深蓝学院
shenlanxueyuan.com

Magic Lab

路径与轨迹规划

丁文超

大纲

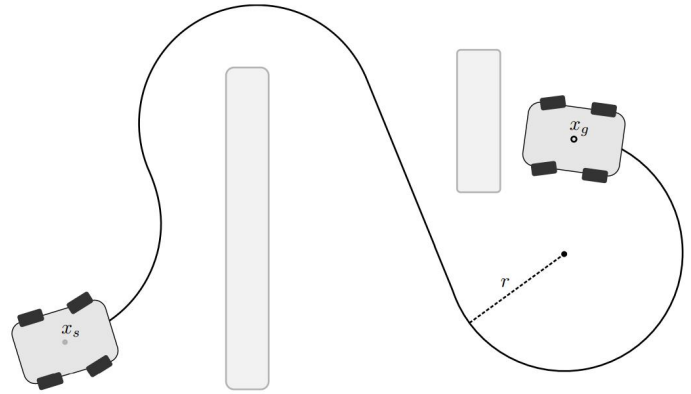
- 基于搜索的路径规划
- 基于采样的路径规划
- 基于优化的路径规划

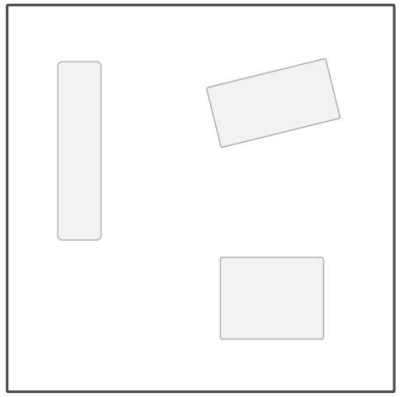
基于搜索的路径规划

A*算法

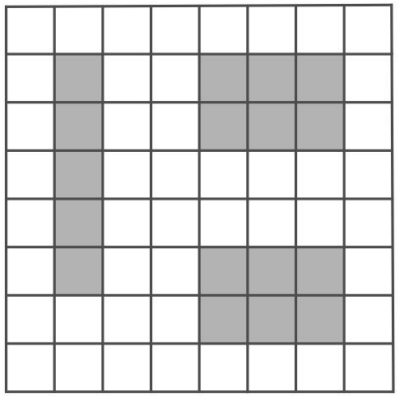
```
make an openlist containing only the starting node
make an empty closed list
while (the destination node has not been reached):
    consider the node with the lowest f score in the open list
    if (this node is our destination node) :
        we are finished
    if not:
        put the current node in the closed list and look at all of its neighbors
        for (each neighbor of the current node):
            if (neighbor has lower g value than current and is in the closed list) :
                replace the neighbor with the new, lower, g value
                current node is now the neighbor's parent
            else if (current g value is lower and this neighbor is in the open list ) :
                replace the neighbor with the new, lower, g value
                change the neighbor's parent to our current node

            else if this neighbor is not in both lists:
                add it to the open list and set its g
```





Continuous World

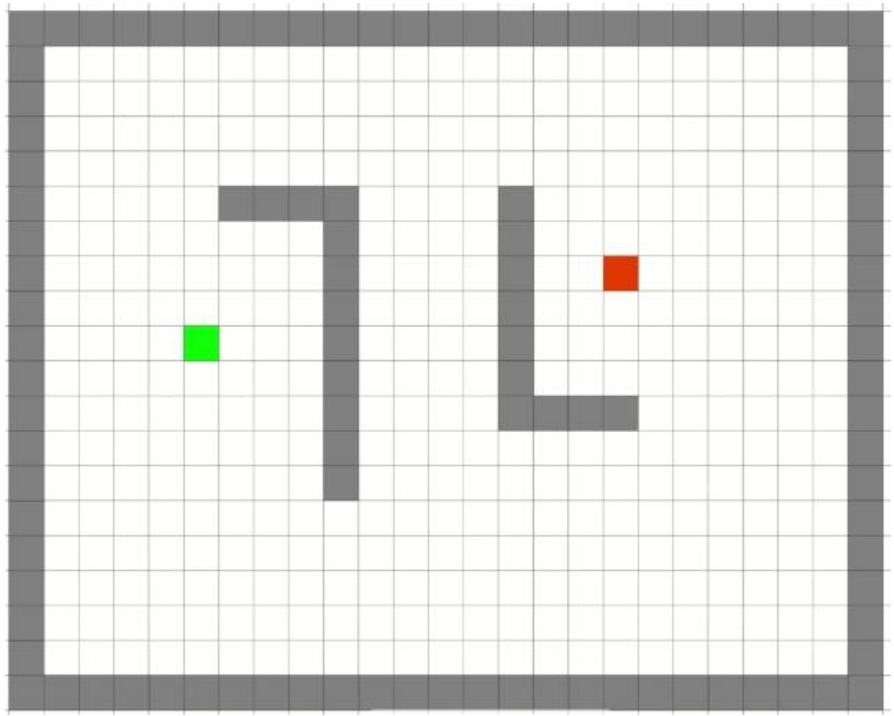


Binary Obstacle Grid

Kurzer K. Path planning in unstructured environments: A real-time hybrid a* implementation for fast and deterministic path generation for the kth research concept vehicle[J]. 2016.

基于搜索的路径规划：A*算法

A*算法



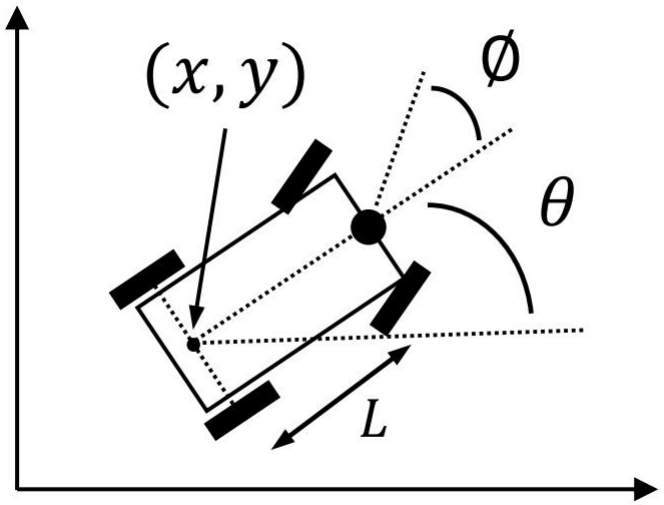
基于搜索的路径规划：A*算法

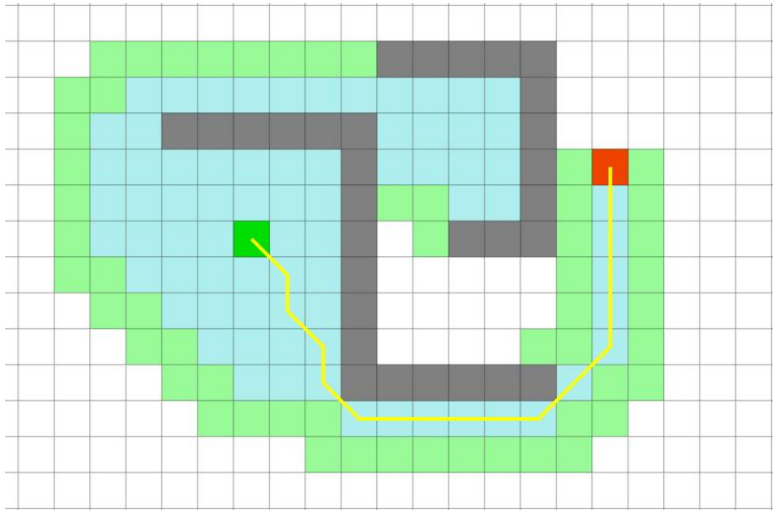
二维格点上进行A*算法缺点：不满足车辆的运动学约束

以后轴中心为原点的车辆运动学模型：

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix}=\begin{pmatrix}v\cos\theta\\ v\sin\theta\\ \frac{v}{L}\tan\phi\end{pmatrix}$$

$$|v|\le v_{max},$$
$$|\phi|\le \phi_{max}<\frac{\pi}{2}$$





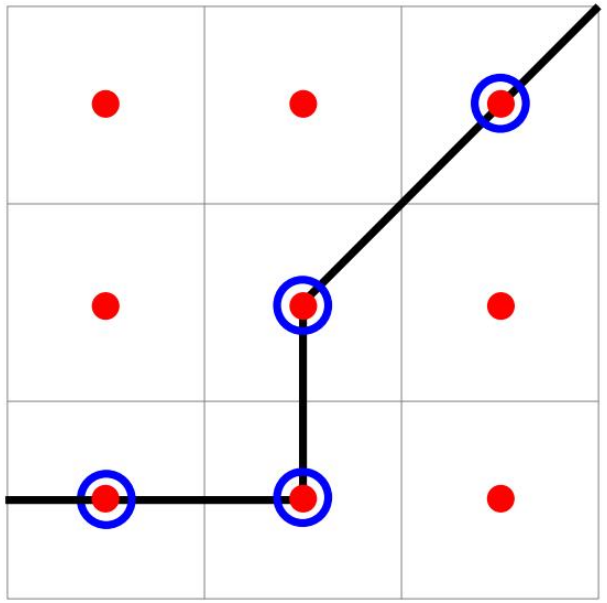
https://qiao.github.io/PathFinding.js/visual/

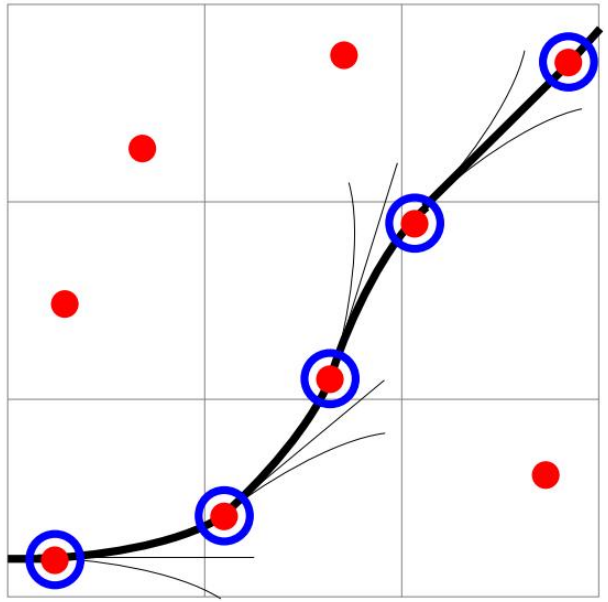
基于搜索的路径规划：Hybrid A*算法

Magic Lab

Hybrid含义:

- 节点的拓展基于车辆运动学模型;
- 代价的计算基于栅格化地图。





Dolgov D, Thrun S, Montemerlo M, et al. Practical search techniques in path planning for autonomous driving[J]. Ann Arbor, 2008, 1001(48105): 18-80.
Dolgov D, Thrun S, Montemerlo M, et al. Path planning for autonomous vehicles in unknown semi-structured environments[J]. The international journal of robotics research, 2010, 29(5): 485-501.

