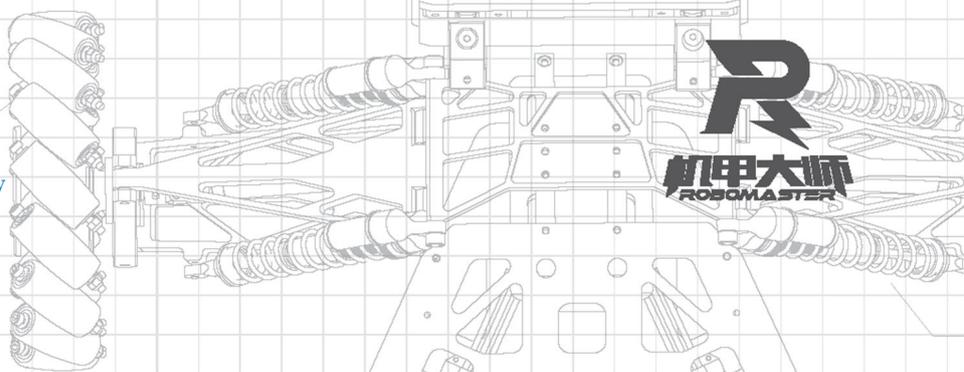




山东理工大学

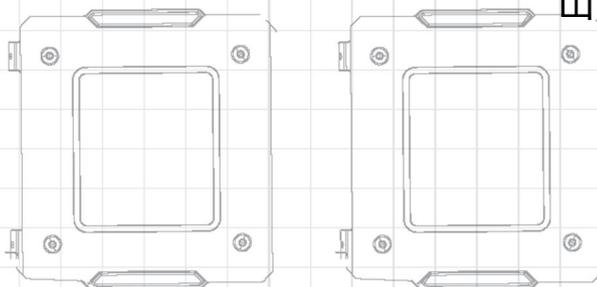
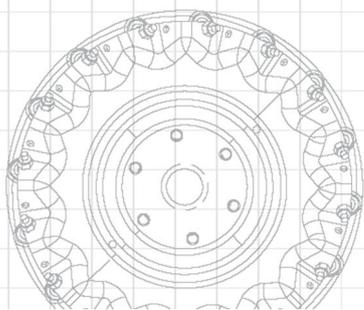
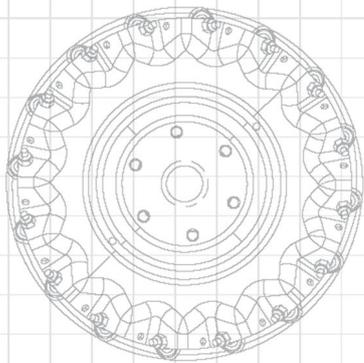
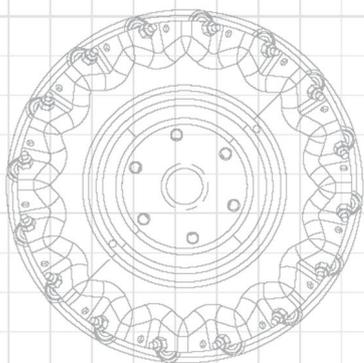
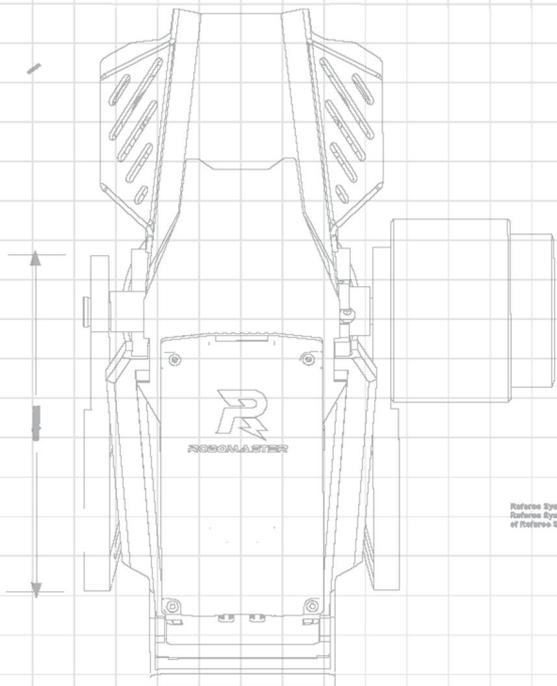
shandong university of technology



Exclusively designed for the RoboMaster M550S P18 Brushless DC Motor and C2000 Brushless DC Motor Speed Controller, this M550S Assembly Kit includes several cables and a terminal board.

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

Kit M550S Assembly Kit includes several cables and a terminal board, which are compatible products with other for the independent motor.



ROBOMASTER

机甲大师超级对抗赛

技术方案

山东理工大学 齐奇 编制

2024年06月 发布



前言

本成本报告由山东理工大学 齐奇编制，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	张昊阳	
硬件	薛福祥	
软件	薛福祥	李徽之
算法	踪金坤	夏齐隆
其他	于紫焯	于丽敏

目录

前言	2
1. 概述	4
1.1. 背景&目标	4
1.2. 其它学校机器人分析综述	5
1.2.1. 矿石获取机构	5
1.2.2. 矿石兑换机构	7
1.2.3. 自定义控制器	9
1.3. 机器人功能定义	11
1.3.1. 工程机器人功能	11
1.3.2. 功能量化举例	11
1.4. 机器人核心参数	12
1.4.1. 工程机器人	12
1.4.2. 自定义控制器	13
2. 设计方案	14
2.1. 机械结构设计	14
2.1.1. 底盘	14
2.1.2. 机械臂	19
2.1.3. 银矿获取机构	24
2.1.4. 自定义控制器	27
2.2. 硬件设计	30
2.2.1. 整机硬件方案框图	30
2.2.2. 自定义控制器控制板	31
2.2.3. 关键器件	33
2.3. 软件设计	35
2.3.1. 机器人软件系统架构	35
2.3.2. 重点功能	40
2.4. 导航算法设计	59
2.4.1. 功能简介	59
2.4.2. 重要算法原理阐述、公式推导	59
2.4.3. 算法库介绍	61
2.5. 其它	63
3. 研发迭代过程	64
3.1. 测试记录	64
3.2. 版本迭代过程记录	64
3.3. 重点问题解决记录	65
4. 团队成员贡献	66
5. 参考文献	67

1. 概述

1.1. 背景&目标

相较于 2023 赛季，RoboMaster2024 超级对抗赛的规则对于兑换机制以及经济体系进行了较大改动。首先单次兑换所能获得的金币价值将随这期间经历的时间下降，这就使得工程机器人在兑换矿石时需要更加的精准、迅速，而且大资源岛由之前的不规则矿石姿态改为“探囊取物”，这其实更加强调了矿石获取机构的精准以及稳定；再者，今年的规则中引入了半自动的概念，给半自动机器人发送指令需要消耗金币，而且买活、快速呼叫空中支援等也需要消耗大量的金币，所以全队经济总量，直接影响步兵、哨兵、英雄、无人机等兵种的作战能力。

在我们的理解中，工程机器人在本赛季赛场上的作用主要包括两点：

- (1) 快速获取矿石并兑换尽可能多的金币
- (2) 支援作用（拖拽救援，卡位参团）

在这两点中，我们认为最重要的是仍然是精准快速的成功兑换高难度姿态。针对这一目标，我们队伍以及其他大部分队伍在上赛季所采用的传统“龙门架”结构的工程机器人已经不太容易满足精准、快速兑换高难度姿态的目标。不过部分高校开始应用的一次性获取多块银矿石的结构仍然采用“龙门架”结构，“龙门架”的稳定性和可靠性仍然值得肯定。

机械臂拥有灵活，高自由度，收纳尺寸小、伸展尺寸大等优势，能够满足精准、快速兑换高难度姿态的目标。

根据 2023、2024 两个赛季的规则变动，我们认为工程机器人正在逐渐朝着机械臂方向发展，“龙门架”结构可以凭借其可靠性对机械臂进行辅助，于是我们将机械臂作为新一代工程机器人的研发重点并设计简洁的“龙门架”结构来应对复杂的战场环境。

本赛季工程机器人的研发目标如下：

- (1) 设计符合尺寸、重量限制的工程机器人，机构包括：底盘、机械臂、银矿获取机构。
- (2) 能够抓取大资源岛中央位置的金矿石，能够抓取掉落在地上的矿石。
- (3) 能够同时抓取多块小资源岛内的银矿石。
- (4) 能够通过自定义控制器控制机械臂精准运动。
- (5) 能够在工程机器人被击杀后保证机械臂状态稳定，不会造成机构损伤。

1.2. 其它学校机器人分析综述

2023 赛季规则改动之后，对工程机器人取矿和交换机构的自由度提出了较高要求，大部分队伍将往年主流的多个平移轴配合具有 1-2 个自由度的夹爪或吸盘升级为具有 Yaw、Roll、Pitch 多个自由度的更加灵活的取矿/交换结构。2024 赛季对于各兑换等级的兑换站姿态变动不大，所以上赛季各个队伍比较成熟的交换结构方案仍然适用。

本技术方案中，参考了其他学校较为成熟的取矿机构开源方案，也学习了部分诸如底盘、避震等结构方案。以下分析所参考的资料主要来源于论坛开源方案、官方以及各个队伍 B 站视频、线上线下队伍间的交流。

1.2.1. 矿石获取机构

1.2.1.1. 夹爪

优点：夹爪的结构相对简单，易于开发。经过过往的几个赛季的技术沉淀，大多数学校也有了一套比较成熟的夹爪结构。缺点：夹爪结构通常体积比较大，占用较大空间，空间对抗性弱于吸盘，并且也不能很好的适应不规则的大资源岛矿石，随着 2023 赛季兑换机制的改动，新的兑换站由于尺寸较小，不利于夹爪发挥，导致在 2023 赛季的赛场上使用夹爪的学校不多。

