

队伍编号	21190020030
题号	A

基于混合 A*搜索算法的无人车调头轨迹多级规划优化模型

摘 要

随着机器人、人工智能和控制理论等科学技术的发展，无人驾驶，尤其是无人车运动轨迹规划，成为近年来最热门的研究领域之一。其中，调头的轨迹规划是一个既有趣又有重要实际研究需求的场景。如何在过于狭窄的调头区域、静态甚至动态的障碍物、人行横道等阻碍，以及复杂多变的路况信息下，设计一套低复杂度、高成功率、稳定安全、高速通行、规范驾驶、体验舒适的调头路径规划模型和算法，至关重要。

为了更好地建立轨迹规划模型，本文首先建立无人车的单车运动简化模型，并对运动特征进行分析，得出无人车各运动量间的关系。为了更好地对提供的地图进行求解，对地图进行了平面坐标转换与栅格化处理。

做好模型准备后，本文基于混合 A*算法，建立无人车调头轨迹多级规划模型，将调头问题分为多段进行优化。对于车辆的第一段转弯，使用最小圆弧优化模型进行求解，对于对向车道的轨迹优化与避障使用混合 A*算法规划轨迹。该方法大幅降低了混合 A*算法的搜索空间，在兼顾算法求解成功率的同时降低了算法复杂度。

对问题 1 对向车道无障碍物场景，运用上述多级路径规划和速度规划模型与算法，选取最左侧车道为目标，得到较少压线的轨迹如图 8 所示，轨迹点信息如表 3 所示。从图 10 仿真得到的运动参数变化可看出各项指标在无人车运动中变换均较为平缓，规划结果满足良好的驾乘体验与通行效率。

对问题 2 单一对向车道的调头场景，求得了临界条件是附件数据中 A 左边线与 D 右边线距离为 6，并给出了三种场景下的轨迹仿真结果如图 11。

问题 3 针对对向车道存在单个或多个静态障碍物的场景，使用上述模型与算法得到了较优解，取得了良好的避障效果，得到调头轨迹点信息如表 4，仿真过程中运动参数变化如图 12。

问题 4 针对人行横道存在的场景，讨论了无人车不压人行横道调头的两种情况，并得出了针对题目中的场景，如要转弯时不压斑马线，需要先驶过斑马线再进行调头的结论。并对轨迹进行了求解，求解结果如图 13 和表 5。

问题 5 对于对向车道存在动态障碍物的场景，根据障碍物的速度不同，讨论并分析了根据障碍物运动速度高低划分的三种场景。并对三种不同的场景给出了模拟仿真。模型在三种场景中均可得到优化轨迹，如图 15~17 所示。

问题 6 结合对问题中的五种场景的分析，对算法的求解成功率与算法复杂度提出了优化方案，得出的结论为 A*搜索算法需要结合 RS 曲线来缩小解空间，以取得更低的求解耗时，另外，需要对得到的 RS 曲线进行平滑处理，使混合 A*搜索算法取得更好的收敛性和稳定性。

关键词：无人车调头场景；运动轨迹规划；阿克曼转向几何；混合 A*搜索；速度规划

目录

一、 问题重述	1
1.1 问题背景	1
1.2 问题描述	1
1.2.1 问题 1 的描述	1
1.2.2 问题 2 的描述	1
1.2.3 问题 3 的描述	1
1.2.4 问题 4 的描述	1
1.2.5 问题 5 的描述	1
1.2.6 问题 6 的描述	2
二、 问题分析	2
2.1 模型要求	2
2.2 优化条件分析	2
三、 模型假设	2
四、 名词解释和符号说明	3
4.1 名词解释	3
4.2 符号说明	3
五、 模型建立与求解	4
5.1 模型的结构	4
5.2 模型的准备	5
5.2.1 车辆运动学模型	5
5.2.2 二维平面建模	7
5.3 问题 1 模型的建立与求解	9
5.3.1 多级轨迹规划模型	9
5.3.2 车辆速度规划模型 ^[8]	12
5.3.3 问题 1 模型的求解	13
5.4 问题 2 模型的建立与求解	14
5.5 问题 3 模型的建立与求解	15
5.6 问题 4 模型的建立与求解	16
5.7 问题 5 模型的建立与求解	17
5.8 问题 6 模型的建立与求解	18
5.9 结果总结	19
六、 模型评价与推广	19
6.1 模型优点	19
6.2 模型缺点	19
6.3 模型推广	19
七、 模型的改进	20
八、 参考文献	20
九、 附录	22
9.1 使用的软件	22

9.2 相关的源程序代码	22
---------------------------	-----------

一、问题重述

1.1 问题背景

随着机器人、人工智能和控制理论等科学技术的发展，无人驾驶已经成为近些年来最热门的研究领域之一。在无人驾驶领域中，运动轨迹的规划是重点研究内容。无人车的运动轨迹规划，任务在于规划出无人车从初始位置运动到目标位置的一系列符合无人车车辆动力学性质的指令，如速度、加速度、方向角等，并满足安全快速、规范驾驶、舒适驾乘体感等要求。

日益增多的车流量，日渐拥堵的道路，都给无人的轨迹规划造成了巨大的困难，其中一个非常实际且充满趣味的轨迹规划场景，是调头过程中的运动轨迹规划。车辆在进行自主调头轨迹规划时，可能会遇到过于狭窄的调头区域、静态障碍物、人行横道与动态障碍物等不同的阻碍。另外，道路情况较为复杂，且由于调头过程中附近障碍物可能因视野的局限性而被遮挡，环境信息变数较大。因此，提出一套低算法复杂度的、可以根据路况的实时变化动态规划车辆调头路径的模型和算法显得至关重要。

1.2 问题描述

1.2.1 问题 1 的描述

问题 1 主要研究的是无人车调头中最理想的路况场景，即调头区域宽敞、存在足够的对向车道，不考虑人行横道、动态或者静态障碍物等。在这种理想的场景下，无人车在最左侧车道准备调头，需要分析调头过程中控制点的位置，建立无人车调头的数学模型，并通过合理的算法设计得出调头轨迹。此题所建立的无人车调头数学模型，是之后研究更一般情形下的模型的基础。

1.2.2 问题 2 的描述

问题 2 研究的是调头区域狭窄时，得出判断无人车完成调头是否需要倒车的标准，并设计合理的算法，得出无人车调头的仿真结果。

1.2.3 问题 3 的描述

问题 3 在问题 1 的无人车调头模型基础上，增加了对道路上的静止障碍物的考虑，而静止障碍物可以分成两类，一类存在于无人车原本进行调头的路径上，另一类存在于无人车对向车道上，需要讨论其中一类或两类障碍物同时存在下的无人车调头模型，并设计合理的避障算法，得出调头轨迹。

1.2.4 问题 4 的描述

问题 4 在问题 1 的无人车调头模型基础上，增加了对调头道路上存在人行横道的情形的考虑，无人车需要尽可能不压人行横道通过调头区域，若无法做到，则需要后轮越过人行横道后再开始调头，不允许骑人行横道调头，即后轮仍在人行横道上时不能开始调头，需要建立对应的无人车调头数学模型，给出合理的算法设计和相应调头轨迹。此外，需要结合问题 3 中的避障调头模型和算法，得到同时遵守交通规则和避障的模型和算法。

1.2.5 问题 5 的描述

问题 5 在问题 3 和问题 4 的基础上，考虑调头过程中无人车遇到的以一定速度平行于道路移动的障碍物，如对向来车等。需要建立更一般的无人车调头轨迹规划模型，并设计合理的算法，得出无人车调头策略和仿真结果。

1.2.6 问题 6 的描述

出于安全性的考虑，问题 6 要求在设计无人车轨迹规划算法时，既要考虑尽可能高的求解成功率，即收敛性和稳定性都尽可能最佳，也要考虑尽可能小的求解耗时，即尽可能低的算法复杂度。需要对算法在求解成功率和求解耗时两方面优化，使问题 1 至问题 5 中提出的无人车轨迹规划算法能更好地实际应用于复杂的路况中。

