

东北大学

摇光战队

RMUA 2022 技术报告



中国 沈阳 东北大学

2022 年 4 月 18 日

视频地址:

https://v.youku.com/v_show/id_XNTg2MjcxNjEzNg==.html

视频密码:alkaid

1. 硬件

1.1 机械结构

1.1.1 车轮防护装置

在调试过程中发现,车轮会因为碾压子弹而抬起,由于里程计使用的是电机自带编码器,车轮与地面的脱离会导致里程计误差增大,影响定位精度。因此根据车轮尺寸、车边缘距离地面高度等数据,设计了车轮防护装置。安装防护装置后,提高了里程计数据的稳定性和定位精度,车轮安装防护装置,如图 1.1 所示:

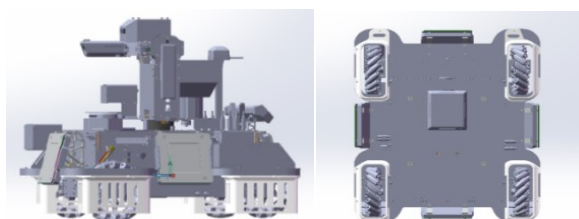


图 1.1 车轮防护装置安装示意图

1.1.2 激光雷达安装架

由于比赛场地围挡高度的降低,为扫描到更丰富的障碍物,安装雷达安于车体车轮的左上角,同时考虑到激光雷达的散热问题,选用空心铝板型材,保证空气流通,以使激光雷达工作在合理的温度范围内,激光雷达安装架如图 1.2 所示:

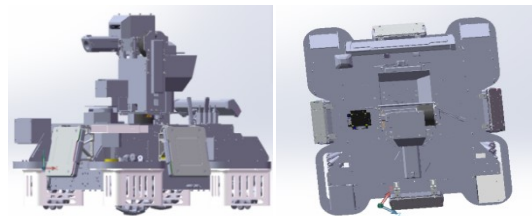


图 1.2 激光雷达定位固定示意图

1.1.3 弹仓

为避免机器人大幅度动作导致弹丸洒出,在比赛规则允许的范围内,设计了一款平行四边形弹仓,采用玻纤板拼接,使用 M3 螺栓和防松螺母固定,确保强度的同时也足够轻量化、易安装,弹仓,如图 1.3 所示:

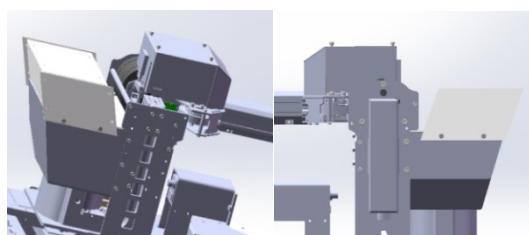


图 1.3 平行四边形弹仓安装示意图

1.1.4 哨岗接口

哨岗采用 5.5*3cm 的铝合金球形万向云台作为哨岗架和相机的连接件,云台可以 360 度水平旋转,90 度垂直旋转调整相机角度,底座使用 1/4 转 M4 的转接螺丝与哨岗架相连,顶部采用 1/4 螺丝连接免驱摄像头,球形万向云台如下图所示。



图 1.4 哨岗机械接口

1.2 传感器

1.2.1 激光雷达

使用 HOKUYO 公司 UST-10LX 2D 激光测距传感器，该产品拥有 10m, 270° 的测量范围，DC12V/24V 输入，25ms 扫描时间，IP65 防护等级，非接触式测量。可用于高速运动机器人避障和位置识别；高精度、高分辨率、宽视场设计给自主导航机器人提供了良好的环境识别能力；紧凑型设计也节约了安装空间，低重量、低功耗，基本满足了比赛场地的定位需求。

1.2.2 工业相机

车载相机使用海康威视工业相机，型号为 MV-CA016-10UC，支持 USB3.0 接口，采用 CMOS 全局快门，可以清晰的捕捉高速运动的物体，分辨率为 1440x1080 的同时最高帧率可达 249.1fps，既满足对测量精度的要求也保证了视觉的帧率上限。

哨岗相机使用全瑞视讯 USBFHD08S 型号的免驱相机，该相机采用 CMOS 全局快门，最大分辨率为 1920x1080，最高帧率可达 60fps，满足机器人识别需求。由 USB3.0 延长线与哨岗相机相连，便于在场间迅速安装调试设备。

