

文章编号:1671-6833(2024)05-0001-07

受限泊车通道下自动驾驶平行泊车的路径规划方法

秦东晨, 张文灿, 王婷婷, 陈江义

(郑州大学 机械与动力工程学院, 河南 郑州 450001)

摘要:针对受限泊车通道下自动泊车规划时间长、成功率低等问题,提出了一种改进混合 A* 算法的路径规划方法。首先,将泊车路径分为前进的位姿调整段与后退的倒车入库段两部分,通过建立直线-圆弧的约束优化模型规划位姿调整段,寻找合适的倒车起始点;其次,通过在混合 A* 算法中额外引入碰撞风险代价,改进节点扩展方式,并通过判断车辆轮廓线是否与障碍物线相交来进行碰撞检测,以提高倒车入库段规划的实时性和安全性;最后,以路径长度、平滑度、偏离度为指标设计目标函数,并考虑汽车的运动学约束,使用二次规划对初始路径进行平滑,得到最终路径。使用 MATLAB 对改进算法与原始算法进行仿真分析,结果表明:在受限泊车通道下,改进算法能得到平滑的无碰撞泊车路径,搜索时间比混合 A* 算法减少了 23.8%,且所得路径更安全,更易进行跟踪控制。

关键词:自动泊车;混合 A* 算法;路径规划;二次规划;自动驾驶

中图分类号: TP242.6

文献标志码: A

doi:10.13705/j.issn.1671-6833.2024.05.006

随着经济发展,汽车保有量增加,城市中车位不足、泊车环境复杂等问题愈发突出,研究自动泊车系统对解决停车难的问题有巨大帮助^[1]。完整的自动泊车系统可分为环境感知、路径规划、跟踪控制 3 部分^[2]。根据环境信息,快速规划出一条安全无碰撞且易于跟踪控制的路径,是实现自动泊车的关键一环,具有重要研究意义。目前,常用的路径规划方法包括几何曲线法、随机采样法、图搜索法等。

基于几何曲线法,张家旭等^[3]采用双圆弧规划泊车路径,并使用回旋曲线进行平滑,其规划方法所得路径需要较大的泊车空间,否则可能规划失败;胡杰等^[4]采用 5 次多项式规划泊车路径,以碰撞和汽车运动学性能为约束,求解最优泊车路径,但该路径只适用于能一次泊入的库位,无法进行库内调整泊车路径;胡杰等^[5]将泊车路径分为入库段与库内调整段,通过重复使用“之”字形的库内调整路径,使车辆通过不断前进后退,最终停至目标位置,但调整次数过多,舒适性较差,对跟踪控制的要求较高。几何曲线法的优势在于计算效率高,可快速求得泊车路径。由于使用预设的曲线类型作为泊车路径,然后根据车辆参数与环境信息确定曲线具体参数,当

泊车空间较小,车辆无法通过预设曲线泊入车位时,几何曲线法就无法计算出所需路径。

基于随机采样法,Jhang 等^[6]采用 RRT* 算法规划泊车路径,通过对状态空间进行采样,使用 Reeds-Shepp (R-S) 曲线连接一系列采样点,得到泊车路径,但该方法实时性差;王海芳等^[7]针对 RRT 算法搜索效率差的问题,提出了 Bi-RRT* 算法,其在起点与终点同时开始搜索,最后交汇于一点,但该方法仅适用于移动机器人,不能满足汽车的运动学约束;Tahir 等^[8]在 Bi-RRT* 的基础上对采样空间进行限制,减少了采样次数,提高了实时性,但对于狭窄空间下的泊车适应性不佳。随机采样法的优势在于其直接对状态空间进行采样,因此具有完备性,即当采样点足够多时,一定能得到可行解。但由于车辆自身的非完整性约束,车辆具有横纵坐标与航向角 3 个状态变量,采样维度的上升会导致采样数成指数级增长,同时由于采样的随机性,在自动泊车这个对于规划路径的精度要求较高的场景下,规划所用时间较长,且规划路径一般不是最优的。

基于图搜索的方法,段书用等^[9]使用 A* 算法,并加入安全距离矩阵,保证机器人搜索出路径的安

收稿日期:2024-02-18; **修订日期:**2024-04-02

基金项目:国家重点研发项目(2018YFB0106204-03);河南省重大科技专项(221100240200)

作者简介:秦东晨(1965—),男,河南温县人,郑州大学教授,博士,主要从事机械强度及结构优化设计、虚拟样机、CAD/CAE/CAM 集成方面研究,E-mail:deqin@zzu.edu.cn。

引用本文:秦东晨,张文灿,王婷婷,等.受限泊车通道下自动驾驶平行泊车的路径规划方法[J].郑州大学学报(工学版),2024,45(5):1-7.(QIN D C,ZHANG W C,WANG T T,et al. A path planning method for parallel parking in constrained parking channels[J].Journal of Zhengzhou University(Engineering Science),2024,45(5):1-7.)

