

Apollo自动驾驶入门课程第⑧讲 — 规划（下）



第八课，规划（下）

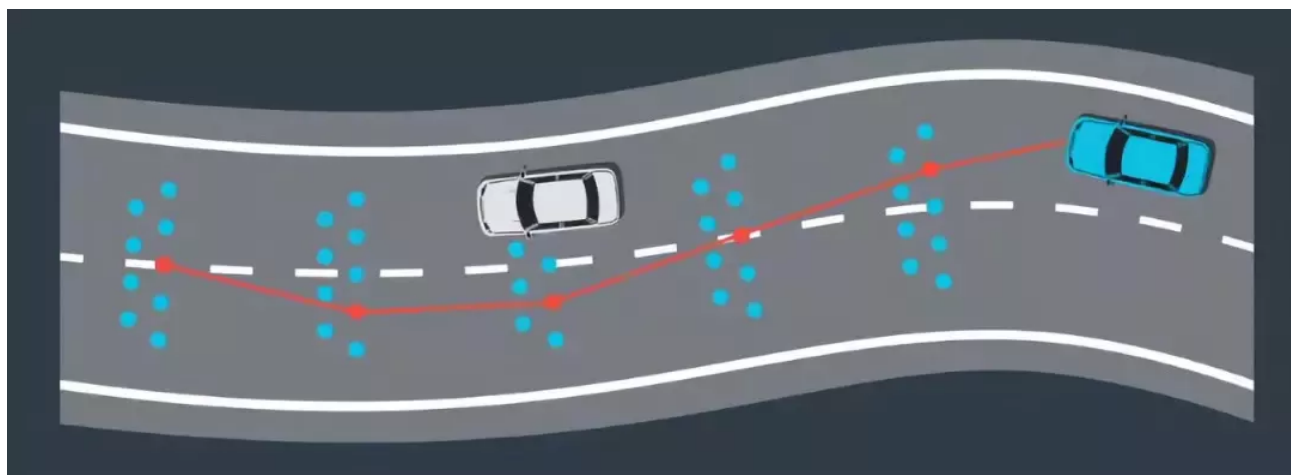
1 | 路径-速度解耦规划

路径-速度解耦规划将轨迹规划分为两步：**路径规划**、**速度规划**。首先在路径规划步骤中生成候选曲线，这是车辆可行驶的路径。使用成本函数对每条路径进行评估，该函数包含平滑度、安全性、与车道中心的偏离以及开发者想要考虑的其他任何因素。然后按成本对路径进行排名并选择成本最低的路径。

下一步是确定沿这条路线行进的速度。我们可能希望改变在该路径上的速度，所以真正需要选择的是与路径点相关的一系列速度，而不是单个速度。我们将该序列称作“速度曲线”。我们可以使用优化功能为路径选择受到各种限制的良好速度曲线。通过将路径和速度曲线相结合可构建车辆行驶轨迹。

