

8-15-2024

Fusion of Improved A* and Dynamic Window Approach for Mobile Robot Path Planning

Rongshen Lai

Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen 361024, China

Lei Dou

Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen 361024, China

Zhiyong Wu

Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen 361024, China

Shuai Sun

Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen 361024, China

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact xtfzxb@126.com.

Fusion of Improved A* and Dynamic Window Approach for Mobile Robot Path Planning

Abstract

Abstract: The traditional A* algorithm is computationally simple and has short planning paths, but it still suffers from redundancy of inflection points, low search efficiency and zigzagging planning paths. Aiming at the above problems, a fusion algorithm combining the improved A* algorithm and the improved dynamic window approach is proposed for the path planning of mobile robots. For the problem of redundant inflection points, the key nodes are extracted to effectively remove the useless inflection points; for the problem of low search efficiency, a dynamic weighting factor is introduced into the heuristic function of the evaluation function, which changes the search neighborhood and reduces the search nodes to improve the operation efficiency of the algorithm; for the problem of zigzagging of the planning paths, the paths are optimized using the improved dynamic window approach, which improves the smoothness of the paths. Through Matlab simulation and comparison experiments, the fusion algorithm is proved to be advantageous in global path optimization, effectively reducing the number of redundant nodes in the path and shortening the path length. In addition, the proposed fusion algorithm can increase the robot path smoothness and motion stability.

Keywords

A* algorithm, dynamic window approach, fusion algorithm, path planning, path smoothing

Recommended Citation

Lai Rongshen, Dou Lei, Wu Zhiyong, et al. Fusion of Improved A* and Dynamic Window Approach for Mobile Robot Path Planning[J]. Journal of System Simulation, 2024, 36(8): 1884-1894.

融合改进A*算法和动态窗口法的移动机器人路径规划

赖荣燊, 窦磊, 巫志勇, 孙帅

(厦门理工学院 机械与汽车工程学院, 福建 厦门 361024)

摘要: 针对传统A*算法存在拐点冗余、搜索效率低和规划路径曲折等问题, 提出一种改进A*算法与改进动态窗口法相结合的融合算法, 用于移动机器人的路径规划。针对拐点冗余问题, 通过提取关键节点有效去除无用拐点; 针对搜索效率低问题, 在评价函数的启发函数中引入动态加权因子, 改变搜索邻域, 减少搜索节点, 提高算法运行效率; 针对规划路径曲折问题, 对规划后的路径使用改进动态窗口法进行优化, 提高路径平滑度。通过MATLAB仿真对比实验, 证明融合算法在全局路径优化方面的优势, 有效减少路径冗余节点数量、缩短路径长度。此外, 所提融合算法能够增加机器人路径平滑度和运动稳定性。

关键词: A*算法; 动态窗口法; 融合算法; 路径规划; 路径平滑

中图分类号: TP242

文献标志码: A

文章编号: 1004-731X(2024)08-1884-11

DOI: 10.16182/j.issn1004731x.joss.24-0233

引用格式: 赖荣燊, 窦磊, 巫志勇, 等. 融合改进A*算法和动态窗口法的移动机器人路径规划[J]. 系统仿真学报, 2024, 36(8): 1884-1894.

Reference format: Lai Rongshen, Dou Lei, Wu Zhiyong, et al. Fusion of Improved A* and Dynamic Window Approach for Mobile Robot Path Planning[J]. Journal of System Simulation, 2024, 36(8): 1884-1894.

Fusion of Improved A* and Dynamic Window Approach for Mobile Robot Path Planning

Lai Rongshen, Dou Lei, Wu Zhiyong, Sun Shuai

(Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen 361024, China)

Abstract: The traditional A* algorithm is computationally simple and has short planning paths, but it still suffers from redundancy of inflection points, low search efficiency and zigzagging planning paths. Aiming at the above problems, a fusion algorithm combining the improved A* algorithm and the improved dynamic window approach is proposed for the path planning of mobile robots. *For the problem of redundant inflection points, the key nodes are extracted to effectively remove the useless inflection points; for the problem of low search efficiency, a dynamic weighting factor is introduced into the heuristic function of the evaluation function, which changes the search neighborhood and reduces the search nodes to improve the operation efficiency of the algorithm; for the problem of zigzagging of the planning paths, the paths are optimized using the improved dynamic window approach, which improves the smoothness of the paths.* Through Matlab simulation and comparison experiments, the fusion algorithm is proved to be advantageous in global path optimization, effectively reducing the number of redundant nodes in the path and shortening the path length. In addition, the proposed fusion algorithm can increase the robot path smoothness and motion stability.

Keywords: A* algorithm; dynamic window approach; fusion algorithm; path planning; path smoothing

收稿日期: 2024-03-13

修回日期: 2024-04-26

基金项目: 福建省自然科学基金(2022J011246); 科技部创新方法工作专项(2019IM010300)

第一作者: 赖荣燊(1983-), 男, 副教授, 博士, 研究方向为大规模个性化设计、机器人路径规划、故障诊断等。

0 引言

由于巨大的市场需求以及广阔的应用前景, 移动机器人广泛应用于工业制造、生活服务、交通运输、医疗健康等领域, 路径规划是提升移动机器人功能性能的关键技术之一。路径规划的主要目的是在有障碍物的环境下寻找到一条从起始位置到目标位置、不发生碰撞的最优路径。根据对环境信息的了解程度不同, 传统路径规划算法可以分为全局路径规划和局部路径规划两类, 目前, 常用的全局路径规划算法包括Dijkstra算法^[1]、A*算法^[2]、D*算法^[3]等, 局部路径规划算法包括动态窗口法^[4]、人工势场法^[5]等。

