复现Apollo参考线平滑之分段spline平滑

描 述：算法来自于Baidu Apollo，本文是算法详解与代码的个人复现，供大家学习交流使用

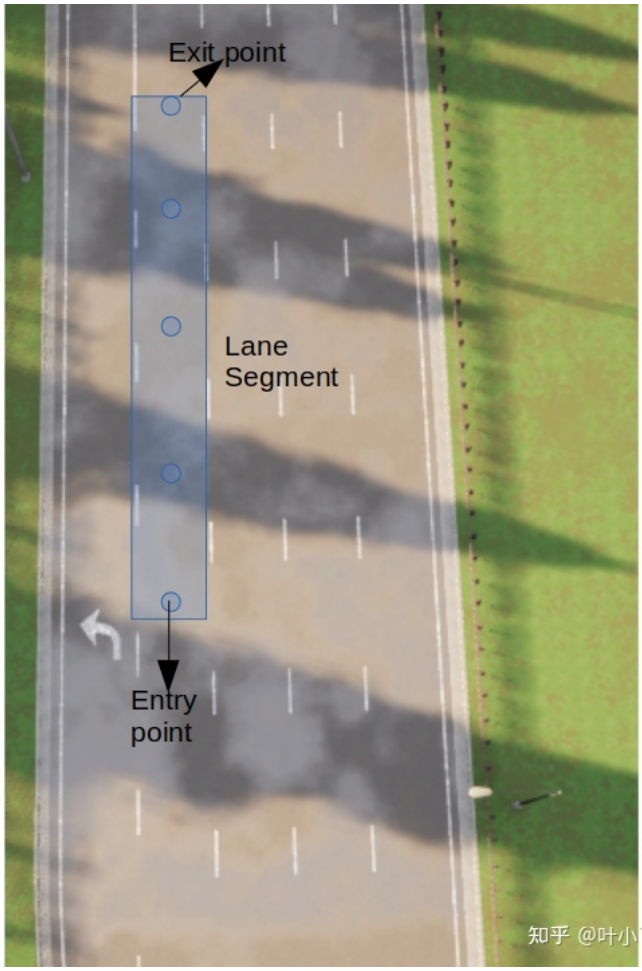
作 者：董泰宏

github: https://github.com/dongtaihong

邮 箱：2396203400@qq.com

0.引言

全局路径规划会生成离散的全局路径点，但是全局规划的算法往往是针对高精度地图构建的有向加权图进行运算的，以最常用的A*算法为例，它得出的结果是全局道路node的链表，然后通过这个链表再读取出高精度地图中每个node的道路中心线坐标点，大多数情况下，每个node内部的中心线坐标是以三次多项式表示（也有螺旋线、直线等类型），因此全局坐标点一定程度上是平滑的，但是各个node之间连接处却可能是割裂的，这就造成全局路径坐标点看起来比较平滑，但是包含了两个问题：1.平滑的程度不够；2.在路段连接点处有时会出现割裂、跳跃的情况。



