

张望, 张载龙, 闫焕文. 机器人穿越狭窄通道的局部路径规划算法[J]. 智能计算机与应用, 2025, 15(7): 29-36. DOI: 10.20169/j. issn. 2095-2163. 250704

机器人穿越狭窄通道的局部路径规划算法

张望, 张载龙, 闫焕文

(南京邮电大学 物联网学院, 南京 210003)

摘要: 针对移动机器人在穿越狭窄通道时的局部路径规划效率低、易陷入局部最优、抵达障碍物附近的目标时可能出现目标不可达等问题, 本文提出一种改进的权值自适应动态窗口法。根据机器人与周围障碍物的距离和正前方一定范围内障碍物密集程度, 自适应调整算法目标函数中权值; 在接近目标时, 根据目标范围内的障碍物的分布进一步调整权值并引入角速度评价因子精确控制方向。仿真实验结果表明: 采用改进的自适应动态窗口法, 机器人可以安全高效地穿越狭窄通道, 抵达终点的时间相比自适应动态窗口法有一定缩短, 终点靠近障碍物, 也能高效精准抵达终点。

关键词: 动态窗口法; 权值自适应; 局部路径规划; 角速度评价因子

中图分类号: TP242

文献标志码: A

文章编号: 2095-2163(2025)07-0029-08

Local path planning algorithm for robots traversing narrow channels

ZHANG Wang, ZHANG Zailong, YAN Huanwen

(School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: To address the issues of low efficiency in local path planning and easy entrapment in local optima when mobile robots traverse narrow passages, as well as the possibility of unreachable targets when approaching obstacles, an improved adaptive dynamic window approach with self-adjusting weights is proposed. The weights in the algorithm's objective function are adaptively adjusted based on the distance between the robot and surrounding obstacles and the density of obstacles within a certain range directly ahead. As the robot approaches the target, the weights are further adjusted according to the distribution of obstacles within the target range, and an angular velocity evaluation factor is introduced to precisely control the direction. Simulation results show that with the improved adaptive dynamic window approach, the robot can safely and efficiently traverse narrow passages, and the time to reach the destination is significantly reduced compared to the adaptive dynamic window approach. Even when the destination is close to an obstacle, the robot can still reach the destination efficiently and accurately.

Key words: dynamic window method; weights adaptive; local path planning; angular velocity evaluation factor

0 引言

在移动机器人中, 局部路径规划是实现自主导航和避障的关键技术之一^[1]。局部路径规划是根据周围环境信息和机器人自身状态, 确定一条安全、较短的路径, 使机器人能够从当前位置移动到临时目的地, 并避免与障碍物发生碰撞^[2]。

局部路径规划面临着许多挑战。机器人所处的环境通常是复杂、动态变化的, 包括静态障碍物和移动障碍物^[3]。因此, 路径规划算法需要在不同的环境

中能够快速适应, 并实时感知和响应障碍物的变化。

近年来, 不少研究者致力于局部路径规划算法的研究。Hossain T 等^[4]提出了一种结合动态窗口法和改进的跟随间隙法的新型局部路径规划算法, 旨在为移动机器人生成无碰撞轨迹, 该算法考虑了机器人与障碍物的相对位置, 确保保持安全距离, 并通过寻找可行的间隙引导机器人朝目标前进, 以实现安全的轨迹生成; Lai X 等^[5]提出了一种增强的动态窗口法, 使用距离函数作为角度评价函数的权重调整函数, 同时提出了一种新的评价函数优化机

基金项目: 江苏省研究生科研与实践创新计划 (KYCX22_1030)。

作者简介: 张望 (1998—), 男, 硕士, 主要研究方向: 机器人路径规划算法; 闫焕文 (1999—), 男, 硕士, 主要研究方向: 智能优化算法。

通信作者: 张载龙 (1966—), 男, 博士, 研究员, 主要研究方向: 边缘计算, 人工智能。Email: ZLZHANG@njupt.edu.cn。

收稿日期: 2023-12-04

哈尔滨工业大学主办 ◆ 学术研究与应用

机器人的方向角来减小机器人临近目的地时的抖动程度,提高了局部路径规划的性能和适应性;赵青等^[6]提出了一种融合行人预测信息的动态窗口算法,考虑了机器人与行人的相对位置和速度等先验信息,并引入了组合椭圆行人区域的创新概念,该区域反映了机器人和行人速度的动态变化,通过扩展经典动态窗口法的评分函数,实现了智能避让,旨在最大程度减少机器人移动对行人原始行动意图的干扰;Kobayashi M 等^[7]提出了一种融合动态窗口方法和虚拟机械手的新型局部路径规划算法,虚拟机械手利用环境信息及其运动生成反射式速度,而动态窗口法则通过考虑虚拟机械手的输出来调整机器人速度,基于对静态和动态障碍物未来位置的预测生成备选路径;Liu L 等^[8]提出了一种融合算法,将跳点搜索法与经过优化的 A* 算法结合,并与动态窗口方法相融合,用于路径规划,通过跳点搜索法和新的基于曼哈顿距离和欧氏距离定义的距离评估函数改进 A* 算法,实现更有效的全局路径规划,以动态窗口方法为核心,将全局路径信息与安全性要求相结合,生成高度平滑的全局最优路径;王永雄等^[9]提出一种线速度权值自适应的动态窗口法,根据机器人与障碍物的距离和障碍物的密集程度自适应调整线速度权值,一定程度改善了机器人穿越狭窄过道的性能。但是现有的局部路径规划算法依然存在以下问题:

1) 机器人在经过狭窄过道时,可能会绕路或陷入局部最优解;

2) 目标点接近障碍物时,机器人可能受障碍物过度影响,导致陷入局部最优解,从而不能抵达目标点。

在复杂环境中,局部路径规划不仅要规划出一条避开障碍物的路径,还要与障碍物保持一定的安全距离,防止意外发生^[10]。为了提高局部路径规划的准确性,本文针对上述存在的问题提出一种权值自适应动态窗口法,对自适应动态窗口法作出以下改进:

首先,利用机器人自身的传感器如激光雷达、深度相机等,快速获取机器人一定范围内障碍物的密集信息,利用机器人控制邻域的自适应优化思想,对动态窗口法评价函数中权重进行自适应调整。

其次,当感知到机器人离目标点较近时,利用传感器信息获取到目标点附近的障碍物信息,如果检测到目标点靠近障碍物时,在评价函数中加入角速度评价函数,并根据不同场景下的局部信息对权重系数进一步自适应调整。

上述改进可改进动态窗口法穿越狭窄过道和抵

达接近障碍物的目标点的性能,能有效提高算法的效率、安全性和稳定性。

1 自适应动态窗口法

动态窗口法 (Dynamic Window Approach, DWA) 通过定义受机器人环境和速度特性约束的线速度和角速度空间,进行采样并评估局部路径,从而选择最优局部路径对应的线速度和角速度。速度空间受到机器人环境和动力性能的约束,动作选择则考虑当前局部环境的约束。

根据机器人自身动力性能限制,线速度和角速度组合 V_m , 公式如下:

$$V_m = \{(v, w) \mid v \in [v_{\min}, v_{\max}], w \in [w_{\min}, w_{\max}]\} \quad (1)$$

其中, v 表示线速度, w 表示角速度。

在移动机器人动态窗口移动时间间隔内,机器人的速度受到电机的加速和减速性能限制,此时的速度约束如下式:

$$V_t = \{(v, w) \mid v \in [v_c - \dot{v} \times \Delta t, v_c + \dot{v} \times \Delta t], w \in [w_c - \dot{w} \times \Delta t, w_c + \dot{w} \times \Delta t]\} \quad (2)$$

其中, v_c 、 w_c 是当前机器人的线速度和角速度; $\dot{v}$ 是线加速度; $\dot{w}$ 是角加速度。

在动态窗口法中,每次选择最优的线速度和角速度形成的局部路径可能包含多个小段圆弧或线段。为确保在这小段路径内的运动安全,算法对速度空间进行了进一步约束,该约束如下式:

$$V_a = \{(v, w) \mid v \in [v_{\min}, \sqrt{2 \times \text{dist}(v, w) \times v}], w \in [w_{\min}, \sqrt{2 \times \text{dist}(v, w) \times w}]\} \quad (3)$$

最终的速度空间 $V_r(v, w)$:

$$V_r(v, w) = V_m \cap V_t \cap V_a \quad (4)$$

自适应动态窗口法用局部环境信息自适应地调整评价函数中的速度权值,以解决动态窗口法在穿越密集障碍物时路径不合理、速度和安全性无法兼顾的问题。具体调整策略公式如下:

$$\gamma_d = \begin{cases} \gamma_{\min} + k(\gamma_{\max} - \gamma_{\min}) \left(\frac{D_{\min}}{D_s} \right)^a, & D_{\min} \leq D_s \\ \gamma_{\max}, & D_{\min} > D_s \end{cases} \quad (5)$$

其中, $D_{\min}$ 表示机器人离最近的障碍物的距离; a 表示指数; $\gamma_{\min}$ 是通过狭窄通道最安全的线速度权值; $\gamma_{\max}$ 是以最短时间通过狭窄通道的线速度权值。

为防止生成的线速度权值过大或者过小,设置的线速度权值为 $[\gamma_{\min}, \gamma_{\max}]$ 。

自适应动态窗口法通过评价函数对从速度空间

采样到的线速度和角速度进行评价,选出最优的速度组合。自适应动态窗口法评价函数定义如下:

$$G(v, w) = \alpha \times \text{heading}(v, w) + \beta \times \text{dist}(v, w) + \gamma_d \times \text{vel}(v, w) \quad (6)$$

其中, $\text{heading}(v, w)$ 是机器人方向角评价子函数,返回值是按采样线速度和角速度运动指定时间后机器人的方向角与机器人和目标点方向的差值; $\text{dist}(v, w)$ 是障碍物距离评价子函数,返回值是按采样速度组合运动指定时间的路径末端位置与障碍物的最小距离; $\text{vel}(v, w)$ 是线速度评价子函数,返回值是当前局部路径对应的线速度; α, β, γ_d 分别为方向角评价子函数、障碍物距离评价子函数、线速度评价子函数的权值参数。

