

DOI:10.19651/j.cnki.emt.2210160

复杂动态环境下基于 A^* 的改进 DWA 算法研究

邓云峰 黄翼虎

(青岛科技大学自动化与电子工程学院 青岛 266042)

摘要: 针对传统 DWA 算法在复杂动态环境中容易陷入局部最优和动态避障性差等问题,提出一种基于 A^* 的改进 DWA 算法。首先在 DWA 算法中引入碰撞锥对静、动障碍物进行避障检测,并通过速度窗口剔除掉具有碰撞威胁的速度来优化约束空间。其次根据静、动障碍物信息改进评价函数,提高 DWA 算法的轨迹评估能力。最后将改进的 DWA 算法与 A^* 算法相融合,解决 DWA 算法在复杂环境中容易陷入局部最优的问题。仿真结果显示,相比于其他同类型算法,该算法在行进速度和安全距离方面均可提高 50% 以上,不但可以使机器人按照全局最优路径行进,而且有效提高了机器人在复杂动态环境的避障能力。

关键词: 复杂动态环境; A^* 算法; 动态避障; DWA 算法; 碰撞锥; 路径规划

中图分类号: TP242 **文献标识码:** A **国家标准学科分类代码:** 510.80

Research on improved DWA algorithm based on A^* in complex dynamic environment

Deng Yunzheng Huang Yihu

(College of Automation and Electronic Engineering, Qingdao University of Science and Technology, Qingdao 266042, China)

Abstract: Aiming at the problems that traditional DWA algorithm is easy to fall into local optimum and dynamic obstacle avoidance is poor in complex dynamic environment, an improved DWA algorithm based on A^* was proposed. Firstly, a collision cone is introduced into the DWA algorithm to detect the static and moving obstacles, and the speed with collision threat is eliminated through the speed window to optimize the constraint space. Secondly, the evaluation function is improved according to the static and moving obstacle information to improve the trajectory evaluation ability of DWA algorithm. Finally, the improved DWA algorithm is integrated with A^* algorithm to solve the problem that DWA algorithm is easy to fall into local optimum in complex environment. Simulation results show that compared with other similar algorithms, the proposed algorithm can improve the traveling speed and safe distance by more than 50%, which not only can make the robot travel according to the global optimal path, but also effectively improve the robot's obstacle avoidance ability in complex dynamic environment.

Keywords: complex dynamic environment; A^* algorithm; dynamic obstacle avoidance; DWA algorithm; collision cone; path planning

0 引言

近年来,随着科技的不断发展,移动机器人已经在社会多个领域得到广泛的应用,成为社会发展不可或缺的部分。而路径规划作为移动机器人领域的关键部分,其研究和发

展具有重要的意义。现阶段许多已知环境下的路径规划算法已经成熟,能够使机器人实现无碰撞运行。但随着机器人工作环境愈加复杂,要求机器人具备更强的避障能力。如何快速准确地

在复杂动态环境中搜索出一无碰撞路径成为当前机器人避障的技术难点^[1]。

路径规划分为全局路径规划和局部路径规划。全局路径规划是在已知的环境地图中规划一条无碰撞路径。常见的全局路径规划算法有 Dijkstra 算法^[2]、 A^* 算法^[3-4]、蚁群算法^[5-6]等。而实际运行环境中的障碍物是不确定的,当环境中出现未知和动态的障碍物时,全局规划便不能及时规划出新的路径。而局部路径规划可以根据传感器感知周围环境,躲避未知和动态障碍物,具有实时性的优点。常见的局部路径规划有人工势场法^[7]、快速扩展随机树算法 (rapidly-exploring random trees, RRT)^[8]、动态窗口法 (dynamic window approach, DWA)^[9]。

收稿日期:2022-08-23

DWA 算法是在速度空间内采样线速度和角速度,根据机器人的运动学模型预测其下一时间间隔的轨迹,并对评价轨迹进行评分,从而获得更加安全、平滑路径的常用算法。该算法充分考虑了机器人自身的运动特性,具有运算量小,可靠性高的优点。但 DWA 算法也存在以下缺点:一是没有全局导航能力,算法在遇到“C”型障碍物时容易陷入局部最优导致规划失败;二是不能合理选择避障速度,动态避障性差^[10]。针对以上问题,我国的许多研究学者做出了改进。王永雄等^[11]提出一种自适应 DWA 算法,即通过障碍物距离自动调整速度权值来提高 DWA 算法在稠密障碍物的通过性,规划出的路径更安全。常新新等^[12]提出通过障碍物方位删除具有碰撞威胁的速度组合,减少了需要评价的轨迹,提高了 DWA 的算法避障效率。戴凯龙等^[13]通过检测一定距离的障碍物来判断 DWA 算法是否陷入失效环境,并通过设置逃逸子目标点寻找可行区域,避免算法陷入局部最优。槐创锋等^[14]通过扩展 A* 的搜索节点优化全局搜索路径,并结合 DWA 算法成功使机器人实现动态路径规划。袁千贺等^[15]通过引入障碍率自适应调节 A* 算法的评价函数来提高全局搜索效率。并结合 DWA 算法实现了全局路径最优以及躲避未知障碍物。刘耀等^[16]运用 A* 算法初始化信息素,提高了蚁群算法的搜索效率,并用滚动窗口法进行碰撞预测,根据预测的碰撞位置采取不同的避障策略实现局部避障,但算法缺乏对动态障碍物的运动分析,避障效率差。封硕等^[17]通过加入对动态障碍物的运动信息来优化 DWA 算法的评价函数,实现了躲避动态障碍物,但算法并未解决在面对动态障碍物时会陷入局部最优的问题。