二、问题分析

题目可以看作是一个带有约束条件与目标的路径规划问题，技术的关键点有：控制点的选取、目标点的选取、路径的规划、速度规划与优化条件。问题对车的加速度与轨迹曲率进行约束。模型需要在满足约束条件的前提下尽可能满足：1.较高的通行效率 2.尽可能舒适驾乘体感 3.减少不必要的压线行驶 4.较高的求解率与低算法复杂度。下面分别对技术关键点与优化目标进行分析。

2.1 模型要求

- a) 控制点的选取:控制点与优化轨迹重合,为考虑优化问题需要的一个选取的点。
- b) 目标点的选取:车辆最终需要完成掉头,并触碰附件数据中的场景里的绿线,需要在绿线上选取合适的点作为轨迹的目标终点。
- c) 轨迹规划:需要通过汽车动力学模型与轨迹规划模型求解轨迹。而轨迹规划中存在多种约束,如问题 2 中的调头区域的大小、问题 3 中的静态障碍物、问题 4 中人行横道上不允许调头的交通规则、问题 5 中的动态障碍物。
- d) 速度规划:需要对轨迹上各点速度进行规划,并使其在约束条件情况下尽可能满足优化目标。

2.2 优化条件分析

- a) 通行效率:实现较高的通行效率可以理解为需要在约束条件之内,尽可能快地完成路径。需要对车辆行驶轨迹的速度进行合理规划。
- b) 舒适驾乘体感:实现舒适的驾乘体感需要满足速度变化较为平缓、加速度变化较为平缓、法向加速度较小等条件。
- c) 不必要压线行驶:路径规划时尽可能避免使车辆压线。
- d) 高求解率与低算法复杂度:路径规划算法需要在兼顾求解成功率的同时,降低调头路径搜索空间的大小。

三、模型假设

- a) 假设一:假设无人车是左右轴对称的四轮乘用车,车身是一个矩形刚体,行驶过程中采用前轮转向,后轮驱动。
- b) 假设二:无人车的控制点的位置在驾驶的过程中与轨迹点重合,速度方向也与轨迹点的方向角一致,瞬时转向半径与轨迹点的曲率半径相同。
- c) 假设三:假设无人车在平坦路面上行驶,忽略车辆在竖直方向上的运动。
- d) 假设四:忽略纵向和横向的空气动力学。

- e) 假设五：根据我国的交通规则，车辆均需要靠右侧道路行驶，因此假设无人车调头只存在对向车道在左侧的情形，即仅考虑无人车向左调头的情形。
- f) 假设六：假定车辆转弯时最大法向加速度与车辆刹车的最大加速度相同。
- g) 假设七：假定移动的障碍物平行于车道且匀速运动。

四、名词解释和符号说明

4.1 名词解释

- 1. 无人车的轨迹：一条包含位置和时间等信息的曲线，由一系列轨迹点构成。其中每个轨迹点包含两类信息，一类是轨迹点本身的信息，不受通过的无人车影响，即该轨迹点的位置坐标、方向角、曲率、曲率变化率等，而第二类属于通过该轨迹点的无人车的信息，即该无人车的在此轨迹点上的速度和加速度，以及无人车到达此轨迹点的时间。
- 2. 控制点：无人车对称轴上的一点。基于假设一和相应的阿克曼转向模型，我们将无人车的控制点选定为后轴（两后轮中心的连线）的中点，即图 2 中的点 M。

4.2 符号说明

表 1 符号说明

符号	含义	单位
l	车身长度	m
w	车身宽度	m
L	轴距，即前轮中心点连线和后轮中心点连线之间的垂直距离	m
δ_w	方向盘转角	rad
δ	前轮转角	rad
λ	方向盘转角与前轮转角的比值	--
i	引入的轨迹点编号， $i=1, 2, 3, \cdots, n$	--
x_i	轨迹点 i 的横坐标	m
y_i	轨迹点 i 的纵坐标	m
θ_i	轨迹点 i 的方向角	rad

κ_i	轨迹点 i 的曲率	rad/m
α_i	轨迹点 i 的曲率变化率	rad/m ²
v_i	无人车经过轨迹点 i 时的速度	m/s
v_{xi}	无人车经过轨迹点 i 时在惯性坐标系 x-O-y 中沿 x 轴方向的分速度	m/s
v_{yi}	无人车经过轨迹点 i 时在惯性坐标系 x-O-y 中沿 y 轴方向的分速度	m/s
a_i	无人车经过轨迹点 i 时的加速度	m/s ²
t_i	无人车到达轨迹点 i 上的时间	s
ω_i	无人车经过轨迹点 i 时的角速度	rad/s
d_{co}	在某轨迹点时，车身与障碍物之间的最小距离	m
$d_{co\min}$	车身与障碍物之间的最小安全距离	m
R_i	轨迹点 i 的曲率半径，与无人车控制点的瞬时转向半径相同	m
S	规划出的轨迹，即轨迹点 i 的集合	--
s	曲线弧长	m
$a_{\max+}$	最大油门加速度	m/s ²
$a_{\max-}$	极限刹车加速度	m/s ²

五、模型建立与求解

5.1 模型的结构

针对本文第二部分问题分析中的模型需求与优化条件，本文提出了一种基于混合 A* 算法的多级轨迹优化模型，模型结构如图 1 所示：

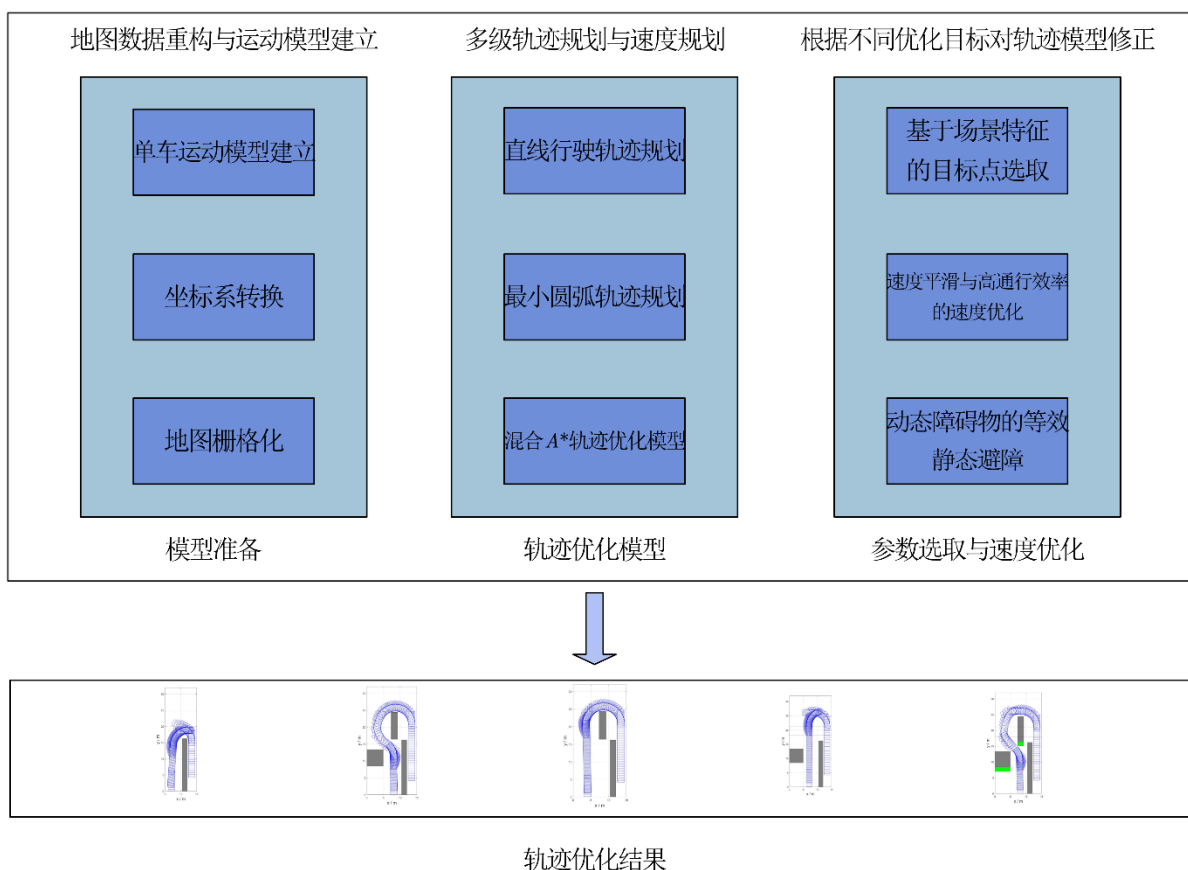


图 1 模型结构图

5.2 模型的准备

5.2.1 车辆运动学模型

与传统的机械装备的运动模型不同，车辆的运动学模型是非线性的。在具体的进行轨迹分析时，需要对无人车的运动学模型进行线性化。根据题目背景可以获知题目中的车辆符合阿克曼转向几何模型^{[1][2]}，如图 2(a)所示。在惯性坐标系 $x-O-y$ 下，点 N 为前轮 A 和 B 之间的前轴的中点，点 M 为后轮 E 和 F 之间的后轴的中点，根据假设一，我们将点 M 作为无人车的控制点。因此点 M 与无人车当前所处的轨迹点 i 重合，点 M 坐标为 (x_i, y_i) ，速度为 v_i ，方向角为 θ_i ，后轮瞬时转向半径与轨迹点 i 的道路曲率半径 R_i 相同。

