题目：自动驾驶中的车辆调头问题

队伍编号：201847100213

指导老师：

自动驾驶中的车辆调头问题

摘 要

本文针对无人车自动驾驶时的调头路线轨迹的确定问题，并基于单目标优化模型、差值拟合算法，求出了车辆调头时无障碍物、有障碍物、是否需进行调头情况、是否通过斑马线情况时的轨迹路线。

对于无障碍物时轨迹路线的确定，首先讨论无人车辆调头的理想路线，在此理想路线上，建立单目标模型进行插值拟合，计算能否得到满足于约束条件的轨迹路线。

对于是否需要进行调头情况的讨论，在前文的基础上，讨论无人车刚好不需要进行调头的路线，确定临界条件，然后同样建立单目标模型，计算满足于约束条件的轨迹路线。

对于存在不同障碍物的情况，分别观察和分析障碍物的情况，找出能够行驶的路线，建立多目标模型，计算得出结果判断是否可行。

对于人行横道的讨论，要考虑不越过和越过两种情况，同时还有障碍物是否存在的情况，当然这与上述只有障碍物的路线相似，只不过多了人行横道这一约束条件，通过建立相关的模型，对可能的路线进行判断，找出能顺利通过的情况。

对于障碍物移动的情况，考虑提出动态路径规划的技术，根据先验环境模型找出从起点到终点中符合条件的最优或次优路径，涉及的根本问题是环境模型的表达和路径搜寻策略。通过使用人工势场法、启发式搜索算法和基于离散优化的方法等进行求解

关键词：单目标优化模型　差值拟合算法　动态路径规划的技术

1 问题重述

1.1 前言

自动驾驶是近年人工智能应用的热门研究领域之一，其中调头是自动驾驶中一个非常实际又很有趣的场景。本文所需要解决的问题便是无人驾驶车辆的自动调头轨迹的确定。无人车的轨迹，指的是一条含有位置和时间等信息的曲线，它由一列轨迹点构成，各个轨迹点中应包含位置坐标、方向角、曲率、曲率变化率、速度、加速度及运动到此点时的时间等信息。这里轨迹指的是车身中某一特定点的轨迹，即控制点。

1.2 相关参数

假设无人车为四轮乘用车，采用前轮转向后轮驱动; 车身可认为是一个矩形，车长 5 米，车宽 2 米，轴距 2.8 米; 方向盘最大转角 470，方向盘与前轮转角的传动比为 16:1（方向盘每转动 16°, 前轮转动 1)，方向盘最大转速为 400°/s; 最大油门加速度 3 m/s^2, 极限杀车加速度-5m/s^2 。

1.3 限制条件

无人车行驶时需要满足以下动力学条件：

（1）任何点的加速度不得高于最大油门加速度，不得低于极限刹车减速度；

（2）无人车朝向和方向盘转角都不能突变，轨迹线必须是连续、且切线方向连续，当无人车不是静止时，曲率也必须连续（静止时可以原地打方向，非静止时不能）；

（3）轨迹线的最大曲率应尽量不高于 0.205，禁止高于 0.21；

（4）按轨迹行驶时，在调头完成之前，无人车车身任何点不得与任何障碍物或者调头区域边界发生碰撞，且与障碍物至少保留一个最小安全距离，一般不小于 30cm。

无人车调头轨迹在保证上述条件之外也应具备尽可能舒适的驾乘体感，以及具备尽可能高的通行效率，同时出于规范驾驶的考虑，尽量减少不必要的压车道线行驶。

1.4 解决问题

问题 1：如图 1 所示场景，无人车正在最左侧车道准备调头，存在三个对向车道。当无人车身触碰到对向的调头区域边界时，认为调头完成。请阐述控制点位置，解释其理由；基于此建立无人车调头的数学模型，并给出一个合理可行的算法设计，给出调头轨迹。

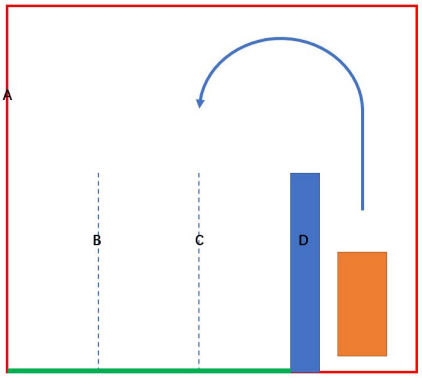

图 1: 无人车调头场景 1

问题 2：当调头区域狭窄时，判断什么样的场景下无人车能够在不倒车的情况下完成调头，什么情况需要至少一次倒车才能通过。

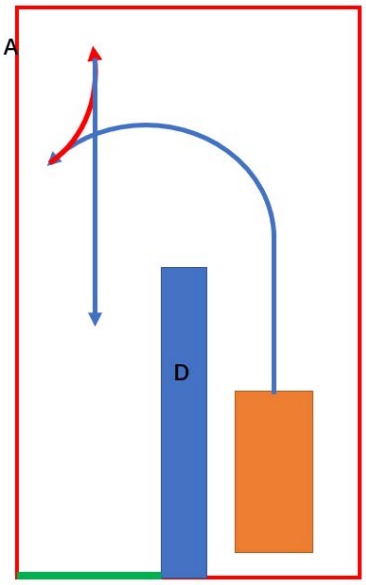

图 2: 无人车调头场景 2

问题 3：如图 3 场景，如果道路上还存在其它静止障碍物，当仅存在 F、仅存在 G、或二者都存在时，建立相应的无人车调头的数学模型，给出合理的算法设计，并给出调头轨迹，你的算法应明确如何进行避障？

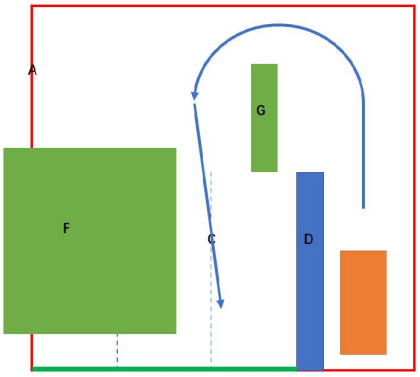

图 3: 无人车调头场景 3

问题 4：如果道路上存在人行横道即斑马线 E，则无人车需要尽可能在不压人行横道的情况下过，如果无法做到，则需要越过人行横道后再开始调头，禁止骑人行横道调头。建立相应的无人车调头的数学模型，给出合理的算法设计和相应的调头轨迹；当斑马线和障碍物同时存在时（如图 5 所示），你的模型和算法如何同时满足交规并进行避障绕行？

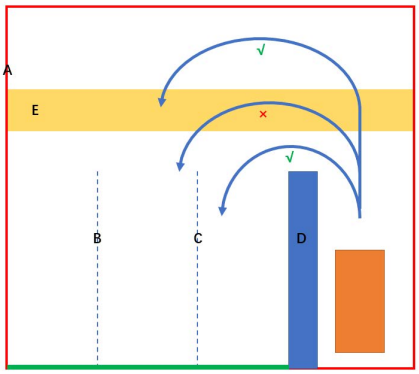

图 4: 无人车调头场景 4

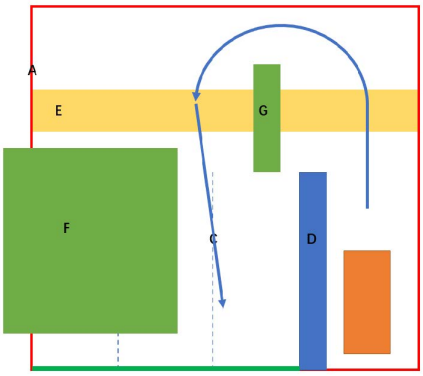

图 5: 无人车调头场景 5

问题 5：无人车在实际路况行驶中进行调头时，道路中的障碍物通常是处于移动状态，如对向来车等，针对图 3 和图 5 的两种场景，假设图中标识的障碍物位置为无人车处在起始时刻的初始位置，障碍物以一定的速度移动，假设只考虑与道路平行方向移动，请建立更一般的无人车调头轨迹规划模型并给出相应算法设计和无

人车调头策略，给出仿真结果。

问题 6：为保证无人驾驶的安全性，无人车的轨迹规划算法必须拥有尽可能高的求解成功率，同时为了能对路况进行快速反应，需要以尽可能高的频率进行计算。你的算法如何在求解成功率和求解耗时这两方面优化？

2 问题分析

2.1 问题一

对于问题一，由于不需要考虑人行横道和其它障碍物等约束条件，这里可以先计算出无人车调头的理想化轨迹，再根据无人车的运动特点输入参数进行拟合。

2.2 问题二

对于问题二，当调头轨迹狭窄时，无人车需要通过倒车才能完成调头，这里其实是对于临界问题的考虑——即无人车刚好不需要通过倒车就能完成调头，所以当无人车的运动状态超过这个临界值时便需要至少一次倒车才能通过。

2.3 问题三

对于问题三，分别要考虑仅存在障碍物 F、仅存在障碍物 G 和两者都存在的这三种情况。首先先观察障碍物位于何处，直观给出可以行驶通过的路线，再通过编程找出正确的路线。

2.4 问题四

对于问题四，增加了个人行横道的条件，要分别考虑在有无障碍物 F 和 G 时不经过人行横道和越过人行横道两种情况。在无障碍物时，只需考虑在人行横道处无人车的调头情况，即圆的半径大小；在有障碍物时，应先观察，找出可能行驶的路线，再通过计算编程得出结果。

