

# RoboMaster2022 机甲大师高校人工智能挑战赛

浙江大学 SWIFT 战队



浙江大学 湖州研究院  
HUZHOU INSTITUTE OF ZHEJIANG UNIVERSITY



## 目录

1. 硬件 .....	2
1.1 机械结构 .....	2
1.2 传感器 .....	3
1.3 计算设备 .....	3
2. 软件 .....	3
2.1 定位 .....	4
2.1.1 基于已知地图的全局定位 .....	4
2.1.2 基于二维码的偏航角确定 .....	4
2.1.3 基于状态的自身快速重新定位 .....	5
2.1.4 算法性能 .....	5
2.2 运动规划 .....	6
2.2.1 基于 MINCO 的轨迹优化 .....	6
2.2.2 状态机和性能评估 .....	7
2.3 自动识别 .....	7
2.3.1 装甲板识别与位姿解算 .....	7
2.3.2 目标状态估计 .....	8
2.4 自动射击 .....	9
2.5 哨岗观测 .....	9
2.6 敌我识别 .....	10
2.7 智能决策 .....	10
2.8 可视化仿真调试系统 .....	11
3. 演示视频 .....	12
4. 参考文献 .....	12



## 1.2 传感器

选用中距激光雷达 OusterOS1-64 激光雷达用于定位系统，其采样频率高、视场角广、精度高，并自带六轴 IMU，适合方案采用的 IMU 与激光雷达紧耦合定位算法。选用迈德威视 MV-SUA133GC-T 工业相机用于自动瞄准识别、二维码识别、哨岗识别，分别搭配 6mm、4mm、2.8mm 镜头。两者参数如图 4 所示。

室内探测距离 (80% 反射率)	0.25 m - 105 m	快门类型	全局快门
室内探测距离 (10% 反射率)	0.25 m - 60 m	分辨率	1280*1024
测距分辨率	0.3cm	镜头焦距 (自瞄)	6mm
垂直分辨率	64线	镜头焦距 (二维码)	4mm
水平分辨率	1024	镜头焦距 (哨岗)	2.8mm
视场角	45°*360°	最大帧速率	245FPS
频率	20Hz	曝光时间(ms)	0.00194~31

图 4 激光雷达参数 (左) 工业相机参数 (右)

## 1.3 计算设备

机载上位机选用英特尔 NUC-幻影峡谷，搭载 11 代酷睿 i7-1165G7CPU，16G 内存，前后的扩展非常丰富，包括 2 给雷电 4 接口、6 个 USB 3.1 Type-A、2.5G 有线千兆网口等，适合作为连接多传感器的车载上位机使用。

