

DOI: 10.11830/ISSN.1000-5013.202203051



移动机器人全局动态路径规划融合算法

葛文雅, 李平

(华侨大学 信息科学与工程学院, 福建 厦门 361021)

摘要: 针对移动机器人全局动态路径规划效率较低的问题, 提出一种基于安全 A* 算法与双速度模型动态窗口法的全局动态路径规划融合算法. 首先, 通过安全 A* 算法得到全局最优路径节点, 将其作为临时目标节点, 为动态规划提供全局信息, 避免出现局部最优. 然后, 采用时间序列 Bottom-Up 算法减少路径节点数, 从而减少迭代次数、计算代价和储存代价, 提高算法效率. 最后, 采用双速度模型对动态窗口法进行改进, 通过避障重规划机制, 解决全局动态路径规划时移动机器人绕远甚至绕圈的问题, 并通过 MATLAB 平台进行仿真实验. 仿真结果表明: 文中算法的规划效率可提高 46.18%, 保证了路径的安全性和移动机器人速度的平稳性, 文中算法的路径质量和规划效率更佳.

关键词: 路径规划; 动态避障; 双速度模型; 动态窗口法; 移动机器人

中图分类号: TP 242

文献标志码: A

文章编号: 1000-5013(2022)06-0809-10

Global Dynamic Path Planning Fusion Algorithm for Mobile Robot

GE Wenya, LI Ping

(College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China)

Abstract: Aiming at the low efficiency of global dynamic path planning of mobile robot, a global dynamic path planning fusion algorithm based on safety A* algorithm and dynamic window method with the two-velocity model is proposed. Firstly, the global optimal path node is obtained through the safety A* algorithm, which is used as the temporary target node to provide global information for dynamic planning and avoid local optimization. Then, the time series Bottom-Up algorithm is used to reduce the number of path nodes, so as to reduce the number of iterations, computational cost and storage cost, and improve the efficiency of the algorithm. Finally, the two-velocity model is used to enhance the dynamic window method, and through the obstacle avoidance and replanning mechanism to solve the problem of mobile robot going far or even in circle in global dynamic path planning, and the simulation experiment is carried out on the MATLAB platform. The simulation results show that the planning efficiency of the proposed algorithm can be improved by 46.18%, which ensures the safety of the path and the stability of the speed of the mobile robot, the path quality and planning efficiency of the proposed algorithm are better.

Keywords: path planning; dynamic obstacle avoidance; two-velocity model; dynamic window method; mobile robot

收稿日期: 2022-03-29

通信作者: 李平(1981-), 女, 副教授, 博士, 主要从事非线性系统与智能控制、复杂控制系统的研究. 通信方式: E-mail: pingping_1213@126.com.

基金项目: 国家自然科学基金资助项目(61603144); 福建省自然科学基金资助项目(2018J01095); 福建省高校产学研合作科技重大项目(2013H6016); 华侨大学中青年教师科技创新资助计划项目(ZQN-PY509)

路径规划是对初始节点到目标节点的可行路径进行规划. 路径规划的实现应解决以下 2 个问题: 一是规划的路径能够安全、有效地躲避障碍物, 最终抵达目标节点; 二是有效减少移动机器人的运动时间、优化路径轨迹、提高安全性、提升运行效率等^[1-4]. 根据环境为时变模型或时不变模型, 路径规划可分为静态路径规划和动态路径规划^[5]. 通常情况下, 静态路径规划是指移动机器人在能够直接了解全局环境信息的前提下, 规划一条安全可行的路径. 与静态路径规划不同, 动态路径规划应用于全局环境信息不可知或随时间变化的情况, 只能得到当前周围范围内的局部环境信息, 并根据得到的信息完成局部路径规划, 不断调整移动机器人的动作, 直到抵达目标位置^[6]. 移动机器人的实际地图信息多变且复杂, 仅依靠某种算法进行路径规划无法满足应用场景的要求, 人与移动机器人的安全无法得到保障^[7-9]. 移动机器人在规划全局最优路径的同时, 更要具备动态避障的能力^[10]. 因此, 需对各类算法应用时存在的不足进行改进, 把各类算法进行融合, 这是目前路径规划的主要研究方向之一^[11-12].

全局静态路径规划算法能够基于已知的静态信息完成规划, 得到全局最优路径, 但因缺少动态障碍物信息, 无法动态避障. 局部动态路径规划算法能够根据当前输入的传感器信息规划局部路径, 不断更新输入、输出, 实现动态障碍物躲避, 但因缺少全局信息, 易陷入局部最优^[13], 导致目标节点不可达. 两种算法各有优缺点, 故一些学者将这两种算法进行结合. 王志中^[14]提出一种基于 A* 算法与改进人工势场法的融合算法, 但人工势场法规划的路径转角较大, 难以应用于实际. 动态窗口法容易实现, 且规划路径平滑, 能够实现局部避障^[15]. 王凡等^[16]将动态窗口法应用于 A* 算法两个路径节点间的局部规划, 可得到更为平滑的路径, 但算法的运行时间较长. Ji 等^[17]提出一种基于优化 A* 算法和动态窗口法的优化算法, 可提高搜索效率, 但未考虑动态障碍物的情况, 且存在速度震荡的问题. Liu 等^[18]提出一种基于 Jump-A* 算法和动态窗口法的融合算法, 考虑了动态障碍物的情况, 可提高路径的平滑度, 但仍然存在移动机器人速度震荡及绕远的问题. 基于此, 本文提出一种基于安全 A* 算法和双速度模型动态窗口法的移动机器人全局动态路径规划融合算法.

1 安全 A* 算法

A* 算法的路径规划存在路径转角过大, 以及路径和障碍物过近的问题. 因此, 提出安全 A* 算法, 主要从两个方面进行改进: 1) 扩展搜索邻域; 2) 改进启发式函数, 加入一个安全项, 增加样本信息.

