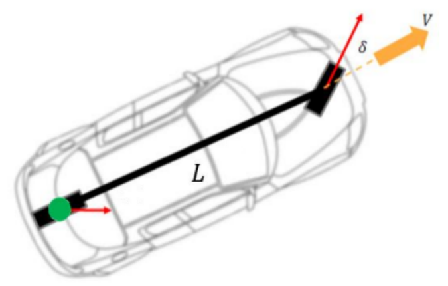


自行车模型

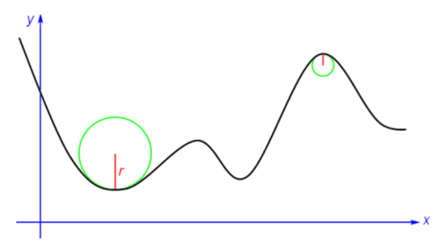
$$\dot{\theta} = \frac{V\tan(\delta)}{L}$$

$$|\kappa| \le \kappa_{max}$$ → 最大曲率

→ 非完整性约束 → 约束不仅取决于机器人的状态 还取决于机器人如何到达当前状态

→ 使规划问题更加复杂

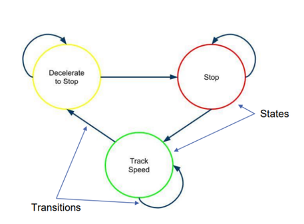
→ 减少了机器人在其工作空间中任意给定点上可以采取的方向数量



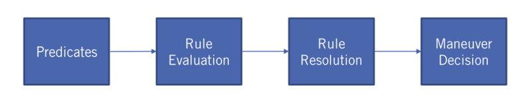
$k = \frac{1}{r}$

$k = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}$

有三种不同的主流架构来解决行为规划问题

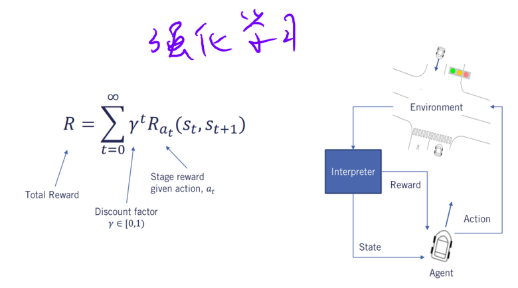


