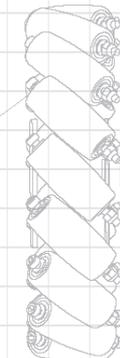




Using a 32-bit motor driver chip and Field-Oriented Control (FOC), the RoboMaster C200 Brushless DC Motor Speed Controller enables precise control over motor torque.



Exclusively designed for the RoboMaster series FOC Brushless DC Motor and C200 Brushless DC Motor Speed Controller, this 12.5mm Axonometric PCB includes several sockets and a terminal board.

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



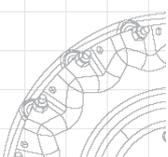
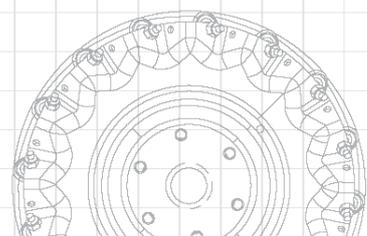
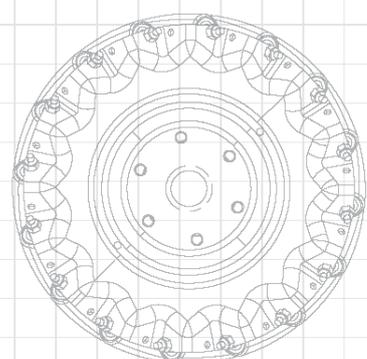
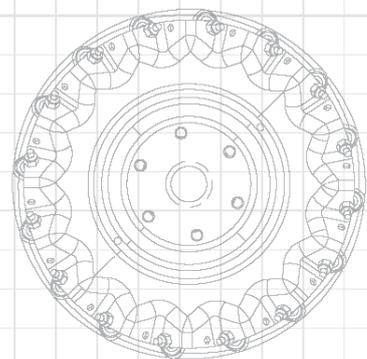
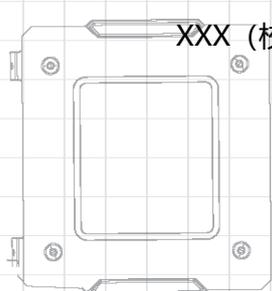
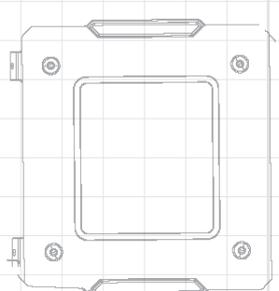
The M2000 Axonometric PCB is a standard PCB and is compatible with other PCBs by their independent design.

# ROBOMASTER 超级对抗赛

## 技术报告

XXX (校名) XXX (队名) 编制

YYYY年MM月 发布



# 目录

<b>1. 步兵机器人</b> .....	<b>5</b>
1.1 其它学校步兵机器人分析综述（2.5） .....	5
1.2 机器人功能定义（5） .....	6
1.3 机器人核心参数（2.5） .....	6
1.3.1 机器人基础参数 .....	6
1.3.2 主要传感器.....	7
1.3.3 超级电容工作参数.....	7
1.3.4 执行器件 .....	7
1.3.5 机器人性能参数 .....	8
1.4 设计方案（25） .....	8
1.4.1 机械结构设计 .....	8
1.4.2 硬件设计 .....	15
1.4.3 软件设计 .....	21
1.4.4 算法设计 .....	30
1.5 研发迭代过程（10） .....	43
1.5.1 版本迭代过程记录.....	43
1.5.2 重点问题解决记录.....	44
1.6 团队成员贡献（2.5） .....	50
1.7 参考文献（2.5） .....	51
<b>2. 哨兵机器人</b> .....	<b>53</b>
2.1 其它学校哨兵机器人分析综述（2.5） .....	53
2.2 机器人功能定义（5） .....	53
2.3 机器人核心参数（2.5） .....	54
2.3.1 机器人基础参数 .....	54
2.3.2 主要传感器.....	54
2.3.3 执行器件 .....	54
2.3.4 机器人性能参数.....	55
2.4 设计方案（25） .....	55
2.4.1 机械结构设计 .....	55
2.5 软件设计 .....	59
2.6 研发迭代过程（20） .....	65
2.6.1 版本迭代过程记录.....	65

2.6.2 重点问题解决记录.....	66
2.7 团队成员贡献（5） .....	68
<b>3. 英雄机器人.....</b>	<b>69</b>
3.1 其它学校英雄机器人分析综述（2.5） .....	69
3.2 机器人功能定义（5） .....	70
3.3 机器人核心参数（2.5） .....	71
.....	71
3.4 设计方案（25） .....	71
3.4.1 机械结构设计 .....	71
3.4.2 硬件设计 .....	79
3.4.3 软件设计 .....	81
3.4.4 算法设计 .....	81
3.5 研发迭代过程（10） .....	81
3.5.1 版本迭代过程记录.....	81
3.5.2 重点问题解决记录.....	82
3.6 团队成员贡献（2.5） .....	83
3.7 参考文献（2.5） .....	84
<b>4. 工程机器人.....</b>	<b>85</b>
4.1 其它学校工程机器人分析综述（2.5） .....	85
4.2 机器人功能定义（5） .....	85
4.3 机器人核心参数（2.5） .....	86
4.4 设计方案（25） .....	89
4.4.1 机械结构设计 .....	89
4.4.2 硬件设计 .....	100
4.4.3 软件设计 .....	103
4.4.4 算法设计 .....	106
4.4.5 UI 设计 .....	109
4.5 研发迭代过程（10） .....	111
4.5.1 版本迭代过程记录.....	111
4.5.2 重点问题解决记录.....	111
4.6 团队成员贡献（2.5） .....	113
4.7 参考文献（2.5） .....	114
<b>5. 飞镖系统 .....</b>	<b>115</b>
5.1 其它学校飞镖系统分析综述（2.5） .....	115
5.2 飞镖系统功能定义（5） .....	115

5.3 飞镖系统核心参数 (2.5)	116
5.4 设计方案 (25)	117
5.4.1 机械结构设计	117
5.4.2 硬件设计	142
5.4.3 软件和算法设计	148
5.5 研发迭代过程 (10)	152
5.5.1 版本迭代过程记录	152
5.5.2 重点问题解决记录	153
5.6 团队成员贡献 (2.5)	157
5.7 参考文献 (2.5)	157
<b>6. 雷达</b>	<b>159</b>
6.1 其它学校雷达站分析综述 (2.5)	159
6.2 机器人功能定义 (5)	160
6.3 机器人核心参数 (2.5)	160
6.4 设计方案 (25)	161
6.4.1 机械结构设计	161
6.4.2 硬件设计	162
6.4.3 软件设计	164
6.4.4 算法设计	168
6.5 研发迭代过程 (10)	182
6.5.1 版本迭代过程记录	182
6.5.2 重点问题解决记录	183
6.6 团队成员贡献 (2.5)	185
6.7 参考文献 (2.5)	185

# 1. 步兵机器人

## 1.1 其它学校步兵机器人分析综述（2.5）

从论坛上的各种开源文档来看，这个赛季有许多战队正在研究甚至已经研究出了舵轮步兵，相对于普通步兵，舵轮底盘具有机动性强、弹丸躲避率高的优势，是地面对战强势单位。通过击杀当下最首要的敌方目标，舵轮步兵能控制比赛节奏，例如击杀敌方英雄、工程、哨兵，干扰敌方打符等，并保护英雄推倒敌方基地。

同时，本队伍也注意到从去年开始，双枪管步兵也开始逐渐崭露头角。双枪管步兵可以在比赛前期集中火力优势，迅速击破敌方机甲，争取到经验优势和数量优势。但是双枪管的射频以及射击命中率有时甚至达不到单枪管步兵的水准，所以双枪管更适合近距离的输出，最好是与单枪管配合作战，以达到优势互补的作用。

不论是何种类型的步兵，其功能都大同小异，经对比赛观摩和论坛文献的查阅，我们对其他学校的步兵功能有如下分析：

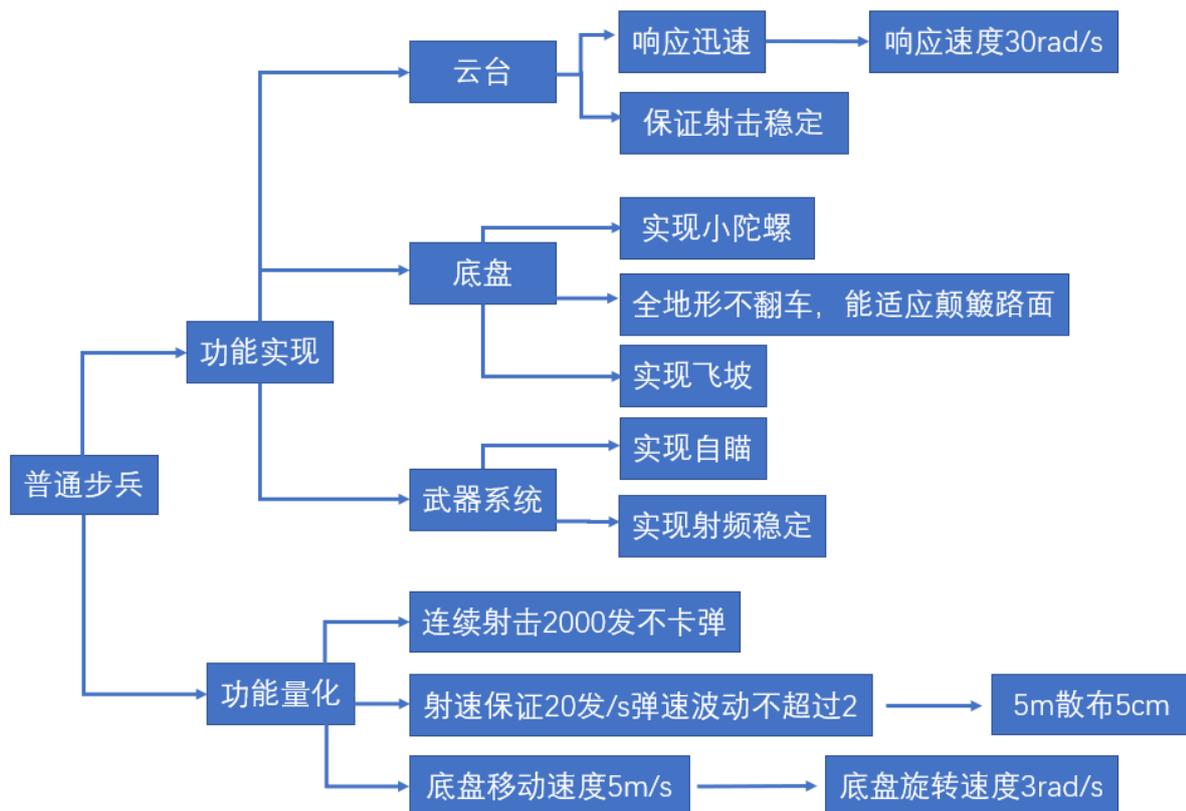
小陀螺：这一功能已是攻击型单位很基础的功能了，有一定积累的战队都已实现此功能甚至搭配超级电容获得更好的性能。

自瞄打符：一些有实力的队伍对于此技术的掌握已经成熟，一些队伍仍在开发，这一功能对于比赛节奏的把控有着十分决定性的作用，能够实现自瞄打符将很大程度上提高己方的主动性。

卡弹情况：查阅论坛大量资料发现，拨弹轮卡弹貌似是一个不可避免的问题，问题在于如何减小这种概率，有的队伍选择电机反转解决卡弹，有的队伍选择调整弹速，寻求最小的卡弹几率。我们也在多次尝试下发现无法彻底解决卡弹问题，所以改变策略。

以上是本队伍截取的比较有共通性的步兵概况，对于一些比较小众的以及对于我们队伍没有较大提升的内容，则没有收录。通过对于其他学校的了解，我们认识到了自己与强队的差距，也知道了自己的努力方向，目前我们正在逐步优化我们的双枪管步兵，期待能够有一个好的效果。

## 1.2 机器人功能定义（5）



## 1.3 机器人核心参数（2.5）

### 1.3.1 机器人基础参数

参数类型	数值
尺寸 (mm,L*W*H)	558*490*489
重量 (kg)	20.5
重心高度(mm)	210

### 1.3.2 主要传感器

型号	参数	数量
BMI088	BMI088 集成了 16 位三轴陀螺仪和 16 位三轴加速度计。具有低于 2/h 的偏置稳定性和低于 15mdps/K 的低温度系数偏移，以及在 ±24g 的测量范围内仅 230ug/√Hz 的低频谱噪声。	1
工业相机	MV-SUA134GC-T 工业相机与 MV-LD-4-4M-G 工业镜头, 在 1280*1024 的图像尺寸下帧率可以达到 211FPS。视场角为 82.9°x66.5°x94°。采用全局快门, 拖影少。	1

### 1.3.3 超级电容工作参数

(1) 底盘功耗最大可达 200W，电容管理电路应提供最大 150W 的功率。

(2) 超级电容组选择 8 个 2.7V30F 和 6 个 2.7V50F 的电容全串联，等效为 37.8V2.59F 电容，能量 1968.3J，在限制值以下。

(3) 当电容电压高于 38V，认为充满，不再充电；当电容电压低于 9V 时（能量占比仅为 5%），关闭电容，不再放电，以防止电容放电电流过大烧毁电路，并且向主控发送标志。

### 1.3.4 执行器件

类型	用途	数量
RoboMaster M3508 减速直流电机	用于底盘移动的动力输出。	4
RoboMaster M3508 减速直流电机（去掉减速箱）	用于发射机构摩擦轮的动力输出。	2

RoboMaster GM6020	用于云台 yaw 轴和 pitch 轴移动的动力输出。	2
RoboMaster M2006 P36 直 流无刷减速电机	用于拨弹盘供弹机构的动力输出。	1

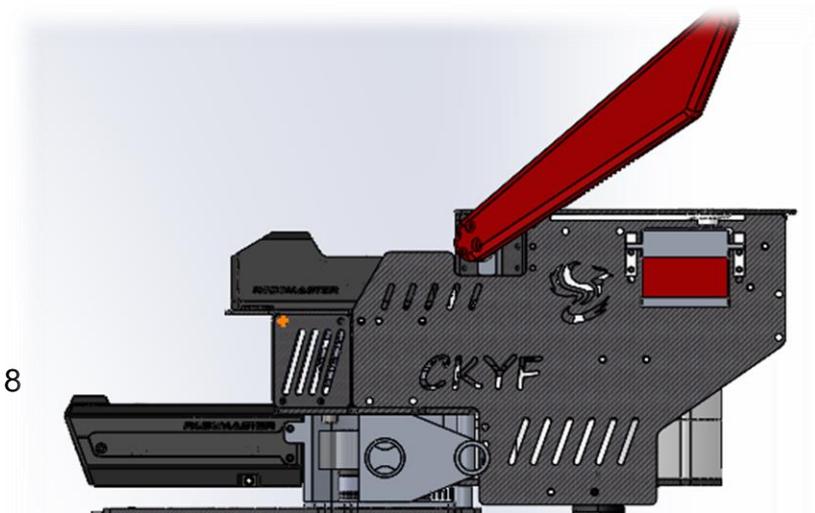
### 1.3.5 机器人性能参数

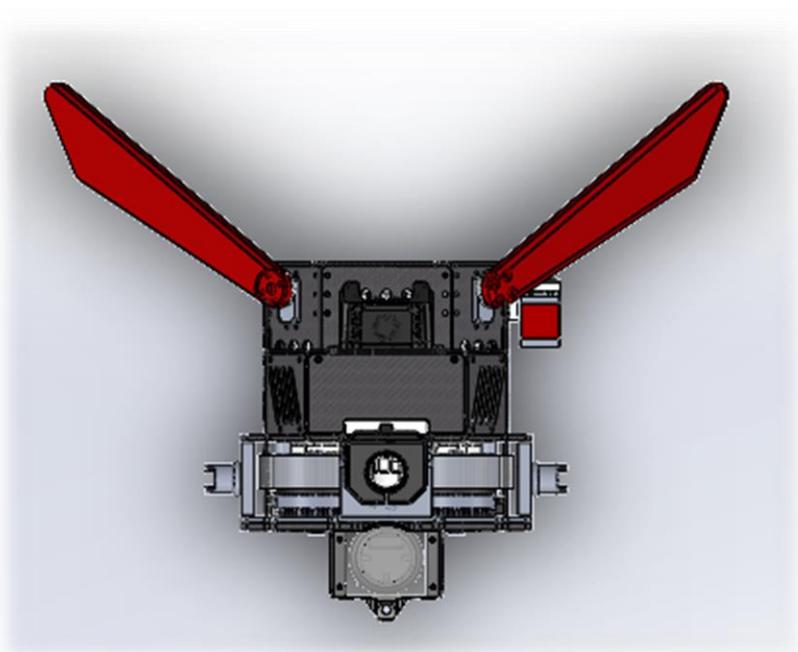
参数类型	数值
车体最大移动速度	4m/s
最大爬坡角度	30°
云台自由度	yaw 轴 360° 旋转；pitch 轴±25° 俯仰
Yaw 轴旋转速度	1.5r/s
最大射速	30m/s
命中率	5m 小装甲命中率≥98%
弹仓容量	≥300 发

## 1.4 设计方案（25）

### 1.4.1 机械结构设计

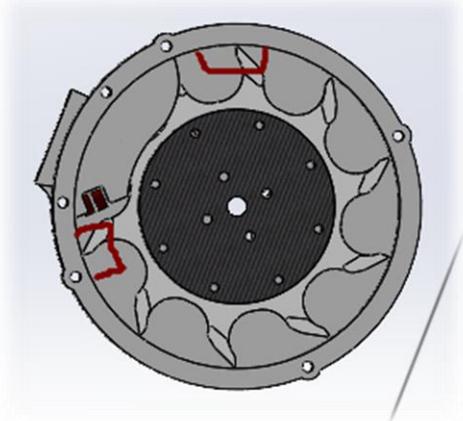
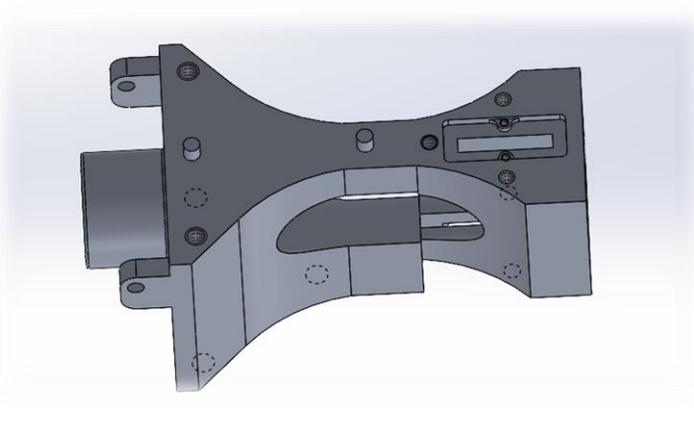
#### 1.4.1.1 云台设计





#### 1.4.1.1.1 发射机构设计

发射机构采用无枪管设计，动力部分使用去除减速箱的 3508 电机作为摩擦轮电机。在发射机构上采用波珠螺丝作单发限位。

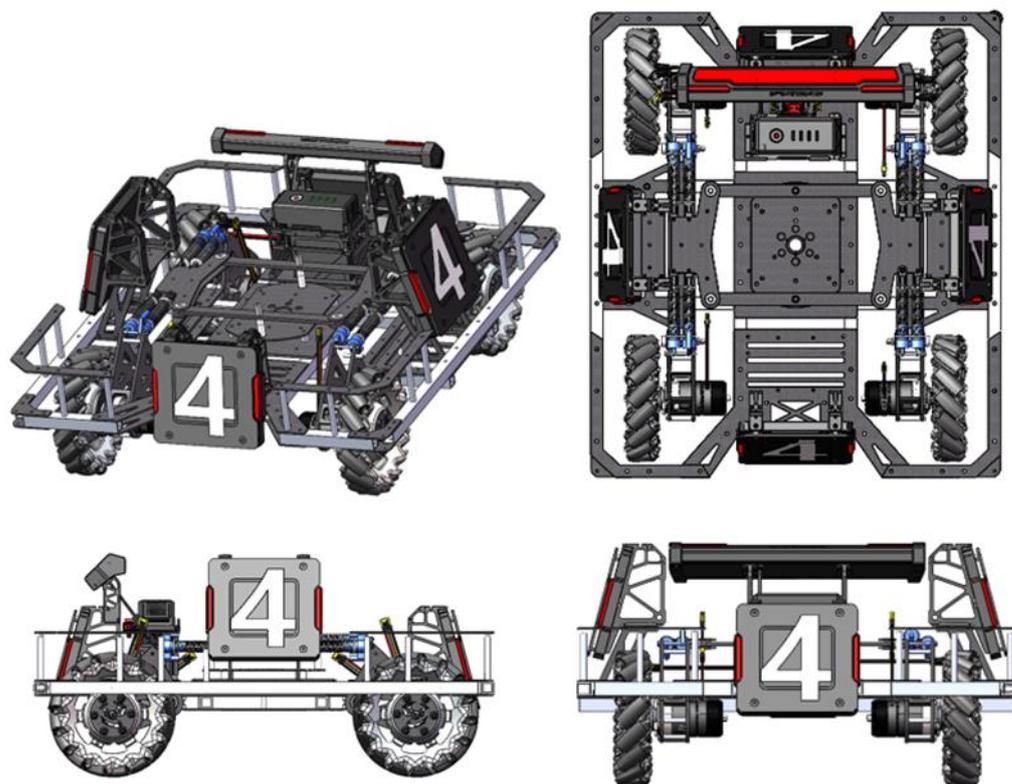


#### 1.4.1.1.2 拨弹盘设计

为解决卡弹问题，本赛季经过多次试验，最终采用挡片加毛刷方案（如右上图红色区域内是毛刷区域），虽未在机械上完全解决卡弹问题，但是已经将卡弹的几率大大降低了。经过测

试概率大概 1/2500。如若卡弹便在电控方面将拨弹盘反转消除卡弹。经实战证明，此方案的拨弹盘未在比赛期间卡过弹。

### 1.4.1.2 底盘设计



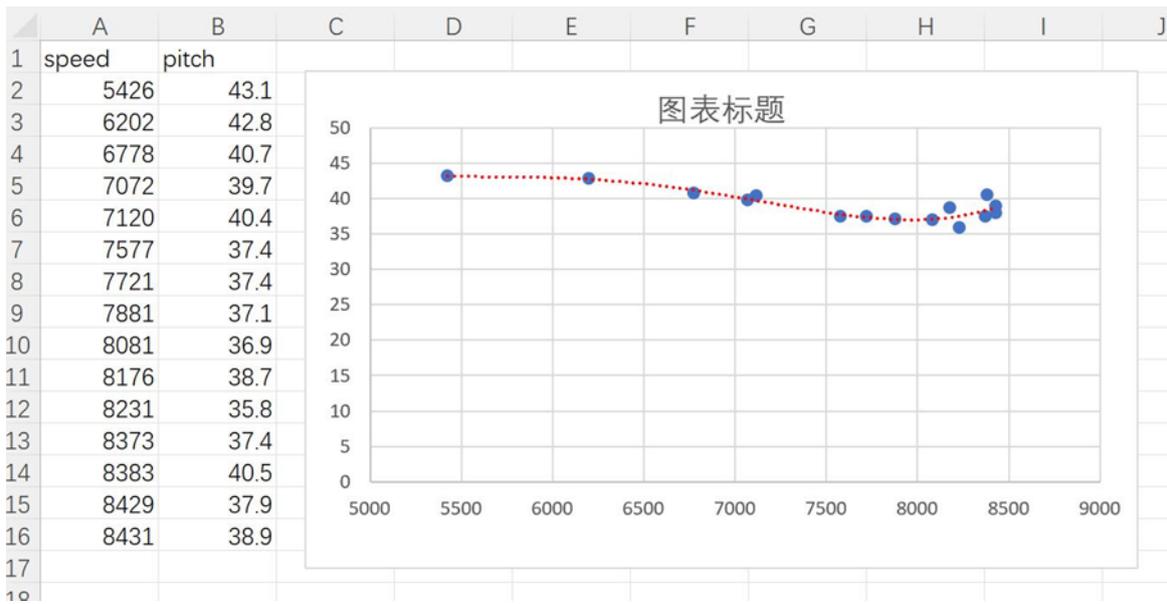
#### 4.1.2.1 整体承力框架

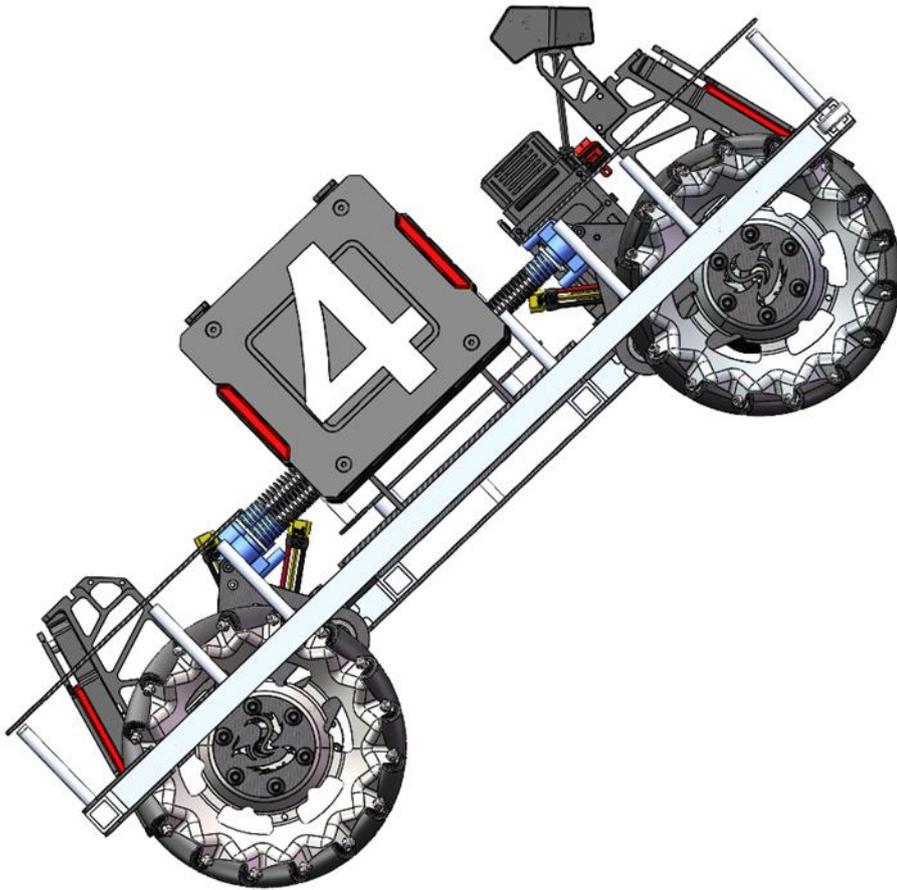
首先根据 Robomaster2022 机甲大师高校系列赛机器人制作规范手册的规定，步兵机器人的最大初始尺寸为 600\*600\*500mm,考虑到结构紧凑和运动灵活的设计需要，最终确定步兵在地面的正投影尺寸为 490\*560mm。

针对承力框架的铝方选型，由表格计算得出不同规格铝方的抗弯强度所对应的最大载荷，最后选取 15\*15\*2mm 的铝方。

铝方选样参数表							
1	H	b	lz	S	质量比强度	Mmax	
2	10	1.8	693.5232	59.04	11.74666667	28.4344512	
3	10	2	725.3333	64	11.33333333	29.73866667	
4	12	1	894.6667	44	20.33333333	30.56777778	
5	12	2	1386.667	80	17.33333333	47.37777778	
6	15	1	1838.667	56	32.83333333	50.25688889	
7	15	2	2998.667	104	28.83333333	81.96355556	
8	20	1	4585.333	76	60.33333333	93.99933333	
9	20	1.5	6373.25	111	57.41666667	130.651625	
10	20	2	7872	144	54.66666667	161.376	

为了满足飞坡的需要，在设计中对框架距离地面的高度作出调整。首先针对飞坡进行定速测试，控制步兵以不同初速度离开斜坡，观测步兵触地时的整体姿态和触地角度，为底盘保护框高度、麦克纳姆轮与保护框的径向间隙设置提供参考。结果显示最大触地角可达到43°左右，因此将保护框高度定为麦克纳姆轮直径75%处附近较为合适，并控制保护框与麦克纳姆轮的径向间隙在5mm左右。

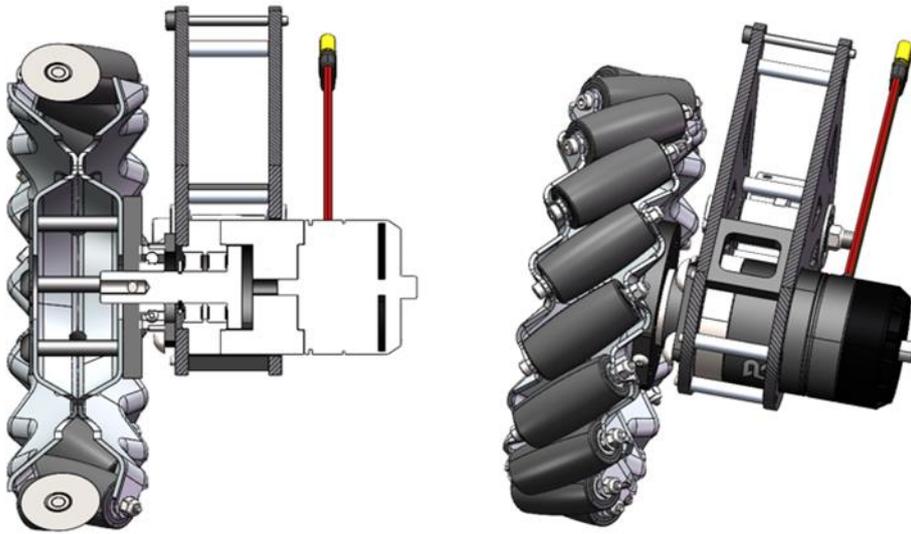




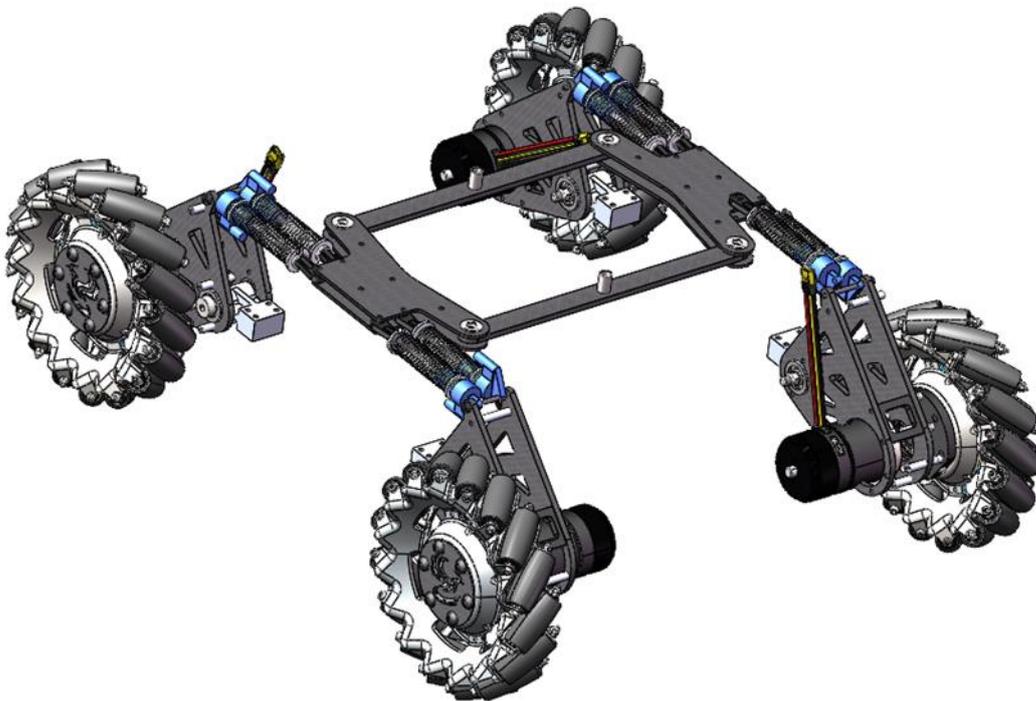
#### 1.4.1.2.2 自适应悬挂系统

为了提高整车悬挂在复杂地形下的稳定性，采用由平行四边形自适应连杆、避震器和轮系构成的自适应悬挂系统。

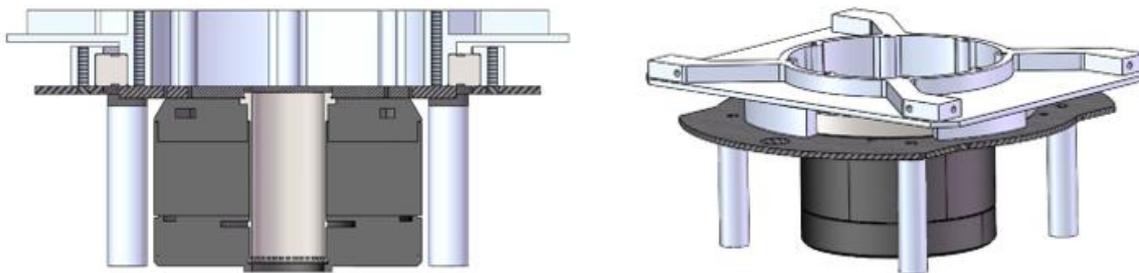
在轮系设计方面，为了尽可能的缩短沿轮轴方向的长度，采用了将 3508 电机嵌入两层碳板中间的方式，靠近电机输出端的碳板开孔直径小于电机外径，与轴承座共同配合，起到轴向固定的作用，而法兰盘既与轴承内圈配合又通过侧面的通孔用螺丝与 3508 电机的输出轴连接，在尽可能缩短电机输出轴到麦克纳姆轮的距离的同时，还保证了两者的同心度和轮轴连接的刚度。



在自适应连杆设计方面，由于在设计时需考虑底盘高度的影响，因此为了避免与云台 yaw 轴 6020 电机干涉，扩大连杆间距，将云台从连杆构成的平行四边形中间穿过，同时在此高度基础上分析避震器的最大压缩量应控制在 25mm 内，由此选择长度 100mm，劲度系数 4N/mm 的全金属负压避震器。



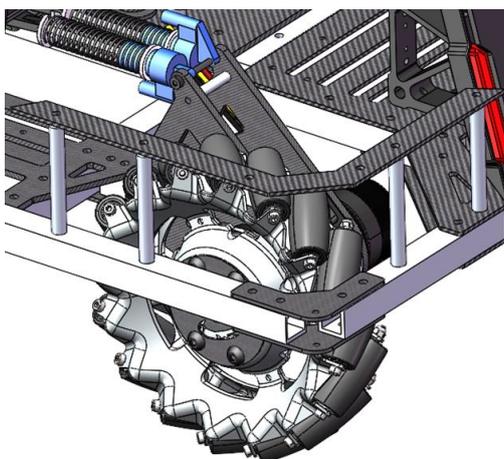
### 1.4.1.2.3 Yaw 轴小陀螺设计



为了实现云台 yaw 轴 360° 旋转的小陀螺功能，采用 2A、24 线小滑环，将滑环安装在 6020 内部，降低云台整体高度。6020 电机在 yaw 轴处的连接需通过 RA8008 交叉滚子轴承来分离云台和底盘的运动，其中电机末端与底盘固定，输出端与一块碳板相连并压住轴承内圈的下端，轴承内圈的上端由铝合金 CNC 加工件压紧，同时该铝合金件还承载整个云台的重量，轴承外圈的下端先是由一块通过铝柱支撑的碳板压紧，该碳板与电机输出端并无连接关系，同时还具有轴承座的安装孔，使得四个 L 形轴承座压紧轴承外圈的上端。这样既保证了 yaw 轴的自由度，又使得承载云台重量的传力路径可以分散传递到底盘上，避免了在运动过程中出现较为明显的应力集中，具有良好的可靠性。

### 1.4.1.2.4 救援框架设计

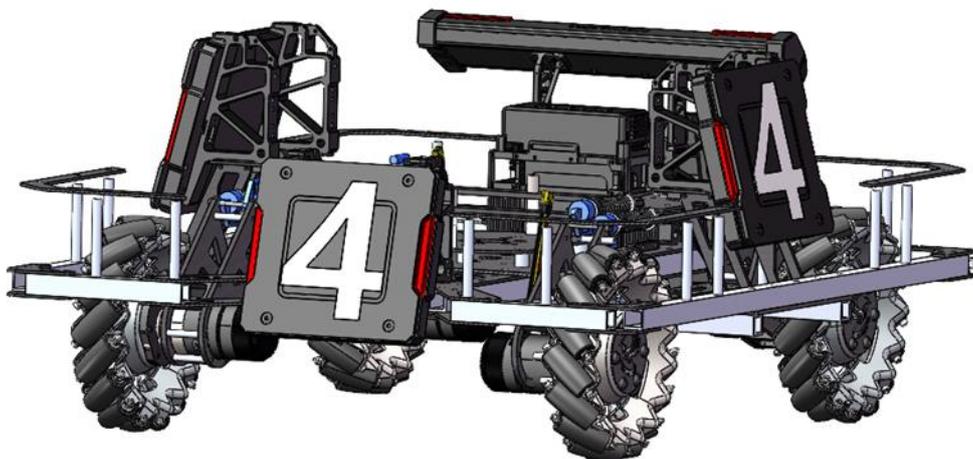
经过与英雄机器人和工程机器人的统一考量，确定救援机构的间距为 350mm，在此基础上，对侧面与前后的保护框钻孔并安装铝柱，另外再基于铝柱添置由碳板制成的救援框，既能适应工程机器人从两侧伸出钩爪的救援方式，又能满足从上方钩住救援框的救援方式。



### 1.4.1.2.5 裁判系统安装

根据 Robomaster2022 机甲大师高校系列赛机器人制作规范手册的规定，对于步兵机器人，

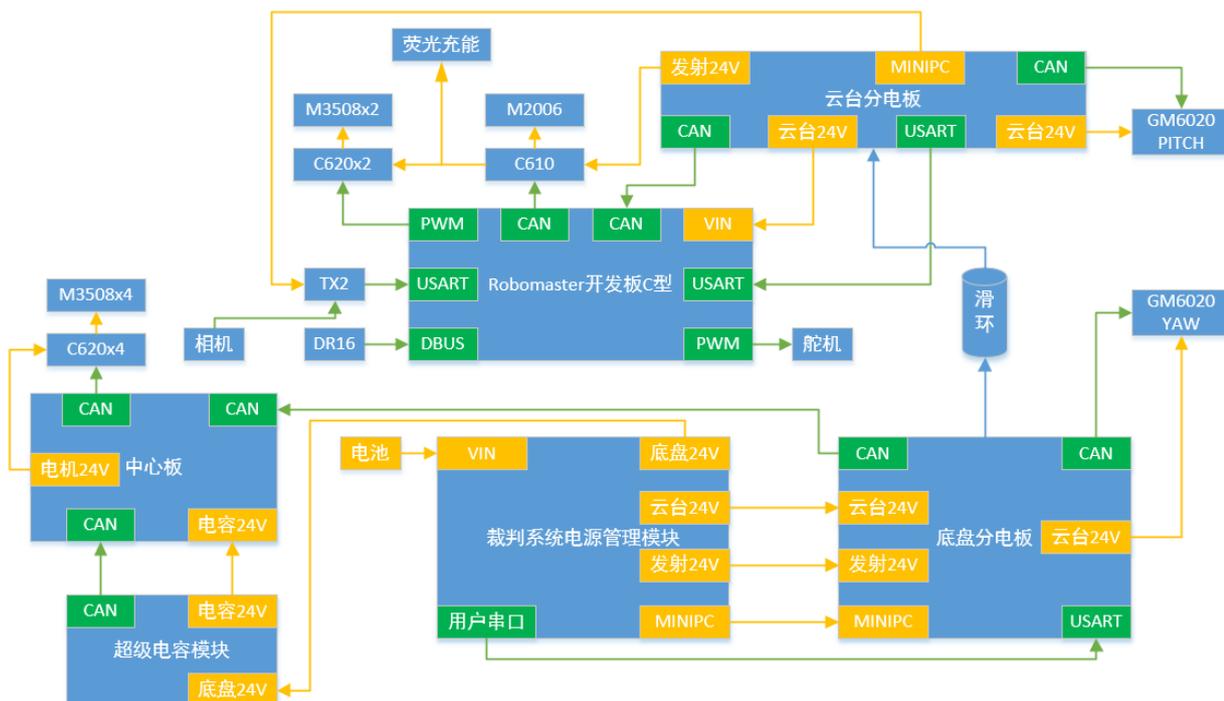
变形前后，装甲模块下边缘距离地面高度必须在 60mm - 150mm 范围内。因此应尽可能使前后装甲与侧面装甲处在最大高度差处，从而降低敌方机器人的装甲命中率。设置前后装甲与侧面装甲下边缘距离地面的高度分别为 80mm 和 145mm。



## 1.4.2 硬件设计

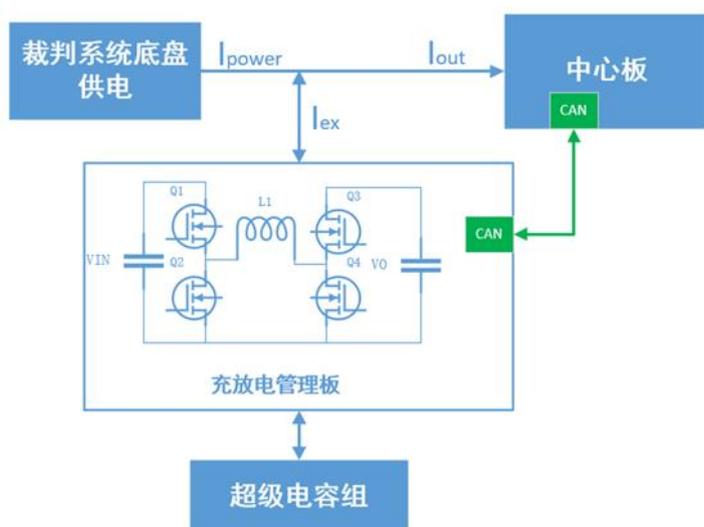
硬件方面主控选择 RoboMaster 开发板 C 型作为云台和底盘的主控模块，NVIDIA Jetson TX2 作为计算平台，自研基于 STM32F3 的超级电容控制模块。通过滑环和两块自制分电板，连接云台和底盘电路，以实现云台 360 度旋转。

### 1.4.2.1.整机硬件框图



### 1.4.2.2 (超级电容) 硬件详细设计

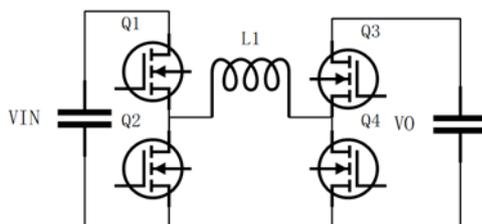
#### 1.4.2.2.1 超级电容硬件框图



## 1.4.2.2.2 电路详细设计

### 1.4.2.2.2.1 主电路设计

主电路采用四开关管 BUCK-BOOST 电路，该电路可以输出较高电压，因此超级电容组的满电电压可以选取较高，放电到相同低电压时，能量利用率更高。



采用简单的对角导通方案，占空比  $D$  定义为  $Q1$  和  $Q4$  导通时间占  $T_s$  的比例，由伏秒平衡可得：

$$V_{IN} \times D \times T_s = V_O \times (1 - D) \times T_s = L_1 \times \Delta I$$

输入端电流和电感电流的关系为：

$$I_{IN} = I_L \times D$$

底盘功率最大取 200W，电容电路需要为母线提供约 150W 的功率，则最大取 6A（方向为输出到母线），由于电容端电压较低时电容的输出电流会很大，越低压，能量释放越快，所以限制  $D$  最小到 0.3，则最大可达 20A，需要非常注意电感的选型。

一般  $\Delta I_L = I_L \times 0.4$ ，开关频率取 100KHz，联立各式，可得电感  $L_1$  取 10uH。

滤波电容主要采用电解电容，在低频下其电压纹波主要由 ESR 引起，电压纹波取 0.2V，电流纹波  $\Delta I_L$  由上分析取 8A，则 ESR 约取 30mΩ。一般来说：

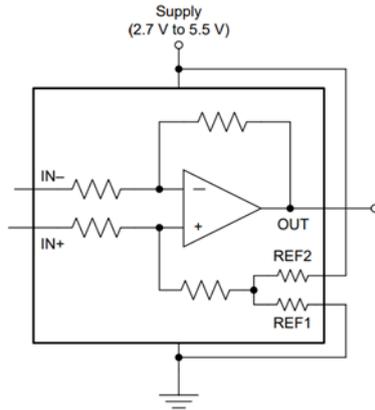
$$C \times ESR = 50 \sim 80 \times 10^{-6}$$

因此最终取滤波电容 2200uF。

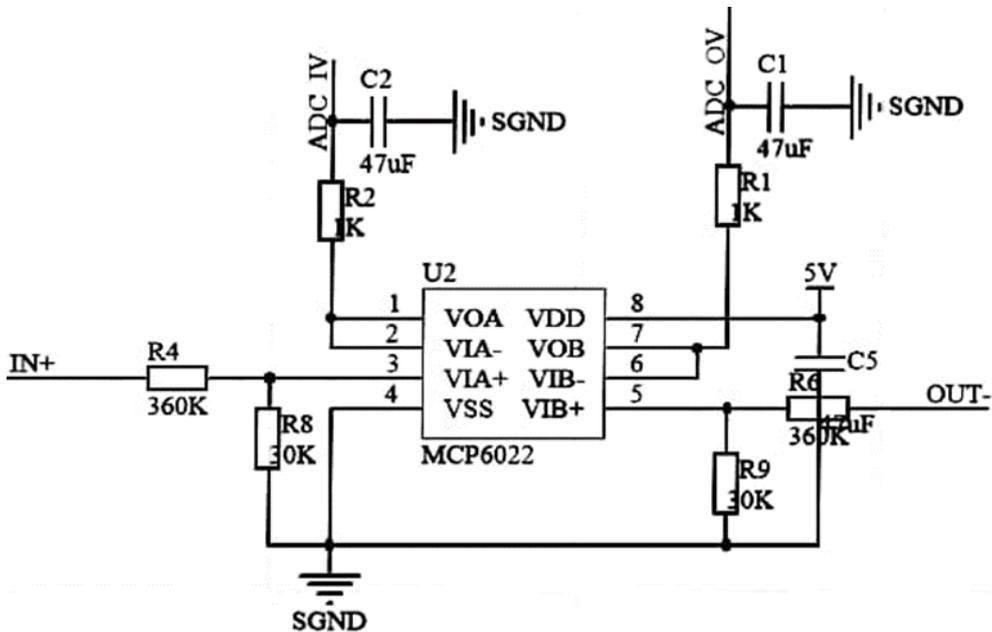
### 1.4.2.2.2.2 调理电路设计

需要采集的信号为母线三端口处的三个电流，电容管理电路的两端电压。

电流采样采用电阻法，电流信号转换成电压信号，再用差分放大电路进行放大，ADC 再将模拟信号转成数字信号。直接用集成芯片，简化 PCB 布局，提高稳定性。



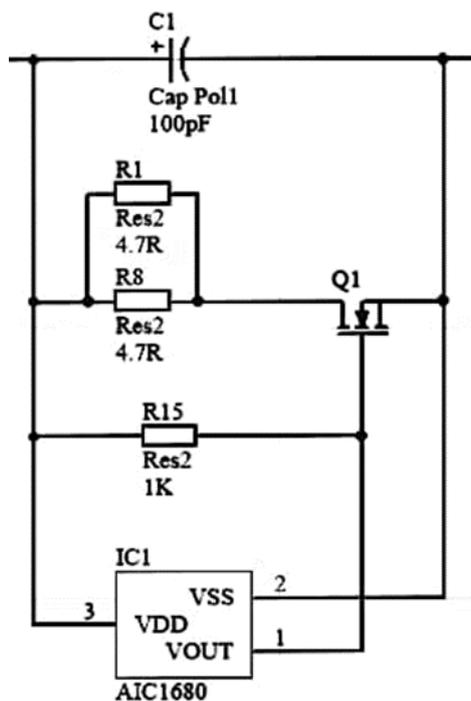
电压采样采用电阻分压法，将较高电压分压到单机电压，再接跟随器匹配阻抗，ADC 再将模拟信号转成数字信号。



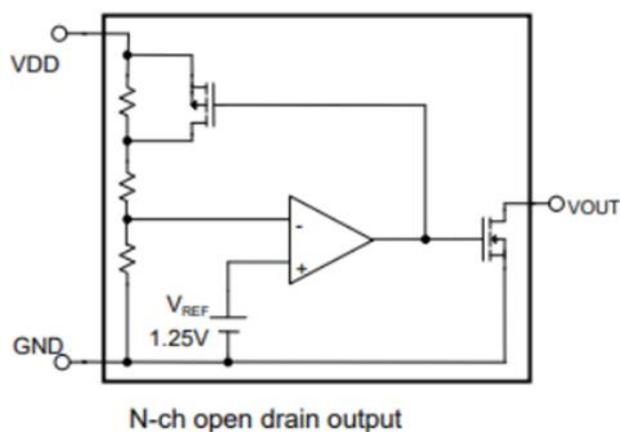
### 1.4.2.2.2.3 超级电容组及其均压电路

采用降压充电、升压放电的方式，结合规则辅助电源容量不得超过 2000J，最终确定超级电容组规格为 2.7V 30F 电容 8 个、2.7V 50F 电容 6 个串联，等效为 37.8V 2.59F 电容。

为了防止充电过程中各电容的充电程度不同导致单个电容过压而损坏，需要设计均压电路，电路如下：



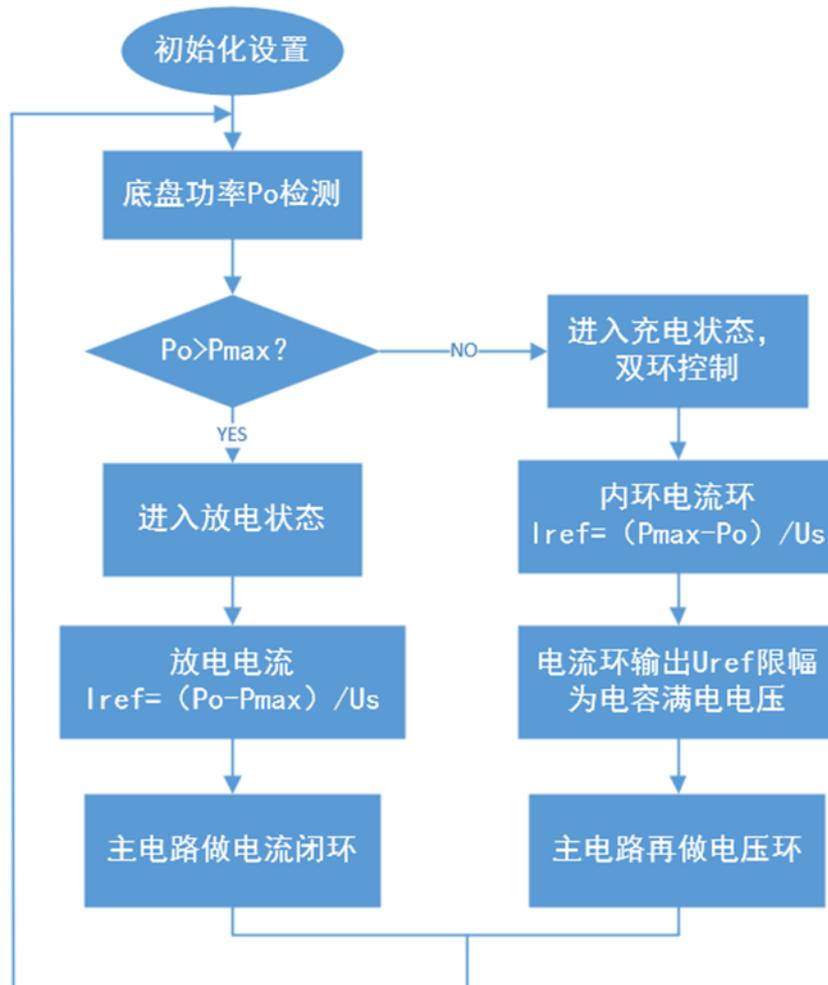
使用 AIC1680 电压检测芯片，基准电压为 2.7V。当检测到电压超过 2.7V 时，芯片输出端电平拉高，从而开启 MOS 管进行泄流。



AIC1680 芯片，N 沟道输出，开漏，需要上拉，当电压 VDD 加到 2.7V，比较器将输出负压，使得上边的 MOS 开通，则比较器负输入端电压进一步升高，输出进一步降低，OUT 处的 MOS 关闭，OUT 上拉输出高电平。

### 1.4.2.2.3 控制方案及框图

主要为电流环控制，通过控制电容管理电路的电流来控制电池的功率。充电模式还需要电容电压做外环；放电模式下，和母线并联，电压取决于母线，所以只需要做电流环。程序框图如下：



### 1.4.2.3 关键器件选型

主控平台使用 RoboMaster 开发板 C 型上的 STM32F407, 具有工作频率 168MHz 的 Cortex™-M4 内核, 算力充足, 外设丰富, 能够满足机器人的控制需求。

IMU 传感器使用 RoboMaster 开发板 C 型板载的 BMI088 传感器, 集成了 16 位三轴陀螺仪和 16 位三轴加速度计。具有低于 2/h 的偏置稳定性和低于 15mdps/K 的低温度系数偏移, 以及在 ±24g 的测量范围内仅 230ug/√Hz 的低频谱噪声。经过校准后, 运用在云台上, 可以实现几乎没有角度漂移, 且角速度测量响应快。如果使用 RoboMaster 开发板 A 型的 MPU6500, 则云台 YAW 轴易出现较大的漂移。

超级电容的调理电路中, 电流采样芯片采用 INA240, 高精度 (可达 0.5%), 高共模电压, 具有 PWM 抑制功能等。而且内置偏置电路, 只需要变换引脚接法即可实现 1/2Vcc 的偏置, 方便 PCB 布局设计。电压跟随器采用运放集成芯片 MCP6022, 具有高精度、低偏置电

流、高带宽、高压摆率等特点。

超级电容的主控采用 STM32F334 芯片，该芯片主要服务于数字电源，具有高精度定时器（4GHz）、高速 ADC（5MSPS）、硬件 FPU 等，可以满足高精度的数字电源控制。此外具有串口、SPI、CAN 等多种通信功能，实现与主控和裁判系统之间的通信。

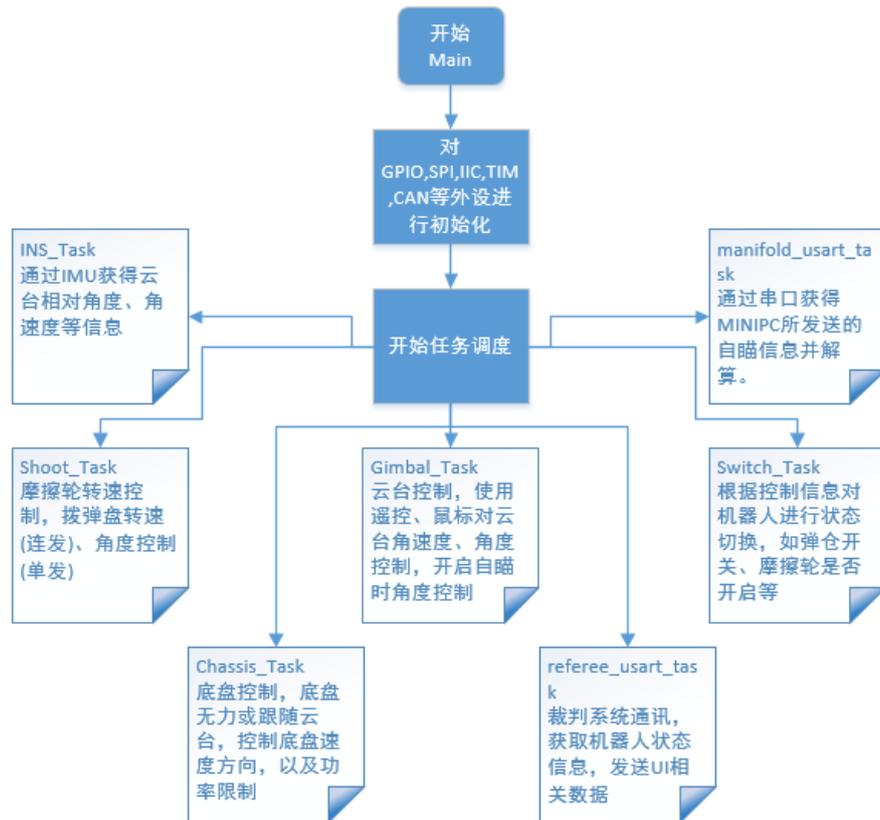
## 1.4.3 软件设计

### 1.4.3.1 系统架构

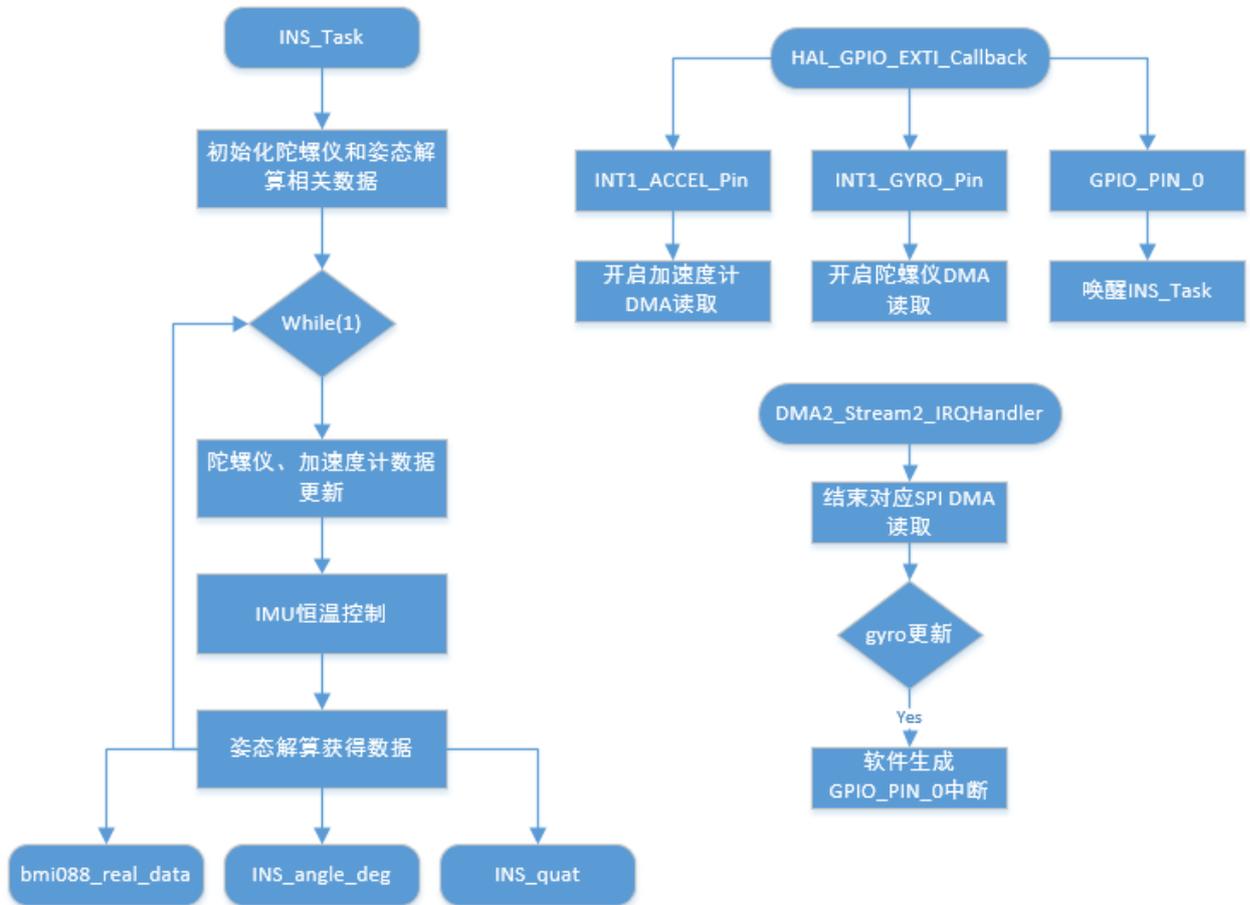
嵌入式软件部分，分为硬件抽象层 (HAL)、操作系统层 (RTOS) 和应用层。硬件抽象层使用 CubeMX 软件配置，STM32 HAL 库编写，参考 Robomaster 开发板 C 型开源程序，重新构建了较为简洁的源码。操作系统层使用 Freertos 进行任务调度，在不同的任务 (Task) 中进行机器人不同部分的控制。应用层主要包括底盘控制、云台及发射控制、传感器数据处理、自瞄计算等。开发环境为 STM32CubeMX+MDK ARM，使用 MDK 的调试功能和串口等观察数据，进行调试。

