中的任务分配算法。该算法设计了一种完全分布式的协调方法，为集群引入高层次协同，实现了编队初始的相似度的提升并消解死锁。

总体而言，论文结构清晰，层次分明，阐述逻辑严密，语言流畅，完全符合学位论文的撰写规范。论文中的实验结果可靠，数据分析合理，所得结论具有很强的说服力，论文达到了硕士学位的学术水平。答辩委员会经无记名投票，一致同意吴泽铭同学通过论文答辩，并建议授予 工学硕士学位。

答辩委员会主席签名：张皓 张皓 2024年 3月 8日

答辩委员会成员签名	职务	姓名	职称	单位	签名
	主席	张皓	教授	同济大学	张皓
	委员	王祝萍	教授	同济大学	王祝萍
	委员	尹慧琳	副教授	同济大学	尹慧琳
	委员	张长柱	副教授	同济大学	张长柱
	委员				
秘书		王云姣	讲师	同济大学	王云姣

注：根据《同济大学学位授予工作细则》第十一条规定：1. 申请人获得全体答辩委员会成员三分之二以上（含）同意票，为建议授予申请人硕士学位；2. 申请人获得全体答辩委员会成员二分之一以上（含）、三分之二以下（不含）同意票，申请人可在一年内修改论文，申请重新答辩一次；3. 申请人获得全体答辩委员会成员二分之一以下（不含）同意票，为建议不授予申请人硕士学位。