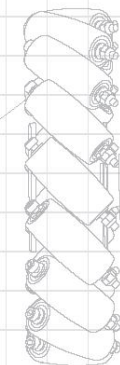


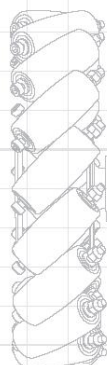


Using a BL-HS motor driver chip and Field-Oriented Control (FOC), the RoboMaster G300 Brushless DC Motor Speed Controller enables precise control over motor torque.



Exclusively designed for the RoboMaster G300, this 12V 30A Brushless DC Motor Speed Controller, the U3000 Assembly Kit includes several cables and a terminal board.

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



The M3000 Assembly Kit includes several cables and a terminal board, providing a complete assembly solution for your RoboMaster system.

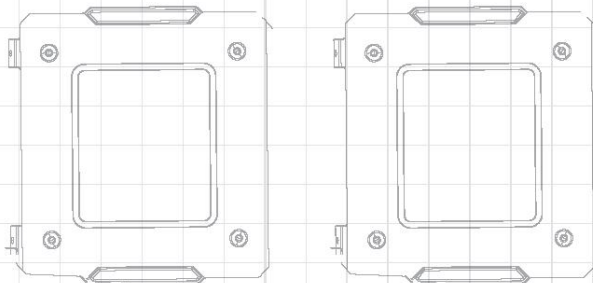
# ROBOMASTER

## 机甲大师超级对抗赛

# 技术方案

佛山大学 醒狮战队 编制

2024年6月 发布



# 前言

本成本报告由佛山科学技术学院醒狮战队编制，适用于 RoboMaster 2024 机甲大师超级对抗赛。主要撰写人员包括：

模块	撰写人员 1	撰写人员 2
机械	陈健安	黄皓杰
硬件	黄皓杰	凌浩天
软件	凌浩天	刘华轩
算法	郭光翀	彭绍辉
其他	黄皓杰	朱熙杨

# 目录

前言 .....	2
1. 概述 .....	4
1.1 背景与目标 .....	4
1.2 其他学校机器人分析综述 .....	4
1.2.1 RMUC23 赛季总结 .....	4
1.2.2 RMUC24 赛季 .....	5
1.3 机器人功能定义 .....	6
1.3.1 整体功能定义 .....	6
1.4 机器人核心参数 .....	7
1.5 设计方案 .....	7
1.5.1 机械结构设计 .....	7
1.5.2.1 整机机电连线设计 .....	17
1.5.2.2 自研编码器 .....	17
1.5.3.1 系统架构和运行流程 .....	20
1.5.3.2 自定义控制器 .....	22
1.5.4 算法设计 .....	23
1.6 研发迭代过程 .....	33
1.6.1 测试记录 .....	33
1.6.1.1 伸出测试 .....	33
1.6.1.2 抬升测试 .....	34
1.6.1.3 底盘避震测试 .....	36
1.6.2 版本迭代过程记录 .....	43
1.6.3 重点问题解决记录 .....	43
1.7 团队成员贡献 .....	45
1.8 参考文献 .....	46
1.9 技术方案复盘 .....	47
1.9.2 赛场性能表现与规划对比分析 .....	47
1.9.3 经验总结 .....	51

# 1. 概述

## 1.1 背景与目标

在 Robomaster 赛场中，工程机器人掌握着己方队伍的经济命脉，间接决定了己方步兵以及英雄的进攻与防御强度，获取经济的效率很大程度上影响了赛场的风向。在 2024 赛季中，工程机器人主要任务为：吸取并兑换矿石，次要任务为配合步兵英雄发起进攻及退回防守。若在比赛中工程机器人出现故障，无法获取紧急，己方队伍只能依靠自然经济增长进行作战，进攻能力将大幅下降，这几乎没有获胜可能。

相较于过往的赛季，大资源岛的金矿石从空中掉落的不稳定状态到放在凹槽的稳定状态，对于工程机器人取矿稳定性有了大量提升，同时因为金矿与银矿价格的差距，以及没有了掉落矿物的倒计时，开局哪一方能更快取到金矿更决定了比赛前期的风向，所以快速取到金矿石成为本赛季的一大目标。在取到矿石后稳定兑换成金币也是必不可少的一环，光有矿没有兑换到钱的矿石只是一个方块，在能兑换前三级难度前提下，兑换更高级的难度增加单个矿石的金币兑换率。

## 1.2 其他学校机器人分析综述

### 1.2.1 RMUC23 赛季总结

23 赛季的工程主要结构为一个六轴的机械臂，辅以龙门架的抬升以及伸出，原先设想是通过龙门架的辅助使用简单的平面二维运动以简化操作手的操作，在 23 赛季的区域赛中工程操作手在比赛过程中多使用了机械臂进行矿石姿态调整，再使用伸出抬升进行矿石位置的调整，达到了功能的实现。但是使用这种结构由此引出了一个大的问题，当机械臂的姿态为垂直于地平面时，占用了一部分的抬升空间，导致抬升机构上升空间被压缩，在夹取大资源岛金矿时机械臂末端过低，金矿下落速度快，无法做到空接的操作，导致金矿掉入异型矿槽，大多数情况下金矿石位姿难以做到比较高效地夹取金矿石以获取经济，总结 23 赛季的工程结构设计，高不成低不就，处于一个十分尴尬的位置。而此种设计反映在 24 赛季的问题则是当兑换框处于深处高处的位置时无法进行兑矿。

而在 23 赛季中不少的队伍使用了三轴或四轴的兑换机构，依旧达到了较好的夹取矿石的效果，参考华南理工大学华南虎战队，采用龙门架加小三轴兑换机械臂，也可以快速地完成五级矿石的兑换。

而采用高自由度纯机械臂的队伍，以上海交通大学交龙战队为例，六轴机械臂，在兑换上有很优秀的表现，但是在某些简单任务，如小资源岛矿石夹取则速度稍逊一筹，以及末端电机需要使用上大扭矩的关节电机，在价格上不是特别美丽，与本战队而言，性价比不高。

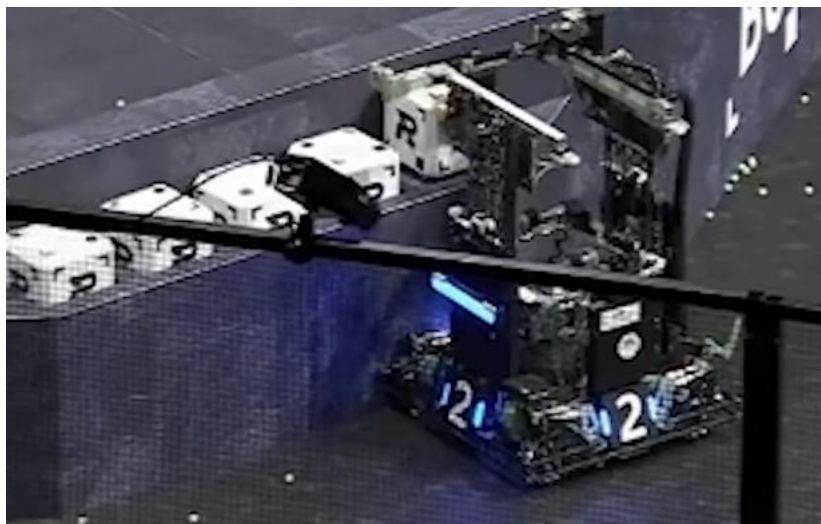


图 1 华南虎工程机器人夹取小资源岛

除此之外，23 赛季 TDT 战队一次夹取三矿石，引起我们对工程机器人执行效率，机械设计，空间利用率等问题的思考。

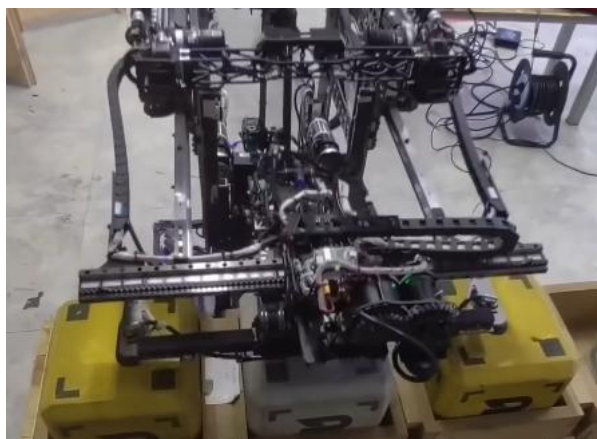


图 2 TDT 战队工程机器人一次三矿

## 1.2.2 RMUC24 赛季

相较于上个赛季，2024 赛季中金矿的重要性更加凸显，因此我们将工程机器人的设计重心放在如何快速夺取中央金矿上，针对这个赛季大资源岛的直型与倾斜矿道，我们设计了一个具备 yaw、pitch 自由度的夹矿专用机械臂，采用公式化动作快速夺取金矿。同时为应对高难度的兑换，采用丝杆配合另外的兑换用机械臂进行兑矿，该机械臂具有 x、y、z、roll、pitch、yaw 共 6 自由度。

华南虎工程机器人让我们看到了较少自由度机械臂与伸出抬升结合的设计。同时结合 24 赛季规则与场地的变化，给出了一个较好的环境以进行机械臂的工作。而这个赛季我们也选择了将吸取矿石与兑换矿石两个任务分给两个独立的吸取结构进行工作，让每个吸取机构拥有更高的一个效率进行工作。

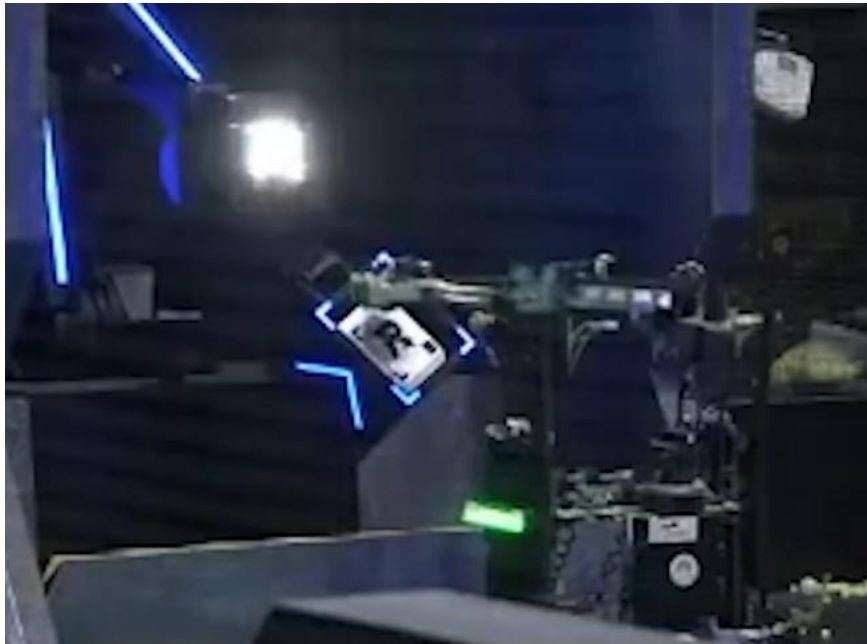


图 3 华南虎工程小三轴完成五级兑换

## 1.3 机器人功能定义

### 1.3.1 整体功能定义

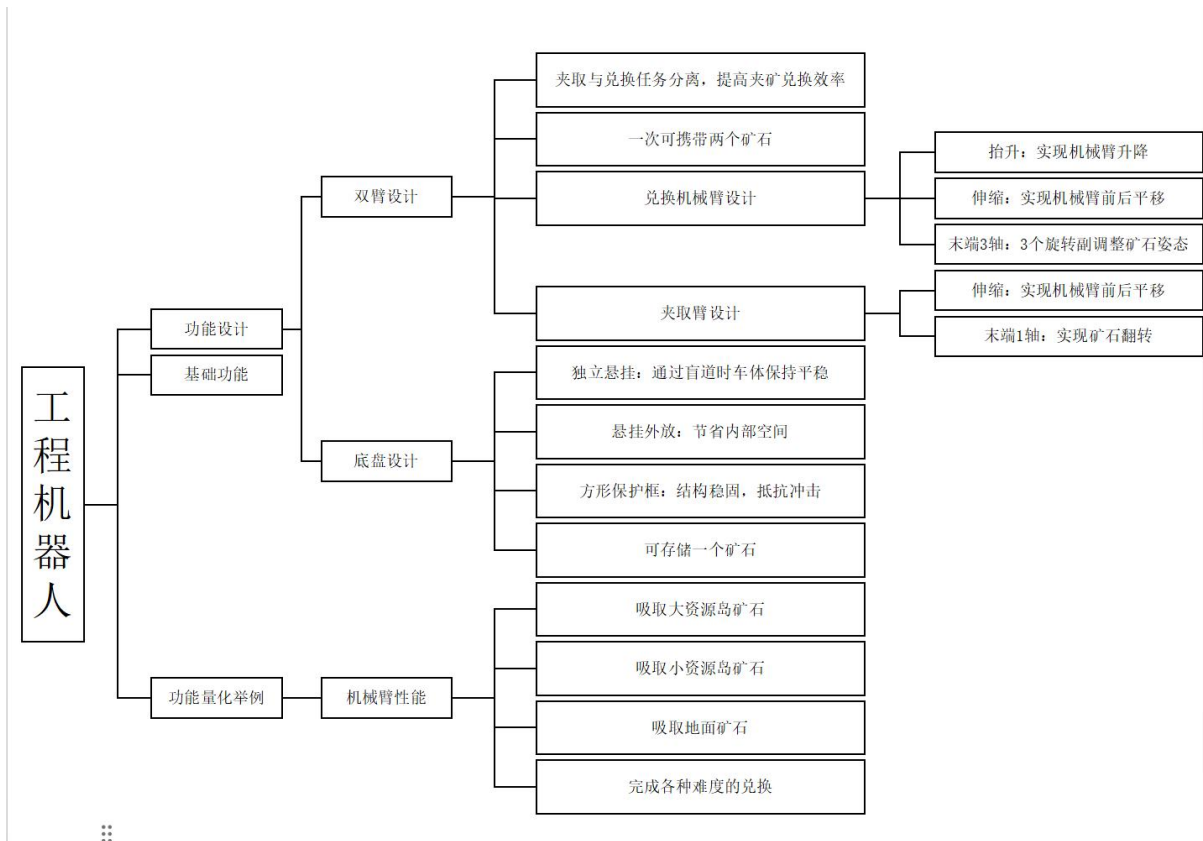


图 4 机器人核心参数

## 1.4 机器人核心参数

表 1 机器人核心参数

名称	参数
重量、重心	32.15kg
尺寸（长宽高）	589*592*585（原始尺寸） 1190*592*1070（变形尺寸）
主要传感器型号、参数、数量	轻触开关 PBS-33B 2个
电路功耗、所有电容总容量、工作时电压范围等	
执行器件（电机、气缸等）用途与数量说明等	3508 电机 8个 2006 电机 3个 DS3120_20KG 舵机 3个
机器人其它核心性能参数 - 如：车体最大移动速度、爬坡角度，云台自由度等	兑换机械臂自由度 5 夹矿机械臂自由度 4

## 1.5 设计方案

### 1.5.1 机械结构设计

#### 1.5.1.1 设计总思路

##### 1.5.1.1.1 机器人战术定位

在 Robomaster2024 赛季的赛场中，工程机器人是将矿石资源转化成金币资源的唯一途径，是队伍主要的经济来源，掌握着整个队伍的经济命脉，决定队伍每场比赛的大风向，是猛烈进攻还是保守防御。因此，吸取和兑换矿石的任务将作为工程机器人的首要任务。在每场比赛中后期，场上可供高效兑换的矿石数量稀少，任务重心也从夹矿兑矿逐渐转移至辅助其他地面机器人发起进攻或是退回防御。