### 1.4.3.2 运行流程

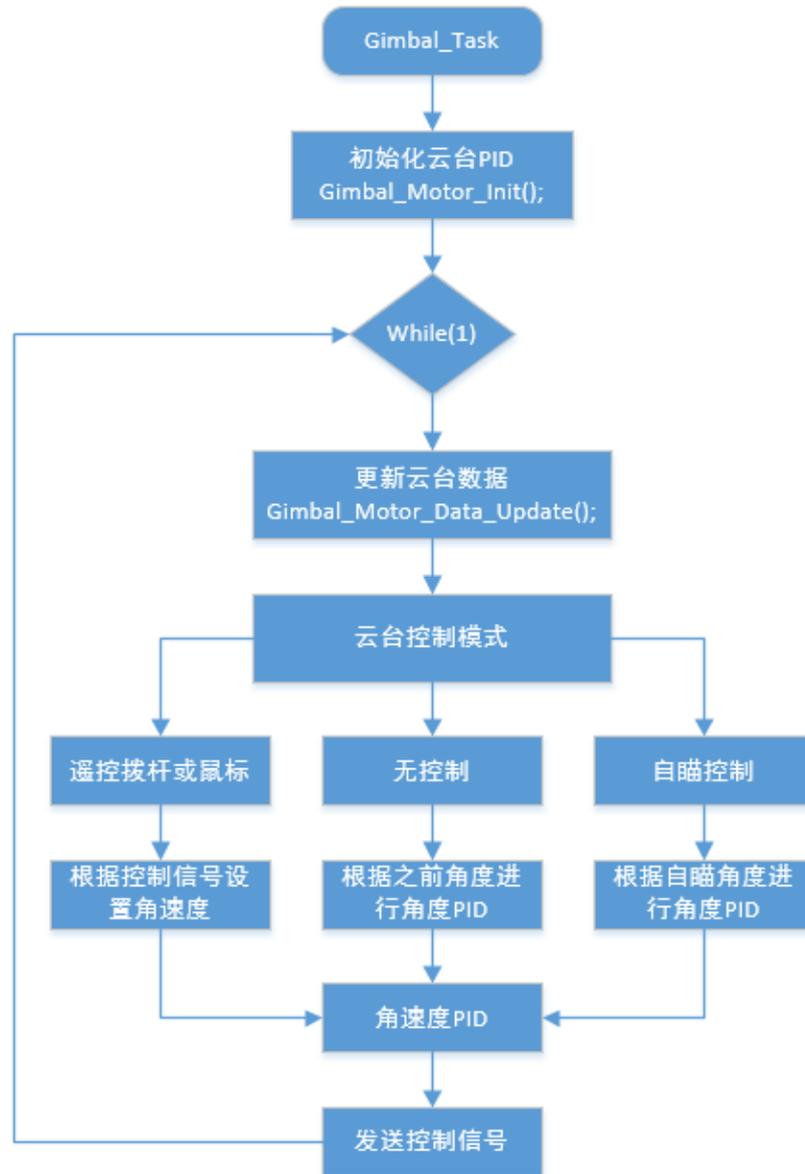
嵌入式软件部分，在主函数中进行外设初始化后，即开启 Freertos 任务调度，目前主要分为 7 个控制任务。



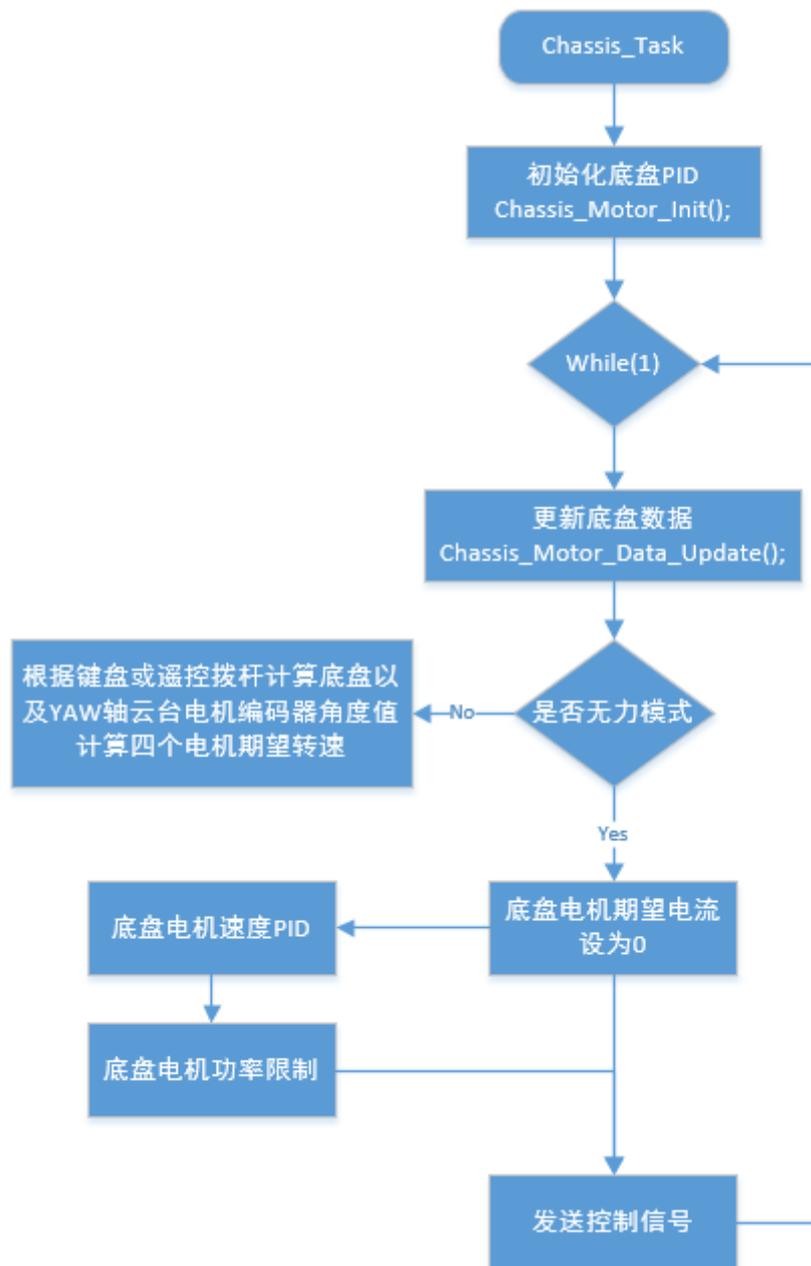
在 INS\_Task 中，通过 IMU 获得云台相对角度、角速度等信息，主要流程如下，使用了外部中断和 DMA 中断进行处理。首先进行六轴陀螺仪 BMI088 和姿态计算相关数据参数的初始化。然后进入主循环，在这里进行陀螺仪和加速度计数据的更新，并进行姿态计算，获取角速度(bmi088\_real\_data),角度(INS\_angle\_deg)和四元数矩阵(INS\_quat)。其中角速度和角度主要用于云台的控制，四元数矩阵用于自瞄计算中的坐标系变换。



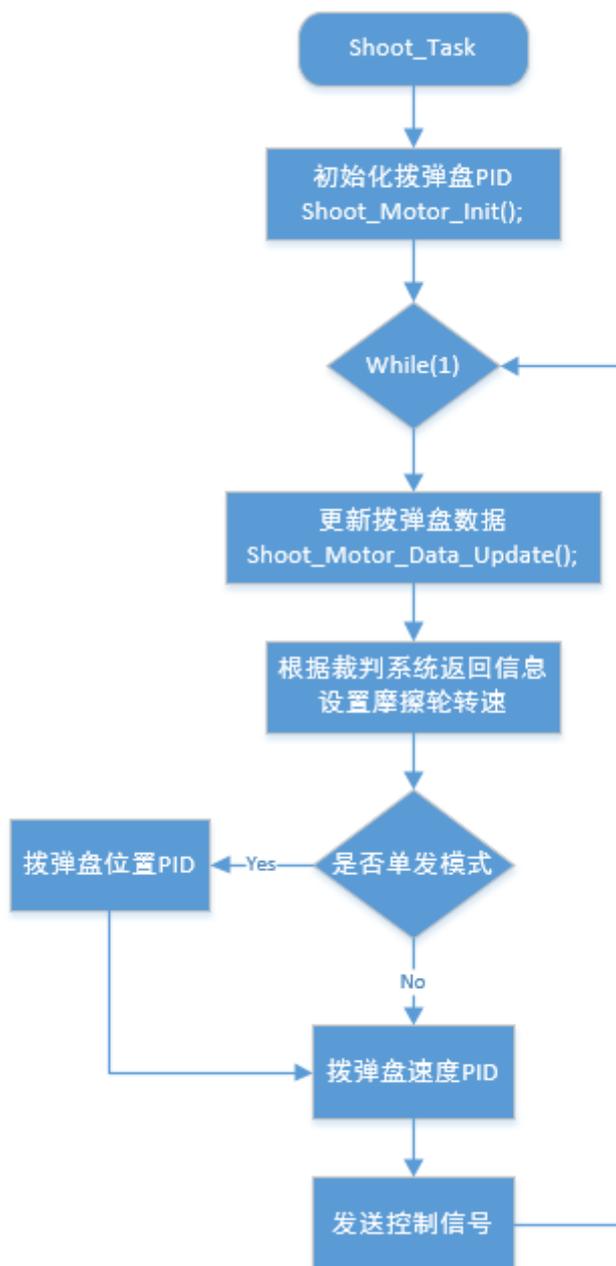
在 `Gimbal_Task` 中，进行云台控制，包括使用遥控、鼠标对云台角速度、角度控制，以及开启自瞄时的角度控制，主要流程如下。首先进行云台初始化，主要为设置初值和 `PID` 参数初始化，然后进入主循环。循环中，首先更新云台数据，包括角度、角速度。然后根据不同的控制模式进行处理。如果鼠标或遥控拨杆有控制，则根据这个控制信号设置角速度，不进行角度环计算。如果没有控制数据，则保持上一刻的角度，根据这个角度进行角度环 `PID` 计算。如果自瞄有识别到目标且操作手开启自瞄，则根据自瞄数据计算出的角度进行角度环 `PID` 计算。角度环 `PID` 计算得出期望角速度。之后，根据以上几种控制模式下的期望角速度，进行速度环 `PID` 计算，获得期望的电机电流值。



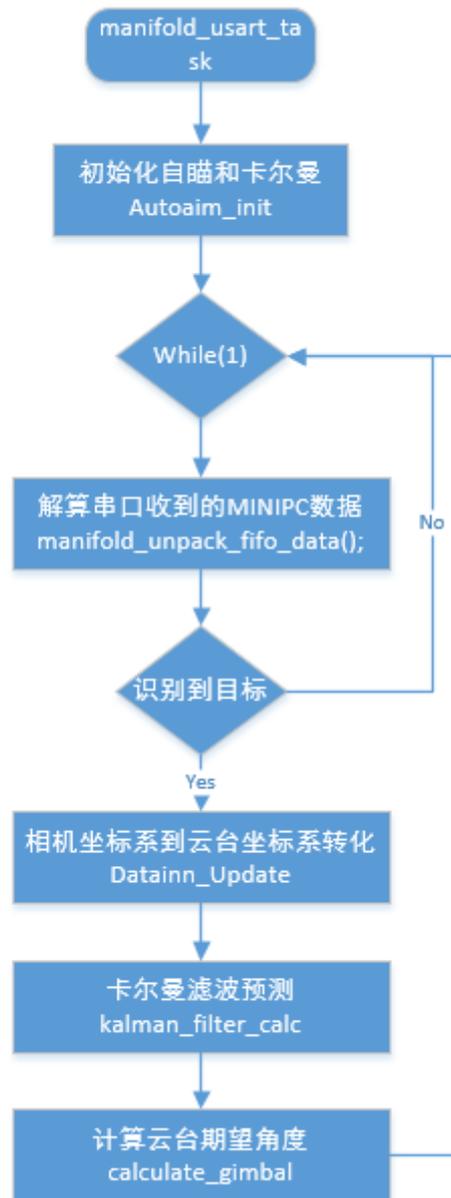
在 Chassis\_Task 中，底盘控制，主要模式为底盘无力或跟随云台，控制底盘速度方向，以及功率限制，主要流程如下。首先进行底盘初始化，主要为设置初值和 PID 参数初始化，然后进入主循环。循环中，首先更新底盘数据，包括底盘电机转速、YAW 轴云台电机编码器角度(用于底盘跟随云台)。然后根据不同的控制模式进行处理。如果底盘为无力状态，则直接设置底盘电机电流值为 0。否则，根据键盘或遥控拨杆的控制，计算机器人 XY 方向期望速度。如果机器人在小陀螺模式，则设置底盘旋转速度；否则，根据 YAW 轴云台电机编码器角度进行 PID 计算期望的底盘旋转速度。之后根据 XY 方向速度和底盘旋转速度计算四个底盘电机分别的期望转速。根据期望转速进行速度环 PID 计算，获得期望的底盘电机电流值。之后，根据超级电容状态和裁判系统返回功率，计算底盘实际的期望电流。



在 Shoot\_Task 中，进行摩擦轮转速控制，以及拨弹盘的控制，主要流程如下。其中，根据裁判系统信息，获得弹丸速度，设置摩擦轮转速。拨弹盘在 Task 开始时，进行 PID 的初始化，根据操作手操作，如果为单发模式，则首先进行位置 PID 的计算；如果为连发模式，则直接设置摩擦轮转速。之后进行拨弹盘速度环 PID，获得期望的电机电流。



在 `manifold_usart_task` 中，通过串口获得 MINIPC 所发送的自瞄信息并解算，具体流程如下。首先，初始化自瞄和卡尔曼滤波相关参数数据。在主循环中，解算串口收到的 MINIPC 数据，表示方法为相机坐标系下的目标位置。如果识别到了目标，则将目标位置通过 IMU 的四元数进行坐标变换，将目标位置映射到云台坐标系下，接着根据云台坐标系的位置，计算自瞄模式下云台的期望角度。



referee\_usart\_task 为裁判系统通讯任务，主要获取机器人状态信息，发送 UI 相关数据。

Switch\_Task为状态切换任务，根据键鼠和遥控控制信息对机器人进行状态切换，如弹仓开关、摩擦轮是否开启等

### 1.4.3.3 重点功能

#### 1. 自动瞄准和卡尔曼滤波、预测

由于比赛中画面延迟相对较高，且使用鼠标控制机器人云台效果较差，因此为了提高发射命中率和改善操作手的操作手感，需要实现自动瞄准的功能。

使用 NVIDIA Jetson TX2 计算平台，配合高帧率工业相机，识别目标装甲板，并通过

SolvePNP 算法解算出目标在相机坐标系下的位置。将目标位置通过串口发送给主控，主控融合陀螺仪的数据，进行坐标变化，将相机坐标系下的目标位置映射到云台坐标系下。接着通过进行卡尔曼滤波、预测，获得目标的位置和估计速度，根据目标速度和机器人发射延迟、弹速等，计算需要的瞄准提前量，从而实现带有提前量的自动瞄准。

## 2. 重力补偿

步兵发射 17mm 荧光弹丸，由于空气阻力及其他因素影响，弹道不能由弹速、云台仰角、重力加速度计算得出。可以参考到的方案有忽略垂直方向空气阻力、收敛法、数学建模等方式，这些方法大都仅在某些距离上表现好，在步兵机器人可击打区域整体表现较差。我们提取了特定角度步兵机器人弹道曲线，直接拟合后发现步兵机器人弹道符合二次曲线簇的特征，所以决定采用直接拟合法。

优点：

- 1、场上计算量小，延迟低，大部分计算拟合台下完成
- 2、准确率高，通用性强
- 3、方法简单，不需要借助复杂设备或软件

### 1.4.3.4 软件测试

#### 1. 自动瞄准和卡尔曼滤波、预测



如图所示，绿色线为 PnP 解算得到的位置曲线，红色线为卡尔曼滤波后位置曲线，可见卡尔曼滤波器有较好的数据平滑效果。而通过设置较大的过程噪声方差和较小的观测噪声方

差，卡尔曼滤波器可得到收敛较快的目标速度最优估计值，对机动目标有较为良好的跟踪效果。

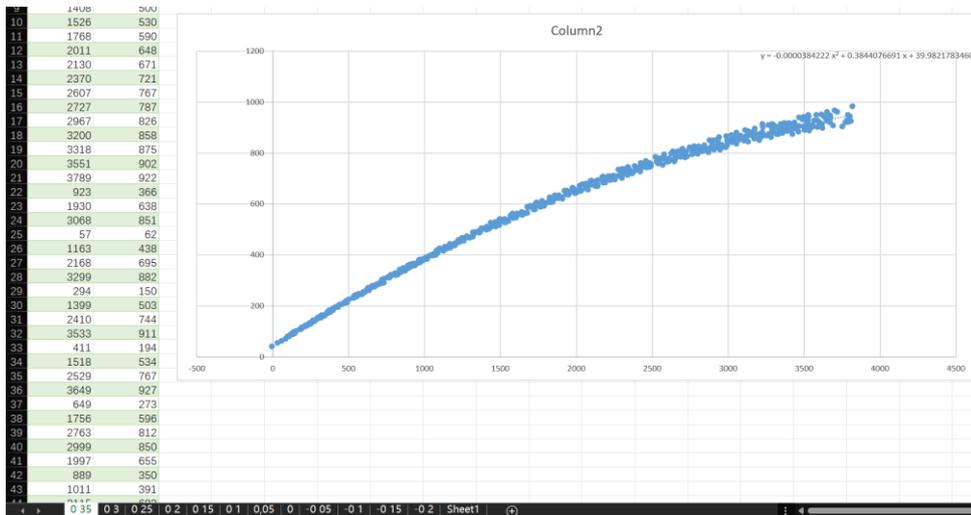
## 2. 重力补偿

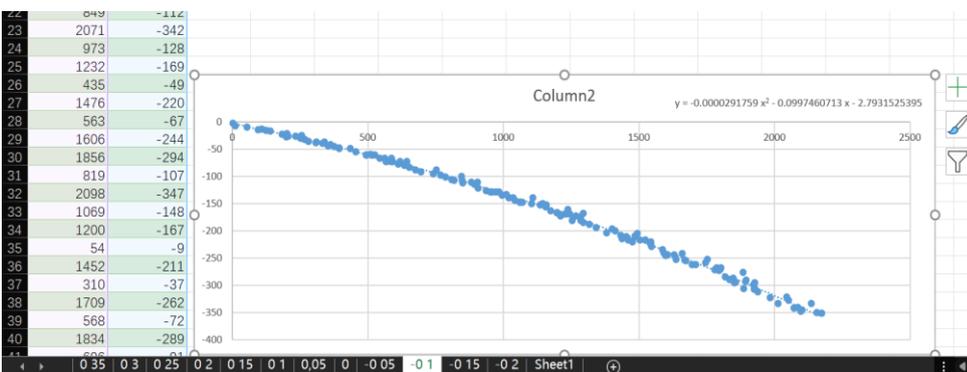
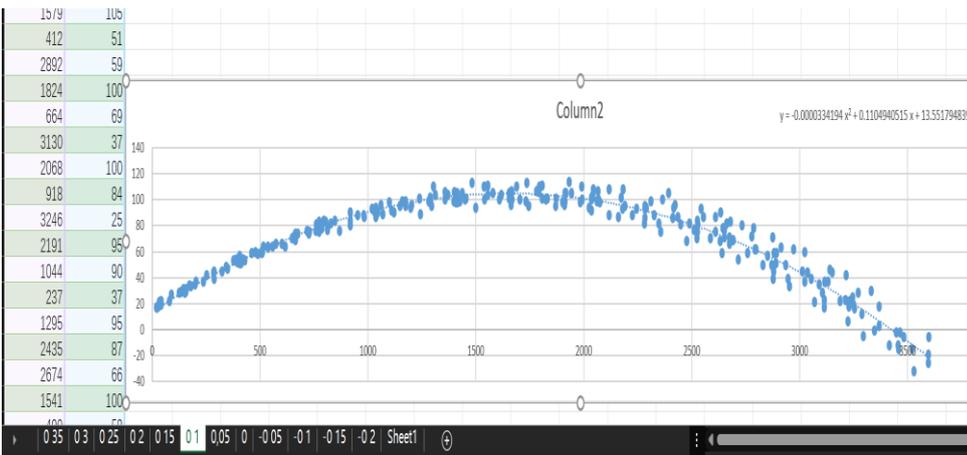
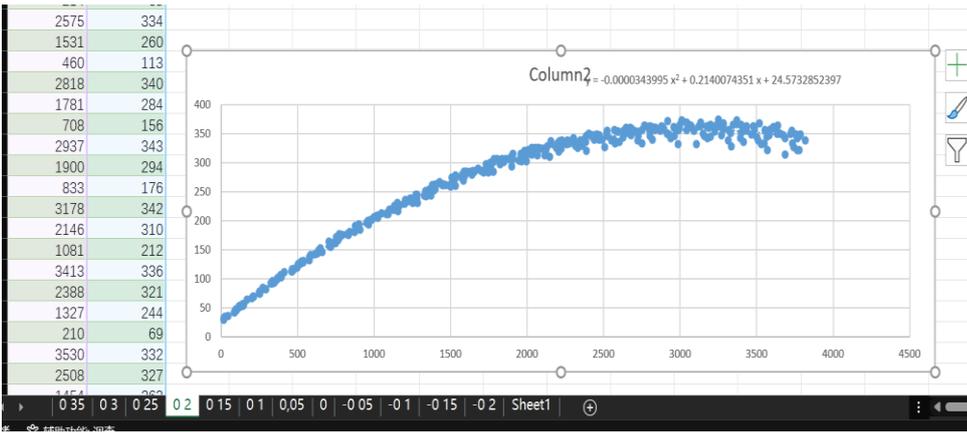
(1)首先采集弹道信息。使用手机或者其它设备拍摄弹丸发射视频，注意尽量正对弹道，消除畸变（略有误差影响也可后期调参消除，尽量准确）。



借助 opencv 鼠标交互，标出每一个弹丸位置，将 pitch 转轴当作原点，同时计算出比例尺。

(2)接着将弹道数据导入 excel 进行弹道拟合，将不同角度的弹道分别添加趋势线，使用二次曲线进行拟合，统计得出弹道和 pitch 轴角度关系。





对二次曲线的各项系数进行拟合，得到公式：

$$y = (a1*p+a2) x^2 + (b1*p+b2) x + (c1*p+c2)$$

整理后将敌方装甲板位置信息带入，计算出云台 pitch 角度。

经过重力补偿后，步兵机器人分别在 1m,3m,5m,8m 均可以准确命中装甲板中心。

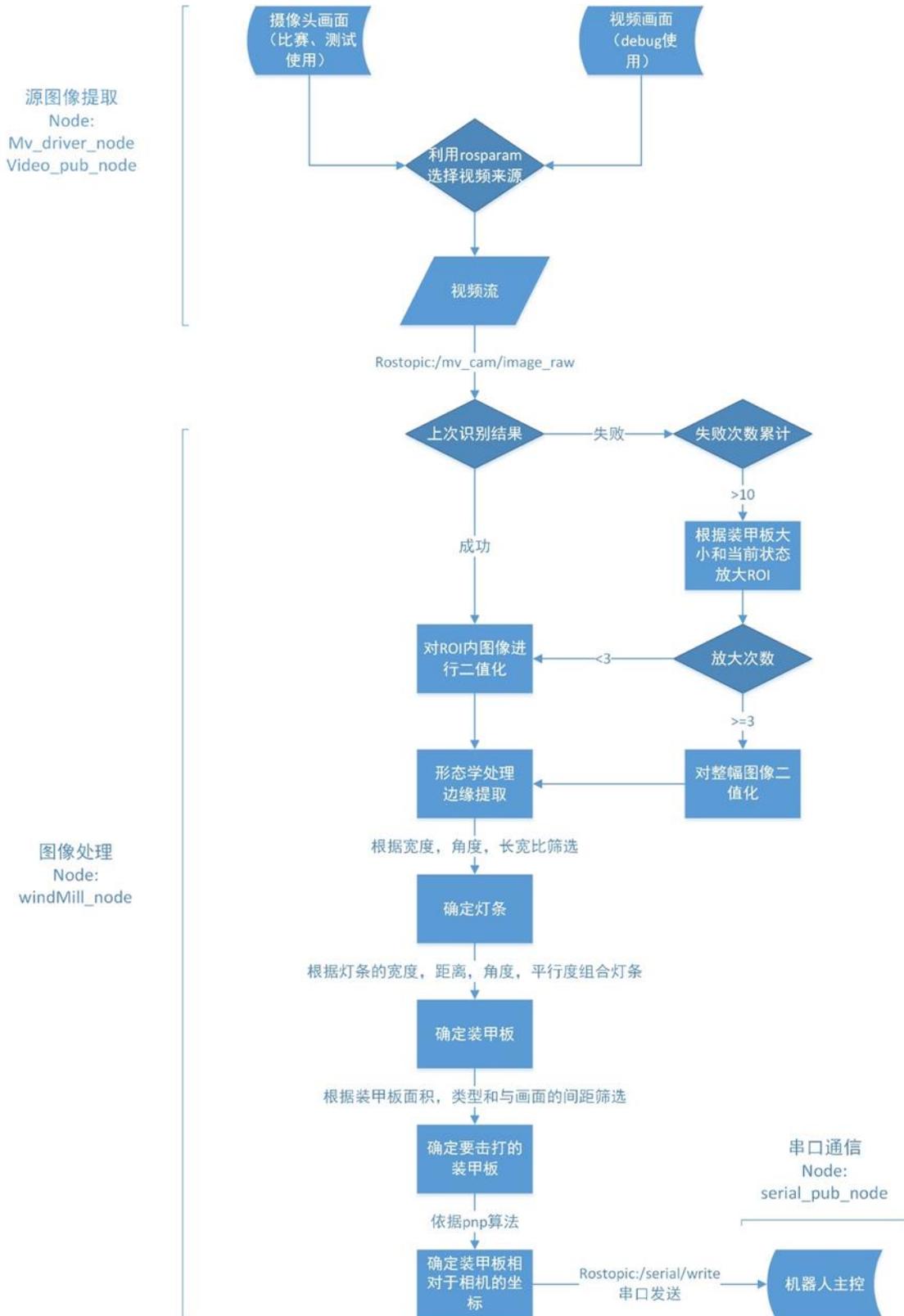
## 1.4.4 算法设计

### 1.4.4.1 功能简介

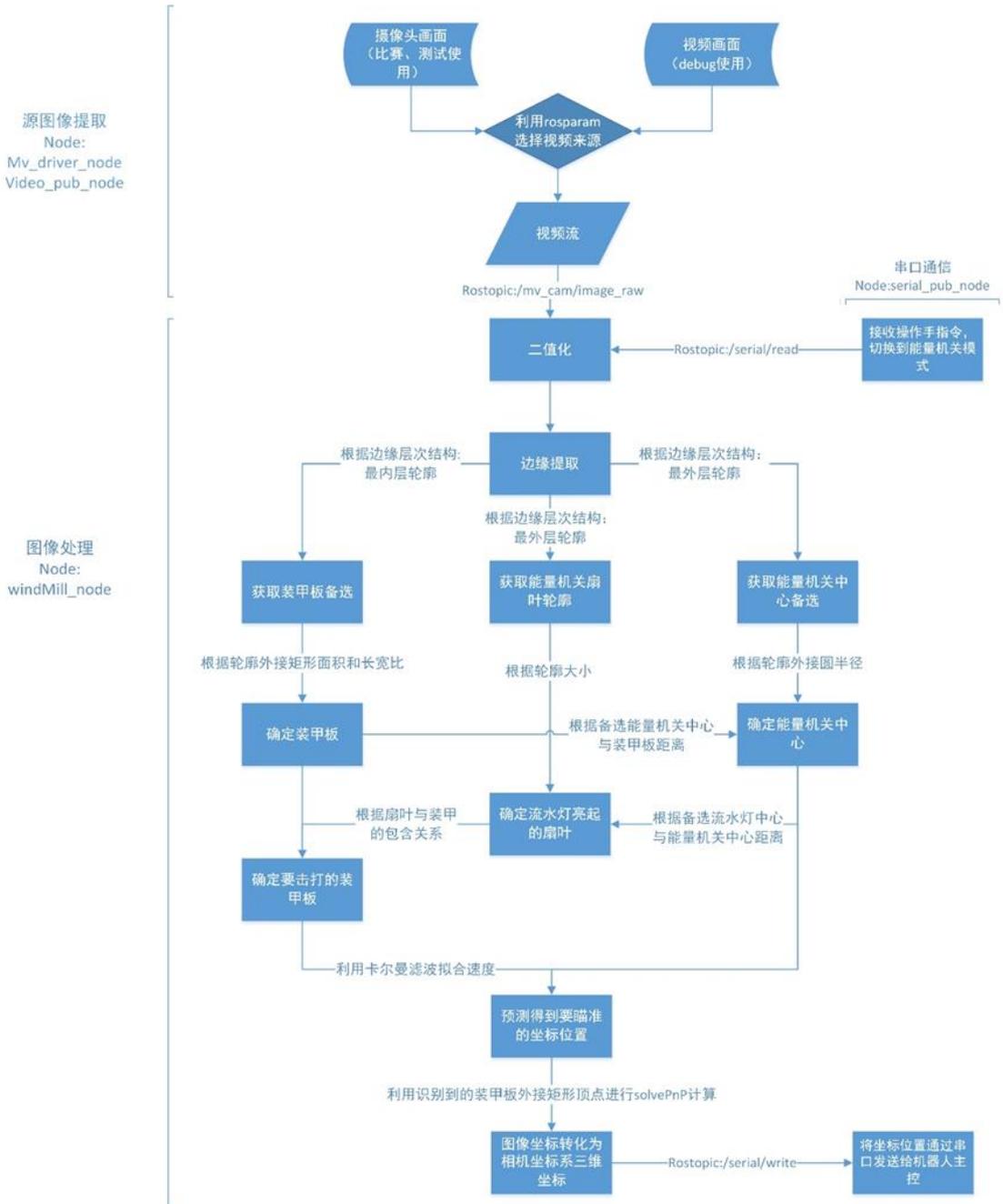
自瞄模式下可以实现对装甲板的自动识别，计算装甲板在相机坐标系下的三维坐标。能

量机关激活模式下可以实现对能量机关的识别和预测。

自瞄模式程序运行流程图如下：



能量机关激活模式程序运行流程图如下：



### 1.4.4.2 重要算法原理阐述

#### 自瞄算法原理阐释：

##### 1. 灯条检测

通过 OpenCV 的 split 函数分离图片的三色彩通道，当敌方为红色时，由红色通道减去蓝色通道，若为敌方为蓝色则反之。对得到的灰度图进行二值化处理，对二值化图像进行形态学处理以消除噪点，再通过 OpenCV 的 findContours 函数寻找轮廓，对找出的轮廓进行 PCA 主

成分分析，判断其长度和角度，即可筛选出目标的灯条。

## 2. 匹配灯条检测装甲板

遍历上一步检测出的灯条，进行双重循环，一一匹配，计算出两灯条的长度差值比例，角度差值，中心点间距，倾斜角，匹配符合条件的灯条为装甲板。

## 3. 挑选目标装甲板

在识别到多块装甲板时，计算每一块装甲板的权重决定打击目标。权重由装甲板长宽比例（即倾斜程度）、装甲板到相机中心点的距离、以及面积大小共同决定。

## 4. 对相机进行姿态解算

使用 OpenCV 的 solvePnP 函数对相机进行姿态解算。

在 solvePnP 需要的参数中：

**objectPoints** :特征点的世界坐标。由于求解的实际上是装甲板在相机坐标系的三维坐标，故以装甲板中心点为原点建立世界坐标系，选取装甲板灯条的四个顶点坐标作为特征点。

**imagePoints** : 特征点在图像中的像素坐标，即为识别到的装甲板灯条的顶点。灯条顶点由灯条轮廓经过 PCA 后得到的朝向和长度求得

**rvec** : 输出的旋转向量。

**tvec** : 输出的平移向量，即为得到的装甲中心在相机坐标系下的三维坐标。

**useExtrinsicGuess** : 用于 solvePnP 迭代的参数。由于我们无法提供初始近似的 rvec 和 tvec 矩阵，故此处设置为 False。

**flags** : PnP 的计算方法。我们在此选用 SOLVEPNP\_ITERATIVE，经过测试此方法可以提供最高的精度。

考虑某个空间点  $P$ ，它的齐次坐标为  $P = [X Y Z 1]^T$ 。在图像 I1 中，投影到特征点  $x_1 = (u_1 + u_2 + u_3)$ （以归一化平面齐次坐标表示）。此时相机的位姿  $R, t$  是未知的。与单应矩阵的求解类似，我们定义增广矩阵  $[R|t]$  为一个  $3 \times 4$  的矩阵，包含了旋转与平移信息。我们把它展开形式列写如下：

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

用最后一行把 $s$ 消去,整理得

$$X * (f_x * R_{11} + u_0 * R_{31} - x * R_{31}) + Y * (f_x * R_{12} + u_0 * R_{32} - x * R_{32}) + Z * (f_x * R_{13} + u_0 * R_{33} - x * R_{33}) = T_3 * x - f_x * T_1 - u_0 * T_3$$

$$X * (f_y * R_{21} + v_0 * R_{31} - y * R_{31}) + Y * (f_y * R_{22} + v_0 * R_{32} - y * R_{32}) + Z * (f_y * R_{23} + v_0 * R_{33} - y * R_{33}) = T_3 * y - f_y * T_2 - v_0 * T_3$$

我们可以看出 $f_x$   $f_y$   $u_0$   $v_0$ 是相机内参,  $XYZ$  是 3D 点的坐标,  $xy$ 是 2D 点的坐标, 共有 12 个未知数, 但是由于旋转矩阵  $R$  是正交矩阵, 每行每列都是单位向量且两两正交, 所以  $R$  自由度为 3, 秩也是 3, 加上平移向量的三个未知数, 一共六个未知数, 而每一组 2D/3D 点可以提供两个方程, 所以有三组 2D/3D 点即可确定六个方程, 解出六个未知数, 所以 PnP 至少需要三组点。有三组点时方程会得出四组解, 其中一组解为所需真解, 将第四个点的世界坐标代入公式, 获得其在图像中的四个投影 (一个解对应一个投影), 取出其中投影误差最小的那个解, 则为正解。

## 能量机关激活算法原理阐释:

识别:

1. 图像处理节点接受来自串口通信节点的消息进行模式切换, 在切换到能量机关激活模式时进行如下操作
2. 对图片进行二值化和边缘提取操作。通过 OpenCV 的 split 函数分离图片的三色彩通道, 当敌方为红色时,由红色通道减去蓝色通道, 若为敌方为蓝色则反之。获得二值化后的图像进行边缘提取操作并获得轮廓层级。
3. 通过轮廓层次关系获取装甲板, 能量机关中心的备选位置。选择最内层, 包围框大小, 长宽比满足条件的轮廓作为备选装甲板, 选择最内层, 包围圆半径满足条件的轮廓作为备选能量机关中心
4. 借助装甲板和能量机关中心的相对位置关系确定能量机关中心的真实位置。
5. 借助叶片的最外层轮廓确定当前需要击打的叶片, 通过位置关系确定需要击打的

装甲板。

6. 利用卡尔曼滤波器进行预测。利用 Kalman 滤波器获得准确估计需要系统模型，本系统采用匀速（Constant Velocity, CV）模型。CV 模型假设目标做匀速直线运动，即目标加速度的理想值为 0，但由于干扰的存在，加速度不能维持在 0 值，而是零均值高斯白噪声  $w_c(t)$ ，即  $w_c(t) \sim N(0, \sigma^2)$ 。

系统状态方程为

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\dot{x}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_c(t)$$

式中，状态转移矩阵  $F$  和噪声转移矩阵  $G$  分别为

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

对连续系统进行离散化。使用拉普拉斯变换的方法求得  $(sI - F)^{-1}$  为

$$(sI - F)^{-1} = \begin{bmatrix} s & -1 \\ 0 & s \end{bmatrix}^{-1} = \frac{1}{s^2} \begin{bmatrix} s & 1 \\ 0 & s \end{bmatrix} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} \\ 0 & \frac{1}{s} \end{bmatrix}$$

对上式进行拉普拉斯反变换，并取  $t = T_0$  得

$$A = \begin{bmatrix} 1 & T_0 \\ 0 & 1 \end{bmatrix}$$

对于离散模型过程噪声  $w(k)$  的协方差矩阵，使用基于连续系统的离散化模型（Discrete Model Based on Continuous System, DCM）求解。

$$\begin{aligned} Q &= E[w(k)w^T(u)] = \int_0^{T_0} e^{F(T_0-\lambda)} G \cdot \sigma^2 \cdot G^T \left[ e^{F(T_0-\lambda)} \right]^T d\lambda \\ &= \int_0^{T_0} \begin{bmatrix} 1 & T_0 - \lambda \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \sigma^2 \cdot \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ T_0 - \lambda & 1 \end{bmatrix} d\lambda \\ &= \int_0^{T_0} \begin{bmatrix} (T_0 - \lambda)^2 & T_0 - \lambda \\ T_0 - \lambda & 1 \end{bmatrix} \sigma^2 d\lambda \\ &= \begin{bmatrix} \frac{1}{3} T_0^3 & \frac{1}{2} T_0^2 \\ \frac{1}{2} T_0^2 & T_0 \end{bmatrix} \sigma^2 \end{aligned}$$

系统观测方程为

$$z(k) = Hx(k) + v(k)$$

综上，离散系统状态方程为

$$\begin{cases} x(k+1) = Ax(k) + w(k) \\ z(k) = Hx(k) + v(k) \end{cases}$$

式中， $E[w(k)] = 0$ ， $w(k)$  协方差矩阵为  $Q$ ， $v(k) \sim (0, r_k)$ 。

能量机关运动是圆周运动，所以观测量和预测量分别为角度和角速度。状态变量为

$$x = \begin{bmatrix} \theta \\ \omega \end{bmatrix}$$

状态转移矩阵  $A$  及观测矩阵  $H$  分别为

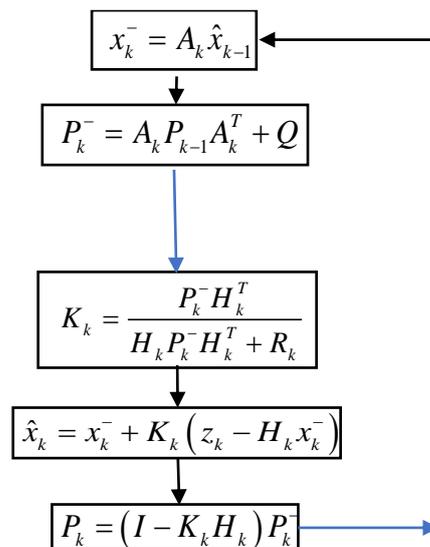
$$A = \begin{bmatrix} 1 & T_0 \\ 0 & 1 \end{bmatrix} \quad H = [1 \quad 0]$$

过程噪声矩阵  $Q$  为：

$$Q = \begin{bmatrix} \frac{1}{3}T_0^3\sigma^2 & \frac{1}{2}T_0^2\sigma^2 \\ \frac{1}{2}T_0^2\sigma^2 & T_0^1\sigma^2 \end{bmatrix}$$

$\sigma$  为过程噪声方差

完整流程为：



## 7. 对卡尔曼进行优化

由于在能量机关的击打过程中，会有从一个扇叶切换到另一个扇叶的情况，角度值会突变，但由于能量机关运动过程连续，角速度不会突变，切换装甲模块时，只更新角度，保留其它参数，卡尔曼收敛加快，同时可能做到在短时间内连续击打多个扇叶。

## 8. 相对位置求解

模拟出预测装甲模块四个角点，通过 solvePnP 解出在相机坐标系下装甲模块位置，得到装甲模块的平移矩阵

$$r_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

相机固连于云台枪管下方，故目标在云台坐标系( $b$ 系)的坐标可由  $r_c$  经过平移后得到：

$$r_b = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = r_c + M_c^b$$

式中， $M_c^b$  为平移向量。

利用 RoboMaster 开发板 C 板载 BMI088 传感器数据得到四元数构成的坐标变换矩阵  $C_b^n$ ，将目标坐标由云台系( $b$ 系)转换至惯性系( $n$ 系)：

$$r_n = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = C_b^n r_b$$

式中：

$$C_b^n = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix}$$

再根据整理补偿计算 pitch 角度，可以 5 发弹丸成功激活大能量机关，

同时，将击打视频上传至 b 站后，获得了 110 万以上的播放量。

### 1.4.4.3 性能分析

目前的算法系统在 Jetson TX2 上运行，在 1280\*512 的分辨率下可以达到 170FPS，640\*480 分辨率下可以达到 360FPS。在测试环境下（无场地装饰灯条）误识别率极低。

优点：

1. 算法系统整体采用传统算法获得了相对较快的运行速度，
2. 传统算法识别四点较为准确，保证后续的 PnP 计算得到的坐标精度较高，
3. 较高的帧率对电控友好。

缺点：

1. 采用传统算法导致抗干扰能力相对较差，可能受场地灯效的影响。
2. 无法在受遮挡的情况下识别灯条。

优化：

1. 向深度学习迁移。计划采用 yolov5s，利用开源数据集训练神经网络作为备用方案。

同时对现有代码进行改进，将目前的传统算法进一步模块化，目标是完成一套可以在上场前随时切换传统算法与深度学习的视觉系统。

2. 采取共享内存以减少数据的重复读写。该算法系统的主要时间开支来源于对内存的反复读写。算法中有许多操作需要对图像进行遍历，另外将相机和识别分为两个进程的操作也增加了数据复制（内存开辟销毁）的开销。改动的思路之一是采取共享内存以减少数据的重复读写，目前这方面的代码已经编写完成并在 GitHub 开源，正在进行测试；

3. 更换处理器。传统视觉算法并不使用 GPU，主要时间开支来源于对内存的反复读写，提升运算平台主频可以有效提升算法运行速度。TX2 的 CPU 最高主频仅有 2GHz，相比于目前主流 CPU 已经有较大劣势，更换为搭载笔记本 CPU 的 NUC 可以有效提升算法运行速度，

在 1280\*512 的分辨率下可以达到 240FPS（受限于相机帧率），但是 NUC 体积较大，尤其是在步兵上部署的时候需要机械做出较大调整。

#### 1.4.4.4 算法库介绍与接口说明

目前视觉系统的架构基于 ROS，在不同 ROS 包中实现不同功能，整个程序松耦合便于调试和修改。

MVCamera 节点：

/MVCamera/image\_raw 发送原始图像

Detect 节点：

/MVCamera/image\_raw 接收原始图像

/serial/write 发送识别结果

/serial/read 接收模式切换信息

SerialCommon 节点：

/serial/write 接收识别结果

/serial/read 发送模式切换信息

Detect 包中完成了主要的识别工作。其中具体的函数说明利用 Doxygen 生成了文档，部分函数说明如图：

**• DetectLEDMarker()**

```
int MarkSensor::DetectLEDMarker ( Mat & img,
                                Marker & res_marker
                                )
```

**MarkSensor::DetectLEDMarker** 识别装甲板 (对相机得到的全图做处理)

**Parameters**

**img** 源图像 (彩色)  
**res\_marker** 要击打的装甲板

**Returns**

int 0成功

**• DetectTopCenter()**

```
int MarkSensor::DetectTopCenter ( Mat & img,
                                 Rect & ROI,
                                 Point2f & res_point,
                                 Marker & top_marker
                                 )
```

**MarkSensor::DetectTopCenter** 反陀螺模式, 识别陀螺

**Parameters**

**img** 源图像 (彩色)  
**ROI** 确定的车辆ROI, 用于之后的追踪  
**res\_point** 车辆中心坐标  
**top\_marker** 识别到的装甲板

**Returns**

int -1失败 0成功

**• getRotatePoint()**

```
Point2f Dafu_Detector::getRotatePoint ( const cv::Mat & srcImage,
                                         cv::Point Points,
                                         const cv::Point rotate_center,
                                         const double angle
                                         )
```

**Dafu\_Detector::getRotatePoint** 将坐标围绕圆心旋转一定角度, 用于补偿弹丸在空中飞行的时间

**Parameters**

**srcImage** 源图像  
**Points** 待旋转的点  
**rotate\_center** 旋转中心 (大符中心)  
**angle** 旋转的角度值, 是经验值

**Returns**

Point2f 旋转后的坐标

**• KFPredict()**

```
float Dafu_Detector::KFPredict ( float angle_arc)
```

**Dafu\_Detector::KFPredict** 利用卡尔曼滤波器预测大符转速

**Parameters**

**angle\_arc** 观测值, 角度

**Returns**

float 转速

更加详细的内容可以参看:

<https://nuaa-rm.github.io/classMarkSensor.html> 和

[https://nuaa-rm.github.io/classDafu\\_\\_Detector.html](https://nuaa-rm.github.io/classDafu__Detector.html)

### 1.4.4.5 结果展示

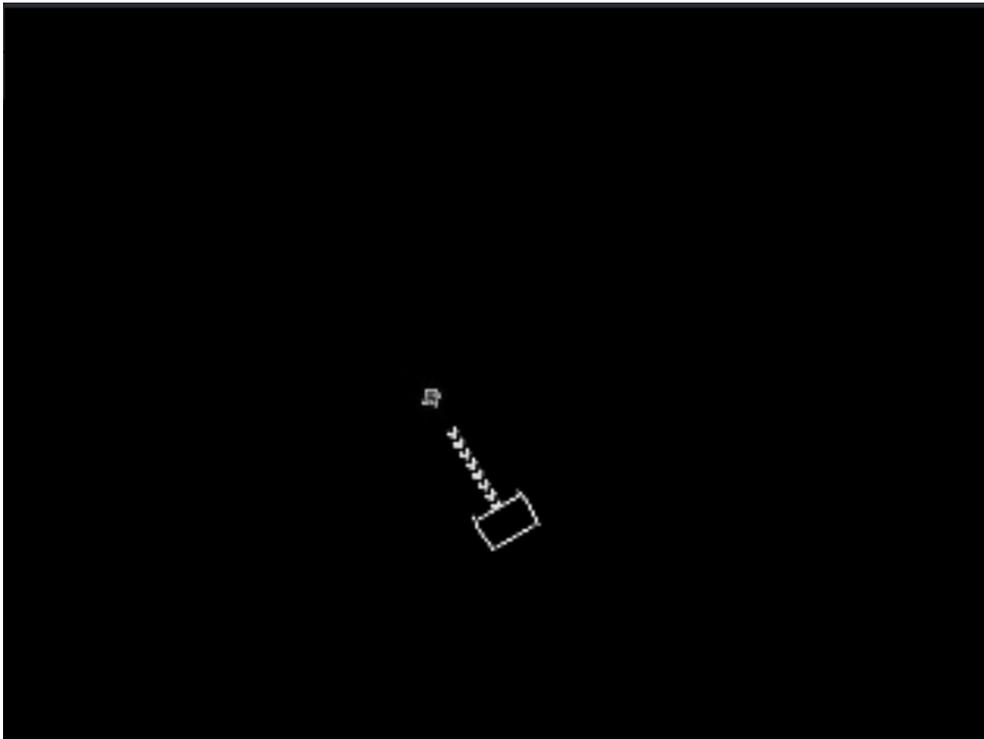
自瞄识别效果



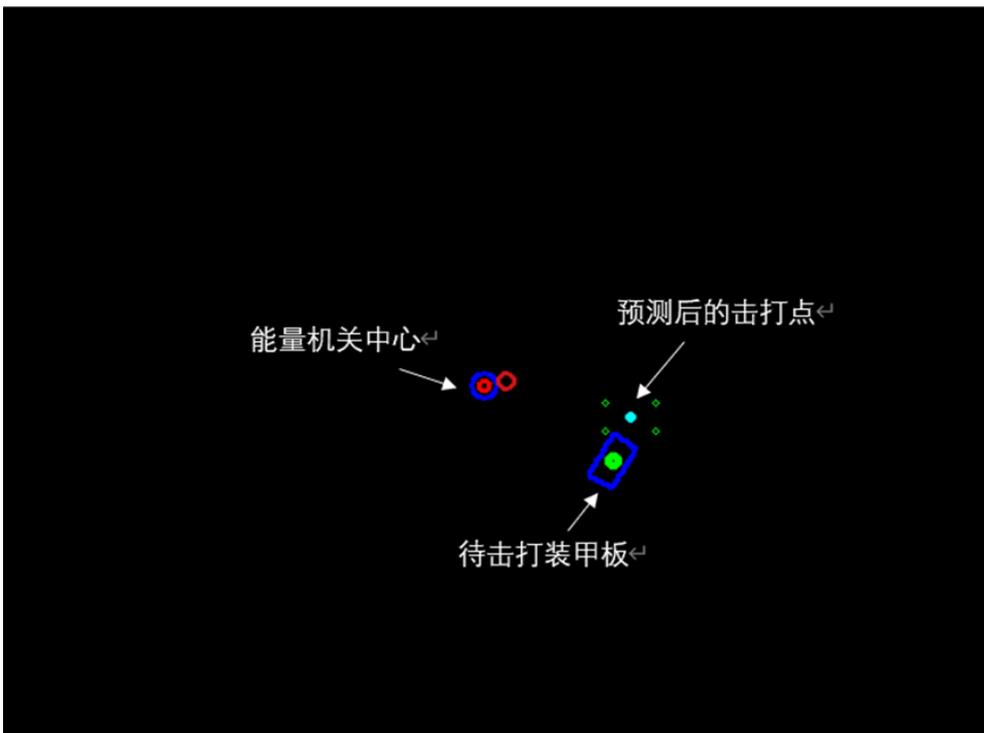
自瞄 ROI 效果与二值化效果



能量机关二值化效果



能量机关识别效果



## 1.5 研发迭代过程（10）

### 1.5.1 版本迭代过程记录

#### 1.5.1.1 机械迭代记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	实现 yaw 轴 360° 旋转小陀螺机构和螺杆供弹机构。	2022.1.5
V1.1	针对底盘优化实现飞坡，针对螺杆供弹优化采用单线供弹方案解决末端卡弹问题。	2022.3.20
V1.2	螺杆供弹采用下端线路，解决空弹问题。	2022.4.15

#### 1.5.1.2 硬件迭代记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	同步整流 BUCK 拓扑，实现充放电基本功能	2021.11.20
V1.0	改良 PCB 设计，减少振铃，较小高频纹波	2021.12.25
V1.1	改用四开关管拓扑，解决电容反向供电的问题	2022.3.1
V1.2	加 CAN 通信芯片，可实现与主控的 CAN 通信	2022.4.15

#### 1.5.1.3 视觉迭代记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	可以实现对装甲板的精确识别，帧率可达 360FPS	2021.10.1

版本号或阶段	功能或性能详细说明	完成时间
V1.1	可以实现运动预测与重力补偿，调整分辨率，扩大了视野。帧率在 1280*512 分辨率下可达 170FPS	2022.3.20
V1.2	可以稳定激活大能量机关	2022.4.10

### 1.5.1.4 电控迭代记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	完成对步兵的底盘、云台控制，以及普通自瞄(无预测和重力补偿)可以使用键鼠控制。	2022.1.10
V1.1	完成自瞄的优化，添加卡尔曼滤波和预测，添加重力补偿。	2022.3.15
V1.2	完成击打大能量机关。	2022.4.10

### 1.5.2 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 本号或阶 段	解决人员
1	步兵进行自瞄时，调试参数无法符合要求，收敛慢或导致震荡。	从相机坐标系到云台坐标系的坐标变化存在问题，导致坐标变化错误。	重新确定云台坐标系和相机坐标系定义，修改错误的变换矩阵。	V1.1	嵌入式软件工程师：王健、杨洛、叶梁杰 算法工程师：李霖杰

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
2	超级电容采用同步整流 BUCK 电路, 实验中发现反复切换充放电状态后, 系统不稳定, 无法再次充电或放电电压升高反灌电池。	程序结构有问题, 控制程序写在主程序, 导致控制周期不确定, 并且加入一些其他程序或者中断较多时, 控制周期变得很长, 导致系统不稳定。	将控制程序写在定时器中断, 并且设置最高优先级, 高于通信等的优先级。	V1.1	硬件工程师: 李高峰
3	超级电容输入输出纹波较大。	MOS 驱动的振铃较大, PCB 布局并没有充分考虑 EMI、散热等问题。	重新设计 PCB, 改善振铃。	V1.1	硬件工程师: 李高峰

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
4	螺杆供弹末端卡弹。	在输出末端螺扇对弹丸的作用力不足以使弹丸随着旋转,截面处五个弹丸全部在做自旋运动的基础上向前平动,因此造成了末端卡弹。	放弃弹丸全部进入供弹的方案,采用单线方案,仅仅使用螺杆使弹丸进行向前的平动,只使用到了螺杆大约 1/5 的部分,每次可以同时运输 5 颗 17mm 小弹丸,在螺杆运动时只与其上的 5 颗弹丸作用。	V1.1	机械工程师:刘金星

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
5	飞坡磕保护框造成整体前翻。	保护框高度太低,保护框前缘与麦克纳姆轮的最短距离过小,使得保护框比麦克纳姆轮更先接触地面。	针对飞坡进行定速测试,结果显示最大触地角可达 $43^{\circ}$ 左右,因此将保护框高度定为麦克纳姆轮直径75%处附近较为合适,并控制保护框与麦克纳姆轮的径向间隙在5mm左右。	V1.1	机械工程师:颜治毅、李佳、杜毓桐

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
6	步兵在 12m/s 或 15m/s 射速下对移动目标的命中率低。	射速低导致弹丸飞行时间无法被忽略,在射击移动目标时有滞后。 同时弹丸飞行时间长导致重力带来的下坠无法被忽视。	引入卡尔曼滤波。利用 CV 模型,将目标在绝对坐标系下的三维坐标作为观测量,得到目标的移动速度,将其叠加至电机控制量上,实现对目标的预测。测定弹丸在不同速度和仰角下的弹道轨迹并将其拟合,实际射击时计算得到pitch轴需要补偿的角度,实现重力补偿。	V1.1	嵌入式软件工程师: 王健 算法工程师: 李晨

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
7	超级电容一开始采用同步整流 BUCK 电路，电源下电之后，电容如残留较多能量，则会反向供电。	反向等效为 BOOST 电路，输入电流通过二极管直接流向输出。	改用四开关管的 BUCK-BOOST 电路拓扑，下电之后输入输出直接隔离。	V1.2	硬件工程师：李高峰
8	螺杆供弹有空弹现象，输出不连续，同时给予操作手极差的操作体验。	由于使用的全没入的结构，使得在供弹时不会于其它弹丸相接触，在弹丸较多时大约 200 发以上会形成一个自锁饿空腔，使得弹丸无法下落造成空弹的问题。	将弹路放到螺杆下部，在两侧留出 18mm 的进弹口，在使用下部供弹的同时使用其它部分进行弹丸的搅动，避免空弹。	V1.2	机械工程师：刘金星

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
9	步兵利用卡尔曼滤波预测大符运动时，亮起叶片切换会导致卡尔曼滤波器预测出的速度不准。	大符亮起叶片切换时会导致观测量出现较大跳变，使得卡尔曼滤波器需要再次收敛。	由于大符在亮起叶片切换时旋转速度不变，选择角度作为观测量，角度和角速度作为状态变量，在叶片切换时手动重置保存的状态变量中的角度并保持当前角速度，使其不需要重复计算。	V1.2	算法工程师：李霖杰

## 1.6 团队成员贡献（2.5）

姓名	基本信息		贡献度
	主要负责工作内容描述 (专业、年级、队内角色)		(所有成员贡献度合计为 100%)
颜治毅	飞行器动力工程、大四、机械设计负责人	负责机器人的底盘设计、云台 yaw 轴设计	10%

姓名	基本信息		贡献度
	(专业、年级、队内角色)		(所有成员贡献度合计为 100%)
叶梁杰	自动化、大四、电控负责人	负责硬件调试, 电控程序编写以及总体规划	10%
李高峰	电气工程及其自动化、大四、电控队员	负责超级电容的研制及测试	9%
李霖杰	飞行器动力工程、大四、视觉负责人	负责自瞄算法的更新升级, 能量机关激活算法的开发	9%
王健	导航、制导与控制、研一、电控队员	负责构建卡尔曼滤波代码以及测试	9%
刘金星	机械工程、大二、机械队员	负责设计双枪管云台设计	9%
李晨	机械工程、大二、视觉队员	负责重力补偿的实现	9%
杨洛	机械工程、大二、电控队员	负责车辆调试以及功能验证; 负责调试卡尔曼滤波	9%
杜毓桐	航空航天类、大二、机械队员	负责优化底盘飞坡及其他机械结构优化	9%
李佳	航空航天类、大二、机械队员	负责优化底盘飞坡及其他机械结构优化	9%
温玮	航空航天类、大二、电控队员	负责车辆调试及功能验证	8%

## 1.7 参考文献 (2.5)

- 1、RoboMaster 开发板 C 例程: <https://github.com/RoboMaster/Development-Board-C-Examples>
- 2、华南理工大学 2021 舵轮步兵开源报告

- 3、东南大学 2021 双枪管机械结构开源报告
- 4、RM2021-华南农业大学 TAURUS 战队-舵轮单枪管步兵-机械结构开源
- 5、RM2021-大连理工大学-凌 BUG-步兵机械结构开源
- 6、OpenCV 官方文档：<https://docs.opencv.org>
- 7、上海交通大学 RoboMaster 2021 赛季视觉代码：<https://github.com/Harry-hhj/CVRM2021-sjtu>

## 2. 哨兵机器人

### 2.1 其它学校哨兵机器人分析综述（2.5）

从论坛上的各种开源文档来看，这个赛季哨兵机器人，相比于上个赛季变化不大。绝大部分哨兵机器人继续采取上下双云台，双弹舱的整体布局。上下双枪口哨兵能够在赛场上同时具有不错的远程攻击能力和对地面强劲的火力压制。

同时，由于最近几个赛季哨兵轨道一直是直线状态，哨兵处在一个易攻难守的位置，对于哨兵来说，快速机动与火力覆盖无疑是仅有的破局之道。

哨兵是 RM 赛场上造型最丰富的兵种之一，经对比赛观摩和论坛文献的查阅，我们对其他学校的哨兵功能有如下分析：

1. 快速机动：能够在轨道上实现快速移动和快速随机变向是哨兵躲避袭击的主要手段。哨兵在快速变向的时候，需要消耗极大的功率。有部分队伍因此设计了轨道刹车装置。
2. 自瞄射击：哨兵的自瞄有别于其他车的自瞄，哨兵的自瞄是一种完全的全自动状态，因此，其要求要比一般的其他车的自瞄要求要高很多，尤其是在自动识别，自动判断击发及避免误识别等方面。
- 3, 反小陀螺：小陀螺这一功能已是攻击型单位很基础的功能了，有一定积累的战队都已实现此功能甚至搭配超级电容获得更好的性能。给哨兵自动识别造成了很大的干扰，目前已经有部分队伍的哨兵初步具有反小陀螺的算法思路。

以上是本队伍截取的比较有共通性的哨兵概况，对于一些比较小众的以及对于我们队伍没有较大提升的内容，则没有收录。通过对于其他学校的了解，我们认识到了自己与强队的差距，也知道了自己的努力方向，目前我们正在逐步优化我们的哨兵，期待能够有一个好的效果。

