

# RoboMaster 2022 平衡步兵技术报告



## 目录

1. 其他学校平衡步兵机器人分析综述 .....	5
1.1 机械部分 .....	5
1.2 视觉部分 .....	7
1.3 超级电容部分 .....	7
2. 机器人功能定义 .....	8
3. 机器人核心参数 .....	10
3.1 机械参数 .....	10
3.2 电路参数 .....	12
3.3 传感器参数 .....	12
4. 设计方案 .....	13
4.1 机械结构设计 .....	13
4.1.1 底盘部分 .....	13
4.1.2 云台部分 .....	23
4.2 硬件设计 .....	28
4.2.1 平衡步兵整车线路拓扑图 .....	28
4.2.2 超级电容 .....	28
4.2.3 C 板扩展板 .....	46
4.2.4 关键器件选型 .....	48
4.3 软件设计 .....	49
4.3.1 系统架构 .....	49
4.3.2 运行流程 .....	50
4.3.3 UI .....	51
4.3.4 平衡机器人的全向移动算法 .....	57

4.3.5 控制优化及创新点.....	60
4.3.6 软件测试流程.....	63
3.4.7 与完整形态的变化.....	66
4.4 算法设计.....	66
4.4.1 关于 YOLO 目标检测神经网络与装甲板自瞄算法的说明.....	66
4.4.2 关于能量机关辅助瞄准算法说明.....	75
4.4.3 弹丸发射弹道补偿.....	82
4.4.4 卡尔曼滤波.....	84
4.4.5 相机与 imu 的联合标定.....	84
4.4.6 四阶龙格库塔法解微分方程.....	85
4.4.7 Ceres 优化最小二乘法拟合能量机关以及外部逻辑的优化.....	88
4.4.8 拟合前的数据处理.....	89
4.4.9 粒子滤波.....	90
4.4.10 未采用的方案.....	95
4.4.11 方案优缺点及未来优化方向.....	97
5. 研发迭代过程.....	99
5.1 版本迭代过程记录.....	99
5.1.1 整体版本.....	99
5.1.2 机械部分.....	100
5.1.3 电控部分.....	101
5.1.4 硬件部分.....	102
5.1.5 视觉部分.....	103
5.2 重点问题解决记录.....	104
6. 团队成员贡献.....	106

7. 参考文献 ..... 108



## 1. 其他学校平衡步兵机器人分析综述

### 1.1 机械部分

平衡步兵作为 RoboMaster 赛事中普通机动地面单位，其机械结构仍可分为底盘和云台两大部分。综合考虑底盘研发难度和队伍状况，我们决定使用上赛季经过比赛验证非常可靠的步兵云台，放弃三轴云台等可能在研发过程中出现问题的选项，将重点放在底盘的研发上。由于论坛上开源的其他学校平衡步兵底盘机械结构较少，所以研发时我们可参考的资料较少，但综合各方面信息，RoboMaster 赛事中采用或将会使用的平衡步兵构型主要为普通的双轮构型。不同学校的研发重点则在于是否具有动量轮或双轮结构的悬挂系统，例如哈尔滨理工大学（荣成校区）开源的具备弹跳越障功能的双轮平衡步兵，以及一些学校研发的具有动量轮结构的双轮平衡步兵，具体分析如下：

- ◆ 双轮结构平衡步兵（哈工深、哈理工荣成等，以哈工程为例[1]）

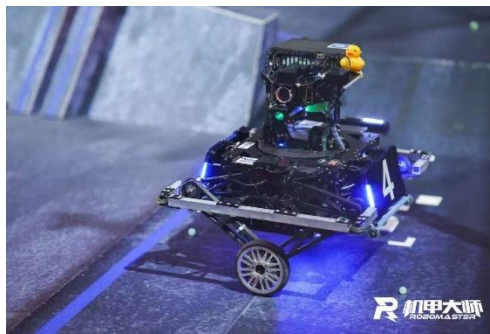


图 1.1：哈工程平衡步兵

**优势：**具有弹跳功能的主动悬挂系统，既可以减少在起伏地面的颠簸，也减少了额外带来的功率损耗。同时主动悬挂还具有弹跳越障的功能。轮腿延展的同时提高云台高度，使得越过地形攻击敌方单位成为可能，丰富战术体系。另外，轮腿的高灵活性使得转弯补偿，高度自适应等功能得以实现，最终提高整车的灵活性和机动性

**劣势：**控制算法复杂，不能全向移动。轮腿电机性能要求高，售价贵，整体成本相对较高

- ◆ 可自动调节重心的平衡底盘（哈工大，港科大等，以合工宣为例[2]）

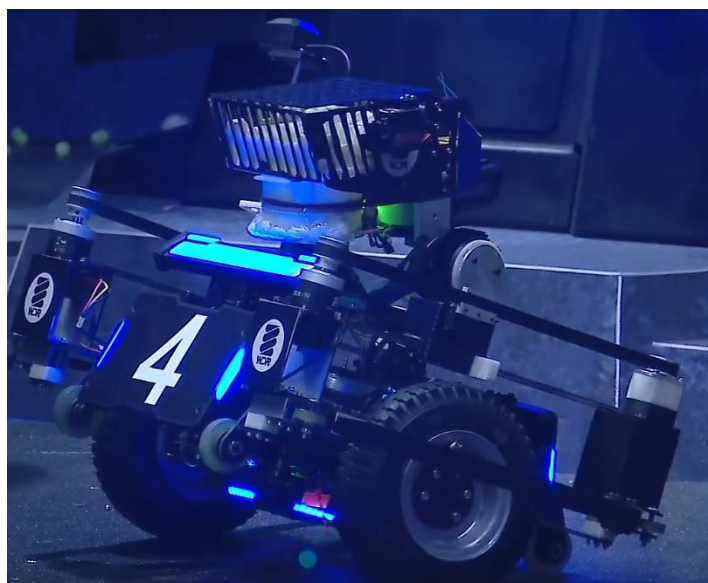


图 1.2：合工宣平衡步兵

**优势：**能够通过自动改变重心来保持平衡，能够减小陀螺仪云台的 yaw 轴晃动，同时也提高了制动和加速时的灵敏度，实现了急停和急走的功能

**劣势：**不能全向移动，底盘结构较复杂，配重增加了整车重量，影响了整车的机动性

- ◆ 普通两轮平衡底盘（北科，中南大等，以北信科为例[3]）

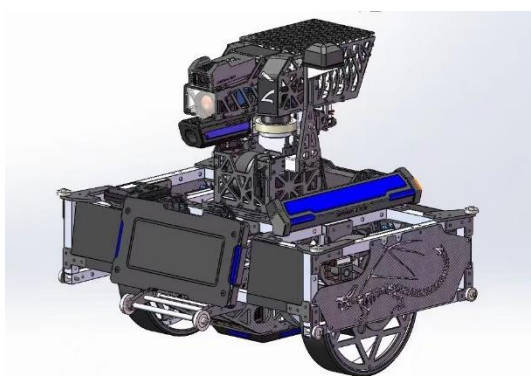


图 1.3：北信科平衡步兵

**优势：**结构简单可靠，算法简单，可以飞坡。

**劣势：**不能全向移动，操控较为困难。

综上，我们在经过技术论证之后，发现无论决定采用哪种方案，都无法实现车身垂直于前进方向的水平移动，因此我们决定将研发重点放在全向平衡底盘，即四麦克纳姆轮共轴结构的平衡底盘。

## 1.2 视觉部分

在其他战队的众多优秀开源项目中，我们主要参考了上海交通大学的视觉系统方案[4]。他们开源了完整的哨兵自瞄系统，其中 3 对于神经网络及 EKF 的尝试为我们提供了新的思路，同时优秀的工程代码风格及文件管理同样值得参考学习。可以发现，视觉系统的识别部分主要可以分为两个方向：1. 以 OpenCV 为主的传统视觉 2. 神经网络。

其中，传统视觉作为本战队上赛季所采用的方案，受限于鲁棒性和帧率的不稳定，没能发挥出预期的效果。因此，本赛季选择更换研究方向，选择神经网络作为识别模型，可以得到更加稳定、准确的装甲板识别。视觉系统的预测部分，卡尔曼滤波是大多数学校采用的方案。该赛季，本战队将该部分从下位机移植到上位机，并尝试将 KF 更换为 EKF，进行深度对比，选择更优秀的方案。

## 1.3 超级电容部分

传统的超级电容模块的充电效率和功率确实可圈可点，但是目前存在的大部分超级电容电路更多的被应用于日常的电动车等设备当中并未完全适配 robomaster 比赛需求。以较为主流的大连理工大学的超级电容方案[5]为例，其问题主要有如下几点

其一：电容和底盘并联无法改变连接，电容放电过低时电调无法启动，非定常的电压也无法让电调最有效地工作。而电容也有 10v 左右的电压无法取出。

其二：充电防倒灌为 ss54 肖特基二极管，压降较大能量损耗较大。

其三：假设机器人处于跑路状态而超级电容又处于低电压，则势必会同时向底盘和电容同时供电，无法将所有能量用于机器人底盘。

在和其他学校交流与研读其他学校同学的开源后，我们决定集百家之长，开发新一代的电容。

## 1. 机器人功能定义

所属类别	功能需求	量化标准
底盘	实现平地正常全向移动	<ol style="list-style-type: none"> <li>能够以自身位置为原点，向任意方向移动且停止时的刹车距离低于普通四麦轮步兵的 1.5 倍</li> <li>全速移动时，底盘功率平均值达到当前功率上限的 95%，方差为 5%，且在未启用超级电容的情况下全程不会因底盘功率超限而掉血</li> </ol>
	实现小陀螺功能	<ol style="list-style-type: none"> <li>原地进行小陀螺时自身中心位置不发生明显移动</li> <li>底盘抖动程度与普通四麦轮步兵相同</li> <li>原地小陀螺时底盘功率达到当前功率上限的 95%，方差为 5%</li> </ol>
	在起伏地段能够实现稳定移动	<ol style="list-style-type: none"> <li>通过起伏地段的过程中不会失去控制</li> <li>操作手能够正常适应的情况下通过起伏地段且需要的时间小于同等底盘功率上限的 1.2 倍</li> </ol>
云台	保证自身稳定	<ol style="list-style-type: none"> <li>将图传准星对准 5 米小装甲板中心，不移动准星的情况下且在平地的情况下开启小陀螺准星不离开小装甲板</li> <li>将准星对准 5 米小装甲板中心，不移动鼠标的情况下静置 5 分钟后准星仍然在装甲板范围内</li> </ol>
	低延迟操作	<ol style="list-style-type: none"> <li>连接图传、携带随机数目弹丸的情况下，操作手以不同速度移动鼠标观察云台移动情况，保证云台不失控、延迟较低且无额外抖动</li> </ol>
	实现转图传功能	<ol style="list-style-type: none"> <li>操作手能够时刻清楚自己的图传朝向（垂直于装甲板/平行于装甲板）</li> <li>连续切换 100 次，舵机不出任何故障</li> </ol>
发射机构	供弹系统稳定	<ol style="list-style-type: none"> <li>连续射击 1000 发弹丸卡弹率为 0</li> <li>设置如反转等防卡弹功能</li> </ol>

所属类别	功能需求	量化标准
	射击精准	<ol style="list-style-type: none"> <li>1. 连续击打八米外小装甲板 100 发，命中率 95%以上</li> <li>2. 配合视觉辅助，在 15 秒内激活小能量机关，在 20 秒内激活大能量机关</li> <li>3. 配合视觉辅助击打两米外的底盘功率 80w 的开启小陀螺的步兵，命中率达到 60%以上</li> <li>4. 利用激光辅助瞄准，使得激光照射到的三米距离的点与弹丸击打的三米距离的目标一致</li> <li>5. 弹速稳定达到上限弹速的 90%，方差为 5%，不能高于上限值</li> </ol>
	射击灵活调整	<ol style="list-style-type: none"> <li>1. 连续单发射击 50 发并夹杂有射击停顿、下台阶、急刹动作，过程中不发生尿弹的情况</li> <li>2. 根据当前的热量上限和每秒冷却确定射频，保证在连续射击 3-5 秒后达到上限热量的同时确保最高射频为 20 发/秒</li> <li>3. 操作手能够自主决定是否开启热量限制，开启时操作手在任何情况下均不会超热量，关闭时则由操作手自己控制。热量限制默认应当处于开启状态</li> </ol>
	实现测速模块切换功能	<ol style="list-style-type: none"> <li>1. 测速模块能够连续切换 100 次不干扰射击弹道</li> <li>2. 摩擦轮转速随当前使用的测速模块所对应的上限而改变</li> <li>3. 拥有自动切换和手动切换测速模块两个模式，自动切换模式下，当一个测速模块当前热量到达上限时自动切换为另一个测速模块；在手动切换模式下，是否切换测速模块由操作手决定</li> </ol>

所属类别	功能需求	量化标准
其他功能	鲁棒性高，易于维护	<ol style="list-style-type: none"> <li>1. 每日进行一小时的实战对抗测试，期间每当电池电量耗尽后进行一次检修，每次修复的内容三分钟内解决</li> <li>2. 刻意产生类似于视觉系统掉线等不会在规则上产生严重惩罚的故障，测试其在故障时的正常运作及问题显示</li> </ol>
	设计美观	<ol style="list-style-type: none"> <li>1. 以黑白为主色调，提供与其他机器人整体风格相同的图案，图案应当在符合要求的情况下足够美观且能够在一定程度上干扰敌方操作手</li> </ol>
	超级电容	<ol style="list-style-type: none"> <li>1. 拥有自动开启和手动开启两个模式，前者在上坡时自动开启，后者则由操作手决定开启与关闭时机</li> <li>2. 开启时能够稳定使机器人最高超出原底盘功率上限 100w 功率保持 20 秒，电能耗尽时自动关闭</li> <li>3. 拥有安全保障设计，保证其不会因正常对抗而产生燃烧等严重后果</li> </ol>
	画面 UI	<ol style="list-style-type: none"> <li>1. UI 能够正常显示且根据需求刷新内容</li> <li>2. 显示车辆底盘当前相对云台的位置、是否转图传、射击辅助线、当前使用的测速模块、摩擦轮开闭、云台 pitch 轴角度、是否产生故障、是否开启小陀螺、发射机构是否解锁、是否处于无敌时间等信息</li> </ol>

表 2.1: 功能定义

## 2. 机器人核心参数

### 3.1 机械参数

平衡步兵的重量、尺寸等，其中尺寸和重心高度可参考下表 3.1 及图 3.1-3.2

名称	参数
重量 (kg)	21
长、宽、高 (mm)	595*440*495

名称	参数
重心高度 (mm)	180
云台与车体自由度	云台 360° 自由度为 2
移动速度 (m/s)	前进: 3.9
	平移: 3.2
最大上坡度数 (°)	纵向: 15
	横向: 14
最大射频 (发/s)	25

表 3.1: 机械参数

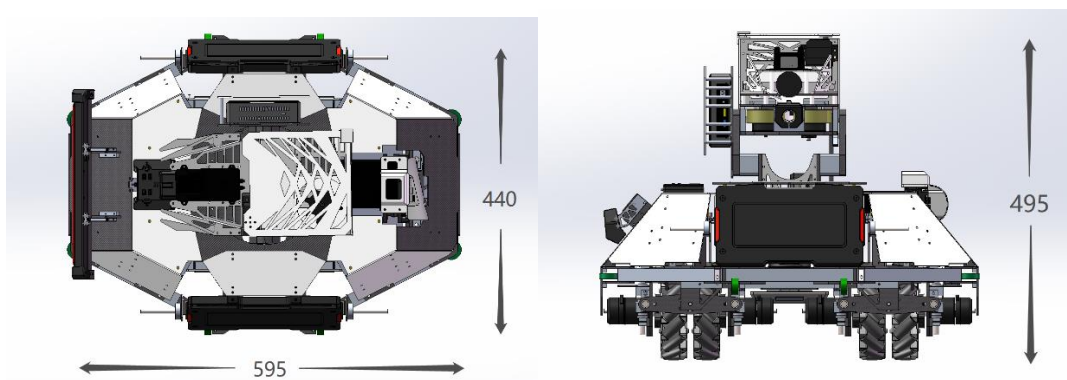


图 3.1: 机器人长宽高示意图

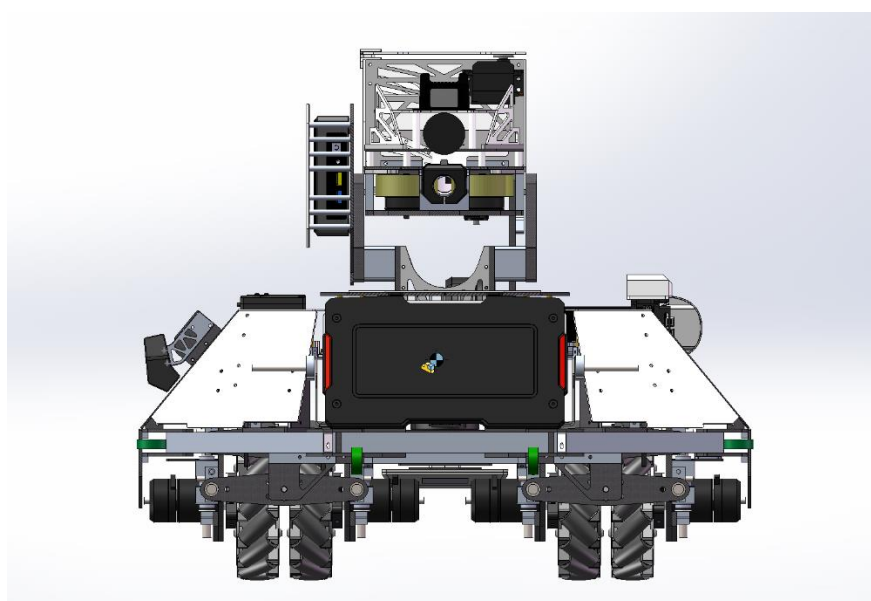


图 3.2: 机器人重心示意图

### 3.2 电路参数

整车有一个超级电容组和两块自主设计拓展板。其参数如下

- ◆ 超级电容容量：加入保护电路后参数为 24.3V/6.67F，标称容量为 1969.284J。工作电压范围为 5 ~27.5V。
- ◆ 超级电容检测模块：运用 ADC 采样电路实时监测超级电容输入输出的电流与电压情况，同时具备 CAN 收发器，实时与机器人的电路系统进行通信。
- ◆ 超级电容充放电模块：用于给超级电容充电与放电，充电部分设计最大功率为 120 W。放电部分将超级电容升压至 26 V 输出，设计最大功率为 260 W。
- ◆ 超级电容保护模块：用于检测模块和功率（充放电）模块之间的隔离与保护，工作电压 3.3v。
- ◆ 底盘分线板：用于合理分配底盘的供电以及信号的传输，额定工作电压 24v（平均值）最大工作电流 5.1A（2.5mm 线宽，50um 铜厚）。
- ◆ C 板拓展板：在兼具信号和供电的拓展以外具有 DAP 功能便于机器人代码的烧录与调试，额定工作电压 24v（平均值）最大工作电流 5.1A（2.0mm 线宽，70um 铜厚）。[6]
- ◆ NUC 供电电源：用于给 NUC 供电，最大功率为 200 W。

### 3.3 传感器参数

不考虑电机自带编码器，传感器种类与数量如下表 3.2：

名称	型号	数量
工业相机	大恒工业相机 MER2-135- 208U3C	1
工业相机镜头	OPT-C0620-2M	1
Mini PC	Intel NUC – I5- 1135G7	1

表 3.2：传感器参数



### 3. 设计方案

#### 4.1 机械结构设计

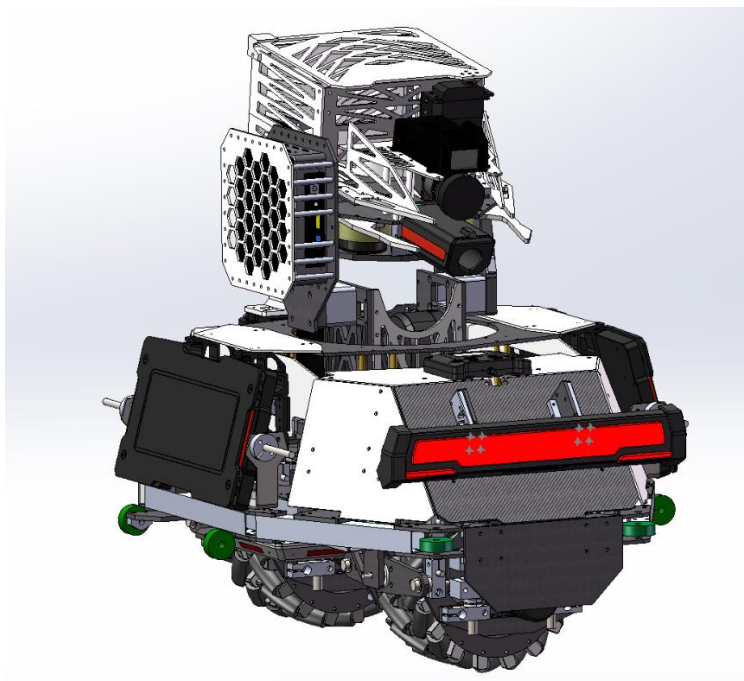


图 4.1: 平衡步兵完整机械结构

##### 4.1.1 底盘部分

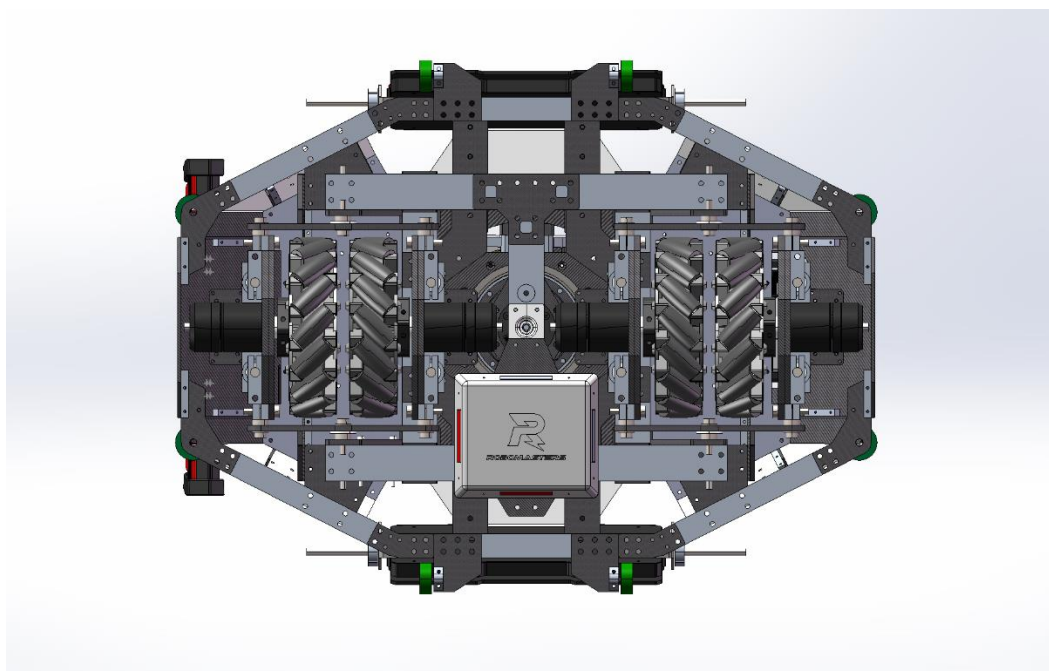


图 4.2: 底盘图

#### 4.1.1.1 共轴四麦克轮底盘整体方案分析

##### ◆ 需求分析

功能	需求
移动	能在保持平衡的状态下进行全向移动
重心	应处于中心平面附近，以减小陀螺时 roll 轴晃动

表 4.1: 底盘需求分析

##### ◆ 底盘整体设计

机器人底盘使用四麦克纳姆轮共轴结构，每个麦克纳姆轮由单独的 3508 电机进行驱动。轮组形式为电机-联轴器-麦轮。

根据测量，单个轮组宽度约为 130mm。因此四麦轮并排排布所需宽度约为 520mm，小于步兵最大宽度 600mm。因此理论上可以使用普通官方麦轮进行设计。其中约 80mm 宽度可用于轮组间隙以及两侧保护装置。

由于轮组占用了底盘的大部分空间，底盘结构的空间分布设计是难点之一，设计思路与普通步兵机器人底盘完全不同。

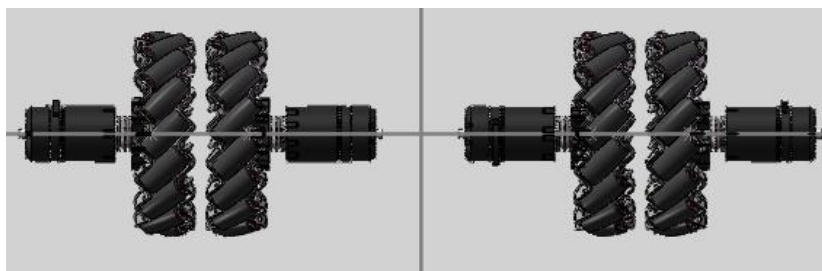


图 4.3: 轮组摆放形式示意图

##### ◆ 共轴麦轮底盘运动学分析&麦轮顺序

四麦轮共轴平衡车麦轮排序主要有两种，分别是“ABAB”以及“ABBA”。

对横移运动过程中的麦轮进行受力分析发现：“ABAB”型麦轮会出现旋转力矩，导致横移时底盘发生旋转。而“ABBA”型排序则会将麦轮的垂直分力抵消。

因此，四麦克纳姆轮共轴底盘应当使用“ABBA”型麦轮排序。

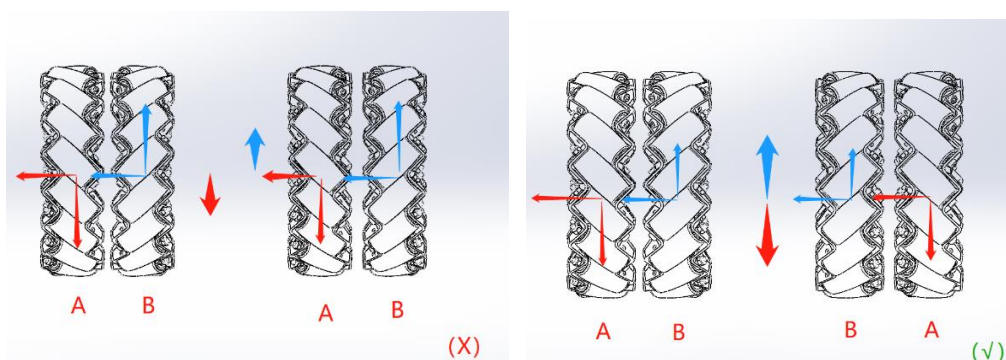


图 4.4: 麦轮顺序示意图

#### 4.1.1.2 轮组&悬挂设计

##### ◆ 需求分析

功能	需求
悬挂	能够自适应各种地形，如坡面，起伏，地台
轮组	有较强的刚性，能适应比赛强度 保护电机转子不受碰撞
加工	尽量避免使用 CNC 加工件以降低成本

表 4.2: 轮组 & 悬挂需求分析

##### ◆ 悬挂设计

由于麦克纳姆轮的工作性质：四个麦克纳姆轮同时着地且力度相同时底盘才能正常运动。由于 2022 赛季场地上存在斜坡，起伏路面等复杂地形。因此需要设计一种基于共轴麦轮结构的悬挂来适应场地要求。

独立悬挂会导致四个麦轮的着地力度不同，会影响平衡机器人的调试与控制。因此，设计了一种共轴麦轮的自适应悬挂结构。原理为通过平行双曲柄结构连接同侧的两个麦轮，从而实现麦轮的 z 轴自由度运动。

单侧悬挂结构如下图：

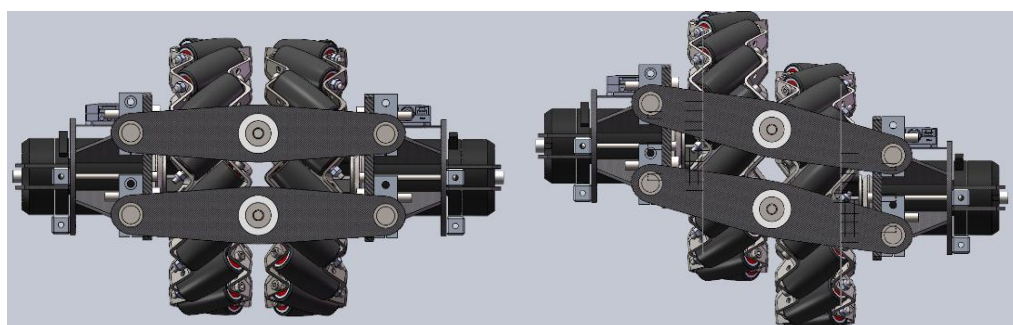


图 4.5: 悬挂结构示意图

这种悬挂结构通过关联悬挂状态与机器人重心实现自适应。由于机器人重心位于车架正中位置，而车架与悬挂位于连杆中间，因此四个麦轮受力大小始终保持相同。这为之后的电控调试降低了难度。

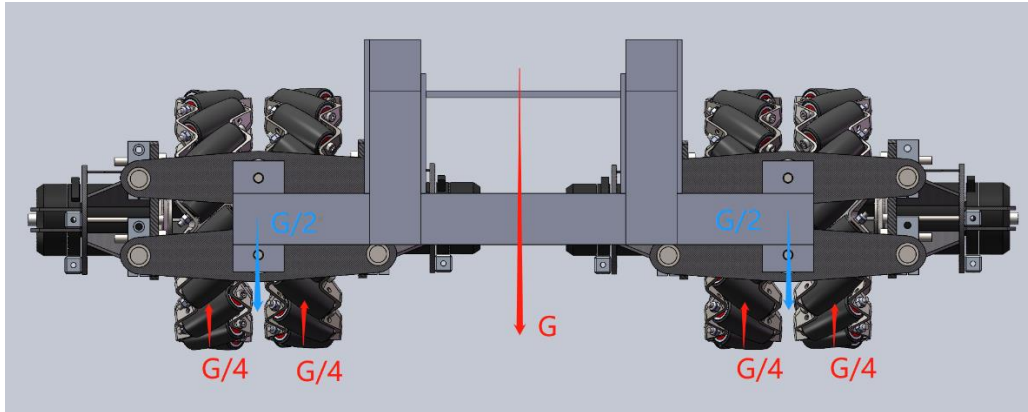


图 4.6: 四个麦轮的受力分析

机器人悬挂能通过重力自适应地形，让机器人的四个轮子在不平坦路面也能保持全部着地的状态。

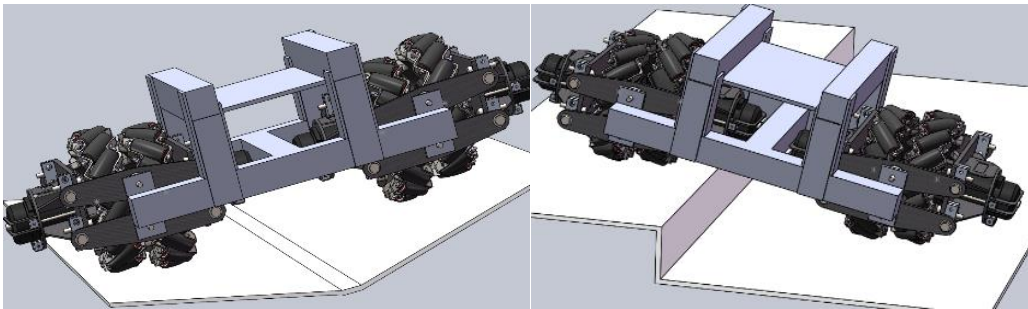


图 4.7: 不平整路面情况下的悬挂示意图

- ◆ 轮组设计（高校联盟赛无减震版本）

- 轮组支撑结构

根据对验证车进行的实验发现，由于使用的官方标准麦轮，且为单侧支撑，车身重力会施加给电机安装板力臂，导致电机安装板刚性不足而发生形变，麦克纳姆轮出现倾斜（外八）。这可能导致电控调试难度增加以及功率损失。另一种方案是双侧支撑麦轮，但是会导致单个轮组宽度增加，因此不适用。



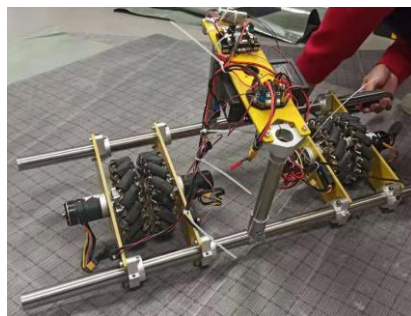


图 4.8: 平衡底盘验证车

根据分析，如果轮组的电机安装板增加横竖两个方向的支撑能减小形变。分别增加横竖两种支撑会导致结构复杂并增大体积，于是通过在轮组的电机方向增加 X 形支撑结构来增加电机安装板的刚性。