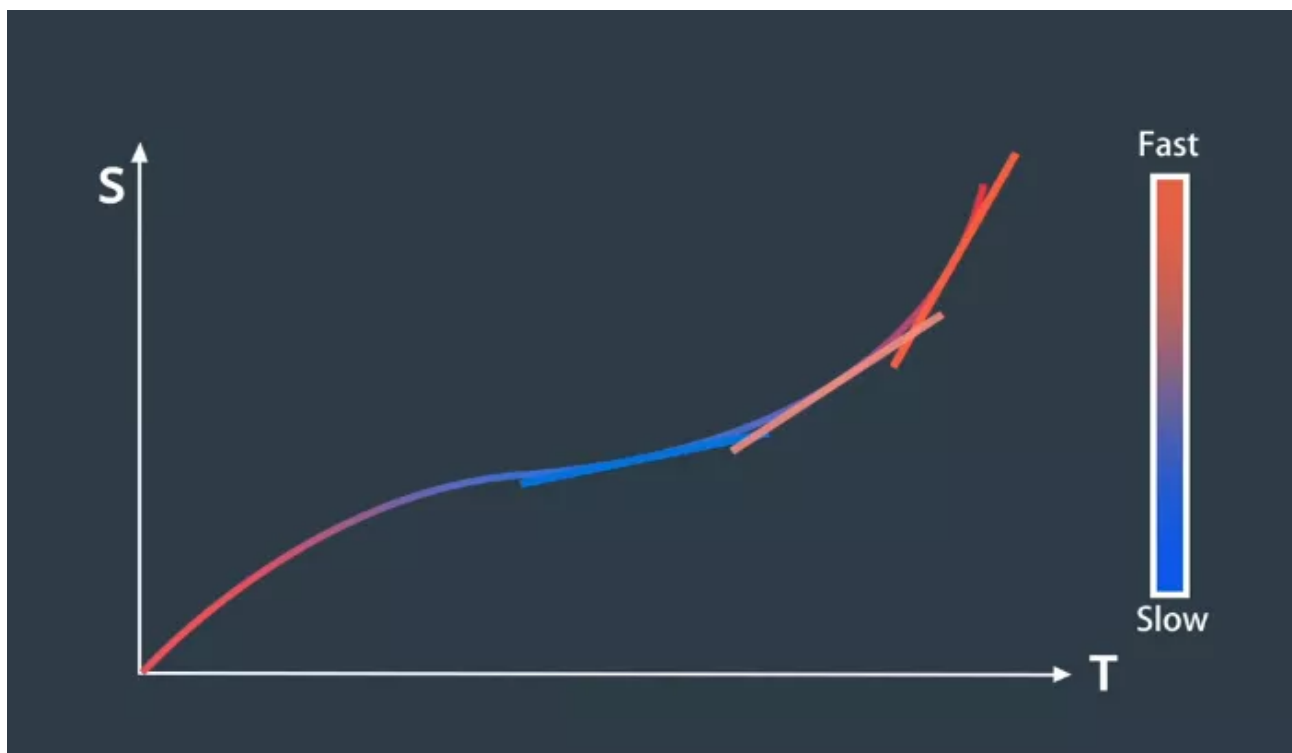
2 | 路径生成与选择

为了在**路径-速度解耦规划**中生成候选路径，首先将路段分割成单元格。然后对这些单元格中的点进行随机采样。通过从每个单元格中取一个点并将点连接，我们创建了候选路径。通过重复此过程可以构建多个候选路径。使用成本函数对这些路径进行评估并选择成本最低的路径，成本函数可能考虑以下因素：与车道中心的偏离、与障碍物的距离、速度和曲率的变化、对车辆的压力、或希望列入的任何其他因素。



