分类号：TP242　　学校代码：10363
密　级：公开　　学　　号：2220910104

安徽工程大学
Anhui Polytechnic University

硕士学位论文

题目 移动机器人路径规划算法研究

论 文 作 者　舒莹
校 内 导 师　曹雏清
校 外 导 师　刘志恒
专业学位类别　计算机技术
研究方向（领域）　智能系统与应用

论文提交日期：2025 年 5 月 9 日

分类号：TP242　　单位代码：10363
密　级：公开　　学　　号：2220910104

移动机器人路径规划算法研究

RESEARCH ON PATH PLANNING ALGORITHM FOR MOBILE ROBOT

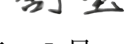
学　生　姓　名：舒 莹
校　内　教　师：曹雏清
校　外　教　师：刘志恒

专业 学位 类别：计算机技术
研究方向（领域）：智能系统与应用

论文答辩日期：2025 年 5 月 10 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。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。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名： 

日期：　　2025 年　5 月　9 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 □，在 ____ 年解密后适用本版权书。

本学位论文属于

不保密 ☑。

学位论文作者签名：舒莹　　指导教师签名：

日期：2025 年 5 月 9 日　　日期：2025 年 5 月 9 日

移动机器人路径规划算法研究

摘 要

随着计算机科学、智能算法和自主控制技术不断取得新突破，移动机器人智能系统在路径决策方面取得了显著进步，这些进步不仅提升了机器人的智能化水平，更为其在更多领域的应用提供了广阔的空间和无限的可能。本文聚焦于移动机器人全局与局部路径规划的核心问题，分层路径规划架构设计，在全局规划中，通过改进的 RRT 算法生成的全局优化路径，其关键路径点序列被定义为分段式子目标点，这些子目标点构成从起点到终点的宏观指导路径，为局部路径规划提供分阶段导航基准；而在局部规划方面，基于改进的 DQN 算法以当前子目标点为短期目标，通过感知实时障碍物信息和环境状态，动态调整局部轨迹。本文的主要研究工作如下：

（1）全局路径规划算法的研究

针对传统 RRT 算法随机采样效率低、环境适应性不足的缺陷，本文结合人工势场法和动态步长调节机制进行改进。在 RRT 算法的采样优化层面，本研究提出融合势场导向机制的改进框架。通过建立目标吸引力场、障碍排斥力场　及随机扰动场　的协同作用模型，构建具有动态权重分配策略的混合势场；该策略在采样阶段通过目标点的牵引作用，同时根据障碍物分布动态调整各分量的权重系数，使随机树在规避障碍的同时保持向目标区域推进的趋势。为提升算法收敛效率，提出动态步长调节方法，当扩展节点处于低势场梯度区域时，采用较大步长加速空间探索；当接近高势场变化区域时，缩小步长以提高避障精度。该机制与势场引导形成协同作用，既缓解了固定步长

导致的震荡现象，又降低了复杂环境中的无效采样概率。最后针对RRT算法生成路径存在的轨迹平滑性不足、冗余点过多，本研究引入贪心算法对初始路径进行优化处理，通过局部最优决策策略，逐次剔除冗余节点并优化生成树点连接轨迹。

（2）局部路径规划算法的研究

针对DQN算法中存在的Q值过估计和策略选择精度不足问题，提出了一种基于双重价值评估机制的深度强化学习改进框架。通过构建动作选择网络与目标价值网络分离的协同架构，并将其与传统DQN目标网络相结合，从而有效缓解Q值过估计问题，提高策略决策的准确性与稳定性。与此同时，本研究还设计了具备自适应功能的ε-greedy策略，达成了探索与利用之间的动态平衡，从而提升策略选择的精度。在训练初期，不断提升环境探索能力，后期则逐步收敛至最优策略，形成渐进式的决策优化。此外为增强决策评估的有效性，对奖励函数进行重构，引入奖励平滑机制与惩罚项，确保算法在路径决策时能准确量化动作的长期价值。这种改进方案既能规避局部最优陷阱，又增强了系统在动态场景下的抗干扰能力。通过仿真实验对比分析，相较于DQN算法，改进算法不仅显著缓解了Q值高估现象，而且增强了算法的稳定性。

关键词：路径规划，快速扩展随机树，深度强化学习，Q学习，DQN算法

RESEARCH ON PATH PLANNING ALGORITHM FOR MOBILE ROBOT

ABSTRACT

With the continuous breakthrough of computer science, intelligent algorithm and autonomous control technology, the intelligent system of mobile robot has made significant progress in path decision. These advances not only improve the intelligent level of robot, but also provide a broad space and infinite possibility for its application in more fields. This paper focuses on the core problem of global and local path planning for mobile robots, and the design of hierarchical path planning architecture. In the global planning, the global optimal path generated by the improved RRT algorithm is defined as a sequence of critical path points, which is defined as a piecewise sub-goal point. It provides a phased navigation benchmark for local path planning. In terms of local planning, the improved DQN algorithm takes the current subgoal point as the short-term goal, and dynamically adjusts the local trajectory by sensing real-time obstacle information and environmental state. The main research work of this paper is as follows:

(1) Research on global path planning algorithm

Aiming at the defects of low random sampling efficiency and insufficient environmental adaptability of traditional RRT algorithm, this paper combines artificial potential field method and dynamic step size adjustment mechanism to improve it. At the sampling optimization level of the RRT algorithm, this study proposes an improved framework that integrates the potential field guidance mechanism. The synergy model of

target attractive field, obstacle repulsive force field and random disturbance field was established, and the hybrid potential field with dynamic weight allocation strategy was constructed. In the sampling stage, the strategy uses the traction effect of the target point, and dynamically adjusts the weight coefficient of each component according to the distribution of obstacles, so that the random tree can avoid obstacles while maintaining the trend of advancing to the target area. In order to improve the convergence efficiency of the algorithm, a dynamic step size adjustment method was proposed. When the expansion node was in the low potential field gradient region, a larger step size was used to accelerate the space exploration. When approaching the region of high potential field variation, the step size is automatically reduced to improve the accuracy of obstacle avoidance. The mechanism and potential field guidance form a synergistic effect, which not only alleviates the oscillation phenomenon caused by fixed step size, but also reduces the invalid sampling probability in complex environments. Finally, in view of the lack of trajectory smoothness and too many redundant points in the path generated by the RRT algorithm, this study introduces a greedy algorithm to optimize the initial path, and iteratively eliminates redundant nodes and optimizes the connection order of the generated tree points through the local optimal decision strategy.

(2) Research on local path planning algorithm

In order to solve the problem of overestimation of Q value and insufficient accuracy of strategy selection in DQN algorithm, an improved framework of deep reinforcement learning based on dual value evaluation mechanism is proposed. By constructing a collaborative architecture that separates the action selection network from the target value network, and combining it with the traditional DQN target network,

the problem of Q-value overestimation is effectively alleviated and the accuracy and stability of policy decision-making are improved. At the same time, this study also designs a ε-greedy strategy with adaptive function to achieve a dynamic balance between exploration and exploitation, so as to improve the accuracy of strategy selection. In the early stage of training, the ability of environmental exploration is constantly improved, and in the later stage, it gradually converges to the optimal strategy, forming a progressive decision-making optimization. In addition, in order to enhance the effectiveness of decision evaluation, the reward function is reconstructed, and the reward smoothing mechanism and penalty term are introduced to ensure that the algorithm can accurately quantify the long-term value of actions during path decision. This improved scheme can not only avoid the local optimum trap, but also enhance the anti-interference ability of the system in dynamic scenarios. Through the comparative analysis of simulation experiments, compared with the DQN algorithm, the improved algorithm not only significantly alleviates the overestimation of Q value, but also enhances the stability of the algorithm.

KEY WORDS : path planning, rapidly-expanding random tree, deep reinforcement learning, Q-learning, DQN algorithm

目录

摘 要 I
ABSTRACT III
目录 VI
第 1 章 绪论 1
1.1 研究背景及意义 1
1.2 路径规划研究现状 2
1.2.1 全局路径规划算法 2
1.2.2 局部路径规划算法 6
1.2.3 深度强化学习算法 10
1.3 路径规划概述 11
1.3.1 路径规划问题 11
1.3.2 移动机器人路径规划特点 12
1.4 本文主要研究内容 12
1.5 本文的组织结构 13
第 2 章 深度强化学习算法相关理论 15
2.1 深度学习算法 15
2.1.1 深度神经网络模型 15
2.1.2 卷积神经网络 19
2.2 强化学习算法 20
2.2.1 马尔科夫决策过程 21
2.2.2 贝尔曼方程 21
2.2.3 强化学习基本函数 22
2.3 本章小结 25
第 3 章基于引力导向动态步长 RRT 算法 26
3.1 RRT 算法介绍 26
3.1.1 RRT 算法的基本原理 26
3.1.2 RRT 算法流程及实现 27

3.1.3 RRT 算法优缺点分析 28
3.2 改进的 RRT 算法 29
3.2.1 引力导向策略 29
3.2.2 动态步长 33
3.3.3 节点的优化 35
3.3.4 改进算法实现 36
3.3 仿真实验与分析 36
3.3.1 环境搭建及参数设置 36
3.3.2 实验仿真与结果分析 37
3.4 本章小结 42
第 4 章 改进的 DQN 路径规划算法 44
4.1 Q-Learning 算法 44
4.1.1 Q-Learning 原理 44
4.1.2 算法收敛性分析 46
4.2 DQN 算法 47
4.2.1 神经网络拟合 Q 值表 47
4.2.2 经验回放机制 48
4.2.3 目标网络 49
4.3 改进 DQN 算法 50
4.3.1 目标网络优化 50
4.3.2 奖励函数设计 51
4.3.3 探索策略设计 51
4.4 仿真实验分析 52
4.4.1 环境搭建及参数设置 52
4.4.2 实验结果与分析 53
4.5 本章小结 59
第 5 章 移动机器人路径规划算法实验验证 60
5.1 移动机器人平台体系架构 60
5.2 混合规划算法流程 61

5.3 路径规划算法的实验验证 63
5.4 本章小结 65
第 6 章 总结与展望 67
6.1 工作总结 67
6.2 工作展望 67
参考文献 69
攻读学位期间取得的科研成果 75
致谢 76

第 1 章 绪论

1.1 研究背景及意义

移动机器人技术是人工智能与自动化交叉融合的重要分支之一，近年来在工业搬运、农业作业、军事侦察、海洋探测、医疗辅助、仓储物流等多个领域展现出广阔的应用前景[1-2]。伴随《中国制造 2025》等国家战略的持续推进，移动机器人已成为推动制造业升级和智能化转型的重要支撑力量[3]。在众多关键技术中，路径规划作为实现机器人自主导航与作业调度的核心环节，其优劣直接关系到机器人的任务执行效率、安全性与环境适应能力[4-5]。

路径规划的目标是在环境中为机器人设计一条从起点到目标点的无碰撞、最优或次优路径。尽管近年来路径规划技术取得了一定进展，但在动态变化或未知环境下仍面临诸多挑战，尤其在高效避障和实时响应方面难以满足实际需求[6]。传统方法如 A*、Dijkstra 等依赖静态地图信息，难以应对突发障碍物或环境动态变化；而基于采样的算法如 RRT 与 PRM 虽在高维空间中具有较强的搜索能力，但往往存在路径不光滑、搜索效率低和收敛慢等问题。此外进化算法如遗传算法在依赖初始种群质量的同时，也存在局部收敛风险，限制了其在复杂场景中的应用效果。目前多数传统方法缺乏自主学习和适应能力，导致在环境不确定性增强或任务复杂度提高时，规划性能迅速下降。强化学习作为近年来兴起的一种智能决策方法，通过环境交互不断学习策略，为动态路径规划问题提供了新的解决思路，特别是在未知环境下，强化学习具有良好的自适应性和鲁棒性。然而强化学习算法在实际应用中也存在一定瓶颈，如奖励函数设计复杂、学习过程缓慢、仿真与现实存在性能差异等问题。在此基础上，深度强化学习（Deep Reinforcement Learning, DRL）应运而生，它结合了深度学习在感知与特征提取方面的优势，与强化学习的策略优化机制相融合，使得机器人能够在高维感知输入下进行端到端的路径决策。DRL 通过智能体与环境持续交互、自主试错的方式，逐步学习最优路径策略，尤其适用于动态障碍物频繁出现、环境信息不完备的场景。在路径规划任务中，DRL 不仅能够处理复杂的状态空间，还具备良好的泛化能力和实时适应能力，有效提升了机器人在复杂环境中的导航效率。其在处理高维传感器数据、动态避障、局部路径优化等方面均表现出较强的优势。尽管目前 DRL 仍

面临如训练数据量大、策略稳定性不足、收敛速度慢等现实问题，但随着计算能力的提升和算法结构的不断优化，这些障碍正逐步被克服。

路径规划作为实现移动机器人自主导航的重要支柱，正朝着智能化、自适应与实时响应方向持续发展。深度强化学习凭借其强大的感知处理能力和策略学习能力，为解决复杂动态环境下的路径规划问题提供了创新思路，也将成为推动下一代智能机器人系统发展的关键技术之一。

1.2 路径规划研究现状

移动机器人的自主导航技术已广泛应用于制造、国防和应急救援等关键领域，其核心在于路径规划能力，这直接影响机器人在复杂环境中的任务执行效率。路径规划可分为全局路径规划和局部路径规划两类[7]，全局路径规划基于对已知环境的全面理解，利用构建的环境地图和相应的寻优算法，为机器人规划从起点到终点的最优或次优路径。局部路径规划在环境信息未知或部分已知的情况下，依赖传感器实时采集的环境数据，侧重于机器人的实时感知与动态避障能力。这种方法要求机器人系统具备高速的信息处理和计算能力，能够对环境的建模与搜索进行动态更新和校正。下面详细介绍这两类算法。

1.2.1 全局路径规划算法

在研究移动机器人的自主导航技术时，全局路径规划是不可或缺的关键部分，全局路径规划[8]充分利用完整的环境信息，适用于那些任务目标已经明确界定的应用场景。在全局路径规划的算法范畴内，我们可以将其进一步细分为两大类，传统算法与智能仿生算法。传统算法中的图搜索路径规划方法，在处理结构化地图内的最短路径查找问题时展现出了显著的优势，这种方法能够有效地在有序的地图数据中找到从起点到终点的最优路线。而基于采样的路径规划方法通过在配置空间内随机生成采样点，构建增量式搜索结构，逐步探索环境拓扑关系以逼近最优路径。下面将详细阐述这些算法中的典型方法。

1）基于图搜索的路径规划算法

基于图搜索的算法将环境建模为图结构，图中节点表示位置，边表示节点之间的连接关系。通过在图中搜索节点和边，找到从起点到目标点的可行路径。以下是几种常见的图搜索算法。

Dijkstra[9]算法是一种求解最短路径的方法，它使用了广度优先搜索的思想，广泛应用于有向图的最短路径计算。算法从初始点出发，通过迭代选择距离初始

点最近的点，逐步扩展至已确定最短路径的顶点集合。随后以新加入的顶点为中心，向目标点方向逐层延伸，并动态更新其余顶点到起点的最短距离及路径信息。当目标点被纳入已确定集合时，算法终止并输出最终结果。由于需要对邻接节点逐一进行遍历操作，其运行效率在大规模图中较低，同时算法无法处理包含负权边的图结构，在负权值的情况下，算法可能无法正确收敛或得出准确结果。传统 Dijkstra 算法的基本流程如图 1- 1 所示，其中集合 S 表示已完成计算的节点，v 表示尚未加入集合节点。

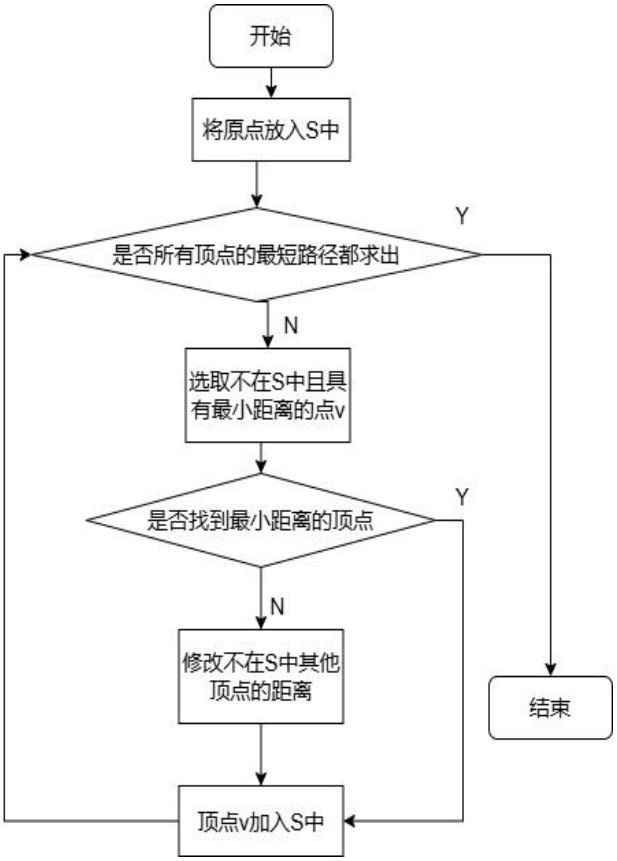


图 1- 1 Dijkstra 算法基本流程图

在电力系统规划中，Guojun[10]等人提出基于自适应分辨率网格的改进 Dijkstra 算法，通过动态调整栅格精度实现输电线路路径优化，提升了算法效率，并成功应用于复杂地形下的路径规划，这一改进在保留 Dijkstra 算法全局最优特性的同时，通过分层搜索策略降低了计算复杂度。Wenbin[11]团队将改进 Dijkstra 与粒子群优化算法融合，提出适用于数字微流体生物芯片的混合路径修复方法，优化路径长度，同时解决了传统 Dijkstra 在高维空间中的局部最优问题。当前研究趋势表明，Dijkstra 算法的改进正朝着多模态融合与场景自适应方向发展。

A*算法由 Hart 等人提出[12]，在 Dijkstra 算法的基础上融入了启发式搜索策

略，通过计算并比较每个节点的总代价，优先探索那些预期总代价最低的节点，从而大幅度提高了搜索过程的效率。在接下来的数十年间，A*的应用范围日益广泛，并催生出众多优化的变体，这些改进进一步增强了 A*算法的适用性和性能，使其在解决复杂搜索问题时展现出更高的效能。Zhang[13]等人提出自适应权重 A*算法，通过动态调整启发函数中的安全系数权重，提升了巡检机器人在动态障碍物环境中路径优化率。Qi 团队[14]针对消防机器人任务特性，引入环境动态响应机制，将火场温度与障碍物运动参数融入代价函数，显著提升算法在紧急救援场景中的鲁棒性。Wang[15]等将改进 A*与模型预测控制结合，构建分层规划框架这种混合架构解决了传统 A*在高维空间中的局部最优问题。A*算法基本流程图如图 1- 2 示。

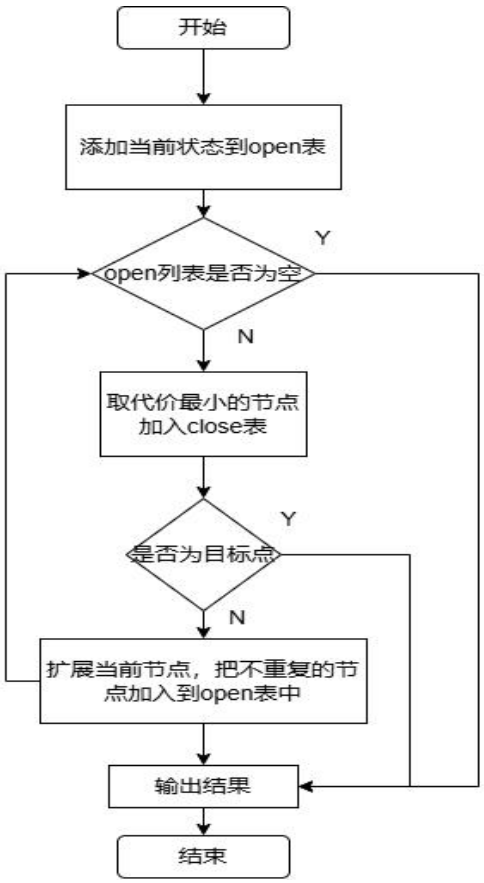


图 1- 2 A*算法基本流程图

2）基于采样的路径规划算法

基于采样的路径规划算法[16]作为解决高维运动规划问题的有效范式，其核心在于通过随机采样机制在构型空间中进行可行路径探索。区别于基于图结构的确定性搜索策略，该方法采用概率性节点生成与增量式图拓扑相结合，通过在状态空间随机布点并建立局部连接关系，规避了对环境几何特征的完整建模需求。特

别是在包含动态障碍物或非结构化特征的复杂场景中，该算法展现出卓越的计算效率优势。以下是两种常见的采样路径规划算法。

作为概率采样规划领域的开创性方法，概率路线图法(PRM)[17]首次系统化构建了随机采样与图搜索结合的求解框架。其核心机理在于通过状态空间的随机采样与局部连接验证，构建可行路径的拓扑结构网络。该方法分为两个阶段，预处理阶段和查询阶段。在预处理阶段，首先在状态空间内大量生成样本点，经过碰撞检测后，再将这些符合条件的点与其邻近的点相连，构造出一张完整的路线图；在查询阶段，则利用图搜索算法在这张图中寻找从起点到目标点的最短路径。图 1- 3 展示了 PRM 算法的整体流程。

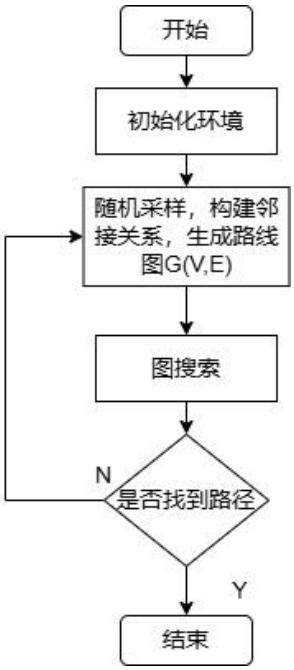


图 1- 3 PRM 算法流程图

为了解决高维空间中路径规划问题，Lavalle[18]提出了 RRT 算法，从起点开始将其作为树的根节点，并不断生成随机点以扩展这棵树，当某个新节点与目标点重合或位于目标点的阈值范围内时，路径规划完成。这种方法的优点在于无需提前构建环境的空间模型，且算法结构简单、易于实现。然而也存在一些不足之处，生成的路径通常包含大量冗余点，路径平滑性较差，导致整体质量不够理想，同时由于算法依赖随机采样，不同运行结果可能存在较大差异，稳定性较弱。诸多学者又对 RRT 算法进行了改进，其中 Karaman 和 Frazzoli 提出了 RRT*[19]算法，RRT*算法是 RRT 算法一个最重要的改进版本，主要是进行了重新选择父节点和重布线的操作，优化了冗余节点，生成的路径是渐进最优的，但存在收敛速度慢，

算法时间开销较大等问题。RRT2connect 算法[20]在 RRT 的基础上引入了双树扩展环节，分别以起始点和目标点为初始节点，将随机树分别扩展至两个随机树相连，从而实现对空间的快速搜索，但存在窄道时其性能会显著下降。WangWei 等人[21]尝试基于学习的 RRT 算法解决机器人在狭窄通道中的机器人路径规划，提高了其在复杂障碍物环境中避障和逃离障碍物区域的能力，但生成的路径并不是理想的最优路径。RRT 算法流程图如图 1- 4 所示。

