

Using a 32-bit motor driver chip and Field-Oriented Control (FOC), the RoboMaster C320 Brushless DC Motor Speed Controller enables precise control over motor torque.

Exclusively designed for the RoboMaster M30S P18 Brushless DC Motor and C320 Brushless DC Motor Speed Controller, the M310S Acromech RT includes several cables and a terminal block.

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

All M30S Acromech RT includes several cables and a terminal block, ensuring a complete and reliable system when for your RoboMaster robot.



ROBOMASTER

机甲大师超级对抗赛

技术方案

华南师范大学 PIONEER 编制

2023 年 8 月 发布

前言

本技术方案由华南师范大学 PIONEER 编制，适用于 RoboMaster2023 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	撰写人员 3
机械	杨阳	张子熠	
硬件	赵俊杰	曾昭庆	
软件	赵俊杰	钟政旭	
算法	蒋荣桀	姚亭屹	肖佳炆
其他	唐天峰	蔡志恒	李观锋

本文档撰写旨在对技术的总结、传承和发展。技术方案包括技术需求分析、技术方案实现和技术迭代过程三个部分。技术需求分析由背景和目标出发，结合其他学校机器人分析和定义机器人功能，确定赛季研发方向和方案细则。设计方案部分详细叙述整车系统功能的设计思路和实现流程。技术迭代过程部分重点记录对关键机构的迭代测试和困难问题的解决过程。

目录

前言.....	2
1 概述.....	4
1.1 背景&目标.....	4
1.1.1 战场定位.....	4
1.1.2 需求分析.....	5
1.1.3 技术方案.....	5
1.2 其它学校机器人分析综述.....	6
1.2.1 步兵底盘构型的百花齐放.....	6
1.2.2 机动枪管加持下云台的千层套路.....	7
1.2.3 老生常谈的 17mm 弹丸发射精度.....	8
1.3 机器人功能定义.....	9
1.4 机器人核心参数.....	10
1.5 设计方案.....	12
1.5.1 机械结构设计.....	12
1.5.2 硬件设计.....	35
1.5.3 软件设计.....	40
1.5.4 算法设计.....	55
1.6 研发迭代过程.....	83
1.6.1 研发迭代记录.....	83
1.6.2 测试记录.....	89
1.6.3 版本迭代过程记录.....	91
1.6.4 重点问题解决记录.....	92
1.7 团队成员贡献.....	93
1.8 参考文献.....	93
1.9 技术方案复盘.....	94
1.9.1 赛场性能表现情况分析.....	94
1.9.2 赛场性能表现与规划对比分析.....	95
1.9.3 经验总结.....	96

1 概述

1.1 背景&目标

RoboMaster 作为一个以装甲板击打作为基础伤害原理的比赛，为了增加自身机器人的生存能力，唯一的办法就是尽可能降低对方的命中率。在 2018 年“小陀螺”由中国矿业大学首次带到了赛场上之后，此功能已经变成了步兵的标配。但由于规则功率限制，和各大战队自瞄水平的提高，加上规则场地的盲道限制，传统麦轮步兵的“小陀螺”时的速度和机动性已经不足以满足战场生存需要，故我们在 2023 赛季采用了全向轮 X 型布置设计，在未改变其直线加速能力的前提下，提高了其“小陀螺”速度。

在 2022 赛季国赛备赛中我们曾研发过一版全向轮步兵，但由于疫情原因该车并未上场实战，经过我们自研测试发现一代全向轮步兵底盘性能不佳，经过排查是由于其轮组复用旧件造成的一些妥协设计导致机械空程较大最终导致效果不佳。由于全向轮底盘理论上优秀的性能，我们决定在新赛季再研第二版全向轮步兵。

由于本赛季哨兵固有枪管数量的改变且有空余的机动枪管存在，我们决定研发双枪管云台，双枪管的高爆发在机器人近距离交战是非常强力的。结合底盘优秀的生存能力，我们想赋予新赛季全向轮步兵一个近身搏斗特化的定位。

本赛季全向轮步兵主要分为以下几个目标实现：第一，底盘上解决运动性能问题，在底盘功率限制不变的前提下提高平移和小陀螺速度。第二，双枪管双发射云台在解决大载重的问题同时优化其响应速度，目标做到与单枪管云台响应速度及操作手感无异。

1.1.1 战场定位

正面对抗特化型地面单位：我们需要一个能力特化的单位处于正面战场，通过其强大的性能，达到站场甚至控场的作用，以掩护其他单位任务的良好进行。

由此战场定位确定了步兵需要承担的任务：

- ①近距离对抗
- ②场面控制
- ③攻击旋转前哨站

1.1.2 需求分析

任务	需求分析	设计方案
近距离对抗	<ol style="list-style-type: none"> 1. 强大的输出火力 2. 稳定优秀的移动性能 3. 防止与场地或机器人固连 4. 重心降低 5. 整车机械结构合理稳定 	<ol style="list-style-type: none"> 1. 双枪管双发射机构云台构型 2. 基于整车状态观测的视觉识别方案 3. 外围防撞圆形投影防卡防固连设计 4. 降低 yaw 轴高度 5. 底盘框架设计框型支撑结构
场面控制	<ol style="list-style-type: none"> 1. 生存能力 2. 持久输出 	<ol style="list-style-type: none"> 1. 采用全向轮底盘的构型，特化小陀螺功能 2. 装甲板竖直错开设计 3. 双枪管切换使用方案
攻击旋转前哨站	<ol style="list-style-type: none"> 1. 对旋转装甲板的特化功能 	<ol style="list-style-type: none"> 1. 视觉反前哨站功能方案

<表 1：步兵需求分析>

1.1.3 技术方案

1.1.3.1 云台

由于本赛季哨兵固有枪管数量的改变且有空余的机动枪管存在，我们决定研发双枪管云台，双枪管的高爆发在机器人近距离交战能占有较大优势，符合我们对该步兵战车定位的需求。

1.1.3.2 底盘

底盘构型基于需要控场能力的需求优先选择全向轮轮组；在框架设计以降低整车重心为主旨，合理分配电路器件摆放，降低 yaw 轴高度；底盘强度上设计承力碳板形成框型结构增强整车稳定性。

1.1.3.3 武器系统

搭载以整车预测为核心的视觉识别方案，通过装甲板位姿预测出目标平移以及旋转的全部状态，再对其他装甲板的位置进行反推，从而实现整车状态预测；本方案的优势为能够获取任意时刻下所有装甲板的位置，即可实现提前预瞄，大大提高击打装甲板的时间和准确率，

且无论是对简单平移运动还是对平移加小陀螺的复杂运动，都有优异的预测性能。

1.2 其它学校机器人分析综述

在过去的 22 赛季，我们观察到其他高校的步兵机器人有着许多出色的、值得我们学习的地方。由于规则对步兵机器人在赛场上的定位，各大高校都在从底盘性能、发射性能和稳定性入手研制步兵机器人。比如有的学校研制半舵半全向底盘构型；有的学校自制减速箱改良底盘驱动动力。在发射上，为了达到更强大持久的火力输出，许多高校尝试双枪管方案，在发射机构，拨弹机构上优化。而 17mm 发射进度更是老生常谈的研发技术点，本赛季风车规则更改后不止对视觉识别有要求，在机械发射精度要求上更是拉高了上限，有的学校在优化单发限位让弹丸对心，有的学校研究其他发射动力如气动发射。

1.2.1 步兵底盘构型的百花齐放

步兵底盘在 RM 创办几年来大致分为三种流派：麦轮，舵轮和全向轮。大家在新赛季中都不约而同的向舵轮或全向轮底盘投入研发资源。这种趋势是当下小陀螺功能的出现和视觉自瞄的发展所推动的。大家都在寻找新的底盘构型来使自己在赛场上更有优势。而传统麦轮由于结构简单较为轻便和其本身的构型使得麦轮在直线的通过性是略优于其他底盘构型，通常在飞坡特异化步兵上出现。下面我们针对其他学校全向轮底盘构型分析。

1.2.1.1 哈尔滨工业大学（深圳）南工骁鹰[1]

值得借鉴之处：

- ①平行四边形的摇臂设计，保证轮子悬挂压缩的稳定性

有待改进之处：

- ②自适应悬挂空间占用较大且分割局部空间，在设计其他部件的摆放位置时可能较难。

对电气元件的安放和维护影响较大

- ③轮组设计较为复杂，簧下质量较大。

1.2.1.2 南京理工大学 Alliance[2]

值得借鉴之处：

- ①全向轮轮子镂空轻量化设计

②轮组联轴器的简化设计和标准件的利用，降低成本

③独立悬挂

有待改进之处：

①四边形水平边长度不够，可能会影响轮组电机的更换

1.2.2 机动枪管加持下云台的千层套路

近两个赛季以来由于机动枪管的加入，很多学校开始研制双枪管云台。双枪管发射不止是测速模块的增加，对于整体的发射架构，高校间有许多优秀的方案。如西交利物浦的三摩擦方案，通过改变其中摩擦轮转动方向达到双枪管切换发射；还有哈工程切枪管方案，单发射机构能使云台整体轻量化；再有深圳大学竖置四摩擦轮单拨盘发射方案等等。想要在短时间内高爆发的输出则需要牺牲轻量化来安装两个发射装置以达到双枪同时输出。增加的机动枪管带来不同云台架构设计的分化，需要根据自身机器人发定位去决定设计方向。

1.2.2.1 西交利物浦大学 Gmaster[3]

采用横置三摩擦轮双拨盘构型

值得借鉴之处：

①切向供弹出口与发射弹链在同一直线上，弹链简单卡弹几率小。

有待改进之处：

①三摩擦轮设计无法双枪管同时发射，从而导致双拨盘设计冗余

②三摩擦轮横置导致云台横向空间占用较大且两个枪管之间相距较远

1.2.2.2 哈尔滨工程大学 Nooploop 创梦之翼[4]

切枪管方案

值得借鉴之处：

①切枪管方案是非双枪同时方案中最简洁的方案

②可复用传统发射云台的大部分设计

有待改进之处：

①切枪管的精准度，机构的稳定性还需要讨论

1.2.2.3 深圳大学 RobotPilots[5]

竖直双发射机构单拨盘方案

值得借鉴之处:

- ①上下摩擦轮构型能减短云台在水平上的空间占用
- ②发射弹链与拨盘出口在同一直线上, 不需要设计复杂的弹链结构

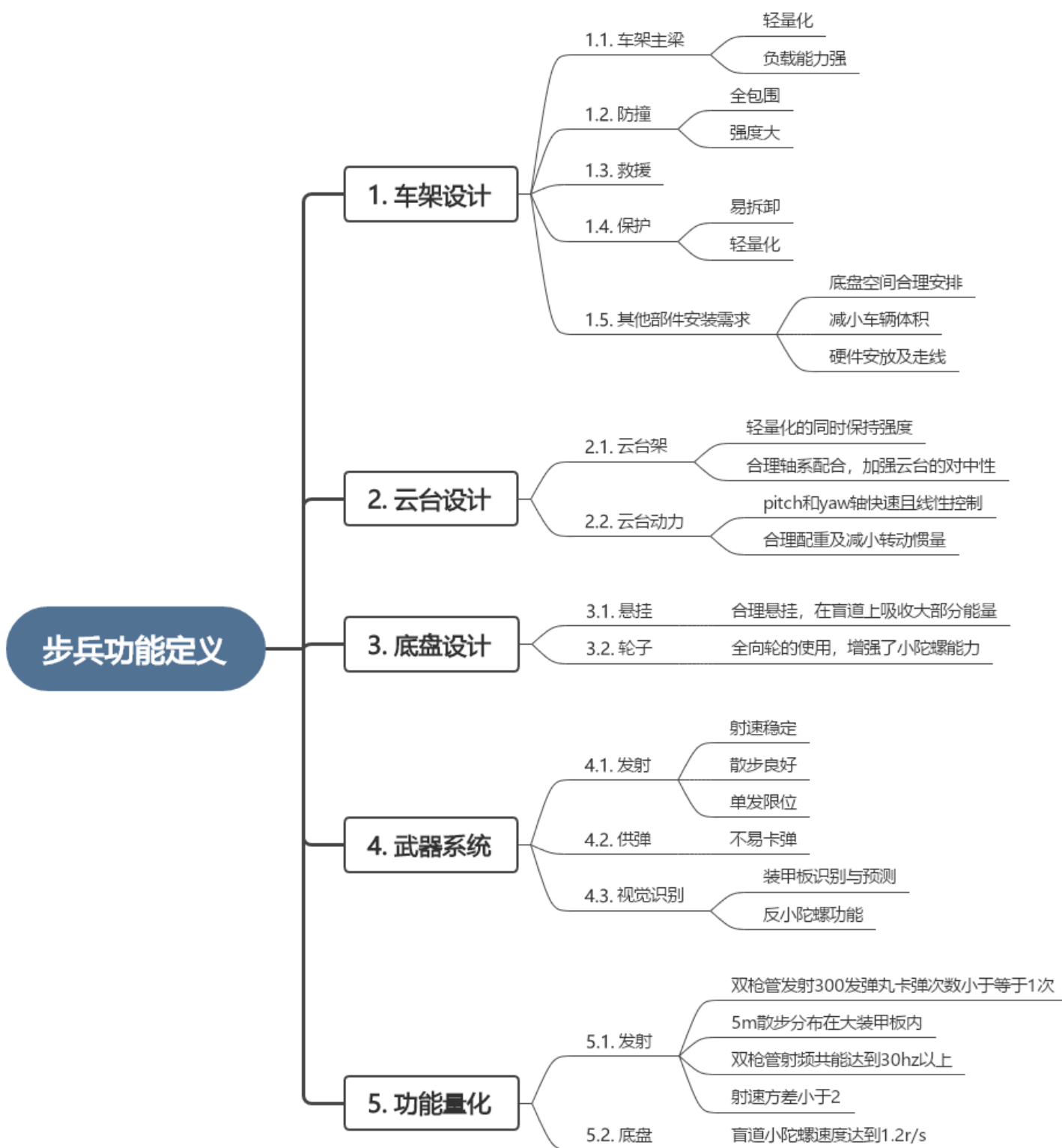
有待改进之处

- ①单拨盘构型无法同时双枪管同时发射
- ②由于拨盘尺寸限制, 枪管之间间距较大, 发射散步可能不理想

1.2.3 老生常谈的 17mm 弹丸发射精度

近两个赛季以来, 各个学校的 17mm 发射机构绝大部分都还是在使用摩擦轮发射方案, 也有少部分学校的 17mm 发射机构使用气动方案。在与其他学校的交流中, 我们发现大部分学校测试出摩擦轮发射方案中使用无枪管比有枪管的发射精度高, 因此无枪管也成为了摩擦轮发射方案发展的趋势。同时, 发射前的弹丸定心也成为了各个队伍发射机构的研发重心。不同的单发限位与不同的定心方案也将带来不同的发展方向。

1.3 机器人功能定义



<图 1.1：功能定义思维导图>

1.4 机器人核心参数

核心结构参数		
总重量	13.4kg（不含裁判系统重量）	
云台重量	4.5kg	
底盘重量	8.9kg	
尺寸	初始	584*584*491mm
	最大	584*584*491mm
轮距	346mm	
轴距	346mm	
重心位置（以整车中心投影地面为原点）	高 190mm；中心偏后 30mm	

<表 1.2：核心机构参数>

执行器件		
RoboMaster M3508 减速直流电机	4 个	底盘运动
RoboMaster GM6020	1 个	Yaw 轴运动
RoboMaster GM6020	2 个	Pitch 轴运动
RoboMaster M3508 减速直流电机（转子）	4 个	发射摩擦轮
RoboMaster M2006 P36 直流无刷减速电机	2 个	拨弹盘
MG996R 金属标准舵机	1 个	弹舱盖开合

<表 1.3：执行器件>

工作参数	
底盘（包括电机以及超级电容管理板，超级电容满电）静态功耗	3.0W
云台部分静态功耗	2.4W
发射部分静态功耗	8.0W
底盘最大移动速度	2.2m/s
底盘极限功率	110W
底盘正常工作电压范围	12V-24V
超级电容	10×100F 2.7V 电容串联，总能量 1458J
机器人额定电压	24V
云台自由度	Pitch 43.5°
	Yaw 360°

<表 1.4：工作参数>

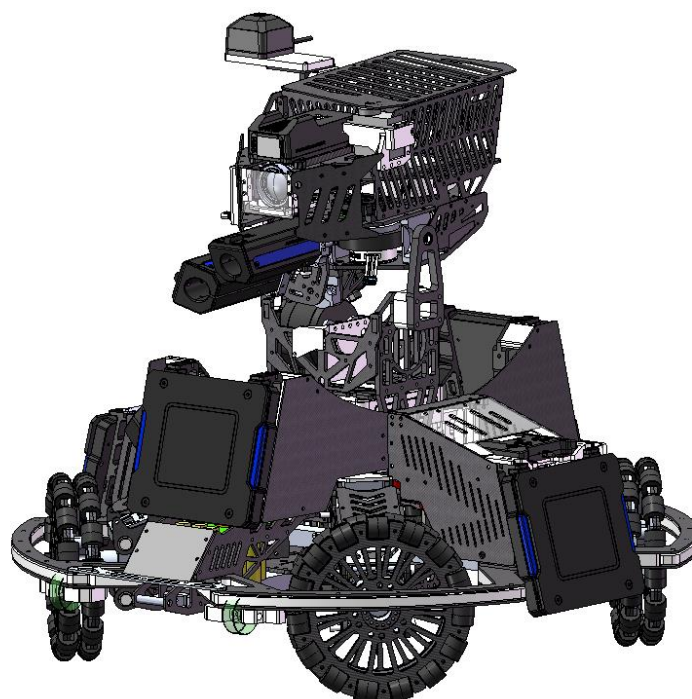
视觉工作平台及性能参数	
工作平台	nuc8i5bek
相机配置	海康相机 MV-CA013-A0UC + 6mm 镜头
工作环境	Ubuntu22.04 + ROS-humble
可跟踪最大速度	2.2m/s
可识别距离	0.3~4m
水平视角	45°
垂直视角	34.5°

<表 1.5：视觉工作平台及性能参数>

1.5 设计方案

1.5.1 机械结构设计

1.5.1.1 整车机械设计



〈图 2.1：整车装配图〉

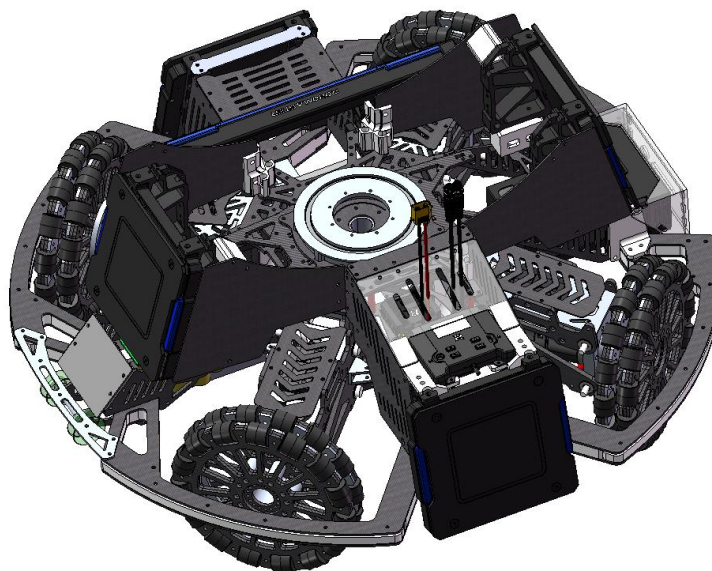
新赛季步兵机器人选用全向轮底盘构型，并尝试使用双枪管发射机构云台配合，以达到攻防一体的近距离特化步兵。底盘针对上赛季初版全向轮步兵暴露的问题进行集中改进，优化了轮组装配方式和增加了悬挂行程；重新设计轮毂大小和合理的镂空，增强步兵极速性能；同时对硬件安装走线也进行了一定的调整。云台设计上在不改变 pitch 轴高度的同时降低了 yaw 轴高度使其获得更大的俯仰角也相对降低了整车重心。

整车主要使用材料	主要使用需求
薄壁铝方管 (1.5mm, 20*15)	底盘车体框架的搭建，主要抗冲击部分
碳纤维板 (2mm、2.5mm、3mm、3.5mm、4mm、5mm、 5.5mm)	底盘结构支撑件，主要受力部分零件和转角连接件，减少整车质量

6061 铝合金 cnc 件	特殊关节零件
----------------	--------

<表 2.1: 整车材料使用>

1.5.1.2 底盘设计



<图 2.2: 底盘装配图>

步兵底盘我们选择以全向轮轮组为基础的十字形框架设计，相比传统麦轮，全向轮的优势有以下几点：

小陀螺消耗功率少，可以将极大部分能量运用在平移上，这能极大增加机器人的生存能力

在盲道场地上，麦轮小陀螺将会消耗极大功率且效果不好，全向轮因为其轮子摆放特性，在盲道上与传统麦轮步兵对抗有相应的优势

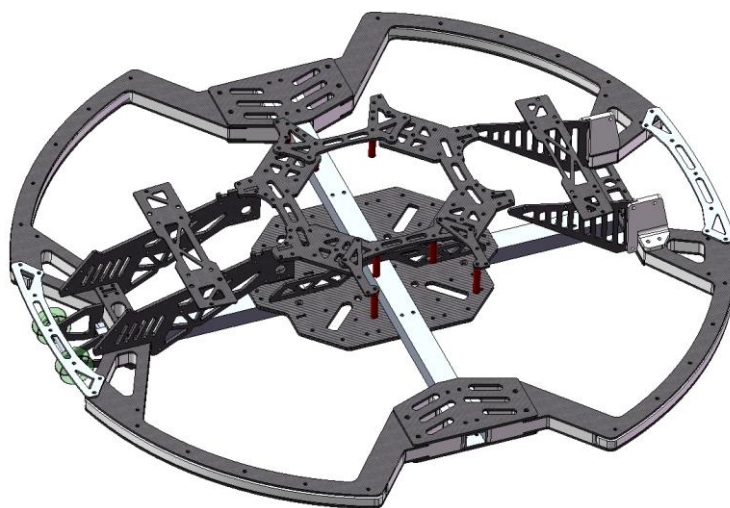
全向轮的轮子布置方式有利于结合外围圆形保险杠。圆润的外形能使机器人通过赛场的狭小空间，同时在被对面机器人卡位时，仍可以使用“小陀螺”手段防御。在赛场上常见到其他学校利用工程机器人卡位，步兵机器人输出的夹击战术，圆形的外围能大大提高己方机器人的生还能力。

1.5.1.2.1 框架设计

需求分析：

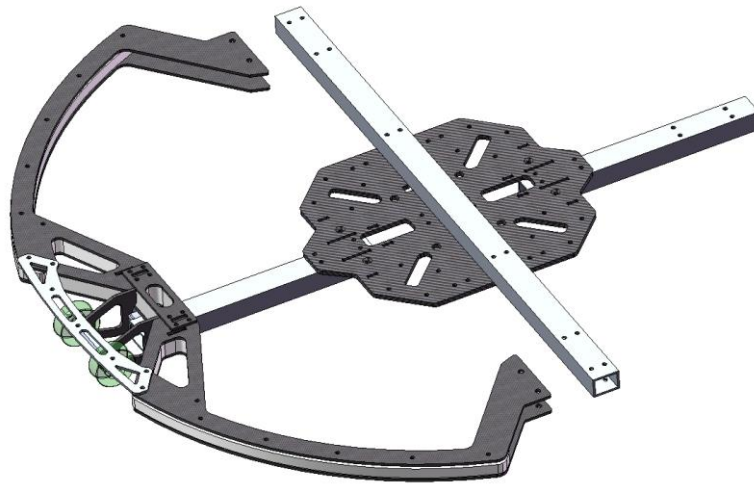
功能	需求
连接底盘机构	空间区分，便于模块化拆卸维护
安放电路元件	摆放逻辑和走线规划合理
承担总体重量	底盘框架稳定，刚度强度满足比赛需求
其他需求	
轻量化	

<表 2.2：底盘框架需求分析表>



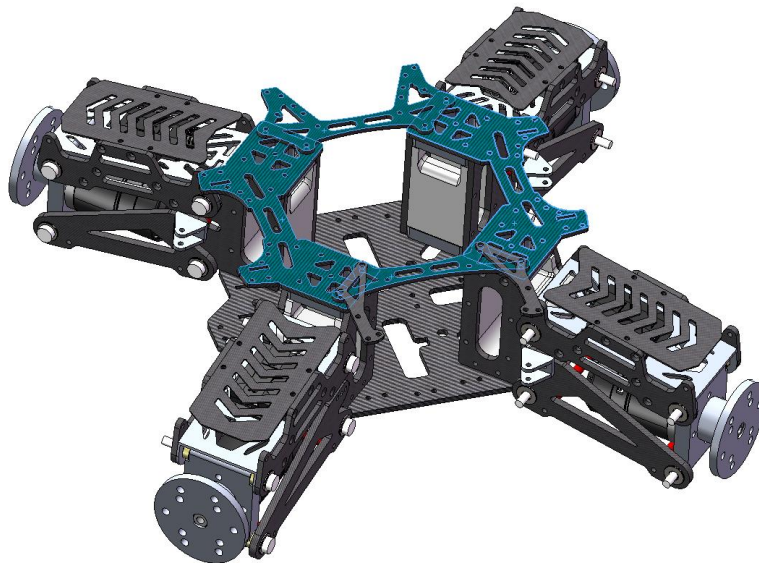
<图 2.3：底盘框架图>

底盘框架以轮组为连接划分为上下两部分。下部分以一块 3.5mm 的碳板为中心，搭建三根铝方延伸至外围防撞框架。简洁的搭建方式在分隔空间提高空间利用率的前提下合理的降低的整车的重心。在激烈的对抗中支撑外围防撞和飞坡导轮，提高整体强度刚度。



<图 2.4：下部分框架与部分防撞和飞坡导轮的连接图>

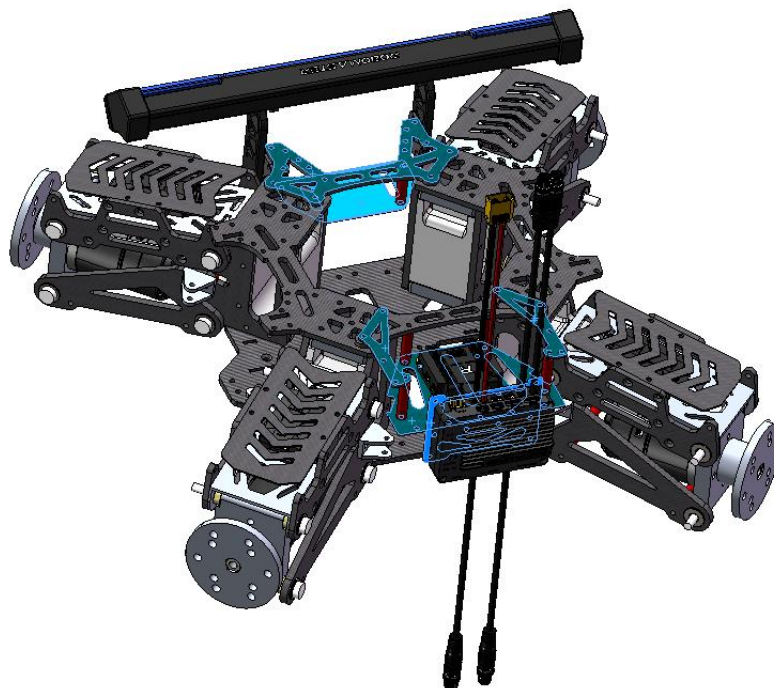
上部分框架主要由 4 块 3mm 碳板组成，相对一整块顶板的连接方式，分板有利于局部拆卸维护，提高了维护效率。同时在加工上相对于大镂空碳板，浪费的材料更少，成本经济。



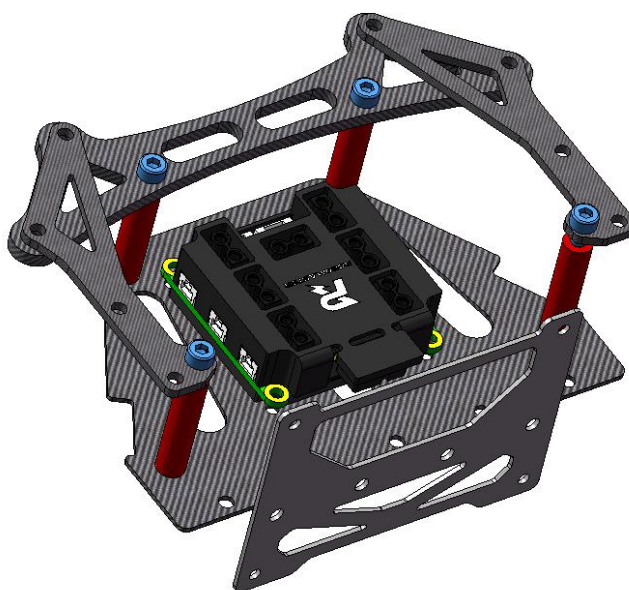
<图 2.5：上部分框架示意图>

在上框架中通过 2mm 碳板延伸作为“吊顶”将部分电子元器件通过长度 35mm 的铝柱“悬挂”在中部空间中（图 6）。此设计充分利用上框架和下框架中部的“灰空间”安放元件，并且使部分重要元件远离容易受到冲击振动的下框架，降低了元件因震动失效的可能。

除此之外，使用铝柱承担电路元件与框架中的连接，可以利用其螺纹特性实现电子元件“快拆”功能。在需要检查元件或更换元件时，只需要将固定铝柱的四颗螺丝拧松（图 7）即可将电子元件连同其固定板一同取下，此举利于元件模块化拆除，提高维护效率。