### 1.5.1.1.2 机器人设计需求分析及设计思路

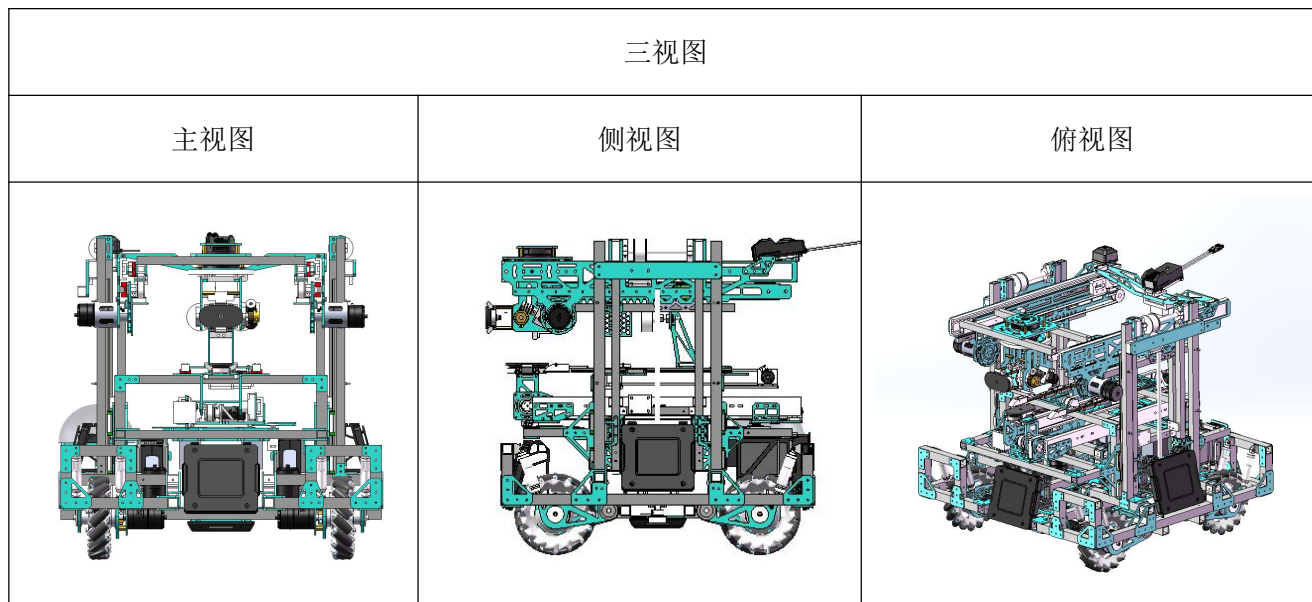
表 2 设计需求分析

需求分析	设计思路
需要一个稳定的底盘能供其他结构持续工作	采用四个独立悬挂的轮组；使用 20*20*2 的铝方管作为框架保证强度；
大资源岛的矿石放置在隧道的凹槽中，夹取空间受限，部分矿石有一定角度的倾斜	采用简单的伸出机构将一个吸盘伸入矿道夹取矿石；整个吸取矿石的机构能旋转 10°
五个金矿石皆放置在凹槽中，没有过往赛季的分批掉落，更容易被对方队伍把矿石夹完	设计存矿机构，尽可能多地一次出门，多矿回家，提高获取矿石效率
兑换站高等级兑换难度进一步上升	采用小型机械臂保证能使矿石调整到任何需求姿态，使用抬升和伸出机构使工作范围足够大

### 1.5.1.2 整车总体结构

整车结构由五大构件组成：底盘、抬升机构、伸出机构、兑换机械臂、取矿机械臂、存矿机构。

表 3 机器人三视图



#### 1.5.1.2.1 底盘

为了保证底盘的强度，整个底盘框架皆使用了 20\*20\*2 的铝方管，使用了两根横向的铝方管作为支撑整个机器人的横梁。



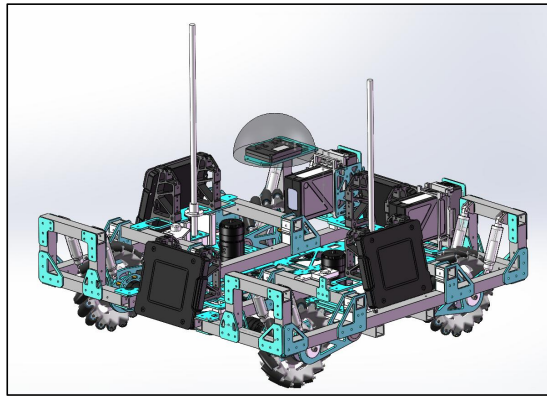


图 5 底盘结构

轮组的四个避震在车体端的安装位置选择在了底盘的四角，过往一般安装在车体中央，占据了车体中间一部分的空间，但车体四角除了轮组本身占据的安装地方几乎没有其他机构会使用到，将避震的车体安装端换到了车体四个角，减少对中间其他机构安装位置的占据或避免为了安装避震座而局限其他机构的安装布局。

对于避震弹簧弹性系数的确定，我们认为的原则是机器人在正常情况下的任何状况车底的零件都不应该接触到地面。同时整体的底盘又应该尽可能地接近地面，在降低整车的重心的同时，给予尽可能多的空间给予上层的其他结构。而工程机器人对于避震的需求相对步兵与英雄这种既需要高机动性又需要高稳定性的机器人不同，工程机器人对于避震器的基本需求只是开车不翻车，吸取兑换矿石底盘稳定即可，所以对于避震弹簧的要求并不高，只需要较为粗略的计算。根据本赛季的比赛规则荒地区突起高度为 10.3mm，设计麦克纳姆轮竖行程为 12mm，即避震器压缩了 10.56mm，整车质量约为 30kg，理想状态下全车 8 根避震平均分配其重量，即每根避震分配到 2.963kg 的质量，即需要的弹簧弹性系数为 2.84kgf/mm，再通过弹性系数和线径的计算公式的逆运算进行对弹簧的选型，最后选择了 2\*19\*60（线径\*外径\*长度）的弹簧进行购买，而在后续根据操作手的反馈，整车晃动比较大，所以换成了 2\*18\*60 的弹簧。

将底盘中部前后装甲板之间的全部部分空余出来给电控的电线、电子元器件等提供较大的布线位置，避免出现以往电子元器件堆积，线路元件相互交错的情况，方便后续检修。

对于电池的安装摆放，因为电池本身重量较重，可以起到一定的配重的作用，所以放置于工程机器人的后端。同时考虑到减少抬升电机的负担，减轻上层机构的重量，于是选择放置于底盘。

考虑到工程机器人的上层机构比较占据地方比较大，需要较多的安装空间，若电池朝向上放置，在电池进行更换时有可能与上层机构干涉，于是将电池朝向车体外。

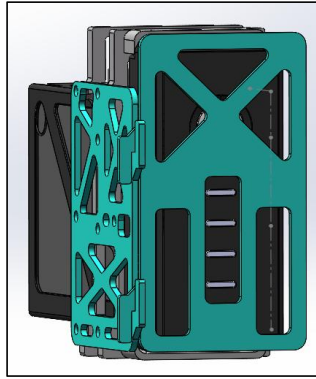


图 6 电池挡板的榫卯结构及其镂空

因为有些电池使用时间较长，部分电池两端卡扣已经损坏，在插入电池座后几乎是靠自身重量与安装架摩擦和电池本身端口与安装座的端口摩擦紧固，在机器人底盘向前加速或者转弯运动时，电池有可能松动甚至被甩出，所以设计了一个榫卯结构的挡板进行限位，在比赛中场时直接将挡板拔出即可进行更换电池。

根据 23 赛季的经验，在比赛中场时因为挡板镂空设计缺陷，无法看到电池剩余电量格数，每次都需要拔出挡板进行查看检查，浪费了赛场中宝贵的检修时间，所以对挡板的镂空进行设计，使其不会对电池格数的显示造成遮挡。

### 1.5.1.2.2 抬升机构

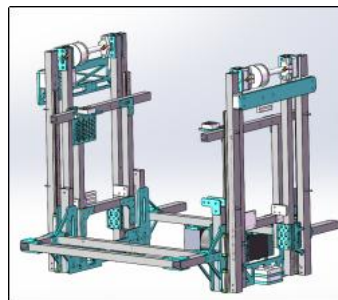


图 7 抬升机构

整体设计为丝杆加上同步轮同步带的二级抬升。

二级抬升：以同步轮（惰轮）作为动滑轮，通过丝杆抬升倒置的动滑轮，而动滑轮两端一端固定，另外一端连接需要抬升的机构从而达到丝杆抬升 1mm，上层的机构上升 2mm 的功能。

因为用于兑换矿石的机械臂选择了小型的三轴机械臂，本身工作范围比较小，所以要求抬升的行程就需要比较大。同时存矿的空间并不是以底盘为底面，而是在整车的正中央，当进行存矿时矿石的上部会与兑换机械臂产生干涉影响存矿，而将存矿安装在第一级抬升，兑换机械臂安装在第二级抬升，这样当随着抬升高度的上升，兑换机械臂与存矿机构在竖直方向上的距离会被拉大直至可以使矿石通过，达到存矿的目的。

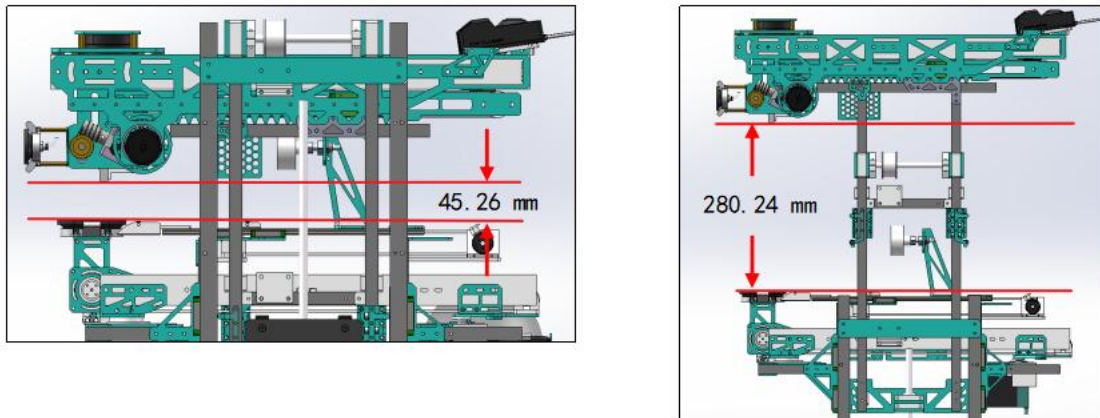


图 8 抬升前后距离变化示意图

丝杆：抬升机构需要带动两个机械臂、存矿、伸出和抬升机构，而且板材类选用的为玻纤板，上层的机构整体重量较重，丝杆的传动比大，从而减少电机的负载；采用梯形丝杆具有自锁的能力，能在己方工程机器人被击杀时防止因为断电造成上层机构竖直下落造成的冲击导致结构的损坏。同时，自锁的特点还能使电机的负载仅为带动上层机构抬升下降，而不需要保持上层机构位置。与此同时抬升电机与丝杆之间也是通过同步带进行传动，通过同步带本身带有的挠性减少在抬升启动瞬间对电机的冲击损坏。

直线导轨、滑块：对于标准件的直线导轨和滑块，整个的游隙小，精度高；直线滑轨安装在铝方管上，有助于防止在抬升机构上升到最高，伸出机构伸长最长时兑换机械臂重力臂增长造成支撑抬升的铝方管的弯曲造成兑换机械臂下沉甚至导致机器人本身结构损坏。

滑轮：制作成本低，方便安装；但是其缺点为间隙大，造成连接的两根铝方管偏移导致整个上层的机构向前倾斜，后面则通过两块小玻纤板对偏移铝方管的夹紧从而减少偏移量。

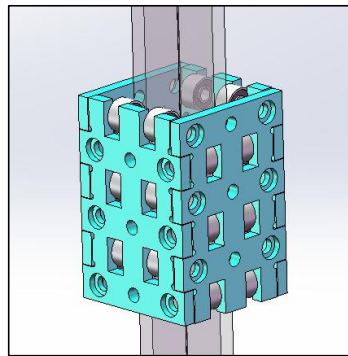


图 9 滑轮及其与铝方管配合示意图

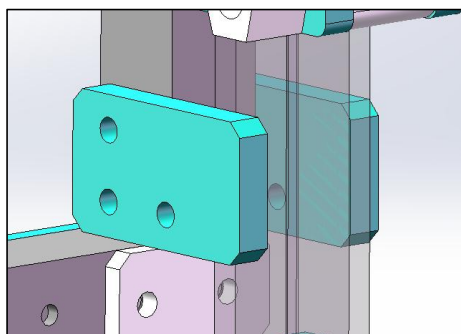


图 10 玻纤板夹紧铝方管示意图

光轴：在装配时因为人为装配误差造成前后两端的铝方管间距与图纸有一定的差异，通过调节一条光轴插入两端光轴安装座深度，可以适应两端铝方管的间距，做到适应两端铝方管不同间距和支撑两端铝方管的功能，如右图所示。

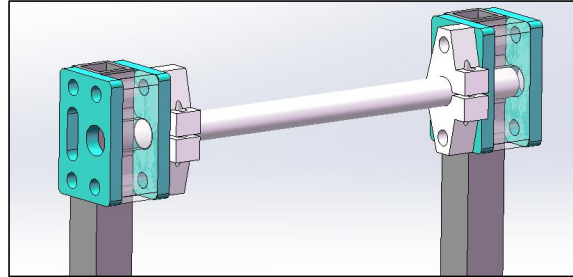


图 11 光轴安装示意图

### 1.5.1.2.3 伸出机构

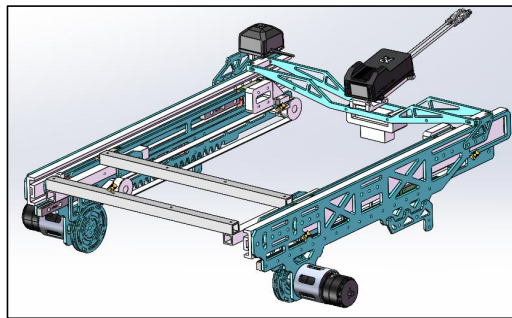


图 12 伸出机构

整体设计为齿轮齿条加上同步轮同步带的二级伸出。

二级伸出：以五级兑换难度的最复杂的情况作为伸出长度目标，以 RoboMaster 2024 机甲大师超级对抗赛比赛规则手册中 5.6.1 对兑换框坐标系的定义，当兑换框  $x=-270$ ， $\theta=\pm 135^\circ$  时，为机械臂的最长伸出条件（如下图所示）。而矿石长 200mm，当吸盘末端平面垂直地平面时，吸盘末端距离兑换机械臂转轴中心 95.45mm，伸出机构完全收起时，兑换机械臂转轴中心位于车体内，距离机器人最前端约 70mm，则伸出机构需要伸出 548.91mm。伸出距离要求较长，所以使用二级伸出。

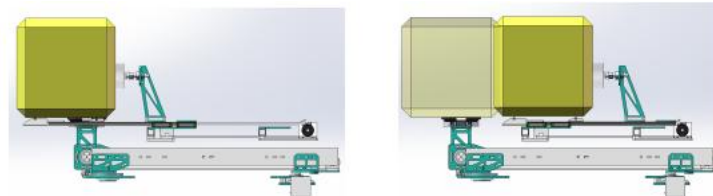
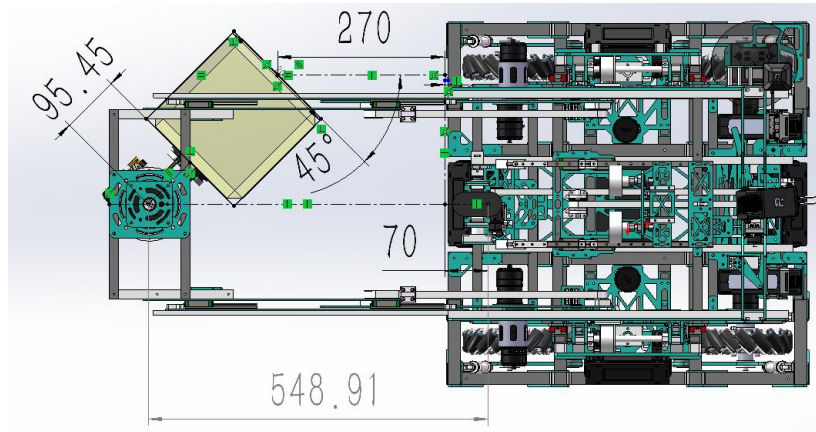


图 13 存矿机构的两个极限位置



抽屉滑轨：成本低，伸出距离长，收缩时长度为 550mm，可伸出 500mm。

直线导轨、滑块：因为抽屉滑轨伸出的距离不足以达到目标，所以通过小段的直线导轨滑块延长整体伸出的距离；因为安装板背面是抽屉滑轨的内轨，无法正常安装螺母，对于安装板背面进行了沉孔的设计（如下图），同时使用薄螺母，将螺母嵌入玻纤板以固定导轨；在实车运行中发现当运行伸出时，齿条容易脱离齿轮，在往后设计了两个挡板阻止齿条的偏移。

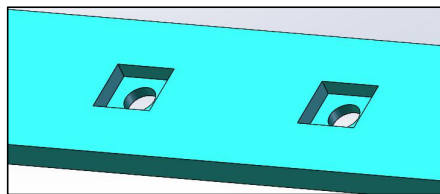


图 14 安装背板的沉孔

#### 1.5.1.2.4 吸取机械臂

整体使用了一个一轴吸盘+伸出的结构。

因为本赛季大资源岛中的金矿石是固定位置姿态，为稳定的状态，只使用一个尽可能简单的机构完成吸取矿石的任务有助于提高吸取矿石的效率。

对于夹取机构伸出的长度，从资源岛边缘到中间矿石表面的距离为 300mm，到两端前端矿石前表面的距离约为 373.38mm，到两端后端矿石的前表面的距离约为 436.34mm，即伸出车体外行程至少需要 440mm，当夹取机构吸盘吸取端面垂直地平面时，端面位于车体外，距离车体前端 28.9mm，即夹取机构需要伸出约 411mm，使用行程为 450mm 的抽屉滑轨，使用 2006 电机+同步带的驱动方式进行伸出。

