全局导航与局部规划（原理）

章品

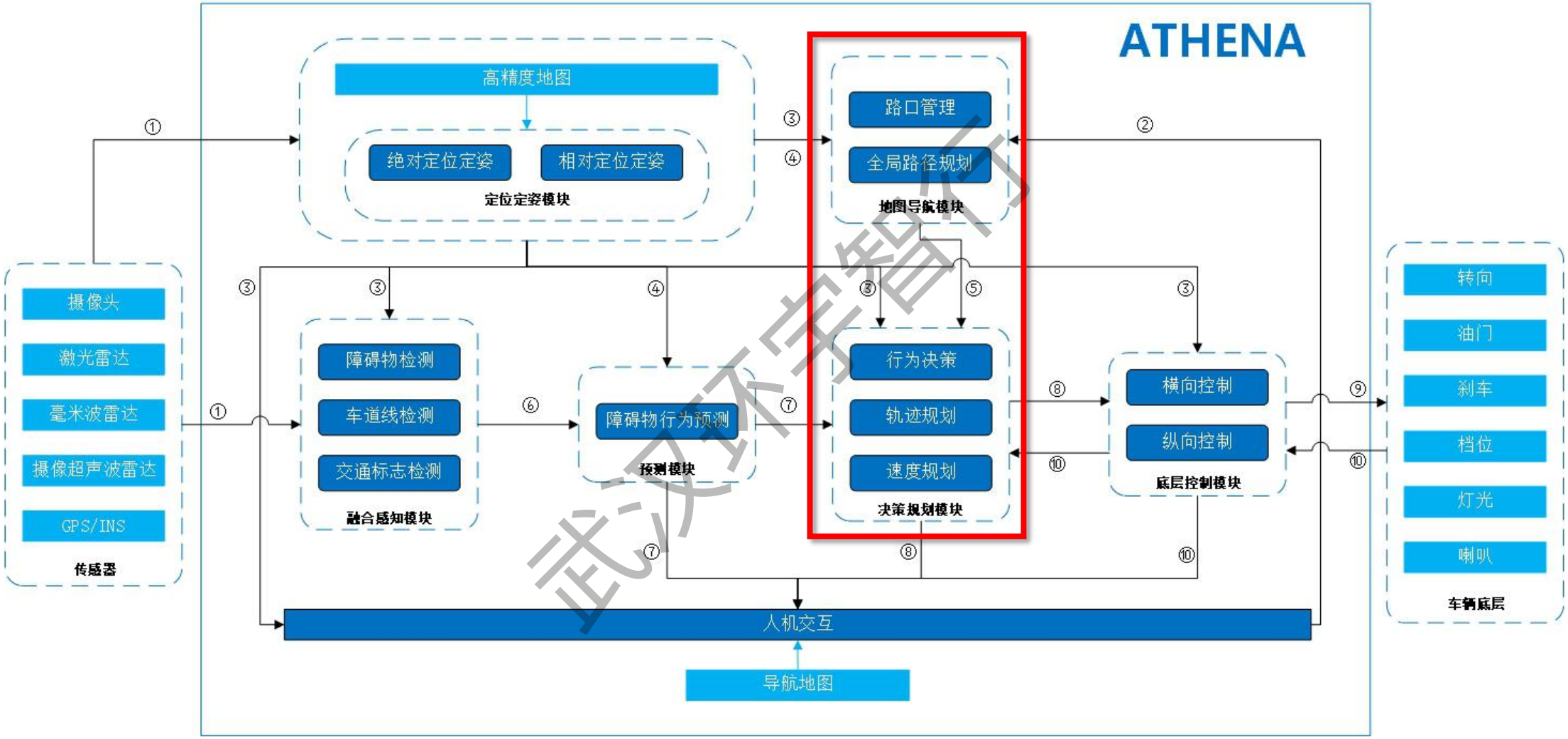
2020-03-25

目录

- 背景及框架
- 全局规划（导航）
- 局部规划

武汉环宇智行

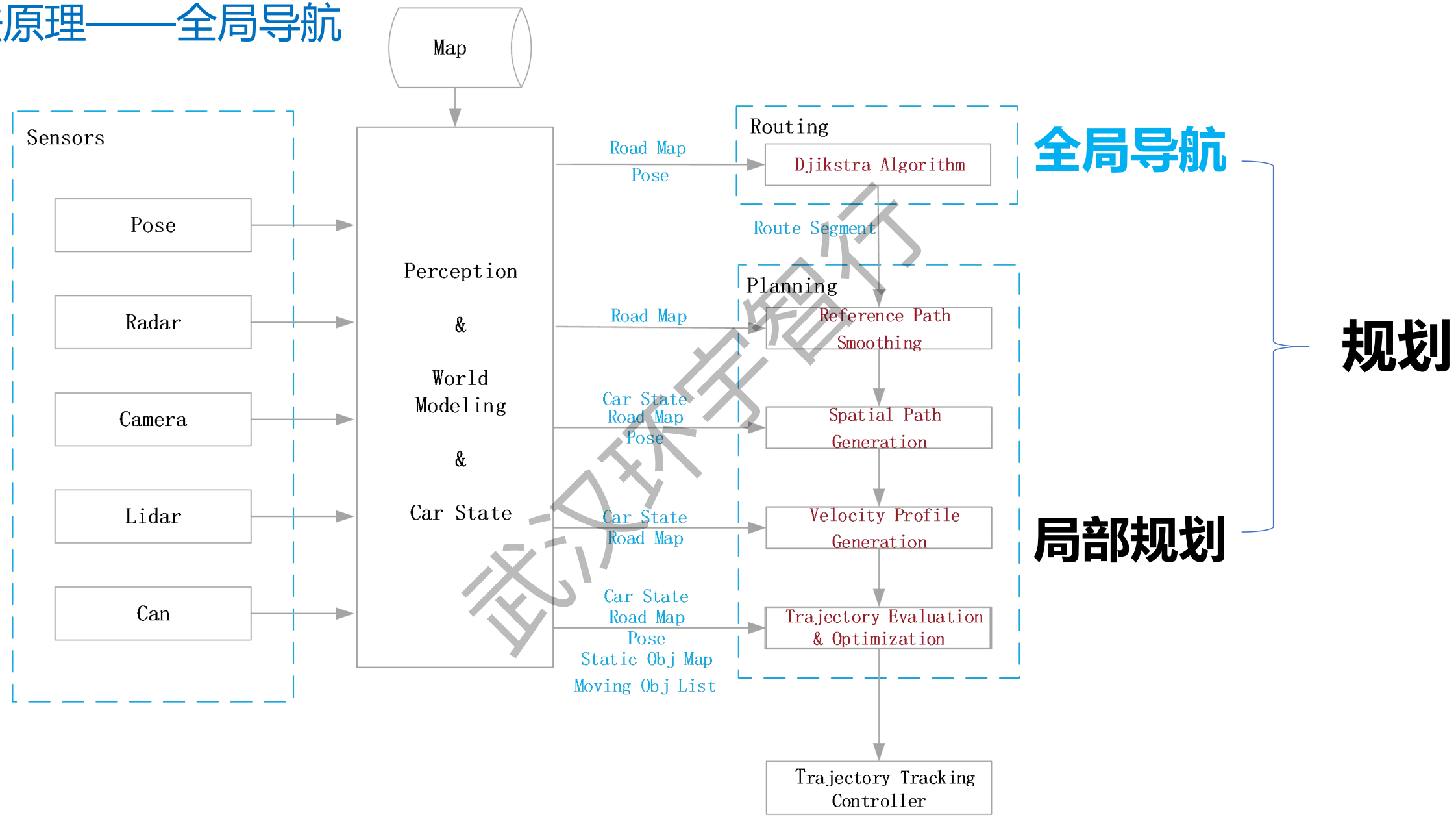
背景及框架



①传感器数据 ②界面操作指令 ③地图数据 ④定位结果 ⑤地图切片 ⑥障碍物信息 ⑦障碍物预测轨迹 ⑧规划轨迹 ⑨控制指令 ⑩车辆信息

武汉环宇智行

算法原理——全局导航



武汉环宇智行

算法原理——全局导航