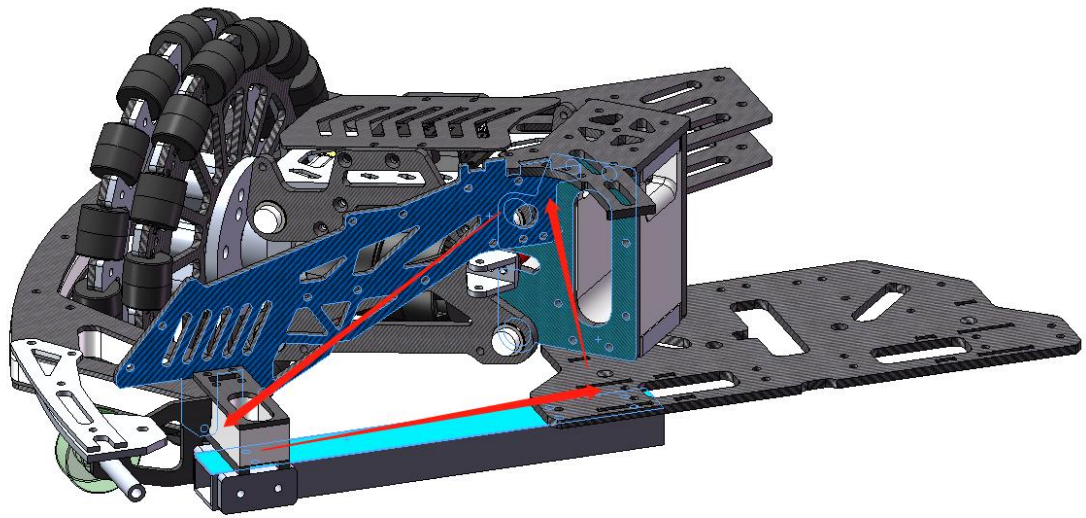
<图 2.6: 电子元件安放位置示意图>



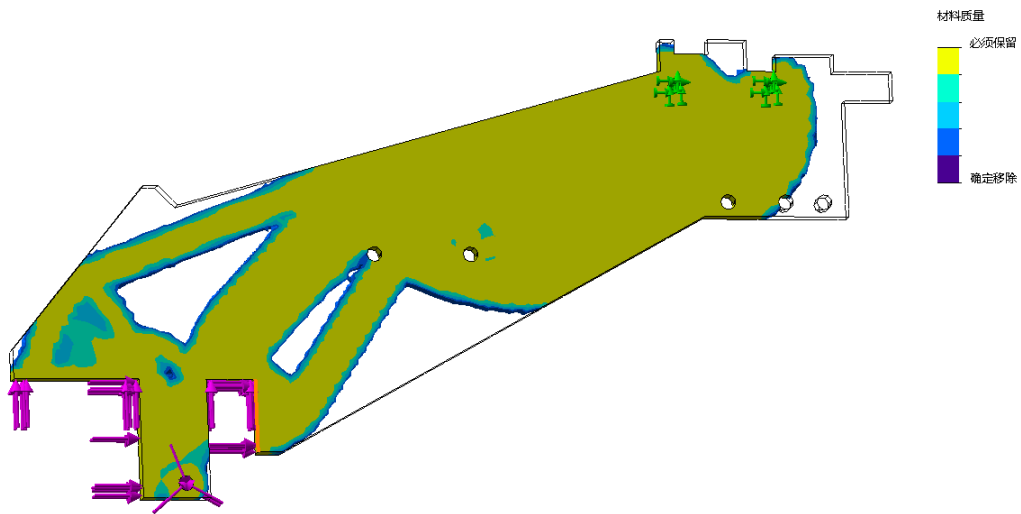
<图 2.7: 电子元件模块快拆示意图>

由于下框架在对抗或飞坡落地时容易受到较大的载荷，并且外围防撞只由铝方延伸部分进行支撑，其绕中心的力矩较大，铝方容易失效。针对此结构强度问题设计了上下框架连接承力碳板（图 2.8）。

加入承力碳板后底盘局部形成以碳板、铝方和轮组固定座三者形成三角形关系（图 9）在防撞外围受到载荷冲击时能极大减小铝方失效的可能，从结构设计上使整车受力分布更合理，提高了机器人整体连接强度刚度。

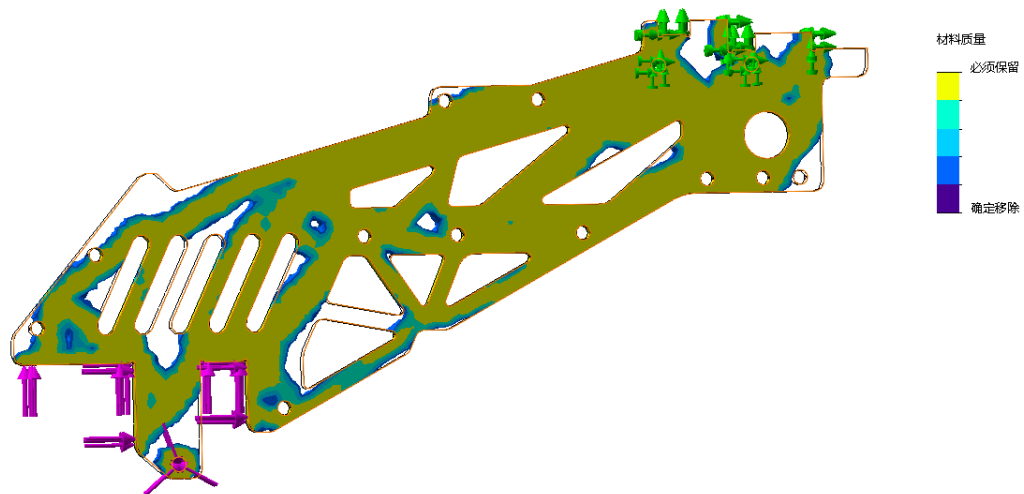


<图 2.8：承力碳板功能示意图>



承力碳板的拓扑优化算例

<图 2.9：优化前算例>



<图 2.10：优化后算例>

1.5.1.2.2 轮组设计

需求分析：

功能	需求
移动	零件配合空程小，轮组装配紧凑；通过性好
缓震	悬挂行程大；受力结构合理
其他需求	
轻量化	

<表 2.3：轮组需求分析>



<图2.11：轮组装配图>

1.5.1.2.2.1 全向轮

1. 减重设计：

轮子参照上赛季初版全向轮进行改良，主要针对轮毂的镂空优化。在新轮毂的设计上参考了南京理工大学开源文档进行较为激进的镂空，轮毂最终重量由上赛季的 113.35g 减重至 73.25g，减少重量约 35%。



<图 2.12: 旧全向轮装配图>



<图 2.13: 新全向轮装配图>

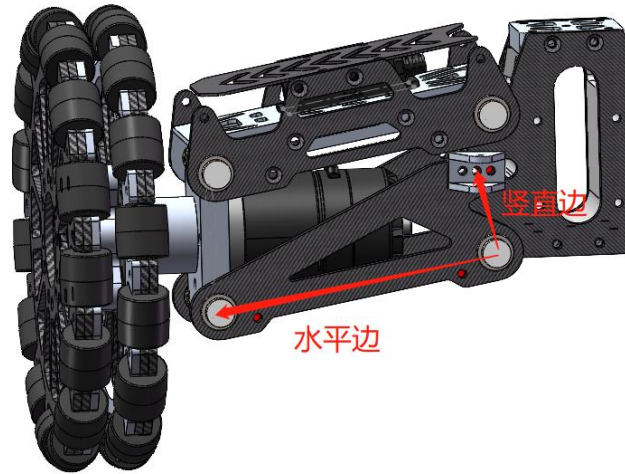
2. 轮径改变:

全向轮直径是 163mm，官方麦轮直径是 151mm。对于轮子的大小选择，其优劣特性是显而易见的。轮径小的轮子在轻量化方面优于轮径大的轮子，但是在场地通过性，极速性能是劣于轮径大的轮子的。我们在使用轮子的动力系统（大疆 3508 电机）时，发现机器人在起步时容易与地面打滑，这是由于胶轮与地面的摩擦力有限而远不及轮组电机的扭矩上限所发生的现象。所以我们在加大轮径相当于在电机减速箱上额外在加上一定的减速比，能使机器人起步时能充分利用电机的性能且提高极速时的峰值上限。

对于轮径增大带来的收益：通过性增加和极速性能的提高是符合我们对步兵盲道的通过性能和飞坡功能的优化期望。

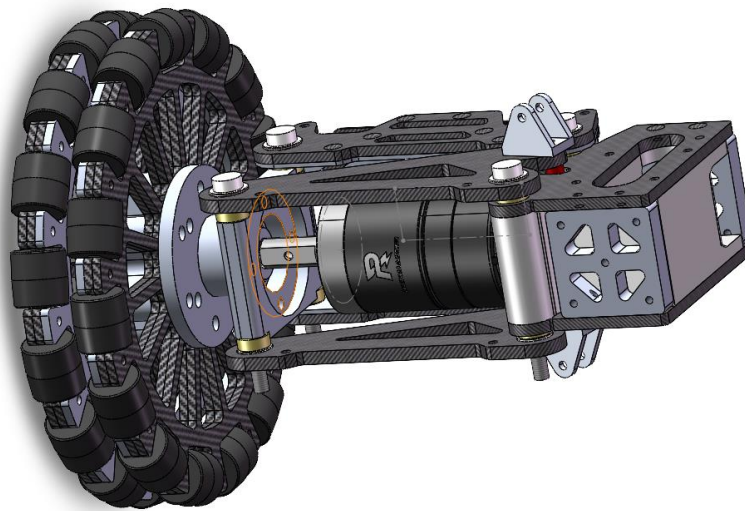
1.5.1.2.2.2 悬挂

悬挂方面我们并未改变初版的自适应悬挂布局，而是针对上赛季悬挂行程的问题进行优化。减短平行四边形悬挂的竖直边，加长水平边，在相同减震器压缩量时增加轮子上下摆动的竖直距离。



<图 2.14: 平行四边形悬挂示意图>

此外，增加水平边距离也方便轮组电机的更换，在不拆除轮组四边形结构就可以更换电机（图 16）



<图 2.15: 电机快拆示意图>

1.5.1.2.3 Yaw 轴设计



〈图 2.16: yaw 轴模块装配图〉

此前我们在步兵的 yaw 轴设计上都是采用 6020 抱紧环直接连接来实现的，6020 的轴承基本额定载荷为 2.2KN，最大径向载荷为 3.5KN；传统单枪管上供弹云台（装弹大约 3kg）经过计算测试在飞坡和盲道对电机径向载荷是远远达不到上限的。抱紧环直接连接避免冗余设计能简化步兵装配或达到轻量化的效果。

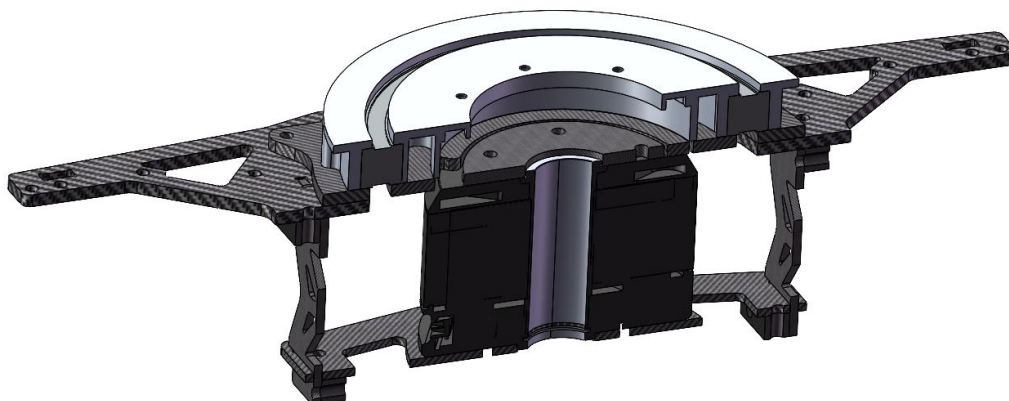
但由于今年的云台设计方案选择双枪管双发射机构双拨盘的构型，并且云台 pitch 采用双 6020 驱动，加上本赛季规则允许预装弹丸，最终云台整体重量较旧版云台大大增加（预装弹丸后达到 6kg）。在经过盲道颠簸甚至飞坡的时候，其云台自身重量对 yaw 轴电机的伤害是不可预计的，所以在底盘设计时需要重新考虑 yaw 轴构型。

需求分析：

功能	需求
承担云台重量	能在载重 8kg 云台保证结构强度和各方向刚度 保护 yaw 轴电机，减少对电机的径向载荷
连接云台和底盘	保证云台与底盘的对心度
其他需求	
轻量化	

〈表 2.4: yaw 轴模块需求分析〉

1.5.1.2.3.1 Yaw 轴构成



<图 2.17: yaw 轴剖视图>

在 yaw 轴的设计上，我们参考了很多学校的开源，在与电机转子的连接上我们学习上海交通大学步兵开源报告^[6]“香盘”的设计结构并进行优化设计，其替代的柔性联轴器功能，能在



一定程度上保护电机（图 2.18）。

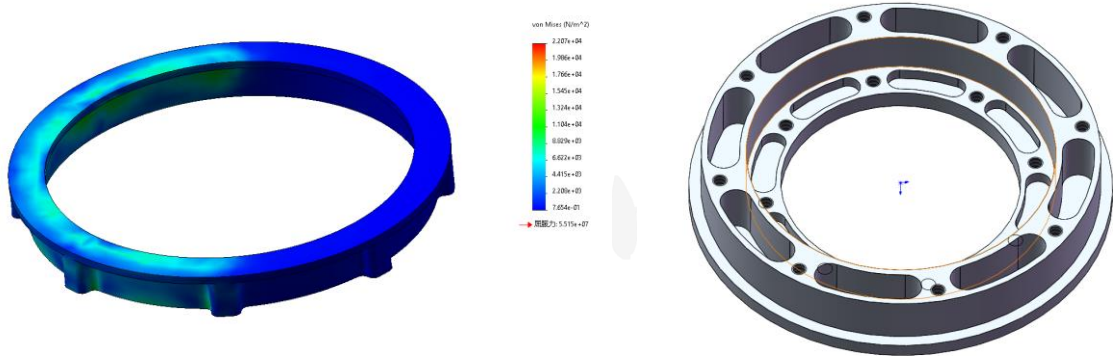
<图 2.18: 6020 转子连接碳板>

在轴承的挑选上，我们对比了交叉滚子轴承、角接触轴承和普通深沟球轴承。交滚是最多学校用的一个方案，但是经过考察发现交滚轴承在重量和价格上都远超我们的预期，薄壁交滚轴承（RA9800）的价格普遍也在 500 左右。而角接触轴承一般是成对使用，所以在轴承座的设计上可能需要添加一些设计，其轴承座加工工艺成本也会提高，此外，一对薄壁角接触轴承（KB020）也需要将近 300 的成本。

最终我们在分析 yaw 轴工况需求和整体成本预算考虑上选择了薄壁深沟球轴承（6816），

成本约在 50 元左右而轴承座的加工价格在 300 元附近，最终 yaw 轴整体重量维持在 460g 左右。

轴承座上我们结合经验和应力分析，发现用碳板设计轴承座在飞坡工况中不足以满足其刚度要求，其碳板形变量会影响整体结构的问题。最终我们选用 6061 铝合金机加工件承担轴承座的位置。



<图 2.19：轴承座应力分析>

<图 2.20：轴承座镂空设计>

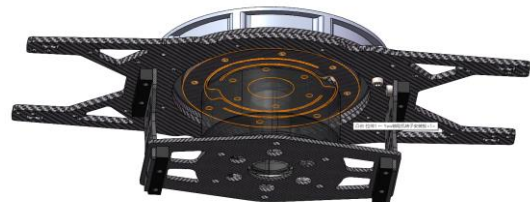
在 yaw 轴铝件轻量化设计上，我们进行了镂空而非镂空设计（图）。在保留薄壁结构基础上进行轻量化相比直接镂空其质量强度比是更优秀的，镂空处的箱体结构保证了受力分散，减少出现应力集中导致失效的可能性。

1.5.1.2.3.2 与其他部件的连接

轴承座通过 8 颗 m3 螺纹孔与云台连接，在四个方向上都有 3 颗螺丝连接固定，主要考虑在出现一颗螺丝失效的状况时，机构还能稳定工作，不会出现大面积的倾倒。



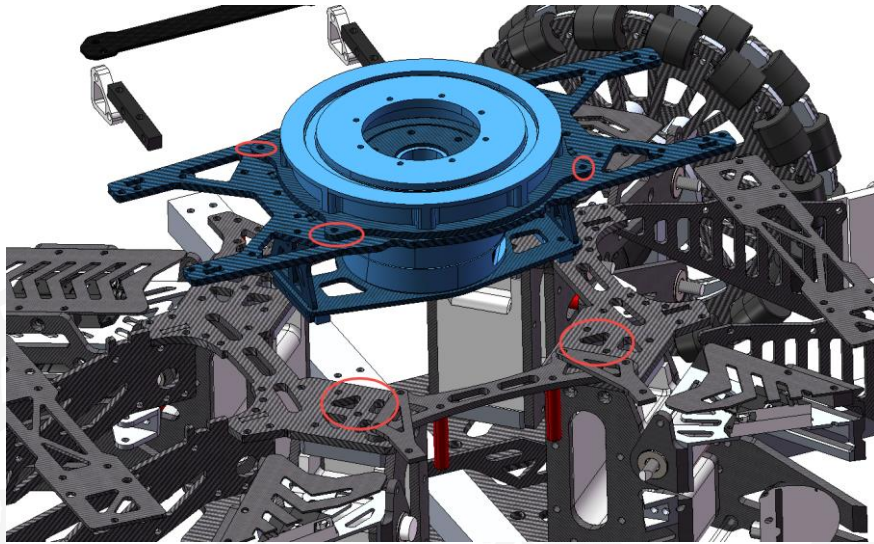
<图 2.21：云台连接螺纹孔>



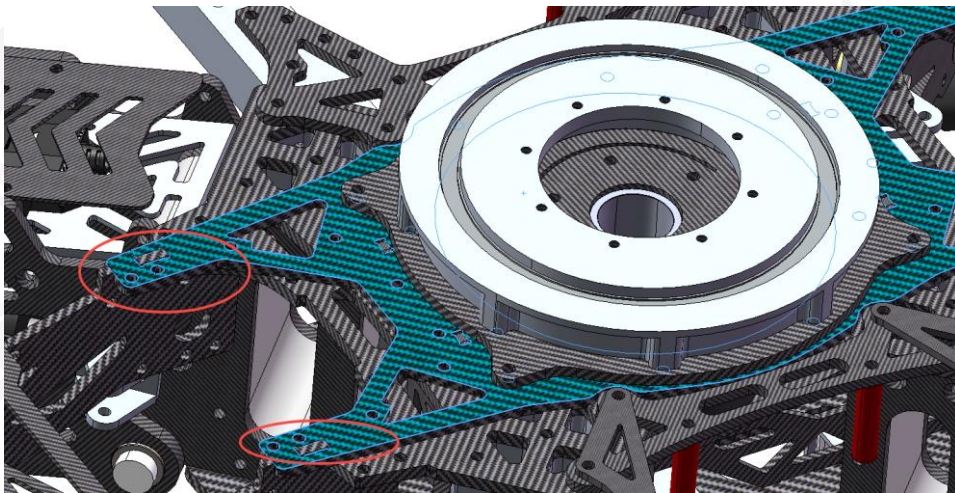
<图 2.22：轴承座连接螺纹孔>

设计中 yaw 轴直接通过与轮组关节座连接来固定在底盘上（图 22），最大程度保证云台与底盘的对心度。

为增加固定螺丝的可靠性，我们在 yaw 轴远端也设计有榫卯+螺纹的固定方式（图 23）进一步减小 yaw 轴侧翻倾倒的趋势。



<图 2.23: yaw 轴连接轮组关节座>



<图 2.24: 远端榫卯+螺纹孔固定>

1.5.1.3 云台总装设计

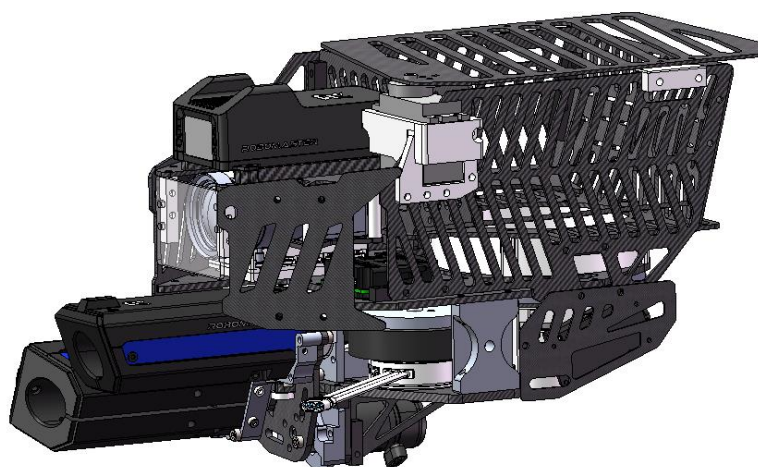


<图 2.25: 云台总装配图>

针对分区赛云台质量过大及质心分配不合理导致相应速度无法满足控制要求的问题，修改了其双发射构型并优化整体云台质量分布。采取切枪管的构型和 pitch 电机下置的方案使新云台能保持双枪管的火力优势也能保证良好的响应速度。

问题表现	原因分析	迭代方案
Yaw 的响应速度和响应效果无法满足控制的期望。 Pitch 电机容易发热。	1. 双发射双供弹构型增加了云台整体的惯量和质量。 2. 为满足 pitch 大载重需求采用双 6020pitch 方案，也大大增加 yaw 轴负担	1. 发射部分改成切枪管单发射模块构型 2. pitch 动力 6020 采用连杆传动，将电机下置内置 3. 整体结构优化空间布局
上发射机构控制难度大	上发射机构中线不与 pitch 轴交叉，使其在发射时后坐力会产生绕 pitch 轴的力矩影响 pitch 精度的控制。	采用单发射切枪管构型，无论使用哪根测速模块，发射机构都是相同的，保证了稳定性。

1.5.1.3.1 云台上装设计



〈图 2.26：云台上装装配图〉

云台上装设计是新云台最关键的点，需要重新思考空间布局和对单一机构处理使整体紧凑

设计功能	设计需求
无枪管发射	设计并测试出一版稳定，可靠的无枪管发射方案，可以满足单发限位的需求且发射散布在 5m 击打小装甲板命中率需达 100%，8m 击打命中率达 70 %。
切枪管机构	设计并测试出一版稳定，可靠的切枪管机构方案，可以保证测速模块与发射机构位置固定且在任意场景下不会发射蹭枪管现象。 机构实现简单，无多余冗余结构。
空间布局	Pitch 轴质心配平且质量分布于轴线附近，整体紧凑 相应机构快拆分离需求和电气仓便利维护需求

1.5.1.3.1.1 整体框架设计

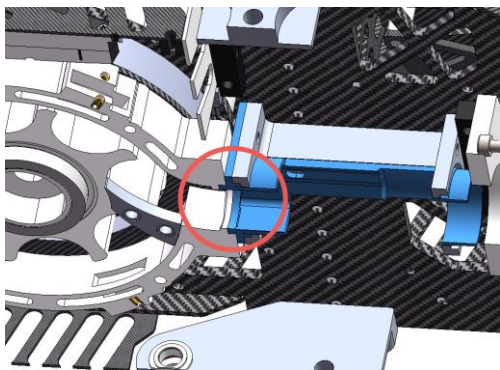
云台 pitch 以上部分主要框架是由三块水平安置的板，2mm 厚的图传相机固定板，4mm 厚的摩擦轮固定板和一块 3mm 厚的拨盘固定板夹两块 pitch 连轴铝件和一块发射模块铝件，侧面由两块 1.5mm 碳板使整体形成框型结构且作为部分保护和弹仓的一部分。整体框架连贯简洁且稳定，避免了多余的连接。



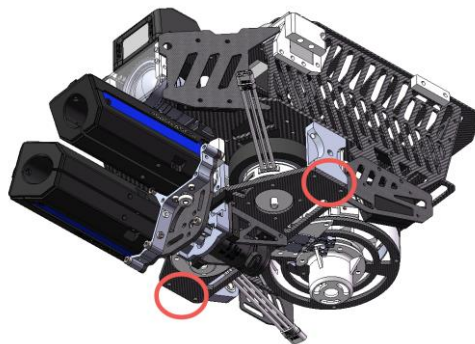
<图 2.27: 上装框架图>

1.5.1.3.1.2 模块空间布局与快拆设计

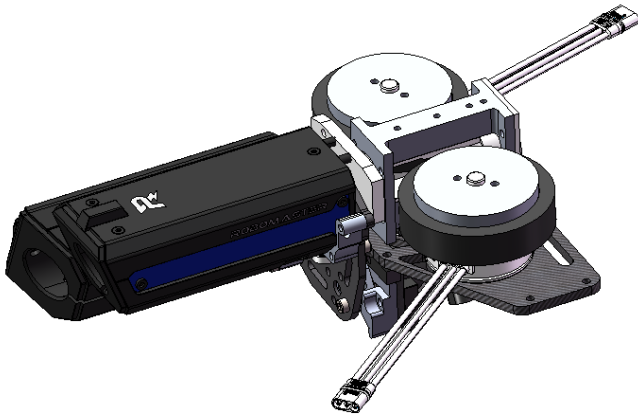
设计中将发射机构中摩擦轮枪管部分与拨盘部分用榫卯而非刚性的连接（图 2.28），使整体结构可实现一个快拆功能：以 pitch 连轴铝件为中心，仅需拆除四颗 M3 螺栓（图 2.29），即可将整个云台上装拆分为两部分，一部分是切枪管机构和发射摩擦轮（图 2.30），另一部分则是拨盘和电气仓（图 2.31），此种设计极大方便了单一机构如切枪管机构和弹链的维护和摩擦轮的更换。



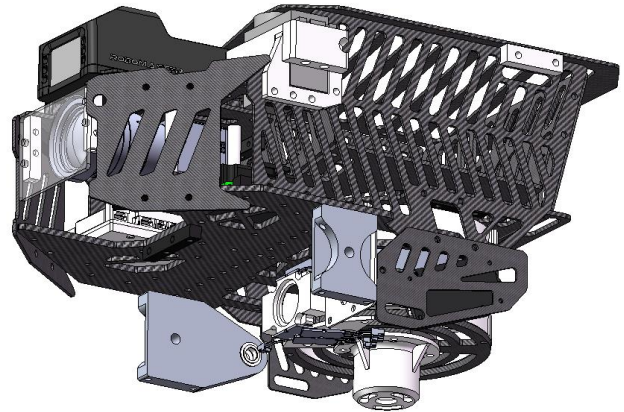
<图 2.28: 枪管和拨盘连接处榫卯示意图>



<图 2.29: 四颗 M3 螺栓示意>

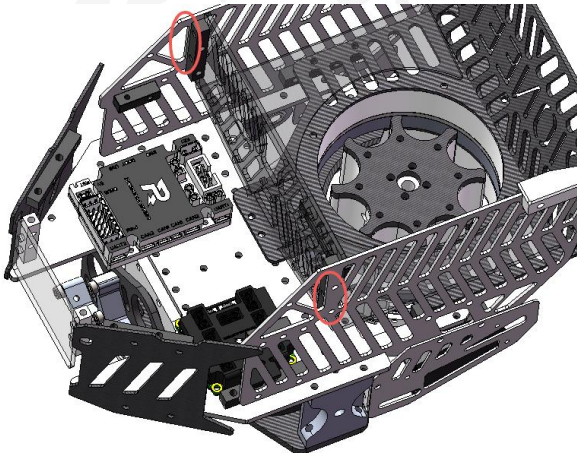


<图 2.30: 分离机构图_第一部分>

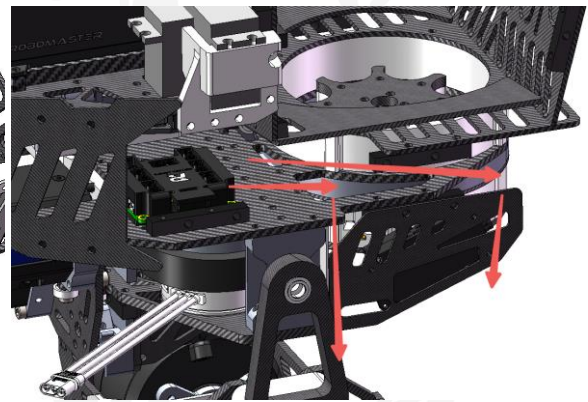


<图 2.31: 分离机构图_第二部分>

电气仓位于摩擦轮正上方，电路元件直接固定在 3mm 拨盘固定板上，因为该云台上装模块快拆的设计无需再设计电路安装快拆板，减少了多余的连接，做到简洁且轻量化。

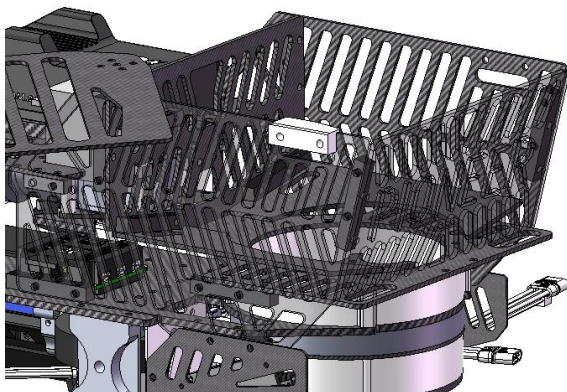


<图 2.32: 电气仓图>

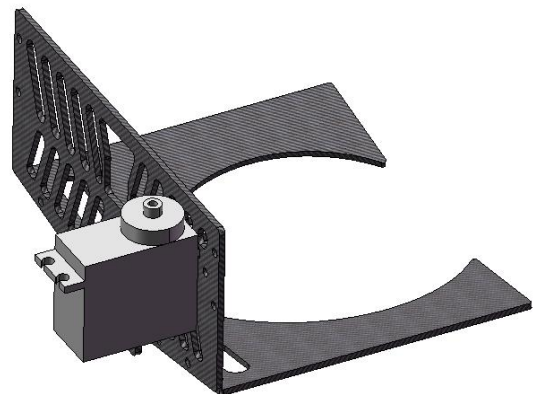


<图 2.33: 走线示意图>

云台电路走线如图所示位于弹仓斜坡缓冲板下方，真正意义上的利用了云台每一寸的空间。并且只用拆除弹仓前壁四颗固定的 M3 螺丝（图 2.32）即能将线材全部展现出来，极大方便于维护及更换元件，避免了电路元件走线的“一次性”装配。



