

基于改进双向 A* 与动态窗口法的移动机器人路径规划

黄嘉佳, 罗 莎, 黄金贵

(湖南师范大学 信息科学与工程学院, 湖南 长沙 410081)

摘 要: 路径规划是移动机器人自主导航中的核心步骤, 旨在从起点到目标点生成一条无碰撞的、代价最小的轨迹。为提升移动机器人路径规划的效率和全局最优性, 提出一种融合改进双向 A* 搜索与动态窗口法的 IBA-IDWA 算法。该算法在 A* 搜索中引入基于障碍物分布的启发式函数以提高搜索效率, 并在动态窗口法中设计基于轨迹端点与局部目标距离的代价函数以增强全局一致性。同时, 该算法通过预测障碍物运动模式、识别危险区域, 并根据环境障碍物密度自适应调整机器人速度窗口大小。仿真实验结果表明, 与传统 A* 算法相比, 改进双向 A* 算法的规划效率提升 14.07%; 在动态避障任务中, 与现有融合算法相比, IBA-IDWA 的避障成功率提高 5% 以上。IBA-IDWA 算法能在静态与动态环境中显著提升路径规划效率与路径质量, 为移动机器人自主导航提供了一种有效方案。

关键词: 移动机器人; 路径规划; 双向 A* 算法; 动态窗口法

DOI: 10.11907/rjdk.251614

扫描二维码阅读全文:



中图分类号: TP312

文献标识码: A

文章编号: 1672-7800(XXXX)0XX-0001-10

Path Planning for Mobile Robots Based on Improved Bidirectional A* and Dynamic Window Method

HUANG Jiajia, LUO Sha, HUANG Jingui

(College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China)

Abstract: Path planning is a core step in autonomous navigation for mobile robots, aiming to generate a collision-free and cost-minimized trajectory from the starting point to the target point. To enhance the efficiency and global optimality of path planning for mobile robots, an IBA-IDWA algorithm that integrates improved bidirectional A* search and dynamic window approach is proposed. This algorithm introduces a heuristic function based on obstacle distribution in A* search to improve search efficiency, and designs a cost function based on the distance between trajectory endpoints and local goals in the dynamic window approach to enhance global consistency. Additionally, the algorithm adaptively adjusts the robot speed window size based on the density of environmental obstacles by predicting obstacle movement patterns and identifying dangerous areas. Simulation experimental results show that compared to the traditional A* algorithm, the planning efficiency of the improved bidirectional A* algorithm is improved by 14.07%. In dynamic obstacle avoidance tasks, compared to existing fusion algorithms, the obstacle avoidance success rate of IBA-IDWA is increased by more than 5%. The IBA-IDWA algorithm significantly improves path planning efficiency and path quality in both static and dynamic environments, providing an effective solution for autonomous navigation of mobile robots.

Key Words: mobile robot; path planning; bidirectional A* algorithm; dynamic window method

0 引言

路径规划是移动机器人自主导航中的关键技术之一,

其主要任务是根据已知、未知或部分已知的环境信息为机器人从起点到目标点生成一条安全、无碰撞且较优的运动轨迹。对于动态场景而言, 环境状态并不是固定不变的, 障碍物可能会随时间发生位置变化, 因此路径规划算法一

收稿日期: 2025-09-03

基金项目: 国家自然科学基金项目 (62573182)

作者简介: 黄嘉佳 (2001-), 女, CCF 学生会会员, 湖南师范大学信息科学与工程学院硕士研究生, 研究方向为机器人的路径规划; 罗莎 (1992-), 女, 博士, 湖南师范大学信息科学与工程学院讲师, 研究方向为机器人控制、路径规划、强化学习、模仿学习、并行计算、数据科学; 黄金贵 (1964-), 男, 博士, 湖南师范大学信息科学与工程学院教授、博士生导师, 研究方向为 NP 难问题的精确算法、近似算法及参数算法的理论研究、深度学习、图神经网络与大模型等及其应用。本文通讯作者: 黄金贵。

方面需要具备一定的环境感知能力,另一方面也要能及时响应外部变化,以保证机器人在复杂环境中的导航过程更加稳定可靠。

现有路径规划方法一般可以分为全局规划和局部规划两大类。全局路径规划更多依赖于对环境信息的完整建模与整体认知,通常会借助优化方法来完成路径搜索,代表性方法包括基于采样的算法,如快速扩展随机树(Rapidly-exploring Random Tree, RRT)^[1-2]、概率道路图(Probabilistic Roadmap Method, PRM)^[3],以及基于图搜索的经典算法,如A*算法^[4-9]、Dijkstra算法^[10-11]、深度优先搜索等。其中,双向A*算法会从起点和目标节点两个方向同时展开启发式搜索,这样能够在一定程度上缩小搜索范围,从而提高路径搜索效率。局部路径规划则更多关注机器人在动态环境和未知环境中的实时避障问题,以及局部范围内的轨迹生成过程,代表性方法包括动态窗口法(Dynamic Window Approach, DWA)^[12-14]、模型预测控制(Model Predictive Control, MPC)、人工势场法(Artificial Potential Field, APF)^[15-16]等。该类方法在处理近距离障碍物和实时避障任务时有较好效果,亦可根据机器人周围环境的变化及时调整运动轨迹。由于局部路径规划一般缺少完整全局环境信息的支持,只能依据当前感知到的局部信息进行决策,在很多情况下的路径整体最优性无法得到保证。

近年来,随着机器人技术的不断发展,移动机器人已经被应用到越来越复杂的动态环境中,其任务需求也更加多样,这对路径规划算法的计算效率、实时响应能力以及环境适应性提出了更高要求。传统A*算法主要采用单向启发式搜索策略,在环境较为简单、任务需求相对固定的情况下能取得较好的路径搜索效果。但当环境变得复杂,或导航任务具有较强的动态性和多样性时,该算法便难以应对。针对这一问题,现有研究提出多种改进A*变种方法,以改善其在动态环境下的表现。例如,Gan等^[17]将Q-Learning引入A*算法的评估函数中,使算法能够在动态环境下表现出更好的自适应能力。然而,该方法通常需要大量且具有多样性的训练样本作为支撑,这在一定程度上限制了其在实际场景中的应用。Liu等^[18]提出一种改进的A*-DWA融合方法,通过对节点扩展策略进行调整来提高路径规划整体性能,但在机器人高速运动的场景中,该方法的不稳定性不佳。Guo等^[19]提出一种改进A*算法与DWA的融合算法,利用关键节点信息对方向角度进行调整,从而达到优化路径长度的目的,然而其有效性主要体现在小规模环境中,当环境规模增大或障碍物分布更加复杂时,算法的适用范围会受到一定影响。Yang等^[20]提出融合增强型DWA与优化A*算法的路径规划方法,但生成的路径存在过多急转弯的问题,平滑性不佳。Zhang等^[21]提出改进A*算法与DWA的融合算法,提取改进A*算法生成的全局路径关键节点作为DWA的局部目标点,解决了传统DWA易陷入局部最优的问题,但融合后算力开销显

