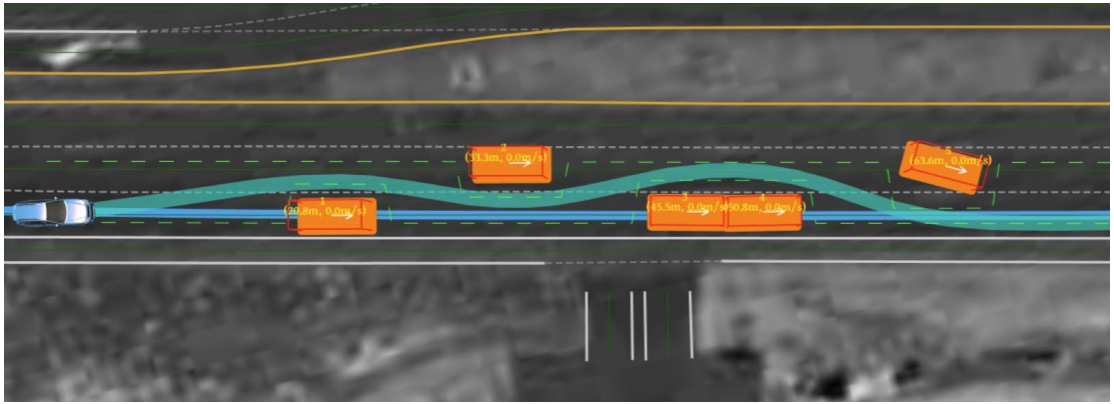
深蓝学院
shenlanxueyuan.com

Apollo 中的规划算法

目录

综述

规划综述



全局路径规划：计算出一条从起点到终点的最佳道路行驶序列

高精地图、拓扑地图、最短路径搜索

行为决策：综合感知预测、地图和定位信息等，决定合适的行驶策略

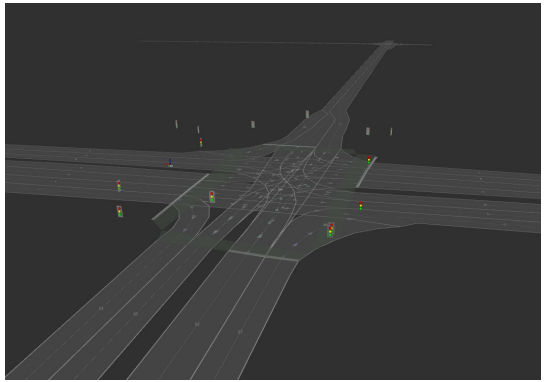
历史决策、障碍物信息、交通标识、交通规则

局部轨迹规划：基于环境地图寻找一条满足车辆运动学/动力学约束和舒适性指标的无碰撞轨迹

参考线平滑、路径规划、速度规划、搜索+优化

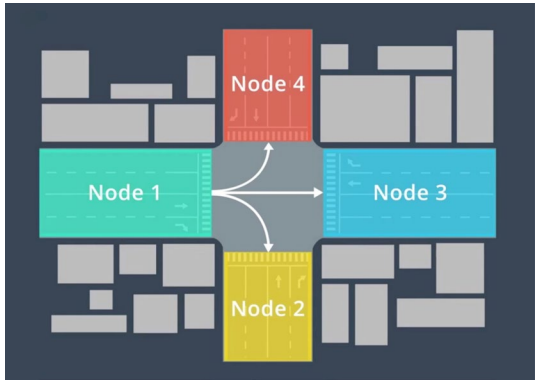
全局路径规划

整体流程



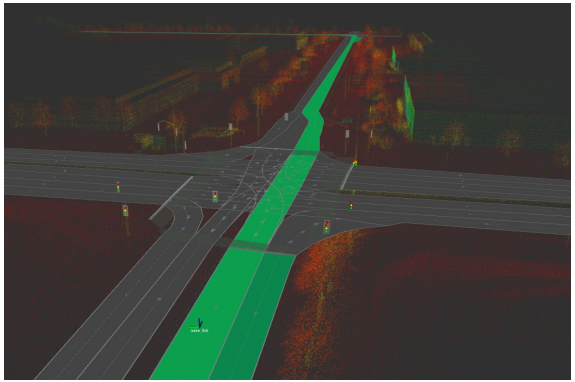
高精地图

面向自动驾驶汽车的一种高精度、精细化定义的地图



拓扑地图

从高精地图抽象出来的拓扑结构，用于全局路径搜索

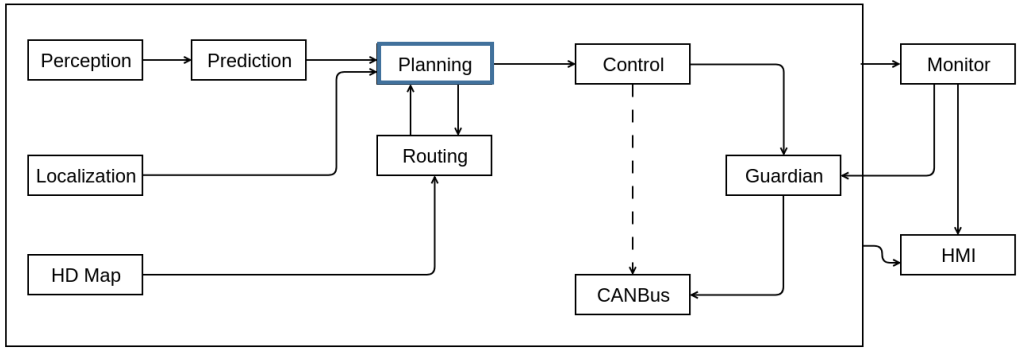


路径搜索

最短路径搜索，支持多个必经点、支持黑名单道路和车道

局部轨迹规划

轨迹规划综述



轨迹规划：根据导航信息以及车辆的当前状态，在有限的时间范围内，计算出一条合适的轨迹供车辆行驶。

输入：感知、预测、定位、高精地图、全局路径规划（Routing）
