

doi: 10.7690/bgzd.2022.12.010

安全性 A*算法融合动态窗口法的路径规划

郭翰卿¹, 付丽霞¹, 张勇¹, 毛剑琳¹, 李勃²

(1. 昆明理工大学信息工程与自动化学院, 昆明 650504; 2. 郑州大学信息工程学院, 郑州 450000)

摘要: 针对移动机器人在路径寻优过程中, 传统 A*算法搜索效率差、所规划路径缺乏安全性、拐点多、转角大且无法实现动态避障等问题, 提出一种安全性 A*算法和动态窗口法(dynamic window approach, DWA)结合的融合算法。全局路径规划中, 在传统 A*算法的评价函数中引入安全估值, 并拓展启发式搜索邻域和精简搜索方向; 进行二次路径优化, 删除冗余节点, 并平滑路径; 运用改进的动态窗口评价函数, 将安全性 A*算法与动态窗口法融合实现机器人沿全局路径行进中的动态避障。仿真实验结果表明: 改进 A*算法相比文献算法在路径长度上和拐角数量上平均减少了 2.39%和 25%, 并在动态复杂环境下验证了其动态避障效果, 能满足机器人路径规划的实际需求, 具有一定的应用价值。

关键词: A*算法; 全局路径; 动态窗口法; 动态避障

中图分类号: TP391.9 **文献标志码:** A

Path Planning Based on Security A* Algorithm and Dynamic Window Method

Guo Hanqing¹, Fu Lixia¹, Zhang Yong¹, Mao Jianlin¹, Li Bo²

(1. School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650504, China; 2. School of Information Engineering, Zhengzhou University, Zhengzhou 450000, China)

Abstract: In the process of mobile robot path optimization, the traditional A* algorithm has some problems, such as poor search efficiency, lack of safety, many turning points, large turning angles and unable to achieve dynamic obstacle avoidance, etc. To solve these problems, this paper proposes a fusion algorithm which combines the safety A* algorithm with the dynamic window approach (DWA). In the global path planning, the safety estimation is introduced into the evaluation function of the traditional A* algorithm, and the heuristic search neighborhood is expanded and the search direction is simplified. Then, the secondary path optimization is carried out to delete redundant nodes and smooth the path. By using the improved dynamic window evaluation function, the safety A* algorithm and the dynamic window method are integrated to realize the dynamic obstacle avoidance of the robot along the global path. The simulation results show that the improved A* algorithm reduces the path length and the number of corners by 2.39% and 25% on average compared with the literature algorithm, and verifies its dynamic obstacle avoidance effect in the dynamic complex environment, which can meet the actual needs of robot path planning and has certain application value.

Keywords: A* algorithm; global path; dynamic window method; dynamic obstacle avoidance

0 引言

当前, 自主机器人技术运用于各行各业, 其中移动机器人可以实现自主躲避障碍物和规划路径, 在不同领域都有广泛应用。路径规划问题可分为静态全局路径规划^[1-2]和动态局部路径规划^[3-4]。全局路径规划是指在环境信息已知的情况下, 为机器人规划一条最优路径; 局部路径规划则是在环境信息完全未知或部分可知的情况下, 侧重机器人当前局部信息, 在局部为机器人规划路径。目前, 复杂地图环境^[5]和动态障碍物环境^[6]下的路径规划是最困难和最具挑战性的课题之一。全局路径规划所规划路线无法避开实际环境中不受控制的动态障碍物, 这时就需要在机器人行进中采用局部路径规划方

法; 因此, 研究一种兼顾全局静态和局部动态的路径规划算法尤为重要。

对于环境信息已知的静态全局路径规划, 常用的全局路径规划算法有 Dijkstra 算法^[7]、A*算法^[8]、RRT 算法^[9]、神经网络算法^[10]等。其中 A*算法得到广泛的研究, 针对 A*算法存在的搜索效率低下, 路径存在过多冗余节点, 且缺乏安全性考虑等问题, 许多学者进行了改进。文献[11]在传统 A*算法的基础上修改搜索策略提出跳点搜索算法, 通过裁剪一些无意义的搜索节点加快了搜索效率, 但该算法在地图较为复杂时搜索范围仍然过于庞大, 且路径转折点过多。文献[12]在 A*算法基础上融入可视图, 可有效减少冗余节点搜索, 增加搜索效率, 但该算

收稿日期: 2022-08-19; 修回日期: 2022-09-25

基金项目: 国家自然科学基金资助项目(61163051); 云南省重点研发计划资助项目(202002AC080001)

作者简介: 郭翰卿(1994—), 男, 河南人, 硕士, 从事移动机器人路径规划研究。E-mail: 791195951@qq.com。

法未对路径长度进行再次优化。文献[13]在 A*算法中进行二次计算，缩小了路径长度，并加入圆弧算法和加权步长分析有效增加算法效率，然而未考虑机器人行进过程中的安全问题。对于动态复杂环境下的路径规划问题，一直是研究难点。动态窗口法^[14]、人工势场法^[15]等算法在局部路径规划问题中被广泛应用。文献[16]对传统人工势场法的模型进行优化，引入新的变量优化参数，实现了机器人对动态环境的适应性，但容易陷入局部最优。文献[17]改进了动态窗口法，可对路径进行一定优化并且可以实现避障，但该算法只能进行局部路径规划。

