

# 基于可行域的多无人车编队变换方法

## 摘要

近年来，随着智慧交通和物流运输领域对多无人车编队需求的不断增长，针对复杂环境下编队算法的适应性与性能也提出了更高要求。在障碍物稠密环境中，传统的多无人车编队算法常面临运行效率低，编队效果不佳等瓶颈。为应对这一挑战，本文提出了队形选择器与编队控制器，能够实时调整无人车队形以适应环境变化，实现复杂环境下的高效编队导航。主要工作包括：

针对传统求解可行区域方法较为保守且不利于编队高效通行的问题，本文提出了基于可行区域的队形选择算法。首先，编队级路径规划算法以圆形区域代替二维位置作为路径点，并在其评分函数中融入编队信息，从而确保路径与障碍物保持安全距离，减少因避障引发的频繁队形调整。随后，基于路径与障碍物信息构建二次规划问题，求解垂直于路径的可通行宽度，作为队形选择的依据。最后，在仿真环境中，进行了不同障碍物数量场景的对比实验和权重分析实验，验证了所提算法的平均安全距离与当前形状更匹配。

针对现有算法在编队变换中收敛效率低的问题，本文提出了基于编队一致性的队形控制算法。首先，基于最优运输的任务分配，建立多无人车与编队形状的对应关系，以减少队形变换所需的总路程。其次，设计了领导无人车重选择策略，集中式地执行队形选择和任务分配方法，提升编队信息的一致性效率。接着，根据编队位置一致性理论和路径信息，设计控制器的参考状态，并构建基于模型预测控制的有约束优化问题。最后，在仿真实验中，设计了队形变换对比实验和编队可扩展性实验，实验结果验证了所提编队控制算法在编队收敛效率方面的优势。

基于上述队形选择算法和队形控制算法，分别在仿真环境和真实场景中搭建实验平台，开展编队导航综合实验验证。首先，在受限环境下进行了编队导航对比仿真实验，实验结果表明本文的方法能够快速减少编队误差并控制多无人车收敛到期望队形。其次，开展了不同规模编队库的编队导航仿真实验，实验结果表明无论编队库规模如何，本文的方法均能从编队库中选取符合当前障碍物环境的编队形状。最后，进行了多辆 Turtlebot4 实机实验，进一步验证了本文所提出算法在实际应用上的可行性和有效性。

**关键词：**多无人车编队，路径规划，非线性规划，模型预测控制

# FEASIBLE REGION-BASED MULTI-UNMANNED VEHICLE FORMATION TRANSFORMATION METHOD

## ABSTRACTS

In recent years, with the growing demand for multi-unmanned vehicle formations in the fields of intelligent transportation and logistics, higher requirements have been placed on the adaptability and performance of formation algorithms in complex environments. In obstacle-dense environments, traditional multi-unmanned vehicle formation algorithms often face bottlenecks such as low operational efficiency and poor formation performance. To address this challenge, this thesis proposes a formation selector and a formation controller capable of dynamically adjusting the formation of unmanned vehicles in real time to adapt to environmental changes, thereby enabling efficient formation navigation in complex environments. The main contributions include:

To address the issue that traditional feasible region-solving methods are overly conservative and hinder efficient formation navigation, this thesis proposes a feasible region-based formation selection algorithm. First, the formation-level path planning algorithm uses circular regions instead of two-dimensional positions as waypoints and incorporates formation information into its scoring function. This ensures that the path maintains a safe distance from obstacles, reducing frequent formation adjustments caused by obstacle avoidance. Next, based on path and obstacle information, a quadratic programming problem is constructed to solve for the passable width perpendicular to the path, which serves as the basis for formation selection. Finally, comparative experiments with varying numbers of obstacles and weight analysis experiments were conducted in a simulation environment, demonstrating that the proposed algorithm achieves a better match between the average safety distance and the current formation shape.

To address the low convergence efficiency of existing algorithms during formation transitions, this thesis proposes a formation control algorithm based on formation

consistency. First, task allocation based on optimal transport establishes correspondences between multiple unmanned vehicles and formation shapes, thereby reducing the total travel distance required for formation transformations. Second, a leader robot reselection strategy is designed to centrally perform formation selection and task assignment, thereby improving the efficiency of formation information consistency. Then, based on formation position consistency theory and path information, a reference state for the controller is designed, and a constrained optimization problem is formulated using model predictive control. Finally, formation transition comparison experiments and formation scalability tests are conducted in simulation. The results verify the superiority of the proposed formation control algorithm in terms of convergence efficiency.

Based on the proposed formation selection and control algorithms, experimental platforms were established in both simulation and real-world environments to conduct integrated formation navigation experiments for validation. First, comparative simulation experiments on formation navigation in constrained environments were performed. The results demonstrate that the proposed method can rapidly reduce formation errors and guide multiple robots to converge to the desired formation. Second, simulation experiments with formation libraries of varying scales were conducted, showing that regardless of the library size, the proposed method can select a formation shape suitable for the current obstacle environment. Finally, real-world experiments with multiple Turtlebot4 robots were carried out, further verifying the feasibility and effectiveness of the proposed multi-unmanned vehicle formation algorithm in practical applications.

**KEY WORDS:** multi-unmanned vehicle Formation, path planning, nonlinear programming, model predictive control

# 目 录

|                            |      |
|----------------------------|------|
| 摘要.....                    | I    |
| ABSTRACTS .....            | II   |
| 目录.....                    | IV   |
| 插图清单.....                  | VI   |
| 表格清单.....                  | VIII |
| 第一章 绪 论.....               | 1    |
| 1.1 研究背景及意义.....           | 1    |
| 1.2 研究现状.....              | 2    |
| 1.2.1 多机器人路径规划.....        | 2    |
| 1.2.2 可行区域计算.....          | 3    |
| 1.2.3 编队控制.....            | 4    |
| 1.3 论文研究内容和章节安排.....       | 6    |
| 第二章 相关技术与基本框架.....         | 9    |
| 2.1 引言.....                | 9    |
| 2.2 相关技术.....              | 9    |
| 2.2.1 无人车的运动学模型.....       | 9    |
| 2.2.2 基础图论.....            | 9    |
| 2.2.3 RRT*算法.....          | 10   |
| 2.3 框架介绍.....              | 11   |
| 2.4 本章小结.....              | 14   |
| 第三章 基于可行区域的队形选择.....       | 15   |
| 3.1 引言.....                | 15   |
| 3.2 基于 RRT*改进的编队级路径规划..... | 15   |
| 3.3 最大可通行宽度检测.....         | 23   |
| 3.4 编队形状选择.....            | 24   |
| 3.5 仿真实验分析.....            | 27   |
| 3.5.1 实验设置.....            | 27   |
| 3.5.2 性能对比.....            | 27   |
| 3.5.3 关键参数分析.....          | 30   |
| 3.5.4 消融实验.....            | 31   |
| 3.6 本章小结.....              | 32   |

|                                   |           |
|-----------------------------------|-----------|
| <b>第四章 基于编队一致性的队形控制.....</b>      | <b>33</b> |
| 4.1 引言.....                       | 33        |
| 4.2 基于最优运输的任务分配.....              | 33        |
| 4.3 领导无人车重选择策略.....               | 34        |
| 4.4 基于编队一致性的控制器.....              | 35        |
| 4.5 基于路径的局部避障策略.....              | 37        |
| 4.6 仿真实验分析.....                   | 38        |
| 4.6.1 编队收敛效率对比分析.....             | 39        |
| 4.6.2 编队可扩展性分析.....               | 43        |
| 4.7 本章小结.....                     | 44        |
| <b>第五章 多无人车编队综合实验平台搭建与验证.....</b> | <b>45</b> |
| 5.1 引言.....                       | 45        |
| 5.2 实验准备.....                     | 45        |
| 5.3 多无人车编队仿真实验.....               | 46        |
| 5.3.1 实验平台搭建.....                 | 46        |
| 5.3.2 受限环境下编队导航实验对比分析.....        | 47        |
| 5.3.3 编队库中不同编队模式数量的实验分析.....      | 49        |
| 5.4 多无人车编队实际场景验证.....             | 50        |
| 5.4.1 实验平台搭建.....                 | 50        |
| 5.4.2 无障碍物环境中的编队收敛实验.....         | 50        |
| 5.4.3 有障碍物环境中的编队导航实验.....         | 52        |
| 5.5 本章小结.....                     | 55        |
| <b>第六章 总结与展望.....</b>             | <b>56</b> |
| 6.1 工作总结.....                     | 56        |
| 6.2 工作展望.....                     | 57        |
| <b>参考文献.....</b>                  | <b>58</b> |

## 插图清单

|   |    |
|---|----|
| 图 1-1 编队的常见应用 .....                             | 1  |
| 图 1-2 论文章节安排框图 .....                            | 7  |
| 图 2-1 RRT*算法流程图.....                            | 10 |
| 图 2-2 可通行宽度示意图 .....                            | 11 |
| 图 2-3 四辆无人车的预设编队库 .....                         | 11 |
| 图 2-4 自适应编队变换方法框架图 .....                        | 14 |
| 图 3-1 均匀采样节点的示意图 .....                          | 17 |
| 图 3-2 多叉树上距离随机节点最近节点的示意图 .....                  | 18 |
| 图 3-3 搜索最新节点示意图 .....                           | 19 |
| 图 3-4 查找与新节点相交的节点集合的示意图 .....                   | 20 |
| 图 3-5 更新父节点的示意图 .....                           | 21 |
| 图 3-6 椭圆采样区域的示意图 .....                          | 21 |
| 图 3-7 计算无障碍物范围的示意图 .....                        | 22 |
| 图 3-8 一条路径预测不同编队形状的例子 .....                     | 22 |
| 图 3-9 宽度检测示意图 .....                             | 23 |
| 图 3-10 在 ESDF 地图上利用公式(3-8)调整局部目标点的示意图 .....     | 26 |
| 图 3-11 不同障碍物规模的测试环境 .....                       | 27 |
| 图 3-12 在不同障碍物数量的环境中对比不同算法的路径长度 .....            | 28 |
| 图 3-13 在不同障碍物数量的环境中对比不同算法的最小安全距离 .....          | 28 |
| 图 3-14 在不同障碍物数量的环境中对比不同算法的平均安全距离 .....          | 29 |
| 图 3-15 不同路径规划算法在有障碍物环境中的求解结果示意图 .....           | 29 |
| 图 3-16 不同编队权重因子对平均安全距离的影响 .....                 | 30 |
| 图 3-17 不同编队权重因子对路径长度的影响 .....                   | 31 |
| 图 3-18 以 RRT*+WidthCheck 的方式计算可行域宽度的示意图 .....   | 32 |
| 图 3-19 以 FrmRRT*+WidthCheck 的方式计算可行域宽度的示意图..... | 32 |
| 图 4-1 基于最优运输的任务分配和领导无人车重选择策略示意图 .....           | 34 |
| 图 4-2 无人车的模型图 .....                             | 35 |
| 图 4-3 不同方法在收敛到期望队形过程中的相似度误差变化趋势 .....           | 42 |
| 图 4-4 不同方法收敛到期望队形的结果 .....                      | 43 |
| 图 4-5 不同数量的无人车的编队控制器计算时间的箱线图 .....              | 43 |
| 图 4-6 不同时刻的编队收敛实验中所有无人车的位置 .....                | 44 |

|   |    |
|---|----|
| 图 5-1 锥型走廊和大量障碍物的场景 .....                 | 46 |
| 图 5-2 无人车集群实验平台 .....                     | 47 |
| 图 5-3 无人车在穿越图 5-1(b)的复杂环境时不同时刻的示意图 .....  | 48 |
| 图 5-4 在锥形走廊中使用大小为 4 的编队库进行编队导航的示意图 .....  | 49 |
| 图 5-5 在锥形走廊中使用大小为 2 的编队库进行编队导航的示意图 .....  | 49 |
| 图 5-6 OptiTrack 动作捕捉系统 .....              | 50 |
| 图 5-7 无人车实验平台 .....                       | 50 |
| 图 5-8 最窄编队模式收敛过程的示意图 .....                | 51 |
| 图 5-9 从三无人车宽度队形收敛到两无人车宽度队形的示意图 .....      | 51 |
| 图 5-10 从四无人车宽度队形收敛到三无人车宽度队形的示意图 .....     | 51 |
| 图 5-11 四辆无人车在锥形环境中编队导航过程中不同时刻的截图 .....    | 52 |
| 图 5-12 四辆无人车在有障碍物的环境中编队导航过程中不同时刻的截图 ..... | 53 |
| 图 5-13 四辆无人车在狭窄的通道中编队导航过程中不同时刻的截图 .....   | 53 |
| 图 5-14 四辆无人车在狭窄的通道中编队导航过程中的编队距离误差图 .....  | 54 |
| 图 5-15 大体积障碍物环境中的编队导航实例（0-203 秒） .....    | 54 |
| 图 5-16 大体积障碍物环境中的编队导航实例（203-301 秒） .....  | 55 |

## 表格清单

|   |    |
|---|----|
| 表 3-1 编队级路径规划中常用的函数及其功能 .....             | 16 |
| 表 3-2 不同方法对评分函数的参数设置 .....                | 27 |
| 表 3-3 不同路径规划算法在图 3-15 的场景中的评估指标 .....     | 30 |
| 表 3-4 FrmRRT*和 WidthCheck 消融实验的评估指标 ..... | 31 |
| 表 4-1 前六种编队模式变换消耗时间的对比 .....              | 39 |
| 表 4-2 后六种编队模式变换消耗时间的对比 .....              | 39 |
| 表 5-1 编队相关参数表 .....                       | 45 |
| 表 5-2 不同方法在受限环境中编队导航的性能对比 .....           | 47 |

# 第一章 绪 论

## 1.1 研究背景及意义

多机器人编队是多机器人系统的重要组成部分<sup>[1]</sup>，通常指多个机器人在执行任务过程中需要保持特定队形，以协同完成诸如搬运物体<sup>[2],[3]</sup>、灾害救援<sup>[4]</sup>、环境侦查<sup>[5]</sup>和智慧交通<sup>[6],[7]</sup>等任务。近年来，多机器人编队在各类企业场景中得到了广泛应用，具体而言：在灯光秀应用中，亿航智能（Ehang）公司实现了一人一台电脑即可操控千台飞行器完成编队表演<sup>[8]</sup>，不仅能够灵活调整编队规模，还大幅减低了飞行编队的成本，如图 1-1 (a) 所示；在协同搬运应用中，海康机器人（Hikrobot）公司开发了多机器人协同系统，能够针对大尺寸或异形货物搬运的需求，灵活调整编队形状，以协同完成不同场景下的特殊搬运任务<sup>[9]</sup>，如图 1-1 (b) 所示；在智慧交通应用中，小马智行（Pony.ai）成功实现了卡车编队自动驾驶<sup>[10]</sup>，采用“1+N”的编队模式，即 1 辆配备安全员的领航车引导 N 辆全车无人 L4 级自动驾驶重卡形成编队协同运输，累计运送货物超过 2500 万吨货物，如图 1-1 (c) 所示。此外，在 2024 年 11 月，工业和信息化部等十二部门<sup>[11]</sup>在《5G 规模化应用“扬帆”行动升级方案》中提出深化基于 5G 的编队行驶。正如上述各行业的应用，复杂环境下的多机器人协同编队受到了业界的广泛关注。

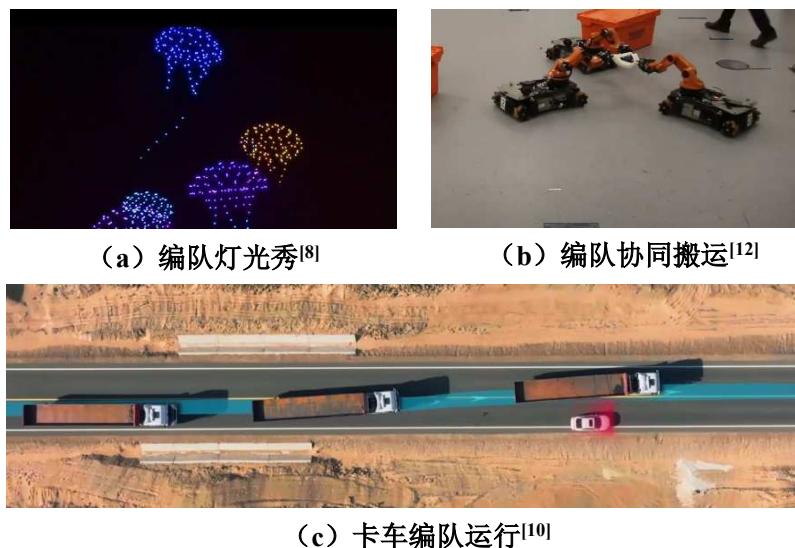


图 1-1 编队的常见应用

**Figure 1-1. Common applications of formation**

对于复杂环境中的多机器人协同编队方法，高效的避障策略和编队形状控制是确保其方法成功运行的关键环节。其中，如何在避障与编队保持之间取得平衡是需要重点考虑的问题。现有的一些方法主要基于轨迹优化，将避障和编队保持

分别作为优化目标函数的约束，并通过调整权重来权衡二者的影响。在小体积障碍物环境中，这些方法通常能较好地维持编队形状，因为其允许障碍物出现在队形内部，使得单个机器人所受的避障约束较低。然而，在大体积障碍物环境中，多个机器人会从障碍物同一侧绕行，导致其可通行区域显著缩小，从而大幅度增加机器人与障碍物发生碰撞的风险。在这种情况下，避障的优先级远高于编队保持，使得机器人为躲避障碍物而无法保持期望的队形。

综上所述，在大体积障碍物环境中，实现高效的多机器人编队控制仍面临一些挑战。为此，本文研究的重点是将避障约束和编队保持约束从目标函数中解耦，分别处理避障问题和编队保持问题。具体而言，首先设计高效的多机器人路径规划算法和可行区域检测算法，动态确定适应环境的期望队形；随后，构建高效的编队变换算法。该方案能确保机器人在复杂环境中仍能保持良好的编队质量。

## 1.2 研究现状

本研究旨在提出一种基于环境信息的多无人车编队变换方法，其核心设计思路可分为三个关键环节：首先，构建融合编队形状信息的多机器人路径规划算法，通过优化机器人与障碍物之间的安全距离来减少编队形变的频率；其次，开发高效的可行区域检测算法，为编队动态变换提供空间基础，显著降低机器人与障碍物发生碰撞的概率；最后，设计具有快速收敛特性的编队控制器，确保系统能够高效稳定地达成期望队形，从而实现编队运行与动态避障的协同。因此，本节将系统性地综述多机器人协同控制领域的研究进展，重点梳理与多机器人协同编队密切相关的三个方面：多机器人路径规划，可行区域计算以及编队控制的代表性算法。

### 1.2.1 多机器人路径规划

多机器人路径规划是单机器人路径规划的拓展，旨在实现多机器人在共同空间内的路径规划和冲突协调<sup>[13]</sup>。早期的多机器人路径规划算法主要基于 A\*算法扩展而来，有 HCA\*<sup>[14]</sup>, ID<sup>[15]</sup> 和 M\*<sup>[16], [17]</sup> 等。HCA\* 算法为每个机器人预设优先级，依次调用 A\* 规划路径，并将高优先级机器人的路径作为低优先级机器人规划时的约束。然而，当机器人数较多时，该方法引入的约束过多，将导致低优先级的机器人难以找到最优路径，甚至可能找不到可行的路径。ID 算法使每个机器人在不考虑其他机器人的情况下规划路径，并在发生碰撞冲突时，将相关机器人归为一组协同解决，从而在一定程度上减少解空间规模，提高求解效率。M\* 算法与 ID 算法相似，同样让机器人各自规划各自的路径，但不同之处在于，M\* 在规划过程中实时进行碰撞检测，一旦检测到冲突，便动态将相关机器人合并成组，以协同解决冲突。与传统基于 A\* 的改进算法不同，Sharon 等人<sup>[18]</sup> 提出了基于冲突搜索（Conflict-based Search, CBS）的方法。CBS 采用双层搜索架构来求

解最优的多机器人路径规划。其低层搜索负责单个机器人的路径规划，而高层搜索构建约束树（Constraint Tree, CT），其中每个节点代表一种冲突解决方案，并在 CT 上运行 A\*算法，以查找最优的冲突解决策略。针对 CBS 算法计算效率低的问题，改善版 CBS（Improved CBS, ICBS）<sup>[19]</sup>引入了优先级冲突机制，以优化约束树的构建过程，从而减少树节点数量，加速求解过程。ECBS<sup>[20]</sup>通过引入参数 w 来缩小节点的搜索范围，使 CBS 的最优性下降为有界最优，但相比传统 CBS，大幅提升了搜索效率，并具备更好的扩展性。CCBS<sup>[21]</sup>针对 CBS 算法在连续搜索中的不足，使其无需对等待和移动动作进行时间量化，从而提升了算法在连续时间域下的适用性。

上述方法在路径规划时优先考虑路径最短，用于编队运行会导致生成的路径在某些情况下紧贴障碍物。这使得多机器人编队在沿着路径到达终点的过程中需要频繁调整队形，从而影响编队稳定性和整体质量。TubeRRT\*<sup>[22]</sup>提出了同伦路径规划（Homotopic Path Planning），在 RRT\*（Rapidly-exploring Random Tree star）框架<sup>[23]</sup>的基础上，将树节点从二维位置拓展为考虑可行区域的二维圆。该方法生成的路径同时考虑路径长度和可行区域的大小，并采用集中式的方式求解每个机器人的路径，使得算法复杂度从  $O(nl(\log n + 1))$  减小为  $O(n(\log n + 1) + l)$ ，显著提升了算法的扩展性。在本文的研究中，将在 TubeRRT\* 的评分函数中加入编队信息项，以减少多无人车因路径距离障碍物过近而频繁调整队形的情况。

### 1.2.2 可行区域计算

无障碍物区域通常是由多个相交的凸集合而成。这种结构使得机器人规划问题能够被高效地求解。Liu 等人<sup>[24]</sup>采用迭代方式，根据每段路径生成无障碍物的椭圆，并利用椭圆与障碍物的交点构造该线段的最大凸多边形，最后将各个线段对应的凸多边形组合起来，形成沿路径的可行区域。然而，由于该方法对点云地图进行了下采样，导致部分可行区域并不安全。相比之下，Toumeh 等人<sup>[25]</sup>采用体素栅格沿特定方向依次膨胀，以填充周围的无障碍物空间，最终在由体素构成的不规则图形中计算最大的内切凸多边形，作为可行区域，并保证凸区域内都是安全的。Chen 等人<sup>[26]</sup>和 Li 等人<sup>[27]</sup>采用在线方式生成由相互重叠立方体构成的无障碍物区域，并将其作为轨迹优化的输入。然而，由于立方体的方向必须与坐标轴对齐，导致最终生成的可行区域较为保守。Gao 等人<sup>[28]</sup>提出 PCCI（Parallel Convex Cluster Inflation）算法，该方法在栅格地图中对体素进行迭代膨胀，最后形成无障碍物的凸多面体。PCCI 能够尽可能地填充周围空间，以获得更大的可行区域。然而，该算法计算复杂度较高，需要依赖并行计算来高效生成无障碍物区域。Deits 等人提出了 IRIS（Iterative Region Inflation by Semidefinite Programming）算法<sup>[29]</sup>，该算法通过交替迭代的方式利用二次规划（Quadratic Programming, QP）求解凸多边形，并使用半正定规划（Semi-definite Programming, SDP）求解椭圆。

然而，该方法在每次计算可行区域时都需要分别求解 QP 和 SDP，导致计算开销较大，尤其在多次调用时，求解效率受限。文献[12], [30], [31], [32], [33], [34]利用 IRIS 算法求解每个机器人的最大无障碍物区域，然后将所有可行区域取交集后作为所有机器人的无障碍物区域进行编队控制。Toumich 等人<sup>[35]</sup>提出了时间感知安全通道（Time-Aware Safe Corridors, TASC），考虑其他机器人未来的位置，来实时调整安全通道的大小。Zhong 等人<sup>[36]</sup>提出了一种基于球面翻转（Sphere Flipping）映射的方式来求解可通行区域。该方法通过空间翻转，使得翻转后的障碍物点所构成的凸包与原空间最内侧的障碍物点相关联。为了得到凸多面体，他们设计了一些启发式的方法将原空间最内侧的多面体裁切为凸多面体，从而便于后续的机器人规划。Guo 等人<sup>[37]</sup>提出了 RSFC(Rectangular Safe Flight Corridor) 算法，该算法通过几何计算确定路径周围的可通行区域，并根据区域大小设置合适的期望编队形状。然而，RSFC 算法仅适用于已知全局地图的情况，并且运行环境相对简单。

综上所述，现有的可行区域求解方法仍不适合用于多机器人编队。因为在具体的编队场景中，需要确保多机器人能根据环境信息选择合适的编队形状，同时与障碍物保持一定距离，减少机器人与障碍物碰撞的概率。本文的研究方法可以检测可行区域的大小并以此得到合适的队形，从而减少编队约束和避障约束发生冲突的频率。同时，通过调整局部目标点，引导多机器人在更安全的区域内实现编队运行。

### 1.2.3 编队控制

在过去十余年间，有大量研究<sup>[38]</sup>实现了多机器人编队控制系统，包括领导-跟随<sup>[39], [40], [41], [42]</sup>，虚拟结构<sup>[43], [44], [45]</sup>，势场法<sup>[46], [47]</sup>，一致性<sup>[48], [49]</sup>，图论法<sup>[50], [51]</sup>和基于行为<sup>[52], [53]</sup>等方法。具体来说，编队形状形成问题得到了广泛的研究。一部分工作集中在编队变换控制器的设计。Turpin 等人<sup>[54]</sup>利用形状向量描述机器人之间的关系，并设计分布式控制器形成期望的队形。Wang 等人<sup>[55]</sup>基于局部感知信息提出了分布式的目标分配来形成期望的编队形状，并使用局部任务交换解决局部分配冲突。贾一飞等人<sup>[56]</sup>分别利用 PID (Proportional Integral Derivative) 控制器和 LQR (Linear Quadratic Regulator) 控制器来控制机器人变换指定的编队形状。Sun 等人<sup>[57]</sup>借鉴均匀漂移（Mean Shift）的思想，实现一致性控制，形成预设的队形，并在机器人数量突然减少的情况下也能恢复队形。在文献 [58], [59], [60] 中，提出了一种类似于软体机器人的机器人结构，称为边界约束群体机器人（Boundary-constrained Swarm Robot）。该机器人由多个轮式机器人组成，相邻机器人之间通过塑料膜连接，使整体呈现出软体机器人的特性，并可通过优化目标函数控制轮式机器人是否启用，以实现不同的编队形态转换。Karimi 等人<sup>[61]</sup>进一步优化了编队控制方法，采用 R 函数的解析表达形式来描述编队形状，从而避免了数值算法的计算开销，实现了实时队形控制。然而，由于边界约

束的群体机器人依赖相邻之间的物理连接，其队形变换的自由度受到一定限制。Morgan 等人<sup>[62]</sup>以分布式的方式利用拍卖任务分配和轨迹优化来形成目标队形。Zhang 等人<sup>[63]</sup>使用编队形状参数作为反馈来控制机器人形成期望的形状。Park 等人<sup>[64]</sup>提出基于改进粒子群优化的编队控制器进行编队。然而，这些工作并未主动利用障碍物信息进行队形变换，无法适应复杂的编队任务。

