

基于混合 A* 算法的自动驾驶汽车轨迹生成

周苏^{1,2} 王若伊¹ 张友发² 张岗¹

(1. 同济大学汽车学院, 上海 201804; 2. 同济大学中德学院, 上海 201804)

【欢迎引用】周苏, 王若伊, 张友发, 等. 基于混合 A* 算法的自动驾驶汽车轨迹生成[J]. 汽车文摘, 2023(2): 44-54.

【Cite this paper】ZHOU S, WANG R Y, ZHANG Y F, et al. Hybrid A* Algorithm-Based Trajectory Generation for Automated Driving Vehicles[J]. Automotive Digest (Chinese), 2023(2): 44-54.

【摘要】自动驾驶汽车混合 A* 算法运行迅速, 但是难以保证曲率的连续性和汽车舒适度, 并且不能保证优化后的结果无碰撞。针对以上问题, 以提高汽车舒适性为目的, 基于最小冲击度对 A* 算法进行优化, 在 MATLAB 和栅格地图中进行了算法预研, 解决了低速场景下自动驾驶汽车运动过程中舒适度不足的问题。然后, 针对优化算法可能导致轨迹碰撞的问题, 对优化算法进行了改进, 保证了汽车的安全性, 并给出了“走廊”自动调整的策略。仿真结果表明: 优化后的轨迹无碰撞且曲率更小, 加速度不超出正常范围, 可以满足正常驾驶需求。

关键词: 混合 A* 算法 轨迹优化算法 贝塞尔曲线 自动驾驶

中图分类号: U463.6; TP18

文献标识码: A

DOI: 10.19822/j.cnki.1671-6329.20220270

Hybrid A* Algorithm Based Trajectory Generation for Automated Driving Vehicle

Zhou Su^{1,2}, Wang Ruoyi¹, Zhang Youfa², Zhang Gang¹

(1. School of Automotive Studies, Tongji University, Shanghai 201804; 2. Sino-German College, Tongji University, Shanghai 201804)

【Abstract】The hybrid A* algorithm of automated driving vehicles runs quickly, but it is difficult to ensure the continuity of curvature and comfort of vehicles, and can not guarantee the optimized results without collision. Aiming at the above problems, this paper aims to improve the comfort of vehicles, and optimize A* algorithm based on minimum impact. Algorithm pre-study in MATLAB and raster map is conducted to solve the problem of insufficient comfort during the motion of automated driving vehicles in low-speed scenarios. Then, the optimization algorithm is improved to ensure the safety of vehicles for the problem that the optimization algorithm may lead to trajectory collision, and the strategy of automatic adjustment of the “corridor” is given. The simulation results show that the optimized trajectory is collision-free and the curvature of the trajectory is smaller, and the acceleration does not exceed the normal range, which can meet the normal driving requirements.

Key words: Hybrid A* algorithm, Trajectory optimization algorithm, Bessel curve, Automated driving

1 引言

自动驾驶汽车配备先进的传感器、控制器和执行器, 可以感知复杂环境、进行智能决策和运动控制, 最终使车辆达到人类驾驶员驾驶水平^[1-2]。

规划作为驾驶自动化系统中不可缺少的环节^[3], 不仅需要从感知层获取当前的周围环境信息和自身定位信息, 还需要将规划好的动作信息传到控制层,

再交由控制层对汽车进行控制。一个规划迅速、反应良好的规划层直接影响无人驾驶汽车的性能。

20 世纪至今, 规划方法发展迅速, 根据具体的计算方法, 规划算法分为 5 类:

- (1) 基于图搜索的规划算法;
- (2) 基于采样的规划算法;
- (3) 基于插值的规划算法;
- (4) 基于数值优化的规划算法;

(5)基于机器学习的规划算法。

各类规划算法研究进展如下:

(1)基于图搜索的规划算法

图搜索算法主要指遍历从起点到终点的状态空间以搜索出当前最优的结果,一般使用栅格、格子来表示一个点的状态。

迪杰斯特拉算法是1种单源最短路径的最优路径规划算法,在迪杰斯特拉算法中配置状态空间,一般被离散为单元网格后再进行搜索^[4]。Dijkstra最早给出了这种算法的严格证明^[5]。在自动驾驶领域,迪杰斯特拉算法已经被应用于多车仿真中^[6],本-富兰克林车队(The Ben Franklin Racing Team)和维克多-探戈(Vector Tango)将其应用于DARPA城市挑战赛中^[7]。

A*算法是带有启发式函数的迪杰斯特拉算法,非常适合已知地图下的路径规划问题^[8],但是当要搜索的区域过大时,其所耗费的内存空间也会随之增大,搜索速度随之下降。

A*算法家族具有很多成员,这些A*类算法相比A*有了很多的提升,也被广泛应用到自动驾驶、机器人等领域。比较常见的有Hybrid A*算法、D*算法(Dynamic A*, D*)、Field A*算法^[9]、Theta*算法^[10]、ARA*算法(Anytime Repairing A*, ARA*)和AD*算法(Anytime D*, AD*)^[11]。在自动驾驶领域中,Ziegler等^[12]在文献中实现了A*算法用于非结构化空间下(停车场)的规划。混合状态A*算法被斯坦福大学用于DARPA挑战赛^[13]。2007年的DARPA挑战赛获胜车辆Boss也使用了AD*算法^[14]。Saeid等^[15]使用了Hybrid A*算法实现了停车场中实验车辆的自动泊车。Francesco等^[16]则通过减少车辆转弯和向后行驶,提高了乘客在自动泊车时的乘坐舒适度。Silvana等^[17]使用正态分布变换算法建图后,使用Hybrid A*算法进行路径规划,并使用模型预测控制算法进行控制器设计,实现了在园区内稳定运行的高尔夫球车。

Howard等^[18]将状态栅格法应用于崎岖地形下的轮式移动机器人的全局规划和局部规划。Likhachev等^[8]根据控制复杂度将状态空间离散化,而McNaughton等^[19-22]考虑了时间和速度,将状态根据时间空间离散化。

(2)基于采样的规划算法

基于采样的算法中也有很多实用且高效的算法。如概率路图法(Probabilistic Roadmap, PRM)^[23],快速搜索随机树算法(Rapidly-Exploring Random Trees, RRT)^[24]。其中,RRT算法也已经被应用到自动驾驶领域。

RRT算法是一种快速的在线搜索算法^[25],可以在半结构化空间中进行快速规划,考虑了非完整约束^[24]。MIT将该算法应用到了DARPA挑战赛中^[26],但其结果并不稳定,而且曲率也不连续。与A*类算法一致,RRT类算法也有很多成员,如RRT*算法^[27]、Informed RRT*算法^[28]、Anytime RRT*算法^[29],以及Kinodynamic RRT*算法^[29]。

