

# 期末作业简单教程

---



# 感知、规划、控制

---

- 感知。小车通过传感器的数据，得到自己的位姿信息和周围环境的信息，构建场景的局部地图。比如在本次实验中，可以使用GPS和IMU得到小车的位姿，激光雷达获取周围环境的信息。
- 规划，在已知局部地图，小车位姿，目标点坐标下，规划路线，因为在移动的过程中地图会更新，所以需要重复规划。
- 控制。已知规划路线下，根据小车位姿和路线，设计控制量，控制小车的车轮速度。

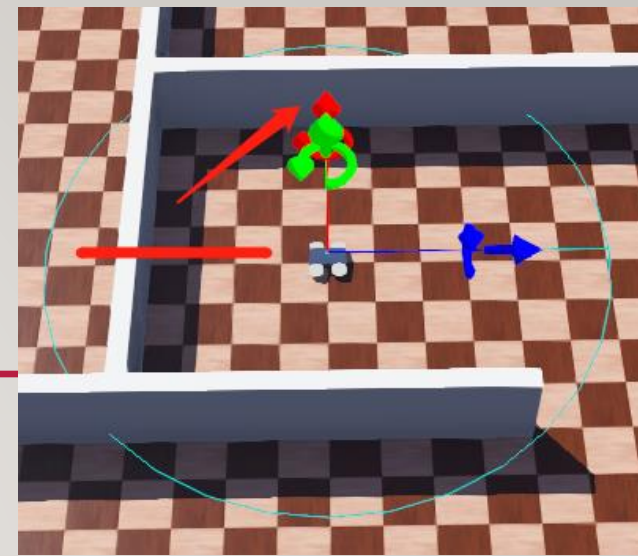
# 感知

---

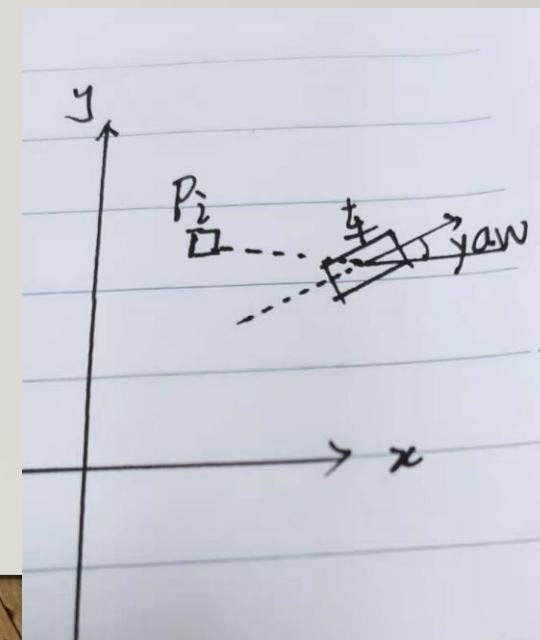
- 1) 雷达是什么
- 激光雷达 (lidar) 。 lidar是一种测距传感器。
- type为rotation时, 激光探头会在一帧时间内, 旋转360°, 获取全向的距离信息。
- 多线雷达有多个探头, 在竖直方向发射多条激光, 获取三维的距离信息。如果只需要二维的信息, 可以使用单线雷达。
- 此外, webots的lidar还提供了PointCloud模式, 直接返回点云信息, 而不是距离信息。

- 
- 2) 地图的种类
  - 2d栅格地图。
  - 最简单的实现，就是一个二维数组，比如场景为5x5m，地图分辨率为0.01x0.01m，则需要的地图大小为500x500的二维数组。
  - 二维数组的元素表示是否被占用（有障碍物），有3种状态，占用（occupy），未占用（free），未知（unknown）。也可以把未知归入未占用，使用occupy和free两种状态来表示地图