2.5 问题五和问题六

对于移动的障碍物，使用动态路径规划技术，按照某些性能指标在其行驶区域内搜索一条从起点到终点的无碰撞的最优或近似最优路径，其本质是一个具有约束的复杂系统优化问题。

3 模型假设与约定

1. 不考虑转弯时的向心力影响；

2. 地面摩擦力足够大；

3. 假定方向盘转角就是前轮胎当前的转角；

4. 忽略悬架系统对车辆动力学特性的影响 [1]；

5. 忽略动力传动系统的影响；

6. 忽略空气动力学以及轮胎纵横向力的耦合关系对车辆动力学特性的影响；

4 符号说明与名词定义

符号	名词
k	曲率
r	曲率半径
L	轴距
t	调头所用总时间
d	障碍物与车辆距离
α	前轮转动角
ε	前轮移速
$\bar{v}$	平均速度
l	轨迹长度

5 模型的建立与求解

5.1 多车道型

5.1.1 控制点位置的选取

控制点选取在车后轮中点处，选取在这里有两个原因：

（1）此时控制点的轨迹方向与车辆的速度方向一致；

（2）便于讨论理想化轨迹。

5.1.2 曲率半径的约束条件

由于轨迹线的最大曲率应尽量不高于 0.205，禁止高于 0.21，为了安全考虑，曲率 k 应满足：

$$k < 0.205 \tag{1}$$

所以曲率半径应满足：

$$r = \frac{1}{k} > 4.878\text{m} \tag{2}$$

5.1.3 无人车理想调头路线过程分析

无人车的调头路线分为三段，第一段为直行，第二段为弧形（向左转 180° 即半个圆），第三段为直行。

第一段直行路线的起点即为无人车的初始位置，第一段和第二段的交点应为过起点且平行于障碍物 D 的直线与障碍物 D 横向直线的交点；然后无人车行驶 180° 后到达第二段与第三段的交点；第三段的终点应先计算出边框 A 与两条车道线和障碍物左侧的交点坐标，得到的相邻的交点坐标的中点即为最佳调头结束点。

5.1.4 计算关键交点的坐标

这里需要计算许多条直线的交点坐标，以计算边框 A 与车道线 B 的交点坐标为例：

$$A_1\left(a_1, a_2\right); A_2\left(a_3, a_4\right); B_1\left(b_1, b_2\right); B_2\left(b_3, b_4\right)$$

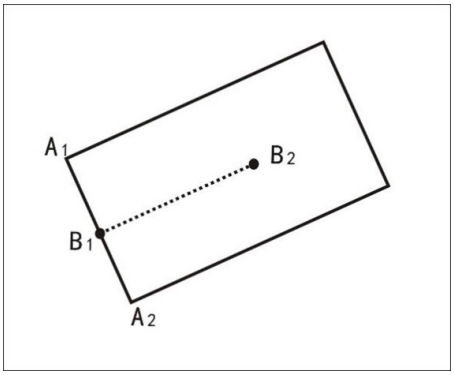

图 6: 交点坐标的计算

得到直线 A、B 的方程为:

$$l_A : \frac{(x-a_1)}{(a_2-a_1)} = \frac{(y-a_3)}{(a_4-,a_4)} \tag{3}$$

$$l_B : \frac{(x-b_1)}{(b_2-b_1)} = \frac{(y-b_3)}{(b_4-,b_4)} \tag{4}$$

联立方程:

$$\begin{cases} \dfrac{(x-a_1)}{(a_2-a_1)} = \dfrac{(y-a_3)}{(a_4-,a_4)} \\ \dfrac{(x-b_1)}{(b_2-b_1)} = \dfrac{(y-b_3)}{(b_4-,b_4)} \end{cases}$$

将相关点的坐标带入求解即可得交点坐标。这里设边框 A 的左顶点，边框 A 与车道线 B、边框 A 与车道线 C、边框 A 与障碍物 D 左侧交点分别为：H_1, H_2, H_3, H_4。所以无人车调头之后的最佳调头结束点为：

$$Z_1 = \frac{H_1+H_2}{2}; \quad Z_2 = \frac{H_2+H_3}{2}; \quad Z_3 = \frac{H_3+H_4}{2}$$

同理可求出其它交点坐标，这里设起点坐标、第一段与第二段的交点坐标分别为：X_1 X_2；

第二段和第三段的交点坐标有三个，分别与调头结束点相对应设为：Y_1 Y_2 Y_3。

5.1.5 行车轨迹

由以上可知，无人车的驾驶轨迹为：

路线 1：$X_1 \to X_2 \to Y_1 \to Z_1$，此时轨迹长度为 l_1；

路线 2：$X_1 \to X_2 \to Y_2 \to Z_2$，此时轨迹长度为 l_2；

路线 3：$X_1 \to X_2 \to Y_3 \to Z_3$，此时轨迹长度为 l_3；

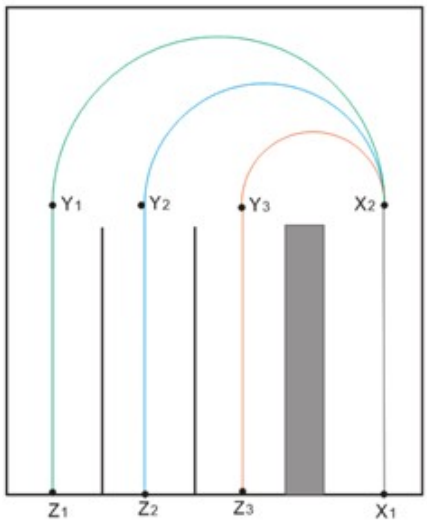

图 7: 多车道调头轨迹示意图

5.1.6 数学模型的建立

设无人车在轨迹中某一点的位置为 x_0，时刻为 t_0，设置自动驾驶调头所用总时间为 t，在某一时刻障碍物与无人车的距离为 d_0，在某一时刻轨迹点的曲率为 k_0，于是构建单目标优化模型 [2]。

决策变量为：x_0

目标函数为：$mint = L/\bar{v}$, $L = l_1$、l_2、l_3，$\bar{v}$ 为行驶时的平均速度。

约束条件为：

设置汽车的加速度为 a，由于任何点的加速度不得高于最大油门加速度，不得低于极限刹车减速度，所以得到：

$$-5 \leq a \leq 3 \tag{5}$$

车辆与障碍物至少保留一个最小安全距离，一般不小于 30cm；

$$d_0 \geq 30 \tag{6}$$

方向盘与前轮转动角的传动比为 16:1，方向盘最大转角 470° 方向盘最大转速

为 400°/s; 设置前轮的转动角为 α ，转速为 ε ，所以有:

$$\alpha = \frac{470}{16} \tag{7}$$

$$\varepsilon < \frac{400}{16} \tag{8}$$

车轮轴距 L=2. 8m，则有:

$$R = \frac{L}{\tan\alpha} \tag{9}$$

由以上可知曲率半径应满足:

$$r > 4.878 \tag{10}$$

综上得:

$$\begin{cases} c-5 \leqslant a \leqslant 3; \\ d_0 \geq 30 \\ \alpha = \dfrac{470}{16}, \varepsilon < \dfrac{400}{16}; \\ R = \dfrac{L}{\tan\alpha} \\ r > 4.878 \end{cases}$$

5.1.7 求解结果

由以上约束条件，将无人车调头过程分为三段，其中第二段和第三段为直行路段，第二段为转弯路段，这里是采用插值拟合的方法，先讨论出合适的调头路线，再进行建立目标优化模型，得到无人车的调头轨迹为：

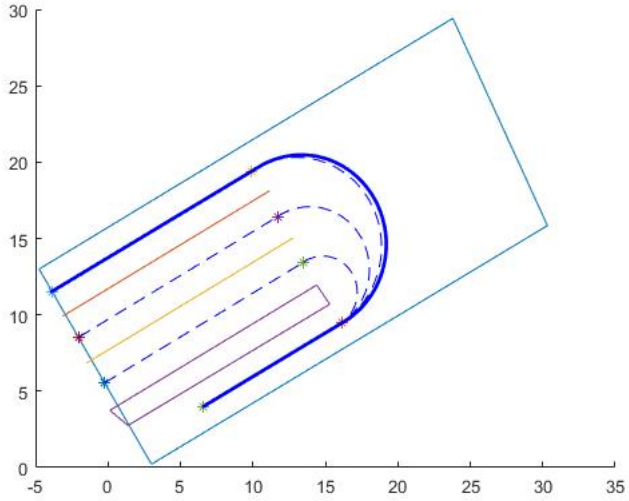

图 8: 无人车调头轨迹图

5.2 倒车型

5.2.1 转弯半径的确定

这里需要求解无人车在调头时需要进行调头的情况，则在这里讨论无人车刚好不需要进行调头，此时的转弯半径 r 变为最小半径，有：

$$r = 4.878 \tag{11}$$

5.2.2 无人车理想调头路线过程分析

无人车的调头路线分为三段，第一段为直行，第二段为弧形（向左转 180° 即半个圆），第三段为直行。

第一段直行路线的起点即为无人车的初始位置，第一段和第二段的交点仍然为过起点且平行于障碍物 D 的直线与障碍物 D 横向直线的交点；然后无人车向左转弯 180°，转弯半径 r=4.478m 到达第二段与第三段的交点；然后车辆沿着与车道线平行的方向行驶，到达调头区域，调头完成。

5.2.3 关键坐标的计算

第一段的起点坐标仍然为 X_1 第二段与第三段的交点坐标 Y_4 为求解过程如下：

记

$$tan\alpha_1 = k_{l_{D_2D_3}} \quad \alpha_2 = \pi - \alpha_1$$

$$Y_4\left(x_{21} - 2r\cos\alpha_2, \quad x_{22} + 2r\sin\alpha_2\right)$$

同理可求得调头的终点，设为 Z_4。

5.2.4 驾驶轨迹

此时无人车的驾驶轨迹为: $X_1 \to X_2 \to Y_4 \to Z_4$; 此时轨迹长度为 l_4。

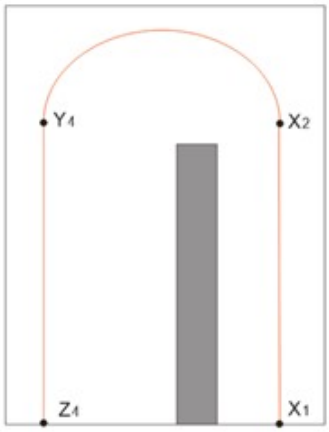

图 9: 倒车型调头轨迹示意图

5.2.5 模型建立

该模型与第一问的模型基本一致，唯一改变的为曲率半径，则目标函数仍然为:
目标函数为:

$$\min t = \frac{L}{\bar{v}} \tag{12}$$

$$L = l_4 \tag{13}$$

约束条件为:

$$\begin{cases} -5 \leqslant a \leqslant 3; \\ d_0 \geq 30 \\ \alpha < \frac{470}{16}, \varepsilon < \frac{400}{16}; \\ R = \frac{L}{\tan\alpha} \\ r = 4.878 \end{cases}$$