(3)基于插值的规划算法

Reeds等^[31-32]使用了直线和圆弧插值算法来近似汽车规划。螺旋曲线插值中对应的曲线是根据菲涅尔积分定义而来的^[33],使用回旋曲线可以定义曲率线性变化的轨迹,平滑过渡直线段和弯曲线段^[34]。多项式插值算法非常适合多段轨迹的插值问题。Piazzi等在文献[35]、[36]和[37]中分别阐述了相关的计算方法。贝塞尔曲线插值广泛应用于航空航天、汽车设计、游戏众多领域,在自动驾驶领域的研究中,三阶和四阶的贝塞尔曲线被用来规划当前可以转弯、避障的轨迹^[38-39]。

(4)基于数值优化的规划算法

基于数值优化^[40-41]的方法时间上是使受不同约束的函数最大化或者最小化,大多数是基于凸优化的算法,还有一些是基于非线性优化的算法。这种算法实际上已经被广泛应用于移动机器人在狭窄通道中避障的势场方法(Potential Field Method, PFM),其轨迹可以包含一阶连续性^[42]甚至更高阶(速度、加速度、冲击度等)连续性的轨迹。

(5)基于机器学习的规划算法

基于机器学习的方法具有较快的计算速度和较强的泛化能力,但是使用机器学习的方法之前需要进行一定的离线学习。以下机器学习相关的方法被应用在规划领域:强化学习^[43]、流形学习^[44]、卷积神经网络^[45]、隐马尔可夫模型^[46]、高斯混合模型^[47]和径向基函数神经网络^[48]。

通过以上分析,目前自动驾驶车辆的运动规划算法存在以下缺陷:

(1)考虑了汽车动力学的规划算法计算较慢,且不能保证规划处的路径总是无碰撞的。

(2)汽车的路径规划与某些机器人和无人机的路径规划不同,汽车的运动规划属于非完整约束问题,且大多数规划算法规划出的结果是路径,并没有规划出其轨迹。而且即便是有轨迹规划的算法,其路径和速度是不匹配的,需要对速度重新进行规划。

(3)大多数规划算法并没有考虑多车协同问题。现在绝大多数的规划算法都是考虑已知地图下的单

车规划问题。

因此,本文以提高汽车舒适性为目的,基于汽车的最小冲击度进行建模,在MATLAB和栅格地图中进行了算法预研,解决了自动驾驶汽车运动过程中舒适度不足的问题。然后,针对优化算法可能导致轨迹碰撞的问题,对优化算法进行了改进,保证了汽车的安全性,并给出了“走廊”自动调整的策略。

2 基于混合状态A*算法的路径规划模型

混合状态A*算法最早出现于2007年美国DRA-PA挑战赛^[49],它并不像迪杰斯特拉算法、A*算法那样经过数学推导而得出,但这并不影响该算法的实用性。2007年,斯坦福大学的车队就是使用该算法跑完了全程。混合A*算法是A*算法的一种变体,它不仅继承了A*算法的特性,还可以在更高维度的状态空间进行搜索,并提出新的状态更新规则。

相比A*算法,混合状态A*算法具有更高的搜索维度,主要体现在以下方面:

(1)一般混合状态A*算法会在x坐标、y坐标、朝向角和转弯半径4个维度进行路径规划。而A*算法会选取当前累计代价和启发式函数代价最小的点作为下一节点来膨胀;

(2)与此类似,混合状态A*算法则是基于A*算法和Reeds_Shepp曲线^[31]来决定下一节点,其节点可以不是栅格的顶点。

图1是混合状态A*算法的流程图。

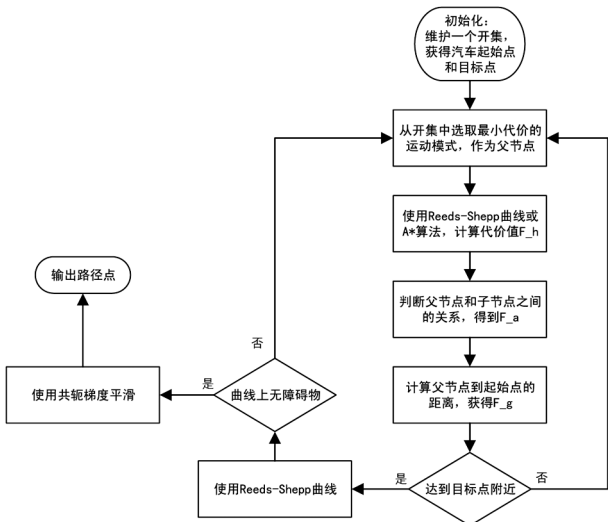


图1 混合状态A*算法流程

3 基于最小冲击度的优化模型

由于很多路径规划算法会将汽车视为点来进行

路径规划,这导致某些规划算法规划出的路径超出了汽车的正常行驶范围,且不能保证车内乘客的舒适性。为了提高汽车在运动过程中的舒适性,本文将对混合A*算法的结果基于最小冲击度进行优化。

3.1 基于最小冲击度的一维优化模型的建立

本文建立的基于最小冲击度的优化模型,取最小冲击度的L₂范数的平方作为优化的目标函数,优化目标是 minimized 最小冲击度的平方,目标函数如式(1)所示。

$$J_k = \frac{1}{T} \int_0^T j_k(t)^2 dt \quad (1)$$

式中, J_k 为目标函数; T 为优化模型时间域; j_k 为最小冲击度。

选取当前的状态量为

$$s_k = [p_k, v_k, a_k] \quad (2)$$

式中, s_k 为当前的状态量; p_k 当前状态的位置; v_k 是当前的速度; a_k 是当前的加速度。同时,将外界输入定义为:

$$u_k = j_k \quad (3)$$

整个系统的状态表达为:

$$\dot{s}_k = f_s(s_k, u_k) = [v_k, a_k, j_k] \quad (4)$$

式中, $f_s(s_k, u_k)$ 为当前状态量与外界输入的状态表达式。

结合庞特里亚金极大值(极小值)原理,可得最终的状态表达式^[50]:

$$s^*(t) = \begin{bmatrix} \frac{\alpha}{120} t^5 + \frac{\beta}{24} t^4 + \frac{\gamma}{6} t^3 + \frac{\alpha_0}{2} t^2 + v_0 t + p_0 \\ \frac{\alpha}{24} t^4 + \frac{\beta}{6} t^3 + \frac{\gamma}{2} t^2 + \alpha_0 t + v_0 \\ \frac{\alpha}{6} t^3 + \frac{\beta}{2} t^2 + \gamma t + a_0 \end{bmatrix} \quad (5)$$

式中, $s^*(t)$ 为系统的最优状态表达式; t 为时间; α 、 β 、 γ 为任意常数; α_0 、 a_0 、 v_0 、 p_0 为 $t=0$ 时的系数、加速度、速度和位置。