1.1 搜索邻域的扩展

针对 A* 算法路径规划存在路径转角过大的问题, 安全 A* 算法将节点 n 的搜索邻域扩展至 24 个. 搜索邻域扩展过程如下: 对于节点 $n(x, y)$, 遍历节点 n 临近的 24 个节点(图 1, 中间黑色区域为节点 n , 周围白色区域为遍历的临近节点, 箭头方向为路径方向), 以作为后续节点 $m(x', y')$, $m = \{(x', y') \mid x' = [x-2, x+2], y' = [y-2, y+2]\}$; 如果后续节点 m 为障碍节点, 或后续节点 m 在 close 表中, 则跳过, 选取下一个临近点作为后续节点; 否则, 计算后续节点的评价函数 $f(m)$, 判断 m 是否在 open 表中, 若判断失败, 将后续节点 m 放入 open 表, 若判断成功, 则选取较小的 $f(m)$, 并更新后续节点 m 的实际代价函数、评价函数及其父节点.

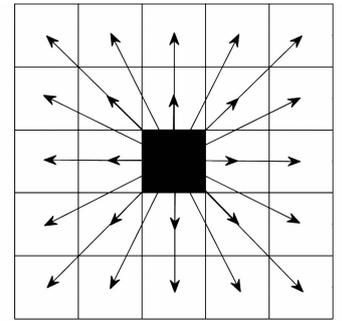


图 1 24 邻域的搜索方向
Fig. 1 Search directions of twenty-four neighborhoods

1.2 启发式函数的改进

扩展节点的搜索邻域可以减小路径转角, 提高移动机器人实际运动中的路径安全性. 然而, 路径和障碍物的距离仍然很近, 路径的安全性还有较大的提升空间. A* 算法的核心在于启发式函数 $h(n)$ 的设计, 安全 A* 算法采用 A* 算法对当前轨迹节点进行评估的思想, 并增加了节点及一定范围内的障碍物信息, 增大了路径和障碍物的距离, 从而具有更高的路径安全性.

安全 A* 算法的评价函数 $f(n)$ 为

$$f(n) = g(n) + h'(n), \quad h'(n) = h(n) + \epsilon L(n), \quad L(n) = k/s. \quad (1)$$

式(1)中: $g(n)$ 为实际代价函数, 即当前已走过的路径距离; $h'(n)$ 为安全 A* 算法的启发式函数; $L(n)$ 为节点 n 对应的危险评估值(安全项); k 为评价范围内障碍物的个数; s 为移动机器人与障碍物的最小距

离; ϵ 为安全项 $L(n)$ 的权重, 文中设定为 100.

1.3 安全 A* 算法的流程

open 表存储搜索过程中的待扩展节点, 并将这些节点按照评价函数值进行升序排序; close 表保存 open 表中评价函数值最小的后续节点; safe 表保存后续节点的安全性指数 S . 在执行路径规划时, 安全 A* 算法主要通过 open 表和 close 表实现节点的扩展和最优节点的选取.

安全 A* 算法的流程有以下 7 个步骤.

步骤 1 将初始节点纳入 open 表, 将障碍节点纳入 close 表.

步骤 2 选取 open 表中评价函数值最小的后续节点, 将其纳入 close 表, 将后续节点的安全性指数纳入 safe 表, 并从 open 表删除此节点.

步骤 3 判断该节点是否为目标节点, 若为目标节点, 则算法退出, 返回当前节点信息; 若该节点非目标节点, 扩展该节点的后续节点 m .

步骤 4 在 open 表中建立从后续节点 m 返回 n 的指针, 并计算 $f(m)$, 即

$$f(m) = g(m) + h'(m), \quad h'(m) = h(m) + \epsilon L(m), \quad L(m) = k/s. \quad (2)$$

步骤 5 判断 open 表中是否存在后续节点 m , 若结果为假, 则将节点 m 纳入 open 表, 并将节点 m 的安全性指数纳入 safe 表; 若结果为真, 则比较不同前向指针的 $f(m)$ 的大小, 并选取较小的 $f(m)$.

步骤 6 更新 $g(m)$, $f(m)$ 及后续节点 m 的前向指针.

步骤 7 按照评价函数值对 open 表重新进行升序排序, 并返回步骤 2.

1.4 基于安全 A* 算法的全局路径规划

采用栅格法^[19]构建地图模型, 栅格法是移动机器人环境建模的常用方法, 它将移动机器人的工作空间划分为网格单元. 栅格地图模型, 如图 2 所示. 图 2 中: 黑色网格为障碍物信息; 白色格子为移动机器人可移动区域.

基于栅格地图, 采用安全 A* 算法进行全局静态规划, 可以得到全局最优路径及安全性曲线, 如图 3 所示. 由图 3 可知: 全局最优路径平滑且距离障碍物较远; 路径的安全性指数始终保持在 10, 路径安全性很高.

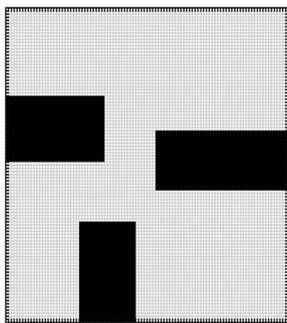
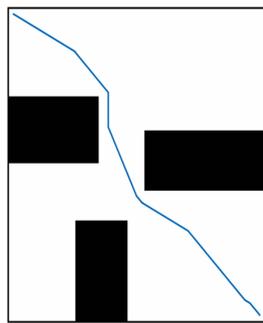
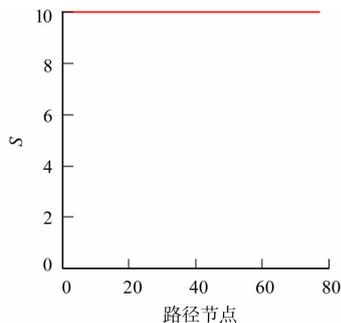


图 2 栅格地图模型

Fig. 2 Raster map model



(a) 全局最优路径



(b) 安全性曲线

图 3 全局最优路径及安全性曲线

Fig. 3 Global optimal path and security curve

2 动态窗口法

动态窗口(DWA)法常用于存在动态障碍环境的路径规划^[20], 它基于采样的当前状态计算下一时刻的合理轨迹, 为每一轨迹附加评估值, 并根据评估值选取该状态的最优预估轨迹^[21-22].