5.2.6 求解结果

由以上模型求解得到的无人车轨迹为：

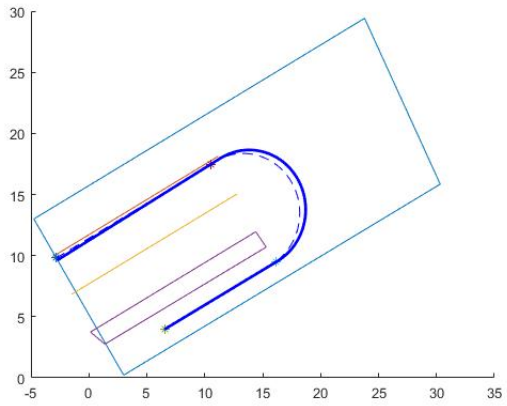

图 10: 无人车调头轨迹

5.3 静止障碍物型

如果道路上还存在其它静止障碍物时，分别对仅存在障碍物 F、仅存在障碍物 G、或二者都存在时这三种情况作分析。

5.3.1 仅存在 F 时

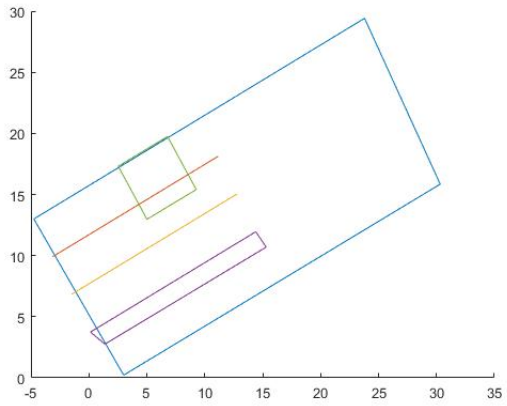

图 11: 无人车调头轨迹

5.3.1.1 无人车理想调头路线过程分析

从图 12 可以看出，障碍物 F 占据了整个 1 车道和半个 2 车道，因此对无人车的路线规划时，只考虑从 3 车道通过。

同样，对于无人车的路线依然分为三段，且第一段直行路线不变，而对第二段的调头和第三段的路线作改变。第二段调头部分再分为三个小部分，如图中的 ② ③ ④ 所示，第三段部分即为 ⑤ ，则根据 ④ 的终点来确定其具体路线。

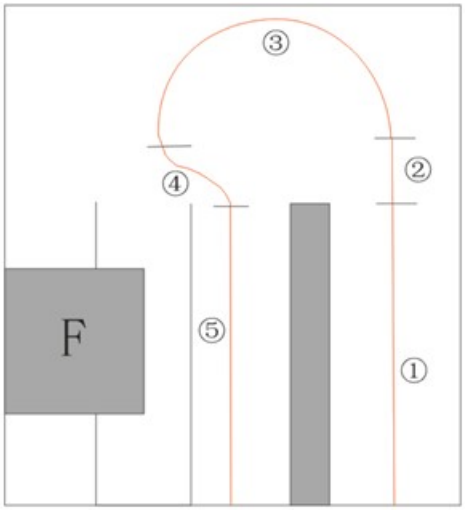

图 12: 无人车调头轨迹

5.3.1.2 路线的确定

路线 1： 与前面一致保持不变；

路线 2： 设置这条路线为直线，目的是拉长距离方便进行转弯调头，路线的长度则根据 ① 的长度和场地大小设定一个参数值，只要保证后续路线符合约束条件即可。

路线 3： 根据曲率求出允许的最小半径，以该半径长度作圆为路线，行驶至一定距离后进入路线 ④ 。

路线 4： 为了使无人车从调头处顺利进入车道，需要将方向盘回调，而这段回调的路线则是右转弯的，因此曲率取负数同样作圆即可。

路线 5： 当路线 ④ 行驶后，只要从其终点，平行于车道行驶至调头区域边界线，则完成整个调头路线。

5.3.1.3 关键坐标的计算

第一段的起点坐标仍然为 X_1

第一段和第二段的交点坐标为 $X_2\left(x_2, y_2\right)$

第二段与第三段的交点坐标 $Y_5\left(x_5, y_5\right)$

第二段中路线 (2) 与路线 (3)、路线 (3) 与路线 (4) 的交点坐标分别为 $Y_5'\left(x_{51}, y_{51}\right), Y_5''$，$\left(x_{52}, y_{52}\right)$

设路线 (2)、路线 (3)、路线 (4) 的长度分别为 L_1 L_2 L_3, 则有:

$$L_1 = \sqrt{\left(x_{51} - x_{21}\right)^2 + \left(y_{51} - y_{21}\right)^2} \tag{14}$$

$$L_2 = 2r \cdot \arcsin\left(\frac{\sqrt{\left(x_{52} - x_{21}\right)^2 + \left(y_{52} - y_{21}\right)^2}}{2r}\right) \tag{15}$$

$$L_3 = 2r \cdot \arcsin\left(\frac{\sqrt{\left(x_5 - x_{21}\right)^2 + \left(y_5 - y_{21}\right)^2}}{2r}\right) \tag{16}$$

过点 $Y_5\left(x_5, y_5\right)$ 作垂直 A 边界的线，即可求得调头的终点，设为 Z_5 。

5.3.1.4 驾驶轨迹

此时无人车的驾驶轨迹为：

$X_1 \rightarrow X_2 \rightarrow Y_5' \rightarrow Y_5'' \rightarrow Z_5$ ； 此时轨迹长度为 l_5

5.3.1.5 求解结果

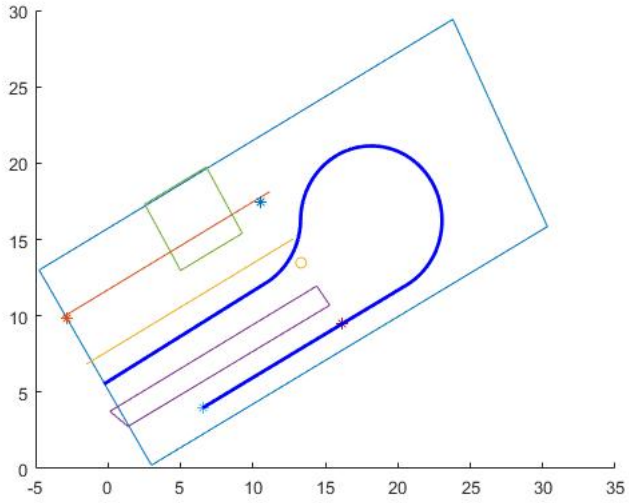

图 13: 仅存在 F 时无人车轨迹图

5.3.2 仅存在障碍物 G

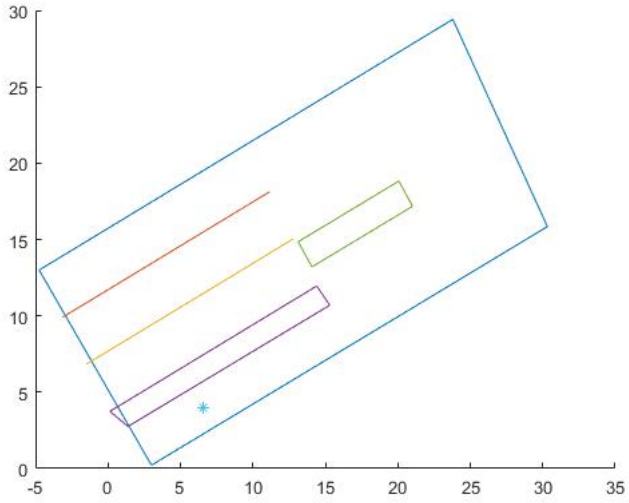

图 14: 仅存在 G 时无人车轨迹图

5.3.2.1 无人车理想调头路线过程分析

从图 15 可以看出，障碍物 G 位于 3 车道的入口处，且几乎将车道完全阻挡，因此考虑无人车从 1 车道和 2 车道通过。对于无人车的路线依然分为三段，第一段直行路线不变，而对第二段的调头和第三段的路线作改变。由于无人车可以从 1 车道或者 2 车道通过，所以当通过第一段后，设定两条路线分别从两车道行驶，即图中的路线 ① 和路线 ② 。

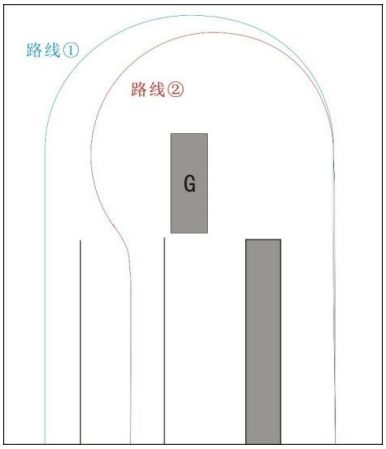

图 15: 仅存在 G 时的路线示意图

5.3.2.2 路线的确定

第 1 段： 与前面一致保持不变；

第 2 段： 由于障碍物的存在，路线 和路线 都需要绕开其而做一个掉头。在路线 ① 中，为了顺利地进入 1 车道，掉头所需的半径则应该增大，由于最低的曲率半径为 4.878m，通过逐步增大半径进行路线测试，最终设定其掉头半径为 5.5m。在路线 ② 中，行驶路线与上一问中仅存在 G 时的路线相似，不同之处就是在于更改了路线长度的参数值。

第 3 段： 当第二段路线行驶至车道中心附近，在保证不压车道线的情况下，平行于车道线行驶至掉头区域边界线即可。

5.3.2.3 关键坐标的计算

第一段的起点坐标仍然为 X_1

第一段和第二段的交点坐标为 $X_2\,(x_2, y_2)$

路线 ① 中:

第二段与第三段的交点坐标 $Y_6\,(x_{61}, y_{61})$

第三段与边界线的交点坐标为 $Z_{61}\,(x_{62}, y_{62})$

路线 ② 中:

与上一问中仅存在 F 时类似，只是交点坐标不同;

第二段中有交点坐标分别为 $Y_6'\,(x_{63}, y_{63})$ $Y_6''\,(x_{64}, y_{64})$

设第二段中三小段的长度分别为 L_1', L_2', L_3', 则有：

$$L_1' = \sqrt{(x_{63} - x_{21})^2 + (y_{63} - y_{21})^2} \tag{17}$$

$$L_2' = 2r \cdot \arcsin\left(\frac{\sqrt{(x_{64} - x_{21})^2 + (y_{64} - y_{21})^2}}{2r}\right) \tag{18}$$

$$L_3' = 2r \cdot \arcsin\left(\frac{\sqrt{(x_{62} - x_{21})^2 + (y_{62} - y_{21})^2}}{2r}\right) \tag{19}$$