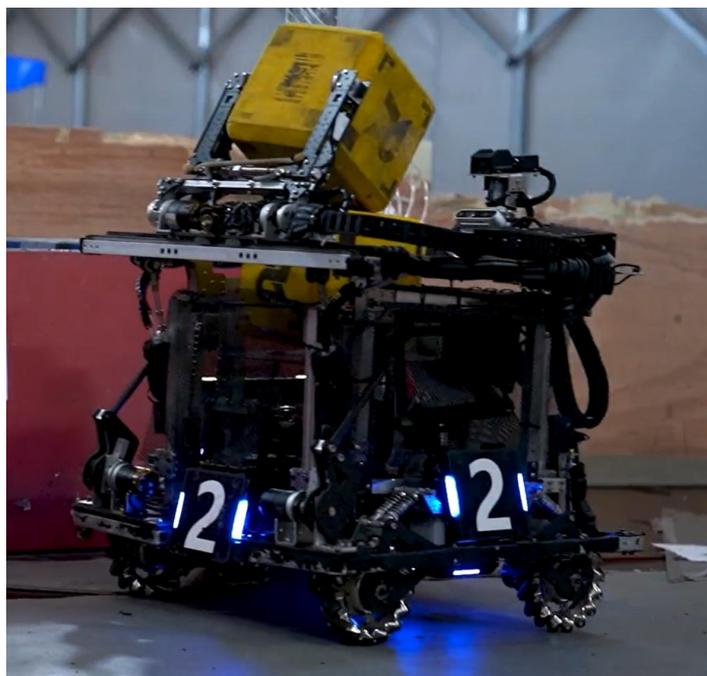


图 1 气动夹爪工程机器人

1.2.1.2. 吸盘

随着东北大学开始使用吸盘作为空接方案，越来越多的学校开始尝试，并且愈加成熟。优点：吸盘在空接过程中的稳定性远高于传统夹爪，2023 赛季甚至出现了哈尔滨工业大学（深圳）与深圳大学空接同时吸住了同一块矿石相互争抢的情况。2024 赛季取消了空接，吸盘仍然可以凭借其体积小而适用于新的规则。缺点：需要搭配真空泵以及冗长的气路使用，气泵振动大，影响车身稳定，过长的气路会导致缠绕的问题，以及可能出现的气管堵塞、破损，影响其稳定性。

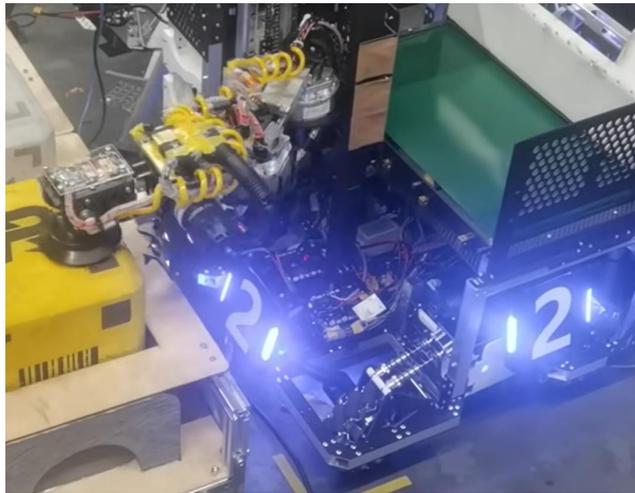


图 2 吸盘工程机器人

1.2.1.3. 滚吸夹

优点：在获取 2023 赛季大资源岛内不规则矿石时有明显优势。缺点：需要迭代很多个版本，且结构较为复杂，占用体积较大，在新规则下的大资源岛不适用。

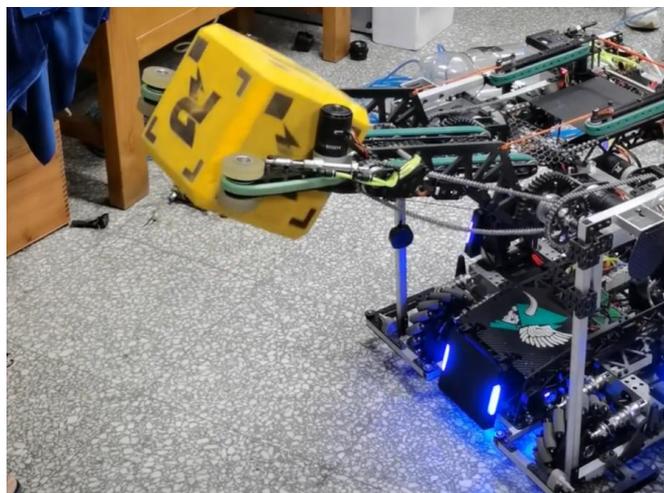


图 3 滚吸夹工程机器人

1.2.2. 矿石兑换机构

1.2.2.1. 大型机械臂

优点：活动范围大，自由度高，比传统“龙门架”结构更加灵活，能较好的应对 2023 赛季的赛场。缺点：占用的空间大，限制了存矿机构的设计。成本也比较高（关节电机价格昂贵，结构强度要求高）。

大型机械臂最早由 2021 赛季广东工业大学开始使用，2023 赛季的赛场上，上海交通大学（如图 4），中国科学技术大学等学校采用了大型机械臂，使用效果不错。如果机械臂前端体积不大，长度足够，那么不失为获取 2024 赛季的大资源岛内矿石的一种解决方案。

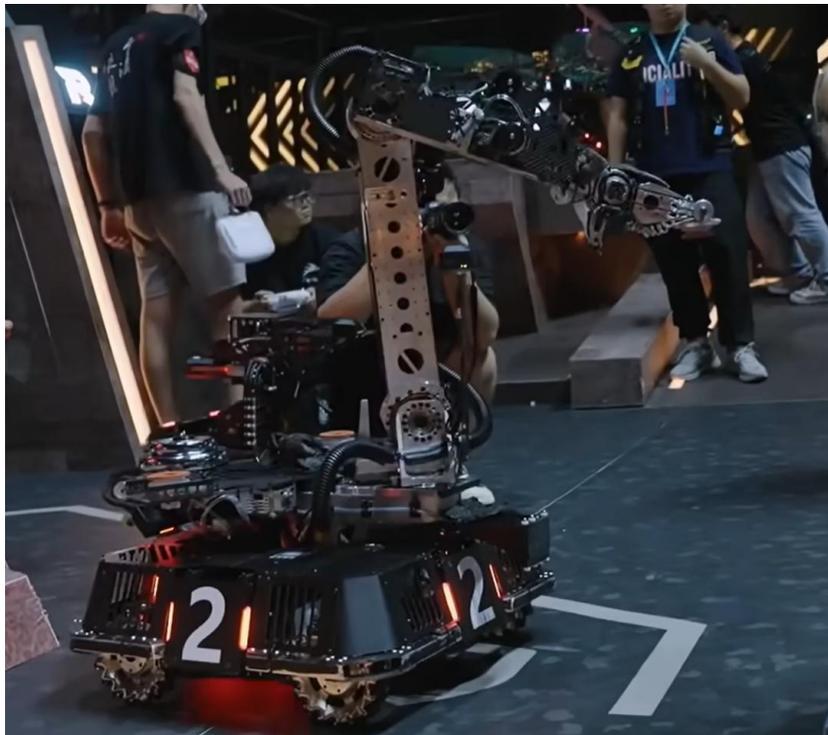


图 4 上海交通大学工程机器人

1.2.2.2. 末端小型机械臂

2023 赛季的赛场上，获取矿石和兑换矿石对于工程机器人兑换机构的自由度数量要求变高，有相当一部分学校选择在保留抬升结构的基础上加装小型机械臂，例如沈阳航空航天大学、深圳大学、南方科技大学等学校。其中沈阳航空航天大学（如图 5）和深圳大学在机械臂的末端采用了差速器结构，我们认为这是一个非常巧妙的设计，有效降低了底部电机的负载且末端体积；南方科技大学则创新性的使用了 SCARA 结构的机械臂（如图 6），SCARA 构型可以有效减小收纳尺寸，且机械臂不需要负责 Z 轴方向的位移，也便于操作手的操作以及运动学解算。是非常优秀的设计。



图 5 沈阳航空航天大学差速器结构



图 6 南方科技大学 SCARA 构型机械臂

1.2.3. 自定义控制器

随着 2023 赛季自定义控制器的引入，到全国赛的赛场上，许多高校都给出了自己的成熟的解决方案。

1.2.3.1. 视觉识别

南方科技大学采用了基于视觉识别的自定义控制器方案，优点是体积小，操控精准，可以应用运动学解算。缺点是需要摄像头和工控机，成本较高。



图 7 南方科技大学自定义控制器

1.2.3.2. 体感控制器

上海交通大学，南京航空航天大学，中国石油大学（华东）等学校采用了体感操作的方式，可通过陀螺仪得到精准的角度姿态，这样的自定义控制器更加小巧，持握手感好。缺点是只能依靠运动学解算，对于部分学校来说有一定的技术难度。



图 8 上海交通大学自定义控制器

1.2.3.3. 示教控制器

西安交通大学的“环太平洋”方案，以及东北大学的并联机器人方案都是采用了示教器的原理，通过电机或者编码器反馈出角度映射到工程机器人对应的机构上。这样的自定义控制器对于操作手而言操作上更加简单易懂，技术相对好实现。缺点是体积较大，过于笨重。考虑到我队的技术积累情况，我们决定采用示教控制器来作为我们自定义控制器方案。

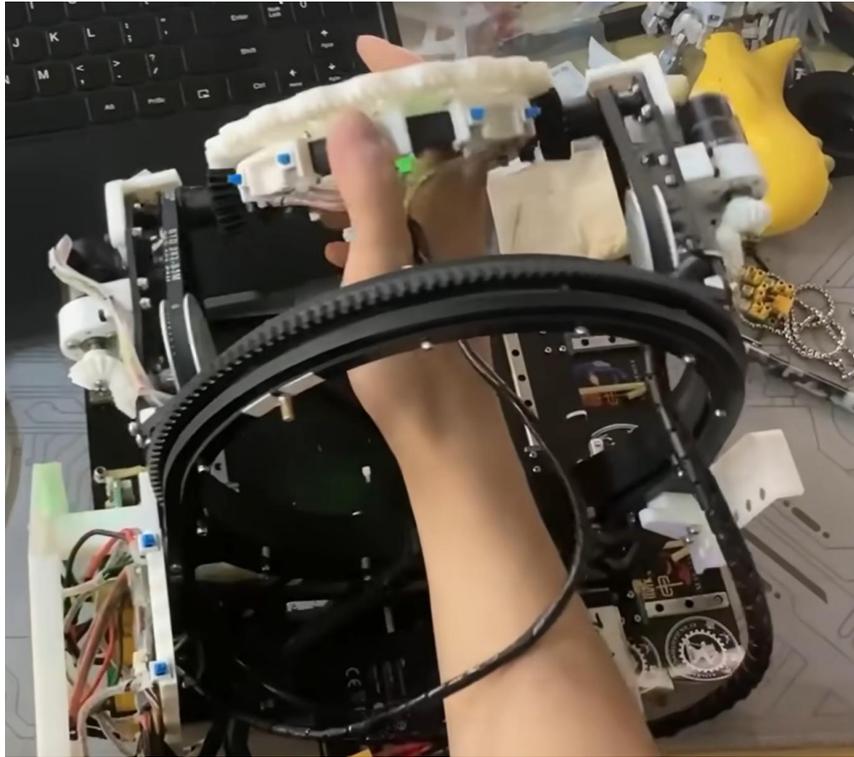


图 9 西安交通大学自定义控制器

1.3. 机器人功能定义

1.3.1. 工程机器人功能

1.3.1.1. 底盘功能设计

(1) 稳定性强

- ① 轮距达到 520mm
- ② 采用劲度系数高的避震弹簧，保证机器人变形时底盘依旧保持稳定

(2) 四驱独立悬挂

1.3.1.2. 兑换机构

我们采用了大型机械臂作为兑换机构，使用四个关节电机作为 Yaw, Pitch1, Pitch2, Roll 电机，满足 X Y Z 轴的空间运动。机械臂两个 Pitch 之间另有一个 3508 带动丝杆，使机械臂可以伸缩。末端使用两个 2006 电机带动差速器结构，满足 Roll 和 Pitch 两个自由度。

