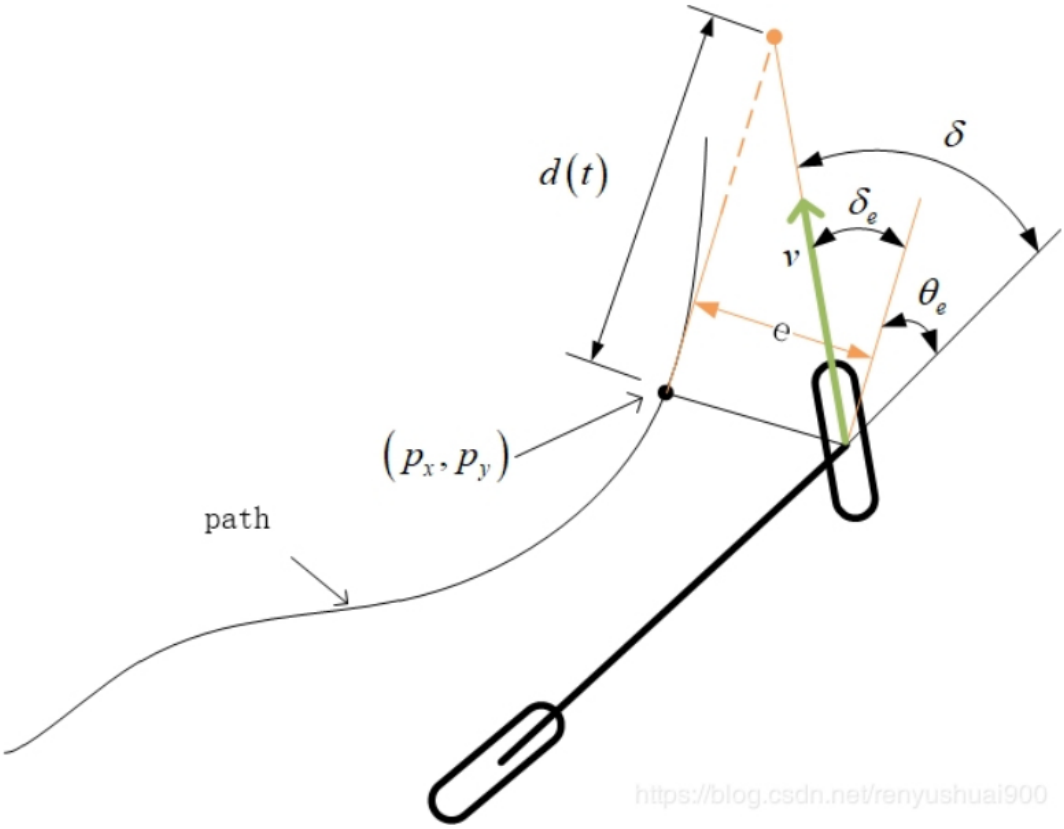
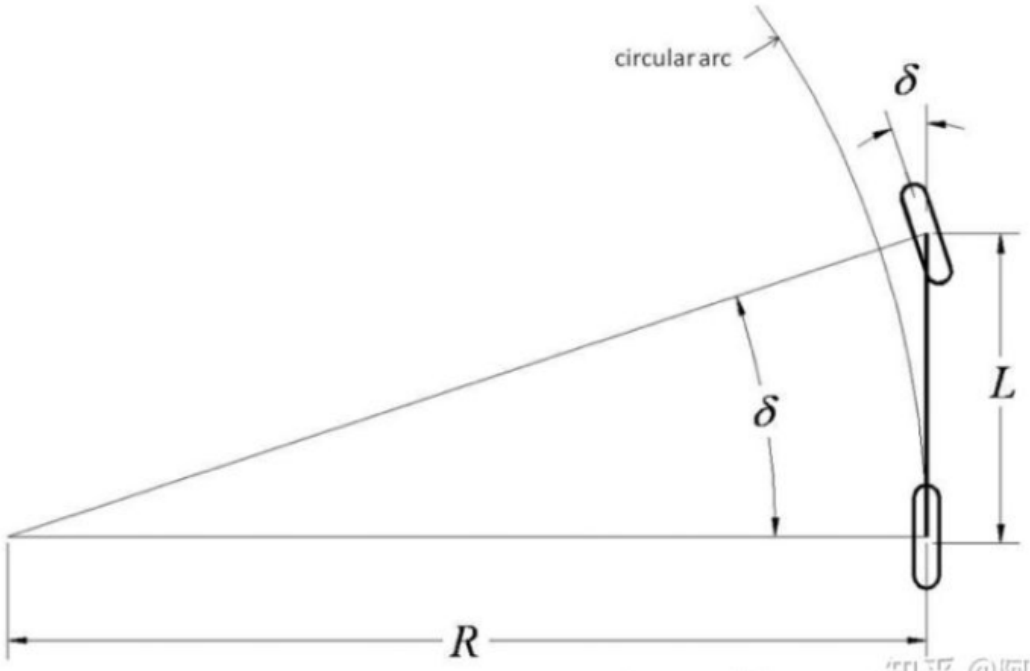
循迹