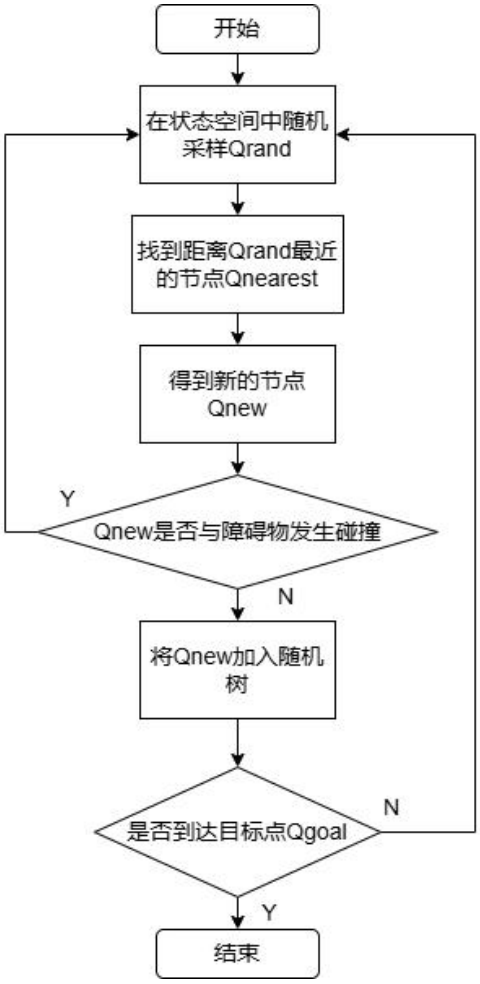


图 1- 4 RRT 算法流程图

1.2.2 局部路径规划算法

在机器人自主导航领域，局部路径规划通过传感器实时感知环境信息，实时构建避障轨迹并保持向目标区域移动。区别于全局路径规划，局部路径规划的核心是在环境变化时能快速响应并进行修正，适用于存在不可预测障碍场景中。以下是几种局部路径规划算法。

蚁群算法[22]受益于蚂蚁在觅食时释放一种叫信息素的物质，告诉其他蚂蚁这是一条寻找食物的有效路径，其他蚂蚁也能通过这种物质找到这个区域，并且在

附近释放信息素，不断释放的信息素会吸引更多的蚂蚁来到这条路径上，最后沿着信息素寻找一条最优路径。蚁群算法因其快速收敛特性，能够在局部区域内高效搜索，因此在 TSP、机器调度等优化问题上得到了广泛应用。然而该算法涉及多个参数，且参数设置较为复杂，容易使算法过早收敛于局部最优解。为提高全局搜索能力和整体效率，必须针对具体问题进行细致的参数调优，从而提升算法的鲁棒性和搜索效果。Wang T C[23]等提出蒙特卡洛改进型蚁群算法，该方法通过引入蒙特卡洛随机采样机制，在路径探索阶段动态调整信息素分布，用概率采样减少无效路径搜索。Suresh K M[24]提出了一种增强型蚁群优化算法，通过引入自适应信息素挥发系数，根据节点剩余能量与路径拥堵程度动态调节信息素沉积规则，采用加权求和法构建复合奖励函数，降低了能耗。Teng R[25]等人出了一种改进的蚁群优化算法，解决送货车辆路径问题的多目标挑战。Zhang D[26]等人在核事故环境下的移动机器人多目标路径规划研究中，提出了一种融合改进蚁群优化与修正 A*算法的混合架构，使算法能够同时兼顾路径长度、障碍规避以及核辐射风险等多个指标，同时对蚁群算法的信息素更新规则和搜索策略进行了优化，提高了收敛速度。蚁群算法流程图如图 1- 5 所示。

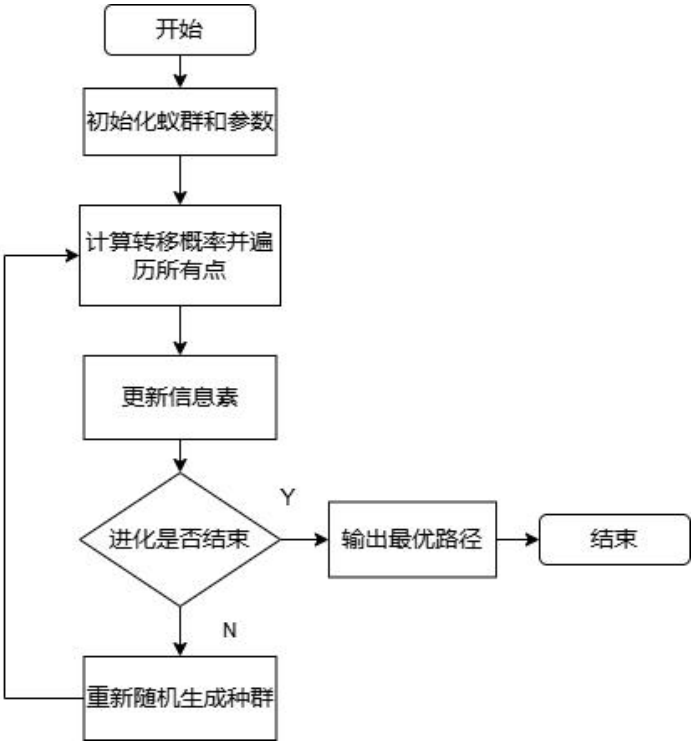


图 1- 5 蚁群算法流程图

遗传算法[27–30]于 1975 年提出，遗传算法模拟生物体 DNA 交换和复制的过程，首先随机初始种群，对这些种群进行选择、交叉和变异。遗传算法可以克服

传统搜索和优化算法遇到的一些障碍，但遗传算法的缺点在于计算量大、需要调节很多参数，执行时间长。Zhu J 等人[31]提出了一种基于网格地图的改进遗传算法，优化了机器人路径规划中的全局搜索能力与收敛效率，改进后的算法在收敛速度、路径优质性以及鲁棒性方面均表现出显著提升，有效解决了传统遗传算法易陷入局部最优和收敛缓慢的问题。Wang Q 与 Yi W[32]提出了一种融合水母搜索算法、粒子群优化与遗传算法的混合路径规划框架，旨在解决复杂城市地形中无人机路径规划面临的多约束优化挑战。Han J[33]等人提出了一种融合马尔可夫链的改进遗传算法，该文针对农业机器人在完全覆盖路径规划中的问题，提出了一种基于马尔可夫链改进的遗传算法方法，通过引入马尔可夫链模型优化搜索策略，使得遗传算法在搜索空间中的探索能力更强，同时马尔可夫链的应用使得算法能够更好地逃避局部最优解，从而提升了全局最优解的搜索能力。Wu G[34]等人提出了一种改进遗传算法的全覆盖路径规划方法，重新设计了编码方式和适应度函数，综合考虑路径覆盖率、行驶距离及转弯成本等多个因素，引入自适应变异机制，增强全局搜索能力并减少早期收敛的风险，显著提高了计算速度和路径平滑度，为无人水面艇在复杂环境下的任务执行提供了一种更高效、实用的解决方案。遗传算法的流程图如图 1- 6 所示。

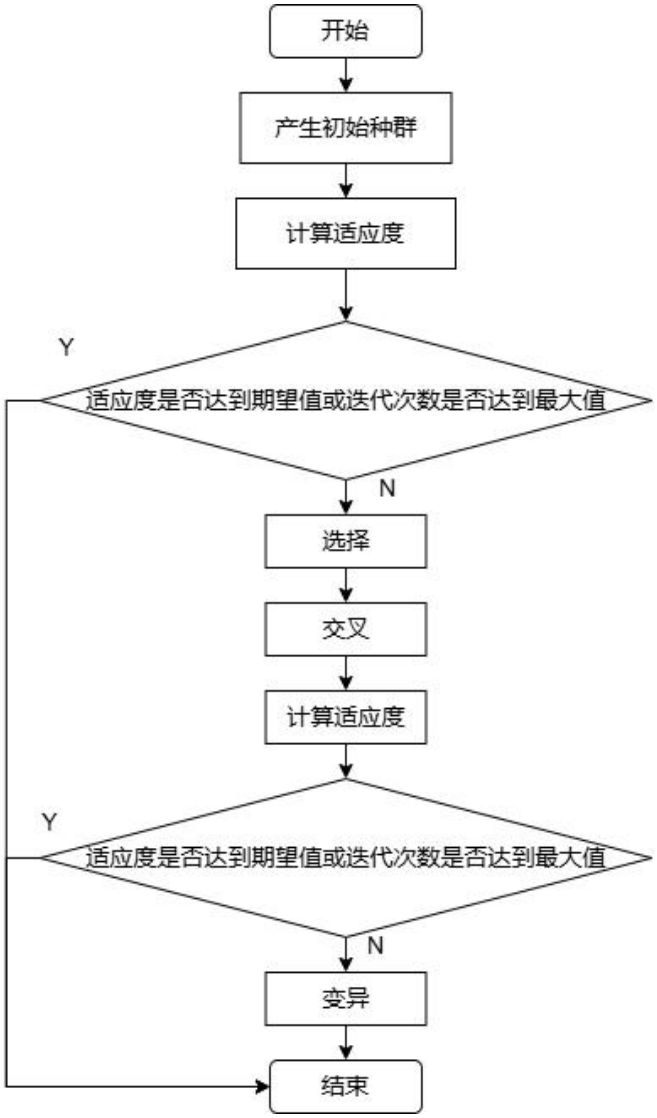


图 1- 6 遗传算法流程图

人工势场法[35–37]采用虚拟势能场建模实现机器人导航。该方法在目标点构造引力场驱动机器人趋近，同时在障碍物周边建立斥力场规避碰撞，通过势场叠加形成的合力矢量场引导机器人运动。图 1- 7 为该算法的流程。于振中[38]等人重新设计了目标引力和障碍物斥力函数，通过优化函数形态和参数调节，降低了陷入局部极小值的风险。江苏大学研究团队[39]提出了一种面向智能车辆的主动碰撞规避算法，引入动态调整机制，根据实时环境反馈动态调整人工势场函数，使算法能够适应复杂和多变的场景，增强了智能车辆在动态环境中的碰撞避免能力。Li D[40]等人提出了一种融合格子玻尔兹曼方法与改进人工势场法的水下蛇形机器人避障控制算法，这种创新方法为水下机器人在复杂流场环境中的自主导航提供了新的技术思路。Yao Q[41]等人设计了一种结合黑洞势场理论与深度强化学习，旨在克服传统人工势场法在处理局部极小值问题和适应动态环境方面的不足。

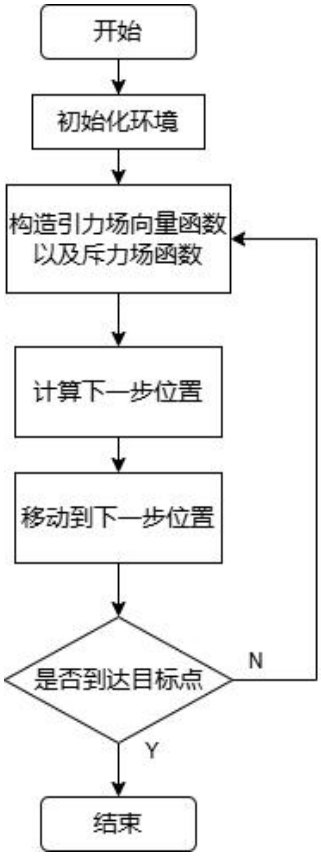


图 1- 7 人工势场法算法流程图

1.2.3 深度强化学习算法

1992 年，Watkins 正式提出 Q-Learning [42]算法，这一里程碑式推动了强化学习从理论模型向实用化算法的转型。Q-Learning 通过构建 Q 函数实现策略优化，使智能体在交互过程中逐步学习状态-动作对的累积奖励。Bo L 等人[43]引入先验知识库优化多机器人间的路径协调，提高了多机器人协同效率。毛国君[44]等提出了根据环境反馈动态调整贪婪因子值，有效降低迭代搜索成本。

DQN[45]由 DeepMind 于 2015 年提出，首次将深度学习与 Q-Learning 结合，解决了传统 Q 学习在复杂环境中的局限性，避免了维度灾难，提升了复杂环境下的适应性。DQN 通过目标网络和经验回放机制，结合深度学习的学习能力，突破了传统强化学习的瓶颈，成为深度强化学习领域的里程碑算法。但在在复杂动态环境中，DQN 算法暴露出收敛效率较低、动作价值评估结果波动较大的问题。董永峰[46]等人提出了一种改进的 DQN 算法，并将其应用于机器人路径规划问题，通过调整奖励函数、完善网络更新机制和改进探索策略，有效缓解了 Q 值过估计和收敛缓慢的问题。Lv L[47]等人提出分阶段经验策略与双重网络结构优化，提升了 DQN 在动态路径规划中的实时性与鲁棒性，为复杂环境下移动机器

人导航提供了有效方法。针对动态障碍物避障与船舶路径规划任务，Yang X 和 Han Q[48]提出了一种新方案，该方案结合了优先经验回放机制与动态障碍物预测模块，有效提升了 DQN 算法在船舶路径规划任务中的实时响应和安全性能，从而为复杂海洋环境下的智能航行提供了新的技术路径。

DDPG[49]算法是在 DQN 框架上进行改进而来，它融合了确定性策略梯度（DPG）和演员-评论家（AC）结构的优势，通过深度神经网络解决连续动作空间问题，并将传统 DQN 在离散控制中采用的经验回放和目标网络机制扩展到了连续控制领域，从而实现了稳定的策略更新和高效训练。张斌[50]等人提出的改进 DDPG 算法在自动驾驶中的研究，通过动态经验回放与多模态数据融合优化 DDPG 算法，提升了自动驾驶在车道保持与动态避障任务中的实时性与安全性。周盛世[51]等人通过改进 DDPG 算法架构与训练策略，提高了其在动态路径规划任务中的计算效率与抗干扰能力。Liu[52]等人提出的 D*-KDDPG 算法，融合运动学分析与 D*算法的动态重规划能力，改善传统 DDPG 在复杂环境中收敛慢、路径质量差的问题。为解决传统 DDPG 在路径规划中遇到的收敛缓慢和策略稳定性不足的问题，Yi Q D[53]等人提出双层奖励函数，引入动态权重调整策略结合 DP 算法对 DDPG 生成的路径进行平滑优化，消除冗余航点，降低路径复杂度并提升航行安全性。

1.3 路径规划概述

1.3.1 路径规划问题

路径规划是移动机器人研究中的核心问题之一，其核心任务是找到一条从起点到终点的可行路径，且满足任务的需求。其中包括如何在静态或动态环境中规避障碍物，同时综合考虑机器人的运动特性、能耗限制及任务优先级等因素。解决路径规划问题的关键在于设计高效算法以生成最优或次优路径，从而提升机器人的运行效率、安全性和适应能力。移动机器人通过构建环境模型、利用算法进行路径搜索，并对生成的路径进行平滑优化，从而完成整个路径规划过程。

路径规划是一种应用范围涵盖多个领域、在实际场景中起着至关重要作用，是一种适应性广、通用性很强的技术。同时随着人工智能技术的快速发展，智能算法在路径规划中的应用日益广泛，为解决复杂环境下的路径搜索提供了新的思路和方法。

1.3.2 移动机器人路径规划特点

移动机器人路径规划具有以下几个特点：

（1）环境复杂性：移动机器人通常面临的环境是复杂多样的，其中包括静态和动态障碍物，移动机器人需要根据环境的变化实时规划可行路线。（2）多目标平衡：路径规划目标具有多样性，有路径长度、时间效率、能源消耗和平滑程度等指标，这就要求算法在多个维度间寻找平衡点。（3）计算时效性：在动态环境中，路径规划算法需要具备快速感知、迅速决策与即时调整的能力，以确保其时效性，从而有效避免碰撞，保障行进安全。（4）多维约束整合：算法设计需要融合机器人本体运动特性、机械结构限制以及具体任务需求等多方面条件。（5）多机器人协作：在多机器人系统中[54]，路径规划必须充分考虑各机器人之间的协同作用，以确保它们的行进路线互不冲突。

本文聚焦于以下三方面性能突破，路径最优性：以最小化路径长度为目标，通过改进 RRT 算法生成全局参考轨迹，结合贪心算法剔除冗余节点，实现几何意义上的最短路径规划；任务时效性：针对动态障碍物的不确定性，提出基于双重价值评估的 DQN 改进框架，通过自适应学习率机制与动态奖励函数设计，在保证避障安全的前提下缩短决策延迟；计算实时性：构建分层规划架构，利用全局路径点序列作为局部规划的子目标引导，将长距离规划分解为多阶段短时序决策问题，显著降低动态环境下的策略搜索复杂度。

1.4 本文主要研究内容

本研究提出分层运动规划架构，针对移动机器人导航任务建立全局与局部协同优化机制。在全局规划方面，针对传统 RRT 算法在随机采样过程中容易出现过度随机和路径冗余的问题，设计了一种结合势场引导与动态步长调节的改进架构，从而使规划路径更加高效和准确。在局部规划层面，为了克服深度强化学习中 Q 值过高估计和策略选择精度不足的问题，提出了融合双重价值评估和自适应探索机制的改进方法，以提高决策的精度和系统的稳定性。主要内容包括：

全局路径规划方面，针对传统 RRT 算法的缺陷，结合人工势场法和步长调节机制进行改进。提出融合势场导向机制的采样框架，通过构建混合势场模型并采用动态权重分配策略，使随机树在规避障碍的同时向目标区域推进。同时引入动态步长调节方法，根据扩展节点所处位置调整步长，以提高算法收敛效率和避障精度。生成路径利用贪心算法进行优化处理，剔除冗余节点并优化生成树点连

接顺序，提升路径平滑性。

局部路径规划方面，提出基于双重价值评估机制的深度强化学习改进框架，构建动作选择网络与目标价值网络分离的协同架构，以抑制价值估计偏差。同时设计具备自适应功能的ε-greedy 策略，提升策略选择的精度，在训练初期增强环境探索能力，后期逐步收敛至最优策略。重构奖励函数并引入奖励平滑机制与惩罚项，确保算法在路径决策时能准确量化动作的长期价值，增强系统在动态场景下的抗干扰能力。仿真实验对比分析显示，相较于基准 DQN 模型，改进算法显著缓解了 Q 值高估现象，并增强了策略收敛的稳定性。

1.5 本文的组织结构

第一章：阐述了路径规划的研究背景和意义，综述了全局路径规划、局部路径规划以及深度强化学习算法在路径规划领域的研究现状，并概述了本文的主要研究内容和组织结构，为后续研究提供了明确的方向。

第二章：阐述了深度学习的基本架构与原理，同时深入探讨强化学习的核心概念，并对强化学习算法分类进行概述。这些理论为后续研究深度强化学习路径规划算法奠定基础。

第三章：本章主要探讨了全局路径规划中对 RRT 算法的改进方法。首先介绍了传统 RRT 算法的基本原理，指出其在路径规划中存在随机性强、路径冗余以及扩展缺乏导向性等问题。针对这些不足，提出了一种自适应目标偏向的 RRT 算法。引入人工势场的概念，构建了目标吸引力、障碍物排斥力和随机扰动协同作用的混合势场。同时结合动态步长调整和自适应权重策略，引导随机树稳步向目标区域扩展，有效缩短了规划时间和路径长度。最后采用贪心算法对路径进行优化。

第四章：探讨了深度 Q 网络在路径规划中的改进方法。首先介绍了 Q-Learning 的基本原理及其 Q 值更新公式；随后详细阐述了 DQN 模型在神经网络拟合 Q 值、经验回放机制和目标函数设计等关键技术。针对传统 DQN 在路径规划中存在的问题，提出了三项改进措施：目标网络优化、奖励函数的调整以及探索策略的设计。

第五章，本章围绕移动机器人路径规划算法的实验验证展开系统性研究，主要从硬件平台构建、算法性能评估及结果分析三个维度进行深入探讨。首先搭建基于自研实体机器人的仿真实验环境，构建包含静态障碍物与动态干扰的

复合测试场景；其次针对本文提出的引力场引导动态步长 RRT 算法与改进型深度 Q 网络（DQN）轨迹规划方法，分别设计实验验证其可行性；最后通过定性结果展示，论证所提算法在复杂环境中的有效性。

第六章：对本研究的主要工作进行了总结，并展望了未来路径规划领域的研究方向。

第 2 章 深度强化学习算法相关理论

深度学习作为人工智能领域的核心研究方向，其技术本质在于通过多层非线性变换从大规模数据集中自动学习特征表示。典型架构如卷积神经网络（CNN）通过层级化卷积核扫描输入数据，逐步提取从局部边缘到全局语义的多级特征，这种端到端的学习方式降低了对人工特征工程的需求，显著提升了模型在图像识别、自然语言处理等任务中的判别能力。

强化学习则是一种基于交互反馈的决策优化方法，其核心在于智能体通过试错机制，在动态环境中依据奖励信号迭代更新策略。与深度学习不同，强化学习更关注时序决策过程，通过 Q 值迭代实现长期收益最大化。

深度强化学习融合两者的优势，利用深度网络处理高维状态输入，同时结合强化学习的决策机制。这种融合架构在自动驾驶与游戏如 AlphaGo 的决策模型等场景中展现出显著优势，但也面临训练数据需求大、样本效率低等挑战。

2.1 深度学习算法

深度学习是人工智能的关键技术，其模型架构受人脑神经结构启发。深度学习（DL）这一理念最早由 Hinton[55]等人提出，通过模拟人脑神经细胞间的连接方式构建模型框架，以实现智能化。人工神经网络的输入端对应人脑细胞的轴突结构，输出端对应树突结果，而人脑细胞的抑制状态与激活状态则通过激活函数来表达。

深度学习不仅能够完成图像目标检测、语音识别等任务，其核心优势在于利用多层网络和非线性变换，从高维数据中自动提取低维特征。基于这些基础模型，研究者们进一步提出了多种改进方案，使得这些网络在目标检测、语音转写等领域得到了广泛应用，并取得了显著成果。这些突破不仅推动了深度学习算法的不断优化，同时也为人工智能技术在现实生活中的广泛应用打下了坚实基础。深度学习是一种静态学习方式，通过迭代训练已有数据集来学习网络参数，然后对未知数据进行分类或识别。

2.1.1 深度神经网络模型

人工神经网络[56]（ANNs）是一种模仿动物神经系统功能特性的数学模型，主要用于实现分布式和并行的信息处理。通过对内部大量节点之间连接权重调节，

具备自我学习能力和适应能力。深度神经网络[57]（DNNs）是一种由多层非线性变换构成的机器学习模型，通过自动提取并组合数据中的低级到高级特征，实现对复杂模式的识别与建模。通过多个隐藏层，每一层级都专注于从输入数据中提炼出更高级别的特征信息，从而实现对复杂数据结构的精准学习。图 2- 1 为神经元结构示意图。

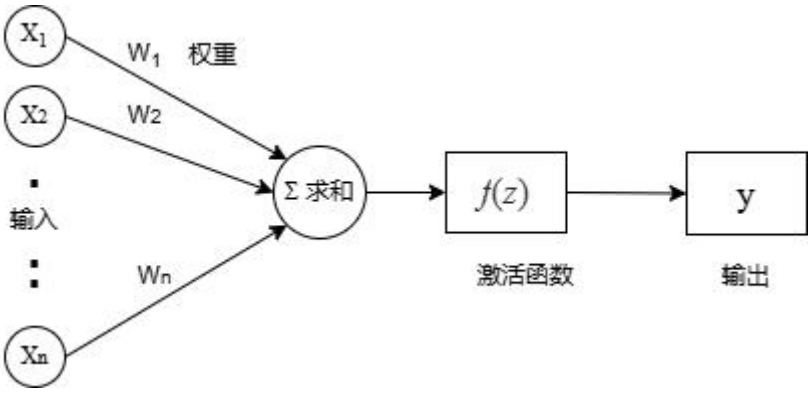


图 2- 1 神经元结构

在神经网络中，神经元通常被视作基本的计算单元，其功能是接收多个输入信号、对这些信号施加相应的权重和偏置，并经过激活函数处理后产生输出。通常使用 $\mathbf{x}=[x_1,x_2,...,x_n]$ 来表示来自神经元的 n 个输入信号，每个输入 x_i 都与一个权重 w_i 相关联，这些权重用向量 $\mathbf{w}=[w_1,w_2,...,w_n]$ 来描述。神经网络的训练过程正是通过不断调整这些权重，使得预测结果达到最佳效果。神经元首先计算其输入的线性组合 $\mathbf{w}\cdot\mathbf{x}$，再加上偏置 b，随后将这一结果传递给激活函数 f，得到最终的输出 y，其数学表达式为：$y=f(z)=f(\mathbf{w}\cdot\mathbf{x}+b)$，这个输出会继续传递到下一层神经元，如果结果位于输出层，则作为整个神经网络的预测结果。神经网络的输出依赖于三个主要因素权重、偏置和激活函数。权重反映了输入信号与神经元之间的关系，这些参数经过随机初始化，并在训练过程中根据数据不断更新；偏置则起到修正加权和的作用；激活函数在神经网络中引入非线性因素，确保多层结构能够捕获复杂的非线性模式。以下是一些常见的激活函数：