1.3.1.3. 银矿获取机构

使用一个 3508 电机带动同步带，二倍放大行程的抬升机构，安装有两个吸盘，可以一次性获取小资源岛的两块银矿石，可以大大缩减获取矿石的时间，在对局前期快速保障经济。

1.3.1.4. 图传机构

随银矿获取机构抬升，并且拥有独立的 Yaw 轴，可以为操作手提供良好的视野。

1.3.2. 功能量化举例

(1) 兑换能力

- ① 可以使用自定义控制器进行矿石获取以及兑换
- ② 15 秒之内稳定兑换 4 级矿石

(2) 获取矿石：半分钟之内获取三块银矿石

(3) 救援能力：能够稳定拖拽全向轮、麦轮底盘

1.4. 机器人核心参数

1.4.1. 工程机器人

名称	参数
重量、重心	32KG X=15.977mm Y=170.775mm Z=11.667mm
尺寸（长宽高）	596mm*598mm*596mm
主要传感器型号、参数、数量	微动开关
电路功耗、所有电容总容量、工作时电压范围等	2.3W 无电容 24V
执行器件（电机、气缸等）用途与数量说明等	官方电机： RM3508 电机：6 个 4 个-底盘轮组电机 1 个-机械臂伸缩丝杆电机 1 个-银矿获取机构抬升·电机 RM2006 电机：2 个 2 个-末端取矿机构差速器驱动电机
	非官方电机： 宇树科技 A1 关节电机：4 个 1 个- 机械臂 Yaw 轴电机 1 个-机械臂 Pitch1 轴电机 1 个-机械臂 Pitch2 轴电机 1 个-机械臂 Roll 轴电机 MG996R 舵机 1 个-图传 Yaw 轴电机
	气动器件： 铠盟 KMDP-C4 有刷真空泵 1 个-为机械臂吸盘提供吸力 2 个-为银矿获取机构提供吸力
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	最大移动速度 2.5m/s 云台 Yaw 轴角度-45° -225°

1.4.2. 自定义控制器

名称	参数
重量、重心	3.01KG X=-50.009mm Y=109.558mm Z=265.989mm
尺寸（长宽高）	356mm*223mm*349mm
主要传感器型号、参数、数量	无
电路功耗、所有电容总容量、工作时电压范围等	1.2W 无电容 24V
执行器件（电机、气缸等）用途与数量说明等	官方电机： RM2006 电机：2 个 1 个-传递 Roll 轴角度 1 个-传递差速器 Pitch 轴角度
	非官方电机： 本末科技 M06C1 电机：3 个 1 个-传递底部 Yaw 轴角度 1 个-传递 Pitch1 轴角度 1 个-传递 Pitch2 轴角度

2. 设计方案

根据队伍上赛季工程机器人的表现，以及本赛的实际情况，对于 2024 赛季的工程机器人我们决定继续采用麦克纳姆轮底盘，以大型六自由度机械臂作为金矿获取机构以及兑换机构，并单独设计银矿获取机构。

2.1. 机械结构设计

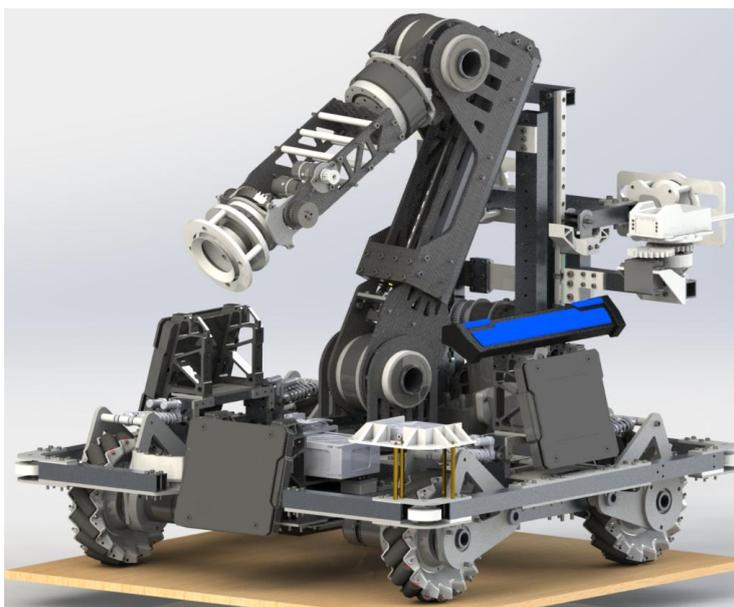


图 10 2024 赛季工程机器人

2.1.1. 底盘

由于本技术方案决定采用大型机械臂，当机械臂进行伸展时重心变化会非常大，所以底盘设计时应尽量低以降低重心。

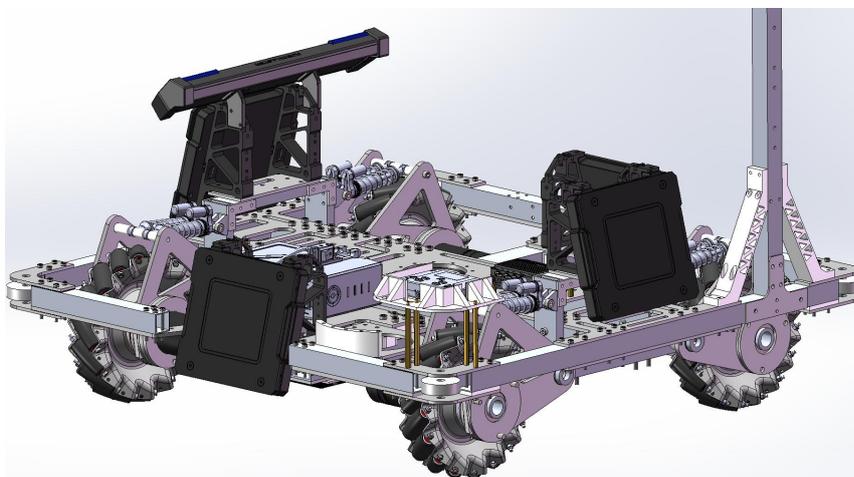


图 11 底盘

2.1.1.1. 轮组与悬挂

(1) 轮组：轮组采用的是我队比较成熟的方案（如图 12），该方案设计优点是轮组两侧都有法兰轴承和板材，不会出现轮组“外八”的情况，缺点是轮组重量较高，与车架固定点更加靠近内侧，不利于减小尺寸。轮组比我队步兵麦轮轮组重 320 克，但是这对于工程机器人 35Kg 的限重来讲是可接受的，也满足设计目标中降低重心的要求，综合考量之后我们决定仍然采用这款轮组。加工工艺方面则选择了较为传统的玻纤板材+成品轴承+成品联轴器方案。该联轴器（图 13）主体为 6061 铝合金，由于 6061 强度不高，D 型轴区域可能会产生较大的破坏，该联轴器在于 D 轴接触的平面位置外加了一块钢片(如图 14)来提高工作时的稳定性。

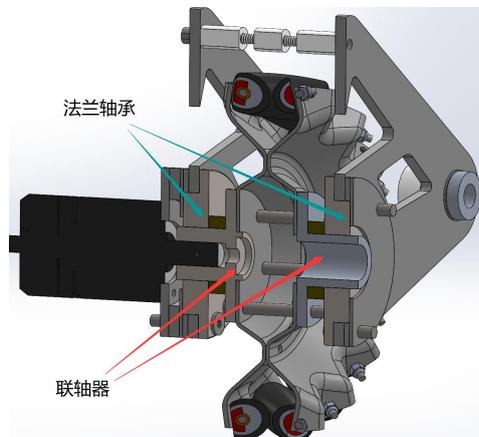


图 12 轮系剖面图



图 13 联轴器



图 14 钢片实物图

(2) 悬挂：2023 赛季工程机器人最求稳妥，悬挂传动角接近 90° 。但在与其他学校交流过程中我们认识到，不同于步兵，英雄等兵种，需要行进间射击，工程机器人的功能机构并不需要在行进间作业，所以悬挂的硬度只需要满足兑换矿石以及获取矿石这种不动或低速移动时的底盘稳定性即可，故新赛季的悬挂设计更多的在于满足放低底盘的要求，以及为裁判系统让位，故采取了 35.57° 传动角（图 15），将底盘安装机械臂的位置降低到了 127.85mm。