于是，为了解决上述两个问题，就引入了参考线平滑模块：针对粗生成的全局路径点，进行一轮细致的平滑，使之满足全程连续、平滑、运动学可达的要求。

1.分段spline平滑算法

1.1 介绍

分段spline平滑的本质是5次样条曲线平滑，是利用优化算法的思想，生成每段路径的参数化曲线，每段路径的方程系数需要通过数值优化的方法去建模求解。

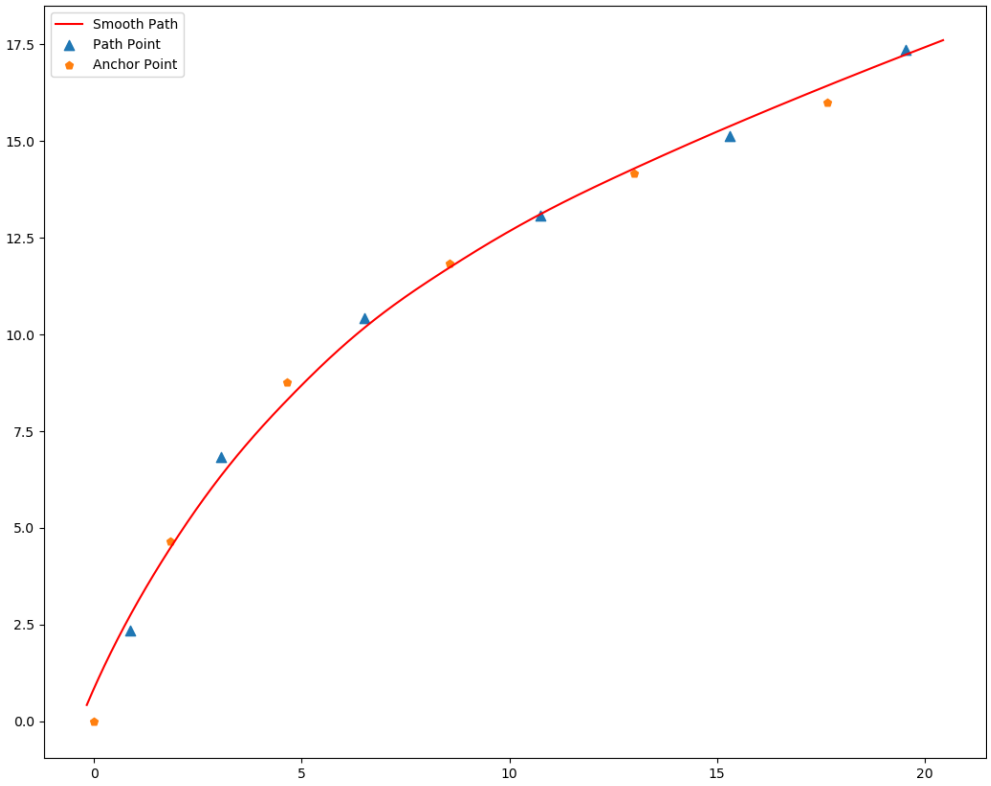
$$\begin{cases} x_i(t) = a_{i0} + a_{i1} \times t + a_{i2} \times t^2 + a_{i3} \times t^3 + a_{i4} \times t^4 + a_{i5} \times t^5 \\ y_i(t) = b_{i0} + b_{i1} \times t + b_{i2} \times t^2 + b_{i3} \times t^3 + b_{i4} \times b^4 + b_{i5} \times t^5 \end{cases}$$

PS:注意这里的t不是时间，是参数！t在[0, 1]之间。

为什么是五次？因为我们希望全局的路径能够满足运动学的一些特性，比如我们衡量驾驶舒适性的一个重要指标jerk，他是加速度a的导数，想要控制jerk，就必须精准的控制a，v，s这三个参数，我们得到一段曲线又需要起点和终点两个信息，于是就对应了6个控制参数，我们想要控制6个参数，那就刚好需要5次多项式。

为什么是分段？全局的路径形状很难通过一条曲线表述完成，需要由多段组合。

1.2 复现效果展示



2 算法剖析

二次规划的标准型(OSQP)：

$$\begin{aligned} minimize \quad & cost = \frac{1}{2} X^T Q X + P^T X \\ s.t \quad & L \leq AX \leq U \end{aligned}$$

从这个标准型问题中，我们可以看到构成问题的诸多要素：**目标函数 + 条件约束**。

PS：传统的QP问题只包含线性的约束项，OSQP库也是针对传统QP问题求解的库，然后很多情况下我们的约束项含有非线性项，此时就无法使用OSQP库了，但是有相应的解决方案（后面会有实例进行讲解）：

a.使用ipopt库（nlp库，适用题型广泛，但是建模比osqp复杂）；

b.将非线性约束转化为线性约束，然后继续使用OSQP。

2.1 目标函数

目标函数有两部分组成: **舒适项+正则化项**。TODO：这里还可以再加上整体曲线偏离原路径的cost项，防止平滑路径过分侧偏。

舒适项使用各自的3阶导作为目标函数，意在控制整体的jerk，虽然这里的x, y单独的jerk很难定义其物理意义，但总体上反映了操作的激进程度，是舒适性的抽象体现。

正则化项设置一个对角矩阵, 防止过拟合的现象。

$$
\begin{aligned}
cost &= cost_{comfort} + cost_{regular} \\
&= \sum_{i=0}^{n-1}\left(\int_{t_{i0}}^{t_{i1}}({x_i}''')^2(t)dt + \int_{t_{i0}}^{t_{i1}}({y_i}''')^2(t)dt\right) + \frac{1}{2}X^T Q_{regular} X \\
&\qquad 其中积分上下限为参数t的范围，t_{i0}=0, t_{i1}=1
\end{aligned}
$$