1.2.3 惯性传感器

使用高精度六轴惯性器件 BMI088，传感器主要参数如表 1.1 所示：

表 1.1 BMI088 参数

角速度测量	加速度测量	供电电	通讯方式	数据读取频
范围	范围	压		率
±2000° /s	±24g	3.3V	SPI/IIC	最大 2000Hz

1.3 计算设备

机载电脑使用英特尔 NUC8i5BEHS 豆子峡谷, 运行内存 16GB, 硬盘容量 256GB.8 核 16 线

程,基本可满足导航、定位、视觉、决策等算法的算力需求。

哨岗采用配置有 RTX3060 显卡的 PC 作为运算设备，具备高算力，可高速运行哨岗程序，保证识别的实时性。

1.4 通讯链路

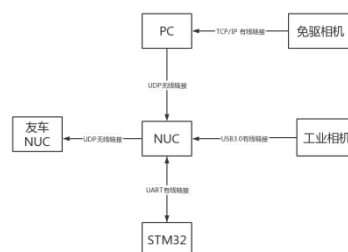


图 1.5 上位层机通讯链路

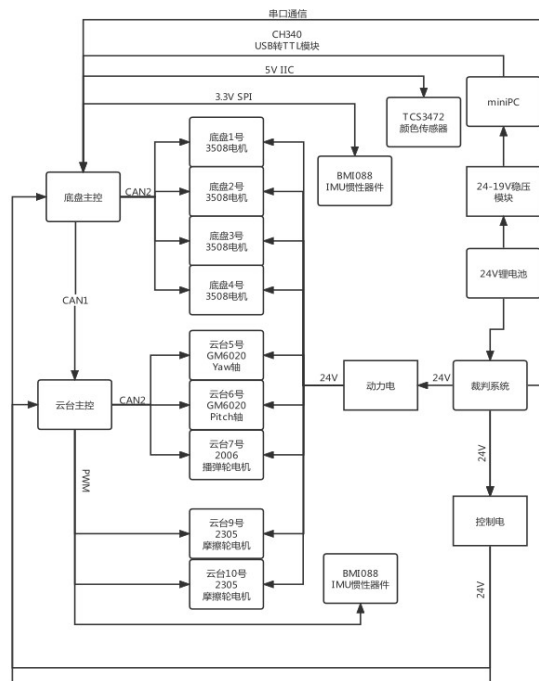


图 1.6 下位机层通讯链路

2. 软件

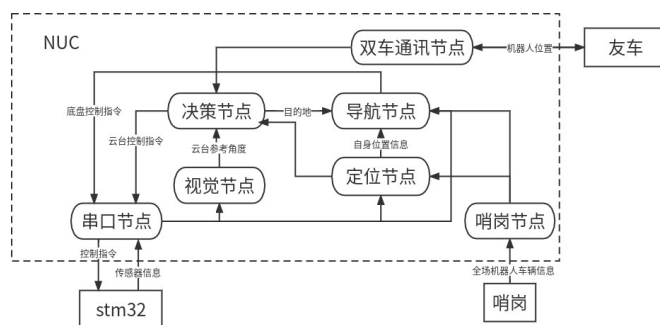


图 2.1 机器人算法

2.1 自动识别

机器人自动识别部分性能如表 2.1 所示：

表 2.1 自动识别部分性能

相机分辨率	视场角	平均帧率
1440*1080	44°*34°	85FPS

2.1.1 装甲板识别

装甲板具有灯条、数字和底纹三个显著特征，针对这三种特征的装甲板识别流程如图 2.2 所示

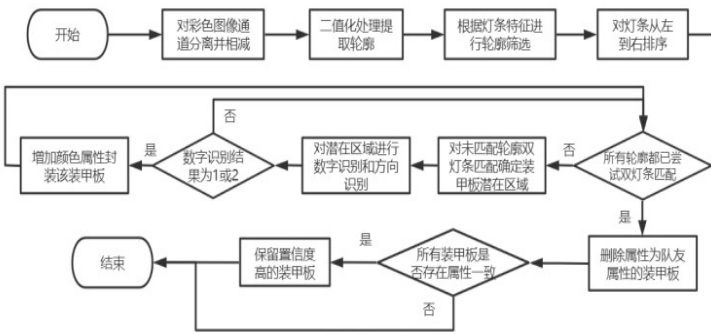


图 2.2 装甲板识别流程图

对于装甲板的灯条特征，采用传统的图像处理方法从颜色、形状和空间位置进行筛选，可以快速的锁定装甲板潜在区域，便于后续的装甲板识别；

对于装甲板的数字特征和方向特征，选择基于 MXNet 深度学习框架的 CNN 实现，MXNet 是个比较轻量且实用的框架，CNN 是一种比较主流的深度学习网络。装甲板识别部分性能如表 2.2 所示：

表 2.2 深度学习部分性能

输入图片尺寸	耗时	识别准确率
56*56	2-3ms	95.3%

2.1.2 装甲板解算

敌方车辆位置检测采用 OpenCV 上 solvePnP 进行解算，利用 PnP 算法解算出目标在相机坐标系下的位姿后，通过相机到机器人自身中心的位移

量和陀螺仪数据计算出的欧拉角信息将相机系下的装甲板坐标换算到以己方车中心建立的世界系下的坐标（以地面为 XOZ 平面）。由此可以确定敌方装甲板相对于己方的位置及朝向。考虑到世界系坐标 X 轴、Z 轴测量较为困难，所以仅展示距离和 Y 轴分量，装甲板解算部分性能如表 2.3 所示：

表 2.3 装甲板解算部分性能

距离	1m	2m	3m	3.5m
距离误差	±1.8cm	±2.5cm	±3.2cm	±4.4cm
Y 轴分量误差	±0.2cm	±0.5cm	±1.2cm	±1.6cm

