

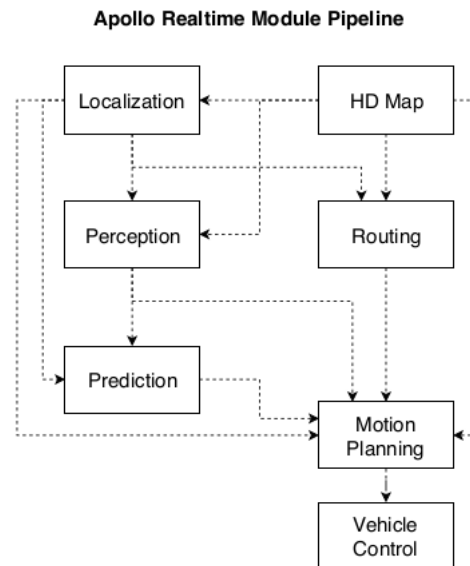
【Baidu Apollo】EM规划算法

笔记本: 我的第一个笔记本
创建时间: 2019/4/27 9:42
作者: 977616672ylycn
URL: <https://blog.csdn.net/yuxuan20062007/article/details/83629595>

更新时间: 2019/4/28 14:27

Apollo EM规划算法

1. Apollo自动驾驶平台车载模块



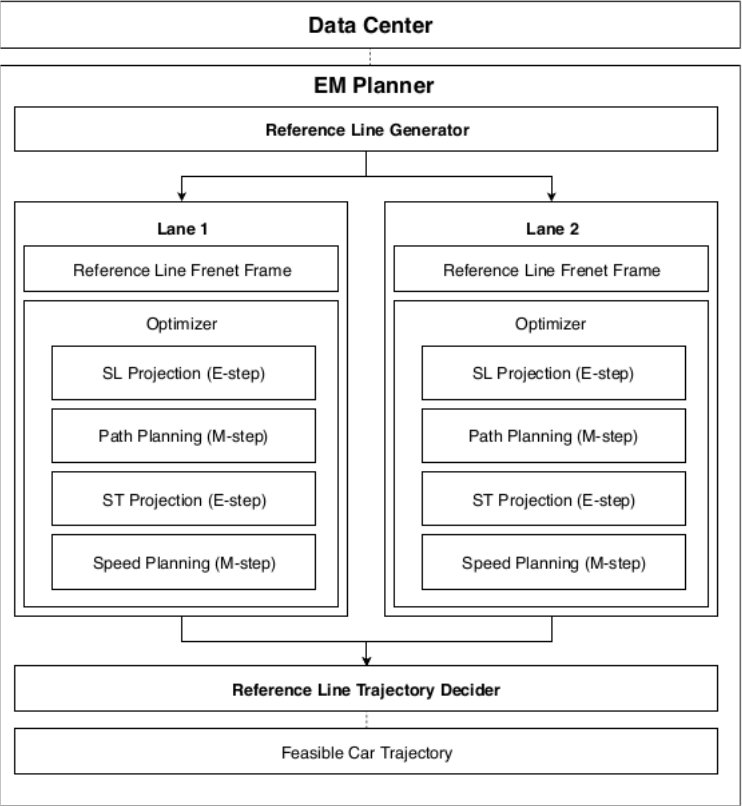
2. 变道超车在EM Planner中是如何考虑的？

对于路径规划而言，变道是重要的组成部分。一般的方法是在所有可能的车道上进行基于cost的最优解搜索。不过这种方法也有以下的缺点：

搜索空间要拓展到不同的车道，算法资源占用相对较多。

不同车道对应了不同的交规，在同样的计算框架下不容易实现。

因此在apollo里面使用了如下的Framework，对不同可能的目标车道分别进行优化寻找最优解，最后的Decider决定最终的路径。



最后的Trajectory decider会根据不同Trajectory的cost， 交规还有车况选择一条最优的trajectory。

3. Path 和 Speed 迭代算法

Path

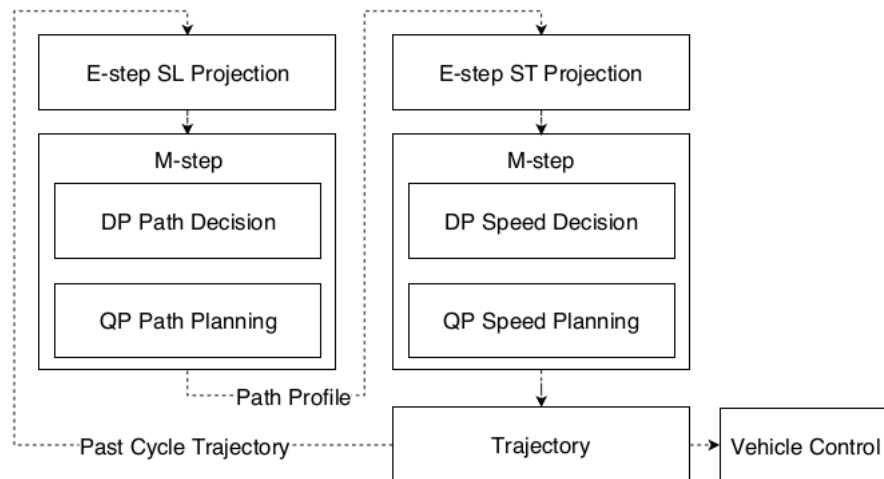
EM Planner会迭代式的对路径和速度进行优化。首先基于上一步计算得到的trajectory也就是

