

密集障碍物环境下改进 A^* -DWA 融合导航算法研究^{*}

杨国亮 李邕腾 翁达里

(江西理工大学电气工程与自动化学院 赣州 341400)

摘要: 针对博物馆、展览馆等密集障碍物环境下腿脚不便群体的移动辅助需求,提出一种融合全局 A^* 算法与局部动态窗口法 DWA 的自主导航路径规划方法。首先通过引入启发式函数优化与路径平滑策略设计了改进 A^* 算法,利用该算法生成全局最优路径;然后针对动态环境特性采用自适应权重 DWA 算法进行实时避障与轨迹优化,在原始 DWA 算法的基础上加入全局路径评价函数和制动距离评价函数,显著提升复杂狭窄通道中的安全性与运动连续性。为验证算法有效性,在模拟展馆环境的实验室场景中构建测试平台。并引入轨迹平滑度、碰撞率及遍历节点数等指标对改进算法进行评估。实验结果表明,相较于传统 A^* 与原始 DWA 算法,本文方法在保证全局路径最优性的同时,不仅提高了动态障碍规避成功率,轨迹抖动幅度也有所降低。该方法为特殊群体在受限室内场景的安全移动提供了可靠技术支撑。

关键词: 密集障碍物环境;路径规划; A^* 算法;局部动态窗口法;安全移动

中图分类号: TP242.6;TN964 **文献标识码:** A **国家标准学科分类代码:** 510.8050

Research on improved A^* -DWA fusion navigation algorithm in dense obstacle environment

Yang Guoliang Li Zhiteng Weng Dali

(School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341400, China)

Abstract: Targeting the mobility assistance needs of people with walking difficulties in environments dense with obstacles like museums and exhibition halls, this study proposes an autonomous navigation path planning method. The method combines a global A^* algorithm with a local dynamic window approach DWA. First, an improved A^* algorithm designs an optimized heuristic function and a path smoothing strategy. This improved algorithm generates the global optimal path. Second, the study employs an adaptive weighting DWA algorithm to handle dynamic environments. This DWA variant adds a global path evaluation function and a braking distance evaluation function to the original DWA. These additions significantly enhance safety and motion continuity within complex, narrow passages. To verify effectiveness, the study constructs a test platform simulating a museum environment in a laboratory setting. Evaluation utilizes metrics including trajectory smoothness, collision rate and number of traversed nodes. Experimental results demonstrate that compared to traditional A^* and original DWA algorithms, this method achieves global path optimality. It also increases the dynamic obstacle avoidance success rate. Furthermore, the method reduces trajectory jitter amplitude. This approach provides reliable technical support for the safe movement of special groups in constrained indoor environments.

Keywords: dense obstacle environments; path planning; A^* algorithm; local dynamic window approach; safe navigation

0 引言

随着无人车与移动机器人在工业制造、物流配送、医疗康复以及博物馆、展馆等场景的快速落地,机器人在密集障碍环境中实现安全、高效的自主导航已成为重要研究课

题^[1]。此类场景往往同时存在静态障碍与动态障碍,通道狭窄、障碍密度高,对路径规划与实时避障算法的可靠性与实时性提出了更高要求^[2]。现有移动机器人路径规划方法大体可分为 3 类:基于图搜索的启发式全局规划、采样/优化类方法以及局部避障方法^[3]。

收稿日期:2025-09-14

^{*} 基金项目:江西省教育厅科技计划项目(GJJ210861)、江西省教育厅科技项目(GJJ200879)资助

全局规划方面, A* 算法(A-star algorithm, A*) 因其在栅格地图上求解最短路径具有理论最优性与确定性而被广泛采用, 但传统栅格化 A* 在路径离散化时易产生转角多、平滑性差的问题, 且难以充分考虑机器人几何尺寸与运动学约束^[4]。为改善路径质量与可行性, 研究者提出了多种改进与变体, 如任意角 A* 搜索算法(any-angle A* search, AA*)、Theta* 算法(theta algorithm, Theta*) 以及跳点搜索算法(jump point search, JPS)等, 虽然改善了路径粗糙度与搜索效率, 但多数方法仍以质点为假设, 未充分考虑机器人尺寸、运动学与在高密度障碍下的可行性限制^[5]。采样/优化类方法(如快速扩展随机树(rapidly-exploring random tree, RRT)、概率路图(probabilistic roadmap, PRM)以及协变哈密顿优化运动规划(covariant hamiltonian optimization for motion planning, CHOMP))等能够处理连续空间与复杂约束, 但在对初始解敏感、计算量较大或需要高频环境更新时仍存在一定局限^[6]。局部避障方法中, 动态窗口法(dynamic window approach, DWA)、速度障碍(velocity obstacles, VO)、最优互惠避碰(optimal reciprocal collision avoidance, ORCA)与模型预测控制(model predictive control, MPC)等在实时避障与动态交互方面表现较好, 但“短视性”可能导致陷入局部最优或振荡, 无法从整体上保证轨迹的全局性和安全裕度^[7]。

为兼顾全局可达性与局部实时性, 学术界与工程实践中常采用“全局—局部”分层框架, 例如以 A* 提供全局参考路径、以 DWA 完成局部避障与跟踪^[8]。然而在密集障碍或狭窄通道环境中, 该简单耦合策略仍可能暴露出若干问题: 全局路经常不考虑障碍分布密度和狭窄通道, 容易产生靠近障碍或不可执行的窄缝段^[9]; 层联式切换处会引发轨迹抖动且全局与局部评价不一致, 影响平滑性与跟踪精度^[10]; 局部规划器对全局参考的依赖弱且对动态障碍的权衡固定, 易陷入局部最优^[11]或频繁重规划引起振荡^[12]; 此外, 实时性、鲁棒性与安全裕度之间缺乏明确的多目标权衡机制, 参数对特定场景的敏感性也限制了方法的推广应用^[13]。针对上述问题, 邓云峰等^[14]提出改进 A* 与 DWA 的融合路径规划思路; 朱洪波等^[15]提出分层平滑优化 A* 引导 DWA 以改善路径质量; 同时, 宋俊辉等^[16]提出在动态环境下融合采样与势场方法以提升复杂环境的可行性与安全性。

基于此, 本文提出一种面向密集障碍场景的改进 A* - DWA 分层融合方法: 在 A* 评价函数中引入障碍密度^[17]、通行约束项并结合邻域去冗余与平滑策略以降低转角与提升可行性^[18]; 在 DWA 阶段评价函数中加入全局路径引导与基于安全距离的约束项, 并通过速度平滑机制减小速度突变, 从而实现局部避障的实时性与全局一致性的兼顾。实验结果表明, 该方法在密集障碍场景下能够有效提升路径平滑度与避障安全性, 并改善局部跟踪稳定性。

1 基于改进 A* 算法路径规划

1.1 传统 A* 算法描述

A* 算法是一种常见、高效的启发式搜索算法, 用于在加权图或网络中找到从起点到终点的最优路径。其结合了 Dijkstra 算法和广度优先搜索算法的优点, 增加了启发函数 $h(n)$ 的引导, 核心机制在于为每个节点构建融合实际代价与启发式预估的综合评估函数, 并依据该函数值动态确定节点优先级。A* 算法通过迭代扩展当前优先级最高的节点, 逐步收敛至全局最优路径, 从而高效获取代价最低且距离最短的可行解。A* 算法总代价函数如式(1)所示。

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

式中: $g(n)$ 为从起点 S 到当前节点 n 的实际路径代价值; $h(n)$ 为当前节点 n 到终点 G 的预估代价值; $f(n)$ 为从起点 S 到终点 G 的估计代价值。常见的计算预估代价值方法有曼哈顿距离、欧几里得距离和切比雪夫距离, 本文所用算法均使用欧几里得距离作为预估代价。欧几里得距离公式为:

$$d = \sqrt{(x_g - x_n)^2 + (y_g - y_n)^2} \quad (2)$$

传统 A* 算法的邻域扩展策略主要分为 4 方向搜索与 8 方向搜索两种模式。本文采用 8 方向邻域扩展策略的 A* 算法。该策略在传统 4 主轴方向基础上, 增加了 4 个对角线方向, 左上、左下、右上、右下。相应地, 机器人运动的最小转向角度为 $\pi/4$ 。移动步长依据方向动态调整, 主轴方向步长为 1 个栅格单位, 对角线方向步长为 $\sqrt{2}$ 个栅格单位。

传统 A* 算法寻路过程中主要依赖两个表来记录节点, 分别是 OpenList 和 CloseList 列表, 其中 OpenList 储存寻路过程中遍历过的所有子节点, 而 CloseList 储存的是具有最小代价函数 $f(n)$ 值的子节点, 并以该节点作为规划轨迹节点, 直到找到目标点。

1.2 A* 算法存在问题及改进策略

传统 A* 算法在障碍物密集环境中存在路径贴近危险区域、难以主动规避高密度障碍物区, 以及轨迹平滑度较差的问题。针对路径贴近危险区域的问题, 本文采用基于障碍物感知的邻域节点筛选改进, 通过自适应剔除高风险子节点, 提升路径安全性; 针对规避高密度障碍物区不足的问题, 引入自适应障碍物密度评价函数, 在路径搜索评价中融入动态调整的障碍物密度因子, 引导全局路径远离密集障碍物; 针对轨迹平滑度较差的问题, 实施安全阈值约束下路径优化策略, 增加路径平滑度。