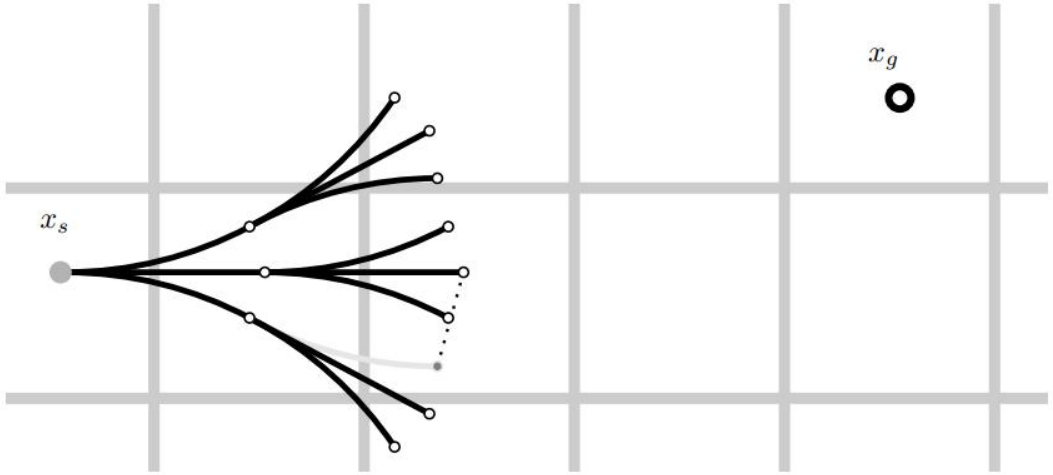
基于搜索的路径规划：Hybrid A*算法

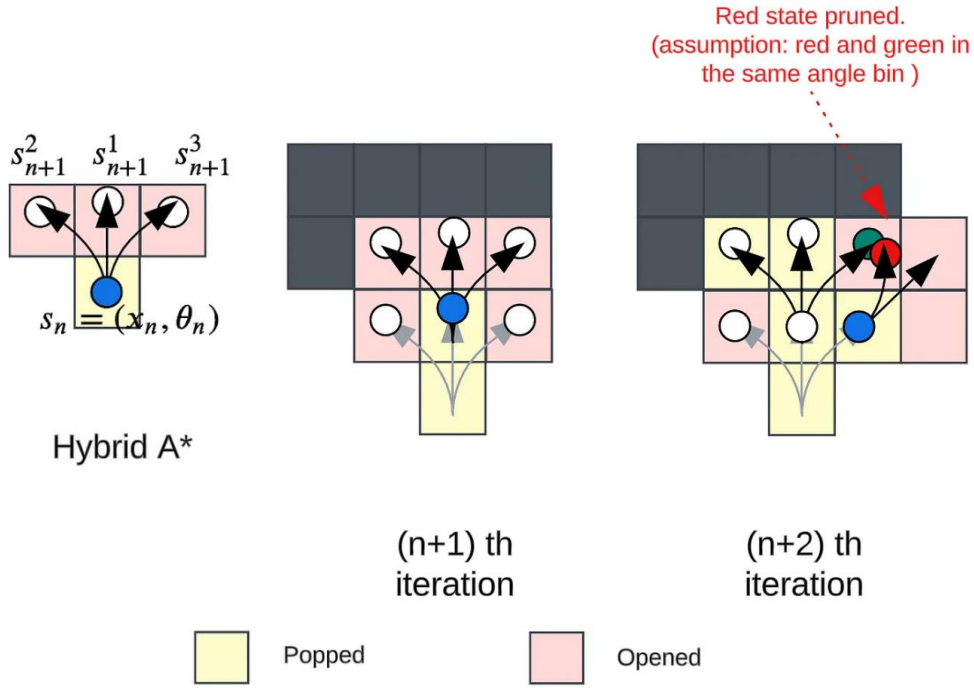
节点的拓展：

- 在与网格精度大致匹配的条件下，使用N种离散的控制动作，比如离散曲率；
- 该路径是一些受车辆转弯半径约束的圆弧和直线；
- **提前终止**：计算从当前状态到达目标状态的*Dubins*和*Reeds Shepp*曲线；

节点的修剪：

- 当多个节点落入同一个*Cell*的时候，进行剪枝，只保留低代价的前继。

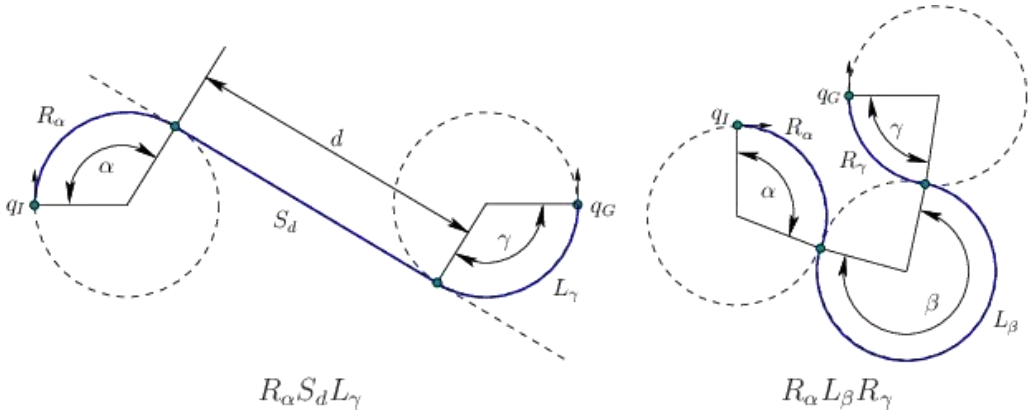




基于搜索的路径规划：Hybrid A*算法

Dubins和Reeds Shepp曲线：

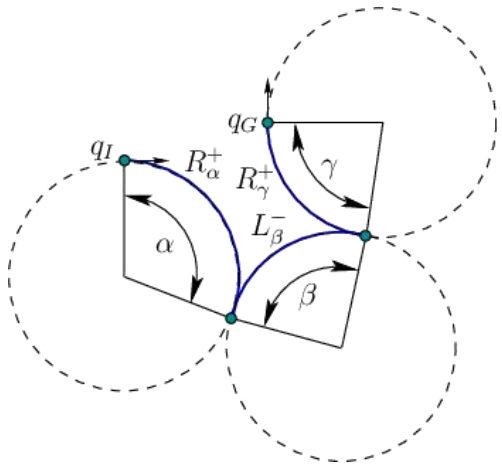
- Dubins曲线是在满足曲率约束和规定的始端和末端的切线（进入方向）的条件下，连接两个二维平面的最短路径，而且限制目标只能向前行进。
- 从起始点到终止点的最短路径始终可以表示为不超过三个运动基元（直行S、左转L、右转R）的组合。最优的Dubins曲线为集合D={LSL, RSR, RSL, LSR, RLR, LRL}六种中的一种。

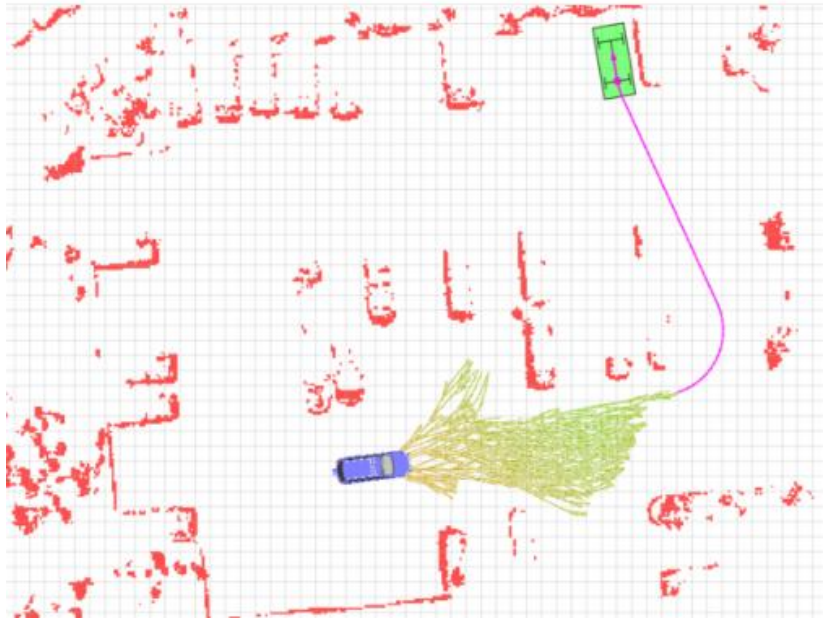


基于搜索的路径规划：Hybrid A*算法

Dubins和Reeds Shepp曲线：

- Reeds-Shepp曲线，与Dubins曲线的唯一区别是允许反向行进，目标既可以前进，又可以倒退；
- 从起始点到终止点的最短路径始终可以表示为不超过六个运动基元（直行S+ S-、左转L+ L-、右转R+ R-）的组合；
- 与Dubins的RLR曲线相比，Reeds-Shepp的R+L-R+曲线更短。





Base word	Sequences of motion primitives
$C\|C\|C$	$(L^+R^-L^+)(L^-R^+L^-)(R^+L^-R^+)(R^-L^+R^-)$
$CC\|C$	$(L^+R^+L^-)(L^-R^-L^+)(R^+L^+R^-)(R^-L^-R^+)$
$C\|CC$	$(L^+R^-L^-)(L^-R^+L^+)(R^+L^-R^-)(R^-L^+R^+)$
CSC	$(L^+S^+L^+)(L^-S^-L^-)(R^+S^+R^+)(R^-S^-R^-)$ $(L^+S^+R^+)(L^-S^-R^-)(R^+S^+L^+)(R^-S^-L^-)$
$CC_\beta\|C_\beta C$	$(L^+R_\beta^+L_\beta^-R^-)(L^-R_\beta^-L_\beta^+R^+)(R^+L_\beta^+R_\beta^-L^-)(R^-L_\beta^-R_\beta^+L^+)$
$C\|C_\beta C_\beta\|C$	$(L^+R_\beta^-L_\beta^-R^+)(L^-R_\beta^+L_\beta^+R^-)(R^+L_\beta^-R_\beta^-L^+)(R^-L_\beta^+R_\beta^+L^-)$
$C\|C_{\pi/2}SC$	$(L^+R_{\pi/2}^-S^-R^-)(L^-R_{\pi/2}^+S^+R^+)(R^+L_{\pi/2}^-S^-L^-)(R^-L_{\pi/2}^+S^+L^+)$ $(L^+R_{\pi/2}^-S^-L^-)(L^-R_{\pi/2}^+S^+L^+)\ (R^+L_{\pi/2}^-S^-R^-)(R^-L_{\pi/2}^+S^+R^+)$
$CSC_{\pi/2}\|C$	$(L^+S^+L_{\pi/2}^+R^-)(L^-S^-L_{\pi/2}^-R^+)(R^+S^+R_{\pi/2}^+L^-)(R^-S^-R_{\pi/2}^-L^+)$ $(R^+S^+L_{\pi/2}^+R^-)(R^-S^-L_{\pi/2}^-R^+)(L^+S^+R_{\pi/2}^+L^-)(L^-S^-R_{\pi/2}^-L^+)$
$C\|C_{\pi/2}SC_{\pi/2}\|C$	$(L^+R_{\pi/2}^-S^-L_{\pi/2}^-R^+)(L^-R_{\pi/2}^+S^+L_{\pi/2}^+R^-)$ $(R^+L_{\pi/2}^-S^-R_{\pi/2}^-L^+)(R^-L_{\pi/2}^+S^+R_{\pi/2}^+L^-)$

共48种Reeds-Shepp曲线

基于搜索的路径规划：Hybrid A*算法

整体流程伪代码:

Hybrid A-star Path Planner Algorithm with Two Waypoints

```
1 algorithm HybridAstarPlanner(M, Veh, qS, qG1, qG2)
2       n_s ← (qS, 0, h(qS, qG1, qG2), 1)
3       O ← {n_s}
4       C = ∅
5       while O ≠ ∅ do
6           n ← node with the minimum f value in O
7           O ← O \ {n}
8           C ← C ∪ {n}
9           if n_s = 1 then
10              if n_x ∈ qG1 then
11                  UpdateSeccessors(M, Veh, O, C, n, 2)
12              else
13                  UpdateSeccessors (M, Veh, O, C, n, 1)
14              end if
15          else if n_s = 2 then
16              if n_x ∈ qG2 then
17                  return planned path
18              else
19                  UpdateSeccessors (M, Veh, O, C, n, 2)
20              end if
21          end if
22      end while
23      return no path
24 end
25 function UpdateSeccessors (M, Veh, O, C, n, s)
26      for all defined steering angles δ do
27          n' ← succeeding state of n using Veh (n_θ, δ)
28          n'_s ← s
29          if n' ∉ C then
30              if M(n' ) = obstacle then
31                  C ← C ∪ {n'}
32              else if ∃ n ∈ O : n_x = n'_x ∧ n_s = n'_s then
33                  compute the optimum node cost
34              else
35                  O ← O ∪ {n'}
36              end if
37          end if
38      end for
39 end
```

基于搜索的路径规划：Hybrid A*算法

代价计算

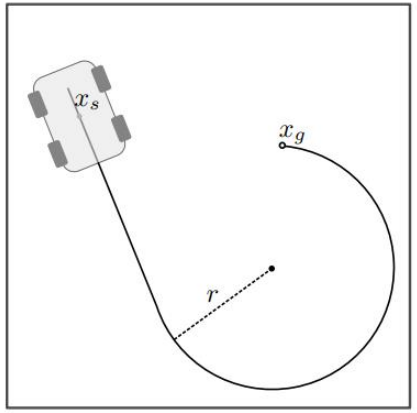
代价包括节点遍历代价和两个启发函数，即$F(n) = g(n) + h_1(n) + h_2(n)$。