全性,但未考虑运动学约束;曹彦博等^[10]对混合 A* 算法的启发值获取方式进行改进,预先创建离线的代价地图,后续搜索过程中无须逐一计算,提高了算法的实时性,但未解决曲率不连续的问题;Zhang 等^[11]针对搜索曲线连接点曲率不连续的问题,使用螺旋线代替 R-S 曲线中的圆弧连接节点,但未能提高搜索效率。图搜索法中的 A* 算法通过遍历整个状态空间,可搜索出最短路径,但未考虑车辆运动学约束,不能直接用于自动泊车的路径规划。混合 A* 算法引入了车辆非完整性约束,所得路径可直接跟踪,但由于其通过采样控制变量,经过计算间接获得状态变量,不具有完备性。

针对受限泊车通道下路径规划时间长、路径质量差等问题,本文改进混合 A* 算法的节点扩展与碰撞检测方式,提高实时性,并通过二次规划对初始路径进行平滑,改善了曲率突变的问题。

1 混合 A* 算法

混合 A* 算法是由斯坦福大学 Dolgov 等^[12]提出的一种算法,是一种在离散 x - y - θ 状态空间使用 A* 策略并融合 R-S 曲线机制的路径规划方法,并在 DARPA 挑战赛中得到了应用。

在 A* 算法的基础上,混合 A* 算法考虑到了车辆的运动学约束,引入航向角 θ 这一状态量,并使用 R-S 曲线进行节点扩展,如图 1 所示,同时在扩展一定次数后,尝试使用 R-S 曲线连接当前节点与终点,以加快搜索速度。

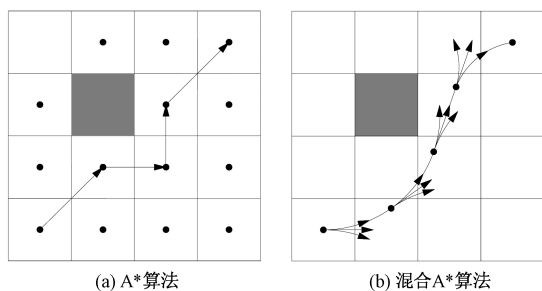


图 1 两种算法节点拓展方式对比

Figure 1 Comparison of node expansion methods in two algorithm

同时,混合 A* 算法更改了启发函数 h 的定义,将 h 定义为两个子启发函数 h_1 和 h_2 的较大者,其中 h_1 代表一条符合车辆运动学约束但无视碰撞的路径长度,通常用 R-S 曲线表征, h_2 代表一条考虑避障但未必运动学可行的路径长度,通常用 A* 路径长度和曼哈顿距离的较大值来表征。然而,对于平行泊车工况这一局部路径规划,由于汽车的运动学约束,起始点与目标点的 h_2 距离一般远小于实际距

离, h_2 无法真实反映情况。而 R-S 曲线由于没有考虑障碍物的影响,同样会远短于实际距离, h_1 同样无法反映真实情况。这两种子启发函数在算法刚开始搜索时误差格外明显,导致搜索了大量的无用节点,延长了搜索时间,且可能导致搜索失败。

因此,将泊车路径分为两段,其中前进段路径由于空间有限,相对简单固定,使用曲线构造,并得到倒车起始点,后退段考虑到目标点周围环境更复杂,不利于混合 A* 算法的加速搜索机制,因此从目标点到倒车起始点进行反向搜索。

2 平行泊车规划

2.1 前进调整段

当车位旁通道足够宽且足够长时,位姿调整路径可以是一条直线。而在通道长宽受限时,位姿调整段需要借用车位空间,以实现在有限空间内的平行泊车,如图 2 所示。首先要确定的是车辆借用车位进行位姿调整的路径,确定倒车起始点,然后规划从目标位置到倒车起始点的路径。

