

doi: 10.19562/j.chinasae.qcgc.2023.12.012

基于双层求解策略的平行泊车轨迹规划*

张洪昌^{1,2,3}, 宁 鹏^{1,2,3}, 杨 杰^{1,2,3}, 宋建伟^{1,2,3}, 郝 麟^{1,2,3}, 曾 娟^{1,2,3}

(1. 武汉理工大学, 现代汽车零部件技术湖北省重点实验室, 武汉 430070;

2. 武汉理工大学, 汽车零部件技术湖北省协同创新中心, 武汉 430070;

3. 武汉理工大学, 湖北省新能源与智能网联车工程技术研究中心, 武汉 430070)

[摘要] 轨迹规划在泊车系统中承接了上层感知和下层控制,是缩短泊车时间和降低跟踪难度的关键手段。针对平行泊车轨迹规划难以兼顾轨迹的生成质量、泛化能力和计算效率等问题,提出了基于双层求解策略的平行泊车轨迹规划方法。该方法分为两层:第一层将平行泊车的路径分为由锚点连接的两段路径,采用路径倒推的思路寻找出锚点,分别规划从终点到锚点的路径和从锚点到起点的路径,之后对路径附加“时间最优”剖面,逆序得到特定时刻的状态量与控制量;第二层采用同步联立正交配置法,将平行泊车最优控制中连续的状态量和控制量转化为轨迹非线性规划的离散点,把第一层得到的状态量与控制量作为初始值带入非线性规划求得数值解。同时,建立了5种平行车位泊车场景模型并进行了仿真分析,结果表明,针对不同的泊车起始位姿和车位尺寸,均能规划出满足约束条件的最优轨迹,提高了轨迹的生成质量和泛化能力,并具有较好的计算效率。

关键词: 轨迹规划; 双层求解策略; 平行泊车; 数值优化

Parallel Parking Trajectory Planning Based on Double-Layer Solution Strategy

Zhang Hongchang^{1,2,3}, Ning Peng^{1,2,3}, Yang Jie^{1,2,3}, Song Jianwei^{1,2,3},Hao Lin^{1,2,3} & Zeng Juan^{1,2,3}

1. Wuhan University of Technology, Hubei Key Laboratory of Modern Auto Parts Technology, Wuhan 430070;

2. Wuhan University of Technology, Auto Parts Technology Hubei Collaborative Innovation Center, Wuhan 430070;

3. Wuhan University of Technology, Hubei Technology Research Center of New Energy and Intelligent Connected Vehicle Engineering, Wuhan 430070

[Abstract] Trajectory planning plays a key role in shortening parking time and reducing tracking difficulty, as it connects the upper-level perception and the lower-level control in parking systems. However, it is challenging to balance trajectory quality, generalization ability, and computational efficiency in parallel parking trajectory planning. To address this issue, the parallel parking trajectory planning based on double-layer solution strategy (DLSS) is proposed. The strategy includes two layers: in the first layer, the parallel parking path is divided into two segments connected by anchor points. The anchor point is identified through a backward path planning approach. The paths from the endpoint to the anchor point and from the anchor point to the starting point are planned separately. Then, a "time-optimal" profile is added to the path, and the state and control variables at specific time points are obtained in reverse order. In the second layer, the simultaneous orthogonal configuration method is used to transform the continuous state and control variables in the parallel parking optimal control into discrete variables in the trajectory nonlinear programming. The state and control variables obtained in the first layer are used as the initial values for the nonlinear programming to obtain numerical solutions. Five parallel parking scene models are estab-

* 教育部创新团队发展计划(IRT_17R83)、新能源汽车科学与关键技术学科创新引智基地(B17034)和武汉理工大学重庆研究院科技创新研发项目(YF2021-07)资助。

原稿收到日期为 2023 年 04 月 05 日,修改稿收到日期为 2023 年 05 月 11 日。

通信作者:曾娟,副教授,博士,E-mail:zhc112@126.com。

lished and simulated, and the results show that the optimal trajectory that meets the constraints requirements can be planned for different parking starting postures and parking space sizes, which improves the generation quality and generalization ability of trajectories, and has satisfactory computational efficiency.

Keywords: trajectory planning; double-layer solution strategy; parallel parking; numerical optimization

前言

随着科技的发展,人们对出行便捷的关注度越来越高,希望自动泊车系统能够更加的智能。路径规划是自动泊车系统的重要组成部分,文献[1]和文献[2]中对目前常用的路径规划方法做了系统性的概述,将常用的路径规划方法分为了图搜索法、采样法、几何法以及数值优化法4类。任秉韬等^[3]结合混合A*与可变半径RS曲线规划了泊车的路径,提高了复杂场景下的泊车规划成功率。He等^[4]采用一种快速A*算法来规划一条路径,提高规划的速度,但是该算法泊车的成功率与车辆的起始位姿有关,对自动泊车路径规划的适应性较差。Han等^[5]使用RRT规划了一条泊车路径,但是路径的曲率不连续,且要求车位的空间较大。Dong等^[6]提出了一种基于经验的RRT算法,解决了不同角度车位的路径规划问题,但路径的最优性无法得到保证。几何法将车辆泊车时可能发生的关键点罗列出来,而后采用多项式曲线^[7-8]、回旋曲线^[9]或B样条曲线^[10-11]等方法对关键点进行曲线拟合,该类方法适用场景有限,无法针对某一指标值达到最优。同时,还有一类基于模糊控制^[12]和神经网络^[13]的自动泊车路径规划算法,但是需要足够的专家经验和数据样本。

轨迹规划在路径规划的基础上增加了时间的维度,是路径规划的后处理。Li等^[14]提出了一种用于自动泊车的数值优化方法统一框架,将轨迹规划问题转变为非线性规划问题,证明了数值优化方法求解轨迹规划问题的可行性,但是非线性规划问题较为依赖初始值,若无有效的初始值,可能会无法求得有效解。文献[15]中采用曲线组合和数值优化相结合的方法规划了一条路径,该方法将泊车过程分为两部分并分别优化,但未能针对任意起点确定一条相应的路径。文献[16]中采用了hp自适应伪谱法求解轨迹规划问题,针对在轨迹规划问题转换后的网格与配点进行自适应调整,提高求解的成功率。Chen等^[17]采用了高斯伪谱法求解轨迹规划问题,引

入了同伦曲线,将不等式约束简化,先求解简单的轨迹规划问题,再将其作为初始值求解下一问题,而后将简化不等式约束逐渐向原始不等式约束靠近,不断迭代求解,最终获得原始问题的最优解,但简化问题与原始问题的过渡难以掌握。文献[18]中提出了一种基于先验知识的初始值策略,根据自动泊车的不同区域,将泊车的问题分为多个子问题,将简单子问题求解得到的值作为下一个子问题的初始值,从而不断迭代求解,直至子问题等于原始问题。但如果子问题分解过少,则可能无法求得有效解;如果子问题分解过多,则会降低计算效率。文献[19]中为了满足自动泊车的实时性要求,提出了一种离线计算储存的在线查表计算方法,该方法将不同初始位姿和车位下的轨迹规划离线计算出来,然后将其作为初始值储存,泊车时根据初试位姿和车位的情况,在线查表计算。Chai等^[20]采用粒子群算法获得初始值,而后求解原始轨迹规划问题,但求解时间较长。