5.3.2.4 驾驶轨迹

此时无人车的驾驶轨迹为：

路线 ①：$X_1 \to X_2 \to Y_6 \to Z_{61}$; 此时轨迹长度为 l_6

路线 ②：$X_1 \to X_2 \to Y_6' \to Y_6'' \to Y_{62} \to Z_{62}$ 此时轨迹长度为 l_6'

5.3.2.5 求解结果

路线 ① ：

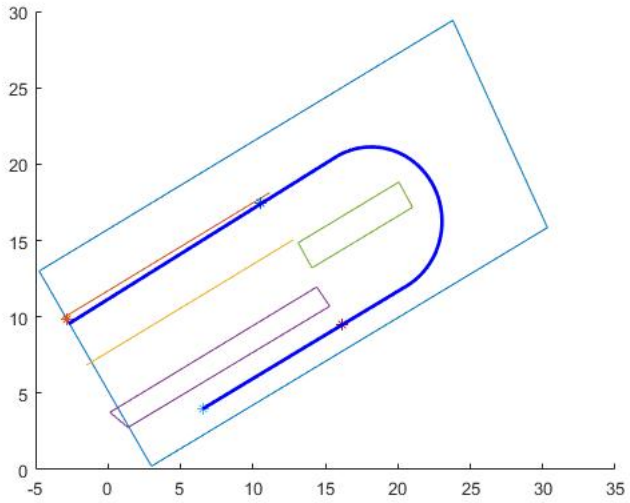

图 16: 仅存在 G 时无人车路线①轨迹图

路线 ② ：

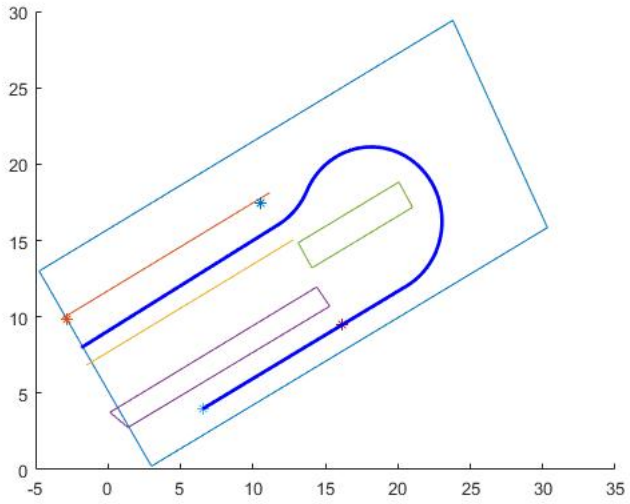

图 17: 仅存在 G 时无人车路线②轨迹图

5.3.3 F 和 G 都存在时

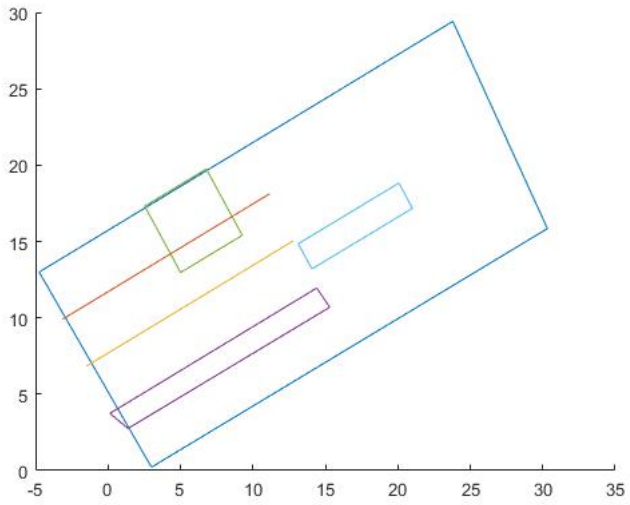

图 18: F 和 G 都存在时的路况示意图

5.3.3.1 无人车理想调头路线过程分析

由于障碍物 F 完全占据 1 车道，障碍物 G 占据 3 车道入口，无人车的路线限制较大，因此设定无人车拐弯后，从 2 车道入口进入，再转向 3 车道直行。

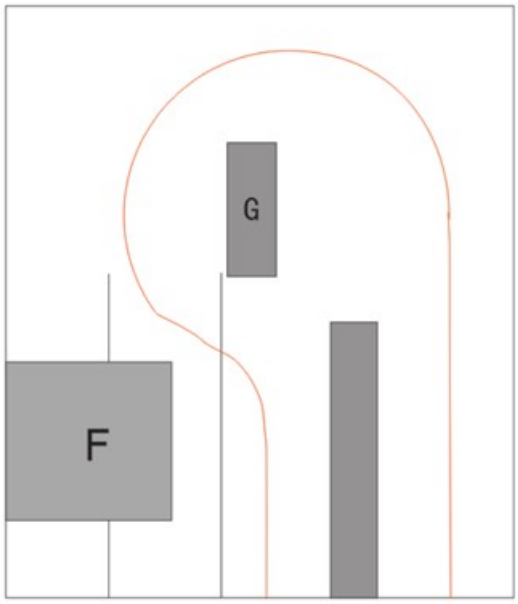

图 19: F 和 G 都存在时的路线示意图

5.3.3.2 路线的确定

从图 19 的路线示意图可看出，路线情况同样与仅存在 F 时相似。

第 1 段： 与前面一致保持不变；

第 2 段： 由于障碍物 G 的存在，需绕过其而做一个转弯，该路线掉头中的左转弯部分与仅存在 G 时的路线 一致，当准备进入车道时，需要将方向盘打回，此时与仅存在 F 时的路线一致，不同之处在于打回的路线中应要注意障碍物 F，为了保持安全距离，所转的弯会比较急。

第 3 段： 当第二段路线行驶至 3 车道的中心线附近，在保证不压车道线的情况下，平行于车道线行驶至掉头区域边界线即可。

5.3.3.3 关键坐标的计算

第一段的起点坐标仍然为 X_1

第一段和第二段的交点坐标为 $X_2\left(x_2, y_2\right)$

与上一问中仅存在 F 时类似，只是交点坐标不同;

第二段中有交点坐标分别为 $Y_7'\left(x_{71}, y_{71}\right), Y_7''\left(x_{72}, y_{72}\right)$

设第二段中三小段的长度分别为 $L_1'', L_2''\ L_3''$, 则有:

$$L_1'' = \sqrt{\left(x_{71} - x_2\right)^2 + \left(y_{71} - y_2\right)^2} \tag{20}$$

$$L_2'' = 2r \cdot \arcsin\left(\frac{\sqrt{\left(x_{72} - x_{71}\right)^2 + \left(y_{72} - y_{71}\right)^2}}{2r}\right) \tag{21}$$

$$L_3'' = 2r \cdot \arcsin\left(\frac{\sqrt{\left(x_7 - x_{72}\right)^2 + \left(y_7 - y_{72}\right)^2}}{2r}\right) \tag{22}$$

过点 $Y_7\left(x_7, y_7\right)$ 作垂直 A 边界的线, 即可求得掉头的终点, 设为 $Z_7\left(x_{73}, y_{73}\right)$ 。

5.3.3.4 驾驶轨迹

此时无人车的驾驶轨迹为: $X_1 \rightarrow X_2 \rightarrow Y_7' \rightarrow Y_7'' \rightarrow Y_7 \rightarrow Z_7$; 此时轨迹长度为 l_7

5.3.3.5 求解结果

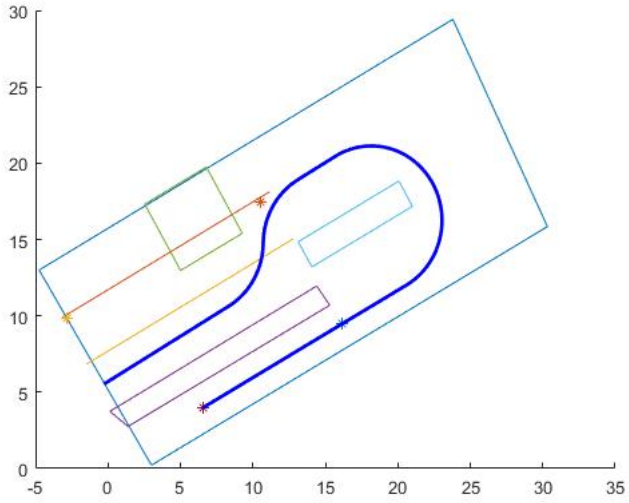

图 20: F 和 G 都存在的无人车轨迹图

5.4 斑马线型

5.4.1 只存在斑马线时

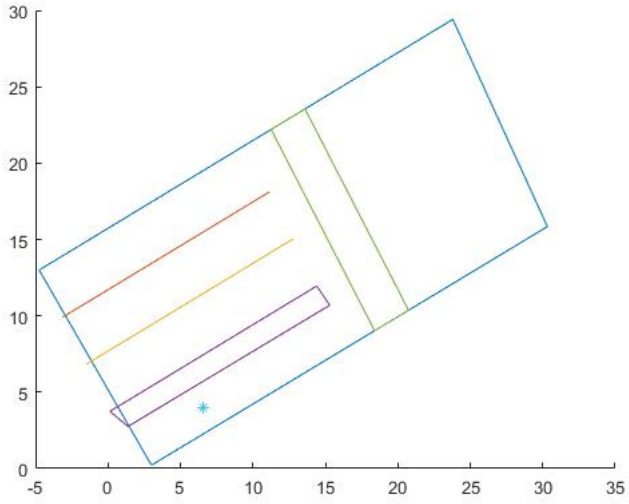

图 21: 存在斑马线时路况示意图

5.4.1.1 无人车调头路线分析

题目需考虑到两种情况：不压斑马线和越过斑马线，分别在这两种情况下判断是否能完成掉头。设路线 ① 为不压斑马线，即从斑马线下方行驶；设路线 ② 为越过斑马线，即从斑马线上方行驶；