针对以上研究存在的问题,本文提出了一种基于 A* 的改进 DWA 算法。首先在 DWA 算法中引入碰撞锥理论,并针对静、动态障碍物进行碰撞分析,通过碰撞锥计算出具有碰撞威胁的速度。其次根据 DWA 算法的运动学约束筛选掉具有碰撞威胁的速度,同时改进 DWA 算法的各评价子函数,加入平滑度评价来改善局部避障路径,实现更高效的避障。最后将改进 DWA 算法与 A* 算法融合,使机器人按照全局最优路径点移动的同时实现更合理的动态避障。通过仿真对比实验显示,与传统 DWA 算法和文献[15]算法相比,本文算法在静、动态障碍环境中都表现出了更好的避障能力且避障路径更平滑,安全性更高。

1 传统 DWA 算法

DWA 算法是在速度空间中采样多组速度,并模拟机器人在这些速度下一段时间的轨迹,最后根据评价函数选取最优轨迹所对应的速度来驱动机器人运动,所以在模拟机器人运动前需要对其建立运动模型^[18]。假设机器人在单位时间 Δt 内的速度不变,则运动学模型为:

$$\begin{aligned} x(t+1) &= x(t) + v_t \cos\theta_t \Delta t \\ y(t+1) &= y(t) + v_t \sin\theta_t \Delta t \\ \theta(t+1) &= \theta(t) + \omega_t \Delta t \end{aligned} \quad (1)$$

其中, v_t 和 ω_t 的变化范围是根据当前机器人与周围障碍物与距离的大小和方位所决定。

得到运动模型便可以根据运动方程推算机器人的预测轨迹,因此需要采集机器人的多组不同的速度和角速度,来预测不同速度所对应轨迹,从而通过评价函数选择评分最高的预测轨迹。机器人的速度受其本身和障碍物距离的限制,所以要将机器人的速度限制在一定范围内:

$$V_m = \{v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (2)$$

机器人的速度受电机性能的影响,因此存在最大加速度限制,机器人轨迹在前向预测周期内存在基于当前速度的一个动态窗口 V_d ,该窗口内的速度是机器人实际所能达到的速度:

$$V_d = \{(v, \omega) \mid v \in [v_t - a_v \Delta t, v_t + a_v \Delta t] \cap \omega \in [\omega_t - a_\omega \Delta t, \omega_t + a_\omega \Delta t]\} \quad (3)$$

式中: v_t 和 ω_t 表示机器人 t 时刻的线速度和角速度, a_v 表示最大线加速度, a_ω 表示最大角加速度。

当出现障碍物时,为了使机器人可以在障碍物前及时停住,机器人可以保证安全的行进速度为:

$$V_s = \{(v, \omega) \mid v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot a_v} \cap \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot a_\omega}\} \quad (4)$$

式中: $\text{dist}(v, \omega)$ 为预测轨迹距离障碍物最近的距离。

依据上述 3 种约束条件,取 3 种速度约束的交集得到机器人最合适的动态窗口 V_r :

$$V_r = V_m \cap V_d \cap V_s \quad (5)$$

根据动态窗口可以采集若干速度组合,推演出若干可行轨迹,因此要通过评价函数评价出最优可行轨迹,得到最优速度^[19]。DWA 算法的评价函数为:

$$G(v, \omega) = \alpha \cdot \text{head}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \lambda \cdot \text{vel}(v, \omega) \quad (6)$$

其中, $\text{head}(v, \omega)$ 为航向角评价, $\text{dist}(v, \omega)$ 为障碍物距离评价, $\text{vel}(v, \omega)$ 为速度评价, α, β, λ 分别为各评价的权值。

2 改进 DWA 算法

2.1 障碍物处理

为了简化计算同时保证冗余,本文分别对机器人和静、动态障碍物进行圆形简化处理。并为了方便仿真分析,将机器人的底盘半径加入到障碍物中进行膨化处理。设 R_r 为机器人底盘半径, R_{ob} 为简化障碍物的半径,则静态障碍物的膨化半径为:

$$R_{r,ob} = R_r + R_{ob} \quad (7)$$

动态障碍物危险区域面积要大于静态障碍物,所以需要考虑速度威胁。假设机器人当前的移动速度为 v_r ,动态障碍物为匀速运动,移动速度为 v_o ,则机器人与障碍物之间的相对速度 $v_{ro} = v_r - v_o$ 。为了考虑障碍物的速度威胁,本文在原膨胀障碍物基础上引入威胁半径 $R_{th} = |v_{ro}|$ 来提高机器人对动态障碍物的识别范围,动态障碍物的膨

