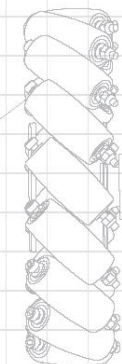




Using a 31-65 motor driver chip and Field-Oriented Control (FOC), the RoboMaster G200 Brushless DC Motor Speed Controller enables precise control over motor torque.



Designed for the RoboMaster G200 P18 Brushless DC Motor and G200 Brushless DC Motor Speed Controller, the M3000 Assembly Kit includes an arm, screws and a terminal board.

Reference System Specification Manual, Reference System User Manual, Introduction of Reference System Kit etc.

The M3000 Assembly Kit includes several screws and a terminal block, which is compatible with the system often by four independent system.

ROBOMASTER 机甲大师超级对抗赛 技术方案

华南农业大学 Taurus 战队 编制

2023 年 08 月 发布

前言

本成本报告由华南农业大学 Taurus 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	纪因	梁智恒
硬件	叶奕帆	
软件	刘旭龙	曾扬晨
算法	申国豪	吴静琳
排版	王蕾	

章节目录

前言 2

1. 哨兵功能综述	7
1.1 背景&目标	7
1.2 其它学校哨兵机器人分析综述	7
1.2.1 哈尔滨工业大学深圳	7
1.2.2 浙江大学	8
1.3 哨兵机器人功能定义	8
1.4 机器人核心参数	10
1.4.1 MiniPC 介绍	12
1.5 设计方案	14
1.5.1 机械结构设计	14
1.5.2 嵌入式硬件设计	27
1.5.3 嵌入式软件设计	36
1.5.4 视觉算法设计	44
1.5.5 导航算法设计	48
1.6 研发迭代过程	63
1.6.1 机械设计	63
1.6.2 嵌入式软件设计	67
1.6.3 视觉算法设计	68
1.6.4 导航算法	68
2. 团队成员贡献	70

图表目录

表格 1.2- 1 哈尔滨工业大学（深圳）哨兵分析	7
表格 1.2- 2 浙江大学哨兵分析	8
表格 1.4- 1 机械结构参数	10
表格 1.4- 2 机器运动参数	10
表格 1.4- 3 嵌入式硬件参数	10
表格 1.4- 4 传感器介绍	11
表格 1.4- 5 执行器件介绍	11
表格 1.4- 6 MiniPC 配置表	12
表格 1.5- 1 主板资源说明	28
表格 1.5- 2 哨兵上底板接口说明	29
表格 1.5- 3 哨兵下底板接口说明	30
表格 1.5- 4 单板硬件说明	31
表格 1.5- 5 底盘模式及行为	38
表格 1.5- 6 云台模式及行为	38
表格 1.5- 7 拨盘模式及行为	39
表格 1.6- 1 舵轮轮系底盘版本迭代过程记录	63
表格 1.6- 2 云台机械版本迭代过程记录	63
表格 1.6- 3 链路、发射器版本迭代过程记录	63
表格 1.6- 4 舵轮轮系机械重点问题解决	64
表格 1.6- 5 云台机械重点问题解决	65
表格 1.6- 6 链路、发射器重点问题解决	66
表格 1.6- 7 软件迭代过程记录	67
表格 1.6- 8 软件重点问题解决方案	67
表格 1.6- 9 导航迭代过程	68
表格 1.6- 10 导航重点问题解决方案	69
图 1.4.1- 1 MiniPC 实物图（斜）	13
图 1.4.1- 2 MiniPC 实物图（正）	13

图 1.5.1- 1 底盘轮系图纸	15
图 1.5.1- 2 轮系设计示意 1	16
图 1.5.1- 3 轮系设计示意 2	17
图 1.5.1- 4 轮系设计最终方案	21
图 1.5.1- 5 部分链路设计图纸	22
图 1.5.1- 6 云台走线示意	25
图 1.5.1- 7 底盘走线示意	25
图 1.5.1- 8 雷达安装位置	25
图 1.5.1- 9 IMU 安装位置	25
图 1.5.1- 10 底盘铝框架静力学分析	26
图 1.5.1- 11 底盘铝框架变形分析	26
图 1.5.1- 12 云台支撑板	26
图 1.5.2- 1 步兵硬件系统拓扑	27
图 1.5.2- 2 核心板硬件拓扑图	27
图 1.5.2- 3 自制核心板（正面）	28
图 1.5.2- 4 自制核心板（背面）	28
图 1.5.2- 5 哨兵上底板（正面）	29
图 1.5.2- 6 哨兵上底板（背面）	29
图 1.5.2- 7 哨兵下底板（背面）	30
图 1.5.2- 8 哨兵下底板（正面）	30
图 1.5.2- 9 降压电路原理图	31
图 1.5.2- 10 MOS 管控制电路开关以及电流采样电路	32
图 1.5.2- 11 电容硬件充放原理图	32
图 1.5.2- 12 主控芯片外围电路原理图	33
图 1.5.2- 13 充放电硬件框图	34
图 1.5.2- 14 超级电容软件控制流程	35
图 1.5.3- 1 步兵机器人软件框架	36
图 1.5.3- 2 系统模式切换图	37
图 1.5.3- 3 系统状态切换图	37
图 1.5.3- 4 底盘状态图	37

图 1.5.3- 5 摩擦轮与拨盘状态控制	38
图 1.5.3- 6 发射器模式切换	39
图 1.5.3- 7 拨盘热量与	41
图 1.5.3- 8 底盘功率控制算法流程图	41
图 1.5.3- 11 陀螺仪软件架构	43
图 1.5.3- 9 陀螺仪数据发送任务	43
图 1.5.3- 10 陀螺仪数据读取任务	43
图 1.5.3- 12 姿态解算任务	44
图 1.5.3- 13 陀螺仪状态检测任务	44
图 1.5.4- 1 识别计算流程图	45
图 1.5.4- 2 自瞄效果图	46
图 1.5.4- 3 预测流程图	46
图 1.5.4- 4 重力补偿计算	47
图 1.5.4- 5 反小陀螺	47
图 1.5.4- 6 X 轴差分曲线	47
图 1.5.4- 7 数字识别效果图	48
图 1.5.5- 1 整体算法设计框架	48
图 1.5.5- 2 Rqt_graph 图	49
图 1.5.5- 3 获取点云数据流程图	50
图 1.5.5- 4 ICP 算法坐标关系	53
图 1.5.5- 5 ICP 算法坐标关系 2	53
图 1.5.5- 6 路径规划流程图	56
图 1.5.5- 7 决策算法结构设计	59
图 1.5.5- 8 哨兵仿真	60
图 1.5.5- 9 导航测试	60
图 1.5.5- 10 动态调参工具示意	61
图 1.5.5- 11 初代哨兵保护壳方案	62

1. 哨兵功能综述

1.1 背景&目标

今年在哨兵机器人规则发生了重大变化。过去，哨兵机器人只能在一个固定的轨道上运行，但现在哨兵机器人可以在整个场地范围内自由移动。此外，在前哨站被攻击前，哨兵机器人不会失去任何血量，这使得机器人在比赛中发挥出更为优异的表现成为可能。

考虑到这些改变，研发一款全场自主运动、自主瞄准并能根据云台手的指令执行特定收集任务的哨兵机器人变得非常必要。我们的目标是开发一种智能化的哨兵机器人，它能够自主感知周围环境，并且利用在视觉和雷达技术方面的进步，确保机器人能够准确地识别目标机器人并自瞄，根据云台手的指令进行特定的任务。

1.2 其它学校哨兵机器人分析综述

结合往年比赛视频、其它学校开源资料、论坛等信息，分析其它学校机器人各项功能的完成度及技术水平，概述值得借鉴之处或者有待改进之处。

由于哨兵规则与往年有较大变化，往年哨兵机器人不具参考意义，主要参考 RMUA 和自动步兵机器人设计。

1.2.1 哈尔滨工业大学深圳

信息来源	功能完成度	技术要点	研发启示
RMUA2021 八强技术答辩 https://www.bilibili.com/video/BV1UU4y1V7yW/?spm_id_from=333.337.search-card.all.click&vd_source=b91afc82e63b14bfbf720e51d03b61b0 (出处: bilibili)	1. 有非常稳定的定位系统，使用了 icp+ekf 的方案 2. 做了比较完善的仿真系统	1. icp+ekf 实现机器人定位 2. gazebo 地图环境搭建和机器人 urdf 模型建立	1. 改进机器人的定位系统 2. 建立更好地仿真环境进行算法验证

表格 1.2-1 哈尔滨工业大学（深圳）哨兵分析

1.2.2 浙江大学

信息来源	功能完成度	技术要点	研发启示
HW 自动步兵不完整形态视频 https://www.bilibili.com/video/BV1XT4y1a7bZ/?spm_id_from=33.337.search-card.all.click&vd_source=b91afc82e63b14bfbf720e51d03b61b0	1.有非常稳定的定位系统，使用了时间同步+icp+ukf 方案 2.决策做反击系统	1.时间同步+icp+ukf 实现机器人的定位 2.读取裁判系统信息大致判断敌方位置	1.改进机器人的定位系统 2.建立更智能的决策系统

表格 1.2-2 浙江大学哨兵分析

1.3 哨兵机器人功能定义

- 底盘功能设计
 - ◆ 全地形下不会出现翻车
 - ◆ 有小陀螺功能
 - ◆ 可以实现飞坡
 - ◆ 机动性强
 - 四驱独立悬挂
- 云台设计
 - ◆ 实现三轴云台，保证射击稳定性
- 射击系统
 - ◆ 实现下供弹链路
 - ◆ 带视觉的辅助瞄准
 - ◆ 系统稳定
 - 不能出现卡弹
 - 射速、射频可以稳定控制在官方设定的上限值
- 定位功能
 - ◆ 实现稳定的定位系统
 - ◆ 受到撞击或雷达受击打损坏处理

- ◆ 定位丢失重定位
- 导航功能
 - ◆ 根据云台手指令到达指定位置
 - ◆ 稳定的避障系统
- 决策功能
 - ◆ 被攻击后躲避或反击功能
 - ◆ 自主巡航功能
- 其他功能
 - ◆ 便于维护
 - 小模块损坏 1 分钟内可修复
 - 大模块损坏 3 分钟内可修复
 - ◆ 稳定性高，不易损坏
 - ◆ 整体外形美观
- 功能量化举例
 - 射击系统稳定
 - 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 26m/s
 - 射速方差不大于 2

1.4 机器人核心参数

项目	参数 (单位)
长	670(mm)
宽	670(mm)
高	523(mm)
整车重量	26.55 (kg)

表格 1.4-1 机械结构参数

项目	参数 (单位)
云台自由度	2
云台俯仰角	-17 ~ 25 (°)
底盘最大移动速度	2.86(m/s)
爬坡角度	45(°)

表格 1.4-2 机器运动参数

项目	参数 (单位)
电路板总功耗	3 ~ 4 (W)
电容组容值	1968.3 (J)
电容组工作电压范围	12.5 - 23.7 (V)
电容组最大充电电流	10 (A)

表格 1.4-3 嵌入式硬件参数

器件	数量/个	功能
思岚 A2 激光雷达	1	场内定位传感器
CH110 九轴陀螺仪	1	底盘 IMU
500W 像素广角摄像头	1	雷达站视觉定位
自制 IMU	1	提供云台的姿态角与角速度

表格 1.4-4 传感器介绍

器件	数量/个	功能
M3508	4	底盘四轮驱动电机
GM6020	6	云台两轴驱动电机 底盘四轮转向电机
M2006	1	发射器拨盘驱动电机
朗宇电机	2	摩擦轮驱动电机

表格 1.4-5 执行器件介绍

1.4.1 MiniPC 介绍

对于小电脑的选型，我们在保证电脑足够的性能下，性价比、体积小巧的 MiniPC，具体参数配置及实物图如下：

处理器	第十代英特尔酷睿i7-10510U 处理器 (1.8GHz-4.9GHz睿频加速技术, 四核八线程, 15W功耗)
内存	DDR4 2133Hz-2666Hz(最大支持64G)
内存	双通道DDR4-2666 SO-DIMM插槽 (最大可达64GB)
图形	英特尔® UHD620核心显卡 HDMI 2.0a 端口, 4K 60Hz DP端口, 4K 60Hz
音频	高达7.1环绕立体声 (通过HDMI 和 DisplayPort) 前面板配有耳机/麦克风接口
外设链接	2个 USB 3.0 端口 (前面板) 2个 USB 3.0 端口 (后面板) 2个内部 USB 2.0 端口 1个Type-c 端口 (不支持视频输出)
存储	1个 M.2 22X80 插槽, 支持 NVME 1个SATA 3.0 插口 支持2.5寸HDD\SSD
网络	英特尔® 千兆有线网卡 英特尔® 双频 Wireless-AC 3160无线网卡 (802.11ac) 2x2 双模蓝牙®4.0
系统BIOS	64Mb 闪存 EEPROM, 英特尔®平台创新框架用于EFI即插即用 高级配置和电源接口 V3.0b, SMBIOS 2.5 英特尔®可视 BIOS 英特尔®快速BIOS更新支持
机箱尺寸	128mmX128mmX38mm
电源	19V, 45W AC-DC 电源适配器

表格 1.4- 6 MiniPC 配置表



图 1.4.1-1 MiniPC 实物图（斜）



图 1.4.1-2 MiniPC 实物图（正）

1.5 设计方案

1.5.1 机械结构设计

作为 23 赛季新兵种自动哨兵，高血量、高冷高热与携带 750 发弹丸的设定，注定了其在风起云涌的赛场上定海神针般的角色担当。而一台赛场稳定发挥的哨兵更是队伍中不可或缺的一支输出火力，机动性较强的底盘设计与减震方案、稳定流畅的链路、精准输出的弹道、整车的轻量化等都对机械设计提出较高性要求。

基于上述设计需求，本赛季我们决定打造一台采用舵轮、下供弹设计搭配朗宇电机发射器的自动哨兵机器人。

1.5.1.1 舵轮底盘轮系设计

基于底盘功能的要求且需要放置激光雷达进行场地扫描，可以得出底盘有以下要求痛点：1、机器需要能在高性能下作战，保持高度的灵敏性同时又不能导致车体摇晃提高雷达识别难度，2、同时也需要提供更快速的响应时间，从而提高车辆的操控性能；3、再者从成本出发，需要其由更少的零件组成，使其更容易维护和修理；4、此外，高耐用性也是不可缺少的，由此可以满足各种地形下的行驶需求。

则可以得出，底盘可以使用自适应的舵轮轮系作为底盘轮系。通过参考发射机构树脂包胶摩擦轮的驱动原理，使用树脂包胶轮，硬度选择 85A；同时使用平行四边形连杆机构作为自适应机构中中心传动连杆与舵之间的连接，，由此确保舵的旋转中心轴始终与地面垂直。

其次是轮径的大小设计，轮径大小对车辆的影响是多方面的。首先，较大的轮子会提高离地高度，从而增加了车辆的通过性。这意味着它可以更容易地通过高速行驶时的障碍物和起伏路段。此外，在起伏路况下，较大的轮胎能够提供更好的牵引和悬挂性能，以保持稳定和舒适的行驶。其次，轮胎的直径对车辆的速度会产生影响。因为轮胎旋转的速度与车速成正比，当轮胎的直径增加时，相同的车速需要更少的轮胎旋转，最后，更大的轮径会改变车辆的悬挂，同时舵需要更大的力矩使其转动，较大的车轮需要更多的扭力来运动。计算轮径的大小需要考虑几个因素，包括车辆行驶速度、发动机驱动件性能和传动机构设计。

以下是一个大致的计算公式，可用于估算实际所需的轮径大小：轮径 = (机器行驶速度 $\times 60 \times 60 \div (2 \times \pi \times$ 理想的发动机转数) + 车辆总传动比) \div (麦克劳林悬挂高度 $\div 50$)

其中，车辆行驶速度是以千米/小时为单位，理想的驱动件转数是指驱动件在最佳运转效率下工作的转数，车辆总传动比包括差速器和变速器的总传动比，麦克劳林悬挂高度是指前后悬挂点的距离。机器行驶速度 $V=2.8\text{m/s}\approx 10\text{km/h}$ 、理想的发动机转数=469rpm、车辆总传动比为 1: 19、麦克劳林悬挂高度 $h=72\text{mm}$ ，通过公式计算可以得出轮径 $D\approx 135\text{mm}$ 。这个公式是一个比较基础的计算方法，仅供参考，实际上还需要根据具体的情况进行调整和修正。例如，在起伏路段中，可能需要更大的轮径来提高牵引力和通过性；而在平滑路面中，更小的轮径可以提高机器的电能使用率和操控灵活性。

最后选用 136.75mm 作为轮径，理想最大移速 2.8m/s。

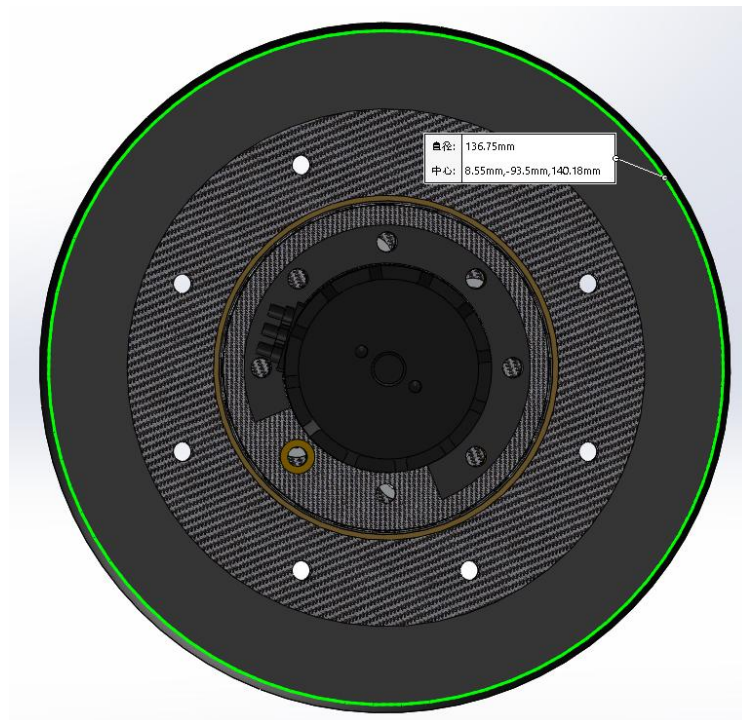


图 1.5.1-1 底盘轮系图纸

使用深沟球 61812-2Z 与玻纤板以及打印件将包胶轮进行链接，此时轮系总重量达到了 1.27kg，自转惯性矩可以表示为： $I = (2/5) * m * r^2$ ，

可以得出 $I=0.0095\text{kg}\cdot\text{m}^2$ ，而转动力矩 $\tau = I * \alpha$ （ α 表示物体角加速度），那么物体的角加速度 α 就可以表示为转动力矩除以自传惯性矩，即 $\alpha=\tau / I$ ；此时使用 6020 电机，在额定扭矩下能提供 $1.2\text{N}\cdot\text{m}$ ，可以得出角加速度 α 最大可以达到 126.3rad/s^2 ，通过 $70*120*8.5$ 大小的餐桌转盘连接舵与轮系结构，随后使用平行四边形连杆机构作为自适应机构，避震弹簧使用两条尺寸为 $2*20*54$ ，材料为 SWP-A 型号弹簧钢。

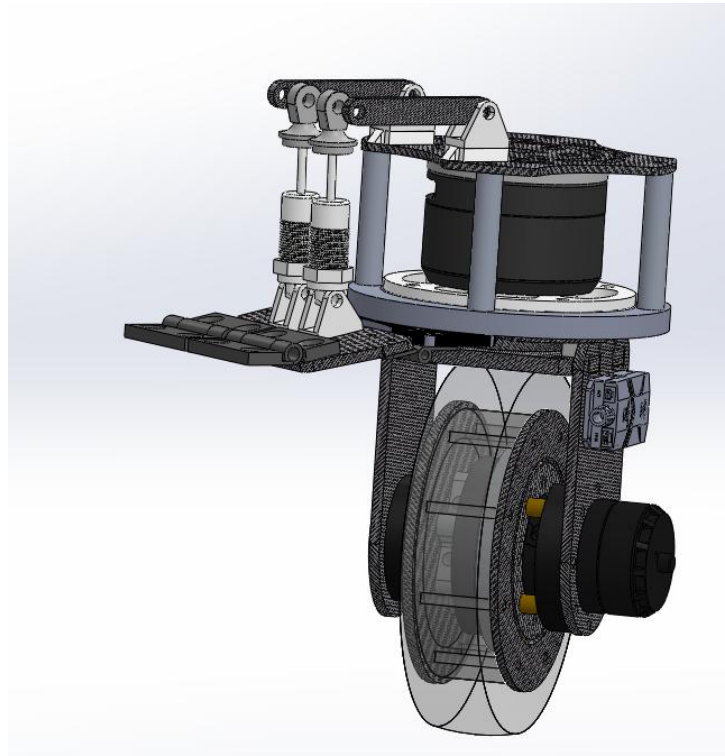


图 1.5.1-2 轮系设计示意 1

随后，在底盘的测试中，自适应机构所能使机器框架稳定的能力范围有限，无法使激光雷达每时每刻皆能准确的读取身边障碍的精确位置，同时自适应的无时无刻的高度反应所造成的的机械震动使陀螺仪产生较大的角度误差，对于数据分析所造成的机器导航难度有所提升；且鉴于哨兵血量较高且前期对方所能造成的伤害不大，所以哨兵对于急速转向能力并不做高度需求，而对于框架平稳稳定方面有较大的需求，同时对于避震机构的反应速度也有更大的需求量；由此可以得到全新的需求痛点：第一、每个车轮都需要有相对应的独立地运动，从而提供更好的悬挂性能，使机体更加平稳，从而激光雷达可以获得更稳定的点云数据，由此获得更加良好的数据采集；第二、更精确的悬挂调校，高度的反应速度能使车辆适应更多地形，而不会使激光雷达出现较大误差。则初版底盘可以得出柔性连接的独立悬挂结构舵轮轮系更适合与本赛季的独立悬挂机器人。

在第二版的轮系设计当中，选择直线滑轨滑块与液压弹簧避震作为轮系与机器底盘的柔性连接，直线导轨滑块与框架平面垂直固定，则舵的旋转中心轴始终与地面垂直；为使总舵轮轮系在旋转中心轴的方向的虚位能尽可能趋近于 0，将 70*120*8.5 的餐桌转盘替换为 80*100*10 的深沟球轴承，将虚位所产生的震动最大程度消除，直线滑轨滑块的应力计算需

要考虑多个因素，如载荷大小、滑动速度、工作环境温度等。一般地，可以使用以下公式来近似计算直线滑轨滑块的应力： $\sigma = P / A$ 其中， σ 为滑块所受应力，单位是帕斯卡（Pa）； P 为滑块的载荷即所承受的力，单位为牛顿（N）； A 为滑块的正面横截面积，单位为平方米（ m^2 ）。还需要综合考虑其它因素，例如材料强度、滑动方式等通过综合考虑，选用 MGN9C 型号的直线导轨滑块作为柔性连接的连接件。