5.4.1.2 路线的确定

路线 ① ：

将该路线分为 5 段，为了第 3 段的掉头能更加顺利，在启程时需要向右行驶一段距离，并保证车头在掉头前是朝向右边的。

第 1 段： 无人车向右行驶一段距离。

第 2 段： 为了与区域 A 保持安全距离，需将路线往回调，但车朝向仍向右。

第 3 段： 保持直线向前行驶，到达掉头路口时即可准备掉头。

第 4 段： 以最小曲率半径 4.878 作半圆为掉头的行驶路线。

第 5 段： 平行于车道向前行驶至掉头区域边界线即可。

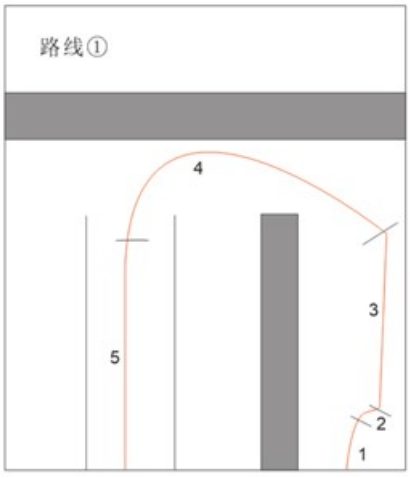

图 22: 不压斑马线无人车路线图

路线 ② ：

将该路线分为 3 段，与问题 1 的路线相似。

第 1 段： 从出发点向前直行一定的距离。

第 2 段： 以最小曲率直径作半圆为行驶路线，判断是否能越过斑马线，若不能越过则增加半径长度。

第 3 段： 行驶至车道入口时，平行于车道向前行驶至掉头区域边界线即可。

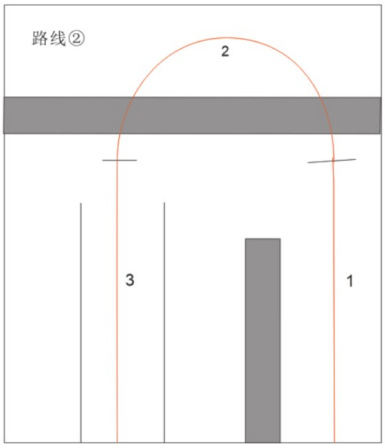

图 23: 越过斑马线无人车行驶路线图

5.4.1.3 求解结果

路线 ① ——不压斑马线时：

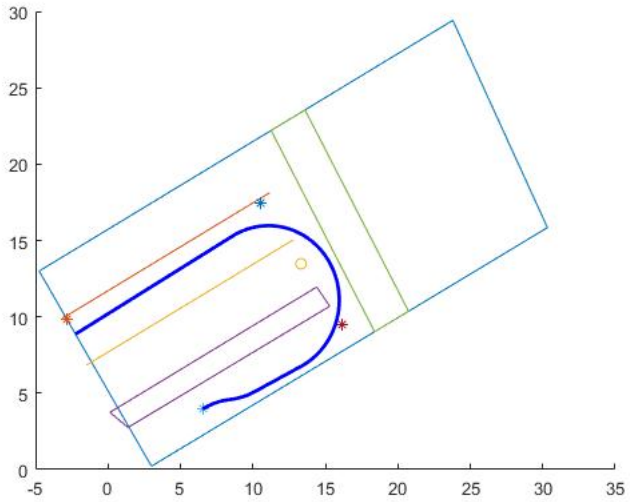

图 24: 无人车不压斑马线时轨迹图

路线 ② ——压斑马线时：

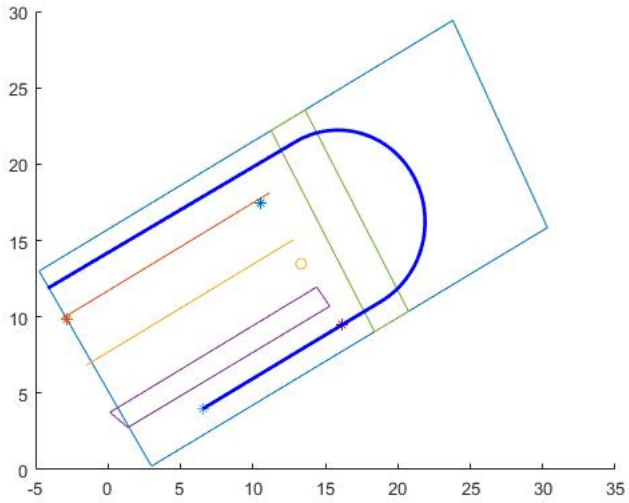

图 25: 无人车越过斑马线时轨迹图

5.4.2 障碍物和斑马线同时存在时

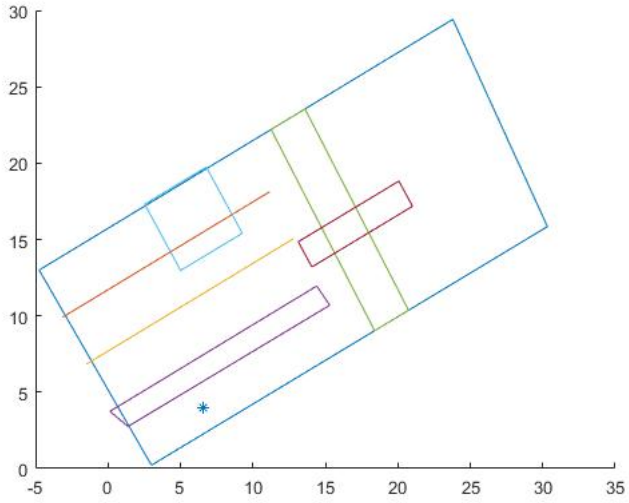

图 26: 斑马线和障碍物都存在时路况示意图

5.4.2.1 无人车调头路线分析

同样也需考虑到两种情况：不压斑马线和越过斑马线，分别在这两种情况下判断是否能完成掉头。设路线 ① 为不压斑马线，即从斑马线下方行驶；设路线 ② 为越过斑马线，即从斑马线上方行驶；

5.4.2.2 路线的确定

路线 ① ：由于斑马线和障碍物的限制，若要从斑马线下过，则只能从障碍物 G 与隔离障碍物 D 间的路口通过。

第 1 段： 无人车向前行驶一段距离。

第 2 段： 到达掉头区域时，以最小曲率半径 4.878 作半圆为掉头的行驶路线。

第 3 段： 平行于车道向前行驶至掉头区域边界线即可。

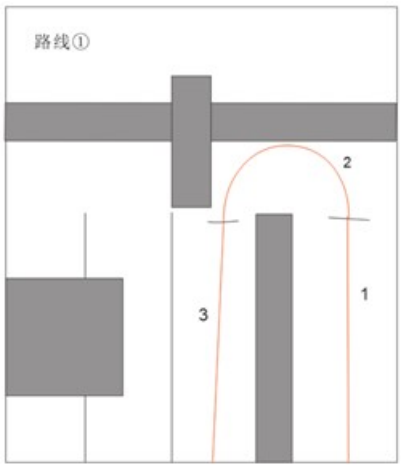

图 27: 不压斑马线无人车路线图

路线 ② ：从图 27 的路线示意图可看出，路线情况与障碍物 F 与 G 同时存在时相似，不同之处在于多了一条斑马线，因此掉头的路线有所改变。

第 1 段： 无人车直行一段距离。

第 2 段： 以最小曲率半径 4.8787m 作圆为路线行驶越过斑马线。

第 3 段： 越过斑马线后进入 2 车道，同时往 3 车道的方向行驶。

第 4 段： 将无人车方向回调，以同样的半径作圆为路线。

第 5 段： 平行于车道向前行驶至掉头区域边界线即可。

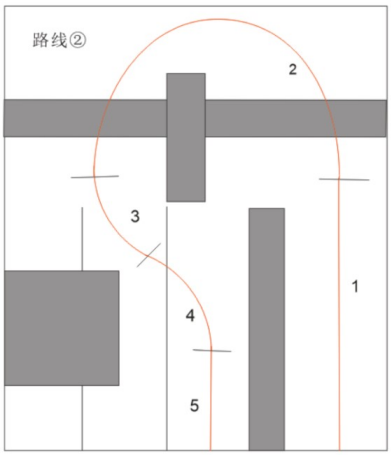

图 28: 越过斑马线无人车行驶路线图

5.4.2.3 求解结果

路线 ① ——不压斑马线时：

通过 matlab 运行出的结果可以看出，这条路线无法顺利通过。

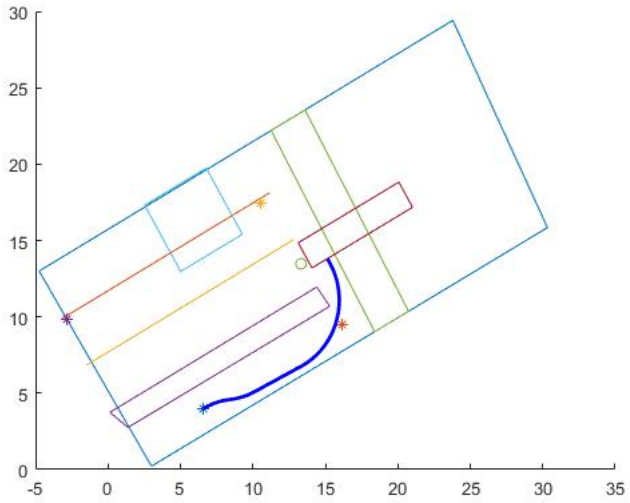

图 29: 无人车不压斑马线时轨迹图

路线 ② ——越过斑马线时：

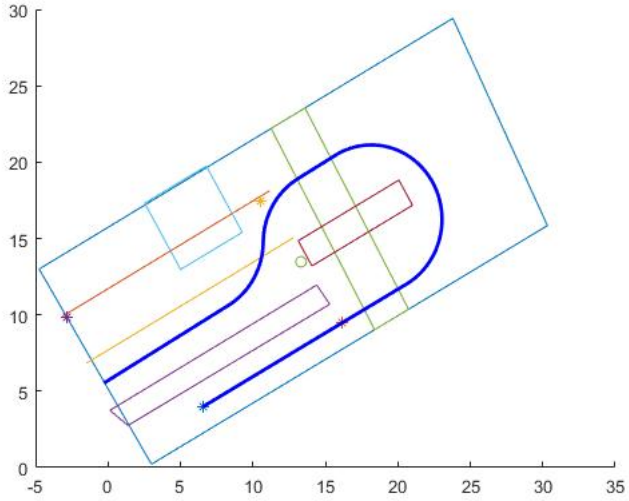

图 30: 无人车越过斑马线时轨迹图

5.5 动态规划型

针对问题 5 和问题 6，采用动态路径规划避障的算法来求解。[3]