为了避免某一项的影响权重过大,算法对于上述3项子函数的返回值一般会进行归一化处理。

自适应动态窗口法在一定程度上改善了机器人穿越狭窄通道时绕路的问题,但是仅自适应调整线速度权值,在穿越狭窄通道时可能偏重于选出线速度较优的局部路径,从而忽略线速度、方向和障碍物距离综合上更优的局部路径,陷入局部最优。

2 改进自适应动态窗口法

DWA 算法在机器人行驶过狭窄过道时,线速度评价子函数权值设置较大可能导致过高速,降低安全性,而设置较小则可能导致低效率。对方向角评价子函数,较大权值可能使机器人陷入局部最小值,而较小权值可能导致路径过于偏重安全和速度。对障碍物距离评价子函数,较大权值可能使机器人陷入局部最小值或绕路,而较小权值可能使机器人过度靠近障碍物,降低安全性。在接近目标点时,较大的线速度评价子函数权值可能导致机器人驶过目标点,而较大的方向角权值可能使机器人忽略前方障碍物。适当权衡这些权值是关键,以平衡安全性、效率,准确导航。

基于以上分析,结合机器人局部环境信息,设计出 DWA 算法自适应权值调整的策略是很有必要的。本文融合了王永雄等^[9]提出的稠密障碍物判断方法并自创接近障碍物目标点几种常见情形的判断策略,提出了一种改进的自适应动态窗口法。

2.1 基于局部环境信息的自适应权值调整

现有移动机器人大多数都携带了能够测距的传感器如激光雷达、深度相机等,可以探测一定范围内的环境信息,本文假设机器人携带了激光雷达,机器人可以检测 360° 方向的障碍物分布情况,一般来说,激光雷达在 0.15~10 m 之间的测量范围内的测

量准确度是较为可靠的。本文选取机器人运行方向左右 90° 即机器人前方 180°, 半径为 1 m 的扇形区域作为权值调整参考的环境信息,如图 1 所示。

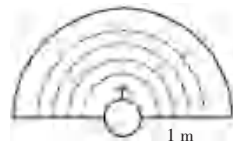


图1 选取传感器检测的局部信息示意图

Fig. 1 Schematic representation of the local information detected by the selected sensors

假设雷达探测到的扇形范围内,存在 C 个障碍物,第 i 个障碍物与机器人的距离定义为 d_i ,方位角为 θ_i 。当 C 大于等于一个设定的阈值(通常为 2),则该区域被认为是一个障碍密集区。当前最近范围内两两障碍物之间的间距 gap_{ij} ,如下式所示:

$$gap_{ij} = \begin{cases} \sqrt{d_i^2 + d_j^2 - 2 \times d_i \times d_j \times \cos(\theta_{ij})}, & \text{if } \theta_{ij} \geq 1 \\ |d_i - d_j|, & \text{if } \theta_{ij} < 1 \end{cases} \quad (7)$$

其中, θ_{ij} 是机器人与第 i 个障碍物和第 j 个障碍物连线的夹角。

当机器人当前所在位置离目标点的距离小于设定的阈值时,计算机器人前方 120°、半径为 dr 的扇形范围内障碍物与目标点的相对分布情况,根据雷达等传感器信息得到第 i 个障碍物与机器人距离为 d_i 和方位角 θ_i ,根据机器人自身定位信息和目标点坐标信息可以计算出机器人与目标点的距离 d_g 和相对机器人的方位角 θ_g ,可以得到机器人和目标点连线与与第 i 个障碍物连线之间的夹角 θ_{gi} ,使用式(7)计算出目标点与扇形范围内障碍物的距离 d_{og} ,保存离目标点小于机器人直径(如果是方形机器人则保存小于机器人最大边长)的障碍物信息(θ_{gi} , d_{og} , d_i , d_g),利用保存的障碍物信息判断障碍物靠近目标点的情况,本文讨论以下几种常见情形:

(1) 目标点在障碍物前方:如果机器人与目标点连线和机器人与第 i 个障碍物连线之间的夹角 θ_{gi} 小于 5° 且机器人到该障碍物的距离大于机器人到目标点的距离,则对于当前位置下的机器人,目标点接近障碍物且在障碍物前方;

(2) 目标点在障碍物后方:如果机器人与目标点连线和机器人与第 i 个障碍物连线之间的夹角 θ_{gi} 小于 5° 且机器人到该障碍物的距离小于机器人到目标点的距离,则对于当前位置下的机器人,目标点接近障碍物且在障碍物后方;

(3) 目标点在障碍物左或右方:如果机器人与

目标点连线和机器人与第 i 个障碍物连线之间的夹角 θ_{gi} 大于 5° , 则对于当前位置下的机器人, 目标点接近障碍物且在障碍物左方或右方。

为了更好地估计机器人遇到特殊场景下的转向能力, 最优地控制机器人的运动, 本文在原动态窗口法的基础上加入角速度评价子函数, 角速度评价子函数定义如下:

$$rvel(v, w) = \begin{cases} \frac{rvel(i)}{\sum_{i=1}^n rvel(i)}, & \text{if 上一次最优角速度是正数} \\ \frac{-rvel(i)}{\sum_{i=1}^n rvel(i)}, & \text{if 上一次最优角速度是负数} \end{cases} \quad (8)$$