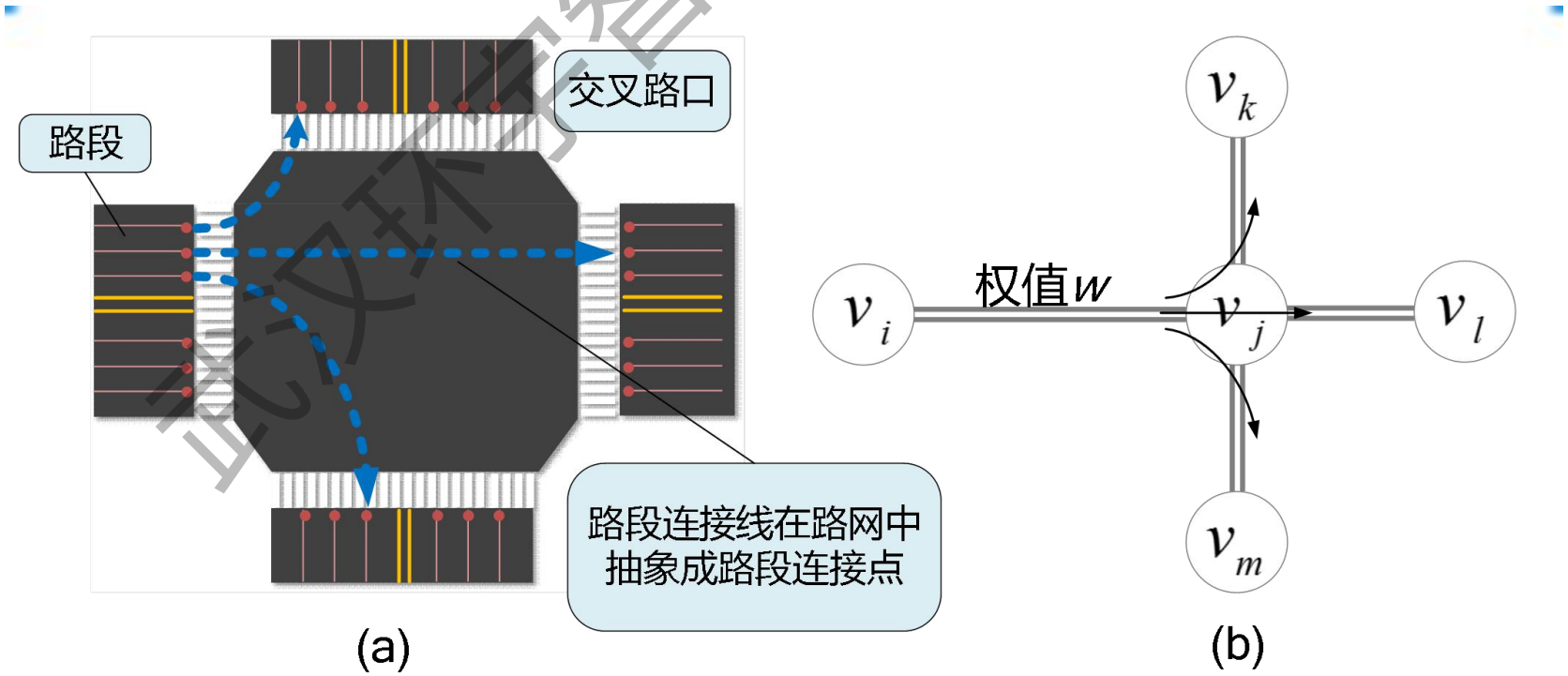
地图转换：

将静态地图抽象转化为带权有向图

（a）构建有向图

（b）计算有向边权值

$$t = t_1 + s / \min(v, v_l)$$



武汉环宇智行

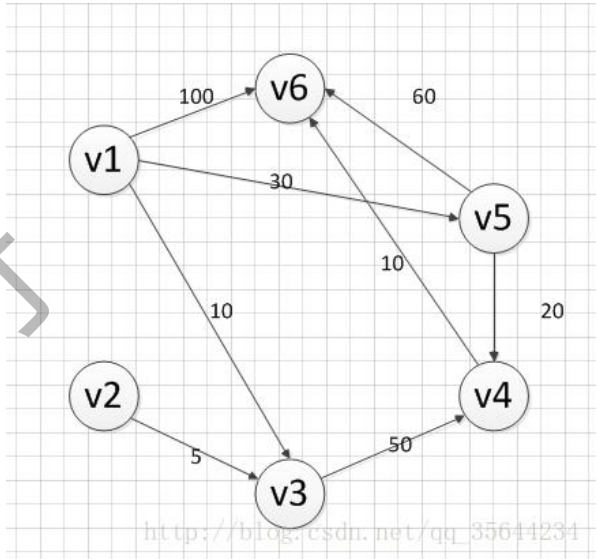
算法原理——全局导航

最优路径搜索：Dijkstra算法

设计路径搜索算法，查找最优路径。当前版本采用Dijkstra算法计算出最优路径。Dijkstra算法是典型的单源最短路径算法，用于计算一个节点到其他节点的最短路径