化半径 R_{rob} 为:

$$R_{rob} = R_r + R_{ob} + R_{th} \quad (8)$$

自此可以在分析中将机器人当作质点处理。

2.2 碰撞模型建立

机器人在规划的过程中可以依靠传感器获得自身和障碍物位置以及速度矢量等信息。如图1所示,设机器人AGV位置A为 (x_a, y_a) ,检测到的障碍物obs位置O为 (x_o, y_o) 。将机器人与膨胀后的障碍物两边做切线,切线间的区域为相对碰撞区域RCC, l_{r_o} 表示相对速度 v_{r_o} 的矢量射线, $\odot O$ 表示膨胀后的障碍物。当 l_{r_o} 与 $\odot O$ 相交,则表示机器人与障碍物在一定时间内会发生碰撞。相对碰撞区域集合RCC如式(9)所示。将RCC沿着 v_o 平移后得到的区域为绝对碰撞区域ACC如式(10)所示。

$$RCC = \{v_{r_o} \mid l_{r_o} \cap \odot O \neq \emptyset\} \quad (9)$$

$$ACC = RCC \oplus v_o \quad (10)$$

式中: \oplus 代表闵可夫斯基矢量和运算。

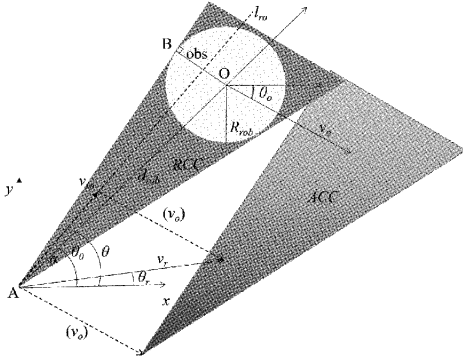


图1 碰撞锥模型

定义绝对碰撞区域内的速度集合为VO,则机器人与障碍物发生碰撞的速度集合VO为:

$$VO = \{v \mid v \in ACC\} \quad (11)$$

机器人与障碍物圆心连线为AO, d_{rob} 表示AO的距离, α 为AO与障碍物边界夹角, θ_r 为机器人当前位姿朝向角, θ_o 为动态障碍物的位姿朝向角, θ 为两者相对速度的朝向角, θ_0 为障碍物与机器人当前位置的夹角。则障碍物威胁可由 θ_0 、 θ 和 α 间的大小关系判断,当障碍物与机器人之间的距离大于危险距离时,机器人不需要避障。当障碍物与机器人之间的距离小于危险距离时,开始判断是否需要进行避障操作:当 $\theta_0 - \alpha < \theta < \theta_0 + \alpha$ 时,表示机器人有碰撞的危险,机器人应当选择其他速度方向行驶;当 $\theta < \theta_0 - \alpha$ 或 $\theta > \theta_0 + \alpha$ 时,表示机器人无碰撞危险,机器人可以按照当前速度方向行驶。若障碍物为静态,则 $v_o = 0$ 、 $\theta = \theta_r$,障碍物威胁判断同上。 d_{rob} 、 α 、 θ_0 和 θ 的的计算公式为:

$$d_{rob} = \sqrt{(x_o - x_a)^2 + (y_o - y_a)^2} \quad (12)$$

$$\alpha = \arcsin(R_{rob}/d_{rob}) \quad (13)$$

$$\theta_0 = \arctan \frac{y_o - y_a}{x_o - x_a} \quad (14)$$

$$\theta = \arctan \frac{v_{ry} - v_{oy}}{v_{rx} - v_{ox}} = \arctan \frac{v_r \sin \theta_r - v_o \sin \theta_o}{v_r \cos \theta_r - v_o \cos \theta_o} \quad (15)$$

2.3 碰撞范围的确定

机器人过早避障虽然可以提高避障安全性,但这也会使机器人过早偏离最优路径,因此设定一个合理的避障范围不但可以减少规划路径,而且可以优化算法的运算速率。如图2所示,A点和B点为机器人躲避障碍物过程的两个特殊点。A点表示机器人起始避障的点,B点表示机器人可以结束避障的点,C点表示避过障碍物点。

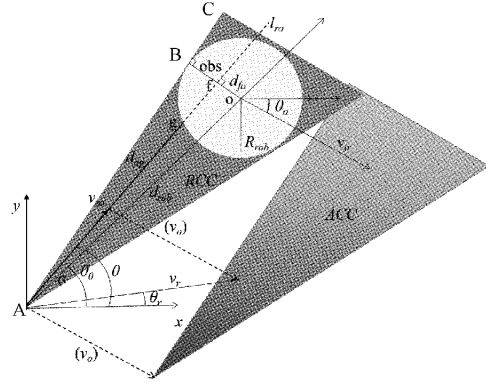


图2 避障过程分析

起始避障:设定最远危险距离。设 t_{ev} 为预测间隔时间,则危险距离为:

$$l_d = |v_r - v_o| t_{ev} \quad (16)$$

d_{ag} 表示为机器人速度方向与障碍物膨胀边缘的直线距离,用 d_{ag} 代替 d_{rob} 判断机器人与障碍物的距离更能准确反映机器人以当前速度到达障碍物的碰撞点。 d_{ag} 计算公式如(17)所示,定义条件如式(18)所示。

$$d_{ag} = d_{rob} \cos(\theta - \alpha) - \sqrt{R_{rob}^2 - (d_{rob} \sin(\theta - \alpha))^2} \quad (17)$$

$$d_{ag} = \begin{cases} d_{ag} & R_{rob} - (d_{rob} \sin(\theta - \alpha)) > 0 \\ \infty & R_{rob} - (d_{rob} \sin(\theta - \alpha)) < 0 \end{cases} \quad (18)$$