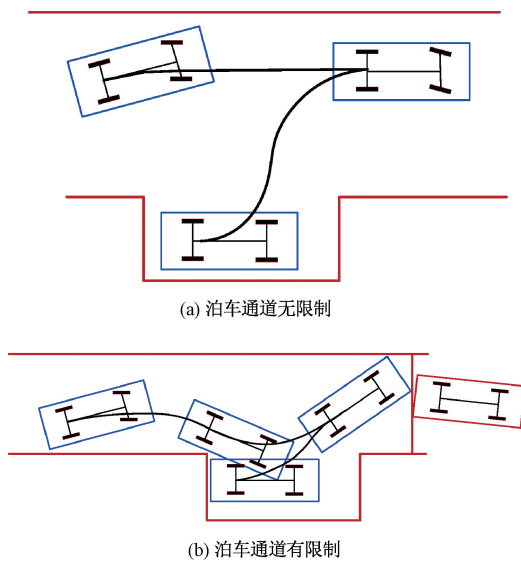


图 2 平行泊车路径对比

Figure 2 Comparison of parallel parking paths

首先,要确定从何处开始借用车位空间;其次,确定汽车与过道边界不发生碰撞的极限位置;最后,规划一条从极限位置到泊车位置的无碰撞路径。

如图 3 所示,当借用车位空间时,车辆侧面不能与障碍物发生碰撞。为尽可能利用车位空间, B 点纵坐标应尽可能小,即目标函数和约束为

$$\begin{cases} \min y_B; \\ \text{s. t. } (x_O - x_E)^2 + (y_O - y_E)^2 \geq (R_A + d_s)^2, \\ (x_O - x_F)^2 + (y_O - y_F)^2 \geq (R_B + d_s)^2. \end{cases} \quad (1)$$

式中: x_O 、 y_O 为汽车转向中心 O 点的横纵坐标; x_E 、

y_E 为图 3 中 E 点的横纵坐标; x_F, y_F 为图 3 中 F 点的横纵坐标; R_A, R_B 为汽车转向中心 O 点到图 3 中 A 点、 B 点的距离; d_s 为安全距离。

相关参数均可用汽车后轴中心坐标 x_1, y_1 和车辆航向角 θ 表示:

$$R_A = \sqrt{(R + W/2)^2 + L_r^2}; \quad (2)$$

$$R_B = \sqrt{(R + W/2)^2 + (L_w + L_f)^2}; \quad (3)$$

$$y_B = y_1 + (L_w + L_f) \sin \theta - 0.5W \cos \theta; \quad (4)$$

$$x_O = x_1 + R \cos(\theta + \pi/2); \quad (5)$$

$$y_O = y_1 + R \sin(\theta + \pi/2)。 \quad (6)$$

式中: R 为汽车最小转弯半径, m ; W 为汽车宽度, m ; L_f 为车辆前悬, m ; L_w 为车辆轴距, m ; L_r 为车辆后悬, m ; θ 为车辆航向角, rad ; x_1, y_1 为汽车后轴中心横纵坐标。

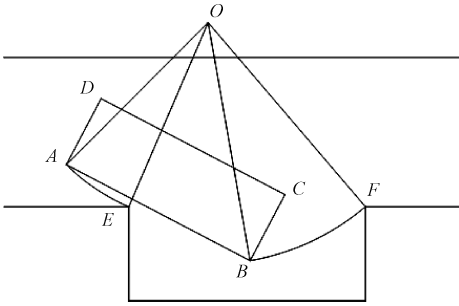


图 3 确定借用车位的位置

Figure 3 Determination of the location of the parking space used

得到车辆的转向中心坐标 (x_O, y_O) 和位置 1 的后轴中心坐标 (x_1, y_1) 及航向角 θ_1 后, 进一步计算后退泊车段的起始点。如图 4 所示, C 点纵坐标应尽可能大, 且 B 点和 C 点不与障碍物发生碰撞, 即目标函数和约束为

$$\begin{cases} \min(-y_C); \\ \text{s. t. } y_C + d_s \leq y_F + h, \\ x_B + d_s \leq x_F + x_{\text{limit}}。 \end{cases} \quad (7)$$

相关参数均可使用变量 α 表示, 其为车辆从位置 1 驶至位置 2 时, 后轴中心与转向中心连接所得线段旋转的角度。

$$\begin{bmatrix} x_2 - x_O \\ y_2 - y_O \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} x_1 - x_O \\ y_1 - y_O \end{bmatrix}; \quad (8)$$