对两种结构进行有限元分析，结论为增加支撑结构后结构形变量与应力明显减小。

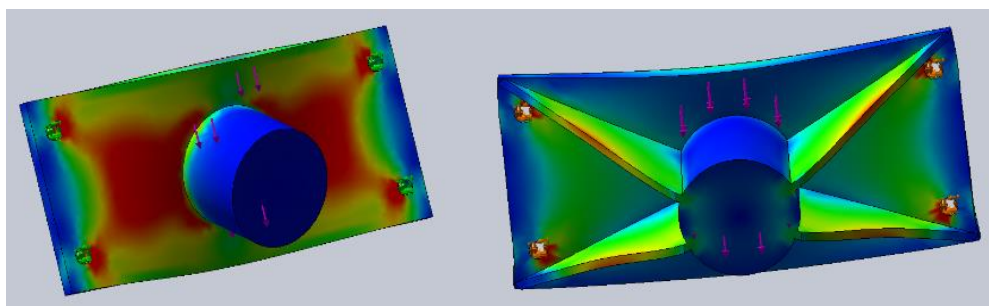


图 4.9: 电机安装板支撑结构的有限元分析对比

实际设计中通过榫卯与锁紧结构将 3508 电机减速箱与 X 形支撑固定。

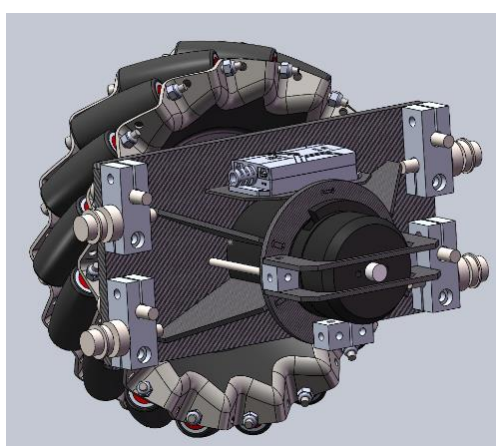


图 4.10: 单个轮组的 X 形支撑结构

#### ➤ 轮组转轴

轮组转轴使用标准件“十字轴座”代替 CNC 加工件，降低成本。

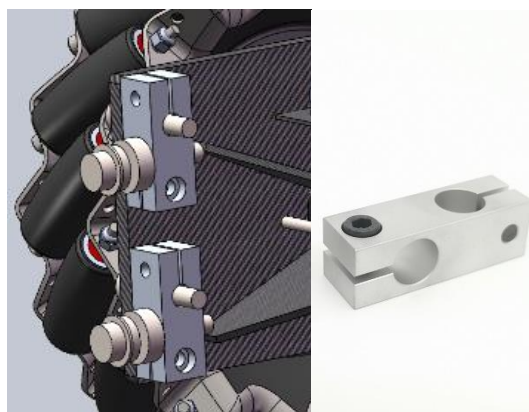


图 4.11: 十字轴座

### ➤ 电机转子保护

为了防止赛场上激烈的撞击导致外框形变，损坏电机。在 3508 电机转子外部增加防撞框。

外侧电机的防撞框为碳板材质半包裹，主要为了预防撞击。

内侧为 3d 件全包裹，防止电机转子卷入导电滑环附近的电线。

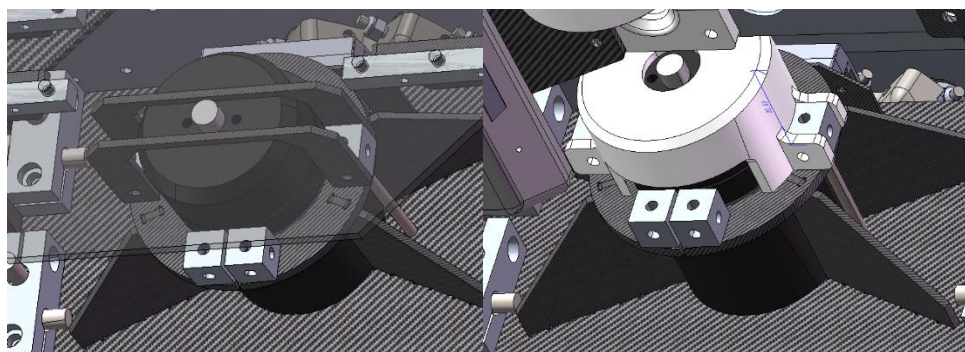


图 4.12: 外/内侧电机转子的保护

### ◆ 轮组减震装置（超级对抗赛版本）

22 赛季场地中起伏路面占据很大一部分面积。因此需要设计减震或者吸能装置来防止车架结构被起伏导致的冲击力损坏。由于底盘的整体结构与布局已经成型，能够改动的空间有限，很难再改动整个悬挂的机械结构。

解决方案为：保持整体悬挂结构不变的情况下，在单个麦轮轮组上增加垂直的减震装置，使单个麦轮轮组能发挥减震和吸能的作用。悬挂系统中的连杆和轮组分别负责自适应和减震的功能

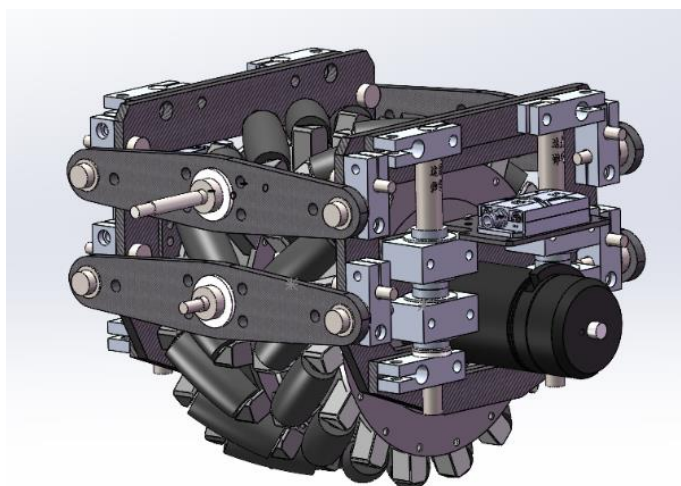


图 4.13: 减震型麦轮悬挂系统

单个减震型麦轮轮组由内外轮组两部分组成，内轮组用于固定麦克纳姆轮和电机，外轮组用于连接悬挂连杆。内外轮组通过直线轴承和钢轴进行连接，钢轴外部套入弹簧（弹簧参数为 1.3-13-40）。钢轴-弹簧-直线轴承在实现轮组上下自由度功能的同时实现了弹簧减震器的功能，这种解决方案最小的占用了底盘空间。钢轴底端加入硅胶垫片，用来缓冲轮组回弹产生的冲击力。

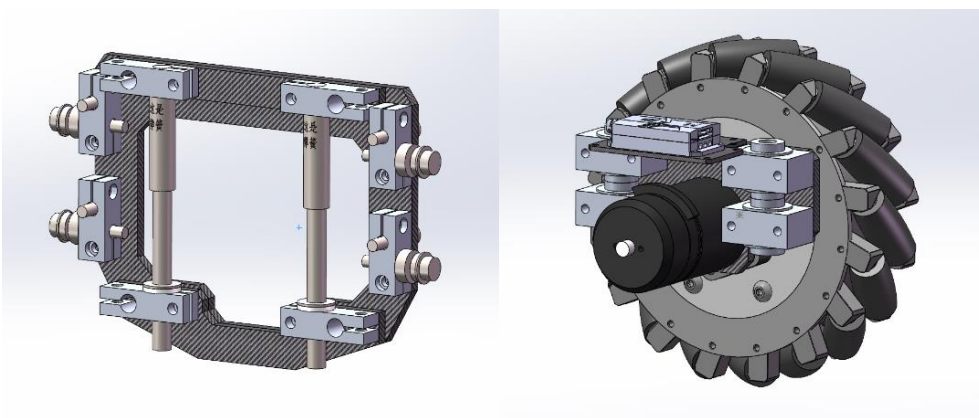


图 4.14 外/内轮组结构

减震系统设计中标准件“铝座箱式直线轴承”来实现轮组的垂直移动。由于轮组受力不垂直于钢轴，因此应当使用两个直线轴承串联来减小滑动阻力。



图 4.15: 铝座箱式直线轴承

#### 4.1.1.3 车架设计

- ◆ 需求分析

功能	需求
车架	有足够的刚性与强度,
外壳	包裹严实, 没有线材裸露
车架整体	结构紧凑, 同时有足够空间布置硬件以及走线
加工	尽量使用低成本外包加工件, 如铝方管, 板材 避免焊接

表 4.3: 车架需求分析

- ◆ 车架整体设计方案

车架整体通过铝方管与板材搭建, 无加工件。使用螺丝以及铆钉进行连接, 无需焊接。车架易损部位的铝方管内置填充件增加强度。

车架可以分为内框架以及外框架两部分。内框架用于连接轮组和云台, 结构受力大, 有铝方管内填充内嵌件进行加固。外框架用于安装外壳, 保护框, 裁判以及硬件系统, 主要受外部冲击力。



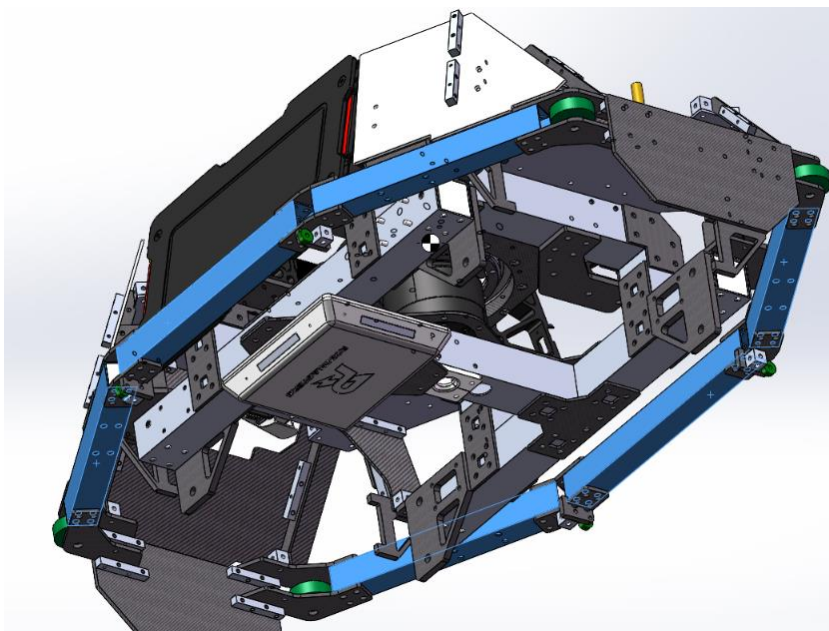


图 4.16: 车架

◆ 车架空间布局设计

相比普通步兵底盘，平衡步兵底盘体积更小，而四麦轮共轴结构导致轮组和悬挂占用了底盘的大部分空间，这导致电池，裁判系统，硬件系统和线路的空间不足。

解决方案：通过龙骨结构，在轮组上方搭建空间，从而为裁判，硬件线路和电池提供布置空间。

底盘空间布局如下图所示：

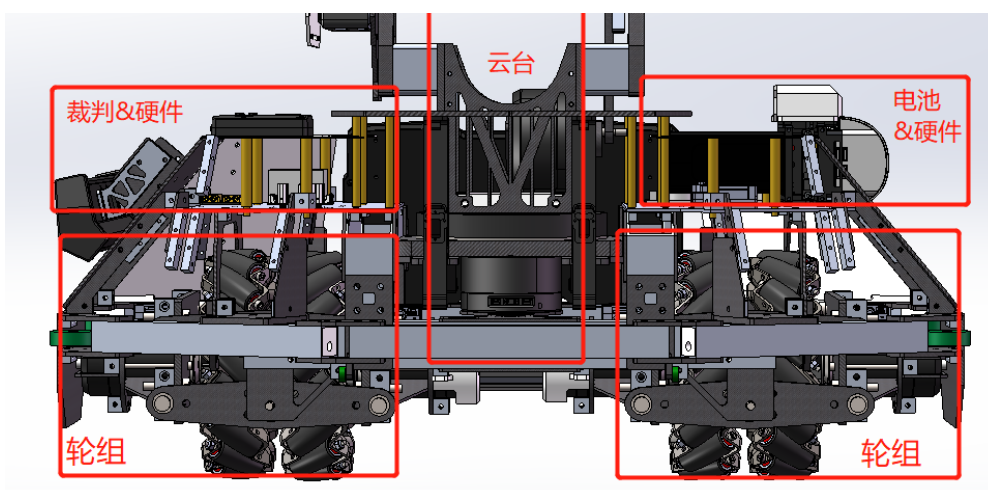


图 4.17: 底盘空间布局

- ◆ 内嵌件设计

对内框架结构进行有限元应力分析，发现横梁与竖梁的连接处的应力最集中，这意味着这个位置最可能出现损坏。需要对此处的强度进行增强。

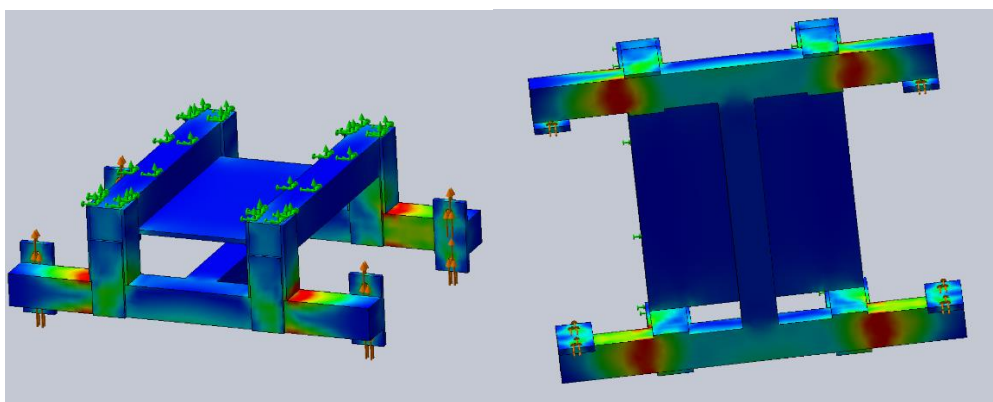


图 4.18：车架的有限元分析结果

解决方案为在应力集中部位的铝方管内，设置内嵌件填充。内嵌件采用碳板包内嵌 3d 打印件方案，利用榫卯结构，方便装配。

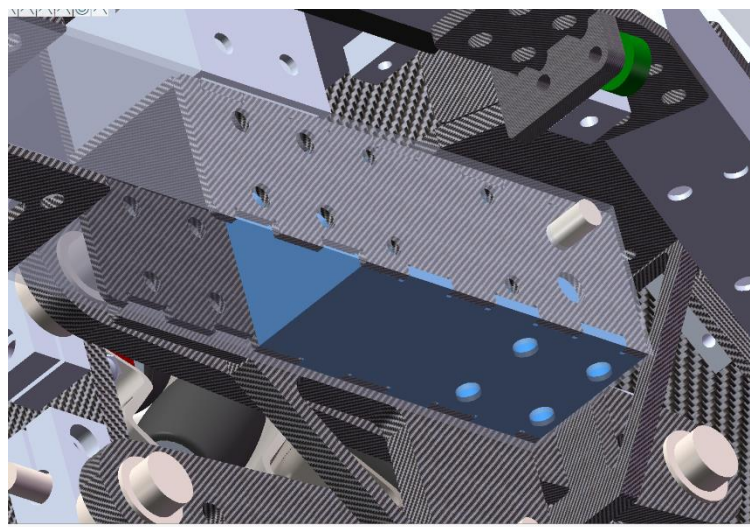


图 4.19：铝方管内嵌件

- ◆ 救援装置

救援装置使用钣金件连接钢轴作为拖拽位置，钣金件直接连接外框架。

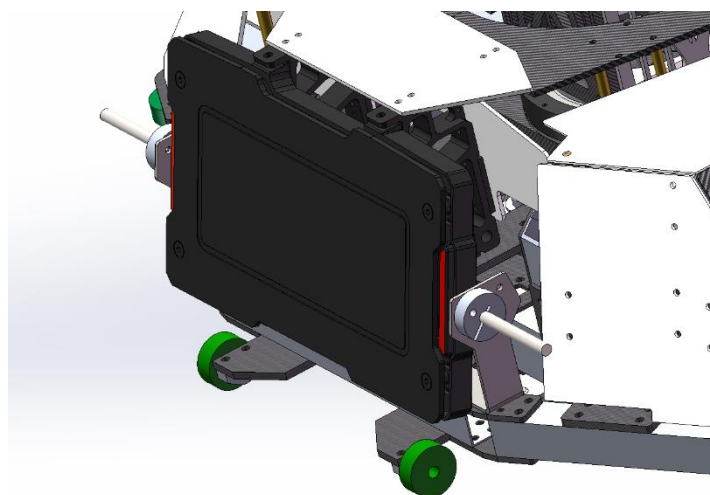


图 4.20: 救援装置

#### 4.1.1.4 底盘部分目前存在的问题&改进方向

- 由于车架的铝管多是通过铆钉进行铆接，导致较为严重的累积误差。实际组装过程中，经常出现零件无法对齐的情况，需要外力进行矫正。这导致结构存在内应力，可能增加故障发生的概率。
- 设计欠缺模块化，装配需要严格按照顺序进行，这导致维修维护难度较大。
- 减震机构行程短，飞坡，下台阶等操作可能会导致结构损坏。

#### 4.1.1.5 底盘相较于完整形态的变化

- 轮组增加减震结构
- 取消挡板设计，装甲板向外扩展移动
- 增加救援支架

#### 4.1.2 云台部分

步兵机器人云台部分采用了模块化设计理念，可根据需求选择单/双枪管设计与是否添加为采用为平衡步兵设计的图传模块旋转模块。不同步兵之间云台通用，互相兼容。本文将步兵云台归类为基本模块部分与特殊模块部分。

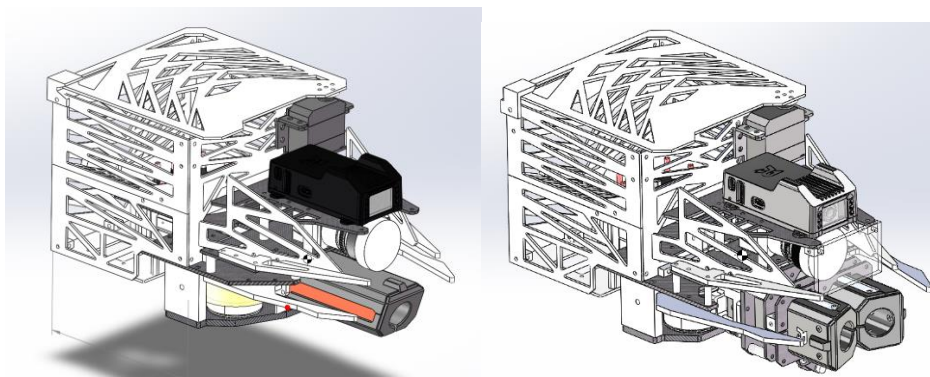


图 4.21: 单枪管云台和双枪管云台示意图

#### 4.1.2.1 发射模块

- ◆ 需求分析

功能	需求
弹仓容量	填满弹仓的情况下装弹量不少于 250 发 17mm 弹丸
射速	最高射速达到 30m/s，射速稳定
射击精度	5 米小装甲板最高射速命中率为 100%
充能装置	充能装置在 3 分钟之内完成维修，更换
整体结构	长度尽可能短，轻量化并保持较低的制作成本

表 4.4: 发射模块需求分析

- ◆ 发射机构设计方案

采用上供弹方式，弹仓位于云台后方，拨弹轮采用侧置方案，经测算，相较于传统中置拨弹轮方案可减少约 20mm 云台长度，同时使弹道更为平直，减少拨弹阻力。

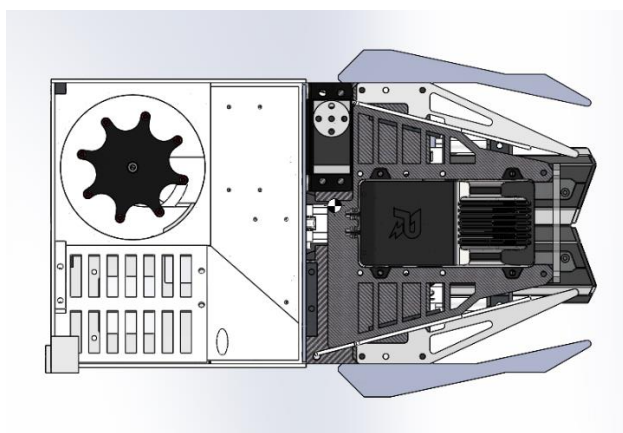


图 4.22: 发射机构俯视图

根据去年参赛经验，发现充能模块损坏率较高，因此设计快拆结构。

充能装置经由一片玻纤板固定在弹链侧边，玻纤板通过螺丝贯穿固定，两侧无遮挡，可实现快速拆装更换的需求。

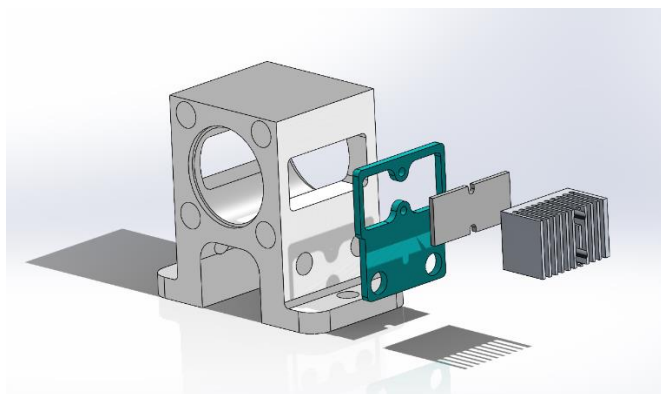


图 4.23：充能装置

定心装置采用两颗 U 型轴承，在发射过程中，弹丸先通过接触轴承并被轴承挤压完成定心，并在离开轴承之前被摩擦轮挤压发射。保证了弹丸发射前初始位置的一致性，提高了弹丸的射击精度。

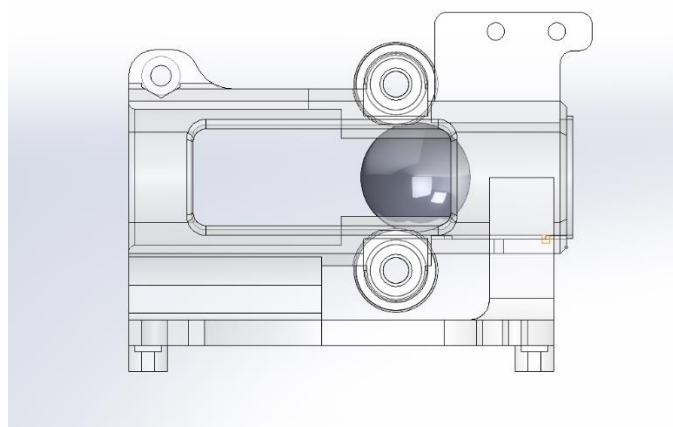


图 4.24：定心装置

#### ◆ 云台内部空间设计

侧置拨弹轮结构为云台后端另一侧预留了较大空间，以安装 RoboMaster 开发板板 C 型与云台分线板，该区域位于弹仓下方，不易因受到冲击或弹丸击打而导致损坏。



同时，云台侧板采用螺丝固定，可在三十秒内完成拆卸，拆卸后 RoboMaster 开发板 C 型的 IO 接口与云台分线板直接裸露，便于程序调试，线路维护等。

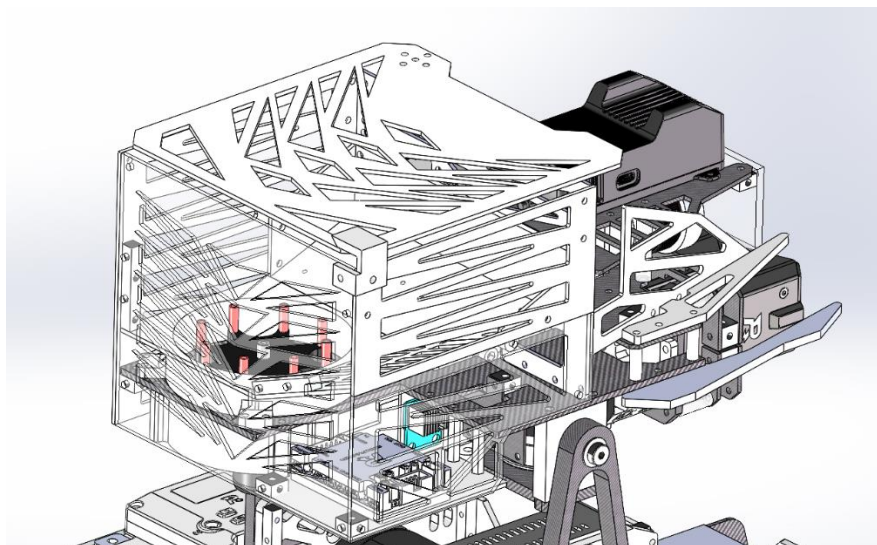


图 4.25：云台内部空间分布

#### 4.1.2.2 双枪管切换模块

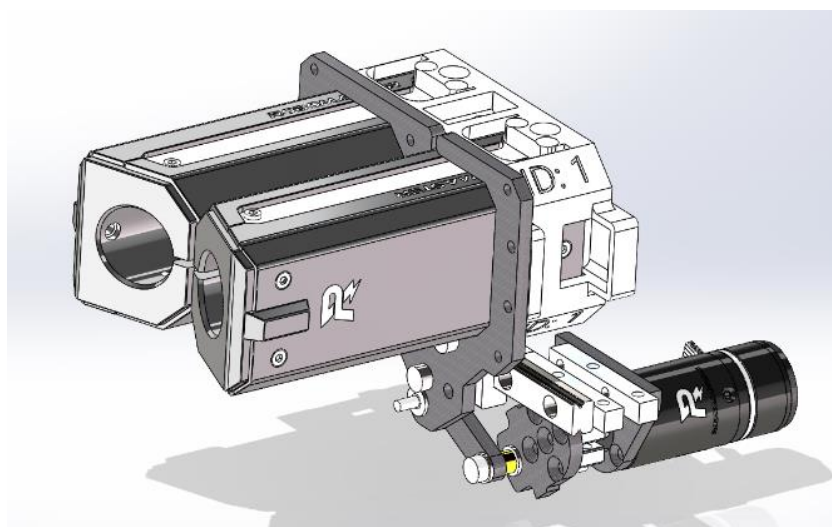


图 4.26：双枪管切换模块

#### ◆ 需求分析

功能	需求
切换测速模块	切换速度快，误差小，电控易于控制
结构	结构体积尽可能小，重量尽可能轻 能够适配普通云台

表 4.5：切换模块需求分析

◆ 结构设计

两个测速通过锁紧结构进行固定，并通过滑轨滑块连接云台。

利用连杆结构自锁的特性，防止枪口晃动。

利用连杆结构的精确特性，能减小切枪前后的位置误差。

连杆由 2006 电机驱动，占用空间小，重量轻，切换速度快。

短连杆上设计有机械限位，方便电控控制与校准。

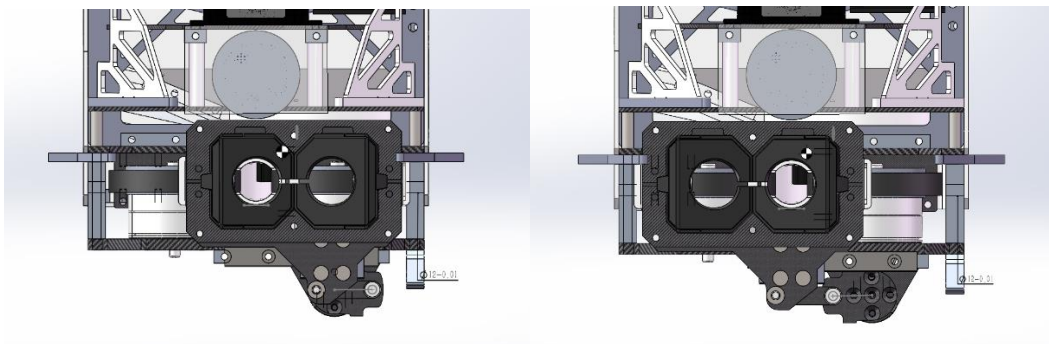


图 4.27：测速切换功能示意

## 4.2 硬件设计

### 4.2.1 平衡步兵整车线路拓扑图

平衡步兵整车线路拓扑图

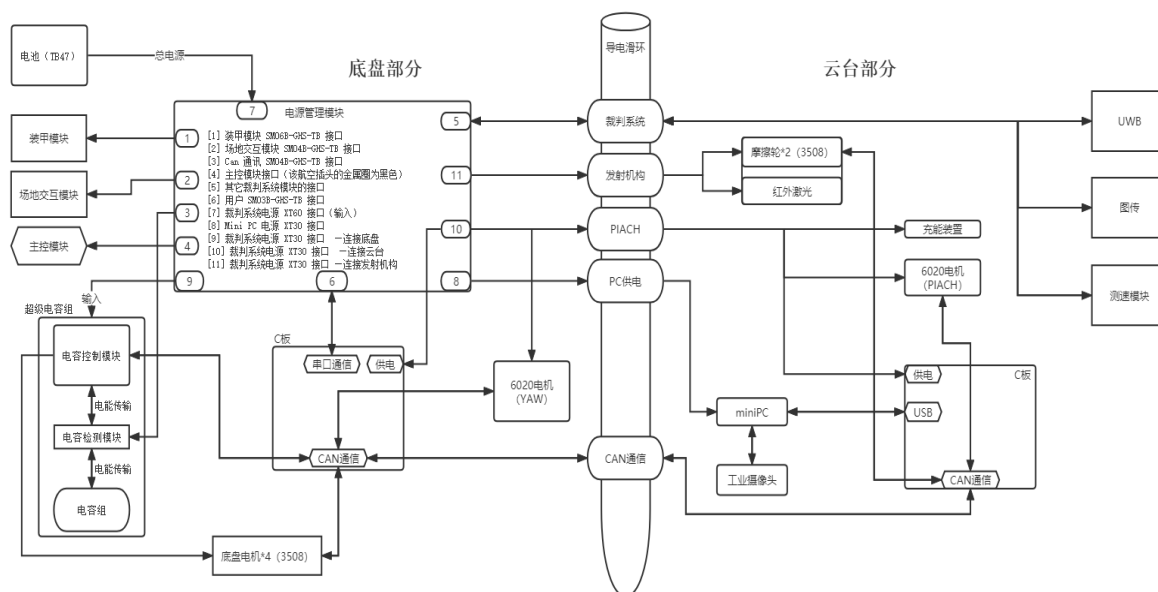


图 4.28：平衡步兵整车线路拓扑图

### 4.2.2 超级电容

#### 4.2.2.1 简介

电源管理模块的主要任务是重新分配，管理并检测机器人底盘的电源路径。因为竞赛规则允许使用超级电容器，超级电容模块的主要功能是将部分电能储存在超级电容器中，使超级电容器在需要时能向机器人底盘产生 200W 以上的功率，进而使机器人获得更好的机动性，从而高效完成比赛中的一些站位与战略部署。

#### ◆ 运行模式

为了应对赛场上各种多变复杂的场景，超级电容管理模块为中心控制板（C 板）提供了有两种优化方案和一种特殊状态：

- **充电优先：**当电容组的电量未满足时，只用 15% 的最大功率对电容器组进行充电。此时，底盘只能使用最大限制功率功率的 85%。当电容器组的电量超过 95% 时，状态机将切换到非充电模式，所有电源将提供给底盘。这个模式保证了电容组始终在充电或是保持在充满电的状态。



- 用电优先：当电容器组的电量大于 10% 时，状态机工作在超功率状态，底盘供电功率可达 260W。当电容器组的电量小于 10% 时，状态机将切换到不充电模式，裁判系统的全部功率将提供给底盘。这个模式保证了底盘始终能拿到所有功率而不会被电容充电吃掉过多功率。
- 超级电容检测：将超级电容器充满电，充满电后状态机切换到停状态式，直到中西控制板发送信息决定是否启用超级电容。

#### ◆ 设计方案

(电源//底盘)  $\Rightarrow$  DC-DC  $\Rightarrow$  电容

假设电源限制输出 100w，在底盘消耗不足 100w 时剩余功率通过 DC-DC 恒功率向超级电容充电，直到电容充满电。当点需要功率大于电源最大提供功率时，DC-DC 反向升降压提供额外功率使得电源输入功率恒定 100w。

该方案有如下优势：

- 只需要一个 DC-DC 单元，减少 PCB 面积和发热，降低成本。
- 电容基本不存在死电，利用效率取决于 DC-DC 设计水平。
- 电源一直输出，底盘电压恒为电源电压。
- 当消耗功率小于 45w 且电容不充电时效率为 100%，因为 DC-DC 电路不工作。
- 单片机多余算力可以负责底盘运动控制及 can 收发，减轻主控计算和通信压力。

[5]

#### 4.2.2.2 模块方案

我们设计的超级电容模块是一款高度集成的电路模块，是由控制模块，功率模块，保护模块以及电容组紧密连接而成。各个模块被统一安放在由铜柱和玻纤板搭建的框架上并在 H 桥等地方根据发热情况配备了硅脂和散热片。

#### ◆ 控制模块

控制模块是超级电容管理模块的计算,控制单和检测单元，用于控制其它模块的工作状态。我们控制器采用意法半导体公司的 STM32F334C8T6 芯片，该芯片采用 Cortex-M4 内核，32 位 CPU 和 FPU(72mhz max)，单周期乘法和硬件分割 DSP 指

令，具有 64KB Flash 与 16KB SRAM。内部拥有 2 个 ADC, 3 个 DAC, 3 个 comp 以及一个运放。

为了实现控制模块与整车中心控制（C 板）之间的通信，控制板上搭载了 MAX3051 CAN 信号收发器，使其通过 CAN 信号与中心控制板之间实时信息交流。

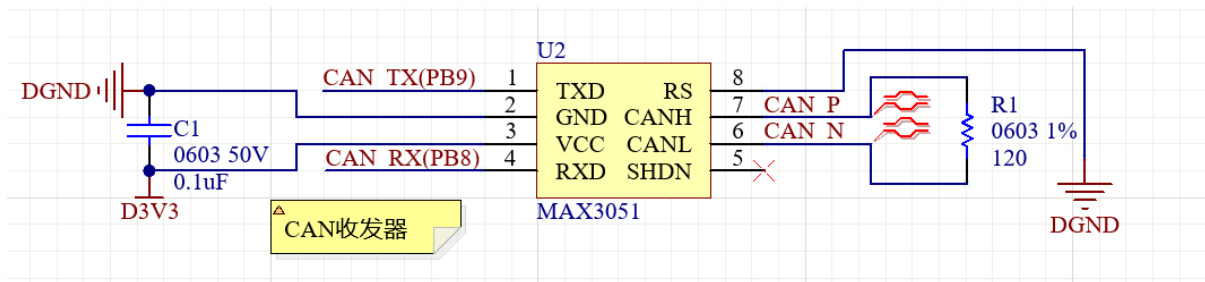


图 4.29: CAN 收发模块

为了实现充电、放电的精准功率监测，我们使用 STM32 自带的高速 16 位 ADC 配合 INA240\_A1PWR 与 OPA2350UA\2K5 外设实现对 3 路电流采样窗口、2 路电压采样窗口进行自动循环采样和数据转移。期间还需要配置 MPU 保护区域、cache 禁止缓存区域、ld 链接文件等。

