

# 分层平滑优化 A\* 引导 DWA 用于机器人路径规划\*

朱洪波 殷宏亮

(安徽理工大学电气与信息工程学院 淮南 232001)

**摘要:**针对 A\* 算法存在的搜索效率低,路径平滑性和安全性差,以及 DWA 融合全局路径规划算法实时寻路效率低等问题,提出了一种分层平滑优化 A\* 引导 DWA(HSA\*-G-DWA)的工业机器人路径规划方法。首先,在 A\* 算法的代价函数中引入双动态加权因子并构建碰撞约束函数剔除路径搜索过程中无关扩展节点的搜索,以提升路径搜索的效率和安全性。其次,利用分层平滑优化策略消除路径中的冗余点和转折点,减少路径点数量和路径长度。之后,通过无障碍约束直线与有障碍约束圆弧插补分段优化生成初始全局路径,保证路径的安全性及平滑性。然后,若移动机器人跟踪全局路径过程中面临未知障碍物则利用全局路径引导 DWA 生成避障与返回全局路径的局部动态修正路径,减少了实时计算量。最后,仿真实验结果表明,静态环境下 HSA\*-G-DWA 算法路径搜索时间和路径点数较 A\* 算法分别平均减少了 88.43% 和 86%,路径的平滑性和安全性更好;未知环境下 HSA\*-G-DWA 算法可以实时避开环境中出现的未知障碍物,路径长度较 DWA 算法、Dijkstra 算法、RRT 算法和现有融合算法分别平均减少了 25.78%、18.65%、30.48% 和 14.59%,路径搜索时间较现有融合算法平均减少了 67.39%。

**关键词:** 移动机器人;路径规划;A\* 算法;碰撞约束;动态加权因子;分层平滑;动态避障

中图分类号: TP24;TN964

文献标识码: A

国家标准学科分类代码: 510.80

## Hierarchical smoothing optimization A\*-guided DWA for robot path planning

Zhu Hongbo Yin Hongliang

(School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan 232001, China)

**Abstract:** Aiming at the problems of low search efficiency, poor path smoothness and security of A\* algorithm, and low real-time pathfinding efficiency of DWA integrated with global path planning algorithm, a hierarchical smooth optimization A\* guided DWA (HSA\*-G-DWA) path planning method for mobile robots is proposed. Firstly, the double dynamic weighting factor is introduced into the cost function of A\* algorithm and the collision constraint is developed to avoid the search of unrelated extension nodes, so as to improve the efficiency and security of path search. Secondly, the hierarchical smoothing optimization strategy is designed to eliminate redundant nodes and turning nodes in the path, and reduce the number of path nodes and the path length. After that, the initial global path is generated by segmented interpolation of lines without any obstacle constraints and arcs with obstacle constraints to ensure the safety and smoothness of the path. Then, if the mobile robot encounters unknown obstacles in the process of tracking the global path, it uses the global path to guide DWA to generate the local dynamic correction path for obstacles avoidance and returning to the remaining global path, which reduces the amount of real-time calculation. Finally, the simulation results show that the path search time and path nodes of the proposed HSA\*-G-DWA algorithm are reduced by 88.43% and 86%, respectively, and the smoothness and security of the path are better than the A\* algorithm in the static environment; and the HSA\*-G-DWA algorithm can avoid unknown obstacles in the unknown environment in real time. Compared with the DWA algorithm, Dijkstra algorithm, RRT algorithm and other fusion algorithms, the path length is reduced by 25.78%, 18.65%, 30.48% and 14.59% on average, and the path search time is reduced by 67.39% on average.

**Keywords:** mobile robots; path planning; A\* algorithm; collision constraint; dynamic weighting factor; hierarchical smoothing; dynamic obstacle avoidance

收稿日期: 2024-04-15 Received Date: 2024-04-15

\* 基金项目: 国家自然科学基金(62003001)、安徽高校自然科学研究项目重大项目(2023AH040157)资助

## 0 引言

近年来,随着工业自动化和人工智能行业不断取得新的突破,移动机器人可以代替人类做一些危险重复的工作,提高工作效率,降低成本,因此被广泛应用于制造业、医疗、军事等多个领域。路径规划问题是移动机器人研究中的一个重要方向<sup>[1]</sup>。

安全可行的最短路径规划是实现移动机器人自主完成各项任务的前置基础技术。近年来,该领域内的学者对其作了大量的研究和实验,提出了各种可行的方法,取得了重大的进展。按照场景环境、路况复杂程度的不同以及先验环境的已知性可将移动机器人路径规划分为动态局部路径规划和静态全局路径规划<sup>[2]</sup>。全局路径规划对局部路径规划起到导向和约束作用,局部路径规划对全局路径规划提供对动态环境实时快速规划的能力。目前常用的全局路径规划算法主要有 A 星算法<sup>[3-5]</sup>(A-star algorithm, A\*)、遗传算法<sup>[6-7]</sup>(genetic algorithm, GA)、蚁群算法<sup>[8-9]</sup>(ant colony optimization, ACO)、迪杰斯特拉算法<sup>[10-11]</sup>(dijkstra's algorithm, Dijkstra)、快速扩展随机树算法<sup>[12-13]</sup>(rapidly-exploring random tree, RRT)、跳点搜索算法<sup>[14-15]</sup>(jump point search, JPS)等;常用的局部路径规划主要有动态窗口法<sup>[16-18]</sup>(dynamic window approach, DWA)、人工势场法<sup>[19-20]</sup>(artificial potential field, APF)等。其中 A\* 算法是一种启发式算法,它利用启发信息寻找最优路径,其优点在于向目标点移动时引入了启发信息作为决策辅助,因此不需要遍历整个地图,降低了计算复杂度,使路径搜索更为直接。DWA 考虑了速度和加速度的限制,是基于预测控制理论的一种次优方法,其优点在于在未知环境下能够规划出平滑安全的路径并且能够实现实时避障,因此这两种算法被广泛应用于静态和动态的路径规划。但是传统的 A\* 算法和 DWA 也存在一些缺点,如传统 A\* 算法路径平滑性差、无法应对未知障碍物且搜索效率不理想难以满足实际需要,而 DWA 只模拟并评价了下一步,易进入陷阱和局部最优且存在大量冗余路段。为了解决以上问题,大量学者对传统 A\* 算法以及 DWA 进行了改进。文献[21]通过改进区域搜索点改进了传统 A\* 算法,有效解决了路径转折问题,但并未考虑路径的搜索效率和安全性。文献[22-23]通过增加子节点选择规则改进传统 A\* 算法,避免路径斜穿过障碍物顶点,在一定程度上提升了路径的安全性,然而,所规划路径并不平滑。文献[24]改进了传统 A\* 算法的节点搜索条件并利用 B 样条曲线拟合路径,一定程度上提升了路径的安全性和平滑性,但路径搜索效率低且在未知场景中无法实现实时动态避障。文献[25-26]通过提取全局路径的关键转折点作为 DWA 的中间目标点,将

全局路径分段,再结合 DWA 对每段局部路径进行优化,弥补了全局规划算法无法应对未知障碍物以及局部规划算法全局能力低下的缺点并提升了路径的平滑性,但无论在任何环境下进行路径规划时都需要利用 DWA 从起点到终点重新进行二次规划,导致算法实时寻路效率低且存在冗余路段。

为了解决传统 A\* 算法存在的搜索效率低,路径平滑性和安全性差,以及 DWA 融合全局路径规划算法实时寻路效率低等问题,提出了一种 HSA\*-G-DWA 的移动机器人路径规划方法。首先,引入双动态加权因子改善 A\* 算法的代价函数并构建碰撞约束函数以提高路径搜索效率和安全性。接着,采用分层平滑优化策略去除冗余路径点和冗余路段并通过无障碍约束直线与有障碍约束圆弧插补分段优化生成初始路径。最后,构建局部避障路径优化生成机制,在移动机器人按照全局路径行驶过程中探测到未知障碍物时结合分层平滑的全局路径引导 DWA 完成局部动态避障,使算法能够适用各种复杂场景,从而最终提高算法效率,减少冗余路段,提升路径的安全性及平滑性。

## 1 分层平滑优化 A\* 算法

### 1.1 环境建模

创建的 42×42 地图环境 M 如图 1 所示,其中黑色方格表示不可到达障碍物,白色区域为移动机器人可自由行走的区域,算法规划完成会生成若干路径点,机器人可以在路径点之间的连线上运动。

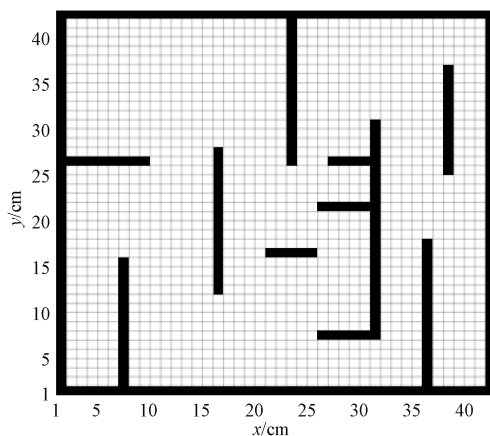


图 1 地图 M

Fig. 1 Map M

### 1.2 双动态加权和碰撞约束的 A\* 算法

为避免现有 A\* 算法的代价函数规划路径时搜索效率低下以及受机器人自身尺寸影响导致安全性差的问题,给出如下双动态加权修正的代价函数和碰撞约束的

A\* 路径规划方法:

$$\min f(\mathbf{x}_n) = w_1 \cdot g(\mathbf{x}_n) + w_2 \cdot h(\mathbf{x}_n) \quad (1)$$

$$s. t. \|\mathbf{x}_n - \mathbf{o}_j\|_1 \geq L, \quad j = 1, \dots, O \quad (2)$$

