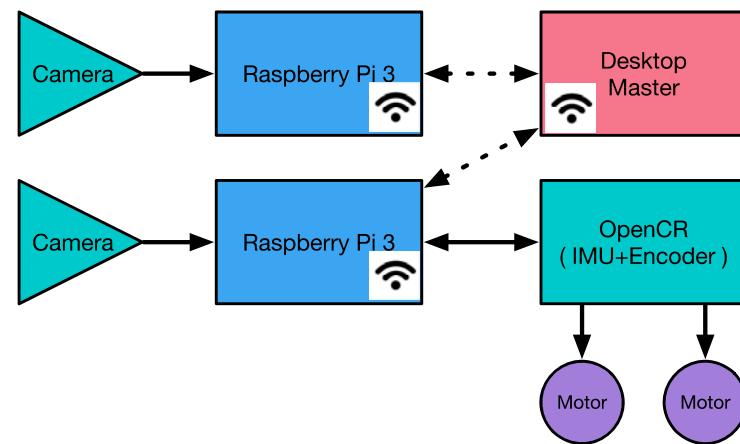


目录

- 1 硬件结构
- 2 场景特点
- 3 方案设计
- 4 比赛策略
- 5 实现效果
- 6 经验总结
- 7 分工情况

1 硬件结构



2 场景特点

机器人

- ▶ 传感器 = {相机 $\times 2$, IMU(MPU9250), 码盘}
- ▶ 驱动器 = {两轮差速地盘}
- ▶ 控制器 = {树莓派 $\times 2$, OpenCR, PC}

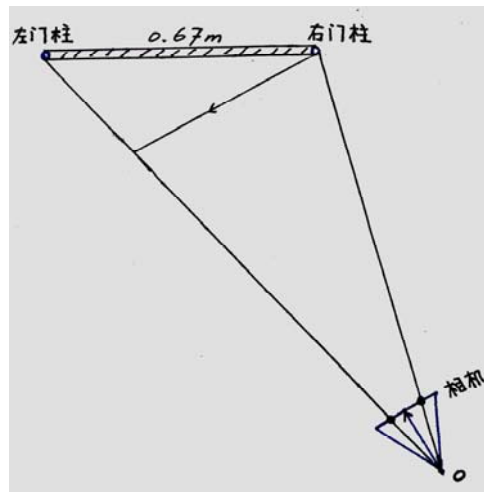
物理环境

- ▶ 实际球场平面凹凸不平
- ▶ 小球重心与球心不重合
- ▶ 小球在地面上容易滚动
- ▶ 周围环境纹理少且重复
- ▶ 相机水平视角 $\theta < 90^\circ$
- ▶ 小车容易出现碰撞打滑
- ▶ 门柱全局位置固定不变

3 方案设计：语义定位原理

输入：偏航角 θ_{yaw} ，门柱像素坐标，左右门柱间距离

输出：全局坐标 (x, y, θ_{yaw})