（1）Sigmoid 函数如式（2-1）所示，函数将任意实数输入 x 映射到（0,1）区间，随着 x 趋向负无穷时输出趋近于 0，而 x 趋向正无穷时输出趋近于 1。其优势在于连续可导、严格单调递增，便于在神经网络中进行梯度计算与优化，适合反向传播算法的实现。但当输入值较大或较小时，其导数趋近于 0，导致梯度在反向传播过程中呈现指数级衰减，容易引发梯度消失问题，降低模型训练效率，

其次其输出范围为（0,1）的非零中心特性，使得后续网络层的输入可能出现偏移，进一步加剧梯度更新的不平衡性。函数图像如图 2- 2 所示。

$$f(x)=\frac{1}{1+e^{-x}} \tag{2-1}$$

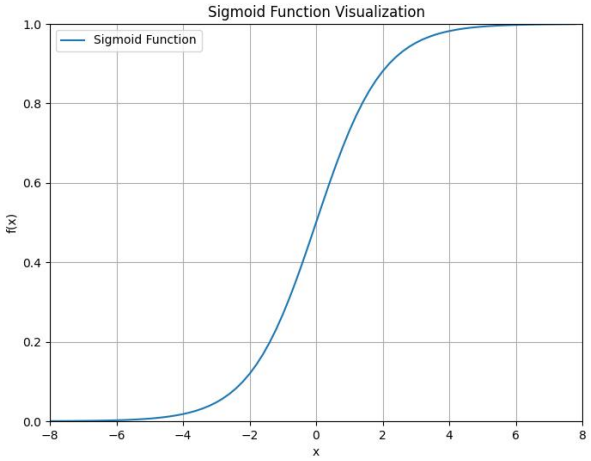


图 2- 2 Sigmoid 函数曲线

（2）Tanh 函数如式（2-2）所示，其输出范围在-1 到 1，具有平滑且连续的非线性特性，函数以原点（0，0）中心对称，其函数图像如图 2- 3 所示，相较于 Sigmoid 函数的 0 到 1 输出范围，Tanh 函数的对称输出减少了后续层输入的偏移问题。但输出在输入值绝对值较大的情况下，趋近于 1，导致导数趋近于 0，从而造成深度网络训练受阻的梯度消失问题。

$$f(x)=\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2-2}$$

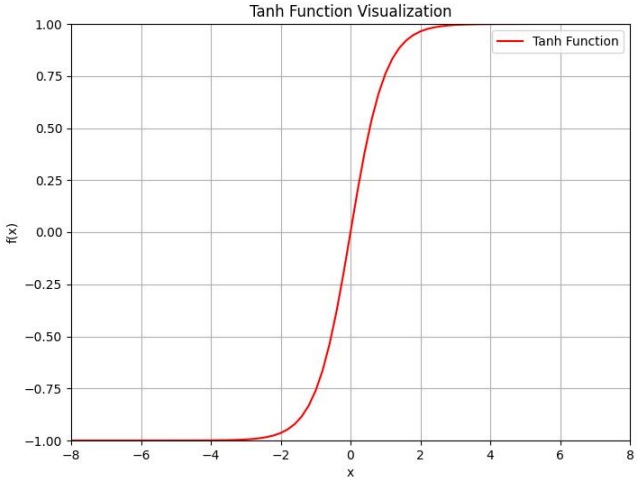


图 2- 3 Tanh 函数曲线

（3）ReLU 函数如式（2-3）所示。与 Sigmoid 和 Tanh 不同，ReLU 的计算仅需比较输入值与 0 的大小并取最大值，避免了指数运算等复杂操作，降低了计算开销，适用于大规模神经网络的训练。在正输入区域，ReLU 的输出与输入呈线性关系且梯度恒为 1，有效缓解了传统激活函数在梯度消失的缺陷，使得深层网络的反向传播能够更稳定地传递误差信号。在负输入区域的输出为 0，其梯度为 0，导致相关神经元在反向传播中无法更新权重，限制模型对负值输入的适应能力，函数图像如图 2- 4 所示。

$$f(x)=\max(0,x) \tag{2-3}$$

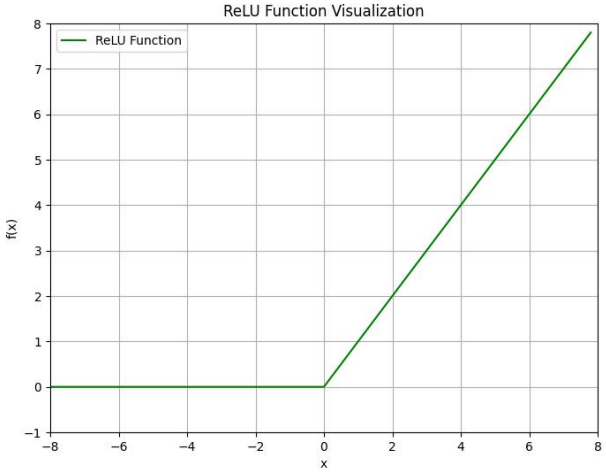


图 2- 4 ReLU 函数曲线

神经网络的标准架构有三个组成部分：输入层，隐藏层，输出层。当隐藏层仅存在单层结构时，整个架构通常被定义为双层网络。输入层神经元数量与样本特征维度直接相关，而输出层的节点数目则由待解决任务的性质决定。在典型的二值分类场景中，采用 Sigmoid 函数作为决策单元，输出层仅需配置单个神经元即可完成判别；对于涉及三个及以上类别的多分类问题，输出层的维度必然与目标类别总数保持严格对应关系。隐藏层的层级深度与单元数量需要根据具体场景预先配置，所以人工调参机制是保证模型适应性的重要手段。图 2- 5 为两层神经网络结构图。

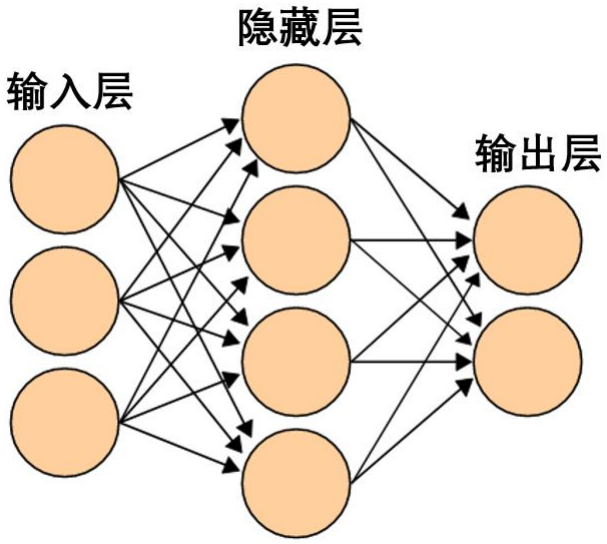


图 2- 5 两层神经网络

2.1.2 卷积神经网络

卷积神经网络（CNN）是一类包含卷积计算且具有深度结构的前馈神经网络。与传统方法需人工设计特征不同，CNN[58–61]卷积运算利用局部感知和参数共享机制，在提取空间特征的同时优化参数效率。随着网络深度增加，特征逐步从低级向高级语义抽象。池化层进一步减少特征维度，提升模型对平移变化的鲁棒性。结合非线性激活函数 ReLU，网络能够学习复杂非线性映射，并通过全连接层将特征向量映射至最终输出。在实际应用中，CNN 凭借其独特的局部特征组合能力，已成为计算机视觉领域的主流方法，尤其在处理高维视觉数据时，CNN 的降维与抽象能力展现出显著优势。图 2- 6 展示了经典的 CNN 架构，输入图像经多层卷积与池化后形成特征金字塔，最终通过全连接层输出预测结果。这种设计既保留了空间局部相关性，又通过深度堆叠实现了复杂模式的建模能力。

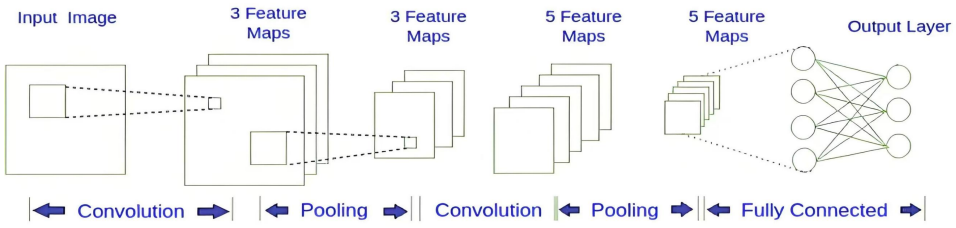


图 2- 6 简单的 CNN 架构示意图

如图 2- 6 所示，原始图像首先输入到网络的第一层，并通过卷积操作得到深度为 3 的特征图，作为第二层的输入，随后对该特征图进行池化处理，生成深度 3 的特征图。重复上述卷积和池化的过程后，得到 5 个特征图，这些特征图会被

依行展开并拼接成一个长向量作为全连接层的输入。全连接层本质上可视为 BP 神经网络，它接收展平的特征向量并进行分类或回归计算，最终输出预测结果，在这个过程中，每个特征图可以视为一个矩阵形式的神经元集合，其结构与 BP 网络中的神经元类似。卷积层一般由多种不同大小和深度的卷积核组成，不同卷积核的尺寸决定感受野范围，从而有效提取图像特征。池化层则对特征图进行降维与信息压缩，常用的方式是最大池化，输出以矩阵形式呈现。最终将这些处理后的数据展平并送入全连接层，网络输出最终的预测结果。

2.2 强化学习算法

强化学习是机器学习中的一个重要分支，目标是让计算机完成那些传统编程难以实现的任务，从而提升机器的自主能力。从广义上来说，机器学习通过大量数据的训练构建出能够解决问题的模型，并用这个模型对未来的情况进行预测。而强化学习的独特之处在于，智能体通过与环境的持续交互来获取奖励，逐步学会在不同情境下采取最佳行动，最终实现奖励的最大化。

在强化学习中，智能体与外部环境交互，通过执行动作获得环境反馈奖励，并根据所观察到的状态不断调整策略，以逐步完成任务目标。其主要构成包括四个要素，动作表示智能体可执行的操作；策略决定了智能体如何选择动作以进入下一状态；奖励是环境对智能体动作的反馈信号，合理的奖励设计能够促进模型快速收敛；状态则是智能体感知到的环境信息。智能体在执行任务时，不断与环境互动，产生新的状态和数据，进而调整其行为策略。强化学习的示意图如图 2- 7 所示。

强化学习因其动态适应能力，在移动机器人控制领域展现出了独特的优势。机器人感知系统获取环境状态信息后，通过与环境的交互获得反馈信号，从而不断优化自身的行为策略，以达成预期目标。不仅解决了在未知环境中路径规划的问题，还为实现更智能化的路径规划提供了重要支持，由于具备自主学习和在线更新的特点，移动机器人能够灵活应对各种复杂场景。

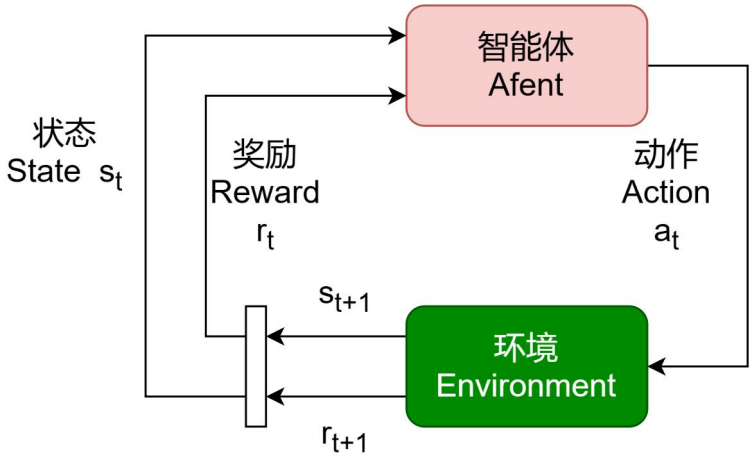


图 2- 7 强化学习示意图

2.2.1 马尔科夫决策过程

马尔可夫决策过程基于概率论框架，通过建立状态转移概率与即时奖励函数的映射关系，将动作、状态、状态转移概率和奖励形式化，以便在环境下寻找最大化预期累积回报的最优策略。若过程具备马尔可夫性质，在当前状态已知时，后续状态仅依赖于当前状态，而不再考虑以往历史状态，体现了无记忆特征。图 2- 8 展示了 MDP 的基本结构[62]，通常以五元组(S,A,R,ϒ,P)定义，其中 S 为所有可能状态的集合，A 为可执行动作的集合，R 为在状态 s 下执行动作 a 并转移至状态 s_{t+1} 时获得的即时奖励，折扣因子取值范围为 0 到 1，衡量当前奖励与未来奖励，当ϒ越接近 1，智能体越关注长期回报，反之则更重视即时收益；$P(s_{t+1}|s,a)$表示在状态 s 下执行动作 a 后转移到下一状态 s_{t+1} 的概率。假设初始状态为 s_t，当执行动作 a_t 后，环境依据 $P(s_{t+1}|s_t,a_t)$更新状态为 s_{t+1}，同时给予奖励 r_t，随后智能体基于策略选择下一动作，状态则进一步转移至 s_{t+2}，如此循环构成完整的决策序列。奖励函数的目标在于通过所选策略最大化累计奖励，累计奖励 G_t 由时刻 t 开始、经折扣后所有未来奖励总和，具体定义如式（2-4）所示。

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{n-1} R_{t+n} \tag{2-4}$$

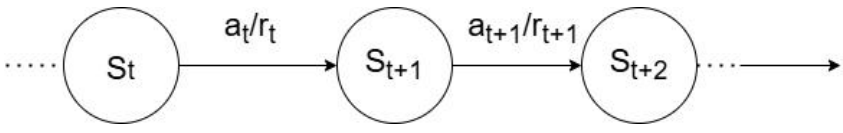


图 2- 8 马尔可夫决策过程

2.2.2 贝尔曼方程

贝尔曼方程描述了当前状态的价值与其后续状态价值之间的联系。在马尔科

夫决策过程中，一个状态的价值被定义为在该状态下所获得的即时奖励加上未来状态预期累积奖励的折扣。对于给定的策略π，贝尔曼期望方程明确了状态值函数 $V_p(s)$和动作值函数 $Q_p(s,a)$的定义。状态值函数 $V_p(s)$代表了从状态 s 出发，在遵循策略π的前提下，智能体所能预期获得的未来折扣奖励总和；动作值函数 $Q_p(s,a)$表示在状态 s 下采取特定动作 a 后，再继续按照策略π时预期获得的未来折扣奖励。状态值函数如式(2-5)：

$$V^{\pi}(s)=\mathbb{E}_{\pi}[R_{t+1}+\gamma V^{\pi}(S_{t+1})\mid S_t=s] \tag{2-5}$$

动作值函数表示为如式(2-6)：

$$Q^{\pi}(s,a)=\mathbb{E}_{\pi}[R_{t+1}+\gamma Q^{\pi}(S_{t+1},A_{t+1})\mid S_t=s,A_t=a] \tag{2-6}$$

$\mathbb{E}_{\pi}[\cdot]$表示在策略π下的期望值，R_{t+1}是立即奖励，$\boldsymbol{\gamma}$是折扣因子。当目标转向寻找最优策略时，定义的最优状态值函数 $V^*(s)$和最优动作值函数 $Q^*(s,a)$为所有策略中能达到最高奖励的函数，它们满足以下条件：

最优状态值函数如式(2-7)：

$$V^{*}(s)=max\mathbb{E}[R_{t+1}+\gamma V^{*}(S_{t+1})\mid S_t=s,A_t=a] \tag{2-7}$$

最优动作值函数如式(2-8)：

$$Q^{*}(s,a)=\mathbb{E}\left[R_{t+1}+\gamma maxQ^{*}(S_{t+1},a')\mid S_t=s,A_t=a\right] \tag{2-8}$$

公式(2-9)为贝尔曼最优性方程，最优策略可由解此方程而得。

$$Q^{*}(s,a)=max_a\mathbb{E}\left[R_{t+1}+\gamma maxQ^{*}(S_{t+1},a')\mid S_t=s,A_t=a\right] \tag{2-9}$$

在强化学习中，计算状态值函数的主要目的在于确定最优的状态-行为策略，每个状态值函数对应一种策略。最优策略$\pi^*(a|s)$为某一动作 a 能使得状态-动作值函数 $Q^*(s,a)$达到最大值，则$\pi^*(a|s)=1$，否则为 0。通过这种方式，贝尔曼方程为我们提供了从初始状态出发实现全局最优奖励的重要理论基础。

2.2.3 强化学习基本函数

奖赏函数、值函数和动作选择策略是强化学习的三要素。奖赏函数为智能体在环境中持续学习提供明确方向，值函数用于在某个状态或采取某个动作时，获得的期望总回报，动作选择策略为智能体提供最优决策。只有这三者相互协作，才能最终实现最佳决策。

（1）奖赏函数

奖赏函数是决策系统中的关键部分，它通过即时量化的方式评估智能体在特

定场景下动作价值。为智能体提供了明确的指导，帮助智能体理解哪些行为会带来积极的回报，哪些可能导致负面结果，从而逐步学会最优策略。奖赏函数的设计应该是客观且清晰的，能够准确、及时地反映系统的学习进展。奖赏函数可以分为两种形式：连续型和离散型。

1）连续型奖励机制在强化学习系统中具有显著的建模优势，其特征体现为状态-行为关联的精细化反馈能力。如式（2-10）所示，通过建立状态空间到实数域的连续映射关系，为智能体决策过程提供梯度信息。相较于离散型奖励机制，有以下优势，在策略优化层面，基于连续可微的奖励曲面能够精确量化状态价值分布，为动作选择提供可微分的优化路径；在环境交互维度，通过建立状态参数与环境动态的系统耦合性描述，使智能体行为与结果间的关联性得到显式表征，从而加速策略收敛效率；在动态适应性方面，具备实时重构奖励拓扑的能力，可根据环境参数扰动进行反馈信号的自适应调整，有效增强智能体应对非稳态环境的能力。

$$R_t = f(s_t, i_t) \tag{2-10}$$

2）离散型激励函数在强化学习系统设计中具有构造简易性特征，为状态-奖励映射的阶跃式响应特性，如式 2-11 所示。通过在关键事件节点设置非连续激励信号，为策略优化提供明确的收敛方向。相较于连续型奖励机制，离散事件驱动的建模方式大幅降低了状态空间划分的维度敏感性；基于事件触发的稀疏反馈机制不需要构建复杂的环境动态模型，增强了模型在未知领域的可扩展性。

$$R_t = \begin{cases} 1, \text{在坏的状态下} \\ -1, \text{在好的状态下} \\ 0, \text{其他情况} \end{cases} \tag{2-11}$$

（2）值函数

值函数是强化学习中用来评估智能体动作优劣的重要工具，同时也为决策提供了依据。分为两类：状态值函数和动作值函数。状态值函数是在当前状态下，智能体按照某个策略行动后，从这个状态开始一直到任务结束所能获得的预期累计折扣奖励；动作值函数为在某个特定状态下执行某个动作后，再继续按照策略行动时，未来可能获得的奖励总和。这些函数的计算依赖于智能体与环境交互过程中积累的经验数据，并通过不断迭代更新来提升决策能力。通过值函数，智能体可以在新环境中做出更优的选择，同时也为评估其表现和优化策略提供了量化

的参考。其表达如式（2-12）。

$$V_{\pi}(s_t) = E_{\pi}(\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} | \ S_t = s) \tag{2-12}$$

（3）探索策略

在未知的环境中，如何选择合适的动作以获得最佳奖励是强化学习需要解决的关键问题。探索策略在面对信息不完全的情境下，智能体通过不断尝试和纠错，逐步学会最优动作。强化学习就是找到最优策略。策略会通过条件概率分布来描述状态与动作之间的映射关系，记作π：S→A，其中 S 表示状态集合，A 表示动作集合，表达式见式（2-13）。

$$\pi(a|s) = P[A_t = a | S_t] \tag{2-13}$$

强化学习框架的核心目标在于指导智能体通过策略优化实现训练周期内的总体收益最大化。训练初始阶段，值函数的收敛状态尚未完成，此时策略生成的动作序列往往偏离最佳选择。随着迭代次数的增加，智能体通过与环境的多轮交互持续修正其决策模型，使值函数逼近最优状态。当值函数收敛至稳定区间时，基于该函数构建的策略能够指导智能体在动态环境中进行精准决策，从而实现收益函数的全局最优化。常用的动作选择策略有如下几种：

1）贪心（greedy）策略

贪心策略，也被称为贪婪策略，指在某个状态 s 下，从所有可选动作中挑选出具有最高 Q 值的动作 a。如果多个动作的 Q 值相同且均为最高值，则可以随机选择其中一个。这种方法实现起来简单，计算速度也快，但容易让智能体陷入局部最优解，同时还可能忽略掉长远来看更优的收益。

2）ε-greedy 策略

ε-greedy 在确定性最优动作选择框架中引入可控概率扰动，以此实现经验利用与空间探索的动态均衡。如式（2-14）所示，策略将状态 s 下的动作选择规则 $\pi(s)$ 分解为两个部分，以概率(1-ε)执行当前 Q 值最大化的开发模式，以概率ε从动作集合 A 中随机选取的探索模式。其中ε∈[0,1]作为超参数控制探索强度。

$$\pi(s) = \begin{cases} \text{argmax}Q(s,a_t), 1-\varepsilon \\ \text{random}(A), \varepsilon \end{cases} \tag{2-14}$$

3）Softmax 策略

Softmax 策略是一种基于概率分布的决策机制，通过指数变换与归一化处理

将动作价值映射为选择概率，通过温度系数 T 调控动作空间的探索熵值，表达式为（2-15），相比于ε-greedy 策略，Softmax 策略在训练初期能够为所有动作提供几乎均等的选择机会，在训练后期则能有效缓解因过度探索导致的不稳定性问题。

$$P(a_i / s) = \frac{k^{\frac{v_i}{T}}}{\sum_{a \in A} k^{\frac{v_i}{T}}} \tag{2-15}$$

2.3 本章小结

本章深入探讨了深度学习与强化学习的基础理论，为后续研究奠定了坚实的理论基础。深度学习部分从生物神经元的启发出发，介绍了人工神经网络的基本架构，包括多层感知机和卷积神经网络的计算原理。详细分析了常用激活函数的特性及其在模型构建中的关键作用，提供了构建和优化深度学习模型的清晰思路。强化学习部分介绍了马尔科夫决策过程，描述了智能体在环境中通过状态空间、动作空间、状态转移概率和奖励函数等要素。深入分析了贝尔曼方程，揭示了状态值函数和动作值函数之间的递归关系，为求解最优策略提供了理论基础。

第 3 章基于引力导向动态步长 RRT 算法

3.1 RRT 算法介绍

3.1.1 RRT 算法的基本原理

基本 RRT 算法以起始点作为初始节点，随后在状态空间中进行随机采样，逐步扩展随机树，每次扩展的新节点需要通过碰撞检测后才能加入到生长树中，不断重复上述过程，直到随机树上的点扩展到目标点的有效范围内。此时若连接目标点的路径未发生碰撞，则将目标点添加至随机树并终止采样。最终通过对树结构进行回溯，即可获得一条从起始点到目标点的可行路径。随机树的生长过程如图 3- 1 所示。