以上的路径规划算法中,基于图搜索、采样和几何分析的方法,都只能规划一条路径,而不是轨迹,且不能得到最优性指标。而基于数值优化的轨迹规划,为了提高轨迹规划问题的求解成功率,往往需要采用多次迭代求解的方法,但是在迭代过程中需要求解多个非线性规划问题,计算效率不高。为了提高求解轨迹规划问题的成功率和计算效率,提出了一种双层求解策略来进行平行泊车的轨迹规划。第一层通过状态采样和混合A*的方法快速获得一条有效的初始路径,而后对初始路径附加时间信息,得到相应的速度、加速度等信息,将状态量和控制量在特定的时间节点离散化,将其作为第二层非线性规划问题的初始值;第二层采用同步联立正交配置法将最优控制问题转换为非线性规划问题,而后将第一层获取的初始值代入计算,求得一条满足约束条件且目标函数最优的轨迹。

1 自动泊车轨迹规划构建

1.1 车辆运动学模型及车辆自身约束

由于车辆在自动泊车入位时低速行驶,可忽略

执行结构误差和车辆侧滑的影响,故采用二轮自行车模型对其近似表示。

车辆运动学模型和车身尺寸参数如图1所示,其中 (x, y) 表示后轴中心的坐标, θ 表示车辆的航向角, v 表示后轴中心的速度, φ 表示等效前轮转角, a 表示后轴中心的加速度, ω 表示等效前轮转向角速度, L 表示车辆轴距, L_f 表示车辆前悬, L_r 表示车辆后悬, W 表示车辆整车宽度, $A(X_A, Y_A)$ 、 $B(X_B, Y_B)$ 、 $C(X_C, Y_C)$ 、 $D(X_D, Y_D)$ 分别表示车身的4个顶点。

建立大地坐标系,其中 O 为原点, X 轴和 Y 轴如图1所示,以后轴中心 (x, y) 为参考点, $(x, y, v, \theta, \varphi)$ 为车辆的状态量,表示车辆各时刻的位姿信息, (a, ω) 为车辆的控制量,选择 a 为车辆的纵向控制,方便车辆的动力学控制,选择 ω 保证了车辆运动的曲率连续性。

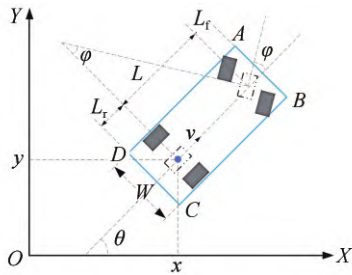


图1 车辆运动学模型和尺寸参数

根据车辆的运动学关系得到车辆运动学模型,如式(1)所示:

$$\begin{cases} \dot{x}(t) = v(t) \cos(\theta(t)) \\ \dot{y}(t) = v(t) \sin(\theta(t)) \\ \dot{v}(t) = a(t) \\ \dot{\theta}(t) = \frac{v(t) \tan \varphi(t)}{L} \\ \dot{\varphi}(t) = \omega(t) \end{cases} \quad (1)$$

车辆在泊车过程中,会受到自身机械结构限制,其 (v, a, φ, ω) 的取值范围如式(2)所示:

$$\begin{cases} v_{\min} \leq v \leq v_{\max} \\ a_{\min} \leq a \leq a_{\max} \\ \varphi_{\min} \leq \varphi \leq \varphi_{\max} \\ \omega_{\min} \leq \omega \leq \omega_{\max} \end{cases} \quad (2)$$

1.2 车辆无碰撞约束

泊车过程中,车辆可能会与周围的障碍物发生碰撞剐蹭,因此需要知道车身的尺寸参数。有了车辆各个顶点的坐标后,如何避免与障碍物的碰撞是轨迹规划的一个关键问题。采用面积法^[14],对任意

多边形以及任意点,可以用该点与多边形各个顶点的连线组成多个三角形,若多个三角形的面积累加大于原多边形的面积,则说明该点在多边形外(对于交叉重叠的情况,最开始也是从一个顶点开始侵入),反之说明该点在多边形的各条边上或多边形内,即发生了碰撞。

根据图1中车辆的几何关系得到车辆各个顶点的坐标:

$$\begin{cases} XA = x + (L + L_f) \cos \theta - \frac{W}{2} \sin \theta \\ YA = y + (L + L_f) \sin \theta + \frac{W}{2} \cos \theta \\ XB = x + (L + L_f) \cos \theta + \frac{W}{2} \sin \theta \\ YB = y + (L + L_f) \sin \theta - \frac{W}{2} \cos \theta \\ XC = x - L_r \cos \theta + \frac{W}{2} \sin \theta \\ YC = y - L_r \sin \theta - \frac{W}{2} \cos \theta \\ XD = x - L_r \cos \theta - \frac{W}{2} \sin \theta \\ YD = y - L_r \sin \theta + \frac{W}{2} \cos \theta \end{cases} \quad (3)$$

车辆的泊车过程以及车位尺寸信息如图2所示。

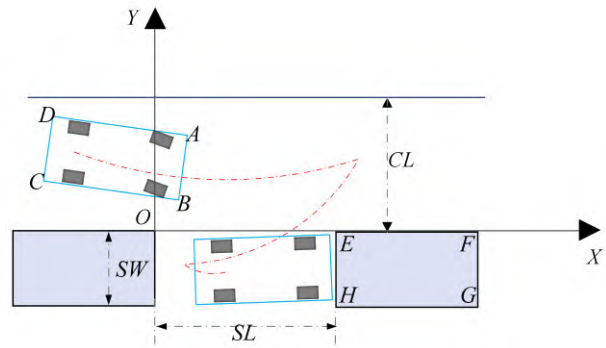


图2 泊车过程与车位尺寸示意图

以车辆不与右侧多边形碰撞为例,车辆的无碰撞约束可表示为

$$\begin{cases} S_{MAB} + S_{MBC} + S_{MCD} + S_{MDA} > S_{ABCD}, M \in (E, F, G, H) \\ S_{NEF} + S_{NFG} + S_{NGH} + S_{NHE} > S_{EFGH}, N \in (A, B, C, D) \end{cases} \quad (4)$$

1.3 两点边界约束

两点边界约束分为车辆起始状态约束和终端状态约束。满足式(1)~式(4)的情况下,规划从 t_0 时刻到 t_f 时刻的无碰撞泊车轨迹,车辆的起始状态如式(5)所示:

$$\begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \\ \theta(t_0) = \theta_0 \\ v(t_0) = v_0 \\ \varphi(t_0) = \varphi_0 \end{cases} \quad (5)$$