<图 2.34 弹仓示意图>



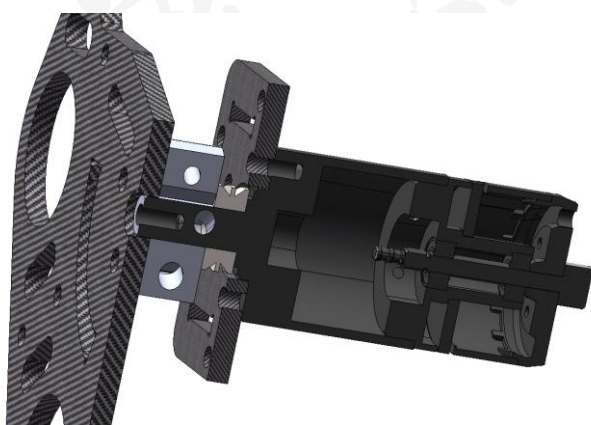
<图 2.35 弹仓前壁拆除模块示意图>

弹仓为储量在 250+颗弹丸，根据赛事观察和以往经验，载弹量满足我们在 RMUC 中一次进攻最大的弹丸使用量。

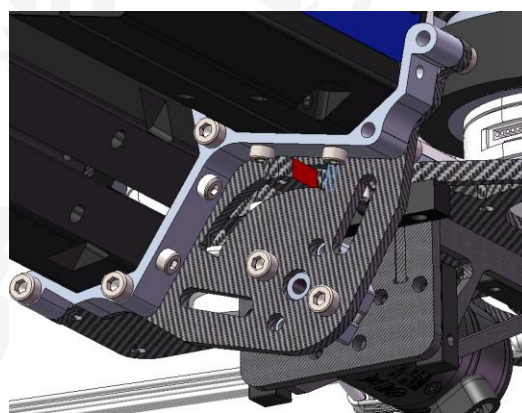
云台整体空间布局紧凑，且大质量元件都处于 pitch 轴附近，极大优化了 pitch 轴的转动惯量，最终实现效果：载弹量在 160~满弹（约 260 发）区间，pitch 控制净差为 0，满足控制使用要求。

1.5.1.3.1.3 切枪管机构

切枪管机构采用 2006 直驱旋转切换测速模块，直驱轴系简单，采用推力轴承和上海交通大学开源的 2006D 孔联轴器组成，简洁实用，D 孔抱紧的设计减少了空程。



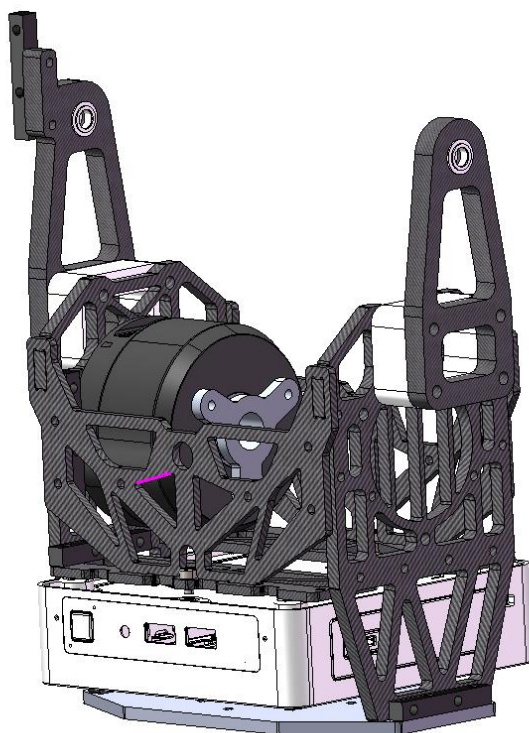
<图 2.36 驱动轴系设计示意图>



<图 2.37 机械限位示意图>

切枪管旋转的机械限位如图所示，旋转过程中铝板中槽的两壁正碰 4mm 碳板以达到机械限位的目的。

1.5.1.3.2 云台架设计



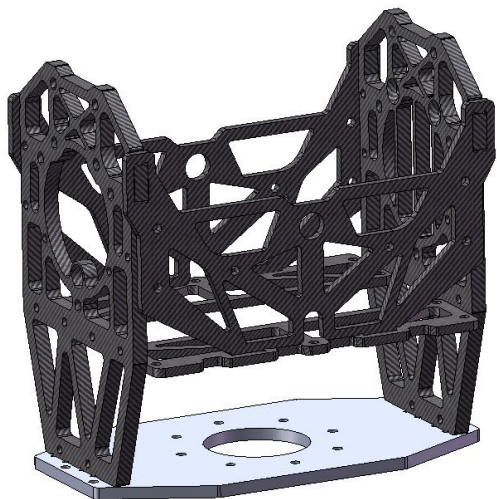
<图 2.38：分离机构图_第一部分>

云台架作为第一关节，承载末端执行器（发射机构）与底盘系统的连接，因为竞赛中机器人对抗强烈且“小陀螺”功能的广泛使用，要想取得优秀的发射精度对云台架有比较高的一个稳定性要求。

功能	需求
连杆传动 pitch 电机动力	连杆的刚度和强度设计要满足传动需求
连接和承担发射与底盘的连接	刚度上要在发射机构任何状态和底盘运动任何状态时保持发射部分与底盘位置的相对稳定，以保证发射精度的效果。 强度上需要在机器人可能使用的任何场景保证结构不会失效。
承载器件的安放	空间布局合理，将质心集中于 yaw 轴线上，且整体质量下置。

1.5.1.3.2.1 整体框架设计

云台架整体设计为满足部分元件摆放需求和云台整体结构强度需求，采取两块 5mm 侧板并用两块 3mm 加强筋使整体受力连续形成稳固的框架形式，板材连接采用有 4 面螺纹孔的标准铝件连接，通用标准零件的大量使用降低了成本支出。



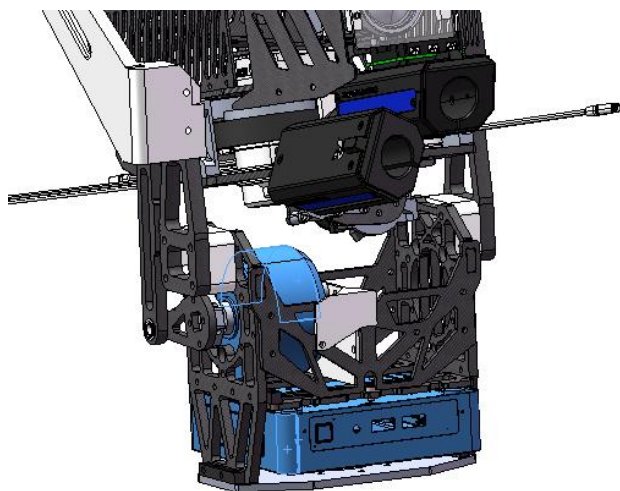
<图 2.39: 云台架框架装配图>



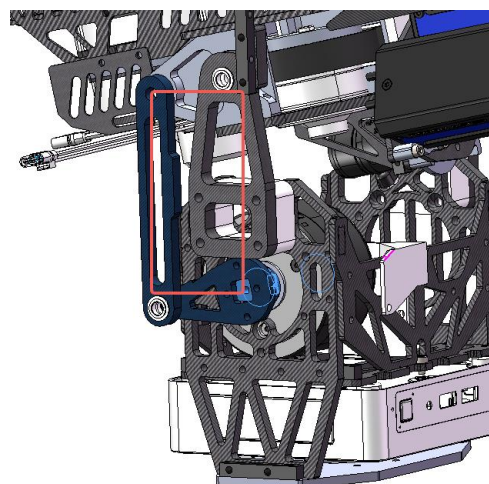
<图 2.40: 标准连接铝件>

1.5.1.3.2.2 元件空间布局

为保障质量分布趋于转轴中心同时降低重心，设计中将大质量元件都放于云台内侧和底部，如视觉运算平台 NUC 位于云台中心最底端，作为 pitch 动力的 6020 电机也放置在云台架内侧底部，且通过连杆驱动上装运动



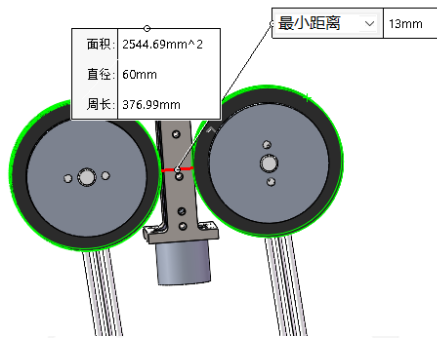
<图 2.40: NUC 和 6020 电机安装位置>



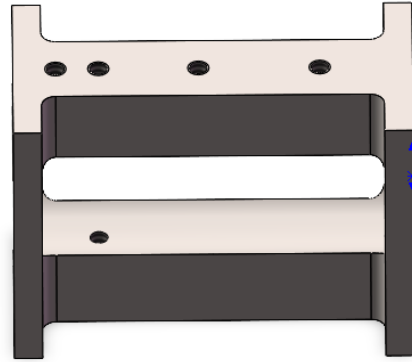
<图 2.41: pitch 轴驱动方案>

1.5.1.3.3 发射机构设计

为提高弹丸速度，我们重新调整了摩擦轮间距与摩擦轮包胶硬度。经多次测试，我们发现当摩擦轮最小间距为 13mm、摩擦轮邵氏硬度为 55 时弹丸速度较高、初速方差较低且散布较低。



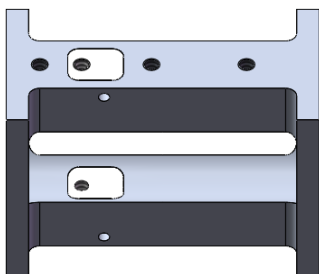
<图 2.42: 摩擦轮间距>



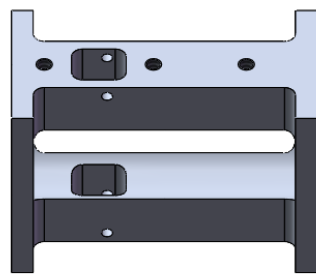
<图 2.43: 单发限位零件>

本赛季暂时采用上下各单波子螺丝单发限位，该限位具有可调节、具有一定行程及较为耐用等优点，为发射机构减少卡弹次数，降低弹速方差与提高精度起到重要的作用。接下来我们将测试上下双波子螺丝单发限位，旨在为单发限位增加弹丸定心功能，进一步提高精度减小误差。

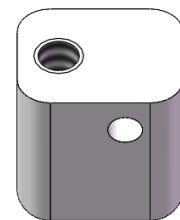
由于新单发限位造价为 196 元对我们队伍来说较为高昂，为此我们决定将旧的单发限位进行改造以重复利用，在节省经费的同时便于下一代单发限位的设计测试。具体方案为在原有 cnc 件基础上再专门设计一相吻合的 cnc 件组合装配，前一 cnc 件负责与枪管与管道的连接，后一 cnc 件负责安装单发限位以实现各项功能。



<图 2.44: 单发限位装配>



<图 2.45: 摩擦轮处枪管>



<图 2.46: 单发限位零件>

经过多次多距离测试，我们发现无枪管发射的精度较有枪管发射要高，我们分别在 3.5m、5.5m、8m 处使用同一固定发射机构进行了多轮散布测试，发现有无枪管发射在 3.5m 距离上无明显差距，在 5.5m 处有枪管散布范围较无枪管发射散布范围多约百分之二十，在 8m 处有枪管散布范围较无枪管散布范围大约百分之三十，为此本赛季决定选用无枪管发射方案。

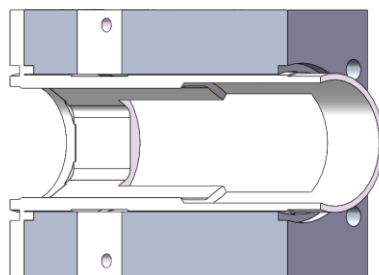


<图 2.47：无枪管装配图>



<图 2.48：测速模块安装图>

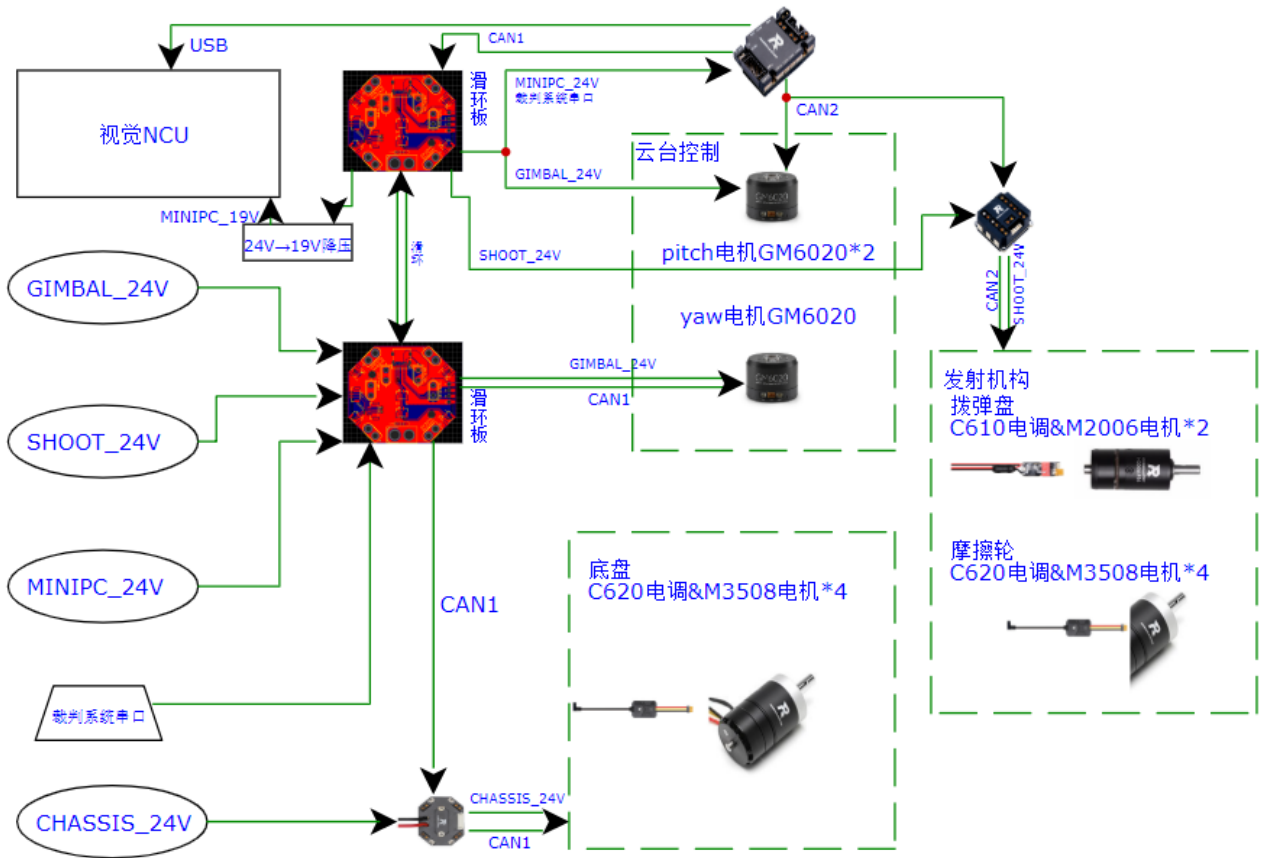
为了进一步提高发射精度，在上一版无枪管发射的基础上增加了定心的功能，同时为降低改造成本，选择嵌套定心光固化打印件的方案实现定心功能。经多次测试，最终发现 17.1mm 为最佳的定心间距，过大无定心效果，过小会弹丸卡死管道。



<图 2.49：无枪管定心示意图>

1.5.2 硬件设计

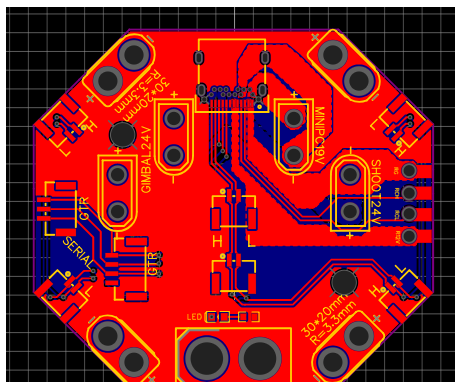
1.5.2.1 整机硬件框图



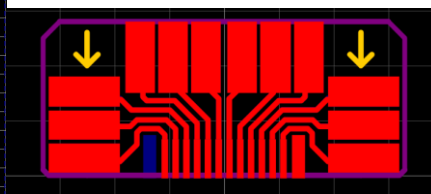
<图 3.1：整机硬件框图>

1.5.2.2 滑环接口

步兵云台采用 MOFLON MMC397 24 线滑环，滑环线束数量多并且线束细，同时还需要经过 GM6020 电机通孔，我们在上一个赛季的自制滑环板上进行改进，制作了新版本的滑环快拆接口。快拆接口通过魔改版本的 type-C 接口连接滑环和分线板（图 48、图 49）



<图 3.2：滑环板硬件图>



<图 3.3：type-C 接口分线板>

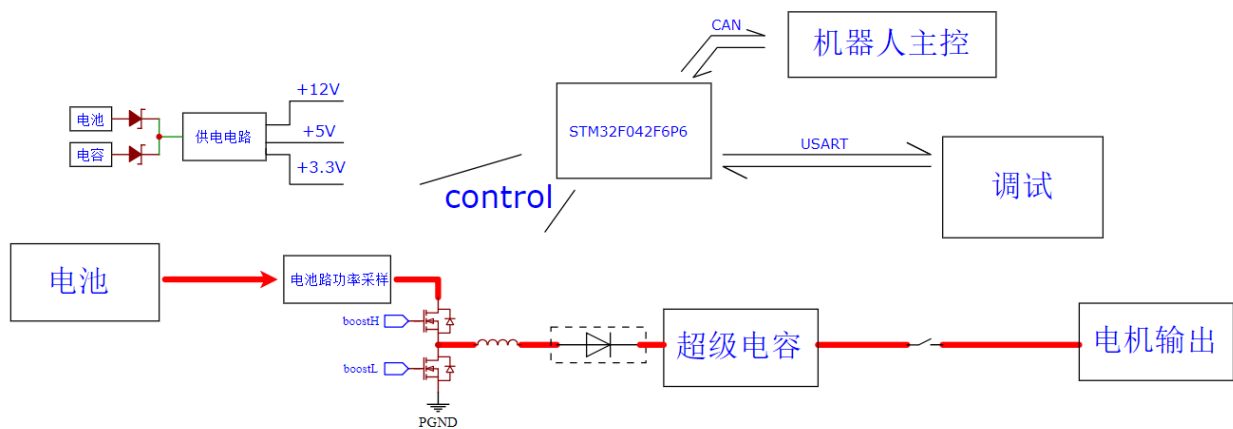
滑环板的母板上焊接 type-C 母座，子板为 type-C 公头夹板与滑环线连接，通过将滑环的 24 线焊接至小板上即可获得通用的接口，云台上下可以采用完全对称的 PCB 底板供小板接入，制作简单。拔走滑环，再更换一个新的带有同样接口的滑环即可。

Type-C 在修改后虽然无法具有正常 type-C 的稳定性，但是 24pin 引脚可以满足 24pin 滑环线的数量要求。Type-C 接口设计上可以通过 5A 电流，可以满足功率线过流要求，同时连接可靠，不易损坏以及短路。采用前插后贴版本的 type-C 母口相比立帖版本可以降低焊接难度、增强稳定性且节约空间。与滑环线连接的夹板在设计上留够了滑环线连接的焊盘，并且大小能通过 GM6020 电机通孔。

1.5.2.3 超级电容

1.5.2.3.1 超级电容控制板硬件设计

本赛季中期曾尝试研究过 buck-boost 并联型三端口电路的功率板设计，然而因人手有限、维修成本大等诸多原因，我们不得不放弃这个设计。直到分区赛打完，我们才开始着手回退之前用的功率板方案，硬件与代码均为自己重新设计，以下为电路拓扑结构：



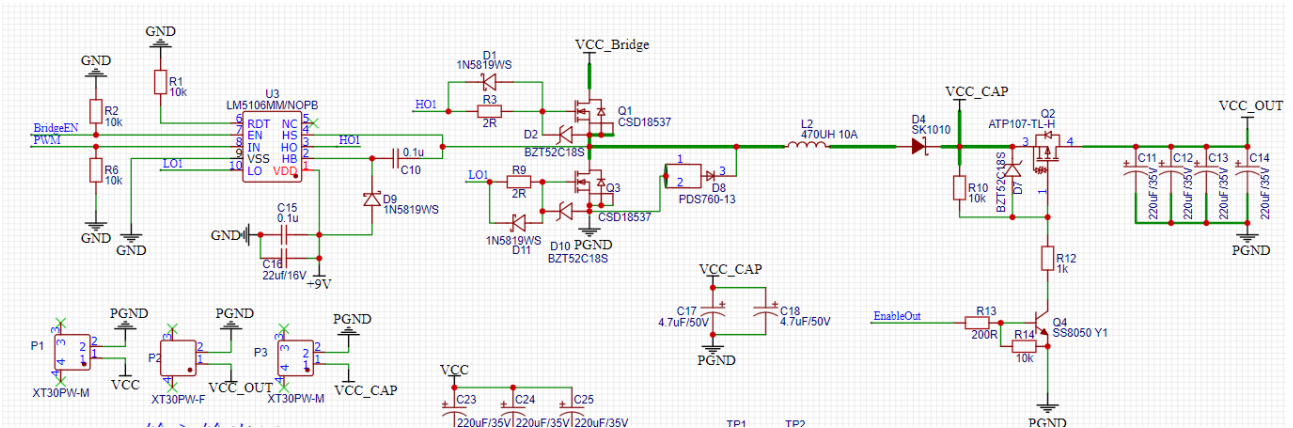
<图 3.4：电路拓扑结构>

大致工作原理：单片机控制 buck 半桥输出电压，当输入功率过大时就自动降低输出电压，保证输入功率不会超过设定功率，此时多余的功率就将由电压较高的超级电容提供。

考虑到时间紧急，硬件方案全选用验证过能稳定可靠使用的芯片，具体如下：

Buck 半桥部分：采用 CSD18537 加 LM5106 搭建，以搭建稳定的电路。

电流采样：采用了共模电压可达 80V 的 INA193 芯片搭建。

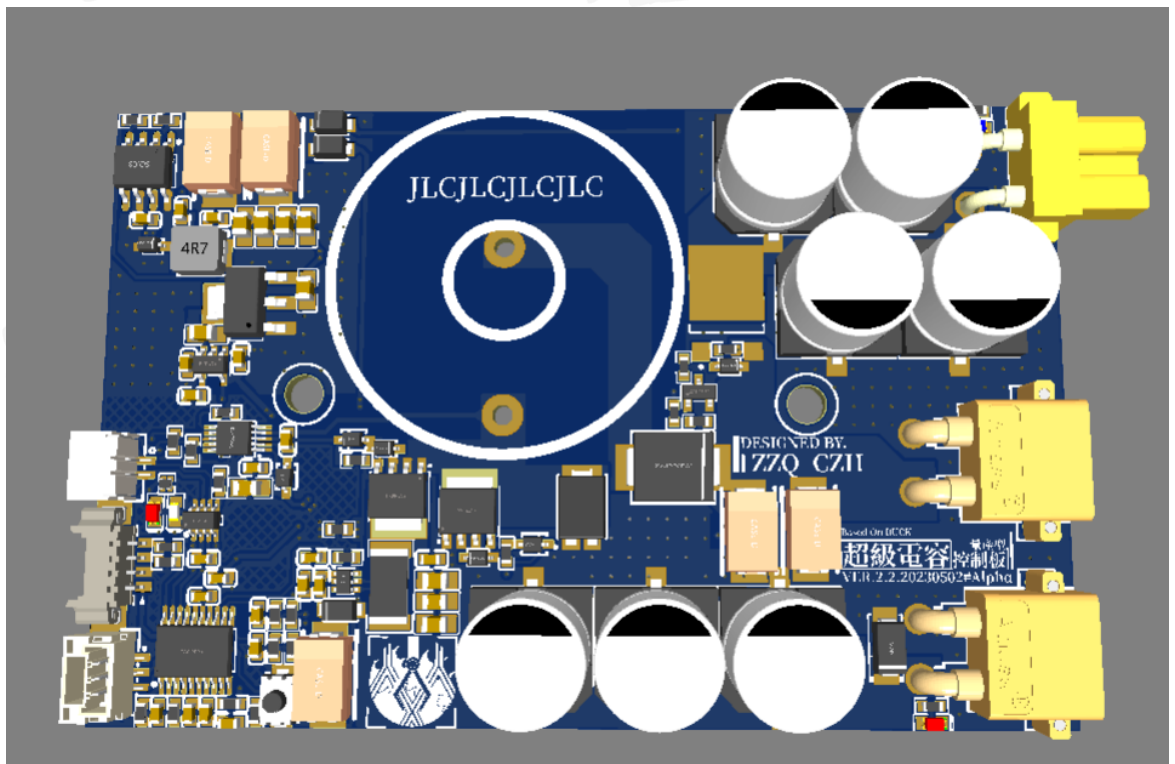


<图 3.5：整体电路图>

在设计过程中，部分元件参数我们直接参考了雾列科技的设计，以确保板子的稳定工作。

最终实测板子在连续十分钟 28V 降压 12V 稳定 95W 输入，输出间断施加 200W 负载工作的情况下，充电效率约有 92%，期间除了输出关断 PMOS 因内阻较大温度升至约 80° C 外别的器件并无明显温升。

最终设计出的成品 3D 渲染图如下，总板成本控制在 ¥150 左右。

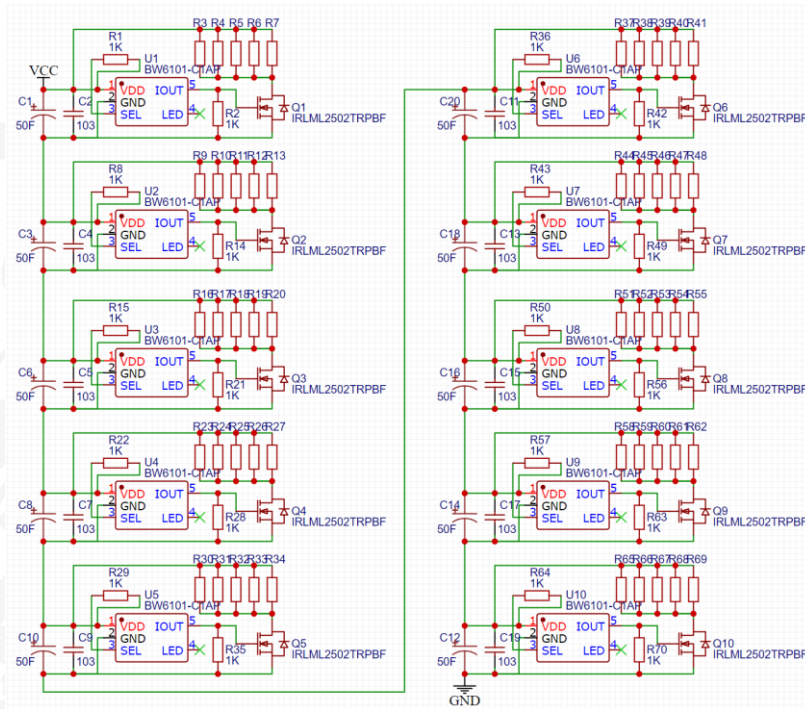


<图 3.6：控制板成品渲染图 >

1.5.2.3.2 超级电容保护板硬件设计

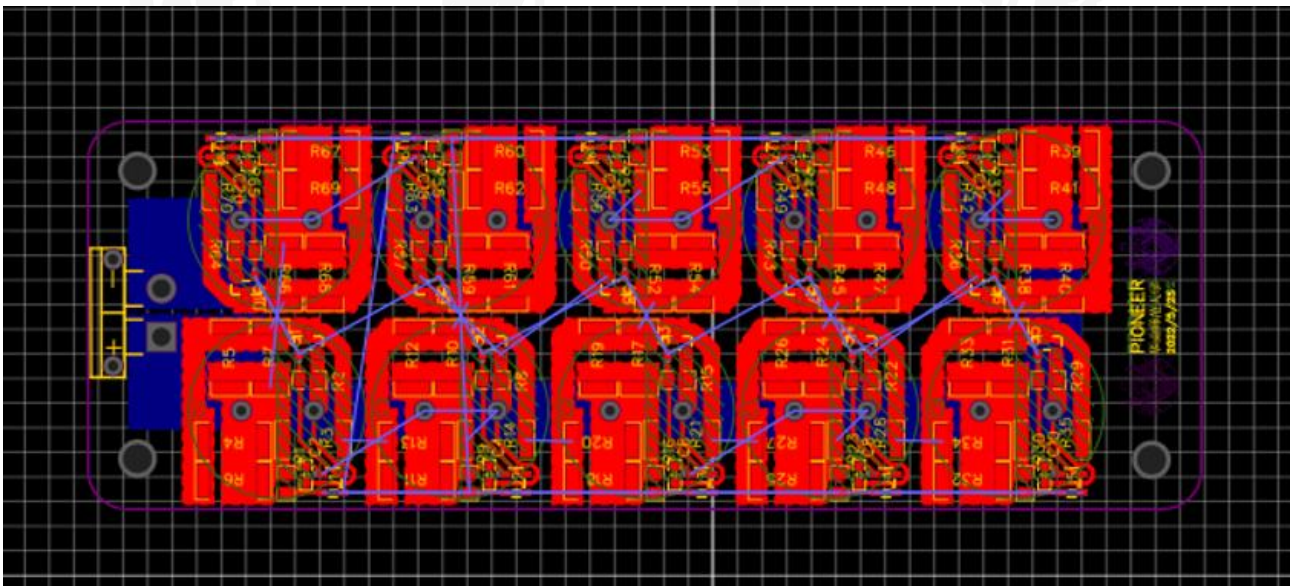
超级电容组使用了 10 个 100F 2.7V 的黄金电容串联组合成为大容量的超级电容，等效 10F 27V 的超级电容，在电容两端的电压大约为 22.68V，使用的最低电压也为 13.5V，通过公式 $P=1/2*C*(U1^2-U2^2)$ 计算可得出改装后的超级电容大约有 1660J。