输出：控制
目标：安全、舒适、高效
避免和障碍物发生碰撞
良好的乘坐体验
在合理的时间内抵达终点/目的地

局部轨迹规划

数学定义

配置空间: $q = (x, y, \theta, \kappa)$

状态: $s = (q, v, a)$

运动规划: $S(t)\colon t \to s = (x, y, \theta, \kappa, v, a), \forall t \in [0, t_{max}]$

s.t.

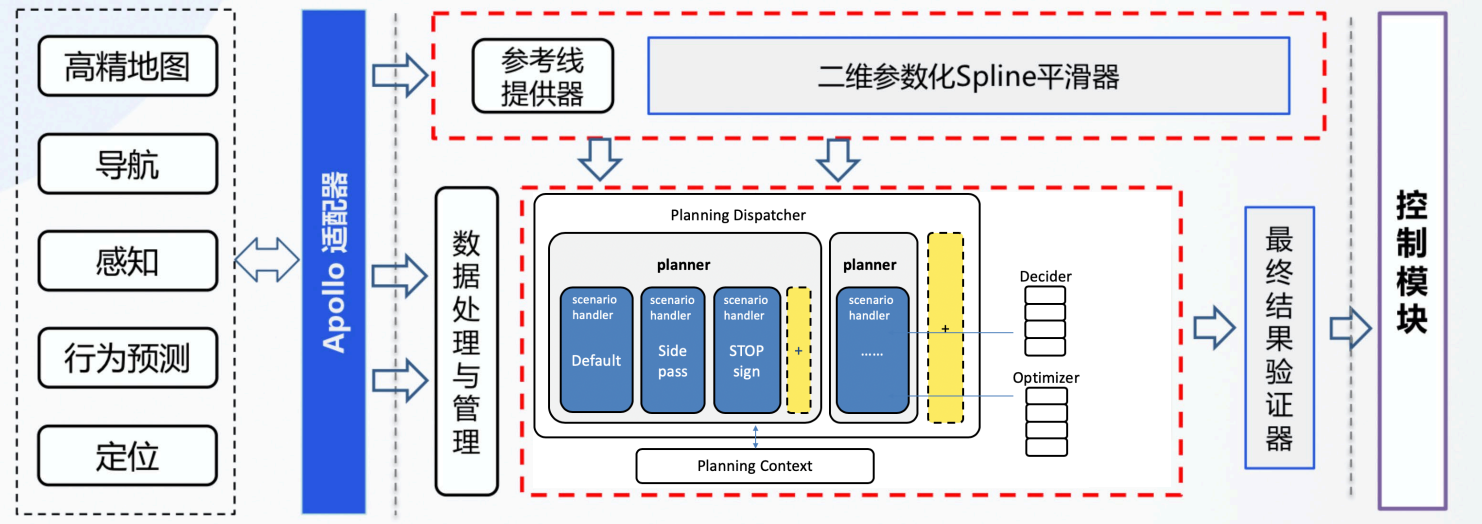
1. 运动学约束
2. 车辆物理限制
3. 碰撞约束

难点:

1. 问题维度高，7 个维度
2. 运动学约束、安全性约束的存在
3. 舒适性要求、加加速度最小、到达时间最小
4. 系统响应速度高：需要计算效率高的算法

局部轨迹规划

Apollo 规划框图



参考线提供器：根据 Routing 的结果和自车状态生成多条局部的、平滑的参考线

Pnc_Map、参考线裁剪和拼接、参考线平滑

决策与规划：根据不同的场景执行不同的决策和优化任务，输出合适的轨迹

场景、决策、搜索与优化、规划器

局部轨迹规划

参考线作用

1. 降低规划的复杂度：用于表达换道的需求，一般换道时会有多条参考线
2. 降低规划的复杂度：减小搜索范围、路径规划和速度规划解耦
3. 降低决策的复杂度：交通规则决策，路径和速度决策(动静态障碍物投影)等基于参考路径