哨岗计算平台选用英特尔 NUC-X15 笔记本，搭载 Core i7-11800H CPU，RTX3070 显卡，32G 内存，适合神经网络的运行和训练。

## 2. 软件

软件部分包括定位，规划，识别，射击，哨岗，决策等模块，框图如图 5 所示。

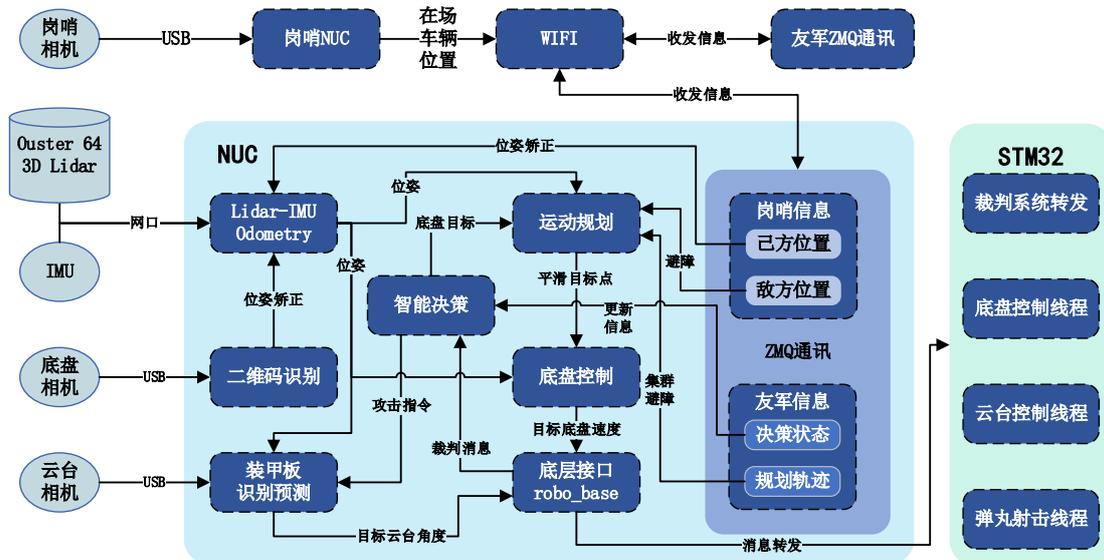


图 5 系统软件框图

## 2.1 定位

三维定位算法借鉴了 FAST-LIO<sup>[1]</sup>激光雷达惯性里程计的框架，并针对比赛场景，适应性设计了一种基于全局已知地图、可纠正系统状态量，能够快速重新恢复定位的里程计框架。整体来说，系统采用紧耦合的迭代拓展卡尔曼滤波器，可将雷达扫描出来的点云特征信息与 IMU 数据进行融合，然后与预先通过 ikd-Tree<sup>[3]</sup>导入的全局已知点云地图匹配。整个定位框架能够实现小车在快速、激进的运动当中的鲁棒定位。此外如果系统检测到漂移，可利用哨岗提供的位置信息与二维码识别提供的小车偏航角信息来恢复定位。系统的整体框架如图 6 所示：

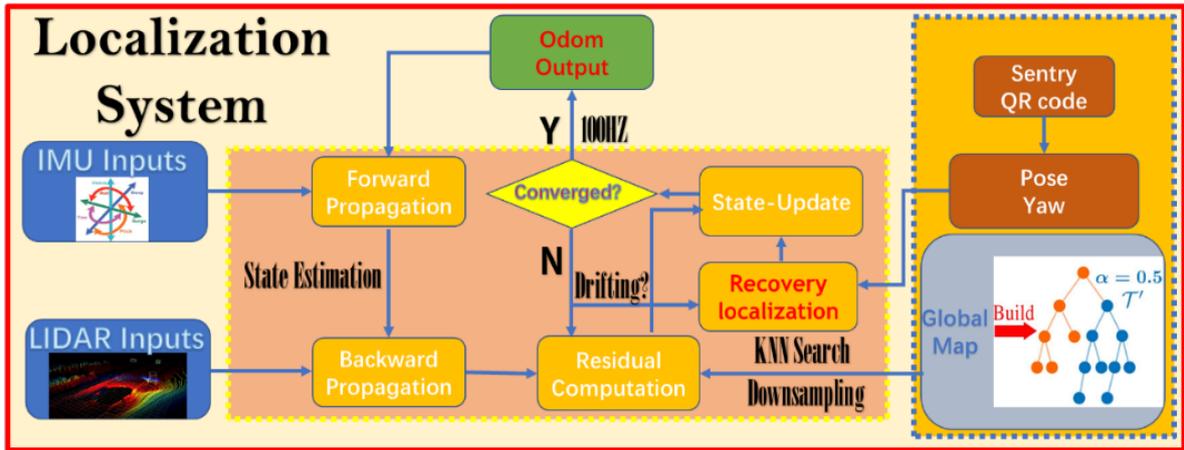


图 6 定位系统框架

### 2.1.1 基于已知地图的全局定位

采用基于预先已知地图的激光雷达惯性里程计框架。使用三维雷达可以预先建立全局场地信息的点云 PCD 文件，作为点云配准获取里程计的先验地图。定位系统在此基础上通过前向传播过程，即利用 IMU 预积分模型，估计系统状态，方差等直至一个新的雷达扫描周期，得到传播状态和协方差。由于一系列雷达数据特征点点云并非同一时刻测量的，这些特征点中存在着相对运动，为此利用前述前向传播过程当中的传播状态和协方差对其进行运动补偿，去除雷达点云在扫描周期当中因为小车快速剧烈的运动产生的畸变，补偿相对运动造成的运动失真。其后利用残差分别与前向传播中得到的估计状态融合，并从中求得最优的状态估计即定位系统的输出里程计信息。如果系统检测到因为剧烈运动产生定位漂移，便可利用哨岗与二维码信息恢复定位，更新系统状态，保证定位效果的鲁棒性。