另一部分工作主要考虑在复杂的环境中完成编队导航。Peng 等人<sup>[65]</sup>优化机器人的偏航角，使得多机器人能够尽可能感知更大的范围，并且构造基于距离的编队轨迹优化。然而，由于该编队定义具有旋转平移不变性，因此无法生成用户指定的形状。Guo 等人<sup>[66]</sup>利用鸽群算法和势场法控制机器人避障，同时采用一致性算法控制机器人形成期望编队。Fu 等人<sup>[67]</sup>结合人工势场算法和一致性算法，分别用于解决障碍物避障和编队形状形成。但这两种方法的避障策略仅考虑单个机器人。当机器人靠近障碍物时，为了避障而调整轨迹，可能导致无法保持预设编队形状。Huang 等人<sup>[68]</sup>能够在编队误差大于阈值时切换更窄的队形以适应环境。Chen 等人<sup>[69]</sup>采用集中式的方式计算可行区域，并根据区域的大小控制预设队形沿坐标轴方向进行伸缩变换，以适应周围环境。然而，该方法仅计算可行区域的横向宽度，只能适应简单的场景，同时，其自适应环境队形变换策略较为简单，导致在狭窄通道中难以有效保持队形。刘福琳等人<sup>[70]</sup>基于领航跟随法和人工势场法提出了 LAPF (leader artificial potential field)，使得机器人在尽量保持队形的基础上有效躲避障碍物。常路等人<sup>[71]</sup>提出了一种领航-跟随法与基于行为法相结合的多机器人编队控制算法。然而，文献[70]和[71]在通过狭窄的通道时，仍会压缩队形，使其编队形状变形。Alonso 等人<sup>[32]</sup>为了实现编队控制，利用所有机器人安全凸包的交集作为可行区域，并在该区域内重新分配期望的编队形状，再规划局部轨迹。然而，该方法计算的可行区域过于保守，限制了机器人利用环境信息自适应变换队形的能力。Lee 等人<sup>[72]</sup>预先设定了一系列编队形状，并提出了编队模式图搜索 (Formation Pattern Graph Search) 方法，以确定每个路径点上无障碍物的队形，然后生成轨迹以控制机器人到达期望位置。李兆博等人<sup>[73]</sup>在基于动态窗算法的路径评价函数中增加了队形保持因子，使得车辆构成的队形能根据障碍物自适应调整。但文献[72]和[73]方法都依赖全局地图的已知信息进行规划，缺乏对动态环境的适应能力。Quan 等人<sup>[74]</sup>在障碍物密集的环境中优化局部轨迹，以控制机器人保持编队运动。Chen 等人<sup>[75]</sup>进一步将 Quan 方法用于无人车编队，并利用无人机感知能力探索环境，为无人车提供更大的感知范围。然而，Quan 的方法在机器人编队保持过程中未充分利用周围环境来降低机器人与障碍物碰撞的风险，导致机器人在狭窄的通道中难以保持期望的队形。该算法通过构造优化问题，将避障约束和编队形成约束整合到同一个优化问题中，以确保机器人在复杂环境维持特定的形状。然而，在大体积障碍物的环境中，权衡避障和编队保持变得极具挑战性。由于避障需求远大于编队保持，机器人往往会因规避障碍物而

导致编队形状扭曲。

综上所述，近年来编队控制研究存在以下局限性：一部分研究工作未能充分考虑障碍物环境。另一部分研究工作尝试将避障与编队控制相结合，在大体积障碍物环境往往难以维持良好的编队性能。相比之下，本文提出了一种编队导航框架，该方法能够充分利用环境信息，自适应匹配合适的编队形状，并通过高效的编队控制器使所有无人机快速形成期望队形，从而在存在大体积障碍物的环境中，有效缓解避障与队形保持之间的冲突。

### 1.3 论文研究内容和章节安排

针对现有机器人编队控制算法在复杂环境中面临的问题，本文将重点研究如何提升多无人机基于环境信息自主提取编队形状的能力，以及在复杂环境中编队收敛效率。

为了高效且准确地匹配适应当前环境的编队形状，设计了一种基于可行域的中心式队形选择方法。首先，提出了一种结合编队信息的基于 RRT\*改进的路径规划算法，该算法能够根据当前形状生成适用于多无人机编队的路径。其次，针对考虑编队形状的路径生成可行域较为保守的问题，提出了最大可通行宽度检测算法，该算法通过构建二次规划问题求解最大可通行宽度，从而匹配最适合当前环境的编队形状。最后，通过多次迭代求解最大可通行宽度，选择最合适的编队形状。

此外，为了快速收敛到期望的队形，设计了一种基于任务分配的多无人机分布式队形控制算法。为了减少无人机在编队变换的总路程，采用基于最优运输的任务分配方法，计算无人机与期望队形的对应关系。为提高编队信息的一致性，提出了领导无人机重选择策略，集中式的执行基于可行域的队形选择算法和任务分配。为控制无人机快速形成期望队形，设计了结合编队位置一致性算法的基于模型预测控制（Model Predictive Control, MPC）的控制器。

围绕以上研究内容，本文的章节安排如图 1-2 所示，具体描述如下：

第一章是绪论，给出了本文的研究背景和意义，详细介绍了多机器人路径规划，可行区域计算和编队控制的现有工作，并进一步说明了本文的研究内容与章节安排。

第二章介绍了本文所涉及的相关技术，并提出一个包括基于可行区域的编队选择和基于编队一致性的队形控制的多无人机编队框架。

第三章，在第二章提出的框架基础上，详细阐述了基于可行区域的队形选择算法。首先，本章阐述了基于 RRT\*改进的编队级路径规划的算法流程；随后，根据从路径上选取的局部目标点和无人机当前位置，计算最大可通行宽度；最后，将编队级路径规划算法和最大可通行宽度检测结合，提出了一种编队形状选择算

法。通过仿真实验对比不同的路径规划算法，并对路径规划算法和最大可通行宽度算法进行消融实验，以此对所提队形选择算法的有效性进行验证。



图 1-2 论文章节安排框图

Figure 1-2. The chapter arrangement diagram of the thesis

第四章，在第二章提出的框架基础上，详细阐述了基于编队一致性的队形控

制算法。首先，本章介绍了基于最优运输的任务分配原理与作用；随后，探讨了领导无人车重选择策略的功能及其影响；接着，阐述了一种基于编队位置一致性的分布式控制器的设计思路。最后，设计了基于路径的局部避障策略，用于应对复杂环境。通过仿真实验对比不同的编队控制算法，并分析队形控制器的可扩展性，从而验证队形控制算法的高效性。

第五章将第三章和第四章算法进行组合，以完成多无人车编队导航任务，其中第三章算法用于检测环境中的障碍物，并实时判断当前最适合的编队队形；第四章算法则根据设定的队形，控制各无人车快速收敛至指定队形。组合第三章和第四章的方法将在 Gazebo 仿真平台与真实的室内场景中完成多无人车编队导航的验证。首先，在两种包含大障碍物的仿真环境下，对比不同编队控制算法的导航性能，并分析编队库规模对算法表现的影响。随后，在真实的室内场景中，设计了多组不同场景的实验，以充分验证本文提出的基于可行区域的多无人车编队变换方法在真实环境中的可行性和有效性。

第六章总结了本文的主要研究成果与贡献，并进一步探讨了未来的重点研究方向。

## 第二章 相关技术与基本框架

### 2.1 引言

本章首先介绍本文涉及的相关技术，包括无人车的运动学模型，基础图论和RRT\*算法。然后，阐述方法的整体框架图，以清晰呈现研究思路与实现流程。

### 2.2 相关技术

#### 2.2.1 无人车的运动学模型

在本文中，多无人车系统由 $N$ 辆无人车组成，其二维工作空间表示为 $\mathcal{W} \in \mathbb{R}^2$ 。环境中障碍物事先未知，用 $\mathcal{O}$ 表示。无人车的自由空间表示为 $\mathcal{F} = \mathcal{W} \setminus \mathcal{O}$ 。每辆无人车的碰撞模型均定义为半径为 $r$ 的圆，并且运动学模型采用两轮差速模型。具体定义如下，第*i*辆无人车的状态 $\mathbf{x}_i := [x_i, y_i, \theta_i]^T$ 包含其二维的位置和朝向，其中 $i \in \{1, \dots, N\}$ ，而其输入 $\mathbf{u}_i := [v_i, w_i]^T$ 包含线速度和角速度。第*i*辆无人车的运动学方程描述如下：

$$\dot{\mathbf{x}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = f(\mathbf{x}_i, \mathbf{u}_i) \quad (2-1)$$

另外，无人车的位置表示为 $\mathbf{p}_{i(\cdot)} = [p_{ix(\cdot)}, p_{iy(\cdot)}, 0]^T \in \mathbb{R}^{3 \times 1}$ 。矩阵 $\mathbf{B} = diag(1, 1, 0)$ 用于将状态 $\mathbf{x}_{i(\cdot)}$ 转换为位置 $\mathbf{p}_{i(\cdot)}$ 。

#### 2.2.2 基础图论

在编队中， $N$ 辆无人车之间的通信关系被建模成一个无向图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，其中 $\mathcal{V} = \{1, 2, \dots, N\}$ 表示顶点的集合，而 $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ 是边的集合。在图 $\mathcal{G}$ 中，第*i*个顶点 $v_i \in \mathcal{V}$ 表示第*i*辆无人车的位置 $\mathbf{p}_i$ 。连接顶点 $v_i \in \mathcal{V}$ 和顶点 $v_j \in \mathcal{V}$ 的边 $e_{ij} \in \mathcal{E}$ 表示第*i*辆无人车和第*j*辆无人车之间可以相互通信。在本文中，假设所有无人车之间都可以相互通信，意味着对应的通信拓扑结构 $\mathcal{G}$ 为完全图。图中每条边 $e_{ij} \in \mathcal{E}$ 的权重 $\omega_{ij}$ 定义为第*i*辆无人车和第*j*辆无人车之间的距离，描述如下：

$$\omega_{ij} = \| \mathbf{p}_i - \mathbf{p}_j \|^2 \quad (2-2)$$

图 $\mathcal{G}$ 相关的邻边矩阵 $\mathbf{A} \in \mathbb{R}^{N \times N}$ 中每个元素 $A_{ij}$ 等于边权重 $\omega_{ij}$ ，而度矩阵 $\mathbf{D} \in \mathbb{R}^{N \times N}$ 对角线以外的元素都为0。度矩阵中每个元素 $D_{ii}$ 定义如下：

$$D_{ij} = \begin{cases} \sum_{j=1}^N A_{ij}, & \text{if } i=j \\ 0, & \text{otherwise} \end{cases} \quad (2-3)$$

图  $\mathcal{G}$  的拉普拉斯矩阵定义如下：

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (2-4)$$

对称归一化的拉普拉斯矩阵（Symmetric Normalized Laplacian Matrix）定义如下：

$$\hat{\mathbf{L}} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \quad (2-5)$$

### 2.2.3 RRT\*算法

快速随机搜索树（Rapidly-exploring Random Tree, RRT）是一种常用的基于采样的路径规划算法。在此基础上，Karaman 等人提出了 RRT\* 算法<sup>[23]</sup>。RRT\* 算法不仅能像 RRT 那样快速覆盖搜索空间，而且能够随着采样数量的增加，逐步收敛到最优路径。RRT\* 规划路径的求解过程如图 2-1 所示，具体描述如下：

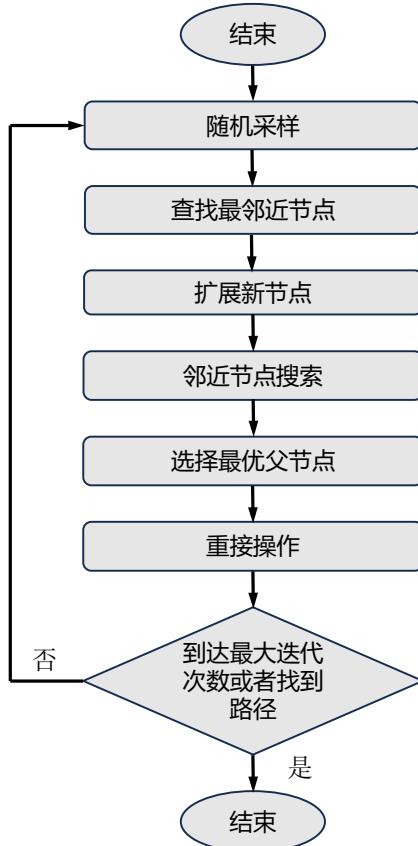


图 2-1 RRT\* 算法流程图

Figure 2-1 Flowchart of the RRT\* algorithm

- (1) 随机采样：在状态空间中以一定概率采集随机点；
- (2) 查找最邻近节点：在已有树结构中找到与随机点距离最近的节点作为拓展起点；
- (3) 拓展新节点：从拓展起点向随机点方向生成新节点，并判断该连接是

否满足环境约束；

- (4) 邻近节点搜索：以新节点为中心，设定半径，搜索树中所有位于该指定半径范围内的节点；
- (5) 选择最优父节点：在邻近节点中选择使路径代价最小的节点作为新节点的父节点；
- (6) 重接操作：遍历邻近节点，将那些通过新节点连接后代价更低的节点的父节点设置为新节点。
- (7) 重复迭代：上述过程持续迭代。若达到最大迭代次数或找到可接受路径，则终止迭代。

RRT\*算法通过多次重接操作不断优化搜索树结构，使得生成的路径能逐渐趋近于最优。RRT\*的时间复杂度主要受到邻近节点搜索和重接操作的影响。在每次插入新节点时，RRT\*算法都在已有节点中搜索邻近节点，复杂度为  $O(\log n)$ 。因此，对于包含  $n$  个节点的树，其总体平均时间复杂度为  $O(n \log n)$ 。

### 2.3 框架介绍

为了改善多无人车在复杂环境中自主编队导航的能力，本文提出了一个编队控制框架，包括基于可行区域的队形选择模块和基于编队一致性的队形控制模块。该框架将队形选择模块作为前端，负责规划路径和队形选择；队形控制模块作为后端，实现具体的编队运动控制，从而共同完成编队导航任务。具体而言，队形选择模块包含编队级路径规划器，最大可行宽度检测子模块；队形控制模块包含任务分配子模块、领导无人车重映射子模块、编队控制器子模块和基于路径的局部避障策略。该框架设计是为了保证算法的高效性并减少计算量：临时领导无人车集中式执行编队级路径规划，最大可行区域检测和任务分配，从而快速实现编队信息的一致性，而对于编队变换控制器和基于路径的局部避障策略则由每辆无人车分布式计算，减少计算负担。此外，本文还探讨了根据任务分配结果确定新的领导无人车的策略。在介绍框架前先引入编队宽度和可通行宽度的概念。

为了用数学形式表示编队形状，用编队模式  $\mathcal{P}_{(.)}$  表示，其矩阵形式表示为  $\mathbf{P}_f = \{\mathbf{p}_{f,1}, \dots, \mathbf{p}_{f,i}, \dots, \mathbf{p}_{f,N}\} \in \mathbb{R}^{3 \times N}$ ，其中  $\mathbf{p}_{f,i} = [p_{f,ix}, p_{f,iy}, 0]^T$  表示第  $i$  辆无人车的相对位置， $\mathbf{p}_{f,0}$  则表示编队坐标系的原点。本文中编队模式仅定义每辆无人车在编队坐标系中的横坐标和纵坐标，而不涉及无人车的朝向。为了根据最大可通行宽度选择最合适的编队形状，本文预先设计了一个编队库，其中包含不同尺寸的编队形状。

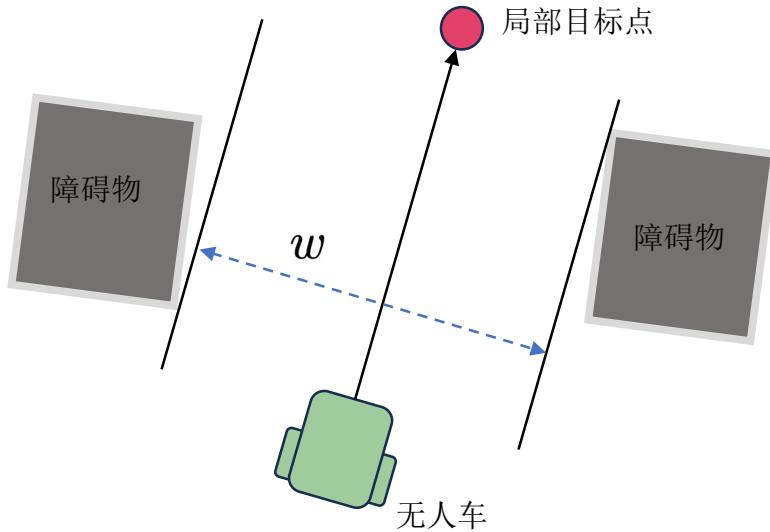


图 2-2 可通行宽度示意图  
Figure 2-2. Diagram of feasible width

在执行编队导航任务时，无人车通过实时检测可通行宽度的大小，从预设的编队库中选择一个宽度小于该可通行宽度的编队模式。具体而言，可通行宽度是指垂直于无人车运动方向的无障碍物间隔。例如，在图 2-2 中，蓝色虚线的长度  $w$  表示可通行宽度。而编队模式宽度指编队中垂直于无人车运动方向上最远两辆无人车之间的距离。如图 2-3 所示，从左到右编队模式的宽度逐渐减小，其中队形模式的宽度定义为无人车 R2 与无人车 R3 之间的距离，除了编队模式  $\mathcal{P}_4$  的宽度是无人车的直径。

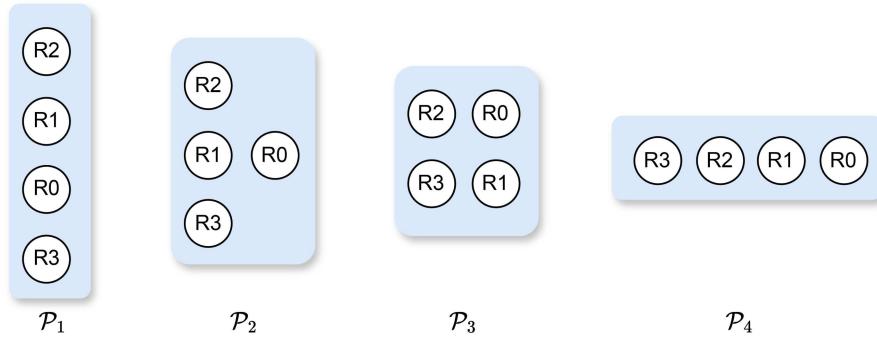


图 2-3 四辆无人车的预设编队库  
Figure 2-3. The predefined formation library of four unmanned vehicles

本文方法的框架如图 2-4 所示，整个算法包含两个流程，分别为编队流程和避障流程。首先介绍编队流程：用户设定导航任务的终点  $\mathbf{p}_g$ ，并将无人车当前的位置与终点作为编队级路径规划器（详见 3.2 节）的输入。领导无人车根据这些信息规划路径  $\mathbf{P}$  并根据无人车的视野范围从路径上获取局部目标点  $\mathbf{p}_{lg}$ ，用于引导无人车到达终点同时确保多无人车编队与障碍物之间保持适当距离，减少队形

调整的频率。此外，领导无人车将路径  $\mathbf{P}$  共享给其他无人车。然后，在最大可行宽度检测模块（详见 3.4 节）中，领导无人车根据周围感知的障碍物计算当前的最大可行宽度  $w_l$ （详见 3.3 节），并结合其他无人车检测的可行宽度  $w_o$ ，共同决定是否调整当前期望的编队形状  $\mathcal{P}_d^*$ ，并根据宽度检测的结果调整更安全的局部目标点  $\mathbf{p}_{ig,r}$ 。编队信息  $w_l$  和  $\mathcal{P}_d^*$  通过无线网络共享给其他无人车，同时作为队形控制模块（详见第四章）的输入。在该模块中包含三个子模块以实现高效的队形控制，分别为任务分配，领导机器人重映射和编队控制器。任务分配模块（详见 4.2 节）计算当前无人车与期望编队之间的对应关系，以最小化编队变换所需的总距离。随后，分配结果  $\gamma^*$  由领导无人车共享给其他无人车。领导无人车重映射模块（详见 4.3 节）用于将编队模式中视野最佳的无人车选为新的领导无人车。为了实时控制无人车到达期望的编队位置并避免相互碰撞，设计了高效的编队控制器（详见 4.4 节）。与领导无人车不同，其他无人车通过宽度检测优化局部目标点  $\mathbf{p}_{ig,r}$ ，并根据接收到的编队信息控制其在编队中的相对位置。

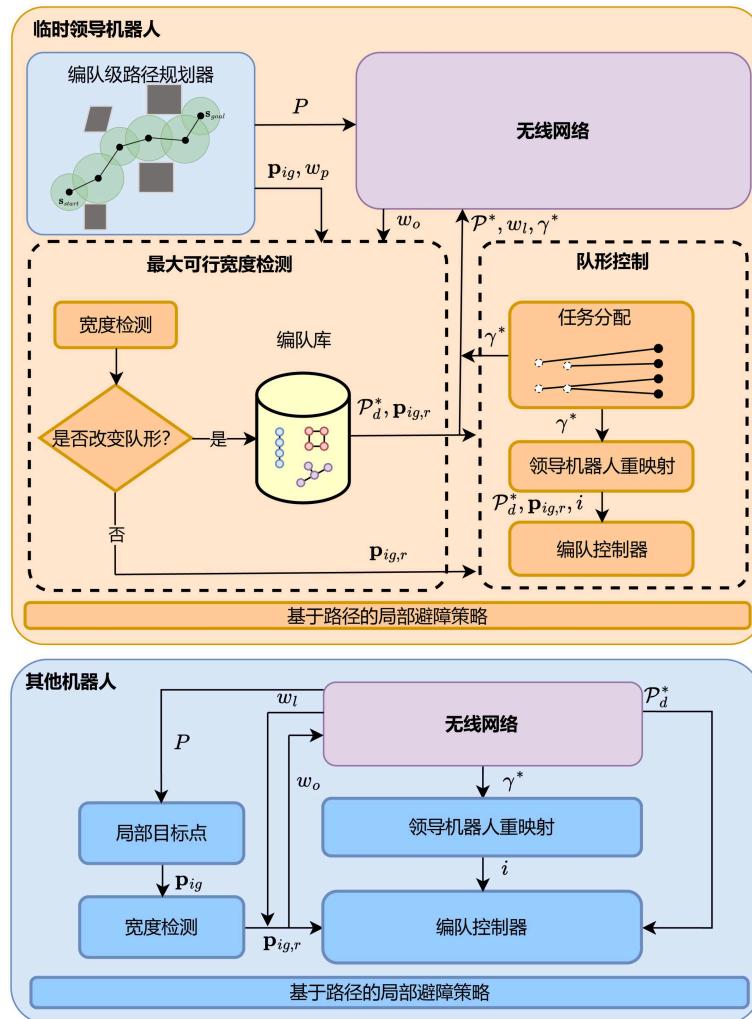


图 2-4 自适应编队变换方法框架图

Figure 2-4. Framework diagram of adaptive formation transformation method

尽管编队级路径规划器和最大可通行区域检测模块能够确保无人车编队过程中与障碍物保持一定距离，但在复杂的环境中，仍可能出现无人车距离障碍物距离过近，从而增加碰撞风险。为了保证无人车的安全，当其与障碍物之间的距离小于设定的阈值，无人车将从编队流程切换到避障流程，利用基于路径的局部避障策略，快速远离障碍物；而当距离大于阈值时，无人车则重新切换回编队流程，调整队形以保持期望的编队形状。

## 2.4 本章小结

本章介绍了相关技术，然后通过对本文方法框架的简要概述展示了实现编队导航的方案。首先，介绍本方法中所采用的无人车运动学模型，基础图论和 RRT 算法。然后，提出了一个结合基于可行区域的队形选择与基于编队一致性的队形控制的多无人车编队框架。

## 第三章 基于可行区域的队形选择

### 3.1 引言

为了使无人车能够根据周围的环境信息自适应地变换队形，首先需要确定适合当前环境的编队形状。因此，本章提出了一种基于可行区域的队形选择方法，主要包括两个部分：编队级路径规划（FormationRRT Star, FrmRRT\*）和基于局部目标点方向的无障碍物宽度检测（WidthCheck）。该方法采用由粗到细的求解策略，逐步确定最适合编队的可行区域。具体而言，FrmRRT\*是一种在 RRT\*基础上改进并融合编队信息的路径规划算法。受到 Mao 方法<sup>[22]</sup>的启发，该方法将路径搜索树的节点从二维位置扩展为二维圆，其中圆半径表示路径点与障碍物的距离。FrmRRT\*规划的结果是一组相交圆，使路径点能与障碍物保持一定的距离，从而允许无人车在保持编队的同时沿着路径运动。此外，利用评分函数可以调整路径点和障碍物之间的距离。然而，FrmRRT\*用圆形近似可行区域，相对来说比较保守。为了更精确地获取可行区域用于匹配最合适的编队形状，提出了无障碍物宽度检测方法。该方法是结合周围障碍物分布和局部目标点，构建二次规划问题，以求解无障碍物宽度，并据此调整局部目标点。随后，通过多次迭代计算可通行宽度，从预设的编队库中选择最适合当前环境约束的队形。在队形选择方法中，FrmRRT\*引导无人车向终点移动，并为无障碍物宽度检测提供局部目标点，同时，其规划的圆直径可作为队形选择的参考。而无障碍物宽度检测算法需要利用 FrmRRT\*的路径进一步评估环境的可行区域。

本章的结构安排如下：首先，详细说明基于 RRT\*改进的编队级路径规划算法的实现流程（详见 3.2 节）。然后，详细阐述最大可通宽度检测算法的实现方法（详见 3.3 节）。接着，给出了基于可行区域的队形选择的算法（详见 3.4 节）。随后，对比不同路径规划算法，分析编队级路径规划关键参数对结果的影响，并进行了 FrmRRT\*和 WidthCheck 算法的消融实验（详见 3.5 节）。最后，对基于可行区域的队形选择算法及实验结果进行了总结（详见 3.6 节）。

### 3.2 基于 RRT\*改进的编队级路径规划

本章提出的编队级路径规划 FrmRRT\*在 RRT\*的基础上拓展了节点的维度，其中每个节点不仅包含位置，还引入了当前位置与最近障碍物之间的距离，可用于辅助编队形状选择。此外，在计算节点的路径成本时，额外考虑了编队信息，使得所规划的路径和障碍物之间的距离与当前编队信息能更好的匹配。

FrmRRT\*也是通过在地图中随机采样节点构建多叉树，然后在树上搜索从起点到终点的可行路径。多叉树定义为  $\mathcal{T} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{N_{rrt}}\}$ ，其中  $N_{rrt}$  是 FrmRRT\* 迭代的次数，树节点  $\mathbf{s} \in \mathbb{R}^3$  定义为二维圆，包含圆心位置  $\mathbf{q} \in \mathbb{R}^2$  和半径  $r \in \mathbb{R}$ 。圆心表示路径点的位置，而半径表示路径点到最近障碍物的距离。通过设计具有不同目的的评分函数，可以生成一条由依次相交的圆形组成的路径，使路径点始终与障碍物保持一定距离，从而便于多辆无人车在保持编队的同时顺利穿越复杂的环境。本节常用函数列于表 3-1，完整的编队级路径规划算法流程如算法 1 所示。