当机器人与障碍物的间距 $d_{ag} > l_d$ 时,机器人不执行避障操作,继续正常行进,当 $R_{rob} < d_{ag} \leq l_d$ 时,此时机器人与障碍物间距小于危险距离,判断机器人速度是否在碰撞区域ACC内,若是,则执行相应的避障操作;若不是,则按当前方向正常行进。

结束避障:当机器人运动过B点位置表示已经避过该障碍物。威胁解除满足的条件为:

$$|\theta - \alpha| > \theta \pm \frac{\pi}{2} \quad (19)$$

当满足条件后,将此障碍物从障碍物集合中删除,机器人结束对此障碍物避障判断,并朝向目标点方向行进。

通过碰撞锥的得到的碰撞速度合集VO为不可选的速度。剩下的可选速度在机器人的运动学约束条件下有部分

是不可达的。因此需要依靠 DWA 算法对机器人的当前速度进行离散采样,从而选取机器人可达的速度进行驱动。机器人在采样时间 Δt 内可达速度为式(3),可以得到机器人在采样时间 Δt 内可以转到的角度 θ_d 为:

$$\theta_d = \left\{ \theta \mid \theta \in \theta_i + \omega_i \Delta t - \frac{1}{2} a_w \Delta t^2, \theta_i + \omega_i \Delta t + \frac{1}{2} a_w \Delta t^2 \right\} \quad (20)$$

其中, θ_i 为机器人当前速度方向。由机器人运动学模型的限制,可知机器人下一时刻可达到的速度集合为式(5),通过采样可以得到不同方向的速度,剔除掉具有碰撞威胁的速度便得到安全的速度集合 V_s :

$$V_s = \{V \mid V \in V_r, V \notin VO\} \quad (21)$$

2.4 评价函数改进

本文针对动、静态障碍物的特点对原评价子函数进行改进,并增加平滑子函数来提高 DWA 算法的轨迹评估能力。

原子函数 $head(v, w)$ 表示模拟速度下的轨迹末端朝向与终点间的角度差。为了机器人在实时避障的同时保证全局最优,本文将全局最优路径点作为局部子目标点。将原 $head(v, w)$ 改为 $phhead(v, w)$ 表示模拟速度下的轨迹末端朝向与当前目标点间的角度差。

原子函数 $dist(v, w)$ 是预测轨迹末端与障碍物圆心的直线距离,该子函数并未考虑当前机器人的行进状态对避障的影响。而避障不但与障碍物的距离有关,还与机器人当前的行进方向有关,垂直距离更能直观影响机器人当前避障的程度,所以本文将原距离评价改进为相对垂直距离评价,即为机器人与障碍物间相对位置向量到相对速度的垂直距离 d_{fo} :

$$d_{fo} = d_{rob} \sin |\theta - \theta_0| \quad (22)$$

机器人在实施规划时会遇到在前一刻选择转向大的预测轨迹,下一时刻选择转向较小的预测轨迹,导致机器人频繁转向产生震荡,降低机器人的行进效率且规划出来的路径不平滑。预测轨迹如图 3 所示,为改善机器人避障轨迹的平滑性,本文增加了平滑子函数 $smo(v, w)$:

$$smooth = \frac{\pi}{2} - |\mu_2 - \mu_1| \quad (23)$$

其中, μ_1 和 μ_2 分别表示机器人起始和预测轨迹末端朝向角。预测轨迹转向角度越小,评分越高。

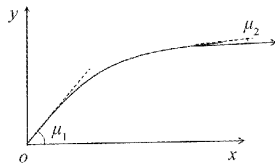


图 3 预测轨迹示意图

改进后的评价函数为:

$$G(v, w) = \alpha \cdot phhead(v, w) + \beta \cdot dist(v, w) + \lambda \cdot vel(v, w) + \tau \cdot smo(v, w) \quad (24)$$

2.5 融合算法

传统 DWA 算法作为局部规划算法,没有全局导航能力,在遇到特殊的“C”型障碍物时容易陷入局部最优,需要全局指引躲避特殊障碍物。因此本文采用 A* 算法进行全局路径规划,将全局最优路径点作为改进 DWA 算法的子目标点作为全局指引,使机器人实现局部避障的同时保证全局路径最优。

