

复杂动态环境中移动机器人双层路径规划方法

胡耀炜¹, 汤萍萍^{1,2,3†}, 张 晖², 陈旭飞¹

(1. 安徽师范大学 物理与电子信息学院, 安徽 芜湖 241002;

2. 南京邮电大学 通信与信息工程学院, 南京 210003;

3. 安徽华东光电技术研究所有限公司, 安徽 芜湖 241002)

摘要: 路径规划是机器人实现自主操作的核心技术之一, 使得机器人在复杂动态环境中能够快速找到安全路径. 然而, 仅依赖全局路径规划无法有效避开动态障碍物, 仅使用局部路径规划则可能会陷入局部极小值导致无法顺利到达目标. 鉴于此, 提出一种适用于动态环境的移动机器人双层路径规划方法. 在第 1 层中: 采用并行采样和双树引导快速探索随机树算法 (PB-RRT) 搜索全局路径. 该层利用并行启发式采样策略, 建立包含距离因子和转角因子的评价函数, 以选取最佳采样节点进行自适应扩展, 构建双树引导机制实现两树快速融合. 然后结合轨迹优化方法, 对初始路径长度和平滑度进行优化, 得到路径关键节点. 在第 2 层中: 结合第 1 层得到的路径关键节点作为安全动态窗口算法 (SDWA) 动态子目标, 并引入一种偏置引导方法以避开动态障碍物. 实验结果表明, 与其他规划算法相比, 所提出方法能够快速生成更高质量的全局路径. 在动态环境中能够高效避开动态障碍物, 验证了所提出方法的有效性.

关键词: 快速探索随机树; 多点并行; 双树引导; 轨迹优化; 偏置引导; 动态路径规划

中图分类号: TP242 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2025.0170

引用格式: 胡耀炜, 汤萍萍, 张晖, 等. 复杂动态环境中移动机器人双层路径规划方法 [J]. 控制与决策, 2025, 40(10): 2995-3004.

Two-layer path planning method for mobile robots in complex and dynamic environments

HU Yao-wei¹, TANG Ping-ping^{1,2,3†}, ZHANG Hui², CHEN Xu-fei¹

(1. College of Physics and Electronic Information, Anhui Normal University, Wuhu 241002, China; 2. College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 3. Anhui Huadong Photoelectric Technology Research Institute Co. Ltd., Wuhu 241002, China)

Abstract: Path planning is one of the critical technologies for robots to achieve autonomous operation, enabling to quickly find the safe path in complex dynamic environments. However, relying on global path planning alone cannot avoid dynamic obstacles, while only using local path planning may lead to falling into local minima and fail to reach the target. Therefore, a two-layer mobile robot navigation method suitable for complex and dynamic environments is proposed. In the first layer, a parallel sampling and bidirectional guidance rapidly-exploring random tree algorithm (PB-RRT) is employed to search the global path. This layer introduces a parallel heuristic sampling strategy and designs an evaluation function incorporating distance and angle factors to select the optimal sampling nodes for adaptive expansion. A bidirectional guidance mechanism is established to accelerate the merging process of the two trees. Then, combined with trajectory optimization methods, the initial path length and smoothness are optimized to obtain the key nodes of the path. In the second layer, the key nodes of the path obtained from the first layer are used as dynamic subtargets for the safe dynamic window approach (SDWA), which introduces an offset guidance method to avoid dynamic obstacles. Extensive experimental results indicate that the proposed method can effectively plan an optimal

收稿日期: 2025-02-19; **录用日期:** 2025-06-13.

基金项目: 国家自然科学基金项目 (62071005, 62071252); 国家重点研发计划项目 (2020YFB2104004); 江苏省重点研发计划项目 (BE2021725); 安徽省自然科学基金项目 (2308085Y02, 2208085MF155); 安徽省高校自然科学基金项目 (KJ2021A0124); 安徽省博士后科研项目 (2024C946); 安徽省育人质量工程项目 (2024cxcysj027).

†通信作者. E-mail: tpping@ahnu.edu.cn.

global path compared with other baseline algorithms. The ability to efficiently avoid dynamic obstacles in a dynamic environment verifies the feasibility of the proposed method.

Keywords: rapidly-exploring random tree (RRT); parallel sampling; bidirectional guidance; trajectory optimization; offset guidance; dynamic path planning

0 引言

近年来, 移动机器人在农业、工业和应急救援等领域发挥着重要作用^[1-3]. 路径规划作为机器人领域的关键技术和主要挑战之一, 得到了广泛研究^[2]. 路径规划可分为全局路径规划和局部路径规划, 实时路径规划和避障是实现机器人自主导航的基本前提^[3]. 复杂动态环境是指同时存在多种静态障碍和动态障碍物、结构布局不规则、环境信息部分未知且具有动态变化特性的场景. 因此, 在复杂动态环境中, 确保机器人安全高效行驶是亟待解决的重要问题^[4].

全局路径规划常用方法之一是基于采样的规划算法, 如快速探索随机树 (rapidly-exploring random tree, RRT) 算法^[5]. 因其较好的全局搜索能力和较低的占用内存, 在复杂环境路径规划中得到了广泛应用^[6]. 但是, 采样的随机性会导致生成路径效率低以及规划路径曲折等问题, 为此, 研究学者做了大量优化改进. Kuffner 等^[7]提出了 Bi-RRT (bidirectional rapidly-exploring random tree) 算法, 从起点和终点分别生长两棵随机树, 减少了路径规划时间; 陈丹等^[8]通过在非障碍物空间生成随机采样点, 降低了算法的碰撞检测概率, 并有效减少了冗余节点的产生; Liu 等^[9]将目标约束采样和偏置扩展 RRT 与 Bi-RRT 相结合, 减少了无效扩展和搜索时间; Jiang 等^[10]用混合约束采样代替随机采样, 使得采样节点分布更接近障碍物间的可通过区域, 从而减少了复杂环境下的无效采样; Hu 等^[11]提出了路径变形策略, 使得路径点远离碰撞点, 在不发生碰撞的情况下生成了轨迹, 结合转向函数进行路径平滑, 快速规划出满足机器人动力学约束的平滑路径; Wang 等^[12]引入了以动态节点为目标的贪婪扩张策略, 在随机采样过程中加入了初始节点排斥机制和采样空间限制, 有效减少了反向采样, 提高了采样速度和效率.

全局路径规划可搜索出从起点到终点的全局路径, 但是无法避开局部动态障碍物^[13]. 在实际应用中, 机器人通常处于动态环境中, 而动态环境的不确定性使得机器人易与障碍物发生碰撞. 因此, 亟需研究更加高效且具有实时响应能力的路径规划方法. Qi 等^[14]在已知图的基础上生成了初始路径和状态树, 并提出了一种帕累托优势路径重规划方法, 在状态树上搜索另一个节点作为替代节点以避开未知障碍物, 但是未考虑动态未知障碍物; Dobrevski 等^[15]提出了结合动态窗口和深度强化学习的自适应动态窗口方法, 使用神经网络预测成本函数中的参数来适应特定环境, 但是需要大量样本训练网络; Liu 等^[16]提出了改进蚁群算法来提升全局规划的效率, 并结合由能耗评估与动态避障评估子函数优化的动态窗口方法实现了全局路径规划与局部避障的协同优化, 但是, 蚁群算法在处理较大规模的环境时, 可能导致计算时间较长, 而在复杂动态环境中单纯依靠动态窗口法, 实时避障效率可能不高; Yasuda 等^[17]改进了动态窗口方法, 通过使用确定性采样来解决具有随机动力学的差动移动机器人在不确定性环境中的规划问题, 但是缺乏全局路径引导, 在分辨率较高的地图中可能会迷失方向.