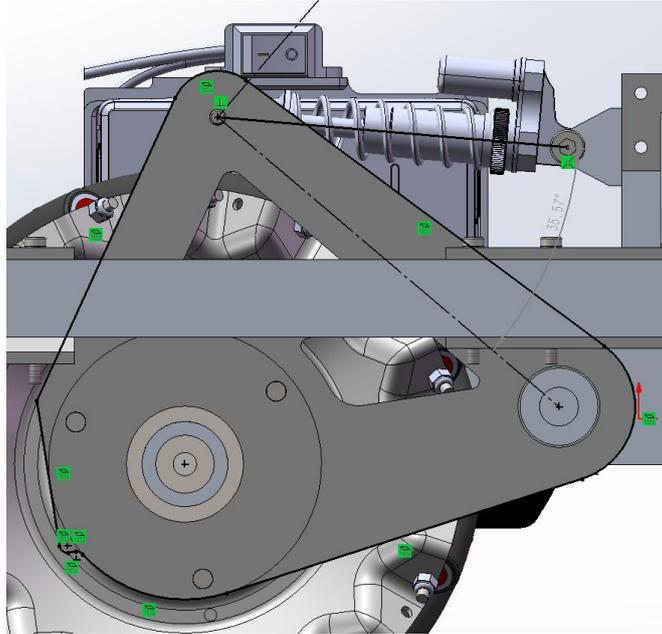


图 15 传动角示意图

2.1.1.2. 底盘框架

(1) 框架设计：在我们的构想中底盘应满足降低重心，稳固的安装机械臂两点要求。在设计过程中，受到南京理工大学 Alliance 战队培训视频启发，我们决定在传统铝管堆叠的基础，在铝管交叉处添加一层十字型板材，采取类似“三明治”的结构（如图 16），这样可以很大程度上缓解铝管过软导致受力时螺丝孔被扩大进而导致铝管间固定产生位移的情况，同时板材的刚性还能帮助铝管承担螺母于螺丝头之间的压紧力。加工工艺方面，不同于其他学校的焊接，我们采用的是更符合我队实际的定制板材+定制铝管+螺栓的方式来制造我们的底盘。

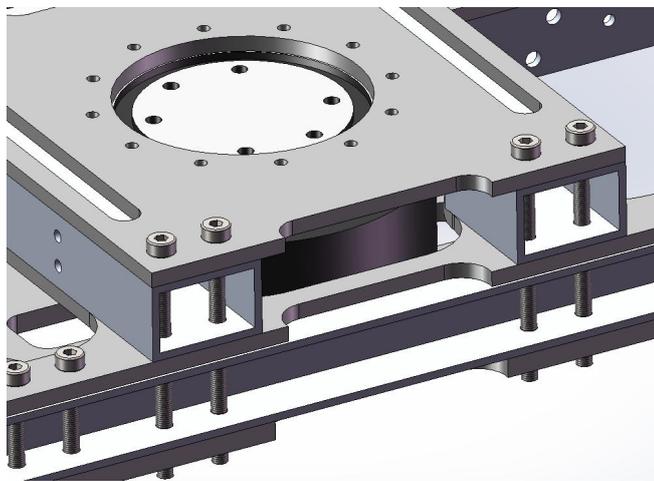


图 16 底盘三明治结构示意图

(2) 走线：考虑到机械臂回转会占用大量上层空间，不利于走线。故将电源管理模块以及 C 板、A 板、自研中心板等均布置在底盘提前划定的区域（如图 17）；在实际走线中，将中心板安置在车架主横梁侧面，直接面向四个轮电机；为了减少电线缠绕的风险，在机械臂上布置了分线板，使机械臂到底盘中心板的电线只有 1 根。这样使整车走线更加集中，设计保护壳时可以有的放矢的保护重点区域。

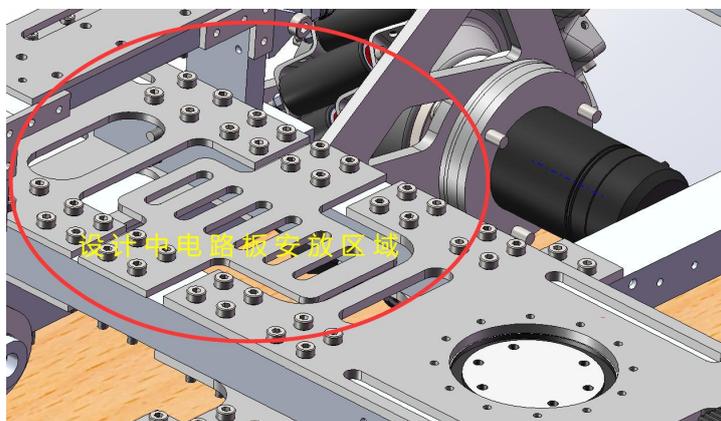


图 17 设计中电路板安放位置

在实际安装过程中，我们使用扎带将电路板固定在该区域，使其避开机械臂回转半径。

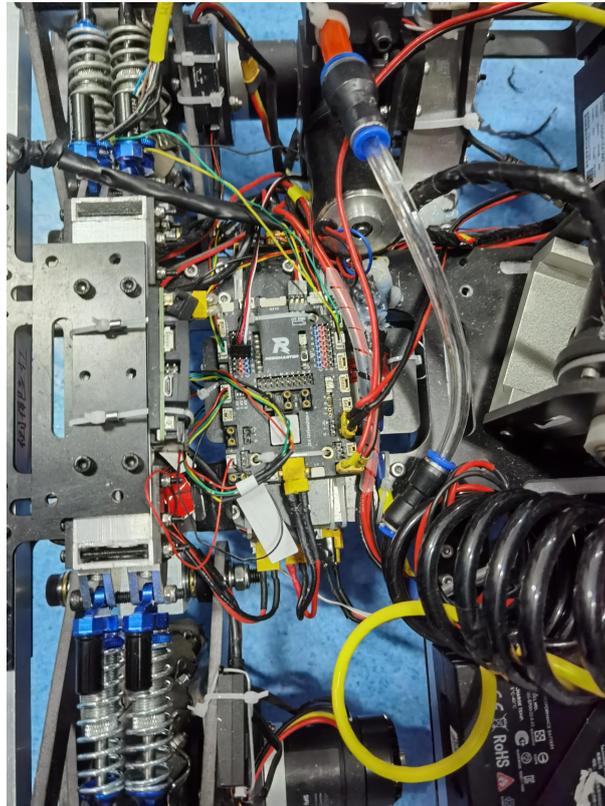


图 18 实际走线展示

2.1.2. 机械臂

由于我们的规划是尽可能低底盘降低重心，故为了能够满足兑换站的最高高度，我们决定采用两段大臂，为了满足折叠之后的检录尺寸，于是又在第一节机械臂中设计有丝杆、滑轨伸缩。

2.1.2.1. Joint1

在机械臂最底端是负责 Yaw 轴转动的宇树科技 A1 关节电机。因为电机固定螺丝型号是 M3.5，无法找到 12.9 级的高强度螺丝，故决定将电机上下安装面的全部螺丝安装位置都用上。为了保证机械臂上层稳定，最终采用板材夹住铝管的“三明治结构”（如图 19）。

图中的机械臂上端底座要承受整个机械臂 Yaw 方向的运动，并且要支撑整个机械臂的重量，故采用 6061 铝合金铣件来增加结构强度。

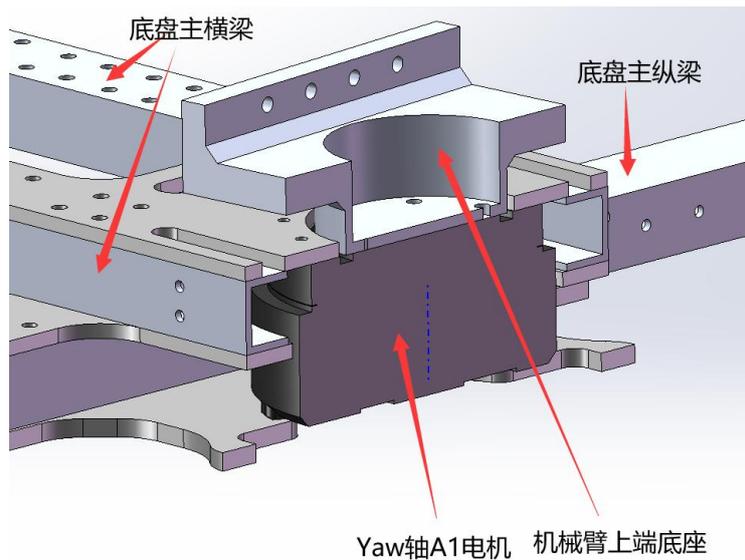


图 19 机械臂 Joint1

2.1.2.2. Joint2&Link1

Joint2 负责机械臂的 Pitch，由宇树科技 A1 电机驱动。由于 Joint2 要负责承担整个机械臂的重量，故决定将电机内置于机械臂内侧，让上下两个固定端面的螺丝都可以发挥作用，增加结构稳定性，输出轴另外一侧则定制了 6061 铝合金的铣件，并安装有轴承，来承担从动端转动。图中 Joint2 夹板要与机械臂上端底座一起承担机械臂的重量，故采用了 6mm 厚度的碳纤维板材。

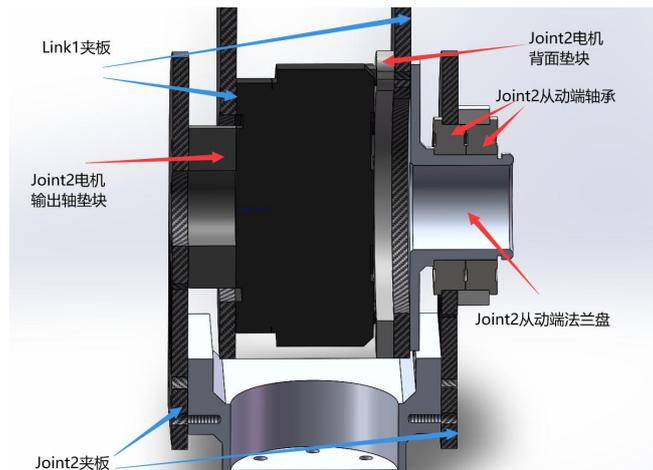


图 20 Joint2 剖面图

Link1 是机械臂最长的一段，使用了两块 5mm 厚的碳纤维板材作为主体，由于计划采用的两段机械臂无法满足最高等级的兑换站高度，故在 Link1 上加装了由 3508 驱动丝杆来带动的伸缩结构，由于伸长后 Link1 收到的径向载荷较大，于是伸缩滑轨则采用了径向载荷承载性能更好的抽屉滑轨。

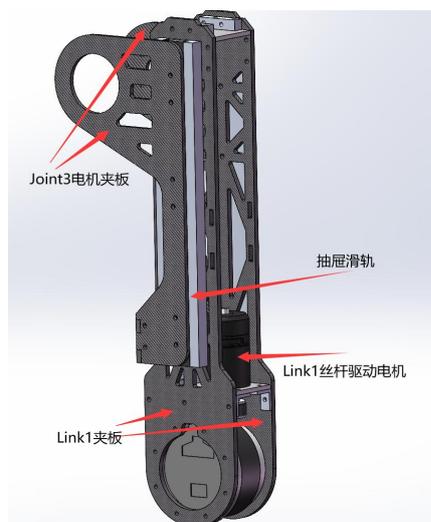


图 21 Link1 示意图

由于 Link1 存在伸缩机构，且机械臂运动过程中要求其尽可能灵活，则走线方面不能使用拖链等传统走线方式，我们采用了将线路与弹簧气管相互缠绕的方式走线（如图 23）在 Link1 伸缩时，随着气管展开，线路也会随着伸长，这样避免了线路拉扯的风险。