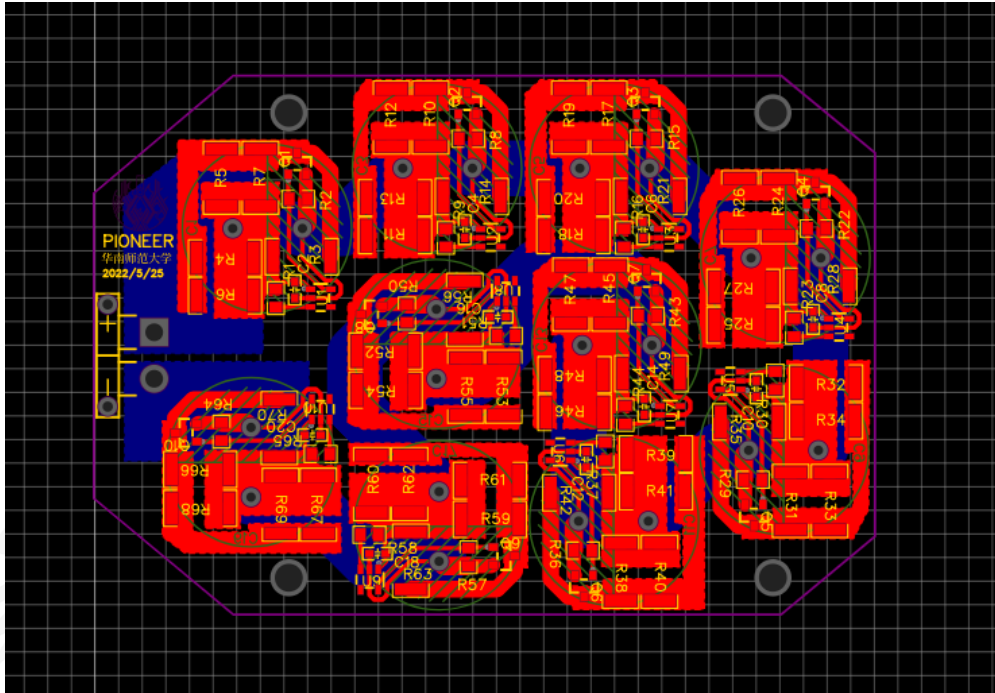
为了防止充电过程中因各电容容值不同而充电分压不同，进而导致的单个电容过压损坏，需要设计均压保护电路，以下为电路原理图。



〈图 3.7：电路原理图〉

电路使用 BW6101 芯片，当检测外部电压超过 2.7V 时，芯片的 Iout 引脚输出高电平驱动 nmos 自动对电容进行电压泄放。





<图 3.8：两种电容组的 PCB 板>

由于上边的电路具有很强的独立性，可以进行任意拆分和组合，因此为了方便机械部分的设计，我们还将 PCB 板可以设计为不同形状。

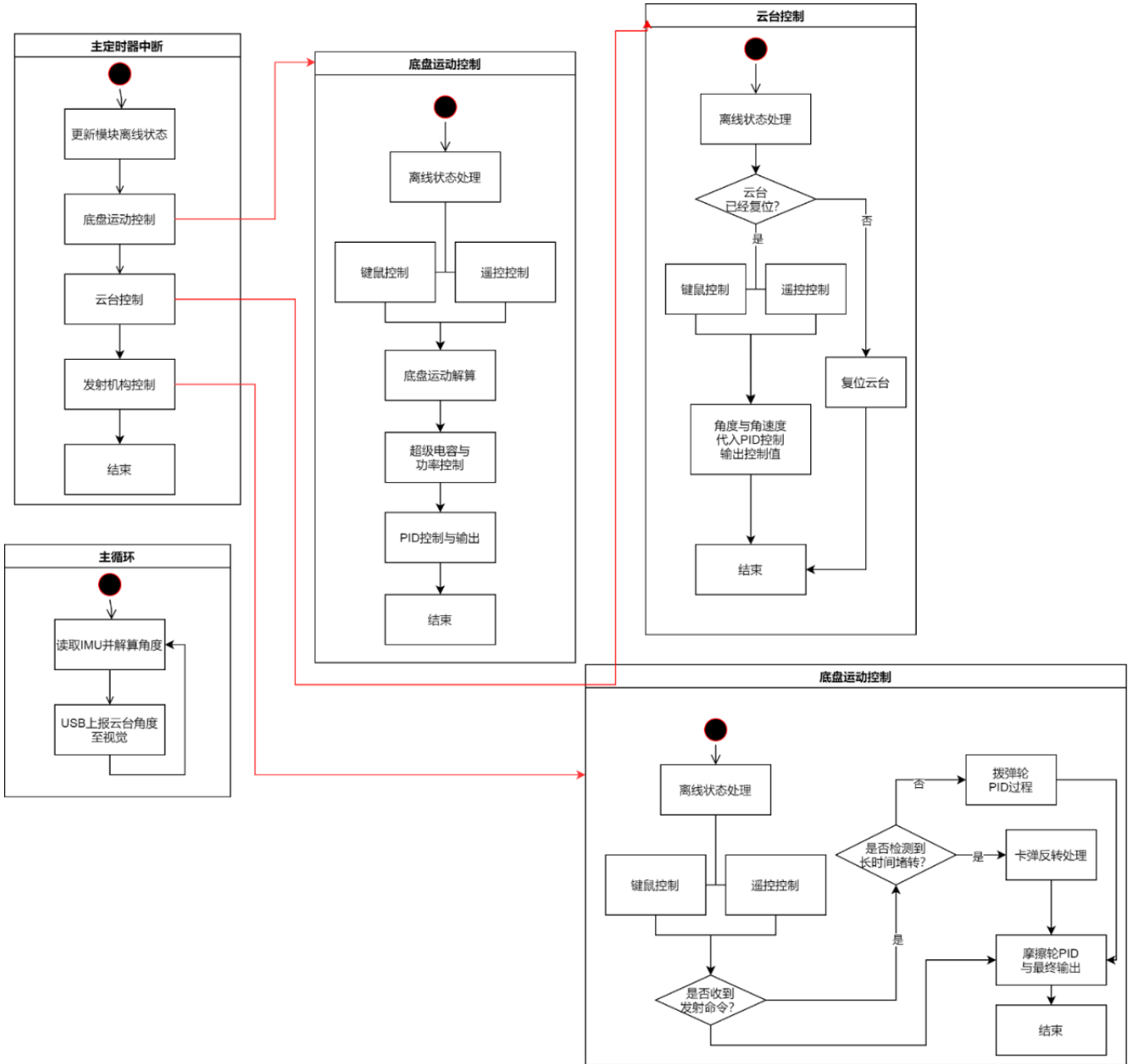
1.5.2.3.3 超级电容控制板控制方案

由于硬件输出上添加了二极管防止电流倒灌，同时输出还会挂载着一个容值巨大的超级电容组，所以代码的实现也较为简单，主要的控制逻辑仅需一个 PID 做功率环即可。

值得注意的是，由于输入功率与设定给超级电容的充电电压存在一层积分的关系，所以 PID 参数必然是以 I 积分项为主，但过高的 I 会导致控制系统响应较慢，当功率板空载时突然接上超级电容组很容易出现“打火”的现象，严重的就会烧毁功率路器件。为此我们代码中设计了通过检测输出功率来对输出是否开路的判断，在输出情况变动时将自动切入输出电压缓启动的模式

1.5.3 软件设计

1.5.3.1 系统流程图



<图 4.1：系统流程图>

软件架构：定时器中断（200Hz）运行控制，主循环内更新 IMU 与视觉数据。

开发调试环境：Keil 为主， OpenOCD 调试为辅。

1.5.3.2 软件框架

采用自主编写的电控软件框架，配合相应说明文档，各车之间易于移植。



<图 4.2：软件框架示意图>

其中框架代码的目录树以及其用途解释如下，base_drv 下的各个模块可通过 drv_conf.h



文件配置

<图 4.3：移植配置指南>

--robot_generic

框架主体

	bus_detect.c	模块离线检测逻辑
	bus_detect.h	
	chassis.c	底盘控制逻辑模块
	chassis.h	
	gimbal.c	云台控制逻辑模块
	gimbal.h	
	linear_motion.c	线性移动控制逻辑（仅工程用）
	linear_motion.h	
	pid_parameters.c	PID 参数保存位置
	pid_parameters.h	
	power_ctrl.c	功率控制逻辑模块
	power_ctrl.h	
	robot.c	机器人功能总成
	robot.h	
	shoot.c	发射控制逻辑模块
	shoot.h	
	vision.c	视觉接口
	vision.h	
	algorithm	常见算法集合
	crc.c	官方 CRC 校验算法
	crc.h	
	filter.c	常见滤波算法（IIR, 均值）
	filter.h	
	imu_fusion.c	六轴 IMU 解算算法
	imu_fusion.h	
	imu_calibrate.c	IMU 校准算法
	imu_calibrate.h	
	kalman.c	卡尔曼滤波算法
	kalman.h	
	pid.c	PID 算法
	pid.h	
	pid_ladrc.c	LADRC 算法
	pid_ladrc.h	
	pid_leso.c	集成 LESO 的 PID 算法

	pid_leso.h	
	util.h	常见数学函数（限幅，S 曲线）
	└─base_drv	
	chassis_base.c	底盘解算基础
	chassis_base.h	
	drv_can.c	CAN 接收基础
	drv_can.h	
	drv_uart.c	串口收发基础
	drv_uart.h	
	drv_conf.h	框架配置文件
	ext_imu.c	外置 HI229 IMU 接收协议（未来可能弃用）
	ext_imu.h	
	motor_ctrl.c	电机控制基础
	motor_ctrl.h	
	rc.c	DR16 遥控/图传串口接收基础
	rc.h	
	referee.c	裁判系统接收基础
	referee.h	
	referee_conf.h	裁判系统数据类型配置
	super_cap.c	超级电容收发基础
	super_cap.h	
	└─tests	测试功能的代码
	robot_tests.c	机器人模块测试
	robot_tests.h	
	ui_tests.c	UI 绘制部分测试
	ui_tests.h	
	mem_log.c	日志模块
	mem_log.h	
	motor_tests.c	电机测试（转速闭环，角度闭环）
	motor_tests.h	

BMI088 驱动代码采用 Bosch 官方提供的库：

<https://github.com/BoschSensortec/BMI08x-Sensor-API>

采用轻量级的 Trice 日志库 (<https://github.com/rokath/trice>) + OpenOCD RTT 读取日

志，后续可能采用串口黑匣子等方式保存日志以供调试

1.5.3.3 重点功能

1.5.3.3.1 线性状态观测器(LESO)

参考 ADRC 部分模块的设计，移植 LESO 到串级 PID 中，其中 ESO 模块本身需要角度和当前输出作为反馈值，ESO 输出反馈到角度，转速以及最终输出。同时针对 6020 云台电机采用对应模型的补偿，根据参考文献[1]，BLDC 的电机数学模型如下

$$L \frac{di}{dt} = Ku - K_e \omega - iR$$

$$J \frac{d\omega}{dt} = K_m i - M_f(\omega)$$

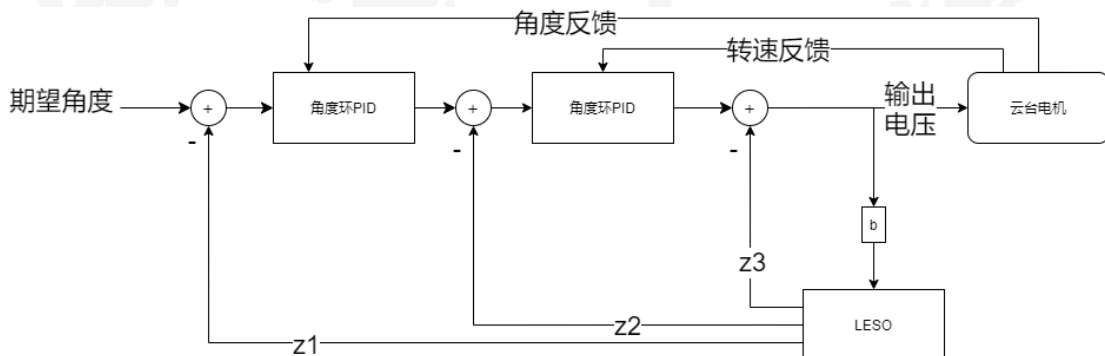
故 $\frac{d\omega}{dt} = -\frac{K_m K_e}{JR} \omega + \frac{KK_m}{JR} u - \frac{1}{J} M_f(\omega)$

转换为状态空间表达式：

$$\begin{cases} \omega = \dot{\theta} \\ \dot{\omega} = -\frac{K_m K_e}{JR} \omega + \frac{KK_m}{JR} u - \frac{1}{J} M_f(\omega) \\ y = \theta \end{cases}$$

因此可以构造一个线性状态观测器进行扰动观测

$$\begin{cases} \epsilon = z_1(k) - y(k) \\ z_1(k+1) = z_1(k) + h(z_2(k) - \beta_{01}\epsilon) \\ z_2(k+1) = z_2(k) + h\left(z_3(k) - \beta_{02}\epsilon + \frac{KK_m}{JR} u(k) - \frac{K_m K_e}{JR} z_2(k)\right) \\ z_3(k+1) = z_3(k) - h\beta_{03}\epsilon \end{cases}$$

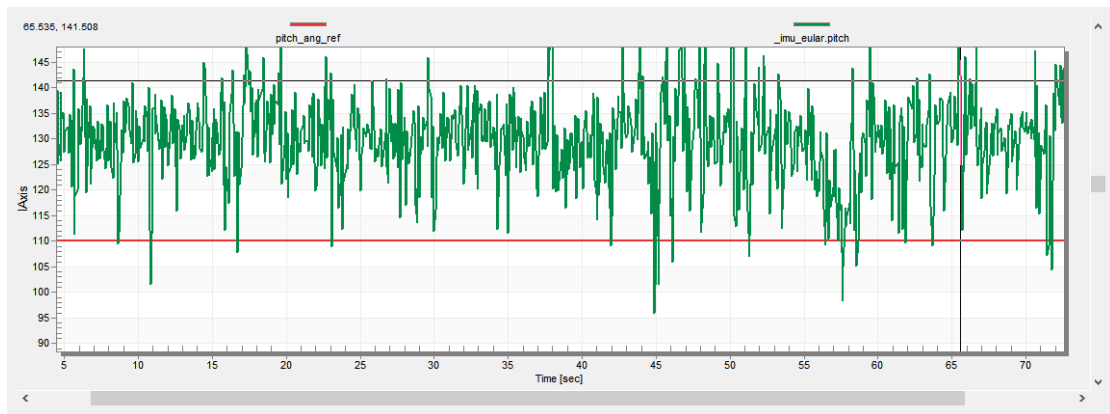


<图 4.4：流程示意图>

其中转速常数 K_e ，转矩常数 K_m 以及线圈电阻 R 可以通过查阅 GM6020 手册得到，角度和转速反馈量使用 IMU 数据，而电压 u 则可以读取 GM6020 反馈的当前真实电压值，转动惯量 J 根据机械估计即可，而三个 β 系数可以使用如下方式获得

$\beta_{01} = 3\omega_0, \beta_{02} = 3\omega_0^2, \beta_{03} = \omega_0^3$ ，其中 ω_0 代表扰动带宽的大小

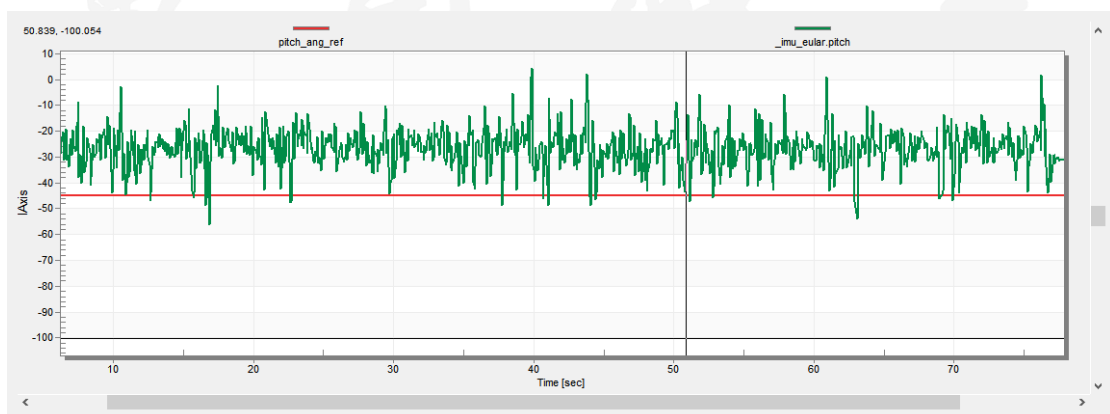
早期实际测试中，LESO 在抗 PID 超调方面效果显著，同时针对击打能量机关的跟随性问题也有不错的效果。但在目前采用的 BMI088 高性能 IMU 与 1kHz 控制频率的大带宽架构下其静态响应的优势并不是十分明显。而在此基础上再增加 LESO 调参的效果是，云台小范围内的阻尼感明显增加，同时在底盘复杂震动较大的情况下（如小陀螺）抗扰动性能有所提升，如下图所示。



原地小陀螺，单串级 PID 控制云台 Pitch（8192 角度制）：

<图 4.5：单串级效果图>

原地小陀螺，串级 PID+LESO 控制云台 Pitch（8192 角度制，地面状况和串级 PID 参数



与上图完全相同）：

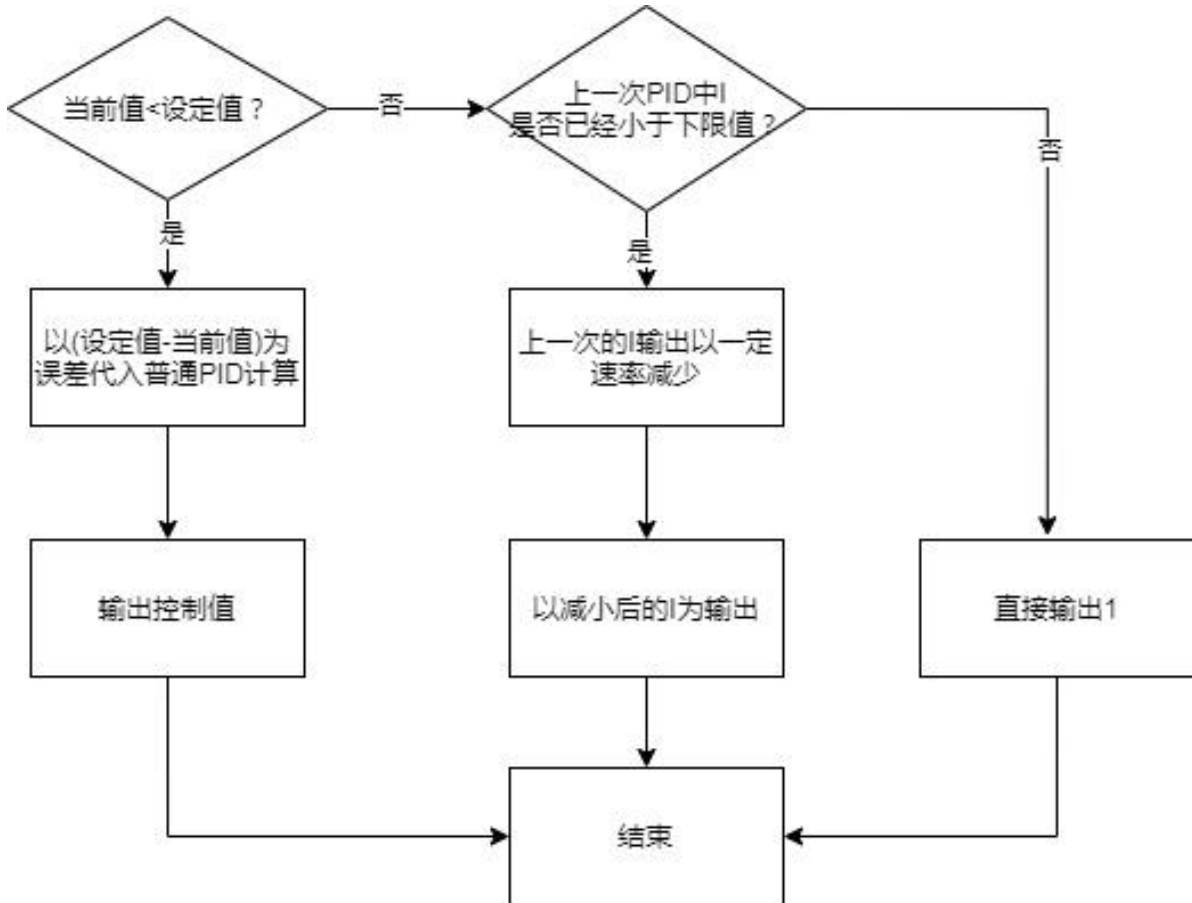
<图 4.6：串级 PID+LESO 效果图>

1.5.3.3.2 主动功率控制

超级电容控制板完成的功能是在限定功率之下给电容充电，通过降压实现对功率的被动限制，而机器人主控部分则是完成超级电容的使用策略管理。基础的思路是采用峰值限制的

单边 PID 输出一个大于 1 的功率除数，除以底盘输出来完成对电机的主动功率限制。单边的 PID 是指，仅当小于期望值（下限值）时 PID 才会有输出的 PID，实现单向输出。为了减少频繁进入一边的输出跳变，在离开小于范围时仍然将上一次 PID 的 I 项直接输出，并以一定速率减少。

具体实现流程如下：



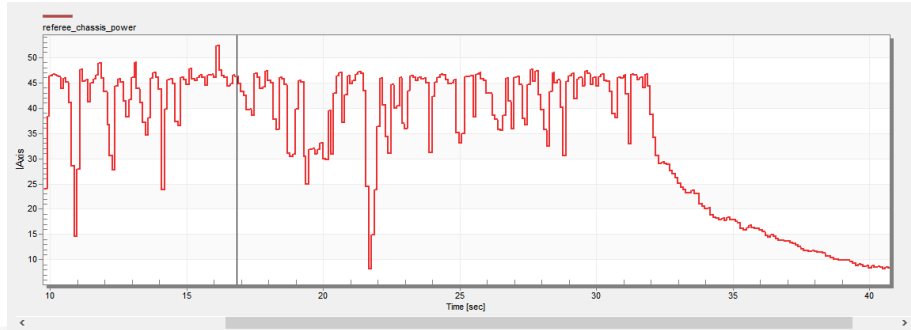
<图 4.7: 功率控制流程图>

以这一特殊的 PID 为基础，步兵机器人上可实现两种限制模式，即超级电容电压保持模式和超级电容释放模式。

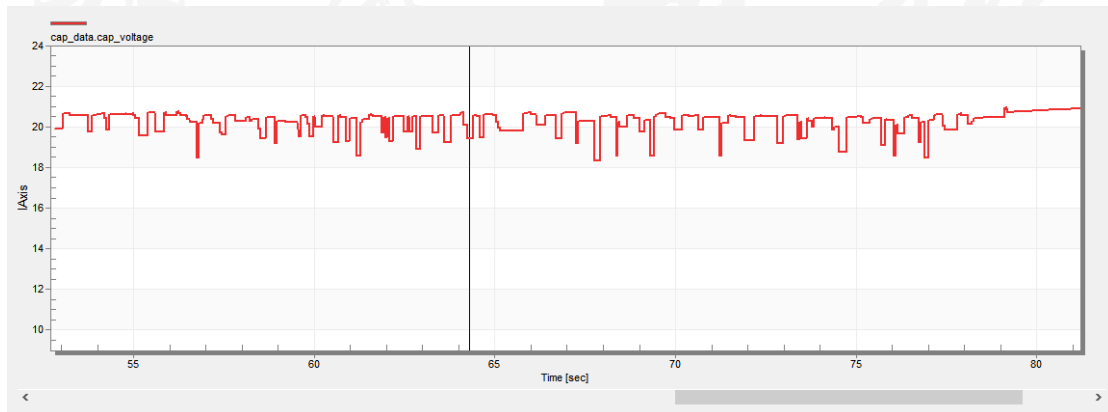
在超级电容电压保持模式下，机器人运动会尽量避免使用超级电容，即主要使用裁判系统功率输入而抑制超级电容电压下降。该算法会记录整个过程中的电压峰值（以电容满电的 95% 的电压作为峰值的最大值，避免在满电的情况下仍然有限制输出），并以此作为下限代入单边 PID 计算功率限制除数。同时通过最大功率和电压估算最大电流对电机输出进行限幅，消除电机输出的尖峰，使底盘加速时更加平稳。

在超级电容释放模式下，机器人输出最大不限，但是超级电容电压必须确保在电调最低工作电压（运动时测量到的极限大概在 10V 左右）之上，因此单边 PID 的限制值设置在 13V 至 14V 之间，上述记录的峰值拉到 10V 确保在模式切换时峰值能正确获取。实际控制中的超

级电容电压保持模式下电容电压不可能完全不下降，在加速时仍然会使用一部分电容能量，最后电压值在 22V(电容满电电压)至 20V 之间波动。在超级电容释放模式下，电容电压最低约为 12V，此时电调仍然能正常工作。小陀螺全速移动下的测试数据如下图



<图 4.8: 裁判系统的底盘输出功率>



<图 4.9: 电容电压（电容抑制模式）>

1.5.3.3.3 IMU 解算

基于 BMI088 数据融合出对应角度，参考文献[2]采用 Mahony 算法进行姿态解算，但在实际应用中需要仍然需要做如下改进

1.5.3.3.3.1 IMU 校准

BMI088 加速度计与角速度计存在偏置与尺度因子误差，加速度计的误差会影响重力加速度的方法估计，使自动瞄准的弹道解算出现误差，而角速度计的偏置则表现为静止状态下云台得缓慢漂移，影响操作手感。即使 BMI088 已经完成出厂校准，但在安装到机器人上前仍然需要经过校准确保精确

加速度计校准方法采用六面校准，参考 Pixhawk 的校准算法

(https://github.com/darenlee/PixhawkPSP_Firmware/blob/master/src/modules/commader/accelerometer_calibration.cpp)

而角速度计只需要连续测量一个固定偏置即可，下面的动态校准将会解决零偏变化的问题。

至于温度补偿或恒温控制，长时间的实际测试中并无明显影响，或者说角速度计噪声和温漂在同一量级上

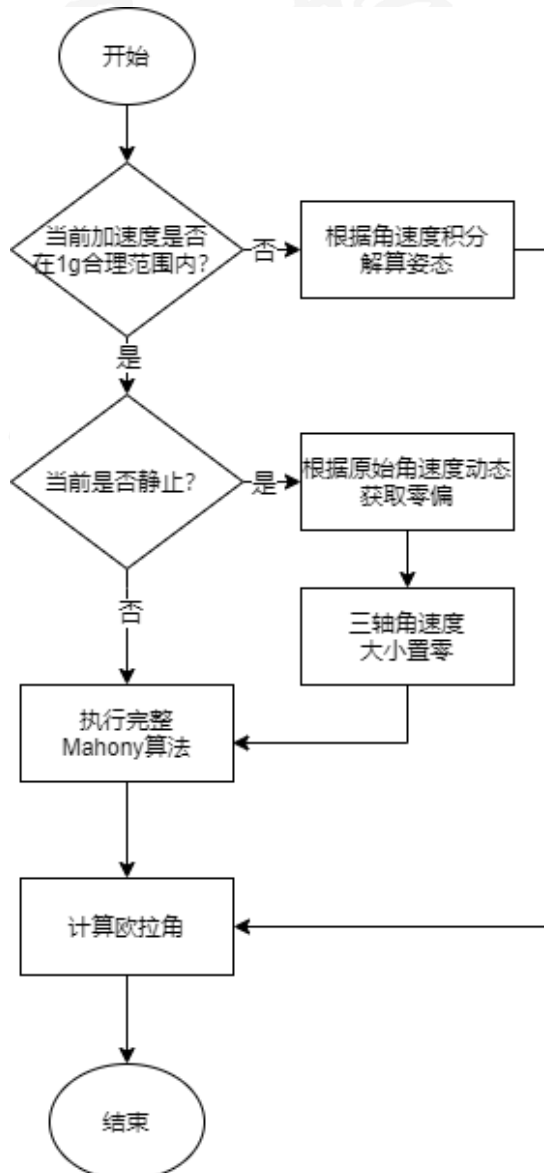
1.5.3.3.3.2 高 G 检测

算法中假设原始加速度数据为纯重力加速度方向，而在赛场中机器人剧烈运动甚至跌落的情况下显然不成立。因此需要设定一个合理的重力加速度区间，当大于该区间时认为加速度计无效而仅用角速度计积分解算角度。

1.5.3.3.3.3 静止检测（防零飘）与动态校准

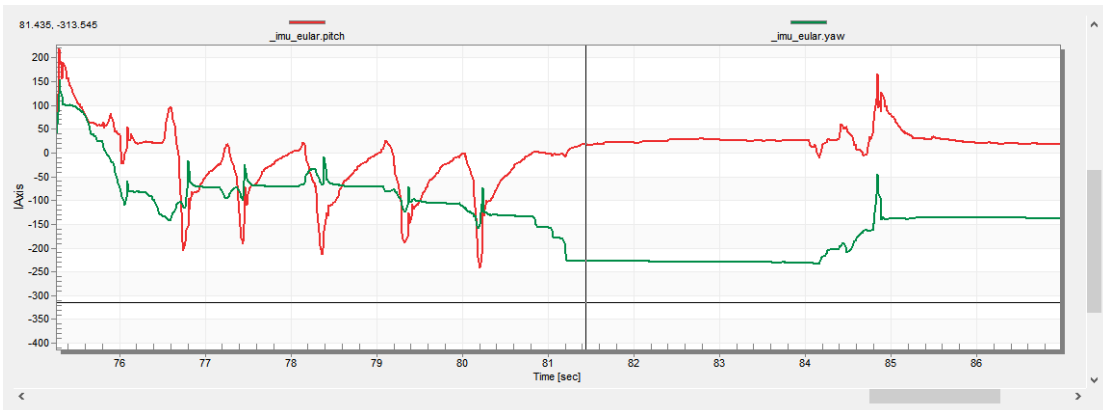
在云台静止的状态下易受零偏的影响，因此在加速度计接近 1g 并且三轴角速度非常小的情况下可以认为云台静止，该状态下可将角速度计的值作为零偏数据采集，经过一个极低带宽的低通滤波器后得到稳定的动态校准值，同时解算时应认为角速度计的输出为 0 以避免其漂移带来的影响。