$$y_C = y_2 + (L_w + L_f) \sin \theta_2 + 0.5W \cos \theta_2; \quad (9)$$

$$x_B = x_2 + (L_w + L_f) \cos \theta_2 + 0.5W \sin \theta_2。 \quad (10)$$

2.2 后退泊车段

后退泊车段采用混合 A^* 算法从目标点到泊车起始点反向搜索, 针对采用原始混合 A^* 算法碰撞检测精度低、规划路径过于靠近障碍物的问题, 改变其

碰撞检测方式, 并引入新的代价函数。

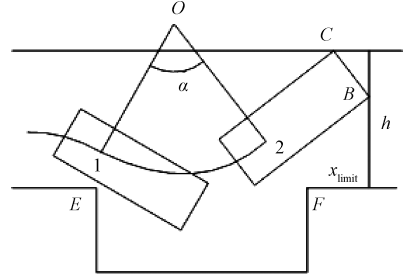


图 4 确定泊车起始点

Figure 4 Determination of the starting point for parking

2.2.1 碰撞检测

混合 A^* 算法使用栅格地图来描绘环境信息, 其碰撞检测通过判断障碍物所占据的栅格与车身所占栅格的距离来判断是否发生碰撞。混合 A^* 算法最初是一种用于全局路径规划的算法, 这种碰撞检测方法可以较快判断车辆是否与数个复杂障碍物发生碰撞, 但其对障碍物与车辆的描述不准确, 存在一定的误差, 而栅格划分的分辨率越低, 这一误差越大^[13]。在泊车这一局部路径规划时, 栅格划分分辨率过低会导致无法规划出可行路径, 而分辨率过高会显著增加规划时间。

由于泊车工况车辆周围环境较为简单, 不存在复杂障碍物, 因此, 只需要考虑车辆停入库位时的碰撞。同时, 考虑到泊车时对碰撞检测的精度要求较高, 用矩形表示车辆轮廓, 用折线段表示车位及障碍物轮廓, 通过快速排斥试验与跨立试验判断车辆轮廓线是否与车位及障碍物轮廓线相交来判断是否发生碰撞。这种方法在世界坐标系下进行计算, 简单方便, 同时精度较高, 能准确描述车辆与障碍物的关系。

$$\begin{cases} \max(M_x, N_x) < \min(P_x, Q_x); \\ \max(M_y, N_y) < \min(P_y, Q_y); \\ \min(M_x, N_x) > \max(P_x, Q_x); \\ \min(M_y, N_y) > \max(P_y, Q_y)。 \end{cases} \quad (11)$$

$$\begin{cases} (\overrightarrow{MN} \times \overrightarrow{MP}) \cdot (\overrightarrow{MN} \times \overrightarrow{MQ}) < 0; \\ (\overrightarrow{PQ} \times \overrightarrow{PM}) \cdot (\overrightarrow{PQ} \times \overrightarrow{PN}) < 0。 \end{cases} \quad (12)$$

对于线段 MN 和线段 PQ , 首先判断是否满足式 (11) 中任意一个条件, 若满足, 两条线段不相交。若都不满足, 再判断是否同时满足式 (12) 中条件, 若同时满足, 则两条线段相交, 否则不相交。

2.2.2 代价函数

混合 A^* 算法搜索出的路径通常较为靠近障碍物, 在泊车场景下, 这会对跟踪控制提出更高的要求。解决方案通常是对车辆或障碍物进行膨胀处

理,而在泊车场景中,空间狭窄,可行区域较小,膨胀处理可能导致算法无法搜索到可行解。为解决这一问题,在代价函数中额外引入碰撞风险函数,即代价函数的形式变为

$$f = g + h + d. \quad (13)$$

式中: f 为扩展节点的代价函数; g 为历史路径长度,表示从起始节点扩展至当前节点的累积代价值; h 为启发函数,表示从当前节点扩展至终止节点代价的估计值; d 为碰撞风险函数。

碰撞风险函数应能快速且较为准确地反映车辆与障碍物之间的距离。扩展节点包含车辆的位姿信息 (x_0, y_0, θ_0) , 以此构造互相垂直的两条直线:

$$y - y_0 = (x - x_0) \tan \theta_0; \quad (14)$$

$$y - y_0 = - (x - x_0) \frac{1}{\tan \theta_0}. \quad (15)$$

两条直线可与障碍物轮廓线产生 4 个交点,各交点到车辆后轴中心的距离分别为 l_1, l_2, l_3, l_4 , 如图 5 所示。

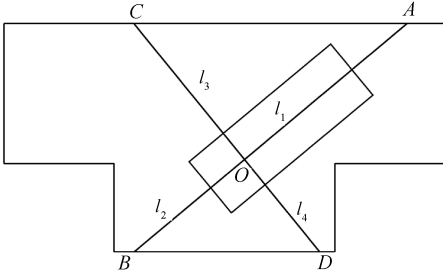


图 5 碰撞风险函数示意图

Figure 5 Schematic diagram of collision risk function

考虑到车辆自身轮廓,令:

$$d_0 = \min \left(l_1 - L_f - L_w, l_2 - L_r, l_3 - \frac{W}{2}, l_4 - \frac{W}{2} \right). \quad (16)$$

$$d = \begin{cases} 0, & d_0 > d_{\text{safe}}; \\ d_{\text{safe}} - d_0, & 0 < d_0 \leq d_{\text{safe}}; \\ \text{non}, & d_0 \leq 0. \end{cases} \quad (17)$$