1) 基于障碍物感知的邻域节点筛选

为进一步提升原始 A* 算法规划路径的安全性和平滑性, 并提高算法效率, 本节在原始算法基础上引入了一种针对临近障碍物的节点筛选机制。其操作逻辑与图 1 和 2 所示场景紧密对应, 该机制的核心步骤为:

(1) 障碍物检测与初步标记

如图 1 和 2 中的蓝色☆所示,以当前节点 (x_{now}, y_{now}) 为中心,遍历其 8 个邻域子节点。若检测到某个邻域节点为障碍物 (x_{obs}, y_{obs}) ,则立即将该障碍物节点的标志位 $flag$ 置为 0,禁止其加入开放列表 OpenList。

(2) 障碍物坐标关系分类

如图 1 所示,若检测到的障碍物 (x_{obs}, y_{obs}) 满足 $x_{obs} = x_{now}$ 且 $y_{obs} = y_{now} \pm 1$,即障碍物位于当前节点的正上方或正下方,则将该障碍物节点坐标加入列表 out_1 。

如图 2 所示,若检测到的障碍物 (x_{obs}, y_{obs}) 满足 $y_{obs} = y_{now}$ 且 $x_{obs} = x_{now} \pm 1$,即障碍物位于当前节点的正左方或正右方,则将该障碍物节点坐标加入列表 out_2 。

(3) 候选节点二次筛选

对于当前待处理的候选节点 $node_{wait}$,其坐标记为 (x_{wait}, y_{wait}) ,且 $flag$ 已被初始化为 1,代表初步可通行。

基于 out_1 的过滤(对应图 1):检查 $node_{wait}$ 是否满足存在某个 $(x_{obs}, y_{obs}) \in out_1$,使得 $y_{wait} = y_{obs}$ 且 $x_{wait} = x_{obs} \pm 1$,即 $node_{wait}$ 与 out_1 中障碍物同 y 轴且在 x 轴相邻。若满足此条件,则将此 $node_{wait}$ 节点的 $flag$ 置为 0,禁止其加入 OpenList。如图 1 所示,位于障碍物 (x_{obs}, y_{obs}) 左右两侧的同 y 轴节点会被筛除。

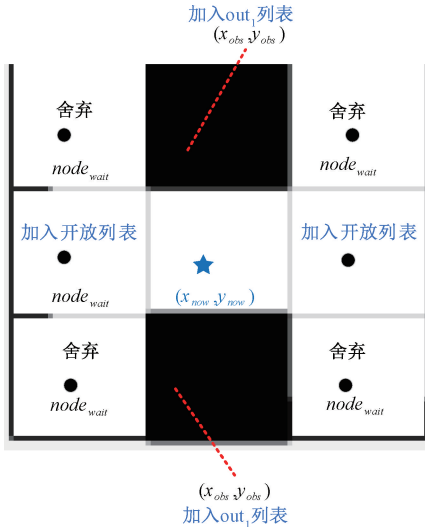


图 1 障碍物与当前点横坐标相同

Fig. 1 The obstacle has the same x-coordinate as the current point

基于 out_2 的过滤(对应图 2):检查 $node_{wait}$ 是否满足存在某个 $(x_{obs}, y_{obs}) \in out_2$,使得 $x_{wait} = x_{obs}$ 且 $y_{wait} = y_{obs} \pm 1$,即 $node_{wait}$ 与 out_2 中障碍物同 x 轴且在 y 轴相邻。若满足此条件,则将此 $node_{wait}$ 节点的 $flag$ 置为 0,禁止其加入 OpenList。如图 2 所示,位于障碍物 (x_{obs}, y_{obs}) 上下两侧的同 x 轴节点会被筛除。

(4) 加入开放列表:经过步骤(1)和(3)后,将剩余所有 $flag = 1$ 的 $node_{wait}$ 节点加入到 OpenList 中,供后续的 A* 算法主循环进行最优选择。

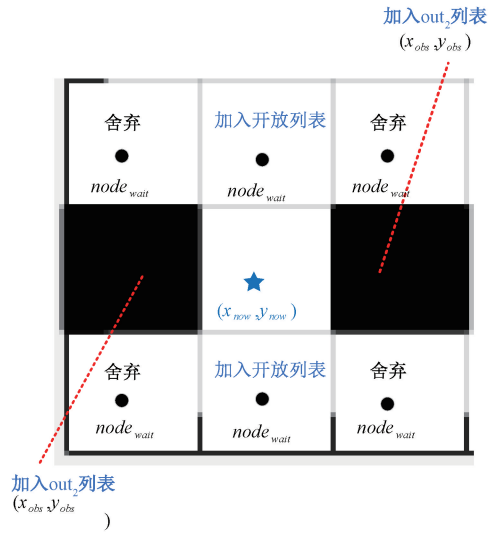


图 2 障碍物与当前点纵坐标相同

Fig. 2 The obstacle has the same vertical coordinate as the current point

该节点筛选机制通过精确识别并排除紧邻障碍物的特定方向节点,显著增强了路径规划的安全性,有效避免了路径紧贴障碍物边缘通过。不仅减少了后续路径优化计算量,也直接促使最终生成的路径更加平滑自然。

2) 自适应障碍物密度评价函数

原始 A* 倾向于生成紧贴障碍物的路径以追求理论最短距离,却忽视了密集障碍物区域固有的高碰撞风险与导航不确定性。这种短但险的路径在实际应用中往往不安全。为克服此局限并提升密集环境下的路径安全性与鲁棒性,本节改进将障碍物密度评价函数 P 引入 A* 算法的评价函数中。

在栅格地图下取当前点 (x_{now}, y_{now}) 和目标点 (x_{goal}, y_{goal}) 所包围的矩形范围(长 $x_{goal} - x_{now}$, 宽 $y_{goal} - y_{now}$),计算该矩形范围下所含障碍物个数占矩形面积的比例,即 $P = num_{obs} / [(x_{goal} - x_{now}) \times (y_{goal} - y_{now})]$,其中 num_{obs} 代表该矩形范围下所含障碍物的个数。

将 P 引入 A* 算法的评价函数中,得到一种新的评价函数 $F(n)$ 为:

$$F(n) = P \times g(n) + h(n) \times \exp(2 - P) \quad (3)$$

由式(3)可知,当障碍物密度较小时, P 比较小, $g(n)$ 的权重系数变小, $h(n)$ 的权重系数大幅增大,这使得评价函数 $F(n)$ 的取值更倾向于 $h(n)$,寻路时间更短,路径搜索效率更高,所遍历的节点更少。而当障碍物密度较大时, P 比较大, $g(n)$ 的权重系数变大, $h(n)$ 的权重系数降低,这使得评价函数 $F(n)$ 的取值倾向于 $g(n)$,导致寻路花费的时间更长,路径搜索效率更低,所遍历的节点更多,但搜索的也更加仔细了。

3) 安全阈值约束下路径优化策略

为降低 A* 全局路径中的冗余节点、提高轨迹平滑性

并满足 AGV 实际体积与安全缓冲要求,本文在 A^* 输出的栅格路径上提出一种基于安全阈值 d 的后处理优化策略。该策略在保证任一连通线段到最近障碍物的最小间隙不小于 d 的前提下,尽可能合并共线或可安全直连的中间节点,以减少轨迹关键点数和便于后续平滑与速度规划。如图 3 所示,网格为可通行格,黑色为障碍,细线为原始 A^* 路径,粗线为优化后路径,黑点为保留的关键节点。

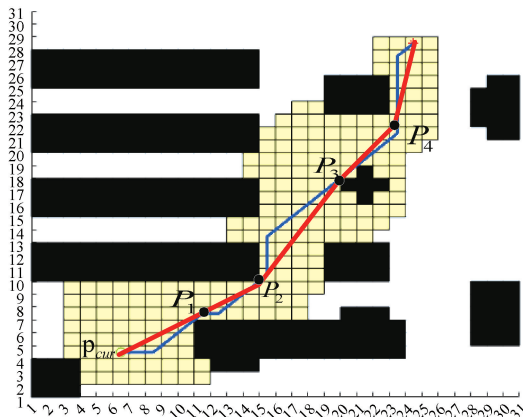


图 3 提取关键节点示意图(未加入安全阈值)

Fig. 3 Schematic diagram of key node extraction (without safety threshold)

图 3 为基于安全直连判定的迭代合并过程:对于非共线节点,以当前参考点 p_{cur} 为起点,依序尝试与后续节点 p_i 直接连线,计算该直线与地图中障碍单元的最小距离。若某条连接直线的最小距离大于阈值 d (此处设 $d = 0.2$),则判定 p_{cur} 与 p_i 可安全直连——将两端间的中间节点标为冗余节点并将参考点跳至 p_i ;反之,当最小距离不满足阈值时,保持当前最远的安全端点并停止对该方向的延伸。重复此过程直至抵达终点,最终将候选冗余节点全部剔除,得到保留关键节点的序列 P_1, P_2, P_3, P_4 。