全局路径规划算法中, A*算法是广泛应用于路径规划的经典算法, 其计算简单、规划路径短, 但仍存在拐点冗余、搜索效率低、运行时间长等问题, 难以在短时间内规划出最优路径。对此, 诸多学者针对传统A*算法的改进展开研究。搜索策略优化方面, 文献[6]提出一种可搜索24邻域的A*算法, 通过将传统A*算法的8邻域搜索策略更改为24邻域搜索策略, 解决传统A*算法中路径长度不优和转折点过多的问题; 文献[7]进一步将传统A*算法的搜索邻域扩展到48邻域, 优化搜索角度并去除同方向上的多余子节点, 增强规划路径质量。评价函数优化方面, 文献[8]在评价函数的设计中加入路径曲率的影响代价, 减少路径中方向变化较大的转弯点, 提高路径平滑度; 文献[9]引入障碍率作为评价函数的影响因子, 避免计算资源的浪费, 提高算法的检索速度。路径平滑优化方面, 文献[10]使用贝塞尔曲线对规划后的路径进行平滑, 分别对规划后路径中曲折较大的部分使用二阶贝塞尔曲线进行平滑, 提高路径平滑度; 文献[11]使用动态窗口法对规划后的路径进行平滑, 考虑移动机器人的运动条件, 提升路径的平滑度和移动机器人的安全性。上述改进算法都在一定程度上提高了A*算法规划路径的质量, 但对算法降低运算时间和减少关键节点方面效果有限。

局部路径规划中较为常用的是动态窗口法, 其模型简单、具有良好的局部避障能力, 但存在运动状态单一、容易陷入局部最优等缺点。对此, 一些学者对动态窗口法的评价函数进行优化, 文献[12]对动态窗口法评价函数中速度评价的权值进行动态调整, 获得更加合理的路径, 使机器人移动更加安全可靠; 文献[13]向动态窗口法评价函数中引入规则评价函数, 即避让障碍物幅度越大越好, 减少碰撞风险, 提高路径安全性。上述改进算法对路径的安全性有一定的提升, 但缺乏对陷入局部最优问题的考虑。针对动态窗口法和全局路径规划算法的融合, 文献[14]针对传统A*算法在栅格数量较多时存在折点多、耗时长, 以及动态窗口法在复杂环境下灵活性差的问题, 提出一种融合改进A*算法和优化动态窗口法的路径规划算法; 文献[15]将A*算法和动态窗口法相结合, 将A*算法规划后的路径作为动态窗口法的全局路径指引, 避免动态窗口法容易陷入局部最优的问题。

基于上述研究, 针对复杂环境下移动机器人路径规划问题, 传统A*算法存在拐点冗余、搜索效率低、运行时间长等不足, 传统动态窗口法存在运动状态单一、容易陷入局部最优等不足。结合现有文献对A*算法和动态窗口法的改进思路分析, 本文提出一种将改进A*算法与改进动态窗口法相结合的融合路径规划算法。

1 改进A*算法

1.1 传统A*算法

A*算法相较于Dijkstra算法, 其优势在于引入启发式函数, 有效避免了盲目搜索, 大大提高了搜索效率。其评价函数通常为

$$F(n) = G(n) + H(n) \quad (1)$$

式中: $F(n)$ 为从起始节点到目标节点的预估总代价; $G(n)$ 为起始节点到当前节点的实际代价; $H(n)$ 为当前节点到目标节点的预估代价。

常用的 $H(n)$ 计算方法有曼哈顿距离、欧几里得距离和切比雪夫距离。为了更符合机器人移动时的实际距离，本文采用欧几里得距离进行路径计算，欧几里得距离计算公式为

$$H(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

式中： x_1 、 y_1 和 x_2 、 y_2 分别为当前节点的坐标和目标节点的坐标。

A*算法主要通过更新OpenList表和CloseList表来实现节点的筛选，OpenList表中存储着待扩展节点以及相应的代价值和父节点信息，CloseList表中存储着已经扩展过的节点以及相应的代价值和父节点信息。

1.2 改进A*算法

1.2.1 5邻域搜索策略

传统A*算法的搜索方式如图1所示，采用8邻域搜索方式来搜索下一个扩展节点。为了避免算法运算资源的浪费，本文将传统8邻域搜索方式更改为5邻域搜索方式，如表1所示，根据当前节点和目标节点的连线与节点 p_2 指向的夹角，假设夹角为 θ ，顺时针方向为正方向，根据夹角 θ 的范围，选择其中5个方向进行探索，减少冗余节点，提高算法效率。

1.2.2 自适应评价函数

传统A*算法的评价函数是由起始节点到当前节点的实际移动成本 $G(n)$ 和当前节点到目标节点的估计移动成本 $H(n)$ 组合而成，两者的占比从路径规划开始到路径规划结束都不会改变。本文根据地图规格和位置信息在传统A*算法启发函数基础上加入动态加权因子 W ，当前节点远离目标节点时，加权因子较大， $H(n)$ 占比较大，加快搜索速度；当前节点接近目标节点时，加权因子较小， $H(n)$ 占比较小，提高搜索精度。