### 1.5.1.2.5 兑换机械臂

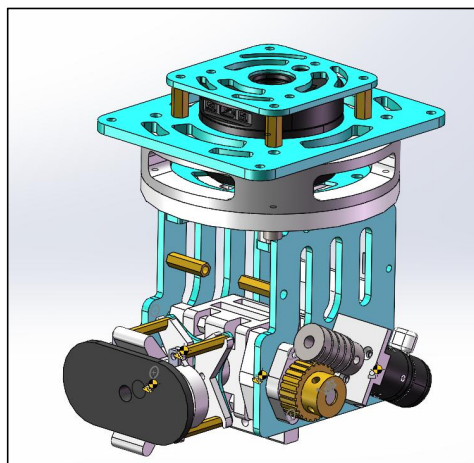


图 15 兑换机械臂

整体为一个小型的三轴机械臂。

对于兑换机械臂的设计，进行了如下讨论，主要为机械臂设计的选型。

#### 小型三轴机械臂：

##### 优点：

机械臂本体相对六轴小巧，结构简单，对于各个电机的负载都比较小；控制简单；

##### 缺点：

工作范围局限，需要配合外部构件运动工作才能做到更广的作业；

#### 大型六轴机械臂：

##### 缺点：

对于一般的六轴机械臂来说，起始的首个 pitch 轴电机负载较大，需要进行更换电机，如更换减速箱，换为关节电机等；或者进行降速增加扭矩的设计，如小同步轮配合大同步轮，使用蜗轮蜗杆等，使用的零件数目多，设计难度相对较大；

整个机械臂重量大，整体移动困难，完全依赖于本体自身杆件等零件的运动；

经过讨论考虑，参考了本队伍机械设计的能力与经验和经济因素的原因，选择了小型的三轴机械臂作为 24 赛季工程机器人的主要功能构件。为了解决三轴机械臂工作范围较小的问题，加以伸出和抬升机构进行辅助机械臂进行工作。同时，因为伸出和抬升的机构控制难度较小，利于电控调试；运动方向独立，操作手的操控难度小，在赛场上需要操作时相对于旋转运动需要思考决策时间更少，所以加装抬升和伸出结构也是较为适合的方案；

结合 23 赛季工程的设计经验，pitch 轴在吸取矿石后会产生一定量的下坠，这部分下坠一部分源于吸盘的变形，另一方面源于舵机在吸取吸盘前后力矩的改变。对于吸盘的变形，选取了更换吸盘种类的方案，

将硅胶吸盘换成了丁腈橡胶吸盘，同时辅以打印件作为支撑。而对于电机下坠方面，将直驱方式改为使用蜗轮蜗杆，使用机械结构本身带有的自锁性减少电机转动造成下坠。

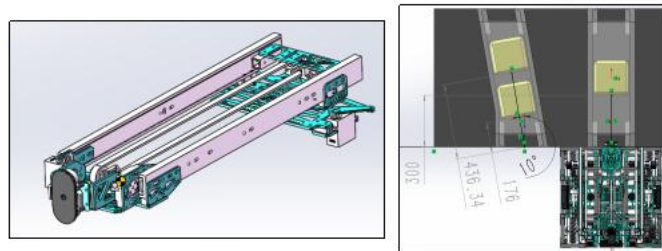


图 16 夹取机械臂以及大资源岛俯视图

因为大资源岛的高度相对较矮，夹取机构吸盘端位于车体中间的前部，其下方即为装甲板，使用 6020 电机安装不方便，同时使用 6020 电机只负责转动  $10^\circ$  的角度，有点浪费电机，于是采用了舵机进行驱动。同时因为抽屉滑轨的重量较重，长度较长，若舵机直接安装在转轴直驱转动整个夹取装置容易损坏舵机，所以将舵机后置，整体为一个曲柄摇杆机构，增长驱动力臂，减少舵机的功耗。同时为了防止抽屉滑轨后端下沉造成结构损坏，在机构后端使用了万向球轮进行支撑。

而前端的吸取的机构就简单使用一个舵机进行直驱使矿石可以进行翻转以进行后续的存矿。

### 1.5.1.2.6 自定义控制器机械设计

#### 需求分析

当今赛季兑换机制迫使工程需要极高自由度的兑换机构来提高兑矿金额，而高自由度的机构意味着操作繁琐与复杂。传统的键盘与摇杆控制对于这种高自由度的机构存在着极大的缺陷，它们往往只能同时操作一到两个自由度且操作方法不够直观。采取为机器人量身打造的控制器可以解决上述难题并极大提升兑换效率。我们选择制作一个与工程机器人兑换机构相似的小型模型，在关节处用编码器作为角度传感器测量各自由度的变化，以此作为数据发送给机器人使得机器人动作最终与模型动作同步。操作时只需摆弄模型便可直观地远程控制机器人。

#### 迭代记录

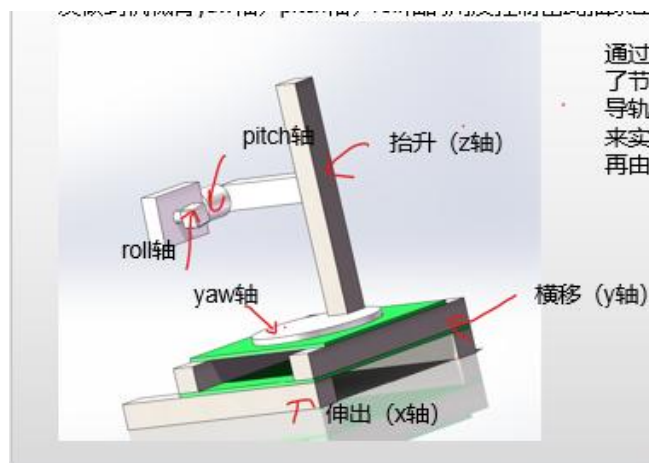


图 17 自定义控制器草图

通过安装 5 条光轴 9 个滑块（x 轴两条导轨 4 个滑块，y 轴两条导轨 4 个滑块，z 轴一条导轨 1 个滑块）来实现控制器在空间中的移动，再由三个打印件构成控制器 yaw 轴，pitch 轴，roll 轴的控制机构。

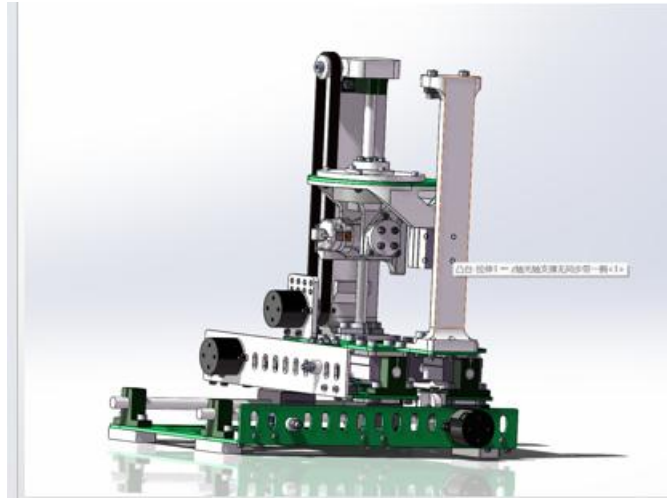


图 18 一代自定义控制器

根据电控的要求和反馈以及有限元分析对物理模型 1.0 做出一定调整得到第一版。

在 z 轴增加了一条光轴以保持机械臂部分平衡不发生倾斜，将控制 roll 轴，yaw 轴 pitch 轴，的结构改成与工程机器人上机械臂类似的小机械臂模型，控制器的状态以机械臂姿态一致，对操作手更友好。

## 部件设计与选择

### 角度传感器结构设计

通过将控制器的运动通过转轴转化为旋转信号传输给编码器，再由编码器发送给对应的电机一控制电机旋转。

转轴旋转带动下方磁铁旋转产生磁场变化，转轴采用 D 型设计防止转轴与连接机构相对滑动。3D 打印的轴承架具有一定弹性便于拆卸检修。



图 19 角度传感器设计图纸

### 机械臂模型结构设计

仿制实际机器人机械臂制作的小型化模型，机器人上电机位置在模型中一一对应用角度传感器替换

借鉴福建师范大学自定义控制器关节处增加阻尼方式，使得模型在脱手后也能保持维持姿态硬件设



计。

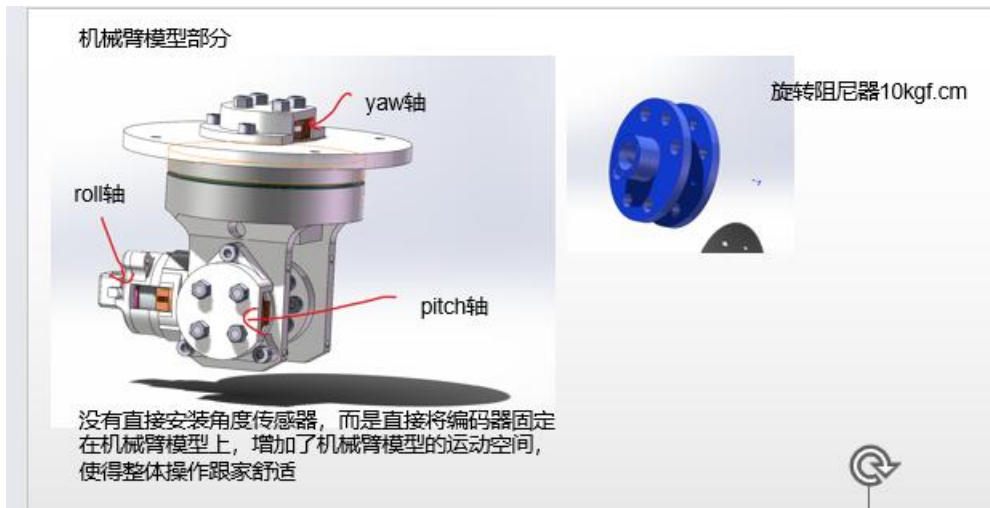


图 20 模型设计

## 1.5.2 硬件设计

### 1.5.2.1 整机机电连线设计

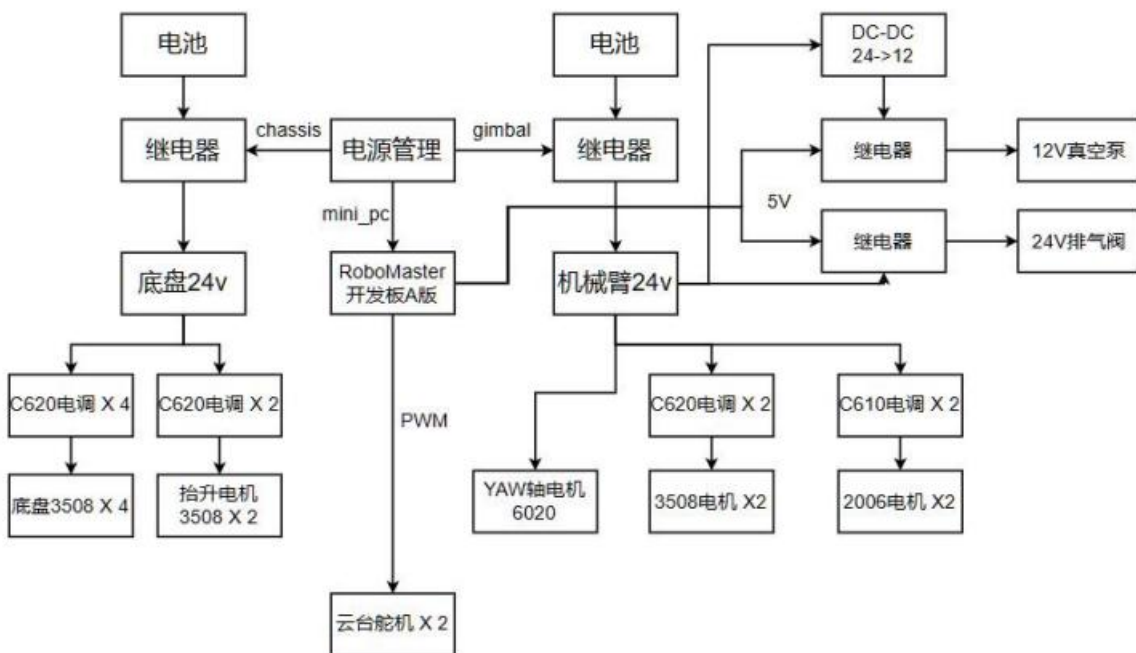


图 21 整机机电连线设计图

### 1.5.2.2 自研编码器

#### 1.5.2.2.1 需求分析

作为一个测量机械臂关节角度的角度传感器，其精度要求误差在大约一度以内；为控制最终控制器的

尺寸并降低机械设计难度，编码器要尽可能做得小；战队经济水平有限成本不能太高。在比较过的光电增量式编码器与磁编码器后，选择了没那么精密但制作成本低的磁编码器，以 as5600 为核心芯片。

### 1.5.2.2.2 电路设计

将 as5600 的 SCL 与 SDA 引脚经过 4.7kΩ电阻后与 VCC 连接达到弱上拉的目的，从而满足 IIC 的硬件通信需求。将 VCC、GND、SCL、SDA 接到 GH1.25-4pin 卧贴接口上，方便配合专用分电板将 SCL、VCC、GND 接到一起。

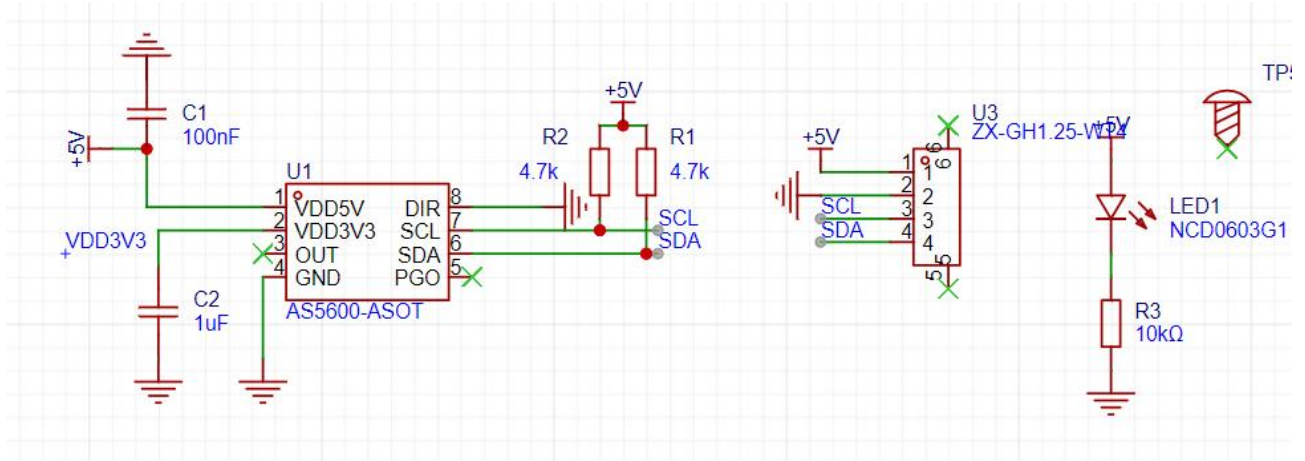


图 22 AS5600 原理图

绘制出编码器的 PCB 制作为直径 28mm 的圆形电路板，AS5600 芯片置于中心便于读取磁力方向。

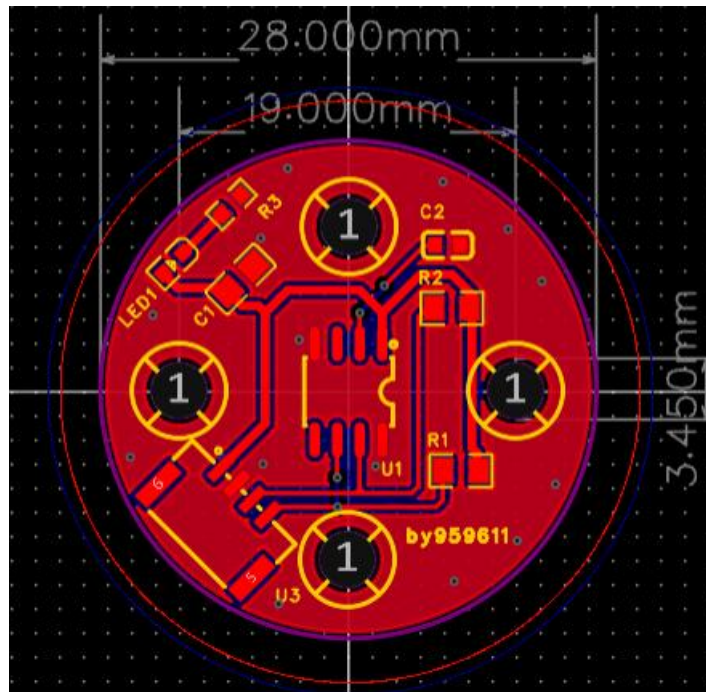


图 23 AS5600 编码器 PCB

由于 IIC 单一总线上可承载的通信设备有限，为了提高通信效率选择采用多条数据线共用一条时钟线的方式，即所有编码器的 VCC、GND、SCL 并联，SDA 分别与单片机连接。