其中,

$$g(\mathbf{x}_n) = \sum_{i=2}^n \|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2 \quad (3)$$

$$h(\mathbf{x}_n) = \|\mathbf{x}_n - \mathbf{x}_{goal}\|_2 \quad (4)$$

$$w_1 = \frac{2 \|\mathbf{x}_{first} - \mathbf{x}_n\|_1}{m} \quad (5)$$

$$w_2 = 1 + \frac{(\|\mathbf{x}_n - \mathbf{x}_{goal}\|_1)^2}{4(l + m)} \quad (6)$$

式中:  $\mathbf{x}_n$  为当前节点位置,  $\mathbf{x}_i$  为第  $i$  个节点位置,  $\mathbf{x}_{first}$  为起始节点位置,  $\mathbf{x}_{goal}$  为目标节点位置,  $\mathbf{o}_j$  为障碍物列表中第  $j$  个障碍物的位置,  $O$  为障碍物的个数,  $l$  和  $m$  分别为地图的长度和宽度,  $f(\mathbf{x}_n)$  为当前节点总的移动代价,  $g(\mathbf{x}_n)$  为起始节点到当前节点的实际移动代价,  $h(\mathbf{x}_n)$  为当前节点到目标节点的估计移动代价。  $w_1$ 、 $w_2$  是分别作用于  $g(\mathbf{x}_n)$ 、 $h(\mathbf{x}_n)$  的动态加权因子, 具体作用如下:

1) 起点  $\mathbf{x}_{first}$  与终点  $\mathbf{x}_{goal}$  距离较小时,  $w_2 \gg w_1$ , 通过大幅提升  $h(\mathbf{x}_n)$  的作用使算法能够快速高效的规划出一条可行路径。

2) 起点  $\mathbf{x}_{first}$  与终点  $\mathbf{x}_{goal}$  距离较大时, 考虑搜索空间的复杂性, 在当前节点  $\mathbf{x}_n$  距离终点  $\mathbf{x}_{goal}$  较远时, 降低  $g(\mathbf{x}_n)$  的作用, 提高  $h(\mathbf{x}_n)$  的作用以极大程度提高搜索速度, 随着  $\mathbf{x}_n$  向  $\mathbf{x}_{goal}$  靠近, 逐渐增大  $g(\mathbf{x}_n)$  的作用, 降低  $h(\mathbf{x}_n)$  的作用以提高搜索精度。

$L$  为碰撞约束阈值, 阈值的选定标准具体如下:

考虑到实际应用中移动机器人自身的尺寸 (不超过一个栅格大小), 若环境地图中可通行区域的最小尺寸为不少于 3 个栅格大小, 设置碰撞约束阈值  $L = 2$ , 如图 2(a) 所示将灰色方格视为了不可搜索区域, 防止规划出紧邻障碍物边缘或与障碍物存在交点的局部不可行路径 (虚线), 进而规划出与障碍物之间存在一定安全距离的可行路径 (实线), 从而减少了移动机器人行驶过程中的碰撞风险, 保证了机器人的安全运行。否则, 设置碰撞约束阈值  $L = 1$  (即不预留安全距离), 为保证所规划的路径不会出现相切或穿越障碍物的情况, 需额外增加如图 2(c) 所示的节点选取规则, 即当障碍物位于当前节点上下左右 4 个方向之一时, 当前节点为 A, 黑色栅格为障碍物节点, 此时 B 和 C 节点不考虑作为下一扩展点, 因此不会生成 AB 和 AC 路径, 其他几个方向同理, 保证规划出图 2(b) 所示的可行路径 (实线), 可以看出机器人大多数情况下是紧邻障碍物边缘的。

为验证上述方法的正确性与合理性, 在图 1 所示的地图 M 中进行寻路搜索仿真实验, 并与引入节点选取规则的 A\* 算法对比, 所得仿真对比结果如图 3(a) 和 (b) 所

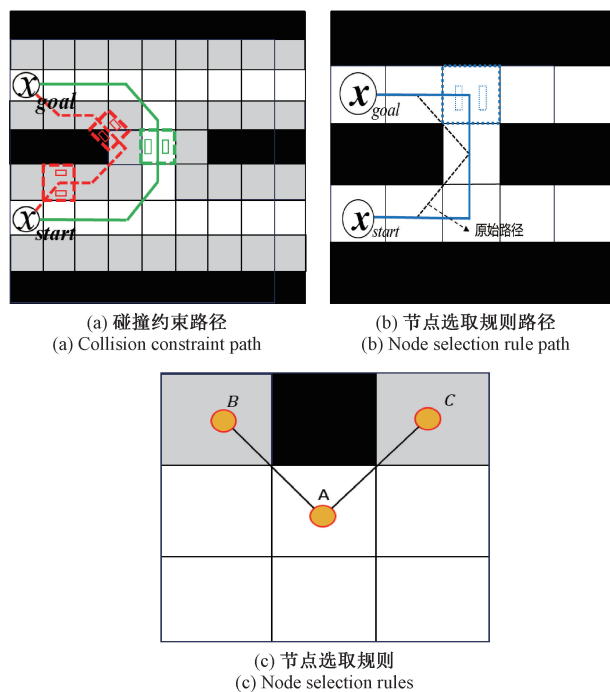


图 2 路径对比与节点选取规则

Fig. 2 Path comparison and node selection rules

示, 具体的实验数据如表 1 所示。

表 1 性能对比

Table 1 Performance comparison

| 算法               | 搜索时间/s | 搜索点数/个 | 与障碍物最小距离/cm |
|------------------|--------|--------|-------------|
| A* 算法            | 33.36  | 708    | 0.5         |
| 双动态加权和碰撞约束 A* 算法 | 3.86   | 70     | 1.41        |

表 1 可以看出改进后的搜索方法相比于引入节点选取规则的 A\* 算法, 平均搜索时间降低了 29.5 s, 与障碍物的最小距离增加 0.91 cm, 搜索点数降低 638 个。

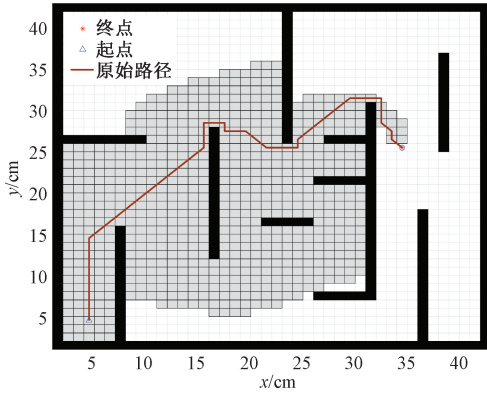
### 1.3 分层平滑优化策略

图 4 可以看出双动态加权和碰撞约束 A\* 算法生成的原始路径中存在大量的冗余路径点、冗余路段、非光滑转折角。因此, 设计并引入了分层平滑优化策略, 其可分为如图 5(a) 所示的首层冗余点优化、图 5(b) 所示的中层转折点优化和图 5(c) 所示的外层路径平滑生成优化。

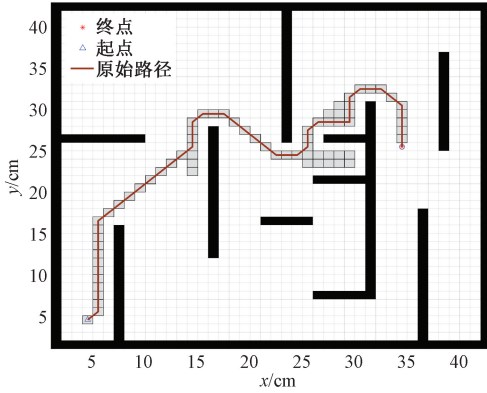
首层优化的目的是对直线路段上的路径点进行精简, 只保留使路径方向改变的节点, 以减少路径段数。通过判断相邻 3 个路径点的中间路径点是否为冗余路径点, 若是则剔除否则保留, 判别公式如式 (7) 所示。

$$f_{angle}(\mathbf{e}_{sc}, \mathbf{e}_{nc}) = \begin{cases} 1, \cos\theta = -1 \\ 0, \text{其他} \end{cases} \quad (7)$$

其中,



(a) A\*算法  
(a) A\* algorithm



(b) 双动态加权碰撞约束A\*  
(b) Double dynamic weighted collision constraint A\*

图3 改进A\*算法性能对比

Fig. 3 Comparison of improved A\* algorithm performance

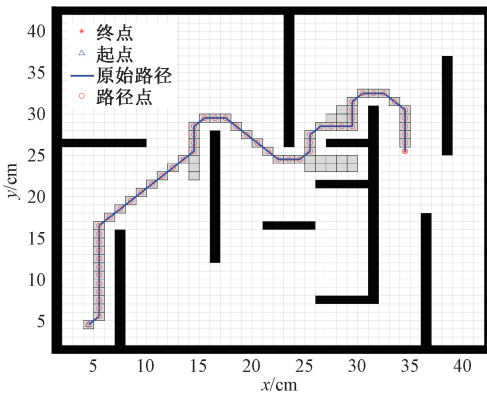


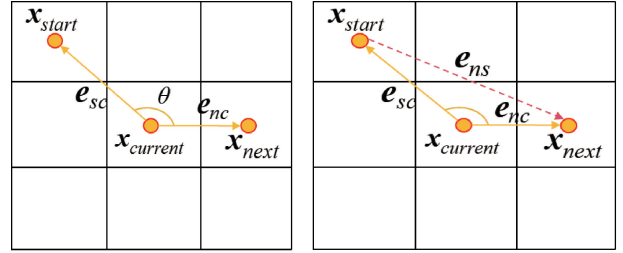
图4 原始路径点

Fig. 4 Original path point

$$\cos\theta = \mathbf{e}_{sc}^T \mathbf{e}_{nc} \quad (8)$$

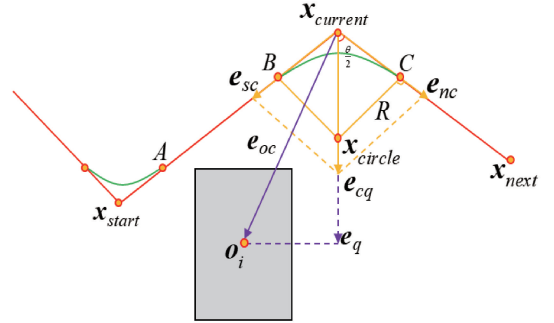
式中:  $\mathbf{e}_{sc} = \mathbf{x}_{start} - \mathbf{x}_{current}$ ,  $\mathbf{e}_{nc} = \mathbf{x}_{next} - \mathbf{x}_{current}$  为相邻3个路径点构成的向量。