著上升,会直接影响移动机器人导航的实时性表现。

针对现有方法在计算效率和路径平滑性方面存在的不足,本研究提出一种融合改进双向A*(Improved Bidirectional A*, IBA)与改进动态窗口法(Improved Dynamic Window Approach, IDWA)的路径规划方法,以兼顾全局路径规划与局部实时避障的需求,主要贡献点为:①在启发式函数中引入基于障碍物分布的自适应权重,使算法能够结合环境中障碍物的分布情况来调整搜索方向,减少了传统A*算法对固定环境信息的依赖,亦不需要额外的训练过程,有助于降低搜索复杂度;②为提高局部路径规划效果,在DWA轨迹评估函数中加入局部目标点引导机制;同时结合动态障碍物的运动模式预测危险区域,并设计了基于障碍物密度的速度窗口自适应调节机制,以增强算法在动态环境下的实时响应能力;③通过消融实验和对比实验验证了所提方法的有效性。实验结果表明,IBA-IDWA方法在路径平均长度与避障成功率方面相较多种基线方法有更好表现。

1 基本理论

1.1 双向A*算法

A*是一类典型的启发式搜索算法,基本思路是在每一轮迭代过程中从候选节点中选取代价值最小的节点进行扩展,并逐步在搜索空间内寻找一条从起点到目标点的最优路径。其评估函数表示为:

$$f(n) = g(n) + h(n) \quad (1)$$

式中: $f(n)$ 表示节点 n 的总估计代价; $g(n)$ 表示从起点到节点 n 的实际代价; $h(n)$ 表示从节点 n 到目标点的启发式估计值。常见的启发式估计方法包括曼哈顿距离、欧几里得距离和切比雪夫距离等。

在启发函数 $h(n)$ 同时满足可采纳性和一致性(单调性)的情况下,A*算法能够保证完备性和最优性,因此在全局路径规划中得到较为广泛的应用。然而,在大规模环境中,随着搜索空间的不断增大,A*算法的时间复杂度会呈现指数级增长,计算效率和内存消耗方面也会面临较大压力。为了缓解这些问题,双向A*算法采用从起点和目标节点同时开展正向搜索与反向搜索的方式,减少了不必要的搜索范围,提高了整体搜索效率。其基本步骤为:①初始化。建立两个开放列表(其中 $open_s$ 用于存储从起点出发的路径节点, $open_g$ 用于存储从目标点出发的路径节点),同时设置对应的两个关闭列表($closed_s$ 和 $closed_g$);②双向扩展。分别在两个开放列表中选择 f 值最小的节点进行扩展,并将已经完成扩展的节点加入到相应的关闭列表中;③交点检查。在每次节点扩展完成后检查 $open_s$ 中的最小 f 值节点是否存在于 $closed_g$ 中,或 $open_g$ 中的节点是否存在于 $closed_s$ 中。若存在,则进入步骤④,否则继续执行步骤②;④路径重建。从起点方向追踪

到交点, 再从目标点方向反向追踪到交点, 由此拼接形成完整路径; 在此基础上消除路径中的冗余节点, 以进一步提高路径的平滑性; ⑤终止条件。不断重复节点扩展和交点检查过程, 直到找到一条有效路径, 或确认当前环境中不存在可行路径为止。

1.2 DWA

DWA 是一类基于采样思想的局部路径规划算法, 其通常在机器人的速度空间内选取若干组线速度与角速度对 (v, w) , 并根据这些速度组合预测机器人在短时间内可能形成的运动轨迹。随后, 算法会利用轨迹评估函数对不同候选轨迹进行综合评价, 从中选择较优的速度对, 使机器人能够在运动过程中兼顾实时避障与轨迹跟踪。

1.2.1 移动机器人运动学模型

在 DWA 中, 机器人会先在给定的速度窗口内选取可行速度, 并基于运动学模型对其后续运动进行预测, 从而生成一组未来候选轨迹。在离散时间间隔 Δt 内, 机器人的运动学模型可表示为:

$$\begin{cases} x = x + v_t \Delta t \cos \theta_t \\ y = y + v_t \Delta t \sin \theta_t \\ \theta_t = \theta_t + w_t \Delta t \end{cases} \quad (2)$$

式中: (x, y) 表示机器人在平面坐标系中的位置; v_t 为线速度; w_t 为角速度; θ_t 为机器人在时刻 t 的航向角。该模型在保证计算简洁性的同时能有效预测机器人在短时间内的运动状态。

1.2.2 速度采样约束

虽然在速度空间中存在无限多速度组合, 但是实际采样必须受到机器人物理特性和环境约束的限制。DWA 通过定义动态窗口, 将采样速度限制在可行范围内, 主要包括以下两个约束条件:

(1) 速度限制表示为:

$$v_m = \{(v, w) | v \in [v_{min}, v_{max}], w \in [w_{min}, w_{max}]\} \quad (3)$$

式中: v_{max} 、 v_{min} 分别为线速度的最大值与最小值; w_{max} 、 w_{min} 分别为角速度的最大值与最小值。

(2) 加速度限制表示为:

$$v_d = \{(v, w) | v \in [v_c - v_a \Delta t, v_c + v_a \Delta t], w \in [w_c - w_a \Delta t, w_c + w_a \Delta t]\} \quad (4)$$

式中: v_c 、 w_c 分别为当前的线速度与角速度; v_a 、 w_a 分别为线速度与角速度的最大加速度。

机器人可行的最终速度集合定义为上述两类约束条件的交集。表示为:

$$(v, w) = v_m \cap v_d \quad (5)$$

1.2.3 轨迹评估函数

为在速度采样空间中选择最优速度对 (v, w) , DWA 通过式(6)中 3 个关键指标的加权和对所有候选轨迹进行评估。表示为:

$$G(v, w) = \alpha \text{Head}(v, w) + \beta \text{Vel}(v, w) + \gamma \text{Dist}(v, w) \quad (6)$$

式中: $\text{Head}(v, w)$ 表示航向角评价函数, 用于描述机器人航向与目标方向之间的角度差 (以 2π 减去该角度差); $\text{Vel}(v, w)$ 表示速度评价函数, 用于鼓励选择较大的线速度, 以提升运动效率; $\text{Dist}(v, w)$ 表示距离评价函数, 用于预测轨迹点到最近障碍物的最小距离, 以保证轨迹的安全性; α 、 β 、 γ 为权重系数, 用于在不同任务和环境条件下平衡 3 类指标的影响。

在动态窗口内对所有满足约束的可行速度对计算 $G(v, w)$ 后, 选取评估得分最大的速度对作为最优解。计算公式为:

$$(v^*, w^*) = \underset{(v, w) \in V_{dw}}{\text{argmax}} G(v, w) \quad (7)$$

式中: V_{dw} 表示可行速度集合, 该集合中的速度对需要同时满足速度约束、加速度约束以及无碰撞要求。经过评价函数筛选后得到的最优速度对 (v^*, w^*) 将作为机器人的下一步控制输入。随后, 机器人在新的规划周期内继续重复速度采样、轨迹预测和评价选择等过程, 以实现轨迹的实时更新与避障控制。

DWA 在动态环境中具有较好的实时性和计算效率, 生成的避障轨迹也相对平滑。然而, 从方法本身来看, DWA 仍然属于局部规划方法, 其规划效果受到感知范围和短期预测能力的限制。因此, 在障碍物分布较复杂或目标引导信息不足的场景下, 机器人可能会陷入局部最优路径, 并出现路径振荡, 甚至发生碰撞等问题。

