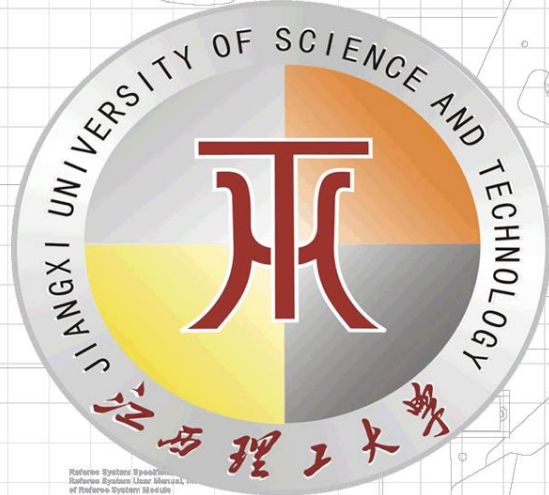




Using a 52-56 motor shroud and
State-Optimized Control (SOC), the
RoboMaster C630 Brushless DC Motor Speed
Controller enables precise control over motor
torque.



Reference System: Speed
Reference System: User Manual
of Reference System: Manual

江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ROBOMASTER 超级对抗赛

乘风战队技术报告

江西理工大学 乘风 编制

2022年08月 发布

目录

1. 其它学校机器人分析综述	4
1.1 机械组	4
1.2 电控组	6
1.3 视觉组	8
1.4 硬件组	9
2. 机器人功能定义	10
3. 机器人核心参数	11
4. 设计方案	12
4.1 机械结构设计	12
4.1.1 整体结构设计	12
4.1.2 工艺选择	24
4.1.3 模块的定位及走线.....	27
4.2 硬件设计	28
4.2.1 整机硬件框图	28
4.2.2 硬件详细设计	28
4.2.3 关键器件的选型	34
4.3 软件设计	37
4.3.1 系统架构	38
4.3.2 运行流程	40
4.3.3 重点功能	42
4.3.4 软件测试	49
4.4 算法设计	50
4.4.1 功能简介	50
4.4.2 核心算法原理	53
4.4.3 算法性能分析	59
4.4.4 算法库介绍与接口说明.....	60
4.4.5 算法结果	62
4.5 其它	63
4.5.1 UI 交互开发.....	63
5. 研发迭代过程	64
5.1 版本迭代过程记录	64
5.2 重点问题解决记录	66
5.3 团队成员贡献（2.5）	70

6. 参考文献 71

1. 其它学校机器人分析综述

1.1 机械组

- 上海交通大学 (<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12241>)

1. 轮组联轴方式

与其他学校相比，上海交通大学在轮组联轴方式的设计上实现了创新。他们是通过使用带有“D”型槽口的铝件，并大胆地将其一分为二，通过螺丝夹紧 M3508“D”型电机轴，使三者紧密贴合，同时将连接孔制作为螺纹孔，使装配、拆卸更加方便。

我队上赛季联轴方式是将法兰盘和 M3508 电机直接配合。但由于法兰盘和 M3508 电机之间只是通过一个 M3 螺丝连接，当轮子在不同地段无规则运动时，由于螺丝受到的扭矩过大，

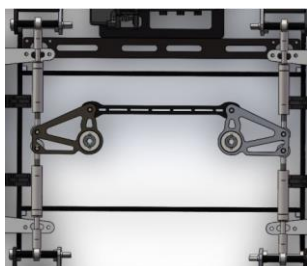


最终导致螺丝断裂在电机轴内以致电机作废的情况。而上海交通大学轮组联轴方式的设计很好的避开了这个问题，且该联轴方式可以传递较大扭矩。虽然增加了压紧盖，但在轮子频繁运动时，夹紧螺丝可能会出现松动情况，最终导致轮子脱落。但总体上来说，该设计方案依旧出色。而本赛季，我队大部分地面机器人在设计轮组联轴方式时也参考了上海交通大学设计理念。

图 1-1 轮组图

2. 自适应悬挂

与传统横臂、纵臂悬挂系统相比，自适应悬挂系统对复杂的地形通过性更高。而作为首个在 RoboMaster 赛场上使用自适应悬挂的大学，上海交通大学的自适应悬挂结构简单，功能



齐全。它的工作原理是当左前方轮子遇障碍抬起时，经悬挂杆带动后方轮子下降，之后再带动活动连杆向右，进而带动右侧前轮下降，后轮上升。同时在悬挂杆内嵌压簧，使英雄机器人行驶在起伏路段时，能够缓和冲击和吸收振动，提高英雄机器人行驶的平稳性，延长电机的使用寿命。

图 1-2 自适应悬挂图

- 大连理工大学 (<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12264>)

1. 拨盘设计

大连理工大学凌 BUG 队所设计的拨盘整体由板材镂空拼接而成，这有效地减去了多余的重量。且该拨盘为侧出拨盘，所以在不断往弹路链推送弹丸时，不会出现电机输出扭矩不够的现象。在单独测试拨盘时，并没有出现卡弹现象。但增加弹仓之后，在弹丸压满的情况下，

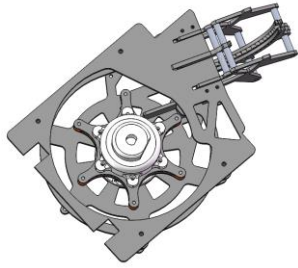


图 1-3 拨盘整体图

卡弹现象较为严重，而大部分可视为边缘性卡弹。产生卡弹的原因是雕刻精度不足，导致拨盘内壁实际大小与理论大小产生偏差，同时内壁可能存在毛刺或缺口，最后导致弹丸卡在拨盘内壁。另一种卡弹现象的原因是拨轮转动产生的力过大导致弹丸轻微跳起，最终卡在拨盘出弹口处。

2. 限位装置

大连理工大学凌 BUG 队的限位装置通过将限位装置的主体零件与拉簧进行连接，通过限位装置主体上左右两个往下倒垂的类圆柱性结构将英雄机器人大弹丸的初始进入位置限制在枪管的中心位置并贴紧枪管的下底面，保证了英雄机器人发射机构的稳定性。同时，由于该

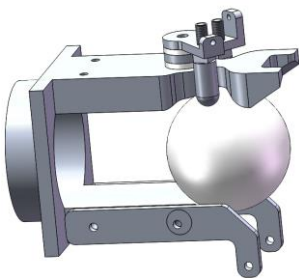


图 1-4 限位装置图

限位机构是通过采用拉簧对限位装置的回位时间进行设置，因此，可以对拉簧的弹簧系数进行不断调节直至达到最佳限位效果。在本赛季英雄机器人限位装置的设计上，在借鉴大连理工大学对于限位装置安装的思路以及结合队伍实际情况之后，队伍最终通过使用了一个用 TPU 材质打印的 3D 打印件实现了对英雄机器人大弹丸的限位功能。

- 哈尔滨工业大学 (<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12253>)

1. 自适应悬挂

与上海交通大学的自适应悬挂不同，哈尔滨工业大学的自适应悬挂是位于车架下方，这样可以将车架上方的空间空余出来，用于放置机器人的其他模块。同时将拉簧与轮组连接，

这样既可以达到自适应效果，又拥有减震效果。此外，哈尔滨工业大学在自适应悬挂中增加



减震之外，还额外增加了一组减震，更有效的适用复杂多变的地形。

但将自适应放置在车架下方，会导致英雄机器人底盘过低，而这最终会降低机器人的路面通过性。对于一些高低起伏的路面来说，底盘低的机器人是很难通过的。所以我们在设计时需要增加底盘的离地间隙，避免与地面刮蹭。

图 1-5 自适应减震图

1.2 电控组

- 东北大学 (<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=11033>)

东北大学运用的是双控制板模式，上板为云台主控板，主要负责对英雄机器人的整体控制，下板为底盘主控板，主要是接受云台主控板的信息进而对底盘等其它电机进行控制。东北大学主要推行基于 C++ 作为框架，将基础的功能封装成块，所有代码统一中间层，所有人运用同一个框架，所以使得书写代码更加快速便捷，减少非常多重复的代码量，进而使得英雄机器人整体的完成度较高。其中底盘在四个 M3508 电机的前提下，加了 4 个 GM6020 电机对底盘电机进行辅助，使得英雄机器人底盘的更加的稳定，功能性更强。而云台则是在基础的串级 PID 的条件下，加入了对 Yaw、Pitch 轴的卡尔曼滤波和前馈控制，让云台的稳定性更强，除了云台之外，东北大学在视觉的接受信息处也增加了卡尔曼滤波，使得视觉的预测更加的准确。而东北大学的发射机构的主要侧重点则是在发射的逻辑关系下，他们写了一套对各种情况的应对方式，主要是针对于防止发射机构超热量和超射速这两种情况。总而言之，东北大学的英雄机器人的总体完成度和技术水平都非常高，有非常多的借鉴之处。

其中东北大学对于上下两块控制板之间的通信想法，对于 CAN 通信的应用灵活性对我们有很大的帮助。他们的 CAN 通信函数在获取数据的那里使用了 memcpy 函数，能够在进行两块控制板通信的时候更加的方便，只需要在所需的任务内自定义数据数组，然后运用函数就行。除此之外云台的卡尔曼滤波和前馈控制的算法结构，底盘的功率控制和发射机构发射逻辑都值得我们借鉴。

- 广东工业大学 (<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12250>)

广东工业大学主要用 `rm-controls` 来对代码进行一个编写。`rm-controls` 是一套在 PC 上运行的无下位机电控软件，基于 `ros-control` 的硬件和仿真接口以及配套的控制器的，用于开发 RoboMaster 机器人和高性能机器人。并且他们非常的注重代码的重复性，所以使得他们代码的通用性非常的强，能够让所有的机器人都通用一套底层代码，不同的机器人只需要更改决策层的开发和加载不同数量和种类的控制器的。其中 `rm_chassis_contorllers` 用于对麦克纳姆轮、舵轮、平衡车的底盘进行控制，`rm_gimbal_controllers` 用于对射击模型和跟踪滤波预测的云台进行控制，`rm_shooter_controllers` 用于操作摩擦轮，拨弹盘完成发射控制器，`rm_calibration_controllers` 用于校准执行器位置。除此之外，还有许多小功能型的控制器对机器人进行控制，所以广东工业大学英雄机器人的整体完成度非常的高，他们对于代码逻辑的框架也值得我们借鉴。除了代码框架之外，广东工业大学对于一些软件的应用也对我们有很大的帮助，例如能够使用 `industrial_ci` 进行持续集成的 `gitlab` 服务器，它能够自动对代码进行编译、测试并发布到 `apt` 源上，使得每个兵种的关联性更加的强，易于对所有机器人进行总体更新。还有能够进行多刚体动力学仿真 `Gazebo`，能够大大的提高发开和迭代的效率。

- 深圳大学 (<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12262&fromuid=54465>)

深圳大学的软件层次主要分为模块层、设备层、控制层、协议层和驱动层。模块层主要由 `gimbal`、`launcher`、`chassis`、`control` 四个部分组成，设备层则由 `motor`、`imu_sensor`、`rc_sensor`、`judge-sensor` 四个部分组成，而控制层则负责各个电机模块的任务和控制方案的编写，协议层则负责 `CAN` 通信、`imu` 姿态解算、视觉等，最后的驱动层则是由 `CAN`、串口、`i2c`、`io` 等底层框架组成。深圳大学的主控则分为底盘和云台两部分，底盘主控作为中转站处理裁判系统、超电主控板和云台主控板的联系，云台主控负责整车逻辑控制并与底盘主控通信，以及自瞄功能的信息处理部分。条例清晰的软件层次和主控使得深圳大学的代码框架非常的完善，拥有非常高的完成度，也体现了他们的高水平技术。

虽然编写的代码框架与我们不同，但是其中使用的算法值得我们去分析借鉴以及改进。在于视觉通信的数据结构也可以参考，能够对我们上一代英雄机器人的通信进行升级。

1.3 视觉组

- 西北工业大学 (<https://github.com/NZqian/WMJ2021/blob/master>)

西北工业大学的视觉算法基于 C++ 为框架，装甲板识别算法使用基于检测目标特征的 OpenCV 传统方法，实现检测识别的中心思想是找出图像中所有敌方颜色灯条，并使用找出的灯条一一拟合并筛选装甲板。筛选装甲板的条件也为经过几何条件的筛选和一些极端条件的滤除，得到符合条件的装甲板，再经过数字识别得到最后的目标装甲板。在 PID 控制方面添加了速度解算器，返回云台速度修正值，跟随过程中较为稳定。

1. PID 调节器与速度解算器

工作原理：

(1) 将目标位姿和当前位姿相减得到位姿误差，作为 PID 调节器的输入量。

(2) PID 调节器根据已记录的误差和当前误差分别计算比例、积分和微分调节量，相加后得到调节修正值。如有必要，可以设置阈值进行分段调节。

(3) 根据上一帧数据和当前数据计算基准速度，然后加上速度修正值得到最终速度并返回。

该算法使用的是增量式 PID，在加上他们自己根据情况写的速度解算机，使用该算法可以保持稳定且快速的跟随。

2. 卡尔曼预测器

工作原理：

外部直接调用 `predict()` 获得预测后的装甲坐标，内部判断调用 `initFilter()` 初始化卡尔曼滤波器，并用第一帧输入的装甲绝对坐标进行多次预测使其收敛。判断当前帧和上一帧装甲相对距离，根据设置的阈值 (`target_change_threshold=0.25m`) 判断装甲板是否切换，装甲板切换需重新初始化滤波器，得到预测结果。

由于增量式 PID，相比于位置式 PID，容易通过加权处理获得比较好的控制效果，并且在系统发生问题时，增量式不会严重影响系统的工作。所以我们使用的也是增量式 PID，因此借鉴了西北工业大学的速度解调器，在原来的 PID 基础上进行优化后，能获得更好的控制效果。

1.4 硬件组

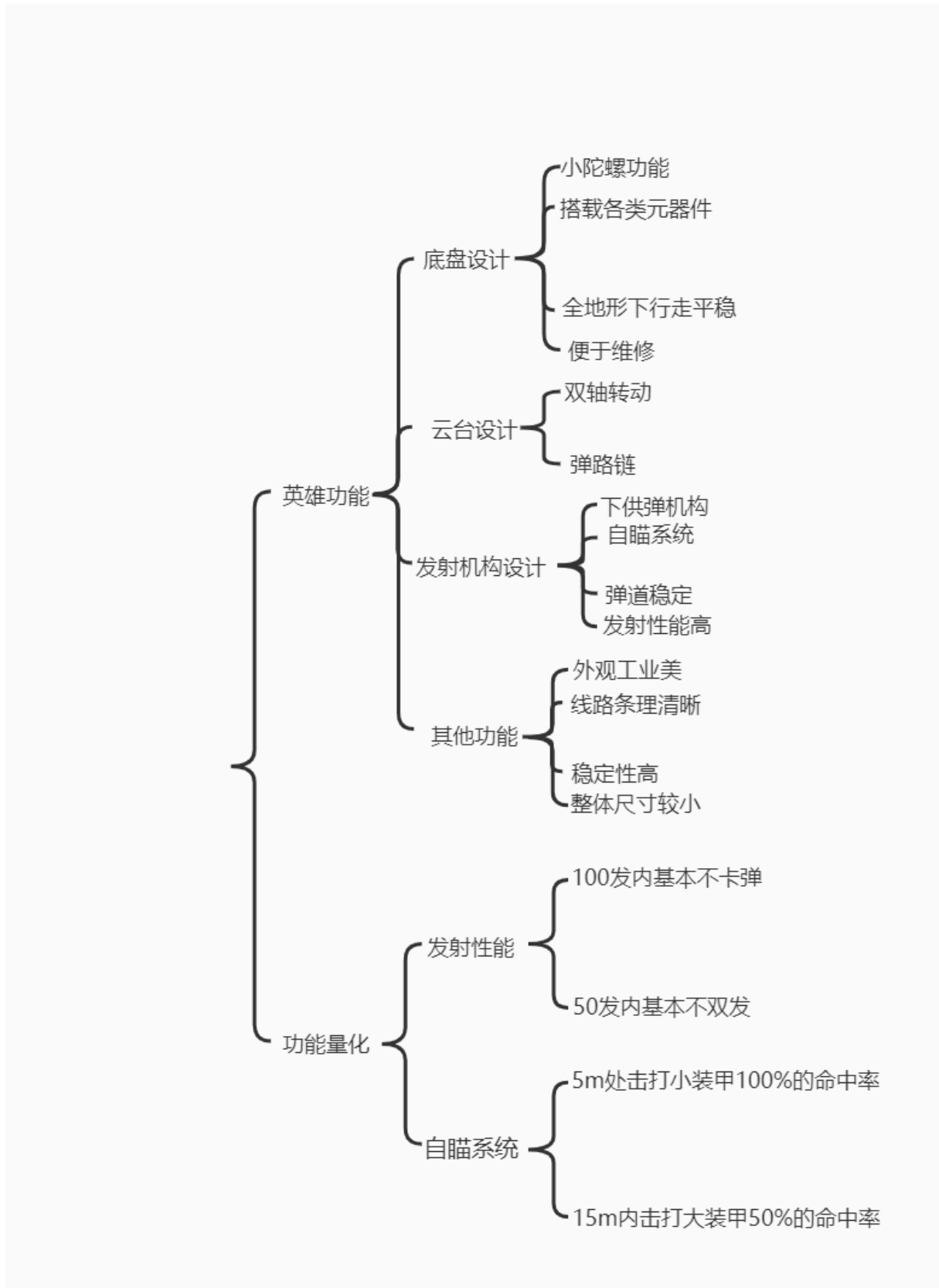
- 大连理工大学 (<https://bbs.robomaster.com/forum.php?Mod=viewthread&tid=12191>)

超级电容方案结合大连理工大学的超级电容硬件设计和本队设计的超级电容软件控制。大连理工大学的电路拓扑结构为电源管理并联底盘，并通过 **dc/dc** 双向升降压电路连接超级电容模组。具有成本低、不存在死电、底盘电压稳定、充放电效率高等优势，很好地解决了以往超级电容底盘电压不稳导致机器人无法动弹及损耗高的问题。

其控制思路为正常模式电源同时为底盘和超级电容供电，当超级电容满电时电源仅给底盘供电，**dc/dc** 不工作；当开启超级电容加速时，通过嵌入式控制高精度 **PWM** 波使得 **H** 桥电流反向放电，从而达到与电源管理同时对机器人底盘进行供电的目的，由于底盘和电源管理直接并联，底盘电机电压始终等于电源管理电压，不存在电机低于工作电压而不转的情况。

因而，我们结合其独特的电路拓扑结构设计了超级电容控制 **PCB**，在实测中很好的解决了因先前因硬件缺陷导致的系列问题；并且参照其通过 **HRTIM** 实现 **PWM** 高精度控制，我们结合实际选用了同样满足设计需求的 **STM32H7** 芯片。在嵌入式设计上，我们参考凌 **BUG** 的高边采样、滤波及线性校准实现电气数据的精确采集等优点，结合本队的设计思路形成本队第二代超级电容控制方案。

2. 机器人功能定义



3. 机器人核心参数

基本参数

质量/kg (含裁判系统)	重心高度/mm	尺寸/mm
29.2	227.88	680×550×650

其他参数

Pitch 轴俯角/°	Pitch 轴仰角/°	云台自由度	爬坡角度/°	最大移动速度/m/s
20	48	2	17	1.7

传感器型号、参数以及个数

传感器类型	传感器型号	个数
海康威视工业相机	MV-CA003-20GC	1

硬件参数

电路消耗/W	电容总容量/F	工作电压范围/V
240	5	8~26

执行器件用途及数量

执行器件	用途	数量
M3508 直流无刷减速电机	提供底盘轮组动力	4
M3508 直流无刷减速电机	用于拨盘拨齿转动	1
RoboMaster GM6020 直流无刷电机	用于云台 Pitch 轴及 Yaw 轴转动	2
M3508 直流无刷减速电机转子	用于摩擦轮的转动	2

4. 设计方案

4.1 机械结构设计

4.1.1 整体结构设计

本赛季，我们战队英雄机器人最大的不同是供弹方式的改变，由上赛季的上供弹英雄机器人变为下供弹英雄机器人。

下供弹英雄机器人可以将拨盘和弹仓置于底盘上，这样可以减少云台重量和体积，降低机器人重心，加快云台的响应速度，有利于云台控制的稳定性。同时拨盘和弹仓的位置的变化使云台空间变大，便于安装其他模块。此外，由于弹仓是放置在底盘之上，所以我们可以增大其容量，使单次载弹量增大。

如今，虽然下供弹英雄机器人技术逐渐成熟，但对于首次制作下供弹英雄机器人的队伍来说其技术难点相对较多。首先，下供弹英雄机器人的弹路链相对较长，增加了卡弹风险。此外，当 Pitch 轴俯仰角过大时，在弹路链和云台发射机构衔接处容易卡弹，卡弹严重时可能导致拨盘 M3508 电机损坏。而我队在下供弹英雄机器人的设计过程中由于经验不足，也遇到了很多困难，经历了两次迭代。我们在 [5.2 重点问题解决记录](#) 中有详细说明。

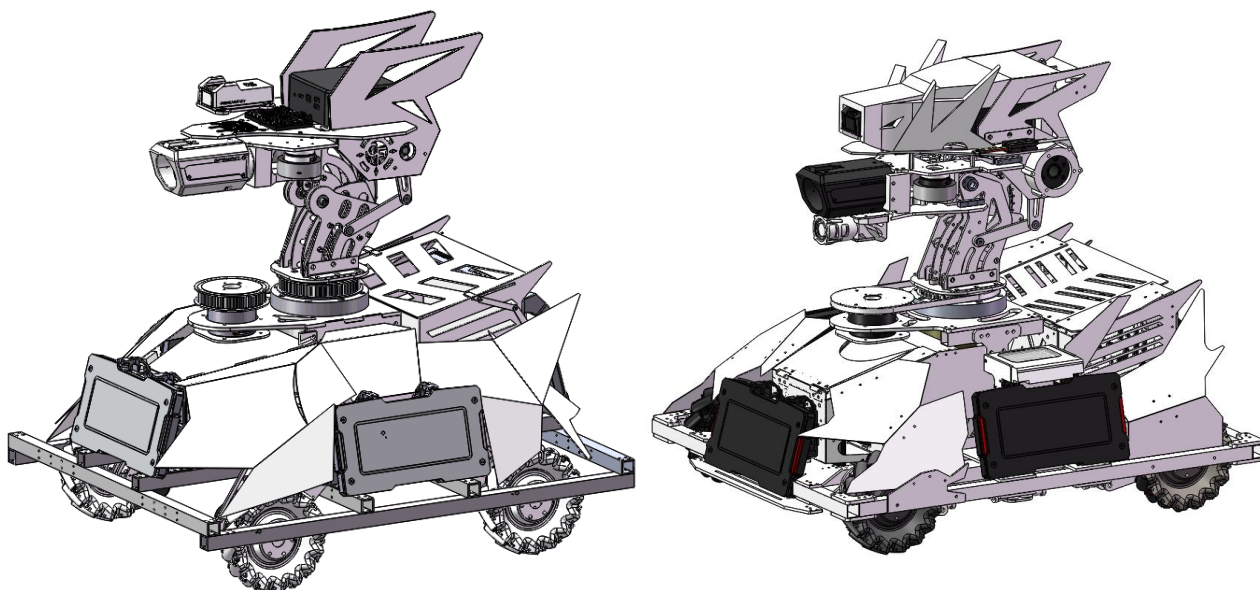


图 4-1-1 两代英雄对比图

4.1.1.1 底盘

本赛季英雄机器人的功能依旧是“攻城车”，该功能对于底盘的需求可表达为：高强度高精度，在这个前提条件下尽可能降低重量，实现轻量化设计，从而使英雄机器人向步兵的机动性靠拢。