车位尺寸如图2所示,取大地坐标系, X 轴为车位的右方, Y 轴为车位的上方,原点位于左侧障碍物的右上角。其中车位长为 SL ,车位宽为 SW ,车辆道路宽度为 CL 。 (E, F, G, H) 为右侧矩形障碍物的4个顶点坐标。泊车入位的要求是车辆的4个顶点都在车位空间,且车辆的运动参数 (v, a, φ, ω) 都为0,车辆的终端状态表示如式(6)所示:

$$\begin{cases} X_A(t_f) \in [\sigma, SL - \sigma] \\ X_B(t_f) \in [\sigma, SL - \sigma] \\ X_C(t_f) \in [\sigma, SL - \sigma] \\ X_D(t_f) \in [\sigma, SL - \sigma] \\ Y_A(t_f) \in [-SW + \sigma, \sigma] \\ Y_B(t_f) \in [-SW + \sigma, \sigma] \\ Y_C(t_f) \in [-SW + \sigma, \sigma] \\ Y_D(t_f) \in [-SW + \sigma, \sigma] \end{cases} \text{ and } \begin{cases} v(t_f) = 0 \\ a(t_f) = 0 \\ \varphi(t_f) = 0 \\ \omega(t_f) = 0 \end{cases} \quad (6)$$

式中 σ 为容许误差,用以表示发生碰撞的安全余量。

1.4 最优轨迹规划

由1.1节到1.3节的信息,可以规划多条从起始状态到终止状态的轨迹,为此需要从中选择目标函数最优的轨迹,本质上是一个最优控制问题。其目标函数是根据实际情况所定义的性能指标函数,约束方程为车辆的运动学约束、车辆自身约束、无碰撞约束以及两点边界约束。性能指标的选择可以是为了保证泊车的时间更短,将其设置为与时间相关;或者为了使得控制量变化更平滑,添加与控制量变化相关的性能指标。以上最优轨迹规划可表示为式(7),其中 λ_1, λ_2 分别表示各性能指标的权重。

$$\begin{cases} \min \lambda_1 \Phi(t_f) + \lambda_2 \int_{t_0}^{t_f} g(x, u, t) dt \\ \text{s. t. 车辆运动学约束式(1)} \\ \text{车辆自身约束式(2)} \\ \text{无碰撞约束式(3)和式(4)} \\ \text{两点边界约束式(5)和式(6)} \end{cases} \quad (7)$$

2 最优轨迹规划转换

最优轨迹规划是一个最优控制问题,其求解方法有间接法和直接法^[21]。直接法直接对最优控制问

题离散化处理,可选择将控制变量离散,或者将控制变量和状态变量同步联立离散化。本文采用将控制变量和状态变量同步联立离散化,其离散点采用Radau正交配点,进而使用内点法求解转换后的非线性规划问题^[22]。

为不失一般性,式(7)可表示为

$$\begin{cases} \min \lambda_1 \Phi(t_f) + \lambda_2 \int_{t_0}^{t_f} g(X, U, t) dt \\ \text{s. t. } \dot{X} = f(X, U, t) \\ B(X, U, t) = 0 \\ C(X, U, t) < 0 \end{cases} \quad (8)$$

式中: X 为状态向量; U 为控制向量。最优控制问题转化为非线性规划问题的步骤如下。

步骤1:划分时间间隔

首先,需要将整个时域 $[t_0, t_f]$ 等分成 NM 份时长相等的 $[t_{i-1}, t_i]$ ($i=1, 2, \dots, NM$),因此,每一份的时间间隔为

$$t_i - t_{i-1} = \frac{t_f}{NM}, i = 1, 2, \dots, NM \quad (9)$$

步骤2:Radau配点

在离散化时,采用Radau配点,在区间 $[0, 1]$ 上时,当正交多项式的阶次为 K 时,Radau的配点为 $\frac{d^{K-1}}{d\tau^{K-1}}(\tau^K(\tau-1)^{K-1})$ 的零点,具体取值如式(10)所示:

$$\tau = \arg\left(\frac{d^{K-1}}{d\tau^{K-1}}(\tau^K(\tau-1)^{K-1}) = 0\right) \quad (10)$$

步骤3:变量离散化

在时间间隔 $[t_{i-1}, t_i]$ ($i=1, 2, \dots, NM$)上将状态向量和控制向量在配点处离散化,有 $(K+1)$ 个状态向量 $(X_{i,0}, X_{i,1}, \dots, X_{i,K})$ 和 K 个控制向量 $(U_{i,1}, U_{i,2}, \dots, U_{i,K})$ 。为了保证状态的连续性,需添加连续条件,如式(11)所示:

$$\begin{cases} X_{1,0} = X_0 \\ X_{i,k} = X_{i+1,0} \\ i = 1, 2, \dots, NM - 1 \end{cases} \quad (11)$$

状态变量和控制变量可以利用Lagrange插值多项式逼近,如式(12)所示:

$$\begin{cases} X(t) = \sum_{j=0}^K L_j(\tau) X_{i,j} \\ U(t) = \sum_{j=1}^K \tilde{L}_j(\tau) U_{i,j} \\ t = t_{i-1} + (t_i - t_{i-1})\tau, \tau \in [0, 1] \end{cases} \quad (12)$$

式中 $X_{i,j}, U_{i,j}$ 分别表示第 i 个时间间隔中第 j 个配点

的状态向量和控制向量。Lagrange 插值多项式如式(13)所示:

$$\begin{cases} L_j(\tau) = \prod_{k=0, k \neq j}^K \frac{\tau - \tau_k}{\tau_j - \tau_k}, & L_j(\tau_k) = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases} \\ \tilde{L}_j(\tau) = \prod_{k=1, k \neq j}^K \frac{\tau - \tau_k}{\tau_j - \tau_k}, & \tilde{L}_j(\tau_k) = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases} \end{cases} \quad (13)$$

由式(12)和式(13)联立可求得微分方程的近似表达,如式(14)所示:

$$\sum_{k=0}^K \left(\frac{d \left(\prod_{m=0, m \neq j}^K \frac{\tau - \tau_m}{\tau_j - \tau_m} \right)}{d\tau} \right) \Big|_{(\tau=\tau_k)} \cdot X_{i,k} = (t_i - t_{i-1}) \cdot f(X_{i,j}, U_{i,j}) \quad (14)$$

式中 $X_{i,j}$ 、 $U_{i,j}$ 是需要求解的离散点,状态向量的导数仅与拉格朗日基函数在 τ 点的导数值有关。

步骤4:积分形式转换

积分性能指标 $\int_{t_0}^{t_f} g(X, U, t) dt$ 可转化为状态量与

控制量离散求和形式,性能指标可转化为

$$\int_{t_0}^{t_f} g(x, u, t) dt = \sum_{i=1}^{NM} \sum_{j=1}^K g(X_{i,j}, U_{i,j}) \quad (15)$$

式中 $X_{i,j}$ 、 $U_{i,j}$ 分别表示第 i 个时间间隔中第 j 个配点的状态向量和控制向量。

步骤5:问题转换

综合式(9)~式(15),问题式(8)可以转换为

$$\begin{cases} \min & \lambda_1 \Phi(t_f) + \lambda_2 \left(\sum_{i=1}^{NM} \sum_{j=1}^K g(X_{i,j}, U_{i,j}) \right) \\ \text{s. t.} & \sum_{k=0}^K \left(\frac{d \left(\prod_{m=0, m \neq j}^K \frac{\tau - \tau_m}{\tau_j - \tau_m} \right)}{d\tau} \right) \Big|_{(\tau=\tau_k)} \cdot X_{i,k} = (t_i - t_{i-1}) \cdot f(X_{i,j}, U_{i,j}) \\ & B(X_{i,j}, U_{i,j}) = 0 \\ & C(X_{i,j}, U_{i,j}) < 0 \end{cases} \quad (16)$$