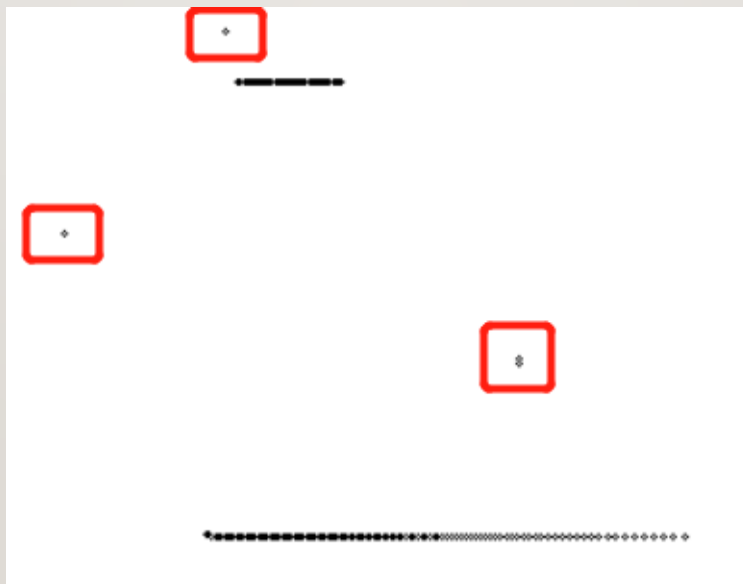




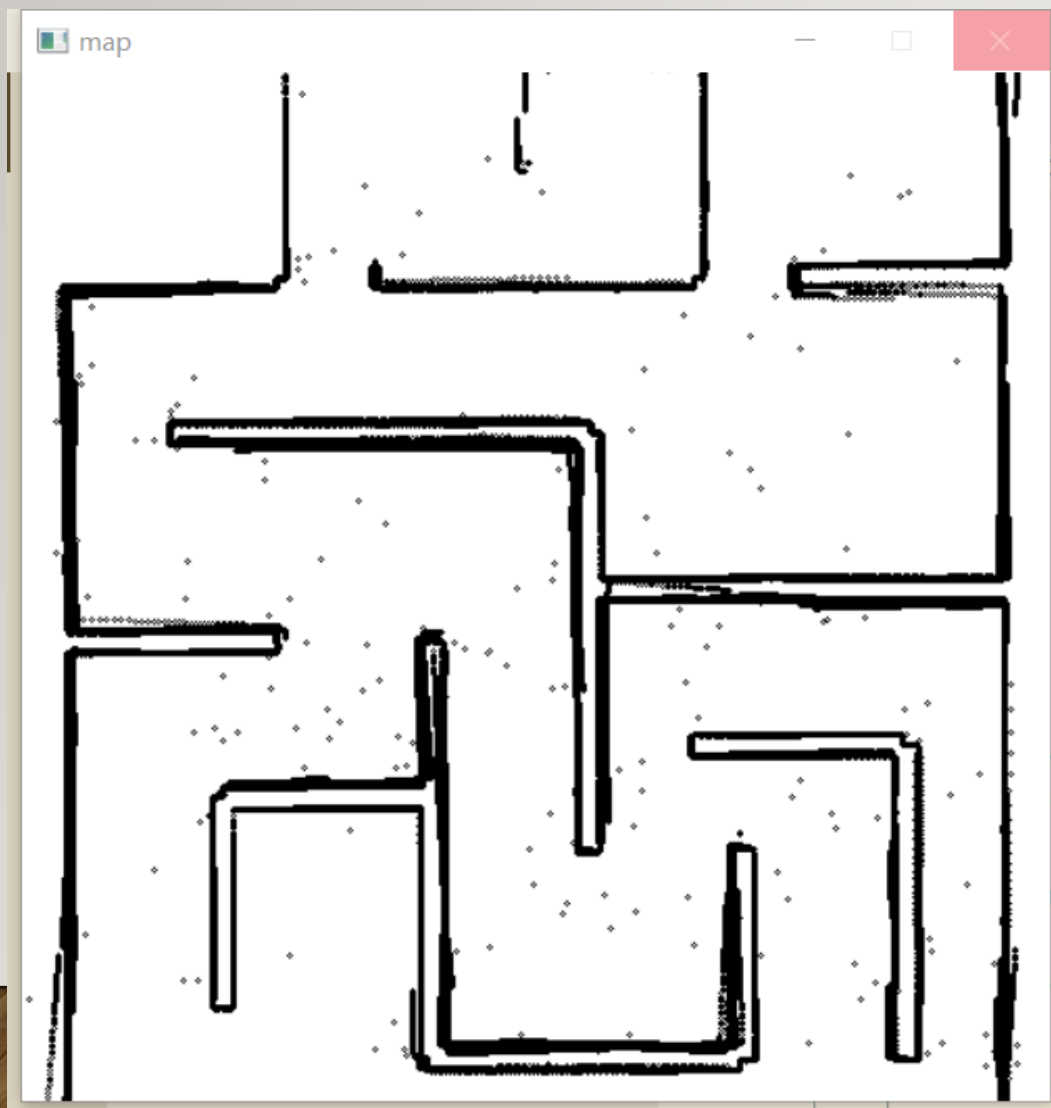
- 3) 如何处理雷达数据
- 根据距离和夹角，使用三角函数求点的坐标。d为距离，a为夹角
- $x = d \cos(a), y = d \sin(a)$
- webots的雷达，是从负z轴方向（图中蓝色轴），顺时针旋转获取点。
- 所以a的计算，假设车的朝向是yaw，则第i个点的角度
- $$\alpha = yaw + \pi - \frac{i}{numPts} * 2\pi$$