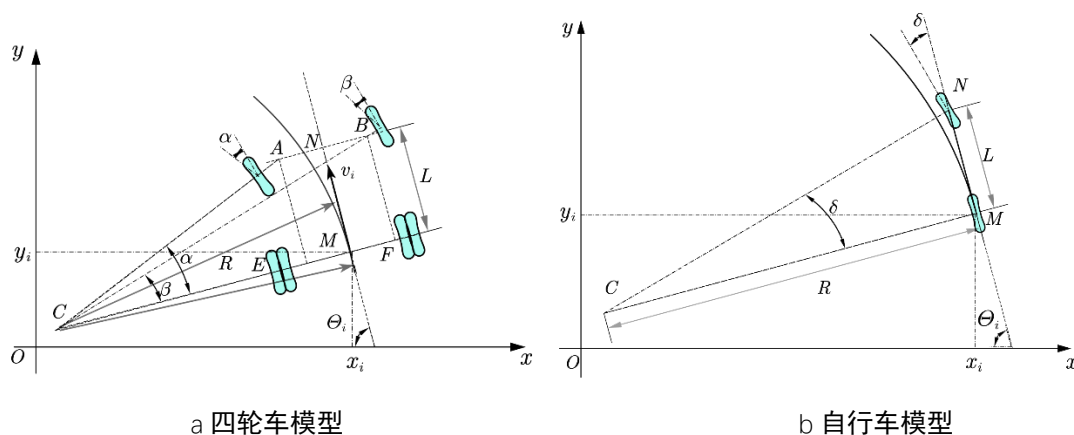


图 2 阿克曼转向几何模型示意图

在低速且不考虑动力学因素情况下,对该模型进一步简化可以得到图 2(b)所示的车辆单轨模型,即自行车模型。对于正低速经过某轨迹点的无人车而言,前轮转角与车轴长和无人车控制点瞬时转向半径有公式(1)所示的关系。

$$\tan(\delta) = \frac{L}{R} \quad (1)$$

根据以上几何关系,定义无人车在平面内三个自由度的运动:

$$\dot{x} = v_x = v \cos(\theta) \quad (2)$$

$$\dot{y} = v_y = v \sin(\theta) \quad (3)$$

$$\dot{\theta} = \omega = v \frac{\tan(\delta)}{L} \quad (4)$$

其中

$$\omega = \frac{v}{R} \quad (5)$$

参考[2]对车辆运动模型的推导,由公式(2)(3)和(4)可知,简化后的无人车运动模型在平面空间由两个方向的平动速度与一个转动速度组成。

尽管时间序列是连续的,但由于观测得到的时间序列是离散的,为更好的对问题进行分析,方便 5.3.2 中速度规划模型的计算,下面对上述模型进行离散化处理得到差分模型,对无人车运动模型在时间维度进行近似。

首先进行线性化处理,定义

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \frac{v \tan \delta}{L} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = f(X, u) \quad (6)$$

在 $X_r = [x_r, y_r, \theta_r]^T$, $u_r = [v_r, \delta_r]^T$ 处进行泰勒展开,且忽略二阶以上的高阶项得

$$\ddot{X} = \dot{X} - \dot{X}_r = \begin{bmatrix} \dot{x} - \dot{x}_r \\ \dot{y} - \dot{y}_r \\ \dot{\theta} - \dot{\theta}_r \end{bmatrix} \approx \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta} \end{pmatrix} \begin{bmatrix} x - x_r \\ y - y_r \\ \theta - \theta_r \end{bmatrix} + \begin{pmatrix} \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial \delta} \\ \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial \delta} \\ \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial \delta} \end{pmatrix} \begin{bmatrix} v - v_r \\ \delta - \delta_r \end{bmatrix} \quad (7)$$

对公式(7)简化,可得在线性化点 X_r 附近的局部增量模型,即无人车运动模型的线性化形式:

$$\ddot{X} = A_m \dot{X} + B_m \dot{u} \quad (8)$$

$$\Delta \dot{X} = A_m \Delta X + B_m \Delta u \quad (9)$$

$$\text{其中: } A_m = \begin{bmatrix} 0 & 0 & -v \sin \theta \\ 0 & 0 & v \cos \theta \\ 0 & 0 & 0 \end{bmatrix}, B_m = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \frac{\tan \delta}{L} & \frac{v}{L \cos^2 \delta} \end{bmatrix}。$$

在一定周期 T 内, 差分函数的状态预测在可接受的误差范围内, 推导无人车运动学模型线性化后的离散形式如下:

$$A_m \Delta X(k) + B_m \Delta u(k) = \dot{\Delta} = \frac{\Delta X(k+1) - \Delta X(k)}{T} \quad (10)$$

变形可得:

$$\Delta X(k+1) = (I + TA_m) \Delta X(k) + TB_m \Delta u(k) \quad (11)$$

令 $A = (I + TA_m), B = TB_m$, 有:

$$\Delta X(k+1) = A \Delta X(k) + B \Delta u(k) \quad (12)$$

$$\text{其中 } A = \begin{bmatrix} 1 & 0 & -vT \sin \theta \\ 0 & 1 & vT \cos \theta \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} T \cos \theta & 0 \\ T \sin \theta & 0 \\ \frac{T \tan \delta}{L} & \frac{vT}{L \cos^2 \delta} \end{bmatrix}。$$

5.2.2 二维平面建模

为更加容易对题目中所给的场景进行求解, 在此对题目场景进行三种建模简化: 平面坐标系转换、障碍物的标准化与平面地图栅格化。

(1) 平面地图转换

题目中所给场景如图 3(a)所示, 所有的多边形均为相对于水平坐标系均为斜线, 不便对图形的网格化与算法仿真, 因此构建坐标转换矩阵对线性空间进行变换, 变换可分为平移变换、旋转变换两个步骤。首先通过平移变换使边框 A 左下角与坐标系原点重合, 即对坐标进行公式(13)所示变换:

$$\begin{pmatrix} x_1^1 & y_1^1 \\ x_2^1 & y_2^1 \\ \vdots & \vdots \\ x_n^1 & y_n^1 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix} - \begin{pmatrix} x_1 & y_1 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_1 & y_1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ x_2 - x_1 & y_2 - y_1 \\ \vdots & \vdots \\ x_n - x_1 & y_n - y_1 \end{pmatrix} \quad (13)$$

其中坐标 (x_1, y_1) 是边框 A 左下角顶点坐标。

经过平移变换后, 再进行旋转变换, 如公式(14)所示。

$$\begin{pmatrix} x_1^2 & y_1^2 \\ x_2^2 & y_2^2 \\ \vdots & \vdots \\ x_n^2 & y_n^2 \end{pmatrix} = A \begin{pmatrix} x_1^1 & y_1^1 \\ x_2^1 & y_2^1 \\ \vdots & \vdots \\ x_n^1 & y_n^1 \end{pmatrix} \quad (14)$$

选取边框 A 的底边线为基准线, 可得转换矩阵 A 为:

$$A = \begin{pmatrix} \cos \alpha & 0 \\ 0 & \sin \alpha \end{pmatrix} \quad (15)$$