### 2.2 机器人功能定义（5）

哨兵机器人的主要功能有两个：自动移动和自动射击

1. 自动移动：能够在轨道上快速机动且不会撞到两侧柱子。实现功率闭环，避免超功率。
2. 自动射击：该部分包括稳定识别，自动击发，精准弹道，同时通过车间通信并与雷达

站进行配合。

## 2.3 机器人核心参数（2.5）

### 2.3.1 机器人基础参数

参数类型	数值
尺寸 (mm,L*W*H)	420*503*837
重量 (kg)	12.9
重心高度(mm)	轨道上表面下方 40mm

### 2.3.2 主要传感器

型号	参数	数量
BMI088	BMI088 集成了 16 位三轴陀螺仪和 16 位三轴加速度计。具有低于 2/h 的偏置稳定性和低于 15mdps/K 的低温度系数偏移, 以及在 $\pm 24g$ 的测量范围内仅 230ug/ $\sqrt{Hz}$ 的低频谱噪声。	2
工业相机	MV-SUA134GC-T 工业相机与 MV-LD-4-4M-G 工业镜头, 在 1280*1024 的图像尺寸下帧率可以达到 211FPS。视场角为 $82.9^{\circ} \times 66.5^{\circ} \times 94^{\circ}$ 。采用全局快门, 拖影少。	2

### 2.3.3 执行器件

类型	用途	数量
RoboMaster M3508 减速直	用于底盘移动的动力输出。	1

流电机		
RoboMaster M3508 减速直流电机（去掉减速箱）	用于发射机构摩擦轮的动力输出。	4
RoboMaster GM6020	用于双云台 yaw 轴和 pitch 轴移动的动力输出。	4
RoboMaster M2006 P36 直流无刷减速电机	用于拨弹盘供弹机构的动力输出。	2

### 2.3.4 机器人性能参数

参数类型	数值
车体最大移动速度	4m/s
最快转向时间	0.2s
云台自由度	yaw 轴 360° 旋转；pitch 轴±90° 俯仰
Yaw 轴旋转速度	1.5r/s
最大射速	30m/s
命中率	5m 小装甲命中率≥98%
弹仓容量	≥600 发

## 2.4 设计方案（25）

### 2.4.1 机械结构设计

#### 2.4.1.1 云台设计

云台设计采用上下双云台相同的设计思路，以减低设计难度和制造成本，同时保证云台

之间的零部件互换性。云台设计采用双 6020 直连的设计，保证稳定性的同时，简化设计成本。

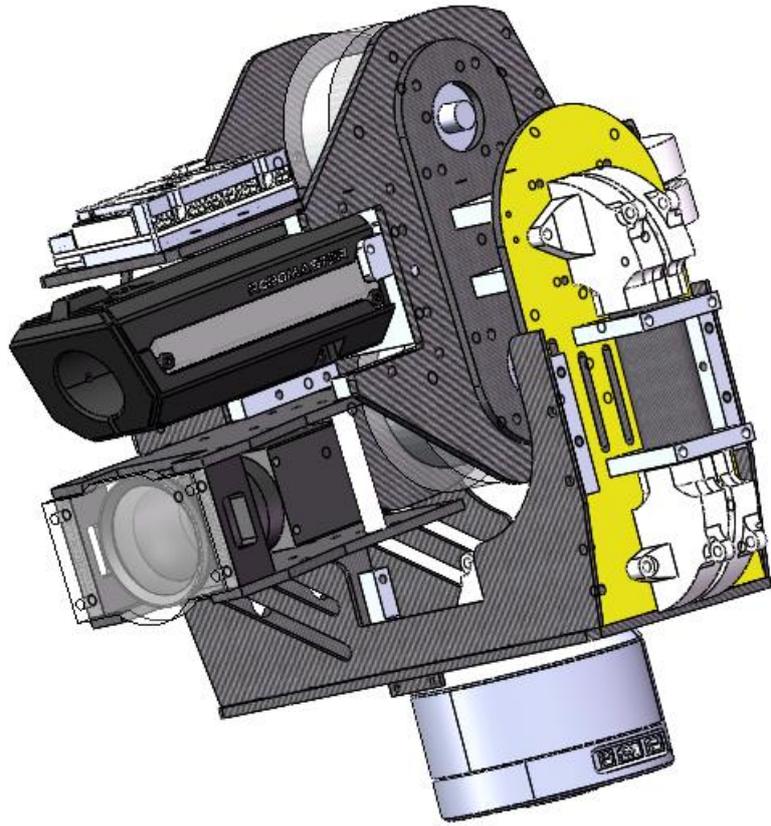


图 2.1 哨兵云台

### 2.4.1.2 发射机构设计

发射机构采用无枪管设计，动力部分使用去除减速箱的 3508 电机作为摩擦轮电机。在发射机构上采用波珠螺丝作单发限位。

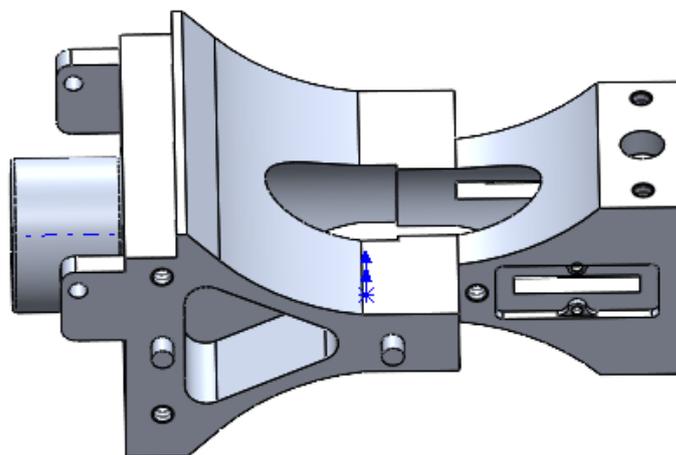


图 2.2 哨兵枪管

### 2.4.1.2 拨弹盘设计

为解决卡弹问题，本赛季经过多次试验，最终采用挡片加毛刷方案（如右上图红色区域内是毛刷区域），虽未在机械上完全解决卡弹问题，但是已经将卡弹的几率大大降低了。经过测试概率大概 1/2500。如若卡弹便在电控方面将拨弹盘反转消除卡弹。经实战证明，此方案的拨弹盘未在比赛期间卡过弹。

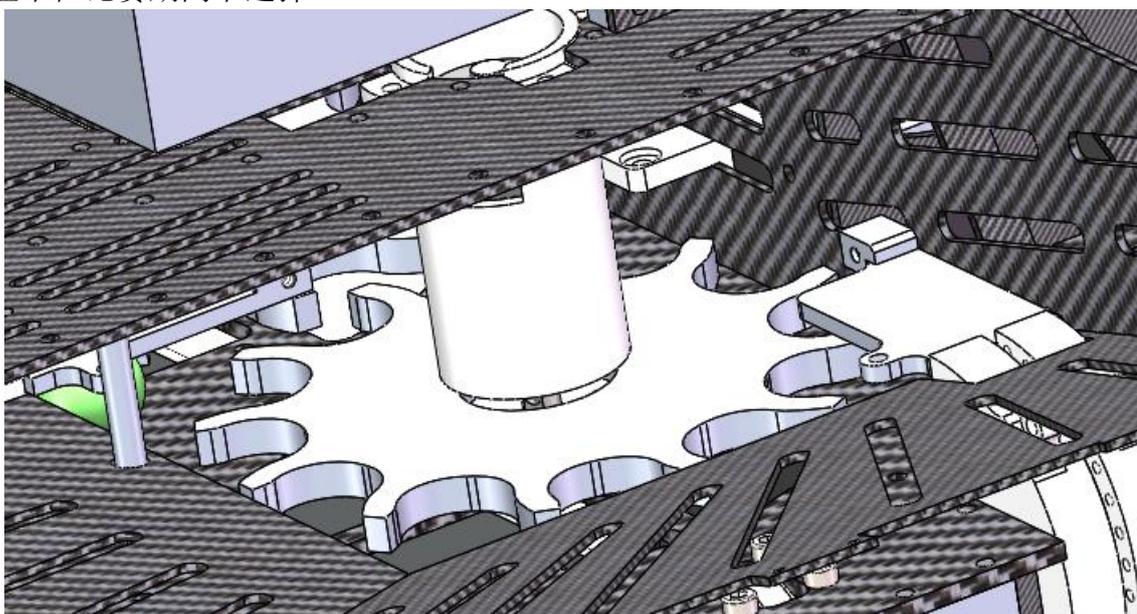


图 2.3 哨兵拨弹盘设计安装

### 2.4.1.3 运动部分设计

哨兵底盘设计采用三轮车设计形式，通过一个主动轮，搭配两个并排的从动轮，保证哨兵能够稳定在轨道上表面，同时依靠哨兵自重实现主动轮与轨道之间的预紧，保证更好的抓地力。主动轮方面，抛开传统的 CNC 包胶轮的设计，创造性的使用 1/10 车模所用的竞速轮胎加 CNC 轮毂，在大大减低轮子的成本的同时，极大的增加了胎皮的抓地力。

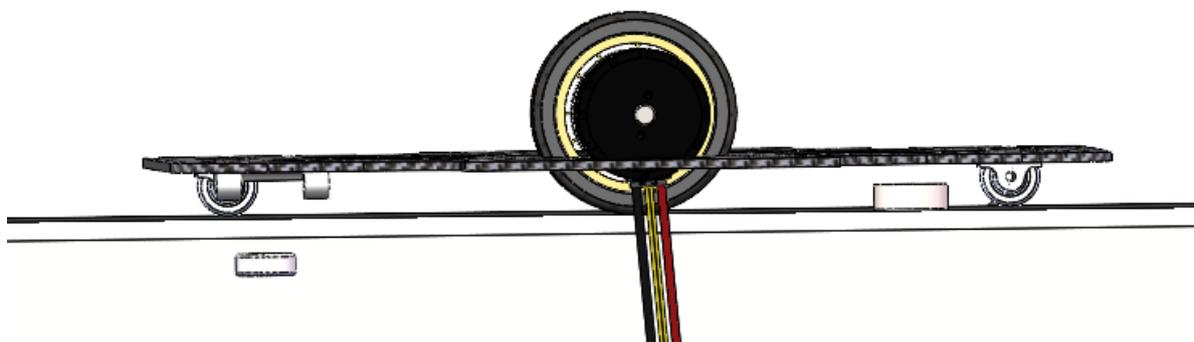


图 2.4 哨兵运动部件

### 2.4.1.4 整体承力框架

哨兵的整体承力框架围绕轨道挂载展开，轨道上方为一块安装底盘的碳纤维板附加一层用来安装上云台的二楼板，以此来增强轨道上放的部分的强度。下方为一块碳纤维板用来安装下云台。上下两层通过合页外加碳板连接起来。挂载后通过箱子用的搭扣进行锁止。

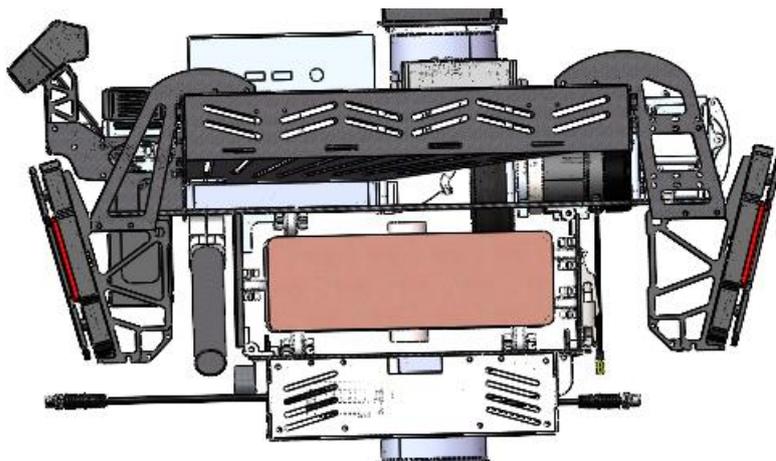


图 2.5 哨兵主承力结构

### 2.4.1.5 裁判系统安装

裁判系统安装根据 Robomaster2022 机甲大师高校系列赛机器人制作规范手册的规定，对

于装甲板，选择将其安装在上边沿距离轨道上表面 43mm 处，较高的装甲安装，能够使哨兵的结构设计更为精巧，同时保证下云台具有良好的俯仰角范围。

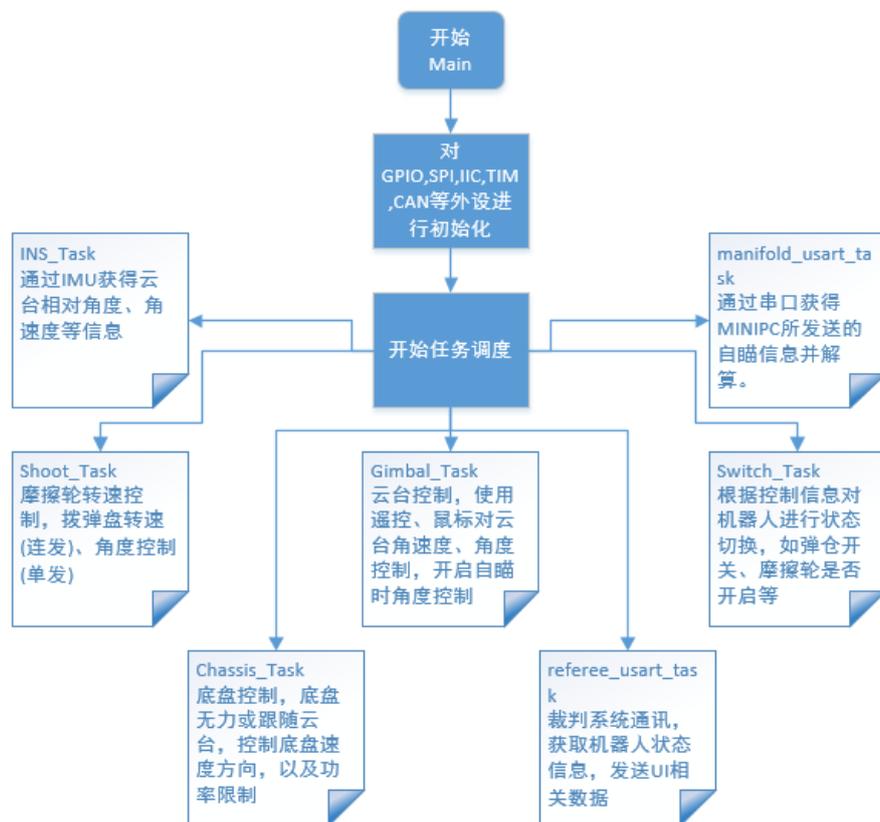
## 2.5 软件设计

### 2.5.1 系统架构

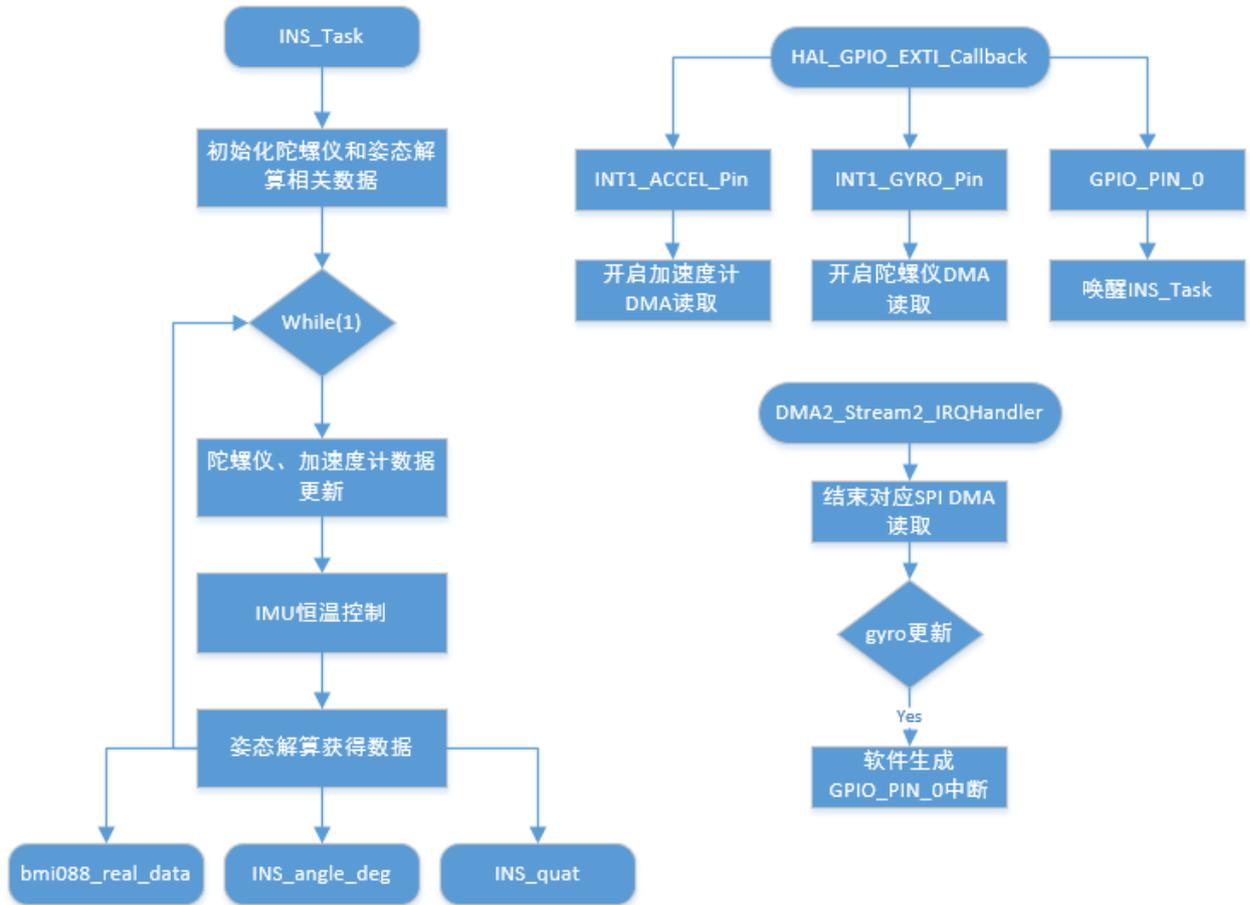
嵌入式软件部分，分为硬件抽象层 (HAL)、操作系统层 (RTOS) 和应用层。硬件抽象层使用 CubeMX 软件配置，STM32 HAL 库编写，参考 Robomaster 开发板 C 型开源程序，重新构建了较为简洁的源码。操作系统层使用 Freertos 进行任务调度，在不同的任务 (Task) 中进行机器人不同部分的控制。应用层主要包括底盘控制、云台及发射控制、传感器数据处理、自瞄计算等。开发环境为 STM32CubeMX+MDK ARM，使用 MDK 的调试功能和串口等观察数据，进行调试。

### 2.5.2 运行流程

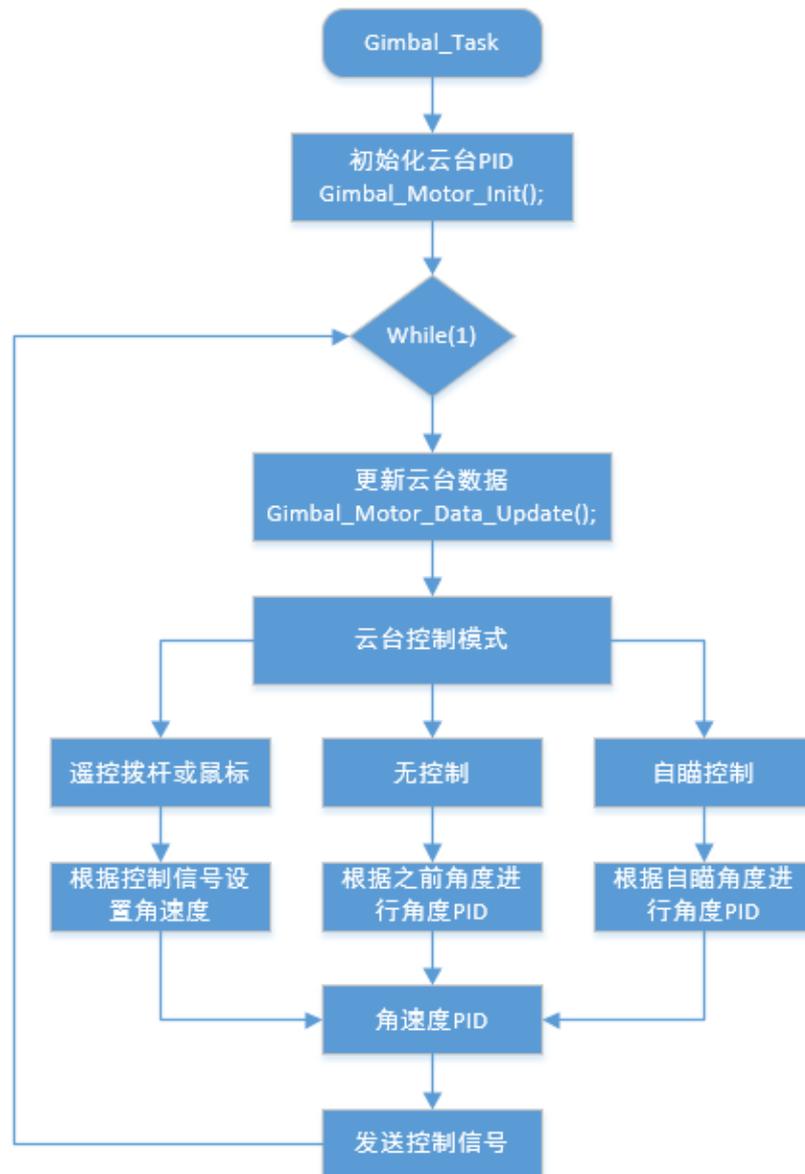
嵌入式软件部分，在主函数中进行外设初始化后，即开启 Freertos 任务调度，目前主要分为 7 个控制任务。



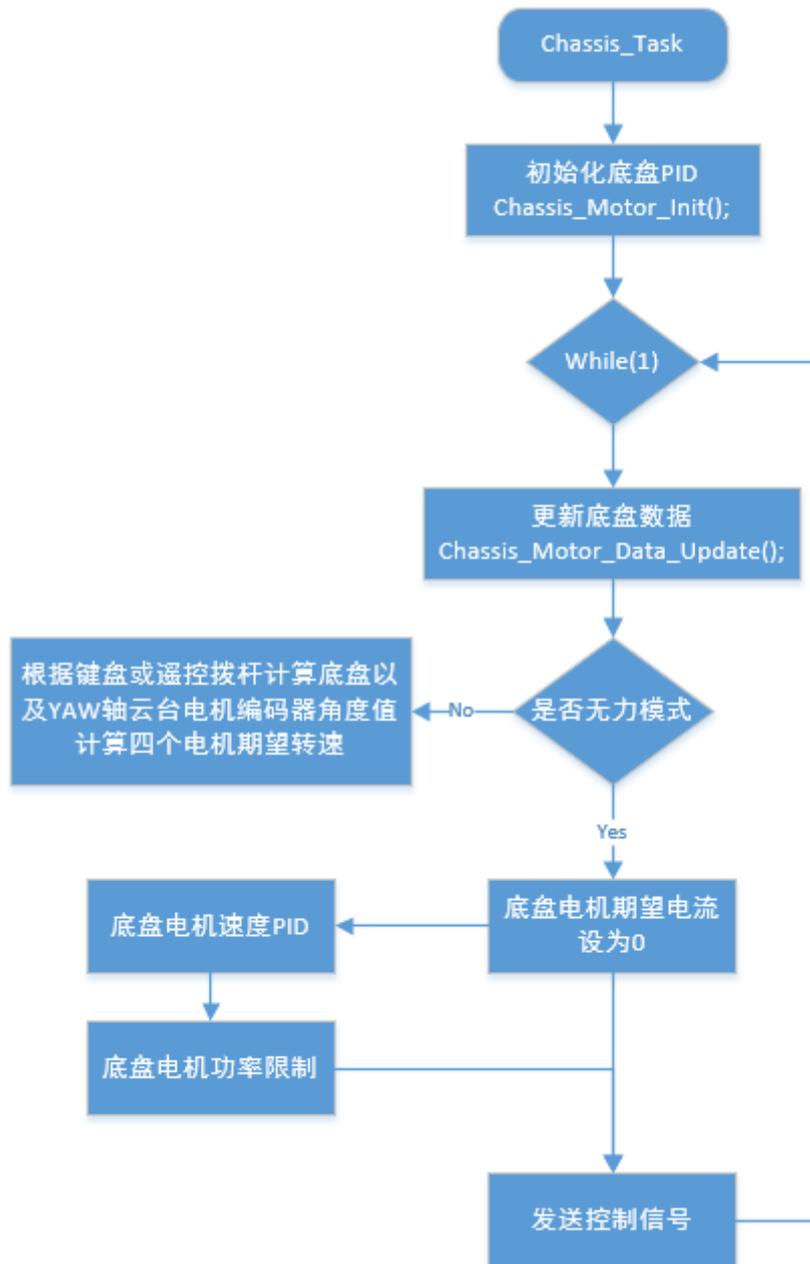
在 INS\_Task 中，通过 IMU 获得云台相对角度、角速度等信息，主要流程如下，使用了外部中断和 DMA 中断进行处理。首先进行六轴陀螺仪 BMI088 和姿态计算相关数据参数的初始化。然后进入主循环，在这里进行陀螺仪和加速度计数据的更新，并进行姿态计算，获取角速度(bmi088\_real\_data),角度(INS\_angle\_deg)和四元数矩阵(INS\_quat)。其中角速度和角度主要用于云台的控制，四元数矩阵用于自瞄计算中的坐标系变换。



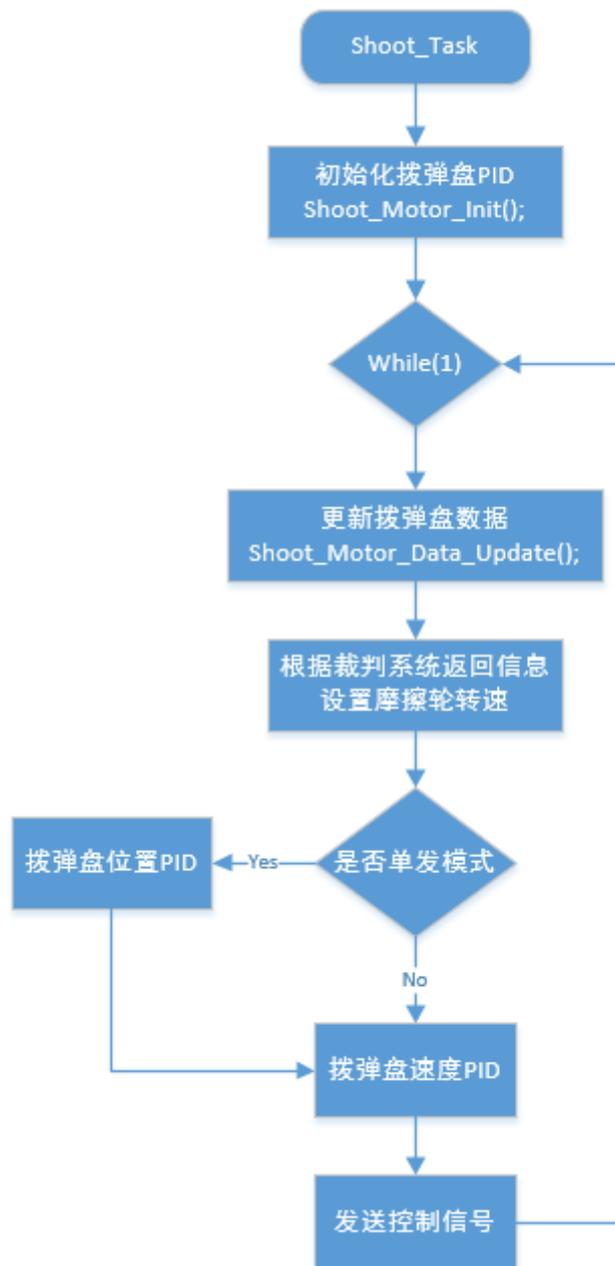
在 Gimbal\_Task 中，进行云台控制，包括使用遥控、鼠标对云台角速度、角度控制，以及开启自瞄时的角度控制，主要流程如下。首先进行云台初始化，主要为设置初值和 PID 参数初始化，然后进入主循环。循环中，首先更新云台数据，包括角度、角速度。然后根据不同的控制模式进行处理。如果鼠标或遥控拨杆有控制，则根据这个控制信号设置角速度，不进行角度环计算。如果没有控制数据，则保持上一刻的角度，根据这个角度进行角度环 PID 计算。如果自瞄有识别到目标且操作手开启自瞄，则根据自瞄数据计算出的角度进行角度环 PID 计算。角度环 PID 计算得出期望角速度。之后，根据以上几种控制模式下的期望角速度，进行速度环 PID 计算，获得期望的电机电流值。



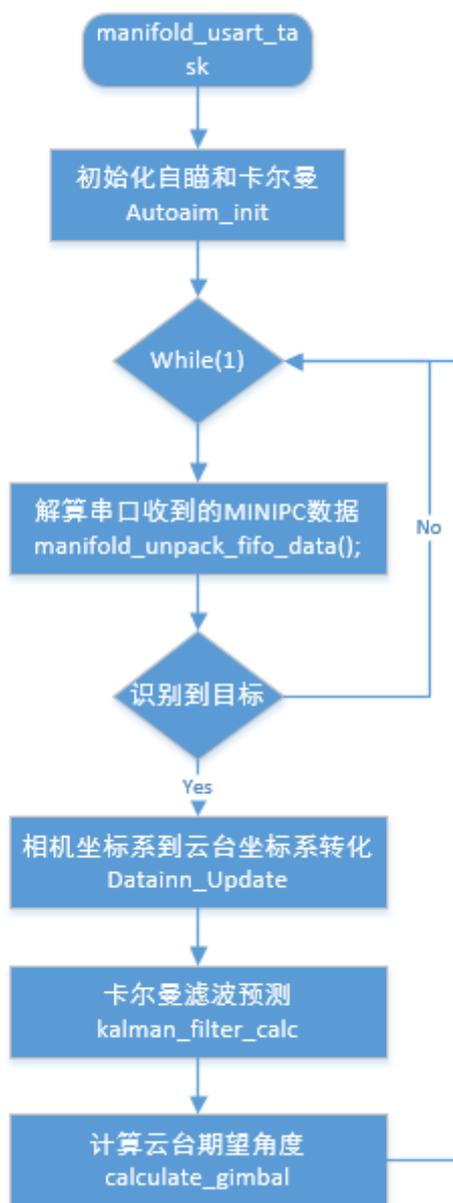
在 Chassis\_Task 中，底盘控制，主要模式为底盘无力或跟随云台，控制底盘速度方向，以及功率限制，主要流程如下。首先进行底盘初始化，主要为设置初值和 PID 参数初始化，然后进入主循环。循环中，首先更新底盘数据，包括底盘电机转速、YAW 轴云台电机编码器角度(用于底盘跟随云台)。然后根据不同的控制模式进行处理。如果底盘为无力状态，则直接设置底盘电机电流值为 0。否则，根据键盘或遥控拨杆的控制，计算机器人 XY 方向期望速度。如果机器人在小陀螺模式，则设置底盘旋转速度；否则，根据 YAW 轴云台电机编码器角度进行 PID 计算期望的底盘旋转速度。之后根据 XY 方向速度和底盘旋转速度计算四个底盘电机分别的期望转速。根据期望转速进行速度环 PID 计算，获得期望的底盘电机电流值。之后，根据超级电容状态和裁判系统返回功率，计算底盘实际的期望电流。



在 Shoot\_Task 中，进行摩擦轮转速控制，以及拨弹盘的控制，主要流程如下。其中，根据裁判系统信息，获得弹丸速度，设置摩擦轮转速。拨弹盘在 Task 开始时，进行 PID 的初始化，根据操作手操作，如果为单发模式，则首先进行位置 PID 的计算；如果为连发模式，则直接设置摩擦轮转速。之后进行拨弹盘速度环 PID，获得期望的电机电流。



在 `manifold_usart_task` 中，通过串口获得 MINIPC 所发送的自瞄信息并解算，具体流程如下。首先，初始化自瞄和卡尔曼滤波相关参数数据。在主循环中，解算串口收到的 MINIPC 数据，表示方法为相机坐标系下的目标位置。如果识别到了目标，则将目标位置通过 IMU 的四元数进行坐标变换，将目标位置映射到云台坐标系下，接着根据云台坐标系的位置，计算自瞄模式下云台的期望角度。



referee\_usart\_task 为裁判系统通讯任务，主要获取机器人状态信息，发送 UI 相关数据。

Switch\_Task为状态切换任务，根据键鼠和遥控控制信息对机器人进行状态切换，如弹仓开关、摩擦轮是否开启等

### 2.5.3 重点功能

#### 1. 自动瞄准和卡尔曼滤波、预测

由于比赛中画面延迟相对较高，且使用鼠标控制机器人云台效果较差，因此为了提高发射命中率和改善操作手的操作手感，需要实现自动瞄准的功能。

使用 NVIDIA Jetson TX2 计算平台，配合高帧率工业相机，识别目标装甲板，并通过

SolvePNP 算法解算出目标在相机坐标系下的位置。将目标位置通过串口发送给主控，主控融合陀螺仪的数据，进行坐标变化，将相机坐标系下的目标位置映射到云台坐标系下。接着通过进行卡尔曼滤波、预测，获得目标的位置和估计速度，根据目标速度和机器人发射延迟、弹速等，计算需要的瞄准提前量，从而实现带有提前量的自动瞄准。

## 2. 重力补偿

步兵发射 17mm 荧光弹丸，由于空气阻力及其他因素影响，弹道不能由弹速、云台仰角、重力加速度计算得出。可以参考到的方案有忽略垂直方向空气阻力、收敛法、数学建模等方式，这些方法大都仅在某些距离上表现好，在步兵机器人可击打区域整体表现较差。我们提取了特定角度步兵机器人弹道曲线，直接拟合后发现步兵机器人弹道符合二次曲线簇的特征，所以决定采用直接拟合法。

优点：

- 1、场上计算量小，延迟低，大部分计算拟合场下完成
- 2、准确率高，通用性强
- 3、方法简单，不需要借助复杂设备或软件

## 2.6 研发迭代过程（20）

### 2.6.1 版本迭代过程记录

#### 2.6.1.1 机械迭代记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	实现 pitch 轴 180° 旋转，及合理供弹。	2022.1.5
V1.1	针对下云台供弹弹路优化，并优化卡弹问题	2022.3.20
V1.2	优化云台发射，解决后坐力问题。	2022.5.15

### 2.6.1.2 视觉迭代记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	可以实现对装甲板的精确识别，帧率可达 360FPS	2021.10.1
V1.1	可以实现运动预测与重力补偿，调整分辨率，扩大了视野。帧率在 1280*512 分辨率下可达 170FPS	2022.3.20

### 2.6.1.3 电控迭代记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	完成对哨兵的底盘、云台控制，以及普通自瞄(无预测和重力补偿)可以使用遥控器控制。	2022.1.10
V1.1	完成自瞄的优化，添加卡尔曼滤波和预测，添加重力补偿。	2022.5.15
V1.2	完成车间通信与雷达站通信。	2022.8.10

### 2.6.2 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	机器人版本号或阶段	解决人员
1	哨兵进行自瞄时，调试参数无法符合要求，收敛慢或导致震荡。	从相机坐标系到云台坐标系的坐标变化存在问题，导致坐标变化错误。	重新确定云台坐标系和相机坐标系定义，修改错误的变换矩阵。	V1.1	嵌入式软件工程师：张千琦，王健 算法工程师：李霖杰，李博文

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
2	波胆盘卡弹	零件设计错误,导致弹丸会以某种角度锁死在波胆盘边缘。	对该零件进行尺寸调整,同时增加波胆盘边缘导角。	V1.1	机械工程师:王宁
3	下云台弹路不顺畅	下云台由于弹路过长导致的转角处弹丸卡顿	增加转角小轴承	V1.1	机械工程师:王宁,孙衍骥
4	发射机构后坐力问题	Pitch 轴转轴中轴线与弹丸初速度方向不相交,导致弹丸发射出膛瞬间云台抬头现象。	重新设计发射机构	V1.2	机械工程师:王宁,孙衍骥
5	只通过编码器无法做卡尔曼滤波预测	6020 编码器本身速度值问题	使用 C 板陀螺仪	V1.1	嵌入式软件工程师:张千琦

## 2.7 团队成员贡献（5）

姓名	基本信息 (专业、年级、队内角色)	主要负责工作内容描述	贡献度 (所有成员贡献度 合计为 100%)
王宁	机械工程、大四、哨兵负责人	负责机器人的初版机械设计及初版程序编写	25%
张千琦	航空航天类、大二、电控队员	负责硬件调试，后期电控程序编写以及整车调试	25%
孙衍骥	航空航天类、大四、机械队员	负责超级电容的研制及测试	25%
李博文	材料科学与技术、大二、视觉负责人	负责哨兵自瞄算法的更新升级	25%

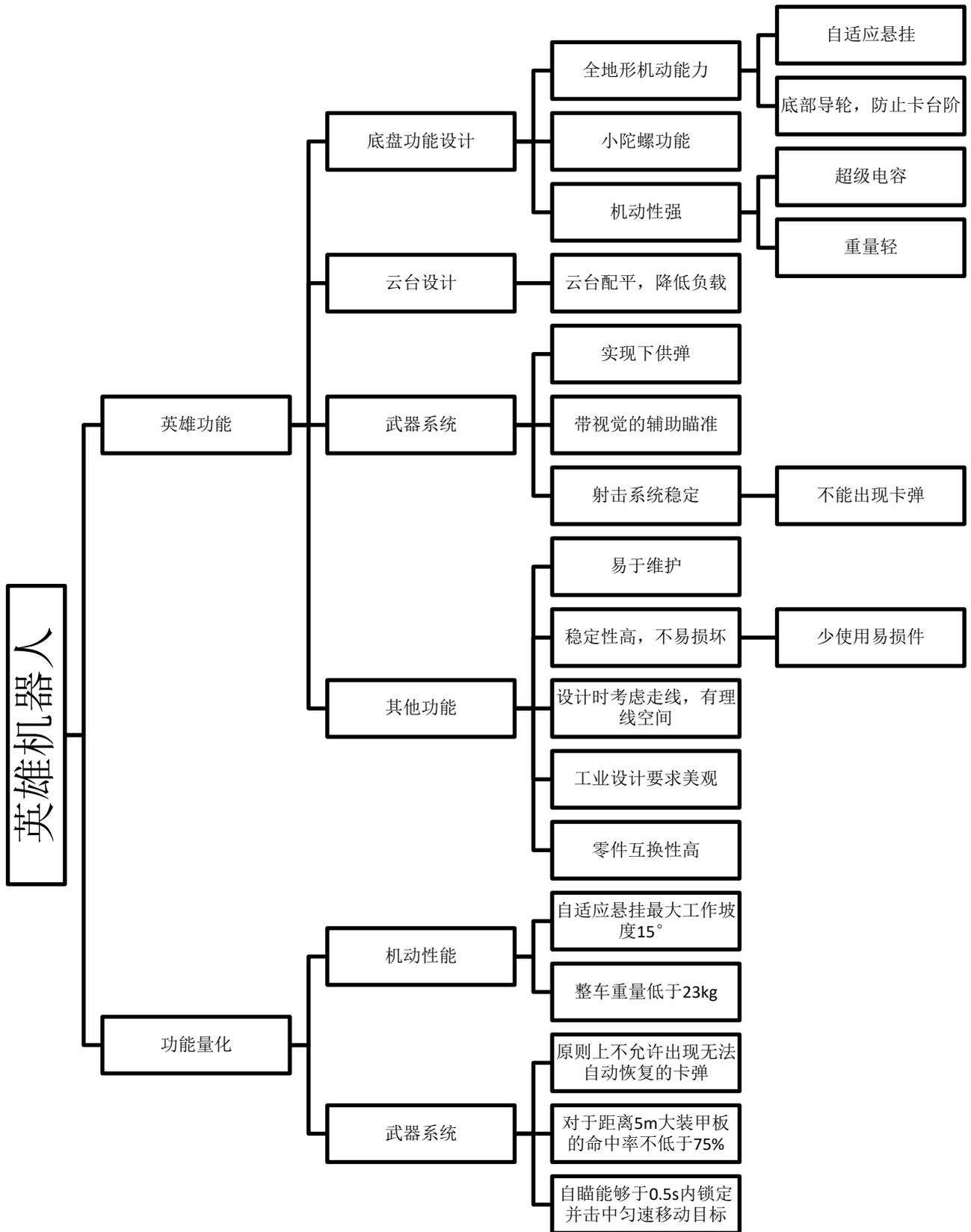
## 3. 英雄机器人

### 3.1 其它学校英雄机器人分析综述（2.5）

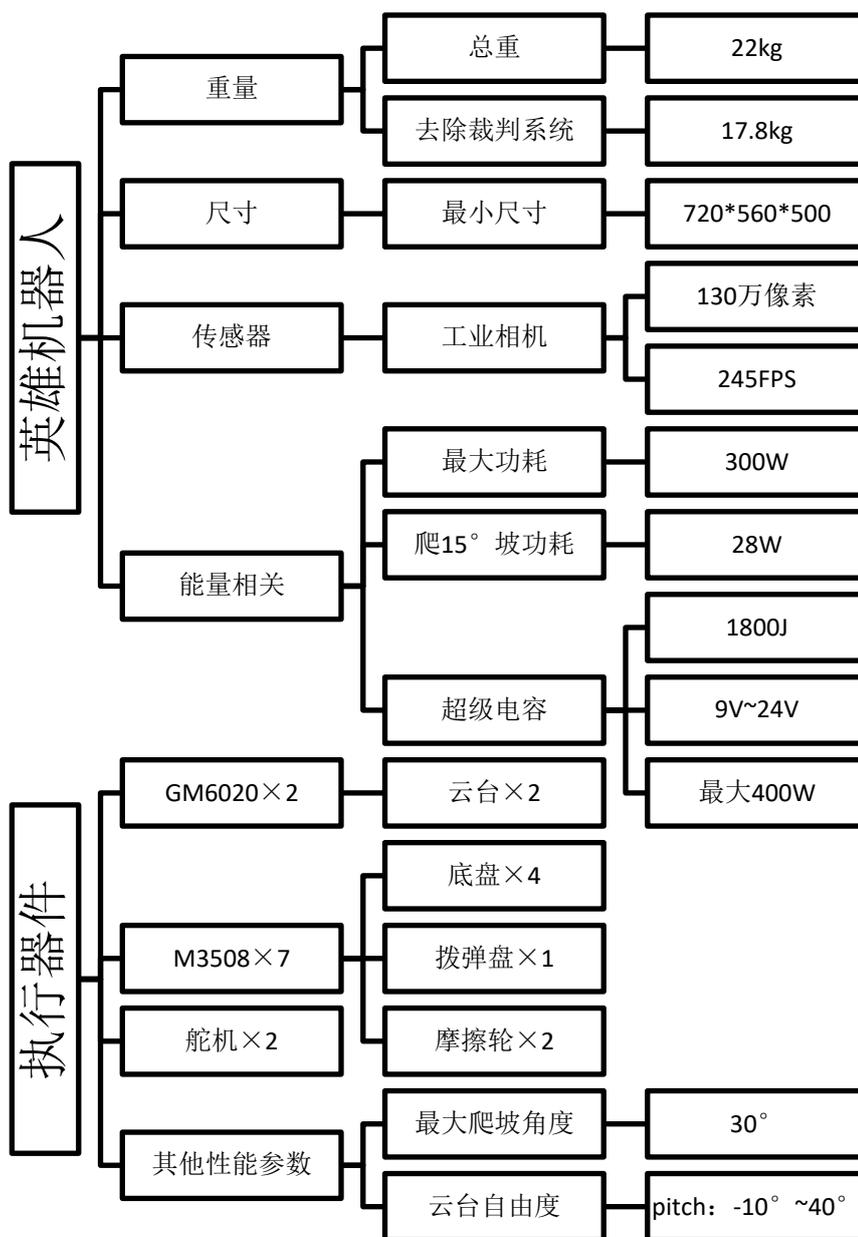
结合 2022 超级对抗赛中部分区赛、线下交流与开源文档的情况，对于 2022 赛季其他学校英雄机器人分析如下：

- 几乎全部队伍的英雄机器人均为下供弹，弹路过 yaw 与 pitch 转轴。该种构型可以简化结构，降低重量，提升可靠性。
- 大部分队伍英雄配备了自适应悬挂系统，可以增强通过性。许多队伍所配备的自适应悬挂仅使用一根连杆，结构位于车底，相较于我们本赛季使用的平行四边形构型重量稍大，但空间占用较低，是下赛季改进方向之一。
- 大部分英雄机器人使用了麦克纳姆轮，少数队伍例如东南大学使用了全向轮，其优点为小陀螺速度快、平移速度快，缺点为越野性能差，最大速度慢。
- 东南大学的英雄机器人的悬挂系统使用了两级减速器串联，刚度小，在起伏路段表现好，但是刚度过于小，对下台阶、飞坡、启动停止与射击精度均有较大影响。可以根据此构型进行改进，使得悬挂刚度与阻尼比进行两档甚至多档调节，使底盘可以适应不同场景。
- 东北大学的英雄机器人发射机构上部配备了许多传感器，例如不同焦段的镜头。这使得英雄可以适应不同自瞄场景，例如其比赛时展现出的对烧饼两种自瞄模式、带预测落点的 UI 等。哈尔滨工业大学的英雄云台上有一 AprilTag，其可以辅助雷达站进行定位。这两种举措都是在算法上做出的突破。本赛季对于英雄算法研发程度不高，未来可以借鉴方案，在算法方向进行改进。

# 机器人功能定义 (5)



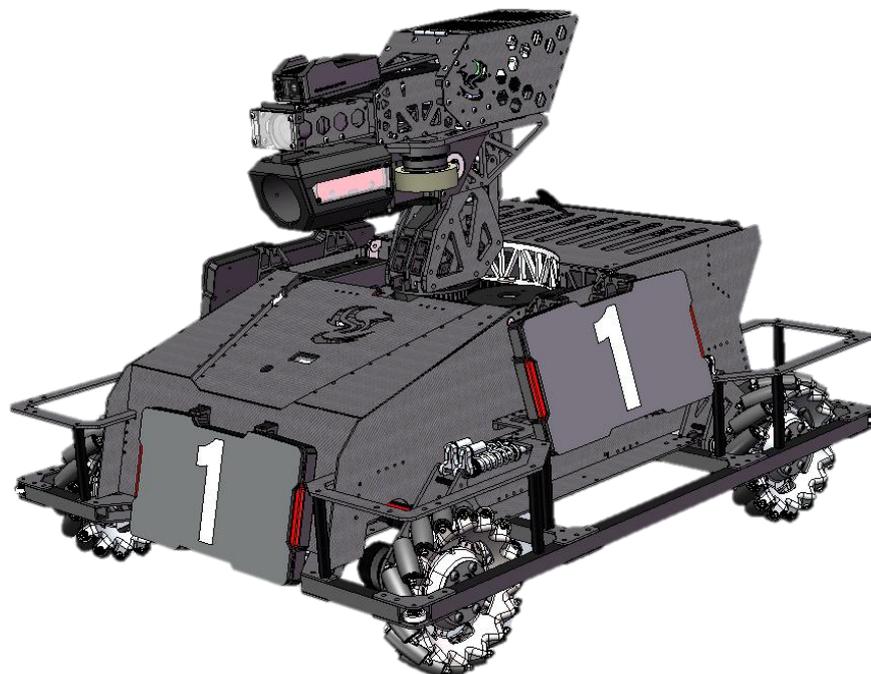
## 3.2 机器人核心参数 (2.5)



## 3.3 设计方案 (25)

### 3.3.1 机械结构设计

#### 7.1.1.1 整体设计



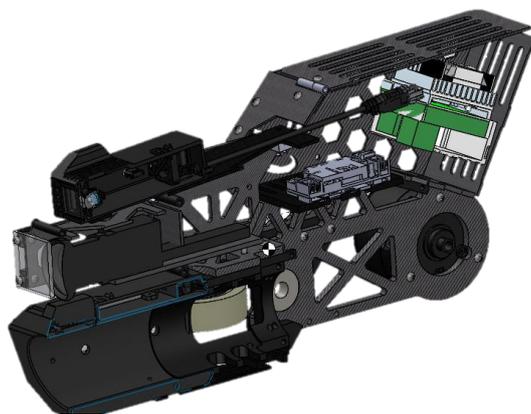
英雄机器人如图所示，射击部分采用下供弹、一对摩擦轮，底盘采用了麦克纳姆轮、自适应悬挂，视觉部分配备了一个摄像头与一个 TX2。

#### 7.1.1.2 发射机构设计

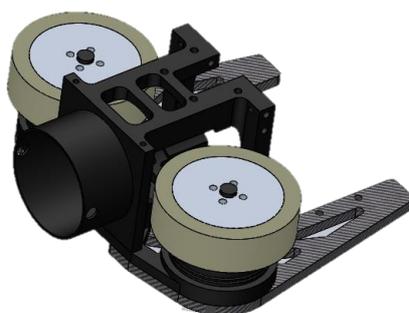
发射方案沿用之前方案，摩擦轮采用了去除了减速箱的 M3508 电机，摩擦轮使用了直径 60mm，硬度邵氏 55 度的聚氨酯包胶轮，摩擦轮间距 97mm。



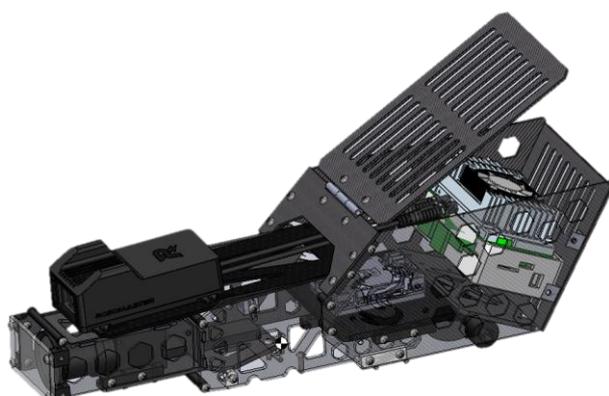
发射机构采用了无枪管构型，42mm 弹丸接触摩擦轮后即与枪管无接触。此构型避免了由于弹丸与枪管内壁碰撞造成的精度损失，同时避免了枪管加工公差造成的散布增大。相比于往年有枪管构型，10m/s 射速下命中率提升约 40%，16m/s 射速提升约 20%（实验结果未严格控制变量，仅供参考）。



M3508 电机凸台与 6061 铝合金定位板连接，保证了摩擦轮间距的误差在 $\pm 0.02\text{mm}$  以内，同时铝合金相较于碳纤维板材抗弯强度高，在重量相等的情况下可有效提升设计准确度。同时摩擦轮设计的易于更换。



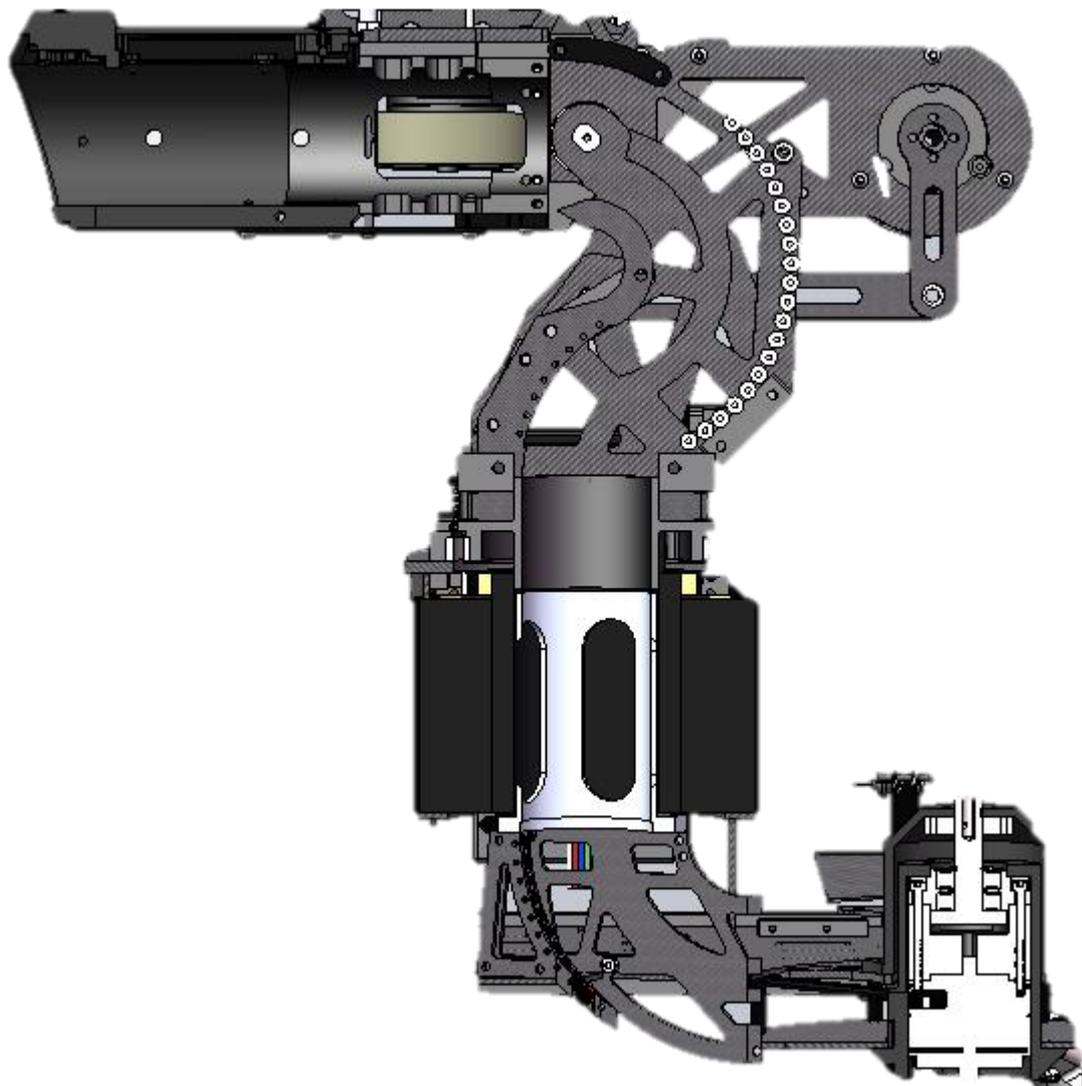
平行于测速机构安装相机与开发板，保证三者坐标系完全平行。图传模块斜向下安装，可以使英雄操作手获得更大视野范围，弹丸落点更靠中心。上部空间用于布线与安装 TX2 运算平台。上盖使用合页便于打开。



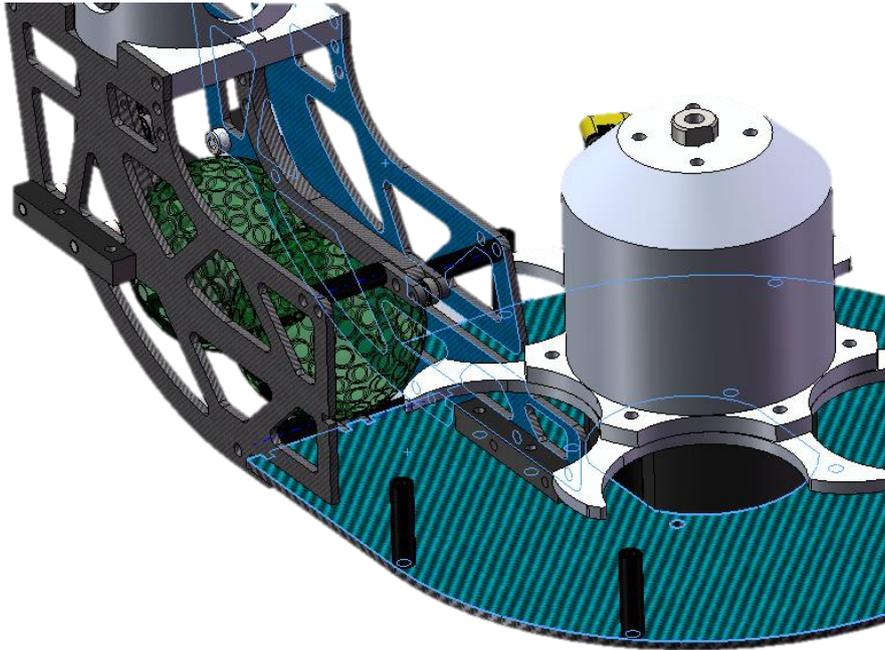
### 7.1.1.3 供弹机构设计

供弹路径如图所示，由后部弹舱供弹，经过中心半径为 50mm 的圆弧，由滑环向上，经过半

径 45 的大圆弧，通过过弹路中心的 pitch 转轴进入摩擦轮。两端圆弧外侧均采用轴承 692 进行过渡，降低弹丸通过阻力。



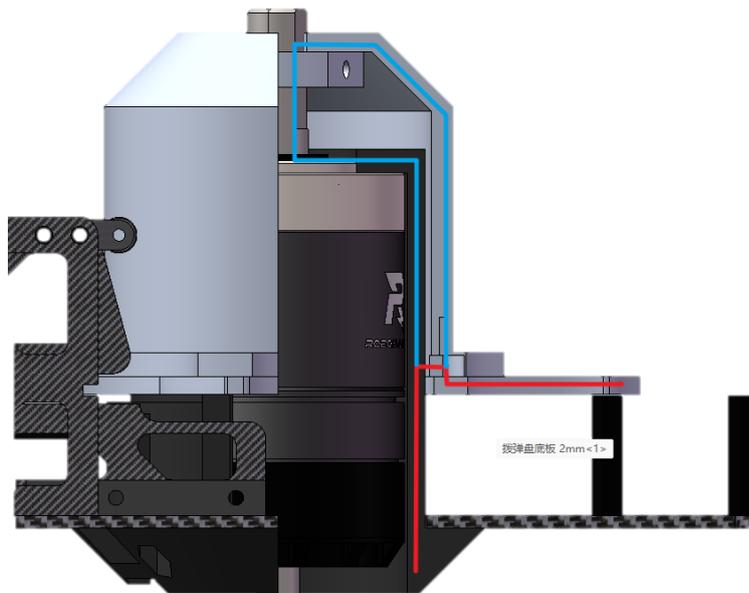
通过受力分析以及在弹路相关位置涂抹颜料观察磨损的方法，我们得到了供弹阻力较大的区域，主要位于拨弹盘底部底板与出口一侧，在此类位置张贴了特氟龙贴纸，对于降低阻力有明显的效果。



前期测试拨弹盘磨损与损坏较为严重，且拨弹盘刚度不足会造成卡弹问题，于是更换拨弹盘与拨弹盘连接件为金属材质。

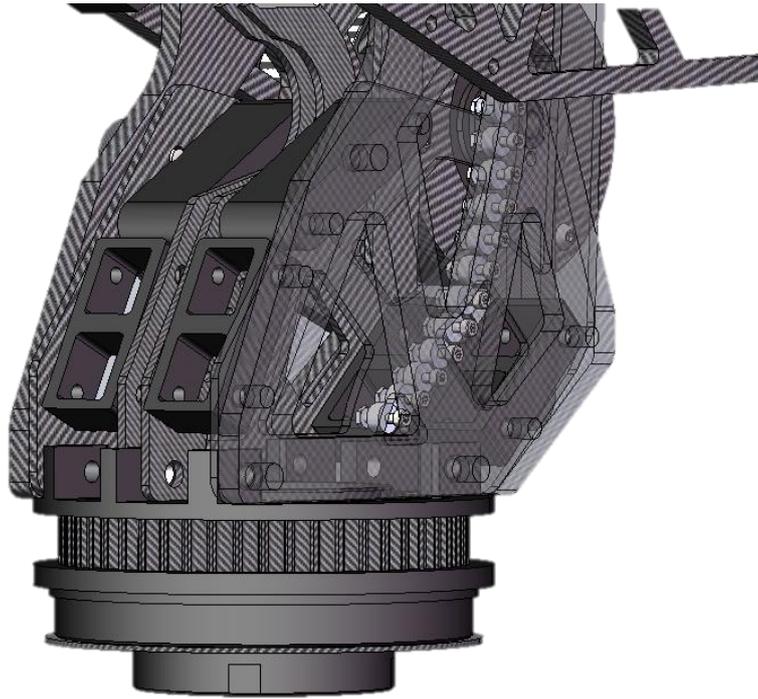
拨弹盘与弹丸接触位置为渐开线设计，且考虑到弹丸旋转问题降低了  $r$  值。但是经过测试仍会出现弹丸水平受力方向为钝角，弹丸自锁从而卡弹，后更改了拨弹盘材质后问题得到解决。

