



北京大学

本科生毕业论文

题目：基于人工势场法的机器人任务

与运动规划算法研究

Robot Task and Motion

Planning Based on Artificial

Potential Field

姓 名：王帅康

学 号：1900011011

院 系：工学院

专 业：机器人工程

导师姓名：国萌 助理教授

二〇二三年六月

版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以任何方式传播。否则，引起有碍作者著作权之问题，将可能承担法律责任。

摘要

机器人的导航与运动规划在现实生活中有着广泛的应用。导航势函数 (Navigation Function, NF) 是具有全局收敛性并且无局部最小值的势场, 可用于生成安全可靠的机器人导航策略。其核心思想是对特定形状的障碍物进行微分同胚映射, 从而将障碍物相互重叠而成的丛林空间 (Forest World) 变换成相互独立的圆形空间 (Sphere World), 并在其中定义出势函数。导航势函数的优点在于, 给定障碍物的隐式表达函数后, 相关的微分同胚映射和势函数可以解析地表示出来。然而, 传统的导航势函数存在局部震荡、路径代价不是最优等缺点。并且, 在障碍物环境比较复杂时, 计算相关的转换函数十分繁琐, 特别是在机器人逐步探索未知工作空间的情况下, 往往需要重新计算转换函数, 导致计算复杂度较高。

此外, 线性时序逻辑 (Linear Temporal Logic, LTL) 公式是描述复杂时间任务的一种形式化方法, 在机器人的任务规划领域有着广泛的应用。形式化表达的语言可以通过模型检测工具进行求解, 从而得到最优的任务规划结果。但是, 当使用线性时序逻辑公式在未知环境中规划复杂任务时, 子任务之间的导航代价往往基于初始环境进行离线估计, 从而会产生次优结果。而且, 一般情况下, 任务规划和运动规划是独立完成的, 但它们的相互耦合对整体规划的时效性和安全性有很大的影响。

针对上述问题, 本研究提出了一种集成任务与运动规划的复合框架, 以实现在任务自动机的指导下高效地在线迭代计算势场, 并根据环境变化在线调整任务规划以得到最优结果。特别地, 本研究提出了一种双层导航树结构 (Dual-Layer Navigation Tree, DLNT): (i) 引入带有角度偏向的搜索算法实现上层任务规划; (ii) 利用基于定向谐势场的搜索树实现底层运动规划。该双层规划结构可以根据在线执行期间的环境变化进行更新和调整, 以反映在真实工作空间中的导航可行性和最优性。通过将任务规划和运动规划相耦合, 所提方法不仅能保证全局导航的安全性和收敛性, 还能保证机器人运动的最优性和稳定性。同时, 双层规划结构可以避免传统势场法出现的局部振荡和高代价的路径, 以及传统搜索法出现的不满足运动学约束和高计算复杂度等已知问题。最后, 本文进行了大量的仿真实验和硬件实验。实验结果表明, 所提方法能够实现机器人在未知环境中高效地在线执行任务, 与一些前沿方法相比具有明显的优越性。

关键词: 任务规划, 运动规划, 任务自动机, 人工势场, 非完整约束

Abstract

Robot navigation and motion planning have significant practical applications in real-world scenarios. Navigation functions (NF) provide globally convergent potential fields that are provably free of local minima. Due to its analytical format, it is particularly suitable for generating safe and reliable robot navigation policies. The main idea is to map the original workspace to a sphere world via diffeomorphic transformations. However, traditional navigation functions have shortcomings such as oscillations and long-detours. Additionally, for complex environments that consist of a large number of overlapping obstacles, the computation of associated transformation functions can be tedious. This becomes more apparent when the workspace is initially unknown and the underlying potential fields are updated constantly as the robot explores it gradually.

Furthermore, Linear Temporal Logic (LTL) formulas are a formal method for describing complex temporal tasks and suitable for task planning. Typically, LTL formulas can be solved by automaton to obtain the optimal planning results. However, when using LTL formulas to plan complex tasks in unknown environments, the navigation cost between subtasks is often estimated off-line based on initial knowledge, yielding a sub-optimal plan for the actual environment. Moreover, while the task planning and motion planning are often tackled independently, their interplay has a great impact on the overall safety and efficiency.

Thus, this work proposes an efficient and automated scheme to construct harmonic potentials incrementally online as guided by the task automaton. A novel dual-layer navigation tree (DLAT) structure is introduced that facilitates the hybrid combination of oriented search algorithms for task planning and harmonic-based navigation controllers for non-holonomic robots. Furthermore, both layers are adapted during online execution to reflect the actual navigation feasibility and cost within the updated workspace. It not only retains the global safety and convergence property, but also improves the smoothness of the generated trajectory. Known issues such as oscillation or long-detours for purely potential-based methods and sharp-turns or high computation complexity for purely search-based methods are prevented. Extensive numerical simulation and hardware experiments are conducted against several strong baselines.

KEYWORDS: Task Planning, Motion Planning, Task Automaton, Harmonic Potentials, Non-holonomic

目 录

第一章 绪论	1
1.1 研究背景及意义	1
1.2 研究现状	2
1.3 主要贡献与章节安排	5
第二章 预备知识与问题描述	7
2.1 预备知识	7
2.1.1 导航势函数的定义与性质	7
2.1.2 微分同胚映射与谐势场	7
2.1.3 线性时序逻辑与Büchi自动机	9
2.2 问题描述	10
第三章 基于谐势场的双层导航树构建	12
3.1 障碍物环境的建模	12
3.2 导航势函数的构建	15
3.2.1 星形空间到圆形空间的微分同胚映射	15
3.2.2 丛林空间到星形空间的微分同胚映射	17
3.2.3 导航势函数性质分析	23
3.3 势场的定向旋转变换	26
3.4 势场跟踪控制器的设计	29
3.5 双层导航树的构建	31
第四章 在线迭代更新	34
4.1 导航势函数的在线迭代计算	34
4.1.1 独立障碍物的迭代计算	34
4.1.2 重叠障碍物的迭代计算	38
4.1.3 一般情况下的迭代计算	40
4.2 定向搜索树的局部更新	41
4.3 导航图的局部更新	42
第五章 算法总结与复杂度分析	43
5.1 算法总结	43

5.2 复杂度分析	43
第六章 仿真与实验	44
6.1 数值仿真	44
6.1.1 环境与任务描述	44
6.1.2 仿真结果	44
6.1.3 对比	46
6.2 硬件实验	47
6.2.1 硬件平台介绍	47
6.2.2 环境与任务描述	48
6.2.3 实验结果	48
第七章 结论	50
参考文献	53
致谢	54
北京大学学位论文原创性声明和使用授权说明	55

第一章 绪论

1.1 研究背景及意义

自从人类开始意识到自己的存在，就不断在努力创造更多、更强大的工具来改善生活。机器人技术的发展，无疑是这一历史进程中最引人注目的一步。机器人的诞生为人类文明带来了前所未有的机遇和挑战，已经深刻地改变了人类的生产生活和工作方式。它们可以代替人类完成危险、繁琐、高风险和高精度的工作，比如在太空、深海和工厂环境中执行复杂任务，极大地提高了生产效率和工作安全性，为人类社会的进步起到了举足轻重的促进作用。

近年来，随着人工智能技术的快速发展，自主移动机器人在生产生活中扮演着越来越重要的角色，成为了现代科技领域的一个重要研究方向。自主移动机器人通过感知、规划和控制等技术，能够在不同环境中替代人类执行复杂任务，已经在工业制造、物流运输、农业生产、社会服务等多个领域得到了广泛的应用。对于自主移动机器人而言，任务与运动规划（Task and Motion Planning, TAMP）是至关重要的技术。其中包括两个关键问题：(i)高层任务规划（High-Level Task Planning），即需要将机器人的子任务进行分解和调度，以完成整体的任务要求；(ii)底层运动规划（Low-Level Motion Planning），即需要设计控制输入以驱动机器人从初始状态运动到目标状态，同时避免与障碍物碰撞。

另一方面，伴随着技术的不断发展和社会需求的增加，移动机器人的应用场景从简单的已知环境拓展到了复杂的未知环境。在未知环境中进行任务与运动规划需要解决两个核心问题：首先是如何根据环境的变化不断调整机器人的控制策略，使其能够安全地从初始状态运动到目标状态 [1, 2, 3, 4]。为了解决这个问题，需要设计动态的运动规划算法，使机器人能够响应式地发现新的障碍物，从而调整控制策略，以确保路径的安全性和可行性。其次是如何根据环境的变化调整任务规划，以使机器人能够高效地完成任务，并保证任务的最优性 [5, 6]。为了解决该问题，需要在线地更新任务点之间的代价，以反映机器人实际的运动代价。传统的任务与运动规划算法往往将任务规划和运动规划作为两个独立的模块进行处理，而忽略了它们之间的密切依赖关系，但两者的相互耦合对整体规划的时效性和安全性有很大的影响 [7, 8]，而且一般情况下，独立的规划方法可能会导致机器人执行效率低下、甚至不安全。因此，如何设计一种集成任务与运动规划复合规划算法，以确保机器人在未知环境中高效地执行复杂任务，是一项具有挑战并且亟待解决的问题 [9, 10]。

在机器人的任务规划算法中，最常见的方法是线性时序逻辑（Linear Temporal Logic, LTL）公式 [11]。LTL是描述复杂时间任务的一种形式化方法，可以表述具有时序属性的系统行为。形式化表达的语言可以通过模型检测工具，如Büchi自动机进行求解，从而得到最优的任务规划结果。LTL的优势在于它可以描述系统的时间性质，如何组合行为以达到特定目标，以及如何在系统行为的不同时间点实现特定的任务要求。这使得LTL成为了一种有效的工具，可用于规范机器人行为，从而完成一系列任务。但是，当使用LTL在未知环境中规划复杂任务时，子任务之间的导航代价往往基于初始环境进行离线估计，从而会产生次优结果。

在机器人的运动规划算法中，人工势场法是近几年比较热门的算法，其中导航势函数（Navigation Functions, NF）是较为前沿的一种方法 [3]。导航势函数是具有全局收敛性并且无局部最小值的势场，可用于生成安全可靠的机器人导航策略。其核心思想是对特定几何形状的障碍物进行微分同胚变映射，从而将障碍物相互重叠而成的丛林空间（Forest World）变换成相互独立的圆形空间（Sphere World），并在其中定义势函数。这种方法的优点在于，给定障碍物的隐式表达函数后，相关的微分同胚映射和势函数可以解析地表示出来。然而，传统的导航势函数存在局部震荡、路径代价不是最优等缺点。并且，在障碍物环境比较复杂时，计算相关的转换函数会十分繁琐，特别是在机器人逐步探索未知工作空间的情况下，往往需要重新计算势函数，导致计算复杂度较高。

针对这些问题，本研究提出了一种集成任务规划和运动规划的复合规划框架：根据线性时序逻辑公式实现高层任务规划，并根据任务自动机的引导在线地构建导航势函数，从而驱动机器人在任务点之间实现安全导航。所提方法解决了传统算法的局限性，实现了机器人在未知环境中高效并且安全地执行复杂任务。

1.2 研究现状

近年来，机器人的任务与运动规划领域受到了科研工作者的广泛关注和研究。为了提高机器人在执行任务时的效率和安全性，研究者们不断探索和提出了许多新的方法和技术。本文旨在对几种常见且最前沿的方法进行总结和概述。

人工势场法：关于机器人运动规划算法，人工势场法 [12, 13, 14, 15] 提供了一种直观而解析的框架来解决机器人导航过程中的安全性和收敛性问题。其主要思想是引入针对目标点的吸引力和针对障碍物的排斥力，从而在工作空间中构建出一个势函数，其在目标点取得全局最小值，而在障碍物边界上取得最大值。当机器人沿着势场的负梯度方向运动时，会自然地收敛到取得全局最小值的目标点。然而，传统的构建势场的方法会在工作空间中产生非期望的局部最小值，当机器人运动到这些局部最小

值点时，受到的合力为零，从而停滞不前。导航势函数 [3] 的问世解决了传统势场存在局部最小值的问题。其核心思想是对特定几何形状的障碍物进行微分同胚变映射，从而将现实中的复杂障碍物空间变换成圆形障碍物空间，并在其中定义势函数。该方法可以严谨地被证明出来所有的局部最小值点都是非退化的鞍点，而且其测度为零。因此，导航势函数保证了机器人运动规划的全局收敛性和安全性。得益于这些性质，导航势函数法已经在多机器人系统、动态环境、动态目标点和移动机械臂等场景中得到了应用。然而导航势函数生成的轨迹会有局部震荡，并且路径的代价比较高。此外，导航势函数的构建必须对一些关键参数进行调节。当机器人在未知环境中并且障碍物环境比较复杂的情况下，调参的过程十分繁琐，导致难以适应在线运行的情形。为了解决该问题，在 [16, 17, 18, 19] 等文献中，作者提出了将复杂障碍物空间微分同胚映射到点空间（point world）的新型变换方法，该方法保留了传统导航势函数法的收敛性和安全性等特点，并解决了需要调参的局限性。此外，在最近发表的 [10, 15, 20] 等文献中，作者避免了将复杂障碍物空间映射到圆形空间或者点空间的繁杂过程，而是采用更广泛的边界基函数集合构建局部势场，并使用优化的方法对这些基函数的权重进行优化，从而保证导航的安全性和收敛性。但是，该方法不能直接得到解析解，而是采用数值方法解决复杂的参数优化问题，因此其计算复杂度较高。综上所述，尽管导航势函数法保证了解析解、安全性和全局收敛性，但是如图 1.1 所示，它们存在以下一些缺陷：(i) 在机器人沿障碍物边界或穿过狭窄通道时，可能会出现局部震荡或抖动；(ii) 当初始位姿略有改变时，在同一势场中可能会出现完全不同的轨迹；(iii) 产生的轨迹在路径长度或控制输出等方面的代价不是最优的；(iv) 机器人到达目标位姿的最终方向无法控制。

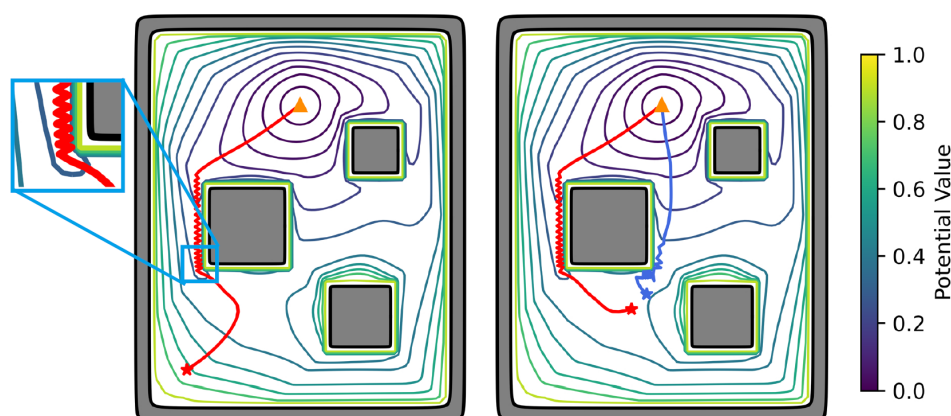


图 1.1 传统导航势函数法的轨迹图。左图：机器人能安全地从出发点（红色星号）运动到目标点（橙色三角），但在运动过程中会发生严重的局部震荡，且路径代价很高；右图：机器人分别从相近的两个出发点（红色星号和蓝色星号）运动到目标点（橙色三角），但两者的最终轨迹有很大的差异，且到达目标点的方向无法控制。

基于采样的算法：此外，基于采样和搜索的方法，如文献 [1] 中的 RRT*、[21] 中的 PRM、[22] 中的 FMT* 等，是解决高维运动规划问题的主要方法，尤其是对于存在几何约束和运动学约束的系统。但由于需要在采样状态之间进行碰撞检测并且需要大量采样点才能收敛到最优路径，其计算复杂度较高。由于人工势场法是解析的且能够保证安全性和收敛性，因此将这两种方法结合起来具有很大潜力。近几年，在这个方向上也有一些相关的研究，例如在文献 [23] 作者中使用矢量场来偏置搜索树的分支，从而提高采样效率，减少迭代次数；文献 [24]、[25] 等采用了基于势函数的 RRT* 算法，通过构建势场并在沿着势场负梯度的方向对采样点进行延伸，使得搜索树生长的方向尽可能沿着目标点的方向。与传统的 RRT* 算法相比，这些方法显示出更高的效率和更快的收敛速度。然而，这些方法主要集中在简单的而且是完全已知的环境中进行任务导航，所有规划是离线执行的，在执行过程中不需要在线更新势场，也不需要调整搜索算法的结构。因此，很难适应于未知环境下的运动规划。

未知环境下的导航算法：在未知环境中实现机器人导航时，必须采用在线的方法，因此初始时刻的规划结果必须根据实时感知得到的实际环境（如发现新的障碍物）进行调整。例如，在文献 [26] 中，作者提出了一种完全自主的调节机制来更新势场，而在文献 [27] 中，作者则提出了动态窗口的概念，以处理机器人运行期间环境的动态变化。此外，在文献 [10] 中，作者进一步改进了基于谐势场的导航方法，以应对未知环境，主要思路是对工作空间中的基函数的权重进行在线优化和更新。文献 [9] 采用类似的方法，自主地调整谐势场中的关键参数以确保全局导航的收敛性和安全性。另外，在 [28] 中，作者分析了随时间变化的动态目标的情况，并设计了一个随时间演变的势函数。另一方面，基于搜索的方法也被扩展到未知环境中，文献 [29, 30] 提出了多种在线更新搜索树结构的方法和技术。值得注意的是，将搜索的方法与势场法相结合，随着环境的变化增量式地更新两者以避免重复的计算过程，将为未知环境下的导航提供一种新思路。

线性时序逻辑：最后，当机器人的目标是在复杂的任务点之间进行导航，而非上文所述的点到点的简单导航时，就需要上层的任务规划作为引导。对于这个问题，线性时序逻辑（LTL）公式提供了一种形式化的语言，可以描述复杂的时序任务，例如连续的访问、监控和响应，并且可以使用自动机进行快速求解，得到任务规划的结果。许多较前沿的相关论文将基于线性时序逻辑的任务规划与底层的运动规划相结合，例如在 LTL 任务下的单个机器人导航 [6, 31, 32]，全局任务下的多机器人系统导航 [33, 34, 35] 等。然而，大多数上述工作常常需要假定存在一个可靠的底层控制器，或者考虑一个简单而且已知的环境，并且只针对非重叠的感兴趣区域。而在未知环境中，将复杂时间任务与导航势函数相结合的动态协同规划方法尚没有相关的研究。

1.3 主要贡献与章节安排

为了应对上述方法的局限性，本研究提出了一种集成任务规划与运动规划的框架，称为双层导航树（Dual-Layer Navigation Tree, DLNT），旨在充分利用基于导航势函数的定向搜索树进行底层运动规划和基于任务自动机的导航图进行上层任务规划，整体流程框架如图 1.2 所示。其中定向搜索树的核心思想是在任务点之间引入中间路径点以提高规划轨迹的平滑度和最优性，它是将基于任务自动机引导的谐势场在线构建和基于势场梯度跟踪的控制器共同作用的结果。此外，针对非完整约束机器人，本文提出了一种新颖的定向谐势场，并基于此设计了非线性势场跟踪控制器，以提高机器人运动的平滑性。在线执行过程中，随着机器人逐渐探索未知环境，本文提出了一种高效的在线更新方案，以迭代计算导航势函数并且相应地调整定向搜索树。在更新势场的过程中，所提方法对微分同胚映射变换中的某些中间变量进行重复利用和更新，以避免完全重新计算，与传统计算导航势函数的方法相比，降低了计算复杂度。总的来说，本研究的主要贡献在于在线混合式的规划控制框架，将控制和规划中的两

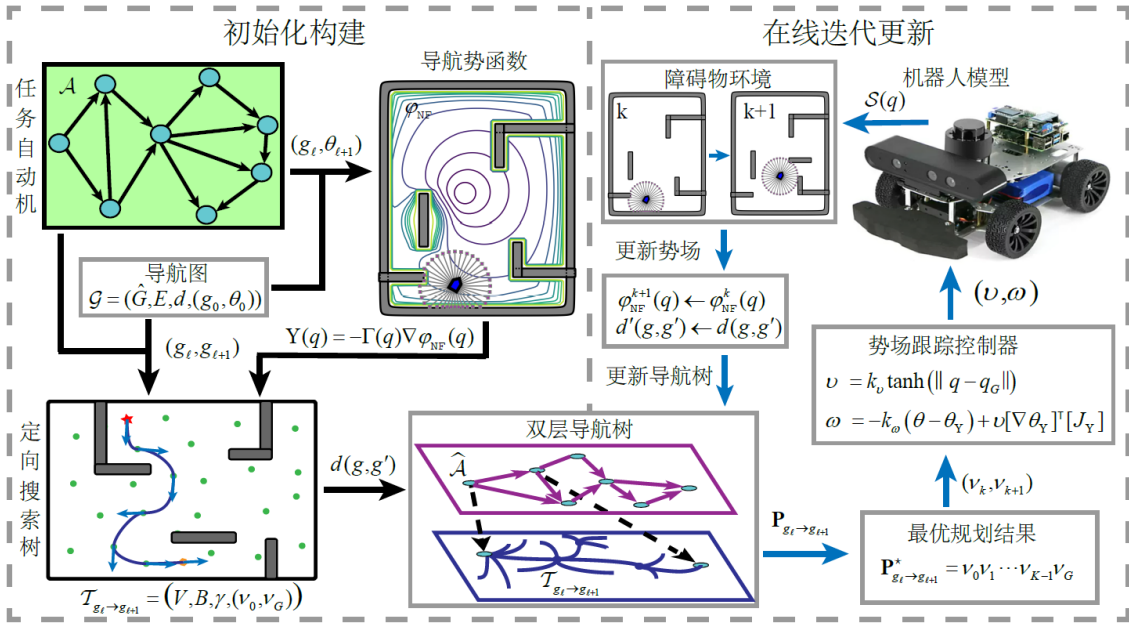


图 1.2 整体框架图。所提方法的核心为双层导航树，总体分为初始化构建和在线迭代更新两部分。其中初始化构建主要包括任务自动机、导航势函数和定向搜索树三个部分；在线迭代更新主要是根据环境的变化，迭代地计算导航势函数，并根据相应的控制器更新所估计的代价，从而对定向搜索树做出调整。

种强大方法结合起来，用于驱动非完整约束机器人在未知环境中完成复杂任务。核心贡献总结为如下几点：(i)提出了一种新的导航结构，即自动机引导的双层导航树，它将任务规划和运动控制统一起来，能保证安全性和收敛性；(ii)提出了一种新的谐势场的构建方法，可用于重叠的障碍物环境，并针对在线情况对该方法进行了优化，可

以随着障碍物的增加迭代更新势场；(iii)设计了一种基于定向谐势场的新型非线性控制器，以确保机器人的安全和平稳导航；(iv)提出了一种综合的迭代方法，可在线同时递归地更新谐势场和定向搜索树。最终，本文从理论分析、数值仿真和硬件实验上，验证了所提方法的有效性，并与传统的方法进行了对比，进一步体现了方法的优越性。实验结果表明，所提方法能避免传统势场法普遍存在的振荡或高代价问题，同时避免了纯搜索法的急转弯或高计算复杂度等问题。

针对以上所提框架和主要贡献，本文的章节安排如下：

第一章是引言，首先分析了自主移动机器人对社会发展所产生的影响，以及其在规划控制层面的主要研究问题；其次介绍了该领域的常见方法和最新进展，并分析了不同方法的优缺点和潜在的改进思路；接着介绍了本文所提方法的框架，总结了文章的核心贡献；最后对本文的文章结构进行了梳理。

第二章是对预备知识进行了介绍，主要包括导航势函数的定义与性质、微分同胚映射与谐势场、线性时序逻辑与任务自动机等几个重要知识；最后对本文所要解决的问题进行了数学上的描述。

第三章介绍了如何基于谐势场构建双层导航树。具体来说，主要包括基于方圆形环境建模、导航势函数的构建、势场的定向旋转变换、势场跟踪控制器的设计、双层导航树的构建等几个方面进行了循序渐进的介绍。

第四章介绍了在线执行期间，随着环境的变化，即障碍物数量的增加，对初始构建好的规划框架进行迭代和更新。主要包括导航势函数的增量式迭代计算、双层导航树的局部更新和任务规划的渐进收敛这三个层面。

第五章对本文所提框架进行了总结，描述了初始化构建和在线更新过程中具体实现流程，并对其复杂度进行了分析。

第六章是仿真与实验部分。首先对机器人模型、环境和任务进行了描述，然后对仿真实验的结果进行了叙述和分析，紧接着将所提方法与一些前沿的方法进行了对比，最后进行了硬件实验，证明了所提方法在现实场景中应用的可行性。

第二章 预备知识与问题描述

2.1 预备知识

在详细介绍本研究所提方法之前，本文对一些必要的预备知识进行了补充，包括导航势函数的定义与性质、微分同胚映射与谐波势场、线性时序逻辑与Büchi自动机这三个方面。

2.1.1 导航势函数的定义与性质

导航势函数最初被定义为解析的莫尔斯函数（Morse Function），详细可见文献[36]。为了简单起见，本文放宽了对莫尔斯函数的一些限制条件，给出了一种更直观的定义：