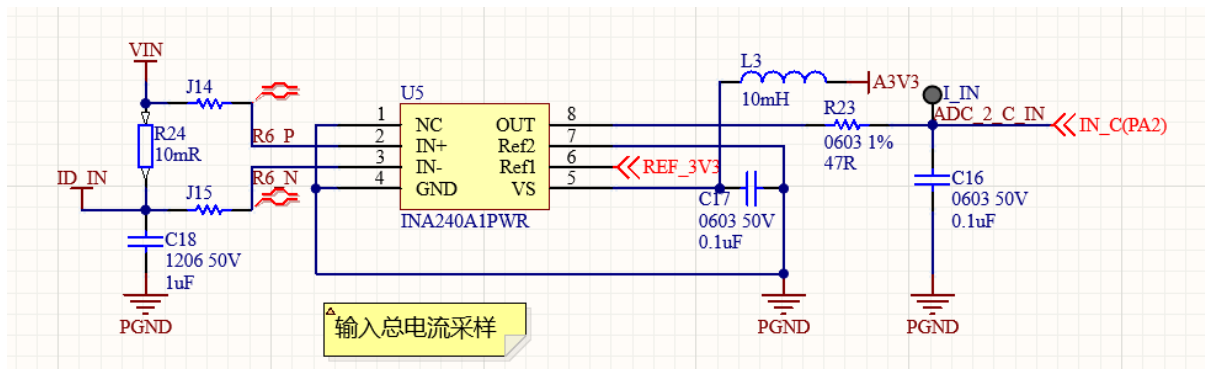


图 4.30: 电流采样

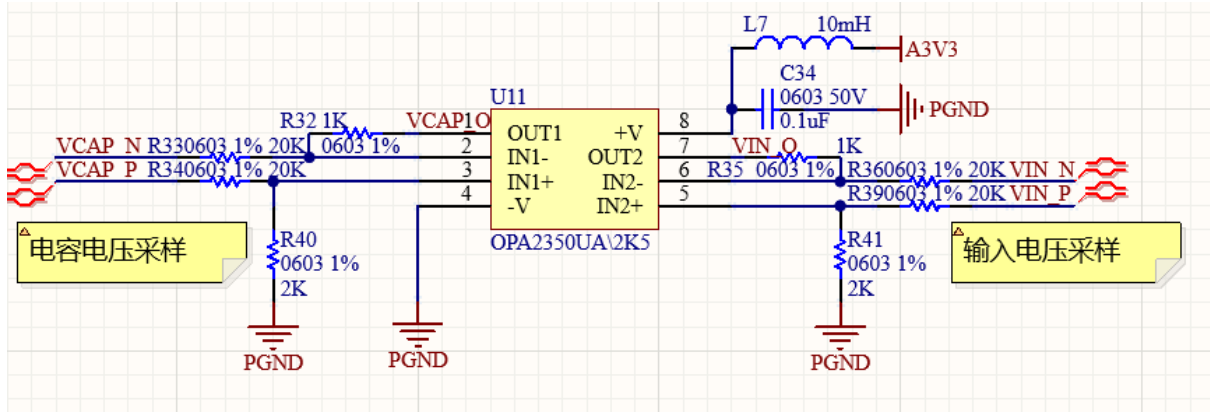


图 4.31: 电压采样

◆ 功率模块

该模块主要负责控制电容组的充放电功能，通过构建非反相降压-升压转换器（即 buck - boost 电路）完成电容两端的双向升降压，继而将电容中的所有电能全部利用而不是和传统的电容控制那般电压低于 7V 则无法放电。这里我们选用的驱动为英飞凌公司的 2EDF7275K，一款快速双通道隔离 MOSFET 门驱动芯片，通过无芯变压器(CT)技术提供功能(2EDFx)或增强(2EDSx)输入到输出隔离。由于高驱动电流，优秀的共模抑制和快速的信号传播，2EDi 特别适合于驱动在快速开关电源系统中高压 MOSFET。

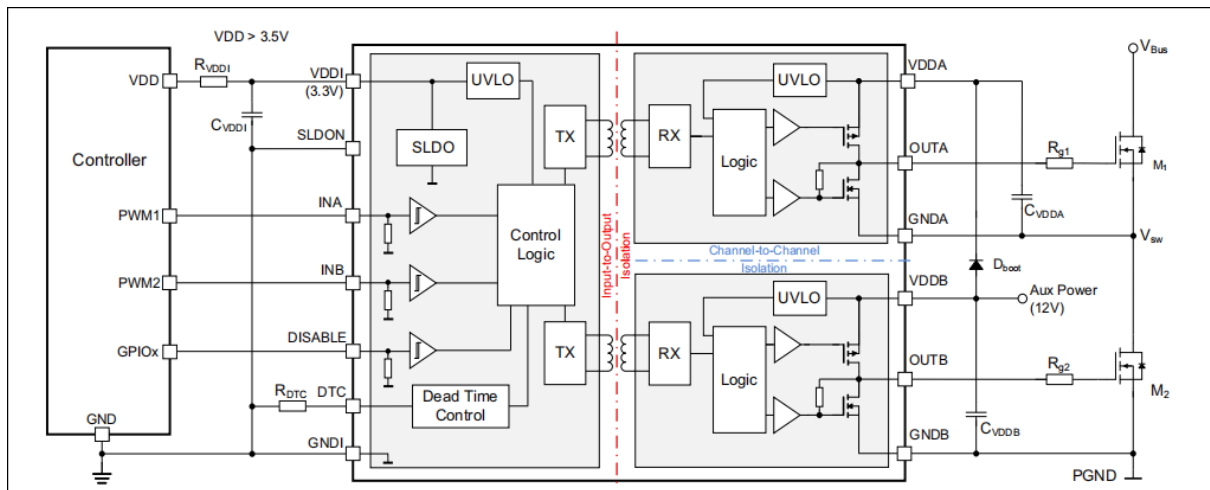


图 4.32: MD\_2EDF7275K[7]

为了满足超级电容 buck - boost 电路中大电流高频率以及超高发热量的设计需求，我们选择了同样也是英飞凌公司的 BSC070N10NS5 N 型 MOSFET 进行 H 桥的构建。该 MOSFET 有着 100V 的额定电压和 80A 的额定电流，死区时间仅在 13-24ns。其 PQFN8 的封装也极大的解决了散热问题，使得超级电容的瞬时工作最高温度在 140°C 以下（无散热器裸板测试情况下）。

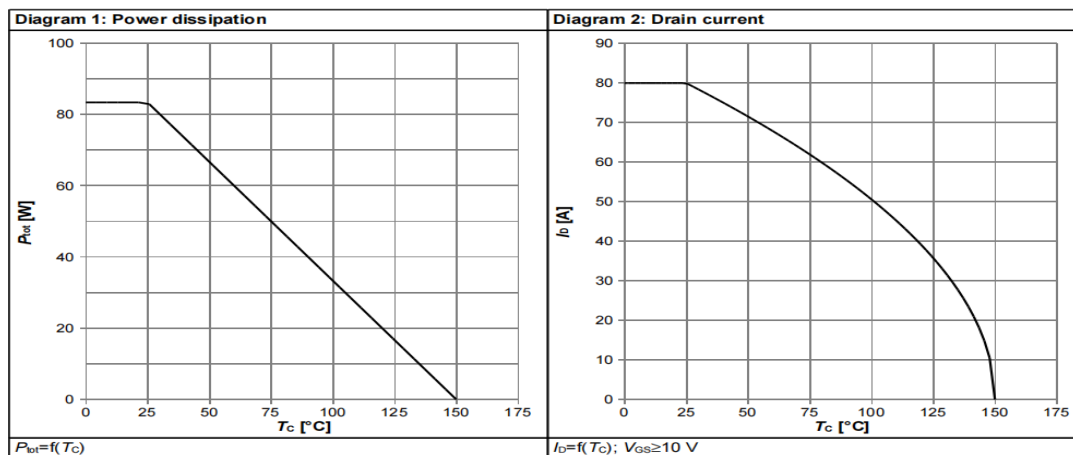


图 4.33: BSC070N10NS5 温度-电流曲线[8]

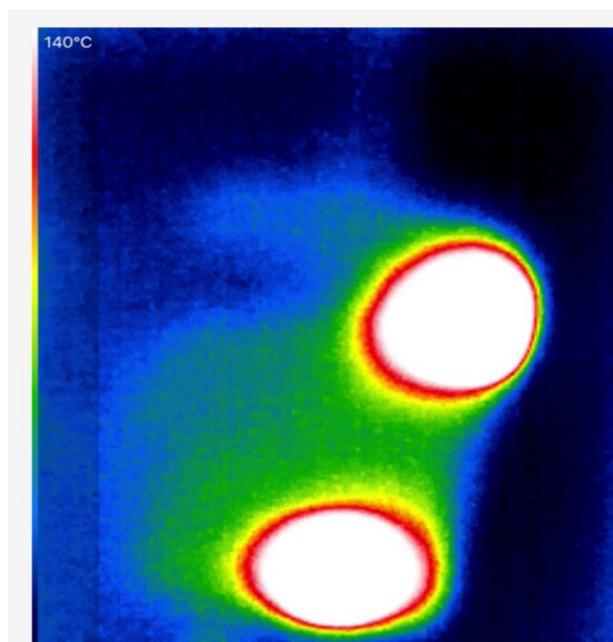


图 4.34: BSC070N10NS5 实验时实际最高温度

在充电时，该模块可受控于微控制器，通过控制 H 桥 MOS 的开关频率来控制电容的充电功率以及充电电流。本设计中需要控制的是充电电流，在超级电容不同电压

下，模块恒定输出电流与输入电流差距很大，因此需要微控制器的参与来通过调整输出电流以实现输入电流（即输入功率）的控制。

在放电时，该模块同样也可受控于微控制器，通过控制 H 桥 MOS 的开关频率来控制电容的放电功率以及放电电压。通过 boost 电路的不断升压，将输出电压一直稳定在 24V，保证输出电压的同时榨干了电容里的每一滴电能，理论上电能的使用效率将会接近 100%，而不会像传统的半桥电容设计方案那样存在部分电能储存与电容中无法使用的情况（电压小于电调的最小启动电压）。

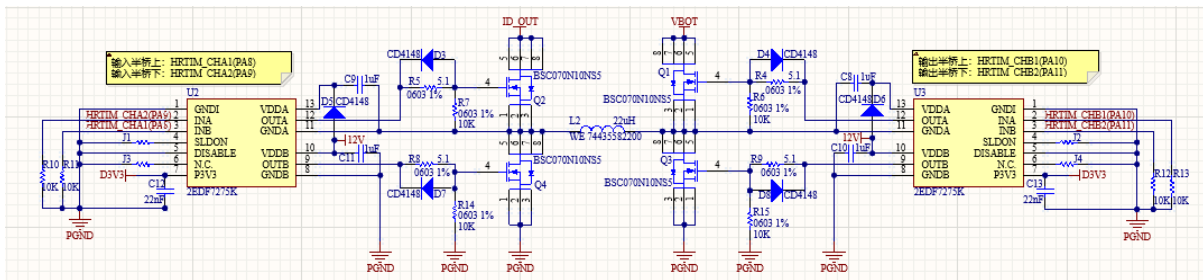


图 4.35: 非反相降压-升压转换器

- ◆ DC-DC 电源设计相关参数计算

- BUCK 电感参数计算

定义占空比为  $D = t_{on}/(t_{on} + t_{off})$ ，其中  $t_{on}$  和  $t_{off}$  为 MOS 通态和关态时间，那么在 BUCK 电路中输入与输出的关系如下：

$$V_{out}/V_{in} = t_{on}/(t_{on} + t_{off}) = D$$

在上臂 MOS 导通( $t_{on}$ )，电感充能期间，电感两端电压为  $V_{in} - V_{out}$ ，电感电流线性增加：

$$U_L = L \times di/dt$$

那么我们可以得到电感电流增加量  $\Delta I_{L1}$ ：

$$\Delta I_{L1} = t_{on} \times V_{in} - V_{out}/L$$

在上臂 MOS 关断( $t_{off}$ )，电感释放能量，电感两端电压为  $V_{out}$ ，电感电流线性减小，那么我们得到电感电流减小量  $\Delta I_{L2}$ ：

$$\Delta I_{L2} = t_{off} \times V_{out} / L$$

理想地，BUCK 电路稳定工作时，有  $\Delta I_L = \Delta I_{L1} = \Delta I_{L2}$ ，并且  $\Delta I_L = K I_{out} - max$ ，其中 K 通常取 10~20% 经验值。

那么电感值可以计算为：

$$L_{BUCK} = V_{in} - V_{out} / \Delta I_L \times D / f_{SW} = V_{in}(1 - D) D / f_{SW} K I_{out} - max$$

在整个开关周期中电感量都要满足要求，电感要取最大值，我们取 D=50% 来计算，假设以 120W 的功率向电容组充电，且电池供电电压为 25V，则电流为 4A，那么以输入电压  $V_{in}=25V$ ，输出电流  $I_{out} - max = 4.8A$ ，电源纹波  $V_{pp} - in / V_{pp} - out$  均为 200mV，开关频率  $f_{SW}=288kHz$  为例，K 取 20%，计算得：

$$L_{BUCK} = 22.6\mu H$$

### ➤ BOOST 电感参数计算

定义占空比为  $D = t_{on} / (t_{on} + t_{off})$ ，其中  $t_{on}$  和  $t_{off}$  为 MOS 通态和关态时间，那么在 BOOST 电路中输入与输出的关系如下：

$$V_{out} / V_{in} = t_{on} + t_{off} / t_{on} = 1 / (1 - D)$$

在上臂 MOS 导通 ( $t_{on}$ )，电感充能期间，电感两端电压为  $V_{in}$ ，电感电流线性增加：

$$U_L = L \times di/dt$$

那么我们可以得到电感电流增加量  $\Delta I_{L1}$ ：

$$\Delta I_{L1} = t_{on} \times V_{in} / L$$

在上臂 MOS 关断 ( $t_{off}$ )，电感释放能量，电感两端电压为  $V_{out} - V_{in}$ ，电感电流线性减小，那么我们得到电感电流减小量  $\Delta I_{L2}$ ：

$$\Delta I_{L2} = t_{off} \times V_{out} - V_{in} / L$$

同样理想地，BOOST 电路稳定工作时，有  $\Delta I_L = \Delta I_{L1} = \Delta I_{L2}$

电感值：根据对于  $\Delta I_L$  的推导，有：

$$L_{Boost} = V_{out} - V_{in} / \Delta I_L \times (1 - D) / f_{SW} = V_{out}(1 - D) D / f_{SW} K I_{out} - max$$

在整个开关周期中电感量都要满足要求，电感要取最大值，我们取  $D=50\%$  来计算，假设机器人地盘在大多数工况下，功耗在 200-300W 范围内，且电池供电电压为 25-26V，则电流范围主要集中在 8-12A 范围，那么以输出电压  $V_{out}=25V$ ，输出电流  $I_{out-max} = 10A$ ，电源纹波  $V_{pp-in}/V_{pp-out}$  均为 200mV，开关频率  $f_{SW}=288kHz$  为例，K 取 10%，计算得到

$$L_{Boost} = 21.7\mu H$$

可见电路在特定工况下是对称的，两种工况下所需的 L 值非常相近。为了满足所有工况的要求，L 应该取最大值，即  $22.6\mu H$ ，然而市面上无法购买到  $22.6\mu H$  或  $23\mu H$  的电感器，因此我们最终选择的电感值为  $22\mu H$

#### ➤ 输入输出电容参数计算

输入电容的作用主要是保障输入电压的稳定，若输入电容过小，当 BUCK 输出重载时，可能拉低输入电压；输入电容的纹波大，将污染输入电源；所以输入电容要求足够大，在电容上产生的纹波小于输入电源纹波

将滤波电容等效为一个理想电容 C 和一个等效串联电阻(ESR)，那么输入/输出纹波电压  $\Delta V_{in}/\Delta V_{out}$  计算为：

$$\Delta V_{out} = \Delta V_C + \Delta V_R = ESR \times \Delta I + \Delta I / 8Cf_{SW}$$

其中， $\Delta V_C$  为理想电容两端的纹波电压， $\Delta V_R$  为 ESR 两端的电压，为了方便计算，同时减小体积提高性能，我们采用了单体 ESR 低于  $10m\Omega$  的陶瓷电容，多个并联后 ESR 可忽略不计，则 C 计算为：

$$C = \Delta I / 8\Delta V_{out}f_{SW}$$

根据前面的参数，计算得出  $C=41.7\mu F$ ，在我们的设计中，由于 BUCK-BOOST 电路需要持续为电机供电，而电机属于功率会大范围变动的动态负载，BUCK-BOOST 电路的工况较为恶劣，因此为了保证稳定性，我们保留 150% 余量，则为  $110\mu F$ 。

#### ◆ 保护模块

超级电容组的功率模块主要进行着高频率和大电流的开关工作，在 MOS 开关以及电容状态切换的过程中不可避免的出现的反电动势以及大的浪涌。尽管所选用的 IC

驱动内部自带隔离，但为了安全性与稳定性考虑，我们制作了这一模块将控制模块与功率模块隔离。

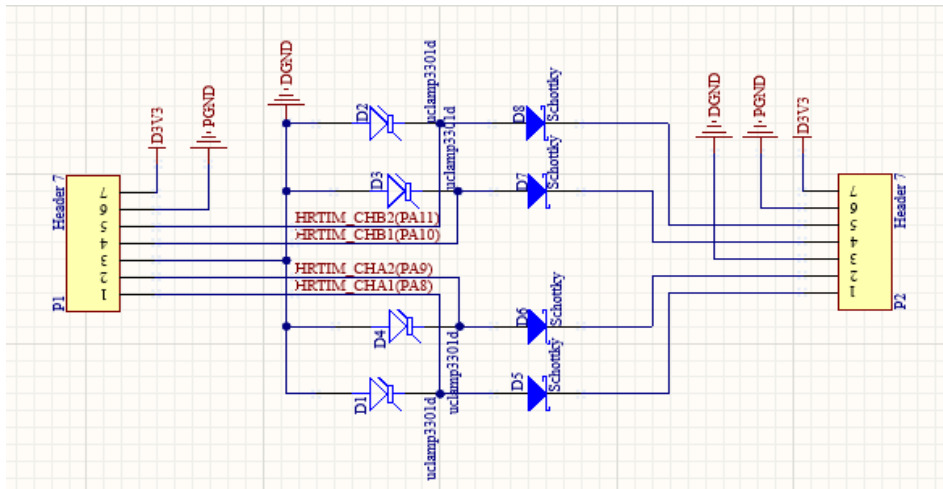


图 4.36 保护模块

模块中每一路都具备一个肖特基二极管和 TVS 二极管用于吸收反冲的电能，充分保护控制模块不受影响。

#### ◆ 电容组

电容模组使用 2.7V/60F 电容单体，组合成九串的模组，模组标称参数为 24.3V/6.67F。由于超级电容单体保护使用阈值为 2.65V 的 BW6101 保护芯片在单体出现过压时比较器输出高电平控制 N-MOS 管开启，所以加入了单体保护的超级电容模组可得到新的标称参数 23.7V/6.67F，则有下面的公式：

$$E = 1/2 * C * V^2$$

其中 E 为电容组总能量，C 为电容组总电容，V 为电容组工作电压，那么计算得到电容组在设计工况下的总能量为 1969.284J，低于规则中 2000J 的限制。

泄放回路为串联式，当单体触发泄放时，总是向下一个单体输出，直到该串中最后的一个电容向 GND 泄放。这种设计的优点是在单体充电平衡时，并不直接浪费所有能量，可以提高效率；缺点是最后一个泄放 MOS 管总优先承受最大的泄放电流，直到该串所有的电容电压相等，在使用过程中将使得热量集中于模组的一端，有需要情况下在布局时交叉摆放以减少这种影响。



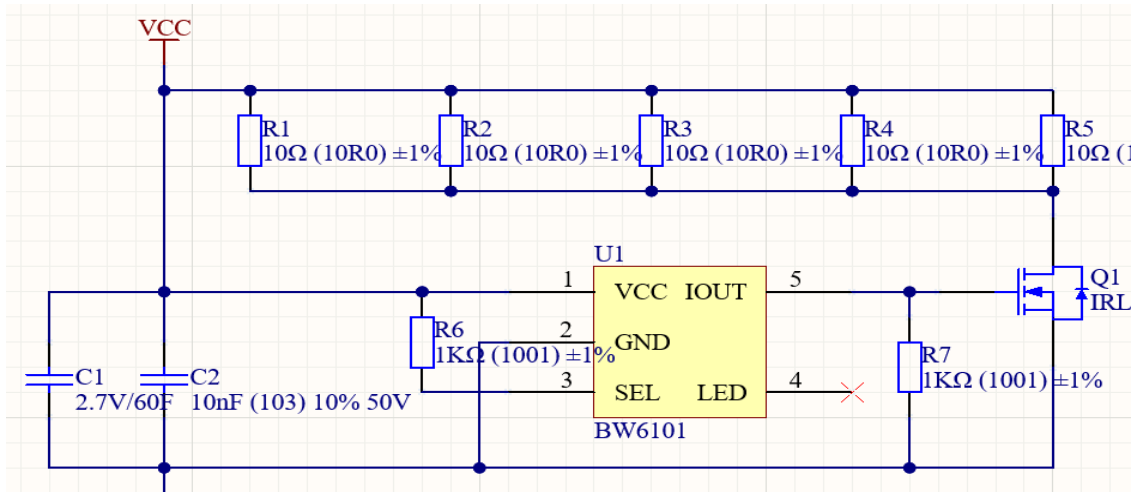


图 4.37：电容组模块单元

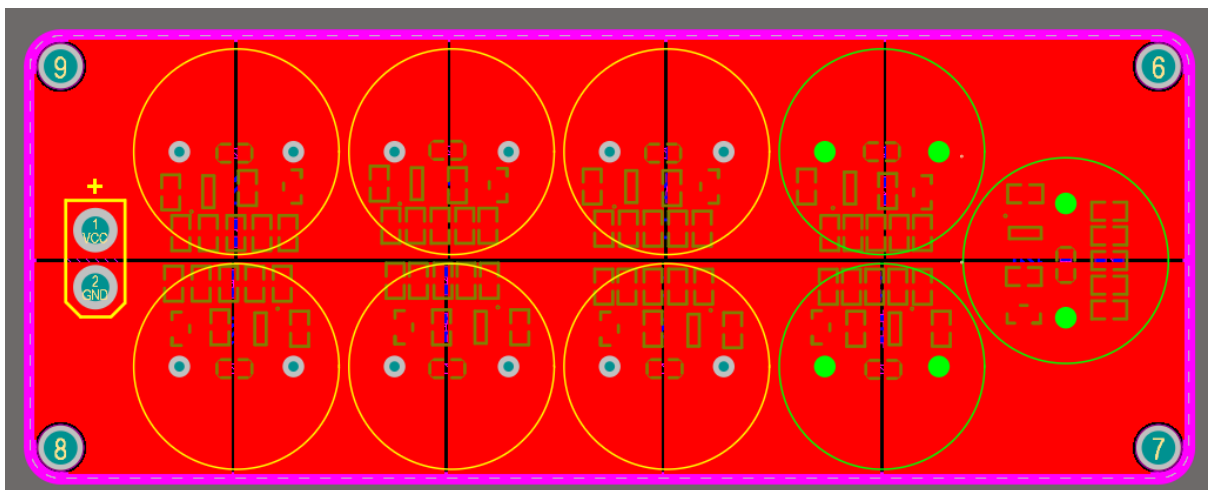


图 4.38：电容组布线

#### 4.2.2.3 主要算法

- ◆ 采样校准算法

由于 ADC 自身以及采样芯片的误差, 构建最朴素的线性误差模型

$$Y = aX + b$$

则误差参数被确定为零偏  $b$  和比例因子  $a$ , 使用两点拟合法即可计算出所有误差参数. 在电压采样校准中, 取  $X_1=20V$ ,  $X_2=24V$ , 在电流采样中, 取  $X_1=2A$ ,  $X_2=4A$ 。

- ◆ 自检算法

在系统上电时, 需要对所有采样进行自检, 避免因采样异常导致的功率失控. 其中, 过欠压检测门限  $V_{th}$  被设定为  $30V$  和  $19V$ , 电流检测异常条件为上电读数超过  $\pm 0.5A$  或

输入采样 $I_{in}$ 与输出采样 $I_{out}$ 差值超过 0.5A. 若存在以上问题, 程序将不会输出驱动信号, 并以 LED 闪烁的方式报错。

#### ◆ 功率控制闭环

整个控制闭环由双串级 PID 组成, 分别是外环功率环和内环并行计算的电压环电流环。

输入采样数据首先通过一个大小为 10 的滑动窗口滤波器, 然后根据输入电压采样 $V_{in}$ 和电流采样 $I_{in}$ 计算出输入功率 $P_{in}$ , 与设定功率 $P_{set}$ 一起进行功率环计算, 得到计算充放电功率。

若充放电功率大于 0, 则为充电模式, 由以下公式确定电流环设定值

$$I_{set} = \min\max(P_{charge}/(V_{in} - V_{out}))$$

电流环反馈值计算方法为

$$I_{ref} = I_{in} - I_{out}$$

电压环设定值为电容组充电限制电压 23V, 反馈值为电容电压采样 $V_{cap}$ . 最终输出占空比 D 为

$$D_{out} = \min(D_{cloop}, D_{vloop})$$

若充放电功率小于 0, 则为放电模式, 进行单电流环计算, 电流环设定值

$$I_{set} = \min\max(P_{charge}/(V_{in} - V_{out}))$$

反馈值

$$I_{ref} = I_{in} - I_{out}$$

#### ◆ 动态电流算法

根据 DC-DC 电路原理, 电感两边的电流满足以下条件

$$I_{in}/I_{out} = D$$

为防止电感两边压差过大导致的 mos 过流, 提出了一套动态电流算法. 当压差较小时, 充放电电流可以被设定在较高的值, 当压差很大时, 为了避免电容 mos 过流, 充放电电流应被限制. 计算公式如下

$$I_{lim} = \max((V_{cap}/V_{in}) * (I_{max} - 1A), 1A)$$

#### ◆ 故障检测

运行时故障检测可以及时发现故障并关断输出, 避免进一步损坏. 运行时故障检测分为过欠压异常, 电流过流, 充放电异常. 其中需要重点说明的是充放电异常, 其能够检测 mos 损坏造成的充放电功率异常, 并防止 mos 击穿导致的全车短路, 具体判断方法为, 若 DC-DC 设定功率和实际计算功率长时间相差过大, 则认为发生充放电异常.

#### ◆ 代码加速

对于数字电源这类高实时性场景, 代码执行速度是优化的重点. 放弃了操作系统转而使用裸机开发, 避免了上下文切换对于代码执行效率和实时性的影响。

使用与 Cortex 内核紧耦合的 CCMRAM 执行代码, 能够在 72MHz 以 0 等待周期执行代码, 对于实时的计算密集型程序, 例如超级电容的控制环, 该器件大大减少了关键任务的执行时间. 对于放入 CCMRAM 执行的中断服务函数, 还能够保证最短的延迟时间。

为了减少 CPU 处理和搬运数据时的消耗, 对于裁判系统串口接收任务, 使用空闲中断和 DMA 以避免频繁进出中断的性能损耗。对于 ADC 采样任务, 使用连续转换和扫描模式, 配合 DMA 传输, 达到 CPU 零参与的目的。

#### ◆ 安全性

安全性也是值得关注的重点, 从软硬两方面入手, 尽量避免漏洞的产生. 软件上, 代码编写需要符合编码规范, 避免动态内存分配, 每个错误都有相应的处理程序, 具有良好的故障分级制度. 硬件上, 通过模拟看门狗监测电容电压电流数据, 一旦超出阈值即刻停止升降压电路 PWM 输出, 使用窗口看门狗监测代码执行情况, 超出预定时间即触发复位, 保证超级电容软件的安全可靠。

超级电容状态码如下:

状态码	指示灯描述	描述
NORM	白灯闪 x 下	使用 x 号配置文件
OVV	红蓝交替闪烁, 蓝灯点亮时间大于红灯	输入过压保护
LOV	红蓝交替闪烁, 红灯点亮时间大于蓝灯	输入欠压保护

状态码	指示灯描述	描述
CSE	红绿交替闪烁	电流采样异常

表 4.6: 状态码 (启动)

状态码	指示灯描述	描述
CAPF	绿灯常亮	电容冲满
CAPC	蓝灯常亮	电容充电
CAPD	红灯常亮	电容放电
CAPL	红灯闪烁	电容电量低
OVV	红蓝交替闪烁, 蓝灯点亮时间大于红灯	输入过压保护
LOV	红蓝交替闪烁, 红灯点亮时间大于蓝灯	输入欠压保护
OVC	红绿交替闪烁	过流保护
DRVE	黄色闪烁	充放电异常
CANE	青色闪烁	CAN 通信异常

表 4.7: 状态码 (运行)

#### 4.2.2.4 超级电容模块附图

下面展示了超级电容模块中各个模块的原理图, PCB 图以及实物图。超级电容模块的原型验证及设计耗时近一年, 是整车电路中最复杂的部分。我们采用集成是的设计方式, 使超级电容的体积更小 (控制板仅有银行卡 80% 的大小), 各个模块更加稳定, 在设计的过程中也锻炼了组内同学们的协作能力。是战队今年的重大突破之一。