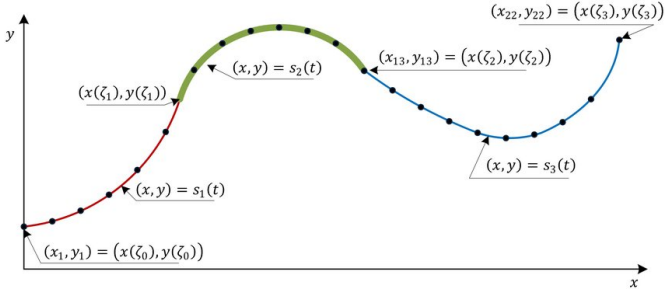
参考线平滑

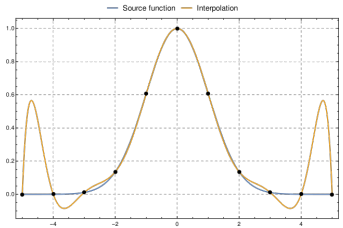
分段多项式曲线

$$x = f_i(s) = a_{i0} + a_{i1}s + a_{i2}s^2 + a_{i3}s^3 + a_{i4}s^4 + a_{i5}s^5$$
$$y = g_i(s) = b_{i0} + b_{i1}s + b_{i2}s^2 + b_{i3}s^3 + b_{i4}s^4 + b_{i5}s^5$$

分段螺旋曲线

$$\theta(s) = a + bs + cs^2 + ds^3 + es^4 + fs^5$$
$$x(s) = \int_0^s \cos(\theta(s))ds$$
$$y(s) = \int_0^s \sin(\theta(s))ds$$





局部轨迹规划

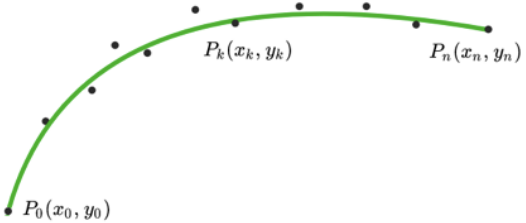
基于离散点的参考线平滑算法

优化变量：$P_k = [x_k, y_k]^T$

目标函数：

$$\min \sum_{k=1}^{n-2} ||2P_k - P_{k-1} - P_{k+1}||_2^2 + \sum_{k=0}^{n-1} ||P_k - P_{k\_ref}||_2^2 + \sum_{k=0}^{n-2} ||P_{k+1} - P_k||_2^2$$

平滑度　　偏移量　　长度



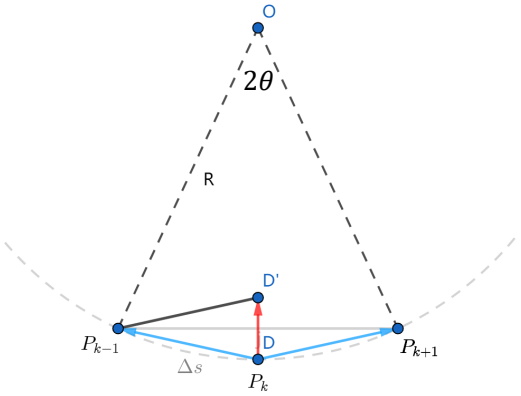
约束：

$$P_{k\_ref} - P_L \le P_k \le P_{k\_ref} + P_U \qquad \text{位置约束}$$

$$||2P_k - P_{k-1} - P_{k+1}||_2^2 \le d_{k\_ref}^2 / R_{min} \qquad \text{曲率约束}$$