A* 算法一种是在 Dijkstra 算法基础上改进的启发式搜索算法,它是从起点开始,根据估价函数搜索周围节点,得到当前代价最低的节点作为子节点继续搜索,直至搜索到目标点。估价函数 $f(n)$ 为:

$$f(n) = g(n) + h(n) \quad (25)$$

其中, $g(n)$ 表示节点 n 与起点的实际代价, $h(n)$ 表示节点 n 与终点的估计代价。

但 A* 算法得到的路径节点存在很多冗余点,容易使机器人转向频繁影响移动效率。因此需要对得到的路径点进行二次删减,具体步骤如下:

1) 设路径起点为 A, 后继节点为 B 和 C。检测相邻的节点 A、B、C 是否共线;若是,则去掉 B 节点;若不是,则保留 B 节点,以此遍历所有节点。

2) 对剩余的节点 AC 间进行障碍物判断:若存在障碍物,则保留中间节点 B;若不存在障碍物,则剔除中间节点 B,最终得到全局最优路径点。

综上所述,改进 DWA 算法的步骤为:首先用改进 A* 算法得到全局最优路径,在全局最优路径的基础上对冗余路径点进行删减,得到最优路径点并作为全局目标指引点。然后利用机器人自身的传感器实时检测最近障碍物是否到达危险区域,并计算碰撞锥威胁区域内的速度和角度范围,筛选出没有碰撞威胁的速度,最后利用改进 DWA 算法的评价函数选择最优预测路径对应的速度驱动机器人依据全局最优路径点进行导航移动。规划流程图如图 4 所示。

3 仿真与分析

实验采用 MATLAB R2016 软件进行仿真。为了验证本文算法在复杂环境下路径规划的可行性和优越性,本文分别构建环境复杂的静态障碍物地图和动态障碍物地图,并对 DWA 算法、文献[17]算法和本文改进 DWA 算法进行仿真对比实验。最后通过复杂动态环境地图进一步验证本文算法的有效性。关于机器人的自身参数、算法参数、初始评价函数参数如表 1~3 所示。

3.1 静态障碍环境实验

为了验证本文算法的静态避障能力,实验在 14×14 的地图上建立地图。起点为 (3,3), 终点为 (13,13), 机器人起始航向角为 $\frac{\pi}{4}$, 起始前进速度为 0 m/s, 起始角速度为 0 rad/s, 圆圈代表膨胀半径为 0.5 m 的障碍物。

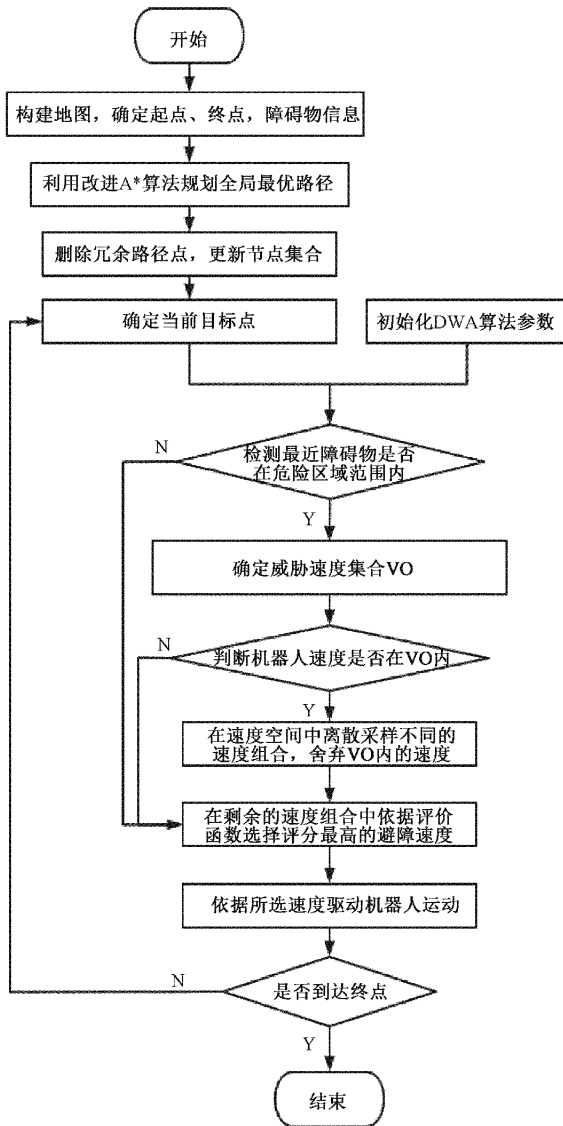


图4 融合算法流程

由图5可知,图5(a)为传统DWA算法仿真规划的路径。图5(b)为文献[17]算法规划的路径。图5(c)为本文算法仿真规划的路径,虚线为本文改进A*算法规划的全局最优路径。图5(d)为不同算法在规划过程中与最近障碍物的安全距离对比图。在静态环境中3种算法都可以规划出最终路径,但传统DWA算法在遇到密集障碍物区时容易绕行障碍物,使规划的总路径变长。且在躲避坐标为(5,5)障碍物时陷入局部最优,安全性大大降低。文献[17]算法虽然在评价函数中加入障碍物运动信息,但在面对静态障碍物依靠的是传统DWA算法的避障性能,算法在检测到多个障碍物时仍不能较快地合理避障,转折点多,移动效率差,且规划出来的路径不平滑。而本文算法可以在检测到避障范围内的障碍物之初快速删减掉有碰撞风险的速度,大大提高了算法的避障效率,使机器人不但可以按照全局指引顺利穿过密集障碍物区,而且行进路径遵循全局最优,避障轨迹更加平滑。由表4显示,本文算法比传统DWA算法在路径长度、行进步数方面各缩短20.3%、49.5%,平均速度和最小安全距离方面各提高61.2%、1400%;比文献[15]算法在路径长度和行进步数方面各缩短2.1%、36.2%,在平均速度和最小安全距离方面各提高66%、66%。