定义 1. 给定机器人的工作空间 $\mathcal{W}$ 和全局目标点 q_G ，函数 $\varphi_{NF} : \mathcal{W} \rightarrow [0, \mu]$ ($\mu \geq 1$)，是工作空间 $\mathcal{W}$ 上的导航势函数，当且仅当：

- 1) φ_{NF} 在工作空间 $\mathcal{W}$ 上是至少二阶光滑 (C^2 smooth) 的；
- 2) φ_{NF} 在工作空间 $\mathcal{W}$ 中只有唯一的全局最小值 $q_G \in \mathcal{W}$ ；
- 3) φ_{NF} 在工作空间 $\mathcal{W}$ 中除 q_G 外的所有临界点 (critical points) 都是孤立的鞍点 (isolated saddle points)，并且它们吸引域的测度都为零。
- 4) φ_{NF} 在工作空间的边界上 $\partial\mathcal{W}$ 具有统一的最大值 μ ，并且在 $\mathcal{W}$ 上是可接受的 (admissible)，即满足特定的边界条件。

当机器人在未知环境中运动时，上述导航势函数需要根据环境的改变进行更新，即每当新的障碍物被机器人检测到并添加到工作空间中时，要对 φ_{NF} 进行更新。因此，为确保机器人运动的安全性，每当环境更新时必须调整障碍边界上的导航势函数的最大值。即将上述函数的定义修正为 $\varphi_{NF} : \mathcal{W} \rightarrow [0, \mu(t)]$ ，其中 $\mu(t) \geq 1$ 是与时间 t 相关的分段常数。具体的更新策略可见文章后续章节。

2.1.2 微分同胚映射与谐波势场

考虑一个紧致连通的平面圆形空间 $\mathcal{M} \subset \mathbb{R}^2$ ，它由一个中心为 q_0 ，半径为 ρ_0 外部圆形工作空间，记作 $O_0 = \{q \in \mathbb{R}^2 : \|q - q_0\|^2 - \rho_0^2 \leq 0\}$ 和 M 个不相交的中心分别为 q_i ，半径为 ρ_i 的内部圆形障碍物，记作 $O_i = \{q \in \mathbb{R}^2 : \|q - q_i\|^2 - \rho_i^2 \leq 0\}$, $i = 1, \dots, M$ 组成。在圆形空间内部，含有一个全局目标点，记作 q_G 。通过引入一个微分同胚映射，可以将这个圆形空间变换成一个包含 M 个点障碍物 $P_i \in \mathbb{R}^2$ 的无界点空间，记作

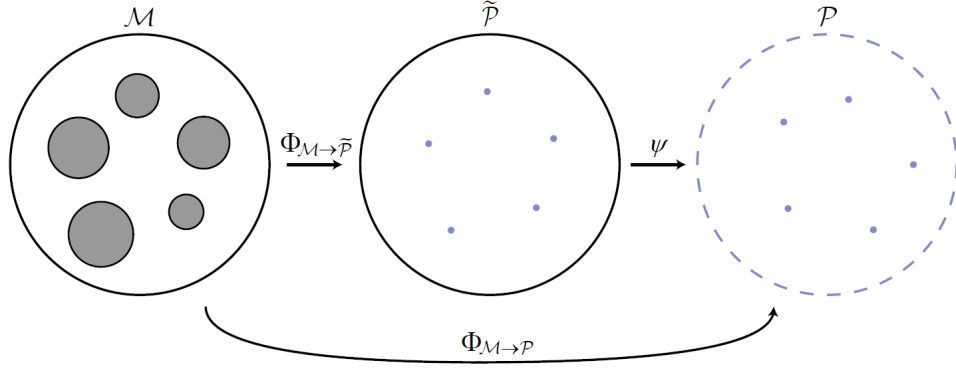


图 2.1 微分同胚映射的示意图。变换 $\Phi_{\mathcal{M} \rightarrow \tilde{\mathcal{P}}}$ 将平面圆形空间 $\mathcal{M}$ 先变换成有界点空间 $\tilde{\mathcal{P}}$ 再进一步变换成无界点空间 $\mathcal{P}$ 。

$\mathcal{P} = \mathbb{R}^2 - \{P_1, \dots, P_M\}$, 如图 2.1 所示。详细的变换过程可见文献 [18]。特别地, 给定圆形空间 $\mathcal{M}$, 相应的微分同胚映射定义如下:

$$\begin{aligned}\Phi_{\mathcal{M} \rightarrow \mathcal{P}}(q) &\triangleq \psi \circ \Phi_{\mathcal{M} \rightarrow \tilde{\mathcal{P}}}(q), \\ \Phi_{\mathcal{M} \rightarrow \tilde{\mathcal{P}}}(q) &\triangleq \text{id}(q) + \sum_{i=1}^M (1 - s_\delta(q, O_i))(q_i - q), \\ \psi(\tilde{q}) &\triangleq \frac{\rho_0}{\rho_0 - \|\tilde{q} - q_0\|}(\tilde{q} - q_0) + q_0,\end{aligned}\tag{2.1}$$

其中 $\Phi_{\mathcal{M} \rightarrow \tilde{\mathcal{P}}}(q)$ 先将圆形空间映射成一个有界点空间, 记作 $\tilde{\mathcal{P}} = O_0 - \{\tilde{P}_1, \dots, \tilde{P}_M\}$ 。该有界点空间包含 M 个点形障碍物 $\tilde{P}_i$, $i = 1, \dots, M$, 其通过变换 $\tilde{P}_i = \Phi_{\mathcal{M} \rightarrow \tilde{\mathcal{P}}}(q_i)$ 得到。公式 (2.1) 中的 $s_\delta(q, O_i)$ 函数是对障碍物 O_i 的半径放缩变换。更确切地, $s_\delta(q, O_i)$ 由放缩函数 $\eta_\delta(x)$ 、切换函数 $\sigma(x)$ 和平滑函数 $b_i(x)$ 复合得到, 即 $s_\delta(q, O_i) = \eta_\delta \circ \sigma \circ b_i(q)$ 。关于这些函数的详细定义可以参考文献 [18]。更进一步, 为了在空间中定义无限域的谐函数, 需要通过变换 $\psi(\tilde{q})$ 将有界点空间 $\tilde{\mathcal{P}}$ 映射成无界点空间 $\mathcal{P}$ 。该变换将 $\tilde{\mathcal{P}}$ 的外部边界映射到无穷远处, 相应地, 将内部点障碍物映射成新的点障碍物, 其对应关系可表示为 $P_i = \psi(\tilde{P}_i)$, $i = 1, \dots, M$ 。

给定上述推导出的点空间 $\mathcal{P}$, 可以借助谐函数在其中定义出相应的谐势能 [9, 37], 具体的定义如下:

定义 2. 在点空间中的谐势能定义为从点空间到实数域上的映射 $\phi_{\mathcal{P}} : \mathcal{P} \rightarrow \mathbb{R}^+$:

$$\phi_{\mathcal{P}}(x) \triangleq \phi(x, P_G) - \frac{1}{K} \sum_{i=1}^M \phi(x, P_i),\tag{2.2}$$

其中 $\phi : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ 为对数函数, 即 $\phi(x, q) = \ln(\|x - q\|^2)$; $\phi(x, P_d)$ 是与目标点 $P_G = \Phi_{\mathcal{M} \rightarrow \mathcal{P}}(q_G) \in \mathbb{R}^2$ 相关的势函数, 而 $\phi(x, P_i)$ 是与点障碍物 $P_i \in \mathbb{R}^2$ 相关的势函数; $K > 1$ 是设计参数。 ■

此外，可以采用逻辑函数（logistic function）

$$\sigma(s) = \mu \cdot \frac{e^s}{1 + e^s} \quad (2.3)$$

将无界的势函数 $\phi_P(x)$ 转化到有界区间 $[0, \mu]$ 上，其中 $\mu \geq 1$ 。转化结果如下：

$$\sigma \circ \phi_P(x) = \mu \cdot \frac{\|x - P_G\|^2}{\|x - P_G\|^2 + (\prod_{i=1}^M \|x - P_i\|^2)^{\frac{1}{K}}}, \quad K, \mu \geq 1. \quad (2.4)$$

值得注意的是，函数 $\sigma \circ \phi_P(x)$ 在目标点处的取值为 $\sigma \circ \phi_P(P_G) = 0$ ，而在点障碍物处的取值为 $\sigma \circ \phi_P(P_i) = 1$ ， $i = 1, \dots, M$ 。符合在定义 1 中描述的导航势函数的边界条件。关于导航势函数的详细推导及其性质分析见文章后续章节。

2.1.3 线性时序逻辑与Büchi自动机

线性时序逻辑（Linear Temporal Logic, LTL）是一种用于描述时间约束的逻辑形式，规范了某个系统必须遵守的时序性质，常用于形式化验证、任务规划和自主控制等领域。线性时序逻辑公式由一组原子命题（AP）加上几个布尔和时态运算符组成，其中原子命题是布尔变量，可以为真或为假 [11]。特别地，线性时序逻辑的语法，如文献 [11] 中所定义，可以表示为： $\varphi \triangleq \top \mid p \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid \bigcirc \varphi \mid \varphi_1 \cup \varphi_2$ ，其中 $\top \triangleq \text{True}$ ， $p \in AP$ ， $\bigcirc$ （下一步）， $\cup$ （直到）和 $\perp \triangleq \neg \top$ 。为了简洁起见，此处省略了其他运算符的推导，例如 $\square$ （始终如此）， $\diamond$ （最终）， $\Rightarrow$ （蕴含）等，对于线性时序逻辑的语义和语法的完整描述可以参考文献 [11]。此外，给定一个线性时序逻辑公式 φ ，都存在一个对应的非确定性Büchi自动机（Non-deterministic Büchi Automaton, NBA），其定义如下：

定义 3. 非确定性Büchi自动机定义为一个五元组： $\mathcal{A} = (S, S_0, \Sigma, \delta, S_F)$ ，其中 S 是状态集合； $S_0 \subseteq S$ 是初始状态集合； $\Sigma = AP$ 是给定的字母表； $\delta : S \times \Sigma \rightarrow 2^S$ 是转移关系； $S_F \subseteq S$ 是正在接受的状态集合。 ■

此外，由字母表 2^{AP} 组成的无限字符串 w 可以被定义为一个无限序列 $W = \sigma_1 \sigma_2 \dots, \sigma_i \in 2^{AP}$ 。公式 φ 的语言定义为满足公式 φ 的所有字的集合，即 $\mathcal{L} = \text{Words}(\varphi) = \{W \mid W \models \varphi\}$ ，其中 $\models$ 是需要满足的关系。此外， w 在自动机 $\mathcal{A}$ 中运行的结果定义为一个无限序列 $\rho = s_0 s_1 s_2 \dots$ ，其中 $s_0 \in S_0$ 且 $s_i \in S$ ， $s_{i+1} \in \delta(s_i, \sigma_i)$ 对于所有的 $i \geq 0$ 成立。对于一次运行，如果 $\inf(\rho) \cap S_F \neq \emptyset$ 成立，则称这个运行是可接受的（accepting），其中 $\inf(\rho)$ 是在 ρ 中无限次出现的状态集合。一般来说，一次可接受的运行能用前缀-后缀结构（prefix-suffix structure）表示，其中前缀从初始状态开始，以接受状态结束；后缀是一个包含相同接受状态的循环路径。通常情况下，自动机 $\mathcal{A}$ 的大小相对于公式 φ 的长度来说是双指数级别的。

2.2 问题描述

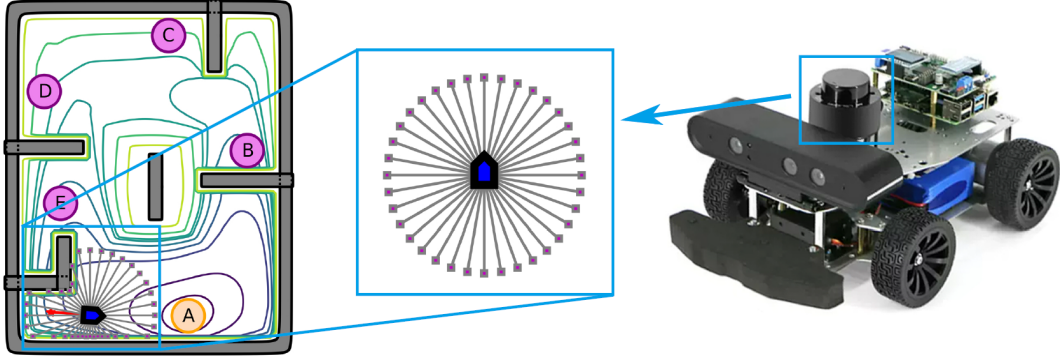


图 2.2 机器人模型。左图为在仿真环境中机器人通过传感器感知环境中的障碍物，将障碍物的边界返回成2D点云；中间的图为放大后的完整机器人模型；右图为带有激光雷达的机器人硬件模块。

如图 2.2所示，考虑一个自主移动机器人，它占据半径为 $r_r > 0$ 的圆形区域，并遵循如下运动学模型：

$$\begin{aligned}\dot{x} &= v \cos(\theta); \\ \dot{y} &= v \sin(\theta); \\ \dot{\theta} &= \omega;\end{aligned}\tag{2.5}$$

其中 $q = (x, y) \in \mathcal{W}$ 是机器人的坐标位置， $\theta \in [-\pi, \pi]$ 表示机器人的朝向， (v, ω) 是线速度 v 和角速度 ω 的控制输入。上述机器人在一个紧致且连通的工作空间 $\mathcal{W}_0 \subset \mathbb{R}^2$ 运动， $\mathcal{W}_0$ 内含有 M 个不相交的障碍物 $\mathcal{O}_i$ ，满足 $\mathcal{O}_i \subset \mathcal{W}_0$ 并且对于任意 $i, j \in 1, \dots, M$ ，有 $\mathcal{O}_i \cap \mathcal{O}_j = \emptyset$ ， $i \neq j$ 。因此，机器人运动的可行空间可以表示为 $\mathcal{W} \triangleq \mathcal{W}_0 - \bigcup_{i=1}^M \mathcal{O}_i$ 。

初始时刻 $t = 0$ 时，机器人仅知道工作空间的一部分，即只知道工作空间的外部边界和一些内部障碍物。从任意的初始状态 (q_0, θ_0) 开始，机器人可以在工作区内运动并通过传感器检测到更多的障碍物，该感知过程通过一个有限探测范围的传感器进行建模，具体表示如下：

$$S(q) \triangleq \{\hat{q} \in \mathcal{W} \mid (\hat{q} \in \mathcal{D}_{r_s}(q)) \wedge (\mathcal{L}(q, \hat{q}) \subset \mathcal{W})\},\tag{2.6}$$

其中 $S(q)$ 是机器人在位置 $q \in \mathcal{W}$ 处观察到的边界点集 $\hat{q}$ ， $\mathcal{D}_{r_s}(q, \hat{q})$ 是以 q 为圆心、 r_s 为半径的圆盘， $\mathcal{L}(q, \hat{q})$ 是连接 q 和 $\hat{q}$ 的直线。该传感器模型模拟了一个360°激光雷达传感器模型，它会返回在机器人感知范围内的任何障碍物表面的2D点云，如图 2.2所示，相同的方法也在文献 [10]中所使用。

最后，在工作空间 $\mathcal{W}$ 中存在一组相互之间不重叠的感兴趣区域(regions of interest) $g_n \subset \mathcal{W}$ ， $n = 1, \dots, N$ 。这些感兴趣区域可以表示机器人需要到达的任务点，与其相关的原子命题可以记为 $G = g_n$ ，表示机器人在区域 g_n 内，即 $q \in g_n$ 。因此，对于移动机器

人的任务规划可以表示为在命题 G 上的线性时序逻辑公式，即 $\varphi = LTL(G)$ ，相关的语法遵循在 2.1.3 中定义的符号逻辑。对于给定的机器人轨迹 $\mathbf{q}$ ，其运动路径可以由它随时间变化而与感兴趣区域相交的序列表示，即

$$\omega(\mathbf{q}) = g_{\ell_1} g_{\ell_2} \cdots$$

，其中 $g_{\ell_k} \in G$ ， $q(t_k) \in g_{\ell_k}$ ，对于某一时刻 $0 \geq t_k \leq t_{k+1}$ ， $k \in \mathbb{Z}$ 。因此，本文的目标是为机器人设计一种在线规划和控制的策略，使得机器人从初始姿态 (q_0, θ_0) 开始，运动生成的轨迹 $\omega(\mathbf{q})$ 满足给定任务 φ 的约束，同时避免与所有障碍物发生碰撞。

注 1. 虽然上述问题的表述非常标准，并已有了一些相关的工作，但本文主要突出以下三个方面，引文在这些方面的问题尚未得到很好的解决：(i) 有效地利用微分同胚映射处理重叠的复杂障碍物，以构造更普遍的机器人工作环境；(ii) 随着机器人不断探测到新的障碍物，采用迭代的方法更新导航势函数，提高算法的在线计算性能；(iii) 将导航势函数和基于搜索的方法相结合，以改善生成轨迹的平滑度和最优性。 ■

第三章 基于谐势场的双层导航树构建

如第一章中的图 1.2所示，本文所提出的方法是一个混合式的规划控制框架，采用双层导航树（DLNT）结构，即将基于谐势场的导航势函数和基于任务自动机的方向感知搜索树相结合。初始时刻，在部分已知的工作空间 $\mathcal{W}$ 中，根据初始任务自动机的引导构建定向搜索树，并使用具有方向调节的谐势场进行局部导航，从而根据导航控制器估算控制代价。随后，在在线执行期间，随着机器人探索空间并检测到更多的障碍物，使用在线更新的方案来逐步迭代计算谐势场并相应地调整定向搜索树，从而进一步提高规划的效率并确保导航的安全性。

本节将详细介绍上述控制规划框架下初始规划和控制的构建方法。首先，根据一种特殊形状的平面几何图形构建障碍物环境模型，并分析了该几何图形的性质和用法。其次，根据障碍物模型构建导航势场，并对势场进行定向旋转变换得到一个定向势场，用于控制机器人沿着特定的方向运动到目标点。接着，设计了一款非线性势场跟踪控制器，用于驱动非完整约束的机器人，并证明了该控制器的安全性和收敛性。最后，根据控制器的特点引入了代价函数，用于估计机器人在离散路径点之间的控制代价，并在此基础上构建以任务自动机的引导的双层导航树，从中得出初始时刻的离散规划结果。

3.1 障碍物环境的建模

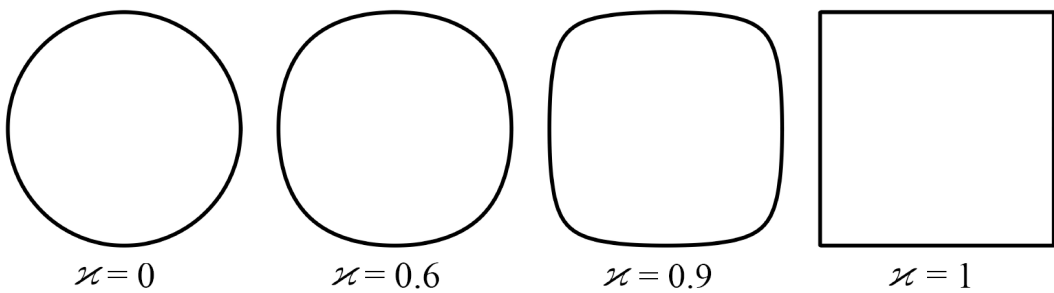


图 3.1 方圆形的形状随参数 κ 的变化。

如公式 (2.6)所述，机器人所携带的激光传感器返回一个由感知范围内任何障碍物表面上的点组成的2D点云。更具体地，用 $\mathbf{D}_t = \{d_j\}$ 表示已经从局部坐标变换到全局坐标的一组二维点云，其中 $d_j \in \partial\mathcal{W}$ 。首先，这些点云数据可以通过某种算法被划分为 K 个簇，分别代表 K 个独立的障碍物，即 $\mathbf{D}_t = \{\mathbf{D}_{t,k}\}$ ，例如，可以通过检查相邻点之间的相对距离和斜率变化来实现障碍物的分离。确切的分离方法和参数选取取决于

激光雷达的规格和类型。通过上述分离算法得到 K 个独立的障碍物点云簇后，再将每个簇拟合到特定的二维障碍物边界上。

在大多数现实环境中，可以使用类似长方形的几何图形模拟简单障碍物，并且可以将这些图形相互重叠来模拟复杂障碍物。在某些简单环境中，比如柱子和家具等独立的障碍物可以用矩形和圆形进行建模；其他更复杂的环境，比如有走廊和房间的室内环境，可以用矩形的重叠组合进行建模。在导航势函数法中，最常见的障碍物是独立圆形和矩形，因为其形式较为简单，建模相对容易，如文献 [9, 38] 中使用的方法。在文献 [28] 中，作者引入“方圆形”来对现实中的障碍物环境进行建模，通过相互重叠的方圆形来构造复杂场景。方圆形（squircle）是一种介于方形和圆形之间的光滑几何图形，如图 3.1 所示。本章首先介绍二维平面上的单位方圆形的隐式表达式，并通过相似变换得到一般形状下的表达式，最终推导了一种从方圆形内部中心点到边界点的半径长度的计算方法，并证明了其性质。

方圆形最初的解析表达式由 Fernandez-Guasti 引入，因此又被称为 Fernandez-Guasti Squircle，其最初的解析表达式为：

$$\frac{x^2 + y^2}{r^2} - \varkappa^2 \frac{x^2 y^2}{r^4} = 1, \quad (3.1)$$

其中 $0 < \varkappa < 1$ 是正常数， r 是方圆形的“半径”。 $0 < \varkappa < 1$ 又被称为方形参数，当 $\varkappa = 0$ 时，上述方程变成半径为 r 的圆形；当 $\varkappa = 1$ 时，上述方程变成边长为 $2r$ 的正方形；当 $\varkappa = 1$ 在 0 到 1 之间时，方程是一条平滑曲线，并介于圆形和方形之间。然而，上述 Fernandez-Guasti 方圆形的表达式包含两组曲线，从而不能直接用作矩形障碍物的隐式表示。根据隐函数表示的定义，障碍物的隐函数只在障碍物的边界上等于零。因此，在文献 [28] 中，作者给出了一个修正后的方圆形表达式，保留了公式 (3.1) 的内部曲线并消除了四个角落处的伪曲线。具体来说，单位方圆形的隐式解析式由如下定义给出：

定义 4. 在二维平面 $\mathbb{R}^2$ 上，以原点为中心的单位方圆形的隐式表达式定义为：

$$\beta(x, y) \triangleq \frac{x^2 + y^2 + \sqrt{x^4 + y^4 - (2 - 4\varkappa^2)x^2 y^2}}{2} - 1, \quad (3.2)$$

其中 $\varkappa \in (0, 1)$ 是正实数。 ■

对于方程 3.2，当 $\beta(x, y) < 0$ 时，点 (x, y) 在单位方圆形内部；当 $\beta(x, y) = 0$ 时，点 (x, y) 恰好在单位方圆形的边界上；当 $\beta(x, y) > 0$ 时，点 (x, y) 在单位方圆形外部。因此满足障碍物隐函数的要求。根据上述定义，可以容易得到单位方圆形的解析表达式在 $\mathbb{R}^2 - \{0\}$ 上是光滑的，证明见如下引理。

引理 1. $\beta(x, y)$ 在集合 $\mathbb{R}^2 - \{0\}$ 上是光滑的。

证明. 对于 $\beta(x,y)$ 的 n 阶偏导数, 唯一可能不连续的项是根号内的部分, 即 $x^4+y^4-(2-4\kappa^2)x^2y^2$ 。注意到不等式 $0 < \kappa < 1$ 是严格成立的。如果 $x \neq 0$ 且 $y \neq 0$ 则有:

$$x^4+y^4-(2-4\kappa^2)x^2y^2 \geq x^4+y^4-2x^2y^2 = (x^2-y^2)^2.$$

因此根号内的部分是正定的, 这保证了 n 阶偏导数的存在性。即使原点 $(0,0)$ 处的导数是不存在的, 但这不会影响导航势函数的构建, 因为 $(0,0)$ 总是在障碍物的内部, 即总在自由空间的外部。

为了计算单位方圆形的射径长度 (length of rays), 即从几何中心到边界上某一点的半径长度, 可以将公式 (3.2)用向量形式表示出来。对于任意的向量 $q = (x,y) \in \mathbb{R}^2$, 将 $q = xe_1 + ye_2$ 带入公式 (3.2), 则可以得到如下方程:

$$\beta_{sc}(q) \triangleq \frac{q^2 + \sqrt{q^4 - 4\kappa^2 [(q^\top e_1)(q^\top e_2)]^2}}{2} - 1, \quad (3.3)$$

其中 $\kappa \in (0, 1)$ 是正实数; e_1 和 e_2 是 $\mathbb{R}^2$ 中的两个基向量。更进一步地, 将向量 q 写成极坐标的形式, 即 $q = r\hat{q}$, 其中 r 表示 q 的长度, $\hat{q}$ 表示 q 的方向向量。将 $q = r\hat{q}$ 带入公式 (3.3), 可以得到如下形式:

$$\beta_{sc}(r, \hat{q}) = r^2 \frac{1 + \sqrt{1 - 4\kappa^2 [(\hat{q}^\top e_1)(\hat{q}^\top e_2)]^2}}{2} - 1, \quad (3.4)$$