$$F(n) = G(n) + W \cdot H(n) \quad (3)$$

式中： W 为根据地图的长宽之和以及当前节点与目标节点的曼哈顿距离计算的加权因子。

$$W = 1 + \frac{|x'_1 - x'_2| + |y'_1 - y'_2|}{m + n} \quad (4)$$

式中： x'_1 、 y'_1 和 x'_2 、 y'_2 为待求曼哈顿距离的当前节点和待求节点的坐标； m 、 n 为栅格地图的长和宽。

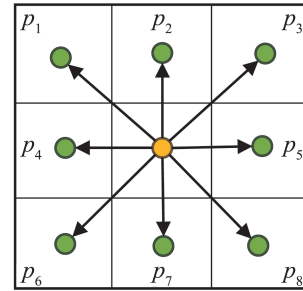


图1 节点扩展方向示意图

Fig. 1 Schematic diagram of node extension direction

表1 节点方向选择表

角度值	保留5方向	舍去3方向
$[22.5^\circ, 67.5^\circ)$	$p_1 \setminus p_2 \setminus p_3 \setminus p_5 \setminus p_8$	$p_4 \setminus p_6 \setminus p_7$
$[67.5^\circ, 112.5^\circ)$	$p_2 \setminus p_3 \setminus p_5 \setminus p_7 \setminus p_8$	$p_1 \setminus p_4 \setminus p_6$
$[112.5^\circ, 155.5^\circ)$	$p_3 \setminus p_5 \setminus p_6 \setminus p_7 \setminus p_8$	$p_1 \setminus p_2 \setminus p_4$
$[155.5^\circ, 202.5^\circ)$	$p_4 \setminus p_5 \setminus p_6 \setminus p_7 \setminus p_8$	$p_1 \setminus p_2 \setminus p_3$
$[202.5^\circ, 247.5^\circ)$	$p_1 \setminus p_4 \setminus p_6 \setminus p_7 \setminus p_8$	$p_2 \setminus p_3 \setminus p_5$
$[247.5^\circ, 292.5^\circ)$	$p_1 \setminus p_2 \setminus p_4 \setminus p_6 \setminus p_7$	$p_3 \setminus p_5 \setminus p_8$
$[292.5^\circ, 337.5^\circ)$	$p_1 \setminus p_2 \setminus p_3 \setminus p_4 \setminus p_6$	$p_5 \setminus p_7 \setminus p_8$
$[337.5^\circ, 360^\circ) \cup [0^\circ, 22.5^\circ)$	$p_1 \setminus p_2 \setminus p_3 \setminus p_4 \setminus p_5$	$p_6 \setminus p_7 \setminus p_8$

1.2.3 关键节点提取策略

针对传统A*算法在规划得到最优路径之后，最优路径仍然存在拐点冗余的问题，本文对规划后的路径采用关键节点提取算法，通过判断节点之间的连线是否经过障碍物，有效去除无效拐点。关键节点提取如图2所示，算法的具体实现步骤如下。

步骤1：连接 p_1 和 p_3 节点，计算 p_1 和 p_3 的连接线是否穿过障碍物，如未穿过障碍物，则将规划后路径中的 p_2 节点删除；

步骤2：连接 p_1 和 p_4 节点，计算 p_1 和 p_4 的连接线是否穿过障碍物，如果 p_1 和 p_4 的连接线穿过障碍物，则 p_3 为关键拐点，不可删除；

步骤3：将 p_3 作为关键路径选择的新起始节点，重复计算步骤1和步骤2，直到遍历完规划后

路径中的所有节点为止;

步骤4: 将删除冗余拐点后的路径作为规划后的最终路径。

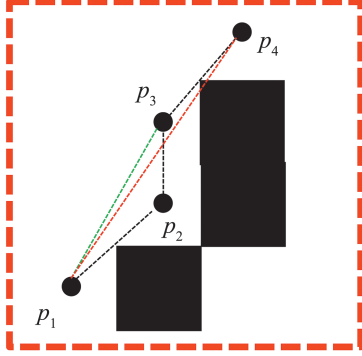


图2 关键节点提取
Fig. 2 Key node extraction

2 改进动态窗口法

动态窗口法是一种局部路径规划算法, 通过对机器人速度信息进行采样, 得到在规定速度区间内的各个速度值, 然后针对每个速度值生成对应的预测轨迹。通过轨迹评价函数进行轨迹评估, 从所有采样的预测轨迹中选择出一条最优的轨迹来指导机器人移动。针对该算法缺乏全局指引容易陷入局部最优的问题, 向传统动态窗口法的评价函数中引入偏离距离函数, 使机器人移动时更加贴近全局最优路径。

2.1 机器人运动轨迹模型

动态窗口模型是一种模拟机器人运动轨迹的方法, 它基于机器人的初始位姿和速度信息, 通过设定一段时间间隔 Δt 内的匀速运动, 生成机器人可能的动作轨迹信息。

$$\begin{cases} x_n = x_{n-1} + v\Delta t \cos \theta_{n-1} \\ y_n = y_{n-1} + v\Delta t \sin \theta_{n-1} \\ \theta_n = \theta_{n-1} + \omega\Delta t \end{cases} \quad (5)$$

式中: x_n 、 y_n 、 θ_n 为机器人在下一时刻的位姿信息; x_{n-1} 、 y_{n-1} 、 θ_{n-1} 为机器人在当前时刻的位置信息; v 、 ω 为机器人的线速度和角速度, Δt 为维

持匀速运动的时间。

2.2 速度采样

在速度环境中, 有多组可以选择的线速度和角速度, 由于机器人存在自身的性能和周围环境的限制, 需要对采样速度的范围进行限制。

(1) 机器人自身速度约束

$$v_m = \{(v, \omega) \mid v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (6)$$

式中: v_{\min} 、 v_{\max} 为机器人最小线速度和最大线速度; ω_{\min} 、 ω_{\max} 为机器人最小角速度和最大角速度。

(2) 机器人电机性能加减速约束

$$v_d = \{(v, \omega) \mid v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t], \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t]\} \quad (7)$$

式中: \dot{v}_b 、 \dot{v}_a 为最小线加速度和最大线加速度, $\dot{\omega}_b$ 、 $\dot{\omega}_a$ 为最小角加速度和最大角加速度; v_c 为初始线速度; ω_c 为初始角速度。

(3) 机器人安全制动约束

$$v_a = \{(v, \omega) \mid v \leq \sqrt{2\text{dist}(v, \omega)\dot{v}_b}, \omega \leq \sqrt{2\text{dist}(v, \omega)\dot{\omega}_b}\} \quad (8)$$

式中: $\text{dist}(v, \omega)$ 为在当前速度 (v, ω) 时距离障碍物的最小距离。

最终动态窗口的采样速度需要满足集合 v_r , 即满足上述3种约束的集合:

$$v_r = v_m \cap v_d \cap v_a \quad (9)$$

基于对线速度和角速度样本数量的考量, 对连续速度矢量空间 v_r 进行离散化处理, 以获得离散采样点 (v, ω) 。在获得这些采样点之后, 利用机器人运动学模型对每个采样点进行分析, 以预测机器人在下一个时刻的运动轨迹, 如图3所示。

2.3 优化评价函数

在速度信息采样预测的轨迹中, 除去已经与障碍物发生碰撞的轨迹, 还存在许多可选的运动轨迹, 需要通过评价函数对这些轨迹进行计算, 筛选出评价指标最高的运动轨迹, 传统的评价函数为