① 有限状态机

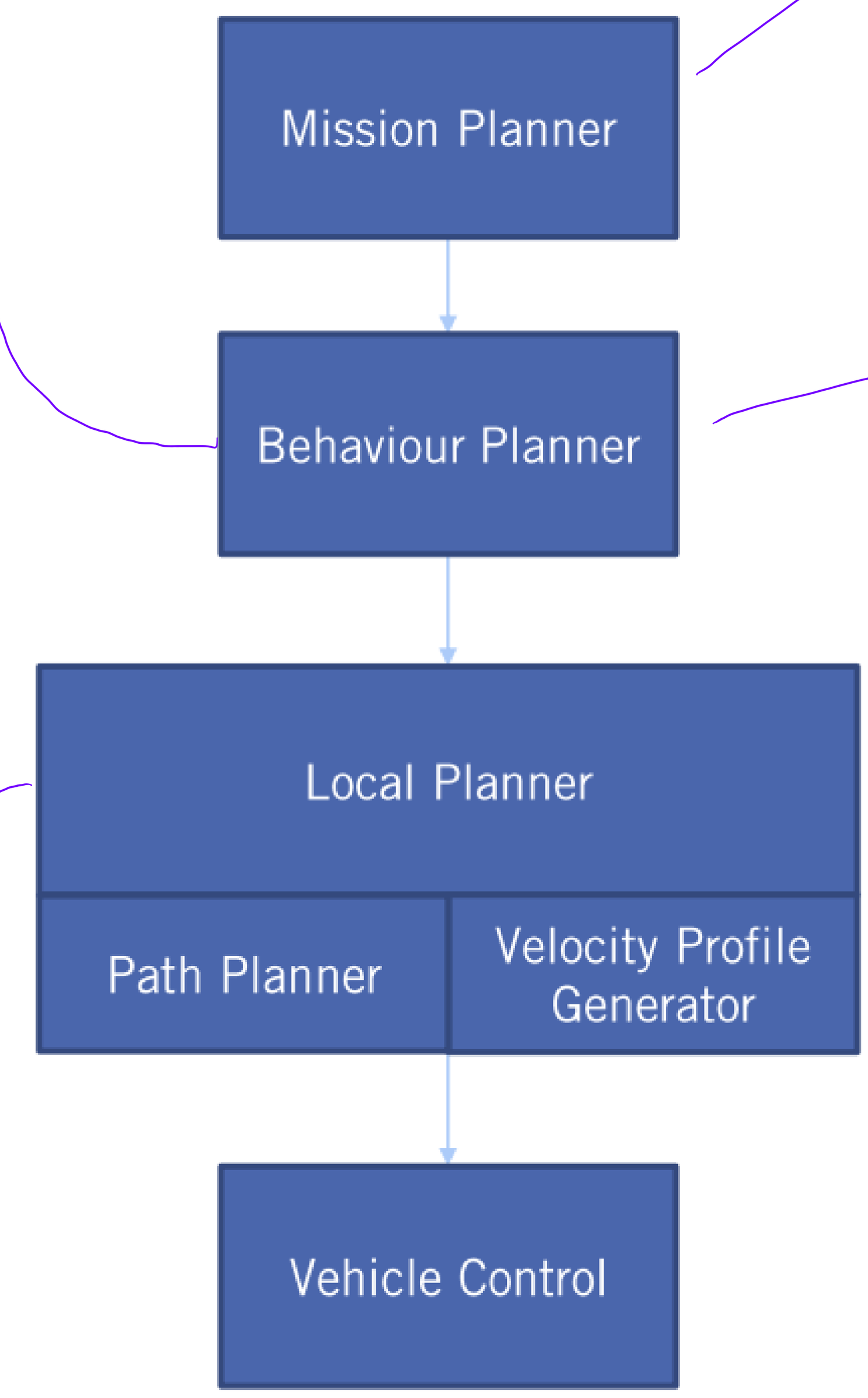


② 基于规则的系统

强化学习



③ 基于机器学习



Mission Planner → 最高级别的优化问题 → 关注的是从车辆当前位置到给定目的地的地图级导航（在约束条件下）

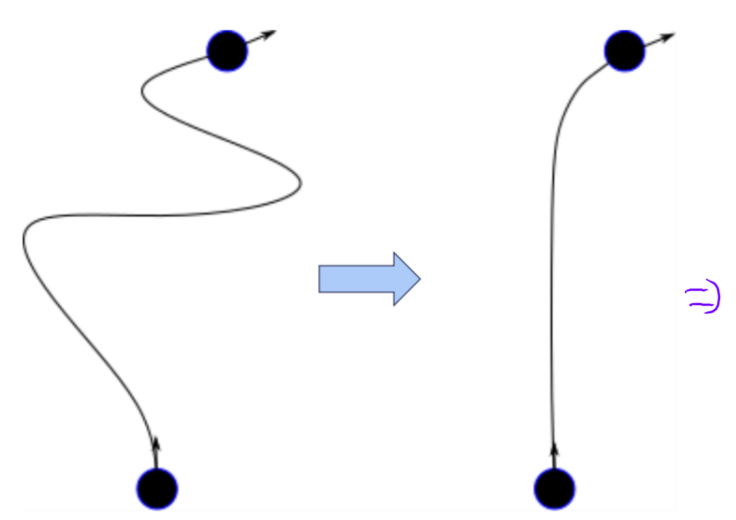
Behaviour Planner → 关注当前的驾驶场景 并根据工作空间的其它 agents 和道路规则决定执行哪种机动

Local Planner → 生成运动上可行的无碰撞的路径 以及舒适的速度剖面

- Path length:
  - Minimize the arc length of a path to generate the shortest path to the goal

$$s_f = \int_{x_i}^{x_f} \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx$$