其中，

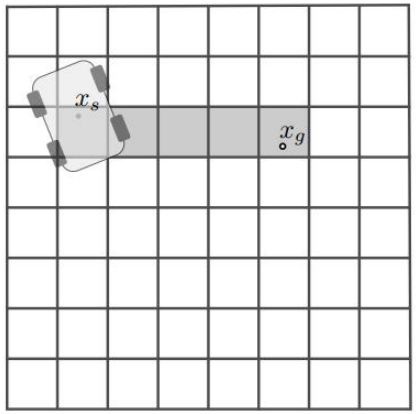
$g(n)$主要考虑路径长度、运动学约束、方向变换的成本；

$h_1(n)$只考虑车辆的运动学约束而不考虑障碍物；

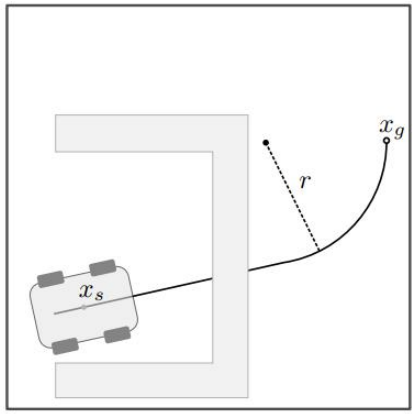
$h_2(n)$只考虑障碍物信息而不考虑车辆的运动学约束。



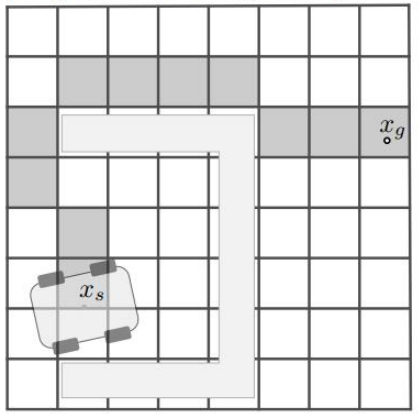
(a) The constrained heuristic accounting for the goal heading.



(b) The unconstrained heuristic heavily underestimating the cost of the path, due to the wrong heading.



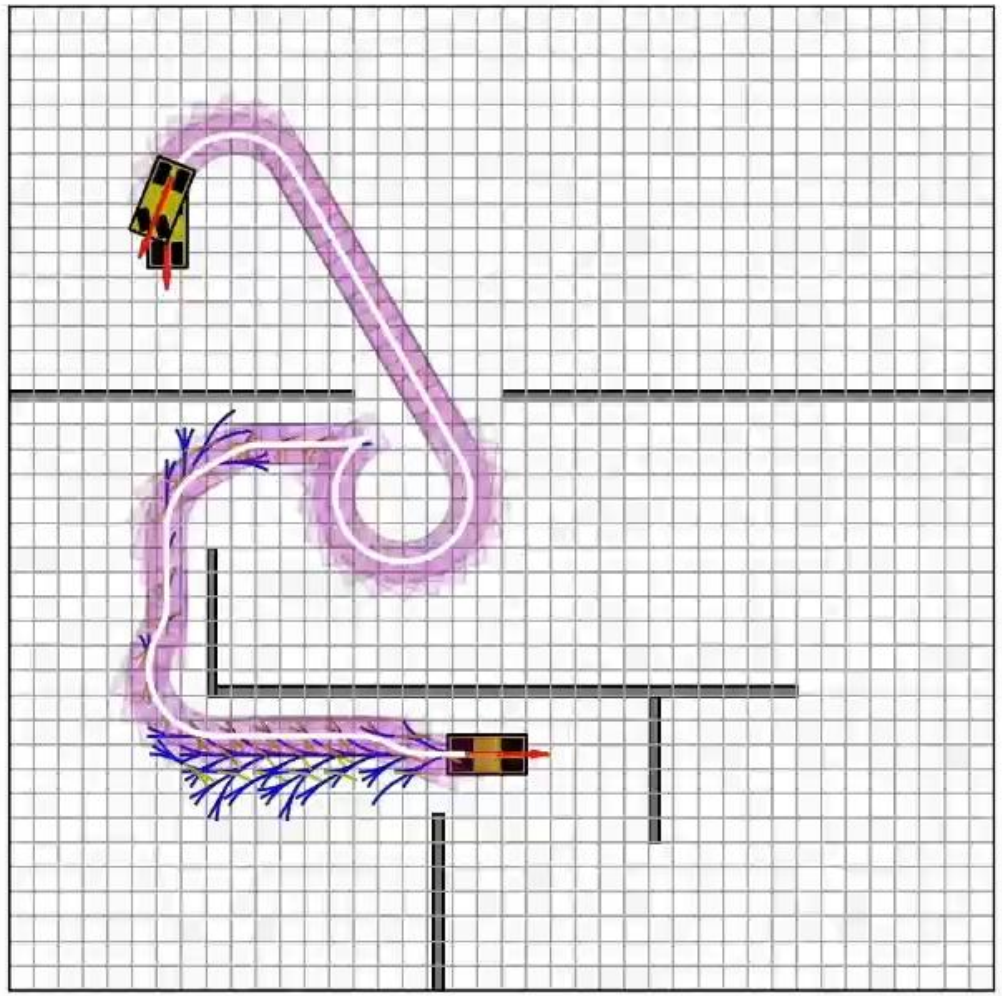
(c) The constrained heuristic heavily underestimating the cost of the path, due to ignoring obstacles.



(d) The unconstrained heuristic accounting for obstacles and dead ends.

基于搜索的路径规划：Hybrid A*算法

实例



大纲

- 基于搜索的路径规划
- 基于采样的路径规划
- 基于优化的路径规划

基于采样的路径规划

Frenét Frame方法

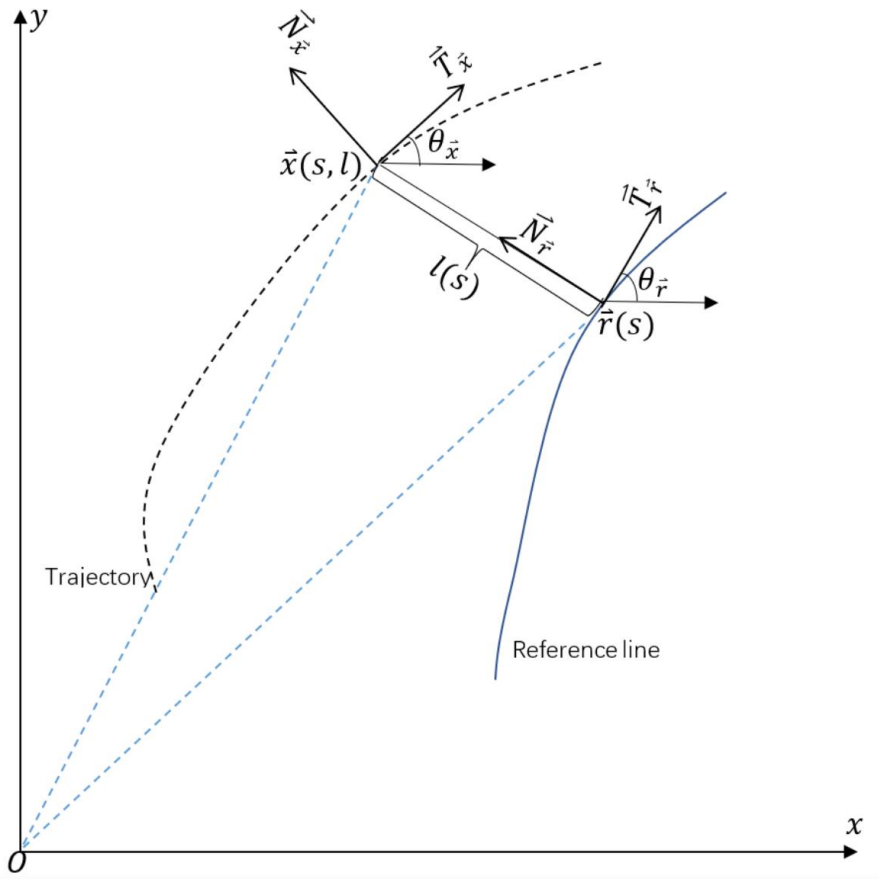
Frenét坐标系基于自车横向以及纵向行为建立，更符合车辆运动特点。

- Cartesian坐标系下，车辆运动：$x, y, \theta_x, k_x, v, a$，

  分别表示横纵坐标、航向角、曲率、速度、加速度;

- Frenét坐标系（S-L坐标系）下，车辆运动：$s, \dot{s}, \ddot{s}, l, l', l''$，

  分别表示纵向位移、纵向速度、纵向加速度、横向位移、横向速度、横向加速度。



基于采样的路径规划

Cartesian →Frenét 1D

(1) 求解投影点，即找到参考线上距离自车质心最近的参考点。

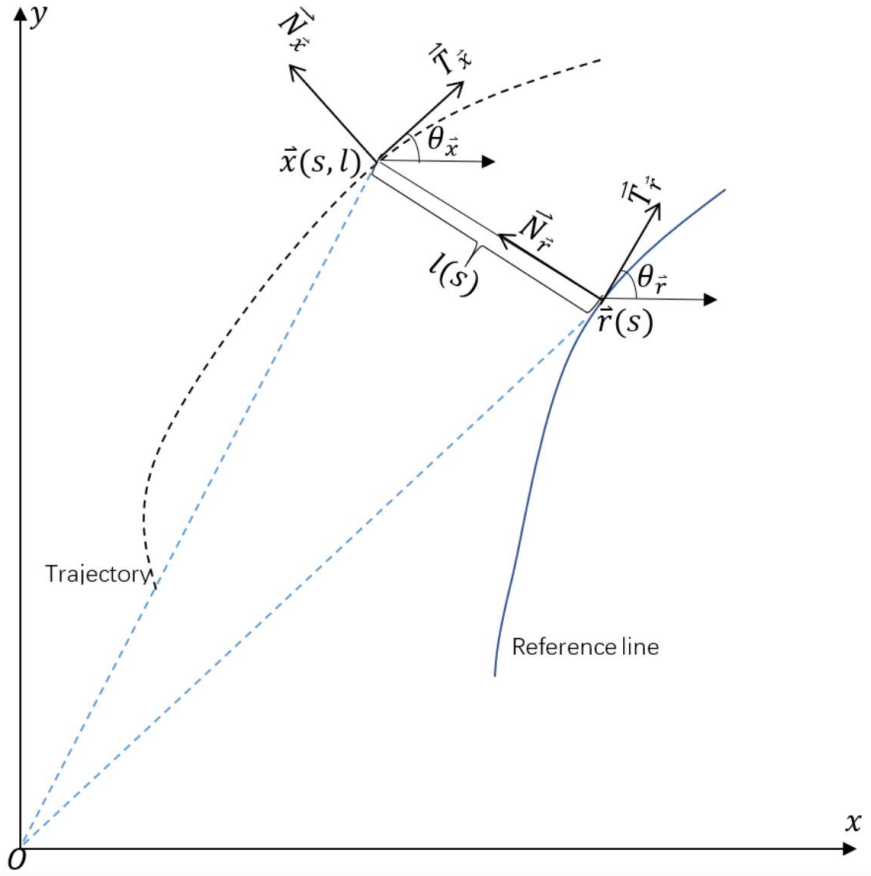
(2) 求纵向坐标： $s = \sum s_r(t)$

(3) 求横向坐标 l ， 如图可知$\vec{l} = (x_c - x_r, y_c - y_r)$,其中自车质心和投影点在笛卡尔坐标下的向量分别为$\vec{x} = (x_c, y_c)$和$\vec{r} = (x_r, y_r)$.

根据三角形向量的关系，可得： $\vec{x} = \vec{r} + l \cdot \overrightarrow{n_r}$.