3.2 多段优化模型边值问题求解

对于轨迹规划问题,工程中很难只用一段多项表达式就拟合出一个符合预期的从起点到终点的轨迹,更一般性的做法是用多段连续的多项表达式拟合。本文中,后续出现的多项式连续条件均指位置连续、速度连续和加速度连续。同时,路径的多项表达式还需要满足运动的初始条件和终止条件。

对于多项表达式,可以用如下表达式来表示:

$$f(t) = \begin{cases} f_1(t) = \sum_{i=0}^N p_{1,i} t^i & T_0 \leq t \leq T_1 \\ f_2(t) = \sum_{i=0}^N p_{2,i} t^i & T_1 \leq t \leq T_2 \\ \vdots \\ f_M(t) = \sum_{i=0}^N p_{M,i} t^i & T_{M-1} \leq t \leq T_M \end{cases} \quad (6)$$

式中, $f(t)$ 为多段连续多项式的总表达式; $f_M(t)$ 为各段多项式表达式; $T_{M-1} \leq t \leq T_M$ 为时间分段。

同时还需满足如下约束:

$$f_j^{(k)}(T_j) = f_{j+1}^{(k)}(T_j) \quad (7)$$

$$\begin{cases} f_j^{(k)}(T_{j-1}) = x_{0,j}^{(k)} \\ f_j^{(k)}(T_j) = x_{T_j}^{(k)} \end{cases} \quad (8)$$

式(7)是指多项表达式中每相邻2段表达式之间连接处需要满足连续性约束;而式(8)是指第 j 段表达式的头和尾部需要满足该处的状态约束(位置、速度以及加速度)。本文中,每相邻2段的状态约束只包含位置约束。

有了以上目标函数和约束还不够,为了求解多段表达式的最小冲击度,还需要知道每段上的时间。本文中的时间按照常数或者梯形速度曲线进行分配。

综上可以得出最终优化表达式:

$$\begin{aligned} \min & \begin{bmatrix} p_1 \\ \vdots \\ p_M \end{bmatrix}^T \begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & Q_M \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_M \end{bmatrix} \\ \text{s.t.} & \mathbf{A}_{eq} \begin{bmatrix} p_1 \\ \vdots \\ p_M \end{bmatrix} = \mathbf{d}_{eq} \\ & T_i = f_i(s_i, I, I_i, \hat{v}); \end{aligned} \quad (9)$$

式中, \min 式为凸目标函数; s.t. 式为约束函数; T_i 为仿射函数。

3.3 基于最小冲击度的优化结果

在以下场景中,设置汽车起始位置为[9,16,0],再绕过障碍物后,需要停在停车位上,最终的状态为[12, 8, π] (π 表示停车时车头朝向角为 π ,后文均采用此种表述)。混合状态A*算法的结果图2所示。方点代表汽车的起点,圆点代表终点,点线代表算法的历史前向搜索记录,点划线代表算法的历史后向搜索记录,实线是最终的运动轨迹。

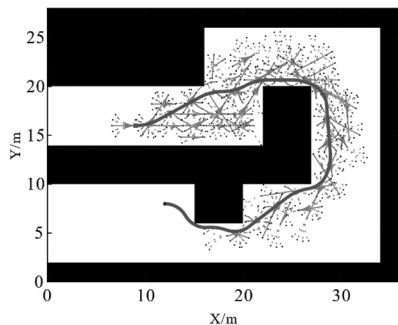


图2 混合状态A*算法的搜索结果

根据优化算法得出的结果,可以计算出此时的速度图像、加速度图像,来进一步分析优化后的结果。

图3是根据优化后的结果得出的汽车状态图像。

从中可知优化后的质心速度大多数时间稳定在3 m/s左右,而质心加速度则变化幅度比较大。实际上,汽车在行驶过程中,最大加速度一般在10 m/s²左右,此处计算出的加速度已经超过了实际的最大加速度。在这种搜索步长下,优化算法对混合状态A*算法搜索出的结果优化效果不明显。同时,又由于缺少不等式约束,导致计算出的加速度容易超出汽车实际的最大加速度。因此算法值得进一步的改进。

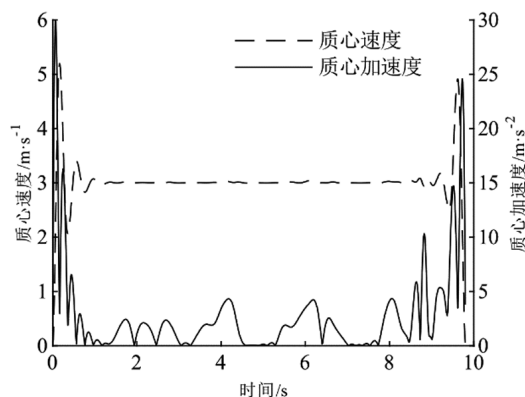


图3 优化后汽车的运动速度和加速度

优化过程需要花费的时间也是优化算法十分重要的评价指标。基于最小冲击度的优化算法的运行速度实际上是比较快的。如表1所示,Hybrid A*代表混合状态A*算法的运行时间,QP和CF则分别代表二次规划求解器和闭式解求解方法的运行时间。在混合状态A*算法搜索出50个路径点的条件下,基于二次规划求解器的求解算法和基于闭式解的求解算法生成了576个系数,共7224个数值。在这种计算规模下,MATLAB中的3次试验结果显示,不论是基于二次规划求解器的求解方式,还是基于闭式解的求解方式,优化算法的计算时间都不会超过0.1 s。即便是加上混合状态A*算法的耗时,运算时间也基本保持在0.5 s左右。

表1 算法运行时间比较 s

算法	Hybrid A*	QP	Hybrid A*+ QP	CF	Hybrid A*+ CF
第1次	0.249 9	0.023 4	0.273 3	0.027 2	0.277 1
第2次	0.421 8	0.076 2	0.498 0	0.042 8	0.464 6
第3次	0.354 5	0.027 2	0.381 7	0.028 8	0.383 3
平均	0.342 1	0.042 3	0.384 3	0.032 9	0.375 0

对于前文中(图3)加速度超出实际范围的情况,可以在优化之后再单独加一层速度规划层,便可以基于优化结果得出更好的速度和加速度结果。此处采用梯形速度分配公式来做速度规划,设置最高速度为