综合上述几个改进，整个算法的流程如下图所示：

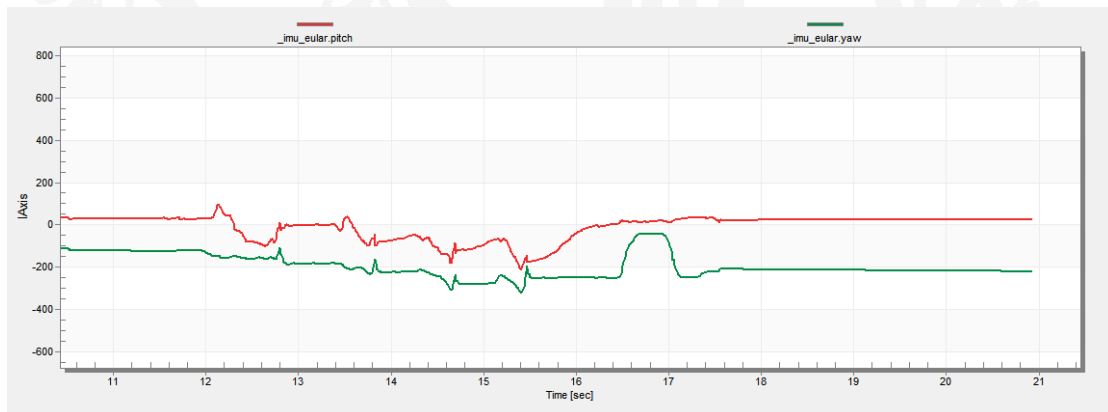


<图 4.10: 算法流程示意图>

单 C 板手动摔落测试的角度波形如下图所示:



<图 4.11: 未使用高 G 检测的算法解算结果>



<图 4.12: 使用高 G 检测的算法解算结果>

实际测试中，解算出的云台角度值静止时稳定无漂移，飞坡与下台阶时云台角度基本稳定，在跌落中云台角度未出现明显改变。

1.5.3.3.4 实时弹道解算

同理于装甲板姿态解算，我们将弹道解算在电控下位机端完成。

弹道解算的最大难点在于精度与实时性，后者在下位机内核资源不宽裕的条件下更为明显。而已有的弹道解算方案考虑到预测时间与空气阻力多数采用迭代逼近的算法，效率略低。下面提出针对电控端弹道解算的方案及其优化。

- 理论推导

设 Pitch 轴的角度为 θ ，发射初速度为 v_0 ，下面推导无空气阻力的抛物线模型下根据目标位置逆向求解发射 Pitch 角度的过程（假设此时发射点和目标以及弹道均在一个平面上）

$$\begin{cases} x_0 = v_0 t \cos \theta \\ z_0 = v_0 t \sin \theta - \frac{1}{2} g t^2 \end{cases}$$

$$t = \frac{x_0}{v_0 \cos \theta}$$

$$z_0 = x_0 \tan \theta - \frac{g x_0^2}{2 v_0^2 \cos^2 \theta}$$

$$z_0 \cos^2 \theta = \sin \theta \cos \theta x_0 - \frac{g x_0^2}{2 v_0^2}$$

$$\frac{1 + \cos 2\theta}{2} z_0 = \frac{x_0}{2} \sin 2\theta - \frac{g x_0^2}{2 v_0^2}$$

$$z_0 + \frac{g x_0^2}{v_0^2} = \left(\frac{x_0}{\sqrt{x_0^2 + z_0^2}} \sin 2\theta - \frac{z_0}{\sqrt{x_0^2 + z_0^2}} \cos 2\theta \right) \sqrt{x_0^2 + z_0^2}$$

$$\frac{z_0 + \frac{g x_0^2}{v_0^2}}{\sqrt{x_0^2 + z_0^2}} = \sin(2\theta - \phi), \phi = \arctan\left(\frac{z_0}{x_0}\right)$$

$$\text{let } l = \sqrt{x_0^2 + z_0^2}, \text{ when } z_0 + \frac{g x_0^2}{v_0^2} \leq l$$

$$\alpha = \arcsin\left(\frac{z_0 + \frac{g x_0^2}{v_0^2}}{l}\right)$$

$$\therefore \theta = \frac{\alpha + \phi}{2} \quad \text{or} \quad \theta = \frac{\pi - \alpha + \phi}{2}$$

<图 4.14：弹道解算理论推导>

奮
創
新
激
發
浪
潮

可以看到相比迭代上述过程不仅能计算有无解的情况，还可以计算出完整的两个解的抛物线。同时其解算精度只取决于几个反三角函数的精度，并且不存在收敛性判定的问题。进一步地可以考虑空气阻力模型下的解算。

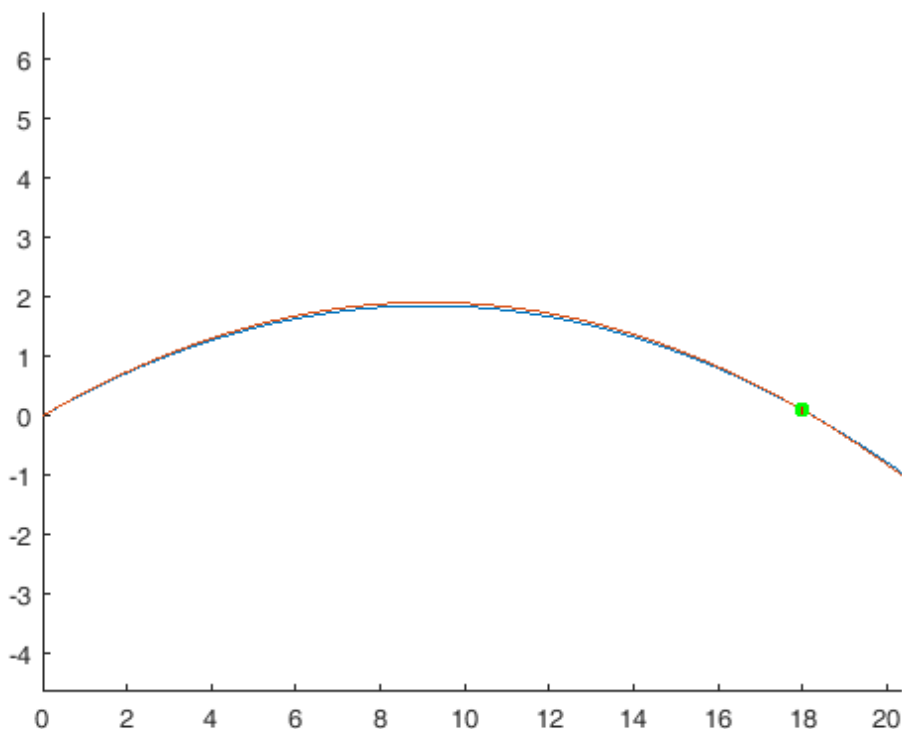
$$\begin{aligned}
 & \text{assume } \vec{f} = -k\vec{v}, \lambda = \frac{k}{m} = \frac{C\rho\pi r^2}{m} \\
 & \begin{cases} \dot{\vec{r}} = \vec{v}_0 e^{-\lambda t} + \frac{\vec{g}}{\lambda}(1 - e^{-\lambda t}) \\ \vec{r} = -\frac{\vec{g}}{\lambda}t + \frac{\vec{v}_0 - \frac{\vec{g}}{\lambda}}{\lambda}(1 - e^{-\lambda t}) \end{cases} \\
 & \begin{cases} x_0 = v_0 \cos\theta \frac{1 - e^{-\lambda t}}{\lambda} \\ z_0 = -\frac{g}{\lambda}t + (v_0 \sin\theta + \frac{g}{\lambda}) \frac{1 - e^{-\lambda t}}{\lambda} \end{cases} \\
 & t = -\frac{1}{\lambda} \ln\left(1 - \frac{\lambda x_0}{v_0 \cos\theta}\right) \\
 & z_0 = \frac{g}{\lambda^2} \ln\left(1 - \frac{\lambda x_0}{v_0 \cos\theta}\right) + (v_0 \sin\theta + \frac{g}{\lambda}) \frac{x_0}{v_0 \cos\theta} \\
 & f(\theta) = z(\theta) - z_0 \\
 & = \frac{g}{\lambda^2} \ln\left(1 - \frac{\lambda x_0}{v_0 \cos\theta}\right) + (v_0 \sin\theta + \frac{g}{\lambda}) \frac{x_0}{v_0 \cos\theta} - z_0 \\
 & = \frac{g}{\lambda^2} \ln\left(1 - \frac{\lambda x_0}{v_0 \cos\theta}\right) + x_0 \tan\theta + \frac{g x_0}{\lambda v_0 \cos\theta} - z_0 \\
 & \text{let } t_0 = \frac{x_0}{v_0 \cos\theta}, \text{ when } \lambda t_0 < 1 \text{ and } v_0 t_0 > x_0 \\
 & \tan\theta = \pm \sqrt{\frac{1}{\cos^2\theta} - 1} = \pm \sqrt{v_0^2 t_0^2 - x_0^2} \\
 & \therefore f(t_0) = \frac{g}{\lambda} \left[\frac{1}{\lambda} \ln(1 - \lambda t_0) + t_0 \right] \pm \sqrt{v_0^2 t_0^2 - x_0^2} - z_0 \\
 & f'(t_0) = \frac{g}{\lambda} \left(\frac{1}{\lambda} \frac{-\lambda}{1 - \lambda t_0} + 1 \right) \pm \frac{2v_0^2 t_0}{2\sqrt{v_0^2 t_0^2 - x_0^2}} \\
 & = \pm \frac{v_0^2 t_0}{\sqrt{v_0^2 t_0^2 - x_0^2}} - \frac{g t_0}{1 - \lambda t_0} \\
 & \text{using Newton's Method} \\
 & t_{n+1} = t_n - \frac{f(t_n)}{f'(t_n)} \\
 & \text{finally, } \theta = \arccos\left(\frac{x_0}{v_0 t}\right)
 \end{aligned}$$

<图 4.15: 空气阻力模型下的弹道解算推导>

由于考虑空气阻力的情况下需要求解含对数项的超越方程，可以将目标 z 值误差转换为关于时间 t 的函数，利用牛顿迭代法求解方程的解。由于空气阻力对弹道影响较小，其

飞行时间差异不大，故其解应该在无空气阻力模型下的飞行时间的解的值附近。利用这一点，可以将上面弹道解出的两个解的作为牛顿迭代的初始值加快其收敛速度。

对上述算法编写 Matlab 脚本仿真，模拟英雄 16m/s 吊射 20m 远目标（空气阻力通过公式预先计算），迭代精度取 $10e-4$ ，可验证其正确性如图所示（蓝线为无空气阻力弹道，橙线为有空气阻力模型，绿点为目标，在绿点上的红线代表小装甲板的范围）。可见两者弹道差异不是十分明显，解算角度两种模型也只有 0.5 度的误差



<图 4.16: 模型仿真效果>

- 实现与优化

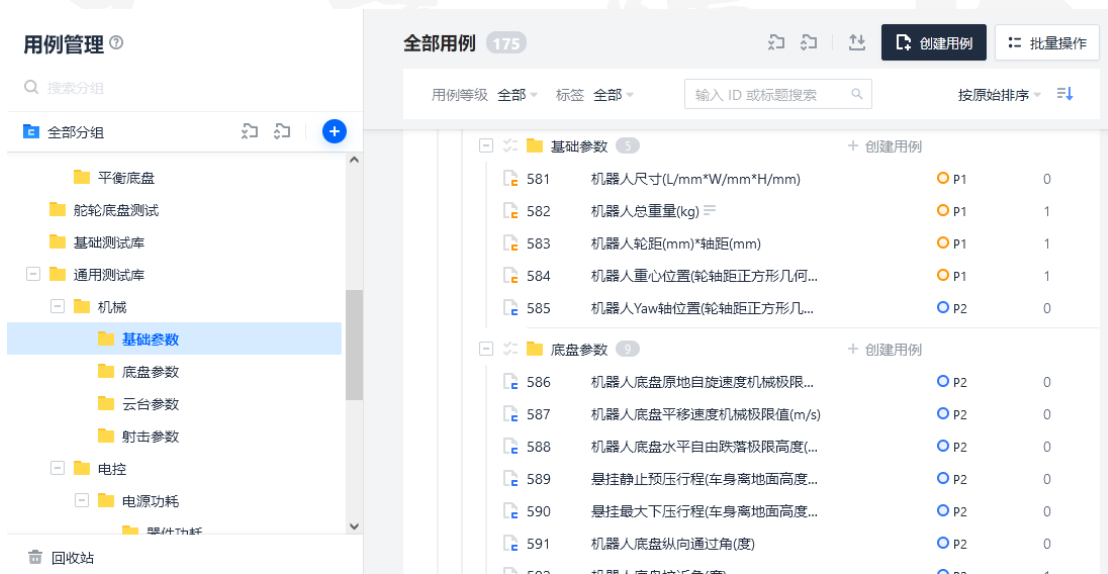
考虑到实现上的计算的复杂程度和精度，采用无空气阻力的模型作为步兵或英雄的弹道解算更加合适。由于控制周期为 1ms，因此每次弹道计算耗时必须远小于 1ms，这对其实时性提出极高的要求。

弹道计算耗时最多的位置应该在反三角函数的计算，此处采用 CORDIC (Coordinate Rotation Digital Computer) 算法实现，相比于传统计算方法一次 CORDIC 计算可以完成两种运算，例如 atan 和 sqrt 同时完成可大大降低计算时间。

实际在 STM32F103 (72Mhz, Cortex-M3) 验证优化时 (-O0)，所有计算包括对数均采用 CORDIC 计算后，单次完整计算（带空气阻力迭代过程）总耗时 0.3ms，在上述以无空气阻力抛物线飞行时间为初解开始迭代，仅需 2-3 次即可达到 0.1mm 以下的计算精度。实际在 RoboMaster 开发板 C 型 (STM32F407, 带 FPU, 168Mhz) 未开优化的情况下计算时间已经低于 0.1ms，基本满足要求。

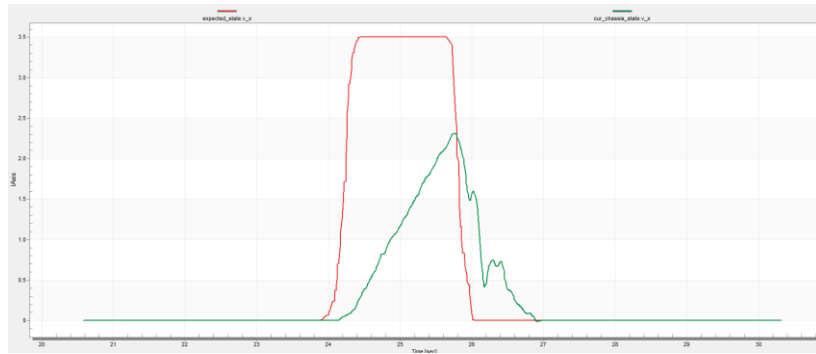
应用于实际击打哨兵效果良好，除了由于 IMU 加速度计未完全校准带来的角度偏差外，能够适应各种距离的目标并击打。

1.5.3.4 软件测试



<图 4.17: 测试用例>

通过 freemaster 调试检查机器人各类数据，确保机器人测试的完整性。针对电控的测试主要有功耗测试（静态功耗，底盘功率控制测试），底盘测试（极限速度大小，飞坡测试），云台测试（响应速度测试，控制精度测试，抗震稳定性测试）以及发射测试（弹道散布，射速稳定性测试）等。以全向轮测试为例，测试其速度及其加速曲线，在网页端测试计划中可记录测试结果。



<图 4.18: 全向轮速度曲线 (根据轮速计算) >



<图 4.19: 测试用例使用示意图 >

1.5.4 算法设计

1.5.4.1 常规装甲板自瞄算法

1.5.4.1.1 功能概述

本赛季视觉代码以 ROS2 humble 为运行框架，将各功能模块封装成多个功能包，以 ROS composable node 的形式编译运行，各个包之间独立运行，以实现松耦合、易迭代、易维护的优点。功能包中包含多个 ROS 节点，有 camera 节点、armor_detector 节点、armor_processor 节点、serial_driver 节点，各节点之间通过 ROS topic 和 ROS service 通信，其中 camera 节点和 detector 节点运行同一于 ComposableNodeContainer 中，两个节点之间传输的是 Image::msg 信息，实现在单独的进程中运行多个节点，进程内通信可以更低资源开销的同时更高效。

代码目录结构：

```

src
├── rm_pioneer_config // 描述及配置文件
│   ├── LICENSE
│   ├── rm_pioneer_bringup
│   │   ├── config // 参数文件
│   │   │   ├── camera_info.yaml
│   │   │   └── params.yaml
│   │   ├── launch // 启动文件
│   │   │   └── vision_bringup.launch.py
│   │   ├── CMakeLists.txt
│   │   └── package.xml
│   └── rm_pioneer_description // 机器人模型及坐标系描述文件
│       ├── launch // 用于模型预览的启动文件及 rviz 显示配置文件
│       │   ├── view_model.launch.py
│       │   └── view_model.rviz
│       ├── meshes // 存放模型文件
│       │   ├── camera_base_link.STL
│       │   ├── pitch_link.STL
│       │   └── yaw_link.STL
│       ├── CMakeLists.txt
│       ├── package.xml
│       └── urdf
│           └── gimbal.urdf.xacro // 描述文件
└── rm_pioneer_vision // 代码主要部分
    ├── Dockerfile

```

```
|— LICENSE
|— README.md
|— rm_auto_aim
|   |— armor_detector // detector 节点
|   |   |— include
|   |   |   |— armor_detector
|   |   |   |   |— armor.hpp
|   |   |   |   |— detector.hpp
|   |   |   |   |— detector_node.hpp
|   |   |   |   |— number_classifier.hpp
|   |   |   |   |— pnp_solver.hpp
|   |   |— src
|   |   |   |— detector.cpp // 装甲板识别算法核心部分
|   |   |   |— number_classifier.cpp // 数字识别功能封装
|   |   |   |— pnp_solver.cpp // 位置解算
|   |   |   |— detector_node.cpp //适配单目相机的识别节点类
|   |   |— template //用于数字识别的匹配模板
|   |   |   |— 1.png
|   |   |   |— 2.png
|   |   |   |— 3.png
|   |   |   |— 4.png
|   |   |   |— 5.png
|   |   |   |— base.png
|   |   |   |— guard.png
|   |   |   |— outpost.png
|   |   |— package.xml
|   |   |— CMakeLists.txt
|   |   |— test
|   |   |   |— test_node_startup.cpp
|— armor_processor //processor 节点
|   |— include
|   |   |— armor_processor
|   |   |   |— extended_kalman_filter.hpp
|   |   |   |— processor_node.hpp
|   |   |   |— tracker.hpp
|   |   |— src
|   |   |   |— extended_kalman_filter.cpp // 卡尔曼滤波器基础实现
|   |   |   |— processor_node.cpp // ros 封装
|   |   |   |— tracker.cpp // 跟踪器
|   |   |— CMakeLists.txt
|   |— package.xml
```

```

| | | └─ test
| | |   └─ test_kalman_filter.cpp
| └─ auto_aim_interfaces // 自定义 ROS 消息包
| | └─ msg
| | | └─ Armor.msg
| | | └─ Armors.msg
| | | └─ DebugArmor.msg
| | | └─ DebugArmors.msg
| | | └─ DebugLight.msg
| | | └─ DebugLights.msg
| | |   └─ Target.msg
| | └─ CMakeLists.txt
| | └─ package.xml
| └─ LICENSE
| └─ README.md
| └─ rm_auto_aim
| | └─ config
| | | └─ default.yaml
| | └─ launch
| | | └─ auto_aim.launch.py // 单识别部分启动文件
| | └─ CMakeLists.txt
| | └─ package.xml
└─ rm_serial_driver // 串口包
| └─ 89-stm-virtual-com.rules
| └─ config
| | └─ serial_driver.yaml
| └─ include
| | └─ rm_serial_driver
| | | └─ crc.hpp // crc 校验
| | | └─ packet.hpp // 串口数据包及处理函数
| | | └─ rm_serial_driver.hpp
| └─ launch
| | └─ serial_driver.launch.py
| └─ CMakeLists.txt
| └─ package.xml
| └─ src
| | └─ rm_serial_driver.cpp
└─ ros2_mindvision_camera // mv 相机包
| └─ 88-mvusb.rules
| └─ config
| | └─ camera_info.yaml

```

```
| | └─ camera_params.yaml
| | └─ docs
| | └─ rqt.png
| | └─ launch
| | └─ calibration.launch.py // 标定程序启动文件
| | └─ mv_launch.py // 单相机节点启动
| └─ LICENSE
| └─ mvsdk // 相机 SDK
| | └─ include
| | | └─ CameraApi.h
| | | └─ CameraDefine.h
| | | └─ CameraStatus.h
| | └─ lib
| | └─ libMVSDK.so
| └─ CMakeLists.txt
| └─ package.xml
| └─ README.md
| └─ src
|   └─ mv_camera_node.cpp
└─ ros2_realsense_param // realsense 相机包
  └─ config
  | └─ d455.yaml
  └─ launch
  | └─ rs_launch.py
  └─ CMakeLists.txt
  └─ package.xml
└─ startup.sh
  └─ vision.rviz // rviz2 可视化启动配置文件
```


1.5.4.1.1.1 自定义 ROS 消息类型介绍

- Armor

数据	数据类型	数据说明
number	uint8	装甲板数字
distance_to_image_center	float32	装甲板中心到相机中心的距离
pose	geometry_msgs/Pose	装甲板基于相机坐标系的三维坐标以及旋转向量
color	uint8	装甲板颜色

<表 5.1: Armor 数据>

- Armors

数据	数据类型	数据说明
header	std_msgs/Header	包含当前信息的时间戳和 frame_id
armors	Armor[]	装甲板信息容器

<表 5.2: Armors 数据>

- Target

数据	数据类型	数据说明
header	std_msgs/Header	包含当前信息的时间戳和 frame_id
id	string	目标的数字
tracking	bool	代表当前是否锁定目标
position	geometry_msgs/Point	目标中心的 x、y、z 三维坐标
yaw	float64	目标当前的倾斜角度
velocity	geometry_msgs/Vector3	目标分别在 x、y、z 上的速度

v_yaw	float64	目标旋转的角速度
radius_1	float64	目标其中一组装甲板相对中心的半径
radius_2	float64	目标另一组装甲板相对中心的半径
z_2	float64	tracking 中的装甲板的上一块装甲板的 z 轴位置

<表 5.3: target 数据>

以下为 Debug 模式下用到的数据，这些数据通过 ROS 发布，作为调参时的依据

- DebugArmor

数据	数据类型	数据说明
center_x	int32	装甲板中心在图像坐标系下的横坐标
is_armor	bool	判断结果，代表是否为装甲板
light_ratio	float32	两个灯条的长宽比
center_distance	float32	两个灯条中心的距离与灯条平均长度的比值
angle	float32	装甲板的倾斜角度
armor_type	bool	判断结果，true 为小装甲板，false 为大装甲板

<表 5.4: DebugArmor 数据>

- DebugLight

数据	数据类型	数据说明
center_x	int32	灯条中心在图像坐标系下的横坐标

is_light	bool	判断结果，代表是否为灯条
ratio	float32	灯条长宽比
angle	float32	灯条倾斜角度

<表 5.5: DebugLight 数据>

- **DebugArmors**

数据	数据类型	数据说明
data	DebugArmor[]	用于 Debug 的装甲板信息容器

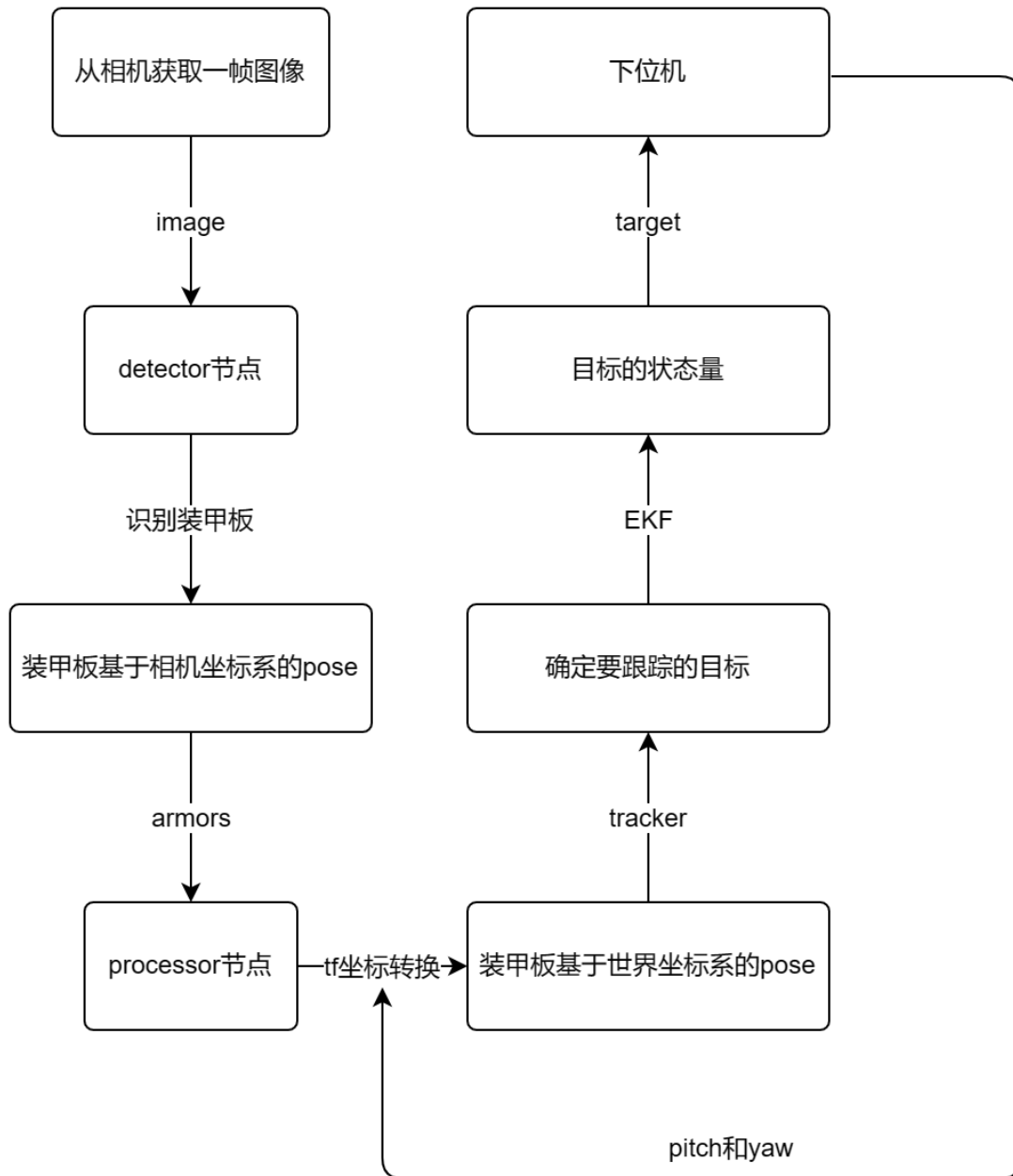
<表 5.6: DebugArmors 数据>

- **DebugLights**

数据	数据类型	数据说明
data	DebugLight[]	用于 Debug 的灯条信息容器

<表 5.7: DebugLights 数据>

1.5.4.1.2 工作流程



<图 5.1：系统工作流程图>

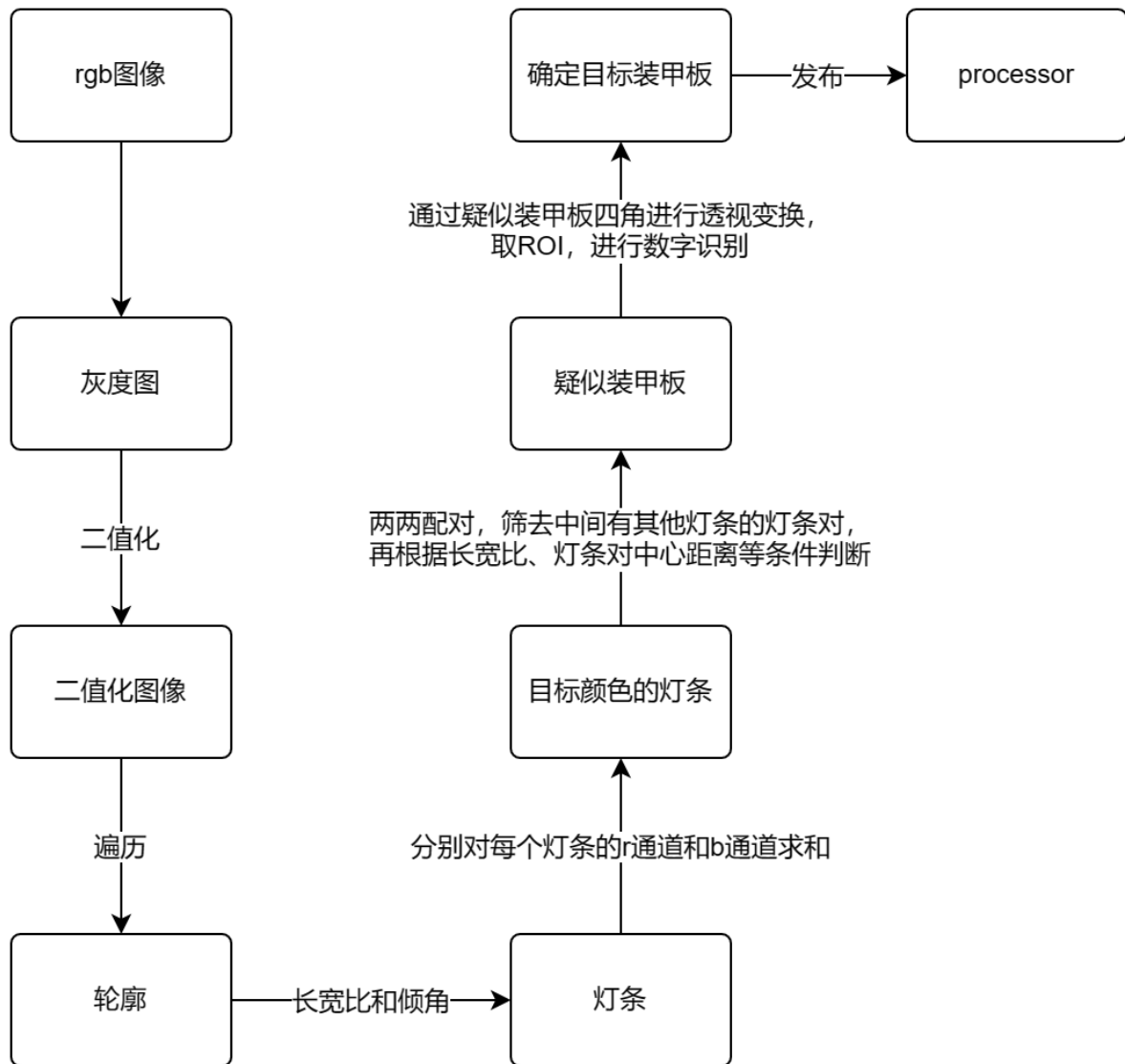
1.5.4.1.3 camera 节点

从相机读取图像数据并以 `image::msg` 的格式通过 ROS 发布到“/image_raw”话题。

1.5.4.1.4 detector 节点

订阅“/image_raw”话题的图像信息，并通过 `imagertransport` 转化为 OpenCV 的 Mat 类型图像，然后进行一系列图像处理，最终识别出装甲板。