输入： 带权有向图 $G=(V,E)$，起点 v_0，终点 v_n

输出： 规划好的路径（记为 $Path$）。



带权有向图

Dis	0	∞	10	∞	30	100

Dis	0	∞	10	60	30	100

Dis	**0**	∞	10	50	30	90

Dis	0	∞	10	50	30	60

Dis	0	∞	10	50	30	60

武汉环宇智行

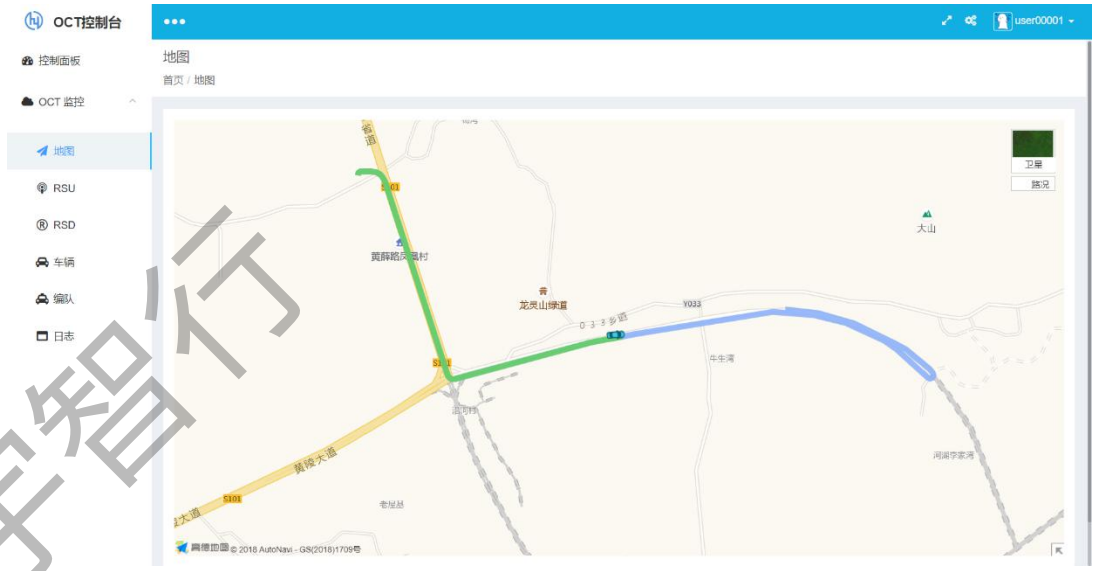
算法原理——全局导航

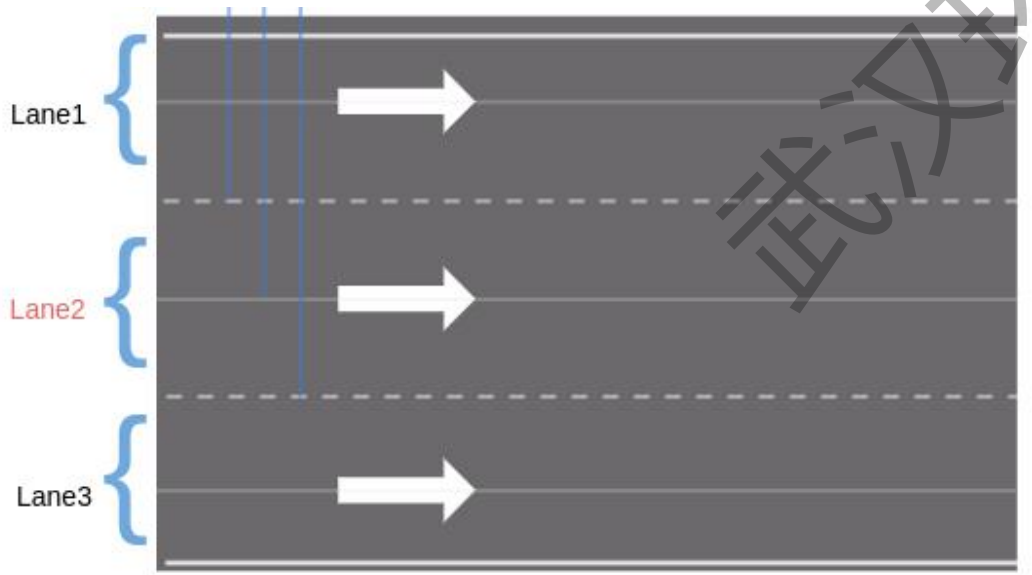
路由切片：

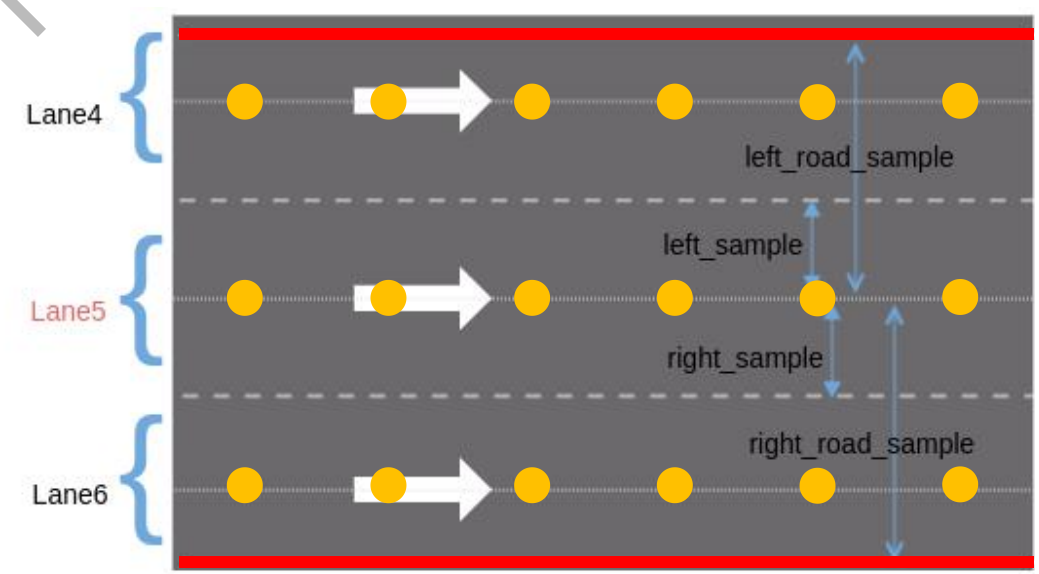
1. 地图中心线切片点
- 坐标
- 航向角
- 曲率

2. 边界线
- 坐标
- 类型

3. 交通信息
- 停止线
- 地图限速

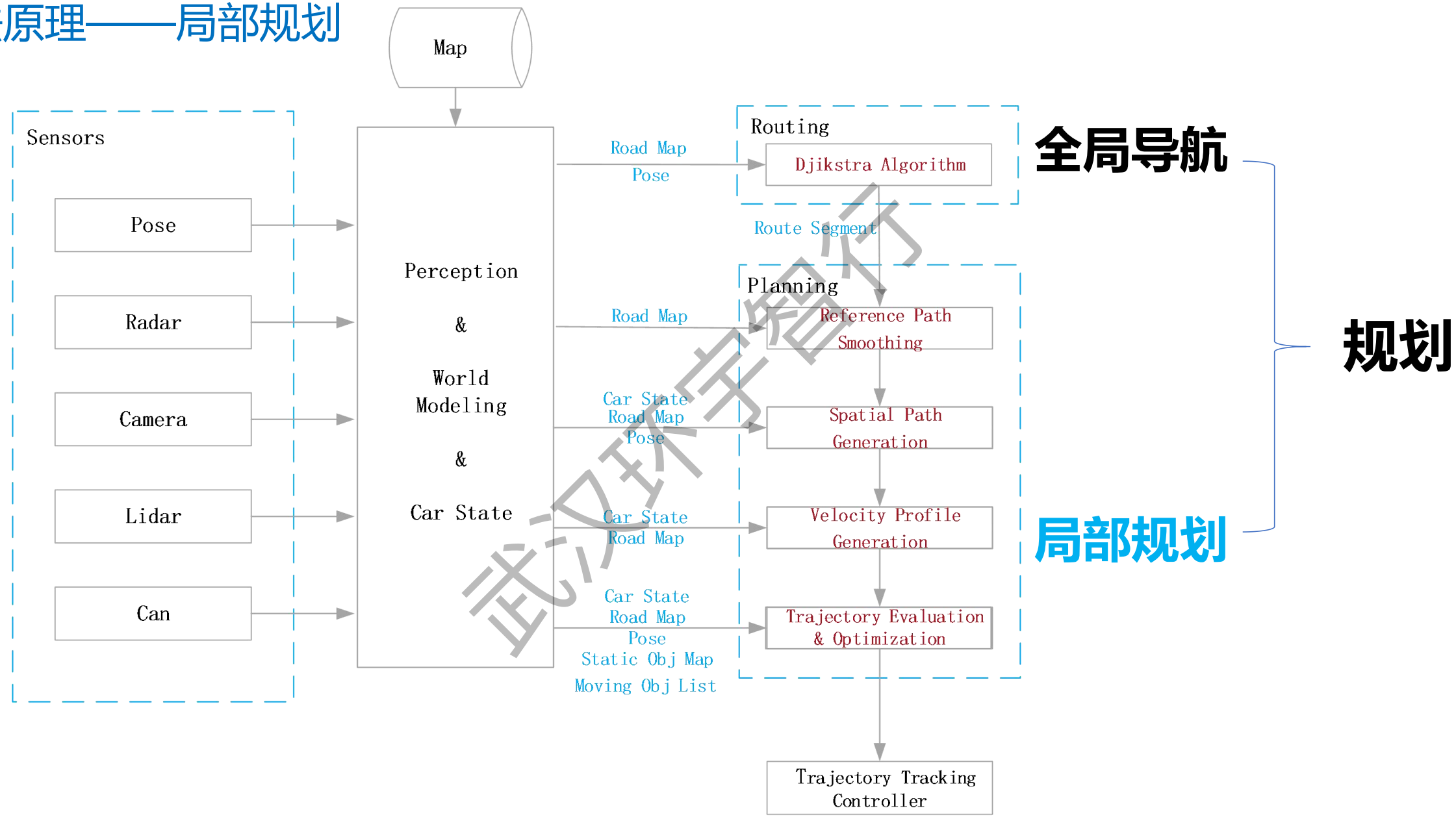






武汉环宇智行

算法原理——局部规划



武汉环宇智行

算法原理——局部规划

参考线的平滑处理

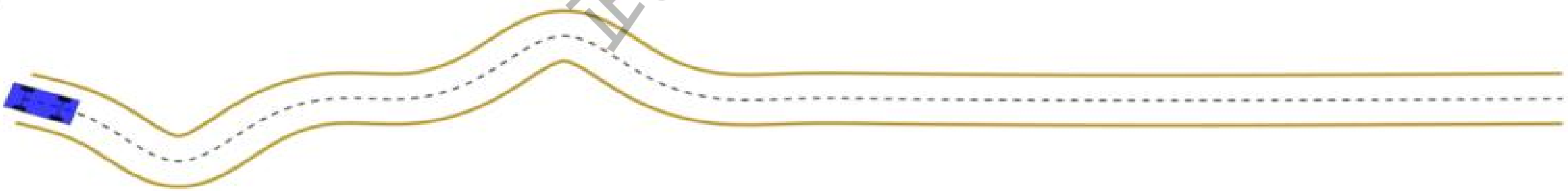
不平顺 → **拟合**

间隔大 → **内插**

$$\begin{cases} x_{bf}(s) = a_{x,i}(s-s_i)^3 + b_{x,i}(s-s_i)^2 + c_{x,i}(s-s_i) + d_{x,i} \\ y_{bf}(s) = a_{y,i}(s-s_i)^3 + b_{y,i}(s-s_i)^2 + c_{y,i}(s-s_i) + d_{y,i} \end{cases}$$

$$\begin{cases} \theta_{bf} = \arctan \dfrac{dy_{bf}}{dx_{bf}} \\ \kappa_{bf} = \dfrac{x_{bf}' y_{bf}'' - x_{bf}'' y_{bf}'}{(x_{bf}' + y_{bf}')^2} \end{cases}$$