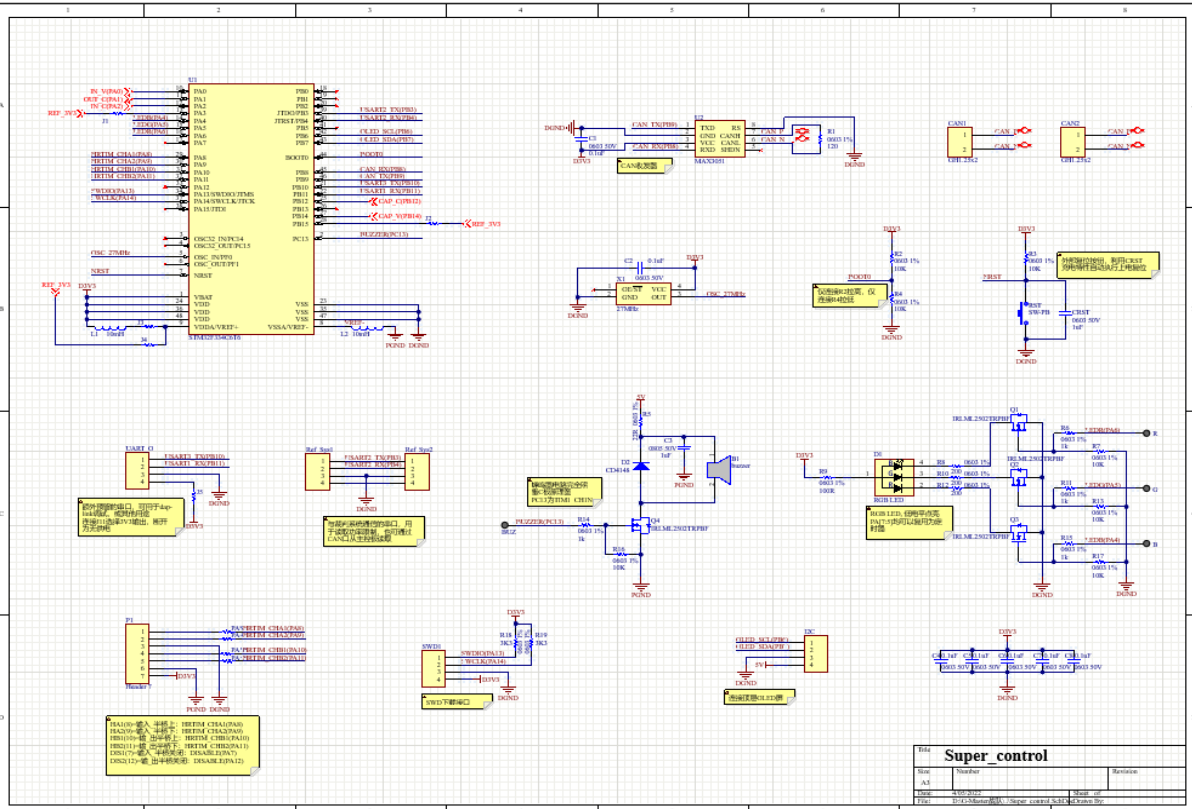


图 4.39: 控制模块-系统控制部分

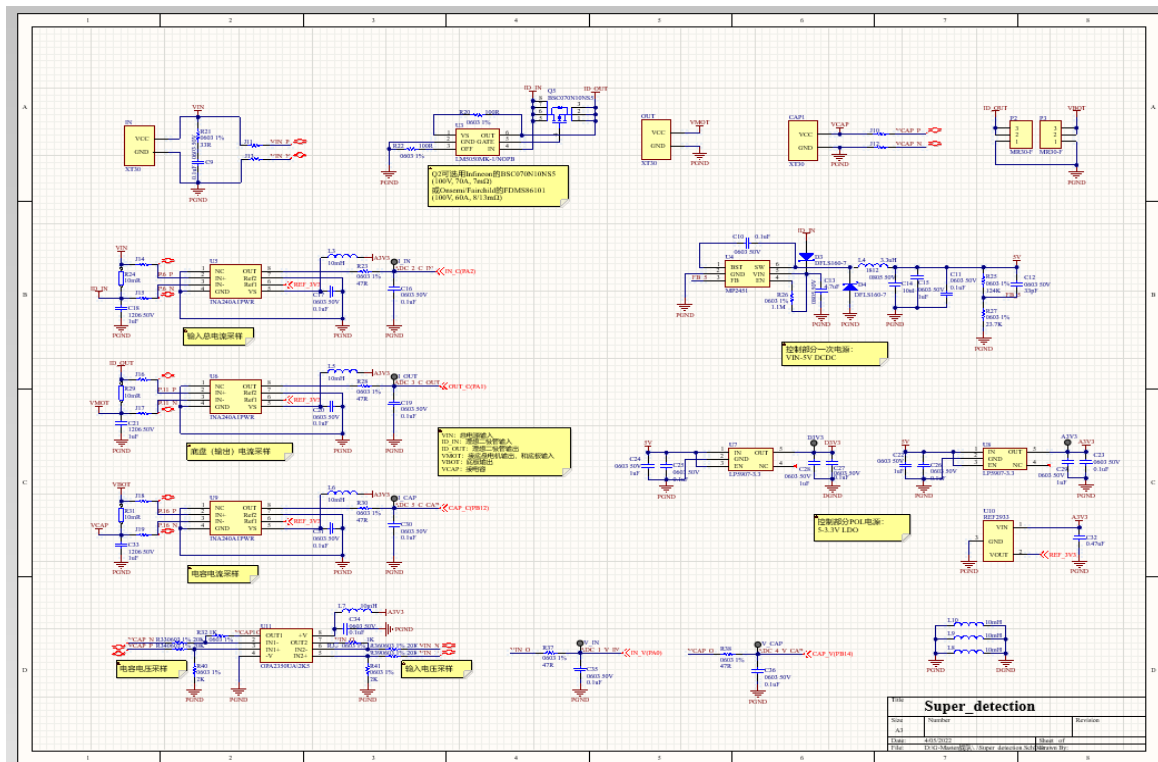


图 4.40: 控制模块-ADC 采样部分

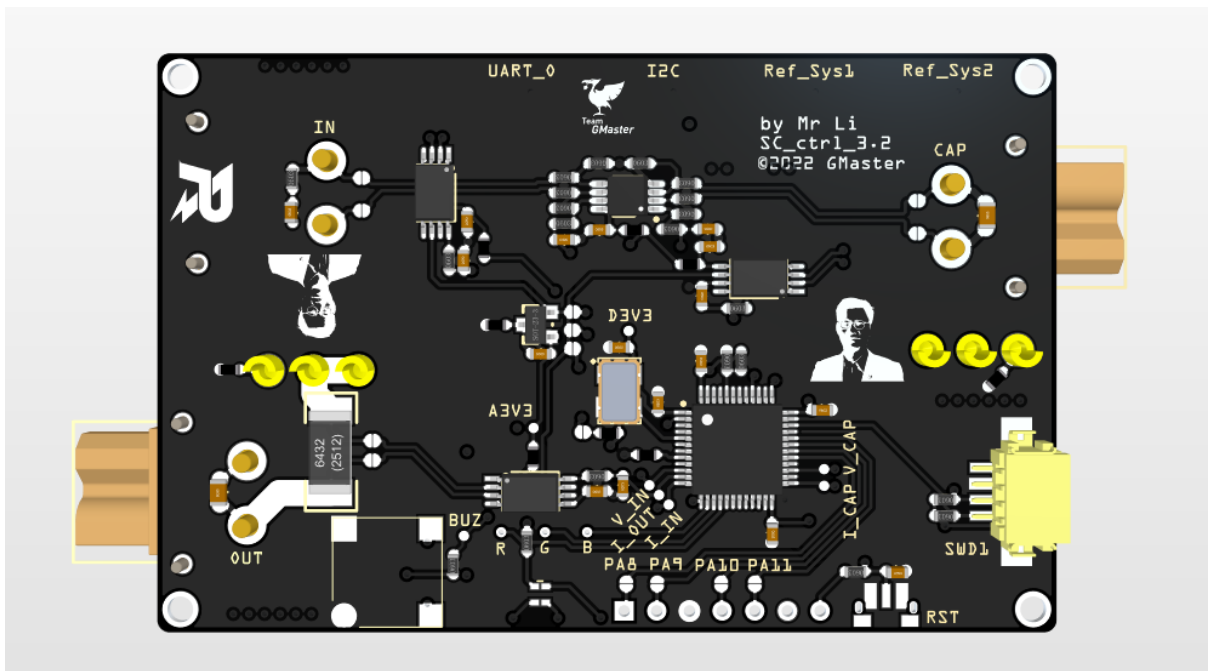


图 4.41: 控制模块 PCB 渲染图

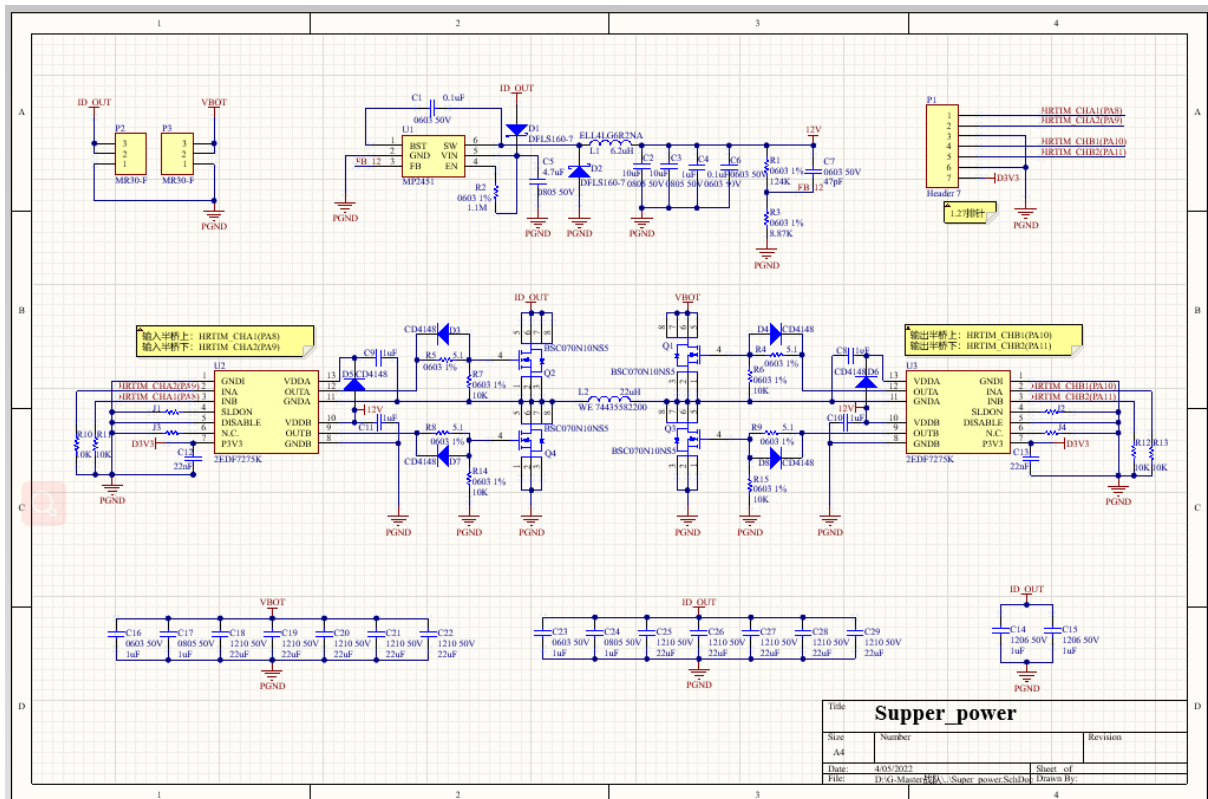


图 4.42: 功率模块

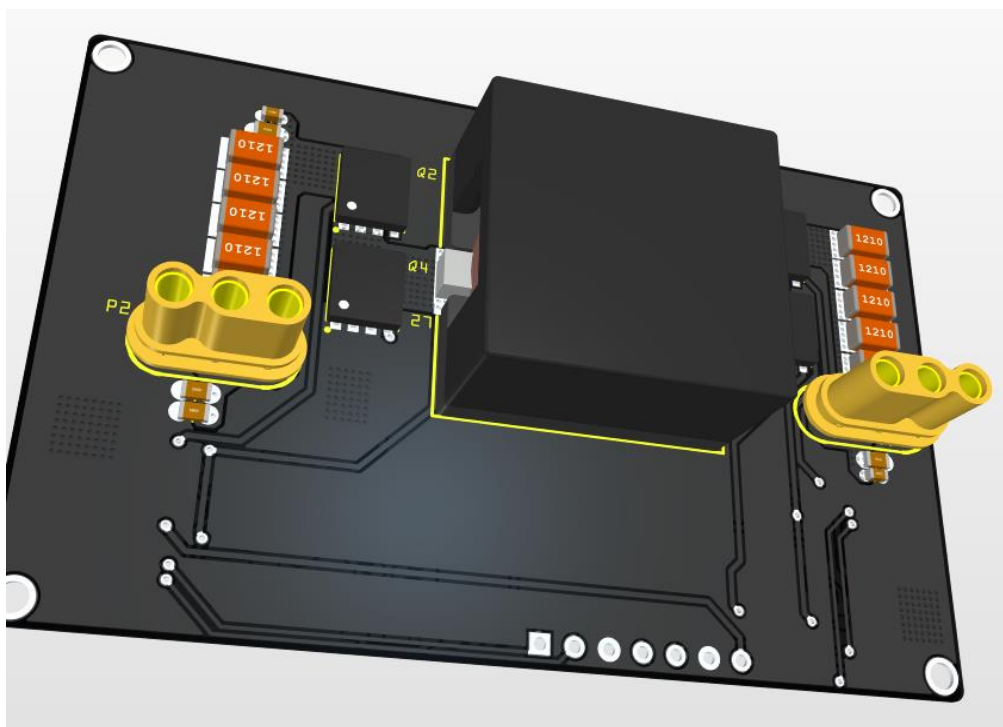


图 4.43: 功率模块 PCB 渲染图

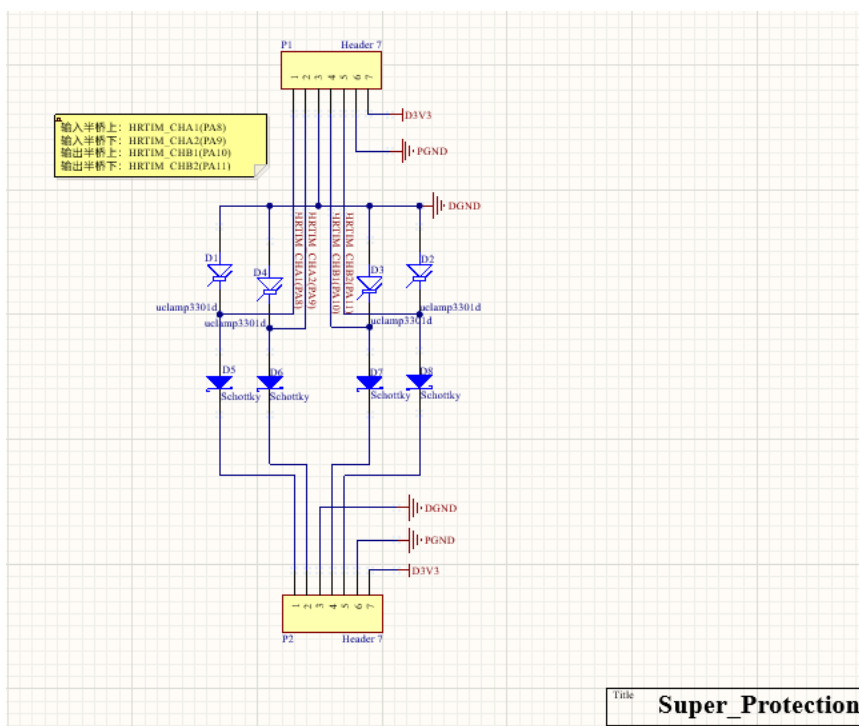


图 4.44: 保护模块



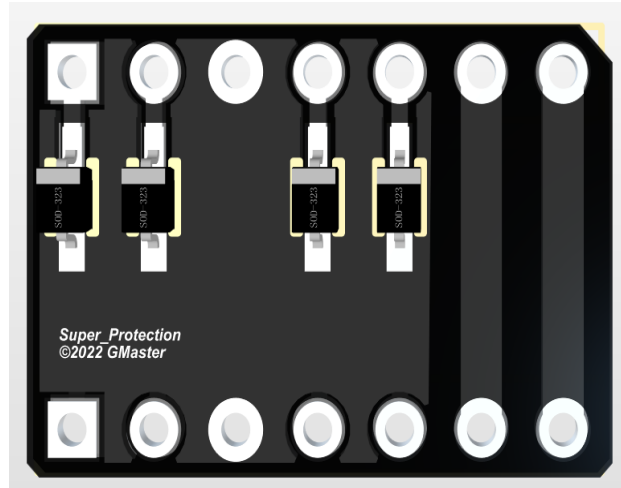


图 4.45: 保护模块 PCB 渲染图

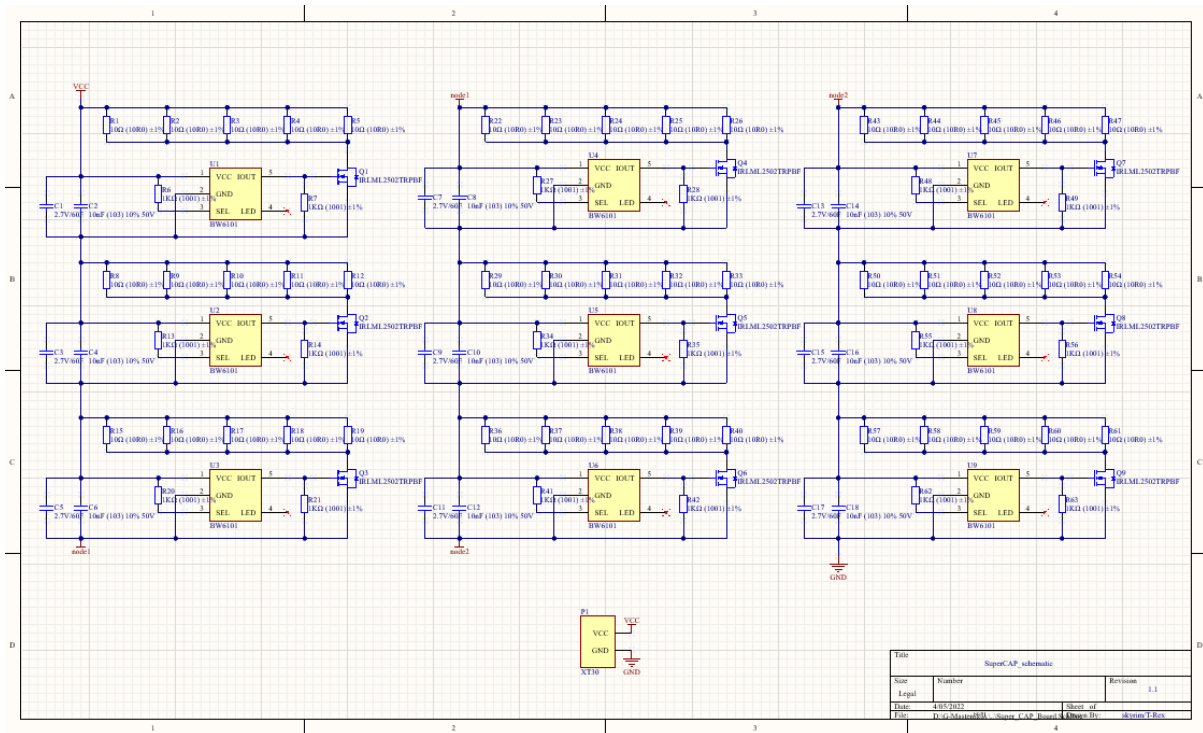


图 4.46: 电容组



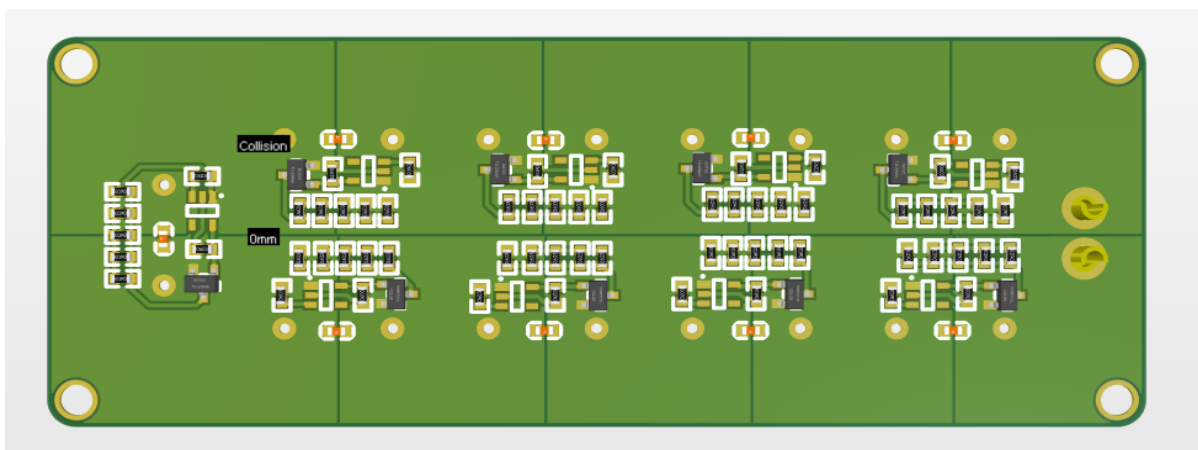


图 4.47: 电容组 PCB 渲染图

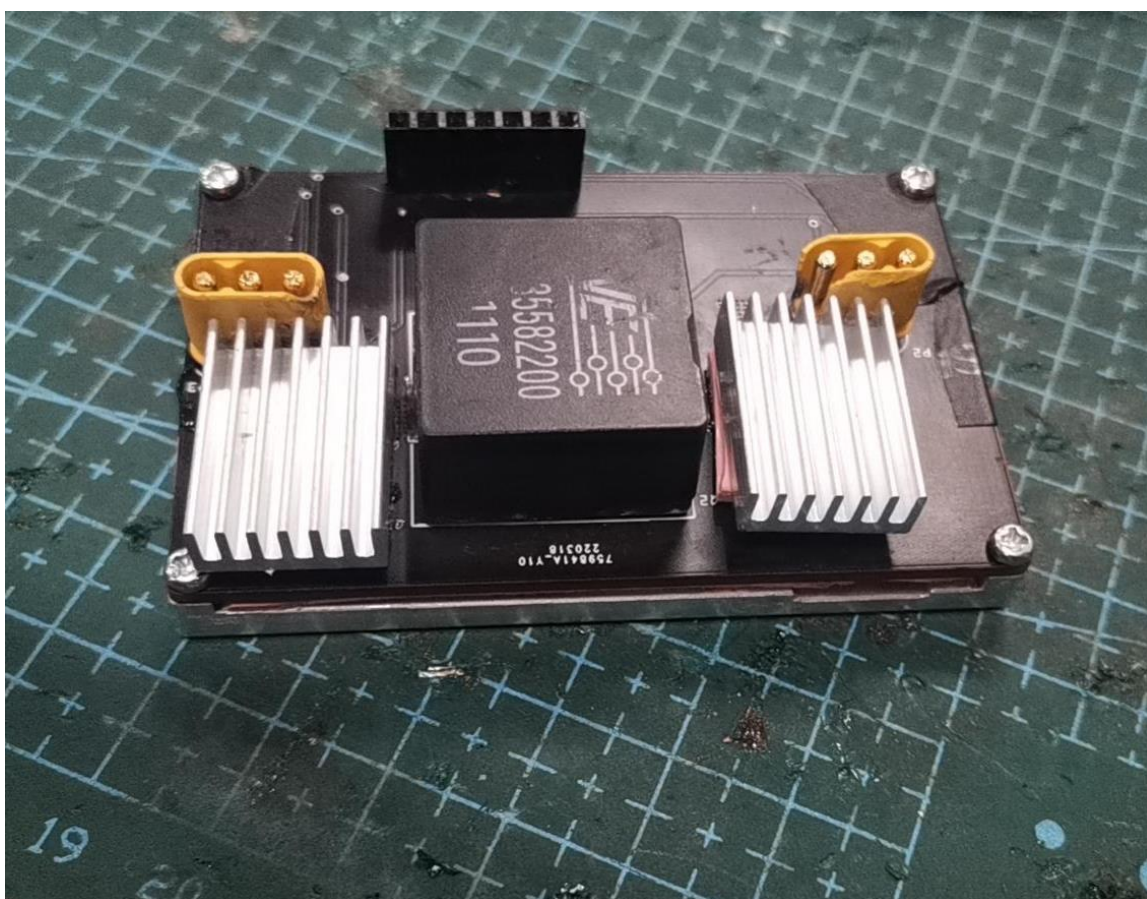


图 4.48: 电容控制模块（功率部分）实物图

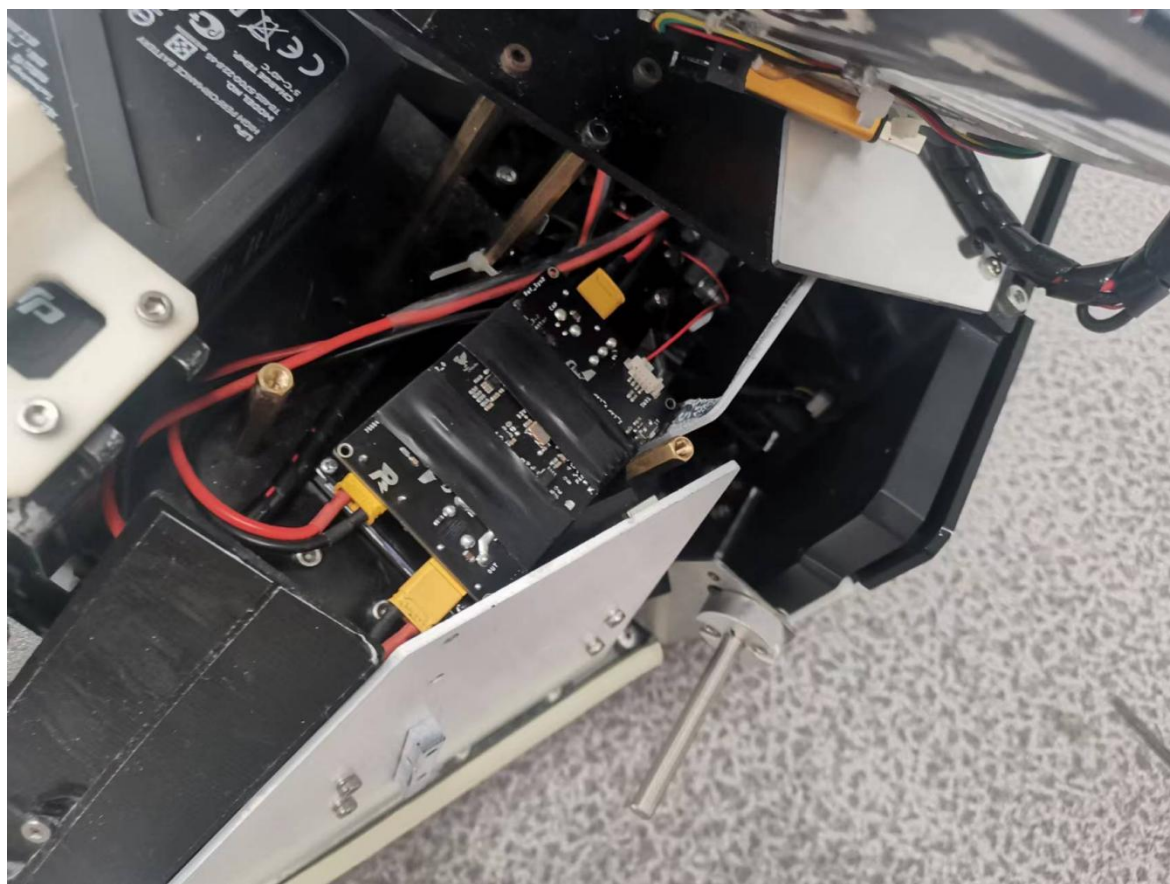


图 4.49.: 电容控制模块平衡步兵实装图

### 4.2.3 C 板扩展板

C 板扩展版是我们根据实际的接线需求而为 C 板定制的一块扩展版，他的大小可以和 C 板孔位完全贴合，通过铜柱连接后，可以使得两者成为一个整体。在这块分线板上我们有着：

- 24V 电压拓展（1 分 4）
- Can 信号拓展（1 分 4）
- 烧录拓展口

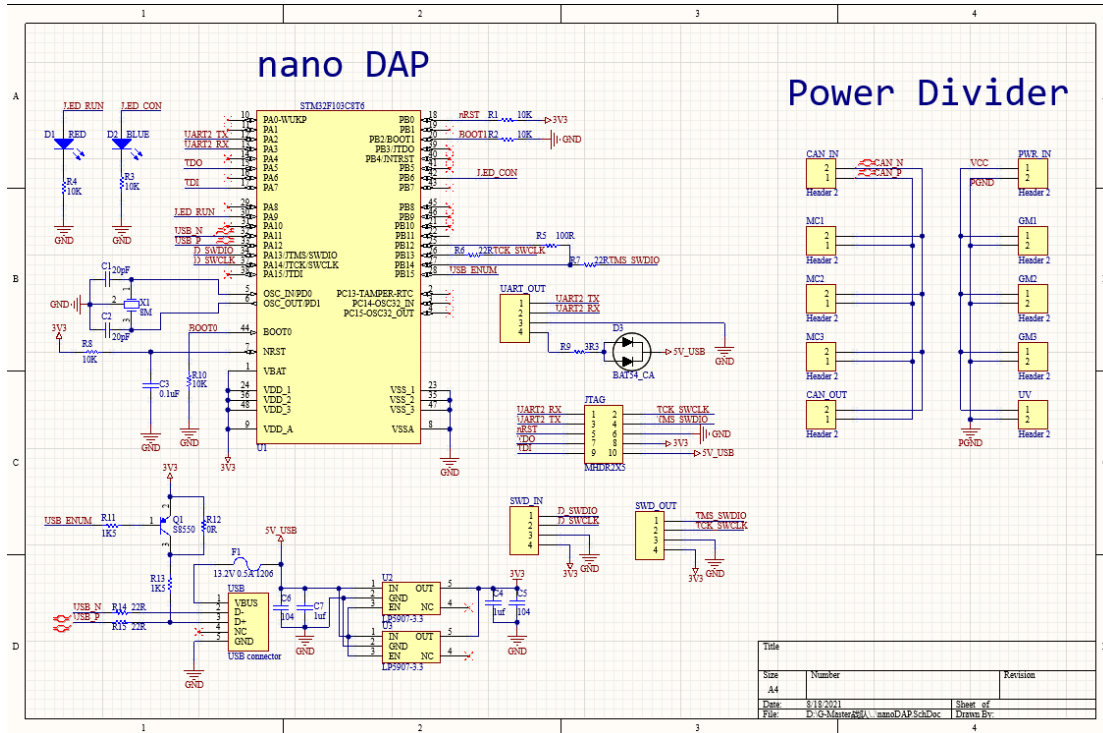


图 4.50: C 板扩展版原理图

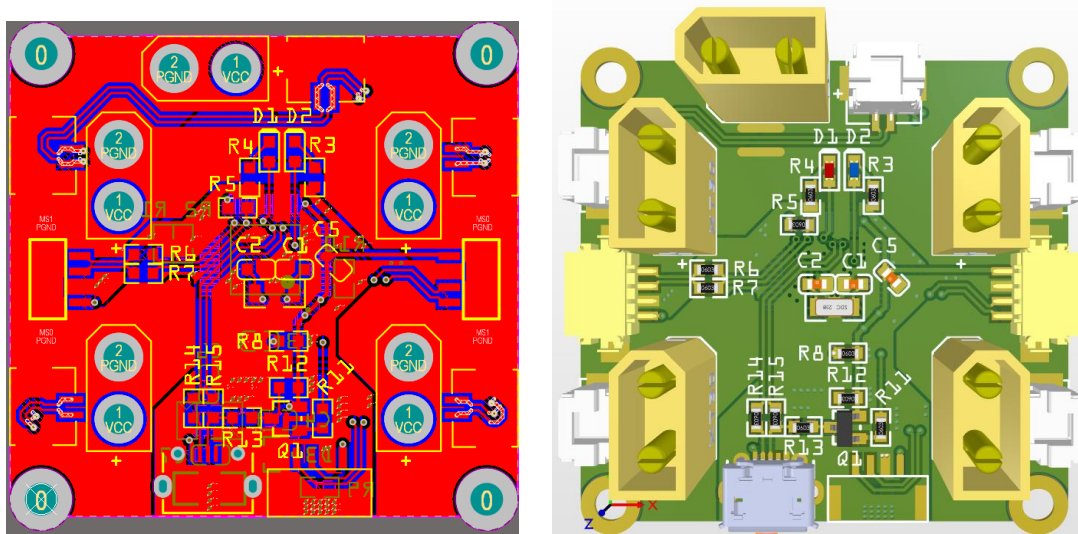


图 4.51: C 板扩展版 PCB 图

◆ 优点与好处:

诚然我们的这一设计是成功的，我们这块 C 板扩展板的设计仿照了一些大疆原版的设计。加以改进后，我们在确保其稳定工作的前提下大大降低了制作成本，使其更加符合我们步兵的整体设计要求，同时，他的存在也让布线，检修与烧录代码的工作便利了不少，大大提高了备赛时的容错率。

## 4.2.4 关键器件选型

### 4.2.4.1 MCU

电容控制板：STM32F334C8T6 [9]

Arm® Cortex®-M4 32-bit CPU with FPU (72 MHz max), single-cycle multiplication and HW division DSP instruction

- Memories
  - Up to 64 Kbytes of Flash memory
  - Up to 12 Kbytes of SRAM with HW parity check
  - Routine booster: 4 Kbytes of SRAM on instruction and data bus with HW parity check (CCM)
- CRC calculation unit
- Reset and supply management
  - Low-power modes: Sleep, Stop, Standby
  - VDD, VDDA voltage range: 2.0 to 3.6 V
  - Power-on/Power-down reset (POR/PDR)
  - Programmable voltage detector (PVD)
  - VBAT supply for RTC and backup registers
- Clock management
  - 4 to 32 MHz crystal oscillator
  - 32 kHz oscillator for RTC with calibration
  - Internal 8 MHz RC (up to 64 MHz with PLL option) – Internal 40 kHz oscillator

C 板拓展板：STM32F103C8T6 [10]

- ARM® 32-bit Cortex®-M3 CPU Core
  - 72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access
  - Single-cycle multiplication and hardware division
- Memories
  - 64 or 128 Kbytes of Flash memory
  - 20 Kbytes of SRAM
- Clock, reset and supply management
  - 2.0 to 3.6 V application supply and I/Os
  - 4-to-16 MHz crystal oscillator
  - Internal 8 MHz factory-trimmed RC
  - Internal 40 kHz RC
  - 32 kHz oscillator for RTC with calibration

### 4.2.4.2 IC 驱动

电容功率板：MD\_2EDF7275K [11]

The EiceDRIVER™ 2EDi is a family of fast dual-channel isolated MOSFET gate-driver ICs providing functional(2EDFx) or reinforced (2EDSx) input-to-output isolation by means of coreless transformer (CT) technology. Due to high driving current, excellent common-mode rejection and fast signal propagation, 2EDi is particularly well suited for driving medium- to high-voltage MOSFETs (CoolMOS™, OptiMOS™, CoolSiC™) in fast-switching power systems.

### 4.2.4.3 ADC 采样芯片

#### 电容控制板：INA240\_A1PWR [12]

The INA240 device is a voltage-output, current-sense amplifier with enhanced PWM rejection that can sense drops across shunt resistors over a wide common-mode voltage range from -4 V to 80 V, independent of the supply voltage.

#### OPA2350UA\2K5 [13]

The OPA350 series of rail-to-rail CMOS operational amplifiers are optimized for low voltage, single supply operation. Rail-to-rail input and output, low noise ( $5\text{nV}/\sqrt{\text{Hz}}$ ), and high speed operation (38MHz, 22 V/ $\mu\text{s}$ ) make the amplifiers ideal for driving sampling Analog-to-Digital (A/D) converters.