### 2.1.2 基于二维码的偏航角确定

为实现定位偏航角计算辅助检测是否需要重新定位，增强定位鲁棒性，基于 Aruco 库实现对赛场上特定位置的字母 Marker 的识别检测实现小车偏航角的确定。赛场上字母 Marker 的位

置与位姿固定，当固定在车身上的工业相机拍摄到在赛场上特定位置 Marker 图片时，为提高识别效率，先降采样，再通过读取 IMU 数据判断此时车的运动导致的图片运动模糊程度，根据该数据设计维纳滤波器对运动模糊图像进行复原，之后再通过 Opencv 采用通道分离，开闭操作，滤波，二值化处理等操作提高识别鲁棒性。处理后的图片由 Aruco 库函数根据初始化的 Marker 数据，进行分割，提取轮廓操作识别出 Marker 的 id，通过提取角点采用 PNP 算法进行位姿估计得到小车旋转矩阵  $R$ 。进行坐标转换后便可计算出小车在赛场上的偏航角。

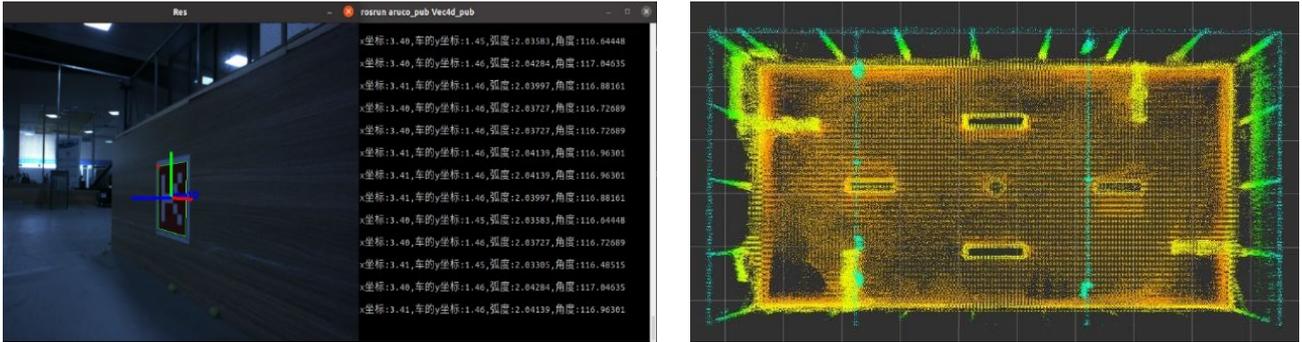


图7 二维码识别（左） 已知全局地图点云（右）

### 2.1.3 基于状态的自身快速重新定位

在仅仅有激光雷达惯性里程计的定位方法中，有可能因为暴烈的旋转和强力的车体震荡而出现定位漂移/失败的现象，故设计了基于外部的、准确的、全局的信息来帮助恢复定位。上述二维码偏航角定位因镜头视野捕获的问题容易丢失信息不能连续提供，所以通过融合底盘的旋转四元数信息来得到一个持续的且相对准确的高频率的车体旋转角信息，即由 Aruco 码识别的位姿信息中旋转部分来矫正底盘 Odom 的旋转四元数。场地中哨岗的可提供全局的位置关系以及速度信息，频率为 20Hz 左右。通过自身状态判断来确定是否需要重置迭代拓展卡尔曼滤波器维护的状态量，以恢复激光雷达惯性里程计自身的定位。若重置状态量，位置和速度信息来自哨岗，旋转信息来自 Aruco 码识别。

### 2.1.4 算法性能

整个定位系统能够实现随机位置初始化，即简单的通过 rviz 可视化界面通过 2D Pose Estimate 插件发布/intialpose 话题初始化状态量实现快速定位。小车在快速激进，旋转碰撞下也能够获得稳健良好的定位信息，静止状态下定位误差为厘米级别，提供的高频率即 100HZ 的定位信息能够为小车的规划，控制提供良好的保障。重新纠正系统定位的策略能够提高系统的鲁棒性。

## 2.2 运动规划

运动规划算法分为三部分，分别为 1) 基于 ESDF 的先验地图构建<sup>[4]</sup>；2) A\*<sup>[5]</sup>前端路径搜索；3) 基于 MINCO 的轨迹优化<sup>[6]</sup>。其中最后一部分是规划的重点内容。