令 $\beta_{sc}(r, \hat{q}) = 0$, 则可以得到单位方圆形从几何中心到边界点的射径长度。即只要给定从几何中心出发的某一方向向量 $\hat{q}$, 方圆形的射径长度计算如下:

$$\rho_{sc}(\hat{q}) = \sqrt{\frac{2}{1 + \sqrt{1 - 4\kappa^2 [(\hat{q}^\top e_1)(\hat{q}^\top e_2)]^2}}} \quad (3.5)$$

。

此外, 对于非单位的方圆形, 可以通过对单位方圆形进行相似变换 (放缩、旋转、平移) 得到其解析表达式和射径长度。具体地, 对于2D平面空间, 引入如下放缩矩阵 $\mathbf{A}$ 、旋转矩阵 $\mathbf{R}$ 和平移变换向量 l :

$$\mathbf{A} \triangleq \begin{bmatrix} a^1 & 0 \\ 0 & a^2 \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \quad \mathbf{R} \triangleq \begin{bmatrix} r^{11} & r^{12} \\ r^{21} & r^{22} \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \quad l \triangleq \begin{bmatrix} l^1 \\ l^2 \end{bmatrix} \in \mathbb{R}^2 \quad (3.6)$$

然后, 通过以上三种变换对方圆形进行坐标变换:

$$q = \mathbf{A}^{-1} \mathbf{R}^{-1} (q' - l), \quad (3.7)$$

其中 q' 是变换后的新坐标。因此, 将变换后的坐标带入公式 (3.3)就可以得到非单位方

圆形的表达式。对于非单位圆形的射径长度，旋转变换和平移变换并不会对其产生影响，因为圆形的尺寸并不会随着旋转和平移而发生变化。因此，只有放缩变换会改变非单位圆形的射径长度，改变后的长度计算公式如下：

$$\rho'(\hat{q}) \triangleq \frac{1}{\|\mathbf{A}^{-1}\hat{q}\|} \rho_{sc} \left(\frac{\mathbf{A}^{-1}\hat{q}}{\|\mathbf{A}^{-1}\hat{q}\|} \right), \quad (3.8)$$

其中 $\hat{q} = \frac{q - q_i}{\|q - q_i\|}$ 是向量 $q - q_i$ 归一化后的单位向量， $q_i = l^T$ 是圆形的几何中心的坐标。

根据以上推导出的圆形的表达式，可以对现实环境中的障碍物进行建模。总的来说，根据机器人的传感器检测得到的点云集合，可以通过最小二乘法（LSF）对圆形的边界进行拟合估计，从而得到障碍物的解析表达。因此，可以用 $\hat{\mathcal{O}}_t$ 表示在时间 t 检测到和拟合完成的障碍物的集合。

3.2 导航势函数的构建

给定初始时刻根据传感器和最小二乘法近似得到的障碍物集合 $\hat{\mathcal{O}}_0$ ，导航势函数 $\phi_{NF}(q)$ 的构建通过三步关键的微分同胚变换而得到：(i) $\Phi_{\mathcal{F} \rightarrow \mathcal{S}}$ 通过“消除变换”将丛林空间 $\mathcal{F}$ 变换为星形空间 $\mathcal{S}$ ；(ii) $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}$ 将星形空间 $\mathcal{S}$ 变换为圆形世界 $\mathcal{M}$ ；(iii) $\Phi_{\mathcal{M} \rightarrow \mathcal{P}}$ 将圆形空间 $\mathcal{M}$ 变换为点空间 $\mathcal{P}$ 。给定上述三步微分同胚映射，完整导航势函数表示如下：

$$\phi_{NF}(q) = \sigma \circ \phi_{\mathcal{P}} \circ \Phi_{\mathcal{M} \rightarrow \mathcal{P}} \circ \Phi_{\mathcal{S} \rightarrow \mathcal{M}} \circ \Phi_{\mathcal{F} \rightarrow \mathcal{S}}(q), \quad (3.9)$$

其中 $\sigma \circ \phi_{\mathcal{P}}$ 是预备知识章节中的公式 (2.2)；变换 $\Phi_{\mathcal{M} \rightarrow \mathcal{P}}$ 由预备知识章节中的公式 (2.1) 给出。本小节将详细描述两个不平凡且关键的变换 $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}$ 和 $\Phi_{\mathcal{F} \rightarrow \mathcal{S}}$ 。

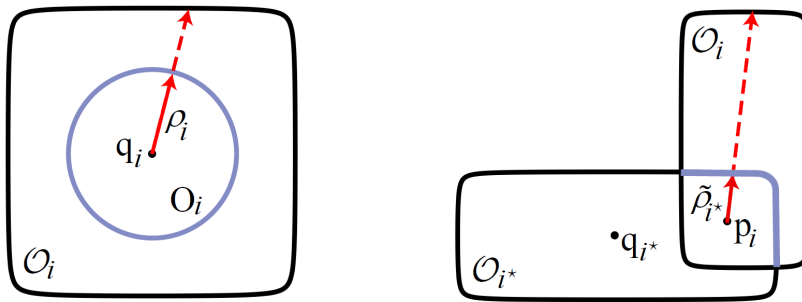


图 3.2 径向放缩变换。左图为星形到圆形的径向放缩变换，即将星形的边界变换成圆形边界；右图为消除变换的径向放缩变换，即将子障碍物的边界变换到父障碍物的部分边界上。

3.2.1 星形空间到圆形空间的微分同胚映射

对于一个二维平面图形，该图形是星状图形，当且仅当从它的某一个中心出发的任意射线只与该图形的边界有一个交点。数学上可以表述为：如果 $\beta(q)$ 是一个平面几

何图形的光滑解析函数，也即 $\mathcal{O} \triangleq \{q \in \mathbb{R}^2 : \beta(q) \leq 0\}$ ，那么 $\mathcal{O}$ 是星形的，当且仅当存在一个常数 $\Delta > 0$ ，使得 $\nabla\beta(q) \cdot (q - c) \geq 2\Delta$ 对任意 $q \in \partial\mathcal{O}$ ，其中 c 为几何图形的中心。

考虑一个有界的二维工作空间 $\mathcal{W}_0 = \{q \in \mathbb{R}^2 : \beta_0(q) \leq 0\}$ ，其中包含 M 个内部障碍物 $\mathcal{O}_i = \{q \in \mathbb{R}^2 : \beta_i(q) \leq 0\}$ ， $i = 0, 1, \dots, M$ 。如果工作空间外部的边界 $\mathcal{W}_0$ 和所有的内部障碍物 $\mathcal{O}_i$ 都是星形的 (Star World)，那么该空间可以称为星形空间。值得注意的是，方形是严格的星状图形 [28]，因此，由它构成的空间为星形空间，记作：

$$\mathcal{S} \triangleq \mathcal{W}_0 - \bigcup_{i=1}^M \mathcal{O}_i. \quad (3.10)$$

给定上述星状图形，可以通过径向放缩变换 [38] 将其映射成一个圆形，如图 3.2 所示。具体而言，给定星形空间 $\mathcal{S}$ ，可以借助变形因子 (deforming factor)，解析切换 (analytic switch)、放缩映射 (translated scaling map) 将其通过微分同胚映射变换到对应的圆形空间 (Sphere World) 中。具体定义如下：

定义 5. 给定星形空间 $\mathcal{S}$ ，其外部工作空间 $\mathcal{W}_0$ 和内部星形障碍物 $\mathcal{O}_i$ ， $i = 0, 1, \dots, M$ ，的变形因子定义为如下函数：

$$v_0(q) \triangleq \rho_0 \frac{1 - \beta_0(q)}{\|q - q_0\|}, \quad v_i(q) \triangleq \rho_i \frac{1 + \beta_i(q)}{\|q - q_i\|}, \quad i = 1, 2, \dots, M, \quad (3.11)$$

其中 q_0 和 q_i 分别是外部边界和内部障碍物的几何中心， ρ_0 和 ρ_i 是相应的变换后的圆形的半径。 ■

定义 6. 给定星形空间 $\mathcal{S}$ ，其外部工作空间 $\mathcal{W}_0$ 和内部星形障碍物 $\mathcal{O}_i$ ， $i = 0, 1, \dots, M$ ，的解析切换定义为如下函数：

$$\sigma_i(q) \triangleq \frac{\gamma_G(q) \bar{\beta}_i(q)}{\gamma_G(q) \bar{\beta}_i(q) + \lambda \beta_i(q)}, \quad i = 0, 1, \dots, M, \quad (3.12)$$

其中 λ 是正实数， $\gamma_G(q) \triangleq \|q - q_G\|^2$ 是与目标点相关的函数， $\bar{\beta}_i(q) \triangleq \prod_{j=0, j \neq i}^M \beta_j(q)$ 为省略乘积 (omitted product)。 ■

定义 7. 给定星形空间 $\mathcal{S}$ ，其外部工作空间 $\mathcal{W}_0$ 和内部星形障碍物 $\mathcal{O}_i$ ， $i = 0, 1, \dots, M$ ，的放缩映射定义为如下函数：

$$T_i(q) \triangleq v_i(q) (q - q_i) + q_i, \quad i = 0, 1, \dots, M, \quad (3.13)$$

其中 $v_i(q)$ 为变形因子， q_i 为几何中心。 ■

对于上述的星形空间及其对应的圆形空间，需要满足两个条件，即放置条件 (placement condition) 和包含条件 (containment condition)，具体定义如下：

定义 8. 给定一个星形空间 $\mathcal{S}$ ，与之相对应的圆形空间 $\mathcal{M}$ 满足放置条件，当且仅当：

$$q'_i = q_i, \quad i = 0, 1, \dots, M \quad \text{并且} \quad q'_G = q_G, \quad (3.14)$$

其中 q'_i 和 q_i 分别为圆形空间和星形空间中的障碍物的中心， q'_G 和 q_G 分别是在圆形空间和星形空间中的目标点。 ■

定义 9. 给定一个星形空间 $\mathcal{S}$ ，与之相对应的圆形空间 $\mathcal{M}$ 满足包含条件，当且仅当：

$$v_i(q) \leq 1 \quad \forall q \in \mathcal{O}_i(\varepsilon), \quad i = 1, \dots, M, \quad (3.15)$$

并且

$$v_0(q) \geq 1 \quad \forall q \in \mathcal{W}_0(\varepsilon), \quad (3.16)$$

其中 ε 是一个充分小的正实数，满足 $\mathcal{W}_0(\varepsilon) \in \mathcal{S} \cup \partial \mathcal{W}_0$ 并且 $\mathcal{O}_i(\varepsilon) \in \mathcal{S} \cup \partial \mathcal{O}_i$ 。 ■

最后，根据上述的关键变量的定义和需要满足的条件，可以构建从星形空间到圆形空间的微分同胚映射 $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}$ ，具体定义如下：

定义 10. 给定一个星形空间 $\mathcal{S}$ 以及与其对应的圆形空间 $\mathcal{M}$ ，从 $\mathcal{S}$ 到 $\mathcal{M}$ 的变换可以定义为 $\Phi_{\mathcal{S} \rightarrow \mathcal{M}} : \mathcal{S} \rightarrow \mathcal{M}$ ，满足：

$$\Phi_{\mathcal{S} \rightarrow \mathcal{M}}(q) \triangleq \left(1 - \sum_{i=0}^M \sigma_i(q)\right) \text{id}(q) + \sum_{i=0}^M \sigma_i(q) T_i(q), \quad (3.17)$$

其中 $\text{id}(q)$ 为恒等变换， $\sigma_i(q)$ 为解析切换， $T_i(q)$ 为放缩映射， $i = 0, 1, \dots, M$ 。 ■

注 2. 关于变换 $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}(q)$ 的正确性，需要证明存在一个常数 $\Lambda > 0$ ，当 $\lambda > \Lambda$ 时，变换 $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}(q)$ 是微分同胚映射。具体来说，需要证明两个子问题：(i) 当 $q \in \mathcal{S} - \partial \mathcal{S}$ 时， $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}(q)$ 的雅克比矩阵是非奇异的；(ii) 在工作空间的边界 $\partial \mathcal{S}$ 上， $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}(q)$ 是双射。由于证明过程较为繁琐，并且不是本文的重点，此处省略。详细证明过程推荐读者参考文献 [28, 38]。 ■

3.2.2 丛林空间到星形空间的微分同胚映射

如文献 [9, 28, 38] 中所描述，除了不相交的星形空间外，还需要考虑由重叠的星形障碍物构成的工作空间，又称为丛林空间 (Forest World)。丛林空间由有限个障碍群组成，每个障碍群是由有限个星形障碍物的并集组合形成被称之为星形树 (Tree of Stars)，其邻接图满足树形结构。由于无法直接将重叠的障碍群作为一个整体的障碍物进行微分同胚映射，因此每棵树都必须通过连续地“消除变换 (purging transformation)”将他们的子障碍物放缩到父障碍物里面，最后变成一个根障碍物 (root

obstacle)。具体来说，丛林空间可以被描述为：

$$\mathcal{F} = \mathcal{W}_0 - \bigcup_{n=1}^N \mathcal{T}_n, \quad (3.18)$$

其中 $\mathcal{T}_n$ 是 N 棵星形树中的第 n 棵。每棵星形树都有一个唯一的根障碍物，并且其所有的障碍物按父子偏序关系进行排列，这种排列方式有如下性质：(i) 如果在星形树的偏序关系中存在两个星形障碍物 $\mathcal{O}_j$ 和 $\mathcal{O}_m$ 满足 $\mathcal{O}_j \prec \mathcal{O}_m$ ，那么不可能存在第三个障碍物 $\mathcal{O}_k$ 使得 $\mathcal{O}_j \prec \mathcal{O}_k \prec \mathcal{O}_m$ （这种关系又记作 $\mathcal{O}_j \preceq \mathcal{O}_m$ ），这个性质强调父障碍物 $\mathcal{O}_m$ 的中心必须在子障碍物 $\mathcal{O}_j$ 里面；(ii) 对于星形树中任意一个障碍物 $\mathcal{O}_j$ ，有且只有一个障碍物 $\mathcal{O}_m$ 满足关系 $\mathcal{O}_j \preceq \mathcal{O}_m$ 。图 3.3 给出了这样一个丛林空间的例子，其中每棵树的深度随着树中障碍物排列的层级不同而变化。不失一般性的，用 d_n 表示树 $\mathcal{T}_n$ 的深度，用 $\mathcal{L}$ 表示所有树中叶障碍的下标集合，用 $\mathcal{I}$ 表示在丛林空间 $\mathcal{F}$ 中所有障碍物的下标集合。此外，星形树中的任何一个障碍物可以表示为 $\mathcal{O}_i \triangleq \{q \in \mathbb{R}^2 : \beta_i(q) \leq 0\}$ ，其中 $\beta_i(q)$ 是公式 (3.3) 中定义的方圆形的解析函数。作为一个重要的关系，障碍物 $\mathcal{O}_i$ 的“父亲”被表示为 $\mathcal{O}_{i^*}$ ，即满足 $\mathcal{O}_{i^*} \preceq \mathcal{O}_i$ 。为了在消除变换中计算方便，将父障碍物和子障碍物的中心点选为同一个点，记为：

$$p_i \in \mathcal{O}_i \cap \mathcal{O}_{i^*}. \quad (3.19)$$

给定上述的一个丛林空间 $\mathcal{F}$ ，从丛林空间 $\mathcal{F}$ 到星形空间 $\mathcal{S}$ 的微分同胚映射可以通过对

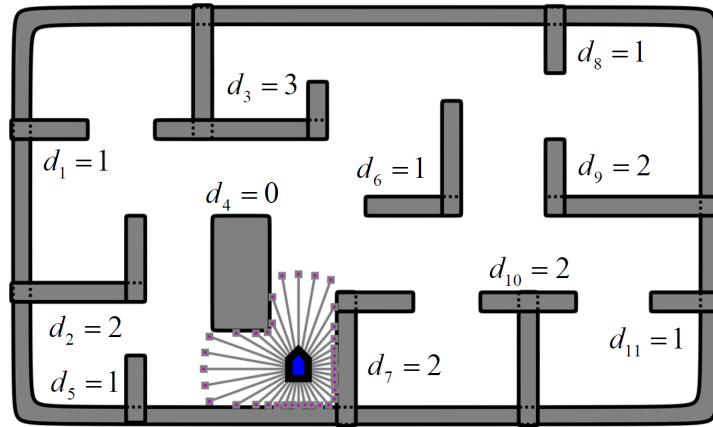


图 3.3 丛林空间的举例图。不同的星形树 $\mathcal{T}_n$ 有不同的深度 d_n 。

每棵树进行连续的消除变换来构建，具体表示为：

$$\Phi_{\mathcal{F} \rightarrow \mathcal{S}}(q) \triangleq \Phi_N \circ \cdots \circ \Phi_2 \circ \Phi_1(q), \quad (3.20)$$

其中 $\Phi_n(q)$, $n = 1, \dots, N$ 是对第 n 棵树 $\mathcal{T}_n$ 的叶障碍物的一系列复合消除变换, 它的具体形式如下:

$$\Phi_n(q) \triangleq f_{n,1} \circ f_{n,2} \cdots \circ f_{n,d_n}(q), \quad (3.21)$$

其中 $f_{n,i}(q)$, $i = 1, \dots, d_n$ 是对第 n 棵树的第 i 个障碍物的消除变换。具体而言, 这种消除变换 $f_{n,i}(q)$ 是通过径向放缩过程来构建的 [28], 即将子障碍物变换到其父障碍物里面, 具体变换过程如图 3.2 所示。

给定一个丛林空间 $\mathcal{F}$, 首先定义出消除某一个障碍物后的空间为:

$$\hat{\mathcal{F}} = \mathcal{F} \bigcup_{i \in \mathcal{L}} (\mathcal{O}_i - \mathcal{O}_{i^*}). \quad (3.22)$$

根据第 3.1 节中描述的方圆形特定的表达形式, 从父障碍物和子障碍物的共同中心 p_i 点到父障碍物 $\mathcal{O}_{i^*}$ 边界点的射径长度可以通过以下公式计算:

$$\tilde{\rho}_{i^*}(\hat{q}) \triangleq \frac{1}{\|\tilde{\mathbf{A}}_{i^*}^{-1} \hat{q}\|} \rho_{sc} \left(\frac{\tilde{\mathbf{A}}_{i^*}^{-1} \hat{q}}{\|\tilde{\mathbf{A}}_{i^*}^{-1} \hat{q}\|} \right), \quad (3.23)$$

其中 $\hat{q} = \frac{q - p_i}{\|q - p_i\|}$ 是对向量 $q - p_i$ 归一化后的单位向量, $\rho_{sc}(\cdot)$ 是从公式 (3.5) 得到的单位方圆形的射径长度, 而 $\tilde{\mathbf{A}}_{i^*} \in \mathbb{R}^{2 \times 2}$ 是与父障碍物相关的分段放缩矩阵, 它为对角矩阵, 且对角元素为:

$$\tilde{a}_{i^*}^\ell(\hat{q}) \triangleq a_{i^*}^\ell + \mathbf{sgn}(\hat{q}^\top e_\ell) [q_{i^*} - p_i]^\top e_\ell, \quad \ell = 1, 2, \quad (3.24)$$

其中 $a_{i^*}^\ell$ 是父障碍物从单位方圆形变换过来的放缩矩阵中的对角元素, q_{i^*} 是父障碍物的几何中心, e_ℓ 是 $\mathbb{R}^2$ 上的两个单位基向量, $\ell = 1, 2$ 。

由于 $\tilde{\rho}_{i^*}(q)$ 是通过分段对角矩阵 $\tilde{\mathbf{A}}_{i^*}$ 计算得来的, 因此需要证明它的光滑性, 详细证明见如下引理:

引理 2. $\tilde{\rho}_{i^*}(q)$ 在 $\mathcal{W}_0 - \mathcal{O}_i \cup \mathcal{O}_{i^*}$ 上是光滑的。

证明. $\tilde{\rho}_{i^*}(q)$ 的导数可以通过链式法则得到, 结果如下:

$$\nabla \tilde{\rho}_{i^*}(q) = \frac{\|q - p_i\|^2 \mathbf{I} - (q - p_i)(q - p_i)^\top}{\|q - p_i\|^3} \nabla \tilde{\rho}_{i^*}(\hat{q}),$$

其中, $\tilde{\rho}_{i^*}(\hat{q})$ 的导数简化为计算 $\rho_{sc}(q)$ 的导数和 $Z(\hat{q}) = \tilde{\mathbf{A}}_{i^*}^{-1} \hat{q}$ 的导数。 $\rho_{sc}(q)$ 的导数和光滑性可以直接从公式 (3.3) 中得到。由于 $\nabla a_{i^*}^\ell(\hat{q}) = \nabla (a_{i^*}^\ell + \mathbf{sgn}(\hat{q}^\top e_\ell) [q_{i^*} - p_i]^\top e_\ell) = 0$, 对 $\ell = 1, 2$ 恒成立, 那么 $Z(\hat{q})$ 的导数可以简化为:

$$\nabla Z(\hat{q}) = \tilde{\mathbf{A}}_{i^*}^{-1} \mathbf{I} - \tilde{\mathbf{A}}_{i^*}^{-1} [\nabla \tilde{\mathbf{A}}_{i^*}] \tilde{\mathbf{A}}_{i^*}^{-1} \hat{q} = \tilde{\mathbf{A}}_{i^*}^{-1}.$$

而且, 由于 $\|q - p_i\| > 0$ 对所有满足 $q \in \mathcal{W}_0 - \mathcal{O}_i \cup \mathcal{O}_{i^*}$ 的点都成立, 因此 $\tilde{\rho}_{i^*}(q)$ 的一阶导数存在且有意义。另一方面, $\nabla \tilde{\rho}_{i^*}(q)$ 是连续的因为它的所有复合函数都是连续的。因

此可以得到, $\tilde{\rho}_{i^*}(q)$ 至少是一阶光滑。更进一步地, 可以使用相同的方法得到更高阶的光滑性, 这里省略不赘述。

注 3. 与公式 (3.5) 相比, 公式 (3.23) 中的修改后的射径长度 $\tilde{\rho}_{i^*}(q)$ 是关键, 这是因为父障碍物与子障碍物的共同中心 p_i 不总是与父障碍物的几何中心 q_{i^*} 重合。该方法是受到了 [28] 中类似方法的启发, 该作者所使用的方法是在父障碍物和子障碍物之间引入了一个虚拟障碍物, 但是会导致障碍物一些信息的缺失, 从而影响导航势函数的性质。与原始方法相比, 本文提出了一种分段放缩矩阵来近似射径的长度, 并且所得到的射径公式可以被证明是光滑的, 因而能保留导航势函数的性质, 并且计算效率更高。 ■

随后, 在上述推导的基础上构建消除变换。与从星形空间到圆形空间的变换类似, 构建消除变换也包括三个核心变量, 分别为: 变形因子 (deforming factor), 解析切换 (analytic switch) 和放缩映射 (translated scaling map), 三者的具体定义如下:

定义 11. 给定丛林空间中的第 n 棵星形树 $\mathcal{T}_n$, 且其深度为 d_n , 则对于星形树上的第 i 个障碍物 $\mathcal{O}_i$ 的变形因子定义为如下函数:

$$v_i(q) = \tilde{\rho}_{i^*}(q) \frac{1 + \beta_i(q) \tilde{\kappa}_i(q)}{\|q - p_i\|}, \quad i = 1, 2, \dots, d_n, \quad (3.25)$$

其中 $\tilde{\rho}_{i^*}(q)$ 是其父障碍物的射径长度, p_i 是父障碍物和子障碍物的共同中心, 并且 $\tilde{\kappa}_i(q)$ 定义为:

$$\tilde{\kappa}_i(q) \triangleq \beta_{i^*}(q) + (\beta_i(q) - 2E_i) + \sqrt{\beta_{i^*}^2(q) + ((\beta_i(q) - 2E_i))^2},$$

其中 $E_i > 0$ 为几何常数, 满足 $\mathcal{O}_i(2E_i) \cap \mathcal{O}_j(2E_j) = \emptyset, \forall i, j \in \mathcal{I}$, 且 $i \neq j$ 。 ■

定义 12. 给定丛林空间中的第 n 棵星形树 $\mathcal{T}_n$, 且其深度为 d_n , 则对于星形树上的第 i 个障碍物 $\mathcal{O}_i$ 的解析切换定义为如下函数:

$$\sigma_i(q, \xi_i) \triangleq \frac{\gamma_G(q) \bar{\beta}_i(q)}{\gamma_G(q) \bar{\beta}_i(q) + \xi_i \beta_i(q)}, \quad i = 1, 2, \dots, d_n, \quad (3.26)$$

其中 $\xi_i > 0$ 是与 i 相关的正实数, $\gamma_G(q) \triangleq \|q - q_G\|^2$ 是与目标点相关的函数, $\bar{\beta}_i(q)$ 为省略乘积 (omitted product), 定义为:

$$\bar{\beta}_i(q) \triangleq \left(\prod_{j \in \mathcal{I} \setminus \{i, i^*\}} \beta_j(q) \right) \left(\prod_{j \in \mathcal{L} \setminus \{i\}} \beta_j(q) \right) \tilde{\beta}_i(q),$$

其中 $\tilde{\beta}_i(q)$ 的定义与 $\tilde{\kappa}_i(q)$ 类似:

$$\tilde{\beta}_i(q) \triangleq \beta_{i^*}(q) + (2E_i - \beta_i(q)) + \sqrt{\beta_{i^*}^2(q) + (2E_i - \beta_i(q))^2},$$

其中 $E_i > 0$ 为几何常数, 满足 $\mathcal{O}_i(2E_i) \cap \mathcal{O}_j(2E_j) = \emptyset, \forall i, j \in \mathcal{I}$, 且 $i \neq j$ 。 ■