中层优化的目的是消除路径中不必要的转弯,以进一步减少路径的拐角和路径的长度。通过判断相邻3个



(a) 冗余点优化  
(a) Redundancy point optimization

(b) 转折点优化  
(b) Turning point optimization



(c) 轨迹平滑生成优化  
(c) Optimization of trajectory smooth generation

图5 分层平滑优化示意图

Fig. 5 Layered smooth optimization diagram

路径点中首尾路径点连线的路径与最近障碍物中心点的距离  $d_i$  是否大于安全距离  $dist$ ,若是则以连线的路径代替原来的折线路径,否则保留原先的路径。判别公式如式(9)所示。

$$f_{judge}(\mathbf{e}_{ns}, \mathbf{e}_{sc}, \mathbf{e}_{nc}) = \begin{cases} 1, d_i > dist \text{ 且 } |\mathbf{e}_{ns}| > |\mathbf{e}_{sc}| + |\mathbf{e}_{nc}| \\ 0, \text{其他} \end{cases} \quad (9)$$

其中,

$$\mathbf{e}_{ns} = \mathbf{x}_{next} - \mathbf{x}_{start} \quad (10)$$

式中:  $\mathbf{e}_{ns}$  为首尾路径点构成的向量,若首尾路径点连线与最近障碍物的距离小于安全距离,则将其设为  $+\infty$ 。

外层优化的目的是通过直线与圆弧插补分段优化生成初始全局路径,以消除路径中的非光滑转折角,使路径平滑且安全。核心思想如下:

1) 有障碍约束内切圆弧轨迹优化生成

首先将  $\mathbf{e}_{sc}$  与  $\mathbf{e}_{nc}$  合向量的单位向量记为  $\mathbf{e}_{cq}$ ,公式如式(11)所示。

$$\mathbf{e}_{cq} = \frac{\mathbf{e}_{sc} + \mathbf{e}_{nc}}{\|\mathbf{e}_{sc} + \mathbf{e}_{nc}\|_2} \quad (11)$$

之后将当前节点  $\mathbf{x}_{current}$  到距  $\mathbf{x}_{current}$  最近障碍物位置的方向向量记为  $\mathbf{e}_{oc}$ ,向量  $\mathbf{e}_{oc}$  在向量  $\mathbf{e}_{cq}$  方向的投影长度记为  $u$ ,具体公式如式(12)所示。

$$u = (\mathbf{o}_i - \mathbf{x}_{current})^T \mathbf{e}_{cq} \quad (12)$$

其中,



$$i = \arg \min_{j=1,\dots,O} \| \mathbf{o}_j - \mathbf{x}_{\text{current}} \|_2 \quad (13)$$

$$s. t. (\mathbf{e}_{oc} - \mathbf{e}_{sc})^T (\mathbf{e}_{oc} - \mathbf{e}_{nc}) \leq 0 \quad (14)$$

最后得到内切圆圆心  $\mathbf{x}_{\text{circle}}$ , 半径  $R$  及切点  $\mathbf{x}_C$ , 具体生成公式如式 (15) ~ (17) 所示。

$$\mathbf{x}_{\text{circle}} = \mathbf{x}_{\text{current}} + \frac{u}{2} \mathbf{e}_{cq} \quad (15)$$

$$R = \frac{u}{2} \sin \frac{\theta}{2} \quad (16)$$

$$\mathbf{x}_C = \mathbf{x}_{\text{current}} + \frac{u}{2} \cos \frac{\theta}{2} \mathbf{e}_{nc} \quad (17)$$

同理可求得  $\mathbf{x}_A, \mathbf{x}_B$ 。

## 2) 转接速度约束

实际应用中当移动机器人从图 6(b) 所示的直线路段进入圆弧路段时, 机器人可能不会按照事先规划的路径行驶, 因此需要根据一定的限制条件对直线路段以及两段轨迹转接处速度进行设定, 使移动机器人实现两段轨迹之间的安全高速运动。

针对以上存在的问题, 使用图 6(a) 所示的圆弧插补法得到两端轨迹间的转接速度, 图中  $DE, EF$  为两相邻插补步长,  $E_d$  为插入圆弧的半径允许误差。

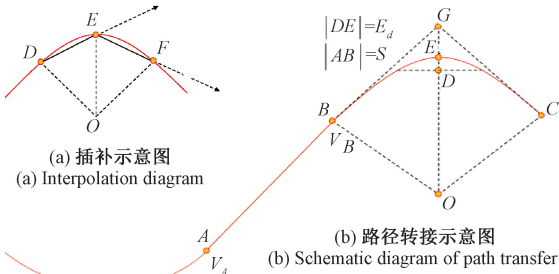


图6 转接速度及圆弧插补示意图

Fig. 6 Transfer speed and interpolation schematic diagram

由于采用圆弧转接方法时, 轨迹转接速度也就是圆弧的插补速度  $V_B$ , 此速度由以下 3 个条件限制。

首先为保证相邻两轨迹段  $BG, GC$  转接处速度矢量的变化分散到多个插补周期内, 需满足式 (18):

$$V_1 \leq \frac{2R}{T} \cos \frac{\theta}{2} \quad (18)$$

式中:  $T$  为插补周期。

之后考虑圆弧半径误差限制, 当插入圆弧半径较小时, 由圆弧半径允许轨迹误差限制条件决定的转接速度如式 (19) 所示。

$$V_2 \leq \frac{2R}{T} \sqrt{1 - (1 - \frac{E_d}{R})} \quad (19)$$

最后还需考虑加速度的限制, 当圆弧半径较大时, 由于插补步长远小于圆弧半径, 相邻两步长的转角可以忽略, 因此无需考虑加速度的限制, 当圆弧半径较小时则需

考虑, 在满足转接速度限制条件下其插补速度如式 (20) 所示。

$$V_3 \leq \sqrt{a_B R} \quad (20)$$

其中,

$$a_B = \min(a_{BG-\max}, a_{GC-\max}) \quad (21)$$

式中:  $a_{BG-\max}, a_{GC-\max}$  分别为路径  $BG, GC$  最大允许的加速度。综上轨迹间转接速度由式 (22) 所示。

$$V_B = \min\{V_1, V_2, V_3\} \quad (22)$$

由式 (18) ~ (22) 可得:

$$a_B = \frac{a_B(\mathbf{x}_O - \mathbf{x}_B)}{\|\mathbf{x}_O - \mathbf{x}_B\|} \quad (23)$$

$$V_B = \frac{V_B(\mathbf{x}_B - \mathbf{x}_A)}{\|\mathbf{x}_B - \mathbf{x}_A\|} \quad (24)$$

同理可求得  $a_A, V_A$ 。

## 3) 无障碍约束直线轨迹优化生成

为确保实际应用中移动机器人能够顺利通过直线路段, 机器人在直线路段内用 5 次多项式表示轨迹, 具体公式如式 (25) 所示。

$$\mathbf{x}(t) = \sum_{i=0}^5 (t - t_A) p_i, t_A \leq t \leq t_B \quad (25)$$

式中:  $p_i$  为直线路段中时间参数的系数,  $t_A, t_B$  分别为到达直线路径两端路径点的时刻。

为满足机器人在路径规划中的安全性和自身的物理性质规律, 机器人在直线路段内应该满足如下约束:

$$\arg \min_{\mathbf{x}(t)} J = \int_{t_A}^{t_B} \frac{d^3 \mathbf{x}^T(t)}{dt^3} \frac{d^3 \mathbf{x}(t)}{dt^3} dt \quad (26)$$

$$s. t. \begin{cases} \frac{d^2}{dt^2} \mathbf{x}(t_A) = \mathbf{a}_A \\ \frac{d^2}{dt^2} \mathbf{x}(t_B) = \mathbf{a}_B \\ \frac{d}{dt} \mathbf{x}(t_A) = \mathbf{V}_A \\ \frac{d}{dt} \mathbf{x}(t_B) = \mathbf{V}_B \\ \mathbf{x}(t_A) = \mathbf{x}_A \\ \mathbf{x}(t_B) = \mathbf{x}_B \end{cases} \quad (27)$$

HSA\* 算法步骤如下:

算法 1 (HSA\* 算法)

首层优化

步骤 1) 利用地图环境、起点、终点和式 (1) ~ (6) 获取初始全局路径点并存入路径点列表 Path 中, 将起始相邻的 3 个路径点记为  $\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{current}}, \mathbf{x}_{\text{next}}$ 。

步骤 2) 判断  $\mathbf{x}_{\text{next}}$  是否为终点, 若不是, 则继续; 否则, 跳至步骤 4)。

步骤 3) 将 3 个路径点代入式 (7) 计算, 若式 (7) 为 1, 则将  $\mathbf{x}_{\text{current}}$  从 Path 中移除并记  $\mathbf{x}_{\text{next}}$  为  $\mathbf{x}_{\text{current}}, \mathbf{x}_{\text{next}}$  下一