1. 底盘框架

底盘框架是英雄机器人底盘的主要构成部分，不仅连接底盘各个模块，决定整体布局，而且起到了承载云台以及其他单元的作用。所以底盘框架的设计需求是“跟随需求”布局空间。

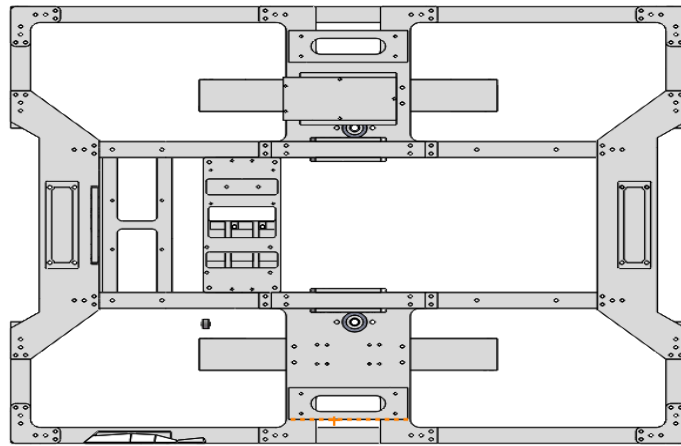


图 4-1-2 底盘框架俯视图

而在加工工艺的选择上，框架整体以 1mm 壁厚铝方管为主体，在保证轻量化的同时又确保了强度需求，并且考虑到人力、精力和加工精度等原因，决定外包工厂加工。其余连接件采用了玻纤板为耗材，自行加工，留下改进空间，便于随时调整。

底盘框架需求分析表：

种类	指标
横向杆	强度高，重量轻
连接件	强度高，支撑云台和各裁判系统等模块
保险杠	全包围，结构紧凑
底盘整体	尺寸紧凑，重心低，强度高，轻量化

2. 轮组模块

轮组模块	上赛季设计缺陷	本赛季设计方案
联轴器	法兰盘通过 M3 螺丝直连 M3508 电机。在电机处于工作状态时，M3 螺栓所受到的剪切力过大容易断裂堵塞 M3508 的螺纹孔。从而导致联轴器失灵，较为严重的是断裂在电机内的螺栓无法清除。这使得电机的功能受损。	在分析其原因后，设计者及时调整方案；采用压轴片来代替 M3 螺栓，其原理是：将 M3 的螺栓所受剪切力转变成压轴片所受轴向拉力；连接压轴片与 M3508 的电机则是 M5 螺栓。经过计算可得，M5 螺栓的抗拉强度远大于 M3 螺栓的抗剪切强度。
衔接设计	上赛季英雄机器人存在“外八”现象。摇臂与机器人主体（机架）衔接处的结构由两个法兰轴承与连接轴组成。	设计者在查找了大量资料后发现其“外八”的原因：首先法兰轴承的特点是：能够承受较大的径向载荷及较小的轴向载荷，当轴向载荷过大时，法兰轴承无法对其轴向（x 轴）约束。在实际分析中（具体见下）法兰轴承会承受较大的径向与轴向载荷，x 轴的约束不足从而出现“外八”现象。故为了保证机器人在全向移动过程中不会出现“外八”的现象，我们采用了两个法兰轴承加两个推力球轴承的方案设计。

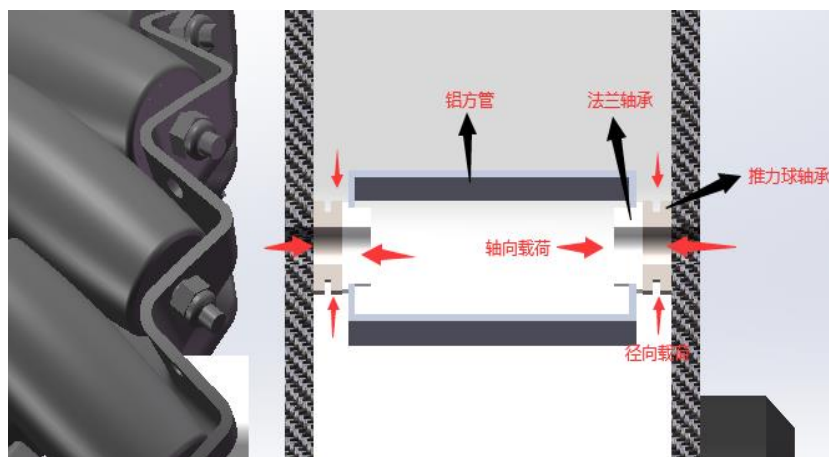


图 4-1-4 轮组分析图

3. 悬挂

由于悬挂最初的方案考虑不够充分，致使英雄机器人行驶时摇摇晃晃，而且比赛将至，拉簧一直都不是很适合，所以不得不放弃原来的方案改为传统独立悬挂。

- 原方案

自适应悬挂：

在 2022 赛季超级对抗赛场地更新后，场地起伏路段增加，极大考验了机器人底盘结构对于复杂地形的适应性以及通过能力，为更好的适应新的比赛场地，本赛季英雄机器人对底盘进行了机械自适应设计，所采用机械自适应结构为：每一个轮组通过拉簧（下图蓝线所示位置）与中间连接板相连接。下图为自适应结构初版，由连杆、鱼眼轴承和连接板连接四个轮组，测试后发现减震效果不佳，后续连杆会更新为拉簧，由此在起伏路段能起到良好的减震效果，且在比赛中也能应对各种意外带来的冲击，起到良好的缓冲作用。

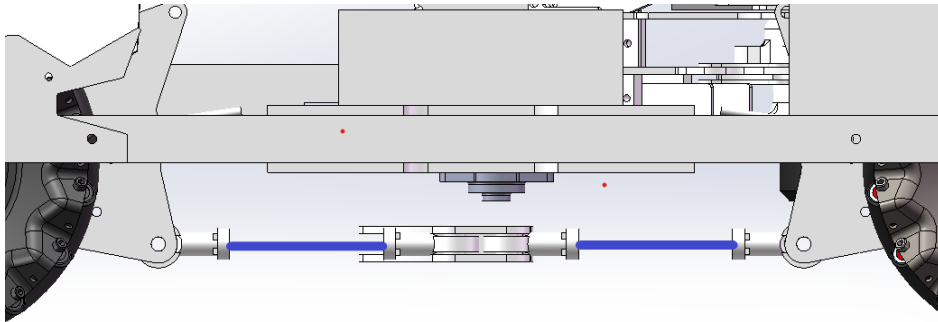


图 4-3 自适应减震侧视图

英雄机器人的自适应设计具体如下，其原理为，当轮子不在同一平面时，其拉簧拉紧，从而通过活动连杆牵动另一侧转点转动，偏转轮子所处位置，使其向同一平面偏转实现接触。从而增强其稳定性，减少其因上下斜坡以及台阶倾翻的可能性。

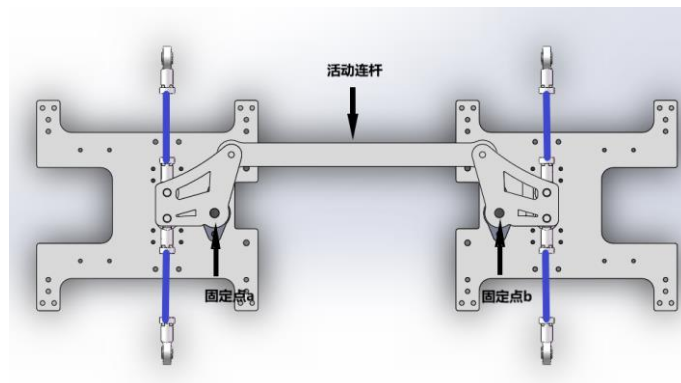


图 4-1-5 自适应减震俯视图

● 最终方案

独立悬挂：

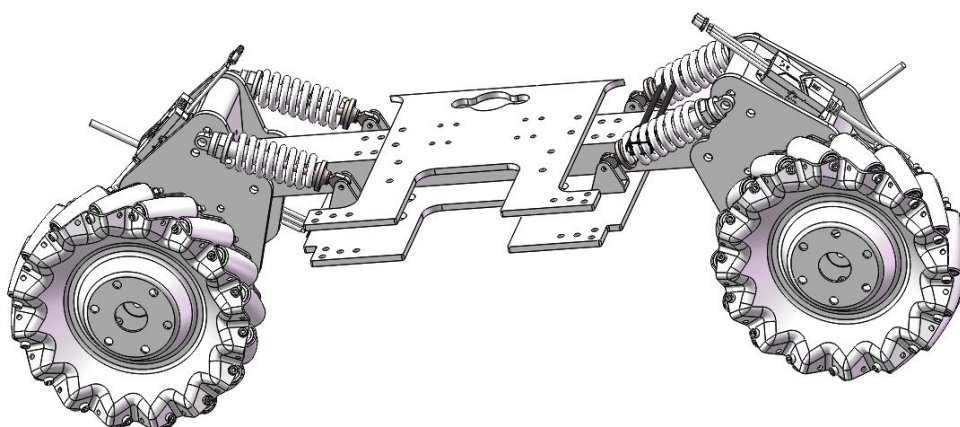


图 4-1-6 独立悬挂示意图

与非独立悬挂系统相比，独立悬挂系统是每侧车轮单独通过减震器连接在车架或车身下面。它的优点是质量较轻，可节省较多空间，提高了车轮的地面附着力。通过更改不同线径的弹簧，可以改善减震效果，提高操作手操作的舒适性。虽然纵臂悬挂会造成“点头”现象，但可以通过调控减轻该现象。

由于底盘侧面放置超级电容等、主控等模块，所剩空间不足，所以只能选择比较短的 85mm 的减震器。虽然 85mm 的减震器压缩量较小，但依旧可以滤震，能够比较顺利的通过盲道。

4.1.1.2 保护壳

在机械设计上，我们不仅要技术也要艺术。在技术上，保护壳的设计可以很好的保护裸露的电线、防护车体外部较为脆弱的转接板和裁判系统以及减少弹丸被射入车体内部而影响功能的可能性。在艺术上，一个经过充分设计的保护壳兼具美观与记忆点，响应 RM 比赛对于外观的要求，且充分展现本战队在 2022 赛季奋发昂扬的精神状态。

云台保护壳的要求是能够减缓对方弹丸的冲击，由此决定使用 10mm 厚度的珍珠棉作为保护壳的材料，并使用铁丝连接珍珠棉与玻纤板的孔洞，确保安装的稳定性。

车体的保护壳在设计、选材和加工工艺上都经过充分考虑。保护壳主要由合页连接，通过螺丝螺母与车架固定。保护壳选用了 1mm 厚度的玻纤板，在降低保护壳重量的同时又能保证其防护性。

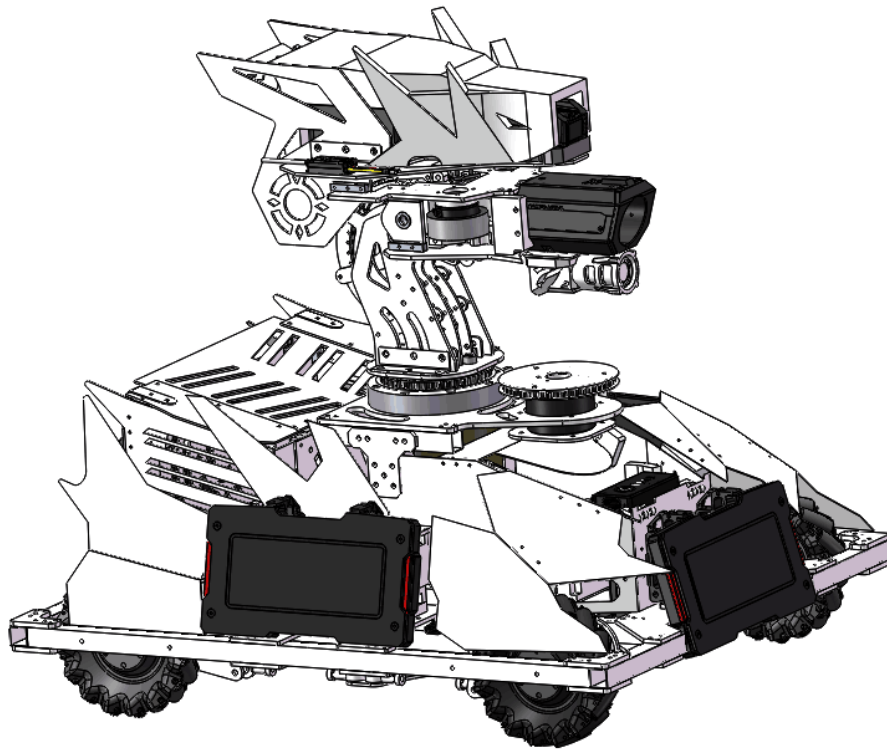


图 4-1-6 保护壳展示图

4.1.1.3 云台

1. 发射机构

英雄机器人不同于步兵机器人发射 17mm 小弹丸，英雄机器人在比赛中通过发射 42mm 大弹丸击打敌方机器人。对于英雄机器人的发射机构，鉴于英雄机器人超热量后果严重，且大弹丸的价格昂贵。因此，保证英雄机器人的发射稳定成为了英雄机器人发挥自身“攻城车”作用的重中之重，而对于英雄机器人的发射稳定，在英雄机器人的发射机构上设置起始件和单发限位装置可以很好的保证英雄机器人弹路稳定以及避免英雄机器人超热量现象的出现。

在对其他优秀学校开源报告中为保证英雄机器人发射稳定所选的各种方案和队伍研发中的实际情况进行分析后，我们认为影响英雄机器人发射稳定的关键在于保证大弹丸在被发射前的初始状态在一个比较小的范围内浮动，即尽可能的保证每一发英雄机器人大弹丸在接触摩擦轮瞬间都有相同的初始能量和初始角度。针对这种情况，我们在英雄机器人发射机构炮管前端设置了一个起始件和限位装置来保证大弹丸在接触摩擦轮时的状态稳定。如图（1）所示。我们首先是在英雄机器人弹路链和炮管的连接处设置一个起始件，通过起始件的一个连接确保了英雄机器人在云台进行俯仰角改变时大弹丸行进路程一致和每一发大弹丸在经过弹

路链到达炮管时处在同一个预备进弹位置且不改变大弹丸在英雄机器人弹路链中的行进路程。如图（2）（3）所示，

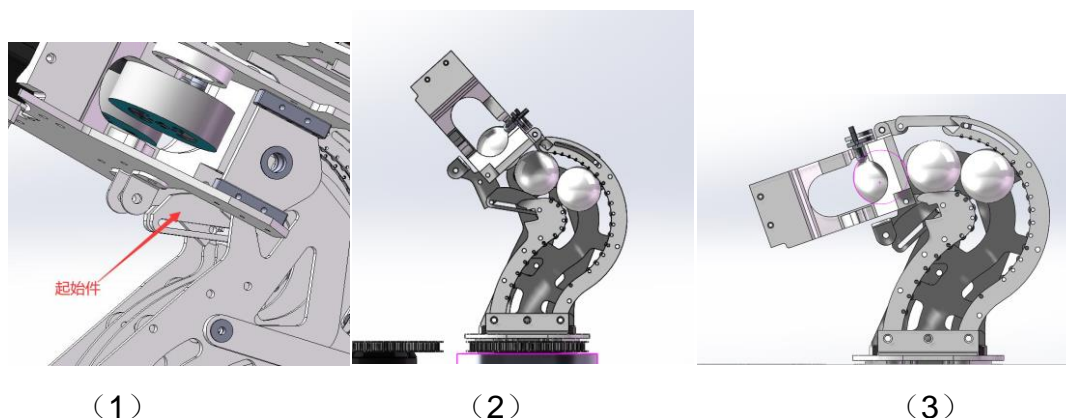


图 4-1-7 发射机构分析图

图示为当云台在最高仰角和最低俯角时预备进弹位置的大弹丸处于一个相同的位置，通过这样保证当前一发预发射大弹丸在被摩擦轮发射出去时，预备进弹位置的大弹丸在受到后一发大弹丸的推力进入预发射位置所经历的这一过程在持续拨弹时的位移进给量或者说状态变化量保持一致。确保大弹丸能够平缓的进入预发射位置，减少当预备进弹位置的大弹丸在向前行进到达预发射位置的状态不同对发射稳定性所造成的影响。除此之外，对于预发射位置的大弹丸，通过参考大连理工大学英雄开源报告对于英雄机器人大弹丸限位装置的安装思路，将英雄机器人的限位装置安装在了预发射位置的上方，如图（4）（5）：



图 4-1-8 限位分析图

通过使用一个半“X”型的 TPU 材质的打印件将预发射位置的大弹丸压在炮管的下方和中心位置，确保每一发大弹丸在处于预发射位置时的状态保证一致。同时，通过限位装置向下延伸的两个“小圆柱”使得预发射位置大弹丸处于一个临界状态，既在预备进弹位置大弹丸给予预发射装置大弹丸一定的推力才能使得预发射位置大弹丸去接触摩擦轮，从而实现一个单发的功能，避免英雄机器人出现因连发现象而导致的扣血。

2. Yaw 轴

本赛季下供弹英雄机器人的 Yaw 轴整体采用同步带结构，同步带相对于主轴直连可以降低速度；增加扭矩，提高加速性；同时能够更好的缓冲且留有足够的空间过大弹丸与电滑环。为了确保云台的稳定性及响应性，轴承采用虚位较小的交叉滚子轴承。

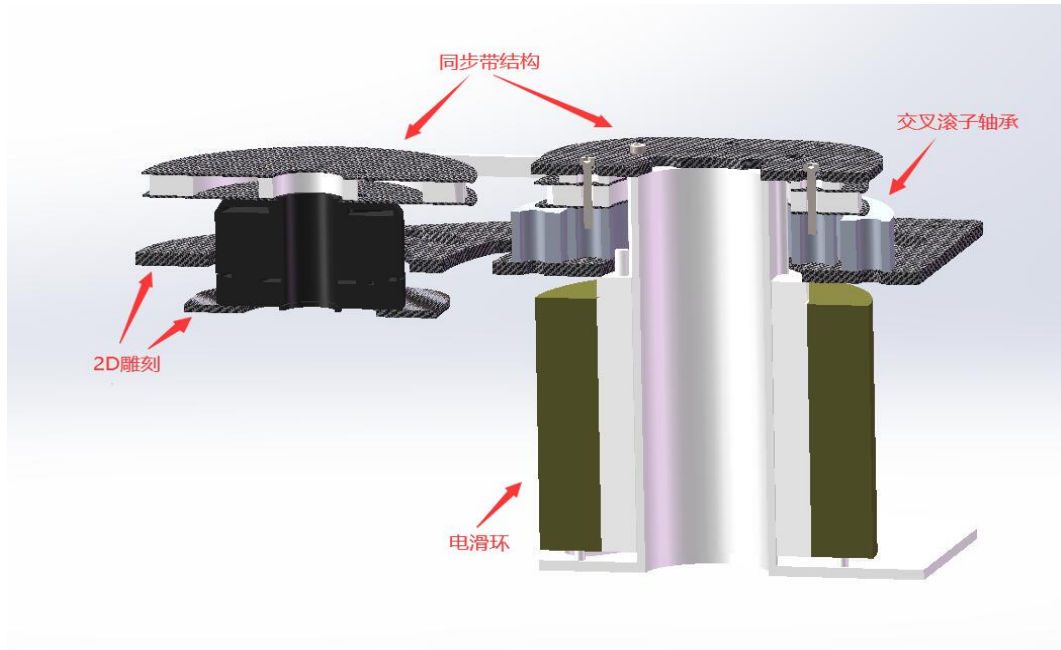


图 4-1-9 Yaw 轴分析图

3. Pitch 轴

Pitch 轴中的 GM6020 电机通过连杆机构来控制云台的俯仰角；考虑到云台的重心位置，我们将电脑的位置安装于 Pitch 轴转动中心的后面。保证了云台的整体重心处于中轴上。

传感器使用的是 MV-CA003-20GC 型号的海康威视工业相机，安装在 42mm 测速模块的下方，优点是帧率高、焦距大、可视距离远，且更改摄像头的曝光、焦距更加方便，唯一的缺点是由于焦距高因而可视范围小。

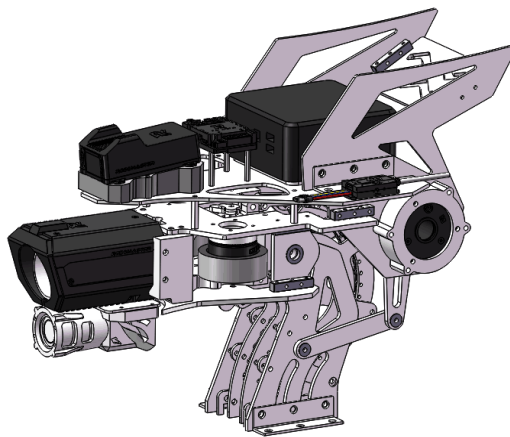


图 4-1-10 Pitch 轴整体示意图

4.1.1.4 供弹机构

1. 弹仓

英雄机器人的弹仓作为其存储大弹丸的机构，需有足够的空间。经建模计算以及实际试验之后，确认弹仓可存储的大弹丸上限为 67 发，对于超级对抗赛来说其大弹丸的存储空间已然非常充足。

弹仓盖的开合使用了象鼻锁，象鼻锁多用于开关门，其特点是通过按压次数更改其开合状态，满足体积小巧的要求，并且十分牢固，能够满足弹仓盖在运动中不会因意外而打开的需求。

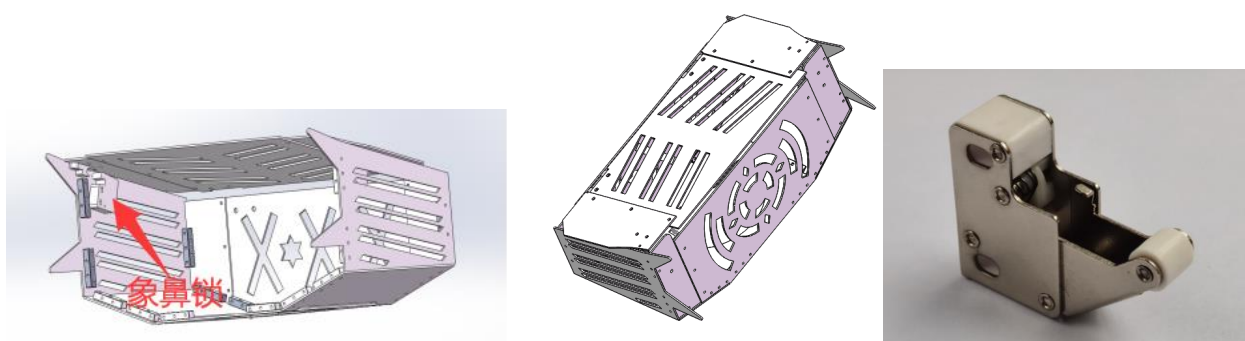


图 4-1-11 弹仓模型及象鼻锁模型

2. 拨盘

第一代拨盘：

本赛季是本队第一次尝试下供弹英雄机器人，因为没有成熟的方案和经验的累积，在敲定方案时选择了看似最简单的中出拨盘。独立拨盘测试时，不存在卡弹问题，拨弹流畅丝滑。但当拨盘与弹路链、弹仓相连后，暴露出了中出下供弹的弊端。在中出拨盘中，电机带动拨轮转动给弹丸的力是斜对于拨弹舱出口的，如图（1），所以每拨一发弹丸都会有一部分力与挡块平衡掉。所以，当不断推送弹丸的时候，由于弹路链中弹丸重量的不断增加，会出现电机输出转矩不够最终造成转不动的现象。同时，由于拨轮的力是斜向上的，当拨轮推送弹丸的时候，会存在弹丸卡在弹路链内壁(在测试过程中确实出现过该现象)，如图（2）。