目标函数有两部分组成，则相应的Q矩阵也由两部分组成

$$Q_{comfort}，\ Q_{regular}$$

2.1.1 优化变量*X*

参数方程的系数（每一段有12个系数，那全局n个点，则对应12n个系数---ps: 6个点应该对应5段曲线啊？需要加上终点过后的一段）,以 *i* 段为例:

$$
\begin{cases}
\overrightarrow{a_i} = \begin{bmatrix} a_{i0} & a_{i1} & a_{i2} & a_{i3} & a_{i4} & a_{i5} \end{bmatrix}_{1*6} \\
\overrightarrow{b_i} = \begin{bmatrix} b_{i0} & b_{i1} & b_{i2} & b_{i3} & b_{i4} & b_{i5} \end{bmatrix}_{1*6}
\end{cases}
$$

所以有n个参考点时的输入变量为:

$$
\vec{X} = \begin{bmatrix} a_{00} & \dots & a_{05} & \dots & a_{(n-1)0} & \dots & a_{(n-1)5} & b_{00} & \dots & b_{05} & \dots & b_{(n-1)0} & \dots & b_{(n-1)5} \end{bmatrix}_{12n*1}
$$

2.1.2 Q矩阵

Q矩阵为X中每个元素的二次项系数，cost函数中

第一项：Qcomfort

$$
\begin{aligned}
{x_i}'(t) &= \begin{bmatrix} 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \end{bmatrix} \cdot \overrightarrow{a_i}^T \\
{x_i}''(t) &= \begin{bmatrix} 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \end{bmatrix} \cdot \overrightarrow{a_i}^T \\
{x_i}'''(t) &= \begin{bmatrix} 0 & 0 & 0 & 6 & 24t & 60t^2 \end{bmatrix} \cdot \overrightarrow{a_i}^T \\
{y_i}'(t) &= \begin{bmatrix} 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \end{bmatrix} \cdot \overrightarrow{b_i}^T \\
{y_i}''(t) &= \begin{bmatrix} 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \end{bmatrix} \cdot \overrightarrow{b_i}^T \\
{y_i}'''(t) &= \begin{bmatrix} 0 & 0 & 0 & 6 & 24t & 60t^2 \end{bmatrix} \cdot \overrightarrow{b_i}^T \\
&其中，记 \begin{bmatrix} 0 & 0 & 0 & 6 & 24t & 60t^2 \end{bmatrix}_{1*6} 为 T'''(t)
\end{aligned}
$$

于是:

$$
\begin{aligned}
\int_{t_{i0}}^{t_{i1}} {x_i'''}^2(t)dt &= \overrightarrow{a_i}\int_{t_{i0}}^{t_{i1}} (T'''(t)^T \cdot T'''(t)\cdot \overrightarrow{a_i}^T)dt \\
&= \overrightarrow{a_i}\int_{t_{i0}}^{t_{i1}} \left(\begin{bmatrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&36&144t&360t^2\\ 0&0&0&144t&576t^2&1440t^3\\ 0&0&0&360t^2&1440t^3&3600t^4 \end{bmatrix} \cdot \overrightarrow{a_i}^T \right) dt \\
&= \overrightarrow{a_i} \begin{bmatrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&36&72&120\\ 0&0&0&72&192&360\\ 0&0&0&120&360&720 \end{bmatrix} \cdot \overrightarrow{a_i}^T
\end{aligned}
$$

同理 $\int_{t_{i0}}^{t_{i1}} {y_i'''}^2(t)dt = \overrightarrow{b_i} \begin{bmatrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&36&72&120\\ 0&0&0&72&192&360\\ 0&0&0&120&360&720 \end{bmatrix} \cdot \overrightarrow{b_i}^T$