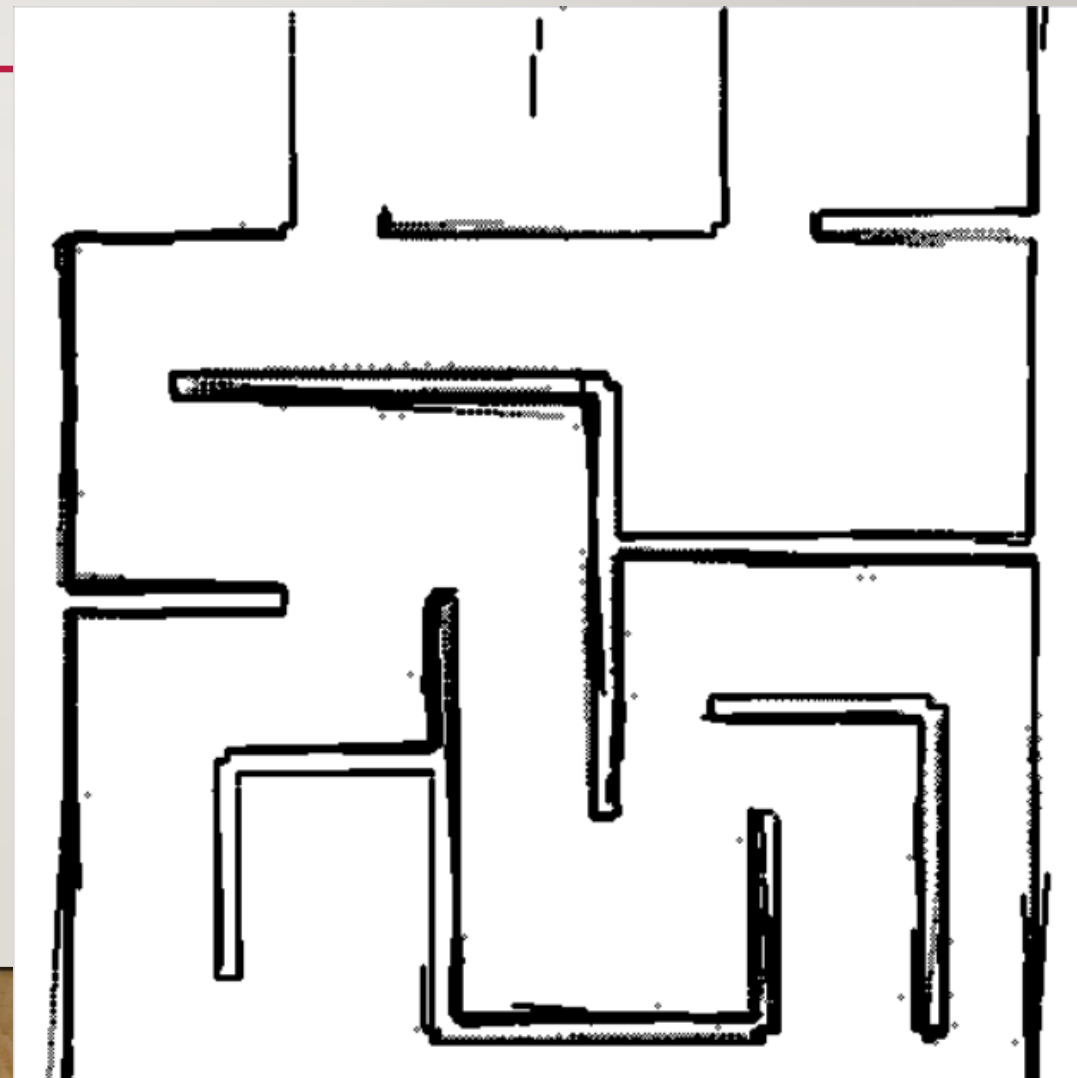
- 
- 离群点 (outlier) , 传感器数据不会完全准确, 测距会有误差。
  - 一个最简单的处理方法, 就是求雷达点  $p$  与邻近若干点的距离差, 如果大于某个阈值, 判断为离群点, 舍弃。