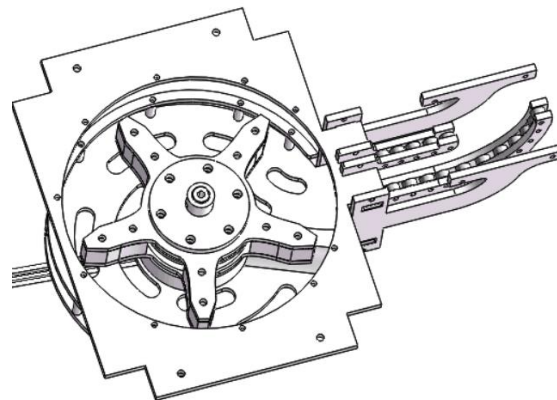
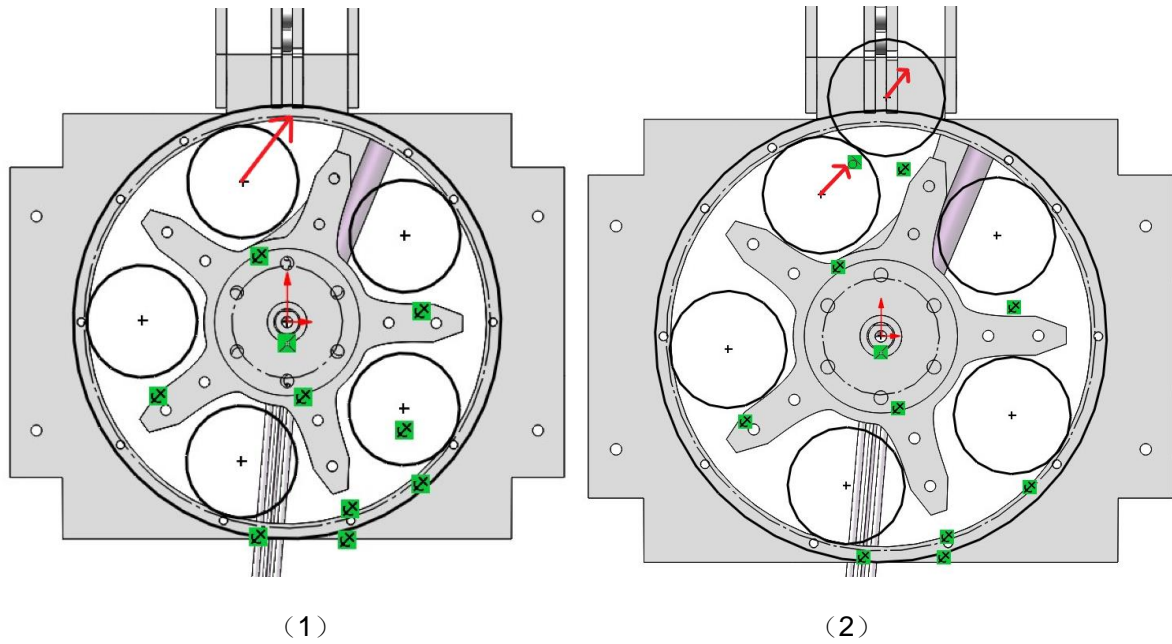


图 4-1-12 第一代拨盘

第二代拨盘：

在前期测试中，发现中出拨盘的弊端之后，通过学习其他高校的开源资料之后就立马调整方向选择更适合下供弹的侧出拨盘。由于前期将大量时间投入到中出拨盘的设计和测试上，另外解决英雄机器人其他方面的问题也同样需要花费大量时间，如果我们重新设计一款侧出拨盘，还需要再次测试卡弹问题，这将花费大量时间。因此在综合考虑了制作成本以及与我们机器人的适配程度等相关因素，我们借鉴了大连理工大学英雄机器人拨盘的设计思路。

在测试拨盘时遇到的问题很多。其中最大的问题是使用 M3 螺丝将拨齿与 M3508 电机连接，使用这种连接会使螺丝受到的剪切力过大最终造成螺丝断在电机轴里面。经过分析，解

决这个问题的关键在于减小螺丝受到的剪切力，我们的解决办法是通过联轴器连接拨轮和电机，让联轴器承受大部分力，避免螺丝断裂的风险。

此外就是卡弹问题，首先就是弹丸会卡在电机连接件与拨盘内壁之间。（如下图）：

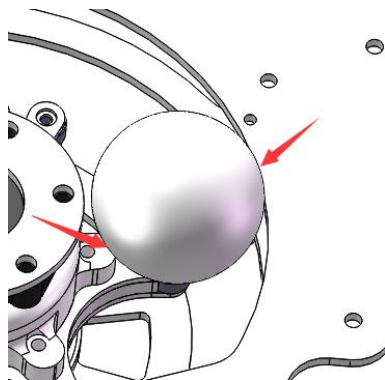


图 4-1-13 卡弹示意图

于是我们改良了电机连接件卡弹部分。（左边为大连理工大学，右边为改良版）

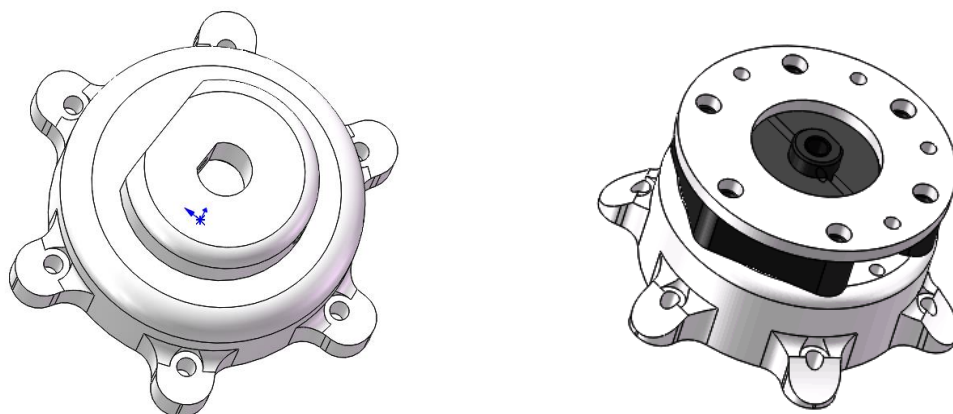


图 4-1-14 电机连接件图

另外一处就是出弹口处，如图（1）：

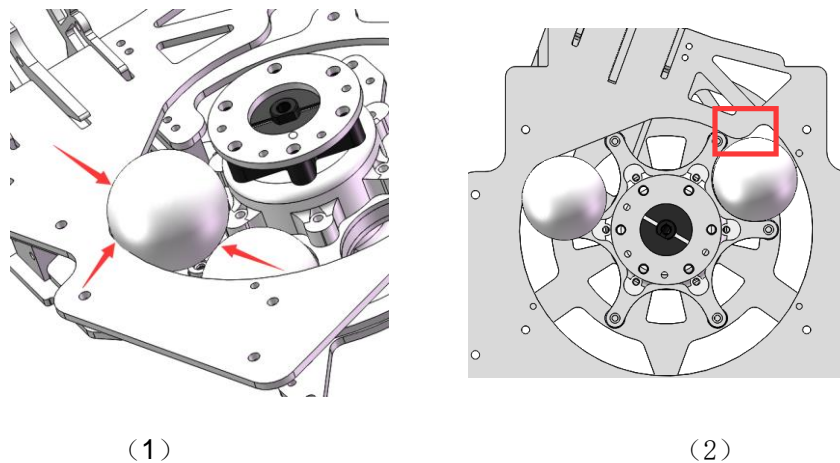


图 4-1-15 卡弹示意图

此处卡弹问题是最难解决的一处，而我们也是将大部分时间用于解决这里的卡弹问题。产生该现象的原因是当电机每次转动一个角度时，产生的力会导致弹丸微微跳起，最终卡死。最初我们只是简单的封住卡弹部位，可实际上并未解决实际问题，大弹丸仍会卡在挡板处。

最后，经过大量的思考、概率统计和测试，我们利用拨盘上板特有形状，如图（2），以此约束大弹丸路径。在拨盘卡死时，可以通过反转拨齿，生成空隙，从而使大弹丸顺利进入拨盘弹路链，解决该问题。但需要注意的是反转角度不宜过大，避免损伤打印件和电机。同时在出口处增加轴承，使弹丸更加顺畅的进入拨盘弹路链内。

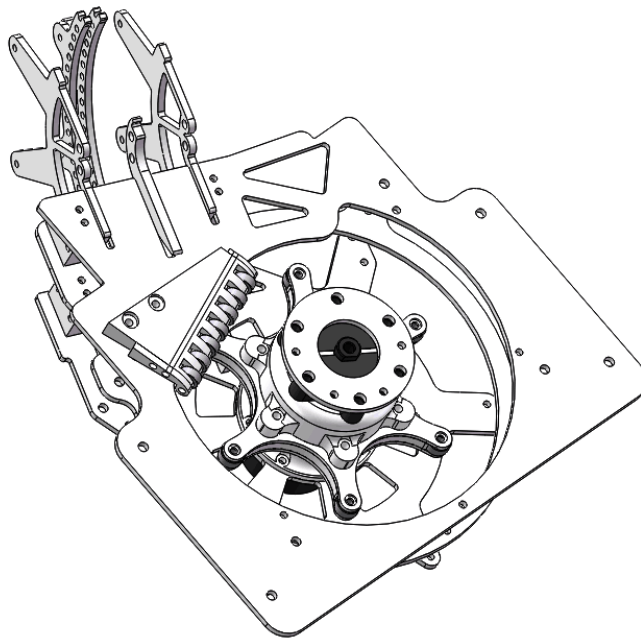


图 4-1-16 第二代拨盘

3. 弹路链及云台支撑架

对于英雄机器人来说，由于我们的英雄机器人采用的是下供弹的供弹方案，因此，在英雄机器人供弹链路就需要保证弹路链的供弹阻力要小以保证拨盘的电机在供弹时能提供相应的力矩。通过参考上海交通大学 2021 赛季英雄弹路链部分的方案，我们使用了四块玻纤板进行堆叠作为供弹链路，同时，通过添加轴承和增大弹路链转角处的曲率半径来减少英雄机器人大弹丸在链路中的阻力。

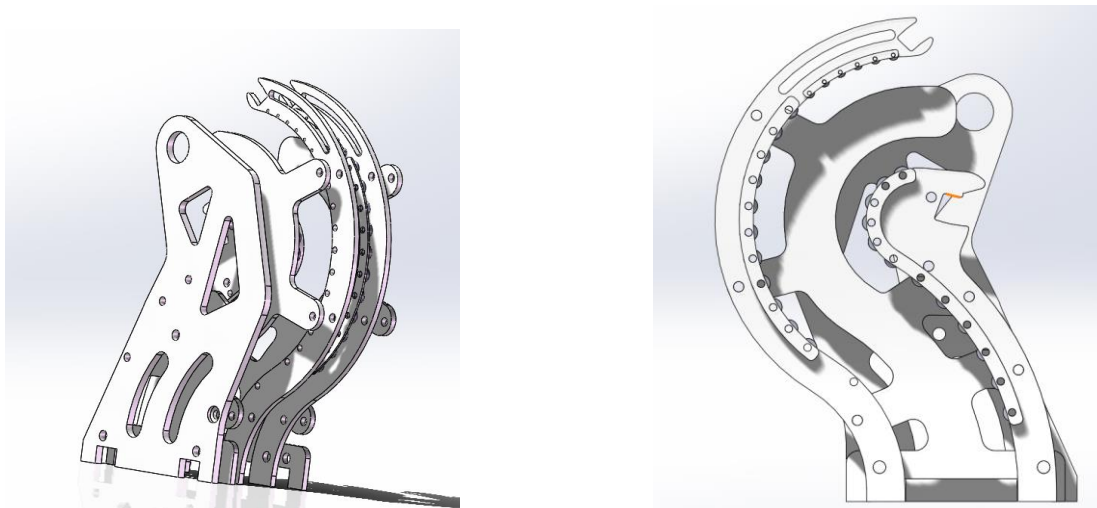


图 4-1-17 弹路链示意图

同时，对于云台的支撑架我们采用了两块玻纤板作为主支持，并通过拓扑算例分析了云台架在英雄机器人上的受力情况，在保证强度和刚度的前提下减小支持板的质量，并确定云台支架最终的形状。



图 4-1-18 拓扑算例分析

4.1.2 工艺选择

英雄机器人整体主要使用了玻璃纤维板材，铝方管及铝件等材料，并通过使用实验室的铣床、车床以及雕刻机等设备进行自主加工来实现对板材的 2D 雕刻和管材的车铣加工。同时，由于队伍的自主加工仍存在着一些局限性，因此，对于英雄机器人上的部分高精度零部件和

复杂零件采用通过外包加工厂进行定制加工的方式以保证零件精度和减轻队伍零件加工的压力。除此之外，由于钣金件存在着精度较低的问题，铝件的价格贵且精度亦有不足，因此，在对英雄机器人的整体设计中应尽量减少钣金件和铝件的使用。必要的钣金件也应当采用外包加工厂的方式来进行代加工。在英雄机器人实际的装配和迭代过程中，对于一些存有疑问板材和零件，可以先使用亚克力板和 3D 打印件来进行测试，在测试合适后再选择是否更换材料。

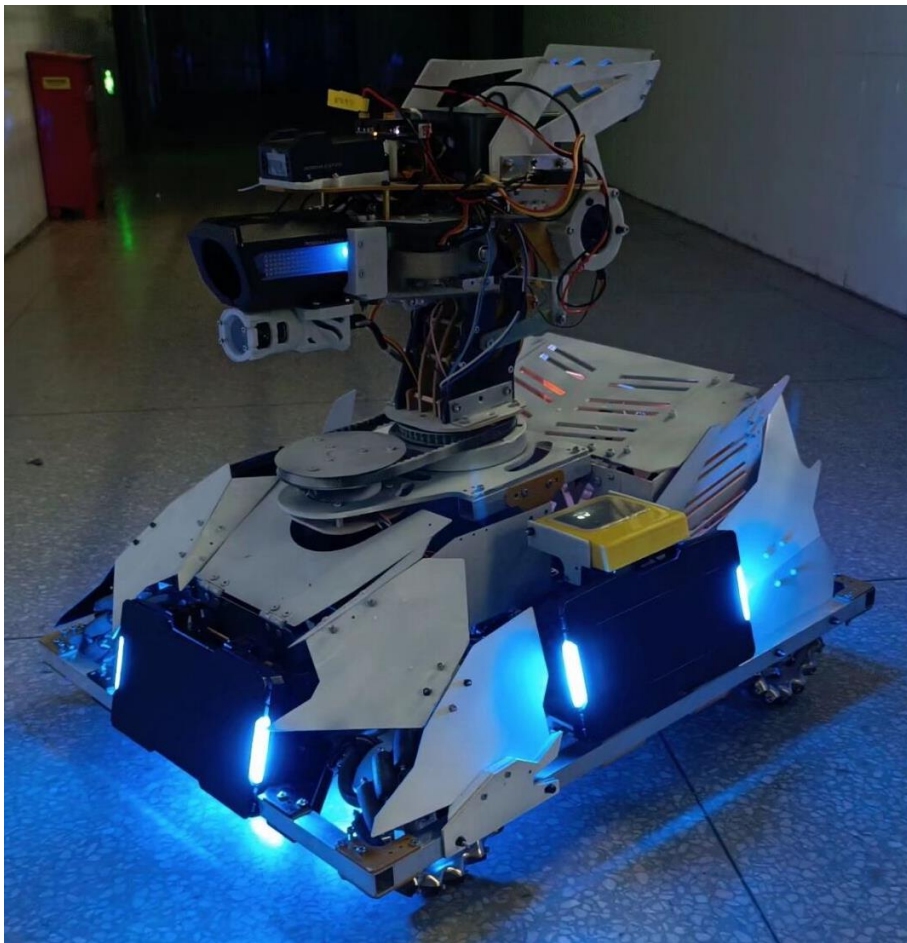


图 4-1-19 英雄实物图



图 4-1-20 2D 雕刻机



图 4-1-21 2D 雕刻成品



图 4-1-22 3D 打印机



图 4-1-23 3D 打印成品

4.1.3 模块的定位及走线

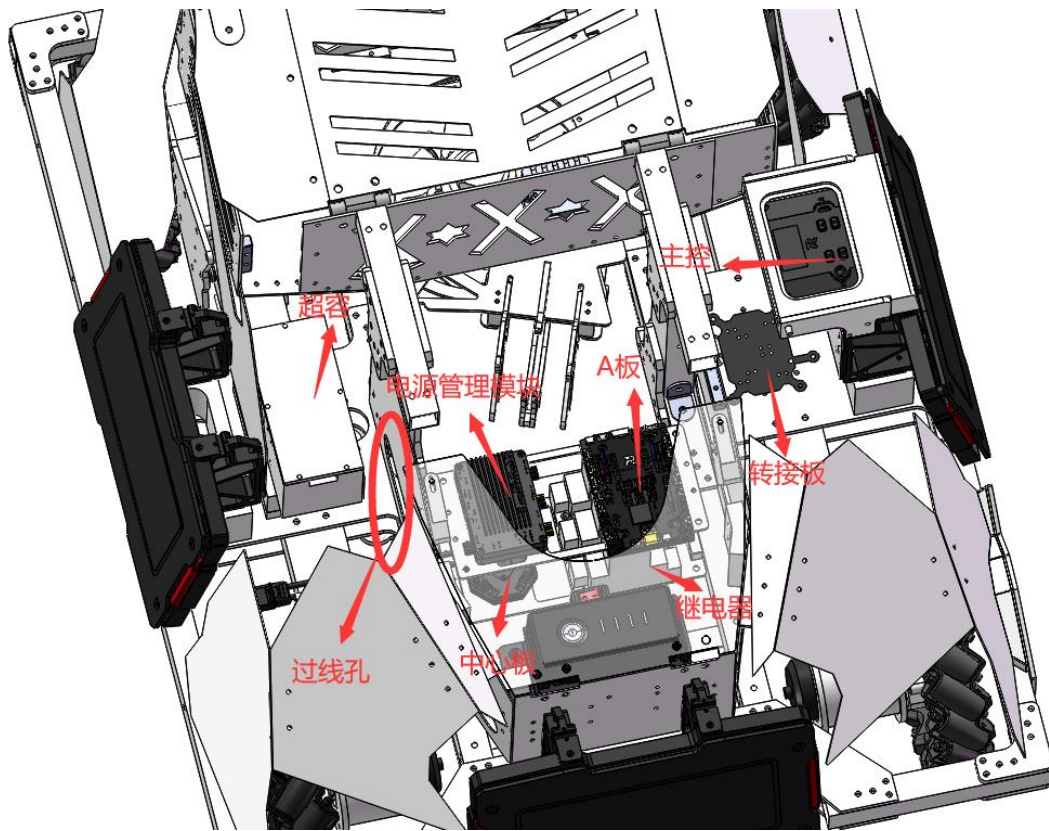


图 4-1-24 模块定位走线示意图

模块定位和布线的最低要求是布局合理，互不干涉，在这一基础上再进行布线美观的设计。

英雄机器人弹仓较大且置于底盘后半部分，则将模块布局底盘前半部分。超级电容与电池因其体积较大，分别放置于装甲板后方空位，而主控模块要求方便读取和操作，则配合打印件保护壳放置于另一块装甲板后侧。

此外为方便电控组成员接线，将底盘前半部分空间划分为上下两层放置电源管理模块和A板等核心模块；同时为避免底盘前半部分堆积过多线材，于是将转接板放置在侧边。并在保护壳两侧分别留出过线孔，方便电控组成员过线。

4.2 硬件设计

4.2.1 整机硬件框图

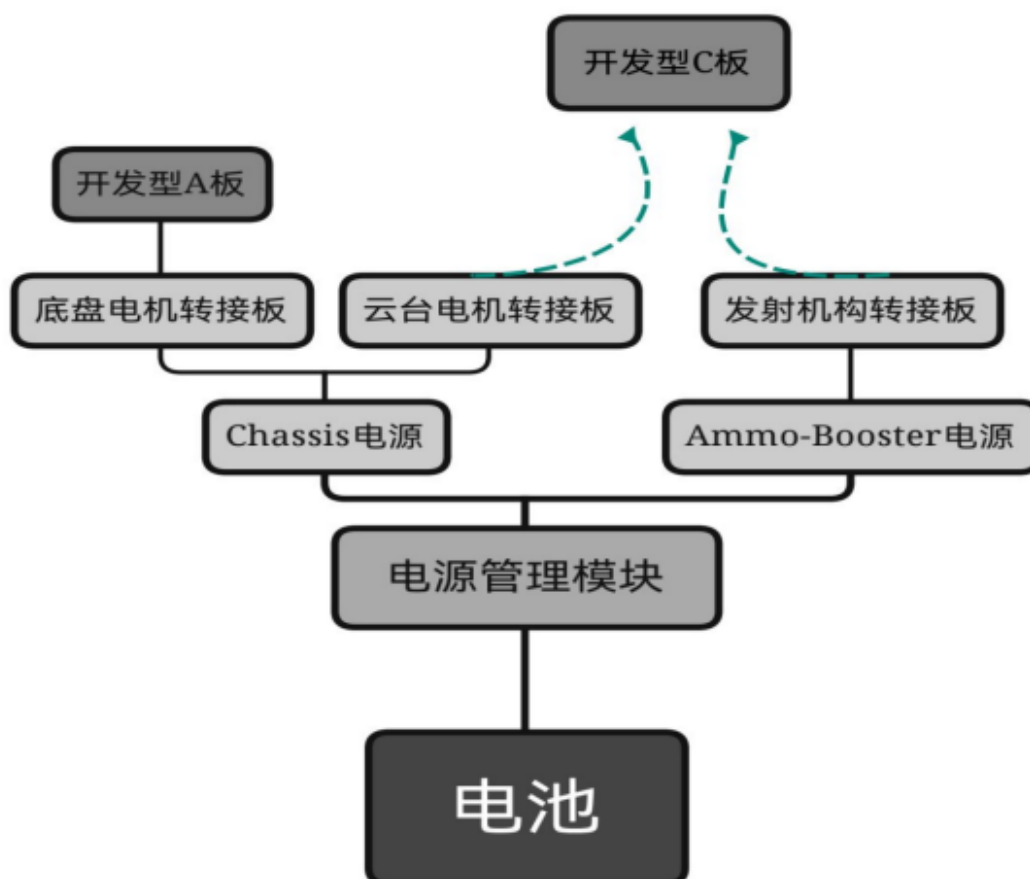


图 4-4-1 整机硬件框图

4.2.2 硬件详细设计

4.2.2.1 背景介绍

在 RM 赛场上，步兵机器人和英雄机器人的底盘运动受到了规则的限制，极大影响了机器人在赛场的活动能力。为了提高机器人的灵活性和生存能力，本队伍最初采用了溪地雾列控制的超级电容模组以适应赛场需求，但由于其较简单的控制电路存在电容有死电的缺陷，

并且无法保证稳压输出，导致在赛场的效果不佳。为了在 2022 赛季中取得更好的表现，本队伍在借鉴大连理工大学凌 BUG 战队超级电容方案的基础之上对超级电容进行研发。

4.2.2.2 设计思路

设计新一代超级电容控制方案，将电源和底盘并联并通过 dc/dc 连接超级电容实现电容的充放电。通过双向升降压实现输出电压为底盘电压，让底盘电机调正常工作；超级电容主控对裁判系统进行解析和对电源输入，管理超级电容，底盘输出进行电压和电流采样，运用采样电路电气参数进行内部运算闭环输出高精度 PWM 信号控制 BUCK -BOOST 电路达到升降压，自动实现能量闭环，实现能量的最大利用。