笔者针对上述改进算法并未完全解决传统 A*算法难以应对复杂环境，且存在搜索范围大、效率低、安全性差、冗余节点和转折点多等问题，提出一种安全性 A*算法，在拓展目标点搜索方向的基础上精简搜索方向，提高搜索效率，并改进评价函数以提高路径安全性，然后对所规划路径进行二次优化，最后融合动态窗口法，实现机器人沿全局路径行进过程中动态避障，在避开动态障碍的同时保证了路径长度最优。

1 算法基础

1.1 传统 A*算法

A*算法是可以实现全局路径规划的启发式算法，主要应用于采用栅格法建模的地图中，其核心思想为从路径起点开始对周围 8 邻域搜索，计算当前节点周围每个节点到起点的实际代价值和到终点的估计代价值，选择总代价最小的节点作为下个选择节点，按照此方式搜索至目标节点，即可完成路径规划并找到最优路径。A*算法估值评价函数如下：

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

式中： k 为当前节点； $F(k)$ 为节点 k 的总代价； $G(k)$ 为机器人从节点 k 到达起始节点的实际代价； $H(k)$ 为机器人从节点 k 到达目标节点的估计代价，这是估值评价函数最重要的部分，选择合适的 $H(k)$ 可以提高算法的准确性、高效性。

常见的计算估计代价的方法有曼哈顿距离、切比雪夫距离、欧式距离，分别是：

$$H(k) = |x_2 - x_1| + |y_2 - y_1| \quad (2)$$

$$H(k) = \max(|x_2 - x_1|, |y_2 - y_1|) \quad (3)$$

$$H(k) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4)$$

式中 (x_1, y_1) 和 (x_2, y_2) 分别表示当前节点和目标节点

的坐标。

如图 1 所示，在栅格地图中，传统的 A*算法主要搜索方向为 8 个。这种方法需要在每个当前节点都对周围 8 个方向的栅格进行搜索，并且需要不断查询 Open List 和 Close List 中的节点，消耗大量时间在搜索节点上，其中存在大量冗余节点被搜索，造成时间上的浪费，且所规划路径转折点多、转角过大，不符合机器人的运动学原理，影响机器人行驶速度和安全性。另外，由于将机器人当作质点，传统 A*算法所规划路径会出现与障碍物边缘碰撞的情况，得到一条机器人实际不可行路径。针对以上问题，笔者提出一种安全性 A*算法。

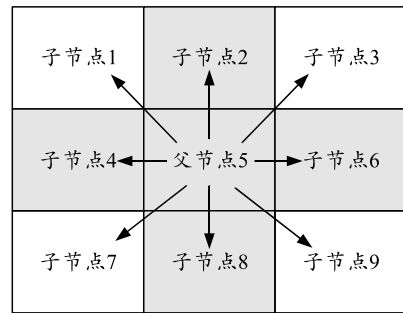


图 1 传统搜索方向

1.2 安全性 A*算法

1.2.1 优化估价函数

传统 A*算法规划的路径没有考虑到安全性。为了保证机器人规划路径的安全性，避免其行进中与障碍物边缘碰撞，笔者预先为机器人设定一个安全半径，并根据待选节点和障碍物的最小距离在 A*算法的实际代价函数中引入安全估值计算，当待选节点与障碍物的最短距离超过安全半径时，认为该待选节点的安全估值为零。改进后 A*算法的实际代价函数 $G(k)$ 为：

$$G(k) = \sum_{i=1}^k \omega(l_i + s_i) \quad (5)$$

其中 s_i 为：

$$s_i = \begin{cases} k(1 - d_i / r) & 0 < d_i \leq r \\ 0 & d_i > r \end{cases} \quad (6)$$

式中： l_i 为机器人在路段 i 的实际代价（即路段 i 的长度）； s_i 为路段 i 末端节点的安全估值； ω 为安全估值系数； d_i 为路段 i 末端节点与障碍物的最小距离； r 为安全半径； k 为安全估值系数；且都大于零；当 $d_i > r$ 时，安全估值为零。

笔者的估计代价函数 $H(k)$ 采用欧式距离计算，改进后的估值评价函数为：

$$F(k) = \sum_{i=1}^k \omega(l_i + s_i) + H(k) \quad (7)$$

1.2.2 拓展启发式搜索邻域

传统 A*算法是在 8 邻域方向搜索节点, 搜索方向和角度有所限制, 所以在栅格地图中规划出的机器人路径并非最优, 且存在较多的转折点和较大的转角。为解决传统 A*算法在邻域搜索方向上的限制以获得更优路径, 拓展一个启发式搜索领域, 在传统 8 邻域方向以外增加当前节点到目标节点方向的搜索方向, 在该搜索方向步长为 1 的位置拓展一个待选节点; 如果该节点与目标节点的距离小于 1 则将其当作目标点, 该拓展搜索节点的方向角度公式如下:

$$\text{angle} - \exp = \arcsin(\text{del_}Y_i / r_i) \quad (8)$$

式中: $\text{del_}Y_i$ 为该拓展节点与目标节点 Y 轴的差值; r_i 为该拓展节点与目标节点的距离。在路径寻优中, 需要判断当前节点与拓展节点之间的连线是否与障碍物相交, 若不相交, 则将拓展节点加入开放列表。