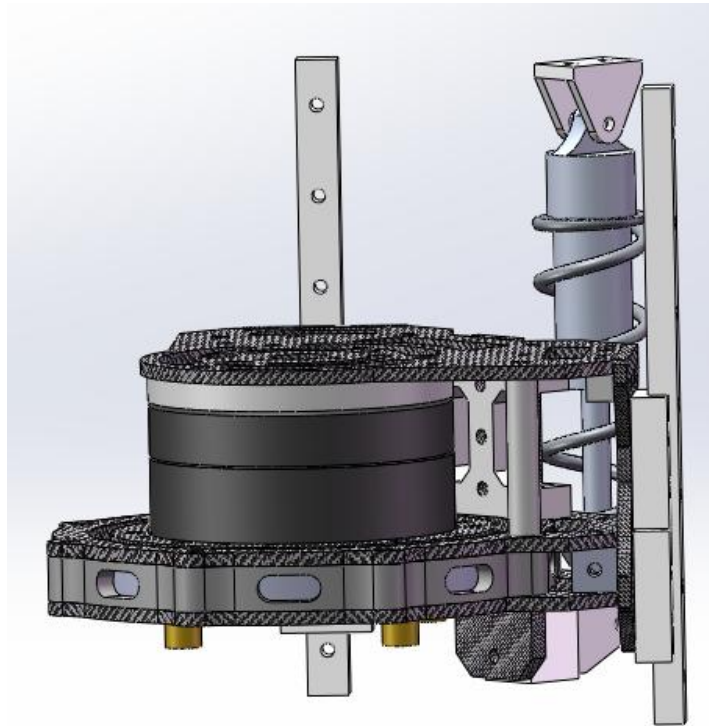


图 1.5.1-3 轮系设计示意 2

同时，原使用的弹簧为 SWP-A 型号弹簧钢制成的尺寸为 2*20*54 的避震弹簧。

按照胡克定律，弹簧的弹力与其形变量成正比。设该段弹簧的劲度系数为 k ，则当它被压缩 18mm 时，所产生的恢复力 F 可以用下列公式计算：

$$F = k * x$$

其中， x 表示该段弹簧的形变量，即 18mm，在国际单位制中应换算为 0.018m。因此，上式可以写作

$$F = k * 0.018$$

要确定该段弹簧的劲度系数 k ，可以使用以下公式：

$$k = G * d^4 / (8 * D^3 * n)$$

其中，G 表示弹簧钢的剪切模量，d 表示弹簧钢线径，D 表示弹簧钢外径，n 表示弹簧钢的圈数。

已知该段弹簧的材料是弹簧钢，根据资料可知其剪切模量通常在 77-81 GPa 之间，根据材料弹性模量 $E=206\text{GPa}$ ；弹簧钢的泊松比一般取 0.3，则剪切模量 $G = E / (2 * (1 + \nu)) = 79.2\text{GPa}$ ，为了保守起见我们取其中一个均值 79 GPa。将其他已知参数代入公式可以得到：

$$k = 79 \times 10^9 \text{ Pa} \times (2 \times 10^{-3} \text{ m})^4 / (8 \times (20 \times 10^{-3} \text{ m})^3 \times 12)$$

k 约等于 1645.8 N/m 。

将 k 带入第一个公式可得该段弹簧被压缩 18mm 时的最大弹力：

$$F = 1645.8 \text{ N/m} \times 0.018 \text{ m}$$

F 约等于 29.62N。

因此，该段弹簧被压缩 18mm 时的最大弹力约为 30 N。

当弹簧被压缩到最大行程时，它所储存的最大弹性势能等于弹簧的变形能，即所需的功。设该段弹簧在被压缩 18mm 后储存了最大的弹性势能 E，根据物理学中的公式，可以用下列公式计算：

$$E = (1 / 2) * k * x^2$$

其中，k 表示该段弹簧的劲度系数，x 表示弹簧的压缩位移，在本例中为 0.018m。而上一问计算得到该段弹簧的劲度系数为约 0.76 N/m，代入上式可计算出：

$$E = (1 / 2) * 1645.8 \text{ N/m} * (0.018 \text{ m})^2$$

约等于 0.267 J（焦耳）

因此，当该段弹簧被压缩到 18mm 时，储存的最大弹性势能为约 0.267 J（焦耳）。

一个轮系有两条避震组成则一个轮系中避震弹簧结构最大提供 60 N 弹力、最大储存 0.534 J 弹性势能。也就是需要较大的力才能使其变形，其运动速度变慢。对于起伏路段悬挂调校效果较差

于是更换为弹簧线径 1.8mm，外径 24mm，长度为 110 毫米的中轴加粗避震，用一根避震

代替两根，做到其由更少的零件组成，使其更容易维护和修理，同时，使用同样的计算方法可以得到，

设该段弹簧的劲度系数为 k ，则当它被压缩 35mm 时，所产生的恢复力 F 可以用下列公式计算：

$$F = k * x$$

其中， x 表示该段弹簧的形变量，即 35mm，在国际单位制中应换算为 0.035m。因此，可以写作

$$F = k * 0.035$$

$$k = G * d^4 / (8 * D^3 * n)$$

$$k = 79 \times 10^9 \text{ Pa} \times (1.8 \times 10^{-3} \text{ m})^4 / (8 \times (24 \times 10^{-3} \text{ m})^3 \times 10)$$

k 约等于 750 N/m。

将 k 带入第一个公式可得该段弹簧被压缩 35mm 时的最大弹力：

$$F = 750 \text{ N/m} \times 0.035 \text{ m}$$

F 约等于 26.25 N。

因此，该段弹簧被压缩 35mm 时的最大弹力约为 26.25N。

当弹簧被压缩到最大行程时，它所储存的最大弹性势能等于弹簧的变形能。

设该段弹簧在被压缩 35mm 后储存了最大的弹性势能 E

$$E = (1 / 2) * k * x^2$$

其中， k 表示该段弹簧的劲度系数， x 表示弹簧的压缩位移，为 0.035m。该段弹簧的劲度系数为约 750 N/m，代入上式可计算出：

$$E = (1 / 2) * 750 \text{ N/m} * (0.035 \text{ m})^2$$

E 约等于 0.46 J（焦耳）

所以该避震机构弹簧最大提供 26.25 N 弹力、最大储存 0.46 J 弹性势能。更小的变形需求，可应对地面的起伏与车身的倾斜做出更快的响应。

同时其中的液压缸可以提供最大 140N，可以承受飞坡所需的缓冲要求。

在接下来的测试当中，下轮系转速一直无法达到理论最大的 2.8m/s，经过实际测试，得出结果为包胶轮在地胶等硬度低地面上行驶，会使轮子陷入地面当中，使轮子与地面的接触面积增大，在相同压力的情况下，接触面积增大则单位面积上所分配的压力减小，根据摩擦力计算公式 $F_f = \mu * F$ ，则在接触双方不变化摩擦系数不改变的情况下，单位面积上的摩擦力减小，则由摩擦力所体现出来的单位面积上的牵引力也相对减小，由于接触面积增大以及轮子的形状构造，压力所造成的摩擦力方向不同，部分单位面积上的摩擦力由体现为牵引力转化为不体现或体现为阻力，从而使转动阻力增大。

为解决此问题，下轮系需要减少轮子与地面之间的接触，同时将接触面集中于轮子外圆柱面上，同理可得其所体现出的牵引力增大，转动加速度与转动最大速度增大，同时由于弹簧更换，轮径需要再次计算，机器行驶速度 $V = 2.5\text{m/s} \approx 9\text{km/h}$ 、理想的发动机转数=469rpm、车辆总传动比为 1: 19、麦克劳林悬挂高度 $h = 110\text{mm}$ ，可以得出轮径 D 大小为 120mm，同时将原来的树脂包胶轮转换为胎皮包裹轮毂，胎皮所使用的橡胶更为柔软，对于路面上的细小起伏具有缓冲能力，以及更好的耐用性；胎皮上的花纹能减少轮子与地面的接触面积，提供更好的抓地力以及更好的制动能力；同时花纹内的缝隙能更好的散发摩擦带来的热量，延长使用寿命；胎皮最小厚度为 6mm，则可以轮毂外径大小为 108mm。

最后可以得到以下轮系：

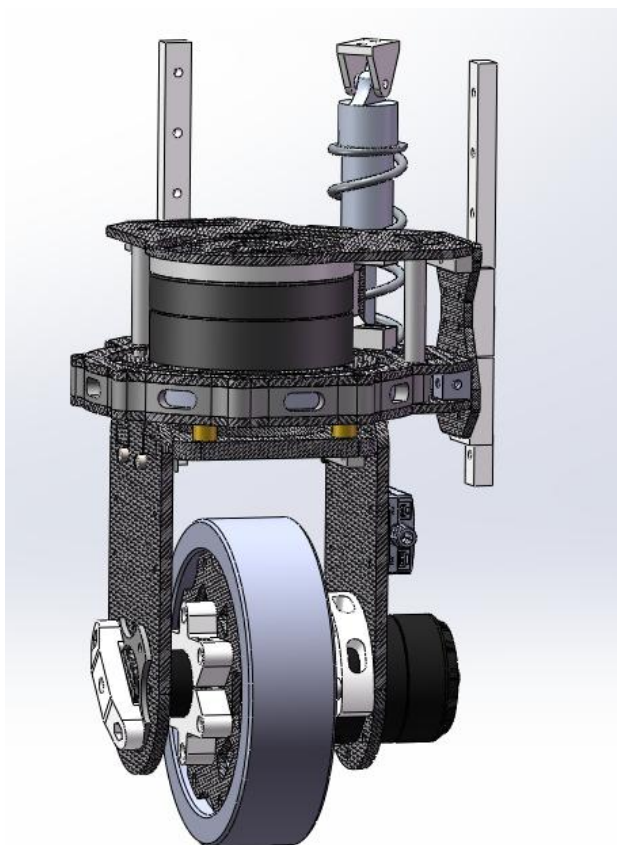


图 1.5.1-4 轮系设计最终方案

1.5.1.2 下供弹链路设计

下供弹方案自带长链路、多直角转弯、能量损耗多等不可避免的痛点，导致其对设计、零件公差、装配等的要求更高且容错量更小。在学习 RM 圆桌第一期下供弹机构的秘密和借鉴队内旧哨兵的链路设计，对本台自动哨兵链路设计有了初代的设计。

首先，考虑到本赛季下供弹自动哨兵先天链路转弯多个的缺陷，我们决定将发射机构置于云台 p 轴前端以此多余直角转弯，所以本台自动哨兵共有五个直角转弯。另外，由于滑环过子弹要求导致定制的滑环高度尺寸达 $100\text{mm}+$ ，子弹竖直上升直通道也较长，这里我们采用外径 20 内径 18 的碳管充当上升链路。由于链路的长度过长与通道内径存在余量导致子弹活动空间积累叠加，本台自动哨兵无法精确保证每一颗子弹每一刻的相对坐标，因此链路无法时时刻刻做到从拨齿至限位的空间中子弹刚好整数颗填充。而每一颗子弹受力不一也造成了本台机器在进行弹道测试时效果不佳，甚至因为链路中子弹的错位无规则排列出现能量损耗较大而无法突破限位发射子弹。在后续测试迭代中，我们通过减少轴承间间隙、调测转弯合适曲率、使用小轴承更贴切拟合子弹路径来减少链路中能量损耗。另外，自动哨兵的拨

盘继续沿用队内研发的切拨盘设计，但由于底盘结构设计及弹仓尺寸考虑，出拨盘口到达云台 yaw 轴也只能链路加长。在整车链路中，我们多用半径 20mm 的子弹路径进行直角转弯，同时搭配使用 2*5*2.5 的迷你小轴承，子弹通道内径保留 17mm 的空间。本台机器链路是利用光固化转接件与碳板拼接而成，最终在测试中整车链路基本达到使用要求。

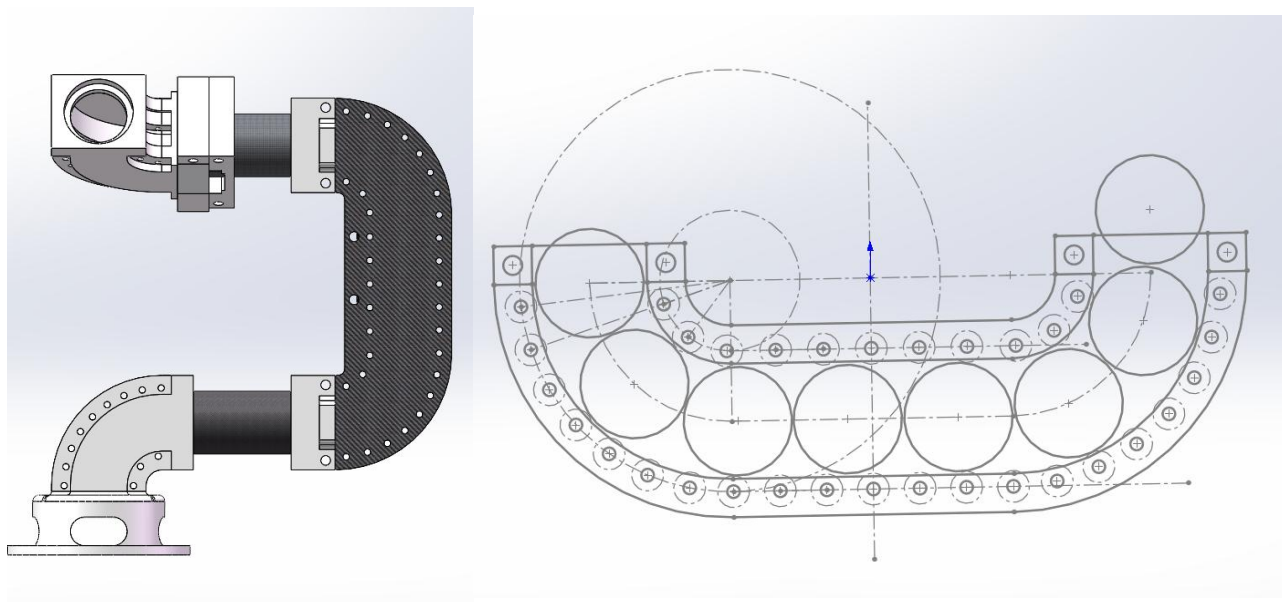


图 1.5.1-5 部分链路设计图纸

1.5.1.3 朗宇发射器设计

在本赛季中自动哨兵使用的是摩擦轮搭配朗宇电机的发射器方案。

由于链路长导致链路中能量损耗过大，因此相比于步兵发射器，自动哨兵在限位顶针作用我们削弱很多。其次，在这里我们使用的是 u 型轴承来进行子弹限位矫正入射姿态，设计光固化打印件作为转接件与枪管的一体化整体。但由于链路并非能够被子弹整数填充与限位顶针作用的削弱，在测试中子弹限位调测范围十分有限而且精度要求也十分苛刻，一方面要防止限位太紧导致卡弹，另一方面又要保证限位作用起效，实属左右为难。而在这其中唯一可以安慰的地方就是，我们也测试过无限位以此想要直接避免此问题，但测试中漏弹+链路由于未填满/抬头发射而有时出现子弹拨出后一直滚动，达到发射点时状态极其不一，出现 a4 纸效果。因此无限位当场否决。

另外，在初代测试中，我们发现摩擦轮损耗十分严重，小于一千发，有效稳定 550 发以内，之后出现白痕且掉速严重。在一开始中我们使用的是沿用许久的旧朗宇电机（这里称为朗宇 1.0），后我们选用限位测试 1.0 的方案进行了单一变量控制测试（仅更换电机，摩擦轮损耗确保无明显差异情况下）。在后面对比后，发现朗宇 2.0 较 1.0、3.0 更适合。其电机拉满状态下最高射速可达 35m/s，因此相对充足的电机余量也给电机对摩擦轮明显损耗时能做到维持射速调节的更多时长。在限位测试 1.0 中，朗宇 2.0 将摩擦轮使用寿命提升成至少 1500 颗。

在朗宇电机发射器测试中，我们选择以下方案：

朗宇电机 980kv->1250KV

摩擦轮硬度 50->60 摩擦轮间距 12.6->12.75

散布：连发 110 发，限位和枪管做好的话，7m 能够 90%在一个小装甲板内。

轴承方案：单轴承前限位

磨损度：老方案 1000 发外壁有白皮有明显大凹陷且明显掉速->新方案 3000 发外壁少量白皮无明显凹陷无明显掉速。

连发射速波动范围：26.9—29.4 平均 28 左右

单发射速波动范围：27.8—29.4 平均 28.5 左右

1.5.1.4 工艺选择

1) CNC 加工：

加工质量稳定，加工精度高，重复精度高且多品种、小批量生产情况下生产效率较高。另外可加工常规方法难以加工的复杂型面，甚至能加工一些无法观测的加工部位。在精度高且加工件能同时满足高强度要求，CNC 无疑是精加工零件较为完美的选择。但 CNC 费用较为昂贵，而且加工件本身密度大较光固化打印件重，所以在选择是往往需要多方考虑。对于 CNC，本台自动哨兵多用于电机联轴器、发射器高精度要求零件、部分传动件、需要加工螺纹孔的零件诸如此类。

2) 3D 打印（SLA、PLA、ABS）：

首先 SLA 其工作原理是使用高功率激光来硬化容器中的液态树脂，以产生所需的 3D 形状。简而言之，该工艺使用低功率激光和光聚合以逐层方式将光敏液体转化为 3D 固体塑料。而 SLA 本身精度要高于常规 3D 打印，其打印表面光滑、可接受打印复杂的形状。其缺点是外包加工工期叫实验室打印机打印要长很多且价格较贵，光固化本身也易脆，耐磨性输于 PLA 材料打印件。对于光固化，我们多选择用于链路中具有复杂需求的直角弯转接件、高精度打印件、发射器限位转接。另外在强度、精度要求要求不高的零件多用队里 3D 打印机进行零件打印，多 PLA 材料，例如 yaw 轴垫高件、弹仓落弹斜台、拨盘、轴承限位圈等。

3) 板材加工（玻纤板、碳纤维板）：

从板材本身出发，其实玻纤与碳板并无很大区别，其强度、硬度等无明显差异。但碳板较轻与玻纤，而且其加工精度也高于玻纤一些，因此碳板也较价格较贵。在测试版本中的机器我们往往会选用玻纤进行初代测试以此减少研发成本，而自动哨兵长链路高精度要求我们选择碳板作为链路板材对链路轴承进行较高精度定位与安装。另外，在要求较低的板材需求，我们也多考虑选用玻纤解决问题。

4) 铝管加工（手工切除打孔、机床加工孔位）：

在机器底盘的铝材框架中我们选择用铝方管，额外发加工亚克力板作为孔位定位板进行手工打孔加工。我们利用夹具将亚克力板与铝管进行一个夹紧定位，先用手钻打出定位孔，之后更换钻头进行梯度性扩孔确保同心度，减少孔位的打歪。而对于精度非常高的定位基准孔，我们会选择在向店家购买铝管同时额外加价机床打孔。在手工切除或者店家免费切除铝管时要求正公差，若有实际偏差会进行手磨削减。

5) UPE 雕刻加工:

UPE 板极好的耐磨性，良好的耐低温冲击性、自润滑性。其机械性能高于一般的高密度聚乙烯，具有突出的抗冲击性、耐应力开裂性、低摩擦系数、抗疲劳性、噪音阻尼性等。因此我们选用 UPE 作为自动哨兵的轮毂，甚至保护杠一部分。

1.5.1.5 传感器的设计安装、电路板的固定及走线情况

在本台自动哨兵中最特别的便是选择将两个激光雷达放置于舵轮上（安装位置如图 1.5.1-8 所示），利用舵轮的减震来减少因车身运动震动对扫描数据的不实影响。为了尽可能配平底盘重心，所以我们选择将 IMU 放置于电源管理模块旁边（如图 1.5.1-9 所示）。其次，利用板材镂空对底盘线路进行走位，合理布局电路板位置方便检查拔插、保护壳设计简易（如图 1.5.1-7 所示）。对于 yaw 轴滑环走线，我们利用轴承内圈与链路碳管之间的圆环空间进行线路走位，同时在云台中间横板、侧板镂空出足够的空间，尽可能使云台核板、小电脑走侧板内面以此板材自身增加电路保护功能，减少额外的保护壳设计（如图 1.5.1-6 所示）

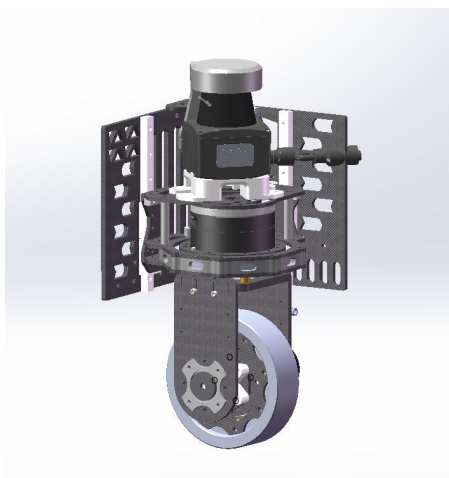


图 1.5.1-8 雷达安装位置

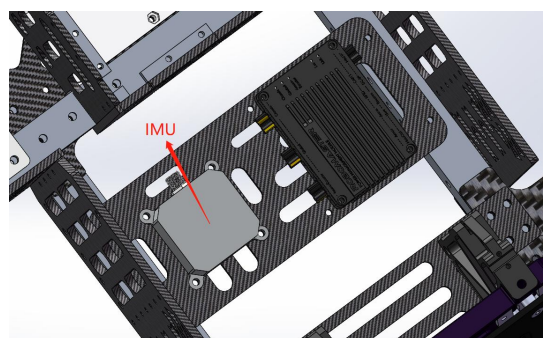


图 1.5.1-9 IMU 安装位置

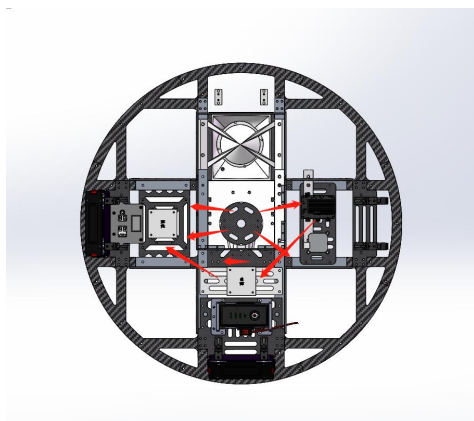


图 1.5.1-7 底盘走线示意

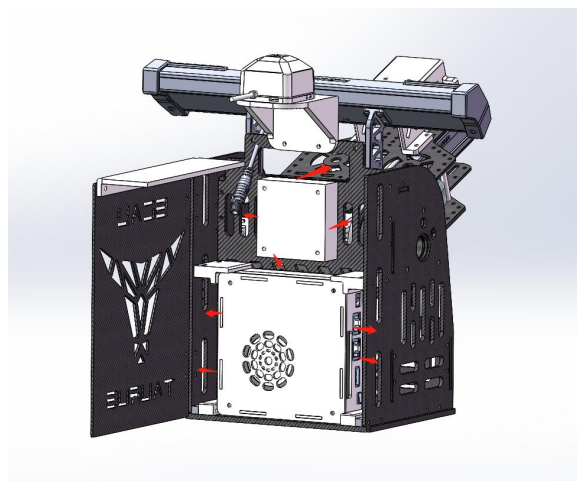


图 1.5.1-6 云台走线示意

1.5.1.6 核心零件的有限元分析、静力学分析

底盘框架核心由四根 20*20 的铝方管通过板材连接四个舵轮轮系，中间两条铝方管侧面承受云台支撑板传递过来的云台重量，而对此我们所做的静力学分析、有限元分析（如图 1.5.1-10、1.5.1-11 所示）。在设计云台架时考虑到弹仓隔离板、轮系连接板、云台架支撑问题，我们决定三个化为一体，充分发挥利用板材，减少额外用材，其中一块云台支撑板（如图 1.5.1-12 所示）