3 m/s,便可以得出如图4所示的结果。

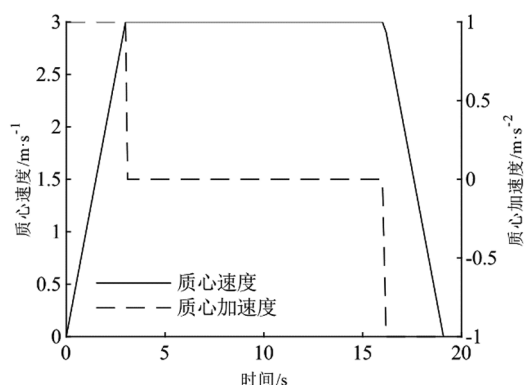


图4 速度、加速度重规划

本文采用的地图是栅格地图,以上试验全部是在静态层进行的试验。实际应用中,从起点到终点的运动过程中,汽车需要不断地感知周围的环境。在静态地图中,当汽车接近突然出现的障碍物或汽车运动到某位置才检测到周围具有障碍物时,可能导致之前规划的轨迹失效,需要规划器重新做出规划。所以此处给汽车栅格地图障碍物层加入障碍物信息,并给汽车加上传感器再次进行试验。

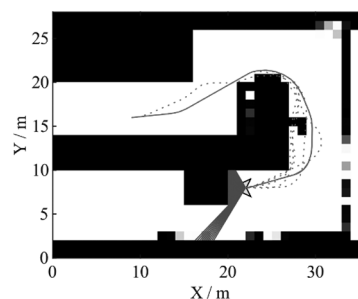
假设汽车可以感知到周围环境 120° 范围内的障碍物,探测范围在 $[0, 20]$ 之间。设置汽车起始位置为 $[9, 16, 0]$, 最终停在停车位上的状态为 $[22, 8, \pi]$ 。用实线代表当前算法搜索出的轨迹,虚线代表历史搜索出的轨迹,而辐射范围代表传感器可探测到的范围,方块代表检测到障碍物后膨胀的方格。需要注意的是,如果规划的轨迹存在“局部碰撞”(未来可能发生碰撞),算法也会重新规划路径。最后图5显示了2种算法到达终点附近的运动历史轨迹,由虚线可知,2种算法均会做出多次重新规划。同时,优化算法在某些障碍物附近的曲率变化略小于混合状态A*算法。

直观的来看,基于优化算法的最终轨迹相对于混合状态A*算法的曲率变化率较小,但是还是不能改变混合状态A*算法在某些点距离障碍物较近的缺点。虽然这种缺点是有可能通过调整混合状态A*算法的参数来解决的,但仅从优化的角度来看,基于最小冲击度的优化算法是很难解决这样的问题的。

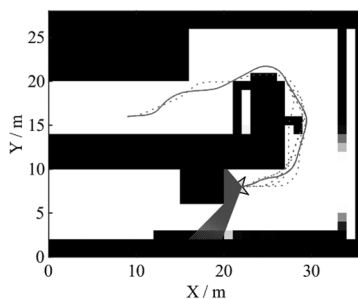
4 硬约束下基于最小冲击度的优化模型

由于很多路径规划算法会将汽车视为点来进行路径规划,这导致某些规划算法规划出的路径超出了汽车的正常行驶范围,且不能保证车内乘客的舒适性。为提高汽车在运动过程中的舒适性,实际工程中

会根据最小冲击度的大小对当前运动情况进行调整。所以,在本文对混合A*算法的结果基于最小冲击度进行优化,但是发现基于最小冲击度的优化算法在大步长时生成的轨迹会碰撞到障碍物(图6a)。如果假设在自由空间 E_{Free} 存在一条从起点到终点的无碰撞“走廊”(图6b),规划出的轨迹位于走廊之内,那么最终的轨迹无论如何都不可能发生碰撞。因此本文尝试采用此方法求解最终轨迹。

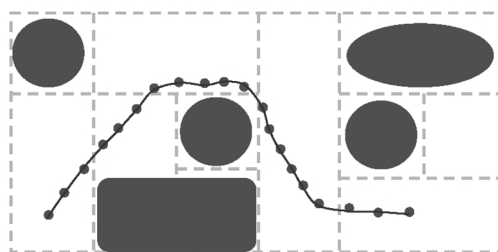


(a)混合状态A*算法搜索

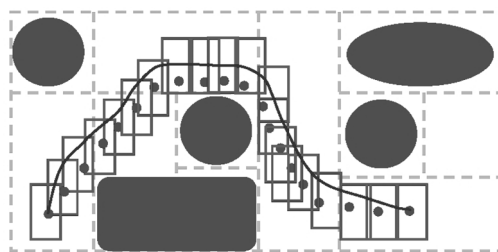


(b)基于最小冲击度优化的轨迹

图5 两种算法的最终运动轨迹



(a)大步长时碰撞轨迹示意



(b)无碰撞“走廊”示意

图6 算法改进策略

硬约束下基于冲击度的优化模型就是要将最终

的目标函数化为带有不等约束的目标函数,此处依旧按照解凸优化问题的求解器进行求解。与之相反的,软约束下的优化模型就是要将所有的约束写入目标函数中,最终使用梯度下降法、牛顿法或拟牛顿法进行求解。

本文基于贝塞尔曲线实现轨迹的无碰撞,其优点如下:

(1)传统的基于贝塞尔曲线的轨迹规划方式中,不同阶数的贝塞尔曲线对应着不同数量的控制点,业内经常利用4次和3次贝塞尔曲线来做插值^[50-51]。通常的做法是,根据速度、加速度、贝塞尔曲线的阶数约束,先得到相应阶数下从起点到终点的控制点,然后再根据控制点计算出最终的轨迹。而本文中,贝塞尔曲线的阶数是固定的,前文已经证明基于最小冲击度的轨迹可以用5次多项式来表示,所以此处的伯恩斯坦多项式也是5次多项式。

(2)本文将首先根据规划算法(混合A*类算法)规划出全局的路径点。然后,在规划出的路径点附近形成一系列连续的“走廊”,这些“走廊”全部位于自由空间EFree,最后在走廊中生成贝塞尔曲线。

(3)传统的贝塞尔曲线无所避免的会出现“打结”现象,而本文中只固定起点和终点,而“释放”其它规划的路径点,最终形成的多段贝塞尔曲线可以避免“打结”。

(4)本文中利用贝塞尔曲线来构建一个基于最小冲击度的优化问题,属于优化问题范畴;传统贝塞尔插值式的路径规划算法属于插值式的规划算法,属于规划范畴。

本文中“走廊”内规划的思想也与自动驾驶时代“高精度地图+管道式行驶”的精神相契合。

4.1 贝塞尔曲线

由规划算法得出的路径点得到“走廊”之后,如何确保最终生成的轨迹不超出走廊是目前需要考虑的问题。而贝塞尔曲线无疑是一个比较好的选择,只要贝塞尔曲线的控制点都在“走廊”以内,那么贝塞尔曲线也在“走廊”以内。这个逻辑之所以成立,是因为贝塞尔曲线具有凸包性。贝塞尔曲线对应的多项式具有以下形式:

$$B_j(t) = c_j^0 b_n^0(t) + c_j^1 b_n^1(t) + \dots + c_j^n b_n^n(t) = \sum_{i=0}^n c_j^i b_n^i(t) \quad (10)$$

$$b_n^i(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

式中, $B_j(t)$ 为各走廊内贝塞尔曲线表达式; c_j^i 为贝塞尔曲线的控制点; $b_n^i(t)$ 为各控制点的系数表达式,与

控制点个数与时间 t 有关。

本文中主要利用了贝塞尔曲线的以下性质:

(1)贝塞尔曲线的凸包性,即由一组控制点形成的贝塞尔曲线始终位于这组控制点组成的凸包以内。

(2)贝塞尔曲线的导数仍然是贝塞尔曲线。

(3)贝塞尔曲线中的 t 始终在 $[0,1]$ 区间内。

4.2 硬约束下基于最小冲击度优化模型的建立

由于仍然是基于最小冲击度的一维优化模型,所以硬约束下最小冲击度模型的目标函数仍然与式(9)相似,但需要将一般多项式转化为伯恩斯坦多项式。另一方面,为了解决速度、加速度会超出计算范围的问题,还需要给模型加上不等式约束。等式约束仍旧是连续性约束(位置连续、速度连续和加速度连续)、边界约束(初始时刻和终止时刻的位置约束、速度约束和加速度约束)和中间点位置约束。由于此处的目标函数已经转化为伯恩斯坦多项式,所以此处所有的约束也需要转换为相应形式。

对于如下形式的一般多项式:

$$p_j(t) = p_j^0 + p_j^1 t + p_j^2 t^2 + \dots + p_j^n t^n \quad (11)$$

式中, $p_j(t)$ 为 n 阶伯恩斯坦多项式; $p_j^0 \dots p_j^n$ 为多项式中的算子。

要将其转化为式(10)所示的形式,需要建立一个映射矩阵如:

$$p = M \cdot c \quad (12)$$

式中, p 为一般多项式(11)中的系数向量; c 为贝塞尔曲线多项式(10)中的系数向量; M 为系数向量 p 、 c 的映射矩阵。

添加速度约束和加速度约束:

$$\begin{aligned} v_m^- &\leq n \cdot (c_{\mu}^i - c_{\mu}^{i-1}) \leq v_m^+ \\ a_m^- &\leq n \cdot (n-1) \cdot (c_{\mu}^i - 2c_{\mu}^{i-1} + c_{\mu}^{i-2}) \leq a_m^+ \\ \mu &\in \{x, y\}, i = 0, 1, 2, \dots, n \end{aligned} \quad (13)$$

式中, v_m^- 、 v_m^+ 为速度约束上、下限; a_m^- 、 a_m^+ 为加速度约束上、下限; μ 为位置取值范围。

根据这些目标函数和约束,可以最后推到出多段硬约束下基于最小冲击度下的优化模型如式所示:

$$\begin{cases} \min & c^T Q_0 c \\ s.t. & A_{eq} c = b_{eq} \\ & A_{ineq} c \leq b_{ineq} \\ & T_i = f_i(s_i, v_i, a_i, \hat{v}) \\ & c_j \in \Omega_j, j = 1, 2, \dots, m \end{cases} \quad (14)$$

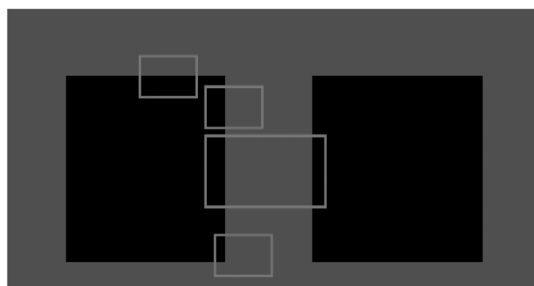
式中, \min 式为凸目标函数; $s.t.$ 式为约束函数。

最终,问题转换成了一个典型的凸优化问题,可以采用求解凸优化问题的方法来求解相应的系数。

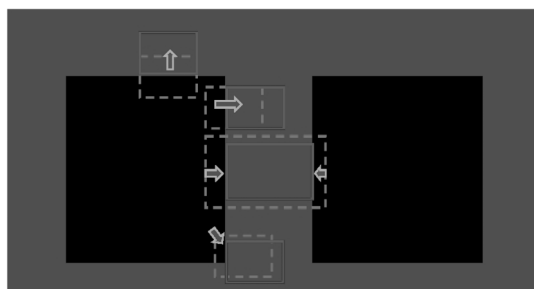
4.3 对混合状态A*算法的优化结果

由于基于最小冲击度的优化对混合A*的优化效果并不明显,且加速度会超出实际汽车的最大加速度。本文采用硬约束下基于最小冲击度的优化算法对混合状态A*算法进行优化,当路径点距离障碍物较近时,在原来路径点基础上得出走廊会存在和障碍物重合的现象。这时得出的最终轨迹也会因此发生碰撞,优化结果应在无碰撞的基础上,对原来的路径点做出优化。

本文采用矫正走廊位置的方法,即将当前存在碰撞的“走廊”向自由空间挪动。如图7,对于一个顶点和障碍物重合的情况,可以沿着对角线附近方向微调,不一定沿着对角线方向,因为这样的调整方法可能会造成“走廊”不连续的情况。对于2个顶点和障碍物重合的情况,只需要将走廊中心沿着另一端平移即可。而对于4个顶点与障碍物重合但中心不重合的情况,则需要沿2端的连线方向缩小走廊,然后再平移走廊。所以,可以得出图8的走廊调整算法。



(a)“走廊”与障碍物重合情况示意



(b)“走廊”调整方向示意

图7 “走廊”调整策略

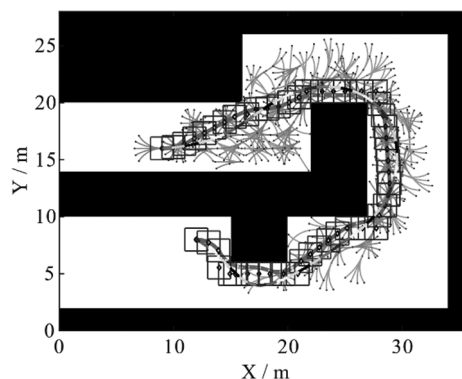
设置初始状态为[9,16,0],最终状态为[12.8,pi],绕过障碍物后汽车需要停在停车位上。星点代表了由混合状态A*算法搜索得到的路径点,方框内的实线是混合状态A*算法得出的轨迹,菱形是根据混合状态A*算法搜索得到的路径点调整的“走廊”的中心,方框构成了最终的“走廊”。在不改变其它条件的情况下,调整“走廊”在x和y方向上的长度都为3m。如图9所示,对比起走廊约束分别为2m的结果,更大的约束下