武汉环宇智行



算法原理——局部规划

生成备选轨迹簇：

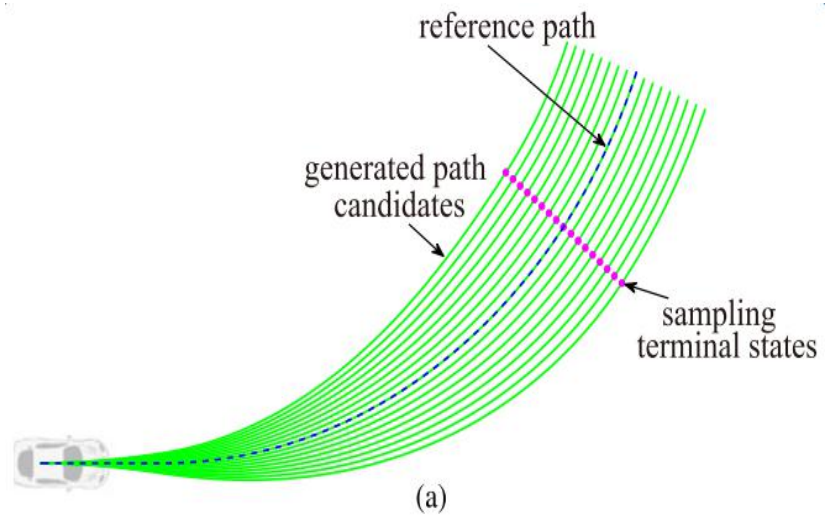
1. 撒点：

根据参考中心线获得采样点

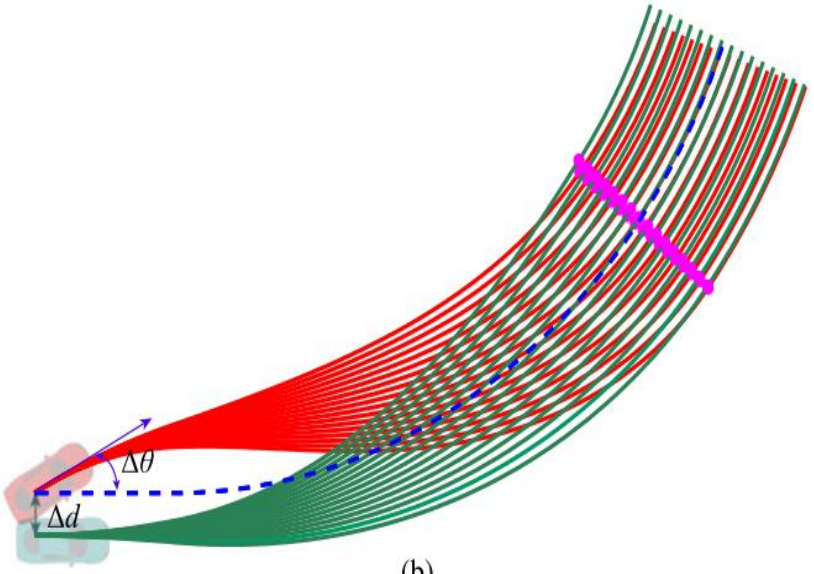
2. 拟合：

初始状态 ------→ 目标状态

三次样条 ------→ 五次样条



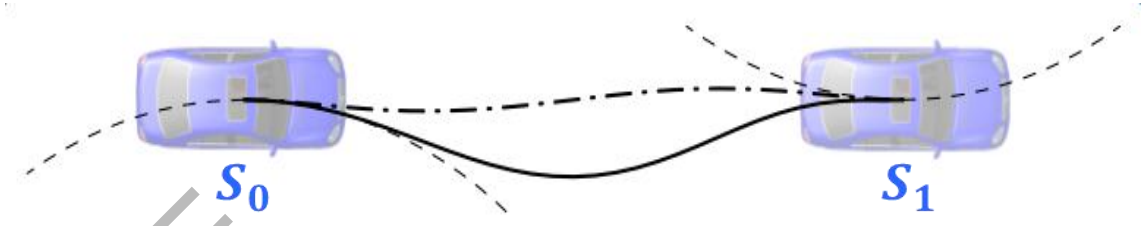
(a)



(b)

武汉环宇智行

算法原理——局部规划



五次多项式求解：

已知：

$S_0(x_0, y_0, heading_0, k_s_0)$

$S_1(x_1, y_1, heading_1, k_s_1)$

$$y = a0 + a1(x - x_0) + a2(x - x_0)^2 + a3(x - x_0)^3 + a4(x - x_0)^4 + a5(x - x_0)^5$$

求解：

$a_0, a_1, \ldots, a_5$

$$y^{\bullet} = a1 + 2a2(x - x_0) + 3a3(x - x_0)^2 + 4a4(x - x_0)^3 + 5a5(x - x_0)^4$$

$$y^{\bullet\bullet} = 2a2 + 6a3(x - x_0) + 12a4(x - x_0)^2 + 20a5(x - x_0)^3$$

$$y^{\bullet}(x_0) = d_start = \tan(heading0)$$

$$y^{\bullet}(x_1) = d_end = \tan(heading1)$$

$$y^{\bullet\bullet}(x_0) = dd_start = k_s0(1 + d_start^2)^{\frac{3}{2}}$$

$$y^{\bullet\bullet}(x_1) = dd_end = k_s1(1 + d_end^2)^{\frac{3}{2}}$$

武汉环宇智行

算法原理——局部规划

五次多项式求解：

$$y = a0 + a1(x-x_0) + a2(x-x_0)^2 + a3(x-x_0)^3 + a4(x-x_0)^4 + a5(x-x_0)^5$$

$$y^{\bullet} = a1 + 2a2(x-x_0) + 3a3(x-x_0)^2 + 4a4(x-x_0)^3 + 5a5(x-x_0)^4$$

$$y^{\bullet\bullet} = 2a2 + 6a3(x-x_0) + 12a4(x-x_0)^2 + 20a5(x-x_0)^3$$

$$y^{\bullet}(x_0) = d_start = \tan(heading0)$$

$$y^{\bullet}(x_1) = d_end = \tan(heading1)$$

$$y^{\bullet\bullet}(x_0) = dd_start = k_s0(1+d_start^2)^{\frac{3}{2}}$$

$$y^{\bullet\bullet}(x_1) = dd_end = k_s1(1+d_end^2)^{\frac{3}{2}}$$

$$\begin{cases} y_0 = a_0 \\ y_1 = a0 + a1(x-x_0) + a2(x-x_0)^2 + a3(x-x_0)^3 + a4(x-x_0)^4 + a5(x-x_0)^5 \\ d_start = a1 \\ d_end = a1 + 2a2(x-x_0) + 3a3(x-x_0)^2 + 4a4(x-x_0)^3 + 5a5(x-x_0)^4 \\ dd_start = 2a2 \\ dd_end = 2a2 + 6a3(x-x_0) + 12a4(x-x_0)^2 + 20a5(x-x_0)^3 \end{cases}$$

$$\begin{cases} a_0 = y_0 \\ a1 = d_start \\ a2 = dd_start/2 \\ a3 = [20delta_y - (12d_start + 8d_end)delta_x + (dd_end - 3dd_start)delta_x2]/(2delta_x3) \\ a4 = [-30delta_y + (16d_start + 14d_end)delta_x + (3dd_start - 2dd_end)delta_x2]/(2delta_x4) \\ a5 = [12delta_y - (6d_start + 6d_end)delta_x + (dd_end - dd_start)delta_x2]/(2delta_x5) \end{cases}$$

其中：

$$\begin{cases} delta_y = y_1 - y_0 \\ delta_x = x_1 - x_0 \\ delta_x2 = delta_x * delta_x \\ delta_x3 = delta_x * delta_x2 \\ delta_x4 = delta_x2 * delta_x2 \\ delta_x5 = delta_x * delta_x2 * delta_x2 \end{cases}$$