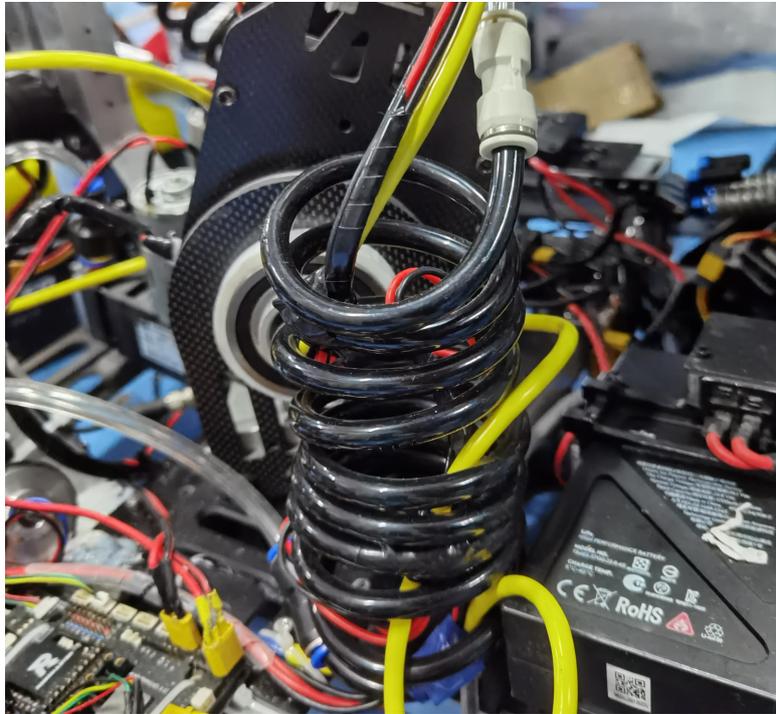


图 22 弹簧气管走线

2.1.2.3. Joint3&Joint4

Joint3 与 Joint2 结构类似，都是负责机械臂的 Pitch 轴，故结构也类似（如图 24），同样也是使用了两个固定端面来增加结构强度。

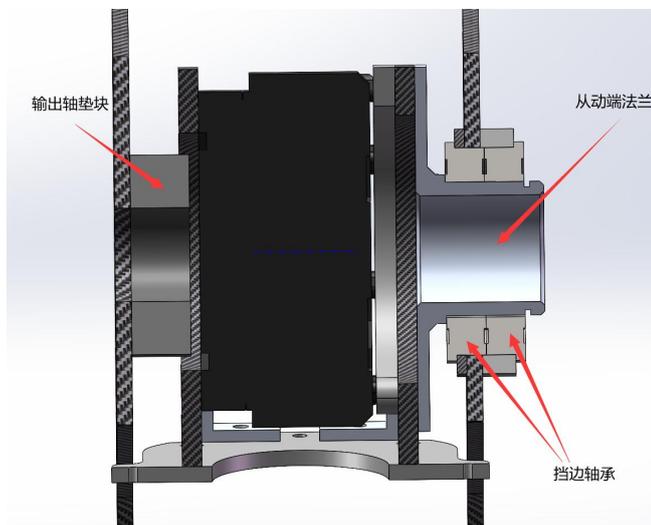


图 23 Joint3 剖面图

Joint4 是负责连接机械臂最前端差速器结构的，该位置载荷较小，同样使用宇树科技 A1 电机的情况下，只固定了背面的四颗螺丝，输出轴位置使用垫块与前端差速器相连。（如图 25）

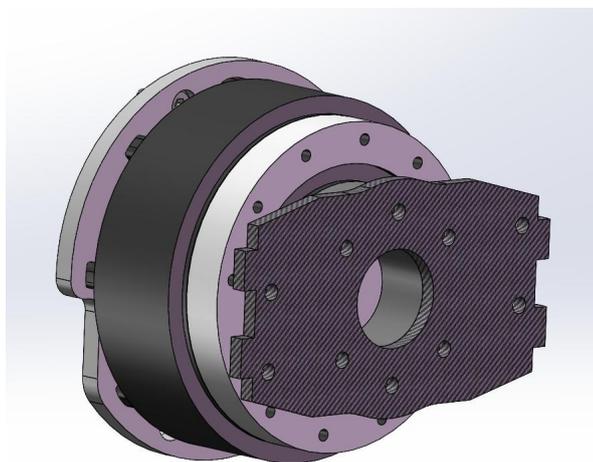


图 24 Joint4

2.1.2.4. Link2&Joint5

在研究了论坛中的开源后，发现深圳大学，沈阳航空航天大学等学校都在矿石获取机构中使用了差速器结构，我们队伍在去年也存在 Pitch 轴电机距离载荷过远，导致电机过热的问题，而差速器可以很好的缩短 Pitch 轴作用点到载荷的距离，降低力矩，并且差速器还可以设计减速机构，减少电机过热的情况发生，故我们队伍决定在机械臂的末端采用差速器结构（如图 26）

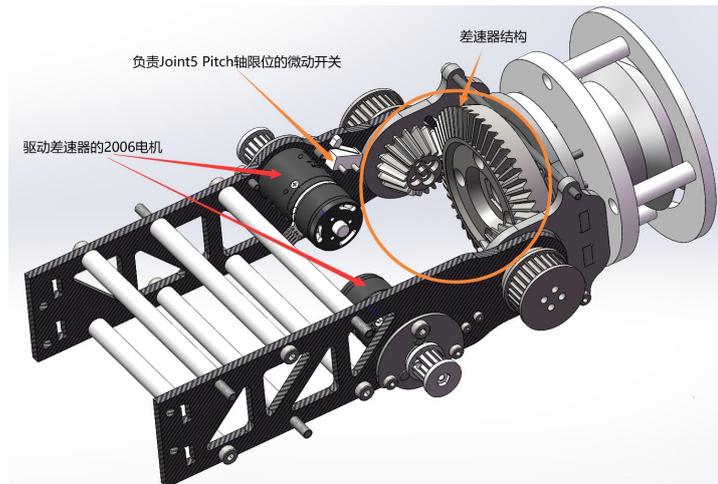


图 25 Link1&Joint5

差速器结构使用了两个 2006 电机作为动力，先经过同步带轮 1:2 减速，又通过齿轮 9:26 减速。30 齿的带轮与齿轮都使用了 3D 打印件，经过实际测试，3D 打印齿轮用在这个位置，强度是足够的，于是就没有在更换机加工零件。

由于 2006 无法得知自己的绝对角度，又在差速器 Pitch 轴的极限位置处安装了微动开关，在上电时可以自动复位，标定角度。

2.1.3. 银矿获取机构

在全国赛赛场上，东北大学的工程机器人可以一次性获取多块银矿石，我们认为该功能是有必要的，所以在 2024 赛季的工程机器人上设计了一套银矿获取机构。

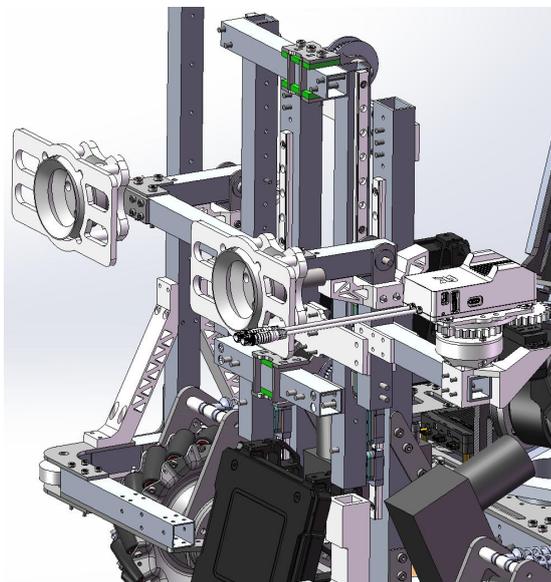


图 26 银矿获取机构

2.1.3.1. 抬升机构

由于空间布置问题，决定采用一次性获取两块矿石的设计。

抬升机构我们使用了同步带二级倍程抬升（如图 28）。由一个 3508 电机在底部驱动同步带轮，3508 电机的安装位置同样设计在抬升结构上，这样就可以实现 2 倍行程。

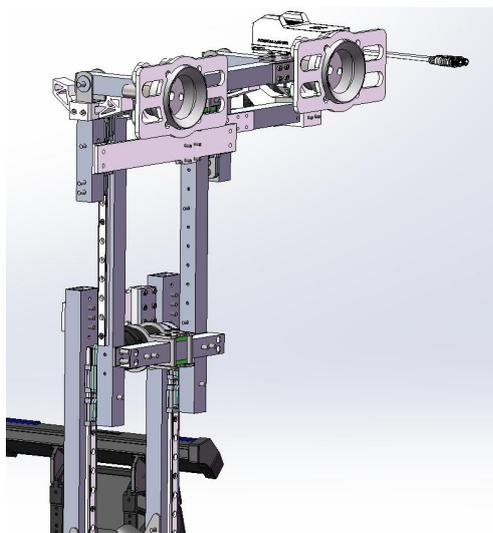


图 27 二倍行程机构

加工工艺上采用了定制铝管+螺栓+定制板材+3D 打印件+定制滑轨滑块的形式。

2.1.3.2. 吸盘优化

我们队伍的吸盘在上赛季存在空接不稳的情况（如图 29），即吸力不足。因为新设计的银矿获取机构是通过吸住银矿石侧面来进行获取，考虑到小资源岛凹槽大小以及矿石摩擦力，我们认为吸力不足的问题必须解决。



图 28 上赛季吸盘

我们认为是风琴吸盘无法压缩完全，导致有一段风琴状的行程会发生弹性形变，这样会导致在颠簸或者遇到阻力时，使吸盘和矿石接触面出现缝隙，是原本应该处于负压的吸盘腔与大气联通，导致矿石掉落；随后我们尝试更换了更软的风琴吸盘（如图 30），试图使其能在工作过程中被压缩到底，减少多余晃动。改吸盘对于空接确实有所提升，但是在侧取银矿石时效果不如之前更硬的版本。