拓展启发式搜索邻域后的搜索邻域如图 2 所示, A*算法的路径节点不再受限于栅格中心点, 并且搜索角度也不再局限为 $\pi/4$ 的整数倍。

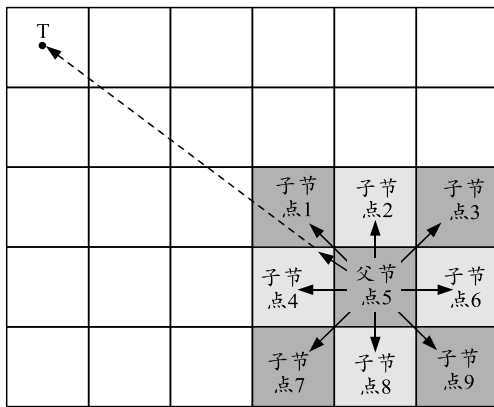


图 2 拓展启发式搜索邻域

1.2.3 精简搜索方向

由于传统 A*算法搜索方向为 8 个方向搜索, 但是当目标节点远离起始点时, 在实际情况中对于远离目标点的 3 个搜索方向是冗余搜索方向, 在计算过程中会造成一定程度上的时间浪费。为提高搜索效率, 删除远离目标点的搜索方向以精简搜索方向。

如图 3 所示, 当目标节点为 1 方向时, 删除 6、9 和 8 方向的搜索; 当目标节点为 2 方向时, 删除 7、8 和 9 方向的搜索; 当目标节点为 3 方向时, 删除 4、7 和 8 方向的搜索;

同理, 都删除远离目标节点的 3 个冗余方向, 从而进一步提高搜索效率。

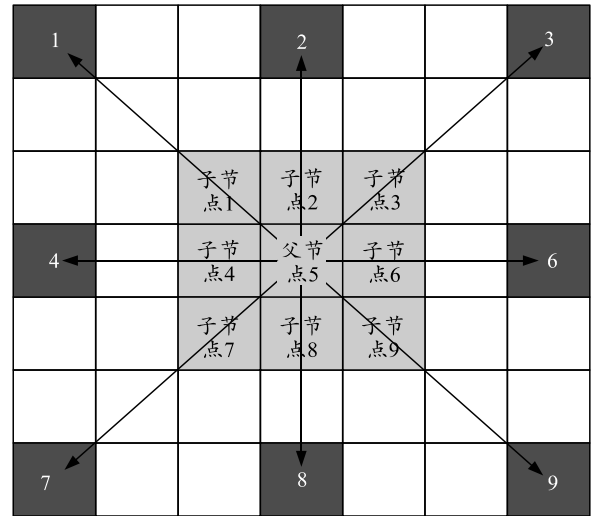


图 3 精简搜索方向

1.2.4 二次路径优化

针对 A*算法在栅格化环境中规划路径时存在冗余节点多和转折角度过大的问题, 笔者采用一种结合障碍物判断的路径关键节点提取方法, 双向删除冗余节点, 并结合 3 次 B 样条插值法^[18]进行转角圆滑处理, 有效优化转折度和减少路径长度。二次路径优化步骤如下:

步骤 1: 首先读取 A*算法规划的路径, 把所有路径节点放入集合 $M\{G_i, 1 \leq i \leq n\}$, 其中: G_1 表示规划路径的起点, G_n 表示规划路径的终点。然后创建一个只含初始值 G_1 和 G_n 的关键节点集合 $P\{G_1, G_n\}$, 用于存放路径优化后的关键节点。

步骤 2: 删掉路径中同一线段上的中间点, 只保留路径的起点、转折点和终点。

步骤 3: 从点 G_1 开始, 在转折点 G_i, G_j 之隔 d 步长取点 G_{i+d} , 并判断所取两节点的连线是否与障碍物相交。若没有与障碍物相交, 则选择当前节点 G_{i+d} 替代集合 P 中的 G_i 为路径节点; 否则将不选择。

步骤 4: 从路径的终点 G_n 出发, 重新取点判断 (采用步骤 3 的判断方法), 将所有取得的关键转折点放入集合 P 中。

步骤 5: 对路径转角进行 3 次 B 样条圆滑优化^[19], 增加路径平滑度并进一步缩短路径。

图 4 所示为二次路径优化过程。

2 动态窗口算法

动态窗口方法 (DWA) 基于机器人所载传感器实时检测局部环境信息, 结合机器人运动学和动力

学进行速度空间采样, 实现路径规划。该算法在速度空间采样多组机器人数据, 并根据机器人运动学模型预测机器人的运动轨迹, 然后通过评价函数选择最佳轨迹来完成局部路径规划; 但是传统的 DWA 算法进行规划时缺少对全局路径的判断, 仅在局部对环境信息进行分析, 只能单纯地指引机器人向着目标安全行进, 所规划路径并不是全局最优, 且在障碍物过多时容易陷入局部最优。

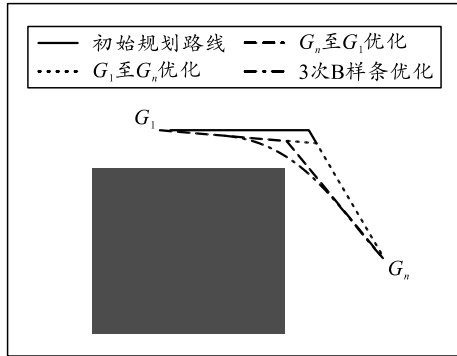


图 4 二次路径优化

2.1 机器人运动学模型

动态窗口法通过运动学方法将机器人的速度信息转换成位置信息。为更好地模拟机器人的运动, 需要分析其运动模型。假设移动机器人的运动轨迹为弧线, 则一对速度 (v, w) 代表一条圆弧轨迹, 机器人的运动学模型如下:

$$x_t = x_{t-1} + v_x \Delta t \cos \theta_{t-1} - v_y \Delta t \sin \theta_{t-1}; \quad (9)$$

$$y_t = y_{t-1} + v_x \Delta t \sin \theta_{t-1} + v_y \Delta t \cos \theta_{t-1}; \quad (10)$$

$$\theta_t = \theta_{t-1} + w \Delta t. \quad (11)$$

式中: x_t, y_t, θ_t 为机器人在时间 t 的位姿信息, x_t, y_t 为机器人在时间 t 的位置坐标, θ_t 为机器人在时间 t 内的方向角; $x_{t-1}, y_{t-1}, \theta_{t-1}$ 为机器人在时间 $t-1$ 处的位姿信息; v, w 为机器人当前的线速度和角速度; Δt 为瞬时时间变化(即采样时间)。