- 数据
  - gps
  - 艏向，速度
- 1.无激光雷达
  - 运动控制，到达目标点
    - track
      - gps 位置信息
        - unionstrong/gpfpd 当前位置
          - lon_m lat_m 目标点
      - vtg
        - hat
          - a0；
  - 任务分配
    - 1.控制算法 简单比例控制
    - 2.gps计算速度
    - 3发布topic的节点
      - 1.hat 需要python
      - 2.控制速度节点 python
      - 3.发布最佳速度（track）
- 2.有激光雷达
  - 同时避障
- 路径
  - 1.gps
  - 2.google map https://www.google.com/maps/@23.0657945,113.3879571,19z
    - https://developers.google.com/maps/documentation/javascript/reference?hl=zh-cn api js
- 避障算法
  - stanley https://blog.csdn.net/renyushuai900/article/details/98460758

Stanley method

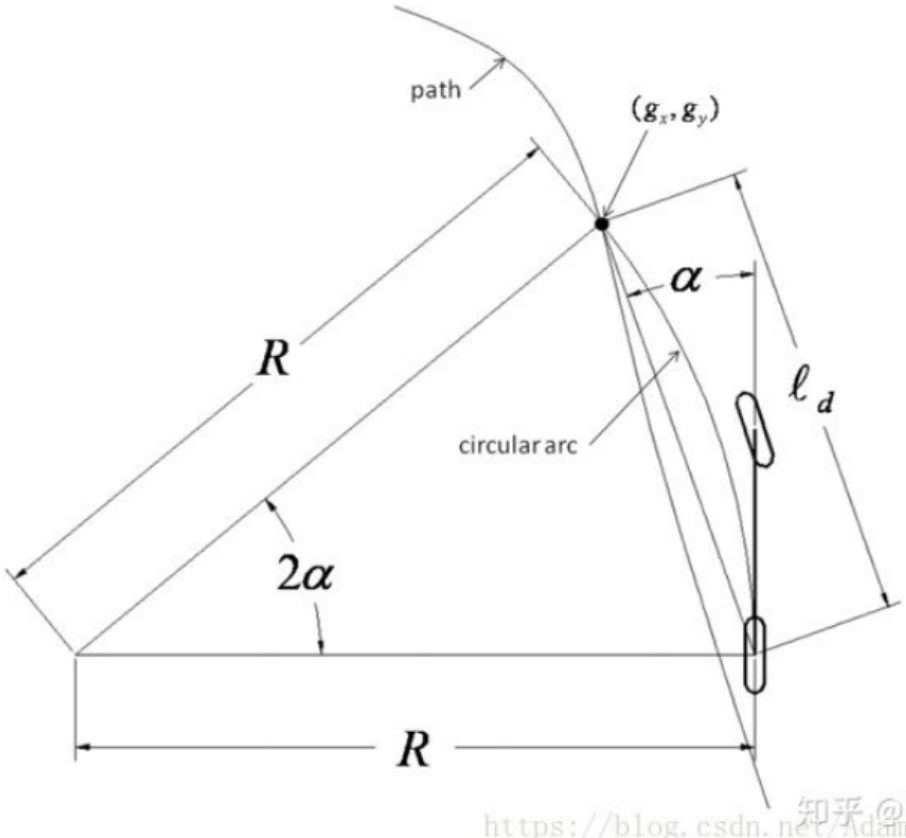


- pure pursuit
  - https://zhuanlan.zhihu.com/p/48117381



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：

$$\tan(\delta) = \frac{L}{R}$$



$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)}$$

$$\frac{l_d}{2\sin\alpha\cos\alpha} = \frac{R}{\cos\alpha}$$

$$\frac{l_d}{\sin\alpha} = 2R$$

因为道路的曲率 $\kappa = \frac{1}{R}$ ，上式也可以表示为：

$$\kappa = \frac{2\sin(\alpha)}{l_d}$$

则由式子 $\tan(\delta) = \frac{L}{R}$ ，可得

$$\delta = \arg\tan(\frac{L}{R}) = \arg\tan(\kappa L) = \arg\tan(\frac{2L\sin(\alpha)}{l_d})$$

再把时间这个变量加进来，得到

$$\delta(t) = tan^{-1}\left(\frac{2Lsin\left(\alpha(t)\right)}{l_d}\right)$$

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 l_d 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： e_l —— 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：

$$\sin(\alpha) = \frac{e_l}{l_d}$$

则曲率可以表示为：

$$\kappa = \frac{2\sin(\alpha)}{l_d} = \frac{2}{l_d^2} e_l$$

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 $\frac{2}{l_d^2}$ ，这个P控制器受到参数 l_d （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， l_d 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 $l_d = Kv_x$ ，那么前轮的转角公式就变成了：

$$\delta(t) = tan^{-1}\left(\frac{2Lsin\left(\alpha(t)\right)}{kv_x(t)}\right)$$

- 角度计算