2 IBA-IDWA 算法

2.1 IBA 算法

传统双向 A* 算法在简单环境中通常能够取得较好的搜索效果, 但当环境中障碍物分布较密集时, 其路径搜索性能容易受到影响。针对这一问题, 本研究提出 IBA 算法, 通过引入基于局部障碍物密度的启发式函数, 并结合冗余节点消除策略来改善路径搜索的全局最优性和计算效率。

2.1.1 基于障碍物密度调整的启发式函数

传统的启发式函数一般主要考虑机器人当前位置与目标点之间的几何距离, 而对环境复杂度的影响考虑较少。这种处理方式在无障碍物或障碍物较少的场景中较为有效, 但在复杂环境下, 尤其是障碍物密集区域, 容易使代价估计产生偏差, 进而生成次优路径; 或在搜索过程中引入较多冗余节点。虽然欧几里得距离在无障碍物场景中是一种有效的启发式方法, 但在障碍物密集的环境中, 其往往会导致估计成本 $h(n)$ 与实际路径成本之间产生显著差异, 这是由于障碍物的存在和分布破坏了欧几里得估计中固有的直线假设。为此, 本研究提出一种能够根据局部障碍物密度动态调整的启发式函数。表示为:

$$f(n) = g(n) + (1 + K\rho(n))h(n) \quad (8)$$

式中: $g(n)$ 表示从起点到当前节点 n 的实际代价; $h(n)$

表示当前节点 n 到目标点的欧几里得距离; K 为权重因子, 用于控制障碍物密度对启发式的影响程度; $\rho(n)$ 表示节点 n 周围固定半径邻域内的归一化障碍物密度, 即邻域内被占用单元格数与总单元格数的比值。令 $\alpha(n) = 1 + K\rho(n)$, 则调整后的启发式为 $\alpha(n)h(n)$ 。如果原始启发式 $h(n)$ 是可采纳的 (即 $h(n) \leq h^*(n)$, 其中 $h^*(n)$ 是真实的最优代价), 则 $\alpha(n)$ 的引入对算法最优性与完备性的影响可分为以下两种情况:

(1) 严格可采纳性条件。若满足式(9):

$$\alpha(n)h(n) \leq h^*(n) \quad \forall n \quad (9)$$

则 $\alpha(n)h(n)$ 保持可采纳性, 保留了 A* 的最优性与完备性。这一条件等价于式(10)。表示为:

$$K \leq \min_{n: \rho(n) > 0} \frac{\frac{h^*(n)}{h(n)} - 1}{\rho(n)} \quad (10)$$

在实际问题中, $h^*(n)$ 不可直接获得, 因此上界是难以精确计算的, 但可以作为保证可采纳性的理论参考。

(2) 有界次优性保证。在实际应用中, 如果只确保 $\alpha(n) \leq \alpha_{\max}$, 其中 $\alpha_{\max} = 1 + K \cdot \max_{\rho(n)}$, 则搜索过程等价于权重为 $w = \alpha_{\max}$ 的加权 A* 算法。加权 A* 算法的解满足式(11)。表示为:

$$C \leq w \cdot C^* \quad (11)$$

式中: C^* 为最优解的代价。该界限来自 $g(n) + wh(n)$ 的单调性以及加权 A* 算法的终止条件。

2.1.2 权重因子 K 的实验评估

表 1 为 IBA 算法在不同地图尺寸和权重因子 K 值下的性能表现, 评价指标为路径长度与规划时间。可以看出, 在 50×50 的地图上, 当 K 值处于 $[1, 3.5]$ 区间时, 算法在路径最优性与计算效率之间达成平衡。在 40×40 的地图上, 规划时间对 K 值的变化相对不敏感, 且在 K 值处于 $[2, 3]$ 范围内时可获得最短路径。在 30×30 的地图上, 当 K 值处于 $[0, 2]$ 范围内时, 生成的路径接近最优。综合分析表明, K 的最优值为 2, 此时在路径长度与规划时间之间提供了最佳折衷。值得注意的是, 当 K 大于 5 时, 性能趋于稳定。这是由于在高障碍密度区域, 修正后的启发式函数 $(1 + K\rho(n))h(n)$ 逐渐主导总成本 $f(n)$, 削弱了实际成本 $g(n)$ 的影响。因此, 节点扩展方向主要受启发式部分的控制, 而 K 的进一步增加对算法行为的影响微乎其微。

2.1.3 冗余节点消除策略

全局路径规划算法生成的初始轨迹中通常会包含一定数量的冗余节点, 这些节点可能会使机器人产生不必要的转向, 或使路径偏离理论最短轨迹。为进一步提高路径效率和路径简洁性, 本研究设计一种受障碍物约束的冗余节点消除策略。通过该策略处理后, 路径中的中间节点数量减少, 整体轨迹更加平滑, 同时仍能保证路径的安全性和可行性。消除过程包括以下步骤: ①迭代节点检查。从全局路径的第二个节点开始, 算法依次检查后续路径节

Table 1 Performance of the IBA algorithm under different map sizes and weight factor K values

表 1 IBA 算法在不同地图尺寸和权重因子 K 值下的性能表现

地图大小	评价指标	K									
		0	0.5	1	1.5	2	2.5	3	3.5	4	≥ 5
50×50	路径长度/m	66.56	66.56	66.49	66.49	66.49	66.49	66.49	66.49	66.49	68.39
	规划时间/s	0.30	0.21	0.17	0.17	0.22	0.18	0.29	0.23	0.36	0.16
40×40	路径长度/m	56.06	56.06	55.67	55.67	55.29	55.29	55.29	59.86	59.95	59.95
	规划时间/s	0.16	0.15	0.13	0.15	0.16	0.16	0.15	0.16	0.15	0.15
30×30	路径长度/m	44.46	44.46	44.46	44.46	44.46	45.11	45.11	44.90	45.27	45.27
	规划时间/s	0.13	0.13	0.13	0.17	0.14	0.13	0.19	0.15	0.16	0.16

点。若当前节点满足冗余判定条件, 则将其从路径中移除; ②冗余性评估。对于每一个候选节点, 算法判断其父节点与后续节点之间的直接连接, 即是否会与环境中的障碍物发生相交。如图 1 所示, 长方形表示障碍物, 实线表示原始全局路径, 虚线表示路径点之间的直接线性连接。当该连接线不与任何障碍物相交时, 说明当前节点对路径连通性和避障没有作用, 可将其视为冗余节点并移除 (如 c_1 节点)。相反, 如果连接线与障碍物相交, 则说明该节点在绕障过程中仍然必要, 需予以保留 (如 c_2 节点)。