2.1 移动机器人运动模型

动态窗口法需要在每个采样时刻计算下一时刻所有的可能速度, 通过运动模型预估下一时刻所有的可能路径. 因此, 应用动态窗口法首先应建立研究对象的运动模型, 模型的准确度会影响算法的实际控制效果.

假设采样时间间隔(Δt)很短, 移动机器人的线速度(v)控制加速与减速, 角速度(ω)负责转向, 记当前时刻(t)移动机器人位姿为(x_t, y_t, θ_t), x_t, y_t 分别为当前时刻移动机器人在 x 轴和 y 轴的位置, θ_t 为

当前时刻移动机器人与 x 轴的夹角;下一时刻的预估速度为 (v_{t+1}, ω_{t+1}) , v_{t+1}, ω_{t+1} 分别为下一时刻移动机器人的线速度和角速度. 由于采样时间间隔很短,故将移动机器人运动的圆弧轨迹近似为直线,下一时刻的预估位姿记为 $(x_{t+1}, y_{t+1}, \theta_{t+1})$, 计算公式为

$$\left. \begin{aligned} x_{t+1} &= x_t + v_x \Delta t \cos \theta_t - v_y \Delta t \sin \theta_t, \\ y_{t+1} &= y_t + v_x \Delta t \sin \theta_t + v_y \Delta t \cos \theta_t, \\ \theta_{t+1} &= \theta_t + \omega_t \Delta t. \end{aligned} \right\} \quad (3)$$

式(3)中: v_x, v_y 分别为 v_{t+1} 投影在 x 轴和 y 轴上的线速度; ω_t 为当前时刻的角速度.

2.2 移动机器人速度模型

根据移动机器人运动模型,可通过当前位姿和施加的速度控制量预估下一时刻的运动轨迹. 基于当前时刻的速度 (v_t, ω_t) , 构造下一时刻速度的采样空间,理论上存在无数下一时刻的预估速度 (v_{t+1}, ω_{t+1}) . 由于受限于移动机器人速度极限、动力学性能及环境等因素,移动机器人的速度被约束在一定的范围内.

1) 速度极限约束. 在物理系统中,受限于移动机器人本身的设计与结构,移动机器人的线速度与角速度都有上限和下限.

移动机器人受速度极限约束所能达到的速度空间 V_m 为

$$V_m = \{(v, \omega) \mid v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\}. \quad (4)$$

式(4)中: v_{\max}, v_{\min} 分别为线速度的上限和下限; $\omega_{\max}, \omega_{\min}$ 分别为角速度的上限和下限.

2) 动力学性能约束. 每个移动机器人的电机性能并非完全相同,导致移动机器人的加速、减速也不相同,因此,移动机器人在下一个时刻的速度会受到约束.

移动机器人受动力学性能约束所能达到的速度空间 V_g 为

$$V_g = \{(v, \omega) \mid v \in [v_t - a_{v,\min} \Delta t, v_t + a_{v,\max} \Delta t], \omega \in [\omega_t - a_{\omega,\min} \Delta t, \omega_t + a_{\omega,\max} \Delta t]\}. \quad (5)$$

式(5)中: $a_{v,\min}, a_{v,\max}$ 分别为线加速度的最小值和最大值.

3) 最小制动距离约束. 计算路径规划预估速度的最小制动距离,判断其对障碍物的躲避能力,以确保移动机器人移动的安全性.

移动机器人受最小制动距离约束所能达到的速度空间 V_a 为

$$V_a = \{(v, \omega) \mid v \leq \sqrt{2 \times \text{dist}(v, \omega) \times v}, \omega \leq \sqrt{2 \times \text{dist}(v, \omega) \times \omega}\}. \quad (6)$$

式(6)中: $\text{dist}(v, \omega)$ 为速度 (v, ω) 预测的机器人和最近障碍物之间的距离(距离评价函数).

由此可得,移动机器人运动速度空间合集 V 为

$$V = V_m \cap V_g \cap V_a. \quad (7)$$

2.3 评价函数

考虑如何施加合适的控制量,使得到的速度采样空间能够规划出更优的轨迹. 所有的控制量对应一个预估速度,经计算可得对应的预估轨迹,所有的预估轨迹组成的预估轨迹集即为一个动态窗口,动态窗口在每个采样时间内完成更新. 评价函数是对动态窗口内的全部预估轨迹进行评价,选择评价价值最高的预估轨迹作为下一时刻的目标轨迹. 评价函数主要从方位角、距离及速度3个方面进行设计.

1) 方位角评价函数 $\text{heading}(v, \omega)$. 方位角是指移动机器人按照采样得到的线速度和角速度运动到预估轨迹的末端时,移动机器人的航向与移动机器人中心指向目标节点方向的角度差. 角度差越大,方位角评价函数值越小.

2) 距离评价函数 $\text{dist}(v, \omega)$. 移动机器人按照采样得到的线速度和角速度运动到预估轨迹的末端时,移动机器人和最近障碍物之间的距离为 $\text{dist}(v, \omega)$. 距离评价函数值越高,轨迹越安全,舍弃与障碍物相交的轨迹.

