



Using a BL-HS motor driver module and Field-Oriented Control (FOC), the RoboMaster G30 Brushless DC Motor Speed Controller enables precise control over motor speed.



Exclusively designed for the RoboMaster series, the BL-HS Brushless DC Motor and G30 Brushless DC Motor Speed Controller, the BL-HS Assembly Kit includes several cables and a terminal board.

RoboMaster Reference System Specification Manual, RoboMaster System User Manual, Introduction of RoboMaster System Module

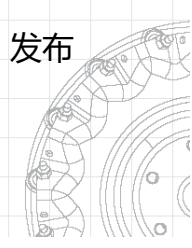
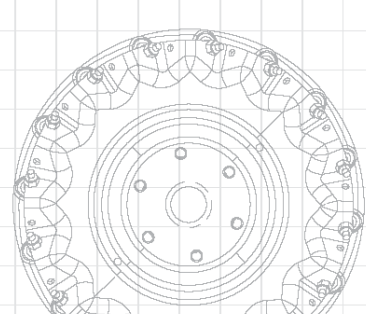
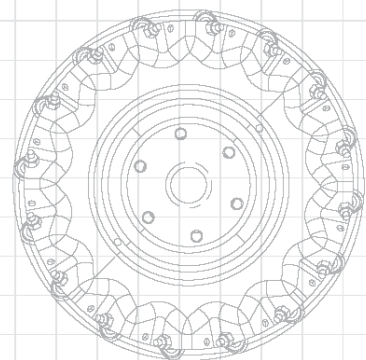
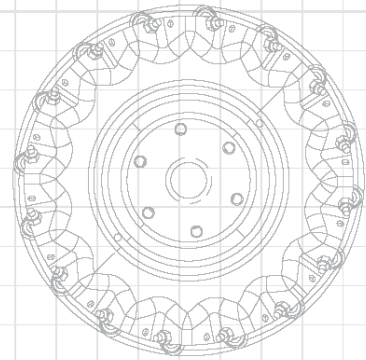
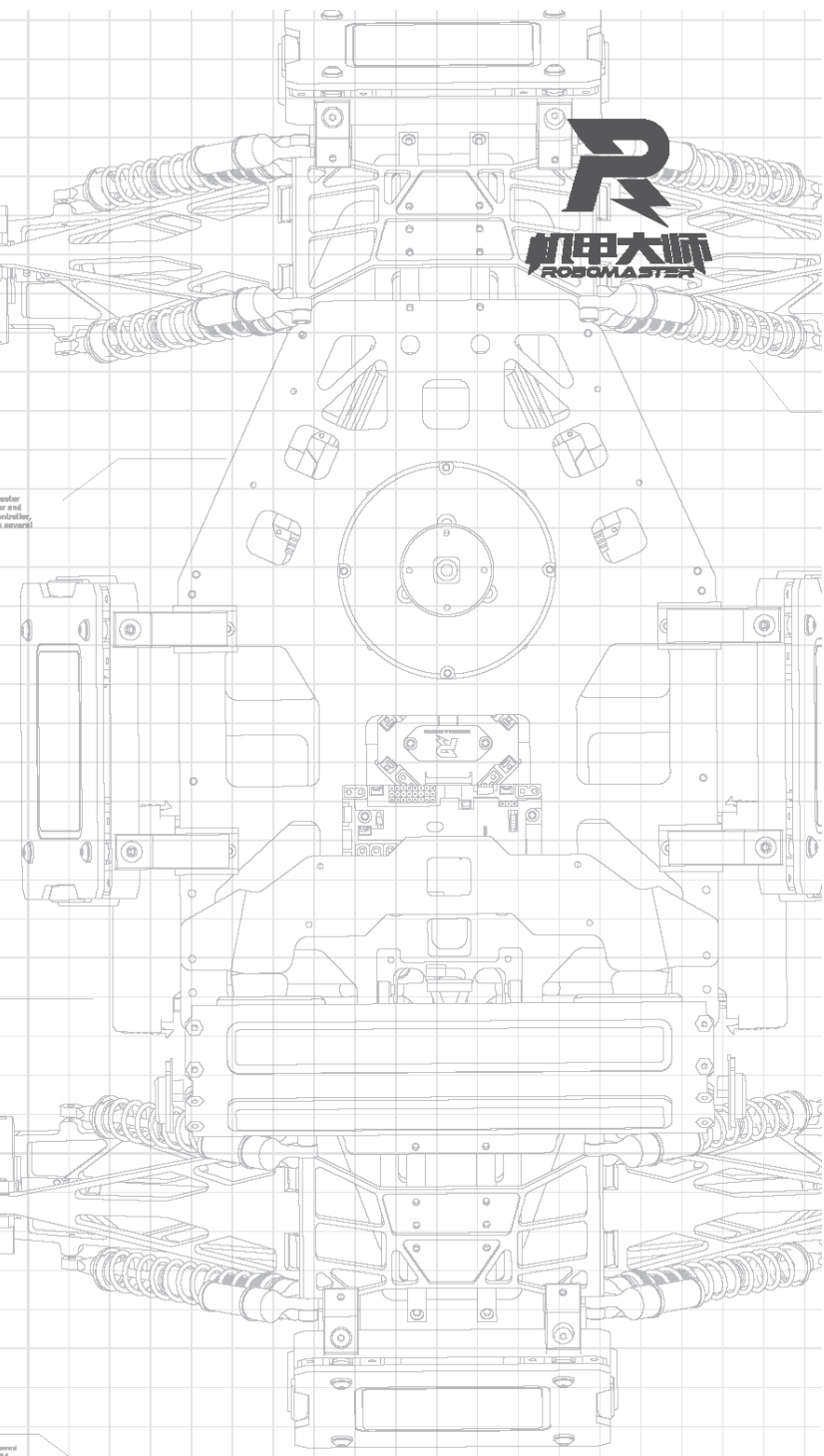
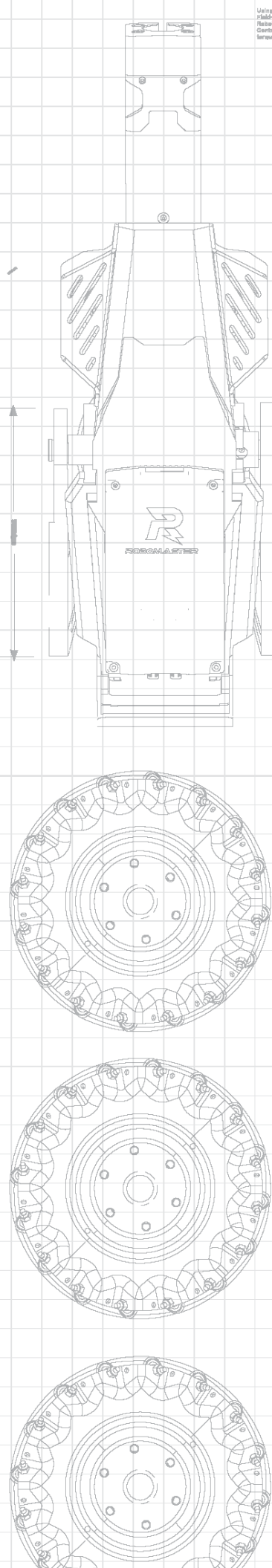
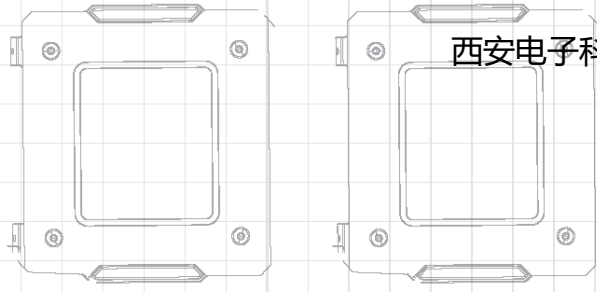


The M3090 Assembly Kit includes several cables and a terminal board, enabling a complete robotic system driven by four independent systems.

ROBOMASTER 机甲大师超级对抗赛 技术方案

西安电子科技大学 IRobot 战队 编制

2023 年 8 月 发布



前言

本文档由西安电子科技大学 IRobot 战队编制,适用于 RoboMaster 2023 机甲大师超级对抗赛。
主要撰写人员包括:

模块	撰写人员 1	撰写人员 2
机械	卢宇豪	任朔
硬件	祁灵	
软件	祁灵	
算法	张必豪	池威律
其他	张必豪	卢宇豪

目录

前言.....	2
1. 概述.....	5
1.1 背景与目标.....	5
1.2 其它学校机器人分析综述.....	5
1.3 机器人功能定义.....	6
1.4 机器人核心参数.....	7
1.4.1 机器人基础参数.....	7
1.4.2 主要传感器.....	7
1.4.3 执行器件.....	8
1.4.4 机器人性能参数.....	9
2. 设计方案.....	10
2.1 机械结构设计.....	10
2.1.1 云台模块.....	10
2.1.2 底盘模块.....	15
2.2 硬件设计.....	18
2.3 软件设计.....	19
2.3.1 系统架构.....	19
2.3.2 运行流程.....	19
2.3.3 重点功能.....	20
2.4 算法设计.....	24
2.4.1 自瞄算法设计.....	24
2.4.2 定位导航算法设计.....	32
2.4.3 决策算法设计.....	37
2.5 其它.....	39
3. 研发迭代过程.....	40
3.1.1 测试记录.....	40
3.1.2 版本迭代过程记录.....	42
3.1.3 重点问题解决记录.....	44
4. 团队成员贡献.....	47
5. 参考文献.....	48
6. 技术方案复盘.....	49

6.1.1 赛场性能表现情况分析	49
6.1.2 赛场性能表现与规划对比分析	49
6.1.3 经验总结	49

1. 概述

1.1 背景与目标

哨兵机器人为全自动机器人，具有比其他机器人更优的比赛参数：前期无敌，发弹量高，功率大，血量厚。并且哨兵机器人作为基地的最后一道防线，对于比赛的输赢具有至关重要的作用。哨兵机器人可以通过定位导航算法实现在赛场上的灵活移动，摧毁前哨站，追杀敌方机器人为战队带来前期的优势。但是如果在己方前哨站被摧毁的情况下，哨兵不能及时回到巡逻区，基地护甲将会展开，所以对于哨兵的定位导航系统的稳定性要求很高。而且由于新赛季哨兵没有操作手实时操控需要自主做出决策，因此我们需要编写“决策代码”使哨兵知道什么时候需要移动导航，什么时候需要旋转小陀螺，什么时候需要开火反击。

在机械方面，为了保证哨兵不被其他机器人卡住，导致导航异常，需要特制边框防止被其他机器人卡住。由于规则允许最多拥有两个发射机构，云台设计时要充分利用双发射机构优势，让哨兵爆发能力充分发挥。

1.2 其它学校机器人分析综述

● 华南农业大学哨兵分析

从 22 赛季自动步兵技术报告和 23 赛季开源哨兵资料，可以看出华农的哨兵具有很完善的机器人系统，哨兵的导航及决策击打效果很好。

华农的哨兵采用双单线雷达感知的结构，通过离线的标定将两个雷达合并的扫描范围设成为 360 度。

- 定位算法：使用 IMU 和里程计得到一个相对准确的相对定位信息，同时使用 scan-to-map 的匹配算法，将激光雷达扫描数据直接与地图进行匹配，获得机器人的姿态。然后使用 EKF 将这个坐标和里程计和 IMU 数据做融合，最终获得一个准确的定位信息。
- 导航算法：采用 move-base 中的全局规划算法框架，重写了速度控制器，使用位置 PID 进行速度规划，到达路径跟踪效果。
- 击打决策：采用以多参数状态机切换机器人的行为，针对机器人的击打和防御建立模型：最优攻击点与防御点的获取通过采样地图上多个离散的目标点，根据目标函数（敌我距离，直线距离间是否有障碍物等约束条件）计算得出权值最高的目标点，随后导

航到目标点。

● 东北大学 T-DT 战队哨兵分析

本赛季东北大学的哨兵在比赛的前期都发挥着很重要的作用，对敌方进行火力压制。哨兵采用上下双云台的设计方案，对高地和地面的机器人都能做到很好的识别和击杀。双云台结构对于导航和自瞄的控制自由度都比较高，能够有一个很好的表现效果。

● 哈尔滨工业大学 I Hiter 战队哨兵分析

从青工会的开源中，学习到哨兵的全向感知设计。他们的哨兵也采用上下双云台的机械机构，将激光雷达和 3 个 USB 相机放置在上云台上面，有很好的全局视野。哨兵的决策通过裁判系统信息，全局视野的敌方相关信息，辅瞄目标相关信息，定位相关信息等输入，根据云台手按键进入不同的模式，输出哨兵运动目标点和运动模式。并且哨兵具有对目标兵种（英雄）跟踪击打的效果，只要能够正常感知和锁住，就能将敌方机器人击杀。

1.3 机器人功能定义

■ 底盘功能设计

- ◆ 全地形下不会出现翻车
- ◆ 有小陀螺功能
- ◆ 弹舱可以容纳 750 发弹丸
- ◆ 机动性强
 - 四驱独立悬挂
 - 可以实现上坡

■ 云台设计

- ◆ 使用二轴云台，pitch 轴配平，保证射击稳定性
- ◆ 切枪管设计

■ 射击系统

- ◆ 实现下供弹链路
- ◆ 视觉自瞄自主决策射击
 - 在不同距离射频不同
- ◆ 可维持高射频持续射击
 - 射频维持在 8Hz
 - 持续射击 20s
- ◆ 系统稳定