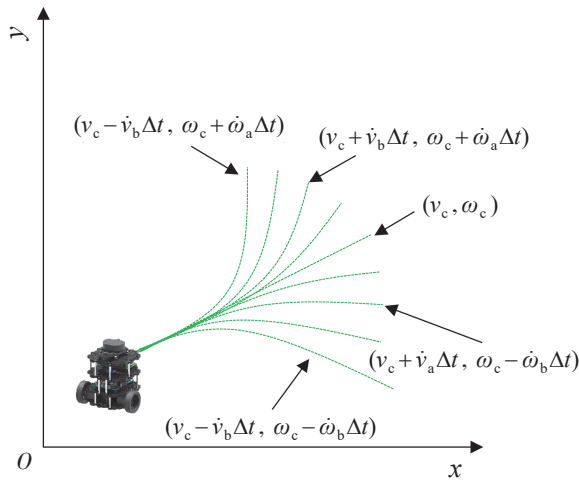


图3 轨迹预测
Fig. 3 Trajectory prediction

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega)) \quad (10)$$

式中： $\text{heading}(v, \omega)$ 为方位角评价函数，表示在当前选择的速度组模拟的轨迹末端朝向与目标点位置航向角之间的角度偏差，角度偏差越小取值越大； $\text{dist}(v, \omega)$ 为障碍物评价函数，表示在当前选择的速度组模拟的轨迹距离障碍物的最小距离，距离越大取值越大； $\text{velocity}(v, \omega)$ 为速度评价函数，表示在当前选择的速度组模拟的轨迹的速度大小，速度越大取值越大； σ 为平滑系数， α 、 β 、 γ 为加权系数。

针对传统动态窗口法容易陷入局部最优，易产生偏离全局路线的现象，为确保路径的平滑性和安全性，本文在其原有评价函数的基础上，引入偏离距离评价函数，即计算轨迹路线与全局路线之间的距离，将评价函数修改为

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega) + \delta \cdot \text{dist_off}(v, \omega)) \quad (11)$$

式中： $\text{dist_off}(v, \omega)$ 为在当前选择的速度组模拟的轨迹路线与全局路线之间的距离，距离越小取值越大。

2.4 改进A*算法与改进动态窗口法融合

本文将改进A*算法与改进动态窗口法相结合，使用改进A*算法进行全局路径规划，得到最优路径；对最优路径使用关键点提取策略进行关键点提取，剔除不必要的冗余拐点；将提取后的

关键节点作为改进动态窗口法的全局指导，在规定的速度区间内按采样频率进行采样，得到预测轨迹，利用改进评价函数选择出最优的轨迹，达到局部路径优化的效果，具体流程如图4所示。

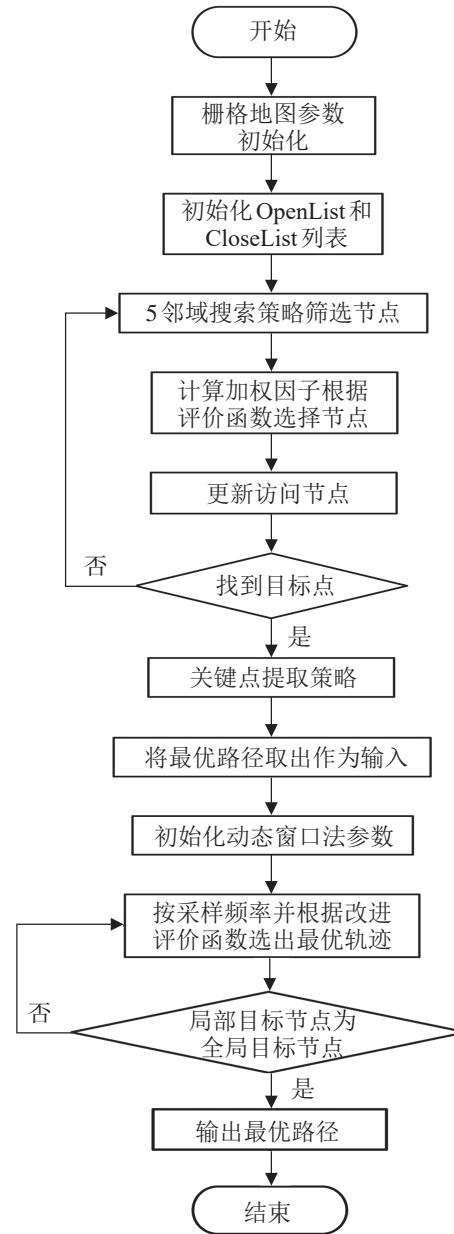


图4 融合算法流程图
Fig. 4 Flowchart of fusion algorithm

3 仿真与实验

3.1 实验环境

本次仿真环境由操作系统 Windows 10、显卡

NVIDIA GeForce RTX 3060、处理器 Intel(R) Core (TM) i5-10400 CPU@2.90GHz、仿真软件 MATLAB R2021a 组成。使用 MATLAB 建立路径规划区域和栅格地图, 初始化移动机器人属性以及动态窗口法初始属性如表2所示。

表2 移动机器人属性以及动态窗口法初始属性
Table 2 Mobile robot properties and initial properties of the dynamic window method

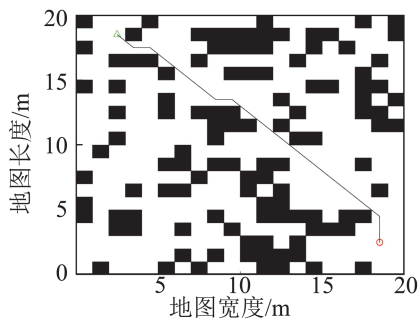
参数	数值	参数	数值
最大线速度 $v_{\max}/(\text{m/s})$	1.0	方位角评价系数	0.2
最大角速度 $\omega_{\max}/((^\circ)/\text{s})$	20	障碍物距离评价系数	0.1
线加速度 $\dot{v}/(\text{m/s}^2)$	0.2	速度评价系数	0.3
角加速度 $\dot{\omega}/((^\circ)/\text{s}^2)$	50	偏移距离评价系数	0.4
线速度分辨率 $d_v/(\text{m/s})$	0.01	采样时间/s	3
角速度分辨率 $d_\omega/((^\circ)/\text{s})$	1	距离阈值/m	0.5