求解式(16)即可得到一条目标函数最优的轨迹。式(16)是一个高维的非线性规划问题,对初始值较为敏感,若无初始值,直接求解会存在解的质量不高甚至无法求解的情况,为此,提出了一种提供初始解的双层求解策略,具体流程见第3节。

3 双层求解策略

本节提出了一种双层求解策略,具体为:先规划一条初始路径,而后再将粗糙路径的状态量和控制量

在特定节点离散化。将离散化的状态量和控制量作为式(16)的初始值代入求解。

3.1 双层求解策略的整体框架结构

在泊车入位时,车位以外的区域较为宽敞,称该空间为非风险区;车位空间较为狭小,称之为风险区,泊车是从非风险区到风险区的过程。

若仅仅使用混合 A* 规划从初始位姿到终止位姿的路径,由于风险区的空间较小,混合 A* 需要将栅格地图的分辨率设置的非常小,导致计算所需的内存较大、时间较长,甚至寻径失败。为了提高自动泊车的效率与可行性,将整个泊车的过程分为两个阶段,即非风险区的路径与风险区的路径。

在非风险区采用混合 A* 规划路径,在风险区采用状态采样的方法规划路径,两条路径的交点称之为锚点,锚点的位姿决定了初始路径的质量。由于非风险区路径规划对锚点的位姿要求更低,即非风险区的路径规划比风险区的路径规划更容易,因此将非风险区的路径在风险区路径规划之后再规划。风险区、非风险区及锚点示意图如图3所示。

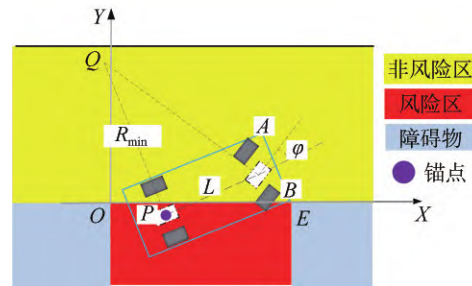


图3 泊车区域及锚点示意图

采用逆推的思路,首先规划一条从风险区内目标点(终止点)到锚点的路径,而后再规划一条从锚点到非风险区内目标点(起始点)的路径。将两段路径整合,加入“时间最优”剖面,即假定车辆能以当前时刻的最快速度行驶,在路径确定的情况下,可以得到相应的最短时间,而后再得到与时间相对应的状态量和控制量,在特定节点(radau 节点)离散化,逆序输出得到相应的状态量和控制量,再将其作为初始值代入式(16)求解。平行泊车轨迹规划及双层求解策略的具体框架如图4所示。

3.2 锚点的选择及风险区路径规划

锚点作为连接风险区与非风险区的关键点,其位姿会影响寻径的成功率和效率,锚点须兼顾两边的路径,从而使得路径总体上达到最优。锚点的必要条件是汽车前端或者后端的两个顶点同时在非风

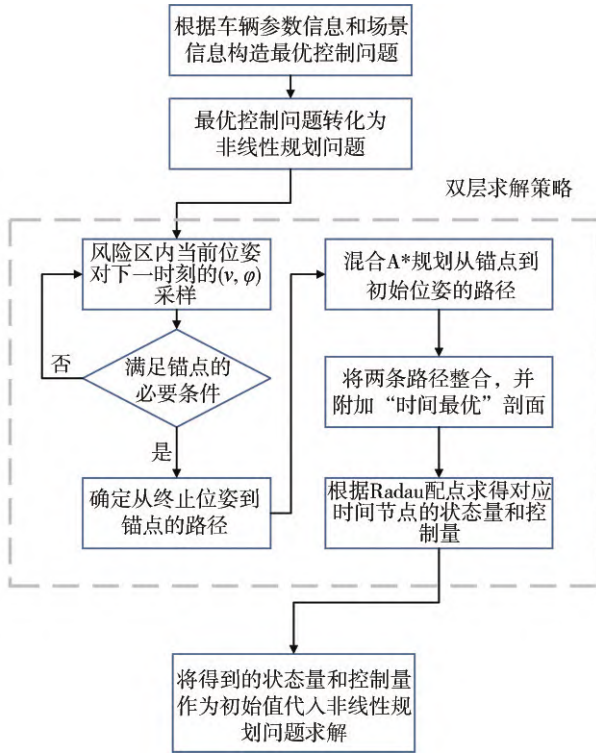


图4 平行泊车轨迹规划和双层求解策略框架

险区,且车身与障碍物没有发生碰撞。采用逆推的思路,即由车辆出库路径逆推泊车路径,当某一路径点满足锚点的必要条件时,该路径点即为所找的锚点。

假定车辆停在车位中,车辆的4个顶点及其运动状态都满足式(6),在驶出车位时,采用“时间最优”(车辆以当前时刻下最大车速运动,进而推导出各个状态量和控制量)方法,对车辆的下一时刻的 (v, φ) 进行采样,车辆的状态量由式(1)求出。在风险区时,车辆的无碰撞约束式(4)可以转换为式(17),其中 σ 表示容许误差,该约束方程的转换可以简化原始高维非线性的无碰撞约束,从而提高计算效率。

$$\begin{cases} \sigma < X_A < SL - \sigma \\ \sigma < X_B < SL - \sigma \\ \sigma < X_C < SL - \sigma \\ \sigma < X_D < SL - \sigma \end{cases} \text{ and } \begin{cases} Y_A \geq -SW + \sigma \\ Y_B \geq -SW + \sigma \\ Y_C \geq -SW + \sigma \\ Y_D \geq -SW + \sigma \end{cases} \quad (17)$$

由于风险区的空间较小,车辆可能需要往复的前进后退,为避免碰撞,以车辆前进时为例,车辆的车速 v 需要满足(后退同理)式(18)。

$$\begin{cases} v_p = v_{i-1} + a_{\max} T_s \\ v_d = \sqrt{2a_{\max} d_{\text{dist}} / \cos \theta} - a_{\max} T_s \\ v = \min(v_p, v_d, v_{\max}) \end{cases} \quad (18)$$

式中: v_{i-1} 表示当前车速; v_p 表示下一采样时刻预测最

大速度; v_d 表示为避免碰撞的最大速度; d_{dist} 表示前进方向上车身的前端顶点与障碍物的最近距离; T_s 为采样时间。

为了保证车辆能够驶出,车辆的前轮转角需要在车辆停止时处于回正的姿态(若不回正,则会导致车辆一直在车位里往复运动,若等车停了再回正,则会加剧轮胎的磨损),当前轮转角处于 φ 时,车轮最短回正时间为 $T_\varphi = \frac{|\varphi|}{\omega_{\max}}$,车辆最短停车时间 $T_v = \frac{|v|}{a_{\max}}$,车辆的前轮转角 φ 须满足式(19)。

$$\begin{cases} \varphi_p = \varphi_{i-1} + \omega_{\max} T_s (T_\varphi < T_v) \\ \varphi_p = \varphi_{i-1} - \omega_{\max} T_s (T_\varphi \geq T_v) \\ \varphi = \min(|\varphi_p|, \varphi_{\max}) \end{cases} \quad (19)$$