定义 13. 给定丛林空间中的第 n 棵星形树 $\mathcal{T}_n$ ，且其深度为 d_n ，则对于星形树上的第 i 个障碍物 $\mathcal{O}_i$ 的放缩映射定义为如下函数：

$$T_i(q) \triangleq v_i(q)(q - p_i) + p_i, \quad i = 1, 2, \dots, d_n, \quad (3.27)$$

其中 $v_i(q)$ 为变形因子， p_i 为共同中心。 ■

最后，根据上述关键变量的定义，可以构建对丛林空间 $\mathcal{F}$ 中的第 n 棵星形树 $\mathcal{T}_n$ 上的第 i 个障碍物 $\mathcal{O}_i$ 的消除变换 $f_{n,i}(q)$ ，具体定义如下：

定义 14. 给定一个丛林空间 $\mathcal{F}$ ，其中的第 n 棵星形树 $\mathcal{T}_n$ 上的第 i 个叶障碍物为 $\mathcal{O}_i$ ，那么将该障碍物放缩到其父障碍物 $\mathcal{O}_{i^*}$ 内的变换，即从空间 $\mathcal{F}$ 到空间 $\hat{\mathcal{F}}$ 的变换 $f_{n,i}(q)$ 是一个连续的变换函数，定义为：

$$f_{n,i}(q) \triangleq (1 - \sigma_i(q, \xi_i)) \text{id}(q) + \sigma_i(q, \xi_i) T_i(q), \quad i = 1, 2, \dots, d_n \text{ 且 } n = 1, \dots, N, \quad (3.28)$$

其中 $\text{id}(q)$ 为恒等变换， $\sigma_i(q, \xi_i)$ 为解析切换， $T_i(q)$ 为放缩映射。 ■

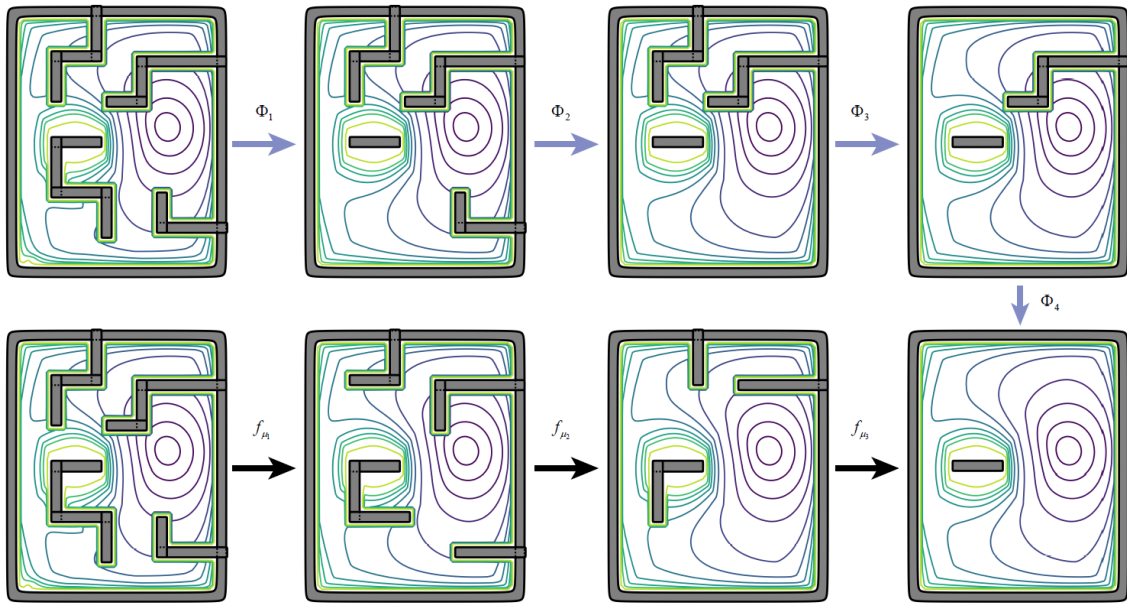


图 3.4 消除变换过程。蓝色箭头为本文提出的消除变换方法，即通过变换 $\Phi_i(q)$ 对每棵星形树的障碍物按顺序进行放缩，从而将整棵树变换到根障碍物里面；黑色箭头为文献 [28] 所采用的消除变换方法，即 $f_{\mu_i}(q)$ 每次将所有星形树的叶障碍物同时消除。

注 4. 在文献 [38] 中，作者首次提出了消除变换，提出的消除方法是通过布尔组合处理相互重叠的障碍物，因此仅适用于平面和抛物线类型的障碍物。在文献 [28] 中，作者提出了一种更简单的消除方法，用可变的射径长度代替固定的射径长度来构建径向放缩变换，从而提高了方法的可扩展性。然而，这些方法必须同时对所有叶障碍物进行

消除变换, 这不适用于障碍物一个接一个地添加到工作空间中的动态场景。因此, 在公式 (3.28) 中, 本文提出了一种新的方法, 可以逐个对单个障碍物进行消除变换。图 3.4 展示了本文提出的消除变换方法与传统方法的对比。 ■

针对所提出的新的消除变换, 下面对方法的正确性进行证明, 即证明从 $\mathcal{F}$ 到 $\hat{\mathcal{F}}$ 的变换 $f_{n,i}(q)$ 是微分同胚映射。为了证明这一性质, 首先给出了如下命题:

命题 1. 假设 $\mathcal{X} \subset \mathbb{R}^n$ 和 $\mathcal{Y} \subset \mathbb{R}^n$ 是 $\mathbb{R}^n$ 上的紧致且连通的拓扑流形, 且具有 $M+1$ 个不相交的边界分量。假设 $\mathcal{C} \subset \partial\mathcal{X}$ 是 $\partial\mathcal{X}$ 上的一个紧致闭集, 一个连续的映射 $f: \mathcal{X} \rightarrow \mathbb{R}^n$ 使得 $f \in C^{(k)}[\mathcal{X} - \mathcal{C}, \mathbb{R}^n]$, 对 $k > 1$, 与 $\mathcal{Y}$ 是微分同胚的, 当且仅当:

- 1) f 的雅可比矩阵在 $\mathcal{X} - \mathcal{C}$ 上是非奇异的;
- 2) f 是从 $\partial_j\mathcal{X}$ 到 $\partial_j\mathcal{Y}$ 的双射, $i = 0, 1, \dots, M$ 。

证明. 见文献 [28, 38]。

引理 3. 对于任意给定的丛林空间 $\mathcal{F}$ 和其经过消除变换 $f_{n,i}(q)$ 后的空间 $\hat{\mathcal{F}}$, 都存在一个常数 $\Xi_i > 0$, 使得当 $\xi_i > \Xi_i$ 时, $f_{n,i}(q): \mathcal{F} \rightarrow \hat{\mathcal{F}}$ 是一个微分同胚映射。

证明. 根据命题 1, 要证明 $f_{n,i}(q)$ 是微分同胚映射, 需要证明两个子条件: (i) $f_{n,i}(q)$ 的雅可比矩阵在 $\mathcal{F} - \partial\mathcal{F}$ 上是非奇异的; (2) $f_{n,i}(q)$ 在 $\mathcal{F}$ 的边界上是双射。首先, 由于已经在引理 2 中证明了 $\tilde{\rho}_{i^*}(q)$ 是光滑的, 因此 $f_{n,i}$ 的雅可比矩阵存在并且有如下形式:

$$\begin{aligned} J_{f_{n,i}} &= \sigma_i(q - p_i) \nabla v_i^\top + (v_i - 1)(q - p_i) \nabla \sigma_i^\top \\ &\quad + (1 - \sigma_i) \mathbf{I} + \sigma_i v_i \mathbf{I}. \end{aligned}$$

利于与文献[28]引理 7 和引理 8 中类似的方法, 可以证明存在一个常数 $\Xi_i > 0$, 当 $\xi_i > \Xi_i$ 满足时, $J_{f_{n,i}}$ 不仅在障碍物 $\mathcal{O}_i$ 的附近区域 $\mathcal{O}_i(\epsilon_i) = \{q \in \mathbb{R}^2 : \beta_i \leq \epsilon_i\}$, 而且在远离障碍物的区域 $\mathcal{A}(\epsilon_i) = \{q \in \mathbb{R}^2 : \beta_i \geq \epsilon_i\}$, 对任意 $\epsilon_i > 0$, 都是微分同胚的。由于详细的推导过程比较繁杂, 此处省略。读者可以参考文献[28, 38]

其次, 还需要证明 $f_{n,i}(q)$ 在边界上是双射, 即既满足单射也满足满射。首先在障碍物 $\mathcal{O}_i$ 的边界上, 有如下关系:

$$f_{n,i}(q)|_{q \in \partial\mathcal{O}_i} = \frac{\tilde{\rho}_{i^*}(q)}{\|q - p_i\|} (q - p_i) + p_i.$$

可以通过反证法来证明单射性, 即假设在障碍物的边界上存在两点 $q, q' \in \partial\mathcal{O}_i$ 使得 $f_{n,i}(q) = f_{n,i}(q')$, 则有如下关系:

$$\begin{aligned} \frac{\tilde{\rho}_{i^*}(q)}{\|q - p_i\|} (q - p_i) + p_i &= \frac{\tilde{\rho}_{i^*}(q')}{\|q' - p_i\|} (q' - p_i) + p_i \\ \Rightarrow \frac{\tilde{\rho}_{i^*}(q) \|q' - p_i\|}{\tilde{\rho}_{i^*}(q') \|q - p_i\|} (q - p_i) &= (q' - p_i), \end{aligned}$$

这说明了 $q - p_i$ 和 $q' - p_i$ 这两个向量共线。如前文所述，由于障碍物都是星形的，因此从中心出发的任意射线只能与障碍物的边界相交一次。因此， q, q' 必然是同一个点，与假设矛盾，故可得到 $f_{n,i}(q)$ 在障碍物边界上是单射的。另一方面，由于上面已经证明了只要 $\xi_i > \Xi_i$ 成立， $f_{n,i}(q)$ 的雅克比矩阵是非奇异的。因此，根据逆函数定理， $f_{n,i}(q)$ 是一个局部的同胚映射。另外，因为从紧致空间到连通空间的局部同胚是满射的，所以 $f_{n,i}(q)$ 在障碍物边界上是双射的。综上所述，存在一个常数 $\Xi_i > 0$ ，当 $\xi_i > \Xi_i$ 时， $f_{n,i}(q) : \mathcal{F} \rightarrow \hat{\mathcal{F}}$ 是一个微分同胚映射。

3.2.3 导航势函数性质分析

根据前几小节对公式 (3.9) 的描述，完整的导航势函数可以简化为如下公式：

$$\varphi_{\text{NF}}(q) = \sigma \circ \phi_{\mathcal{P}} \circ \Phi(q), \quad (3.29)$$

其中 $\Phi(q) = \Phi_{\mathcal{M} \rightarrow \mathcal{P}} \circ \Phi_{\mathcal{S} \rightarrow \mathcal{M}} \circ \Phi_{\mathcal{F} \rightarrow \mathcal{S}}(q)$ 是从丛林空间到点空间的微分同胚映射，因此 $\Phi(q)$ 满足微分同胚变换的性质。本节将讨论导航势函数 $\varphi_{\text{NF}}(q)$ 的性质，根据这些性质可以得出结论：当选择合适的仅依赖于障碍物数量的调节参数 K 时， $\varphi_{\text{NF}}(q) = \sigma \circ \phi_{\mathcal{P}} \circ \Phi(q)$ 在机器人的工作空间 $\mathcal{W}$ 中是有效的导航势函数。下面将逐步证明该性质，具体证明过程参考文献 [9, 18, 38]。首先，下面第一个引理证明了点空间中的谐势场 $\phi_{\mathcal{P}}(x)$ 在目标点 P_G 以外没有任何局部极小值。

引理 4. 除了目标点 P_G 以外，定义在点空间中的谐势场 $\phi_{\mathcal{P}}(x) : \mathcal{P} \rightarrow \mathbb{R}$ 没有任何局部最小值。

证明. 由于 $\phi_{\mathcal{P}}(x)$ 是由一系列谐函数 $\ln(\|x - q\|^2)$ 相加而成，因此 $\phi_{\mathcal{P}}(x)$ 也是谐函数。因此 $\phi_{\mathcal{P}}(x)$ 的Hessian矩阵的迹严格等于0。在任何非退化的局部最小值点， $\phi_{\mathcal{P}}(x)$ 的Hessian矩阵的两个特征值都是严格大于0的。由于海森矩阵的迹等于特征值的和，因此最小值一定只有目标点 P_G 。

假设存在一个临界点 P_c 形成了点空间中谐势场 $\phi_{\mathcal{P}}(x)$ 的退化的局部最小值。由于 $\phi_{\mathcal{P}}(x)$ 是解析的，因此它在 P_c 的局部邻域内不可能是常数。所以，在 P_c 点处， $\phi_{\mathcal{P}}(x)$ 一定在 P_c 的附近沿某个方向严格递增。这个性质要求在 P_c 附近的某个方向上， $\phi_{\mathcal{P}}(x)$ 的二阶导数在趋近 P_c 时严格为正。但是，因为 $\phi_{\mathcal{P}}(x)$ 是谐函数，因此在这些点的某些其他方向上， $\phi_{\mathcal{P}}(x)$ 具有严格的负二阶导数。由于这些点与 P_c 任意接近，因此产生矛盾，故退化的临界点只能是鞍点而不是 $\phi_{\mathcal{P}}(x)$ 的局部最小值。

引理 4 强调 $\sigma \circ \phi_{\mathcal{P}}(x)$ 只在 P_G 点有唯一的全局最小值。下面这个引理建立了一个条件，在该条件下 $\phi_{\mathcal{P}}$ 的负梯度形成的轨迹不会在点空间 $\mathcal{P}$ 中逃逸到无穷远。

引理 5. 当参数 K 满足 $K > M$ 时, 恒有 $\lim_{\|x\| \rightarrow \infty} \phi_{\mathcal{P}}(x) = +\infty$, 其中 M 为点空间 $\mathcal{P}$ 中的所有点障碍物的个数。

证明. 根据公式 (2.2), 有:

$$\phi_{\mathcal{P}}(x) = \ln(\|x - p_G\|^2) - \frac{1}{K} \sum_{i=1}^M \ln(\|x - P_i\|^2),$$

因此对于 $K > M$, 有 $\lim_{\|x\| \rightarrow \infty} \phi_{\mathcal{P}}(x) = \ln\left(\lim_{\|x\| \rightarrow \infty} \frac{\|x\|^{2K}}{\|x\|^{2M}}\right) = +\infty$. 得证。

下一个命题表明, 谐函数 $\phi_{\mathcal{P}}(x)$ 可能有退化的鞍点。然而, 可以证明出来这样的鞍点不影响导航势函数的性质, 即 $\phi_{\mathcal{P}}(x)$ 仍可以形成有效的导航势函数。

命题 2. 对于合适的调节参数 $K \geq 1$ 的选择, 都存在某些特殊的点障碍物, 使得 $\phi_{\mathcal{P}}(x)$ 在 $\mathcal{P}$ 中存在退化鞍点。

证明. 见文献 [37] 中的命题 1。

下面的命题和引理表明, 即使 $\phi_{\mathcal{P}}(x)$ 在 $\mathcal{P}$ 中存在退化的鞍点, $\phi_{\mathcal{P}}(x)$ 仍然可以形成有效的导航势函数。

命题 3. 点空间 $\mathcal{P}$ 中的谐势场 $\phi_{\mathcal{P}}(x) : \mathcal{P} \rightarrow \mathbb{R}$ 中的所有鞍点都是孤立点。

证明. 因为平滑的谐函数通常是莫尔斯函数。而莫尔斯函数只有非退化的临界点, 也就意味着它的 Hessian 矩阵具有非零的特征值。并且非退化的临界点, 尤其是非退化鞍点总是孤立点。因此, 考虑这样一个在点空间 $\mathcal{P}$ 中的退化的鞍点 P_c , 满足 $D^2\phi_{\mathcal{P}}(P_c) = [0]$ 。如果该鞍点是非孤立的, 那么通过鞍点 P_c 的函数 $\phi_{\mathcal{P}}(x)$ 的等值曲线将包含一个鞍点的连续集。由于函数 $\phi_{\mathcal{P}}$ 是解析的, 因此在其定义域内不可能是常数。故, 在 P_c 处, $\phi_{\mathcal{P}}(x)$ 的某些泰勒展开项必须非零。“永久性原理”表明, 解析函数 $\phi_{\mathcal{P}}(x)$ 必须具有通过 P_c 并且延伸至无穷的等势线, 但这是不可能的, 因为 $\lim_{\|x\| \rightarrow \infty} \phi_{\mathcal{P}}(x) = +\infty$ 。故产生矛盾。因此点空间 $\mathcal{P}$ 中的谐势场 $\phi_{\mathcal{P}}(x) : \mathcal{P} \rightarrow \mathbb{R}$ 中的鞍点一定是孤立点。

下一个引理说明了 $\phi_{\mathcal{P}}(x)$ 的鞍点在 $\mathcal{P}$ 中吸引盆的性质。

引理 6. $\phi_{\mathcal{P}}(x)$ 在点空间 $\mathcal{P}$ 中的任何一个鞍点的吸引盆都是零测集。

证明. 由于 $\phi_{\mathcal{P}}(x)$ 是定义在 $\mathcal{P}$ 上的谐函数, 因此它的 hessian 矩阵的迹, 即 $\text{tr}(D^2\phi_{\mathcal{P}})$ 在 $\mathcal{P}$ 中的所有点 $x = (x_1, x_2)$ 都是恒为 0 的。另外, 由于 Hessian 矩阵的迹等于梯度 $\nabla\phi_{\mathcal{P}}(x) = (\frac{\partial}{\partial x_1}\phi_{\mathcal{P}}, \frac{\partial}{\partial x_2}\phi_{\mathcal{P}})$ 的散度。因此梯度系统 $\frac{d}{dt}x(t) = -\nabla\phi_{\mathcal{P}}(x(t))$ 的流保持面积形式。如果一个鞍点 P_c 吸引了一个初始点的开邻域, 那么随着这些点向鞍点流动, 该邻域的面积必须收缩到零, 而这与 $\phi_{\mathcal{P}}(x)$ 的零发散性质相矛盾。故 $\phi_{\mathcal{P}}(x)$ 在点空间 $\mathcal{P}$ 中的每个鞍点的吸引盆是零测集。

根据上述定理和命题的推论，证明了在点空间 $\mathcal{P}$ 中函数 $\sigma \circ \phi_{\mathcal{P}}(x)$ 的所有与导航势函数相关的性质。最后，下面这个定理表明，在机器人的工作空间 $\mathcal{W}$ 上，完整的导航势函数 $\varphi_{\text{NF}}(q) = \sigma \circ \phi_{\mathcal{P}} \circ \Phi(q)$ 是一个有效的导航势函数，即满足定义1中的所有性质。

定理 7. 将函数 $\sigma \circ \phi_{\mathcal{P}} : \mathcal{P} \rightarrow \mathbb{R}$ 与从工作空间到点空间的导航变换 $\Phi : \mathcal{W} \rightarrow \mathcal{P}$ 组合起来，得到完整的函数 $\varphi_{\text{NF}}(q) = \sigma \circ \phi_{\mathcal{P}} \circ \Phi(q)$ 。当参数满足 $K > M$ 并且 $\mu \geq 1$ 的条件时， $\varphi_{\text{NF}}(q)$ 在机器人的工作空间 $\mathcal{W}$ 上形成了一个有效的导航势函数（如定义 1 中所述）。

证明. 首先，函数 φ_{NF} 是光滑的，因为它由光滑函数复合而成。根据定理 4， $\sigma \circ \phi_{\mathcal{P}}$ 除了 P_G 外没有任何局部最小值点。应用链式法则，可以得到 $\nabla \varphi_{\text{NF}} = [J_{\Phi}]^T \nabla(\sigma \circ \phi_{\mathcal{P}})$ ，其中 J_{Φ} 表示函数 Φ 的 2×2 雅可比矩阵。由于 Φ 是 $\mathcal{W}$ 上的微分同胚映射，在集合 $\mathcal{W}$ 中， J_{Φ} 是非奇异的。因此，在工作空间 $\mathcal{W}$ 中 φ_{NF} 的临界点和点空间 $\mathcal{P}$ 中 $\phi_{\mathcal{P}}$ 的临界点之间存在一一对应关系。

接下来考虑 φ_{NF} 在 $\mathcal{W}$ 上的临界点类型。对 $\nabla \varphi_{\text{NF}}$ 在临界点上应用链式法则，可得： $D^2 \varphi_{\text{NF}} = [J_{\Phi}]^T D^2(\sigma \circ \phi_{\mathcal{P}}) [J_{\Phi}]$ 。因此， $D^2 \varphi_{\text{NF}}$ 的特征值的符号与每个临界点处 $D^2(\sigma \circ \phi_{\mathcal{P}})$ 的特征值符号相同。由于除了 $P_G = \Phi(q_G)$ 外， $\sigma \circ \phi_{\mathcal{P}}$ 的所有临界点都是鞍点，因此 φ_{NF} 除了 q_G 外的所有临界点也都是鞍点。

接下来，验证在 q_G 处的 φ_{NF} 的唯一全局最小值是非退化的。由于 $\Phi(q)$ 保持临界点的性质不变，因此只需要证明 $\sigma \circ \phi_{\mathcal{P}}$ 在 P_G 处有唯一的非退化的最小值，其中 $P_G = \Phi(q_G)$ 。函数 $\sigma \circ \phi_{\mathcal{P}}$ 的梯度为：

$$\nabla(\sigma \circ \phi_{\mathcal{P}})(x) = \frac{\mu}{D^2} (2D \cdot (x - P_G) - \|x - P_G\|^2 \nabla D),$$

其中 $D = \|x - P_G\|^2 + \prod_{i=1}^M \|x - P_i\|^{\frac{2}{k}}$ ，因此 $\sigma \circ \phi_{\mathcal{P}}$ 在 P_G 点的Hessian矩阵为：

$$D^2 \sigma \circ \phi_{\mathcal{P}}(P_G) = \frac{2\mu}{\prod_{i=1}^M \|P_G - P_i\|^{\frac{2}{k}}} \mathbf{I}.$$

由于 P_G 是点空间 $\mathcal{P}$ 中 $\sigma \circ \phi_{\mathcal{P}}$ 函数的非退化最小值，因此 q_G 是 $\mathcal{W}$ 上 φ_{NF} 的非退化最小值。需要注意，由于所有其他临界点都是鞍点，因此 q_G 是唯一的最小值。

最后考虑 φ_{NF} 在 $\mathcal{W}$ 中的可接受性(admissibility)。如之前对 $\mathcal{W}$ 的描述， $\partial \mathcal{W}_0$ 是 $\mathcal{W}$ 的外边界， $\{\partial \mathcal{O}_1, \mathcal{O}_w, \dots, \partial \mathcal{O}_M\}$ 是 $\mathcal{W}$ 的内部障碍物边界的集合。由于 Φ 将每个 $\mathcal{O}_i$ 映射为 P_i ，对于 $i = 1, \dots, M$ 。因此，有 $\sigma \circ \phi_{\mathcal{P}} \circ \Phi(\partial \mathcal{O}_i) = \mu$ ，并且有 $\sigma \circ \phi_{\mathcal{P}}(P_i) = \mu$ ，对于外边界 $\partial \mathcal{W}_0$ ，存在关系 $\lim_{q \rightarrow \partial \mathcal{W}_0} \|\Phi(q)\| = +\infty$ 。结合引理 5，这意味着：

$$\lim_{q \rightarrow \partial \mathcal{W}_0} \sigma \circ \phi_{\mathcal{P}} \circ \Phi(q) = \mu,$$

符合导航势函数的可接受性的性质。综上所述，当参数满足 $K > M$ 并且 $\mu \geq 1$ 的条件时， $\varphi_{\text{NF}}(q)$ 在机器人的工作空间 $\mathcal{W}$ 上形成了一个有效的导航势函数。

3.3 势场的定向旋转变换

上述推导得到的势场 (3.9) 可以被用来驱动完整约束的机器人从任意出发点运动到期望的目标点 q_G ，比如机器人可以沿着势场的负梯度方向前进，相关的方法可以参考文献 [9, 10, 20, 26]。然而，这些方法无法保证机器人到达目标点的朝向是可控的，因此不适用于产生带有角度朝向的路径。为了解决已有方法的局限性，本文提出了一种对原始势场进行旋转变换的方法，可以控制机器人到达目标点的方向，并且给出了该旋转变换依旧能保持原势场的安全性和收敛性的证明。如图 3.5 所示，所提方法的核心