表 3-1 编队级路径规划中常用的函数及其功能

Table 3-1. Common functions and their functionalities in formation-level path planning

| 函数  | 功能  |
|---|---|
| <b>UniformSample</b> ( $\mathcal{M}$ )                | 在地图 $\mathcal{M}$ 上均匀采样节点                             |
| <b>InformedSampe</b> ( $\mathbf{s}_1, \mathbf{s}_2$ ) | 在由节点 $\mathbf{s}_1$ 和 $\mathbf{s}_2$ 作为焦点的椭圆上均匀采样     |
| <b>Nearest</b> ( $\mathbf{s}_n, \mathcal{T}$ )        | 在多叉树 $\mathcal{T}$ 上查找距离 $\mathbf{s}$ 最近的节点           |
| <b>ExtendTree</b> ( $\mathbf{s}_1, \mathbf{s}_2$ )    | 在由节点 $\mathbf{s}_1$ 和 $\mathbf{s}_2$ 构成的直线上查找新节点      |
| <b>NearConnect</b> ( $\mathbf{s}, \mathcal{T}$ )      | 获取多叉树上与节点 $\mathbf{s}$ 相交的节点集合                        |
| <b>ConnectSphere</b> ( $\mathbf{S}, \mathbf{s}$ )     | 将节点与在节点集合 $\mathbf{S}$ 中代价最低的节点相连                     |
| <b>Rewire</b> ( $\mathbf{s}, \mathbf{S}$ )            | 利用节点 $\mathbf{s}$ 优化集合 $\mathbf{S}$ 中节点的父节点           |
| <b>Score</b> ( $\mathbf{s}_1, \mathbf{s}_2$ )         | 根据公式(3-1)计算从节点 $\mathbf{s}_1$ 到达 $\mathbf{s}_2$ 所需的代价 |
| <b>Cost</b> ( $\mathbf{s}$ )                          | 表示从节点 $\mathbf{s}$ 到达起始节点的累计代价                        |
| <b>CheckObstacle</b> ( $\mathbf{s}_1, \mathbf{s}_2$ ) | 检查节点 $\mathbf{s}_1$ 与 $\mathbf{s}_2$ 构成的直线是否与障碍物相交    |

首先，初始化多叉树  $\mathcal{T}$ ，将无人车当前位置设为根节点  $\mathbf{s}_{start}$ ，并将终点  $\mathbf{p}_g$  作为目标树节点  $\mathbf{s}_{goal}$  的位置。同时，定义变量 *FoundSolution* 以表示是否找到可行路径，并将其初始化为 *False*。算法刚开始时，在地图  $\mathcal{M}$  上均匀采样树节点  $\mathbf{s}_{random}$ ，如图 3-1 所示。随后，如图 3-2 所示，通过遍历树  $\mathcal{T}$  搜索距离采样点  $\mathbf{s}_{random}$  最近的已有节点  $\mathbf{s}_{nearest}$ ，具体的实现细节可参考算法 2 的伪代码。

---

### 算法1 基于 RRT\*改进的编队路径规划

---

符号：多叉树  $\mathcal{T}$ ，节点  $\mathbf{s}$ ，节点最小半径  $r_{min}$ ，路径  $\mathbf{P}$ ，地图  $\mathcal{M}$ ；

```

1: 初始化:  $\mathcal{T} \leftarrow \emptyset \cup \{\mathbf{s}_{start}\}$ ,  $\mathbf{q}_{goal} \leftarrow \mathbf{p}_g$ , FoundSolution  $\leftarrow False$ 
2: for  $i = 1$  to  $N_{rrt}$  do
3:   if not FoundSolution then
4:      $\mathbf{s}_{random} \leftarrow \text{UniformSample}(\mathcal{M})$ 
5:   else

```

---

## 算法 1 基于 RRT\*改进的编队路径规划 (续表)

---

```

6:       $\mathbf{s}_{random} \leftarrow \text{InformedSample}(\mathbf{s}_{start}, \mathbf{s}_{goal})$ 
7:      end if
8:       $\mathbf{s}_{nearest} \leftarrow \text{Nearest}(\mathbf{s}_{random}, \mathcal{T})$ 
9:       $\mathbf{s}_{new} \leftarrow \text{ExtendTree}(\mathbf{s}_{nearest}, \mathbf{s}_{random})$ 
10:     if  $r_{new} > r_{min}$  then
11:          $\mathbf{S}_{near} \leftarrow \text{NearConnect}(\mathbf{s}_{new}, \mathcal{T})$ 
12:          $\mathcal{T} \leftarrow \text{ConnectSphere}(\mathbf{S}_{near}, \mathbf{s}_{new})$ 
13:          $\mathcal{T} \leftarrow \text{Rewire}(\mathbf{s}_{new}, \mathbf{S}_{near})$ 
14:         if  $\|\mathbf{q}_{goal} - \mathbf{q}_{new}\| < d_{tol}$  then
15:             FoundSolution  $\leftarrow True$ 
16:         end if
17:     end if
18: end for
19:  $\mathbf{P} \leftarrow \text{RetrievePath}(\mathcal{T}, \mathbf{s}_{goal})$  // 根据多叉树和终点回溯到起点, 得到路径
20: Return  $\mathbf{P}$ 

```

---

在算法 2 中, 通过在节点  $\mathbf{s}_{random}$  和节点  $\mathbf{s}_{nearest}$  的位置组成的线段上选择与节点  $\mathbf{s}_{nearest}$  相交的  $\mathbf{s}_{new}$ , 运行结果如图 3-3 所示, 算法中的 **FindMaxRadius**( $\mathbf{q}_{new}$ ) 是在 ESDF (Euclidean Signed Distance Field) 地图上获取位置  $\mathbf{q}_{new}$  与最近障碍物的距离。

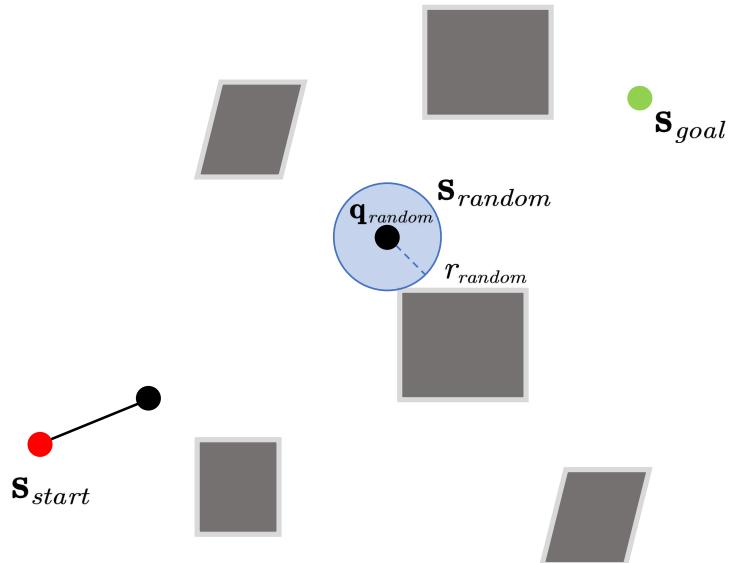


图 3-1 均匀采样节点的示意图

Figure 3-1. Diagram of uniformly sampled nodes

---

**算法2 基于邻近节点与随机节点的新节点生成**


---

**输入:** 采样节点  $\mathbf{s}_{random}$ , 最近节点  $\mathbf{s}_{nearest}$   
**输出:** 新节点  $\mathbf{s}_{new}$

```

1:    $(\mathbf{q}_{new}, r_{new}) \leftarrow \mathbf{s}_{random}$ ,  $(\mathbf{q}_{nearest}, r_{nearest}) \leftarrow \mathbf{s}_{nearest}$ 
2:    $d \leftarrow \|\mathbf{q}_{new} - \mathbf{q}_{nearest}\|$ 
3:    $\mathbf{t} \leftarrow (\mathbf{q}_{new} - \mathbf{q}_{nearest}) / \|\mathbf{q}_{new} - \mathbf{q}_{nearest}\|$ 
4:   while  $r_{new} + r_{nearest} \leq d$  or  $d < \max(r_{new}, r_{nearest})$  do
5:      $d \leftarrow \max(r_{new}, r_{nearest})$ 
6:      $\mathbf{q}_{new} \leftarrow \mathbf{q}_{nearest} + d \cdot \mathbf{t}$ 
7:      $r_{new} \leftarrow \text{FindMaxRadius}(\mathbf{q}_{new})$  // 获取与最近障碍物的距离
8:   end while
return  $\mathbf{s}_{new} \leftarrow (\mathbf{q}_{new}, r_{new})$ 

```

---

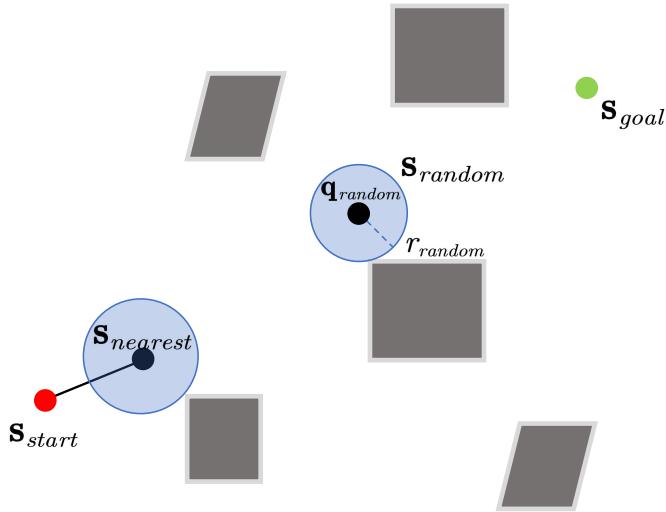


图 3-2 多叉树上距离随机节点最近节点的示意图

Figure 3-2. Diagram of the nearest node to a random node in a multi-way tree

回到算法 1 中, 当从 **ExtendTree**( $\mathbf{s}_{nearest}, \mathbf{s}_{random}$ ) 返回的节点  $\mathbf{s}_{new}$  的圆半径  $r_{new}$  大于预设的最小半径  $r_{min}$  时, 考虑将其加入树  $T$ 。接着, **NearConnect**( $\mathbf{s}_{new}, T$ ) 通过遍历树  $T$  来获取与  $\mathbf{s}_{new}$  相交的节点, 并用集合  $\mathbf{S}_{near}$  来表示, 具体如图 3-4 所示。**ConnectSphere**( $\mathbf{S}_{near}, \mathbf{s}_{new}$ ) 在节点集合  $\mathbf{S}_{near}$  中查找  $\mathbf{s}_{new}$  的父节点, 使得从节点  $\mathbf{s}_{new}$  回溯到起点  $\mathbf{s}_{start}$  的分数总和最小, 具体实现细节可参考算法 3, 其中 **Score**( $\mathbf{s}_{nearest}, \mathbf{s}_{new}$ ) 表示节点间的评分函数。具体来说, 评分函数设计了不同的目标项, 包括节点之间的距离, 两个节点构成的圆相交的面积和与期望编队形状宽度的差值, 具体表达式如下:

$$\text{Score} = \rho_d \frac{\|\mathbf{q}_{new} - \mathbf{q}_{near}\|}{\|\mathbf{q}_{goal} - \mathbf{q}_{start}\|} + \rho_v (V_{int} + \varepsilon)^{-1} + \rho_w \frac{|r_{new} + r_{near} - 2w_{des}|}{2w_{des}} \quad (3-1)$$

其中  $\rho_d$ 、 $\rho_v$  和  $\rho_w$  分别表示不同目标项的权重， $V_{int}$  表示节点  $s_{new}$  和  $s_{near}$  构成的圆相交面积。 $\varepsilon$  是常数，用于防止相交的面积太小，即用于防止  $(V_{int} + \varepsilon)^{-1}$  在公式(3-1)的占比过大。 $w_{des}$  表示期望的编队形状宽度。通过适当调整公式(3-1)中的权重  $\rho_w$ ，可以使得规划的路径点与障碍物的距离更符合期望的编队形状，从而减少因路径点过于接近障碍物而导致不必要的编队变换。在算法 3 中，**Cost**( $s_{near}$ ) 计算的是从起点  $s_{start}$  到节点  $s_{near}$  之间的总代价。具体而言，该计算过程通过依次遍历  $s_{start}$  和  $s_{near}$  之间的路径点，对每一对相邻的节点应用评分函数进行计算，并将所有得分累加，最终得到 **Cost**( $s_{near}$ ) 的数值。**CheckObstacle**( $s_{new}, s_{near}$ ) 用于检测节点  $s_{new}$  和节点  $s_{near}$  位置组成的线段是否与障碍物相交。算法 3 通过遍历节点集合  $S_{near}$ ，寻找从  $s_{start}$  到  $s_{new}$  之间累积分数最小的节点  $s_{min}$ ，并将其设为  $s_{new}$  的父节点，从而使  $s_{new}$  加入树  $\mathcal{T}$ 。

---

### 算法3 基于代价最小化的父节点连接

---

```

1:    $s_{min} \leftarrow s_{nearest} \in S_{near}$ 
2:    $c_{min} \leftarrow \text{Cost}(s_{nearest}) + \text{Score}(s_{nearest}, s_{new})$ 
3:   for  $s_{near} \in S_{near}$  do
4:     if  $\text{Cost}(s_{near}) + \text{Score}(s_{near}, s_{new}) < c_{min}$  then
5:       if CheckObstacle( $s_{near}, s_{new}$ ) then
6:          $s_{min} \leftarrow s_{near}$ 
7:          $c_{min} \leftarrow \text{Cost}(s_{near}) + \text{Score}(s_{near}, s_{new})$ 
8:       end if
9:     end if
10:   end if
11:    $\mathcal{T} \leftarrow \mathcal{T} \cup \{(s_{near}, s_{new})\}$ 

```

---

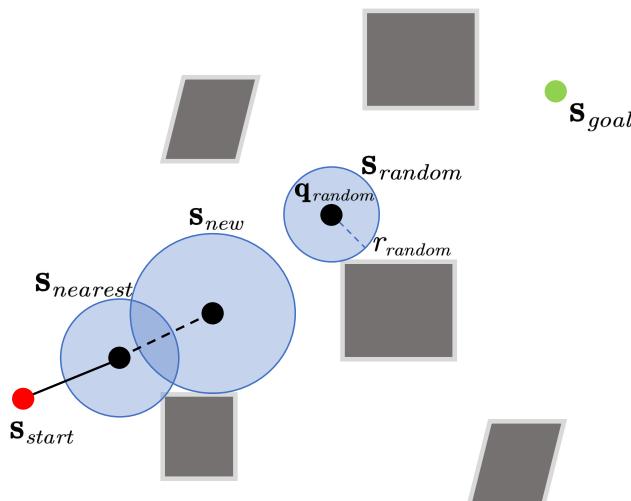


图 3-3 搜索最新节点示意图

Figure 3-3. Diagram of searching for the latest node

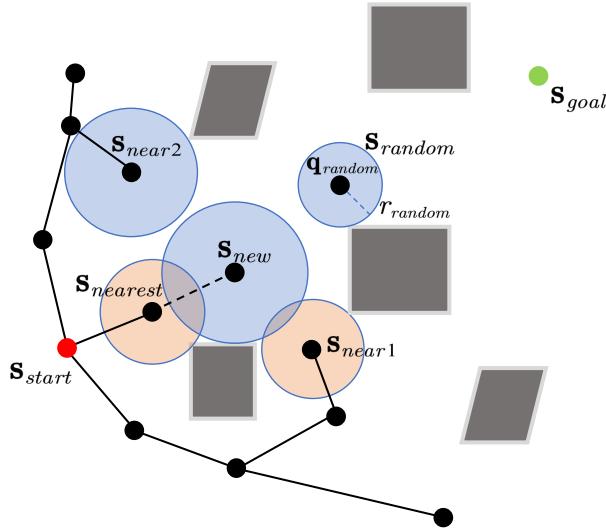


图 3-4 查找与新节点相交的节点集合的示意图

Figure 3-4. Diagram of finding the set of nodes intersecting with the new node

当  $s_{new}$  加入树  $T$  后, 算法 1 中的 **Rewire**( $s_{new}, S_{near}$ ) 检查其邻近节点集合  $S_{near}$ , 并尝试连接  $s_{new}$  作为父节点, 以查找代价更低的路径, 具体实现细节在算法 4。

---

#### 算法4 更新邻近节点的父节点以降低路径代价

---

```

1:   for  $s_{near} \in S_{near}$  do
2:     if CheckObstacle( $s_{near}, s_{new}$ ) then
3:       if Cost( $s_{new}) + Score(s_{new}, s_{near}) < Cost(s_{near})$  then
4:         Parent( $s_{near}) \leftarrow s_{new}$  // 将节点  $s_{new}$  作为节点  $s_{near}$  的父节点
5:       end if
6:     end if
7:   end for

```

---

在图 3-5 中, 节点  $s_{near1}$  通过与节点  $s_{new}$  连接可以获得更低的路径代价, 因此, 将  $s_{near1}$  原来的父节点 (图中的虚线连接) 修改为  $s_{new}$ , 从而优化整体路径。在算法 1 中, 经过多次迭代后, 节点  $q_{new}$  与终点  $q_{goal}$  之间的距离小于设定阈值  $d_{tol}$  时, 将 *FoundSolution* 置为 *True*, 表示已成功找到了一条从起点到终点的可行路径。接下来根据 Informed RRT\*<sup>[76]</sup>的思想, 缩小树节点的采样空间, 提升优化路径的效率。在算法 1 中, **InformedSample**( $s_{start}, s_{goal}$ ) 将采样空间缩小为椭圆区域, 其中椭圆的参数是由起点  $s_{start}$ , 终点  $s_{goal}$  和路径长度  $L(P)$  来确定:

$$c = 0.5 \cdot \|q_{start} - q_{goal}\| \\ a = 0.5 \cdot L(P) \quad (3-2)$$

其中  $c$  是椭圆的焦距,  $a$  是椭圆的长轴, 短轴为  $b = \sqrt{a^2 - c^2}$ 。如图 3-6 所示, 深蓝色椭圆区域表示新的采样空间, 其中  $s_{random}$  将在该椭圆区域内均匀采样, 随着路径代价的降低, 椭圆区域逐步减少, 以提高采样效率。最终, 迭代次数到达

$N_{rrt}$  时, 算法 1 返回搜索的路径  $P$ 。

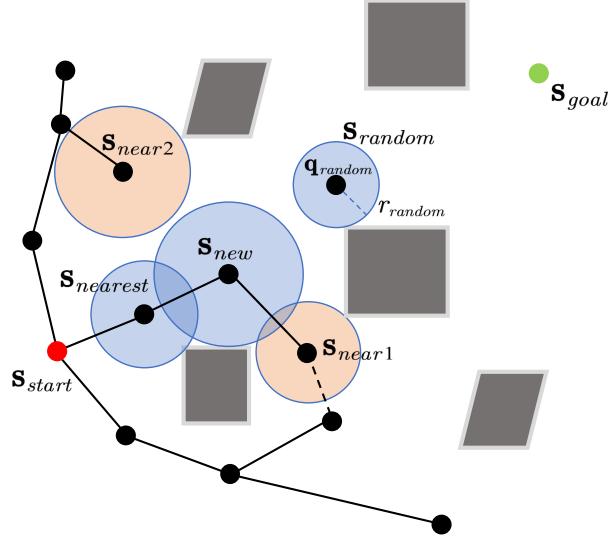


图 3-5 更新父节点的示意图

Figure 3-5. Diagram of updating the parent node

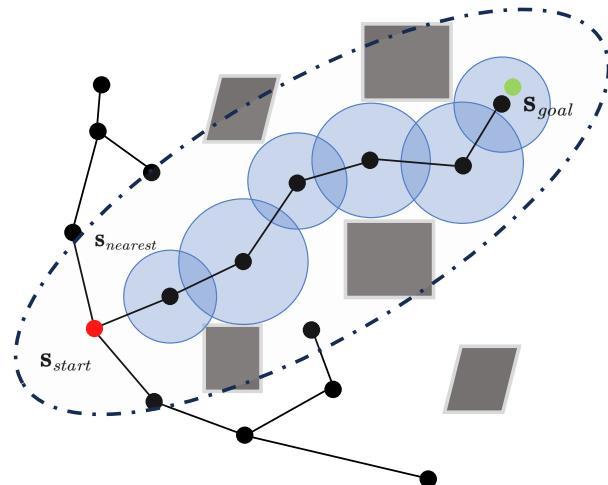


图 3-6 椭圆采样区域的示意图

Figure 3-6. Diagram of the elliptical sampling area

根据 FmRRT\*生成的相交圆路径可为多无人车编队运行提供与障碍物保持安全距离的路径点。通过对路径点进行插值计算, 可获取路径上任意点的近似无障碍物范围, 从而选择合适的编队形状, 以合理分配多无人车的运动路径。具体的插值计算中, 使用  $w_{rrt} \in \mathbb{R}$  表示路径点的无障碍物范围。如图 3-7 所示, 根据无人车的视野范围在路径上通过二分法快速从路径上搜索局部目标点  $\mathbf{p}_{lg}$  (用蓝色点表示), 利用距离  $\mathbf{p}_{lg}$  最近的相交圆弦长和圆半径 (黑色的虚线) 来插值计算无障碍物范围  $w_{rrt}$ :

$$k_r = \frac{\|\mathbf{p}_{lg} - \mathbf{q}_{i,0}\|}{\|\mathbf{q}_{i,0} - \mathbf{q}_{j,0}\|} \quad (3-3)$$

$$w_{rrt} = \|\mathbf{p}_{lg,1} - \mathbf{p}_{lg,2}\| = \|(\mathbf{q}_{i,1} - \mathbf{q}_{i,2}) + k_r(\mathbf{q}_{j,1} - \mathbf{q}_{j,2} - \mathbf{q}_{i,1} + \mathbf{q}_{i,2})\| \quad (3-4)$$

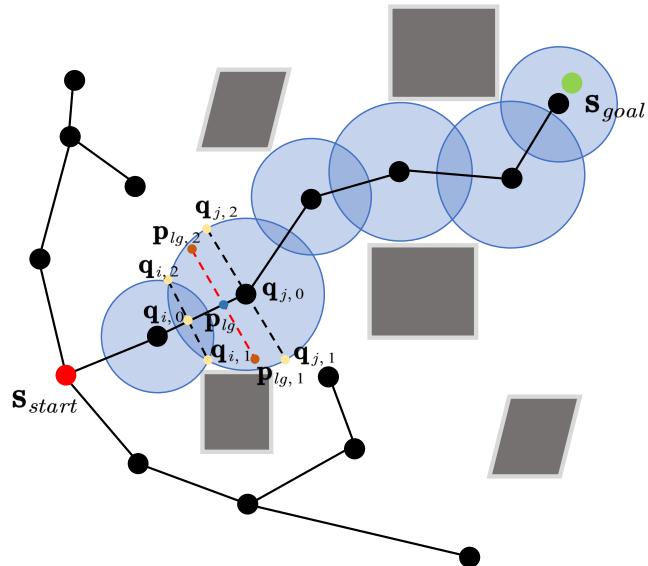


图 3-7 计算无障碍物范围的示意图

Figure 3-7. Diagram of calculating the obstacle-free region

根据插值计算的无障碍物范围  $w_{rrt}$ ，从编队库中匹配最合适的编队形状。如图 3-8 所示，以 4 辆无人机为例，依据每个路径点的无障碍物范围匹配不同的编队形状，以适应复杂环境。编队形状中的每个位置均由预先编号标识，将相同编号的编队位置依次连接后，可生成与无人机数量相匹配的路径，并分别分配给对应编号的无人机。如图 3-8 所示，不同的编号用不同的颜色表示。

具体而言，编队级路径规划算法采用领导无人机中心式的求解方式，由领导无人机计算 FrmRRT\* 生成全局路径，并将其共享给所有无人机，使得每辆无人机能够对未来环境具备一定的预测能力，从而有效减少与障碍物碰撞的风险。此方法随着无人机数量的增加，计算复杂度不会显著增加。对于复杂的障碍物环境，采样节点的数量在一定程度上有所增加，使得该方法的计算时间略微增加。

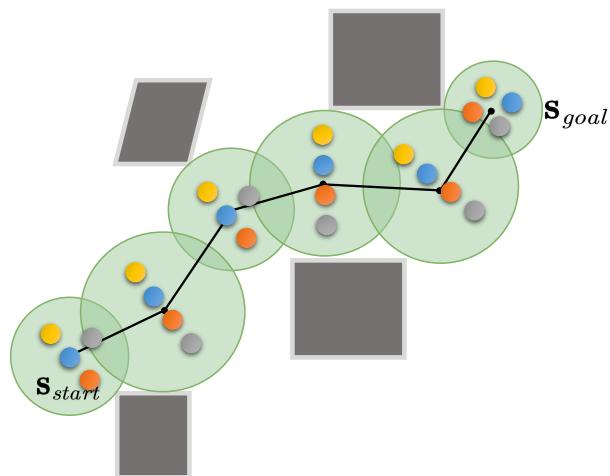
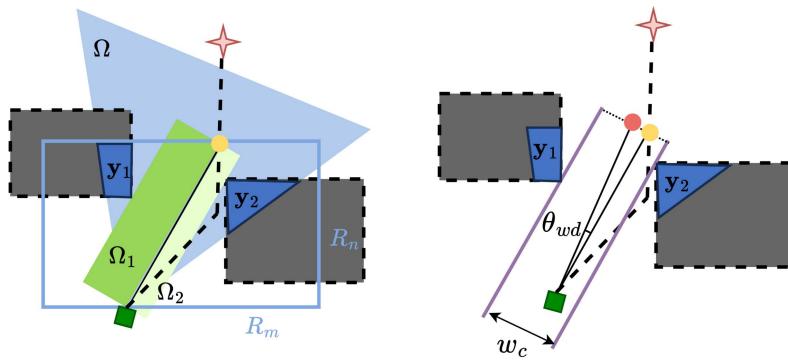


图 3-8 一条路径预测不同编队形状的例子

Figure 3-8. An example of predicting different formation shapes along a path

### 3.3 最大可通行宽度检测

为了更精确地提取环境中的可行区域，并确保其适用于无人车编队运动，提出了最大可通行宽度检测方法 WidthCheck。该方法定义的宽度是垂直于无人车运动方向的最大无障碍物距离。由于无人车沿着路径运动，而路径本身已避开障碍物，因此当前环境可通行无人车数量取决于垂直于运动方向的无障碍物距离。可通行宽度直接影响编队形状的选择，使其能够适应环境约束，从而确保多辆无人车能够顺利通过狭窄区域。此外，该方法仅需要通过求解二次规划问题来获取最大可通行宽度，降低计算复杂度，使得无人车快速的求解可通行宽度，并确定最合适编队形状，从而提升无人车编队在复杂环境中的适应性。具体而言，最大可通行宽度检测方法描述如下：