式中: d_{safe} 表示安全距离; non 表示车辆与障碍物发生碰撞,不向该节点进行扩展。

在向节点扩展时,优先考虑碰撞风险函数 d 更小的子节点。当碰撞风险差距不大时,优先向原代价函数 $g + h$ 更小的节点扩展。这样,在不影响搜索可行解成功率的情况下,可以使车辆尽可能远离障碍物,减少跟踪难度。

2.3 路径平滑

由于前文所得路径为通过前轮转角进行离散采样获得的,初始路径存在曲率不连续的问题,因此需要对初始路径进行平滑的后处理。在泊车场景中,

存在进退挡位切换的情况,因而对初始路径进行分段处理,每段的处理如下。

混合 A* 算法通过采样可获得 m 个等距的离散路径点,记为 $p_0(x_0, y_0), p_1(x_1, y_1), \dots, p_{m-1}(x_{m-1}, y_{m-1})$ 。目标函数由 3 项组成,表示形式如下:

$$J = \omega_1 \sum_{i=1}^{m-1} [(x_{i+1} + x_{i-1} - 2x_i)^2 + (y_{i+1} + y_{i-1} - 2y_i)^2] + \omega_2 \sum_{i=0}^{m-1} [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2] + \omega_3 \sum_{i=0}^{m-1} [(x_i - x_{i_{\text{ref}}})^2 + (y_i - y_{i_{\text{ref}}})^2]. \quad (18)$$

式中: $\omega_1, \omega_2, \omega_3$ 为各项优化目标的权重系数。等式右侧第 1 项优化目标是使优化路径尽可能平滑,其物理意义为向量 $\overrightarrow{P_1P_0}$ 和向量 $\overrightarrow{P_1P_2}$ 相加所得 $\overrightarrow{P_2P_3}$ 的模的平方,其值越小,代表 P_0, P_1, P_2 这 3 点组成的曲线越趋于平直,因此可反映路径的平滑性;第 2 项优化目标是使优化路径尽可能短;第 3 项优化目标是使优化路径尽可能不过于偏离初始路径。

由于本文对泊车路径分段进行优化处理,因此每段的起始点和终点的位置和航向角均保持不变,可得到位置约束:

$$\begin{cases} x_i = x_{i_{\text{ref}}}; \\ y_i = y_{i_{\text{ref}}}; \\ i = 0, 1, m-2, m-1. \end{cases} \quad (19)$$

在优化过程中,需要对坐标 (x_i, y_i) 的变化范围进行一定的限制,以防优化后的路径与障碍物发生碰撞,可得碰撞约束:

$$\begin{cases} l_{\text{bound}} \leq x_i - x_{i_{\text{ref}}} \leq u_{\text{bound}}; \\ l_{\text{bound}} \leq y_i - y_{i_{\text{ref}}} \leq u_{\text{bound}}; \\ i = 0, 1, \dots, m-1. \end{cases} \quad (20)$$

由于车辆自身的运动学特性,优化路径存在最大曲率的限制,可得曲率约束:

$$(\Delta s^2 \text{cur})^2 \geq (x_{i+1} + x_{i-1} - 2x_i)^2 + (y_{i+1} + y_{i-1} - 2y_i)^2, \quad i = 1, 2, \dots, m-1. \quad (21)$$

由于曲率约束为非线性约束,可通过线性化方法将其转化为线性约束。令 $F = (x_{i+1} + x_{i-1} - 2x_i)^2 + (y_{i+1} + y_{i-1} - 2y_i)^2$, 在参考路径处泰勒展开,保留线性项可得

$$F'(P_{\text{ref}})P \leq (\Delta s^2 \text{cur})^2 - F(P_{\text{ref}}) + F'(P_{\text{ref}})P_{\text{ref}}. \quad (22)$$

综上所述,路径平滑可以看作以式(18)为目标函数,以式(19)、(20)、(22)为约束条件的优化问题,使用二次规划的方法求解。

3 仿真分析

3.1 平行泊车路径规划

路径搜索仿真所用车辆长度为 4.6 m,宽度为 1.8 m,轴距为 2.7 m,前悬为 0.9 m,后悬为 1.0 m,最大前轮转角为 34°。车位长 6.0 m,宽 2.4 m,泊车通道宽 3.8 m,长度方向受限。起点位姿为 (1.1, 3.9, 0),终点位姿为 (7.1, 1.4, 0)。分别使用混合 A* 算法与本文方法进行泊车路径搜索,路径规划结果如图 6 所示。