两边同时乘以$\overrightarrow{N_r}^T$，可得 $\vec{x} \cdot \overrightarrow{N_r}^T = \vec{r} \cdot \overrightarrow{N_r}^T + l \cdot \overrightarrow{N_r} \cdot \overrightarrow{N_r}^T$， 通过简化后得： $l = (\vec{x} - \vec{r}) \cdot \overrightarrow{N_r}^T$

代入$\overrightarrow{N_r}$，得出$l = sign\big((y_c - y_r)\cos(\theta_r) - (x_c - x_r)\sin(\theta_r)\big)\sqrt{(x_c - x_r)^2 + (y_c - y_r)^2}$



基于采样的路径规划

Magic Lab

Frenét → Cartesian 1D

$$\begin{cases} x = x_r - \sin\theta_r \times l \\ y = y_r + \cos\theta_r \times l \end{cases}$$

```python
def frenet_to_cartesian1D(rs, rx, ry, rtheta, s_condition, d_condition):
    if fabs(rs - s_condition[0])>= 1.0e-6:
        print("The reference point s and s_condition[0] don't match")

    cos_theta_r = cos(rtheta)
    sin_theta_r = sin(rtheta)

    x = rx - sin_theta_r * d_condition[0]
    y = ry + cos_theta_r * d_condition[0]

    return x, y
```

基于采样的路径规划

Cartesian →Frenét 3D

$x, y, \theta_x, k_x, v, a \longrightarrow s, l, \dot{s}, \ddot{s}, l', l''$

$$
\begin{cases}
s = & s_r \\
\dot{s} = & \dfrac{v_x cos(\theta_x - \theta_r)}{1 - k_r l} \\
\ddot{s} = & \dfrac{a_x cos(\theta_x - \theta_r) - \dot{s}^2 \left[l' \left(k_x \dfrac{1 - k_r l}{cos(\theta_x - \theta_r)} - k_r \right) - (k_r' l + k_r l') \right]}{1 - k_r l} \\
l = & sign\big((y_x - y_r) cos(\theta_r) - (x_x - x_r) sin(\theta_r)\big) \sqrt{(x_x - x_r)^2 + (y_x - y_r)^2} \\
l' = & (1 - k_r l) tan(\theta_x - \theta_r) \\
l'' = & -(k_r' l + k_r l') tan(\theta_x - \theta_r) + \dfrac{(1 - k_r l)}{cos^2(\theta_x - \theta_r)} \left(\dfrac{1 - k_r l}{cos(\theta_x - \theta_r)} k_x - k_r \right)
\end{cases}
$$

基于采样的路径规划

Frenét → Cartesian 3D

$s, l, \dot{s}, \ddot{s}, l', l'' \longrightarrow x, y, \theta_x, k_x, v, a$

$$
\begin{cases}
x_x = x_r - l\sin(\theta_r) \\
y_x = y_r + l\cos(\theta_r) \\
\theta_x = \arctan\left(\dfrac{l'}{1-k_r l}\right) + \theta_r \in [-\pi, \pi] \\
v_x = \sqrt{[\dot{s}(1-k_r l)]^2 + (\dot{s}l')^2} \\
a_x = \ddot{s}\dfrac{1-k_r l}{\cos(\theta_x-\theta_r)} + \dfrac{\dot{s}^2}{\cos(\theta_x-\theta_r)}\left[l'\left(k_x\dfrac{1-k_r l}{\cos(\theta_x-\theta_r)} - k_r\right) - (k_r' l + k_r l')\right] \\
k_x = \left((l'' + (k_r' l + k_r l')\tan(\theta_x-\theta_r)\right)\dfrac{\cos^2(\theta_x-\theta_r)}{1-k_r l} + k_r\dfrac{\cos(\theta_x-\theta_r)}{1-k_r l}
\end{cases}
$$

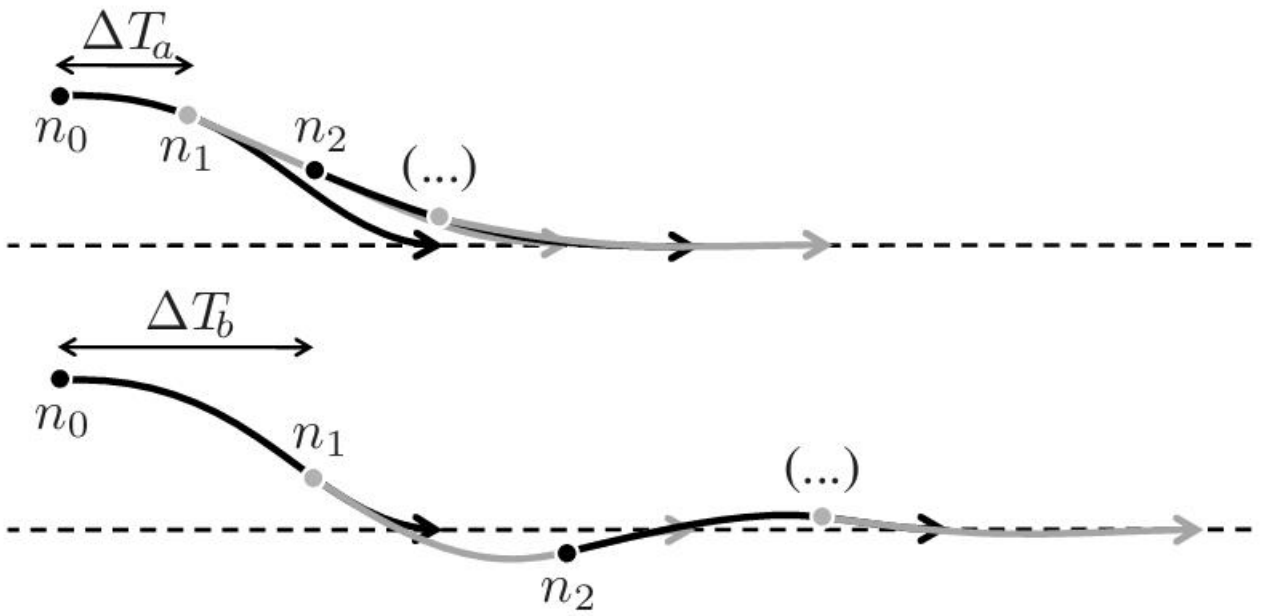
基于采样的路径规划

贝尔曼Bellman最优性原理：

对于每个$t = 0,1,2, \ldots$

$V(t, k_t) = max_{c_t}[f(t, k_t, c_t) + V(t+1, g(t, k_t, c_t))]$。

在每个规划步骤中，都遵循先前计算出的轨迹的剩余部分，提供时间一致性。



基于采样的路径规划

符合Bellman最优性 → 解决最优控制问题

问题定义：

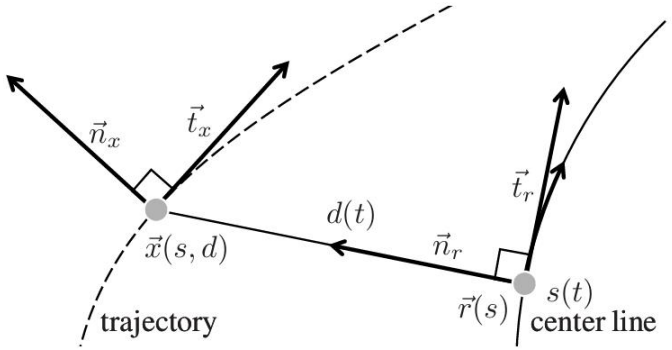
- 给定t_0时刻的起始状态$P_0 = [p_0, \dot{p}_0, \ddot{p}_0]$，和$t_1 = t_0 + T$时刻的终止状态$P_1 = [p_1, \dot{p}_1, \ddot{p}_1]$

- 假设求解目标是，最小化jerk平方的积分：

$$J_t(p(t)) := \int_{t_0}^{t_1} \dddot{p}^2(\tau) d\tau$$

- 该最优控制问题的最优解是五次多项式。

定理：给定t_0时刻的起始状态$P_0 = [p_0, \dot{p}_0, \ddot{p}_0]$，和$t_1 = t_0 + T$时刻的终止状态$P_1$中的$\dot{p}_1, \ddot{p}_1$，给定任意函数$g$和$h$，且$k_j, k_t, k_p > 0$，最小化代价函数$C = k_j J_t + k_t g(T) + k_p h(p_1)$的解是五次多项式问题。（反证法）



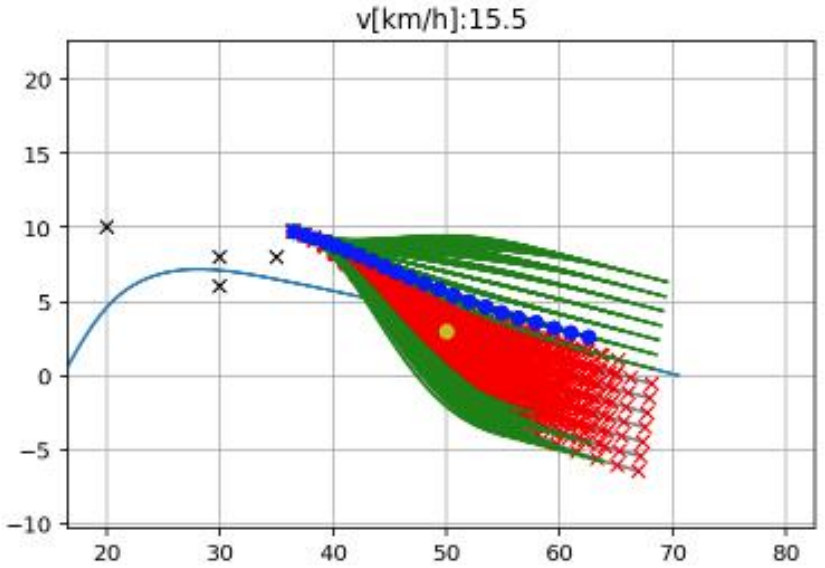
基于采样的路径规划

高速轨迹采样——横向

高速——横向运动相比于纵向运动相对幅度非常小，近似横纵向解耦，横纵向可独立计算

高速情况轨迹规划问题转化为曲线（五次多项式）拟合问题：

- 初始状态：$D_0 = [d_0, \dot{d}_0, \ddot{d}_0]$
- 终止状态：$D_1 = [d_1, \dot{d}_1 = 0, \ddot{d}_1 = 0]$ （终止状态可以根据采样策略灵活确定）
- 时间采样：在参考时间T附近离散采样
- 代价函数：$C = k_j J_t + k_t g(T) + k_p h(p_1) \rightarrow C_d = k_j J_t(d(t)) + k_t T + k_d d_1^2$
- 求解最优d_1的问题，根据前面最优控制的性质，可以转换为在空间内采样d_1再择优



基于采样的路径规划

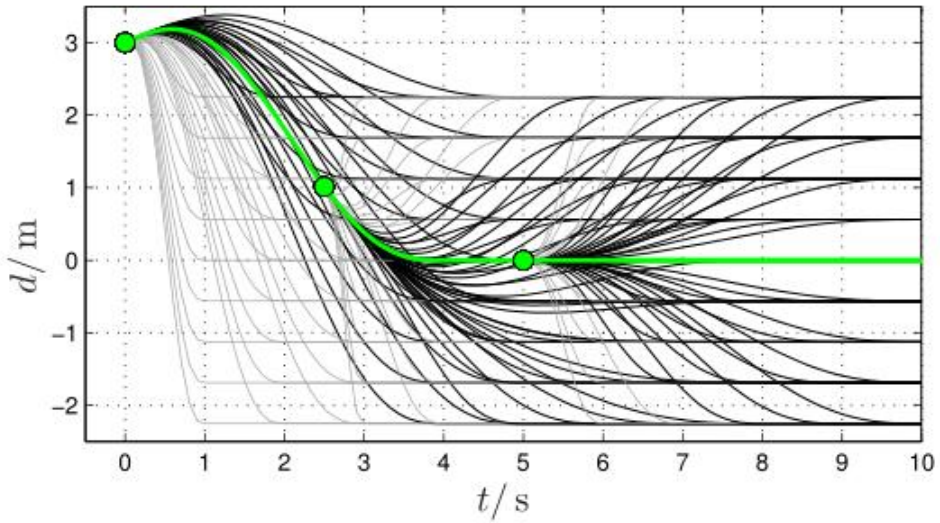
高速轨迹采样——横向

高速——横向运动相比于纵向运动相对幅度非常小，近似横纵向解耦，横纵向可独立计算

计算方式：

- 通过结合不同的终止状态d_i和T_j生成轨迹库$[d_1, \dot{d}_1, \ddot{d}_1, T]_{ij} = [d_i, 0, 0, T_j]$;
- 选出最低代价的有效轨迹;