2.1.3 装甲板筛选

考虑到场地内存在的镜子和玻璃可能会对识别结果造成干扰所以需要解算后的装甲板进行删选,筛选流程如图 2.3 所示。

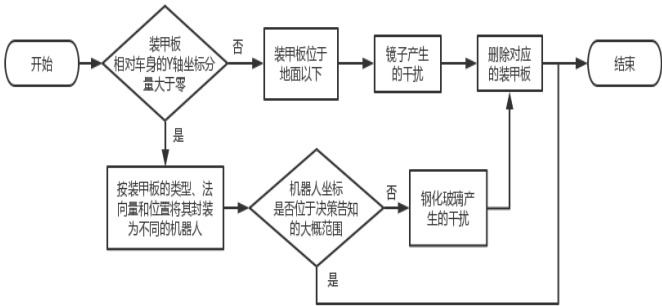


图 2.3 装甲板筛选流程图

在经过上述筛选后，对视野中不同类型的装甲板分别选择面积最大的装甲板为选中的装甲板，结合决策发来的攻击指示、不同方向装甲板的权重以及历史目标选择优先击打目标。

2.1.4 装甲板预测

为了有效击打选中的装甲板，我们采用了基于匀加速度的卡尔曼滤波算法进行预测，原理是利用线性系统状态方程，通过系统输入输出观测数据，

对系统状态进行最优估计。根据运动学公式，其离散模型如下，状态方程：

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathbf{w}_k \quad (2-1)$$

观测方程：

$$\mathbf{y}_k = H \mathbf{x}_k + \mathbf{v}_k \quad (2-2)$$

其中式中 \mathbf{x}_k 为状态矩阵， \mathbf{u}_k 为控制矩阵， \mathbf{y}_k 为观测矩阵， A_k 为状态转移矩阵， B_k 为状态控制矩阵， H 为观测矩阵， \mathbf{w}_k 为 k 时刻的动态噪声，为 k 时刻的观测噪声。

通过相机对敌方装甲板的 pnp 解算，我们可以得到敌方装甲板的位置，通过相邻帧对时间的微分可以得到敌方装甲板的速度和加速度，考虑到加速度噪声太大，而敌方又存在加速和减速，那么把位置和速度作为系统的状态变量，则系统的观测矩阵

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \text{ 则状态转移矩阵 } A_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}; \text{ 把加}$$

$$\text{速度作为控制变量，则状态控制矩阵 } B_k = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}。$$

卡尔曼滤波算法通过上一时刻的估计值和当前时刻的观测值来完成对状态变量的估计。可以分为时间更新和状态更新两个部分。

时间更新：

$$\begin{cases} \hat{\mathbf{x}}_k^- = A \hat{\mathbf{x}}_{k-1}^- + B \mathbf{u}_{k-1} \\ P_k^- = A P_{k-1}^- A^T + Q \end{cases} \quad (2-3)$$

状态更新：

$$\begin{cases} K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{y}_k - H \hat{\mathbf{x}}_k^-) \\ P_k = (I - K_k H) P_k^- \end{cases} \quad (2-4)$$

式中 P_k 为协方差矩阵， K_k 表示卡尔曼增益，

R 为观测噪声的协方差矩阵， Q 为系统噪声的协方差矩阵， $\hat{\mathbf{x}}_k$ 是当前时刻的最优估计。通过时间更新和状态更新对模型参数进行更新，保证滤波的效果，通过当前时刻的最优估计和当前时刻的外部控制预测下一时刻的状态，进而达到预测的效果。

同时考虑到敌方机器人很大概率在进行摇摆，而装甲板在旋转到一定角度后会出现攻击无效的情况，所以根据装甲板法线角度来计算允许的预测区间和开火区间，降低匀加速度模型和实际情况不附带来的影响。

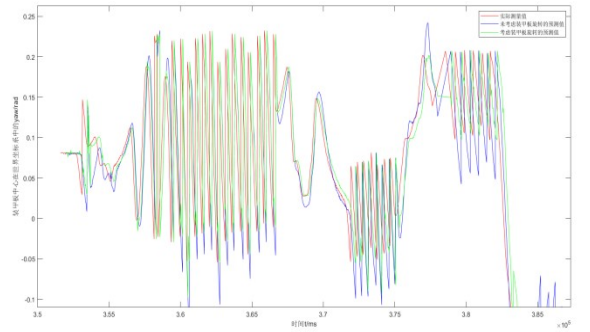


图 2.4 装甲板 yaw 角预测优化

如图 2.4，红色曲线为敌方装甲板 yaw 角的实际测量值，蓝色曲线为未考虑敌方装甲板旋转进行的卡尔曼预测，可以看出在未考虑装甲板旋转时针对敌方变向进行的预测会出现很大的超调，而绿色曲线为考虑了敌方装甲板旋转的预测，可以看出虽然会导致在敌方变向时预测角度有一定滞后，但极大的抑制了超调，在实际对战中减少了子弹的浪费。