5.5.1 动态路径规划技术

动态路径规划技术主要分为两大类: 第一类是全局路径规划，它是根据先验环境模型找出从起点到终点中符合条件的最优或次优路径，涉及的根本问题是环境模型的表达和路径搜寻策略；第二类是局部路径规划，是指在未知或部分未知的环境下通过传感器获取周围环境信息，并使自动驾驶汽车自主获得一条无碰撞最优规划的路径，它侧重于考虑车辆当前局部环境信息。[4]

5.5.2 动态路径规划避障算法

本题提出的动态路径规划方法是在已知车辆初始状态信息 (包括车辆的位置、车头方向等) 的基础上，产生一条安全又舒适的行驶路线。在已知全局路线的情况下进行局部规划，全局路线通过车道级的高精度导航系统获取。全局路线可以由一组道路边缘的有序点组成。如图 31 所示，所用方法包括三个部分：基准线生成、候选路径生成和最优路径选择。

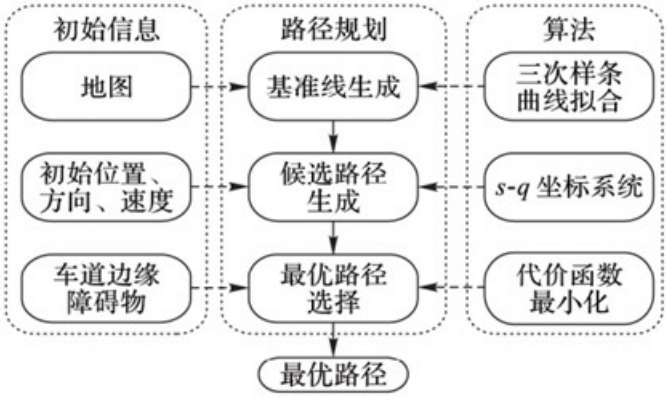

图 31: 动态路径规划算法流程

6 模型评价

6.1 路径规划与运动规划

自动驾驶车辆中的调头问题需要给出车辆的轨迹路线，同时给出车辆沿该路径运动的速度信息，并且使车辆在整个运动过程中满足动力学约束及其它来自于外部环境和内部系统等约束条件。[5]

本文采用的是路径规划，与运动规划相比，路径规划仅仅只需生成满足空间约束条件的几何曲线，而如果采用运动规划，则会有以下难点：

（1）运动规划的过程复杂；

（2）通过运动规划生成的轨迹难以在后继控制模块中被有效跟踪；

（3）决策规划模块仅需生成路径而非轨迹，车辆实际运动速度暂时无法或无需决策。[6]

6.2 模型优缺点

本文解决问题的方法为插值拟合法，插值拟合法具有以下优点：

（1）求解路径的速度快；

（2）能够满足路径平滑性，可行性；

（3）方法简单。

但是插值拟合法由于采用特定类型的曲线函数描述行车路径，因此这一类方法无法充分发挥车辆的全部运动能录，因为车辆的动力学方程并不等价于插值拟合法中所使用的曲线函数。[7]。

6.3 模型优化

对于路径规划还可以采用其它方法如采样法、机器学习方法、最优控制方法等，其中最优控制方法能够有效地改进上述问题的不足，最优控制方法是采用建立微分方程组来描述车辆运动，同时在运动学微分方程的基础上补充必要的约束条件以及车辆行驶的性能指标式，这一方法将车辆运动规划问题描述为最优问题，具有客观、直接、统一、完备的优点。[8]

参考文献

[1] 祁志远 and 肖仕武. 三种直角坐标牛顿潮流算法的收敛性比较. *华北电力大学学报 (自然科学版)*, 5, 2015.

[2] 余卓平, 李奕姗, and 熊璐. 无人车运动规划算法综述. *同济大学学报 (自然科学版)*, 45(8):1150–1159, 2017.

[3] 周慧子, 胡学敏, 陈龙, 田梅, and 熊豆. 面向自动驾驶的动态路径规划避障算法. *计算机应用*, 37(3):883–888, 2017.

[4] 王富奎. 高动态环境下智能车局部路径规划研究. Master's thesis, 电子科技大学, 2018.

[5] Bai Li, Zhijiang Shao, Youmin Zhang, and Pu Li. Nonlinear programming for multi-vehicle motion planning with homotopy initialization strategies. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 118–123. IEEE, 2017.

[6] 李柏, 张友民, and 邵之江. 自动驾驶车辆运动规划方法综述. *收藏*, 6, 2018.

[7] Bai Li, Youmin Zhang, Yiheng Feng, Yue Zhang, Yuming Ge, and Zhijiang Shao. Balancing computation speed and quality: A decentralized motion planning method for cooperative lane changes of connected and automated vehicles. *IEEE Transactions on Intelligent Vehicles*, 3(3):340–350, 2018.

[8] 姚君延. 基于深度增强学习的路径规划算法研究. Master's thesis, 电子科技大学, 2018.

附录

A 问题一

main.m

```
clc, clear

dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
throttle_dec_max = -5;
wheel_angle = 0;
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
    t];
```

```
A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
    18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
    9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
    19.744528];
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
    20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
    18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')
```

```
plot(x0, y0, '*')

acc = 2;



[Int_A_B_x, Int_A_B_y] = Int_Point([B_x(1), B_y(1)], [B_x
    (2), B_y(2)], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);



[Int_A_C_x, Int_A_C_y] = Int_Point([C_x(1), C_y(1)], [C_x
    (2), C_y(2)], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);



[Int_A_D_x, Int_A_D_y] = Int_Point([D_x(3), D_y(3)], [D_x
    (4), D_y(4)], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);



turn_end1 = [(A_x(4) + Int_A_B_x) / 2, (A_y(4) + Int_A_B_y)
     / 2];
turn_end2 = [(Int_A_C_x + Int_A_B_x) / 2, (Int_A_C_y +
    Int_A_B_y) / 2];
turn_end3 = [(Int_A_C_x + Int_A_D_x) / 2, (Int_A_C_y +
    Int_A_D_y) / 2];
plot(turn_end1(1), turn_end1(2), '*');
plot(turn_end2(1), turn_end2(2), '*');
plot(turn_end3(1), turn_end3(2), '*');
```

```
[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');

delta_x = D_x(3) - Int_A_D_x;
delta_y = D_y(3) - Int_A_D_y;
Int_2 = [turn_end1(1) + delta_x, turn_end1(2) + delta_y];
Int_3 = [turn_end2(1) + delta_x, turn_end2(2) + delta_y];
Int_4 = [turn_end3(1) + delta_x, turn_end3(2) + delta_y];
plot(Int_2(1), Int_2(2), '*');
plot(Int_3(1), Int_3(2), '*');
plot(Int_4(1), Int_4(2), '*');

distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_3; turn_end2], 'euclidean');

line([x0, Int_1(1)], [y0, Int_1(2)], 'linestyle', '--', '
    color', 'b')
line([turn_end1(1), Int_2(1)], [turn_end1(2), Int_2(2)], '
    linestyle', '--', 'color', 'b')
line([turn_end2(1), Int_3(1)], [turn_end2(2), Int_3(2)], '
    linestyle', '--', 'color', 'b')
line([turn_end3(1), Int_4(1)], [turn_end3(2), Int_4(2)], '
    linestyle', '--', 'color', 'b')

curR1 = sqrt((Int_2(1) - Int_1(1))^2 + (Int_2(2) - Int_1(2)
```

```
    )^2) / 2;
curR2 = sqrt((Int_3(1) - Int_1(1))^2 + (Int_3(2) - Int_1(2)
    )^2) / 2;
curR3 = sqrt((Int_4(1) - Int_1(1))^2 + (Int_4(2) - Int_1(2)
    )^2) / 2;

circle_center1 = [(Int_2(1) + Int_1(1)) / 2, (Int_2(2) +
    Int_1(2)) / 2];
circle_center2 = [(Int_3(1) + Int_1(1)) / 2, (Int_3(2) +
    Int_1(2)) / 2];
circle_center3 = [(Int_4(1) + Int_1(1)) / 2, (Int_4(2) +
    Int_1(2)) / 2];




distance2_1 = Arc_Length(Int_2, Int_1, circle_center1);
distance2_2 = Arc_Length(Int_3, Int_1, circle_center2);
distance2_3 = Arc_Length(Int_4, Int_1, circle_center3);

drawArc(Int_2, Int_1, circle_center1)
drawArc(Int_3, Int_1, circle_center2)
drawArc(Int_4, Int_1, circle_center3)



i = 1;
distance = 0;

while i == 1
```

```
    if distance <= distance1
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + distance2_1
        curR = curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + distance2_1 + distance3
      + 5
        curR = 0;
        speed = 2;
        acc = 0;
    else
        break
    end

    [NextTrackPointData, distance] = Next_TrackPoint(
      ALLTrackPointData(end, :), speed, acc, dt, curR,
      distance);

    ALLTrackPointData = [ALLTrackPointData;
      NextTrackPointData];

    i = inpolygon(NextTrackPointData(1), NextTrackPointData
      (2), A_x, A_y);
end

plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
  linestyle', '-', 'linewidth', 2, 'color', 'b');
```

B 问题二

```
clc, clear

dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
throttle_dec_max = -5;
wheel_angle = 0;
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
    t];

A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
```

```
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
    18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
    9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
    19.744528];
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
    20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
    18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')



plot(x0, y0, '*')
```

```
acc = 2;


[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');


k = (D_y(3) - D_y(2)) / (D_x(3) - D_x(2));
Int_2(1) = Int_1(1) - sqrt((4.878 * 2)^2 / (1 + k^2));
Int_2(2) = Int_1(2) + sqrt((4.878 * 2)^2 / (1 + 1 / k^2));
plot(Int_2(1), Int_2(2), '*')


[Int_3(1), Int_3(2)] = Int_Point([Int_2(1), Int_2(2)], [
    Int_2(1) + 20, Int_2(2) + 20 * ((D_y(4) - D_y(3)) / (D_x
    (4) - D_x(3)))], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);
plot(Int_3(1), Int_3(2), '*')

distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_2; Int_3], 'euclidean');


line([x0, Int_1(1)], [y0, Int_1(2)], 'linestyle', '--', '
    color', 'b')
line([Int_2(1), Int_3(1)], [Int_2(2), Int_3(2)], 'linestyle
    ', '--', 'color', 'b')
```

```
curR1 = 4.878;

circle_center = [(Int_1(1) + Int_2(1)) / 2, (Int_1(2) +
   Int_2(2)) / 2];


distance2 = Arc_Length(Int_2, Int_1, circle_center);

drawArc(Int_2, Int_1, circle_center)



i = 1;
distance = 0;

while i == 1

    if distance <= distance1
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + distance2
        curR = curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + distance2 + distance3 +
       5
        curR = 0;
        speed = 2;
        acc = 0;
```

```
        else
            break
        end

        [NextTrackPointData, distance] = Next_TrackPoint(
            ALLTrackPointData(end, :), speed, acc, dt, curR,
            distance);

        ALLTrackPointData = [ALLTrackPointData;
            NextTrackPointData];

        i = inpolygon(NextTrackPointData(1), NextTrackPointData
            (2), A_x, A_y);
    end

    plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
        linestyle', '-', 'linewidth', 2, 'color', 'b');
```