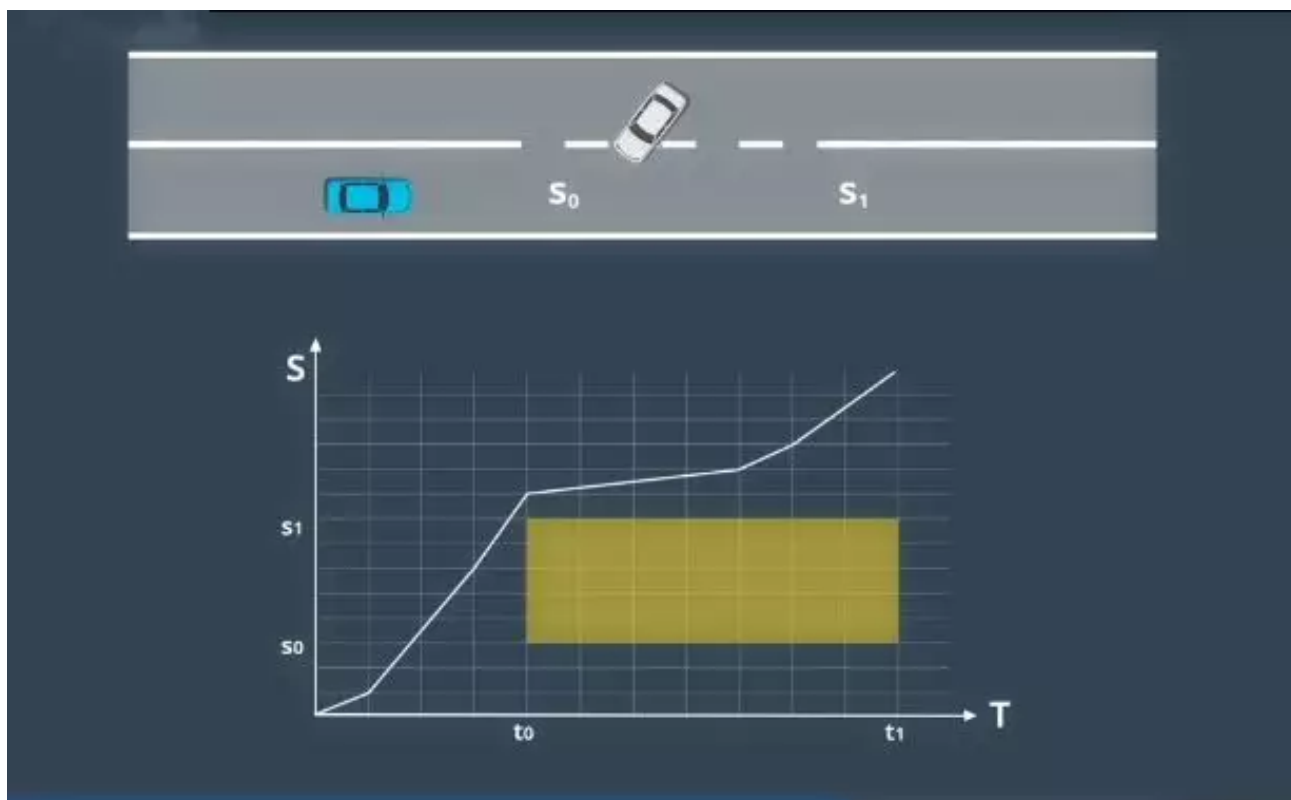
3 | ST图

选择路径后的下一步是选择与该路径关联的速度曲线，一个被称为“ST 图”的工具可以帮助设计和选择速度曲线。在ST图中，“s”表示车辆的纵向位移、“t”表示时间。ST 图上的曲线是对车辆运动的描述，因为它说明了车辆在不同时间的位置。由于速度是位置变化的速率，所以可以通过查看曲线的斜率从 ST 图上推断速度。斜坡越陡则表示在更短的时间段内有更大的移动，对应更快的速度。



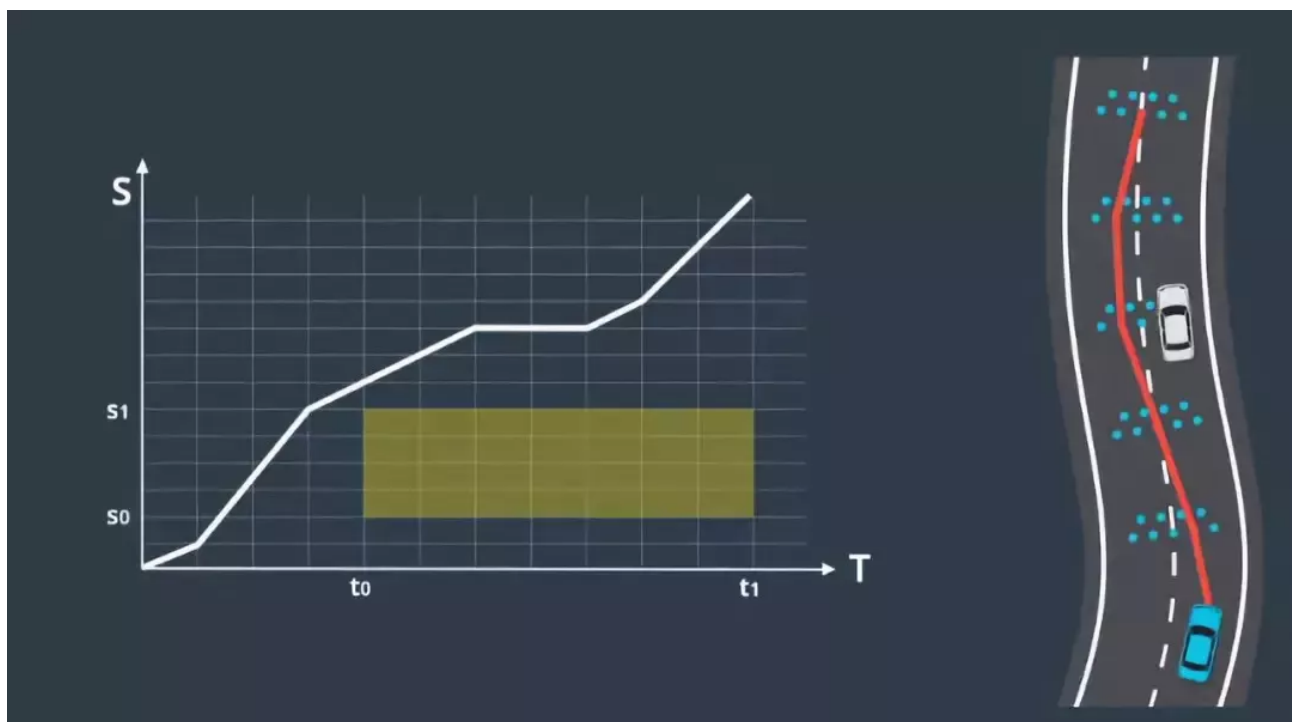
4 | 速度规划

为构建最佳速度曲线需要将 ST 图离散为多个单元格。单元格之间的速度有所变化，但在每个单元格内速度保持不变，该方法可简化速度曲线的构建并维持曲线的近似度。在 ST 图中可以将障碍物绘制为在特定时间段内阻挡道路的某些部分的矩形。例如，假设预测模块预测车辆将在 t_0 到 t_1 的时间段内驶入的车道。由于该车将在此期间占据位置 s_0 到 s_1 ，因此在 ST 图上绘制了一个矩形，它将在时间段 t_0 到 t_1 期间阻挡位置 s_0 到 s_1 。为避免碰撞，速度曲线不得与此矩形相交。既然有了一张各种单元格被阻挡的 ST 图，便可以使用优化引擎为该图选择最佳的速度曲线。优化算法通过复杂的数学运算来搜索受到各种限制的低成本解决方案。这些限制可能包括：法律限制，如速度限制；距离限制，如与障碍物的距离；汽车的物理限制，如加速度限制。