针对全局路径规划不能适应动态环境, 局部路径规划缺乏方向引导, 在复杂环境中易陷入局部最优等问题, 本文提出移动机器人双层路径规划方法, 将动态路径规划问题划分为快速全局路径规划层和导航过程中的高效局部动态避障层. 在第 1 层中: 采用 PB-RRT 算法快速生成全局路径, 轨迹优化后提取路径关键节点作为局部路径规划的动态子目标; 第 2 层结合 SDWA 算法和偏置引导方法进行局部路径规划, 避开未知动态障碍物. 双层方法框架如图 1 所示.

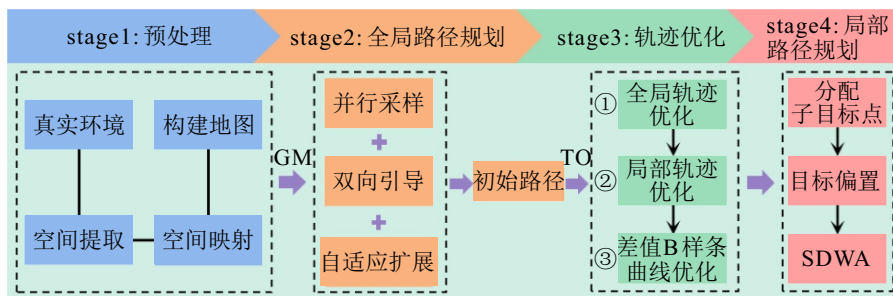
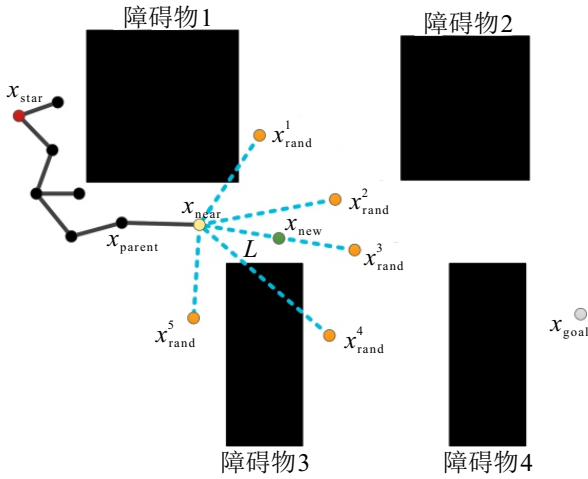


图1 双层方法框架

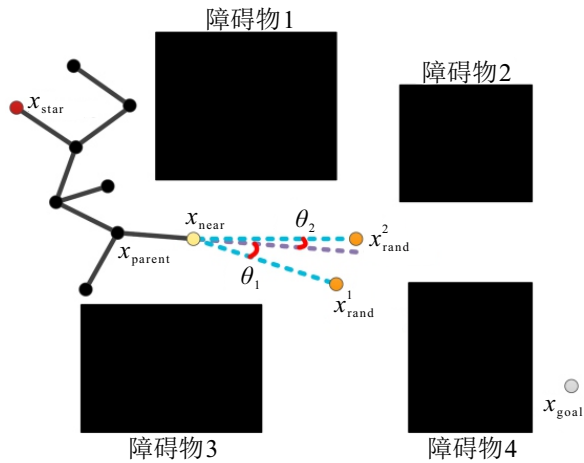
1 算法设计

1.1 PB-RRT 算法

由于随机性采样, 传统 RRT 算法向各方向扩展节点, 严重影响了算法的规划效率, 为此, 本文提出了并行启发式采样策略. 该策略在每次迭代期间随机采样多个节点, 并舍弃落入障碍物内的采样点, 使用评价函数来评估剩余采样点, 以识别最佳节点. 图 2(a) 为随机并行采样 5 个节点, 计算每个采样点的评价函数, 选择具有最低评估值的节点用于扩展.



(a) 并行采样



(b) 扩展节点选择

图2 并行采样

评价函数 $F(x)$ 计算采样点到起始点和目标点的路径长度, 引导随机树更快地向目标点生长. 对于当前并行采样的候选节点 x^i_{rand} 、最近节点 x_{near} 以及最近节点的父节点 x_{parent} , 计算 3 个路径节点间的角度, 最大限度地减少采样期间产生的不必要转弯, 从而降低路径成本提高路径质量. 由于距离和角度两个度量属于不同的维度, 需要进行归一化处理, 将不同数据源统一至一个参照系中, 归一化方法如下所示:

$$T_n = \frac{\theta_{x_{parent}, x_{near}, x^i_{rand}}}{\sum_{i=1}^n \theta_{x_{parent}, x_{near}, x^i_{rand}}}, \quad (1)$$

$$\theta_{x_{parent}, x_{near}, x^i_{rand}} = \arccos(\mathbf{N}_1, \mathbf{N}_2) = \frac{\mathbf{N}_1 \cdot \mathbf{N}_2}{\|\mathbf{N}_1\| \cdot \|\mathbf{N}_2\|}, \quad (2)$$

$$A(x^i_{rand}, x_{start}) = \frac{d_{x^i_{rand}, x_{start}}}{\sum_{i=1}^n d_{x^i_{rand}, x_{start}}}, \quad (3)$$

$$N(x^i_{rand}, x_{goal}) = \frac{D_{x^i_{rand}, x_{goal}}}{\sum_{i=1}^n D_{x^i_{rand}, x_{goal}}}. \quad (4)$$

其中: T_n 为路径角度归一化函数; $\theta_{x_{parent}, x_{near}, x^i_{rand}}$ 为最近节点的父节点 x_{parent} 、候选节点最近节点 x_{near} 以及并行采样候选节点 x^i_{rand} 间构成路径段的角度的值, 其取值范围为 $(0, \pi)$; \mathbf{N}_1 为从节点 x_{parent} 到节点 x_{near} 的向量; \mathbf{N}_2 为节点 x_{near} 到节点 x^i_{rand} 的向量; $A(x^i_{rand}, x_{start})$ 和 $N(x^i_{rand}, x_{goal})$ 为节点距离归一化函数; $d_{x^i_{rand}, x_{start}}$ 为 x^i_{rand} 与 x_{start} 两点间的欧氏距离; $D_{x^i_{rand}, x_{goal}}$ 为 x^i_{rand} 与 x_{goal} 两点间的欧氏距离. 选取最佳采样节点时, 利用自适应因子 α 和 k 调节距离和角度对于评价函数的影响, 提升采样点的质量, 以此为基础构建如下评价函数:

$$F(x) = \alpha \cdot A(x) + \left(1 + \frac{d_n}{D_n}\right) \cdot N(x) + k \cdot T_n, \quad (5)$$

$$A(x) = \|x^i_{rand}, x_{start}\|, \quad (6)$$

$$N(x) = \|x_{goal}, x^i_{rand}\|. \quad (7)$$

这里: D_n 为起始点到目标点的欧氏距离, d_n 为当前节点到目标点的欧氏距离, α 为距离因子, k 为转角因子. 图 2(b) 中, x^2_{rand} 与 x_{goal} 、 x_{start} 间的欧氏距离比 x^1_{rand} 与 x_{goal} 、 x_{start} 间的欧氏距离长, 但是, 相邻路径段间的夹角 $\theta_2 < \theta_1$, 因此, 综合距离和角度的成本代价, 选择在 x^2_{rand} 与 x_{near} 间生成路径. 若最近节点向最佳采样点扩展过程中与障碍物发生碰撞, 则其评价价值将被设为无穷大, 即从候选采样点集中剔除该采样点, 并从候选点集中继续选择最佳采样点进行扩展.