## 4.3 软件设计

### 4.3.1 系统架构

#### ◆ 第三方中间件

云台开发板和底盘开发板均使用 FreeRTOS 作为嵌入式开发的操作系统。陀螺仪数据使用 DJI 的 AHRS 姿态数据融合算法库。操作手客户端 UI 绘制模拟器使用 PyQt5 作为 GUI。

#### ◆ 开发环境

项目开发平台为 Windows，使用开发软件为 STM32CubeIDE，STM32CubeMonitor，Clion 等。Git 协作平台采用 GitHub。



### 4.3.2 运行流程

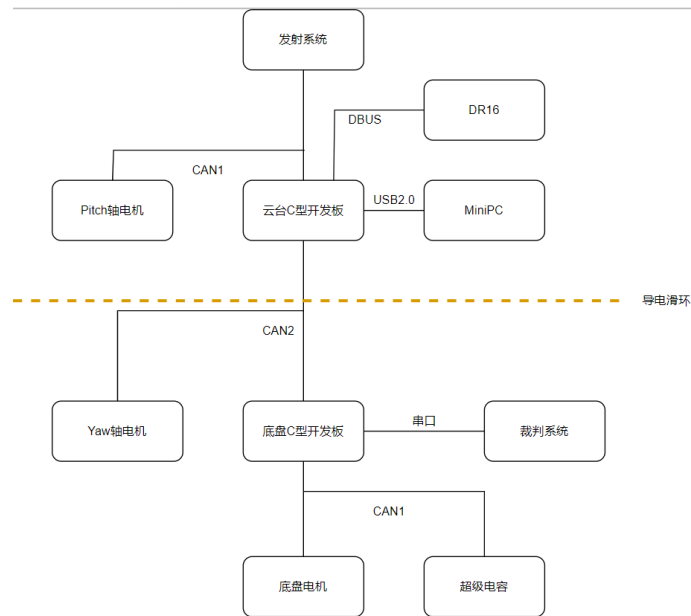


图 4.52: 平衡步兵结构框图

为了完成底盘控制任务，控制底盘平衡状态需要在底盘上安装IMU模块获取角度。同时，考虑到单个CAN总线挂载电机数量的限制以及裁判系统串口位于底盘的情况，将底盘电机直接与底盘C板CAN1回路相连，两块C板使用CAN2总线实现相互通信和数据共享。

云台开发板作为全车的决策模块，负责控制全车的运行控制，包括云台角度控制，发射机构控制，视觉上位机通信，遥控器数据接收，与底盘通信等任务，云台角度控制任务通过读取云台IMU角度和遥控器输入，通过CAN总线控制云台电机，发射机构控制负责读取遥控输入，根据发射模式控制摩擦轮和拨弹轮电机，视觉上位机通信任务负责通过USB CDC串口与安装在云台上的NUC通信，发送当前自瞄模式和陀螺仪四元数以及时间戳，接收视觉下发的目标是否发现以及目标角度。云台C板作为整车的控制中心，同样需要负责将一次控制循环得到的底盘速度 $V_x$ ， $V_y$ ， $W_z$ ，和底盘控制模式经CAN协议发送至底盘开发板。

底盘开发板作为步兵底盘速度的执行模块，负责底盘的控制和通信，底盘控制任务将接收到的云台下发的目标速度和底盘模式作为控制目标量，将底盘陀螺仪数据和电机编码器数据作为反馈值进行控制，同时通过读取IMU平衡机器人底盘角度，底

盘 C 型开发板需要与裁判系统进行通信，通过串口接收裁判系统发送的比赛数据，同时发送自定义 UI，裁判系统下传数据将通过 CAN 发送到云台 C 型开发板，超级电容同样被连接到了底盘 CAN 总线上，底盘 C 型开发板可以根据裁判系统反馈和电容数据进行功率控制。

### 4.3.3 UI

#### 4.3.3.1 操作手客户端底盘状态显示

操作手 UI 作为人车交互的重要方式，可以极大的简化操作手的操作难度。一个功能强大的 UI 是必不可少的。同时，UI 界面上的底盘状态显示有利于操作手更好的感知运动底盘状态，由于底盘小陀螺状态时，客户端原始的伤害来源方向提示不再准确。因为我们通过读取云台数据，结合裁判系统信息，给出准确的伤害来源方向，并在体现在车辆状态上。

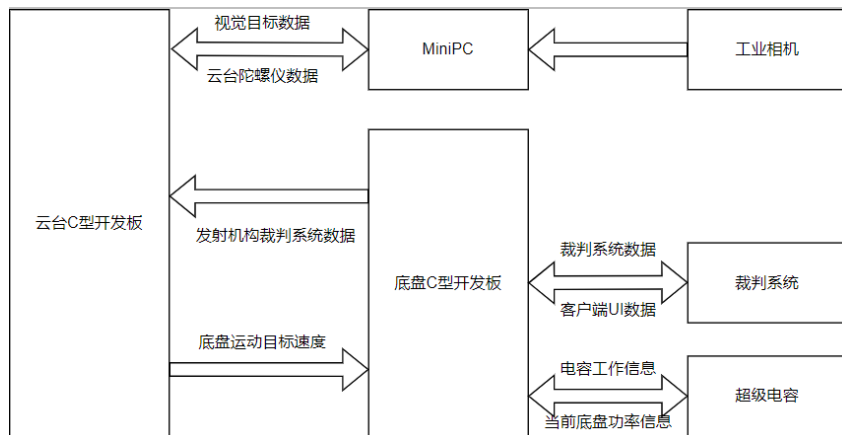


图 4.53: 数据流向

#### 4.3.3.2 功能架构

由于不同兵种需要不同的 UI 界面，为了提高开发效率，设计了具有高可移植性的架构。



图 4.54: UI 架构图

Widget 和 Platform 都可独立使用。无论机器人方面如何实现功能，只需要调用 Platform 模块中相应的接口即可实现 UI 功能对相关的数据的引用。可修改 Core 来改变显示样式和内容适应不同兵种，而完成这一条件的需要 Widget 即控件的支持。

控件要求在使用时只需设置好相应的结构体，然后将结构体传入相应的函数即可。开发目标为在创建结构体时写明该控件的位置，提供与控件相应的内容刷新函数（如 ‘info\_change(info\_handle, "example")’ 修改显示信息）。目前一共开发了三种类型的控件如下：

控件	功能
信息提示	可以一键修改显示信息，只需提供一次信息类型，如 SPEED: XX，刷新时只需提供 XX 即可
车辆	引用云台绝对角度和底盘相对角度，实现对车辆状态的绘制与刷新。调用裁判系统数据，将最近 0.5S 收到击打的底盘边框变色
进度条	可设定“框”与“条”的颜色，一键修改进度百分比

表 4.8: UI 控件

实机运行流程如下：



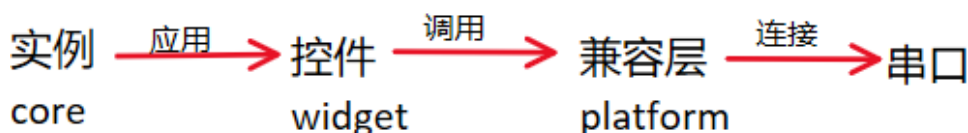


图 4.55: 实机运行流程

### 4.3.3.3 模拟器

传统 UI 绘制需要裁判系统，开启局域网，开启裁判系统服务器，开启操作手客户端，面临着诸多的问题。

为了方便测试，我们开发了一套在电脑上运行的模拟环境，并开发了相应的“兼容层”文件，使绘图指令在通过套接字传输至模拟器端，实现 UI 模拟。调试时只需更换相应的兼容层文件，即可将代码“无缝对接”至模拟器。

模拟器运行流程如下：

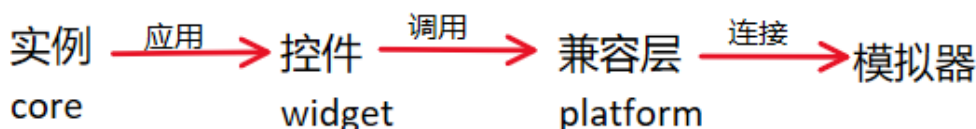


图 4.56: 模拟器运行流程

模拟器展示如下：

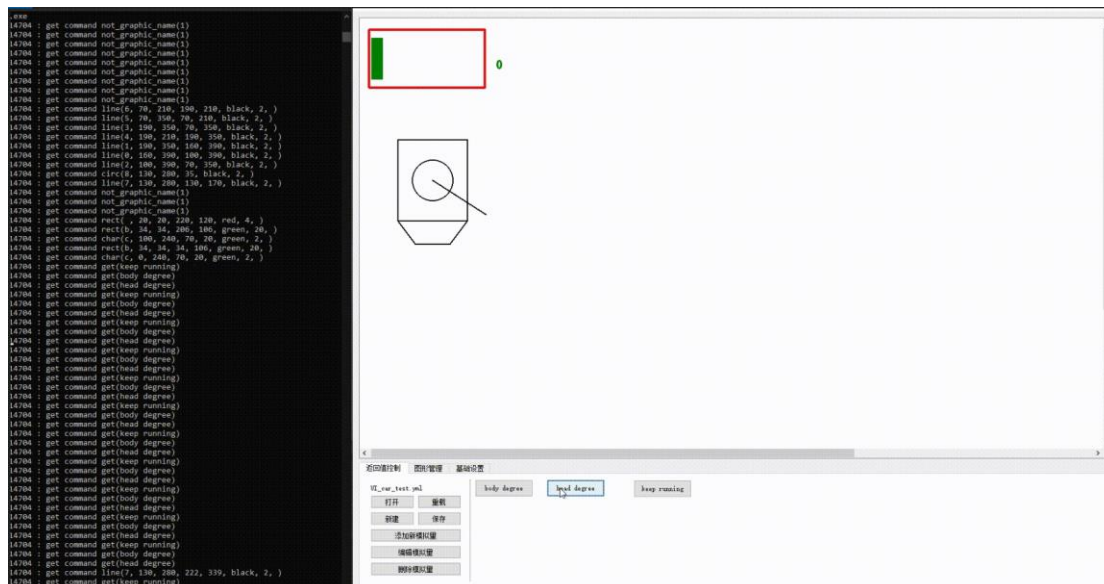


图 4.57: 模拟器展示图

- ◆ 车辆控件

车辆控件需要先创建`car\_handle`结构体，结构体中`basic\_config`的值需要手动设置。相关参数示意：

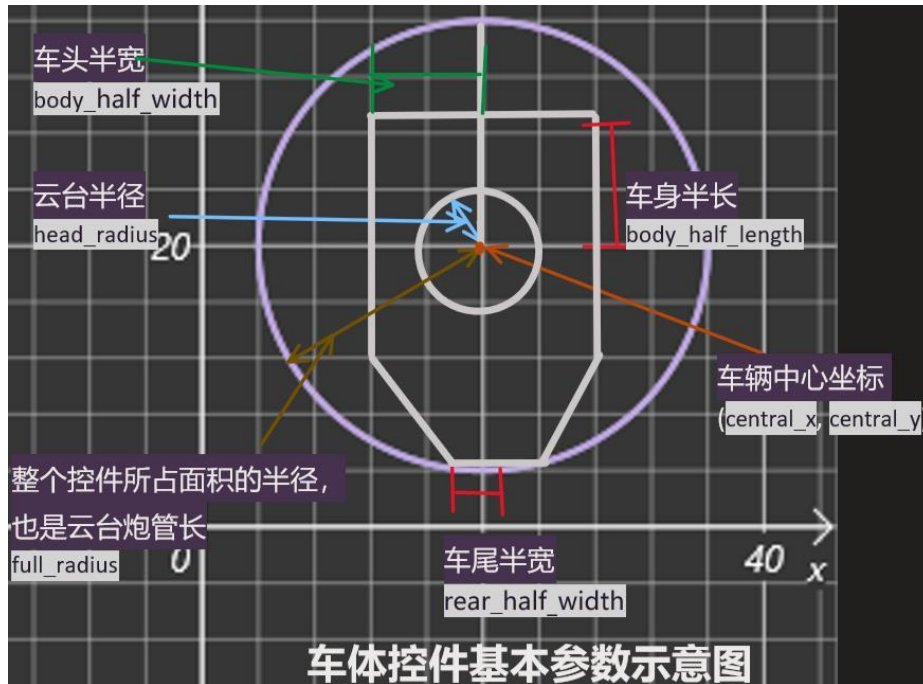


图 4.58: 车体控件基本参数示意图

本控件中所有度数选择了对人类更为友好的 $[0, 359]$ 角度制，且以正上方为  $0^\circ$ ，顺时针为正方向。当控件状态发生改变（如云台旋转）时，无需手动操作句柄，无需手动检测是否需要修改，这些任务由控件执行。

#### ◆ 标签控件

标签控件主要显示文字信息，为操作手提供视觉、自瞄、拨弹轮、摩擦轮的实时信息。标签控件由 `UI_LABEL_DATA` 储存状态数据，使用 `UI_LABEL_STATIC` 函数绘制初始化状态，使用 `UI_LABEL_CHANGE` 函数绘制变化后的状态。

#### ◆ 部署过程

在实现这些控件后，我们以如下步骤将控件部署到机器人端：

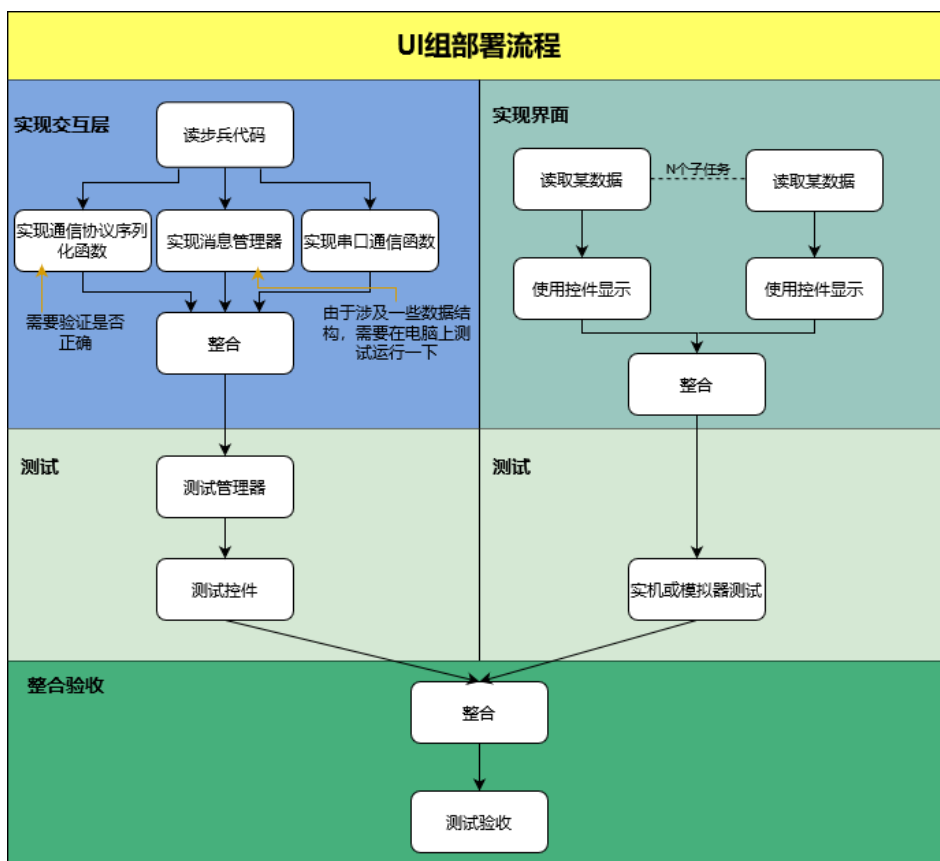


图 4.59：部署过程流程图

其中，实现“消息管理器”因为串口协议允许一次发送七条命令，然而我们的绘制函数调用一次只生成一条命令。消息管理器就是把这些命令暂存起来，到适当的时候一次性发出去。“通信协议序列化”则是将命令打包为可直接进行串口通信的数据。

借助此过程，通过“多进程开发”弥补架构带来的开发慢的负面影响。

◆ 进度条控件

该进度条控件由一个进度条一个实时数字提示组成。其中实时数据坐标与进度条坐标分离，两者坐标应在初始化时设定。对于进度条，分边框 border（开头量）和填充 ProgressFilling（开头量）两部分，这两部分共用同一手动输入坐标（边框坐标），使用时应手动设置各自的颜色编码和图形宽度；实时数据提示 percent 与进度条分离，故除去实时数据外需自己设置字体颜色，大小，位置。所有量均在 `progress\_bar\_data` 结构体中。使用时需先创建 `progress\_bar\_data` 结构体。每

需新建一个进度条（不含该进度条名称的标签，若要添加需使用 label 控件），即需要创建`progress\_bar\_data`结构体，该结构体包含所有参数信息。

#### ◆ 刷新任务

为了处理运行中裁判系统意外断开后重连情况，提高的程序运行的鲁棒性，我们将主程序分为初始化（init）、循环（loop）。其运行关系如图：

在每次检测到裁判系统离线时，停止 UI 刷新，当检测到裁判系统重新上线时，先发送指令，清除所有图层内容，再重新进行 UI 的绘制。

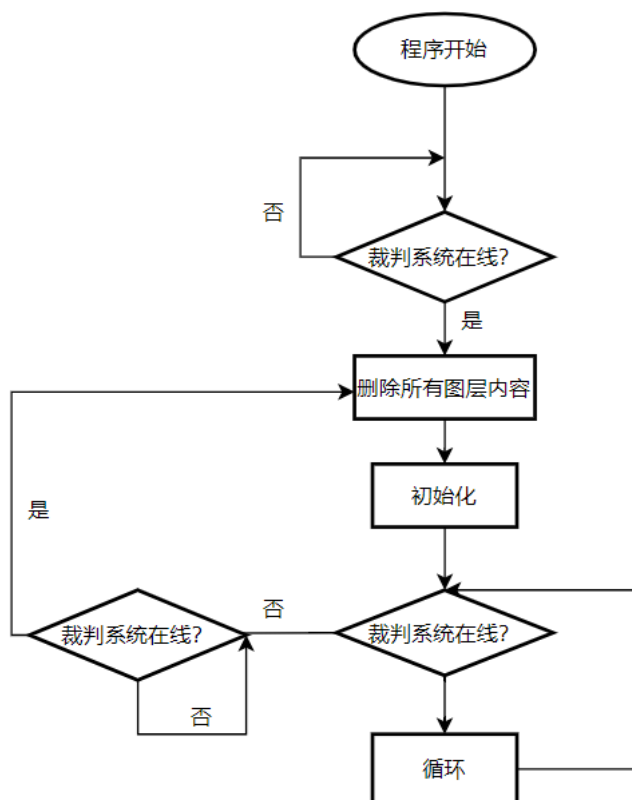


图 4.60: UI 程序流程图

#### ◆ GUI

使用山东理工大学开源的自定义 UI 的封装库[14]实现客户端对于点线的绘制以及其他图形的操作。

由于裁判系统通信速度不稳定，计算三角函数耗时过长，底盘小陀螺旋转速度较快。车辆控件在比赛过程中并没有发挥出预期效果，在未来可以考虑简化车辆控件，使用其他的方式来表现车辆受击方向。

#### 4.3.4 平衡机器人的全向移动算法

常规平衡车不需要进行横向运动，因此只需要使用陀螺仪对底盘Yaw轴角速度和底盘角度进行闭环。但是由于麦轮平衡车需要进行移动小陀螺等更加复杂的运动，需要更为精确的横向运动速度。同时由于横向运动的速度传感器不方便感知（加速度传感器误差过大），因此为了更好的控制底盘三个方向的运动速度，在控制算法的设计上，我们使用麦轮步兵的逆向运动学公式，对底盘电机速度进行闭环[15]。该方法可以取消转向环，保证横向速度的精确输出。

平衡机器人的全向移动算法作为最关键的一部分，该算法设计之初便以实现其全向移动的特点而搭建。与传统两轮平衡车的系统相比，取消了转向环，应用逆向运动学公式。最终实现了移动小陀螺等特色功能。

##### ◆ 运动学分析

对于全向运动的机器人，其逆向运动学公式可以简单的认为是三个方向速度的相加，得到最终的四轮速度。同轴麦轮的全向移动公式与普通麦轮步兵基本一致，但因电机朝向不同，其电机旋转方向会有所差异。

逆向运动学公式如下：

$$V_1 = V_x + V_y + k_1 V_w$$

$$V_2 = -V_x + V_y - k_2 V_w$$

$$V_3 = V_x - V_y - k_2 V_w$$

$$V_4 = -V_x - V_y + k_1 V_w$$

这里 $V_1, V_2, V_3, V_4$ 均为电机速度，而非轮子旋转速度。 $k_1, k_2$ 为轮子距离旋转中心的距离。

##### ◆ 算法流程

最终，麦克纳姆轮平衡步兵的控制流程如图 4.56。主要分为两部分，第一部分为  $V_x$  与底盘角度控制，第二部分为平移旋转控制。分别由两个控制模块并行计算，各自输出四轮速度。

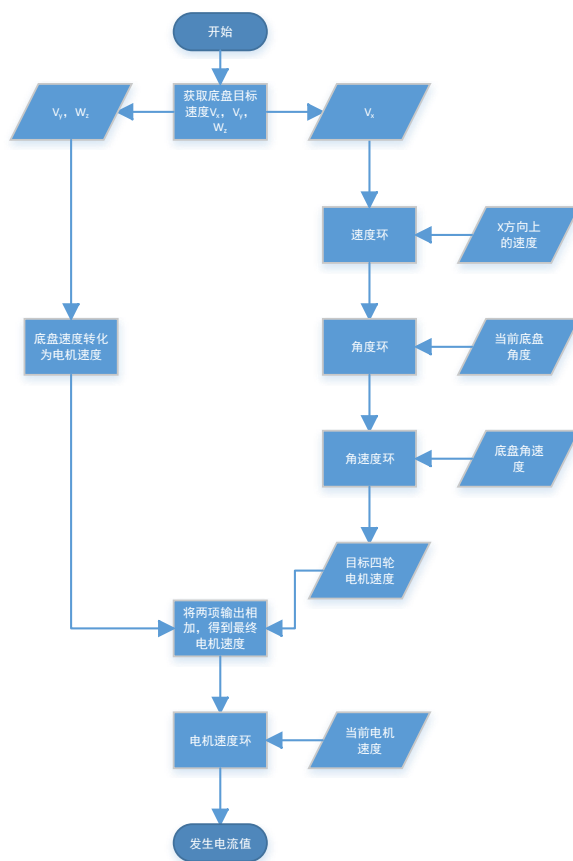


图 4.61：底盘控制示意图

◆ 前进与底盘角度控制

该模块主要有三级串行 PID 构成，第一环为速度环，控制底盘输出合适的角度以达到目标速度，第二级底盘角度闭环，根据上一级的目标角度，输出合适的底盘角速度，实现对底盘角度的控制，第三级根据底盘角速度输出合适底盘电机速度。

速度环  $V'_x$  的反馈数据求解逆向运动学方程组可得：

$$V'_x = -V_1 + V_2 - V_3 + V_4$$

该算法将速度环与角度环串联，并设置为同样大小的控制时间，通过角度环输出最大值的限制，不会造成角度与速度的正反馈控制的情况。但是目前角度环仍然存在静止时震荡的情况，这是由于 3508 减速箱虚位，造成  $V_x$  速度反馈噪声较大，使用

一阶低通滤波后的微分项仍然无法取较大值造成的。之后可以更换驱动电机，使用 LRQ 等更先进的控制器来解决该问题。

#### ◆ 旋转与平移运动控制

根据逆运动学方程，以  $V_1$  代表的电机为例，逆运动学方程为

$$V_1 = V_x + V_y + k_1 V_w$$

观察公式得出，旋转速度可以直接带入逆向运动学方程并求解，其他电机同理，由于底盘 C 型开发板控制系统设计原因，该控制系统设计为

$$F_{set} = PID(V_{set}, V_{fdb})$$

可以看到，并无设计机器人的转向环，即底盘并不能自行闭环控制其角度，为了实现转向的目标，云台 C 板的 chassis\_task 负责处理底盘的 Yaw 轴角度环，通过 IMU 角度和 Yaw 轴云台电机角度作为控制量和反馈值，计算出底盘旋转角速度，并通过 CAN 通信发送到底盘 C 板，实现了底盘跟随云台。底盘 C 板直接收到角速度直接带入逆向运动学方程即可，经测试，该开环系统可以得到较为准确的角速度。

#### ◆ 移动小陀螺

基于麦轮的平衡步兵相较于传统两轮平衡步兵，速度解算公式存在  $V_y$  速度，理论上具有全向移动能力，因此可以实现移动小陀螺运动，平衡小车加速时，根据重力分力公式

$$F = mg \sin \theta_1 \cos \theta_2$$

$\theta_1$  为底盘倾角， $\theta_2$  为底盘与云台相对角度。

根据公式，加速度需要底盘保持一定倾角以通过重力的分量产生加速度。因此在进行移动小陀螺运动时会产生一个周期性变化的 Roll 轴摆动。限于平衡步兵原理和整车系统性能，无法得到较大的旋转速度和旋转时的移动速度，移动小陀螺仅作为在旋转时调整自身位置的一个手段。

目前该算法实现了麦轮平衡步兵的全向移动功能，具有较好的平衡移动效果。但是还有一些需要改进的地方，可以使用 LQR 等更为先进的控制器优化平衡效果，并考虑其他算法优化移动小陀螺的倾角问题。



## 4.3.5 控制优化及创新点

### 4.3.5.1 云台 PID

本赛季继续沿用上赛季所采用的基于微分先行、Gain-Schedule 搭配一阶低通滤波的云台双环串级 PID 控制算法，该算法具有追踪性能好，且对于自瞄系统输入的近似阶跃信号相应速度快、无震荡等优点。

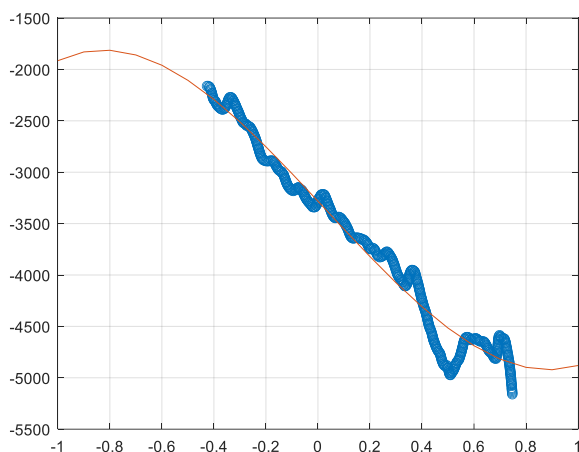


图 4.62：云台角度与稳态电流拟合效果

本赛季在该算法基础上对 Pitch 轴加入前馈，使用 MATLAB 系统识别工具拟合出云台角度与稳态误差电流的函数。此法将减少由积分项所造成的延迟，使弹舱满载情况下追踪性能由上赛季的 70ms 延迟提升至 55ms 左右。

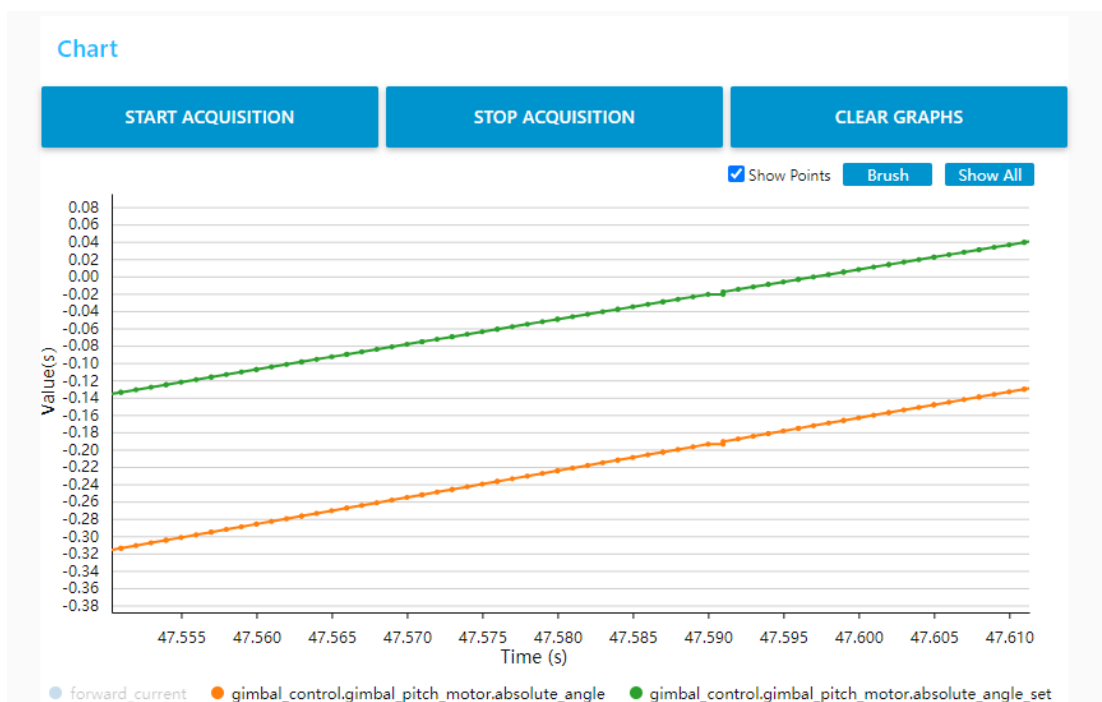


图 4.63：优化效果图

#### 4.3.5.2 底盘电机功率控制算法

本赛季为我们队第一次大规模使用超级电容模组，共使用两种不同的拓扑结构。针对电容与底盘并联与非并联的拓扑结构。我们研发出一套同时适用不同拓扑结构的串级 PID 的底盘功率控制算法，进行底盘输入功率的合理分配和底盘功率控制。该算法通过将电容容量稳定在任意大小，以实现主动的开启或关闭超级电容输出的能力，同时在任何电容电量下都可以保证底盘裁判系统输出功率优先响应操作手指令。

##### ◆ 缓冲能量闭环

根据裁判系统反馈数据的功率上限和当前缓冲能量，计算出当前最大底盘输入功率，发送给超级电容控制板，将缓冲能量稳定在 20 焦耳。该闭环可以防止长时间高功率输入的控制误差对缓冲能量造成的连续消耗，从而避免超功率。同时可以迅速使用飞坡增益的高额缓冲能量，以快速恢复电容电量。

##### ◆ 电容电量闭环

根据超级电容原理

$$P_{in} = P_{cap} + P_{chassis}$$

在输入功率不变的情况下，可以通过主动控制底盘电机功率输出，来改变电容的充放功率。因此我们通过限制四个底盘电机的电调发送电流值，来限制底盘电机功率输出，最终实现电容容量控制。

为了提高该环的响应速度，我们让底盘电机输出功率在底盘输入功率上限的基础上进行浮动。根据测试结果，在计算初始浮动值时，我们简单的认为 CAN 通信发送值 N 与底盘功率有如下关系。

$$N_{cmd} = P_{motor} \times 250$$

该环反馈值为当前电容容量或电容电压，设置值为目标电容容量或电容电压，输出值为 CAN 通信发送值的变化量，通过 PI 控制将电容容量稳定在目标值。

最终，我们得到了缩放系数 k

$$k = \frac{P_{in} \times 250 - PID_{out}}{\sum N_{cmd}}$$

注：电机输出减小，电容容量变大，公式中注意 PID 输出符号为负。

给电机的最终指令为  $N'_{cmd} = kN_{cmd}$ ，在 k 大于 1 时不进行缩放。

该环实现效果为，可以将电容容量稳定在任意大小，当设定值从高容量，设定为低容量时，即为开启超级电容，电容额外放电，容量变小并最终稳定；当设定值从低容量设定为高容量时，电容充电，容量增大并最终稳定。

#### ◆ 动态电容目标值设定

分区赛使用电容与底盘并联的拓扑结构，将电容目标值设定为两个固定值，在开启电容时，电容电压设定在 15V，关闭时电容电压设定为 *电池电压-1V*。但是简单的两个目标值会出现，当操作手在低电容容量选择关闭超级电容时，该算法会大幅度限制底盘功率以将电容容量恢复至较高状态。此时操作手几乎会失去对于底盘的控制。

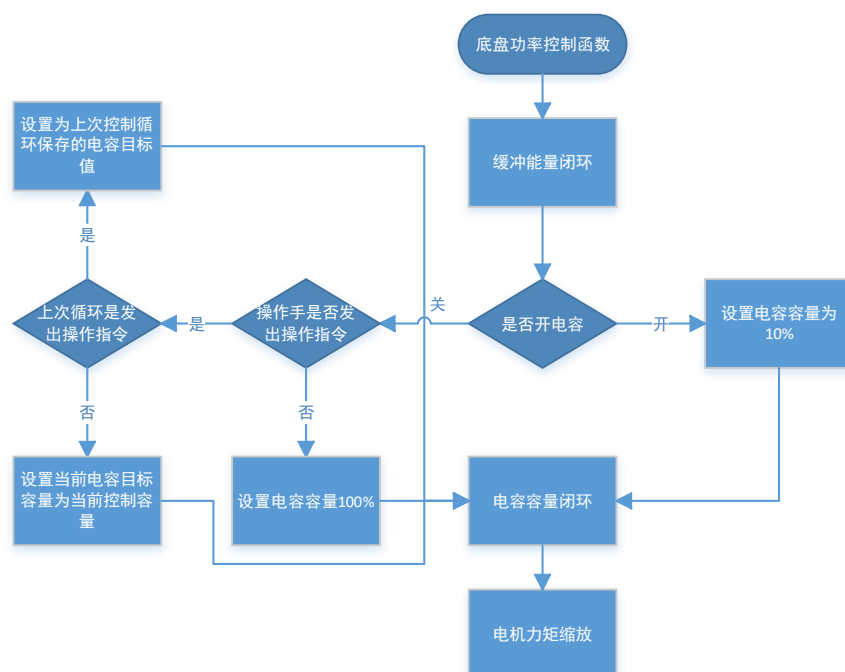


图 4.64: 超级电容控制闭环流程图

分区赛后我们使用动态的电容目标值以解决该问题。在开启电容时仍使用固定的低容量目标值。

在关闭电容时，当操作手未发送移动指令，即未按下 WASD，小陀螺按键或主动刹车按键，将选择一个高电容容量为目标电压值。

当操作手发送移动指令，将选择当前电容容量为目标电容值并保持该目标大小，直到操作手停止发送指令。由于积分项响应速度较慢，不能每次控制循环都选择一个

新的目标值，应该在连续运动时保持相同目标值。此刻电容将停止充电和放电，底盘功率将优先用于响应操作手指令。

当操作手停止发送控制指令，目标值将恢复为高容量。

动态电容目标值将保证在操作手发出移动指令时，底盘功率不会因为电容充电而被严重限制。同时，因为没有指令时电容容量为高容量值，该算法还可以实现一个动态的被动刹车功率控制：低电容容量时刹车功率小，高电容容量时刹车功率高。

由于基础功率浮动值随电池电压，底盘回路电压等其他因素影响较大，目前该算法主要依靠积分项进行稳定的功率限制。对于电容与底盘并联的拓扑结构，在不同的底盘电压下，电机的输出功率与指令值的关系会有所不同。在未来，还可以通过更加精确的电机功率模型来减少对积分项的依赖，以提高响应速度。还可以将底盘目标速度进行动态的调整，减少对于被动限制的依赖，间接提高响应速度，同时保证底盘精确的移动。

#### 4.3.6 软件测试流程

平衡步兵软件测试重点内容分为三部分，平衡稳定性，加减速及电容容量控制，具体情况如下表 4.9

名称	内容	结果
平衡稳定性	观察平衡时的底盘稳定性，调整角速度环参数。保持机器人长时间处于平衡状态，监测其有无软件问题（操作系统实时任务问题、内存溢出）	平衡时底盘有肉眼可见的极小轻微摆动，原因主要来源于3508减速箱虚位以及速度环与角度环串联的系统设计。但对于云台的影响较小。长时间平衡未出现底盘失控问题
加减速	长距离极限加减速，多次测试。观察其角速度环性能能否保证急速情况下刹车，并恢复底盘状态	未出现角速度性能不够的情况
电容容量控制	在50W功率下，在起伏地面，上下坡等地形进行小幅度加减速，小陀螺，急速行驶刹车，长时间移动等测试。观察电容	电容容量跟随性能良好，未出现超功率或底盘失控等问题