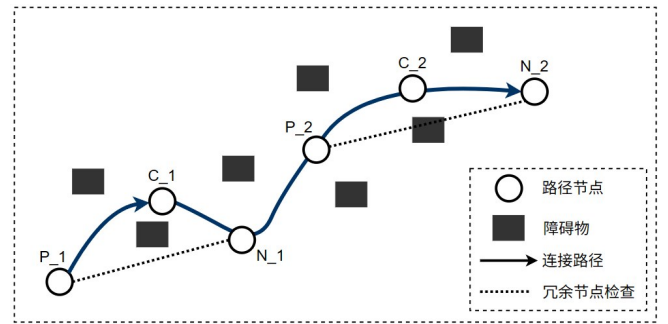


Fig. 1 Redundancy detection

图 1 冗余点检测

冗余节点消除策略能尽可能保留路径中必要的转折节点, 从而使生成路径更加直接和平滑。IBA 算法结合了障碍物感知启发式函数和冗余节点消除策略, 其伪代码为:

```

输入: 起始点 start, 目标点 goal
输出: 全局路径 path
1. open_s, open_g ← [start], [goal]
2. closed_s, closed_g ← ∅, ∅
3. WHILE open_s ≠ ∅ and open_g ≠ ∅
4.   u_f ← open_s.pop_min()
5.   IF u_f ∈ closed_g THEN
6.     path ← ReconstructPath(u_f)
7.     RETURN remove_redundant_nodes(path)
8.   ENDIF
9.   closed_s.add(u_f)
10.  FOR each v ∈ neighbors(u_f) DO
11.    update f(v)
12.  ENDFOR
13.  u_b ← open_g.pop_min()
14.  IF u_b ∈ closed_s THEN

```

```

15.  path ← ReconstructPath( u_b )
16.  RETURN remove_redundant_nodes( path )
17.  ENDIF
18.  closed_g.add_( u_b )
19.  FOR each v ∈ neighbors( u_b ) DO
20.    update f( v )
21.  ENDFOR
22.  ENDWHILE
23.  ReconstructPath( meet_node ):
24.    path_start ← path from start to meet_node
25.    path_goal ← path from meet_node to goal
26.    RETURN path_start + path_goal
27. RETURN None

```

该算法的第4-12行对应从起始节点开始的前向搜索过程。第13-21行对应从目标节点开始的后向搜索过程。在两阶段搜索中,邻接节点的启发式值均通过式(8)计算。第23-26行详细说明了路径重建函数的实现过程。当算法识别到相遇节点后,该函数将起点至相遇节点与相遇节点至目标点的两段路径拼接,得到最终完整路径。

2.2 IDWA

为应对传统 DWA 在局部最优问题上的不足,并提高局部路径规划与全局路径之间的配合程度,本研究提出 IDWA。该方法主要在以下 3 个方面进行优化:①轨迹评估函数优化;②动态障碍物的危险区域建模;③基于障碍物密度的自适应动态窗口优化。

2.2.1 轨迹评估函数优化

轨迹评估函数是 DWA 中用于筛选候选轨迹的重要部分,其作用是在多条预测轨迹中选择当前最优的运动轨迹。然而,传统 DWA 通常采用固定权重的加权评价方式,在简单场景下能够满足基本规划需求,但在复杂环境或动态障碍物较多的情况下,算法对环境变化的适应能力相对有限,并且往往需要反复进行参数调优。为增强算法的目标导向性,本研究在传统评价函数的基础上加入基于局部路径点的距离项,记为 $Dist_score(v, w)$,用于测量在时间 t 预测的轨迹终点 (x_t, y_t) 与局部目标点 (x_g, y_g) 之间的欧几里得距离。定义为:

$$Dist_score(v, w) = \sqrt{(x_g - x_t)^2 + (y_g - y_t)^2} \quad (12)$$

改进后的综合评价函数被重新定义为:

$$G(v, w) = Dist_score(v, w) + \frac{1}{Dist(v, w)} + Vel(v, w) + Head(v, w) \quad (13)$$

式中: $Dist(v, w)$ 为障碍物距离函数,用于描述机器人与周围障碍物之间的最小距离(取倒数是为了将“最大化间距”转化为“最小化代价”); $Vel(v, w)$ 为速度评价函数,用于衡量当前速度与最大可达速度之间的差值; $Head(v, w)$ 为航向角评价函数,用于计算预测终点方向与局部目标方向之间的角度偏差。

传统 DWA 方法通常需要根据不同场景对各个评价项

的权重进行反复调整。相比之下,本文方法通过将所有权重设置为 1 来简化实现。这样的经验性设置避免了在多评价项组合情况下进行大量重复实验,有助于提高算法部署和基准测试的效率。在多数常规应用场景中,例如仓库或办公室的室内巡检、商场或医院走廊中的服务机器人导航,以及障碍物分布较为规则的实验室环境,统一加权形式通常能够取得较为稳定的导航效果。然而,对于安全性要求较高或对运行效率有特殊要求的任务场景,不同评价项的重要性仍需进一步细化调整,这也是后续研究中需完善的内容。

算法在动态窗口内遍历所有满足动态约束和无碰撞条件的可行速度对,并分别计算其评价函数 $G(v, w)$ 。随后,选取评价函数值最小的速度对作为机器人的执行控制命令。表示为:

$$(v^*, w^*) = \underset{(v, w) \in V_{dw}}{\operatorname{argmin}} G(v, w) \quad (14)$$

2.2.2 动态障碍物的危险区域建模

传统 DWA 主要依据当前时刻障碍物的位置状态来选择速度控制量,对高速动态环境的预测能力不足。为克服这一不足,本研究设计了一种预测驱动的危险区域建模机制。该机制利用动态障碍物的速度信息对其未来位置进行估计,并在预测结果的基础上构建危险区域,使机器人能够提前识别可能发生碰撞的位置。图 2 为危险区域的定义,其中实线代表全局路径,虚线框表示动态障碍物,实线框表示危险区域,虚线箭头表示障碍物的运动方向。

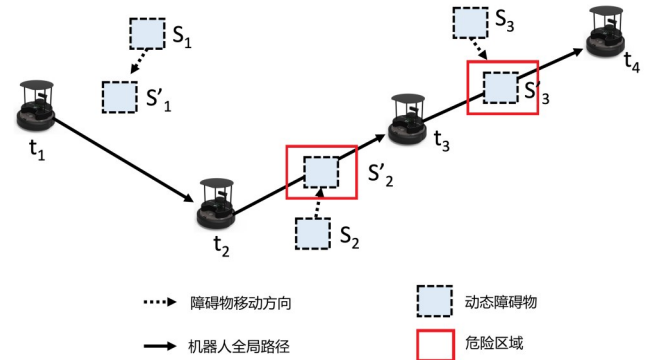


Fig. 2 Hazardous zone definition

图 2 危险区域定义

该机制主要包括以下 3 个步骤:

(1) 候选碰撞点预测。系统设置 1 s 的预测窗口,用于估计每个动态障碍物在短时间内可能经过的运动轨迹。在每一个采样时刻,系统会先从周围障碍物中选取与机器人距离最近的障碍物,然后进一步判断该障碍物的预测路径是否与全局路径产生交点。如果检测到二者存在交点,则将该位置记录为候选碰撞点,并将其作为后续风险评估的输入。候选碰撞点的定义如图 3 所示。

(2) 基于时间差 ΔT 的碰撞风险评估。为进一步量化机器人与动态障碍物之间的潜在碰撞风险,定义以下关键参数(见图 3):① T_{obs} 。表示障碍物到达预测碰撞点的时