(a) 宽度检测提及的元素

(b) 宽度检测的结果

(c) 图(a)和图(b)的符号说明

图 3-9 宽度检测示意图

**Figure 3-9. Diagram of width detection**

如图 3-9 所示，无人车利用深度相机实时感知附近的障碍物，其视野范围 (Field of View, FOV) 用  $\Omega$  表示。为了更精确的获取无人车周围的障碍物信息，在无人车前方设置局部区域，正如图 3-9 中的浅蓝色矩形，边长分别为  $R_m$  和  $R_n$ 。 $R_m$  是预设编队库中最大的编队模式宽度，如中的  $\mathcal{P}_1$ 。 $R_n$  的值等于无人车当前位置  $\mathbf{p}_{ic}$  和局部目标点  $\mathbf{p}_{ig}$  在纵坐标上的差值，其中  $\mathbf{p}_{ig}$  是从 FrmRRT\* 求解的路径上截取固定长度的路径点。值得说明的是，本文提到的无人车前方或运动方向都是指向量  $\mathbf{p}_{ig} - \mathbf{p}_{ic}$  的方向。为了加速最大可通行宽度检测方法的计算速度，将深度相机的视野范围和局部区域的交集作为局部障碍物，是无人车附近有效的障碍物信息。然后，在这些障碍物中搜索最大可通行宽度被建模成二次规划问题，具体形式如下：

$$\begin{aligned}
& \max_{b_2, b_1} \|b_2 - b_1\|^2, \\
s.t. \quad & \mathbf{a}^T \mathbf{y}_1 + b_1 > 0, b_1 > b, \\
& \mathbf{a}^T \mathbf{y}_2 + b_2 < 0, b_2 < b, \\
& |b_1 - b| < w_{mf} \|\mathbf{a}\|, \\
& |b_2 - b| < w_{mf} \|\mathbf{a}\|,
\end{aligned} \tag{3-5}$$

其中， $\mathbf{a} \in \mathbb{R}^{2 \times 1}$  和  $b \in \mathbb{R}$  是直线方程的系数，通过无人车当前位置  $\mathbf{p}_{ic}$  和局部目标点  $\mathbf{p}_{ig}$  计算。具体计算方式如下：

$$\begin{aligned}
\mathbf{a} &= \begin{bmatrix} \mathbf{p}_{ig}[2] - \mathbf{p}_{ic}[2] \\ \mathbf{p}_{ic}[1] - \mathbf{p}_{ig}[1] \end{bmatrix}, \\
b &= \mathbf{p}_{ig}[2]\mathbf{p}_{ic}[1] - \mathbf{p}_{ig}[1]\mathbf{p}_{ic}[2]
\end{aligned} \tag{3-6}$$

公式(3-5)中， $\mathbf{y}_1$  和  $\mathbf{y}_2$  分别表示局部障碍物二维位置的集合，对应图 3-9(a)中直线方程  $l: \{\mathbf{x}_i | \mathbf{a}^T \mathbf{x}_i + b = 0, \mathbf{x}_i \in \mathbb{R}^2\}$  两侧的局部障碍物。 $w_{mf}$  表示目前编队库中编队队形的最大宽度，如中的  $\mathcal{P}_1$ 。 $b_1$  和  $b_2$  是公式(3-5)的优化变量。公式(3-5)第一个约束将直线  $l_a: \{\mathbf{y} | \mathbf{a}^T \mathbf{y} + b_1 = 0, \mathbf{y} \in \mathbb{R}^2\}$  约束在图 3-9(a)的  $\Omega_1$  区域并且第二个约束将直线  $l_b: \{\mathbf{y} | \mathbf{a}^T \mathbf{y} + b_2 = 0, \mathbf{y} \in \mathbb{R}^2\}$  约束在图 3-9(a)的  $\Omega_2$  区域。目标函数是求解  $b_1$  与  $b_2$  之间最大的差值，等价于求解直线  $l_a$  和  $l_b$  之间的最大距离。由于预设编队库的存在最宽编队，在求解公式(3-5)的有约束优化问题时，利用第三个约束和第四个约束限制求解的宽度不超过  $2w_{mf}$ ，用于处理周围没有障碍物的情况，保证算法的鲁棒性。在图 3-9(b)展示一次优化的结果，令求解的结果为  $b_1^*$  和  $b_2^*$ ，最大可通行宽度表示为：

$$w_c = \frac{|b_1^* - b_2^*|}{\|\mathbf{a}\|} \tag{3-7}$$

图 3-9(b)中两条紫色直线表示求解的结果。为了改善局部目标点  $\mathbf{p}_{ig}$  的安全性，将两条紫色直线离局部目标点最近的两个端点组成的线段中心作为宽度检测的局部目标点，表示为  $\mathbf{p}_{ig,w}$ （如图 3-9(b)中红色目标点所示）。该宽度检测模块由单个无人车以中心式方式执行，随着无人车规模的扩大，其计算复杂度并不会显著增加。另一方面，随着环境复杂度的提升，单位面积内障碍物数量增多，在一定程度上使得宽度检测模块的计算时间有所上升。

### 3.4 编队形状选择

编队形状选择方法结合 FmRRT\* 和最大可通行宽度检测方法，以快速确定匹配环境约束的编队形状，并更新更安全的局部目标点  $\mathbf{p}_{ig,r}$ 。为提高宽度检测的稳定性，该方法采用多次迭代求解最大可通行宽  $w_c$ ，直到  $w_c$  趋于稳定后终止迭

代，并根据所计算的宽度选择最佳的编队模式，具体实现细节参考算法 5：

---

### 算法5 最优队形选择

---

```

输入:  $\mathbf{p}_{ig}$ ,  $\mathbf{p}_{ic}$ ,  $I_{max}$ ,  $\theta_{th}$ ,  $w_{th}$ 
输出:  $\mathbf{p}_{ig,r}$ ,  $\mathbf{P}_d^*$ ,  $w_c^*$ 

1: Init:  $I \leftarrow 0$ ,  $\theta_0 \leftarrow \arccos(\mathbf{p}_{ig} \cdot \mathbf{p}_{ic} / \|\mathbf{p}_{ig}\| \cdot \|\mathbf{p}_{ic}\|)$ ,  $\mathbf{p}_{ig,w} \leftarrow \mathbf{p}_{ig}$ ,  $w_c \leftarrow 2w_{th}$ 
2: While  $I < I_{max}$  and  $|\theta_{wd}| < \theta_{th}$  and  $|w_c - w_l| > w_{th}$ 
3: do
4:    $w_l \leftarrow w_c$ 
5:    $w_c$ ,  $\mathbf{p}_{ig,w} \leftarrow \text{WidthCheck}(\mathbf{p}_{ic}, \mathbf{p}_{ig,w})$  // 根据位置信息计算可通行宽度
6:    $\theta_i \leftarrow \arccos\left(\frac{\mathbf{p}_{ig,w} \cdot \mathbf{p}_{ic}}{\|\mathbf{p}_{ig,w}\| \|\mathbf{p}_{ic}\|}\right)$ 
7:    $\theta_{wd} \leftarrow \theta_i - \theta_0$ 
8:    $I = I + 1$ 
9: End while
10:  $\mathbf{p}_{ig,r} \leftarrow \text{RefineLocalGoal}(\mathbf{p}_{ig,w})$  // 更新局部目标点, 详见公式(3-8)
11:  $w_c^* \leftarrow \text{WidthCheck}(\mathbf{p}_{ic}, \mathbf{p}_{ig,r})$ 
12:  $\mathbf{P}_f^* \leftarrow \text{UpdateShape}(w_c^*)$  // 根据可通行宽度选择编队形状
13:  $\theta^* \leftarrow \arccos\left(\frac{\mathbf{p}_{ig,r} \cdot \mathbf{p}_{ic}}{\|\mathbf{p}_{ig,r}\| \|\mathbf{p}_{ic}\|}\right)$ 
14:  $\mathbf{R} \leftarrow \text{AngleToRotMat}(\theta^*, \mathbf{I}_z)$  // 将角度 $\theta^*$ 转换成旋转矩阵
15:  $\mathbf{P}_d^* \leftarrow \mathbf{R} \cdot \mathbf{P}_f^*$ 

```

---

输入包括局部目标点  $\mathbf{p}_{ig}$  和无人车当前位置  $\mathbf{p}_{ic}$ ，以及多个条件判断阈值：最大迭代次数  $I_{max}$ ，最大角度变化量  $\theta_{th}$  和最小宽度变化量  $w_{th}$ ，其中  $\mathbf{p}_{ig}$  是基于 FrmRRT\*生成的路径并结合无人车的感知范围进行确定。首先，初始化变量：迭代次数  $I$ ，局部目标点和无人车位置组成直线与 x 轴的夹角  $\theta_0$ ，每次宽度检测后更新的局部目标点  $\mathbf{p}_{ig,w}$ ，上次更新的宽度  $w_l$  和目前最佳的宽度  $w_c$ 。随后，通过多次迭代不断更新  $w_c$ 。迭代进行需要同时满足三个条件：迭代次数未超过设定阈值，每次更新局部目标点后不能偏离原先局部目标过远和宽度检测的变化量大于设定阈值，其中第二个条件通过限制更新后的目标点  $\mathbf{p}_{ig,w}$  保持在原本目标点  $\mathbf{p}_{ig}$  附近，以防止为了提高局部目标点的安全性而偏离设定的终点。在迭代中，**WidthCheck**( $\mathbf{p}_{ic}, \mathbf{p}_{ig,w}$ ) 会根据无人车位置和局部目标点，求解公式(3-5)以获得当前最佳宽度  $w_c$ ，并据此调整局部目标点  $\mathbf{p}_{ig,w}$  的位置。

迭代结束后，**RefineLocalGoal**( $\mathbf{p}_{ig,w}$ ) 结合 ESDF 地图的障碍物信息和当前局部目标点  $\mathbf{p}_{ig,w}$ ，计算距离障碍物更远，更安全的局部目标点  $\mathbf{p}_{ig,r}$ ，从而为无人车提供更大的操作空间。具体实现方法是求解无约束优化问题：

$$\mathbf{p}_{ig,r} = \underset{\mathbf{p}_{ig,r}}{\operatorname{argmin}} w_{r1} \max\{d_{th,r} - d_o(\mathbf{p}_{ig,w}), 0\}^2 + w_{r2} \|\mathbf{p}_{ig,r} - \mathbf{p}_{ig,w}\|^2 \quad (3-8)$$

其中  $w_{r1}$  和  $w_{r2}$  是目标函数中实现不同任务的权重。 $d_o(\mathbf{p}_{ig,w})$  是从 ESDF 地图中获取  $\mathbf{p}_{ig,w}$  与最近障碍物的距离，而  $d_{th,r}$  是相应的阈值。具体来说，当  $\mathbf{p}_{ig,w}$  与最近障碍物的距离大于等于  $d_{th,r}$ ，公式(3-8)中第一项  $\max\{d_{th,r} - d_o(\mathbf{p}_{ig,w}), 0\}^2$  始终为零，只有第二项  $\|\mathbf{p}_{ig,r} - \mathbf{p}_{ig,w}\|^2$  发挥作用，求解的结果为  $\mathbf{p}_{ig,r}$  等于  $\mathbf{p}_{ig,w}$ 。当  $\mathbf{p}_{ig,w}$  与最近障碍物的距离小于  $d_{th,r}$ ，公式(3-8)中第一项和第二项都发挥作用。根据不同的权重  $w_{r1}$  和  $w_{r2}$ ，有不同的倾向。当  $w_{r1} > w_{r2}$  时， $\mathbf{p}_{ig,r}$  与最近障碍物的距离更靠近阈值  $d_{th,r}$ 。反之， $\mathbf{p}_{ig,r}$  与  $\mathbf{p}_{ig,w}$  之间的距离更近。

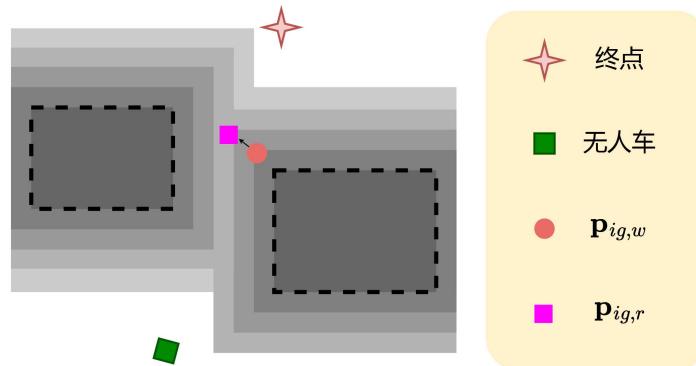


图 3-10 在 ESDF 地图上利用公式(3-8)调整局部目标点的示意图

Figure 3-10. Diagram of adjusting the local goal point on the ESDF map using Eq.(3-8)

通过公式(3-8)求解的  $\mathbf{p}_{ig,r}$  是在  $\mathbf{p}_{ig,w}$  附近且距离障碍物更远的局部目标点，其求解结果如图 3-10 所示。由于局部目标点发生变化，调用 **WidthCheck**( $\mathbf{p}_{ic}, \mathbf{p}_{ig,r}$ ) 计算最终的可通行宽度  $w_c^*$ 。**UpdateShape**( $w_c^*$ ) 从预设编队库中匹配小于可通行宽度  $w_c^*$  的最宽编队模式，其矩阵形式为  $\mathbf{P}_f^*$ 。为了确保编队能够沿着 FrmRRT\* 生成的路径运动，每次执行队形选择算法时，都会计算编队模式的角度：

$$\theta^* = \arccos\left(\frac{\mathbf{p}_{ig,r} \cdot \mathbf{p}_{ic}}{\|\mathbf{p}_{ig,r}\| \cdot \|\mathbf{p}_{ic}\|}\right) \quad (3-9)$$

以二维队形选择为例，**AngleToRotMat**( $\theta^*, \mathbf{I}_z$ ) 将绕 Z 轴旋转  $\theta^*$  的角轴表示转换为对应的旋转矩阵：

$$\mathbf{R} = \begin{bmatrix} \cos \theta^* & -\sin \theta^* & 0 \\ \sin \theta^* & \cos \theta^* & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-10)$$

最终适应环境的期望编队模式  $\mathcal{P}_d^*$  的矩形形式表示如下：

$$\mathbf{P}_d^* = \mathbf{R} \cdot \mathbf{P}_f^* \quad (3-11)$$

最后，队形选择算法输出期望的编队模式  $\mathcal{P}_d^*$ ，可通行宽度为  $w_c^*$  以及结合可通行宽度和障碍物信息调整后的局部目标点  $\mathbf{p}_{ig,r}$ ，其中  $\mathcal{P}_d^*$  和  $\mathbf{p}_{ig,r}$  将作为 4.4 节中

基于编队一致性的控制器输入，使得无人车快速收敛到期望队形。而  $w_c^*$  则通过无线路由器在无人车间共享，从而实现期望编队模式的动态扩展或收缩。

### 3.5 仿真实验分析

#### 3.5.1 实验设置

在本节首先将 FrmRRT\*与现有算法进行对比分析。然后，分析 FrmRRT\*算法中关键参数  $\rho_w$  对性能的影响。最后，针对 FrmRRT\*方法和 WidthCheck 方法进行消融实验。本节实验基于机器人操作系统（Robot Operating System, ROS）进行。以下是本节实验中使用的评估指标：

- (1) 路径长度：路径中所有线段长度的总和。
- (2) 最小安全距离：路径上路径点距离障碍物最近的距离。
- (3) 平均安全距离：路径上路径点的平均圆半径。

#### 3.5.2 性能对比

FrmRRT\*算法与 RRT\*[23]和 TubeRRT\*[22]算法在路径长度，最小安全距离和平均安全距离方面进行对比分析，以验证 FrmRRT\*能够生成接近期望编队宽度的路径点。所有的算法均用 C++实现并在一台 CPU 型号为 Intel i7-14650HX 和内存为 16G 的笔记本电脑上运行。测试环境的尺寸为  $50m \times 50m$ ，且障碍物均为随机生成。所有算法分别在障碍物规模为 40, 50, 60, 70 的测试环境中进行，如图 3-11 所示。每次实验的起点和终点相同。在公式(3-1)中，FrmRRT\*和 TubeRRT\*的参数设置在表 3-2，其中仅 FrmRRT\*需考虑编队宽度，因此需要设置参数  $w_{des}$ ，而 TubeRRT\*无需该参数。

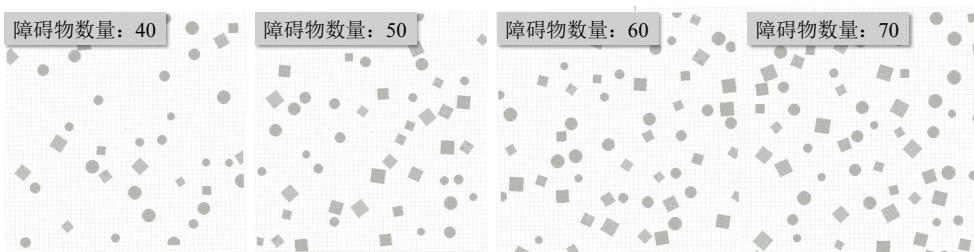


图 3-11 不同障碍物规模的测试环境

Figure 3-11. Test environments with different obstacle scales

表 3-2 不同方法对评分函数的参数设置

Table 3-2. Parameter settings of the scoring function for different methods

| 方法       | $\rho_d$ | $\rho_v$ | $\rho_w$ | $\varepsilon$ | $w_{des}$ |
|----------|----------|----------|----------|---------------|-----------|
| FrmRRT*  | 10       | 0.15     | 50       | 0.01          | 1.8       |
| TubeRRT* | 10       | 0.15     | 0        | 0.01          | -         |

所有算法在不同障碍物数量的环境下，其各项指标的评估结果分别展示在图 3-12、图 3-13 和图 3-14 中。具体而言，在图 3-12 和图 3-13 中，FrmRRT\*的路

径长度和最小安全距离均与 TubeRRT\*相近，但大于 RRT\*。其主要原因在于 RRT\* 仅优化路径长度，使其最短，因此往往会选择更靠近障碍物的路径点；而 TubeRRT\* 与 FrmRRT\* 在规划时不仅要最小化路径长度，还需最大化路径点的圆相交面积，使得路径长度相对较长，并使路径点距离障碍物更远。相比之下，FrmRRT\* 在 TubeRRT\* 基础上额外考虑了编队宽度  $w_{des}$ ，因此其最小安全距离平均而言更接近编队宽度。此外，随着障碍物的增加，所有算法的路径长度均呈现小幅度增长趋势，而最小安全距离则逐渐下降。

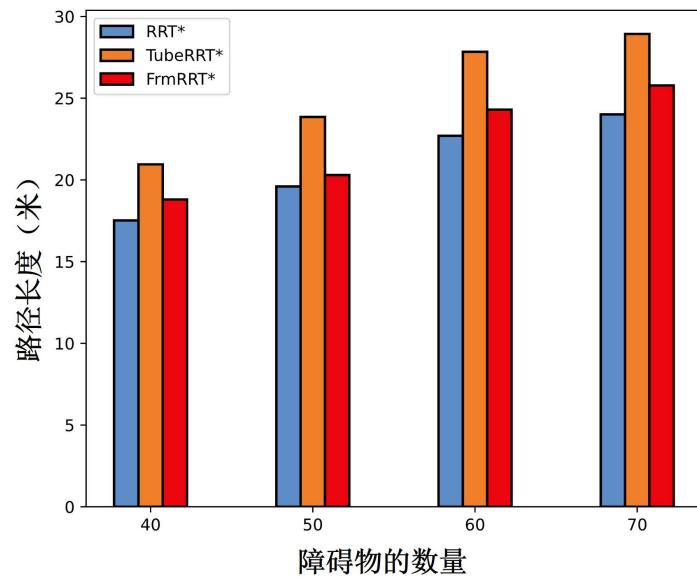


图 3-12 在不同障碍物数量的环境中对比不同算法的路径长度

Figure 3-12. Comparison of path lengths for algorithms in environments with different numbers of obstacles

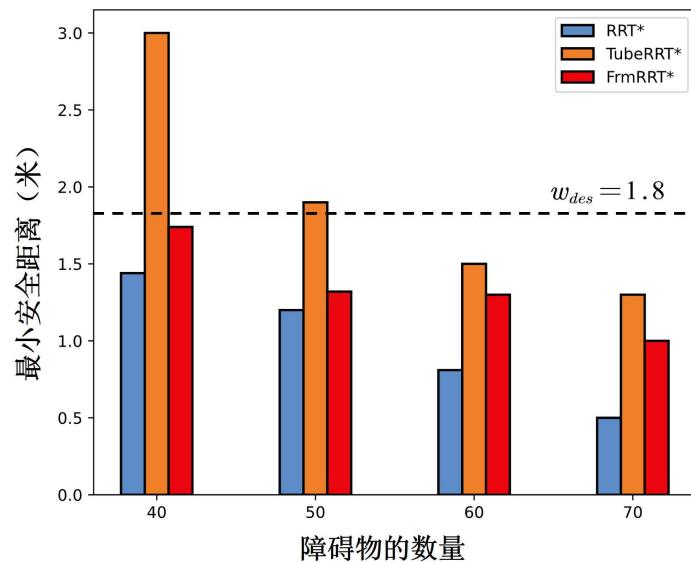


图 3-13 在不同障碍物数量的环境中对比不同算法的最小安全距离

Figure 3-13. Comparison of minimum safe distance for algorithms in environments with different numbers of obstacles

在图 3-14 中, 由于 RRT\*在规划时节点仅包含二维位置, 与 FrmRRT\*和 TubeRRT\*节点为二维圆形, 因此无法计算 RRT\*平均安全距离, 故仅统计了 FrmRRT\*和 TubeRRT\*的平均安全距离数据。在图中可以看出, 在编队宽度的约束下, FrmRRT\*的平均安全距离相较于 TubeRRT\*相比更接近  $w_{des}$ 。此外, 随着障碍物数量的增加, FrmRRT\*和 TubeRRT\*的平均安全距离均呈现下降趋势。

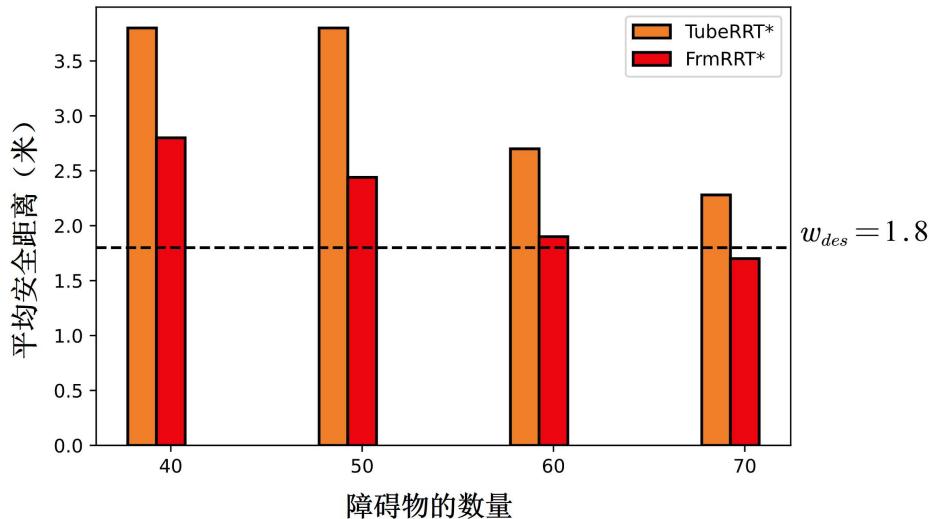


图 3-14 在不同障碍物数量的环境中对比不同算法的平均安全距离

Figure 3-14. Comparison of average safe distance for algorithms in environments with different numbers of obstacles

在图 3-15 中, 展示了不同路径规划算法在同一环境中求解的路径。该场景中包含随机生成的 50 个障碍物, 在图 3-15 仅展示了测试环境的一部分。不同算法的评估指标展示在表 3-3 中, 与上述描述基本一致。

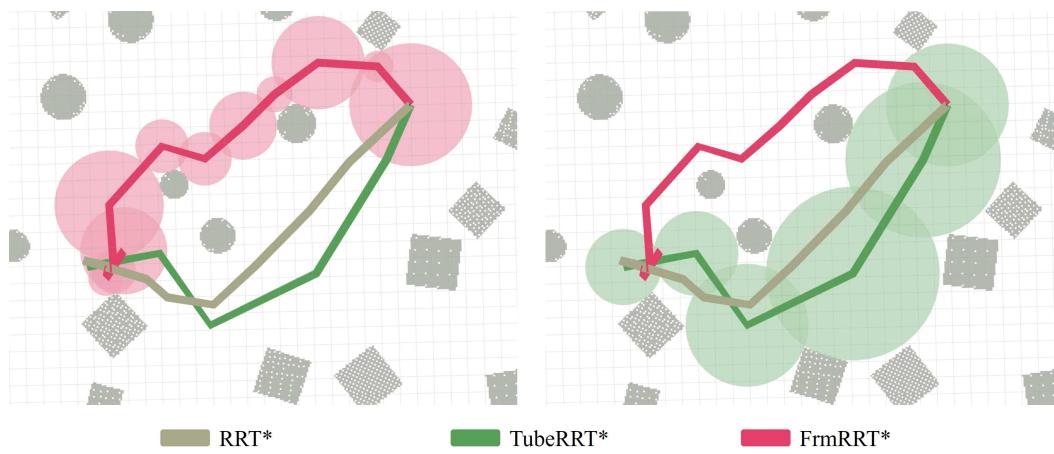


图 3-15 不同路径规划算法在有障碍物环境中的求解结果示意图

Figure 3-15. Diagram of the solutions obtained by different path planning algorithms in an obstacle environment

表 3-3 不同路径规划算法在图 3-15 的场景中的评估指标

**Table 3-3. Evaluation metrics of different path planning algorithms in the scenario of Figure 3-15**

| 方法       | 路径长度  | 最小安全距离 | 平均安全距离 |
|----------|-------|--------|--------|
| RRT*     | 26.39 | 1.2    | -      |
| TubeRRT* | 29.18 | 2.1    | 3.12   |
| FrmRRT*  | 30.46 | 1.5    | 1.82   |

总的来说, FrmRRT\*在 TubeRRT\*的基础上进行了改进, 在评估函数中引入了编队宽度约束, 使得 FrmRRT\*能够保持平均安全距离接近设定的编队宽度, 同时路径长度与 TubeRRT\*基本相当。这一改进有效降低了多无人车编队过程中队形变换的频率, 从而提升了编队运行的整体效率。

### 3.5.3 关键参数分析

在本小节将分析公式(3-1)中参数  $\rho_w$  对于 FrmRRT\*的性能的影响。测试环境使用图 3-11 中不同障碍物数量地图, 除了  $\rho_w$  以外其他参数均按照表 3-2 进行设置。在不同的环境中进行多次测试, 统计结果如图 3-16 和图 3-17 所示。从图中可以看出, 越高的  $\rho_w$  值能使得平均安全距离更接近预设的编队宽度  $w_{des}$ , 但同时导致路径长度相对更长。尽管平均安全距离更接近编队宽度有助于减少编队变换的频率, 降低与障碍物碰撞的风险, 但同时也会相应增加编队的运行时间。这表明, FrmRRT\*通过增加路径长度来换取编队运行的安全, 因此选择合适的  $\rho_w$  对于提高编队运行效率至关重要。随着障碍物数量的增加, 平均安全距离减低而路径长度增加。然而, 在一些障碍物多的场景中 (如图 3-16 所示), 较高的  $\rho_w$  能使平均距离更接近编队宽度。总体而言, FrmRRT\*算法能够有效引导多无人车编队在复杂环境中安全到达终点。