2.2 机器人速度采样

机器人的运动速度受到多种因素制约, 其中主要包括机器人自身和环境的约束, 所以需要采样约束分析, 得到速度限制就可以预测轨迹, 然后计算这些轨迹的评价值, 选择最优评价值的轨迹到达目标点。在速度 (v, w) 的采样空间中, 存在多组速度集合, 所以要根据实际情况的限制将采样速度约束在一定范围内。

1) 机器人的自身最小、最大速度制约:

$$V_m = \{v \in [v_{\min}, v_{\max}], w \in [w_{\min}, w_{\max}]\}. \quad (12)$$

式中 V_m 表示在机器人自身条件所允许情况下的最大、最小线速度和角速度集合。

2) 电机性能对机器人的速度制约。

移动机器人由于其电动机转矩有限, 因此存在最大的加速和减速限制。机器人在 Δt 内, 电机加减速限制带来的速度约束为:

$$V_d = \left\{ \begin{aligned} &v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t] \wedge \\ &w \in [w_c - \dot{w}_b \Delta t, w_c + \dot{w}_a \Delta t] \end{aligned} \right\}. \quad (13)$$

式中: v_c 为当前线速度; w_c 为当前角速度; \dot{v}_a 为最大线加速度; \dot{v}_b 为最大线减速度; \dot{w}_a 为最大角加速度; \dot{w}_b 为最大角减速度。

3) 机器人制动安全约束。

为确保机器人安全, 当机器人在运动过程中遇到障碍物时, 需要在最大减速度的约束下, 在碰撞到障碍物前将其速度减为 0, 其制动安全约束为:

$$V_a = \left\{ (v, w) \mid v \leq \sqrt{2 \text{dist}(v, w) \cdot \dot{v}_b} \wedge \sqrt{2 \text{dist}(v, w) \cdot \dot{w}_b} \right\}. \quad (14)$$

式中 $\text{dist}(v, w)$ 为机器人轨迹和障碍物的最近距离。

综上, 机器人的实际速度空间是 3 个速度空间的交集。机器人的动态速度窗口如下:

$$V_r = V_m \cap V_d \cap V_a. \quad (15)$$

2.3 评价函数

机器人运动中因为速度采样不同会出现无数条轨迹, 需要设计评价函数以寻找最优轨迹。在局部路径规划中, 尽最大可能使机器人避开障碍物并以最快的速度移至目标点, 评价函数的表达式如下:

$$G(v, w) = \rho[\alpha \text{head}(v, w) + \beta \text{dist}(v, w) + \gamma \text{vel}(v, w)]. \quad (16)$$

式中: $\text{head}(v, w)$ 为方位角评价函数, 引导方位角始终朝向目标节点; $\text{vel}(v, w)$ 为速度评价函数; $\text{dist}(v, w)$ 为距离评价函数, 用于评价机器人当前轨迹与障碍物的最近距离。如果该参数为常数, 表示无障碍物。具体函数如下所示:

$$\text{head}(v, w) = 180^\circ - \theta; \quad (17)$$

$$\text{vel}(v, w) = V; \quad (18)$$

$$\text{dist}(v, w) = \begin{cases} d_{\max}, & d_{\max} \leq \text{disttmp} \\ \text{disttmp}, & d_{\max} > \text{disttmp} \end{cases}. \quad (19)$$

式中: θ 为方位角, 是机器人当前轨迹与目标点方向的夹角; V 为当前轨迹速度, $V \subset V_r$; d_{\max} 为最大距离的设定值, 其作用是当移动机器人周围没有障碍物时防止 $\text{dist}(v, w)$ 函数过大; disttmp 为当前预测轨迹的末端和障碍物的最近距离。

2.4 评价函数归一化处理

对评价函数的归一化处理，其目的是统一评价函数中不同单位的因素，将这些不同量纲的量转化为无量纲的量，使其具有可比性。方位角评价函数、速度评价函数、距离评价函数归一化处理方法如下：

$$\text{normal_head}(i) = \text{head}(i) / \sum_{i=1}^n \text{head}(i); \quad (20)$$

$$\text{normal_dist}(i) = \text{dist}(i) / \sum_{i=1}^n \text{dist}(i); \quad (21)$$

$$\text{normal_vel}(i) = \text{vel}(i) / \sum_{i=1}^n \text{vel}(i). \quad (22)$$

式中： n 为被采样的所有轨迹的数量， i 为待评价轨迹的序号。

3 融合算法

传统 DWA 是基于移动机器人检测到的局部窗口环境信息实时判断和规划路径，具有良好的避障能力。该方法根据机器人当前位姿和目标点之间的方位角来确定路径探索方向，缺乏全局路径最优的考虑。

笔者首先运用所改进的安全性 A*算法进行全局路径规划，然后结合动态窗口法进行未知环境下的局部路径规划，从而实现机器人沿全局规划路径行进时能避开随机出现的障碍物，获得具有动态自主避障特性的全局最优路径，既提高了 A*算法的安全性和适应性，又达到了动态避障的效果。为使机器人能沿全局规划最优路径行进，改进 DWA 评价函数如下：

$$G(v, w) = \rho[\alpha \times S \times \text{head}(v, w) + \beta \times \text{dist}(v, w) + \gamma \times \text{vel}(v, w)]. \quad (23)$$