式中 φ_p 为下一采样时刻的最大前轮转角,由 (v, φ) 可以求出各状态量和控制量。若在前进的某一时刻,有式(20):

$$\begin{cases} R_{\min} = \frac{L}{\tan \varphi_{\max}} \\ X_Q = x - R_{\min} \sin \varphi_{\max} \\ Y_Q = y + R_{\min} \cos \varphi_{\max} \\ R_{B\min} = (X_Q - X_B)^2 + (Y_Q - Y_B)^2 \end{cases} \quad (20)$$

同时满足 $(X_Q - X_E)^2 + (Y_Q - Y_E)^2 > R_{B\min}$

其中: R_{\min} 为后轴中心的最小转弯半径; $R_{B\min}$ 表示车辆顶点B的最小转弯半径; (X_Q, Y_Q) 为车辆转向时的圆心坐标; (X_E, Y_E) 为右侧障碍物的顶点坐标。此时打死转向盘($\varphi = \varphi_{\max}$),则当 (Y_A, Y_B) 同时大于0时,该时刻后轴中心 (x, y, θ) 所在的位置,即图3中的P点称为锚点。后退时,锚点所求方式与前进同理,从停车位到锚点对 (v, φ) 采样所生成的路径即为风险区的路径规划。

3.3 非风险区路径规划

在非风险区的路径规划上,采用混合A*寻径^[23]。混合A*是A*算法与RS曲线的结合,RS曲线将车辆所有的直线段与圆弧段排列组合,一共有48种,是衔接一条起始位姿与终止位姿的最短路径^[24]。混合A*与A*算法的最大区别是扩张子节点的方式不一样。如图5所示,传统的A*算法扩张的子节点有8个,在扩张下一个子节点时没有考虑车辆的运动学特性,会造成无法跟踪的现象,而混合A*扩张的子节点有6个(前进后退下的直行,左转,右转),满足车辆的运动学,并每隔一定的迭代次数会进行RS曲线的验证,若验证的RS曲线满足无碰撞约束,则直接退出迭代,返回RS曲线得到的路径。在混合

A*规划路径后,该路径只有位姿信息,没有与时间相关的信息,需要对其进行后处理,加入“时间最优”剖面(与上文“时间最优”方法一致),得到特定时间节点的状态量和控制量。

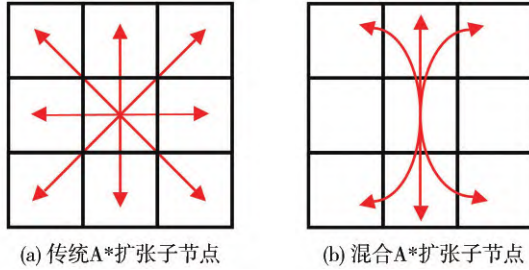


图5 传统A*与混合A*扩张子节点

4 仿真结果与分析

仿真在CPU为AMD Ryzen 4600H上运行,其中非线性规划的建模软件为AMPL,求解包为IPOPT(线性求解器为mumps和HSL_MA57)。

在评价轨迹的生成质量时,可以根据实际需求定义相关的目标函数,本文选择泊车时间以及控制量的变化幅度为评价指标。4.2节研究了使用状态采样和混合A*结合所生成的轨迹,以及双层求解策略所生成的轨迹。4.3节研究了在几种具有代表性的泊车起始位姿和狭窄车位下的轨迹规划,以此体现基于双层求解策略的轨迹规划的泛化能力;在文中的5种场景下,与文献[18]就成功率和计算效率进行了对比。

4.1 车辆及场景信息

以某款SUV车型为研究对象,车辆参数如表1所示。场景一的起始状态和车位尺寸如表2所示。(车辆自身约束式(2)的其最大值和最小值可以根据实际场景的需要设定不同的值,本文中以 $v_{\min}=-v_{\max}$, $a_{\min}=-a_{\max}$, $\varphi_{\min}=-\varphi_{\max}$, $\omega_{\min}=-\omega_{\max}$ 为例),其中将总时间等分为 $NM=20$ 份,正交多项式的阶次 $K=3$ 。

4.2 原始轨迹与双层求解策略求解轨迹

4.2.1 “最优时间”剖面的轨迹生成与目标函数为泊车时间的轨迹生成

结合表1和表2的参数,运用混合A*与状态采样相结合的方法规划一条初始路径,在该路径下加入了“时间最优”剖面后,其轨迹如图6所示,泊车时间为35.20 s。

表1 车辆参数

参数	符号	单位	数值
车辆轴距	L	m	2.830
车辆前悬	L_f	m	1.006
车辆后悬	L_r	m	1.070
车辆宽度	W	m	1.862
泊车最大车速	v_{\max}	m/s	1.0
泊车最大加速度	a_{\max}	m/s ²	0.5
车辆最大前轮转角	φ_{\max}	rad	0.576
车辆最大前轮转向角速度	ω_{\max}	rad/s	0.576

表2 场景一的车辆起始状态和车位尺寸

参数	符号	单位	数值
车辆起始位置(水平方向)	x_{i0}	m	-5
车辆起始位置(垂直方向)	y_{i0}	m	2
车辆起始航向角	θ_{i0}	rad	0
车辆起始速度	v_{i0}	m/s	0
车辆起始前轮转角	φ_{i0}	rad	0
车位长	SL	m	6
车位宽	SW	m	2.5
道路宽度	CL	m	4

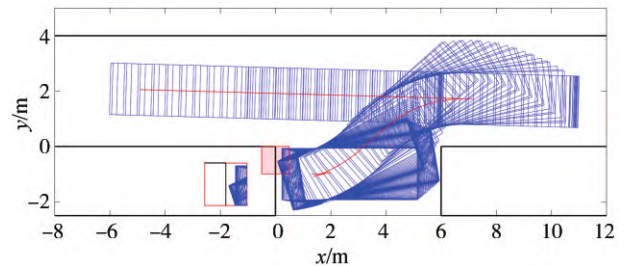


图6 “时间最优”剖面下的轨迹($t_f=35.20$ s)

采用双层求解策略,目标函数设定为泊车时间最短,规划出来的轨迹如图7所示,泊车时间为24.20 s。

采用双层求解策略后,泊车时间从35.20 s减少

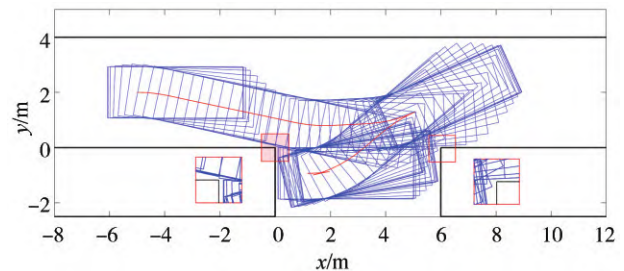


图7 时间最优下的轨迹($t_f=24.20$ s)