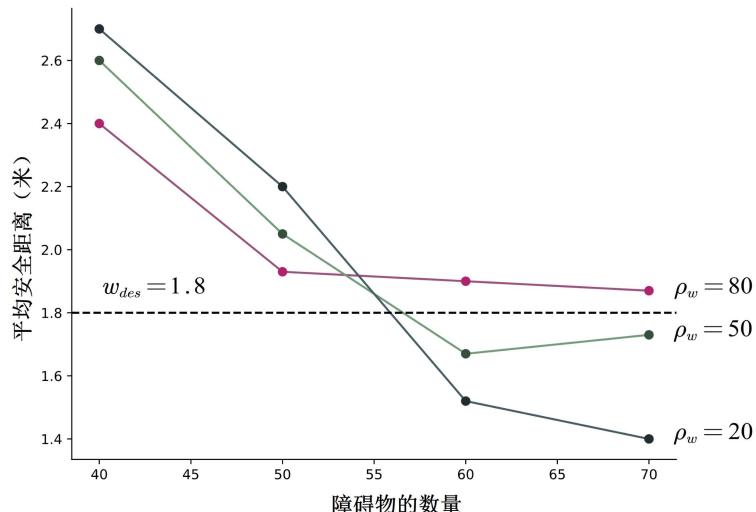


图 3-16 不同编队权重因子对平均安全距离的影响

**Figure 3-16. The impact of different formation weight factors on the average safety distance**

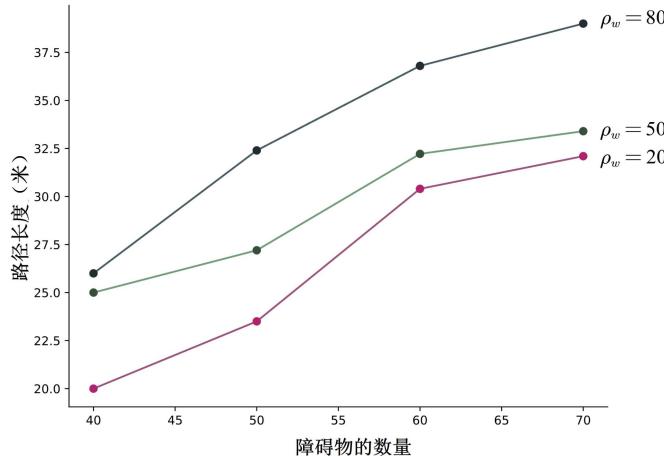


图 3-17 不同编队权重因子对路径长度的影响

Figure 3-17. The effect of different formation weighting factors on path length

### 3.5.4 消融实验

本小节将对 FrmRRT\* 和 WidthCheck 进行消融实验，以验证组合 FrmRRT\*+WidthCheck 在检测更准确的编队宽度并调整更安全的局部目标点方面的有效性。测试环境采用图 3-11 中障碍物数量为 50 的地图。在本次消融实验中，FrmRRT\* 通过公式(3-4)插值计算无障碍物范围  $w_{rrt}$  作为可行域宽度。此外，FrmRRT\* 上的局部目标点  $\mathbf{p}_{ig}$  和 WidthCheck 调整的局部目标点  $\mathbf{p}_{ig,w}$  与障碍物的距离均作为目标点安全距离，统计结果如表 3-4 所示。从表 3-4 可以看出，FrmRRT\* 方法的可行域宽度最小，这是因为其计算路径时，为了在相交圆中找到更安全可行域宽度，牺牲了部分无障碍物区域。而 RRT\*+WidthCheck 方法基于 RRT\* 进行规划路径，由于仅优化路径长度，导致路径点距离障碍物较近，因此其目标点安全距离最小（如图 3-18 所示）。相比之下，FrmRRT\*+WidthCheck 方法在 FrmRRT\* 规划路径时已充分考虑路径点与障碍物的距离，在此基础上利用 WidthCheck 方法进一步调整局部目标点，因此能够获得更大的目标点安全距离。如图 3-19 所示，FrmRRT\*+Width 调整后的局部目标点  $\mathbf{p}_{ig,w}$ （紫色点）相比原始局部点（灰色点）更加安全。此外，FrmRRT\*+WidthCheck 在 FrmRRT\* 规划的路径基础上进一步计算可行域宽度，能够找到比 RRT\*+WidthCheck 方法更大的可行域宽度。综上所述，FrmRRT\*+WidthCheck 方法充分利用了 FrmRRT\* 的路径信息，并基于此计算更准确的可行域宽度，可为队形选择提供基础。

表 3-4 FrmRRT\* 和 WidthCheck 消融实验的评估指标

Table 3-4. Evaluation metrics for FrmRRT\* and WidthCheck ablation study

| 方法                 | 可行域宽度 | 目标点安全距离 |
|--------------------|-------|---------|
| FrmRRT*            | 1.63  | 2.01    |
| RRT*+WidthCheck    | 2.21  | 1.19    |
| FrmRRT*+WidthCheck | 3.60  | 2.11    |

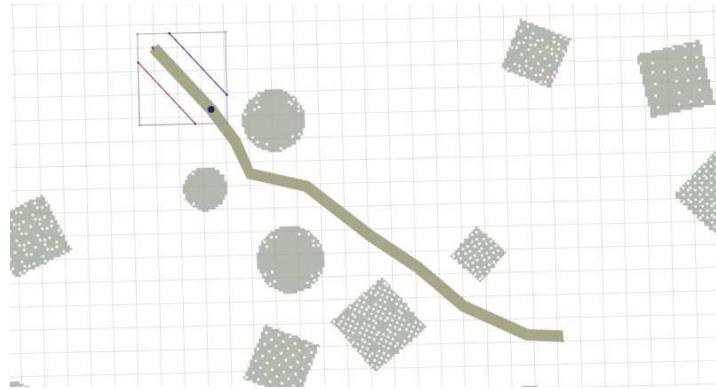


图 3-18 以 RRT\*+WidthCheck 的方式计算可行域宽度的示意图

Figure 3-18. Diagram illustrating the calculation of feasible region width using RRT\*+WidthCheck

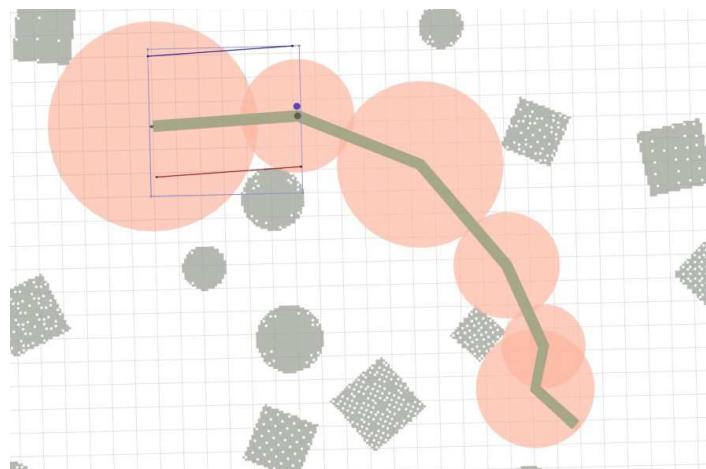


图 3-19 以 FrmRRT\*+WidthCheck 的方式计算可行域宽度的示意图

Figure 3-19. Diagram illustrating the calculation of feasible region width using FrmRRT\*+WidthCheck

### 3.6 本章小结

本章提出了一种基于可行区域的队形选择方法，结合编队级路径规划与最大可通行宽度检测，以优化无人车编队运行的路径选择和安全性。基于 RRT\*改进的编队路径规划算法考虑编队宽度来提升安全性。实验表明，适当增加编队相关权重，可以在一定程度控制路径点与障碍物的距离。然后，本章提出了最大可通行宽度检测算法。消融实验表明，该算法能够比编队级路径规划更准确地计算无障碍物宽度，并优化局部目标点。

## 第四章 基于编队一致性的队形控制

### 4.1 引言

在上一章中，无人车根据自身运动方向和周围障碍物信息计算最合适的编队模式，和更安全的局部目标点。本章根据上一章输出的信息，进一步提出一种基于编队一致性的局部控制器，以确保无人车能够快速收敛至匹配环境的期望队形。首先，根据期望队形，采用任务分配策略重新调整无人车在编队模式中的相对位置，以最小化编队变换所需的总路程。随后，设计基于编队一致性的局部控制器，控制无人车快速收敛至目标编队模式。为了加速编队信息在无人车之间达成一致性，本文采用中心化方法进行队形选择与任务分配，并设计领导无人车重选择策略，选定最合适的领导无人车。最后，在极端环境中，当无人车与障碍物距离过近时，为确保安全性，切换控制器的参考输入，使无人车能够迅速远离障碍物。

本章的结构安排如下：首先，介绍基于最优运输的任务分配方法（详见 4.2 节）。其次，阐述领导无人车重选择策略的作用及必要性（详见 4.3 节）。接着，给出了基于编队一致性的控制器实现（详见 4.4 节）。然后，说明基于路径的局部避障策略的作用及实现（详见 4.5 节）。随后，对比不同编队控制方法的编队收敛效率，并分析不同数量无人车的编队收敛性能（详见 4.6 节）。最后，对队形控制算法及实验结果进行了总结（详见 4.7 节）。

### 4.2 基于最优运输的任务分配

在受限环境中实现编队导航时，无人车需要频繁地调整队形以适应环境变化，因此在队形变换前进行任务分配，合理调整无人车在新队形中的相对位置，可最大限度减少因任务切换所引发的整体路径长度增量，从而加快编队收敛。本节采用最优运输 (Optimal Transport, OT) 的离散形式来实现任务分配，其中无人车的当前位置与目标编队模式分别建模为两个离散分布，并计算其最优分配策略。具体而言，任务分配可建模为以下优化问题：

$$\begin{aligned} \boldsymbol{\gamma}^* = \operatorname{argmin}_{\boldsymbol{\gamma}} & \sum_{i,j} \boldsymbol{\gamma}_{i,j} \mathbf{M}_{i,j}, \\ \text{s.t. } & \boldsymbol{\gamma} \mathbf{1}^N = \mathbf{1}^N; \boldsymbol{\gamma}^T \mathbf{1}^N = \mathbf{1}^N, \end{aligned} \quad (4-1)$$

其中优化变量  $\boldsymbol{\gamma} \in \mathbb{R}^{N \times N}$  是  $N$  行  $N$  列的矩阵，求解的结果  $\boldsymbol{\gamma}^*$  是只包含 0 和 1 的矩阵， $\boldsymbol{\gamma}_{i,j}^* = 1$  表示将第  $i$  辆无人车分配给新编队模式的第  $j$  个位置。 $\mathbf{1}^N \in \mathbb{R}^N$  是  $N$  个元素全为 1 的列向量。 $\mathbf{M} \in \mathbb{R}^{N \times N}$  是代价矩阵，其第  $i$  行第  $j$  列的元素  $\mathbf{M}_{i,j}$

表示第  $i$  辆无人车到新编队模式的第  $j$  个位置的代价，本文取其为欧式距离的平方：

$$M_{i,j} = \| \mathbf{p}_{ie} - \mathbf{p}_{f,j} \|^2 \quad (4-2)$$

公式(4-1)所描述的优化问题可采用专门求解最优运输问题的开源库 POT (Python Optimal Transport) [77]求解。文献[77]指出，POT 是一种高效的最优运输 (OT) 求解器，能求解公式(4-1)所描述的整数规划问题。该方法基于熵正则化技术，将离散最优运输问题转化为凸优化问题，并通过 Sinkhorn 算法实现快速求解。其起算杂度为  $O(N^2)$ ，适用于中等规模编队 ( $N \leq 15$ )。图 4-1 中展现了基于最优运输的任务分配结果。

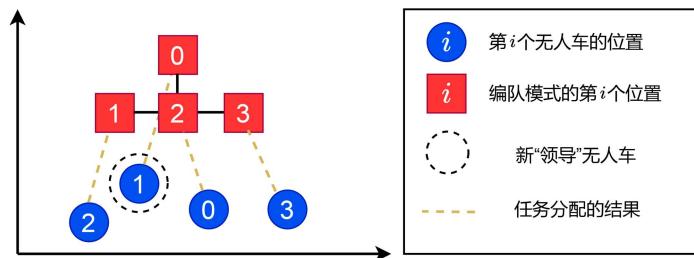


图 4-1 基于最优运输的任务分配和领导无人车重选择策略示意图

Figure 4-1. Diagram of task allocation based on optimal transport and the leader robot reselection strategy

### 4.3 领导无人车重选择策略

在编队控制中，“领导”无人车负责确定符合当前环境信息的最佳编队形状，并重新分配无人车与编队形状的对应关系，以确保所有无人车能够快速达成编队信息的一致性。然而，与传统固定的“领导”机器人不同，本文的“领导”无人车是动态调整的。在每次任务分配后，当前“领导”无人车会触发领导无人车重选择策略，重新选择新的“领导”无人车。如图 4-1 所示，当无障碍物宽度小于当前队形（如图中的蓝色圆圈形成的队形）时，“领导”无人车执行基于可行域的队形选择方法（详见第三章），从预设的编队库中选取当前最佳的编队模式（如图中的红色方块形成的队形）。随后，在图 4-1 中，基于最优运输的任务分配将无人车分配至新编队模式的相应位置。为了方便描述，每个编队模式中的位置均设有编号。当某个无人车被分配至编队模式的 0 号位置时，这辆无人车将选定为新的“领导”无人车。如图 4-1 所示，原本的 1 号无人车被选定为新“领导”无人车，而先前的“领导”无人车则被分配到新编队模式的 2 号位置。此外，编队库中的每个编队模式都会将最有利于观察障碍物信息的位置设置为 0 号（如图 4-1 所示）。这一策略使“领导”无人车能够充分利用环境信息进行集中式的决策，加快多无人车编队信息的一致性。该策略的计算时间不会随着无人车规模增加和环境复杂度发生显著变化。

#### 4.4 基于编队一致性的控制器

为了使无人车能够快速的收敛到期望的编队模式，本节基于 MPC 设计了一种基于编队一致性的控制器。在使用模型预测控制对无人车进行控制之前，先设计控制器的参考状态  $\mathbf{x}_{ir}$  和参考输入  $\mathbf{u}_{ir}$ 。基于位置一致性原则，控制器的参考位置  $\mathbf{x}_{ir}$  计算方法描述如下：

$$\mathbf{x}_{ir} = \frac{1}{N_i + 1} (\mathbf{x}_{formation} + \mu \mathbf{x}_{navigation}) \quad (4-3)$$

其中包含编队位置项  $\mathbf{x}_{formation}$  和导航位置项  $\mathbf{x}_{navigation}$ ；  $N_i$  表示属于集合  $\mathcal{N}_i$  的无人车的数量。编队位置项结合期望编队模式的位置与无人车当前位置，以计算参考位置，具体定义如下：

$$\mathbf{x}_{formation} = \sum_{j \in \mathcal{N}_i} (\mathbf{Bx}_j + \mathbf{p}_{f,ij}) \quad (4-4)$$

其中  $\mathcal{N}_i$  表示除第  $i$  辆无人车以外的无人车的集合；  $\mathbf{Bx}_j$  表示集合  $\mathcal{N}_i$  中其他无人车当前的位置。  $\mathbf{p}_{f,ij} = \mathbf{p}_{f,i} - \mathbf{p}_{f,j}$  表示编队模式中第  $i$  个位置  $\mathbf{p}_{f,i}$  相对于第  $j$  个位置  $\mathbf{p}_{f,j}$  的位置。导航位置项用于引导无人车到达终点，具体定义如下：

$$\mathbf{x}_{navigation} = \mathbf{p}_{ig,r} \quad (4-5)$$

其中  $\mathbf{p}_{ig,r}$  表示局部目标点。公式(4-3)中  $\mu$  是一个常数，用于调节编队收敛的速度。当  $\mu$  大于 1 时，表明控制器参考位置  $\mathbf{x}_{ir}$  中导航位置项占比更大，促使无人车快速到达终点。与之相反，当  $\mu$  小于 1 时，表明控制器的参考位置  $\mathbf{x}_{ir}$  中编队位置项占比更大，促使无人车能更快地收敛到设定的编队模式并保持。

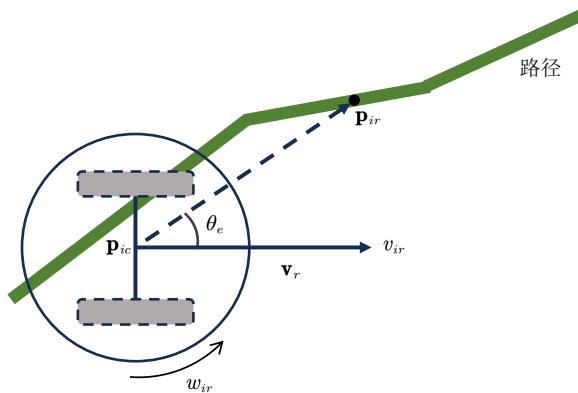


图 4-2 无人车的模型图

Figure 4-2. Model diagram of the unmanned vehicle

控制器的参考输入  $\mathbf{u}_{ir} = [v_{ir}, w_{ir}]^T$  设计如下：

$$v_{ir} = \begin{cases} \frac{\|\mathbf{p}_{ir} - \mathbf{p}_{ic}\|}{x_{th}} \cdot v_{max}, & \|\mathbf{p}_{ir} - \mathbf{p}_{ic}\| \leq x_{th}, \\ v_{max}, & \|\mathbf{p}_{ir} - \mathbf{p}_{ic}\| \geq x_{th} \end{cases}, \quad (4-6)$$

$$w_{ir} = 1/\pi \cdot w_{max} \cdot \theta_e \quad (4-7)$$

其中  $v_{max}$  和  $w_{max}$  分别表示无人车最大的线速度和角速度；  $x_{th}$  表示线速度控制阈值；  $v_{ir}$  和  $w_{ir}$  分别表示无人车的角速度和线速度。如图 4-2 所示，  $\mathbf{p}_{ir} = \mathbf{B}\mathbf{x}_{ir}$  表示第  $i$  辆无人车的参考位置；  $\mathbf{p}_{ic} = \mathbf{B}\mathbf{x}_{ic}$  表示第  $i$  辆无人车的位置，  $\mathbf{v}_r \in \mathbb{R}^3$  表示当前无人车朝向的单位方向向量，  $\theta_e$  表示无人车朝向与参考位置  $\mathbf{p}_{ir}$  相差的角度，具体定义如下：

$$\theta_e = \arccos\left(\frac{\mathbf{v}_r \cdot (\mathbf{p}_{ir} - \mathbf{p}_{ic})}{\|\mathbf{v}_r\| \cdot \|\mathbf{p}_{ir} - \mathbf{p}_{ic}\|}\right) \quad (4-8)$$

公式(4-6)中，参考线速度  $v_{ir}$  可以根据无人车和参考位置的距离  $\|\mathbf{p}_{ir} - \mathbf{p}_{ic}\|$  调节大小。当  $\|\mathbf{p}_{ir} - \mathbf{p}_{ic}\|$  小于阈值  $x_{th}$  时，  $v_{ir}$  小于  $v_{max}$  并与  $v_{max}$  成正比。反之，  $v_{ir}$  等于最大线速度。其设计可以控制无人车的线速度平稳变化。公式(4-7)中，参考角速度  $w_{ir}$  根据角度  $\theta_e$  调节大小和方向，而  $1/\pi \cdot w_{max}$  系数的目的是将  $w_{ir}$  的取值范围限制在  $[-\pi, \pi]$ 。当  $\theta_e$  超过  $\pi$  时，这样的设计使得参考角速度的值接近最大角速度，控制小车快速旋转，减少角度差值  $\theta_e$ 。这样可以有效防止 MPC 控制器求解的控制量  $\mathbf{u}$  的线速度是负，从而避免无人车后退，这对搭载视野有限的相机的非完整约束的无人车来说极为危险。

模型预测控制根据运动学方程和初始状态预测未来一段时间的状态，并通过构建有约束的优化问题，使求解结果可以控制无人车以最优的方式接近参考位置。在预测时间内，第  $i$  辆无人车在  $t_k$  时刻的预测状态为  $\mathbf{x}_i(t_{k+m|k}) (m=0, \dots, M-1)$ ，而预测输入为  $\mathbf{u}_i(t_{k+m|k})$ ，其中  $M$  表示预测未来状态的单位时间数量。基于分布式模型预测控制框架的非线性约束优化问题描述如下：

$$\begin{aligned} \min_{v_i, w_i, \varepsilon} \quad & \sum_{m=0}^{M-1} \|\mathbf{x}_i(t_{k+m|k}) - \mathbf{x}_{ir}\|_{\mathbf{Q}} + \|\mathbf{u}_i(t_{k+m|k}) - \mathbf{u}_{ir}\|_{\mathbf{R}} + w_{\varepsilon} \varepsilon^2 \\ \text{s.t.} \quad & \mathbf{x}_i(t_{k+m+1|k}) = f(\mathbf{x}_i(t_{k+m|k}), \mathbf{u}_i(t_{k+m|k})), \\ & 0 \leq v_i(t_{k+m}) \leq v_{max}, \\ & |w_i(t_{k+m|k})| \leq w_{max}, \\ & \varepsilon_{th} \leq \varepsilon \leq 0, \\ & d_{ij}(t_{k+m|k}) \geq d_{safe} + \varepsilon, \quad \forall j \in \mathcal{N}_i, \end{aligned} \quad (4-9)$$

其中矩阵  $\mathbf{Q} \in \mathbb{R}_{+}^{3 \times 3}$  表示目标函数中状态误差惩罚权重矩阵和  $\mathbf{R} \in \mathbb{R}_{+}^{2 \times 2}$  表示目标函数中控制输入误差惩罚权重矩阵。 $\varepsilon$  是较小的负数，用于限制最后一个约束，近似的将最后一个约束从硬约束转换为软约束，提高求解的成功率。 $d_{safe}$  表示无人车间预设的安全距离，一般赋值为  $2.5r$ 。 $d_{ij}(t_{k+m|k})$  是当前无人车（第  $i$  辆）的预测位置和其他无人车（第  $j$  辆）之间的距离，具体定义如下：

$$d_{ij}(t_{k+m|k}) = \|\mathbf{x}_i(t_{k+m|k}) - \mathbf{x}_j\|_{\mathbf{B}} \quad (4-10)$$

在公式(4-9)中，目标函数旨在最小化预测状态和参考状态之间的差距，预测输入与参考输入的偏差以及 $\varepsilon$ 的值，从而保证无人车能快速的到达参考状态，并使无人车之间的距离大于安全距离 $d_{\text{safe}}$ 。公式(4-9)中的第一个约束是无人车的运动学模型(2-1)的离散形式，具体离散化状态方程描述如下：

$$\mathbf{x}_i(t_{k+m+1|k}) = \begin{bmatrix} x_i(t_{k+m+1|k}) \\ y_i(t_{k+m+1|k}) \\ \theta_i(t_{k+m+1|k}) \end{bmatrix} = \begin{bmatrix} x_i(t_{k+m|k}) + v_i(t_{k+m|k}) \cos \theta_i(t_{k+m|k}) \cdot \Delta t_c \\ y_i(t_{k+m|k}) + v_i(t_{k+m|k}) \sin \theta_i(t_{k+m|k}) \cdot \Delta t_c \\ \theta_i(t_{k+m|k}) + \omega_i(t_{k+m|k}) \cdot \Delta t_c \end{bmatrix} \quad (4-11)$$

其中 $\Delta t_c$ 表示采样的时间间隔。公式(4-9)中的第二，三和四约束分别限制了线速度，角速度以及 $\varepsilon$ 的取值范围，其中线速度的取值范围大于0，以确保无人车不会后退，从而避免与后方未被观测到的障碍物发生碰撞。公式(4-9)中的最后一个约束保证多无人车之间保持一定距离，相互之间不会发生碰撞。

## 4.5 基于路径的局部避障策略

无论是编队级路径规划还是最大可通行宽度检测算法，均利用环境信息引导无人车在更安全的区域进行编队控制。然而，在一些极端的环境下，无人车仍然可能与障碍物发生碰撞。多无人车编队导航系统中预设两种模式：编队模式和避障模式。当无人车与障碍物的距离大于阈值时，说明无人车很安全，执行编队模式；而当无人车与障碍物的距离小于阈值时，说明无人车有碰撞的风险，执行避障模式。与障碍物相关的距离阈值用符号 $d_{th,o}$ 表示。本节将介绍避障模式下的控制器设计。

当无人车处于避障模式时，将执行编队级路径规划，以生成一条从当前位置通往终点的路径，确保避障的同时不会偏离终点。然后，根据机器人的感知范围在路径上选取局部避障目标点 $\mathbf{p}_{ig,o}$ ，来引导无人车远离障碍物。接下来，避障模式的控制部分将在原先 MPC 控制器的优化问题(4-9)中添加障碍物约束：

$$\|\mathbf{B}\mathbf{x}_i(t_{k+m|k}) - \mathbf{p}_{obs}\| \geq 0.5d_{\text{safe}}, \forall \mathbf{p}_{obs} \in \mathcal{S}_i \quad (4-12)$$

其中集合 $\mathcal{S}_i$ 表示距离第*i*辆无人车最近且位于半径 $d_{th,o}$ 范围内的 $N_o$ 个障碍物的位置集合； $\mathbf{p}_{obs} \in \mathbb{R}^{2 \times 1}$ 表示集合 $\mathcal{S}_i$ 中障碍物位置。

为了提高 MPC 控制器的求解速度，将添加约束约束(4-12)的优化问题(4-9)的预测步长 $M$ 缩小到原来的十分之一，并将最后一个约束进行简化，其中 $\mathcal{N}_i$ 表示距离当前无人车最近的三辆其他无人车。

在控制无人车时，为了减少无人车进一步靠近障碍物的可能性，本文采取相对保守的方式控制无人车靠近 $\mathbf{p}_{ig,o}$ 。开始先将 MPC 的参考状态 $\mathbf{x}_{ir}$ 设置为 $[\mathbf{p}_{ic}, 0]^T$ ，参考输入设置 $\mathbf{u}_{ir}$ 设置为 $[0, \theta_{oe}]^T$ ，其中 $\theta_{oe}$ 的定义如下：

$$\theta_{oe} = \arccos\left(\frac{\mathbf{v}_r \cdot (\mathbf{p}_{ig,o} - \mathbf{p}_{ic})}{\|\mathbf{v}_r\| \cdot \|\mathbf{p}_{ig,o} - \mathbf{p}_{ic}\|}\right) \quad (4-13)$$

控制无人车原地旋转至朝向避障目标点  $\mathbf{p}_{ig,o}$ 。然后，将 MPC 的参考状态设置为  $[\mathbf{p}_{ig,o}, 0]^T$ ，参看输入设置为  $[v_{ir}, w_{ir}]^T$ ，控制小车快速到达避障局部目标点，远离障碍物。最后，当无人车与障碍物的距离大于  $d_{th,o} + r$  时，就会从避障模式切换到编队模式。

## 4.6 仿真实验分析

本节仿真实验首先将所提方法与另外两种编队控制方法在编队收敛效率方面进行比较，随后分析不同无人车数量对编队收敛效率的影响。实验场景利用 Gazebo 仿真软件搭建，包含四辆 Turtlebot3 轮式机器人。为量化编队控制性能，需计算当前编队模式  $\mathcal{P}_c$  和期望编队模式  $\mathcal{P}_d$  之间的编队误差。受到 Quan 的方法<sup>[74]</sup>和 Park 的方法<sup>[64]</sup>的启发，采用两种评估编队误差的方式。第一种评估方法是 Park 的工作中使用的平均编队距离误差  $\bar{e}_{dist}$ ：

$$\mathbf{x}_{id} = \frac{1}{N_i} \sum_{j \in \mathcal{N}_i} (\mathbf{Bx}_j + \mathbf{p}_{f,ij}) \quad (4-14)$$

