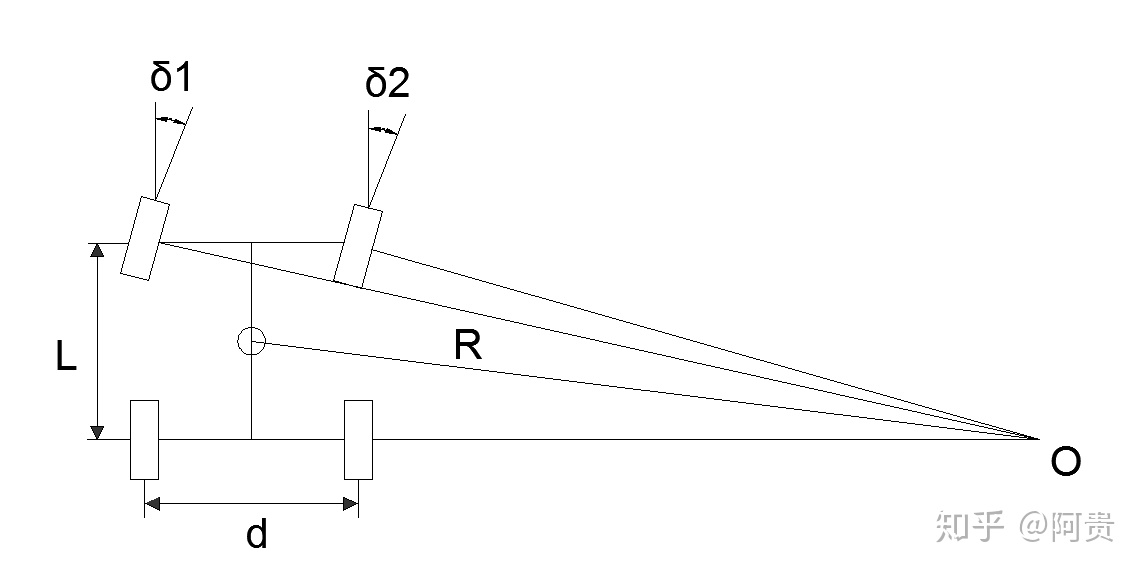
Pure Pursuit In Carla

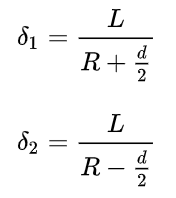
## 原理

1. 阿克曼转向几何

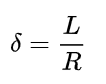
低速时（如在停车场的操控）轮胎不需要产生侧向力，在这种情况下，轮胎滚动没有侧偏角，此时车辆必须如下图所示进行转向。