3.2 改进A*算法仿真实验及分析

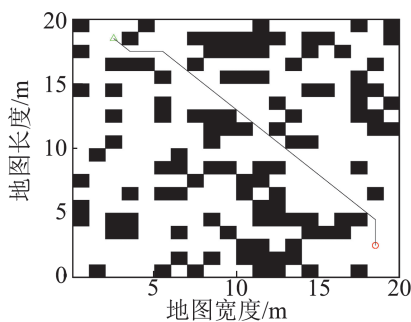
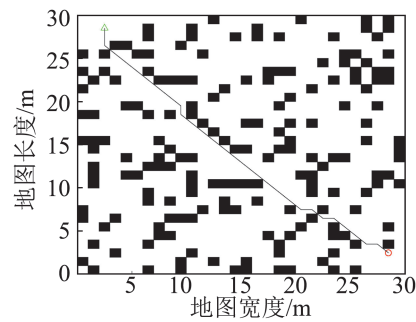
为了验证改进A*算法的有效性, 采取不同环境进行仿真实验, 将本文改进A*算法与传统A*

算法、文献[16]改进A*算法进行对比; 仿真实验环境为二维栅格地图, 黑色栅格为障碍物节点、白色栅格为可通过节点、绿色三角形为起始节点、红色圆圈为目标节点。仿真实验分别设置了2组障碍率为20%但规格不同的地图, 仿真实验结果如图5所示, 数据对比如表3所示。

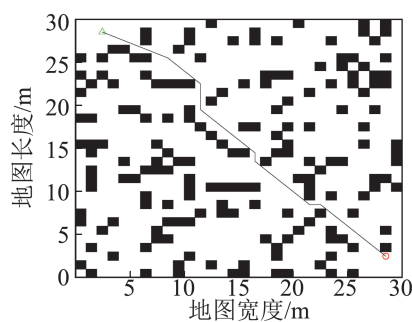
分析表3数据可以得到以下结论: 在栅格规模为 20×20 的环境下, 在路径长度方面, 相比传统A*算法文献[16]中改进A*算法减少0.21%, 本文改进A*算法减少1.07%; 在算法运算时间方面, 相比传统A*算法文献[16]改进A*算法减少16%, 本文改进A*算法减少36%。在栅格规模为 30×30 的环境下, 路径长度方面, 文献[16]中改进A*算法减少0.13%, 本文改进A*算法减少0.69%; 在算法运算时间方面, 文献[16]改进A*算法减少34%, 本文改进A*算法减少67%。综上所述, 本文改进A*算法随着地图栅格规模的增加, 虽然在最优路径长度方面的减少百分比并没有逐渐增加, 但在算法运算时间方面的减少百分比增加更加明显, 可以有效提高路径规划的运行效率。



(a) 传统A*算法



(b) 文献[16]改进A*算法



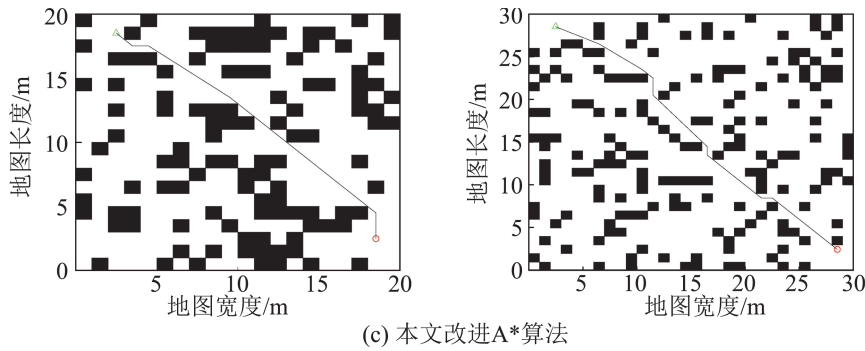


图5 不同A*算法路径对比

Fig. 5 Comparison of A* algorithm paths

表3 改进算法结果对比

Table 3 Comparison of results of improved algorithms			
地图规格	算法名称	路径长度/m	运算时间/s
20×20	传统A*算法	23.799	0.025
	文献[16]改进A*算法	23.748	0.021
	本文改进A*算法	23.545	0.016
30×30	传统A*算法	39.527	0.085
	文献[16]改进A*算法	39.476	0.056
	本文改进A*算法	39.253	0.028

3.3 融合算法仿真实验及分析

为验证本文改进动态窗口法以及本文融合算法的有效性,将本文改进A*算法分别与传统动态窗口法、文献[17]中改进动态窗口法、本文改进动态窗口法相融合,并分别在不同的实验环境下进行实验对比。仿真实验环境为二维栅格地图,黑色栅格为障碍物节点、白色栅格为可通过节点、绿色三角形为起始节点、红色圆圈为目标节点。仿真实验设置的地图为有效障碍率为20%的20×20规格地图和有效障碍率为30%的30×30规格地图。仿真实验结果如图6所示,数据对比如表4所示。

由于改进A*算法规划后的路径依然不是很平滑,故使用动态窗口法再次进行平滑操作。融合传统DWA算法和融合文献[17]改进DWA算法均发生碰撞障碍物情况,融合本文改进DWA算法没有出现碰撞情况。分析表4数据可以得到以下结论:在栅格规模为20×20且有效障碍率为20%的环境下,在路径长度方面,融合本文改进DWA算法相比于融合传统DWA算法减少1.35%,相比于

融合文献[17]改进DWA算法减少1.22%;在运行时间方面,融合本文改进DWA算法相比融合传统DWA算法增加6.83%,相比融合文献[17]改进DWA算法增加5.4%。在栅格规模为30×30且有效障碍率为30%的环境下,路径长度方面,融合本文改进DWA算法相比于融合传统DWA算法减少1.98%,相比于融合文献[17]改进DWA算法减少1.93%;在算法运行时间方面,融合本文改进DWA算法相比于融合传统DWA算法增加4.74%,相比于融合文献[17]改进DWA算法增加5.83%。综上所述,随着地图规模的增加,融合本文改进DWA算法虽然在运算时间方面有所增长,但在路径长度方面减少明显。

在物理学中,姿态角度、线速度和角速度是描述物体运动的重要物理量,其方差和标准差反映有关运动特性的统计信息。角度是描述物体朝向或者旋转状态的物理量,方差和标准差分别蕴含角度均值及其变化的不确定性或稳定性信息,方差和标准差越小,表示角度均值和角度变化越小,旋转相对稳定。速度(含线速度和角速度)是描述物体在某一时间点的位置变化率,方差和标准差可以提供关于速度均值及其变化的不确定性或稳定性信息,方差和标准差越小,表示速度均值和速度变化越小,运动相对稳定。计算融合算法移动机器人的姿态角度、角速度和线速度的方差以及标准差,如图7所示。