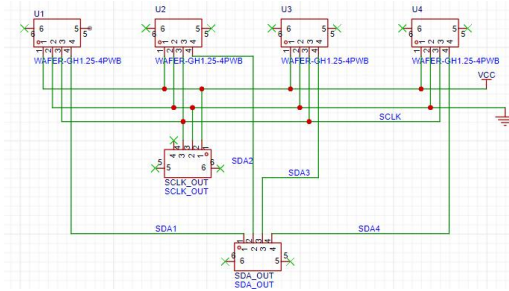


图 24 分电板原理图

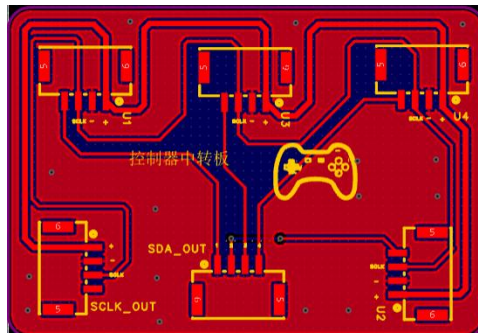


图 25 PCB 打样

SCLK\_OUT 接口连接 VCC、GND、SCLK，SDA\_OUT 分别与 u1、u2、u3、u4 的 SDA 连接。

## 1.5.3 软件设计

### 1.5.3.1 系统架构和运行流程

#### 1.5.3.1.1 系统架构

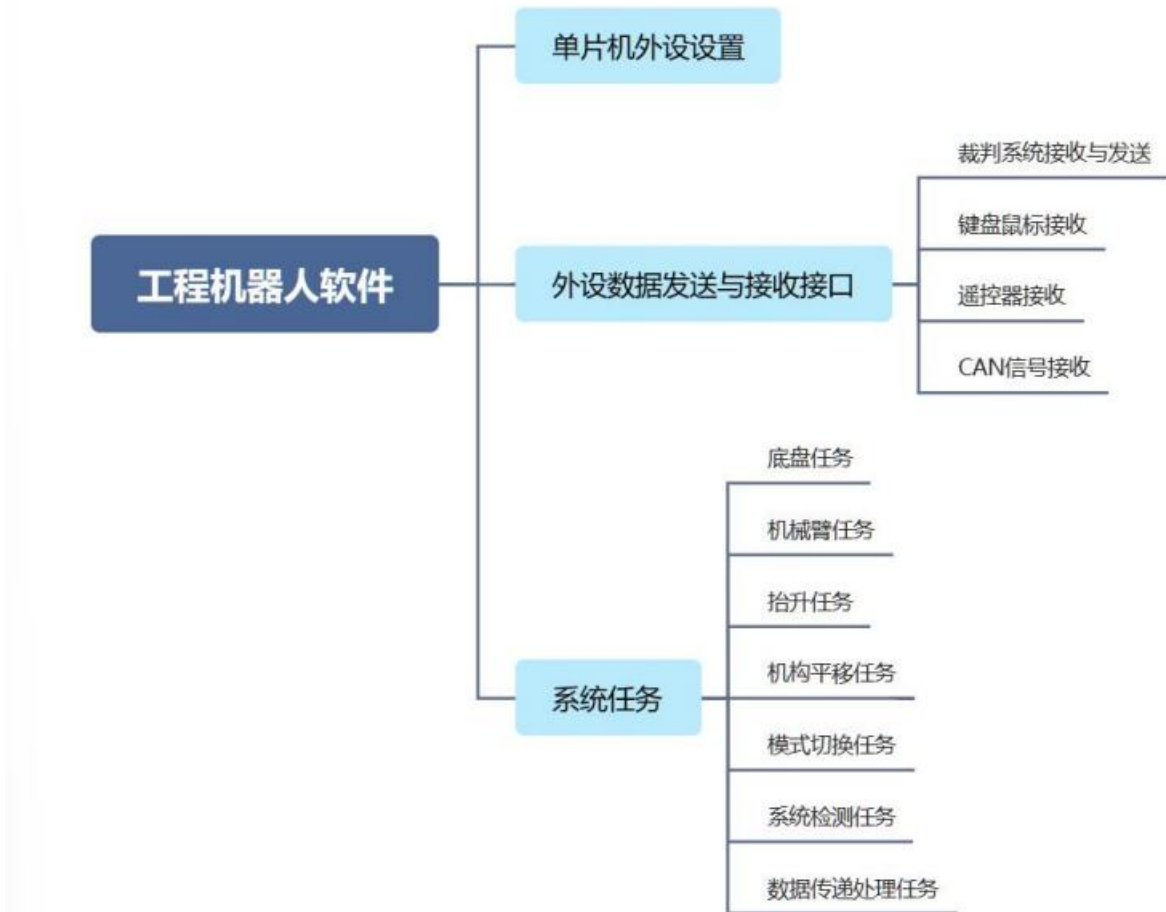


图 26 工程机器人软件框图

工程程序基于 FreeRTOS 系统, 通过任务调度控制多个运动模块, 主要分为底盘任务、机械臂任务、抬升任务和机构平移任务等。各任务会通过各自模式执行对应项目, 保证项目实行的实时性。

#### 1.5.3.1.2 运行流程

软件启动后将初始化硬件与软件参数, 并启动任务调度, 随后检测任务将扫描硬件反馈信号以检查硬件连接状态是否正常, 随后将进入各执行模块的电机角度校准, 完成后通过模式的切换执行不同的工程任务。

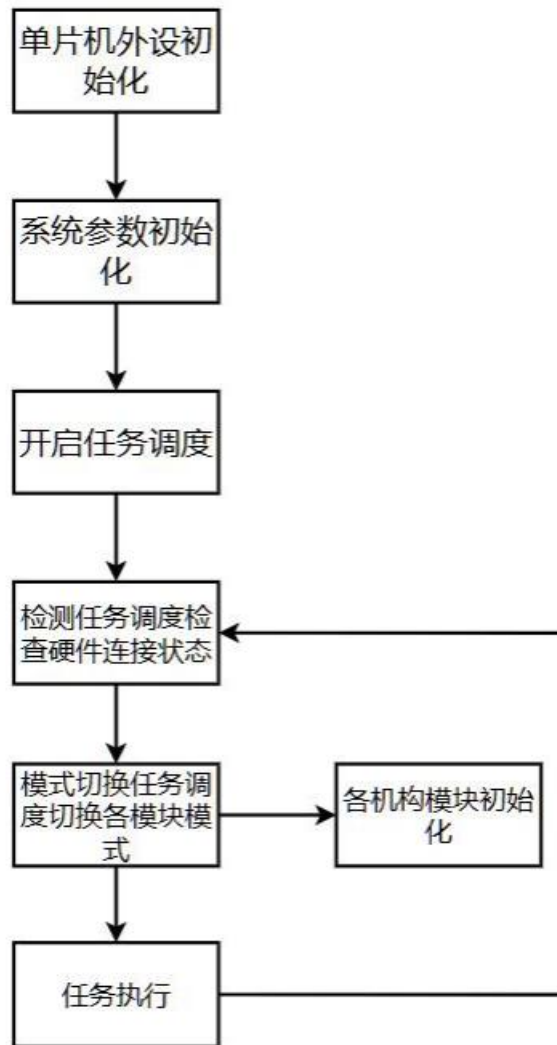


图 27 工程机器人任务流程图



图 28 工程机器人模式选择

工程总模式有普通模式、工程模式、掉电模式。普通模式用于普通移动。工程模式用于 执行工程的取矿、兑换、搬运障碍块等任务。掉电模式则是机器人停机或机器人出现异常时 紧急断电。电机位置初始化分为两种情况，一种是对减速比 1; 1 的 6020，只需直接读取其编码值 11 即可。另一种则是对有减速比的电机进行位置校准，在不外接编码器的情况下，不能直接使 用编码值进行角度处理，需要先以恒定转速转动，直至到达机械限位发生堵转时，将电机转 速设置为 0，避免堵转发热，随即记录当前编

码值的角度转换值作为初始值，之后进行角度控制时只需加上初始值即可。通过先运行速度环完成机构移动，后通过角度环完成定位，避免了由角度控制的过快响应造成的结构损失。



图 29 电机位置校准流程

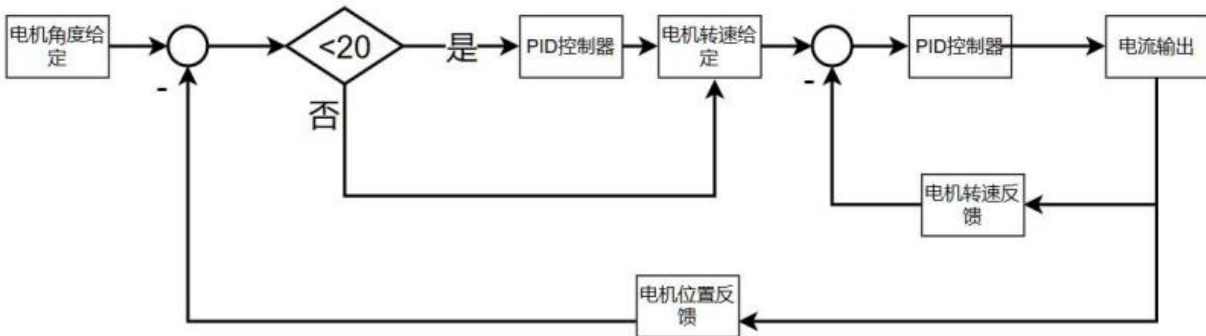


图 30 电机 PID 控制框图

### 1.5.3.2 自定义控制器

#### 1.5.3.2.1 编码器版

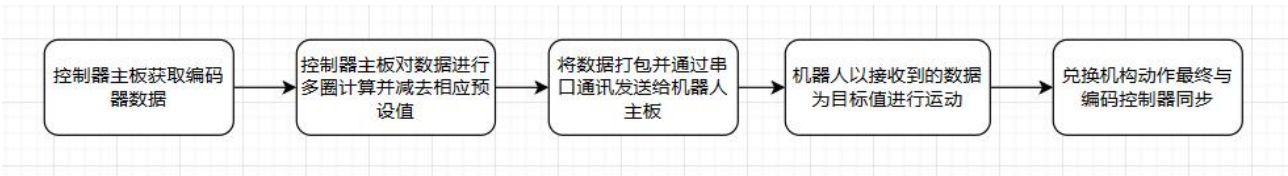


图 31 编码控制器流程图

AS5600 编码器采用 IIC 的通讯方式，为了能同时读取 pitch、yaw、roll、x、y、z 自由度上的六个编码器。

码器的数据，我们选择自己设计软件模拟的 IIC 通讯，定义六个引脚为 SDA 分别接收对应编码器的数据，另取一个引脚为 SCLK 作为所有编码器的时钟线。

主机发送数据时，在 SCLK 为低电平时将对应 SDA 置位即可。

主机接收数据时，利用双层循环外层循环 8 次表示接收数据所在一个字节的位数，内层循环 6 次遍历所有编码器，然后将所接收的数据从最高位起右移当前位数存入作为存储数据的数组。

#### 1.5.3.2.2 陀螺仪版

该版控制器选择使用 c 板上陀螺仪并配合惯性导航算法进行数据读取，并将数据处理后发给机器人。

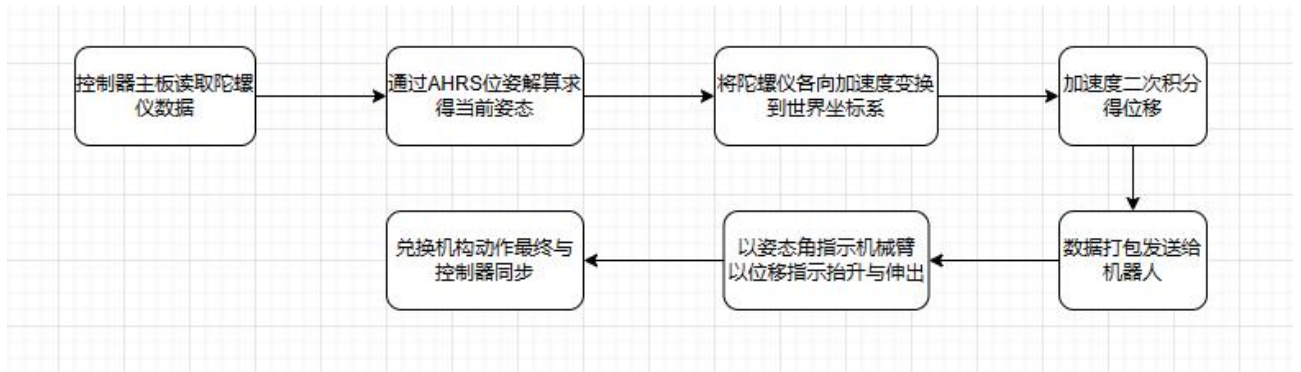


图 32 陀螺仪控制器流程图

该版控制器优点在于仅需要陀螺仪模块与单片机，相比编码器版不依赖于机械结构且更加小型化。缺点在于陀螺仪所测得的加速度存在微小误差，利用其二次积分得到的位移误差视姿态的不同而变化且会不断累积，稳定性不如编码控制器。

## 1.5.4 算法设计

### 1.5.4.1 自主导航算法

#### 1.5.4.1.1 导航部分

##### 导航算法简介与流程

运行环境：NUC 一台，安装 Ubuntu20.04 系统，Eigen3 线性代数库，ROS1 系统（noetic 版本），pcl 点云库，mid-360 激光雷达，以及 ros1 的 serial 功能包，LivoxSDK2 雷达驱动和 Livox\_ros\_driver2 雷达驱动功能包。

导航算法在 NUC 安装的 ROS1 系统下运行，包含定位，规划，建图部分以及若干个功能节点，程序运行时需要通过接收激光雷达发出的点云进行定位，再通过 SLAM 算法解算出的里程计配合先验地图进行路径规划，实现导航流程。

##### 程序文件结构

awakelion\_navigation

├─ .catkin\_workspace

├─ .vscode

| └─ c\_cpp\_properties.json

| └─ settings.json

└─ src

├─ .vscode

| └─ c\_cpp\_properties.json

- | └─ settings.json
- | └─ CMakeLists.txt
- | └─ FAST\_LIO: 开源 3d 激光雷达 SLAM 算法 fast\_lio
- | └─ Point-LIO: 开源 3 激光雷达 SLAM 算法 point\_lio
- | └─ .vscode
  - | └─ c\_cpp\_properties.json
  - | └─ settings.json
- | └─ livox\_repub: 点云消息格式转发功能包
  - | └─ CMakeLists.txt
  - | └─ livox\_repub.cpp
  - | └─ livox\_repub.launch
  - | └─ package.xml
  - | └─ readme.md
- | └─ livox\_ros\_driver2: mid-360 官方 ROS 驱动
- | └─ navigation: ROS 官方功能包集
- | └─ navigation\_build.sh: 编译脚本
- | └─ robot\_navigation: 机器人导航功能节点与配置功能包
  - | └─ .vscode
    - | └─ c\_cpp\_properties.json
    - | └─ settings.json
  - | └─ CMakeLists.txt
  - | └─ include
    - | └─ sign\_point.h
  - | └─ launch
    - | └─ lidar\_start.launch
    - | └─ mapping.launch
    - | └─ move\_base.launch
    - | └─ robot\_model\_visualization.launch



```
| └─ sentry_navigation_competition.launch
|   └─ sentry_start.launch
└─ map
    │ └─ rmuc.pgm
    │   └─ rmuc.yaml
    └─ package.xml
└─ params
    │ └─ costmap_common_params.yaml
    │ └─ eband_local_planner_params.yaml
    │ └─ global_costmap_params.yaml
    │ └─ global_planner_params.yaml
    │ └─ local_costmap_params.yaml
    │ └─ move_base_params.yaml
    │   └─ recovery_behaviors.yaml
└─ src
    │ └─ odom_trans.cpp
    │ └─ pointcloud_filter.cpp
    │   └─ sign_point.cpp
└─ urdf
    └─ mini_mec_robot.urdf
```

## 部分算法原理阐述

3d 激光雷达 SLAM 算法采用开源算法 Point\_Lio，该算法会创建一个程序运行时的初始系 camera\_init 和解算出的里程计 aft\_mapped 系，由于 aft\_mapped 系不能直接用作机器人的底盘系，故通过 odom\_trans 节点建立 camera\_init 系与底盘 base\_link 系之间的动态 tf 变换，实现定位，进而用于路径规划。

```

static tf::TransformBroadcaster odom_broadcaster;
tf::Transform transform;
tf::Quaternion q(msg->pose.pose.orientation.x, msg->pose.pose.orientation.y, msg->pose.pose.orientation.z, msg->pose.pose.orientation.w);
transform.setOrigin(tf::Vector3(msg->pose.pose.position.x, msg->pose.pose.position.y, msg->pose.pose.position.z));
transform.setRotation(q);
// // 将雷达里程转换为底盘里程然后发布到tf关系中
tf::Transform laser_transform;
laser_transform.setOrigin(tf::Vector3(trans_x, trans_y, trans_z)); //设置在父坐标系下子坐标系的原点坐标（即平移向量）
laser_transform.setRotation(tf::Quaternion(rot_x, rot_y, rot_z, rot_w)); //设置在父坐标系下子坐标系的旋转变换（旋转矩阵）
transform = transform * laser_transform;
odom_broadcaster.sendTransform(tf::StampedTransform(transform, msg->header.stamp, "camera_init", "base_link")); //后两个参数分别为父坐标系，子坐标系