对转向时（假定小转向角）的正确几何关系，可以得出转向角为：

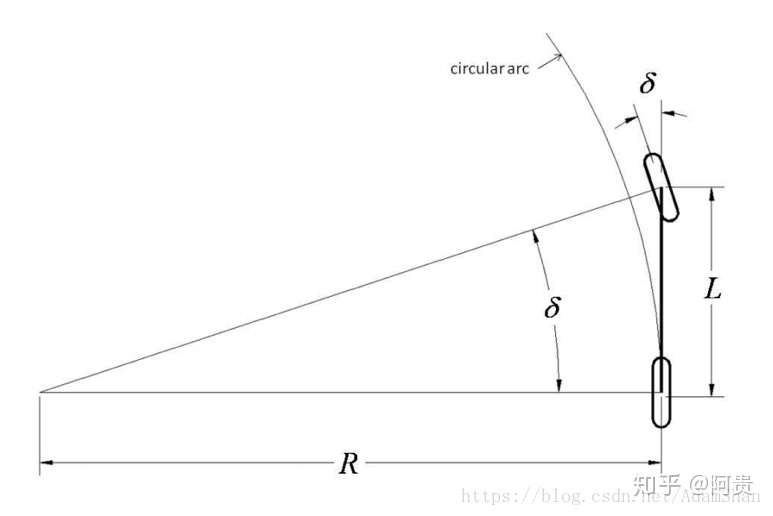


则前轮的平均转向角（假定小转向角）定义为阿克曼角：

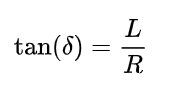


术语“阿克曼转向”或“阿克曼几何”通常用来表示如上图所示的前轮的精确几何关系。

1. 车辆单轨模型（即自行车模型）



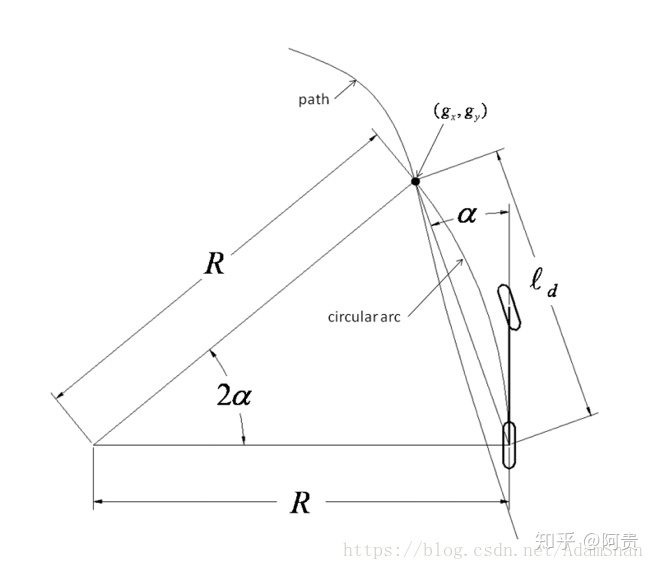
自行车模型实际上是对阿克曼转向几何的一个简化，我们知道，自行车模型将4轮车辆简化成2轮的模型，并且假定车龄只在平面上行驶，采用自行车模型的一大好处就在于它简化了前轮转向角与后轴将遵循的曲率之间的几何关系，其关系如下式所示：



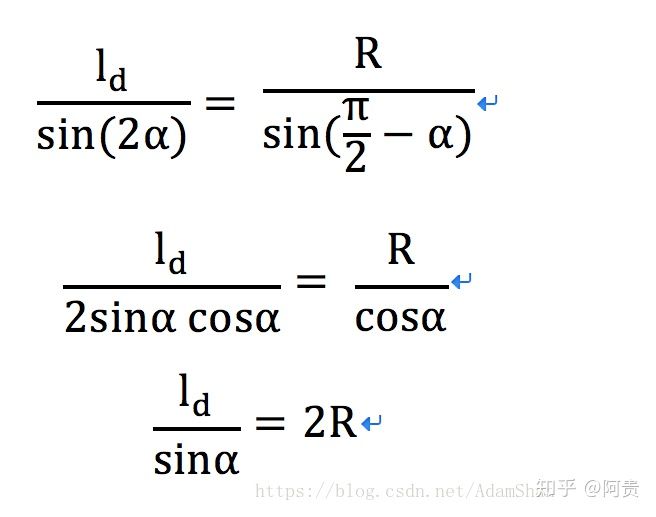
其中δ表示前轮的转角，L为轴距（Wheelbase），R则为在给定的转向角下后轴遵循着的圆的半径。这个公式能够在较低速度的场景下对车辆运动做估计。

1. 纯跟踪算法

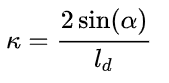
从自行车模型出发，纯跟踪算法以车后轴为切点, 车辆纵向车身为切线, 通过控制前轮转角 ， 使车辆可以沿着一条经过目标路点（goal point）的圆弧行驶，如下图所示：

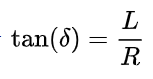


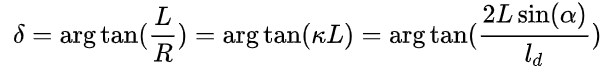
图中是我们下一个要追踪的路点，它位于我们已经规划好的全局路径上，现在需要控制车辆的后轴经过该路点，表示车辆当前位置（即后轴位置）到目标路点的距离，表示目前车身姿态和目标路点的夹角，那么根据正弦定理我们可以推导出如下转换式：



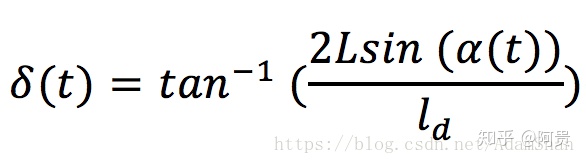
因为道路的曲率，上式也可以表示为：



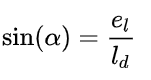
则由式子，可得



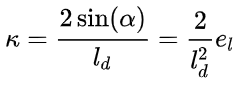
再把时间这个变量加进来，得到



这里我们把时间考虑进来，在知道t时刻车身和目标路点的夹角 α(t) 和距离目标路点的前视距离  的情况下，由于车辆轴距 L 固定，我们可以利用上式估计出应该作出的前轮转角 δ ，为了更好的理解纯追踪控制器的原理，我们定义一个新的量：—— 车辆当前姿态和目标路点在横向上的误差，由此可得夹角正弦：