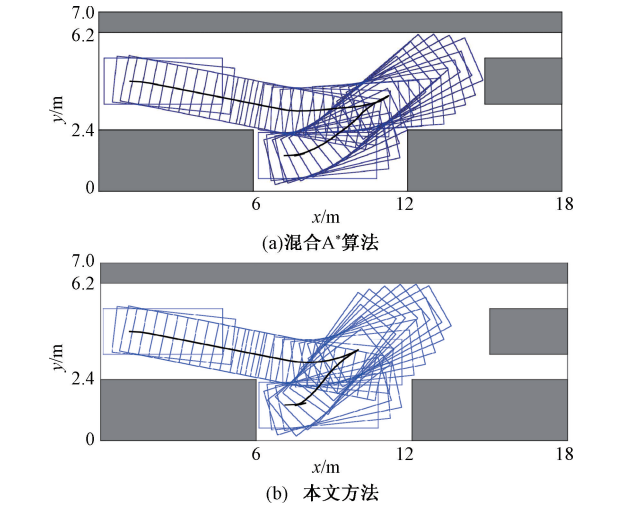


图 6 工况 1 路径规划结果

Figure 6 Path planning results of case one

对比两种方法所得结果可知,两种方法都可以搜索出无碰撞且符合车辆运动学约束的泊车路径,但混合 A* 算法所得路径在车辆后退驶入库位时,右前方车头过于靠近障碍物,存在碰撞风险,对路径跟踪的精度要求较高。而本文方法由于引入了代表碰撞风险的代价函数,使节点扩展时尽量远离障碍物,因此所得路径与障碍物间保持着较为安全的距离,降低了后续进行跟踪控制的难度。两种方法所得结果的主要参数如表 1 所示,相较于混合 A* 算法,本文所提方法搜索节点数减少了 19.4%,搜索时间减少了 23.8%,提高了路径搜索的实时性。

表 1 工况 1 仿真结果		
Table 1 Test results of case one		
算法	搜索节点数	搜索时间/s
混合 A* 算法	216	0.863
本文方法	174	0.658

在工况 1 的基础上,将起点位姿改变为 (1.1, 3.9, 0.2),以进一步验证本文方法在不同起始位姿时的路径规划效果,泊车路径搜索结果如图 7 和表 2 所示。

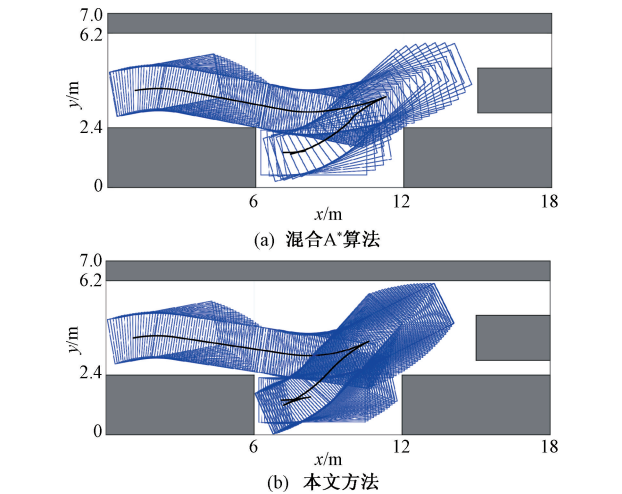


图 7 工况 2 路径规划结果

Figure 7 Path planning results of case two		
表 2 工况 2 仿真结果		
Table 2 Test results of case two		
算法	搜索节点数	搜索时间/s
混合 A* 算法	238	0.962
本文方法	183	0.734

根据图 7 和表 2 可知,在起点位姿的航向角不为零时,本文方法仍能快速地搜索出一条无碰撞泊车路径,相较于混合 A* 算法,本文方法搜索节点数减少了 23.1%,搜索时间减少了 23.7%,提升了实时性。

为进一步验证本文方法在泊车通道受限情况下的路径规划能力,在工况 1 基础上,进一步限制泊车通道的宽度,将其设为 3 m,其他设置保持不变。分别使用混合 A* 算法与本文所提方法进行路径搜索,路径规划结果如表 3 和图 8 所示。

表 3 工况 3 仿真结果		
Table 3 Test results of case three		
使用算法	搜索节点数	搜索时间/s
混合 A* 算法	1 383	7.973
本文方法	269	1.287

根据表 3 及图 8 可知,两种方法均可得到可行路径。混合 A* 算法搜索节点个数与耗费时间显著增加,搜索得到的路径在车位中多次改变行车方向,增加了跟踪控制的难度,影响行车的舒适性。观察车位内的路径可得,车辆在调整位姿时过多进行直线行驶,降低了车辆泊入车位的效率。而本文方法所得路径由于碰撞风险函数影响,在位姿调整时更多以最小转向半径前进后退,更符合驾驶员操作习惯,整个泊车流程共切换挡位 5 次,显著少于混合 A* 算法的规划结果。