对自身运动的预测与其他运动物体的预测进行碰撞检测， 讲可能的碰撞点投射到SL图中。这里在

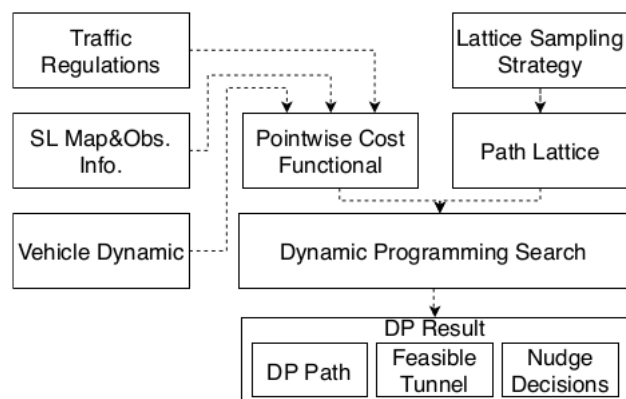
SL图中只考虑低速移动物体或对向行驶的物体， 对于高速物体， EM Planner基于安全原因更倾向于变道。

E步骤中， 将障碍物投射到SL及ST坐标系。M步骤通过动态规划和二次规划组合生成路径和速度曲线。

路径优化器包括两个步骤： 基于动态规划的路径决策和基于样条的路径规划。整个路径的规划分为DP的粗糙规划， 和QP的平滑规划， QP主要就是对DP的路径进行平滑处理。



DP路径包括格子采样器、成本函数和动态规划。



(1) 格子采样器基于Frenet坐标。首先根据车速、道路结构、车道变换等确定间隔。不同行之间的点通过五次多项式平滑连接。

(2) 这里Cost有三个组成部分：平滑Cost，离障碍物距离Cost，离引导线偏差Cost。

$$C_{total}(f(s)) = C_{smooth}(f) + C_{obs}(f) + C_{guidance}(f)$$

$$C_{smooth}(f) = w_1 \int (f'(s))^2 ds + w_2 \int (f''(s))^2 ds + w_3 \int (f'''(s))^2 ds.$$

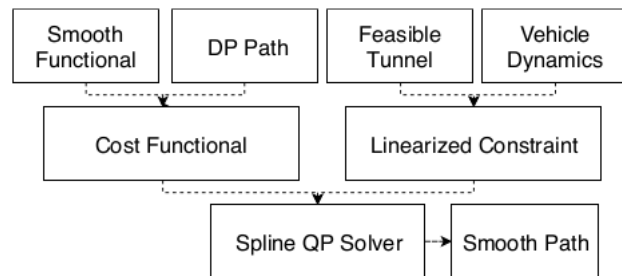
$$C_{obs}(d) = \begin{cases} 0, & d > d_n \\ C_{nudge}(d - d_c), & d_c \leq d \leq d_n \\ C_{collision} & d < d_c \end{cases}$$