其中公式(4-14)与公式(4-4)相同，仅基于期望编队模式和其他无人车的位置计算期望编队位置  $\mathbf{x}_{id}$ 。

$$\bar{e}_{dist} = \frac{1}{T_{sum}} \sum_{d=1}^D \sum_{i \in I} \|\mathbf{x}_{id}(t_d) - \mathbf{x}_{ic}(t_d)\|_{\mathbf{B}} \quad (4-15)$$

其中  $t_d$  表示记录无人车状态的时间点； $D$  为等间隔采样的时间点数量； $\Delta t = t_{d+1} - t_d$  为采样间隔；矩阵  $\mathbf{B}$  用于将无人车的状态转换为位置；完成任务的总时间为：

$$T_{sum} = \sum_{d=1}^D \Delta t \quad (4-16)$$

公式(4-15)先计算无人车期望位置和当前位置的偏差，然后对所有无人车的偏差求和取平均。同时， $\bar{e}_{dist}$  评估的编队误差是具有平移不变性，如果当前的编队模式仅通过平移即可与期望的编队模式完全重合，则此时的编队误差为 0。

第二种评估方法是基于图论，并在 Quan 的方法中使用。具体来说，每个编队模式都有一个相关的图  $\mathcal{G}$ ，每条边的权重  $w_{ij}$ ，邻边矩阵  $\mathbf{A}$  和度矩阵  $\mathbf{D}$ 。平均编队相似性误差  $\bar{e}_{sim}$  评估方法，定义如下：

$$\bar{e}_{sim} = \frac{1}{T_{sum}} \sum_{d=1}^D \|\hat{\mathbf{L}}(t_d) - \hat{\mathbf{L}}_{des}(t_d)\|_F^2, \quad (4-17)$$

其中  $\|\cdot\|_F$  表示弗罗贝尼乌斯范数 (Frobenius Norm);  $\hat{\mathbf{L}}(t_d)$  表示在  $t_d$  时刻当前编队形状的对称归一化拉普拉斯矩阵;  $\hat{\mathbf{L}}_{des}(t_d)$  表示期望编队形状的对称归一化拉普拉斯矩阵。 $\bar{e}_{sim}$  通过距离来衡量无人车之间的编队关系, 因此其同时具备旋转和平移不变性。此外, 由于  $\bar{e}_{sim}$  采用归一化的图拉普拉斯矩阵, 其计算结果具备尺度不变性。如果当前无人车构成的形状可通过相似性变换与期望的编队模式匹配时,  $\bar{e}_{sim}$  的值为 0。

#### 4.6.1 编队收敛效率对比分析

为了评估不同编队控制方法在编队收敛效率上的表现, 设计了十二组不同的四辆无人车编队变换实验, 包括从编队模式  $\mathcal{P}_1$  到编队模式  $\mathcal{P}_2$ ;  $\mathcal{P}_1$  到  $\mathcal{P}_3$ ;  $\mathcal{P}_1$  到  $\mathcal{P}_4$ ;  $\mathcal{P}_2$  到  $\mathcal{P}_1$ ;  $\mathcal{P}_2$  到  $\mathcal{P}_3$ ;  $\mathcal{P}_2$  到  $\mathcal{P}_4$ ;  $\mathcal{P}_3$  到  $\mathcal{P}_1$ ;  $\mathcal{P}_3$  到  $\mathcal{P}_2$ ;  $\mathcal{P}_3$  到  $\mathcal{P}_4$ ;  $\mathcal{P}_4$  到  $\mathcal{P}_1$ ;  $\mathcal{P}_4$  到  $\mathcal{P}_2$ ;  $\mathcal{P}_4$  到  $\mathcal{P}_3$ , 编队库如所示, 将所有编队变换组合都进行测试。在无障碍的环境中, 当编队误差  $\bar{e}_{dist}$  或  $\bar{e}_{sim}$  低于设定的参考阈值时, 认为无人车已收敛到期望的编队, 完成一次测试。为了减低误差的干扰, 每次测试会重复做五轮。表 4-1 和表 4-2 展示了三种方法成功收敛到预设编队模式所消耗的平均时间。在图 4-3 中展示了 12 种编队收敛过程中编队误差的变化趋势。具体来说, 由于 Quan 的方法基于图论, 无人车形成的编队具有旋转, 平移和尺度不变性, 因此  $\bar{e}_{dist}$  无法正确的判断编队是否收敛到期望队形。为了克服这一问题, 所有方法均采用  $\bar{e}_{sim}$  展示编队误差的变化趋势。此外, 由于 Quan 的方法的编队描述方法具有旋转不变性, 所以在表 4-1 和表 4-2 中, 从编队模式  $\mathcal{P}_1$  到  $\mathcal{P}_4$  和从编队模式  $\mathcal{P}_4$  到  $\mathcal{P}_1$  的测试消耗时间为 0。这是因为编队模式  $\mathcal{P}_1$  和编队模式  $\mathcal{P}_4$  只需旋转 90 度就能重合。在 Quan 方法的编队描述下,  $\mathcal{P}_1$  和  $\mathcal{P}_4$  表示同一种形状。

表 4-1 前六种编队模式变换消耗时间的对比

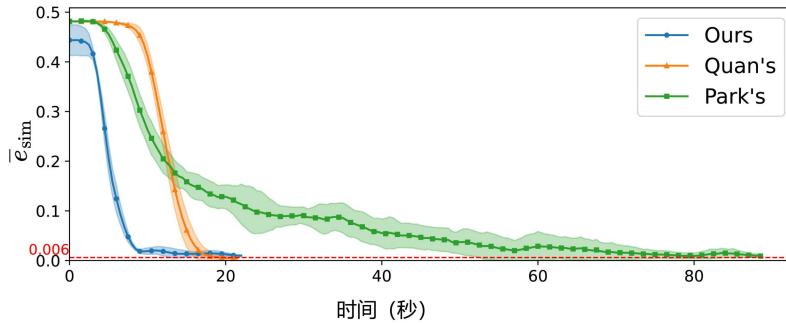
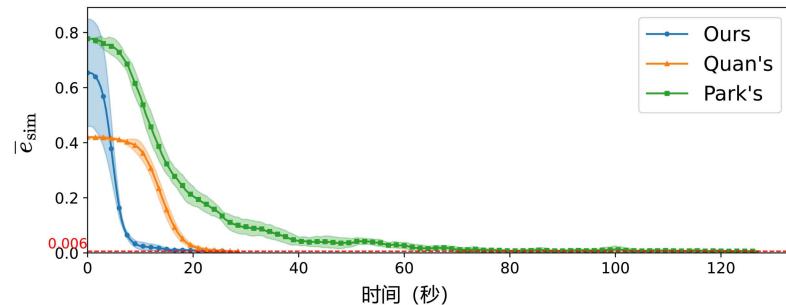
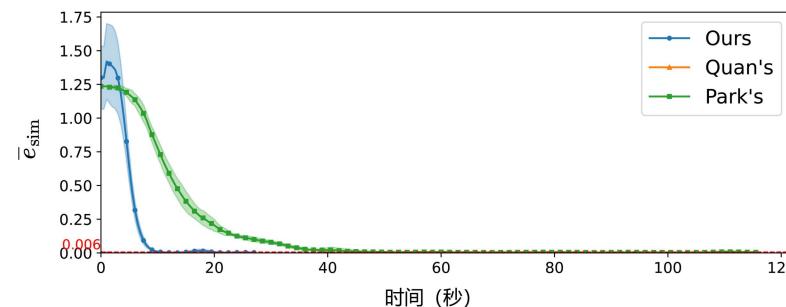
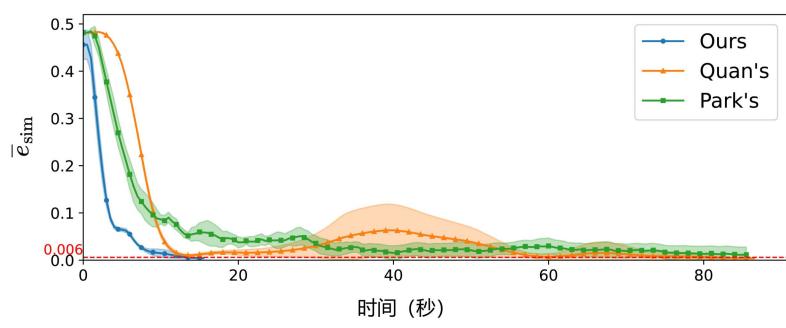
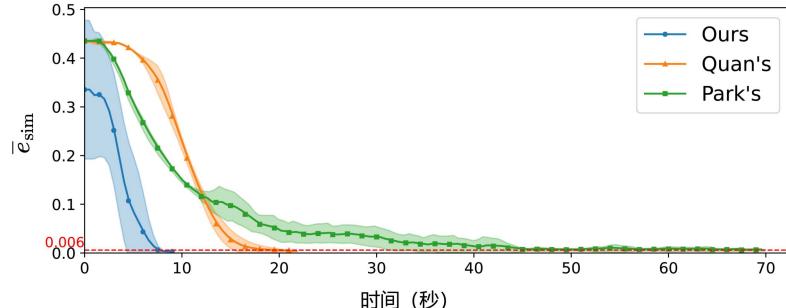
Table 4-1. Comparison of time consumption for the first six formation pattern transitions

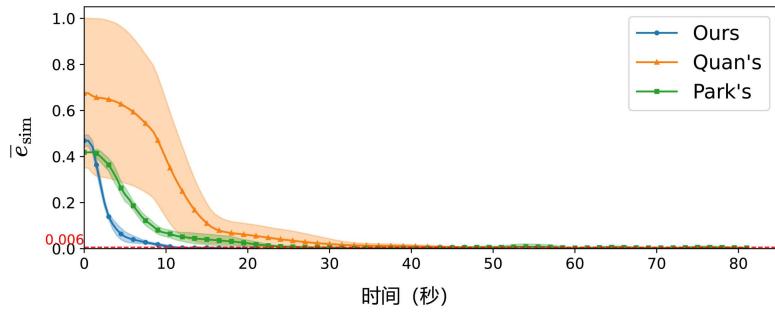
| 方法          | $\mathcal{P}_1 \rightarrow \mathcal{P}_2$ | $\mathcal{P}_1 \rightarrow \mathcal{P}_3$ | $\mathcal{P}_1 \rightarrow \mathcal{P}_4$ | $\mathcal{P}_2 \rightarrow \mathcal{P}_1$ | $\mathcal{P}_2 \rightarrow \mathcal{P}_3$ | $\mathcal{P}_2 \rightarrow \mathcal{P}_4$ |
|-------------|---|---|---|---|---|---|
| Quan        | 39s                                       | 42.3s                                     | -   | 16.3s                                     | 47.1s                                     | 20.5s                                     |
| Park        | 72.6s                                     | 68.2s                                     | 69.3s                                     | 62.8s                                     | 59.9s                                     | 76.3s                                     |
| <b>Ours</b> | <b>12.5s</b>                              | <b>19.2s</b>                              | <b>18.2s</b>                              | <b>8.9s</b>                               | <b>12.6s</b>                              | <b>16.4s</b>                              |

表 4-2 后六种编队模式变换消耗时间的对比

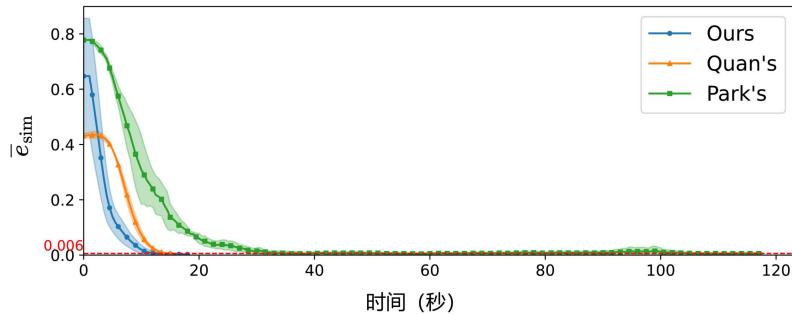
Table 4-2. Comparison of time consumption for the last six formation pattern transitions

| 方法          | $\mathcal{P}_3 \rightarrow \mathcal{P}_1$ | $\mathcal{P}_3 \rightarrow \mathcal{P}_2$ | $\mathcal{P}_3 \rightarrow \mathcal{P}_4$ | $\mathcal{P}_4 \rightarrow \mathcal{P}_1$ | $\mathcal{P}_4 \rightarrow \mathcal{P}_2$ | $\mathcal{P}_4 \rightarrow \mathcal{P}_3$ |
|-------------|---|---|---|---|---|---|
| Quan        | 29.4s                                     | 24.6s                                     | 52.7s                                     | 42.5s                                     | 28.7s                                     | 32.8s                                     |
| Park        | 52.3s                                     | 58.7s                                     | 70.2s                                     | -   | 64.3s                                     | 75.6s                                     |
| <b>Ours</b> | <b>13.2s</b>                              | <b>10.3s</b>                              | <b>14.6s</b>                              | <b>19.2s</b>                              | <b>16.5s</b>                              | <b>21.3s</b>                              |

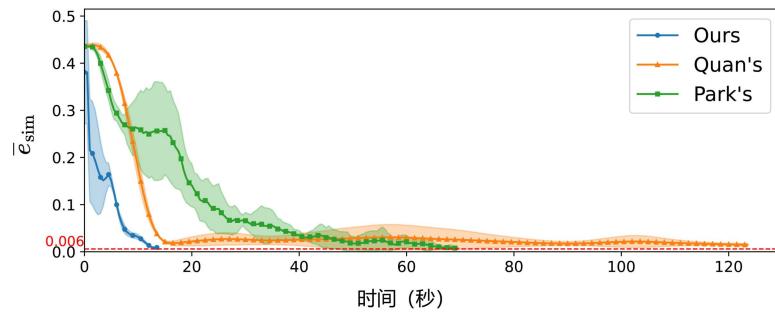
(a) 从编队模式  $\mathcal{P}_4$  收敛到  $\mathcal{P}_3$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势(b) 从编队模式  $\mathcal{P}_4$  收敛到  $\mathcal{P}_2$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势(c) 从编队模式  $\mathcal{P}_4$  收敛到  $\mathcal{P}_3$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势(d) 从编队模式  $\mathcal{P}_3$  收敛到  $\mathcal{P}_4$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势(e) 从编队模式  $\mathcal{P}_3$  收敛到  $\mathcal{P}_2$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势



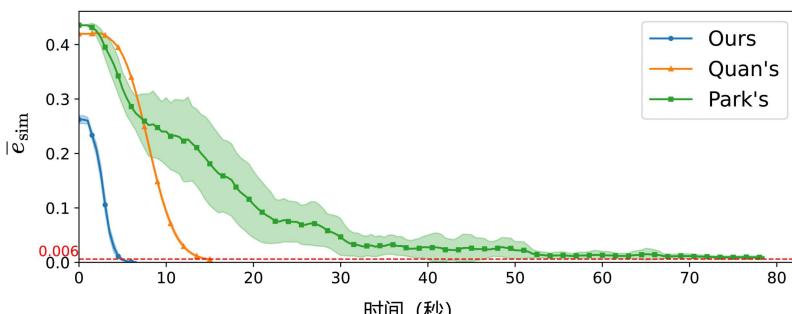
(f) 从编队模式  $\mathcal{P}_3$  收敛到  $\mathcal{P}_1$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势



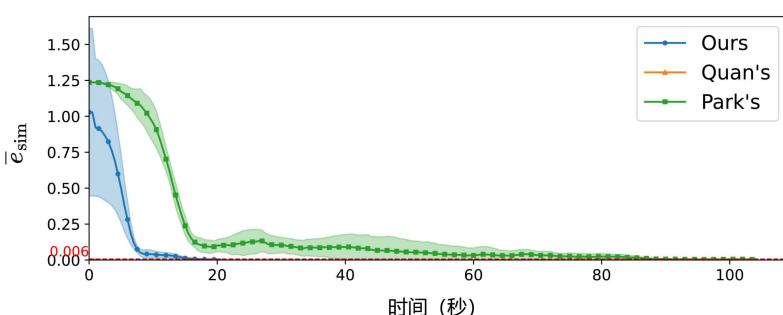
(h) 从编队模式  $\mathcal{P}_2$  收敛到  $\mathcal{P}_4$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势



(i) 从编队模式  $\mathcal{P}_2$  收敛到  $\mathcal{P}_3$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势



(j) 从编队模式  $\mathcal{P}_2$  收敛到  $\mathcal{P}_1$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势



(k) 从编队模式  $\mathcal{P}_1$  收敛到  $\mathcal{P}_4$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势

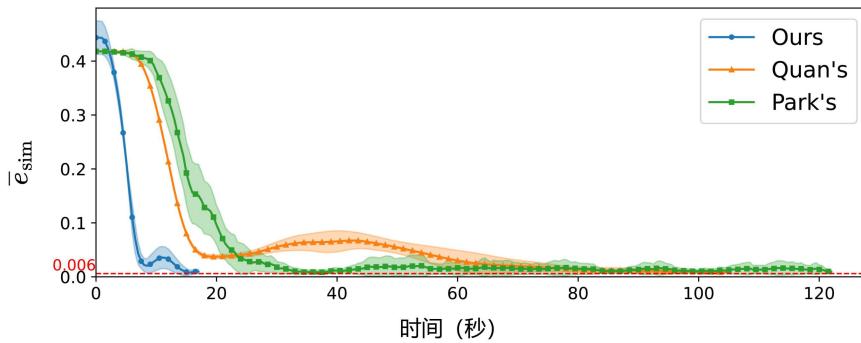
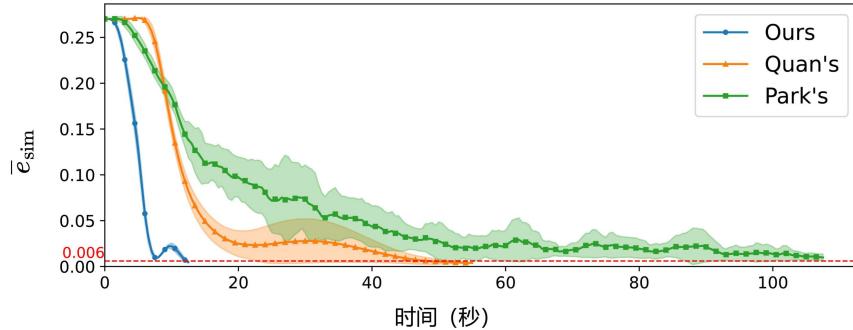
(l) 从编队模式  $\mathcal{P}_1$  收敛到  $\mathcal{P}_3$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势(m) 从编队模式  $\mathcal{P}_1$  收敛到  $\mathcal{P}_2$  过程中不同方法的平均编队相似度误差  $\bar{e}_{sim}$  变化趋势

图 4-3 不同方法在收敛到期望队形过程中的相似度误差变化趋势

**Figure 4-3. The trend of similarity error changes during the convergence to the desired formation for different methods**

表 4-1 和表 4-2 的结果显示, Park 的方法在编队收敛效率实验中的表现最差, 其主要原因在于该方法在编队收敛速度方面陷入局部最优, 具体原因包括: 未采用任务分配机制, 导致无人机在收敛过程中行程路径较长, 从而增加了时间消耗。此外, Park 的论文并未提供粒子群优化算法的具体参数, 且未公开源代码, 因此本文对 Park 的方法进行了复现, 并尽最大努力调整算法参数, 以提高其性能水平。与 Park 的方法相比, Quan 的方法在编队收敛效率上更具优势, 其主要原因在于该方法采用拟牛顿 (Quasi-Newton) 方法求解无约束优化问题, 其求解器的效率高于使用粒子群优化算法的 Park 的方法。然而, Quan 的方法的编队收敛效率仍然低于本方法。在 Quan 的方法中, 编队相似性约束与动力学可行性约束被整合至同一个目标函数, 并分别赋予不同的权重, 然而, 由于无人机平台的速度较低, 为了确保轨迹的可行性, 需要提高动力学可行性约束的权重, 但这可能会影响编队误差的快速下降。如果权重设置不当, 无人机平台可能无法有效跟随求解得到的轨迹, 从而导致编队中的无人机无法顺利收敛到期望编队形状。

Quan 的方法收敛效率低的另一个原因在于, 其编队相似性约束与  $\bar{e}_{sim}$  具有相同的性质, 使得编队形状具有旋转和尺度不变性。因此, 收敛后的队形可能相较于期望编队形状发生放大, 导致多无人机整体运行路程较长。当  $\bar{e}_{sim}$  低于预设阈值时, 收敛后的编队形状可能在旋转角度和尺度大小上与预设编队存在差异。如图

4-4 所示，提出的方法和 Park 的方法均收敛至期望的编队形状，而 Quan 的方法收敛的队形相较于期望编队有所放大。

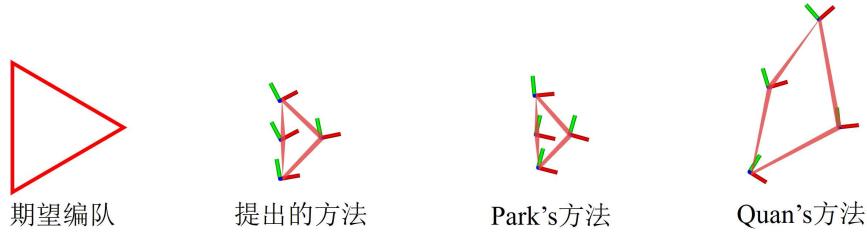


图 4-4 不同方法收敛到期望队形的结果

**Figure 4-4. Results of different methods converging to the desired formation**

最后，正如表 4-1，表 4-2 所展示的测试结果，本文的方法能快速的收敛到期望的编队模式，其主要原因在于任务分配模块有效减少了多无人车的总运行路程；同时，高效的编队控制器在最小化收敛时间方面表现更优，从而进一步提升了编队变换的效率。

#### 4.6.2 编队可扩展性分析

为了进一步测试所提队形控制方法的可扩展性，本节使用更多无人车开展编队变换收敛实验。由于多无人车自适应编队变换系统采用集中式的最优队形选择算法，并结合分布式编队控制器，从而保证系统的高效性。随着无人车数量的增加，编队的主要影响因素在于分布式编队控制器，因为每辆无人车的 MPC 控制器都需要同时考虑与周围多辆无人车保持距离，以避免发生碰撞。因此，本次实验将在第 4.6.1 节实验的基础上，进一步测试不同数量无人车的编队收敛效率。

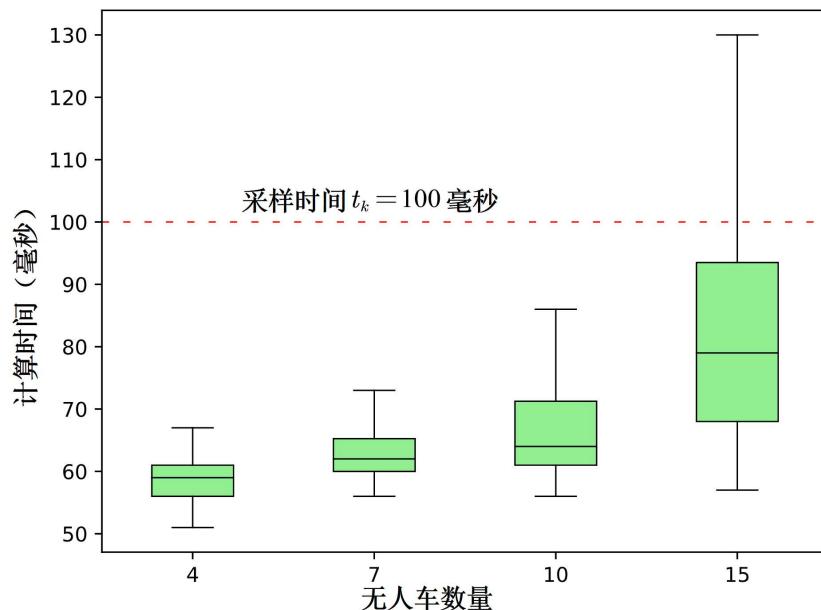


图 4-5 不同数量的无人车的编队控制器计算时间的箱线图

**Figure 4-5. Box plot of the computation time of the formation controller for autonomous vehicles with different numbers**

在编队的可扩展性实验中，仍以编队误差小于预设的阈值作为实验结束的标志。每次编队收敛实验通常执行几十次到上百次之间，具体次数取决于实验的时间。为了提高 MPC 控制器在求解控制量时的效率，将公式(4-9)中最后一个约束考虑的其他无人车的数量限制为 3。本节进行了 4, 7, 10, 和 15 辆无人车编队收敛实验，并用箱线图对每次实验的数据进行统计，实验结果如图 4-5 所示。在图 4-5 可以看出，随着无人车数量的增加，编队收敛过程中会更加频繁地受到无人车之间最小距离的约束，导致求解的时间有所增加。然而，针对 15 辆无人车的测试结果，大部分情况下计算时间仍低于 MPC 预测状态的采样时间，表明该系统能够有效处理 15 辆无人车的编队任务。图 4-6 展示了 15 辆无人车编队收敛实验中不同时刻的编队情况。

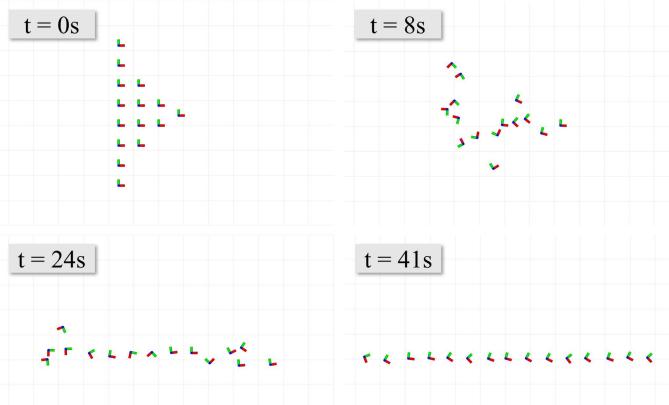


图 4-6 不同时刻的编队收敛实验中所有无人车的位置

**Figure 4-6. The positions of all unmanned vehicles in the formation convergence experiments at different time steps**

## 4.7 本章小结

本章提出了一种基于编队一致性的队形控制方法，包括基于最优运输的任务分配，基于编队一致性的控制器和基于路径的局部避障策略。为加快编队信息一致性，提出领导无人车重选择策略，实现集中式任务分配。通过任务分配减少编队变换所需的总路程。控制器基于几何特性计算参考状态，实现对局部目标点的实时跟踪，并设计基于路径引导的避障控制器，保证编队安全。实验对比表明，该方法在编队收敛效率上优于现有算法，并能高效控制 15 辆无人车完成编队变换，展现良好的适应性和可扩展性。

## 第五章 多无人车编队综合实验平台搭建与验证

### 5.1 引言

为验证基于可行域的多无人车编队方法的高效性和有效性，本章将结合基于可行域的编队选择方法（将环境信息作为输入，输出考虑环境信息的编队形状）和基于编队一致性的队形控制方法（输入为期望的编队形状，输出无人车的控制量）来完成编队导航任务，开展综合仿真对比和真实室内场景的实验。具体而言，首先，将与同类编队方法在复杂环境中进行仿真对比，以评估本文方法在编队效率和质量的实际表现；随后，通过仿真实验分析不同编队库规模对本文方法的影响；最后，在真实室内环境中验证本文方法能够充分利用环境信息，实现高效的编队导航。