其中, $rvel(i)$ 是第 i 组角速度, 设置该评价子函数的权值为 δ 。

在接近目标时, 受目标周围的障碍物影响, 如果采用原目标函数计算局部最优路径可能形成局部最小环路 (即局部最小点形成一条环路), 从而使机器人陷入局部最优, 加入角速度评价函数, 产生目标方向的“引力”, 原理如图 2 所示, 避免形成局部最小环路。



图 2 目标“引力”原理图

Fig. 2 Schematic diagram of target “gravity”

$$\alpha_d = \begin{cases} \alpha_{\max} - (\alpha_{\max} - \alpha_{\min}) \times (1.0 - d_{\min}), & \text{if } (d_{\min} \leq 1.0) \cap (\theta_{\min} < 5.0) \\ \alpha_d + (\alpha_{\max} - \alpha_d) \times d_{\min}, & \text{if } (d_{\min} \leq 1.0) \cap (\theta_{\min} \geq 5.0) \\ \alpha_{\max}, & \text{if } (d_{\min} > 1.0) \end{cases} \quad (10)$$

其中, $\alpha_{\min}$ 是经典动态窗口法能够保证机器人安全穿过狭窄过道的最小方向权值, $\alpha_{\max}$ 是保证机器人安全穿过笔直狭窄过道且行驶的轨迹最平滑的最大方向权值, 控制方向权值在 $[\alpha_{\min}, \alpha_{\max}]$ 范围内, $\theta_{\min}$ 是机器人与前方最近障碍物的角度差, $d_{\min}$ 是机器人与前方最近障碍物的距离。

当机器人前方扇形区域内出现障碍物, 机器人障碍物距离评价函数权值应该提高, 保持路径与障碍物的安全距离, 如果机器人前方扇形区域无障碍物时, 此时局部路径应减少对障碍物距离的考量, 使机器人保持趋向终点的方向运动, 此时权值取最小值, 因此定义障碍物距离评价函数的自适应动态权

动态窗口法的评价函数没有涉及机器人转向的精准性, 为了精准控制机器人的转向, 在评价函数加入角速度评价子函数, 本文自适应动态窗口法的评价函数如下式:

$$G(v, w) = \alpha_d \times heading(v, w) + \beta_d \times dist(v, w) + \gamma_d \times vel(v, w) + \delta \times rvel(v, w) \quad (9)$$

其中, $\alpha_d, \beta_d, \gamma_d$ 是根据局部信息动态调整的方向角评价子函数、障碍物距离评价子函数、线速度评价子函数的权值, δ 是角速度评价子函数的权值。

为了避免机器人非必要转向, 在机器人与目标点的距离未小于阈值时, δ 设置为 0; 在机器人与目标点的距离小于阈值后, 根据局部信息进行设置。

机器人执行从起点到终点的运动任务时, 安全是一个非常重要的指标, 因此只有障碍物间距 gap_{ij} 满足大于 2 倍当前机器人半径加膨胀设定的安全半径 $safe_r$ (设定为 0.1 m) 时, 才能判断机器人可以从障碍物中间安全穿越。为机器人可以在狭窄过道平滑安全地穿行, 分别设计方向角评价子函数、障碍物距离评价子函数、线速度评价子函数的权值参数调整策略。

当机器人正前方不远处出现障碍物, 机器人方向评价函数权值应该下降, 如果机器人非正前方附近出现障碍物时, 机器人应该根据障碍物远近程度调整方向评价函数的权值, 如果机器人附近无障碍物时, 机器人尽可能保持趋向终点的方向运动, 此时权值应取最大值, 因此, 方向评价函数的自适应动态权值 α_d 如下式:

值 β_d 如下式:

$$\beta_d = \begin{cases} \beta_{\min} + (\beta_{\max} - \beta_{\min}) \times (e^{c+c_{\max}} \div e), & \text{if } c \geq 1 \\ \beta_{\min}, & \text{if } c = 0 \end{cases} \quad (11)$$

其中, $\beta_{\min}$ 是经典动态窗口法能够保证机器人安全穿过狭隘过道的最小障碍物距离权值, $\beta_{\max}$ 是保证机器人能够安全不停滞通过狭隘过道的最大障碍物距离权值, 控制 β_d 在 $[\beta_{\min}, \beta_{\max}]$ 范围内; c 是机器人正前方扇形区域实时检测到的障碍物数量; $c_{\max}$ 是设定的检测的扇形范围内障碍物最大数量。

当机器人离目标点距离小于阈值, 该阈值视环境复杂程度设定, 通过传感器采集到的信息, 计算与

机器人和终点连线的夹角小于 60° 的障碍物信息, 如果障碍物位置与目标点位置小于设定阈值 (设定为机器人直径或机器人最大边长), 则保留信息, 根据标准判断情形, 这几种情形下的评价函数和评价函数权值调整:

