

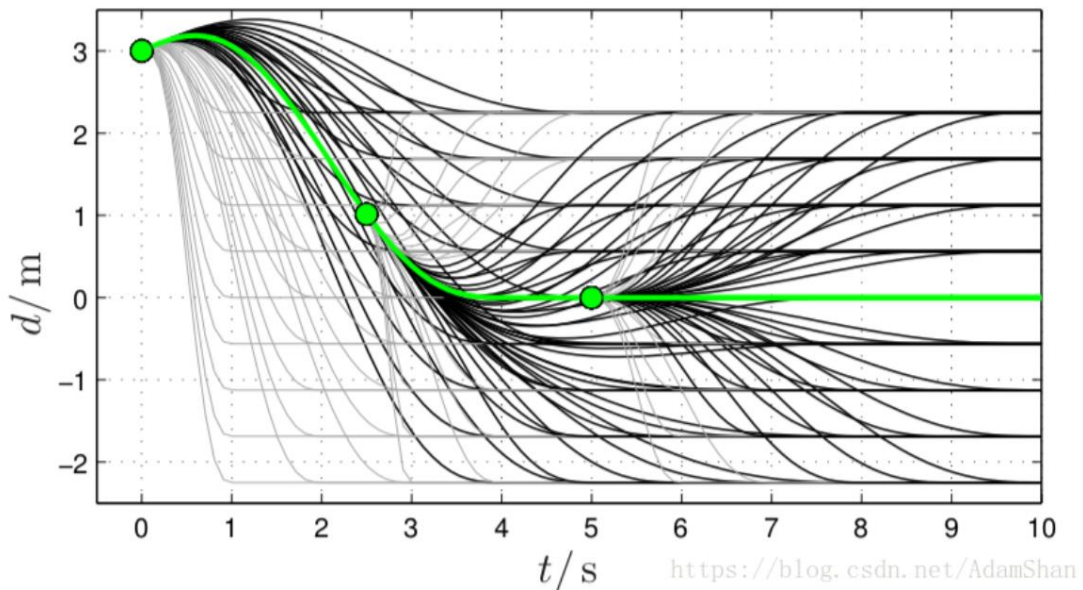
第六章作业 LQR 控制

第六章作业是使用 Lattice 的方法实现横向的曲线规划和纵向的速度规划，主要是将代码 `frenet_optimal_trajectory` 中 To-Do 的内容编写完成

课件和视频对于具体的实施描述的不是非常充分，可以参考下方的文章

[\(100 条消息\) 无人驾驶汽车系统入门（二十一）——基于 Frenet 优化轨迹的无人车动作规划方法_AdamShan 的博客-CSDN 博客](#)

采样的基本逻辑是在一定宽度和时间内去拟合多项式曲线，如下图，这个确定了终止条件。有了初始和终止条件结合课件以及上面的博文就可以编写采样的代码，主要的代码就是遍历不同终止条件，然后生成一系列多项式曲线。最后再结合约束和碰撞条件去筛选出来合适的曲线。



另外生成的曲线是在 Frenet 坐标下，需要转换到笛卡尔坐标系才能给到车辆使用，所以有坐标转换的过程。需要注意的是，在转换的时候可以依据下式直译去转换，比如知道 θ_x 了可以依据相邻两个时刻的值去计算曲率，而不需要严格按照曲率复杂的对应公式（下方第 2 个曲率公式）去转换

Cartesian coordinate: $[\vec{x}, v_x, a_x, \theta_x, \kappa_x]$

\vec{x} : a position vector in Cartesian coordinate

$v_x = \|\dot{\vec{x}}\|_2$: speed in Cartesian coordinate

$a_x = \frac{dv_x}{dt}$: acceleration in Cartesian coordinate

θ_x : heading in Cartesian coordinate

$\kappa_x = \frac{d\theta_x}{ds}$: curvature

$$k_x = \left((l'' + (k_r' l + k_r l') \tan(\theta_x - \theta_r)) \frac{\cos^2(\theta_x - \theta_r)}{1 - k_r l} + k_r \right) \frac{\cos(\theta_x - \theta_r)}{1 - k_r l}$$

编写完成代码之后，使用 ROS 的 catkin_make 编译，编译完成没有错误之后，

使用作业视频里面的命令运行，可能要加一行 source devel/setup.bash

注意要在代码的工作空间目录运行

最终运行的效果参考附件视频，车辆会沿着计算出来的轨迹行驶，并且在距离
终点位置附近停止。