具体工作流程如下：



<图 5.2: detector 节点工作流程图>

部分过程处理:

- 预处理:

由于一般工业相机的动态范围不够大, 导致若要能够清晰分辨装甲板的数字, 得到的相机图像中灯条中心就会过曝, 灯条中心的像素点的值往往都是 $R=B$ 。根据颜色信息来进行二值化效果不佳, 因此此处选择了直接通过灰度图进行二值化, 将灯条的颜色判断放到后续处

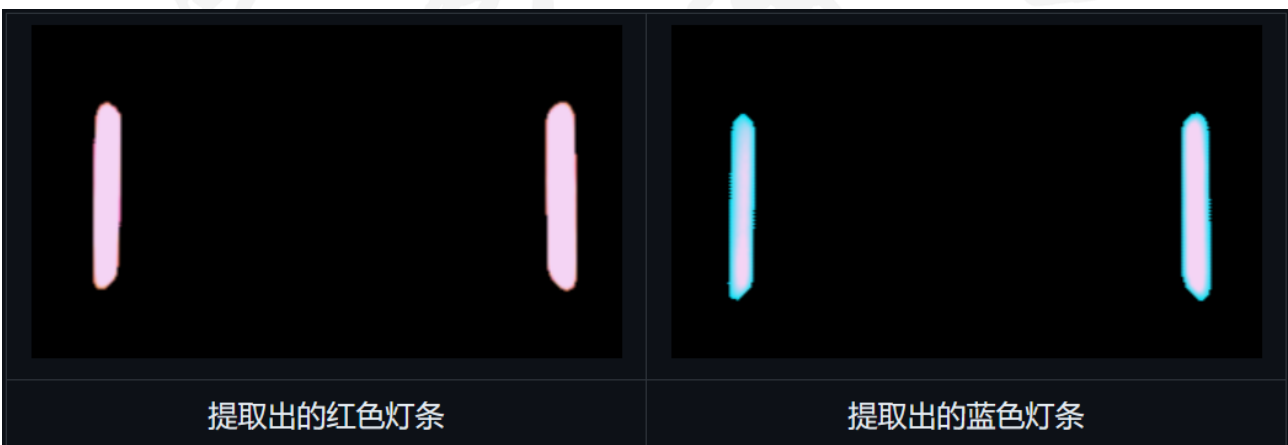


理中。

<图 5.3: 灯条预处理示意图>

- 寻找灯条:

通过 `findContours` 得到轮廓, 再通过 `minAreaRect` 获得最小外接矩形, 对其进行长宽比和倾斜角度的判断, 可以高效的筛除形状不满足的亮斑。判断灯条颜色这里采用了对轮廓内的 R/B 值求和, 判断两和的的大小的方法, 若 $\text{sum}_r > \text{sum}_b$ 则认为是红色灯条,



反之则认为是蓝色灯条。

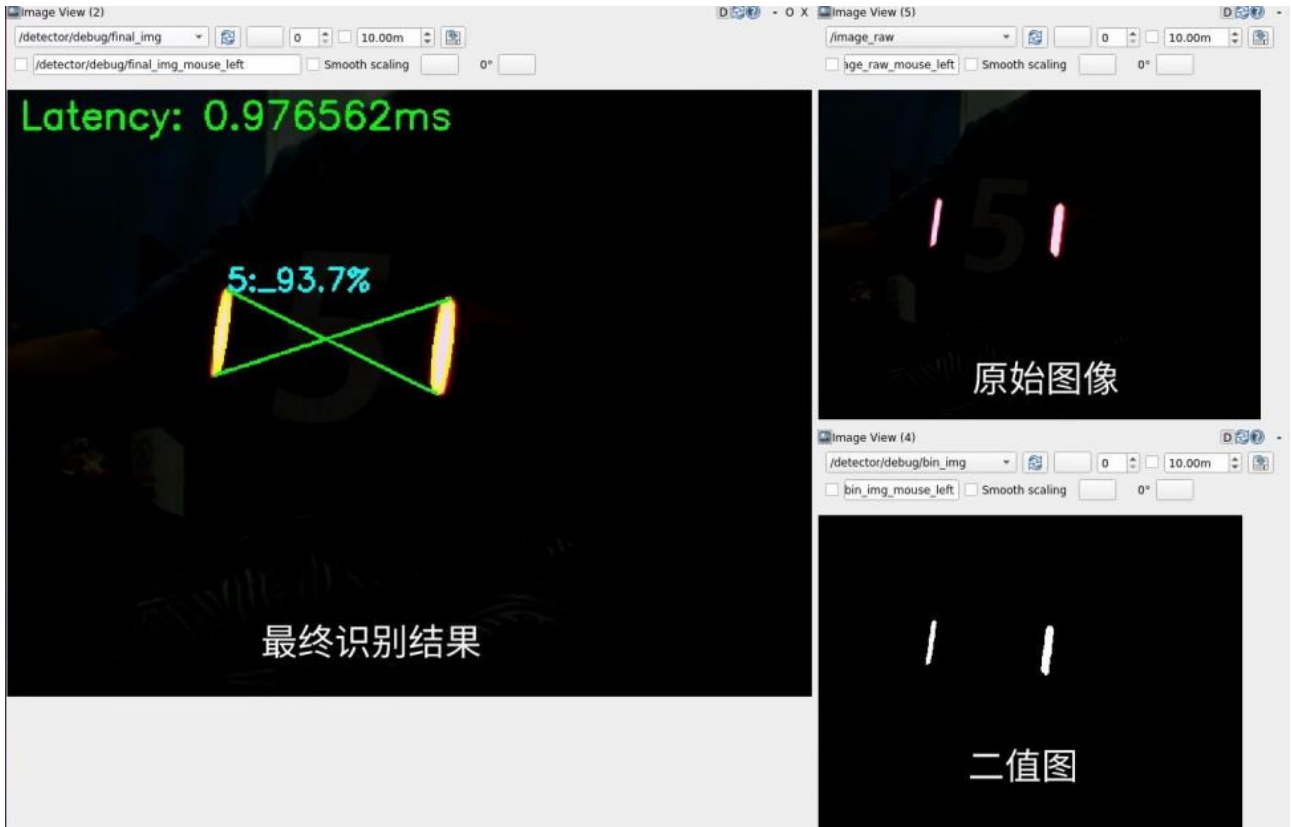
<图 5.4: 提取灯条示意图>

- 匹配灯条:

根据 `detect_color` 选择对应颜色的灯条进行两两配对, 首先筛除掉两条灯条中间包含

另一个灯条的情况，然后根据两灯条的长度之比、两灯条中心的距离、配对出装甲板的倾斜角度来筛选掉条件不满足的结果，得到形状符合装甲板特征的灯条配对。

效果如下：



<图 5.5：灯条识别效果图>

- 数字识别部分：

因为原始图像曝光度低，对比度低，难以进行识别，所以先通过灯条和装甲板边长的比例进行透视变换，然后取ROI区域，再使用大津法进行二值化处理。

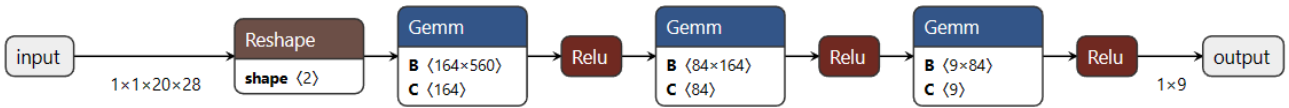


<图 5.6：数字识别处理流程图>

成功提取到二值化数字图像后，使用多层感知机（MLP）进行分类。网络结构中定义了

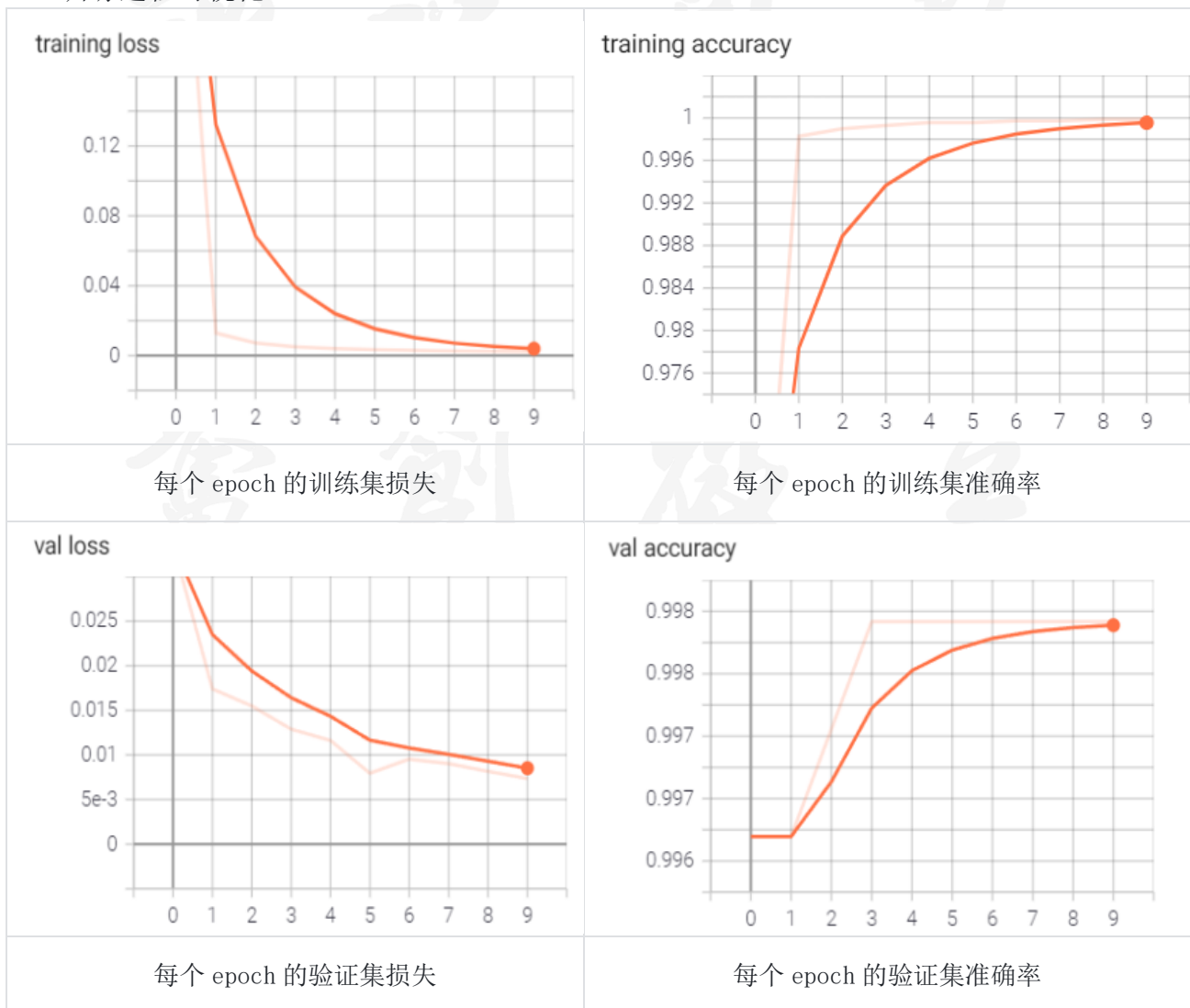
两个隐藏层和一个分类层，将二值化后的数字展平成 $20 \times 28 = 560$ 维的输入，送入网络进行分类，即可得到装甲板的 id。

网络结构如图：



<图 5.7: 神经网络结构流程图>

训练过程可视化：



<表 5.8: 训练过程可视化>

其识别效果如下图：

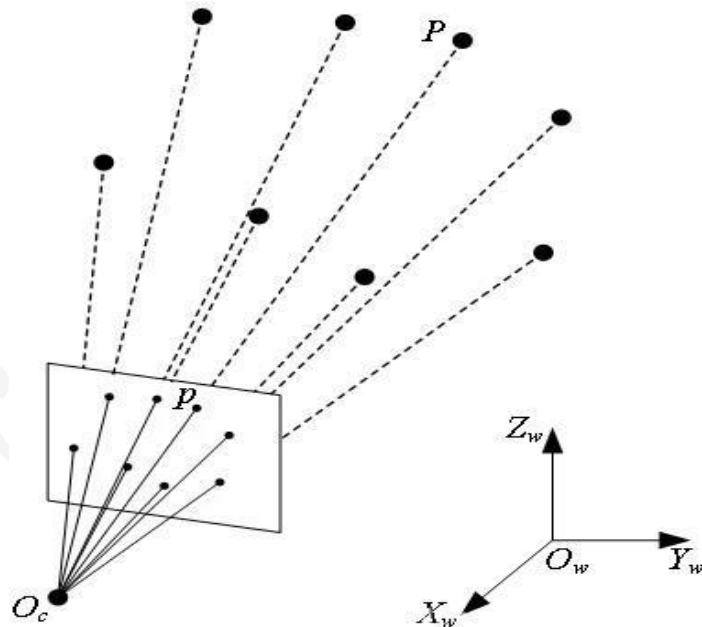


<图 5.8：识别效果图>

- 位置解算部分

接收相机节点发送的相机内参，使用 PnP（Perspective-n-Point）算法对识别所得装甲板进行解算，得到相机坐标系下的三维坐标点并以自定义的 ros 消息发布，用于下一步处理。

PnP 算法基本原理：已知相机内参，根据三维空间中的目标物体在图像坐标系上的投影及其真实尺寸，求解物体从三维空间到图像平面的变换关系（旋转矩阵、平移向量），从而得到目标物体在相机坐标系下的三维坐标，是单目测距中较为常用的方法。



<图 5.9: PnP 原理图>

在装甲板位置解算的任务中，由于解算目标为装甲板平面，我们调用 openCV 的 solvePnP 函数来完成，使用 IPPE (Infinitesimal Plane-based Pose Estimation) 方法，利用装甲板矩形的四个顶点进行变换关系的解算。

1.5.4.1.5 processor 节点

- 坐标转换部分

接收来自 armor_detector 的 armors 消息和来自 robot_state_publisher 的 tf 信息，节点中调用 tf_buffer 储存历史 tf，利用 tf 自动插值，实现陀螺仪数据与识别结果数据时间戳的软同步，完成三维点的坐标系转换。

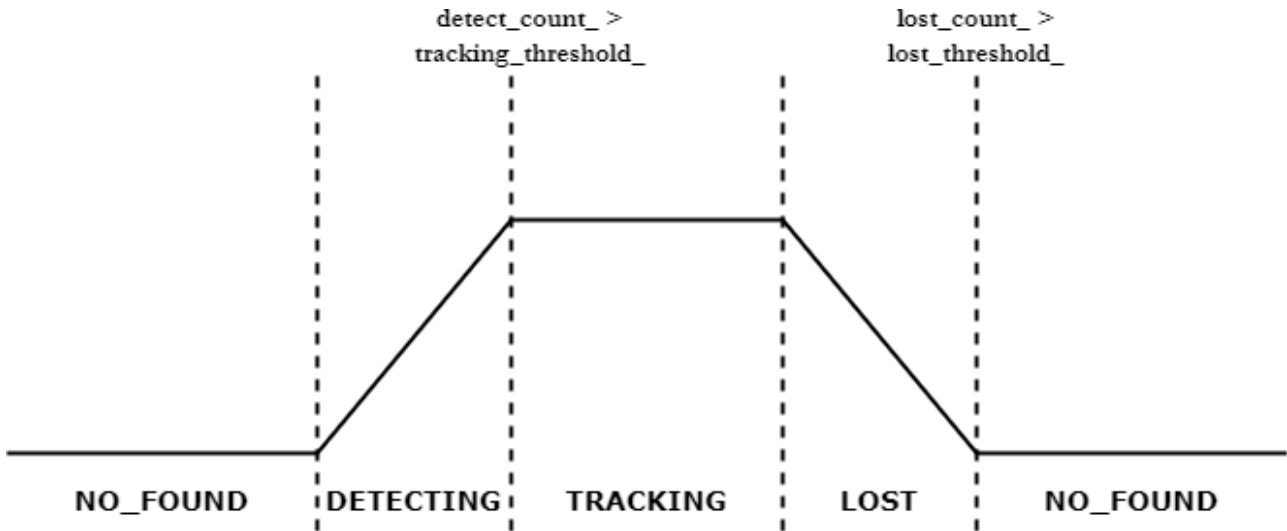
- 目标跟踪部分

跟踪器共 4 个跟踪状态：NO_FOUND（完全丢失），DETECTING（未跟踪），TRACKING（跟踪），LOST（暂时丢失目标）。

设置 detect_count 和 lost_count 用于状态判断，每获取一帧图像的目标数据都根据目标与滤波器预测位置进行筛选匹配，确定当前帧的击打目标。匹配成功时 detect_count 增加，到达阈值后状态更新为 TRACKING，detect_count 清零，匹配失败时 lost_count 增加，到达阈值后状态更新为 LOST，lost_count 清零。

跟踪器逻辑保证只有连续多帧识别到目标时才发送目标信息，有效防止因偶然误识别造成发送错误的目标信息。

跟踪状态变化如图：



<图 5.10：跟踪状态示意图>

- EKF 部分

卡尔曼滤波通俗理解就是使用上一个状态的最优估计（后验估计）来得出当前的先验估计，在由当前的观测值来修正得到最优估计，即后验估计的一个递归过程。

普通卡尔曼滤波只适用于状态系统线性化、误差符合高斯分布的情况，对整车状态预测并不适用，因此采用可以处理非线性化系统的拓展卡尔曼滤波模型。

$$\begin{aligned}
 x &= x_{-1} + v_{x_{-1}} * dt \\
 y &= y_{-1} + v_{y_{-1}} * dt \\
 z &= z_{-1} + v_{z_{-1}} * dt \\
 yaw &= yaw_{-1} + v_{yaw_{-1}} * dt
 \end{aligned}$$

在本任务中，假设目标在平地上同时有平移运动和自转运动，运动模型如下：

其中 x、y、z 为目标车中心的三维坐标，yaw 为目标的倾斜角度。

因为观测到的是装甲板的 x、y、z、yaw，因此有从观测到状态的公式如下：

$$\begin{aligned}
 x_c &= x_a + r * \cos(yaw) \\
 y_c &= y_a + r * \sin(yaw) \\
 z_c &= z_a \\
 yaw_c &= yaw_a
 \end{aligned}$$

其中 x_c 、 y_c 、 z_c 、 yaw_c 为目标车的状态， x_a 、 y_a 、 z_a 、 yaw_a 为观测到的装甲板的状态， r 为装甲板到车中心的距离。

拓展卡尔曼滤波的矩阵定义：

X: 代表目标的状态

$$[x, y, z, yaw, v_x, v_y, v_z, v_{yaw}, r]^T$$

Z: 代表观测到的数据

$$[x_a, y_a, z_a, yaw_a]^T$$

f: 状态转移矩阵

$$[x + v_x * dt, y + v_y * dt, z + v_z * dt, yaw + v_{yaw} * dt, v_x, v_y, v_z, v_{yaw}, r]^T$$

h: 状态空间到测量空间的映射矩阵

$$[x_{pri} - r_{pri} * \cos(yaw_{pri}), y_{pri} - r_{pri} * \sin(yaw_{pri}), z_{pri}, yaw_{pri}]^T$$

F: 状态空间转移关系的雅可比矩阵

$$\begin{bmatrix} 1 & 0 & 0 & 0 & dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

H: 状态空间到测量空间的映射关系对状态量的雅可比矩阵

$$\begin{bmatrix} 1 & 0 & 0 & r * \sin(yaw) & 0 & 0 & 0 & 0 & -\cos(yaw) \\ 0 & 1 & 0 & -r * \cos(yaw) & 0 & 0 & 0 & 0 & -\sin(yaw) \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Q: 过程噪声矩阵

$$\begin{bmatrix}
 q_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & q_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & q_z & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & q_{yaw} & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & q_{vx} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & q_{vy} & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & q_{vz} & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{vyaw} & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_r
 \end{bmatrix}$$

R: 测量噪声矩阵

$$\begin{bmatrix}
 r_x & 0 & 0 & 0 \\
 0 & r_y & 0 & 0 \\
 0 & 0 & r_z & 0 \\
 0 & 0 & 0 & r_{yaw}
 \end{bmatrix}$$

预测和更新公式:

预测:

$$\begin{aligned}
 X_{pri} &= f \\
 P_{pri} &= F * P_{post} * F^T + Q
 \end{aligned}$$

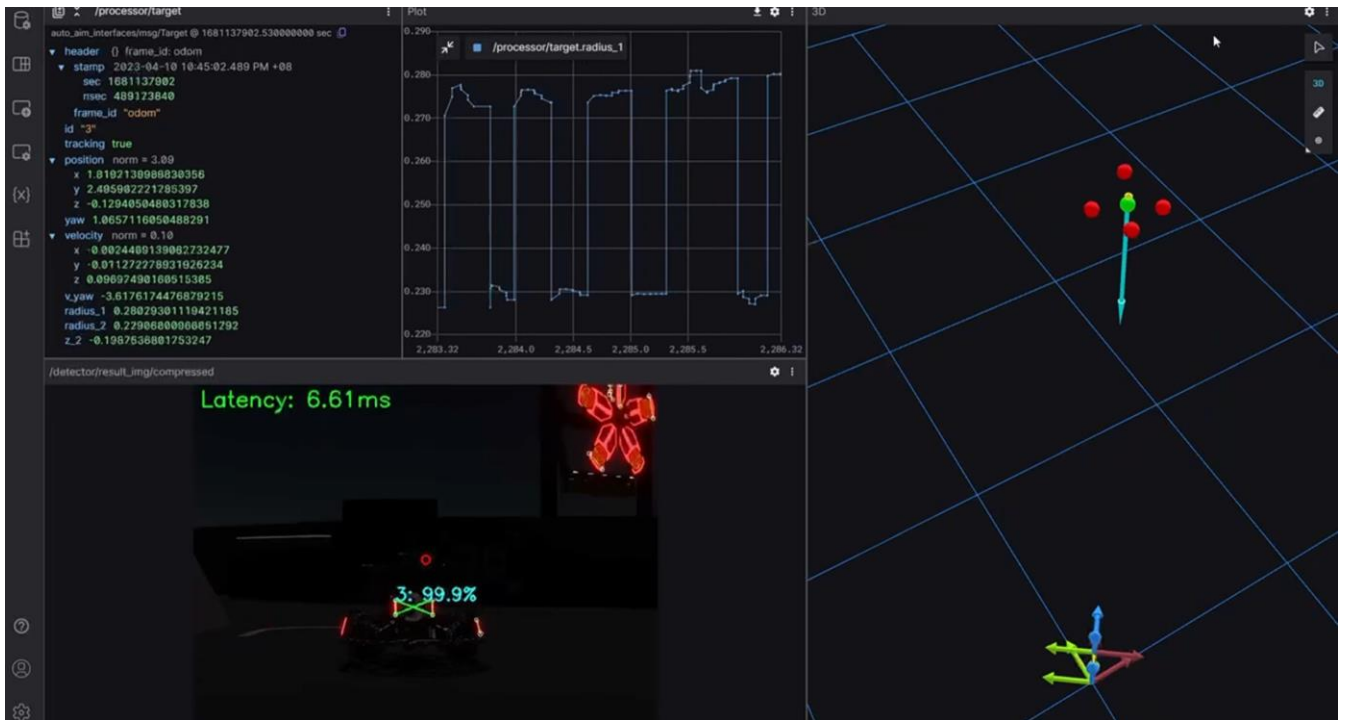
其中 P 为状态量的协方差矩阵

更新:

$$\begin{aligned}
 K &= P_{pri} * H^T * (H * P_{pri} * H^T + R)^{-1} \\
 X_{post} &= X_{pri} + K * (Z - h) \\
 P_{post} &= (I - K * H) * P_{pri}
 \end{aligned}$$

其中 I 为 9*9 的单位矩阵

通过仿真器部署运行后实际效果：



<图 5.11：仿真部署可视化效果图>

1.5.4.1.6 serial_driver 节点

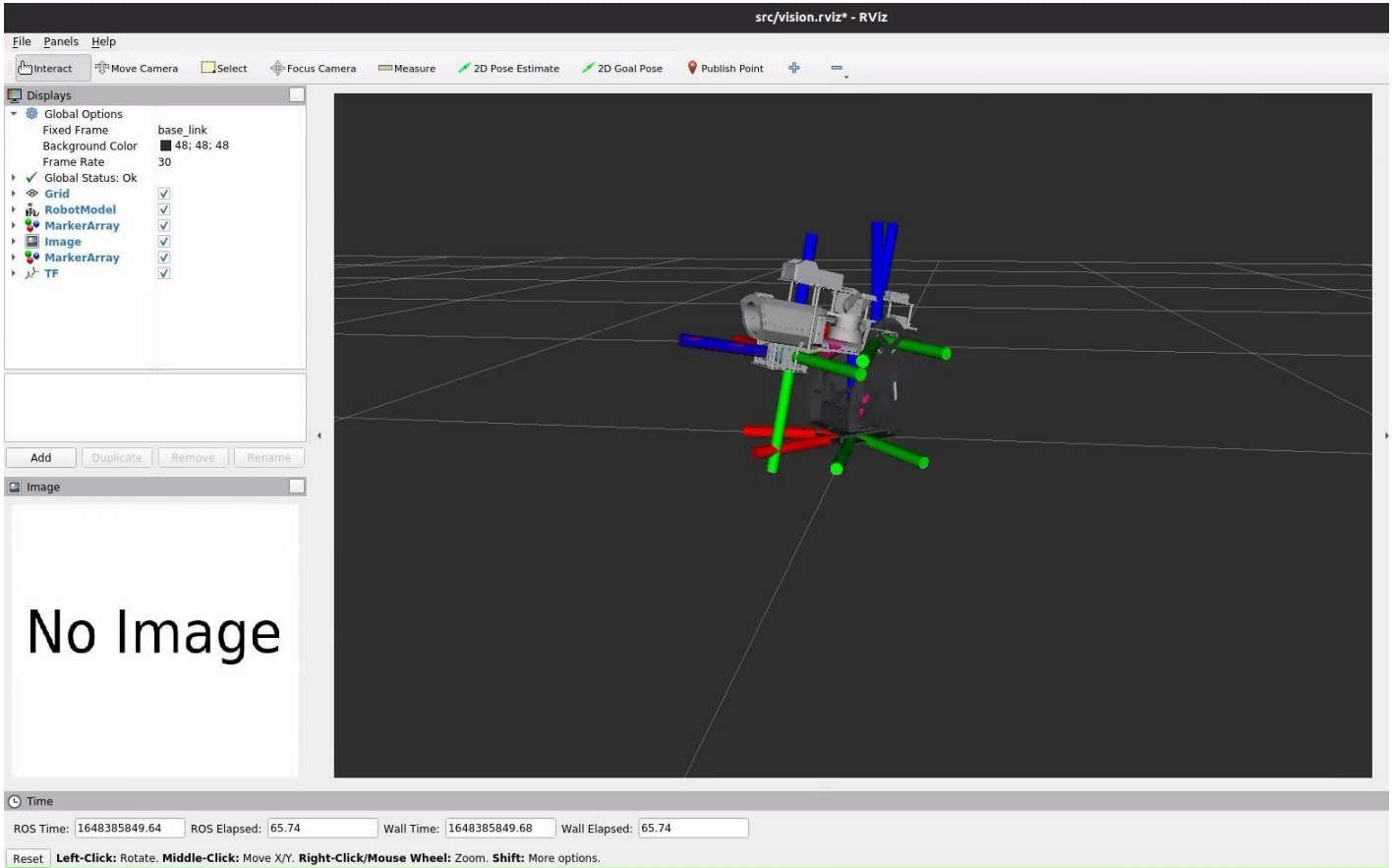
- 接受

接收来自 `processor` 节点的目标消息并通过串口传递到下位机。

- 发布

从下位机接收机器人当前的云台位姿，通过 `/joint_stats` 话题发送到 `robot_state_publisher` 节点，`robot_state_publisher` 节点接收到数据后对 `joint_state` 进行处理转化为 `tf` 信息后发布。

`robot_state_publisher` 发布的 `tf` 坐标信息及云台模型如下图：



<图 5.12: tf 坐标信息及云台模型图>

- 串口协议说明

上位机接收数据

数据	数据说明
Header	0x5A, 为数据包的帧头
Pitch	float32, 世界坐标系下云台当前的 pitch
Yaw	float32, 世界坐标系下云台当前的 yaw
Color	Bool, 机器人自身颜色, 1 为红, 0 为蓝