综上所述,图 4 提出的基于阈值 d 的路径优化方法在仿真截图 4 中验证了其效果:在保证与障碍物最小安全间隙的条件下,显著减少了关键路径点并使路径更为平滑,便于后续轨迹跟踪与运动控制。

2 基于改进 DWA 算法路径规划

为确保行动不便群体在展览馆中安全、顺畅地观展,本章设计了具备自主避障能力的改进算法。由于场馆内存在流动人群等动态障碍物,改进的全局路径规划算法虽能生成起点至终点的静态路线,但缺乏实时避障功能,无法有效应对局部环境中突发的、移动的障碍物,威胁人车安全。

DWA 算法被引入以解决实时避障问题。然而,传统 DWA 算法在本应用场景中存在明显不足:1)局部优化视野容易导致轨迹偏离全局最优路径;2)评价函数通常采用固定权重优化,无法根据障碍物与观光车的实时相对运动

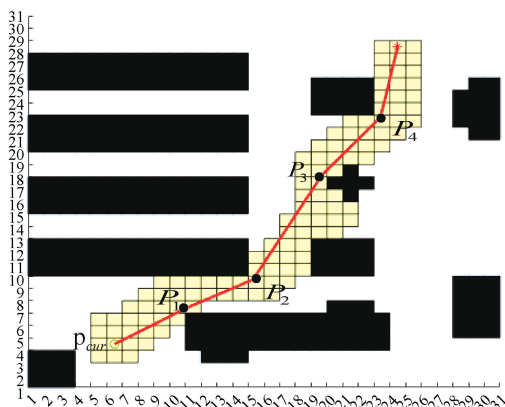


图 4 最终优化路径示意图(加入安全阈值)

Fig. 4 Schematic of the final optimized path (with safety threshold added)

关系进行自适应调整,导致在高风险场景下牺牲安全性;3)对观光车当前速度下所需的安全制动距离缺乏显式评估,在近距离障碍物突然出现时存在碰撞风险。

针对上述问题,本章对 DWA 进行了关键改进,提出基于多目标优化的自适应评价函数设计。其核心改进措施在于:1)引入全局路径评价函数,将局部规划轨迹与全局路径的偏离程度纳入评价体系,使局部规划更贴合全局最优路线。2)设定动态权重系数,动态调整速度评价函数权重与全局路径评价函数权重。当靠近障碍物时,提高避障安全性;并且大幅度提升全局路径贴合度。3)加入制动距离评价函数,评估观光车在当前行驶速度下所需的制动距离,并与到最近障碍物的实时距离进行比较,在评价函数中严厉惩罚那些制动距离大于安全距离的候选速度,为紧急避障提供理论保障。

改进后的自适应 DWA 算法流程如下:基于机器人运动学模型约束,在动态窗口内对线速度 v 和角速度 ω 进行离散采样,生成候选速度对集合;通过微分运动学推演各速度对在有限时域内的运动轨迹;利用集成了目标趋近度、自适应避障安全性、全局路径贴合度及制动安全距离等多目标的改进评价函数,对候选轨迹进行综合评价,筛选出最优轨迹。机器人最终执行该最优轨迹对应的 $v - \omega$ 控制量,实现位姿状态 $[x, y, \theta]^T$ 的动态调整,其运动方向与姿态由实时位移驱动和航向调节耦合决定。

2.1 机器人运动学模型分析

在 DWA 算法中,模拟机器人移动轨迹前需要先构建出机器人的运动学模型。本文采用阿克曼小车进行实验,其通过前轮转向角度控制车辆方向,并非全向移动的,只能前进和旋转。根据上述特点,构建出阿克曼小车的运行学模型如图 5 所示。

首先分别建立以底盘重心 o' 为原点,横坐标 x_{robot} ,纵坐标 y_{robot} 的底盘坐标系;以及以 o 为原点,横坐标 x ,纵坐标 y 的地面坐标系 xoy 。

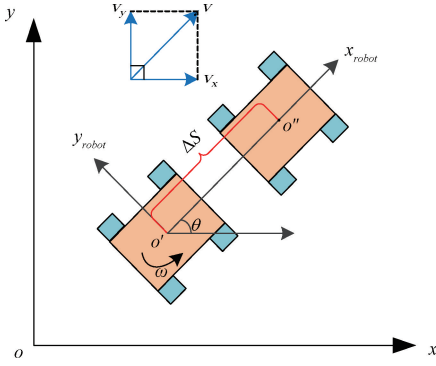


图 5 运动位置计算

Fig. 5 Motion position calculation

假设小车当前线速度和角速度分别为 v 和 ω , 小车当前位置 Q 为:

$$Q = (x, y, \theta) \quad (4)$$

在时间 Δt 内, 小车以当前线速度 v 移动距离为 ΔS :

$$\Delta S = v \times \Delta t \quad (5)$$

将移动距离 ΔS 映射到地面坐标系 xoy 上, 得到对于地面坐标系而言的相对变化坐标 Δx 和 Δy :

$$\Delta x = v \times \Delta t \times \cos \theta \quad (6)$$

$$\Delta y = v \times \Delta t \times \sin \theta \quad (7)$$

在原坐标基础上加上相对变化坐标, 可以得到小车经过 Δt 时间后, 下一状态的位置坐标:

$$x = x + \Delta x \quad (8)$$

$$y = y + \Delta y \quad (9)$$

$$x = x + v \times \Delta t \times \cos \theta \quad (10)$$

$$y = y + v \times \Delta t \times \sin \theta \quad (11)$$

$$\theta = \theta + \omega \times \Delta t \quad (12)$$

更新位置坐标 Q 为:

$$Q = (x, y, \theta) \quad (13)$$

2.2 速度矢量空间采样

实行 DWA 的第 1 步是进行速度矢量空间采样, 移动机器人的速度采样空间受硬件、机械结构和障碍物环境限制, 主要分 3 大方面约束:

1) 硬件条件限制速度空间 V_m 为:

$$V_m = \{(v, \omega) \mid v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (14)$$

2) 驱动电机力矩限制加减速速度空间 V_d 为:

$$V_d = \{(v, \omega) \mid v \in [v_c - v'_b \Delta t, v_c + v'_a \Delta t] \cap \omega \in [\omega_c - \omega'_b \Delta t, \omega_c + \omega'_a \Delta t]\} \quad (15)$$

式中: v_c, ω_c 分别为移动机器人当前线速度和角速度; v'_a, ω'_a 为最大加速度; v'_b, ω'_b 为最大减速度。

3) 环境障碍物限制速度范围 V_a :

$$V_a = \{(v, \omega) \mid v \leq \sqrt{2 \times \text{dist}(v, \omega) \times v'_b} \cap \omega \leq \sqrt{2 \times \text{dist}(v, \omega) \times \omega'_b}\} \quad (16)$$

式中: $\text{dist}(v, \omega)$ 表示当前速度下模拟轨迹与障碍物之间

的最近距离; v'_b, ω'_b 为最大减速度。

4) 综合上述 3 类速度限制, 最终的速度采样空间是三者交集:

$$V_s = V_m \cap V_d \cap V_a \quad (17)$$

图 6 为预测在特定时间间隔 Δt 内的候选轨迹集合。

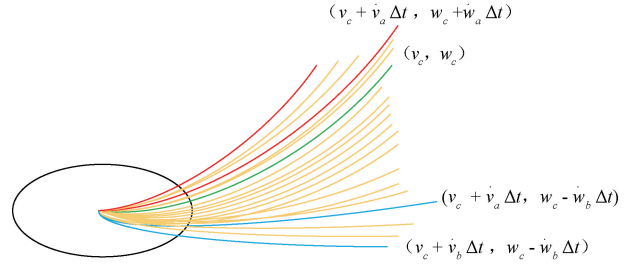


图 6 速度采样曲线

Fig. 6 Velocity sampling curve

完成速度采样后, 在设定时间段内模拟所有可能轨迹, 通过传统 DWA 评价函数评分, 选取最高分轨迹作为最终路径。传统 DWA 的评价函数如式(18)所示。

$$G(v, \omega) = \alpha \times H(v, \omega) + \beta \times D(v, \omega) + \gamma \times V(v, \omega) \quad (18)$$

式中: $H(v, \omega)$ 为偏航角评价函数, 评估在当前采样速度下轨迹终点方向与目标点连线的夹角差, 其公式为 $180^\circ - \theta$; $D(v, \omega)$ 为距离评价函数, 评估轨迹与最近障碍物的距离; $V(v, \omega)$ 为速度评价函数, 以机器人当前线速度大小指标, 线速度越大得分越高; α, β 和 γ 为各自的权重系数。

2.3 基于多目标优化的自适应评价函数设计

传统 DWA 评价函数依据方位角偏差, 轨迹与障碍物最近距离, 当前速度大小这 3 方面进行评分。虽能实现局部避障与航向控制, 但在复杂非结构化环境中仍存在全局路径偏离风险和障碍物响应不足两类显著缺陷。针对上述问题, 本节在传统 DWA 评价函数基础上引入全局路径评价函数及制动距离评价函数, 得到新的评价函数为:

$$G(v, \omega) = \alpha \times H(v, \omega) + \beta \times D(v, \omega) + \gamma \times V(v, \omega) + \theta \times T(v, \omega) + \varepsilon \times B(v, \omega) \quad (19)$$