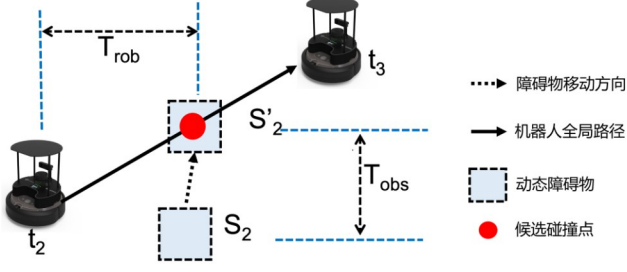


Fig. 3 Candidate collision point definition

图3 候选碰撞点的定义

间;② T_{rob} 。表示机器人到达同一碰撞点的时间;③ $\Delta T = |T_{obs} - T_{rob}|$ 。表示绝对时间差。

根据 ΔT 值进行风险分级:①安全区。如果 $\Delta T \geq 0.2$ s,说明机器人与障碍物发生碰撞的可能性较低,通常不需要进行路径调整。例如在图2中,障碍物从 S_1 到 S_1' 的运动不会与机器人路径冲突;②碰撞风险。如果 $\Delta T < 0.2$ s,则存在显著碰撞风险。例如在图2中,障碍物 S_2 和 S_3 在分别接近 S_2' 、 S_3' 时会在0.2 s内与机器人轨迹重合,构成了潜在威胁。

(3)危险区域划分。若障碍物预测轨迹与机器人未来路径发生交点,且候选点被评估为高风险碰撞点,则根据障碍物运动方向划分危险区域:①上行运动。危险区域位于全局路径上方(例如从 S_2 到 S_2');②下行运动。危险区域位于全局路径下方(例如从 S_3 到 S_3')。

该机制不仅提升了实时性和安全性,而且在保证全局路径连续性的同时有效减少了路径长度和整体规划时间。

2.2.3 基于障碍物密度的自适应动态窗口优化

为实现复杂环境下的安全与高效导航,本研究提出一种基于障碍物密度的自适应动态窗口优化策略。该策略通过引入障碍物密度指标 ρ 对动态窗口的搜索范围进行自适应调节,从而在不同环境条件下实现速度与安全性的动态平衡。其核心机制为:

(1)动力学调节系数设定。引入动力学调节系数 $k_v = 0.8$ 和 $k_w = 0.8$,随密度 ρ 对原始最大线速度与角速度 (v_{max}, w_{max}) 进行连续缩放。

(2)密度驱动的动态调整与下限约束。为避免在高密度环境下速度衰减至零,分别设置线速度与角速度的最小阈值 (v_{min}, w_{min}) ,对应的自适应上界分别为:

$$v'_{max}(\rho) = \max[v_{max} \cdot (1 - k_v \cdot \rho), v_{min}] \quad (15)$$

$$w'_{max}(\rho) = \max[w_{max} \cdot (1 - k_w \cdot \rho), w_{min}] \quad (16)$$

(3)结合加速度约束构造动态窗口。设当前速度为 (v, w) ,最大线/角加速度为 (a_v^{max}, a_w^{max}) ,控制周期为 Δt 。依据DWA,先由加速度约束得到可达区间,再与式(15)、式(16)给出的密度自适应上界/下界相交,形成最终可行窗口。表示为:

$$v^- = \max(v_{min}, v - a_v^{max} \Delta t), v^+ = \min(v'_{max}(\rho), v + a_v^{max} \Delta t) \quad (17)$$

$$w^- = \max(w_{min}, w - a_w^{max} \Delta t), w^+ = \min(w'_{max}(\rho), w + a_w^{max} \Delta t) \quad (18)$$

为便于实现,定义信息向量为:

$$v_{info} = [v_{min}, v'_{max}, w_{min}, w'_{max}] \quad (19)$$

由加速度约束给出的可达区间向量为:

$$v_{move} = [v^-, v^+, w^-, w^+] \quad (20)$$

二者的交集即为可行窗口。

(4)最终速度生成。在二维可行窗口 $[v^-, v^+] \times [w^-, w^+]$ 内按固定步长采样候选对 (v_i, w_i) ,并采用标准DWA评价函数 $G(v, w)$ (式13)进行打分,选择得分最小的一组作为输出。表示为:

$$(v_f, w_f) = \arg_{v \in [v^-, v^+], w \in [w^-, w^+]} \max G(v, w) \quad (21)$$

该策略通过对速度空间进行障碍物密度驱动的自适应调节,实现了稀疏环境下的高效运动与拥挤环境下的安全约束。

综上所述,IDWA算法的伪代码为:

输入:障碍物密度信息 obstacle_density

输出:局部最优路径 best_traj

```

1.  $vw \leftarrow AdjustVW(obstacle\_density)$ 
2.  $min\_score \leftarrow \text{float}("inf")$ 
3. FOR  $v \in \text{range}(vw[1])$  DO
4.   FOR  $w \in \text{range}(vw[2])$  DO
5.      $craj \leftarrow \text{cal.traj}(v, w)$ 
6.      $craj\_score \leftarrow d\_s + v\_s + t\_s + h\_s$ 
7.      $intersect.time \leftarrow \text{closest\_obs collision time}$ 
8.      $t\_end \leftarrow \text{robot collision time}$ 
9.     IF  $|intersect.time - t\_end| < 0.2$  THEN
10.       $direction \leftarrow \text{direction of closest\_obs}$ 
11.      IF  $direction$  is up THEN
12.         $dangerous\_area \leftarrow \text{upper part}$ 
13.      ELSE
14.         $dangerous\_area \leftarrow \text{lower part}$ 
15.      ENDELSE
16.    ENDELSE
17.  ELSE
18.     $dangerous\_area \leftarrow \emptyset$ 
19.  ENDELSE
20. ENDFOR
21. IF  $craj \in dangerous\_area$  THEN
22.   $craj\_score \leftarrow \text{float}("inf")$ 
23. ENDFOR
24. IF  $min\_score > craj\_score$  THEN
25.   $best\_traj \leftarrow craj$ 
26. ENDFOR
27. ENDFOR
28. RETURN  $best\_traj$ 

```

该算法的第1行基于障碍物的密度动态调整速度窗口的大小;第3-4行遍历调整后窗口内的所有候选速度组

合;第6行计算每个候选轨迹的综合评分函数,其中 d_s 为新增加的距离评价子函数, v_s 为速度评价函数, t_s 为障碍物距离评价函数, h_s 为角度评价函数。第9-20行为执行危险区域判定的代码段。

传统的全局路径规划算法(如 A* 算法)通常能在已知地图中搜索出一条无碰撞路径,但该方法更多关注静态环境下的路径生成。局部路径规划方法 DWA 虽然具有较好的实时避障能力,但由于其主要依赖局部信息进行决策,缺少对全局路径的整体感知,在复杂环境中容易出现局部最优问题。为克服这一互补局限,本研究提出集成的 IBA-IDWA 算法,该算法采用 IBA 进行全局路径规划,采用 IDWA 执行局部实时避障。针对全局路径过于稀疏而导致的实际偏差问题,本研究引入关键点密集化策略,在相邻关键节点之间采用线性插值生成额外的中间目标点,并将这些目标点作为 IDWA 的局部导航目标。该机制能引导机器人在避障过程中更加贴近全局最优轨迹,同时保持对动态障碍物的适应能力,进而在安全性、实时性与路径最优性之间取得较好平衡。

3 实验方法与结果分析

3.1 实验设置

为验证 IBA-IDWA 算法在全局与局部路径规划中的性能表现,本研究设计并开展较为系统的仿真实验。实验运行环境均为 Python 3.9.12,硬件平台为 Intel i5-8250U CPU。实验主要包括两个部分:①消融实验。用于单独分析各项改进策略对算法性能的影响;②比较实验。通过与现有经典方法进行比较,进一步说明所提算法的综合性能表现。