经上述分析,本文算法规划的路径比传统DWA算法和文献[17]算法更短更接近全局最优。且在密集障碍物区的移动效率有大幅提升、安全性更高。体现了本文算法跟踪能力和避障能力的优越性。

3.2 动态障碍环境实验

为了验证本文算法的动态避障能力,本文选择危险程度最高的相向动态避障情景,并对DWA算法、文献[17]算法以及本文算法进行对比实验。实验在20×20的地图上分别设置3种不同速度的动态障碍物进行分析:设机器人起点为(1,8),终点为(16,11),初始速度为0 m/s,初始方向为0,障碍物起点为(13,9)。速度方向为-π。障碍物移速

表1 机器人自身参数

最高前进速度/ (m·s ⁻¹)	最大角速度/ (rad·s ⁻¹)	最低前进速度/ (m·s ⁻¹)	最小角速度/ (rad·s ⁻¹)	前进加速度/ (m·s ⁻²)	角加速度/ (rad·s ⁻²)
1	50	0	-50	0.2	20

表2 算法参数

速度分辨率/ (m·s ⁻¹)	采样时间 间隔/s	预测轨迹采样 间隔/s	转速分辨率/ (rad·s ⁻¹)
0.01	0.1	3	1

表3 初始评价函数参数

α	β	λ	τ
0.1	0.2	0.1	0.1

v_o 分别为0.1、0.2、0.4 m/s。 t_1 为机器人躲避动态障碍物的时刻。黑色虚线为障碍物的行进轨迹。

由图6可知,在图6a中,障碍物移速为0.1 m/s,3种算法都完美避过了障碍物。而在图6b中,障碍物移速提高到0.2 m/s,DWA算法下机器人在躲避动态障碍物时进入了威胁区域,文献[17]算法在评价函数中加入了动态障碍物运动信息来影响预选速度的选择,因此会比DWA算法更快执行避障操作。而本文算法利用相对速度 v_{ro} 扩大了威胁区域且优化动态窗口后会比文献[17]更早做出判断,仍然可以使机器人保持在威胁区域外安全避障。在

表 4 算法数据对比

地图	算法	路径长度/m	行进步数	平均速度/(m·s ⁻¹)	最小安全距离/m
环境 1	DWA 算法	18.04	571	0.31	0.03
	文献[17]算法	14.67	452	0.30	0.27
	本文算法	14.36	288	0.50	0.45

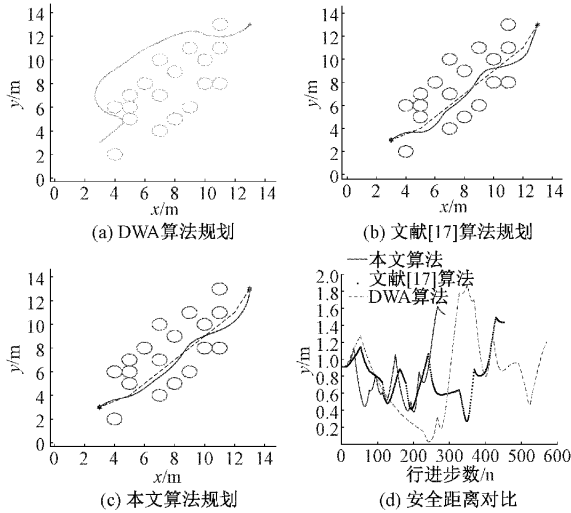


图 5 静态障碍环境路径规划仿真

图 6c 中,障碍物移速提高到 0.4 m/s, DWA 算法下机器人在检测到动态障碍物后不能及时做出避障反应而与障碍物相撞,文献[17]算法在 t_1 时刻遇到了障碍物在机器人预选速度方向和终点方向同向的连线上而陷入了短暂的局部最优,致使机器人的速度下降,并与动态障碍物“擦肩而过”,大大降低了行进安全性。而本文算法凭借对动态窗口优化提早筛选掉了与终点方向同向且存在障碍物的速度,避免了机器人陷入局部最优,且可以成功避开移速较快障碍物。如表 5 数据可见,在动态障碍环境中,传统 DWA 算法由于没有设置威胁半径,在面对移速较低的障碍物可以成功躲避,但随着障碍物移速提高,算法会因反应距离过短而与障碍物发生碰撞。文献[17]算法虽然比 DWA 算法避障效果好,但当机器人躲避速度较快的障碍物时仍然有可能陷入局部最优影响移动效率,降低安全性。而本文算法在遇到不同速度的障碍物时可以自行调节威胁半径并在预选时迅速筛掉危险速度进行避障,消除了文献[17]算法存在的局部最优问题,避障效果明显且最小安全距离都在 1 m 以上,可见本文算法更能广泛适用于动态障碍环境。