至 24.20 s,节省了 30% 左右的时间(在本节所示的初始位置下节省的时间),即采用双层求解策略可以减少泊车时间,提高轨迹的生成质量。

为便于观察,将原始轨迹与双层求解策略求得的轨迹进行对比,如图 8 所示。

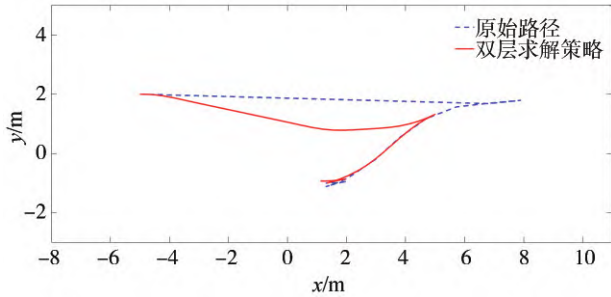


图 8 原始轨迹与双层求解策略轨迹对比图

由图 8 可以看出,双层求解策略规划的轨迹更能充分利用场景,从而得到一条更优的轨迹。

4.2.2 目标函数为时间与控制量变化加权和最小的轨迹生成

图 7 是性能指标为最短泊车时间的情况下所产生的轨迹。当目标函数是其他指标时,采用双层求解策略也能求得,如在行驶的过程中,除了泊车时间更短外,往往希望轨迹更容易被跟踪,执行机构的磨损更小,因此可以添加控制量变化幅度作为性能指

标,目标函数变为式(21),其中 λ_1 、 λ_2 分别表示各性能指标的权重:

$$\min \lambda_1 t_f + \lambda_2 \left(\sum_{i=2}^{NM} [(a_{i,1} - a_{i-1,K})^2 + (\omega_{i,1} - \omega_{i-1,K})^2] + \sum_{i=1}^{NM} \sum_{j=1}^{K-1} [(a_{i,j+1} - a_{i,j})^2 + (\omega_{i,j+1} - \omega_{i,j})^2] \right) \quad (21)$$

当 $\lambda_1 = \lambda_2 = 1$ 时,泊车时间为 34 s,采用双层求解策略求得的轨迹如图 9 所示。

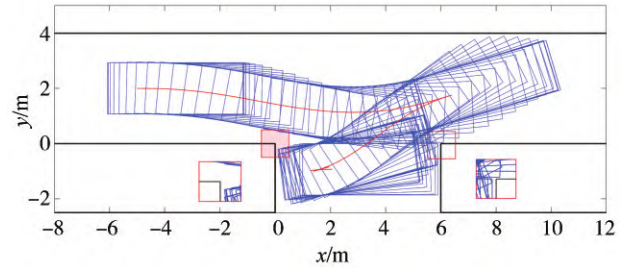


图 9 时间与控制量变化加权和最小的轨迹($t_f=34$ s)

图 9 和图 7 的生成的路径较为相似,但两者的泊车时间却差别较大,这是因为两者的目标函数不一样所致,即双层求解策略是一种时空耦合的算法,可以同时求得轨迹的状态量和控制量。

各轨迹下的状态量和控制量的变化情况如图 10~图 12 所示。

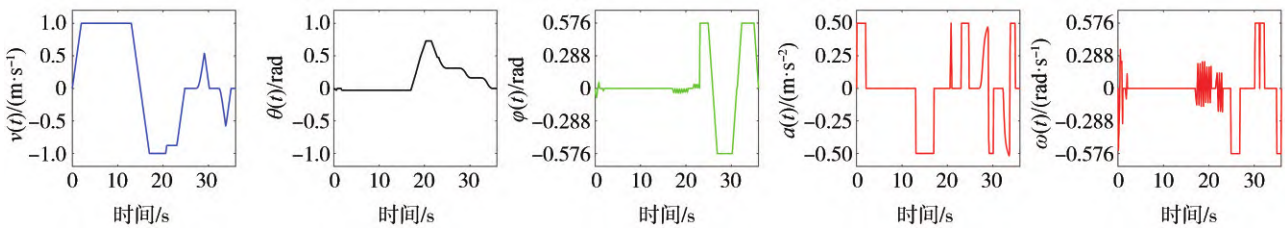


图 10 “时间最优”剖面下的状态量和控制量变化关系($t_f=35.20$ s)

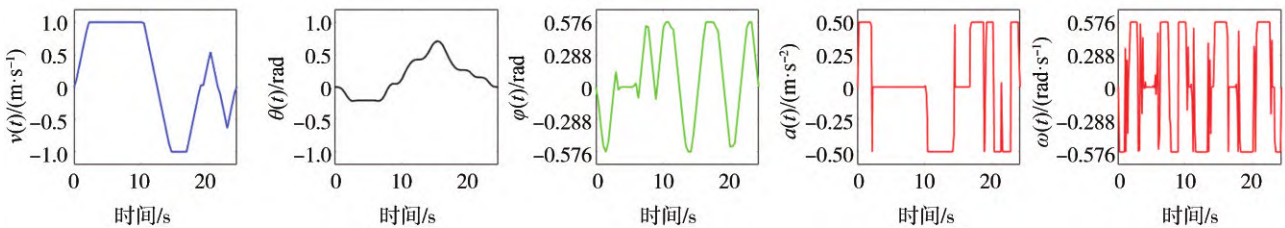
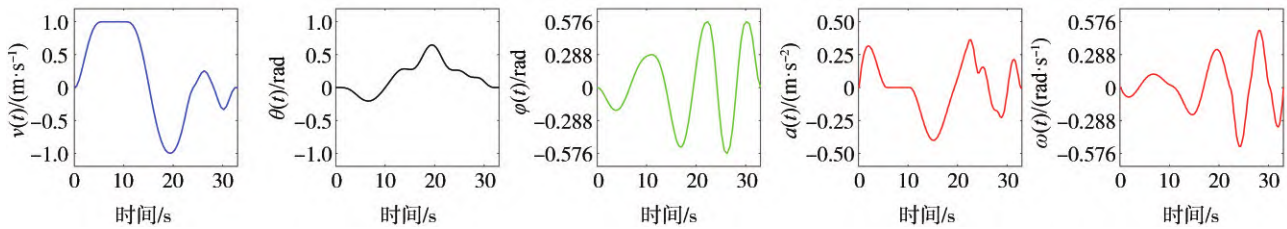


图 11 时间最优下的状态量和控制量变化关系($t_f=24.20$ s)

图 10 是状态采样和混合 A* 结合生成的轨迹在“时间最优”剖面下的状态量和控制量变化关系,图 11 和图 12 则分别是双层求解策略在时间最优下的

状态量和控制量变化关系、时间与控制量变化加权和最小的状态量和控制量的变化关系。在图 10 和图 11 中,控制变量(a , ω)呈现“bang-bang”控制模

图12 时间与控制量变化加权和最小的状态量和控制量变化关系($t_f=34$ s)