3) 速度评价函数 $\text{vel}(v, \omega)$. 采样时刻移动机器人的速度会影响移动机器人的运行时间与工作效率. 在预估速度集中,速度评价函数值与速度的大小成正比.

总评价函数 $G(v, \omega)$ 为

$$G(v, \omega) = \alpha \text{heading}(v, \omega) + \beta \text{dist}(v, \omega) + \gamma \text{vel}(v, \omega). \quad (8)$$

式(8)中: α, β, γ 分别为方位角评价函数、距离评价函数、速度评价函数的权重系数.

2.4 基于安全 A* 算法和动态窗口法的动态路径规划

动态窗口法根据得到的局部环境信息可以很好地躲避静态或动态的障碍物, 得到一条不发生碰撞的最优路径. 然而, 动态窗口法只根据局部环境信息进行轨迹规划, 缺少全局环境信息的指引, 很容易陷入局部最优, 甚至出现无法顺利到达目标位置等非常严重的问题.

之前, 学者提出基于 A* 算法和动态窗口法的融合算法, 规划时将 A* 算法规划得到的全局路径节点作为临时目标节点, 到达路径节点 p_i 后, 路径节点 p_{i+1} 将成为下一个临时的目标节点, 直至到达目标节点. 经过实验仿真, 该算法能使移动机器人成功抵达目标位置, 但移动机器人接近一个临时目标节点时, 速度会急速下降, 甚至直接降为 0, 导致移动机器人整体运行不够平稳.

进一步地, 对基于安全 A* 算法和动态窗口法的融合算法(初步融合算法)进行仿真实验, 可得初步融合算法的速度曲线, 如图 4 所示. 图 4 中: t_p 为路径规划时间.

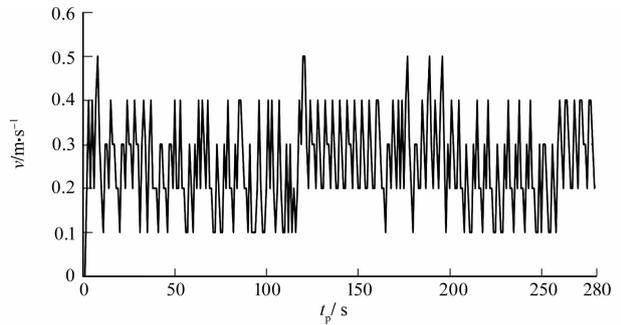


图 4 初步融合算法的速度曲线

Fig. 4 Velocity curve of preliminary fusion algorithm

3 双速度模型动态窗口法

初步融合算法进行规划时, 无法区别临时目标节点和全局目标节点, 以致无差别地给出执行减速停止的指令. 因此, 需对动态窗口法的速度模型进行改进, 采用双速度模型.

当判定临时目标节点为全局目标节点, 或距离临时目标节点还有一定距离时, 速度模型不变, 仍采用式(4)~(6)的约束条件; 当判定临时目标节点不是全局目标节点, 且距离临时目标节点小于一定距离时, 线速度保持不变(仍为上一时刻的线速度($v_{t-\Delta t}$)), 角速度选取原则不变.

此时, 双速度模型如下.

1) 速度极限约束. 移动机器人受速度极限约束所能达到的速度空间 V'_m 为

$$V'_m = \{(v, \omega) \mid v = v_{t-\Delta t}, \omega \in [\omega_{\min}, \omega_{\max}]\}. \quad (9)$$

2) 动力学性能约束. 移动机器人受动力学性能约束所能达到的速度空间 V'_g 为

$$V'_g = \{(v, \omega) \mid v = v_{t-\Delta t}, \omega \in [\omega_t - a_{v, \min} \Delta t, \omega_t + a_{v, \max} \Delta t]\}. \quad (10)$$

3) 最小制动距离约束. 为了得到安全的移动轨迹, 移动机器人的最小制动距离应小于移动机器人与最近障碍物之间距离. 移动机器人受最小制动距离约束所能达到的速度空间 V'_a 为

$$V'_a = \{(v, \omega) \mid v = v_{t-\Delta t}, \omega \leq \sqrt{2 \times \text{dist}(v, \omega) \times \omega}\}. \quad (11)$$

由此可得双速度模型的速度空间合集为

$$\left. \begin{aligned} V_1 &= V_m \cap V_g \cap V_a, & p &= G \cup \text{dist } p > d, \\ V_2 &= V'_m \cap V'_g \cap V'_a, & p &\neq G \cap \text{dist } p \leq d. \end{aligned} \right\} \quad (12)$$

式(12)中: p 为临时目标节点; G 为全局目标节点; $\text{dist } p$ 为移动机器人当前位置和临时目标节点之间的距离; d 为常数, 文中取 3.

4 基于安全 A* 算法和双速度模型动态窗口法的融合算法

基于安全 A* 算法和双速度模型动态窗口法, 提出移动机器人全局动态路径规划融合算法(文中算法). 采用基于栅格地图的安全 A* 算法进行全局静态路径规划, 可得全局最优路径; 采用时间序列 Bottom-Up 算法对全局最优路径进行特征点提取, 以作为临时目标节点; 在全局最优路径的信息引导下, 采用双速度模型动态窗口法进行动态规划与避障, 躲避动态障碍物后, 采用避障重规划机制进行路径重规划, 确保规划路径的全局最优.

文中算法的思路框架, 如图 5 所示.

4.1 时间序列 Bottom-Up 算法

针对现有融合算法中临时目标节点过多导致效率较低、计算代价较大的问题, 采用时间序列 Bot-



图5 文中算法的思路框架

Fig. 5 Thinking frameworks of proposed algorithm

tom-Up 算法,对安全 A* 算法得到的全局最优路径节点进行预处理,拟合分段得到特征路径节点作为临时目标节点,在保留原路径特征的前提下,减少全局路径节点的数量.特征路径节点的选取,如图6所示.图6中:蓝色实线为安全 A* 算法得到的全局最优路径;蓝色空心圆为安全 A* 算法得到的全局最优路径节点(80个);红色实心圆为经过时间序列 Bottom-Up 算法预处理后的特征路径节点(7个).由图6可知:临时目标节点的数量由80个减少为7个,有效降低了储存和计算代价,提高了路径规划的效率.