(1) 目标点在障碍物前方: 如果方向角评价函数的权值设置较小, 障碍物距离评价子函数权值设置过大时, 机器人可以受到目标点处障碍物的影响, 导致方向偏离或者停滞, 如果角速度评价子函数权值设置不为 $\delta_{\max}$ 时, 可能会导致机器人在原地自旋, 综合考虑并经过多次测试, 设置 α_d 、 β_d 、 δ 为 $\alpha_{\max}$ 、 $\beta_{\min}$ 、 $\delta_{\max}$ 较好;

(2) 目标点在障碍物后方: 如果方向角评价函数的权值设置过大, 障碍物距离评价子函数权值设置过大时, 机器人可以受到目标点处障碍物的影响, 导致方向偏离或者停滞, 如果角速度评价子函数权值设置为 $\delta_{\min}$ 时, 可能会导致机器人转向不精确, 综合考虑并经过多次测试, 设置 α_d 、 β_d 、 δ 为 $\alpha_{\min}$ 、 $\beta_{\min}$ 、 $0.66 \times \delta_{\max}$ 较好;

(3) 目标点在障碍物左或右方: 如果方向角评价函数的权值设置过小, 障碍物距离评价子函数权值设置过大时, 机器人可能受到目标点处障碍物的影响, 导致方向偏离或者陷入局部最优, 如果角速度评价子函数权值设置为 $\delta_{\max}$ 时, 可能会导致机器人转向幅度过小, 综合考虑并经过多次测试, 设置 α_d 、 β_d 、 δ 为 $\alpha_{\max}$ 、 $\beta_{\min}$ 、 $\delta_{\max}$ 较好。

2.2 改进自适应动态窗口法流程

Step 1 根据传感器信息计算得到机器人前方一定范围内的局部信息, 判断是否进入狭窄过道, 计算最近的两两障碍物之间的间隔 gap, 判断机器是

否能够穿行;

Step 2 由 Step 1 获取到的局部信息, 包括障碍物距离、数量和角度信息, 根据公式(5)、(10)、(11) 计算当前环境较优的 DWA 算法中评价函数的权值 α_d 、 β_d 、 γ_d , 同时计算机器人与目标点的距离, 如果距离小于设定阈值, 判断是否有障碍物处于目标点附近, 有则进一步判断属于哪种情形, 根据具体情形进一步调整权值 α_d 、 β_d 、 γ_d 、 δ , 没有则权值保持不变;

Step 3 根据机器人运动约束以及安全约束确定线速度和角速度空间;

Step 4 根据从速度空间采样到的速度计算每组速度的模拟路径的方向评价值、障碍物距离评价值、线速度评价值、角速度评价值, 方向评价值、障碍物距离评价值、线速度评价值进行归一化处理, 角速度评价值按照式(8)处理, 和 Step 2 的权值一起代入式(9) 计算最终总评价值, 选出最优速度组合 $(v, w)_{\text{best}}$, 作为下一次机器人运行的速度;

Step 5 执行速度指令, 判断机器人是否到达目标点, 是则结束任务, 否则返回 Step 1 继续循环。

3 仿真实验和结果分析

为了验证本文改进的自适应动态窗口法在经过狭窄过道和抵达一个靠近障碍物的目标点时是否可以优化出一条最优路径, 进一步提高安全性能, 在相同的模拟环境下进行自适应 DWA 算法和本文自适应 DWA 算法的对比实验。

3.1 算法和机器人性能参数初始化

DWA 算法运行开始前, 需要初始化算法参数和机器人性能参数, 模仿实际情形和机器人常规硬件配置参数, 设置机器人和算法参数见表 1~5。

表 1 机器人性能参数

Table 1 Robot performance parameters

最大线速度 $v_{\max}/$ ($\text{m} \cdot \text{s}^{-1}$)	最小线速度 $v_{\min}/$ ($\text{m} \cdot \text{s}^{-1}$)	最大角速度 $w_{\max}/$ ($\text{rad} \cdot \text{s}^{-1}$)	最小角速度 $w_{\min}/$ ($\text{rad} \cdot \text{s}^{-1}$)	线加速度 $\dot{v}/$ ($\text{m} \cdot \text{s}^{-2}$)	角加速度 $\dot{w}/$ ($\text{rad} \cdot \text{s}^{-2}$)
0.18	-0.09	1.8	-0.9	0.13	1.6

表 2 DWA 算法参数

Table 2 Parameters of the DWA algorithm

狭窄过道判定 阈值 C	线速度分辨率/ ($\text{m} \cdot \text{s}^{-1}$)	角速度分辨率/ ($\text{rad} \cdot \text{s}^{-1}$)	时间分辨率/ s	轨迹预测时长/ s	距离阈值/ m
2	0.1	0.1	0.1	3	0.1