坐标系变换后得到的场景如图 3(b)所示。

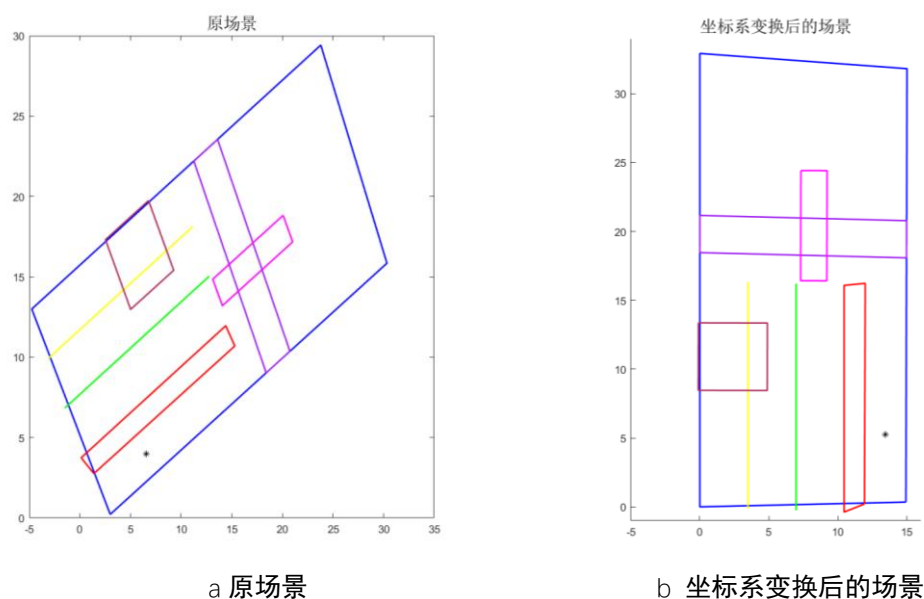


图 3 调头场景坐标系变换前后对比图

(2) 障碍物的标准化

如图 3(b)所示,障碍物的形状近似为矩形,但个别边有略微的倾斜角。为简化分析,对障碍物、无人车、人行道,调头区域变为底边和左边分别与 x 轴和 y 轴重合的标准矩形形状,同时将无人车初始朝向近似为沿 y 轴正方向。具体变换方式如表 2, 变换效果如图 4 所示。

表 2 图内各元素标准化替代方式

种类	替换方式
障碍物	使用最小的完全的包络矩形
调头区域	使用最大的被包络矩形
人行道	使用最小的完全的包络矩形
无人车初始朝向	使用 $\pi/2$

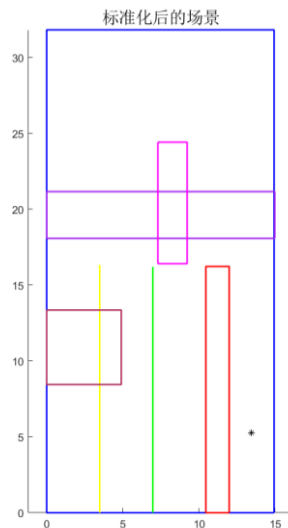


图 4 标准化后的调头场景

(3) 平面地图栅格化^[1]

栅格化是指在一个给定的平面内，使用一系列相互平行的水平线与竖直线对平面空间进行分割，分割后可将连续的平面空间离散化处理的方法。对题目中场景进行合理栅格化后，在使用混合 A* 算法搜索最优路径时可大幅度降低搜索空间的大小，降低算法的复杂度。

根据测试结果，为了兼顾算法的效果与较低的算法复杂度。在此处选择对平面 $m \times n$ 的网格划分，划分后结果如图 5。

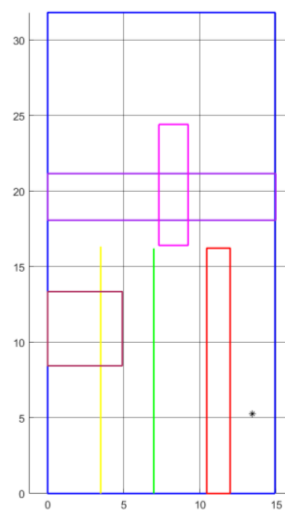


图 5 栅格化后的调头场景

5.3 问题 1 模型的建立与求解

5.3.1 多级轨迹规划模型

(1) 直线行驶轨迹规划

无人车从初始位置到调头区域前，初始朝向近似于平行于道路线，车道上无障碍物，

基于尽可能高的通行效率和尽可能舒适的驾乘体感的考虑，无人车到达调头区域前，驾驶轨迹应当规划为直线。

(2) 最小圆弧轨迹规划^[3]

无人车驾驶的轨迹规划要确保达到驾乘时安全与舒适的要求，就轨迹的性质而言，其中很重要的一点是轨迹的曲率变化率尽可能小，对规划出来的路径而言，即需要各轨迹点曲率变化率的绝对值的平均值尽可能小，有式(16)所示规划目标：

$$S \text{ s.t. } \min \overline{|\alpha|} \quad (16)$$

而圆弧的曲率变化率始终为 0，因此，调头路径规划为圆弧。确定调头路径的形状后，规划需要进一步确定圆弧的半径和圆心。

对于半径而言，出于无人车车辆动力学性质和安全性要求，轨迹线有最大曲率的约束，即 $\kappa_{\max} \approx 2.05 \leq 2.10$ ，曲率与轨迹线曲率半径的关系如式(17)

$$R = \frac{1}{\kappa} \quad (17)$$

其中曲率 κ 表示曲线偏离直线的程度，定义为曲线上某点的切线方向角对弧长的微分，即

$$\kappa = \lim_{\Delta s \rightarrow 0} \left| \frac{\Delta \theta}{\Delta s} \right| = \frac{d\theta}{ds} \quad (18)$$

由式(17)可得轨迹线存在最小曲率半径 $R_{\min} = \frac{1}{\kappa_{\max}} \approx 0.4878m$ ，由此可以得到调头路径圆弧的最短半径。

对于圆心而言，则需要考虑无人车车身与调头区域边界的最小安全距离 $d_{CO\min}$ （本题中 $d_{CO\min} = 0.3m$ ），以及无人车初始位置 (x_i, y_i) 和初始方向角（由 5.2.2（2）知初始方向角可近似为 $\pi/2$ ，即无人车初始朝向平行于车道线）、调头区域边界位置、车身大小等。如图 6 所示，矩形 ABCD 为调头区域边界，其中由于点 A 和点 B 为调头区域最靠近无人车的两个顶点，无人车车身最有可能与之距离低于 $d_{CO\min}$ ，以点 A 和点 B 为圆心， $d_{CO\min}$ 为半径，分别作圆 A 和圆 B，则当车身进入圆 A 或圆 B，则说明规划的轨迹不安全，而当车身刚好与圆 A 或圆 B 相切或相离时，则说明轨迹安全，为尽量缩短通行时间，车身应刚好与圆 A 或圆 B 相切于点 $T(x_T, y_T)$ ，假设规划的圆弧路径圆心为点 Q，无人车控制点为点 P。

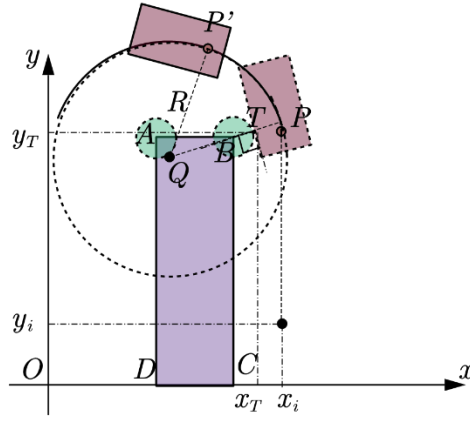


图 6 最小圆弧轨迹规划示意图

分析图 6 中，可得以下几何模型：

$$x_Q = x_i - R_{\min} \quad (19)$$

$$l_{BQ} = l_{PQ} - d_{CO\min} - \frac{w}{2} = R_{\min} - d_{CO\min} - \frac{w}{2} \quad (20)$$

$$l_{BQ}^2 = (y_B - y_Q)^2 + (x_B - x_Q)^2 \quad (21)$$

由式(17)，(19)~(21)，可求得规划出的最小圆弧路径圆心点 Q 的坐标 (x_Q, y_Q)

$$x_Q = x_i - \frac{1}{\kappa_{\max}} \quad (22)$$

$$y_Q = y_B \pm \sqrt{\left(\frac{1}{\kappa_{\max}} - d_{CO\min} - \frac{w}{2}\right)^2 - \left(x_B - \frac{1}{\kappa_{\max}} + d_{CO\min}\right)^2} \quad (23)$$