3 方案设计：白板检测与定位

输入：白板成像高度

输出：全局坐标分量 x 或 y

Step1 背景分割



Step3 白板检测



Step2 绿色球场检测

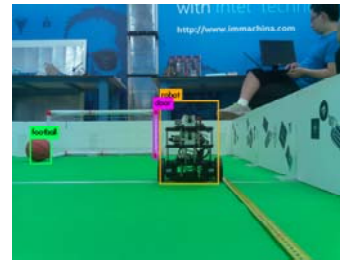


Step4 检测效果



3 方案设计：感知

- ▶ 传统方法: HSV 颜色空间 ▶ 深度学习: Tiny YOLO



- ## ► 深度学习 + 目标跟踪

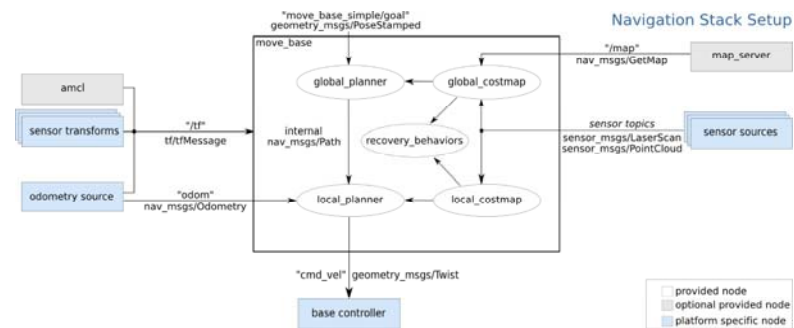
最终方案

- ▶ Tiny YOLO + GPU(MX150) 加速

3 方案设计：规划

筛选方案

- ▶ 视觉伺服
- ▶ 全局路径规划与控制跟踪：ROS Navigation Stack



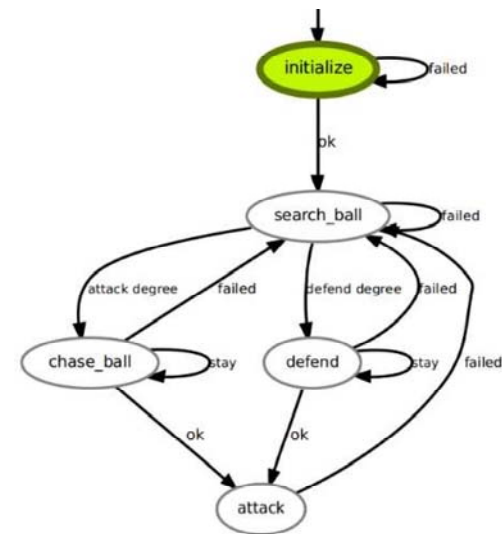
最终方案

- ▶ 视觉伺服，采用 PID 控制算法实现进攻和防守功能

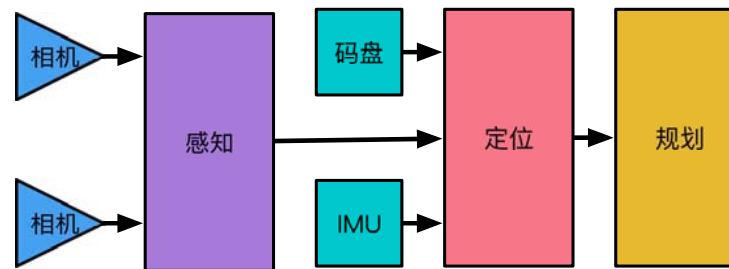
4 比赛策略：二号线

Algorithm

1. 发现小球时：
 - ▶ 距离远时，靠近小球
 - ▶ 距离近时
 - ▶ 满足条件时，进攻
 - ▶ 否则，绕球后方并防守
 2. 未发现小球：
 - ▶ 旋转寻找小球或继续完成防守流程
-

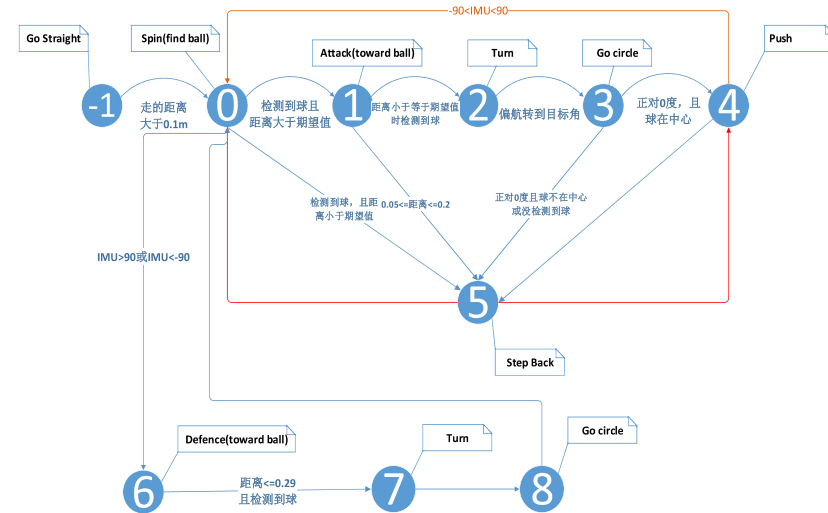


4 比赛策略：二号线框架

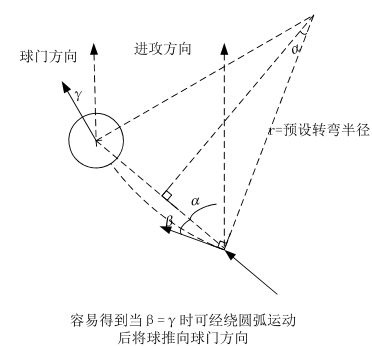
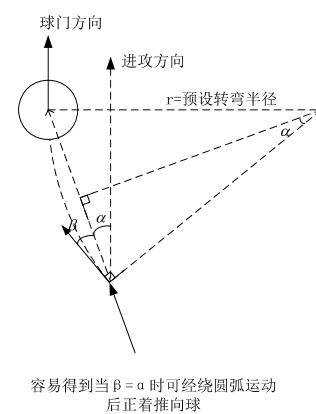


- ▶ 上相机 (640*480) 用于门柱定位，下相机 (320*240) 用于小球检测
- ▶ 感知模块采用 Tiny YOLO，定位门柱和小球的像素坐标
- ▶ 码盘提供 (x, y) ，IMU 提供 θ_{yaw}
- ▶ 规划采用增量式 PID 控制算法实现

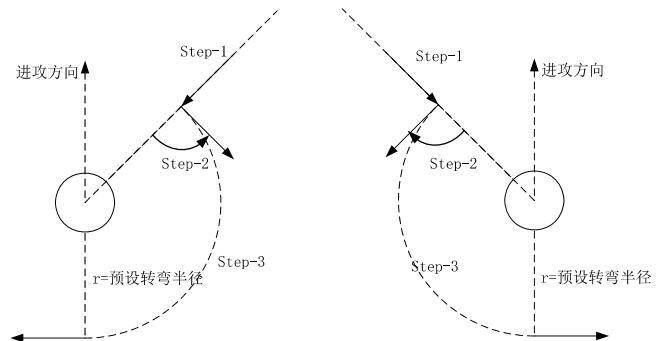
4 比赛策略：五号车



4 比赛策略：进攻策略



4 比赛策略：防守策略



5 实现效果

TODO: 视频

6 经验总结

鲁棒 工程中，算法的鲁棒性比精度重要

细节 一个实用的程序会有很多的工程设计和实现技巧

- ▶ 防止乌龙球
- ▶ 碰撞后重定位

简单 简洁的架构可以保证系统的稳定性和扩展性

Trade-off 稳定性与系统复杂性之间的平衡

理论 vs 工程 Paper 中的算法很难直接用于实际工程

TODO Lists

1. 全局定位系统 (多传感器融合)
2. 多机器人协作
3. ShuffleNet v2 移植到树莓派