2.2 定位

2.2.1 全局定位

在比赛中，需要实时获取机器人的位置。我们采用 AMCL 算法进行全局定位，算法输入预先画好的 yaml 格式地图，利用 IMU 数据和编码器数据

松耦合实现里程计输入，该部分作为 AMCL 算法的先验部分，是控制模型的确立。根据先验更新粒子位姿后，激光雷达数据作为后验计算每个粒子的权重，从而实现最终位姿的确定。

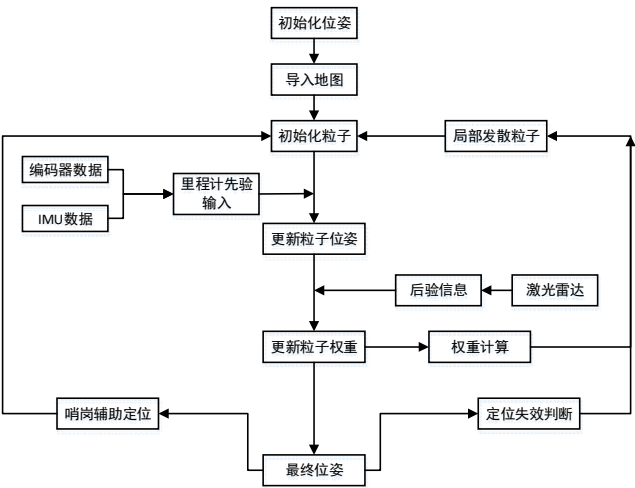


图 2.5 粒子滤波定位算法框架

AMCL 定位算法的性能如表 2.3 所示：

表 2.3 定位算法各部分性能参数

全局定位	局部更新	X 方向精度	Y 方向精度	哨岗辅助
频率	频率			定位频率
10Hz	20Hz	5-10cm	5-10cm	40Hz

1.2 局部发散粒子

该部分是针对 AMCL 部分的粒子丢失全局重定位功能的改进。全局重定位功能用于解决机器人位姿丢失和机器人绑架情况下的全局定位问题。然而在实际比赛中并不会出现机器人绑架问题，且位姿丢失后的机器人相较于丢失前位姿距离很近。因此将全局定位部分的位姿发散进行优化，主要分两方面。

a. 以位姿丢失（权重失效）上一时刻位置为中心，向 x，y 方向发散 0.8m，角度为 360° 的方形范围进行位姿找回。如图 2.6(a) 所示；

b. 由于比赛场地中心对称，因此在靠近场地中间部分的位姿发散容易造成镜像问题。我们将粒子

发散范围限制分为四部分，以中心障碍物为直角坐标系，将地图分为面积相同的四个长方体部分。在粒子发散前先进行位置判断，确定机器人所处的是哪一部分，根据 a 中的方法进行粒子发散，如果出现粒子发散超出本部分的粒子，进行删除处理。从而解决了中心对称问题。

c. 由于 IMU 的存在，陀螺仪角度在比赛时间内整体漂移小于 3 度，因此粒子发散的角度都是以当前时刻的机器人实际角度为准，减少不必要的角度发散，增加粒子收敛速度和准确性。

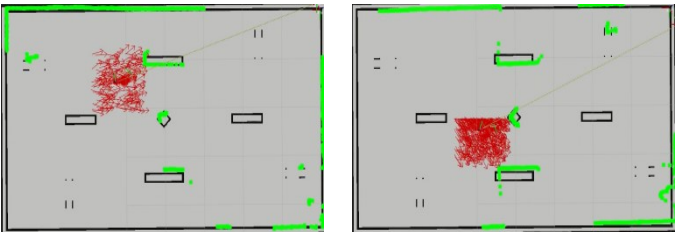


图 2.6 (a) 粒子发散范围 (b) 场地边缘粒子发散效果

1.3 哨岗辅助定位

哨岗具有全局监视作用，机器人获取哨岗发送的位置信息，与当前位姿对比。当差值超过设定的阈值，认为机器人定位失真。此时以哨岗发送的位置为中心，在局部范围(1.2m*1.2m)散发粒子，如图 2.7(a)，可迅速解出当前正确位姿，如图 2.7(b)。由于各方面延时，接收的位置信息存在滞后性。若仍以高斯分布粒子，丢失正确位姿概率较大，故以均匀分布散布粒子，以减少延迟带来的影响。

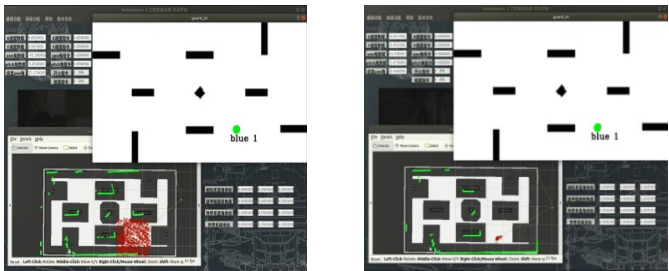


图 2.7 (a) 哨岗位置发散粒子 (b) 位姿解算后

2.3 运动规划

借鉴香港科技大学空中机器人实验室开源运动规划框架 Btraj 的思路,同时结合比赛实际场景,构建了以下运动规划框架,经过实际测试规划频率 50hz-100hz,最大运动速率 1.5m/s,摇摆中导航较少出现碰撞问题。