式中: $T(v, \omega), B(v, \omega)$ 分别为全局路径评价函数和制动距离评价函数; θ, ε 为它们对应的权重系数。式(19)中各评价子函数需进行归一化处理。

1) 全局路径评价函数

本文设计的全局路径评价函数为:

$$T(v, \omega) = \frac{1}{(x+1)} \exp(-x) \quad (20)$$

式中: 时间步长设置为 0.1 s; $x = \sum_{i=1}^n \min(d(n_i, N_i))$, 表示每 0.1 s 内局部路径上每个点到全局路径的最短距离之和; n_i 表示 0.1 s 内局部路径上某点, N_i 表示到局部路径点 n_i 距离最短的全局路径段上某点。

依据所设定的时间步长 0.1 s, 将 (v, ω) 离散化, 对每 0.1 s 内局部路径上每个点计算到全局路径的最短距离之和, 并取负指数。当局部路径完全拟合全局路径时, $T(v, \omega)$ 函数值达到最大值 1; 当局部路径远离全局路径时, $T(v, \omega)$ 函数值趋近于 0。

同时加入全局关键点进行约束, 使路径规划具有更强的全局导向性。具体步骤如下:

(1) 在 path_keypoint 列表中, 依次提取全局路径规划出的关键节点 *keypoint*。

(2) 依次将每个 *keypoint* 作为当前节点的子目标点, 若该子目标点距离障碍物的距离小于设定阈值 $z_1 = 0.5$ m, 或当前位置距离子目标点的距离小于设定阈值 $z_2 = 1.5$ m, 则选取下一个关键轨迹节点 *keypoint* 作为当前点的子目标点。

(3) 依次循环 1、2 操作, 直到遍历完所有的关键轨迹节点 *keypoint* 且找到终点时完成任务。

当式(20)中的 $x > 5.0$ m 时, 让全局路径评价函数 $T(v, \omega)$ 前的权重 θ 减小为 0.5, 对偏离轨迹做出惩罚。而其余情况下, 权重 θ 设置为 1.5, 不做任何改变。

2) 制动距离评价函数

由于移动机器人在运动时, 往往会伴随着一定的线速度。如果某时刻线速度过快, 而加速度设定不足会导致机器人易与障碍物发生碰撞。因此在本节引入制动距离评价函数, 一定程度上实现远离障碍物的路径规划。

$$B(v, \omega) = \begin{cases} 1.0, & b_d(v, \omega) \leq 0.5 \\ \frac{k}{b_d(v, \omega)}, & 0.5 < b_d(v, \omega) < 2.0 \\ 0.1, & 2.0 \leq b_d(v, \omega) \end{cases} \quad (21)$$

式中: $b_d(v, \omega)$ 是当前速度 v 和角速度 ω 下的制动距离; k 是一个常数, 将 k 设置为 0.5。

在路径规划中, 既不希望轨迹离障碍物太近, 也不希望轨迹离障碍物太远导致多绕路, 因此在该实验中设定制动距离阈值范围为 0.5~2.0 m。当制动距离过长 (≥ 2.0 m) 时, 可能存在碰撞风险, 给予最低评分, 当制动距离适中 (0.5~2.0 m) 时, 评分随制动距离线性减小, 当制动距离过小 (≤ 0.5 m) 时, 认为安全性有保障, 给予最高评分。

此外, 为进一步确保在制动距离过长时, 移动机器人能尽快刹住车, 还对速度评价函数 $V(v, \omega)$ 前的权重 γ 施加动态变化。当轨迹到周围最近障碍物的制动距离 > 2.0 m 时, 将权重 γ 变小为 0.1, 让评价函数尽量地不要倾向于往速度大的趋势发展。而正常情况下权重 γ 设置为 0.8。

为了解决权重 γ 的分段选择会引起评价函数不连续从而导致速度突变的问题, 在局部代价函数中引入了连续的平滑权重, 构造连续平滑权重 $\gamma(d)$, 采用 sigmoid:

$$\gamma(d) = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min}) \frac{1}{1 + \exp(k'(d - d_0))} \quad (22)$$

其中, $\gamma_{\max} = 0.8$, $\gamma_{\min} = 0.1$, $d_0 = 2.0$ m 为平滑函数的中心点阈值, $k' = 3.0$ 。

3) 平滑度定量分析

引入曲率二次积分公式: 检测的是轨迹空间上的平滑度, 即轨迹的曲率变化。它量化的是单位弧长上的航向变化率的平方的累积, 越小表示越平滑, 越少急弯和抖动。

$$\text{Smoothness} = \int_{s=0}^L k(s)^2 ds \quad (23)$$

$$\text{其中, } k(s) = \frac{d\theta}{ds}$$

归一化:

$$\overline{\text{Smoothness}} = \frac{1}{L} \int_0^L k(s)^2 ds \quad (24)$$

离散近似:

设路径点 (x_i, y_i) , $i = 1, \dots, N$, 用弧长差 $s_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$, 离散近似计算每段的航向 $\theta_i = \text{atan2}(y_{i+1} - y_i, x_{i+1} - x_i)$; 计算相邻航向差 $\Delta\theta_i = \text{unwrap}(\theta_{i+1} - \theta_i)$; 每段对应的曲率近似 $k_i \approx \Delta\theta_i / s_{i+1}$

得出曲率二次积分公式:

$$\int k^2 ds \approx \sum_i (k_i)^2 \cdot s_{i+1} = \sum_i \frac{(\Delta\theta_i)^2}{s_{i+1}} \quad (25)$$

归一化: 将曲率二次积分公式除以总弧长 $L = \sum s_i$ 。

3 融合改进 A* 及 DWA 算法路径规划

由于 A* 算法和 DWA 算法都有各自的缺点, A* 算法虽能迅速规划出一条较佳路径, 但只能对静态障碍物进行避障, 无法绕开动态障碍物; DWA 算法虽能进行实时避障, 但在复杂环境比如“C”形障碍物处避障效果不好, 无法预见更远的障碍以及容易陷入局部最优。结合两者的优点, 本文将改进二者进行融合。在通过改进 A* 算法进行全局路径规划后, 提取改进 A* 算法规划出的关键轨迹节点作为 DWA 的子目标点, 引导局部路径规划轨迹尽可能往已经规划好的全局路径轨迹靠, 保证移动机器人既能够规划出一条较优路径, 也能够进行实时避障和逃离局部最小值点。

图 7 为融合算法流程图。首先, 基于栅格地图, 初始化 A* 算法的 OpenList 和 CloseList, 将起点加入 OpenList 并设为父节点; 随后, 遍历父节点的邻域节点, 筛选掉靠近障碍物或在 CloseList 中的节点, 将符合条件的加入 OpenList。并在 A* 评价函数中引入障碍物密度函数以提升路径安全性, 更新节点的 f, g, h 值后, 将 f 值最小节点移入 CloseList 并设为新父节点; 重复此过程直至到达目标点; 接着, 对 A* 输出的原始路径过滤冗余节点并优化曲线, 提取关键节点 *keypoint* 形成全局参考路径; 最后, 在局部导航层启动 DWA; 进行速度采样, 将当前最近的关键节点设为子目标点, 预测候选轨迹, 在 DWA 评价函数中扩展融入全局路径与制动距离评价项, 经轨迹评价筛选出最优

轨迹执行;循环执行 DWA 步骤直至抵达最终目标点。

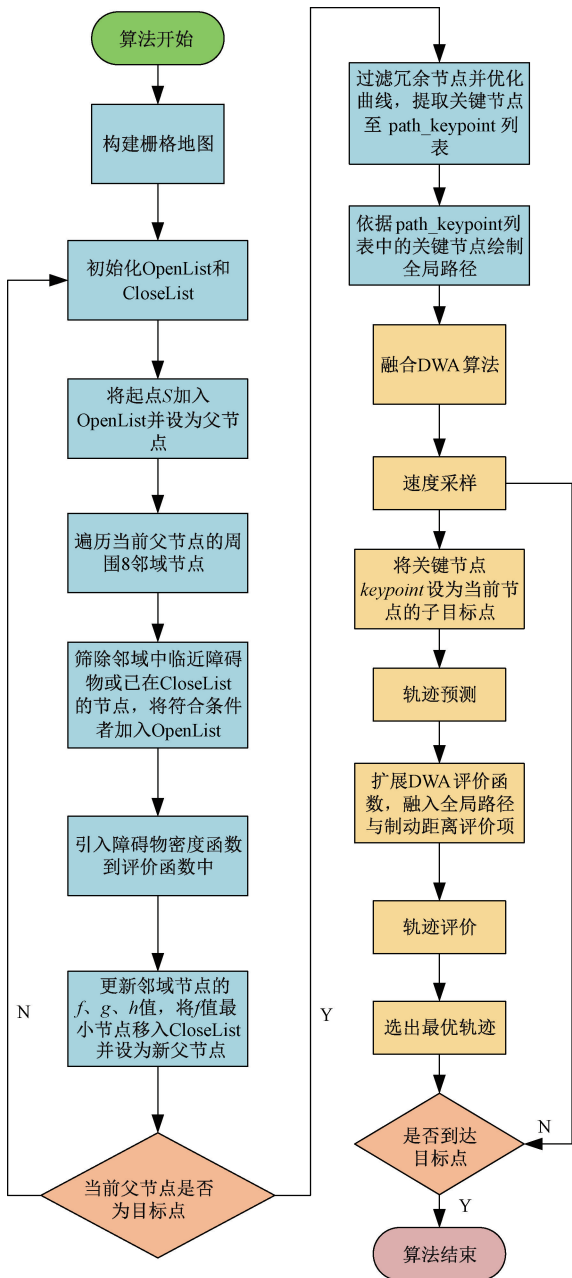


图 7 融合改进 A* 与 DWA 算法流程

Fig. 7 Flowchart of the fusion-improved A* and DWA

4 仿真实验结果及分析

为验证融合算法的有效性,在 Matlab R2017a 上以 31×31,分辨率 10 cm×10 cm 的栅格地图进行仿真实验。在模拟地图中,起点坐标统一为(6.5,4.5),终点坐标统一为(24.5,28.5),椭圆形代表移动障碍物,方块代表在全局路径规划后添加的障碍物,网格部分为 A* 算法全局规划中遍历的所有节点。