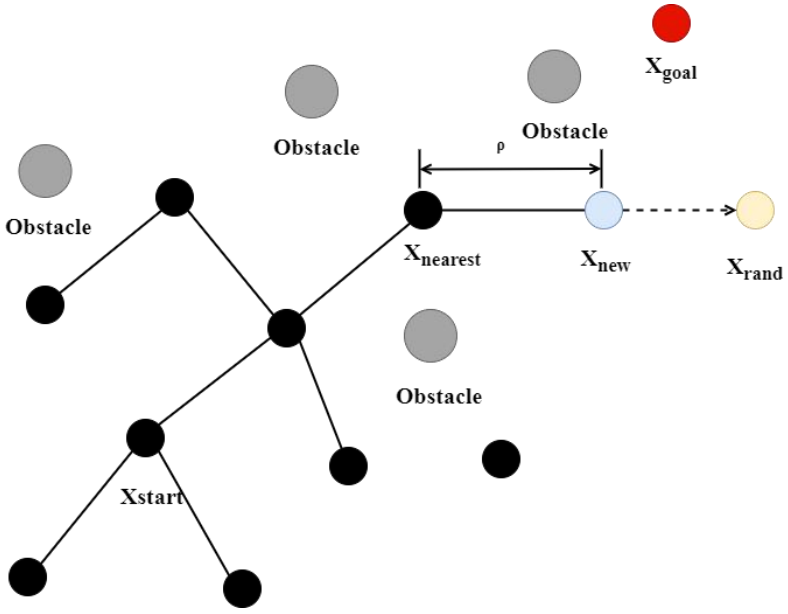


图 3- 1 RRT 算法节点扩展原理图

为了阐明 RRT 算法的原理，设机器人运动空间为 C，其中 Q 表示随机树，设 x_{start} 为随机树的起点，x_{goal} 为目标点，在状态空间中随机生成一个采样点，记为 x_{rand}，$x_{nearest}$ 表示找到随机树中与 x_{rand} 最近的节点，从 $x_{nearest}$ 沿着指向 x_{rand} 的方向向前扩展一个步长ρ生成一个新节点 x_{new}。如果 $x_{nearest}$ 与 x_{new} 之间的连线与障碍区域（Obstacle）发生碰撞，则需要重新采样生成新的 x_{rand}；否则将 x_{new} 加入到随机树 Q 中，并继续进行下一轮采样。最终当机器人到达目标区域范围内，或迭代次数超过预设上限时，搜索终止，此时从随机树目标节点沿回溯路径返回至 x_{start} 所得到的路径即为规划出的路径。RRT 算法的伪代码如表 3- 1 所示：

表 3- 1 基本 RRT 算法伪代码

RRT 算法伪代码

Q←InitializeTree()
For k←1 to N do
x_{rand}←RandomSample(C)
$x_{nearest}$←NearestNeighbor(x_{rand},Q)
x_{new}←NewState($x_{nearest}$,x_{rand},ρ)
E_{new}←NewEdge($x_{nearest}$,x_{new})
If Obstacleless($x_{nearest}$,x_{new}) than
Q.AddNode(x_{new})
Q.AddEdge(E_{new})
End if
If(Distance(x_{new},x_{goal})<=ρ)
break
End if
End for
return Q

在上述算法伪代码中，函数 RandomSample()的作用是在状态空间内随机生成一个采样点；函数 NearestNeighbor()用于在随机树中查找距离随机点 x_{rand} 最近的节点 $x_{nearest}$；接着，函数 NewState()的作用是将节点 $x_{nearest}$ 沿着随机点 x_{rand} 方向延申一定距离，从而生成新的节点 x_{new}；随后函数 NewEdge()用来连接 $x_{nearest}$ 与 x_{new} 所形成的边；此外函数 Obstacleless()用来检测每次扩展的新节点 x_{new} 与其最近节点 $x_{nearest}$ 的连线是否与障碍物发生碰撞，若无碰撞则将新节点加入到随机树中，若碰撞，则返回 For 循环重新生成随机点，否则继续执行；最后函数 Distance()用于计算新节点 x_{new} 与目标点的距离，当新节点与目标点距离小于步长时，连接最后生成的新节点与目标节点，算法结束。

3.1.2 RRT 算法流程及实现

为了方便直观理解，表 3- 2 列出了 RRT 算法需要的参数。

表 3- 2 RRT 算法中各参数代表的意义

参数	含义
$x_{nearest}$	离随机点最近的点
x_{rand}	随机点
x_{new}	新节点

续表 3- 3 RRT 算法中各参数代表的意义

参数	含义
x_{start}	起点
x_{goal}	目标点
ρ	步长

RRT 算法的构建扩展过程步骤如表 3- 4 所示。

表 3- 4 RRT 算法构建扩展步骤

步骤	操作
Step1	初始化 x_{start}、x_{goal}、Obstacle 位置
Step2	生成随机点 x_{rand}
Step3	找到距离随机点最近的树节点 $x_{nearest}$
Step4	从 $x_{nearest}$ 出发，沿着 x_{rand} 方向延伸固定步长ρ生成新节点 x_{new}
Step5	检测 $x_{nearest}$ 与 x_{new} 之间的连线是否与障碍物发生碰撞，如果碰撞则跳转到 step2，否则将 x_{new} 加入随机树
Step6	计算 x_{new} 与目标点 x_{goal} 之间的距离，如果距离小于等于步长ρ,则连接 x_{new} 与 x_{goal}，否则返回到 step2

3.1.3 RRT 算法优缺点分析

RRT 算法的优点如下：

（1）RRT 算法基于随机采样策略，不依赖于环境的先验知识，即使是在结构复杂、障碍物分布不规则的场景下，依然能高效地探索出有效路径。

（2）RRT 算法是一种概率完备的算法，随着采样点数量的增加，RRT 算法能够逐渐覆盖整个状态空间，理论上可以找到可行路径

RRT 算法的缺点如下：

（1）由于随机采样的特性，搜索过程中没有没有目标性，生成的路径往往存在较多迂回、曲折的情况，这条路径并非是最优路径，并且每一次规划的路径都不相同。

（2）RRT 算法的性能受到步长，采样参数影响较大，步长过大会导致碰撞风险增加，步长过小则降低探索效率，不同的采样策略也会影响 RRT 算法运行的时间。

3.2 改进的 RRT 算法

3.2.1 引力导向策略

针对 RRT 算法搜索随机、搜索没有目标的缺点，通过借鉴人工势场法中引力场和斥力场思想，本文在基本 RRT 算法的框架中引入一种目标导向策略，将随机树的扩展与引力导向和斥力导向策略相结合，增加了随机树扩展向目标点的概率，不仅有效的减少了随机点的数量，也大大缩短了路径长度，同时缩短了路径规划时间。

人工势场法通过将机器人所在的环境抽象为一个虚拟的势场来进行路径规划。该势场由引力场和斥力场共同组成。其中引力场由目标点产生，其作用是引导机器人向目标位置靠近。随着机器人逐渐接近目标点，引力会不断减小，从而引导机器人持续向目标移动。而斥力场则由障碍物生成，其作用是排斥机器人，使其远离障碍物以避免碰撞。在引力和斥力的共同作用下，机器人沿着合力的方向逐步向目标点前进。图 3- 2 展示了机器人在人工势场算法中的受力情况。

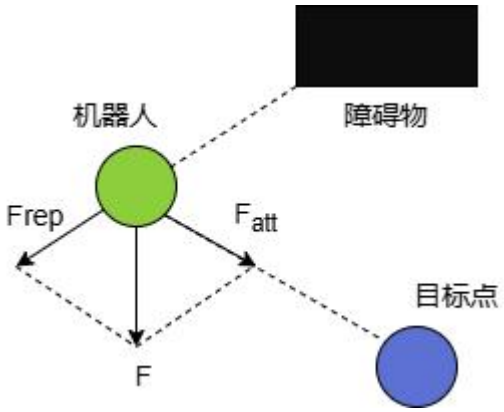


图 3- 2 机器人在人工势场中受力示意图

加入引力导向策略后，随机树每次扩展就会受到引力和斥力及随机点引力的作用，每次扩展的新节点就会受到合力的作用下向目标点扩展。改进的 RRT 算法如图 3- 3 所示，其数学公式如式(3-1)所示：

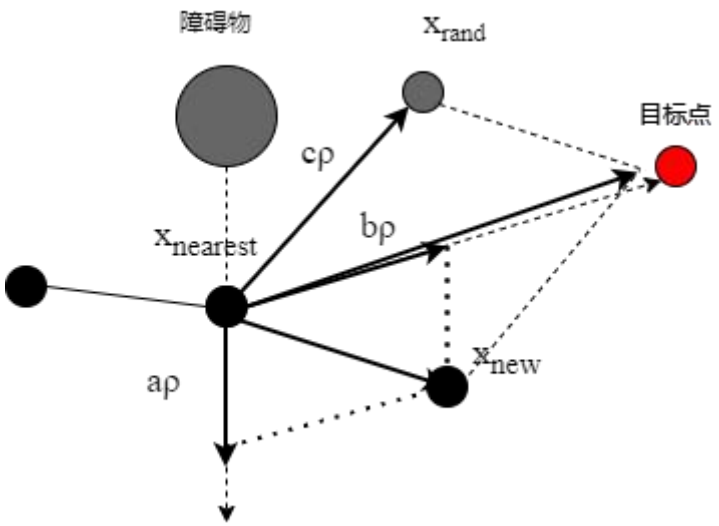


图 3- 3 改进 RRT 算法扩展图

$$F(x_{nearest}) = aF(x_{nearObstacle}) + bF(x_{goal}) + cF(x_{rand}) \tag{3-1}$$

其中a为最近节点受到附近障碍物产生的斥力作用函数的权值，b为目标点对最近节点产生引力作用函数的权值，c为最近节点受到随机点产生引力作用函数的权值，他们满足$a+b+c=1$。$F(x_{nearObstacle})$为最近节点受到附近障碍物产生的斥力作用函数，$F(x_{goal})$为最近节点受到目标点引力作用函数，$F(x_{rand})$为最近节点受到随机点产生引力作用函数$F(x_{nearest})$为最近节点受到附近障碍物斥力和目标引力以及随机点引力合力的作用函数。

节点 $x_{nearest}$ 受到附近障碍物斥力作用函数$F(x_{nearObstacle})$如式(3-2)所示，ρ_o为移动机器人在遇到障碍物时受影响的最远距离。

$$F(x_{nearobstacle}) = \begin{cases} \rho\left(\dfrac{x_{nearest} - x_{nearObstacle}}{\| x_{nearest} - x_{nearObstacle} \|}\right), 0 \le \rho(x_{nearest} - x_{nearObstacle}) \le \rho_o \\ 0, \rho(x_{nearest} - x_{nearObstacle}) \ge \rho_o \end{cases} \tag{3-2}$$

在移动机器人执行任务导航至指定目标点的过程中，其受到的引力会随距离目标点的逼近而逐渐减小。若目标点周围存在障碍物，这将可能导致目标点变得不可达。特别是当障碍物产生的斥力与引力大小相等且方向相反时，机器人可能会在未抵达目标点的情况下陷入局部最优解的状态。为有效克服这一挑战，本文提出通过将原有的式(3-2)修正为新的式(3-3)，以优化移动机器人的路径规划策略。

$$F(x_{no})=\begin{cases}\rho\sqrt{\left(\frac{x_{nearest}-x_{nearObstacle}}{\|x_{nearest}-x_{nearObstacle}\|}\right)\left(\frac{x_{goal}-x_{nearest}}{\|x_{goal}-x_{nearest}\|}\right)}, 0\le\rho(x_{nearest}-x_{nearObstacle})\le\rho_0\\ 0, \rho(x_{nt}-x_{nearObstacle})\ge\rho_0\end{cases} \tag{3-3}$$

在斥力作用函数公式加入了当前节点到目标点的距离，当移动机器人到达目标点，此时斥力减小到零，从而使局部最优和目标不可达问题得到解决。在移动机器人远离目标点的情形下，斥力的相对增强会导致算法陷入局部最小值的问题，为解决相对距离存在造成斥力过大的问题，通过设定一个目标点附近障碍物阈值 ρ_g，当最近节点与目标点距离大于这个阈值 ρ_g 时使用式(3-2)进行节点的扩展，当最近节点与目标点距离小于 ρ_g 时，再使用式(3-3)进行扩展，这就解决了斥力过大的问题。

节点 $x_{nearest}$ 受到目标点引力作用函数 $F(x_{goal})$ 如式(3-4)所示：

$$F(x_{goal})=\begin{cases}\rho\left(\frac{x_{goal}-x_{nearest}}{\|x_{goal}-x_{nearest}\|}\right), \rho(x_{nearest}-x_{goal})<d\\ \rho d, \rho(x_{nearest}-x_{goal})\ge d\end{cases} \tag{3-4}$$

式（3-4）中d为常数，是目标点对最近节点 $x_{nearest}$ 产生引力作用函数的距离阈值，当 $\rho(x_{nearest}-x_{goal})$ 大于等于 d 时，$F(x_{goal})$ 为固定值 ρd；当 $\rho(x_{nearest}-x_{goal})$ 小于 d 时，$F(x_{goal})$ 的大小随着最近节点到目标点的距离变化而动态的变化。通过设定 d，$F(x_{goal})$ 实现了随着 $\rho(x_{nearest}-x_{goal})$ 的变化进行自适应变化。

节点 $x_{nearest}$ 受到随机点引力作用函数 $F(x_{rand})$ 如式(3-5)所示：

$$F(x_{rand})=\rho\left(\frac{x_{rand}-x_{nearest}}{\|x_{rand}-x_{nearest}\|}\right) \tag{3-5}$$

将节点 $x_{nearest}$ 受到来自附近障碍物斥力作用函数 $F(x_{nearObstacle})$ 与目标引力作用函数 $F(x_{goal})$ 以及受到随机点 $F(x_{rand})$ 所产生的引力结合即可得到节点 $x_{nearest}$ 处所受合力作用函数 $F(x_{nearest})$，将式(3-2)、(3-4)、(3-5)代入式(3-1)，可得式(3-6)，将式(3-3)、(3-4)、(3-5)代入式(3-1)，可得式(3-7)。

$$\begin{aligned}F(x_{nearest})=&a\rho\left(\frac{x_{nearest}-x_{nearObstract}}{\|x_{nearest}-x_{nearObstract}\|}\right)+\\ &b\rho\left(\frac{x_{goal}-x_{nearest}}{\|x_{goal}-x_{nearest}\|}\right)+c\rho\left(\frac{x_{rand}-x_{nearest}}{\|x_{rand}-x_{nearest}\|}\right)\end{aligned} \tag{3-6}$$

$$F(x_{nearest})=a\rho\sqrt{\left(\frac{x_{nearest}-x_{nearObstract}}{\|x_{nearest}-x_{nearObstract}\|}\right)\left(\frac{x_{goal}-x_{nearest}}{\|x_{goal}-x_{nearest}\|}\right)}+b\rho\left(\frac{x_{goal}-x_{nearest}}{\|x_{goal}-x_{nearest}\|}\right)+c\rho\left(\frac{x_{rand}-x_{nearest}}{\|x_{rand}-x_{nearest}\|}\right) \tag{3-7}$$

在改进的 RRT 算法中，新节点 x_{new} 的位置通过综合障碍物 $x_{Obstacle}$ 的排斥力、目标点 x_{goal} 的吸引力以及随机点 x_{rand} 的影响共同作用下确定，解决了扩展点随机分布的缺点，使扩展点朝着目标方向扩展。经典 RRT 算法扩展随机树时，沿着 x_{rand} 指向 x_{new} 的方向扩展新节点步长为ρ，此时扩展的新节点 x_{new} 的坐标为式(3-8)。

$$x_{new}=x_{nearest}+\frac{(x_{rand}-x_{nearest})}{\|x_{rand}-x_{nearest}\|}\times\rho \tag{3-8}$$

其中$\|x_{rand}-x_{nearest}\|$的值等于 $x_{nearest}$ 指向 x_{rand} 的向量长度，新节点的坐标 x_{new} 为 $x_{nearest}$ 的坐标加上步长ρ乘以$(x_{rand}-x_{nearest})$的单位向量。

参考式(3-8)加入引力导向策略后，随机树生成的新节点 x_{new} 的位置坐标可表示为式(3-9)或式(3-10)。

$$x_{new}=x_{nearest}+\rho\left(a\left(\frac{x_{nearest}-x_{near0bstract}}{\|x_{nearest}-x_{near0bstract}\|}\right)+b\left(\frac{x_{goal}-x_{nearest}}{\|x_{goal}-x_{nearest}\|}\right)+c\left(\frac{x_{rand}-x_{nearest}}{\|x_{rand}-x_{nearest}\|}\right)\right) \tag{3-9}$$

$$x_{new}=x_{nearest}+\rho\left(a\sqrt{\left(\frac{x_{nearest}-x_{nearObstract}}{\|x_{nearest}-x_{nearObstract}\|}\right)\frac{x_{goal}-x_{nearest}}{\|x_{goal}-x_{nearest}\|}}+b\left(\frac{x_{goal}-x_{nearest}}{\|x_{goal}-x_{nearest}\|}\right)+c\left(\frac{x_{rand}-x_{nearest}}{\|x_{rand}-x_{nearest}\|}\right)\right) \tag{3-10}$$

下表给出了各个符号与相应的释义如表 3- 5。

表 3- 5 目标导向公式符号对照表

符号	释义
$F(x_{nearest})$	最近节点 $x_{nearest}$ 受到合力的作用
$F(x_{nearObstacle})$	最近节点 $x_{nearest}$ 受到附近障碍物斥力的作用
$F(x_{goal})$	最近节点 $x_{nearest}$ 受到目标点引力的作用

续 3- 6 目标导向公式符号对照表

符号	释义
$F(x_{rand})$	最近节点 $x_{nearest}$ 受到随机点引力的作用
a	最近节点 $x_{nearest}$ 受到附近障碍物产生的斥力作用函数的权值
b	目标点对最近节点 $x_{nearest}$ 产生引力作用函数的权值
c	随机点对最近节点 $x_{nearest}$ 产生引力作用函数的权值
ρ	随机数扩展节点的步长
d	目标点对最近节点 $x_{nearest}$ 产生引力作用函数的距离阈值
ρ_o	障碍物区域对移动机器人运动产生影响的最大距离
ρ_g	最近节点到目标点的距离阈值

通过引入引力导向策略后，随机树的扩展从原本无目的的随机扩展转变为向目标方向扩展。通过调整权值参数，可以灵活调整不同力的作用强度，从而获得不同的路径规划结果。这一改进不仅保留了原算法概率完备性的特点，还有效减少了随机树扩展过程中的盲目性，提高了搜索效率。最终随机树的生长会朝着目标点，从而加速路径的生成速度。这一改进为路径规划提供了一种更高效且可控的解决方案，尤其在高维或动态环境中展现了更大的潜力。

3.2.2 动态步长

虽然改进后的 RRT 算法虽然在一定程度上解决了节点随机扩展的问题，并显著提高了路径规划的效率，但是这一算法仍然无法获得最优路径，算法的时间开销仍然很大，固定步长每一次扩展的步长都是一个固定的数值，当步长较小时，算法需要更多的迭代次数才能到达目标点，从而增加了计算成本；而当步长较大时，可能会导致机器人在扩展过程中频繁碰撞障碍物，进而降低路径规划的成功率。固定步长 ρ 的选择严重影响算法的性能，导致算法在路径规划的过程中缺乏灵活性，在复杂环境中宜显出不足，路径质量较低。为了解决这一问题，本文提出动态步长通过自适应调整步长大小，使算法能够根据环境的变化灵活调整扩展策略，从而提高路径规划的效率和避障能力。

动态步长机制分为两个阶段：慢开始阶段和碰撞避免阶段。在初始阶段，为

了避免因步长过大而导致机器人在扩展过程中直接碰撞障碍物，步长ρ被初始化为 1。如果当前扩展的新节点未与障碍物发生碰撞，则更新步长ρ为 2，如果再次扩展还没有碰到障碍物，则更新步长ρ为 4，以此类推成指数式增长，这个阶段称之为慢开始。其目的是在确保安全性的前提下逐步扩大步长，以提高搜索效率。然而，随着步长ρ的指数增长，可能会出现步长过大的情况，从而增加碰撞风险。为增强动态步长机制的理论基础，本文引入了步长阈值（threshold）的设定原则。该阈值初始设定为起点与终点之间距离的一半，旨在在搜索初期提供足够的探索空间，避免过早限制步长增长，从而提升搜索效率。当步长达到或超过该阈值时，算法进入碰撞避免阶段，步长增长方式由指数增长转为线性增长，以降低碰撞风险。此外在扩展过程中首次发生碰撞时，门限值 threshold 将被更新为当前碰撞点距离的一半，同时步长被重新设置为 1，重新进入慢开始阶段。若后续再次发生碰撞，门限 threshold 设置为当前碰撞点距离的一半，而步长不再重置为 1，而是直接设置为新的门限值 threshold，并立即进入碰撞避免阶段。这种动态调整机制使算法能够在复杂环境中灵活应对障碍物分布的变化，从而显著提高避障能力和路径规划的鲁棒性。动态步长示意图如下图 3- 4 所示。

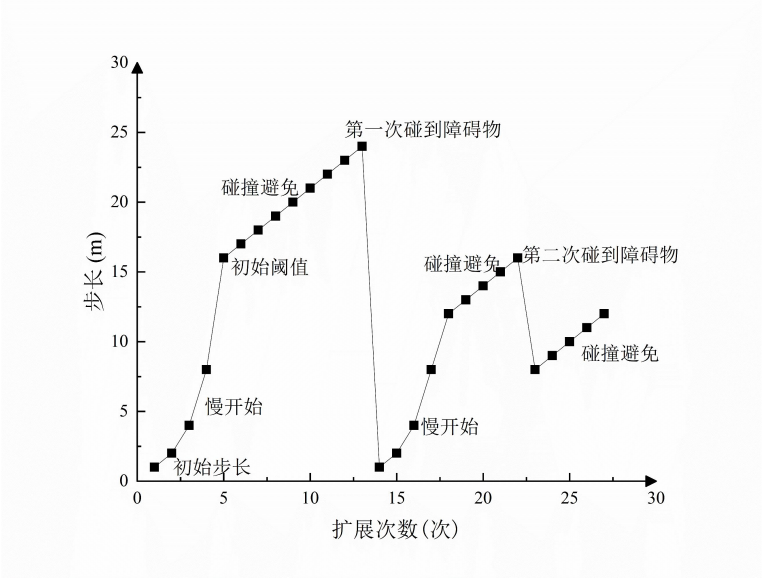


图 3- 4 动态步长示意图

为了研究动态步长机制在不同阈值设定下的性能表现，以及其在不同环境中的适应性和有效性，设计实验，通过在三种不同环境下设置阈值分别为起点与终点之间距离的 1/4、2/4、3/4，观察和分析动态步长机制对路径规划时间的影响。环境一：开阔环境，障碍物数量较少且分布稀疏，路径相对较为简洁。环境二：

中等复杂环境，障碍物数量适中，分布较为均匀，存在一些狭窄通道和局部复杂区域。环境三：复杂狭窄环境，障碍物密集分布，存在众多狭窄通道和复杂结构。实验结果如表 3- 7 所示：

表 3- 7 不同环境不同阈值下路径路径搜索耗时

环境编号	阈值为 1/4 路径长度耗时（秒）	阈值为 1/2 路径长度耗时（秒）	阈值为 3/4 路径长度耗时（秒）
环境一	0.00241126	0.00234286	0.00210320
环境二	0.01256104	0.01208877	0.01226638
环境三	0.03859968	0.02780277	0.04510949

由表可知，在环境较为简单时（环境一），阈值变化对路径规划时间的影响较小，选择较大的阈值可以在一定程度上缩小路径规划时间，但效果不明显。在环境复杂度适中时（环境二），阈值过大或过小都会增加路径规划的难度和耗时，适中的阈值能够取得较好的效果。在环境较为复杂时（环境三），较大的阈值可能会增加路径规划的复杂度和碰撞风险，导致耗时显著增加，而适中的阈值仍能保持较好的性能。

3.3.3 节点的优化

为了进一步优化全局路径的安全性和路径成本，在改进的动态步长 RRT 算法生成完整路径后，本文引入了一种基于贪婪算法的路径优化策略，旨在去除冗余节点并提升路径的整体质量。与传统的路径平滑处理方法相比，该策略在局部路径规划阶段即实施路径优化操作，从而提高了算法的执行效率。传统方法通常需要逐个遍历从起点到终点的所有路径节点，并对每一段路径进行障碍物碰撞检测，这一过程不仅计算复杂度较高，而且容易导致路径中存在不必要的冗余节点。而本文提出的优化方法通过直接在路径生成过程中嵌入平滑处理机制，有效减少了路径中的冗余点，同时保证了路径的安全性和可行性。首先从起始位置出发，尝试将当前节点直接连接到路径中的扩展新节点，并进行碰撞检测。如果新节点与起点之间的连线无障碍，则直接将两点相连，从而跳过中间的冗余节点；否则选择新节点的父节点，判断其与起点之间的连线是否发生碰撞。若无碰撞，则将该父节点加入优化后的路径中，并继续选取下一个节点进行相同的操作。这一过程不断重复，直至达到目标位置，算法终止。