5 | 优化

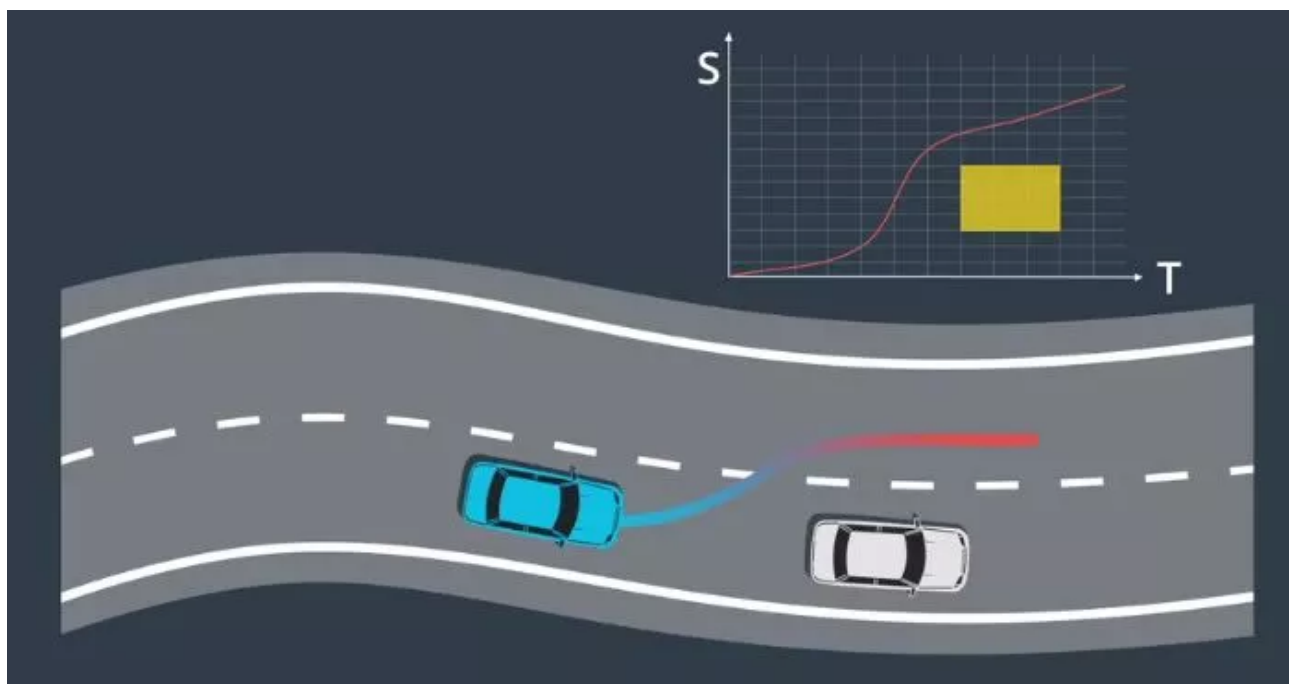
路径-速度解耦规划在很大程度上取决于离散化。路径选择涉及将道路划分为单元格，速度曲线构建涉及将 ST 图划分为单元格。尽管离散化使这些问题更容易解决，但该解决方案生成的轨迹并不平滑。



为了将离散解决方案转换为平滑轨迹，可使用“二次规划”技术（Quadratic Programming）。二次规划将平滑的非线性曲线与这些分段式线性段拟合。尽管二次规划背后的数学运算很复杂，但对于我们的目的而言，细节并不是必需的。我们只需简单使用几种不同的优化包中的一种，包括一种由 Apollo 推出的运行方案来生成平滑的轨迹，一旦路径和速度曲线就绪，便可以用其构建三维轨迹。