本章安排如下：首先，介绍了实验平台，优化问题的求解器以及关键参数（详见 5.2 节）。随后，进行了多无人车的仿真实验平台的搭建与方法验证（详见 5.3 节）。接着，在不同的室内场景中进行多无人车的实物实验平台的搭建与方法验证（详见 5.4 节）。最后，对本章进行了总结（详见 5.5 节）。

### 5.2 实验准备

实验中采用的实验平台是 Turtlebot 平台，搭载了深度相机。前文已详细论述了多无人车自适应编队系统的规划和控制部分。然而，完整的自主编队系统还需考虑无人车的感知部分。在本实验中，无人车周围的障碍物信息是通过深度相机的深度图像来获得的。具体来说，根据相机的内参和外参，将深度图像实时转换为三维点云，并构建局部概率栅格地图，以此获取障碍物信息。本实验采用基于 ROS (Noetic 版本) 和 TCP/IP 协议实现无人车数据共享，以确保数据传输的稳定性，同时具备出错重传机制。实验重要参数如表 5-1 所示。本文采用开源求解器 LBFGS-Lite<sup>1</sup>处理无约束优化问题，如公式(3-8)，并利用 CasADi<sup>[78]</sup>求解带约束优化问题，包括公式(3-5)和公式(4-9)。

表 5-1 编队相关参数表

Table 5-1. Table of formation-related parameters

| 参数          | 符号        | 值       |
|-------------|-----------|---------|
| 公式(4-6)的距离值 | $x_{th}$  | 0.5m    |
| 最大线速度       | $v_{max}$ | 0.22m/s |

<sup>1</sup><https://github.com/ZJU-FAST-Lab/LBFGS-Lite>

表 5-1 编队相关参数表 (续表)

| 参数                                     | 符号                 | 值                 |
|--|--------------------|-------------------|
| 最大角速度                                  | $w_{max}$          | 1.5rad/s          |
| 状态误差权重矩阵                               | $\mathbf{Q}$       | diag(0.1,0.1,0.0) |
| 控制输入误差矩阵                               | $\mathbf{R}$       | diag(0.02,0.02)   |
| $\varepsilon$ 相关的优化权重                  | $w_\varepsilon$    | 100               |
| $\varepsilon$ 的最小值                     | $\varepsilon_{th}$ | -0.3              |
| 无人车之间约束的最小距离                           | $d_{safe}$         | 0.3m              |
| 采样时间                                   | $t_k$              | 0.1s              |
| 预测未来状态的单位时间数量                          | $M$                | 40                |
| 最大编队宽度 (四辆无人车为例)                       | $w_{mf}$           | 2.4m              |
| 公式(3-8)中考虑局部目标点信息的权重                   | $w_{r1}$           | 5                 |
| 公式(3-8)中考虑局障碍物信息的权重                    | $w_{r2}$           | 3                 |
| 公式(3-8)中与障碍物的安全距离                      | $d_{th,r}$         | 1.2m              |
| 公式(4-12)中集合 $\mathcal{S}_i$ 考虑最近障碍物的个数 | $\mathcal{N}_o$    | 4                 |

### 5.3 多无人车编队仿真实验

#### 5.3.1 实验平台搭建

仿真实验使用的硬件平台是搭载英特尔 i7-14650HX CPU 和 16G RAM 的笔记本，软件平台是 Ubuntu 20.04。图 5-1 中展示了测试环境：(1 米+3.1 米) × 7.4 米的锥形走廊和 20 米×15 米的随机障碍物场景，其中，锥形走廊场景用于评估编队方法对于可变通行宽度的适应能力，而随机障碍物场景通过仿真不同尺寸障碍物，验证编队方法的有效性。图 5-2 展示了四辆无人车实验平台。

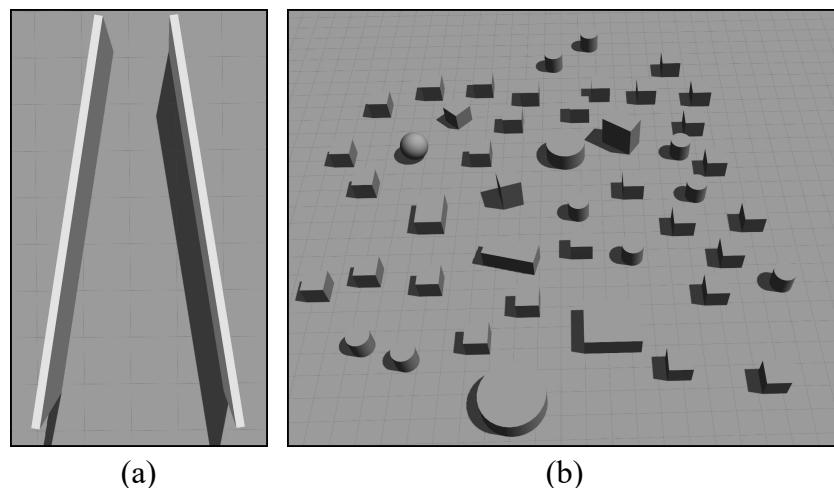


图 5-1 锥型走廊和大量障碍物的场景

Figure 5-1. Scenes of a conical corridor and numerous obstacles

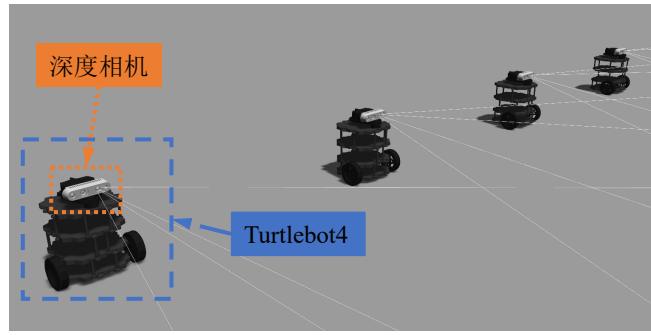


图 5-2 无人车集群实验平台

Figure 5-2. Autonomous vehicle swarm experiment platform

### 5.3.2 受限环境下编队导航实验对比分析

为了最小化障碍物对多无人车编队的干扰，需实时重构队形以适应动态环境变换。基于此，本实验在 Gazebo 仿真环境下对两种编队控制方法进行对比分析：Quan 的方法<sup>[74]</sup>和 Park 的方法<sup>[64]</sup>。Park 的方法未考虑避障机制，难以直接应用于受限的环境。因此，本实验对其进行以下修改：引入基于可行区域的队形选择方法（详见第三章）和基于路径的局部避障策略（见 4.5 节）。这些改进提升了该方法在复杂环境下的适应性，同时保持了其在无障碍物环境中原有的性能。在测试实验中，每种场景随机选择 4 个终点，每个终点测试 5 次，总共测试 20 次。评估维度包括：

- (1) 编队导航的成功率：无人车在编队导航过程中没有发生碰撞次数占全部测试数量的比重；
- (2) 运行时间：无人车从起点运行到终点的总时间；
- (3) 平均编队距离误差  $\bar{e}_{dist}$ ：计算方式参考公式(4-15)；
- (4) 平均编队相似度误差  $\bar{e}_{sim}$ ：计算方式参考公式(4-17)；

实验结果如表 5-2 所示，其中 Quan 方法在两种场景下均无法提供编队距离误差的数值。这是因为 Quan 的方法定义的编队形状具有旋转，平移和尺度不变性，而  $\bar{e}_{dist}$  评估的编队形状仅具有平移不变性，因此无法用  $\bar{e}_{dist}$  有效评估 Quan 的方法的编队性能。

表 5-2 不同方法在受限环境中编队导航的性能对比

Table 5-2. Performance comparison of different methods for formation navigation in restricted environments

| 场景    | 方法       | 成功率(%) | 时间(s)  | $\bar{e}_{dist}$ | $\bar{e}_{sim}$ |
|-------|----------|--------|--------|------------------|-----------------|
| 锥形走廊  | Park     | 80     | 121.17 | 2.65             | 0.25            |
|       | Quan     | 95     | 82.54  | -                | 0.26            |
|       | Proposed | 100    | 77.28  | 1.9              | 0.11            |
| 大量障碍物 | Park     | 75     | 250.28 | 2.51             | 0.23            |
|       | Quan     | 95     | 142.32 | -                | 0.22            |
|       | Proposed | 100    | 142.16 | 1.72             | 0.079           |

如表 5-2 所示, Park 的方法展现了较差的编队性能, 主要原因是: 当无人车接近障碍物时, 控制器需要根据避障策略计算新的局部目标点, 并引导无人车远离障碍物; 并且 Park 的方法改进是在传统的 PSO 算法的基础上加入新的粒子过滤条件, 虽然该方法能够收敛至任意队形, 但减少了粒子的搜索空间, 限制了无人车对于可行空间的探索能力, 导致无人车在复杂环境中反应不够及时, 从而增加了与障碍物发生碰撞的风险, 降低了成功率; 此外, 该方法未考虑任务分配, 因此在面对障碍物时需要频繁调整队形, 导致无人车的总运行路程增加, 从而延长了整体运行时间。总而言之, 与 4.6.1 小节中分析的原因相同, 该方法仅能找到编队收敛相对较慢的解, 使得在障碍物或编队扭曲的场景中难以快速收敛到期望的编队模式。

Quan 的方法验证了在小体积障碍物场景下编队保持的有效性, 其实验场景允许障碍物出现在编队内部, 编队的尺寸远大于障碍物。然而, 在图 5-1 场景中障碍物的体积相对较大, 无人车构成的编队只能从障碍物的一侧通过。Quan 的方法并没有利用环境信息来变换队形, 而是在无约束优化问题中同时考虑避免碰撞的约束和编队相似度约束。当环境的可通行区域减小时, 避免碰撞约束的优先级高于编队相似度约束, 导致在图 5-1 场景中编队相似度误差  $\bar{e}_{sim}$  较高, 编队性能下降。此外, 若无人车编队进入只能容纳单辆无人车的狭窄区域, Quan 的方法可能无法有效调整队形, 导致无人车碰撞障碍物, 从而降低成功率。

最后, 表 5-2 展示本文的方法在成功率、编队相似性误差和编队距离误差方法等性能指标均优于两种对比方法。这是因为自适应编队变换方法能够根据环境的可通行宽度实时的切换队形, 从而显著降低与障碍物发生碰撞的概率。此外, 高效的编队控制器确保无人车能够快速的收敛到期望队形。在大量障碍物的场景中, 如图 5-1(b)所示, 本文的方法将多辆无人车作为一个整体, 确保无人车之间的距离不会超过预设值  $N \cdot d_t$  ( $N$  表示机器人的数量,  $d_t$  表示编队中预设的无人车之间距离), 有效减少单辆无人车分散绕过障碍物时可能导致的运行总路程增加, 从而有利于提升多无人车编队的通信稳定性。此外, 图 5-3 展示四辆无人车通过多次编队变换在图 5-1(b)的场景中安全地完成编队导航。

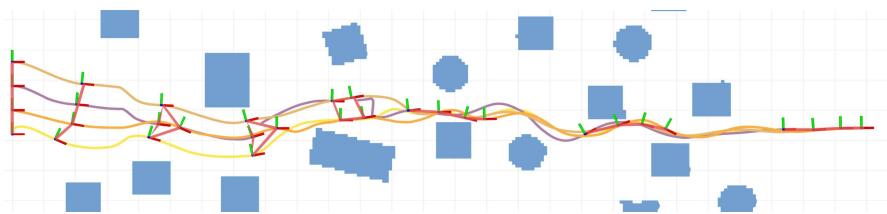


图 5-3 无人车在穿越图 5-1(b)的复杂环境时不同时刻的示意图

**Figure 5-3.** A diagram of the unmanned vehicle at different time points while traversing the complex environment in Figure 5-1(b)

### 5.3.3 编队库中不同编队模式数量的实验分析

在 2.2 节中提到，编队库是在系统运行前预先定义的。本节将分析编队库中编队模式数量对编队性能的影响。实验的仿真环境采用了锥形走廊，如图 5-1(a)所示，该环境包含了所有可能出现的可通行宽度：从最宽编队模式逐渐递减到最窄编队模式。此环境能够尽可能全面地展现编队库中的各类编队模式。具体实验中设计 10 辆无人车的在锥形走廊环境中的编队导航实验，准备了两个编队库，一个编队中包含 4 种编队模式，另一种只包含 2 种编队模式。实验效果展示在图 5-4 和图 5-5，其中图 5-4 使用包含 4 种编队模式的编队库，图 5-5 使用包含 2 种编队模式的编队库。与图 5-5 的实验相比，图 5-4 实验展示的编队模式更好地匹配当前可通行宽度。在图 5-5 中，当“领导”无人车检测可通行宽度小于当前编队的宽度会通知所有无人车变换为窄的队形，但由于编队中只有两个编队模式，所选择的编队模式比当前可通行宽度要小得多。尽管如此，这仍能保证无人车安全地通过锥形走廊。因此，本文的方法能够在给定条件下找到最佳的编队模式，无论编队库大小如何，编队库中至少存在一个最窄的编队模式。如果编队库中的编队模式的种类越多，在执行基于可行区域的队形选择算法时，就越有可能从中找到与可通行宽度匹配的编队模式。

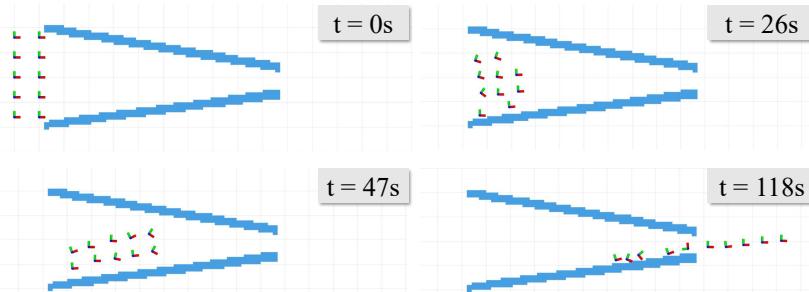


图 5-4 在锥形走廊中使用大小为 4 的编队库进行编队导航的示意图

Figure 5-4. A diagram of formation navigation in a conical corridor using a formation library of size 4

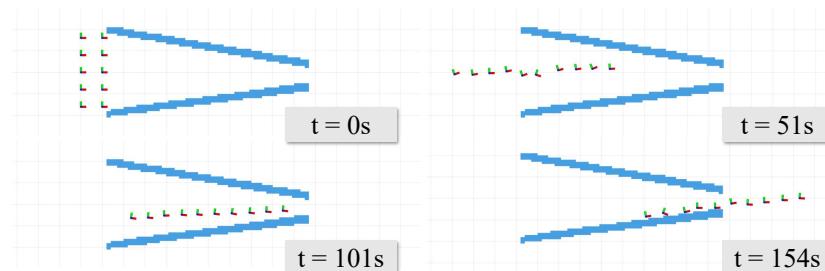


图 5-5 在锥形走廊中使用大小为 2 的编队库进行编队导航的示意图

Figure 5-5. A diagram of formation navigation in a conical corridor using a formation library of size 2

## 5.4 多无人车编队实际场景验证

上一节中搭建的 Gazebo 仿真环境可在接近现实环境的模拟环境下高效测试多无人车编队性能。为了进一步验证本文所提算法在现实场景中的实际效果，将在宽 7 米，长 6 米的室内场景中，使用四辆轮式无人车分别测试编队收敛和编队导航的运行情况。

### 5.4.1 实验平台搭建

所有实际室内实验采用 Turtlebot4 实验平台，配置完全相同，其搭载深度相机 OAK-D-Lite 和动作捕捉反光球，计算单元搭载 NVIDIA 开发板，其上都安装了 ROS 系统作为无人车的运行模块。Turtlebot4 实验平台间通过无线（WIFI）模块实现编队数据共享，每个无人车的实时姿态由 OptiTrack 动作捕捉系统采集，如图 5-6 所示。无人车实验平台如图 5-7 所示。

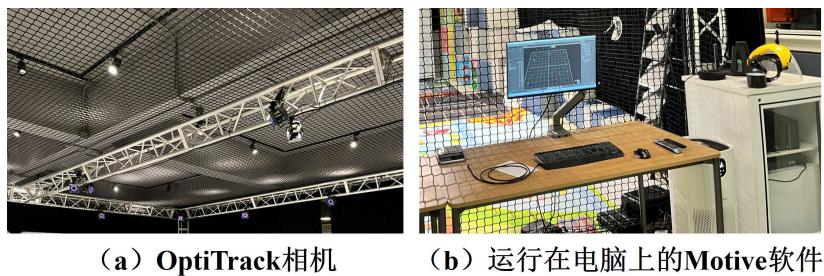


图 5-6 OptiTrack 动作捕捉系统

Figure 5-6. OptiTrack motion capture system

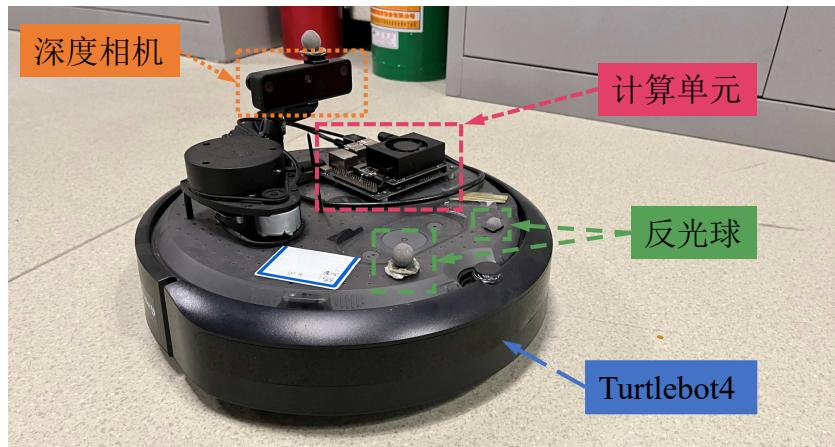


图 5-7 无人车实验平台

Figure 5-7. Unmanned vehicle experiment platform

### 5.4.2 无障碍物环境中的编队收敛实验

由于场地大小的限制，本次实验将在场地的边界处设置期望的编队模式。其他的实验配置详见第 5.2 节。实验中验证了三组编队变换的收敛效果，分别是从编队模式  $\mathcal{P}_3$  收敛到  $\mathcal{P}_4$ ，从  $\mathcal{P}_2$  收敛到  $\mathcal{P}_3$  和从  $\mathcal{P}_1$  收敛到  $\mathcal{P}_2$ 。从图 5-8，图 5-9 和图 5-10 的编队收敛过程可以看出，实验结果与仿真实验基本一致。

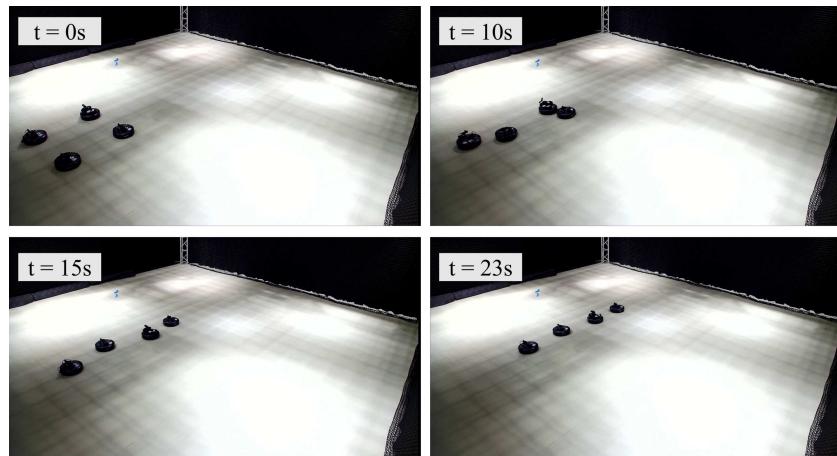


图 5-8 最窄编队模式收敛过程的示意图

Figure 5-8. A diagram of the convergence process of the narrowest formation mode

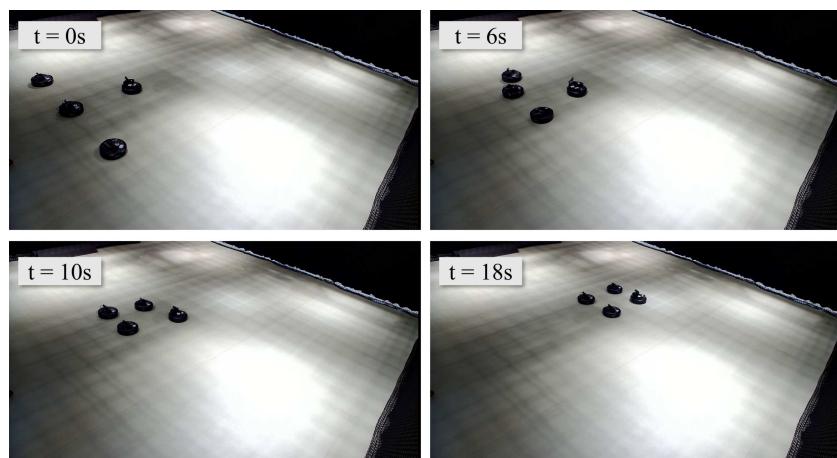


图 5-9 从三无人车宽度队形收敛到两无人车宽度队形的示意图

Figure 5-9. A diagram of the convergence process from a three-vehicle-wide formation to a two-vehicle-wide formation

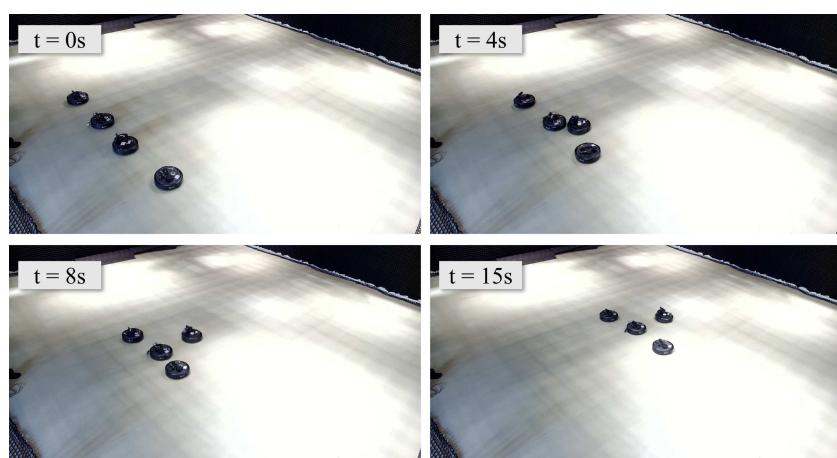


图 5-10 从四无人车宽度队形收敛到三无人车宽度队形的示意图

Figure 5-10. A diagram of the convergence process from a four-vehicle-wide formation to a three-vehicle-wide formation

### 5.4.3 有障碍物环境中的编队导航实验

本小节在障碍物的环境中进行编队导航实验。为了充分验证多无人车编队在不同场景下的适应性，选择了四种不同障碍物分布的室内环境进行测试。

第一个室内环境为锥形走廊环境。如图 5-11 所示，初始编队模式为中的  $\mathcal{P}_1$ ，即最宽的编队模式。由于在锥形走廊环境中可通行宽度逐渐缩小。当运行时间为 12 秒时，四辆无人车收敛至的编队模式  $\mathcal{P}_2$ ，此时编队模式与环境较为匹配。随着，四辆无人车持续前进，并在 26 秒时进一步收敛至编队模式  $\mathcal{P}_3$ ，顺利通过锥形走廊。当运行到 32 秒时，无人车保持编队到达终点。实验结果表明，所提的方法在现实环境中能根据环境的可通行宽度实时的调整编队形状，确保编队运行的安全性。

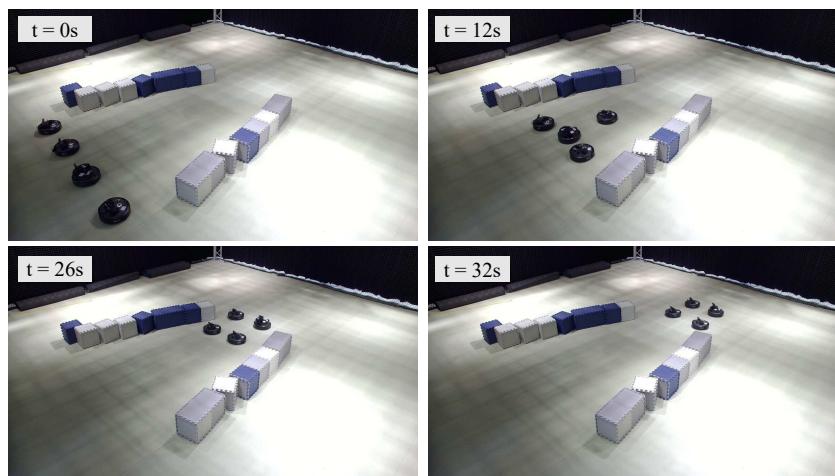


图 5-11 四辆无人车在锥形环境中编队导航过程中不同时刻的截图

**Figure 5-11. A set of screenshots of four autonomous vehicles at different times during their formation navigation in a conical environment**

第二个室内环境包含随机摆放的障碍物，如图 5-12 所示。四辆无人车初始的编队形状仍为中的  $\mathcal{P}_1$ ，即最宽的编队模式。由于环境的可通行宽度迅速变窄，在运行时间为 6 秒时，四辆无人车的形状直接收敛至中的编队模式  $\mathcal{P}_3$ 。但由于相机噪声的存在，“领导”无人车在检测最大可通行宽度时，测得的宽度小于场景中真实的宽度，因此在 17 秒时，四辆无人车的编队模式进一步调整为的  $\mathcal{P}_4$ ，即最窄的编队模式。随着时间的推移，“领导”无人车检测到真实的最大可通行宽度，使得四辆编队无人车在 33 秒时再次变换为的编队模式  $\mathcal{P}_3$ 。实验结果表明，即使在传感器噪声干扰的情况下，该方法仍能根据当前环境条件选择最佳的编队模式，确保无人车安全，无碰撞的完成编队导航任务。

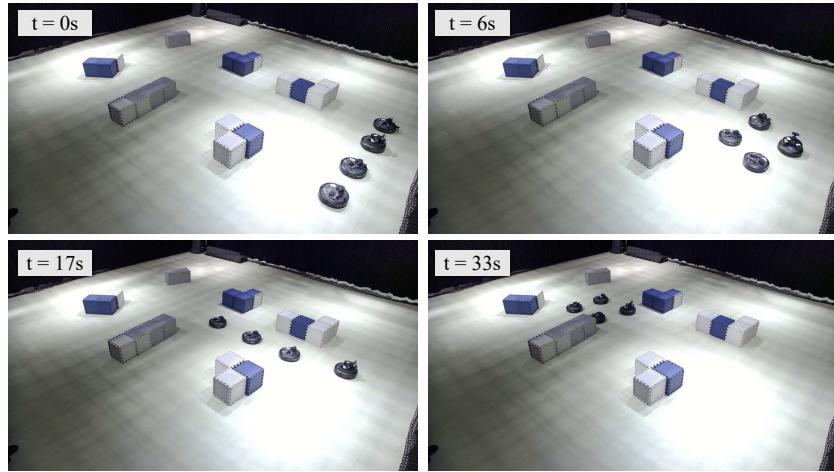


图 5-12 四辆无人车在有障碍物的环境中编队导航过程中不同时刻的截图

**Figure 5-12. A set of screenshots of four autonomous vehicles at different times during their formation navigation in an environment with obstacles**