得出的优化结果与混合状态A*得出的轨迹差别更大。

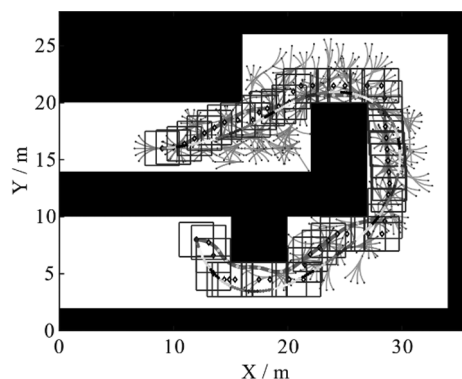
```

走廊调整算法
算法
输入:规划得出的路径点集P,当前x,y方向的位置约束
输出:调整后的走廊
新的走廊中心 New_P=[]
新的走廊约束 X=[],Y=[]
For i = 1:len(P)
    While (当前点的走廊位置与障碍物重合)
        移动走廊中心
        if (走廊未重合部分与障碍物发生了重合)
            缩小当前走廊的x,y约束,并移动走廊
        end
    end
end
New_P.push_back(当前移动后的走廊中心)
X.P.push_back(new_x)
Y.P.push_back(new_y)
end
return 根据新的走廊中心New_P,X和Y构造出移动后的走廊
    
```

图8 “走廊”调整算法



(a)走廊约束为2m的结果



(b)走廊约束为3m的结果

图9 不同走廊约束下的优化结果

表2是在混合状态A*算法搜索出49个路径点,优化算法得到588个系数,4949个路径点的条件下,在MATLAB中混合状态A*算法以及硬约束下的优化算法的运行时间。此处进行了3次试验。

在相同条件下试验,最终的走廊不再与障碍物重合,所以最终的轨迹也不再发生碰撞。不过从运行时间来看,优化算法的运行时间并不快。与上一节类

似,最后需要在栅格地图的障碍物层加入障碍物信息,给汽车加上传感器,以探究动态过程中算法的表现。从起点到终点的运动过程中,汽车需要不断地感知周围的环境。在地图中,当汽车接近突然出现的障碍物或汽车运动到某位置才检测到周围具有障碍物时,需要规划器重新做出规划。

表2 硬约束下基于最小冲击度优化算法运行时间 s

算法	Hybrid A*	HCOP	Hybrid A*+ HCOP
第1次	0.234 1	0.395 8	0.630 0
第2次	0.257 5	0.563 7	0.821 2
第3次	0.222 3	0.430 8	0.653 1
平均	0.238 0	0.463 5	0.701 4

注:HCOP为 Hard Constraints Optimization。

4.4 对优化结果的评价

此处对图9(b)的结果进行进一步讨论。已知初始状态为 $[9,16,0]$,最终状态为 $[12,8,\pi/2]$,绕过障碍物后汽车需要停在停车位上。混合状态A*算法的最小转弯半径为2 m,搜索步长为2 m。设置“走廊”在x和y方向上的长度都为3 m。汽车的最大速度为20 m/s,汽车的最大加速度为 10 m/s^2 。

在得出从起点到终点的路径之后,对多项式求导便可以得出其速度、加速度和冲击度。由此获得速度曲线、加速度曲线和冲击度曲线如图10和图11。由这2幅图可知,优化后的曲线是近似梯形的,最高速度是 3.87 m/s ,对应的速度代价值,即路程为60.85 m。同时,最大加速度也在合理范围内,为 3.97 m/s^2 。除去起始和终止时刻,冲击度也较小,优化后的代价值为 $213.66 \text{ m}^2/\text{s}^5$ 。

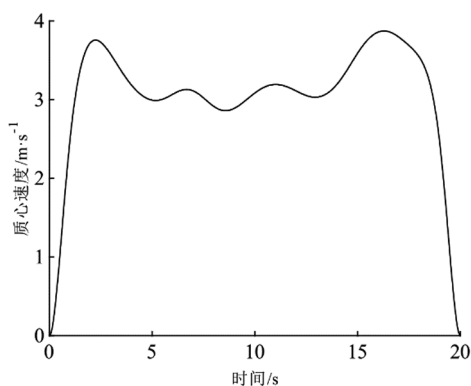


图10 速度曲线

至于优化前后轨迹的曲率,可以通过优化前后的路径点来计算。除去起点和终点的曲率无法计算外,其它点最终都可以得出一个曲率值。优化前有320个路径点,最终得出318个点的曲率。优化后的点有505个点,

最终得出503个点的曲率。可由图12知,优化前的曲率振荡振幅很大。由于优化前后计算曲率的方法一致,考虑到可能是由于优化前的点过于稠密,所以删除点的个数为原来的1/4,便可得出图12b所示的曲率。此时,优化后的曲率是小于优化前的曲率。由于数值计算的不稳定,所以初始和终止的曲率会比较大。事实上,当具体分析起点和终点的曲率数值时,可以看出初始时刻,只有4个点是过大的,终止时刻,则有8个点较大,但是相比起505点,这12个点只占2%左右。也就是说,绝大多数计算出的曲率是在正常范围内的。

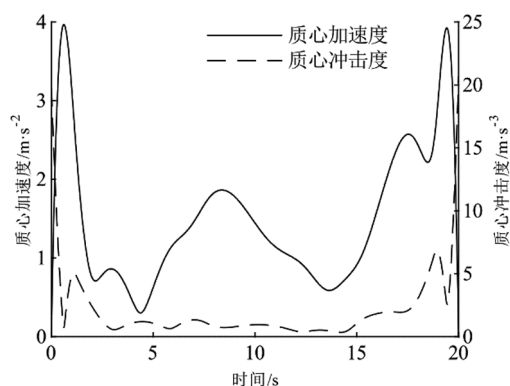
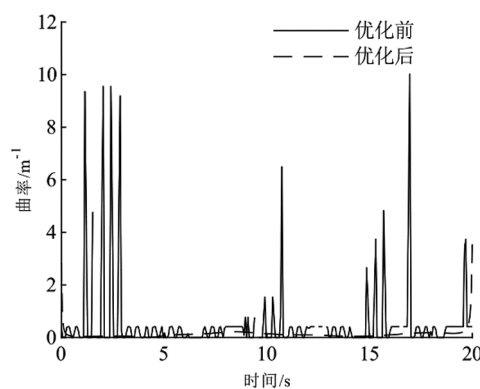
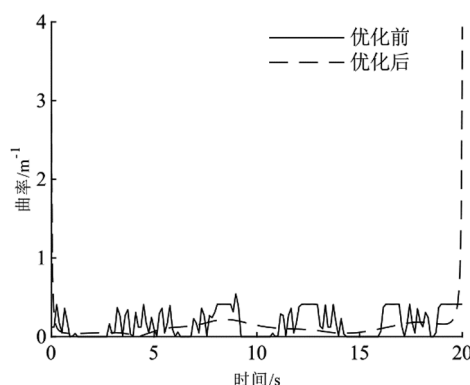