2.3.1 前端搜索 fast marching

FastMarching 是一种特殊的水平集方法,利用 Eikonal 方程(式 2-5)描述波的传播,且自带时间参数,符合全局最优,时间复杂度为 $O(N\log(N))$,使用 fast marching 方法搜索路径,更易于后续对轨迹进行优化。

$$|\nabla T(x)| = \frac{1}{f(x)} \quad (2-5)$$

同时考虑到尽可能使机器人远离障碍物更安全的行驶,基以下公式构建速度场

$$f(d) = \begin{cases} v_m(\tanh(d - e) + 1)/2, & 0 \leq d \\ 0, & d < 0 \end{cases} \quad (2-6)$$

基于上述所构建的速度场,使用 Fast marching 算法进行前端路径搜索,将得到相对平滑的路径。

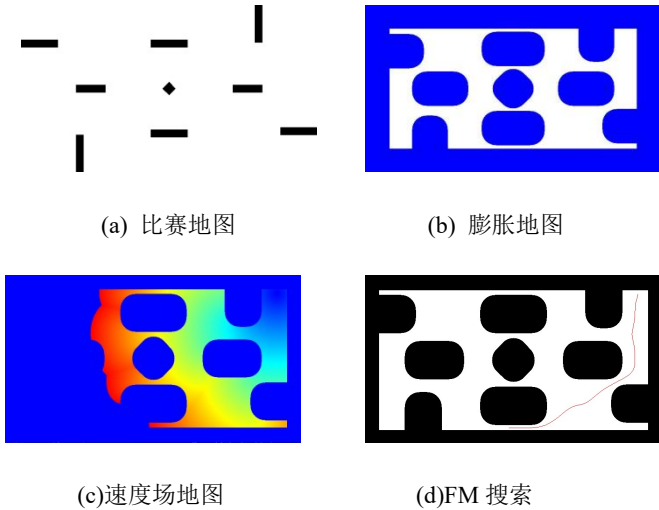


图 2.8 前端路径搜索

2.3.2 后端优化 minimum jerk

在前端搜索得到的路径基础上,同时考虑运动学和动力学,利用 minimum jerk 方法对前端搜索得到的路径进行优化。

第一步:设定优化目标

视所规划的路径为多项式组合,如公式

$$p(t) = \begin{cases} [1, t, t^2, \dots, t^n] \cdot p_1 & t_0 < t < t_1 \\ [1, t, t^2, \dots, t^n] \cdot p_2 & t_1 < t < t_2 \\ \dots & \dots \\ [1, t, t^2, \dots, t^n] \cdot p_k & t_{k-1} < t < t_k \end{cases} \quad (2-7)$$

为使得路径更加平滑易于控制器执行构建如下优化目标:

$$\text{minimum jerk: } \min f(p) = \min (p^{(3)}(t))^2 \quad (2-8)$$

第二步:构建等式约束

综合考虑起点和终点的位置、速度、加速度约束, $K-1$ 个中间点的位置约束、 $3K-3$ 个中间点连续性位置、速度、加速度约束,由此构建出等式约束。

第三步:构建不等式约束

根据膨胀地图的障碍物信息,构建飞行走廊,将轨迹约束在可同行区域内。

第四步:求解 QP 问题

由上述步骤易构建出典型的 QP 问题,如式(2-9)

$$\begin{aligned} \min \quad & p^T Q p \\ \text{s.t.} \quad & \begin{cases} A_{eq} = b_{eq} \\ A_{ie} \leq b_{ie} \end{cases} \end{aligned} \quad (2-9)$$

求解出多项式的参数,形成可用于跟踪的轨迹。

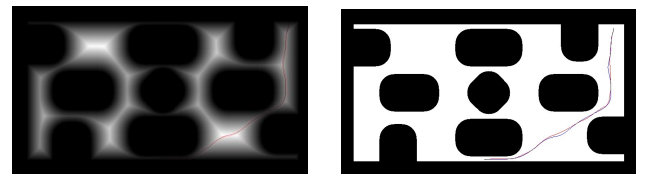


图 2.9 (a) 膨胀地图不等式约束 (b) minimum jerk 优化路径

2.3.3 动态避障

由于激光雷达安装在车轮前侧,激光扫描范围被限制,难以利用激光雷达实现对全局动态障碍物的

的感知，故利用哨岗感知全场的动态障碍物(敌方车辆和己方车辆)，并接受哨岗信息更新膨胀地图。

2.3.4 轨迹跟踪 PID

出于优先打击敌装甲板和充分利用全向移动(后退、平移、摇摆中前进等)的考量，特将移动与旋转解耦，导航算法不再对偏航角进行控制，仅控制全局坐标系下(以初始位置为原点)车体 x 、 y 线速度。选取恰达的前视距离，分解至 x 、 y 方向上，并利用两个方向上计算出的距离误差，并在梯形速度规划的前提下，利用 PID 算法不断反馈调节下发的速度指令。