- 连续射击 1000 发弹丸的卡弹率为 0
- 最大射速稳定接近最高速 30m/s
- 射速方差不大于 1

■ 导航系统

- ◆ 可以边扫描敌人边导航
- ◆ 实现自主避障
- ◆ 可以到达地图任意位置

■ 其他功能

- ◆ 可以响应云台手指令改变攻击策略
- ◆ 便于维护
 - 核心模块损坏 5 分钟内可修复
- ◆ 稳定性高，不易损坏
- ◆ 整体外形美观，云台外型添加龙头元素
- ◆ 有便于视觉挂载屏幕的位置

1.4 机器人核心参数

1.4.1 机器人基础参数

参数类型	数值
尺寸 (mm, L*W*H)	685*685*600
重量 (kg)	25.8
重心高度 (mm)	230

1.4.2 主要传感器

型号	参数	数量
BMI088	BM1088 集成了 16 位三轴陀螺仪和 16 位三轴加速	2

型号	参数	数量
	度计。具有低于 2 / h 的偏置稳定性和低于 15mdps/K 的低温度系数偏移。	
Livox-Mid 360 激光雷达	近处盲区 0.1 m, FOV 水平 360°, 垂直 -7° ~52°, 测距随机误差 ³ (1σ) ≤ 2 cm (@ 10m) ≤ 3 cm (@ 0.2m)	1
工业相机	大恒水星一代	1

1.4.3 执行器件

类型	用途	数量
M3508	用于轮组提供动力	4
M3508 (拆除减速箱)	用于摩擦轮	2
GM6020	用于 yaw 轴及 pitch 轴的动力输出	2
RoboMaster M2006 P36 直流 无刷减速电机	用于拨盘	1

1.4.4 机器人性能参数

参数类型	数值
车体最大移动速度	2.5m/s
云台自由度	Yaw360° ; pitch-20° ~60°
最大射速	30m/s
弹舱容量	750 发

2. 设计方案

2.1 机械结构设计

2.1.1 云台模块

云台模块轴测图以及侧视图如下：

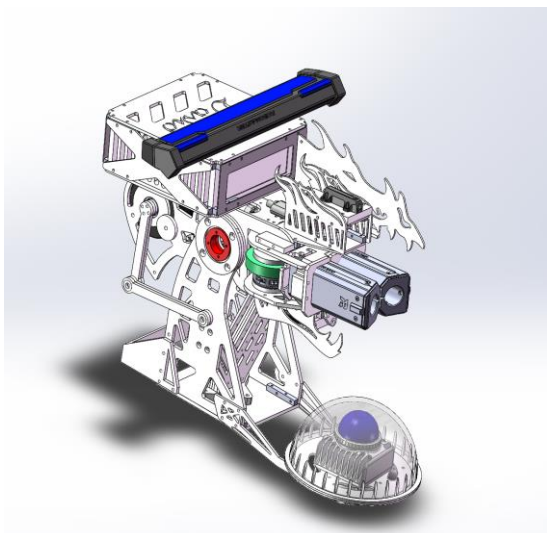


图 2.1.1 云台侧视图

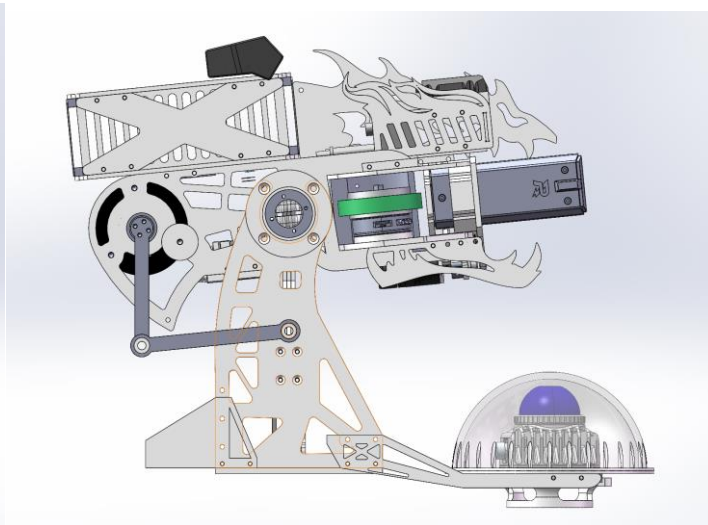


图 2.1.2 云台轴测图

1. 发射机构

根据 Robomaster2023 机甲大师高校系列赛机器人制作规范手册的规定，哨兵机器人最多拥有两个发射机构。

为了能充分利用双发射机构的优势，在分区赛版本哨兵采用并列双枪管设计，如图 2.1.3。纵置的两套并列发射机构可以同时进行发射，这样设计可以最大限度的发挥双发射机构的爆发能力，但是其结构复杂，需要额外的一套供弹链路，会大幅增加云台重量，导致云台响应速度变慢，控制精度变差。同时，由于并列双发射机构的两个发射机构存在一定距离，两个发射机构发射的弹丸会产生两个散布中心，不利于视觉调整自瞄补偿，降低弹丸利用率。

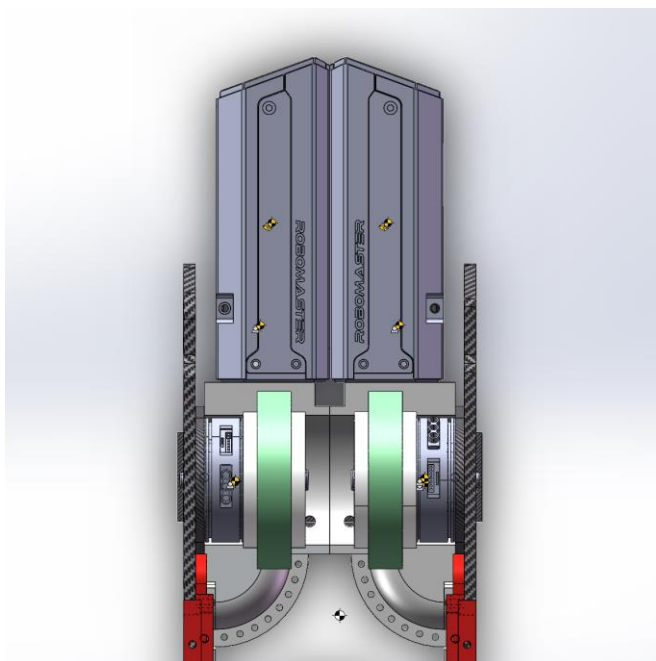


图 2.1.3 并列发射机构

鉴于分区赛时，并列双发射机构没有很好的发挥爆发优势，且哨兵本身射频较高即使单发射机构爆发能力也较强，在设计全国赛版云台决定减少一组发射机构，换取更快的云台响应和控制精度。同时，为了利用双测速机构，设计了切枪模块，如图 2.1.4、图 2.1.5，让哨兵可以以高射频持续输出。

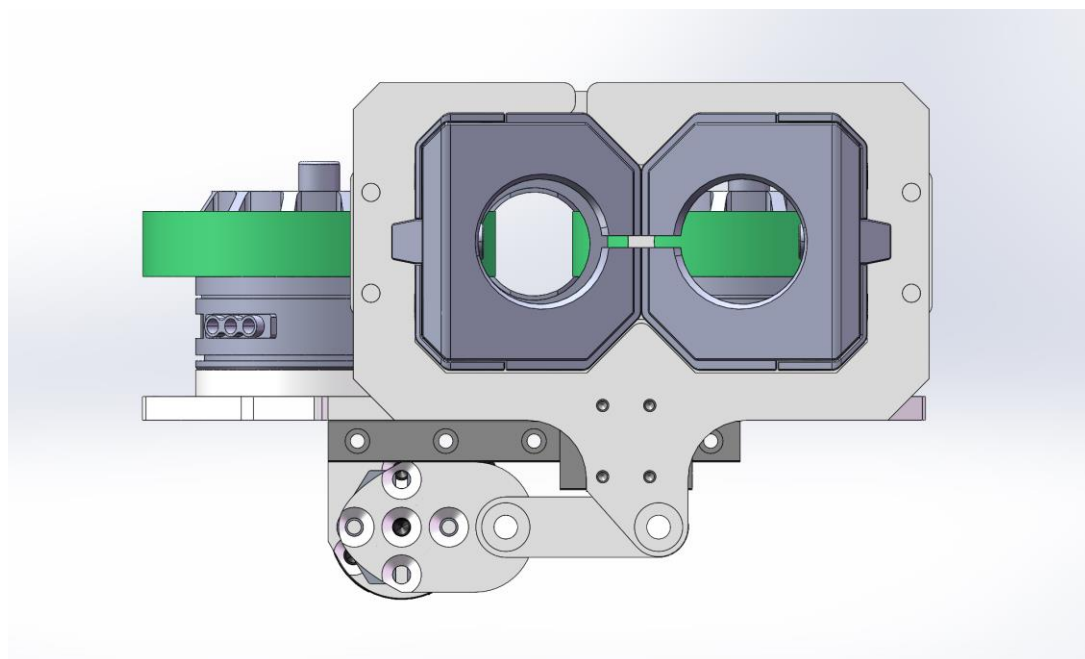


图 2.1.4 切枪机构正视图

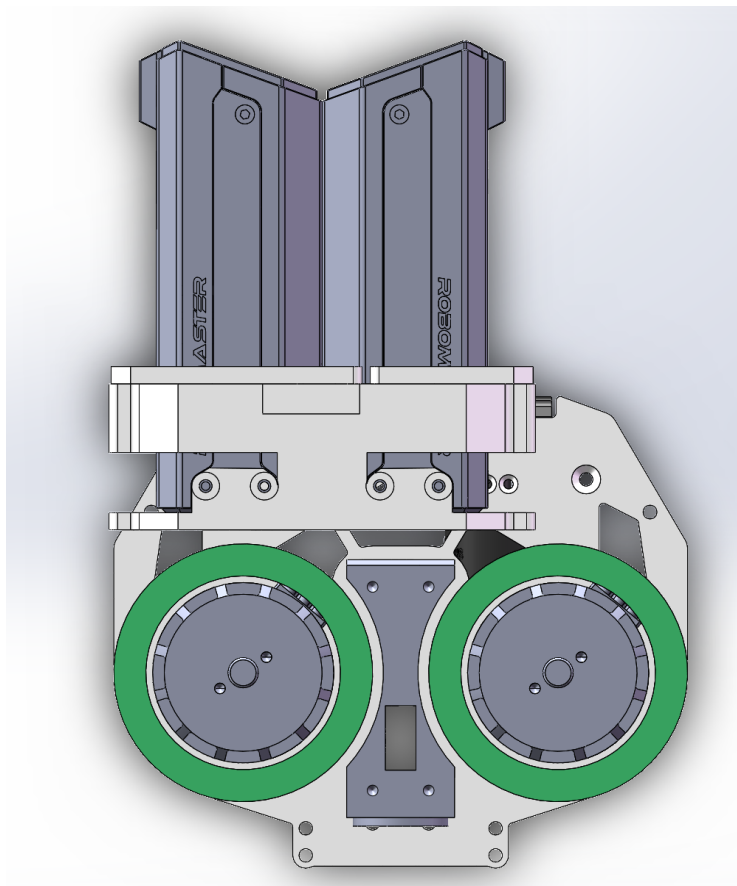


图 2.1.5 切枪机构俯视图

切枪机构以 2006 电机驱动,通过直线导轨约束测速机构在水平方向运动。因为 M2006 电机长期堵转容易发烫消磁,所以采用自锁连杆结构,在测速机构就位后,如图 2.1.4,水平方向的力无法使电机旋转,结构自锁,电机只需维持角度控制,负载较小。

由于在测试时发现右侧测速机构弹道呈两点分布,经检查发现是弹道略微偏左且在装配时由于过盈配合使右侧测速机构偏右,部分弹丸会打到右侧测速机构左侧。由于只是单根测速弹道异常,此问题由电控让摩擦轮差速较难解决,所以在机械侧通过将两段两杆同时增加 0.5mm,可以只让右侧发射机构向左偏置 1mm,有效避免了其干扰弹道。

2. Pitch 轴连杆

Pitch 轴连杆使用 GM6020 电机驱动,采用平行四边形连杆传动,如图 2.1.6,但是由于在侧板上的连接点过低,导致标红连杆过长。该连杆与限位圆片经常碰撞导致与电机连接的四颗螺丝经常松动产生虚位。虽然尝试上高强度螺纹胶,但是由于视觉调试时习惯把屏幕放在云台 pitch 轴上,导致云台失去配平,电机发热严重融化螺纹胶,故螺纹胶难以