武汉环宇智行

算法原理——局部规划

生成纵向信息：

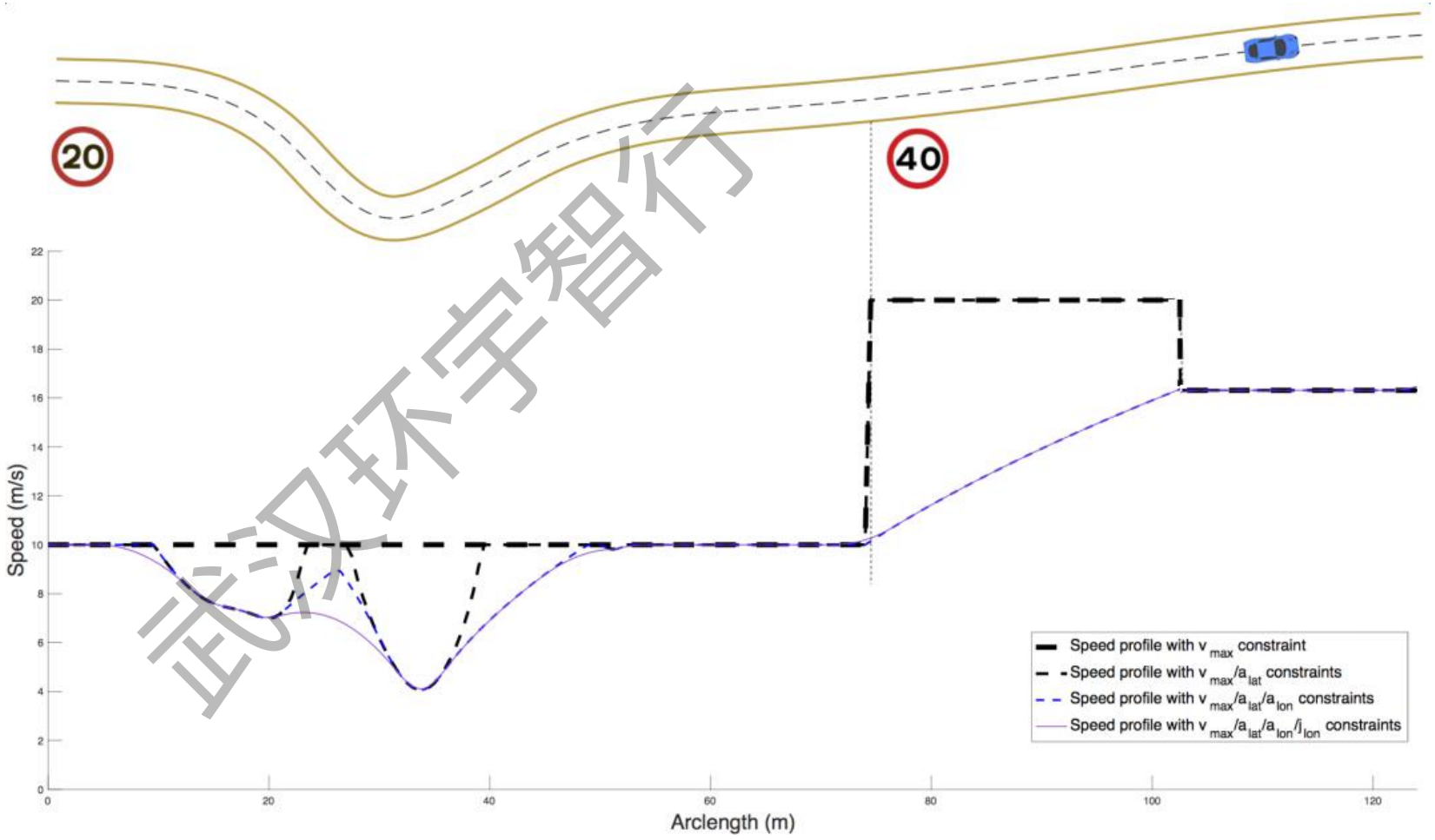
道路限速约束

障碍物约束

横向加速度约束

纵向加速度约束

连续性约束



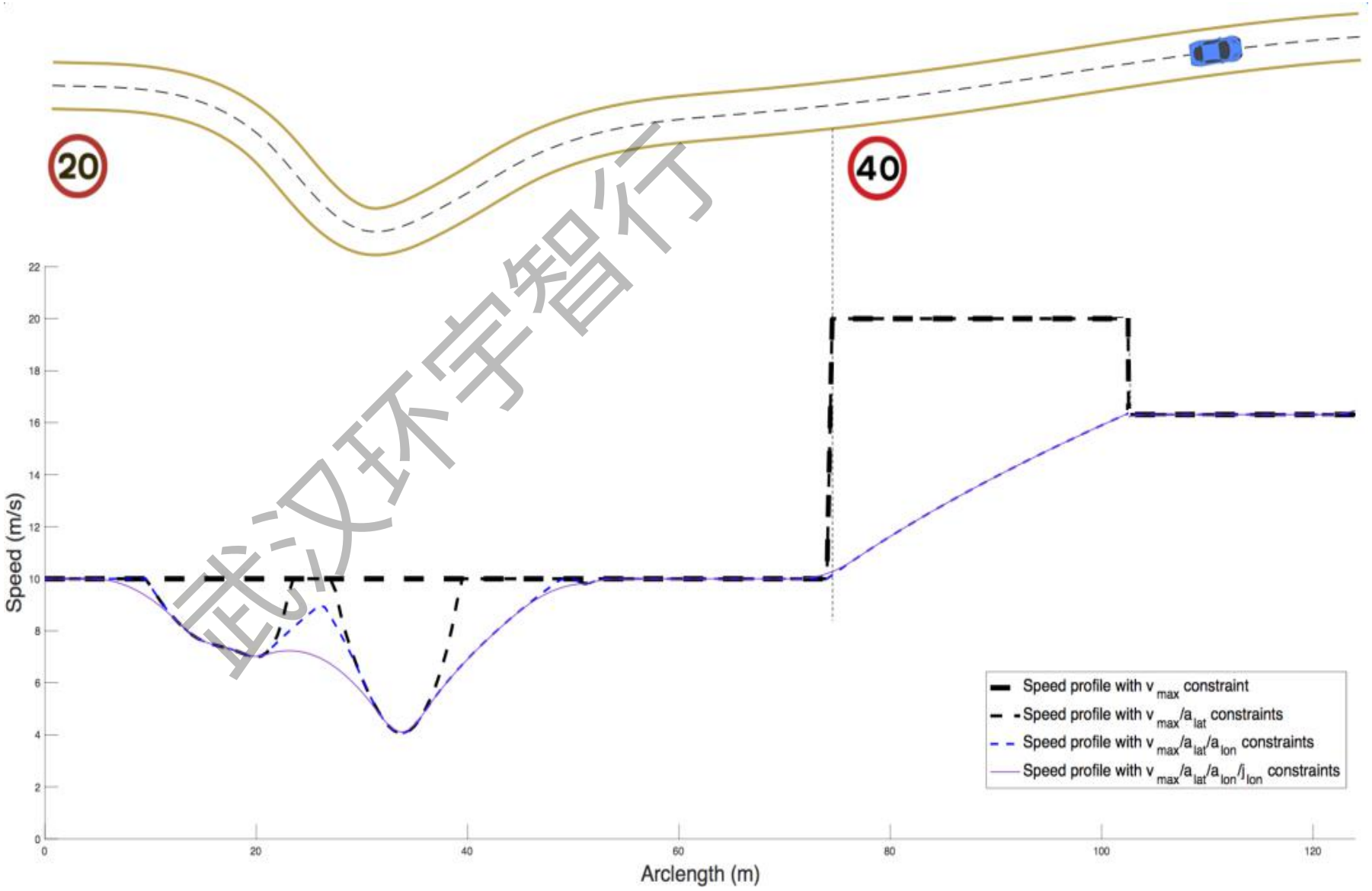
武汉环宇智行

算法原理——局部规划

生成纵向信息：

最大限速：

可能来源于交通限速牌的限速，或车辆本身的一些性能，使得主车在该路段上，不允许超过该速度v_{max}。



武汉环宇智行

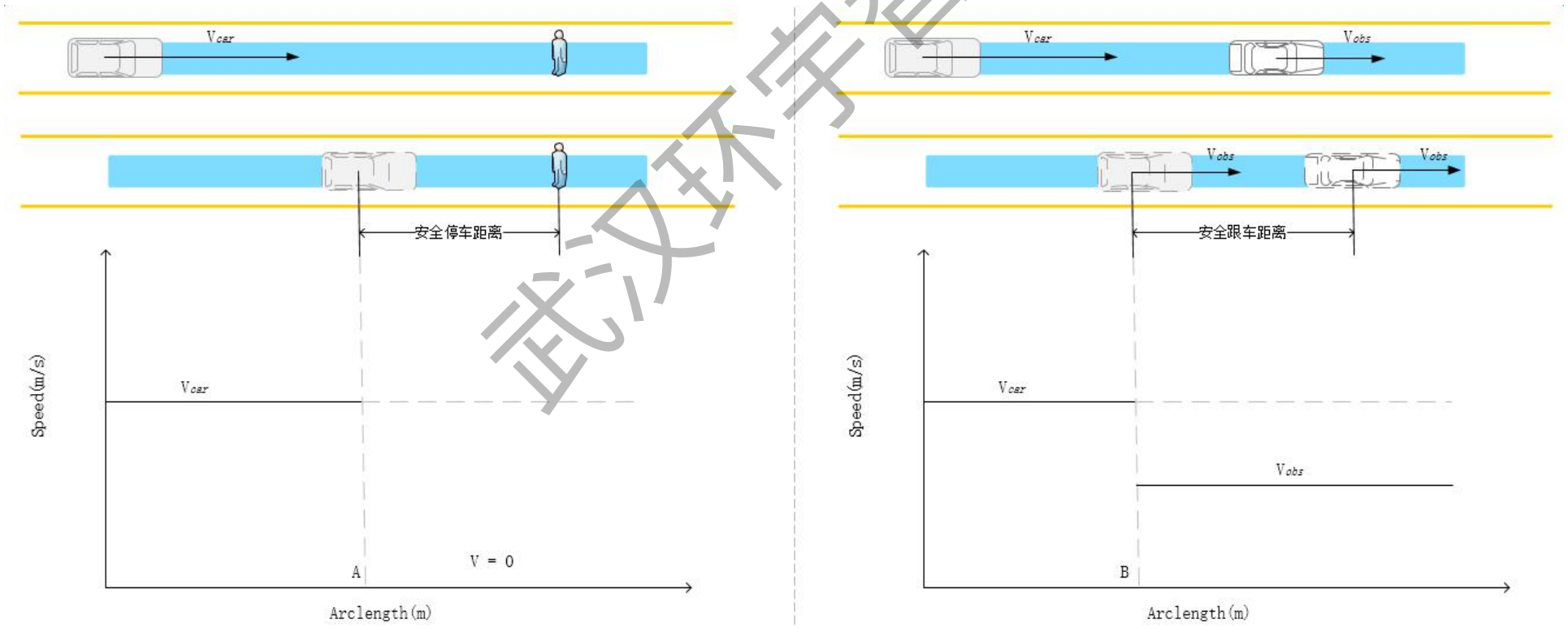
算法原理——局部规划　生成纵向信息：

障碍物速度约束：

不同运动状态的障碍物也会影响车辆规划的速度。

例如，左上图，车辆速度为v_{car}，规划路线上出现行人，车辆需要在A点前刹车，保证不与行人发生碰撞，则速度规划结果如左下图所示。

如右上图所示，车辆速度为v_{car}，前方障碍车速度为v_{obs}，计算本车要在B点处与障碍车同速，保证一定距离的安全跟车，则速度规划结果如右下图所示。