拓扑结构如下所示：

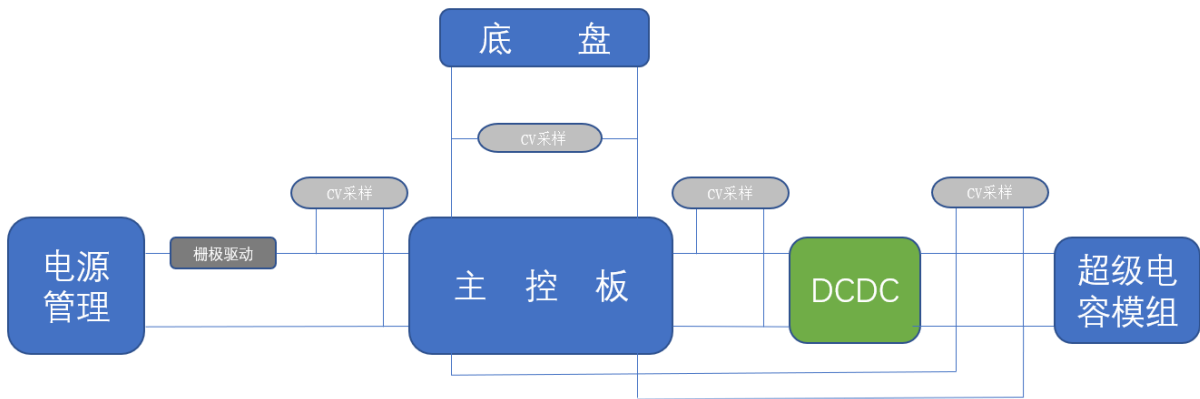


图 4-4-2 拓扑结构示意图

控制逻辑：

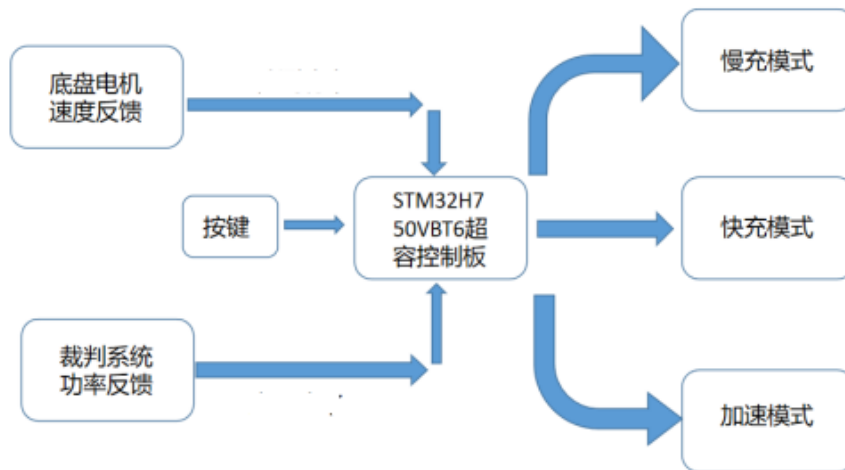


图 4-4-3 控制逻辑

4.2.2.3 充放控制

• 充电控制

在未开启加速模式时，电源管理模块同时向底盘和超级电容供电，输入功率根据相应等级实时更新，在满足底盘输出功率的基础上将剩余功率作为对超级电容的充电功率。在充电过程中 dc/dc 控制电路采用 cc 模式达到恒功率输入的目的。



图 4-4-4 充电控制图

• 放电控制

在开启超级电容后，底盘输出功率大于电源管理模块输入功率，此时超级电容和电源管理模块同时作为电压源向底盘供电。



图 4-4-5 放电控制图

4.2.2.4 升降压变换

通过 dc/dc 升降压电路对超级电容进行充电与放电，如下图，我们采用双向同步整流 BUCK-BOOST 电路在同一方向实现升降压功能，对应于电源管理模块对超级电容的降压充电及超级电容对底盘的升压放电功能。通过主控板输出控制的高精度 pwm 波控制 MOS 管开关占空比实现精准的升降压比例。当控制 MOS 管在 A, B 状态之间变换时 dc/dc 是处于降压模式；当 MOS 管在 B,C 状态之间切换式，电路处于升压模式。

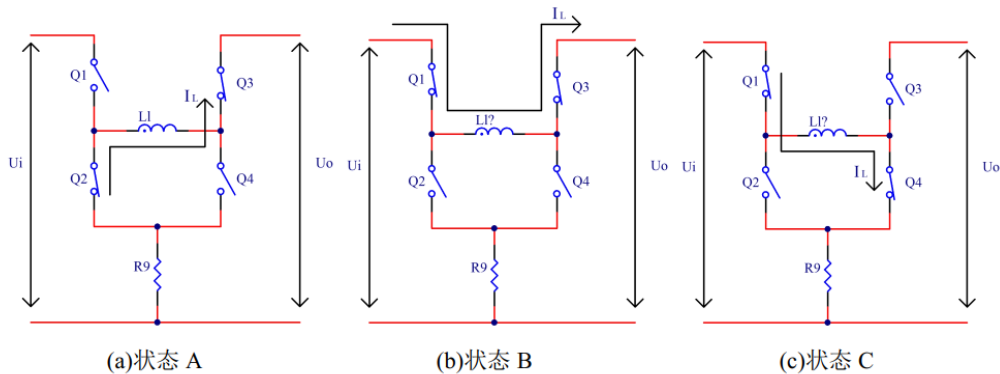


图 4-4-6 升降压电路

BUCK 电路电压增益公式： $U_0 / U_{ix} = D_{buck}$ ；

BOOST 电路电压增益公式： $U_0 / U_{ix} = 1 / (1 - D_{boost})$ ；

由上可推得 BUCK-BOOST 电路电压增益公式： $U_0 / U_{ix} = D_{buck} / (1 - D_{boost})$ ；

4.2.2.5 实物展示

• 超级电容模组

超级电容选择的是 2.7V 50F 的牛角电容，10 个串联在一起形成超级电容组，超级电容组的容量是 27V 5F。控制板实物如图所示：

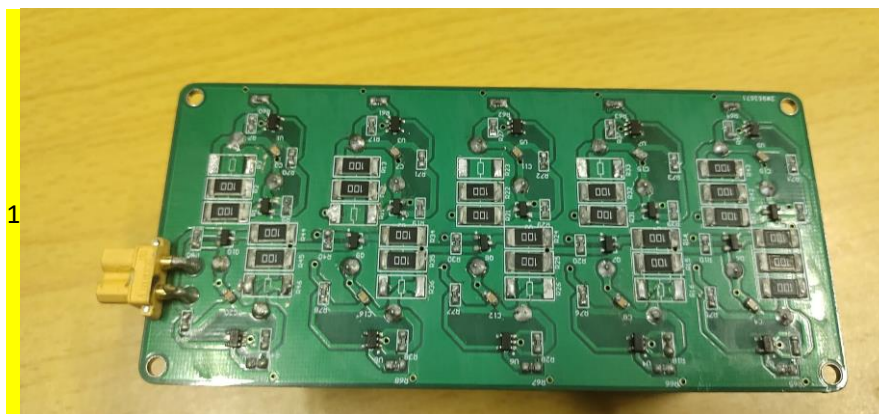


图 4-4-7 超级电容控制板

4.2.2.6 电路设计

• 超级电容模组电路

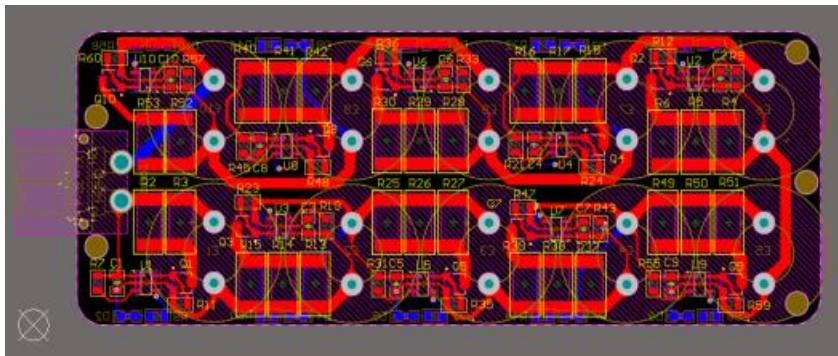


图 4-4-8 超级电容模组电路图

• 超级电容控制板电路

本超容控制方案选用四层 PCB 作为超容控制电路，作为 ADC 采集的数据及裁判系统数的处理和超容控制电路的控制。

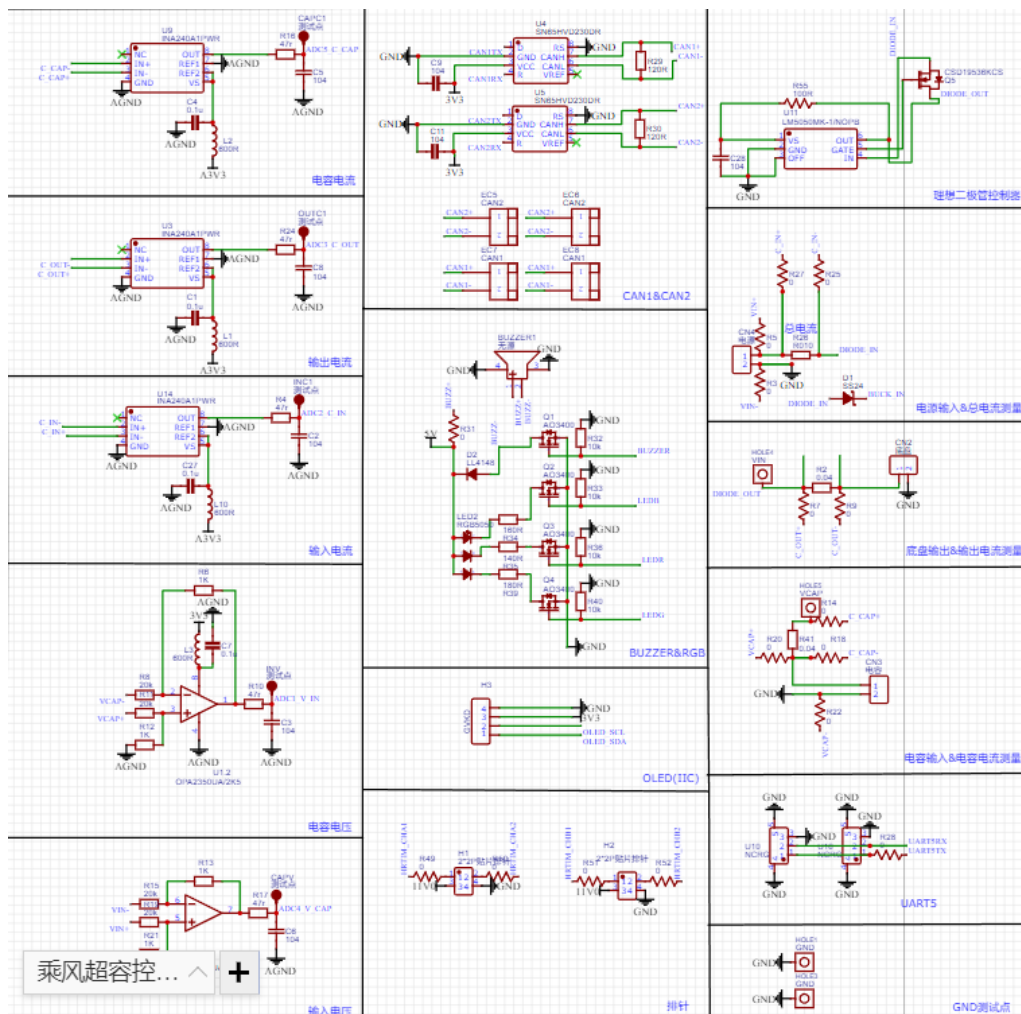


图 4-4-9 超级电容控制板电路图

• 超级电容功率板电路

铝基板作为功率控制板，用于承载功率器件。

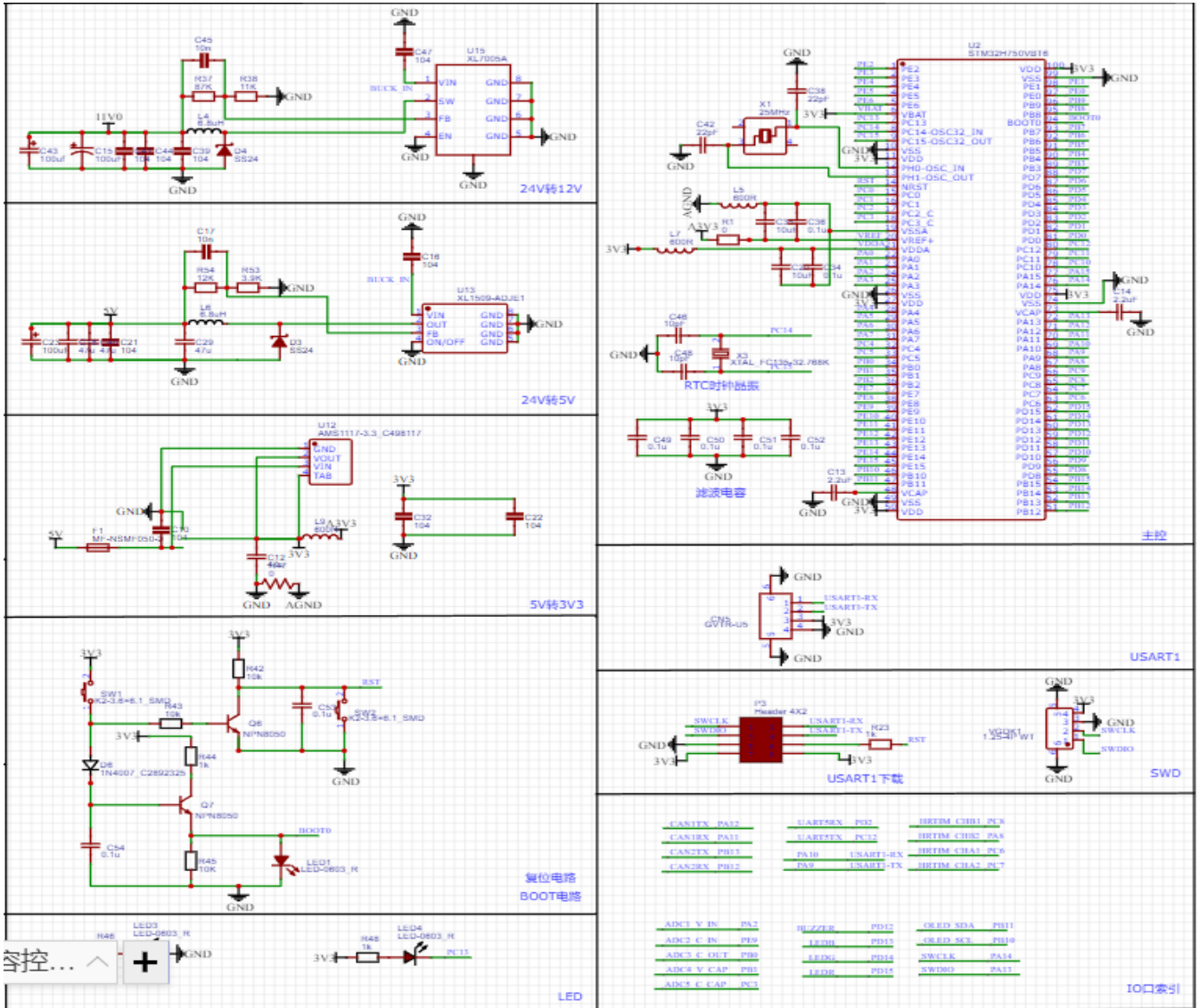


图 4-4-10 超级电容功率板电路图

4.2.2.7 超容控制板

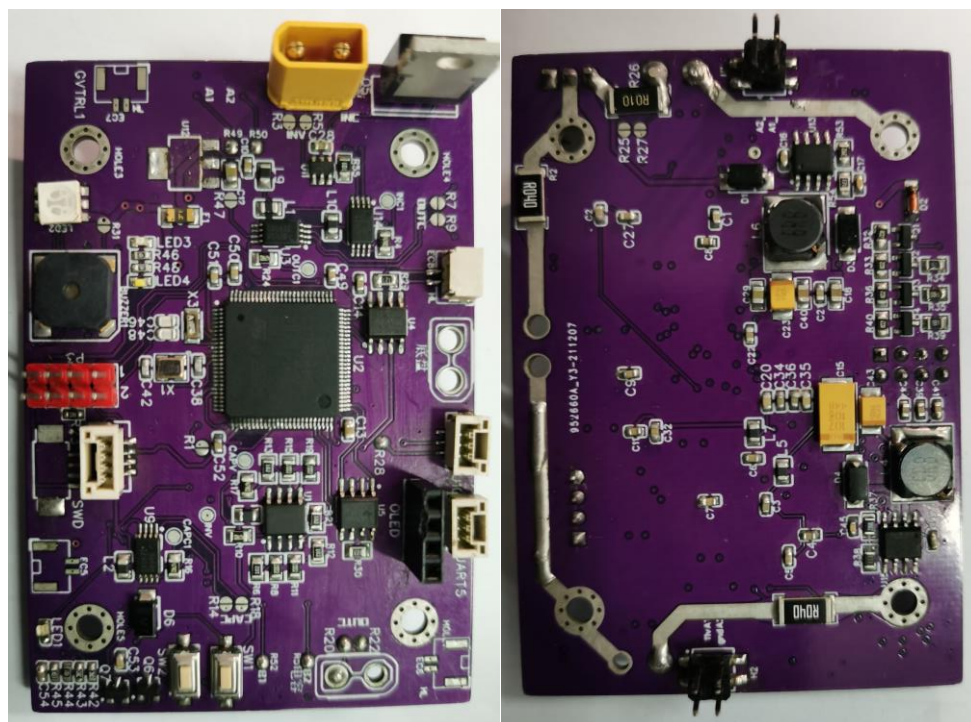


图 4-4-11 超容控制板实物图

4.2.3 关键器件的选型

4.2.3.1 主控芯片

超级电容控制板是超级电容管理模块的核心计算单元，我们采用了意法半导体公司的 STM32H750VBT6 作为主控芯片，STM32H750VBT6 的频率可达 480M，并且提供 HRTIM 高精度 PWM 输出满足了对 MOSFET 的控制。



图 4-4-12 主控芯片实物图

该芯片采用 Cortex-M7 内核，具有六级双发射超标量流水线及分支预测器、单精度及双精度浮点计算单元、紧密耦合内存 TCM、高速缓存 L1 cache、DSP 及 MPU 单元。当其工作在 480MHZ 的主频下时，可以达到约 1027DMIPS 的计算能力，足以满足超级电容控制板对于功率检测、功率校验及功率控制的计算要求。

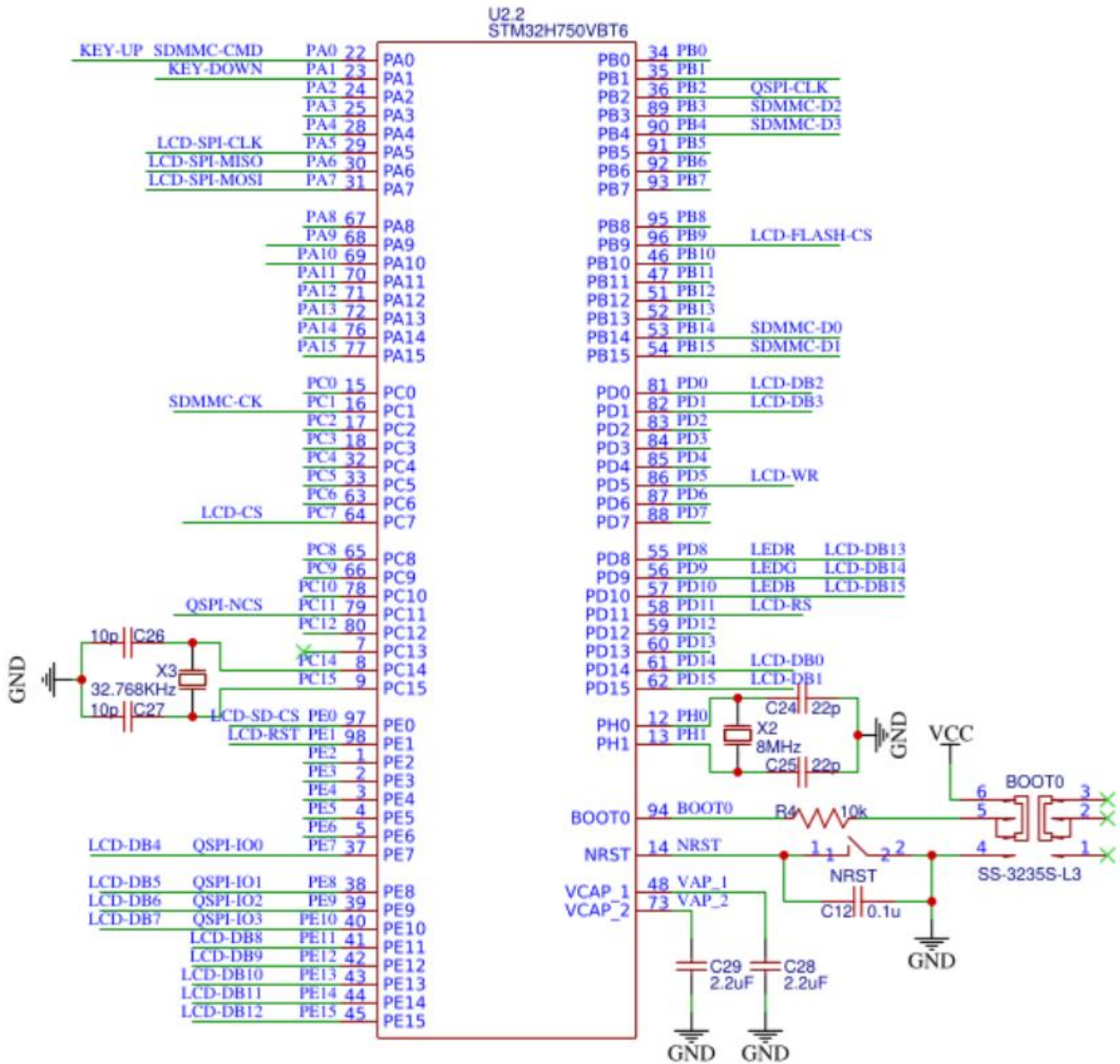


图 4-4-13 主控芯片

4.2.3.2 升压稳压器

在超级电容输出端口串接升压稳压器，使供给给底盘的电压趋于稳定，防止因电压过低导致 C620 电调开启低压保护造成底盘失能。其最大工作电流可达 10A,输出电压可在 1.25-30V 间调节，可满足底盘正常工作电压需求。它大大提高 DC/DC 变换器的效率并且不存在由肖特基势垒电压而造成的死区电压。

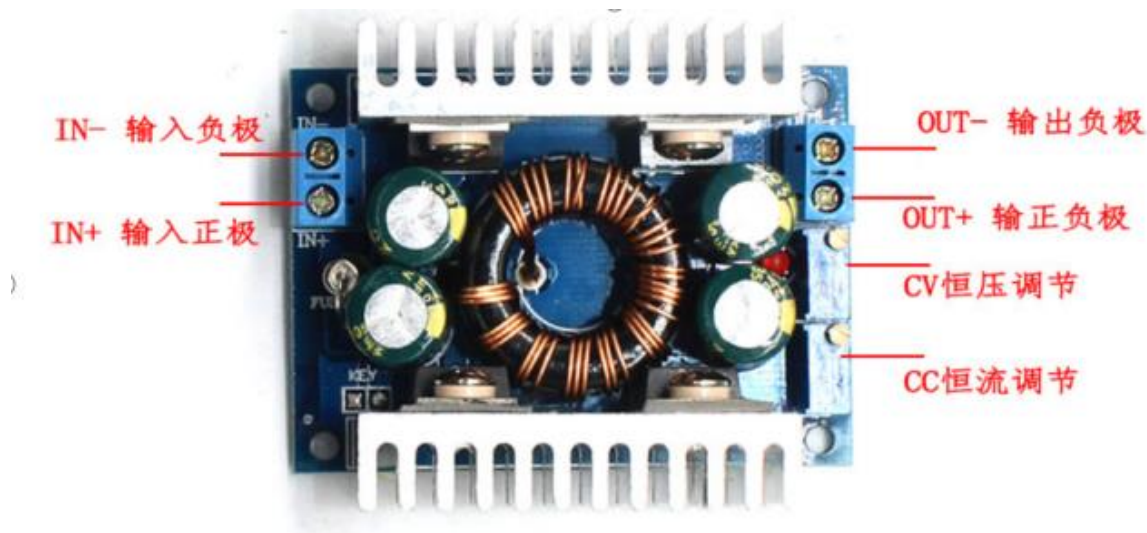
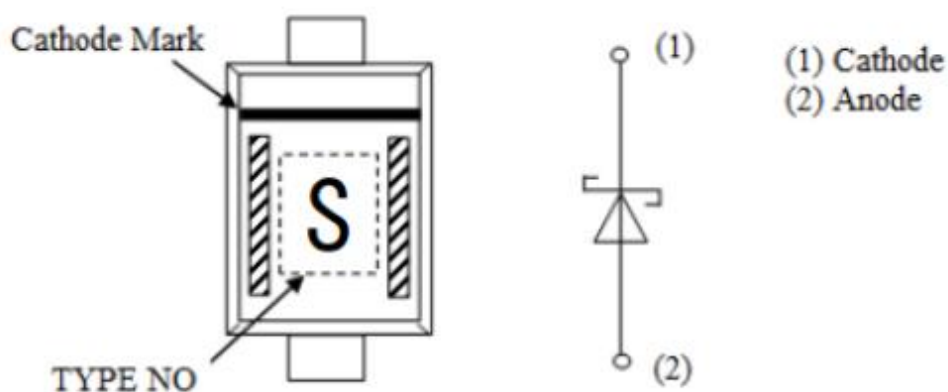


图 4-4-14 升压稳压器

4.2.3.3 快恢复二极管

在超级电容输出端口正接一个快恢复二极管，防止因底盘急刹产生电流倒灌。



「S」 is the sign which shows RB521SU-40.
There is dot on the package which tells production year,
week and its product manufacture.