4.1 改进 A* 算法对比仿真实验

本小节分别对 A* 算法改进前后、JPS 及文献[19]算法、ACO+GA、双向 Informed-RRT*、Smooth-RRT 进行对比实验。

如图 8(a)~(g)所示,分别给出了不同算法路径规划结果,可以看出,改进后的 A* 算法较其余算法展现出全面优势。改进后的 A* 算法在安全性方面实现重大突破,规划路径与障碍物始终保持合理安全距离,有效规避碰撞风险。相比原始 A* 算法存在的安全隐患、JPS 算法的尖锐转折缺陷、文献[19]算法的路径冗余、GA+ACO 算法的转折过多、双向 Informed-RRT*、Smooth-RRT 总转折度数过大及总遍历节点数过多等问题,本文改进算法通过优化路径平滑性、增强安全避障能力和提升计算效率,展现出全面优越性。

表 1 给出了不同算法性能分析,可以看出,改进算法在路径转折次数上较原始 A* 算法降低了 42.86%,较文献[19]算法同样降低了 42.86%,较 JPS 算法降低了 33.33%,较 GA+ACO 算法降低了 55.56%,较双向 Informed-RRT*、Smooth-RRT 算法更是降低很多;在总转折度数方面,较原始 A* 算法降低了 65.23%,较文献[19]算法降低了 65.51%,较 JPS 算法降低了 63.73%,较 GA+ACO 算法降低了 76.75%,较双向 Informed-RRT* 降低了 91.34%,较 Smooth-RRT 降低了 89.68%;路径长度较原始 A* 算法缩短 3.9%,较文献[19]算法缩短 5.01%,较 JPS 算法缩短 3.9%,较 GA+ACO 算法缩短 2.67%,较双向 Informed-RRT* 缩短 14.51%,较 Smooth-RRT 缩短 12.49%;遍历节点数较原始 A* 算法减少 47.89%,较文献[19]算法减少 59.19%,较 GA+ACO 算法减少 76.58%,较双向 Informed-RRT* 减少 57.47%、较 Smooth-RRT 减少 68.99%。并且这些改进显著缩短了运行时间。

综合而言,改进后的 A* 算法在多个关键指标上超越对比算法;较原始 A* 算法实现路径平滑度、安全性和计算效率的全面提升;较文献[19]算法和 GA+ACO 算法在缩短路径的基础上大幅提升安全性能;较 JPS 算法在保持路径长度优势的同时显著改善运动连续性;较双向 Informed-RRT*、Smooth-RRT 路径更短,且转折度数大幅减少,有更强的抉择性。这种多目标协同优化的特性,使本改进算法成为复杂环境下移动机器人导航的更优解决方案。

4.2 融合改进 A* 与 DWA 导航算法仿真实验

1)基本参数

在移动机器人自主导航系统中,参数配置直接影响算法性能与运动控制精度。本节从机器人运动特性、融合算法实现及局部轨迹评价 3 个维度,对关键参数进行下列分类定义与说明:

(1)移动机器人机械特性:最小线速度、最大线速度、最小角速度、最大角速度、最大线加速度、最大角加速度。

(2)融合算法基本参数:障碍物半径、线速度分辨率、角速度分辨率、时间分辨率。

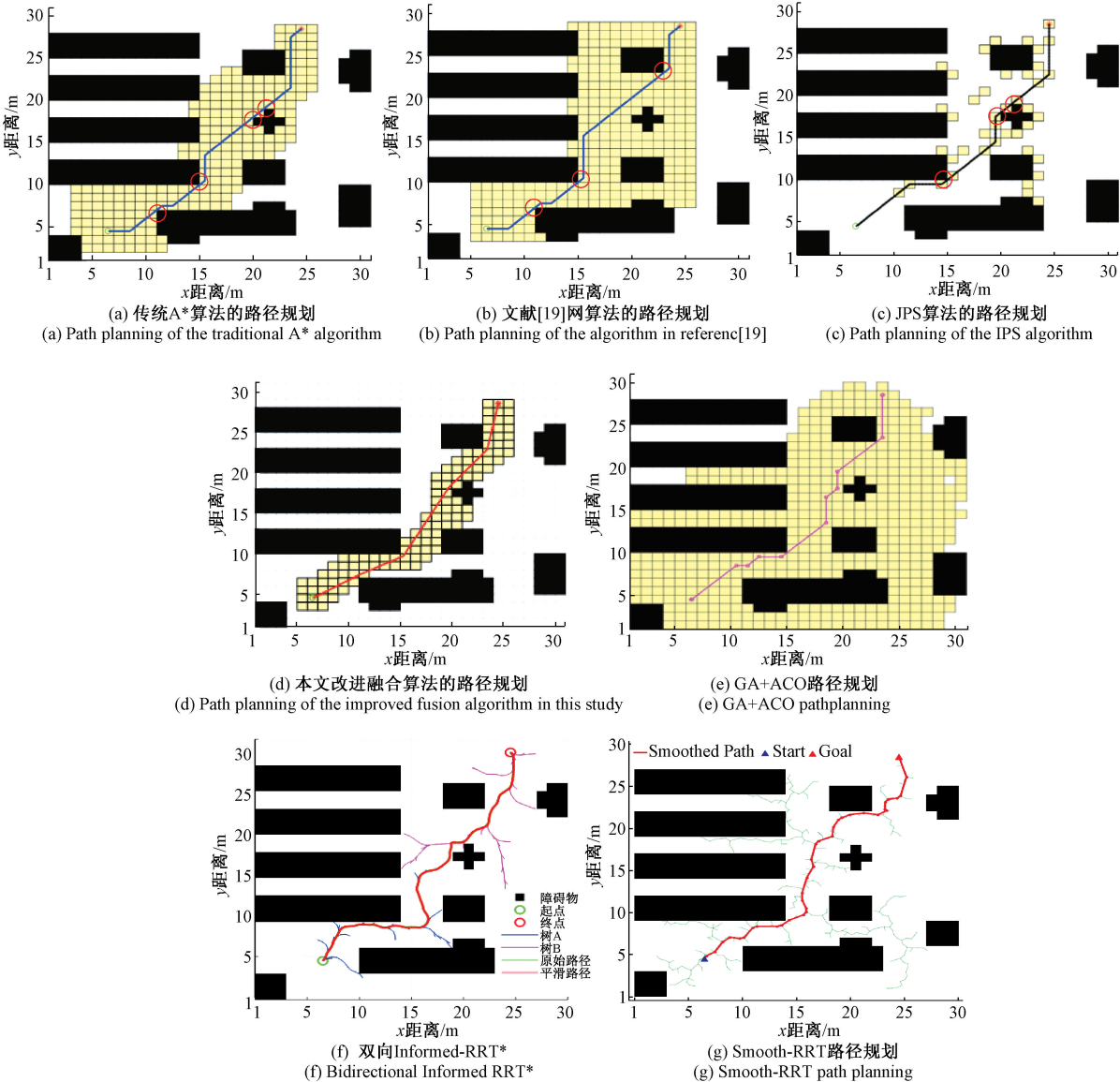


图 8 全局算法横向对比

Fig. 8 Cross-comparison of global planners

表 1 不同路径规划算法的性能指标比较

Table 1 Comparison of performance metrics of different path planning algorithms

算法名称	转折次数	总转折度数/(°)	路径长度/m	所遍历的节点个数	导航时间/s	碰撞点个数
原始 A*	7	315.000 0	33.213 2	213	0.045 339	4
文献[19]	7	316.100 0	33.603 5	272	0.030 940	3
JPS 算法	6	302.000 0	33.213 2	42	0.313 060	3
本文改进 A*	4	109.526 1	31.923 3	111	0.043 578	0
GA+ACO	9	469.000 0	32.799 0	474	14.870 619	2
B Informed-RRT*	—	1 265.320 0	37.340 0	261	12.988 600	0
Smooth-RRT	—	1 061.510 0	36.480 0	358	7.053 000	0