拨弹盘电机 M3508 采用倒置嵌入式安装，同时于电机于波胆盘连接件之间添加了 6 个润滑轴承，降低了弯矩的传力路径，降低了卡弹的风险。如图，红色为改良后的路径。



为了避免射击时后坐力对于弹丸影响过大，以及防止翻车后云台架受到损坏，云台架采用了双层碳板设计以，云台架底座由 6061 铝合金一体构成，通过 RA8008 交叉滚子轴承与底盘

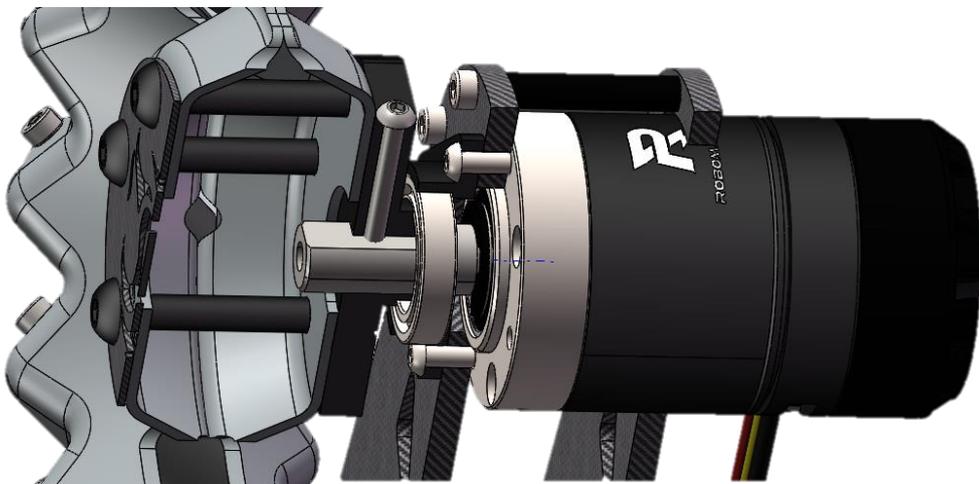
连接以提升云台强度。

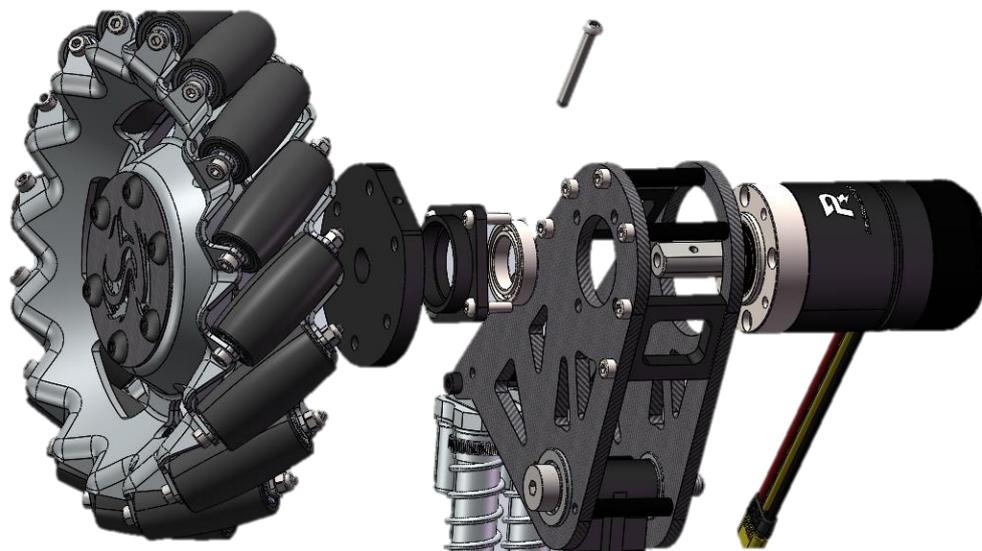


#### 7.1.1.4 轮系设计

轮组于上一年设计上改进，减少了零件数量于体积，降低重量。麦克纳姆轮通过法兰盘于电机相连，法兰盘中间掏有 D 型孔，垂直轴有一 3mm 通孔，通过螺钉与电机上的 M3 螺纹孔相连。螺钉可以使 D 型轴与法兰盘 D 型孔紧密贴合，消除了游隙。拆除该固定螺钉即可拆除法兰与麦克纳姆轮，整体拆装小于 1 分钟，更换电机小于 3 分钟，便于维修。

法兰盘外部套有轴承，外部轴承盖与电机固定孔位相同，可以将力传递至纵臂上，增强了轮系强度。





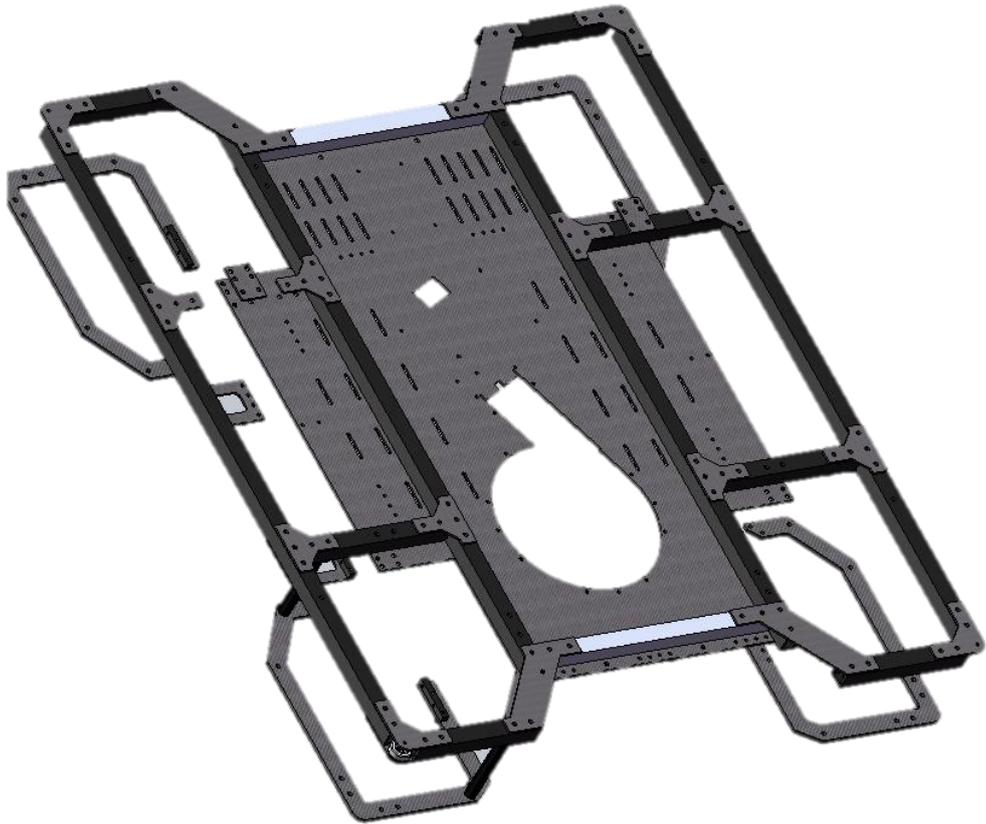
采用了平行四连杆连杆构型，两个红圈圈出的轴承与车体连接，为固定转轴，避震器水平安装，连杆运动使得避震器支点同时运动，左前轮抬起的同时右后轮同时抬起，这可以使机器人在上坡时能够四轮与地面都能够接触。

相比于其他构型，平行四边形连杆结构简单，重量轻，但是其占用空间大，影响了对弹舱影响较大。

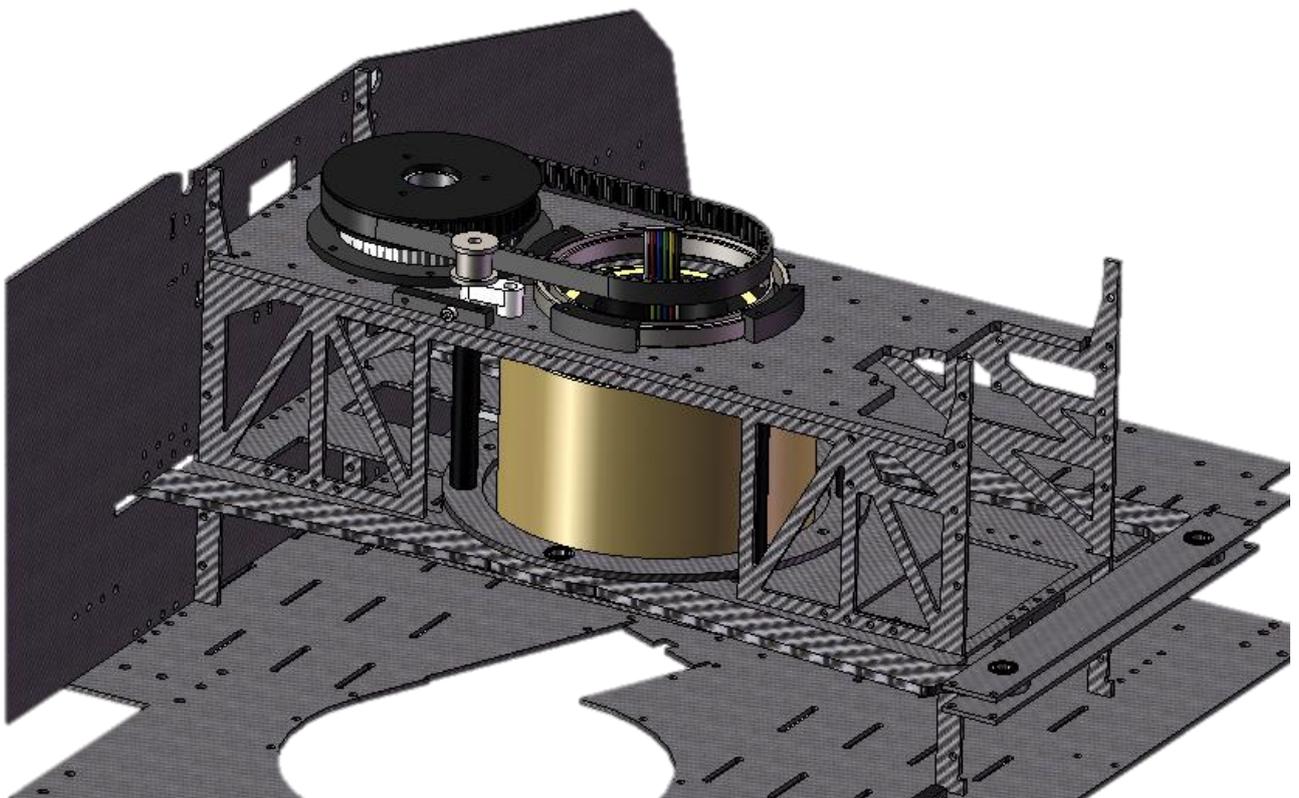


#### 7.1.1.5 车体设计

车体底部由四根铝方管纵向贯穿，横向由八根短铝方管组成，所有铝方管在同一水平面上。铝方管选型与步兵相同。铝方管上部为一整块碳纤维板，提供整体强度。



中部云台支撑采用了一块 4mm 厚碳纤维板，同时两块 2mm 碳纤维板材作为肋提升云台与两同步轮的刚度，提升设计命中率、防止同步带脱齿。



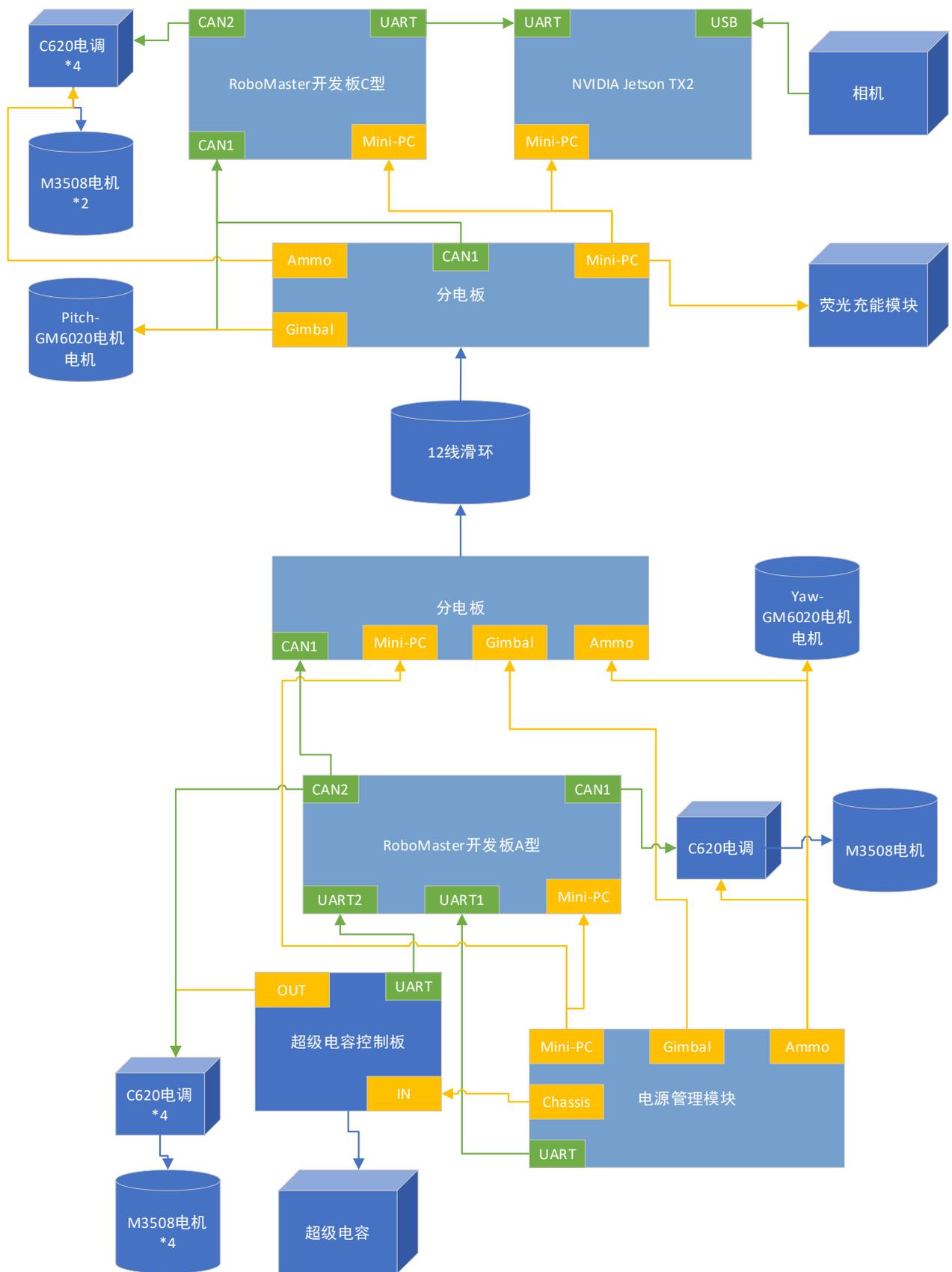
车体前部为电气元件区，安装了电源管理、控制板、分电板、超级电容等零件。后部为弹舱，可容纳 80 发 42mm 弹丸。



### 3.3.2 硬件设计

硬件方面主控选择 RoboMaster 开发板 C 型作为云台的主控模块，RoboMaster 开发板 A 型作为云台的主控模块，NVIDIA Jetson TX2 作为计算平台，自研基于 STM32F3 的超级电容控制模块。通过滑环和两块自制分电板，连接云台和底盘电路，以实现云台 360 度旋转。

### 3.3.2.1 整机硬件框图



### 3.3.2.2 超级电容硬件设计

英雄超级电容设计与 1.4.2.2 步兵超级电容设计相同，这里不再赘述

### 3.3.3 软件设计

软件设计与 1.4.3 步兵的软件设计基本相同，这里仅对不同部分进行说明。

#### robomaster 开发板 C 型

云台控制运用了一块 robomaster 开发板 C 型，在其上运行 1.4.3 步兵的软件设计除 Chassis\_Task 与 referee\_usart\_task 以外的其他任务。与底盘通信采用利用 CAN1 总线通信，发送 6 字节信息，分别为两字节  $V_x$ ，两字节  $V_y$ ，一字节  $W_z$ ，一字节状态信息。

#### robomaster 开发板 A 型

底盘控制运用了一块 robomaster 开发板 A 型，在其上运行 Chassis Task 与 referee\_usart\_task，两任务与 1.4.3 步兵的软件设计相同。除任务部分配置方法与 robomaster 开发板 C 型相同。通过 CAN1 总线通信，转发裁判系统信息。

### 3.3.4 算法设计

算法设计与 1.4.4 步兵算法设计完全相同，详见上文。

## 3.4 研发迭代过程（10）

### 3.4.1 版本迭代过程记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	完成了第一版完整英雄的组装，完成了基本代码调试	2022.1.10
V1.1	修改了拨弹盘机构	2022.3.10

版本号或阶段	功能或性能详细说明	完成时间
V2.0	修改了底盘结构，重做侧板，修改弹舱位置，加强云台支撑	2022.4.10
V2.1	优化云台弹路，修改拨弹盘机构	2022.4.25
V2.2	修改了摩擦轮间距	2022.6.10

### 3.4.2 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	机器人版本号或阶段	解决人员
1	英雄最大弹速仅能达到 12/s	摩擦轮过硬	更换摩擦轮	V1.0	机械工程师: 曹阔
2	英雄拨弹盘位置卡弹	拨弹盘形状设计错误	重新设计拨弹盘	V1.1	机械工程师: 曹阔
3	弹舱设计不合理，卡弹	弹舱设计不合理	重新设计弹舱	V2.0	机械工程师: 曹阔
4	卡弹	云台弹路有缺口，拨弹盘摩擦轮过大	云台补上缺口，更换拨弹盘材质	V2.1	机械工程师: 曹阔、路程
5	超级电容烧毁	对于裁判系统数据处理有问题，充电功率过大	修改超级电容程序	V2.1	硬件工程师: 李高峰

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
6	更换发光弹丸后，无法达到设计弹速	摩擦轮间距过大	更改定位板，减小摩擦轮间距	V2.2	机械工程师：路程

### 3.5 团队成员贡献（2.5）

姓名	基本信息 (专业、年级、队内角色)	主要负责工作内容描述	贡献度 (所有成员贡献度合计为 100%)
曹阔	飞行器动力工程，大四，机械、电控/嵌入式成员	负责机器人整个底盘设计 赛季初负责嵌入式开发，包括底盘控制、云台控制、视觉系统的嵌入式环境开发等	20%
路程	机械工程，大二，机械组成员	负责机器人整个云台设计	20%
曾新宇	机械工程，大二，电控/嵌入式成员	赛季末负责嵌入式开发，包括底盘控制、云台控制、裁判系统通信与 UI 绘制等	20%
李霖杰	飞行器动力工程，大四，算法组成员	负责视觉系统的运算平台端开发	10%
王健	导航、制导与控制，研一，电控/嵌入式成员	负责视觉系统的嵌入式环境开发	10%
杨洛	机械工程，大二，电控/嵌入式成员	负责视觉系统的嵌入式环境开发	10%

姓名	基本信息 (专业、年级、队内角色)	主要负责工作内容描述	贡献度 (所有成员贡献度 合计为 100%)
李高峰	飞行器动力工程，大四，硬件 组成员	负责超级电容模块开发	10%

### 3.6 参考文献 (2.5)

## 4. 工程机器人

### 4.1 其它学校工程机器人分析综述（2.5）

通过观察现场比赛及视频分析，发现大部分学校在这个赛季使用吸盘进行矿石的拿取以及兑换，相较于曾经使用夹爪进行矿石的夹取和兑换，由于工程机器人对功率没有限制，且由于矿石的结构特征，吸盘相较于夹爪更加稳定，在对矿石的空接部分中，吸盘相较于夹爪可以到达更高的高度，且吸盘的尺寸更小，更节约空间，所以在第三版工程上采用的是吸盘作为取矿以及兑矿的机构。

工程机器人的重心也是一个问题，东北大学的工程机器人在分区赛中表现十分优秀，其速度很快且十分稳定，经过观察发现其抬升机构在不工作时可以收至底盘高度，这是二级抬升无法做到的，很显然其采用了三级抬升机构；哈尔滨工业大学的工程机器人则采用的是二级抬升，但是由于其机构精细控制优秀，也能表现出不错的结果；中国矿业大学的工程机器人在机器人前方设计了用于兑换矿石以及拿取地面上矿石的小型机械臂，虽然没有常规机械臂的多自由度，但是也是一个非常优秀的尝试，认为机械臂是在时间充足的情况下最优秀的机构。

### 4.2 机器人功能定义（5）

#### 工程机器人设计思路

第一版的设计初衷为验证设想的可行性的试验机，也作为后续的备用机。设计上都是实用性为重，保证能实现拿取矿石、兑换矿石和救援其他机器人的基本功能。第二版则是基于第一版，对功能和结构进一步改善，更改并删去不合理的机构，引入了高压气体作为动力。第三版工程机器人针对国赛的变化进行改进，主要针对矿石兑换机构进行设计。

工程机器人设计时分为底盘、抬升和取兑矿模组三部分。底盘负责完成救援和存矿的功能，抬升部分负责确定高度以及伸出的长度，取兑矿模组则负责完成取矿、兑矿和拿取障碍块的功能。

**工程机器人底盘部分：**①救援：救援分为刷卡救援以及抓取救援方式，其中抓取救援方式主要针对英雄机器人，刷卡救援方式针对平衡步兵机器人以及英雄机器人。②存矿：能够在获取矿石之后让矿石保持在底盘上不影响到走线即可。

工程机器人抬升部分：能够到达最大尺寸的同时不超出限制尺寸即可。

工程机器人取兑矿部分：在抬升机构到达预定位置之后，能够保证不掉落矿石，同时在兑换矿石的时候完成 x、y、z、pitch 和 roll 五个自由度的兑矿要求。

### 4.3 机器人核心参数（2.5）

第一版基础参数：

参数类型	数值
重量	34.57kg
重心高度	287mm
初始尺寸	600*600*600mm
伸展尺寸	600*1200*1000mm

执行器件：

名称	数量	用途
Robomaster 3508 减速直流电机	4	抬升机构竖直方向运动提供动力
Robomaster 3508 减速直流电机	4	车体运动
Robomaster 2006 P36 直流无刷 减速电机	1	刷卡救援
Robomaster 2006 P36 直流无刷 减速电机	2	转动矿石
75kg 舵机	2	拖拽救援
35kg 舵机	1	云台转动
75kg 舵机	2	抓取障碍块

110kg 无刷舵机	1	夹取矿石
Robomaster 2006 P36 直流无刷 减速电机	2	翻转夹取机构
Robomaster 2006 P36 直流无刷 减速电机	2	伸出机构

第二版基础参数：

参数类型	数值
重量	36.08kg
重心高度	234mm
初始尺寸	600*600*600mm
伸展尺寸	600*1200*1000mm

执行器件：

名称	数量	用途
Robomaster 3508 减速直流电机	4	抬升机构竖直方向运动提供动力
Robomaster 3508 减速直流电机	4	车体运动
Robomaster 2006 P36 直流无刷 减速电机	1	刷卡救援
Robomaster 2006 P36 直流无刷 减速电机	1	转动矿石
75kg 舵机	2	拖拽救援
35kg 舵机	1	云台转动
75kg 舵机	2	抓取障碍块
Robomaster 2006 P36 直流无刷	2	翻转夹取机构

减速电机		
------	--	--

其他参数：

参数类型	数值
最大移动速度	4m/s
最大爬坡角度	30°
云台自由度	1
使用气体压力	0.7mpa

第三版基础参数：

参数类型	数值
重量	24.88kg
重心高度	234mm
初始尺寸	600*600*600mm
伸展尺寸	600*1300*1000mm

执行器件：

名称	数量	用途
Robomaster 3508 减速直流电机	2	抬升机构竖直方向运动提供动力
Robomaster 3508 减速直流电机	4	车体运动
Robomaster 2006 P36 直流无刷 减速电机	1	刷卡救援
Robomaster 2006 P36 直流无刷 减速电机	5	吸盘移动
30kg 舵机	1	吸盘移动

脉塔 RMD-L5010 无刷伺服电机	2	云台转动
40kg 舵机	2	钩爪救援

其他参数：

参数类型	数值
最大移动速度	4m/s
最大爬坡角度	30°
云台自由度	2
吸盘自由度	5

## 4.4 设计方案（25）

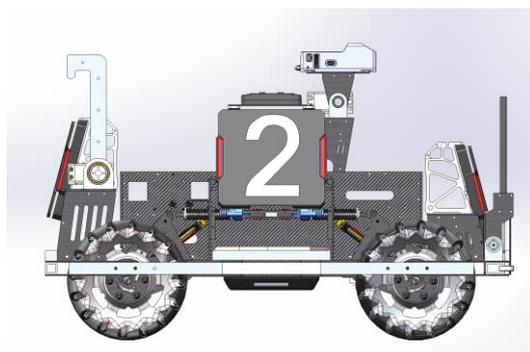
### 4.4.1 机械结构设计

第一版机械结构设计说明

第一版的设计初衷为验证设想的可行性的试验机，也作为后续的备用机。设计上都是实用性为重，没有突出的特点。

#### （1）底盘设计

为了能进行全向移动，底盘采用 4 个麦克纳姆轮 X 型分布，使用 4 个 M3508 电机，同时为了防止侧翻以及减轻起伏路段的影响，选用自适应悬挂。底盘主要框架由铝方组成，铝方转角处使用 2mm 碳板连接，保证了底盘结构的牢固性，轮系结构与步兵一致。底盘上包括了拖拽救援和拾取障碍块的机构，均由舵机实现驱动。由于结构的限制，视角的转换通过安装



于车体不同位置的摄像头实现，带有图传的云台置于底盘前部，仅有 pitch 向自由度，通过观看屏幕获得其他摄像头的视角。

图 4.4.1 第一版工程底盘

## (2) 钩爪救援机构

拖拽救援机构采用舵机直驱，救援时舵机旋转，钩子钩住被救援车辆的救援部位，可以有效防止被救援车辆因上下颠簸而脱落，保证了救援的可靠性。

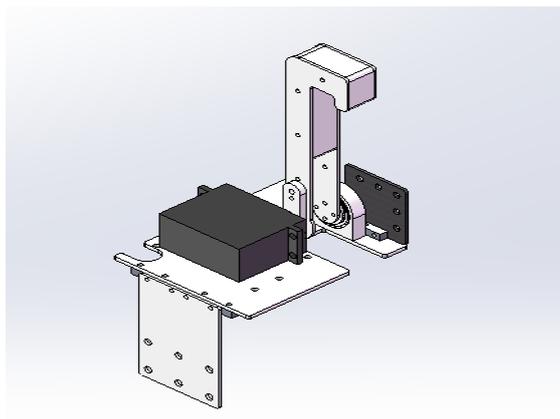
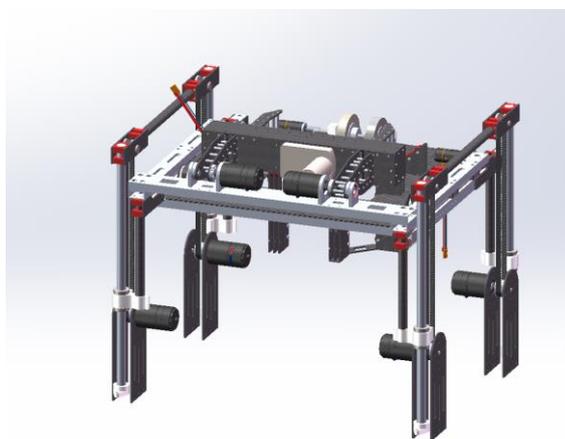


图 4.4.2 拖拽救援机构



## (3) 抬升及转矿机构

抬升部分使用四根光轴作为滑轨，直线轴承为滑块。使用四个 3508 减速电机实现上下运动。转矿以及存矿机构安装在抬升机构的上层，由 2006 减速电机驱动两对包胶轮实现两个轴的转动产生摩擦力进而实现矿石两个自由度的转动。

图 4.4.3 抬升机构

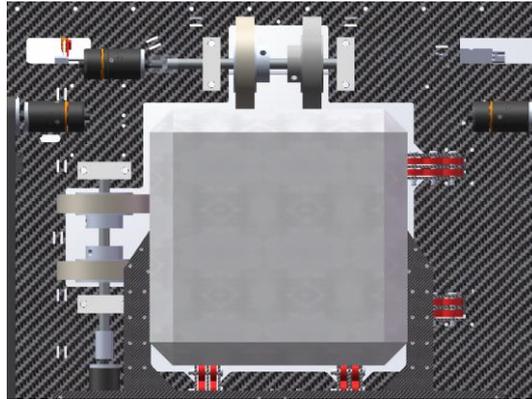


图 4.4.4 转矿机构

#### (4) 前伸机构

前伸机构也安装在抬升机构的上层，通过 2006 减速电机带动同步轮，实现前后伸缩。前端夹爪由舵机提供动力，采用曲柄滑块机构实现夹爪的横移。通过对舵机的位置控制，借助连杆和滑块，控制夹爪的夹紧和放松。矿石重约 600~700g，材料为 EVA，通过改变夹取结构与矿石的摩擦系数，使其摩擦系数至少为  $\mu=0.5$ 。其夹紧力为  $F = mg / (2\mu) = 6.86\text{N}$ 。考虑长度宽度的合理性，摇杆夹角取  $45^\circ$ ，长度取 0.05m，得到舵机扭矩至少需为  $T=7\text{kg}\cdot\text{cm}$ ，考虑安全系数  $S=2$  以及摩擦损耗。舵机扭矩至少  $20\text{kg}\cdot\text{cm}$ 。夹爪伸出部分加矿石估计不超过 2kg，重力臂为 0.1m，所需最大扭矩为  $T_2=1.96\text{N}\cdot\text{m}$ ，考虑安全系数  $S=2$ ，扭矩至少为  $T_{2\text{min}}=4\text{N}\cdot\text{m}$ 。查得 M3508 电机稳定扭矩  $3\text{N}\cdot\text{m}$ ，则采用两个 M3508 电机进行控制。

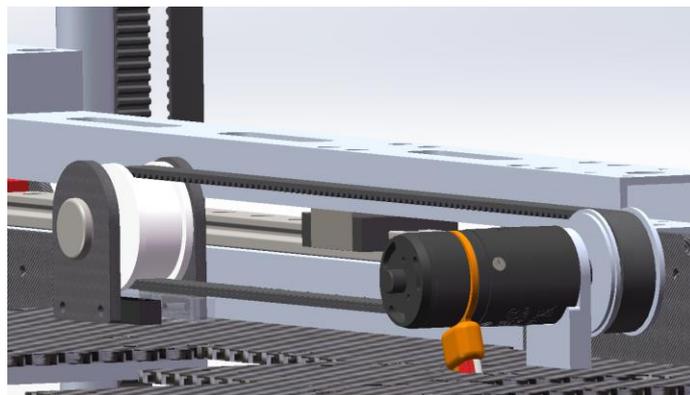


图 4.4.5 前伸机构整体

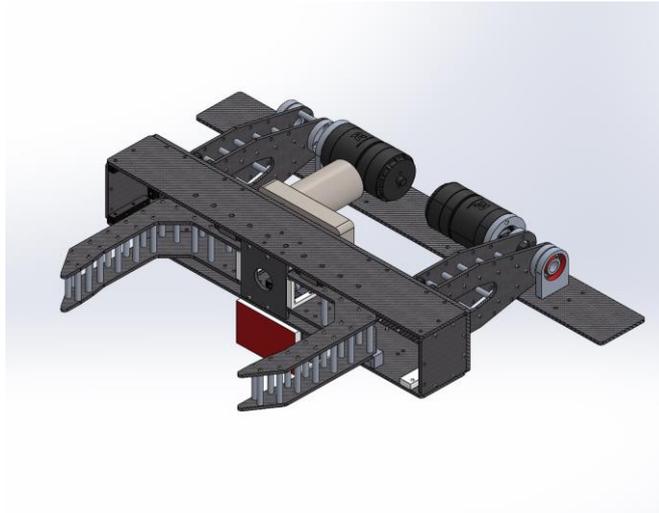


图 4.4.6 夹爪部分

## 第二版机械结构设计说明

第二版的机械结构在第一版的基础上进行了修改，但基本排布以及工作原理并没有较大的变动，对一些机构进行精细化，对于一些不合适的机构进行了替换。

### (1) 底盘设计

第二版工程机器人的底盘进行了加固，将第一版机器人底盘上由于冲撞而受损的部分替换成铝方，但并没有改变轮系的布置，仍采用自适应悬挂。

图 4.4.7 第二版工程机器人底盘



### (2) 障碍块拾取机构

增加了障碍块夹取机构,为了抬起重达 2250g 的障碍块,使用两个 75kg·cm(即 7.5 N·m)的舵机。舵机转轴与碳管前端距离 225mm。抬起障碍块时障碍块重心几乎在碳管末端,所需扭矩  $T=F*d=5.06 \text{ N}\cdot\text{m}$ 。取安全系数  $S=2$ ,则至少需要的扭矩为  $T=10.12 \text{ N}\cdot\text{m}$ 。两个舵机所提供扭矩为 15 N·m,故可行。又由于障碍块形状的原因,该机构将障碍块抬起时角度不能过大,会与车体干涉,故实际应用时在碳管中部缠上胶带,防止障碍块下滑。

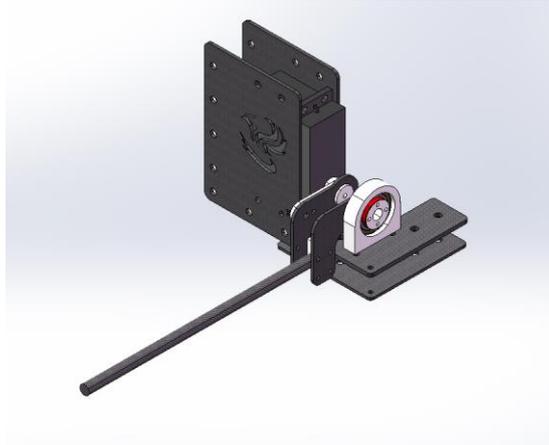


图 4.4.8 障碍块拾取机构

### (3) 矿石拾取机构

地面矿石拾取通过一个夹取机构实现,夹取机构由一个 2006 驱动一个正反牙丝杆组成,丝杆导程为 2mm,足够夹紧矿石,翻转机构由一个 3508 驱动,夹取机构加矿石大致为重量为 1.675kg,重心距离转轴大约 100mm。故翻转所需扭矩  $T=F*d=1.675 \text{ N}\cdot\text{m}$ 。取安全系数  $S=1.5$ ,则至少需要  $T=2.51 \text{ N}\cdot\text{m}$ 。3508 扭矩为 3 N·m,故可行。伸出机构由一个 3508 电机驱动一个导程为 2 的滚珠丝杆,通过连杆控制夹取与翻转机构的伸出与升降。

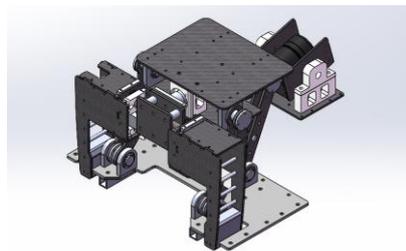


图 4.4.9 拾取机构

### (4) 伸卡救援机构

工程伸卡救援结构一般位于工程车的底部,其目的是要将救援卡伸于其他兵种底盘下的 RFID(即场地交互模块),持续一段时间后其他兵种机器人即可复活。首先要考虑行程问题,伸卡结构要从工程底盘伸出到达其他兵种的底盘,因其他兵种底盘大小限制为

600mm\*600mm，且 RFID 一般位于底盘中央，因此伸卡结构行程至少要有 300mm，在实际情况下需要预留 50mm 左右的行程，一般在 350-400mm 较好。其次需要考虑高度问题，高度由三个因素决定，分别是底盘高度、离地高度和其他兵种的底盘高度。底盘高度限制了范围，离地高度决定高度下限而他兵种的底盘高度决定高度上限。同时还需要保证宽度和强度，综合考虑之后采用 2006 加上齿轮齿条进行驱动。

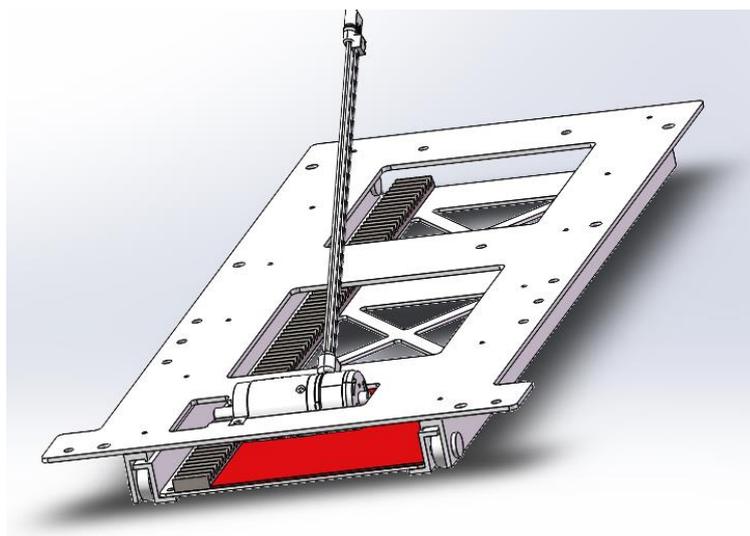
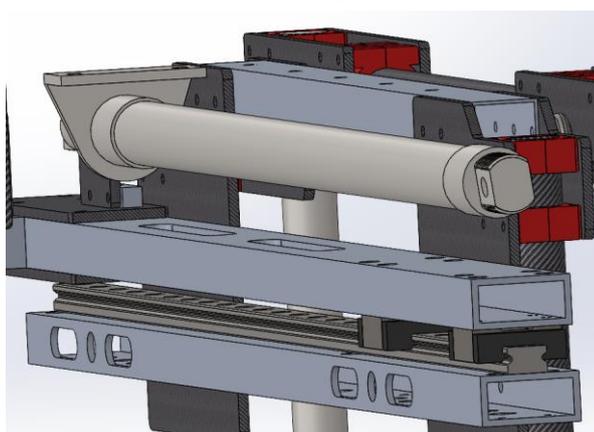


图 4.4.10 伸卡救援机构

第二版工程机器人的抬升机构部分相较于第一版基本没有变化，依然采用光轴与直线导轨的组合。

#### （5）前伸及夹取机构

前伸部分在则不再采用采用气动驱动，响应较快，且力量较大，分区赛时在某场比赛时出现仅用单边气缸驱动的意外情况，但依然能稳定工作。缺点是无法指定行程，完全是由气缸行程确定设计的前伸尺寸，在国赛兑换站的要求下，不具有优势。备赛国赛期间，设计时改为了由齿轮齿条机构，便于闭环控制。首先尝试了模数 1 的齿轮齿条，测试时发现由于外伸



梁挠度的问题导致啮合不稳定的情况，更换为模数 2 的齿轮齿条后，问题有所改善。

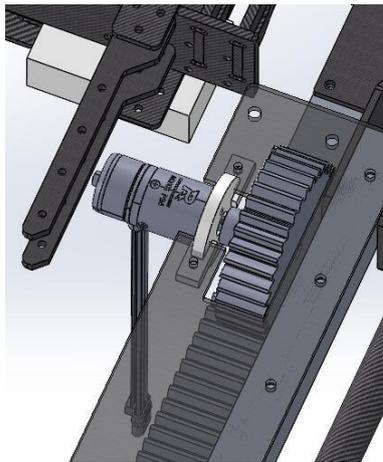


图 4.4.11 前伸部分——气动

图 4.4.12 前伸部分——齿轮传动

夹取部分采用常规双 3508 驱动摇臂带动爪子旋转，爪子采用气动驱动，以达到响应空接矿石的速度。主体气缸为淘宝购置的夹爪气缸，相对与使用单个普通气缸，有对称移动的优势，能在左右移动相同距离，对准 2、4 号矿石，同时夹爪两边几乎同时接触矿石，也增大了空接的成功率，但是本身重量较大。空接时为光电传感器配合电磁阀，当检测到低电平后电磁阀响应，使夹爪闭合。云台也同时在前端，以获得最佳的参考视野，但是缺少对车体周围的观察能力。

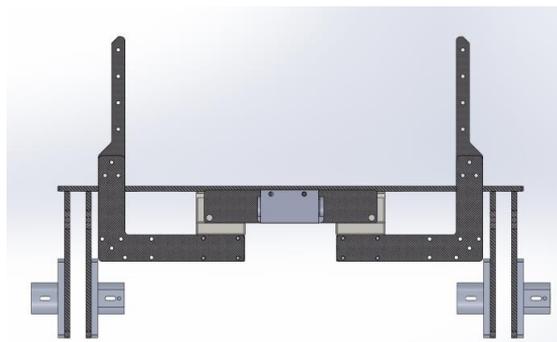


图 4.4.13 夹爪机构

### 第三版机械结构设计说明

#### (1) 底盘设计

整车质量初步估计重 30kg，采用 4 个麦克拉姆轮 X 型分布，需用 4 个 M3508 电机。悬挂系统采用独立式悬挂，可有效减轻地面起伏的影响，经过分析论证，认为自适应性悬挂在工程

机器人上并不是必须存在的结构，并且自适应底盘的空间占用较大，为了尽可能减小工程机器人的尺寸，第三版工程机器人采用独立悬挂，使用铝方的连接作为底盘的基本框架，在铝框上使用板材作为走线以及结构的安装平台。独立悬挂曾用于前几个赛季的英雄与步兵机器人，故不在此赘述。

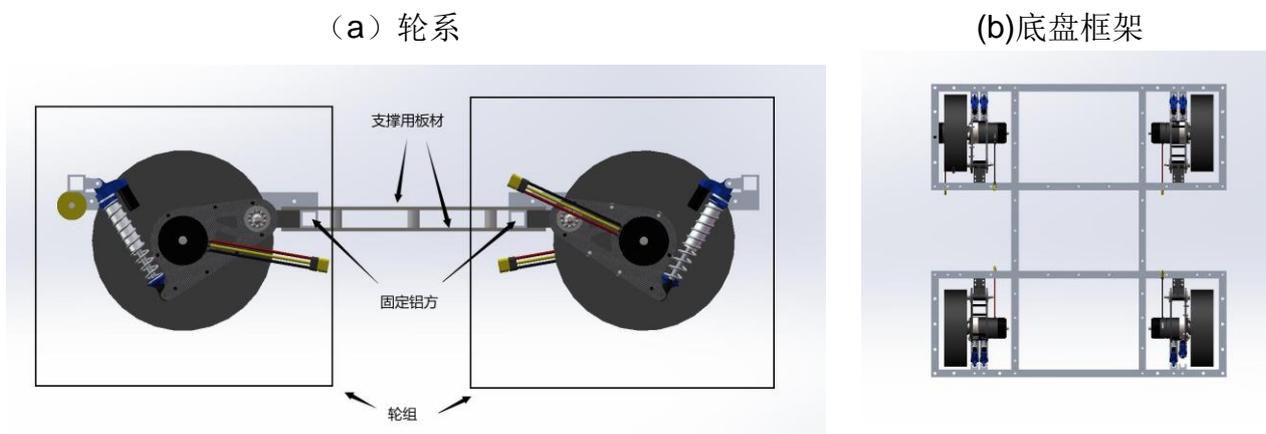
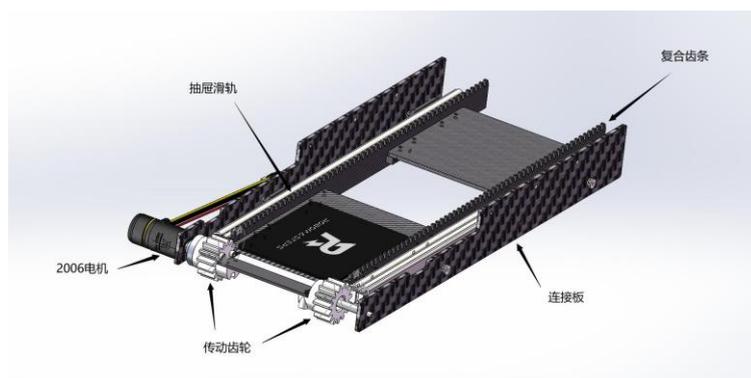


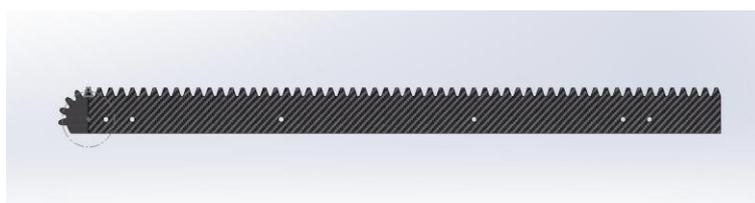
图 4.4.14 第三版工程底盘

## (2) 刷卡救援机构

使用复合齿轮机构以及摇臂实现将 2006 电机的转动转化为救援卡的下放和伸出，同时使用抽屉滑轨保证结构强度。两侧使用板材作为连接件，固定于底盘的铝方上，使用 3D 打印件作为齿轮，齿条则使用板材切割。齿轮和齿条的复合可以实现将救援卡先绕传动齿轮转动再向前伸出平动，转动时的自由度由六根摇臂保证，平动时的自由度由抽屉滑轨提供。



(a) 刷卡机构整体



(b) 复合齿条

图 4.4.15 刷卡救援机构

### (3) 钩爪救援机构

通过简单曲柄摇杆机构进行控制，舵机控制曲柄部分，摇臂部分负责钩取保护框。采用四连杆机构可以通过空间结构进行竖直方向的自锁，限制 z 向运动，在一定程度上避免脱钩。

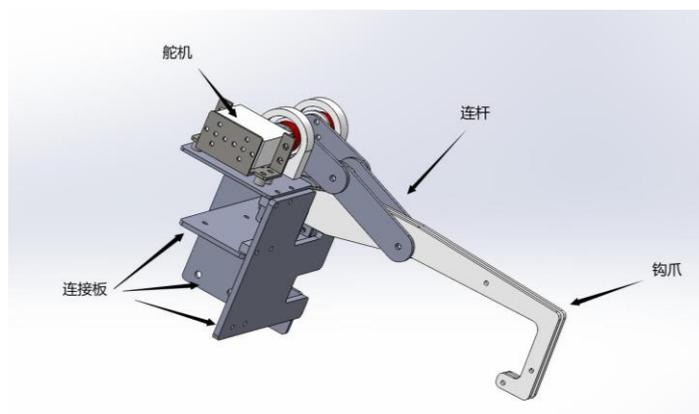
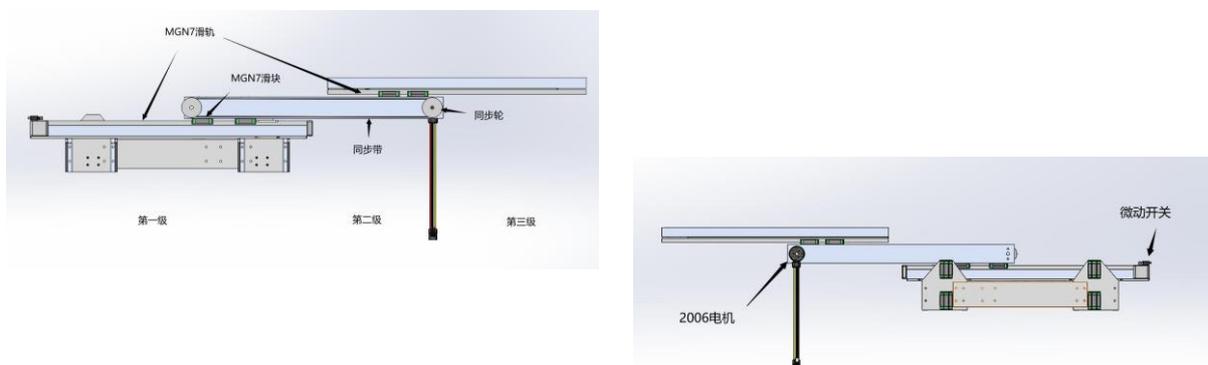


图 4.4.16 钩爪救援机构（左侧）

### (4) 三段式二级伸出机构

使用三段式二级伸出机构，应用于第三版工程上的抬升、前伸以及横移机构上，目的是保



证伸展长度的同时尽量减少初始尺寸。将动力部分置于第二级，工作部分置于第三级，各级之间由滑块滑轨进行连接，使用 MGN7 滑块及其配套滑轨，传动部分使用同步带进行传动。

(a) 内侧视图

(b) 外侧视图

图 4.4.17 前伸部分

### (5) 吸盘部分

使用同步轮同步带进行传动，使用两个 2006 电机控制吸盘的 pitch 运动，使用舵机控制吸盘的 roll 向转动。吸盘部分的左右平移则使用三段式二级伸出机构以减小整体的体积，主体部分则使用的是铝方通支撑，为了保证能够在前伸至最长的时候。

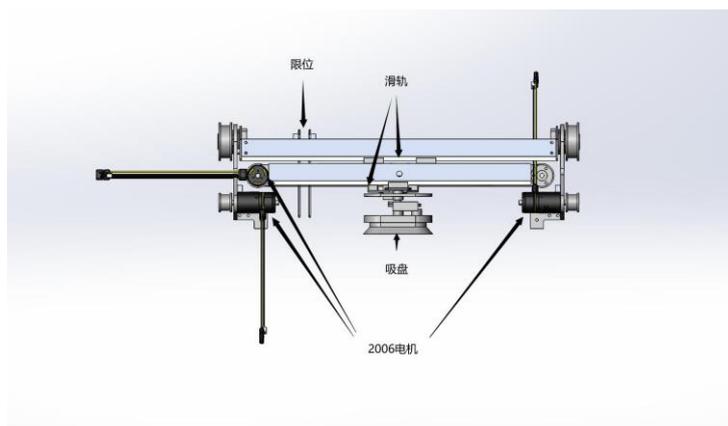


图 4.4.18 吸盘部分

### (6) 云台部分

最简单的二自由度云台，使用两个脉塔 5010 电机分别控制云台的 yaw 轴和 pitch 轴，将云台安装在抬升部分的尾部，保证能够在抬升至最高点时能够观察到资源岛上的矿石情况，与此同时能够在移动时观察到路况以及转矿时观察到矿石的情况。



图 4.4.19 云台部分

### (7) 转矿部分

使用两个滚筒进行转矿。通过控制滚筒的转动方向完成矿石的翻转功能，共有两个滚筒，每个滚筒使用一个 2006 电机控制转速和转向。为了保证接触面积尽可能大，采用将 2006 电机内置于 PVC 管的安装方式，安装件为了避免干涉，结构较为复杂，采用 3D 打印的方式制造。

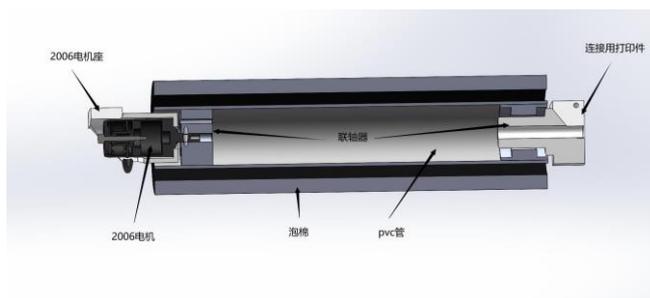


图 4.4.20 转矿部分

## 工艺选择

由于此版工程机器人的设计与制作均位于暑假期间，从画图至出车历时近一个月，轮系使用与前代英雄、步兵一样的通用轮系结构，使用铝方通作为抗弯抗扭的主承力结构，在铝方通的基础上使用玻纤板进行连接，玻纤板可以使用小型雕刻机进行加工，复杂结构件使用 3D 进行制造，故能够较快地对不合理结构及进行更改修正。

(1) 2D 雕刻：所有的板材采用，尽量避免出现复杂形状的结构，便于进行快速结构修正和迭代。

(2) 外包加工：主要针对铝方管，为了节约时间的同时保证加工精度，将铝方管外包进行加工。

(3) 3D 打印：对于承力小，形状复杂且尺寸较小的结构件，采用 3D 打印的方式进行制造。

## 传感器安装以及电路板固定方式

使用传感器主要有视觉识别摄像头 Intel RealSense、吸盘机构 pitch 轴陀螺仪模块、底盘陀螺仪、图传云台陀螺仪。摄像头使用英制螺丝进行固定于吸盘机构地相邻侧。吸盘机构陀螺仪使用尼龙铆钉以及 3M 胶固定在吸盘机构铝方上。除传感器外，另有四个用于校准初始机械位置的微动开关，使用胶枪粘连于伸出机构中间级的尾部以及抬升机构底部下方的底盘处。底盘陀螺仪和云台陀螺仪均为板载陀螺仪，分别随开发板使用螺丝固定于底盘和云台上。其他电路板的固定方式则使用 3M 胶，固定于底盘有空间的地方。

## 有限元分析以及简单力学分析

采用 Solidworks 中的 Simulation 模块。以抬升部分承力的铝方管进行分析，设定约束为四个螺栓固定面，载荷情况为 750N 的单向受力。可以得出如图所示的应力应变与位移分析。针对其应变与位移观察可以看出，均在小数点后两个量级，也就是认为在可以接受的范围内。对于前伸机构，采用滑轨均为 MGN7 滑轨，左右各一个，容许的静力矩为 4.8NM，基本额定动载荷为 1.68kN。一侧使用两个滑块，两侧均有滑轨，前伸最长为 500mm，矿石的重量假设为 800g，伸出接矿石的时候下落的高度为 200mm，假设受到冲击的时间为 0.1s，则等效的力的大小为 16N，前伸机构的整体重量取整为 10kg，重心位于第二级伸出的中间，距近处的滑块为 250mm，计算得静力矩为 2.7NM，动载荷取静载荷得 3 倍为 348N，且考虑实际状况时工况比计算载荷要低，可得该零件的强度已经很满足真实应用场景。

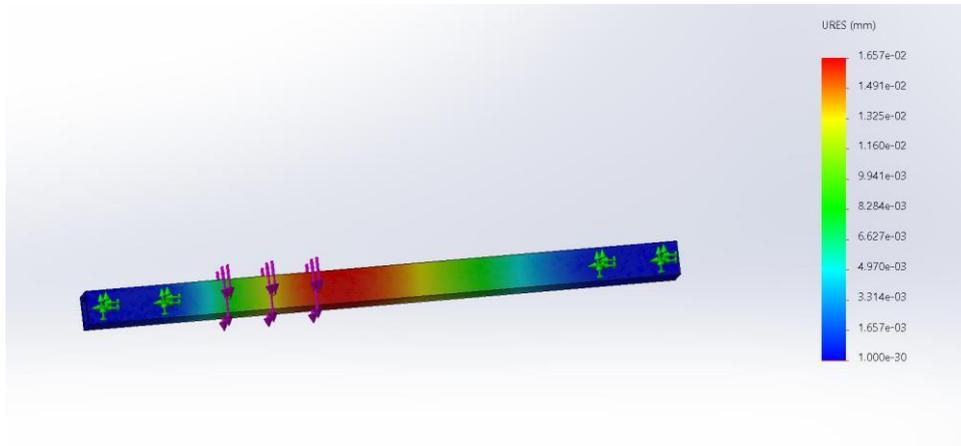
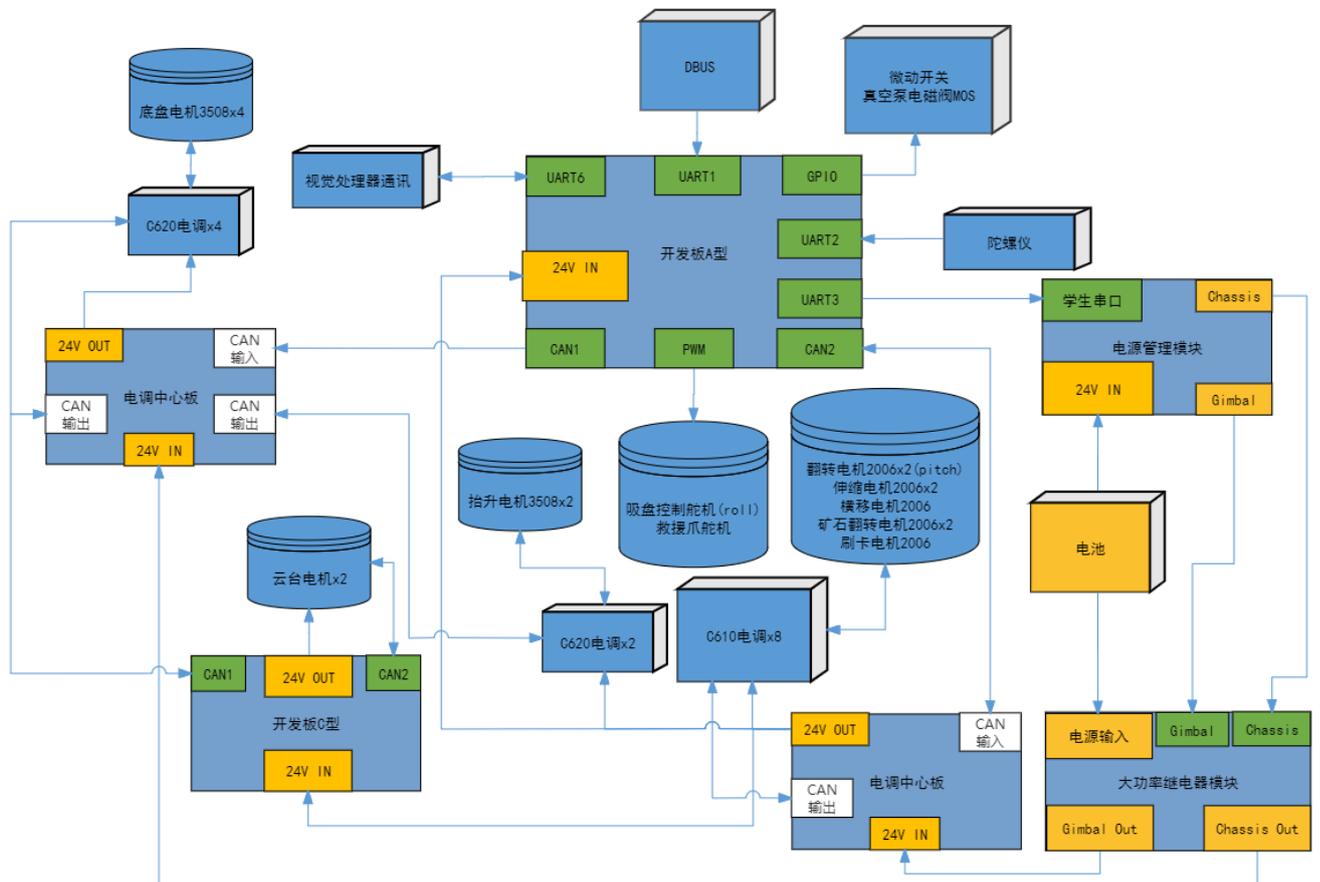