紧固。之后使用生料带缠绕螺丝，缓解了这一问题，但是还是要定期检查加紧。所以以后连杆设计时应该注意尽量减短与电机连接的连杆长度。

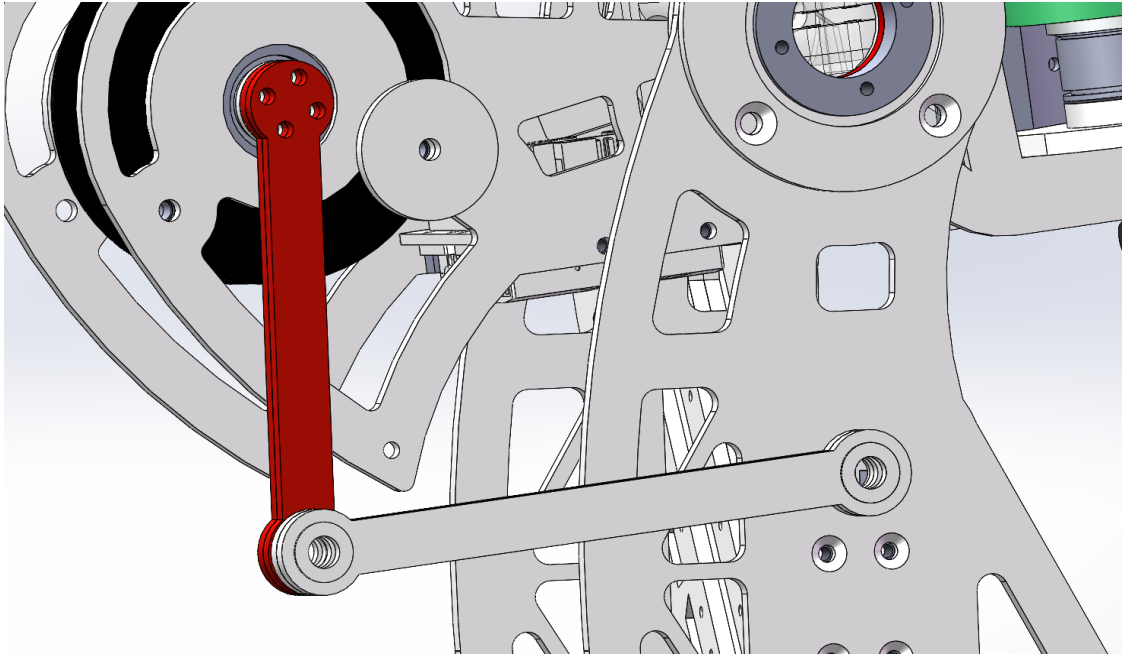


图 2.1.6 pitch 轴连杆

3. 供弹链路

主要链路采用鹅颈设计，进入枪管的则是一段直角链路，如图 2.1.7，通过减少直角弯以及增加轴承，减少对弹丸阻力。

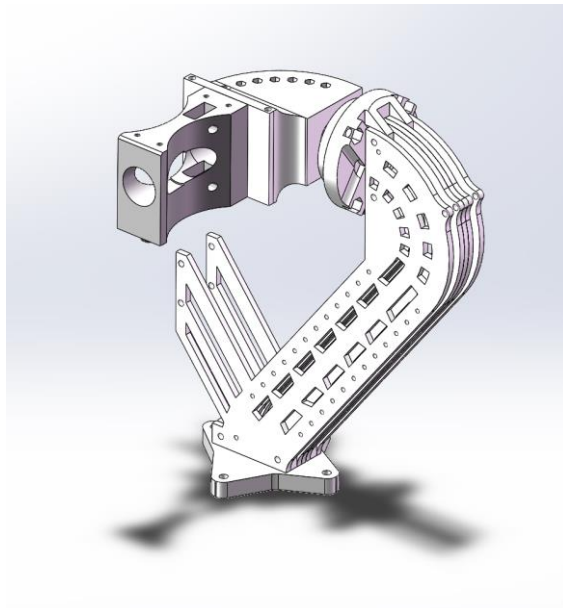


图 2.1.7 供弹链路

在测试过程中发现还是存在卡弹现象。由于是下供弹方式，弹丸在枪管处推力极小，所以通过增大枪管定位轴承间距直接减少对弹丸的阻力，后经过测试增大定位轴承间距对弹道散布无影响。同时扩大了枪管入口直径，使弹丸进入枪管经过一个阶梯，如图 2.1.8，避免生产误差和安装误差导致弹丸进入枪管时受到阻力。

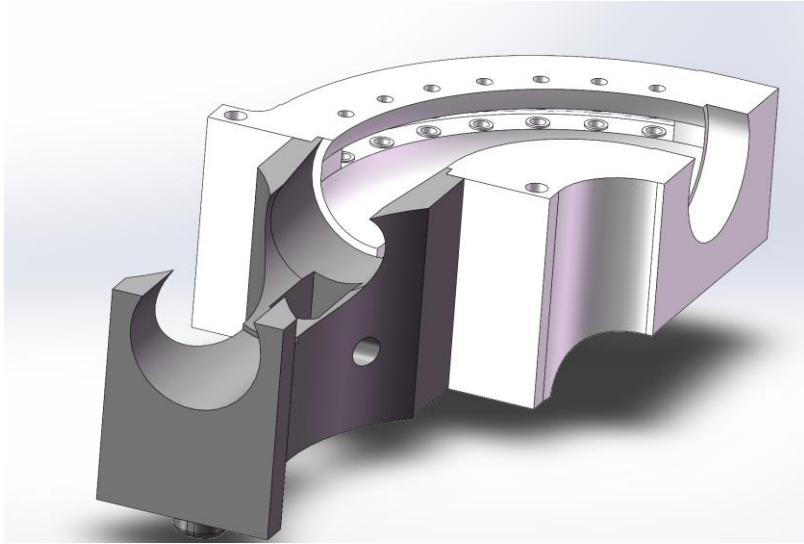


图 2.1.8 枪管供弹链路连接处剖视图

4. 雷达散热

由于视觉组使用雷达时反馈激光雷达长时间运行后会发热导致定位漂移，于是在激光雷达基座下加装了散热装置。选用黑鲨冰封散热背夹 2，如图 2.1.9，安装在图 2.1.10 中标红零件上方，与雷达间使用散热硅脂和铝板连接，保证散热均匀，防止冷凝水产生。经测试雷达可以长期使用温度保持在 20-30℃，满足定位需求。



图 2.1.9 黑鲨冰封散热背夹 2

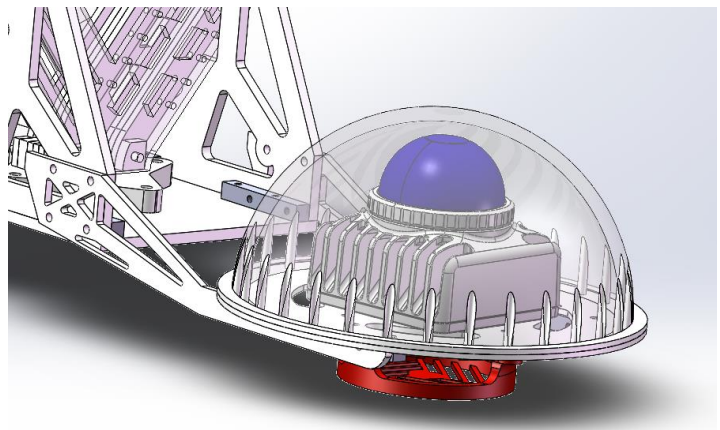


图 2.1.10 雷达散热安装位置

2.1.2 底盘模块

底盘模块下视图如下：

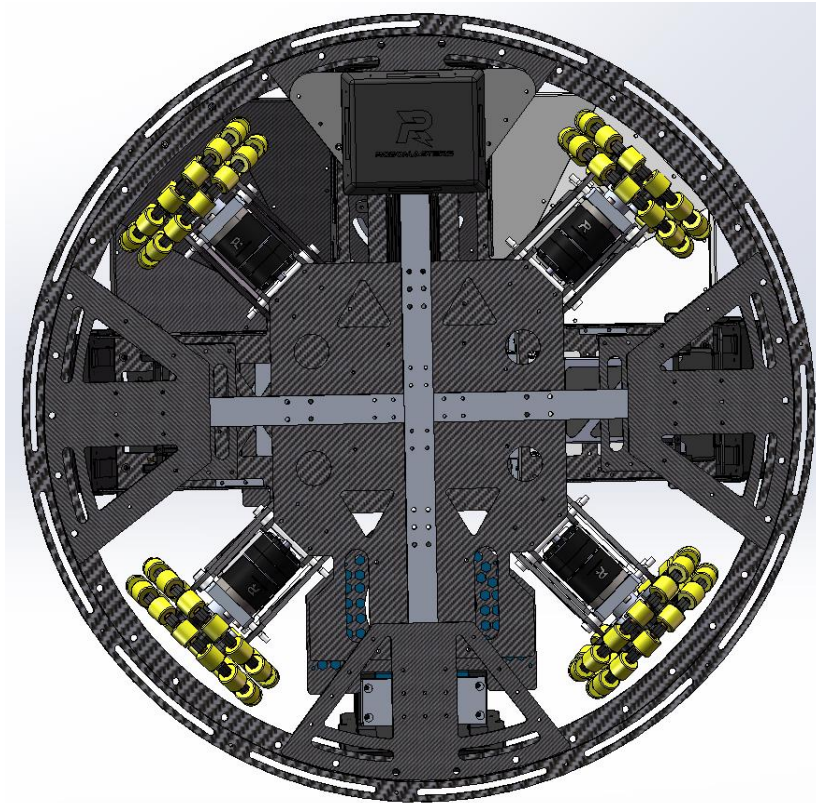


图 2.1.11 底盘下视图

1. 尺寸设计

根据 Robomaster2023 机甲大师高校系列赛机器人制作规范手册的规定，哨兵机器人的最大初始尺寸为 700*700*700mm，考虑到哨兵云台质量较大可能导致的重心偏高问题，最终确定哨兵在地面的正投影尺寸为 685*685mm。

2. 保护框设计

为了防止哨兵被敌方干扰，设计整体呈圆形，将保护框外层与内层通过轴承连接，在外层被卡住时内部仍可进行小陀螺旋转

该部分采用内层与外层的结构，如图 2.1.12，内层为上下两层中间夹深沟球轴承的方式，通过垫片实现轴承灵活转动。外层理论上为三层板，上下两层内径较小，可以夹住轴承，中间层内径较大，与内层的轴承实现过盈配合。经过实践检验该方案可以很好的实现目的，外层保护框可以顺滑的转动。但由于添加了较多深沟球轴承，整体重量较大，之后考虑会进行

有限元分析，进行镂空减重，并适量减少深沟球轴承的数量。

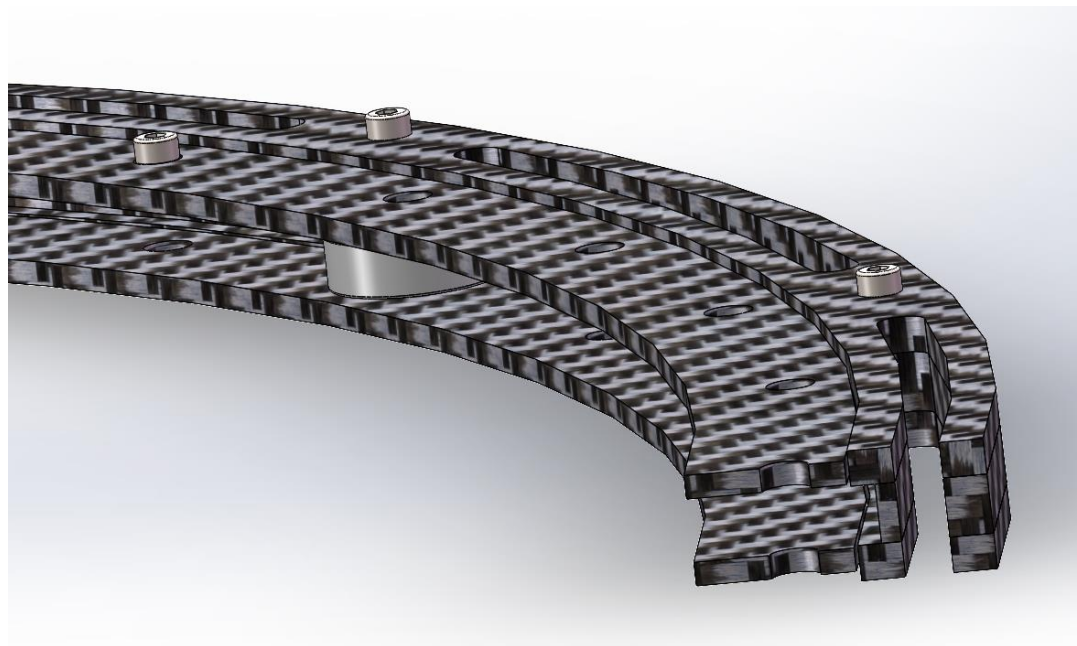


图 2.1.12 保护框剖视图

3. 悬挂设计

为了实现较快速小陀螺及全向移动，轮组采用一般的平行四边形独立悬挂，自制全向轮，结构较为简单，维护方便，为了防止外八，在联轴器与铝件间加入滚针轴承，经过测试寿命较长，整赛季基本无需更换。需要注意的是该联轴器压紧部分，最初使用碳板，在平地测试较为良好，但经过上坡及起伏路段等负载大的场景时，碳板会很快磨损。之后尝试更换为钢制，使用寿命大幅增强。由于自适应的缺失，哨兵上坡在部分姿态会有问题，需要旋转上坡。

