技术文档丨二次规划（QP）样条路径

Apollo的Planning分为**参考线平滑、决策、路径规划、速度规划**等部分。

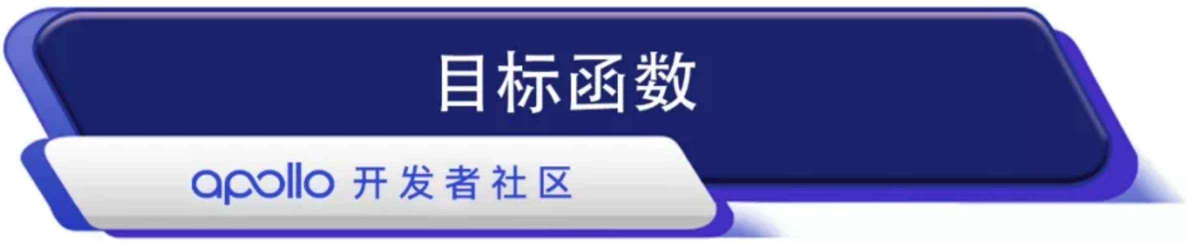
从整体上来说，规划模块的架构分为两个部分：一部分负责对数据的监听、获取和预处理；另一部分负责管理各个优化模块。数据进入后，对其综合处理为规划模块的内部数据结构，由任务管理器调度合适的优化器进行各个优化任务。综合优化的结果，经过最终的验证后，输出给控制模块。

在设计上，实现了策略的可插拔，使得各个优化器可以灵活配置不同策略，提升迭代效率。

> **EM-Planner**是具体的规划实施类，它基于高精地图、导航路径及障碍物信息作出实际的驾驶决策，包括路径、速度等方面。
>
> 首先使用**DP（动态规划）**方法确定初始的路径和速度，再利用**QP（二次规划）**方法进一步优化路径和速度，以得到一条更平滑的轨迹，既满足舒适性，又方便车辆操纵。
>
> 基于样条的**车辆轨迹优化二次规划**，为了寻求更优质更平滑，体感更好的路径，需要使用二次规划的方法寻找。需要的限制条件有：曲率和曲率连续性、贴近中心线、避免碰撞。

今天，就让阿波君和开发者们一起了解**二次规划样条路径**是如何实现的。

以下，ENJOY：



获得路径长度

路径定义在 `station-lateral`坐标系中。

参数**s的取值范围为车辆的当前位置到默认规划路径的长度。**

获得样条段

将路径划分为n段，每段路径用一个**多项式**来表示。

定义样条段函数

每个样条段 i 都有沿着参考线的累加距离 d_i 。每段的路径默认用5阶多项式表示：

$$l = f_i(s) = a_{i0} + a_{i1} \cdot s + a_{i2} \cdot s^2 + a_{i3} \cdot s^3 + a_{i4} \cdot s^4 + a_{i5} \cdot s^5 (0 \le s \le d_i)$$

定义每个样条段优化目标函数

$$cost = \sum_{i=1}^{n} \left(w_1 \cdot \int_0^{d_i} (f_i')^2(s) ds + w_2 \cdot \int_0^{d_i} (f_i'')^2(s) ds + w_3 \cdot \int_0^{d_i} (f_i''')^2(s) ds \right)$$

将开销（cost）函数转换为QP公式

QP公式如下所示：

$$
\begin{aligned}
& minimize \frac{1}{2} \cdot x^T \cdot H \cdot x + f^T \cdot x \\
& s.t. \quad LB \le x \le UB \\
& \qquad A_{eq}x = b_{eq} \\
& \qquad Ax \ge b
\end{aligned}
$$

下面是将开销（cost）函数转换为QP公式的例子：

$$
f_i(s) = \begin{vmatrix} 1 & s & s^2 & s^3 & s^4 & s^5 \end{vmatrix} \cdot \begin{vmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{vmatrix}
$$

且

$$
f_i'(s) = \begin{vmatrix} 0 & 1 & 2s & 3s^2 & 4s^3 & 5s^4 \end{vmatrix} \cdot \begin{vmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{vmatrix}
$$