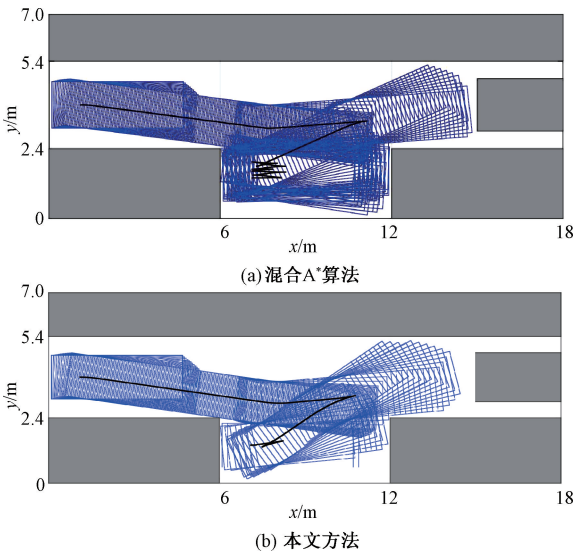


图 8 工况 3 路径规划结果

Figure 8 Path planning results of case three

3.2 路径平滑

使用本文路径平滑方法对图 6(b)中规划路径进行平滑,耗时约为 0.152 s,满足实时性要求,所得结果如图 9、图 10 所示。

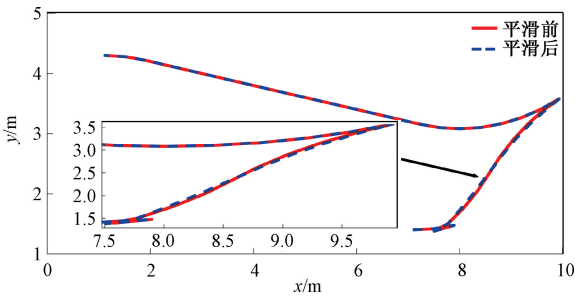


图 9 路径平滑结果

Figure 9 Results of path smoothing

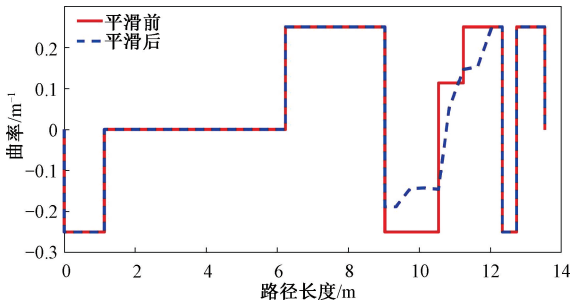


图 10 路径平滑后的曲率变化

Figure 10 Curvature change after path smoothing

结合图 9 和图 10 可以看出,在开始段与结束段,车辆始终以最小转向半径进行转向,因此平滑后的路径与原始路径相同,保证了进退挡位切换点的车辆姿态保持不变。而在中间段存在曲率变化时,经平滑方法处理后的路径曲率变化更合理,改善了曲率突变的情况。

4 结论

本文提出了一种针对泊车通道受限情况下的平行泊车路径规划方法,该方法首先构造前进的位姿调整段路径,确定合适的泊车起始点;其次,改变混合 A* 算法的碰撞检测方式并引入代表碰撞风险的代价函数,搜索后退的泊车路径;再次,使用二次规划对所得初始路径进行平滑;最后,设计仿真试验进行验证。结果表明,本文所提方法满足路径搜索的实时性和安全性要求。

参考文献:

[1] 孙超,李世杰,张鹏,等. 出行系统最优配置下共享停车选择及泊位分配[J]. 郑州大学学报(工学版), 2023, 44(1): 38-43.

SUN C, LI S J, ZHANG P, et al. Shared parking selection and berth allocation for optimal configuration of travel system[J]. Journal of Zhengzhou University (Engineering Science), 2023, 44(1): 38-43.

[2] 黄江,魏德奎,秦良艳,等. 自动泊车系统路径规划与跟踪控制方法研究[J]. 汽车技术, 2019(8): 39-45.

HUANG J, WEI D K, QIN L Y, et al. Research on path planning and path tracking control method of automatic parking system[J]. Automobile Technology, 2019(8): 39-45.

[3] 张家旭,赵健,施正堂,等. 基于回旋曲线的平行泊车路径规划和跟踪控制[J]. 吉林大学学报(工学版), 2020, 50(6): 2247-2257.