```

图 33 部分代码展示

### 1.5.4.1.2 决策部分

#### 决策算法简介与流程

运行环境：NUC 一台，安装 Ubuntu20.04 系统，ROS1 系统（noetic 版本），ros1 的 behaviortree-cpp-v3 功能包，ros1 的 serial 功能包，基于 BehaviorTree.CPP 的框架实现。

流程：通过接收导航实现最后发送的消息、下位机单片机发来的裁判系统消息以及当前机器人的位姿反馈生成 blackboard，以便各个行为树子节点能进行逻辑的判断。行为树从树节点开始运行进入 Sequence 序列或者 fallback 序列执行一系列操作，通过导航接收然后返回一个当前需要前进的线速度或者角速度，然后通过串口通信发送，实现对工程机器人的控制。

#### 程序文件结构

awakenlion\_behaviour\_tree

src

- ├─ CMakeLists.txt
- ├─ rm\_decision
  - │ ├─ CMakeLists.txt
  - │ ├─ config
    - │ │ └─ main\_tree.xml : 行为树的整体框架
  - │ ├─ include
    - │ │ └─ rm\_decision
      - │ │ │ └─ add\_blood.h : 没血补血节点
      - │ │ │ └─ blackboard.h : 储存数据交流的节点
      - │ │ │ └─ dense.h : 防御模式
      - │ │ │ └─ exchange.h : 兑矿操作
      - │ │ │ └─ fetch\_mine.h : 抓矿操作
      - │ │ │ └─ game\_start.h : 比赛开始，作出行为树开始的初始节点

- | | | └─ get\_point.h : 对操作手发送点进行处理
- | | | └─ patrol.h : 没有任何任务时的自动操作
- | | └─ launch
- | | | └─ all\_server.launch : 测试消息与行为树运行状态
- | | | └─ run\_tree.launch : 启动行为树各节点程序
- | | └─ package.xml
- | | └─ src
- | | | └─ decision\_node.cpp : 数据的传入和传出的数据调度
- | | | └─ tree\_node.cpp : 树节点的使用和注册
- | └─ rm\_msgs
- | | └─ CMakeLists.txt
- | | └─ msg : 行为树各子节点的消息接收与输出
- | | | └─ behaviour\_blackboard
- | | | | └─ add\_blood\_rx.msg
- | | | | └─ dense\_rx.msg
- | | | | └─ exchange\_rx.msg
- | | | | └─ fetch\_mine\_rx.msg
- | | | | └─ game\_start\_rx.msg
- | | | | └─ get\_point\_rx.msg
- | | | | └─ patrol\_rx.msg
- | | | └─ behaviour\_output
- | | | | └─ add\_blood\_tx.msg
- | | | | └─ dense\_tx.msg
- | | | | └─ exchange\_tx.msg
- | | | | └─ fetch\_mine\_tx.msg
- | | | | └─ game\_start\_tx.msg
- | | | | └─ get\_point\_tx.msg
- | | | | └─ patrol\_tx.msg

```

| | └─ rule
| |   └─ GameBuff.msg
| |   └─ GameResult.msg
| |   └─ GameRobotHP.msg
| |   └─ GameStatus.msg
| |   └─ GameZone.msg
| |   └─ RobotDamage.msg
| |   └─ RobotStatus.msg
| └─ package.xml
| └─ src
└─ robot_driver
  └─ .vscode
    └─ c_cpp_properties.json
    └─ settings.json
  └─ CMakeLists.txt
  └─ include
    └─ robot_driver
      └─ MYAPI.h : ROS 的 API 使用
      └─ msgSerializer
        └─ check.h : CRC 校验
      └─ msg_serialize.h : 发送接收的数据结构体
        └─ serialPort.h : serialPort 类的使用
  └─ launch
    └─ robot_driver.launch : 串口程序的开启
  └─ msg
    └─ vision_rx_data.msg
    └─ vision_tx_data.msg
  └─ package.xml

```

```

|   └─ src
|   |   └─ .vscode
|   |       └─ c_cpp_properties.json
|   |           └─ settings.json
|   |               └─ main.cpp : 对串口接收与发送数据的处理
|   |                   └─ serialPort.cpp : 串口的开启与配置
|   |                       └─ yaml
|   |                           └─ config.yaml : 串口用到的部分参数
└─ serial : ros 的串口包

```

而针对行为树的整体架构设计，其框架的 main.xml 大概为：

```

<root main_tree_to_execute="MainTree">
  <BehaviorTree ID="MainTree">
    <Fallback>
      <GameStart/
      <ReactiveSequence>
        <AddBlood/
        <GetPoint/
        <Sequence>
          <FetchMine/
          <Exchange/
        </Sequence>
      </ReactiveSequence>
      <Patrol/
    </Fallback>
  </BehaviorTree>
</root>

```

## 部分算法原理阐述

Sequence

其中 Sequence（序列）是其中一种常用的节点类型，表示子节点按顺序执行，如果有任何一个子节点返回失败，则整个 Sequence 节点也返回失败，只有当所有子节点都返回成功才会返回成功。

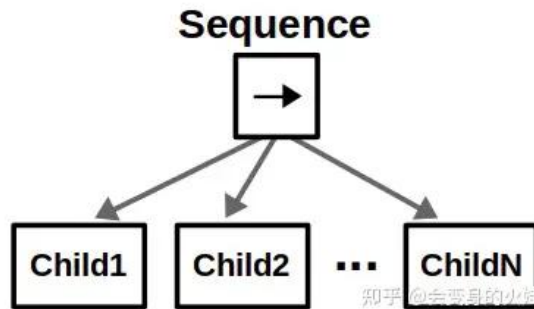


图 34 Sequence 节点

以下是 Sequence 节点的伪代码实现：

```
function run() {
    for each child in children {
        status = child.run();
        if status == FAILURE {
            return FAILURE;
        }
    }
    return SUCCESS;
}
```

在上面的伪代码中，SequenceNode 包含一个子节点列表 children，在运行时会依次执行每个子节点的 run 方法。如果其中一个子节点返回 FAILURE，则整个 SequenceNode 也会返回 FAILURE，只有当所有子节点都返回 SUCCESS，整个 SequenceNode 才会返回 SUCCESS。

除了普通的 Sequence 节点，还有一种称为 Reactive Sequence（反应式序列）节点。这两者之间的主要区别在于执行方式和逻辑。普通的 Sequence 节点是按顺序执行其所有子节点，只有当所有子节点依次返回成功时，Sequence 节点才会返回成功；一旦有任何一个子节点返回失败，则整个 Sequence 节点也会返回失败。这种方式适合描述需要依次执行一系列动作的情况。

而 Reactive Sequence 节点则更加灵活，它在每次更新时都会从头开始执行子节点，直到找到一个返回成功的子节点为止，然后停止执行并返回成功；如果所有子节点都返回失败，则 Reactive Sequence 节点才会返回失败。这种方式适合处理需要快速响应某些事件或条件的情况，能够更加及时地执行相应的动作。比如说当工程机器人处于取矿或者兑矿，有机器人攻击它并使它低于正常的血量濒临死亡时，可以及时反应并回到补血区，能够更好地发挥工程机器人的作用并最终保证队伍的经济，该反应式序列也可以

及时对系统的不稳定状态进行调整，及时中断处于死循环的任务。

## fallback

Fallback（回退）节点是行为树中另一个常用的节点类型，用于处理子节点的优先级逻辑，如果一个子节点执行失败，可以尝试执行下一个子节点，直到找到一个成功的节点为止。

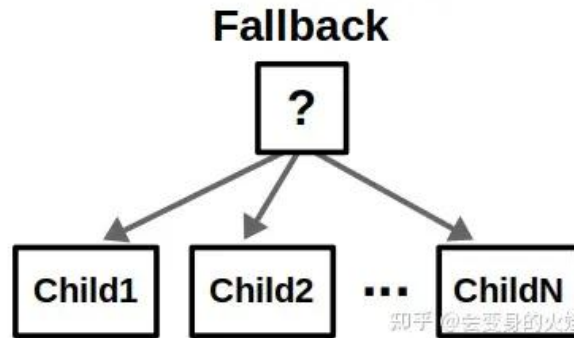


图 35 Fallback 节点

以下是 Fallback 节点的伪代码实现：

```

function run() {
  for each child in children {
    status = child.run();
  }
  if status == SUCCESS {
    return SUCCESS; }
  return FAILURE; }

```

在上面的伪代码中，FallbackNode 包含一个子节点列表 children，在运行时会依次执行每个子节点的 run 方法。如果其中一个子节点返回 SUCCESS，则整个 FallbackNode 也会返回 SUCCESS，否则会继续执行下一个子节点，直到所有子节点都返回 FAILURE，最终 FallbackNode 返回 FAILURE。这种机制使得程序能够根据子节点的优先级顺序来执行相应的逻辑。

## tick

类似于数据结构中“tree”的概念，行为树是控制任务执行流程的分层节点树。一个称为“tick”的信号，由开发者发送到树的根部并在树的节点中传播，直到它到达树的底部——叶节点。接收到 tick 信号的节点会执行它的回调，即被设定的该节点的任务，然后向上返回执行的结果。在大多数开源库中，该结果有且只有 3 种：成功完成 SUCCESS，执行失败 FAILURE，正在执行 RUNNING。当一个行为树在运行时，会被周期性地 tick 操作，以触发行为树中各个节点的运行逻辑，从而实现对角色行为的控制。

在进行 tick 操作时，行为树会从根节点开始，按照节点类型和逻辑依次调用每个节点的执行方法，

直到达到叶子节点（行为节点），然后执行具体的行为逻辑。在这个过程中，每个节点可能会根据其自身逻辑和状态返回不同的结果（比如成功、失败、运行中等），从而影响整个行为树的执行流程。

通过不断进行 tick 操作，行为树可以根据当前的环境和条件动态地调整角色的行为，使得角色能够根据情况作出相应的反应和决策。tick 是行为树运行和更新的基本单位，也是实现行为树逻辑的核心概念之一。

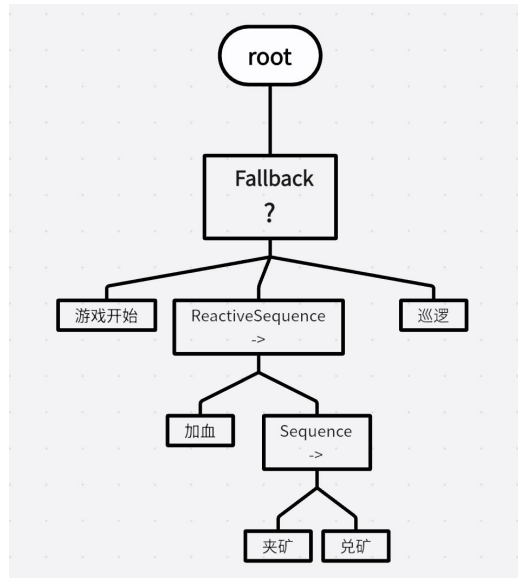


图 36 行为树框架



## 1.6 研发迭代过程

### 1.6.1 测试记录

#### 1.6.1.1 伸出测试

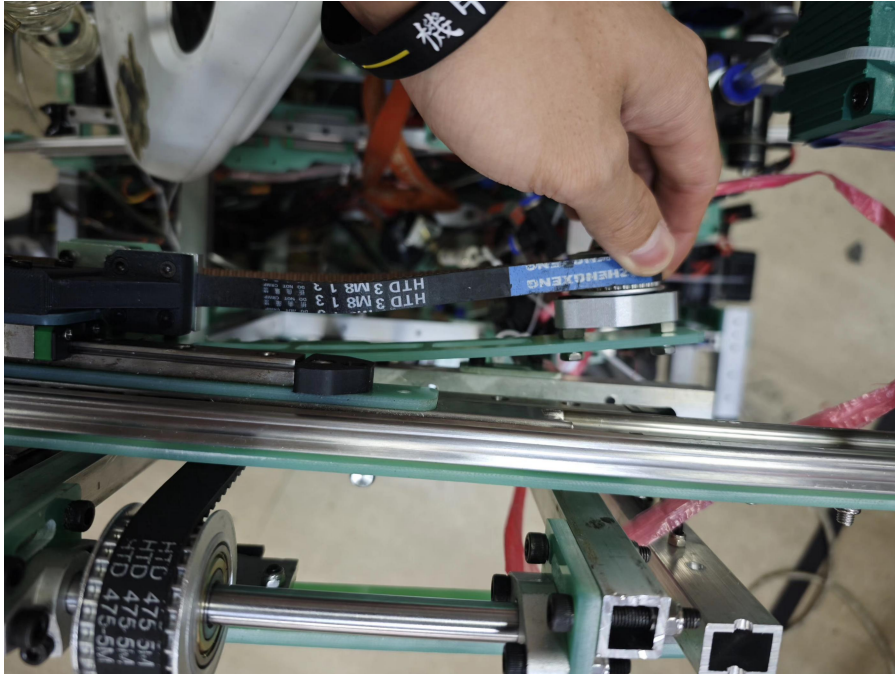


图 37 伸出齿条形变

一二代车在伸出齿条上的设计强度过低，导致齿条在运动过程中形变的概率很高，十分影响伸出的效率，以及会导致左右伸出不一样的角度。经观察存在一定的装配问题，但过于激进的设计同样需要修改。

经多次修改，上述出现问题已经解决，但由于多次安装拆除，以及装配技术问题，导致安装精度较低，龙门架伸出出现轻微外八，导致伸出，回收时出现卡顿现象。抽屉滑轨也损坏严重。