2.4 自动射击

2.4.1 惯性传感器数据处理

通过对 BMI088 传感器的数据进行处理，传感器的零漂最好状态下可达到 $1^{\circ}/(30\text{min})$ ，常态稳定性 $1^{\circ}/(15\text{min})$ ，通过对原始数据进行采集，通过 MATLAB 做 4096 点的 FFT，得出结果如图 2.10:

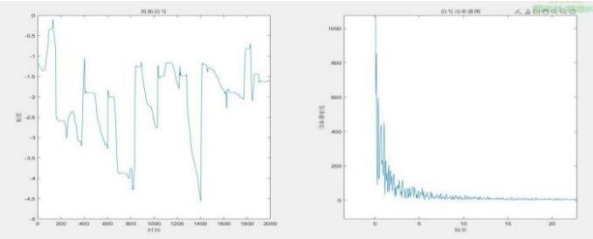


图 2.10 惯性传感器数据采集及 FFT 变换

通过查看采集到的原始数据的频谱，基本可以确定基波频率在 0-10Hz，故选取二阶低通滤波器对采集到的数据进行处理，采样频率设定到 50hz，截止频率设定为 10hz，最终通过数据对比二阶低通滤波对缓解车体突然加速和撞击产生的高频振动具有明显效果，陀螺仪的数据稳定性进一步提高，在使用过程中发现，随着环境温度的变化，陀螺仪的温漂无法通过软件算法进行控制，于是通过读取 BMI088 传感器内部温度数据，进行闭环温度控制，最终使得传感器达到以上所说效果。

2.4.2 射击平台控制策略

电控部分尝试了多种方案，发现无论是底盘通过云台 Yaw 编码器跟随云台，还是云台跟随底盘，在实际运行中极容易出现车体疯转的情况，因此决定采用底盘云台分开控制的逻辑，底盘姿态通过底盘主控板上的陀螺仪进行控制，云台姿态通过云台主控板上的陀螺仪进行控制。

在对云台 Yaw 轴的控制算法上，决定使用基于前馈补偿的 PID 控制算法，即微分先行控制算法，其优势在于比传统的位置控制 PID 算法要有更高的系统跟踪性能，同时可以针对视觉 pnp 解算的 yaw 轴角度存在高频小范围抖动的情况进行优化，其结构图如下所示。

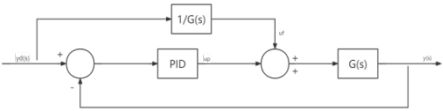


图 2.11 基于前馈补偿的 PID 控制

对于 Pitch 轴的控制，由于实际对战中 Pitch 变化并不剧烈，所以对其视觉的 Pitch 指令实时跟随性能要求不太高，但云台的俯仰会受到重力的影响，其数学模型不同于 Yaw 轴，并且 Pitch 轴活动角度范围 -25° 到 $+22^{\circ}$ 范围内，所以先采用常见的串级 PID 控制，算法简单。为了提高控制精度，将电机绝对位置编码器反馈回来的角度进行处理，因为编码器反馈角度范围是 0-8192 对应度数 360° ，所以经过转换之后 Pitch 轴的控制精度可以达到 0.05° ，在视频中命中率达到 $62/80=0.775$ 。

2.5 哨岗全局观测

2.5.1 哨岗任务

哨岗具有全局视角，可以一定程度弥补视觉视角窄的缺陷，可以辅助决策层有的放矢的寻找目标，此外哨岗提供的场上机器人坐标还可为导航的路径规划提供障碍物信息，潜伏阶段后决策层利用

哨岗信息结合雷达定位信息锁定击打目标。

2.5.2 哨岗检测

哨岗采用帧差法+KCF 跟踪实现车体的目标检测。因为场地和相机都是固定不变的，场景内只有机器人在移动，因此我们采用帧差法检测移动物体。帧差法对移动物体非常敏感，无论机器人处于摇摆状态或前进状态，都可以通过帧差法确定机器人轮廓，获取选框后对框内物体使用 KCF 算法进行跟踪，此方法相比与基于深度学习的目标检测极大地提高了程序的运行效率和鲁棒性，且对运行设备没有过高的环境依赖，可随时更换为上场队员的随身电脑，保证系统稳定性。

3.5.3 哨岗定位

采用通道相减和 SVM 分类器定位机器人相对地图坐标系的坐标，KCF 算法获取稳定车体框后，在车体框内使用 RGB 色彩通道中的红蓝通道相减，可分割出机器人上具有明显特征的血条和灯条，通过长宽匹配确定装甲板的位置后，使用 SVM 分类器进行装甲板数字识别，最后使用 OpenCV 中自带的 solvePNP 方法对装甲板的灯条进行位姿解算，得到装甲板相对相机的旋转矩阵和平移矩阵，由于哨岗高度和安装位置及安装角均可在比赛前测量获得，那么通过坐标变换可得到精确的相对地图坐标系的坐标。