表 3 本文自适应 DWA 子函数权值参数

Table 3 Adaptive DWA subfunction weight parameters in this paper

方位角权值 [$\alpha_{\min}, \alpha_{\max}$]	障碍物距离权值 [$\beta_{\min}, \beta_{\max}$]	线速度权值 [$\gamma_{\min}, \gamma_{\max}$]	角速度权值 [$\delta_{\min}, \delta_{\max}$]	设定障碍物 最大数量 $c_{\max}$
[1.0, 10.0]	[2.0, 5.0]	[5.0, 20.0]	[0, 1.5]	10

表 4 初始机器人状态参数

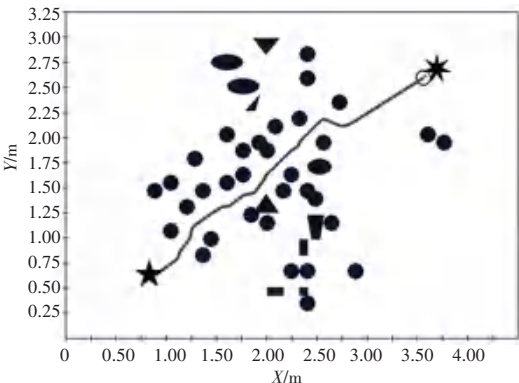
Table 4 Initial robot state parameters			
机器人半径/ m	初始方向/ (°)	初始线速度/ (m · s ⁻¹)	初始角速度/ (rad · s ⁻¹)
0.1	0.0	0.0	0.0

表 5 其它自适应 DWA 子函数权值参数^[9]
Table 5 Other adaptive DWA subfunction weight parameters^[9]

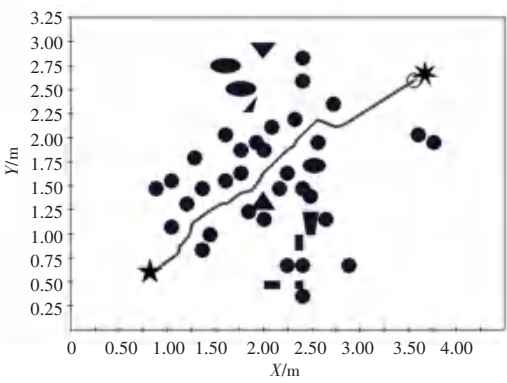
方位角权值 α	障碍物距离权值 β	线速度权值 $[\gamma_{\min}, \gamma_{\max}]$
1.0	5.0	[2.0, 20.0]

3.2 算法定量仿真结果对比与分析

本文自适应 DWA 算法和其它自适应 DWA 算法^[10]的仿真结果如图 3~8 所示,用五角星标示机器人起点位置,七角星标示机器人任务终点位置,黑色圆形表示模拟障碍物,多个障碍物模拟狭窄过道。仿真结果数据对比见表 6。



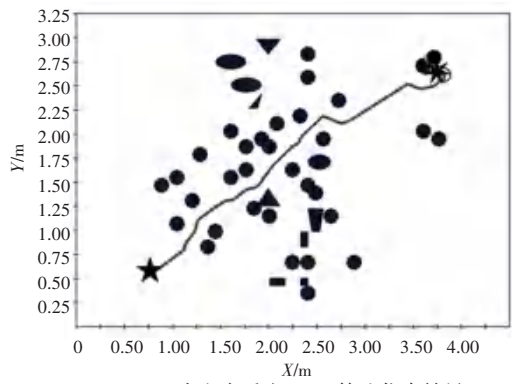
(a) 本文自适应 DWA 算法仿真结果



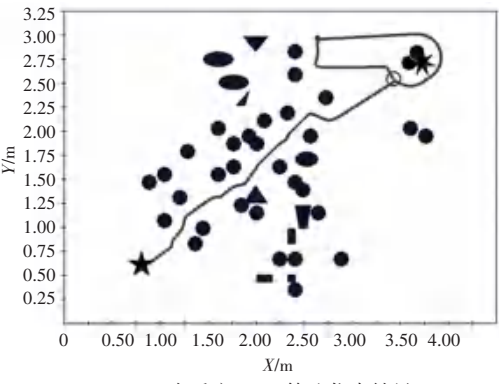
(b) 自适应 DWA 算法仿真结果

图 3 算法在稠密障碍物环境下的仿真结果

Fig. 3 Simulation results of the algorithm in dense obstacle environment



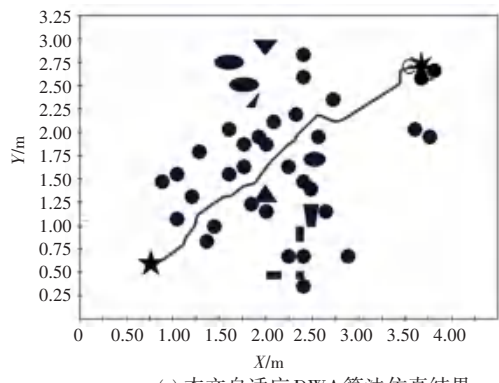
(a) 本文自适应 DWA 算法仿真结果



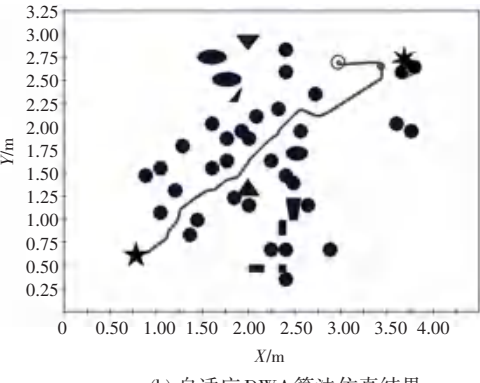
(b) 自适应 DWA 算法仿真结果

图 4 算法在稠密障碍物且目标在障碍物右侧环境下的仿真结果

Fig. 4 Simulation results of the algorithm in the dense obstacle and the target is on the right side of the obstacle environment