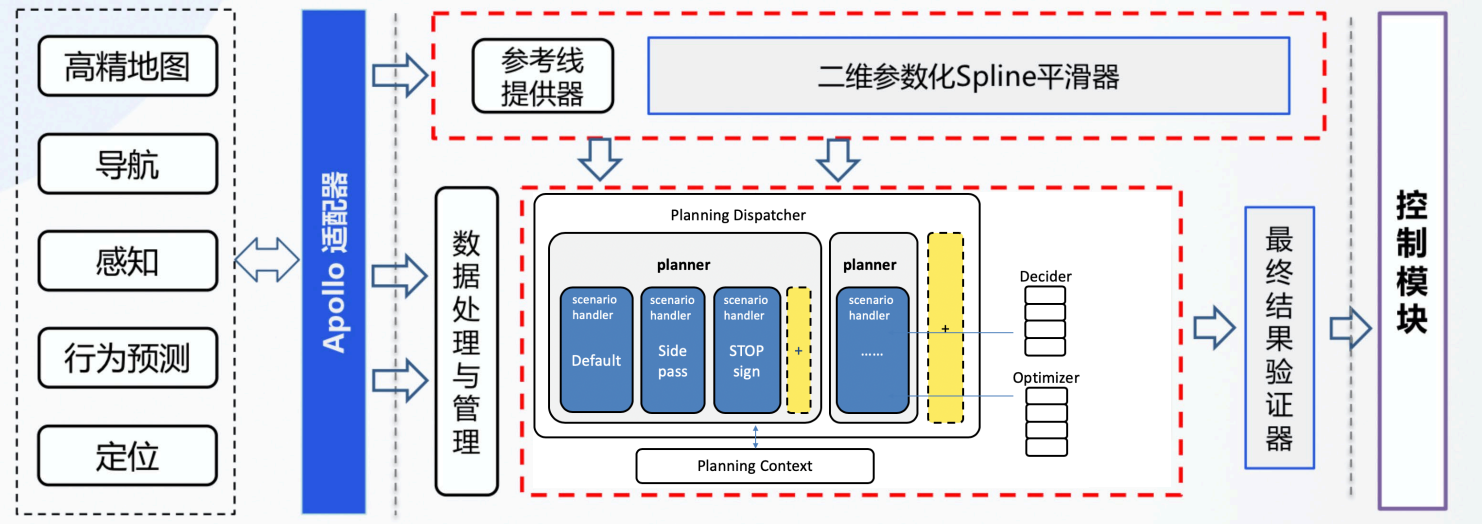
其中包含以下近似：

$$\begin{aligned} ||2P_k - P_{k-1} - P_{k+1}||_2 &\approx 2||P_k - P_{k-1}||_2 \sin\left(\frac{\theta}{2}\right) \\ &\approx ||P_k - P_{k-1}||_2 \theta \\ &\approx ||P_k - P_{k-1}||_2^2 / R \end{aligned}$$