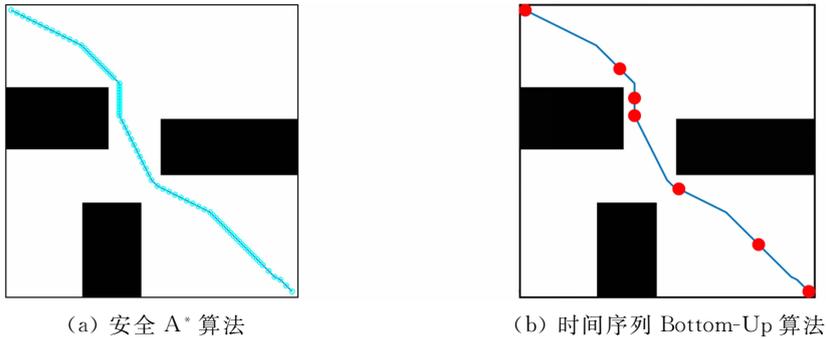


图6 特征路径节点的选取

Fig. 6 Selection of feature path nodes

4.2 避障重规划机制

当环境中出现未知障碍物和动态障碍物时,初步融合算法根据当前窗口不断更新速度(v, ω),使移动机器人顺利地躲避障碍物.由于脱离原全局最优路径已有一段时间,当重新恢复至原来模式时,会出现移动机器人绕远甚至绕圈的问题(图7的红色路径).这是由于避障模式结束后,原临时目标节点没有改变,但此时移动机器人很可能已经走过原临时目标节点,且方位角无法立即改变.此时,如果不改变临时目标节点,移动机器人则会继续按照原临时目标节点进行规划,出现移动机器人绕远甚至绕圈的问题.为了解决此问题,提出一种避障重规划机制.在路径规划的过程中,全程监控临时目标节点、移动机器人当前位置及下一临时目标节点三者形成的夹角(θ_a).如果 $\theta_a < 90^\circ$,则临时目标节点保持不变,继续进行路径规划;如果 $\theta_a \geq 90^\circ$,当前规划可能出现移动机器人绕远甚至绕圈的问题,则重新选择下一特征节点作为临时目标节点进行路径规划.夹角的不同情形,如图8所示.

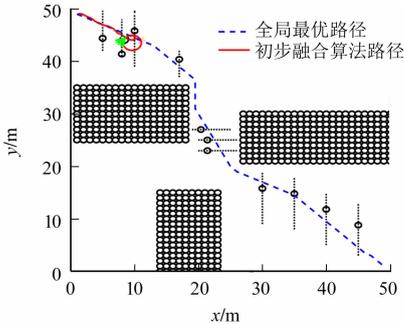


图7 初步融合算法动态规划路径绕圈

Fig. 7 Preliminary fusion algorithm dynamic planning path in circle

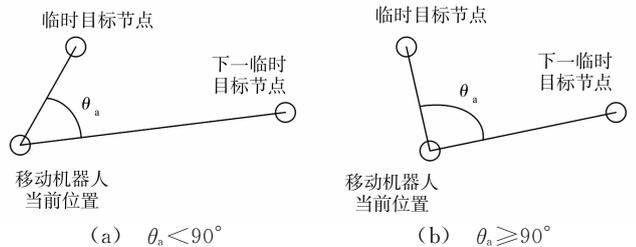


图8 夹角的不同情形

Fig. 8 Different cases of included angles

4.3 文中算法流程

动态窗口法根据实时获取的局部环境信息,实现路径规划和动态避障,但由于忽略全局信息,导致

易陷入局部最优、目标节点不可达等问题. 因此, 文中算法首先通过安全 A* 算法进行全局静态路径规划; 然后, 将规划得到最优路径上的节点作为临时目标节点, 两个临时目标节点之间采用动态窗口法实现动态规划, 融合了静态规划的全局信息和动态规划的动态避障. 若简单进行两种方法的融合, 则会出现移动机器人运动不平滑、速度震荡、移动机器人绕远且效率低等问题. 针对以上问题使用时间序列 Bottom-Up 算法对安全 A* 算法得到的全局最优路径进行预处理, 提取全局最优路径的特征节点作为融合算法的临时目标节点, 并对动态窗口法的速度模型进行改进, 基于全局最优路径信息, 采用双速度模型动态窗口法进行动态规划, 动态避障后采用避障重规划机制, 避免出现移动机器人绕远甚至绕圈的问题. 文中算法流程图, 如图 9 所示.