ZHANG J X, ZHAO J, SHI Z T, et al. Path planning and tracking control for parallel parking based on clothoid curve[J]. Journal of Jilin University (Engineering and Technology Edition), 2020, 50(6): 2247-2257.

[4] 胡杰,张敏超,徐文才,等. 自动驾驶车辆的平行泊车轨迹规划[J]. 汽车工程, 2022, 44(3): 330-339.

HU J, ZHANG M C, XU W C, et al. Parallel parking trajectory planning for autonomous vehicles[J]. Automotive Engineering, 2022, 44(3): 330-339.

[5] 胡杰,朱令磊,陈瑞楠,等. 狭小车位平行泊车路径规划方法研究[J]. 汽车工程, 2022, 44(7): 1040-1048.

HU J, ZHU L L, CHEN R N, et al. Research on parallel parking path planning method for narrow parking space[J]. Automotive Engineering, 2022, 44(7): 1040-1048.

[6] JHANG J, LIAN F L, HAO Y. Human-like motion planning for autonomous parking based on revised bidirectional rapidly-exploring random tree* with Reeds-Shepp curve[J]. Asian Journal of Control, 2021, 23(3):

1146-1160.

[7] 王海芳,张瑶,朱亚锐,等. 基于改进双向 RRT* 的移动机器人路径规划算法[J]. 东北大学学报(自然科学版), 2021, 42(8): 1065-1070, 1142.

WANG H F, ZHANG Y, ZHU Y K, et al. Mobile robot path planning based on improved bidirectional RRT* [J]. Journal of Northeastern University (Natural Science), 2021, 42(8): 1065-1070, 1142.

[8] TAHIR Z, QURESHI A H, AYAZ Y, et al. Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments[J]. Robotics and Autonomous Systems, 2018,108:13-27.

[9] 段书用,王启帆,韩旭,等. 具有确保安全距离的 A* 路径优化方法[J]. 机械工程学报, 2020, 56(18): 205-215.

DUAN S Y, WANG Q F, HAN X, et al. Improved A-star algorithm for safety insured optimal path with smoothed corner turns[J]. Journal of Mechanical Engineering, 2020, 56(18): 205-215.

[10] 曹彦博,颜京才,李旭升,等. 基于改进混合 A* 算法的自动泊车系统路径搜索方法[J]. 汽车技术, 2023 (6): 37-41.

CAO Y B, YAN J C, LI X S, et al. A method of path search for automatic parking system based on improved hybrid A* algorithm[J]. Automobile Technology, 2023 (6): 37-41.

[11] ZHANG S Y, CHEN Y, CHEN S T, et al. Hybrid A-based curvature continuous path planning in complex dynamic environments[C]//2019 IEEE Intelligent Transportation Systems Conference (ITSC). Piscataway: IEEE, 2019: 1468-1474.

[12] DOLGOV D, THRUN S, MONTEMERLO M, et al. Path planning for autonomous vehicles in unknown semi-structured environments[J]. International Journal of Robotics Research, 2010, 29(5): 485-501.

[13] 陈翔翔. 基于 Hybrid A* 算法的自动泊车路径规划与跟踪控制研究[D]. 西安: 长安大学, 2022.

CHEN X X. Research on Hybrid A* algorithm path planning and tracking control for automatic parking [D]. Xi'an: Chang'an University, 2022.

A Path Planning Method for Parallel Parking in Constrained Parking Channels

QIN Dongchen, ZHANG Wencan, WANG Tingting, CHEN Jiangyi

(School of Mechanical and Power Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: To solve to the problems of long planning times and low success rates in automatic parking in constrained parking lanes, a modified approach to path planning was proposed, which could improve the hybrid A* algorithm. Firstly, the parking path was divided into two parts: a forward pose adjustment segment and a reverse parking segment. A linear-arc constrained optimization model was established to plan the pose adjustment segment, while finding a suitable starting point for reverse parking. Subsequently, collision risk cost was introduced as an additional factor in the hybrid A* algorithm. The node expansion process was improved by checking whether the vehicle contour intersects with obstacle lines, enhancing the real-time performance and safety of the reverse parking segment. Finally, a cost function was designed with criteria such as path length, smoothness, and deviation, taking into account the vehicle’s kinematic constraints. A quadratic programming approach was used to smooth the initial path, resulting in the final path. Simulation analysis was conducted using MATLAB to compare the modified algorithm with the original algorithm. The results showed that, in constrained parking lanes, the improved algorithm could produce a smooth, collision-free parking path with a reduced search time of 23.8% compared with the hybrid A* algorithm. Additionally, the obtained path was safer and more suitable for tracking control.

Keywords: automatic parking; hybrid A* algorithm; path planning; quadratic programming; autonomous driving