局部轨迹规划

Apollo 规划框图



参考线提供器：根据Routing的结果和自车状态生成多条局部的、平滑的参考线

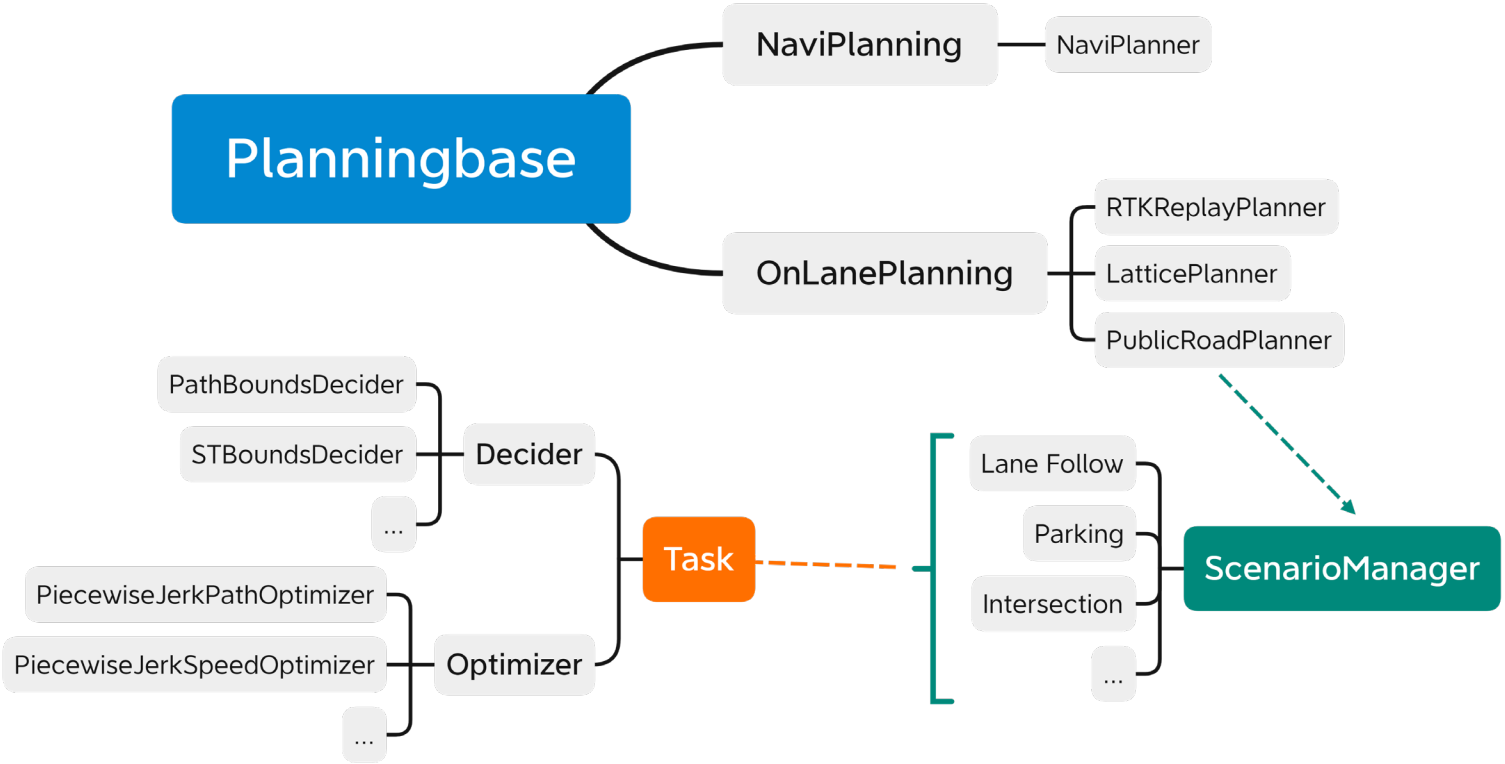
Pnc_Map、参考线裁剪和拼接、参考线平滑

决策与规划：根据不同的场景执行不同的决策和优化任务，输出合适的轨迹

场景、决策、搜索与优化、规划器

局部轨迹规划

Apollo 规划框图

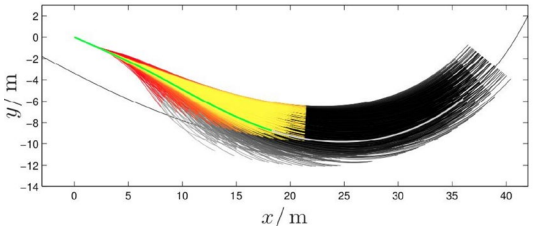


局部轨迹规划

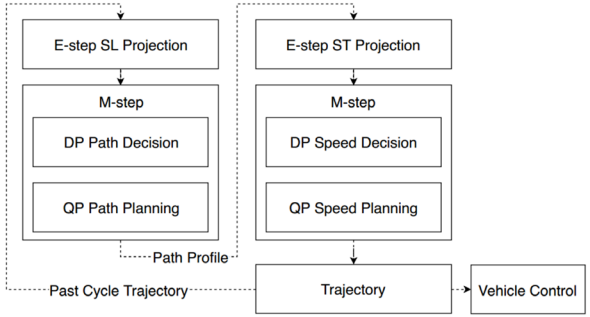
Apollo 规划器

Planner	Description	Introduce
RTK Planner	根据录制的轨迹来规划行车路线	v1.0
EM Planner	基于DP+QP的路径规划和速度规划	v1.5
Lattice Planner	基于状态网格采样的轨迹规划	v2.5
Navi Planner	基于实时相对地图的轨迹规划	v3.0
Open Space Planner	开放空间下的轨迹规划	v3.5
Public Road Planner	开放道路下的轨迹规划	v3.5

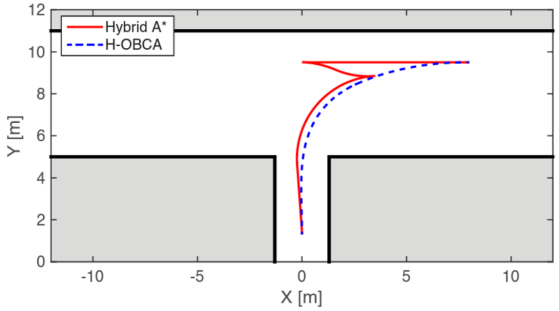
Lattice Planner



EM Planner



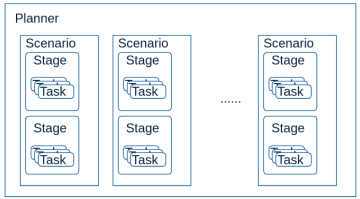
Open Space Planner

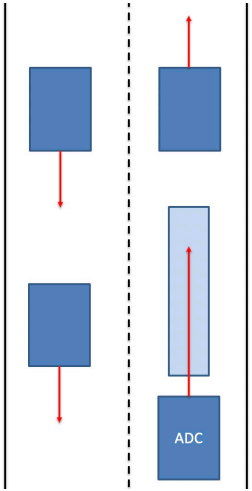


局部轨迹规划

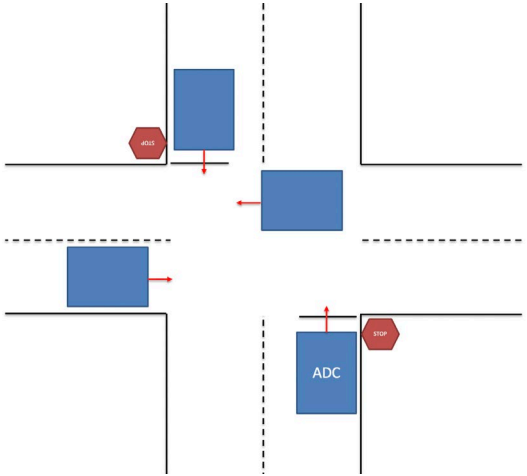
场景和阶段

Apollo 规划是按照场景(Scenario)来执行的，每个场景可分为几个阶段(Stage)，每个阶段会执行一系列的任务(Task)。不同场景间的切换是由状态机 (ScenarioManager)来控制的。

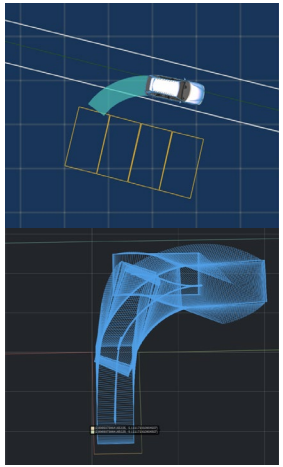




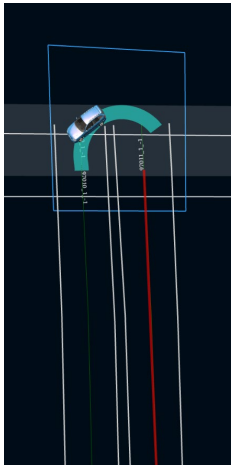
Lane Follow



STOP Sign(Unprotected)