整理成二次型就是:

$$
Q_{comfort} = 2 \cdot \begin{bmatrix} 0&0&0&0&0&0&\dots&0&0&0&0&0&0\\ 0&0&0&0&0&0&\dots&0&0&0&0&0&0\\ 0&0&0&0&0&0&\dots&0&0&0&0&0&0\\ 0&0&0&36&72&120&\dots&0&0&0&0&0&0\\ 0&0&0&72&192&360&\dots&0&0&0&0&0&0\\ 0&0&0&120&360&720&\dots&0&0&0&0&0&0\\ &&&&&&\dots&&&&&&\\ 0&0&0&0&0&0&\dots&0&0&0&0&0&0\\ 0&0&0&0&0&0&\dots&0&0&0&0&0&0\\ 0&0&0&0&0&0&\dots&0&0&0&0&0&0\\ 0&0&0&0&0&0&\dots&0&0&0&36&72&120\\ 0&0&0&0&0&0&\dots&0&0&0&72&192&360\\ 0&0&0&0&0&0&\dots&0&0&0&120&360&720 \end{bmatrix}_{12n*12n}
$$

第二项：Qregular

正则化惩罚项按照Apollo设置为对角矩阵:

$$
Q_{regular} = 2 \cdot \begin{bmatrix} 10^{-5}&0&0&0&0&0&\dots&0&0&0&0&0&0\\ 0&10^{-5}&0&0&0&0&\dots&0&0&0&0&0&0\\ 0&0&10^{-5}&0&0&0&\dots&0&0&0&0&0&0\\ 0&0&0&10^{-5}&0&0&\dots&0&0&0&0&0&0\\ 0&0&0&0&10^{-5}&0&\dots&0&0&0&0&0&0\\ 0&0&0&0&0&10^{-5}&\dots&0&0&0&0&0&0\\ &&&&&&\dots&&&&&&\\ 0&0&0&0&0&0&\dots&10^{-5}&0&0&0&0&0\\ 0&0&0&0&0&0&\dots&0&10^{-5}&0&0&0&0\\ 0&0&0&0&0&0&\dots&0&0&10^{-5}&0&0&0\\ 0&0&0&0&0&0&\dots&0&0&0&10^{-5}&0&0\\ 0&0&0&0&0&0&\dots&0&0&0&0&10^{-5}&0\\ 0&0&0&0&0&0&\dots&0&0&0&0&0&10^{-5} \end{bmatrix}_{12n*12n}
$$

所以标准二次型中的:

$$Q_{12n*12n} = Q_{comfort} + Q_{regular}$$

2.1.3 P矩阵

在本问题中，从目标函数可见，并没有一次项，所以P矩阵为零矩阵，维度为1*12n。

$$P = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \end{bmatrix}_{1*12n}$$

2.2 约束条件

2.2.1 连续约束

由于每段spline是单独的，但整体拼接起来我们希望能够足够的平滑，这要求段与段之间的连接点0～3阶导都是连续连续的。

$$\begin{array}{ll} x_i(t_{i1}) = x_{i+1}(t_{(i+1)0}) & \quad | \quad y_i(t_{i1}) = y_{i+1}(t_{(i+1)0}) \\ x_i'(t_{i1}) = x_{i+1}'(t_{(i+1)0}) & \quad | \quad y_i'(t_{i1}) = y_{i+1}'(t_{(i+1)0}) \\ x_i''(t_{i1}) = x_{i+1}''(t_{(i+1)0}) & \quad | \quad y_i''(t_{i1}) = y_{i+1}''(t_{(i+1)0}) \\ x_i'''(t_{i1}) = x_{i+1}'''(t_{(i+1)0}) & \quad | \quad y_i'''(t_{i1}) = y_{i+1}'''(t_{(i+1)0}) \end{array}$$

每两段曲线有一个连接点，对应4x2个条件，那么n段曲线的约束规模就是4x2x(n-1)。

以x参数为例子:

$$\begin{aligned} 0 &= x_i(t_{i1}) - x_{i+1}(t_{(i+1)0}) \\ &= a_{i0} + a_{i1} \cdot t_{i1} + \ldots + a_{i5} \cdot t_{i1}^5 - a_{(i+1)0} + a_{(i+1)1} \cdot t_{(i+1)0} + \ldots + a_{(i+1)5} \cdot t_{(i+1)0}^5 \\ &= \begin{bmatrix} 1 & t_{i1} & t_{i1}^2 & t_{i1}^3 & t_{i1}^4 & t_{i1}^5 & -1 & -t_{i0} & -t_{i0}^2 & -t_{i0}^3 & -t_{i0}^4 & -t_{i0}^5 \end{bmatrix} \cdot \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{(i+1)0} \\ a_{(i+1)1} \\ a_{(i+1)2} \\ a_{(i+1)3} \\ a_{(i+1)4} \\ a_{(i+1)5} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{(i+1)0} \\ a_{(i+1)1} \\ a_{(i+1)2} \\ a_{(i+1)3} \\ a_{(i+1)4} \\ a_{(i+1)5} \end{bmatrix} \end{aligned}$$