式中： $S \cdot \text{head}(v, w)$ 为当前目标点方向和机器人当前轨迹方向的夹角，与传统 DWA 算法不同的是，当前目标节点是机器人沿当前轨迹方向上距离其最近的全局最优路径中的序列节点； α 、 β 和 γ 分别为

$S \cdot \text{head}(v, w)$ 、 $\text{dist}(v, w)$ 和 $\text{vel}(v, w)$ 的权重系数，权重越大，则该项评价函数的主导作用就越大； ρ 为归一化系数。

通过此评价函数可使得局部路径规划遵循全局规划最优路径的轨迹点，从而保证避开动态障碍的同时得到全局最优路径。融合算法流程如图 5 所示。

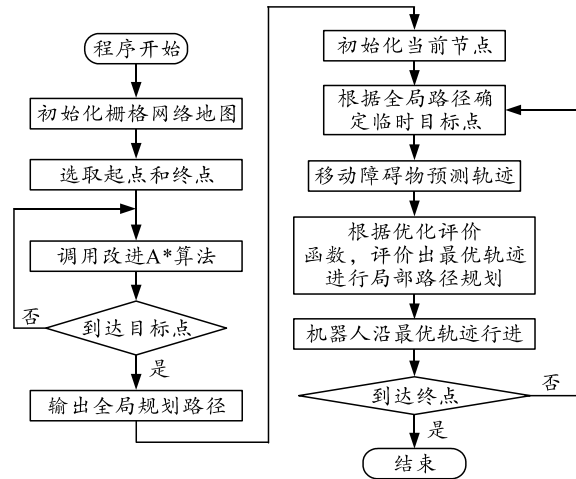


图 5 融合算法流程

4 仿真实验

4.1 安全性 A*算法仿真实验

为验证笔者提出方法的有效性，对算法进行仿真验证。仿真环境为 Matlab2018b 软件、8GB 的 64 位 WIN10 系统。

首先建立栅格地图模拟环境，然后在相同的地图环境中，分别用传统 A*算法、文献[19]算法和本文中安全性 A*算法进行路径规划实验。实验中，本文中安全性 A*算法参数安全半径 r 为 0.5，安全估值系数 k 为 0.25，实验一采用 20×20 的栅格地图环境，最终路径规划结果如图 6 所示；实验二在实验一基础上增大障碍物覆盖率模拟更复杂的环境，采用 30×30 的栅格地图环境，路径规划结果如图 7 所示。在不同环境中，本文中算法运行 10 次，取其 10 次结果的平均值，实验结果如表 1 所示。

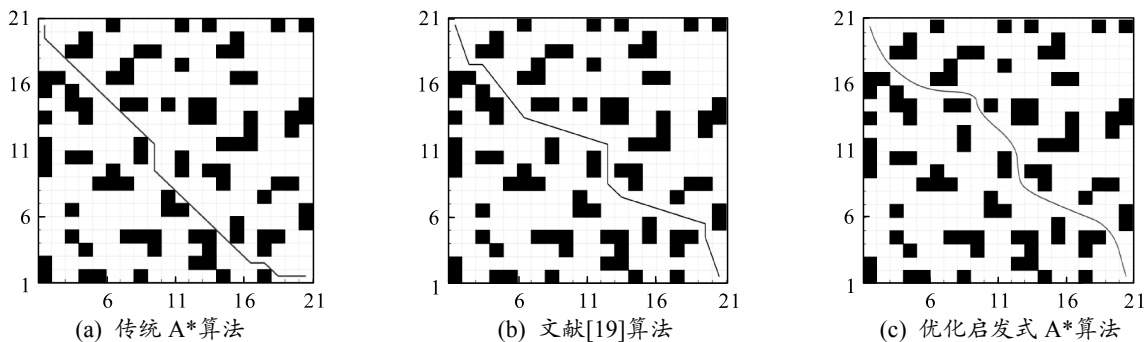


图 6 20×20 栅格环境中 3 种算法的路径规划结果

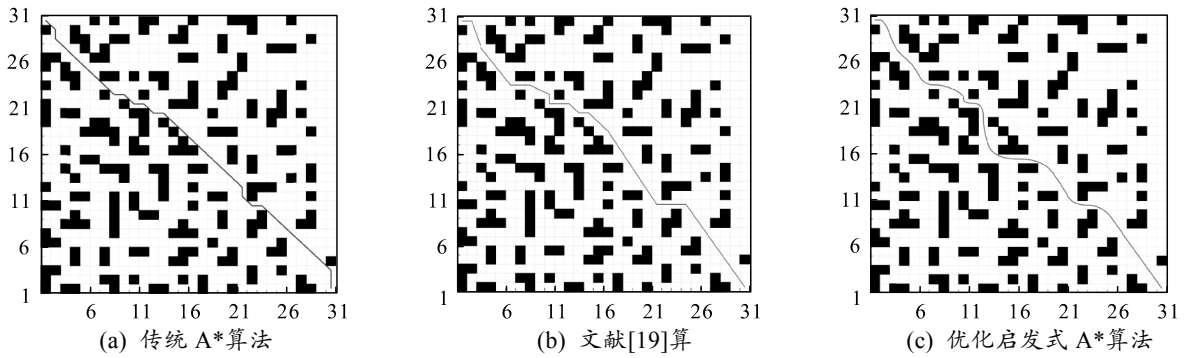


图 7 30×30 栅格环境中 3 种算法的路径规划结果

表 1 不同环境下 3 种算法的路径规划性能比较

栅格场景	路径规划算法	路径长度/m	拐点个数	运行时间/s
20×20	传统 A*算法	28.63	6	6.41
	文献[19]算法	30.39	8	2.54
	本文中安全性 A*算法	29.18	6	2.88
30×30	传统 A*算法	43.36	13	9.59
	文献[19]算法	44.89	12	4.26
	本文中安全性 A*算法	44.53	9	4.37