个路径点为  $x_{next}$  ;若式(7)为 0,则保留  $x_{current}$  并将  $x_{start}$  之后相邻的 3 个路径点记为  $x_{start}$  ,  $x_{current}$  ,  $x_{next}$  ;返回步骤 2)。

中层优化

步骤 4) 首层优化后 Path 中起始 3 个相邻路径点记为  $x_{start}$  ,  $x_{current}$  ,  $x_{next}$  。

步骤 5) 判断  $x_{next}$  是否为终点,若不是,则继续;否则,跳至步骤 8)。

步骤 6) 将 3 个路径点代入式(7),若式(7)为 1,则移除  $x_{current}$  并将  $x_{next}$  记为  $x_{current}$  ,  $x_{next}$  下一个路径点记为  $x_{next}$  ,然后跳回步骤 5);若式(7)为 0,则继续。

步骤 7) 将 3 个路径点代入式(9),若式(9)为 1,则保留  $x_{current}$  并将  $x_{start}$  之后相邻的 3 个路径点记为  $x_{start}$  ,  $x_{current}$  ,  $x_{next}$  ;若式(9)为 0,则将  $x_{current}$  从 Path 中移除并将  $x_{next}$  记为  $x_{current}$  ,  $x_{next}$  下一个路径点记为  $x_{next}$  ;跳回步骤 5)。

外层优化

步骤 8) 中层优化后 Path 中起始 3 个相邻路径点记为  $x_{start}$  ,  $x_{current}$  ,  $x_{next}$  。

步骤 9) 利用式(11)~(24)优化生成圆弧轨迹并记录圆弧轨迹起始路径点,之后由式(25)~(27)优化生成直线轨迹。

步骤 10) 判断  $x_{next}$  是否为终点,若是,则结束外层优化,否则将  $x_{start}$  之后相邻的 3 个路径点记为  $x_{start}$  ,  $x_{current}$  ,  $x_{next}$  后返回步骤 9)。

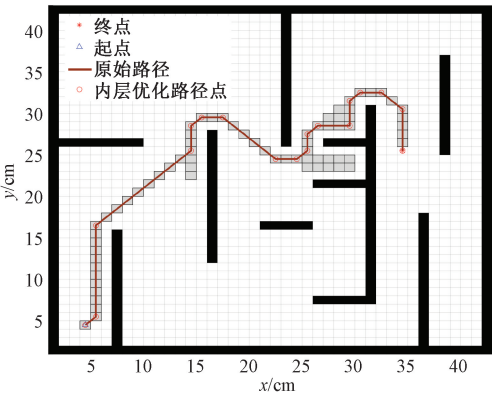
1.4 HSA\* 算法优化对比分析

为了验证上述 HSA\* 算法的可行性,在 matlab 中用图 1 M 所示地图进行仿真分析验证。所得对比仿真结果以及实验数据如图 7(a)~(c)和表 2 所示。

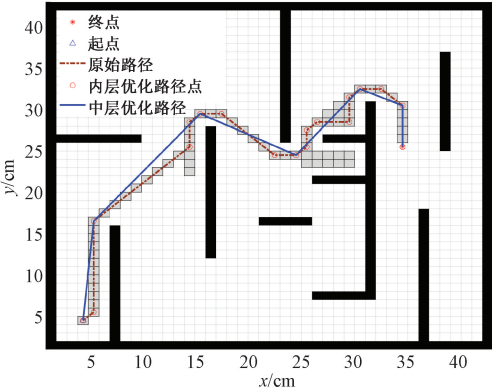
表 2 分层平滑优化前后性能对比

| Table 2 Performance comparison before and after hierarchical smoothing optimization |     |       |        |       |
|-------------------------------------------------------------------------------------|-----|-------|--------|-------|
| 算法                                                                                  | 路径点 | 路径转折点 | 非光滑转折点 | 路径长度/ |
|                                                                                     | 数量  | 数量    | 数量     | cm    |
| A* 算法                                                                               | 50  | 14    | 14     | 57.28 |
| 内层优化                                                                                | 18  | 15    | 15     | 62.69 |
| 中层优化                                                                                | 7   | 5     | 5      | 58.21 |
| 外层优化                                                                                | 7   | 5     | 0      | 55.83 |

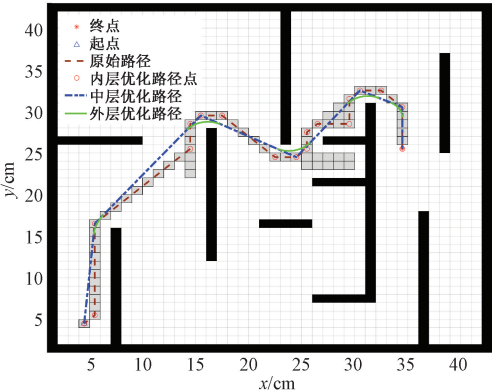
由图 7 和表 2 可以看出分层平滑优化后的 A\* 算法与 A\* 算法相比,路径点数量减少 43,降低了 86%,路径转折点数量减少 9 个,降低了 64.29%,路径长度减少 1.45 cm,降低了 2.53%,非光滑转折点全部改善。综上,改进后的 A\* 算法与传统 A\* 算法相比,在已知静态环境中既能够快速高效规划路径并且安全避开障碍物,同时所规划的路径长度短,转角数量少,平滑度高。



(a) 内层优化  
(a) Inner optimization



(b) 中层优化  
(b) Middle level optimization



(c) 外层优化  
(c) Outer layer optimization

图 7 分层平滑优化图

Fig. 7 Layered smooth optimization diagram

2 分层平滑优化 A\* 引导 DWA

2.1 DWA 路径规划算法

1) 移动机器人运动模型

移动机器人与两轮差速驱动/四轮驱动均存在类似的非全向约束,通过控制两侧轮胎的相对速度实现转向,通过线速度、角速度描述其运动。移动机器人的运动模

型如图 8 所示。由于两个相邻时刻内机器人运动距离很短,可以将相邻两点之间的运动轨迹看成直线。

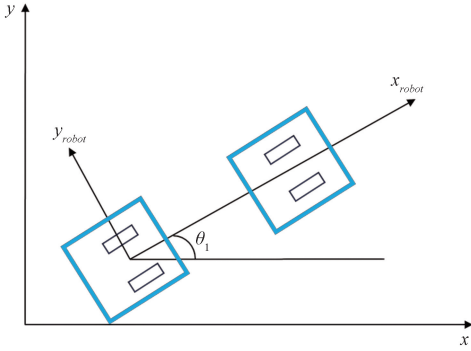


图 8 移动机器人运动模型

Fig. 8 Mobile robot motion model

时间段  $(t_2, t_1)$  内移动机器人位姿变换信息如式(28)~(30)所示。

$$\mu_{x_{i2}} = \mu_{x_{i1}} + v\Delta t \cos(\theta_{1\Delta t}) \quad (28)$$

$$\varepsilon_{x_{i2}} = \varepsilon_{x_{i1}} + v\Delta t \sin(\theta_{1\Delta t}) \quad (29)$$

$$\theta_{i2} = \theta_{i1} + w\Delta t \quad (30)$$

式中:  $\mu_{x_{i2}}$ ,  $\mu_{x_{i1}}$  和  $\varepsilon_{x_{i2}}$ ,  $\varepsilon_{x_{i1}}$  分别为  $t_2$ ,  $t_1$  处移动机器人的  $x$ ,  $y$  坐标值,  $\theta_{i2}$ ,  $\theta_{i1}$  分别为  $t_2$ ,  $t_1$  处移动机器人的航向角,  $v$  为移动机器人线速度,  $w$  为移动机器人角速度,  $\Delta t$  为  $t_1 \sim t_2$  时间段内的时间间隔。

## 2) 速度采样

在实际应用中,对速度空间  $[v, w]$  采样需要考虑移动机器人的性能和环境因素。由于机器人的速度限制, DWA 搜索求解的最大范围应该受到如下约束:

$$V_s = \{(v, w) \mid v_{\min} \leq v \leq v_{\max}, w_{\min} \leq w \leq w_{\max}\} \quad (31)$$

式中:  $v_{\min}$ ,  $v_{\max}$  分别为机器人的最小和最大线速度;  $w_{\min}$ ,  $w_{\max}$  分别为机器人最小和最大角速度,  $V_s$  代表线速度和角速度的集合。

在给定当前线速度  $v_c$  和角速度  $w_c$  条件下,下一时刻动态窗口  $V_d$  满足:

$$V_d = \{v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t], w \in [w_c - \dot{w}_b \Delta t, w_c + \dot{w}_a \Delta t]\} \quad (32)$$

式中:  $\dot{v}_b$ ,  $\dot{w}_b$  分别为移动机器人当前时刻的最大线减速度和最大角减速度;  $\dot{v}_a$ ,  $\dot{w}_a$  分别为移动机器人当前时刻的最大线加速度和最大角加速度。

移动机器人运动可行区域的线速度与角速度集合  $V_a$  满足:

$$V_a = \{v \leq \sqrt{2d(v, w) \dot{v}_b}, w \leq \sqrt{2d(v, w) \dot{w}_b}\} \quad (33)$$

式中:  $d(v, w)$  为路径上与障碍物最近的距离。

最终的速度范围为上述 3 个速度的集合,定义动态窗口  $V_r$  满足:

$$V_r = V_s \cap V_d \cap V_a \quad (34)$$

## 3) DWA 评价函数

DWA 的评价函数综合考虑了移动机器人与目标点之间的角度偏差、轨迹末端点与障碍物的距离、以及轨迹末端点到终点和全局路径的距离。对这些数值进行归一化处理,并分别赋予权重后求和,从而得到轨迹评估函数如式(35)所示。

$$E(v, w) = k(\alpha \cdot \text{heading}(v, w) + \beta \cdot \text{target}(v, w) + \gamma \cdot \text{path}(v, w) + \sigma \cdot \text{od}(v, w)) \quad (35)$$