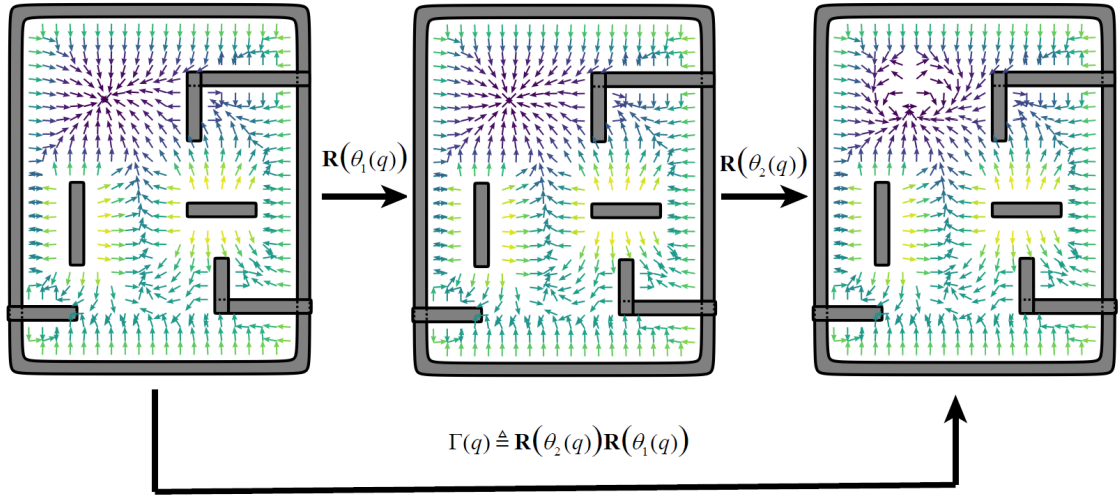


图 3.5 两步势场旋转变换。 $R(\theta_1(q))$ 为第一步旋转变换， $R(\theta_2(q))$ 为第二步旋转变换， $\Gamma(q) \triangleq R(\theta_2(q))R(\theta_1(q))$ 为整体旋转变换。

思想是对原始势场进行两步旋转变换，这样可以在目标点产生一个类似电偶极子的势场，从而使得机器人靠近达目标点附近时，可以进行方向的调整，最终以给定方向趋近于目标点。该旋转变换由以下公式给出：

$$\Gamma(q) \triangleq R(\theta_2(q)) R(\theta_1(q)), \quad (3.30)$$

其中 $R(\theta)$ 是文献 [2] 中的标准旋转矩阵，并且与角度 θ 有关系，详细表示为：

$$R(\theta) \triangleq \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix};$$

其中变量 $\theta_1(q)$ 是第一步需要转动的角度，使得在目标点附近旋转后的势场方向严格地指向目标点，即在 q 点的势场的方向都沿着 $q_G - q$ 的方向， $\theta_1(q)$ 的具体定义如下：

$$\theta_1(q) \triangleq s_d(q)\delta_\theta(q) + \mathbf{sgn}(\delta_\theta(q))[1 - s_d(q)]\delta_c, \quad (3.31)$$

其中 $\delta_\theta(q) \triangleq \theta_{q-q_G} - \theta_{\nabla\varphi_{\text{NF}}(q)}$ 是向量 $q - q_G$ 和梯度 $\nabla\varphi_{\text{NF}}(q)$ 的相对角度； $\delta_c \triangleq 2\pi$ 是一个角度常数； $s_d(q) \triangleq \exp(\tau - \frac{\tau\mu}{(\mu - \varphi_{\text{NF}}(q))^2})$ 是一个光滑的开关函数，其在目标点取值为1，在障碍物的边界取值为0，其中 $\tau \in (0, 1)$ 是一个设计参数， μ 是导航势函数 $\varphi_{\text{NF}}(q)$ 的最大值。

此外，变量 $\theta_2(q)$ 是第二步需要转动的角度，使得旋转后的势场在目标点附近类似于一个电偶极子的势场， $\theta_2(q)$ 的具体定义如下：

$$\theta_2(\check{q}) \triangleq s_d(\check{q})\delta'_\theta(\check{q}) + \text{sgn}(\delta'_\theta(\check{q}))[1 - s_d(\check{q})]\delta'_c + \delta'_c, \quad (3.32)$$

其中 $\check{q}$ 是经过坐标变换后得到的新坐标系下的坐标，新坐标系是以目标点 q_G 为坐标原点，以期望的朝向角 θ_G 为 x 轴的正方向，该坐标变换过程的详细定义如下：

$$\check{q} \triangleq \mathbf{R}^{-1}(\theta_G)(q - q_G), \quad (3.33)$$

其中 $\mathbf{R}(\theta_G)$ 是与目标角度相关的标准旋转矩阵， $\delta'_\theta(\check{q}) \triangleq \theta_{\check{q}}$ 是向量 $\check{q}$ 的角度； $\delta'_c \triangleq \pi$ 是一个角度常数；且 $s_d(\cdot)$ 的定义与公式 (3.31) 中的定义相同。上述旋转变换的概念图可见图 3.6。

将上述两步旋转变换作用到原始势场 $\varphi_{\text{NF}}(q)$ 上，可以得到一个新的势场，称为定向谐势场，该势场的定义和表述如下：

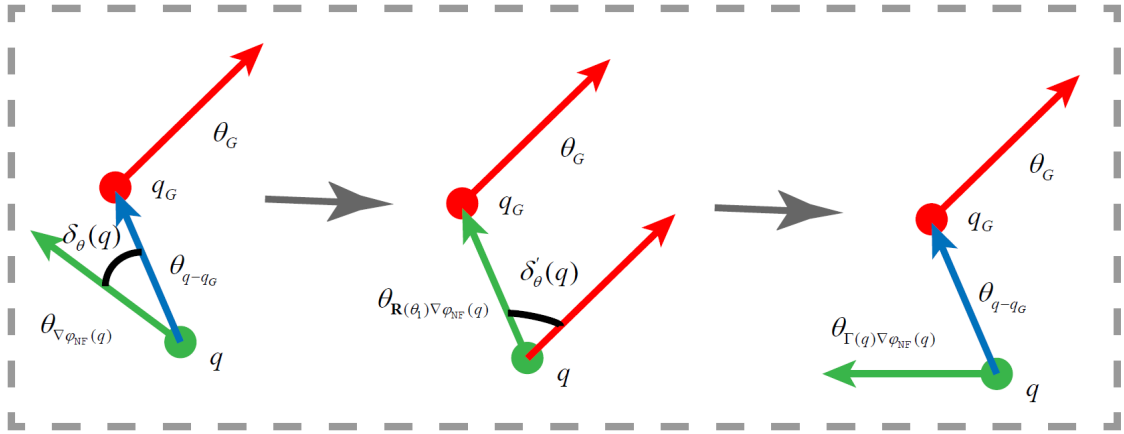


图 3.6 两步势场旋转变换概念图。

定义 15. 定向谐势场定义为对原势场进行两步旋转变换得到的势场，表示为：

$$\Upsilon(q) \triangleq -\Gamma(q)\nabla\varphi_{\text{NF}}(q), \quad (3.34)$$

其中 $\nabla\varphi_{\text{NF}}(q)$ 原始导航势函数的梯度。

给定上述的定向谐势场 $\Upsilon(q)$ ，从出发点 q_0 的积分曲线对应于如下自治偏微分系统

的轨迹 $\bar{\xi}(t)$:

$$\frac{d}{dt}\bar{\xi}(t) = \Upsilon(\bar{\xi}(t)) \in \mathbb{R}^2, \quad \bar{\xi}(t_0) = q_0 \quad (3.35)$$

对上述积分曲线的安全性和收敛性的证明总结为如下引理:

引理 8. 所有定义在 (3.34) 中的定向谐势场 $\Upsilon(q)$ 从任意出发点的积分曲线沿着目标方向 θ_G 渐进收敛到目标点 q_G , 并且不会与任何障碍物发生碰撞。

证明. 首先, 在工作空间的边界 $q \in \partial\mathcal{W}$ 上, $\varphi_{\text{NF}}(q) = \mu$ 成立。所以, 在公式 (3.31) 和公式 (3.32) 中的光滑开关函数满足 $s_d(q) = 1$, 因此可以得到 $\theta_1(q) = \theta_2(q) = \pi$ 。在这种情况下, 旋转矩阵变为单位矩阵, 即 $\mathbf{R}(\theta_1) = \mathbf{R}(\theta_2) = \mathbf{I}$, 从而 $\Gamma(q) = \mathbf{I}$ 也是一个单位矩阵, 即旋转变换后的势场是原始导航势函数的负梯度, 为 $\Upsilon(q) = -\nabla\varphi_{\text{NF}}(q)$ 。因为已经在定理 7 中证明了 $\varphi_{\text{NF}}(q)$ 是一个有效的导航势函数, 它的负梯度在障碍物的边界上永远障碍切线, 即指向远离边界的方向。所以经过旋转变换后的势场仍然满足不与障碍物碰撞的性质, 即安全性。

其次, 旋转变换 $\Gamma(q)$ 是非奇异的, 因为其中的两步变换 $\mathbf{R}(\theta_1)$ 和 $\mathbf{R}(\theta_2)$ 都是非奇异的。因此, 在旋转变换前后, 导航势函数的所有临界点都保留其性质不变, 也就是其所有临界点还是只有全局最小值 q_G 和所有的测度为零的非退化的鞍点。此外, 由于在封闭并且密集的集合中所有积分曲线都是有界的, 那么在工作空间 $\overline{\mathcal{W}}$ (上划线表示闭集) 中, 一定存在一个极限集, 使得积分曲线收敛到其中。但是如文献[39]的引理 4 中所描述, 这样的极限集并不存在, 因为不存在除 q_G 外的其他吸引子。因此, 目标点 q_G 是在 $\overline{\mathcal{W}}$ 中唯一的极限集的吸引成分, 也即 $\Upsilon(q)$ 的积分曲线收敛到目标点 q_G 。

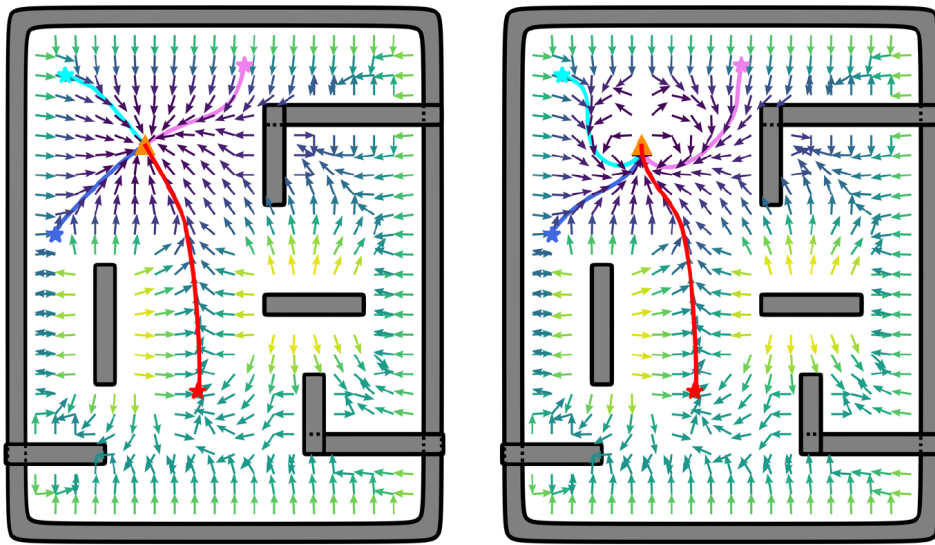


图 3.7 旋转前后的轨迹。左图为沿着旋转前势场运动的轨迹图, 到达目标点的方向无法控制; 右图为沿着旋转后的势场的轨迹图, 机器人到达目标点的方向可以控制, 这里的目标方向为 $\frac{\pi}{2}$ 。

最后, 还需要证明积分曲线是沿着目标朝向 θ_G 趋近于 q_G 的。根据旋转变换矩阵的定义, 当 q 充分靠近 q_G 的时候, 光滑开关函数满足 $s_d(q) = 1$ 。因此, $\theta_1(q)$ 和 $\theta_2(q)$ 变为 $\theta_1(q) = \delta_\theta(q)$ 和 $\theta_2(q) = \delta'_\theta(q) + \pi$ 。因此, 变换后的势场 $\Upsilon(q)$ 可以简化为如下公式:

$$\Upsilon(q) = \begin{bmatrix} \|\nabla\phi_{NF}(q)\| \cos(\delta'_\theta(q) + \theta_{q-q_G}) \\ \|\nabla\phi_{NF}(q)\| \sin(\delta'_\theta(q) + \theta_{q-q_G}) \end{bmatrix},$$

其中 $\delta'_\theta(q) \in (-\pi, \pi]$ 定义在公式 (3.32)中。然后, $\Upsilon(q)$ 和向量 $q_G - q$ 的内积可以表示为:

$$\langle \Upsilon(q), q_G - q \rangle = -\|\nabla\phi_{NF}(q)\| \|q_G - q\| \cos(\delta'_\theta(q)).$$

值得注意的是 $\Upsilon(q)$ 的积分曲线总是沿着最快的方向收敛到目标点 q_G , 也就意味着需要沿着 $\delta'_\theta(q) = \pi$ 的区域收敛。由于 $\delta'_\theta(q) = \theta_{\check{q}}$ 并且 $\check{q} = \mathbf{R}^{-1}(\theta_G)(q - q_G)$, 因此 $\delta'_\theta(q)$ 趋近于 π 意味着 $\theta_{q_G - q}$ 趋近于 θ_G 。所以, 上述积分曲线沿着所期望的朝向 θ_G 收敛到目标点。引理得证。

注 5. 很多已有的导航势函数法, 如文献 [9, 10, 20, 26, 37]中描述的方法, 能满足全局收敛性质, 但是往往不能保证沿着特定的方向趋近于目标点。而文本所提出的方法能实现这一要求。与本文类似的方法也在文献 [39]中被使用, 但是作者所提的方法只适用于目标点在原点, 并且目标朝向只能是 x 的正方向。因此, 该方法与本文所提的两步旋转变换的方法相比, 具有很大的局限性。 ■

3.4 势场跟踪控制器的设计

给定上述旋转后的势场 $\Upsilon(q)$, 本文提出了一种适用于非完整约束(Non-Holonomic)机器人的非线性控制器, 可以驱动机器人沿着势场的负梯度方向前进, 最终到达目标点。具体来说, 机器人的线速度和角速度可以通过如下公式计算:

$$v = k_v \tanh(\|q - q_G\|); \quad (3.36a)$$

$$\omega = -k_\omega (\theta - \theta_r) + v [\nabla\theta_r]^\top [J_r] \Theta, \quad (3.36b)$$

其中 q, θ 是机器人的坐标位置和方向角; $k_v, k_\omega > 0$ 是控制参数; θ_r 是向量 $\Upsilon \triangleq [\Upsilon_x, \Upsilon_y]^\top$ 的方向角; $\nabla\theta_r$ 是 Υ 对 θ_r 的梯度, 表示为 $\nabla\theta_r = [\frac{\partial\theta_r}{\partial\Upsilon_x}, \frac{\partial\theta_r}{\partial\Upsilon_y}]^\top$; J_r 是 Υ 对 q 的 2×2 雅克比矩阵, 其中的每一项为 $[J_r]_{ij} = \frac{\partial\Upsilon_i}{\partial q_j}$, $i, j \in \{1, 2\}$; 且 $\Theta \triangleq [\cos\theta, \sin\theta]^\top$ 。该控制器的核心思想是将机器人的控制系统分解为两个子系统: 通过公式 (3.36b)将机器人的朝向快速地调整为沿着 θ_r 的方向; 而公式 (3.36a)将机器人沿着给定的速度较慢地趋向目标点 q_G 。对于该控制器的收敛性证明见如下命题:

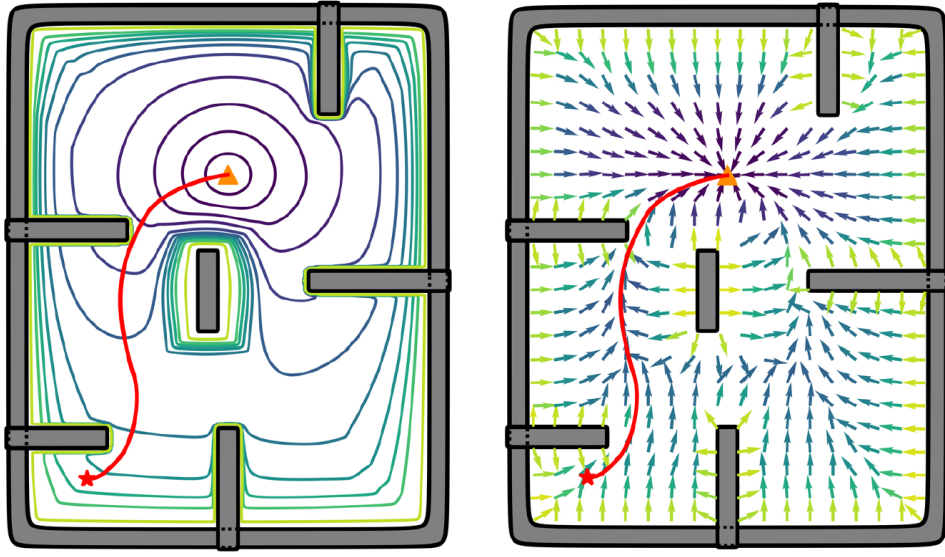


图 3.8 势场跟踪控制器的轨迹图。左图为绘制在等高线中的轨迹；右图为绘制在向量场中的轨迹。出发点为红色五角星符号，目标点为橙色三角符号。

命题 4. 在所提控制器(3.36)的驱动下，如公式(2.5)所描述的非完整约束机器人从工作空间 $\mathcal{W}$ 中的任意位姿出发，都可以渐进地沿着目标方向 θ_G 收敛到目标点 $\mathcal{W}$ 。

证明. 如前文所描述，公式(3.36)所描述的机器人模型，在控制器(3.36)下可以通过奇异扰动[40]分解为两个子系统：即关于方向角的快速子系统(3.36b)，和关于位置的较慢子系统(3.36a)。公式(3.36b)所描述的快速闭环子系统可以被化简为如下公式：

$$\begin{aligned}
 \omega &= -k_\omega(\theta - \theta_r) + v[\nabla \theta_r]^\top [J_r] \Theta \\
 &= -k_\omega(\theta - \theta_r) + \frac{\partial \theta_r}{\partial \Upsilon_x} \left(\frac{\partial \Upsilon_x}{\partial x} v \cos \theta + \frac{\partial \Upsilon_x}{\partial y} v \sin \theta \right) \\
 &\quad + \frac{\partial \theta_r}{\partial \Upsilon_y} \left(\frac{\partial \Upsilon_y}{\partial x} v \cos \theta + \frac{\partial \Upsilon_y}{\partial y} v \sin \theta \right) \\
 &= -k_\omega(\theta - \theta_r) + \frac{\partial \theta_r}{\partial \Upsilon_x} \left(\frac{\partial \Upsilon_x}{\partial x} \dot{x} + \frac{\partial \Upsilon_x}{\partial y} \dot{y} \right) + \frac{\partial \theta_r}{\partial \Upsilon_y} \left(\frac{\partial \Upsilon_y}{\partial x} \dot{x} + \frac{\partial \Upsilon_y}{\partial y} \dot{y} \right) \\
 &= -k_\omega(\theta - \theta_r) + \frac{\partial \theta_r}{\partial \Upsilon_x} \frac{\partial \Upsilon_x}{\partial t} + \frac{\partial \theta_r}{\partial \Upsilon_y} \frac{\partial \Upsilon_y}{\partial t} \\
 &= -k_\omega(\theta - \theta_r) + \dot{\theta}_r,
 \end{aligned}$$

该结果保证机器人的朝向角 θ 可以以指数形式快速收敛到该点势场的方向 θ_r 。因此，在这种情况下，慢速子系统可以重新表述为： $\dot{q} = v \frac{\Upsilon}{\|\Upsilon\|}$ 。该系统的轨迹对应于势场 $\Upsilon(q)$ 的积分曲线。由于已经在引理3中证明了 $\Upsilon(q)$ 的积分曲线沿着方向角 θ_G 收敛到目标点 q_G ，那么该慢速子系统也收敛到目标点。因此，在控制器(3.36)的驱动下，机器人能从工作空间 $\mathcal{W}$ 中任意的起点全局地并且渐进的沿着目标方向角 θ_G 收敛到目标

点 q_G

最后，在所提出的控制算法 (3.36) 的驱动下，机器人从 (q_0, θ_0) 到 (q_G, θ_G) 的控制代价可以用以下代价函数进行估计：

$$\gamma((q_0, \theta_0), (q_G, \theta_G)) = r(q_0, q_G) + \mathbf{w}, \mathbf{Q}^\top(\theta_0, \theta_G), \quad (3.37)$$

其中，第一部分 $r(q_0, q_G) = |q_0 - q_G|$ 表示直线距离代价；第二部分用于计算角度转向的代价：(i) $\mathbf{w} = [w_1, w_2]^\top$ 是权重，其中 $w_1, w_2 > 0$ ；(ii) $\mathbf{Q}(\theta_0, \theta_G) = [Q_1, Q_2]$ ，其中 $Q_1 = \|\theta_r - \theta_0\|$ 表示机器人当前方向和参考向量方向之间的转向代价； $Q_2 = \|\theta_{0G} - \theta_G\|$ 表示向量 $q_G - q_0$ 和目标方向之间的转向代价。在后续在线迭代部分，将根据机器人实际的运动代价对公式 (3.37) 中的权重 $\mathbf{w}$ 进行更新。

3.5 双层导航树的构建

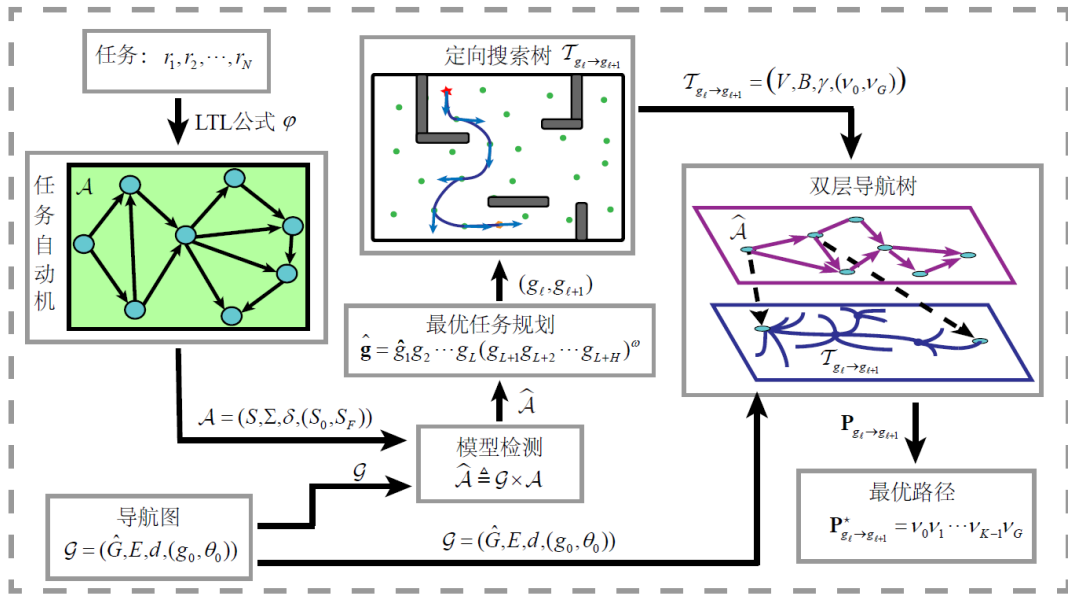


图 3.9 双层导航树概念图。

如图 3.9 所示，本文所提出的双层导航树的概念图主要包括这几个模块：基于线性时序逻辑的任务自动机、基于环境的导航图和基于任务序列的定向搜索树。双层导航树主要工作流程为：首先使用线性时序逻辑描述任务点 $\{r_1, r_2, \dots, r_N\}$ 之间的时序关系，然后将其转化为任务自动机 $\mathcal{A}$ 。接着将其与基于环境的导航图 $\mathcal{G}$ 做乘积，得到乘积自动机 $\hat{\mathcal{A}}$ ，并通过对其求解得到最优的任务序列，然后在任务点之间依次构建定向搜索树，从而得到最优的路径规划结果。下面将详细介绍双层导航树的构建过程。

首先根据第 2.1.3 节的描述，与线性时序逻辑公式 φ 相关的 Büchi 自动机 NBA 由公式

$$\mathcal{A} = (\mathcal{S}, \Sigma, \delta, (\mathcal{S}_0, \mathcal{S}_F))$$

给出，其中 $\mathcal{A}$ 描述了满足任务约束的所有可能的轨迹。然后，初始化的“导航图”被构建为一个加权且完全连通的图：

$$\mathcal{G} \triangleq (\hat{G}, E, d, (g_0, \theta_0)), \quad (3.38)$$

其中：(i) $\hat{G} = G \times \Theta$ 是感兴趣区域的集合以及一个选定整数 K 后对应的方向角集合 $\Theta = 2k\pi/K, k = 0, \dots, K-1$ ；(ii) $E = \hat{G} \times \hat{G}$ 是完全连通图的转移边的集合；(iii) $d : E \rightarrow \mathbb{R}^+$ 是节点之间的代价函数，其初始化为 $d((g, \theta), (g', \theta')) \triangleq |g - g'|_2 + w|\theta - \theta'|$ ，其中 $\forall (g, \theta), (g', \theta') \in \hat{G}$ ，参数满足 $w > 0$ ；并且 (g_0, θ_0) 是初始姿态。需要注意的是，初始时刻， E 是完全连通的，因为从一个区域到另一个区域的导航可行性只能在相关控制器构建之后确定。由于同样的原因，初始的导航代价通过欧几里得距离进行估计，后续将进行在线更新。