对于由式(23)中的±号，当 $x_i \leq x_B + d_{CO\min}$ 时，取+号，否则，取-号。

(3) 回旋曲线轨迹优化模型

由于无人车方向盘转速的限制，无人车行驶过程中，曲率不能发生突变，而从(1)中的直线行驶轨迹到(2)中的最小圆弧轨迹，曲率发生较大变化，而无人车仍在移动，所以需要一段过渡的轨迹。(2)中为简化模型而没有讨论该段过渡的轨迹，经过速度规划实验验证，(2)中的轨迹曲率突变无法在计算出的时间间隔内发生，所以需要本部分的回旋曲线轨迹优化模型^{[3][4][5]}对规划的轨迹进行修正。

如图 7 图 6(a)所示，回旋曲线被定义为满足式(24)的曲线。

$$\kappa(s) = \alpha s \quad (24)$$

如图 7 (b)所示，回旋曲线被定义为满足式(24)的曲线。结合式(18)和式(24)可得

$$\theta = \frac{1}{2} \kappa s = \frac{1}{2} \alpha s^2 \quad (25)$$

因此，有

$$s = \frac{\kappa}{\alpha} = \frac{2\theta}{\kappa} = \sqrt{\frac{2\theta}{\alpha}} \quad (26)$$

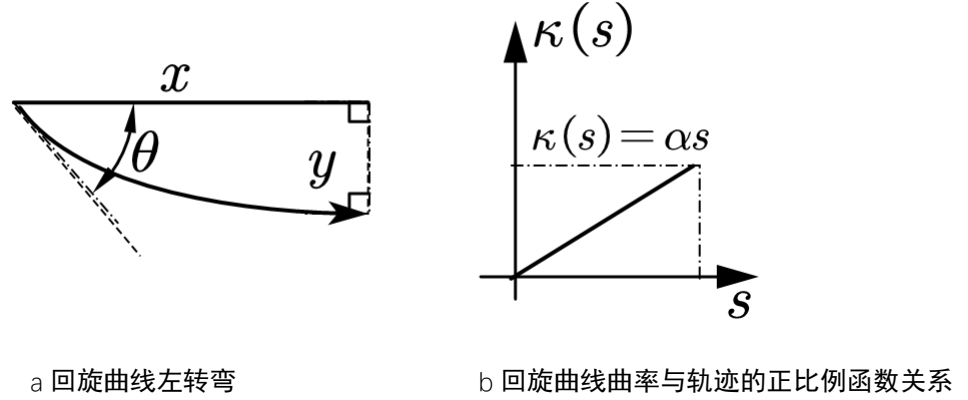


图 7 回旋曲线性质示意图

(4) 混合 A*搜索轨迹优化模型^[6]

A*搜索算法是一种启发式最短路径搜索算法，不仅利用初始状态到当前状态节点 n 的代价函数 $g(n)$ ，还利用当前状态节点 n 到目标结点最低代价的估计函数 $h(n)$ ，搜索到目标结点的最短路径，其从初始状态经由状态到目标状态 n 的代价估计可表示为

$$f(n) = g(n) + h(n) \quad (27)$$

利用 A*搜索算法，可以在栅格化后的场景搜索得到无人车调头的最优轨迹。然而，在格子数量过多时，A*搜索算法处理的结点数目过多，会导致算法时间复杂度大大增加，用时也大大增加，因此需要一种算法使轨迹收敛到更小的解空间中。本文采用 Reeds-Shepp 曲线模型^[7]对 A*搜索算法规划到的轨迹作进一步的收敛优化。

5.3.2 车辆速度规划模型^[8]

建立无人车调头轨迹规划模型后，为了规划无人车在每一个轨迹点的速度、加速度等信息，以实现预先规划好的路径，需要建立车辆速度规划模型。

文献[8]提出一种给定最大加速度、初始位置和速度、目标位置和速度，就能规划出汽车在特定轨迹上运动的最短时间速度序列的车辆速度规划模型如下：

$$\varphi = \sqrt{\frac{1}{a_{\max}}} \quad (28)$$

$$\beta = \varphi a_{\max} \quad (29)$$

$$\tau = \beta t \quad (30)$$

$$z_1 = \varphi \beta s \quad (31)$$

$$z_2 = \varphi \frac{ds}{dt} \quad (32)$$

引入控制变量 $u \in [-1, +1]$ ，有

$$f_t/F_{\max} = u\sqrt{1 - (z_2^2/R(z_1))^2} \quad (33)$$

其中

$$m \frac{d^2 s}{dt^2} = f_t, \quad \frac{m \left(\frac{ds}{dt} \right)^2}{R(s)} = f_n \quad (34)$$

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = u\sqrt{1 - (z_2^2/R(z_1))^2} \quad (35)$$

$$H = 1 + \lambda_1 z_2 + \lambda_2 u\sqrt{1 - (z_2^2/R(z_1))^2} \quad (36)$$

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial z_1} = -\lambda_2 u \frac{z_2^4}{\sqrt{1 - (z_2^2/R(z_1))^2}} \frac{R'}{R^3} \quad (37)$$

$$\dot{\lambda}_2 = -\frac{\partial H}{\partial z_2} = \lambda_1 + 2\lambda_2 u \frac{z_2^3}{R^2 \sqrt{1 - (z_2^2/R(z_1))^2}} \quad (38)$$

$$R' = \frac{\partial R}{\partial z_1} \quad (39)$$

$$u^* = -\operatorname{sgn} \lambda_2 \quad (40)$$

5.3.3 问题 1 模型的求解

问题一提供的场景为简单的车辆调头问题，道路上没有障碍物。仅有将双向车道分离的障碍物。选取最左侧车道为目标车道，车道底边线上贴近车道线且留有安全距离的点作为轨迹规划目标点，根据轨迹优化模型对问题进行求解。得到的轨迹为直线、圆弧、直线、圆弧、直线。具体的轨迹结果如图 8。

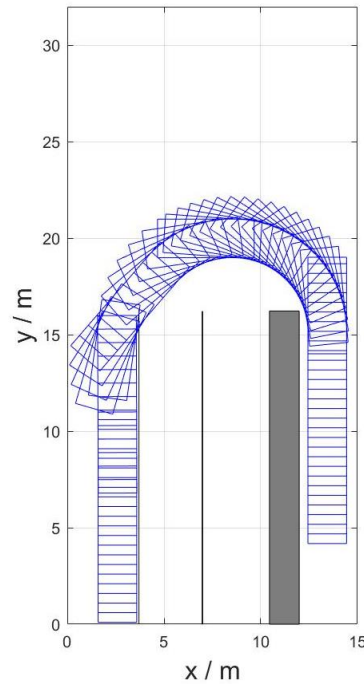


图 8 运动轨迹

根据图 8 求得的轨迹较好的符合了优化目标，在通行效率尽可能高的情况下减少不必要的压线。该轨迹首先通过一段最大要求曲率的圆弧，由于场景限制，如若选择中间车道作为目标车道，则会产生压车道线的情况。因此选择车道三作为目标车道可以在保证较少压线的同时保证较高的通行效率。

根据速度与轨迹，得出轨迹上各点的时间，部分数据如表 3：

表 3 问题 1 轨迹数据图

X	Y	位移	速度	时间	速度差	加速度	曲率	方位角
6.28	1.5708	0.5	2.44949	0.204124	0.71744	3.514724	0	13.44
6.78	1.5708	0.5	3	0.166667	0.55051	3.30306	0	13.44
7.28	1.5708	0.5	3.4641	0.144338	0.4641	3.215378	0	13.44
7.78	1.5708	0.5	3.87298	0.1291	0.40888	3.167168	0	13.44
...
6.8264	-1.5708	0.5	4.47214	0.111803	-0.52786	-4.72133	0	2.76857
6.3264	-1.5708	0.5	3.87298	0.1291	-0.59916	-4.64107	0	2.76857
5.8264	-1.5708	0.5	3.16228	0.158114	-0.7107	-4.49486	0	2.76857
5.3264	-1.5708	0.5	2.23607	0.223607	-0.92621	-4.14214	0	2.76857