<表 5.9: 上位机接受数据>

上位机发送数据

数据	数据说明
Header	0x5A, 为数据包的帧头

id	string, 目标的数字
tracking	bool, 代表当前是否锁定目标
x	float32, 目标在世界坐标系下的 x 坐标
y	float32, 目标在世界坐标系下的 y 坐标
z	float32, 目标在世界坐标系下的 z 坐标
vx	float32, 目标在世界坐标系下 x 方向的速度
vy	float32, 目标在世界坐标系下 y 方向的速度
vz	float32, 目标在世界坐标系下 z 方向的速度
yaw	float32, 目标在世界坐标系下的倾斜角度
vyaw	float32, 目标旋转的角速度
radius_1	float32, 目标其中一组装甲板相对中心的半径
radius_2	float32, 目标另一组装甲板相对中心的半径
z_2	float32, tracking 中的装甲板的上一块装甲板的 z 轴位置

<表 5.10: 上位机发送数据>

1.5.4.2 能量机关算法

本部分封装为一个 ROS 包，作为一个功能模块参与到整个机器人自瞄系统的工作中

1.5.4.2.1 接口说明

订阅来自相机或视频的图像流进行能量机关的识别，识别完成后根据图像和时间戳信息进行预测，直接将最终击打目标信息发送到 `serial` 节点。

- **订阅：**

相机参数 `/camera_info`

彩色图像 `/image_raw`

云台位姿 `/tf`

- **发布：**

最终预测的目标位置 `/processor/target`

1.5.4.2.2 任务分析

能量机关的击打是 RM 赛事一直以来的一大特色，在对抗赛中击打能量机关获得的增益往往能成为队伍取胜的强大推力，在赛场上具有极高的战略地位。

23 赛季的能量机关机制，主要强调一个准字，相比 21、22 赛季，其改动主要体现在增益机制和灯效。

- **机制方面：**

小能量机关除提供 25%防御增益外还提供经验转化，比赛前期高效击打能量机关能够帮助队伍快速积累优势，加快比赛节奏；大能量机关仍提供攻击增益，但本赛季的攻击增益根据击打能量机关命中的环数提相应的攻击和防御增益，击打环数越高则增益幅度越大，同时在能量机关机制中，双方同时击打能量机关时，即使一方率先完成击打，若对方在一定时间内也完成击打且环数更高，则环数更低方仅能获得 50%增益。这一改动大大提高了射击精度和算法预测精度能够带来的收益上限，也对发射机构和视觉部分提出了更高的要求。

- **灯效方面：**

能量机关在保持了基本的“风车”元素的前提下灯效布局发生了较大的变化，其复杂程度和图案细节都有一定提升，基本宣告了以往赛季的识别思路的死亡，对一直以来使用 openCV 传统视觉进行识别的队伍提出了新的挑战，因此能量机关的识别也成为了本赛季能量机关项目的重要部分之一。

1.5.4.2.3 实现思路

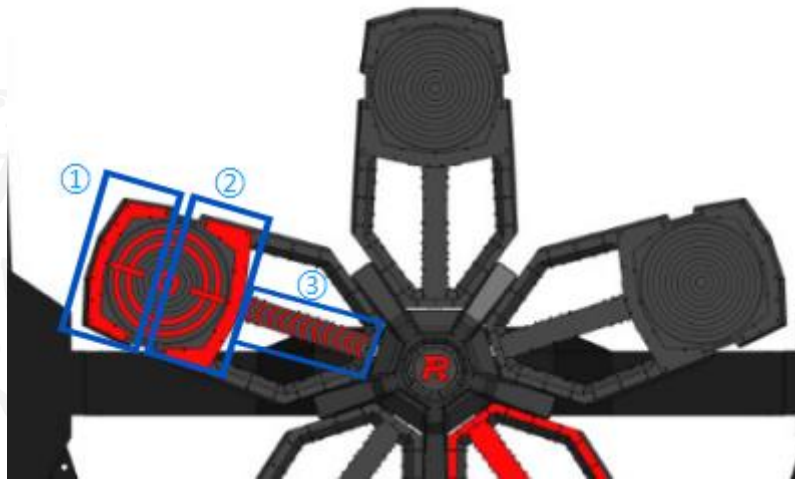
小能量机关转速为匀速，只需识别出能量机关并获取当前角度，利用获取图片到发弹的延迟时间即可完成预测；

大能量机关的转速为一给定范围内的随机正弦函数，预测击打的主要思路是解算出角速度正弦函数的各参数，然后根据函数相位和当前时间，利用图像获取时刻到预测击中的时间差对速度进行积分作为预测的角度差值，当前角度加上角度差值即为预测角度，从而预测出弹丸击打扇叶时扇叶的位置。

1.5.4.2.3.1 识别

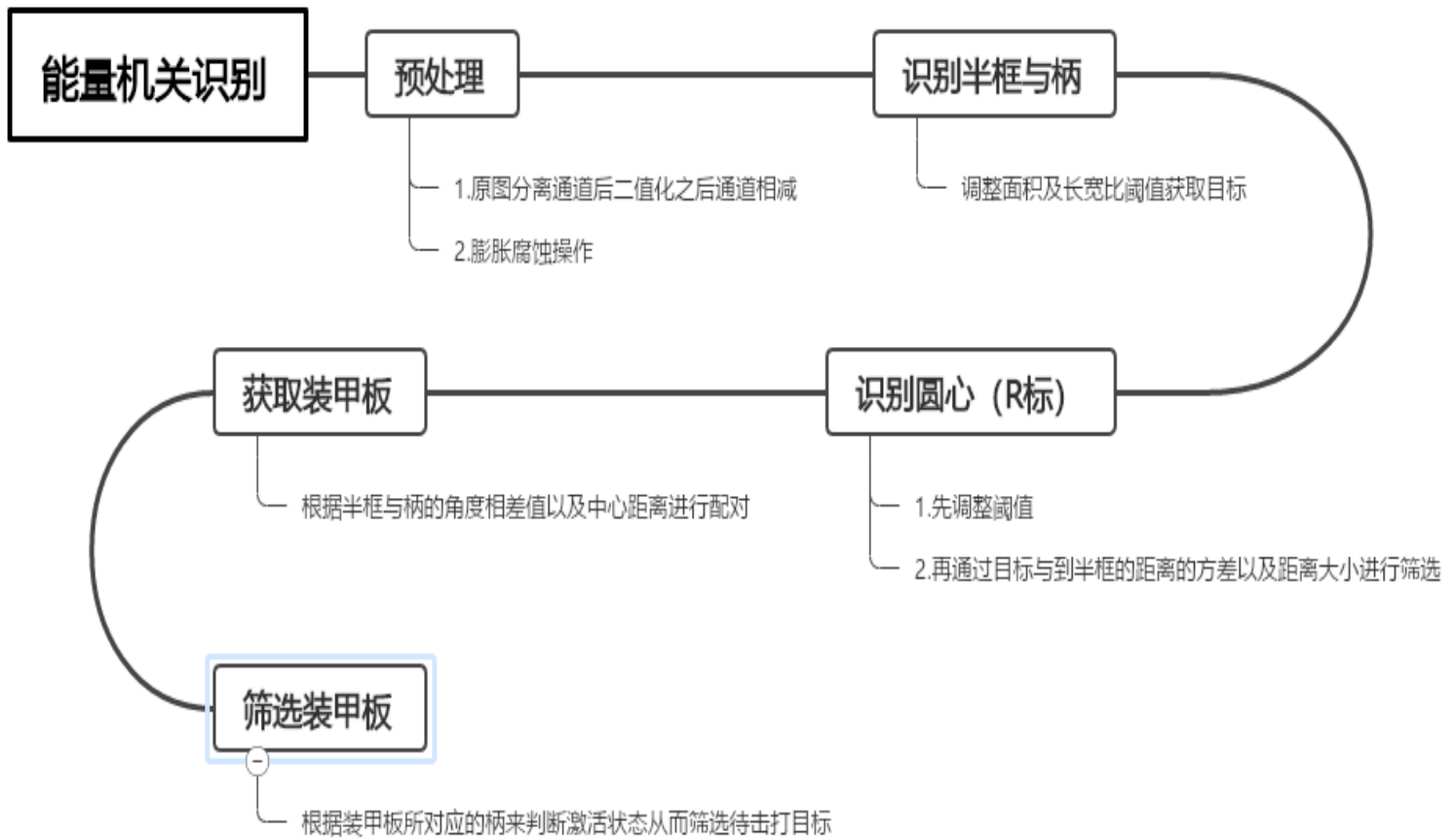
队伍视觉组深度学习方向基础较为薄弱，因此本赛季目前仍然考虑使用 openCV 传统视觉的方案进行能量机关识别。

能量机关未激活叶片的灯效主要由装甲模块的半框（如图中①）、柄（如图中②和③组合而成）和装甲模块中间瞄准灯效四个部分组成，主要针对装甲模块内外框进行识别并以其作为参考进行位置解算。



<图 5.13: 风车灯效组成分析图>

基本流程：



<图 5.14：能量机关识别流程图>

场地道具制作条件和时间有限，灯效颜色亮度无法完全均匀，在预处理方面对参数要求较为苛刻，但在合适的参数下也能达到较为稳定的识别和跟随效果。

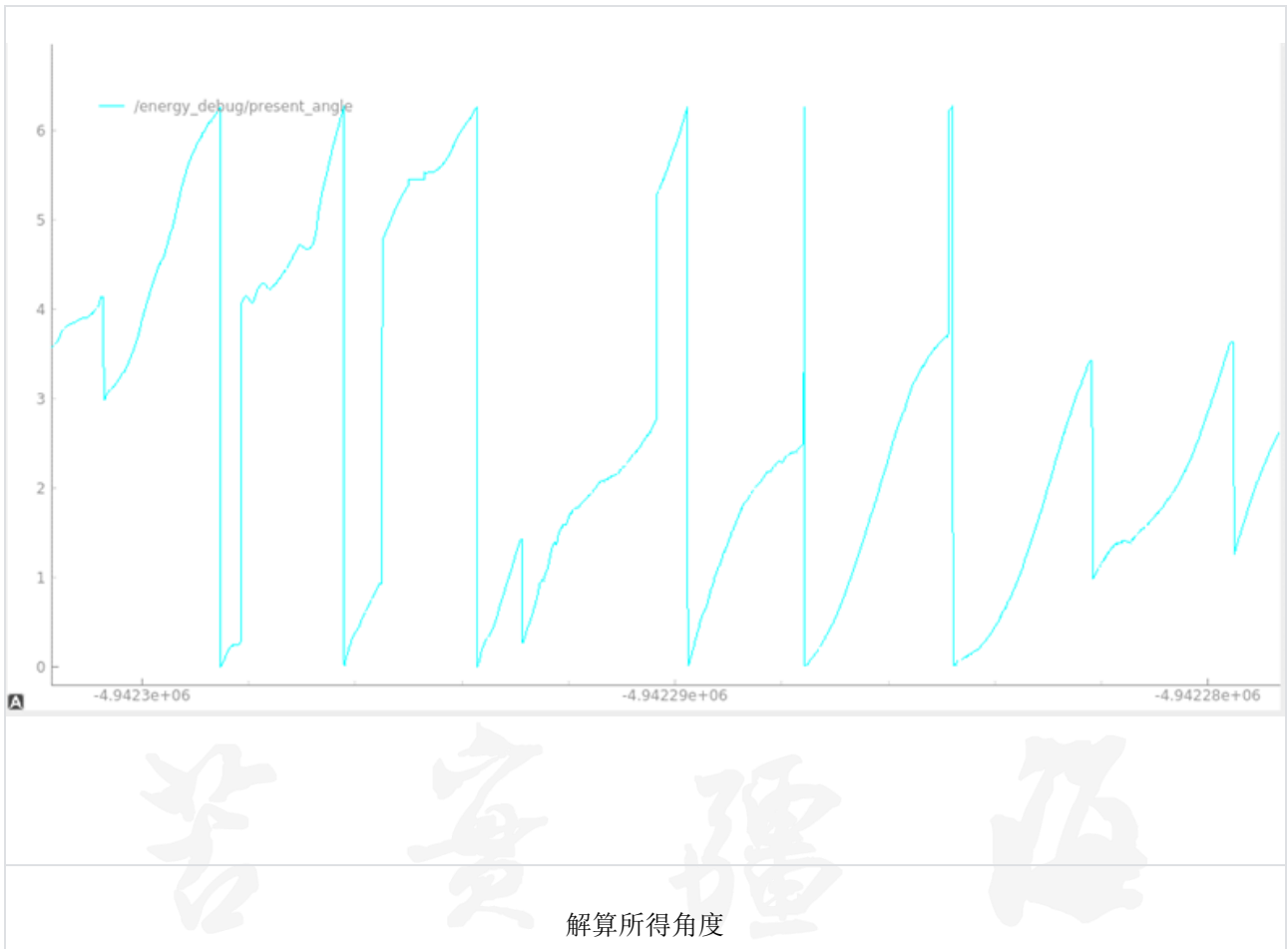


<图 5.15: 预处理效果图>

1.5.4.2.3.2 预测

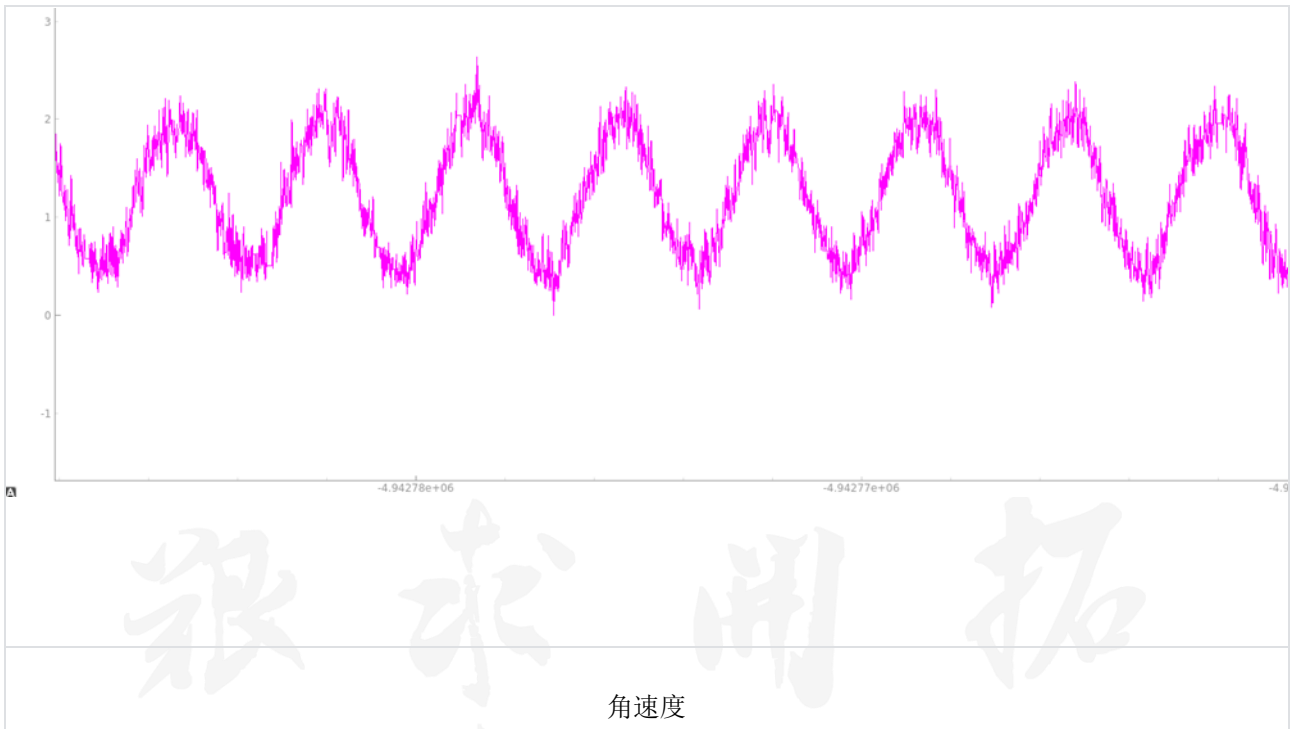
1. 角速度提取

由于步兵在击打点击打能量机关时正对能量机关，即使在跟随扇叶时云台位姿发生改变，能量机关扇叶在图像平面上的投影可能发生轻微畸变，但在 6-7 米的距离下，云台跟随装甲板运动时的运动幅度对图像上装甲板形状的影响很小，误差在可以接受的范围内，因此考虑直接在图像平面内，利用圆心和装甲板横纵坐标，以水平向右为极坐标轴，求解装甲板与坐标轴的夹角，作为装甲板当前角度。如装甲板图像坐标为 (x_1, y_1) 圆心坐标为 (x_2, y_2) ，则当前角度即为 $\arctan((y_2 - y_1)/(x_1 - x_2))$ （此处坐标系为 openCV 图像坐标系）；



<图 5.16: 角度解算效果波形图>

在获取角度后，取当前时刻与上一时刻的角度差值与两帧之间的时间差作商，作为当前时刻的角速度，当角速度较上一帧变化较大时判断是否属于装甲板正好经过零点或者激活后下一叶亮起的情况并针对跳变进行处理，得到角速度后存入 vector



<图 5.17: 角速度解算效果波形图>

2. 函数求解

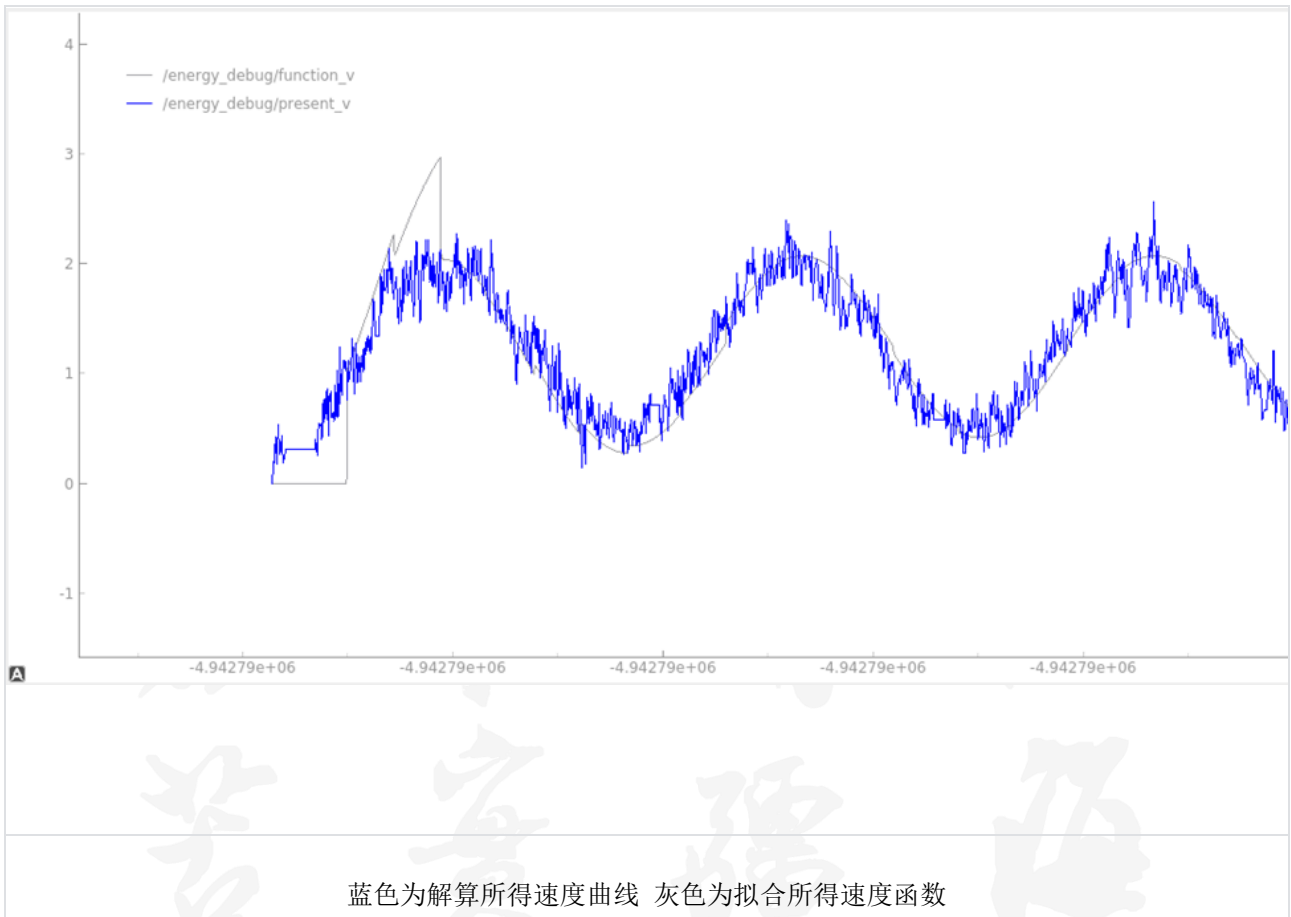
目前使用的求解方案为利用历史角速度数据进行最小二乘拟合，求解角速度。

已知求解的目标函数为一形如 $v = a * \sin(\omega * t + \varphi) + b$ 的三角函数，函数中各参数起始时在给定范围内随机生成且 a 与 b 相关，因此构造一个最小二乘问题以估计其中的未知参数 a 、 ω 和 φ ，得到该最小二乘问题的代价函数：

$$\min_{a, \omega, \varphi} \frac{1}{2} \sum_{i=1}^N \|v_i - (a * \sin(\omega * t_i + \varphi) + (2.090 - a))\|^2$$

由此，该函数求解任务便转化为一个非线性优化问题，目标即为使上式值最小。

该方法速度函数随角度数据实时更新，在函数求解的准确度和稳定性方面都有一定优势，理论上一个速度周期内（3.3s 左右）稳定能得到一个可靠的速度函数，实测 2s 即可得到一个较稳定的函数，配合基于图像的角度提取，视野中出现能量机关 2s 后即可开始击打，相较上赛季的椭圆拟合方法，在目标位置解算准确度方面，对相机成像精度有了更高的要求，但大大提高了击打速度。



<图 5.18：速度解算效果波形图>

在测试过程中出现拟合结果有相位延迟、迭代运算速度不理想导致进程阻塞等问题；前期解决方案主要是沿用拟合库，但迭代拟合与接收新数据异步进行，在拟合更新过程中使用上一次拟合更新结果的函数进行预测。后期测试发现抽样使用速度数据，降低拟合所用数据量，对拟合效果不会有太大影响，但大大减少了单次拟合所需时间。

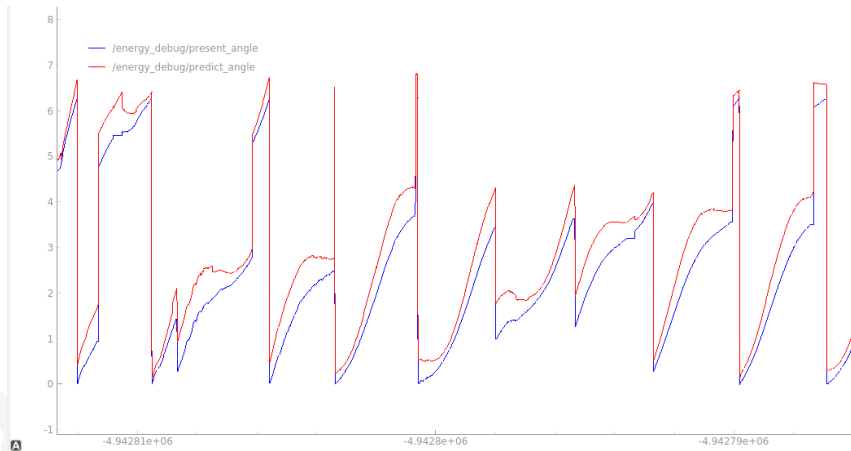
3. 目标预测

使用当前角度加上对求得的速度函数对时间差值进行积分，即可得到预测的目标角度

在成功获取角速度函数后直接根据圆心坐标和能量机关在图像坐标下的半径，预测目标在图像坐标系下的中心坐标，由于只需解算目标中心的坐标而不需要目标装甲板的位置（旋转矩阵），因此直接使用当前帧识别到的装甲板角度加上角度差值得到预测角度，根据圆心的图像坐标、能量机关半径和预测的角度，得到预测目标中心点的图像坐标，以平移后

的四个装甲板顶点作为 PnP 解算的四个特征点进行解算，作为最终击打目标点。

当前角度和预测角度波形图（蓝色为当前角度，红色为预测角度）：



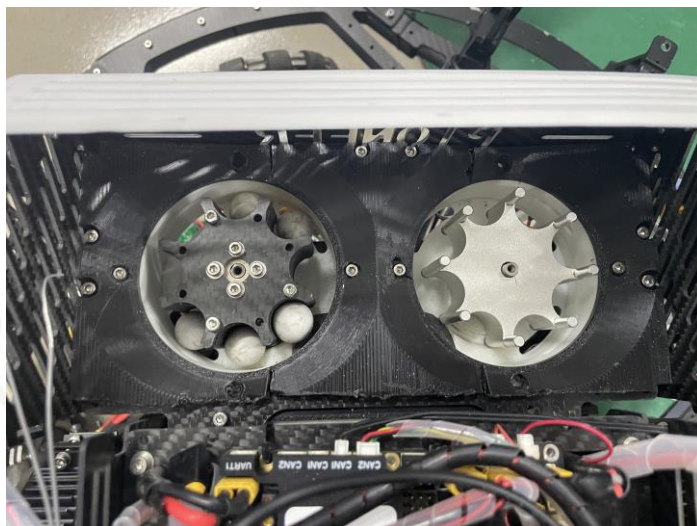
<图 5.19：角度预测波形图>

敢
不
苦
奮
創
新
疆
破
去
海
立
浪

1.6 研发迭代过程

1.6.1 研发迭代记录

1.6.1.1 拨盘迭代



〈图 7.1：碳板打印件拨爪（左）和 cnc 拨爪（右）〉

作为发射机构供弹链路最重要的一环，拨爪的稳定性与可靠性显得至关重要，为此我们预先使用前一代稳定的 cnc 拨爪作为对照对碳板一打印件拨爪进行试验对比。

通过定性实验我们发现碳板一打印件拨爪在拨弹阻力、对弹仓弹丸影响等方面均优于上一代 cnc 拨爪。

1.6.1.2 全向轮底盘迭代

第一版全向轮底盘是 2022 赛季备赛国赛期间研制出来的，当时由于工期紧张，底盘基础构型借鉴哈深 2021 赛季开源的全向轮步兵，在局部上适配本土零件和云台及“呼啦圈”方案进行适配优化。在完成第一版全向轮底盘的组装后我们进行了大量测试并发现以下问题

机构	第一版问题表现	原因分析	迭代方案和考量
防撞	外围“呼啦圈”防撞方案在实战中并未起到理想效果。	迫于裁判系统安装位置和云台俯仰角需求，该防撞高度小于其他正常地面单位防撞高度。	舍弃“呼啦圈”方案使用圆形底盘足以满足设想使用需求，“呼啦圈机构”存在冗余且效果不佳故舍弃。

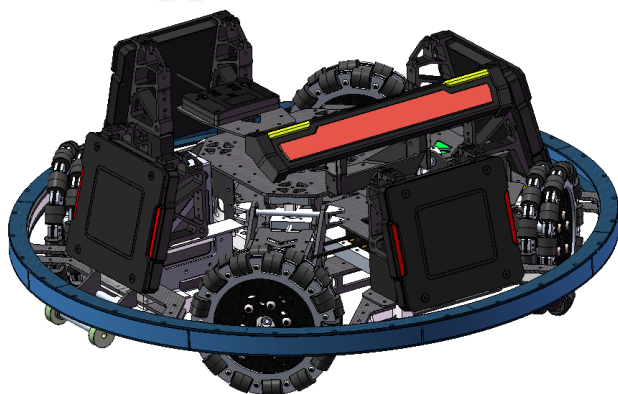
轮组	1.轮组平行四边形结构上的螺栓容易松动； 2.轮组悬挂行程短	1.平行四边形结构的固定螺丝使用螺纹配合，在运动时容易松动脱落 2.设计之初并未慎重考虑	1.放弃使用原有旧件，重新加工新的铝件采用螺丝螺母配合 2.设计增加悬挂压缩量。
Yaw 轴	云台在运动和下台阶时晃动严重	固定电机与底盘框架仅使用3mm 碳板固定，刚度强度不足	为配合大质量云台，设计轴承承担云台重量并保证结构强度。

<表 7.1：初版问题迭代表格>

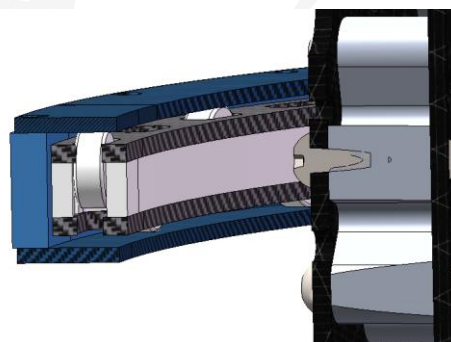
以下逐步阐述各个问题原因分析及迭代过程。

(1) “呼啦圈”防撞方案

最外围的“呼啦圈”防撞方案，我们研发时设想是在复杂的环境下即使外圈被敌方单位或着墙体卡住或固连，内圈也能保证“小陀螺”的功能正常使用，而“呼啦圈”作为一个大轴承隔绝内外圈连接。