传统 RRT 算法随机树以固定步长进行扩展. 然而, 扩展步长对于规划效率和路径质量有很大影响. 在简单区域中, 使用较大的步长加速空间探索; 而在复杂区域中, 较小的步长可减少冗余节点, 尤其是当可行区域非常狭窄时, 这种优势会更加明显. 因此, 采用势场梯度来自适应地调整扩展步长. 确定最近节点 $x_{nearest}$ 使用排斥势扩展到新节点 x_{new} , 排斥势为

$$U_d = \begin{cases} \frac{1}{2}\eta_d \left(\frac{1}{d_o} - \frac{1}{d^*} \right)^2, & d_o \leq d^*; \\ 0, & \text{otherwise;} \end{cases} \quad (8)$$

$$d_o = \|x_{\text{nearest}}, x_{\text{obs}}\|. \quad (9)$$

其中: η_d 为决定势场强度的正常数, d^* 为障碍物影响半径. 势场函数梯度 $\nabla U_d \in R^n$ 可由下式计算, 势场梯度决定随机树扩展步长:

$$\nabla U_d = \begin{cases} \eta_d \left(\frac{1}{d_o} - \frac{1}{d^*} \right) \frac{1}{d_o^2} \nabla d_o, & d_o \leq d^*; \\ 0, & \text{otherwise;} \end{cases} \quad (10)$$

$$L_{\text{step}} = L_0 - \beta \|\nabla U_d\|. \quad (11)$$

这里: L_0 为在势场影响范围外时使用的恒定扩展步长; $\beta > 0$ 为调节因子, 用于调节势场梯度对于扩展步长的影响程度.

与传统的双向快速探索随机树法 (bidirectional rapidly-exploring random tree, Bi-RRT) 随机树生长方式不同, 本文构建了双树引导机制. 在起始点和目标点同时建立两棵随机探索树, 起点树向目标点生长并为终点树提供启发式信息, 引导终点树向起点树生长, 从而实现两棵树快速融合. 如图3所示: 起点树新节点 x_{new_1} 是通过并行启发式采样策略生成的, 而终点树新节点 x_{new_2} 是将起点树新生成的路径节点 x_{new_1} 作为局部目标点通过并行启发式采样策略生成的. 终点树选取最佳扩展节点的评价函数为

$$F(x) = \varepsilon \cdot B(x) + \left(1 + \frac{l_n}{L_n}\right) \cdot S(x) + m \cdot T_n, \quad (12)$$

$$B(x) = \|x_{\text{rand}}^i, x_{\text{goal}}\|, \quad (13)$$

$$S(x) = \|x_{\text{new}}^{T1}, x_{\text{rand}}^i\|. \quad (14)$$

其中: L_n 为终点到 x_{new}^{T1} 的欧氏距离, l_n 为当前节点到 x_{new}^{T1} 的欧氏距离, ε 为距离因子, m 为转角因子. 图3中: 紫色虚线为传统 Bi-RRT 算法随机树扩展方向, 绿色虚线为采用双树引导机制的随机树扩展方向. 将终点树扩展目标从起点改为起点树新生成的节点, 可以更快地完成双树连接.

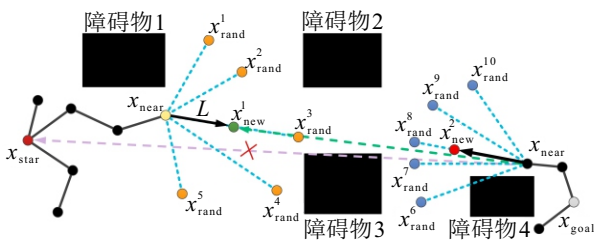


图3 双树引导

1.2 轨迹优化

改进后的算法虽然提高了规划效率, 但是, 规划

的路径仍然存在冗余节点, 路径曲率不连续. 因此, 通过轨迹优化策略滤除初始路径非关键节点, 提升路径质量. 首先, 以终点为线段的起始端点回溯初始路径, 依次连接后续路径节点, 检查连接的路径段是否与障碍物发生碰撞. 若没有发生碰撞, 则删除两个节点间所有的路径节点; 反之, 则将该节点的子节点作为新的线段起始端点, 重复执行上述操作, 回溯至起点结束. 全局重规划过程如图4(a)所示. 其中: 黑色路径为算法规划出的初始路径, 绿色路径段为全局重布线后的路径. x_{goal} 与 x_7 、 x_6 与 x_3 、 x_3 与 x_{start} 间连接未与障碍物发生碰撞, 因此, x_{10} 、 x_9 、 x_8 、 x_5 、 x_4 、 x_2 以及 x_1 为冗余节点, 可直接删除.

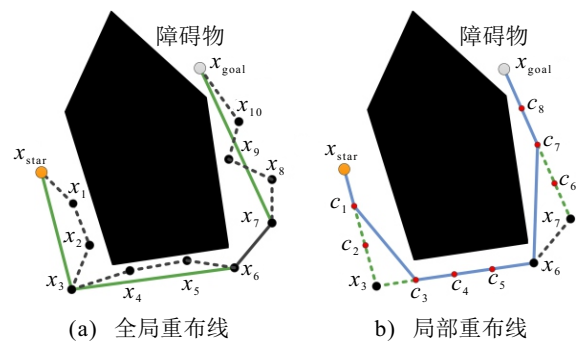


图4 重布线轨迹优化

经全局重规划后路径节点显著减少, 但是, 仍然由初始路径的部分节点连接构成, 采用局部路径重规划进一步提升路径质量. 由于全局重规划后路径节点比较稀疏, 对新生成的路径段进行插值处理, 得到节点较密集的路径进行局部重规划. 从起点开始, 依次连接第3个节点与后续路径点, 判断连接的路径段是否与障碍物发生碰撞. 若与障碍物没有发生碰撞, 则删除中间节点和相邻路径段; 反之, 则以该节点的父节点为线段的起始端点再次执行上述操作, 直至到达目标点. 局部路径重规划过程如图4(b)所示. 其中: 红色节点为算法插入的路径节点, 蓝色路径为局部重规划后得到的路径.