赛场中的静态障碍物全局已知，在初始化阶段存储一个较为稠密的全局栅格地图，并计算欧式截断距离场 ESDF(Euclidean Signed Distance Field)。它轻量化、效率高、产生趋向最优的结果。ESDF 适合于基于梯度下降的优化运动规划，可以将机器人轨迹“推离”障碍物。A\*算法是机器人领域常用的路径搜索算法，它相较于 Dijkstra 算法、广度优先搜索算法在这个场景下具有更高的效率。在本次比赛的运动规划任务中，它作为前端，用以找到无碰的初始路径。

### 2.2.1 基于 MINCO 的轨迹优化

该方案使用 MINCO 表示轨迹，是关于时间的分段多项式函数。

$$\mathfrak{T}_{MINCO}: \{p(t): [0, T] \mapsto \mathbb{R}^m \mid \mathbf{c} = \mathbf{c}(\mathbf{q}, \mathbf{T})\}$$

是一种稀疏轨迹表示，其中  $\mathbf{q} = [q_1, \dots, q_{M-1}]$ ,  $q_i \in \mathbb{R}^m$  是轨迹中路径点的位置信息，不包含始末状态， $m$  表示位置空间的维度， $M$  表示轨迹中的分段数； $T = \sum_1^M T_i$ ,  $\mathbf{T} = [T_1, \dots, T_M]^T$ ,  $T_i \in \mathbb{R}_{>0}$  是每个轨迹段的时间； $\mathbf{c} = [c_1^T, \dots, c_M^T]^T$ ,  $\mathbf{c} \in \mathbb{R}^{2Ms \times m}$  是多项式系数矩阵， $s$  表示控制量阶次，当  $s=0$  时表示控制量为位置，该方案中  $s=3$ ，控制量为 jerk。由于闭式解的存在， $\mathfrak{T}_{MINCO}$  可以保证轨迹  $s-1$  阶连续，并且最小化控制作用量。在本比赛中，轨迹优化的数学描述如下：

$$\begin{aligned} \min_{u_j(t), \forall j} H = & \sum_{j=1}^M \int_0^{T_j} u_j(t)^T u_j(t) dt + \rho T \\ & + \underbrace{W_v \sum_{j=1}^M \int_0^{T_j} \mathcal{K}(\|p_j^{(1)}(t)\|_2^2 - v_{max}^2) dt}_{\text{最大速度惩罚}} + \underbrace{W_a \sum_{j=1}^M \int_0^{T_j} \mathcal{K}(\|p_j^{(2)}(t)\|_2^2 - a_{max}^2) dt}_{\text{最大加速度惩罚}} \\ & + \underbrace{W_c \sum_{j=1}^M \int_0^{T_j} \mathcal{C}(d_{safe} - d(p_j(t))) dt}_{\text{碰撞惩罚}} + \underbrace{W_d \sum_{j=1}^M \sum_{k=1}^N \int_0^{T_j} \mathcal{K}(d_{dyn}^2 - \|p_j(t) - \hat{p}_k\|_2^2) dt}_{\text{动态障碍碰撞惩罚}} \\ & + \underbrace{W_s \sum_{j=1}^M \int_0^{T_j} \mathcal{K}(d_{swa}^2 - \|p_j(t) - \bar{p}(t + t_0)\|_2^2) dt}_{\text{集群碰撞惩罚}} \\ \text{s.t. } & u_j(t) = p_j^{(s)}(t), \forall t \in [0, T_j]; \quad T_j > 0, \forall j \in 1, \dots, M; \\ & \underbrace{p_j^{[s-1]}(T_j) = p_{j+1}^{[s-1]}(0), \forall j \in 1, \dots, M-1;}_{\text{连续性约束}} \quad \underbrace{p_1^{[s-1]}(0) = \bar{p}_0, p_M^{[s-1]}(T_M) = \bar{p}_f}_{\text{初末状态约束}} \end{aligned}$$

其中  $\rho$  为优化时间的权重， $p_j(t)$  为每段轨迹位置对时间的函数， $W_v, W_a, W_c, W_d, W_s$  分别为各项惩罚系数， $d(x)$  为 ESDF 函数， $\mathcal{K}(x) = \max(x, 0)^3$  为立方惩罚， $\mathcal{C}(x)$  为 ReLU 惩罚：

$$C(x) = \begin{cases} -0.5\mu, & x < 0 \\ x - 0.5\mu, & 0 \leq x \leq \mu \\ (\mu - 0.5x) \cdot \left(\frac{x}{\mu}\right)^3 & x > \mu \end{cases}$$