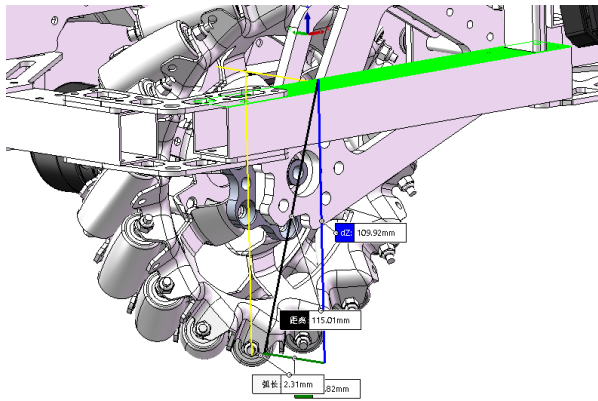


<图 7.2：“呼啦圈”示意图>

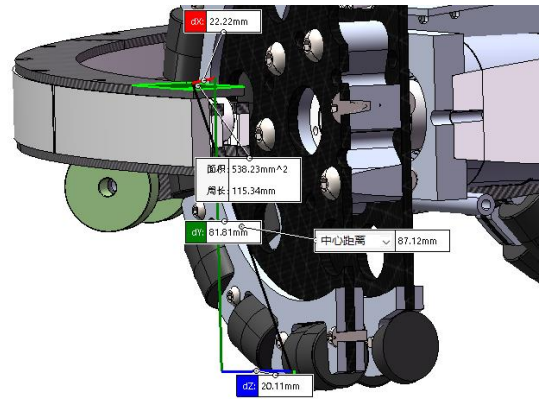


<图 7.3：“呼啦圈”原理示意图>

但通过训练赛实测发现，由于该全向轮底盘防撞高度低于其他单位防撞高度，固“呼啦圈”不能成为步兵的最外圈接触其他单位，则该机构完全没有起到设想的功能效果。而因为该机构是处于机器人最外圈，如果我们要加高它的高度，装甲板因为规则范围遮挡要求也要进行加高，在图纸上我们发现继续加高装甲板会影响云台的俯仰角度，综合考虑下，我们在未来第二版全向轮底盘上将该方案舍弃。

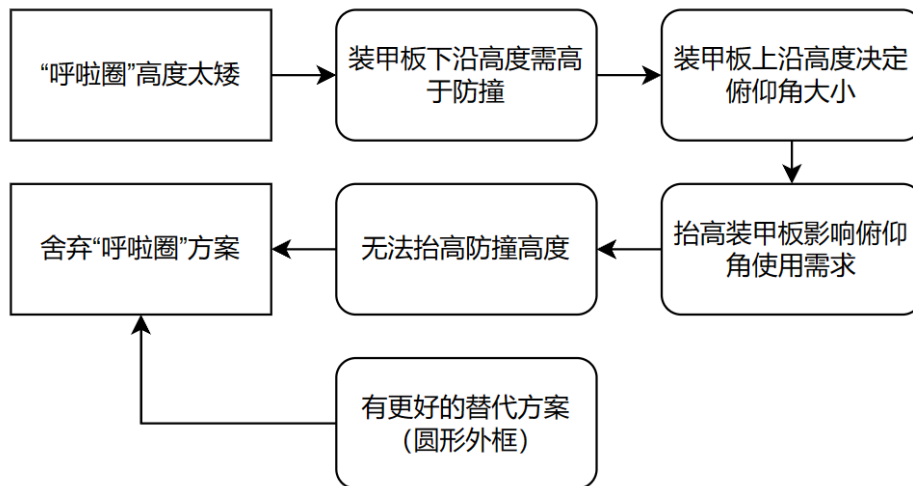


<图 7.4：队内麦轮步兵防撞高度>

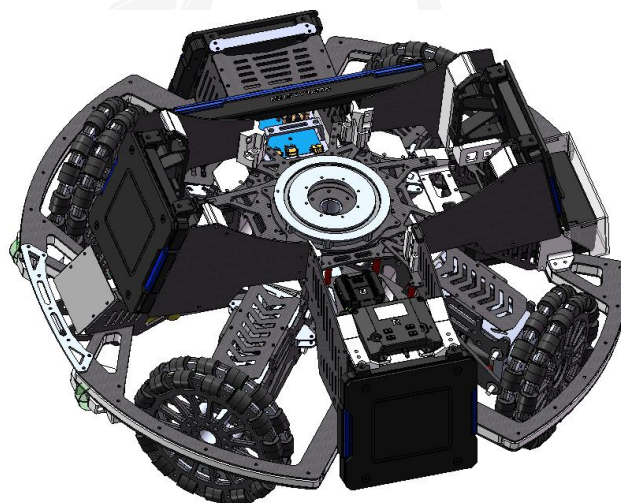


<图 7.5：“呼啦圈”防撞高度>

另外，在备赛期间与其他学校队伍交流发现，即使不使用该机构“呼啦圈”，只要防撞外圈是圆形的，就足以满足当初设想的使用需求。



<图 7.6：迭代方案思考逻辑图>



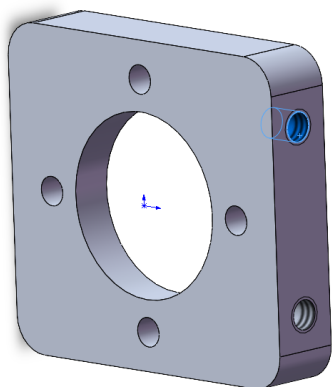
<图 7.7：第二版全向轮防撞示意图>

第二版底盘在外圈舍弃了“呼啦圈”方案，直接采用圆形底盘，经测试和赛场检验发现

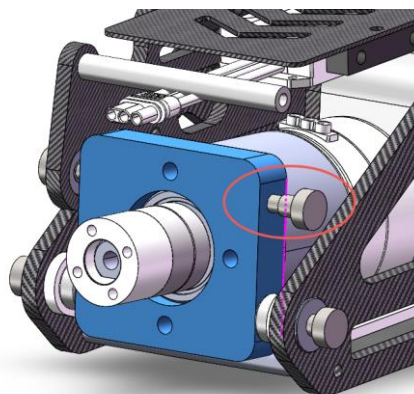
其足以满足设想需求。

(2) 轮组方案

第一版全向轮底盘轮组电机座迫于工期和成本原因，使用的是实验室存留的电机座（下图左），在设计中使用使用该电机座的 M5 螺纹孔配合螺栓作为轮组平行四边形结构的一个固定转点（下图右）。但在实际训练赛测试过程中发现该点的螺栓极容易松落，但拧紧该螺丝又会导致转点的平行四边形运动阻力加大，故没有良好的办法解决该方案的问题。

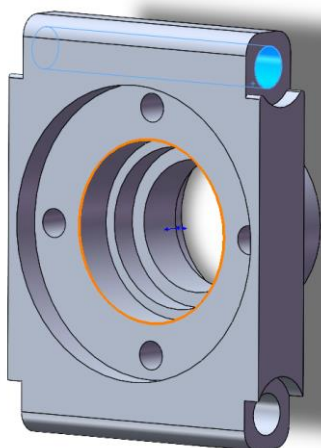


<图 7.8: 旧件>

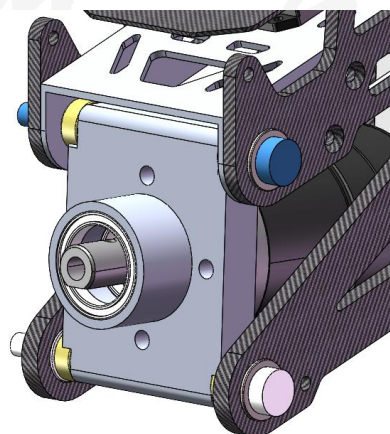


<图 7.9: 松动螺栓安装示意>

在第二版全向轮底盘中，重新加工了电机座（下图左），加工的孔能直接穿过直径为 6 的塞打螺栓并使用螺纹固定，经实测没有发现松动问题



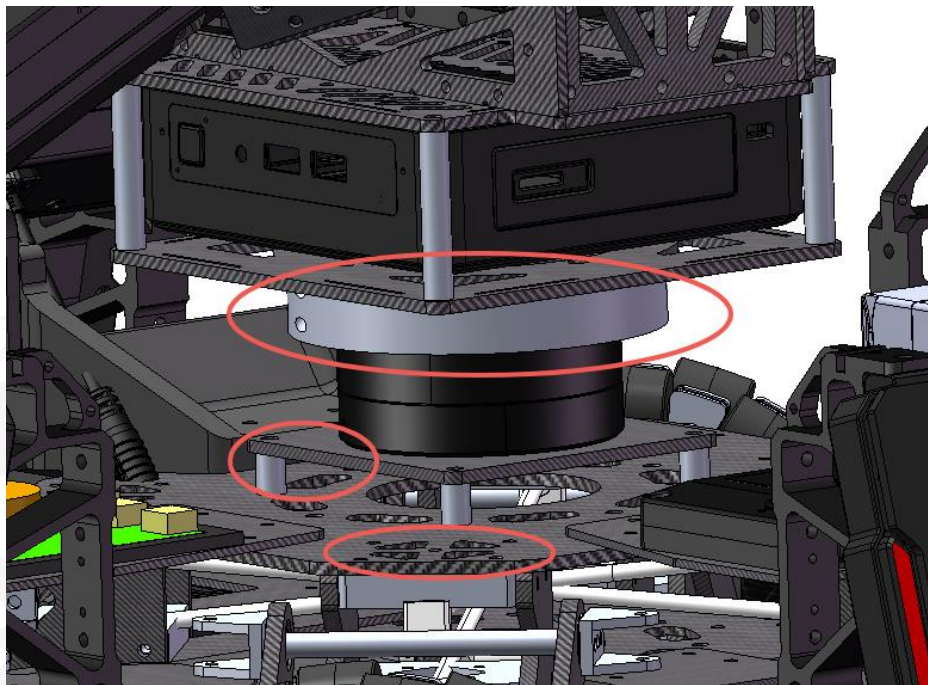
<图 7.10: 新电机座>



<图 7.11: 新电机座螺栓安装示意>

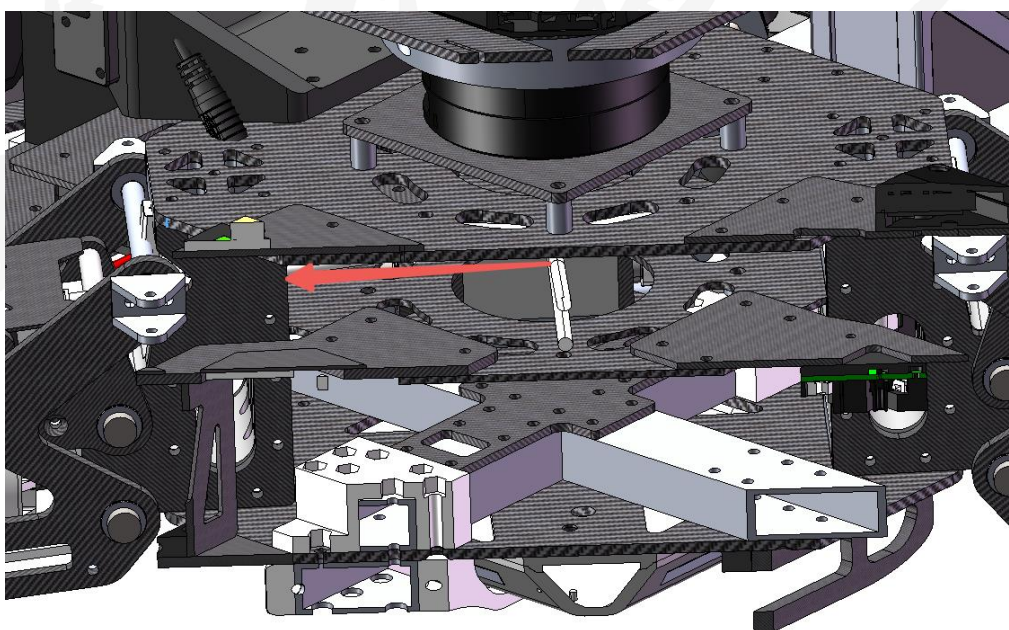
(3) yaw 轴方案

第一版全向轮底盘的 yaw 轴仅使用一块 3mm 碳板固定 6020 电机定子，并且通过 4 个铝柱固定在底盘机架上，在实际使用中发现 yaw 有极大晃动并且铝柱螺丝容易脱落。



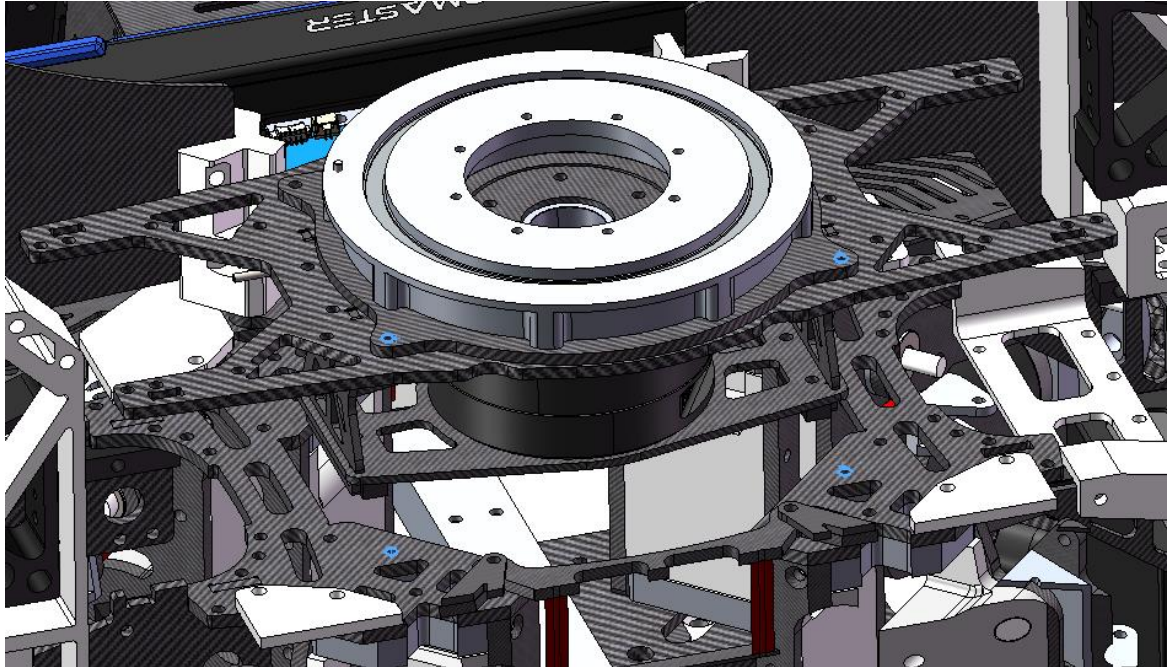
<图 7.12: yaw 轴零件组成图>

其次，yaw 轴固定的底盘机架为大面积的 3mm 碳板，并且 yaw 受力点里真正固定的框架距离较远，所以 yaw 轴无法满足刚度需求。



<图 7.13: 大面积 3mm 固定碳板与连接部位示意图>

在第二版全向轮底盘中使用薄壁深沟球轴承作为 yaw 轴方案保证结构的刚度和精度，在与底盘机架的连接上，直接通过螺栓将 6mm 厚碳板与轮组固定座连接解决了连接刚度的问题。

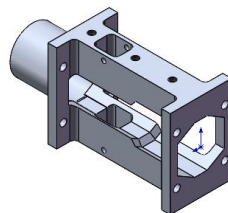


<图 7.14：第二版 yaw 轴连接示意图>

1.6.1.3 无枪管迭代发射测试

在联盟赛与分区赛中，我们发现我们当下的有枪管方案与无枪管方案都会出现弹道因单发限位安装出现大幅度弹道漂移的情况。经排查后发现是 18mm 枪管内径过大导致 17mm 弹丸在接触摩擦轮时姿态不确定从而导致弹道不确定。

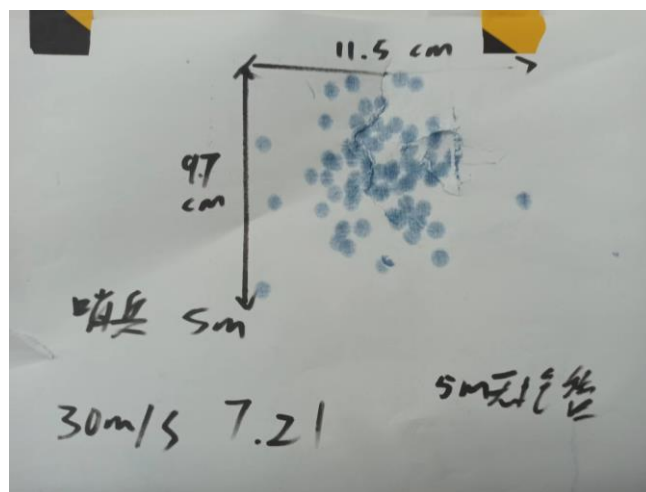
为方便测试，我们最先采用一体光固化限位固定进行测试，限位距离分别取 17.2mm、17.1mm、17.0mm 与 16.9mm 并各制作三个避免加工精度带来的影响。



<图 7.15：无枪管发射图纸>

经安装到步兵上测试发现限位距离为 17.1mm 时弹丸仍可小幅度偏移幅度，而在 17.0 与

16.9mm 间距下会出现部分弹丸堵塞的现象但精度都比上一代发射更好，最终选择 17.1mm 作为弹丸定心距离。经改进后弹道在高弹速下表现较为稳定且可在 7m 距离将散布缩小至一块小装甲板内。

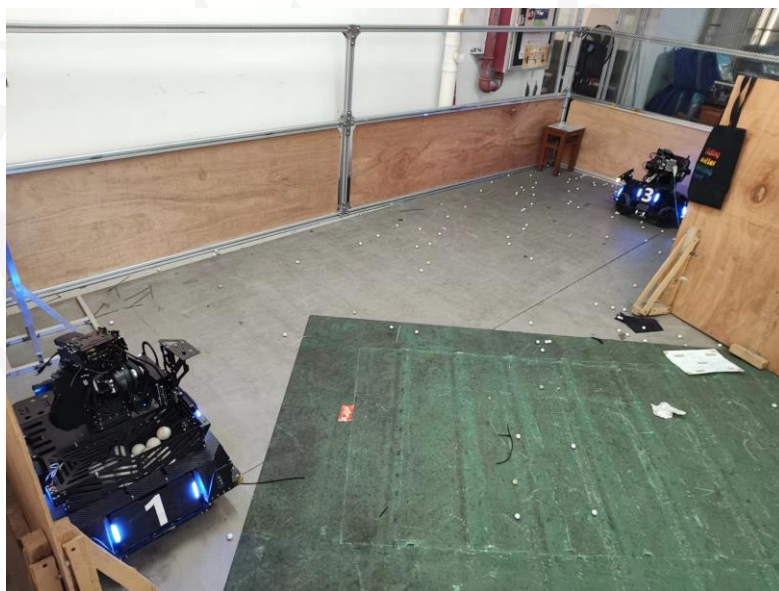


<图 7.16: 无枪管 5m 散布>

1.6.2 测试记录

1.6.2.1 发射机构测试

1.6.2.1.1 双枪管发射机构测试



<图 8.1: 双枪管发射测试>

在 5m 处对使用双枪管步兵机器人对英雄机器人进行击打并拍摄视频便于后续统计。弹速通过裁判系统的测速模块进行测量。

得出结论：

发射机构卡弹测试：400 发弹丸连续发射无卡弹

发射机构散布测试：5 米距离上对大装甲板命中率为百分之 95

发射机构弹速稳定测试：28m/s 设定弹速下弹速变化幅度为 $\pm 0.4\text{m/s}$

1.6.2.1.2 无枪管发射测试



<图 8.2：无枪管发弹测试平台>

通过使用相同无关变量，只改变枪管长度作为变量实现控制变量法对有无枪管发射散布做定量分析。我们分别在 3.5 米、5.5 米与 8 米的距离下进行了多组测试。

其中一组 5.5m 距离下无枪管散布（下左图）与有枪管散布（下右图）



<图 8.3 无枪管散步>



<图 8.4：有枪管散步>

为了更加直观观察到 17mm 弹丸落点位置，我们采用“A4 纸—复写纸—A4 纸”的三层三明治结构定量分析散布。

最终得出结论：无枪管发射精度较有枪管发射精度更胜一筹。

1.6.3 版本迭代过程记录

1.6.3.1 机械部分

版本号或阶段	功能或性能详细说明	完成时间
V1.0	第一版全向轮整车完成	2022.8.2
V1.1	第一版摩擦轮发射测试平台	2022.8.6
V1.2	第二版轮组设计完成	2022.10.18
V1.3	第二版轮组测试平台搭建	2022.10.24
V1.4	第一版通用全向轮底盘框架设计	2022.11.12
V1.6	第二版全向轮步兵特化底盘设计	2022.12.1
V1.7	第一版双枪管弹链设计测试	2022.12.16
V1.8	第一版双枪管云台总体设计	2022.12.20
V2.0	第二版全向轮整车完成	2023.1.3
V2.1	全向轮底盘强度优化	2023.2.12
V2.2	全向轮保护板材增加	2023.2.19

1.6.3.2 电控部分

版本号或阶段	功能或性能详细说明	完成时间
V1.0	框架初步移植运行	2023.1.13
V1.1	底盘参数调节, 旋转平移正常	2023.1.18
V1.2	IMU 驱动移植, 解算代码	2023.1.25
V1.3	云台参数调节, pitch、yaw 轴正常控制	2023.2.5
V1.4	发射机构参数调节, 正常发射弹丸	2023.2.9
V1.5	云台与拨弹盘参数冲突及其他 bug 修复	2023.2.14
V1.6	裁判系统组装及全车硬件链接完成	2023.2.25
V1.7	视觉联调, 防卡逻辑调整, 弹道解算测试完成	2023.3.1
V1.8	自动瞄准弹速与预测时间自动获取	2023.3.4

版本号或阶段	功能或性能详细说明	完成时间
V1.9	根据裁判系统数据调控代码完成	2023.3.15
V2.0	参数调整, 代码优化, bug 修复, 增强稳定性	2023.4.1

1.6.3.3 视觉部分

版本号或阶段	功能或性能详细说明	完成时间
V1.3	将代码部署到 ros-humble 版本	2022-10-13
V1.4	回滚部分功能, 实现建议开火, 并使程序正常工作	2023-03-25
V2.0	修改程序逻辑, 实现整车状态预测	2023-04-10

1.6.4 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	第一版全向轮轮组连轴不稳定, 经常出现“掉轮子”问题	轮子与电机采用涨套件配合, 安装不规范导致配合不在涨套工作区间内	通过观察轮组连轴配合, 锁定问题原因。解决后没有出现状况	V1.0	机械工程师: 彭宇
2	步兵在摩擦轮打开后, 云台 pitch 控制产生抖动	摩擦轮和 pitch 轴电机共用 can 线及标识符, 控制时数据产生冲突	增加全局变量, 用于传输拨弹盘控制数据, 于云台控制同时发送	V1.5	嵌入式软件工程师: 赵俊杰
3	自动瞄准后, 弹丸散布较大	自瞄后 pitch 和 yaw 出现微小晃动, 影响弹道	重新调整 pitch 和 yaw 轴的 pid_leso 控制参数	V1.7	嵌入式软件工程师: 赵俊杰

1.7 团队成员贡献

姓名	基本信息	主要负责工作内容描述	贡献度
杨阳	光电信息科学与工程、大二、步兵机械结构设计负责人、步兵车组长、副队长	负责步兵机器人云台机械结构和底盘机械结构设计，负责统筹与推进车组进度	(机械: 70%)
张子熠	光电信息科学与工程、大二、机械组组长、步兵云台机械负责人	负责步兵机器人发射模块的测试与迭代。	(机械: 30%)
赵俊杰	光电信息科学与工程、大二、软件开发负责人	负责机器人嵌入式开发、及部分硬件设计	(控制: 80%) (硬件: 30%)
钟政旭	光电信息科学与工程、大二、电控组组长、步兵车组长	负责统筹推进组内电控进度及新系统的部署	(控制: 20%)
曾昭庆	光电信息科学与工程、大二、电控组组长	负责电容功率板的设计及测试，负责功率使用代码开发。	(硬件 70%)
蒋荣桀	信息工程、大二、视觉组组长	负责整车视觉算法方面代码部署与实现	(算法: 10%)
姚亭屹	信息工程、大三、视觉组成员	负责能量机关识别研发测试	(算法: 10%)
陈君	通信工程、大四、顾问	负责视觉新系统的研发测试	(算法: 30%)
肖佳扬	光电信息科学与工程、大一、视觉组成员	负责后续能量机关识别研发测试	(算法: 25%)
曾泽嘉	光电信息科学与工程、大一、视觉组成员	负责后续步兵基础视觉部署与调试	(算法: 25%)

1.8 参考文献

- [1]RM2021-哈尔滨工业大学(深圳)-南工骁鹰-全向轮步兵开源
- [2]RM2022-南京理工大学-Alliance-机械结构开源-步兵机器人
- [3]RM2021-西交利物浦大学-三摩擦轮双枪口步兵机械开源
- [4]RM2022-哈尔滨工程大学 Nooploop 创梦之翼-双枪平衡步兵机械开源
- [5]RM2021_深圳大学_RobotPilots 战队_自适应舵轮双枪步兵_机械开源
- [6]RM2020-上海交通大学-交龙战队-步兵机器人机械技术开源

1.9 技术方案复盘

1.9.1 赛场性能表现情况分析

1.9.1.1 赛场表现综述

通过赛场回顾，全向轮步兵在赛场上移动和发射等基础功能稳定，没有发生过违反规则扣血阵亡的现象；火力输出上即使选择弹速优先，双枪的构型使得发射时两个枪管的热量切换，实际也能保证较持久输出，在整车观测的自瞄系统和火控逻辑下保证弹无虚发快速的击杀敌方的单位。全向轮底盘的构型也保证了机器人在赛场上拥有不俗的自旋速度，对比自家麦轮步兵和全向轮步兵的表现来看，全向轮步兵的生存能力远大于麦轮步兵。

但该步兵在赛场上也暴露出不少问题，其一是通过性较差，外围保险杠高度过低导致步兵在车身不在水平姿态的时候容易碰撞导致翻车，简单而言就是水平面上各向接近角不能满足使用需求，并且前期导轮安装位置的问题导致步兵下台阶时如果姿态不好容易翻车。另外，该步兵由于轮子增大，在坡上启动时或低速上坡时表现不佳导致步兵在通过盲道和上坡时没有满足实际需求。

1.9.1.2 方案落地情况

功能定义和需求	采取设计方案	赛场情况反馈
优秀稳定的输出能力	1.上供弹双枪方案 2.整车观测视觉识别与火控	发射系统稳定执行，没有出现过卡弹，尿弹，超热量超射速等现象，但有少量蹭枪管情况发生；火力输出强劲，自瞄和火控系统加持下命中率高，但有少许情况下火控无法打出弹丸；
整车质心降低且轻量化设计	1.yaw轴下置 2.全车轻量化设计	检录重量 17.25kg，在赛场上没有出现过质心不合理导致翻车情况
防止与场地或机器人固连	外围圆形投影防撞设计	比赛中未出现过固连现象
生存能力	1.全向轮底盘构型 2.高低装甲板错开	赛场上全向轮自旋速度快，生存能力较高，在分区赛时实现数局 0 死亡战绩

1.9.2 赛场性能表现与规划对比分析

技术	规划	赛场性能表现分析
机械	<ol style="list-style-type: none"> 1. 使用全向轮底盘，做到拥有更快的自旋速度 2. 使用切枪管云台构型，提供持久的火力输出 	<ol style="list-style-type: none"> 1. 底盘机械结构正常运作，但底盘通过性较差，在下台阶时容易翻车 2. 切枪管装置稳定性不足，容易出现蹭枪管现象 3. 部分机械机构没有考虑翻车工况，在翻车时容易造成结构失效
硬件	<ol style="list-style-type: none"> 1. 研制电容功率控制板，做到使用电容上下 13° 和 15° 坡无阻力 2. 滑环板沿用旧板，期望机器人基础硬件稳定 	<ol style="list-style-type: none"> 1. 功率控制板可以实现上下坡自如，但由于电容容量太小，在复杂的使用场景下，有可能无法实现顺利上坡。另外，功率板稳定性较差，在赛场上容易出现底盘无法控制情况 2. 机器人基础硬件稳定，步兵硬件线材、焊接、接口问题基本没有出现，能保证步兵在赛场上的正常供电和信号传输
软件	<ol style="list-style-type: none"> 1. 新视觉数据的处理和火控 2. 新超电功率限制 3. 双枪管及切枪管步兵的控制 4. 增加诸多新功能与优化代码保护 	<ol style="list-style-type: none"> 1、新视觉与火控的配合显著的提升了命中率，降低了弹丸的使用量，在赛场中大多数情况能正常发挥，但是偶尔有出现打不出弹的问题，需要后续 debug，同时选板中没有考虑到预测量导致目标移动时选板可能出错，视觉数据出错或不精准时的保护不够完善，在迅速接近目标时预测目标位置会出现在本车背后导致转头的问题还没有解决 2、超电控制能在功率板正常工作时保证不超功率，但是对于底盘功率无法很好的吃满，通过在底盘控制室增加了双环（整车速度和轮速）的 pid 可以一定程度上更完善的消耗底盘功率 3、双枪管和切枪管步兵都能正常控制，赛场上也都正常 4、新功能基本都能在赛场上正常使用，一些冗余或是不完善的代码保护在修改后解决了很多 bug
算法	<ol style="list-style-type: none"> 1. 部署整车观测性系统，能做到对机器人和前哨站等单位装甲板识别和预测，对任意状态的单位装甲板进行预测 2. 能量机关识别与预测 	<ol style="list-style-type: none"> 1. 视觉跟随在击打静态小陀螺单位效果极佳，场上发挥正常稳定，但对目标预测的收敛时间较长，实际赛场中当双方机器人有快速机动时视觉系统无法正常使用。 2. 能量机关识别算法无法在赛场上使用，有出现误识别情况。

1.9.3 经验总结

1.9.3.1 测试

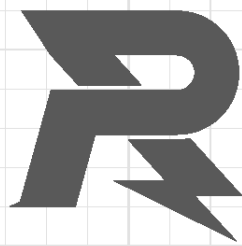
测试强度不能完全贴合赛场实际强度，应当在尽可能的模拟赛场场地环境下进行高强度的测试并且充分锻炼操作手熟悉场地和机器人性能。在测试中对新研发功能需要大量测试才能保证赛场上不出现问题。

对于机器人常规功能，应当做好合理的测试用例，并且在逐年备赛中根据赛场真实反馈不断优化。

1.9.3.2 硬件

本赛季进行了新滑环板设计的尝试 type-C、网口、sff8643 均有不同问题导致实际效果不佳或是无法实装，上赛季使用的金手指小板方案在充分保护下仍能稳定使用，在对滑环板重新排线设计之后，其面积缩小至 50mm*50mm，可轻松在底盘和云台安装，并且能通过不同的焊接处理简化理线，后续可以沿用此设计，sff8643 应该还有升级尝试的可能性

古 貴 疆 海
奮 創 破 立
鬥 新 志 浪



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