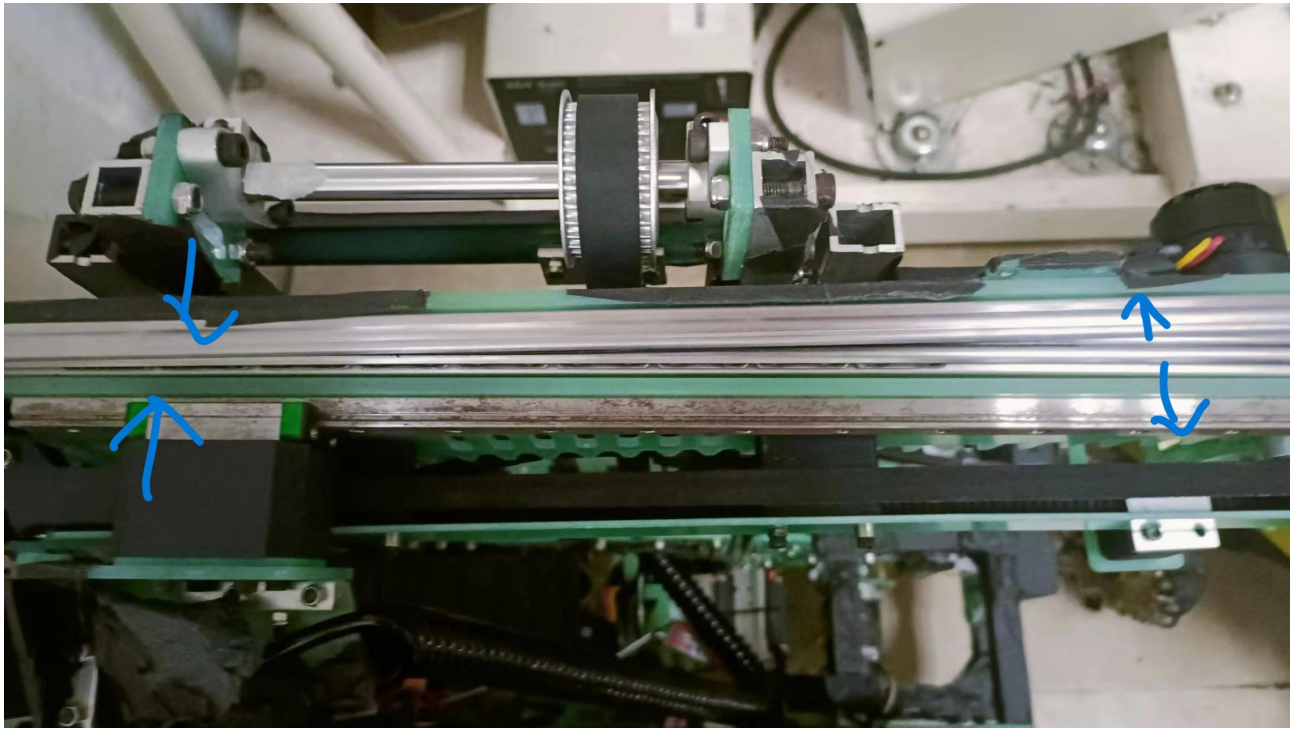


图 38 二代车龙门架外八导致抽屉滑轨严重变形损坏

### 1.6.1.2 抬升测试

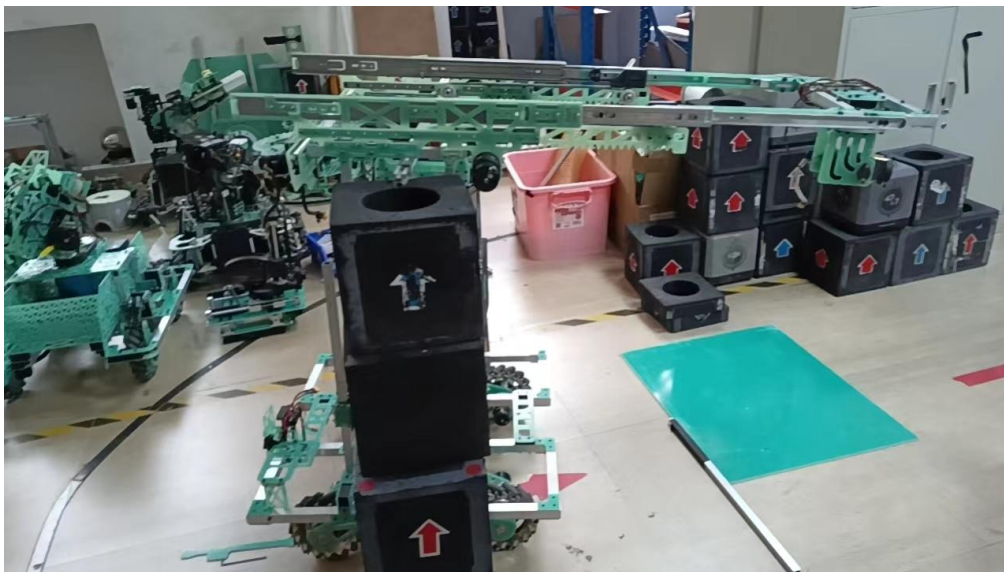


图 39 一代车云台塌陷

一代车在一二级抬升的固定上存在较大的设计缺陷，导致支撑杆形变严重，导致伸出末端下陷。

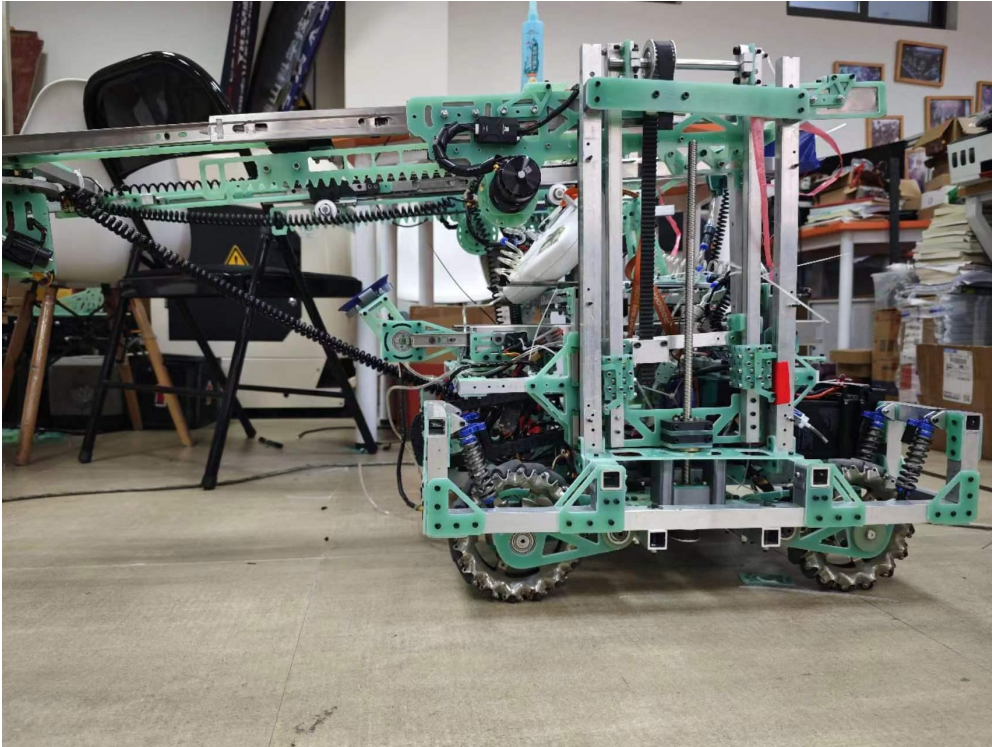


图 40 二代车云台伸出塌陷

二代车交换了丝杆与滑车的位置，抬升一级抬升支撑杆不会形变，相较于一代车有改善，但是由于二级抬升使用了滑车，其本身具有一定的虚位，导致依旧未能解决云台塌陷的问题，后续将滑车换成线轨。

第一将滑车改换为线轨，云台塌陷问题有所好转但仍然存在一定的下沉导致无法达到兑换要求高度，且由于塌陷问题，单根铝方管使用玻纤板夹铝方管固定在滑块的方式经常会出现松动导致云台出现严重摇晃。

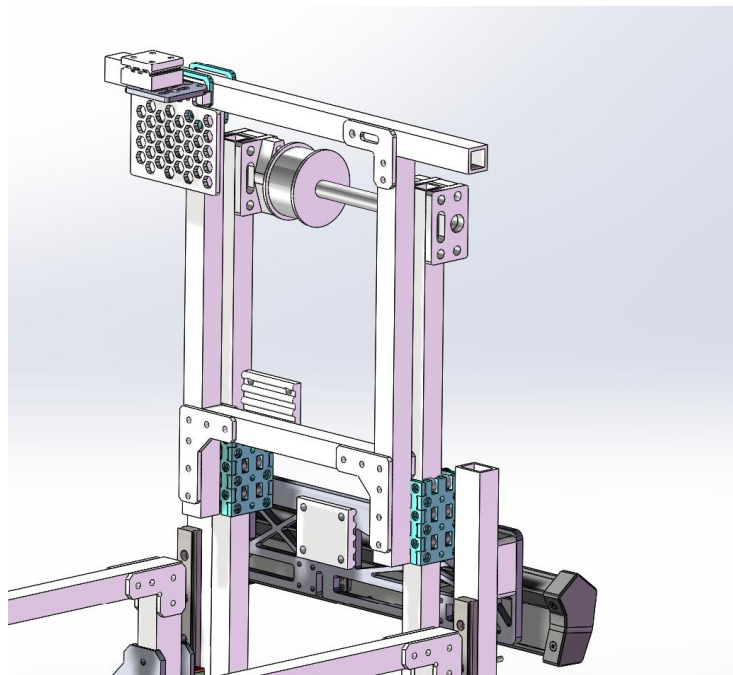


图 41 一代抬升图

二次修改，在来的基础上两侧各增加一条线轨，将玻纤板夹铝方管固定在滑块的方式改为玻纤板夹打

印件，打印件连接铝方管，再使用内三角形形状镂空的玻纤板连接同一侧铝方管的结构。一定程度实现二级抬升安装加固且方便日常维护。

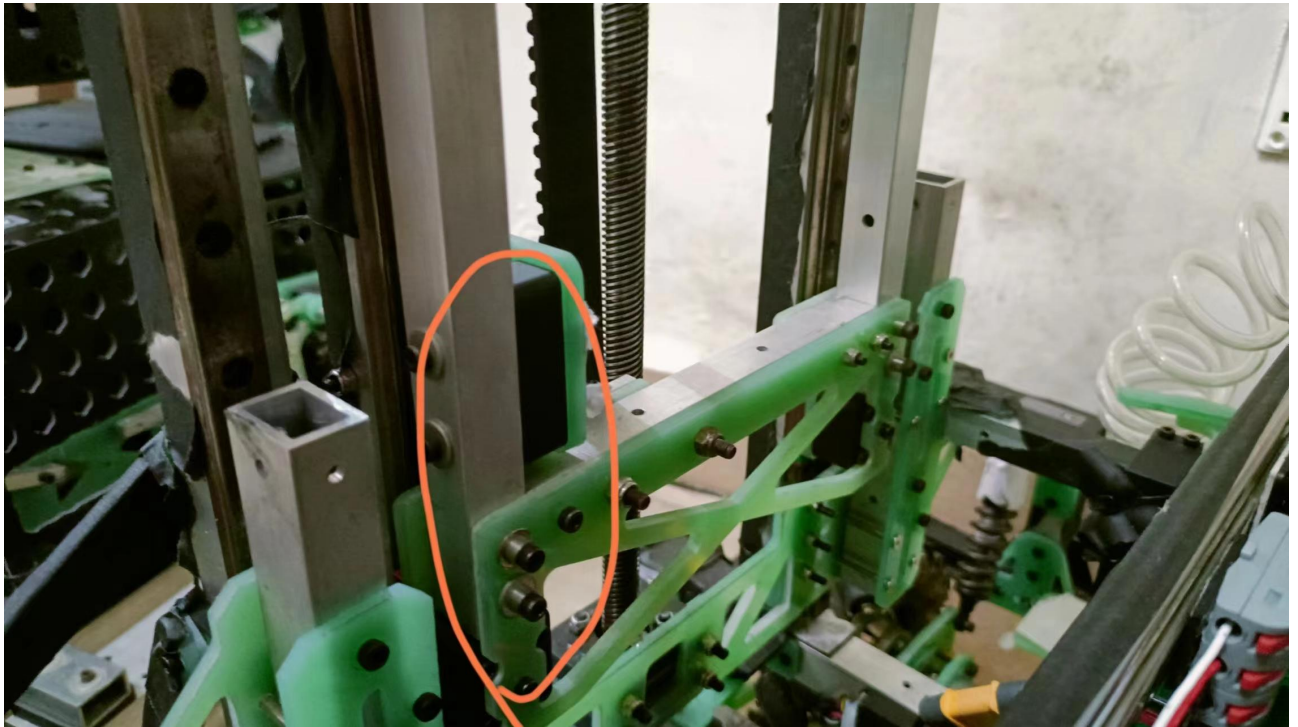


图 42 加固后抬升图

### 1.6.1.3 底盘避震测试



图 43 一代车避震塌陷

工程一代车避震方向问题导致前轮无法承受伸出后的重心变化，二代车修改了避震方向，解决了这个问题。

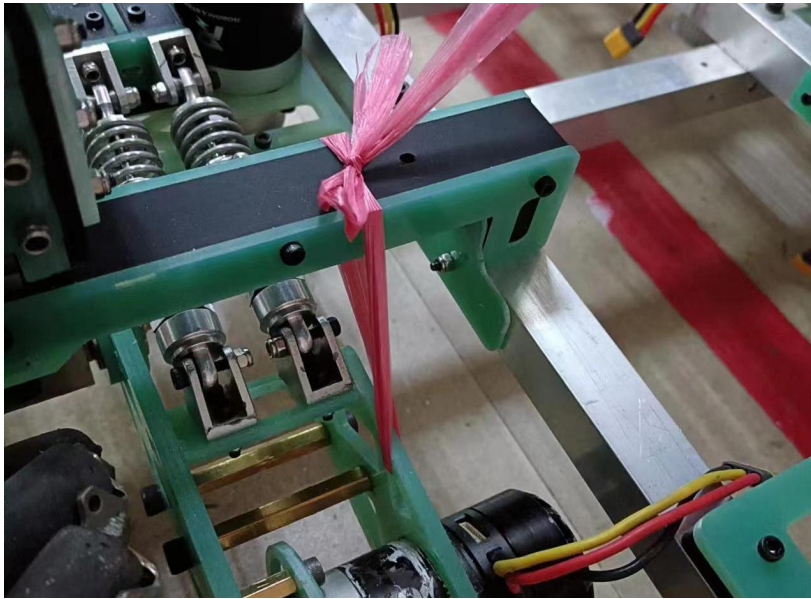


图 44 避震塌陷尝试解决

一代车尝试过解决这个问题，方案为支撑前避震，拉住后避震，在一定程度上缓解了，但是依旧是重心偏移过多，没有很好的效果。

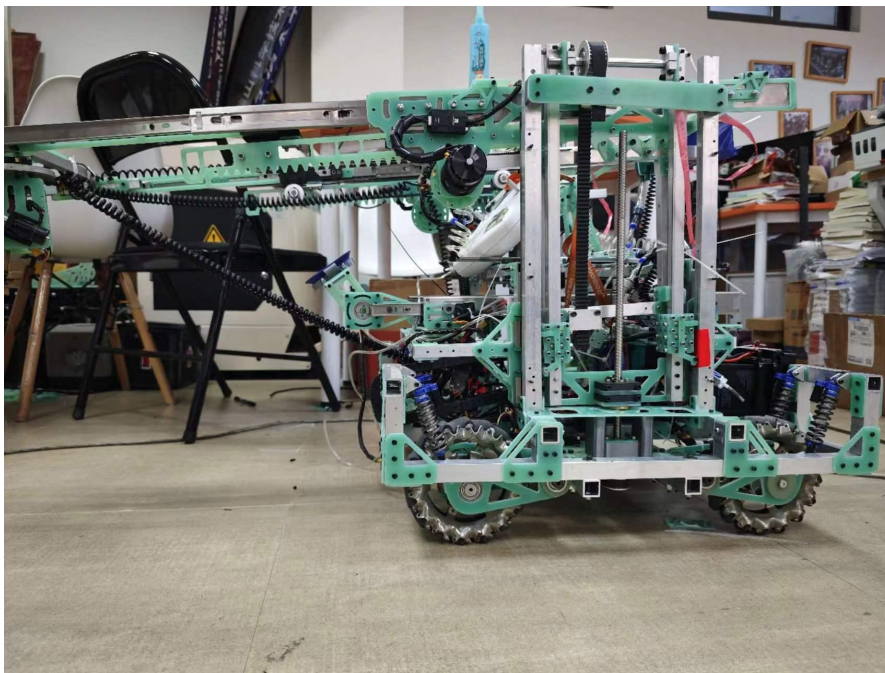


图 45 二代车避震塌陷程度展示

#### 1.6.1.4 辅助夹取机构



图 46 侧向夹取机构塌陷

本赛季大资源岛机制的修改，使得自动化动作的优势变得很高，所以我们希望通过设计一个辅助夹取机构，以提高工程机器人运行的效率。

一代车侧向夹取机构采用滑轨，未考虑到滑轨受力范围，导致整个机构有很严重的下陷。

二代车直接舍弃了侧向夹取的机构，换成正向的，设计上轻便了很多，也不会出现下陷的情况。

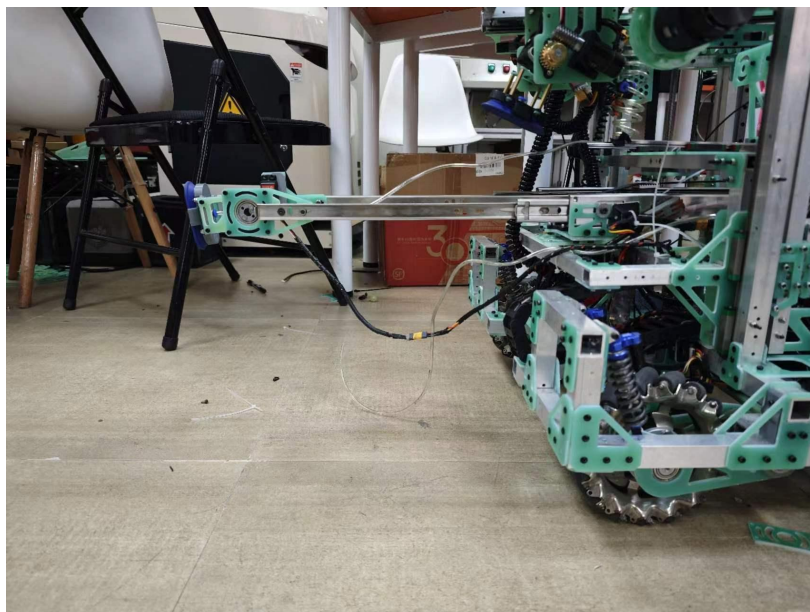


图 47 二代车正向夹取机构

二代车辅助夹取结构相比于一代提升效果明显，但由于设计时未能考虑到力臂过长的问题，结构后面未设计好安装的方式，导致在机械臂伸出过程或吸住矿石时，该机构后方会出现翘起，及后电机安装位置变形导致同步带滑齿，无法有效保持矿石姿态稳定。