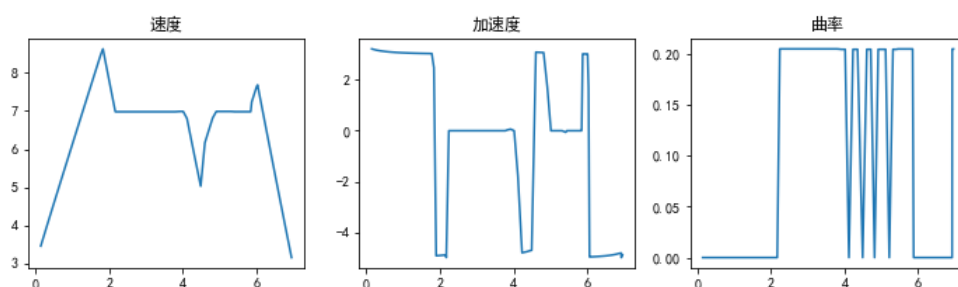


图 9 运动参数变动图

根据图中数据结果可以看出，加速度、速度，曲率等各项指标变换均较为平缓，规划结果满足良好的驾乘体验与通行效率。

5.4 问题 2 模型的建立与求解

问题二题目给出的场景为道路中无障碍物的场景。需要算法解出什么情况下至少需要一次调头才能够顺利调头，首先通过采用得出可能的三种情况如图 11。

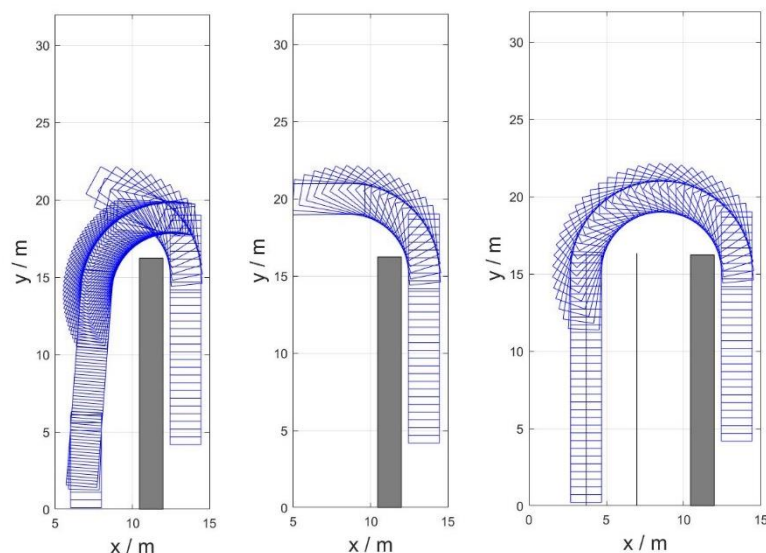


图 10 规划结果示例

根据圆弧轨迹的规划模型可知，当车辆轨迹为圆弧且曲率最大时转弯效率最大。因此，当车辆以完全圆弧轨迹调头时，若车辆的外形的最大包络线与墙体边缘相切时刚好可以完成调头动作。如若对向车道距离较小，则至少需要一次倒车才能掉头。

因此，当无人车转弯轨迹为最大曲率圆弧且车辆调头完成时刻恰好与左边障碍物相距一个安全距离时为临界条件，通过计算此时 A 左边框与 D 右边线距离 8.2 米。因此当两者距离小于 5.2 米时，就至少需要一次倒车才能完成调头（如图 3 左 1），否则就会与 A 相撞（图 3 右 1）。

5.5 问题 3 模型的建立与求解

问题三场景要求考虑对向车道存在多个障碍物的情况，使用模型进行求解后结果如图 11。

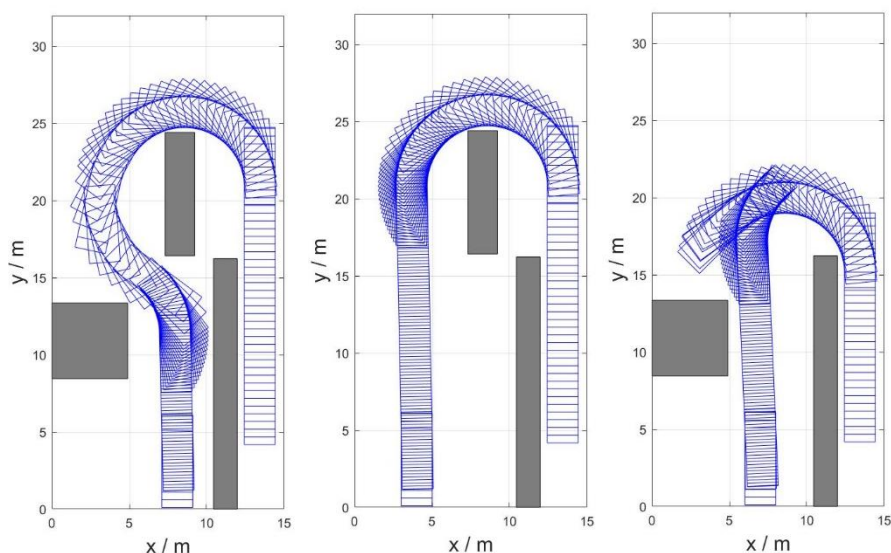


图 11 多个障碍物时轨迹规划结果

根据轨迹优化结果可以看出，当 G、F 都存在时，轨迹通过优化可以得出较为平滑的曲线。将路径点带入速度规划模型，求得各项如表 4：

表 4 轨迹参数表

X	y	位移	速度	时间	速度差	加速度	曲率	方位角
13.44	6.28	0.5	1.73205	0.288675	1.73205	5.99999	0	1.5708
13.44	6.78	0.5	2.44949	0.204124	0.71744	3.51472	0	1.5708
13.44	7.28	0.5	3	0.166667	0.55051	3.30306	0	1.5708
13.44	7.78	0.5	3.4641	0.144338	0.4641	3.21538	0	1.5708
...
3.99129	5.48318	0.2	3.85132	0.05193	-0.25144	-4.84187	0	4.73245
3.9953	5.28322	0.2	3.58227	0.055831	-0.26905	-4.81904	0	4.73245
3.99902	5.09787	0.185387	3.31342	0.05595	-0.26885	-4.80514	0	4.73245
3.99929	5.08326	0.014613	3.2913	0.00444	-0.02212	-4.98228	0.205	4.72946

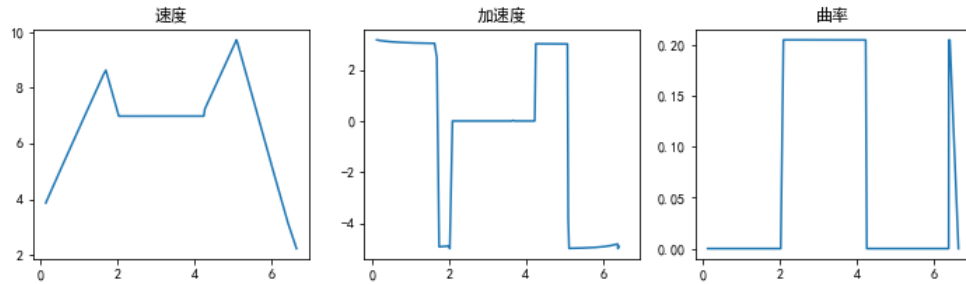


图 12 运动参数变化图

根据图中数据结果可以看出，加速度、速度，曲率等各项指标变换均较为平缓，规划结果满足良好的驾乘体验与通行效率。

5.6 问题 4 模型的建立与求解

第四问求解要求考虑道路上存在的斑马线，通过对题目分析可以得知有两种方式在不压斑马线的情况下完成调头：①不经过斑马线，在斑马线前完成调头。②经过斑马线后再进行调头。现针对场景带入模型求解，发现针对题目三与题目四中的两种场景，方法①均无法完成。因此对于两种场景均使用方法②，及限制无人车完全开过斑马线后再开始调头，效果如图 13。