C 问题二

```
clc, clear

dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
```

```
throttle_dec_max = -5;
wheel_angle = 0;
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
    t];

A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
    18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
    9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
```

```
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
    19.744528];
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
    20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
    18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')

plot(F_x,F_y,'-')

plot(x0, y0, '*')

acc = 2;


[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');


k = (D_y(3) - D_y(2)) / (D_x(3) - D_x(2));
Int_2(1) = Int_1(1) - sqrt((4.878 * 2)^2 / (1 + k^2));
Int_2(2) = Int_1(2) + sqrt((4.878 * 2)^2 / (1 + 1 / k^2));
```

```
plot(Int_2(1), Int_2(2), '*')


[Int_3(1), Int_3(2)] = Int_Point([Int_2(1), Int_2(2)], [
    Int_2(1) + 20, Int_2(2) + 20 * ((D_y(4) - D_y(3)) / (D_x
    (4) - D_x(3)))], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);
plot(Int_3(1), Int_3(2), '*')

distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_2; Int_3], 'euclidean');




curR1 = 4.878;

circle_center = [(Int_1(1) + Int_2(1)) / 2, (Int_1(2) +
    Int_2(2)) / 2];
plot(circle_center(1), circle_center(2), 'o')

distance2 = Arc_Length(Int_2, Int_1, circle_center);



i = 1;
distance = 0;

while i == 1
```

```
if distance <= distance1 + 5
    curR = 0;
    speed = 2;
    acc = 0;
elseif distance <= distance1 + 5 + distance2 + 5
    curR = curR1;
    speed = 2;
    acc = 0;
elseif distance <= distance1 + 5 + distance2 + 10
    curR = -curR1;
    speed = 2;
    acc = 0;
elseif distance <= distance1 + 5 + distance2 + 10 +
   distance3
    curR = 0;
    speed = 2;
    acc = 0;
else
    break
end

[NextTrackPointData, distance] = Next_TrackPoint(
   ALLTrackPointData(end, :), speed, acc, dt, curR,
   distance);

ALLTrackPointData = [ALLTrackPointData;
   NextTrackPointData];

i = inpolygon(NextTrackPointData(1), NextTrackPointData
   (2), A_x, A_y);
```

```
end

plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
    linestyle', '-', 'linewidth', 2, 'color', 'b');

clc, clear

dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
throttle_dec_max = -5;
wheel_angle = 0;
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
    t];
```

```
A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
    18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
    9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
    19.744528];
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
    20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
    18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')

plot(F_x,F_y,'-')
```

```
plot(G_x,G_y,'-')
plot(x0, y0, '*')

acc = 2;

[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');


k = (D_y(3) - D_y(2)) / (D_x(3) - D_x(2));
Int_2(1) = Int_1(1) - sqrt((4.878 * 2)^2 / (1 + k^2));
Int_2(2) = Int_1(2) + sqrt((4.878 * 2)^2 / (1 + 1 / k^2));
plot(Int_2(1), Int_2(2), '*')


[Int_3(1), Int_3(2)] = Int_Point([Int_2(1), Int_2(2)], [
    Int_2(1) + 20, Int_2(2) + 20 * ((D_y(4) - D_y(3)) / (D_x
    (4) - D_x(3)))], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);
plot(Int_3(1), Int_3(2), '*')


distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_2; Int_3], 'euclidean');
```

```
curR1 = 4.878;

circle_center = [(Int_1(1) + Int_2(1)) / 2, (Int_1(2) +
   Int_2(2)) / 2];



distance2 = Arc_Length(Int_2, Int_1, circle_center);



i = 1;
distance = 0;

while i == 1;

    if distance <= distance1 + 5
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2
        curR = curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2 + 3
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2 + 8
        curR = curR1;
```

```
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2 + 13
        curR = -curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2 + 10 +
       distance3
        curR = 0;
        speed = 2;
        acc = 0;
    else
        break
    end

    [NextTrackPointData, distance] = Next_TrackPoint(
       ALLTrackPointData(end, :), speed, acc, dt, curR,
       distance);

    ALLTrackPointData = [ALLTrackPointData;
       NextTrackPointData];

    i = inpolygon(NextTrackPointData(1), NextTrackPointData
       (2), A_x, A_y);
end

plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
   linestyle', '-', 'linewidth', 2, 'color', 'b');

clc, clear
```

```
dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
throttle_dec_max = -5;
wheel_angle = 0;
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
    t];

A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
```

```
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
   18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
   9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
   19.744528];
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
   20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
   18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')



plot(G_x,G_y,'-')
plot(x0, y0, '*')

acc = 2;
```

```
[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');


k = (D_y(3) - D_y(2)) / (D_x(3) - D_x(2));
Int_2(1) = Int_1(1) - sqrt((4.878 * 2)^2 / (1 + k^2));
Int_2(2) = Int_1(2) + sqrt((4.878 * 2)^2 / (1 + 1 / k^2));
plot(Int_2(1), Int_2(2), '*')


[Int_3(1), Int_3(2)] = Int_Point([Int_2(1), Int_2(2)], [
    Int_2(1) + 20, Int_2(2) + 20 * ((D_y(4) - D_y(3)) / (D_x
    (4) - D_x(3)))], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);
plot(Int_3(1), Int_3(2), '*')

distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_2; Int_3], 'euclidean');


line([x0, Int_1(1)], [y0, Int_1(2)], 'linestyle', '--', '
    color', 'b')
line([Int_2(1), Int_3(1)], [Int_2(2), Int_3(2)], 'linestyle
    ', '--', 'color', 'b')

curR1 = 4.878;

circle_center = [(Int_1(1) + Int_2(1)) / 2, (Int_1(2) +
    Int_2(2)) / 2];
```

```
distance2 = Arc_Length(Int_2, Int_1, circle_center);

drawArc(Int_2, Int_1, circle_center)

i = 1;
distance = 0;

while i == 1;

    if distance <= distance1 + 5
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 8 + distance2
        curR = curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 8 +distance2 + 3
        curR = -curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 8 + distance2 +
       distance3 + 10
        curR = 0;
        speed = 2;
        acc = 0;
    else
        break
```

```
    end

    [NextTrackPointData, distance] = Next_TrackPoint(
        ALLTrackPointData(end, :), speed, acc, dt, curR,
        distance);

    ALLTrackPointData = [ALLTrackPointData;
        NextTrackPointData];

    i = inpolygon(NextTrackPointData(1), NextTrackPointData
        (2), A_x, A_y);
end

plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
    linestyle', '-', 'linewidth', 2, 'color', 'b');
```