针对路径拐点处不平滑现象, 利用 B 样条曲线对路径进行拟合优化. 但是, 局部重布线优化后路径节点数量较少, 直接使用样条曲线平滑会导致路径拟合程度低, 且极易与障碍物发生碰撞. 针对这一问题, 本文在相邻节点间构建了控制点, 通过设置阈值选择构建控制点方式, 判断相邻两个节点间路径段长度是否小于阈值. 若小于阈值, 则在路径段中点构建控制点; 反之, 则以两个节点为起始点分别取正常步长长度构建控制点. 利用 B 样条曲线对构建控制点的路径进行拟合, 生成满足机器人最大曲率约束且拟合度高的平滑路径. k 次 B 样条曲线可定义为

$$C(x) = \sum_{i=1}^{n+1} N_{i,k}(x)D_i. \quad (15)$$

其中: $D_i (i = 1, 2, \dots, n + 1)$ 为控制点; $N_{i,k}(x)$ 为 k 次 B 样条基函数, 可由 DeBoor-Cox 递归公式得到, 方法如下所示:

$$N_{i,1}(x) = \begin{cases} 1, & x_i < x < x_{i+1}; \\ 0, & \text{otherwise}; \end{cases} \quad (16)$$

$$N_{i,k}(x) = N_{i,k-1}(x) \frac{x - x_i}{x_{i+k-1} - x_i} + N_{i+1,k-1}(x) \frac{x_{i+k} - x}{x_{i+k} - x_{i+1}}. \quad (17)$$

这里: x_i 为路径节点集 $\{x_1, x_2, \dots, x_{m+1}\}$ 中第 i 个节点, $m = i + k$. 样条优化过程如图 5 所示. 其中: 红色路径为直接使用样条曲线优化生成的路径, 黑色节点为新增的控制点, 绿色路径为构建控制点后使用样条曲线优化生成的路径.

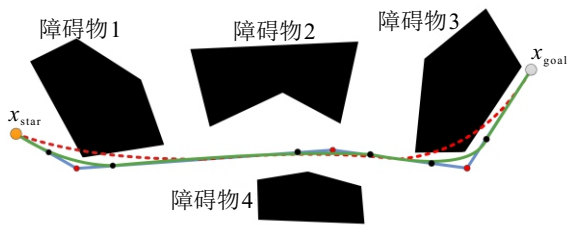


图5 B 样条曲线优化

1.3 复杂度分析

算法复杂度取决于随机采样、最近节点搜索和碰撞检测. 并行采样策略每次迭代采样多个节点计算评价函数值, 其时间复杂度为 $O(n)$. 动态扩展步长方法的时间成本取决于最近节点搜索和碰撞检测, 因此复杂度为 $O(n \log(n))$. 对于双树引导机制, 起点树和终点树的采样和扩展方法均与单树结构相同, 因此, 规划初始路径阶段的时间复杂度为 $O(n \log(n))$. 在轨迹优化时, 重新回溯路径节点并检测连接, 插入控制点 B 样条平滑的时间复杂度均小于或等于 $O(n)$. 因此, 该算法的时间复杂度为 $O(n \log(n))$. 在空间复杂度方面, 存储随机树节点需要 $O(n)$ 空间, 而每个新节点存储附近节点需要 $O(n \log(n))$ 空间. 因此, 该算法的空间复杂度为 $O(n \log(n))$. 相对于 RRT 算法具有相同的复杂度. 根据实验验证可知, 相比于其他算法, PB-RRT 算法可以实现更快的收敛速度和更优的路径解, 且存储的路径节点最少.

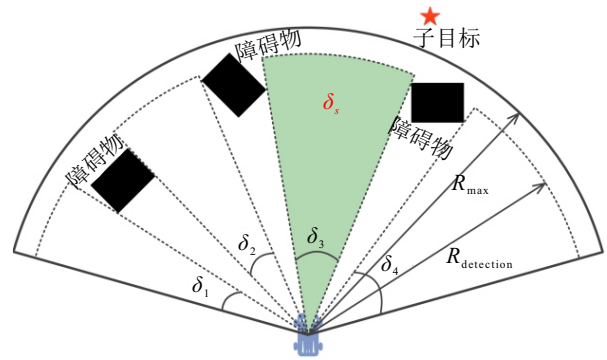
1.4 SDWA 算法

传统动态窗口法速度窗口保守, 评价函数机制

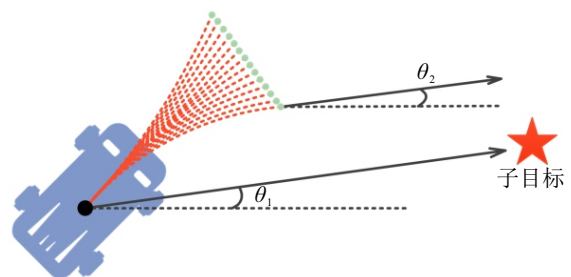
不合理, 导致机器人在复杂环境中易陷入局部最优. 本文提出了 SDWA 算法, 机器人在行驶过程中, 根据当前运动的线速度和角速度约束、加减速度约束以及制动距离约束, 计算在固定时间 Δt 窗口内可达的线速度与角速度区间, 形成动态速度窗口. 然后, 在该可行速度区间内对线速度 v 和角速度 ω 进行离散采样, 生成一系列速度对 (v_i, ω_i) . 对于每个速度对, 利用机器人运动学模型预测未来一段时间的轨迹. 本文通过自适应动态窗口动态调节探索窗口大小, 如图 6(a) 所示: 得到所有可行区域 δ , 确定距离动态子目标最近的预测轨迹区域 δ_s 为最优可行区域. 在最优可行区域内, 机器人通过轨迹评价函数中不同评价因素共同调节来确定最优行进方向. 轨迹评价函数为

$$G(v, \omega) = k_1 \cdot \text{Head}(v, \omega) + k_2 \cdot \text{Dist}(v, \omega) + k_3 \cdot \text{Vel}(v, \omega) + k_4 \cdot \text{Dist}_{\text{GP}}(v, \omega) + k_5 \cdot \text{Subt}(v, \omega). \quad (18)$$

其中: $\text{Head}(v, \omega)$ 为航向评价函数, $\text{Dist}(v, \omega)$ 为障碍物威胁评价函数, $\text{Vel}(v, \omega)$ 为速度评价函数, $\text{Dist}_{\text{GP}}(v, \omega)$ 为全局路径偏离函数, $\text{Subt}(v, \omega)$ 为终端收敛函数, $k_1 \sim k_5$ 为权重系数.