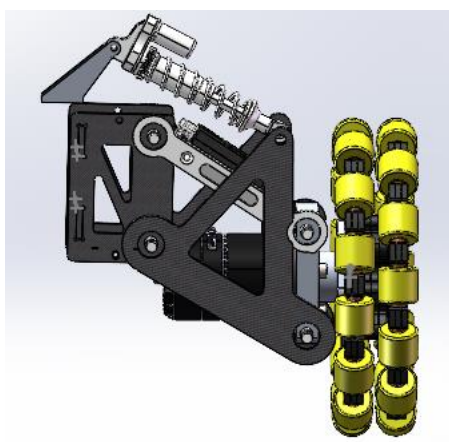


图 2.1.13 悬挂侧视图

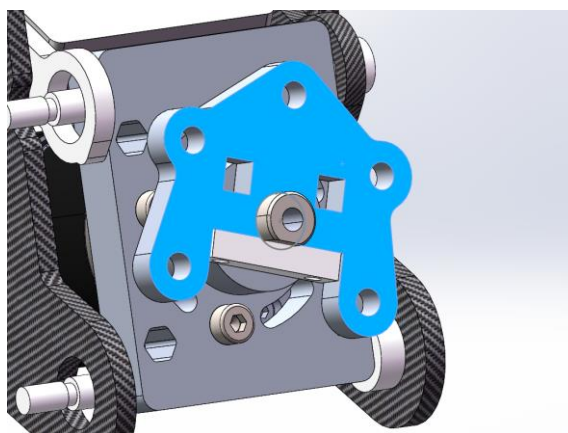


图 2.1.14 轮组联轴器

4. 主体框架设计

主体结构采用三层板材，通过铝管搭建框架支撑，如图 2.1.15，中间可以容纳较大的滑环，同时为下供弹链路提供空间，同时内部两层结构空间较大，为 NUC 安装及电控布线留下充足空间。低侧装甲板安装在铝管上，最初使用 50*40*1 铝管，在装甲板经受较多打击后铝管会变形，后更换 50*40*2 铝管；高侧装甲板通过板材垫高安装于第二层碳板，较为牢固。

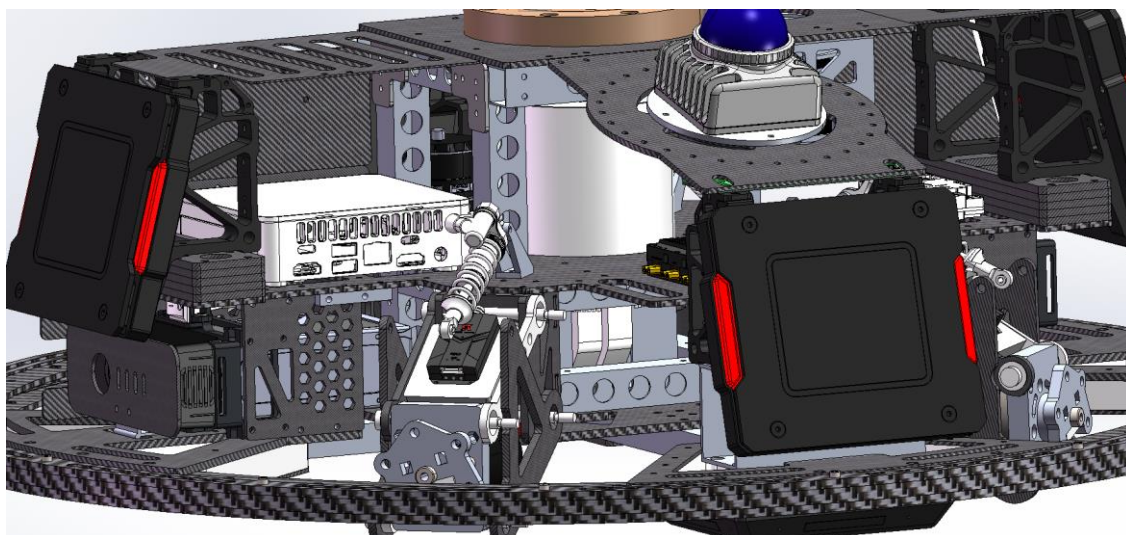


图 2.1.15 主体框架

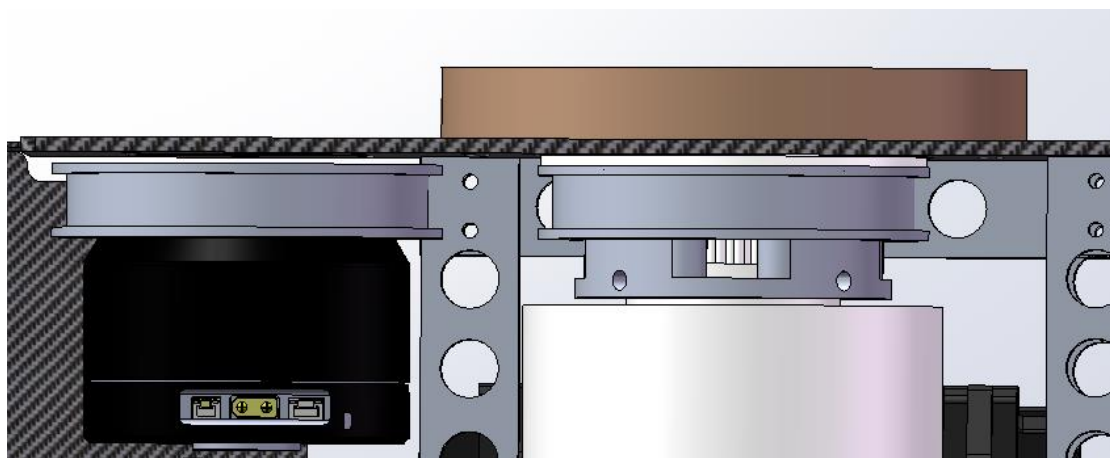


图 2.1.16 Yaw 轴传动部分

5. Yaw 轴设计

为实现下供弹，Yaw 轴传动部分采用同步带传动，如图 2.1.16（其中同步带未画出），通过两个同步轮连接 yaw 轴电机与滑环，上方通过交叉滚子轴承连接云台。最初使用 GM6020 电机，但由于云台转动惯量较大，发现 GM6020 扭矩无法满足要求，后更换达妙 4310 关节电

机，问题得以解决。同时达妙 4310 存在问题为定子固定螺丝为 3 颗 M2*6 沉头螺丝（图 2.1.17 中红框），强度较低，在经过长时间负载后出现松动，使用高强度螺纹胶仍无法解决该问题，后与商家沟通确认为电机设计缺陷。

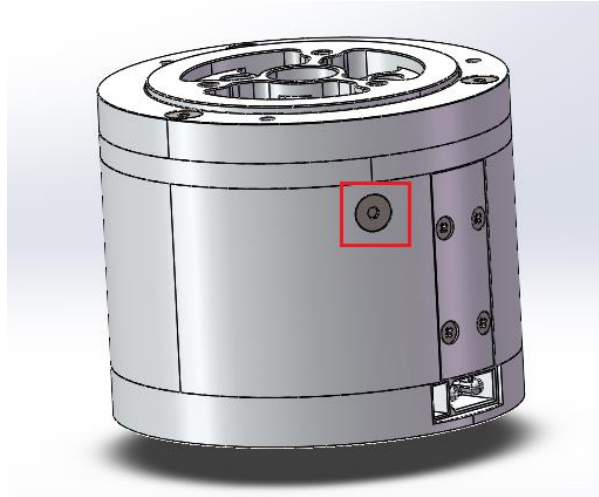


图 2.1.17 达妙 4310 电机

2.2 硬件设计

整机硬件方案框图如下：

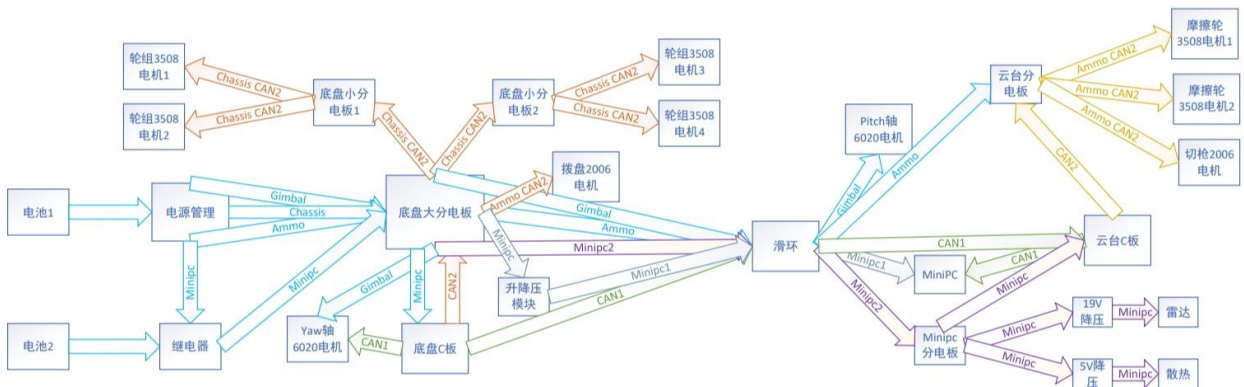


图 2.2.1 整体硬件方案图

2.3 软件设计

2.3.1 系统架构

底盘与云台嵌入式算法均通过 FreeRTOS 分任务进行。

2.3.2 运行流程

1. 云台

a) CalculateThread

用于控制 Yaw 轴与 Pitch 轴电机，从而实现云台姿态的自由控制。

其控制信号的来源分别是遥控器，视觉 minipc，以及裁判系统给出的小地图按键信息。遥控器的控制优先级最高，用于控制无力、调试以及比赛三种模式的切换。而视觉 minipc 的控制主要在于自瞄指令的给出。在自瞄指令显示为未识别到目标时，云台自主切换为 Yaw 轴旋转，Pitch 轴正弦抖动的扫描模式。裁判系统给出的小地图按键信息则控制扫描模式下，Yaw 扫描的角度以及 Pitch 轴的最大俯仰角。

b) CanSendTask

用于向底盘定频发送通信数据，频率为 500Hz。

主要通信数据为，接收到的遥控器数据，云台 Yaw 轴陀螺仪角速度数据，发射机构切枪状态信息。

c) ShootThread

用于控制摩擦轮的转动，以及根据热量情况来切换枪管。

2. 底盘

a) ChassisTask

用于控制底盘四个轮组电机，从而实现底盘的多种运动模式。

其控制信号的来源分别是遥控器，视觉 minipc。遥控器的控制优先级最高，用于控制无力、调试以及比赛三种模式的切换。而视觉 minipc 的控制主要在于控制底盘的运动速度，与是否进行小陀螺，从而实现自主运动。

b) **CanSendTask**

用于给云台定频发送通信数据，频率为 1kHz。主要通信数据为，是否接收到比赛开始指令。从而使整车运行模式从保底模式切换为比赛模式。

c) **RefereeTask**

用于接收裁判系统数据，经过解算后，按需求发给 miniPC 以及云台。

d) **ShootTask**

用于控制拨盘的转动，从而实现子弹的发射。

3. 通用

AttitudeThread

用于读取 C 板中自带的 bmi088 的测量值，并且估计此时的 C 板姿态，从而判断云台与底盘相对于绝对坐标系下的姿态，以及云台与底盘之间的相对姿态。

2.3.3 重点功能

1. 随机小陀螺

由于视觉的反小陀螺的出现，单纯的定速小陀螺也无法较好的规避子弹。为此，对小陀螺的情况进行一定的随机化，从而降低被敌方击中装甲板的概率。

为实现随机小陀螺的效果，利用了随机数生成函数，做出旋转速度的随机话，旋转时间的随机化，以及反转时间的随机化，从而在面对有反小陀螺的自瞄系统时，有更长的存活时间。

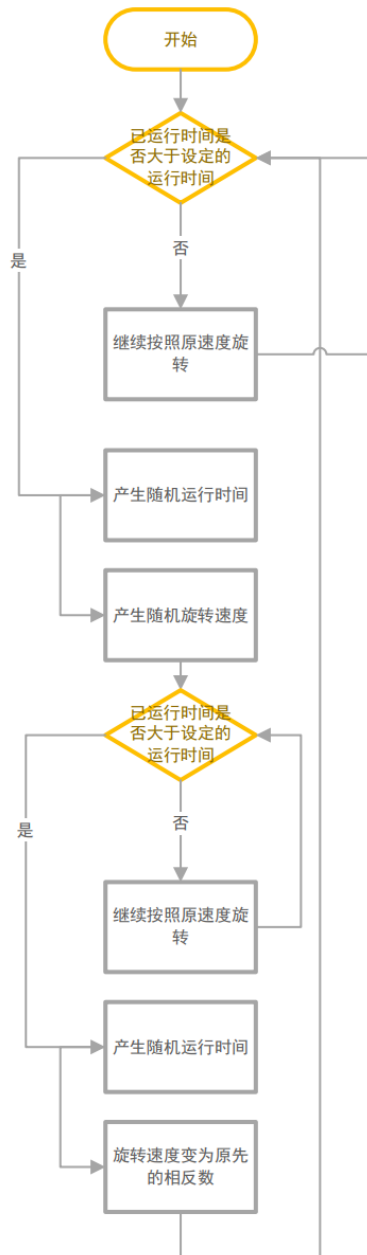


图 2.3.1 随机小陀螺实现框图

2. 刹车

由于车体质量较大，使用常规的轮子卡死的刹车方式依然会由于轮子的打滑，造成车体无法立刻停下。这对于控制精度较高的哨兵来说存在较大的问题，如果车体移动，必将会导致其路径规划出错，从而使车体在一个小范围内不停运动，无法停下。

为此,通过判断前一刻与这一时刻的速度的变化趋势,来判断是否应该进入刹车模式,再利用反向的速度,使得车体能够较好的停在某处。若不需要刹车,则用缓启动的方式来对控制速度的突变进行处理。