武汉环宇智行

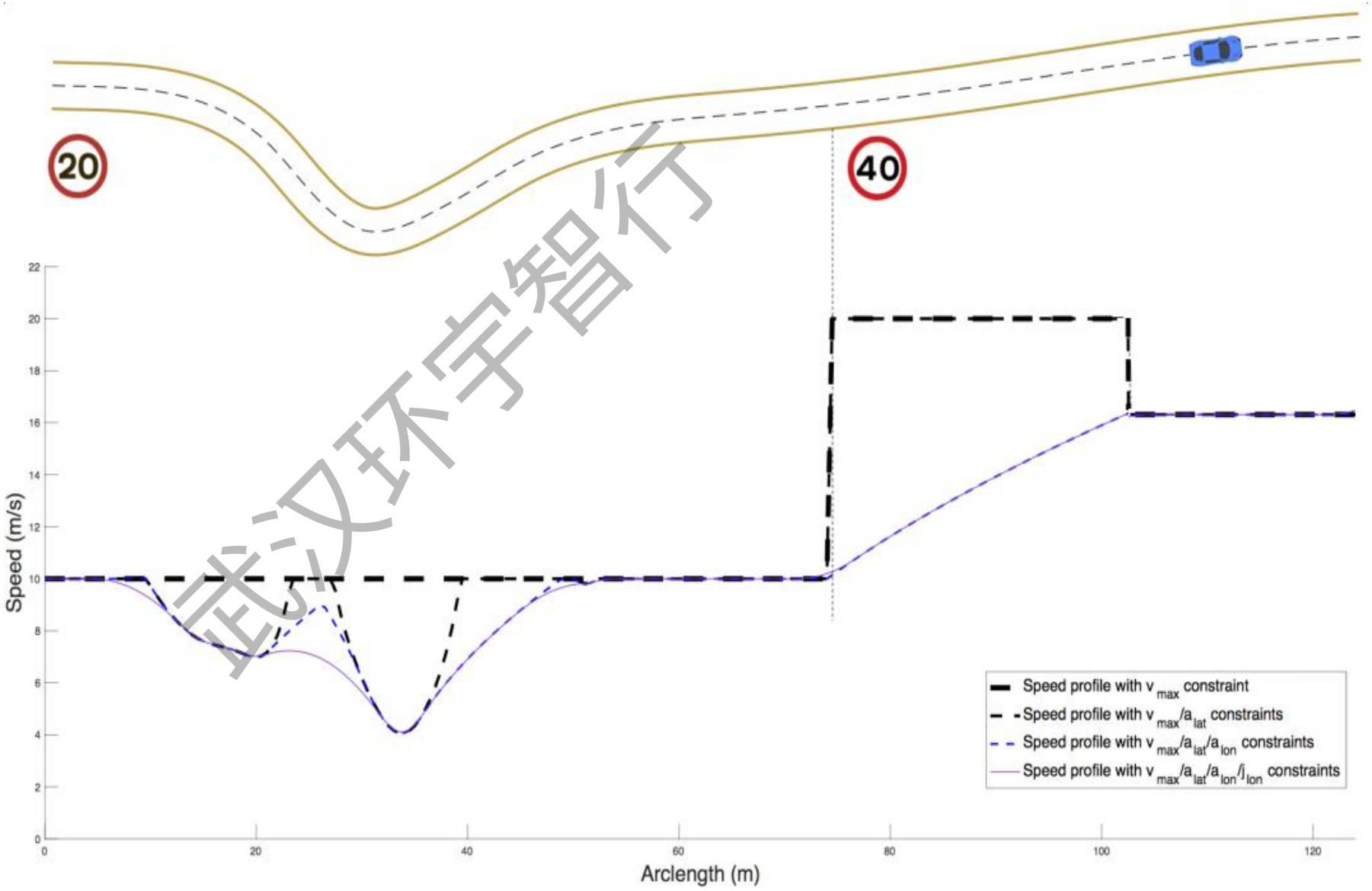
算法原理——局部规划

生成纵向信息：

横向加速度约束：

在转弯处，转弯半径越小，速度越大，离心力越大，人的舒适感越不好，并且如果离心力过大，高速情况下，车辆可能会冲出轨迹，因此需要对离心加速度做一个限制：

$$v_i \le min\{\sqrt{\frac{a_{max}^{lat}}{\kappa_i}}, v_{max}\}$$



武汉环宇智行

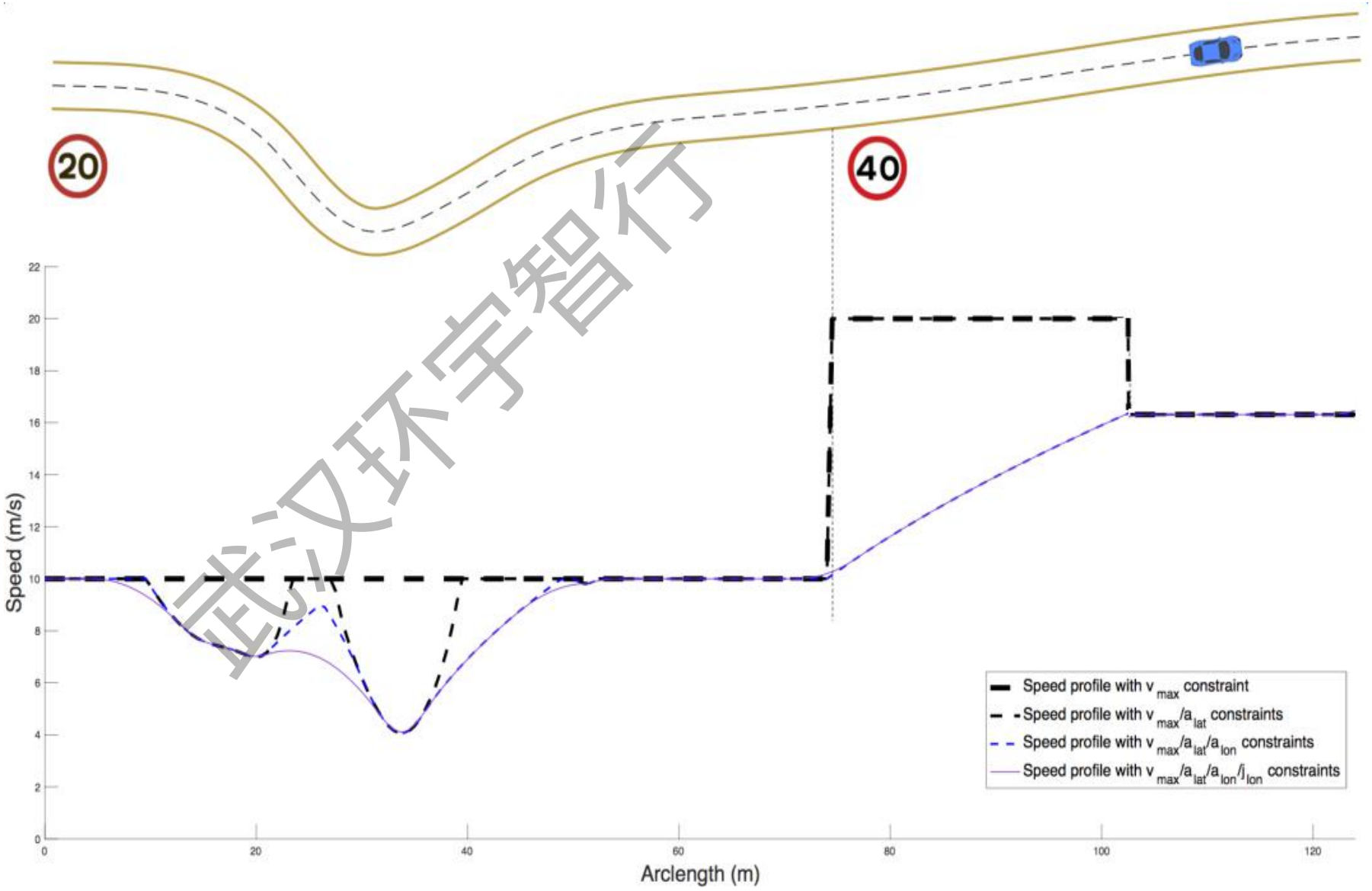
算法原理——局部规划

生成纵向信息：

纵向加速度约束：

考虑到车辆动力学特性，车在加速或者减速过程中，其加速度也有一定的限制：

$$-d_{max}^{lon} \le \frac{v_i^2 - v_{i-1}^2}{2|s_i - s_{i-1}|} \le a_{max}^{lon}$$