6 | 路径-速度规划的轨迹生成

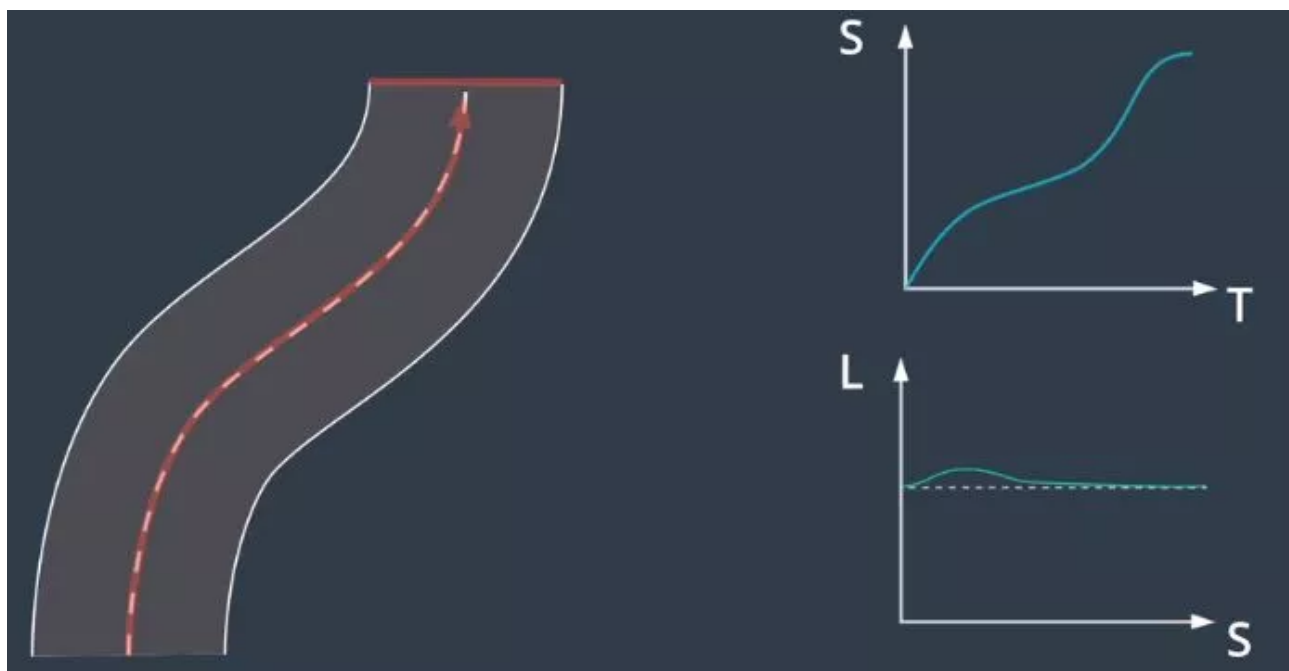
回顾一下**端到端路径-速度解耦规划**。假设我们正在路上行驶，感知系统观察到一辆缓慢行驶的车辆离我们越来越近。首先，在这辆车的周围生成多条候选路线，使用成本函数对这些候选路径进行评估并选择成本最低的路径。然后使用 ST 图来进行速度规划，根据其他车辆随时间变化的位置阻挡了 ST 图的部分区域。优化引擎可帮助确定该图的最佳速度曲线，该曲线受制于约束和成本函数。我们可以使用二次规划让路径和速度曲线变平滑。最后，将路径和速度曲线合并构建轨迹。这里的轨迹在速度较快时为红色，在速度较慢时为蓝色。我们使用该轨迹来安全地避开其他车辆并继续我们的旅程。