图 4-4-15 快恢复二极管图

4.3 软件设计

● 最终方案:

相比于比赛前的软件设计，该方案是针对于在赛场后遇到的问题之后进而对一些逻辑方面的一些修改，软件设计的整体框架和流程并没有多大的改变。

在本次的比赛当中，遇到的问题主要为四个方面，第一个是射击的弹速问题，第二个为 Yaw 轴电机疯转问题，第三个为超级电容供电和控制问题，第四个为 C 板控制板上电慢的问题。

首先，针对于弹速的问题，该问题出现的原因主要是因为两个方面，一是因为我们英雄机器人的摩擦轮用了很久，一直没有更换，导致摩擦轮中部凹陷，使得弹速一直升不上去，二是因为官方的子弹和原版的子弹的不匹配，当时没有对弹丸进行租界，导致弹丸的弹速偏差较大，从而使得 UI 的基准线不准的问题。所以只能在比赛是临时更换摩擦轮，而且在上场的准备时间对其的弹速进行修改，而基准线只能稍作修改。

其次的 Yaw 轴的疯转问题，其主要出现的原因的关于底盘跟随云台的 PID 的调节问题，当英雄机器人因为外界因素导致云台和底盘的中性轴偏差过大，底盘不能及时跟随云台，从而使得云台电机超调，进而导致 Yaw 轴电机疯转，所以在这我们将原来的单环控制改为了双环控制，使其更加的稳定。

关于超级电容的问题，主要更改的地方在于它的控制方面，因为比赛场地的起伏路段和地面的摩擦轮的问题，所以我们主要是根据比赛的电容电量的消耗来对其进行修改，尽可能的避免出现供电不足的问题。然后就是后期打到第三场的时候会出现超级电容电量耗尽的现象，到了第三场之后，英雄机器人出现明显供电不足的问题，因为该问题出现在最后几场，所以留在下一届解决。

C 板上电慢是一个比较严重的问题，主要是体现在英雄机器人死亡然后复活后，它因为某种原因从而需要半分钟左右才能进行一个供电，平时上电也会出现这个问题，不过因为是有准备时间，所以在非死亡的时候影响不大，该问题我觉得可能是由于双板控制导致的，一个赛季下来，我认为两块核心板对英雄机器人进行控制，在非必要的情况下还是不太建议，会出现很多关于信号的问题，所以在下一届应该不会考虑双核心板控制。

4.3.1 系统架构

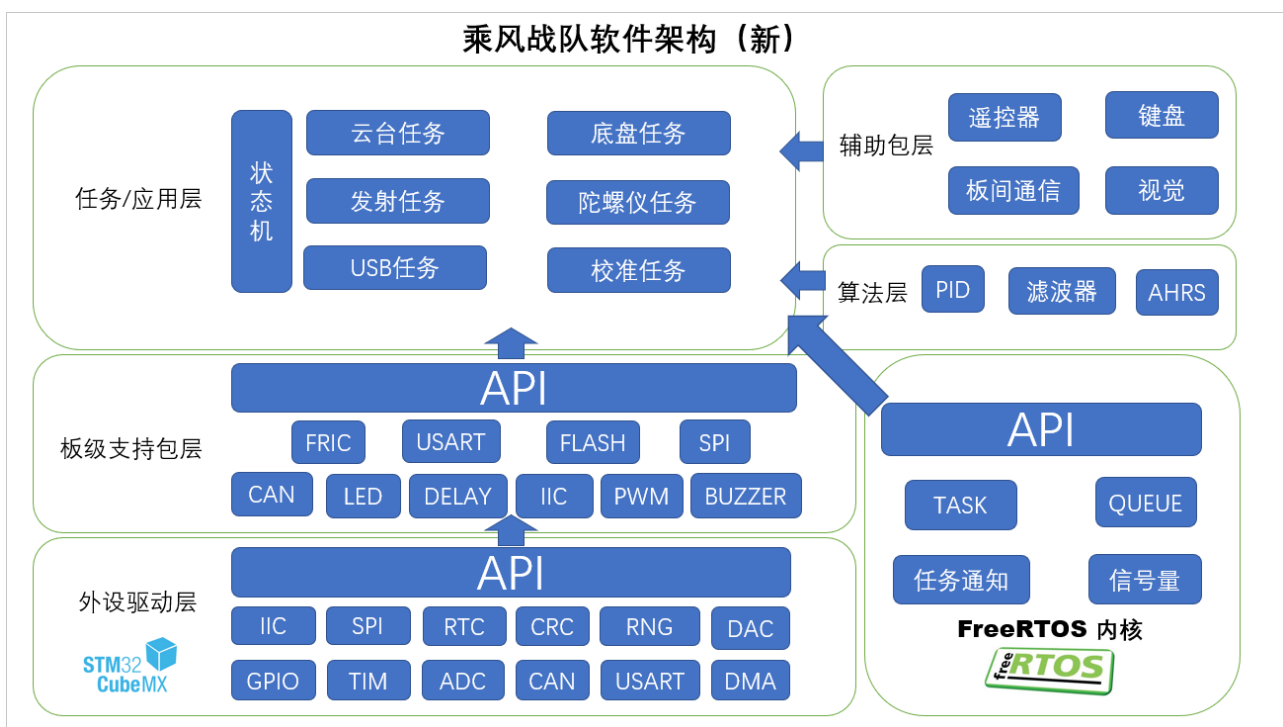
开发环境：WIN10

编译器： Keil5 v5.36

调试器： J-Link v9、CMSIS-DAP 、ESP8266-CMSIS-DAP

今年采用队内统一制定的系统架构，RTOS 采用 FreeRTOS。外设驱动层由 STM32 CubeMX 生成，板级支持包（BSP）为上层提供接口调用，为了避免耦合，我们将算法层包含了卡尔曼、PID、AHRS 等算法，为任务/应用层提供接口。我们把一些辅助性的任务均放在辅助包层，也是为应用层提供接口。

为了方便开发,在中间件中我们使用了 STM32 的 USB 驱动文件,由 CubeMX 生成导入。在辅助包层也使用了开源的 FIFO 库和 CRC8-CRC16 库。



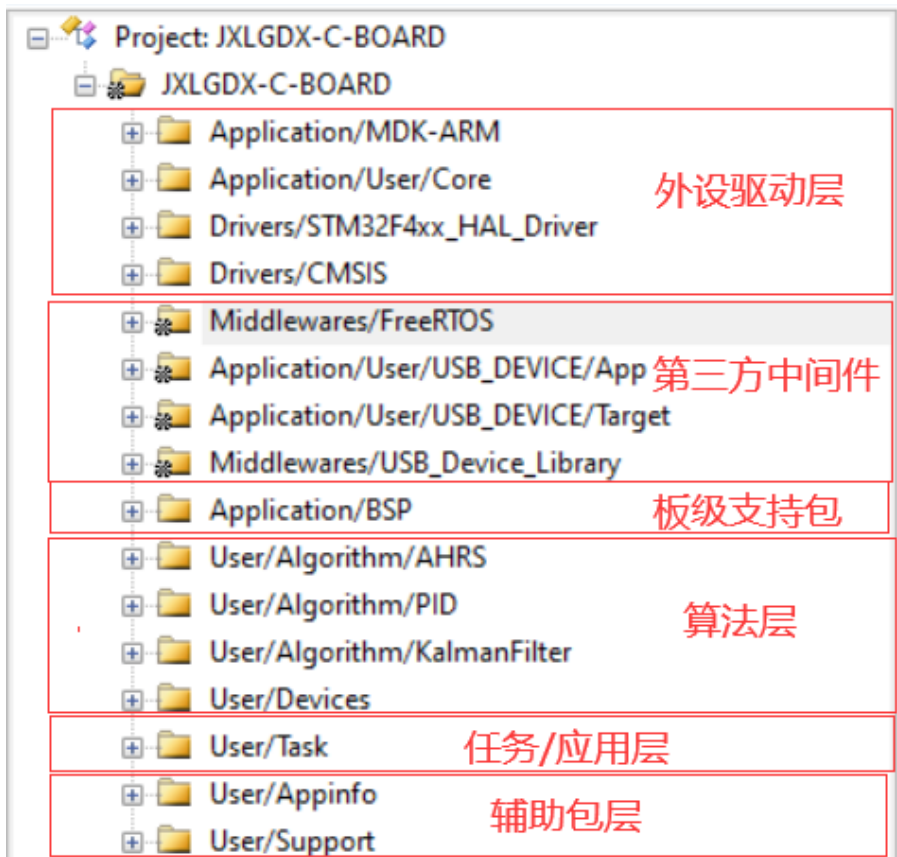


图 4-2-1 系统架构图

我们与视觉的通信协议，参考官方学生串口协议进行改写。

江西理工大学乘风战队电控一» 视觉协议		版本: V2.1
数据协议格式		
SOF: 0xA5	CMD_ID	CRC8校验 Data数据部分 CRC16校验
帧格式说明		
SOF: 0xA5	固定帧头: 0xA5	
CMD_ID	0x01	自瞄模式
	0x02	大风车模式
	0x03	保留
CRC8校验	1-byte	
Data数据部分	CMD_ID:0x01	
	Robot_ID	机器人ID
	Speed	弹丸速度
	Pitch	Pitch角度
	Yaw	Yaw角度
	CMD_ID:0x02	
	Robot_ID	机器人ID
	Speed	弹丸速度
	Pitch	Pitch角度
	Yaw	Yaw角度
CRC16校验		

图4-2-2 视觉协议

4.3.2 运行流程

英雄机器人的任务运行主要是基于 FreeRTOS 的 SVC、PendSV 和 SysTick 进行任务的启动和切换。其中 SVC 用于任务的启动，能够让用户程序使用 SVC 发出对系统服务函数的呼叫请求，以这种方法来调用它们从而间接的访问硬件，产生一个 SVC 异常，进而使任务启动。PendSV 则用于完成任务的切换，不仅是因为它可以像普通的中断一样被挂起的，而且是由于它的最大特性是如果当前有优先级比它高的中断在运行，PendSV 会延迟执行，直到高优先级中断执行完毕，这样子产生的 PendSV 中断就不会打断其他中断的运行，进而完成任务的切换。最后 SysTick 则是用于产生系统节拍时钟，提供一个时间片，如果多个任务共享同一个优先级，则每次 SysTick 中断，下一个任务将获得一个时间片。具体的则是由我们在软件架构中的外设驱动层进行任务的创建，并且对任务的堆载、优先级等进行配置，之后在任务/应用层进行代码的编写和测试。

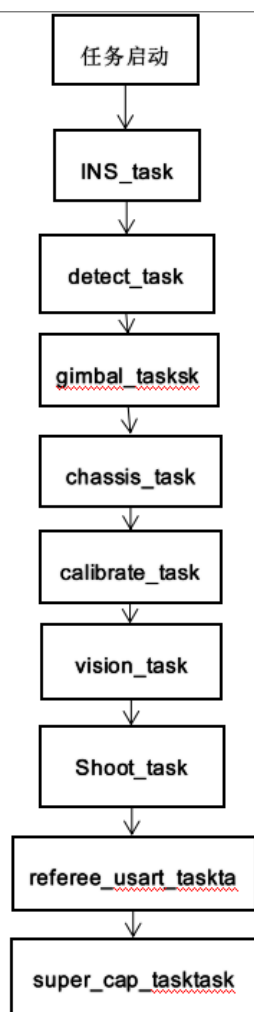


图 4-2-3 应用层任务启动流程

应用层任务名	内容介绍	优先级
INS_task()	主要是利用陀螺仪 bmi088, 磁力计 ist8310, 完成姿态解算, 得出欧拉角, 提供通过 bmi088 的 data ready 中断完成外部触发, 减少数据等待延迟, 并且通过 DMA 的 SPI 传输节约 CPU 时间	3
detect_task()	检测错误任务, 通过接收数据时间来判断。并且提供检测钩子函数, 错误存在函数	0
gimbal_task()	完成云台控制任务, 采用大疆提供的九轴陀螺仪提供角度。云台控制分为小陀螺, 自动跟随, 正常模式这三种模式。	2
calibrate_task()	校准陀螺仪的值	2
shoot_task()	完成发射机构的控制任务, 控制摩擦轮电机和拨盘电机	2
chassis_task()	完成底盘控制任务, 控制底盘的四个电机完成行走	0
vision_task()	完成自瞄的控制任务, 接受视觉发送过来的值, 并且发送视觉所需要的值	2
referee_task()	RM 裁判系统数据处理	0
super_cap_task()	完成超级电容的控制任务	0

4.3.3 重点功能

1. 云台控制设计

根据 GM6020 电机的特性以及陀螺仪的加入，我们设计了角度-速度 PID 进行双环控制。我们在第一环角度控制器中的角度反馈被分为电机编码器反馈和 IMU 反馈的角度值。在第二环速度控制器中的速度反馈被分为电机编码器反馈和 IMU 反馈的速度值。根据选用模式不同，我们角度反馈值和速度反馈值的来源也不同。具体运行逻辑如下：

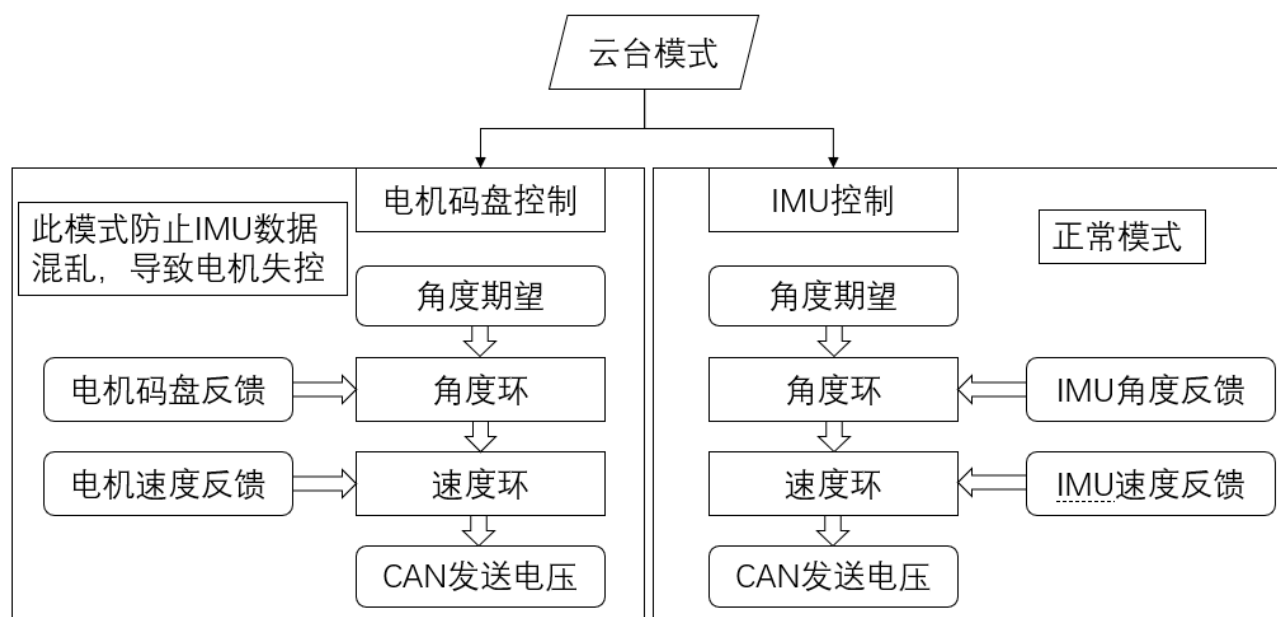


图 4-2-4 云台运行逻辑图

在常规模式下，默认情况下开启 IMU 控制，而电机编码控制的引入是为了防止 IMU 数据混乱导致的失控。我们这里还是以 IMU 控制为例，解释我们双环 PID 的设计。在赛季初期，我们采用的是最为常规的串级 PID，但是效果不尽理想。电机反馈的角度和速度的分辨率都远不及 IMU 反馈的高，我们在设计时，着重在 IMU 双环的设计上。常规角度环使用微分补偿相位，其动态效果具有一定的滞后性。在 PID 调参阶段容易产生超调，从而引起在云台震荡。为此我们在角度环上微分先行，采用 IMU 反馈的角速度作为微分项。此法在 Pitch 轴上效果颇佳，但是 Yaw 轴存在较大的转动惯量会引起较大的变化。故我们必须在 Yaw 上的微分系数进行一定增益补偿。设计的串级 PID 公式如下，其中 $error_delta$ 为 IMU 反馈的角速度， $e_{角度}(k)$ 为当前角度与角度期望的差值。

$$u_{\text{角度}}(k) = k_{p1} * e_{\text{角度}}(k) + k_{i1} * \sum_{i=0}^k e_{\text{角度}}(i) + k_{d1} * \text{error_delta}$$

$$e_{\text{速度}}(k) = u_{\text{角度}}(k) - \text{error_delta}$$

$$u_{\text{速度}}(k) = k_{p2} * e_{\text{速度}}(k) + k_{i2} * \sum_{i=0}^k e_{\text{速度}}(i) + k_{d2} * [e_{\text{速度}}(k) - e_{\text{速度}}(k-1)]$$

经过上述公式计算过后的 $u_{\text{速度}}(k)$ 就是发送给云台电机的电压值，我们需要对 k_{p1} 、 k_{i1} 、 k_{d1} 、 k_{p2} 、 k_{i2} 、 k_{d2} 进行参数的整定。经过整定后的效果已经可以满足比赛要求，云台效果也比较明显。在实验过程中，我们从 RobotPilots 战队开源的步兵代码得到启发，将 $e_{\text{角度}}(k)$ 的数据进行一阶卡尔曼滤波(公式如下)。

$$\text{状态方程: } X(k|k-1) = A * X(k-1|k-1) + B * U(k) + W(k)$$

$$\text{观测方程: } P(k|k-1) = A * P(k-1|k-1) * A^T + Q$$

$$\text{更新卡尔曼增益: } Kg(k) = \frac{P(k|k-1) * H^T}{H * P(k|k-1) * H^T + R}$$

$$\text{修正估计值: } X(k|k) = X(k|k-1) + Kg(k)[Z(k) - H * X(k|k-1)]$$

$$\text{更新误差协方差: } P(k|k) = [1 - Kg(k) * H] * P(k|k-1)$$

运用在 IMU 控制上，云台的抗干扰效果和响应速度均成倍提高。此时云台的 PID 参数的整定变得十分简单，效果比较明显。为了进一步的提高 Pitch 轴云台的控制效果，我们在上述整个流程下，加入前馈控制，以提高云台的响应速度。整定前馈参数，通过记录云台在每个角度稳定的所需电压值，然后通过拟合的方法，计算出云台的前馈方程。最终，我们的云台控制设计就到此结束。