式中:  $\text{heading}(v, w)$  为方位角评价函数,使方位角不断朝向终点位置;  $\text{target}(v, w)$  为目标距离评价函数,以不断缩短与终点距离;  $\text{path}(v, w)$  为局部轨迹评价函数,计算轨迹末端到全局路径的距离;  $\text{od}(v, w)$  为障碍物距离评价函数,评价机器人轨迹到障碍物距离;  $k(\cdot)$  为归一化函数,  $\alpha, \beta, \gamma, \sigma$  为各项权重。

## 2.2 局部避障路径引导优化生成机制

A\* 算法只能用于环境信息全部已知的情况,如果 A\* 算法规划的全局路径上出现了障碍物,那么移动机器人将无法避开它,因此引入如图 9 所示的局部避障路径引导优化生成机制,当移动机器人按照初始全局路径运动检测到未知障碍物时,利用 DWA 实现局部动态避障。

当移动机器人探测到未知障碍物时,为了引导 DWA 成功避开未知障碍物,将移动机器人当前时刻位置记为  $x_{j-1}$ ,移动机器人下一时刻位置记为  $x_j$ ,首先确定局部路径引导点  $x_{ga}$ ,具体公式如式(36)所示。

$$x_{ga} = 2x_o - x_j \quad (36)$$

式中:  $x_o$  为未知障碍物的位置。之后确定路径点  $x_{st}$ ,具体公式如式(37)所示。

$$x_{st} = \underset{x_{route} \in \{x_j, \dots, x_{target}\}}{\text{argmin}} \|x_{ga} - x_{route}\| \quad (37)$$

最后确定局部路径引导点  $x_{it}$  的具体公式如式(38)所示。

$$x_{it} = \underset{x_{route} \in \{x_j, \dots, x_{target}\}}{\text{argmax}} \frac{e_{gi}^T (x_{route} - x_{ga})}{\|x_{route} - x_{ga}\|} \quad (38)$$

其中,

$$e_{gi} = \frac{e_{gt} + e_{gs}}{\|e_{gt} + e_{gs}\|} \quad (39)$$

$$e_{gs} = \frac{x_{st} - x_{ga}}{\|x_{st} - x_{ga}\|} \quad (40)$$

式中:  $e_{gi}$  为  $e_{gt}$  与  $e_{gs}$  合向量的单位向量,  $e_{gs}$  为引导点  $x_{ga}$  到  $x_{st}$  方向的单位向量,  $e_{gt}$  为引导点  $x_{ga}$  处移动机器人速度的方向向量。

## 2.3 HSA\*-G-DWA 算法步骤及流程

算法 2(HSA\*-G-DWA 算法)步骤如下:

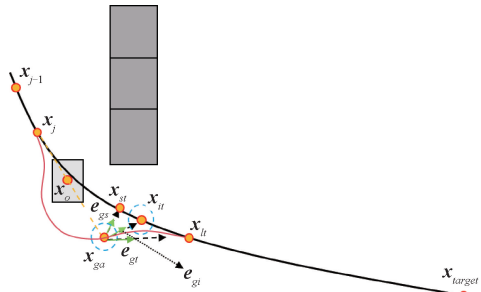


图 9 局部避障路径生成

Fig. 9 Local obstacle avoidance path generation

步骤 1) 如果移动机器人在按照算法 1 规划的初始全局路径行驶时没有探测到未知障碍物,那么它将按照算法 1 规划的全局路径行驶。

步骤 2) 当移动机器人探测到未知障碍物时,移动机器人当前位置记为  $x_{j-1}$ ,下一时刻的位置记为  $x_j$ 。

步骤 3) 利用式 (36) 确定局部路径引导点  $x_{ga}$ 。

步骤 4) 利用式 (37) 将全局路径中距离引导点  $x_{ga}$  最近的位置记为  $x_{st}$ 。

步骤 5) 利用式 (38) 来确定局部路径引导点  $x_{it}$ ,并将全局路径中  $x_{it}$  后 3d 位置处记为局部目标点  $x_{lt}$ 。

步骤 6) 设置图 9 蓝色虚线圆圈所示的中间引导点生效区域,圆圈范围为有效区域,设置圆圈半径为  $d$ ,当移动机器人进入蓝色虚线圆圈范围内时,中间引导点由  $x_{ga}$  转变为  $x_{it}$ 。

步骤 7) 获取移动机器人在路径点  $x_j$  处的线速度  $v$ ,角速度  $w$ ,航向角  $z$ ,利用 DWA 进行局部路径规划,将机器人的所有状态参数更新到状态参数集中,记录路径的所有节点和位姿信息,其中  $z$  的计算公式如式 (41) 所示。

$$z = \arctan\left(\frac{\varepsilon_{x_{ga}} - \varepsilon_{x_j}}{\mu_{x_{ga}} - \mu_{x_j}}\right)$$

(41)

式中:  $z$  为弧度值,  $\varepsilon_{x_{ga}}$ 、 $\varepsilon_{x_j}$ 、 $\mu_{x_{ga}}$ 、 $\mu_{x_j}$ , 分别为路径点  $x_j$  与局部引导点  $x_{ga}$  对应的  $x$  轴、 $y$  轴坐标值。

步骤 8) 通过局部引导点  $x_{ga}$ 、 $x_{it}$ ,局部终点  $x_{lt}$  依次替代局部目标点,引导 DWA 生成局部避障路径。

步骤 9) 若之后路径再次出现未知障碍物或动态障碍物,则重复步骤 2)~8), 否则返回步骤 1)。

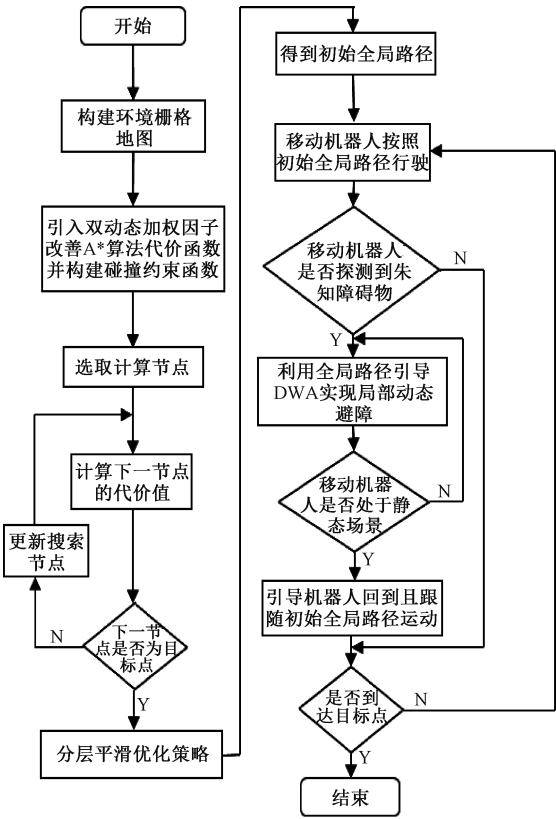


图 10 算法 2 流程图

Fig. 10 Algorithm 2 flowchart

3 仿真实验与分析

为验证 HSA\*-G-DWA 算法的正确性与合理性,在图 11~13 中将 HSA\*-G-DWA 算法和传统 A\* 算法、DWA 算法进行仿真对比实验分析,所得实验数据如表 3~5。在图 14 中将 HSA\*-G-DWA 算法与现有融合算法、RRT 算法、Dijkstra 算法以及单动态加权 A\* 算法进行仿真对比实验分析,所得实验数据如表 6 所示。其中图 11(a)~(f) 为已知静态环境,图 12(a)~(i)、图 13(a)~(f) 和图 14(a)~(c) 为随机加入未知静态和动态障碍物的未知环境。

表 3 静态场景仿真实验对比数据

Table 3 Comparison data of static scene simulation experiments

| 测试场景         | 算法         | 是否能够<br>到达终点 | 转折点<br>数/个 | 非光滑转折<br>点数/个 | 总转折角度/<br>(°) | 路径长度/<br>cm | 与障碍物的最小<br>距离/cm |
|--------------|------------|--------------|------------|---------------|---------------|-------------|------------------|
| 起点 1<br>静态场景 | 传统 A* 算法   | 是            | 12         | 12            | 585           | 55.94       | 0                |
|              | DWA 算法     | 否            | —          | —             | —             | —           | —                |
| 起点 2<br>静态场景 | HSA*-G-DWA | 是            | 5          | 0             | 323.56        | 55.83       | 0.83             |
|              | 传统 A* 算法   | 是            | 7          | 7             | 315           | 43.53       | 0                |
|              | DWA 算法     | 是            | 5          | 0             | 367.54        | 55.29       | 0.12             |
|              | HSA*-G-DWA | 是            | 1          | 0             | 91.52         | 41.81       | 0.85             |