(a) 轨迹窗口选择



(b) 航向角偏差计算

图6 安全动态窗口

改进优化航向评价函数, 如图 6(b) 所示, 综合机器人当前位置和预测轨迹端点相对于动态子目标点的角度表示为 $|\theta_2 - \theta_1|$, 由此预测轨迹可获得更高的航向评估值, 从而使得机器人尽可能快地转向理想

航向. 对障碍物威胁评价函数进行细分, 划分为机器人与静态障碍物和动态障碍物的距离, 使得机器人行驶路径更加安全. 障碍物威胁评价函数如下所示:

$$\text{Dist}(v, \omega) = \lambda \cdot \text{Dist}_S(v, \omega) + (1 - \lambda) \cdot \text{Dist}_D(v, \omega); \quad (19)$$

$$\text{Dist}_S(v, \omega) = \begin{cases} D_s, & D_s \geq L; \\ 0, & \text{otherwise}; \end{cases} \quad (20)$$

$$\text{Dist}_D(v, \omega) = \begin{cases} D_d, & D_d \geq S; \\ 0, & \text{otherwise}; \end{cases} \quad (21)$$

$$D_s = \min \|x_p, x_s\|, \quad D_d = \min \|x_p, x_d\|; \quad (22)$$

$$S = v_t \cdot \Delta t, \quad L = 2R_r. \quad (23)$$

其中: D_s 为轨迹预测端到静态障碍物的最短距离, D_d 为轨迹预测端到动态障碍物的最短距离, S 为机器人在预测时间 Δt 内运动的距离, R_r 为机器人回转半径, λ 为权重系数. 引入全局路径偏离函数, 在进行局部避障的前提下增强机器人对全局路径的追踪性能. 全局路径偏离函数为

$$\text{Dist}_{GP}(v, \omega) = \begin{cases} \text{dist}_{gp}, & \text{dist}_{ob} \geq L; \\ 0, & \text{otherwise}; \end{cases} \quad (24)$$

$$\text{dist}_{ob} = \min \|x_p, x_{uo}\|, \quad \text{dist}_{gp} = \min \|x_p, x_{gp}\|. \quad (25)$$

这里: dist_{gp} 为预测轨迹端到全局路径的最短距离, dist_{ob} 为当前时刻机器人与未知障碍物的最短距离. 引入终端收敛函数, 计算预测轨迹端点与子目标点的最短距离, 使得靠近子目标点的轨迹具有更高的权值, 从而避免机器人在目标点附近徘徊, 促使其快速靠近到达目标. 终端收敛函数为

$$\text{Subt}(v, \omega) = \begin{cases} \text{dist}_{st}, & \text{dist}_{rs} \leq R_{dete}; \\ 0, & \text{otherwise}; \end{cases} \quad (26)$$

$$\text{dist}_{st} = \min \|x_p, x_{st}\|, \quad \text{dist}_{rs} = \min \|x_r, x_{st}\|. \quad (27)$$

其中: dist_{st} 为预测轨迹端点与子目标的最短距离, dist_{rs} 为机器人与目标间的欧氏距离, R_{dete} 为传感器的障碍物最大探测距离.

在障碍物密集的环境中, 若不能及时调整线速度和角速度的大小可能会导致机器人速度过快或转角过小, 从而增加与障碍物碰撞的风险. 因此, 在障碍物较多的区域, 采用较低的线速度和较高的角速度可使得机器人平缓行驶, 确保轨迹机器人更快更安全地避开障碍物到达目标点. 机器人速度自适应调整如下所示:

$$v(t+1) = v(t) + \alpha_1 \frac{\delta_s(t+1) - \delta_s(t)}{\delta} v_{\max}, \quad (28)$$

$$\omega(t+1) = \omega_{\max} - \alpha_2 \frac{v(t)}{v_{\max}} \omega(t). \quad (29)$$

其中: v 和 ω 分别为机器人的线速度和角速度, δ_s 为预测轨迹可行区域, δ 为传感器最大探索区域, α_1 和 α_2 为调整参数.

2 双层路径规划方法

全局路径规划算法可以规划出一条质量较优的路径, 但是无法避开动态障碍物. 局部路径规划可以实时避开动态障碍物, 但是缺乏方向指引, 易陷入局部极小值. 为此, 本文提出了融合 PB-RRT 算法和 SDWA 算法优点的双层路径规划方法, 双层结构如图 7 所示, 充分利用了 PB-RRT 算法高效的全局路径规划性能和 SDWA 算法处理未知动态环境的能力. 当启发式全局路径规划完成后, 优化后的启发式路径上的关键节点被逐一取出, 作为 SDWA 算法的子目标点, 实时跟踪全局路径的同时避开未知的动态障碍物, 即通过 PB-RRT 算法得到初始路径, 经轨迹优化后得到有 n 个关键路径节点的全局路径. 然后, 以机器人当前位置作为局部路径规划的起始点, 将关键路径节点作为局部子目标, 采用 SDWA 算法进行局部路径规划.

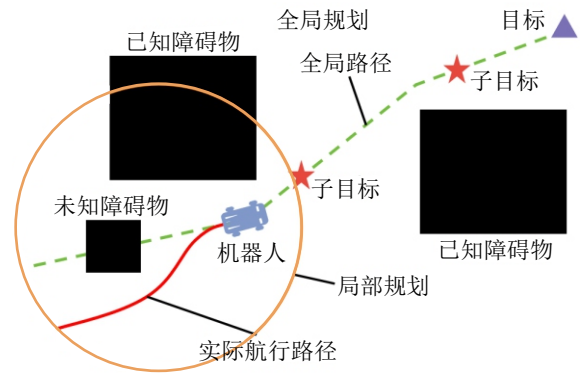


图7 双层方法结构

由于 DWA 算法在接近局部目标点时速度会明显降低, 本文采用了一种动态子目标策略. 计算机器人当前位置与动态子目标的距离, 若小于所设阈值, 则动态子目标将沿着动态子目标序列进行推移; 否则, 将保持在当前子目标点, 直至机器人安全行驶至目标点位置区域.

在双层方法的基础上, 引入了一种遵守避碰规则并提高机器人行驶安全性的方法. 主要利用双层方法始终存在一个动态子目标的特点, 通过修改动态目标点的位置, 引导机器人避开动态障碍物. 如图 8 所示, 当机器人移动至动态障碍物范围边界时, 确定动态障碍物的移动方向和机器人与动态障碍物连线间的角度 θ_{or} . 当角度 θ_{or} 在 $(-\pi/2, \pi/2)$ 范围内时,

需要偏置动态子目标引导. 偏置引导方向是机器人速度与动态障碍物速度矢量和的相反方向, 全局路径子目标从动态障碍物与全局路径冲突点向偏置引导方向平移距离 d_{bs} , 以引导机器人灵活避开动态障碍物. 当角度 θ_{or} 不在 $(-\pi/2, \pi/2)$ 范围内或机器人行驶出动态障碍物范围边界时, 动态子目标返回全局路径, 以引导机器人快速返回全局路径.

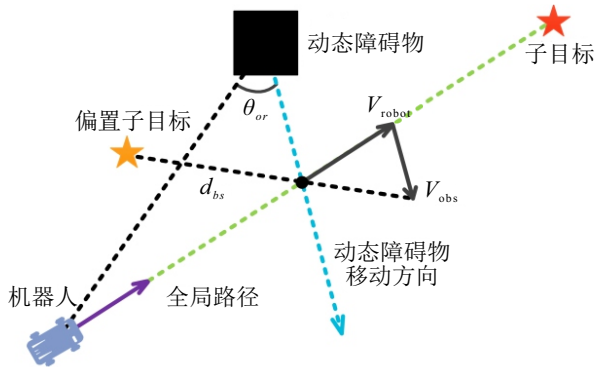


图8 子目标偏置引导

3 实验分析

为验证所提出方法的有效性和优越性, 首先, 构建 4 种环境地图, 将 PB-RRT 算法与 Bi-RRT^[7] 算法、GB-RRT^[9] 算法、RB-RRT^[12] 算法进行规划性能比较; 然后, 在不同环境地图中对比轨迹优化各阶段路径质量, 验证初始路径轨迹优化的必要性和优越性; 接着, 对双层方法在实时路径规划中的应用进行实验分析; 最后, 对真实实验场景进行环境建模, 规划生成机器人行驶路径, 验证所提出方法在实际应用中的有效性. 实验环境为 64 bit Windows11 操作系统, Matlab R2022b, 处理器为 Intel(R) Core(TM) i5-1340P, 主频 1.9 GHz, 内存 16 GB.