s_f：累计弧长；x_f：路径的结束 x 坐标；x_i：路径的起始 x 坐标



⇒ 第一条路径比第二条路径的长度更长 到达目的地的效率更低

$S_f = \int_{x_i}^{x_f} \sqrt{1 + (\frac{dy}{dx})^2}\, dx$

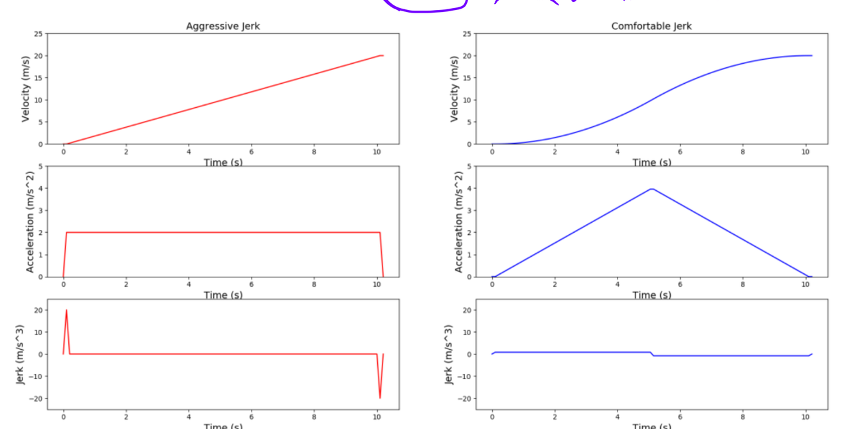
- Travel time:
  - Minimize time to destination while following the planned path

$$T_f = \int_0^{s_f} \frac{1}{v(s)} ds$$

T_f：轮廓的总时间；$v(s)$：纵向速度

$$\int_0^{s_f} \|\dddot{x}(s)\|^2 ds$$

Jerk ⇒ 变化的速度的加速度（关于时间或位置的三阶导数）

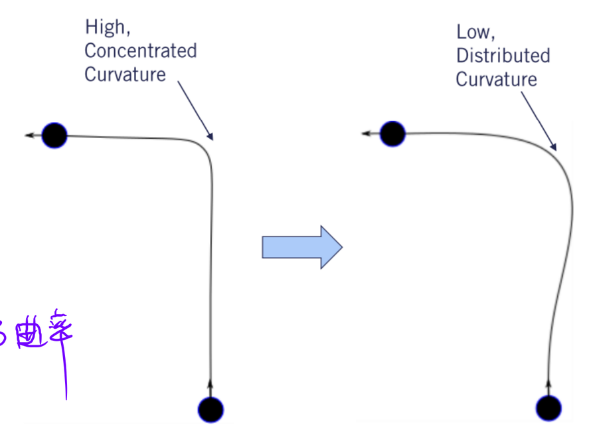


为了确保我们避开路径上的高曲率点 需要对大的绝对曲率值制定某种惩罚

用来表示这种惩罚的目标函数 称为路径的弯曲能量

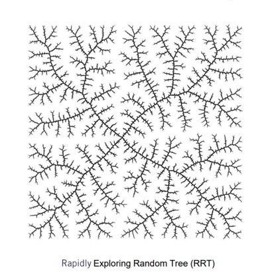
$$\int_0^{s_f} \|\kappa(s)\|^2 ds$$

沿路径积分的平方曲率

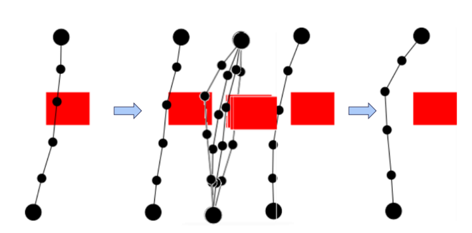


好的路径规化算关键在于减少优化的搜索基数

路径规划有三大类

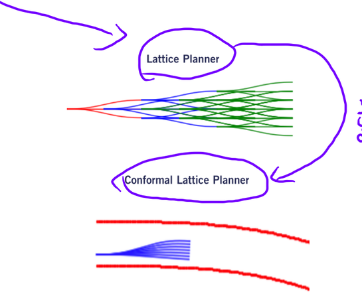


① 基于采样的规划



② 变分规划

→ 依靠微积分变化来优化轨迹函数（chomp算法）



③ 网格规划

- Smoothness

$$\int_0^{s_f} \|\dddot{x}(s)\| ds$$

- Deviation from reference

$$\int_0^{s_f} \|v(s) - v_{ref}(s)\| ds$$

- Lateral acceleration limit

$$v^2 \le \frac{a_{lat_{max}}}{\kappa}$$