其中， $\mu$ 为参数，本任务中取 $\mu = 0.01$ 。 $d_{safe}, d_{swa}, d_{dyn}$ 分别为静态避障、集群避障和动态避障安全距离， $N, \hat{p}_k$ 分别为动态障碍物数量和位置， $\tilde{p}, t_0$ 分别为友车轨迹和时间差。由 $\mathfrak{T}_{MINCO}$ 的特性，约束得以消解，最终利用 L-GFBS 算法求解。

### 2.2.2 状态机和性能评估

为了确保轨迹的安全性，方案以 20Hz 的频率对轨迹的安全性约束进行检查，并以 100Hz 的频率进行重规划。经测试，轨迹生成最长用时 2.9929ms，平均 9.453ms，车辆轨迹跟踪精确，且全程无碰撞。可见该运动规划系统运算计算资源占用低，且生成的轨迹在确保高机动性的情况下满足动力学和运动学约束。

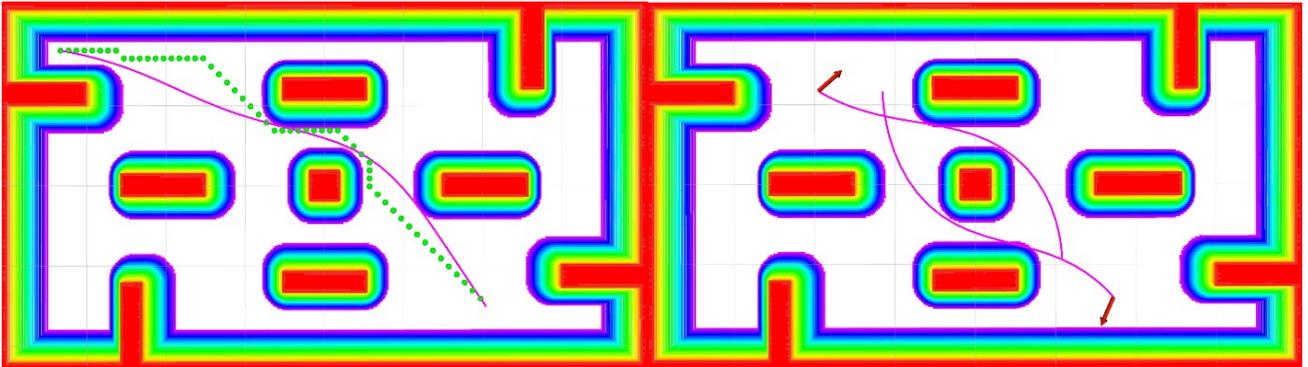


图 8 单车运动规划（左）多车运动规划（右）

## 2.3 自动识别

### 2.3.1 装甲板识别与位姿解算

装甲板识别选用 MV-SUA133GC-T 工业相机，搭配 6mm 镜头，曝光时间设为 0.8ms，极高的全局抓图帧率使得目标在高速运动的条件下也能保持清晰的输入图像，使得可以对高速运动目标稳定识别。采用基于 OpenCV 的装甲板传统识别算法，以识别蓝方装甲板为例，首先在图像预处理模块，将相机彩色图片分离为红、绿、蓝三通道，将蓝色通道和绿色通道做差后进行二值化，并将原彩色图片处理为灰度图片后进行二值化，将两张二值化后图片进行与运算后，输入灯条检测模块。灯条检测模块通过 `cv::findContours()` 函数找出所有可能的灯条，并通过倾斜角度、长宽比进行初步筛选，将筛选后的所有可能灯条输入灯条匹配模块进行匹配，满足灯条距离、角度差、长度比例、装甲板角度等设定条件即判断为装甲板，最后在装甲板检测模块