从实验结果可知，在不同环境下，传统 A*算法所规划路径最短，但其路径与障碍物边缘碰撞，所规划路径在机器人实际工作中不可行。本文中的安全性 A*算法在不同环境下所规划路径长度较文献[19]算法都有一定优势，在 20×20 和 30×30 栅格环境下，路径长度较文献[19]算法分别减少 3.98% 和 0.80%。在规划时间上，由于搜索方向由传统的 8 方向搜索精简为 6 方向搜索，本文中算法比传统 A*算法耗时间减少一倍以上，较文献[19]略有增加，原因是本文中算法更多的二次路径优化处理导致计算时间增加。但是在路径转折点数量上，在 20×20 栅格环境下，本文中安全性 A*算法和传统 A*算法持平，较文献[18]算法减少 25%；30×30 栅格环境下，本文中安全性 A*算法较其他 2 种算法分别减少 31%和 25%，且路径更平滑，更符合机器人的运行学特征，更利于机器人安全行进。综上所述，本文中安全性 A*算法具有更好的全局路径规划性能。

4.2 融合算法仿真实验

为验证笔者所提出融合算法的动态避障能力，在 40×40 的复杂栅格地图环境中添加动态障碍物，并分别运用本文中安全性 A*算法和融合算法进行路径规划实验。实验中，动态窗口法参数设置如下：最大线速度为 1.5 m/s，最大角速度为 20.0(°)/s，最大线加速度为 0.2 m/s²，最大角加速度为 50.0(°)/s²，采样时间为 0.1 s，d_max=0.5，评价函数参数 α=0.05，β=0.2，γ=0.3，ρ=0.4 实验结果如图 8 所示。

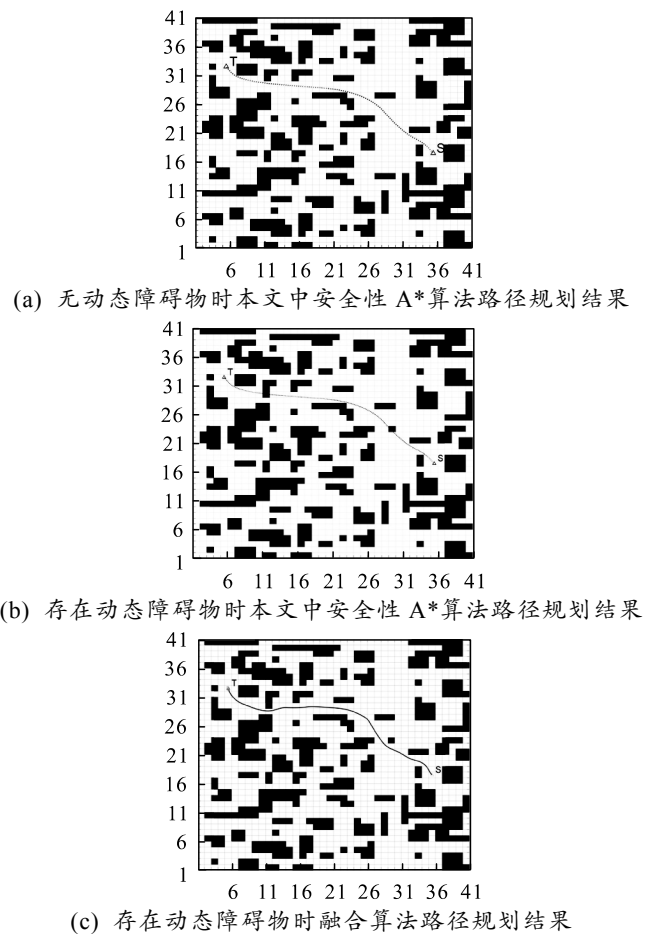


图 8 动态避障路径规划

由上图可知，在地图中没有动态障碍物出现时，本文中安全性 A*算法可以规划出一条全局最优路径；但在所规划全局路径中添加动态障碍物后，安全性 A*算法由于仅适用于全局路径规划而无法避开突然出现的动态障碍物，本文中安全性 A*算法结合动态窗口法的融合算法可以在遵循全局最优路径轨迹的同时，避开机器人行进中随机出现的动态障碍物，且路径平滑。综上所述，笔者提出的融合算法可以得到全局规划最优路径，并实现机器人沿全局路径行进中动态避障且路径平滑。

5 实验验证

仿真算法实验中将机器人化简为一个质点，忽略了实际情况下障碍物和机器人自身的体积。为验证本文中算法在实际应用中的效果，将本文中算法应用到基于树莓派的移动机器人中，实验场景为实验室走廊内的一段过道，其中放置 3 个障碍物，如图 9 所示。实验中，通过改变障碍物位置来模拟动态障碍物。