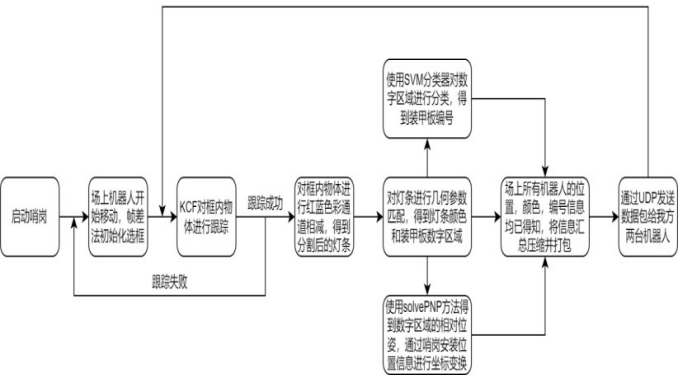


图 2.12 哨岗识别总流程



图 2.13 哨岗 KCF 跟踪

2.6 智能决策

2.6.1 智能决策框架-行为树(Behavior Tree)

技术报告的决策部分是以行为树作为决策的基本框架。行为树是一种构建在自主智能体中不同任务中间切换的方法，是创建具有模块化和反应性复杂系统的一种有效方式。

和有限状态机不同，在 BT 中，每个节点都表示一个行为，而这些节点是有层次的，子节点由父节点控制。每个节点的执行都有一个结果 (Success/Failure/Running)，父节点根据这些执行结果决定接下来的行动。节点不需要维护向其他节点的转换，其模块性得到了增强，让复杂的上位机系统能够高效、准确地执行决策逻辑。BT 部分节点类型如表 2.4 所示。

表 2.4 节点类型

Sequence	顺序执行所有子节点返回成功，否则失败
Loop	循环执行子节点到指定次数后返回成功
Condition	根据条件的比较结果，返回成功或失败
Action	根据动作结果返回成功，失败或运行
Wait	指定时间过去后则返回成功

在视频展示的 1V1 和 2V2 机器人对战中，机器人从 Blackboard 中获取裁判系统机器人血量、弹量、是否进入潜伏状态、底层 imu 数据等信息并与其他功能包进行信息交互；主机器人在比赛开始

后根据我方状态信息完成加弹、加血等任务，而副机器人则在我方障碍区躲藏等待，在补充子弹后进入敌方半场，配合主机器人对敌方两台机器人按照先后顺序进行双车夹击，保证在最短的时间内击杀敌方机器人；如果一台机器人阵亡，则活着的机器人完成所有行为逻辑；潜伏后依据哨岗信息继续完成对敌方机器人的装甲识别与追击。