且

然后得到，

从聚合函数中提取出常量得到，

最后得到，

请注意我们最后得到一个6阶的矩阵来表示5阶样条插值的衍生开销。

应用**同样的推理方法**可以得到2阶，3阶样条插值的衍生开销。

假设第一个点为，和。

其中，表示横向的偏移，并且规划路径的起始点的第一，第二个点的衍生开销可以

从，计算得到。

将上述约束转换为QP约束等式，使用等式：

下面是转换的具体步骤:

且

且

其中，i 是包含　　　的样条段的索引值。

和起始点相同，终点　　　也应当按照**起始点的计算方法生成约束条件**。

将起始点和终点组合在一起，得出约束等式为：

该约束的目的是**使样条的节点更加平滑**。假设两个段 和 互相连接，且 的累计值s为 。计算约束的等式为：

$$f_k(s_k) = f_{k+1}(s_0)$$

下面是计算的具体步骤：

$$\begin{vmatrix} 1 & s_k & s_k^2 & s_k^3 & s_k^4 & s_k^5 \end{vmatrix} \cdot \begin{vmatrix} a_{k0} \\ a_{k1} \\ a_{k2} \\ a_{k3} \\ a_{k4} \\ a_{k5} \end{vmatrix} = \begin{vmatrix} 1 & s_0 & s_0^2 & s_0^3 & s_0^4 & s_0^5 \end{vmatrix} \cdot \begin{vmatrix} a_{k+1,0} \\ a_{k+1,1} \\ a_{k+1,2} \\ a_{k+1,3} \\ a_{k+1,4} \\ a_{k+1,5} \end{vmatrix}$$

然后，

$$\begin{vmatrix} 1 & s_k & s_k^2 & s_k^3 & s_k^4 & s_k^5 & -1 & -s_0 & -s_0^2 & -s_0^3 & -s_0^4 & -s_0^5 \end{vmatrix} \cdot \begin{vmatrix} a_{k0} \\ a_{k1} \\ a_{k2} \\ a_{k3} \\ a_{k4} \\ a_{k5} \\ a_{k+1,0} \\ a_{k+1,1} \\ a_{k+1,2} \\ a_{k+1,3} \\ a_{k+1,4} \\ a_{k+1,5} \end{vmatrix} = 0$$

将 $S_0=0$ 代入等式。

同样地，可以为下述等式计算约束等式：

$$f_k'(s_k) = f_{k+1}'(s_0)$$
$$f_k''(s_k) = f_{k+1}''(s_0)$$
$$f_k'''(s_k) = f_{k+1}'''(s_0)$$

点采样边界约束

在路径上均匀的取样m个点，**检查这些点上的障碍物边界**。将这些约束转换为QP约束不等式，使用不等式：

$$Ax \geq b$$

首先基于道路宽度和周围的障碍物找到点 (s_j, l_j) 的下边界 $l_{lb,j}$ ，且 $j \in [0,m]$ 。计算约束的不等式为：

$$\begin{vmatrix} 1 & s_0 & s_0^2 & s_0^3 & s_0^4 & s_0^5 \\ 1 & s_1 & s_1^2 & s_1^3 & s_1^4 & s_1^5 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & s_m & s_m^2 & s_m^3 & s_m^4 & s_m^5 \end{vmatrix} \cdot \begin{vmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{vmatrix} \geq \begin{vmatrix} l_{lb,0} \\ l_{lb,1} \\ \cdots \\ l_{lb,m} \end{vmatrix}$$

同样地，对上边界 $l_{ub,j}$ ，计算约束的不等式为：

$$\begin{vmatrix} -1 & -s_0 & -s_0^2 & -s_0^3 & -s_0^4 & -s_0^5 \\ -1 & -s_1 & -s_1^2 & -s_1^3 & -s_1^4 & -s_1^5 \\ \dots & \dots - & \dots & \dots & \dots & \dots \\ -1 & -s_m & -s_m^2 & -s_m^3 & -s_m^4 & -s_m^5 \end{vmatrix} \cdot \begin{vmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{vmatrix} \geq -1 \cdot \begin{vmatrix} l_{ub,0} \\ l_{ub,1} \\ \dots \\ l_{ub,m} \end{vmatrix}$$

END