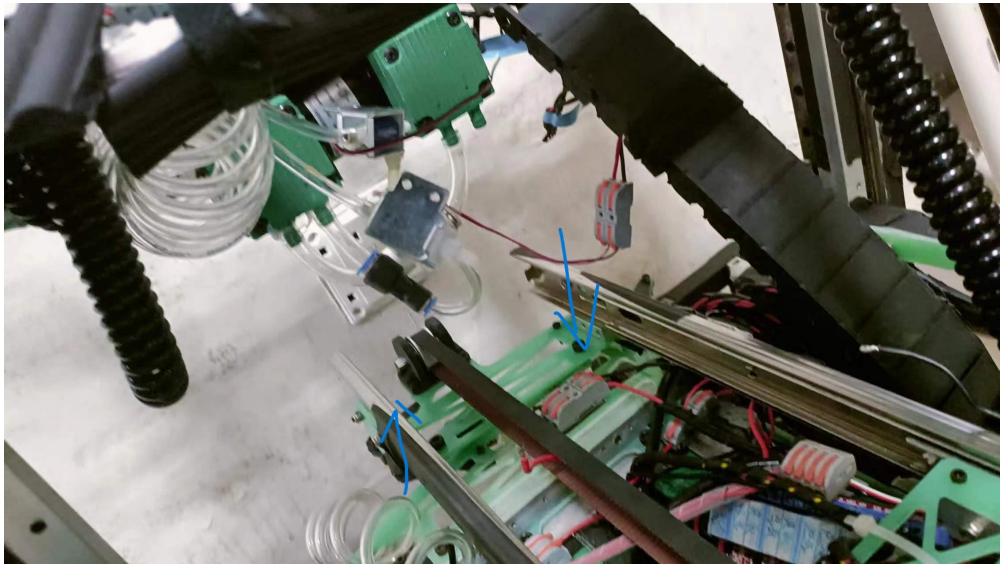


图 48 辅助夹取机构翘起部分示意图

后经讨论决定在后方下面增加一连杆来辅助固定左侧翘起部分。翘起有所改善但由于最初设计本身存在较大缺陷无法进行更有效的加固方式，后期仍然存在一点翘起。

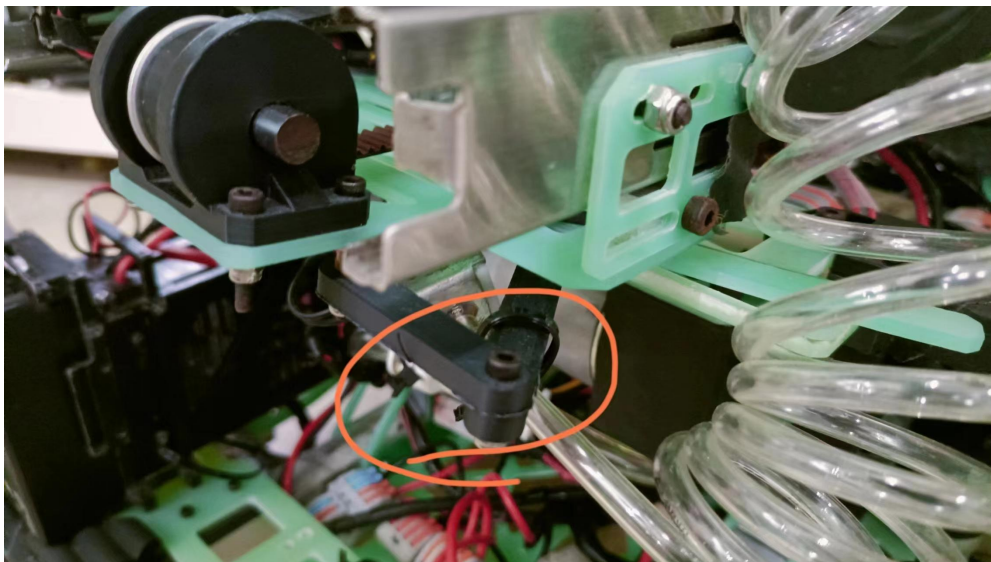


图 49 解决该问题所新增机构

### 1.6.1.5 兑换用机械臂

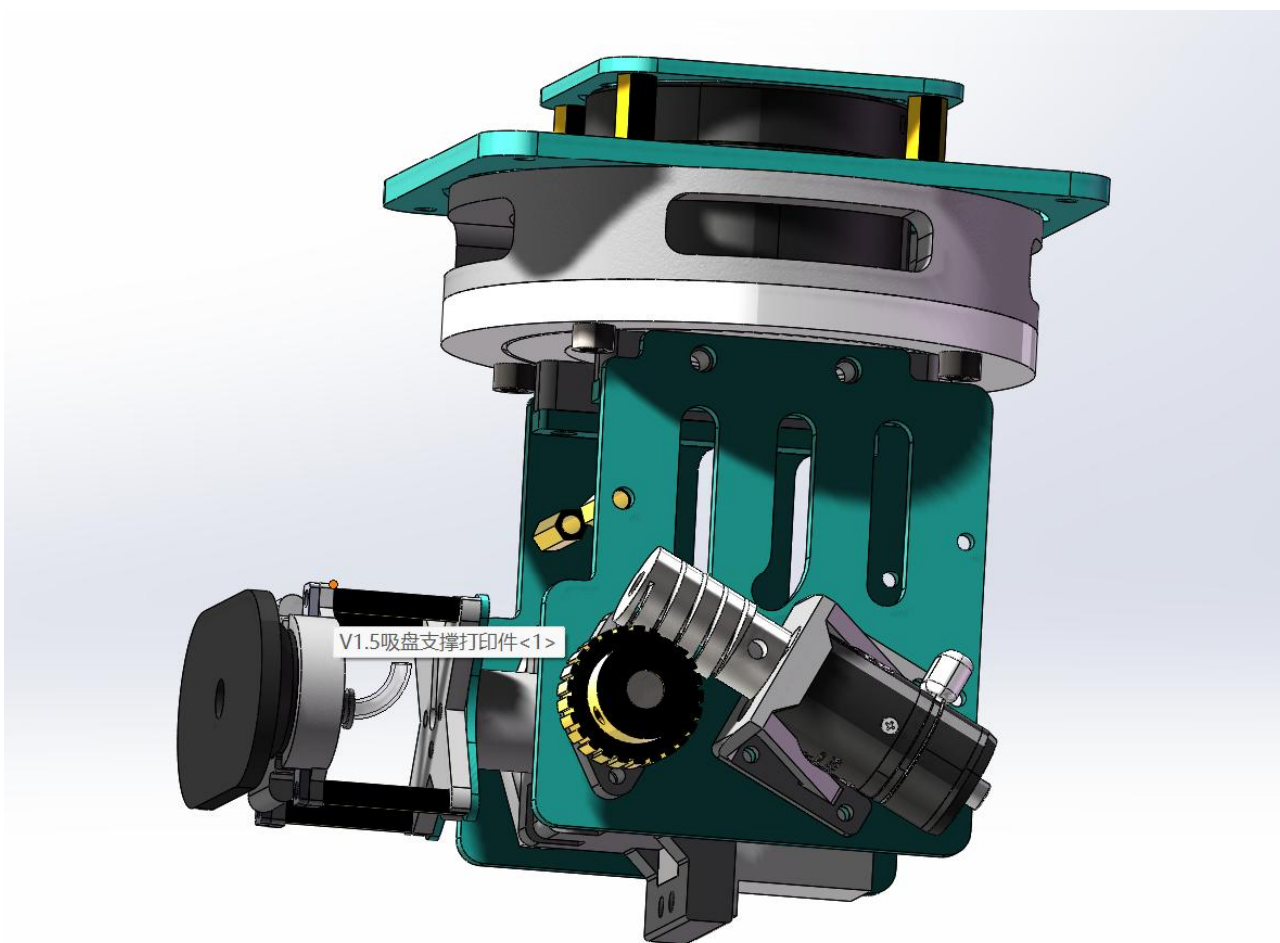


图 50 一代机械臂

一代机械臂能够实现矿石的五级兑换难度，但由于抬升机构的设计缺陷，缺乏转矿机构导致兑换时无法保证矿石二维码正确朝向。这一代兑换用的小型机械臂还存在以下小问题；

Pitch 轴所有 2006 电机配合蜗轮蜗杆传动，将扭矩放大；使用打印件制电机架。这样设计机械臂更轻量化，更小巧空间利用率更高；但经测试发现，扭矩过大，打印件制的电机架无法承受会发生断裂，后修改打印件加固。但仍存在轻微形变基本不影响。





图 51 断裂的电机架

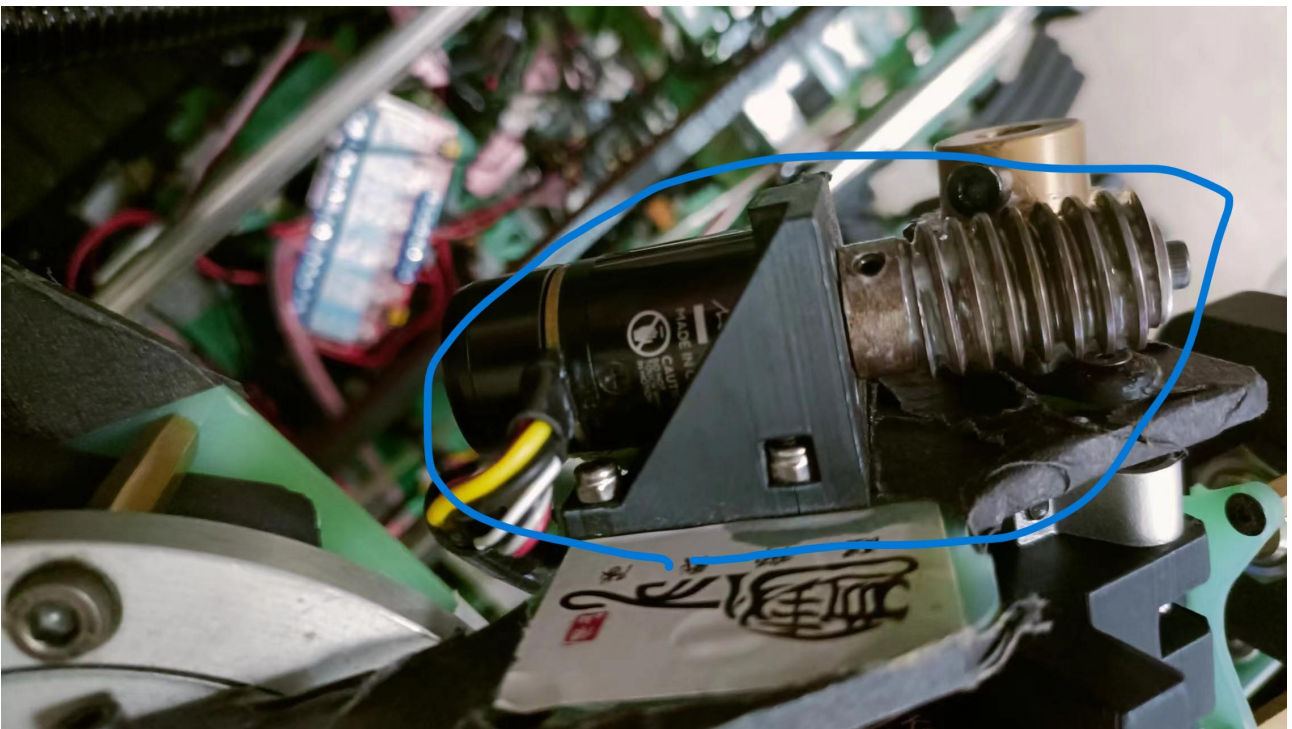


图 52 加固新电机架

新电机架，在原来的基础上进行加固并改善安装孔位，使得安装更加牢固，强度更高。

由于蜗轮蜗杆是采用单边固定，在较大负载下会出现跳齿长时间的使用会导致蜗轮蜗杆间隙增大无法有效啮合，在设计初没有考虑到这一问题，因此并有很好的解决方案，而且临近比赛后降低出现调整电机安装角度并加固蜗轮蜗杆轴向固定，有一定改善但长时间使用依然会出现跳齿。

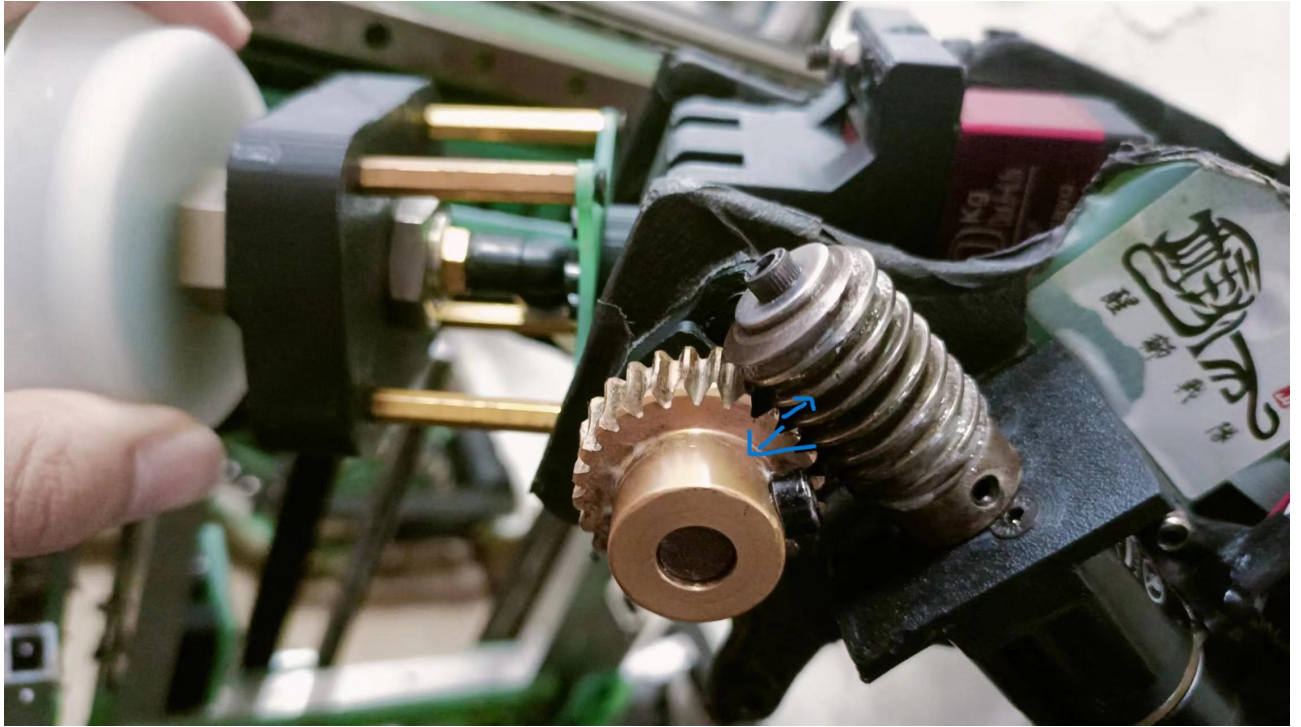


图 53 蜗轮蜗杆长期使用导致间隙增大无法正常啮合

二代机械臂：取得矿石并保证矿石二维码朝向正确，二代机械臂在原机械臂的基础上进行较大改动，将理论最高兑换难度降低到四级为代价来满足这一需求。

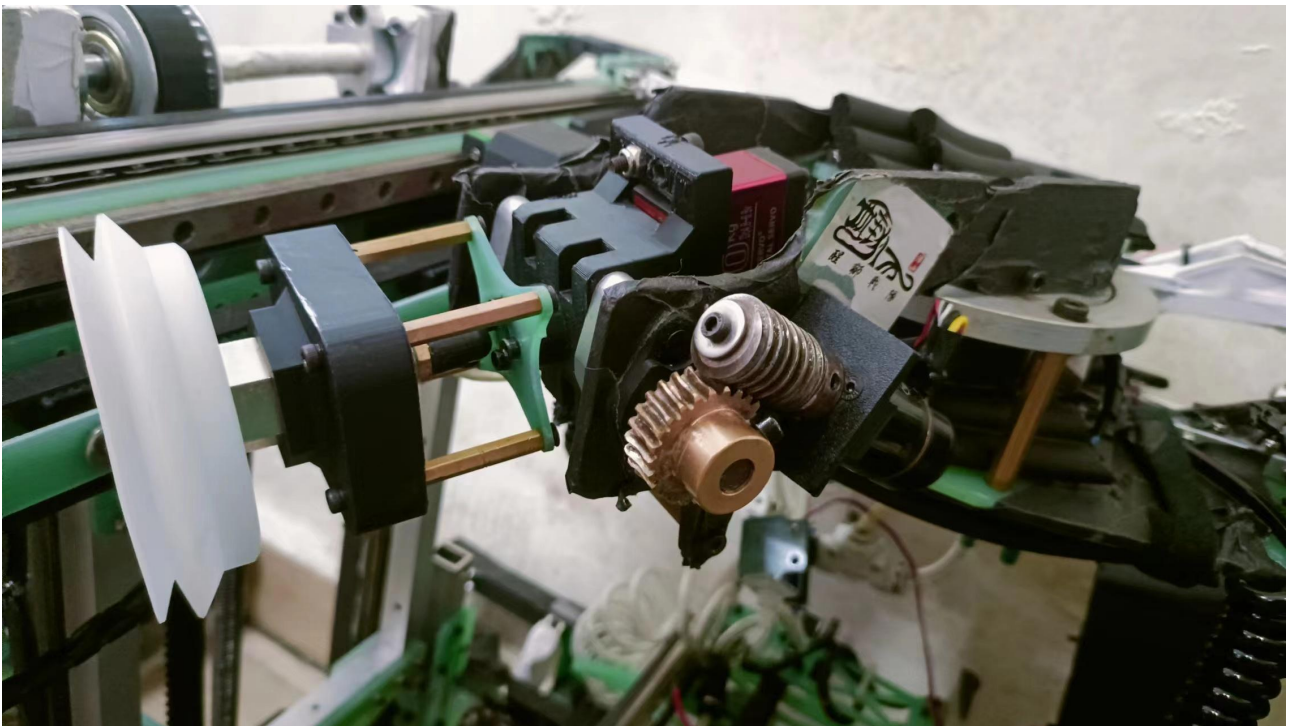


图 54 二代机械臂实物图

Yaw 轴最大旋转角度有正负 180 度，调整为正负 100 度，其他自由度基本不变，但机械臂受限于伸出机构的设计问题，YAW 轴只能在特定的伸出量后才有比较大的移动范围，这点十分致命。