下图展示了Bellman最优性的实际表现。

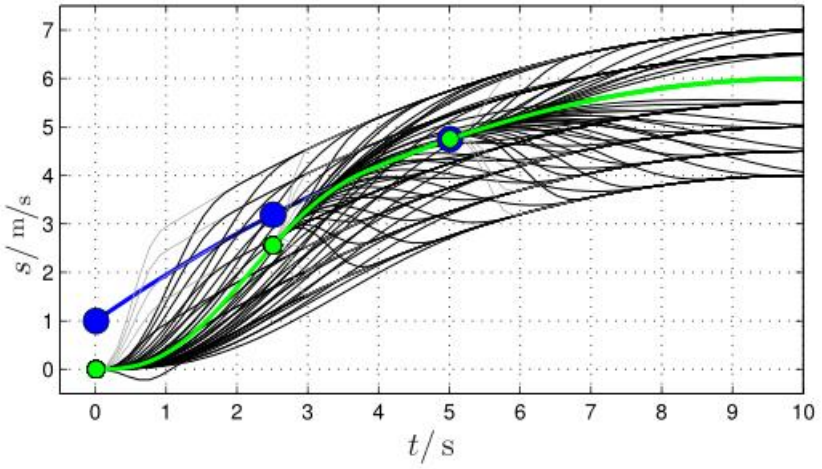


基于采样的路径规划

高速轨迹采样——纵向

前方有车：

- 初始状态：$S_0 = [s_0, \dot{s_0}, \ddot{s_0}]$
- 由不同的Δs_i和T_j生成终止轨迹集合：$[s_1, \dot{s}_1, \ddot{s}_1, T]_{ij} = [[s_{\text{target}}(T_j) + \Delta s_i], \dot{s}_{\text{target}}(T_j), \ddot{s}_{\text{target}}(T_j), T_j]$
- 代价函数：$C_t = k_j J_t + k_t T + k_s [s_1 - s_d]^2$



基于采样的路径规划

高速轨迹采样——纵向

跟车：

为保证安全距离 $s_{target}(t) := s_{lv}(t) - [D_0 + \tau \dot{s}_{lv}(t)]$,

假设前车加速度 $\ddot{s}_{lv}(t) = \ddot{s}_{lv}(t_0) = const$,

在时间上积分可得 $\dot{s}_{lv}(t) = \dot{s}_{lv}(t_0) + \ddot{s}_{lv}(t_0)[t - t_0]$，$s_{lv}(t) = s_{lv}(t_0) + \dot{s}_{lv}(t_0)[t - t_0] + \frac{1}{2}\ddot{s}_{lv}(t_0)[t - t_0]^2$,

那么有$\dot{s}_{target}(t) = \dot{s}_{lv}(t) - \tau \ddot{s}_{lv}(t)$，$\ddot{s}_{target}(t) = \ddot{s}_{lv}(t_1) - \tau \dddot{s}_{lv}(t) = \ddot{s}_{lv}(t_1)$。

变道：

$s_{target}(t) = \frac{1}{2}[s_a(t) + s_b(t)]$。

停车：

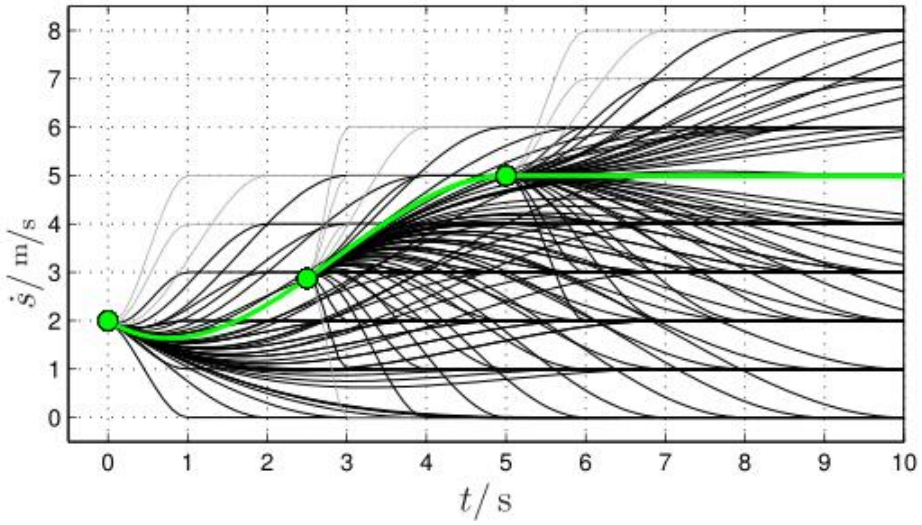
$s_{target} = s_{stop}$，$\dot{s}_{target} \equiv 0$，且$\ddot{s}_{target} \equiv 0$。

基于采样的路径规划

高速轨迹采样——纵向

前方无车：

- 无需达到 s_{target}，但要保证$\dot{s}_d = const$
- 初始状态：$S_0 = [s_0, \dot{s}_0, \ddot{s}_0]$
- 通过改变$\Delta\dot{s}_i$和T_j生成终止轨迹集合：$[\dot{s}_1, \ddot{s}_1, T]_{ij} = [[\dot{s}_d + \Delta\dot{s}_i], 0, T_j]$
- 代价函数：$C_v = k_j J_t(s(t)) + k_t T + k_{\dot{s}}[\dot{s}_1 - \dot{s}_d]^2$



基于采样的路径规划

低速轨迹规划——横向

低速——横向运动和纵向运动可比，而车辆有运动学约束，横纵向解耦会忽略运动学约束

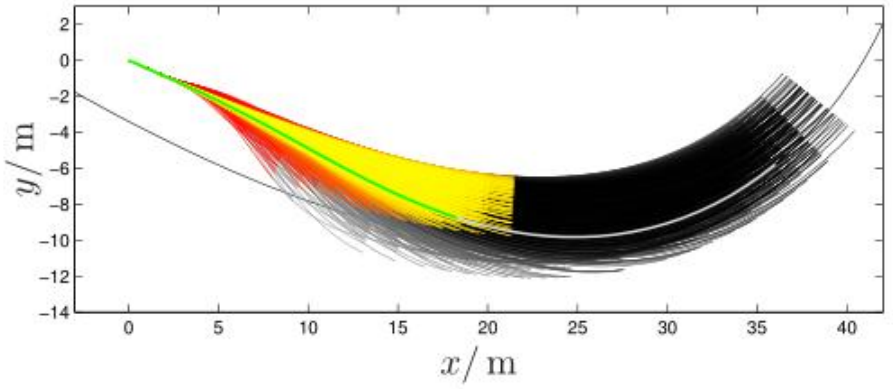
求解过程拆解：

- 横纵向关系为 $\vec{x}(s(t),d(t))=\vec{r}(s(t))+d(s(t))\vec{n}_r(s(t))$
- 可以证明当$S=s_1-s_0$，$(\cdot)':=\frac{\partial}{\partial s}(\cdot)$，$J_s(d(s)):=\int_{s_0}^{s_1}d'''^2(\sigma)\mathrm{d}\sigma$时，代价函数$C_d=k_jJ_s(d(s))+k_tS+k_dd_1^2$是关于$s$的函数，仍满足Bellman最优性原理
- 所以初始状态$D_0=[d_0,d_0',d_0'']$，沿s采样生成终止状态集合：$[d_1,d_1',d_1'',T]_{ij}=[d_i,0,0,T_j]$

基于采样的路径规划

横纵向采样轨迹结合以及代价评估

- 检查加速度限幅：$\ddot{s}$，$\ddot{d}$或d''；
- 联合代价：$C_{tot} = k_{lat}C_{lat} + k_{lon}C_{lon}$；
- 碰撞检查的范围随着时间扩张；
- 如何确保曲率符合要求？
- 当需要变更参考线时：
  ① 将当前终点$(x, \theta_x, k_x, v_x, a_x)(t_0)$投影到新的中心线上，
  ② 然后确定对应的 $[s_0, \dot{s}_0, \ddot{s}_0, d_0, \dot{d}_0, \ddot{d}_0]$ 或$[s_0, \dot{s}_0, \ddot{s}_0, d_0, d'_0, d''_0]$，
  ③ 最小化问题：$s = argmin_\sigma ||x - r(\sigma)||$。



基于采样的路径规划

总结

状态空间中采样得到候选轨迹

初始状态已知，采样轨迹末状态

横纵向分别采用五次多项式进行曲线拟合

遍历横纵向曲线，将横纵向组合成候选轨迹簇

↓

计算代价，选出最优轨迹

纵向=目标+舒适性+碰撞+向心加速度

横向=横向偏移+舒适性

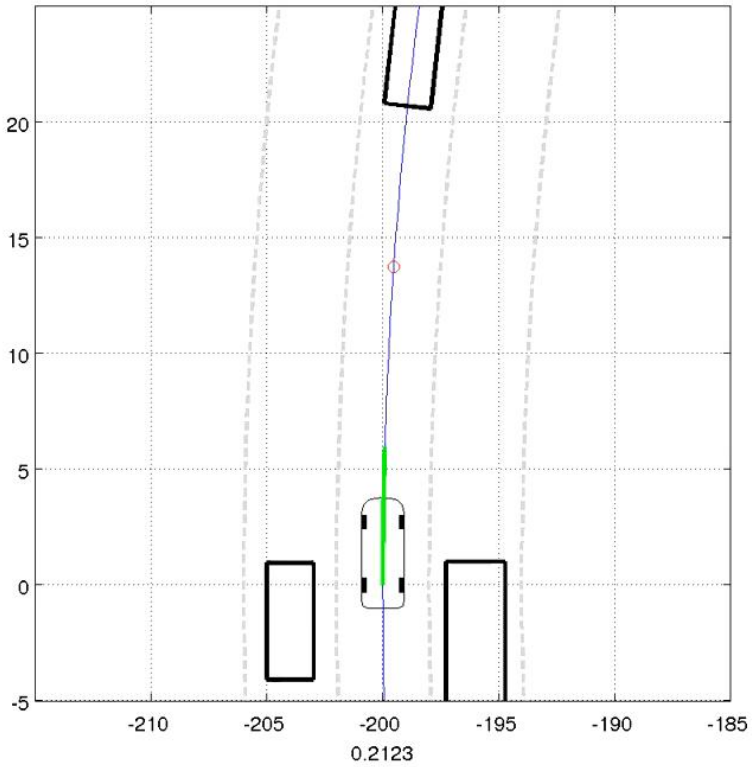
↓

检查约束和碰撞

纵向速度/加速度/加速度率(jerk)是否超出允许范围

曲率是否超出允许范围

横向加速度/加速度率(jerk)是否超出允许范围



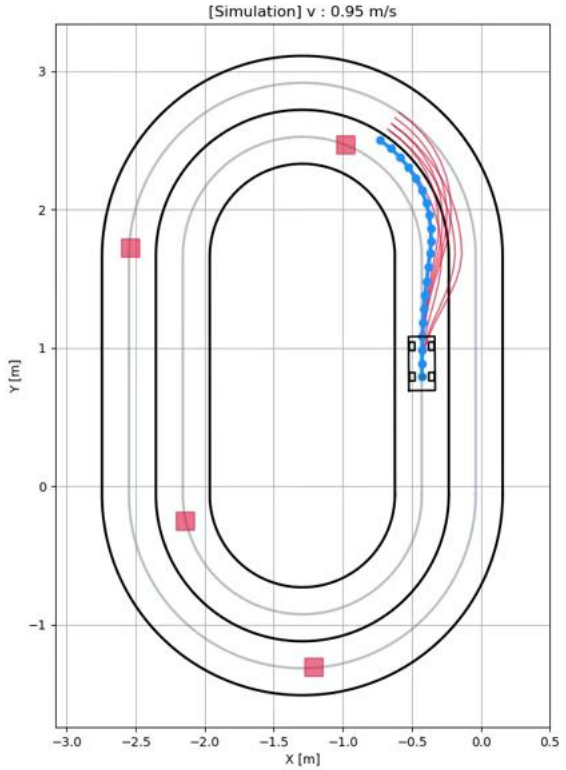
基于采样的路径规划

简单示例

```python
def frenet_optimal_planning(csp, s0, c_speed, c_accel, c_d, c_d_d, c_d_dd, ob):
    fplist = calc_frenet_paths(c_speed, c_accel, c_d, c_d_d, c_d_dd, s0)
    fplist = calc_global_paths(fplist, csp)
    fplist = check_paths(fplist, ob)

    # find minimum cost path
    min_cost = float("inf")
    best_path = None
    for fp in fplist:
        if min_cost >= fp.cf:
            min_cost = fp.cf
            best_path = fp

    return best_path
```