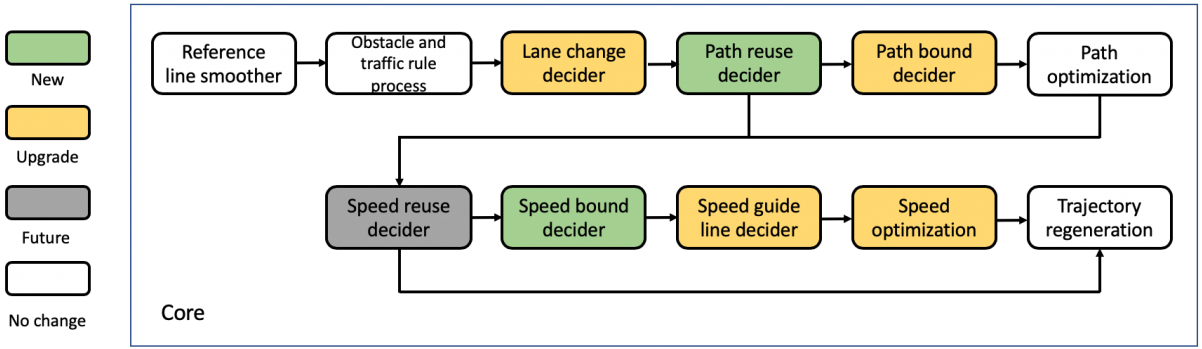
Valet Parking



Dead End

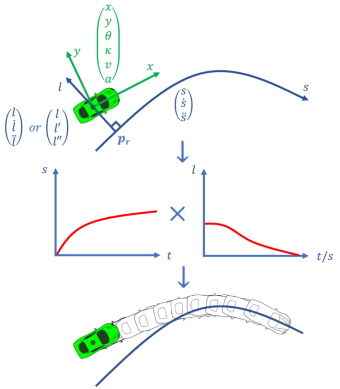
局部轨迹规划

Lane Follow 场景中的任务



降低规划维度

- 路径与速度的分离，先确定路径，再确定沿路径的速度分配
- 路径规划，规划行驶轨迹的几何形状，$s \rightarrow (x, y, \theta, \kappa)$
- 速度规划，规划沿路径的速度分配，$t \rightarrow (s, v, a)$



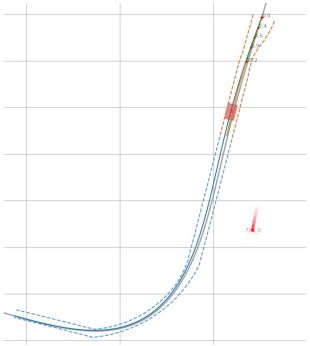
障碍物避让策略

- 路径规划避让静态障碍物
- 速度规划避让动态障碍物

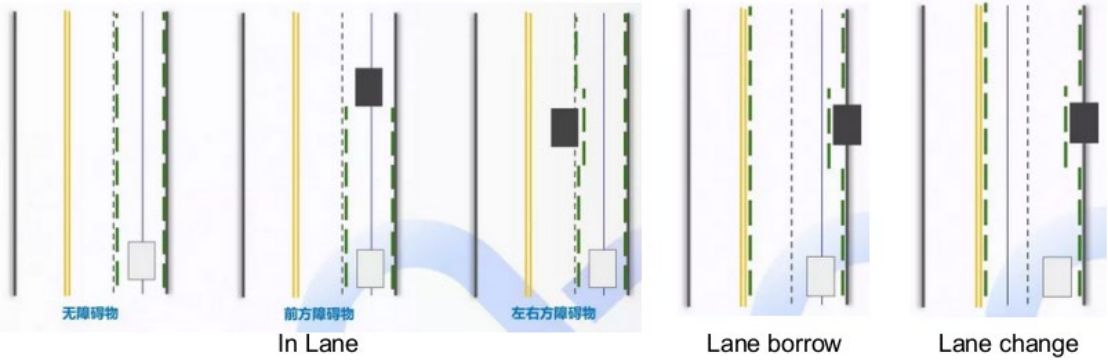
局部轨迹规划

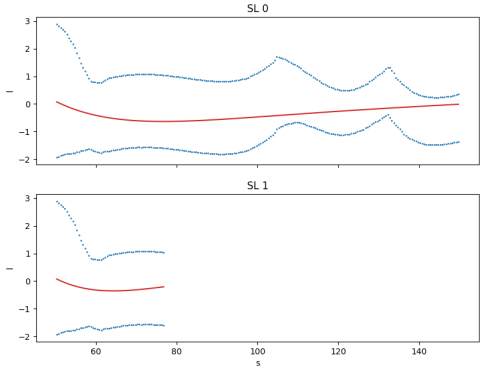
路径规划 Path Planning

1. 路径需要有足够的灵活度，能够避让复杂、拥挤城市环境中的障碍物。
2. 需要做到路径平滑，路径几何特性的平顺变换是舒适性的前提。
3. 需要遵守车辆的运动学限制条件，规划出的路径可执行。
4. 需要考虑车辆的运动状态；低速和高速的运动状态下需要适当的调整车辆的几何形状，以保证横向运动的舒适性。