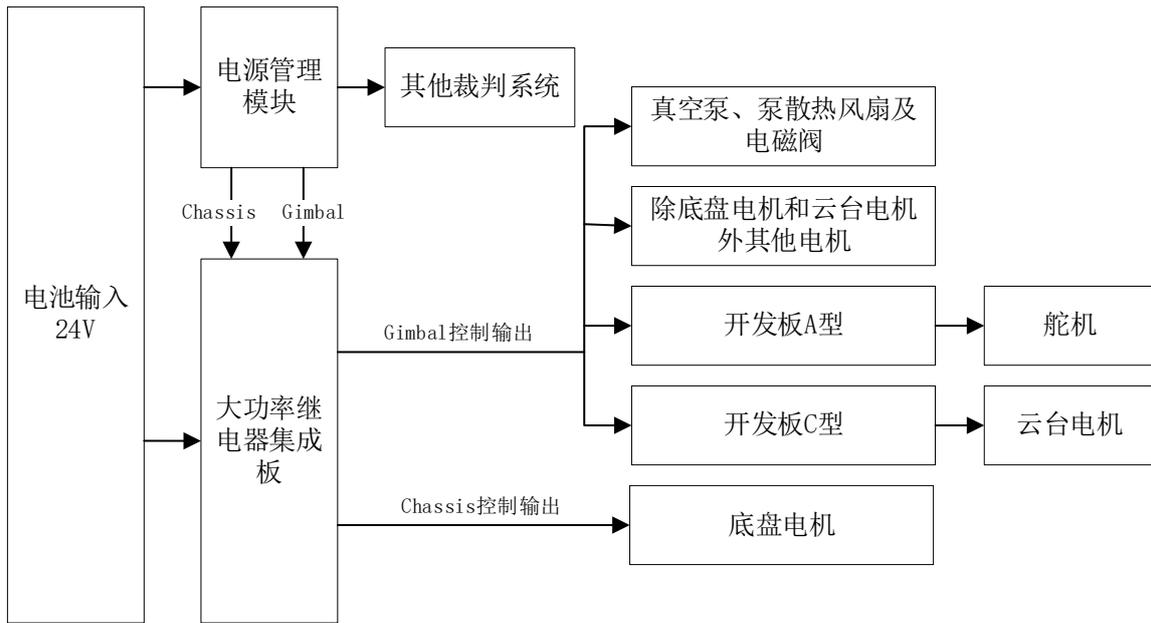
图 4.4.21 有限元分析例

## 4.4.2 硬件设计

### 4.4.2.1 整机硬件方案图

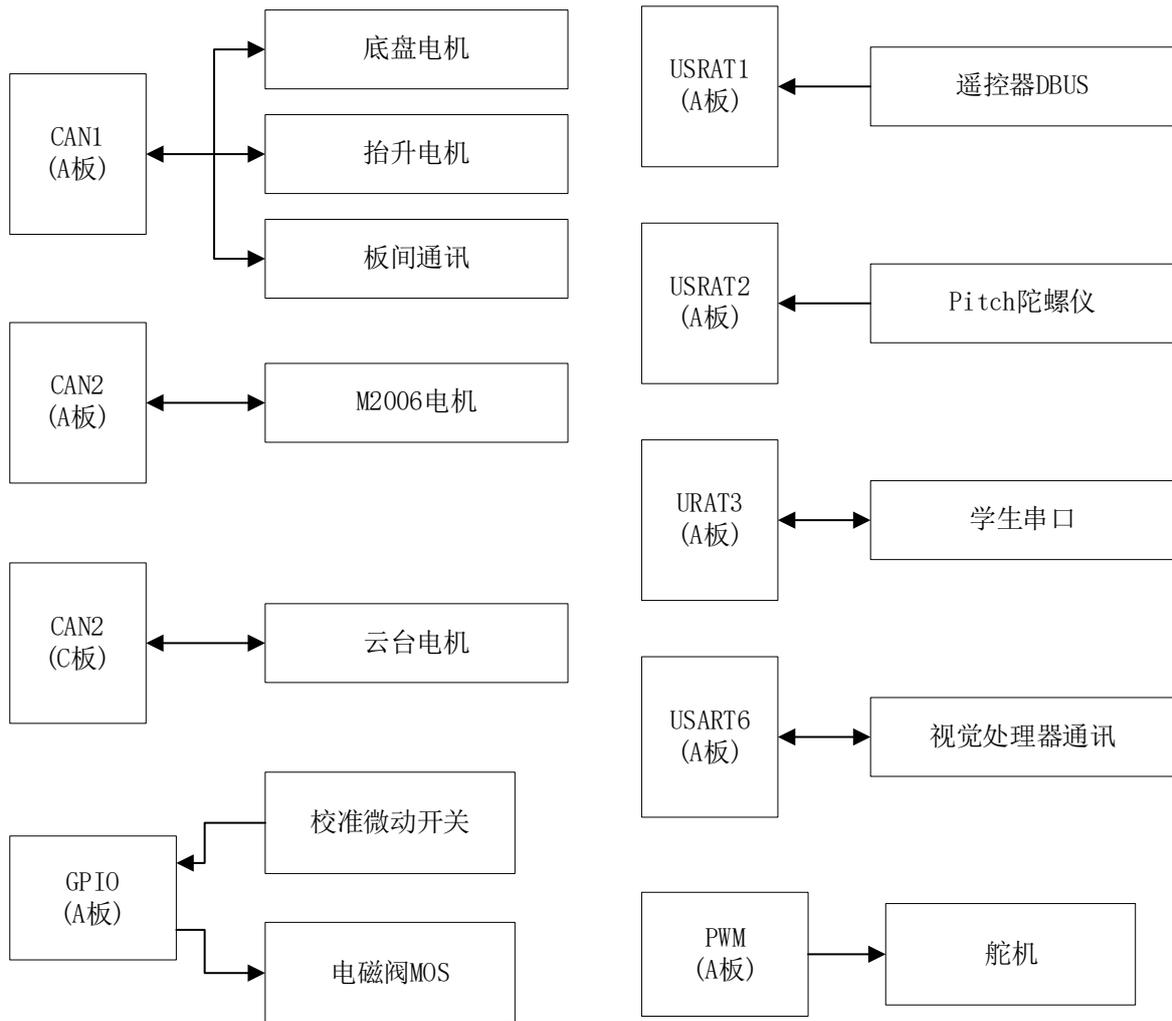


4.4.2.2 硬件整体框图



4.4.23 电源树

图 4.4.24 信号流向



### 4.4.2.2 自研硬件设计

由于工程机器人使用功率尤其是底盘功率较高，电源管理模块的输出无法满足功率需求，因此需要使用能过较大电流的继电器进而从电池直接供电，而使用单独的继电器线材连接较为繁琐，因此自行设计 PCB 将控制输入、电源输入和输出的 XT 接头以及大功率继电器集成到同一 PCB 板上，减小了线材和接头所占空间，使走线更加简洁美观。为确保模块能耐受大电流，进行大面积铺铜和开窗处理，保证了 2 倍以上的安全余量。将拨杆推到底，使底盘以最大功率工作 30 秒，以此来测试模块的安全性。经测试，该模块的安全余量达到要求。

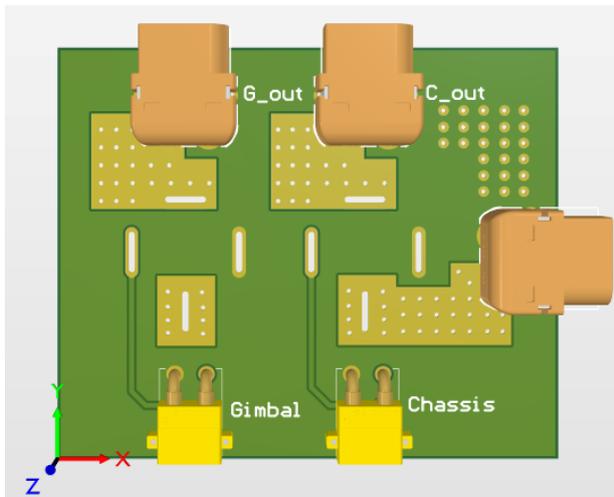


图 4.4.25 大功率继电器模块 PCB 图（正面）

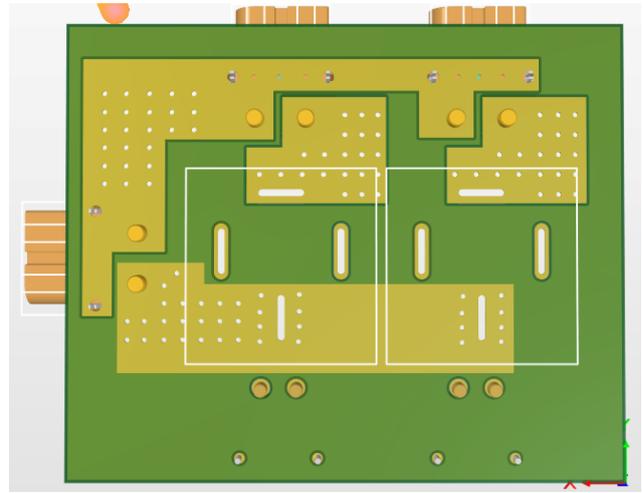


图 4.4.26 大功率继电器模块 PCB 图（反面）

### 4.4.2.3 关键器件选型

#### 控制板选型

由于取矿兑矿执行动作较为复杂，需要用到数量较多的执行电机、伺服舵机、传感器等，需要用到较多的接口，因此主控使用资源较为丰富的 Robomaster 开发板 A 型（以下简称 A 板）。云台部分需要精度较高的陀螺仪，A 板陀螺仪 MPU6500yaw 轴漂移较大不满足要求，同时云台电机同样为 CAN 通讯设备，因此选择集成高精度陀螺仪 BMI088 的 Robomaster 开发板 C 型（以下简称 C 板）作为云台控制器，并与 A 板通过 CAN 进行通讯。同时要使用十多个 CAN 通信的电机，需要多块电调中心板以满足电调 CAN 通信接口和电源接口以及其他电源的需要，因此选择具有较多接口且能承载较大电流同时带有保护外壳的电调中心板 2。

单板	设计需求	风险评估
主控制板	有丰富的接口和资源，能满足	资源丰富的同时若出问题维

单板	设计需求	风险评估
	功能需求	护的难度也会增大
电调中心板	扩展多个电源输出和 CAN 通讯接口	同时接通多个电调，需要能承受较大的电流，同时保证高速 CAN 通讯的稳定
云台控制板	集成高精度陀螺仪，有 CAN 通讯功能，同时能为云台电机供电	云台电机堵转时产生较大的电流可能会损坏控制板

### 大功率继电器

由于工程机器人不限功率，且重量较大，使用的电机较多，整车的功率较大，电源管理模块输出功率无法满足需求。因此需要大功率的继电器来控制供电。为了保证继电器有较大的安全余量，选择汽车级的大功率继电器 JD2912，经测试可以稳定工作。

### pitch 轴陀螺仪模块

为了保证取矿的稳定性，取矿机构俯仰角（pitch）陀螺仪仅起到校准的作用，校准完成后，pitch 角通过 pitch 电机 M2006 的传感器进行计算和控制，防止赛场出现陀螺仪失灵或损坏而导致无法正常取矿的情况发生，因此该处陀螺仪对精度并没有特别高的要求，最终选择正点原子的陀螺仪 IMU601 模块，该模块价格较低，且经测试静态精度能达到  $0.05^\circ$ ，具有较为成熟的算法以及上位机调试软件，降低了调试的难度。

## 4.4.3 软件设计

整个嵌入式系统软件部分采用 STM32CubeMX+MDK5 进行开发，即使用图形配置工具 STM32CubeMX 生成 STM32 硬件抽象层 API 及 MDK 工程，并在此基础上在 MDK 的开发环境中进行代码编写和调试。操作系统使用 freeRTOS，分成多个任务对机器人进行控制。采用双板控制，其中 A 板负责控制除图传云台外的其他所有任务，C 板仅根据当前从 A 板得到图传的目标角度对云台进行控制，A 板与 C 板通过 CAN 进行通讯，因此主要介绍 A 板的控制流程。A 板放置在底盘上，板载陀螺仪 MPU6500 提供底盘旋转角速度数据，C 板放置在图传云台上，板载陀螺仪 BMI088 提供 pitch 和 yaw 两轴数据。A 板控制逻辑流程图如下：

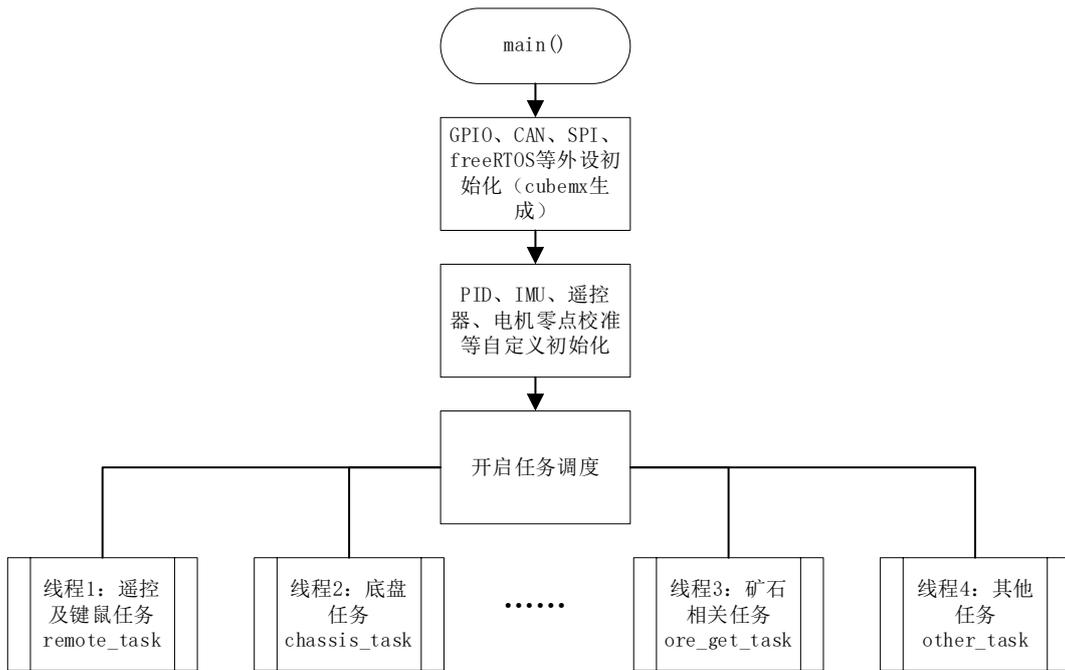


图 4.4.27 总任务调度

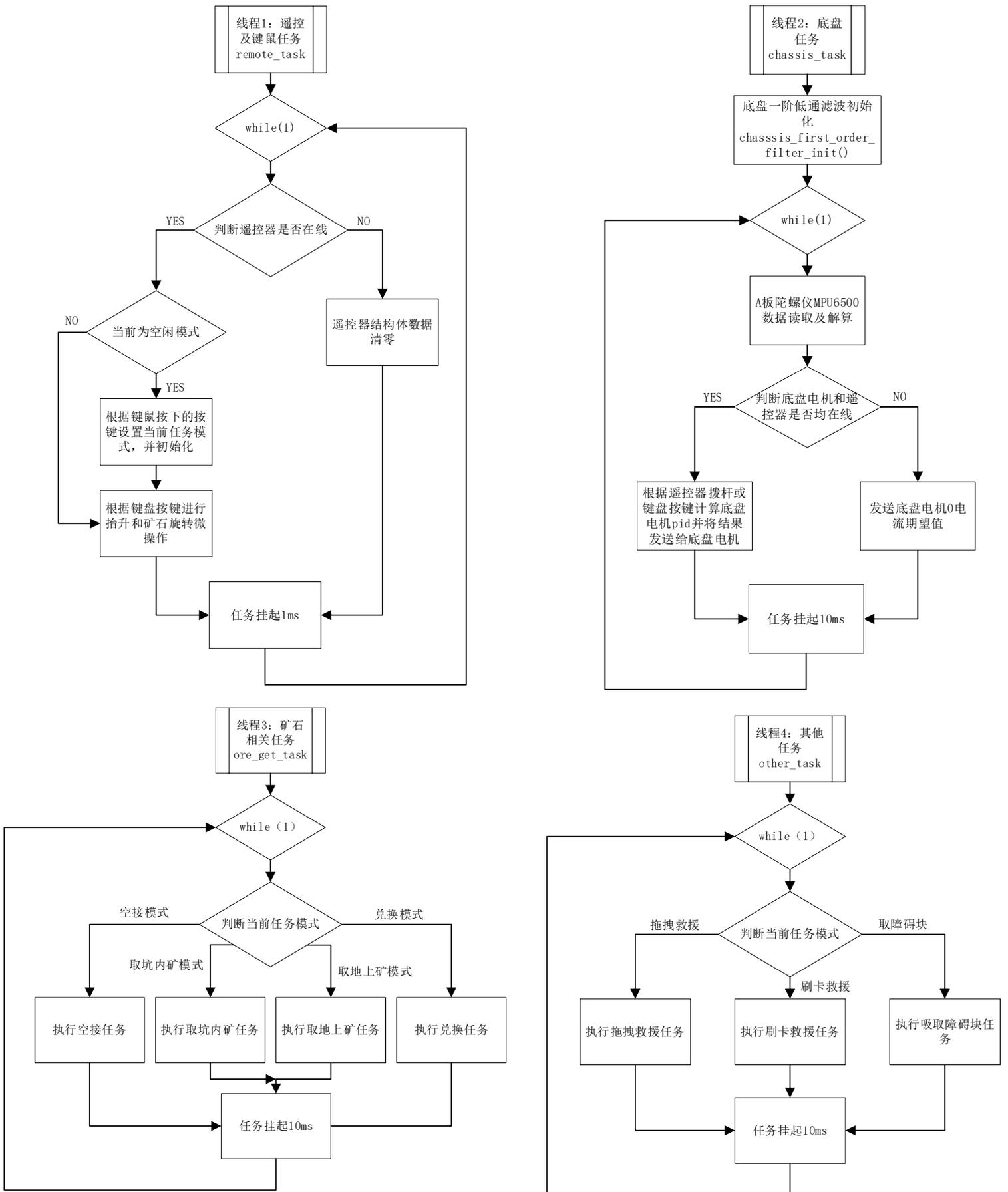


图 4.4.28 执行工程目标功能任务流程

除以上所示执行工程机器人要求功能的任务外，还有如下控制任务：

**detect\_task:**参考了 Robomaster 开发板 C 型开源程序中的错误检测函数，即通过接收数据时间来判断设备是否离线。该任务负责检测所有电机以及遥控器、陀螺仪等设备是否在线正常工作。

**PIDcal\_task:**负责计算除底盘电机外其他所有电机 pid 并将控制信号发送给电机。

**refree\_task:**裁判系统通讯任务，负责获取机器人的状态信息，并不断刷新 UI。

**port\_task:**负责向上位机通过串口发送数据进行调试。

**gimbal\_tx\_task:**负责与 C 板进行 CAN 通讯。

所有串口通讯如遥控机数据、pitch 陀螺仪、发送上位机数据以及与视觉处理器通讯均使用 DMA 并在串口中断中处理，减轻 MPU 负担。

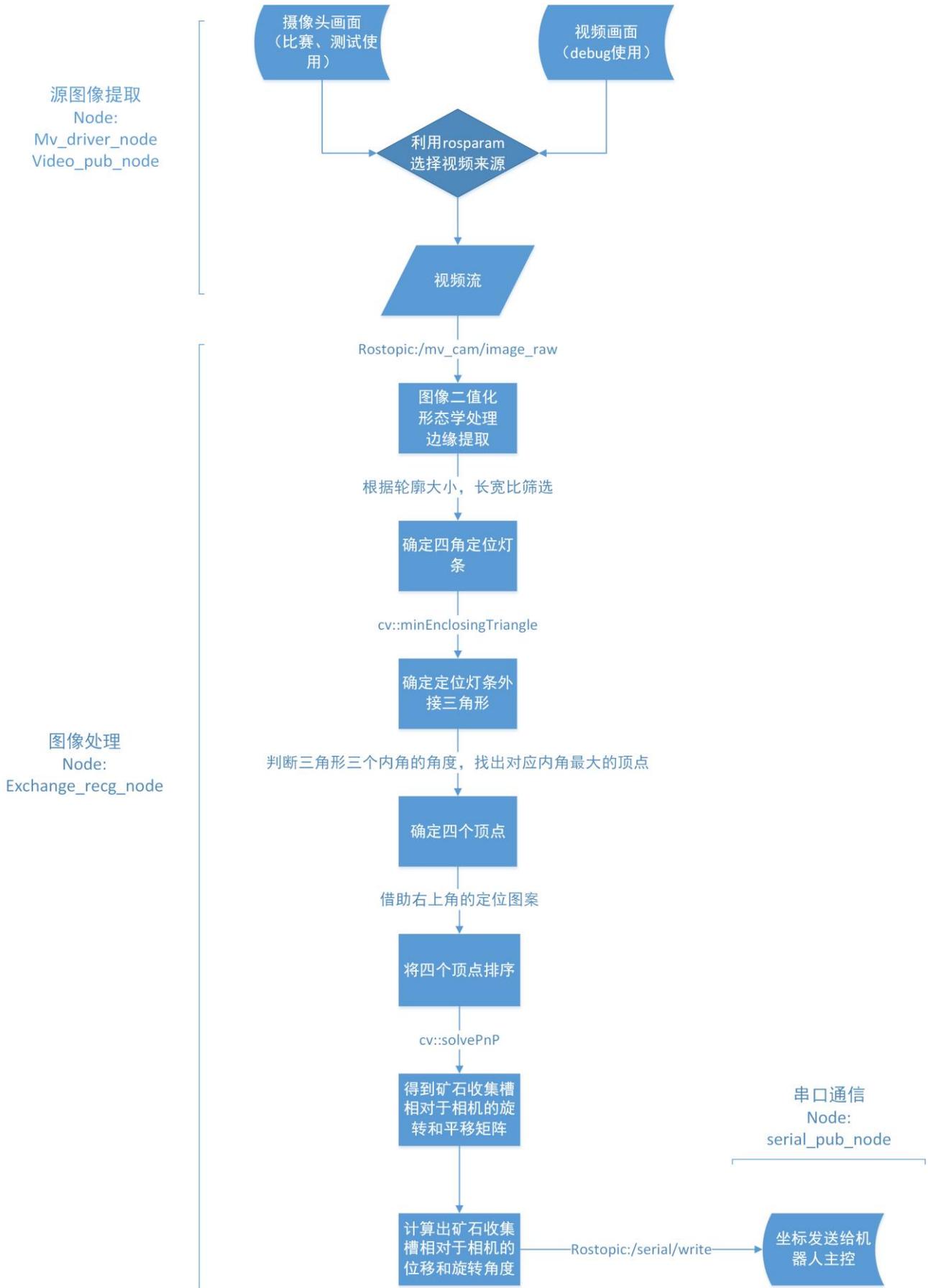
矿石吸取及兑换均采用“半一键执行”模式，即按下按键，可以通过判断当前电机的位置来自动执行下一步，但为了防止误操作、没有吸住矿石、位置没有对准以及场上其他情况的发生，自动执行的过程中设置了需要操作手确认的环节，即通过按鼠标左键和右键来确定是否继续执行或回到之前的步骤重新执行。矿石兑换设置了两种模式：通过视觉识别自动兑换模式和手动兑换模式。自动兑换模式通过视觉处理发送的三轴角度进行矿石自动兑换，且在兑换前操作手可以进行微调使矿石准确对准兑换槽。手动兑换模式需要操作手进入兑换后全程手动调整各轴位置使矿石对准兑换槽。

## 4.4.4 算法设计

### 4.4.4.1 算法简介与流程

由于国赛修改了矿石兑换站形态，引入了更多的自由度。为了快速、稳定地完成兑换，与完整形态技术方案相比，工程机器人引入了视觉识别。视觉识别的主要功能是识别并计算矿石收集槽相对于工程车的位移和旋转角度，自动完成矿石对准矿石收集槽的任务。

算法流程如下：



#### 4.4.4.2 重要算法原理阐述、公式推导

算法的总体思路是找到矿石收集槽的四个顶点，使用 `solvePnP` 即可求得矿石收集槽相对于相机的位姿，但考虑到兑换站可能旋转一定角度产生透射变换，无法直接使用 `minAreaRect` 来求得四个顶点。

故利用 `OpenCV` 的函数 `minEnclosingTriangle()` 为每个定位灯条的轮廓寻找最小外接三角形，此函数返回一个包括三角形三个顶点坐标的数组。

为了确定三个顶点中位于直角上的顶点，采用的方法是计算三角形三条边之间的夹角，夹角最大的角的顶点即为所需的点。

`solvePnP` 原理在前已有详述，此处不再重复。利用四点坐标可求得矿石收集槽相对于相机的平移和旋转向量。平移向量即为矿石收集槽相对于相机的位移，旋转向量可以利用 `Rodrigues()` 函数将其转换为旋转矩阵  $\mathbf{R}$ ，则有

$$\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{V}$$

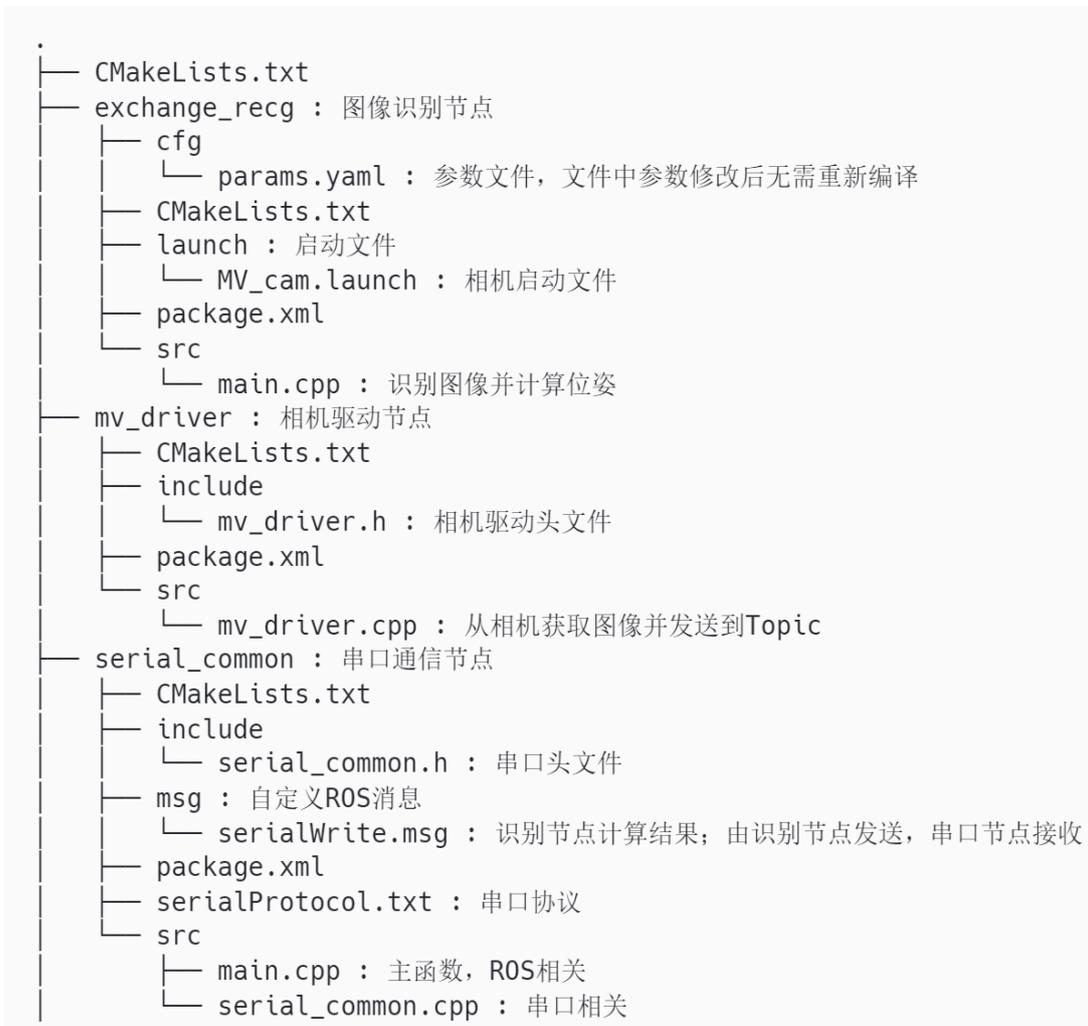
其中  $\mathbf{V}$  是矿石收集槽在相机坐标系下的法向量，由此可求出矿石收集槽的欧拉角。

#### 4.4.4.3 优化方向

按照目前的设计，相机固定于工程机架上，测定出位姿后靠编码器确定夹爪的位姿，进而控制夹爪的移动，将矿石填入矿石收集槽。未来的优化方向包括：

1. 采用更为准确的方式确定夹爪位姿，如在夹爪上粘贴 `AprilTag` 进行识别
2. 换用机械臂后，可以采用机械臂开源控制库，在 `miniPC` 上完成对机械臂的控制。

#### 4.4.4.4 程序结构介绍



#### 4.4.5 UI 设计

UI 根据工程机器人当前的任务状态，大致分为两种情况：当机器人执行取矿或兑换相关任务时，界面出现当前的任务模式、机器中心准线以及矿石定位线；当机器人没有执行矿石相关任务时，界面仅出现任务模式提示，使操作手视野清晰。

执行矿石相关任务时，UI 界面分为三个部分：当前任务模式、矿石定位线和机器人中心线，如下图所示：

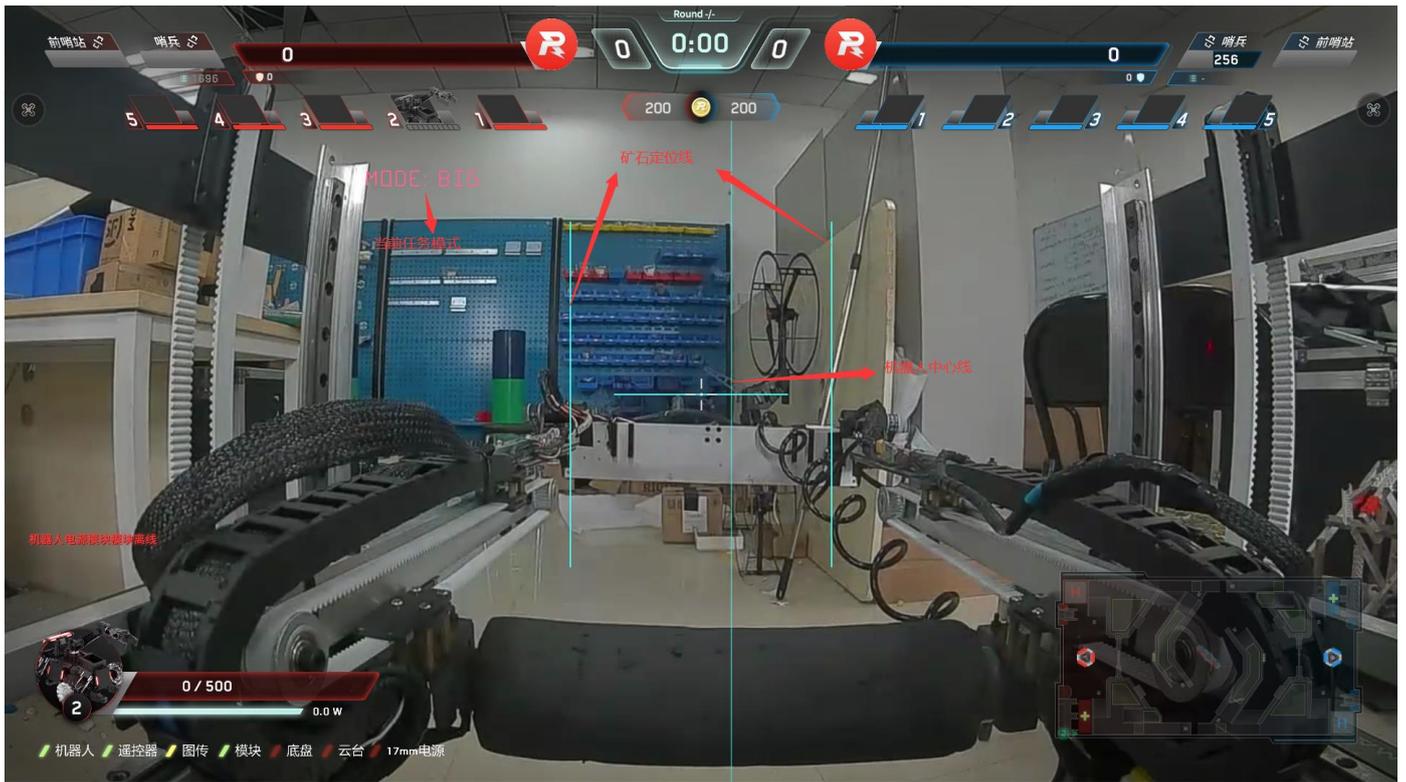


图 4.4.29 矿石相关任务 UI 界面

执行其他任务或空闲时，UI 界面如下图所示：

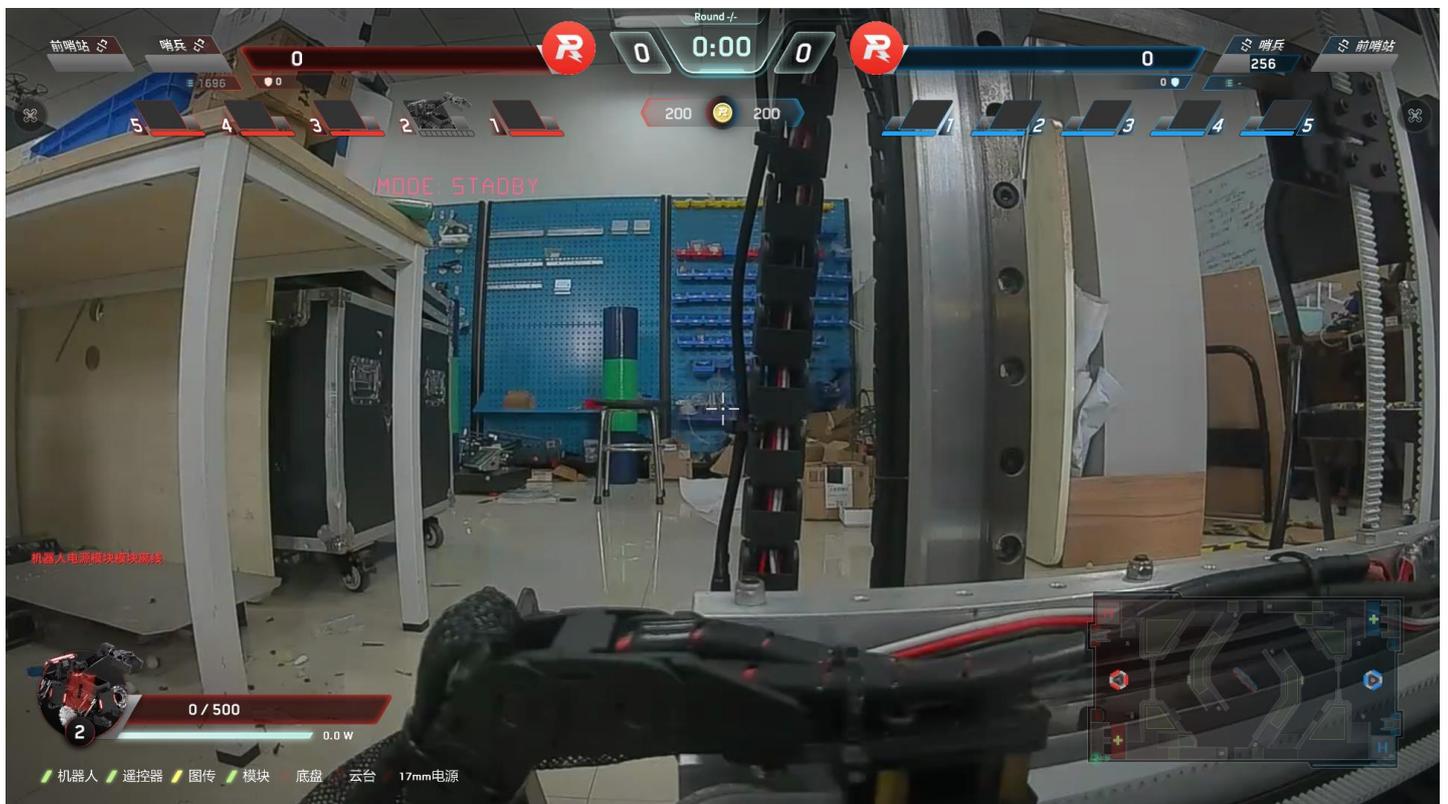


图 4.4.30 其他时刻 UI 界面

## 4.5 研发迭代过程（10）

### 4.5.1 版本迭代过程记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	初步实现既定功能	2022.3.18
V1.1	采用气动作为夹爪及前伸动力	2022.4.2
V1.2	吸盘取矿以及五自由度兑矿，救援，拿取障碍块。	2022.7.30

### 4.5.2 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	机器人版本号或阶段	解决人员
1	抬升4个电机不同步	机械装配时，物理限位的位置不同，导致阻力不同	重新调整了各个物理限位的位置	V1.0	机械工程师：贺雷 硬件工程师：李雯 嵌入式软件工程师：李雯
2	气动气爪的响应速度不能空接矿石	两个两位三通电磁阀同时上下电，导致进气一侧与排气一侧在某一瞬时是相同压力，导致响应速度慢	先让排气一侧下至于低电平联通大气压，形成了两侧最大的压差	V1.1	机械工程师：黎柯佚 商清影 硬件工程师：庄宝伟 嵌入式软件工程师：李雯

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
3	刷卡救援 RFID 卡无法被识别	作为垫板的碳纤维是导体,	将垫板换为了木板材质	V1.1	机械工程师: 黎柯佚 硬件工程师: 庄宝伟 嵌入式软件工程师: 李雯
4	使用真空泵时存在较大的震动, 传递到整车。	由于刚性连接, 并无减震措施, 同时真空泵震动也较大	使用橡皮筋进行捆绑安装, 多次调整知道找到最佳状态	V1.2	机械工程师: 贺游 硬件工程师: 李雯 嵌入式软件工程师: 李雯
5	在抬升机构收缩至最小时与前装甲板干涉	前装甲板由于刚性不足而加强之后高度变化	通过电控设置高度使吸盘回转	V1.2	机械工程师: 贺游 硬件工程师: 李雯 嵌入式软件工程师: 李雯
6	钩爪救援时易过起伏路段时容易脱钩	钩爪直接连接在舵机上, 传力路径不合理	使用四连杆机构设置死点避免脱钩	V1.2	机械工程师: 黎柯佚 硬件工程师: 李雯 嵌入式软件工程师: 李雯

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
7	A板上电但电机未上电时若拨动拨杆，电机上电后会疯车	A板不断发送PID计算出得控制信号给电机，导致上电后疯车	加入设备离线检测任务，当电机离线时发送0电流期望值，解决了疯车问题	V1.3	嵌入式软件工程师：李雯

## 4.6 团队成员贡献（2.5）

姓名	基本信息		主要负责工作内容描述	贡献度
	（专业、年级、队内角色）			（所有成员贡献度合计为100%）
黎柯 佚	机械工程	大二 机械组成员	负责第二版机械的结构审核和设计及后期的拼装改进	15%
商清 影	飞行器制造工程	大二 机械组成员	负责第二版机械中底盘结构的设计各种小问题的优化	10%
廖彬 璇	应用化学	大二 机械组成员	负责查阅开源资料，提供设计思路，以及刷卡救援的设计及改进	5%
庄宝 伟	软件工程	大二 电控组成员	负责辅助机器人的嵌入式开发，编写一键实现既定功能的快捷程序，检查程序漏洞	15%

基本信息		主要负责工作内容描述	贡献度
姓名	(专业、年级、队内角色)		(所有成员贡献度合计为 100%)
李雯	自动化 大四 电控组成员	负责整个机器人的嵌入式开发, 包括底盘控制、云台控制、视觉系统的嵌入式环境开发等	15%
贺雷	飞行器制造工程 大四 机械组成员	负责安排机械组任务, 以及第一版机器人的机械结构设计以及装配维修。	10%
贺游	飞行器设计工程 大四 机械组成员	负责第三版机器人的机械结构设计以及装配维修。	15%
练桥	飞行器制造工程 大四 机械组成员	负责第三版机器人的机械结构设计以及装配维修。	15%

## 4.7 参考文献 (2.5)

- 1、RoboMaster 开发板 C 例程: <https://github.com/RoboMaster/Development-Board-C-Examples>
- 2、RoboMaster 开发板 A 例程: <https://github.com/RoboMaster/DevelopmentBoard-Examples>
- 3、RM2021-东北大学-T-DT 战队-工程机器人-机械结构开源
- 4、RM2021-桂林电子科技大学 Evolution 战队工程机器人机械结构开源

## 5. 飞镖系统

### 5.1 其它学校飞镖系统分析综述（2.5）

现阶段其他学校飞镖系统中的飞镖主要以无控或者只进行姿态稳定控制为主，但是也有使用 OPENMV、FPGA 等视觉识别系统进行制导的苗头。无控或者只进行姿态稳定控制的飞镖，优点是成本低，易更换维护，缺点是对发射系统的定位能力要求高，容易出现一次中全部中，一次不中全部不中的场面。

现有发射架主要有两种思路，摩擦轮发射和弹射，飞镖的最大尺寸提高，在有宽大弹翼的情况下摩擦轮结构布置出现困难，若要实现末端制导，摩擦轮发射的冲击可能影响陀螺仪等器件的工作，现有许多学校采用的摩擦轮发射方式有这两点局限，而弹射类型的发射架有更多问题，使用皮筋会老化，单滑轨装弹复杂，拉伸储能速度慢，多轨道体积重量大，结构复杂，初速度难以达标等等。

### 5.2 飞镖系统功能定义（5）

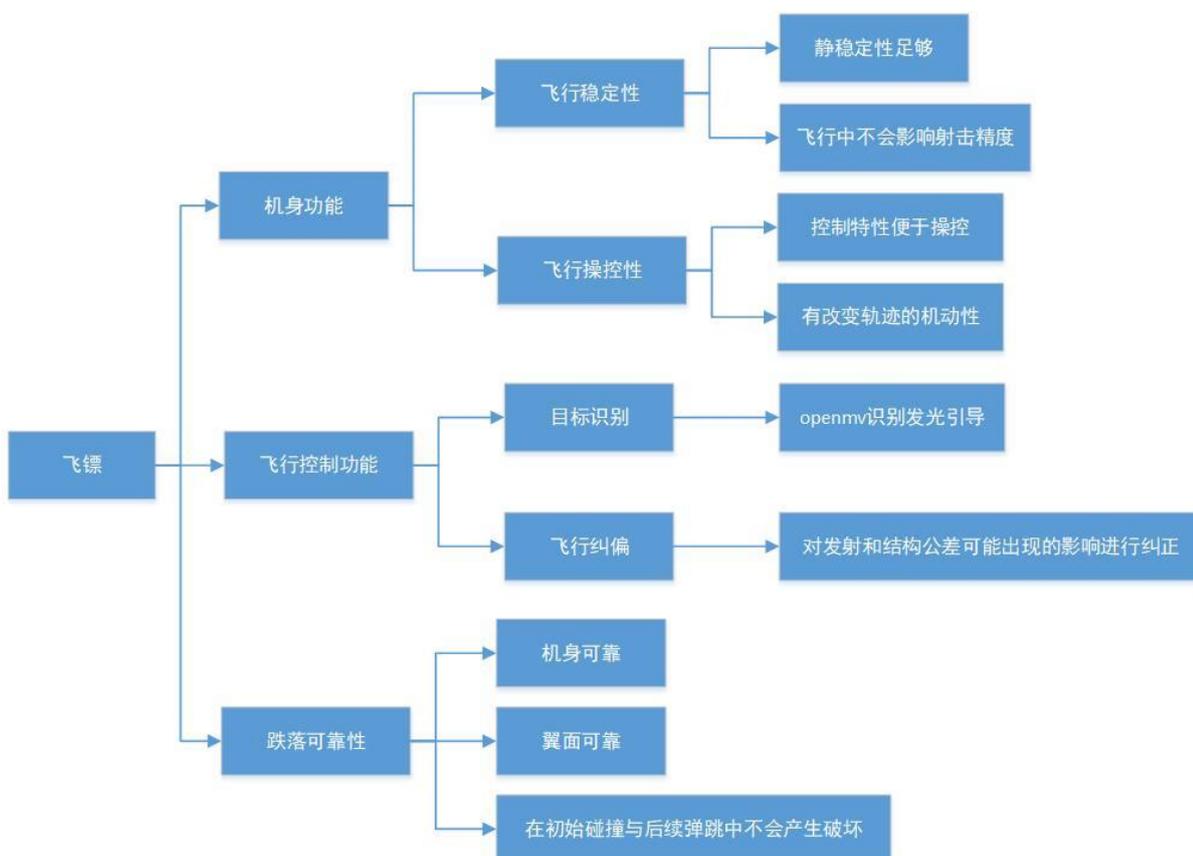


图 6.2.1 飞镖功能定义



图 6.2.2 飞镖发射架功能定义

## 5.3 飞镖系统核心参数（2.5）

飞镖系统发射架采用 Robomaster A 型开发板为主控板。作为击发装置，飞镖系统发射架使用 2 个 M3508 电机、2 个 M2006 电机和 5 个 MG996R 舵机完成飞镖的装填、发射结构的蓄能和发射机构的击发。为了实现对于前哨站和基地的分别打击，使用一个 GM6020 电机完成发射架 YAW 轴的调节。另外还有 4 个按键完成飞镖发射架工作模式的设置，用于场上定位和调试。整个飞镖发射架系统的工作电压为 24V。

飞镖系统飞镖标体使用 STM32H750 作为视觉处理器，STM32F401 作为姿态控制系统。同时搭载 MPU6050 六轴 IMU 单元、OV7225 摄像头 MPU-6050 的角速度全格感测范围为  $\pm 250$ 、 $\pm 500$ 、 $\pm 1000$  与  $\pm 2000^\circ / \text{sec}$  (dps)，加速器全格感测范围为  $\pm 2g$ 、 $\pm 4g$  与  $\pm 8g$  与  $\pm 16g$ 。透过 400kHz 的 IIC 和 STM32F401 进行通信，并通过卡尔曼滤波后完成飞镖的姿态

解算。OV7725 是一款分辨率为 30 万像素的 CMOS 摄像头，通过视觉处理器完成目标的查找，并完成飞镖的末端制导。飞镖的工作电压为 7.4V。

## 5.4 设计方案（25）

### 5.4.1 机械结构设计

#### 5.4.1.1 发射架总览

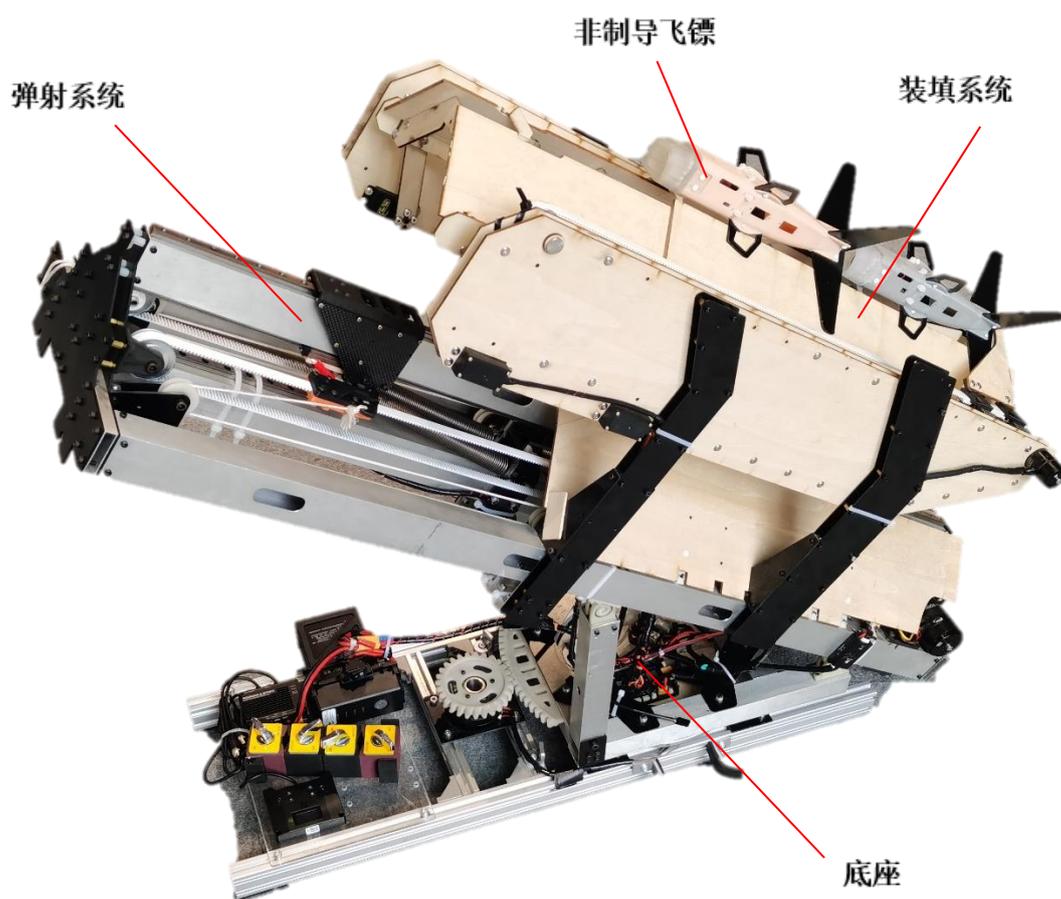


图 6.4.1.1 飞镖系统整体

表 6.4.1.1 主要参数

名称	参数
Pitch 范围	不可调 35°

Yaw 范围	$\pm 15^\circ$
发射架质量/kg	15
长宽高/mm	930*276*810
15s 最大射频	2
发射初速度 m/s	连续可调

## 5.4.1.2 弹射系统

### 5.4.1.2.1 滑台

滑台会与飞镖共同加速，其质量会影响飞镖的加速，所以要控制滑台的重量；并且滑台要在较短的距离快速减速停止，有着较大加速度，滑台又需要足够的强度；为了避免发射及装弹过程中弹翼与滑台之间的干涉，滑台需要一定的高度，将飞镖与驱动组件隔开，使得滑台有较大的尺寸。这三个因素对结构设计有着较高的要求。

综合考虑后滑台采用尼龙 3d 打印与碳纤维板材复合。

使用 mgn9 滑块作为滑台导向，在驱动过程中会产生一个弯矩，如图 6.4.1.2(a)所示，第一版设计中认为这个弯矩会对滑台产生影响，因此设计了两个前后错位的滑台进行抵消，在后来的测试中发现减少一个滑台并不会较大影响，减少的滑台重量还带来许多好处，经过测试没有出现问题。

使用 4mm 碳纤镂空板将拉力传递至上方，利用其较高的刚度，进行击发和上膛的控制，并且承受减速冲击。

将两块较薄的碳板分隔一段距离，中间填充打印件用螺栓贯穿固定。在纵向，发射飞镖时产生的弯矩和剪力由碳板承担，传递到下方；在横向，可大幅增加惯性矩，增加刚度提高稳定性，打印件承受较小的横向弯曲剪力，碳板承受正应力。使 0.5mm 厚碳板承受主要纵向及横向载荷，对打印件进行镂空减重。透视图如图 6.4.1.2(b)所示，打印件上部及左右大幅镂空，与飞镖接触部分为了避免发射时的刮蹭，镂空没有完全穿透。

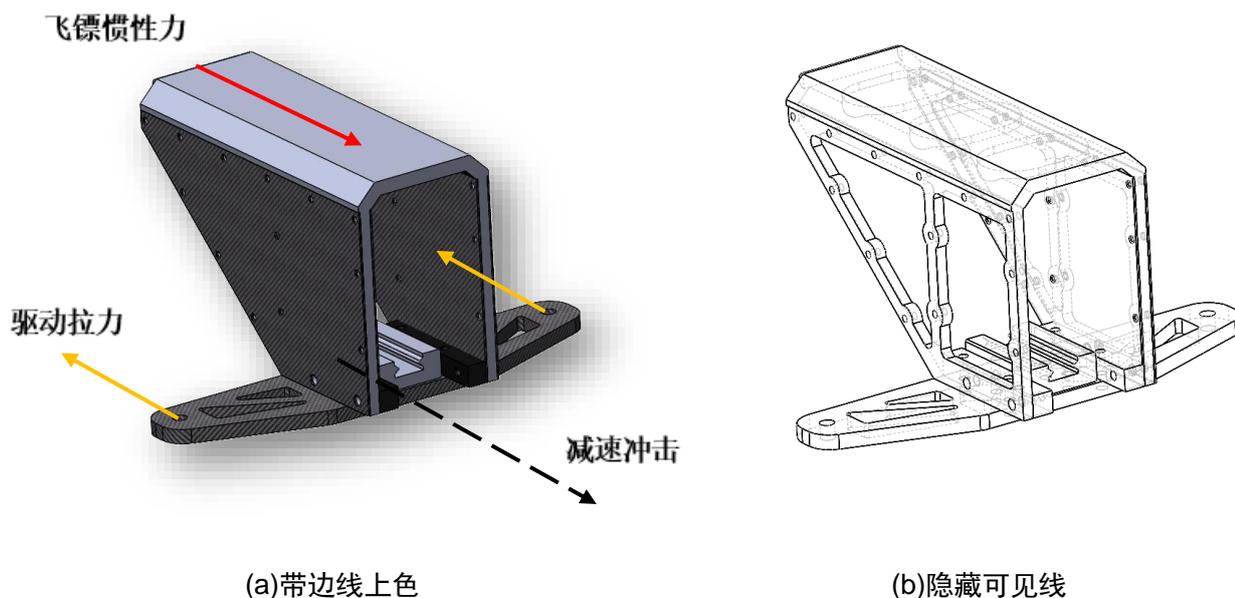


图 6.4.1.2 滑台示意图

### 5.4.1.2.2 储能

飞镖发射架采用拉簧与动滑轮组组合方式弹射发射。由于钢制拉簧有着较大的质量，在收缩过程中加速会消耗大量能量；制导飞镖的硬件对最大加速度有限制，需要使加速平缓，使用较长的加速行程；拉簧收缩行程和机械结构产生矛盾。为了解决上述问题，使用四级动滑轮连接拉簧，结构简图如图 6.4.1.3 所示

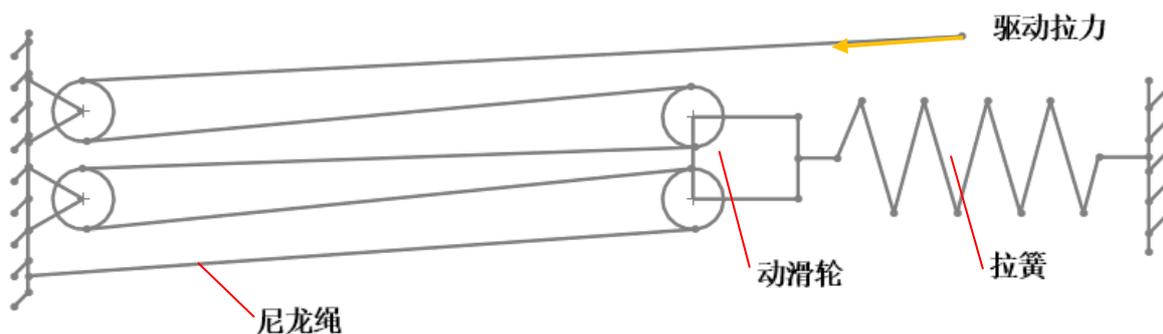


图 6.4.1.3 储能机构简图

其中为了确保稳定性，动滑轮采用 mgn9 滑块进行导向。 尼龙绳

对弹簧进行选型，弹簧的计算公式为：

$$k = \frac{Gd^4}{8nD^3}$$

$$\tau = \frac{8kDP}{\pi d^3}$$

其中  $d$  为线径， $D$  为弹簧中径， $P$  为轴向载荷， $G$  为剪切模量， $n$  为工作圈数， $\tau$  为切应力。

采用能量法计算飞镖获得的速度,其中弹簧储存的能量为:

$$E = 0.5(2 * F_0 + L * k) * L$$

其中  $F_0$  为拉簧初始张力， $L$  为发射行程， $k$  为弹性模量

飞镖获得的速度为:

$$V = \sqrt{2 / (m_{huatai} + m + 1/16 * m_{quarterV}) * (E - f * L)};$$

编写 matlab 程序进行计算可快速迭代。改变不同参数以获得不同的发射初速度。但由于加工方式等不同，导致拉簧初张力难以得出，算出合适弹簧参数时还需要实验验证，第一次采购的弹簧便因为初始张力大于设计时拉伸一段距离的工作张力，导致发射速度过快，更换弹性系数更小的弹簧。

为确保储能的重复性，增加力的反馈，使用拉力传感器进行初始张力测量，安装位置如图 6.4.1.4 所示。通过测得的拉力、各部分质量与发射的初速度可对未知参数进行确定，加快迭代的速度，便于确定不同的参数对初速度的影响，对于落点标定有易。

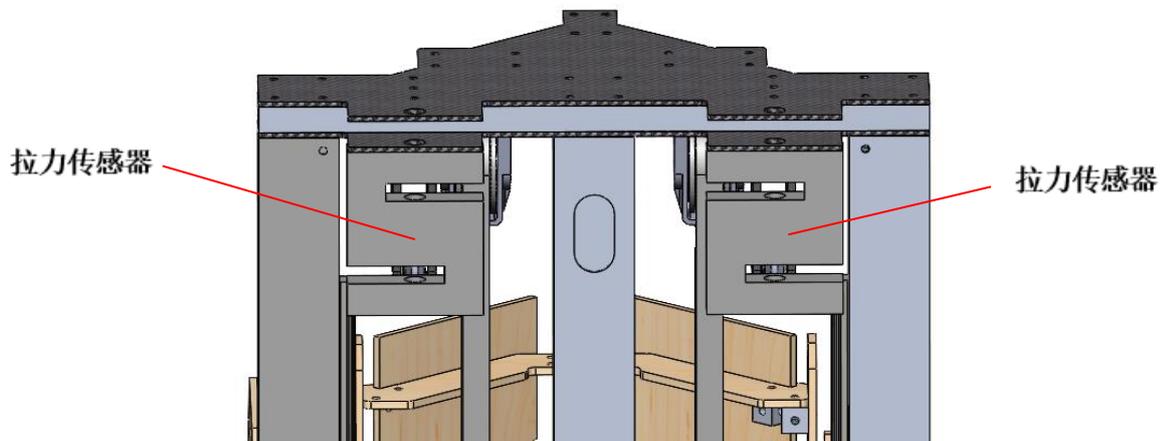


图 6.4.1.4 拉力传感器安装位置

### 5.4.1.2.3 拉紧与击发

拉紧通过两个 3508 电机带动同步带，将滑台拉回扳机，再返回最前端等候发射，使用打印件接驳同步带，并且带动滑台向后运动至扳机处，如图 6.4.1.5 所示。

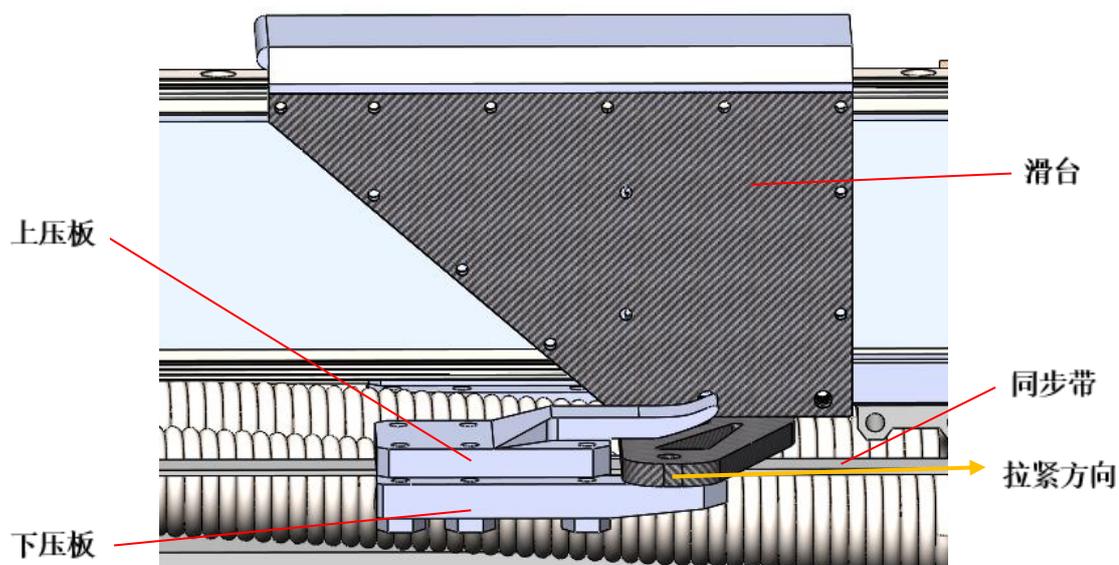


图 6.4.1.5 滑台拉紧示意图

发射机构的扳机部分为整体可运动的，通过调节发射位置改变加速行程来调节初速度，使用丝杆控制前后；用 mgn9 滑块对整体进行左右上下限位，以便提高丝杆抗压的稳定性，不发生失稳；使用推力轴承对丝杆进行前后限位，以保证在上膛有压力时仍然可以进行初位置调节；使用弹性联轴器与 2006 电机相连，避免同轴度不高的影响；使用舵机在下方进行击发，通过柔性绳索连接扳机，扳机通过一个小型拉簧与 mgn9 滑块相连，保持拉紧状态，可以在拉紧滑台时不需要主动操作便能卡上扳机；扳机与轴承接触的地方采用一对轴承接触，减少摩擦力，并且将 4mm 碳板与轴承接触的地方打磨出斜面，降低击发所需力矩。