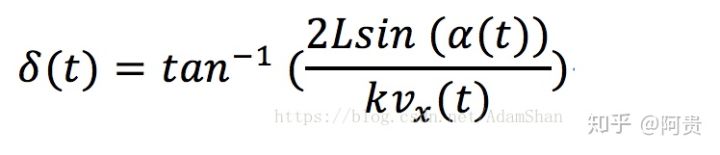


则曲率可以表示为：



考虑到本质是横向上的误差，由上式可知纯追踪控制器其实是一个横向转角的P控制器，其P系数为，这个P控制器受到参数（即前视距离）的影响很大，如何调整前视距离变成纯追踪算法的关键，通常来说，被认为是车速的函数，在不同的车速下需要选择不同的前视距离。

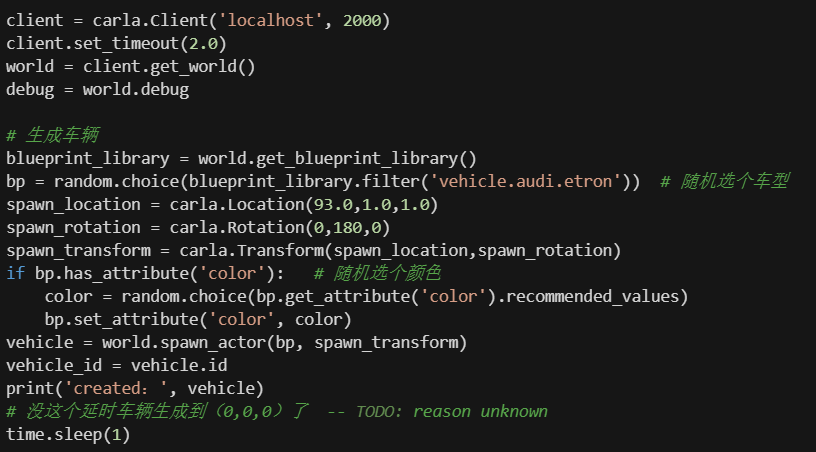
一种最常见的调整前视距离的方法就是将前视距离表示成车辆纵向速度的线形函数，即 ，那么前轮的转角公式就变成了：



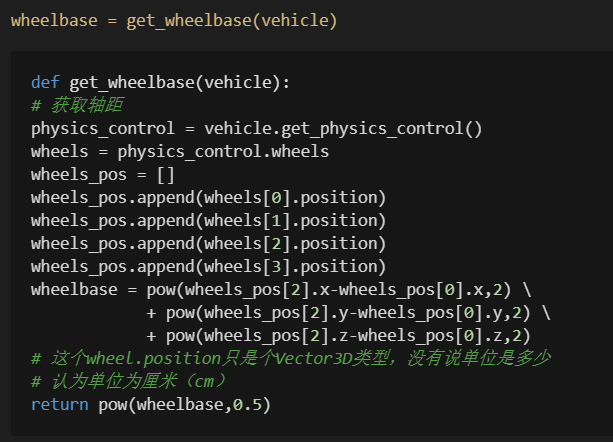
那么纯追踪控制器的调整就变成了调整系数k，通常来说，会使用最大，最小前视距离来约束前视距离，越大的前视距离意味着轨迹的追踪越平滑，小的前视距离会使得追踪更加精确（当然也会带来控制的震荡）。