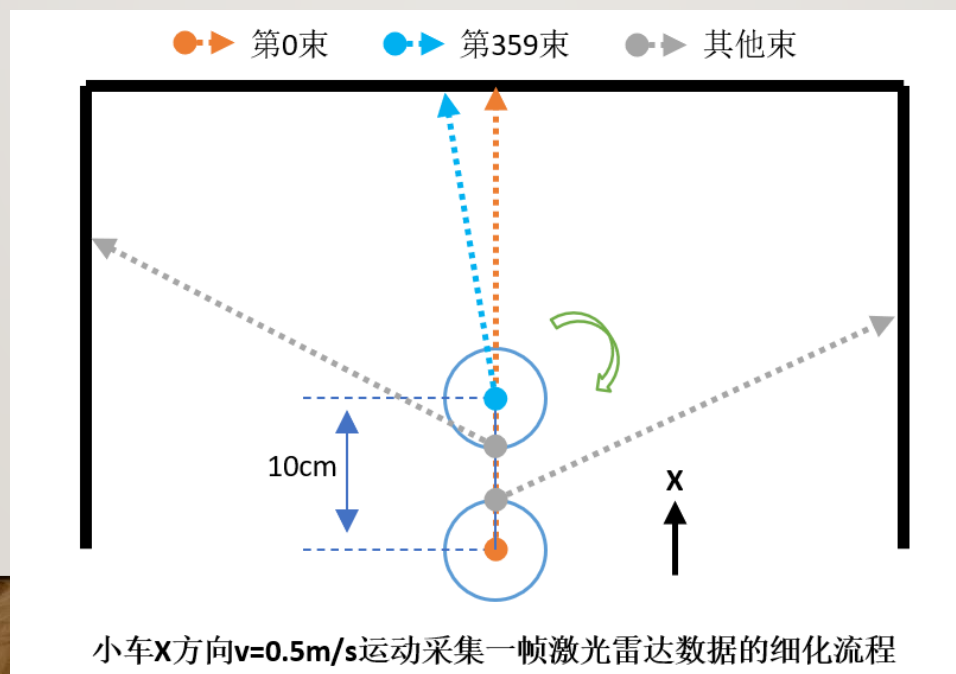
不处理离群点



处理掉离群点



- **运动畸变**：雷达在获取数据时，若雷达同时在运动，则一帧雷达数据中，雷达点会在不同位置被采集。比如，扫描频率5Hz的激光雷达，一帧数据的首尾时间差200ms，若机器人以0.5m/s的速度向x方向移动，扫描前面的墙面，那么200ms后尾部的测量距离和首部的测量距离在x方向上相差10cm，如下图所示：





- 
- 运动畸变的处理方法。
  - 解决方法：在一帧激光雷达数据中，通过插值，为每个雷达点，计算一个小车位姿，根据该位姿进行坐标变换。
  - 最后是建图频率。可以设置得比较低，比如1Hz，具体速度看实际需求，如果小车速度快，可能建图就要快一点。

# 规划

---

- 根据雷达构建的局部地图，使用PRM或RRT算法，就可以得到路径。
- 轨迹平滑。PRM或RRT得到的路径，是折线的，在实际的控制中，曲折的路线会使控制变得复杂。
- 如果想要得到一条平滑的路径，可以进行轨迹平滑，其实就是做插值，在规划路径上取某些点，然后用多项式插值或样条插值。

# 控制

---


- 根据小车位姿和规划路线，就可以设计控制量来控制小车

## 其他事项

---

- 1) webots的仿真，仿真时间和代码实际运动时间是不一样的。如timestep为32ms，若while循环不满32ms，会休眠程序直到时间满32ms，进入下一个循环。
- webots仿真时间后有一个系数，0.92x，表示仿真时间是实际时间的0.92倍。

- 
- 如果快进，就会发现，系数变为3.19x

- 

- 
- 2) 控制台提醒: The requested velocity 30 exceeds 'maxVelocity' = 20.
  - 电机可以设置maxVelocity=30或更高, 即可解决
  - 3) webots提供离线文档, 如果翻墙不方便, 可以在Help - Offline Docs查阅文档。