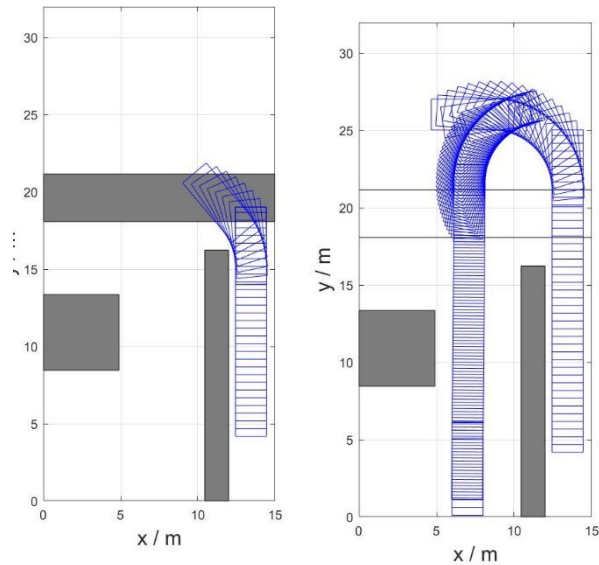


图 13 两种避免压到斑马线规划方法的轨迹

将路径结果导入速度规划模型中，并计算各轨迹点的时间。各项规运动参数如

图 14 所示。

表 5 轨迹参数表

X	y	位移	速度	时间	速度差	加速度	曲率	方位角
13.44	6.28	0.5	1.73205	0.288675	1.73205	5.99999	0	1.5708
13.44	6.78	0.5	2.44949	0.204124	0.71744	3.51472	0	1.5708
13.44	7.28	0.5	3	0.166667	0.55051	3.30306	0	1.5708
13.44	7.78	0.5	3.4641	0.144338	0.4641	3.21538	0	1.5708
...
3.99129	5.48318	0.2	3.85132	0.05193	-0.25144	-4.84187	0	4.73245
3.9953	5.28322	0.2	3.58227	0.055831	-0.26905	-4.81904	0	4.73245
3.99902	5.09787	0.185387	3.31342	0.05595	-0.26885	-4.80514	0	4.73245
3.99929	5.08326	0.014613	3.2913	0.00444	-0.02212	-4.98228	0.205	4.72946

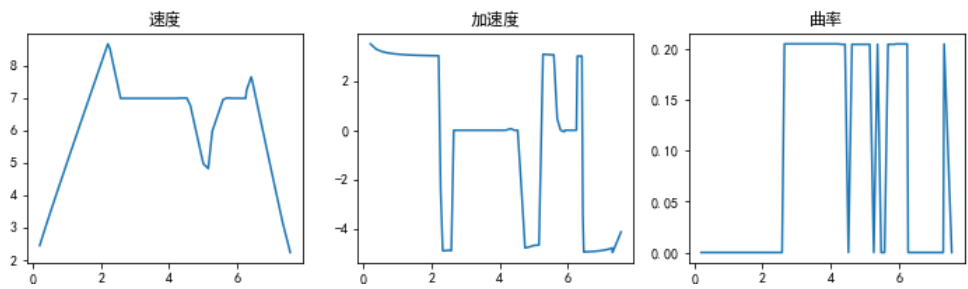


图 14 运动参数变化图

根据图中数据结果可以看出，加速度、速度，曲率等各项指标变换均较为平缓，规划结果满足良好的驾乘体验与通行效率。

5.7 问题 5 模型的建立与求解

问题要求考虑障碍物移动的状态，根据障碍物的速度，将场景分为三种情况：①障碍物低速场景 ②障碍物中速场景 ③障碍物高速场景。下面分别对三种场景进行轨迹优化分析。

对于障碍物的低速场景，此时障碍物虽为运动状态，但速度较为缓慢，可以将障碍物等效为静止的几何形状扩大后的障碍物。基于此对规划结果进行仿真，结果如 15。

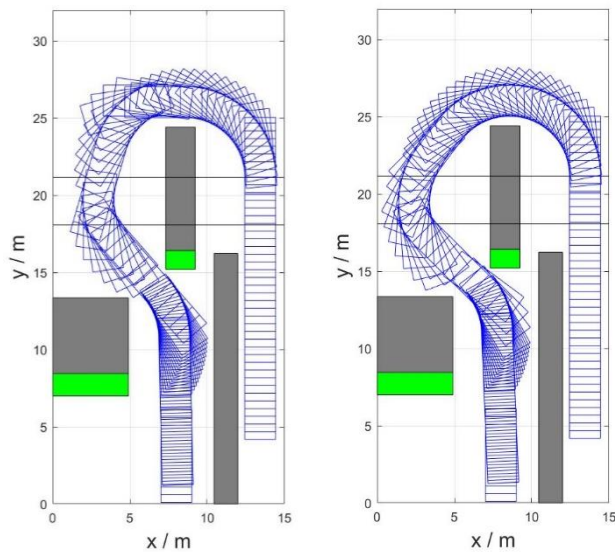


图 15 障碍物低速状态下的轨迹优化结果

对于障碍物中等移动速度场景，此时障碍物移动速度与无人车速度接近，如图 16 不适宜进行超车。需要进行等待。

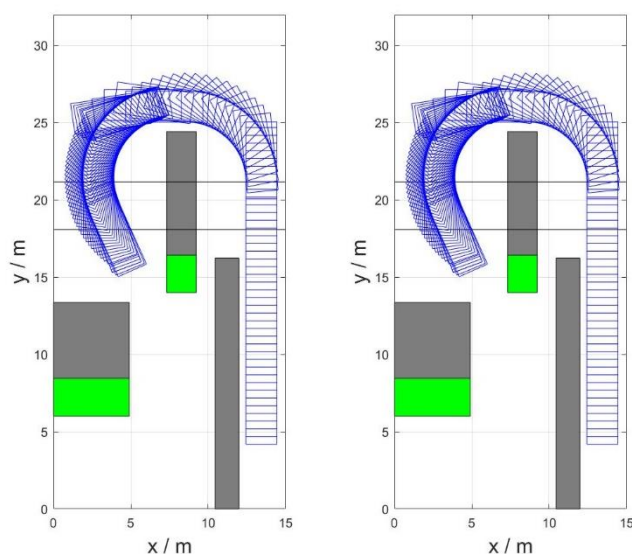


图 16 中等速度障碍物下的轨迹优化结果

对于障碍物高速移动的场景，此时障碍物移动速度大幅超过无人车速度，不对无人车轨迹产生影响，可直接进行轨迹规划。轨迹如图 17。

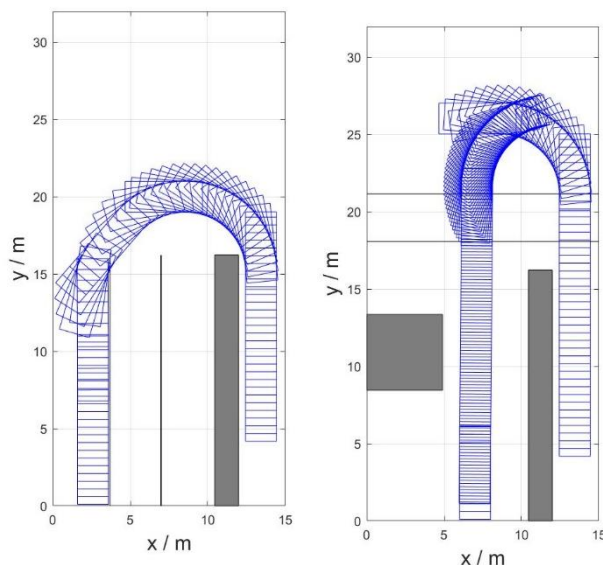


图 17 高速障碍物下轨迹优化结果

5.8 问题 6 模型的建立与求解

关于求解耗时的减小优化，混合 A*算法将 A*算法与 RS 曲线结合，前者搜索空间较大，求解耗时容易过长，而后者大大减少了 A*算法的解空间，明显减小了求解耗时，使无人车能以尽可能高的频率进行计算，从而对路况进行快速反应。