3.3.4 改进算法实现

改进后 RRT 算法的伪代码如表 3- 8 所示。

表 3- 8 改进后 RRT 算法的伪代码

改进的 RRT 算法伪代码
Q←InitializeTree()
For k←1 to N do
x_{rand}←RandomSample(C)
$x_{nearest}$←NearestNeighbor(x_{rand},Q)
$x_{nearObstacle}$←NearestObstacle($x_{nearest}$,Q,ρ_o)
x_{new}←NewState($x_{nearObstacle}$,x_{rand},x_{goal},ρ_d)
E_{new}← NewEdge($x_{nearest}$,x_{new})
If Obstacleless($x_{nearest}$,x_{new}) than
Q.AddNode(x_{new})
Q.AddEdge(E_{new})
End if
If(Distance(x_{new},x_{goal})<=ρ_d)
break
End if
End for
return Q

ρ_d 为随机数扩展的动态步长。

3.3 仿真实验与分析

3.3.1 环境搭建及参数设置

根据 3.2 节提出的基于引力导向动态步长 RRT 算法，为了验证该算法的优越性，本节通过计算机实验仿真来验证算法的可行性。实验的仿真硬件平台基于 Windows 10 操作系统，配备 11th Gen Intel(R) Core(TM) i5-1135G7 2.40GHz CPU 和 16GB RAM，以确保仿真实验的高效运行。软件通过 MATLAB R2021 编程实现，利用其强大的数值计算能力和丰富的工具箱功能，构建了路径规划算法的仿真环境。MATLAB 的可视化特性也为实验结果的分析和展示提供了便利。为了全面评估不同路径规划算法的性能，根据障碍物的分布情况，将实验环境划分为两种典型场景：少障碍物环境和多障碍物环境，障碍物的详细信息分别如表 3- 9、表 3- 10 所示。通过仿真实验，对比分析了经典 RRT 算法、改进 RRT 算法、动态步长 RRT 算法及其优化版本，以探究各自的性能差异。

表 3- 9 少障碍物环境下障碍物信息

障碍物	位置/m	半径/m

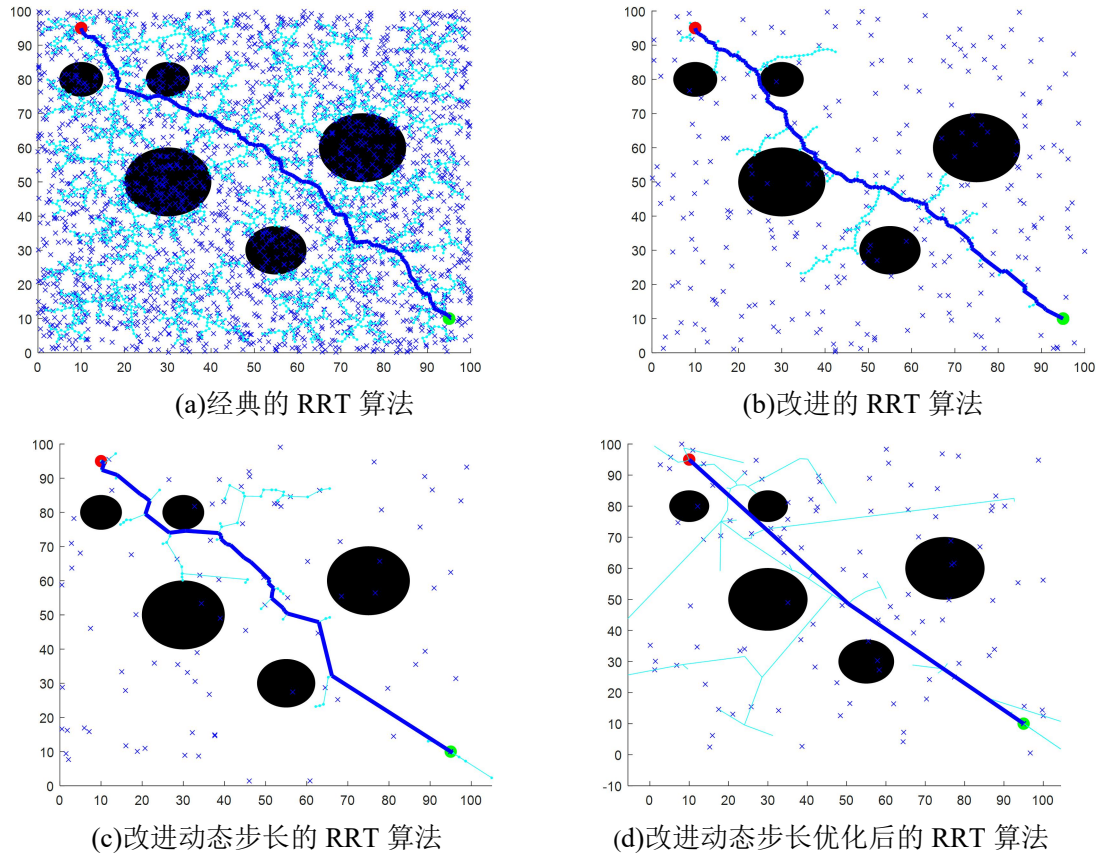
1	(30,80)	5
2	(55,30)	7
3	(10,80)	5
4	(75,60)	10
5	(30,50)	10

表 3- 10 多障碍物杂环境下障碍物信息

障碍物	位置/m	半径/m
1	(30,78)	5
2	(60,90)	6
3	(55,30)	7
4	(10,80)	5
5	(50,60)	10
6	(15,50)	10
7	(35,20)	15
8	(75,25)	8
9	(80,50)	15

3.3.2 实验仿真与结果分析

在本文的路径规划仿真实验中，我们采用了 100m×100m 的仿真地图作为测试环境。为了确保算法能够在有限时间内完成路径搜索，最大迭代次数被限制为 3000 次。少障碍物环境中仿真结果如图 3- 5 所示，多障碍物环境中仿真结果如图 3- 6 所示，其中黑色实心圆形代表障碍物，蓝色×代表随机树生成的随机点，红色实心圆形代表起始点，绿色实心圆形代表终点，青色线代表随机树扩展生成的路径，蓝色粗线代表最终生成的路径。在路径规划中，设定障碍物斥力增益系数 a ＝0.3，目标点引力增益系数 b＝0.5，随机点引力增益系数 c＝0.2，目标点吸引力的有效距离 d=30m，障碍物的安全排斥距离 ρ_o=10m。每次路径扩展的步长 ρ=1m，新扩展节点至目标点的距离阈值 ρ_g =20m。



(a)经典的 RRT 算法　(b)改进的 RRT 算法

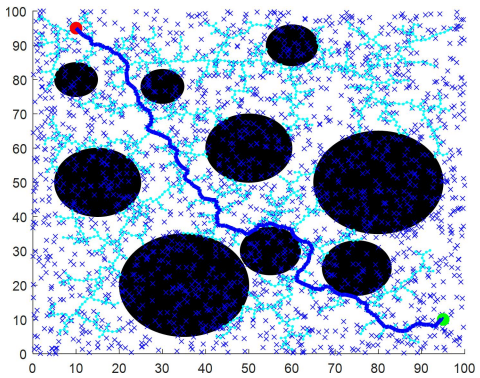
(c)改进动态步长的 RRT 算法　(d)改进动态步长优化后的 RRT 算法

图 3- 5 少障碍物环境下算法对比实验结果图

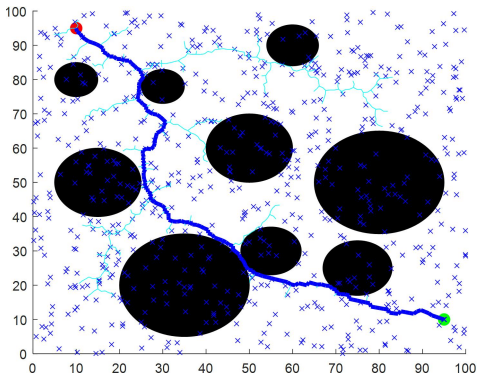
图 3- 5 展示了在少障碍物环境下，不同 RRT 算法及其改进版本的仿真结果对比。图 3- 5（a）展示了在少障碍物环境下，经典 RRT 算法的路径规划结果。由于原始 RRT 算法具有较强的随机性，其扩展过程缺乏明确的方向性，导致地图中生成了大量随机采样点。这些随机节点分布较为分散，使得生成的路径呈现蜿蜒曲折的特点，路径长度较长且转向次数较多。尽管该算法能够成功找到从起始点到目标点的可行路径，但其路径质量和计算效率较低。图 3- 5（b）展示了改进后的 RRT 算法在相同环境下的仿真结果。与原始算法相比，改进后的 RRT 算法在随机树的生长过程中展现出了一定的方向性，这一改进显著减少了生成的随机节点数量，并能够在更短的时间内找到一条从起始点到目标点的有效路径。通过引入方向性指导，算法在路径规划过程中的效率和准确性得到了显著提升。在图 3- 5（c）中，展示了在改进算法基础上加入动态步长调整的 RRT 算法的仿真结果。该算法能够根据当前环境动态地调整随机树扩展的步长，在靠近障碍物碰撞时，算法动态调整步长，以增加路径的灵活性和安全性；而在远离障碍物时，则增大步长，以加快路径的探索速度。这一动态步长调整策略不仅进一步减少了生成树节点的数量，还极大地降低了路径的转向次数，使得生成的路径更加简洁

和高效。图 3- 5（d）展示了在少障碍物环境下，采用改进动态步长优化后的 RRT 算法的仿真结果。该算法在保持上述改进的基础上，进一步去除了路径中的冗余节点，使得生成的路径更加平滑和连续。这一优化不仅提高了路径的平滑性，还进一步提升了路径执行的效率和稳定性。

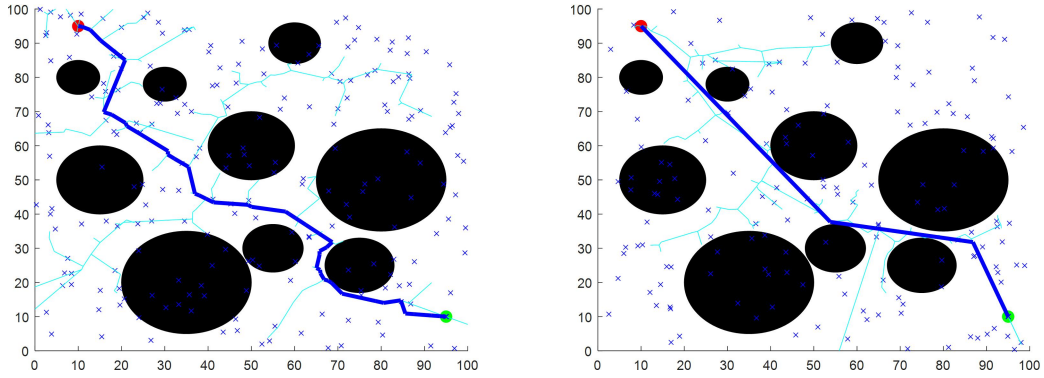
图 3- 6 展示了在多障碍物环境下，不同 RRT 算法及其改进版本的仿真结果对比。图 3- 6（a）展示了多障碍物环境下经典 RRT 算法的仿真结果。在该环境下，算法生成的节点显得杂乱无章，且并非每次都能找到可行的路径，这凸显了经典 RRT 算法在复杂环境中的局限性。图 3- 6（b）展示改进 RRT 算法在多障碍无的路径规划结果，相较于经典 RRT 算法，改进算法通过引入引力导向策略，显著减少了随机树中冗余节点的数量，路径的整体平滑性得到了一定程度的提升。尽管改进后的算法在路径质量上有所提高，但生成的路径中仍然存在较多冗余节点。在图 3- 6（c）中，展示了在改进算法基础上加入动态步长调整的 RRT 算法的仿真结果。与改进后的 RRT 算法相比，该算法生成的路径中节点数量明显减少，这得益于动态步长。动态步长使得算法能够根据障碍物信息动态调整步长，从而生成更加简洁和高效的路径。图 3- 6（d）展示了多障碍物环境下采用改进动态步长优化后的 RRT 算法的仿真结果。与改进动态步长的 RRT 算法相比，优化后的算法在路径中冗余节点数量上实现了显著减少，路径变得更加平滑。



(a)经典的 RRT 算法



(b)改进的 RRT 算法



(c)改进动态步长的 RRT 算法　　(d)改进动态步长优化后的 RRT 算法

图 3- 6 多障碍物环境下算法对比实验结果图

为了验证算法的鲁棒性，我们将上述算法分别在少障碍物环境和多障碍物环境中连续运行 100 次，并记录了平均仿真数据，实验结果分别如表 3- 11 和表 3- 12 所示。通过对这些数据的深入分析，我们发现改进的 RRT 算法在更短的时间内能够找到长度更短的路径，同时生成的树节点数量也最少。在少障碍物环境下，与传统 RRT 算法相比，最终改进算法的路径长度缩短了约 18.56%，时间缩短了约 99.80%，路径节点个数减少了约为原来的 97.87%。路径中的冗余节点大大减少，生成的路径轨迹更加光滑。同样，在多障碍物环境下，与传统 RRT 算法相比，最终改进算法的路径长度缩短了约 15.26%，时间缩短了约 98.66%，路径节点数量减少了约为原来的 96.71%。这一改进大大增强了算法在多障碍物环境中的避障能力。以上数据不仅有效地证明了本文提出的改进动态步长优化后的 RRT 算法在稳定性方面的优势，同时也显著提高了规划效率，兼顾了路径的平滑性，并有效减少了路径长度。

表 3- 11 少障碍物环境下算法仿真数据

环境	算法种类	路径长度/m	路径节点个数	规划时间/s
少障碍物环境	经典的 RRT 算法	154.9200	153.6400	1.1519
	改进的 RRT 算法	137.9400	138.1400	0.0132
	改进动态步长的 RRT 算法	145.5173	38.14	0.0102
	改进动态步长优化后的 RRT 算法	126.1729	3.2700	0.0023

表 3- 12 多障碍物环境下算法仿真数据

环境	算法种类	路径长度/m	路径节点个数	规划时间/s
	经典的 RRT 算法	157.8306	159.2000	0.8357

多障碍物环境	改进的 RRT 算法	146.5539	148.0400	0.0313
	改进动态步长的 RRT 算法	151.5626	59.0800	0.0124
	改进动态步长优化后的 RRT 算法	133.7444	5.2300	0.0112

为了进一步验证本文提出的改进动态步长优化后的 RRT 算法的性能，我们分别在相同环境下对其与 RRT-Connect、RRT*进行了 100 次仿真实验，比较了寻路时间和所需迭代次数，结果如图 3- 7 和图 3- 8 所示。从图 3- 7 可以明显看出，本文算法在 100 次模拟实验中所用的时间要比其他三种算法少得多，由于坐标精度的限制，速度的提升无法被明显观测到，因此 RRT*实验的次数和时间关系图是单独绘制出来的。从图 3- 8 可以看出，由于动态步长的加入，本文算法在迭代次数相较于其他算法也展现出了较为明显的优势。以上数据不仅有效地证明了本文提出的改进动态步长优化后的 RRT 算法在稳定性方面的优势，同时也显著提高了规划效率，兼顾了路径的平顺性，并有效减少了路径长度。

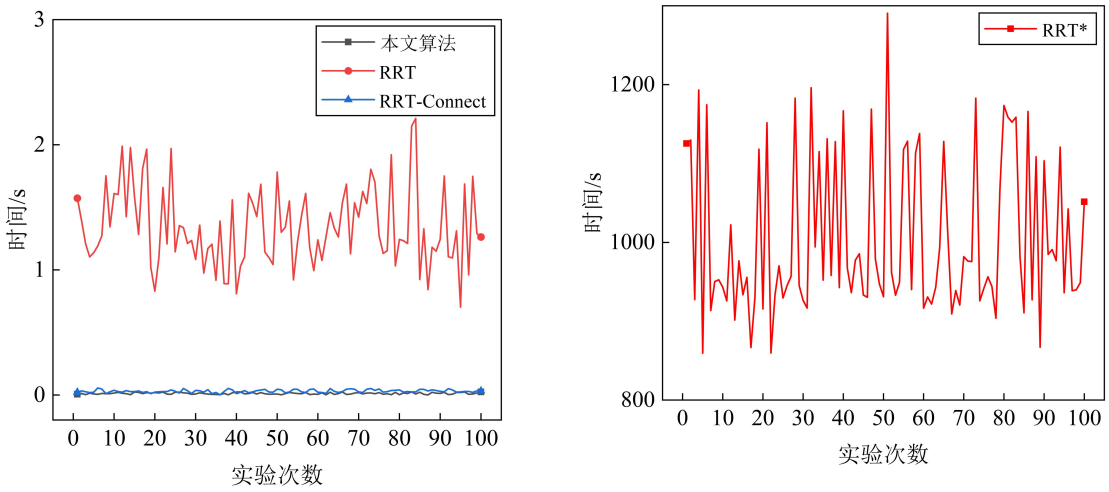


图 3- 7 寻路时间对比图

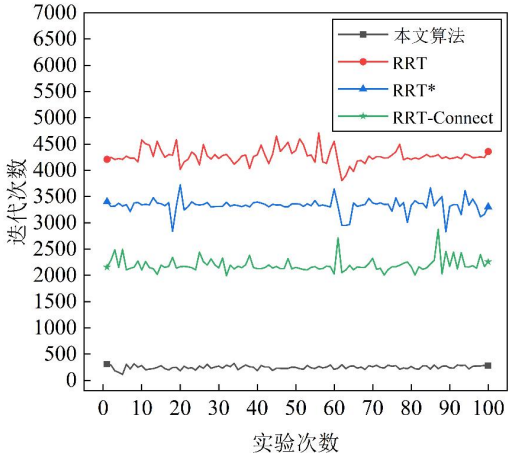


图 3- 8 迭代次数对比图

表 3- 13 展示了四种算法在 100 次实验中的迭代次数、路径长度以及生成路径所需时间的平均值统计结果。通过对这些数据的深入分析，我们可以得出以下结论：与经典的 RRT 算法相比，本文算法在迭代次数上展现出极大的效率提升，仅需 RRT 算法的 65.4%即可完成路径搜索，这显著降低了算法的计算复杂度。同时，在路径长度方面，本文算法也仅需 RRT 算法的 81.4%，能够生成更为高效的路径。此外，在路径规划时间这一关键指标上，本文算法更是仅需 RRT 算法的 1.1%。与 RRT-Connect 算法相比，本文算法同样表现出显著的提升。在路径规划时间上，本文算法仅需 RRT-Connect 算法的 41.3%。在迭代次数上，本文算法较 RRT-Connect 算法减少了 87.7%。在路径长度上，本文算法也较 RRT-Connect 算法减少了 24.6%。与 RRT*算法相比，本文算法在路径规划时间方面实现了惊人的 99.9%的减少。这主要是因为 RRT*算法包含了复杂的重选父节点和重布线过程，从而增加了算法的时间成本。尽管 RRT*算法能够生成质量较高的路径，但本文算法在保持路径质量相当的前提下，通过减少迭代次数和时间成本，实现了更加高效的路径规划。本文算法在迭代次数上较 RRT 算法减少了 92.1%，证明了本文算法在计算效率上的巨大优势。最终生成的路径长度与 RRT*算法相比并没有显著的差异，表明了本文算法在路径规划方面的有效性和实用性。

表 3- 13 仿真实验结果比较

算法	迭代次数	路径长度	生成时间
RRT	4216.00	154.9200	1.1519
RRT*	3482.71	122.0858	864.69
RRT-Connect	2236.63	167.2458	0.0298
本文最终改进算法	275.8500	126.1729	0.0123

综上所述，本文所提出的改进算法在迭代次数、路径长度以及路径生成时间方面均展现出了显著的优势。这些优势不仅提高了路径规划的效率，还保持了路径的质量，为路径规划领域的研究提供了新的思路和方法，具有重要的学术价值和实践意义。

3.4 本章小结

本章节主要探索 RRT 算法的优化提升，通过深入剖析经典 RRT 算法，结合多方面改进策略，旨在满足移动机器人在不同环境下高效路径规划的需求。详细

介绍了经典 RRT 的基本原理、算法流程及节点扩展方式，并指出存在随机性过强和路径冗余的问题。为此本文引入了人工势场中的目标引力理念，通过建立目标吸引力场、障碍排斥力场以及随机扰动场，实现动态权重分配，从而使随机树在规避障碍的同时保持向目标区域前进。为提高收敛效率和避障精度，设计了一种动态步长调节机制，使得在空旷区域步长可较快增加，而在复杂区域则自动缩小，从而保证安全性与准确性。最后针对生成路径存在的冗余节点和平滑性不足问题，采用贪婪算法对初始路径进行局部优化，既提高了路径质量，也降低了计算开销。通过 MATLAB 仿真实验，对比分析了扩展节点数量、搜索时间和路径长度等指标，结果表明改进后的算法能显著缩短路径、加快搜索速度，为全局路径规划提供了更为有效的解决方案。

第 4 章 改进的 DQN 路径规划算法

4.1 Q-Learning 算法

Q-Learning 算法是强化学习领域中一类重要的时序差分算法，主要用于解决马尔可夫决策过程相关问题。该算法的优势在于其无模型特性，它无需预先掌握环境的完整模型即可开展学习，为复杂场景下的应用提供了便利。在移动机器人路径规划中，Q-Learning 能够通过学习最优路径策略，帮助机器人在动态复杂的环境中高效完成从起点到目标点的导航任务。

在强化学习框架中，Q-Learning 通过构建状态-动作价值实现智能决策机制。其本质在于建立环境状态空间与可行动作集合之间的映射关系矩阵，每个元素 Q(s,a)不仅反映当前状态 s 下执行动作 a 的即时收益，也包含对未来多步决策潜在收益的预估。这种时间跨度上的价值累积特性，使得 Q 表成为智能体进行长期收益最大化决策的核心依据。执行动作时，智能体通过检索 Q 表来选择当前状态下最大 Q 值的动作。表 4- 1 为 Q 表，其中 s 代表状态，a 代表动作，给出了不同状态动作组合下的值分布情况。

表 4- 1 Q 值表示例

Q-table	a1	a2	a3	…
s1	Q(s1,a1)	Q(s1,a2)	Q(s1,a3)	…
s2	Q(s2,a1)	Q(s2,a2)	Q(s2,a3)	…
s3	Q(s3,a1)	Q(s2,a2)	Q(s2,a3)	…
…	…	…	…	

4.1.1 Q-Learning 原理

Q-Learning 算法致力于对状态-动作对所对应的 Q 值函数 Q(s,a)进行精准估计。在算法的运行过程中，它借助迭代学习机制，对各种潜在的行为进行全面考量，保障学习路径能够实现稳健的收敛，确保算法在复杂多变的环境中能够有效地学习到最优策略。其运行流程与 TD 学习有着诸多相似之处，首先对初始状态 s1 下 Q(s,a)的初始化，随后智能体遵循策略选取动作 a，使状态从 s1 转变为 s2，在此过程中，智能体收获即时奖励 r，并据此更新 Q 值，直至达成预设目标状态或达到最大迭代次数，一个完整的学习循环才算结束。这一过程会不断循环往复，

直至算法最终收敛。

Q 值的更新公式如（4-1）所示，s_t 表示 t 时刻的状态，a_t 为在该状态下所选择的动作，r_t 则是从状态 s_t 过渡到 s_{t+1} 所获得的即时回馈。在到达下一个状态后，算法从动作集合 A 中挑选出最大 Q 值的动作，以此作为 Q 值更新。

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha\left[r_t+\gamma \max_{a\in A} Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\right] \tag{4-1}$$

在 Q 值公式中，α是学习率，其取值范围在[0,1]之间，主要用于控制 Q 值的更新速度。若α取值偏高，算法虽可能迅速收敛，但容易陷入局部最优的困境，难以触及全局最优解；反之若α取值偏低，虽能确保算法的平稳运行，但收敛过程将变得冗长，耗时显著增加。合理设定α的值，对于优化 Q-Learning 算法的性能至关重要。γ为折扣因子，其取值范围为[0,1]，$\max_{a\in A} Q(s_{t+1},a_{t+1})$ 为从动作集合 A 里挑选出使 Q 值最大的那个动作。