同理，一阶导、二阶导、三阶导分别对t求导，然后带入ti0=0, ti1=1即可:

$$
\begin{aligned}
0 &= x_i\prime(t_{i1}) - x_{i+1}\prime(t_{(i+1)0}) \\
&= \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{(i+1)0} \\ a_{(i+1)1} \\ a_{(i+1)2} \\ a_{(i+1)3} \\ a_{(i+1)4} \\ a_{(i+1)5} \end{bmatrix} \\
0 &= x_i\prime\prime(t_{i1}) - x_{i+1}\prime\prime(t_{(i+1)0}) \\
&= \begin{bmatrix} 0 & 0 & 2 & 6 & 12 & 20 & 0 & 0 & -2 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{(i+1)0} \\ a_{(i+1)1} \\ a_{(i+1)2} \\ a_{(i+1)3} \\ a_{(i+1)4} \\ a_{(i+1)5} \end{bmatrix} \\
0 &= x_i\prime\prime\prime(t_{i1}) - x_{i+1}\prime\prime\prime(t_{(i+1)0}) \\
&= \begin{bmatrix} 0 & 0 & 2 & 6 & 24 & 60 & 0 & 0 & 0 & -6 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{(i+1)0} \\ a_{(i+1)1} \\ a_{(i+1)2} \\ a_{(i+1)3} \\ a_{(i+1)4} \\ a_{(i+1)5} \end{bmatrix}
\end{aligned}
$$

标准约束形式为：

$$
\begin{gathered}
A_{continue} \cdot X = 0 \\
\Rightarrow \vec{0} \leq A_{continue} \cdot X \leq \vec{0}
\end{gathered}
$$

$$
\text{其中} A_{continue} \text{维度为} [4 * 2 * (n-1), 12n], \text{其中} \vec{0} \text{维度为} [4 * 2 * (n-1), 1]
$$

其中A-continue的矩阵构造这里不展开了，参照上面的公式结合我的代码一起看。

2.2.2 方向约束

整条轨迹的起点方向与终点方向与初始值保持一致。所以约束的维度只有2。

$$
\begin{aligned}
heading_{start} &= arctan(y_0\prime(t0), x_0\prime(t0)) \\
&= \frac{y_0\prime(0)}{x_0\prime(0)} \\
&= \frac{0 + b_{01} + 2b_{02}\cdot t_0(0) + 3b_{03}\cdot t_0(0)^2 + 4b_{04}\cdot t_0(0)^3 + 5b_{05}\cdot t_0(0)^4}{0 + a_{01} + 2a_{02}\cdot t_0(0) + 3a_{03}\cdot t_0(0)^2 + 4a_{04}\cdot t_0(0)^3 + 5a_{05}\cdot t_0(0)^4} \\
heading_{end} &= arctan(y_{n-1}\prime(t1), x_{n-1}\prime(t1)) \\
&= \frac{y_{n-1}\prime(1)}{x_{n-1}\prime(1)} \\
&= \cdots\cdots
\end{aligned}
$$