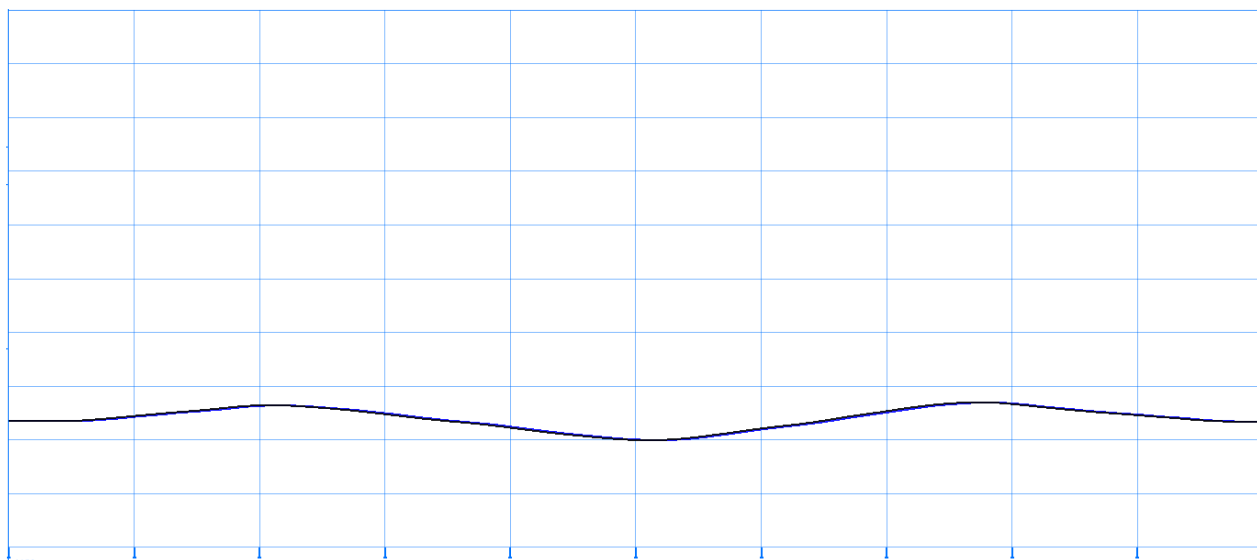


图 4-2-5 云台 Yaw 轴当前值和设定值曲线

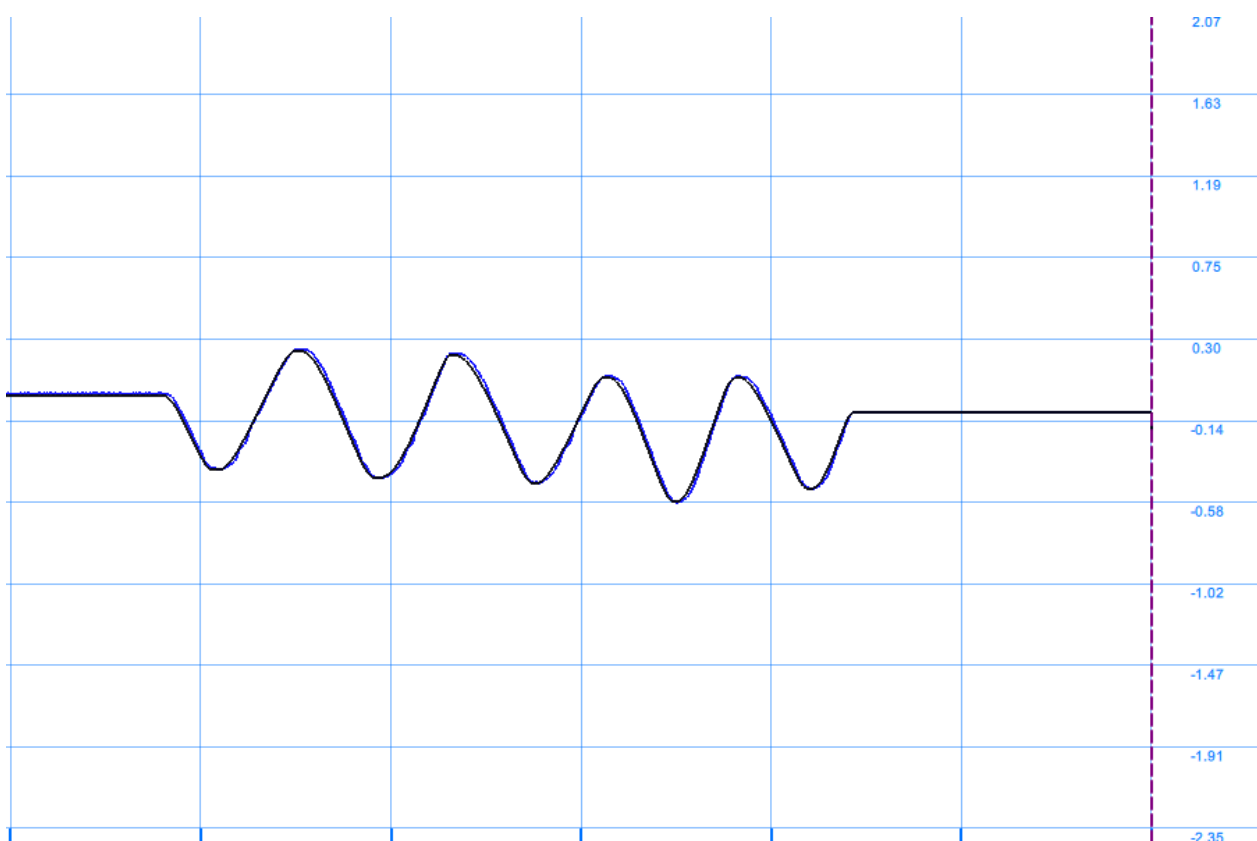


图 4-2-6 云台 Pitch 轴当前值和设定值曲线

2. 键盘鼠标任务

由于客户端鼠标移动是反馈速度，我们如果不经过处理直接赋值在战车上，会出现一卡一卡的移动。这是由于每次发送的速度值的差值变化幅度大，我们要需要经过一定的平滑处理外加一点预测。为此，我们需要先将原始数据进行一阶卡尔曼滤波，然后再经过一个长度 70 的滑动窗口滤波。

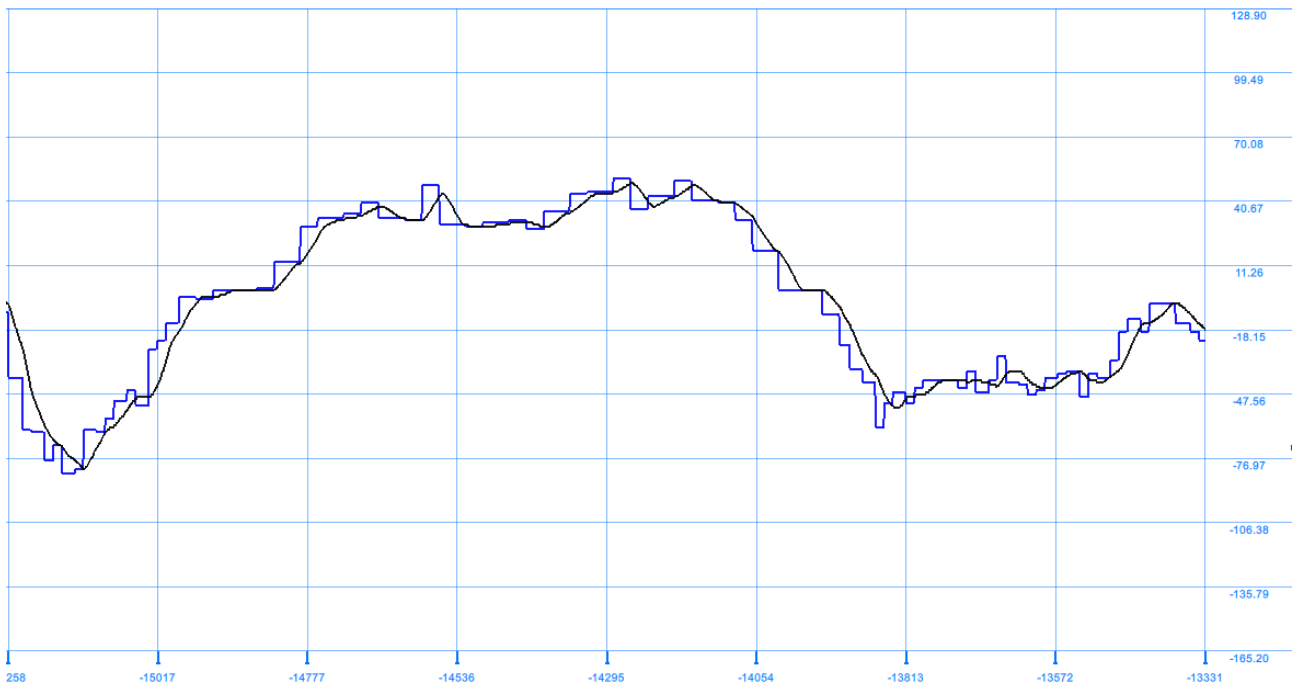


图 4-2-7 鼠标 X 方向修改前后对比

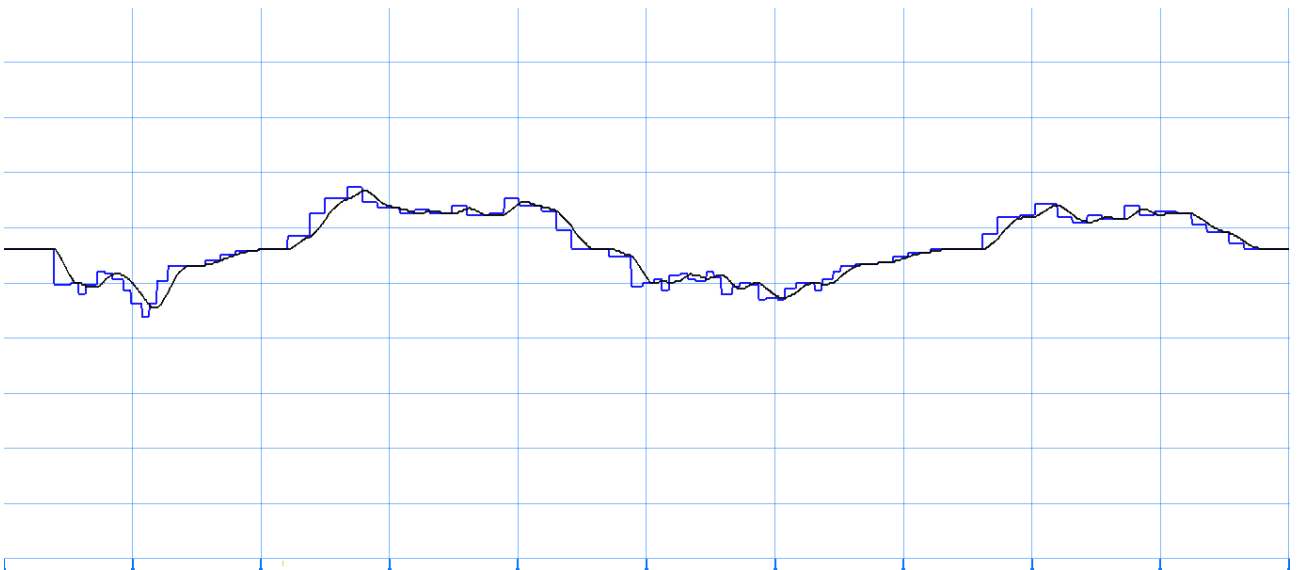


图 4-2-8 鼠标 Y 方向修改前后对比

我们还需要对“WSAD”这四个键进行一个映射，将其映射成遥控器的摇杆。我们需要设计一个三段的斜坡函数。在第一段的斜率低于第二段的斜率，第三段为下降的斜率，其远远大于前两端，使其能快速刹车。并且我们还为其他键盘按键设计了一套判断逻辑。

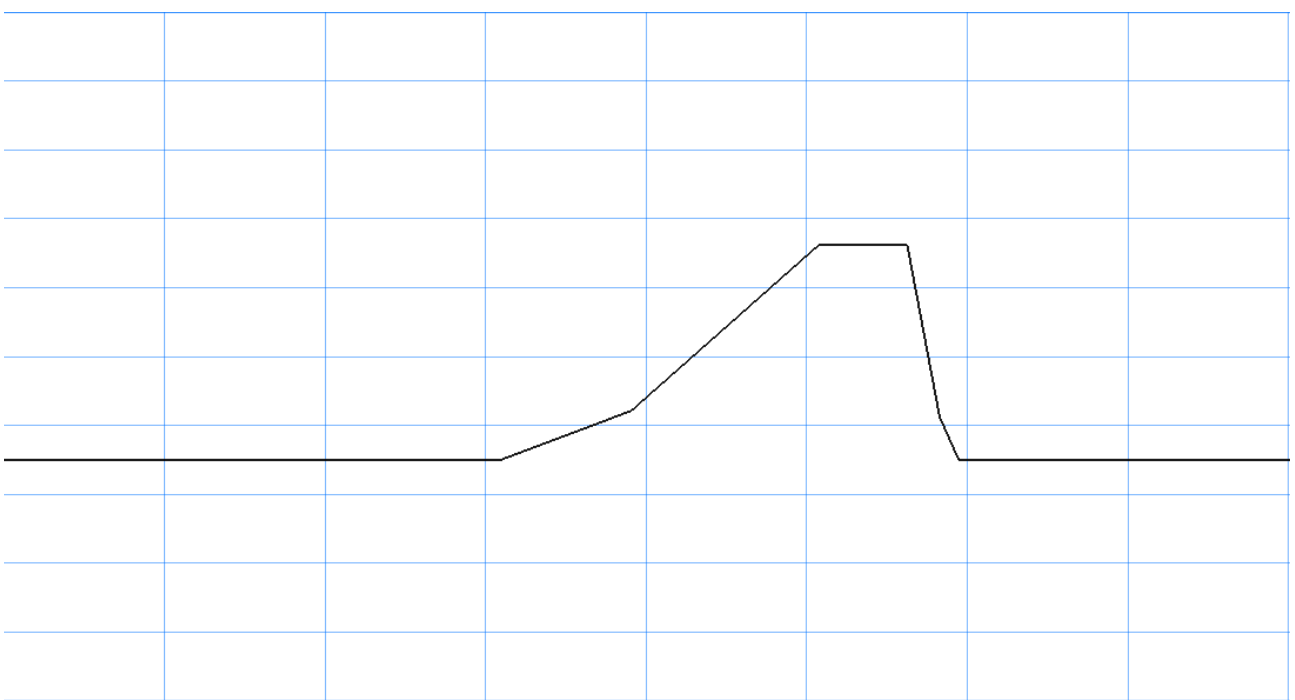


图 4-2-9 底盘电机启动打印值

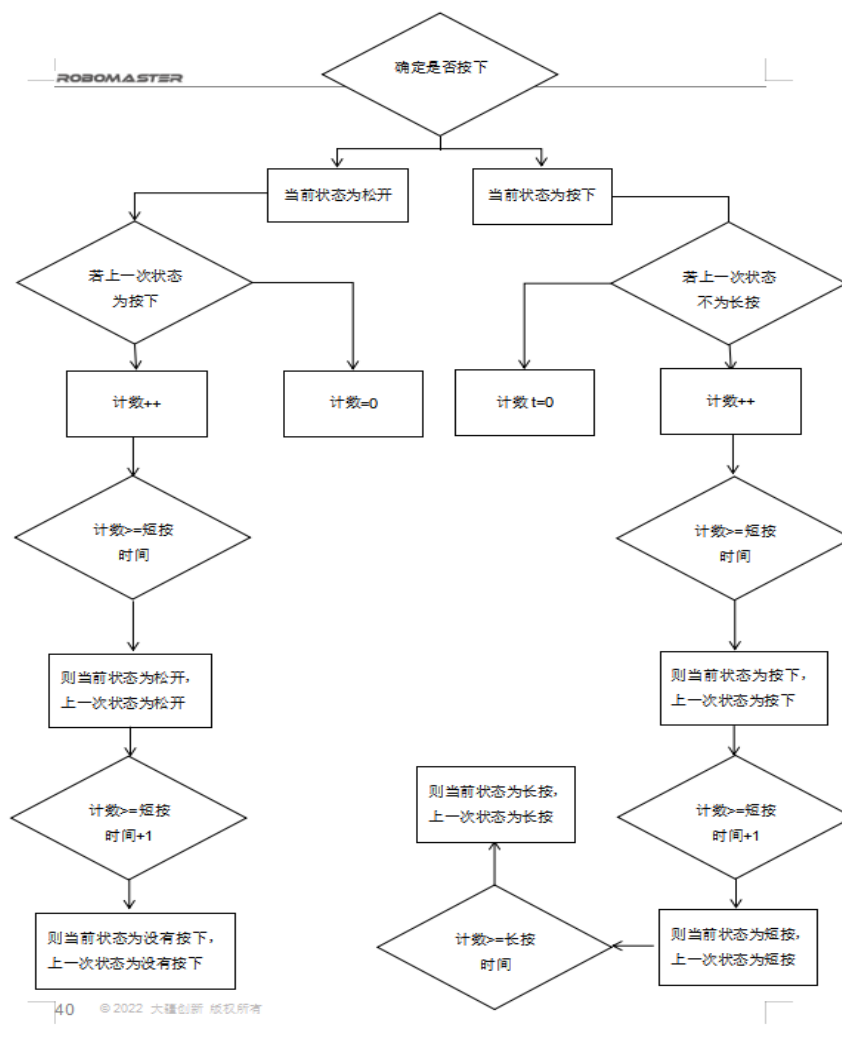


图 4-2-9 按键逻辑图

3. 变速换向大陀螺

在比赛中，陀螺转已经变得越来越必不可少。各个参赛队伍也已经具备反陀螺算法，我们针对反陀螺算法，在电控层面可以把普通的陀螺转添加一个变速和换向的因素。变速和换向需要做到无规律，并且不能影响操作手。为此，我们参考大风车是速度公式，将陀螺转的速度按照以下公式进行。其中， a 和 w 是在每次进入陀螺转时，通过 STM32 的 RNG 取得随机数。 t 为运行时间（单位：秒/s）， $b = 2.0 - a$ ； c 的值为 1 或 -1，决定转向；

$$\text{spd}(t) = [a * \sin(w * t) + b] * c$$

陀螺的实现比较简单，我们仅需将云台的速度期望映射在底盘上。公式如下， V_{x1} 和 V_{y1} 为给底盘的 x 轴方向和 y 轴方向的分量， V_{x2} 和 V_{y2} 是云台的速度期望， θ 是云台电机与底盘机械中值的角度。底盘的旋转分量为 $\text{spd}(t)$ ， $\cos \theta$ 和 $\sin \theta$ 是调用 CMSIS DSP Library 并且开启浮点加速器用于加速运算。

$$\begin{cases} V_{x1} = V_{x2} * \cos \theta - V_{y2} * \sin \theta \\ V_{y1} = V_{x2} * \sin \theta + V_{y2} * \cos \theta \end{cases}$$

4. 超级电容任务

针对于 2022 赛季的规则及结合本战队的实际情况，我们整理出一套适合本队伍的超级电容控制方案，并且充分利用到缓冲能量提高英雄机器人及步兵机器人在赛场上的灵活性。

关于我们的超级电容控制逻辑，分为开启超级电容和未开超级电容两大块，而在这两大块中，又分为多个具体的模式以满足机器人在 RoboMaster 赛场上不同情况下的需求。具体模式在未开启超级电容时有低速慢充模式，低速快充模式及开启超级电容的加速模式，另外还有利用缓冲能量时针对飞坡及猛转向撤退等特殊情况下的其它模式。

通常情况下，机器人是处于低速慢充模式，即通过不断测试得到合适的速度，使得底盘供电不仅由电源管理提供，还有一部分由超级电容提供。而超级电容本身处于充电状态，当给定的速度达到充电于放电的分界线之上，即充电速率略高于放电速率时，即可满足超级电容处于充电状态的同时，又能保证在未完全开启超级电容时的最大速度。当机器人开启加速模式时，设定速度达到最高，此时超级电容大幅度给底盘供电，耗电速率远高于充电速率。

为避免超级电容过度消耗能量造成机器人底盘无法动弹情况，我们获取从超级电容控制模块传送的电容电压等信息实时检测超级电容的电量情况。因此除操作手可在超级电容消耗过度前手动关闭加速模式外，当机器人检测到超级电容能量不足时也会自动跳转到高速慢充

模式。机器人处于慢充模式时，设定速度此时降至最低以保证超级电容只充不耗，并且为保证机器人的正常运动能力，直到超级电容有一足够电量时才能转到低速慢充模式。此外，为防止不同模式切换时造成的速度突变导致功率急增，我们采用分段斜坡在模式转换时提供缓冲。

根据 2022 赛季 RoboMaster 规则，在正常情况下英雄机器人拥有 60J 的缓冲能量，在飞坡时于机器人触发飞坡增益拥有 250J 的能量。为了满足英雄机器人的底盘需要大功率的情况，我们充分考虑到缓冲能量并进行利用。第一，由于英雄机器人体型相对庞大，并且有急速转头的需求，导致急转时底盘需大功率支撑，为防止在超级电容耗尽时无法提供高功率，我们提供只利用电源管理和缓慢消耗缓冲能量的模式；第二，为保证英雄机器人能够在赛场飞坡，我们设定了由电源管理，超级电容和缓冲能量共同供电的飞坡模式，使得本队英雄机器人能够顺利飞坡。

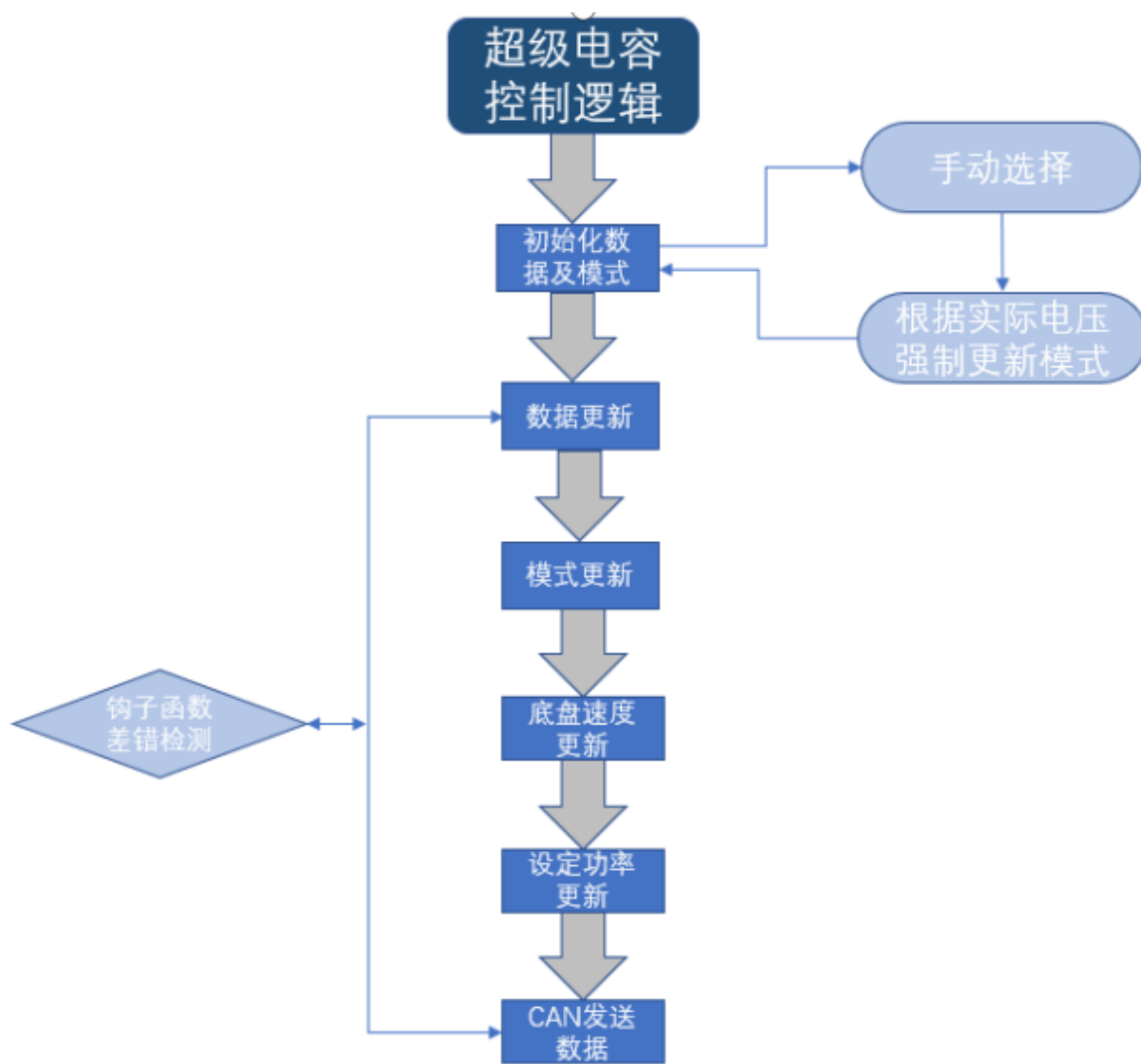


图 4-2-10 超级电容任务逻辑图

4.3.4 软件测试

1. 单元测试

单元测试是指对软件中的最小可测试单元进行检查和验证。在电控的单元测试中一般是通过 keil 自带的 Debug 模式来进行函数等单元的测试，通过判断该函数所包含的结构体数据是否正常来判断函数的有效性。除此之外，我们还会通过 keil5 自带的检测属性来检测和判断函数的有效性，从而完成整个单元测试。