## 实现流程

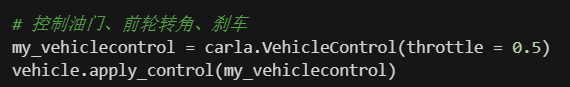
1. 先决条件
   1. 连接Carla并生成车辆



* 1. 获取车辆轴距信息

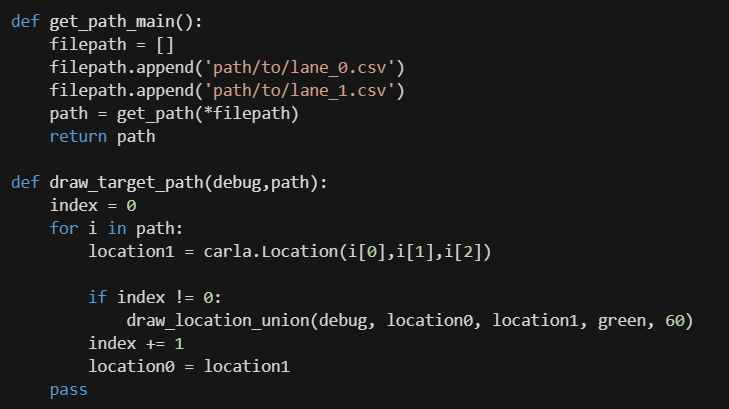
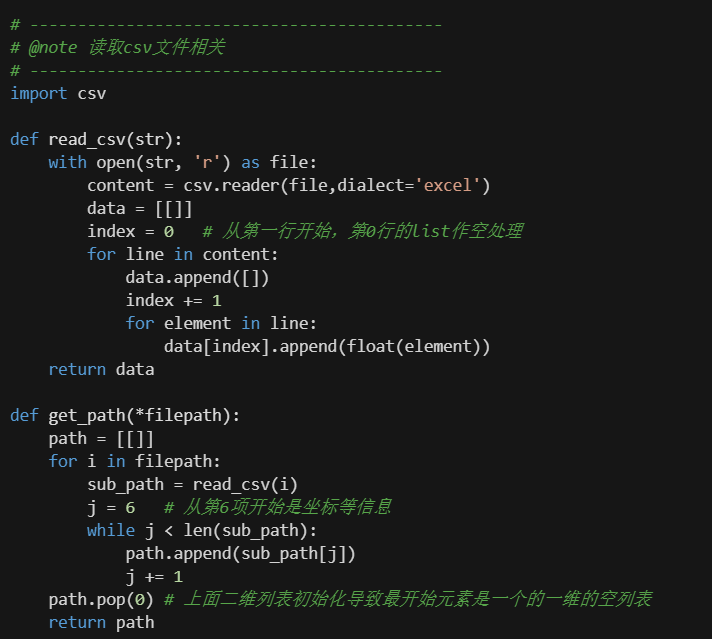