图 29 软吸盘

基于减少多余晃动的思路，我们决定在吸盘可压缩行程中间设计一个支撑架（如图 31），增大对矿石的硬支撑面积，此时，吸盘继续提供拉力，吸盘架则提供支持力，使接触面压力增大，可以增大矿石与吸盘支架的摩擦力，使矿石被吸得更牢固，同时在矿石遭遇外力影响时，吸盘支架提供额外支持力（如示意图），使吸盘与矿石之间不产生缝隙，从而降低掉落的风险。

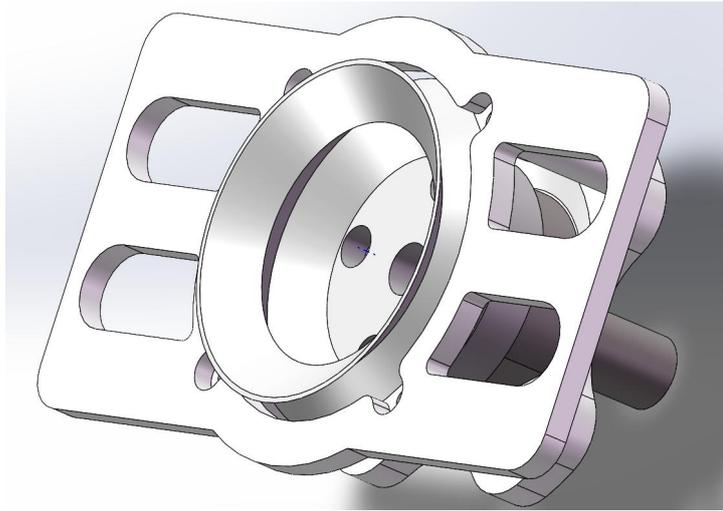


图 30 吸盘支撑架

2.1.4. 自定义控制器

根据我队人员配置和技术积累，我们决定采用较为简单的示教控制器方案。

根据队内情况，我们决定使用本末科技 M06 电机映射机械臂 Yaw, Pitch1, Pitch2 电机，在最前方使用两个 2006 映射 Roll 轴关节电机以及差速器 Pitch 轴。差速器 Roll 轴可以无限转动，故采取键盘按键映射。

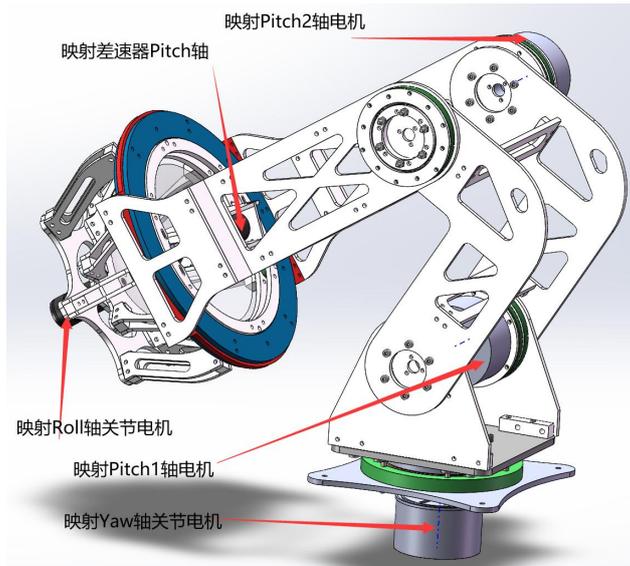


图 31 自定义控制器

2.1.4.1. 轴系设计

(1) 映射 Yaw 轴。自定义控制器的 Yaw 轴采用了类似步兵云台 Yaw 轴的结构（如图 33）有一块玻纤切割的挠度盘来保证 Yaw 轴稳定，设计了轴承座，安装有深沟球轴承，增加其载荷承载能力。

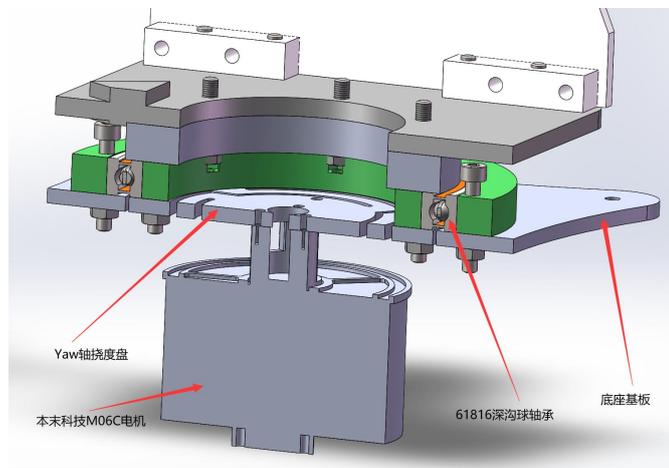


图 32 自定义控制器 Yaw 轴剖面

(2) 映射 Pitch 轴。自定义控制器 Pitch 轴选用的也是本末科技 M06C 电机，该电机只有背面和输出轴有固定孔位，如果将其前后固定孔位都用上，那么自定义控制器留给手部的运动空间会略显局促，于是我们决定只使用电机输出轴固定孔，在通过三块内径不同的板材组成轴承座，安装有 61809 深沟球轴承，来为电机分担径向载荷（如图 34），这样电机输出轴的螺丝只受拉和受扭转，电机自重更多的交给了轴承。

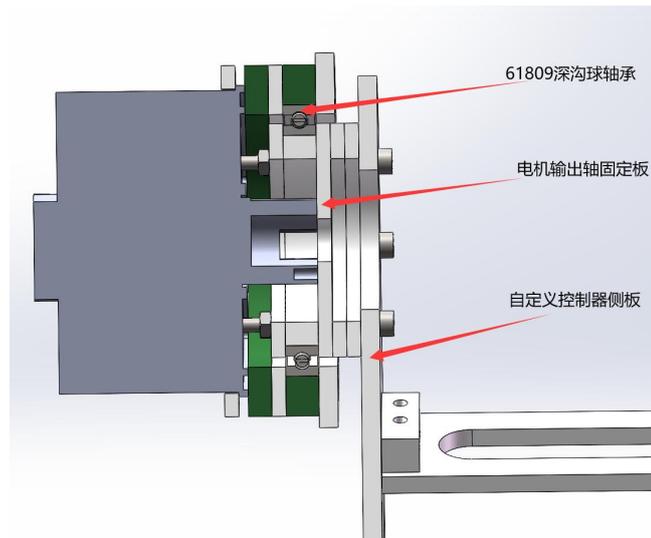


图 33 自定义控制器 Pitch 轴

(3) 映射 Roll 轴&差速器 Pitch 轴。用于映射差速器 Pitch 轴的是 2006 电机，该位置需要由人手持握，所以尺寸较大，又因为该结构前端连接着映射 Roll 轴的 2006 电机，需要旋转，如果在该位置使用轴承，会导致重量过大，手部负担加大，我们使用了三层板材堆叠出一个环形槽，在里面均匀放置 6 个聚氨酯辊子，在转动过程中，聚氨酯辊子发挥类似轴承滚珠的作用，从而使该结构流畅旋转（如图 35）。