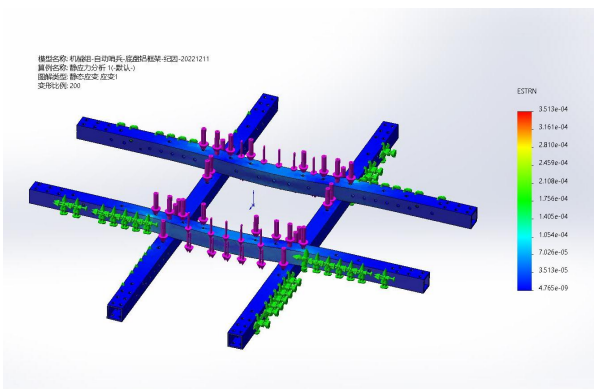


图 1.5.1-10 底盘铝框架静力学分析

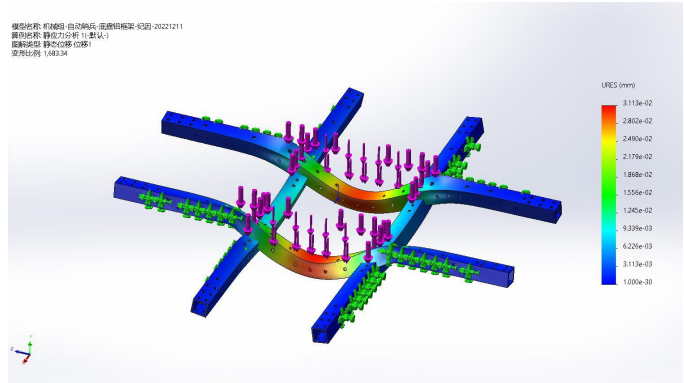


图 1.5.1-11 底盘铝框架变形分析

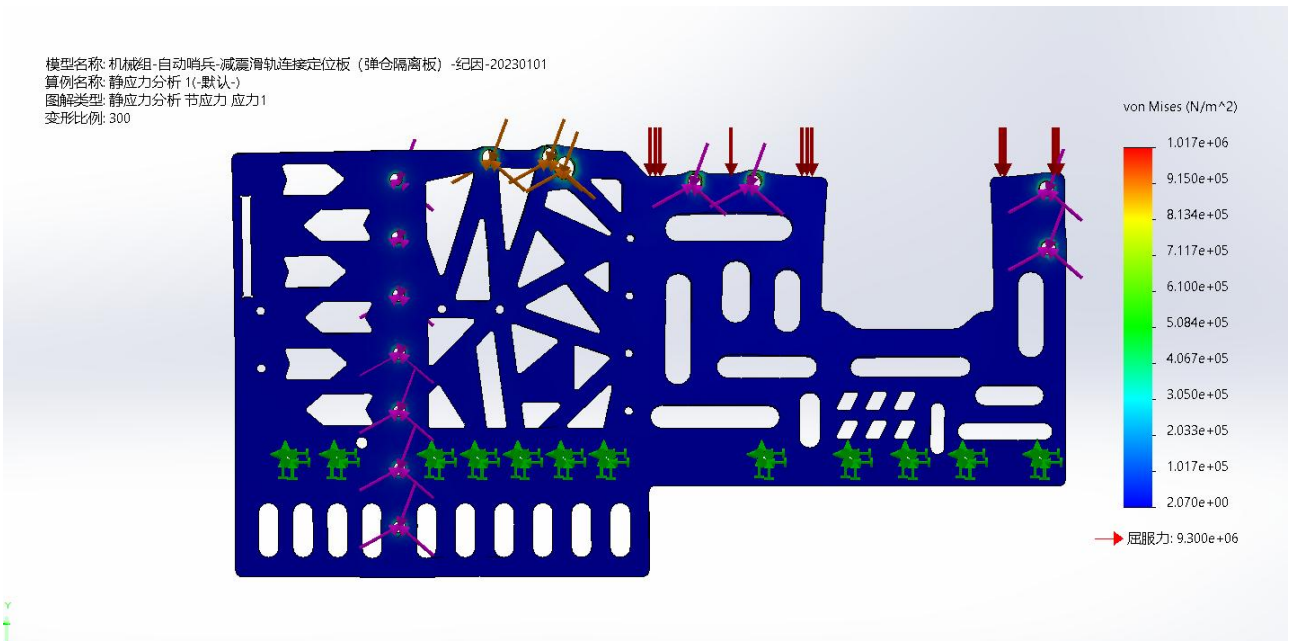


图 1.5.1-12 云台支撑板

1.5.2 嵌入式硬件设计

控制电路板方面，哨兵统一采用“自制核心板 + 接线底板”的方案：相同的核心板，根据不同云台和底盘的需求而定制具有不同接口的底板，兼顾了稳定性与灵活性。针对超级电容，我们自主设计了电容控制板，具有电容状态检测、电控充放电控制以及电容充电保护功能。

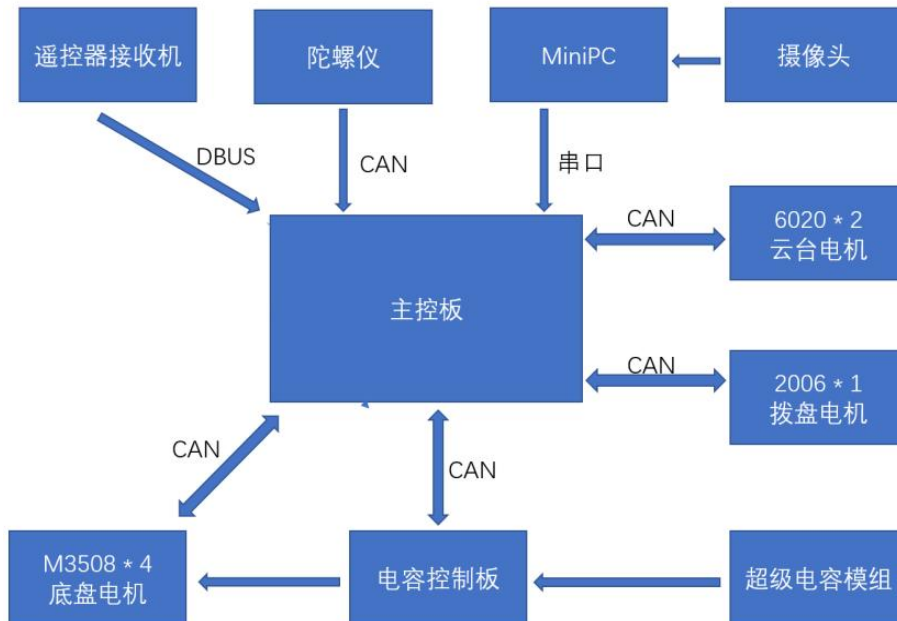


图 1.5.2-1 步兵硬件系统拓扑

1.5.2.1 核心板及底板设计介绍

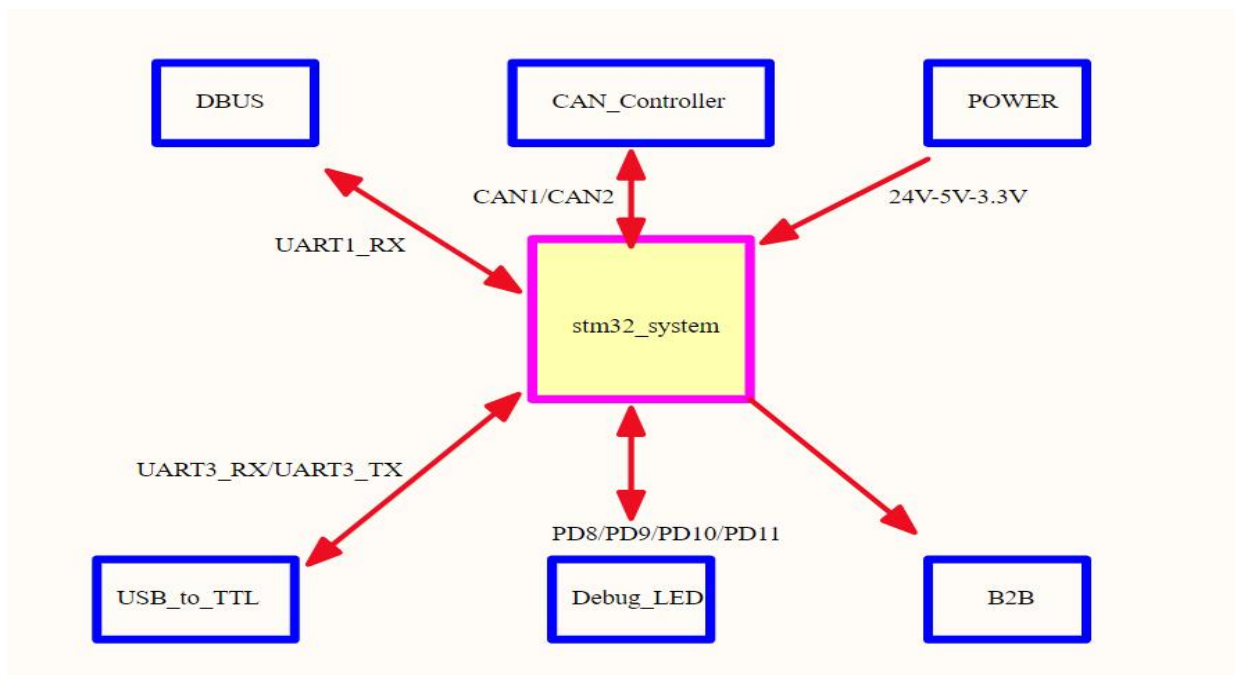


图 1.5.2-2 核心板硬件拓扑图

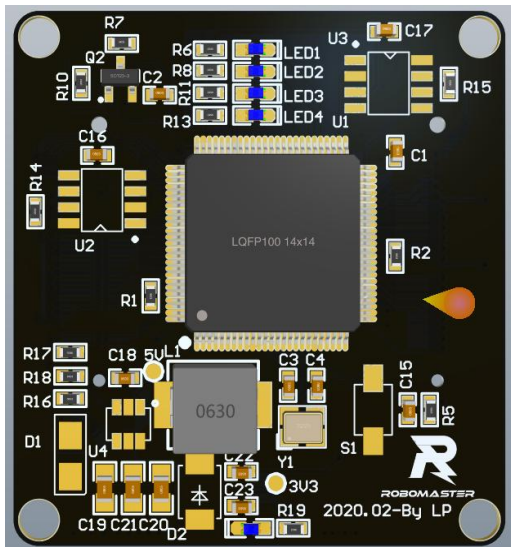


图 1.5.2-3 自制核心板（正面）

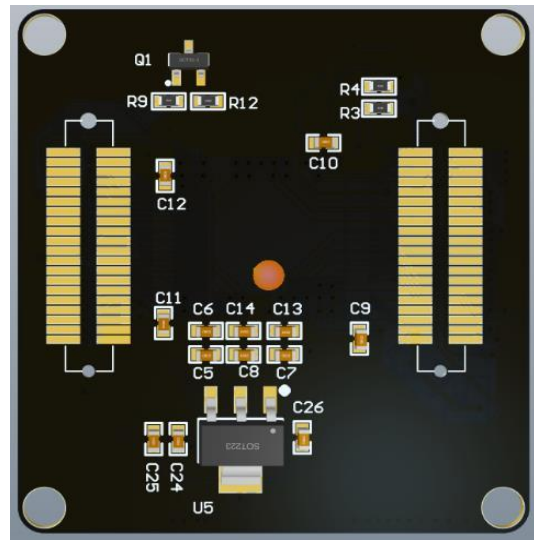


图 1.5.2-4 自制核心板（背面）

主板资源说明	
1	主控为 STM32F407VET6，主频达 168MHz
2	5V1A 稳压电路
3	4 路 pwm 接口
4	1 路 6pin 通信接口（裁判系统串口，底盘 can）
5	3 路串口，1 路虚拟 usb，1 路 SWD
1u16	4 路状态指示灯

表格 1.5-1 主板资源说明

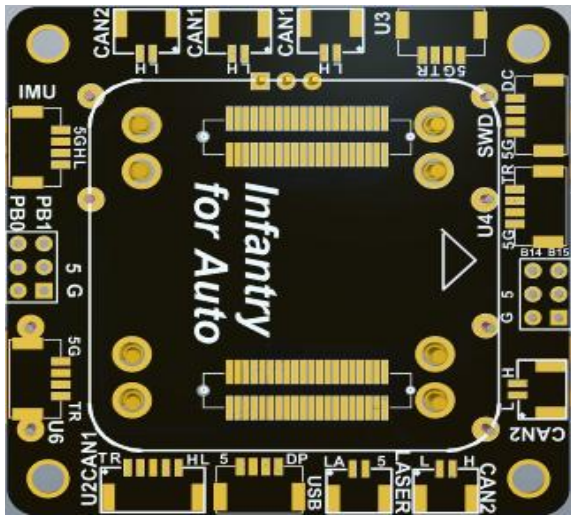


图 1.5.2-5 哨兵上底板（正面）

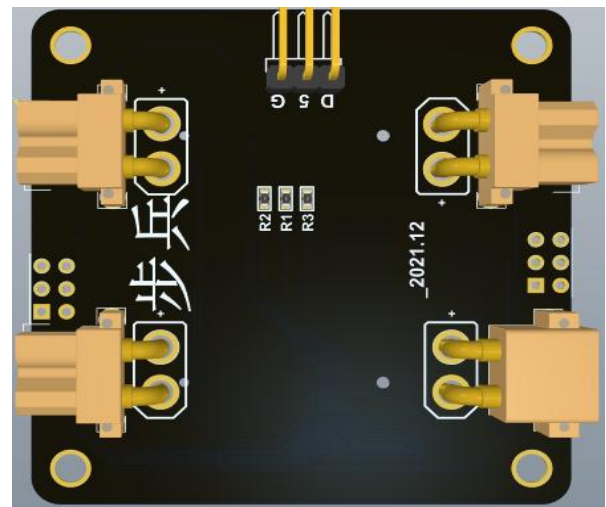


图 1.5.2-6 哨兵上底板（背面）

哨兵上底板接口说明	
1	1 路 6 pin 通信端口
2	1 路 SWD 端口
3	1 路激光控制端口
4	1 路陀螺仪 CAN 端口
5	1 路 DBUS 排针
6	1 路 XT30 电源输入端，3 路 XT30 电源输出端
7	4 路 PWM 排针
8	一路虚拟 USB
9	若干 CAN 端口

表格 1.5-2 哨兵上底板接口说明

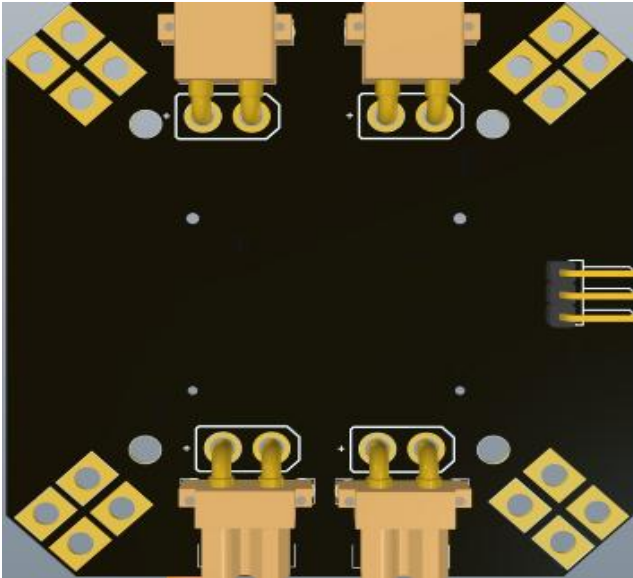


图 1.5.2-8 哨兵下底板（正面）

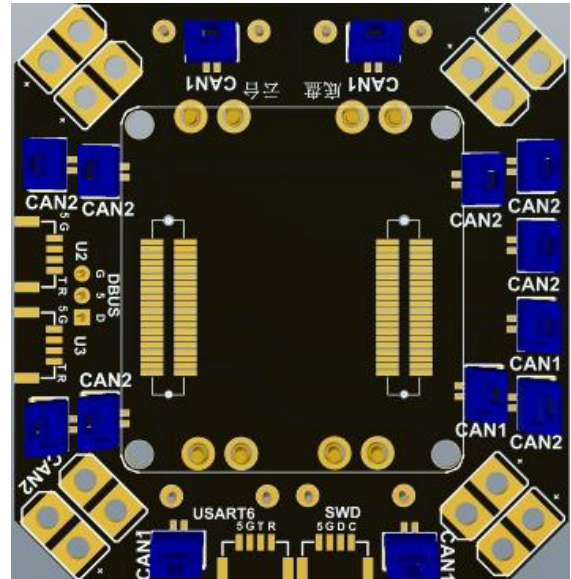


图 1.5.2-7 哨兵下底板（背面）

哨兵下底板接口说明	
1	1 路 SWD 端口
2	1 路 DBUS 排针
3	2 路 XT30 电源输入端，10 路 XT30 电源输出端
4	若干 CAN 端口

表格 1.5-3 哨兵下底板接口说明

1.5.2.2 单板硬件说明

单板	设计需求	风险评估
核心主控板	根据队内统一需求，将 STM32F407 作为主控芯片，并附加 CAN 芯片，引出芯片各引脚，通过板对板连接器进行外接，从而提高核心主控板的拓展性。	中

单板	设计需求	风险评估
自动哨兵底板	基于核心主控板的研发，我们只需要给主控板画相应需求的载板，将核心板上的引脚在底板上以端子的形式引出。	低
分电分 CAN 板	基于电控布线的需求，不可避免的需要对电源的分接和 CAN 的拓展。	低
陀螺仪	根据电控对陀螺仪精度的需求，自研陀螺仪。	低
超级电容模块	基于上一年的电容控制板存在严重的散热问题，2022 赛季硬件组自行设计了新的超级电容模块，主要是为了解决原有散热问题和提高电容效率。	低

表格 1.5-4 单板硬件说明

1.5.2.3 超级电容方案设计

基于上一年的电容控制板存在严重的散热问题，2023 赛季我们决定对电容控制板的充电芯片进行改进，选择了一种最新的国产芯片，此芯片同时有升降压的拓扑功能，电压电流等各方面也符合我们的要最终决定在此芯片上开始了 2023 赛季超级电容控制板的研发，研发周期两个月。下面将对其中的部分细节作介绍。