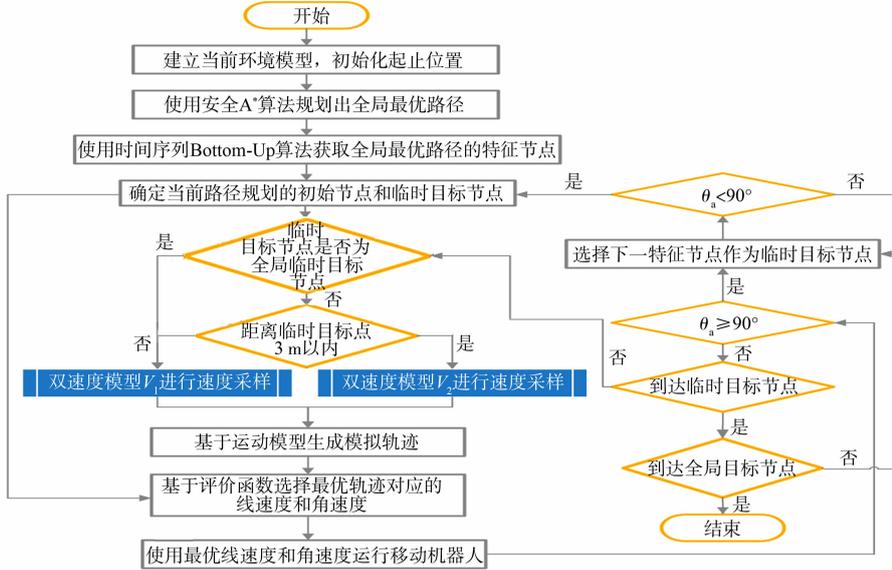


图 9 文中算法流程图

Fig. 9 Flow chart of proposed algorithm

5 仿真结果与分析

仿真实验的主要参数设置如下: $v_{max} = 1.0 \text{ m} \cdot \text{s}^{-1}$; $\omega_{max} = 20 \text{ rad} \cdot \text{s}^{-1}$; $\Delta t = 0.1 \text{ s}$; $\alpha = 0.05$; $\beta = 0.5$; $\gamma = 0.1$; $a_{v,max} = 0.2 \text{ m} \cdot \text{s}^{-2}$; 角加速度最大值 $a_{\omega,max} = 50 \text{ rad} \cdot \text{s}^{-2}$.

两种算法的路径轨迹, 如图 10 所示. 由图 10 可知: 初步融合算法和文中算法都能在静态环境下根据全局信息的指引进行全局路径规划; 文中算法路径更加平滑.

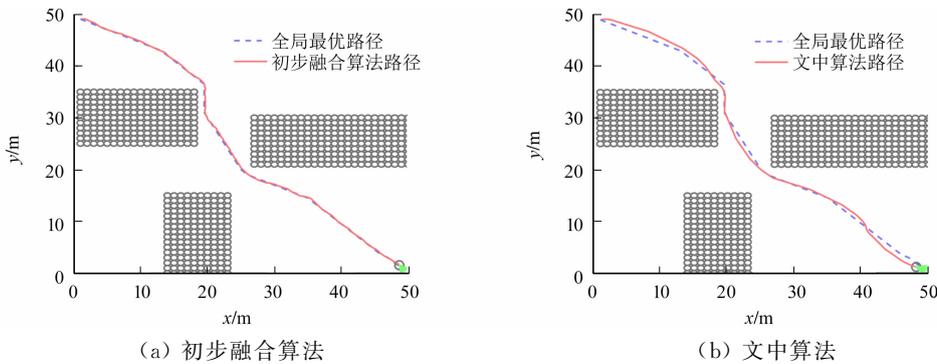


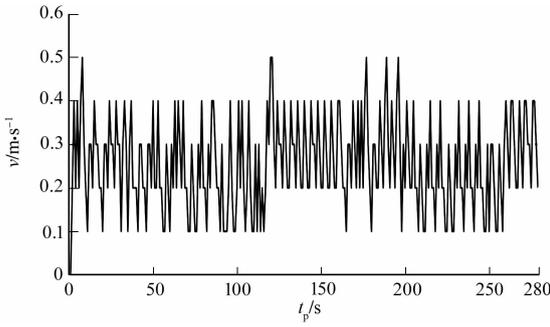
图 10 两种算法的路径轨迹

Fig. 10 Path trajectories of two algorithms

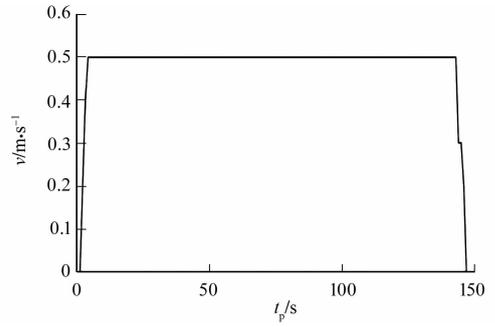
两种算法的速度曲线, 如图 11 所示. 由图 11 可知: 初步融合算法的移动机器人多次加速、减速, 速度曲线震荡剧烈, 导致移动机器人运行不稳定, 且规划时间较长, 效率不高; 文中算法的速度曲线平滑, 移动机器人运行平稳, 规划时间较短.

因此, 文中算法可解决移动机器人减速导致的运行不平稳、速度震荡等问题, 提高了移动机器人的

运行效率.



(a) 初步融合算法



(b) 文中算法

图 11 两种算法的速度曲线

Fig. 11 Velocity curves of two algorithms

算法性能指标的对比,如表 1 所示.表 1 中: l 为路径长度.由表 1 可知:算法改进前后,路径长度几乎没有变化,说明文中算法仍然可以较好地在全局最优路径信息的指引下进行规划,以保证搜索路径的全局最优性和安全性;路径规划时间从 202.24 s 缩短到 108.85 s,文中算法的规划效率提高了 46.18%.

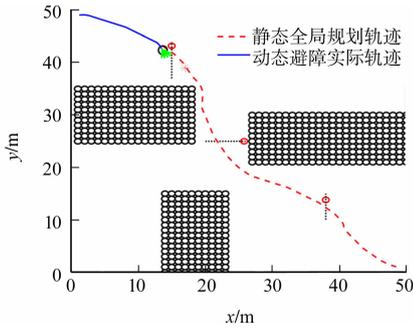
表 1 算法性能指标的对比

Tab. 1 Comparison of performance indexes of algorithms

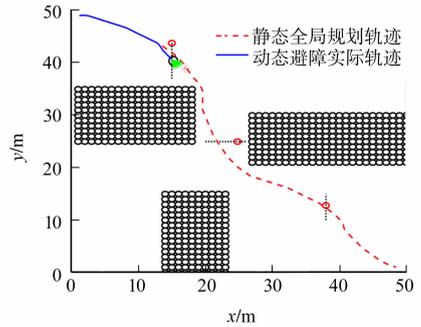
算法	l/m	t_p/s
初步融合算法	71.60	202.24
文中算法	71.80	108.85

为了验证文中算法的动态避障性能,在环境中增加 3 个持续来回运动的障碍物.文中算法的动态避障规划,如图 12 所示.图 12 中:红色圆形为动态障碍物,按照黑色虚线来回移动;蓝色实线为文中算法进行全局动态规划时遇见动态障碍物进行避障的实际规划路径(动态避障实际轨迹);红色虚线为文中算法在没有动态障碍物时进行全局动态规划的路径(静态全局规划轨迹).