名称	内容	结果
	容量是否稳定在设定值，是否出现底盘失控的情况	

表 4.9：软件测试重点内容

平衡步兵的完整测试结果见下表 4.10

项目	性能	备注
爬坡能力	<ol style="list-style-type: none"> <li>有无超级电容情况下，均不具备上长坡（15°）的能力，但是在无超级电容 70w 功率及有超级电容的情况下，可以上 13°坡</li> <li>不具备上障碍块的能力</li> </ol>	<ol style="list-style-type: none"> <li>有/无超级电容情况下多少功率能够拥有上长坡能力；</li> <li>有/无超级电容情况下多少功率能够不停顿上长坡；</li> <li>障碍块放置在台阶位置的爬坡；</li> </ol>
射击精度	<ol style="list-style-type: none"> <li>5 米在评审材料中射击小装甲板 100 命中率，散布半拳</li> <li>7 米的精准度散布一拳左右</li> </ol>	<ol style="list-style-type: none"> <li>5 米散布</li> <li>7 米散布</li> </ol>
自瞄	<ol style="list-style-type: none"> <li>对于不开小陀螺的单位，命中率 100%</li> <li>对于开启小陀螺的单位，命中率 50%以上</li> </ol>	1、2、4、5 米击打开/关小陀螺敌人的能力

项目	性能	备注
能量机关	大、小能量机关均能在 5-10 秒内开启，具体时间取决于两发弹丸的时间间隔，一般不存在开启失败需重新击打的可能	7 米距离
飞坡	从机械角度出发，不宜飞坡	未使用 22 赛季坡（17 度），使用的坡为 19 赛季的坡（20 度，空隙比本赛季大 200mm）
弹速稳定度	<ol style="list-style-type: none"> <li>1. 30m/s --&gt; 27.5 - 28.5m/s</li> <li>2. 18m/s --&gt; 16.2 - 17m/s</li> <li>3. 15m/s --&gt; 13.6 - 13.9m/s</li> </ol>	<ol style="list-style-type: none"> <li>1. 15m/s</li> <li>2. 18m/s</li> <li>3. 30m/s</li> </ol>
下台阶能力	从机械角度出发，不宜下台阶	200mm 台阶
起伏地段行动能力	<ol style="list-style-type: none"> <li>1. 前后左右穿行不垂直于起伏地段时基本不受影响</li> <li>2. 小陀螺会导致云台晃动干扰较大，不宜射击大于 3 米的目标</li> </ol>	<ol style="list-style-type: none"> <li>1. 前后左右移动</li> <li>2. 小陀螺</li> </ol>
射频	能够按照需求实现，双枪切枪不会因此掉血	两档：7/20



项目	性能	备注
弹仓容量	最大接近 300 发，赛场上为防止浪费保持 250 发	
画面 UI	均稳定实现并能够在发生变化时有所切换	辅助线 1. 1、2、4、5、7，11m（近战、近战、桥打哨兵、前哨站打英雄、能量机关，超远程吊英雄）每两个线为一种颜色 2. 超级电容剩余容量显示 3. 底盘位置显示

表 4.10：平衡步兵测试结果

### 3.4.7 与完整形态的变化

- 使用动态电容目标值设定
- 优化电容电量闭环算法

## 4.4 算法设计

### 4.4.1 关于 YOLO 目标检测神经网络与装甲板自瞄算法的说明

#### 4.4.1.1 卷积神经网络和优势

卷积神经网络在计算机视觉领域的应用和升级在这两年尤为激烈。AlexNet 的诞生，重新打开了自 LeNet-5 之后被封存已久计算机视觉神经网络的梦幻之门。不到十年的发展时间，计算机视觉发展处了众多深层细节的领域，从图像分类，行人检测，

到通用目标检测，语义分割，关键点检测等。举世闻名的 Resnet 系列神经网络，YOLO 系列目标检测网络层出不穷。

在装甲板自瞄程序中，我们采用了“黑箱”形式的神经网络-YOLO 目标检测系列。仅需要输入原始图像，而不需要经过任何人工的操作，即可由训练好的神经网络自动进行目标检测并标注出位置，实现目标跟踪。该方案至少节省了以下几个步骤：颜色处理，像素处理，ROI 筛选，装甲板 ID 确认，灯条拟合等。使用卷积神经网络和目标检测模型，不仅能大幅减少算法开发人员的负担，还能提升识别准确率，为操作手操作提供更准确和有效的帮助。

#### 4.4.1.2 YOLO 的介绍

- ◆ 目标检测传奇系列---You Only Look Once

YOLO 算法[16][17]，作为计算机视觉中的一大亮点，为计算机视觉的研发和应用做出了巨大的贡献。YOLO 系列虽还未发布对应的官方技术报告和技术论文，但已经被广大视觉工程师，计算机视觉爱好者使用。

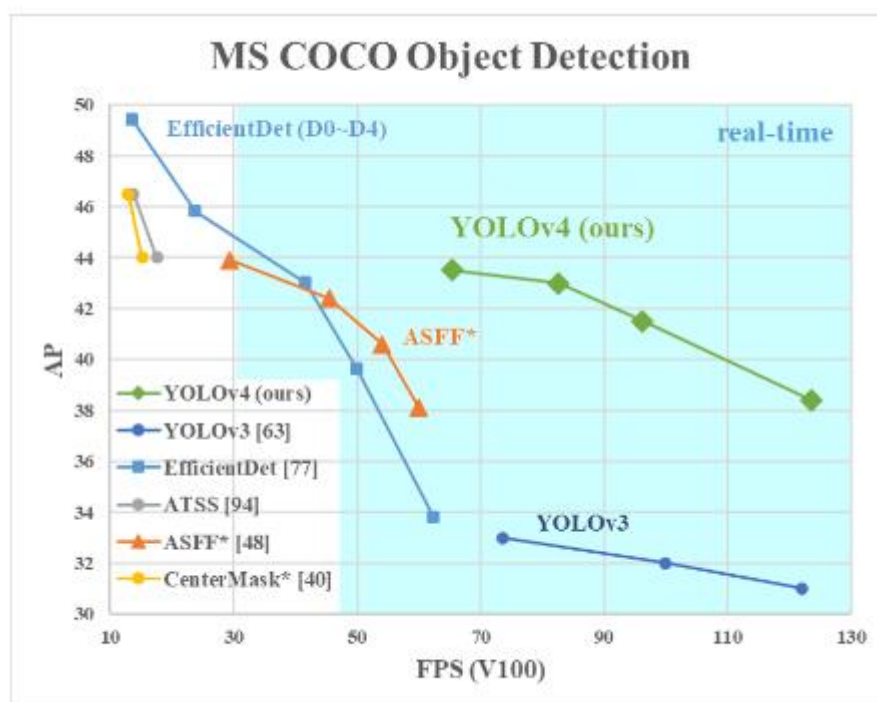


图 4.65: YOLO 系列 (v3 和 v4) 与于当时主流神经检测和追踪网络的效率对比

YOLOv4 继承了前作 YOLOv3 中“快”的优势，还大幅度提升了准度。其相对于其他神经网络有着极大的速度优势，使得我们团队在实际对战中的实时目标检测部署 YOLO 系列成为首选。

#### ◆ Batch Normalization —— 批标准化

批标准化和普通的数据标准化类似，是将分散的数据统一的一种做法，也是优化神经网络的一种方法。与传统数据标准化不同，批标准化将每层的每一个神经元标准化，对每一个神经元进行训练，而不是对整层进行标准化。一般情况下，在神经网络进行卷积或池化操作后，且使用激活函数前，对每个数据输出进行初始化。在下列公式中，前三行是对所有输入的 Batch 进行数据归一化。而在最后一行引入  $\gamma$  和  $\beta$ ，实现批标准化。

$$\begin{aligned}
 &\textbf{Input:} \text{ Values of } x \text{ over a mini-batch: } \mathcal{B} = \{x_1 \dots x_m\}; \\
 &\quad \text{Parameters to be learned: } \gamma, \beta \\
 &\textbf{Output:} \{y_i = \text{BN}_{\gamma, \beta}(x_i)\} \\
 &\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean} \\
 &\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance} \\
 &\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize} \\
 &y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}
 \end{aligned}$$

图 4.66：批标准化公式

通过批标准化，能够有效的提升收敛，降低正则化需要，防止过拟合，还能去掉 dropout 层。目前，批标准化已经成为了主流视觉卷积神经网络不可缺少的工具。

#### ◆ 自适应 Anchor 和分组卷积

YOLO 系列采用了分组卷积这一种提升检测速度的方案。通过不同尺度的卷积核，有效的获取了图片的浅层信息（如颜色等）和深层次的抽象信息。再将这心信息进行叠加，提升对图像识别的准确度。YOLO 没有采用与 F-RCNN [18] 类似的滑动式选择框来提取某一块的区域，因此对背景能力的处理上远强于 F-RCNN。与 F-RCNN 等使用传统的人工手动标记 Anchor 大小不同。在 YOLOv5 中，使用了自适应计算 Anchor 的流程。在 YOLOv5 算法中，针对不同的数据集，一般会预先设定

固定的 Anchor。同样的，在我们的神经网络训练前，我们也采用了该方案。下图展示了 YOLOv5 在 COCO 数据集上的初始设定框。

```
1 anchors:
2   - [10,13, 16,30, 33,23] # P3/8
3   - [30,61, 62,45, 59,119] # P4/16
4   - [116,90, 156,198, 373,326] # P5/32
```

图 4.67：初始设定框

通过预设 anchor 框，我们能够很快的对应上不同卷积尺度大小的图片的特征提取。试下你高效的匹配。但是，在实际部署中，我们不一定能准确的设定一个正确的初始框。

YOLOv5 允许不预设 anchor 而采用自适应的方式进行重新聚类 anchor。在读入数据集之后，将 bbox 从相对坐标改为绝对坐标，并筛选 bbox，保留所有 w 和 h 都大于等于 2 个像素的 bboxes，我们使用了经典的 K-means 聚类方法，得到了 n 个 anchor 并使用遗传算法随机对 anchors 的 w 和 h 进行变导。选择变异后效果更好的复制给 anchors。在这里的变异种，我们设定为 200 次。

#### 4.4.1.3 YOLO 的部署与优化

- ◆ 部署 YOLOv5s 以应用于装甲板识别

与众多视觉爱好者相同，YOLOv5 算法[19]对我们团队的视觉组成员有较大启发。YOLOv5 的官方开源代码中，共有 5 个版本，其性能如下图所示。

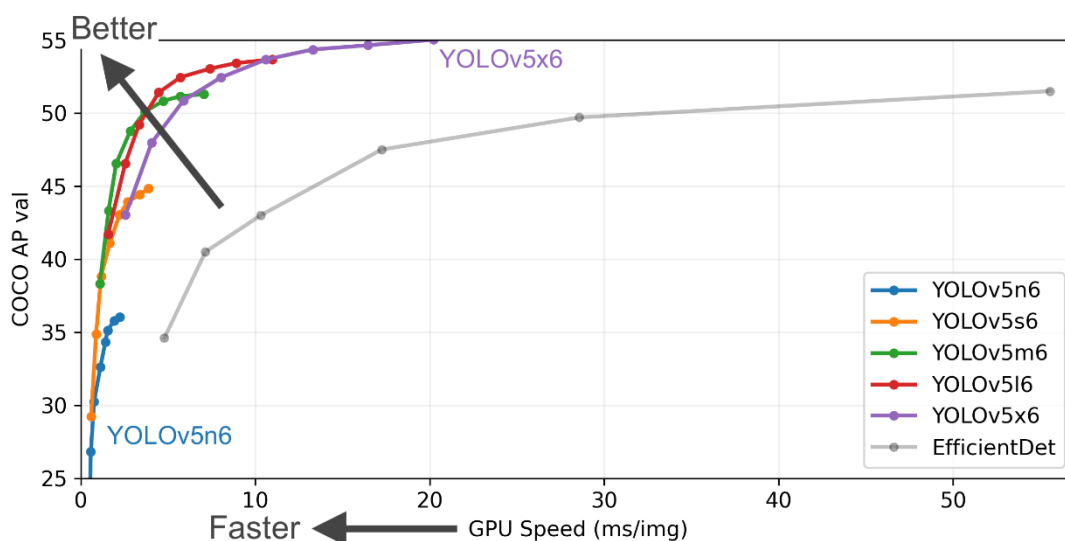


图 4.68：YOLOv5 中不同参数数量的神经网络处理速度和准确度

由图可知，YOLOv5 中的 n 和 s 版本拥有更小的运算量和更快的速度。虽然其正确率相对于参数量更大的 x 版本损失了不少的 mAP，但在算力条件受限的异构机器人上，保证能够正常进行实时识别是首要任务。因此，运算量较小且 mAP 较高的 YOLOv5s 成为了我们的首选。在接下来的内容中，我们将详细讨论我们是如何将 YOLOv5s 训练并部署至我们的机器人上的。

#### ◆ 数据集制作

我们团队按照 VOC 格式，通过往届竞赛视频的抓取处理，制作了包含 2000 张图片的数据集。其分辨率均为 YOLOv5 默认输入大小：640\*480 像素。训练集，验证集，测试集的比例为 7:2:1. 共使用了 23 个标签，并对图像进行手动标注。虽然目前自动标注技术已经出现，但我们认为，在小量级的数据集标注上，使用手动标注更能保证标注的准确度和精度。

#### ◆ 数据集加强

以下两个理由解释了我们为什么要进行数据加强：

- 原始图像数量太少。这使得深度学习网络很难以一个较大的 batch-size 进行训练，容易出现波动。通过数据增强，能有效减少这些问题。
- 数据增强，能够在拥有较多数据的数据集下，比较不同的神经网络和特征提取部分的速度，mAP，和一些相关阈值。更加容易找到神经网络训练中的问题。

我们队伍通过对 hyp.scratch.yaml 文件进行了部分的调试，例如调整了图像增强中的角度，图像 X,Y 轴坐标反转，马赛克等操作，提升了训练集的质量。通过查看混淆矩阵，使用了图像增强算法训练后的神经网络有更高的正确率。

#### ◆ 更换骨干网络

虽然 YOLOv5s 已经拥有较高的速度。但是其仍然不能在我们机器人的算力上实现稳定的高帧率实时监测。这使得我们不得不改进自己的算法。我们发现，YOLOv5 所采用的特征提取器存在很大的优化空间，我们决定将其更换为目前主流的轻量化神经网络代表—ShuffleNetV2 [20]，下面的内容将呈现我们使用 ShuffleV2 作为骨干网络的原因。

首先，我们知道轻量化网络的四大原则[20]可以概括为：

- 原则 1：输入输出的通道数相同的时候，内存访问量是最小的。
- 原则 2：分组卷积的组数太多不利于内存读取。
- 原则 3：碎片化操作对并行加速并不友好。
- 原则 4：逐个元素操作带来的内存和耗时不可忽略。

YOLO 的每一个版本都在之前的版本上对骨干网络有所改进，例如，YOLOv4 [15]使用的是以 CSPD-DarkNet53 的骨干网络进行分类，这种骨干网络的效率相较于经典的 VGG 等特征提取网络已经有了极大的效率提升。但我们发现，ShuffleNetV2 在效率方面更胜一筹。

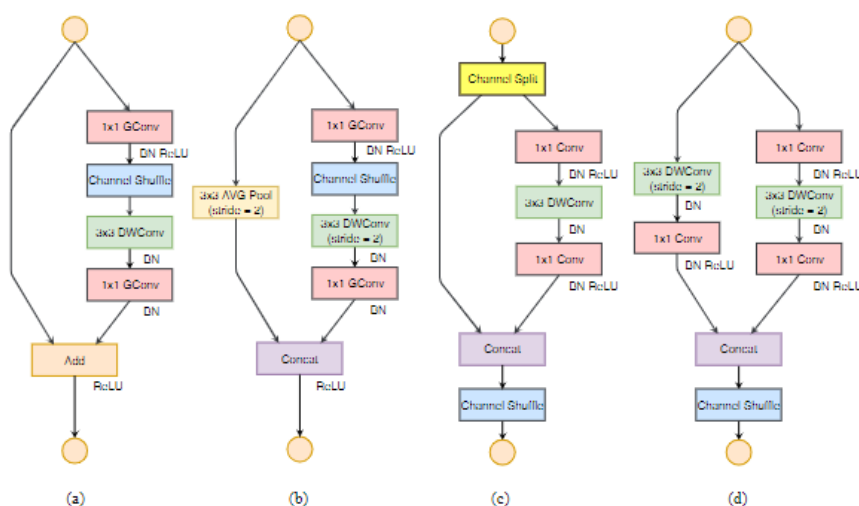


图 4.69：（a）图和（c）图分别展示了 shufflenetV1 和 shuffleNetV2 的骨干架构，（b）和（d）图分别展现了 ShuffleNetV1 和 ShuffleNetV2 的样结构

在 ShuffleV2 之前，所有的特征提取网络都不能很好的切合上述提到的四点原则。例如，例如，ShuffleNetV1、严重依赖于群体争执和瓶颈状构建块，这违反了原则 1 和原则 2，Xception 等基于 Inception 架构的特征提取拥有太多的碎片化操作。而 ShuffleNet 使用的方法则有效的解决这一个问题。ShuffleV2 在尽量不使用 1\*1 卷积的情况下，提升了 channel 的数量，通过 Split 操作将通道分成了两组，实现了部分卷积的功能。采用 concat 替代了 add 操作，仅仅进行叠加操作节省了所有的运算复杂度。

#### ◆ 优化器选择

目前，在使用了 L2 正则优化之后，主流的优化器分别是 SGDWM 优化器和 AdamW 优化器，我们队伍分别使用了两种不同的优化器进行训练。SGDWM 有精度较高的优势，但在硬件限制且样本过于稀疏时会使得整体效率降低的缺点。采用 AdamW 自适应梯度下降，有效地解决了前梯度下降慢的问题，但其损失率过高，无法达到有效的训练。最后，我们采用以动量梯度下值为 0.937 的 SGDMW 进行训练，并加入了 L2=0.2 的正则化。

#### ◆ 测试参数优化

YOLOv5 的原始测试参数虽然在 COCO 数据集上表现完美，但是运用在装甲板识别上还需要进行一定的调整。我们将原始的 YOLOv5 的部分参数进行改动，包括：

- 设定了更高的置信度，以提高准确率
- 设定了更低的 IOU 阈值，由于图像中特征的稀疏特性，该操作能有效的提升 NMS 的作用，同时将一些偏移度较高的图像进行识别而不被误判成背景。
- 将“--augment”形参修改为 True，有效提升了正确率

#### 4.4.1.4 测试网络

通过训练好的神经网络实现对测试集图像进行检测，取得了良好的成就。在测试过程中，视频读取数稳定在 70 帧，误差在 5 帧范围之内。充分展现了 YOLOv5 在部署到实际作战中的高效，准确的检测能力。



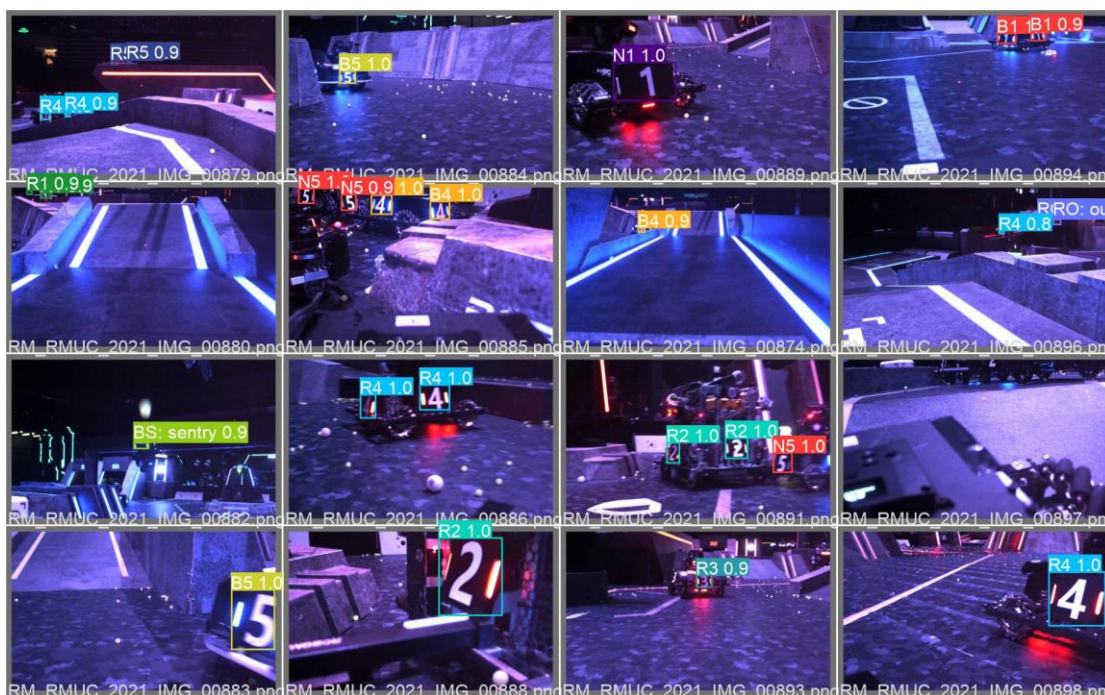


图 4.70: 使用训练好的神经网络对测试图片进行目标检测

#### 4.4.1.5 迭代更新部分

- ◆ 将通用视觉识别模型更换为 Yolov7-Tiny 模型

首先还是效率。当前，通用目标检测模型正在朝着移动平台发展，使得其模型能在算力受限的移动设备上流畅的运行和推理。目前，Yolov7 已经击败 YoloV5 成为了最快的通用推理框架。其次，Yolov7 所使用的工程推理代码，数据集要求格式均可以与 YoloV5 兼容，这极大方便了我们团队更换结构。

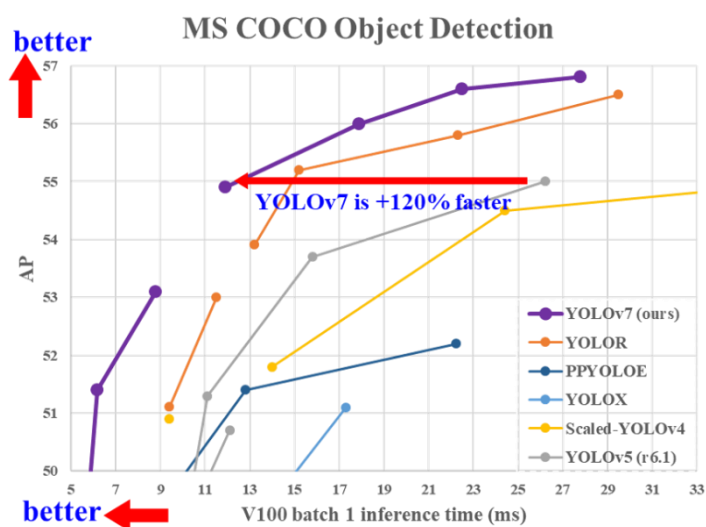


图 4.71: Yolov7 与其他 Yolo 的比较

在此基础上，我们团队尝试将 yolov7 的 width 和 height 进行了适当的减少，发现，在缩小了网络深度和宽度后，其性能仍然能够满足我们的需求。

于此同时，我们团队尝试将训练图像和测试图像的大小修改为了 480\*480，在获得新的聚类参数之后进行训练，在推理时候的速度获得有效的提升。

- ◆ 数据集扩充和平衡
  - 提升了数据集的数量

一个庞大的数据集是深度学习能够有效实现功能的重要要求。我们团队在原先数据集的基础上，将其数量扩充至原先的 200%。重点添加了本次赛季经常出现但原来数据集中没有的 3 4 5 号装甲板。我们尽可能调整使得每个类型的数据数量相似，同事配合 Yolov7 的标签权重分配算法，提升训练后的网络效果。

- 提升了数据集的质量

在原先的数据集中，我们团队参考上海交通大学的数据集进行训练，但是在实际使用的相机和曝光中效果并不理想。因此，我们团队在训练基地使用我们的相机拍摄了更多数据集，在原来的基础上增加了分辨率，曝光时长，曝光增益的调整。同时，我们参考了四川大学 2022 年开源内录视频。

- 减少了无用的分类

分类的数量影响了神经网络的推理速度和准确率，我们团队在确定了操作手的需求后，删除了基地和未亮灯装甲板的类别，并提高了置信度阈值，能够以更快的效率推理识别。

- ◆ 将神经网络部署至能量机关识别

在以往，使用图像阈值分割，寻找特征的方式寻找能量机关待打击部位，虽然速度较快，但其受到的干扰较为严重，如受到装甲板，灯光，曝光调整的干扰，鲁棒性较低。因此，在参考了沈阳航空航天大学的视觉代码[21]之后，我们团队也将 yoloX 关键点检测模型（五点模型）部署于能量机关识别，经过论证发现，在其速度性能损失在可以接受的范围内，其准确性和鲁棒性实现了质的突破。

## 4.4.2 关于能量机关辅助瞄准算法说明

### 4.4.2.1 能量机关自瞄辅助算法流程图

下图展现了由我们团队设计的关于能量机关从识别到跟踪自瞄的大致思维流程图。

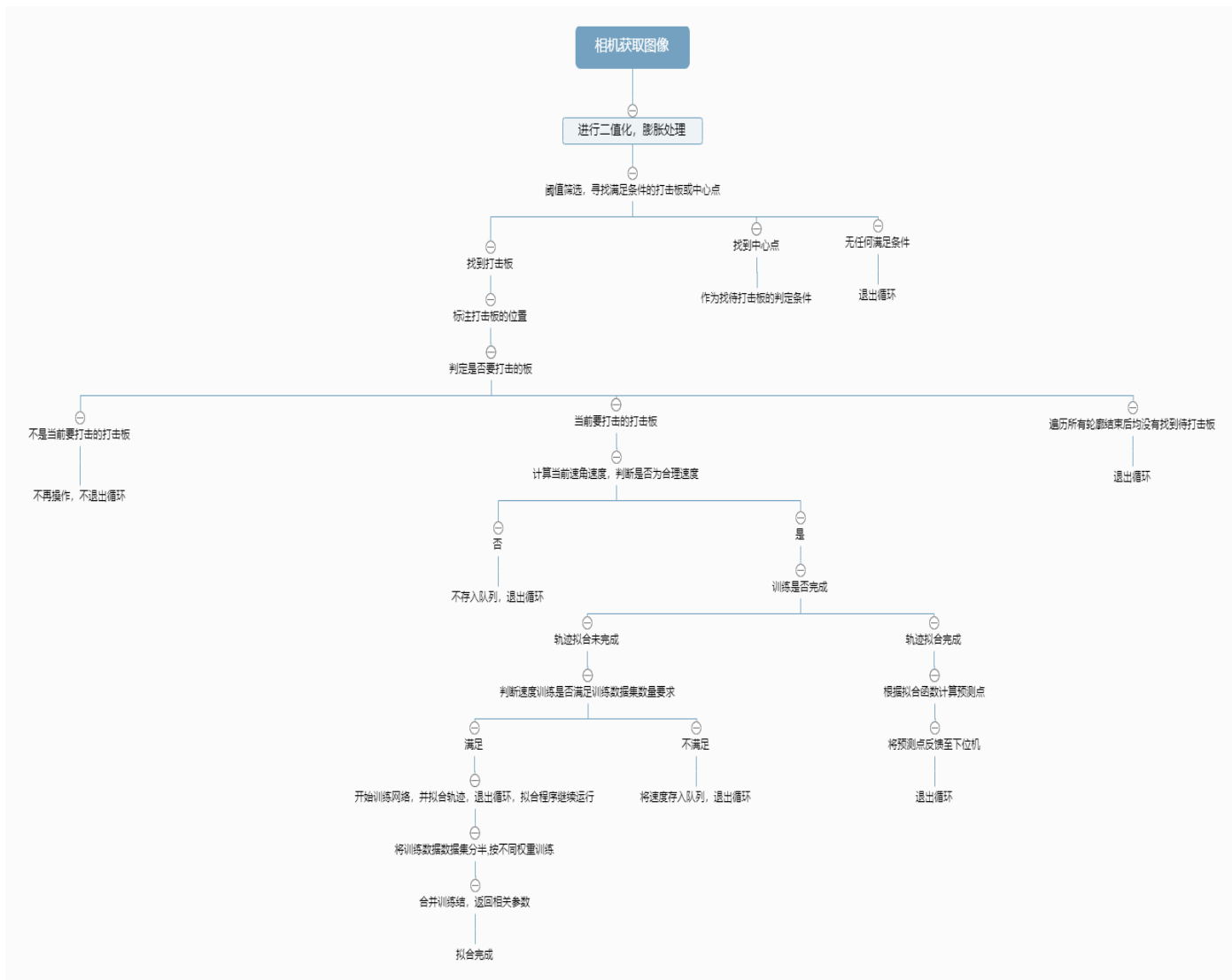


图 4.72: 我们团队使用的能量机关自瞄辅助算法流程图

### 4.4.2.2 能量机关的色彩分割和形态学处理

能量机关的位置设置特殊，机器人面向能量机关时候，摄像头与地面均有较大的仰角，且机器人机打能量机关时，无需完成其他任务。使得我们设置阈值可以较为宽松，提升代码的鲁棒性。

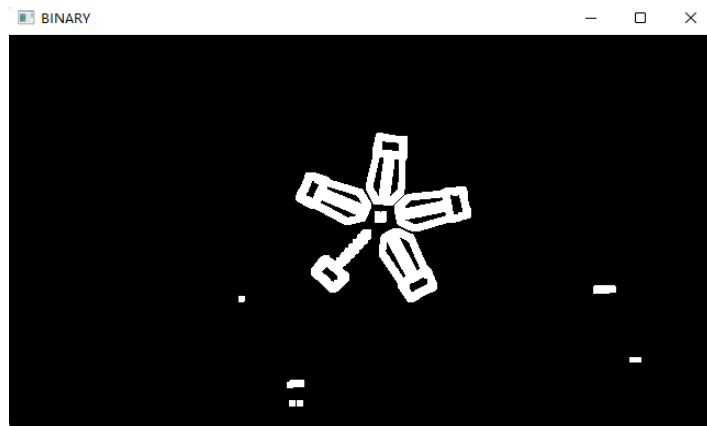


图 4.73: 大能量机关经过二值化和膨胀之后的效果图

能量机关共有两种颜色，分别是蓝色和红色，但经过研究和测试，在实际赛事相机内录视屏中，绿通道也能辅助进行分割，因此我们团队采用三通道（RGB）进行色彩阈值分割。

- 对于蓝色能量机关，我们使用的范围为 B:[170,255], G [20,255], R: [0,120]
- 对于红色能量机关，我们使用的范围为 B:[150,255], G [150,255], R: [210,255]

由于在不同场地下，能量机关的颜色会略有不同。我们使用了往届来自不同赛区不同场地的能量机关视频进行识别，均能实现较好的运行。

随后，我们将上述满足范围内的颜色区域均变为白色，不满足上述范围内的颜色均变为黑色，实现将原图转换为二值图。

在二值图像中使用最普遍的算子是形态学算子[22]，此算子可以改变二值图像中物体的形状。为了实现此类算子，我们首先用一个而知的结构元素与而知图像进行卷积，根据卷积结果的阈值选择二值的输出结果。我们队伍采用 5\*5 的卷积核对获取到的二值图像进行膨胀。采用的是如下的标准膨胀公式，在下列公式中，令，用一个 5\*5 的结构元素  $s$  卷积一个二值图像  $f$  后，得到整数  $C$

$$dilate(f, s) = \theta(c, 1)$$

$$c = f \otimes s$$

通过膨胀运算，使得物体扩张。我们的团队曾是通过开运算来平滑图像，但发现效果不佳，便舍弃了这个操作。

#### 4.4.2.3 轮廓寻找和筛选

边缘检测是计算机视觉中常见的操作，边缘检测在二值图上的应用也较容易实现。我们的团队通过边缘检测，并对每一个识别的闭合边缘进行检测，并通过一系列指标分析该边缘是否是装甲板，我们通过两层筛查，在第一层筛查中，我们筛出所有的装甲板（无论是否是当前要打击的）和能量机关中心点位置。第二层筛查中，我们将通过椭圆离心率和椭圆的位置关系来判断改装甲板是否为要机打的装甲板。

我们队伍发现，通过膨胀之后的图像，R 字图标与实心圆形相似，而且能被边缘检测正确识别出来。我们假定某一轮廓的中心点为 $(x, y)$ ，填充椭圆后的离心率设为 $e$ ，当前图像的宽为 $w$ ，高为 $h$ ，轮廓的面积为 $s$ ，若该轮廓满足以下条件：

$$0 \leq e \leq 0.1$$

$$0.2w \leq x \leq 0.8w$$

$$0.2h \leq y \leq 0.8h$$

$$50pt \leq s \leq 150pt$$

则将该轮廓识别为圆心。

对于打击板识别，我们采用类似圆心的方案进行识别，设当前轮廓的面积为 $s$ ，当前椭圆的长轴为 $a$ ，短轴为 $b$ ，若该轮廓满足以下条件， $l$ 为其轮廓点距离被判定为圆心的中心点的距离

$$200pt \leq s \leq 600pt$$

$$1.3 \leq \frac{a}{b} \leq 2.6$$

$$40pt \leq l \leq 90pt$$

则将该轮廓识别为装甲板。

对于第二轮的筛选，也是利用椭圆的离心率进行筛选，由于带打击装甲板和已打击装甲板存在很大的差异，椭圆的离心率也有很大的差别，这使得分辨是否为带打击装甲板变的容易，如下图所示。