把上述等式转化成矩阵等式形式：

$$
\begin{aligned}
0 &= \begin{bmatrix} 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \end{bmatrix} \cdot \begin{bmatrix} b_{00} \\ b_{01} \\ b_{02} \\ b_{03} \\ b_{04} \\ b_{05} \end{bmatrix} - heading_{start} \cdot \begin{bmatrix} 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \end{bmatrix} \cdot \begin{bmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{03} \\ a_{04} \\ a_{05} \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} b_{00} \\ b_{01} \\ b_{02} \\ b_{03} \\ b_{04} \\ b_{05} \end{bmatrix} - heading_{start} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{03} \\ a_{04} \\ a_{05} \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
0 &= \begin{bmatrix} 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \end{bmatrix} \cdot \begin{bmatrix} b_{(n-1)0} \\ b_{(n-1)1} \\ b_{(n-1)2} \\ b_{(n-1)3} \\ b_{(n-1)4} \\ b_{(n-1)5} \end{bmatrix} - heading_{start} \cdot \begin{bmatrix} 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \end{bmatrix} \cdot \begin{bmatrix} a_{(n-1)0} \\ a_{(n-1)1} \\ a_{(n-1)2} \\ a_{(n-1)3} \\ a_{(n-1)4} \\ a_{(n-1)5} \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} b_{(n-1)0} \\ b_{(n-1)1} \\ b_{(n-1)2} \\ b_{(n-1)3} \\ b_{(n-1)4} \\ b_{(n-1)5} \end{bmatrix} - heading_{end} \cdot \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} a_{(n-1)0} \\ a_{(n-1)1} \\ a_{(n-1)2} \\ a_{(n-1)3} \\ a_{(n-1)4} \\ a_{(n-1)5} \end{bmatrix}
\end{aligned}
$$

标准约束形式为：

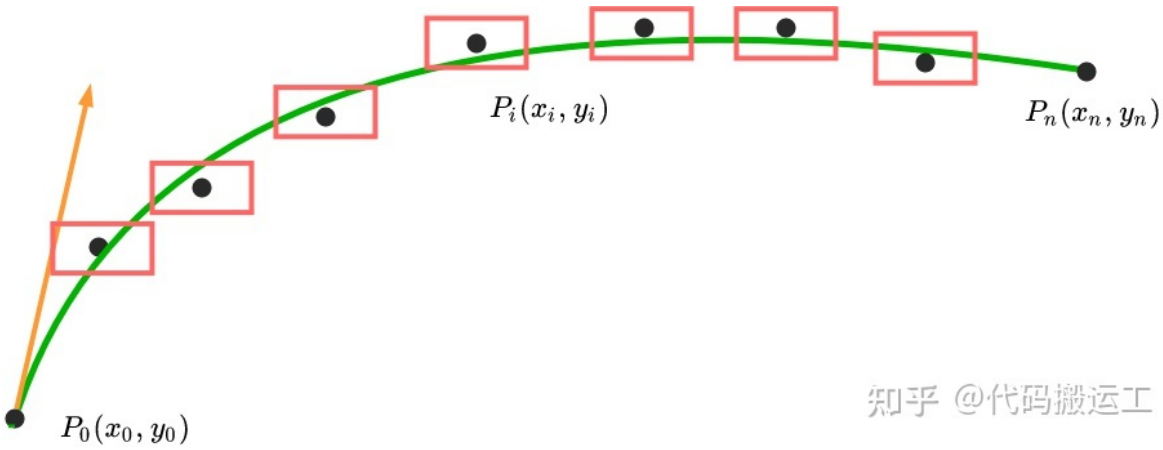
$$
\begin{gathered}
A_{start-end} \cdot X = 0 \\
\Rightarrow \vec{0} \le A_{start-end} \cdot X \le \vec{0}
\end{gathered}
$$

$$
\text{其中 } A_{continue} \text{维度为 } [2, 12n], \vec{0} \text{维度为 } [2, 1]
$$

其中A-startend的矩阵构造这里不展开了，参照上面的公式结合我的代码一起食用。

2.2.3 位置约束

平滑参考线时，我们希望曲线能够平滑就用到了五次多项式，然而如果严格固定了参考点的话，如果两个点的间距又很小，则很可能造成龙阁库塔现象，所以需要对参考线提供一定的裕度。



$$x_i(t_l) - x_l \leq boundary$$
$$y_i(t_l) - y_l \leq boundary$$

为什么要用鞍点？我们选择每段曲线的中点为anchor point（也可以直接使用原始参考点作为anchor point，这样的话t=0），去进行位置约束考虑，同时终点之后的那段不考虑。则位置约束的维度为2x(n-1)。

写成矩阵形式：

$$x_i(t_{0.5}) \leq x_l + boundary$$
$$\begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \end{bmatrix}_{t=0.5} \cdot \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{bmatrix} \leq x_l + boundary$$
$$y_i(t_{0.5}) \leq y_l + boundary$$
$$\begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \end{bmatrix}_{t=0.5} \cdot \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{bmatrix} \leq y_l + boundary$$