3.1 对比实验分析

在不同环境地图中进行规划路径对比测试, 测试环境分别为随机地图、走廊地图、仓库地图和迷宫地图, 地图规模均为 500×500 , 为机器人提供不同的场景环境. 随机地图由形状不规则障碍物随机组成; 走廊地图可行道路狭窄; 仓库地图整齐排列许多矩形货架; 迷宫地图中障碍物密集, 存在误导路径. 采用路径搜索时间、采样节点数、路径长度和路径节点数 4 个度量测试算法在不同环境中的适应性, 全面评估算法性能.

图 9 为各算法在 4 种环境中具体规划路径结果, 从左到右依次为 Bi-RRT、GB-RRT、RB-RRT 和 PB-RRT. 由图 9(a) ~ 图 9(d) 可见: Bi-RRT 算法采样缺乏方向性导致过度探索空间, 采样点分布广, 生成过

多无用节点, 路径冗余性较强; GB-RRT 算法采样扩展具有一定的方向性, 但是仍然存在多余的分支, 路径曲折, 在迷宫环境中采样效率极低; RB-RRT 算法通过减少不相关区域的搜索, 采样点更加集中, 但是, 在障碍物密集区域节点排斥机制可能会导致生成大量无效采样节点, 同时, 贪婪扩展策略也会增加更多的碰撞检测计算, 导致复杂区域中随机树扩展速度较慢; 在相同地图中, PB-RRT 算法随机树扩展具有目的性和方向性, 起点树与终点树相互引导, 减少了局部空间无效采样, 以极少的采样节点得到了更优的路径.

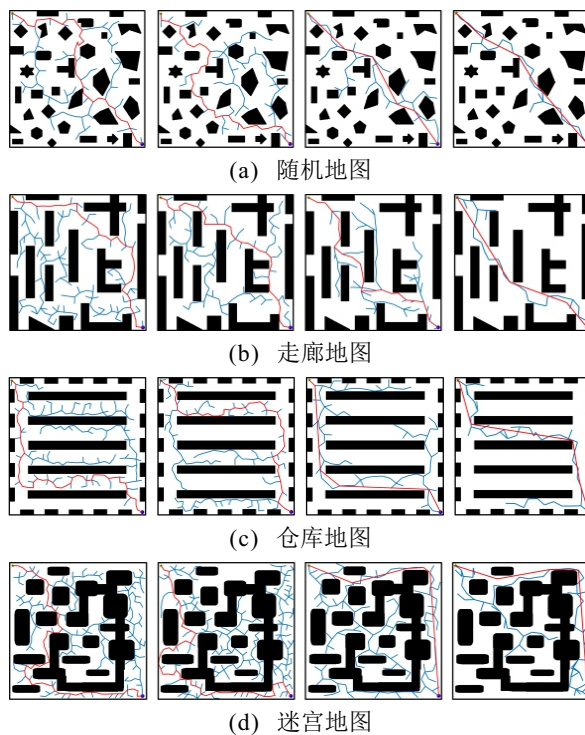


图9 算法规划路径对比

表 1 记录了经过 50 组独立实验规划路径的平均时间、标准差、平均采样节点数、平均路径长度和平均路径节点数. 由表 1 可以明显看出, 在任何地图环境中, PB-RRT 算法均可以以更少的采样节点和更短的时间得到更优的路径. 以随机地图为例, PB-RRT 算法平均规划时间比 Bi-RRT 减少了 67.7%, 比 GB-RRT 减少了 63.0%, 比 RB-RRT 减少了 22.2%. 在复杂的仓库地图和迷宫地图中平均规划时间缩短得更为明显, 且在所有地图中, PB-RRT 算法平均规划时间的标准差最小, 表明其规划性能稳定, 算法鲁棒性较强. 在规划过程中, PB-RRT 算法的采样节点数、路径长度以及路径节点数相较于 Bi-RRT 算法、GB-RRT 算法与 RB-RRT 算法均有明显的降低, 即 PB-RRT 算法可以使用更少的采样节点探索空间, 找出更优的路径解.

表1 算法实验数据对比

环境	算法	平均时间/s	最小时间/s	最大时间/s	标准差	平均采样数	平均路径长度	平均路径节点
随机地图	Bi-RRT	2.002	1.611	2.754	0.383	96.7	951.0	39.6
	GB-RRT	1.750	1.455	2.223	0.217	86.8	927.8	38.7
	RB-RRT	0.832	0.671	1.034	0.164	50.2	716.8	6.3
	PB-RRT	0.647	0.501	0.834	0.076	36.2	695.4	5.1
走廊地图	Bi-RRT	1.880	1.569	2.287	0.205	85.9	883.0	36.9
	GB-RRT	1.783	1.366	2.262	0.239	80.0	889.8	37.2
	RB-RRT	0.856	0.654	1.052	0.153	49.3	720.8	5.2
	PB-RRT	0.642	0.483	0.865	0.084	35.9	702.4	4.6
仓库地图	Bi-RRT	3.181	2.423	3.940	0.366	170.0	1025.2	52.6
	GB-RRT	2.620	1.961	3.402	0.408	129.4	1009.3	51.3
	RB-RRT	1.237	0.984	1.542	0.377	80.1	852.8	7.1
	PB-RRT	1.006	0.748	1.349	0.152	56.3	830.5	5.4
迷宫地图	Bi-RRT	5.179	4.332	6.347	0.454	258.5	1189.5	60.9
	GB-RRT	4.670	3.629	5.667	0.546	238.6	1196.3	61.3
	RB-RRT	2.421	1.786	3.214	0.531	123.6	936.2	6.5
	PB-RRT	1.841	1.309	2.710	0.336	103.4	915.1	5.2

3.2 轨迹优化结果

验证轨迹优化策略的必要性和优越性,将初始路径进行全局优化和局部优化后,采用插值B样条曲线拟合平滑.初始路径存在大量冗余节点,导致路径长度较长,全局重布线优化后,路径节点和路径长度大幅减少,但是,优化路径仍然由初始路径的部分节点连接构成,存在非必要拐点.经局部重布线优化后,再次删除路径冗余节点以提升路径质量.

轨迹优化实验结果如图10所示.其中:蓝色路径为初始路径,黑色路径为全局重布线优化得到的路径,红色路径为局部重布线优化后采用样条拟合优化后的路径.由图10可见:在不同环境地图中,轨迹优化后得到的路径更短更平滑,相比于初始路径,路径质量显著提高,更适合机器人行驶.