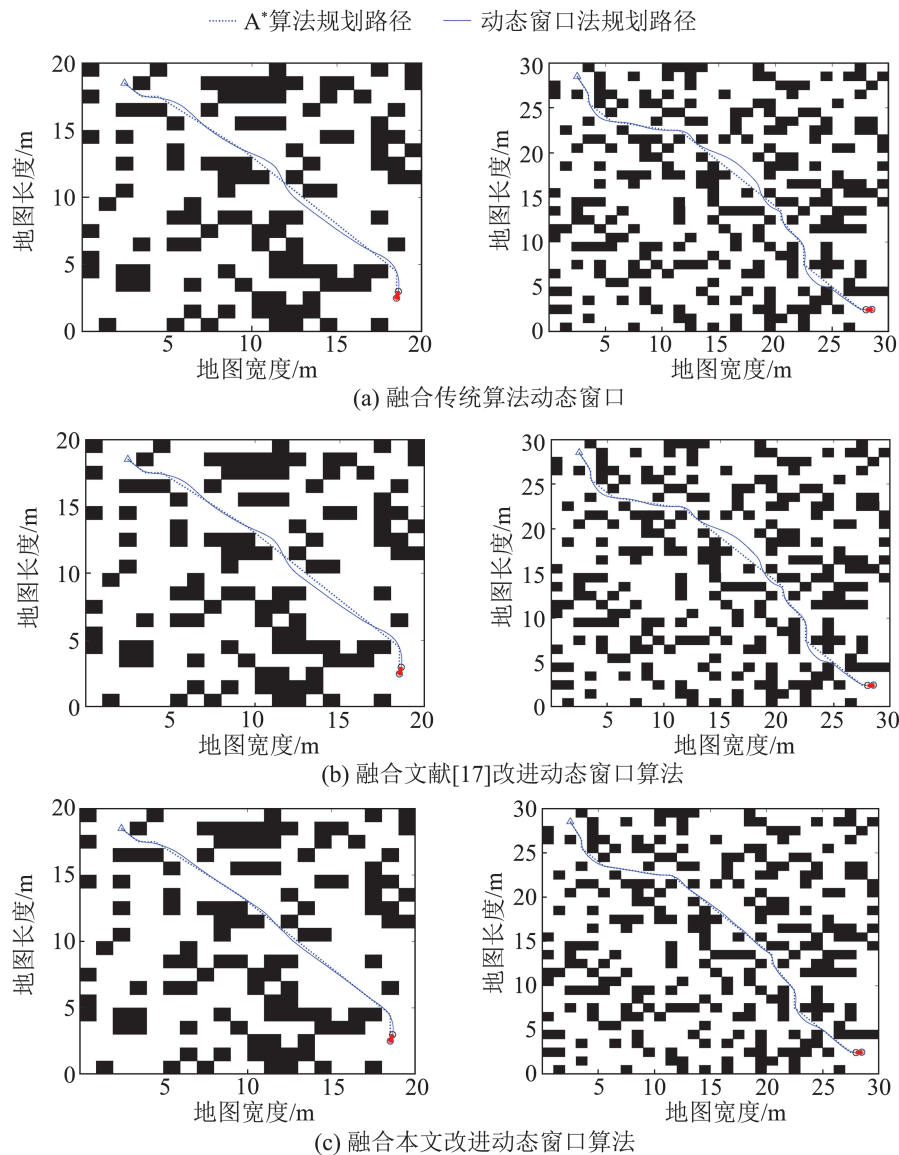


图6 不同融合动态窗口算法路径对比

Fig. 6 Comparison of fusion algorithm paths

表4 融合算法结果对比

Table 4 Comparison of fusion algorithm results			
地图规格	算法名称	路径长度/m	运算时间/s
20×20	融合传统DWA算法	24.529	143.04
	融合文献[17]改进DWA算法	24.497	144.99
	融合本文改进DWA算法	24.197	152.82
	融合传统DWA算法	40.247	527.12
30×30	融合文献[17]改进DWA算法	40.226	521.69
	融合本文改进DWA算法	39.449	552.09

通过在地图规格为20×20有效障碍率为20%和地图规格为30×30有效障碍率为30%的二维栅格地图环境下进行实验, 分别计算融合算法规划后机器人状态中的角度、线速度和角速度的方差和标准差, 运算结果如表5~7所示。

分析表5~7可得, 基于本文所提融合算法规划得到的移动机器人角度、线速度和角速度的方差和标准差都更小, 说明移动机器人运动过程中角度、线速度和角速度的总体波动幅度更小, 运动更加平稳。