给定导航图 $\mathcal{G}$ 和任务自动机 $\mathcal{A}$ ，可以按照标准的模型检测流程来找到满足给定任务 φ 约束的规划结果。具体而言，可以构建 $\mathcal{G}$ 和 $\mathcal{A}$ 的同步乘积，即

$$\hat{\mathcal{A}} \triangleq \mathcal{G} \times \mathcal{A} = (\hat{S}, \hat{\delta}, \hat{d}, (\hat{S}_0, \hat{S}_F)),$$

其中 $\hat{S} = \hat{G} \times \mathcal{S}$ ， $\hat{S}_0, \hat{S}_F \subset \hat{S}$ 是初始状态和接受状态的集合， $\hat{\delta} \subset \hat{S} \times \hat{S}$ 满足 $(\langle g, s \rangle, \langle g', s' \rangle) \in \hat{\delta}$ 当且仅当 $(g, g') \in E$ 且 $s' \in \delta(s, g)$ ； $\hat{d}(\langle g, s \rangle, \langle g', s' \rangle) = d(g, g')$ ，对于所有 $(\langle g, s \rangle, \langle g', s' \rangle) \in \hat{\delta}$ 成立。值得注意的是，乘积 $\hat{\mathcal{A}}$ 仍然是一个 Büchi 自动机，其接受的运行满足第 2.1.3 节中描述的前缀-后缀结构。更具体地，考虑以下 $\hat{\mathcal{A}}$ 的一次运行：

$$\hat{S} = \hat{s}_1 \hat{s}_2 \cdots \hat{s}_L (\hat{s}_{L+1} \hat{s}_{L+2} \cdots \hat{s}_{L+H})^\omega,$$

其中 $\hat{s}_1 \hat{s}_2 \cdots \hat{s}_L$ 是执行一次的前缀， $\hat{s}_{L+1} \hat{s}_{L+2} \cdots \hat{s}_{L+H}$ 是无限重复的后缀， $\hat{s}_1 \in \hat{S}_0$ ， $\hat{s}_{L+1} \in \hat{S}_F$ ， $L, H \geq 1$ 。采用嵌套的 Dijkstra 算法可以求解出最佳的初始和接受状态对 $(\hat{s}_0^*, \hat{s}_F^*)$ 。因此，给定初始环境，通过将 $\hat{S}$ 投影到 $\hat{G}$ 上，可以获得最优的任务规划结果，即

$$\hat{g} = \hat{g}_1 \hat{g}_2 \cdots \hat{g}_L (\hat{g}_{L+1} \hat{g}_{L+2} \cdots \hat{g}_{L+H})^\omega, \quad (3.39)$$

其中 $\hat{g}_\ell = \hat{s}_\ell|_{\hat{G}}$ 表示感兴趣区域。关于求解算法更详细的描述可以参考文献 [33]。

给定上述最优的任务序列，可以通过设计适当的势场，按顺序遍历任务规划结果 $\hat{g}$ 中的每条转移边 $(\hat{g}_\ell, \hat{g}_{\ell+1})$ 。这一方法在文献 [9, 20] 中有所提及。然而，正如在 1.2 节中提到的，当 $(\hat{g}_\ell, \hat{g}_{\ell+1})$ 在部分已知的工作空间中相距较远时，得到的轨迹可能会产生局部震荡或抖动（尤其是靠近工作空间的边界时），并且会导致较大的控制代价。因

此, 本文提出了一种结构化的定向搜索树来解决这些问题。具体而言, 与 $(\hat{g}_\ell, \hat{g}_{\ell+1})$ 相关联的定向搜索树是由四元组定义的树状结构:

$$\mathcal{T}_{\hat{g}_\ell \rightarrow \hat{g}_{\ell+1}} \triangleq (V, B, \gamma, (v_0, v_G)), \quad (3.40)$$

其中 $V \subset \mathcal{W} \times (-\pi, \pi]$ 是顶点的集合; $B \subset V \times V$ 是边的集合; $\gamma: B \rightarrow \mathbb{R}_{\geq 0}$ 是需要估计的边的代价; $v_0 = \hat{g}_\ell$ 和 $v_G = \hat{g}_{\ell+1}$ 是初始和目标位姿。因此接下来的目标就是在 $\mathcal{T}$ 中找到一系列的顶点, 作为从 v_0 到 v_G 的路径点。初始时, $V = \{v_0, v_G\}$, $B = \emptyset$ 。然后, 顶点集合可以通过各种常用的方法生成, 例如直接离散化, 包括均匀分割、三角剖分和文献 [2]中的Voronoi分割; 或者采用基于采样的方法, 如 PRM [21]和RRT [1]。因此, 从这点可以看出, 定向搜索树并不局限于某种特定的方法。值得一提的是, 在工作空间 $\mathcal{W}$ 中的这些顶点都应该带有角度倾向, 以使得其更适用于非完整约束机器人。得到上述顶点集合之后, 任何一个顶点都与其自由邻域内的所有顶点相连, 即 $(v, v') \in B$, 只要对于某一给定半径 $\varepsilon > 0$, 满足 $|v - v'| < \varepsilon$ 。对于每条可行边 $(v, v') \in E$, 从 v 到 v' 的路径可以通过构建以 v' 为目标点的谐波势场得到, 从而驱动公式(2.5)中描述的机器人的控制代价 $\gamma(v, v')$ 可以使用公式 3.37进行估计。因此, 给定上述的任何加权有向连接树 $\mathcal{T}_{\hat{g}_\ell \rightarrow \hat{g}_{\ell+1}}$, 可以使用文献中 [2]的任何搜索算法进行求解, 例如Dijkstra算法和A*算法(将 $|v - v_G|$ 作为启发式函数), 从而找到从 v_0 到 v_G 的最短路径。将路径结果记为:

$$\mathbf{P}_{\hat{g}_\ell \rightarrow \hat{g}_{\ell+1}} \triangleq v_0 v_1 \cdots v_{K-1} v_G, \quad (3.41)$$

其中 $v_k \in V$, $(v_k, v_{k+1}) \in B$, $\forall k \in [0, K-1]$ 。换句话说, 路径上的每个顶点都可以作为从初始位姿到目标位姿导航的中间路径点。在上述过程中, 使用导航代价公式(3.37)可以计算搜索树 $\mathcal{T}_{\hat{g}_\ell \rightarrow \hat{g}_{\ell+1}}$ 中的每条边的控制代价, 从而更准确地估计从任务区域 $\hat{g}_\ell$ 到区域 $\hat{g}_{\ell+1}$ 的导航代价。因此, 导航图 $\mathcal{G}$ 中的实际代价函数由下式给出:

$$d(\hat{g}, \hat{g}') = \sum_{k=0}^{K-1} \gamma(v_k, v_{k+1}), \quad (3.42)$$

其中 $\mathbf{P}_{\hat{g} \rightarrow \hat{g}'} = v_0 v_1 \cdots v_K$ 是搜索树 $\mathcal{T}_{\hat{g} \rightarrow \hat{g}'}$ 中从 $\hat{g}$ 到 $\hat{g}'$ 的最短路径。

第四章 在线迭代更新

根据上述初始规划和执行策略，机器人可以在工作空间内自主导航并探索其周围环境。随着机器人检测到更多的障碍物，初始规划的双层导航树需要被更新，其中既包括上层树结构的更新又包括底层导航势函数的更新。在对这两部分进行更新时，如果完全重新规划导航树并且重新计算导航势函数将会导致较大的计算量，这使得算法不适用于在线的情形。针对该问题，本文提出了使用递归算法来逐步更新导航势函数和导航树的结构，从而能提高在线执行的效率。

4.1 导航势函数的在线迭代计算

传统计算导航势函数的方法，如文献[9, 28, 38]中的方法，在构建势场的时候，必须一次性将已知工作区域中的所有障碍物全部考虑进去。这意味着每次有新的障碍物添加到工作空间中时，整体的计算过程都必须从头开始，这会带来很多不必要的重复计算。一种更直观的方法是通过存储和重用一些中间变量，增量式地更新和计算导航势函数的不同部分。具体而言，考虑以下两种情形：(i) 新添加的障碍物与已有的障碍物相互独立；(ii) 新添加的障碍物与已有的障碍物相互重叠。

4.1.1 独立障碍物的迭代计算

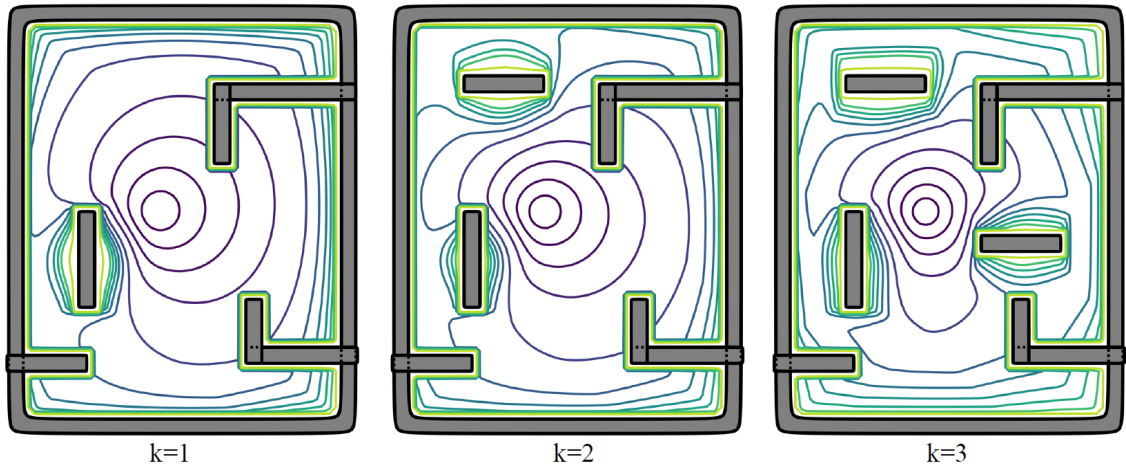


图 4.1 独立障碍物的迭代计算。

首先考虑简单的情形，如图4.1所示，即将一组不重叠的星形障碍物 $\mathcal{O}_k, k = 1, \dots, M$ ，按顺序依次添加到工作空间 $\mathcal{W} = \mathcal{O}_0$ 中。如第3.2节所讨论的那样，对于每个障碍物，

可以通过公式(7)中的径向放缩变换构建从星形到球形的变换。具体而言,在添加完第 k 个障碍物后,从星形空间到圆形世界的微分同胚映射可以重新表示为:

$$\Phi_{S \rightarrow \mathcal{M}}^k(q) \triangleq \text{id}(q) + \sum_{i=0}^k \sigma_i^k(q) [v_i(q) - 1] (q - q_i), \quad (4.1)$$

其中 $\sigma_i^k(q)$ 是与 i 和 k 相关的解析切换函数; $v_i(q)$ 是定义在公式(3.11)中的变形因子; q_i 是障碍物 $\mathcal{O}_i$ 的几何中心,其中 $i = 0, 1, \dots, k$ 。接下来,定义如下中间变量:

$$F_i^k(q) \triangleq \sigma_i^k(q) r_i(q), \quad (4.2)$$

其中 r_i 定义为 $r_i \triangleq [v_i(q) - 1](q - q_i)$ 。因此,公式(4.1)中表示的变换可以调整为如下公式:

$$\Phi_{S \rightarrow \mathcal{M}}^k(q) = \text{id}(q) + \sum_{i=0}^k F_i^k(q). \quad (4.3)$$

值得注意的是当 $k = 0$ 时,对于工作空间外边界的变换函数 $F_0^0(q)$ 可以使用公式(3.17)中描述的初始变换进行计算。之后,每当新的障碍物 $\mathcal{O}_{k+1}$ 添加到工作空间中时,对于已有障碍物的变换函数 $F_i^k(q)$ 需要更新到 $F_i^{k+1}(q)$, $\forall i = 0, 1, \dots, k$,并且对于新添加的障碍物的变换函数 $F_{k+1}^{k+1}(q)$ 也需要计算出来。为了方便表示,定义以下用于递归计算的多元辅助函数:

$$\Psi(F, r, \tilde{r}, \alpha) \triangleq \frac{\|F\| \tilde{r}}{\alpha(\|r\| - \|F\|) + \|F\|}, \quad (4.4)$$

其中 $F, r, \tilde{r}$ 表示向量,而 α 是一个标量,并且满足 $\alpha(\|r\| - \|F\|) + \|F\| \neq 0$ 。借助该辅助函数,与工作空间外边界相关的变换函数 $F_0^{k+1}(q)$ 可以从函数 $F_0^k(q)$ 迭代计算得到,即有如下公式:

$$F_0^{k+1}(q) = \Psi(F_0^k(q), r_0(q), r_0(q), \alpha^k(q)), \quad (4.5)$$

其中 $r_0(q)$ 是定义在公式(4.2)中的与工作空间外边界相关的变量,变量 $\alpha^k(q)$ 定义为 $\alpha^k(q) = \frac{\lambda_{k+1}}{\lambda_k \beta_{k+1}(q)}$, λ_k 和 λ_{k+1} 是定义在解析切换公式(3.17)中的正实数。关于迭代计算公式(4.5)合理性的证明见如下引理。

引理 9. 在公式(4.5)中,关于工作空间的外部边界的变换函数 $F_0^{k+1}(q)$ 的迭代计算是有效的,其中 $k = 0, 1, \dots, M-1$ 。

证明. 可以通过数学归纳法进行证明。首先, $F_0^k(q)$ 的初始值由初始公式(3.17)计算得到,即 $F_0^0(q) = \sigma_0^0(q) r_0(q)$,该结果满足变换函数(4.2)。现假设存在某一个特定的正整数 $m \geq 0$,使得

$$F_0^m(q) = \sigma_0^m(q) r_0(q)$$

成立，其中

$$\sigma_0^m(q) = \frac{\gamma_G(q)\bar{\beta}_0^m(q)}{\gamma_G(q)\bar{\beta}_0^m(q) + \lambda_m\beta_0(q)},$$

那么，通过函数(4.5)推导出的 $F_0^{m+1}(q)$ 结果为：

$$F_0^{m+1}(q) = \frac{\|r_0(q)\| r_0(q)}{\alpha^m(q)(\|r_0(q)\| - \|\sigma_0^m(q)r_0(q)\|) + \|\sigma_0^m(q)r_0(q)\|}。$$

由于

$$\|r_0(q)\| \geq 0, \quad \sigma_0^m(q) \geq 0, \quad \text{并且} \quad \alpha^k(q) = \frac{\lambda_{k+1}}{\lambda_k\beta_{k+1}(q)},$$

那么 $F_0^{m+1}(q)$ 可以化简为：

$$\begin{aligned} F_0^{m+1}(q) &= \frac{\lambda_m\sigma_0^m(q)\beta_m(q)}{\sigma_0^m(q) + \lambda_{m+1}(1 - \sigma_0^m(q))} r_0(q) \\ &= \frac{\lambda_m\gamma_G(q)\bar{\beta}_0^m(q)\beta_m(q)}{\gamma_G(q)\bar{\beta}_0^m(q) + \lambda_{m+1}(\gamma_G(q)\bar{\beta}_0^m(q) + \lambda_m\beta_0(q) - \gamma_G(q)\bar{\beta}_0^m(q))} r_0(q) \\ &= \frac{\lambda_m\gamma_G(q)\bar{\beta}_0^m(q)\beta_m(q)}{\gamma_G(q)\bar{\beta}_0^m(q) + \lambda_{m+1}\lambda_m\beta_0(q)} r_0(q)。 \end{aligned}$$

此外，因为

$$\sigma_0^{m+1}(q) = \frac{\gamma_G(q)\bar{\beta}_0^{m+1}(q)}{\gamma_G(q)\bar{\beta}_0^{m+1}(q) + \lambda_{m+1}\beta_0(q)}。$$

并且

$$\bar{\beta}_0^{m+1}(q) = \bar{\beta}_0^m(q)\beta_{m+1}(q)。$$

所以 $F_0^{m+1}(q)$ 可以进一步化简为：

$$F_0^{m+1}(q) = \sigma_0^{m+1}(q)r_0(q)。$$

该结果也满足变换函数(4.2)。综上所述， $F_0^{k+1}(q) = \Psi(F_0^k(q), r_0(q), r_0(q), \alpha^k(q))$ 对 $\forall k = 0, 1, \dots, M-1$ 都成立。

此外，对每个内部障碍物 $\mathcal{O}_i$ ， $i = 0, 1, \dots, k$ ，相关的变换函数 $F_{i+1}^{k+1}(q)$ 也可以通过公式 $F_0^{k+1}(q)$ 递推计算得到，即：

$$F_{i+1}^{k+1}(q) = \Psi(F_i^{k+1}(q), r_i(q), r_{i+1}(q), \alpha_i(q))。 \quad (4.6)$$

其中 $r_i(q)$ 和 $r_{i+1}(q)$ 是定义在公式(4.2)中的与第 i 个和第 $(i+1)$ 个障碍物相关的变量，且变量 $\alpha_i(q)$ 定义为 $\alpha_i(q) = \frac{\beta_{i+1}^2(q)}{\beta_i^2(q)}$ 。关于迭代计算公式(4.5)有效性的证明见如下引理。

引理 10. 在公式(11)中，关于每个内部障碍物的变换函数 $F_0^{k+1}(q)$ 的迭代计算是有效的，

其中 $i = 0, 1, \dots, k$ 。

证明. 同样可以通过数学归纳法证明。首先, $F_{i+1}^{k+1}(q)$ 的初始值由初始公式(3.17)计算得到, 即 $F_0^{k+1}(q) = \sigma_0^{k+1}(q) r_0(q)$, 该结果满足变换函数(4.2)。现假设存在某一个特定的正整数 $m \geq 0$, 使得

$$F_m^{k+1}(q) = \sigma_m^{k+1}(q) r_m(q)$$

成立, 其中

$$\sigma_m^{k+1}(q) = \frac{\gamma_G(q) \bar{\beta}_m^{k+1}(q)}{\gamma_G(q) \bar{\beta}_m^{k+1}(q) + \lambda_{k+1} \beta_m(q)}。$$

那么, 通过函数(11)推导出的 $F_{m+1}^{k+1}(q)$ 结果为:

$$F_{m+1}^{k+1}(q) = \frac{\|\sigma_m^{k+1}(q) r_m(q)\| r_{m+1}(q)}{\alpha_m(q) (\|r_m(q)\| - \|\sigma_m^{k+1}(q) r_m(q)\|) + \|\sigma_m^{k+1}(q) r_m(q)\|}。$$

由于

$$\|r_m(q)\| \geq 0, \quad \sigma_m^{k+1}(q) \geq 0, \quad \text{并且} \quad \alpha_i(q) = \beta_{i+1}^2(q) / \beta_i^2(q)。$$

那么 $F_{m+1}^{k+1}(q)$ 可以化简为:

$$\begin{aligned} F_{m+1}^{k+1}(q) &= \frac{\sigma_m^{k+1}(q) \beta_m^2(q)}{\sigma_m^{k+1}(q) \beta_m^2(q) + (1 - \sigma_m^{k+1}(q)) \beta_{m+1}^2(q)} r_{m+1}(q) \\ &= \frac{\gamma_G(q) \bar{\beta}_m^{k+1}(q) \beta_m^2(q)}{\gamma_G(q) \bar{\beta}_m^{k+1}(q) \beta_m^2(q) + (\gamma_G(q) \bar{\beta}_m^{k+1}(q) + \lambda_{k+1} \beta_m(q) - \gamma_G(q) \bar{\beta}_m^{k+1}(q)) \beta_{m+1}^2(q)} r_{m+1}(q) \\ &= \frac{\gamma_G(q) \bar{\beta}_m^{k+1}(q) \beta_m(q)}{\gamma_G(q) \bar{\beta}_m^{k+1}(q) \beta_m(q) + \lambda_{k+1}} r_{m+1}(q)。 \end{aligned}$$

此外, 因为

$$\sigma_{m+1}^{k+1}(q) = \frac{\gamma_G(q) \bar{\beta}_{m+1}^{k+1}(q)}{\gamma_G(q) \bar{\beta}_{m+1}^{k+1}(q) + \lambda_{k+1} \beta_{m+1}(q)},$$

并且

$$\bar{\beta}_{m+1}^{k+1}(q) = \frac{\bar{\beta}_m^{k+1}(q) \beta_m(q)}{\beta_{m+1}(q)},$$

所以 $F_{m+1}^{k+1}(q)$ 可以进一步化简为:

$$F_{m+1}^{k+1}(q) = \sigma_{m+1}^{k+1}(q) r_{m+1}(q)。$$

该结果也满足变换函数(4.2)。综上所述, $F_{i+1}^{k+1}(q) = \Psi(F_i^{k+1}(q), r_i(q), r_{i+1}(q), \alpha_i(q))$ 对 $\forall i = 0, 1, \dots, k$ 都成立。

注 6. 公式(4.5)是对函数 $F_0^{k+1}(q)$ 的上标 k 进行迭代计算, 并且其初值为 $F_0^0(q)$, 而公

式(11)是对函数 $F_{i+1}^{k+1}(q)$ 的下标 i 进行迭代计算，并且其初值为 $F_0^{k+1}(q)$ 。需要进行两次迭代的原因是，当第 $(k+1)$ 个障碍物添加到已有的工作空间中，要先将与工作空间的外边界有关的变换函数 F_0^k 更新到 F_0^{k+1} ，这是因为要考虑新添加的障碍物的信息，其次，以函数 F_0^{k+1} 作为初始值，通过迭代计算公式(11)增量式地计算对每个内部障碍物的变换函数 F_{i+1}^{k+1} 。

4.1.2 重叠障碍物的迭代计算

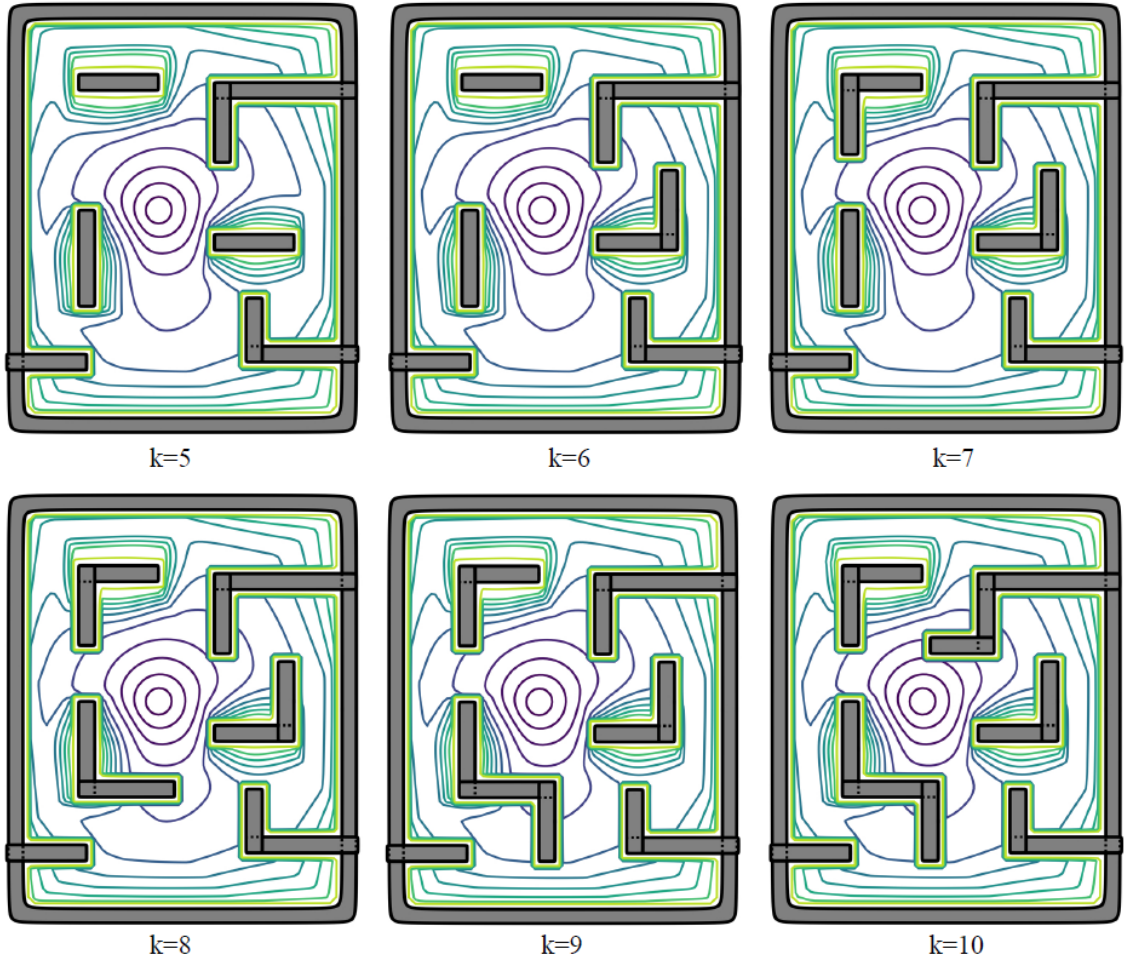


图 4.2 重叠障碍物的迭代计算。

其次，需要考虑新添加的障碍物可能与已有的障碍物相互重叠。更具体地，假设一系列障碍物 $\mathcal{O}_k, k = 1, \dots, N$ ，添加到一个星形空间中，并与一个现有障碍物相互重叠，形成一个丛林空间，如图4.2所示。如第3.2节所讨论的静态场景中所述，从丛林空间到星形空间的消除变换(3.21)可以通过径向放缩变换(3.27)进行构建。详细来讲，对上述丛林空间中某个星形树的第 k 个障碍物的消除变换可以重新表示为：

$$f_k(q) \triangleq \text{id}(q) + \sigma_k(q, \xi_k) [(v_k(q) - 1)(q - p_k)], \quad (4.7)$$