标准约束形式为：

$$A_{position} \cdot X = 0$$
$$\Rightarrow \begin{bmatrix} x_l \\ y_l \end{bmatrix} - boundary \leq A_{start-end} \cdot X \leq \begin{bmatrix} x_l \\ y_l \end{bmatrix} + boundary$$
$$\text{其中 } A_{position} \text{维度为 } [2*(n-1), 12n], \begin{bmatrix} x_l \\ y_l \end{bmatrix} \text{维度为 } [2*(n-1), 1]$$

其中A-position的矩阵构造这里不展开了，参照上面的公式结合我的代码一起看。

那么**总体的约束**为：

$$lowerbound \leq A \cdot X = \begin{bmatrix} A_{continue} \\ A_{start-end} \\ A_{position} \end{bmatrix} \cdot X \leq upperbound$$
$$\text{其中 } A \text{维度为 } [10*(n-1)+2, 12n], bound \text{维度为 } [10*(n-1)+2, 1]$$

3 算法仿真

3.1代码注意事项与调试指南

1.本文的代码是每隔5m采样一个路径点，然后在每两个路径点中间也就是2.5m左右采样一个鞍点，你们可以自行修改这个采样间隔。因此你的全局路径输入数据的密度不能低于2.5m/per point。

```
//此处代码位于SplineSmoother.cpp 59~68行
    if (distance > 4.7 && distance < 5.3) {
      input_points.emplace_back(temp_point);
      input_index++;
    }
    //在两个参考点的中间也就是2.5米左右采一个anchor point
    if (distance > 2.2 && distance < 2.8) {
      anchor_points.emplace_back(temp_point);
      anchor_index++;
    }
```

2.本文的位置约束是0.2m，也就是平滑的轨迹不能偏离鞍点超过0.2m。

```
//此处代码位于SplineSmoother.hpp 61行
double boundary = 0.2;
```

3.本文的目标函数只考虑了舒适项，有可能发生整体曲线以位置约束boundary值偏离原路径情况，因此需要加入偏离cost项。

```
//此处可在SplineSmoother.cpp 87行中自行添加
Eigen::SparseMatrix<double> SplineSmoother::CostFunction(){}
```

3.2具体代码

本文2.1.2节对应代码：Eigen::SparseMatrix CostFunction();

2.2节对应代码：void ConstriantFunction();

问题构造完成后调用OSQP求解：bool Solution();

SplineSmoother.hpp

```
#include <vector>
#include <iostream>
#include <string>
#include <fstream>
#include <sstream>
#include <cmath>
//#include <chrono>

//依赖的三方库
#include "matplotlibcpp.h"
//#include "spdlog/spdlog.h"
#include "OsqpEigen/OsqpEigen.h"

using namespace std;
namespace plt = matplotlibcpp;

class SplineSmoother
{
public:
```

```
    SplineSmoother(const char* filename);
    ~SplineSmoother() = default;
private:
    //step1: 整理输入的数据
    void GetSourceData(const char* filename);
    //step2: 构建目标函数的Q矩阵
    Eigen::SparseMatrix<double> CostFunction();
    //step3: 构建目标函数的A矩阵以及约束边界
    void ConstriantFunction();
    //step4: 求解问题
    bool Solution();
    //step5: 从参数方程转化为x,y路径点
    void ParametricToCartesian();
public:
    int number;  //需要平滑的路径点的数量
    vector<pair<double, double>> input_points;    //原始路径点
    double heading_start;     //
    double heading_end;
    vector<pair<double, double>> anchor_points;   //限制路径的鞍点

    //这里matrix类对象需要用vector包一下，因为类对象单独作为成员函数，只能使用初始化列表初始化，非常受限制
    vector<Eigen::SparseMatrix<double>> Q;

    vector<Eigen::VectorXd> Derivative;
    vector<Eigen::VectorXd> bound;
    vector<Eigen::VectorXd> gradient;
    vector<Eigen::SparseMatrix<double>> A;
    //位置约束的范围
    double boundary = 0.2;

    int total_constriant_number;
    vector<Eigen::VectorXd> smooth_parametric;
    vector<pair<double, double>> smooth_points;   //result路径点
};
```

4.参考资料

参 考: https://github.com/ApolloAuto/apollo/tree/r6.0.0

https://zhuanlan.zhihu.com/p/345175084