板载供电系统

首先是板载供电系统

两个最主要的部分：

24V 转 5V 降压电路

5V 转 3.3V 降压电路

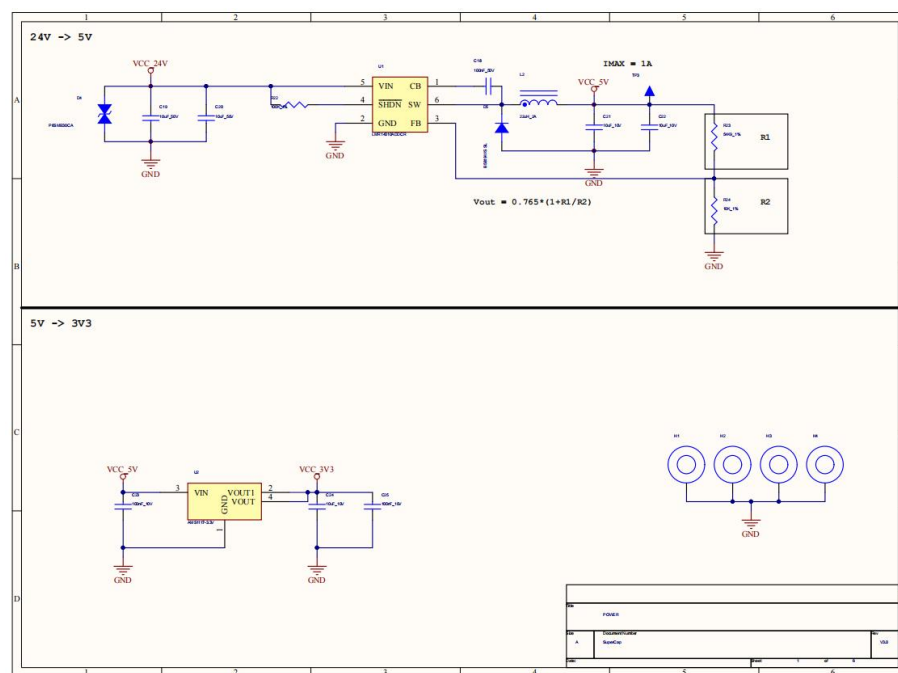


图 1.5.2-9 降压电路原理图

硬件充电芯片外围电路

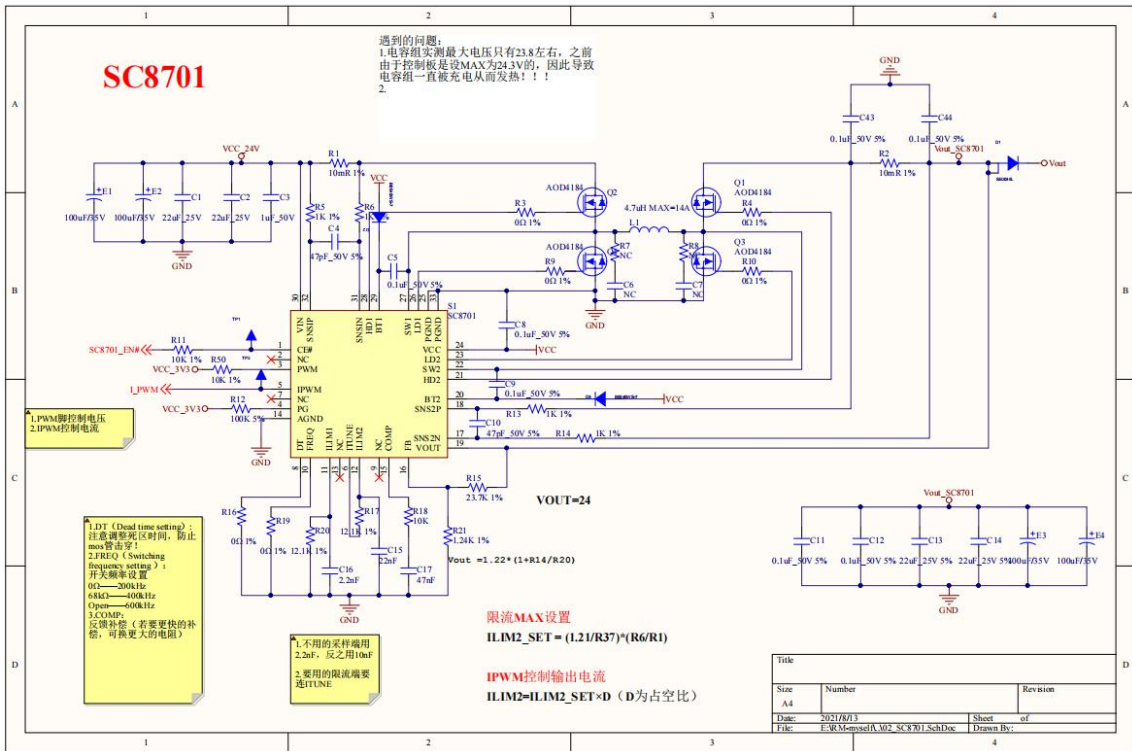


图 1.5.2- 11 电容硬件充放原理图

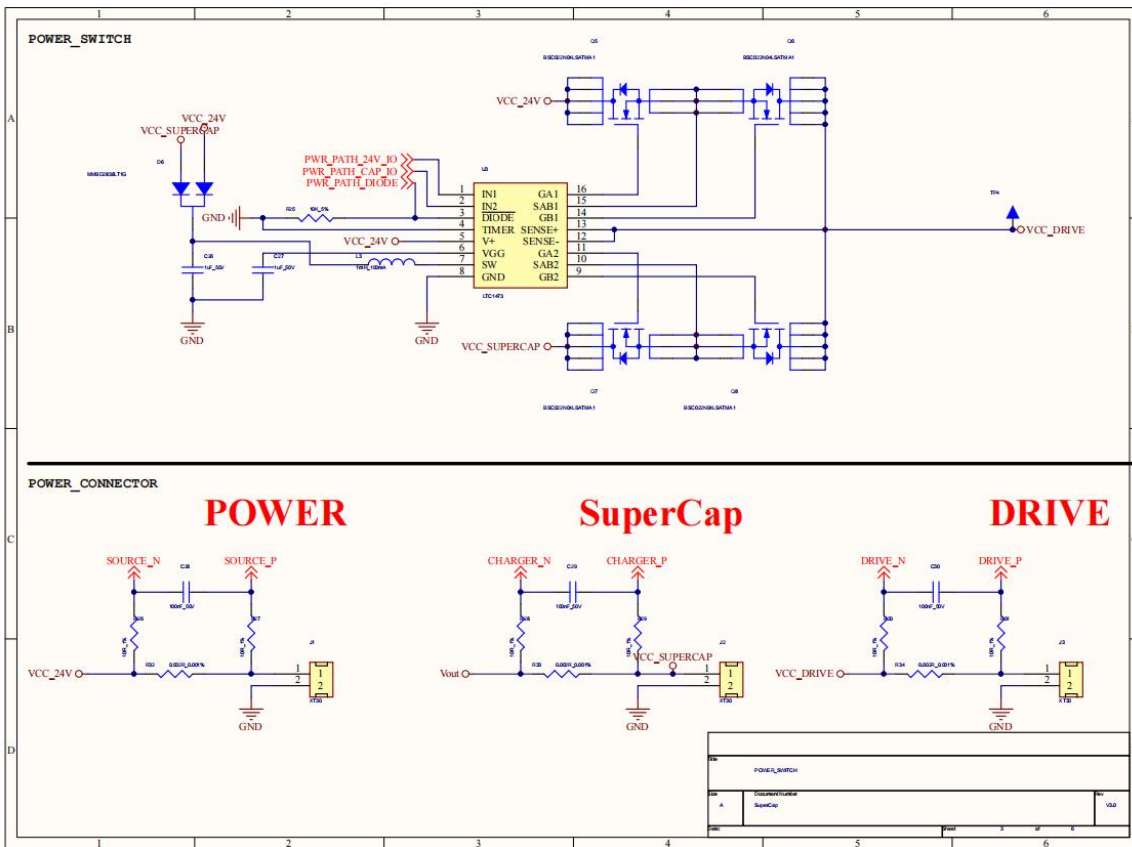


图 1.5.2- 10 MOS 管控制电路开关以及电流采样电路

主控 STM32F103RCT6 外围电路

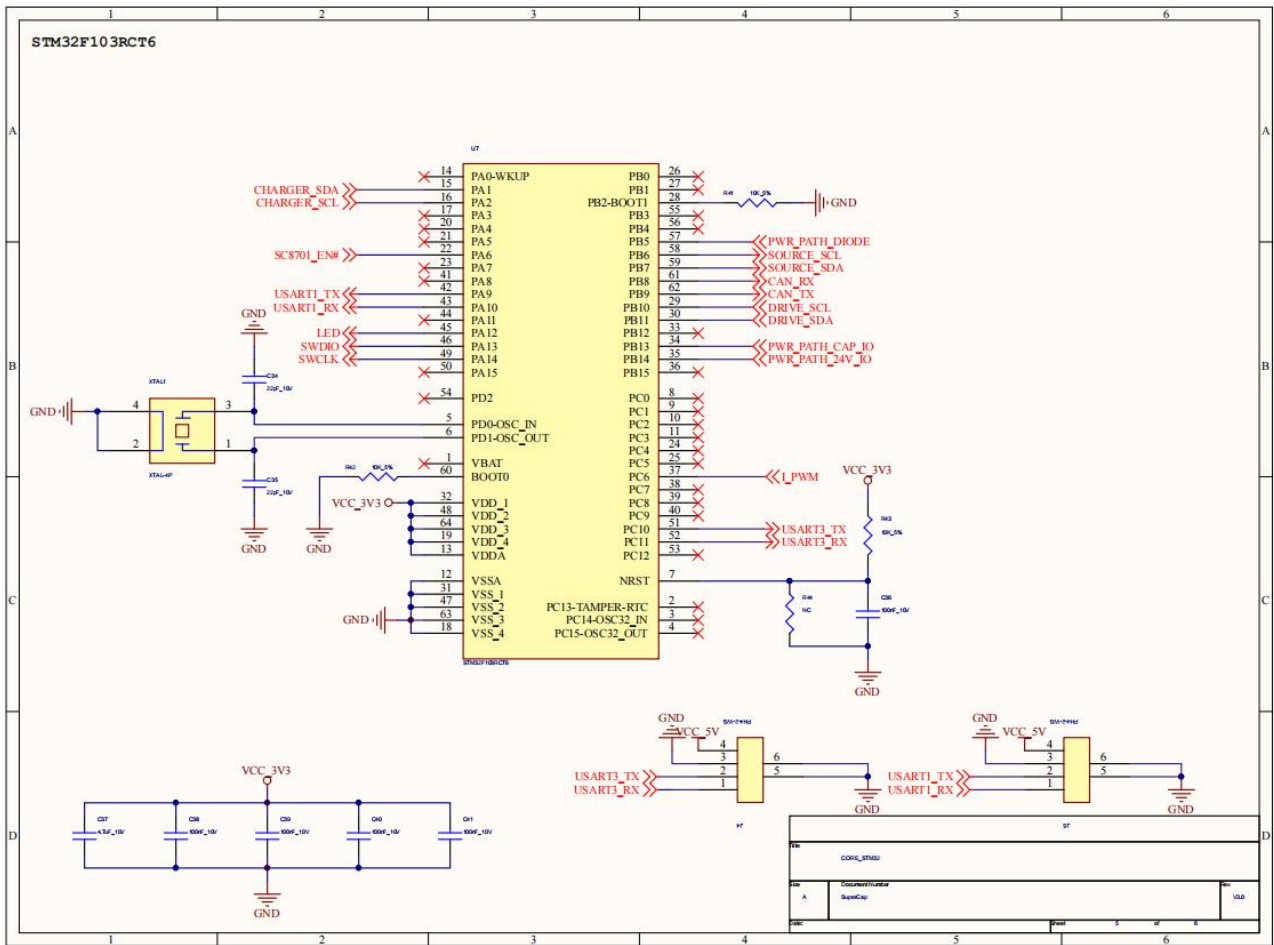


图 1.5.2-12 主控芯片外围电路原理图

电容组参数及相关计算，我们选用的是 9 个 2.7V 60F 的电容器：总电压为 $24.3V = 2.7V * 9$ ；

充放电硬件策略

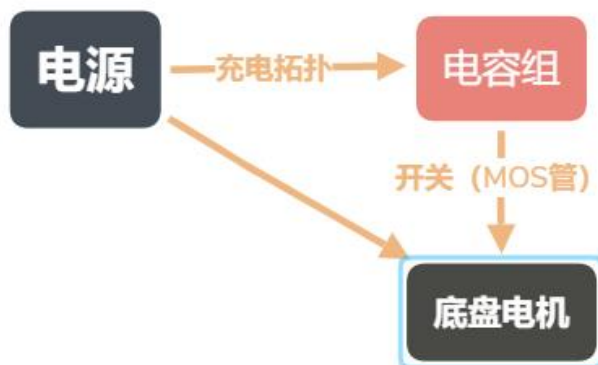


图 1.5.2-13 充放电硬件框图

1. **充电拓扑**：选用国产芯片 SC8701（此芯片支持最大 10A 电流对电容组进行充电，同时具有过压保护和过热保护，电压接近顶峰可进行恒流充电）。
2. **放电策略**：利用 MOS 管特性控制电容组与电机的单向通路。
3. **实际效果**：在实际利用上，电容组最低可放电到 12V。电容利用率达到 80%左右；电容控制板有功率控制作用，并未出现超功率的现象。

充放电软件控制

在实际超级电容的使用中，除去超级电容控制板自身的硬件和软件的驱动，还需要主控端配合整台机器底盘行为进行控制。程序流程图如下：

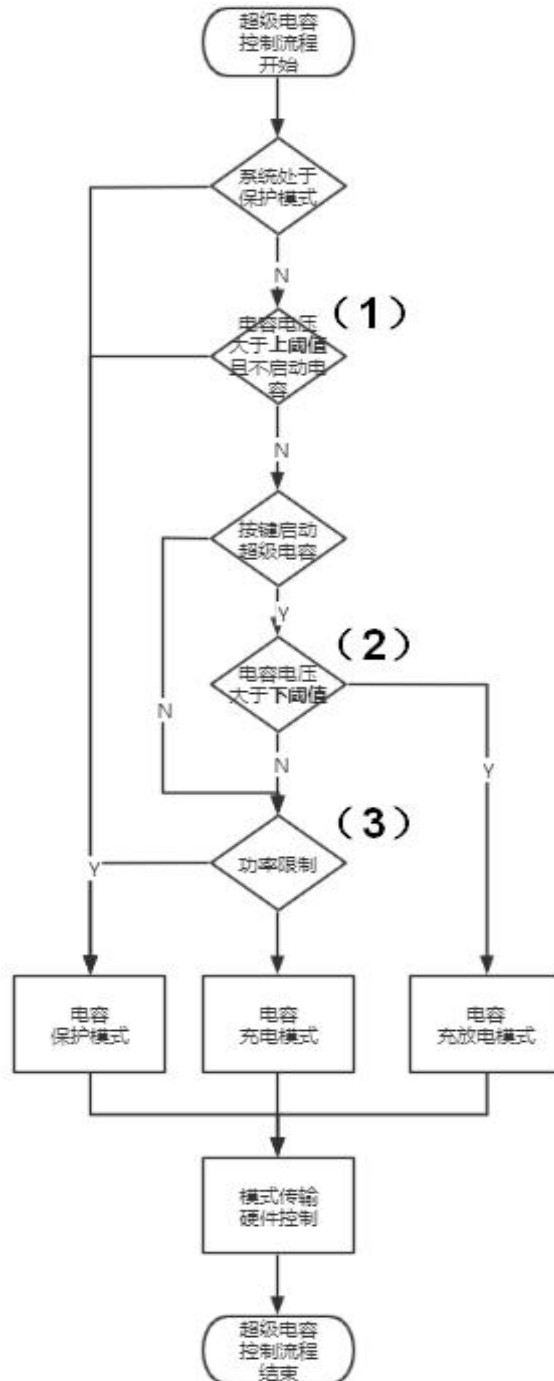


图 1.5.2- 14 超级电容软件控制流程

其中，条件（1）和（2）是软件层面对于电容组进行保护，防止过冲过放；条件（3）是为了配合功率控制功能：当底盘正处在功率控制状态，说明底盘已经超功率，在消耗缓冲能量，此时停止给电容组充电；反之，说明底盘功率有剩余，则用于给电容充电以实现功率使

用率最大化。

1.5.3 嵌入式软件设计

1.5.3.1 系统架构

哨兵程序使用 FreeRTOS 系统，采用任务调度的控制逻辑，主要设定底盘任务，云台任务，发射器任务来控制整台步兵的运行状态。整个系统中，共有系统模式切换任务、通信状态检测任务、底盘任务、云台任务、发射器任务、视觉串口发送任务、CAN 发送任务和裁判系统通信任务。

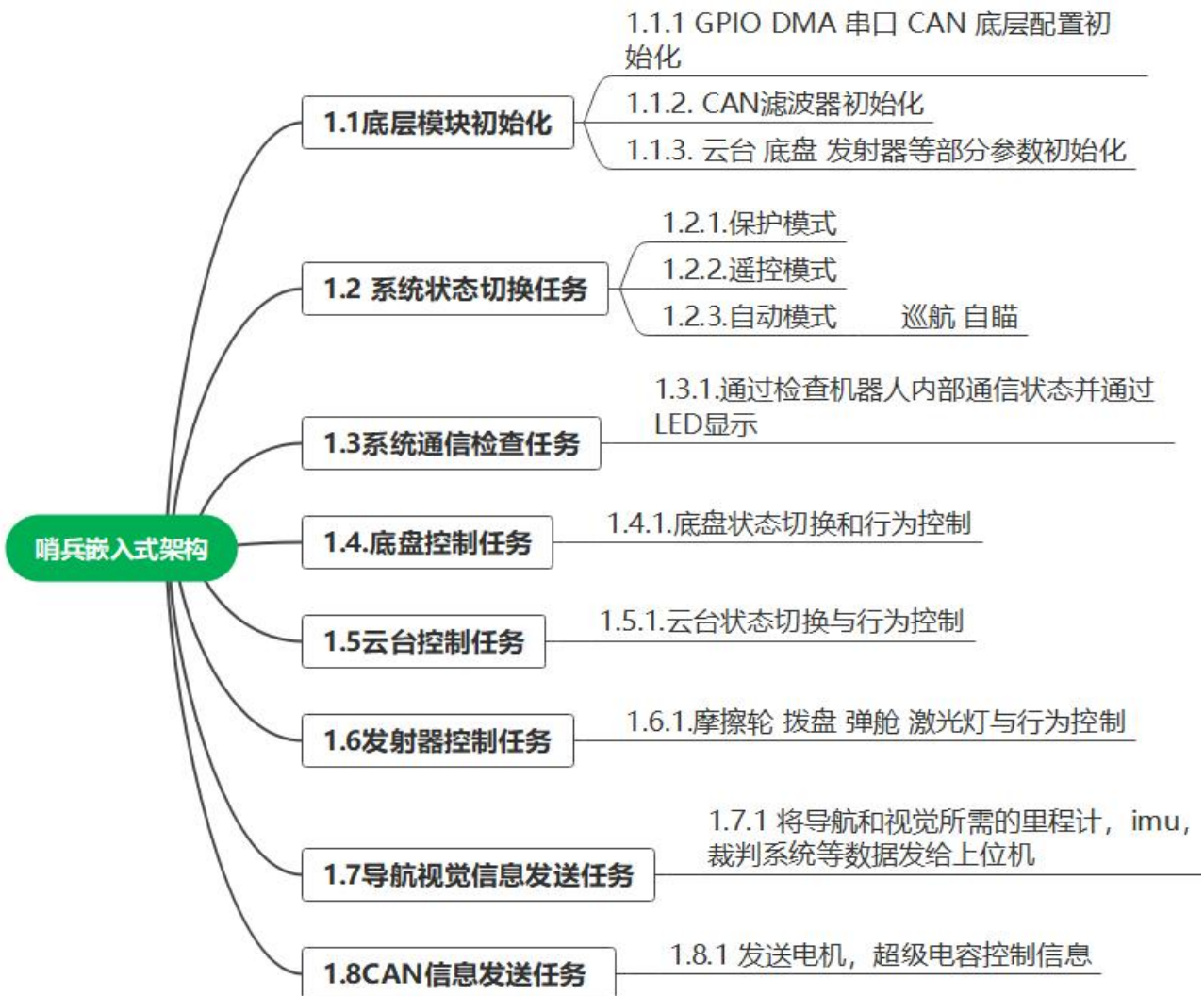


图 1.5.3-1 步兵机器人软件框架

1.5.3.2 系统状态机

遥控模式切换

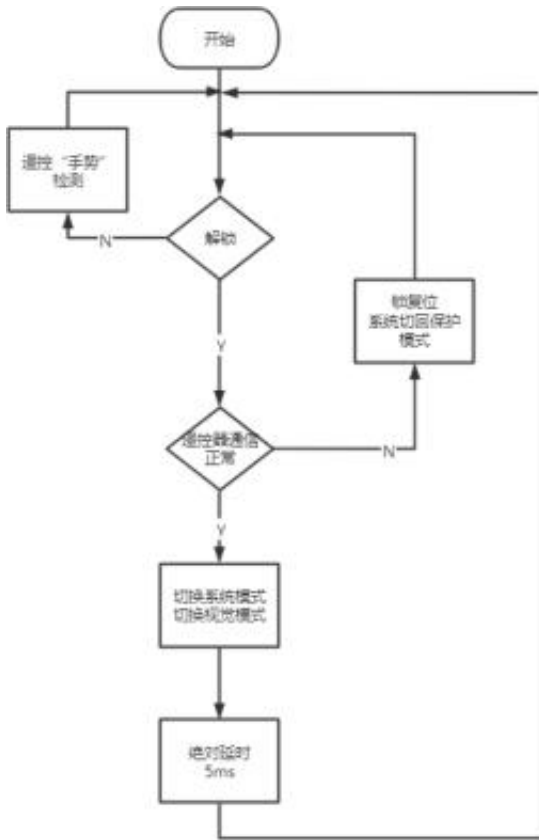


图 1.5.3-2 系统模式切换图

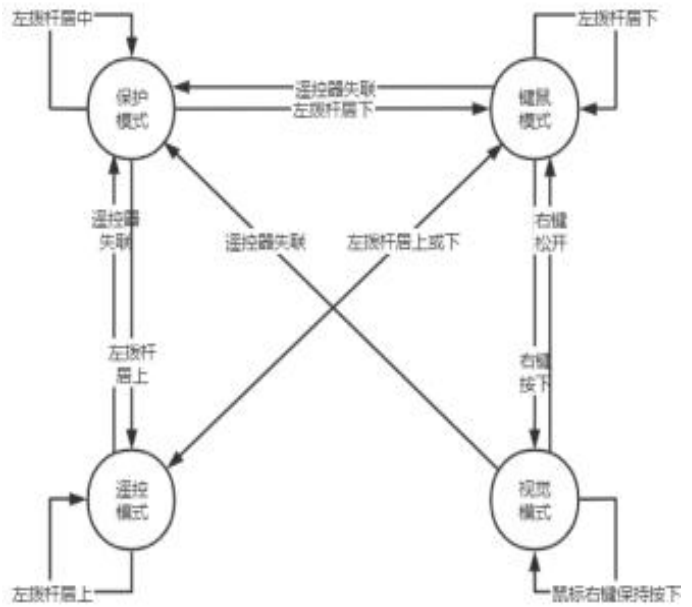


图 1.5.3-3 系统状态切换图

底盘模式切换

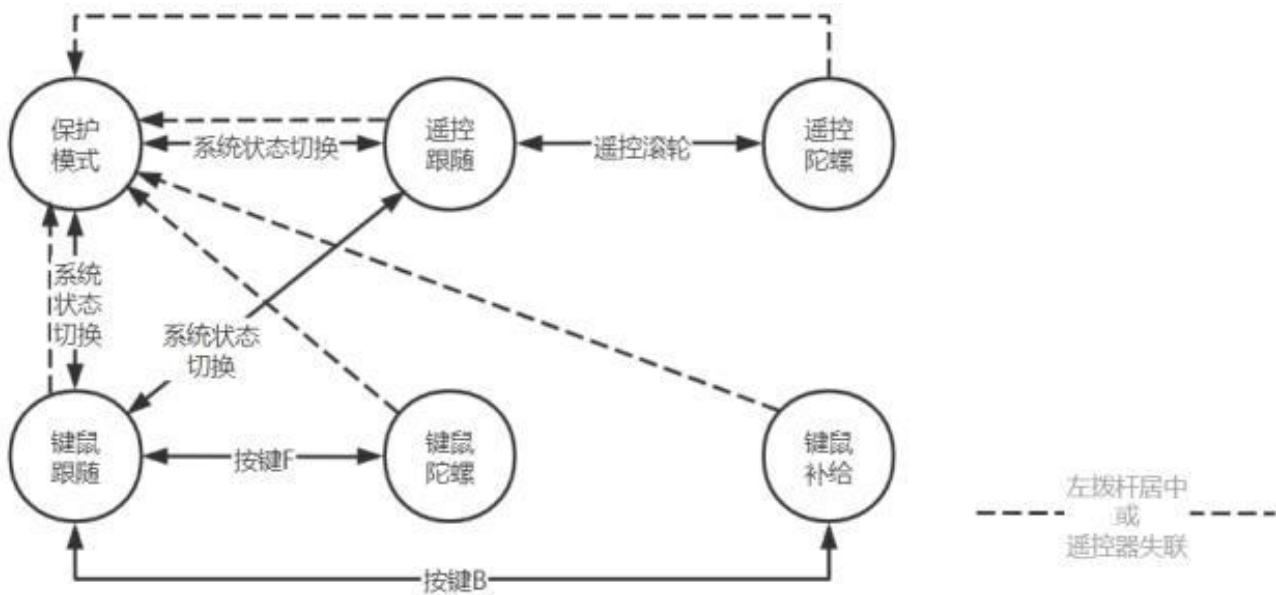


图 1.5.3-4 底盘状态图

底盘模式	底盘行为
保护模式	底盘电机电流置 0，底盘无力
遥控跟随模式	底盘跟随云台运动
遥控陀螺模式	底盘开启小陀螺
自动模式	底盘交给上位机控制

表格 1.5-5 底盘模式及行为

云台状态机

云台状态在跟普通步兵相同的情况下，哨兵新增加了一个自动模式，将云台的所有权全交给上位机操控。

云台模式	云台行为
保护模式	云台电机电流置 0，云台无力
遥控模式	云台由遥控器遥感控制
自动模式	云台交给上位机控制

表格 1.5-6 云台模式及行为

发射器状态机

发射器状态主要分为：摩擦轮状态，拨盘状态，弹舱盖状态三部分

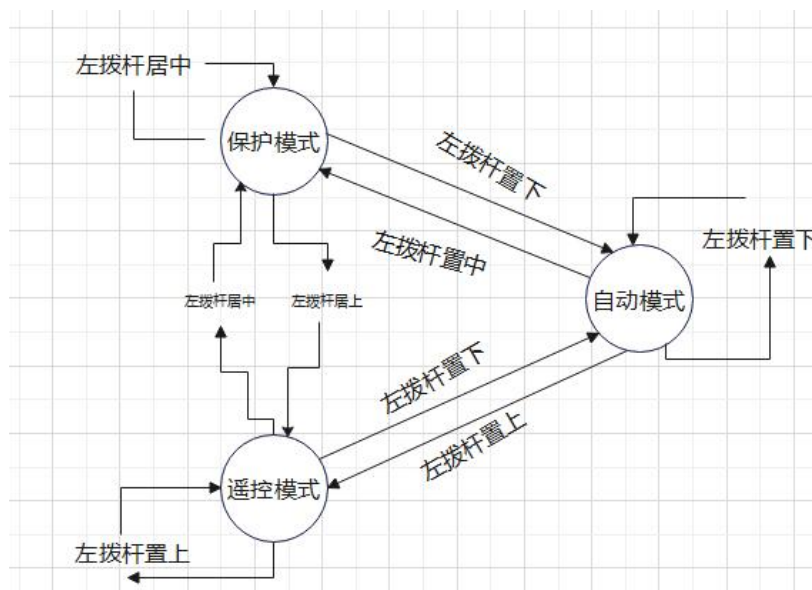


图 1.5.3-5 摩擦轮与拨盘状态控制

拨盘模式	拨盘行为
保护模式	拨盘电机电流置 0，拨盘无力
(遥控) 手动单发模式	遥控器一拨一发
(遥控) 自动单发模式	遥控右拨杆居下，先连发后热量控制单发
(遥控) 自动连发模式	遥控右拨杆居下，无热量控制
(自动) 单发模式	由上位机控制，有热量和射频控制
(自动) 连发模式	由上位机控制，有热量和射频控制