Q-Learning 算法伪代码如表 4- 2 所示。

表 4- 2 Q-Learning 算法伪代码

Q-Learning 算法
定义状态空间 S，动作空间 A，折扣因子ϒ，学习率α，针对状态空间 S 中任意状态 s 以及动作空间 A 中的任意动作 a，随机初始化 Q(s,a)
For 每一幕 do
初始化开始状态 s
Repeat
根据状态 s 通过ε-greedy 策略选择动作 a
执行动作 a，观察状态 s_{t+1} 和奖赏函数 r
更新 Q(s,a): $Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha\left[r_t+\gamma \max_{a\in A} Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\right]$
$s \leftarrow s_{t+1}$
until s 是终止状态
end for

Q-Learning 的流程图如图 4- 1 所示。

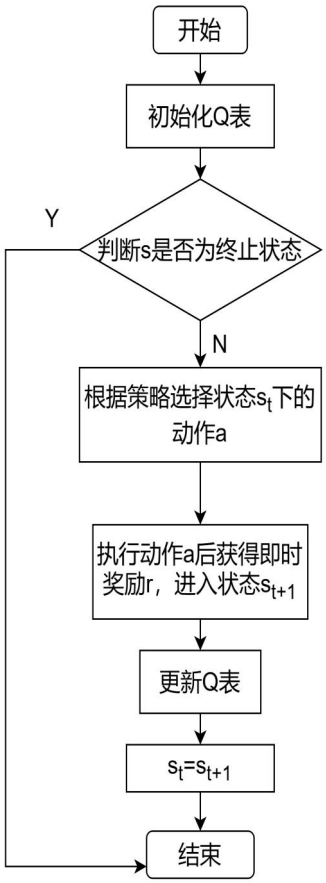


图 4- 1 Q-Learning 流程图

4.1.2 算法收敛性分析

Q-Learning 算法的核心机制在于构建一个状态-动作价值的映射表，此表实时更新并存储智能体在与环境交互过程中累积的决策知识。值得注意的是，只有当后续状态的价值评估趋于稳定时，当前状态的估值才能达到稳定态；若后续状态的 Q 值持续波动，当前状态的更新过程将陷入无限递归的困境。其更新公式可表示为式（4-2）：

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[R_t + \gamma \max_a Q(s',a')] \tag{4-2}$$

式（4-2）的中 $\max_a Q(s',a')$ 项的渐进稳定性反映了 Q 函数迭代的收敛趋势，确保后续状态的价值变动不会逆向干扰当前估值。以初始 Q 值为基础执行首次迭代后，得到迭代后的价值评估如式（4-3）所示：

$$Q_1(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[R_t + \gamma \max_a Q(s',a')] \tag{4-3}$$

当执行第二次参数更新时，将 Q_1 代入原式进行运算，经过迭代变换后形成式（4-4）：

$$Q_2(s,a) \leftarrow (1-\alpha)^2 Q(s,a) + [1-(1-\alpha)^2][R_t + \gamma max_a Q(s',a')] \tag{4-4}$$

以此类推，经过 N 次迭代后，得到$Q_n(s,a)$的通式，如式（4-5）所示：

$$Q_n(s,a) \leftarrow (1-\alpha)^n Q(s,a) + [1-(1-\alpha]^n][R_t + \gamma max_a Q(s',a')] \tag{4-5}$$

其中，学习率α的取值范围在(0,1)之间，此时 0<1−α<1。当迭代次数 n 趋于无穷大时，$(1-\alpha)^n$趋近于 0。此时$Q(s,a)$逼近$R_t + \gamma max_a Q(s',a')$，此时当前状态的 Q 值趋于稳定，因此$Q(s,a)$收敛。这一过程不仅展示了 Q-Learning 算法的数学严谨性，也揭示了其在实际应用中实现智能体行为优化的潜力。

4.2 DQN 算法

Q-Learning 路径规划算法在理论上展现出较强的学习能力并能实现收敛，但在实际应用中仍面临诸多挑战。现实环境往往呈现出复杂多变的特征，随着场景复杂度的提升，环境特征数量急剧增加，导致状态和动作空间呈现指数级扩张。当训练过程中状态空间快速扩展时，传统 Q-Learning 依赖的 Q 值表将面临存储和计算难题，状态维度的增加会导致 Q 值表规模呈指数增长，这不仅对计算机内存资源构成严峻挑战，还会降低算法的处理效率，最终引发“维度灾难”。在这种情况下，算法难以有效学习到合理的路径规划策略，影响实际应用效果。针对这一问题，深度学习技术凭借其强大的特征提取能力，能够从高维数据中提炼出关键特征。而深度强化学习则通过深度神经网络架构，成功建立起高维状态空间与连续动作空间之间的映射关系。这种技术优势使得深度强化学习在解决复杂大规模决策控制问题时表现出显著优势，为突破传统 Q-Learning 的应用瓶颈提供了新的解决方案。

DQN 算法通过神经网络架构实现了 Q 值表的高效模拟，降低了 Q-Learning 计算复杂度。尽管这种近似处理可能导致智能体在探索环境时的精度有所下降，但它成功突破了高维状态空间下的维度灾难瓶颈。通过将状态特征映射到动作价值的非线性变换，为复杂系统中感知与决策的融合问题提供了有效的解决途径。这种基于深度网络的 Q 值估计方法，既保留了强化学习的目标导向特性，又借助深度学习的表征能力实现了对高维信息的高效处理。

4.2.1 神经网络拟合 Q 值表

深度强化学习框架中，通过构建多层卷积结构实现 Q 值函数的参数化建模，建立了从原始状态空间到决策价值空间的高维非线性映射。使用权值矩阵集合ω

表征的深度网络替代传统查表机制，将状态-动作对的评估过程转化为神经网络的参数优化问题。在网络设计中，输入状态 s，经由网络末端的全连接层计算得到所有候选动作对应的 Q(s,a,ω)值，其定义如公式（4-6）所示。

$$Q(s,a,\omega) \approx Q(s,a) \tag{4-6}$$

4.2.2 经验回放机制

深度 Q 网络算法中的关键组成部分是经验回放机制，其根本目的在于应对强化学习中数据样本高度相关性所带来的训练不稳定性。在传统强化学习的背景下，智能体与环境互动产生的数据常常伴随着强烈的时序依赖性，若直接将这些数据用于深度神经网络的训练，可能会引发模型损失值的剧烈震荡乃至训练发散。为解决这一问题，DQN 算法引入经验回放机制，通过构建经验存储池对历史交互数据进行管理。通过将智能体与环境交互产生的经验数据以四元组(s,a,r,s')的形式存储到经验池中。其中 s 为当前状态，a 为执行动作，r 为即时奖励，s'为下一状态，在训练阶段采用随机采样策略从池中抽取小批量数据进行模型更新。这种随机采样方式具有多重优势，打破了数据间的时间依赖性，使训练数据更接近独立同分布假设，显著升了模型的稳定性；其次通过复用历史数据减少了对实时交互的依赖，降低了计算成本；多样化的样本分布避免了参数更新陷入单一梯度方向，有效缓解了过拟合风险。每次参数更新基于不同的随机样本组合，使得梯度方向的调整更具多样性，从而减少了损失值震荡现象。这种设计使得 DQN 在处理高维状态空间和复杂决策任务时表现出更强的鲁棒性，经验回放机制结构如图 4- 2 所示。

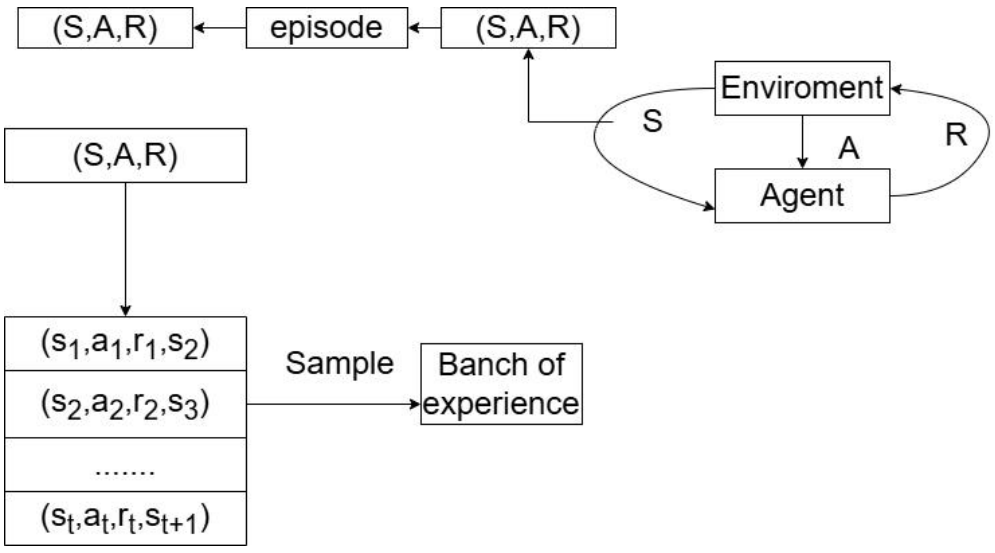


图 4- 2 经验回放机制结构

4.2.3 目标网络

在 DQN 算法的设计框架中，目标网络是保障学习过程稳定性的关键环节。目标网络的目标函数记为$Q(s,a,\omega^-)$，其中ω^-代表目标网络的参数，在实际的训练进程里，目标网络会从当前 Q 网络获取参数，每经过一定次数的迭代，目标网络便会从当前 Q 网络中复制一份参数，使目标网络在每隔 N 步更新时能够保持相对稳定。这种稳定性对于整个 DQN 算法的性能提升意义重大，有效地减少了学习过程中的波动，使得算法在复杂环境下也能更加稳健地运行。

在 DQN 算法的运行机制中，状态-动作值函数由当前网络输出，与之相对，目标网络输出$Q^-(s_{t+1},a_{t+1},\omega^-)$，在固定时间间隔内复制当前网络的权重。当前网络值函数公式如式（4-7）所示：

$$y_{DQN}=R_t+\gamma max_{a_{t+1}}Q(s_{t+1},a_{t+1},\omega) \tag{4-7}$$

公式结合了即时奖励R_t和对未来状态的估计，通过状态值函数的迭代更新，Q 值函数得以不断改进。随后当前网络会依据这一更新结果对权重参数进行调整，以优化模型的输出。在经过一定步数后，目标网络会复制当前网络的参数，实现自身的更新，目标网络的目标值函数计算公式见式（4-8）：

$$y_{DQN}=R_t+\gamma max_{a_{t+1}}Q^-(s_{t+1},a_{t+1},\omega^-) \tag{4-8}$$

神经网络通过循环更新的工作机制，逐步调节其内部连接权值，促使目标函数逐步收敛至最优值。在参数空间搜索过程中，模型采用梯度下降策略持续修正各层节点间的权重系数，使得系统输出的误差度量呈现递减趋势，最终完成对网络结构的优化配置。损失函数的表达式如式（4-9）所示：

$$L(\omega)=E_{(s,a,r,s')}\left[\left(R_t+\gamma max_{a_{t+1}}Q^-(s_{t+1},a_{t+1},\omega^-)-Q(s,a,\omega)\right)^2\right] \tag{4-9}$$

接着通过对损失函数关于网络参数求偏导来计算梯度，计算方式如式（4-10）：

$$\frac{\partial L(\omega)}{\partial \omega}=E_{(s,a,r,s')}\left[\left(R_t+\gamma max_{a_{t+1}}Q^-(s_{t+1},a_{t+1},\omega^-)-Q(s,a,\omega)\right)\nabla_\theta Q(s,a,\omega)\right] \tag{4-10}$$

在 DQN 算法中，通过引入双网络架构分离了时间差分误差的计算形式，包含当前网络和目标网络。当前网络用于预测动作价值，其根据当前状态与已执行动作，动态估计下一状态的最大潜在收益，并将此预测值与实际获得的即时奖励进行对比，从而生成 TD 误差。目标网络则专注于提供稳定的目标 Q 值参考，其

参数在周期内保持固定，仅在周期性同步当前网络参数的方式进行缓慢调整。两套网络通过协同作用，共同推动算法收敛，从而使 DQN 在复杂强化学习任务中展现出卓越的性能。

4.3 改进 DQN 算法

4.3.1 目标网络优化

在深度强化学习领域，传统 Q 学习采用查表法处理低维状态时表现良好，但面对高维复杂场景时神经网络的应用成为必然选择。尽管 DQN 算法通过函数拟合突破了传统方法在复杂场景中的应用限制，其估值虚高现象成为制约性能的关键瓶颈，当网络对动作价值的预测偏离真实水平时，智能体的决策质量将显著下降。为破解这一难题，Van Hasselt 等人[63]提出的双网络架构，为 DQN 开创了新思路，通过构建决策网络与评估网络之间的协同机制，实现动作选择与价值估计两个核心步骤的分离，从而有效降低了 Q 值高估的风险，并提升了算法的整体稳定性。目标 Q 值的计算公式如式（4-11）所示，其中目标网络的参数定期从当前网络同步，以确保学习过程的连续性与可靠性。

$$\mathrm{y_{DDQN}} = \mathrm{R_t} + \gamma Q(s_{t+1}, \mathrm{argmax_{a_t}} \mathrm{Q(s_{t+1}, a_t}, \omega); \omega') \tag{4-11}$$

在模型优化过程中，为兼顾不同算法特性带来的知识增益，将 DQN 的快速响应优势与 DDQN 的精准评估特性相融合，改进后的目标 Q 值计算公式如式（4-12）所示，其中权重系数根据训练阶段动态调整。这种设计不仅继承了 DQN 的快速收敛特性，还通过双网络架构降低了 DDQN 对 Q 值的过高估计风险，从而在复杂任务中实现更稳定的策略学习。

$$y = x_n y_{dqn} + (1 - x_n) y_{ddqn}, x_n \in [0,1] \tag{4-12}$$

其中，$\mathrm{x_n}$ 是权重系数，其计算公式如式（4-13）：

$$x_n = \frac{1}{1 + \lambda \cdot n} \tag{4-13}$$

其中，n 为每次训练迭代的次数，λ是一个正的常数，取值范围在 0-0.1 之间，用于控制权重系数的衰减速度。随着训练次数的增加，x_n 逐渐降低，这意味着在训练初期，DQN 的目标 Q 值占较大比重，而在训练后期，DDQN 的目标 Q 值逐渐占据主导地位。这种动态调整权重系数的方法有助于平衡 DQN 和 DDQN 的优点，从而进一步优化目标 Q 值的计算。

4.3.2 奖励函数设计

强化学习系统中，环境反馈机制的核心在于奖励信号的精准设计，经典 DQN 框架采用三阶奖励结构如式（4-14）所示，在成功抵达目标位置时触发正向激励，遭遇障碍碰撞时启动负向惩罚，而中间过渡状态则处于无反馈区域。这种设计虽然能够清晰标记关键节点，但中间的大部分动作缺乏有效引导容易陷入低效探索的困境。

$$r(s_t,a_t)=\begin{cases} r_{reach} & \text{if } s_{t+1} \text{ is the target state} \\ r_{crash} & \text{if } s_{t+1} \text{ is the obstacle state} \\ r_{step} & \text{otherwise} \end{cases} \tag{4-14}$$

为解决这一问题，本文对奖励函数进行改进，在智能体执行每个动作时根据其与目标的距离提供动态反馈，通过引入调节参数 $C1$ 和 $C2$ ，结合路径评估指标 $r1$ 和 $r2$ ，构建了动态激励机制。奖励函数改进为式 4-15 所示。

$$R_t = C_1 r_1 + C_2 r_2 \tag{4-15}$$

假设当前状态的坐标是 s_t，下一状态的坐标为 s_{t+1}，目标状态的坐标为 s_g。我们把当前状态点到目标点的距离记为 $d(s_t,s_g)$，下一状态点到目标点的距离记为 $d(s_{t+1},s_g)$，这两个距离可以通过式(4-16)和式(4-17)来表示。

$$d(s_t,s_g)=\sqrt{\left(x_t-x_g\right)^2+\left(y_t-y_g\right)^2} \tag{4-16}$$

$$d(s_{t+1},s_g)=\sqrt{\left(x_{t+1}-x_g\right)^2+\left(y_{t+1}-y_g\right)^2} \tag{4-17}$$

定义 $\Delta d = d(s_{t+1},s_g)-d(s_t,s_g)$，当 Δd >0 时，表明移动方向偏离目标区域，系统将触发负反馈；反之则给予正反馈以强化趋近行为。r_2 仍沿用静态阈值判断法，仅在完全抵达或碰撞时触发奖惩事件。

4.3.3 探索策略设计

在智能体路径寻优过程中，动作决策机制直接影响环境探索与策略优化的平衡。完全依赖即时奖励引导易使移动体陷入局部极值困境，而过度随机探索又会显著降低决策效率。现有路径规划方法中广泛采用的ε-greedy 策略虽能缓解这一矛盾，通过设定探索概率参数ε实现经验利用与未知区域探测的折中，但在实际应用时仍存在改进空间。传统方法的探索参数多采用固定阈值设计，这在算法训练后期会引发新的矛盾，当环境信息趋于完整时，持续保持固定比例的随机探索

不仅降低决策质量，更可能使智能体错失潜在优化路径。针对这一痛点，本研究设计了具有自适应特性的探索策略，公式如（4-16）所示：

$$\ln\left(1.698x_n+1\right) \tag{4-16}$$

参数 x_n 定义为当前训练轮次与总轮次之比。随着训练进程推进，探索强度呈现非线性递增趋势，这种动态调节机制在训练中后期展现出独特优势：当 x_n 值趋近于 1 时，神经网络基于经验知识的决策权重显著提升，使得智能体既能避免早期过度探索导致的低效徘徊，又能防止后期随机干扰引发的策略退化。函数图像如图 4- 3 所示。

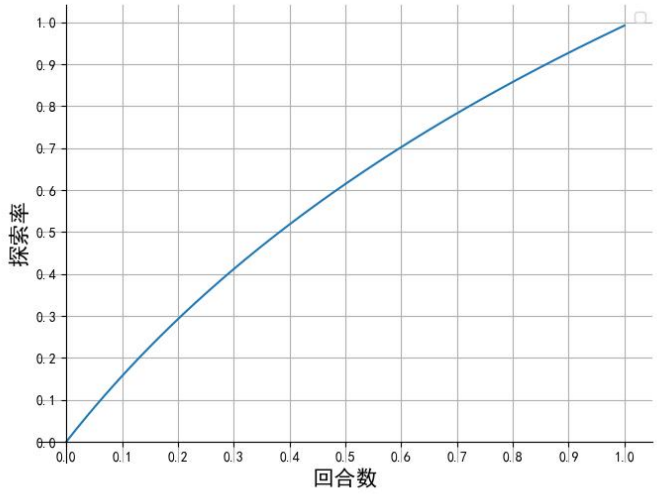


图 4- 3 探索函数

4.4 仿真实验分析

4.4.1 环境搭建及参数设置

本文仿真实验配置如表 4- 3 所示。

表 4- 3 仿真实验配置

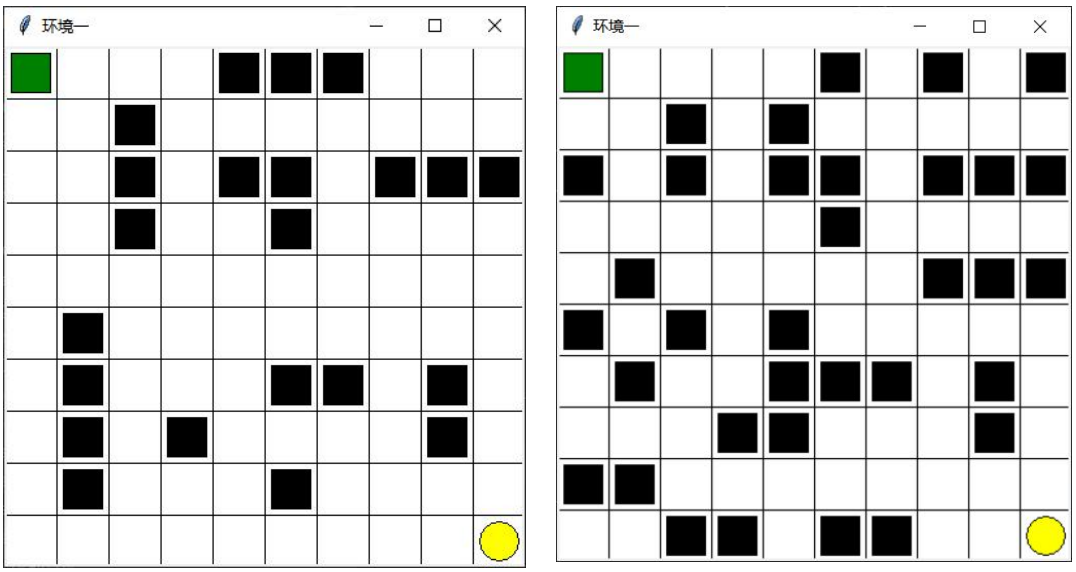
名称	版本型号
CPU	i5-1135G7
RAN/GB	16
系统	Windows 10 专业版
Gpu	NVIDIA GeForce MX330
Python	3.9
Gym	0.8.0
TensorFlow	1.10

仿真系统依托 Python 编程环境搭建，借助 Tkinter 图形库，设计了两种不同

的训练场景，具体展示于图 4- 4 所示。图 4-4（a）呈现的是相对简单的环境，而图 4- 4（b）则展示了一个更为复杂的环境。两种环境的像素尺寸均为 400×400，智能体的最小移动单位设定为 40 像素。在每一个栅格中，智能体均代表一个状态，且在每个状态下，智能体可执行上、下、左、右四个基本动作。图中智能体的起始位置由左上角的绿色正方形标识，黑色正方形则代表障碍物，右下角的黄色圆圈为目标点，其余白色方块均为安全区域。为提升任务复杂度，两种环境下的目标点均被置于较远位置。为确保实验对比的公平性，传统 DQN 与改进 DQN 算法的所有超参数均被统一设定，具体初始化参数配置详见表 4- 4。