(a) 本文自适应 DWA 算法仿真结果



(b) 自适应 DWA 算法仿真结果

图 5 算法在稠密障碍物且目标在障碍物左侧环境下的仿真结果

Fig. 5 Simulation results of the algorithm in the dense obstacle and the target is on the left side of the obstacle environment

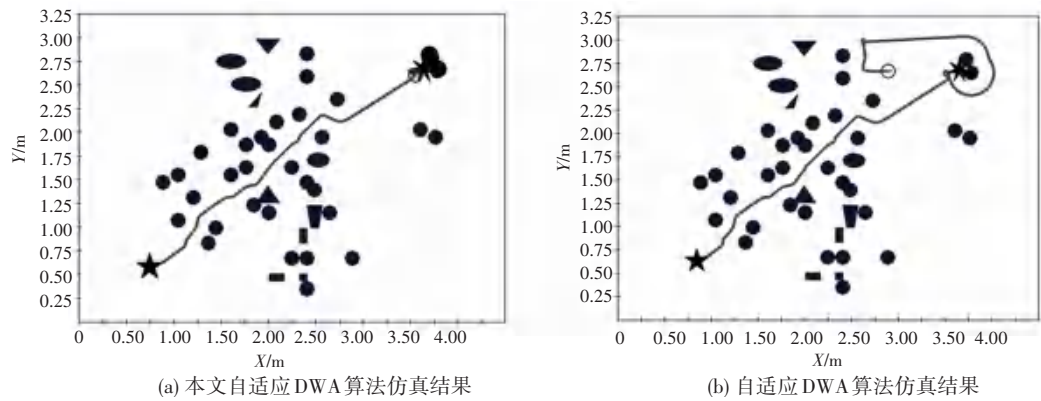


图 6 算法在稠密障碍物且目标在障碍物前侧环境下的仿真结果

Fig. 6 Simulation results of the algorithm in the dense obstacle and the target is on the front side of the obstacle environment

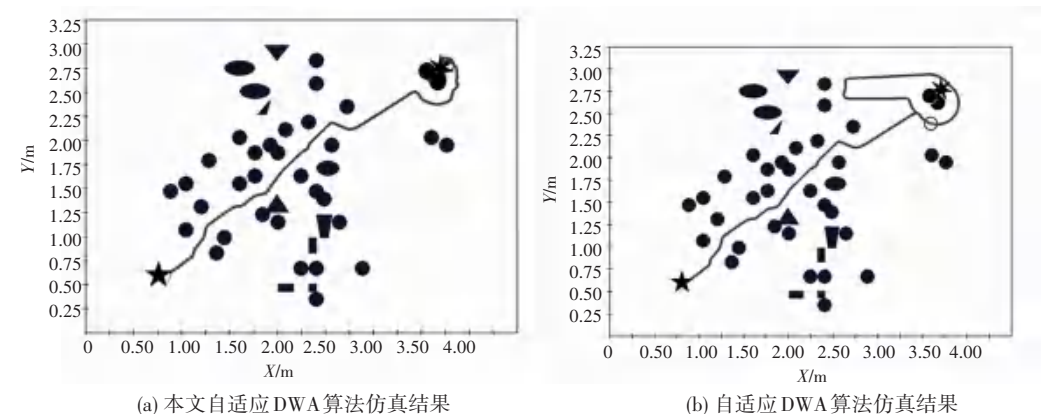


图 7 算法在稠密障碍物且目标在障碍物后侧环境下的仿真结果

Fig. 7 Simulation results of the algorithm in the environment with dense obstacles and the target on the back side of the obstacle

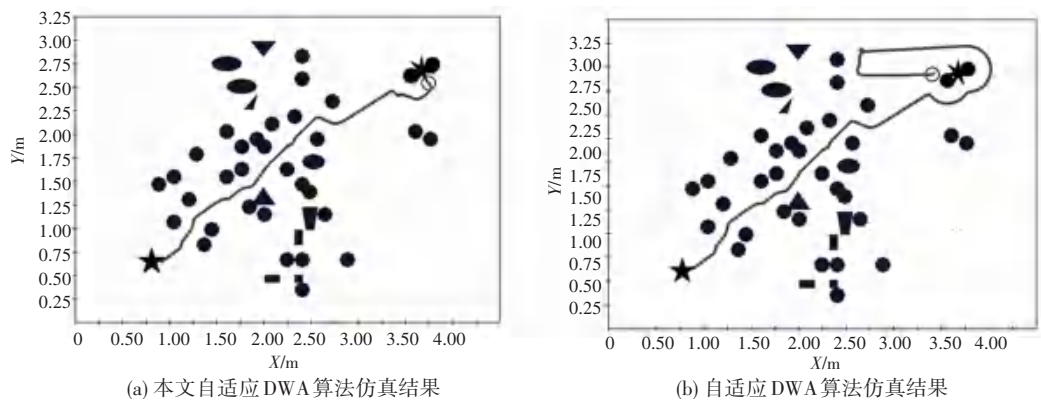


图 8 算法在稠密障碍物且目标在障碍物中间环境下的仿真结果

Fig. 8 Simulation results of the algorithm in the dense obstacle and target in the middle of the obstacle environment