式,这是最短时间内跟踪轨迹的常见结构,间接证明了双层求解策略求得轨迹的有效性。然而车辆频繁加减速,会加剧执行机构的磨损,甚至出现无法跟踪轨迹的情况,因此,可以在原目标函数中加入控制量变化最小的性能指标,如式(21)所示。由图12可以看出,控制变量(a , ω)变化更为平缓,减轻了执行机构的负担,车辆更容易跟踪轨迹。双层求解策略可以根据需求自定义目标函数,而后求解得到一条目标函数最优的轨迹,与状态采样和混合A*结合所生成的轨迹相比,所提出的方法能够提高轨迹的生成质量。

4.3 不同起始位姿和车位下的轨迹规划

为了验证双层求解策略的泛化能力,选取几种具有代表性的起始位姿和车位尺寸,场景二表示在起始位姿车位正上方的情况,场景三表示起始位置已经越过了车位正上方的情况,即场景一至场景三分析了车位外的几种典型场景,场景四和场景五表示

狭窄车位的情况,其中场景五的车位长度为5.60 m,而选用车型的车身对角长度为5.25 m,车位长与车身对角长的比值仅为1.067。场景二至场景五均为以车辆的泊车入库时间最少为目标函数。具体起始位姿和车位尺寸数值如表3所示。

以上场景(2-4)所对应的轨迹如图13所示。

由图7和图13可以看出,对于3种典型的车位外场景和3种不同大小的车位尺寸,在车辆起始的航向角不同的情况下,双层求解策略均能规划出一条有效的轨迹,表明了该方法对轨迹规划具有一定的泛化能力。

同时,为了验证双层求解策略的成功率和计算效率,与文献[18]所述的方法进行了对比。文献[18]中提供初始值的方法是将原问题需要求解的未知量分为多个区域,先求解部分区域,而后不断更新迭代求解,逼近原问题,该方法是将原问题化为简单的子问题,而后求解子问题进而达到求解原问题的

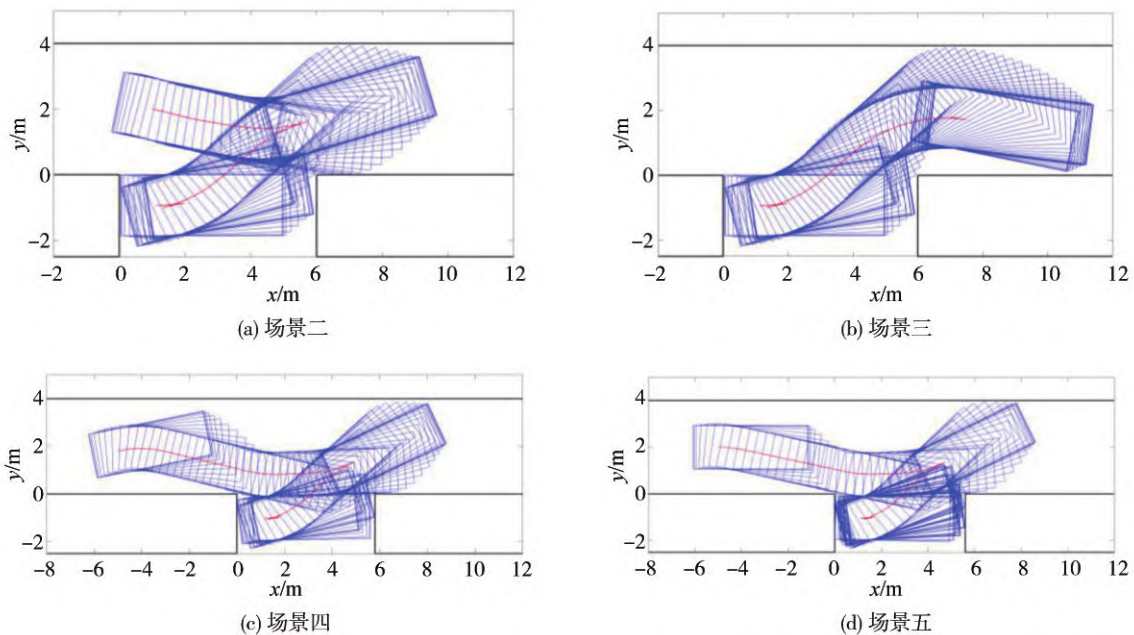


图13 不同场景下的平行泊车轨迹

表3 不同起始位姿和车位尺寸

场景	起始位姿和车位尺寸
场景二	$\begin{cases} x_{i0} = 1 \\ y_{i0} = 2 \\ \theta_{i0} = -0.2 \text{ and } \begin{cases} CL = 4 \\ SL = 6 \\ SW = 2.5 \end{cases} \text{ and } \begin{cases} a_{i0} = 0 \\ w_{i0} = 0 \end{cases} \\ v_{i0} = 0 \\ \varphi_{i0} = 0 \end{cases}$
场景三	$\begin{cases} x_{i0} = 7 \\ y_{i0} = 1.8 \\ \theta_{i0} = -0.2 \text{ and } \begin{cases} CL = 4 \\ SL = 6 \\ SW = 2.5 \end{cases} \text{ and } \begin{cases} a_{i0} = 0 \\ w_{i0} = 0 \end{cases} \\ v_{i0} = 0 \\ \varphi_{i0} = 0 \end{cases}$
场景四	$\begin{cases} x_{i0} = -5 \\ y_{i0} = 1.8 \\ \theta_{i0} = 0.2 \text{ and } \begin{cases} CL = 4 \\ SL = 5.8 \\ SW = 2.5 \end{cases} \text{ and } \begin{cases} a_{i0} = 0 \\ w_{i0} = 0 \end{cases} \\ v_{i0} = 0 \\ \varphi_{i0} = 0 \end{cases}$
场景五	$\begin{cases} x_{i0} = -5 \\ y_{i0} = 2 \\ \theta_{i0} = 0 \text{ and } \begin{cases} CL = 4 \\ SL = 5.6 \\ SW = 2.5 \end{cases} \text{ and } \begin{cases} a_{i0} = 0 \\ w_{i0} = 0 \end{cases} \\ v_{i0} = 0 \\ \varphi_{i0} = 0 \end{cases}$

经典求解思路,需要求解多次非线性规划问题,耗时较长;而本文所提出的双层求解策略是直接给特定节点的值代入求解,只需要求解一次非线性规划问题。在双层求解策略中,初始路径 CPU time 表示寻得初始路径所需要的计算时间, CPU time 表示有初始值输入后的 CPU 计算时间。具体对比数据如表4所示。

表4 不同求解策略下求解轨迹规划问题的对比

方法	没有初值	双层求解策略				文献[18]		
	CPU time	初始路径 CPU time	CPU time	CPU time*	泊车 时间	CPU time	CPU time*	泊车 时间
场景一	失败	0.311 s	5.725 s	0.866 s	24.20 s	失败	6.524 s	25.56 s
场景二	失败	0.173 s	5.226 s	0.758 s	18.94 s	失败	失败	失败
场景三	失败	0.102 s	5.318 s	0.792 s	16.22 s	失败	7.112 s	25.96 s
场景四	失败	0.241 s	5.882 s	0.924 s	25.21 s	失败	9.189 s	25.67 s
场景五	失败	0.272 s	6.021 s	1.198 s	32.23 s	失败	失败	失败