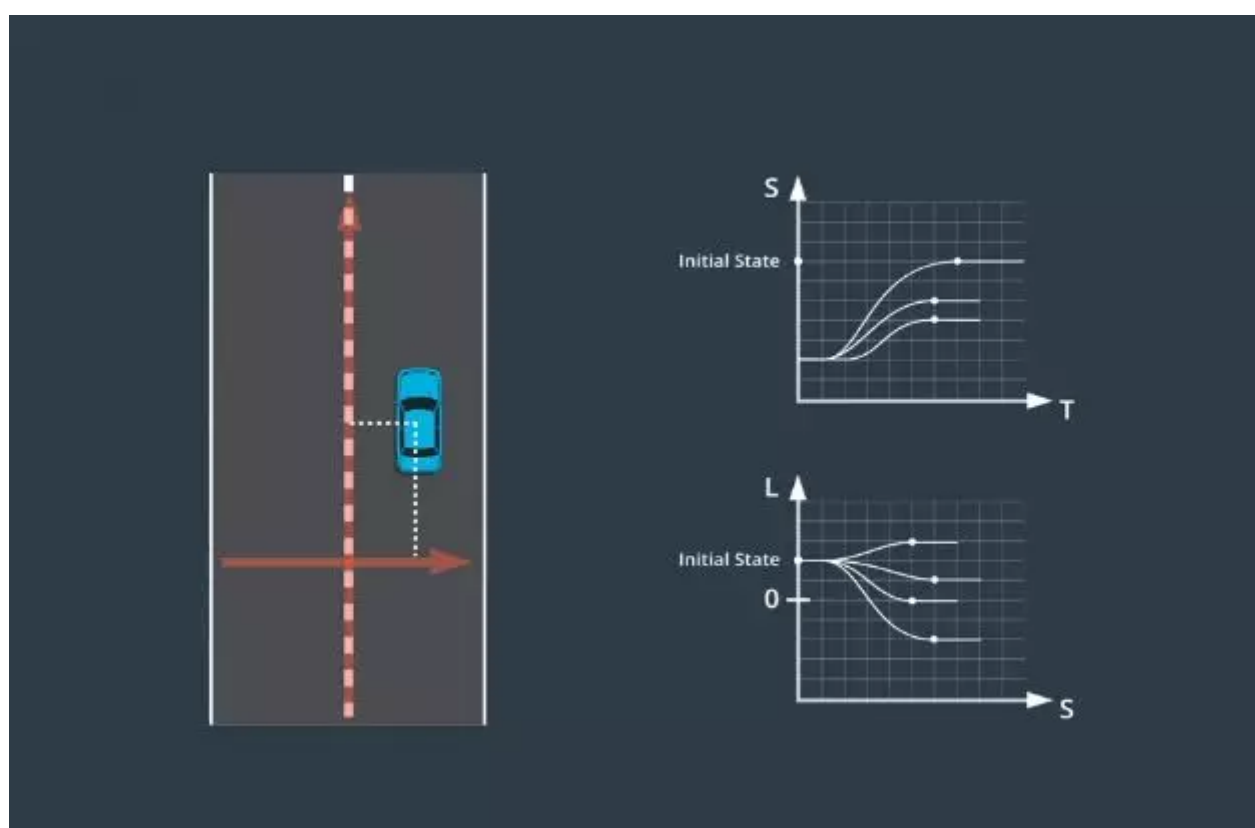
7 | Lattice 规划

现在来学习一种称为 **Lattice 规划** 的轨迹生成方法，通过使用 Frenet 坐标可以将环境投射到纵轴和横轴上，目标是生成三维轨迹：纵向维度、横向维度、时间维度。

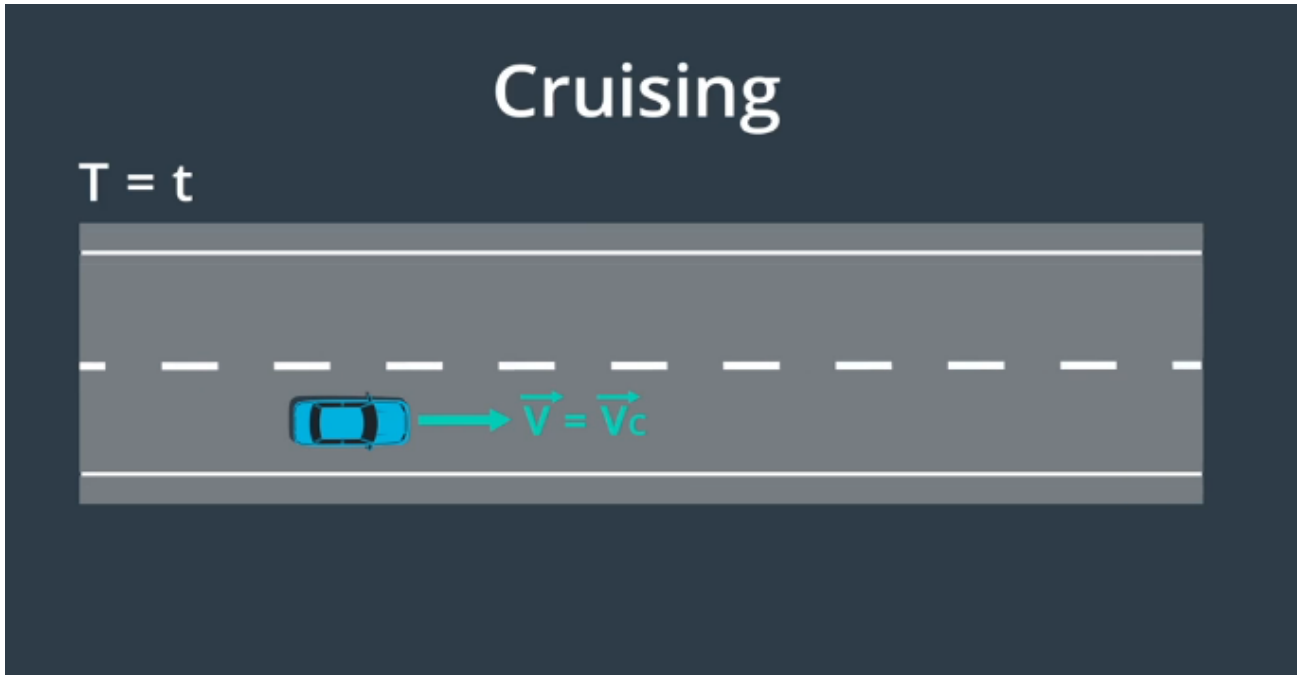
可以将三维问题分解成两个单独的二维问题，这是通过分离轨迹的纵向和横向分量来解决的。其中一个二维轨迹是具有时间戳的纵向轨迹称之为 ST 轨迹，另一个二维轨迹是相对于纵向轨迹的横向偏移称之为 SL 轨迹。