图11 加速度和冲击度



(a)优化前的点未作稀疏处理



(b)优化前的点作稀疏处理

图12 优化前后的曲率

最后,分析优化前后路径点距离周围障碍物的最小距离,从图 13 可知,优化前路径点到障碍物的最小距离为 0.20 m,而优化后路径点到障碍物的最小距离为 0.30 m。优化前距离的代价值为 6.19,而优化后为 3.76,在距离这个评价指标上,优化后的路径点也是优于优化前的。

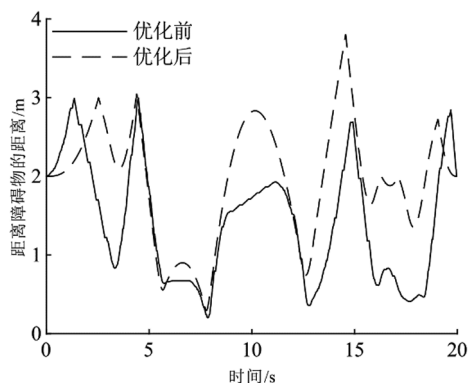


图 13 优化前后路径点到障碍物的距离

从上面的试验结果可以看出,优化后的轨迹不仅是基于最小冲击度,同时也可以得出近似梯形的速度曲线。优化后的曲率和到障碍物的距离也优于优化前的曲率和距离。这说明硬约束下基于最小冲击度的优化算法在整体上是优于优化前的算法,可以在保证舒适性的前提下,得出较优的轨迹,具有应用于实际场景的潜力。

5 结语

本文主要针对自动驾驶汽车的轨迹优化问题展开研究,聚焦如何通过优化手段,得出汽车最终可行驶优化轨迹的同时提高自动驾驶汽车的舒适性。主要研究了最小冲击度下汽车的轨迹生成问题,基于硬约束下的最小冲击度模型对原有的优化模型进行了改进。对比分析后可知:虽然硬约束下的优化算法计算速度不占优势,但是优化后的轨迹无碰撞,且在基于最小冲击度的前提下,曲率更小、加速度保持在正常范围内、距离代价值也降低了 39.3%。总体上硬约束下基于最小冲击度的优化算法解决了原有模型优化后相关状态量超出实际范围及优化后出现碰撞的问题,使得原有模型更加适合于非结构化空间场景。

参考文献

- [1] 宋飞扬. 无人驾驶汽车及其发展[J]. 中国高科技, 2019(5): 24-27.
- [2] 袁师召, 李军. 无人驾驶汽车路径规划研究综述[J]. 汽车工程师, 2019(5): 11-13+25.
- [3] 朱大奇, 颜明重. 移动机器人路径规划技术综述[J]. 控制

与决策, 2010, 25(7): 961-967.

- [4] MARCHESE F M. Multiple mobile robots path-planning with MCA[C]//International Conference on Autonomic and Autonomous Systems (ICAS'06), Silicon Valley, CA, USA: IEEE, 2006: 56-56.
- [5] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269-271.
- [6] KALA R, WARWICK K. Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization[J]. Journal of Intelligent & Robotic Systems, 2013, 72(3): 559-590.
- [7] BACHA A, BAUMAN C, FARUQUE R, et al. Odin: Team Victor Tango's entry in the DARPA Urban Challenge[J]. Journal of field Robotics, 2008, 25(8): 467-492.
- [8] LIKHACHEV M, FERGUSON D. Planning long dynamical-ly feasible maneuvers for autonomous vehicles[J]. The International Journal of Robotics Research, 2009, 28(8): 933-945.
- [9] FERGUSON D, STENTZ A. Using interpolation to improve path planning: The Field D* algorithm[J]. Journal of Field Robotics, 2006, 23(2): 79-101.
- [10] DANIEL K, NASH A, KOENIG S, et al. Theta*: Any-angle path planning on grids[J]. Journal of Artificial Intelligence Research, 2010, 39(1): 533-579.
- [11] LIKHACHEV M, FERGUSON D, GORDON G, et al. Any-time search in dynamic graphs[J]. Artificial Intelligence, 2008, 172(14): 1613-1643.
- [12] ZIEGLER J, WERLING M, SCHRODER J. Navigating car-like robots in unstructured environments using an obstacle sensitive cost function[C]//2008 IEEE intelligent vehicles symposium, Eindhoven, Netherlands: IEEE, 2008: 787-791.
- [13] MONTEMERLO M, BECKER J, BHAT S, et al. Junior: The Stanford entry in the urban challenge[J]. Journal of field Robotics, 2008, 25(9): 569-597.
- [14] FERGUSON D, HOWARD T M, LIKHACHEV M. Motion planning in urban environments[J]. Journal of Field Robotics, 2008, 25(11-12): 939-960.
- [15] SEDIGHI S, NGUYEN D V, KUHNERT K D. Guided Hybrid A-Star Path Planning Algorithm for Valet Parking Applications[C]//2019 5th International Conference on Control, Automation and Robotics, Beijing, China: IEEE, 2019: 570-575.
- [16] ESPOSTO F, GOOS JORRIT, TEERHUIS A, et al. Hybrid Path Planning for Non-Holonomic Autonomous Vehicles: An Experimental Evaluation[C]//2017 5th IEEE Interna-