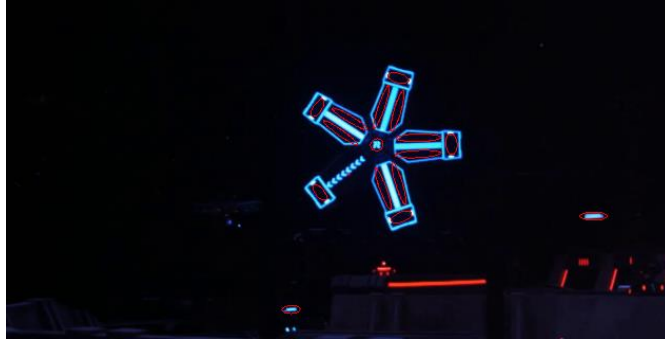


图 4.74：通过填充椭圆并验证椭圆离心率以筛选是否为待打击能量机关的装甲板

很明显，即将被打击的装甲板并不存在两个离心率较大的椭圆，寻找每个离心率较大的椭圆距离被判定为打击板的位置，即可区分出哪一个装甲为即将打击的打击板。

#### 4.4.2.4 能量机关当前待打击的装甲板定位，解算

单目相机三维姿态解算可以大致概括为以下过程：首先，通过相机标定得出相机的内参数，然后利用已知物体在三维空间的坐标，在图像上一一对应的图像像素坐标以及相机内参数进而求解出此时相机相对于空间已知物体的外参数，即旋转向量以及平移向量。最后对旋转向量进行数据分析处理，求解出此时相机相对于已知物体空间坐标的三维姿态欧氏角，即俯仰角，偏航角，滚轮角。

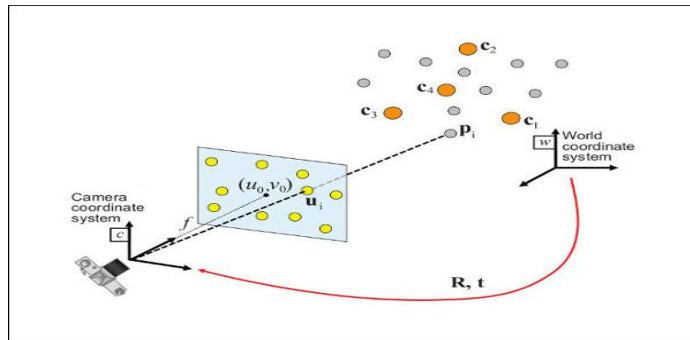


图 4.75：N 点透视位姿求解示意图

对于 N 点透视求解的核心计算公式如下：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

在经过前文的筛选环节后，我们能够确定当前待打击的打击板位置，我们将其进行标记并传入函数进行 PNP 解算。返回的结果如下图所示，其中圆心已经被正确标注并用绿色圆圈覆盖，白色点代表即将打击装甲板的当前打击板中心位置。红色，蓝色，绿色的线段分别代表当前点经过解算的 X,Y,Z 轴。

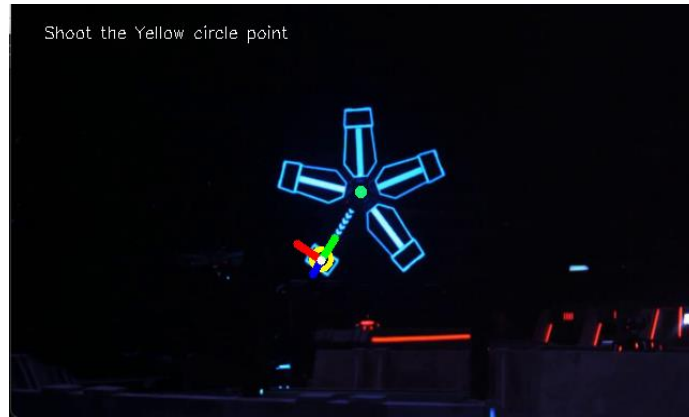


图 4.76：被正确识别的待打击装甲板的坐标和解算坐标轴

#### 4.4.2.5 通过正弦回归预测角度

2022 赛季中，小能量机关做匀速圆周运动。其中，角速度为10RPM(r/min)，大能量机关做变速圆周运动，其角速度将依照按给定的三角函数呈周期性变化，速度单位为rad/s，时间单位 $t$ 的初始值为0，大能量机关的角速度满足以下公式：

$$spd = a \sin(\omega t) + b$$

$$b = 2.090 - a$$

$$a \in [0.780, 1.045]$$

$$\omega \in [1.884, 2]$$

下文中，我们的团队使用能正弦回归来预测能量机关的运动轨迹和速度轨迹，并反馈能量机关的预测位置。

#### 线性回归的相关基础知识

线性回归[23][24]是机器学习中的重要内容之一。线性回归通过拟合一个线性模型，实现准确预测输出的标记。令一个数据集 $D = \{(x^{(i)}, y^{(i)})\} (i = 1, 2, \dots, m)$ 且该数据集有且仅有一个属性值。线性回归通过对数据样本进行学习，拟合出函数 $h(x^i)$ ，使得 $h(x^{(i)}) = \omega x + b \approx y^{(i)}$ 。通过不断训练和拟合，确定两个参数 $\omega$ 和 $b$ 的最优值。通过引入均方差误差损失函数来衡量 $h(x^{(i)})$ 和 $y^{(i)}$ 的相似度，其公式如下



$$J(\omega, b) = \frac{1}{2m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2$$

通过寻找使得 $J(\omega, b)$ 最小的 $\omega$ 和 $b$ ，即为最优值。

在拟合过程中，我们需要引入梯度下降函数，以便控制学习率，贴近最佳 $\omega$ 和 $b$ 。

梯度下降法公式如下：

$$\omega_j := \omega_j - \alpha \frac{\partial J(\omega, b)}{\partial \omega_j}$$

其中 $\alpha$ 为学习率，其控制更新参数 $\omega_j$ 的幅度。 $\frac{\partial J(\omega, b)}{\partial \omega_j}$ 为对损失函数求偏导。对于一次函数，其损失函数应是二次函数。下图展示了一个一次函数损失函数进行梯度下降的过程(设截距为 0)。

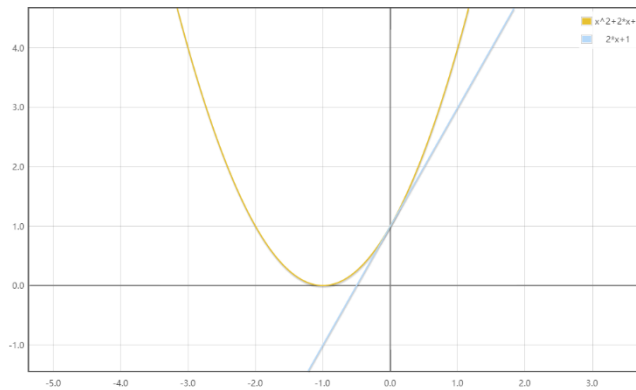


图 4.77：一次函数的二次损失函数示例

对其损失函数做切线，若切线斜率为正。则 $\frac{\partial J(\omega)}{\partial \omega_j}$ 为正，此时有

$$\omega_j := \omega_j - \alpha \frac{\partial J(\omega)}{\partial \omega_j}$$

其中 $\alpha > 0$ ,  $\frac{\partial J(\omega)}{\partial \omega_j} > 0$ ,  $\alpha \frac{\partial J(\omega)}{\partial \omega_j} > 0$ 。经过该轮迭代， $\omega_j$ 将减少， $J(\omega)$ 也随之减小。经过多轮迭代， $\omega$ 将不断逼近该函数的最小值， $J(\omega)$ 在 $\omega$ 的切线斜率随着 $\omega$ 的减小而减小，此时 $\frac{\partial J(\omega)}{\partial \omega_j}$ 也减小。当 $J(\omega)$ 取得最小值时， $\omega$ 即为最小值并返回结果。

◆ 在非线性回归上使用梯度下降法

若实值函数 $F(x)$ 在点 $a$ 处可微且有定义，那么函数 $F(x)$ 在 $a$ 点沿着梯度相反的方向 $-\nabla F(a)$ 下降最快，设 $b = a - \gamma \nabla F(a)$ ，对于 $\gamma > 0$ 为一个够小数值成立，那么， $F(a) > F(b)$ [25]。非线性梯度下降法利用上述方法进行实现。这使得拟合三角函数可以实现。我们的团队使用梯度下降法拟合能量机关速度。在我们的算法代码中，给定初始的 $a$ 和 $\omega$ 为 $a=0.990$ ， $\omega = 1.990$ 。 $\varphi$ 在 $[-\frac{\pi}{2}, \frac{\pi}{2}]$ 区间中的一个随机值。在完成初始化之后，更新四个函数，即预测函数 `predict()`，计算损失函数的函数 `loss()`，微分函数 `defer()`，更新函数 `update()`。

- 1. `predict()`函数的实现，即计算 $y = A \sin(\omega x + \varphi) + B$ 并储存。
- 2. `loss()`函数的实现可以通过以下公式表达

$$J(\omega_j) = \frac{1}{2m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2$$

其中， $\omega_j$ 为上文提到的四个待拟合参数， $h(x^{(i)})$ 是由`predict()`计算得到的当前状态下的预测值， $y^{(i)}$ 是实际我们通过识别后计算得到的瞬时角速度。

- 3. `defer()`函数的实现

我们团队将求导的结果写入函数，其公式可以表示为如下，

$$\frac{d}{dA} (y_i - A \sin(\omega x + \varphi) - B)^2 = 2(A \sin(\omega x + \varphi) + B - y_i) \sin(\omega x + \varphi)$$

$$\frac{d}{d\omega} (y_i - A \sin(\omega x + \varphi) - B)^2 = 2Ax(A \sin(\omega x + \varphi) + B - y_i) \cos(\omega x + \varphi)$$

$$\frac{d}{d\varphi} (y_i - A \sin(\omega x + \varphi) - B)^2 = 2A(A \sin(\omega x + \varphi) + B - y_i) \cos(\omega x + \varphi)$$

$$\frac{d}{dB} (y_i - A \sin(\omega x + \varphi) - B)^2 = 2(A \sin(\omega x + \varphi) + B - y_i)$$

- 4. `update()`函数实现的公式如下

$$\omega_j := \omega_j - \alpha \frac{\partial J(\omega_j)}{\partial \omega_j}$$

通过不断执行该四个函数，实现梯度下降并拟合。当结果精度误差值小于 $\pm 0.001$ 时，退出循环，返回拟合后的能量机关轨迹，下图展示了通过获取连续帧数能量机关运动图像并进行拟合后返回的三角函数的可视化图像。

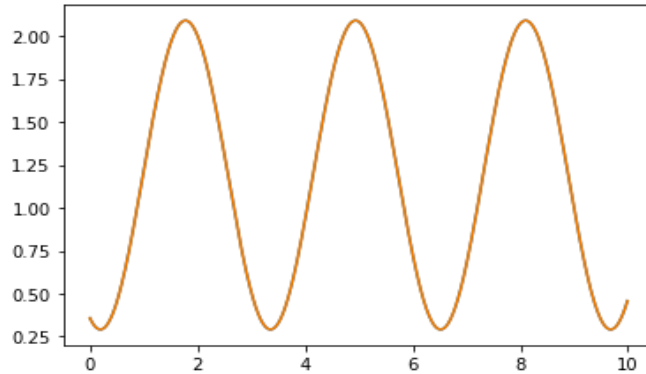


图 4.78: 通过拟合后的可视化三角函数图像

#### 4.4.2.6 完善并部署正弦回归系统

在真正的实战中，速度是我们首先考虑的因素。因此，我们设定一个参数 `length`，在读取帧的数量未达到 `length` 时候，只读取速度，不进行训练，为识别提供更多的算力。当读取帧的数量超过 `length` 帧时便开始训练。通过设定相关的阈值剔除打击板突变带来的数据异常，对于读入的第  $i$  帧，可令  $X[i] = 0.01i$ 。对每一帧的训练为 `cnt0` 次，当训练轮次达到了 `cnt1` 次的时候，完成训练。通过不断优化 `cnt0` 和 `cnt1` 的值，提升效率。由于后期的单次训练会影响之后读取和训练运算的效率，我们的团队将数据集切割成两份，前半份训练  $\frac{3}{5}cnt0$  次，后半份  $\frac{2}{5}cnt$  次。

无论是前面提到的方案还是后面的优化方案，`length0` 长度必须足够，即能涵盖超过一个周期，否则很容易出现梯度爆炸或者落到局部最优上，在保证 `length0` 足够的情况下，可以进行适量的压缩提高效率并节省内存。

#### 4.4.3 弹丸发射弹道补偿

弹道补偿[26]，通过计算实时识别和实际打击间的误差，实现更精准的打击。设我们的摄像头直对目标，令子弹发射的初速度为  $v_0$ ，仰角为  $\theta$ ，发射点与打击目标的水平距离为  $x$ ，竖直距离为  $h$ ，显然， $\tan\theta = \frac{h}{x}$ 。仅考虑重力和空气阻力下，令重力加速度为  $g$ ，空气阻力为  $F = kv^2$ ，设弹丸的质量为  $m$ ，则有以下方程：

$$kv^2 = ma$$

其中  $a = \frac{dv}{dt}$

解该方程可得

$$T = \frac{e^{\frac{kx}{m}} - 1}{\frac{k}{m} v_0 \cos\theta}$$

令发射角为  $\theta$ 。则抛射的状态可以拆分为以下三种情况:

若  $\theta < 0$ ，则可列方程： $mg - kv^2 = ma$ ，其中  $a = \frac{d^2x}{dt^2}$ ，解该方程可得

$$y = -\frac{m \ln\left(1 - v_0 \sin^2\theta \frac{k}{mg}\right)}{2k} - \frac{m \ln(\cosh(\sqrt{\frac{gk}{m}} t - \operatorname{arctanh}(v_0 \sin\theta \sqrt{\frac{k}{mg}})))}{k}$$

若  $\theta \geq 0$ ，则要讨论是否在打击到目标点前， $h$  已经达到最大值。则在竖直分量上，有

$$C = \frac{\operatorname{arctan}(v_0 \sin\theta \sqrt{\frac{k}{mg}})}{\sqrt{\frac{gk}{m}}}$$

若  $C > T$ ，则

$$y = \frac{m}{k} \ln\left(\frac{\cos((C - T)\sqrt{\frac{gk}{m}})}{\cos(C\sqrt{\frac{gk}{m}})}\right)$$

若  $C \leq T$ ，则

$$y = \frac{m}{2k} \ln\left(1 + \frac{k}{mg} v_0^2 \sin^2\theta\right) - \frac{m}{k} \ln(\cosh((C - T)\sqrt{\frac{gk}{m}}))$$

若目标高度为  $h$ ，实际这个角度会打击到的目标为  $y$ ，调整的差  $\Delta = h - y$ （在我们的代码中  $\Delta = dy$ ）。此时，定义  $y\_temp$  参数，初始化为  $y$ ，更新  $y\_temp = y\_temp + \Delta$ ，以调整下一个状态。当进入新的一个状态时更新  $\theta$ ，对于第  $i$  个状态， $\theta_i = \operatorname{arctan}\left(\frac{y\_temp}{x}\right)$ ，当  $\theta_i$  达到最高值，结束迭代。该方案能减少时间上的浪费，在满足精度的条件下减少更多的迭代操作。使用该方案，还能有效降低调参难度。

#### 4.4.4 卡尔曼滤波

卡尔曼滤波是我们在预测部分的方案之一。

上位机接收下位机发送的陀螺仪数据，利用卡尔曼滤波对解算结果进行运动预测。利用解算得到的平移向量和所需的陀螺仪数据得到目标单位相对于陀螺仪的新世界坐标  $X$ 、 $Y$ 、 $Z$ 。将该世界坐标作为当前状态的观测数据传入卡尔曼滤波器进行目标下一状态的位置预测。卡尔曼滤波器的预测及更新方程如下：

- 计算当前状态的先验估计值： $\hat{X}_{k|k-1} = A * \hat{X}_{k-1|k-1}$
- 计算先验误差协方差矩阵： $P_{k|k-1} = A * P_{k-1|k-1} * A^T + Q$
- 计算卡尔曼增益： $K_k = P_{k|k-1} * H^T * (H * P_{k|k-1} * H^T + R)^{-1}$
- 更新当前状态的后验估计值： $\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k * (Z_k - H * \hat{X}_{k|k-1})$
- 更新后验误差协方差矩阵： $P_{k|k} = (I - K_k * H) * P_{k|k-1}$

其中， $Q$ 、 $R$  等参数及其他初始值利用实验确定并修正。

再此基础之上，为了达到解藕观测变量，消除变量间的相关性，使用 EKF 代替 KF 进行进一步测试。

#### 4.4.5 相机与 imu 的联合标定

基于弹道预测的需要，我们团队需要及时获取 imu 与相机之间的转换参数，我们团队通过采用 ros 系统下的 kalibr 标注工具，实现了相机与 imu 的联合标定，在之后的预测击打中实现了准确率的有效提升。下图展示了联合标定的结果，对应的数据通过 txt 文件进行了保存。

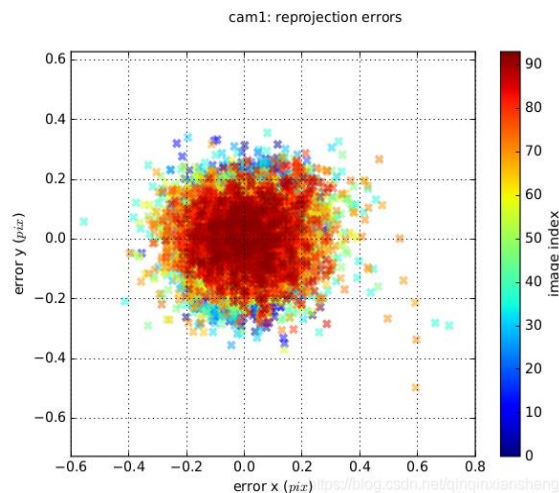


图 4.79：联合标定结果

#### ◆ 标定环境修改

相机的内参和畸变是一个定值。但是在测试这个定值的时候，时常会因为场地灯光等因素影响实际测量的数据。因此，我们团队提前在模拟场地，使用已经安装在车上调整好的摄像头和 imu 进行标定，经过对比，相较于在实验室进行标定，使用场地标注的相机内参得到的预测点更加准确。解决了之前在实验室出现的部分车辆预测点位不准的问题。

### 4.4.6 四阶龙格库塔法解微分方程

#### 4.4.6.1 原理

在含有空气阻力的弹道模型中，不仅是可以直接暴力积分出一个直接计算出相关数值的式子，我们同样也可以转化成相关的微分方程，然后利用四阶龙格库塔法来解决这样的问题。

龙格-库塔法[28]是用于非线性常微分方程的解的重要一类隐士或显式迭代法。经常被称为“RK4”的四阶法有如下描述：

令初值问题表述如下：

$$y' = f(t, y), y(t_0) = y_0$$

则，对于该问题的 RK4 可以由下面的方程给出：

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

其中：

$$k_1 = f(t_n, y_n)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_n + h, y_n + hk_3)$$

这样的话，下一个值 $y_{n+1}$ 就由现在的值 $y_n$ 加上时间间隔 $h$ 和一个估算的斜率的乘积决定的。该斜率是以下斜率的加权平均：

- $k_1$ 是时间段开始时的斜率
- $k_2$ 是时间段中点的斜率，通过欧拉法采用斜率 $k_1$ 来决定  $y$  在  $t_n + \frac{h}{2}$  的值
- $k_3$ 同样是中点的斜率，但这次采用斜率 $k_2$ 来决定  $y$  值

➤  $k_4$ 是时间段终点的斜率，其  $y$  值用 $k_3$ 决定

当四个斜率取平均的时候，中点的斜率由更大的权值：

$$slope = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

RK4 法是四阶方法，也就是说每步的误差是  $h$  阶，而总共累计的误差同样是  $h$  阶。

这个公式对于标量或者向量函数都同样使用。

#### 4.4.6.2 物理模型建立方程组

我们依旧先寻找我们可以用到的数据：

- 初速度 $v_0$
- 距离和高度 $x, y$
- 重力加速度 $g$ 和空气阻力系数 $k$ ，弹丸质量 $m$

弹丸在飞行中，会受到空气阻力和中立两个力的作用，即：

$$F_g = mg$$

$$f = \frac{1}{2} C \rho S v^2$$

重力方向竖直向下，空气阻力方向相反，由牛顿第二定律得：

$$a_g = g$$

$$a_f = \frac{\frac{1}{2} C \rho S v^2}{m}$$

在相同种类的弹丸中 $\frac{\frac{1}{2} C \rho S}{m}$ 几乎相同，直接用  $k$  代替，则有

$$a = kv^2$$

依照之前所说的物理情景，列出相应方程：

$$v = \sqrt{u^2 + w^2}$$

$$\frac{dy}{dt} = w$$

$$\frac{dx}{dt} = u$$



$$\frac{du}{dt} = -kv^2 \cos\theta$$

$$\frac{dw}{dt} = -kv^2 \sin\theta - g$$

可以化简得到

$$\frac{du}{dt} = -kuv$$

$$\frac{dw}{dt} = -kvw - g$$

式子中的 $u, w, x, y$ 是关于自变量 $t$ 的微分方程组，即自然坐标系的弹丸质心运动微分方程组，那么上面的参数确定，弹道的迭代结果也唯一确定。所以当我们知道了 $t$ 的时候便可求解。

优化：这里的 $t$ 作为自变量且没有计算出解析解的时候，在该模型下不好处理 $t$ 的准确值，所以可能会造成误差。如果以 $x$ 作为自变量，在目标位置已经确定下，该值已经确定。

在自然坐标系下有 $\frac{dy}{dx} = p$ ， $p$ 为目前弹丸质心的运动方向。 $\lim_{t \rightarrow 0^+} p = \frac{u_0}{v_0} = \tan\theta_0$ 则可以优化的方程如下：

$$\frac{du}{dx} = -ku\sqrt{1+p^2}$$

$$\frac{dp}{dx} = \frac{-g}{u^2}$$

$$\frac{dy}{dx} = p$$

化简得：

$$\frac{du}{dx} = -ku$$

$$\frac{dp}{dx} = \frac{-g}{u^2}$$

$$\frac{dy}{dx} = p$$

$$\frac{dt}{dx} = \frac{1}{u}$$

$$v = u\sqrt{1+p^2}$$

在解得下列方程后，对 $\frac{du}{dx}$ 和 $\frac{dp}{dx}$ 进行龙格库塔法，迭代相应的次数，在达到一定的精度后，更新 pitch，即抬枪补偿的过程结束。

#### 4.4.6.3 相应参数的选取

在前面的信息我们知道，需要提前求得的常数有重力加速度  $g$  和  $k$ ，其中， $k = \frac{\frac{1}{2}C\rho S}{m}$ 。

阻力系数小弹丸： $C = 0.47$ ，

阻力系数大弹丸： $C = 0.275$ ，

空气密度： $\rho = \frac{1.169kg}{m^3}$

带入公式得到结果。

#### 4.4.7 Ceres 优化最小二乘法拟合能量机关以及外部逻辑的优化

在拟合速度函数的参数的时候，因为十分依赖于相位，所以实际上在每一次堆入有效速度的时候，需要拟合一次相位，那么应该有的外部逻辑操作是：先堆入一定量的速度放入 deque 中，然后拟合四个参数，知道四个参数拟合出的效果较为正常，最后每得到一个较为正常的风车转速，就拟合一次相位。原有的方案便难以架构。之后更新了利用 Ceres 第三方库进行优化[29]。

##### 4.4.7.1 Ceres 的优化细节

###### ◆ 代价函数的构建

常规的写法是需要构造 CostFunction 结构体用来描述代价函数，并且需要重载括号运算符，构造残差的时候，应该分成两种，第一个结构体的目标是拟合四个参数，第二个结构体是要拟合初相。

$$f_{cost} = \hat{y} - A\sin(\omega\hat{x} + \varphi) - B$$

$$f_{cost} = \hat{y} - \hat{A}\sin(\hat{\omega}\hat{x} + \varphi) - \hat{B}$$

###### ◆ 优化的上下限问题

由于比赛规定的大符参数是在一定范围内取值，可以利用 Ceres::Problem 类中的 SetParameterLowerBound 和 SetParameterUpperBound 两个函数进行限制优化上下限。然而因为场地中大符的参数与实际的风车参数可能会有偏差，即使已在规则中告知参数也需要通过场地实测来改变优化上下限的实际阈值。

#### 4.4.7.2 拟合结果的验证

对于拟合结果是需要验证的，如果拟合的数据与预期差距较大，是不能直接对相应的结果进行预测的。常用的验证手段是 MAE(平均绝对误差)和 RMSE(均方根误差)。其表示如下：

$$MAE = \frac{1}{m} \sum_{i=1}^m |h(x_i) - y_i|$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x_i) - y_i)^2}$$

MAE 是一个线性分数，所有的个体差异在平均值上的权重相等，而 RMSE 会放大对于错误对最终值的影响。而对于该场景中，不能放过对于错误较大的结果进行预测，所以采用 RMSE 进行验证。

#### 4.4.7.3 外部逻辑的优化

这里分成三个主要步骤：

##### ➤ 堆速度阶段

将计算好的风车转速堆入 deque 中，直到达到设定好的阈值 length1。这里如果阈值 length1 的设定值过小的话容易造成梯度爆炸或者落入局部最优解

##### ➤ 拟合核心阶段

length1 已经达到要求，在该次运行中可以使用拟合。然而如果不通过 RMSE 或者没有新的速度堆入，将无法进行下一阶段，直到拟合完成才能进入相位维护阶段；在此它将会不断弹出旧的速度推入新的速度

##### ➤ 相位维护阶段

在  $A, B, \omega$  三个参数已经拟合完成之后，需要每一次运行的时候拟合相位，也需要通过 RMSE 验证。此时 deque 的长度可以低于 length1，加快处理速度

#### 4.4.8 拟合前的数据处理

常规方法是使用帧差法，我们假定已经知道两帧或者几帧之间的装甲板和 R 的相对位置关系又知道采样的时间，我们就可以直接算出来风车的在这一段时间的角速度。

$$\omega = \frac{1}{T} (\theta_2 - \theta_1)$$

在预测的时候我们同样需要判断风车的旋转方向，可以同样以上述的相对位置关系，构建三维向量，然后对这个向量进行叉积，求其正弦值，以其正负判断方向。由于存在系统误差，在计算风车方向上可能会出现计算相反的情况，可以通过取众数法，暂时使用当前的旋转方向进行预测，并存储方向的符号，当这个方向的符号大于错误的符号并且这个数量达到一个值的时候，就可以存储这个方向并沿用到整个过程结束。

常规方法是使用帧差法，我们假定已经知道两帧或者几帧之间的装甲板和 R 的相对位置关系又知道采样的时间，我们就可以直接算出来风车的在这一段时间的角速度。

$$\omega = \frac{1}{T}(\theta_2 - \theta_1)$$

在预测的时候我们同样需要判断风车的旋转方向，可以同样以上述的相对位置关系，构建三维向量，然后对这个向量进行叉积，求其正弦值，以其正负判断方向。由于存在系统误差，在计算风车方向上可能会出现计算相反的情况，可以通过取众数法，暂时使用当前的旋转方向进行预测，并存储方向的符号，当这个方向的符号大于错误的符号并且这个数量达到一个值的时候，就可以存储这个方向并沿用到整个过程结束。

#### 4.4.9 粒子滤波

粒子滤波(PF) [30]，会直接利用一组点经过模型转换后再采样，用此后续数据来近似任意分布。它会直接根据粒子的权重进行融合得到状态的一种可能假设。根据大数定律通过采样近似一个分布，样本越多结果越接近真实分布，实质是一种非参数化的蒙特卡洛方法[31]。

我们定义：

- $x$  为状态
- $y$  为测量到的数据
- $f$  为状态函数
- $h$  为测量函数
- $v, n$  为测量噪声和观测噪声

讨论的情形为：

$$x_k = f_k(x_{k-1}, v_{k-1})$$

$$y_k = h_k(x_k, n_k)$$

从贝叶斯理论的观点来看，状态估计问题就是根据之前一系列已有数据 $y_{1:k}$

#### 4.4.9.1 蒙特卡洛采样

假设我们可以从一个目标概率分布 $p(x)$ 中采样到一系列的样本(粒子) $\{x_n\}$ ，那么就能利用这些样本去估计这个分布的某些函数的期望值。如：

$$E(f(x)) = \int_a^b f(x)p(x)dx$$

$$Var(f(x)) = E\left(f(x) - E(f(x))\right)^2 = \int_a^b \left(f(x) - E(f(x))\right)^2 p(x)dx$$

上面的式子其实是计算期望的问题，只是被积函数不同。蒙特卡洛思想实际上是用平均值来代替积分。求期望：

$$E(f(x)) = \frac{1}{N} \sum_{i=1}^N f(x_i)$$

假设有一粒质地均匀的骰子。规定在一次游戏中，连续四次抛掷骰子，至少出现一次 6 个点朝上就算赢。现在来估计赢的概率。我们用 $x_k^{(n)}$ 来表示再第  $n$  次游戏中，第  $k$  次投掷的结果( $k = 1, 2, 3, 4$ )。对于分布均匀的骰子，每次投掷服从均匀分布，即

$$x_k \sim u(1, 6)$$

这里的区间是取整数，1, 2, 3, 4, 5, 6，代表 6 个面。由于每次投掷都是独立同分布的，所以目标函数  $p(x)$  也是一个均匀分布。

为了估计取胜的概率，在第  $n$  次游戏中定义一个指示函数：

$$f(x^{(n)}) = I\left\{0 < \sum_{k=1}^4 I\{x_k^{(n)} = 6\}\right\}$$

其中 $I(x)$ 为若条件满足为 1，否则为 0。这里的意义就是单次游戏中，若四次投掷中只要有一个 6 朝上， $f(x) = 1$ 。由此就可以估计在这样的游戏中取胜的期望，也就是取胜的概率：

$$\theta = E(f(x)) \approx \frac{1}{N} \sum_{i=1}^N f(x^{(n)})$$

所以当抽样的次数  $N$  足够大的时候，上式便能接近真实的取胜概率了。

#### 4.4.9.2 蒙特卡洛方法应用于粒子滤波

假设可以从后验概率中采样到  $N$  个样本，那么后验概率可以计算成：

$$\hat{p}(x_n|y_{1:k}) = \frac{1}{N} \sum_{i=1}^N \delta(x_n - x_n^{(i)}) \approx p(x_n|y_{1:k})$$

其中，在这个蒙特卡洛方法中，我们定义  $f(x) = \delta(x_n - x_n^{(i)})$ 。 $\delta(x)$  为狄拉克函数。可以替代上面提到的指示函数。

既然用蒙特卡洛方法能够直接估计后验概率，现在要做图像跟踪或者滤波，实际上是要知道当前状态的期望值：

$$E(f(x)) \approx \int f(x_n) \hat{p}(x_n|y_{1:k}) dx_n = \frac{1}{N} \sum_{i=1}^N \int f(x_n) \delta(x_n - x_n^{(i)}) dx_n = \frac{1}{N} \sum_{i=1}^N f(x_n^{(i)})$$

也就是用这些采样的粒子状态值直接平均就得到了期望值，也就是滤波后的值，这里的  $f(x)$  就是每个粒子的状态函数。这就是粒子滤波了，只要从后验概率中采样很多的粒子，用它们的状态求平均就得到了滤波结果。

#### 4.4.9.3 重要性采样和重采样

实际上我们并不知道中后验概率，无法从目标分布中采样，就从一个已知的可以采样的分布取采样，这样上面的期望问题就变成：

$$E(f(x_k)) = \int f(x_k) \frac{W_k(x_k)}{p(y_{1:k})} q(x_k|y_{1:k}) dx_k$$

$$\text{其中, } W_k(x_k) = \frac{p(y_{1:k}|x_k)p(x_k)}{q(x_k|y_{1:k})} \propto \frac{p(x_k|y_{1:k})}{q(x_k|y_{1:k})}$$

上面的期望计算可以通过蒙特卡洛方法来解决它，也就是说通过采样  $N$  个样本  $\{x_k^{(i)}\} \sim q(x_k|y_{1:k})$ ，用样本的平均来求得它们的期望，所以上面的(3)可以近似为

$$E(f(x_k)) \approx \frac{\frac{1}{N} \sum_{i=1}^N W_k(x_k^{(i)}) f(x_k^{(i)})}{\frac{1}{N} \sum_{i=1}^N W_k(x_k^{(i)})} = \sum_{i=1}^N \hat{W}_k(x_k^{(i)}) f(x_k^{(i)})$$

$$\text{其中, } \hat{W}_k(x_k^{(i)}) = \frac{W_k(x_k^{(i)})}{\sum_{i=1}^N W_k(x_k^{(i)})}$$

这个式子，不再是原有的式子直接的相加求平均了，而是一种加权和的一种形式。不同的粒子都有它们相应的权重，如果粒子的权重大，说明信任该粒子比较多。

在应用这个过程之中，存在一个退化问题。经过几次迭代之后，很多粒子的权重已经变得很小，基本可以忽略，只有少数的粒子权重比较大。并且粒子的权值的方差随着时间的增大，状态空间的有效粒子数较少。随着无效采样粒子数目增加，大量机会会花费在几乎不起作用的粒子上，使得估计性能下降。