PATH_BOUNDS_DECIDER

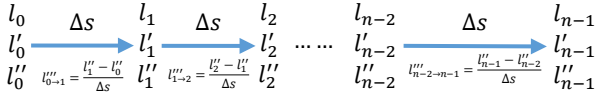




局部轨迹规划

PIECEWIZE_JERK_PATH_OPTIMIZER

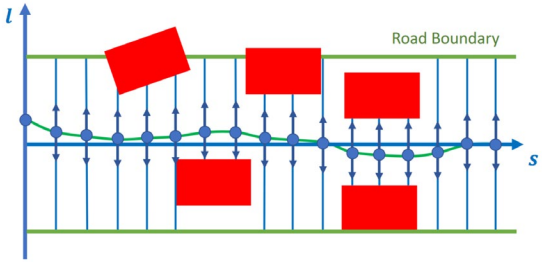
路径参数化：$[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}],\ \mathbf{x}_i = [l, l', l'']$



贴近指引线

目标函数：

$$f = w_1 \sum_{i=0}^{n-1}(l_i - l_i^{ref})^2 + w_2 f_{end}(\mathbf{x}_{n-1}, \mathbf{x}^{end}) + w_3 \sum_{i=0}^{n-1} {l_i}^2 + w_4 \sum_{i=0}^{n-1} l_i'^2 + w_5 \sum_{i=0}^{n-1} l_i''^2 + w_6 \sum_{i=1}^{n-1}(\frac{l_i'' - l_{i-1}''}{\Delta s})^2$$



约束：边界约束、连续性约束和初始状态约束

$$l_i \in [l_{l_i}, l_{u_i}], l_i' \in [l_{l_i}', l_{u_i}'], l_i'' \in [l_{l_i}'', l_{u_i}'']$$

$$l_i''' = (l_{i+1}'' - l_i'')/\Delta s \in [J_{min}, J_{max}]$$

$$l_{i+1}' = l_i' + l_i'' * \Delta s + 0.5 * l_{i\to i+1}''' * \Delta s^2$$

$$l_{i+1} = l_i + l_i' * \Delta s + 0.5 * l_i'' * \Delta s^2 + 1/6 * l_{i\to i+1}''' * \Delta s^3$$

$$l_0 = l_{init}, l_0' = l_{init}', l_0'' = l_{init}''$$

QP

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}x^T P x + q^T x \\ &\text{subject to} && l \le Ax \le u \end{aligned}$$

NLP

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} && f(x) \\ &\text{s.t.} && g^L \le g(x) \le g^U \\ & && x^L \le x \le x^U, \end{aligned}$$

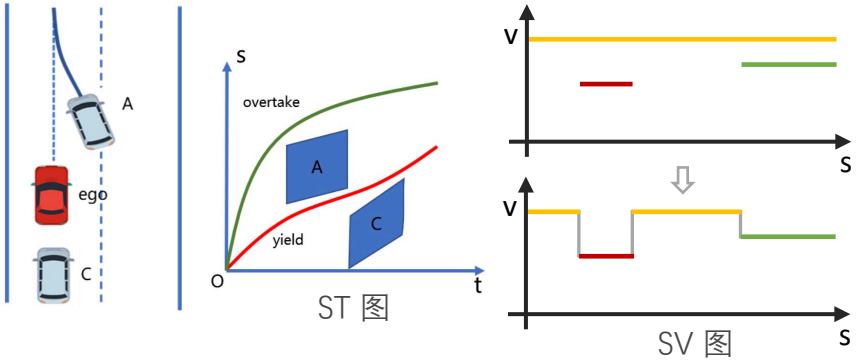
局部轨迹规划

速度规划 Speed Planning

1. 需要有足够的灵活度，能够避让复杂、拥挤城市环境中的障碍物。
2. 速度分配平滑，速度、加速度的平顺变换是舒适性的前提。同时考虑曲率。
3. 需要遵守车辆的运动学限制条件，规划出的轨迹可执行。
4. 需要遵循交通法规，在限速范围内规划。
5. 需要完成到点或到达指定速度的任务，并在保证舒适度的情况下，时间尽可能短。