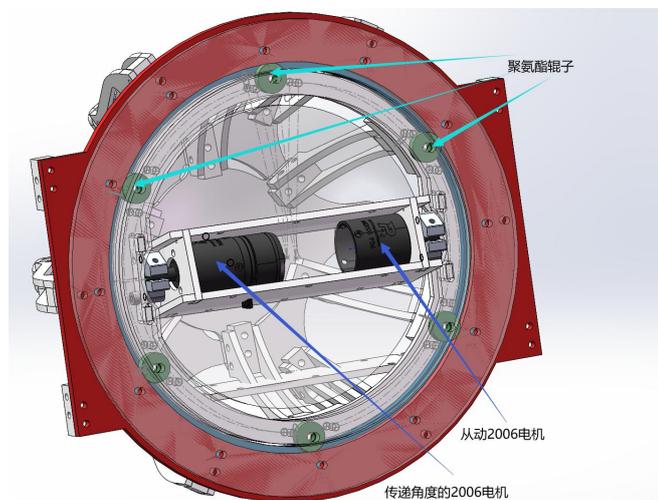


图 34 自定义控制器差速器旋转结构

Roll 轴部分设计有两套不同外径的盘爪结构，一套固定在机架上作为固定端，一套固定在握持位置，作为转动端（如图 36）

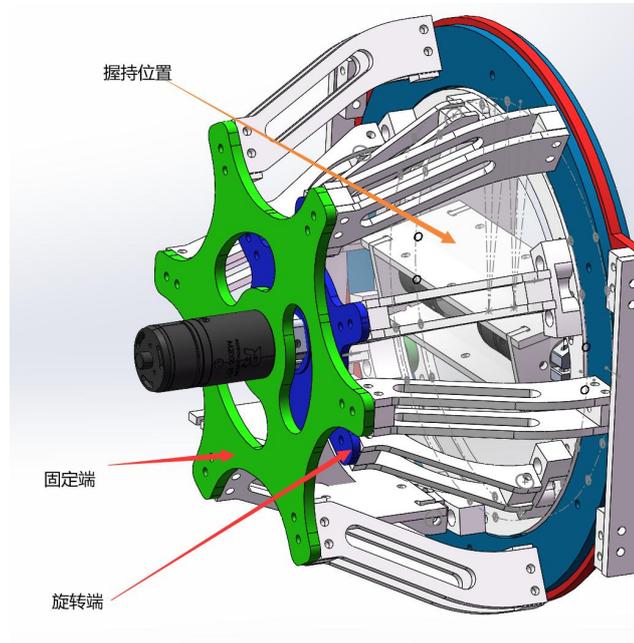


图 35 自定义控制器旋转盘爪

2.2. 硬件设计

2.2.1. 整机硬件方案框图

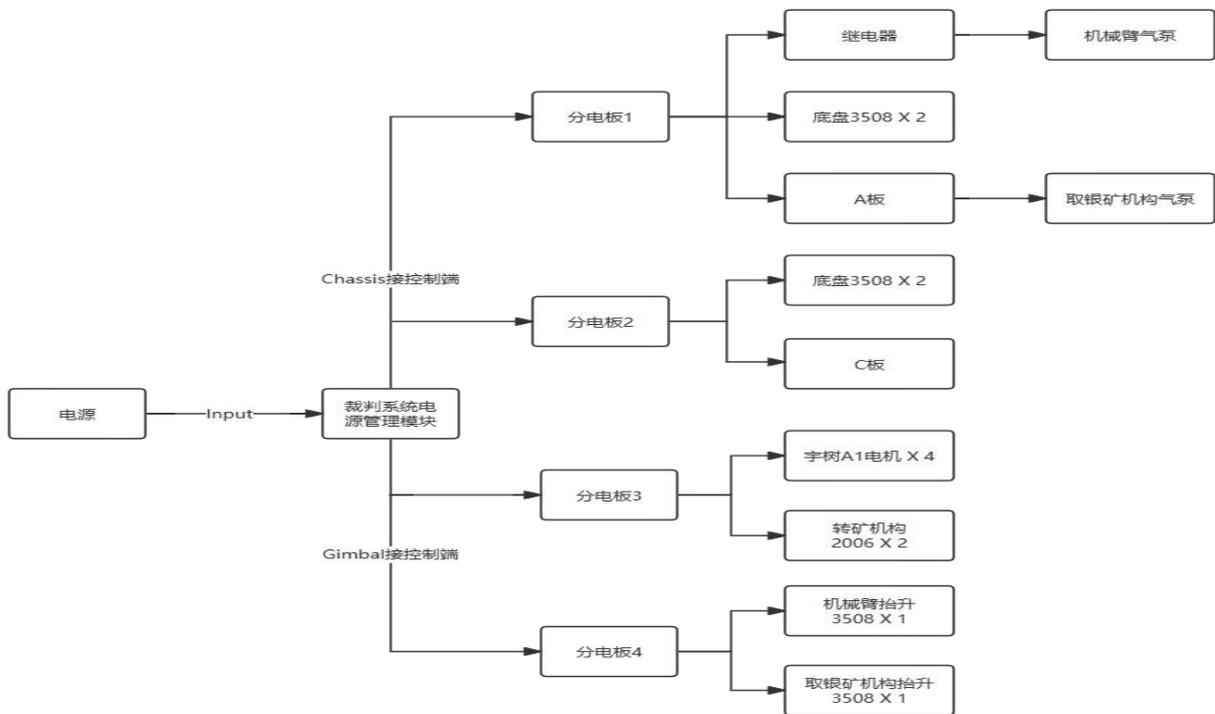


图 36 供电拓扑图

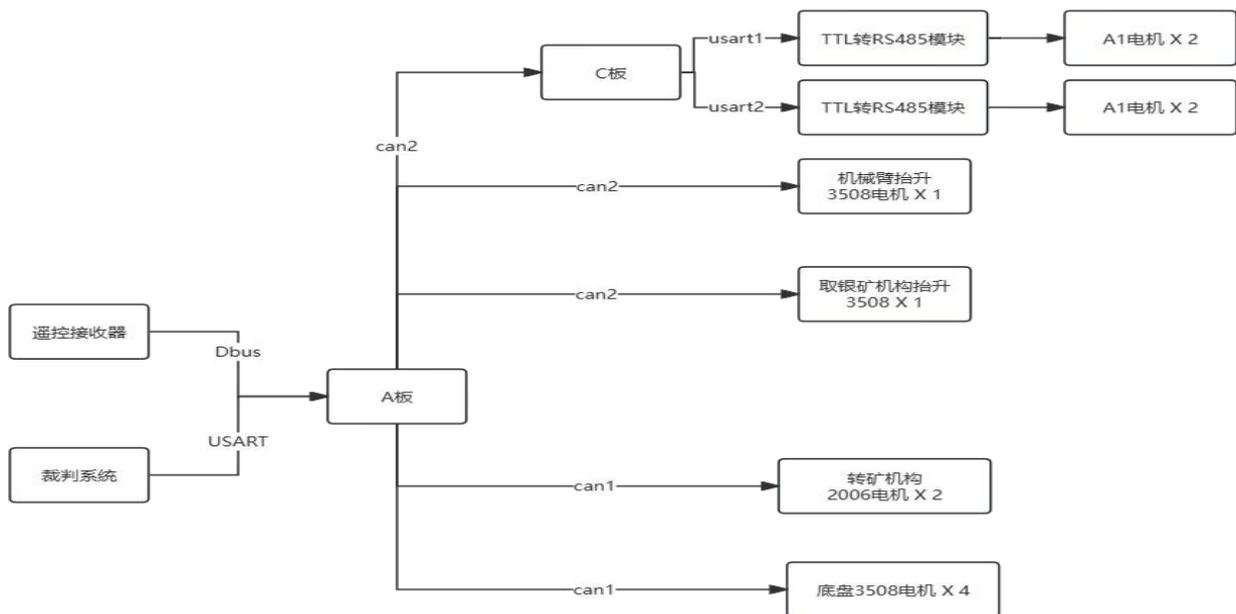


图 37 通讯拓扑图

2.2.2. 自定义控制器控制板

由于新赛季抓取和兑换矿石的难度增加，工程机器人在本赛季改用机械臂进行兑矿。如果想要实现机械臂的灵活操作，想要靠传统键鼠和现有 DR16 遥控器不太方便，因此，我们设计了一款自定义控制器来实现灵活控制机械臂的功能。为了满足自定义控制器的性能我们为此设计了一块主板。