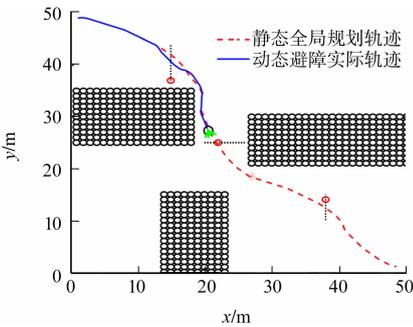
由图 12 可知:文中算法能够躲避动态障碍物,且未出现初步融合算法中移动机器人绕远甚至绕圈的问题,可成功地完成全局动态路径规划任务.



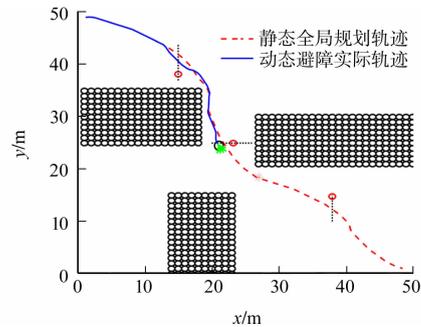
(a) 避障前(1号动态障碍物)



(b) 避障后(1号动态障碍物)



(c) 避障前(2号动态障碍物)



(d) 避障后(2号动态障碍物)

文中算法全局动态规划路径与速度曲线,如图 13 所示.由图 13 可知:文中算法可在全局最优路径信息的指引下很好地完成全局动态路径规划,躲避动态障碍物,保证路径的平滑性和安全性;文中算法

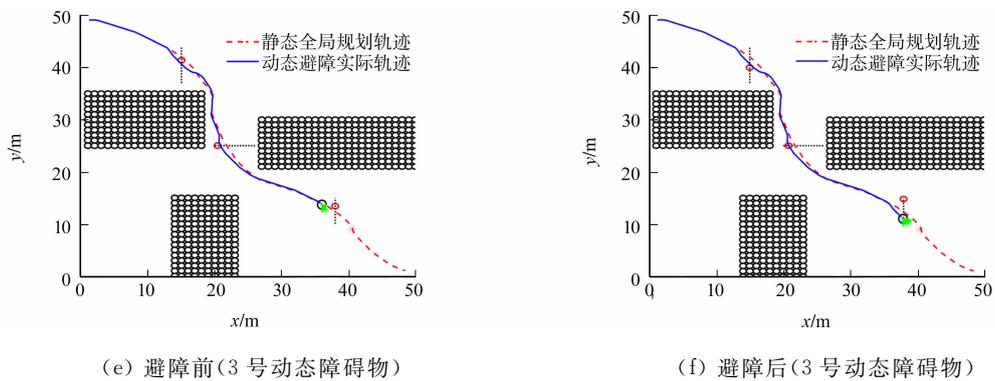


图 12 文中算法的动态避障规划

Fig. 12 Dynamic obstacle avoidance planning of proposed algorithm

动态避障规划时对应的移动机器人速度曲线较为平滑,有少数减速;文中算法速度平稳,遇到动态障碍物时能够适当减速,可保证移动机器人整体运行的安全性。

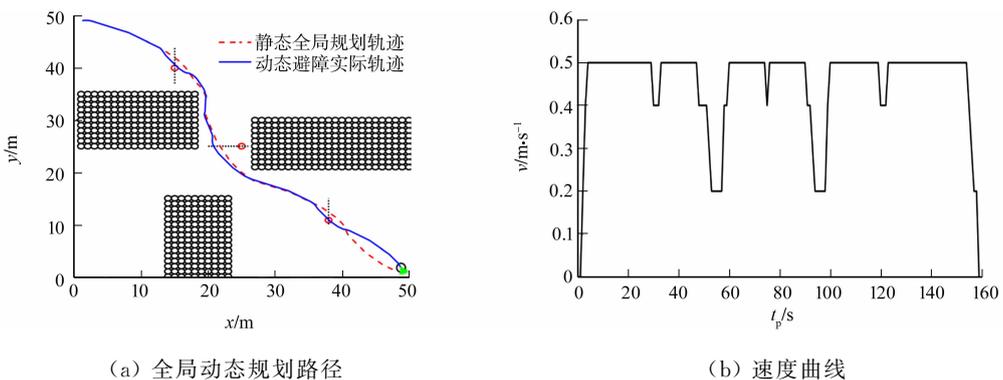


图 13 文中算法全局动态规划路径与速度曲线

Fig. 13 Global dynamic planning path and velocity curve of proposed algorithm

由仿真实验可知,文中算法能够很好地解决 A* 算法和动态窗口法融合过程中移动机器人运动速度曲线不平滑、速度震荡及移动机器人绕远导致规划效率低的问题;文中算法遇到未知障碍物时也能表现出很好的性能。因此,文中算法能够高效地完成全局动态规划任务,并保证路径的安全性和最优性。