注:“—”表示非必要



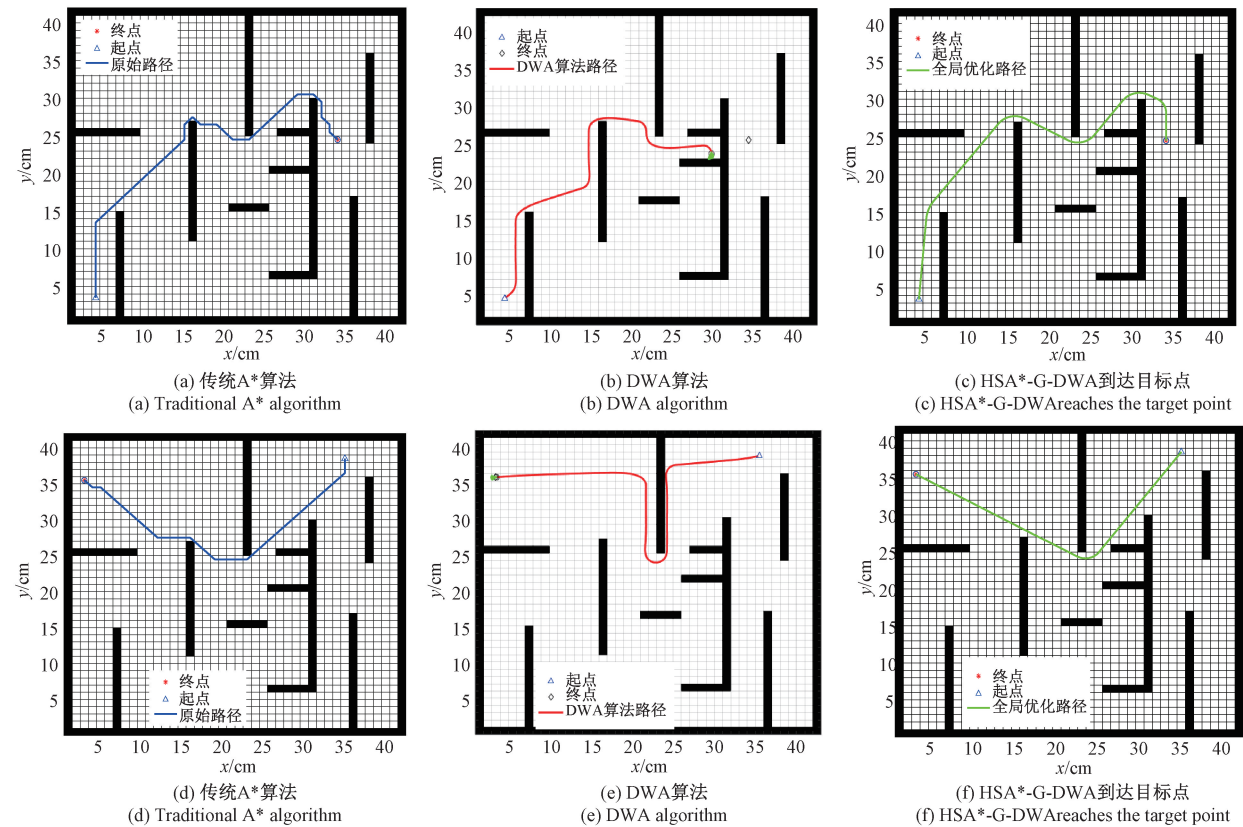


图 11 静态场景对比仿真结果

Fig. 11 Static scene comparison simulation results

表 4 未知场景仿真实验对比数据

Table 4 Comparison data of unknown scene simulation experiments

| 测试场景         | 算法         | 是否能够到达终点 | 转折点数/个 | 非光滑转折<br>点数/个 | 总转折角度/<br>( $^{\circ}$ ) | 路径长度/cm | 与障碍物的最<br>小距离/cm |
|--------------|------------|----------|--------|---------------|--------------------------|---------|------------------|
| 起点 1<br>未知场景 | 传统 A* 算法   | 是        | 12     | 12            | 585                      | 55.94   | 0                |
|              | DWA 算法     | 否        | —      | —             | —                        | —       | —                |
|              | HSA*-G-DWA | 是        | 10     | 0             | 420.57                   | 57.53   | 0.74             |
| 起点 2<br>未知场景 | 传统 A* 算法   | 是        | 7      | 7             | 315                      | 43.53   | 0                |
|              | DWA 算法     | 是        | 7      | 0             | 457.55                   | 58.29   | 0.12             |
|              | HSA*-G-DWA | 是        | 6      | 0             | 231.55                   | 43.26   | 0.77             |

注:“—”表示非必要

表 5 未预留安全距离仿真实验对比数据

Table 5 Comparative simulation data on the minimum safe distance without reservation

| 测试场景 | 算法         | 是否能够到达终点 | 转折点数/个 | 非光滑转折<br>点数/个 | 总转折角度/<br>( $^{\circ}$ ) | 路径长度/cm | 与障碍物的最<br>小距离/cm |
|------|------------|----------|--------|---------------|--------------------------|---------|------------------|
| 静态场景 | 传统 A* 算法   | 是        | 16     | 16            | 810                      | 88.25   | 0                |
|      | DWA 算法     | 否        | —      | —             | —                        | —       | —                |
|      | HSA*-G-DWA | 是        | 10     | 0             | 505.57                   | 88.16   | 0.5              |
| 未知场景 | 传统 A* 算法   | 是        | 16     | 16            | 810                      | 88.25   | 0                |
|      | DWA 算法     | 否        | —      | —             | —                        | —       | —                |
|      | HSA*-G-DWA | 是        | 16     | 0             | 731.96                   | 90.14   | 0.5              |

注:“—”表示非必要

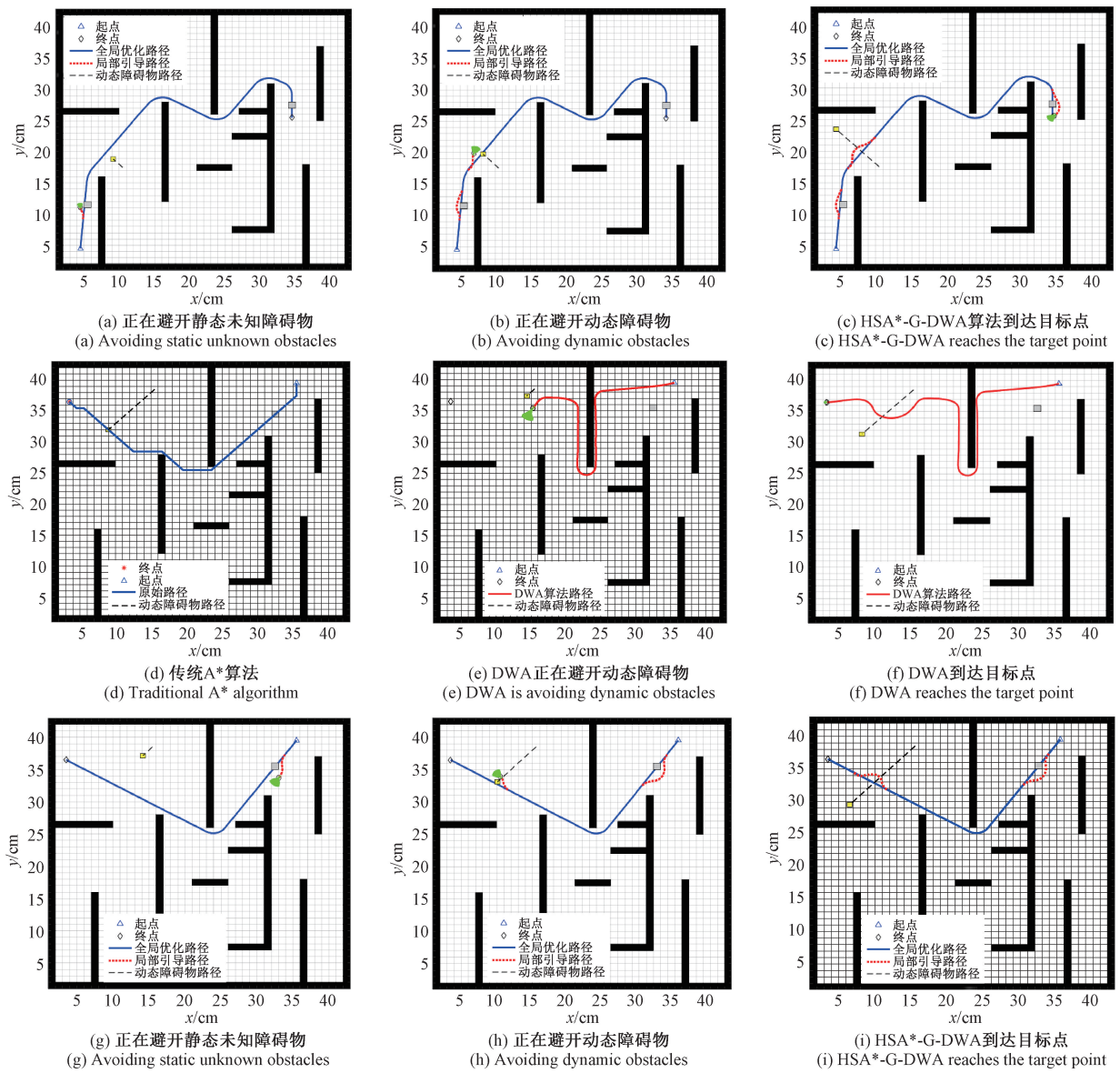


图 12 未知场景对比仿真结果

Fig. 12 Comparison of simulation results for unknown scenarios

表 6 不同算法实验对比数据

Table 6 Comparison data of experimental results for different algorithms

| 算法             | 路径长度/cm | 搜索时间/s | 与障碍物最小距离/cm |
|----------------|---------|--------|-------------|
| 现有 Dijkstra 算法 | 69.00   | 57.12  | 0           |
| RRT 算法         | 80.74   | 30.56  | 0           |
| 单动态加权 A* 算法    | 59.04   | 15.49  | 0           |
| 现有融合算法         | 65.72   | 67.59  | 0.61        |
| HSTA* -G-DWA   | 56.13   | 13.76  | 0.72        |

由于在图 12 起点 1 未知场景以及图 13 中 DWA 算法因陷入局部最优而无法完成路径规划;传统 A\* 算法不能实时避障来完成未知场景的路径规划,其未知场景仿真结果路径同图 11(a)、13(a) 所示,因此以上仿真不再