Lattice 规划具有**两个步骤**即先分别建立 ST 和 SL 轨迹，然后将它们合并为生成纵向和横向二维轨迹。先将初始车辆状态投射到 ST 坐标系和 SL 坐标系中，通过对预选模式中的多个候选最终状态进行采样。来选择最终车辆状态。对于每个候选最终状态构建了一组轨迹将车辆从其初始状态转换为最终状态，使用成本函数对这些轨迹进行评估并选择成本最低的轨迹。



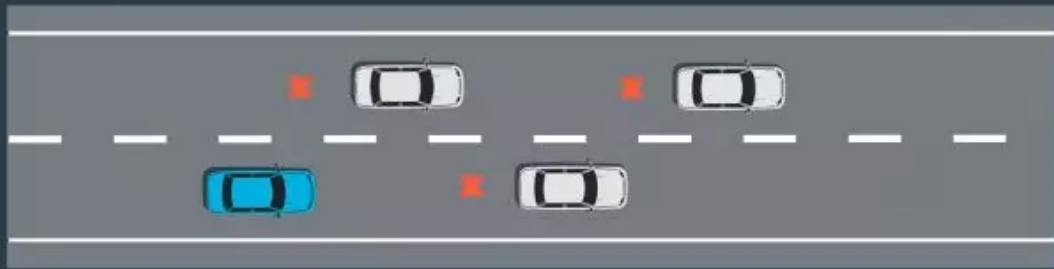
根据情况可以将状态分成 3 组：**巡航**、**跟随**、**停止**。巡航意味着车辆将在完成规划步骤后定速行驶，实际上在对图上的点进行采样，在图中横轴代表时间，纵轴代表速度。对于该图上的点，这意味着汽车将进入巡航状态，在时间 t 以 s 点的速度巡航，对于这种模式，所有最终状态的加速度均为零。



下一个要考虑的模式为跟随车辆，在这种情况下要对位置和时间状态进行采样，并尝试在时间 t 出现在某辆车后面，在跟随车辆时，需要与前方的车保持安全距离，这时速度和加速度将取决于要跟随的车辆，这意味着在这种模式下，速度和加速度都会进行修正。