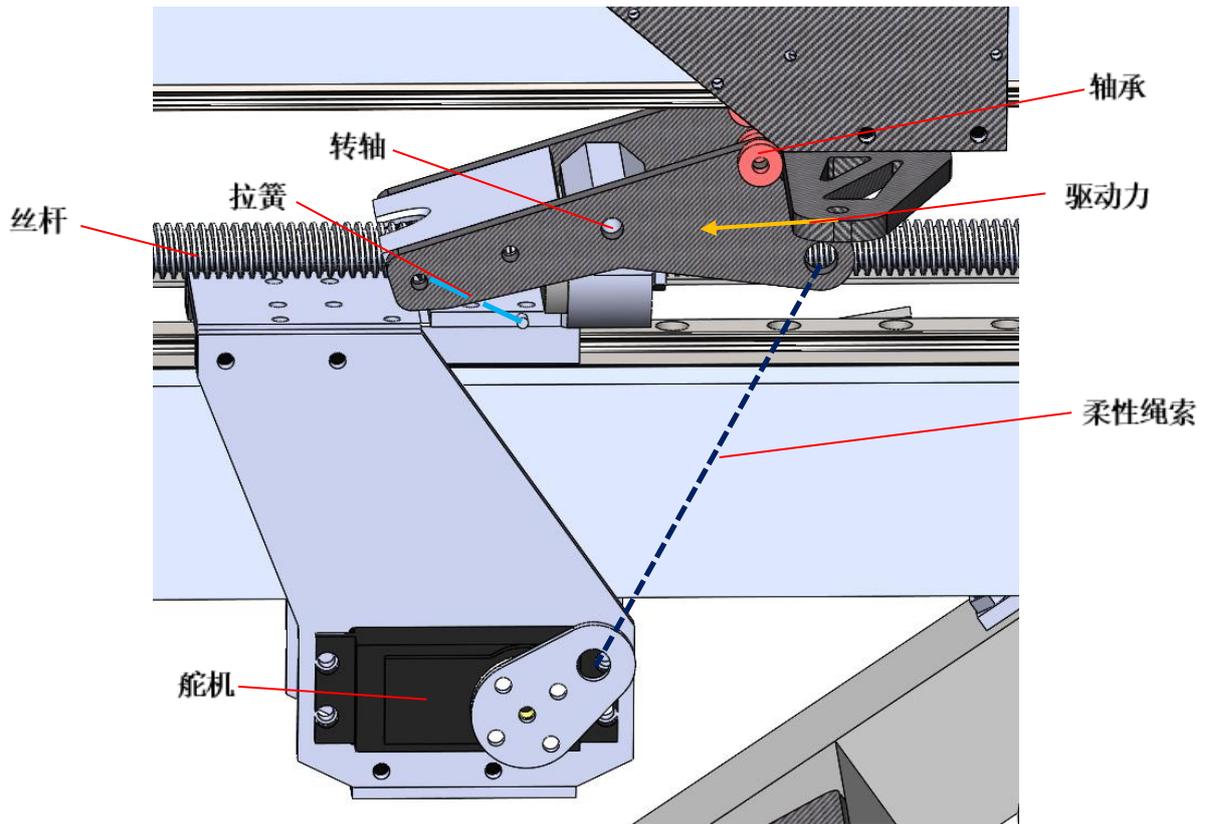


图 6.4.1.6 扳机示意图

由于尺寸限制，只能使用单个舵机进行击发，势必会对舵机安装处和扳机产生一个扭矩，这个扭矩造成的形变若过大，会无法击发，不幸的是 mgr9 滑轨由于极惯性矩较小抗扭刚度小，在铝方管设计之初未考虑到设计安装孔余量，后来增加抗扭设计，如图 6.4.1.7 所示为抗扭设计。击发时，在柔性绳索两端会产生较大拉力，力由舵机经安装的板材传递至滑轨，整体会产生扭转变形，舵机连杆至扳机的距离变近，使得无法完成击发，增加一块垫块后，在前后移动扳机时不会与铝方管发生摩擦，击发时产生变形垫块与铝方管接触，将扭力转化为一对力偶传递至铝方管，铝方管的抗扭刚度较大，使得击发时的变形大幅减小。

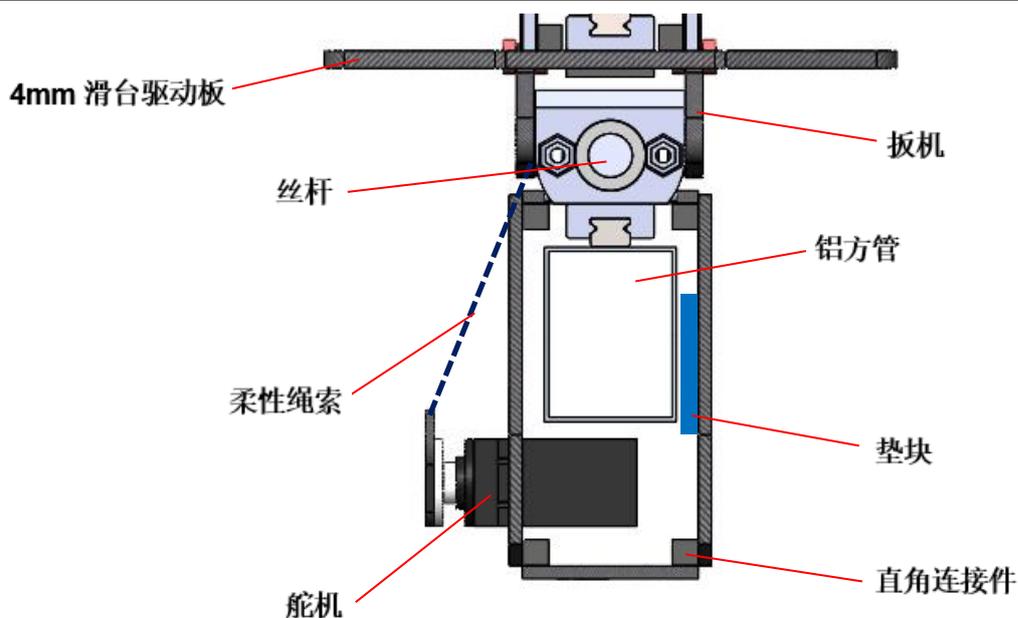


图 6.4.1.7 抗扭设计

拉紧和发射过程需避开其他结构避免干涉，飞镖、滑台、上膛、击发等结构都需要协调布置。最终前视效果如图 6.4.1.8 所示。

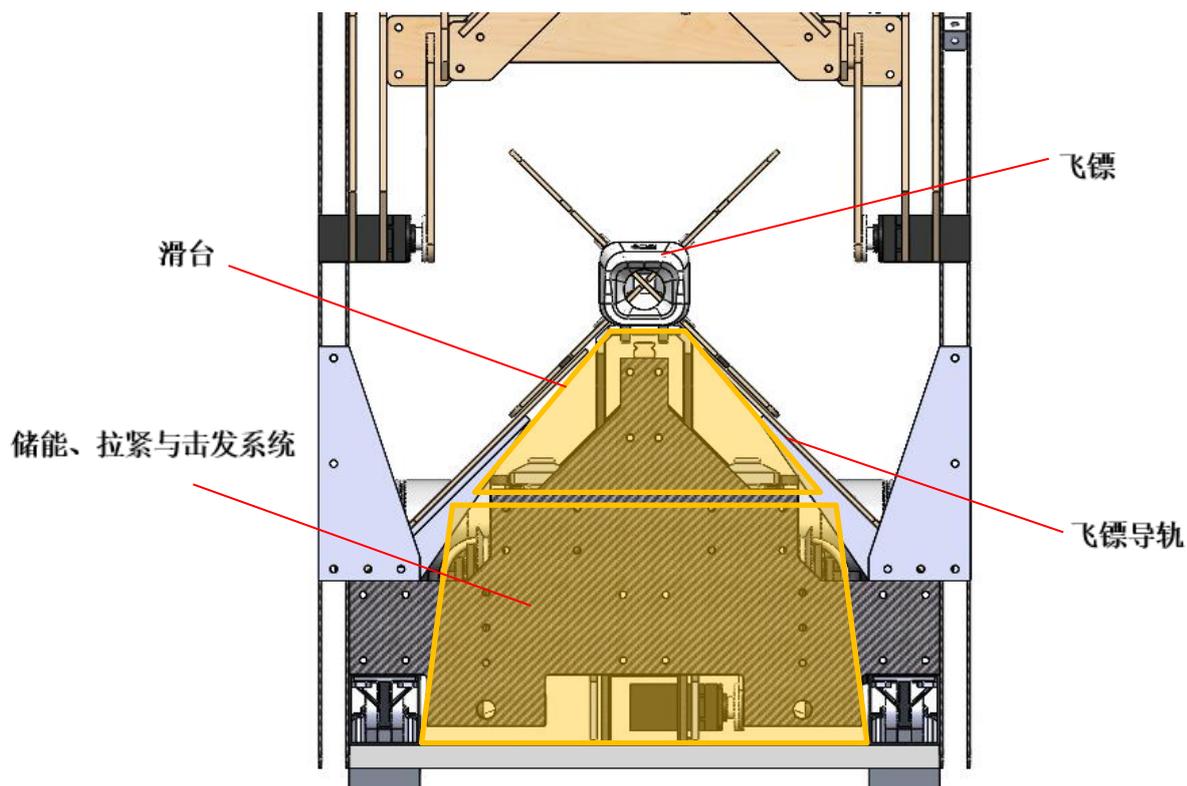


图 6.4.1.8 发射机构前视图

#### 5.4.1.2.4 滑台缓冲

滑台会以近 20m/s 的速度在较短的距离中减速，会产生较大冲击，为了让结构不被破坏，

经过了 3 次迭代。初始利用油压缓冲器，但缓冲距离过短，阻尼项大，高速时冲击太大，后来选用拉簧进行缓冲，阻尼小冲击较小，但是会有较大振动，拉簧减震第一版拦阻器过于看重刚度，导致重量较大，更增大了冲击。最终采用 mgn9 滑块+tpu 打印件+尼龙绳的方案进行拦阻。如图 6.4.1.9 所示，飞镖与滑台分离，拉簧伸长。

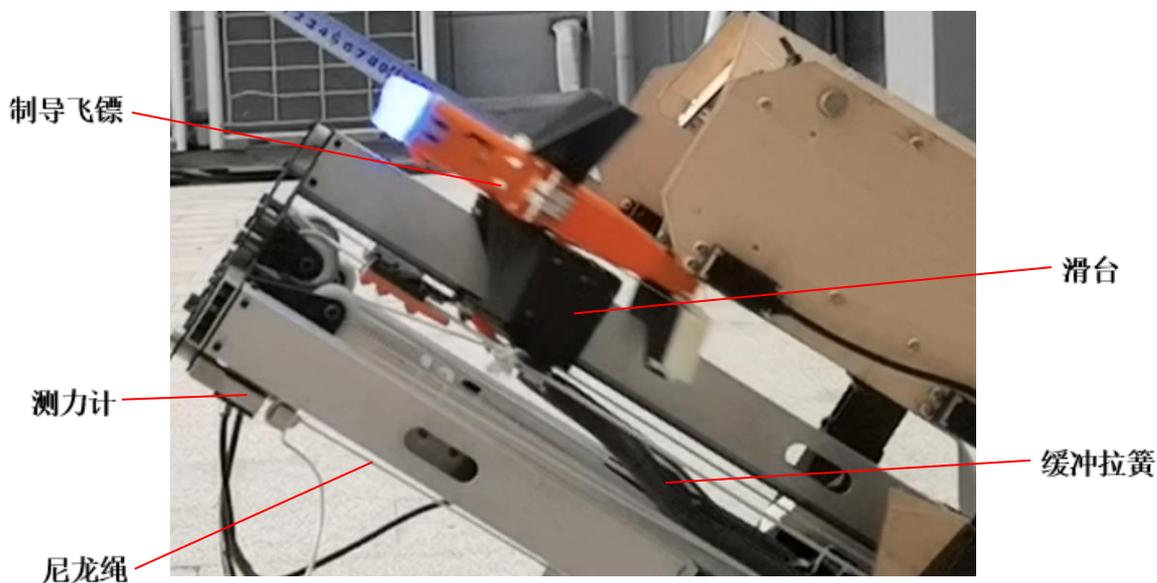


图 6.4.1.9 拉簧缓冲瞬间

### 5.4.1.3 装填系统

由于制导飞镖的尺寸较大，换弹又是巨大的考验。第一版设计为横向放置，通过飞镖左右及前后的移动减少空间占用，再从弹舱掉落至下方的滑轨上，但在实际测试中发现掉落过程会出现一些不可控的碰撞和反弹影响可靠性，会出现无法正常发射的情况。

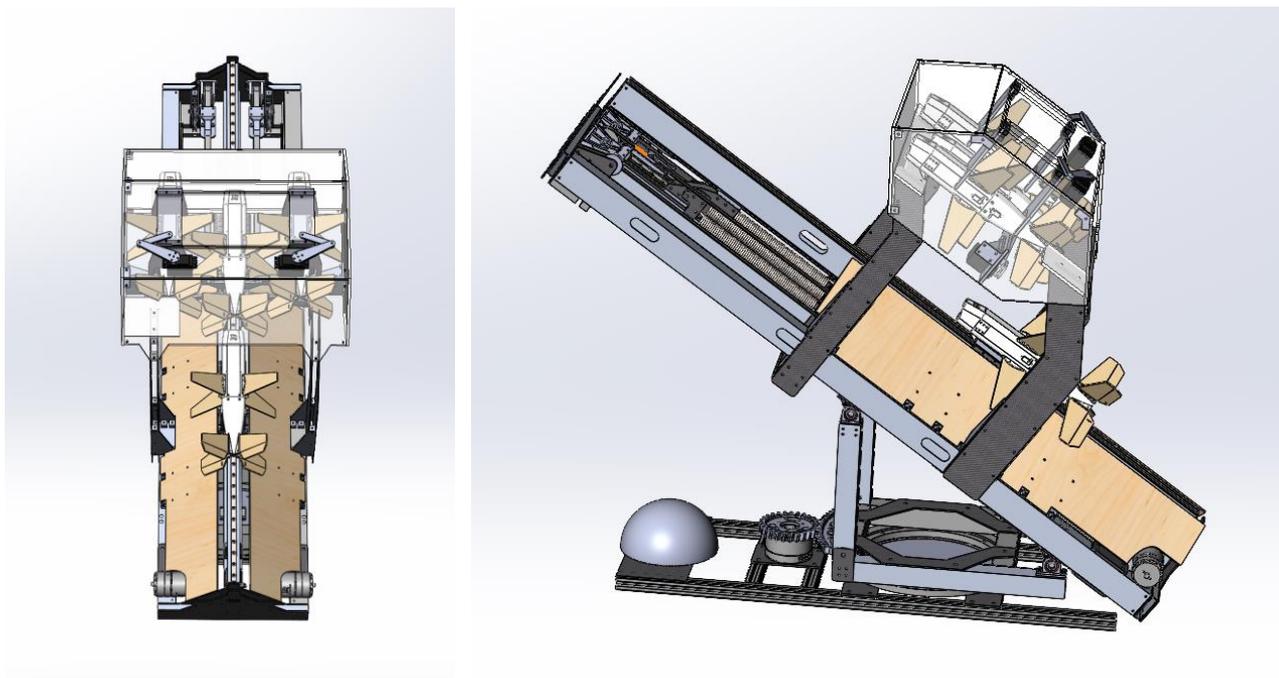


图 6.4.1.10 横向换弹方案

于是在空间上做出牺牲采用纵向装弹的方案，飞镖储存于上层滑轨，通过同步带进行前后控制，将下部的飞镖推至上部，通过平行四边形装弹机构转移至下层滑轨，滑落后卡在滑台上，平行四边形装弹机构上移避开发射路线，再进行发射。经过实验验证这种方案十分稳定。

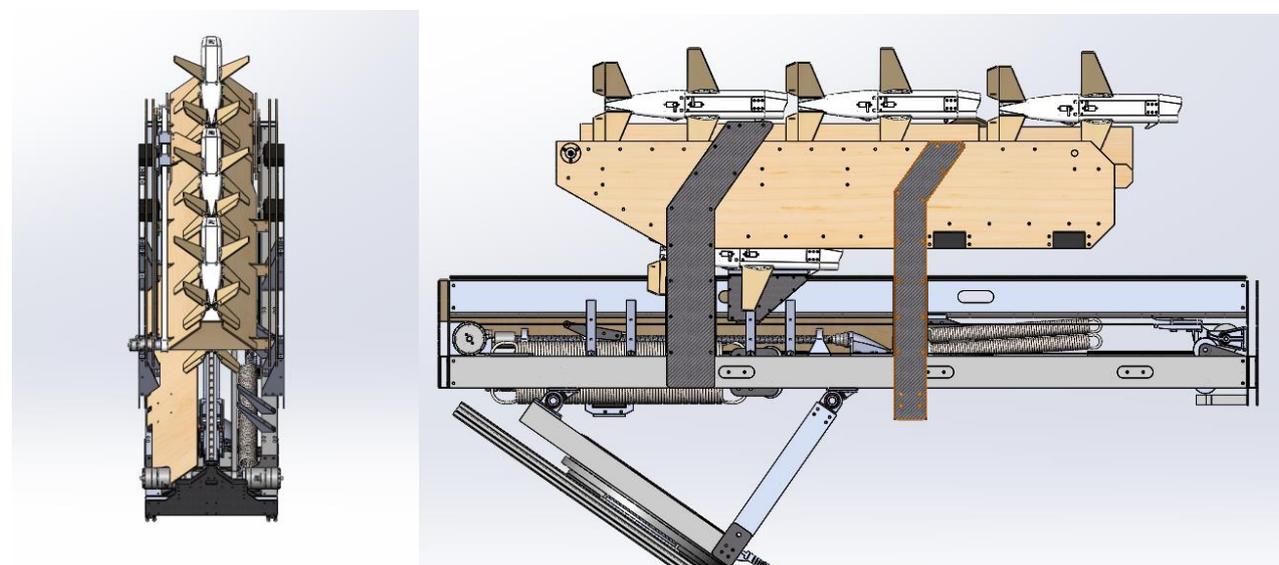


图 6.4.1.11 纵向换弹方案

对于材料的选择，静强度设计一般以比强度作为参数来选择材料，不同受载比强度的表达不同，主要考虑弯曲强度和刚度，弯曲时的比强度为 $\sqrt[3]{\sigma_b^2/\rho}$ ，弯曲时的比刚度为 $\sqrt{E/\rho}$ ，根据图 6.4.1.12 和图 6.4.1.13 选择。蓝色虚线为轻木的弯曲比强度和比刚度，可以看到在顺纤维方向，木材有着相当高的比强度和比刚度，尽管在垂直于纤维的方向较低，但经过垂直方向

复合的层木板依然有较好的力学表现。装填系统无较大集中载荷，所以对绝对的刚度强度要求不高，在受到较大弯矩的支撑柱使用玻璃纤维板材，在飞镖轨道采用层木板。

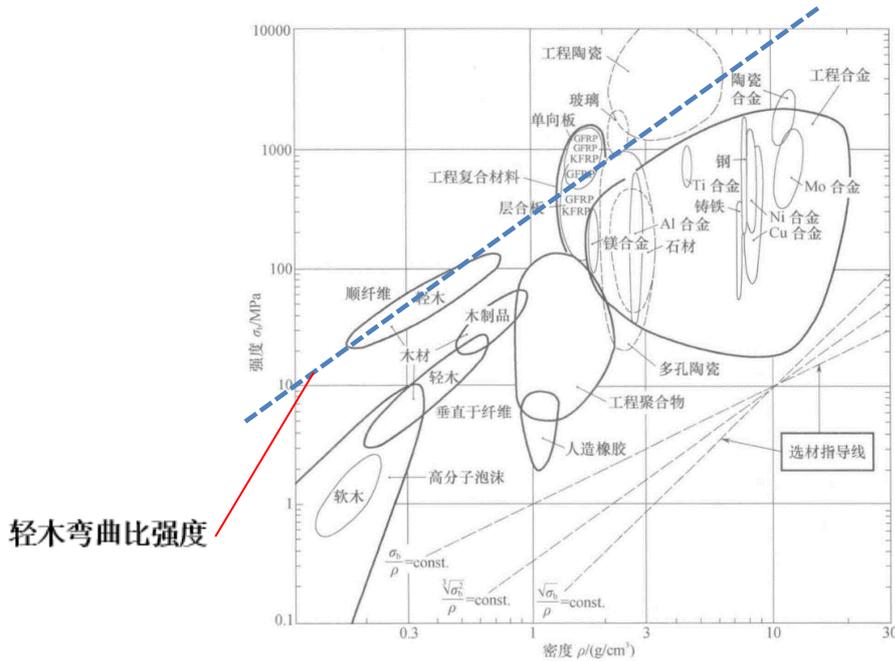


图 6.4.1.12 材料强度重量特性

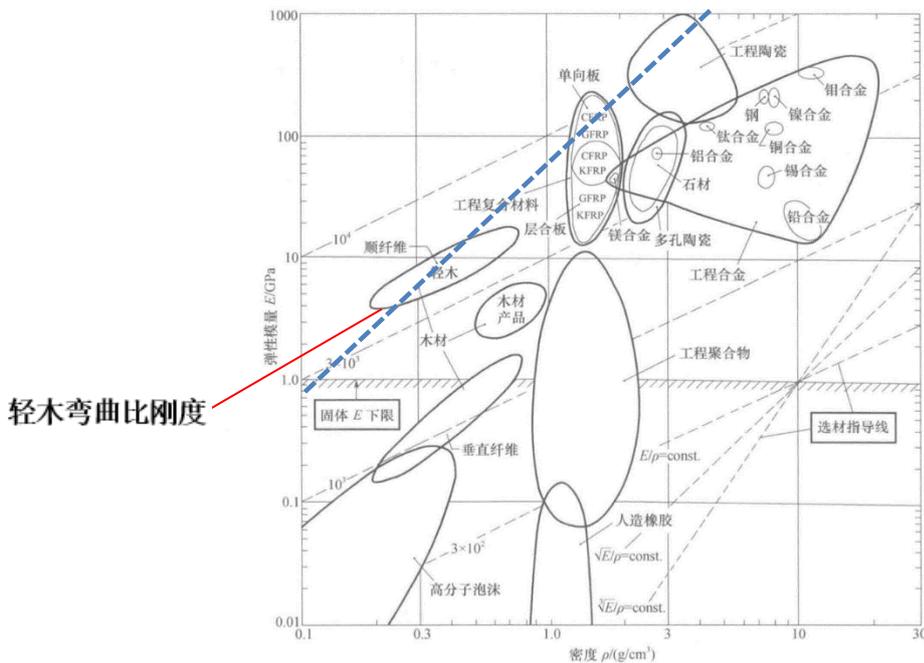


图 6.4.1.13 材料刚度重量特性

为进一步提高刚度强度降低重量，将结构板材设计为双层，中间使用铜柱连接抵抗弯曲剪应力，增大了结构的惯性矩，在支撑根部弯矩较大的地方进行加强，进一步增大刚度如图 6.4.1.14 所示，在飞镖轨道上布置诸多肋，增加刚度。

木板成本极低，大面积非承力部件上使用大幅降低了成本。

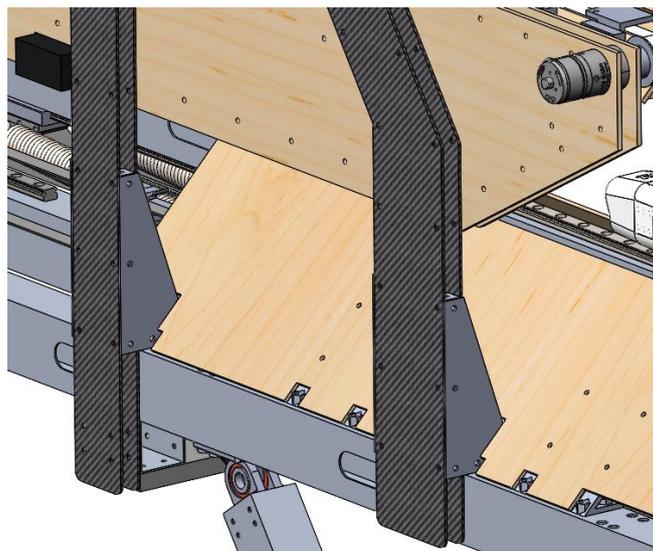


图 6.4.1.14 装填系统双层板材

#### 5.4.1.4 底座

底座如图 6.4.1.15 所示，为了提高发射的重复度，底座需要给发射机构提供稳定的初始角度。Yaw 轴通过一个 6020 云台电机调整，通过齿轮传动，齿轮使用尼龙材料，进行镂空减重降低打印成本，斜齿轮设计使得传动平稳，可调节位置设计使得齿轮传动几乎没有背隙；Pitch 轴设计为不可调，仅通过改变发射初速度改变落点，若可调会增加一个不必要的变量，增加的驱动机构也会产生误差，特别是有巨大后坐力和晃动的情况下。

使用餐桌轴承对上层结构和底座进行连接，防止晃动影响发射的角度。餐桌轴承通过垫板将上层铝型材和底座铝方管相连，连接刚度较好。

底座使用 4020 铝型材惯性矩较 2020 大四倍，底座更加稳定，下方粘贴橡胶条增大与地面的摩擦力，减少后坐力的影响，使用磁吸底座，在赛场上进一步提高稳定性。

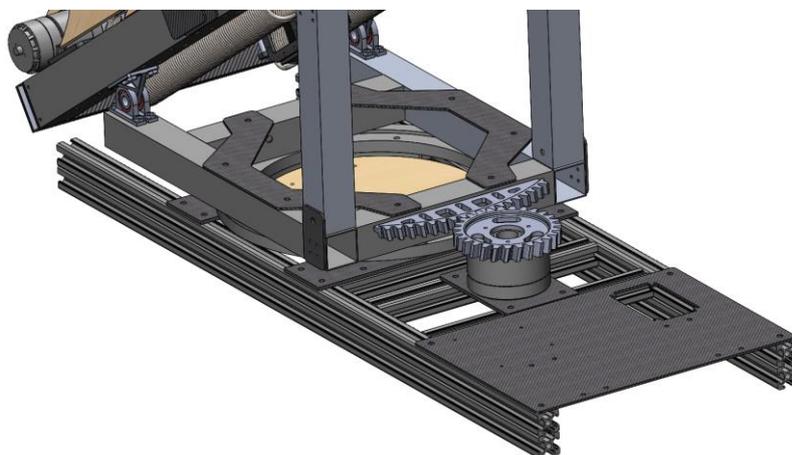


图 6.4.1.15 yaw 轴模块图

为了便于运输发射架，设计有快速折叠功能，大幅减少空间占用，如图 6.4.1.16 所示。

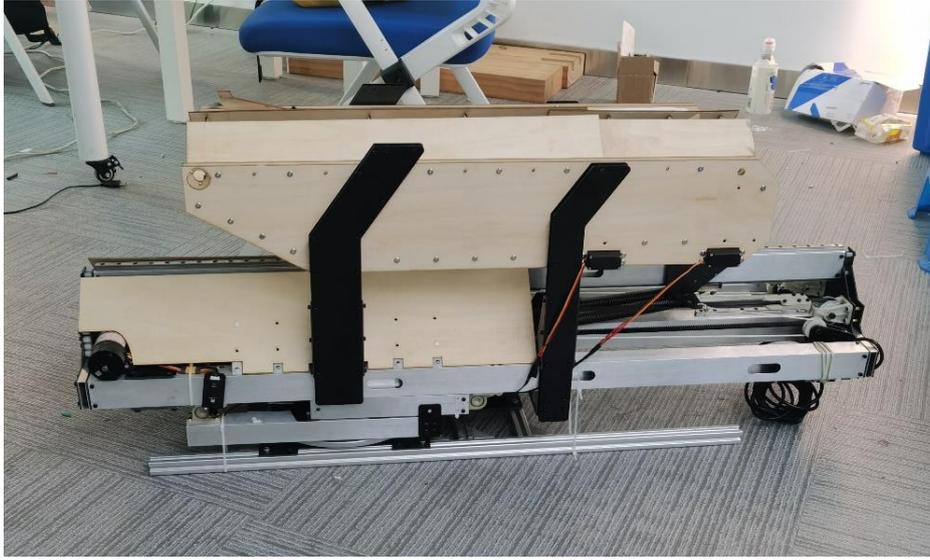


图 6.4.1.16 折叠状态

### 5.4.1.5 制导飞镖

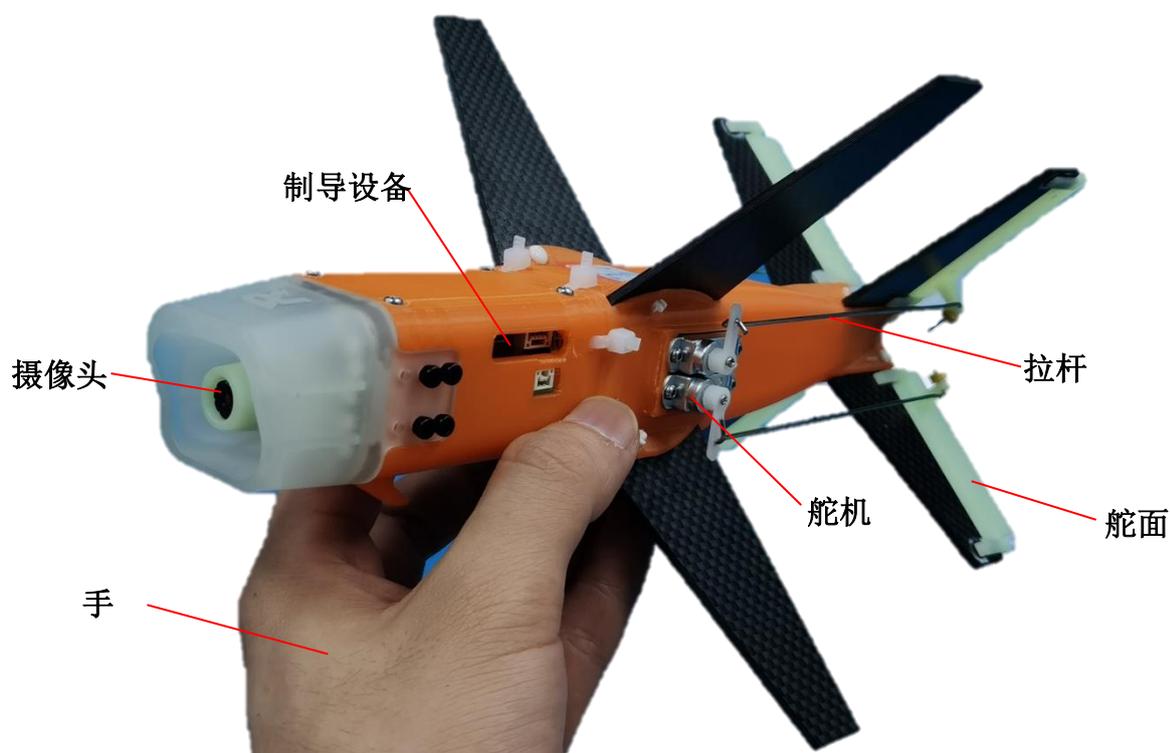


图 6.4.1.17 制导飞镖实物图

表 6.4.1.2 制导飞镖主要参数

名称	参数
制导飞镖质量/g	180
长宽高/mm	243*143*143
主翼面积/m <sup>2</sup>	0.0062 (可增加)
尾翼面积/m <sup>2</sup>	0.0033
舵面面积/ m <sup>2</sup>	0.00085
材料	碳纤维板、tpu、8228 树脂

### 5.4.1.6 气动设计

机身的截面形状选择为与飞镖头相同，机身的侧面形状拉伸出一段存放电池与飞控板的空间，后端收缩，形成一个流线型，减少压差阻力，避免涡街，提高稳定性，留出舵机安装

空间。

在机翼设计方面，当制导飞镖的飞行速度为 15m/s，机翼特征长度为 0.035m，在压强为 101.325kPa、温度为 20℃的条件下求出雷诺数  $Re = \frac{\rho v l}{\mu} = 35928$ ，雷诺数较低，粘性力占主要地位，会出现层流分离效应和非定常效应，容易受到湍流度和表面粗糙度的影响，大多数常规翼型在低雷诺数时的性能都不好，雷诺数减小，翼型的最大升力系数会降低，升力线斜率在过小的情况戏下也会降低，由于粘性的增强阻力系数不断增加，最大升阻比也会有较为复杂的变化，如图 6.4.1.18 所示。对翼型的优化收益较小，所以直接采用 **平板翼型**，加快迭代速度减少成本，升力不足再增大机翼面积，优化前缘半径、相对厚度、弯度等。

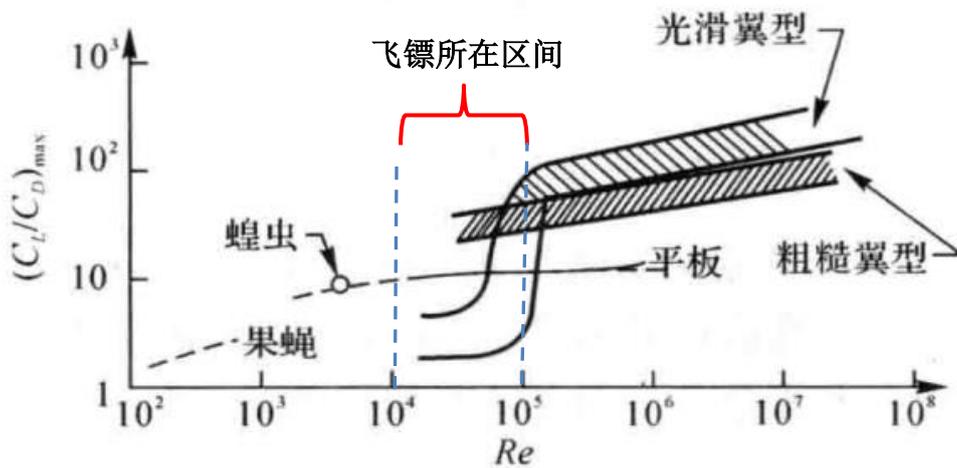


图 6.4.1.18 不同雷诺数下的大升阻比

对于提供升力的主翼面，根据升力线理论，椭圆速度环量分布的平直机翼

$$C_L = \frac{C_l^\alpha}{1 + \frac{C_l^\alpha}{\pi\lambda}} \alpha$$

$$C_{Di} = C_L \Delta\alpha_i = \frac{C_L^2}{\pi\lambda}$$

其中  $C_l^\alpha$  为升力线斜率， $C_L$  为升力系数， $\lambda$  为展弦比， $\alpha$  为迎角， $C_L$  为升力系数， $C_{Di}$  为升致阻力。可以看出升力线斜率随着展弦比增大而增大，诱导阻力随着展弦比增大而减小，所以为了减小阻力和提高升力应该选用较大的展弦比，展弦比公式为  $\lambda = l^2/S$ ， $l$  为机翼的展长， $S$  为面积，所以应该在规则的范围选用更大的展长。

而在大展弦比直机翼的失速特性中，由于诱导下洗速度沿翼展是不变的，因而沿展向各翼剖面的有效迎角也不变。所以，随着  $\alpha$  的增大，整个展向各翼剖面同时出现分离，同时达到翼

型的最大升力系数，同时发生失速，失速特性良好，如图 6.4.1.19 所示。

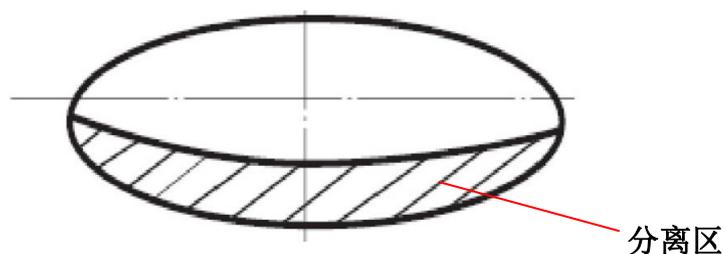


图 6.4.1.19 椭圆机翼失速示意图

综上所述，且考虑到与发射架的配合发射，无法采用十字型翼面，所以采用 x 型翼面，并且最大限度利用规则尺寸。在实际测试中，发现鸭翼操纵面在前，鸭翼在着陆时先接触地面，作为操纵面强度低，容易破坏，将飞镖布局变为传统布局，先触地的主翼为贯穿木板，将力分散传导至机身，抗冲击能力大幅提高。且在梯形翼梢根比 0.4 左右，具有与椭圆翼相似的效果。

XFLOW 软件是新一代的计算流体力学数值仿真软件，xflow 软件模拟流场时是基于介观格子玻尔兹曼理论方法和大涡模拟（LBM-LES）的，LES 可对微观小尺度进行模拟。计算模型无需划分网格，对模型要求不高，相比传统的数值模拟软件（如 Fluent 等），xflow 软件操作相对容易，可以实现计算机的并行计算，计算效率较高，如图 6.4.1.20 和图 6.4.1.21 所示为两个算例，分别计算  $0^\circ$  滚转角和  $45^\circ$  滚转角时  $10^\circ$  迎角状态下的涡量云图。

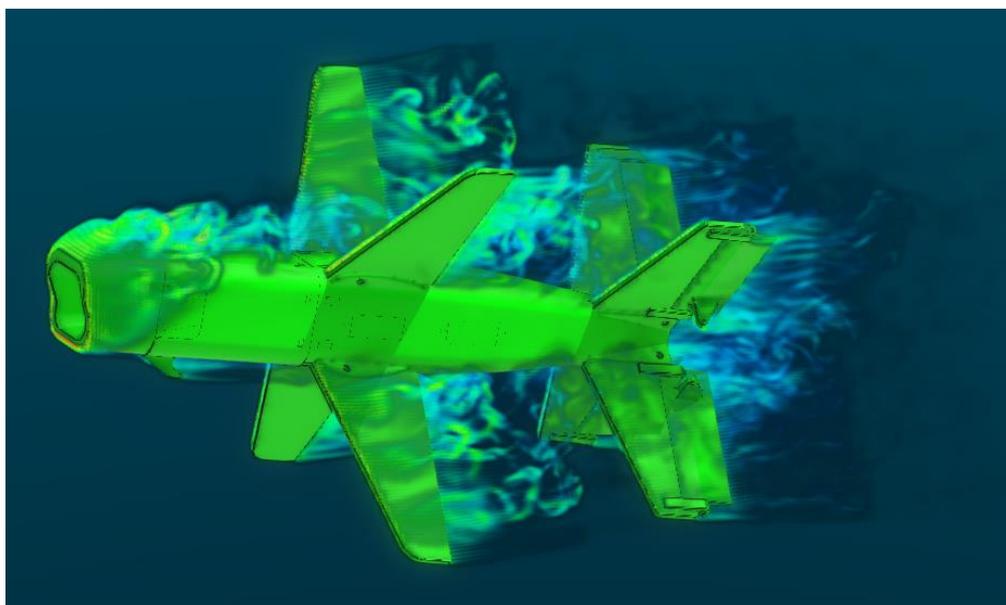


图 6.4.1.20  $0^\circ$  滚转角制导飞镖  $10^\circ$  迎角涡量云图

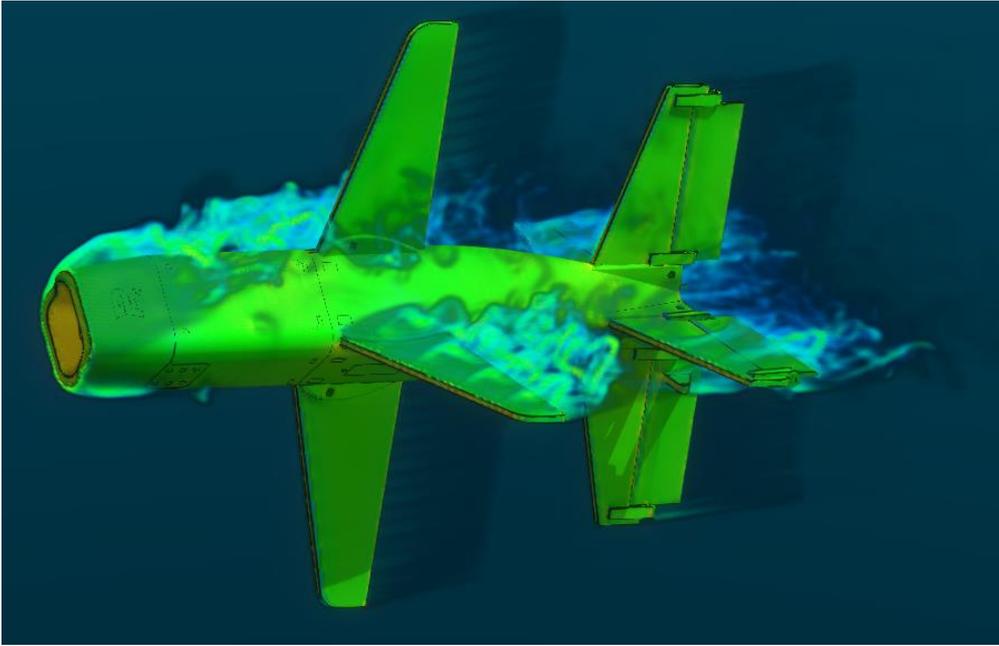


图 6.4.1.21 45° 滚转角制导飞镖 10° 迎角涡量云图

基于 XFLOW 的特点，一是可以对不同迎角的气动系数进行计算；二是可以研究力矩系数随迎角的非线性变化，给出合适的最大迎角；三是可以进行整个起飞过程的计算。

对 15m/s 的速度进行计算，舵面角度不变的情况下，改变迎角，得到升力系数与阻力系数如图 6.4.1.22 和 6.4.1.23 所示，由于为轴对称构型，也可代表不同侧滑角时的升力系数。由于存在两个方向：十字状态和 X 状态，对它们的升阻特性分别进行计算，得到的结果较为相近。

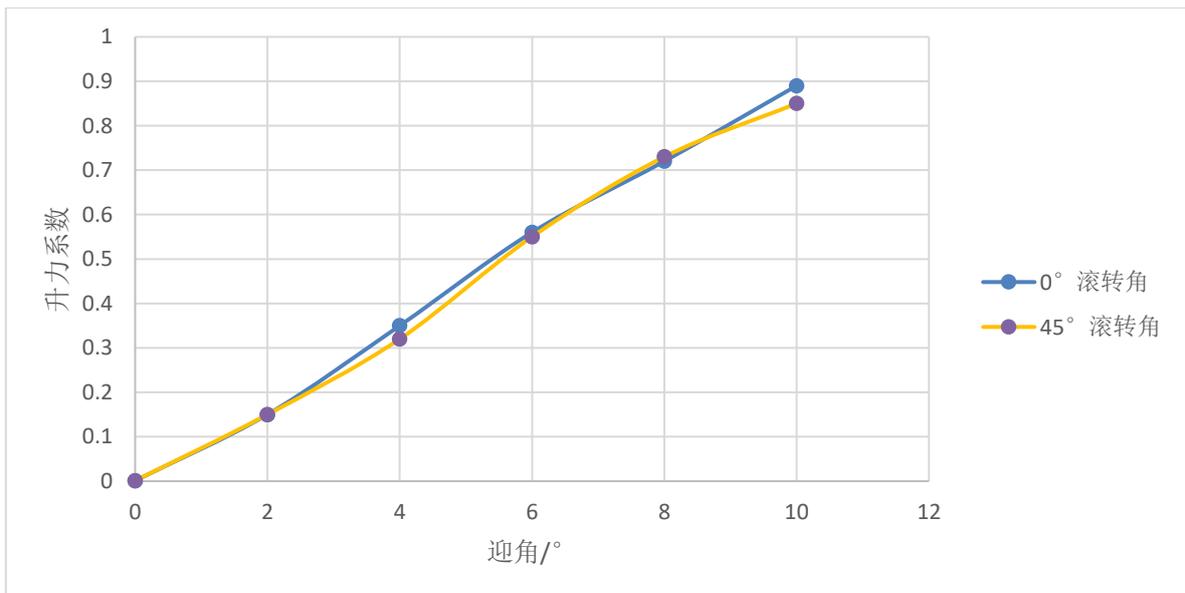


图 6.4.1.22 0-10° 迎角升力系数

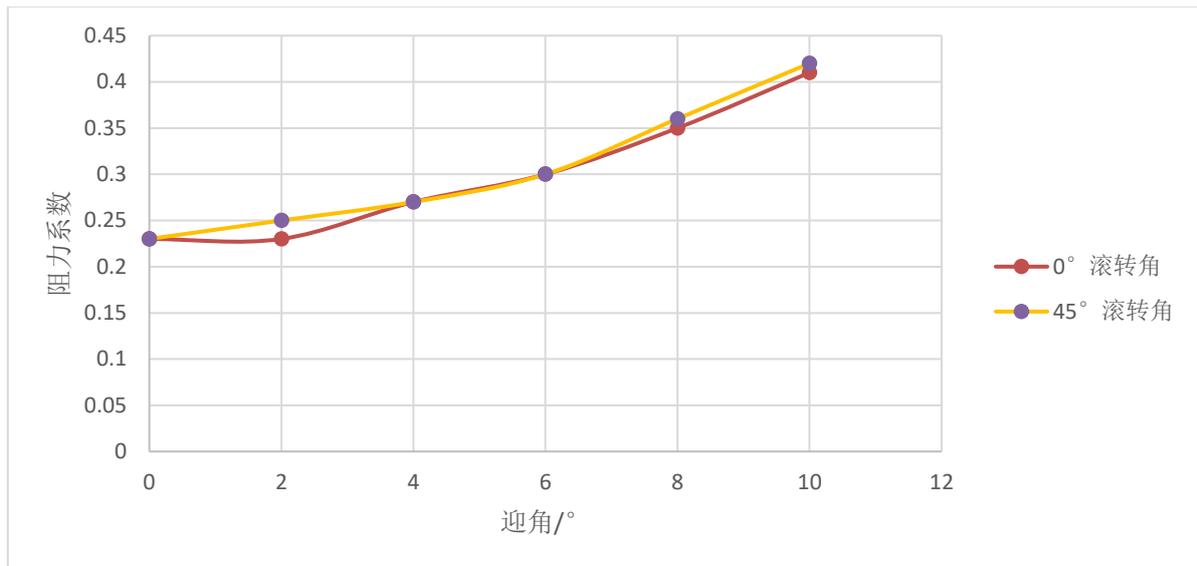


图 6.4.1.23 0-10° 迎角阻力系数

相比 XFLOW，还有另外两大类的计算方法，工程方法涡格法，例如计算软件 OPENVSP AERO，计算更为快速，但由于使用平板翼型无法直接求解，使用雷诺平均方法的 FLUENT，尽管更为主流准确，但操作较为麻烦，并且都不能得到准确的结果。得到的结果只有一定参考意义，为后面轨迹仿真做准备，误差源于其力矩状态未平衡，舵面没有角度，力矩平衡和重心的位置和舵面的角度有很大关系，并且计算出的气动参数因为雷诺数较低都有较大误差，需要根据实验数据做调整。

### 5.4.1.7 结构设计

制导飞镖的总体结构如图 6.4.1.24 所示。

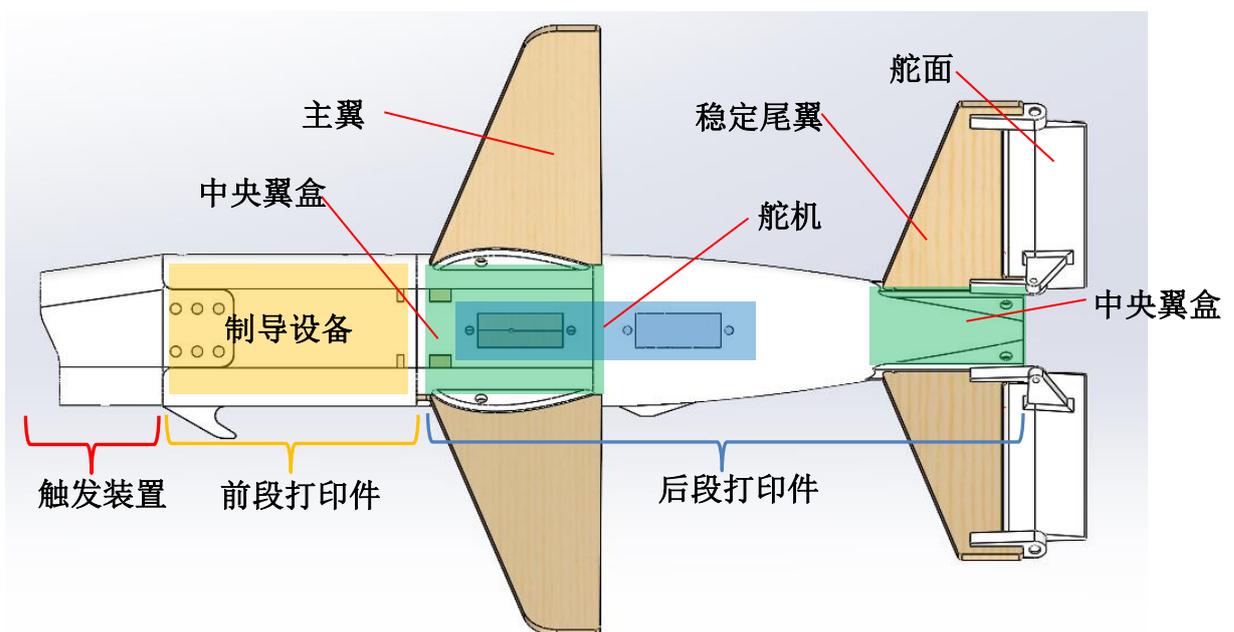


图 6.4.1.24 制导飞镖结构

飞镖机身的结构需要选择合适的机身外形，有效容积大，外形光滑阻力小；机身结构受力型式与主要受力构件的布置，应与各相连接部件的结构受力型式相协调，能传递头部的碰撞，传递机翼的碰撞，以及整体巨大加速度下的传力；合理使用机身的有效容积满足各种各种装在要求，舱位布局能满足重心的要求，有合适的稳定裕度；应便于更换电池、电路、舵机保证易维护性。

机身需要承受的外载荷主要为落地时正向的冲击力，以及弹跳后的侧向冲击，正向冲击通过飞镖头传导至机身，侧向冲击通过机翼传导至机身；而气动力和质量力可视为小量，保持足够的刚度即可满足，飞镖的结构设计考虑冲击载荷为主。硬壳式机身由少量隔框和厚蒙皮组成，蒙皮既承受正应力又承受剪应力，隔框用于维持机身外形支持外形和扩散集中力，与半硬壳式机身相比，硬壳式机身的最大缺点是重量特性不好，在有开口的地方需要补强额外增加了重量机身打印件如图 6.4.1.25 所示。

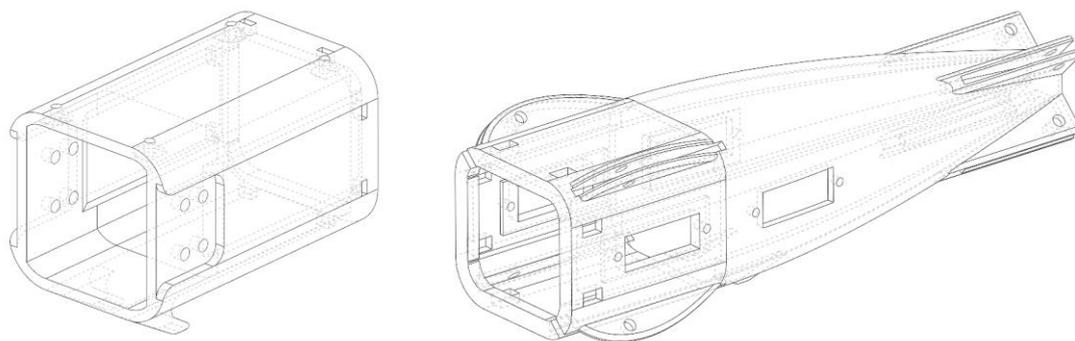


图 6.4.1.25 制导飞镖机身 3d 打印结构

机翼的大展长和落地冲击之间存在矛盾，参考现有的飞机继续以设计，不适于如此小空间的布置，例如梁式机翼，翼梁是主要受力构件，蒙皮和桁条较弱，当机翼的高度较小，材料利用率低，破损安全性较差；单块式机翼蒙皮、桁条和梁缘条组成的上下壁板具有较强的面内抗拉压及剪切能力，单块式比梁式的刚度特性好，在厚度较小时材料利用率高；多腹板式机翼多应用于高速薄翼飞机上，蒙皮较厚，有较多的墙，翼肋较少，主要由蒙皮传递弯矩，材料分散性更大，刚度更大材料利用率更好。条件允许时机翼应尽量设计成整体贯通机身，若采用接头与机身连接，机身将会承受冲击带来的弯矩，打印件的强度难以承受，否则会大幅增加重量。若有中央翼盒，弯矩则由机翼承受，机身打印件只受到正应力，虽然会占用一部分机内空间，但可以接受，如图 6.4.1.26 所示。

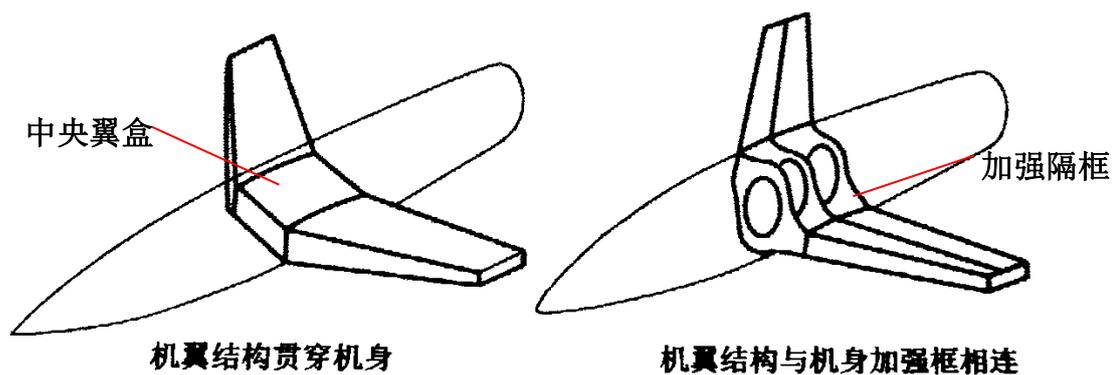


图 6.4.1.26 中单翼与机身连接的典型结构形式

木材表现十分好，但由于木材是各向异性材料，需考虑木材的种类和纤维方向，在顺纤维方向强度高，在垂直方向强度较低，在碰撞中主要为弯矩载荷，需加强展向方向的强度，考虑厚度与强度的平衡，开始选用 2mm 椴木层板激光切割，成本很低，为进一步增加可靠性减少维护，选用碳纤维板，弹翼形状如图 6.4.1.27 所示。

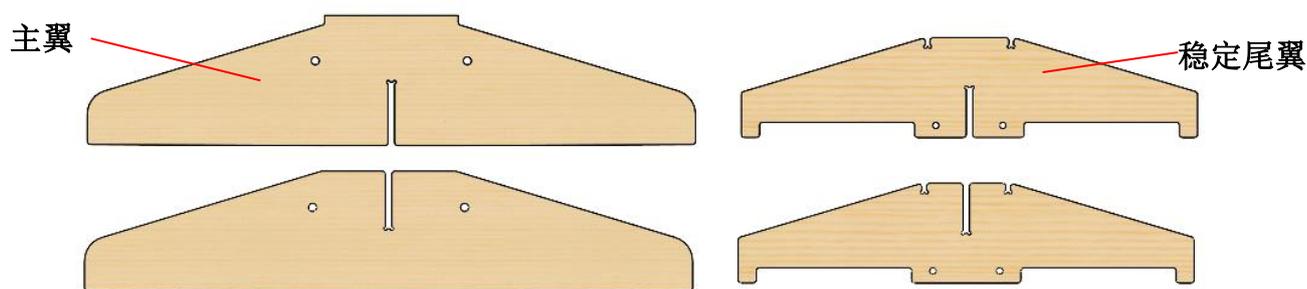


图 6.4.1.27 制导飞镖弹翼

尽管舵面在机身最后，但在摔落时会产生弹跳翻转，尾巴弹跳后落地时十分容易被破坏，稳固的舵面十分重要，开始参考航模采用胶带对舵面进行固定，使用现成舵臂拼接，但是可靠性较低，后来采用 3d 打印整体舵面与转轴，材料选用 8228 树脂，可靠性大幅提升，利用阵列打印舵面大幅减少成本，零件与装配状态如图 6.4.1.28 所示。

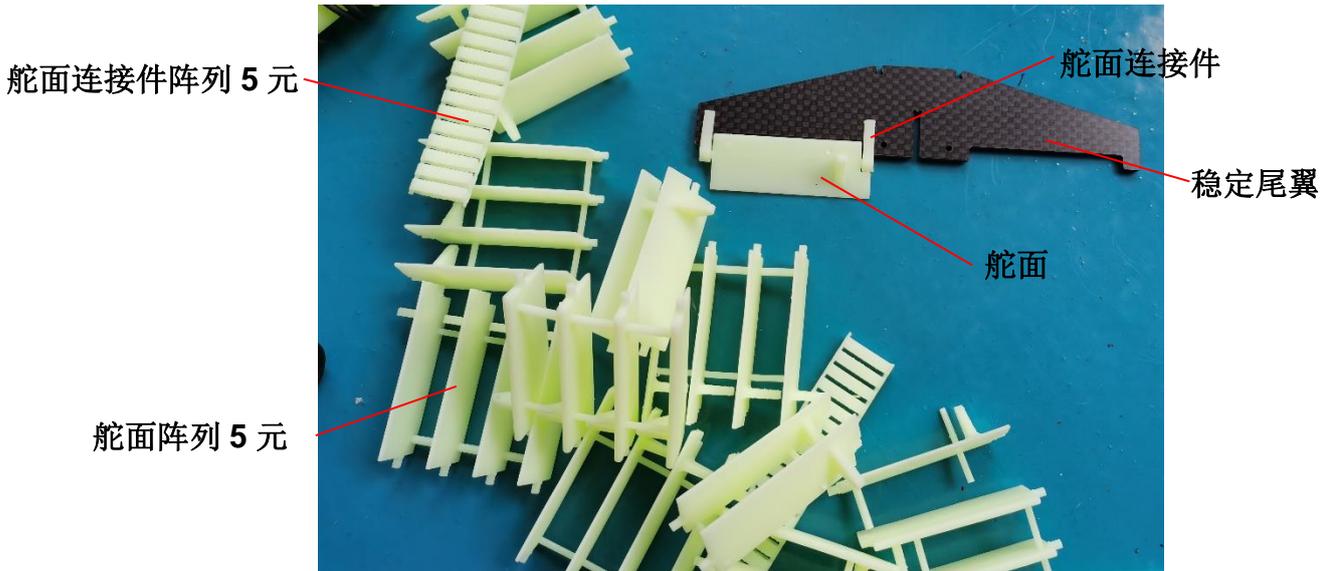


图 6.4.1.28 制导飞镖舵面及安装状态

迭代过程：飞镖的结构型式结合上述常用的机身机身结构特点，综合考虑加工工艺的特点，第一版无控飞镖采用尼龙烧结工艺，可以快速批量打印出大部分形状，第一版开口较少，仿照合理布置桁条和隔框，加强桁条将飞镖头的集中力进行扩散传递到机身结构上去，考虑剖面的应力状态，桁条的粗细从前向后变细，并且无控飞镖可做鸭式有控飞镖的结构验证。