- tional Conference on Models and Technologies for Intelligent Transportation Systems, Naples: IEEE, 2017: 25–30.
- [17] GUIRGUIS S, GERGIS M, ELIAS C M, et al. ROS-Based Model Predictive Trajectory Tracking Control Architecture Using LiDAR-Based Mapping and Hybrid A* Planning [C]//2019 IEEE Intelligent Transportation Systems Conference. Auckland, New Zealand: IEEE, 2019: 2750–2756.
- [18] HOWARD T M, KELLY A. Optimal rough terrain trajectory generation for wheeled mobile robots[J]. The International Journal of Robotics Research, 2007, 26(2): 141–166.
- [19] MCNAUGHTON M, URMSON C, DOLAN J M, et al. Motion planning for autonomous driving with a conformal spatiotemporal lattice[C]//2011 IEEE International Conference on Robotics and Automation. Shanghai, China: IEEE, 2011: 4889–4895.
- [20] GU T Y, SNIDER J M, DOLAN J, et al. Focused trajectory planning for autonomous on-road driving[C]//2013 IEEE Intelligent Vehicles Symposium (IV). Gold Coast, QLD, Australia: IEEE, 2013: 547–552.
- [21] WERLING M, KAMMEL S, ZIEGLER J, et al. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds[J]. The International Journal of Robotics Research, 2012, 31(3): 346–359.
- [22] FURGALE P, SCHWESINGER U, RUFLI M, et al. Toward automated driving in cities using close-to-market sensors: An overview of the v-challenge project[C]//2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia: IEEE, 2013: 809–816.
- [23] KAVRAKI L E, SVESTKA P, LATOMBE J C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces[J]. IEEE transactions on Robotics and Automation, 1996, 12(4): 566–580.
- [24] LAVALLE S M, KUFFNER J J. Randomized kinodynamic planning[C]//Proceedings 1999 IEEE International Conference on Robotics and Automation. Detroit, MI, USA: IEEE, 2001: 378–400.
- [25] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning[J]. The international journal of robotics research, 2011, 30(7): 846–894.
- [26] KUWATA Y, TEO J, FIORE G, et al. Real-time motion planning with applications to autonomous urban driving[J]. IEEE Transactions on control systems technology, 2009, 17(5): 1105–1118.
- [27] KUFFNER J J, LAVALLE S M. RRT-connect: An efficient approach to single-query path planning[C]//Proceedings 2000 ICRA, Millennium Conference, IEEE International Conference on Robotics and Automation. Symposia Proceedings. San Francisco, CA, USA: IEEE, 2000, 2: 995–1001.
- [28] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic[C]//2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. Chicago, IL, USA: IEEE, 2014: 2997–3004.
- [29] KARAMAN S, WALTER M R, PEREZ A, et al. Anytime motion planning using the RRT[C]//2011 IEEE International Conference on Robotics and Automation. Shanghai, China: IEEE, 2011: 1478–1483.
- [30] WEBB D J, VAN DEN BERG J. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics[C]//2013 IEEE International Conference on Robotics and Automation. Karlsruhe, Germany: IEEE, 2013: 5054–5061.
- [31] 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.
- [32] HORST J, BARBERA A. Trajectory Generation for an On-Road Autonomous Vehicle[R/OL]. (2006–06–26)[2022–12–06]. <https://www.nist.gov/publications/trajectory-generation-road-autonomous-vehicle>.
- [33] BREZAK M, PETROVIĆ I. Real-time approximation of clothoids with bounded error for path planning applications [J]. IEEE Transactions on Robotics, 2013, 30(2): 507–515.
- [34] FUNKE J, THEODOSIS P, HINDIYEH R, et al. Up to the limits: Autonomous Audi TT[C]//2012 IEEE Intelligent Vehicles Symposium. Madrid, Spain: IEEE, 2012: 541–547.
- [35] PIAZZI A, BIANCO C G L, BERTOZZI M, et al. Quintic $G/\text{sup } 2/\text{-splines}$ for the iterative steering of vision-based autonomous vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2002, 3(1): 27–36.
- [36] GLASER S, VANHOLME B, MAMMAR S, et al. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction[J]. IEEE Transactions on intelligent transportation systems, 2010, 11(3): 589–606.
- [37] SIMON A, BECKER J C. Vehicle guidance for an autonomous vehicle[C]//Proceedings 1999 IEEE/IEEE/JSAI International Conference on Intelligent Transportation Systems. Tokyo, Japan: IEEE, 1999: 429–434.
- [38] RASTELLI J P, LATTARULO R, NASHASHIBI F. Dynam-

- ic trajectory generation using continuous- curvature algorithms for door to door assistance vehicles[C]//2014 IEEE Intelligent Vehicles Symposium Proceedings. Dearborn, MI, USA:IEEE, 2014: 510-515.
- [39] HAN L, YASHIRO H, NEJAD H T N, et al. Bezier curve based path planning for autonomous vehicle in urban environment[C]//2010 IEEE intelligent vehicles symposium. La Jolla, CA, USA:IEEE, 2010: 1036-1042.
- [40] ZIEGLER J, BENDER P, DANG T, et al. Trajectory planning for bertha—a local, continuous method[C]//2014 IEEE intelligent vehicles symposium proceedings. Dearborn, MI, USA: IEEE, 2014: 450-457.
- [41] WERLING M, KAMMEL S, ZIEGLER J, et al. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds[J]. The International Journal of Robotics Research, 2012, 31(3): 346-359.
- [42] REN J, MCISAAC K A, PATEL R V. Modified Newton's method applied to potential field-based navigation for mobile robots[J]. IEEE Transactions on Robotics, 2006, 22(2): 384-391.
- [43] KONAR A, CHAKRABORTY I G, SINGH S J, et al. A deterministic improved Q-learning for path planning of a mobile robot[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013, 43(5): 1141-1153.
- [44] 连传强. 基于近似动态规划的优化控制方法及在自动驾驶车辆中的应用[D]. 长沙: 国防科学技术大学, 2016.
- [45] BOJARSKI M, DEL TESTA D, DWORAKOWSKI D, et al. End to end learning for self-driving cars[J/OL]. Computer Science(2016-04-25)[2022-12-06]. <https://arxiv.org/abs/1604.07316>.
- [46] LEFEVRE S, CARVALHO A, BORRELLI F. A learning-based framework for velocity control in autonomous driving [J]. IEEE Transactions on Automation Science and Engineering, 2015, 13(1): 32-42.
- [47] 黄如林. 无人驾驶汽车动态障碍物避撞关键技术研究[D]. 合肥: 中国科学技术大学, 2017.
- [48] GORINEVSKY D, KAPITANOVSKY A, GOLDENBERG A. Neural network architecture for trajectory generation and control of automated car parking[J]. IEEE Transactions on Control Systems Technology, 1996, 4(1): 50-56.
- [49] THRUN S, MONTEMERLO M, DAHLKAMP H, et al. Stanley: The robot that won the DARPA Grand Challenge [J]. Journal of field Robotics, 2006, 23(9): 661-692.
- [50] 张友发. 低速场景下自动驾驶汽车轨迹生成[D]. 上海: 同济大学, 2021.
- [51] RASTELLI J P, LATTARULO R, NASHASHIBI F. Dynamic trajectory generation using continuous- curvature algorithms for door to door assistance vehicles[C]//2014 IEEE Intelligent Vehicles Symposium Proceedings. Dearborn, MI, USA: IEEE, 2014: 510-515.

【作者简介】

周苏,男,同济大学,教授,博士生导师。主要研究方向为未来汽车的控制及故障诊断,自动驾驶汽车路径规划以及信号处理。

E-mail:suzhou@tongji.edu.cn

王若伊,同济大学,硕士研究生。主要研究方向为自动驾驶和燃料电池。

E-mail:13161952933@163.com

张友发,硕士研究生。主要研究方向为自动驾驶。

E-mail:y Zhang_12@163.com

张岗,博士生。主要研究方向为自动驾驶和燃料电池。

E-mail:1911564@ tongji.edu.cn