选择出目标装甲板进行 PnP 解算，得到目标在相机坐标系下的位姿，输出给后续的状态估计模块。

经测试，装甲板识别算法对红蓝装甲板在不同光照条件、不同运动状态下均能稳定识别，距离测量误差在 5mm 以内。

### 2.3.2 目标状态估计

在解算出目标装甲板的位姿后，输入给状态估计模块进行状态估计。在比赛中，敌方机器人为高机动目标，需要在模型中考虑目标机动的表示。Singer 模型是被广泛使用的机动目标运动模型，相比使用高斯白噪声而言，Singer 模型更能够适应机动目标的运动，用有色噪声描述目标机动加速度，适合描述介于匀速直线运动与匀加速直线运动之间的机动行为。Singer 模型的离散状态方程表示为：

$$\begin{bmatrix} x(k+1) \\ v(k+1) \\ a(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & \frac{\alpha T + e^{\alpha T} - 1}{\alpha^2} \\ 0 & 1 & \frac{1 - e^{-\alpha T}}{\alpha} \\ 0 & 0 & e^{-\alpha T} \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \\ a(k) \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2\alpha} - \frac{\alpha T + e^{\alpha T} - 1}{\alpha^3} \\ \frac{\alpha T + e^{\alpha T} - 1}{\alpha^2} \\ \frac{1 - e^{-\alpha T}}{\alpha} \end{bmatrix} w(k)$$

其中， $T$ 表示采样周期， $\alpha$ 为机动频率， $w(k)$ 为零均值随机过程噪声序列， $v(k)$ 为零均值测量噪声序列。依据运动模型预测和相机识别观测，通过卡尔曼滤波器对装甲板进行状态估计。对于上式所描述的线性系统，卡尔曼滤波算法分为以下六步。

$$\begin{cases} \hat{x}_k = A_k \hat{x}_{k-1} + B_k w_k \\ \hat{P}_k = A_k \hat{P}_{k-1} A_k^T + Q_k \\ y_k = C_k x_k + v_k \\ K_k = \hat{P}_k C_k^T (C_k \hat{P}_k C_k^T + R) \\ \hat{x}_k = \hat{x}_k + K_k (y_k - C_k \hat{x}_k) \\ \hat{P}_k = (I - K_k C_k) \hat{P}_k \end{cases} \quad \text{其中，} \quad A_k = \begin{bmatrix} 1 & T & \frac{\alpha T + e^{\alpha T} - 1}{\alpha^2} \\ 0 & 1 & \frac{1 - e^{-\alpha T}}{\alpha} \\ 0 & 0 & e^{-\alpha T} \end{bmatrix} \quad B_k = \begin{bmatrix} \frac{T^2}{2\alpha} - \frac{\alpha T + e^{\alpha T} - 1}{\alpha^3} \\ \frac{\alpha T + e^{\alpha T} - 1}{\alpha^2} \\ \frac{1 - e^{-\alpha T}}{\alpha} \end{bmatrix} \quad C_k = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T$$

$R_k$ 为测量噪声协方差矩阵，可通过静态采集传感器数据求得， $Q_k$ 为过程噪声协方差矩阵，为对称矩阵，根据 Singer 模型推导出的过程噪声协方差矩阵如下式：

$$Q_k = \begin{bmatrix} \frac{1 - e^{-2\alpha T} + 2\alpha T + \frac{2}{3}\alpha^3 T^3 - 2\alpha^2 T^2 - 4\alpha T e^{-\alpha T}}{\alpha^4} & \frac{1 + e^{-2\alpha T} - 2e^{-\alpha T} + 2\alpha T e^{-\alpha T} - 2\alpha T + 2\alpha^2 T^2}{\alpha^3} & \frac{1 - e^{-2\alpha T} + 2\alpha T e^{-\alpha T}}{\alpha^2} \\ \frac{1 + e^{-2\alpha T} - 2e^{-\alpha T} + 2\alpha T e^{-\alpha T} - 2\alpha T + 2\alpha^2 T^2}{\alpha^3} & \frac{4e^{-\alpha T} - 3 - e^{-2\alpha T} + 2\alpha T}{\alpha^2} & \frac{1 + e^{-2\alpha T} - 2e^{-2\alpha T}}{\alpha} \\ \frac{1 - e^{-2\alpha T} + 2\alpha T e^{-\alpha T}}{\alpha^2} & \frac{1 + e^{-2\alpha T} - 2e^{-2\alpha T}}{\alpha} & 1 - e^{-2\alpha T} \end{bmatrix}$$