表 4- 4 网络模型超参数设定

参数	取值
学习率	0.01
折扣因子	0.95
经验池	2000
批处理数	50
更新频率	100
训练回合数	100



（a）简单环境地图　　（b）复杂环境地图

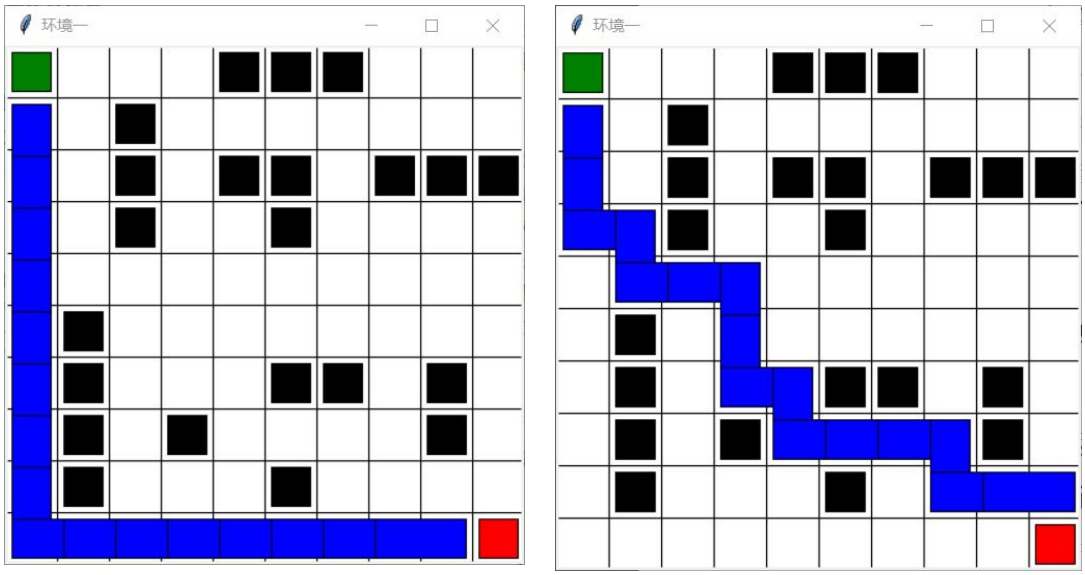
图 4- 4 环境地图

4.4.2 实验结果与分析

（1）在简单环境中仿真及分析

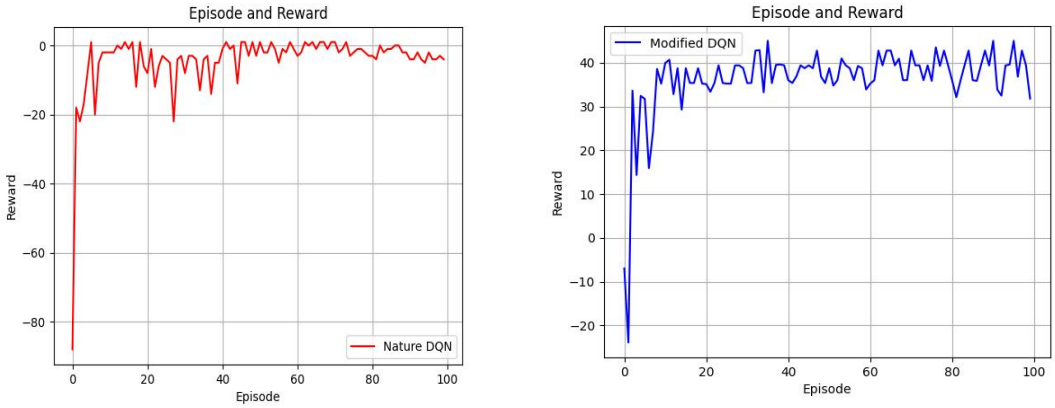
在简单环境下，实验结果显示两种仿真算法均得到了最终的路径规划方案，

详见图 4- 5，累积奖赏值如图 4- 6 所示，损失函数如图 4- 7 所示。



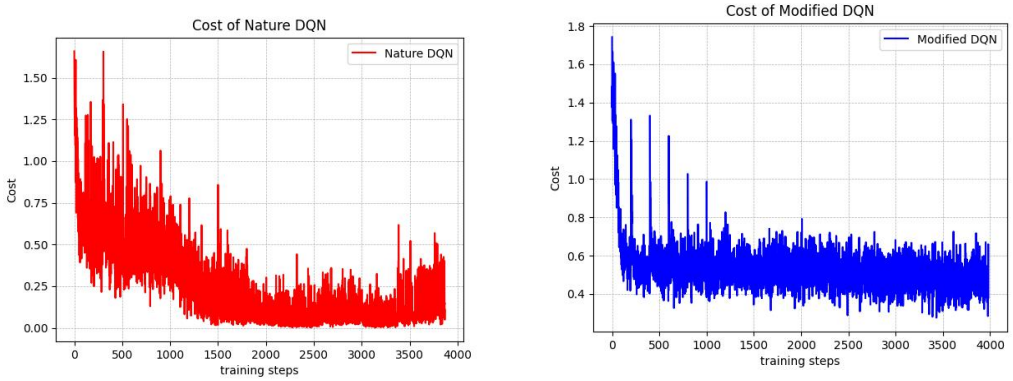
（a）传统 DQN 算法　　（b）改进 DQN 算法

图 4- 5 简单环境下仿真路径规划图



（a）传统 DQN 算法　　（b）改进 DQN 算法

图 4- 6 简单环境下两种算法奖励值变化图



（a）传统 DQN 算法　　（b）改进 DQN 算法

图 4- 7 简单环境下两种算法损失函数图

从图 4- 5（b）中，我们可以看到智能体采用改进的 DQN 算法进行路径规划时，其每次前进的趋势是朝向目标点的。蓝色路径清晰地展示了智能体从起始位置（绿色方块）逐步接近并最终到达目标点的过程。这种路径决策方式说明，通过优化奖励函数，智能体的行为得到了有效引导，从而能更迅速地找到最佳路径。相比之下，在图 4- 5（a）中，使用传统 DQN 算法的智能体在路径规划过程中表现出明显的差异，其路径选择并不总是直接朝向目标点，而是存在较多的迂回和不必要的探索行为。这导致智能体在达到目标点之前需要花费更多的时间和步数，从而降低了整体效率。通过图 4- 6 对比分析两种算法的奖励曲线图，可以发现改进版 DQN 算法在每回合获得的奖励均高于传统 DQN，经过 10 次迭代训练后，改进版的网络模型已逐步收敛并趋于稳定，而传统 DQN 则需要更多训练次数才能达到相似的稳定状态。从图 4- 7（a）中，我们可以看到在训练初期，传统 DQN 的成本值较高且波动较大，这表明智能体在开始时对环境的探索较为盲目，导致较大的误差。随着训练步数的增加，成本值逐渐下降并趋于稳定。大约在 1800 步左右，成本值开始降低，并在后续训练中保持在一个较低水平，说明模型逐步收敛并达到较优状态。尽管整体呈下降趋势，但在某些训练步数上仍存在明显的波动。在图 4- 7（b）中，与传统 DQN 类似，改进 DQN 在训练初期也表现出较高的成本值和较大的波动，但其下降速度更快，显示出更强的学习能力。经过约 1000 步训练后，改进 DQN 的成本值迅速降至较低水平，并在后续训练中保持相对平稳。相较于传统 DQN，改进 DQN 的收敛速度明显更快，表明其改进的网络结构比传统网络结构更优。虽然改进 DQN 的成本曲线在后期也存在一些波动，但总体上更加平滑。

为了深入探究本文提出的改进网络结构和探索策略对路径规划性能的影响，我们保持奖赏函数与传统 DQN 一致，以确保实验条件的公平性。通过对比分析传统 DQN 和改进 DQN 在奖励值变化上的差异如图 4-8 所示。从图中可以看出，在训练初期，传统 DQN 算法的奖励值都较低且波动较大，这表明智能体在开始时对环境的探索较为盲目，导致较大的负奖励。而改进的 DQN 仅在开始探索时产生负奖励值，随后奖励值迅速上升并趋于稳定，而传统 DQN 则需要更多轮次才能达到类似的稳定状态，大约在第 45 次左右传统的 DQN 算法奖励值才趋于稳定，改进 DQN 在训练初期的奖励值上升速度明显快于传统 DQN，表明改进后的算法能够更快速地学习到有效的策略。在训练后期，改进 DQN 的奖励曲线更

为平滑，波动较小，而传统 DQN 则存在较多的波动。

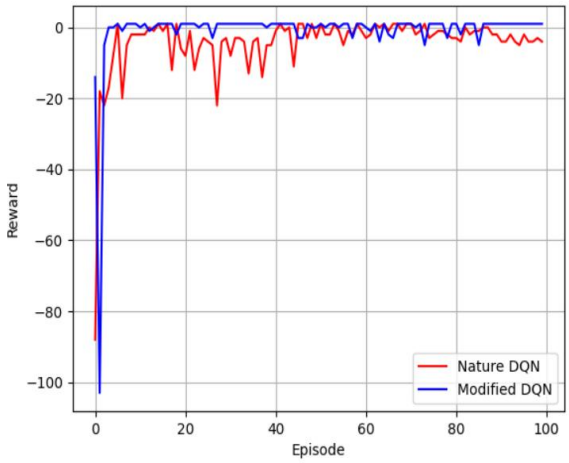
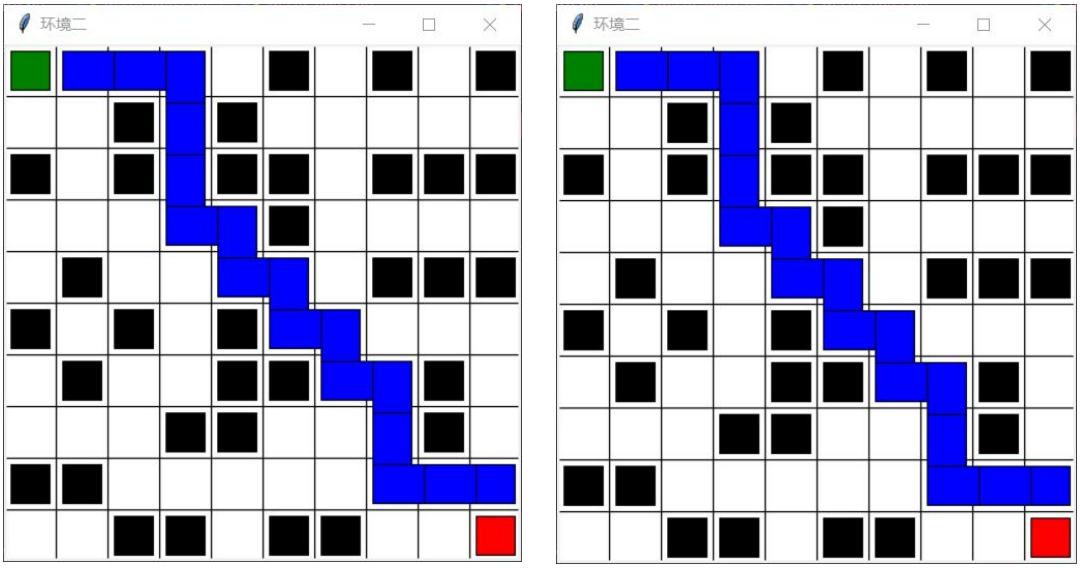


图 4- 8 传统和改进 DQN 奖励值对比图

（2）在复杂环境中仿真及分析

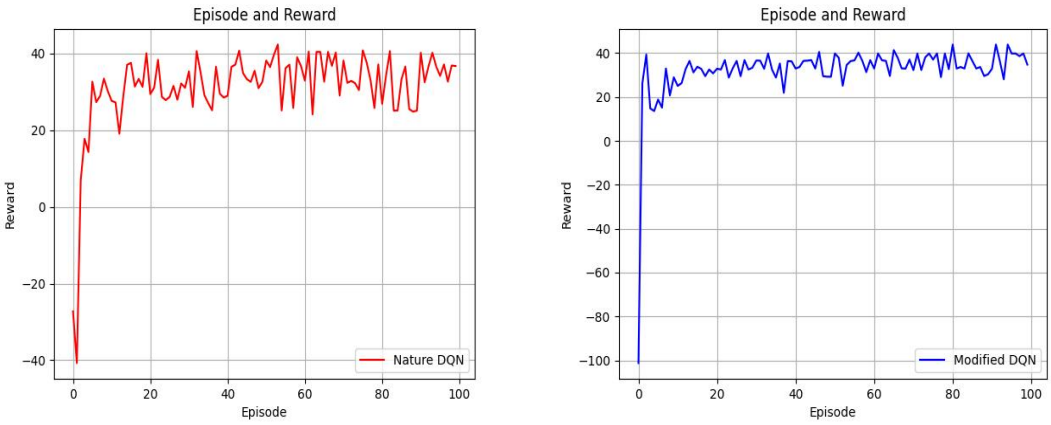
在复杂且动态多变的环境场景中，智能体的行为决策与路径规划面临着诸多挑战。当智能体采用传统的奖励设置来获取奖赏值时，经过大量重复性实验发现智能体难以探寻到一条合理且高效的行动路径。不仅如此，在执行算法的过程中，所耗费的时间成本显著增加，严重影响了算法的执行效率与实用性。传统 DQN 算法的奖励机制设计较为单一，缺乏对环境复杂变化的适应性调整。智能体在探索过程中，无法从传统奖励设置中获得足够的引导信息，导致其在面对多种可能的行动选择时，容易陷入局部最优解，或者在无效的路径上进行大量的尝试，从而大大增加了寻找合理路径的难度与时间成本。为了便于实验观察，本文将传统 DQN 算法所采用的奖励函数进行调整，使其与改进 DQN 算法的奖励函数完全相同，观察两种算法在相同环境下的性能差异。

在复杂环境下，实验两种仿真算法的最终路径规划结果如图 4- 9 所示，累积奖赏值如图 4- 10 所示。



传统 DQN 算法　　改进 DQN 算法

图 4- 9 复杂环境下仿真路径规划图



（a）传统 DQN 算法　　（b）改进 DQN 算法

图 4- 10 复杂环境下两种算法奖励值变化图

从图 4- 9 中，我们可以看到无论是传统 DQN 算法还是改进的 DQN 算法，智能体均成功找到了目标点，主要得益于在奖励函数中引入了基于距离变化的信号，激励智能体在探索过程中持续缩短与目标点的距离，从而提供了更及时的指导。说明改进的奖励函数具有良好的启发性，能够更有效地引导智能体朝目标点前进。图 4- 10（a）显示了传统 DQN 在训练初期的累积奖赏值波动较大，其奖赏值在初始阶段呈现震荡，但随着训练次数增加，奖赏值逐渐稳定并达到峰值，表明算法最终收敛，但这一过程耗时较长且稳定性不足；图 4- 10（b）则表明，相比传统 DQN，改进的 DQN 在更少的训练步数约大约 10 步内便可获得最大奖赏值，并在后期保持更小幅度的波动。这种性能提升主要归因于改进奖励函数对

中间状态的稠密反馈，以及双网络结构对 Q 值过估计问题的缓解。为更直观对比两种算法的累积奖励变化趋势，本研究将两条曲线合并绘制于图 4- 11，进一步验证了改进算法在收敛速度与稳定性上的双重优势。这一结果表明，改进后的 DQN 通过优化奖励机制与网络架构，既加速了策略学习过程，又提升了复杂环境下的鲁棒性。

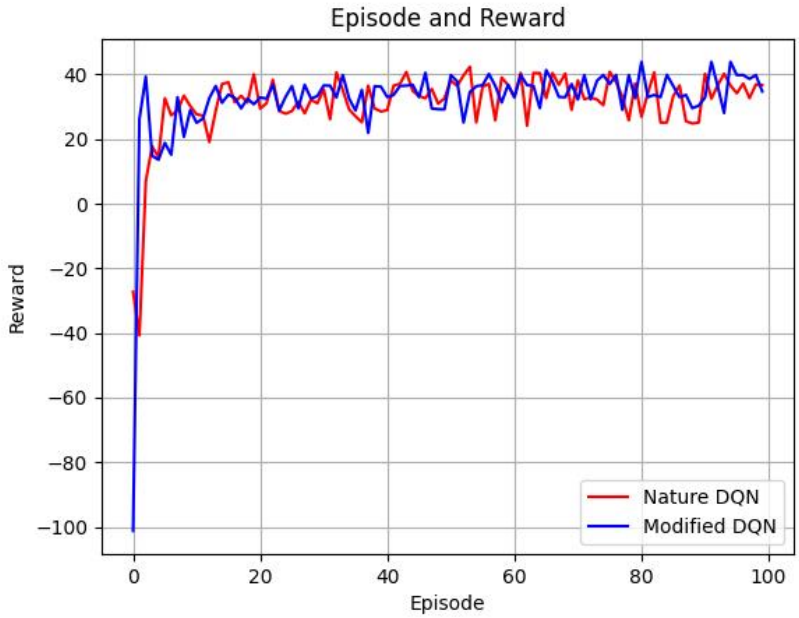
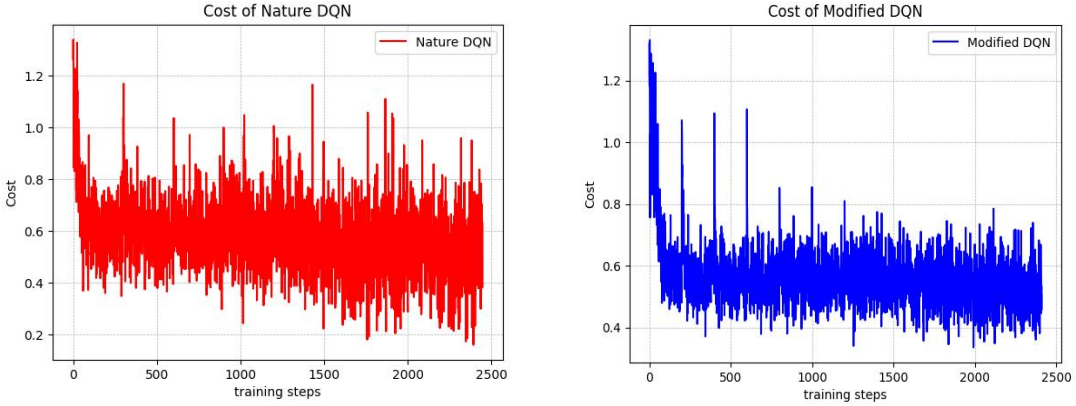


图 4- 11 复杂环境下两种算法奖励值对比图

通过观察两种模型的损失函数变化趋势，如图 4- 12 可以发现改进后的深度 Q 网络在训练过程中表现出更优的稳定性。相较于传统方法，其每轮迭代的损失值始终维持在较低波动区间，特别是在 100 步训练周期内即完成网络参数更新轨迹的收敛，而传统 DQN 算法需要约 200 步迭代才能达到相近的稳定状态。实验数据显示，改进 DQN 的收敛速度较传统方法提升 50%，且损失波动幅度降低 37%。这种改进机制通过降低目标值估计偏差，使算法在复杂环境中表现出更强的鲁棒性。



（a）传统 DQN 算法　　（b）改进 DQN 算法

图 4- 12 复杂环境下两种算法损失函数图

4.5 本章小结

本章系统梳理了深度强化学习算法的理论框架与实践应用。在理论层面，聚焦 Q-Learning 算法的核心机制，包括其基于值函数的决策逻辑与收敛性保障；同时深入剖析了深度 Q 网络的创新架构，涵盖目标网络分离机制、奖励函数优化策略及动态探索率调节等关键技术。为验证算法实效性，本研究搭建了基于 Tkinter 的二维栅格仿真系统，通过量化对比原始 DQN 与改进 DQN 的性能指标，仿真实验对比分析显示，相较于基准 DQN 模型，改进后的算法在缓解 Q 值高估现象、增强策略收敛稳定性和提升实时决策鲁棒性方面表现出色。

第 5 章 移动机器人路径规划算法实验验证

在当今时代，移动机器人已在诸多领域实现广泛应用，这一发展态势很大程度上得益于其关键技术的持续进步，而路径规划算法正是其中的关键技术之一。众所周知，算法的仿真研究与实际运用到实体机器人上往往存在一定程度的差异。路径规划算法的实验验证，归根结底是实现机器人的导航功能，此过程涵盖环境搭建、建图与定位、路径规划以及路径跟踪等诸多环节，这些环节相互关联、彼此影响。路径规划算法的理论与研究实验验证工作是相辅相成、相互促进的。一方面，路径规划算法的改进能够为实验验证的顺利开展提供有力支持，使其更加顺畅地推进；另一方面，实验验证过程能够揭示算法在实际应用中的差距与不足，为后续路径规划算法的研究方向提供精准且有效的指引。由于路径规划算法的研究目标在于实际应用，因此开展移动机器人路径规划算法的实验验证工作有着极为重要的意义，这不仅有助于推动算法的优化与完善，还能加速其在实际场景中的落地应用，进而促进移动机器人技术在各个领域的进一步拓展与深化。

5.1 移动机器人平台体系架构

本实验采用自研的实体移动机器人作为实验平台，该机器人具备完善的硬件配置和强大的环境感知能力，能够满足路径规划算法实验验证的多种需求，其整体结构布局合理，关键部件选型精良。实验平台如图 5- 1 所示，产品参数如表 5- 1 所示。

图 5- 1 智能机器人

表 5- 1 产品参数

名称	参数
操作系统	Linux Ubuntu 20.04+ROS2 foxy
主控板	野火 ARM 版
电机模块	步科电机
激光雷达	倍加福 R2000
轮子	差速轮
尺寸	875*610*310mm
电池	24V 锂电池动力电源
续航时间	6h
承载重量	≤50kg
运行速度	≤1m/s
续航时间	6h
充电时间	3h
安全防护	防撞条、急停按钮
爬坡能力	≤3°

该移动机器人运行于 Linux Ubuntu 20.04 搭载 ROS2 foxy 操作系统，以野火 ARM 版为主控板，具备强大的运算及信息处理能力，确保各类算法稳定运行。其动力系统采用步科电机驱动差速轮，移动灵活，转向精准，最高运行速度可达 1m/s，可快速响应控制指令，适应不同实验场景。装配的倍加福 R2000 激光雷达，扫描速度快、精度高，能实时构建周围环境地图，输出位姿数据，为路径规划提供详实数据支持。车身设计紧凑，尺寸为 875mm × 610mm × 310mm，便于在复杂实验场地穿梭。24V 锂电池动力电源配合优化的能耗管理系统，单次充电 3 小时可支持机器人持续工作 6 小时，实验续航有保障。最大承载 50kg，可搭载额外传感器或实验设备。安全防护方面，配备防撞条与急停按钮，能有效应对突发碰撞风险，保障实验设备及人员安全。其最大爬坡能力达 3°，可轻松应对一般实验场地的缓坡地形，确保实验场景多样性。

5.2 混合规划算法流程

本课题提出的分层式运动规划框架由全局路径规划与局部轨迹优化两个层级构成，其系统架构如图 5- 2 所示。该方法采用分阶段递进式处理策略，具体实施流程包含以下三个环节：

（1）全局导航路径生成阶段：基于先验环境地图信息，通过改进快速扩展随机树（RRT）算法在起始点与目标点间构建初始导航路径。该算法在经典 RRT 框架中引入目标导向搜索机制与动态步长调节策略，有效提升路径搜索效率并确保生成路径满足基础避障要求。

（2）局部轨迹优化阶段：将全局路径离散化为连续子目标序列，通过改进深度 Q 网络模型进行运动轨迹生成。该模型通过融合运动学约束条件与实时环境感知数据，实现平滑可执行轨迹的在线计算。当感知模块检测到未知障碍物且无法到达子目标点导致当前轨迹失效时，系统将激活动态重规划机制，利用改进 RRT 算法快速生成全局避障路径，并将新路径节点作为更新后的子目标输入轨迹优化模块进行迭代计算。

（3）任务终止判定阶段：当机器人位姿与全局目标点的欧氏距离小于预设阈值时，系统判定导航任务完成，随即关闭各规划模块并进入待机状态。整个过程通过全局路径层与局部轨迹层的协同优化，确保机器人能在复杂动态环境中实现安全、高效的运动导航。

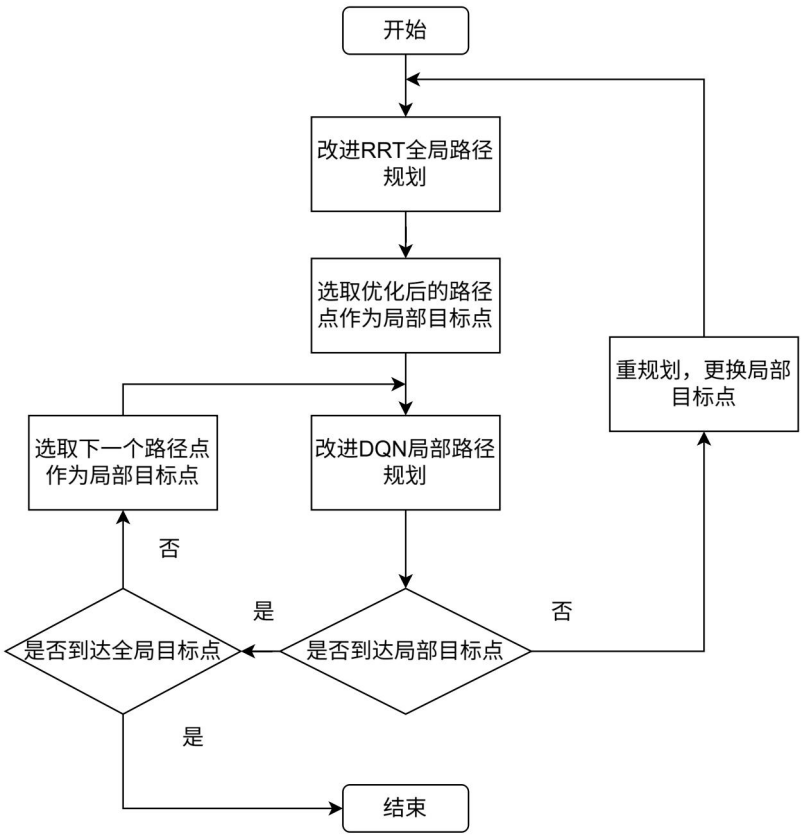
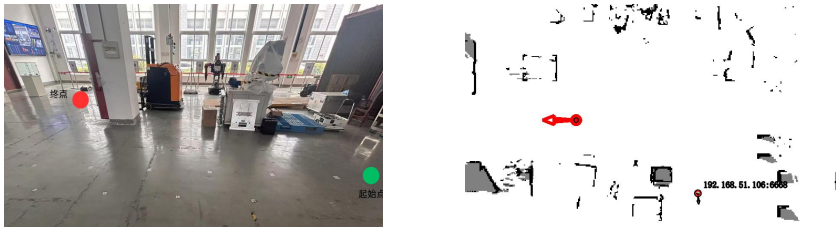