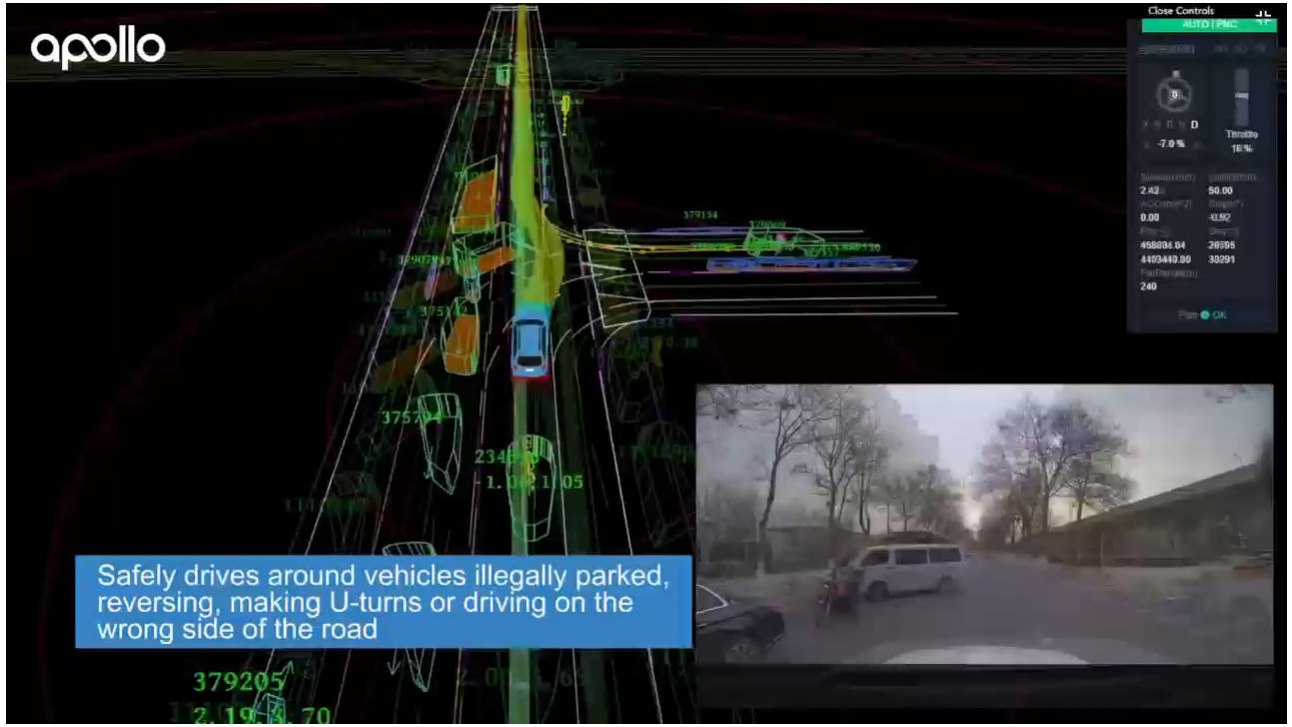
大纲

- 基于搜索的路径规划
- 基于采样的路径规划
- 基于优化的路径规划

基于优化的路径规划

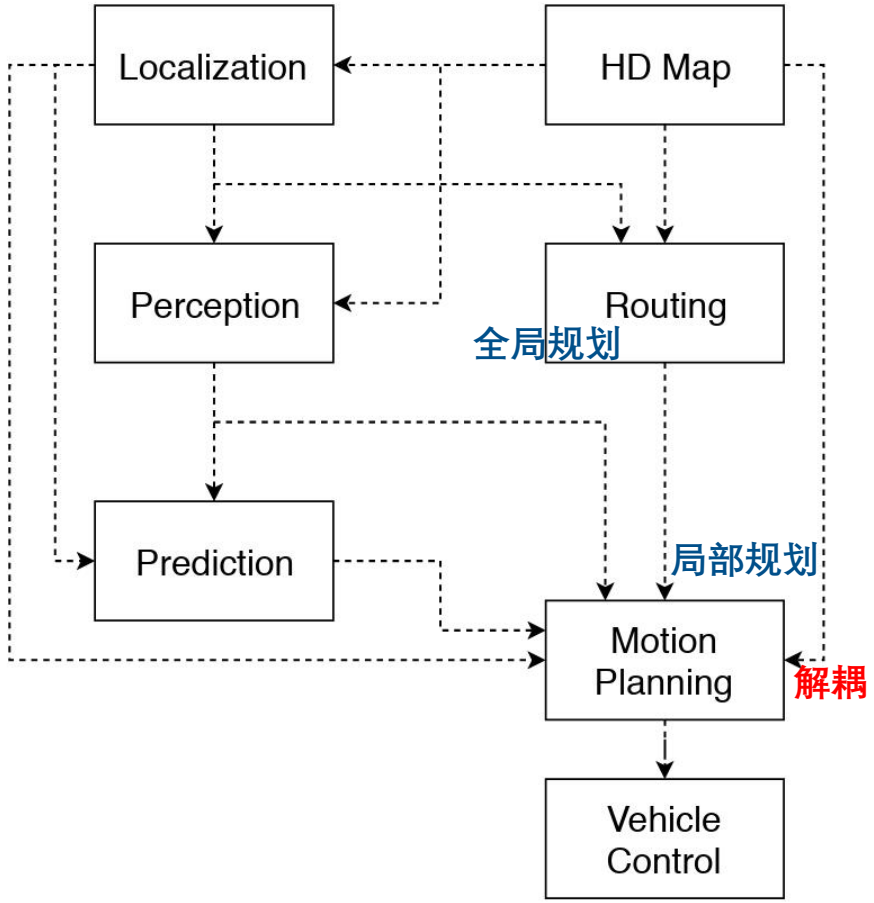
以百度Apollo EM planner为例

- 先以车道级别生成一个最可行的轨迹， 然后基于安全性和代价函数两方面来选择车道;
- 解耦成横向规划和速度规划。

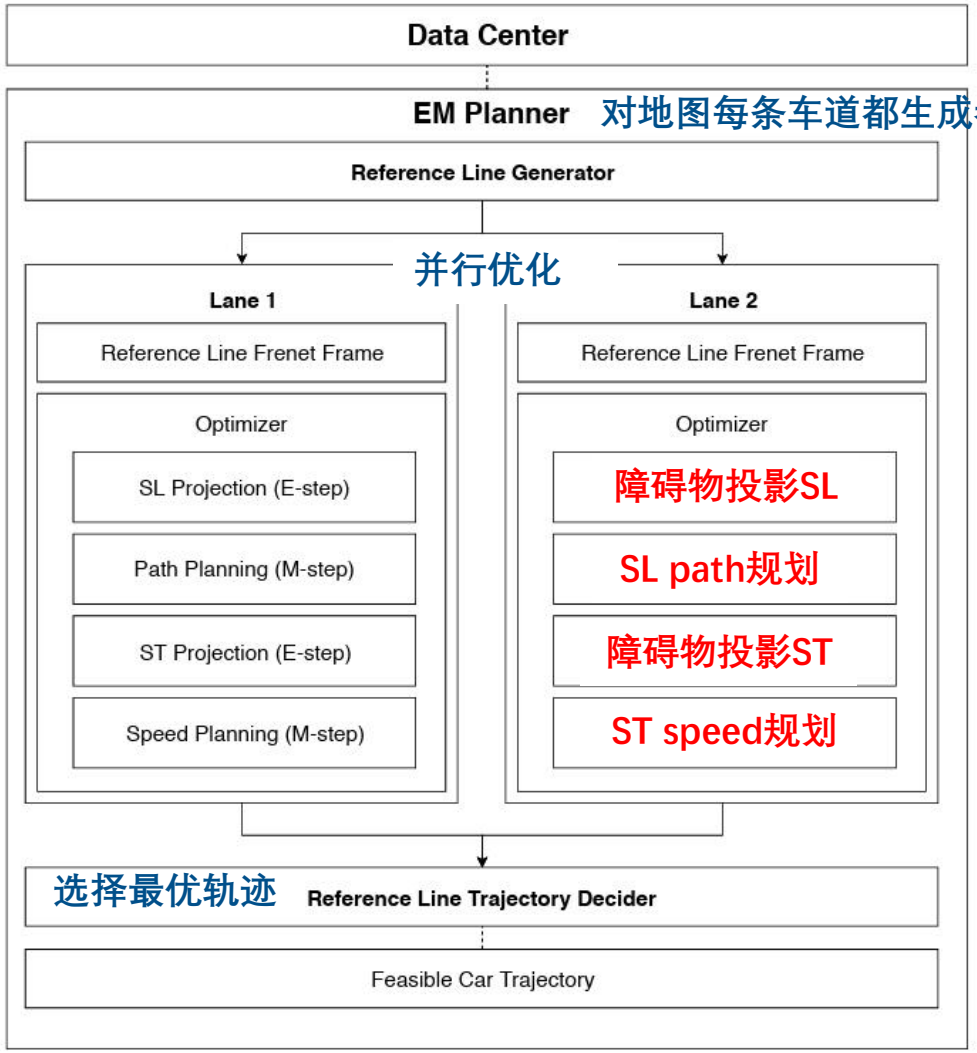


基于优化的路径规划

整体框架



上游感知、地图、定位

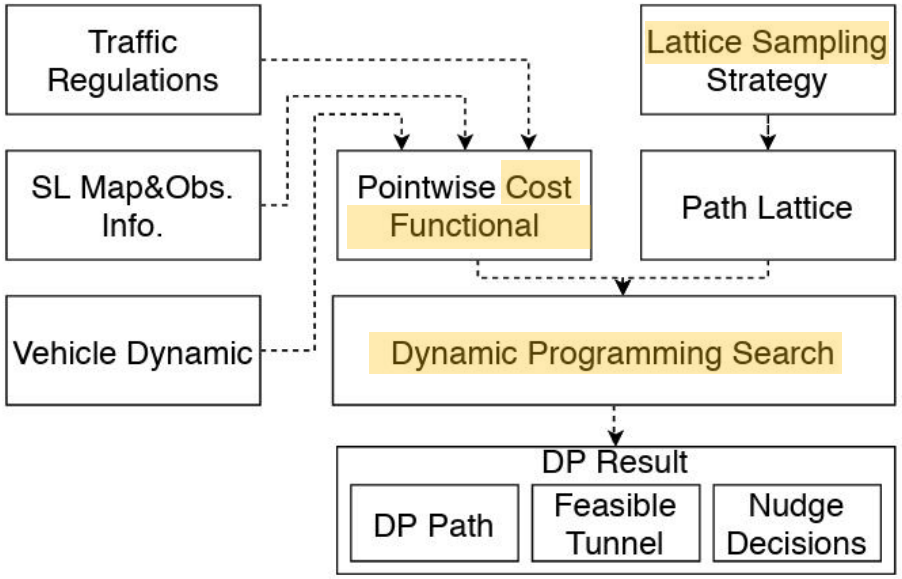


Baidu Apollo EM Motion Planner, https://arxiv.org/pdf/1807.08048.pdf

基于优化的路径规划

M-Step DP Path

- 在非凸SL空间优化横向坐标$l = f(s)$
- 横向采样→代价函数评估→动态规划



在未来至少8秒或200米的范围内采样

采样间隔依据场景决定

基于优化的路径规划

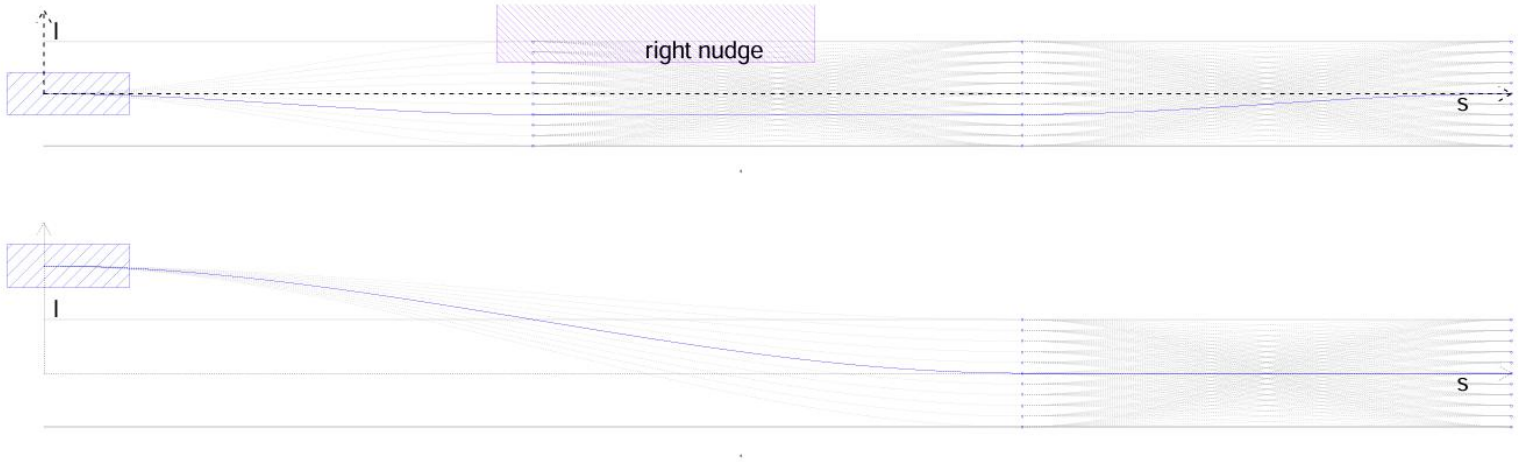
M-Step DP Path

代价函数： $C_{total}(f(s)) = C_{smooth}(f) + C_{obs}(f) + C_{guidance}(f)$

- $C_{smooth}(f)=\omega_1 \int (f'(s))^2 ds + \omega_2 \int (f''(s))^2 ds + \omega_3 \int (f'''(s))^2 ds$