D 问题四

```
clc, clear

dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
throttle_dec_max = -5;
wheel_angle = 0;
```

```
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
    t];

A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
    18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
    9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
    19.744528];
```

```
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
    20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
    18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')
plot(E_x,E_y,'-')


plot(x0, y0, '*')

acc = 2;


[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');


k = (D_y(3) - D_y(2)) / (D_x(3) - D_x(2));
Int_2(1) = Int_1(1) - sqrt((4.878 * 2)^2 / (1 + k^2));
Int_2(2) = Int_1(2) + sqrt((4.878 * 2)^2 / (1 + 1 / k^2));
plot(Int_2(1), Int_2(2), '*')
```

```
[Int_3(1), Int_3(2)] = Int_Point([Int_2(1), Int_2(2)], [
    Int_2(1) + 20, Int_2(2) + 20 * ((D_y(4) - D_y(3)) / (D_x
    (4) - D_x(3)))], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);
plot(Int_3(1), Int_3(2), '*')

distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_2; Int_3], 'euclidean');




curR1 = 4.878;

circle_center = [(Int_1(1) + Int_2(1)) / 2, (Int_1(2) +
    Int_2(2)) / 2];
plot(circle_center(1), circle_center(2), 'o')

distance2 = Arc_Length(Int_2, Int_1, circle_center);




i = 1;
distance = 0;

while i == 1

    if distance <= 1.8
        curR = -curR1;
```

```
        speed = 2;
        acc = 0;
    elseif distance <= 3.7
        curR = curR1;
        speed = 2;
        acc = 0;
    elseif distance <= 7.35
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= 22.8
        curR = curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 - 3.5 + distance2 +
       distance3 + 5
        curR = 0;
        speed = 2;
        acc = 0;
    else
        break
    end

    [NextTrackPointData, distance] = Next_TrackPoint(
       ALLTrackPointData(end, :), speed, acc, dt, curR,
       distance);

    ALLTrackPointData = [ALLTrackPointData;
       NextTrackPointData];
```

```
    i = inpolygon(NextTrackPointData(1), NextTrackPointData
        (2), A_x, A_y);
end

plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
   linestyle', '-', 'linewidth', 2, 'color', 'b');

clc, clear

dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
throttle_dec_max = -5;
wheel_angle = 0;
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
```

```
    t];

A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
    18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
    9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
    19.744528];
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
    20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
    18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')
```

```
plot(E_x, E_y, '-')


plot(x0, y0, '*')

acc = 2;


[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');


k = (D_y(3) - D_y(2)) / (D_x(3) - D_x(2));
Int_2(1) = Int_1(1) - sqrt((4.878 * 2)^2 / (1 + k^2));
Int_2(2) = Int_1(2) + sqrt((4.878 * 2)^2 / (1 + 1 / k^2));
plot(Int_2(1), Int_2(2), '*')


[Int_3(1), Int_3(2)] = Int_Point([Int_2(1), Int_2(2)], [
    Int_2(1) + 20, Int_2(2) + 20 * ((D_y(4) - D_y(3)) / (D_x
    (4) - D_x(3)))], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);
plot(Int_3(1), Int_3(2), '*')

distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_2; Int_3], 'euclidean');
```

```
curR1 = 4.878;

circle_center = [(Int_1(1) + Int_2(1)) / 2, (Int_1(2) +
   Int_2(2)) / 2];
plot(circle_center(1), circle_center(2), 'o')

distance2 = Arc_Length(Int_2, Int_1, circle_center);



i = 1;
distance = 0;

while i == 1

    if distance <= distance1 + 3
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 6.5 + distance2
        curR = 6;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 6.5 + distance2 +
       distance3 + 5
        curR = 0;
        speed = 2;
        acc = 0;
```

```
    else
        break
    end

    [NextTrackPointData, distance] = Next_TrackPoint(
        ALLTrackPointData(end, :), speed, acc, dt, curR,
        distance);

    ALLTrackPointData = [ALLTrackPointData;
        NextTrackPointData];

    i = inpolygon(NextTrackPointData(1), NextTrackPointData
        (2), A_x, A_y);
end

plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
    linestyle', '-', 'linewidth', 2, 'color', 'b');

clc, clear

dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
throttle_dec_max = -5;
wheel_angle = 0;
```

```
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
    t];

A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
    18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
    9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
    19.744528];
```

```
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
    20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
    18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')
plot(E_x,E_y,'-')
plot(F_x,F_y,'-')
plot(G_x,G_y,'-')
plot(x0, y0, '*')

acc = 2;


[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');


k = (D_y(3) - D_y(2)) / (D_x(3) - D_x(2));
Int_2(1) = Int_1(1) - sqrt((4.878 * 2)^2 / (1 + k^2));
Int_2(2) = Int_1(2) + sqrt((4.878 * 2)^2 / (1 + 1 / k^2));
plot(Int_2(1), Int_2(2), '*')
```

```
[Int_3(1), Int_3(2)] = Int_Point([Int_2(1), Int_2(2)], [
    Int_2(1) + 20, Int_2(2) + 20 * ((D_y(4) - D_y(3)) / (D_x
    (4) - D_x(3)))], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);
plot(Int_3(1), Int_3(2), '*')

distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_2; Int_3], 'euclidean');




curR1 = 4.878;

circle_center = [(Int_1(1) + Int_2(1)) / 2, (Int_1(2) +
    Int_2(2)) / 2];
plot(circle_center(1), circle_center(2), 'o')

distance2 = Arc_Length(Int_2, Int_1, circle_center);




i = 1;
distance = 0;

while i == 1

    if distance <= 1.8
        curR = -curR1;
```

```
        speed = 2;
        acc = 0;
    elseif distance <= 3.7
        curR = curR1;
        speed = 2;
        acc = 0;
    elseif distance <= 7.35
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= 15.4
        curR = curR1;
        speed = 2;
        acc = 0;

    else
        break
    end

    [NextTrackPointData, distance] = Next_TrackPoint(
        ALLTrackPointData(end, :), speed, acc, dt, curR,
        distance);

    ALLTrackPointData = [ALLTrackPointData;
        NextTrackPointData];

    i = inpolygon(NextTrackPointData(1), NextTrackPointData
        (2), A_x, A_y);
end
```

```
plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
    linestyle', '-', 'linewidth', 2, 'color', 'b');

clc, clear

dt = 0.1;
car_length = 5;
car_width = 2;
car_wheel_base = 2.8;
safedistance = 0.3;
wheel_angle_max = 470/16;
wheel_speed_max = 400/16;
throttle_acc_max = 3;
throttle_dec_max = -5;
wheel_angle = 0;
x0 = 6.53023;
y0 = 3.970426;
theta = 0.5194235;
cur = 0;
cur_max = 0.205;
curR_min = 1 / cur_max;
dcur = 0;
speed = 0;
acc = 0;
t = 0;
ALLTrackPointData = [x0, y0, theta, cur, dcur, speed, acc,
    t];

A_x = [2.998806; 30.323206; 23.792806; -4.748594;
    2.998806];
```

```
A_y = [0.220528; 15.844528; 29.414528; 12.996528;
    0.220528];
B_x = [11.149906; -3.120794];
B_y = [18.121528; 9.912528];
C_x = [12.777706; -1.492994];
C_y = [15.037528; 6.828528];
D_x = [1.380606; 15.301006; 14.405506; 0.134806; 1.380606];
D_y = [2.756528; 10.696528; 11.953528; 3.744528; 2.756528];
E_x = [18.39721694; 20.73791694; 13.59911694; 11.25831694;
    18.39721694];
E_y = [9.025287575; 10.36928757; 23.54228757; 22.19828757;
    9.025287575];
F_x = [6.783106; 2.529706; 5.012206; 9.265606; 6.783106];
F_y = [19.744528; 17.311528; 12.971528; 15.404528;
    19.744528];
G_x = [20.072206; 13.127906; 14.071306; 21.015606;
    20.072206];
G_y = [18.822528; 14.850528; 13.201528; 17.173528;
    18.822528];

figure(1)
hold on
plot(A_x, A_y, '-')
plot(B_x, B_y, '-')
plot(C_x, C_y, '-')
plot(D_x, D_y, '-')
plot(E_x,E_y,'-')
plot(F_x,F_y,'-')
plot(G_x,G_y,'-')
plot(x0, y0, '*')
```

```
acc = 2;


[Int_1(1), Int_1(2)] = Int_Point([x0, y0], [x0 + 20, y0 +
    20 * ((D_y(4) - D_y(3)) / (D_x(4) - D_x(3)))], [D_x(2),
    D_y(2)], [D_x(3), D_y(3)]);
plot(Int_1(1), Int_1(2), '*');


k = (D_y(3) - D_y(2)) / (D_x(3) - D_x(2));
Int_2(1) = Int_1(1) - sqrt((4.878 * 2)^2 / (1 + k^2));
Int_2(2) = Int_1(2) + sqrt((4.878 * 2)^2 / (1 + 1 / k^2));
plot(Int_2(1), Int_2(2), '*')


[Int_3(1), Int_3(2)] = Int_Point([Int_2(1), Int_2(2)], [
    Int_2(1) + 20, Int_2(2) + 20 * ((D_y(4) - D_y(3)) / (D_x
    (4) - D_x(3)))], [A_x(1), A_y(1)], [A_x(4), A_y(4)]);
plot(Int_3(1), Int_3(2), '*')


distance1 = pdist([Int_1; x0, y0], 'euclidean');
distance3 = pdist([Int_2; Int_3], 'euclidean');




curR1 = 4.878;
```

```
circle_center = [(Int_1(1) + Int_2(1)) / 2, (Int_1(2) +
   Int_2(2)) / 2];
plot(circle_center(1), circle_center(2), 'o')

distance2 = Arc_Length(Int_2, Int_1, circle_center);



i = 1;
distance = 0;

while i == 1;

    if distance <= distance1 + 5
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2
        curR = curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2 + 3
        curR = 0;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2 + 8
        curR = curR1;
        speed = 2;
        acc = 0;
```

```
    elseif distance <= distance1 + 5 + distance2 + 13
        curR = -curR1;
        speed = 2;
        acc = 0;
    elseif distance <= distance1 + 5 + distance2 + 10 +
       distance3
        curR = 0;
        speed = 2;
        acc = 0;
    else
        break
    end

    [NextTrackPointData, distance] = Next_TrackPoint(
       ALLTrackPointData(end, :), speed, acc, dt, curR,
       distance);

    ALLTrackPointData = [ALLTrackPointData;
       NextTrackPointData];

    i = inpolygon(NextTrackPointData(1), NextTrackPointData
       (2), A_x, A_y);
end

plot(ALLTrackPointData(:, 1), ALLTrackPointData(:, 2), '
   linestyle', '-', 'linewidth', 2, 'color', 'b');
```

E ArcLength

```
function Arc_Length = Arc_Length(P1, P2, O)
```

```
    r = sqrt((P1(1) - O(1))^2 + (P1(2) - O(2))^2);
    [alpha] = cart2pol(P1(1) - O(1), P1(2) - O(2));
    [beta] = cart2pol(P2(1) - O(1), P2(2) - O(2));
    Arc_Length = r * (abs(alpha - beta));
end
```

F drawArc

```
function drawArc(p1, p2, c)


    r = sqrt((p1(2) - c(2))^2 + (p1(1) - c(1))^2);
    [alpha] = cart2pol(p1(1) - c(1), p1(2) - c(2));
    [beta] = cart2pol(p2(1) - c(1), p2(2) - c(2));

    t = linspace(alpha, beta);
    x = c(1) + r * cos(t);
    y = c(2) + r * sin(t);
    plot(x, y, 'linestyle', '--', 'color', 'b');
end
```

G NextTrackPoint

```
function [NextTrackPointData, distance] = Next_TrackPoint(
   TrackPointData, speed, acceleration, dt, curvature_radius
   , distance)
```

```
x_data = TrackPointData(1, 1);
y_data = TrackPointData(1, 2);
theta_data = TrackPointData(1, 3);
curvature_data = TrackPointData(1, 4);
curvature_variation_data = TrackPointData(1, 5);
speed_data = speed;
acceleration_data = TrackPointData(1, 7);
t_data = TrackPointData(1, 8);

distance = distance + speed * dt;

t_next = t_data + dt;

if curvature_radius ~= 0
    curvature_next = 1 / curvature_radius;
    delta_theta = speed * dt / curvature_radius;
    length = 2 * curvature_radius * sin(delta_theta /
       2);
else
    curvature_next = 0;
    delta_theta = 0;
    length = speed * dt;
end

theta_next = theta_data + delta_theta;
x_next = x_data + length * cos((theta_data + theta_next
   ) / 2);
```

```
    y_next = y_data + length * sin((theta_data + theta_next
        ) / 2);

    curvature_variation_next = (curvature_next -
        curvature_data) / sqrt((x_next - x_data)^2 + (y_next
        - y_data)^2);
    speed_next = speed_data + acceleration * dt;
    acceleration_next = acceleration;

    NextTrackPointData = [x_next, y_next, theta_next,
        curvature_next, curvature_variation_next, speed_next,
         acceleration_next, t_next];
end
```