Following a Car

$T = t$



✗ Ending Positions

最后一种模式是停止，对于这种模式只需对汽车何时何地停止进行抽样，这里速度和加速度会被修正为 0。

Stopping at a Point

$T = t$

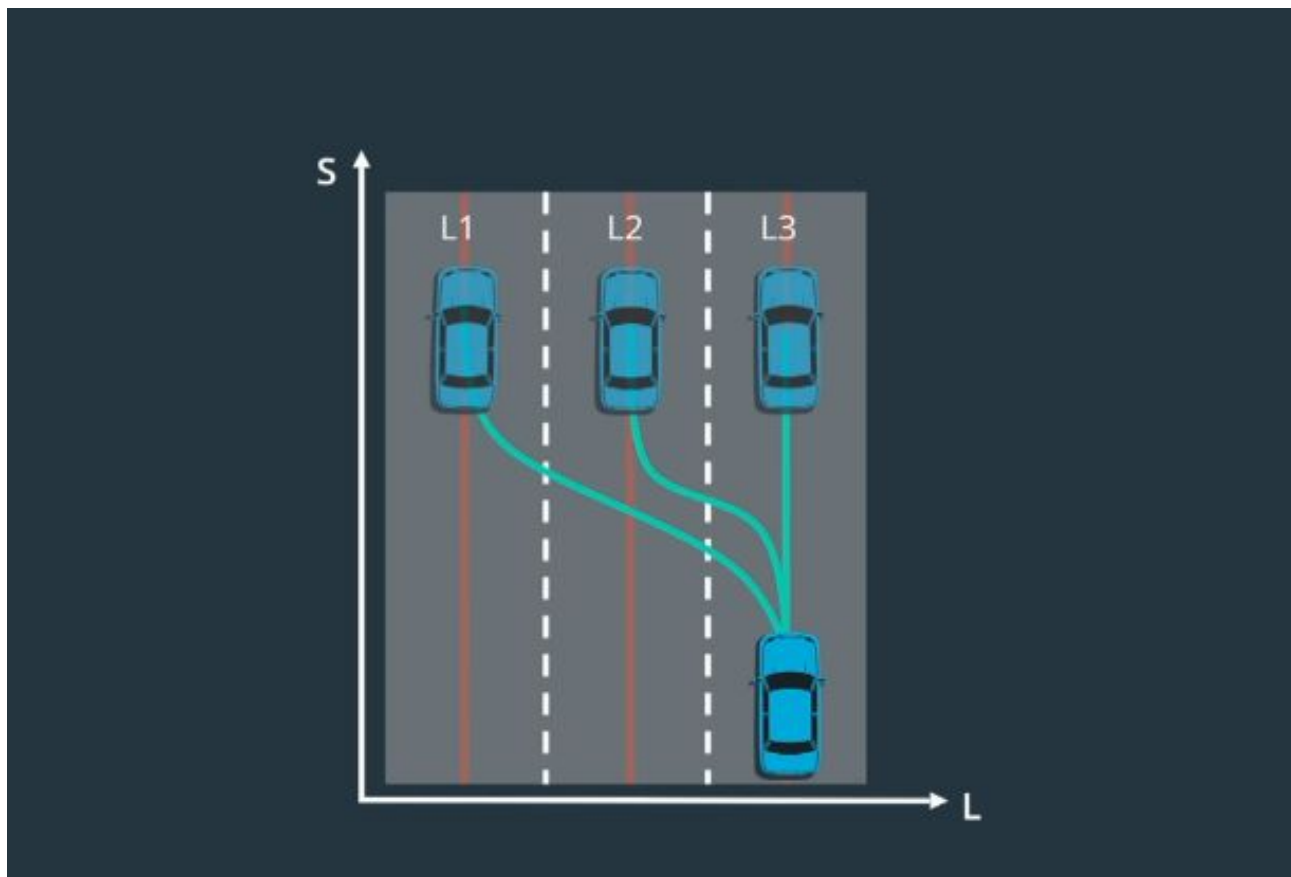


✗ Ending Positions

9 | SL轨迹的终止状态

根据这样一个假设来进行SL 规划，即无论车辆进入怎样的终止状态，车辆都应该稳定地与车道中心线对齐。这意味着只需要在一个小区域内，对横向终止位置进行采样。具体来说采样的是道路上相邻车道

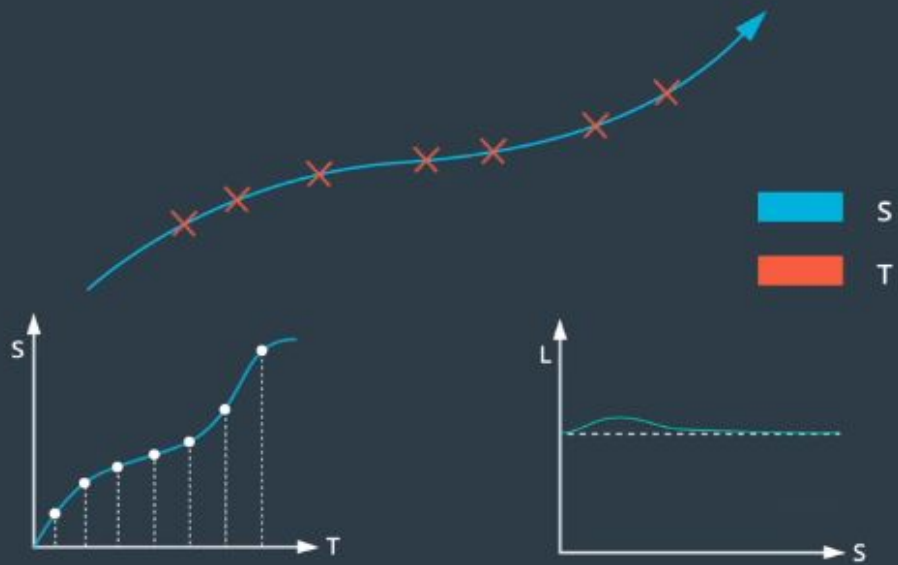
中心线周围的位置。为了确保稳定性，汽车驶向的终止状态应该与车道中心一致。当用横向位置与纵向位置作图时，想要的候选轨迹应该以车辆与车道对齐并直线行驶而结束。为了达到这种终止状态，车的朝向和位置的一阶和二阶导数都应该为零。这意味着车辆既不是横向移动的，那是一阶导数；也不是横向加速，那是二阶导数。这意味着车辆正沿着车道直行。



10 | Lattice规划的轨迹生成

一旦同时拥有了 ST 和 SL 轨迹，就需要将它们重新转换为笛卡尔坐标系。然后可以将它们相结合构建由二维路径点和一维时间戳组成的三维轨迹。ST 轨迹是随时间变化的纵向位移，SL 轨迹是纵向轨迹上每个点的横向偏移。由于两个轨迹都有纵坐标 S，所以可以通过将其 S 值进行匹配来合并轨迹。

Trajectory Generation



Apollo