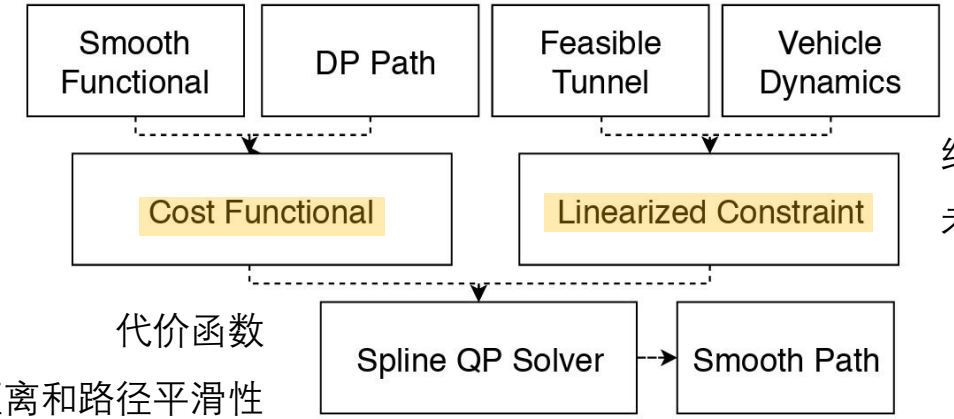
其中，$f'(s)$表示自车和车道线的航角差； $f''(s)$和path的曲率有关； $f'''(s)$和自车曲率的导数有关；

- $C_{obs}(d) = \begin{cases} 0, & d > d_n \\ C_{nudge}(d - d_c), & d_c \leq d \leq d_n \\ C_{collision}, & d < d_c \end{cases}$，由自车和障碍物的边界框之间的距离计算；
- $C_{guidance}(f) = \int (f(s) - g(s))^2 ds$，同时道路框外的path点会有较高的惩罚。



基于优化的路径规划

M-Step QP Path ——修正DP的path



线性约束

考虑了边界约束和动态可行性

代价函数

用于平衡障碍物距离和路径平滑性

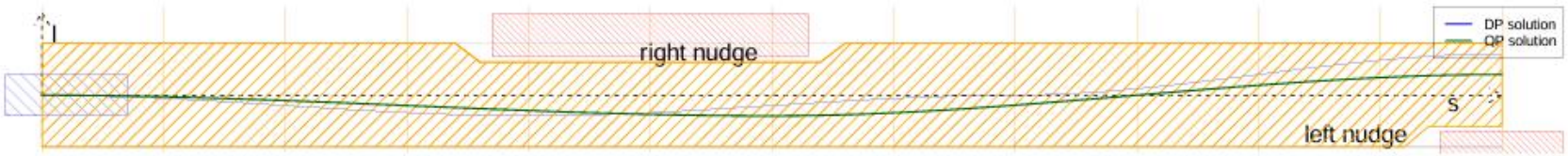
基于优化的路径规划

M-Step QP Path ——修正DP的path

代价函数：

$$C_s(f)=\omega_1\int(f'(s))^2ds+\omega_2\int(f''(s))^2ds+\omega_3\int(f'''(s))^2ds+\omega_4\int(f(s)-g(s))^2ds$$

$g(s)$表示DP过程的结果，提供了障碍物的推进距离。



基于优化的路径规划

M-Step QP Path ——修正DP的path

线性约束：

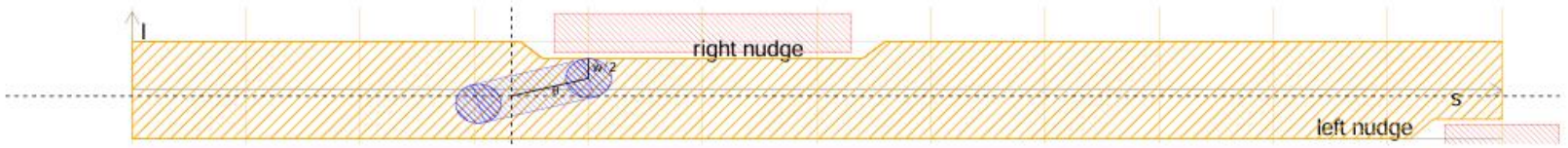
- 边界约束

以汽车左前角点为例：

$l_{left\ front\ corner} = f(s) + sin(\theta) l_f + \omega/2$,

其中l_f是汽车质心到前边界的距离，ω是汽车的宽度。

近似为 $f(s) + sin(\theta) l_f + \omega/2 \leq f(s) + f'(s) l_r + \omega/2 \leq l_{left\ corner\ bound}$。



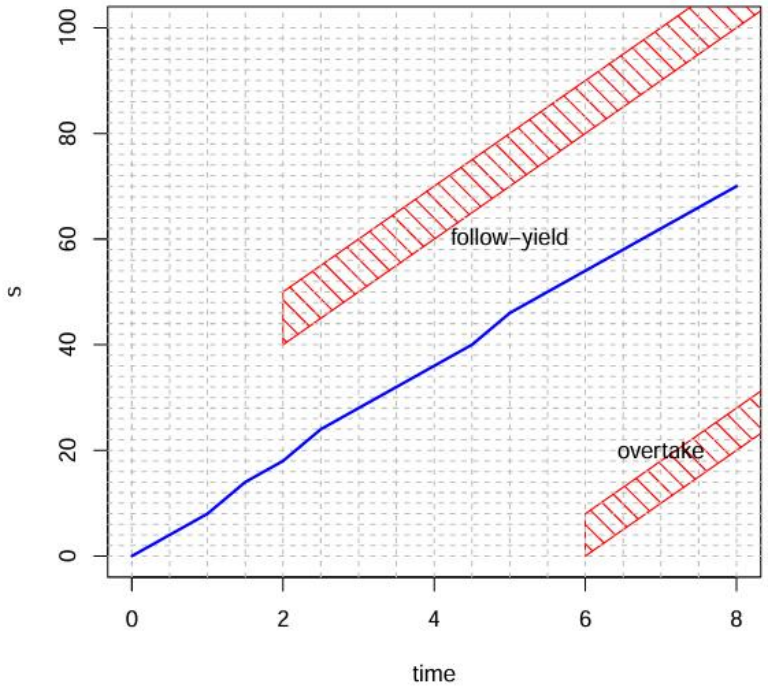
- 动态可行性

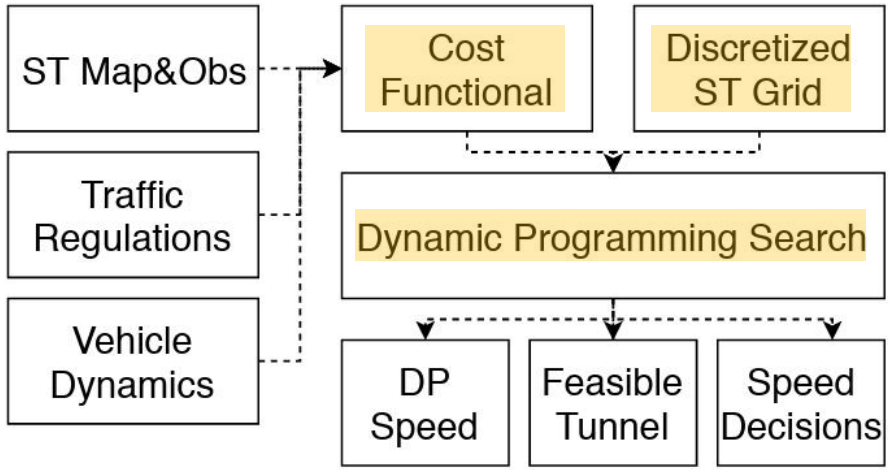
$f''(s)$和$f'''(s)$需要符合自车的初始横向位置及其导数$(f(s_0), f'(s_0), f''(s_0))$的约束。

基于优化的路径规划

M-Step DP Speed Optimizer

- 非凸问题
- 使用动态规划结合样条二次规划
- 在ST图上找到一个平滑的速度初值





基于优化的路径规划

M-Step DP Speed Optimizer

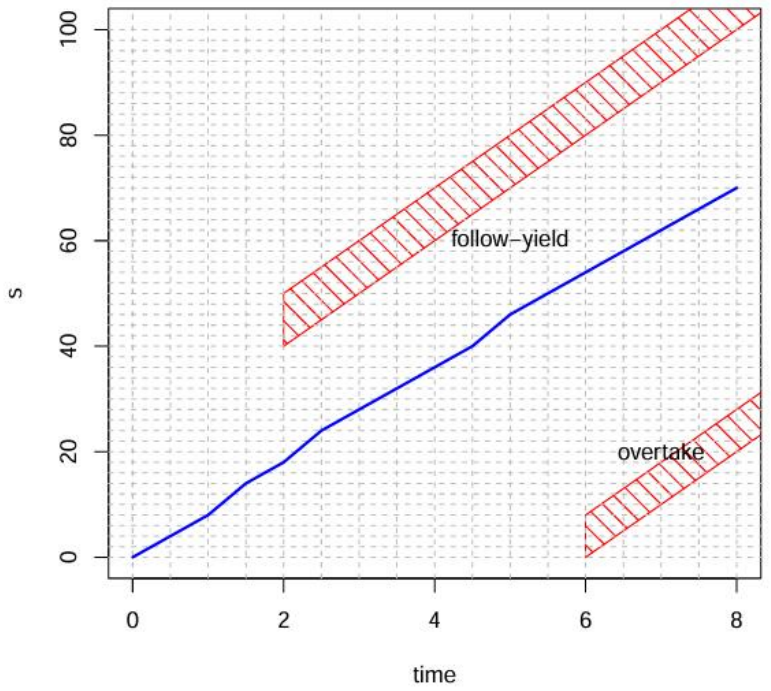
ST图：

- 障碍物信息离散为格子；
- 横轴时间：间隔dt的$(t_0, t_1 ..., t_n)$；
- 纵轴速度：分段线性速度初值$S = (s_0, s_1 ..., s_n)$，且

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

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

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



基于优化的路径规划

M-Step DP Speed Optimizer

在约束内在ST图上优化代价函数:

$$C_{total}(s) = \omega_1 \int_{t_0}^{t_n} g(S' - V_{ref})dt + \omega_2 \int_{t_0}^{t_n} (S'')^2 dt + \omega_3 \int_{t_0}^{t_n} (S''')^2 dt + \omega_4 C_{obs}(S)$$