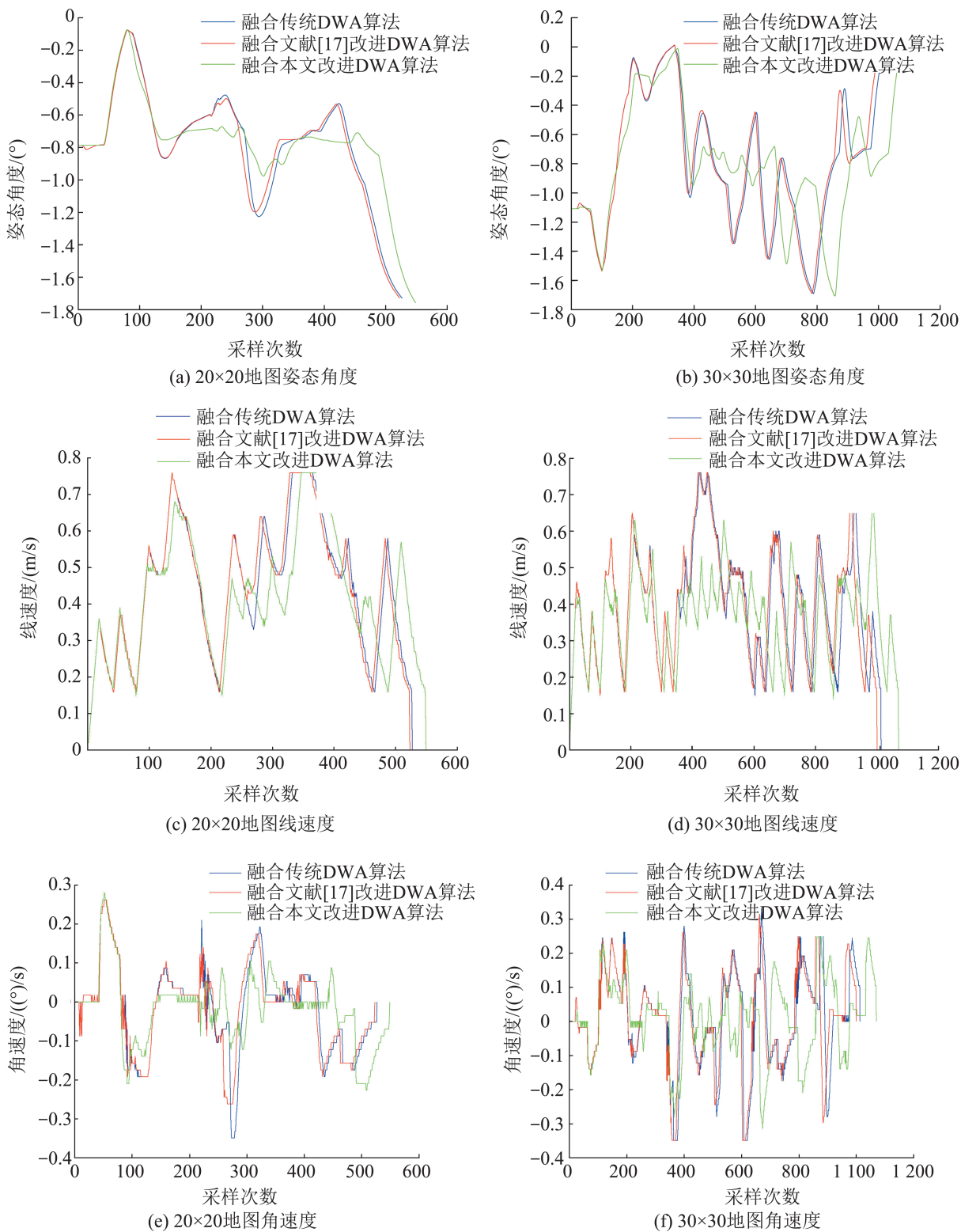


图 7 不同融合算法运动状态对比
Fig. 7 Comparison of motion states of fusion algorithms

表5 不同融合算法角度变化
Table 5 Fusion algorithm angle variation

地图规格	算法名称	角度方差	角度标准差
20×20	融合传统DWA算法	0.341 8	0.116 8
	融合文献[17]改进DWA算法	0.339 3	0.115 1
	融合本文改进DWA算法	0.292 0	0.085 3
	融合传统DWA算法	0.442 2	0.195 5
30×30	融合文献[17]改进DWA算法	0.443 1	0.196 3
	融合本文改进DWA算法	0.392 3	0.153 9

表6 不同融合算法线速度变化
Table 6 Fusion algorithm line speed change

地图规格	算法名称	线速度方差	线速度标准差
20×20	融合传统DWA算法	0.179 5	0.032 2
	融合文献[17]改进DWA算法	0.180 1	0.032 5
	融合本文改进DWA算法	0.168 0	0.028 2
	融合传统DWA算法	0.153 3	0.023 5
30×30	融合文献[17]改进DWA算法	0.152 7	0.023 3
	融合本文改进DWA算法	0.111 8	0.012 5

表7 不同融合算法角速度变化
Table 7 Fusion algorithm angular velocity change

地图规格	算法名称	角速度方差	角速度标准差
20×20	融合传统DWA算法	0.114 9	0.013 2
	融合文献[17]改进DWA算法	0.112 1	0.012 6
	融合本文改进DWA算法	0.090 3	0.008 2
	融合传统DWA算法	0.140 6	0.019 8
30×30	融合文献[17]改进DWA算法	0.141 2	0.019 9
	融合本文改进DWA算法	0.110 5	0.012 2

4 结论

本文介绍一种结合改进A*算法和改进动态窗

口法的融合算法。首先将传统A*算法8邻域搜索策略更改为5邻域搜索策略,其次向其评价函数的启发函数中添加动态加权因子,然后对规划得到的路径进行关键节点提取,去除冗余拐点,得到一条拐点较少的路径。最后,改进动态窗口法的评价函数,将改进A*算法得到的路径作为改进动态窗口法的全局路径指引,得到最优平滑路径。经过仿真实验分析可得,基于本文所提融合算法得到的规划路径更加平滑、运动角度和速度变化更加平缓,有效提高移动机器人的运动稳定性和安全性。

然而,论文仍存在以下不足:一方面,加权因子根据当前节点与目标节点间的距离动态调整,以平衡搜索速度与搜索效率,实际应用中可能导致调试困难,未来将进一步探索如何根据具体问题自适应确定加权因子;另一方面,虽然动态窗口法有效提高了路径平滑度,但可能导致计算复杂度增加,拟将对算法计算效率进行深入分析和优化。

参考文献:

- [1] Tang Meiqin, Sheng Jiawen, Sun Shaoyan. A Coverage Optimization Algorithm for Underwater Acoustic Sensor Networks Based on Dijkstra Method[J]. IEEE/CAA Journal of Automatica Sinica, 2023, 10(8): 1769-1771.
- [2] 许建民, 宋雷, 邓冬冬, 等. 基于多尺度A*与优化DWA算法融合的移动机器人路径规划[J/OL]. 系统仿真学报. (2023-11-22) [2024-04-24]. <https://doi.org/10.16182/j.issn1004731x.joss.23-1089>.
Xu Jianmin, Song Lei, Deng Dongdong, et al. Path Planning of Mobile Robot Based on the Integration of Multi-scale A* and Optimized Dynamic-window Approach Algorithm[J/OL]. Journal of System Simulation. (2023-11-22) [2024-04-24]. <https://doi.org/10.16182/j.issn1004731x.joss.23-1089>.
- [3] 黄鲁, 周非同. 基于路径优化D* Lite算法的移动机器人路径规划[J]. 控制与决策, 2020, 35(4): 877-884.
Huang Lu, Zhou Feitong. Path Planning of Moving Robot Based on Path Optimization of D* Lite Algorithm [J]. Control and Decision, 2020, 35(4): 877-884.
- [4] 常路, 单梁, 戴跃伟, 等. 未知环境下基于改进DWA的多机器人编队控制[J]. 控制与决策, 2022, 37(10): 2524-