表格 1.5-7 拨盘模式及行为

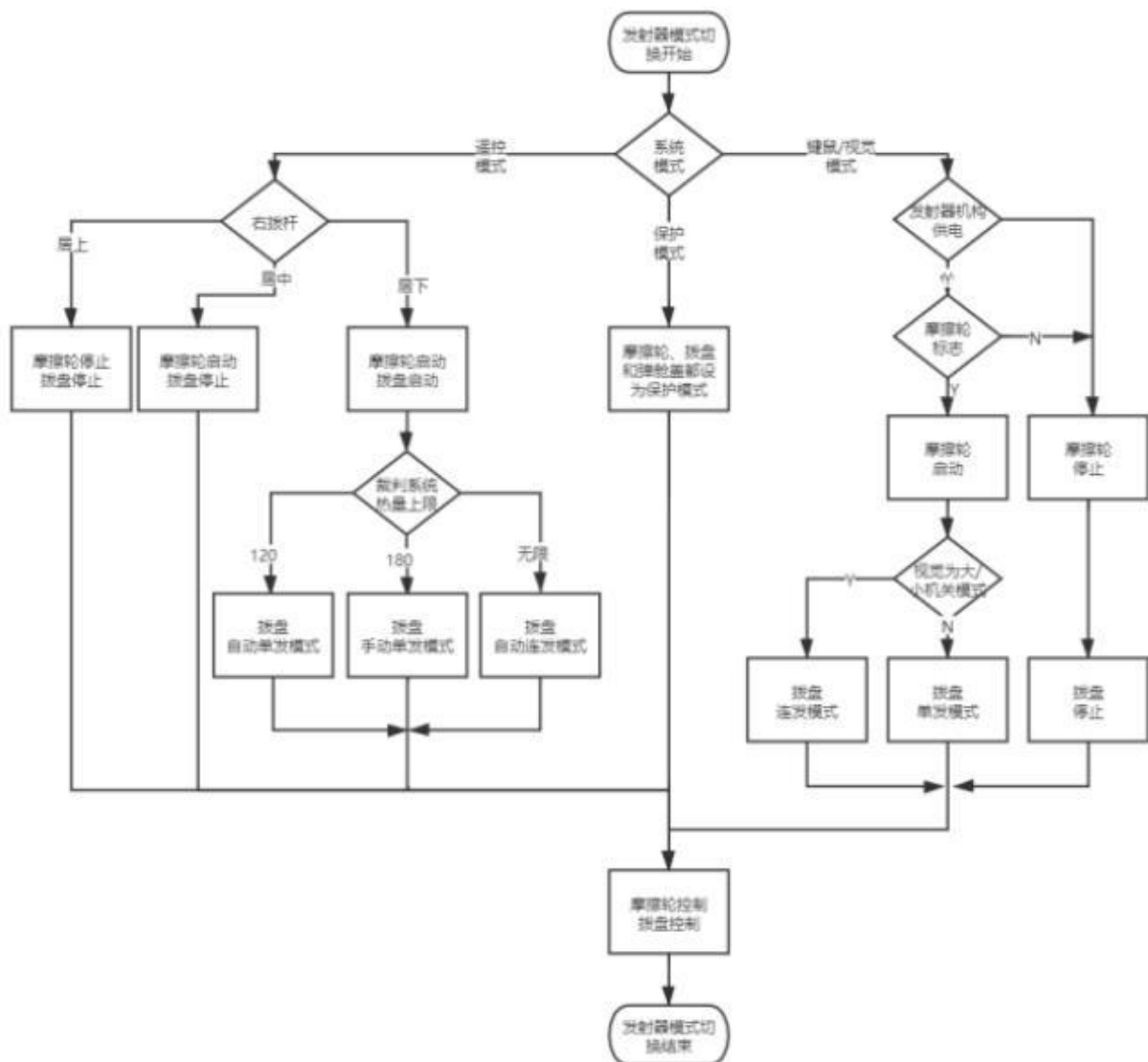


图 1.5.3-6 发射器模式切换

1.5.3.3 通信检测任务

通信检测任务主要是定时检测各个电机、超级电容、陀螺仪和遥控器的通信情况并改变标志位，最终通过指示灯来反馈，用于在场上快速排查问题。

1.5.3.4 小电脑虚拟 usb 通讯

上位机通讯任务主要是将当前机器云台俯仰角，视觉模式，裁判系统信息，里程计等数据打包，通过虚拟 usb 发送给小电脑，并接收小电脑发送的云台和底盘控制指令，经过测试速率能够达到 500hz。

1.5.3.5 CAN 信息发送任务

底盘任务、云台任务、发射器任务执行完后唤醒 CAN 发送标志，CAN 信息发送任务在 CAN 发送标志唤醒后执行，分别通过相应的 ID 控制底盘电机、云台电机和发射器模块。

1.5.3.6 裁判系统信息发送任务

裁判系统信息发送用于发送给云台手自动步兵此时的状态信息以及接收云台手的指令。

1.5.3.7 重点功能

为更加好的完成哨兵在场上的导航定位，自主击打等功能，我们自主编写了底盘功率控制、自瞄以及舵轮里程计等算法。

底盘功率控制算法

功率控制是保证场上除工程机器人和空中机器人以外其他所有地面机器人正常运动的重要因素，仅依靠裁判系统 UI 与操作手人工控制是难以实现的。对此，经过多种方案的测试，最终采用电压不变，限制电流的思路，实现底盘功率控制。

算法流程如图 1.5.3-7 所示。

算法结构简单清晰，效果明显。其中 $\kappa = \kappa_{\rho} * (BufferEnergyMin - BufferEnergy)$ 功率限制系数。在底盘功率调试时，只需要调节 κ_{ρ} 和 $BufferEnergyMin$ 两个超参。 $(BufferEnergy_{pMin} - BufferEnergy)$ 表征功率控制强度， κ_{ρ} 表征功率控制强度的斜率。

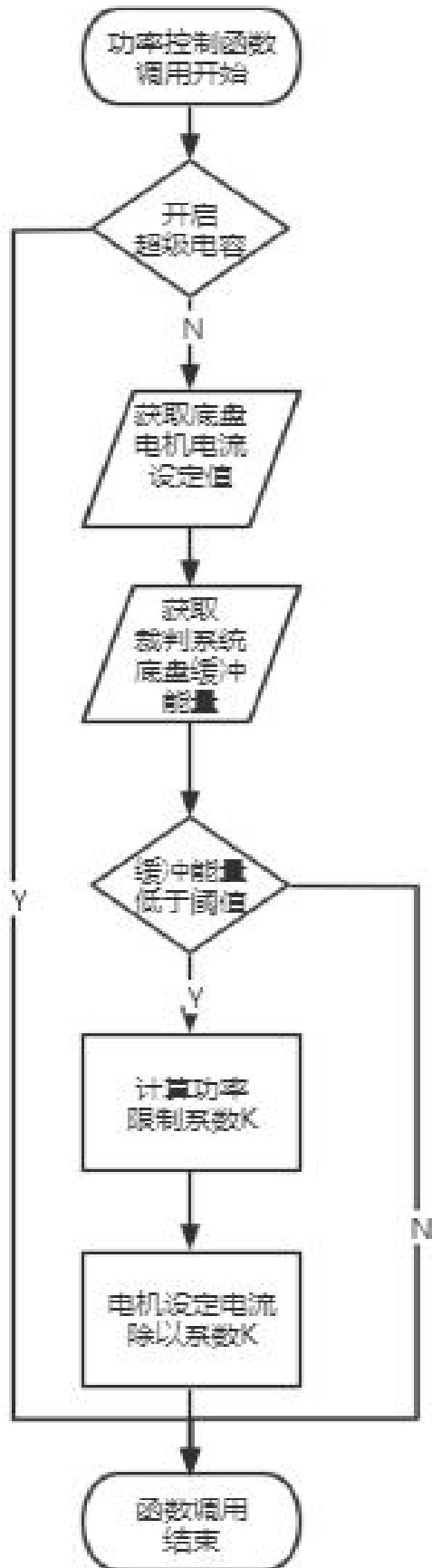


图 1.5.3-8 底盘功率控制算法流程图

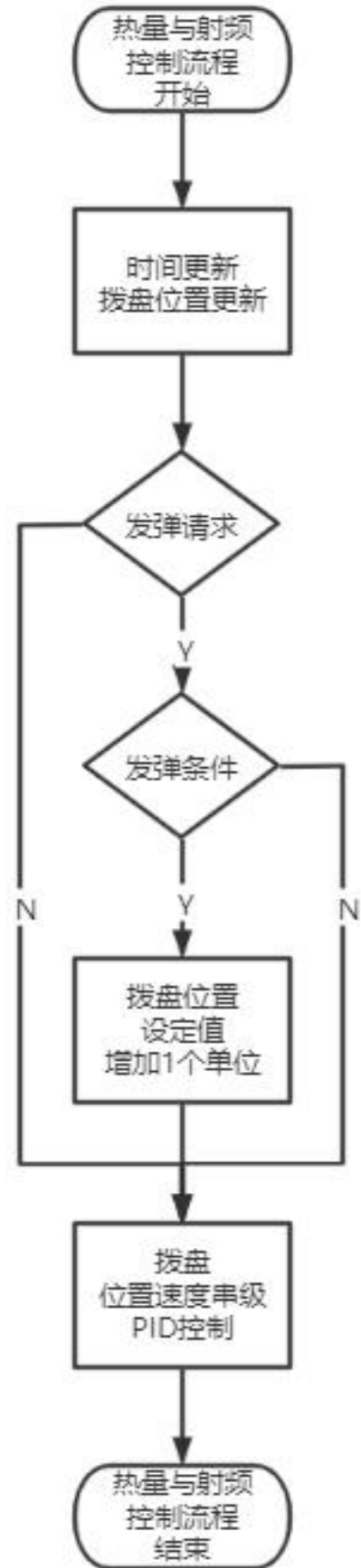


图 1.5.3-7 拨盘热量与射频控制算法流程图

发射机构热量与射频控制算法

类似于底盘功率控制算法，发射机构热量与射频控制算法能够优化操作手的实际体验，直接提高了机器在复杂赛场环境中的生存能力，并能最大化火力输出。

算法的实现前提是发射器拨盘部分，采用位置闭环控制，能够精确到单颗弹丸的控制。热量控制的实现是根据自行计算的枪管热量与裁判系统反馈的枪管最大热量，限制拨盘的转动，同时，考虑到实际缺弹漏弹的情况，需要预留部分热量余量。对于频率控制方面，由于任务周期固定，为实现对发射弹丸的频率实现精确控制，只需要定时使能拨盘，在满足热量控制的条件下，经过鼠标右键或遥控器控制发弹，即可使得发射器以固定频率发射弹丸。经拨盘电控调试，当前射频设置为 11.1Hz。

其中，发弹条件共有 3 点：

- 距离上一次发弹时间间隔大于发弹设定周期；
- 当前枪管热量大于热量设定下限；
- 当前拨盘位置偏差小于 0.5 倍单颗弹丸位置偏差常数。

第一、二点分别对应于射频与热量控制部分，第三点主要用于规避的是拨盘电机响应误差，累积致使控制滞后的问题

舵轮里程计算法

为了给导航更加精准的定位，我们通过 4 个舵的 6020 以及 4 个 3508 的编码器值，解算出此时底盘 x, y, z 的速度，单位为 m/s，经过测试里程计有较高的精准度能够完成导航所需。

$$V_x = (V_{bl} * \cos\theta_{bl} + V_{br} * \cos\theta_{br} + V_{fl} * \cos\theta_{fl} + V_{fr} * \cos\theta_{fr}) * 1/4$$

$$V_y = (V_{bl} * \sin\theta_{bl} + V_{br} * \sin\theta_{br} + V_{fl} * \sin\theta_{fl} + V_{fr} * \sin\theta_{fr}) * 1/4$$

$$V_z = (-V_{bl} * \cos\theta_{bl} + V_{br} * \cos\theta_{br} - V_{fl} * \cos\theta_{fl} + V_{fr} * \cos\theta_{fr}) * 1/4 * \frac{1}{offseta} + (-V_{bl} * \sin\theta_{bl} - V_{br} * \sin\theta_{br} + V_{fl} * \sin\theta_{fl} + V_{fr} * \sin\theta_{fr}) * 1/4 * \frac{1}{offsetb} * 1/2$$

其中 $\frac{1}{offset}$ 表示由舵的中心到底盘中心的 x, y 轴距离

V 表示四个 3508 的速度， θ 表示 4 个舵的角度

陀螺仪软件设计

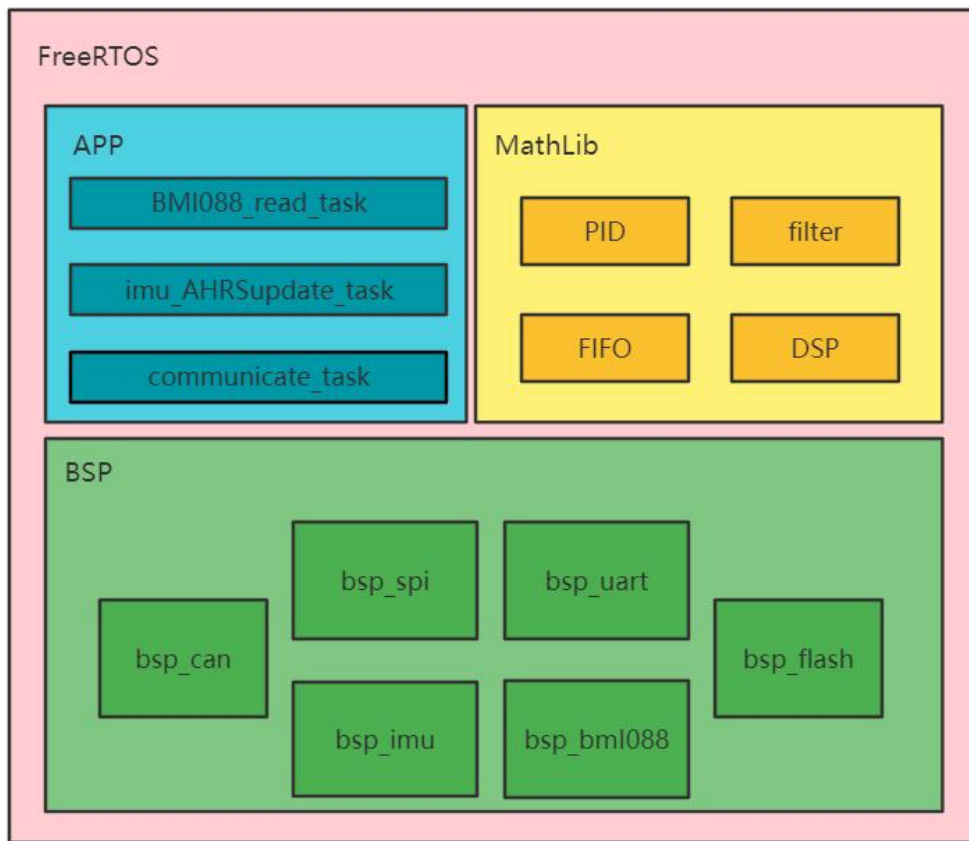


图 1.5.3- 11 陀螺仪软件架构

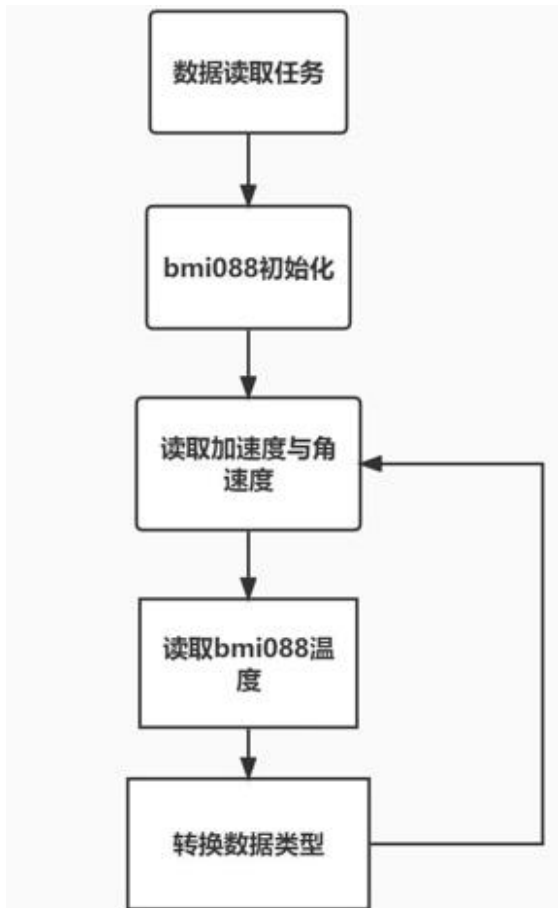


图 1.5.3- 10 陀螺仪数据读取任务

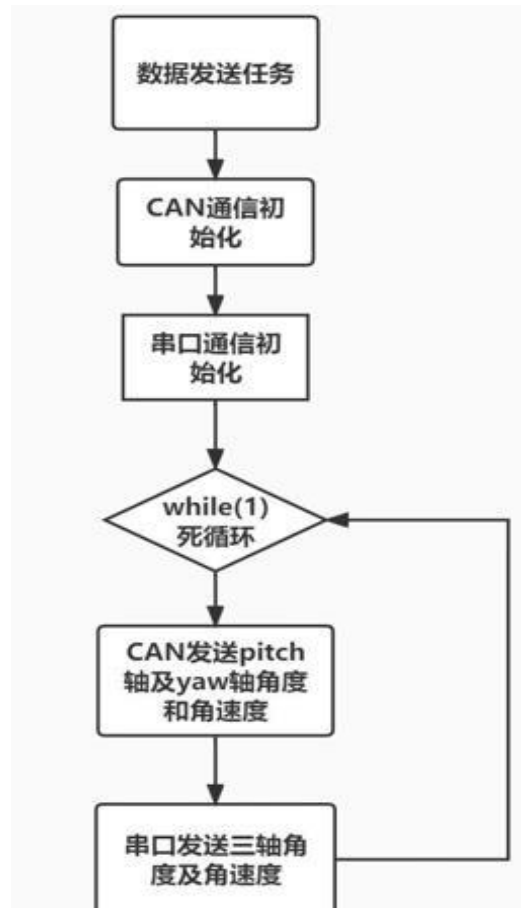


图 1.5.3- 9 陀螺仪数据发送任务

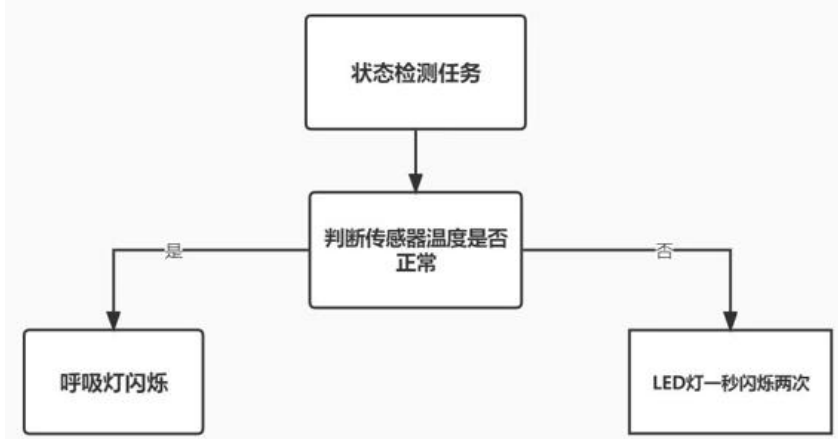


图 1.5.3- 13 陀螺仪状态检测任务

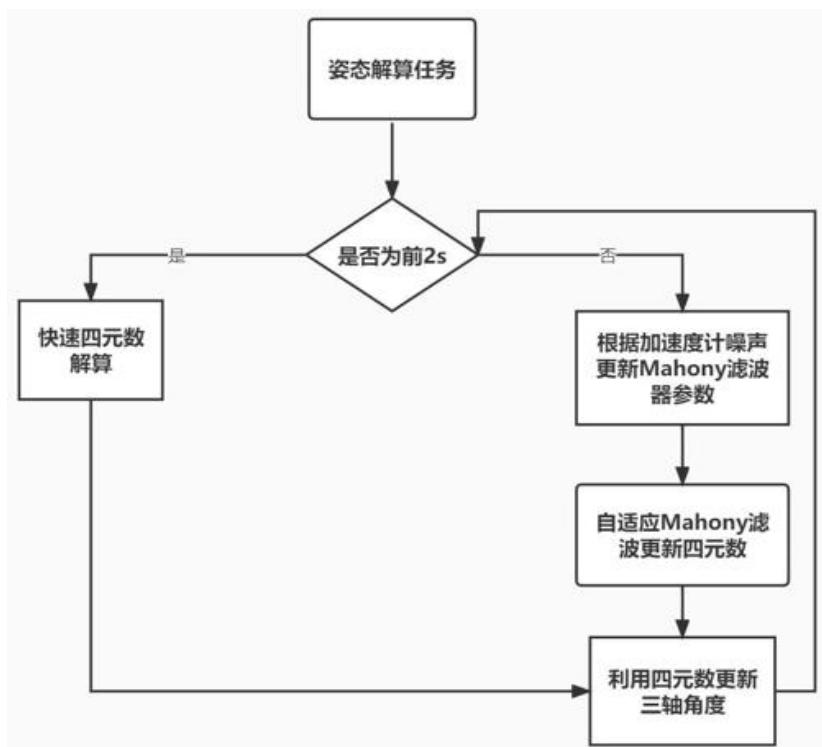


图 1.5.3- 12 姿态解算任务

1.5.4 视觉算法设计

我们使用 Ubuntu20.04、vscode 和 opencv3.4.12 作为开发环境。

1.5.4.1 运行流程

1. 预处理

我们对低曝光图像进行颜色通道相减得到灰度图，调节合适的颜色阈值参数和亮度阈值参数，进行二值化和形态学等预处理操作，同时融合图像中装甲板的颜色信息和亮度信息，得到了较好的二值化图像。