3.3 复杂动态环境实验

为了验证本文算法在复杂动态环境的避障能力,本文建立 30×30 的静态环境地图,起点为(1,1),终点为(30,30),并用改进 A* 算法规划出全局最优路径,如图 7 黑线所示。之后在此地图上逐步随机添加未知和动态障碍物来增加环境的复杂程度,机器人沿着全局规划的路径移动并实时

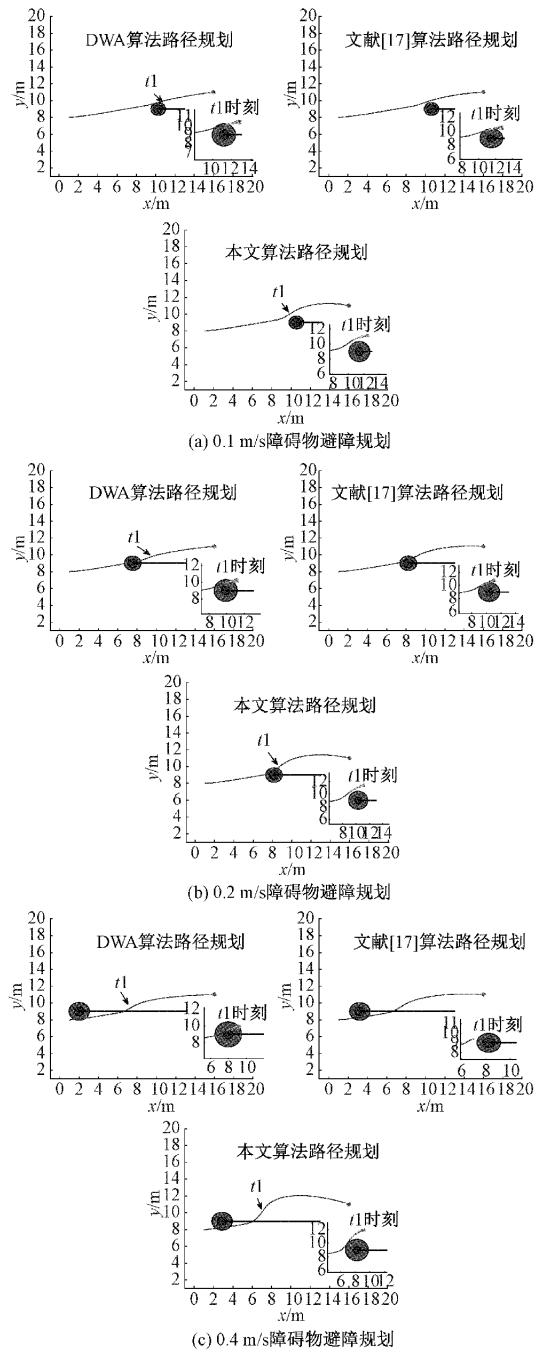


图 6 动态障碍环境的避障规划对比

检测周围的障碍物执行避障,实验结果如图 8~11 所示。

图 8 为本文算法在静态环境下的规划路径。由图 8 可知,在只有静态障碍物的情况下,本文算法表现出了较好

表5 算法数据对比

Obs 速度/ (m·s ⁻¹)	算法	路径 长度/ m	控制 节点数/ n	最小 安全 距离/m
0.1	DWA 算法	14.84	274	0.541
	文献[17]算法	14.85	238	0.732
	本文算法	15.09	243	1.217
0.2	DWA 算法	14.89	275	0.373
	文献[17]算法	14.97	241	0.782
	本文算法	15.25	246	1.263
0.4	DWA 算法	14.97	275	-0.042
	文献[17]算法	14.88	302	0.031
	本文算法	15.98	254	1.116

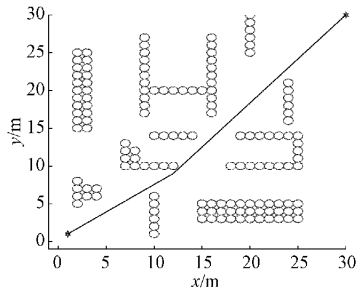


图7 改进A*算法全局路径规划

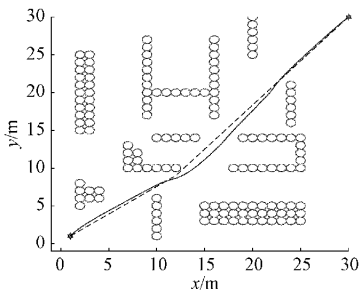


图8 环境1机器人静态环境路径规划

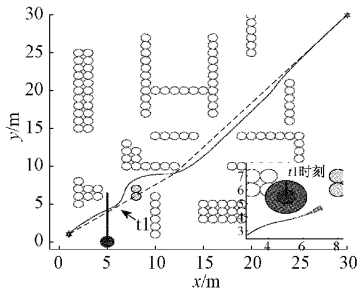


图9 环境2加入移速为0.1 m/s的动态障碍物

的跟踪能力,规划的路径与全局最优路径相近。图9~11中逐次加入了移速为0.1、0.4、0.2 m/s的动态障碍物和未知障碍物,其中实心圆圈为随机添加的障碍物,t1~t3分别为机器人躲避动态障碍物的时刻。可以看到本文算法