由对比仿真图 3~8 和表 6 可见,本文自适应动态窗口法较自适应动态窗口法^[9]在经过狭窄过道总消耗时间和迭代步数大致相同,本文自适应动

态窗口法在保持安全穿越狭窄过道的基础上较自适应动态窗口法^[9]极大地提高了抵达目标点的成功率。

表 6 仿真结果数据对比

Table 6 Comparison of data of simulation results of algorithms

仿真环境	算法	安全距离/m	迭代步数/步	路长/m	总时间/s	抵达目标点
稠密障碍物环境	自适应 DWA ^[9]	0.011 8	315	4.50	31.5	成功
	本文自适应 DWA	0.012 1	312	4.50	31.2	成功
稠密障碍物且目标在障碍物右侧环境	自适应 DWA ^[9]	0.011 8	>600	>8.55	>60.0	失败
	本文自适应 DWA	0.012 0	341	4.90	34.2	成功
稠密障碍物且目标在障碍物左侧环境	自适应 DWA ^[9]	0.011 8	>600	>8.55	>60.0	失败
	本文自适应 DWA	0.012 1	322	4.62	32.2	成功
稠密障碍物且目标在障碍物前侧环境	自适应 DWA ^[9]	0.011 8	>600	>8.55	>60.0	失败
	本文自适应 DWA	0.012 1	330	4.73	33.0	成功
稠密障碍物且目标在障碍物后侧环境	自适应 DWA ^[9]	0.011 8	>600	>8.55	>60.0	失败
	本文自适应 DWA	0.012 1	400	5.74	40.0	成功
稠密障碍物且目标在障碍物中间环境	自适应 DWA ^[9]	0.011 8	>600	>8.53	>60.0	失败
	本文自适应 DWA	0.012 1	340	4.88	34.1	成功

4 结束语

针对机器人穿越狭窄通道时的局部路径规划效率低、易陷入局部最优、抵达障碍物附近的目标时可能出现目标不可达等问题,本文提出一种改进自适应动态窗口法。根据算法不同阶段不同局部信息设计了障碍物距离评价权值和方向角评价权值自适应调整函数,并在评价函数中加入角速度评价函数。仿真结果表明,改进后的自适应动态窗口法可以使机器人安全穿越狭窄过道,提高了抵达靠近障碍物的目标点的成功率。未来的研究方向是将改进算法应用在具体机器人上来验证算法的优越性。

参考文献

[1] ABDALLAOUI S, AGLZIM E H, CHAIBET A, et al. Thorough review analysis of safe control of autonomous vehicles; Path planning and navigation techniques[J]. Energies, 2022, 15(4): 1358.

[2] AN R, GUO S, ZHENG L, et al. Uncertain moving obstacles avoiding method in 3D arbitrary path planning for a spherical underwater robot[J]. Robotics and Autonomous Systems, 2022, 151: 104011.

[3] LIU W, YU B, GAN Y, et al. Archytas: A framework for synthesizing and dynamically optimizing accelerators for robotic localization[C]//Proceedings of MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture. 2021: 479-493.

[4] HOSSAIN T, HABIBULLAH H, ISLAM R, et al. Local path planning for autonomous mobile robots by integrating modified dynamic-window approach and improved follow the gap method [J]. Journal of Field Robotics, 2022, 39(4): 371-386.

[5] LAI X, WU D. Enhanced DWA algorithm for local path planning of mobile robot[J]. Industrial Robot: The International Journal of Robotics Research and Application, 2022, 50(1): 186-194.

[6] 赵青,陈勇,罗斌,等. 一种融合行人预测信息的局部路径规划算法[J]. 武汉大学学报(信息科学版),2020,45(5):667-675.

[7] KOBAYASHI M, MOTOI N. Local path planning: Dynamic window approach with virtual manipulators considering dynamic obstacles[J]. IEEE Access, 2022, 10: 17018-17029.

[8] LIU L, YAO J, HE D, et al. Global dynamic path planning fusion algorithm combining jump - A* algorithm and dynamic window approach[J]. IEEE Access, 2021, 9: 19632-19638.

[9] 王永雄,田永永. 穿越稠密障碍物的自适应动态窗口法[J]. 控制与决策, 2019 (5): 927-936.

[10] CAO S, FAN P, YAN T, et al. Inland waterway ship path planning based on improved RRT algorithm[J]. Journal of Marine Science and Engineering, 2022, 10(10): 1460.