2. 装甲板识别

二值化图像中，我们使用最小包围矩形拟合灯条，初步筛选出合适的灯条。再使用灯条角度差排序，加入长宽比、角度差、中心高度差、长度差等筛选条件，灯条两两匹配筛选出装甲板对象。决策上以同一装甲板（最接近 roi 中心）、同一数字（同一装甲板可能不同的装甲板）依次为优先进行打分，选择出最佳装甲板的信息。

3. 角度解算

上一步可以得到装甲板灯条上合适的四个点，通过使用 PNP（Perspective-n-Point）解算方法对这四个点进行解算，从而计算出相机与目标装甲板之间的空间位置关系。这是一个相对的位置关系，然后根据陀螺仪记录下来的角度，通过矩阵运算，将相对的相机坐标系下的目标装甲板坐标点转成世界坐标系下的坐标点。对该装甲板坐标进行重力补偿及预测后，最终得到一个绝对的云台需要转动的 pitch 及 yaw 轴角度。

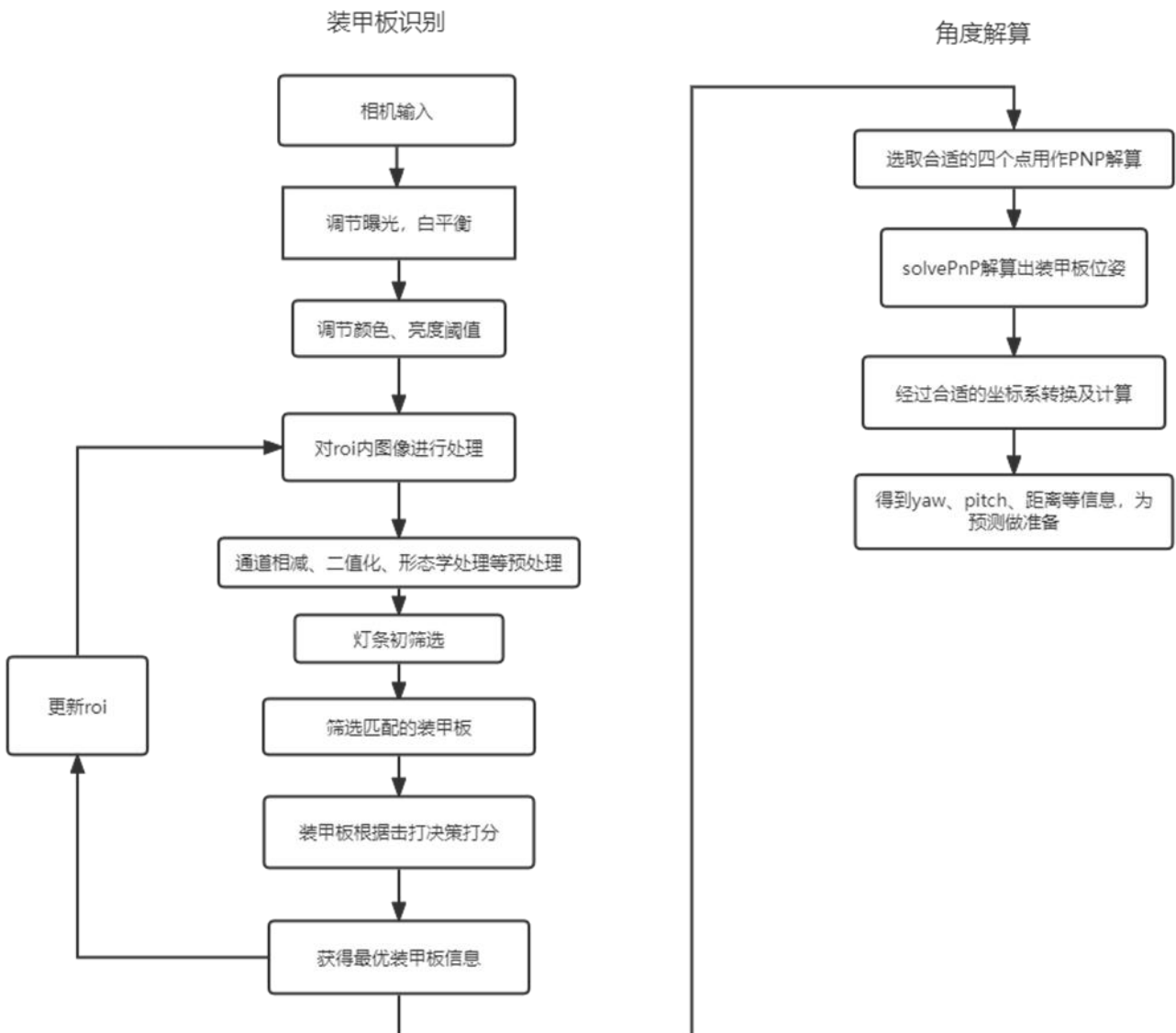


图 1.5.4-1 识别计算流程图

1.5.4.2 重点功能

自瞄实现

在实际击打中，击打中装甲板，装甲板会进行闪烁，不可避免地会丢帧。利用拉格朗日插值法等进行合理地补帧插值，来配合预测器的运行。

由陀螺仪得到云台 pitch 轴以及 yaw 轴的转角经电控发送至视觉电脑。两者的数据进行合适的拟合及对齐时间戳，得到目标准确的绝对角度及角速度等信息。对世界坐标系下的目标坐标点，我们运用二阶卡尔曼，考虑相应的时间延迟（子弹飞行、通信延迟、发射延迟等），实现对目标位置的预测。对于 pitch 轴的转角，需要经过重力补偿，计算得到抬升合适角度的发射角。最后的数据转换为角度发送给电控进行控制。

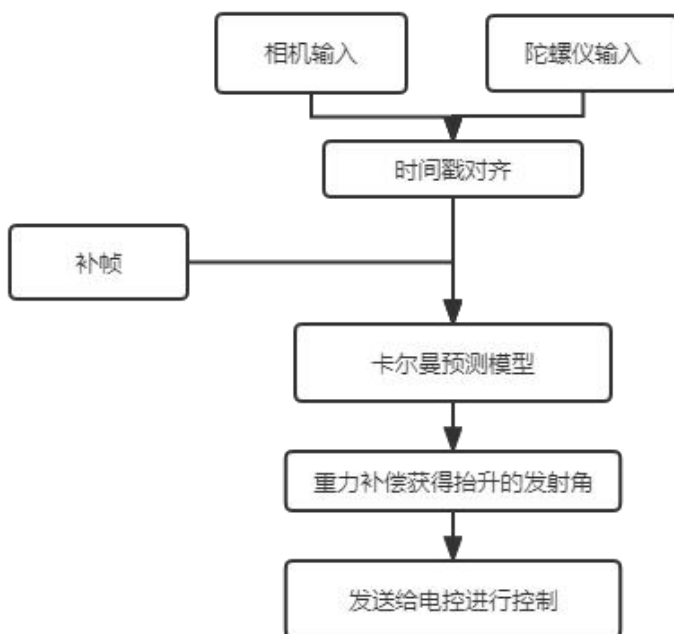


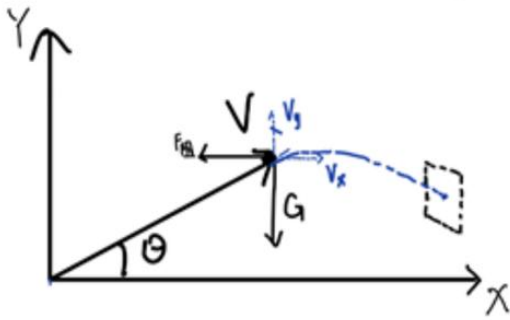
图 1.5.4-3 预测流程图



图 1.5.4-2 自瞄效果图

重力补偿

在基础的瞄准方案上，因为弹丸受重力以及空气阻力的影响，且今年规则规定弹丸速度，因此算法考虑重力以及空气阻力的因素，进行重力补偿修正。本算法使用目标装甲板在以枪管口为原点的坐标系中的坐标(摄像头与枪管同轴且固定在一起)，用枪管的俯仰角和弹丸初速来计算重力补偿。使用的弹道模型考虑了水平方向的空气阻力，因此在水平方向上为变加速运动，且阻力系数由实际实验情况调试得出，对速度进行微分，计算出重力补偿后正确俯仰角与当前枪管俯仰角的差值以进行重力补偿修正。



重力下坠模型

注：其中 Q 为空气阻力系数，V₀ 为子弹速度，x 为水平距离，y 为竖直距离。

$$F = ma = -Qv^2; \quad m \frac{dv}{dt} = -kv^2$$

$$\frac{dv}{v^2} = -\frac{k}{m} dt; \quad -v^{-1} - \frac{1}{v_0} = -\frac{Q}{m} t$$

$$v = \frac{v_0 m}{m + v_0 k t}; \quad k = -\frac{Q}{m}$$

$$A = \frac{(e^{kx} - 1)^2}{2k^2 v_0^2}; \quad B = -\frac{e^{kx} - 1}{k}$$

$$C = A + y; \quad \text{ang} = \frac{(-b) - \sqrt{b^2 - 4ac}}{2a}$$

$$\text{pitch} = \arctan(\text{ang}) * 57.2957805$$

反小陀螺、反扭腰

图 1.5.4-4 重力补偿计算

观测敌方陀螺后的数据图如下，由于每次只能观测到一个面的装甲板，因此会产生时间为 T 的锯齿状的波形，可以发现有明显的周期性，无论是从旋转向量方向考虑还是仅仅从向量大小考虑都可以完成对陀螺数据的拟合。我们实现算法前一旦发现拟合上需要关闭插帧操作以防止数据遭受插帧影响。

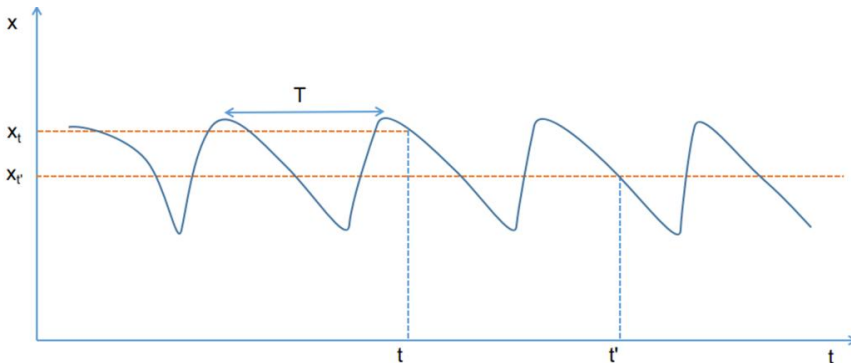


图 1.5.4-6 X 轴差分曲线

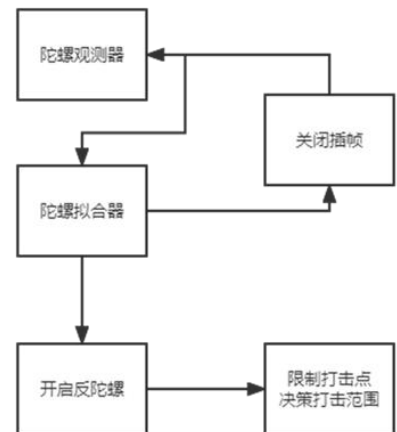


图 1.5.4-5 反小陀螺反扭腰流程图

击打小陀螺我们的方案是仅打击装甲板旋转出现一侧，如果打击旋转消失一侧很可能打击点已经消失，会出现云台甩飞或者打击虚空点。反陀螺以及反扭腰我们采用同样的观测器以及决策方式，不同的是需要对出现方向做相反处理。

数字识别

数字识别可以应对场地误识别，进行目标装甲板的筛选、进行目标跟踪等。

我们使用了 mobile_net 轻量化网络。为了增加网络模型的“鲁棒性”，我们将灯条也采集进数据集，同时，对数据集进行增广操作，减少亮度、对比度等对识别的影响。通过网络“剪枝”操作降低运算量，提高了识别速度。经过测试，我们的数字识别方案面对小陀螺状态下转动的装甲板也能正确识别，帧率稳定在一百帧以上。

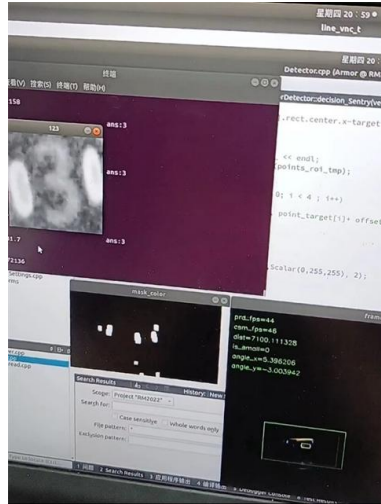


图 1.5.4-7 数字识别效果图

1.5.5 导航算法设计

1.5.5.1 系统架构

整体系统架构是基于 Ubuntu 操作系统和 ROS 机器人操作系统的自主定位导航机器人系统，ROS 系统采用模块化设计，节点间松耦合，采用消息传递机制，可重用性高，使得机器人操作系统具有更高可维护性和可拓展性。

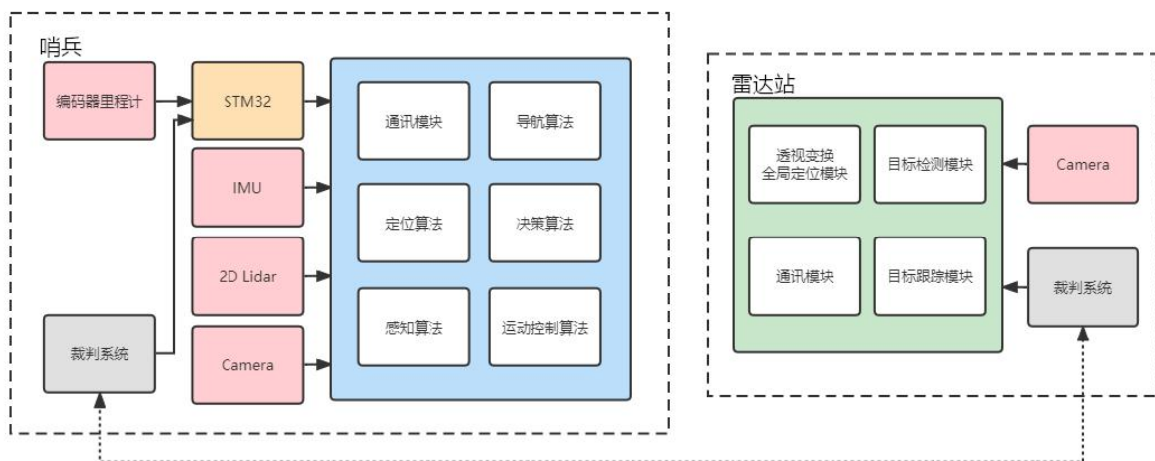


图 1.5.5-1 整体算法设计框架

定位算法：基于 AMCL 的定位算法、基于 EKF-ICP 的定位算法

感知算法：装甲板目标识别、定位与运动预测、坐标变换、下位机时间轴匹配

导航算法：全局 A*路径搜索、move_base 框架

运动控制算法：局部 TEB 跟踪器、纯跟踪轨迹跟踪、PID 跟踪器

决策算法：基于行为树的决策行为

Rqt_graph 图显示了机器人整体 ROS 机器人系统所启动的节点和话题之间的关系：

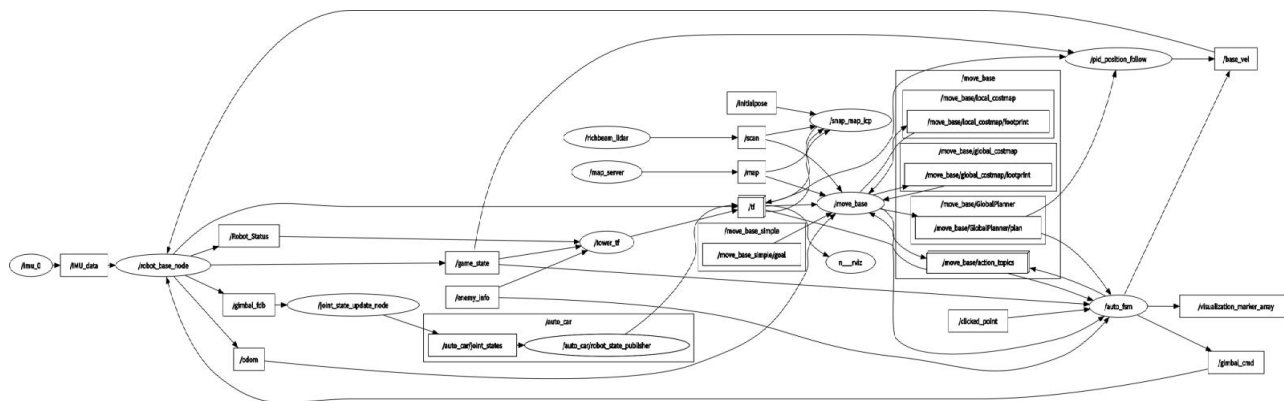


图 1.5.5-2 Rqt_graph 图

1.5.5.2 定位算法设计

在机器人的定位方案上，最常用的方法采用轮式里程计和惯性测量单元（IMU）进行积分来确定机器人的位置与姿态。由于舵轮本身的特性，使用编码器解算得到的角速度信息误差较大，我们使用 imu 进行角速度的测量，里程计进行线速度测量。

同时需要注意的是，无论是里程计还是 IMU 都存在累积误差的问题，进而可能造成定位的不准确。为了解决这个问题，综合和，成本和效果的考虑，我们使用了 2 个 2d 激光雷达，两个雷达数据做点云拼接得到全向点云数据，来对误差进行修正。在许多开源的解决方案中，AMCL（自适应 Monte Carlo 局部化）是一个常用的定位修正算法。然而，由于 AMCL 的粒子收敛速度相对较慢，定位效果不佳。因此，我们选择了更为高效的点云匹配算法 ICP+EKF 实现机器人定位。

ICP 是基于激光雷达的一种点云匹配算法，在借助前后连续两帧激光雷达扫描数据之间的变化（如机器人在环境中移动）的情况下，使用最小二乘优化方法来确定机器人的姿态角。相比之下，ICP 算法具有更高的计算效率和定位准确性。

1.5.5.3 激光雷达运动畸变去除

机器人在移动过程中，激光雷达的扫描角度和距离会发生变化，也就是产生了运动畸变，产生运动畸变的原因主要是当机器人朝向不同方向或者靠近墙壁时，整个机体会发生旋转

和平移，从而导致激光雷达所起到的作用不同。因此，在测量同一个位置时，激光雷达返回的数据也会有所不同，效果如下图所示。

2D 激光雷达运动畸变产生后，会导致点云数据的质量下降，从而影响到对机器人周围环境进行定位和规划路径等任务的准确性和可靠性。所以在进行点云配准前需要对雷达数据进行去畸变处理。

为了解决运动畸变问题，需要对激光雷达信息进行去畸变处理。通常使用 IMU（惯性测量单元）加轮式里程计测量机器人的姿态和运动状态，从而实现点云数据的去畸变处理。具体流程如为在一帧雷达周期内，获取 IMU 和轮式里程计传感器信息。其中，IMU 能够提供机器人的姿态信息，包括位置、速度、加速度和旋转角度等；轮式里程计可以提供车体的相对运动信息，例如车辆的线速度和角速度等。

然后根据 IMU 和轮式里程计数据，计算机器人在两次扫描之间所发生的位姿变化。通常使用集成导航算法，将 IMU 和轮式里程计提供的数据结合起来估计出机器人的运动路径，并计算出当前位置相对于初始位置的变化量，也就是机器人的位姿变换。

最后通过计算相邻帧数据时机器人的位姿变换，将一帧中不同位姿的点云数据变换到同时刻位姿中去，实现激光点云数据去畸变。将每个激光束的位置信息投射至车体坐标系中，然后将车体坐标系转换到世界坐标系中，就能够得到精确的运动畸变矫正后的点云数据。

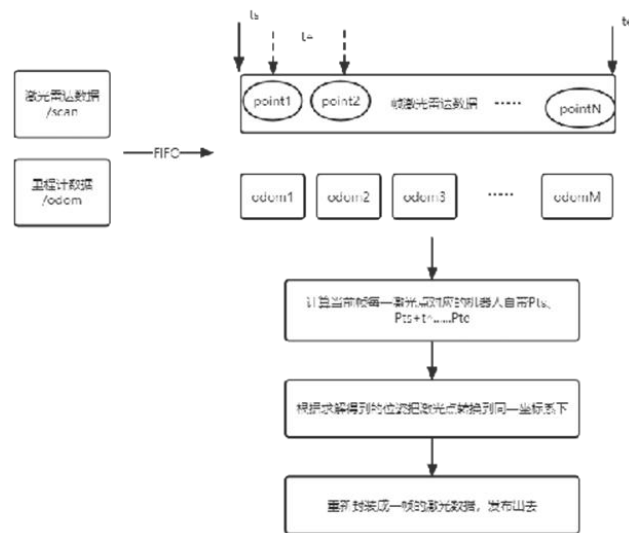


图 1.5.5-3 获取点云数据流程图

1.5.5.4 AMCL 自适应蒙特卡洛算法

AMCL 是一种基于蒙特卡罗方法的自适应定位算法，它的主要思想是在地图中随机放置

大量粒子，每个粒子代表一个假设的机器人位置。粒子根据机器人获取的传感器信息进行权重更新，在权重更新后重新抽样并迭代，最终粒子会收敛到机器人所在的位置。AMCL 算法不受场景限制，简洁高效且鲁棒性高，因此被广泛应用于机器人定位领域。

尽管 AMCL 算法有很多优点，但仍存在一些缺点需要注意。首先，由于 AMCL 算法是基于 Monte Carlo 方法的，所以当粒子数量较少时，定位精度会降低；反之，当粒子数量过多时，定位速度会变慢。其次，AMCL 算法的收敛速度比较慢，当机器人的速度较快时，可能会导致定位不准确，从而影响机器人的运动控制。

在前期的测试验证中，我们使用了 AMCL 算法进行机器人定位，但是由于哨兵机器人速度比较快，AMCL 算法定位效果较差，导致机器人在直道也会出现摇晃的情况，浪费时间并有一定危险性。为了解决这个问题，我们选择了更为高效的定位算法，例如基于激光雷达的点云匹配算法（如 ICP），以提高定位精准度和速度。

1.5.5.5 ICP 迭代最近点算法+EKF

为了改善定位效果，我们决定放弃 AMCL 算法，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，直接获得小车的位姿，在使用 EKF 进行定位更新，最终获得较好的效果。