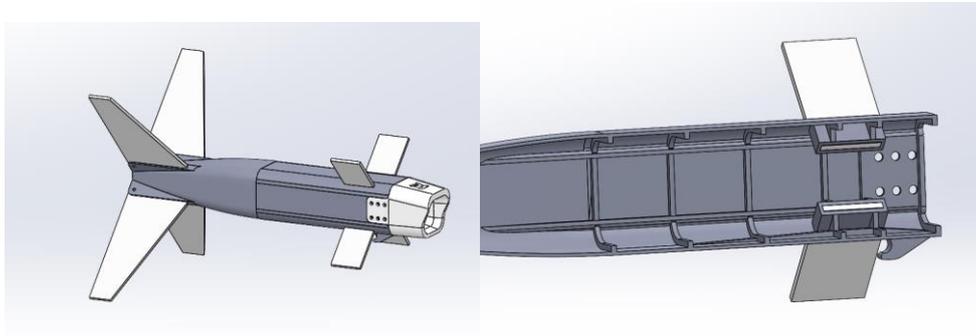


图 6.4.1.29 无控验证飞镖及内部结构

但是尼龙烧结工艺成本较高，若大量迭代会消耗太多经费，并且由于刚度较大，抗冲击性能不如 tpu，在实际测试中发现在一些角度碰撞会发生破坏，但是采用 fdm 工艺打印，许多高处的隔框由于悬空便不能打印，改变工艺也需要进行结构上的变化，加厚蒙皮来维持取消框的刚度减少。若是采用鸭式布局，鸭翼会受到巨大的冲击，舵面基本为一次性，所以有控飞镖不能使用鸭式布局。

结合实验的经验将布局选择为正常布局，前主翼后控制舵面，根据 fdm 的工艺特点和破坏结果改进了内部结构，设计分离面，飞镖分为前后两部分，便于安装贯穿主翼和舵机，贯穿式弹翼在受冲击时有较长寿命，操纵面在后得到保护，不会使舵机受到扭矩冲击，在保证

操纵效率的同时，有着较大的稳定裕度，在分离面附近设计加强隔框，便于连接处击中力的传递，提高抗弯抗拉刚度。

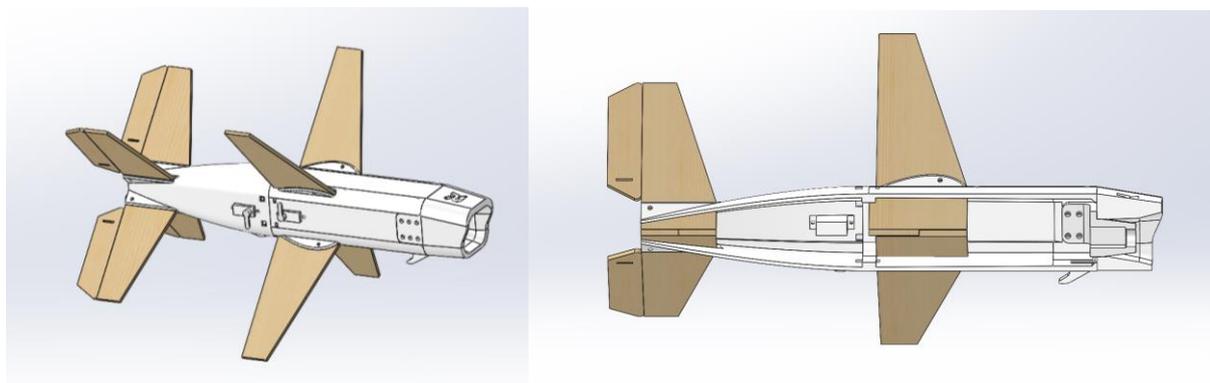


图 6.4.1.30 有控飞镖及内部结构

### 5.4.1.8 飞行轨迹仿真

将运动方程简化为质心在铅锤面内运动的运动方程，飞行速度矢量和作用于质心上的外力均位于对称平面内。飞机受到气动力、重力和推力作用，运动方程可分解为航迹切向和航迹法向：

$$m \frac{dV}{dt} = T \cos \alpha - D - mg \sin \gamma$$

$$mV \frac{d\gamma}{dt} = T \sin \alpha + L - mg \cos \gamma$$

式中 $\gamma$ 为航迹角，几何关系的参照如图 6.4.1.31 所示：

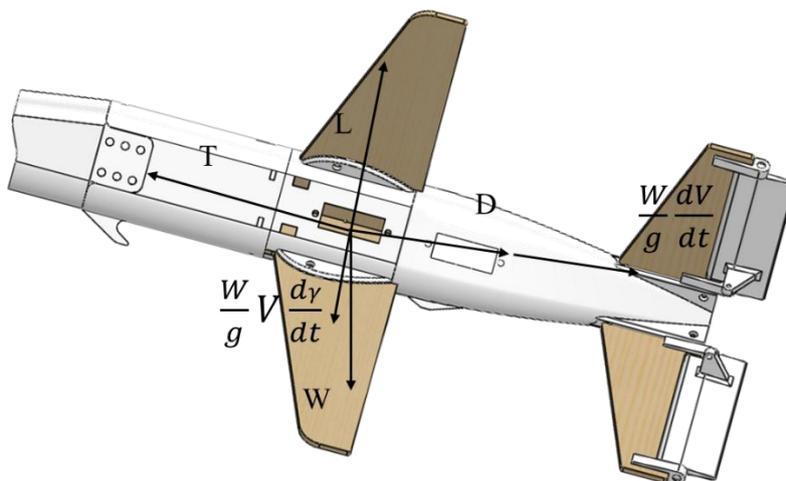


图 6.4.1.31 几何关系的参照

其中 T 为推力，在机体坐标系中；L 为升力、D 为阻力在气流坐标系中；W 为重力始终朝下，在地坐标中； $\frac{W}{g} V \frac{d\gamma}{dt}$  为法相加速度， $\frac{W}{g} \frac{dV}{dt}$  为切向加速度在航迹坐标系中，几个坐标系

涉及到相互转化，在三维的情况与二维相似，但多了偏航与滚转的参数，机体坐标系与地坐标系转化如图 6.4.1.32 所示。

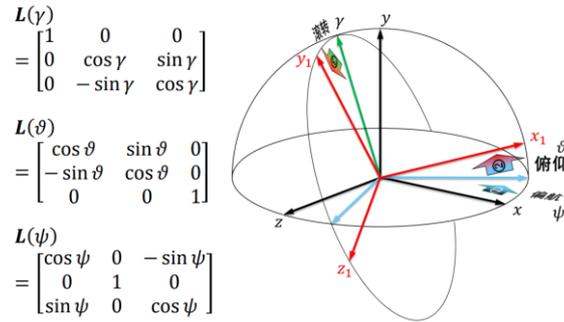


图 6.4.1.22 坐标系转化

使用 matlab 编写二维及三维的飞行轨迹计算程序，部分代码如图 6.4.1.23 所示，三维的仿真程序中，可对一般的飞行轨迹进行仿真，研究迎角、侧滑角、速度、初始角度、密度、升阻特性的变化对飞行轨迹的影响，如图 6.4.1.34 所示，不同颜色为不同迎角和侧滑角的飞行轨迹。

```

%-----解析计算-----
Cl_pitch=Cl_arfa*pitch; %尾翼上偏为正
Cl_yaw=Cl_arfa*yaw; %尾翼右偏为正
Cd=Cd0+A*Cl_pitch^2+A*Cl_yaw^2;

X_d=X_d+V*cos(theta_d)*cos(fai_d)*dt;
Y_d=Y_d+V*sin(theta_d)*dt;
Z_d=Z_d+V*sin(fai_d)*cos(theta_d)*dt;

q=0.5*rhou*V^2;
a_x=-q*S*Cd/m; %阻力加速度
a_y=q*S*Cl_pitch/m; %升力加速度
a_z=q*S*Cl_yaw/m; %偏航力加速度

%仰角变化
if((cos(theta_d)*V-a_y*sin(theta_d)*dt)>0)
    theta_d=atan((sin(theta_d)*V-g*dt+a_y*cos(theta_d)*dt)/(cos(theta_d)*V-a_y*sin(theta_d)*dt));
end
if((cos(theta_d)*V-a_y*sin(theta_d)*dt)==0)
    theta_d=0.5*pi;
end
if((cos(theta_d)*V-a_y*sin(theta_d)*dt)<0)
    if((sin(theta_d)*V-g*dt+a_y*cos(theta_d)*dt)>0)
        theta_d=pi+atan((sin(theta_d)*V-g*dt+a_y*cos(theta_d)*dt)/(cos(theta_d)*V-a_y*sin(theta_d)*dt));
    end
    if((sin(theta_d)*V-g*dt+a_y*cos(theta_d)*dt)==0)
        theta_d=pi;
    end
    if((sin(theta_d)*V-g*dt+a_y*cos(theta_d)*dt)<0)
        theta_d=-pi+atan((sin(theta_d)*V-g*dt+a_y*cos(theta_d)*dt)/(cos(theta_d)*V-a_y*sin(theta_d)*dt));
    end
end
%仰角变化
    
```

图 6.4.1.33matlab 部分代码

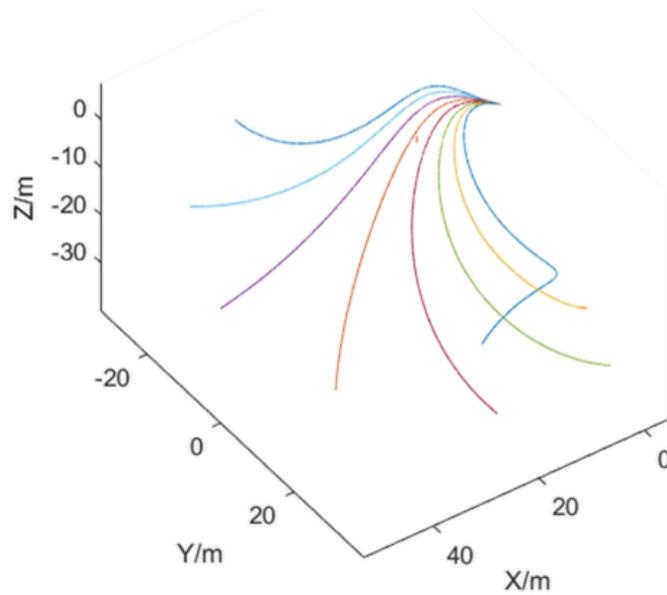


图 6.4.1.34 三维飞行轨迹

也可在二维情况分析不同参数变化的影响，例如图 6.4.1.35 得到了不同速度情况下对轨迹、速度、传感器读数变化的影响，不同颜色的曲线代表不同的初速度，结果对于制导算法的验证和无控制飞镖的精度分析具有很大意义。

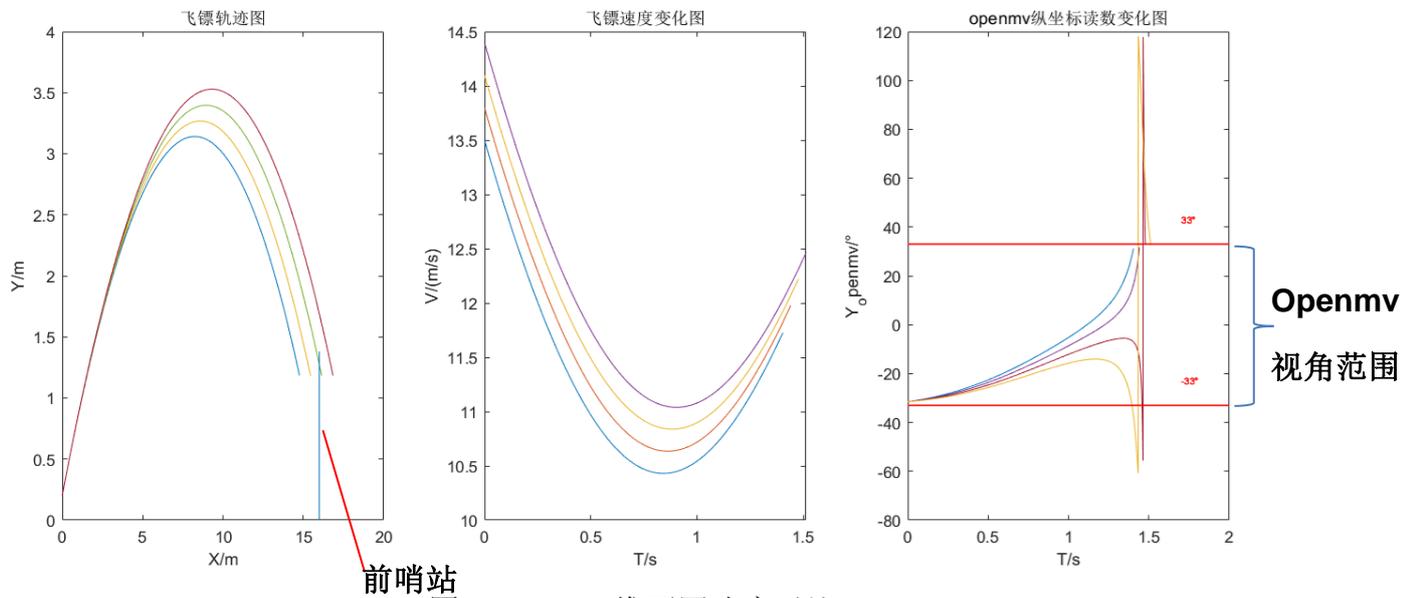


图 6.4.1.35 二维不同速度对比

将前文气动仿真结果带入二维轨迹仿真代码中，计算 $-10^\circ$ 迎角到 $+10^\circ$ 迎角对落点的影响，结果如图 6.4.1.36 所示，可见气动力产生的法向加速度对轨迹产生了很大影响，前后落点有近 6 米的区别，但这样的影响是全程累加的，若是末端制导可能会不足，但主翼面积可以通过木板拼接增加面积，由碳纤维板材承受冲击载荷，增加的木板不易损坏，并且可以增加气动力以提高机动性。

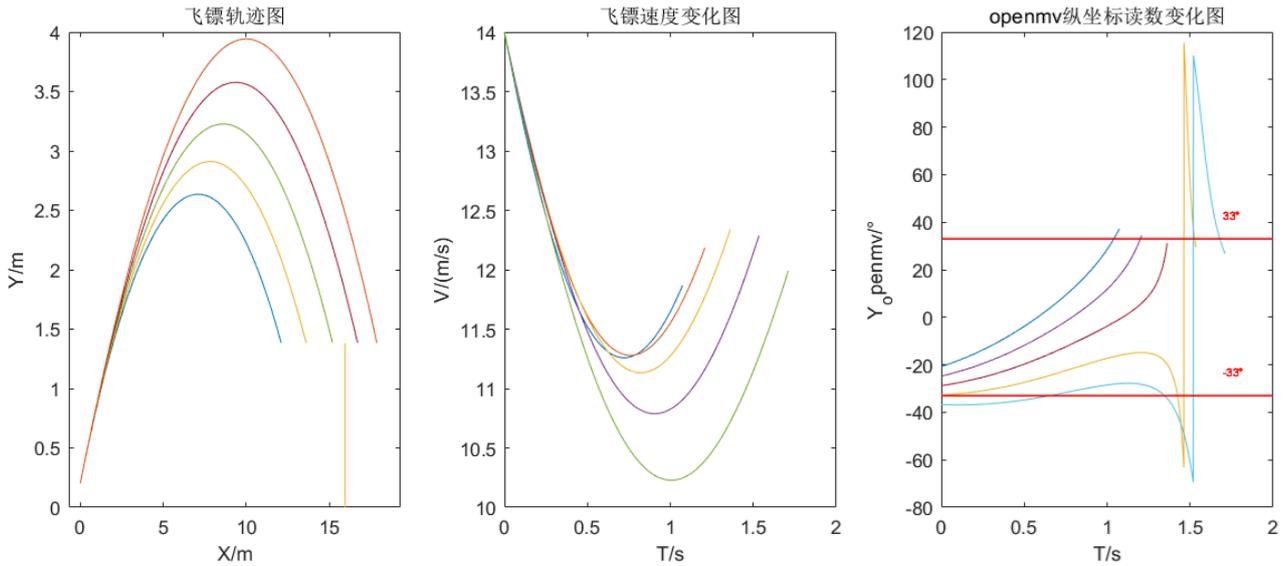


图 6.4.1.36 二维不同迎角对比

有了飞行轨迹仿真代码，可结合飞镖架初速度代码研究完整过程中的参数影响，例如图 6.4.1.37 所示为不同弹簧初始张力与发射行程对落点位置的影响，可以得到他们之间的定量关系，这在改变快速落点位置时起到重要作用，通过多项式拟合写入单片机中，便能在赛场上快速得到落点的位置。

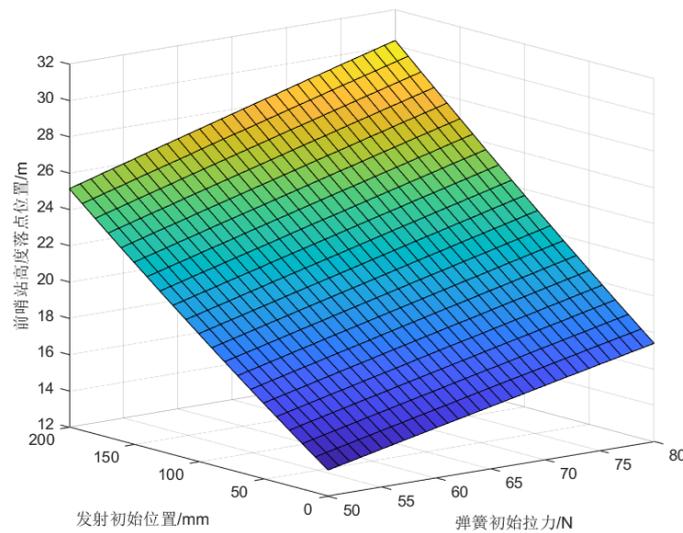


图 6.4.1.37 不同初始拉力和初始发射位置对应的落点位置

将飞镖的动力学代码去除升力项保留阻力项，改变初始速度、质量和阻力系数，再加上牛顿迭代便可得到大小弹丸的重力补偿的关系。如图 6.4.1.38 所示，为小弹丸在 15m/s 时击打不同位置目标需要的射角补偿与时间补偿，经过一次简单修正便具有较高的准确度，大幅减少了标图的时间。

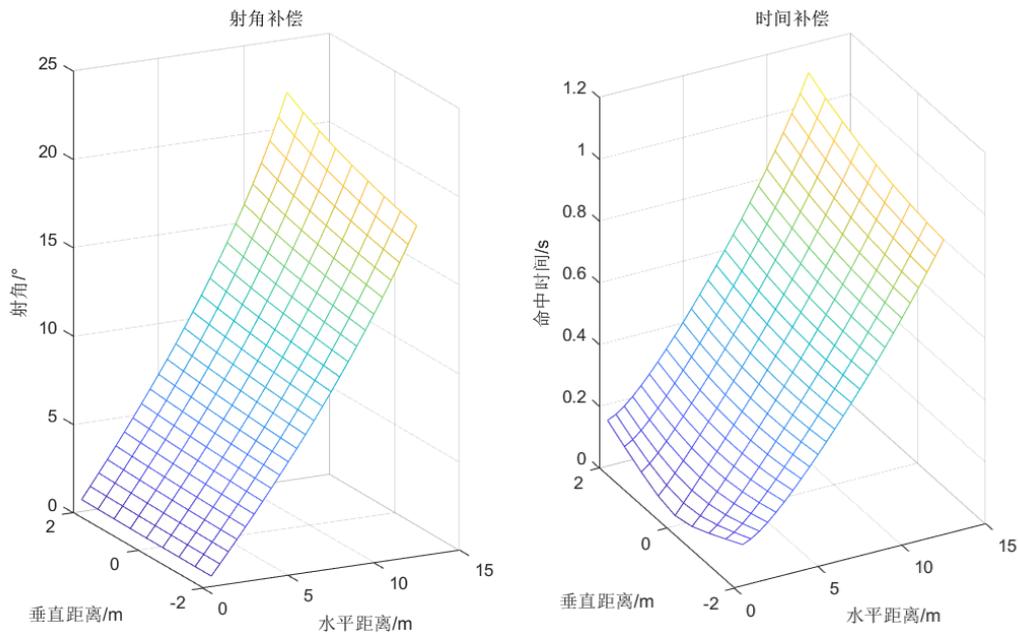


图 6.4.1.38 小弹丸重力补偿

### 5.4.1.9 无控制飞镖

无控飞镖需要较高的重复度，不需要主翼面提供法向加速度以改变轨迹，并且因为装配及加工的公差会使主翼面有一个初始的迎角，产生的随机法向加速度会影响落点，经过实验也的确如此，所以无控制飞镖仅保留较小的主翼面来适应双轨道自动装弹的环节，并且采用镂空设计进一步减小翼面积来减小升力，增大尾翼面积进一步增加稳定性，除了翼面，其他结构都与制导飞镖相同，模块化设计减少了成本，无控制飞镖结构如图 6.4.1.39 所示，实物图为 6.4.1.40 所示。得益于先前的结构设计，无控飞镖在换装碳纤维弹翼后表现出极好的可靠性，四只飞镖测试了一个月，赛前再加工了四只飞镖作为备件。

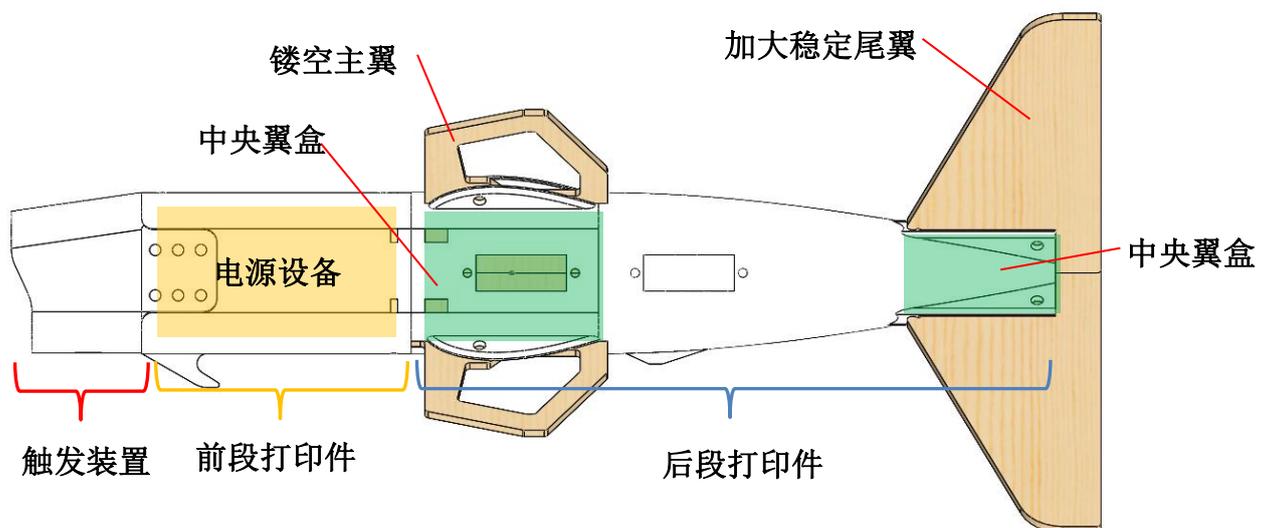


图 6.4.1.39 无控制飞镖



图 6.4.1.40 无控制飞镖实物图（左一）

## 5.4.2 硬件设计

### 5.4.2.1 发射架硬件方案

作为飞镖的发射系统，飞镖发射架的稳定性和耐用性至关重要，故飞镖发射系统所用控制系统和动力输出系统均采用 ROBOMASTER 官方产品。在硬件选型时，由于发射架装弹装置、拨杆装置和击发装置均使用舵机作为输出装置，故所需舵机口较多，且拉力传感器需要 12V 直流电源供电，故我们选择接口多、电源丰富的 ROBOMASTER A 型开发板作为主控板。控制板通过 IIC 接口与一个 0.91 寸的 OLED 屏幕连接，用以滚动显示飞镖发射架当前 YAW 轴位置和计算的基地位置、前哨站位置、拉力数据等必要数据。YAW 轴电机为 M6020，用以锁定和调整发射架 YAW 轴角度，其绝对值编码器可以方便赛后对发射架数据复盘统计，发射架硬件如图 6.4.2.1 所示。

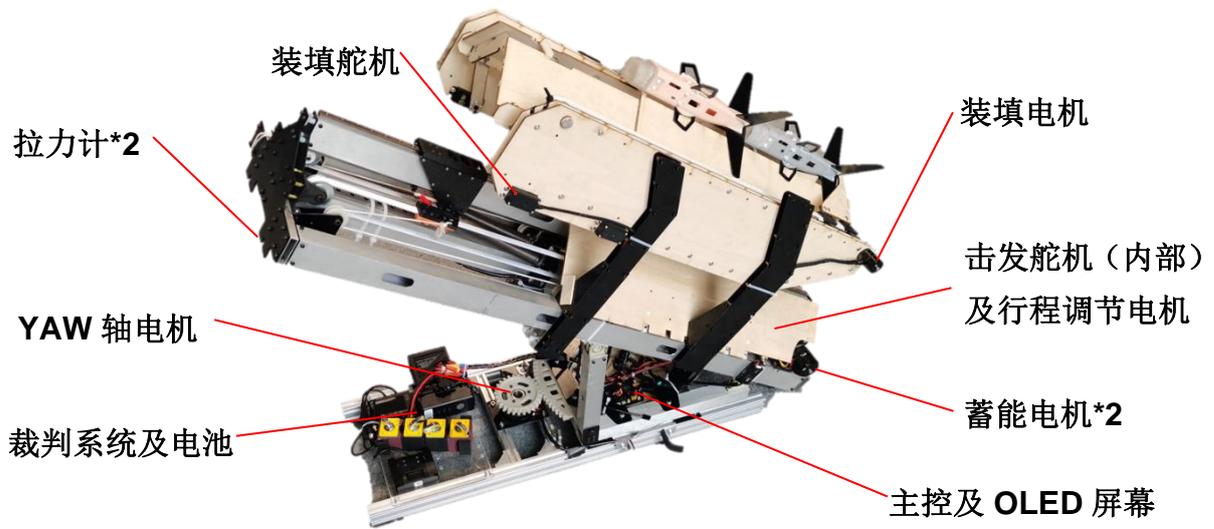


图 6.4.2.1 发射架硬件

飞镖蓄能装置对电机的扭力要求高，故我们使用了两套 M3508 无刷动力系统拉动弹簧蓄能。而在对扭力要求较小的装填和行程调节部分，我们使用了 M2006 无刷动力系统作为驱动电机，做到物尽其用、节约成本。飞镖的平行四边形装填机构和击发装置使用了 MG996 舵机，5 个舵机成本一共 60 元。为了方便安装，发射架拉力传感器使用了众精科技的 S 型拉力传感器，两个拉力传感器的数据通过变送器电路转化并输出 RS485 信号，RS485 信号通过 RS485 转串口模块与 ROBOMASTER A 型开发板通信。发射架的硬件图如图 6.4.2.2 所示：

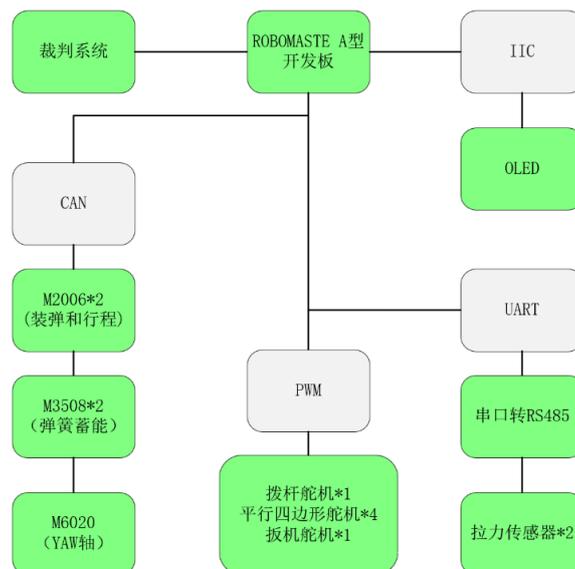


图 6.4.2.2 发射架硬件框图

### 5.4.2.2 有控飞镖硬件方案

若要实现飞镖的制导，至少需要姿态控制系统和视觉识别系统。故作为飞镖的电控，我选择了 STM32F401CCU6+MPU6050+333hz 高刷新率的舵机的解决方案。选择 STM32F401 的原因是在 2022 赛季备赛期，芯片价格疯涨，为了降低成本，我翻出了学长叶佬百宝箱里面的 POS 机拆机板，拆下主控作为飞镖的主控。MPU6050 作为一款畅销且便宜的 IMU，其稳定性和易用性得到了大家的认可，用在飞镖上也完全合情合理。在这三者中成本最高的是 333hz 微型舵机，单个舵机达到 39 元，但飞镖发射后距离短、速度快，传统航模里面所使用 50hz 刷新率的舵机很难满足需求，为了给控制留出余量，我们选择了蓝箭 333hz 刷新率 4.3g 舵机控制飞镖镖体尾翼的舵面，实现了高刷新率的姿态及轨迹控制。

对于视觉识别方案，我们选择了 OPENMV 平台，用低成本的 STM32H750（40 元）来作为视觉处理器，OV7725 摄像头（25 元）作为识别摄像头，其实际使用中在 QQVGA 画质下能够达到 60FPS 的识别帧率。姿态控制系统和视觉识别系统通过串口进行通信，达到末端制导的效果。



图 6.4.2.3 飞镖硬件结构图

在飞镖控制板的设计中，为了方便调试，我们给两个单片机都预留了 SW 调试接口，同时也提供了 macro usb 接口用于调试 OPENMV，飞控板元件如图 6.4.2.4 所示。



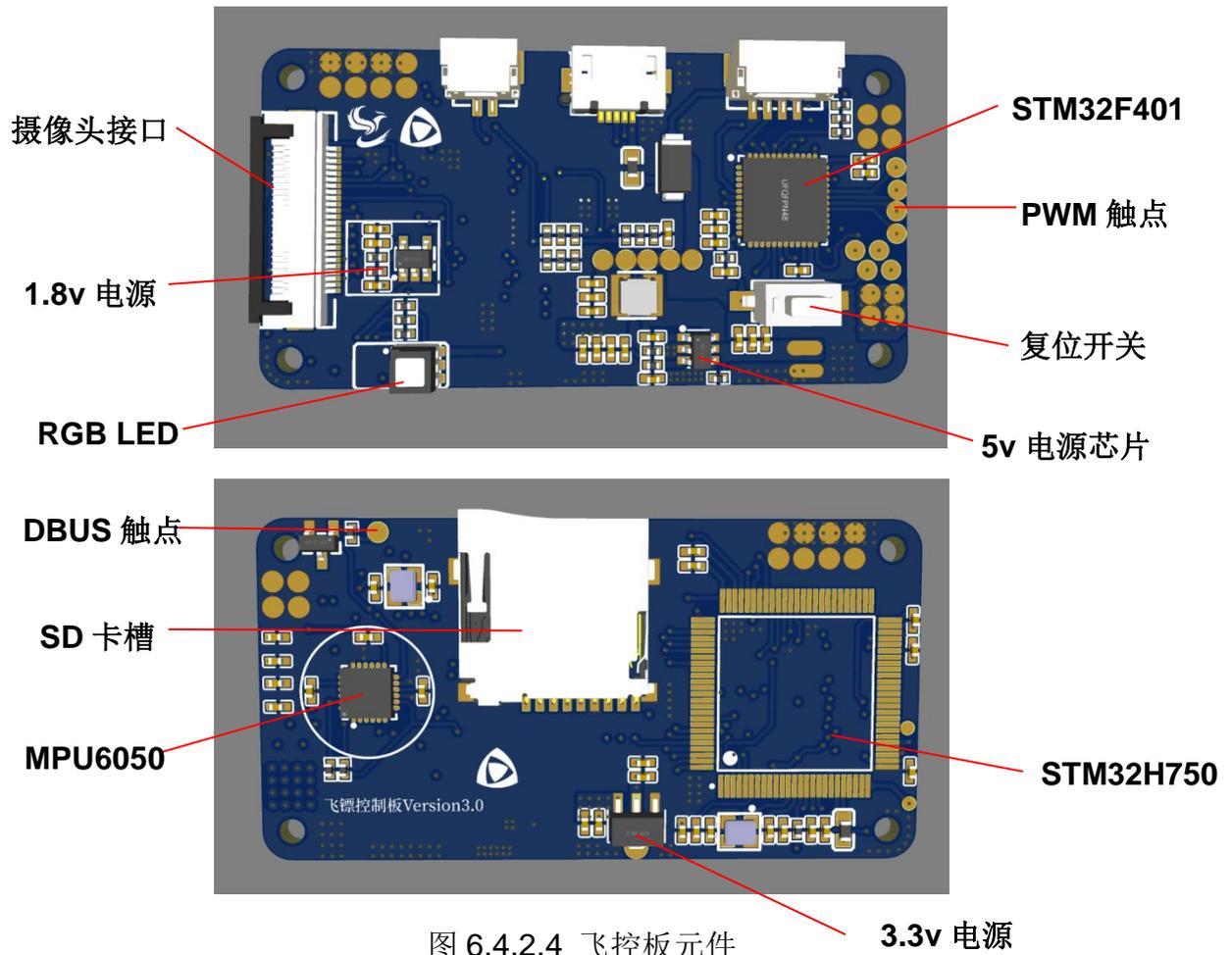


图 6.4.2.4 飞控板元件

同样为了节省成本，我们在飞镖上所使用的电池皆为前面所述 POS 机的拆机电池，为了满足有控飞镖的电源需求，我们采用了串联的方式将两个 3.7V，300mah 的电池组成了 2S 的小电池。因为我们是使用 TPU 作为镖体材料，在电池固定上我们给飞镖头加上了一层极薄的 TPU 膜用以固定飞镖电池，经验证该设计经济实用，相比其他固定方法能大幅降低重量，改进后的前段打印件如图 6.4.2.5 所示。

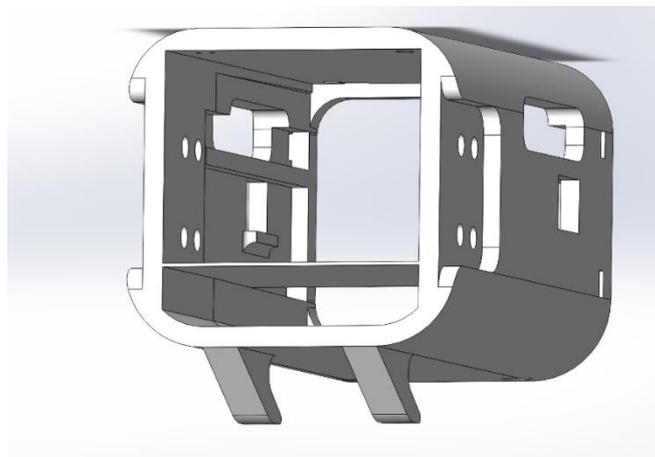


图 6.4.2.5 有控飞镖头电池仓（下部）

为了满足舵机、芯片、摄像头对电源的需求，我们使用了 PW2162、AMS1117、TPS73101DBVR 分别提供了 5V、3.3V、1.8V 电源，如图 6.4.2.6 所示。主控板原理图如图 6.4.2.7 所示。

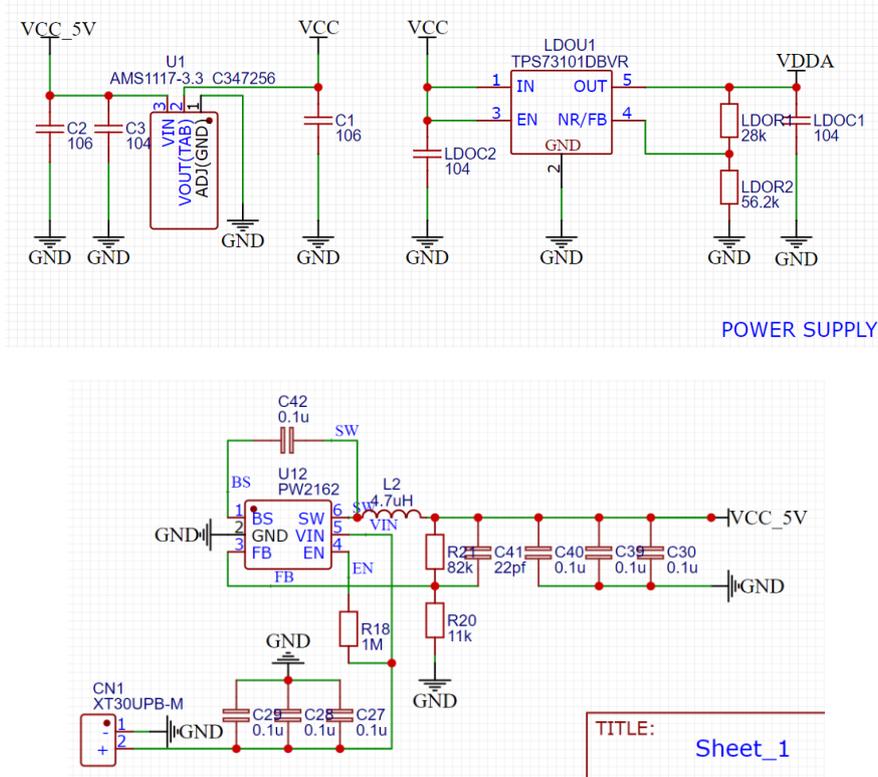


图 6.4.2.6 3.3V、1.8V 和 5v 电源

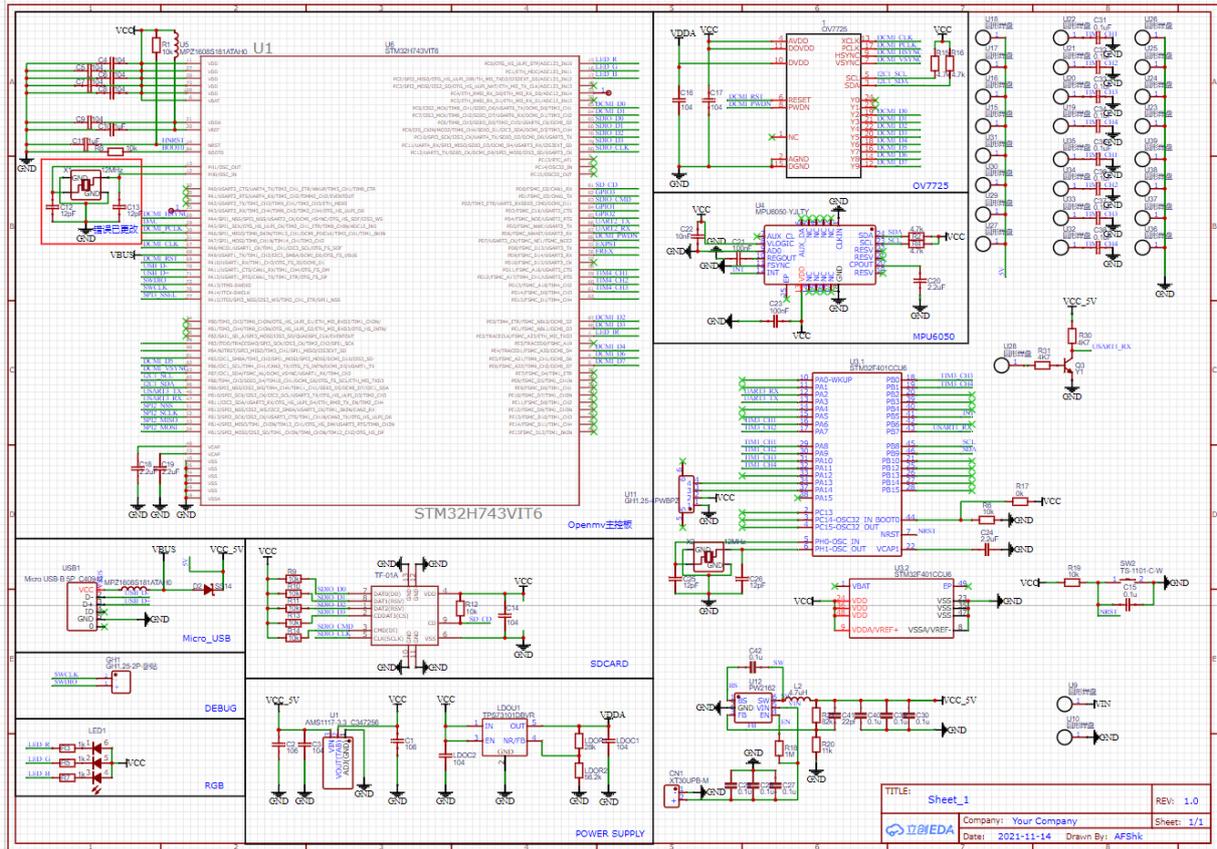


图 6.4.2.7 有控飞镖主控板原理图

### 5.4.2.3 无控飞镖硬件方案

和有控飞镖不同，无控飞镖仅需提供一路 5V 电源给飞镖触发装置供电，故我们选择了 PW5100 作为无控飞镖的供电芯片，将 3.7V 锂电池升压到 5V 给触发装置供电，分电板如图 6.4.2.8 所示。

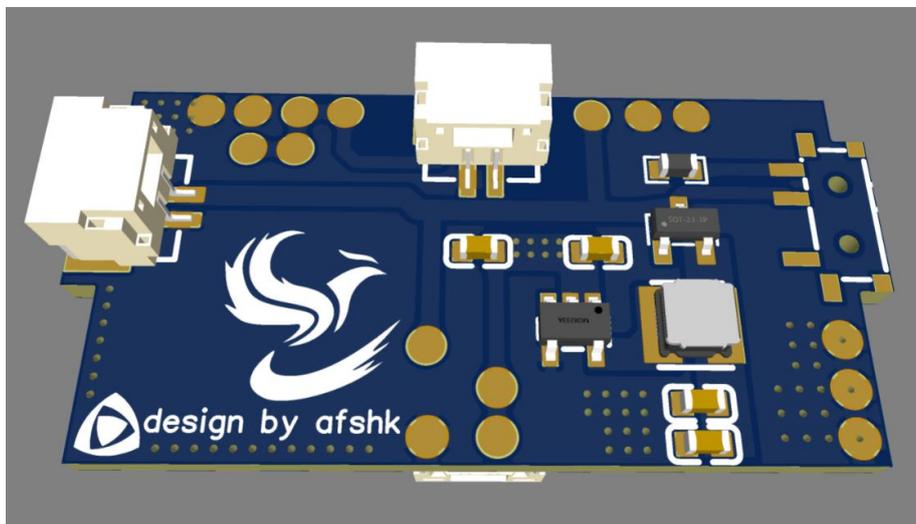


图 6.4.2.8 无控飞镖分电板

## 5.4.3 软件和算法设计

### 5.4.3.1 飞镖发射架软件算法方案

相对于 RM 赛场上的其他兵种，飞镖发射架的程序算法都比较简单。我们使用了 FreeRTOS 操作系统作为系统平台，飞镖架上的 YAW 轴电机、蓄能电机、装填电机和行程调节电机均采用双环 PID，经测试能达到需要的效果，下面给出飞镖发射架的程序结构，流程图如图 6.4.3.1 所示；

程序结构：

```
-DART_2022
  --Application/MDK-ARM
  --Application/User/Core
  --Drivers/STM32F4xx_HAL_Driver
  --Drivers/CMSIS
  --APP
-app_remote_control.c //接收机数据处理程序
-app_my_func.c //实现发射架功能：装填、拨动等，使用宏定义封装方便调用
-app_freertos.c //实时任务
--ALG
  -alg_pid.c //pid 算法
--BSP
  -bsp_oled.c //oled 驱动
  -bsp_can.c //can 过滤器配置及中断函数
  -bsp_rc.c //接收机程序
  -bsp_dl101.c //拉力传感器程序
--Middlewares/FreeRTOS
--CMSIS
```

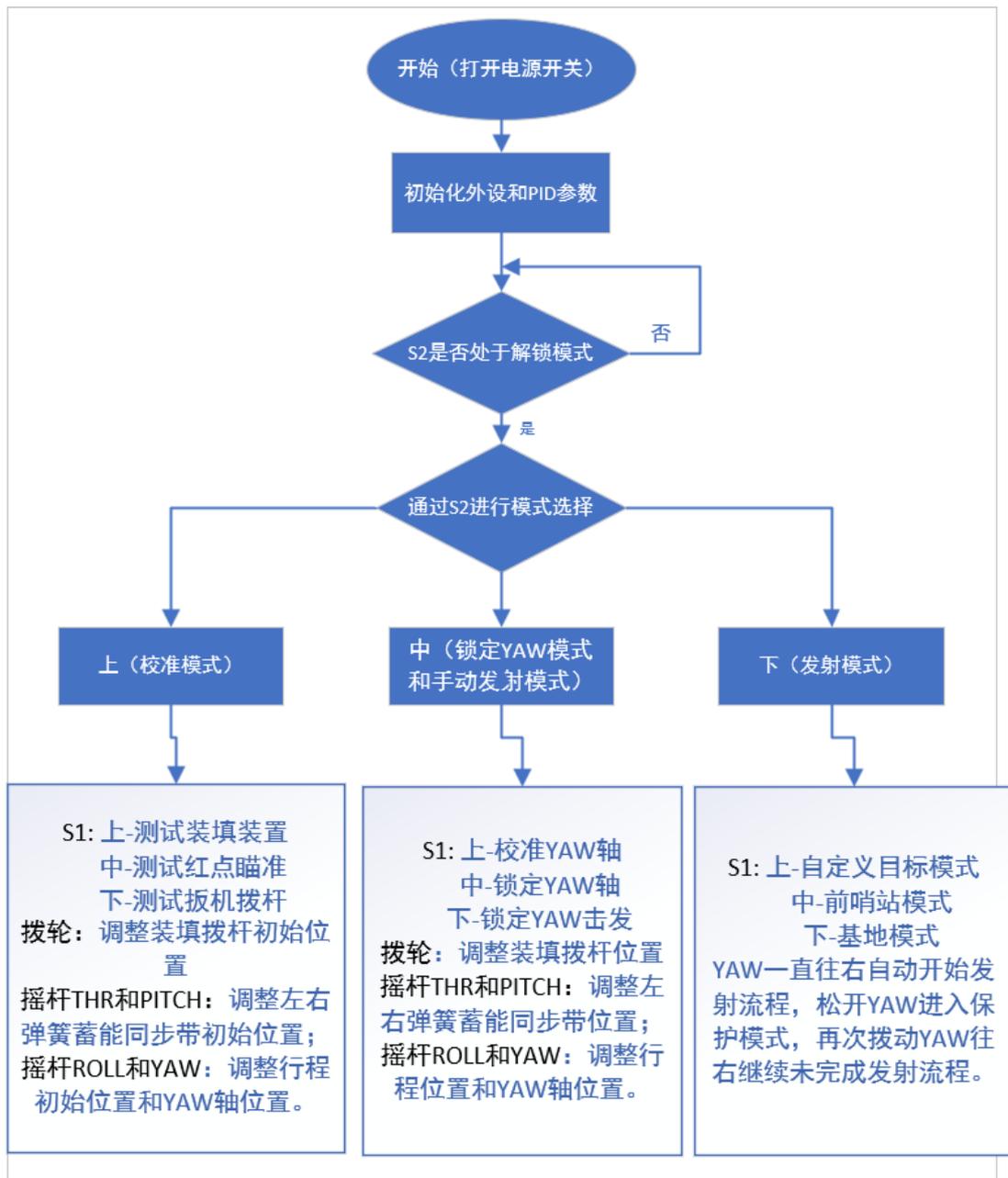


图 6.4.3.1 飞镖发射架的程序结构

对于飞镖发射架的发射程序，我们采用了分步执行，在每一步执行前都会检查摇杆是否为要求位置，以防止误触或者意外 **BUG** 造成人员伤亡和发射架损坏，最大程度保证飞镖比赛和调试时的安全性。对于发射架电机的 **PID** 计算和电流输出，我们使用了定时器去唤醒 **FreeRTOS** 任务，并为其分配了安全开关，既保证了电机控制的实时性，也保证了安全性。为了简化分布程序的书写，我们将常用函数用宏定义封装，让程序编写过程中简单易懂且可维护性强，例如：

```

if(flag==0){
    FUNC_PUT_SERVO_ON;    //抬起拨杆
  
```

```

    FUNC_SEND_OFF;           //关闭扳机

    FUNC_DART_RESET;        //装填机构复位

    M3508_LENGTH_RESET;     //复位蓄能机构

    MOTOR_DELAY;           //根据电机编码器延时

    flag++;

}

```

飞镖发射具有窗口期，为了让发射程序能够以最快速度运行，我们使用了电机编码器数据来延时切换下一步的时间，既能保证电机能运动了指定位置，也能让延时时间达到最短，提高发射时间效率。相关代码如下。

```

while(abs(motor_measure_chassis[0].code+M3508_length)>1000)
    osDelay(1);
while(abs(motor_measure_chassis[1].code-M3508_length)>1000)
    osDelay(1);

```

### 5.4.3.2 有控飞镖软件算法方案

有控飞镖的软件算法分为两部分：目标识别部分和姿态控制部分。在姿态控制部分我们使用 MPU6050 通过卡尔曼滤波算法得到飞镖的 ROLL 和 PITCH 轴角度，再通过主控控制舵机达到保持姿态的效果。对于目标识别我们采用 OV7725 和 OPENMV 平台，通过查找色块识别目标，在和主控通信，从而控制飞镖飞行的轨迹。OPENMV 识别目标的代码比较简单，通过大幅降低曝光时间能够实现较好的识别效果。

有控飞镖代码结构：

```

-AFSFC_2022

--Application/MDK-ARM

--Application/User/Core

--Drivers/STM32F4xx_HAL_Driver

--Drivers/CMSIS

--Middlewares/FreeRTOS

--APP

```

```

-app_tasks.c          //任务控制
-app_openmv_data.c    //openmv 数据处理
--ALG
    -alg_pid.c        //pid 算法
    -alg_kalman.c     //通过 kalman 滤波解算姿态
--BSP
    -bsp_mpu6050.c    //mpu6050 驱动
--CMSIS

```

OPENMV 通过串口与主控板控制，根据查找出色块的面积来决定飞镖的行为，从而大致达到对距离引导灯不同距离的轨迹的分段控制，图 6.4.3.2 为使用引导灯测试（未更换翼面），图 6.4.3.3 为数据传输路线。



图 6.4.3.2 飞镖识别效果与姿态控制测试

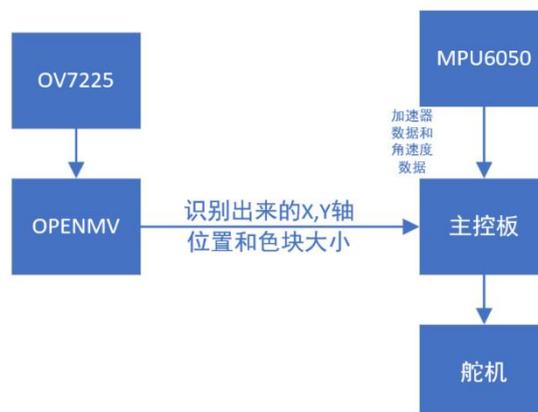


图 6.4.3.3 数据传输图

## 5.5 研发迭代过程（10）

### 5.5.1 版本迭代过程记录

版本号或阶段	功能或性能详细说明	完成时间
<b>MV1.0</b>	发射架可执行飞镖发射，左右瞄准，调整发射初速度	2022.1.9
<b>MV1.1</b>	发射架增加自动装弹功能，增强发射稳定性	2022.4.15
<b>SV1.0</b>	无控飞镖可容纳需搭载设备，稳定飞行	2022.5.1
<b>SV1.1</b>	改进工艺和结构，提高寿命，减少成本	2022.1.5
<b>SV1.2</b>	飞镖增加控制功能	2022.4.30
<b>YW0.1</b>	飞镖控制板完成	2021.12.1
<b>YW1.1</b>	飞镖控制板更换为四层板，测试全部功能正常	2022.1.15
<b>MRV0.0</b>	发射架完成发射程序测试	2021.12.15
<b>MRV0.1</b>	发射架完成装填及全自动发射程序测试	2022.4.25
<b>MRV0.2</b>	更新基地和前哨站两种发射程序及不同发射模式	2022.6.15
<b>RV0.1</b>	完成飞镖控制板传感器读取，板内通信，舵机输出测试	2021.12.30
<b>RV0.2</b>	更换飞镖控制板系统姿态解算算法完成测试	2022.3.1
<b>RV0.3</b>	完成有控飞镖的安装调试	2022.6.30
<b>RV0.4</b>	完成有控飞镖的发射测试	2022.8.10

## 5.5.2 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	机器人版本号或阶段	解决人员
1	弹射滑台与拦截机构命短	滑台将飞镖加速后自身的减速方案不合理,滑台拦截机构重量偏重,弹簧劲度系数偏大	将滑台拦截机构简化为单个滑块减轻重量,调整弹簧劲度系数,增加tpu缓冲块,将碰撞接触面移动至碳板与打印上,增加了滑块寿命	MV1.0	机械工程师:贺游
2	击发机构与装弹同步带滑块重力形变干涉,装弹时产生碰撞	飞镖击发机构所需力较大,使舵机尺寸偏大,安装位置尺寸限制	将接触面设置为斜面,减少张紧力产生的力矩;更改击发结构原理,下移舵机位置,避开干涉位置	MV1.0	机械工程师:贺游

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
3	飞镖鸭翼操纵面寿命短	鸭翼在着陆时先接触地面,作为操纵面强度低,容易破坏	将飞镖布局变为传统布局,先触地的主翼为贯穿木板,将力分散传导至机身,抗冲击能力大幅提高	SV1.0	机械工程师:贺游
4	横向装弹机构工作可靠性低	飞镖在装至发射位置时,路径会有突起螺丝,连杆等可能的碰撞结构,一旦发生碰撞无发纠偏	重新设计为纵向装弹机构,在上层滑道存放,通过平行四边形机构将上层飞镖转移至下层滑道,利用重力滑动至指定位置	MV1.1	机械工程师:贺游

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
5	飞镖操纵舵机力矩不足	未考虑连接产生的阻力，第一版舵机安装空间较小，不足以放置更大尺寸的舵机	重构飞镖结构，将四个舵机布置于两侧，实验不同重量的舵机达到可用的效果	SV1.1	机械工程师：贺游
6	飞镖不便于安装和调试	分离面设计不合理，拆装飞镖需要拆装舵面连杆	重新设计分离面和内部结构，使重量轻且强度高	SV1.2	机械工程师：贺游
7	视觉板和控制板通信工作不正常	布线出现失误，引发错误中断。	重新设计电路板，完全解决问题	YV0.0	电控工程师：胡成成
8	陀螺仪读取芯片和数据不正常	芯片为淘宝磨片山寨芯片。	更换在立创商城买的“高价”芯片后解决	YV0.1	电控工程师：胡成成
9	飞镖控制板和飞镖标体不适配	飞镖主控板使用两层板导致体积比较大	使用四层板后减少体积后解决	YV0.2	电控工程师：胡成成

序号	问题描述	问题产生原因	问题解决方案&实际解决效果	机器人版本号或阶段	解决人员
10	飞镖摄像头固定不牢	摄像头太小 原有结构无法完全固定 牢固	使用新的固定结构后解决	YV1.0	电控工程师：胡成成
11	系统运行卡死	原有工程生成时出现错误	重新配置工程后解决	RV0.1	电控工程师：胡成成
12	舵机响应频率不够	飞镖飞行很快，原有舵机响应频率不够	更换 333HZ 舵机后解决	YV1.1	电控工程师：胡成成
13	原有姿态解算不准	飞镖控制板上没有磁力计，不能使用原有算法解算姿态	更换姿态解算算法后解决	RV0.2	电控工程师：胡成成

## 5.6 团队成员贡献（2.5）

姓名	基本信息 (专业、年级、队内角色)	主要负责工作内容描述	贡献度 (所有成员贡献度合计为 100%)
贺游	飞行器设计与工程、大四、机械设计负责人	负责整个机器人的机械设计，包括飞镖发射架方案设计、详细机械结构设计，飞镖总体、结构、气动设计	50%
胡成成	飞行器工程与信息工程	负责整个机器人的电控部分，包括飞镖架的自动装填发射程序，校准程序等，飞镖的控制程序。	50%

## 5.7 参考文献（2.5）

参考文献	收获点分析
飞行器结构力学（史治宇）	板杆薄壁结构的受力分析，结构应力与变形的计算方法，优化设计传力路径，做出坚固轻盈的结构，所有结构都适用
飞机飞行动力学自编讲义（陈永亮）	了解飞行器在大气中运动规律，以及外形参数对气动特性的影响，对飞镖构型设计和优化有巨大作用
复合材料力学（矫桂琼）	连续纤维增强复合材料刚度强度、应力应变的分析方法，对各向异性材料设计有感性认识，设计中进行优化
空气动力学（陆志良）	流体力学基础和飞行器空气动力学，学习流体的基本控制方程，升力阻力的近似计算，了解如何设计翼面和提高升阻比

微型飞行器设计导论（昂海松）	了解低雷诺数空气动力学和非定常空气动力学的特点，对一些估算结果进行优化，布局设计结构设计和总体优化
飞机结构设计（姚卫星）	对飞镖的结构设计细节启发很大，帮助满足飞镖的气动、重量、工艺、维护的要求，比如在轻量的结构中设计出可靠的结构，开口补强、优化传力路径、扩散集中力等
飞机总体设计基本原理（余雄庆）	涉及知识面广，工程实践性强，飞镖设计参考了布局形式、翼载荷等基本参数估算、重心分析、飞行性能、气动特性分析等参数分析
多旋翼飞行器设计与控制（全权）	涉及无人机飞行器的控制及姿态解算部分知识。
导弹总体设计导论（刘新建）	涉及飞镖系统弹道规划与制导，姿态控制设计等知识点。

## 6. 雷达

### 6.1 其它学校雷达站分析综述（2.5）

雷达站本赛季参与的官方技术交流活动共有：2022 超级对抗赛中部分区赛（适应性训练 1 场，小组赛 3 场，16 强 1 场，8 强 1 场），2022 单项赛飞镖打靶。在备赛及参赛过程中，一共调研了 6 支队伍的雷达站情况。分析见下表：

队伍	硬件方案	功能实现	值得借鉴之处
东南大学	3 工业相机+mid70 激光雷达	基于计算机视觉的小地图，预警	未了解实际效果，暂无借鉴之处
南航金城	无雷达		暂无
太原工业学院	监控摄像头+mid70 激光雷达	场地画面	使用监控摄像头提供画面，更清楚，有利于云台手观测
华南理工大学	观赛限制，结构未知	基于计算机视觉的小地图，血量显示，预警	雷达站 UI 画面美观，排布清晰
上海交通大学（2021 开源方案）	3 工业相机+mid70 激光雷达	基于计算机视觉的小地图，预警，自动决策	小地图定位算法精度高，神经网络识别车辆准确度高；具备自动决策功能
国防科技大学	观赛限制，结构未知	基于计算机视觉的小地图，血量显示	雷达站 UI 画面美观，排布清晰

## 6.2 机器人功能定义（5）



图2-1 机器人功能定义

## 6.3 机器人核心参数（2.5）

表 1.1: 雷达站主要技术参数

主要参数	数据
重量（含线缆）	5.7kg
总体尺寸（长*宽*高）	700mm*700mm*1450mm
激光雷达数量	1
相机数量	3

表 1.2: 雷达站传感器类型

传感器	型号
左右相机	MV-SUA134GC-T 8mm 1:2.0 1/1.8"