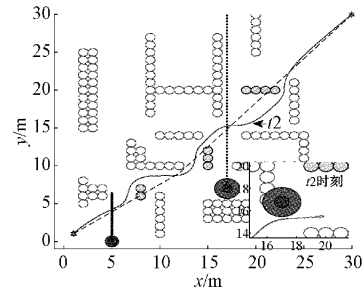


图10 环境3加入移速为0.4 m/s的动态障碍物

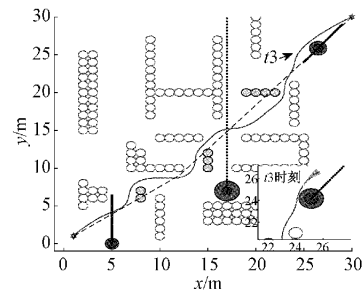


图11 环境4加入移速为0.2 m/s的动态障碍物

在规划的过程中能够有效避开随机添加的未知障碍物和不同移速的动态障碍物,且在避障之后可以快速跟踪最优路径,证明了该方法的有效性。

4 结 论

传统DWA算法没有全局导航能力,容易陷入局部最优导致规划失败,且动态避障性差。针对以上问题,本文提出了一种基于A*的改进DWA算法,首先在DWA算法中基于碰撞锥理论对静、动态障碍物进行防碰撞分析,筛选掉具有碰撞威胁的速度。其次对DWA算法的评价子函数进行改进,并用改进后的评价函数对剩余速度的模拟轨迹进行评价得到最优速度,最后用A*算法的全局最优路径点作为DWA算法的目标点指引,解决DWA算法易陷入局部最优的问题。经过实验验证,在复杂的静态障碍环境和不同速度的动态障碍环境下,本文算法不但可以解决局部最优问题,而且在避障效率和安全性上比其他同类算法都有较为明显的优势,能完美适应复杂动态环境,鲁棒性更强。

参考文献

- [1] 李晓旭,马兴录,王先鹏. 移动机器人路径规划算法综述[J]. 计算机测量与控制, 2022, 30(7): 9-19.
- [2] 翟文鹏,刘润南,朱承元. 基于改进Dijkstra算法的滑行路径优化[J]. 中国民航大学学报, 2022, 40(1): 1-6.
- [3] 黄令苇,全燕鸣,王荣辉. 基于安全A*算法的AGV路径规划[J]. 自动化与仪表, 2021, 36(1): 45-48.
- [4] 陈豪,李勇,罗靖迪. 基于改进A*算法优化的移动机器人路径规划研究[J]. 自动化与仪器仪表, 2018(12): 1-4.

- [5] 王刚,张方,严大亮,等.基于改进蚁群算法的机器人三维路径规划[J].国外电子测量技术,2020,39(11):1-6.
- [6] 杨立炜,付丽霞,王倩,等.多层优化蚁群算法的机器人路径规划研究[J].电子测量与仪器学报,2021,35(9):10-18.
- [7] 张贺,缪存孝,唐友军,等.移动机器人自主动态避障方法[J].北京航空航天大学学报,2022,48(6):1013-1021.
- [8] 林依凡,陈彦杰,何炳蔚,等.无碰撞检测 RRT* 的移动机器人运动规划方法[J].仪器仪表学报,2020,41(10):257-267.
- [9] 严浙平,黄俊儒,吴迪.基于 RRT 和 DWA 的欠驱动 UUV 路径规划[J].数字海洋与水下攻防,2020,3(3):258-264.
- [10] 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.
- [11] 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.
- [12] 常新新,胡为,姬书得,等.基于改进动态窗口法的移动机器人避障研究[J].组合机床与自动化加工技术,2021(7):33-36,39.
- [13] 戴凯龙,贾子彦,潘玲佼.基于改进动态窗口法的无人
- 机避障航迹规划[J].智能计算机与应用,2022,12(7):114-117.
- [14] 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.
- [15] 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.
- [16] 刘耀,毛剑琳.动态环境下基于改进蚁群算法的路径规划研究[J].电子测量技术,2020,43(7):82-87.
- [17] 封硕,吉现友,程博,等.融合动态障碍物运动信息的路径规划算法[J].计算机工程与应用,2022,58(21):279-285.
- [18] 李苑.室内移动机器人控制与路径规划研究[D].哈尔滨:哈尔滨工程大学,2020.
- [19] 田永永,李梁华.基于速度方向判定的动态窗口法[J].农业装备与车辆工程,2018,56(8):39-42.

作者简介

邓云峰(通信作者),硕士研究生,主要研究方向为自主导航及智能控制。

E-mail:dyz1970109@163.com

黄翼虎,博士,副教授,主要研究方向为化工自动化仪表、数字信号处理、检测理论及应用、路径规划等。

E-mail:hyhuizi@163.com