在匹配算法上，我们使用迭代最近点（Iterative Closest Point, 下简称 ICP）算法，是最为经典的数据配准算法。通过求取源点云和目标点云之间的对应点对，基于对应点对构造旋转平移矩阵，并利用所求矩阵，将源点云变换到目标点云的坐标系下，估计变换后源点云与目标点云的误差函数，若误差函数值大于阈值，则迭代进行上述运算直到满足给定的误差要求。最终获得源点云变换到目标点云的旋转平移矩阵。

估计对应点（Correspondences estimation）

将激光雷达作为源点云，将二维栅格地图的每个像素点转换为一个点得到目标点云，为了加速计算，可以不需要计算 Target 点云中每个点到 Source 点云中一点的距离。可以设定一个阈值，当距离小于阈值时，就将其作为对应点。

迭代优化

找到对应点后，我们就构建了两组对应的点集：

$$P = \{p_1, p_2, \dots, p_n\}$$

$$Q = \{q_1, q_2, \dots, q_n\}$$

求欧式变换 R, t 使得:

$$\forall i, q_i = Rp_i + t$$

由于噪声和错误匹配, P 和 Q 并不能对应到空间中的同一点, 会存在一定的误差, 同时匹配算法必须得到一个确定的解, 所以我们要最小化的目标函数为:

$$\arg \min_{R, t} \sum_{i=1}^n \|(Rp_i + t) - q_i\|^2$$

求解 R 和 t 的方法主要有 SVD 和非线性优化, 由于时间有限, 本项目直接使用 PCL 库中的 ICP 实现, PCL 中的 ICP 算法是基于 SVD (Singular Value Decomposition) 实现的, 可以得到源点云到目标点云的坐标变换。

坐标变换与定位实现

用前面的方法就能得到当前小车坐标点到小车真实坐标的变换矩阵, 要实现修正小车定位, 我们沿用了 AMCL 中发布 `map` 到 `odom` 坐标变换的方式。

为了方便表示变换, 下文用欧式变换来表示坐标变换, 其中 R 为旋转矩阵, t 为平移矩阵。为了区分, 我们将变换前的 `odom` 和 `base_link` 坐标命名为 `odom_err` 和 `base_link_err`, 变换后的命名为 `odom_r` 和 `base_link_r`, `map` 坐标系为全局坐标系保持不变

其中 $T_{base_link_err}^{base_link_r}$ 为 ICP 匹配得到的坐标变换

需要求解 `map` 到 `Odom` 的变换, 即 $T_{map}^{odom_r}$

坐标关系如下图所示。

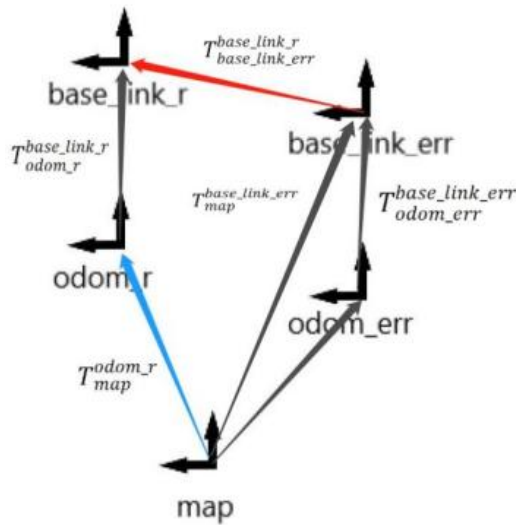


图 1.5.5-4 ICP 算法坐标关系

$$T_{map}^{odom_r} = (T_{odom_r}^{base_link_r})^{-1} T_{base_link_err}^{base_link_r} T_{map}^{base_link_err}$$

$$T_{odom_r}^{base_link_r} = T_{odom_err}^{base_link_err}$$

化简得：

$$T_{map}^{odom_r} = (T_{odom_err}^{base_link_err})^{-1} T_{base_link_err}^{base_link_r} T_{map}^{base_link_err}$$

结果即为修正定位需要发布的变换，但是由于 ICP 匹配需要进行多次迭代，定位效果存在延迟（实测 ICP 平均计算时间为 0.1s），并且匹配的频率较低（雷达频率为 10HZ 左右），不能满足使用需求，我们需要将这段时间小车运动加入到变换中。

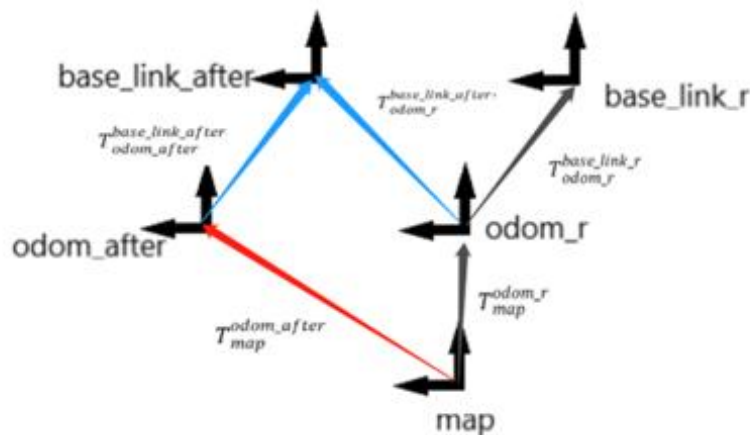


图 1.5.5-5 ICP 算法坐标关系 2

$$T_{map}^{odom_after} = (T_{odom_after}^{base_link_after})^{-1} T_{odom_r}^{base_link_after} T_{map}^{odom_after}$$

$$T_{odom_after}^{base_link_after} = T_{odom_r}^{base_link_r}$$

化简得：

$$T_{map}^{odom_after} = (T_{odom_r}^{base_link_r})^{-1} T_{odom_r}^{base_link_after} T_{map}^{odom_after}$$

将最终匹配结果和 imu 和里程计数据使用卡尔曼滤波进行融合，达到了很好的效果，该方案的定位频率理论上可以达到 200hz（与 imu 一致），且比较稳定，定位修正速度快。

1.5.5.6 SAC-IA + ICP 重定位算法设计

在哨兵机器人定位工作中，当机器人的传感器出现故障或数据异常时，会影响到机器人的定位精度。特别是在动态环境下，机器人也难以通过传感器获取准确的环境信息，从而造成定位错误，同时由于对抗赛上有复杂的场地，机器人存在定位丢失的问题。

我们使用了 SAC-IA+ICP 算法。SAC-IA 算法是一种基于采样一致性（Sample Consensus, SAC）思想的刚体点云配准方法。它适用于不同尺度、不同形状、带噪声和离群点的数据集匹配，特别适用于全局点云匹配。SAC-IA 算法通过迭代次数、采样率和阈值等参数，在匹配多个重叠点云之间寻找共同区域，从而得到初始的相对位姿；之后使用修正后的均方误差来评估结果，并在推进轮次到达设定最大值或结果满意度达标时停止。

ICP 算法是一种局部点云配准方法，该方法将两个数据集之间的配准问题视为最小二乘问题，并不断迭代寻找与目标云匹配的刚性变换，如旋转和平移。

针对机器人定位丢失问题，我们采用了 SAC-IA 和 ICP 配准算法。首先，SAC-IA 算法对机器人所处环境进行全局点云匹配，并通过筛选掉不合格的数据，求出一个相对较精确的初始定位矩阵；然后，ICP 算法利用这个初始矩阵对机器人的运行进行优化匹配，得到更为精确的位姿估计值。通过使用这两种方法的组合，我们成功实现了机器人在定位丢失后的自我重定位。

结合使用 SAC-IA 和 ICP 可以充分发挥它们的优点。SAC-IA 算法尤其适用于全局点云匹配，在获取初始变换矩阵方面表现良好。ICP 算法则能够实现高精度的局部点云匹配，适用

于逐步细化姿态信息。而将这两种方法结合使用，SAC-IA 提供粗略的全局匹配，去除误差大的显著错误数据；ICP 通过细小范围的匹配，完成对原始姿态的微调。在实际应用中，我们不断优化算法参数、运动模型等策略，提高算法的鲁棒性和匹配精度。

1.5.5.7 导航算法设计

哨兵机器人的导航算法是保证机器人能够安全、高效地移动至目标点的关键。在实现哨兵机器人的导航过程中，我们采用了 ROS 中 Navigation 工具包作为整体框架，并改进和优化。

首先，Map Server 通过激光雷达进行扫描，将机器人周围环境信息转化为形式类似于网格和标记的形式，并生成一个全局地图。这个全局地图成为后续路径规划所依赖的基础。而在这一过程中，涉及到了 ROS 中的雷达数据获取、点云处理、建图等技术。

其次，costmap_2d 模块在得到 Map Server 的地图之后，在雷达点云数据的支持下，costmap_2d 会计算出环境中不同区域的行动代价，并为这些区域分配不同的代价值。这些区域可以是静态的障碍物，也可以是半静态和动态的障碍物。同时，costmap_2d 还可以为机器人周围的环境设置安全区域和行动代价，以完善路径规划和控制策略，避免机器人与环境发生碰撞。

全局规划和局部规划是 Navigation 工具包中最核心的两个模块之一。在全局路径规划中，将当前车辆内部“表示”与全局描述地图合并，输出连续的曲线轨迹。ROS 中提供了多种全局路径规划算法，例如基于 A*、Dijkstra 等搜索算法的规划方式或者基于膨胀地面网格或可变边界倾向前沿算法(Voronoi Diagram)的算法。这些算法都包含了对不同行动代价值的处理方法，可以为机器人选择最优路径。

在局部路径规划中，机器人需要根据当前状态和全局路径规划的结果，在遵循安全区域和行动代价约束的条件下，选择合适的速度和方向进行移动，并不断进行感知和反馈调节，以确保机器人在追踪过程中保持稳定性并尽可能避免碰撞。常见的局部路径规划算法有基于微分约束边界发射方法(DWA)，使用多项式曲线(Polynomial Curve)描述算法或非线性模型预测上下文(Trajectory Optimization Based-On-Nonlinear Model Predictive Control, TONMPC) 等。

TEB (Tree-structure Elastic-band planner) 算法是我们在实践中测试的一种局部路径规划算法，它可以基于优化理论、轨迹拟合、轨迹增量走积分等技术，针对当前位置周围较小半径范围内的障碍物进行筛选并转换为函数形式，然后定义全局障碍物设定，最后通过在规划空间内解决较小的力学优化问题，得出机器人行进的最终路径。

虽然 TEB 算法在一定程度上可以解决导航中的难点和问题，但该算法本身也存在着参数过多难以调整或是不能使哨兵机器人小陀螺过程中运动的问题。因此我们使用了 PID Position Pursuit 轨迹追踪算法 + 人工势场避障法来解决这些问题。PID 控制器本质上是一个包含比例、积分、微分三个部分的反馈控制系统，可以将机器人的位置误差加权求和后调整机器人朝向角度，实现目标路线的追踪；而人工势场避障法则是利用在动态环境中植入的障碍物感知结果来构建安全渠道和通道，使其无论何时都可以保证机器人的安全。

1.5.5.8 路径规划

在全局规划方面，我们使用 move base 导航包，其提供的 A*算法利用 map_server 所提供的由小车通过激光雷达所创建出来的全局代价 地图和局部代价地图、机器人当前自身的位置和目标的位置等数据来规划出从起点到目标终点的一条最优路径。

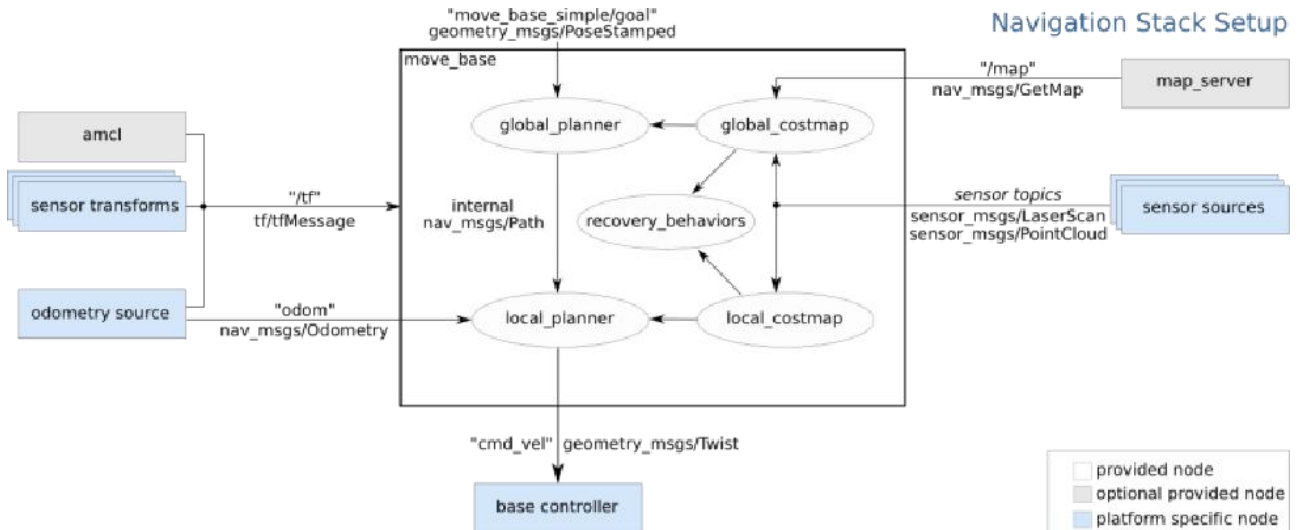


图 1.5.5-6 路径规划流程图

其中使用 action 方法来实现目标点的发布，可以回传当前导航的状态，同时通过判断导航点与自身位置的欧式距离进行发布下一目标点

1.5.5.9 运动规划与轨迹跟踪

TEB 算法

在前期的研发中我们使用了开源的 teb 算法，该算法把路径规划问题描述为一个多目标优化问题，即对最小化轨迹执行时间、与障碍物保持一定距离并遵守运动动力学约束等目标进行优化。

作为一种比较常见的路径规划方法，teb 算法被广泛应用于差分、全向、阿克曼模型等各种类型的车辆行驶。并且由于该算法采用了开源的形式，可以很容易地进行算法实现和代码调试，为研究人员和开发者提供了便利。

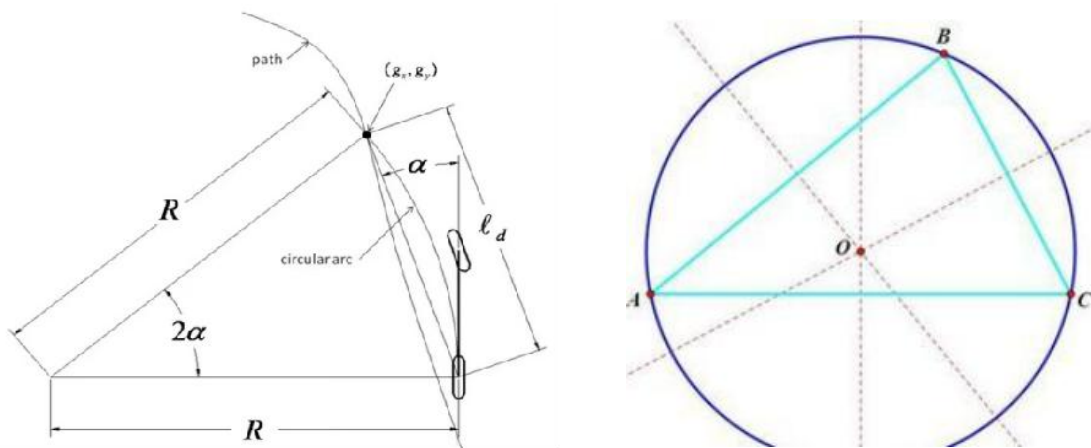
然而，teb 算法在实际场景中也存在着一些局限性和缺点。首先，该算法的计算复杂度较大，可能会导致速度和角度波动较大、控制不稳定等问题；其次，步兵移动过程不能进行小陀螺，在具体实现中要考虑其可行性和实际效果。同时，尽管该算法具有通用性和灵活性，在实际应用中可能需要调整过多的参数，以在复杂情况下达到最优解。

Pure Pursuit 纯追踪算法

为了解决 teb 以上问题，我们使用 Pure Pursuit 纯追踪算法重写了轨迹跟踪，该方法将车辆简化为自行车运动学模型

$$\delta(t) = \tan^{-1} \left(\frac{2l \sin(\alpha(t))}{l_d} \right)$$

其中 L 为轴距， R 为当前转角下的转向半径， (g_x, g_y) 是下一个要跟踪的路径点，它位于已经规划好的全局路径中， a 代表当前车身姿态和目标路径点的夹角，需要控制车辆使得车辆的后轴经过该路径点，纯追踪算法的控制量的最终表达式为：



纯跟踪算法中前视距离的选择比较关键，过近步兵会出现左右震荡，过远会导致转弯时装上障碍物，我们使用三点法计算前方一段路径的曲率，然后通过曲率变化动态调整前视距离，得到了不错的控制效果。

$$R = \frac{abc}{\sqrt{(a+b-c)(a-b+c)(b+c-a)(a+b+c)}}$$

PID Position Pursuit 轨迹追踪算法 + 人工势场避障法

为了使哨兵能小陀螺跟踪全局路径，我们重写了速度规划器。速度规划器通过检测全局路径话题是否存在，当接收到全局路径便根据当前坐标选取机器人前方固定距离全局路径上的点作为局部目标点，通过监听 `base_link` 与 `map` 坐标系的 `tf` 变换获取旋转矩阵，根据当前航向角再将速度分解到 `base_link` 坐标系的 `x` 和 `y` 方向上，同时给机器人固定的 `wz` 角速度，就能实现小陀螺移动。

在动态避障上，我们使用了人工势场法，使用雷达数据，在机器人周围创建一个二维势场模型，将机器人看作一个带电粒子，以势能作为代价函数，动力学模拟机器人在环境中运动，并在动态遇到障碍物时实现避障、躲避和前进的控制。

其数学模型表示为以下公式：

$$F = \nabla U = \nabla(U_{att} + U_{rep})$$

其中，`F` 表示实际作用于机器人上的力；`Uatt` 表示吸引机器人移动向目标点的引力场能量，即机器人受到正向梯度的约束；`Urep` 表示排斥体系的斥力能量，即机器人受到反向梯度的约束。假设机器人和目标点位置分别为 `X0` 和 `Xg`，则其函数形式可表示如下：

$$U_{att} = \frac{1}{2} \bar{k}_{att} (x - x_g)^2$$

$$U_{rep} = \begin{cases} \frac{1}{2} \bar{k}_{rep} \left(\frac{1}{q} - \frac{1}{r} \right)^2 & (r < q) \\ 0 & (r \geq q) \end{cases}$$

其中，`r` 表示机器人的位置，`q` 为机器人的安全距离。当机器人距离障碍物的距离小于 `q` 时，其会受到斥力影响，可能会躲避避障；反之，当机器人距离障碍物的距离大于 `q` 时，其不会感受到障碍物的影响。最终实现机器人轨迹跟踪与避障效果。

1.5.5.10 决策算法设计

在决策算法上，我们使用了行为树作为哨兵的决策，行为树是一种面向目标的、可扩展的决策框架，它通过将机器人所需的所有行为以树形结构进行组织，简化了复杂的决策过程。为了更好地实现机器人的高效运行和易于修改，我们将整个行为树分为底盘行为、云台行为和总行为树。

其中总行为树负责订阅各种信号源，包括了裁判系统比赛状态、机器人的血量、剩余子弹数量、枪管热量、受攻击装甲板 id、机器人当前位置、导航当前状态（导航中，到达，导航失败等）、获取当前状态信息，另外也包括对不同信号的分类处理等工作。底盘状态和云台状态则由独立的线程进行管理，并封装成单独的类，为其他模块提供接口，实现状态信息更新和读取等功能。

底盘状态和云台状态分别由单独的线程进行管理并封装到单独的类中，提供了状态更新和获取当前状态的接口。云台状态主要是准备状态、搜索状态、反击状态和攻击状态，其中反击状态根据裁判系统得到的受攻击的装甲板判断大致的角度位置。底盘状态主要是重定位状态、低速小陀螺状态、导航状态、低速小陀螺状态、躲避状态和重定位状态，其中巡逻通过在多个点之间循环进入导航状态实现，当短时间多次受到攻击进入躲避状态。

整体结构能够方便地理清各状态之间的关系，为赛场上的决策提供坚实的基础。

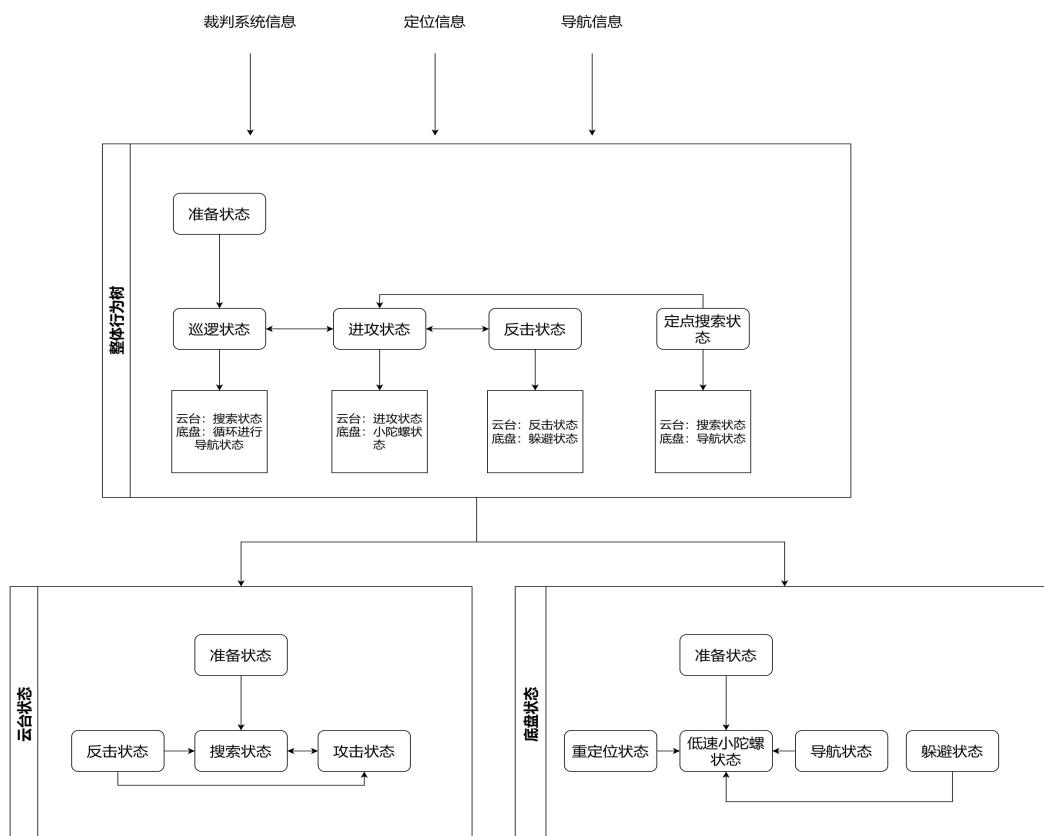


图 1.5.5-7 决策算法结构设计

1.5.5.11 仿真与可视化设计

仿真和可视化技术在比赛和算法验证中发挥重要作用，能够大幅度提高研发过程中的效率和准确性。同时，它也具备许多优点，如可以快速更改环境参数、难以观测的反应物，均能得到较为精确的评估和分析结果等。

