

自动驾驶决策规划算法第二章第二节(中)

控制接口, 轨迹拼接

现控制接口(控制输入) $x_r, y_r, \theta_r, k_r, v_r, a_r$

想一想: x_r, y_r, \dots, a_r 怎么来的

规划: 规划出了 trajectory, 包含一系列的 x, y, θ, k, v, a, t

trajectory $\left\{ \begin{array}{l} x_1, x_2, \dots, x_n \\ y_1, y_2, \dots, y_n \\ \vdots \\ a_1, a_2, \dots, a_n \\ 0, t_2, t_3, t_4, \dots, t_n \end{array} \right.$

规划周期为 100ms, 在 trajectory 找到 $t=100ms$ 所对应的 x, y, θ, k, v, a (找不到就找 100ms 相邻的 2 个时间, 做插值), 作为规划输出发出去

在规划周期同步时, 无问题

--- 不同步, 有问题

因为规划已经规划出了 10ms, 20ms, ..., 100ms 的轨迹, 但只把 100ms 后的点发了出去
必然造成控制效果变差

所以: 原代码中的控制接口不可取, 需要改进

改进: 规划直接发一条带时间的轨迹给控制

trajectory $\left(\begin{array}{l} x_1, x_2, \dots \\ t, 0, 0.01, 0.02, \dots, 4.99, 5.00 \end{array} \right)$ 规划输出

控制 10ms 搜索 trajectory

10ms $t=0.01$ 对应的 x_r, y_r, \dots

20ms $t=0.02$ 对应的 x_r, y_r, \dots

\vdots

控制根据当前控制周期的时间, 搜索轨迹, 找到当前时间对应的 x, y, \dots, v, a , 输出给控制算法去跟踪

方法很简单，细节是魔鬼

问题一：轨迹的时间是相对时间，还是绝对时间？

例：16:00:00 开始规划 16:00:10 规划完成，规划出了 4 秒轨迹

有三种发给控制的选择

(1) trajectory-t = 10. ----- 4) 相对时间

(2) trajectory-t = (16:00:00 ---- 16:04:00) } 绝对时间，区别 { 规划起点在 16:00:00
(3) trajectory-t = (16:00:10 ---- 16:04:10) } 16:00:10

答：发绝对时间（推荐），因为做拼接，做控制，绝对时间更直观方便

例：16:00 规划了 4 秒

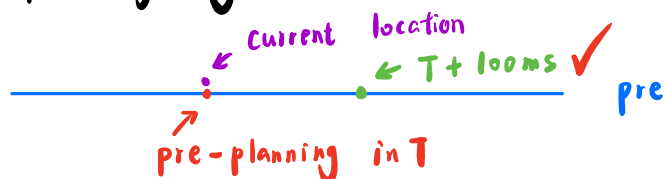
本期：16:00:10 首先获得本同期的时间（16:00:10），直接查找上一时刻规划的 16:00:10 对应的点



对于控制：16:00:13 → 查 trajectory 对应的 16:00:13 的点，输出给算法去跟踪

问题二：规划起点

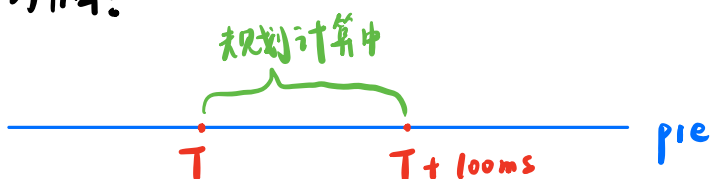
pre-trajectory



设在本同期规划开始的时间为 T

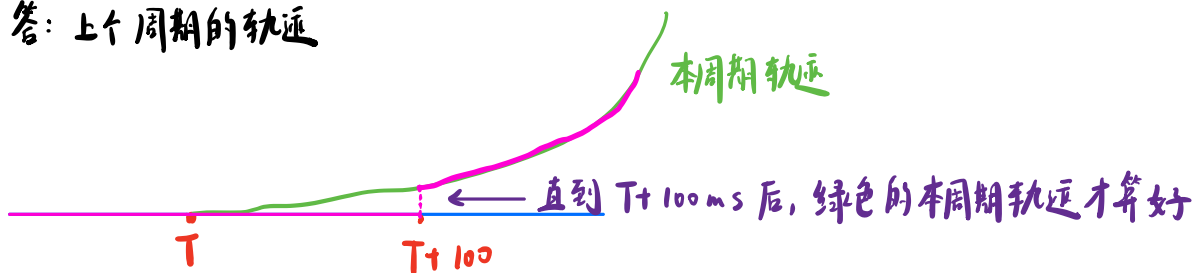
规划起点是 T，还是 T + 100ms

为什么？

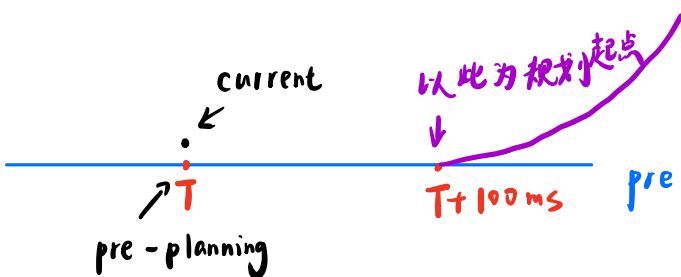
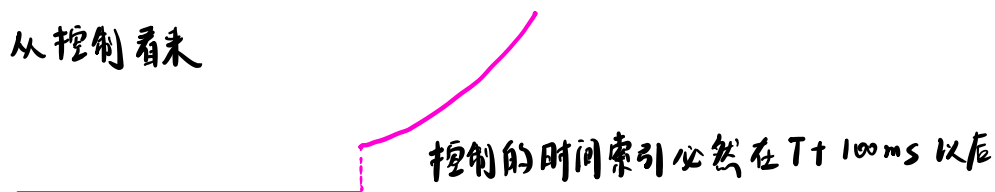


控制频率比规划快, 在 $[T, T+100\text{ms}]$ 时, 此时规划未计算完成, 那么控制跟踪的是哪个轨迹

答: 上个周期的轨迹



从控制看来



从控制看来



解释 为什么不能从定位的点作为规划起点

控制不完美



控制根据时间来找 $x_r, y_r, \theta_r, k_r, v_r, a_r$

规划同期与同期之间是跳变的