给出。  
由图 11 以及表 3 结果可知,在起点 1 静态场景中 HSA\* -G-DWA 算法相比于传统 A\* 算法路径转折点数量减少了 7 个,降低了 58.33%,总转折角度减少了

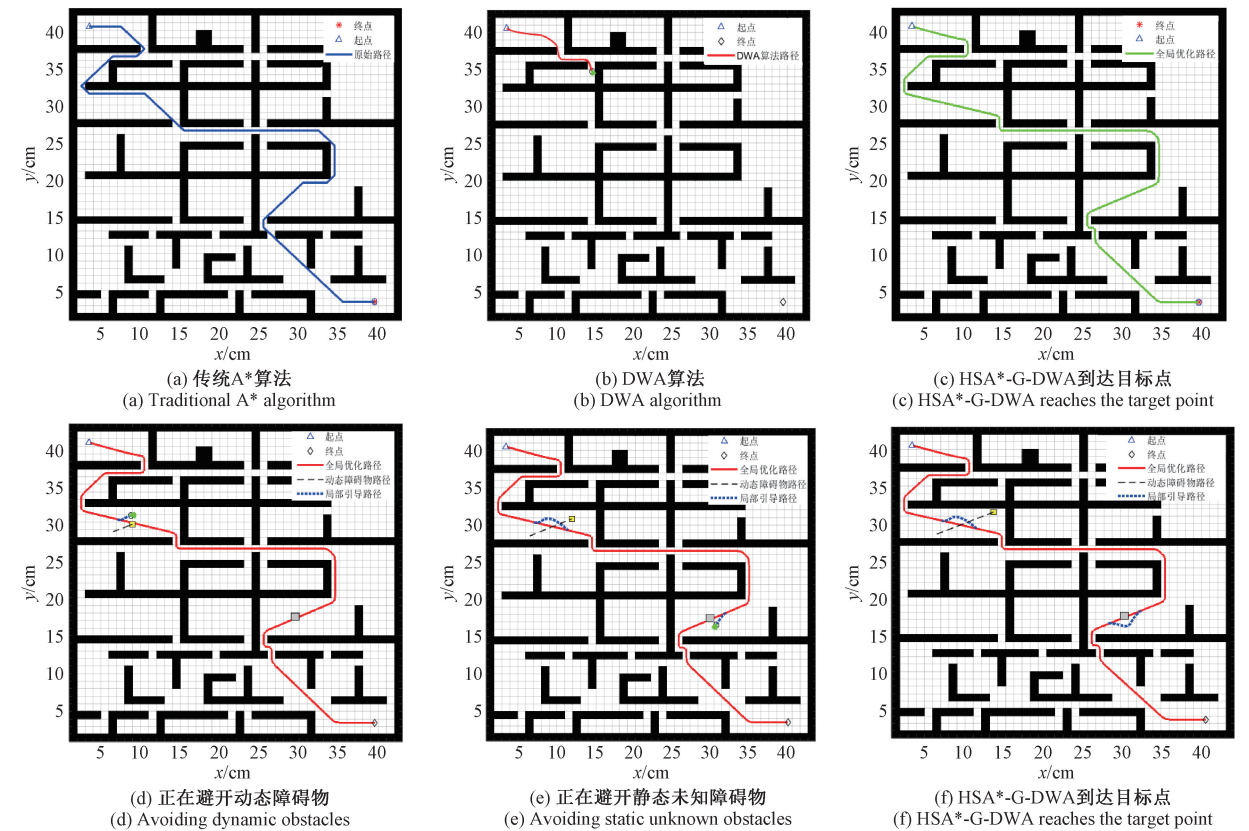


图 13 未预留安全距离的对比仿真结果

Fig. 13 Simulation results of contrasting without incorporating safety distances

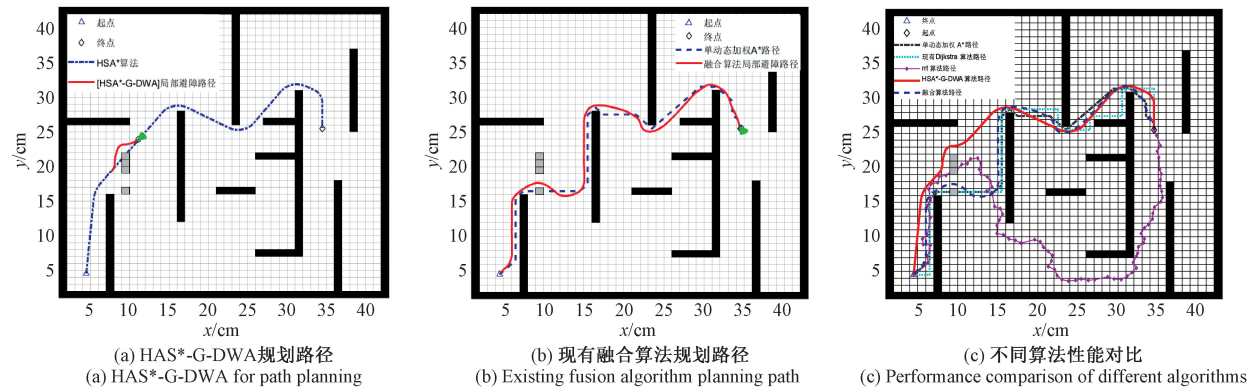


图 14 不同算法实验对比仿真结果

Fig. 14 Comparative simulation results of different algorithms

261.44°,降低了44.69%,路径平滑性和安全性提升,路径长度略有降低。与DWA算法相比,解决了DWA算法易陷入陷阱从而目标不可达问题。在起点2静态场景中3种算法都能到达目标点,但是HSA\*-G-DWA算法相比于A\*算法路径转折点数量减少6个,下降了85.71%,总转折角度减少了223.48°,降低了70.95%,路径长度减少了1.72cm,降低了3.95%,路径平滑性和安全性提升。与DWA算法相比,转折点数量减少了4个,降低了

80.0%,总转折角度减少了276.02°,降低了75.1%,路径长度减少了13.48cm,降低了24.4%,冗余路段减少,路径安全性提升。由图12以及表4结果可知,在起点1未知场景中HSA\*-G-DWA算法相比于传统A\*算法路径转折点数量下降了2个,降低了16.67%,总转折角度减少了164.43°,降低了28.11%,路径长度略有增加,路径平滑性提升且能够成功避开未知障碍物。与DWA算法相比,解决了DWA算法存在大量冗余路段以及易陷入陷



阱从而目标不可达的问题。在起点2未知场景中3种算法都能到达目标点,但是HSA\*-G-DWA算法相比于传统A\*算法路径转折点数量下降了1个,降低了14.29%,总转折角度减少了83.45°,降低了26.49%,路径平滑性提升,路径长度略有降低,且能够成功避开未知障碍物。与DWA算法相比,两者都能避开未知障碍物,但是HSA\*-G-DWA算法总转折角度减少了226°,降低了49.39%,路径长度减少了15.03 cm,降低了25.78%;;路径安全性提升。由图13以及表5可知,在未预留安全距离的静态场景中HSA\*-G-DWA算法相比于传统A\*算法路径转折点数量减少了6个,降低了37.5%,总转折角度减少了304.43°,降低了37.58%,路径长度略有降低,路径安全性与平滑性提升。与DWA算法相比,解决了DWA算法易陷入陷阱从而目标不可达问题。在未预留安全距离的未知场景中HSA\*-G-DWA算法相比于传统A\*算法总转折角度减少了78.04°,降低了9.63%,路径长度略有增加,但路径平滑性提升且能够成功避开未知障碍物。与DWA算法相比,解决了DWA算法存在大量冗余路段以及易陷入陷阱从而目标不可达的问题。由图14以及表6可知,在存在未知障碍物的相同未知环境中HSA\*-G-DWA算法相比于现有Dijkstra算法、RRT算法、单动态加权A\*算和现有融合算法,路径长度分别降低了18.65%、30.48%、5.09%和14.59%,搜索时间分别降低了75.91%、54.97%、11.17%和67.39%,路径安全性提升。

## 4 结 论

针对传统A\*算法搜索效率与安全性低下、路径转角多、不能作为实时路径,DWA算法存在冗余路段、易陷入陷阱以及DWA融合全局路径规划算法计算量大导致效率低下等问题,给出了一种名为HSA\*-G-DWA的机器人路径规划算法,该算法将动态加权因子和碰撞约束函数的调整机制引入到A\*算法中,提升了路径搜索效率并提高了路径的安全性。此外,该算法还采用分层平滑优化策略,减少了路径冗余点和路径长度并提升了路径的平滑性。最后,通过建立局部避障路径引导优化生成机制,在移动机器人探测到未知障碍物时结合分层平滑的全局路径引导DWA生成避障与返回全局路径的局部动态修正路径,改进了现有融合算法存在冗余路段、实时寻路效率低和实时计算量大的缺点。因此,HSA\*-G-DWA是一种安全、高效、稳定、可行的实时路径规划算法。