为提高实验结果的稳定性和统计可靠性,本研究在 1000 个随机生成的 40×40 网格地图上进行测试,并记录平均性能指标。实验设置为:①起始位置和目标位置分别以小五角星与大五角星表示;②方格表示静态障碍物,圆形表示动态障碍物。在实验环境中,动态障碍物被设置为在整个地图范围内进行匀速直线运动,其速度在两个区间内分配,即(1~2) m/s 或(2~3) m/s。每次实验中,动态障碍物的初始位置和速度均随机生成,但其运动方向保持不变,且不具备自主决策能力,因此可以将其视为可预测的匀速移动体。这样的设置既能够模拟动态环境中常见的移动障碍物,又能保证不同实验之间条件相对一致,便于后续结果的复现与分析。

DWA 参数设置:最大线速度为 2 m/s,最大线加速度为 1.5 m/s²,线速度分辨率为 0.1 m/s,最大角速度为 1 rad/s,最大角加速度为 2.1 rad/s²,角速度分辨率为 0.2 rad/s,预测时间范围为 2 s。性能评估指标包括:规划时间(s)、路径长度(m)、成功率(达到目标的试验百分比)、最短路径偏差 p (计算公式为式 22)、路径长度方差 s (计算公式为式 23)。

其中,成功率的判定标准为机器人最终位置与目标点的欧几里得距离不超过 0.1 m。

$$p = \sqrt{\frac{1}{n} \sum_{i=1}^n (L_i - L)^2} \quad (22)$$

$$s = \frac{1}{n-1} \sum_{i=1}^n (L_i - \bar{L})^2 \quad (23)$$

式中: n 为实验次数; L_i 为第 i 次试验的实际路径长度; L 为理论最小路径长度(通过起点与目标点之间的欧几里得距离近似计算); $\bar{L}$ 为平均路径长度。

3.2 消融实验

为评估 IBA 和 IDWA 的独立贡献,本研究设计以下 6 个消融模型:① A* + DWA;② 双向 A*(BA*) + DWA;③ IBA + DWA;④ A* + IDWA;⑤ BA* + IDWA;⑥ IBA + IDWA (本文方法)。消融实验结果见表 2。首先在保持局部规划器不变的前提下比较 A*、BA* 与 IBA 3 种全局规划器的性能差异。以 DWA 作为局部规划器时,与传统 A*+DWA 相比,BA*+DWA 的规划时间缩短 2.51 s,路径长度减少 1.66 m,成功率提升 3.20%;而 IBA+DWA 的规划时间相较于 A*+DWA 缩短 9.84 s,路径长度减少 2.15 m,成功率提升 12.60%。这表明双向搜索机制(BA*)主要提升了搜索效率,而引入障碍物密度自适应启发式函数与冗余节点消除策略(IBA)后,算法在路径最优性、规划效率和鲁棒性方面均获得显著改善。其次,在保持全局规划器不变的前提下比较 DWA 与 IDWA 的性能差异,以传统 A* 算法作为全局规划器时,A*+IDWA 的规划时间相较于 A*+DWA 缩短 23.69 s,路径长度减少 0.68 m,成功率提升 12.70%;以 IBA 为全局规划器时,IBA+IDWA 的规划时间相较于 IBA+DWA 缩短 17.83 s,路径长度减少 0.84 m,成功率提升 4.80%。这表明 IDWA 通过轨迹评估函数优化、危险区域建模与自适应速度窗口调整显著提升了局部规划的实时响应能力与安全性;最后,综合比较各项改进措施的独立贡献。在规划时间方面,IDWA 带来的提升更为显著,与 A*+DWA 相比,A*+IDWA 的规划时间降幅达 28.81%,IBA+DWA 的规划时间降幅为 11.97%;在路径长度优化方面,IBA 的贡献更为突出,IBA+DWA 的路径长度相较于 A*+DWA 减少 2.15 m,而 A*+IDWA 的路径长度相较 A*+DWA 仅减少 0.68 m;在成功率提升方面,两者贡献相近,IDWA 独立提升约 12.70%,IBA 独立提升约 12.60%。本文方法(IBA+IDWA)将二者融合,在各项指标上的表现更为均衡,不仅具有较高的规划效率,而且能获得更短的路径长度和更高的成功率。

3.3 比较实验

3.3.1 全局规划比较实验

为全面评估全局规划性能,将 IBA 算法与 5 种基线方法进行比较,分别为传统 A* 算法、双向 A* 算法、RRT* (Rapidly-exploring Random Tree Star) 算法以及文献[20]与[21]中提出的改进 A* 算法。其中,RRT* 是一种基于采样的全局路径规划算法,通过引入渐进最优性机制,使算

Table 2 Ablation experiment results

表2 消融实验结果

消融模型	规划时间/s	路径长度/m	成功率/%
A*+DWA	82.23	58.59	73.60
BA*+DWA	79.72	56.93	76.80
IBA+DWA	72.39	56.44	86.20
A*+IDWA	58.54	57.91	86.30
BA*+IDWA	57.44	56.02	86.30
IBA+IDWA	54.56	55.6	91.00

法在随机采样过程中逐步收敛到最优解,常用于大规模复杂环境下的路径规划。图4为不同算法在静态环境下的规划路径结果。可以看出,IBA 相较基线方法展现出显著优势,包括更短的路径长度、更少的扩展节点、更平滑的轨迹以及更小的转角幅度。尤其是如图4(d)所示,在某些复杂环境中,仅有IBA算法和RRT*算法成功生成可行路径,而其他方法均未能成功。