其中 $\sigma_k(q, \xi_k)$ 是与 k 相关的解析切换函数, $v_k(q)$ 是定义在公式(3.25)中的变形因子, 并且 p_k 是定义在公式(3.19)中的障碍物 $\mathcal{O}_k$ 和其父障碍物 $\mathcal{O}_{k^*}$ 的共同中心。与公式(4.2)类似, 定义如下中间变量:

$$H_k(q) \triangleq \sigma_k(q, \xi_k) r_k(q), \quad (4.8)$$

其中 $r_k(q)$ 定义为 $r_k(q) \triangleq [v_k(q) - 1](q - p_k)$ 。因此, 在添加完第 k 个障碍物后, 从丛林空间到星形空间的变换可以调整为如下公式:

$$\Phi_{\mathcal{F} \rightarrow \mathcal{S}}^k(q) \triangleq H_1 \circ H_2 \circ \cdots \circ H_k(q). \quad (4.9)$$

需要注意, 当 $k = 1$ 时, 新添加的障碍物构成了一棵新的星形树, 并且对于障碍物 $\mathcal{O}_1$ 的消除变换可以通过公式(3.28)进行计算。然后, 每当第 $(k + 1)$ 个障碍物添加到工作空间中, 公式(4.9)可以按照如下方式进行更新:

$$\Phi_{\mathcal{F} \rightarrow \mathcal{S}}^{k+1}(q) = \Phi_{\mathcal{F} \rightarrow \mathcal{S}}^k \circ H_{k+1}(q), \quad (4.10)$$

其中函数 $H_{k+1}(q)$ 可以通过公式(4.4)从函数 $H_k(q)$ 迭代得到, 即:

$$H_{k+1}(q) = \Psi(H_k(q), r_k(q), r_{k+1}(q), \alpha_k(q)), \quad (4.11)$$

其中 $\Psi(\cdot)$ 是定义在公式(4.4)中的辅助函数, $r_{k+1}(q)$ 和 $r_k(q)$ 是定义在公式(4.8)中分别与第 $(k + 1)$ 和第 k 个障碍物相关的变量, 且 $\alpha_k(q)$ 是额外的变量, 具体形式如下:

$$\alpha_k(q) = \zeta_k \frac{\beta_{k+1}(q) \tilde{\beta}_k(q) \beta_{(k+1)^*}(q)}{\beta_k^3(q) \tilde{\beta}_{k+1}(q) \beta_{k^*}(q)},$$

其中 $\zeta_k = \xi_{k+1}/\xi_k$, 且 ξ_k 和 ξ_{k+1} 是定义在公式(3.21)中的设计参数, $\tilde{\beta}_{k+1}(q)$ 和 $\tilde{\beta}_k(q)$ 的定义与在公式(3.21)中的定义相同。关于迭代计算公式(4.11)有效性的证明见如下引理。

引理 11. 在公式(4.11)中, 关于每个新添加的叶障碍物的变换函数 $H_{k+1}(q)$ 的迭代计算是有效的, 其中 $k = 0, 1, \dots, N - 1$ 。

证明. 该引理同样可以通过数学归纳法证明。首先, $H_{k+1}(q)$ 的初始值由初始公式(3.28)计算得到, 即 $H_1(q) = \sigma_1(q, \xi_1) r_1(q)$, 该结果满足变换函数(4.8)。现假设存在某一个特定的正整数 $l \geq 0$, 使得

$$H_l(q) = \sigma_l(q, \xi_l) r_l(q)$$

成立, 其中

$$\sigma_l(q, \xi_l) = \frac{\gamma_G(q) \bar{\beta}_l(q)}{\gamma_G(q) \bar{\beta}_l(q) + \xi_l \beta_l(q)}.$$

那么, 通过函数(4.11)推导出的 $H_{m+1}(q)$ 结果为:

$$H_{l+1}(q) = \frac{\|\sigma_l(q, \xi_l) r_l(q)\| r_{l+1}(q)}{\alpha_l(q) (\|r_l(q)\| - \|\sigma_l(q, \xi_l) r_l(q)\|) + \|\sigma_l(q, \xi_l) r_l(q)\|}.$$

由于

$$\|r_{l+1}(q)\| \geq 0, \quad \sigma_l(q, \xi_l) \geq 0, \quad \text{并且} \quad \alpha_l(q) = \frac{\xi_{l+1}}{\xi_l} \frac{\beta_{l+1}(q) \tilde{\beta}_l(q) \beta_{(l+1)^*}(q)}{\beta_l^3(q) \tilde{\beta}_{l+1}(q) \beta_{l^*}(q)},$$

那么 $H_{l+1}(q)$ 可以化简为:

$$\begin{aligned} H_{l+1}(q) &= \frac{\sigma_l(q, \xi_l) \xi_l \beta_l^3(q) \tilde{\beta}_{l+1}(q) \beta_{l^*}(q)}{\gamma_G(q) \bar{\beta}_l(q) \xi_l \beta_l^3(q) \tilde{\beta}_{l+1}(q) \beta_{l^*}(q) + (1 - \sigma_l(q, \xi_l)) \xi_{l+1} \beta_{l+1}(q) \tilde{\beta}_l(q) \beta_{(l+1)^*}(q)} r_{l+1}(q) \\ &= \frac{\gamma_G(q) \bar{\beta}_l(q) \xi_l \beta_l^3(q) \tilde{\beta}_{l+1}(q) \beta_{l^*}(q)}{\gamma_G(q) \bar{\beta}_l(q) \xi_l \beta_l^3(q) \tilde{\beta}_{l+1}(q) \beta_{l^*}(q) + \xi_l \beta_l(q) \xi_{l+1} \beta_{l+1}(q) \tilde{\beta}_l(q) \beta_{(l+1)^*}(q)} r_{l+1}(q) \\ &= \frac{\gamma_G(q) \bar{\beta}_l(q) \beta_l^2(q) \tilde{\beta}_{l+1}(q) \beta_{l^*}(q)}{\gamma_G(q) \bar{\beta}_l(q) \beta_l^2(q) \tilde{\beta}_{l+1}(q) \beta_{l^*}(q) + \xi_{l+1} \beta_{l+1}(q) \tilde{\beta}_l(q) \beta_{(l+1)^*}(q)} r_{l+1}(q). \end{aligned}$$

此外, 因为

$$\sigma_{l+1}(q, \xi_{l+1}) = \frac{\gamma_G(q) \bar{\beta}_{l+1}(q)}{\gamma_G(q) \bar{\beta}_{l+1}(q) + \xi_{l+1} \beta_{l+1}(q)},$$

并且

$$\bar{\beta}_{l+1}(q) = \frac{\bar{\beta}_l(q) \beta_l^2(q) \beta_{l^*}(q) \tilde{\beta}_{l+1}(q)}{\beta_{l+1}^*(q) \tilde{\beta}_l(q)},$$

所以 $H_{l+1}(q)$ 可以进一步化简为:

$$H_{l+1}(q) = \sigma_{l+1}(q, \xi_{l+1}) r_{l+1}(q),$$

该结果也满足变换函数(4.8)。综上所述, $H_{k+1}(q) = \Psi(H_k(q), r_k(q), r_{k+1}(q), \alpha_k(q))$ 对 $\forall k = 0, 1, \dots, N-1$ 都成立。

注 7. 注意到, 对独立障碍物的迭代计算和对重叠障碍物的迭代计算这两种情况使用了相同的辅助函数 $\Psi(\cdot)$ 进行递归计算, 这是因为该辅助函数是从公式(3.17)和公式(3.21)导出的, 而这两者具有相同的形式。 ■

4.1.3 一般情况下的迭代计算

综合以上两种情形, 可以使用一种统一的方法来描述当新的障碍物添加到工作空间中时更新导航势函数。更具体地, 当第 k 个障碍物 $\mathcal{O}_k$ 添加到工作空间中后, 从丛林空间到圆形空间的递归变换如下:

$$\Phi_{\mathcal{F} \rightarrow \mathcal{M}}^{(k)}(q) \triangleq \Phi_{\mathcal{S} \rightarrow \mathcal{M}}^{(k)} \circ \Phi_{\mathcal{F} \rightarrow \mathcal{S}}^{(k)}(q), \quad (4.12)$$

其中如果新的障碍物 $\mathcal{O}_{k+1}$ 作为一个独立的障碍物添加到工作空间中时，需要使用公式(4.1)将变换 $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}^{(k)}$ 更新为 $\Phi_{\mathcal{S} \rightarrow \mathcal{M}}^{(k+1)}$ ；否则，如果新的障碍物 $\mathcal{O}_{k+1}$ 与已有的障碍物相重叠，就需要使用公式(4.10)将变换 $\Phi_{\mathcal{F} \rightarrow \mathcal{S}}^{(k)}$ 更新为 $\Phi_{\mathcal{F} \rightarrow \mathcal{S}}^{(k+1)}$ 。另外，每当一个新的独立的障碍物添加到工作空间中时，定义在点空间中的谐势场需要使用如下的递归函数进行更新：

$$\phi_p^{k+1}(x) = \frac{1}{K+1} \left(\phi(x, P_d) + \phi(x, P_{M+1}) \right) + \frac{K}{K+1} \phi_p^k(x), \quad (4.13)$$

其中函数 $\phi_p^k(x)$ 和 $\phi_p^{k+1}(x)$ 分别是第 k 步和第 $(k+1)$ 步的谐势场，函数 $\phi(x, P_{M+1})$ 是新添加的障碍物对应点在点空间中的谐函数。根据更新后的变换 $\Phi_{\mathcal{F} \rightarrow \mathcal{M}}^{(k)}(q)$ 和点空间中的谐函数 $\phi_p^{k+1}(x)$ ，可以相应地得到在新的环境下的导航函数为 $\phi_{\text{NF}}^{k+1}(q)$ 。

4.2 定向搜索树的局部更新

随着新的障碍物被检测到并添加到工作空间中，在第3.5节中构建的定向搜索树 $\mathcal{T}_{\hat{g}_\ell \rightarrow \hat{g}_{\ell+1}}$ 的某些部分可能会变得无效，因为其中的某些顶点可能会落在新的障碍物内。更重要的是，由于导航势函数随着环境不断更新，搜索树的某些边的相关代价需要重新进行估算。针对这些情况，本节描述了如何在检测到新的障碍物时在线地更新搜索树，而不是完全重新构建。

定向搜索树的在线更新包含三个主要的步骤：(I) 剪支。对于每个搜索树 $\mathcal{T}_{\hat{g}_\ell \rightarrow \hat{g}_{\ell+1}}$ ，找出其中位于新加入障碍物内的节点，即 $\mathbf{v} \in \mathcal{O}_k$ ，并将它们从顶点集 V 中移除。而与它们所连接的边也需要相应地从边集 E 中删除。因此，剩下的树可能会变成不连通的，那么从当前顶点到目标点的路径可能不再存在。(II) 基于势场的节点再生成。为了重新将定向搜索树与目标节点进行连通，就需要产生新的节点。可以使用与第3.5节中类似的方法生成节点，但是这些方法往往是在全局空间中进行。然而，取决于工作空间的变化方式，将搜索树的生长方向集中在已经修改过的障碍物区域会更加有效。由于谐势场的更新很大程度上取决于工作空间中局部变化的区域，因此可以在势场变化大的区域重新生成新节点。根据特定的工作空间离散化的形式，可以使用以下具有势场变化倾向的方法生成新节点：

$$b(q) \propto |\zeta^{k+1}(q) - \zeta^k(q)|, \quad (4.14)$$

其中 $\zeta^k(q) \triangleq \exp(1 - \frac{\mu}{\phi_{\text{NF}}^k(q)})$ ，且 $\zeta^{k+1}(q)$ 与 $\zeta^k(q)$ 的定义类似。上式中 $|\zeta^{k+1}(q) - \zeta^k(q)|$ 描述了势场在添加完障碍物 $\mathcal{O}_k$ 前后的变换，这样就可以让搜索树在势场变化较大的区域进行生长并与目标节点连通。(III) 路径修正。一旦产生新的节点添加到 V 中并与已有节点进行连接，与每条边相关的代价就需要根据势场的更新重新进行估计。为了更准确地计算这些代价值，本文提出了一种线性拟合的方法，根据机器人经过的顶点

之间的边的实际代价来调整代价函数(3.37)中的权重 $\mathbf{w}$ 。具体来说, 假设机器人已经穿过一组 M 对顶点的边, 其中实际代价的集合表示为 $\bar{\gamma}^* = [\gamma_1^*, \gamma_2^*, \dots, \gamma_M^*]$; 使用代价函数(3.37)计算出来的估计代价的集合表示为 $\bar{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_M]$; 其中直线距离代价的集合表示为 $\bar{d} = [d_1, d_2, \dots, d_M]$; 转向代价的集合表示为 $\bar{\mathbf{Q}} = [\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_M]$ 。然后, 通过以下方式更新代价权重:

$$\mathbf{w}^* = (\bar{\mathbf{Q}}^T \bar{\mathbf{Q}})^{-1} \bar{\mathbf{Q}}^T (\bar{\gamma}^* - \bar{d}). \quad (4.15)$$

因此, 这些最优权重 $\mathbf{w}^*$ 可以替换掉原始权重 $\mathbf{w}$ 来估计每条边的代价。之后, 可以通过相同的搜索算法找到从当前顶点到目标顶点的最优路径, 即 $\mathbf{P}_{\hat{g} \rightarrow \hat{g}'} = \mathbf{v}_t \cdots \mathbf{v}_{K-1} \mathbf{v}_G$ 。

4.3 导航图的局部更新

在更高的任务层面, 由于定向搜索树已经更新, 导航图 $\mathcal{G}$ 的可行性和代价也要相应地进行更新, 从而会导致不同的任务规划结果 $\hat{\mathbf{g}}$ 。具体而言, 导航图 $\mathcal{G}$ 分为两个步骤进行更新: (i) 根据路径 $\mathbf{P}_{\hat{g} \rightarrow \hat{g}'}$ 是否在搜索树 $\mathcal{T}_{\hat{g} \rightarrow \hat{g}'}$ 内重新评估从 $\hat{g}$ 到 $\hat{g}'$ 的导航可行性。如果不存在, 则将转换边 (g, g') 从 E 中删除。而这种情况通常是由于新发现的障碍物阻挡了一些自由通道造成的。(ii) 在新的搜索树内更新路径后, 根据代价函数(3.37)重新计算 E 中转换边的代价。因此, 可以说导航图 $\mathcal{G}$ 每次都必须根据最新的环境模型进行更新。给定更新的乘积自动机 $\hat{\mathcal{A}}$, 最优任务规划 $\hat{\mathbf{g}}^*$ 将重新被构建。这就意味着, 随着环境的逐渐探索, 被无限重复的任务计划后缀将渐近地收敛到最优。

第五章 算法总结与复杂度分析

5.1 算法总结

总的来说，在初始化构建的时候，首先构建与导航图 $\mathcal{G}$ 的每个转移边相关联的定向搜索树 $\{\mathcal{T}_{\hat{g} \rightarrow \hat{g}'}\}$ ，然后构建与每个 $\mathcal{T}_{\hat{g} \rightarrow \hat{g}'}$ 的边相关联的定向谐势场 $\{\Upsilon_{v \rightarrow v'}\}$ ，从而分层地构建导航图 $\mathcal{G}$ 。给定 $\mathcal{G}$ ，通过计算乘积自动机 $\hat{\mathcal{A}}$ ，可以得到初始的规划结果 $\hat{g}$ 。然后，通过跟踪与每个转移相关联的转移序列和与每个转移关联的边序列，分层地执行计划 $\hat{g}$ 。同时，当机器人探索到更多障碍物时，首先递归的更新谐势场 Υ ，然后局部地更新定向搜索树 $\{\mathcal{T}_{\hat{g} \rightarrow \hat{g}'}\}$ 和导航图 $\mathcal{G}$ 。因此，给定当前机器人的位姿和更新后的乘积自动机 $\hat{\mathcal{A}}$ ，任务规划 $\hat{g}$ 会进行动态地调整，以适应新的环境。这个迭代更新的过程将不断地重复进行，直到机器人完成所有给定任务。

5.2 复杂度分析

由于机器人的探索过程中需要实时更新导航图和定向搜索树等结构，因此需要考虑算法的计算复杂度。具体而言，离线初始化期间，构建乘积自动机 $\hat{\mathcal{A}}$ 的计算复杂度为 $\mathcal{O}(N^2M^2)$ ，其中 N 是感兴趣的区域的数量， M 是在 B_ϕ 中的状态数量。构建定向搜索树 $\mathcal{T}$ 的复杂度为 $\mathcal{O}(D)$ ，其中 D 是中间路径点的数量。对于具有 K 个星形障碍物和 H 颗星形树的环境，计算初始定向势场 Υ 的复杂度为 $\mathcal{O}(K^2 + H^3)$ ，因此计算完整的导航图 $\mathcal{G}$ 的复杂度为 $\mathcal{O}(N^2M^2 + D(K^2 + H^3))$ 。在线执行期间，每次添加障碍物时， Υ 的更新复杂度为 $\mathcal{O}(K + H)$ 。对于局部更新的 $\mathcal{G}$ 和 $\mathcal{T}$ ，它们的复杂度分别为 $\mathcal{O}(XM^2)$ 和 $\mathcal{O}(Y)$ ，其中 X 是需要添加的顶点数量， Y 是生成的中间路径点的数量。

第六章 仿真与实验

为了进一步验证本文提出的方法的有效性，本节进行了必要的数值仿真和硬件实验，并与前言的方法进行了定量比较。仿真在Intel Core i7-1280P CPU上使用Python3实现。硬件实验的平台将在后续小节中进行介绍。

6.1 数值仿真

首先，本文针对两种不同的障碍物环境和不同的时序任务进行了数值仿真。本节先对障碍物环境和复杂任务进行了描述，随后展示了仿真结果和对比结果，并对结果进行了定量的对比分析。

6.1.1 环境与任务描述

一个满足非完整约束模型的机器人被部署在一个 $4m \times 5m$ 的工作区域内，如图6.1所示。机器人的大小为 $0.2m \times 0.3m$ ，装备有虚拟的激光雷达传感器，最大检测范围为 $1m$ 。机器人实时的位置用 q_t 表示。控制器公式(3.36)中的控制增益设为 $k_v = 0.5$ 和 $k_\omega = 0.3$ 。导航势函数的公式(2.2)中的初始参数被设为 $K = 2$ 和 $\mu = 1$ 。如图6.1和图6.3所示，整个环境模拟了一个非平凡的办公室环境，其由许多重叠的方圆形的障碍物构成。最初，机器人预先知道工作区域的外部边界和几个内部障碍物。

此外，工作区域中有五个被标记为 r_1, r_2, r_3, r_4, r_5 的感兴趣区域，即任务区域。这里考虑了两个不同的任务，分别表示为LTL公式：(i) 机器人应以任意顺序经过这些感兴趣区域，即 $\varphi_1 = (\Diamond r_1) \wedge (\Diamond r_2) \wedge (\Diamond r_3) \wedge (\Diamond r_4) \wedge (\Diamond r_5)$ ；(ii) 机器人应从存储室 r_1 和 r_2 中的任意一个将物体运输到目的地 r_3, r_4 和 r_5 ，即 $\varphi_2 = \Diamond((r_1 \vee r_2) \wedge (\Diamond r_3)) \wedge \Diamond((r_1 \vee r_2) \wedge (\Diamond r_4)) \wedge \Diamond((r_1 \vee r_2) \wedge (\Diamond r_5))$ 。可以看出，有许多方式可以完成这些给定的任务，而其运行的代价高度依赖于实际工作空间的结构。

6.1.2 仿真结果

在第一个场景中，实际的工作区包含17个障碍物，如图6.1所示，机器人从已知6个障碍物的工作空间中，从起始位置 $q_0 = (0.5m, 0.45m)$ 开始运动。初始时刻，与 φ_1 相关的NBA具有32个状态和243个边。与感兴趣区域相关的有限转移系统初始化为全连接的图，并且每条边的权重由欧几里得距离估计。因此，两者乘积具有192个状态和1435条边，其中最优任务规划结果是 $P_0^* = q_0, r_1, r_2, r_3, r_4, r_5$ 。在这个上层任务规划

的引导下，机器人以第一个任务点 r_1 为当前目标点，使用与公式 (3.9) 相关的转换来构建初始导航势函数 $\varphi_{NF}^0(q)$ ，在工作空间的每个点上，导航势函数 $\varphi_{NF}^0(q)$ 的平均计算时间为 $13.89ms$ ，相应的等高线以绿色曲线表示，如图 6.1 所示。给定 $\varphi_{NF}^0(q_t)$ ，利用 (3.36) 中的非线性控制器跟随定向势场 $\Upsilon(q_t)$ 驱动机器人朝着当前任务运动，机器人的实时轨迹用红色曲线表示。随着机器人逐步执行当前规划并探索未知的工作空间，

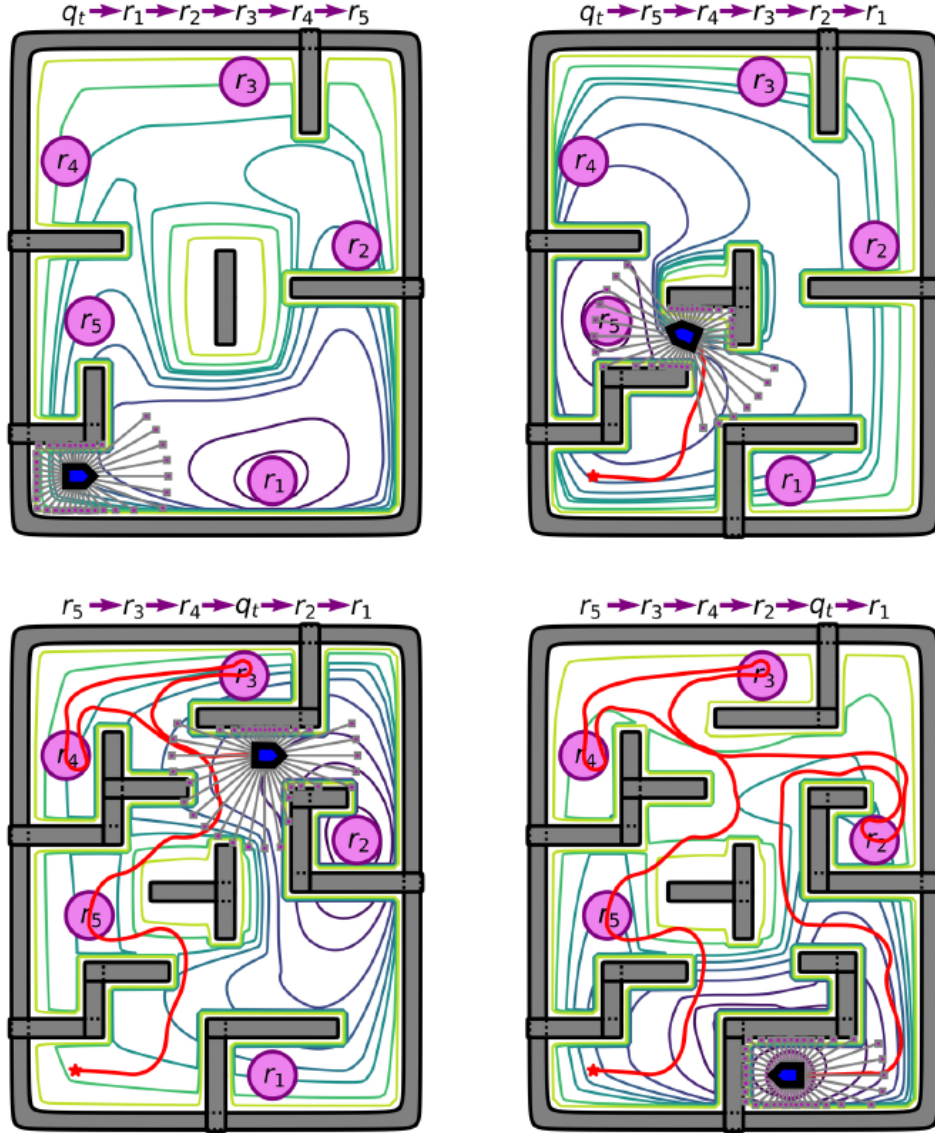


图 6.1 场景一：机器人以任意顺序经过任务点。

新的障碍物在离散时间点被激光传感器检测到，如图 6.1 所示。在这些离散时刻 t_k ，对于 $k = 1, 2, \dots, 11$ ，导航图 $\mathcal{G}$ 被相应地更新以满足可行性和代价最优性。如图 6.1 所示，实时的任务计划相应地更新为 $P_4^* = q_{t_4}, r_5, r_4, r_3, r_2, r_1$ 和 $P_9^* = q_{t_9}, r_2, r_1$ 。给定新的任务规划结果，通过 (4.1) 中描述的方法增量地计算 t_k 时刻的导航势函数 $\varphi_{NF}^k(q)$ ，其中转换 $\Phi_{S \rightarrow M}^k$ 和 $\Phi_{F \rightarrow S}^k$ 可以通过迭代计算的方法得到。图 6.2 显示了本文所提的增量式的