## 参考文献

- [1] ZHENG L, YU W, LI G, et al. Particle swarm algorithm path planning method for mobile robots based on artificial potential fields[J]. *Sensors*, 2023, 23(13): 1-15.
- [2] 刘军,冯硕,任建华,等. 移动机器人路径动态规划有向D\*算法[J]. *浙江大学学报(工学版)*, 2020, 54(2): 291-300.  
LU J, FENG SH, REN J H, et al. Directed D\* algorithm for dynamic path planning of mobile robots[J]. *Journal of Zhejiang University (Engineering Science)*, 2023, 23(13): 1-15.
- [3] 金震,黄卫华,李传奇,等. 基于改进JPS和A\*算法的组合路径规划[J]. *高技术通讯*, 2022, 32(4): 412-420.  
JIN ZH, HUANG W H, LI CH Q, et al. Combined path planning based on improved JPS and A\* algorithm[J]. *High Technology Letter*, 2022, 32(4): 412-420.
- [4] 张浩杰,张玉东,梁荣敏,等. 改进A\*算法的机器人能耗最优路径规划方法[J]. *系统工程与电子技术*, 2022, 45(2): 1-9.  
ZHANG H J, ZHANG Y D, LIANG R M, et al. Energy efficient path planning method for robots based on improved A\* algorithm[J]. *Systems Engineering and Electronics*, 2022, 45(2): 1-9.
- [5] XU H M, YU G H, WANG Y, et al. Path planning of Mecanum wheel chassis based on improved A\* algorithm[J]. *Electronics*, 2023, 12(8): 1-15.
- [6] 徐兴,俞旭阳,赵芸,等. 基于改进遗传算法的移动机器人全局路径规划[J]. *计算机集成制造系统*, 2022, (6): 1660-1672.  
XU X, YU X Y, ZHAO Y, et al. Global path planning of mobile robot based on improved genetic algorithm[J]. *Computer Integrated Manufacturing Systems*, 2022, (6): 1660-1672.
- [7] 王雷,王艺璇,李东东,等. 基于改进遗传算法的移动机器人路径规划研究[J]. *华中科技大学学报(自然科学版)*, 2024, 52(5): 158-164.  
WANG L, WANG Y X, LI D D, et al. Research on path planning of mobile robot based on improved genetic algorithm[J]. *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, 2024, 52(5): 158-164.
- [8] 肖金壮,余雪乐,周刚,等. 一种面向室内AGV路径规划的改进蚁群算法[J]. *仪器仪表学报*, 2022, 43(3): 277-285.  
XIAO J ZH, YU X L, ZHOU G, et al. An improved ant colony algorithm for indoor AGV path planning[J]. *Chinese Journal of Scientific Instrument*, 2022, 43(3): 277-285.
- [9] 杨立炜,付丽霞,王倩,等. 多层优化蚁群算法的移动机器人路径规划研究[J]. *电子测量与仪器学报*, 2021, 35(9): 10-18.



- YANG L W, FU L X, WANG Q, et al. Multi-layer optimal ant colony algorithm for mobile robots path planning study [J]. Journal of Electronic Measurement and Instrumentation 2021, 35(9): 10-18.
- [10] 黄翼虎, 于亚楠. 基于改进 Dijkstra 算法的防冲突最短路径规划研究[J]. 计算机与现代化, 2022(8): 20-24.  
HUANG Y H, YU Y N. Anti-collision shortest path planning based on improved Dijkstra algorithm [J]. Computer and Modernization, 2022(8): 20-24.
- [11] 左松涛, 毛占利, 范传刚, 等. 基于地铁站场景的改进型 Dijkstra 算法疏散路径规划研究[J]. 铁道科学与工程学报, 2022, 20(5): 1-12.  
ZUO S T, MAO ZH L, FAN CH G, et al. Evacuation path planning based on improved Dijkstra algorithm in metro station scene [J]. Journal of Railway Science and Engineering, 2022, 20(5): 1-12.
- [12] 陈都, 侯明, 张学东, 等. 改进 RRT 结合 B 样条的移动机器人路径规划研究[J]. 电子测量技术, 2022, 45(23): 38-44.  
CHEN D, HOU M, ZHNAG X D, et al. Research on path planning of mobile robot based on improved RRT and B spline [J]. Electronic Measurement Technology, 2022, 45(23): 38-44.
- [13] 骆海涛, 孙嘉泽, 高鹏宇, 等. 基于改进 RRT\* 算法的智能轮椅全局路径规划研究[J]. 仪器仪表学报, 2023, 44(10): 303-313.  
LUO H T, SUN J Z, GAO P Y, et al. Intelligent wheelchair global path planning research based on the improved RRT\* algorithm [J]. Chinese Journal of Scientific Instrument, 2023, 44(10): 303-313.
- [14] 黄健萌, 吴宇雄, 林昭谢. 移动机器人平滑 JPS 路径规划与轨迹优化方法[J]. 农业机械学报, 2021, 52(2): 21-29.  
HUANG J M, WU Y X, LIN X ZH. Smooth JPS path planning and trajectory optimization method of mobile robot [J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(2): 21-29.
- [15] 黄智榜, 胡立坤, 张宇, 等. 基于改进跳点搜索策略的安全路径研究[J]. 计算机工程与应用, 2021, 57(1): 56-61.  
HUANG ZH B, HU L K, ZHANG Y, et al. Research on security path based on improved skip search strategy [J]. Computer Engineering and Applications, 2021, 57(1): 56-61.
- [16] 陈超勇, 熊禾根, 陶永, 等. 基于高效模板法与动态窗口法的服务机器人全覆盖路径规划方法[J]. 高技术通讯, 2020, 30(9): 949-958.  
CHEN CH Y, XIONG H G, TAO Y, et al. Coverage path planning method for service robot based on efficient template algorithm and dynamic window approach [J]. Chinese High Technology Letters, 2020, 30(9): 949-958.
- [17] 彭育强, 黄泽龙, 李少伟, 等. 基于动态窗口法的移动机器人自动避障导航研究[J]. 自动化仪表, 2020, 41(10): 26-33.  
PENG Y Q, HUANG Z L, LI SH W, et al. Research on automatic obstacle avoidance navigation of mobile robot based on dynamic window approach [J]. Process Automation Instrumentation, 2020, 41(10): 26-33.
- [18] 李薪颖, 单梁, 常路, 等. 复杂环境下基于多目标粒子群的 DWA 路径规划算法[J]. 国防科技大学学报, 2022, 44(4): 52-59.  
LIN X Y, SHAN L, CHANG L, et al. DWA path planning algorithm based on multi-objective particle swarm optimization in complex environment [J]. Journal of National University of Defense Technology, 2022, 44(4): 52-59.
- [19] 郑维, 王昊, 王洪斌. 动态环境下基于自适应步长 Informed-RRT\* 和人工势场法的机器人混合路径规划[J]. 计量学报, 2023, 44(1): 26-34.  
ZHENG W, WANG H, WANG H B. Adaptive step size informed-RRT\* and artificial potential field algorithm for hybrid path planning of robot [J]. Journal of Metrology, 2023, 44(1): 26-34.
- [20] 袁朝春, 翁杰丰, 何友国, 等. 基于改进人工势场法的路径规划决策一体化算法研究[J]. 农业机械学报, 2019, 50(9): 394-403.  
YUAN CH CH, WENG SH F, HE Y G, et al. Integration algorithm of path planning and decision-making based on improved artificial potential field [J]. Transactions of the Chinese Society for Agricultural Machinery, 2019, 50(9): 394-403.
- [21] ZHANG C, AO L, YANG J, et al. An improved A\* algorithm applying to path planning of games [J]. Journal of Physics: Conference Series, 2020.
- [22] 余文凯, 章政, 付雪画, 等. 基于地图预处理及改进 A\* 算法的路径规划[J]. 高技术通讯, 2020, 30(4): 383-390.  
YU W K, ZHANG ZH, FU X H, et al. Path planning based on map partition preprocessing and improved A\* algorithm [J]. Chinese High Technology Letters, 2020, 30(4): 383-390.
- [23] 李晓露, 熊禾根, 陶永, 等. 基于改进 A\* 算法的移动机器人全局最优路径规划[J]. 高技术通讯, 2021, 31(3): 306-314.  
LI X L, XIONG H G, TAO Y, et al. Global optimal

- path planning for mobile robots based on improved A\* algorithm[J]. Chinese High Technology Letters, 2021, 31(3):306-314.
- [24] 姜媛媛,张阳阳. 改进 8 邻域节点搜索策略 A\* 算法的路径规划[J]. 电子测量与仪器学报, 2022, 36(5): 234-241.
- JIANG Y Y, ZHANG Y Y. Improved path planning of A\* algorithm of domain node search strategy 8 [J]. Journal of Electronic Measurement and Instrumentation, 2022, 36(5):234-241.
- [25] 汪馨,姚念民,谭国真. 改进 BIT\* 与 DWA 算法的动态路径规划[J]. 计算机工程与应用, 2022, 58(20): 247-254.
- WANG X, YAO N M, TAN G ZH. Dynamic path planning based on improved BIT\* and DWA algorithm [J]. Computer Engineering and Applications, 2022, 58(20): 247-254.
- [26] 袁千贺,魏国亮,田昕,等. 改进 A\* 和 DWA 融合的移动机器人导航算法[J]. 小型微型计算机系统, 2023, 44(2):334-339.
- YUAN Q H, WEI G L, TIAN X, et al. Improved navigation algorithm of mobile robot based on the fusion of

A\* and DWA[J]. Journal of Chinese Computer Systems, 2023, 44(2):334-339.

## 作者简介



朱洪波, 2017 年获得中国科学技术大学博士学位, 现为安徽理工大学副教授, 主要研究方向为移动机器人定位、导航与控制等。

E-mail: hbzhu@aust.edu.cn

**Zhu Hongbo** received his Ph. D. degree in control science and engineering from the University of Science and Technology of China in 2017. Now he is an associate professor at Anhui University of Science and Technology. His main research interests include mobile robot positioning, navigation and control, etc.



殷宏亮(通信作者), 2021 年获得合肥师范学院学士学位, 现为安徽理工大学硕士研究生, 主要研究方向为移动机器人路径规划。

E-mail: 1253929967@qq.com

**Yin Hongliang** (Corresponding author) received his B. Sc. degree from Hefei Normal University in 2021. Now he is a M. Sc. candidate at Anhui University of Science and Technology. His main research interest includes path planning.