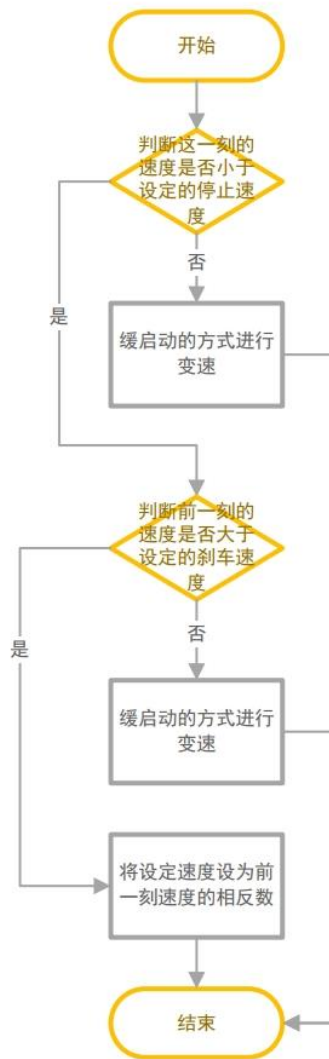


图 2.3.2 刹车实现框图

3. 云台旋转的情况下的底盘自主移动

为使哨兵在导航情况下依然能够识别敌方装甲板并且攻击,需要将底盘的控制与云台进行分离。但在分离后,由于雷达安装于云台上,底盘无法得到雷达数据与绝对坐标系的关系,因此无法较好的执行控制指令。利用雷达解算的陀螺仪数据后,仍然有较大的偏差,原因在于雷达陀螺仪解算频率较低,在云台旋转的情况下有着较大的延时,导致了较大的角度偏差。

而云台 C 板解算出的陀螺仪数据有较大的零漂问题，也无法较为精确地控制底盘运动。

通过定期停止云台转动，结合雷达陀螺仪数据与云台 C 板陀螺仪数据进行多次的零点校准，再根据云台 C 板陀螺仪数据做坐标系转化，从而较好地控制底盘的运动。同时，需要对云台 C 板陀螺仪数据进行滤波，从而更好控制。

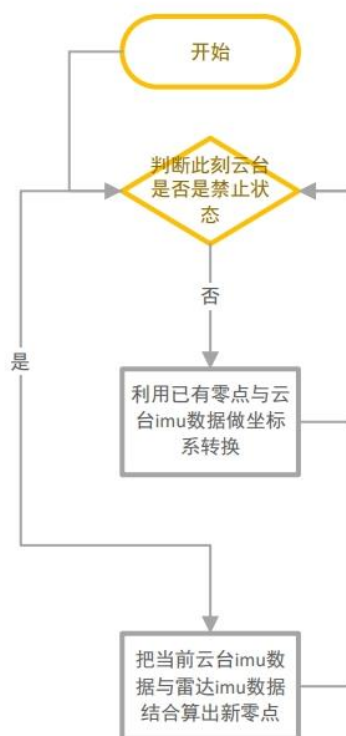


图 2.3.3 1. 云台旋转的情况下的底盘自主移动实现框图

4. 上坡检定

由于上坡时所需要的功率较大，而 minipc 给出的控制速度是固定不变的。

为此，通过底盘 C 板陀螺仪数据，来判断此时是否在坡上，并等比放大此时的运动速度，从而实现底盘功率的充分利用。

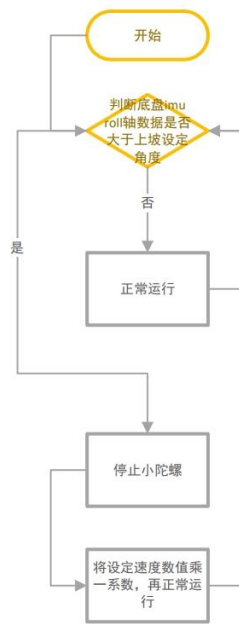


图 2.3.4 上坡检定实现框图

5. 自动校准 Yaw 轴初始位置

由于每次机械拆装云台都会导致云台系与底盘系的角度产生偏差，需要重新更改数据。为此，通过在校准模式下将此刻的角度数据写入 Flash 的方式，实现更加简易的初始化操作。

2.4 算法设计

哨兵的算法部分分为自瞄，定位导航和决策这三部分，其中自瞄控制云台，导航控制底盘的速度，决策负责接收裁判系统信息并控制底盘的模式。

2.4.1 自瞄算法设计

在这赛季的自瞄开发中，我们选择了基于 ROS2 进行开发。在之前赛季的开发中，我们发现，如果没有一套优良的框架，自瞄算法在整个赛季的优化过程中容易非常变得混乱，尤其是不同的人负责不同功能的实现的时候，很容易出现“堆屎山”的现象。因此在搭建自瞄框架的时候，我们目标就是最大程度上增强可拓展性，利用 C++ 的基类与继承的思想，我们将算法中大部分的对外接口一致封装，并且将算法部分和 ROS2 的节点进行完全解耦，只在节点中调用预先封装好的公共接口，这样在一定程度上规范了后续开发的流程，只需要继承基类的外部接口，再在内部进行开发即可。

1. 目标检测

在本赛季的研发过程中，我们顺延了上赛季的研发思路，将研发路径一分为二，一部分去做传统识别的优化，一部分尝试基于 YOLO 的装甲板识别。最终经过实际的测试发现，基于 YOLO 的装甲板识别算法虽然识别鲁棒性极强，能够更好地适应各种复杂的光照环境，但是由于角点的检测没有平面约束，我们当时也没有做到对角点的有效优化，导致识别的精度较差，除此之外 YOLO 算法相对更高的耗时也影响了算法的性能。相反的是，经过优化以后的传统识别算法，在鲁棒性上虽不及 YOLO，但相较于去年依然有了很大的提升，并且因为本赛季我们在识别后的模型建立上改变了思路，对丢帧的容忍性大大提高，意味着我们的精度的要求要大过于识别率，因此最终上场我们采用了识别率相对低但精度更高的传统识别算法。下面对基于 YOLO 的装甲板识别方案和数字识别和分类方案进行介绍。

基于 YOLO 的装甲板识别方案：

在目标检测任务中使用卷积神经网络相较于传统机器视觉而言，具有鲁棒性强的特点，而且在大量的数据集训练后，准确率也较高。

但大模型的神经网络计算耗时长，使用在实时目标检测领域最新和最优秀的 YOLO（you only look once）能实现在较快的推理速度下，同时保证检测的精度。我们将赛季初使用的 YOLOv5 升级为最新研究成果的 YOLOv7，进一步提升检测的精度和速度。

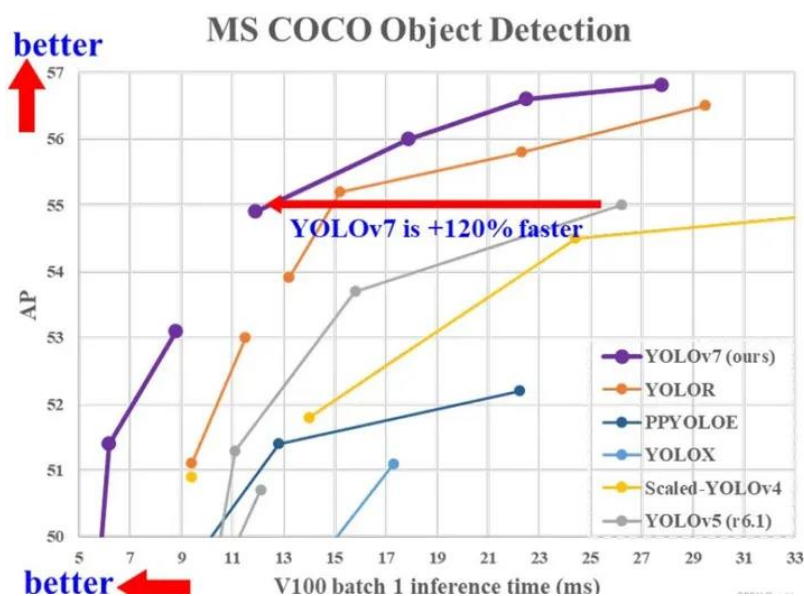


图 2.4.1 YOLOv7 性能对比图

但因为实际机器人设备性能限制与检测需求的不同，我们急需的就是网络推理速度一定要快，否则就失去了上场的可能，所以原版的 yolov7 网络模型并不能满足我们的需求，我们以 yolov7-lite 为基础，修改网络结构。

1) ShuffleNetV2 是一款轻量级网络，对移动设备推理进行了特殊优化，并提出了指导轻量级网络设计的四条准则。遵照这几条准则，我们将 backbone 部分换成了 ShuffleBlock，并进行了一些其他微调，以加快推理速度

```
# custom backbone

backbone:

  # [from, number, module, args]

  [ [-1, 1, StemBlock, [16, 3, 2] ], # 0-P2/4

  [-1, 1, Shuffle_Block, [48, 2]], # 1-P3/8

  [-1, 2, Shuffle_Block, [48, 1]], # 2

  [-1, 1, Shuffle_Block, [96, 2]], # 3-P4/16

  [-1, 5, Shuffle_Block, [96, 1]], # 4

  [-1, 1, Shuffle_Block, [192, 2]], # 5-P5/32

  [-1, 2, Shuffle_Block, [192, 1]], # 6

  [-1, 1, SPPF, [192, 5]],]
```

2) 深度可分离卷积 (Depthwise separable convolution) 相较于常规的卷积操作，其参数数量和运算成本比较低，我们将 yolov7 的检测头借鉴于更小的 v5 的检测头，并换为深度可分离卷积，提高检测头部分的推理速度。

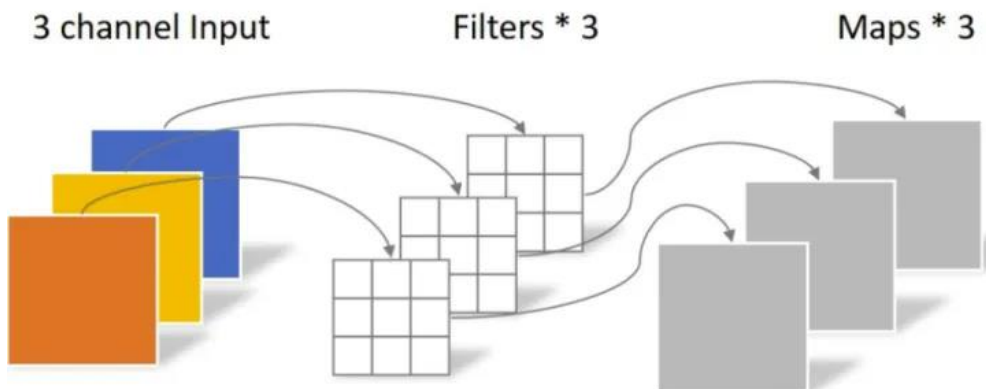


图 2.4.2 深度可分离卷积示意图

- 3) 解耦数字与类别，降低分类输出类别数量。我们将数据集的类别改为带小数的数字（小数部分代表颜色）实现类别和颜色的读入网络，将原本的关键点的置信度改为计算颜色的 BCEloss，从而实现将分类数由“类别×颜色”变为“类别+颜色”。有效减少了分类维度，在数据集总量不变的前提下，增大了各类的样本数，降低网络的学习难度，提升推理的速度。



图 2.4.3 神经网络识别效果图

在数据集的制作上，我们对已有数据集进行了数据增强操作，包括替换复杂背景，添加椒盐噪声，适当修改亮度，饱和度，图片角度等。这样可以快速得到大量数据集的同时，增加数据集的多样性，并平衡各类别的数量比例，有效缓解模型训练的过拟合程度。



图 2.4.4 数据增强示例图

OpenVINO 是英特尔基于自身现有的硬件平台开发的一种可以加快高性能计算机视觉和深度学习视觉应用开发速度工具套件，支持各种英特尔平台的硬件加速器上进行深度学习，并且允许直接异构执行。当我们应用 Pytorch 训练好模型之后，就面临了模型部署上线机器人车 NUC 的问题。OpenVINO 是解决这一问题的优秀工具。同时使用 OpenVINO 进行模型优化和相关推理引擎的优化，可以进一步提高神经网络推理的速度。

除此之外，我们还尝试了将 YOLO 和传统识别相结合的方案，大致思路如下：将整个识别部分分为两个阶段，第一阶段我们利用 YOLO 算法找出的装甲板角点和给出装甲板类型，通过比例放大，框出一个包含装甲板的 roi，因为我们只需要 roi 和数字，因此对角点的精度要求不高，因此我们可以将整个网络裁切得很小，以解决 YOLO 算法耗时过长的的问题。第二阶段，我们在装甲板 roi 区域，再做一次利用传统算法的识别，得到装甲板的角点，如果此时传统方案找不到角点，则我们将第一阶段找到的角点作为最终输出。

数字识别和分类方案：

本赛季视觉装甲板目标检测算法在传统视觉识别和深度学习 YOLOv7 识别的分别测试之下，传统视觉算法以速度较快和较好参数下的准确识别而采用于最终上场使用。

相较于神经网络的直接识别分类出装甲板颜色和数字，传统视觉识别则需先识别出装甲板两侧灯条，再通过两灯条的长度，角度差和倾斜角等匹配条件进行匹配出正确的装甲板。之后，将匹配出的装甲板部分图像裁剪出 ROI 进行数字识别来给装甲板提取出数字，用于后续进行装甲板整车建模和判定击打策略。

在使用 SVM (Support Vector Machines) 机器学习算法进行数字分类时，其对于赛场上复杂灯光和不同曝光下的装甲板鲁棒性很差，对曝光要求极高，且对于“3”和“5”之类有可能分类错误。