图 10—11 为 ROS 系统中输出的实验结果。图中 A 处圆点为移动机器人，规划路径中 3 个阴影部分为机器人激光雷达实时采集到的障碍物信息，图

12 为实验中机器人的实时线速度、角速度、姿态参数。

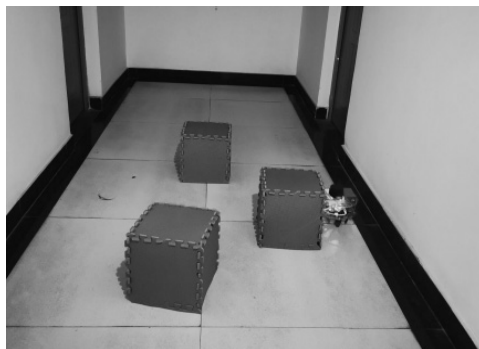


图 9 实验场地

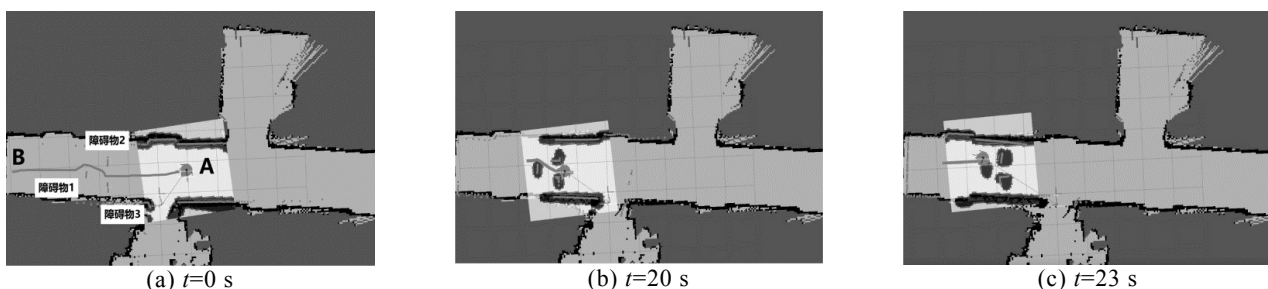


图 10 第 1 次实验规划

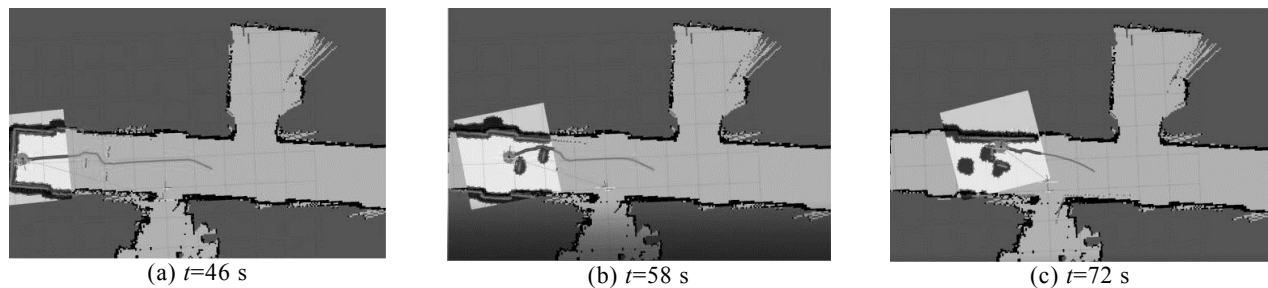


图 11 第 2 次实验规划

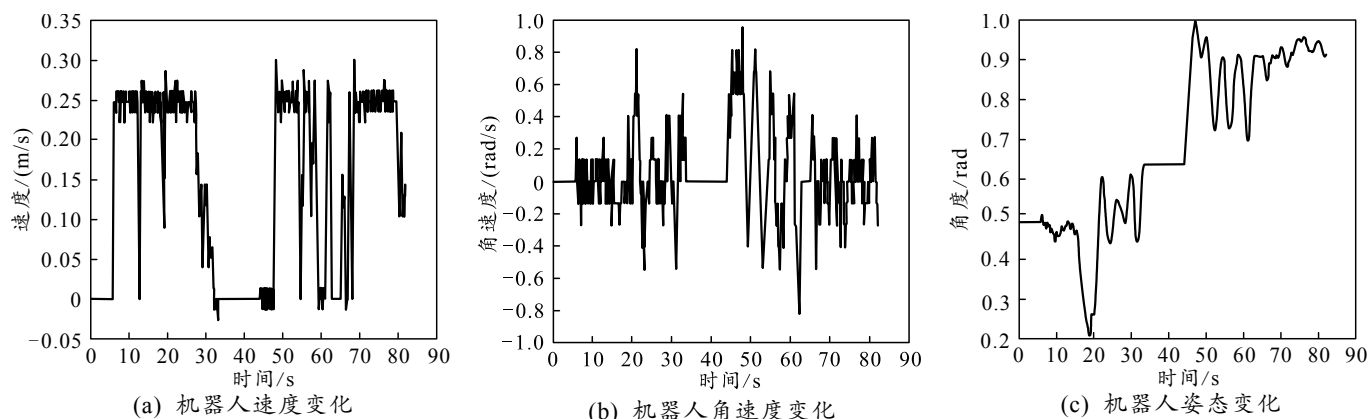


图 12 机器人参数

实验开始后，机器人首先通过激光雷达实时检测环境信息，并对障碍物进行膨胀处理，运用安全性 A*算法规划出当前点至终点的全局最优路径，然后机器人在沿全局路径行进中运用动态窗口法实时规划局部路径。第 1 次实验中，机器人从 A 点到 B