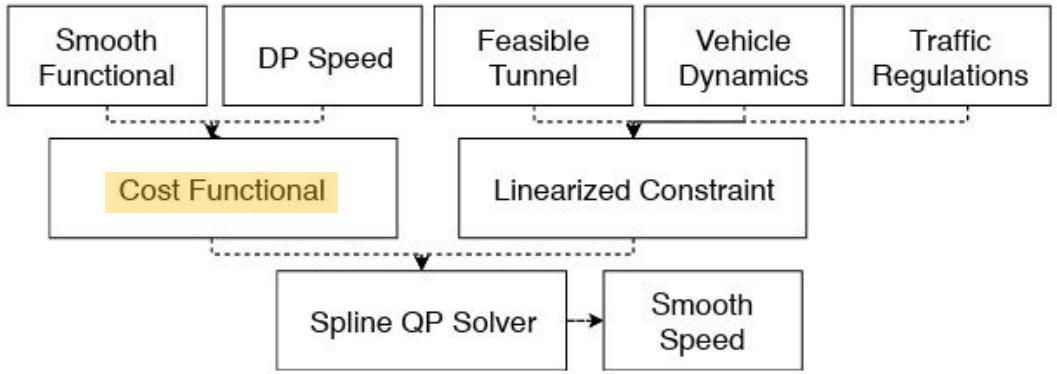
自车按照V_{ref}运动的度量
V_{ref}涉及限速等交通规则

表征了平滑性

计算了自车和所有障碍物的距离和

基于优化的路径规划

M-Step QP Speed Optimizer ——使得DP生成的分段线性速度初值满足动态要求

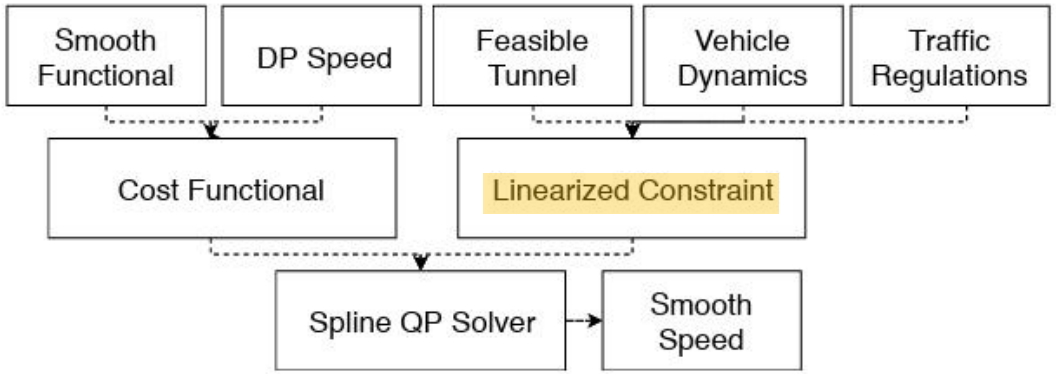


代价函数用于平衡引导线和平滑性：

$$C_{total}(s) = \omega_1 \int_{t_0}^{t_n} (S - S_{ref})^2 dt + \omega_2 \int_{t_0}^{t_n} (S'')^2 dt + \omega_3 \int_{t_0}^{t_n} (S''')^2 dt$$

基于优化的路径规划

- **M-Step QP Speed Optimizer ——使得DP生成的分段线性速度初值满足动态要求**



线性约束包括边界约束和速度、加速度约束：

$$S(t_i) \le S(t_{i+1}),\quad i = 0,1,2,\ldots,n-1,$$

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

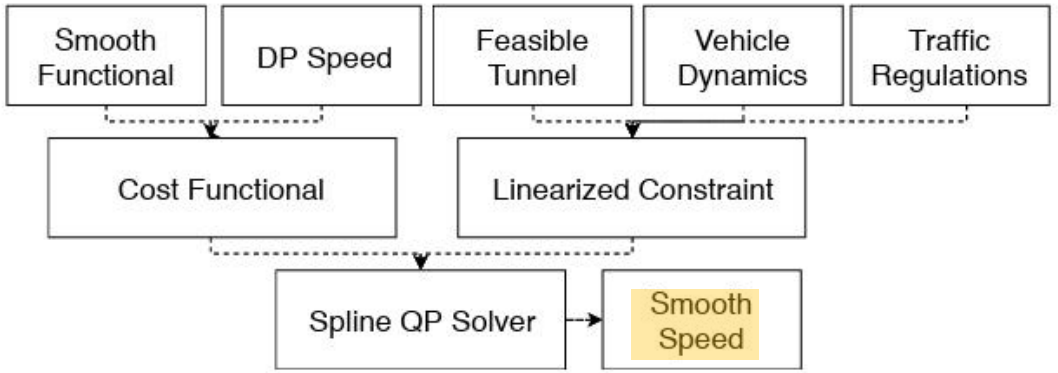
$$S'(t_i) \le V_{upper},$$

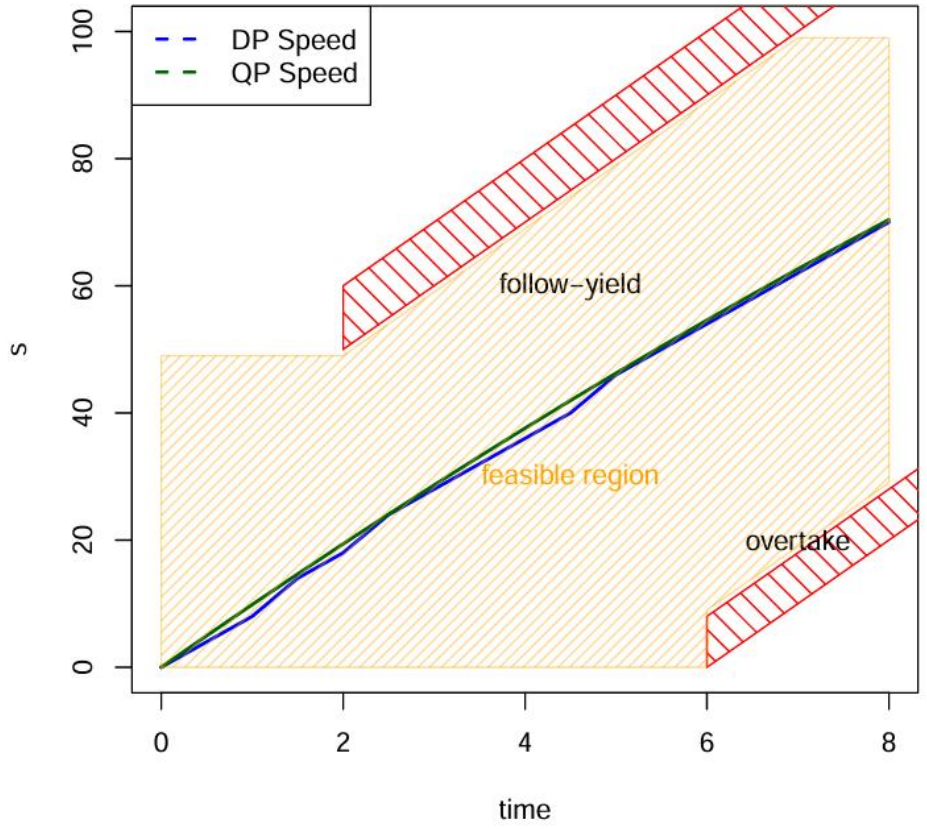
$$-Dec_{max} \le S''(t_i) \le Acc_{max},$$

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

基于优化的路径规划

- **M-Step QP Speed Optimizer ——使得DP生成的分段线性速度初值满足动态要求**





基于优化的路径规划

总结

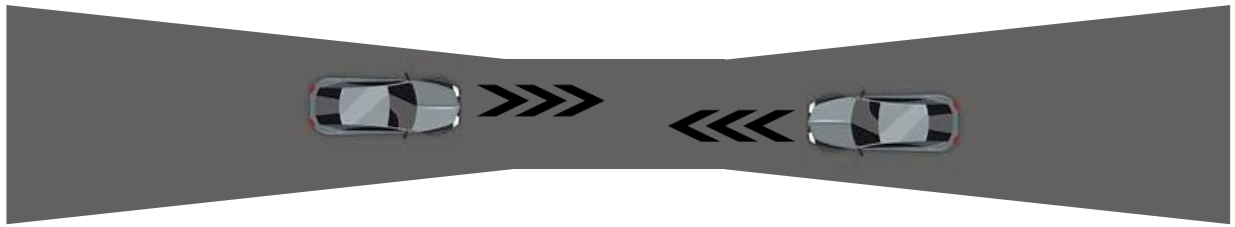
DP+QP

DP：生成凸空间

QP：在凸空间求解

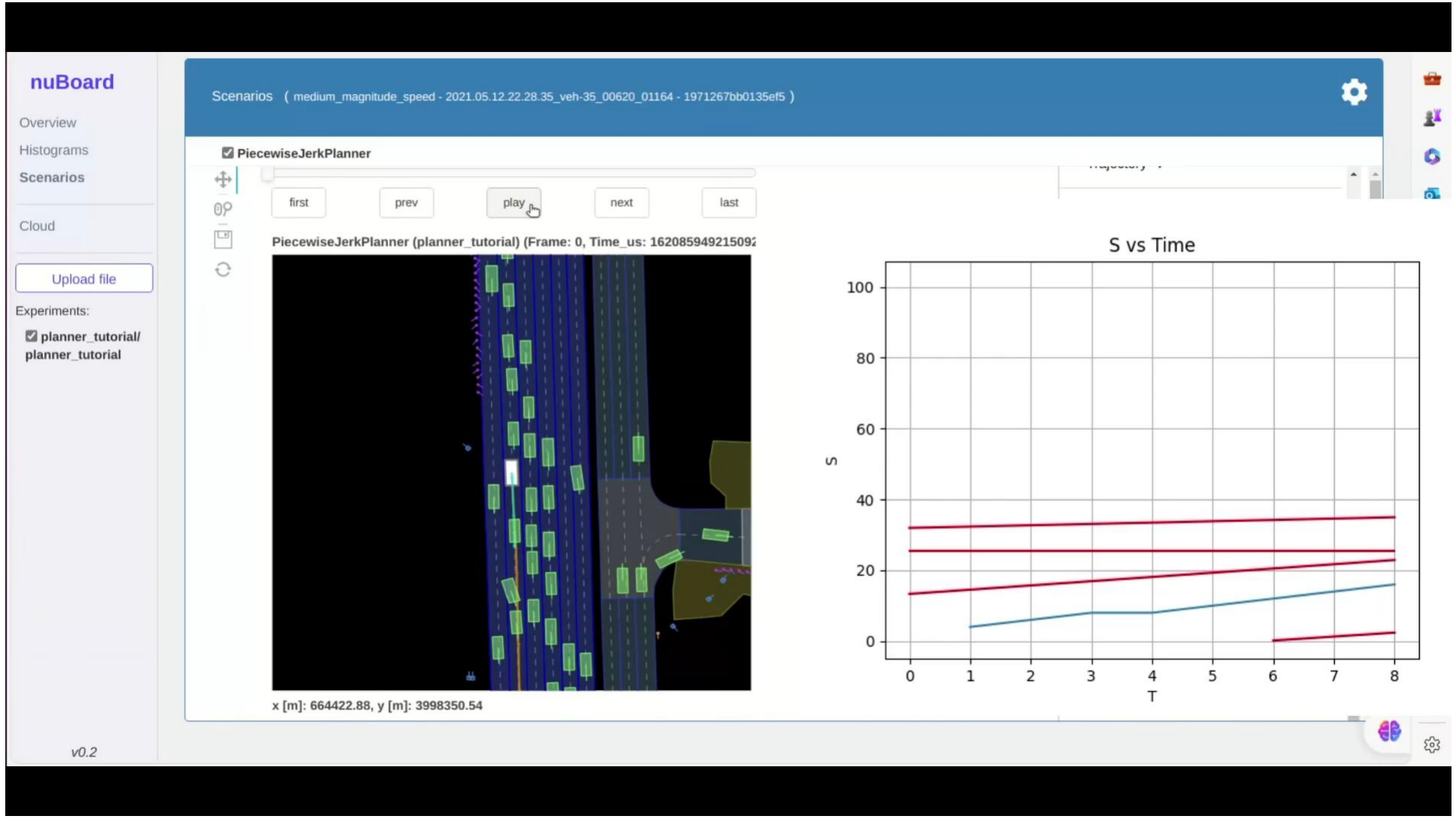
缺点

需要横纵向联合动作的场景，如：窄道会车



基于优化的路径规划

示例



深蓝学院
shenlanxueyuan.com

Magic Lab

感谢聆听!

Thanks for Listening