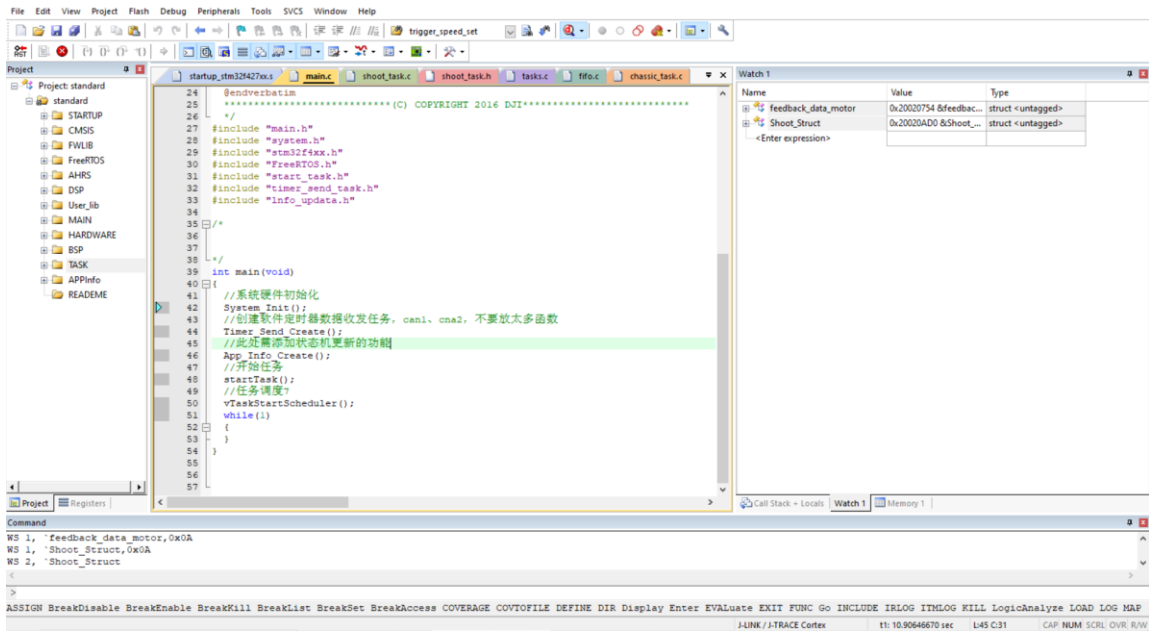


图 4-2-11 单元测试代码图

2. 集成测试

进行完所有的单元测试之后，将每个单元组合起来同时进行运用，由这些所有的单元测试组成的模块进行总体的测试，一般我们是通过检测电机的运行情况来对这个模块的测试进行判断，进而完成这一部分的集成测试。

3. 确认测试

在完成了所有的集成测试之后，即保证所有的模块都能够稳定的运行之后，做一个最后的确认，确认之后上传至 Coding 进行一个版本的迭代和保存。然后由机械组的成员从外部来对软件进行一个测试，即做一次黑盒测试。

4. 系统测试

在完成了前面几个测试之后，对该部分的整体进行一个系统测试，对整个软件的功能、性能、验收、安装等进行一个总体的测试，并且根据测试的结果进行一个评估和总结，做出一个最优设计方案，并进行一个保存，作为以后的数据资料储存起来。

4.4 算法设计

4.4.1 功能简介

1. 功能介绍

模块	功能
装甲板识别	检测敌方机器人装甲板位置信息并识别其数字
角度解算以及控制器	根据上述位置信息解算目标相对枪管的 Yaw、Pitch 角度及距离，使用 PID 速度环解算器以及卡尔曼线性预测器实现跟随预测
相机驱动	海康威视工业相机 SDK 封装，实现相机参数控制及图像采集
串口通讯	与下位机通信，传输机器人姿态信息及操作手反馈视觉的控制信息
反陀螺控制器	判断敌方是否处于陀螺状态，根据装甲板位置信息解算出陀螺中心位置以及陀螺旋转半径，通过预测实现击打小陀螺

2. 算法框架

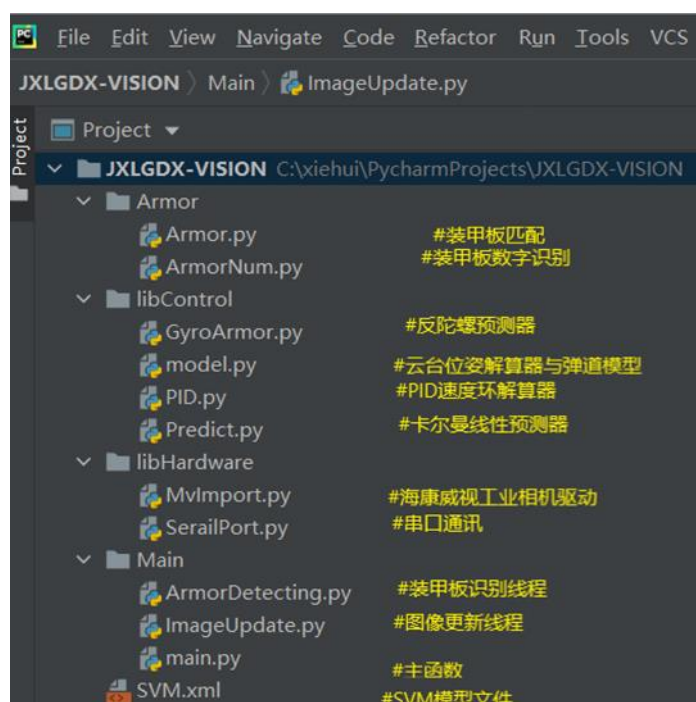
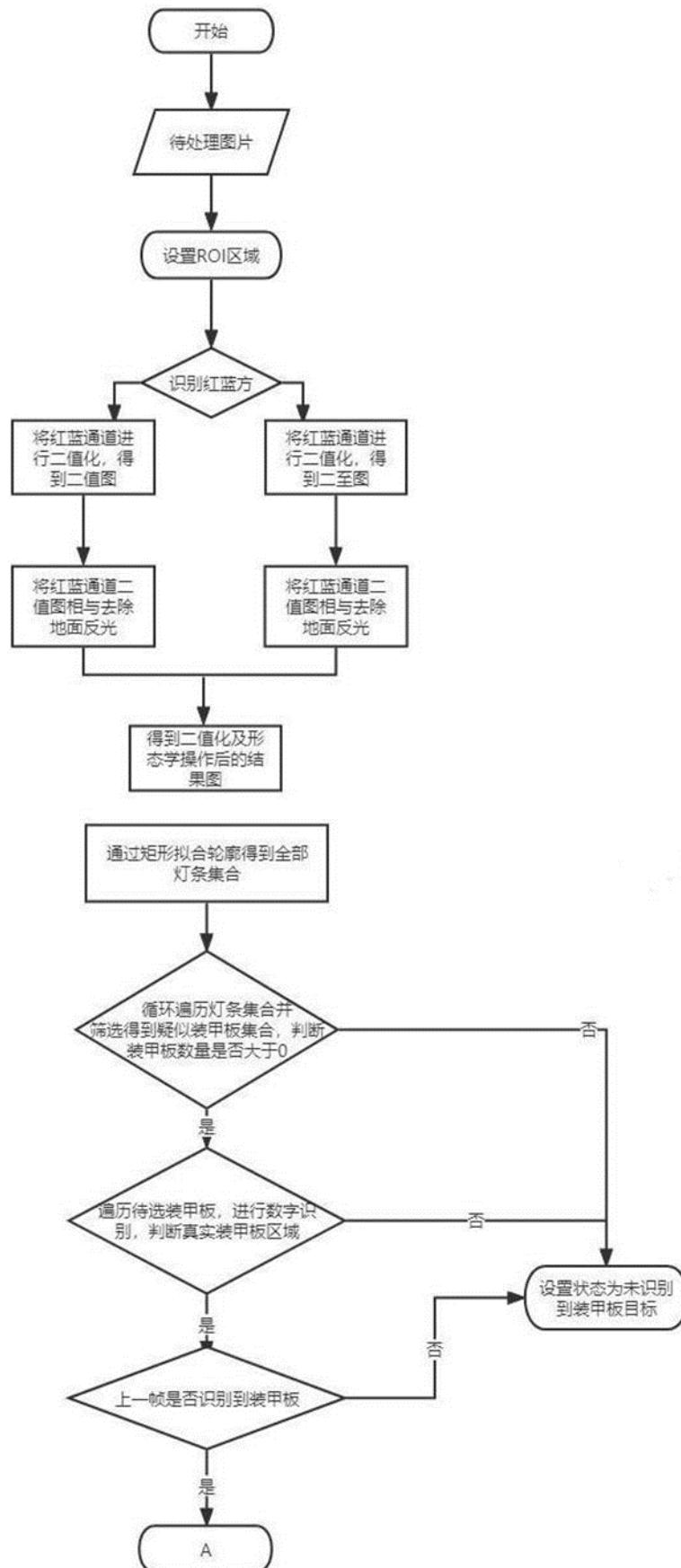


图 4-3-1 算法框架图

3. 算法流程图



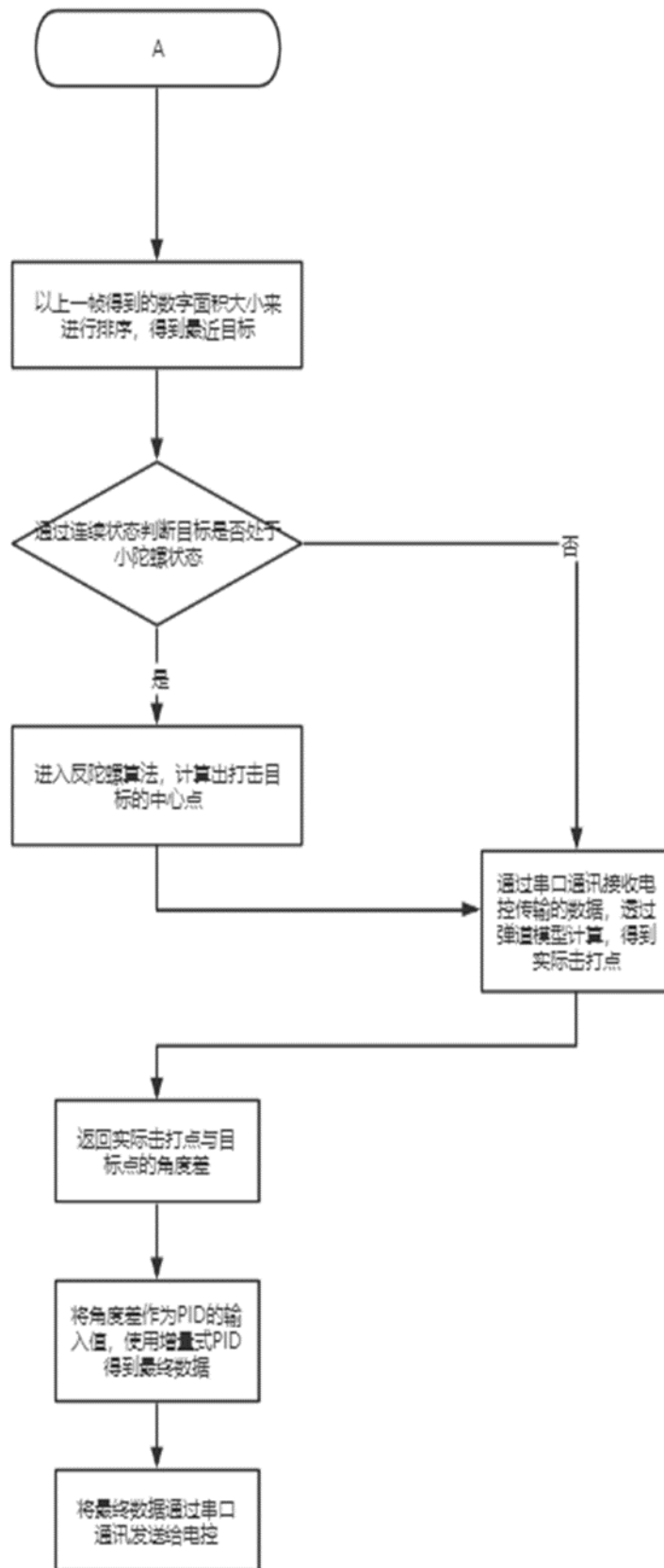


图 4-3-2 算法流程图

4.4.2 核心算法原理

1. 装甲板识别（筛选装甲板）

装甲板识别算法使用基于检测目标特征的 OpenCV 传统方法，实现检测识别的中心思想是找出图像中所有敌方颜色灯条，并使用找出的灯条一一拟合并筛选装甲板。主要步骤分为：图像预处理、灯条检测、装甲板匹配、装甲板数字识别及最终的目标装甲板选择。

但此算法最特别的一点在于装甲板匹配，多数学校的装甲板匹配方法：对检测到的灯条进行两两匹配，通过判断两个灯条之间的位置信息：角度差大小、错位角大小、灯条长度差比率和 X, Y 方向投影差比率，筛选合适的装甲板。而我们则只需两两匹配，之后依靠装甲板独有的数字直接进行筛选。由于图像预处理时，我们为了更好的适应环境，已将曝光调低，因此我们使用了 Gamma 校正使我们能得到装甲板上的数字。因此由效果图我们可以得知两两匹配的待筛装甲板中有数字的轮廓面积最大，因此我们可以根据待筛装甲板中的轮廓面积最大且在一定阈值中，来直接筛选装甲板。

Gamma 校正原理：

$$\text{公式: } f(I) = I^\gamma$$

结合下图看 Gamma 校正的作用:1) 当 $\gamma < 1$ 时，如虚线所示，在低灰度值区域内，动态范围变大，进而图像对比度增强(当 $x \in [0, 0.2]$ 时， y 的范围从 $[0, 0.218]$ 扩大到 $[0, 0.5]$)；在高灰度值区域内，动态范围变小，图像对比度降低(当 $x \in [0.8, 1]$ 时， y 的范围从 $[0.8, 1]$ 缩小到 $[0.9, 1]$)，同时，图像整体的灰度值变大。

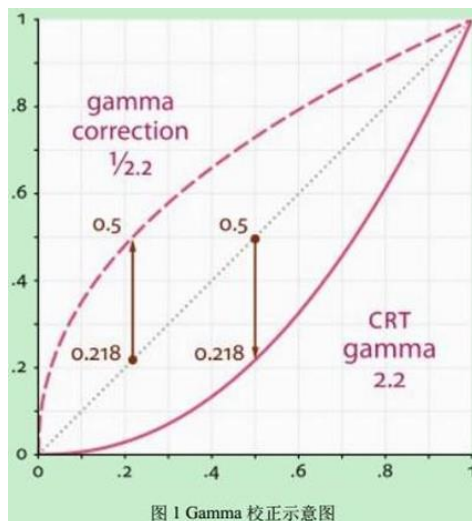


图 4-3-3 原理示意图

当 $\gamma > 1$ 时，如实线所示，低灰度值区域的动态范围变小，高灰度值区域在动态范围变大，降低了低灰度值区域图像对比度，提高了高灰度值区域图像对比度。同时，图像整体的灰度值变小。

2. 反陀螺算法

利用针孔相机模型 ($D = (W * F) / P$ [摄像头到物体的距离=物体的真实宽度*摄像头的焦距/物体的像素宽度]) 和三角形相似原理根据装甲的倾斜所造成的宽度像素差计算该状态下对应的圆心角度。如右图所示点 O 为相机光轴原点，BC 为投影宽度，AB 为理想正对宽度，二者比值即为所对应圆心角的余弦值。即 $\angle 1, \angle 1 = \angle 2$ 。

保留位置坐标和角度，记录每两帧间的角度差和移动位置差，根据角度差关系确定置信等级，置信等级达到设定阈值为判断为陀螺的基本要求，在此基础上增加连续多帧旋转角度的方差计算，将方差和位置记录总次数作为另一判断依据，增加判断和计算的准确性。

根据所保留的位置坐标的 x 和 z 坐标进行最小二乘拟合，估计对应运动范围截距，根据保留角度极值计算半径和运动圆心，进而完成下一步运动估计。

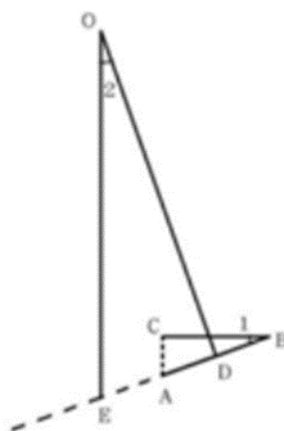


图 4-3-4 原理示意图

3. 弹道模型

● 原方案：

由于使用 PNP 时，在角度方面总会存在一点误差，而且英雄弹丸下坠影响力较大，因此我们选择在 Pitch 轴方面用测量出的距离自己写弹道模型，Yaw 轴则沿用 PNP 解算出来的角度。根据物理斜抛模型，已知弹丸发射的初速度 v ，重力加速度 g ，以及得到的距离 D 和 PNP

得到的 Pitch 角度 a 来构建弹道模型得出枪口要达到的角度。枪口距离装甲板的高 $H = \tan(\langle 1)$

* D 。 ($\langle 1$ 为上述反陀螺算法中的 $\langle 1$)

根据上述已知量，构建弹道模型：

根据 H 的范围不同，击打的类型不同，弹道模型也时就不同。

当击打步兵：

当 D 小于枪口平抛时的距离时：

$$h1 = \frac{2 * v^2 * H}{g * D^2} - 1 + \frac{v^4}{D^2 * g^2}$$

$$a = \arctan \left(-\frac{v^2}{D} + \sqrt{h1} \right)$$

当 D 大于枪口平抛时的距离时：

$$h1 = 2 * v^2 * H * g + v^4 - g^2 * D^2$$

$$a = -\arctan \left(\frac{v^2}{D} - \sqrt{\frac{h1}{g^2 * D^2}} \right)$$

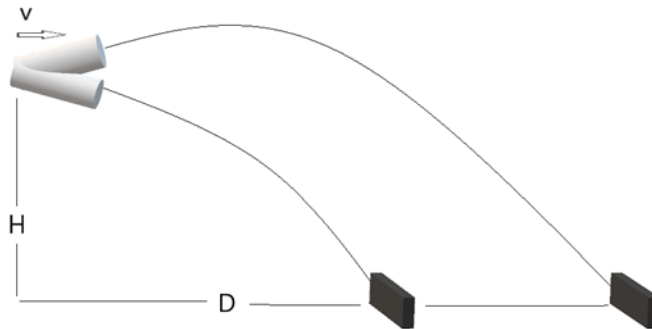


图 4-3-5 弹道模型

当击打哨兵时：

由于枪口只会向上，无论距离远近，模型只有一种：

$$h1 = -\frac{2 * v^2 * H}{g * D^2} - 1 + \frac{v^4}{D^2 * g^2}$$

$$a = -\arctan \left(\frac{v^2}{D} - \sqrt{h1} \right)$$

● 最终方案:

在原方案中 pitch 轴除了需要得到距离 D, 初速度 v, 从电控得知的 pitch 角, 外还需要计算得到 h, 而 h 与反陀螺算法中的 κ 以及 D 有关, 因此误差常常有点大, 那样并不会提高击打的准确率。而后经过我的实验, 找到了另一种更为准确的方法。相当于我把枪口对准装甲板中心后又往上抬了一个受下坠力影响的角度 a, 而所需要的高度 h 则是通过使用距离 D, 装甲板中心坐标 (x, y), 摄像头像素中 y 轴方向的一半 240, 以及焦距 f 做小孔成像 ($D = (W * F) / P$ [摄像头到物体的距离=物体的真实宽度*摄像头的焦距/物体的像素宽度]) 得到: $h = \lfloor |240 - y| * D / f / 100.0 \rfloor$

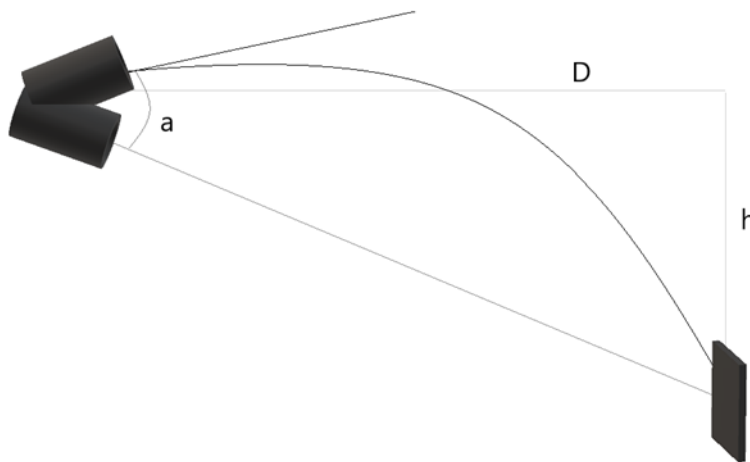


图 4-3-6 弹道模型

枪口以水平为 0° , 向上为负向下为正。根据上述已知量, 构建弹道模型:

当枪口向上:

装甲板位于摄像头镜头中心的上方:

$$h1 = - \arctan \left(\frac{\frac{D}{100.0} * |\tan (pitch)| + h}{\frac{D}{100.0}} \right)$$

$$a = h1 - \lfloor \arcsin \left(\frac{\frac{D}{100.0}}{\cos (h1) * v^2} \right) / 2 \rfloor$$

装甲板位于摄像头镜头中心的下方:

$$h1 = -\arctan \left(\frac{\frac{D}{100.0} * |\tan (pitch)| - h}{\frac{D}{100.0}} \right)$$

$$a = h1 - |\arcsin \left(\frac{\frac{D}{100.0}}{\cos (h1) * v^2} \right) / 2|$$

当枪口向下:

装甲板位于摄像头镜头中心的上方:

$$h1 = -\arctan \left(\frac{\frac{D}{100.0} * |\tan (pitch)| - h}{\frac{D}{100.0}} \right)$$

$$a = h1 - |\arcsin \left(\frac{\frac{D}{100.0}}{\cos (h1) * v^2} \right) / 2|$$

装甲板位于摄像头镜头中心的下方:

$$h1 = -\arctan \left(\frac{\frac{D}{100.0} * |\tan (pitch)| + h}{\frac{D}{100.0}} \right)$$

$$a = h1 - |\arcsin \left(\frac{\frac{D}{100.0}}{\cos (h1) * v^2} \right) / 2|$$

4. PID 算法

PID 算法的执行流程是非常简单的,即利用反馈来检测偏差信号,并通过偏差信号来控制被控量。而控制器本身就是比例、积分、微分三个环节的加和。其功能框图如下:

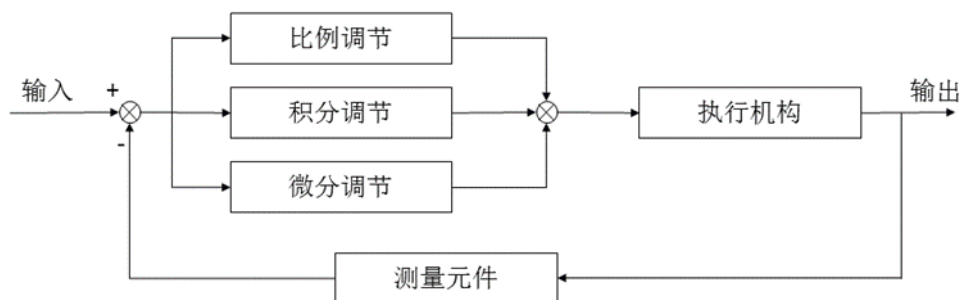


图 4-3-6 功能框图

根据上图我们考虑在某个特定的时刻 t ，此时输入量为 $rin(t)$ ，输出量为 $rout(t)$ ，于是偏差就可计算为。于是 PID 的基本控制规律就可以表示为如下公式：

$$U(t) = kp(err(t) + 1/T \int err(t) dt + (T_D d err(t)) / dt$$

因为我们传的数据是离散的，所以将 PID 公式离散化，我们使用的是增量式 PID，因此 PID 公式为：

$$U(k) = k_p(err(k) - err(k-1)) + k_i(err(k)) + k_d(err(k) - 2err(k-1) + err(k-2))$$

将云台位姿解算与弹道模型算法得到的值作为输入值传给增量式 PID 后，再将输出值通过串口通讯算法传给电控。

卡尔曼预测算法：

Kalman 滤波是一种线性滤波与预测方法，Kalman 滤波分为 2 个步骤，预测(predict)和校正(correct)。预测是基于上一时刻状态估计当前时刻状态，而校正则是综合当前时刻的估计状态与观测状态，估计出最优的状态。预测与校正的过程如下：

预测：

$$x_k = Ax_{k-1} + BU_{k-1} \quad (1)$$

$$P_k = AP_{k-1}A^T + Q \quad (2)$$

校正：

$$K_k = P_k H^T (HP_k H^T + R)^{-1} \quad (3)$$

$$x_k = x_k + K_k(z_k - Hx_k) \quad (4)$$

$$P_k = (I - K_k H)P_k \quad (5)$$

公式 1 是状态预测，公式 2 是误差矩阵预测，公式 3 是 kalman 增益计算，公式 4 是状态校正，其输出即是最终的 kalman 滤波结果，公式 5 是误差矩阵更新。