最后自动识别及状态估计的频率在 130Hz 以上，对距离 2m 的随机机动运动目标也能保持 70%以上的命中率。

## 2.4 自动射击

云台跟踪和发射控制的逻辑框图如图 9 所示：

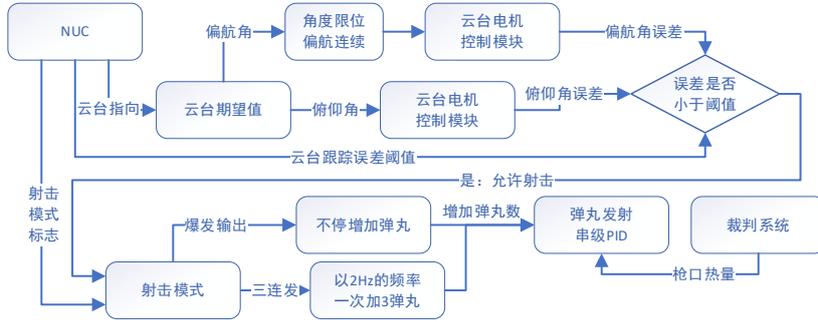


图 9 云台控制和弹丸发射逻辑框图

云台控制线程在接收到 NUC 的云台期望值之后，会对目标值进行限幅和连续性处理，之后送入带有微分跟踪器和串级 PID 的云台电机控制模块，对电机进行控制（图 10）。当误差小于容许阈值的时候，发射手线程会自动根据射击模式标志位对期望设计弹丸数进行增加。由弹丸发射的串级 PID 控制器对拨弹电机进行位置控制发射弹丸。



图 10 云台电机控制模块

经过测试，云台位置环的控制带宽可以达到 5Hz 以上，满足自动瞄准要求；拨盘电机响应速度快，最高可达到 17Hz 的射频。

## 2.5 哨岗观测

哨岗采用 YoloV5 中等模型，随机初始化参数进行训练，类别共有八种：原图车、转换视图车、红色 1、2 号装甲、蓝色 1、2 号装甲、红色灯条和蓝色灯条。总体方案框图如图 11。

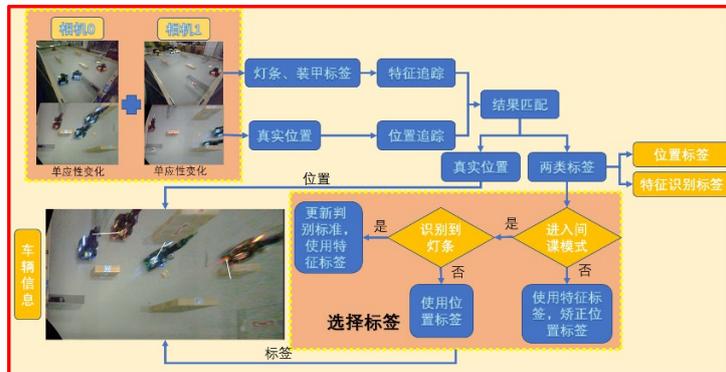


图 11 哨岗观测模块框图

位置追踪是根据距离最近的原则，不断更新每一个车的坐标，初始的位置和车辆标签信息由手动赋值。使用 YoloV5 识别到“转换视图车”框的中点作为车的真实坐标。首先得到当前帧两个相机的真实坐标，然后采用距离筛选同时识别到“转换视图车”并采用由更近距离相机给出的坐标，计算每一个筛选后坐标与上一帧所有车真实坐标的欧氏距离。以距离和最小为目标函数，将当前有效坐标与上一帧真实坐标匹配，并更新车的坐标，完成一轮位置追踪。

特征追踪部分首先分别对两个相机原图中的“原图车”、“灯条”、“装甲”框，使用最大 IOU（交并比）进行匹配，给“原图车”类赋予装甲和灯条信息，再取“原图车”框底边中点做单应性变换，得到近似坐标，然后将两个相机中带有标签的坐标根据距离最近进行匹配，进行信息互补，得到带有装甲和灯条信息的近似坐标，最后将近似坐标与位置追踪第一步筛选出的真实坐标采用距离最近原则进行匹配，得到带有特征标签信息的真实坐标，再与位置追踪结果根据距离最近进行匹配。

## 2.6 敌我识别

敌我识别的过程就是根据标签信息进行机器人类别判断的过程。经过位置追踪和特征追踪，可以得到每个车的位置标签和特征标签。其中位置标签是进行位置初始化时人工输入的标签，位置追踪实质是不断更新这些标签的位置信息。特征标签便是每一帧识别的真实结果，未进入间谍模式时，使用特征标签作为车辆判别标准，并按照一定频率矫正位置追踪标签。进入间谍模式后，若当前特征标签中存在灯条信息，则更新车辆判别标准（例如在特征标签中，装甲信息为“红2”，灯条为“蓝”，则认为带有“红2”装甲板的为“蓝2”）；若没有灯条信息，判断车辆判别标准是否更新，对更新了标准的车，使用特征标签判断，若没有更新则使用位置标签。