2534.
Chang Lu, Shan Liang, Dai Yuewei, et al. Multi-robot Formation Control in Unknown Environment Based on Improved DWA[J]. Control and Decision, 2022, 37(10): 2524-2534.
- [5] 鲜斌, 宋宁. 基于模型预测控制与改进人工势场法的多无人机路径规划[J/OL]. 控制与决策. (2024-01-02) [2024-02-01]. <https://doi.org/10.13195/j.kzyjc.2023.0892>. Xian Bin, Song Ning. A Multiple UAVs Path Planning Method Based on Model Predictive Control and Improved Artificial Potential Field[J/OL]. Control and Decision. (2024-01-02) [2024-02-01]. <https://doi.org/10.13195/j.kzyjc.2023.0892>.
- [6] 崔宝侠, 王淼弛, 段勇. 基于可搜索24邻域的A*算法路径规划[J]. 沈阳工业大学学报, 2018, 40(2): 180-184. Cui Baoxia, Wang Miaochi, Duan Yong. Path Planning for A* Algorithm Based on Searching 24 Neighborhoods [J]. Journal of Shenyang University of Technology, 2018, 40(2): 180-184.
- [7] 槐创锋, 郭龙, 贾雪艳, 等. 改进A*算法与动态窗口法的机器人动态路径规划[J]. 计算机工程与应用, 2021, 57(8): 244-248. Huai Chuangfeng, Guo Long, Jia Xueyan, et al. Improved A* Algorithm and Dynamic Window Method for Robot Dynamic Path Planning[J]. Computer Engineering and Applications, 2021, 57(8): 244-248.
- [8] Min Haitao, Xiong Xiaoyong, Wang Pengyu, et al. Autonomous Driving Path Planning Algorithm Based on Improved A* Algorithm in Unstructured Environment[J]. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2021, 235(2/3): 513-526.
- [9] 姬鹏, 张新元, 高帅轩, 等. 融合改进A*算法与动态窗口法的路径规划研究[J/OL]. 系统仿真学报. (2023-09-07) [2024-03-13]. <https://doi.org/10.16182/j.issn1004731x.joss.23-0619>. Ji Peng, Zhang Xinyuan, Gao Shuaxuan, et al. Path Planning Based on Improved A* and Dynamic Window Approach[J/OL]. Journal of System Simulation. (2023-09-07) [2024-03-13]. <https://doi.org/10.16182/j.issn1004731x.joss.23-0619>.
- [10] Lai Rongshen, Wu Zhiyong, Liu Xiangui, et al. Fusion Algorithm of the Improved A* Algorithm and Segmented Bézier Curves for the Path Planning of Mobile Robots[J]. Sustainability, 2023, 15(3): 2483.
- [11] 王彬, 聂建军, 李海洋, 等. 优化A*与动态窗口法的移动机器人路径规划[J]. 计算机集成制造系统, 2024, 30(4): 1353-1363.
- Wang Bin, Nie Jianjun, Li Haiyang, et al. Mobile Robot Path Planning Based on Optimized A* and Dynamic Window Approach[J]. Computer Integrated Manufacturing Systems, 2024, 30(4): 1353-1363.
- [12] 王永雄, 田永永, 李璇, 等. 穿越稠密障碍物的自适应动态窗口法[J]. 控制与决策, 2019, 34(5): 927-936. Wang Yongxiong, Tian Yongyong, Li Xuan, et al. Self-adaptive Dynamic Window Approach in Dense Obstacles [J]. Control and Decision, 2019, 34(5): 927-936.
- [13] 张伟龙, 单梁, 常路, 等. 基于改进DWA的多无人水面艇分布式避碰算法[J]. 控制与决策, 2023, 38(4): 951-962. Zhang Weilong, Shan Liang, Chang Lu, et al. Distributed Collision Avoidance Algorithm for Multiple Unmanned Surface Vessels Based on Improved DWA[J]. Control and Decision, 2023, 38(4): 951-962.
- [14] 邹文, 韩丙辰, 李鹏飞, 等. 融合改进A*算法和优化动态窗口法的路径规划[J]. 计算机集成制造系统, 2024, 30(1): 184-195. Zou Wen, Han Bingchen, Li Pengfei, et al. Path Planning by Integrating Improved A* Algorithm and Optimized Dynamic Window Approach[J]. Computer Integrated Manufacturing Systems, 2024, 30(1): 184-195.
- [15] 程传奇, 郝向阳, 李建胜, 等. 融合改进A*算法和动态窗口法的全局动态路径规划[J]. 西安交通大学学报, 2017, 51(11): 137-143. Cheng Chuanqi, Hao Xiangyang, Li Jiansheng, et al. Global Dynamic Path Planning Based on Fusion of Improved A* Algorithm and Dynamic Window Approach [J]. Journal of Xi'an Jiaotong University, 2017, 51(11): 137-143.
- [16] 张涛, 陈璋, 李玉梅, 等. 融合改进A*算法与动态窗口法的机器人避障研究[J]. 仪表技术与传感器, 2023(4): 102-106. Zhang Tao, Chen Zhang, Li Yumei, et al. Research on Robot Obstacle Avoidance Based on Improved A* Algorithm and Dynamic Window Method[J]. Instrument Technique and Sensor, 2023(4): 102-106.
- [17] 郭园园, 袁杰, 赵克刚. 基于改进A*算法和动态窗口法的机器人路径规划[J]. 计算机工程与科学, 2022, 44(7): 1273-1281. Guo Yuanyuan, Yuan Jie, Zhao Kegang. Robot Path Planning Based on an Improved A* Algorithm and an Improved Dynamic Window Method[J]. Computer Engineering & Science, 2022, 44(7): 1273-1281.