由于赛季初装甲板的灯条匹配鲁棒性较高，于是为了解决数字识别对于曝光的苛刻依赖，引入神经网络进行数字识别。为保证识别准确率，首先考虑的即是 ResNet (残差神经网络)。同时为了兼顾传统视觉算法速度快的优点，并且匹配灯条后的装甲板背景较为单一，数字特征明显，我们选择自己手搓两个 ResBlock，再进行少量的卷积操作，实现轻量

化，减少大规模运算的同时，具备很高的准确性。

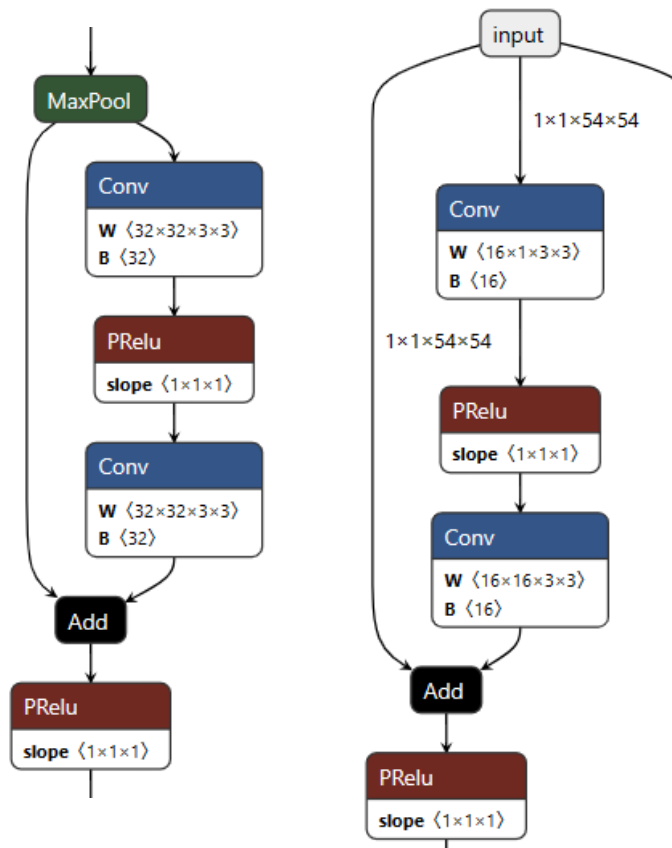


图 2.4.5 ResBlock

在数据集的制作上，我们使用以实际识别下的装甲板采集数据集，并使用适当改变亮度，旋转一定角度等进行数据增强，减小过拟合。此外，传统灯条识别和匹配中，偶尔会出现错误匹配灯条的情况，根据实际识别装甲板的所有数据集图像中筛选出错误识别的图片，并制作成为 error 类别数据集，能够实现在数字识别中，进一步纠正误识别的装甲板，大大提高了识别的准确性。



图 2.4.6 装甲板数据集示例图



图 2.4.7 装甲板数据集示例图

综上，在配合使用卷积神经网络的数字识别的传统装甲板识别具有速度快，正确率高，鲁棒性强的特点，将传统算法对于曝光的苛刻要求大大降低，能够适应场上的曝光变化，并且对于场地灯效可能造成的误识别在数字识别阶段再次排除，避免了在赛场上错误打弹。

2. 整车状态建立

在本赛季，我们参考了华南师范大学陈君的开源方案，采用了基于 ekf 的整车建模算法，关于参考的部分，因为是开源方案，因此不加赘述。主要讲的是，我们后期在此开源方案基础上，对建模的方案做了一定改进：因为比赛场上往往是电光火石，刀光剑影，两辆车照面的瞬间就要交出胜负，因此模型的从建立到收敛的速度一定要很快，但这也意味着，模型很容易因为外部的精度误差而错误，并且可能会产生很大的抖动。因此我们想到赛场上的车绝大多数都存在相当多的具有统一性的限制条件，比如两个半径的差值一般不会大于 0.15m，比如相邻两个装甲板的高度差不会大于 0.055m，利用这些条件对模型加以限制可以

更好地提高算法的鲁棒性。

3. 自动开火设计

在针对高速旋转的物体，或者英雄打旋转前哨站的时候，我们的云台是不完全随动的，如果这个时候还是把开火权交给操作手，就会出现浪费子弹的现象，因此我们本赛季首次在全兵种推行了自动开火机制，其基本原理如下：

假设枪口射出的子弹是一条弧线，弧线所在的平面应该大致垂直于枪口所在的平面，我们把弧线所在的平面称为“子弹平面”，因为我们发射出的所有子弹都在这个平面上。而我们的相机和枪管之间是没有 roll 角度上的差异的，因此我们可以假设子弹平面和相机的成像平面有一条交线，如果在理想条件下（枪管中心和相机光心完全对齐），我们如果采用的图像大小是 1280x720，那么这条交线就是过 (640, 360)，垂直于图像上下边界的一条直线。在相机图像中，我们所有发射的子弹理论上都在这条直线上，我们把这条线称为“开火线”。

有了事先测量出的开火线之后，我们将绝对坐标系下的预测位置，转回相机坐标系，再转到像素坐标系，此时我们的预测点已经成为二维像素平面上的一个点，我们通过这个点和“开火线”的距离是否小于阈值来判断此时是否应该开火。

同时，开火阈值并不是一个不变的量，考虑到打击效率的问题，我们把开火阈值设成了一个与目标物体转速反相关的变量，当对方物体处于低速旋转或者没有旋转，只是处在静止和平移的时候，阈值是一个很大的量，这相当于间接取消了自动开火，将开火权交回给操作手，只有当目标车辆处于高速旋转的时候，才会开始有节制地进行开火。英雄打旋转前哨站的时候，参数与面对高速旋转物体时相同。

新赛季哨兵的自瞄没有操作手实时操控需要完全自主瞄准并完成开火击打，这要求自瞄代码有更强的鲁棒性及自主性，因此我们需要在战队已有自瞄代码的基础上做出调整以适配哨兵。同时哨兵的装弹量血量等参数显著不同于步兵英雄等兵种，哨兵的开火逻辑也需要新的调整。

考虑到 IRobot 战队步兵英雄等兵种采用 nuc9、nuc11 等上位机使用 1080*720 的图像跑自瞄代码能跑到 180 帧，而哨兵使用了魔方 i9 性能更加强健，因此考虑更大的 1920*1200 图像以获得更广的视野范围更容易发现敌方装甲板。经过验证，在同时跑导航代码，自瞄代码，

决策代码的情况下，哨兵自瞄同样能跑到 180 帧。同时，魔方 i9 共有大小核共 20 个，小核适合与通信决策等性能开销小的程序绑定，大核适合与图像识别、自瞄解算、导航规划等性能开销大的程序绑定，我们通过绑核可以再提高一部分自瞄的帧率。

由于哨兵携带了 700 发子弹远超步兵，在实战中我们也发现哨兵 700 发子弹时难以打完的，因此区别于步兵的精准打击，哨兵需要“泼水”的感觉。通过拉高到最高射频以及调整自瞄内的火控参数，我们使哨兵的自瞄更具爆发性和侵略性。

2.4.2 定位导航算法设计

1. 定位算法

前期尝试 VIO 算法进行定位，使用 T265 内置算法和 ORB-SLAM3 做视觉里程计，AMCL（自适应蒙特卡洛算法）。但是由于光照环境和动态物体的影响，视觉里程计的精度和鲁棒性比较差。在长时间的测试和调试中，多次出现定位飘掉和 AMCL 不收敛的情况。

随着 Livox-mid360 的发行和考虑到 LIO 算法的精度和稳定性较高，我们使用 mid360 对几种精度较高的 VIO 算法进行长时间和长距离测试和算法验证，如下：

表 2.4.1 LIO 算法对比

	回环误差 x (m)	回环误差 y (m)	回环误差 z (m)
Lio-Livox	0.15	0.256	0.18
Fast-lio	0.02	0.066	0.108
Point-lio	0.008	0.010	0.05
Faster-lio	0.006	0.008	0.06

通过以上测试数据，我们采用 Point-lio 和 Faster-lio 两种 LIO 算法作为我们定位的激光里程计算法，进行导航的调试。

2. 导航算法

导航算法我们参考 CMU-Exploration 局部探索算法，结合 Far-Planner 高级规划器的思想，最终采用 A*全局规划+cmu-exploration 局部规划算法。所有 waypoint 都是使用局部规划器规划轨迹以到达这个点，所以这些 waypoint 离机器人都比较近，而且是在一个可到达的区域。实际操作过程中需要机器人到达一个比较远的地方，这个时候就需要一个上层规划器（全局规划）将 waypoint 拆分成很多小的目标点，一步一步引导机器人到达目的地。

A*全局规划算法：

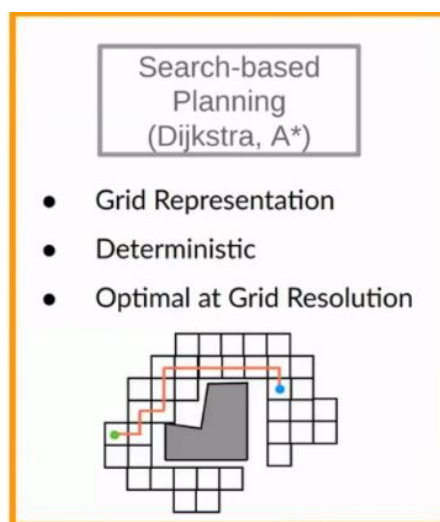


图 2.4.8 A*算法图解

A* (A-Star) 算法是一种静态路网中求解最短路径最有效的直接搜索方法，也是解决许多搜索问题的有效算法。A* 算法是贪心算法和 Dijkstra 算法的结合，贪心算法 ($f(n)=h(n)$) 在规划路径时只寻找到目标节点最近的节点来更新路径，Dijkstra 算法 ($f(n)=g(n)$) 考虑的是当前节点与其周围节点之间的距离，遍历完所有节点才可以找到最优路径。

由 LIO 算法保存的 PCD 点云图进行下采样处理，进行 Z 轴 2D 压缩成 2D 占用栅格地图。我们在离线建好的栅格地图上进行膨胀，由膨胀层和障碍层构成全局 costmap，在全局 costmap 中运用 A* 算法进行路径搜索，最终规划出一条全局路径。

如下图，探索环境为机器人实验室，绿色的线即为 A* 算法规划出的全局路径。

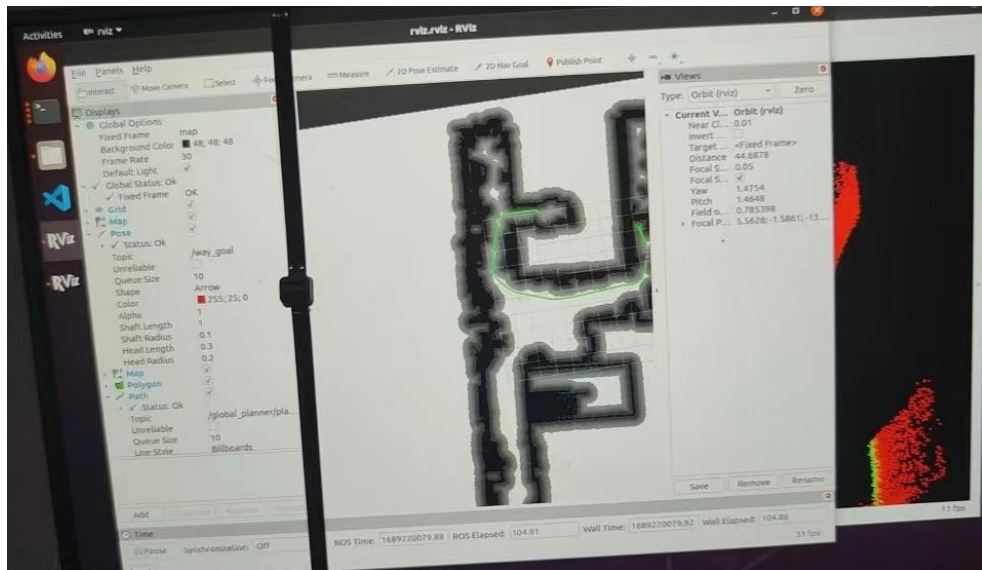


图 2.4.9 全局路径可视化

CMU-Exploration 局部规划算法:

地形分析模块检查车辆周围局部地形的可穿越性。该模块构建一个代价地图，其中图上的每个点都与通过代价相关联。代价由周围地形的平整度决定。我们使用体素网格来表示环境，并分析相邻体素中数据点的分布来估计地面高度。如果这些点离地面更远，则它们与更高的通过成本相关联。图 2 给出了一个以车辆为中心覆盖了 40m*40m 地区的地形图的例子。图中这些绿色的特征点为可通过区域，红色的特征点为不可通过区域。此外，地形分析模块可以处理负面障碍，这些障碍通常会导致地形图上没有数据点的空旷区域。负面障碍应该为与地面在同一平面内但是没有点云信息，会将他们判定为类似于坑的障碍，当启用负障碍物处理时，模块将这些区域视为不可通行。由于安装在云台前方的 mid360 扫描范围有一定的遮挡，单靠激光雷达的点云无法对云台后方的探索环境进行感知，所以我们在云台后方加了一个 realsense-D435 深度相机。将深度相机输出的点云进行下采样和体素滤波处理后，与激光雷达的点云进行拼接，得到一个全向环境的点云。

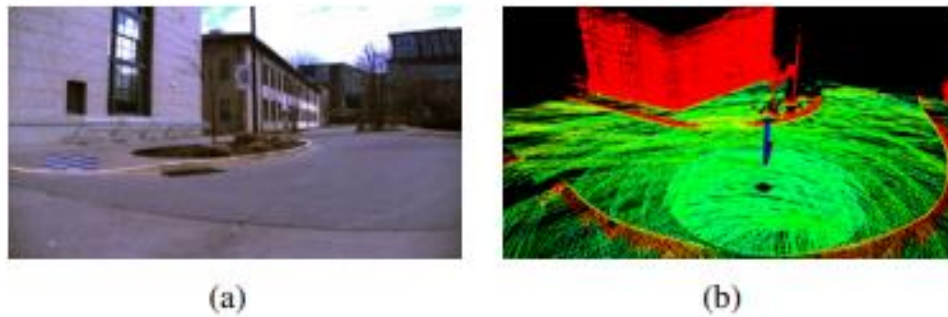


Fig. 5. Example terrain map. The image in (a) is taken from the location of the coordinate frame in (b). The green points in (b) are traversable and the red points are non-traversable.