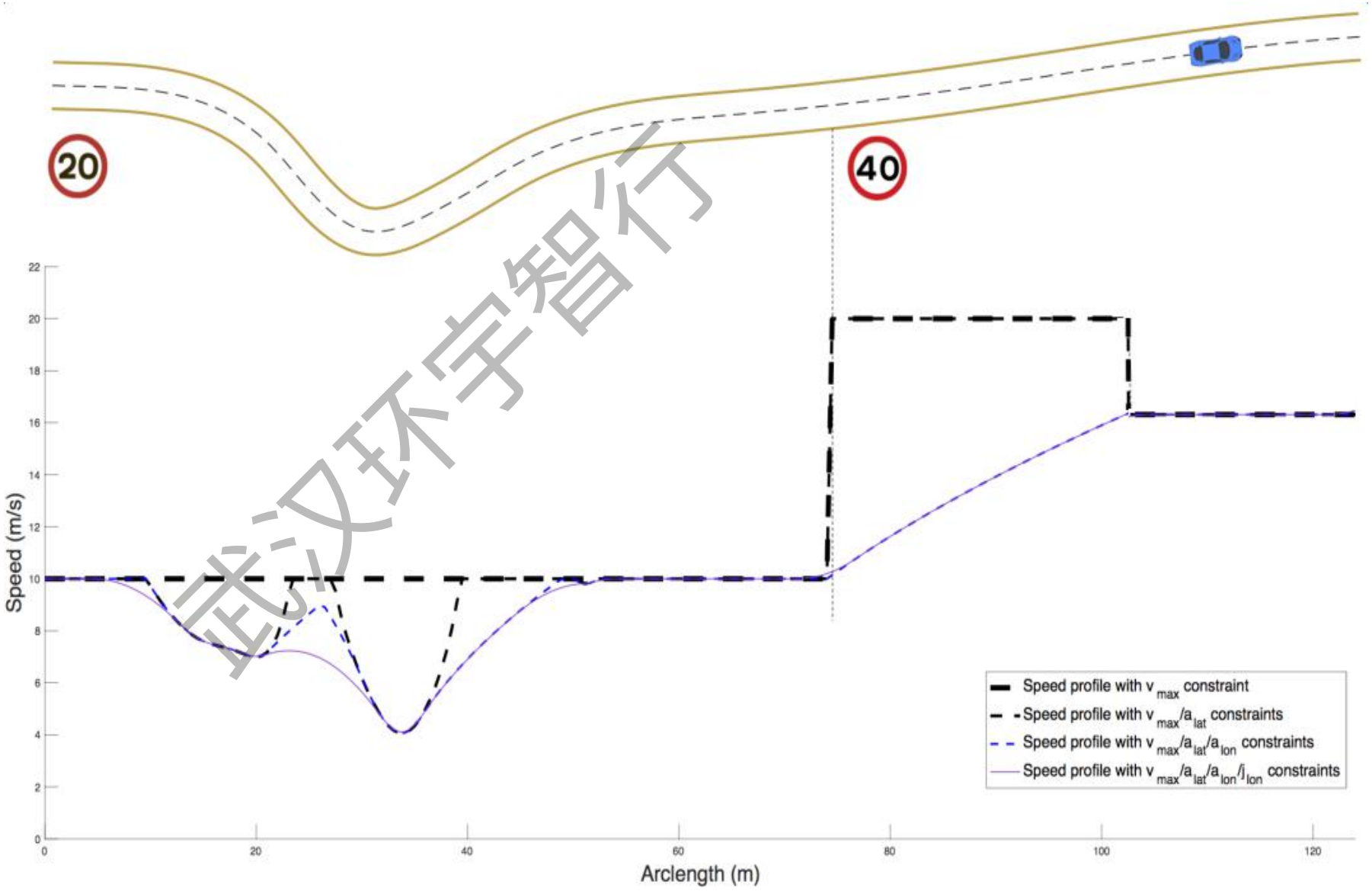
武汉环宇智行

算法原理——局部规划

生成纵向信息：

连续性约束：

考虑以上约束后得到的速度规划曲线可能还不够连续平滑，所以我们用类似于地图中心线平滑处理的方法，对速度曲线也进行三次样条拟合，保证其连续性。

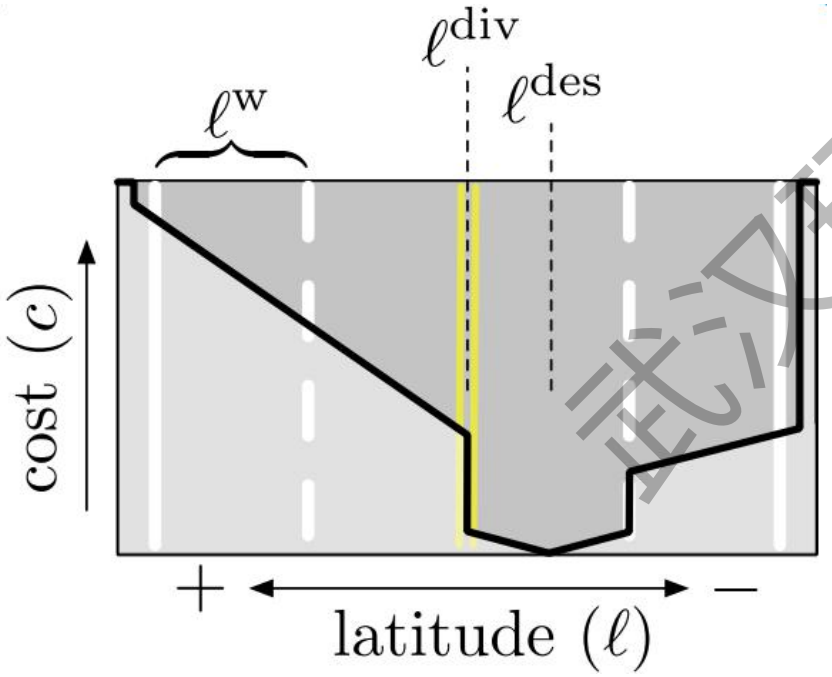


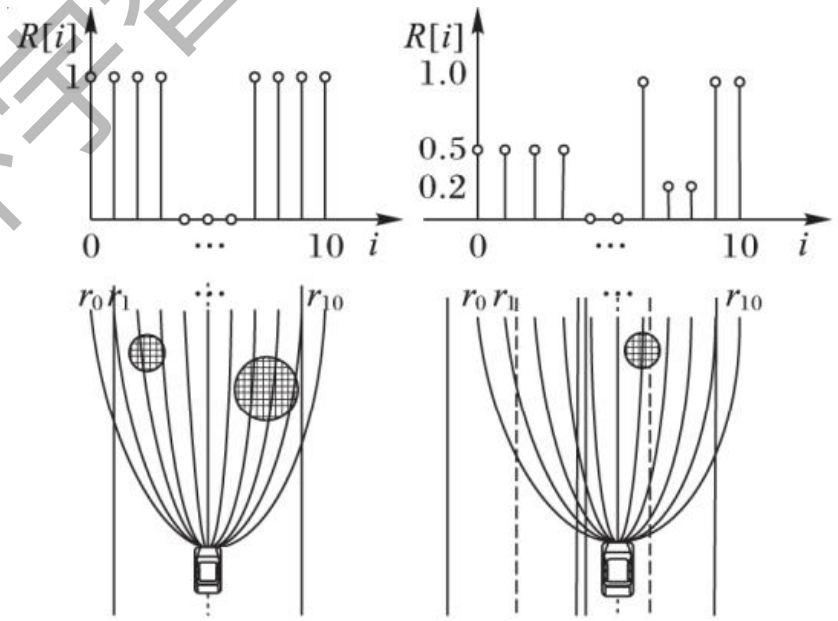
武汉环宇智行

算法原理——局部规划

选择最优轨迹 $c(l) = w_{safe}c_{safe}(l) + w_{smo}c_{smo}(l) + w_{dyn}c_{dyn}(l) + w_{coh}c_{coh}(l) + w_{kin}c_{kin}(l)$

安全性代价函数





(a) 单车道　　(b) 多车道

武汉环宇智行

算法原理——局部规划

选择最优轨迹 $c(l) = w_{safe}c_{safe}(l) + w_{smo}c_{smo}(l) + w_{dyn}c_{dyn}(l) + w_{coh}c_{coh}(l) + w_{kin}c_{kin}(l)$