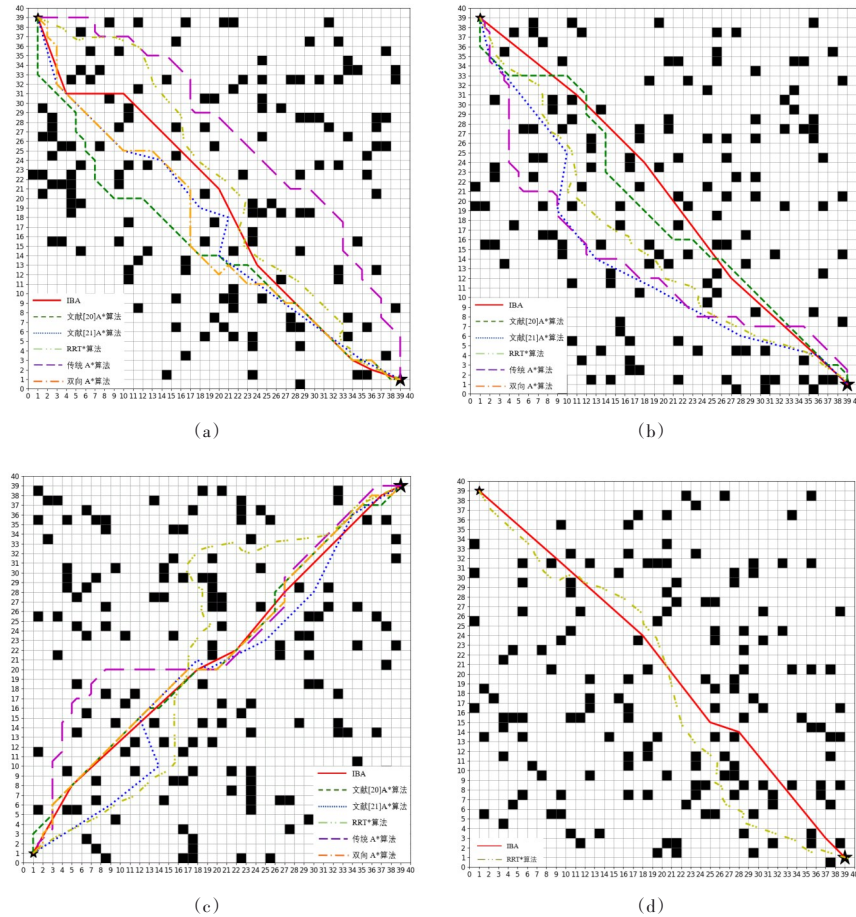


Fig. 4 Planning path results of different algorithms in a static environment

图4 不同算法在静态环境下的规划路径结果

如表3所示,与传统A*算法与RRT*算法相比,IBA算法的平均路径长度分别减少7.01 m和3.68 m。与文献[20]方法相比,IBA算法的节点减少43个;与RRT*算法相比,其节点减少37个。IBA算法实现了更低的最大偏差值,相较文献[20]和[21]方法分别降低3.42和4.07。IBA算法的成功率为97.70%,相较传统A*算法与RRT*算法分别提高14.07%和7.78%。综上所述,IBA算法在路径最优性、计算效率与鲁棒性方面均优于基线方法。

3.3.2 融合算法比较实验

将IBA-IDWA融合算法与文献[20]、[21]和[22]提出的融合算法进行比较。图5为各算法在动态障碍物速度低(1~2 m/s 范围内随机设定)且数量少(10~12个)情况下的图6为各算法在动态障碍物速度高(2~3 m/s 范围内随

Table 3 Global planning algorithm performance comparison

表3 全局规划算法性能比较分析

算法	平均路径长度 度/m	平均节点个数	最短路径偏差	成功率/%
A*	64.31	111	11.03	85.65
双向A*	61.13	51	7.50	88.55
文献[20]A*	60.65	50	7.08	93.90
文献[21]A*	60.59	12	7.73	94.85
RRT*	60.98	44	7.01	90.65
IBA	57.30	7	3.66	97.70

路径规划结果。此时整体环境的动态性弱,各算法均能在保持高安全性的前提下完成路径规划,其中IBA-IDWA算法表现出更优的避障性能。

机设定)且数量多(15~18个)情况下的路径规划结果。此

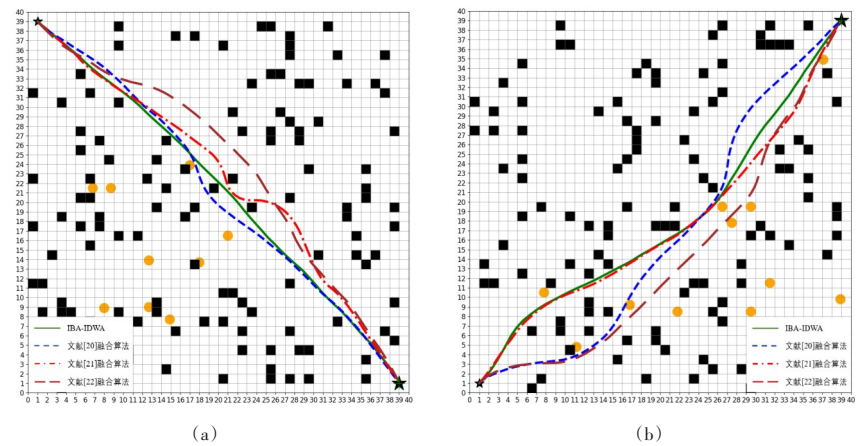


Fig. 5 Comparison of path planning results of various fusion algorithms in low dynamic environments

图5 低动态环境下各融合算法路径规划结果比较

时环境动态性显著增强,高速移动的障碍物使各算法的避障难度与计算复杂度均大幅提升。可以看出,文献[20]与[21]的融合算法在面对密集高速动态障碍物时的实际运动轨迹与全局最优路径的偏离程度明显增大,部分轨迹在障碍物交汇区域出现绕行与振荡现象,这主要是由于其局部规划器缺少有效的动态障碍物预测机制,导致只能在障碍物接近时才作出反应。文献[22]算法的整体路径相对平稳,但在多个动态障碍物同时出现的区域局部最优问题

依然存在。相比之下,IBA-IDWA 算法生成的轨迹更加贴近全局路径,在穿越动态障碍物时能够保持较为平滑的行驶方向,未出现大幅绕行。综上所述,IBA-IDWA 算法在高动态环境下不仅路径长度更短,而且避障成功率显著优于对照算法。而对照算法由于全局规划效率低或局部规划缺少动态预测与自适应调节机制,在复杂动态环境中容易出现轨迹偏离、路径振荡甚至避障失败等问题。

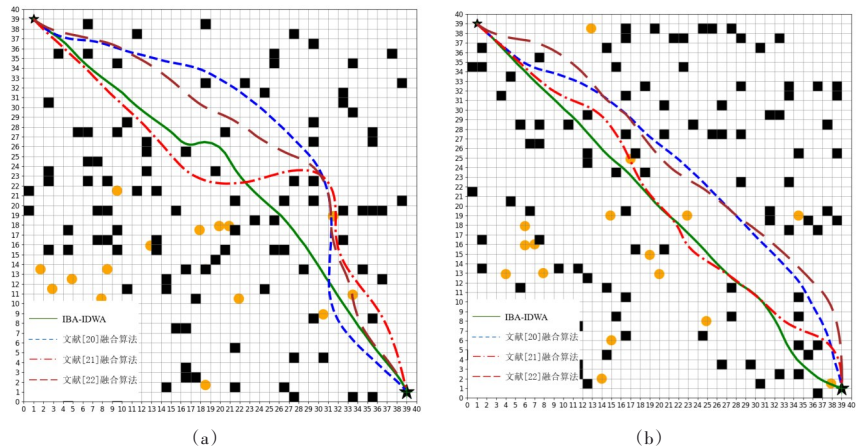


Fig. 6 Comparison of path planning results of various fusion algorithms in high dynamic environments

图6 高动态环境下各融合算法路径规划结果比较

具体来说,如表 4 所示,在低速、少障碍物的条件下,IBA-IDWA 算法的平均路径长度相较文献[20]、[21]和[22]融合算法分别减少 2.07 m、1.28 m、0.63 m;最短路径偏差分别减少 3.91、4.87 和 1.82;平均规划时间分别减少 22.63 s、11.34 s 和 9.91 s;成功率分别提高 10.28%、6.78% 和 5.04%;路径长度方差分别降低 5.04、3.51 和 2.16。

如表 5 所示,在高速、多障碍物的条件下,IBA-IDWA 算法的平均路径长度相较于文献[20]、[21]和[22]融合算法分别减少 3.48 m、2.71 m 和 0.80 m;最短路径偏差分别减少 3.38、4.47 和 1.31;平均规划时间分别减少 29.83 s、16.15 s 和 9.20 s;成功率分别提高 13.38%、12.53% 和 6.01%;路径

Table 4 Performance comparison of fusion algorithms under low-speed and sparse obstacle conditions