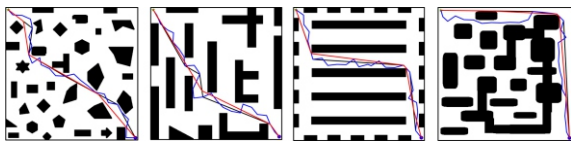
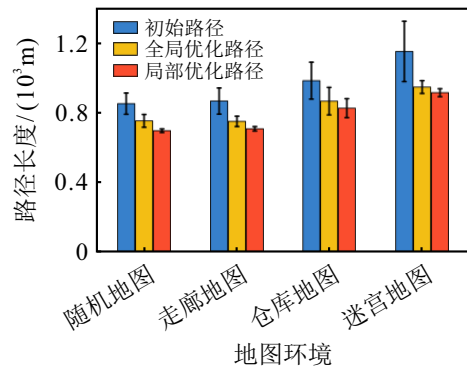


图10 初始路径优化对比

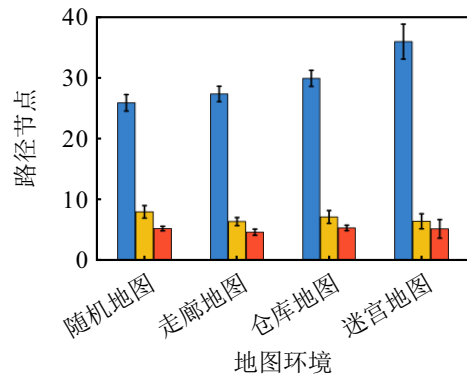
图11为4种场景下轨迹优化各阶段路径长度和路径节点的对比结果.由图11可见,轨迹优化后路径质量明显升高,路径长度和路径节点数均大幅度降低且波动幅度较小,这表明了轨迹优化策略对于提高路径质量有一定的效果,特别是在复杂环境中.

3.3 动态环境下实验结果

在随机地图和走廊地图中分别添加一些动态障碍物,障碍物在蓝色路径上来回移动.以文献[16]方法为对比进行动态路径规划实验,验证所提出双层



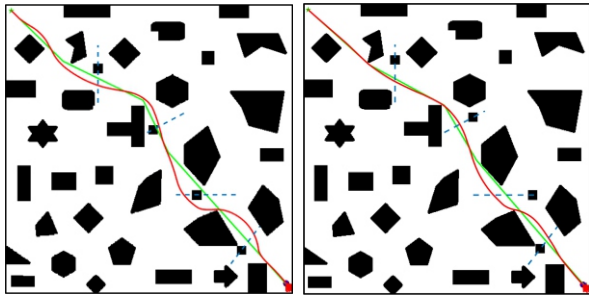
(a) 路径长度



(b) 路径节点

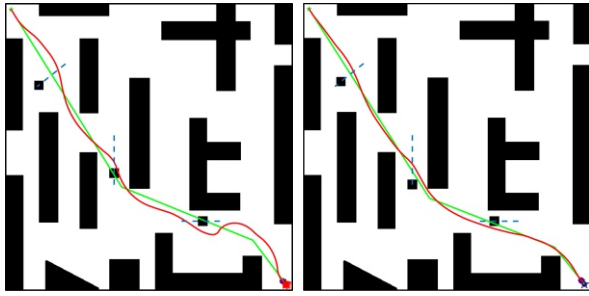
图11 轨迹优化各阶段数据对比

方法的性能.轨迹评价函数中权重系数如下: $k_1 = 0.1$, $k_2 = 0.2$, $k_3 = 0.2$, $k_4 = 0.3$, $k_5 = 0.2$.规划路径对比结果如图12和图13所示.文献[16]方法在遇到障碍物时由于速度控制不合理而产生具有显著波动的路径.相比之下,由于动态子目标引导,所提出双层方法生成的路径具有良好的平滑性.在导航期间,机器人实时运动轨迹贴合全局路径,转角处速度控制合理,避开动态障碍物的同时能够通过动态子目标点返回全局路径,直至到达全局目标位置.



(a) 文献[16]方法 (b) PB-RRT-SDWA

图12 随机地图动态环境实验结果对比



(a) 文献[16]方法 (b) PB-RRT-SDWA

图13 走廊地图动态环境实验结果对比

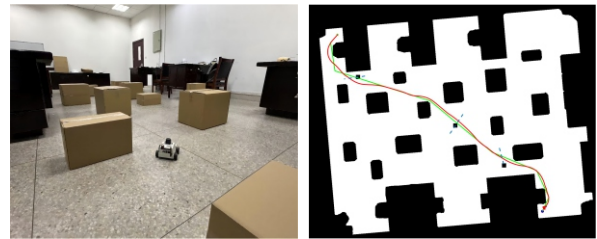
表2为从50组动态路径规划实验中获得的平均数据,包括规划时间、路径长度以及与动态障碍物的最小距离 D_{min} 。由表2可见,所提出方法的规划时间最短,路径长度最短,与动态障碍物的最小距离值最大。由于地图分辨率较大,文献[16]采用改进蚁群算法搜索全局路径的时间较长。在随机地图中:所提出方法的规划时间比文献[16]方法减少了41.6%;此外,规划的路径长度缩短了4.6%。仿真结果表明,动态子目标策略和合理的速度控制可显著减少规划时间和路径长度,偏置引导策略使得机器人能够与动态障碍物保持相对安全的距离。

表2 算法实验数据对比

环境	算法	时间/s	路径长度	D_{min}
随机地图	文献[16]方法	236.7	758.4	5.1
	PB-RRT-SDWA	138.2	723.7	8.7
走廊地图	文献[16]方法	213.7	764.6	5.6
	PB-RRT-SDWA	121.1	726.8	8.4

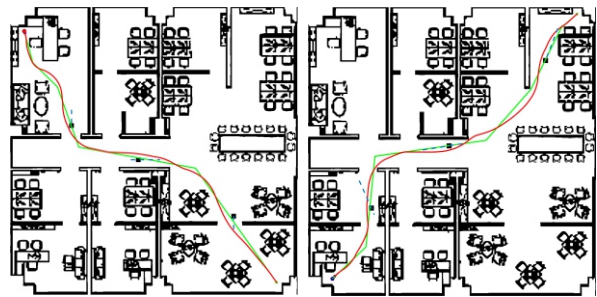
在真实室内环境中测试双层路径规划方法在实际应用中的有效性,测试场地为图14(a)所示的10m×7m的办公室,地图分辨率为1cm。机器人线速度和角速度被约束为 $v \in (0, 0.4) \text{ m/s}$ 和 $\omega \in (-3.14, 3.14) \text{ rad/s}$ 。实验结果如图14(b)所示,机器人找到了质量较优的全局路径。按照全局路径规划子目标进行局部路径规划,在到达未知动态障碍物碰撞范围内时,快速重新规划选择合适的行驶路径并发送

给机器人以规避动态障碍物。机器人行驶出动态障碍物影响边界后,快速回到全局路径并安全到达目标点。然后,选择更大更复杂的办公室楼层场景,如图15所示。由于有动态子目标引导,机器人不会在复杂环境中迷失方向,子目标偏置策略使得机器人灵活避开动态障碍物。由图15可见,在导航过程中,机器人在拐弯处速度控制更优,及时避开动态障碍物后快速回到全局路径,平稳到达目标。最后,选择一个真实的复杂室外环境进行机器人路径规划和避障实验,如图16所示。机器人快速规划出全局路径,沿着全局路径安全避开6个动态障碍物移动至目标位置。实验结果表明,双层路径规划方法在实际应用中是有效的。