第三个室内环境是狭窄通道场景，如图 5-13 所示，四辆无人车自主选择合适的编队模式，以安全通过狭窄的通道。初始编队模式为中的  $\mathcal{P}_2$ 。当“领导”无人车检测到狭窄通道的宽度小于当前编队模式的宽度时，控制无人车收敛至更窄的编队模式。如图 5-13 所示，在运行到 12 秒时，编队模式收敛为的编队模式  $\mathcal{P}_2$ 。当四辆无人车编队安全通过狭窄通道后，可通行宽度增大，在 27 秒时，四辆无人车重新变换回最初的编队模式。在图 5-14 展示了编队导航中编队误差的变化情况。当编队模式发生变化时，编队误差会瞬间增加，随后四辆无人车调整至新的编队模式，使编队误差迅速下降。实验结果表明，该方法能够根据环境变化自主的扩大或缩小编队模式，使无人车编队适应不同可通行宽度，实现安全高效的导航。

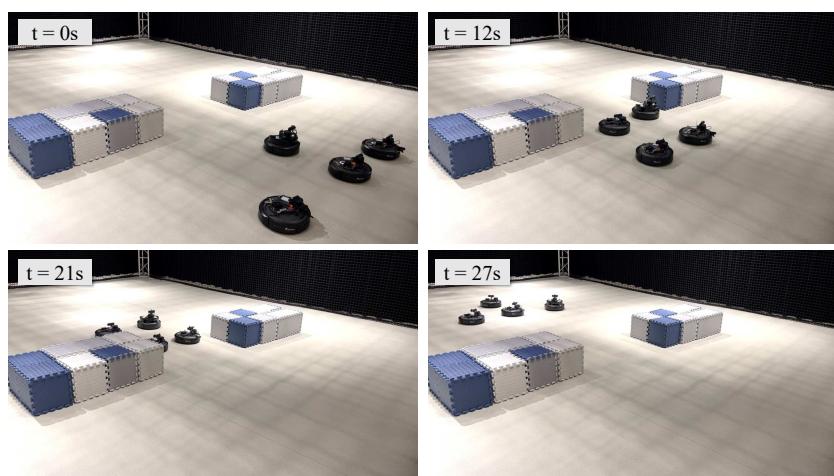


图 5-13 四辆无人车在狭窄的通道中编队导航过程中不同时刻的截图

**Figure 5-13. A set of screenshots of four autonomous vehicles at different times during their formation navigation in a narrow passage**

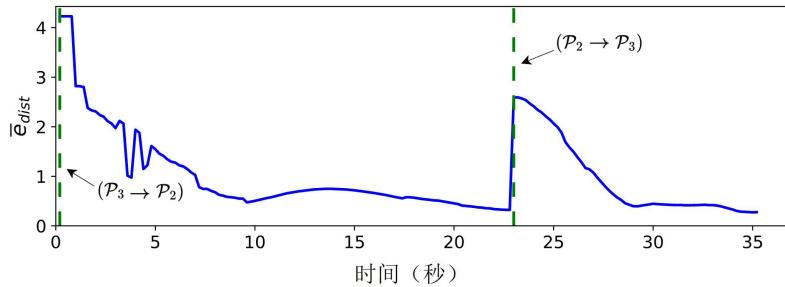


图 5-14 四辆无人车在狭窄的通道中编队导航过程中的编队距离误差图

Figure 5-14. A plot of the formation distance error during the formation navigation process of four driverless vehicles in a narrow passage

为了在环境中尽可能多的展示编队适应环境的能力，设计了第四个场景，包含两个大体积障碍物。实验结果如图 5-15 和图 5-16 所示，其中蓝色渐变的多段直线表示已知的全局路径，多条白色直线组成编队形状。为更清晰地呈现整个编队导航过程，用图 5-15 展示前 203 秒内的多无人车编队导航过程，而图 5-16 则展示从 203 秒到 301 秒多无人车编队导航情况。在图 5-15 中，由于环境的约束，多无人车在运行过程中多次变换队形，队形的顺序依次是中  $P_2$ ,  $P_3$ ,  $P_2$ ,  $P_4$ ,  $P_2$ ,  $P_3$ 。在图 5-16 中，为适应环境中可通行宽度的变换，多无人车编队依次变换为  $P_3$ ,  $P_3$ ,  $P_4$ ，最终完成导航任务。实验结果表明，即使在存在大体积障碍物且可通行宽度随机变化的环境中，本文的方法仍能准确的检测环境的可通行宽度，并从编队库中选择合适的编队模式适应复杂环境。

所有实验效果的视频链接：<https://www.bilibili.com/video/BV1ptEEzQEHm>。

代码仓库链接：<https://github.com/NeSC-IV/DEFORM>。

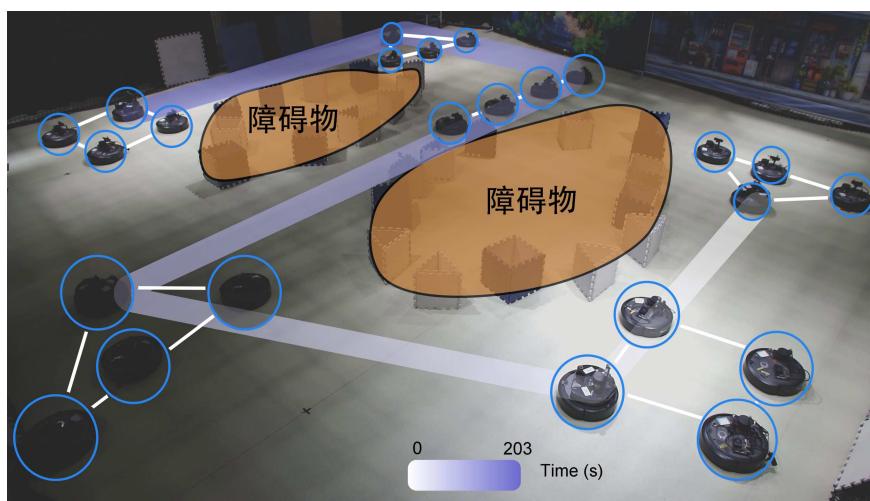


图 5-15 大体积障碍物环境中的编队导航实例（0-203 秒）

Figure 5-15. An example of formation navigation in a large obstacle environment (0-203s)

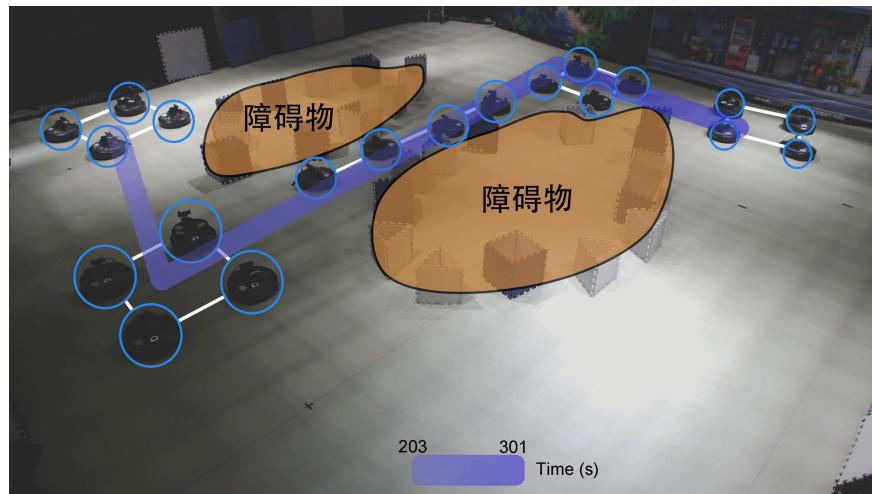


图 5-16 大体积障碍物环境中的编队导航实例（203-301 秒）

Figure 5-16. An example of formation navigation in a large obstacle environment (203-301s)

## 5.5 本章小结

针对复杂环境下的多无人车自适应编队变换算法，本章搭建了基于 Gazebo 仿真和 Turtlebot4 集群的多无人车实验平台。在 Gazebo 仿真实验中，设计了两种不同的测试环境，并用于开展受限环境下的编队导航的对比实验和不同编队库规模的实验。实验表明，本文所提方法与相比于现有方法，在关键指标上有明显提升，并能根据不同编队库规模动态适应环境。在 Turtlebot4 实机实验中，设计了四种测试场景，以验证了本文方法在不同障碍物环境下的适应能力。

## 第六章 总结与展望

### 6.1 工作总结

本文针对多无人车在复杂环境中的编队导航问题展开研究，提出了一种基于可通行宽度自适应调整队形的多无人车编队算法。该算法首先采用中心化方法求解编队级路径规划，并结合可通行宽度检测，获取考虑当前编队宽度的导航路径和匹配期望编队模式的无障碍物宽度。随后，基于编队一致性的队形控制方法以分布式的方式实现多无人车的局部控制，实现期望的编队模式，动态调整队形以适应环境变化。大量实验结果表明，该方法在复杂环境中能够高效完成编队导航，尤其是在大体积障碍物环境。

本文的主要研究成果如下：

针对编队控制对周围环境信息利用不足的问题，本文提出了一种基于可行区域的队形选择方法。该方法首先采用以圆形区域代替二维位置来表示路径点的路径规划算法，并在其评分函数中引入当前的编队宽度。实验表明，该策略可以在一定程度上调整路径与障碍物的距离，在提升安全性的同时减少编队模式的频繁变换，从而提高编队导航的整体效率。接着，提出了最大可通行宽度检测算法。该算法根据局部目标点和周围环境的几何信息，构造二次规划问题以求解垂直于路径的最大无障碍物宽度。通过消融实验表明，与编队级路径规划相比，该算法能更准确地计算无障碍物宽度。最后，通过多次迭代运行最大可通行宽度检测算法，最终得到的无障碍宽度作为队形选择的依据。

针对现有算法编队收敛效率低的问题，本文提出了基于编队一致性的队形控制方法。该方法首先根据期望队形，利用基于最优运输的任务分配来建立无人车与期望编队之间的对应关系，从而减少编队变换的总路程。为了加快无人车间的编队信息一致性，设计了领导无人车重选择策略，使临时的领导无人车执行基于可行区域的队形选择和基于最优运输的任务分配，从而缩短信息共享的时间。最后，设计了基于编队一致性的分布式控制器，以控制无人车快速形成期望编队。通过与其他编队控制算法对比，验证了本章的队形控制方法能够高效地引导无人车收敛到期望队形。此外，还进行了编队可扩展性实验，结果表明本章提出的算法能够支持多达 15 辆无人车的编队运行。

针对本文提出的基于可行域的多无人车编队变换方法，进行了综合仿真与实物实验。首先，通过与其他编队控制算法进行了仿真对比实验，验证了本文算法在编队导航的优势。此外，开展了不同编队库规模的编队控制实验，结果表明多无人车编队导航几乎不受编队库规模的影响。为了进一步证明本文算法在实际环

境中的运行效果，本文基于 Turtlebot4 实验平台搭建了多种室内场景。实验结果表明，本文提出的多无人车自适应编队变换方法在真实环境中具有良好的可行性和有效性。

## 6.2 工作展望

在现有研究的基础上，可进一步开展的研究包括：

(1) 设计更高效和通用的控制器。

目前，多无人车编队方法主要针对平坦路面进行编队控制。然而，现实世界中的大多数场景为室外崎岖地形，现有方法仍存在一定的局限性。未来研究可进一步扩展本文方法至三维空间，并结合坡面拟合算法，使机器人能够更好地适应崎岖地形，提高其在实际应用中的适应性和鲁棒性。此外，可进一步优化控制器设计，以提升计算效率和协调能力，从而实现更大规模的编队控制。

(2) 结合多机器人分布式定位。

对于完全自主的机器人系统，应包括定位、感知、规划和控制模块。然而，本文提出的方法仅涵盖感知、规划和控制。为了拓展算法的适用场景，可进一步设计分布式多机器人相对定位系统，以增强系统的自主性和适应性。

(3) 改善任务分配。

目前任务分配采用集中式的方式，当算法应用于大规模机器人编队任务时，计算负担显著增加，影响编队运行效率。为此，应设计分布式通信架构，以降低因机器人数量增长而带来的计算负担，提高多机器人系统的扩展性和实时性。

(4) 添加动态障碍物处理策略

当前编队导航框架的运行频率为 10 Hz，能够应对较为简单的动态环境。然而，对于如行人密集街道等复杂动态场景，仍需引入更为完善的动态障碍物处理策略。针对动态障碍物的应对大致可分为两个方向：一是提升现有框架的响应速度，例如加快控制器的求解效率，以增强对快速变化环境的适应能力；二是引入动态障碍物轨迹预测方法，使系统能够提前预判潜在风险，从而实现更主动的规避与调整。

## 参考文献

- [1] Verma J K, Ranga V. Multi-robot coordination analysis, taxonomy, challenges and future scope[J]. Journal of Intelligent & Robotic Systems, 2021, 102(10): 1-36.
- [2] Pei L, Lin J, Han Z, et al. Collaborative planning for catching and transporting objects in unstructured environments[J]. IEEE Robotics and Automation Letters, 2023, 9(2): 1098-1105.
- [3] Zhi H, Zhang B, Qi J, et al. Non-Prehensile Object Transport by Nonholonomic Robots Connected by Linear Deformable Elements[J]. IEEE Robotics and Automation Letters, 2024, 9(10): 8651-8658.
- [4] Hu J, Bhowmick P, Jang I, et al. A decentralized cluster formation containment framework for multirobot systems[J]. IEEE Transactions on Robotics, 2021, 37(6): 1936-1955.
- [5] Tallamraju R, Price E, Ludwig R, et al. Active perception based formation control for multiple aerial vehicles[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 4491-4498.
- [6] Hu B B, Zhang H T, Yao W, et al. Spontaneous-ordering platoon control for multirobot path navigation using guiding vector fields[J]. IEEE Transactions on Robotics, 2023, 39(4): 2654-2668.
- [7] 高力, 陆丽萍, 褚端峰, 张勇, 吴超仲. 基于图与势场法的多车道编队控制. 自动化学报, 2020, 46(1): 117-126.
- [8] 亿航智能. 空中媒体[EB/OL]. 2020. <https://www.ehang.com/cn/formation/>.
- [9] 海康机器人 Hikrobot. CeMAT ASIA 2020 | 海康机器人“众智”集结, 邀您开启无限智造之旅! [EB/OL]. 2020. [https://mp.weixin.qq.com/s/lO\\_HOu8K\\_kmcRym31Ew7IA](https://mp.weixin.qq.com/s/lO_HOu8K_kmcRym31Ew7IA).
- [10] 小马智行. 穿越沙海风暴: 小马智卡编队自动驾驶开启无人化新时代[EB/OL]. 2024. <https://mp.weixin.qq.com/s/nshDl0x22kmJOa5-gL8puQ>.
- [11] 工业和信息化部等十二部门. 5G 规模化应用“扬帆”行动升级方案[EB/OL]. 2024. [https://www.gov.cn/zhengce/zhengceku/202411/content\\_6989412.htm](https://www.gov.cn/zhengce/zhengceku/202411/content_6989412.htm).
- [12] 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.
- [13] Abujabal N, Fareh R, Sinan S, Baziyyad M, Bettayeb M. A comprehensive review of the latest path planning developments for multi-robot formation systems. Robotica. 2023, 41(7):2079-2104.
- [14] Silver D. Cooperative pathfinding[C]. //Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. California, USA: AAAI, 2005, 1(1): 117-122.
- [15] Standley T. Finding optimal solutions to cooperative pathfinding problems[C]. //Proceedings of the AAAI Conference on Artificial Intelligence. California, USA: AAAI, 2010, 24(1): 173-178.
- [16] Wagner G, Choset H. M\*: A complete multirobot path planning algorithm with performance bounds[C]. //2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). California, USA: IEEE, 2011: 3260-3267.
- [17] Wagner G, Choset H. Subdimensional expansion for multirobot path planning[J]. Artificial Intelligence, 2015, 219: 1-24.
- [18] Sharon G, Stern R, Felner A, et al. Conflict-based search for optimal multi-agent pathfinding[J].

- Artificial Intelligence, 2015, 219: 40-66.
- [19] Boyarski E, Felner A, Stern R, et al. ICBS: The improved conflict-based search algorithm for multi-agent pathfinding[C]. //Proceedings of the International Symposium on Combinatorial Search. Ein Gedi, Israel: AAAI, 2015, 6(1): 223-225.
  - [20] Barer M, Sharon G, Stern R, et al. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem[C]. //Proceedings of the International Symposium on Combinatorial Search. Prague, Czech Republic: AAAI, 2014, 5(1): 19-27.
  - [21] Andreychuk A, Yakovlev K, Surynek P, et al. Multi-agent pathfinding with continuous time[J]. Artificial Intelligence, 2022, 305: 103662.
  - [22] Mao P, Lv S, Quan Q. Tube RRT\*: Efficient homotopic path planning for swarm robotics passing-through large-scale obstacle environments[J]. IEEE Robotics and Automation Letters, 2025, 10(3): 2247-2254.
  - [23] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. The international journal of robotics research, 2011, 30(7): 846-894.
  - [24] 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.
  - [25] Toumeh C, Lambert A. Voxel-grid based convex decomposition of 3d space for safe corridor generation[J]. Journal of Intelligent & Robotic Systems, 2022, 105: 87.
  - [26] Chen J, Liu T, Shen S. Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments[C]. //2016 IEEE International Conference on Robotics and Automation (ICRA). Stockholm, Sweden: IEEE, 2016: 1476-1483.
  - [27] Li J, Ran M, Xie L. Efficient trajectory planning for multiple non-holonomic mobile robots via prioritized trajectory optimization[J]. IEEE Robotics and Automation Letters, 2020, 6(2): 405-412.
  - [28] Gao F, Wang L, Zhou B, et al. Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments[J]. IEEE Transactions on Robotics, 2020, 36(5): 1526-1545.
  - [29] Deits R, Tedrake R. Computing large convex regions of obstacle-free space through semidefinite programming[C]. //Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics. Cham: Springer International Publishing, 2015: 109-124.
  - [30] Alonso-Mora J, Montijano E, Nägeli T, et al. Distributed multi-robot formation control in dynamic environments[J]. Autonomous Robots, 2019, 43: 1079-1100.
  - [31] Zhu H, Juhl J, Ferranti L, et al. Distributed multi-robot formation splitting and merging in dynamic environments[C]. //2019 International Conference on Robotics and Automation (ICRA). Montreal, Canada: IEEE, 2019: 9080-9086.
  - [32] Alonso-Mora J, Montijano E, Schwager M, et al. Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus[C]. //2016 IEEE International Conference on Robotics and Automation (ICRA). Stockholm, Sweden: IEEE, 2016: 5356-5363.
  - [33] Alonso-Mora J, Baker S, Rus D. Multi-robot navigation in formation via sequential convex programming[C]. //2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Hamburg, Germany: IEEE, 2015: 4634-4641.
  - [34] 褚晶,李佩文,岳倾. 基于约束优化的多智能体协同编队与避障[J]. 南京航空航天大学学报, 2024, 56(3): 545-560.

- [35] Toumeh C, Lambert A. Decentralized multi-agent planning using model predictive control and time-aware safe corridors[J]. IEEE Robotics and Automation Letters, 2022, 7(4): 11110-11117.
- [36] Zhong X, Wu Y, Wang D, et al. Generating large convex polytopes directly on point clouds[J]. arXiv preprint arXiv:2010.08744, 2020.
- [37] Guo S, Liu B, Zhang S, et al. Continuous-time gaussian process trajectory generation for multi-robot formation via probabilistic inference[C]. //2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic: IEEE, 2021: 9247-9253.
- [38] Oh K K, Park M C, Ahn H S. A survey of multi-agent formation control[J]. Automatica, 2015, 53: 424-440.
- [39] Chen J, Sun D, Yang J, et al. Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme[J]. The International Journal of Robotics Research, 2010, 29(6): 727-747.
- [40] Panagou D, Kumar V. Cooperative visibility maintenance for leader – follower formations in obstacle environments[J]. IEEE Transactions on Robotics, 2014, 30(4): 831-844.
- [41] 乔磊,李宗刚,杜亚江. 有限时间内非完整移动机器人编队控制[J]. 计算机工程与应用, 2023, 59(13): 74-81.
- [42] 蒋鸿福. 多机器人编队控制与避障策略研究[D]. 广东: 华南理工大学, 2023.
- [43] Duran S, Gazi V. Adaptive formation control and target tracking in a class of multi-agent systems[C]. //Proceedings of the 2010 American Control Conference. Maryland, USA: IEEE, 2010: 75-80.
- [44] 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.
- [45] 张硕,吴雨洋,汪洋,等. 基于模型预测控制的无人车编队避障方法[J]. 北京理工大学学报, 2025, 45(1): 34-41.
- [46] Sabattini L, Secchi C, Fantuzzi C. Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation[J]. Autonomous Robots, 2011, 30: 385-397.
- [47] Pan Z, Zhang C, Xia Y, et al. An improved artificial potential field method for path planning and formation control of the multi-UAV systems[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2021, 69(3): 1129-1133.
- [48] Jin J, Gans N. Collision-free formation and heading consensus of nonholonomic robots as a pose regulation problem[J]. Robotics and Autonomous Systems, 2017, 95: 25-36.
- [49] Wei L, Chen M, Shi S. Dynamic event-triggered consensus cost-based switching control for UAV formation with disturbances[J]. IEEE Transactions on Intelligent Vehicles, 2023, 9(2): 3531-3543.
- [50] Quan L, Yin L, Zhang T, et al. Robust and efficient trajectory planning for formation flight in dense environments[J]. IEEE Transactions on Robotics, 2023, 39(6): 4785-4804.
- [51] Xu Y, Zhao S, Luo D, et al. Affine formation maneuver control of high-order multi-agent systems over directed networks[J]. Automatica, 2020, 118: 109004 .
- [52] Xue D, Yao J, Chen G, et al. Formation control of networked multi-agent systems[J]. IET Control Theory & Applications, 2010, 4(10): 2168-2176.
- [53] Lee G, Chwa D. Decentralized behavior-based formation control of multiple robots considering obstacle avoidance[J]. Intelligent Service Robotics, 2018, 11: 127-138.
- [54] Turpin M, Michael N, Kumar V. Trajectory design and control for aggressive formation flight with quadrotors[J]. Autonomous Robots, 2012, 33: 143-156.

- [55] Wang H, Rubenstein M. Shape formation in homogeneous swarms using local task swapping[J]. *IEEE Transactions on Robotics*, 2020, 36(3): 597-612.
- [56] 贾一飞,蒋朝阳. 动态可重构自适应无人车辆编队系统[J]. *兵工学报*, 2024, 45(10): 3654-3673.
- [57] Sun G, Zhou R, Ma Z, et al. Mean-shift exploration in shape assembly of robot swarms[J]. *Nature Communications*, 2023, 14: 3476.
- [58] Karimi M A, Alizadehyazdi V, Busque B P, et al. A boundary-constrained swarm robot with granular jamming[C]. //2020 3rd IEEE International Conference on Soft Robotics (RoboSoft). Connecticut, USA: IEEE, 2020: 291-296.
- [59] Karimi M A, Alizadehyazdi V, Jaeger H M, et al. A self-reconfigurable variable-stiffness soft robot based on boundary-constrained modular units[J]. *IEEE Transactions on Robotics*, 2021, 38(2): 810-821.
- [60] Tanaka K, Karimi M A, Busque B P, et al. Cable-driven jamming of a boundary constrained soft robot[C]. //2020 3rd IEEE International Conference on Soft Robotics (RoboSoft). Connecticut, USA: IEEE, 2020: 852-857.
- [61] Mulroy D, Lopez E, Spenko M, et al. Using R-functions to control the shape of soft robots[J]. *IEEE Robotics and Automation Letters*, 2022, 7(4): 8598-8603.
- [62] Morgan D, Subramanian G P, Chung S J, et al. Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming[J]. *The International Journal of Robotics Research*, 2016, 35(10): 1261-1285.
- [63] Zhang B, Shao X, Zhi H, et al. Leader-Follower Formation Control of Perturbed Nonholonomic Agents along Parametric Curves with Directed Communication[J]. *IEEE Robotics and Automation Letters*, 2024, 9(10): 8603-8610.
- [64] Park S, Lee S M. Formation reconfiguration control with collision avoidance of nonholonomic mobile robots[J]. *IEEE Robotics and Automation Letters*, 2023, 8(12): 7905-7912.
- [65] Peng P, Dong W, Chen G, et al. Obstacle avoidance of resilient UAV swarm formation with active sensing system in the dense environment[C]. //2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Kyoto, Japan: IEEE, 2022: 10529-10535.
- [66] Guo J, Qi J, Wang M, et al. Distributed cooperative obstacle avoidance and formation reconfiguration for multiple quadrotors: Theory and experiment[J]. *Aerospace Science and Technology*, 2023, 136: 108218.
- [67] Fu X, Pan J, Wang H, et al. A formation maintenance and reconstruction method of UAV swarm based on distributed control[J]. *Aerospace Science and Technology*, 2020, 104: 105981.
- [68] Huang C, Xu T, Wu X. Leader – follower formation control of magnetically actuated millirobots for automatic navigation[J]. *IEEE/ASME Transactions on Mechatronics*, 2023, 29(2): 1272-1282.
- [69] Chen Y, Li Z. Formation adaptation in obstacle-cluttered environments via MPC-based trajectory planning[J]. *Science China Information Sciences*, 2024, 67: 174201.
- [70] 刘福琳,李庆鑫. 多移动机器人混合避障算法的编队策略[J]. *系统仿真学报*, 2024, 36(3): 726-734.
- [71] 常路,单梁,戴跃伟,等. 未知环境下基于改进 DWA 的多机器人编队控制[J]. *控制与决策*, 2022, 37(10): 2524-2534.
- [72] Lee S M, Lee J U. Multi-robot formation planning in maze-like environments consisting of narrow passages using graph search[J]. *IEEE Access*, 2024, 12: 167694-167704.
- [73] 李兆博,孙双蕾. 基于鸽群优化改进动态窗的多无人车协同编队避障控制[J]. *工程科学*

- 学报, 2024, 46(7): 1279-1285.
- [74] 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). Pennsylvania, USA: IEEE, 2022: 4979-4985.
- [75] Chen N, Li Z, Quan L, et al. Cost-effective Swarm Navigation System via Close Cooperation[J]. IEEE Robotics and Automation Letters, 2024, 9(11): 9343-9350.
- [76] Gammell J D, Srinivasa S S, Barfoot T D. Informed RRT\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic[C]. //2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Illinois, USA: IEEE, 2014: 2997-3004.
- [77] Flamary R, Courty N, Gramfort A, et al. POT: Python optimal transport[J]. Journal of Machine Learning Research, 2021, 22(78): 1-8.
- [78] Andersson J A E, Gillis J, Horn G, et al. CasADi: A software framework for nonlinear optimization and optimal control[J]. Mathematical Programming Computation, 2019, 11: 1-36.