而关于求解成功率即收敛性和稳定性的优化，根据对前面五个问题的求解验证，混合 A*算法的搜索空间较小，多级轨迹优化模型可以在极短的时间内搜索得较优解。但该算法的求解的收敛性不强，针对不同的优化参数，求得的解有较大的波动。

解的收敛性和稳定性差是由于混合 A*搜索出的路径平滑性较差，可以对得出的 RS 曲线进行平滑处理，处理后路径求解成功率会提升且解易于收敛。

5.9 结果总结

多级轨迹优化模型与最大速度预测模型较好的对无人车的轨迹进行了优化，具体的方法如下：

1. **对向车道无障碍物场景：**选取最左侧道靠近车道线的底边线点作为目标点，直接使用多级规划模型进行优化，得到了图 9 的优化结果。
2. **对单一的调头场景：**考虑临界情况：使用最小圆弧优化模型，求得了无需倒车的临界情况。并使用多级优化算法对需要调头的场景进行了求解。结果如图 10
3. **针对对向车道存在障碍物的场景：**使用多级优化算法，考虑了不同障碍物存在情况下轨迹的优化算法，结果如图 11。
4. **针对斑马线存在的场景：**讨论了不压斑马线的两种情况，并得出了针对题目中得场景，如要转弯时不压斑马线，需要先驶过斑马线再进行调头。并对轨迹进行了求解，求解结果如图 13。
5. **针对障碍物移动的场景：**根据障碍物的速度不同，讨论并分析了三种场景。并对三种不同的场景给出了模拟。多级轨迹优化算法效果较好。结果如图 15。

六、模型评价与推广

6.1 模型优点

- a) 模型首先对倾斜的地图进行坐标转换和近似处理，简化了道路环境数据，更便于进行无人车调头轨迹规划求解。
- b) 模型将调头问题分为了多段任务，使用设计的标准轨迹与混合 A*搜索轨迹共同完成轨迹规划。相比于完全使用混合 A*算法对全局进行搜索，提高了求解成功率与求解效率。
- c) 使用了最大速度规划算法，在保证速度满足横向加速度与最大方向盘转速的前提下使速度变化平滑，提高了驾乘体验与通行效率。
- d) 对于平行于路面的障碍物，为便于对其分析。将移动的障碍物等效于静止的障碍物，等效障碍物的体积与原障碍物尺寸与障碍物移动速度相关。具有较好的泛化性能。

6.2 模型缺点

- a) 模型在对轨迹进行规划时选取的路径均为约束条件的边界值，对抗环境噪声的能力较差。
- b) 在对动态障碍物进行分析时没有考虑其往复运动的情况。

6.3 模型推广

- a) 模型提出了多级规划的思想，该规划思想可应用至多种无人车标准测试场景中，提高轨迹规划的效率。
- b) 模型提出的使用求解最优转弯路径的方法，可拓展至其它无人车转弯限制转弯场景中。
- c) 模型提出的针对动态障碍物的等效方法，可应用至其它场景中动态障碍物的避

障。

七、模型的改进

- a) 模型中使用的混合 A*算法求得路径后没有对求得 RS 进行平滑, 后续可对 RS 曲线进行平滑处理。
- b) 图形栅格化时, 选取的栅格数量为 25*15。后续研究时可合理增加栅格密度, 以提高算法的准确性。
- c) 对路径进行规划时仅使用了混合 A*进行搜索。后续研究可使用多种搜索算法对比, 选取效果最好的优化算法。
- d) 可以将状态参数与轨迹优化算法结合。状态参数估计可以更加准确获得车辆参数, 因此可以将状态估计其加入规划模块, 将状态信息反馈给优化算法, 提高轨迹质量。
- e) 后续研究考虑将算法与强化学习等前沿人工智能算法相结合, 对算法进行改进。

八、参考文献

- [1] 余卓平, 李奕姗, 熊璐. 无人车运动规划算法综述[J]. 同济大学学报(自然科学版), 2017, 45(8): 1150-1159.
- [2] 龚建伟. 无人驾驶车辆模型预测控制[M]. 北京理工大学, 北京理工大学出版社, 2014.
- [3] H. Vorobieva, S. Glaser, N. Minoiu-Enache and S. Mammar, "Automatic Parallel Parking in Tiny Spots: Path Planning and Control," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 1, pp. 396-410, Feb. 2015.
- [4] D. K. Wilde, "Computing clothoid segments for trajectory generation," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 2009, pp. 2440-2445.
- [5] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous curvature paths," IEEE Trans. Robot., vol. 20, no. 6, pp. 1025-1035, Dec. 2004.
- [6] DOLGOV D, THRUN S, MONTEMERLO M, et al. Practical search techniques in path planning for autonomous driving[J]. Ann Arbor, 2008, 1001(48105):18-80.
- [7] REEDS J, SHEPP L. Optimal paths for a car that goes both forwards and backwards[J]. Pacific Journal of Mathematics, 1990, 145(2).
- [8] Velenis, E. , and P. Tsiotras . "Optimal velocity profile generation for given acceleration limits: Theoretical analysis." American Control Conference, 2005. Proceedings of the

2005 IEEE, 2005.

九、附录

9.1 使用的软件

1. 文字处理软件：Word 2019。
2. 编程软件：MATLAB R2019b。
3. 数据处理软件：MATLAB R2019b。
4. 画图软件：AxGlyph, MATLAB R2019b。
5. 公式编辑软件：AxMath。

9.2 相关的源程序代码

1. Matlab 轨迹规划主程序

```
left = 0;
right = pi;
start_point = [13.44, 5.28];
R = 1/0.205;

center = getCircle_R([12, 16.24], 13.44, 2, 0.3, R);
turning_point = [start_point(1), center(2)]

while right - left > 0.01
    mid = (left + right) / 2

    vehicle_TPBV_.x0 = turning_point(1) - (R - R * cos(mid));
    vehicle_TPBV_.y0 = turning_point(2) + R * sin(mid);
    vehicle_TPBV_.theta0 = mid + pi/2;

    [x, y, theta, path_length, completeness_flag] =
SearchHybridAStarPath();

    if (completeness_flag)
        left = mid;
    %     break
    else
        right = mid;
    end
end

for i2 = start_point(2):0.5:turning_point(2)
    V = CreateVehiclePolygon(start_point(1), i2, pi/2);
```

```

    track(end+1, :) = [start_point(1), i2, pi/2];
    plot(V(:,1), V(:,2), 'b'); drawnow
end

for the = pi/2:0.1:vehicle_TPBV_.theta0
    xx = turning_point(1) - (R - R * cos(the - pi/2));
    yy = turning_point(2) + R * sin(the - pi/2);

    V = CreateVehiclePolygon(xx, yy, the);
    track(end+1, :) = [xx, yy, the];
    plot(V(:,1), V(:,2), 'b'); drawnow
end

for ii = 1 : length(x)
    if abs(theta(ii) - 3 * pi/2) < 0.1
        break;
    end

    V = CreateVehiclePolygon(x(ii), y(ii), theta(ii));
    track(end+1, :) = [x(ii), y(ii), theta(ii)];
    plot(V(:,1), V(:,2), 'b'); drawnow
end

for j = y(ii):-0.5:3.9
    V = CreateVehiclePolygon(x(ii), j, -pi/2);
    track(end+1, :) = [x(ii), j, -pi/2];
    plot(V(:,1), V(:,2), 'b'); drawnow
end

xlswrite('Q1.xlsx', track)

```

2. GetCircle_R.m

```

function [center] = getCircle_R(wall, ctrl_point_x, width,
safe_dist, R)
% center      - center of the circle
% wall        - coordinate of wall
% ctrl_point_x - x coordinate of control point
% width       - car width
% safe_dist   - safety distance
% R           - minimum radius (=1/K)

center = [0, 0];

```

```
center(1) = ctrl_point_x - width/2 - R;  
  
hyp = R - safe_dist;  
dx = R - (ctrl_point_x - width/2 - wall(1));  
center(2) = wall(2) - sqrt(hyp^2 - dx^2);  
end
```

3. Matlab 所用函数说明

所用到的函数均已在附录给出，此处限于篇幅不全部给出。