(3)DWA 评价函数参数： α 为方位角评价函数权重、 β 为轨迹与障碍物最近距离的权重、 γ 为当前速度评价函数权重、 θ 跟踪全局规划路径权重、 ϵ 为制动距离评价函数权重、 $B(v, \omega)$ 中的常数 k 。上述参数信息如表 2~4 所示。

表 2 移动机器人机械特性参数

Table 2 Mechanical characteristics parameters of mobile robots

最小线速度/ ($\text{m}\cdot\text{s}^{-1}$)	最大线速度/ ($\text{m}\cdot\text{s}^{-1}$)	最大线加速度/ ($\text{m}\cdot\text{s}^{-2}$)	最小角速度/ ($\text{rad}\cdot\text{s}^{-1}$)	最大角速度/ ($\text{rad}\cdot\text{s}^{-1}$)	最大角加速度/ ($\text{rad}\cdot\text{s}^{-2}$)
0	1.5	0.2	0	0.349	0.873

表 3 融合算法基本参数

Table 3 Basic parameters of the fusion algorithm

单格障碍物面积/ s^2	线速度分辨率/ $(\text{m}\cdot\text{s}^{-1})$	角速度分辨率/ $(\text{rad}\cdot\text{s}^{-1})$	时间分辨率/s
1	0.02	0.017 5	0.1

表 4 DWA 评价函数参数

Table 4 DWA evaluation function parameters

偏航角评价 函数权重 α	距离评价函数 权重 β	速度评价函数 权重 γ	全局路径评价 函数权重 θ	制动距离评价 函数权重 ϵ	$B(v,\omega)$ 中的 常数 k
1.5	1.5	0.1~0.45	0.5/1.5	1.0	0.5

2)融合改进 A* 与 DWA 导航算法仿真结果

为了更加客观地评价所提出的改进融合导航算法的性能,本文在两种典型场景下分别进行了对照实验。图 9(a)为原始融合算法在场景一的局部规划过程:图中虚线为预设的全局参考路径,实线为局部避

障轨迹,短线表示以 0.1 s 为采样周期所生成的候选轨迹。可以看出,原始算法在缺乏全局路径关键节点引导的情况下,生成的局部路径明显偏离全局参考路径并易陷入局部最小值,从而导致规划结果与全局最优解相差较大。

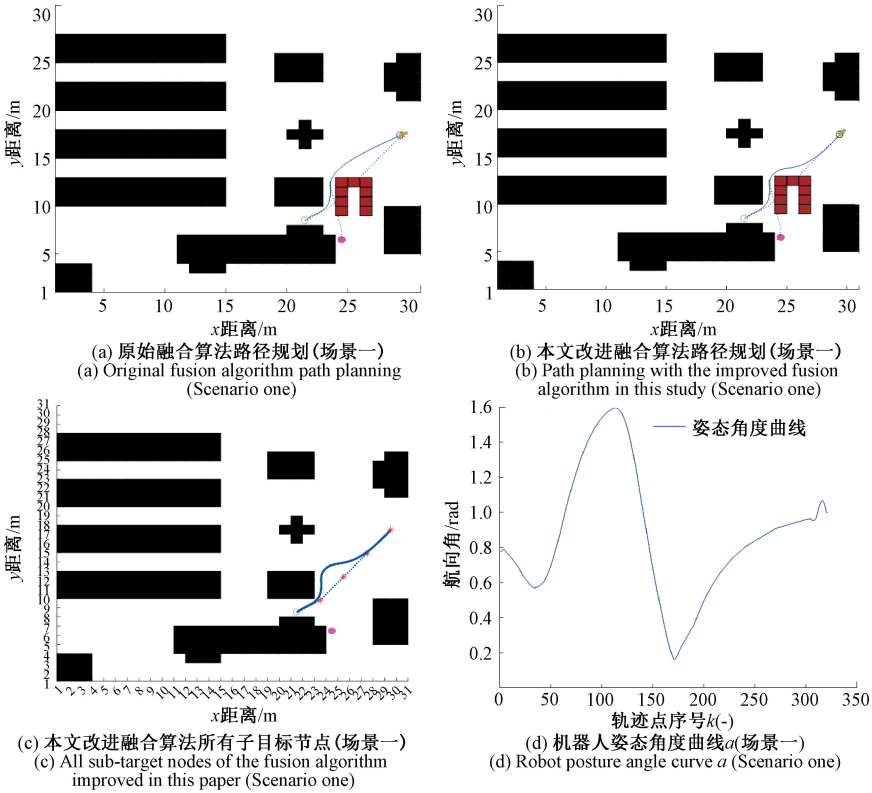


图 9 场景一局部规划过程与算法横对比

Fig. 9 Local planning process and algorithm comparison in scenario one

针对上述问题,本文提出在局部规划中引入全局路径评价函数与制动距离评价函数,并在全局规划过程中动态

提取关键轨迹节点作为局部规划的子目标点。通过该设计,局部规划不仅能继承全局规划的引导信息,还能在靠近障碍物或转向幅度较大时提前减速,减少不必要的速度震荡与姿态突变。图 9(b)给出了改进算法在场景一的规划结果:相比图 9(a)原始方法,改进算法生成的局部路径更靠近全局参考路径,路径形状更加连贯且有效避开了导致局部极小值的危险区域,且其运动姿态平滑(如图 9(d)所示)。

图 9 和 10 验证算法在动态环境下的适应性,本文进

一步在场景一与场景二中分别加入动态障碍物进行测试。为便于客观比较,动态障碍物速度均设为 0.3 m/s、半径为 0.35 m。在场景一中障碍物于程序开始时释放,而场景二中则于 40 s 时释放。局部规划模块通过对候选轨迹与动态障碍物的相交时间进行粗略计算,快速判定潜在碰撞风险,从而提前启动局部避障调整。结果表明,本文改进算法能够有效检测并避开移动障碍物,在动态场景下仍保持路径的平滑与可行性,验证了其在复杂环境中的鲁棒性。

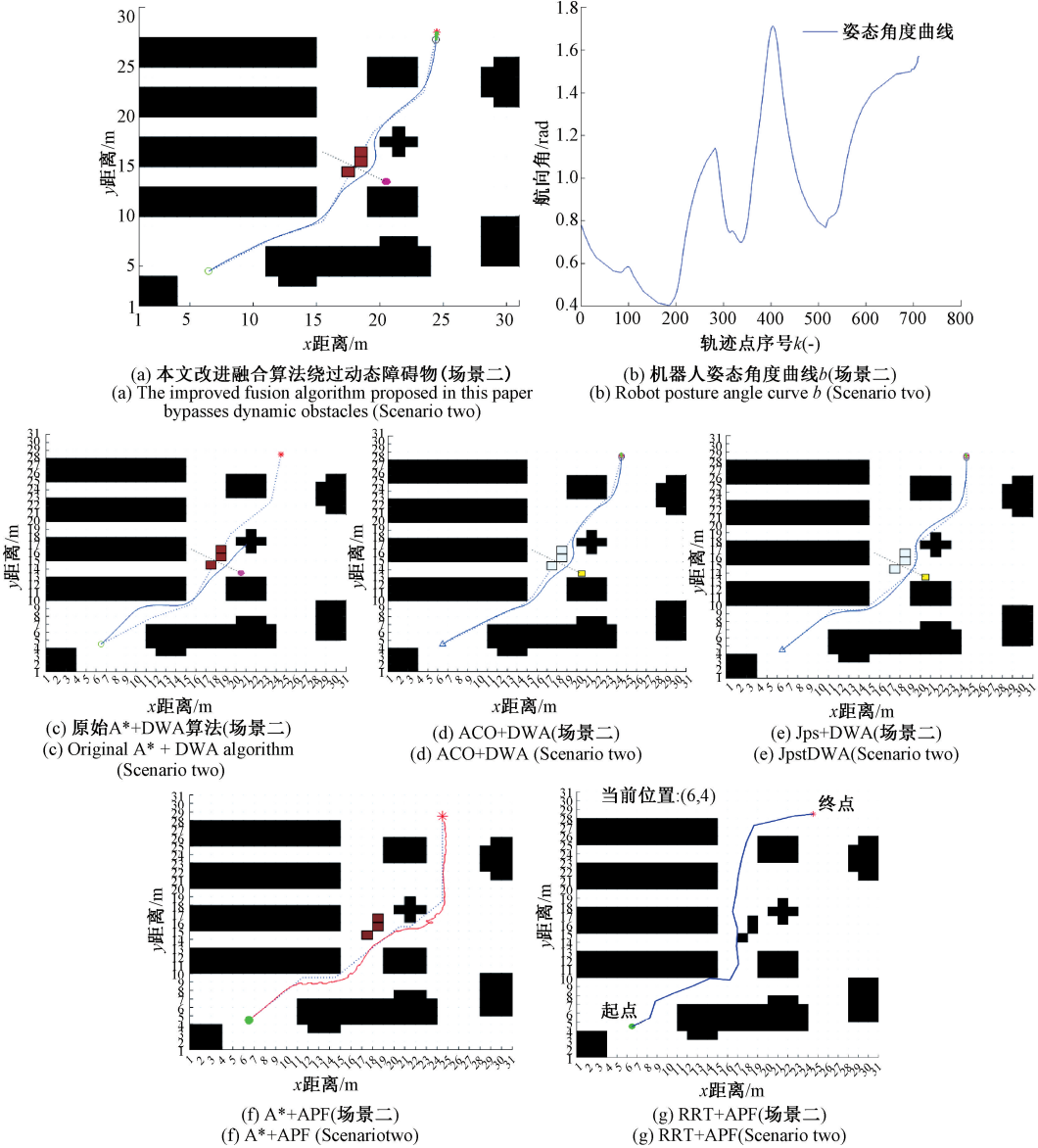


图 10 场景二局部规划过程及各算法横向对比

Fig. 10 Partial planning process for scenario two and lateral comparison of various algorithms