以上是对卡尔曼一个简单的理解，但是如果是对非线性关系的数据进行预测，KF 的假设之一就是高斯分布的 x 预测后仍服从高斯分布，高斯分布的 x 变换到测量空间后仍服从高斯分布。可是，假如 F 、 HF 、 H 是非线性变换，那么上述条件则不成立。显然，线性代数是非常棒的，它让我们可以用一种非常简洁的形式来表示像卡尔曼滤波这样的复杂算法。然而，线性代数并不是全部。顾名思义，线性代数仅限于表示线性关系，即，以直线为特征的关系。但是现实中存在大量的无法直接通过类似这样的线性矩阵来表述变量前后的关系。比方一个简单的线性方程是这样更新的，其中 W_k 为外界扰动项。

Model:

$$\begin{aligned}x_k &= x_{k-1} + w_k \\z_k &= cx_k + v_k\end{aligned}$$

Predict:

$$\begin{aligned}\hat{x}_k &= \hat{x}_{k-1} \\p_k &= p_{k-1} + q\end{aligned}$$

Update:

$$\begin{aligned}g_k &= p_k c (c p_k c + r)^{-1} \\ \hat{x}_k &\leftarrow \hat{x}_k + g_k (z_k - c \hat{x}_k) \\ p_k &\leftarrow (1 - g_k c) p_k\end{aligned}$$

那么非线性则是这样更新的。

Model:

$$\begin{aligned}x_k &= x_{k-1} + w_k \\z_k &= h(x_k) + v_k\end{aligned}$$

Predict:

$$\begin{aligned}\hat{x}_k &= \hat{x}_{k-1} \\p_k &= p_{k-1} + q\end{aligned}$$

Update:

$$\begin{aligned}g_k &= p_k c_k (c_k p_k c_k + r)^{-1} \\ \hat{x}_k &\leftarrow \hat{x}_k + g_k (z_k - h(\hat{x}_k)) \\ p_k &\leftarrow (1 - g_k c_k) p_k\end{aligned}$$

因此我们需要将非线性转到线性，才能使用卡尔曼滤波预测非线性数据。

4.4.3 算法性能分析

1. 传统视觉相对于深度学习，响应速度要快，但是对环境要求严苛，一开始没有降低曝光时的识别效果容易误识别，因此我们解决方案是将曝光调低，对于要清楚的看到数字，则使用 Gamma 矫正算法。Gamma 矫正算法相对于 log 矫正算法，虽然 log 矫正算法适用于对比度较低和灰度图，但也同样对环境要求严苛，因此我们使用 Gamma 算法。

2. 装甲板匹配时装甲板筛选从灯条两两匹配后直接采取数字筛选装甲板，通过数字识别对灯条匹配的待筛选装甲板进行匹配，可以更好的适应图像预处理的二值化阈值。

3. PID 一开始使用位置式 PID，但使用这种该控制器存在的缺点：由于积分饱和带来的影响，控制器需要相当一段长的时间退出饱和区，由此引起系统产生大幅度超调，不仅难以调节，效果也不太理想，而增量式 PID 只求取控制量的变化量，一旦确定了 K_P 、 T_I 、 T_D ，只要使用前后三次测量值的偏差，即可由公式求出控制增量，而得出的控制量 $\Delta u(k)$ 对应的是近几次位置误差的增量，而不是对应与实际位置的偏差，因此没有误差累加，容易通过加权处理获得比较好的控制效果，并且在系统发生问题时，增量式不会严重影响系统的工作。因此决定 PID 使用增量式 PID，相比于位置式 PID 便于调节的同时也足够快速稳定。

4. 弹道模型基于电控发送的速度建立，考虑到英雄大弹丸，下坠的影响力较大，因此结合 PNP 算法经过换算得到的距离以及高度构建物理模型，使弹道更加的稳定，命中更加的准确。

5. 卡尔曼预测算法目前只能用于预测线性数据，因此无法在敌方突然转向的那一瞬间也做到预测，因此期望之后优化卡尔曼算法，做到无迹卡尔曼预测。

4.4.4 算法库介绍与接口说明

4.4.4.1 视觉算法库

1. 串口通讯

与电控的通讯方式采用帧传输方式外加 crc 校验码校验。具体格式如下图：

江西理工大学乘风战队视觉-» 电控协议		
数据协议格式		版本：V2.0
SOF: 0xA5	CMD_ID	CRC8校验 Data数据部分 CRC16校验
帧格式说明		
SOF: 0xA5	固定帧头: 0xA5	
CMD_ID	0x01	自瞄模式
	0x02	保留
CRC8校验	1-byte	
Data数据部分	CMD_ID:0x01	
	Distance	距敌方机器人的距离
	Pitch	Pitch角度
	Yaw	Yaw角度
CRC16校验	2-byte	

图 4-3-7 电控协议

2. PID

使用增量式 PID，将目标位姿和当前位姿相减得到位姿误差，作为 PID 调节器的输入量，PID 调节器根据已记录的误差和当前误差分别计算比例、积分和微分调节量，相加后得到调节修正值。根据上一帧数据和当前数据计算出基准速度，然后加上速度修正值得到最终速度并返回。

接口说明：`def SetStepSignal(self, StepSignal, SystemOutput, SampleTime):`

参数： StepSignal: 根据识别结果解算的目标位姿

SystemOutput: 根据陀螺仪回传的云台当前位姿

SampleTime: 当前时间

3. 卡尔曼算法

初始化参数，根据测距精度，设置过程噪声，测量噪声；从参数文件读取第一帧预测次数，预测数度增益量，控制频率；外部调用 `ShootSpeed()` 设置当前射速等级。外部直接调用 `motion_predict()` 获得预测后的装甲坐标，内部判断调用 `initFilter()` 初始化卡尔曼滤波器，并用第一帧输入的装甲绝对坐标进行多次预测使其收敛，判断当前帧和上一帧装甲相对距离，根据设置的阈值判断装甲板是否切换，装甲板切换需重新初始化滤波器，得到预测结果，击打远距离装甲板弹丸飞行时间更长，不同射速等级弹丸飞行时间不同，所以加入装甲距离和射速调节预测量大小。

接口说明：`def motion_predict(position):`

`def SHootSpeed(shoot_level):`

参数： shoot_level: 射速

a_position: 当前装甲绝对坐标

4.4.5 算法结果

1. 下图为灯条预处理

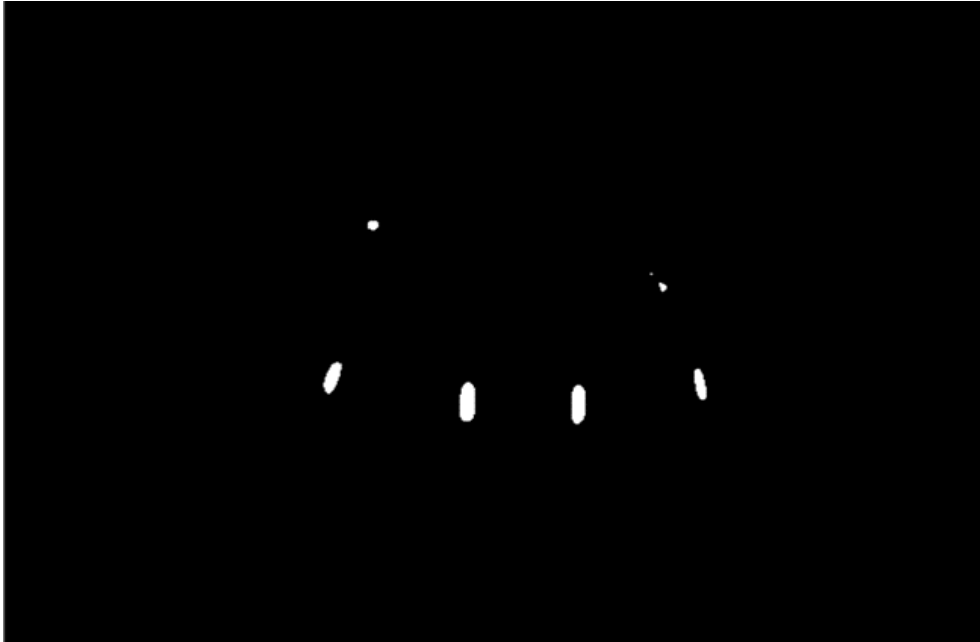


图 4-3-8 灯条预处理

2. 装甲板的识别

下图为装甲板的识别以及数字识别效果图。

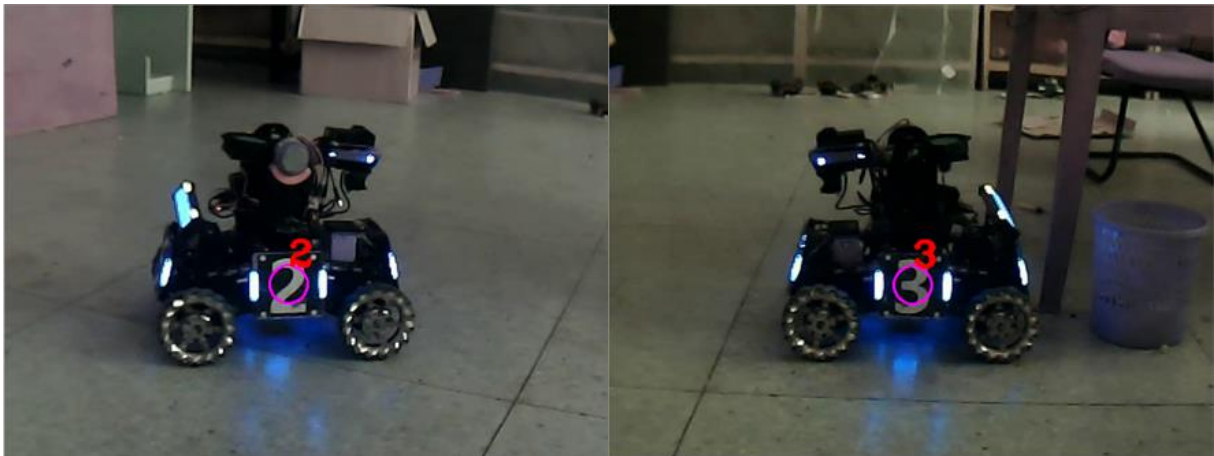


图 4-3-9 装甲板识别图

4.5 其它

4.5.1 UI 交互开发

在 RM 赛场，操作手操作电脑控制机器人，这里除了基本的机器人视野，还要设置一些 UI 为操作手提供瞄准等辅助功能。以下是英雄机器人的 UI。该 UI 界面左边有超级电容开启标志位，摩擦轮开启标志位，超级电容的具体电量和拨盘的相对角度和绝对角度来判断是否卡弹；中间是瞄准线，这个可根据机器人实际而定，保证在一定距离内依靠准线可以击中，由于采用射速优先，高射速情况下，在 5m 之内有较高的准确度，另外在加上一些特殊点位的基准线；正下方是超级电容能量条和云台的绝对角度，超级电容能量条是为了更清楚的判断当前超级电容能量储备情况绘制的，如果能量条较低说明超级电容当前储能低，储能低则电压较低，会造成电机异常转动，整体就是机器人不受控制自己乱动，因此操作手需要注意能量条，能量低则缓慢或者停止运动让超级电容充能。其中静态 UI 的绘制无需多说，直接调用绘制字符串函数和绘制图形函数。动态 UI 即数据或图形实时更新的 UI，一是有些数据需要在变化如超级电容电压和云台角度等，二是某些图形化 UI 可直观的看出机器人状态。



5. 研发迭代过程

5.1 版本迭代过程记录

机械组：

版本号	功能或性能详细说明	完成时间
V1.0	确定基本设计方案，本赛季使用下供弹英雄机器人，并完成第一代英雄机器人的图纸的绘画	2021 年 07 月 10 日
V2.0	考虑第一代英雄机器人云台结构的弊端及制作成本，重新设计了第二代英雄机器人。	2021 年 11 月 05 日
V2.1	完成弹路链的优化，解决卡弹问题	2021 年 12 月 15 日
V2.2	完成第二代拨盘的设计及制作	2022 年 01 月 20 日
V2.3	去除传统的减震方式，增加自适应减震系统。	2022 年 03 月 05 日

电控组：

版本号	功能或性能详细说明	完成时间
V1.0	完成 Pitch、Yaw 轴云台代码的移植和编写，完成 Pitch、Yaw 轴的 PID 初步调节。	2021 年 10 月 15 日
V1.1	完成摩擦轮速度调节、一键启动，完成摩擦轮，Yaw、Pitch 轴电机的 PID 第二次调节。	2021 年 11 月 20 日
V1.2	完成键鼠操作，并实现两板之间的通信。	2021 年 12 月 01 日
V1.3	完成底盘电机的调节，代码初步上英雄机器人进行调节。	2021 年 12 月 15 日
V1.4	完成英雄机器人所有模块的测试，并且与视觉完成对接。	2022 年 01 月 15 日
V2.0	给云台 Pitch 轴、Yaw 轴添加卡尔曼滤波和前馈控制	2022 年 03 月 01 日
V2.1	完成摩擦轮电机的 PID 的优化和拨盘串级 PID 的优化	2022 年 03 月 15 日

视觉组：

版本号	功能或性能详细说明	完成时间
V1.0	完成基本的视觉识别算法以及串口通讯基本实现准确识别以及与电控能进行通信。	2021 年 11 月 15 日
V1.1	新增了数字识别、云台位姿解算算法能识别此时敌方机器人的编号区分该兵种，云台位姿解算定点击打时能实现准确击打。	2022 年 1 月 15 日
V1.2	增加了反陀螺算法以及增量式 PID 算法，反陀螺算法能实现判断敌方机器人是否处于反陀螺状态，增量式 PID 能实现快速的实时跟随。	2022 年 2 月 15 日
V2.0	目前这一版添加了卡尔曼预测算法，实现了对线性数据的基本预测。	2022 年 3 月 15 日

硬件组：

版本号	功能或性能详细说明	完成时间
V1.0	参考 2021 赛季机器人出现的问题，重新设计超级电容控制逻辑，并测试相应参数	2021 年 11 月 5 日
V2.0	重新研发超级电容方案，结合其它队伍的设计思路，完成主控 PCB 设计及功率控制 PCB 设计，完成元器件选项	2021 年 12 月 10 日
V2.1	设计新版超级电容的充放电控制逻辑，完成嵌入式程序框架	2021 年 2 月 1 日
V2.2	优化电流环、电压环、功率环 PID 参数及其它电气参数，添加滤波算法	2022 年 03 月 11 日

5.2 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	解决人员
1	完成第一代英雄机器人图纸的设计之后，发现我们的英雄机器人是通过增加或缩短弹路链长度来达到Pitch轴俯仰角的变化	在进行初代英雄机器人的设计时，对下供弹英雄了解不足，导致在确认方案时考虑不充分	虽然可以通过在英雄机器人中增加传感器来解决这一问题，但为避免调试难度的增加，于是选择重新绘画第二代英雄机器人。将云台弹路链与云台发射机构通过上下各两个连杆连接，这种连接方式可以有效避免当Pitch轴俯仰角时，弹路链长度的变化。	V1.0	队长:葛殷 机械工程师: 吴伊果
2	四个麦轮速度不一致	因为在实际测试过程中因为机器人的重心的位置问题，导致四个麦轮收到的重力也不同，从而导致四个麦轮速度有点不一致	英雄机器人的重心进行比例调整来改善这个问题,现状况挺好	V1.3	嵌入式软件工程师:陈杰 机械工程师: 吴伊果
3	底盘速度突变，导致底盘不稳	在进行键盘对底盘进行控制时。因为是直接给四个电机赋予速度值，所以英雄机器人在进行移动时，速	模仿遥控器的赋值来时电机的速度变为一个累加值，将赋予的速度进行分段斜坡和滑动滤波，从而能够根据改变比例值来对比赛的实际情况进行一个加减速的调节	V1.3	嵌入式软件工程师:陈杰

		度是突变的,从而 导致底盘的稳定 性较差,并且对电 机造成损伤			
4	云台稳定性 不强,电机 发热	云台电机单用串 级 PID 进行调节, 因为 PID 的调节 不好,从而导致云 台稳定性差,而电 机发热因为 Pitch 轴配重比的原因, 使它到达每个位 置所需的电压是 不同的	所以我们对 Yaw、Pitch 轴进 行了一个卡尔曼滤波,对 Yaw、Pitch 轴的值进行一个 预测,进而再次的提高 Yaw、 Pitch 轴的稳定性,再进行测 试之后, Yaw、Pitch 轴的稳 定性再次有了明显的提升, Yaw 轴的稳定性达到了预期 的效果。我们后面加了一个 前馈控制,提前测出到达位 置所需要的电压值,进而进 行一个 PID 控制来进行抗干 扰,从而使 Pitch 轴的稳定性 再次提高,并且解决了电机 易发热的问题	V2.0	嵌入式软件 工程师:陈杰
5	中出拨盘电 机存在转动 困难的现象	由于电机转动时 需要承受弹路链 弹丸重力、拨盘内 壁阻力,以及平衡 拨盘挡块分力,最 终可能造成电机 扭转力不足和电 机发烫严重,缩短 电机使用寿命	重新设计侧出拨盘,由于侧 出拨盘转动时只存在部分来 自拨盘内壁的阻力,这样可 以有效减少电机转动时发热 的现象。	V2.0	队长:葛殷 机械工程师: 吴伊果
6	弹路链卡弹	雕刻机和 3D 打印	通过砂纸打磨以及增加轴承	V2.1	队长:葛殷

		机精度不足; 装配存在误差; 未处理好云台中发射机构与弹路链衔接处	可以在一定程度上润滑弹路链; 调整发射机构与弹路链衔接处板材的形状来实现不卡弹		机械工程师: 吴伊果
7	射击的精度不够, 并且偶尔会卡弹	主要是因为拨盘一开始我们采用的是单角度环的 PID 对拨盘电机进行一个控制, 但后面发现因为该电机用的是 M3508, 对于角度的控制比较差, 在 PID 调节较差的情况下, 使得电机发生抖动, 而且转的角度误差也大	最初我们是改用速度环 PID 对其进行一个 PID 控制, 修改之后, 拨盘虽然在空弹的情况下效果良好, 但因为满载之后, 电机负载过大, 使弹丸推不上去, 所以我们现在改成了串级 PID 来对它进行调节, 在保证稳定性的前提下, 提高电机提供的力, 进而解决了它的问题	V2.1	嵌入式软件工程师: 陈杰
8	拨盘卡弹	雕刻机、3D 打印机精度不足; 装配存在误差; 大弹丸老旧, 磨损严重; 电控代码不稳定	使用砂纸不断打磨拨盘板材内壁; 反转拨盘, 出弹口增加轴承	V2.2	队长: 葛殷 机械工程师: 吴伊果 嵌入式软件工程师: 陈杰
9	弹道模型不稳定	弹丸发射时左右摩擦轮转速不同, 以及每次发射时速度不同	电控将此时预测的弹丸的初速度发给视觉, 视觉将会根据速度的不同自动调整弹道模型	V2.3	嵌入式软件工程师: 陈杰 视觉算法工程师: 陈洋
10	击打转速快的小陀螺	击打时, 由于陀螺速度太快, 因此视	电控与视觉在通讯方面加一个允许视觉发射的内容, 当	V2.4	嵌入式软件工程师: 陈杰

	时，操作手控制发射会有延时，导致击打不准确	觉方面设定击打点为陀螺正面，导致操作手不知何时才是正确击打时间	视觉请求控制发射时，电控给予权限，并返回可以发射		视觉算法工程师：陈洋
11	识别装甲板时数字识别不准确	由于使用的是传统视觉算法，因此为了保证识别稳定，会将曝光调低，使环境变得简单，但同时数字也会变得不清晰	在筛选出装甲板后，截取筛选后的装甲板区域，使用 gamma 算法以及二值化使数字变得清晰，使装甲板准确识别的同时也能准确识别数字	V2.5	视觉算法工程师：陈洋
12	自适应悬挂选材不准确	由于拉簧选择不准确，致使底盘过低无法通过 200mm 台阶	因为时间原因就放弃原方案改为独立悬挂。	V3.0	机械工程师：吴伊果

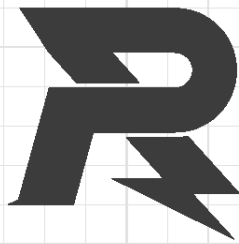
5.3 团队成员贡献

姓名	基本信息	主要负责工作内容描述	贡献度
葛殷	机器人工程	负责合理规划英雄机器人进度安排、协调好各组成员的关系、参与整个英雄机器人的制作，同时提供可行性技术方案	30%
	19 级机器人 1 班		
	队长		
吴伊果	机器人工程	主要负责英雄机器人的结构设计和制作以及后期英雄机器人的维修和保养	20%
	20 级机器人 1 班		
	机械组		
陈杰	机器人工程	主要负责英雄机器人电控代码的编写和完善	20%
	20 级机器人 1 班		
	电控组		
陈洋	软件工程	负责英雄机器人的视觉算法，与电控组成员进行对接	20%
	20 级物联网 2 班		
	视觉组		
熊杨清	机器人工程	负责英雄机器人的单板研发、传感器选型以及超级电容研发	10%
	20 级机器人 4 班		
	硬件组		

6. 参考文献

组别	参考文献	技术收获点
机械组	上海交通大学 RoboMaster2020 赛季英雄机器人开源： https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12241	参考上海交通大学轮组联轴方式以及自适应悬挂
	哈尔滨工业大学（深圳）RoboMaster2020 赛季英雄机器人开源： https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12253	收获了另一种自适应悬挂的设计方案
	大连理工大学 RoboMaster2020 赛季英雄机器人开源： https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12264	参考大连理工大学拨盘以及限位的设计理念
电控组	A new design for a PID plus feedforward controller: https://www.sciencedirect.com/science/article/pii/S0959152403000982	在本文中，提出了一种新的 PID 加前馈控制器的设计和调整程序，它包括确定一个前馈信号，以实现预定义的过程输出转换时间，假设一阶加死区时间模型过程。然后，通过任何常规方法调整 PID 参数以确保良好的负载干扰抑制 和闭环系统的参考信号是通过适当过滤设定点阶跃信号获得的。通过实验结果表明，该方法虽然简单，但优于典型的（逆）基于模型的方法。可以将其研究成果运用在云台 Pitch 轴进行测试。

	<p>A Low-Cost GPS/inertial Attitude Heading Reference System (AHRS) for General Aviation Applications: https://ieeexplore.ieee.org/abstract/document/670207</p>	<p>在本篇参考文献中，我们了解到基于低成本的 AHRS 控制算法，即通过低性能的惯性导航有效地使用仪器来滤除噪音和提供更高的带宽输出。虽然是与 GPS 相关的，但是通过此篇文献，我们进一步了解到 AHRS 在不同需求及情况下的灵活运用，也会我们英雄机器人的姿态解算提供了新的方向与思路。</p>
视觉组	<p>RM2021 西北工业大学 WMJ 战队视觉算法开源-步兵视觉开源 https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12263&fromuid=55730</p>	<p>参考了西北工业大学的 PID 以及卡尔曼算法</p>
	<p>吉林大学 TARS-GO 战队 RoboMaster2020 赛季步兵视觉开源: https://bbs.robomaster.com/forum.php?mod=viewthread&tid=11066&fromuid=55730</p>	<p>参考了吉林大学的数字识别 SVM 模型</p>
硬件组	<p>大连交通大学凌 BUG 战队开源之超级电容: https://bbs.robomaster.com/forum.php?mod=viewthread&tid=6912</p>	<p>借鉴了大连交通大学凌 BUG 战队的超级电容功率控制理念</p>



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镇茶光路1089号集成电路设计应用产业园2楼202