上相机	Realsense D435i
激光雷达	Livox Mid70

## 6.4 设计方案（25）

### 6.4.1 机械结构设计

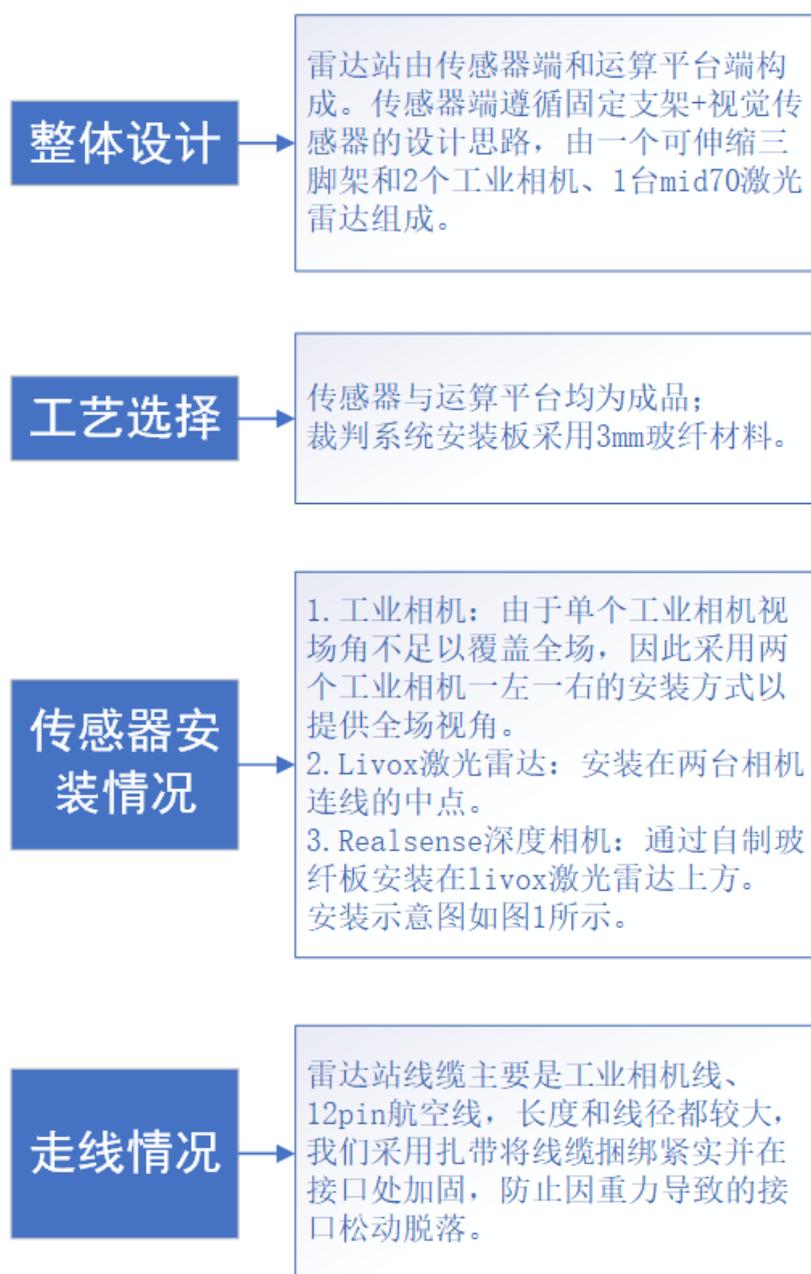


图2-2 机械设计方案

## 6.4.2 硬件设计

雷达站的硬件结构主要分为支架、传感器和配件三部分。

### 支架

我们使用了普通的三脚架作为雷达站支架，同时使用了配套的滑轨和球头云台，使得相机可以在一定范围内自由转动。



图2-3 雷达云台和滑轨

### 传感器

我们一共使用了两个工业相机和一个激光雷达作为主要的传感器，两个相机互相补充视野。同时为了使云台手可以看到完整的画面，我们额外增加了一个广角相机。



图2-4 雷达传感器

## 配件

诸多有趣、实用的小配件也是我们雷达站的一个小特色。

为了解决主控和电源管理模块和运算端的固定问题，我们设计了一个支座，可以将裁判系统和电池都固定上去，提高了检录和上场时的通行速度。

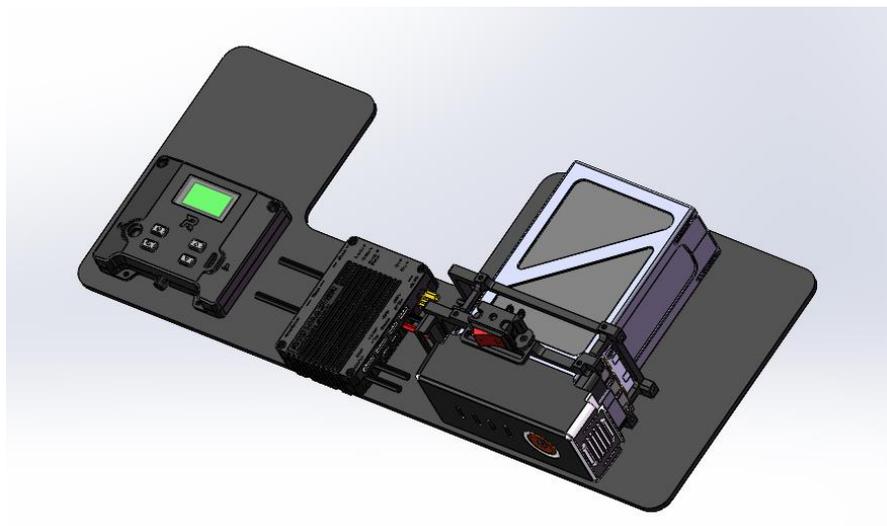


图2-5 裁判系统底座

工业相机和激光使用 M3 螺栓进行固定，而支架上是标准相机螺栓，同时广角相机需要固定到激光雷达上方。于是我们制作了转接板。既解决了固定的问题，又保留了调节角度的功能。

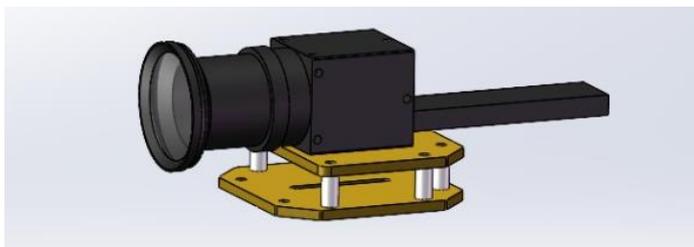


图2-6 工业相机转接板



图2-7 Realsense 转接板

虽然我们使用了两个相机补充视野，但是视场角仍然没有很大的余度，使得我们对上场时雷达的摆放角度要求较高。为了方便在紧张情况下快速将雷达摆放稳妥，我们在雷达上增加了红点瞄准。如图所示：

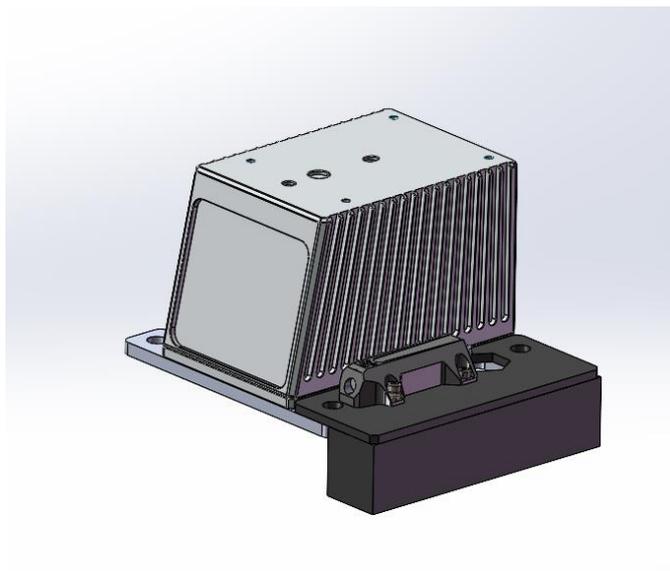


图2-8 红点瞄准器

## 6.4.3 软件设计

### 6.4.3.1 系统架构

#### 系统层级：

整个雷达站采用 ROS（机器人操作系统）开发，代码遵循 ROS 架构。分为 4 层：硬件驱动层，数据处理层，功能逻辑层和前端显示层。

## 第三方中间件及模块:

CUDA\_10.2, TensorRT, Tensorrtx, yolov5-6.0, OpenCV-4.5.4, Cmake

另一些驱动程序, 以 ROS 功能包的形式运行: Livox\_ROS\_Driver, MindVision\_ROS\_Driver, RealSense\_ROS\_Driver, ROS\_Serial

## 开发及运行环境:

Ubuntu20.04(操作系统), CLion(软件开发环境)

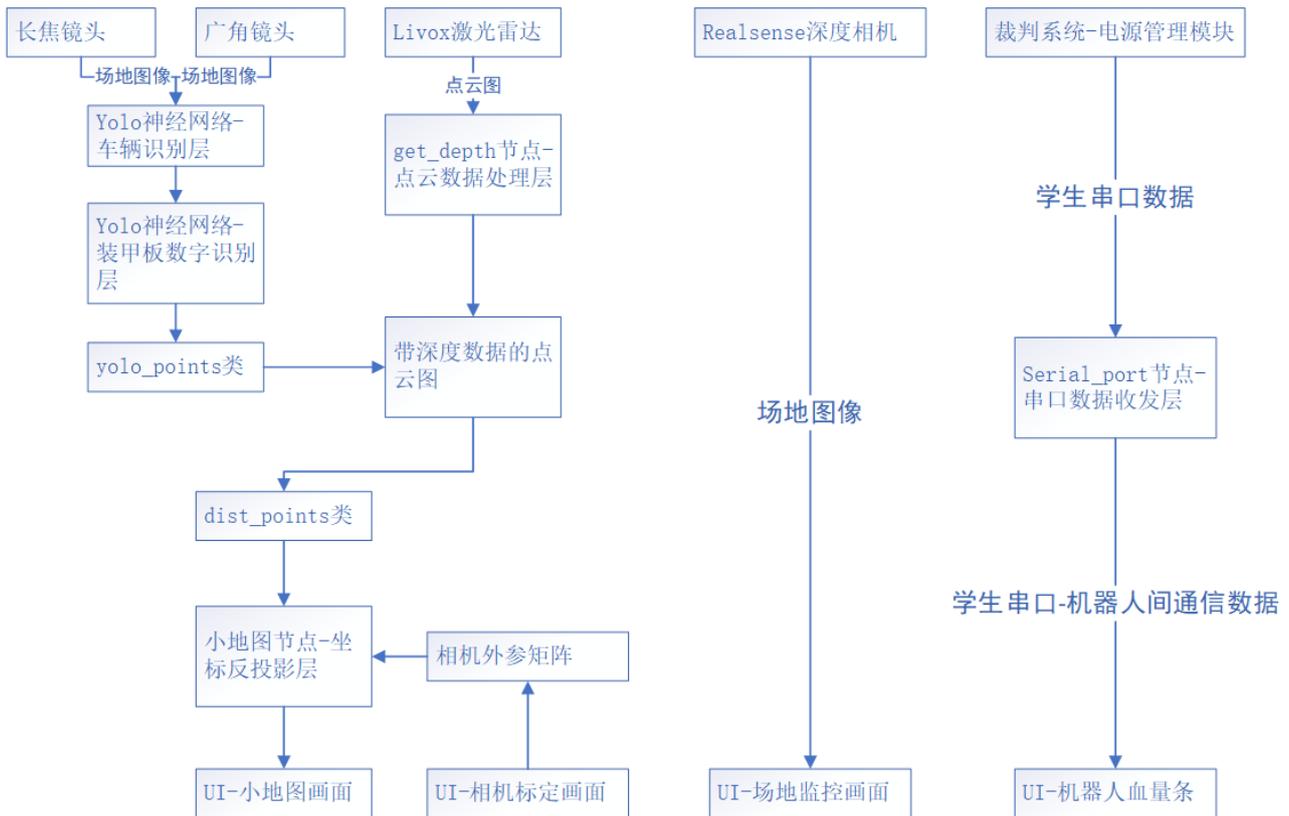


图2-9 运行流程图

### 6.4.3.2 重点功能

#### 小地图功能

小地图功能测算出场地上车辆的世界坐标, 并绘制在场地俯视平面图上。处理过程见运行流程图(图), 在这个功能中, 我们使用两层深度神经网络对车辆及其兵种类别进行了识别, 神经网络使用 yolov5-6.0, 在自制数据集上进行训练。使用 PCL 点云库对 Livox 激光雷达获取的场地点云图进行处理, 得到每个点距离雷达的距离数据。使用事先标定好的雷达与相机之间的旋转平移矩阵将车辆识别框投影至点云图中, 对每个投影框内部的点的深度求取距离

统计值（依照点个数多少分别取不同的统计值），得到该车辆相对雷达站相机的距离，将距离和车辆在相机画面中的二维坐标打包发送至小地图节点。在小地图节点，利用赛前 3 分钟之内标定好的相机与场地外参矩阵及含深度的二维坐标，使用公式 1 求解出车辆的世界坐标，完成车辆世界坐标的解算，并通过 opencv 绘制在 ui 中。

## 哨兵辅助自瞄

在小地图功能的基础上，选取每次小地图解算结果中的己方哨兵世界坐标  $Og$  和所有敌方车辆坐标  $E1, E2, \dots, En$ ，计算出以己方哨兵为原点的相对坐标  $OE1, OE2, \dots, OEn$ 。将相对坐标发送给哨兵，哨兵通过姿态解算得到应朝目标转动的云台  $pitch, roll, yaw$  角度，完成辅助自瞄。

## 英雄前哨站测距

在赛前 3 分钟内，圈出前哨站位置并获取其二维坐标。使用与小地图相同的功能测算出前哨站的世界坐标。在小地图解算结果中选取己方英雄的世界坐标，计算前哨站与英雄的欧氏距离。

## 敌方车辆实时预警

将每次小地图解算结果中的敌方车辆坐标加以判别，若其出现在己方半场的危险区域（程序中定义），则在小地图中将对应区域置为红色高亮显示，并在 ui 消息框中打印预警消息。

## 标定

精准的标定是雷达站能够准确定位的基础，为了使标定更加准确，我们在赛季初进行了很多的尝试。最终使用了如下的方法。

### 相机内参标定

我们使用了 ROS 下的 camera\_calibration 包标定相机内参。相比于 OpenCV 或者 Matlab 中的标注方式，camera\_calibration 的显著优点是自动化程度极高。通过进度条指引标图者调整标定板的角度的，并且无需手动拍照，短时间内便可以自动拍摄和解算大量的图片。操作的方便间接的提高了标定的精度。通过这种方式便可以获得相机较为准确的内参。

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

## 联合标定

首先，我们使用了 livox 官方提供的激光雷达标定方式（[https://github.com/Livox-SDK/livox\\_camera\\_lidar\\_calibration](https://github.com/Livox-SDK/livox_camera_lidar_calibration)）进行手动标定。在拍摄了多角度的图片并录制点云 rosbag 后，分别使用鼠标在点云图和彩色图中按照相同的顺序点出四个角点，通过 PNP 解算即可获得旋转和平移矩阵。

由于为手动标注，工作的繁琐注定了图片和点云图的数量不会太多，同时由于手动选点的误差，使得得到的参数注定不会十分精确。所以还需要想办法进一步提高联合标定的精度。

我们使用了香港大学开源的无目标场景下激光雷达和相机自动标定（[https://github.com/hku-mars/livox\\_camera\\_calib](https://github.com/hku-mars/livox_camera_calib)）的方法。该方法可以自动提取深度图和彩色图中共同特征，并通过迭代的方式，使得误差达到最小。通过这种方式，我们进一步提高了联合标定的精度。

## 串口通信

串口通讯是雷达站发挥作用的重要窗口，通过官方的小地图通信和车间通信，以及裁判系统下发的其他消息，可以大大提高雷达站这个兵种的上限，实现出奇制胜。

我们使用了 USB 转 TTL 模块实现雷达站运算端和裁判系统的通信。使用了 ROS 的 serial 包实现。

### 串口接收

我们从裁判系统中读取了机器人血量、比赛状态、比赛结果、补给站状态、场地机关状态、裁判判罚、飞镖闸门关闭倒计时等数据。程序主循环不断试图从串口中读取数据，当存在可用数据时，便将其读入数组中。从串口中读入可用数据后，依次使用所有消息类型进行检验，通过 CRC 校验判断是否匹配。如果匹配，则为该类型消息；否则，则使用其他类型进行检验。

### 小地图消息

小地图基于官方的通信协议实现。消息以 10Hz 定频发送。来自小地图节点的坐标信息被存放在数组中，依次向裁判系统发送。当数组被清空后，重新从小地图节点获取新的坐标信息。以此来实现小地图通信频率和解算频率的匹配。

### 其他消息发送

我们使用了车间交互消息，来实现雷达赋能其他兵种。消息的发送策略与小地图类似。消息定频发送，当没有交互信息需要发送时，则发送约定好的空消息。

## 6.4.4 算法设计

### 6.4.4.1 重要算法原理阐述、公式推导

#### 小地图反投影算法原理

小地图算法的流程在 1.4.3 中已提到，其可以更抽象地概括为：世界坐标系中一点的坐标向相机坐标系中投影（正投影过程）+投影点使用深度  $d$  和内外参矩阵的逆  $\text{inv}[R,T]$ ,  $\text{inv}M$  向世界坐标系反投影的过程（反投影过程）。其中正投影过程已经在相机的成像过程中完成，而反投影过程则需我们编写算法，使其在运算平台端的计算中完成。这其中亦包括对深度  $d$ 、 $\text{inv}[R,T]$ 、 $\text{inv}M$  的获取和计算，我们将这三个参数称为投影参数。因此，小地图算法的核心步骤即为两个：1.投影过程。2.投影参数的获取及计算。

下面，我分别介绍这两个核心步骤的原理及实现。

1.正投影过程和反投影过程有着数学上互逆的关系，只要理解了正投影过程的公式表示，就可以推出反投影过程。这里以正投影过程为例来讲解。正投影的过程可概括为：世界坐标系->相机坐标系->成像平面坐标系->像素坐标系。

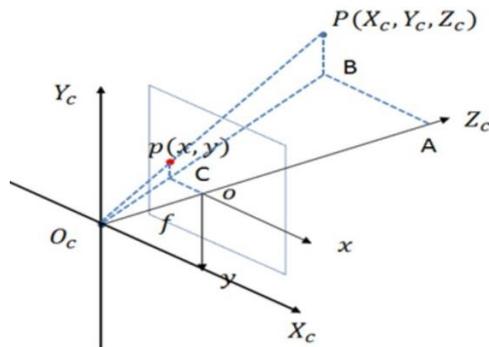


图2-10 小孔成像相机投影示意图

如图 2-1 所示为一个小孔成像相机模型。相机坐标系  $X_c Y_c Z_c$ ，成像平面  $uv$ ，相机坐标系的  $Z_c$  轴与相机光轴重合。有真实世界中一点  $P$ ，其在相机坐标系下的坐标为  $(X_c, Y_c, Z_c)$ ，则根据小孔成像原理， $P$  点与相机坐标系原点  $F_c$  的连线与成像平面坐标系  $xy$  交于点  $P'$ （坐标为  $(x, y)$ ），此即为真实世界中的点  $P$  在相机成像平面上的二维投影点  $P'$ 。由几何学关系可得：

$$\frac{O_c O}{Z_c} = \frac{OC}{BA} \Rightarrow \frac{f}{Z_c} = \frac{x}{X_c} \Rightarrow x Z_c = f X_c \quad (1.1)$$

同理可得

$$yZ_c = fY_c \quad (1.2)$$

将其转化为矩阵关系式，即为

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (1.3)$$

之后，我们将  $P'$  从成像平面坐标系  $xy$  转化至像素坐标系  $uv$ ，需对  $x$  轴和  $y$  轴分别乘以每毫米像素点数(dpm)，并加上成像平面中心点相对  $u$  轴和  $v$  轴的偏移量  $u_0$  和  $v_0$ 。即

$$p(u, v) = p\left(\frac{x}{dx} + u_0, \frac{y}{dy} + v_0\right) \quad (1.4)$$

其中  $1/dx$  和  $1/dy$  分别为  $x$  轴每毫米像素点数、 $y$  轴每毫米像素点数。

将公式 1.4 转化为矩阵乘法形式并与 1.3 合并可得

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (1.5)$$

其中  $f_x=f/dx$ ,  $f_y=f/dy$ . 至此我们完成了相机坐标系->成像平面坐标系->像素坐标系之间的转换。为方便将公式 1.5 中的系数矩阵记为相机内参矩阵  $M$ 。

现在，我们假设世界坐标系为原点在图 2-2 的左下角，向右为  $x$  轴，向上为  $y$  轴，向屏幕外为  $z$  轴。此世界坐标系记作  $X_wY_wZ_w$ 。



图2-11 赛场俯视图

根据空间直角坐标系之间的线性变换关系， $X_w Y_w Z_w$  到  $X_c Y_c Z_c$  的变换关系为

$$\begin{aligned} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} &= R \begin{pmatrix} x_{\text{world}} \\ y_{\text{world}} \\ z_{\text{world}} \end{pmatrix} + T = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x_{\text{world}} \\ y_{\text{world}} \\ z_{\text{world}} \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \\ &= \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \end{aligned} \quad (1.6)$$

其中， $R$  记作两个坐标系之间的旋转矩阵， $t$  记作平移矩阵。

综合公式 1.5 和 1.6，我们可以写出世界坐标系到像素坐标系之间的变换关系

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \cdot \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (1.7)$$

公式 1.7 即为世界坐标系到像素坐标系之间的转换关系，亦即正投影过程。

由线性代数理论推导，可得由世界坐标到像素坐标之间的转换关系（即反投影过程）为

$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \text{inv}R \cdot \left( \text{inv}M \cdot Z_c \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - t \right) \quad (1.8)$$

其中  $\text{inv}R$  是  $R$  的逆矩阵， $\text{inv}M$  是  $M$  的逆矩阵， $Z_c$  是点  $P$  在相机坐标系  $Z_c$  轴的截距，其近似等于通过激光雷达所求得点  $P$  距离雷达光源点的深度  $d$ 。在实际的程序代码中，我们使用深度  $d$  代替  $Z_c$ ，解算精度达到要求。

$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \text{inv}R \cdot \left( \text{inv}M \cdot d \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - t \right) \quad (1.9)$$

## 2. 投影参数的获取与计算

从公式 1.9 中可见须获得的参数是  $\text{inv}R$ 、 $t$ 、 $\text{inv}M$ 、 $d$ 。下面分别介绍他们的获取方法。

(1) 相机的内参矩阵  $M$  我们由棋盘格标定法获得。求解其逆矩阵  $\text{inv}M$ 。

(2) 旋转和平移矩阵  $R$ 、 $t$  由我们在赛前 3 分钟内使用 `solvePnP` 方法临场标定获得。我们使用经典的四点法，事先选取好赛场上的 4 个标定点，根据规则手册计算出以式 1.9 中  $X_w Y_w Z_w$  为基准的世界坐标，在赛前 3 分钟内通过 `ui` 选点操作获取其像素坐标，输入 `OpenCV` 库的 `cv::solvePnP` 函数并计算出旋转和平移矩阵  $R$ 、 $t$ 。通过 `cv::Rodrigues` 将  $R$ 、 $t$  转化为罗德里格斯形式，并求解其逆矩阵  $\text{inv}R$ 。

(3) 使用 `PCL` 点云库对 `Livox` 激光雷达获取的场地点云图进行处理，得到每个点距离雷达的距离数据。使用事先标定好的雷达与相机之间的旋转平移矩阵将车辆识别框投影至点云图中，对每个投影框内部的点的深度求取距离统计值（依照点个数多少分别取不同的统计值），得到该车辆相对雷达站相机的距离  $d$ ，将距离  $d$  和车辆在相机画面中的二维坐标打包发送至小地图节点。

## 车辆目标检测及兵种识别算法原理

在相机节点获取到图像之后，图像被送入神经网络节点进行推理。

由于官方的小地图通信频率只有 10Hz，对于系统的帧率要求并不高；所以我们采用了双神经网络的设计。通过这种方式，解决了在雷达站全画幅画面中装甲板尺度过小，场地中干扰过多的问题，同时也使得数据集标注更加方便。

我们使用的数据集为 `DJI ROCO` 数据集、`RMCV` 视觉开源数据站 (<https://rmcv.52pika.cn>) 中的数据集以及自行制作的数据集。数据集规模为 15000 张左右（识别车辆）和 3000 张左右（识别装甲板）。



图2-12 装甲板数据集示意

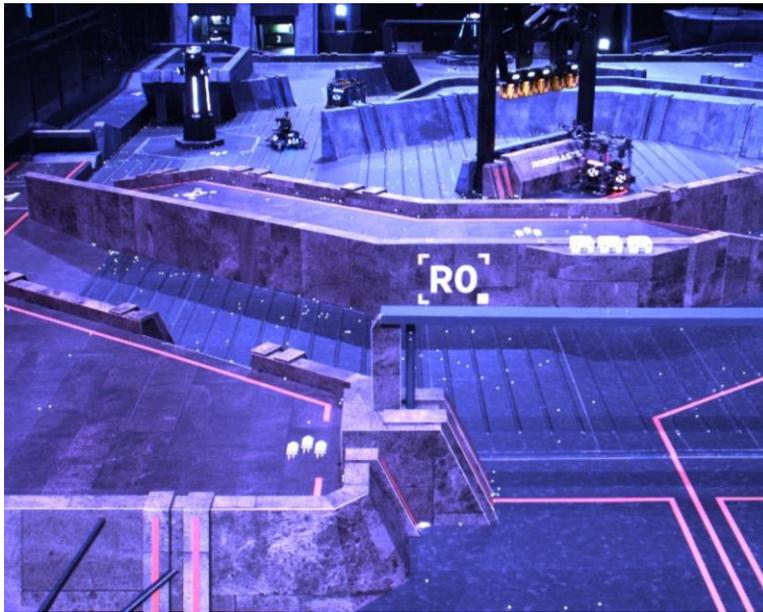


图2-13 车辆数据集示意

并非所有的数据集均为手动标注。在标注时，我们仅先手动标注一小部分，进行训练；然后用所得模型推理新的数据集，然后对数据集进行校对；重新训练，获得新的模型；增加新的数据集……如此，仅需手动标注少量图片（其余需要校对），便可以短时间内标注大量图片。通过这样的方式，标图的效率大大提高。

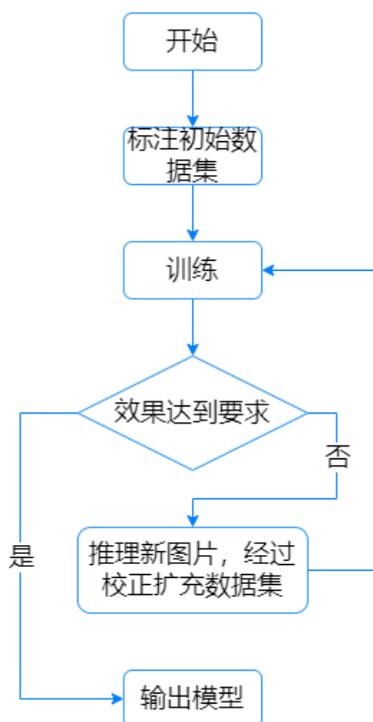


图2-14 数据集标注流程图

我们的内外两层神经网络均使用的是 YOLOv5s 神经网络，并使用了 TensorRT 加速，其中第一层的输入尺寸为  $640 \times 640$ ，第二层为  $128 \times 128$ 。第一层的主要任务是从场地中识别到蓝方车辆和红方车辆，第二层的主要任务是通过识别装甲板数字判断兵种。

具体流程如下。从相机节点获取到图片之后，对于每一张图片进行推理，获得车辆的 `bbox`，并获取与之对应的图片 `ROI`，送入第二层神经网络，以获得每个车辆的 `ID`。考虑到第一层神经网络获得的 `ROI` 在大部分情况下并不唯一，因此我们将多个 `ROI` 合为一个 `batch` 送入第二层。经过多次尝试，我们最终选择以 8 个 `ROI` 为一个 `batch`，使得在大多数场景下，一个 `batch` 可以包含所有的 `ROI`，同时在 `ROI` 较少的情况下也能保持较高的效率。通过这种操作，第二层神经网络对整个系统的影响大大减小。经测试，`batch size` 为 8 时第二层的运行速度，比 `batch size` 为 1 时的速度，在大多数场景下快一倍以上（推理相同数量的 `ROI`）。第二层神经网络的存在，仅仅使得推理时间增加了 15% 左右。

当第二层神经网络只识别到一个装甲板时，则直接输出；当识别到不止一个装甲板时，则选取置信度最高的装甲板输出；当没有识别到装甲板时，则将 `ROI` 缩小一定尺寸（减少车辆本身的镂空造成 `Z` 值偏大和车辆被场地遮挡造成的 `Z` 值偏小），作为结果输出，此时仅能分辨敌方和我方车辆。

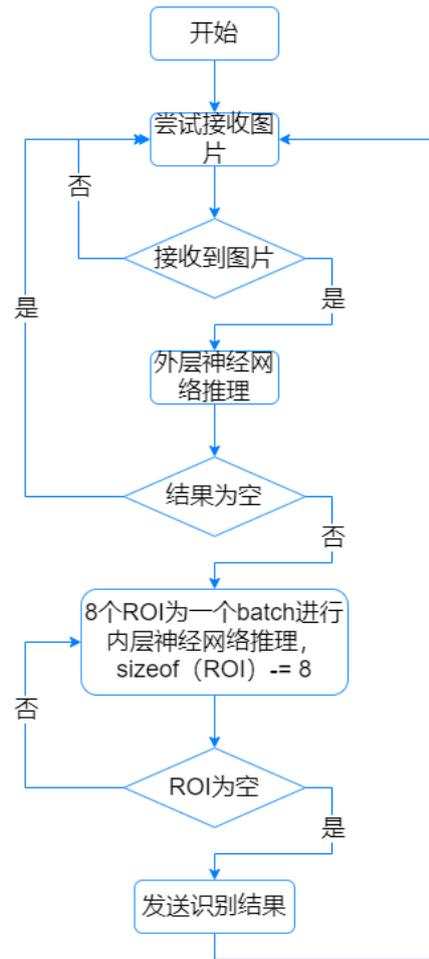


图2-15 神经网络节点运行流程图

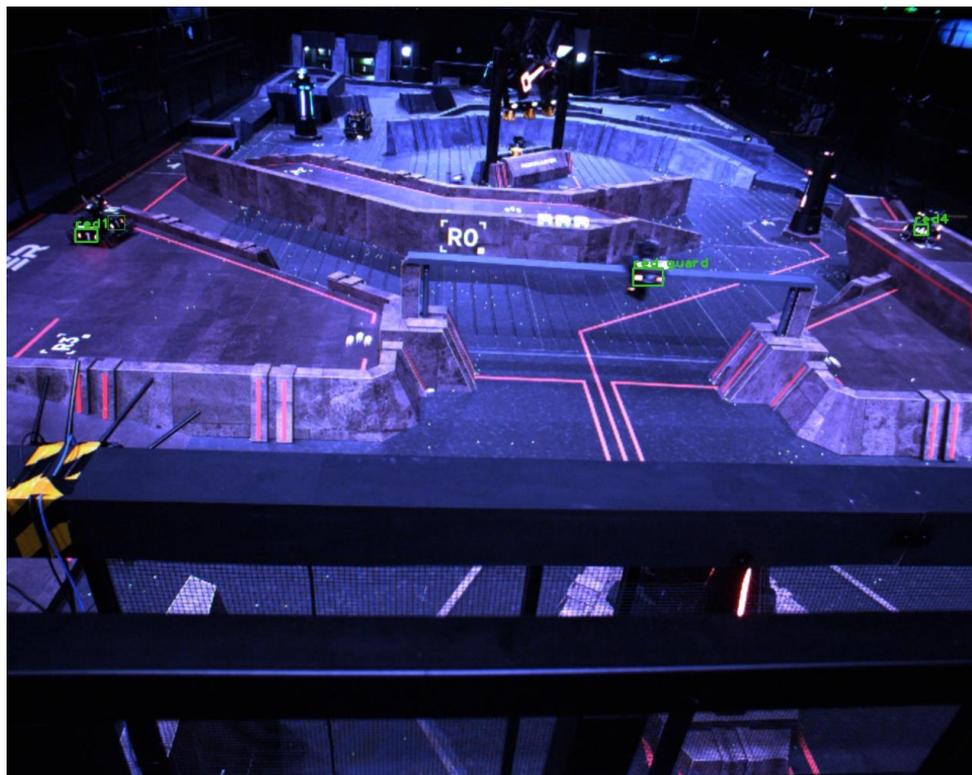


图2-16 双层神经网络识别效果

神经网络仅仅对所有地面单位和哨兵进行了识别，而且由于数据集有限，对于装甲板的识别准确率有限。这是目前神经网络节点存在的主要问题。

#### 6.4.4.2 性能分析及优化方案

与完整形态相比，主要改进有：增加了云台手操作端图形化界面（UI），完成了小地图功能及其派生功能：哨兵辅助自瞄、英雄与前哨站测距、敌方车辆预警。

UI 性能分析及优化方案：我们的 UI 起初在 Ubuntu18.04 系统上进行开发，试运行过程中，发现 UI 启动 30 秒左右之后就会逐渐卡顿直至卡死。经过 Perf 工具的分析我们发现，雷达站 ui 显示、神经网络推理和激光雷达点云数据处理这三个节点的 CPU 占用率最高，其中 ui 显示在三者中最高。我们尝试优化 Qt 程序中的图像显示函数，但效果不明显。最终，我们将雷达站程序迁移至 Ubuntu20.04 系统中重新运行，在一行代码未改的情况下，消除了卡死问题。对于这个问题的成因以及迁移系统即解决问题的内在原理，我们还在探索。

小地图性能分析及优化方案：在精度方面，解算的车辆世界坐标精度在 2m 以内。受制于神经网络识别准确度的限制，当出现误识别情况时，会导致小地图画面中车辆点的颜色和数字绘制错误；同时，由于投影参数  $d$  的误差甚至错误也是影响精度的重要原因。针对此问题，我们尝试记录本次识别结果与上一帧识别结果，将二者进行综合比对，消除识别框中的干扰

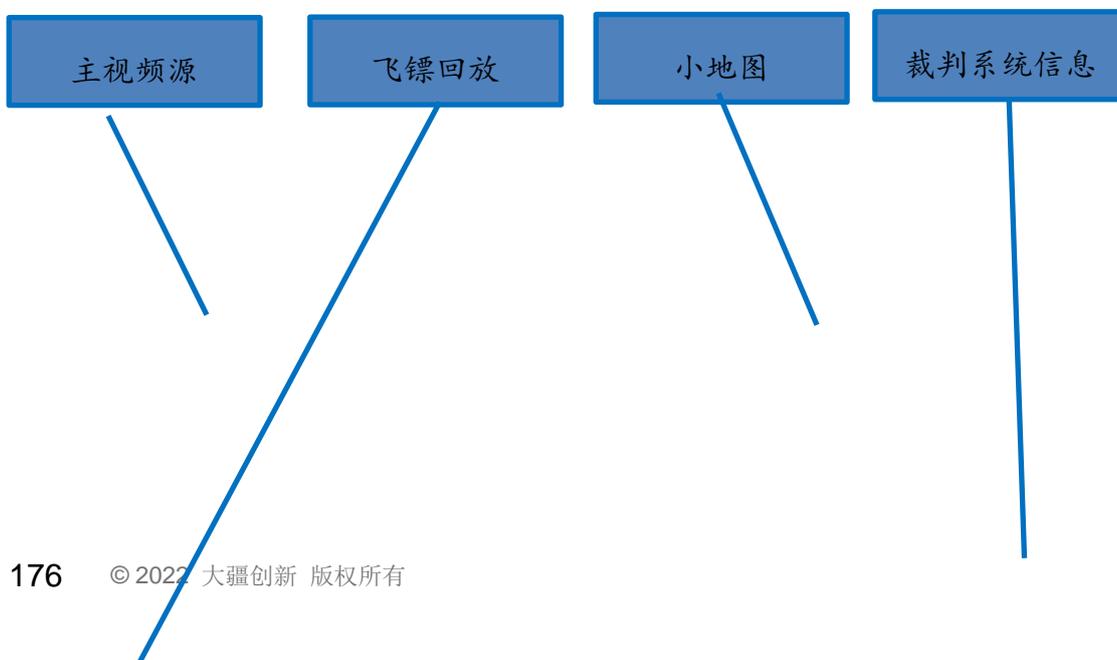
点，以获得正确的深度  $d$ 。由于时间有限，该优化方案尚处在开发测试过程中。在运行速度方面，小地图在每秒内可更新 10 次以上，这已经超过了车间通信的带宽(10Hz)，因此小地图可以最高的刷新率在机器人操作手的小地图画面上显示。

哨兵辅助自瞄性能分析及优化方案：在禁用哨兵自身的视觉自瞄条件下，完全使用雷达站的视觉识别方案可将哨兵的自瞄精度做到 1m 以内（哨兵红点距车辆的距离误差）。这个精度已经可以令哨兵根据雷达站提供的方位将云台快速瞄向目标的大致位置，再使用自身的自瞄系统对目标进行精确的打击。由于时间限制，并未测试辅助自瞄与自身自瞄共同工作时的工况，可以预计到会出现哨兵只使用辅助自瞄而弃用自身自瞄的误决策问题，这类问题有待进一步的测试与优化。

英雄测距性能分析及优化方案：英雄与雷达站距离测量误差在 1.5 米以内。对于英雄的吊射重力补偿程序来讲，这个误差还不够小。我们尝试在距离解算结果上加上一个固定补偿量来修正，结果是对于同一场比赛，修正后的测算距离基本能够保持在 1m 以内的误差，这个结果已被控制在重力补偿程序允许的输入误差范围内，因此我们的优化方案就是在每场比赛开始前，快速地测出固定补偿量的值并传给英雄控制程序，使其在本场比赛中可以使用精度较高的距离值来吊射。

#### 6.4.4.3 算法结果

如图所示，展示雷达站的 PNP 选点，场地敏感地区预警、显示血量、飞镖打击回放、比赛状态显示、战场信息显示的功能。



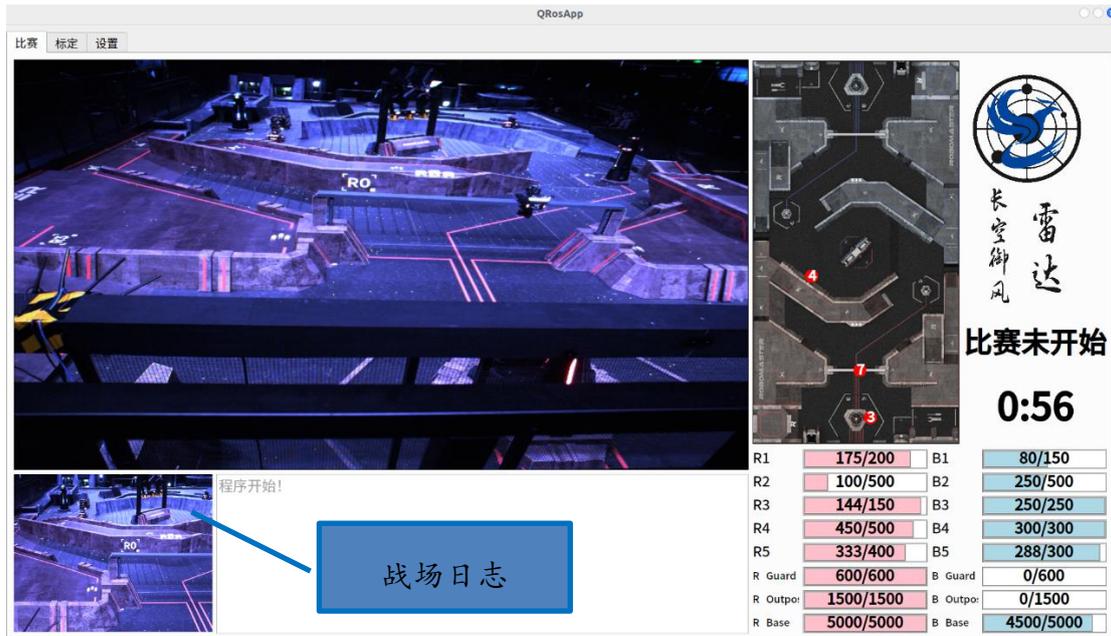


图2-17 自定义 UI 主页面

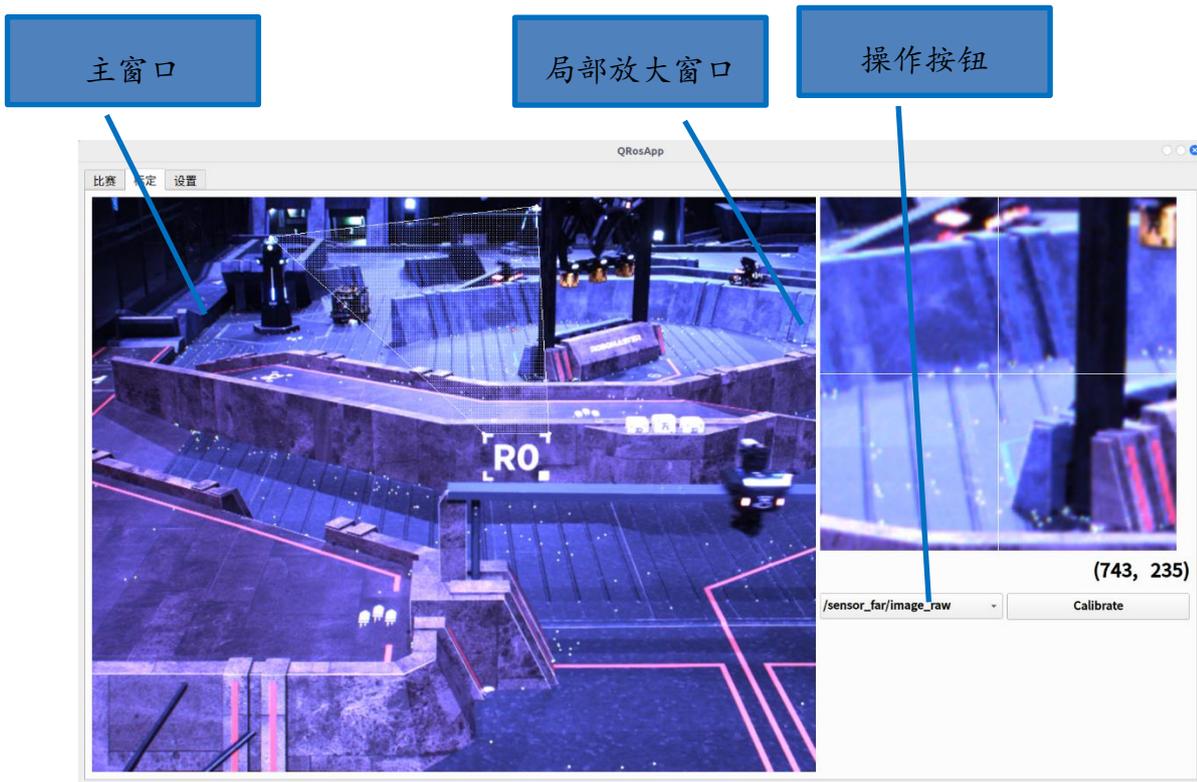


图2-18 自定义 UI 位姿估计页面

## 主视频源

主视频来自雷达站的上相机，为 UI 主页面的主要部分，方便云台手最直观的获取战场信息。基于 Qt5 的 QLabel 类实现，通过不断更换 QLabel 的贴图实现视频的实时显示效果。

## 战场信息显示

场地日志框是我们雷达站的一个特色功能。雷达站通过运算得到的信息（如敌方步兵飞坡、英雄击打前哨战等）和通过裁判系统直接获取到的信息（如判罚信息、能量机关激活情况）可以通过场地日志直接打印出来。

我们根据事件的紧急程度，将其分为五个等级，通过不同的字体、颜色显示。当提示框占满后，后续消息会自动覆盖最久的消息。

## 小地图

小地图是雷达站最基本的功能。雷达站自定义 UI 上的小地图主要承担着两个重要功能，一个是显示双方车辆位置，一个是区域预警消息的直观显示。

在比赛开始后，UI 会根据我方颜色加载正确的小地图背景，以使云台手获得最直观的观感。UI 收到来自小地图节点的解算消息后，根据兵种和阵营，将其绘制在小地图上。当收到来自小地图节点的预警信息后，UI 会在小地图对应区域以 3hz 的频率闪红，同时会在战场信息中进行打印。



图2-19 小地图预警示意

## 飞镖打击回放

飞镖打击回放是为云台手击发飞镖提供便利的一个辅助功能，也是我们探索雷达站赋能全兵种的一次尝试。

在四川大学纯机械飞镖出奇制胜后，无控飞镖成为了主流。在这种背景下，云台手手动的瞄准和微调无疑是至关重要的。出于此目的，我们设计了飞镖打击回放功能。

当从裁判系统读取到飞镖闸门开启消息后，系统会自动开始记录左相机中飞镖行进轨迹部分的画面，将其保存在硬盘中。在飞镖闸门关闭自动从硬盘中加载，并开始连续三次回放，使得云台手有足够的时间评估本轮飞镖的打击效果，并进行微调，使得下轮飞镖打击取得更好的效果。

但是，雷达站作为和飞镖有着千丝万缕联系（雷达站位于飞镖正上方，且在飞镖打靶单项赛可以上场）的兵种，可以为飞镖做的远远不应该是一个简单的回放功能。这个功能，归根结底，是雷达站识别与定位、飞镖控制都不成熟的背景下，一个妥协的产物。

## 裁判系统信息显示

在雷达站的自定义 UI 上，通过读取裁判系统的数据，我们展示了机器人、前哨战和基地的血量，以及比赛阶段和倒计时等信息，方便云台手快速反应。

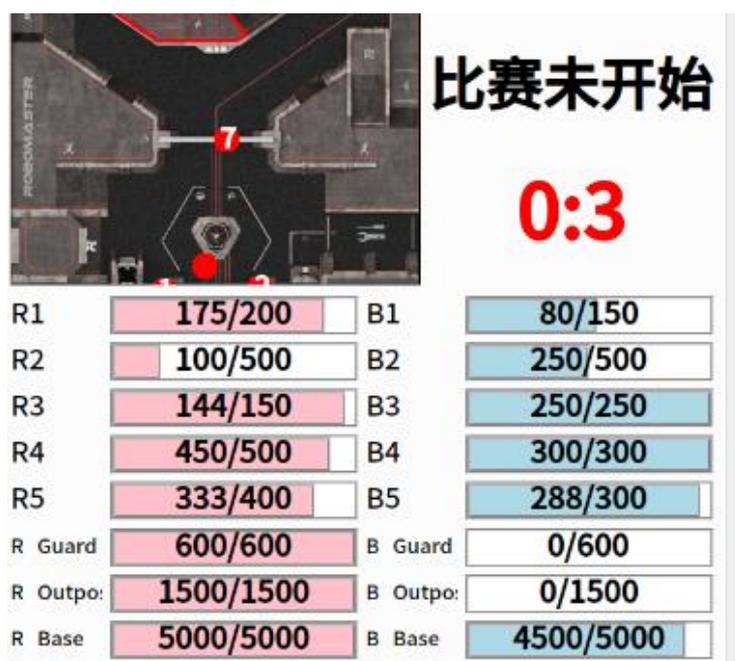


图2-20 裁判系统信息显示

机器人血量可以根据我方阵营，自动调整颜色，以始终保持我方机器人血量在前，敌方机器人血量在后。血条基于 Qt5 的 progressBar 实现。同时，通过获取到的血量信息，通过遍历该兵种所有可能的血量值，便可以粗略估计出该车辆的最大血量，实现血条的效果。当未能成功获取机器人血量信息时，机器人血条显示为繁忙状态，如图所示。

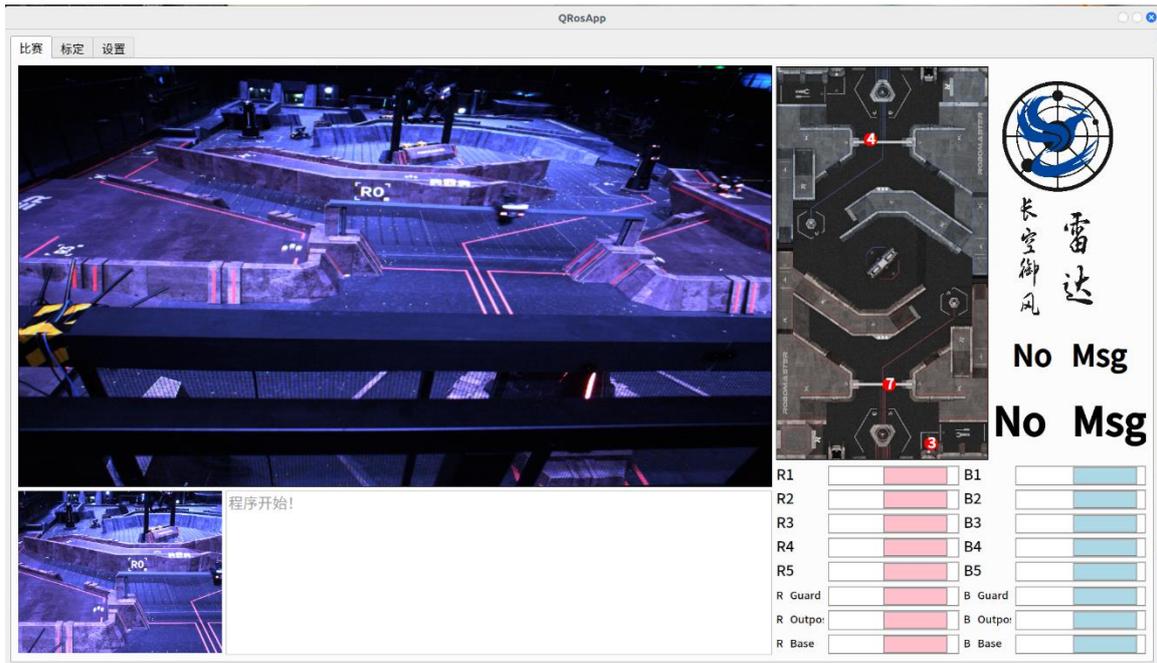


图2-21 血条的繁忙状态

## 相机位姿估计

相机位姿估计是雷达站三分钟准备阶段需要做的最为重要的工作，直接关乎整场雷达站的表现。所以方便雷达上场人员方便、快捷、准确的做好相机的位姿估计工作至关重要。

考虑到如果只对一个相机进行位姿估计工作，需要经过层层坐标变换才可以获得另一个相机的位姿信息，容易造成误差的放大。所以我们选择对两个相机分别进行位姿估计。赛前工作量增多，便对 UI 操作的流畅性提出了更高的要求。

我们选择了使用 R0 上方两个角点、前哨站最高点（左右相机分别为地方和我方前哨站）和敌方基地最高点进行 PNP 解算，使各个点坐标跨度尽可能大，以提高 PNP 解算的精度。

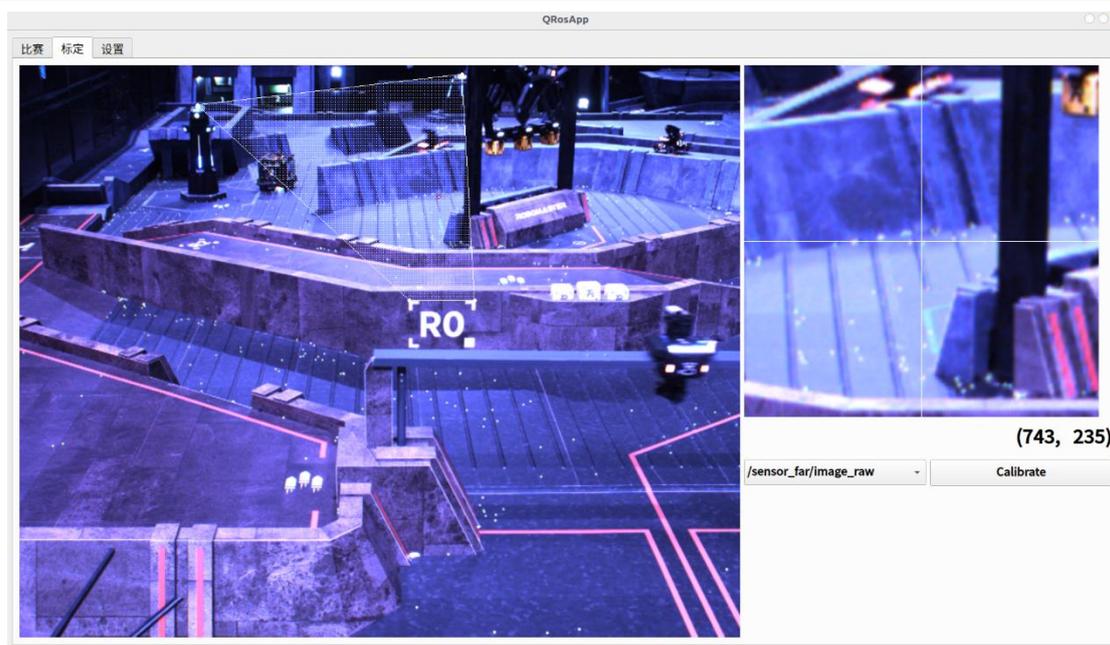


图2-22 左相机 PNP 选点

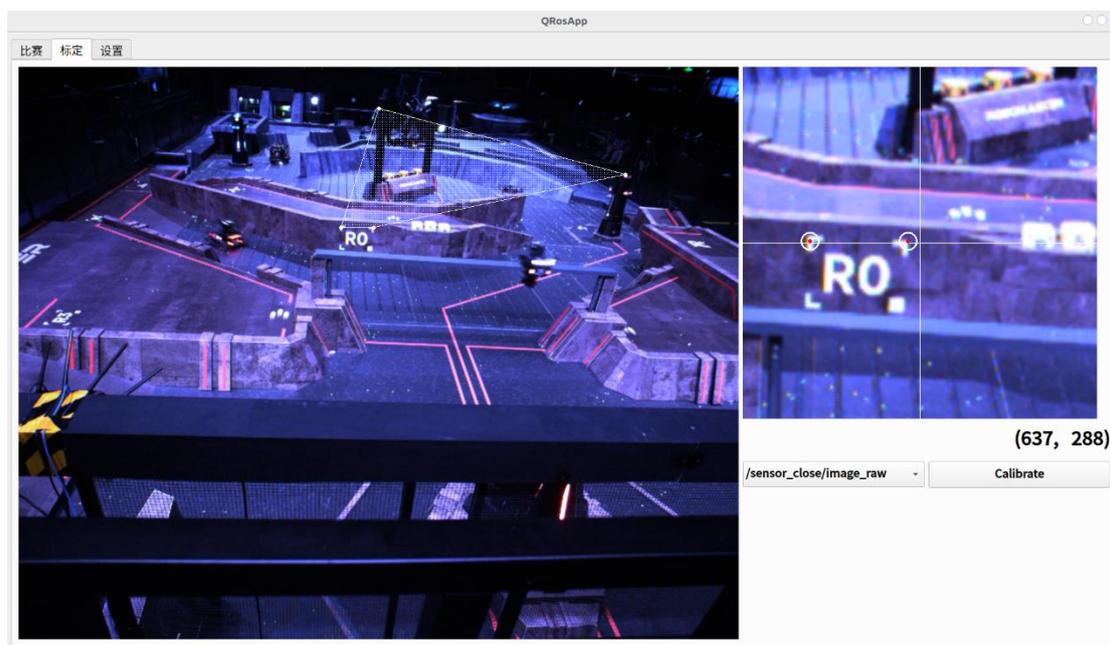


图2-23 右相机 PNP 选点

雷达站 UI 启动后，会首先加载默认的位姿估计结果，并使用该数据进行正常的解算。通过 Qt 的选项卡功能，可以从主页面切换到位姿估计页面。主窗口上绘制了默认的参数。通过鼠标拖动四个角点，即可调整该点的坐标。右侧小窗口会对鼠标的中心位置进行实时放大，并以十字叉丝标注鼠标中心所在位置，放大倍数可以在参数文件中轻松修改。同时，屏幕上实时显示鼠标所在坐标，当鼠标按下后，坐标变为红色，以提示雷达操作人员。

在所有点调整完成后，点击“calibrate”即可保存当前结果，或者通过下拉菜单切换相机自动

保存。在两个相机全部标定完成后，便可通过选项卡切换回主页面。

相机的位姿估计可以随时重新进行，只需重复上述操作，新的结果便可以覆盖旧的结果。同时，由于相机位姿估计是雷达站唯一需要手动进行的操作，我们基于 ROS 的参数服务系统，对所得结果采取了充分的防丢措施。当点的位置改变时，UI 会实时将点的位置保存到参数服务器中，通过这样的操作，当 UI 或者小地图节点意外重启时，原先的结果会自动加载，无需重新进行位姿估计，增强了系统的鲁棒性。

## 6.5 研发迭代过程（10）

### 6.5.1 版本迭代过程记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	第一版 UI 完成，基于 nodejs 等前端设计框架，在浏览器中运行。	2022.3.6
V1.1	激光雷达点云数据处理及深度获取 demo 开发。	2022.3.25
V1.2	用于识别车辆的神经网络训练、部署、可视化完成。	2022.5.24
V1.3	小地图功能 demo 完成。	2022.6.17
V1.4	第二版 UI 完成，完成三个小地图派生功能。目前的最终形态。	2022.8.31

## 6.5.2 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	机器人版本号或阶段	解决人员
1	UI 卡顿。	UI 部署在浏览器，限制了其性能。	使用 Qt5 重新编写 UI，完全解决了问题。	分区赛结束后	刘建航
2	联合标定误差过大。	图的数量太少，部分误差较大的图严重影响了整体的标定效果。	增加图片数量，同时进行二次筛选，删除选点不正的图。问题完全解决。	分区赛结束后	王书钰
3	UI 性能开销过大，导致整个系统卡死。	在 Ubuntu18.04 下，QLabel 贴图会导致 gnome-shell 的 CPU 占用急剧上升。	升级系统版本为 Ubuntu20.04。完全解决了问题	分区赛结束后	刘建航
4	神经网络第二层无输出。	忽略了 OpenCV 取 ROI 为浅拷贝。	使用深拷贝的方式取 ROI。问题完全解决。	分区赛结束后	刘建航

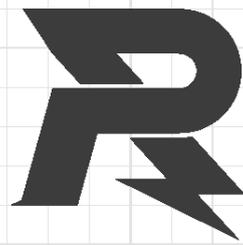
序号	问题描述	问题产生原因	问题解决方案 &实际解决效果	机器人版本号或阶段	解决人员
5	神经网络数据集不足。	之前雷达站的有效信息积累太少。	大量使用开源数据集，同时增大了自制数据集的比例，并且在分区赛中录制了更好的数据集。目前平衡步兵（3、4号大装甲）及5号装甲数据仍较少。	赛季初及分区赛结束后。	刘建航
6	Realsense 驱动引起系统周期性卡顿	原因不明	重写 Realsense 驱动后问题仍然存在。但是不在指定图片配置，全部使用默认参数后问题解决。	分区赛结束后	刘建航

## 6.6 团队成员贡献（2.5）

姓名	基本信息 (专业、年级、队内角色)	主要负责工作内容描述	贡献度 (所有成员贡献度合计为 100%)
王书钰	自动化、博一、视觉组成员	负责雷达站的功能与战术的顶层设计、小地图功能及其派生功能开发、雷达站与机器人间数据交互调试。	40%
刘建航	飞行器设计与工程、大三、视觉组成员	负责雷达站第二版 UI 的设计与开发、硬件模块设计与安装调试、串口通信、神经网络数据集制作与训练、部署	40%
左清宇	飞行器设计与工程、大四、视觉组成员	负责雷达站第一版 UI 的设计与开发，神经网络加速库 tensorrtx 的部署	20%

## 6.7 参考文献（2.5）

- 1.高翔、张涛：《视觉 SLAM 十四讲》
- 2.坐标系变换&坐标变换 - 赖东风的文章 - 知乎 <https://zhuanlan.zhihu.com/p/274976956>
- 3.OpenCV-4.5.4 文档
- 4.Robomaster 2022 超级对抗赛比赛规则手册
- 5.裁判系统学生串口手册 V1.3
- 6.tensorrtx-GitHub 仓库及其使用方法：<https://github.com/wang-xinyu/tensorrtx>



邮箱: [robomaster@dji.com](mailto: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