SPEED_BOUNDS_DECIDER

构造 ST 图和 SV 图

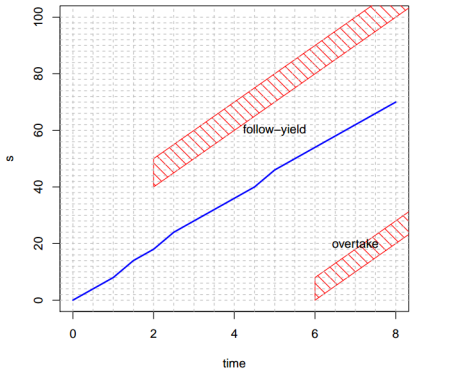


道路限速、周边行人、人行横道、减速带、借道避让时、换道时

SPEED_HEURISTIC_OPTIMIZER

离散化 ST 图，采用动态规划进行搜索

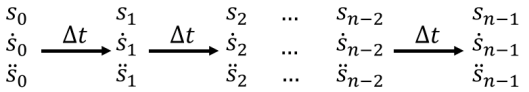
$$cost[t_i, s_j] = \min(cost[t_{i-1}, s_k] + cost_{node}([t_i, s_j]) + cost_{edge}([t_i, s_j], [t_{i-1}, s_k]), cost[t_i, s_j]), k \in \Omega(j)$$



局部轨迹规划

PIECEWISE_JERK_SPEED_OPTIMIZER

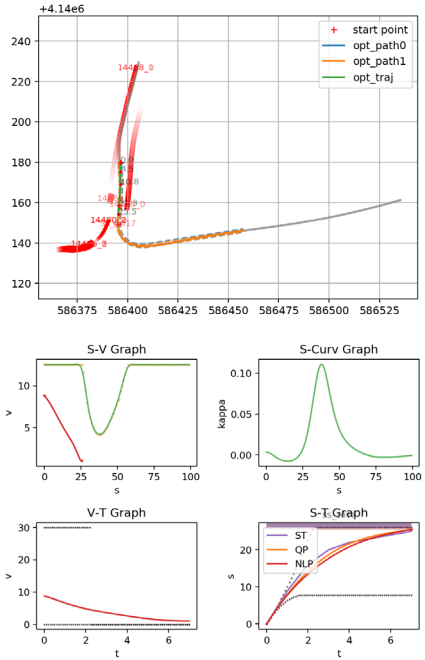
参数化： $[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}],\ \mathbf{x}_i = [s, \dot{s}, \ddot{s}]$



贴近启发路径　贴近启发速度

目标函数：
$$f = w_1 \sum_{i=0}^{n-1}(s_i - s_i^{ref})^2 + w_2 \sum_{i=0}^{n-1}(\dot{s}_i - \dot{s}_i^{ref})^2 + w_3 f_{end}(\mathbf{x}_{n-1}, \mathbf{x}^{end})$$
$$+ w_4 \sum_{i=0}^{n-1}(\ddot{s}_i)^2 + w_5 \sum_{i=1}^{n-1}(\frac{\ddot{s}_i - \ddot{s}_{i-1}}{\Delta t})^2$$

向心加速度尽可能小
$$+ w_{a_c} * \sum_{i=0}^{n-1} \dot{s}_i^2 * \kappa(s_i)$$



约束：边界约束、连续性约束和初始状态约束

$$s_i \in [s_{l_i}, s_{u_i}], \dot{s}_i \in [\dot{s}_{l_i}, \dot{s}_{u_i}], \ddot{s}_i \in [\ddot{s}_{l_i}, \ddot{s}_{u_i}]$$
$$\dddot{s}_i = (\ddot{s}_{i+1} - \ddot{s}_i)/\Delta t \in [J_{min}, J_{max}]$$
$$\dot{s}_{i+1} = \dot{s}_i + \ddot{s}_i * \Delta t + 0.5 * \dddot{s}_{i \to i+1} * \Delta t^2$$
$$s_{i+1} = s_i + \dot{s}_i * \Delta t + 0.5 * \ddot{s}_i * \Delta t^2 + 1/6 * \dddot{s}_{i \to i+1} * \Delta t^3$$
$$s_0 = s_{init}, \dot{s}_0 = \dot{s}_{init}, \ddot{s}_0 = \ddot{s}_{init}$$

$$\dot{s}_i^2 * \kappa(s_i) \in [-a_{c_{max}}, a_{c_{max}}]$$ 向心加速度限制

参考

1. 解析百度Apollo之Routing模块
2. 解析百度Apollo之参考线与轨迹
3. 解析百度Apollo之决策规划模块
4. Apollo公开课丨Apollo运动轨迹规划技术
5. 分享回顾 | Apollo 轨迹规划技术分享
6. 直播回顾 | Apollo决策技术分享

论文	说明
Autonomous Driving Trajectory Optimization with Dual-Loop Iterative Anchoring Path Smoothing and Piecewise-Jerk Speed Optimization	泊车规划算法，用于 Hybrid A* 之后简单的路径和速度的平滑
TDR-OBCA: A Reliable Planner for Autonomous Driving in Free-Space Environment	泊车规划算法
Optimal Vehicle Path Planning Using Quadratic Optimization for Baidu Apollo Open Platform	参考线平滑、路径优化
Optimal Trajectory Generation for Autonomous Vehicles Under Centripetal Acceleration Constraints for In-lane Driving Scenarios	参考线平滑、速度优化
Exploring Imitation Learning for Autonomous Driving with Feedback Synthesizer and Differentiable Rasterization	基于深度学习的轨迹规划
Baidu Apollo EM Motion Planner	

深蓝学院
shenlanxueyuan.com

感谢聆听！

Thanks for Listening