图 2.4.10 地形分析可视化

局部路径规划器(local planner)能够保证机器人安全地到达全局规划器所发出的航路点。它能够计算并跟随无碰撞点路径引导车辆到达目标点。该模块预先计算运动基元库,并将运动基元与车辆附近的 3D 位置相关联。运动基元是指车辆在有限时间内,在自身约束条件下能够行驶的距离。

运动基元被建模为蒙特卡洛样本,并且是分组组织,实时的。当一个地方被障碍物遮挡,模块可以确定在毫秒内与障碍物碰撞的运动基元。然后,模块选择朝向航路点具有最大可能性的运动基元组。在图 3 中,这个红色的路径代表了无碰撞物运动基元。对于地面车辆,车辆的可通行性由地形特征决定。本地路径规划器(local planner)从地形分析模块(terrain analysis)接收这个地势地图。模块 LocalPlanner 与 TerrainAnalysis 通过话题通信建立通信,LocalPlanner 订阅来自 TerrainAnalysis 的话题)。该模块还有一个接口,用于接收额外的距离数据,以避免碰撞,作为扩展选项。

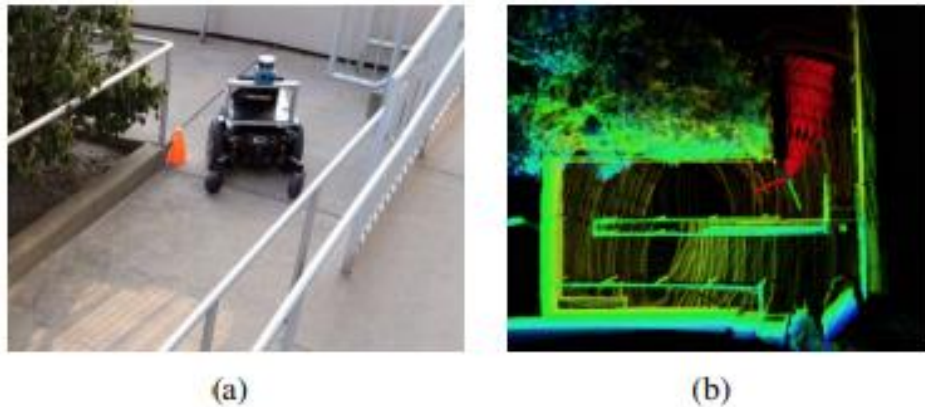


Fig. 4. Example motion primitives. The vehicle in (a) and the coordinate frame in (b) share the same location. The red paths in (b) indicate collision-free motion primitives.

图 2.4.11 机器人探索过程

运动基元库是 Matlab 用采样的离散点做前向模拟三次样条生成的路径点。用采样的离散点做前向模拟，基于机器人当前状态，预测一段范围内可能出现的情况。生成沿着不同路径到达传感器范围边界的路径集合。这些示例路径是基于车辆运动约束生成的，每条路径生成三次样条曲线，一组路径可以看作是从起始状态到传感器范围边界的可行路径。在导航过程中，感知传感器（激光雷达）检测到阻塞某些路径的障碍物，便将该部分路径删除，仅保留可通行的路径子集。

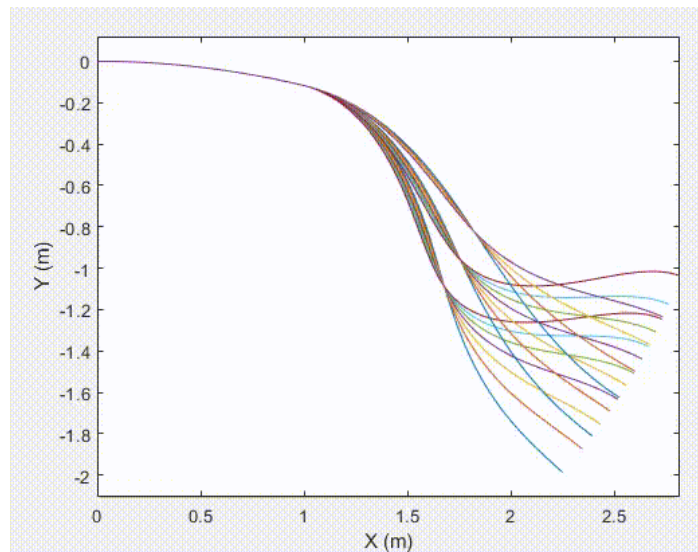


图 2.4.12 采样路径的生成

接下来机器人对局部规划出的路径进行路径跟随。路径跟随的主要思想为 PID 控制思想，通过合理地加减速使得机器人尽可能地贴近规划出的路径。因为在导航的过程中我们需要云台也保持一定的速度旋转，如果在云台系下计算速度会因为延时等问题出现实际走出的轨迹和计算路径相差较大的情况，所以我们设定路径跟随所在的坐标系为 odom 的绝对系，保证云台的旋转对路径规划的计算不产生影响。路径跟随模块计算出在 odom 绝对系下的 v_x, v_y ，将速度指令通过 can 通信发给底盘进行控制。

2.4.3 决策算法设计

1. 决策方案比较

分析可得哨兵获取信息的渠道有以下几种：用于自瞄的大恒相机，用于导航定位的深度相机或激光雷达，裁判系统。为了使哨兵能根据各种信息自主计算出导航的目的地，哨兵的决策代码需要在 nuc 内运行而不是主控板。

在赛季初期导航方案尚未确定，控制方式尚未确定的情况下，我们并不知道能不能边小陀螺边导航也不知道能不能边导航边自瞄开火。同时我们不希望只通过几个简单的 `if else` 语句来控制哨兵在场上的行为，因为那样可能会显得笨重且拓展性差功能难以升级。

因此我们决定引入行为树（也称为决策树）的思想，每种行为都是整个行为树节点下的“叶子”，只需移动“叶子”在整个行为树下的位置，便可以实现不同的决策逻辑，而不需要修改并检查整个决策部分代码。在进入下个赛季或者下下个赛季后，行为树可拓展性的功能便显得更为重要，无论规则改动还是开发者想要新加入任何行为，只需按照代码规范编写节点并加入树中即可。

2. 决策内容

在赛季初期我们采用的了随动导航的方案，即导航向下位机发送线速度 v_x 与角速度 w_z ，下位机控制云台转一个角度后底盘跟随云台转然后前进。然而自瞄状态下也需要云台 yaw 轴的控制权，因此哨兵只能在导航与自瞄间选择一个进行。

在赛季初期我们希望只有哨兵在自身危险状态下再转小陀螺方便获得更快更稳定的导航效果，因此需要根据裁判系统的自身扣血数据以及地方前哨站血量等数据判断哨兵是否处于危险状态。

3. 研发记录

在阅读 nav2 决策树部分的源代码和 behavior tree 的官方文档后，我们在分区赛前完成了哨兵决策部分代码的编写，并与自瞄，导航，控制等部分联调完成了计划中的效果。具体为导航优先，完成导航后云台 360 度扫描，扫描到后锁住敌方根据自动开火指令开火，目标短暂丢失时云台小范围扫描，到达一定时长未发现敌方则重新 360 度扫描。

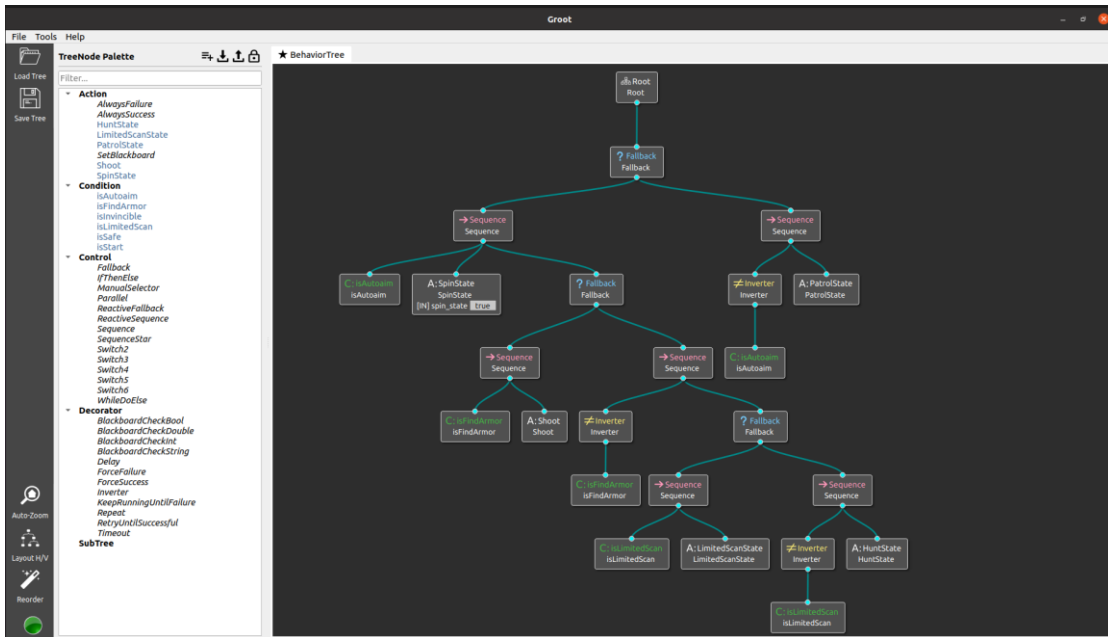


图 2.4.13 决策框架图

但是在调试以及上场时也暴露出了许多问题。例如裁判系统数据先由串口传到主控板，主控板再使用 can 通讯传到 nuc。这两个通信环节都会出现数据收不到的情况，导致决策代码无法正常运行。同时决策代码偶尔出现自启动启动不起来的情况，这在赛场上是非常致命的。

与此同时，在赛季末期全向感知、弹性跟随等复杂功能未能实现，导航上面完成技术突破底盘可以全向移动不需要争抢云台 yaw 轴的控制权的情况下，哨兵只需要简单地控制云台扫描搜索敌方的状态以及底盘是否小陀螺的状态。因此在全国赛上，哨兵云台的决策由主控板进行，nuc 只跑单节点的决策树代码以控制底盘小陀螺的状态。

4. 展望未来

随着代码 bug 的修改使得裁判系统数据较为稳定得读取以及自启动程序趋于稳定，当下

赛季需要加入全向感知、弹性跟随等复杂功能时，决策树这套代码框架将要重新启用，使哨兵在场上可以不依靠云台手点击的目的地自主决策目标点，自主完成各项功能。

2.5 其它

将云台发射机构部分设计成龙头造型，如图 2.5.1，提升了美观性。

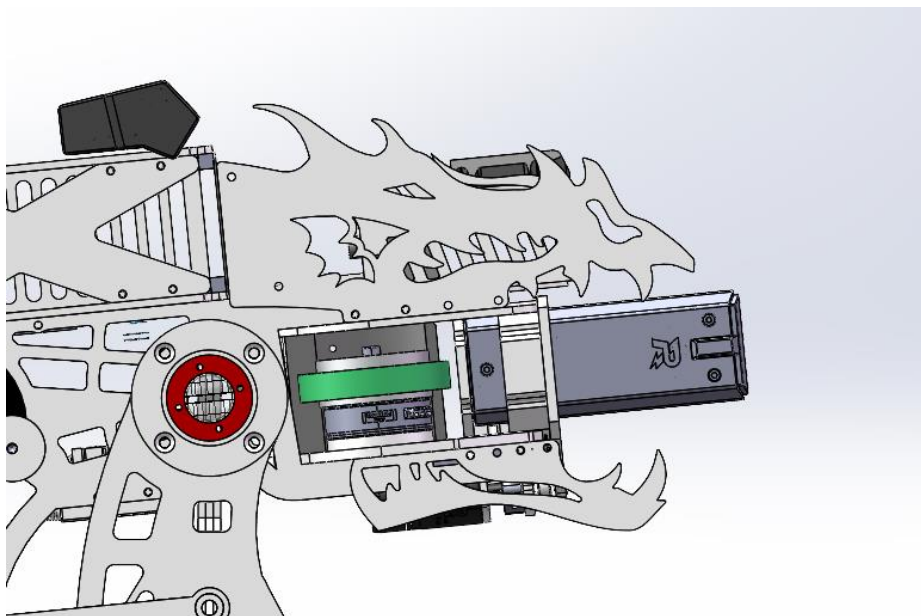


图 2.5.1 发射机构龙头造型

3. 研发迭代过程

3.1.1 测试记录

运动参数测试

运动参数测试主要测试在不同功率限制下的底盘在平面上的移动性能。

陀螺最大转速可以参考如下手段，通过读取陀螺模式下 Yaw 轴电机的码盘实际值获得底盘角度随时间变化的曲线，稳定后曲线应为正/余弦函数形式，取固定一段时间内曲线的周期数可以得到陀螺最大转速。