在备赛过程中中，仿真使得导航算法的开发和机械电控出车能够同步进行，大大提高了研发的效率。此外，在花费过多时间和金钱在实际场景测试上之前，仿真技术可以减少团队的开销和风险，给予更多精力在设计和改进机器人方案上。

可视化能让我们更直观的看出当前程序运行状态，定位出问题所在。

1.5.5.12 赛场仿真设计

在缺乏官方赛场环境的情况下，仿真技术可以提供一个有效的测试平台，并帮助团队更好地掌握机器人竞赛的各项技能和知识。

为了实现这一目标，我们使用 gazebo 软件构建了哨兵的仿真模型，并通过官方提供的图纸完成了仿真环境的搭建。在仿真环境中，我们进行了建图、导航测试等一系列操作，以检验导航与定位算法的可行性和准确性。

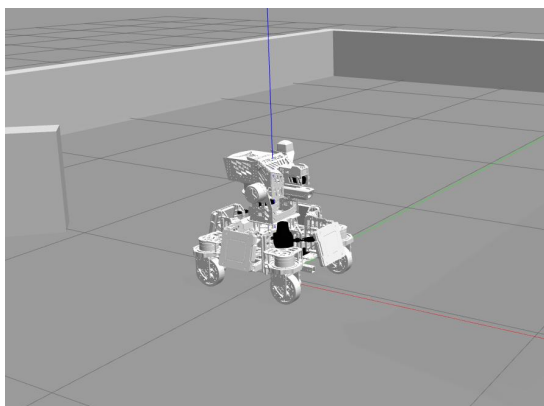


图 1.5.5-8 哨兵仿真

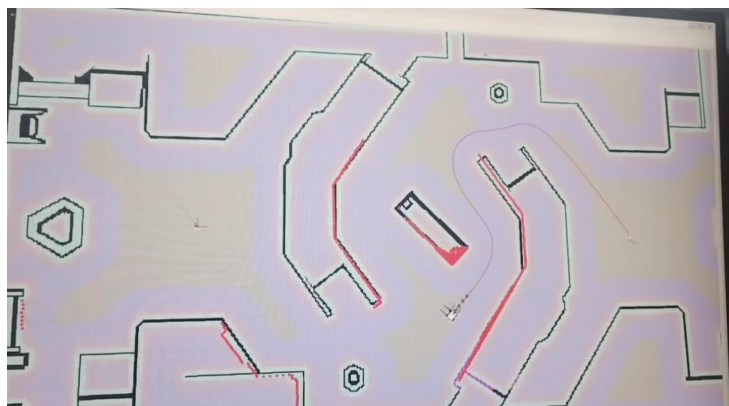


图 1.5.5-9 导航测试

1.5.5.13 点云配准可视化

在点云配准算法测试和调参的过程中，我们利用 ROS（机器人操作系统）中的 `rviz` 和 `dynamic_reconfigure` 工具，开发了一个点云算法测试与动态调参工具。

`rviz` 是一种 ROS 可视化工具，能够对机器人进行三维建模、路径规划、虚拟现实仿真等功能。在本项目中，`rviz` 主要用于导入点云模型，并结合定位信息实现可视化展示。通过 `rviz` 可以观测到点云配准前后的效果，便于调整参数并优化算法。

另外，`dynamic_reconfigure` 工具是 ROS 中的一个高级的动态配置服务器，它能够方便地改变运行时 ROS 节点的参数，而不需要重新编译。在我们的点云算法测试与动态调参工具中，`dynamic_reconfigure` 工具主要被用于快速改变点云配准算法的参数，比如调整匹配窗口大小、匹配距离阈值等重要参量。这样就能在不停止已有进程的情况下，快速迭代参数并及时观察到改变所带来的效果。

设计这款工具的初衷在于增强实验的自动化程度、有效性和效率，通过分析记录结果，调整配置的参数，及时完善当前算法。同时，点云算法测试与动态调参工具中保存了仿真过程中产生的大量数据，可以根据不同要求进行分析实验，帮助我们更好地理解算法性能。

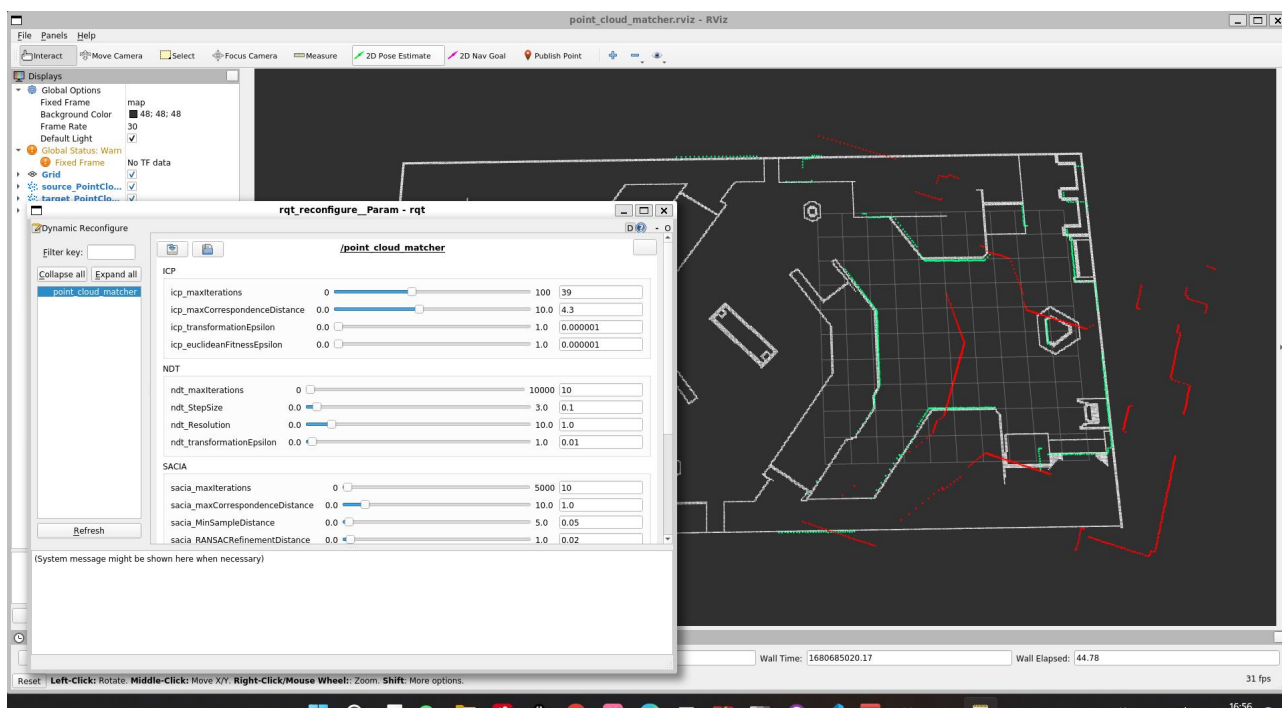


图 1.5.5-10 动态调参工具示意

1.5.5.14 保护壳方案设计

本赛季云台线路保护采用小背包设计，通过合页魔术贴使保护板材方便开合，对电控实时调测检查起到很大的便捷性，另外又能满足赛场上的保护要求。底盘保护通过 sw 进行钣金设计，折叠拼接 1mm 黑色 PC 板作为底盘与轮系保护壳，在能够满足每个轮系单独运动行程变换的实时保护同时设计流线型走线外壳，让机器具有一定设计美感。

全车多处采用三角形结构，有利于结构的稳定，在受到弹丸击打时不易产生变形与破损，各部分的连接采用螺丝与魔术贴相结合，有利于快速拆卸进行内部检修与维护，保护壳与四边的铝管通过轧带固定，便于拆卸的同时能紧密固定与底盘上，在 PC 板的弯曲线后贴有大量纤维胶布，起到强化作用，使其不易变形损坏，提高抗击打强度与使用寿命，整车以哑光黑色为主调，整体呈现出简约轻便的气质。

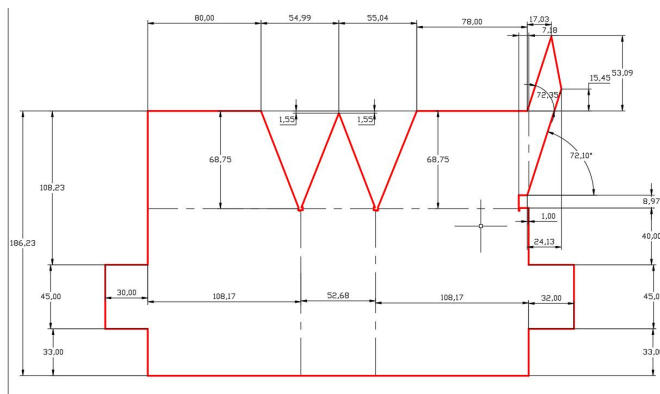
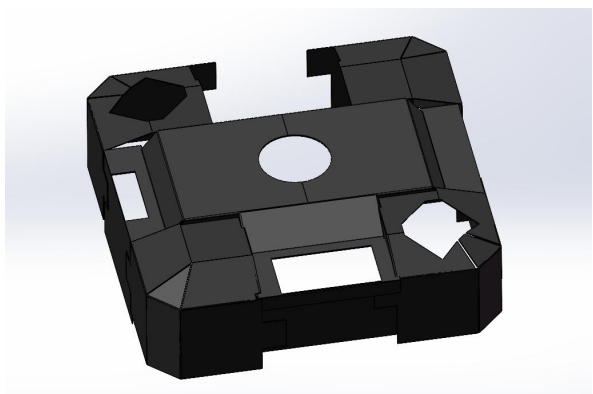
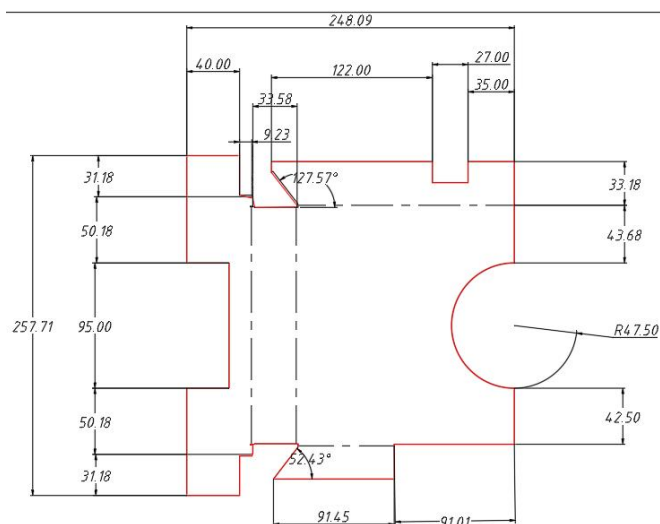
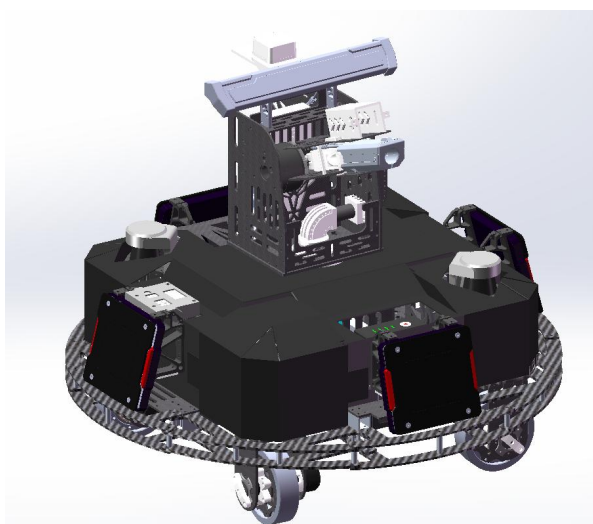


图 1.5.5-11 初代哨兵保护壳方案

1.6 研发迭代过程

1.6.1 机械结构设计

1.6.1.1 机械迭代过程

版本号	功能或性能详细说明	完成时间
V1.0	实现以基本平移、小陀螺以及下台阶	2022.11.16
V1.1	优化悬挂轮系，框架运行状态下更平稳	2023.01.20
V1.2	优化下轮系，减小轮阻达到理论值最大速度	2023.02.15

表格 1.6-1 舵轮轮系底盘版本迭代过程记录

版本号	功能或性能详细说明	完成时间
V1.0	满足机器人制作守则、达到基本设计需求	2022.11.15
V1.2	优化云台，进行 pitch 轴重心配平	2023.01.01
V1.3	优化缩减云台链路、轻化云台	2023.03.01

表格 1.6-2 云台机械版本迭代过程记录

版本号	功能或性能详细说明	完成时间
V1.0	链路流畅不卡弹	2022.11.20
V1.2	结合链路输弹效果调测限位	2023.01.15
V1.3	发射器电机选型测试、硬度调测	2023.02.10
V1.4	优化链路减少能量损耗	2023.03.05

表格 1.6-3 链路、发射器版本迭代过程记录

1.6.1.2 机械解决方案

舵轮轮系机械重点问题解决

	问题描述	原因分析	方案与效果	版本号	负责人
1	激光雷达无法稳定收集准确数据	自适应机构在运动过程会对惯性力有过大反应，在运动过程有过大的摇晃，无法保持平稳导致激光雷达无法准确收集准确数据	将自适应传动机构改为独立悬挂结构，使用直选导轨滑块作为柔性连接	V1.0	梁智恒
2	无法达到最大理论速度	在地胶等硬度低地面上行驶，会使轮子陷入地面当中，使轮子与地面的接触面积增大，牵引力减小，转动阻力增大。	将包胶轮下轮系更改为胎皮轮毂，减少了轮子与地面接触面积，提高了轮速，达到最大理论速度	V1.1	梁智恒
3	小陀螺过程中发射器出现 ROLL 轴转动	底盘框架重量配平不均匀，导致四段柔性连接无法保持一致	通过对框架以及避震以及框架连接点的位置，大幅度减少 ROLL 轴摇晃	V1.2	梁智恒

表格 1.6-4 舵轮轮系机械重点问题解决

云台机械重点问题解决

	问题描述	原因分析	方案与效果	版本号	负责人
1	云台 yaw 轴拆卸零件十分繁琐麻烦	云台 yaw 轴设计装配环节过密集，滑环线线径过大不易折弯，在 yaw 轴轴承内圈中因还有链路碳管而导致走线空间不足。	1.简化 yaw 轴设计，链路装配增加关节点，减少过多“环环相扣”。 2.增加内圈走线空间与预留线路折弯更大半径范围。	V1.0	纪因

	问题描述	原因分析	方案与效果	版本号	负责人
2	云台 pitch 轴电机运行过烫。	云台发射器机构整个前置于 pitch 前端，力臂较大情况下导致电机需要输出更大力矩进行运动，超负载工作造成电机极易发烫。	给 pitch 轴搭配拉簧进行重力配平，减少电机工作负荷。	V1.2	纪因
3	云台链路长，子弹无法进行限位突破。	在链路中子弹会错位无规则排列挤动，部分推力变分力消耗于链路内壁。	通过改变链路转弯曲率、缩小链路内径、调节轴承位置拟合路径走向，减少子弹链路中的错位量，尽可能保证力的传递高效率。	V1.3	纪因

表格 1.6-5 云台机械重点问题解决

链路、发射器重点问题解决

	问题描述	原因分析	方案与效果	版本号	负责人
1	链路卡弹	轴承选型过大且间隔较大，特别是在转弯处容易出现三个链路轴承刚好卡住子弹三力合一，造成链路卡弹。	选用小轴承 2*5*2.5，缩小轴承轴间距拟合路径，防止子弹与轴承三力合一卡住。	V1.0	纪因
2	子弹无法突破限位	云台链路过长，子弹经过五个转弯后能量损耗过大，导致其到达限位时基本已无力的传递而无法进行限位突破。	在原有限位基础上调整限位高度，削弱顶针作用，使得子弹能够克服顶针作用进入发射机构。	V1.2	纪因

	问题描述	原因分析	方案与效果	版本号	负责人
3	子弹出现掉速、射速波动范围大。	在调试中随着摩擦轮包胶磨损，往往射速逐渐不稳定。而原旧款朗宇电机在功率拉满情况下也无法对子弹射速做出一个相对较长时间的维持。	测试朗宇电机 2.0、3.0 版，最终选用 2.0 版。因为其相对充足的电机余量给电机对摩擦轮明显损耗时能做到维持射速调节的更多时长。在限位测试 1.0 中，朗宇 2.0 将摩擦轮使用寿命提升成至少 1500 颗。	V1.3	纪因
4	子弹射速因摩擦轮损耗严重而波动掉速、弹道散化，摩擦轮使用寿命低于 1000 发。	子弹射击中摩擦轮聚氨酯因多次摩擦磨损严重，表面出现坑坑洼洼，后期子弹与摩擦轮表面接触不一，每颗子弹因此加速过程明显不同，弹道效果、射速均出现不同。	在原有限位基础上调测摩擦轮轴间距与摩擦轮硬度，两变量搭配测试，实验测出最佳搭配。	V1.4	纪因

表格 1.6-6 链路、发射器重点问题解决

1.6.2 嵌入式软件设计

1.6.2.1 软件迭代过程

版本号	功能或性能详细说明	完成时间
V1.0	上个赛季最终的上场版本	2022.08.08
V1.1	针对调试新增遥控器模式下的功能	2023.01.01
V2.0	重构代码，分层分模块	2023.02.05
V2.1	新增系统状态监控任务与看门狗复位	2023.02.06
V2.2	增加与导航视觉联合通讯	2023.03.26
V2.3	增加舵轮里程计	2023.04.02

表格 1.6-7 软件迭代过程记录

1.6.2.2 软件解决方案

	问题描述	原因分析	方案与效果	版本号	负责人
1	舵轮里程计数数据不准确	1.舵轮里程计解算代码有较大误差	重新详细推演了舵轮里程计代码。	V2.1	刘旭龙
2	视觉数据时间轴不匹配	视觉侧图像与数据处理、通信具有延时	视觉端传感器存储历史值，进行动态时间轴匹配。	V2.2	刘旭龙
3	视觉自瞄，目标突然出现时，云台会出现过冲	目标突然出现时，原本数据是常数，但由于滤波器处于收敛过程，从而有了一个不存在的变化率	利用上位机精确测定各种极限情况下的滤波器收敛时间，保证收敛后再进行处理。同时，减少非必要滤波，减少收敛时间。	V2.3	刘旭龙
4	与导航通讯数据错误	1.在与导航通讯时，会有一个字节错误	最终发现时结构体字节偏移问题，增加字节偏移数据	V2.4	刘旭龙

表格 1.6-8 软件重点问题解决方案

1.6.3 视觉算法设计

1.6.3.1 数字识别迭代优化

一开始是用 svm 机器学习方法，输入为透视变换后的二值化图像，在固定曝光下速度快，识别准确率比较高，但在受环境影响非常大，不同环境下二值化后结果不同，容易导致误识别，我们试着使用大津法降低影响，但最终效果一般。

后来改为使用 mobile_net 轻量化神经网络。我们对数据集进行增广操作，减少亮度、对比度等对识别的影响。通过网络“剪枝”操作降低运算量，提高识别速度，经过测试，我们的数字识别方案面对倾斜的装甲板也能正确识别，帧率基本稳定在一百帧以上。

1.6.3.2 预测迭代优化

一开始我们是对云台要转的相对角度，加上陀螺仪测到的角速度进行预测。这种预测方法和物体实际的运动轨迹并不相同，于是，我们改为对装甲板在世界坐标系中的坐标点进行预测。

1.6.4 导航算法设计

1.6.4.1 导航迭代过程

对机器人各项功能点进行测试验收，记录测试环境、测试设备等信息。

版本号或阶段	功能或性能详细说明	完成时间
V1.0	amcl 定位与导航	2022.03.18
V1.1	ICP 匹配实现定位，替代 amcl	2022.08.02
V1.2	位置 pid 实现小陀螺运动	2022.12.05
V1.3	EKF+ICP 匹配实现定位	2023.02.08
V1.4	实现与裁判系统通信	2023.03.15
V1.5	实现自主决策，实现占领增益点、回血、巡逻、攻击等模式	2023.04.01

表格 1.6-9 导航迭代过程

1.6.4.2 导航解决方案

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	解决人员
1	使用 amcl 定位效果不佳，且更新频率较低	amcl 算法包本身原理为粒子集更新，更新与迭代会有延迟	使用 icp 匹配替代 amcl	V1.1	算法工程师：申国豪
2	teb 算法下机器人无法在小陀螺过程中导航	teb 算法没有小陀螺实现	使用位置 pid 自己实现轨迹跟踪算法	V1.2	算法工程师：申国豪
3	icp 匹配结果会有波动，且部分时间匹配失败	icp 匹配本身会有误差	使用 ekf 对 icp 匹配和里程计和 imu 的结果进行数据融合	V1.3	算法工程师：申国豪
4	机器人在上坡或者盲道上 2d 激光雷达有异常数据	在上坡或盲道时，雷达会扫描到地面或天空	使用 imu 数据估计雷达点云高度，进行滤波	V1.3	算法工程师：邱旭林
5	位置 pid 轨迹跟踪会出现数据波动	路径规划和定位会有波动，导致机器人不稳定	对规划的结果加入低通滤波	V1.3	算法工程师：梁乐彬

表格 1.6-10 导航重点问题解决方案

2. 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%)
申国豪	人工智能、大三、导航组组长、 哨兵导航定位负责人	负责哨兵的导航、定位与决策开发， 包括定位与重定位、路径规划与运 动控制、行为树等	11%
刘旭龙	电子信息类工科、大三 哨兵电控负责人	负责哨兵电控与舵轮步兵、视觉反 小陀螺算法研发	11%
吴静琳	人工智能、大二 哨兵视觉负责人	负责整体哨兵视觉	11%
纪因	机械设计制造及其自动化 大二、哨兵机械负责人	负责底盘、云台机械结构设计及整 车装配、发射器研发	11%
曾扬晨	软件工程、大三、视觉组组长	负责数字识别与反小陀螺算法开发	11%
邱旭林	自动化、大三、导航组成员	负责下位机与上位机通信、雷达与 imu 的驱动改进以及数据预处理	11%
邱培涛	电子信息类工科、大三 导航组成员	负责与裁判系统通信	8%
梁乐彬	农业机械化及其自动化 大四、前导航组组长、顾问	负责提供技术支持与评估	8%

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%)
梁智恒	机械设计制造及其自动化 大三、机械组组长	负责哨兵舵轮轮系设计	8%
钟雍泰	车辆工程、大一、机械组成员	负责测试架设计及哨兵机械装配	4%
陈秋杰	机械设计制造及其自动化 大三、副队	负责提供技术支持与评估	3%
洪利聪	农业工程与信息技术 研一、导航组成员	负责决策可视化开发	3%

3. 参考文献

类别	文献链接	参考内容
导航	https://github.com/Critical-HIT-hitsz/RMUA2022	参考决策逻辑
导航	https://github.com/Privilger/RoboMaster-Simulator	参考搭建仿真环境
机械	https://bbs.robomaster.com/forum.php?mod=viewthread&tid=7632	下供弹链路设计