$$C_{guidance}(f) = \int (f(s) - g(s))^2 ds$$

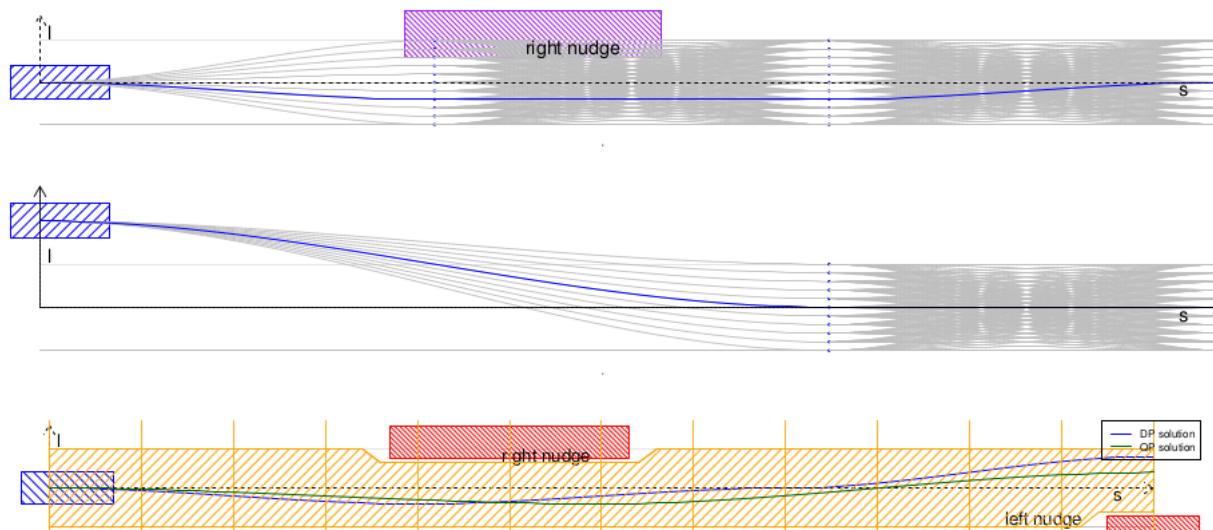
DP结果为：DP路径（分段五次多项式）、可行域、nudge决策。

Nudge Decision包括nudge, yield和overtake，用来生成convex hull凸域来进行QP的spline优化。

QP的Cost相对简单些，就是负责平滑的路径一阶二阶三阶倒数还有和DP结果与引导线的偏差。

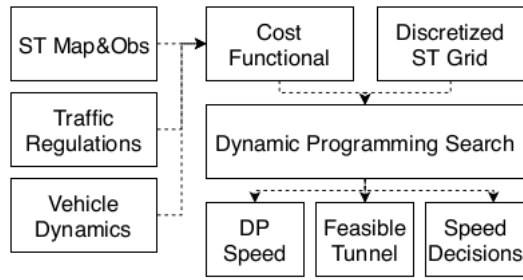


$$C_s(f) = w_1 \int (f'(s))^2 ds + w_2 \int (f''(s))^2 ds + w_3 \int (f'''(s))^2 ds + w_4 \int (f(s) - g(s))^2 ds.$$



Speed

速度优化和路径优化类似，也是先来DP跟着QP。



速度DP Cost多了和目标速度的偏差和与障碍物的距离。分段线性速度曲线。

$$s'_i = v_i \approx \frac{s_i - s_{i-1}}{dt}$$

$$s''_i = a_i \approx \frac{s_i - 2s_{i-1} + s_{i-2}}{(dt)^2}$$

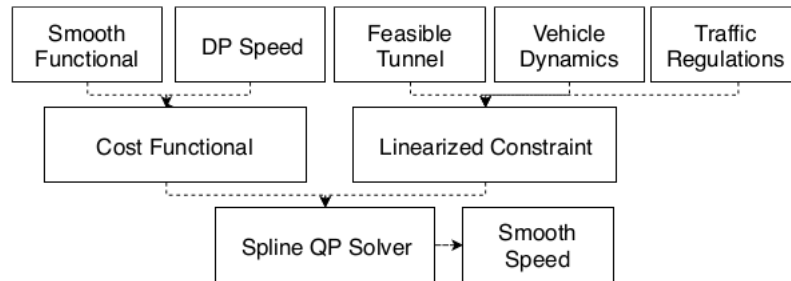
$$s'''_i = j_i \approx \frac{s_i - 3s_{i-1} - 3s_{i-2} + s_{i-3}}{(dt)^3}$$

$$C_{total}(S) = w_1 \int_{t_0}^{t_n} g(S' - V_{ref}) dt$$

$$+ w_2 \int_{t_0}^{t_n} (S'')^2 dt + w_3 \int_{t_0}^{t_n} (S''')^2 dt$$

$$+ w_4 C_{obs}(S)$$

速度QP



The cost functional is described as follows:

$$C_{total}(S) = w_1 \int_{t_0}^{t_n} (S - S_{ref})^2 dt + w_2 \int_{t_0}^{t_n} (S'')^2 dt$$

$$+ w_3 \int_{t_0}^{t_n} (S''')^2 dt.$$

$$S(t_i) \leq S(t_{i+1}), i = 0, 1, 2, \dots, n - 1,$$

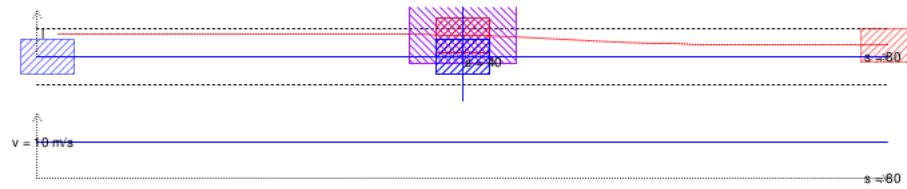
$$S_{l,t_i} \leq S(t_i) \leq S_{u,t_i},$$

$$S'(t_i) \leq V_{upper},$$

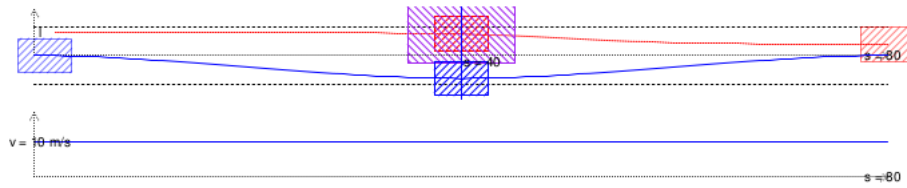
$$-Dec_{max} \leq S''(t_i) \leq Acc_{max}$$

$$-J_{max} \leq S'''(t_i) \leq J_{max}$$

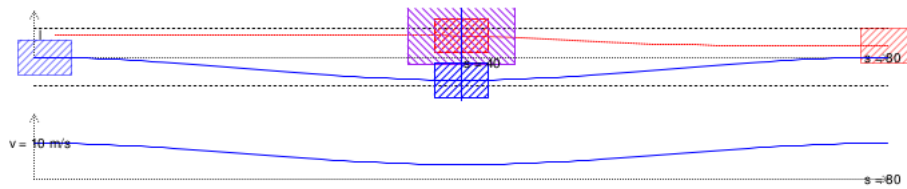
4.实例



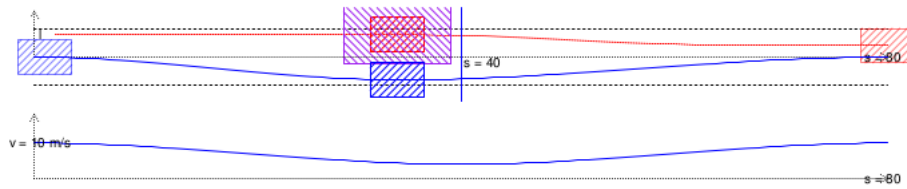
(a) Stage A: Historical Planning



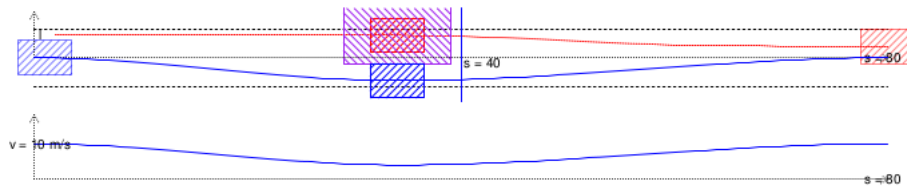
(b) Stage B: Path Planning Cycle 1



(c) Stage C: Speed Planning Cycle 1



(d) Stage D: Path Planning Cycle 2



(e) Stage E: Speed Planning Cycle 2

- 上一个cycle的规划，master vehicle 10m/s直行。
- 第一次迭代，基于上一个cycle的预测，可能的碰撞区域位于 $s = 40\text{m}$ 处，映射到SD Graph进行path优化。
- 基于优化得到的Path进行速度优化，40m处因为距离障碍物较近，会减速到5m/s。
- 这个时候因为 master vehicle 的预测更新了，进行第二次迭代，基于新的速度Profile，可能的碰撞（交叉）区域位于

S = 30m 处, 重新优化Path

e. 基于新优化得到的Path再次进行速度优化, 迭代会一直进行直到收敛。