注:以上数据为每个场景下重复5次实验所取的平均值, CPU time 表示线性求解器为 mumps 时 CPU 求解时间, CPU time* 表示线性求解器为 HSL_MA57 时 CPU 求解时间, 泊车时间表示该方法下的最短泊车时间, 当达到最大迭代次数3 000次, 还没求解出有效值, 则视为求解失败, 其中文献[18]中求解失败仅代表在将原始问题分解为20个子问题时求解失败。

参考文献

[1] ESKANDARIAN A, WU C, SUN C. Research advances and challenges of autonomous and connected ground vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22

由表4可得,文献[18]中所述的方法对求解器的要求比双层求解策略更为严格。双层求解策略使用 IPOPT(线性求解器为 mumps)处理文中所列的5种场景时,均能有效求解,而文献[18](子问题分解为20个)在使用 IPOPT(线性求解器为 mumps)处理文中所列5种场景时均无法求出有效路径。在求解器升级后(线性求解器由 mumps 转换为 HSL_MA57),双层求解策略的计算效率得到明显提高。

5 结论

本文构建了平行泊车的轨迹规划问题,采用同步联立正交配置法将其在特定节点离散化,转变成非线性规划问题。而后提出双层求解策略,规划一条由锚点相连接的初始路径,将其作为初始值代入非线性规划中,降低问题的求解难度。在 AMPL 上建模并求解非线性规划,结果表明,双层求解策略可以规划出目标函数最优的轨迹,提高了轨迹的生成质量,同时在几类具有代表性的起始位姿和狭窄车位尺寸下均能规划出轨迹,具备一定的泛化能力,并且成功率和计算效率相比于原始非线性规划问题和迭代获取初始值的方式都得到了一定的提升。

下一阶段工作将研究基于双层求解策略的轨迹规划在垂直车位场景和斜型车位场景中的应用,同时分析泊车执行部件(转向器、驱动器、制动器)对轨迹规划的影响,以及获取周围场景和自身位姿信息,从而进行硬件在环和实车实验。

(2): 683-711.

[2] GONZALEZ D, PEREZ J, MILANES V, et al. A review of motion planning techniques for automated vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 1135-1145.

- [3] 任秉韬,王浙浙,邓伟文,等. 基于混合A*和可变半径RS曲线的自动泊车路径优化方法[J]. 中国公路学报, 2022, 35(7): 317-327.
REN Bingtao, WANG Xixi, DENG Weiwen, et al. Path optimization on algorithm for automatic parking based in hybrid A* and reeds-sheep curve with variable radius[J]. China Journal of Highway and Transport, 2022, 35(7): 317-327.
- [4] HE J, LI H. Fast A* anchor point based path planning for narrow space parking[C]. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). Indianapolis, IN, USA: IEEE, 2021: 1604-1609.
- [5] HAN L, DO Q H, MITA S. Unified path planner for parking an autonomous vehicle based on RRT[C]. 2011 IEEE International Conference on Robotics and Automation. Shanghai, China: IEEE, 2011: 5622-5627.
- [6] DONG Y, ZHONG Y, HONG J. Knowledge-biased sampling-based path planning for automated vehicles parking[J]. IEEE Access, 2020, 8: 156818-156827.
- [7] 胡勤明,王金刚,张小俊. 五次多项式优化的平行泊车路径规划[J]. 计算机工程与应用, 2022, 58(14): 291-298.
HU Qinming, WANG Jingang, ZHANG Xiaojun. Optimized parallel parking path planning based on quintic polynomial[J]. Computer Engineering and Applications, 2022, 58(14): 291-298.
- [8] CHAND A, KAWANISHI M, NARIKIYO T. Application of sigmoidal gompertz curves in reverse parallel parking for autonomous vehicles[J]. International Journal of Advanced Robotic Systems, 2015, 12(9): 1-9.
- [9] 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 Jiaxu, ZHAO Jian, SHI Zhengtang, 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.
- [10] 李红,王文军,李克强. 基于B样条理论的平行泊车路径规划[J]. 中国公路学报, 2016, 29(9): 143-151.
LI Hong, WANG Wenjun, LI Keqiang. Path planning for parallel parking based on B spline theory[J]. China Journal of Highway and Transport, 2016, 29(9): 143-151.
- [11] SUZUKI T, USAMI R, MAEKAWA T. Automatic two-lane path generation for autonomous vehicles using quartic B-spline curves[J]. IEEE Transactions on Intelligent Vehicles, 2018, 3(4): 547-558.
- [12] KHOSHNEJAD M, DEMIRLI K. Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy behavior-based controller[C]. NAFIPS 2005 - 2005 Annual Meeting of the North American Fuzzy Information Processing Society. Detroit, MI, USA: IEEE, 2005: 814-819.
- [13] CHAI R, TSOURDOS A, SAVVARIS A, et al. Design and implementation of deep neural network-based control for automatic parking maneuver process[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(4): 1400-1413.
- [14] LI B, SHAO Z. A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles[J]. Knowledge-Based Systems, 2015, 86: 11-20.
- [15] 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 Jie, ZHU Linglei, Chen Ruinan, et al. Research on parallel parking path planning method for narrow parking space[J]. Automotive Engineering, 2022, 44(7): 1040-1048.
- [16] CHAI R, SAVVARIS A, TSOURDOS A. Violation learning differential evolution-based hp-adaptive pseudospectral method for trajectory optimization of space maneuver vehicle[J]. IEEE Transactions on Aerospace and Electronic Systems, 2017, 53(4): 2031-2044.
- [17] CHEN C, WU B, XUAN L, et al. A trajectory planning method for autonomous valet parking via solving an optimal control problem[J]. Sensors, 2020, 20(22): 6435.
- [18] LI B, ZHANG Y, SHAO Z. Spatio-temporal decomposition: a knowledge-based initialization strategy for parallel parking motion optimization[J]. Knowledge-Based Systems, 2016, 107: 179-196.
- [19] LI B, WANG K, SHAO Z. Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(11): 3263-3274.
- [20] CHAI R, TSOURDOS A, SAVVARIS A, et al. Two-stage trajectory optimization for autonomous ground vehicles parking maneuver[J]. IEEE Transactions on Industrial Informatics, 2019, 15(7): 3899-3909.
- [21] RAO Anil. A survey of numerical methods for optimal control[J]. Advances in the Astronautical Sciences, 2010. 135.
- [22] BIEGLER L T. Nonlinear programming: concepts, algorithms, and applications to chemical processes[M]. Society for Industrial and Applied Mathematics, 2010.
- [23] DOLGOV D, THRUN S, MONTEMERLO M, et al. Path planning for autonomous vehicles in unknown semi-structured environments[J]. The International Journal of Robotics Research, 2010, 29(5): 485-501.
- [24] REEDS J, SHEPP L. Optimal paths for a car that goes both forwards and backwards[J]. Pacific Journal of Mathematics, 1990, 145(2): 367-393.