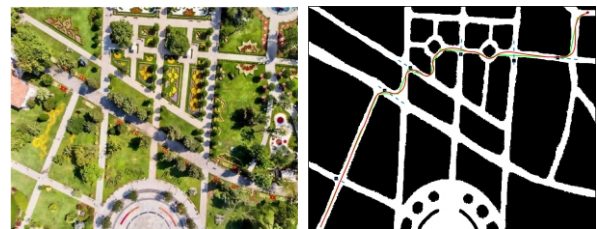
(a) 真实实验场景(办公室) (b) 机器人完整路径

图14 室内环境中实验结果



(a) 复杂办公室1 (b) 复杂办公室2

图15 办公室楼层环境中实验结果



(a) 真实公园场景 (b) 机器人完整路径

图16 室外环境中实验结果

4 结论

本文提出了一种复杂动态环境中双层路径规划方法。在第1层中,采用基于并行采样和双树引导的PB-RRT算法,建立了评价函数对并行采样点进行筛选选择优扩展,同时兼顾了路径长度和路径平滑度;构

建了双树引导机制,提高了双向随机树融合速度;根据节点周围环境信息进行自适应扩展,防止局部空间无效节点生成;结合轨迹优化方法,提高了初始路径质量.为解决基于采样的算法不能避开动态障碍物的问题,提取了优化路径关键节点作为动态子目标点,引入了第2层SDWA算法和偏置引导方法来规划机器人在动态环境中能够运行的轨迹,安全避开了动态障碍物.各种复杂动态环境下的实验结果表明,双层路径规划方法能够快速生成质量较优的全局路径,高效避开动态障碍物并安全到达目标位置,验证了所提出方法的有效性和优越性.

参考文献 (References)

- [1] 魏立新, 张钰银, 孙浩, 等. 基于改进蚁群和 DWA 算法的机器人动态路径规划[J]. 控制与决策, 2022, 37(9): 2211-2216.
(Wei L X, Zhang Y K, Sun H, et al. Robot dynamic path planning based on improved ant colony and DWA algorithm[J]. Control and Decision, 2022, 37(9): 2211-2216.)
- [2] 张腾龙, 李擎. 基于 B-RRT*FND 算法的移动机器人路径规划[J]. 控制与决策, 2023, 38(11): 3121-3127.
(Zhang T L, Li Q. Path planning of AGV based on B-RRT*FND algorithm[J]. Control and Decision, 2023, 38(11): 3121-3127.)
- [3] 曹祥红, 李欣妍, 魏晓鹤, 等. 基于 Dijkstra-ACO 混合算法的应急疏散路径动态规划[J]. 电子与信息学报, 2020, 42(6): 1502-1509.
(Cao X H, Li X Y, Wei X G, et al. Dynamic programming of emergency evacuation path based on dijkstra-ACO hybrid algorithm[J]. Journal of Electronics & Information Technology, 2020, 42(6): 1502-1509.)
- [4] Szczepanski R. Safe artificial potential field-novel local path planning algorithm maintaining safe distance from obstacles[J]. IEEE Robotics and Automation Letters, 2023, 8(8): 4823-4830.
- [5] Zhang Y, Wang H Y, Yin M L, et al. Bi-AM-RRT*: A fast and efficient sampling-based motion planning algorithm in dynamic environments[J]. IEEE Transactions on Intelligent Vehicles, 2024, 9(1): 1282-1293.
- [6] 许万, 杨晔, 余磊涛, 等. 一种基于改进 RRT* 的全局路径规划算法[J]. 控制与决策, 2022, 37(4): 829-838.
(Xu W, Yang Y, Yu L T, et al. A global path planning algorithm based on improved RRT*[J]. Control and Decision, 2022, 37(4): 829-838.)
- [7] Kuffner J J, LaValle S M. RRT-connect: An efficient approach to single-query path planning[C]. IEEE International Conference on Robotics and Automation. San Francisco, 2000: 995-1001.
- [8] 陈丹, 谭钦, 徐哲壮. 基于采样点优化 RRT 算法的机械臂路径规划[J]. 控制与决策, 2024, 39(8): 2597-2604.
(Chen D, Tan Q, Xu Z Z. Robotic arm path planning based on sampling point optimization RRT algorithm[J]. Control and Decision, 2024, 39(8): 2597-2604.)
- [9] Liu H Y, Zhang X B, Wen J, et al. Goal-biased bidirectional RRT based on curve-smoothing[J]. IFAC-PapersOnLine, 2019, 52(24): 255-260.
- [10] Jiang L H, Liu S Y, Cui Y M, et al. Path planning for robotic manipulator in complex multi-obstacle environment based on Improved_RRT[J]. IEEE/ASME Transactions on Mechatronics, 2022, 27(6): 4774-4785.
- [11] Hu B, Cao Z C, Zhou M C. An efficient RRT-based framework for planning short and smooth wheeled robot motion under kinodynamic constraints[J]. IEEE Transactions on Industrial Electronics, 2021, 68(4): 3292-3302.
- [12] Wang L F, Zhang Y D, Guo C L. Path planning for a prostate intervention robot based on an improved Bi-RRT algorithm[J]. IEEE/ASME Transactions on Mechatronics, 2025, 30(1): 668-678.
- [13] 陈锦涛, 李鸿一, 任鸿儒, 等. 基于 RRT 森林算法的高层消防多无人机室内协同路径规划[J]. 自动化学报, 2023, 49(12): 2615-2626.
(Chen J T, Li H Y, Ren H R, et al. Cooperative indoor path planning of multi-UAVs for high-rise fire fighting based on RRT-forest algorithm[J]. Acta Automatica Sinica, 2023, 49(12): 2615-2626.)
- [14] Qi J, Yang H, Sun H X. MOD-RRT*: A sampling-based algorithm for robot path planning in dynamic environment[J]. IEEE Transactions on Industrial Electronics, 2021, 68(8): 7244-7251.
- [15] Dobrevski M, Skočaj D. Dynamic adaptive dynamic window approach[J]. IEEE Transactions on Robotics, 2024, 40(3): 3068-3081.
- [16] Liu H S, Yue M, Liu M H, et al. Two-layer path planning framework for WMRs in dynamic environments: Optimized ant colony algorithm and dynamic window approach[J]. Transactions of the Institute of Measurement and Control, 2025, 47(1): 1-15.
- [17] Yasuda S, Kumagai T, Yoshida H. Safe and efficient dynamic window approach for differential mobile robots with stochastic dynamics using deterministic sampling[J]. IEEE Robotics and Automation Letters, 2023, 8(5): 2614-2621.

作者简介

胡耀炜 (2000-), 男, 硕士生, 主要研究方向为移动机器人自主导航、智能控制, E-mail: hyw0911@ahnu.edu.cn;

汤萍萍 (1981-), 女, 副教授, 博士, 主要研究方向为人工智能、机器人智能感知, E-mail: tpping@ahnu.edu.cn;

张晖 (1982-), 男, 教授, 博士, 主要研究方向为人工智能、机器人智能感知, E-mail: zhhj@126.com;

陈旭飞 (2001-), 男, 硕士生, 主要研究方向为移动机器人自主导航、智能控制, E-mail: chenxufei@ahnu.edu.cn.