点行进时图 10(a)中 3 个障碍物位置不变(静态障碍)，所以机器人沿图 10(a)中规划的全局路径行走并到达 B 点。第 2 次实验中，机器人从 B 点移动至 A 点，在沿图 11(a)中全局规划路径行走的过程中，检测到前方原全局路径上出现障碍物(障碍物 3 的

位置发生改变)后,重新规划当前点至终点的全局路径,于是机器人局部路径跟随全局路径变化,直至到达 A 点。从图 12 可知,机器人在障碍物附近线速度和角速度降低,当避开障碍物后线速度和角速度增加。经过静态障碍物和动态障碍物实验结果表明,本文中融合算法可使机器人尽量贴合全局最优路径行进的过程中安全地绕开障碍物。

6 结论

笔者引入安全估值函数优化传统 A*算法的估价函数,在拓展目标点方向搜索邻域的基础上将原来的 8 方向搜索精简为 6 方向搜索,大大加快了搜索速度,然后对规划后的路径采用双向删除冗余节点和 3 次 B 样条法进行二次优化,进一步减少路径长度、转折点和转角,得到更平滑的利于机器人行进的路径。实验结果证明本文中安全性 A*算法搜索效率更高,所规划路径更优。

为实现机器人在动态复杂环境下的路径规划,将本文中安全性 A*算法与动态窗口法相融合,并运用改进的评价函数,避免了 DWA 在局部路径规划容易陷入死锁的问题,并在使用局部路径避障的同时尽量遵循全局最优路径。

通过 Matlab 仿真实验和基于树莓派的移动机器人的实际应用,结果表明:该算法在全局路径规划和局部路径规划中都有较好的表现,能满足路径规划的实际需求,具有一定的应用价值。

参考文献:

- [1] WANG Z, LI G, REN J. Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm[J]. *Computer Communications*, 2021, 166: 49-56.
- [2] SONGB Y, WANGZ D, ZOU L. On Global Smooth Path Planning for Mobile Robots using a Novel Multimodal Delayed PSO Algorithm[J]. *Cognitive Computation*, 2017, 9(1): 5-17.
- [3] LIN P, CHOI W Y, CHUNG C C. Local Path Planning Using Artificial Potential Field for Waypoint Tracking with Collision Avoidance[C]//2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2020: 1-7.
- [4] JUNG J H, KIM D H. Local path planning of a mobile robot using a novel grid-based potential method[J]. *International Journal of Fuzzy Logic and Intelligent Systems*, 2020, 20(1): 26-34.
- [5] MAC T T, COPOT C, TRAN D T. Heuristic approaches in robot path planning: A survey[J]. *Robotics and Autonomous Systems*, 2016, 86: 13-28.
- [6] PERALTA F, ARZAMENDIA M, GREGOR D, et al. A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: The ypacarai lake case-study[J]. *Sensors*, 2020, 20(5): 1488.
- [7] AKRAM M, HABIB A, ALCANTUD J C R. An optimization study based on Dijkstra algorithm for a network with trapezoidal picture fuzzy numbers[J]. *Neural Computing and Applications*, 2021, 33: 1329-1342.
- [8] SOLTANI A R, TAWFIK H, GOULERMAS J Y, et al. Path planning in construction sites: Performance evaluation of the Dijkstra, A*, and GA search algorithms[J]. *Advanced Engineering Informatics*, 2002, 16(4): 291-303.
- [9] 曹凯, 高佳佳, 李昂. 基于 RRT 优化算法的机器人路径规划[J]. *兵工自动化*, 2018, 37(9): 74-79.
- [10] 李卫硕, 孙剑, 陈伟. 基于 BP 神经网络机器人实时避障算法[J]. *仪器仪表学报*, 2019, 40(11): 204-211.
- [11] AVERSA D, SARDINA S, VASSOS S. Path planning with Inventory-driven Jump-Point-Search[J]. *ArXiv*, 2016(7): 715.
- [12] CAO P, FAN Z, GAO R X, et al. A focal any-angle path-finding algorithm based on a* on visibility graphs[J]. *ArXiv preprint arXiv*, 2017.
- [13] 王洪斌, 尹鹏衡, 郑维, 等. 基于改进的 A*算法与动态窗口法的移动机器人路径规划[J]. *机器人*, 2020, 42(3): 346-353.
- [14] YANG W, WU P, ZHOU X, et al. Improved Artificial Potential Field and Dynamic Window Method for Amphibious Robot Fish Path Planning[J]. *Applied Sciences*, 2021, 11(5): 2114.
- [15] NEKOO S R, ACOSTA J Á, OLLERO A. Collision Avoidance of SDRE Controller using Artificial Potential Field Method: Application to Aerial Robotics[C]//2020 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2020: 551-556.
- [16] BALLESTEROS J, URDIALES C, VELASCO A, et al. A Biomimetical Dynamic Window Approach to Navigation for Collaborative Control[J]. *IEEE Transactions on Human-Machine Systems*, 2017(6): 1-11.
- [17] LI G, TONG S, CONG F, et al. Improved artificial potential field-based simultaneous forward search method for robot path planning in complex environment[C]//IEEE/SICE International Symposium on System Integration. IEEE, 2016.
- [18] WANG A, HE C, ZHAO G. A Method of Designing Non-Uniform Cubic B-Spline Curve Based on Monotone Curvature[J]. *Journal of Computer-Aided Design & Computer Graphics*, 32(1): 140-146.
- [19] 迟旭, 李花, 费继友. 基于改进 A*算法与动态窗口法融合的机器人随机避障方法研究[J]. *仪器仪表学报*, 2021, 42(3): 132-140.