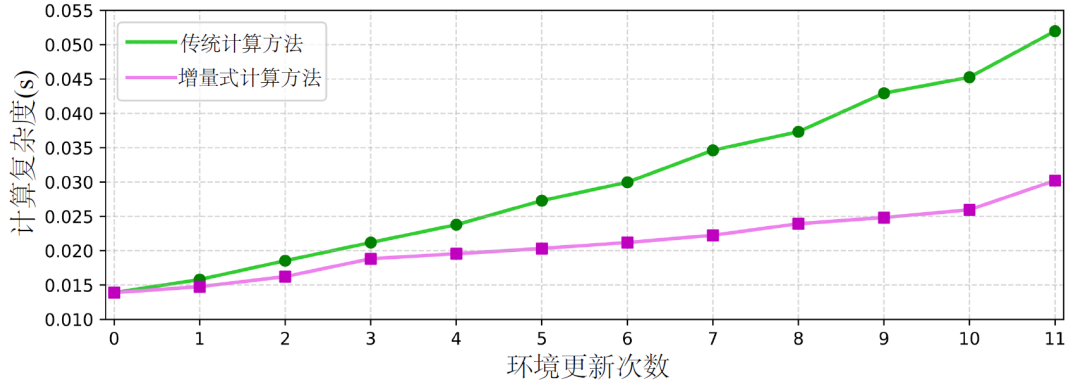


图 6.2 导航势函数的平均计算时间。粉色线条对应的是本文所提增量式的计算方法；绿色的线条对应的是传统的计算方法。

方法和传统的方法 [18, 41] 计算 $\phi_{NF}^k(q)$ 的平均时间。传统的计算方法是随着环境的更新每次都完全重新计算导航势函数，因此计算复杂度较高。从图 6.2 中的结果可以看出，所提方法在计算时间的减少是显著的，这意味着该方法更适合于在线导航。在每次更新后，机器人沿着当前的定向谐势场 $\Upsilon(q_t)$ 继续导航。所有任务在 115.1s 内完成，产生的平滑轨迹的总距离为 23.29m，如图 6.1 所示。

6.1.3 对比

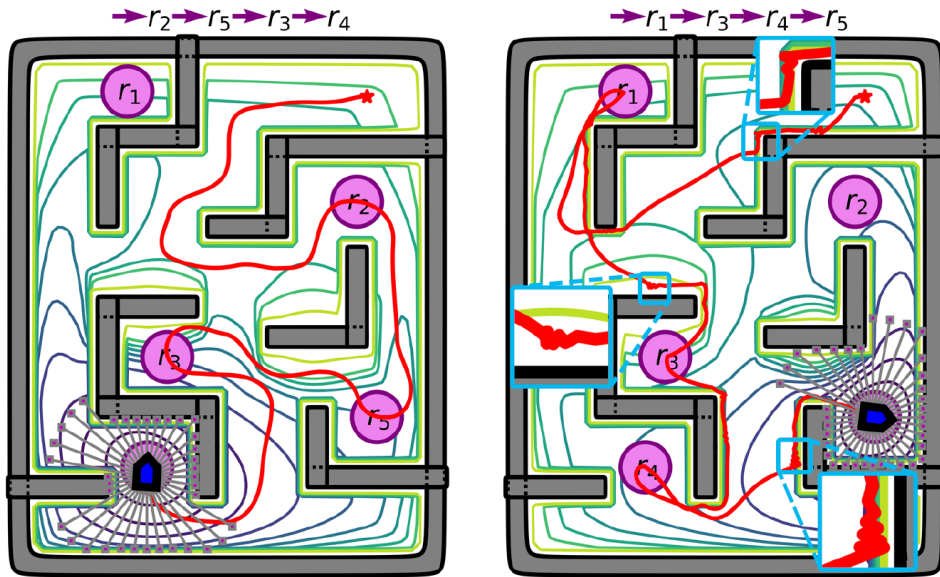


图 6.3 场景二：机器人从储存室搬运并配送物品。左图为本文所提方法运行的结果；右图为传统导航势场法运行的结果。

为了一步展示本文的方法的有效性，下面将所提方法与没有高层任务迭代的常规方法，如文献 [18, 41] 中的方法进行了比较。在这个场景中，实际环境由 14 个重

叠的障碍物组成,如图 6.3所示。机器人从坐标 $(3.4m, 4.6m)$ 开始,运动前已经预先知道6个障碍物。初始时刻的任务规划方法与上一个场景相同,得到的规划结果为 $P_0^* = r_1, r_3, r_4, r_5$ 。在线执行和迭代更新过程中,本文提出的方法动态地调整了高层规划器和底层的势场。最终轨迹如图 6.3所示,且最终的任务规划结果为 $P_{14}^* = r_2, r_5, r_3, r_4$ 。相反,所对比的方法,即没有定向搜索树和高层任务规划引导的传统导航势函数法,对初始计划的盲目执行导致了高代价的轨迹,如图 6.3所示。此外,正如 [9]所指出的那样,传统的导航势函数在在线执行过程中往往会产生振荡轨迹,这在图 6.3的右图中凸显了出来。相比之下,本文所采用的方法不仅产生了平滑的轨迹,而且得到了高效的规划结果。提出方法的总执行时间和轨迹代价分别为83.2s和14.36m,而传统方法则需要114.5s和16.18m。

6.2 硬件实验

为了进一步验证所提方法在现实环境和机器人硬件平台上的运行效果,本文在实验室的平台上开展了硬件实验。

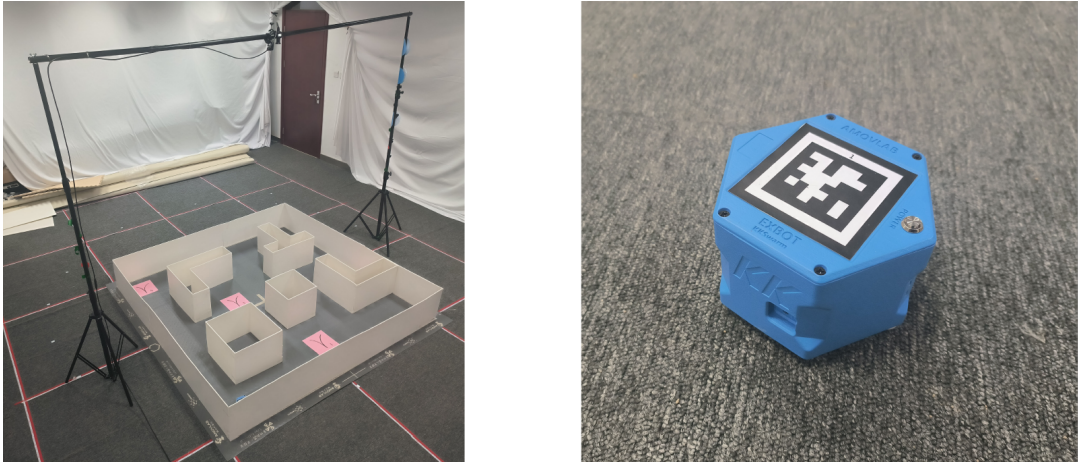


图 6.4 硬件平台。左图为平台环境;右图为无人小车。

6.2.1 硬件平台介绍

本文所使用的硬件平台称为KK-Swarm^①,是基于ROS的一款机器人运动控制硬件平台,如图 6.4所示。该平台使用的无人小车为差速驱动小车,尺寸的大小为 $119.75mm \times 105.01mm \times 79.07mm$,实车如图 reffig:platform所示。小车的坐标位置通过海康威视相机和二维码检测工具获取并传送到主机。对小车发送控制指令速度 v 和角速度 ω 可以实现运动控制,其最大线速度为 $v_{\max} = 0.6m/s$ 。

① 详细介绍参考平台开源项目: <https://wiki.amovlab.com/public/misaro-doc/>

6.2.2 环境与任务描述

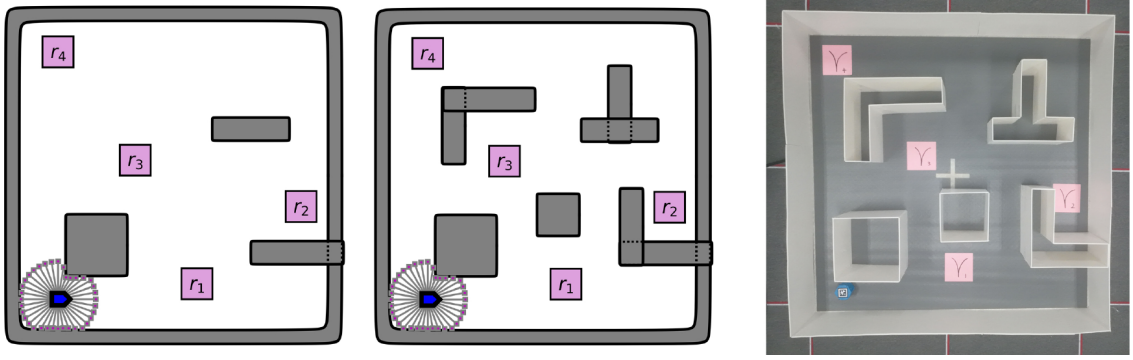


图 6.5 障碍物环境。左图为仿真环境中的初始环境；中图为仿真环境中的真实环境；右图为硬件平台中的环境。

如图 6.5 所示，机器人所处的真实环境由 8 个障碍物和外部工作空间边界组成，其中整体工作空间的尺寸大小为 $2.0m \times 2.0m$ ，障碍物的尺寸大小不等。初始时刻，机器人只知道部分外部工作空间和部分障碍物，如图 6.5 中左图所示。机器人的起始位置在 $(0.5m, 0.4m)$ 点，可以通过虚拟雷达检测环境中未知的障碍物，且雷达的检测范围为 $0.25m$ 。导航势函数的公式 (2.2) 中的初始参数被设为 $K = 3$ 和 $\mu = 1$ 。机器人的控制输入由公式 (3.36) 计算得到，且公式中的控制增益设为 $k_v = 0.2$ 和 $k_\omega = 0.3$ 。控制输入的采样时间间隔为 $0.1s$ 。

在工作空间中，有四个任务区域，记作 r_1, r_2, r_3, r_4 ，如图 6.5 中的右图所示。机器人需要以任意顺序经过这些区域以完成特定任务，该任务可以通过线性时序逻辑公式描述为： $\varphi = (\diamond r_1) \wedge (\diamond r_2) \wedge (\diamond r_3) \wedge (\diamond r_4)$ 。

6.2.3 实验结果

实验的整体运行结果如图 6.6(a)-(f) 所示。初始时刻，任务规划的最优结果为 $P^* = r_1, r_2, r_3, r_4$ ，在该结果的引导下，机器人沿着定向势场朝着 r_1 运动，当机器人检测到新的障碍物时，相应的任务规划的结果依次为 $P^* = r_1, r_3, r_4, r_2$ ，和 $P^* = r_1, r_3, r_4, r_2$ 。最终机器人以较优的时间代价和路径代价完成所有任务，如图 6.6(f) 所示。运行的总时间和总路程分别为 $173.28s$ 和 $8.32m$ 。根据实验结果可知，机器人在现实中的部分已知障碍物环境里面能够安全、平滑地完成所有任务。

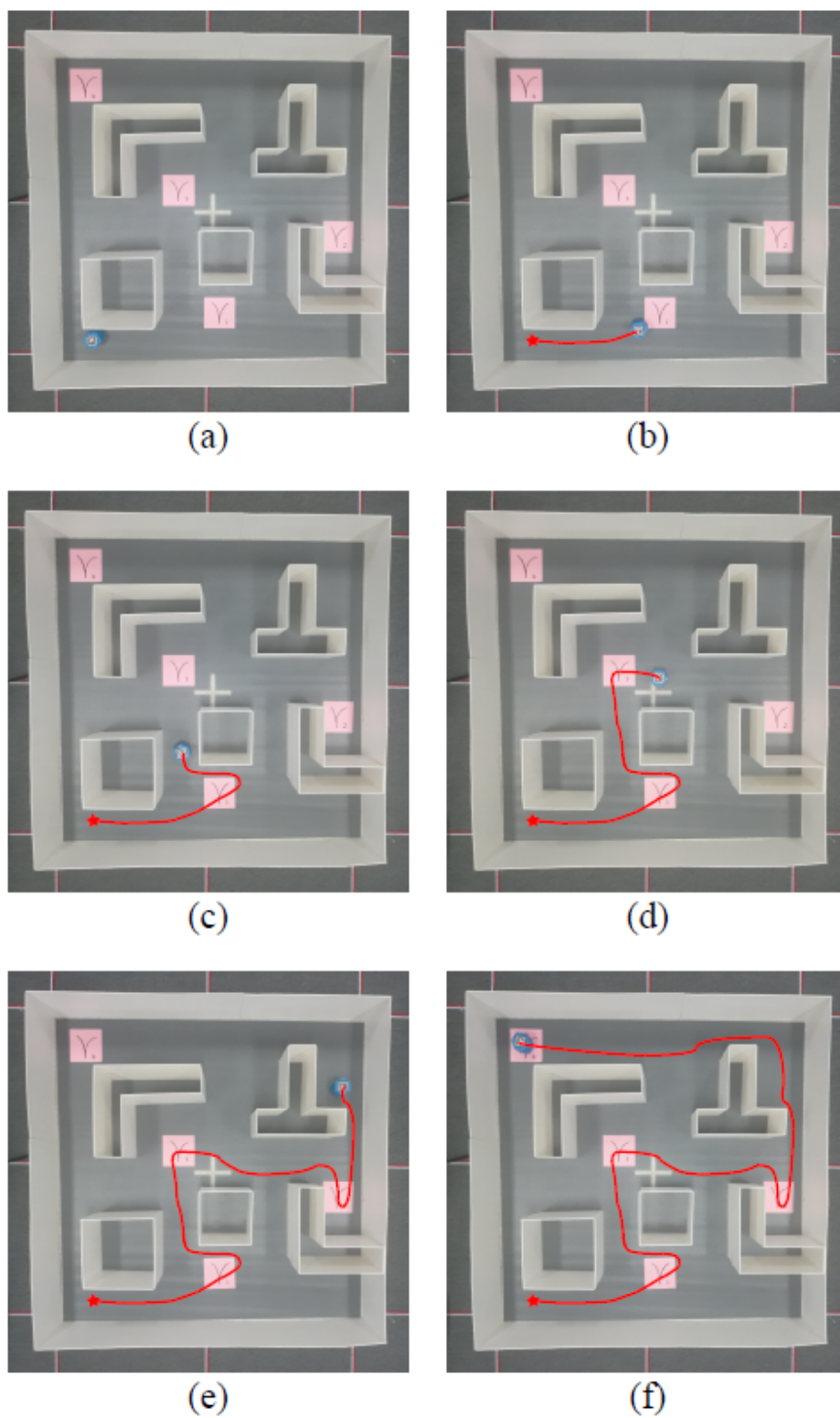


图 6.6 实验结果。机器人的实时运行轨迹由红色曲线绘出。

第七章 结论

本研究提出了一种集成任务规划与运动规划的混合式的控制规划框架，称为双层导航树，它结合了形式化方法进行任务规划和定向谐势场进行在线导航。与已有的方法相比本研究的主要优势在于：

（I）提出了一种新颖的规划控制结构，将任务规划和运动控制统一起来，能保证全局导航的安全性和收敛性。

（II）提出了一种新的导航势函数的构建方法，可用于重叠的障碍物环境，并针对在线情况对该方法进行了优化，可以随着障碍物的增加迭代更新势场；

（III）设计了一种基于定向谐势场的新型非线性控制器，适用于现实生活中常见的欠驱动移动机器人，并且能确保机器人的安全和平稳运动；

（IV）提出了一种混合式的在线迭代方法，可同时递归地更新谐势场、定向搜索树以及任务导航图。

针对所提框架，本文从理论分析、数值仿真和硬件实验上进行了充分的分析和验证，并与传统方法进行了理论上和仿真上的对比。对比和实验结果表明，所提方法能实现机器人在未知环境中高效地完成复杂任务，与已有的方法相比具有明显的优越性。此外，得益于所提方法的优点，其在现实场景中具有较高的应用价值，比如自主导航机器人、无人配送和物流运输等场景。未来的工作考虑将现有方法与动态环境、多机器人系统、无人机导航、深度学习等相关领域相结合，以实现更高级别的自主导航和规划能力。

参考文献

- [1] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7), 846–894.
- [2] LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.
- [3] Koditschek, D. (1987). Exact robot navigation by means of potential functions: Some topological considerations. *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, 4, 1–6.
- [4] Khatib, O. (1999). Mobile manipulation: The robotic assistant. *Robotics and Autonomous Systems*, 26(2-3), 175–183.
- [5] Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. Elsevier.
- [6] Fainekos, G. E., Girard, A., Kress-Gazit, H., & Pappas, G. J. (2009). Temporal logic motion planning for dynamic robots. *Automatica*, 45(2), 343–352.
- [7] Garrett, C. R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L. P., & Lozano-Pérez, T. (2021). Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4, 265–293.
- [8] Kim, B., Shimanuki, L., Kaelbling, L. P., & Lozano-Pérez, T. (2022). Representation, learning, and planning algorithms for geometric task and motion planning. *The International Journal of Robotics Research*, 41(2), 210–231.
- [9] Loizou, S. G., & Rimon, E. D. (2022). Mobile robot navigation functions tuned by sensor readings in partially known environments. *IEEE Robotics and Automation Letters*, 7(2), 3803–3810.
- [10] Rousseas, P., Bechlioulis, C., & Kyriakopoulos, K. (2022). Trajectory planning in unknown 2d workspaces: A smooth, reactive, harmonics-based approach. *IEEE Robotics and Automation Letters*.
- [11] Baier, C., & Katoen, J.-P. (2008). *Principles of model checking*. MIT press.
- [12] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles* (pp. 396–404). Springer.
- [13] Warren, C. W. (1989). Global path planning using artificial potential fields. *IEEE International Conference on Robotics and Automation*, 316–317.
- [14] Panagou, D. (2014). Motion planning and collision avoidance using navigation vector fields. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2513–2518.
- [15] Rousseas, P., Bechlioulis, C. P., & Kyriakopoulos, K. J. (2022). Optimal motion planning in unknown workspaces using integral reinforcement learning. *IEEE Robotics and Automation Letters*.
- [16] Kim, J.-o., & Khosla, P. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE transactions on robotics and automation*, 8(3), 338–349.
- [17] Loizou, S. G. (2011). Closed form navigation functions based on harmonic potentials. *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 6361–6366.
- [18] Loizou, S. G. (2017). The navigation transformation. *IEEE Transactions on Robotics*, 33(6), 1516–1523.
- [19] Vlantis, P., Vrohidis, C., Bechlioulis, C. P., & Kyriakopoulos, K. J. (2018). Robot navigation in complex workspaces using harmonic maps. *IEEE International Conference on Robotics and Automation (ICRA)*, 1726–1731.

- [20] Rouseas, P., Bechlioulis, C., & Kyriakopoulos, K. J. (2021). Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning. *IEEE Robotics and Automation Letters*, 6(2), 2005–2011.
- [21] Hsu, D., Latombe, J.-C., & Kurniawati, H. (2006). On the probabilistic foundations of probabilistic roadmap planning. *The International Journal of Robotics Research*, 25(7), 627–643.
- [22] Janson, L., Schmerling, E., Clark, A., & Pavone, M. (2015). Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *The International journal of robotics research*, 34(7), 883–921.
- [23] Ko, I., Kim, B., & Park, F. C. (2013). Vf-rrt: Introducing optimization into randomized motion planning. *2013 9th Asian Control Conference (ASCC)*, 1–5.
- [24] Qureshi, A. H., & Ayaz, Y. (2016). Potential functions based sampling heuristic for optimal path planning. *Autonomous Robots*, 40(6), 1079–1093.
- [25] Tahir, Z., Qureshi, A. H., Ayaz, Y., & Nawaz, R. (2018). Potentially guided bidirectionalized rrt* for fast optimal path planning in cluttered environments. *Robotics and Autonomous Systems*, 108, 13–27.
- [26] Filippidis, I., & Kyriakopoulos, K. J. (2011). Adjustable navigation functions for unknown sphere worlds. *IEEE Conference on Decision and Control and European Control Conference*, 4276–4281.
- [27] Ogren, P., & Leonard, N. E. (2005). A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics*, 21(2), 188–195.
- [28] Li, C., & Tanner, H. G. (2018). Navigation functions with time-varying destination manifolds in star worlds. *IEEE Transactions on Robotics*, 35(1), 35–48.
- [29] Otte, M., & Frazzoli, E. (2016). Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning. *The International Journal of Robotics Research*, 35(7), 797–822.
- [30] Shen, Z., Wilson, J., Harvey, R., & Gupta, S. (2021). Smarrt: Self-repairing motion-reactive anytime rrt for dynamic environments. *arXiv preprint arXiv:2109.05043*.
- [31] Guo, M., Andersson, S., & Dimarogonas, D. V. (2018). Human-in-the-loop mixed-initiative control under temporal tasks. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6395–6400.
- [32] Lindemann, L., Matni, N., & Pappas, G. J. (2021). Stl robustness risk over discrete-time stochastic processes. *2021 60th IEEE Conference on Decision and Control (CDC)*, 1329–1335.
- [33] Guo, M., & Dimarogonas, D. V. (2015). Multi-agent plan reconfiguration under local ltl specifications. *The International Journal of Robotics Research*, 34(2), 218–235.
- [34] Luo, X., Kantaros, Y., & Zavlanos, M. M. (2021). An abstraction-free method for multirobot temporal logic optimal control synthesis. *IEEE Transactions on Robotics*, 37(5), 1487–1507.
- [35] Leahy, K., Serlin, Z., Vasile, C.-I., Schoer, A., Jones, A. M., Tron, R., & Belta, C. (2021). Scalable and robust algorithms for task-based coordination from high-level specifications (scratches). *IEEE Transactions on Robotics*, 38(4), 2516–2535.
- [36] Audin, M., Damian, M., & Ern , R. (2014). *Morse theory and floer homology*. Springer.
- [37] Loizou, S. G., & Rimon, E. D. (2021). Correct-by-construction navigation functions with application to sensor based robot navigation. *arXiv preprint arXiv:2103.04445*.
- [38] Rimon, E. (1990). *Exact robot navigation using artificial potential functions* [Doctoral dissertation, Yale University].

- [39] Valbuena, L., & Tanner, H. G. (2012). Hybrid potential field based control of differential drive mobile robots. *Journal of intelligent & robotic systems*, 68(3), 307–322.
- [40] Khalil, H. (2002). *Nonlinear systems; 3rd ed.* Prentice Hall.
- [41] Rimon, E., & Koditschek, D. E. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5).

致谢

岁月不居，时节如流。转眼之间，在北大的四年时光到了最后的阶段。如同匆匆而过的云烟，岁月的车轮在不知不觉中驶向了毕业的彼岸。回首往昔，思绪万千，不禁为自己的成长和进步而欣慰。仍记得四年前，一个稚嫩的高中生，怀揣着憧憬和迷茫，踏入了这座理想与现实交汇的学术殿堂，到如今，已成了一名即将毕业的本科生。在四年的时间里，我曾经经历过挫折与困惑，也品尝过成功与喜悦，更深深体会到了自己的成长和变化。在这段求学岁月中，我收获了诸多老师、同学、家人和朋友的支持和帮助，在此向他们致以最诚挚的感谢。

首先，我要衷心感谢我的指导老师，国萌研究员。国老师亦师亦友，在科研、学习和生活中给予了我很大帮助。科研上，他不仅在选题阶段给予了我宝贵的建议，还在整个研究过程中给予了耐心的指导。他的深入洞察和精准的引导帮助我解决了许多科研中遇到的困难和疑惑，使我能够深入思考、准确分析并得出有价值的结论。在论文撰写和修改方面，他对我的写作进行了仔细审阅和评价，不厌其烦地帮助我改进论文的结构和表达，大大提升了我的学术写作水平。除了学术指导，国老师还给予我宝贵的人生指导和生活上的帮助。他经常与我交流，分享科研经验和人生智慧，让我受益匪浅。在节假日期间，国老师经常组织课题组聚餐、郊游等活动，丰富了大家的业余生活，缓解我们学习和科研上的压力。在他身上，我看到了科研工作者对学术的执着追求和对学生的关心支持。与此同时，也非常感谢课题组的同窗们，感谢他们在科研和生活上的陪伴和支持。在课题组的日子里，师兄师弟们是我学术道路上最亲密的伙伴和最可靠的支持者。无论是在每次组会上的讨论，还是在论文写作和解决关键问题时，他们总是乐于分享自己的经验，为我提供了许多宝贵的建议和帮助。

其次，还要感谢与我并肩奋斗的室友。在这段共同生活的日子里，他们的日夜陪伴让我感到温暖和安心。我们一起分享生活和学习中的喜怒哀乐，共同度过了无数个难忘的瞬间。他们的陪伴是我的大学生活中最美好的回忆。

此外，感谢我的父母以及众多的亲朋好友们。在我求学的旅程中，父母一直是我坚强的后盾和最无私的支持者。他们始终给予我无条件的爱和关怀，对我进行了无微不至的支持和鼓励。无论我面临怎样的困难和挑战，他们总是毫不犹豫地站在我身边，给予我坚定的支持和信心。我也要感谢那些给予了我帮助和支持的亲朋好友们，他们的理解和鼓励让我感到无比温暖和坚定，激励着我勇往直前。

最后，感谢四年来坚持不懈的自己，完成了这段意义非凡的学习之旅。

行文至此，落笔为终，致敬之。

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：王冲

日期：2023年6月1日

学位论文使用授权说明

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文；

论文作者签名：王冲 导师签名：国萌

日期：2023年6月1日