* 1. 给定油门以维持车速

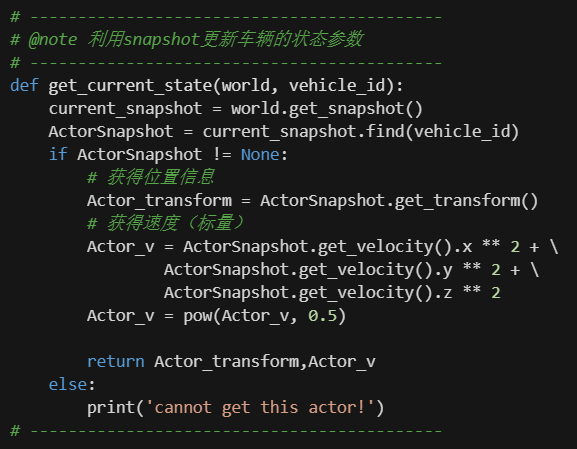


1. 读取csv文件中的waypoint,适配自定义的地图格式类型：

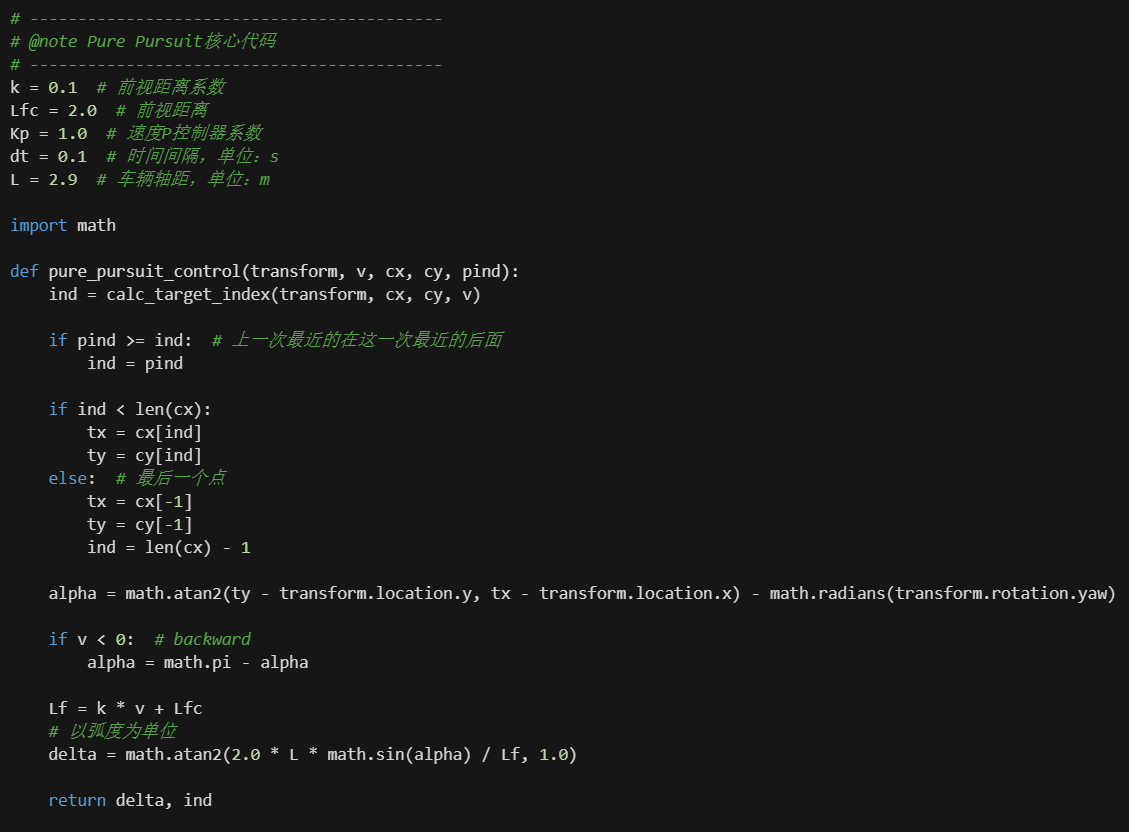


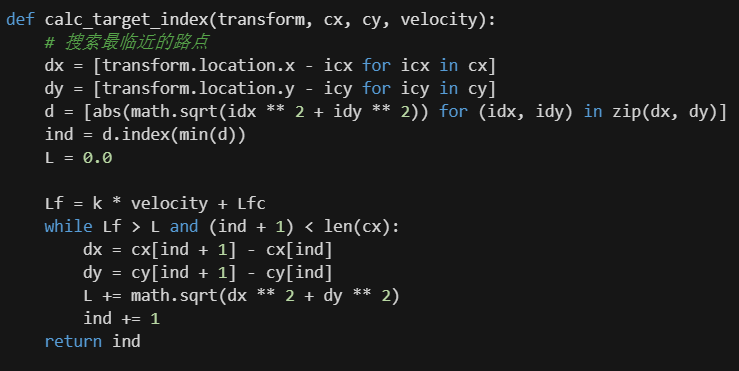


1. 获取车辆当前状态，用于后续更新控制量

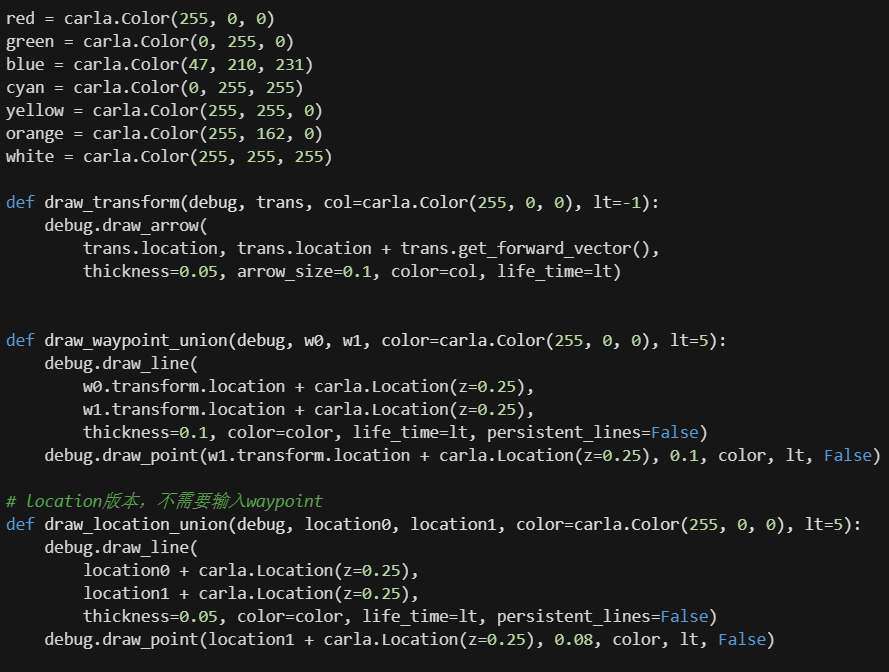


1. Pure Pursuit核心代码





## waypoint可视化展示（便于debug）



## TODO



## 完整代码地址

<https://github.com/QiMingZhenFan/RepoForIntroDocs/blob/master/Pure_pursuit_in_carla/my_actor.py>