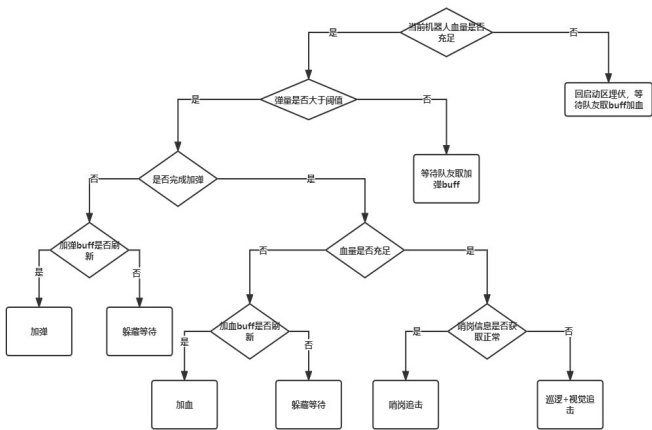


图 2.14 潜伏状态前机器人主逻辑

2.6.2 潜伏模式敌我识别

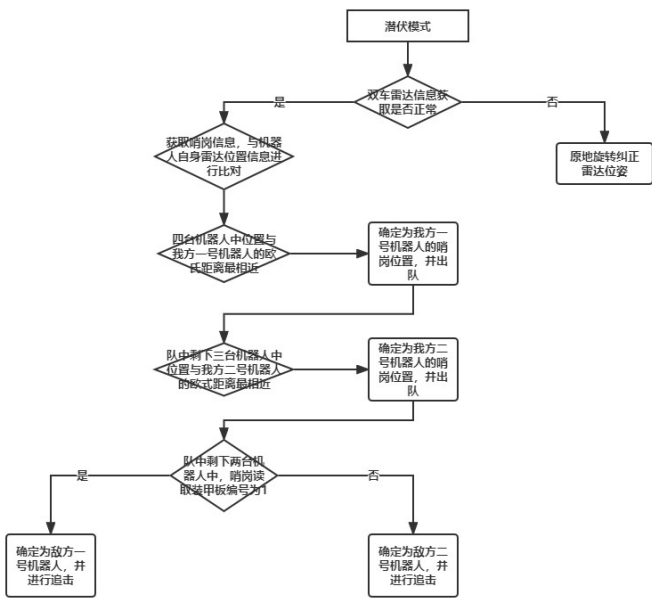


图 2.15 潜伏状态下敌我机器人识别

2.6.3 识别算法

敌我识别部分通过综合哨岗机器人位置信息与我方机器人雷达定位信息来完成。首先，接收我方哨岗发送的场上四台机器人位置信息并存入队

列，并给四个位置信息打上 label，便于之后判断处理。然后，无线通信接收友方机器人雷达定位信息，并计算队列中的位置信息与我方两台机器人的位置信息的欧氏距离。先筛选出与我方一号机器人雷达定位欧氏距离最小的机器人的 label，将其出队并赋值，确认为我方一号机器人；接着在队中剩下三台机器人中筛选出与我方二号机器人雷达定位欧氏距离最小的机器人的 label，将其出队并赋值，确认为我方二号机器人；接着在队中剩下两台机器人中，通过哨岗筛选出装甲板编号为 1 的，确认为敌方 1 号机器人，剩下的一辆自动确认为敌方 2 号机器人，然后对敌方机器人进行追击。

2.7 敌我识别

2.7.1 传感器

根据 2022ICRA 比赛规则要求，在正常比赛场景下，双方车辆接入裁判系统，在时间达到 130s 时车的装甲板灯条以当前方颜色开始闪烁，135s 进入潜伏状态，即双方随机抽取一辆车装甲板灯条变换为敌方颜色，潜伏状态一直持续到比赛结束，变色的部分只有装甲板灯条，RFID 扫描部分和血条不变色。针对这种情况哨岗无法做到百分百区分敌我，而且对于视觉来说也是一项挑战，如何获取己方车辆的装甲板颜色变化，就成了必须要解决的问题，对此决定选用颜色传感器 TCS3472 对灯条颜色进行判断识别，传感器主要参数如下所示：

表 2.5 TCS3472 参数

颜色测量范围	供电电压	通讯方式	频率
RGB(0-255)	3.3V	IIC	10hz



图 2.16 TCS3472 传感器接线图

传感器是根据不同颜色光的波长和频率不同，

通过对其频率和波长进行检测,从而得出当前颜色中 RGB 三原色所占比例,最终确定当前颜色,该传感器集成红外阻挡滤光片可最大限度地减少入射光的红外光谱部分,从而极大的减少了日光灯和太阳光的干扰,装甲板的红蓝两色均为三原色,其特征差异明显,测量简单容易。经过多次实验和人为制造干扰以及模拟比赛中可能出现的情况,软件上进行优化升级,硬件上增加防护,最终可以保证准确获得己方装甲板颜色。

2.7.2 潜伏处理

决策节点接收下位机发送的颜色传感器数据,并以进入潜伏模式作为标志位,在标志位置 1 后判断颜色传感器的数据(0/1)来确认我方机器人当前颜色并反馈给视觉节点,视觉筛选出友方装甲板颜色与编号,确保我方机器人准确识别敌方机器人装甲板颜色与编号并且不攻击友方机器人。

2.8 双车通讯

上场的两台机器人具有互相通信辅助决策的功能,我们使用 UDP 通信协议来实现双车通信。UDP 通信协议传输数据不建立连接,不需要维护连接状态和收发状态,便于小数据包高速传输,适合双车通信的场景。我们将接收消息和发送消息封装在两个不同的 ros 功能包中,防止因接收不到消息造成堵塞影响后续功能。

在收发的数据包中,包含雷达定位得到的自身在场地中的位置,和决策的标志位,进入潜伏之前,决策层通过读取双车通信的功能包广播出来的队友位置和决策状态,进行夹击,追击等一系列动作。进入潜伏之后,再结合哨岗信息进行匹配,得到场上机器人颜色变化情况,进而判断追击目标。

2.9 可视化交互界面

为便于调试,基于 ros_qt 搭建了可视化交互平

台,显示裁判系统信息、传感器信息、视觉识别、控制指令等信息,如图 2.17 所示。



图 2.17 qt_ros 可视化界面

2.10 仿真平台

使用 Gazebo 对步兵机器人的全向移动模型进行仿真,如图 2.18 所示,通过键盘节点发布指令控制机器人移动,并基于此辅助测试导航部分路径规划算法。

使用 ROS Stage 对场地与红蓝双方四台机器人进行仿真,如图 2.19,所示,验证决策部分行为树逻辑。蓝方机器人获取仿真地图信息,在完成加弹、加血的行为后,对红方机器人机器人进行追击。可以使用鼠标拖动红方机器人,对蓝方机器人规避惩罚区并前往加成区,根据全局信息追击敌方等行为进行仿真测试;

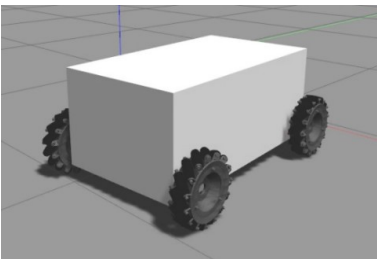


图 2.18 gazebo 仿真图

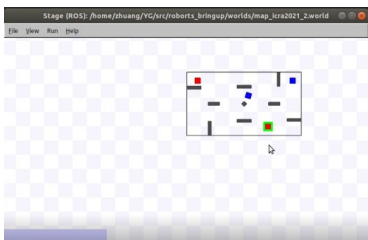


图 2.19 stage 仿真图