平滑性代价函数： $c_{smo}(l) = \int k_l^2(s)ds$

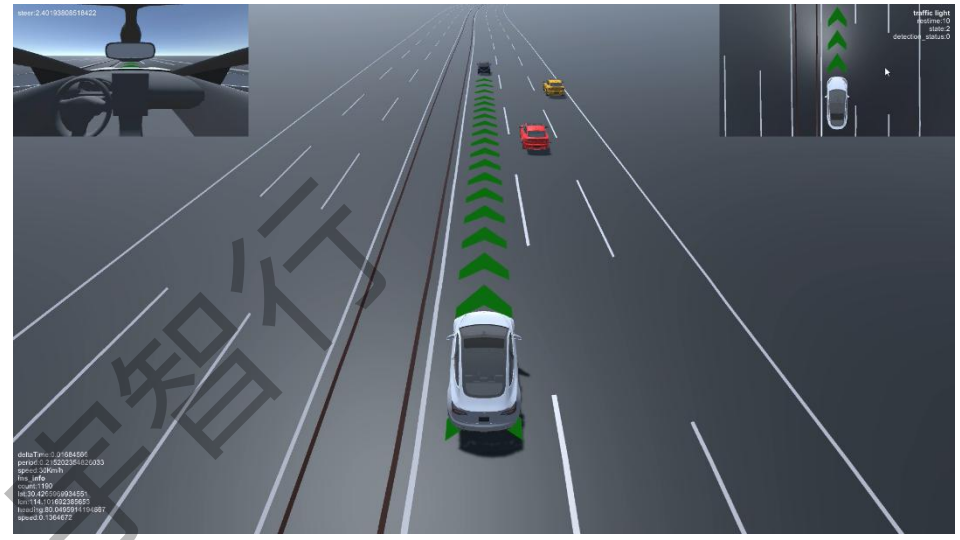
动力学代价函数： $c_{dyn}(l) = \sum_i (|a_l^{lon}| + |a_l^{lat}|)$

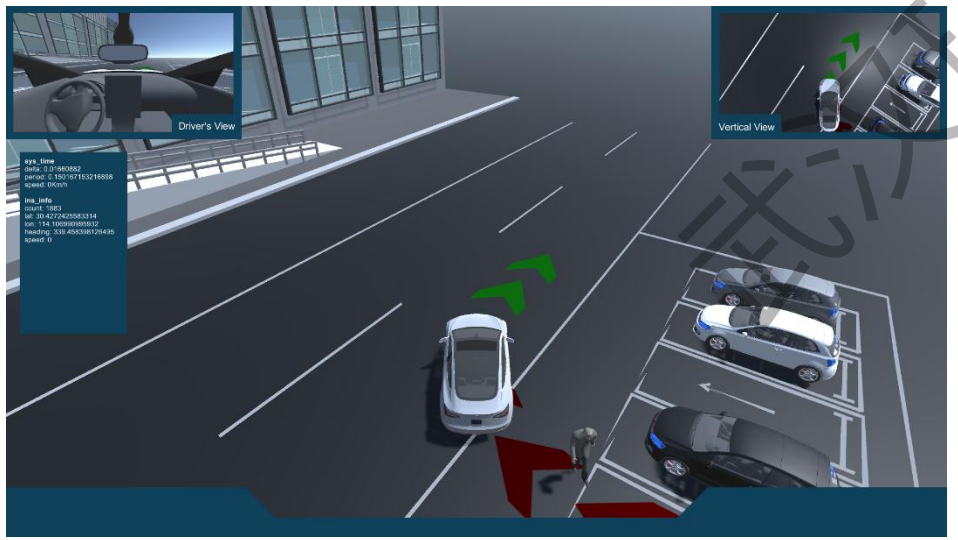
运动学代价函数： $c_{kin}(l) = \begin{cases} 0, & \max(|k_l|) \le \frac{1}{R_{min}} \\ \max(|k_l|) - \frac{1}{R_{min}}, & \max(|k_l|) > \frac{1}{R_{min}} \end{cases}$

连贯性代价函数： $c_{coh}(l) = \frac{1}{s_2 - s_1}\int_{s_1}^{s_2} d_l\, ds$

武汉环宇智行

实际操作——平台介绍





输入： 地图，障碍物，交通信息，定位信息，车身信息等

场景： 站点召车，跟车，超车，路口通行，自动泊车等

输出： 规划轨迹

武汉环宇智行

实际操作——平台介绍

武汉环宇智行

环宇智行
IN-Driving Tech

THANKS FOR WATCHING