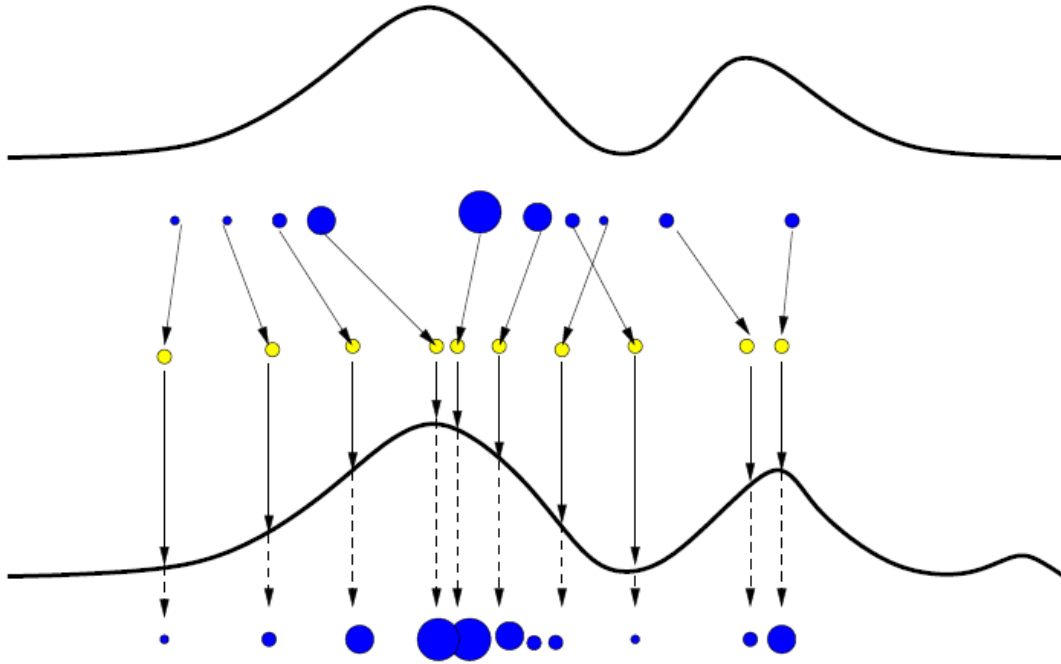


图 4.80：重采样示意图 1

通常采用有效粒子数来衡量粒子权值的退化程度，即：

$$N_{eff} = \frac{N}{1 + Var(w_k^{*(i)})}$$

这个含义是：有效粒子数越小，即权重的方差越大，也就是说权重大和权重小的差距越大，表明权值退化越严重。实际计算中，可以近似为：

$$\hat{N}_{eff} \approx \frac{1}{\sum_{i=1}^N (w_k^{(i)})^2}$$

重采样的思路是：既然那些权重小的不起作用，那就可以丢掉。要保持粒子数目不变，就需要用新的粒子取代。找到新的粒子最简单的方法就是大权重粒子进行复制，复制的各种应该用权重的比例进行分配。

前面已经说明了求某种期望问题变成了这种加权和的形式：



$$p(x_k | y_{1:k}) = \sum_{i=1}^N w_k^{(i)} \delta(x_k - x_k^{(i)})$$

通过重采样后，期望表示成：

$$\tilde{p}(x_k | y_{1:k}) = \sum_{j=1}^N \frac{1}{N} \delta(x_k - x_k^{(j)}) = \sum_{i=1}^N \frac{n_i}{N} \delta(x_k - x_k^{(i)})$$

这里对比一下上下两个式子  $x_k^{(i)}$  是第  $k$  时刻的粒子， $x_k^{(j)}$  是  $k$  时刻重采样以后的粒子。其中  $n_i$  是指粒子在产生新的粒子集  $x_k^{(j)}$  时被复制的次数。等号表明，经过这一轮重采样之后，所有的粒子的权重一样，都是  $\frac{1}{N}$ ，只是有的粒子多出现了  $n_i$  次。

#### 4.4.9.4 标准流程

整个粒子滤波的算法流程为：

- ◆ 粒子集初始化， $k=0$ ：

对于  $i = 1, 2, \dots, N$ ，由先验  $p(x_0)$  生成采样粒子  $\{x_0^{(i)}\}_{i=1}^N$

- ◆ 对于  $k = 1, 2, \dots$ ，循环执行以下步骤：
  - 重要性采样：对于  $i = 1, 2, \dots, N$ ，从重要性概率密度中生成采样粒子，计算粒子权值，并进行归一化；
  - 重采样：对粒子集  $\{x_k^{(i)}, w_k^{(i)}\}$  进行重采样，重采样后的粒子集为  $\{x_k^{(i)}, \frac{1}{N}\}$ ；
  - 输出：计算  $k$  时刻的状态估计值：

$$\hat{x}_k = \sum_{i=1}^N \tilde{x}_k^{(i)} \tilde{w}_k^{(i)}$$

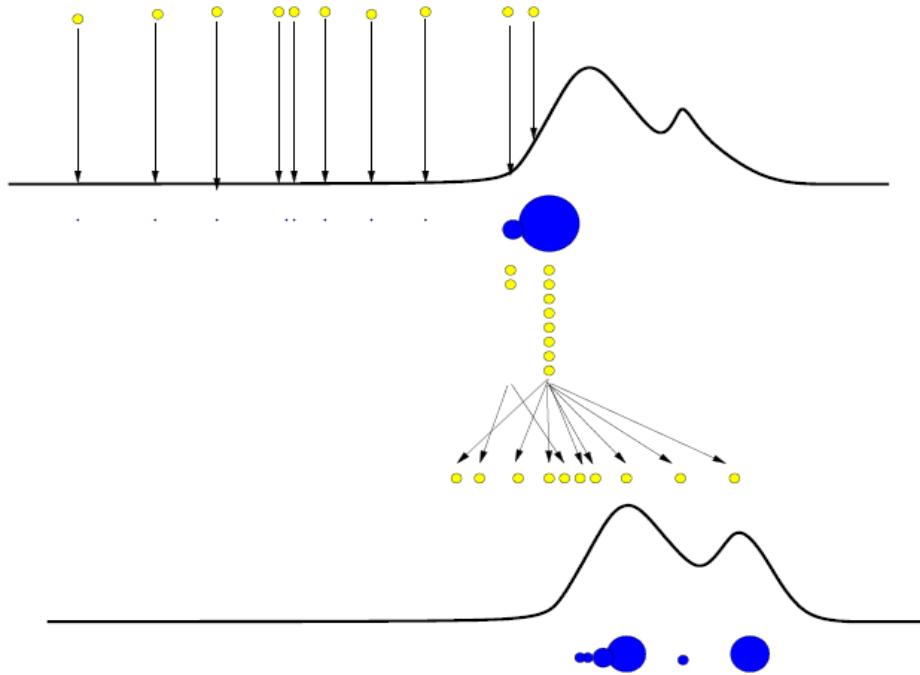


图 4.81：重采样示意图 2

#### 4.4.10 未采用的方案

- ◆ 拟合圆+跳过卡尔曼滤波直接预测能量机关的出现位置

在拟合能量机关打击板位置时，我们曾尝试利用拟合圆和速度预测计算，跳过使用 EKF 拟合，直接预判装甲板接下来出现的位置。

识别时，使用拟合圆标记出轨迹，然后记录下连续两帧之间打击板中心点位移的位置，预测出平均位移的坐标点变化，由于打击板只会在这条拟合圆上的点，既可以直接预测下一次打击板出现在这条轨迹上的点。

我们的团队发现拟合圆部分由于严重于相机和打击板之间画面拍摄角度的相对静止，这在比赛场上是不可能实现的，相机的抖动和发射的后坐力导致的角度偏移，使得拟合圆方案不现实。同时，由于其精度过低，仅仅只根据上一帧和上两帧来进行判断，准确率极低。因此，我们放弃了这个方案。

- ◆ 在进行形态学处理之前使用模板匹配算法寻找“R”标志进行能量机关中心点寻找

我们曾做过不同的 R 模板并搜索全图寻找 R 标志来定位中心点，该方案的正确识别率相对通过离心率判定轮廓和面积筛选较高。但是其算法复杂度太高，需要单独为寻找圆心提供额外的算力，这在算力极其有限的机器人上无法实现高帧率检测。因此我们放弃了这个方案。

- ◆ 采用精度更高的 YOLOv5x, YOLOv5L 等其他 YOLOv5 参数进行神经网络训练

YOLOv5 其他类型的算法在精度上有肉眼提升，但运算效率也大幅度下降。该比赛对识别速度的要求远远高于精度的要求，因此我们更倾向在速度方面优化，包括使用了 ShufflenetV2 网络减少了每张图像输入的参数。尽可能提升识别的速度。

- ◆ 在研发 defer ( ) 函数中，使用导数的定义式进行求导，并设置  $\Delta x = 0.00001$ 。

后发现采用求偏导方式准确率有更大的提升，于是废弃该方案。

- ◆ 队列实时维护算法

我们曾使用过利用一个双端队列 deque，令该队列实时维护的最大长度为 length，则维护的方法便是在队列长度达到 length 前，保持顶端存入，当到达 length 的长度的时候，先进行实时的预测操作，然后再底端弹出一个元素，然后从顶端存入新的元素，保持一个动态平衡。这个方案依旧需要满足 length 超过一个周期，在完成训练之前需要不断实时维护，存入和弹出。且因为存入和弹出不断发生变化，初相会不断发生变化，容易发生梯度爆炸以及卡在局部最优值上，所以舍弃。

- ◆ 牛顿迭代法

对于正弦函数  $y = f(x) = r_0 \sin(r_1 x + r_2) + r_3$ ，有四个未知量  $r_i (i=0,1,2,3)$ ，记其待估计系数向量为  $\beta = [r_0, r_1, r_2, r_3]^T$ ，在此洗漱下，记  $y = f(x; \beta)$ 。

然后我们有已知的  $n$  个点  $(x_i, y_i) (i = 0, 1 \dots, n - 1)$  [构造方式相同]，要使用上面的点集拟合函数  $f(x)$ ，则需要  $S$ ，即残差平方和来构造，具体公式如下：

$$S = \sum_{i=0}^{n-1} (f(x_i; \beta) - y_i)^2$$

也就是使得  $\frac{\partial S}{\partial r_j} = 0 (j = 0, 1, 2, 3)$  最小。

在该条件下，记偏微分方程组求解的记为  $g_j(\beta)$ ，令  $g_j(\beta) = 0$ ，该非线性方程组可以用高斯牛顿法解决。记向量函数  $F(\beta) = [g_0(\beta), g_1(\beta), g_2(\beta), g_3(\beta)]^T$ ，以及雅可比矩阵：

$$F'(\beta) = \begin{bmatrix} \frac{\partial g_0(\beta)}{\partial r_0} & \dots & \frac{\partial g_0(\beta)}{\partial r_3} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_3(\beta)}{\partial r_0} & \dots & \frac{\partial g_3(\beta)}{\partial r_3} \end{bmatrix}$$

对某个系数向量近似解  $\beta_{k+1}$ ，对向量函数  $F(\beta_{k+1})$  做一阶 Taylor 展开，可以得到：

$$F(\beta_k) + F'(\beta_k)(\beta_{k+1} - \beta_k) \approx F(\beta_{k+1}) = 0$$

通过牛顿迭代公式[27]，即：

$$\beta_{k+1} = \beta_k - \frac{F(\beta_k)}{F'(\beta_k)}$$

这要求我们设置初值  $\beta_0$ ，并且带入迭代式进行一定次数的迭代，方可求出指定收敛精度下的近似解  $\beta$ ，使得  $S$  最小。

在计算的过程中，记  $(\beta_{k+1} - \beta_k) = \Delta\beta$ ，将牛顿迭代转化成：

$$\begin{aligned} \beta_{k+1} &= \beta_k + \Delta\beta \\ F'(\beta_k)\Delta\beta &= -F(\beta_k) \end{aligned}$$

并后街线性方程组：

$$\begin{bmatrix} \frac{\partial g_0(\beta)}{\partial r_0} & \dots & \frac{\partial g_0(\beta)}{\partial r_3} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_3(\beta)}{\partial r_0} & \dots & \frac{\partial g_3(\beta)}{\partial r_3} \end{bmatrix} \cdot [\Delta\beta_0, \Delta\beta_1, \Delta\beta_2, \Delta\beta_3]^T = -F(\beta_k)$$

此线性方程组可以通过高斯消元法或雅可比迭代法求解。

最终需要设定  $\varepsilon$  精度值，当  $|\Delta\beta| \leq \varepsilon$  时即可停止迭代。

牛顿迭代相对于正弦回归公式堆叠更多，代码构造难度和调试难度有所增加且仍然需要有足够的序列长度保证不发生梯度爆炸。尤其在构造雅可比矩阵的时候，需要在之前求偏导后的结果再求一次偏导，最终，我们团队放弃了这个方案。

#### 4.4.11 方案优缺点及未来优化方向

- ◆ 抬枪补偿

公式法和龙格库塔法是基于同样的抬枪模型进行计算，也同样需要列出相同的微分方程然后进行求解。效率和精确度上龙格库塔法上会更加优秀，但是理解难度会比较大。

- ◆ 拟合能量机关参数算法

梯度下降法的手打版本在区域赛因为代码耦合原因没有部署，而在参考 Ceres 官方文档后，尝试使用 Ceres 优化最小二乘法拟合，堆入一定量的风车转速后便可开始四个参数拟合，通过 RMSE 之后，每一次拟合只需要拟合初相即可，效率比较可观，但仍然因为拟合前较大的系统误差会造成影响，如当  $A = B$  的时候，能量机关会在某一段时间速度极小，此时风车转速在以逐帧法计算上的误差会非常大，甚至可能出现方向算反的情形。我们考虑优化拟合优化上下界和优化转速计算来减小相关的系统误差。

- ◆ 粒子滤波

粒子滤波是基于蒙特卡洛方法进行计算，当粒子数量越多，越能接近真实分布。它本身已经摆脱了非线性滤波问题时随机量必须满足高斯分布的制约，可以表达比高斯分布更广泛的分布，也对变量参数的非线性特性由更强的建模能力。因此粒子滤波能够比较精确表达基于观测量和控制量的后验概率分布。而它的缺点是需要大量的样本，所以需要控制滤波所用的粒子数量。

#### 4.4.12 相较于完整形态的变化

视觉相较于完整形态，在能量机关的算法上有较多的迭代：

- 抬枪补偿的算法更新成龙格库塔法(RK4),并也将其也部署于针对装甲板的自瞄
- 在能量机关添加了粒子滤波，在进行滤波之后再行拟合
- 部署了 Ceres 优化的最小二乘法，并优化了外部逻辑
- 尝试部署 yolov5，因算力不足在区域赛没有部署，暂时使用传统识别。目前成功部署 yolov7，并尝试优化
- 利用相机-IMU 联合标定计算 $R_{CI}$ 和 $R_{IC}$

## 4. 研发迭代过程

### 5.1 版本迭代过程记录

#### 5.1.1 整体版本

考虑到平衡步兵各组版本不同步，因此在整体版本中仅记录平衡步兵的大版本迭代过程，各组迭代细节及问题可见 5.1.2-5.1.5 各组内部版本说明

版本	功能或性能详细说明	完成时间
V1.0	<ol style="list-style-type: none"> <li>1. 平衡步兵验证车组装完工</li> <li>2. 电控成功使得验证车具有保持平衡并全向移动的能力</li> <li>3. 使用 H 桥 buck - boost 电路，重新设计超级电容</li> <li>4. 视觉能量机关识别中增加对于圆心的识别</li> </ol>	2021/11/10
V2.0	<ol style="list-style-type: none"> <li>1. 平衡步兵联盟赛版完工，安装挡板并扩容弹仓</li> <li>2. 平衡步兵能够在平地进行流畅的小陀螺及移动小陀螺</li> <li>3. 底盘存在偶发失控问题</li> <li>4. 发射机构实现 15m/s 弹速 5m 击打小装甲板 100%命中率，30m/s 弹速 7m 击打小装甲板 80%命中率</li> <li>5. 基本解决单发是尿弹的问题</li> <li>6. 重新设计超级电容控制板</li> <li>7. 使用多种 Yolo 方案进行能量机关及装甲板识别</li> </ol>	2022/1/21
V2.5	<ol style="list-style-type: none"> <li>1. 平衡步兵底盘代码全面优化，彻底解决失控问题，底盘能够稳定静止在原地并流畅移动</li> <li>2. 底盘不贴地的情况下无法上大于等于 15°的坡</li> <li>3. Buck-Boost 版超级电容更换部分超级电容模块，新增保护模块，测试未能完成并安装</li> <li>4. 部署单 Buck 超级电容</li> <li>5. 部署传统视觉完成装甲板识别但不具备击打能量机关能力</li> <li>6. 初步加入 UI 界面，显示底盘朝向、摩擦轮、弹仓盖等基本信息</li> </ol>	2022/6/10
V3.0	<ol style="list-style-type: none"> <li>1. 加入减震、救援机构。</li> <li>2. 移除挡板并外拓装甲板位置</li> </ol>	2022/8/15

版本	功能或性能详细说明	完成时间
	3. 修改摩擦轮间距，使得 30m/s 弹速 7m 击打小装甲板命中率 95% 4. 优化功率控制，借助超级电容能够在底盘不贴地的情况下具备上 15°坡的能力 5. 增加 UI 功能，如自瞄状态 6. Buck-Boost 超级电容成功部署 7. 具备击打大、小能量机关的能力且能够保证在 10 秒内完成	

表 5.1: 整体迭代过程

### 5.1.2 机械部分

版本	内容	问题	日期
V0.0	平衡步兵验证车设计完工	/	2021/10/10
V1.0	平衡步兵验证车组装完工	/	2021/10/15
V1.1	增高整车重心	平衡车重心太低会导致调试困难	2021/10/23
V1.2	设计了新的可调节的轮组安装板	没有悬挂系统的验证车的四个麦轮抓地力不同，导致电控很难控制	2021/11/1
V1.3	通过增厚板子来增加刚性	施加较大下压力时轮组安装版会发生形变，导致麦轮“外八”	2021/11/2
V2.0	平衡步兵实车组装完工	/	2021/12/1
V2.1	增加防弹挡板，弹舱扩容至 300	弹仓容量不足，仅有大概 250	2021/12/15
V2.2	1. 更换 PC 位置，由云台架内部改为外置 2. 增高机器人云台高度	原 PC 位置不合理，存在干涉风险，不利于走线	2021/12/20
V2.3	增加双枪模块	/	2022/1/7
V2.4	优化单发限位	射击时有小概率出现按一下鼠标射击	2022/1/9



版本	内容	问题	日期
		出两发 17mm 弹丸的情况	
V2.5	规则变动, 增加图传云台模块	联盟赛规则中发射机构无法长时间与装甲板方向垂直	2022/2/15
V3.0	增加减震系统	用于适应对抗赛起伏路段, 防止冲击力损坏结构	2022/7/1
V3.1	增加救援支架	用于对抗赛工程救援	2022/7/10
V3.2	更换发射机构摩擦轮间距	弹丸加速度不够导致散布大	2022/8/1

表 5.2: 机械迭代过程

### 5.1.3 电控部分

版本	内容	问题	日期
V1.0	<p>1. 尝试使用一阶倒立摆作为底盘控制模型利用角度和加速度搭建传递方程,同时查找 3508 转矩常数, 得到 3508 转矩常数, 求得 CAN 回路发送值与电机输出扭矩的关系, 进而得到带入车辆数据进行模型搭建在控制上只使用单环 PID 即角度环进行控制</p> <p>2. 使用 Simulink 进行参数整定</p>	/	2021/12/25
V1.5	<p>1. 使用串级 PID, 提高系统相应速度, 并针对角速度噪声较大的问题, 加入低通滤波</p> <p>2. 加入车辆速度环, 使车辆可以静止在原地</p> <p>3. 加入麦轮速度环, 便于横向运动</p>	<p>1.3508 发送 CAN 回路发送值与电机输出扭矩并非线性对应关系切受转速影响严重</p> <p>2. 4 个轮子阻力情况不相同, 无法通过控制 CAN 发送</p>	2022/4/5

版本	内容	问题	日期
	4. 取消转向环，使用麦轮逆向运动学公式进行车辆转向	电流得到电机输出扭矩 3. 平衡车并非理想倒立摆。其重心并不准确在几何中心。若按倒立摆进行控制，则无法进行在原地。因此需要加入车辆速度环单环角度 PID 响应速度慢且微分项噪声严重	
V2.0	1. 解决底盘无预兆失控问题 2. 显著性的优化了底盘角速度环与电机速度环的参数，解决了因角速度环性能不够而导致的高速情况下失控 3. 整车代码达到上场需要	因 3508 减速箱虚位，静止时底盘出现轻微摆动	2022/6/10
V2.1	1. 健全 UI 模块 2. 加入底盘功率控制，配套使用新版超级电容	/	2022/7/8

表 5.3: 电控迭代过程

#### 5.1.4 硬件部分

版本	内容	问题	时间
V1.0	采用 H 桥 buck - boost 电路，超级电容新赛季重做.	旧版超级电容控制板，传统的半桥设计方案，存在明显缺点，电能的利用率低发热严重	2021/10/15
V1.1	经过讨论重新设计控制模块 PCB	控制模块 PCB 设计设计不合理	2021/11/2

版本	内容	问题	时间
V2.0	1. 在原有的设计基础上新增了 3.3v 电压基准芯片用来校准 ADC 采样信息 2. 更换 RGB 指示灯型号解决对 MCU 供电不足的问题 3. 更换连接器，保证板与板连接稳定	1. ADC 采样存在问题 2. MCU 供电不足 板子之间连接不稳定	2022/2/15
V2.1	1. 功率模块更换 IC 驱动 2. 新增保护模块	驱动内部无隔离，反冲的电动势会对控制模块产生不可逆伤害	2022/3/15
V2.2	所有模块调试完毕，超级电容工作状态趋于稳定	/	2022/7/1

表 5.4：硬件迭代过程

### 5.1.5 视觉部分

版本	内容	问题	时间
V1.0	增加对于圆心的识别	旧版击打能量机关时，未考虑云台的移动对摄像头坐标系的影响	2021/11/5
V2.0	为更好的进行神经网络测试，利用网络资源自制包含 2500 余张含装甲板图片作为数据集	/	2021/11/21
V2.1	为得到更优的模型，增加 PP-YOLO 进行装甲板识别的测试	/	2021/12/7
V2.2	为得到更优的模型，增加 YOLOX 进行装甲板识别的测试	/	2021/12/14
V2.3	为得到更优的模型，增加 YOLO-V5s 进行装甲板识别的测试	/	2021/12/27
V2.4	尝试将 YOLO-V5 的 head 替换为 Shufflenet-V2	/	2022/3/1

版本	内容	问题	时间
V2.5	尝试各种尺寸的 YOLO-V5	/	2022/4/1
V2.6	神经网络开发同时，使用传统视觉方案作为第二套自瞄系统	/	2022/6/1
V3.0	尝试使用 YOLO-V7	/	2022/8/15

表 5.5: 视觉迭代过程

## 5.2 重点问题解决记录

序号	问题描述	问题产生的原因	问题解决方案	解决人员
1	电路模块在机器人上的安装位置确定方式	平衡步兵要考虑的电路模块重量的分配，左右重量不均会导致失衡	将电池和 C 板放在左侧，剩余模块均放在右侧保证质量分布均匀	机械：郑浩桦 硬件：李睿 电控：林文浩
2	导电滑环型号的选用与线路分配方式	平衡步兵的电路系统与普通步兵稍有不同	重新制作新的导电滑环部分	硬件：李睿、闫嘉伟
3	云台需要通过 CAN 协议向下板发送 3 个浮点数	直接发送 3 个浮点数需要占用 12 字节，但是由于 CAN 协议每个数据包只能有 8 字节。分开发送将导致同一次速度数据将被不同的控制循环使用	对数据进行归一化操作。将原本-4.000 到 +4.000 的数据转化成 0-8000（16 位无符号整型）并在底盘开发板用相同的方式解包即可。	电控：林文浩
4	移动小陀螺情况下，移动方向偏差	$V_x$ 方向的加速度与 $V_y$ 方向的加速度不一致，导致速度合成方向出现偏差	使用 $V_x$ 方向的当前速度，与合成速度的目标角度作为当前横向速度	电控：林文浩
5	遥控器快速拨杆导致底盘触地	快速拨杆输出的信号近似于跃迁信号	使用一阶低通滤波对遥控器信号进行处理	电控：林文浩

序号	问题描述	问题产生的原因	问题解决方案	解决人员
6	匀速移动小陀螺所出现的云台周期性角度倾斜，尝试使用 sin 函数作为移动函数，反而取得了更差的效果	问题不是因为底盘 $V_x$ （前进）方向的速度不断改变，需要不断输出加速度。而是在匀速移动小陀螺情况下，整车整体速度在匀速移动方向保持不变，此时电机输出仅为克服阻力所做功，而非加速度	现有机械设计情况下只能优化，无法完全消除	电控：林文浩
7	没有悬挂系统的验证车的四个麦轮抓地力不同，这会导致电控很难控制与调试	1. 地面无法做到绝对平整 2. 装配存在误差	为验证车设计了新的可调节的轮组安装板并为实车设计特殊的自适应悬挂	机械：郑浩桦
8	当施加较大下压力时轮组安装版会发生形变，导致麦轮“外八”	单侧支撑轮组常见问题	验证车：通过增厚板子来增加刚性 实车：设计特殊结构的轮组结构来增加轮组刚性	机械：郑浩桦
9	弹舱容量约 200，不够使用，需要扩容	弹舱高度较低	增加弹舱四周板的高度	机械：郑浩桦
10	步兵正面上 15 度坡时倾角不够	机器人重心太低，倾角不足以抵消坡度	增加云台架高度，提高整车重心	机械：郑浩桦
11	能量机关中心点位置确定	采用模板检测“R”字符算法复杂度过高导致识别速度大大降低，不满足比赛需求，且膨胀后识别正确率低	采用最小外接圆包围轮廓，通过离心率和位于图像中的位置来判定是否有可能为中心点	视觉：张昱轩
12	大能量机关的速度预测	利用队列维护进行训练不仅需要实时维护，而且预测出的精度会差很多	改用向量维护，并且无需出栈，当训练完成直接将预测的常数保存，并且不做其他操作	视觉：徐亦乐

序号	问题描述	问题产生的原因	问题解决方案	解决人员
13	平衡步兵无预兆底盘失控	底盘平衡控制任务优先级较低，控制不为实时	将传感器任务和底盘控制任务的优先级调整为实时	电控：林文浩
14	摩擦轮电机重启	裁判系统短暂掉线导致摩擦轮发射转速设置设置为 15m/s 保护速度	调整裁判系统掉线的判定时间	电控：林文浩
15	急速前进会无法减速	角速度环性能不够	重新优化角速度环参数	电控：林文浩
16	弹丸远距离准确度低	摩擦轮间距大，导致弹丸加速度不足	优化发射机构，减小摩擦轮间距，使用质量更大的摩擦轮	机械：郑浩桦
17	大能量机关的速度预测 2.0	对于需要持续拟合相位的情形向量维护非常吃力，如果遇到断层会十分困难	使用 deque 容器,以实际要求做实时维护，在 Ceres 优化下效率依旧较高	视觉：徐亦乐
18	能量机关旋转方向问题	由于系统误差，在风车的瞬时转速极慢甚至为 0 的时候存在计算转速相反的情形，但是计算方法没有问题	采用取众数法，如果计算某一个方向的数量多余另一个，那么选择这个方向并一直沿用	视觉：徐亦乐

表 5.6: 重点问题解决记录

## 6. 团队成员贡献

技术方向	技术方向贡献度	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（总共 100%）
机械	25%	郑浩桦	机器人工程大三机械组组长	参与平衡步兵机器人机械结构的大部分设计	15%

技术方向	技术方向贡献度	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（总共100%）
		陆骁	电子科学与技术 大一 机械组组长	参与审图及部分发射机构机械设计	5%
		来恺杰	信息与计算科学 大二 队长 机械组组长	参与机械审图	3%
		尹章俨	经济学 大二 机械组副组长	参与机械审图	2%
电控	25%	林文浩	电子科学与技术 大二 软件开发负责人	负责整车的嵌入开发，包括云台控制代码，底盘平衡代码，发射代码等	20%
		赵海州	机械电子工程 大三 顾问	平衡车算法原理，云台PID控制原理讲解	5%
视觉	25%	马震	信息与计算科学 大二 视觉组组长	参与机器人视觉系统开发 参与视觉识别与预测开发	9%
		吴子昂	信息与计算科学 大三 视觉组前任组长 兼现任顾问	提供技术指导 参与机器人视觉系统开发 参与神经网络构建	8%
		张昱轩	数据科学与大数据技术	参与视觉识别开发	8%

技术方向	技术方向贡献度	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（总共100%）
硬件	25%		大一 视觉组组长	参与神经网络构建	
		徐亦乐	计算机科学与技术 大一 视觉组组长	参与弹道模型设计 参与视觉预测开发	8%
		李睿	电子科学与技术 大二 硬件开发负责人	负责整辆的电路系统的设计与布置，自研电路模块的研发与制作	10%
		闫嘉伟	电子科学与技术 大一 硬件组成员	负责线路的与导电滑环的制作	5%
		李俊彦	电气工程 博士 硬件组顾问	超级电容方案设计	10%

表 6.1: 团队成员贡献

## 7. 参考文献

[1]. “我们，是冠军” mp.weixin.qq.com.

<https://mp.weixin.qq.com/s/bKNb2ks5vCX29pYHrNqQSQ> (Accessed Sept. 8, 2022)

[2]. “【RM2022】南部赛区小组赛 佛山科学技术学院 vs 合肥工业大学宣城校区”

(June 10, 2022). Accessed: Sept. 8, 2022. [Online Video]. Available:

[https://www.bilibili.com/video/BV1YT411571k?spm\\_id\\_from=333.999.0.0&vd\\_source=763c713f7fb387713c8fa200d87658b8](https://www.bilibili.com/video/BV1YT411571k?spm_id_from=333.999.0.0&vd_source=763c713f7fb387713c8fa200d87658b8)

[3]. “这是“秃头机械”的最后一个投稿” (Aug. 5, 2022). Accessed: Sept. 8, 2022.

[Online Video]. Available:

[https://www.bilibili.com/video/BV15T411L7P8?spm\\_id\\_from=333.999.list.card\\_archive.click&vd\\_source=7b1d59c5310ff104c57dd6d38610f522](https://www.bilibili.com/video/BV15T411L7P8?spm_id_from=333.999.list.card_archive.click&vd_source=7b1d59c5310ff104c57dd6d38610f522)



- [4]. Harry-hjj (2021) “上海交通大学的视觉系统方案” [Source code].  
<https://github.com/Harry-hhj/CVRM2021-sjtu>
- [5]. “RM2021-大连理工大学-凌 bug-双向超级电容硬件开源” bbs.robomaster.com.  
<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12191> (Accessed Mar. 10, 2022)
- [6]. Q.Li. “PCB 设计时走线宽度和电流的关系” elecfans.com.  
<http://www.elecfans.com/d/659395.html> (Accessed Mar. 10, 2022)
- [7]. “MD\_2EDF7275K Datasheet” Available:  
<http://datasheet.eeworld.com.cn/part/2EDF7275KXUMA1,InfineonTechnologies,21521780.html>
- [8]. “BSC070N10NS5 Datasheet” Available:  
<http://datasheet.eeworld.com.cn/part/BSC070N10NS5,INFINEON,5681171.html>
- [9]. “STM32F334C8T6 Datasheet” Available:  
<http://datasheet.eeworld.com.cn/search.html?from=datasheet&t=partno&k=STM32F334C8T6>
- [10]. “STM32F103C8T6 Datasheet” Available:  
<http://datasheet.eeworld.com.cn/search.html?from=datasheet&t=partno&k=STM32F103C8T6>
- [11]. “MD\_2EDF7275K Datasheet” Available:  
[http://datasheet.eeworld.com.cn/part/2EDF7275KXUMA1,Infineon\\_Technologies,21521780.html](http://datasheet.eeworld.com.cn/part/2EDF7275KXUMA1,Infineon_Technologies,21521780.html)
- [12]. “INA240\_A1PWR Datasheet” Available:  
[http://datasheet.eeworld.com.cn/search.html?from=datasheet&t=partno&k=INA240\\_A1PWR](http://datasheet.eeworld.com.cn/search.html?from=datasheet&t=partno&k=INA240_A1PWR)
- [13]. “OPA2350UA\2K5 Datasheet” Available:  
<http://datasheet.eeworld.com.cn/part/OPA2350UA-2K5,TI,13234141.html>
- [14]. “【山东理工大学】自定义 UI 开源” bbs.robomaster.com.  
<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=11924> (Accessed Mar. 10, 2022)
- [15]. “RM2021-西交利物浦大学-三摩擦轮双枪口步兵机械开源”  
bbs.robomaster.com.  
<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12196> (Accessed Mar. 10, 2022)

- [16]. Bochkovskiy A, Wang CY, Liao HYM, YOLOv4: optimal speed and accuracy of object detection, in Conference on Computer Vision and Pattern Recognition, online, 23 Apr 2020.
- [17]. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767.
- [18]. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv2016, arXiv:1506.01497.
- [19]. GitHub. 2022. GitHub - ultralytics/yolov5: YOLOv5 in PyTorch → ONNX → CoreML → TFLite. [online] Available at: <https://github.com/ultralytics/yolov5> [Accessed 30 March 2022].
- [20]. Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), pages 116–131, 2018.
- [21]. “RM2022 沈阳航空航天大学步兵视觉识别算法开源”, [bbs.robomaster.com](https://bbs.robomaster.com/forum.php?mod=viewthread&tid=21978&extra=page%3D1%26filter%3Dtypeid%26orderby%3Ddateline). <https://bbs.robomaster.com/forum.php?mod=viewthread&tid=21978&extra=page%3D1%26filter%3Dtypeid%26orderby%3Ddateline>, (accessed Sept. 8, 2022).
- [22]. 塞利斯基, 艾海舟. 计算机视觉:算法与应用[M]. 清华大学出版社, 2012.
- [23]. Cohen, J., Cohen P., West, S.G., & Aiken, L.S. Applied multiple regression/correlation analysis for the behavioral sciences. Hillsdale, NJ: Lawrence Erlbaum Associates. 2003.
- [24]. Draper, N.R. and Smith, H. Applied Regression Analysis. Wiley Series in Probability and Statistics. 1998.
- [25]. Avriel, Mordecai (2003). Nonlinear Programming: Analysis and Methods. Courier Corporation. ISBN 978-0-486-43227-4.
- [26]. Zhenpeng Ge. 迭代弹道模型. Retrieved 2019.1.3 from [https://robomaster-oss.github.io/rmoss\\_tutorials/#/rmoss\\_core/rmoss\\_projectile\\_motion/projectile\\_motion\\_iteration](https://robomaster-oss.github.io/rmoss_tutorials/#/rmoss_core/rmoss_projectile_motion/projectile_motion_iteration), 2019
- [27]. 李晓辉, 任伟和, and 程长胜. “求解非线性方程的牛顿迭代法.” 科技风 14(2021):3.
- [28] 李庆扬. 数值分析[M]. 清华大学出版社有限公司, 2001.
- [29] A. Bjorck, Numerical Methods for Least Squares Problems, SIAM, 1996
- [30] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” in IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174-188, Feb. 2002, doi: 10.1109/78.978374.

[31] Thrun S. Probabilistic robotics[J]. Communications of the ACM, 2002, 45(3): 52-57.