## 2.7 智能决策

在全面权衡之后，智能决策选择使用行为树。该方案系统简洁、便于验证与测试、适配人类经验，很适合此比赛所需。行为树需要维护赛场各方面信息，并根据信息决定机器人的行为。决策模块采用的行为树如图 12 所示。

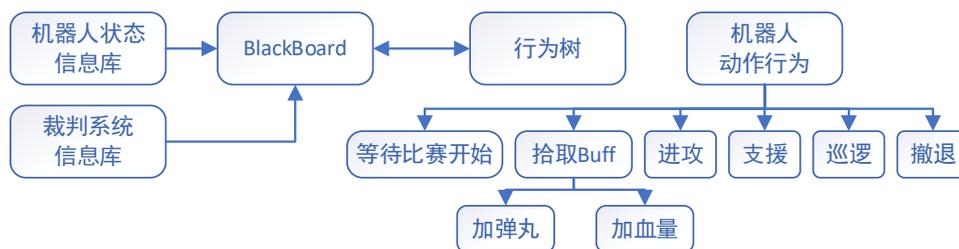


图 12 行为树示意图

在决策模块中，行为树维护两类信息：裁判系统发回的各项信息和机器人自身的状态信息；机器人的行为共设定 7 种：“等待比赛开始”、“拾取加子弹 buff”、“拾取回血 buff”、“巡逻”、“进攻”、“支援”和“撤退”。此外，行为树在我方两台战车中分别部署，形成分布式决策系统，同时，二台战车将共享一部分自身信息，以避免决策运动目标点冲突、实现夹击行为等功能。分布式决策系统使得我方一台战车出现故障时，最大限度降低了对另一台战车正常工作的影响。

在搭建的仿真平台上对决策系统进行了充分验证，我方两台机器人可以联合实现在比赛中的稳定自主决策。

## 2.8 可视化仿真调试系统

仿真环境与可视化界面在机器人行为调试与决策制定中十分重要，可以大幅提高调试速度、提前把握实机运行效果、降低调试成本。我们基于 PyGame 搭建了一个针对 RMUA2022 比赛规则的仿真系统。

该系统可实现车辆运动学模拟、比赛模拟、裁判系统模拟、可视化编辑与交互、定制 ROS 接口等功能，在我们的轨迹规划器设计、控制器设计、决策树设计和后期实车调试的过程中发挥了重要作用。可视化仿真调试系统界面分为地图界面与操作界面，下图展示了我们的可视化仿真调试系统及其元素说明，图中画面截选自一场正在进行的 2v2 仿真比赛。

除了进行仿真模拟赛，我们的可视化仿真调试系统还提供了真实赛场监视功能，通过接收实车发布的 ROS 话题，实现实车调试时的车辆位置、状态、轨迹、行为意图监测，大大降低了实车调试时 bug 定位的耗时，提高了工作效率。



图 13 可视化仿真调试系统框图

### 3.演示视频

优酷视频链接: [https://v.youku.com/v\\_show/id\\_XNTg2MjAzMjYwMA==.html](https://v.youku.com/v_show/id_XNTg2MjAzMjYwMA==.html) 密码 swift

油管视频链接: <https://youtube.com/watch?v=H-zGoG78eIw> 私密连接 (无密码)

### 4.参考文献

- [1] Xu W, Zhang F. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 3317-3324.
- [2] Xu W, Cai Y, He D, et al. Fast-lio2: Fast direct lidar-inertial odometry[J]. arXiv preprint arXiv:2107.06829, 2021.
- [3] Cai Y, Xu W, Zhang F. ikd-Tree: An Incremental KD Tree for Robotic Applications[J]. arXiv preprint arXiv:2102.10808, 2021.
- [4] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," Theory of computing, vol. 8, no. 1, pp. 415–428, 2012.
- [5] Hart P E , MEMBER, IEEE, et al. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science and Cybernetics, 2007, 4(2):100-107.
- [6] Wang Z, Zhou X, Xu C, et al. Geometrically constrained trajectory optimization for multicopters[J]. arXiv preprint arXiv:2103.00190, 2021.