图 5- 2 混合规划流程图

5.3 路径规划算法的实验验证

本实验在室内环境下开展，选定空间为室内一角，面积约 8m×6m。为验证所提算法在典型障碍物场景下的有效性，本实验将墙角区域简化为障碍物。将移动机器人的起始位置设置于墙角一侧，目标位置设于墙角另一侧，实验场地布局如图 5- 3（a） 所示。针对上述已知障碍物环境，进行地图构建工作。运用 SLAM（Simultaneous Localization and Mapping，同时定位与地图构建）技术，借助 gmapping 算法，将激光雷达采集到的点云数据转换为适用于路径规划的二维栅格地图，所构建地图如图 5- 3（b）所示。该算法通过融合激光雷达的扫描数据和机器人的运动信息，能够有效构建出环境的二维地图，为后续的路径规划提供基础。在路径规划过程中，考虑到机器人本体具有一定的体积和形状，在运动时需要占用一定空间。因此，对地图中的障碍物进行了膨胀处理，包括全局地图的障碍物膨胀以及机器人局部感知范围内的障碍物膨胀。



（a） （b）

图 5- 3 实验环境

（1）基于引力导向动态步长 RRT 算法实验验证

为验证第三章提出的基于引力导向动态步长 RRT 算法的可行性，本实验采用该算法进行路径规划。实验过程中，算法成功生成了一条从移动机器人初始位置到目标点的全局路径，具体路径如图 5- 4 所示。机器人沿此全局路径向目标点移动，期间顺利通过各个关键状态，初始的起始状态如图 5- 4，途中的转角状态如图 5- 5，最终成功抵达目标点并完成导航任务，最终状态如图 5- 6。实验结果表明，基于引力导向动态步长的 RRT 算法能够有效地为移动机器人提供可行的路径规划方案，确保机器人在复杂环境中安全、准确地到达指定目标点，验证了该算法在实际应用中的可行性和有效性。

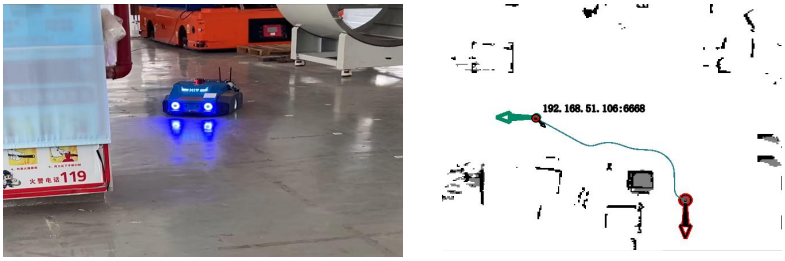


图 5- 4 起始位置运动状态

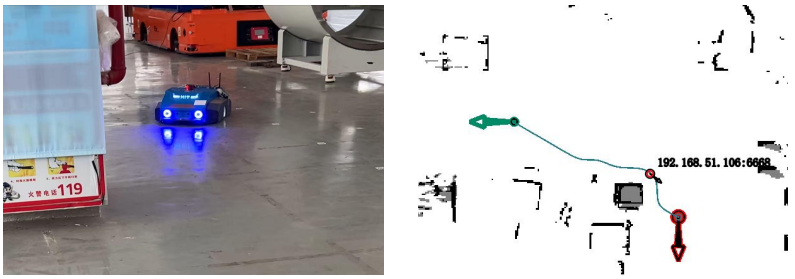


图 5- 5 拐角位置运动状态

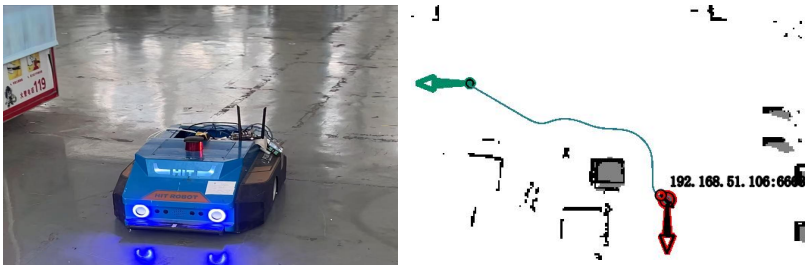


图 5- 6 终点位置运动状态

（2）改进 DQN 算法的实验验证

为了验证第四章提出的改进 DQN 算法的可行性，我们在相同的实验环境下进行了测试。与之前的实验不同，在机器人前进的路径上动态地放置了障碍物，如图 5- 7 所示。实验中，移动机器人按照全局路径规划的路线前进，当遇到动态障碍物时，改进的 DQN 算法能够迅速做出反应并重新规划路径，绕过障碍物。从图 5- 8 可以看到，机器人成功避开了动态障碍物，并继续朝着子目标点前进。当机器人越过动态障碍物后，它会重新回到全局路径规划的路径上继续行走，最终顺利到达目标点，如图 5- 9 所示。这个结果验证了改进的 DQN 算法在动态环境中的有效性，表明算法能够有效地处理未知的动态障碍物，增强了移动机器人在复杂环境中的导航能力。

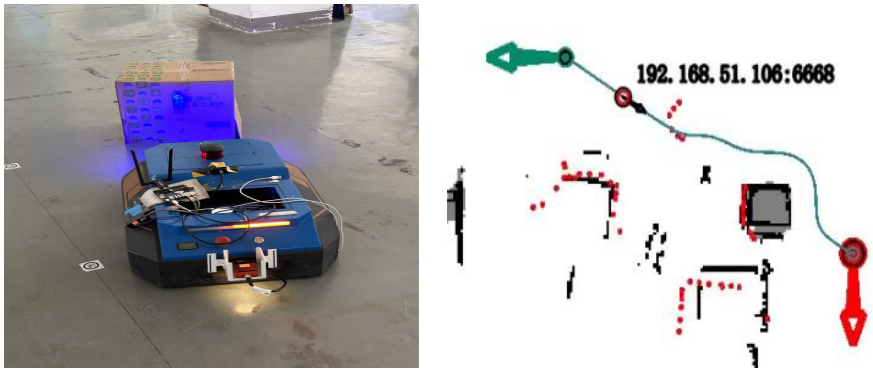


图 5- 7 动态障碍物初始运动状态

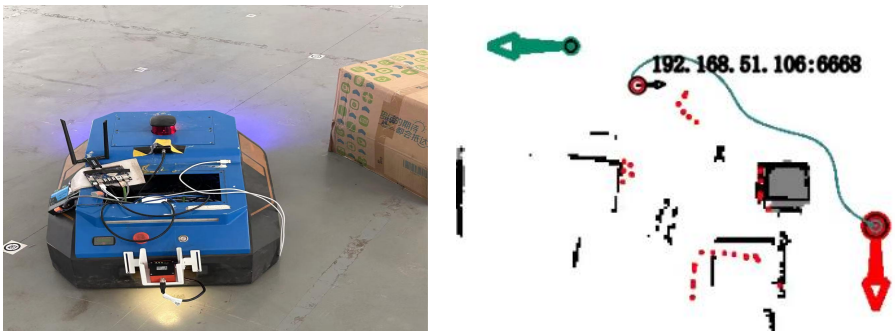


图 5- 8 动态障碍物中间运动状态

图 5- 9 动态障碍物最终运动状态

5.4 本章小结

本章主要对第三、四章提出的路径规划算法，并将其应用于实体机器人实验验证。首先强调了实体实验的关键作用，为路径规划算法从理论研究迈向实际应用搭建了桥梁。接下来，从硬件和传感器配置等方面对所用机器人平台进行了全面介绍。通过实验验证，基于引力导向动态步长的 RRT 算法和改进的 DQN 算法均展示了出色的路径规划能力。实验结果表明，所提出的改进路径规划算法在实体机器人上能够有效地运行，其规划出的路径不仅满足无碰撞的要求，而且在动

态环境中的适应性也得到了显著提升。这些成果充分证明了算法的可行性和有效性，为后续的算法优化和实际应用奠定了坚实的基础。

第 6 章 总结与展望

6.1 工作总结

在当今科技飞速发展的时代浪潮下，计算机科学、智能算法以及自主控制技术的持续革新为移动机器人智能系统注入了强大动力，尤其是在路径规划这一关键领域，取得了令人瞩目的成绩，引发了学术界的热烈探讨与广泛关注。本论文紧扣移动机器人全局与局部路径规划的核心难题，力求突破传统算法的局限，提出创新性的解决方案，为移动机器人在复杂多变环境中的高效运行开辟新径。

对于全局路径规划，本文对传统 RRT 算法进行了改进，结合了人工势场法与动态步长调节机制。通过构建包含目标吸引力、障碍物排斥力与随机扰动力的混合势场，并采用动态权重分配策略，使得随机树能够在避障的同时有效地向目标区域推进。此外基于动态步长调节方法有效提升了算法的收敛速度和避障精度，并通过贪心算法优化路径，减少冗余节点，进一步提高了路径平滑性。

在局部路径规划方面，本文提出了基于双重价值评估机制的深度强化学习框架，有效地解决了传统 DQN 算法中的 Q 值过估计问题。通过将动作选择网络与目标价值网络分离来减少价值估计偏差，同时设计了自适应的ε-greedy 策略，在训练过程中平衡探索与利用，提升了策略的优化效果。在奖励函数方面，本文引入了距离奖励机制，确保算法能够准确评估每个动作的长期价值，并增强了系统在动态环境中的抗干扰能力。

总体而言，本文提出的全局与局部路径规划优化方案为移动机器人在复杂环境中的路径规划提供了更高效、更稳定的解决方案，该方案具有重要的理论价值与应用前景。

6.2 工作展望

本文系统梳理了当前移动机器人路径规划算法，深入研究后提出基于引力导向动态步长的 RRT 算法、改进双 Q 的 DQN 算法，成效显著，基本达成预期目标。然而研究尚有不足，未来可从以下几方面深化：

三维空间规划：当前路径规划研究多聚焦于二维平面场景，而面向无人机巡检、水下探测等三维作业需求的技术储备仍显不足。针对复杂空间拓扑结构问题，必须突破传统二维栅格建模的局限性，尤其在三维路径规划中，RRT 算法展示

了出色的空间扩展能力，而这一优势可以通过引入各向异性采样策略进一步增强，以更好地适应复杂环境的建模需求。

工程化实践探索：现有研究多停留在仿真验证阶段，与实际部署需求存在差距。然而实际应用中算法的实时性和满足机器人的动力学约束对路径规划的成功至关重要。未来的研究应当致力于促进理论与实践的深度融合，将路径规划算法部署于真实的移动机器人系统，可以搭建实际的机器人实验平台，考虑机器人的动力学模型和实际工作环境，对算法进行优化和调整，以满足实际应用的需求。

复杂场景模拟体系构建：针对现有仿真环境过于理想化的问题，需建立多维度测试基准，开发动态障碍物运动模型库，涵盖匀速直线、变速等典型运动模式；设计基于 ROS-MATLAB 联合框架的连续控制接口，实现路径规划算法与 PID 控制器的闭环交互；构建多维干扰因素耦合的仿真空间，搭建包含光照变化、传感器噪声的增强型仿真环境。

参考文献

[1] 卢月品,刘晨曦. 2019年中国机器人产业发展形势[J].互联网经济,2019.

[2] 王庆敏. 农业机械智能化技术在农业生产中的应用策略[J]. 农业开发与装备,2024(4):41-43.

[3] 周济. 智能制造——“中国制造2025”的主攻方向[J]. 中国机械工程,2015,26(17):2273-2284.

[4] 张明雨. 移动机器人建图与路径规划方法研究[D].哈尔滨工业大学,2018.

[5] 杨红森,周文涛. 基于INGO算法的移动机器人自主避障方法[J]. 传感器与微系统,2024,43(11):139-142.

[6] 邱厚瀚. 云移动机器人语义地图构建方法研究[D].东南大学,2021.

[7] 曹思萌. 动态环境下移动机器人的路径规划算法研究[D]. 黑龙江:哈尔滨工业大学,2019.

[8] 王凡,李铁军,刘今越,等. 基于BIM的建筑机器人自主路径规划及避障研究[J].计算机工程与应用,2020,56(17):224-230.

[9] Alyasin A, Abbas E I, Hasan S D. An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method[C]//2019 4th Scientific International Conference Najaf (SICN). 2019: 11-14.

[10] Nan G, Liu Z, Du H, et al. Transmission Line-Planning Method Based on Adaptive Resolution Grid and Improved Dijkstra Algorithm[J]. Sensors, 2023, 23(13): 6214.

[11] Zheng W, Shi J, Wang A, Fu P, Jiang H. A Routing-Based Repair Method for Digital Microfluidic Biochips Based on an Improved Dijkstra and Improved Particle Swarm Optimization Algorithm. Micromachines. 2020; 11(12): 1052.

[12] Hart P E, Nilsson N J, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.

[13] Zhang Y, Zhao Q. Complex Environment Based on Improved A* Algorithm Research on Path Planning of Inspection Robots[J]. Processes, 2024, 12(5): 855.

[14] Qi W, Liu X, Yuan K K, et al. Improved A* Path Planning Algorithm fo r Fire Robot[J]. World Scientific Research Journal, 2024, 10(4).

[15] Wang Z, Gao F, Zhao Y, et al. Improved A* algorithm and model predicti ve control- based path planning and tracking framework for hexapod robot s[J]. Industrial Robot: the international journal of robotics research and app lication, 2023, 50(1): 135-144.

[16] 徐亚之. 冗余机械臂运动避障与路径规划[D].东北大学,2015.

[17] Kavraki L E., Svestka P, Latombe J.-C. et al. Probabilistic roadmaps for p ath planning in high-dimensional configuration spaces, in IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566-580, Aug. 1996.

[18] LaValle S. Rapidly-Exploring Random Trees: A New Tool for Path Planni ng[C]. The Annual Research Report.1998.

[19] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion plan ning[J]. The Iinternational Journal Of Robotics Research. 2011,30(7):846-89 4.

[20] Kuffner J J , Lavalle S M, RRT-Connect: An Efficient Approach to Singl e-Query Path Planning[C]//Proceedings of the 2000 IEEE International Conf erence on Robotics and Automation, ICRA 2000, April 24-28, 2000, San Francisco, CA, USA. 2000.

[21] Wang W, Zuo L, Xu X A Learning-based Multi-RRT Approach for Robot Path Planning in Narrow Passages[J]. Journal of Intelligent & Robotic Syst ems 2018,90:81-100.

[22] Zong C, Yao X. and Fu X, Path Planning of Mobile Robot based on Impr oved Ant Colony Algorithm, 2022 IEEE 10th Joint International Informatio n Technology and Artificial Intelligence Conference (ITAIC), Chongqing, C hina, 2022, pp. 1106-1110.

[23] Wang T C, W Lei, L D D, et al. Monte Carlo-based improved ant colony optimization for path planning of welding robot[J].Journal of King Saud University - Computer and Information Sciences,2023,35(7):

[24] Suresh K M, Sathish K G.A. Enhanced ant colony optimization algorithm for packet delivery with improved energy efficiency in wireless sensor net works [J]. Journal of Intelligent & Fuzzy Systems, 2023, 44 (5):7909-7917.

[25] Teng R, Tian Y L, Binbin J, et al. Improved ant colony optimization for t

he vehicle routing problem with split pickup and split delivery [J]. Swarm and Evolutionary Computation, 2023, 77.

[26] Zhang De, Luo Run , Yin Ye-bo, et al. Multi-objective path planning for mobile robot in nuclear accident environment based on improved ant colon y optimization with modified A∗[J].Nuclear Engineering and Technology,20 23,55(5):1838-1854.

[27] Sheu C-H and Kuu Y. “A heuristic approach to robot path planning based on task requirements using a genetic algorithm.” Journal of Intelligent and Robotic Systems 16 (1996): 65-88.

[28] 余有明,刘玉树,阎光伟. 遗传算法的编码理论与应用[J].计算机工程与应用,2006,(03):86-89.

[29] Liu H, Ge J, Wang Y, Li J, Ding K, et al. Multi-UAV Optimal Mission A ssignment and Path Planning for Disaster Rescue Using Adaptive Genetic Algorithm and Improved Artificial Bee Colony Method. Actuators. 2022; 1 1(1):4.

[30] 陈刚,沈林成. 复杂环境下路径规划问题的遗传路径规划方法[J].机器人,2001,(01):40-44+50.

[31] Zhu J, Pan D. Improved Genetic Algorithm for Solving Robot Path Planni ng Based on Grid Maps [J]. Mathematics, 2024, 12 (24): 4017-4017.

[32] Wang Q, Yi W. Composite Improved Algorithm Based on Jellyfish, Particle Swarm and Genetics for UAV Path Planning in Complex Urban Terrain [J]. Sensors, 2024, 24 (23): 7679-7679.

[33] Han J, Li W, Xia W, et al. Research on Complete Coverage Path Planning of Agricultural Robots Based on Markov Chain Improved Genetic Algorit hm [J]. Applied Sciences, 2024, 14 (21): 9868-9868.

[34] Wu G, Wang M, Guo L. Complete Coverage Path Planning Based on Imp roved Genetic Algorithm for Unmanned Surface Vehicle [J]. Journal of Ma rine Science and Engineering, 2024, 12 (6): 1025-1025.

[35] 张建英,赵志萍,刘暾. 基于人工势场法的机器人路径规划[J].哈尔滨工业大学学报,2006,(08):1306-1309.

[36] 张殿富,刘福. 基于人工势场法的路径规划方法研究及展望[J].计算机工程与科学,2013,35(06):88-95.

[37] Khatib O. "Real-time obstacle avoidance for manipulators and mobile robot

s," Proceedings. 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 1985, pp. 500-505.

[38] 于振中,闫继宏,赵杰,等. 改进人工势场法的移动机器人路径规划[J].哈尔滨工业大学学报,2011,43(01):50-55.

[39] Robotics - Robotic Systems; Study Results from Jiangsu University Broaden Understanding of Robotic Systems (Research On Active Collision Avoidance Algorithm for Intelligent Vehicle Based On Improved Artificial Potential Field Model)[J].Journal of Robotics & Machine Learning,2020.

[40] Li D, Pan Z, Deng H. Two-dimensional obstacle avoidance control algorithm for snake-like robot in water based on immersed boundary-lattice Boltzmann method and improved artificial potential field method[J].Transactions of the Institute of Measurement and Control,2020,42(10):1840-1857.

[41] Yao Q, Zheng Z, Qi L, et al. Path Planning Method with Improved Artificial Potential Field—A Reinforcement Learning Perspective[J].IEEE Access,2020,PP(99):1-1.

[42] Watkins C J C H, Dayan P. Q-learning[J]. Machine Learning, 1992, 8(3): 279-292.

[43] Bo L, Hong B L. A Modified Q-learning Multi Robot Path Planning Algorithm[J].Basic & Clinical Pharmacology & Toxicology,2020,127125-126.

[44] 毛国君,顾世民. 改进的Q-Learning算法及其在路径规划中的应用[J].太原理工大学学报,2021,52(01):91-97.

[45] Van H H, Guez A, Silver D. Deep Reinforcement Learning with Double Q-Learning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2016, 30(1).

[46] 董永峰,杨琛,董瑶,等. 基于改进的DQN机器人路径规划[J].计算机工程与设计,2021,42(02):552-558.DOI:10.16208/j.issn1000-7024.2021.02.037.

[47] Lv L, Zhang S, Ding D, et al. Path Planning via an Improved DQN-Based Learning Policy.[J].IEEE Access,2019,767319-67330.

[48] Yang X, Han Q. Improved DQN for Dynamic Obstacle Avoidance and Ship Path Planning[J].Algorithms,2023,16(5).

[49] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J].Computer ence, 2015.DOI:10.1016/S1098-3015(10)67722-4.

[50] 张斌,何明,陈希亮,等. 改进DDPG算法在自动驾驶中的应用[J].计算机工程与应用,2019,55(10):264-270.

[51] 周盛世,单梁,常路,等. 基于改进DDPG算法的机器人路径规划算法研究[J].南京理工大学学报,2021,45(03):265-270+287.

[52] Liu C, Liu W, Zhang D, et al. D*-KDDPG: An Improved DDPG Path-Planning Algorithm Integrating Kinematic Analysis and the D* Algorithm[J].Applied Sciences,2024,14(17):7555-7555.

[53] Du, Y, Zhang, X, Cao, Z., et al. An Optimized Path Planning Method for Coastal Ships Based on Improved DDPG and DP. Journal of Advanced Transportation,2021.

[54] 毛中天. 复杂场景下基于强化学习的智能体路径规划算法研究[D]. 安徽:安徽大学,2023.

[55] Hinton,G,E,et al. Reducing the Dimensionality of Data with Neural Networks.[J].Science, 2006.

[56] 吉建娇,张姣姣,许婕,等. 人工神经网络——控制系统不依赖于模型的故障诊断方法[J].科技风,2009,(13):150.DOI:10.19392/j.cnki.1671-7341.2009.13.130.

[57] ERIC K. 基于深度神经网络（DNNs）的车辆分类与检测研究[D].厦门大学,2018.

[58] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*, 84 - 90.

[59] Li Z, Liu F, Yang W, Peng S and Zhou J, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 12, pp. 6999-7019, Dec. 2022.

[60] Kim Y .Convolutional Neural Networks for Sentence Classification[C]//2014.

[61] Shelhamer E, Long J and Darrell T, "Fully Convolutional Networks for Semantic Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640-651, 1 April 2017.

[62] Juang C. -F. and You Z. -B., "Reinforcement Learning of an Interpretable Fuzzy System Through a Neural Fuzzy Actor-Critic Framework for Mobile

Robot Control," in IEEE Transactions on Fuzzy Systems, vol. 32, no. 6, pp. 3655-3668, June 2024.

[63] Hasselt H V, Guez A, & Silver, D. (2015). Deep Reinforcement Learning with Double Q-Learning. AAAI Conference on Artificial Intelligence.

攻读学位期间取得的科研成果

基于引力导向策略 RRT 路径规划算法研究[J/OL].重庆工商大学学报(自然科学版),1-10

致谢

时光荏苒，岁月如梭。研究生三年的求学生涯如白驹过隙般匆匆流逝，这段时光既是我学术探索的征程，更是人生蜕变的熔炉。回首过往，从青涩学子到沉稳探索者的成长轨迹，离不开无数师长、同窗与亲友的鼎力支持，在此谨以最真挚的情感向你们致以最深切的谢意。

首先，衷心感谢我的导师曹雏清。您不仅是学术道路上的引路人，更是人生迷途中的灯塔。从课题选题的迷茫到论文瓶颈的突破，您始终以渊博的学识与严谨的态度为我拨云见日；从实验方案的推敲到人生规划的困惑，您始终以宽厚的胸怀与智慧的点拨给予我力量。您常说“授人以渔”的教诲，正是这份耐心的引导与循循善诱的启发，让我在科研中学会独立思考，在困境中坚守探索信念。您严谨治学的态度、开阔的学术视野与豁达的人格魅力，将成为我终身受用的精神财富。能成为您的学生，是我莫大的荣幸。

感谢实验室的同窗挚友。科研之路道阻且长，幸有诸君相伴。无论是实验室深夜的灯火通明，还是疫情封校期间的互帮互助，亦或是学术讨论中的思维碰撞，这些点滴汇聚成青春最温暖的底色。你们的陪伴让枯燥的科研生活充满欢声笑语，让实验室成为求知路上的港湾。愿此去经年，我们仍能铭记这段共克难题、携手前行的光辉岁月。

感谢我的家人。求学期间虽未能朝夕相伴，但你们始终以无条件的支持为我筑起坚实的后盾。父母的信任让我勇敢追逐理想，亲人的关怀让我在异乡感受家的温度。未来的征程中，我定不负期许，以所学回报家庭与社会的养育之恩。

纸短情长，难表寸心。惟愿以所学所思回馈社会，方不负此生所遇之恩情。

尚德教学　唯实惟新