## 1.6.2 版本迭代过程记录

表 4 版本迭代记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	一代车完成，能实现夹地，夹取，兑换等基础功能	2024.1.25
V1.1	解决一代车避震塌陷问题	2024.1.26
v2.0	二代车完成，优化了辅助夹取、伸出、抬升等机构	2024.3.25
v2.0	优化二代二级抬升机构	2024.4.15
V2.0	自定义控制器控制机器人	2024.4.5
v2.0	解决了代码繁杂的问题，对所有电机进行统一配置	2024.4.20
v2.0	优化了整体的布线，大大减少弹簧线的数量	2024.4.26
v2.0	实现吸盘的立即断气以及吸气	2024.5.3
v2.1	解决兑换机械臂能夹取并保证矿石二维码朝向的正确	2024.5.20

## 1.6.3 重点问题解决记录

表 5 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	避震下陷	避震设计方向有问题	采用前顶后拉，效果一般，需要重新设计	V1.0	机械工程师：陈健安
2	一级抬升晃动大	设计缺陷	将一二级抬升的滑车与丝杆位置互换，已解决	V1.0	机械工程师：陈健安
4	二级抬升晃动，安装不牢固	滑车存在缺陷；未考虑到机械臂重量较大；	将滑车换为直线导轨并在此基础上两侧各增加一条直线	V2.0	机械工程师：陈健安

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
		伸出时力臂过长；	导轨，将同一侧二级抬升铝方管相连增加三角结构以达到加固；已解决 更改固定方式方便日常维护；已解决		
3	自定义控制器容易损坏	编码器固定方式有问题	加强编码器设计强度，已解决	V1.0	机械工程师：丁晓中
4	抬升机构同步带过松，导致滑齿	导轮较小，电机固定不稳，设计问题	更换大的导轮，加强电机固定，减缓电机旋转速度，有效缓解	V2.0	机械工程师：陈健安
5	辅助夹取机构后方翘起，导致无法稳定夹取并保持矿石姿态。	设计时未考虑到力臂过长	后下方增加连杆辅助固定	V2.0	机械工程师：丁晓中
6	布线混乱，弹簧线过多	开发板位于底盘，云台所有线需要飞跃整个抬升	将布线仓移至一级抬升平台，减少弹簧线数量，已解决	V2.0	硬件工程师：黄皓杰
7	代码繁杂，写一个动作要在几个c文件中修改	代码将运动过程过于分化	优化代码，在motor_task.c中进行配置和	v2.0	软件工程师：凌浩天

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
			clamp_task.c 中进行夹取动作的设计, 已解决		
8	吸盘不能立即泄气	在大气压下泄气困难	增加一个电磁阀, 改变电磁阀的气管连接 已解决	V2.0	硬件工程师: 黄皓杰
9	二代兑换机械臂达不到取矿要求	二代机械臂在垂直方向上占了很多的空间, 导致无法稳定的在两个吸盘之间传递矿石。	修改二代机械臂结构, 取消五级兑换难度为代价, 保证可以夹取和传递矿石并保证矿石二维码朝向的正确	v2.0	硬件工程师: 黄皓杰
10	控制器连接裁判系统时接收不到数据	代码中发送数据包的长度有问题	将发送数据包的长度改为 30	v2.0	软件工程师: 凌浩天

## 1.7 团队成员贡献

表 6 团队成员贡献

姓名	基本信息 (专业、年级、队内角色)	主要负责工作内容描述	贡献度 (所有成员贡献度合计为 100%)
陈健安	机械电子工程、大二、机器人机械机构设计、维护及制造	负责整个机器人的设计、维护以及改进	30%

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%)
凌浩天	电气工程及其自动化, 大二, 机器人和自定义控制器代码设计、调试以及编写	机器人和自定义控制器代码设计以及编写	30%
黄皓杰	电气工程及其自动化, 大二, 机器人机械机构设计方向、机器人电气布线以及调试	机器人机械机构设计方向、机器人电气布线、维护机器人以及调参	20%
丁晓中	机械电子工程、大一、自定义控制器机械机构设计以及制造	自定义控制器机械机构设计以及制造	15%
黄烨	动物医学、大一、机器人制造	机器人制造、备料	5%

## 1.8 参考文献

[1] 参考上海大学 SRM 战队工程自定义控制器开源文档, A 板上没有 RS232 母头, 因此我们需要一个 ttl 转 rs232 模块以及一个 rs232 公头转 USB 数据线

<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22907&fromuid=86517>

(出处: RoboMaster)

[2] 【RM2023-工程机器人上层结构开源】华南理工大学-华南虎

<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22806&fromuid=86517>

(出处: RoboMaster)

[3] 【RM2024 赛季-自定义控制器开源】福建师范大学 Pikachu 战队

<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22953&fromuid=86517>

(出处: RoboMaster)

[4] 【RM2023-工程开源】东北大学 TDT-技术报告

<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22761&fromuid=86517>

（出处：RoboMaster）

[5] 【RM2022-工程机器人机械结构开源】广东工业大学 DynamicX

<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22169&fromuid=86517>

（出处：RoboMaster）

## 1.9 技术方案复盘

### 1.9.1 赛场性能表现情况分析

#### 工程机器人赛季初的目标：

能够稳定地夹取金矿银矿并同时存储一颗矿石，通过自定义控制器辅助实现较高难度的兑换，并且尝试开局抢夺金矿，为战队提供充足的经济。

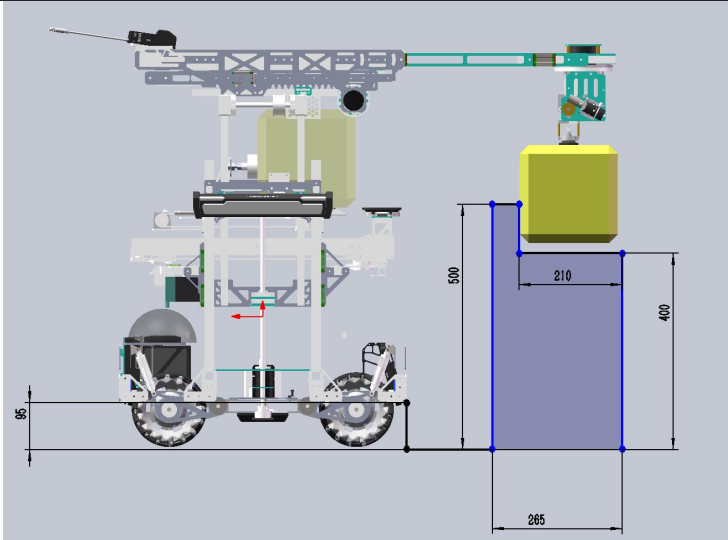
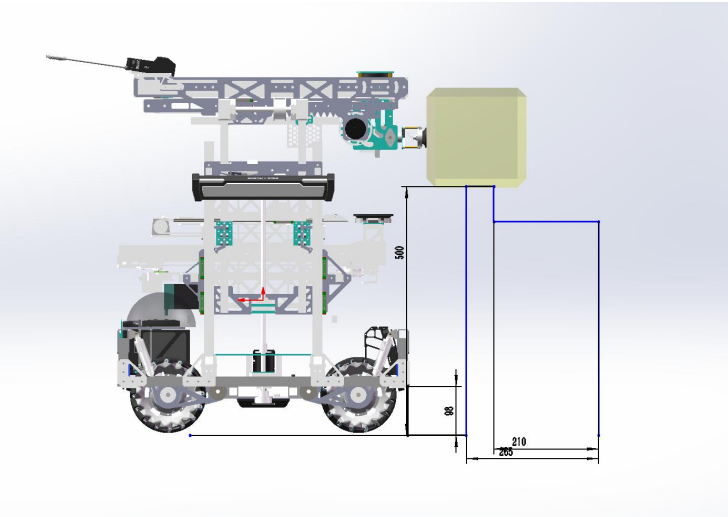
#### 实际比赛中：

1. 所有预先设置好的取矿动作，都与场地有一定的干涉，导致取矿成功率很低。
2. 在某些轴上的动力传输依然出现跳齿等现象，导致机器人性能无法完全发挥出来。
3. 图传视野十分受限，影响操作。
4. 机械结构强度不足，结构形变以及损害情况多发。
5. 自定义控制器延时太高，反而不利于操作。
6. 吸盘吸力不足，导致已经夹取到的矿石因车辆晃动又掉落。
7. 机械结构设计不合理，末端小三轴机械臂运动角度十分受限。

### 1.9.2 赛场性能表现与规划对比分析

表 7 赛场实现功能

实现的功能	如何实现
夹取小资源岛	<p>适应性训练赛发现在夹取小资源岛上，第一方案是从正面夹取，但是会因为矿石在末端有一定的下坠量，导致难以从槽内拔出，成功率较低。</p> <p>后面比赛更换了方案，采用两次动作夹取，比较耗时，但是成功率明显提高，以下为方案二的夹取示意，先从上表面将矿石取出，将矿石放到小资源岛的其他平面上，再从正面夹取。</p>

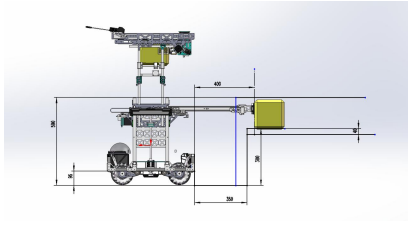
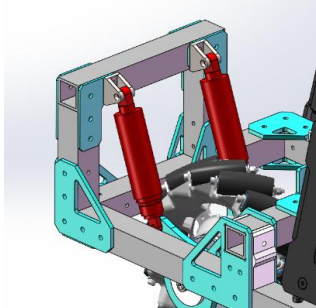
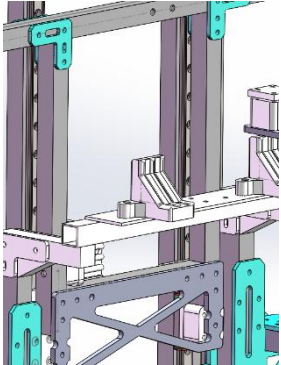
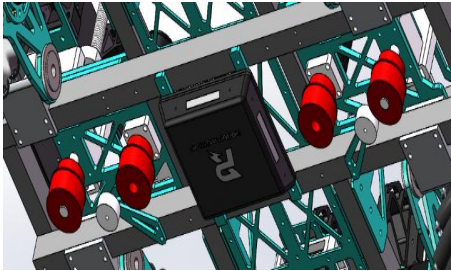
实现的功能	如何实现
	 <p style="text-align: center;">一次夹取</p>  <p style="text-align: center;">二次夹取</p>
完成高难度矿石兑换	<p>这一代机器人可以稳定地兑换四级矿石，兑换一颗矿石需要时间约为 30s，主要得益于三轴机械臂在没有自定义控制器的情况下，需要通过键鼠操作的自由度较少。</p>
机械臂六自由度可灵活控制	<p>利用底盘及龙门架机构实现 xyz 轴的移动，通过小型机械臂实现旋转姿态变化。</p>
吸盘机构配合电磁阀实现矿石牢固夹取与矿石瞬间掉落	<p>通过两个电磁阀同时控制以及改变了气管的连接方式，实现了吸盘可以达到瞬间吸气以及喷气，保证矿石夹取与掉落的动作能流畅进行</p>
代码框架的优化	<p>代码方面由原来的写一个动作要在多个文件中配置，改为统一在 clamp 这一个文件中配置，大大节省了写动作的时间。并且将原先增加一个电机要在 bsp_can.c 和</p>

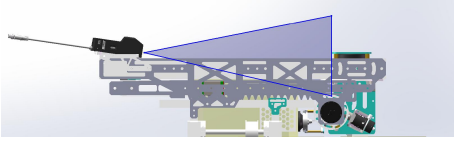
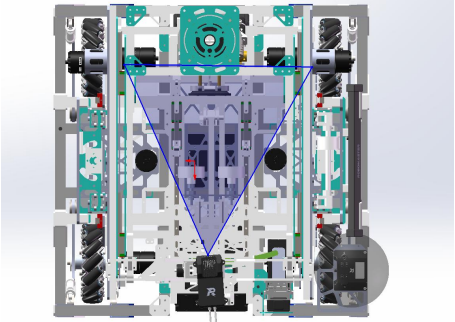


实现的功能	如何实现
	<p>comm_task.c 中进行繁杂的配置改为只用在 motor_task.c 中进行统一配置，不仅高效且明了。高效的代码框架使得改动作的速度大大增加。</p>
<p>布线的优化</p>	<p>原先的开发板位于底盘，云台所有线需要飞跃整个抬升，使得布线混乱且弹簧线的数量很多。优化后布线仓移至一级抬升平台，每一级之间使用坦克带连接，减少了弹簧线数量，使得布线更加简洁美观。</p>
<p>自定义控制器基本控制机器人</p>	<p>用 AS5600 收集控制器每个轴的编码值，再将数值统一打包，通过串口传给机器人。再通过解包把编码值传给每个轴，实现用自定义控制器基本控制机器人。</p>
<p>兑换机械臂可以夹取</p>	<p>修改一代机械臂结构，取消五级兑换难度为代价，保证可以夹取和传递矿石并保证矿石二维码朝向的正确</p>

表 8 赛场未实现的功能

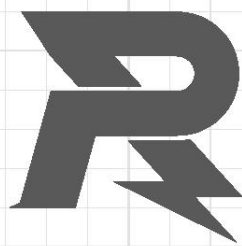
未实现的功能	原因	图片
<p>实现矿石在连机械臂的传递和存储</p>	<p>夹取舵的取矿臂伸出会有跳尺的现象，使得代码写好的电机角度不能保证夹取的吸盘到达理想位置，使矿石不能传递和储存。兑换舵的 pitch 轴运动多次后，发现回到 normal 模式的时候 pitch 轴不能保持水平，使得传递有问题。</p>	
<p>夹取地面矿石</p>	<p>云台设计不合理导致视野受限，无法观察到地面矿石的姿态，且取矿臂（夹取大资源岛矿石）pitch 轴长度吃紧难以很好地夹取地面矿石。</p>	

未实现的功能	原因	图片
<p>稳定夹取大资源的矿石</p>	<p>可以稳定快速地将大资源岛中的矿石从槽中拔起,但是因为参数微调得不够好,导致矿石会与涵道口的金属件产生碰撞,无法拔出。</p>	
<p>通过自定义控制器提高兑矿效率</p>	<p>自定义控制器连接裁判系统的时候,自定义控制器控制机器人会有延迟以及掉包的情况,使得兑换不能流畅进行。</p>	<p>-</p>
<p>底盘的结构问题</p>	<p>比赛中断断了条腿;底盘连接抬升四根铝方管固定(处)松垮;角连接件使用铆钉固定,铆钉强度不足。</p>	
<p>滑块铝方管连接松垮</p>	<p>抬升前倾,高度损失,电控调试困难,兑矿夹矿困难; 通过改变铝方管固定方式后有一定改善,但仍然存在该问题;</p>	
<p>抬升机构问题</p>	<p>左右两边抬升的紧度不一样,紧的那边的铜螺母磨损大,同步带易跳齿;原因:机械结构设计不合理,未完全考虑到装配问题和零件加工以及维护问题; 导致装配误差大,两侧受力不均无法正常使用;</p>	

未实现的功能	原因	图片
<p>抽屉滑轨内轨部分不可控；遮挡图传</p>	<p>为了追求最大化伸出利用；伸出方式采用了倍程伸出的设计机构，忽略了玻纤板刚度不足，及抽屉滑轨内滑轨本身设计不可控导致，整车运动时内层滑轨会不可控伸出。</p> <p>图传上，本赛季原本计划参考广东工业大学图传安装方案使得因此可以上下移动以获得更广阔的视野；但后期考虑到工程上机构众多度没有多余的空间，因此该方案被取消；</p>	 <p>图 55 图传遮挡侧视图</p>  <p>图 56 图传遮挡俯视图</p>

### 1.9.3 经验总结

1. 赛季初造车规划有待完善：本赛季因金矿获取方式改变（由空接改为甬道放置）为了避免大型机械臂的复杂及制造的高成本因此选择了双机械臂方案，但经实际测试发现双机械臂方案容易造成空间拥挤，以及机械臂间矿石传递困难的，整体表现极不稳定；引进自定义控制器的电控方案和机械臂设计上有待改进。
2. 云台设计和安装位置不合理，造成工程视野较差影响兑矿取矿效率不高，下赛季在整体规划上需要优先考虑图传的位置。
3. 机械造车时间太长，导致电控调车缺乏时间，应该适当压缩机械造车时间，留下一定时间给电控调试。
4. 前期动作设计不适合小资源岛的矿石获取，最后在参考其他学校的动作设计后临时更改动作，使得操作手没有充足的时间练习。因此在下赛季中，要仔细阅读规则，并造出准确的场地道具，以便及时发现动作的不合理。
5. 这台车最难的地方是整体方案的成型。这是我们第一次尝试龙门架结构结合双机械臂，技术积累以及经验积累较少，很多地方相当于是从零开始，虽然解决了很多机械结构的问题，但是基本功能的实现仍然很难保证；本赛季所设计的龙门架的结构不适合夹取甬道放置的矿石，因此，下赛季我们会改变机械结构，采用机械臂取矿的方式来适应规则的变化，改用机械臂方案也更能发挥自定义控制器的优势。



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街道仙茶路与兴科路交叉口大疆天空之城T2 22F