运动参数的测试结果见下表：

最大速度测试 (m/s) 录像, 时间 (平均速度)						
功率(W)	测试一	测试二	测试三	测试四	测试五	平均值
不限功率	2.34	2.35	2.41	2.37	2.42	2.378
150	2.13	2.31	1.95	1.98	2.07	2.088

陀螺最大转速测试 (rpm) 录像, 时间						
功率(W)	测试一	测试二	测试三	测试四	测试五	平均值
不限功率	141.1770652	141.8438	141.5104	141.5104	141.3437	141.4771
120	141.5104231	141.6771	141.5104	141.3437	141.3437	141.4771

射击参数测试

射击参数测试主要测试不同情况下的射击精度。

最大连续射频是不考虑哨兵的枪口热量，在代码中修改哨兵的参考射频，测试步兵能达到的实际射频。测试方法是在不同参考射频下连续让哨兵发射至少 100 发弹丸，并记录发射的总时间，利用弹丸数量和射击时间计算射频。下表中填入实际计算的射频，如测试过程中

发生卡弹的情况，则当次结果记录为卡弹。将从未发生过卡弹情况的实际射频作为哨兵的最大连续射频。

最大连续射频测试 (Hz)						
参考射频 (Hz)	测试一	测试二	测试三	测试四	测试五	平均值
近	8	8	9	8	7	8
中	4	5	5	5	4	4.6
远	3	2	3	2	2	2.4

单次测试射击弹丸数在 40-50 发，**射击准确度测试时务必注意安全!!!** 测试结果记录在下表。射击精度参数记录规则是，用一个最小矩形涵盖获得的弹道散布图中的所有弹丸点（可以忽略少数特别离谱的弹道，但忽略的数量不得多于 3 颗），将矩形的长宽记录在下表中。

静止状态下哨兵的射击精度					
测试条件	测试一 (连发)	测试二 (连发)	测试三 (连发)	测试四 (连发)	平均值 (连发)
3m	90*70	95*70	90*65	95*65	97.5*67.5
5m	120*70	115*70	120*65	115*65	117.5*67.5
7m	120*120	115*120	115*115	120*115	117.5*117.5

小陀螺模式下哨兵的射击精度					
测试条件	测试一 (连发)	测试二 (连发)	测试三 (连发)	测试四 (连发)	平均值 (连发)

小陀螺模式下哨兵的射击精度					
3m	170*65	165*70	160*65	165*65	165*66.25
5m	120*90	135*85	125*75	130*80	127.5*82.5

3.1.2 版本迭代过程记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	联盟赛哨兵 <ul style="list-style-type: none"> ● 基础单边发射机构发射能力 ● 基础原地自瞄打击能力 ● 原地随机小陀螺 ● 可动外边框，防止卡位 	2023. 4. 16
V2.0	分区赛哨兵 <ul style="list-style-type: none"> ● 并列双枪管发射能力 ● 携带全部 750 发弹丸 ● 稳定的自瞄打击能力 ● 原地随机小陀螺，随动下可稳定上坡 ● 稳定的随动导航避障，云台手可以控制其到指定点 ● 可由云台手控制改变击打策略 ● 可动外边框，防止卡位 	2023. 5. 18
V2.1	<ul style="list-style-type: none"> ● 增加了云台的校准模式与上坡检定 	2023. 5. 24
V2.2	<ul style="list-style-type: none"> ● 解决 miniPC 在编译或者运行时出现重启问题 	2023. 5. 29
V2.3	<ul style="list-style-type: none"> ● 更改遥控器为天地飞，重新确定遥控器的控制方式. 	2023. 6. 25

版本号或阶段	功能或性能详细说明	完成时间
	<ul style="list-style-type: none"> ● 增加电控方面小地图的控制。 	
V3.0	<ul style="list-style-type: none"> ● 改为切枪云台 ● 自动切枪持续火力压制 ● 携带全部 750 发弹丸 ● 稳定的自瞄打击能力，具有攻击敌方前哨战能力 ● 移动随机小陀螺，小陀螺下可稳定上坡 ● 稳定的边小陀螺边导航避障，云台手可以任意控制其到场地随机点 ● 可由云台手控制改变击打策略（将是否攻击工程加入） ● 可动外边框，防止卡位 ● 更轻的重量以及更小的转动惯量，有更快的响应速度和控制精度 	2023. 7. 11
V3.1	<p>国赛哨兵</p> <ul style="list-style-type: none"> ● 解决了 V3.0 的 Yaw 电机小角度不响应问题 ● 将云台控制与底盘控制分离，同时云台不再受到 minipc 的模式控制，只接收自瞄数据。 	2023. 7. 20

3.1.3 重点问题解决记录

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
1	Yaw 轴大虚位	Yaw 轴同步带未涨紧	增加两组轴承作外涨紧。 解决效果：完全解决	V1.0	机械工程师： 卢宇豪
2	Pitch 轴经常虚位	Pitch 轴连杆过长	在螺丝上缠生料带。 解决效果：暂时解决，需要经常检查加紧	V1.0、 V3.0、 V3.1	机械工程师： 卢宇豪
3	Yaw 轴电机响应不到位	Yaw 轴达妙 4310 电机外壳螺丝容易松动导致定子也有一定虚位	在使用 V3.0 版本的云台后，对 Yaw 轴电机扭矩需求减小，换回了 6020 电机 解决效果： 控制死区变小，Yaw 轴虚位完全解决	V2.0、 V3.0	机械工程师： 卢宇豪 机械工程师： 任朔
4	雷达长时间使用定位漂移	雷达自身发热大，且在保护球罩内不易散热	加装黑鲨手机散热 解决效果：雷达温度稳定在 25-30℃，长时间工作无漂移	V2.0	机械工程师： 卢德为 机械工程师： 卢宇豪

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
5	Minipc 在编译或者运行时出现重启	电池电压下降导致供给电压电流下降,在编译时所需电流电压不够,导致重启。	更换降压模块为升降压模块,即输入电压在一个范围内均可输出固定 19V 的电压。	V1.1	电控工程师: 祁灵
6	右侧测速机构发射弹丸弹道两点分布	弹道略微偏左且在装配时由于过盈配合使右侧测速机构偏右,部分弹丸会打到右侧测速机构左侧	通过将两段两杆同时增加 0.5mm,可以让右侧发射机构向左偏置 1mm。 解决效果:有效避免了其干扰弹道。	V3.0	机械工程师: 卢宇豪
7	随机概率的卡弹	下供弹推力不足;枪管与链路间安装有误差产生梯台;定位轴承过紧;老旧弹丸大小不一	通过增大枪管定位轴承间距直接减少对弹丸的阻力;同时扩大了枪管入口直径,使弹丸进入枪管经过一个扩大的阶梯。 解决效果: 使用新的弹丸不再卡弹,旧弹丸由于形变等原因还是有随机卡弹。	V3.0	机械工程师: 卢宇豪

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
8	云台旋转时无法较好做到底盘精准运动。	雷达陀螺仪数据发送频率过低	改为结合云台 C 板陀螺仪与雷达陀螺仪数据来做坐标系转换。	V3.0	视觉工程师： 张必豪 电控工程师： 祁灵
9	荧光充能不符合官方要求，前四发弹丸不满足充能要求	荧光充能位于供弹链路；自制荧光灯带功率不足无法充能到规则要求；哨兵射频较快，弹丸来不及充能	重新设计进入枪管的供弹链路，将官方荧光充能装置安装在其上。同时在鹅颈主链路上安装自制灯带及官方充能装置，对弹丸进行预充能。 解决效果：可以通过检录，但进入枪管的链路处散热较差，不宜长期使用，且充能装置的大量热量可能导致 miniPC 散热困难，可能导致 miniPC 降	V3.1	机械工程师： 卢宇豪 机械工程师： 任朔

4. 团队成员贡献

姓名	基本信息 (专业、年级、队内角色)	主要负责工作内容描述	贡献度 (所有成员贡献度合计为 100%)
卢宇豪	信息对抗技术、大三、哨兵云台机械负责人	负责哨兵机器人云台设计及维护	18%
张必豪	电子信息工程、大二、哨兵定位导航算法负责人	负责哨兵机器人定位导航算法设计和开发	18%
祁灵	通信工程、大二、哨兵电控负责人	负责整个机器人的嵌入式开发，包括底盘控制、云台控制开发等	18%
任朔	电子科学与技术、大一、哨兵底盘机械负责人	负责哨兵机器人底盘机械设计和维护	18%
池威律	机器人工程、大二、哨兵决策自瞄算法负责人	负责哨兵决策算法设计和自瞄算法调试	18%
卢德维	光电信息工程与技术、大四、哨兵云台机械设计	负责哨兵机器人云台设计	10%

5. 参考文献

参考内容	资料链接
哨兵决策树设计	https://github.com/ros-planning/navigation2 https://www.behaviortree.dev/docs/3.8/intro
哨兵定位算法: fast-lio2	GitHub - hku-mars/FAST_LIO: A computationally efficient and robust LiDAR-inertial odometry (LIO) package
哨兵定位算法: faster-lio	GitHub - gaoxiang12/faster-lio: Faster-LIO: Lightweight Tightly Coupled Lidar-inertial Odometry using Parallel Sparse Incremental Voxels
哨兵导航探索算法	Development Environment (cmu-exploration.com)
华南农业大学导航算法设计	技术报告 华南农业大学 Taurus RMU2022 技术报告开源 【RoboMaster 论坛-科技宅天堂】
南京理工大学机器人底盘设计	https://github.com/Alliance2023-Mechanical-Team/Mechanical-Technical-File
哈尔滨工业大学哨兵机械设计	RM2021-哈尔滨工业大学-I Hiter 战队-哨兵机器人-机械结构开源 【RoboMaster 论坛-科技宅天堂】
IMU 姿态解算	https://sourcelizi.github.io/202201/imu-err-calibration/
UI 绘制与车间通信	https://bbs.robomaster.com/forum.php?mod=viewthread&tid=11885
电控代码框架构建	https://bbs.robomaster.com/forum.php?mod=viewthread&tid=11885

6. 技术方案复盘

回顾本赛季技术方案，结合赛场实际的发挥表现，分析评估本赛季技术方案是否达到预期的效果。

6.1.1 赛场性能表现情况分析

我们的机器人团队设定了明确的功能目标，包括决策、导航和自瞄。我们希望机器人能够准确地感知自身在场上的位置以及其他机器人的位置，并以高效的策略完成比赛任务。决策系统在大部分情况下表现出色，机器人能够做出合理的决策。但是，在极端情况下过于依赖云台手的控制，导致出现前哨站被摧毁而云台手忘记操控哨兵返回巡逻区的意外。导航算法对机器人的控制较为准确，能够在复杂的环境中保持稳定定位及导航。我们的机器人成功完成了大部分比赛任务，包括进攻和防守。在进攻方面，机器人能够快速移动并击杀敌方机器人。在防守方面，成功拦截环形高地的英雄机器人，阻挡敌方进攻。

在与其他团队的比赛中，我们的机器人表现出色，赢得了一些精彩的对抗。然而，在一些比赛中，我们也意识到对手团队的策略可能超出了我们的预期，需要更灵活的应对策略。

我们遇到了一些突发状况，如通信中断等，这些情况影响了部分比赛的表现。在未来，我们需要更好地预防和处理这些问题。

6.1.2 赛场性能表现与规划对比分析

在比赛中，我们的目标是设计一个能够自主导航决策和自瞄的哨兵机器人。我们计划通过激光雷达和深度相机进行环境感知，使用路径规划算法实现智能巡逻，同时在使用工业相机进行自瞄。

实际赛场性能表现：

在比赛中，我们的哨兵机器人在很大程度上达到了预期目标，但也暴露了一些需要改进的问题。

机械方面在偶尔仍有卡弹问题，仍需突破改进，以及弹舱容量略小，可能与计算弹舱容量的方式有关。

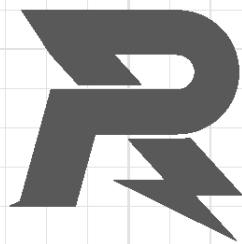
导航算法：哨兵机器人配备的激光雷达和深度相机在感知环境方面表现出色，能够准确

识别障碍物和异常情况。较好的完成了规划目标我们采用了先进的路径规划算法，使机器人能够智能选择巡逻路径，避开障碍物，并覆盖整个赛场。

自瞄与反陀螺：自瞄在学校场地测试较为稳定，基本完成了赛季规划。但在深圳测试出现了没有预料到的 bug，暂无解决方法。同时反陀螺的效果与上交等强队有较大差距。

6.1.3 经验总结

我们计划进一步优化机器人机械性能，采用舵轮等提高运动精度和机械上限。优化感知系统，以减少在密集环境中的误判情况。在决策系统方面，我们将探索更复杂的算法，以应对各种不同情况，并提高决策的灵活性。我们将加强机器人的定位和运动控制，以减少在紧张情况下的定位误差。针对突发状况，我们将设计备用方案，如快速更换电池或建立冗余通信机制。通过对比赛性能的客观分析，我们得出了明确的改进方向，以便在未来的比赛中取得更好的表现。



邮箱: robomaster@dji.com

论坛: <http://bbs.robomaster.com>

官网: <http://www.robomaster.com>

电话: 0755-36383255 (周一至周五10:30-19:30)

地址: 广东省深圳市南山区西丽街道仙茶路与兴科路交叉口大疆天空之城T2 22F