需要说明的是,制动距离评价函数的引入并未显著增加计算负担。在场景一中的实验数据显示,加入该评价函数后总体规划时长不仅未增加,反而较原始算法减少约 4 s。造成这一现象的原因在于:制动距离函数能够在早期

阶段剪除高风险轨迹,减少后续碰撞检测与重规划次数,从而在系统层面上实现了“以少量前期计算换取整体效率提升”的效果。这一结果说明,本文改进算法在保证安全性的同时,还具有较高的计算效率。

如图 10(a)~(g)及表 5 进一步验证算法的综合性能,本文在场景二中对多种主流融合方法。表 5 列出了各算法的路径长度与规划总时长:本文方法(Ours)路径长度为 32.044 0 m,规划总时长 116 s;ACO+DWA 为 32.052 6 m、规划总时长 121 s;JPS+DWA 为 32.747 4 m、规划总时长 129 s;原始 A*+DWA 在场景二中规划失败陷入局部最小值;A*+APF 为 39.270 0 m、规划总时长 110 s;RRT+APF 为 36.029 5 m、规划总时长 105 s。可以看出,原始 A*+DWA 在场景一中可生成可比轨迹,但在更复杂的场景二中已无法完成有效规划,进一步验证了本文改进措施的必要性。

表 5 横向算法对比(场景二)

算法名称	路径长度/m	导航时间/s
本文	32.044 0	116
ACO+DWA	32.052 6	121
JPS+DWA	32.747 4	129
原始 A*+DWA	规划失败	规划失败
A*+APF	39.270 0	110
RRT+APF	36.029 5	105

从定量结果来看,在路径长度上,本文算法与 ACO+

DWA、JPS+DWA 相当,且略优于后者;相较于 A*+APF 和 RRT+APF,本文算法的路径明显更短,并且其在保持避障能力的同时能够更接近全局最优路径。另一方面,在计算时长上,本文改进方法的规划总时长略优于 ACO+DWA 与 JPS+DWA,但稍高于 RRT+APF 和 A*+APF。这种时长上的差异主要来源于算法在局部规划阶段对全局引导信息和制动距离的额外评价与关键节点提取,这类额外计算在换取路径质量与运动平滑性的同时带来了很小的计算开销;相较于因追求速度而牺牲路径质量或易发生抖动的算法,如 A*+APF 和 RRT+APF,本文方法在“路径质量—计算成本—运动平滑性”之间取得了较为平衡的折中。

如图 11(a)~(b)所示,从轨迹平滑性与运动可执行性角度分析,本文改进方法具有明显优势。实验中以 0.1 s 的分辨率生成候选轨迹,并结合平滑化处理后的分段权重函数 γ ,使得线速度曲线在权重切换处保持连续,进一步提升了轨迹的连续性与可控性。如图 11(a)所示的线速度与角速度曲线表明:改进后机器人在整个导航过程中速度与姿态变化平稳,未出现因局部避障导致的速度突变或急转,保证了运动指令的连续性与安全性。而如图 11(b)所示,未加入 γ 平滑度时,线速度有明显突变,角速度的变化幅度明显更大。

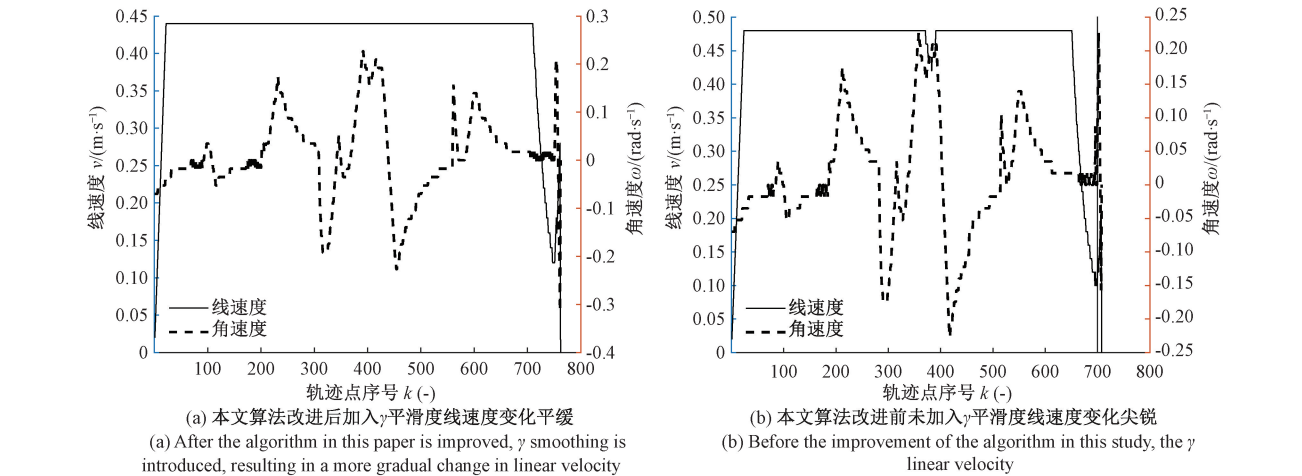


图 11 本文算法加入 γ 平滑度前后线速度对比
Fig. 11 Comparison of linear velocity before and after adding γ smoothness in the proposed algorithm

为了对平滑性进行定量分析,本文计算了平滑度指标与归一化指标,以及平均绝对航向变化率。结果为: $k_2 = 0.192\ 3$ 、 $\hat{k}_2 = 0.055\ 6\ \text{rad}^2/\text{m}^2$ 、平均绝对航向变化为 $0.124\ 7\ \text{rad/m}$ (约 $7.14^\circ/\text{m}$)。该结果表明,本文改进算法生成的路径在曲率与航向变化上均较为平缓,具有良好的平滑性和可执行性。

综上所述,本文提出的改进融合算法通过全局关键点引导、制动距离约束及 γ 平滑化策略的协同设计,兼顾了规划精度、动态避障能力、运动平滑性与计算效率。在

静态与动态环境下的对比实验均表明,本方法在路径质量与鲁棒性方面均较现有典型算法取得了显著提升。

5 实物实验结果及分析

为验证本文融合算法在实际应用中的有效性,在 ROS 操作系统中,通过实车激光雷达 SLAM 建图,并在 rviz 中查看建图效果。本文建图采用 GMapping 算法,是目前最常用的建图算法。

实验环境为运行内存为 4 GB/2 处理器的 64 位

Ubuntu 18.04 操作系统,使用阿克曼小车(如图 12(a)所示)作为移动机器人,搭载 N10P 雷达进行建图扫描。

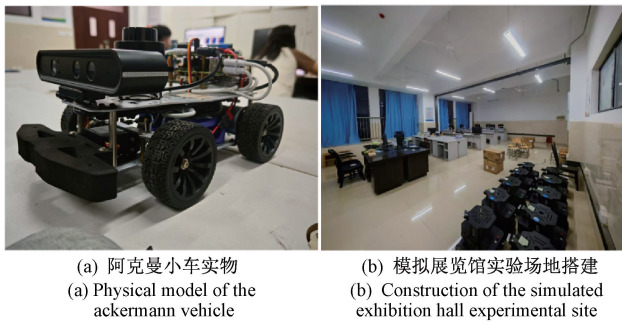


图 12 实验室真实场景及阿克曼小车实物展示
Fig. 12 Laboratory real-world scenario and physical display of the ackermann vehicle

如图 12(b)所示,以学校实验室环境模拟展览馆环境。场景的实际长为 11 m,宽 8 m,分辨率为 10 cm×10 cm,起点为(3.5 m,3 m),终点设为(9.2 m,7.5 m)。为保证移动机器人的安全性,本实验提高了障碍物的膨胀半径,如图 13 模糊黑色阴影部分。



图 13 原始 A* 和 DWA 算法实车导航结果
Fig. 13 Navigation results of the original A* and DWA algorithms on a real vehicle

在实际环境中,分别对改进前后的融合算法进行对比实验。图 13 为改进前的融合 A* 与 DWA 算法所规划的轨迹,由于改进前的融合算法并没有进行路径平滑处理等操作,明显发现轨迹结果中出现较多的路径震荡以及拐点,并且所规划轨迹相较于改进后的融合算法更靠近危险障碍物。

如图 14 所示,基于原始融合算法的缺陷,将本文改进的融合算法导入阿克曼小车中进行实车实验,并且得到以下结果。可以发现,实车规划轨迹和上文第 4 章 Matlab 仿真结果一致,证明了算法的可行性。由于对原始算法增添轨迹平滑处理以及远离障碍物评价函数等操作,改进后的融合算法所规划的路径明显更加平滑,且相比于改进前更远离障碍物。不仅消除了转向震荡对移动机器人的损害,而且提高了路径结果的安全性。