6 结束语

针对移动机器人全局动态路径规划效率较低的问题,提出一种基于安全 A* 算法与双速度模型动态窗口法的全局动态路径规划融合算法。通过时间序列 Bottom-Up 算法对安全 A* 算法规划的全局最优路径进行预处理,将全局最优路径的特征节点作为临时目标节点,减少临时目标节点的数量;使用改进后的双速度模型动态窗口法,基于全局最优信息进行动态规划,使移动机器人的运行速度平稳;通过避障重规划机制,解决动态避障后的路径重规划问题,避免移动机器人出现绕远甚至绕圈的问题。仿真结果表明,文中算法的规划效率提高了 46.18%,可保证路径的安全性和最优性。

参考文献:

- [1] 胡春阳,姜平,周根荣. 改进蚁群算法在 AGV 路径规划中的应用[J]. 计算机工程与应用,2020,56(8):270-278. DOI:10.3778/j.issn.1002-8331.1911-0282.
- [2] TAI Lei,LIU Ming. A robot exploration strategy based on Q-learning network[C]//IEEE International Conference on Real-Time Computing and Robotics. Angkor Wat; IEEE Press, 2016: 57-62. DOI: 10.1109/RCAR.2016.7784001.
- [3] MAC T T,COPOT C,TRAN D T,*et al.* Heuristic approaches in robot path planning: A survey[J]. Robotics and Autonomous Systems,2016,86:13-28. DOI:10.1016/j.robot.2016.08.001.
- [4] YU Lijun,WEI Zhihong,WANG Hui,*et al.* Path planning for mobile robot based on fast convergence ant colony al-

- gorithm[C]//IEEE International Conference on Mechatronics and Automation, Takamatsu;IEEE Press,2017;1493-1497. DOI:10.1109/ICMA.2017.8016037.
- [5] KWAME A B,RYAD C,YAW A A,*et al.* An overview of nature-inspired, conventional, and hybrid methods of autonomous vehicle path planning[J]. Journal of Advanced Transportation,2018,2018;1-27. DOI:10.1155/2018/8269698.
- [6] 张志文,张鹏,毛虎平,等.融合改进A*算法和动态窗口法的全局动态路径规划[J].电光与控制,2021,28(9):39-43. DOI:10.3969/j.issn.1671-637X.2021.09.009.
- [7] 林依凡,陈彦杰,何炳蔚,等.无碰撞检测RRT*的机器人运动规划方法[J].仪器仪表学报,2020,41(10):257-267. DOI:10.19650/j.cnki.cjsi.J2006856.
- [8] DOOPALAM T,BYAMBAA D,JIN L D. Hybrid motion planning method for autonomous robots using kinect based sensor fusion and virtual plane approach in dynamic environments[J]. Journal of Sensors,2015(5):1-13. DOI:10.1155/2015/471052.
- [9] JIAO Zhuqing,MA Kai,RONG Yiling,*et al.* A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs[J]. Journal of Computational Science,2018,25:50-57. DOI:10.1016/j.jocs.2018.02.004.
- [10] FAZLOLLAHTABAR H,HASSANLI S. Hybrid cost and time path planning for multiple autonomous guided vehicles[J]. Applied Intelligence,2018,48(2):482-498. DOI:10.1007/s10489-017-0997-x.
- [11] 李志锟,黄宜庆,徐玉琼.改进逐步长蚁群算法的机器人路径规划[J].电子测量与仪器学报,2020,32(8):15-21. DOI:10.13382/j.jemi.B1902697.
- [12] 张旭,程传奇,郝向阳,等.一种兼顾全局与局部特性的机器人动态路径规划算法[J].测绘科学技术学报,2018,35(3):315-320. DOI:10.3969/j.issn.1673-6338.2018.03.018.
- [13] 贾丙佳,李平.邻近障碍物整体化的机器人路径规划[J].华侨大学学报(自然科学版),2019,40(6):799-805. DOI:10.11830/ISSN.1000-5013.201903010.
- [14] 王志中.复杂动态环境下自主机器人路径规划研究[J].组合机床与自动化加工技术,2018(1):150-154. DOI:10.19287/j.cnki.1005-2402.2018.02.031.
- [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. DOI:10.7652/xjtub201711019.
- [16] 王凡,李铁军,刘今越,等.基于BIM的建筑机器人自主路径规划及避障研究[J].计算机工程与应用,2020,56(17):224-230. DOI:10.3778/j.issn.1002-8331.1911-0178.
- [17] JI Xianyou,FENG Shuo,HAN Qidong,*et al.* Improvement and fusion of A* algorithm and dynamic window approach considering complex environmental information[J]. Arabian Journal for Science and Engineering,2021,46:7445-7459. DOI:1007/s13369-021-05445-6.
- [18] LIU Lisan,YAO Jinxin,HE Dongwei,*et al.* Global dynamic path planning fusion algorithm combining Jump-A* algorithm and dynamic window approach[J]. IEEE Access,2021,9:19632-19638. DOI:10.1109/ACCESS.2021.3052865.
- [19] 程向红,祁艺.基于栅格法的室内指示路径规划算法[J].中国惯性技术学报,2018,26(2):236-240,267. DOI:10.13695/j.cnki.12-1222/o3.2018.02.015.
- [20] 任孝平,蔡自兴.基于阿克曼原理的车式移动机器人运动学建模[J].智能系统学报,2009,4(6):534-537. DOI:10.3969/j.issn.1673-4785.2009.06.011.
- [21] FOX D,BURGARD W. The dynamic window approach to collision avoidance[J]. IEEE Robotics and Automation Magazine,1997,4(1):23-33. DOI:10.1109/100.580977.
- [22] TANG Hongwei,SUN Wei,YU Hongshan,*et al.* A multirobot target searching method based on bat algorithm in unknown environments[J]. Expert Systems with Application,2020,141:112945(1-12). DOI:10.1016/j.eswa.2019.112945.

(责任编辑:钱筠 英文审校:吴逢铁)