表 4 低速少障碍物条件下各融合算法性能比较

算法	平均路径 长度/m	最短路径 径偏差	平均规划 时间/s	成功率/%	路径长 度方差
文献[20]融合算法	57.43	6.98	72.22	85.92	7.29
文献[21]融合算法	56.64	7.94	60.93	88.73	5.76
文献[22]融合算法	55.99	4.89	59.50	90.20	4.41
IBA-IDWA	55.36	3.07	49.59	94.75	2.25

长度方差分别降低 5.76、3.01 和 2.05。

综上所述,IBA-IDWA 算法在简单与复杂环境下的关键性能指标均优于基线方法,展现出更优的导航能力。

Table 5 Performance comparison of fusion algorithms under high-speed and dense obstacle conditions

表5 高速多障碍物条件下各融合算法性能比较

算法	平均路径 长度/m	最短路 径偏差	平均规划 时间/s	成功率/%	路径长 度方差
文献[20]融合算法	61.38	7.02	85.30	79.60	9.00
文献[21]融合算法	60.61	8.11	71.62	80.20	6.25
文献[22]融合算法	58.70	4.95	64.67	85.13	5.29
IBA-IDWA	57.90	3.64	55.47	90.25	3.24

4 结语

针对动态环境下自主移动机器人的导航问题,本研究提出一种集成式全局-局部路径规划算法 IBA-IDWA。该算法将基于双向搜索的全局规划器与改进型动态窗口局部控制器相结合,使机器人在路径规划过程中能够较好地兼顾规划效率、运行安全性与路径最优性。在静态与动态场景下的大量比较实验与消融实验结果表明,该算法在导航成功率、路径长度以及避障性能方面均取得显著提升,能满足复杂和不确定环境下的导航需求,在实际移动机器人系统中具有一定的应用潜力。

然而, IDWA 采用统一权重的评价策略,这种方式更适合一般导航任务。当应用场景对安全性或效率存在明显偏向时,统一权重策略可能难以同时兼顾安全性与效率。针对这一问题,后续研究可引入自适应加权机制,使算法能够根据不同任务需求动态调整评价权重,从而获得更加灵活的性能平衡。此外,本研究建立在障碍物类型和配置已知的基础上,而真实环境中的障碍物通常具有更强的多样性和复杂性。因此后续工作还需要在真实动态环境中对 IBA-IDWA 算法进行进一步部署与验证,重点考虑未知障碍物形状、非确定性运动模式以及环境不确定性等问题,以进一步提升算法的鲁棒性和适用范围。

参考文献:

- [1] WANG L F, ZHANG Y D, GUO C L, et al. 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.
- [2] CUI X J, WANG X J, LI X H. Improved RRT algorithm for robot global path planning [J]. Computer Engineering and Applications, 2025, 61(4): 331-338.
崔锡杰, 王晓军, 李晓航. 改进 RRT 算法的机器人全局路径规划 [J]. 计算机工程与应用, 2025, 61(4): 331-338.
- [3] NOVOSAD M, PERICKA R, VONASEK V, et al. CTopPRM: clustering topological PRM for planning multiple distinct paths in 3D environments [J]. IEEE Robotics and Automation Letters, 2023, 8(11): 1-8.
- [4] DAI Y, LI W J, LI S K, et al. Improving the lifelong planning A-star algorithm to satisfy path planning for space truss cellular robots with dynamic obstacles [J]. Robotica, 2025, 43(4): 1243-1257.
- [5] XU B Y. Precise path planning and trajectory tracking based on improved A-star algorithm [J]. Measurement & Control, 2024, 57(8): 1.
- [6] XIE J, XU C Y, YANG Q X, et al. Robot path planning model based on

improved A* algorithm [J]. International Journal of Advanced Computer Science Applications, 2025, 16(5): 228-242.

- [7] FU X, HUANG Z C, ZHANG G X, et al. Research on path planning of mobile robots based on improved A* algorithm [J]. PeerJ Computer Science, 2025, 11: e2691.
- [8] AN Z H, RUI X P, GAO C J, et al. Improved A* navigation path-planning algorithm based on hexagonal grid [J]. ISPRS International Journal of Geo-Information, 2024, 13(5): 166.
- [9] LI C G, HUANG X, DING J, et al. Global path planning based on a bi-directional alternating search A* algorithm for mobile robots [J]. Computers in Industry Engineering, 2022, 168(Suppl 1): 108123.
- [10] ALSHAMMERY S, BOUBAKER S, KOLSI L, et al. Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance [J]. Computer Modeling in Engineering & Sciences, 2022, 72(3): 5939-5954.
- [11] MA Q L, YANG R Q, LIAN T J, et al. Path planning for mobile robots based on improving Dijkstra algorithm [C]//Huzhou: Proceedings of the International Conference on Cloud Computing, Performance Computing, and Deep Learning, 2023.
- [12] ZHOU R, ZHOU K, WANG L N, et al. An improved dynamic window path planning algorithm using multi-algorithm fusion [J]. International Journal of Control, Automation, and Systems, 2024, 22(3): 1005-1020.
- [13] DAI X Q, LIU C Y, LAI Q, et al. The local path planning algorithm for amphibious robots based on an improved dynamic window approach [J]. Journal of Marine Science and Engineering, 2025, 13(3): 399.
- [14] SU Y X, XIN J Y, SUN C Y, et al. Dynamic path planning for mobile robots based on improved RRT* and DWA algorithms [J]. IEEE Transactions on Industrial Electronics, 2025, 72(10): 10595-10604.
- [15] QI B K, LI M Q, YANG Y, et al. Research on UAV path planning obstacle avoidance algorithm based on improved artificial potential field method [J]. Journal of Physics: Conference Series, 2021, 1948: 012060.
- [16] SUO Y F, CHEN X Y, YUE J, et al. An improved artificial potential field method for ship path planning based on artificial potential field-minimized customary navigation routes [J]. Journal of Marine Science and Engineering, 2024, 12(5): 731.
- [17] GAN X L, HUO Z H, LI W, et al. DP-A*: for path planning of UGV and contactless delivery [J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(1): 907-919.
- [18] LIU Y J, WANG C, WU H, et al. Mobile robot path planning based on kinematically constrained A-star algorithm and DWA fusion algorithm [J]. Mathematics, 2023, 11(21): 4552.
- [19] GUO T, SUN Y Q, LIU Y, et al. An automated guided vehicle path planning algorithm based on improved A* and dynamic window approach fusion [J]. Applied Sciences, 2023, 13(18): 10326.
- [20] YANG H X, TENG X Q. Mobile robot path planning based on enhanced dynamic window approach and improved A* algorithm [J]. Journal of Robotics, 2022, 2022: 2183229.
- [21] ZHANG J F, GUO J L, ZHU D X, et al. Dynamic path planning fusion algorithm with improved A* algorithm and dynamic window approach [J]. International Journal of Machine Learning and Cybernetics, 2025, 16(3): 2057-2071.
- [22] ZHANG Y, LI B H, HUO T, et al. Research on robot dynamic obstacle avoidance by integrating improved A* and DWA algorithms [J]. Simulation Modelling Practice and Theory, 2025, 37(6): 1555-1564.
张艳, 李炳华, 霍涛, 等. 融合改进 A* 算法与 DWA 算法的机器人动态避障方法研究 [J]. 系统仿真学报, 2025, 37(6): 1555-1564.

(责任编辑:尹晨茹)