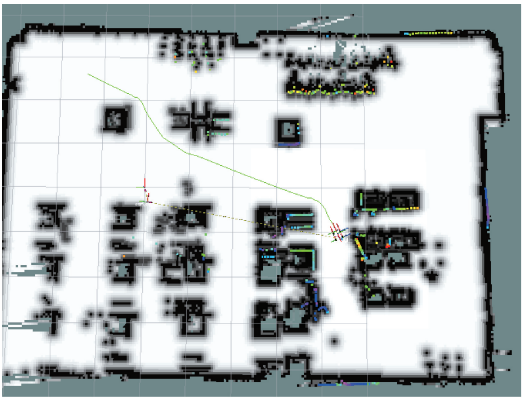


图 14 本文改进融合算法实车导航结果
Fig. 14 Real vehicle navigation results using the improved fusion algorithm in this study

为了更进一步的显现本文改进融合算法的有效性,在实际环境中增添“C”型障碍物实验。如图 15 圆圈所示,在所规划出的路径线上摆放“C”型障碍物,观察发现移动机器人会自主避开“C”型障碍物。并且在遇到有障碍物阻挡原先规划出的路径时,移动机器人并没有出现多余的犹豫以及转向震荡,能够选择出更优的路径,成功抵达终点。针对传统 DWA 算法,在移动机器人在面对“C”型障碍物时,易于陷入局部最优的缺陷进行了较大的改良。

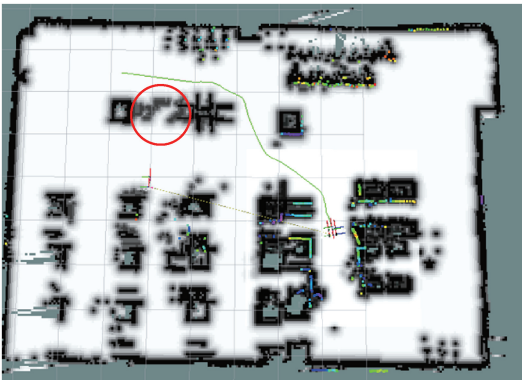


图 15 本文改进融合算法绕开“C”型障碍物
Fig. 15 Illustrates the improved fusion algorithm in this paper bypassing a “C”-shaped obstacle

综合以上分析,采用原始 A* 与 DWA 融合算法并不能满足 AGV 在展览馆中的导航要求。而本文提出的改进融合算法不仅能够成功规划出无碰撞路径,而且在路径的安全性和平滑程度方面有了很大的提升,非常契合展览馆环境对 AGV 的导航要求。

6 结 论

传统 A* 算法在路径规划中存在效率低下、路径不平滑、过度贴近已知障碍物边界以及无法应对动态未知障碍物等局限性。针对这些问题,本研究通过多维度优化策略

实现算法升级:首先,在传统 A* 的评价函数中融合环境风险代价约束以及基于障碍物感知的邻域节点筛选改进,结合栅格地图的障碍物分布信息优化启发函数权重,显著减少无效节点扩展,提升搜索效率;其次,设计路径关键点提取算法,利用曲率约束和冗余点剪枝技术消除路径中的锯齿抖动,大幅减少路径拐点,生成平滑可执行的轨迹。为进一步增强算法动态避障能力,将改进 A* 算法输出的关键路径点与改进后的动态窗口法深度融合,在原 DWA 轨迹评价函数中嵌入全局路径引导权重及制动距离权重,实现全局最优性与局部避障的协同优化。通过 Matlab 仿真平台与实物移动机器人实验的对比验证,改进算法在路径长度、总转折度数、转折次数、遍历的节点数等指标上均优于对比算法,且在动态未知障碍物场景中成功率接近 100%,验证了改进融合算法在复杂环境中的鲁棒性与工程实用性,为移动机器人自主导航提供了更安全高效的解决方案。

参考文献

- [1] SHI Y L, HUANG SH CH, LI M X. An improved global and local fusion path-planning algorithm for mobile robots[J]. *Sensors*, 2024, 24(24): 7950.
- [2] YU M X, LUO Q, WANG H B, et al. Electric logistics vehicle path planning based on the fusion of the improved a-star algorithm and dynamic window approach[J]. *World Electric Vehicle Journal*, 2023, 14(8): 213.
- [3] LAI R SH, WU ZH Y, LIU X G, et al. Fusion algorithm of the improved A* algorithm and segmented bezier curves for the path planning of mobile robots[J]. *Sustainability*, 2023, 15(3): 2483.
- [4] DONG D J, DONG SH Y, ZHANG L L, et al. Path planning based on A-star and dynamic window approach algorithm for wild environment[J]. *Journal of Shanghai Jiaotong University (Science)*, 2024, 29(4): 725-736.
- [5] LIAO T J, CHEN F, WU Y T, et al. Research on path planning with the integration of adaptive a-star algorithm and improved dynamic window approach[J]. *Electronics*, 2024, 13(2): 455.
- [6] ZHANG R, ZHOU L, LIU ZH Y. Dynamic path planning for mobile robot based on RRT* and dynamic window approach[J]. *Journal of System Simulation*, 2024, 36(4): 957-968.
- [7] SHEN CH, SOH G S. Targeted sampling dynamic window approach: A path-aware dynamic window approach sampling strategy for omni-directional robots[J]. *Journal of Mechanisms and Robotics*, 2024, 16(11): 111001.
- [8] WANG T, LI AI J, GUO D K, et al. Global dynamic path planning of AGV based on fusion of improved A* algorithm and dynamic window method[J]. *Sensors*, 2024, 24(6): 2011.
- [9] WU B, ZHANG W, CHI X N, et al. A novel AGV path planning approach for narrow channels based on the Bi-RRT algorithm with a failure rate threshold[J]. *Sensors*, 2023, 23(17): 7547.
- [10] 李娟,张子浩,张宏瀚. 复杂环境下 DWA 与 RRT 算法融合的 AUV 局部路径规划[J]. *智能系统学报*, 2024, 19(4): 961-973.
- [11] LI J, ZHANG Z H, ZHANG H H. Local path planning for AUV by fusing DWA and RRT algorithms in complex environments [J]. *CAAI Transactions on Intelligent Systems*, 2024, 19(4): 961-973.
- [12] HOSSAIN T, HABIBULLAH H, ISLAM R, et al. Local path planning for autonomous mobile robots by integrating modified dynamic-window approach and improved follow the gap method[J]. *Journal of Field Robotics*, 2022, 39(4): 371-386.
- [13] MACENSKI S, MOORE T, DAVID V, et al. From the desks of ROS maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2 [J]. *Robotics and Autonomous Systems*, 2023, 168: 104493.
- [14] YANG L W, LI P, QIAN S, et al. Path planning technique for mobile robots: A review[J]. *Machines*, 2023, 11(10): 980.
- [15] 邓云峥,黄翼虎. 复杂动态环境下基于 A* 的改进 DWA 算法研究[J]. *电子测量技术*, 2023, 46(9): 69-76.
- [16] DENG Y ZH, HUANG Y H. Research on an improved DWA algorithm based on A* in complex dynamic environments[J]. *Electronic Measurement Technology*, 2023, 46(9): 69-76.
- [17] 朱洪波,殷宏亮. 分层平滑优化 A* 引导 DWA 用于机器人路径规划[J]. *电子测量与仪器学报*, 2024, 38(9): 155-168.
- [18] ZHU H B, YIN H L. Hierarchical smoothing optimization A* guided DWA for robot path planning[J]. *Journal of Electronic Measurement and Instrumentation*, 2024, 38(9): 155-168.
- [19] 宋俊辉,刘宇庭,郭世杰. 动态环境下 AIP-RRT* 与 DGF-APF 融合的机器人路径规划[J]. *仪器仪表学报*, 2025, 46(3): 51-64.
- [20] SONG J H, LIU Y T, GUO SH J. Robot path planning by fusing AIP-RRT* and DGF-APF in dynamic environments [J]. *Chinese Journal of Scientific Instrument*, 2025, 46(3): 51-64.
- [21] GONG X, GAO Y F, WANG F B, et al. A local path planning algorithm for robots based on improved DWA [J]. *Electronics*, 2024, 13(15): 2965.
- [22] DONG D J, DONG SH Y, ZHANG L L, et al. Path planning based on A-Star and dynamic window approach algorithm for wild environment[J]. *Journal of Shanghai Jiaotong University (Science)*, 2024, 29(4): 725-736.
- [23] 杨明亮,李宁. 改进 A* 算法的工业机器人路径规划[J]. *机械科学与技术*, 2022, 41(5): 795-800.
- [24] YANG M L, LI N. An improved A* algorithm for mobile robot path planning[J]. *Mechanical Science and Technology for Aerospace Engineering*, 2022, 41(5): 795-800.

作者简介

杨国亮,博士,教授,主要研究方向为智能控制、模式识别与图像处理等。
E-mail: ygliang30@126.com
李郅腾(通信作者),硕士研究生,主要研究方向为智能小车路径规划。
E-mail: 1546301490@qq.com