以下是此项目的原理图和 PCB 照片：

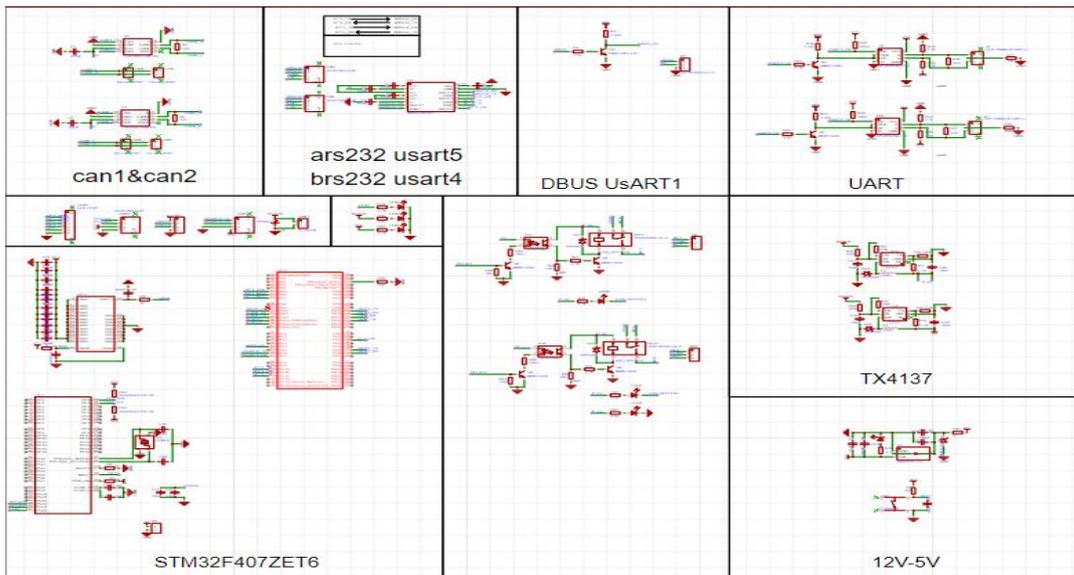


图 38 自定义控制器控制板原理

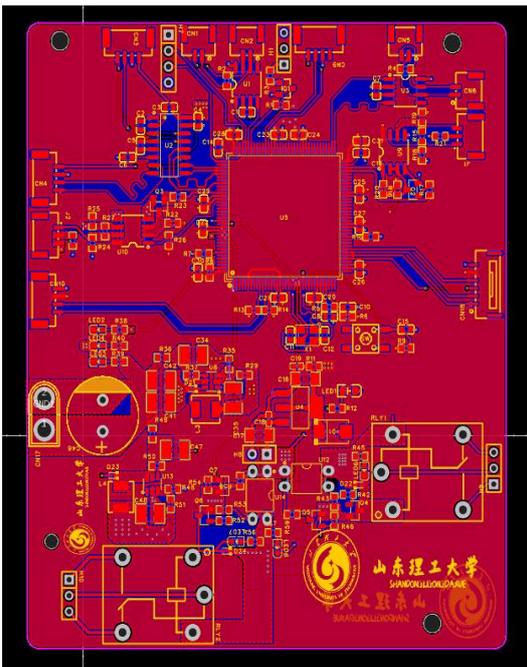


图 39 PCB 正面图

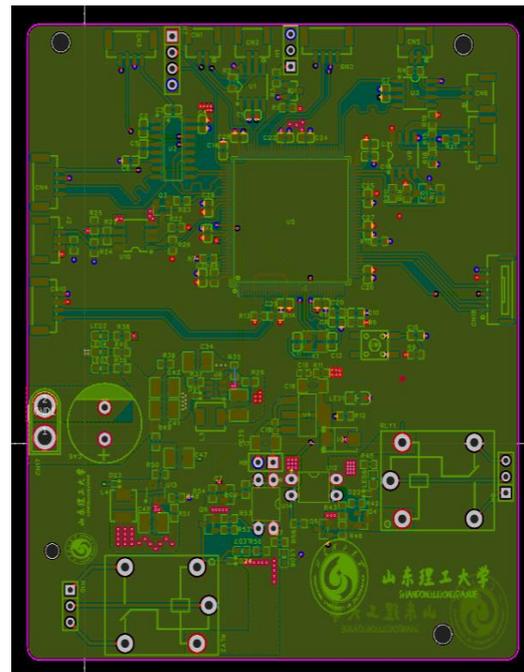


图 40 PCB 内层 1 图

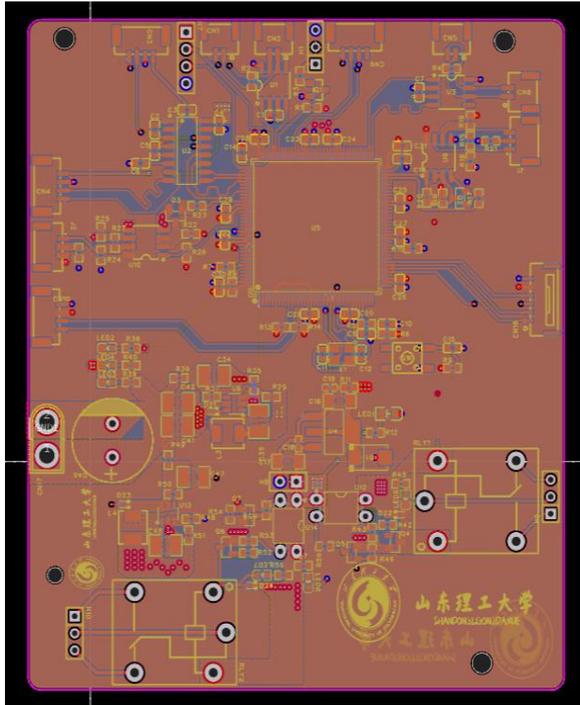


图 41 PCB 内层 2 图

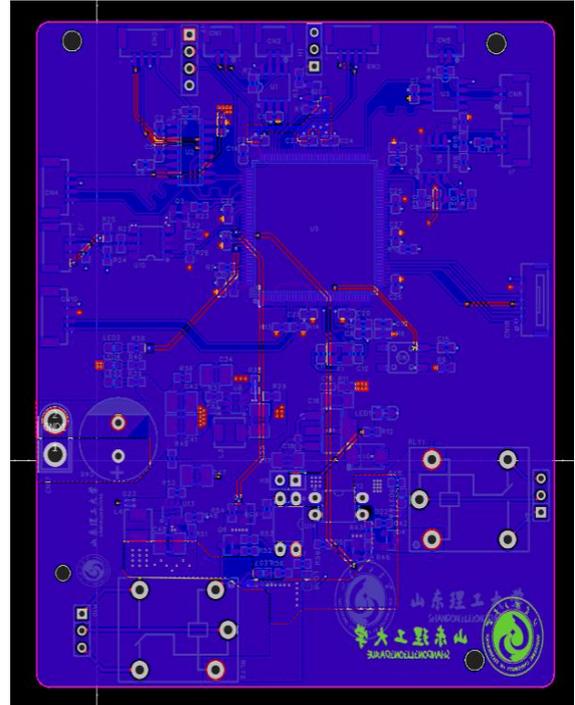


图 42 PCB 底层图

2.2.3. 关键器件

自定义控制器控制板：尺寸：83.1mm*99.78mm，主控板又详细分为主控部分、降压稳压部分、通信部分。

2.2.3.1. 主控部分：

考虑到对于 STM32 系列单片机比较熟悉，主控芯片选择 STM32F407ZET6 作为主芯片板其性能足够进行对三个 M6 电机的控制以及通过 RS232 对电脑数据的收发。

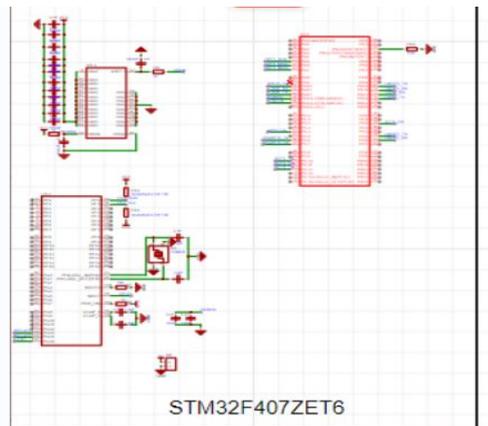


图 43 主控部分

2.2.3.2. 降压稳压部分：

下半部分采用 AMS1117-3.3 作为降压芯片将 12v 电源降为 5v 电源给主控部分供电,同时上半部分使用了 TX4137 这一款电流模式的降压调节器，使其电压趋于稳定。

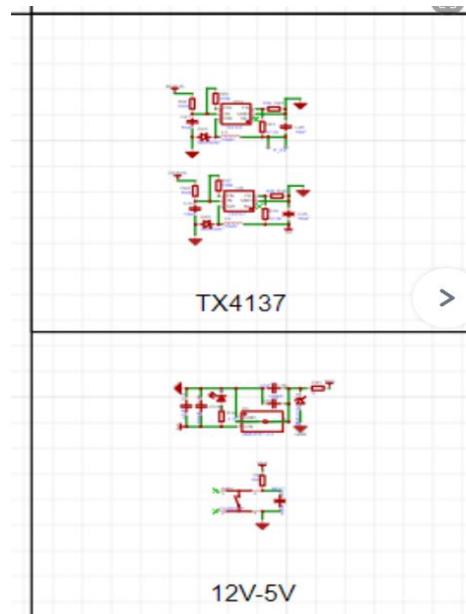


图 44 降压稳压

2.2.3.3. 通信部分:

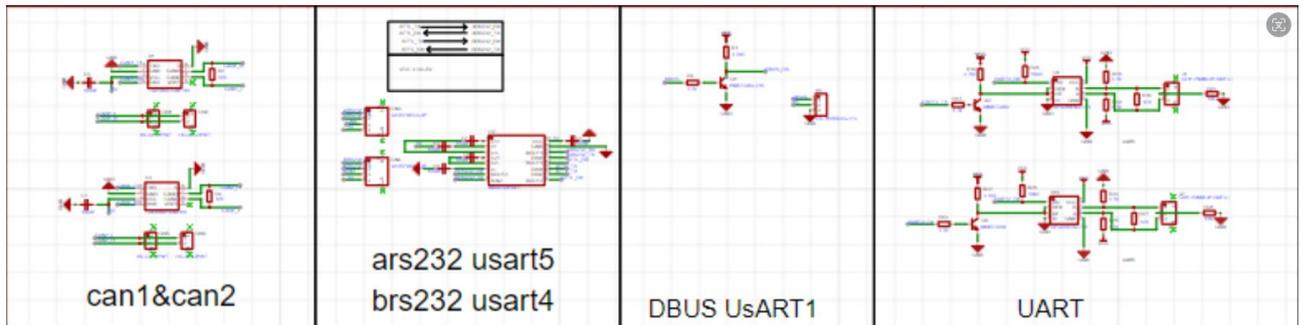


图 45 通信模块

为了满足自定义控制器的通信要求，我们在主板上设计了 RS232 的通信串口便利了自定义控制器的使用。同时配备了两个 UART 口和两个 CAN 口进行对自定义控制器的电机进行控制。

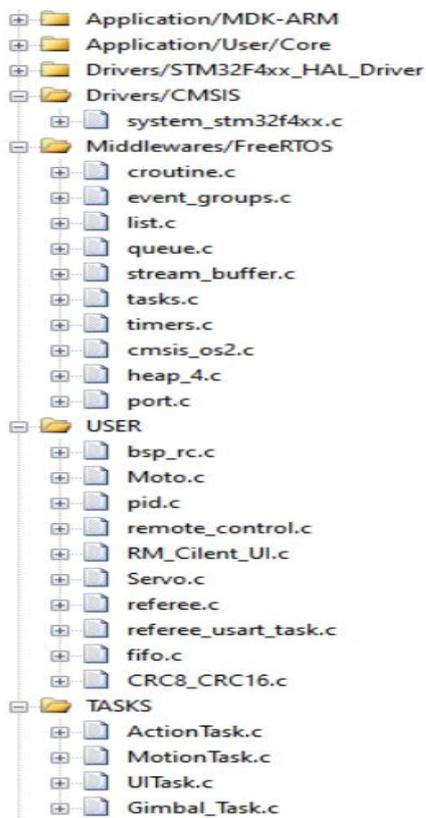
2.3. 软件设计

2.3.1. 机器人软件系统架构

由于工程机器人所需要控制的电机，气泵等电子器件较多，最后使用了 A 板和 C 板两个控制板进行控制。其中 A 板作为主要控制板负责对对外界（遥控器，自定义控制器，键盘）数据的接受，C 板指令的下达以及控制大部分的电机和全部的气泵。而由于机械臂结算以及机械臂的路径规划需要占用较大的内存，所以 C 板只主要控制机械臂的四个 A1 电机。

2.3.1.1. A 板系统框架

软件使用 C 语言作为主要编译语言，通过利用 STM32CubeMX 配置 STM32 的滴答定时器提供时基模拟操作系统实现了任务的简单调度，在用 keil 编写单个任务存在运行、挂起两种状态。建立一个 FreeRTOS 系统，从而对机器人进行控制。



Middlewares/FreeRTOS 主要是嵌入式的实时操作系统，作为一个轻量级的操作系统，功能包括：任务管理、时间管理、信号量、消息队列、内存管理、记录功能、软件定时器、协程等，可基本满足较小系统的需要。

USER 则主要是外界遥控器接收，电机基础配置以及一些需要的算法如 pid，CRC 校验等的基础配置。

TASKS 是主要行动任务，是完成行动的主要代码。主要包括模式行动控制（ActionTask），底盘移动控制（MotionTask），UI 控制（UITask），云台控制（GimbalTask）

主中断大致分为，更新模块离线状态，模式行动控制（ActionTask），底盘移动控制（MotionTask），UI 控制（UITask），云台控制（GimbalTask）。

模式行动控制主要控制思路为通过拨杆来控制工程机器人的运动状态开始数据进行更新，进入遥控器选择模式进行判断进入不同的模式，进入控制量解算得出每种电机，气泵，以及

向 C 板发送什么样的数据等，有关 can 的数据通过 can 发送和接收，有关自定义控制器的通过自定义控制器发送接受结构体变量进行发送接收。

如图为部分代码：

```
void ActionControlTask_Loop(void)
{
    action_update();    //数据更新
    action_mode_set();  //遥控器选择模式
    action_control_set();//控制里解算0
    action_can_tx();    //can发送
    customcontroller_data_pack_send(customcontroller_tx.given_current,30); // 自定义控制器发送接收结构体变量
    osDelay(1);
}
```

图 46 模式设置代码

(1) **底盘移动控制**主要控制是为了稳定底盘稳定的运动状态所使用的各种算法所在区域，如麦轮解算，斜坡函数初始化，电机限位，均值滤波的使用等等。

```
//斜坡函数初始化
ramp_init(&chassis_move.wz_set_ramp_source_type,0.001,5.5,-5.5);
Ave_Init(&gin_z.gin_z_buf,200);
}

static void chassis_update(void)
{
    uint8_t i = 0;
    for (i = 0; i < 4; i++)
    {
        chassis_move.chassis_speed[i] = CHASSIS_MOTOR_RPM_TO_VECTOR_SEN * Moto_Data1[i].speed;
    }
    chassis_move.vx = (-chassis_move.chassis_speed[0] + chassis_move.chassis_speed[1] + chassis_move.chassis_speed[2] - chassis_move.chassis_speed[3]) * MOTOR_SPEED_TO_CHASSIS_SPEI
    chassis_move.vy = (-chassis_move.chassis_speed[0] - chassis_move.chassis_speed[1] + chassis_move.chassis_speed[2] + chassis_move.chassis_speed[3]) * MOTOR_SPEED_TO_CHASSIS_SPEI
    chassis_move.wz = (-chassis_move.chassis_speed[0] - chassis_move.chassis_speed[1] - chassis_move.chassis_speed[2] - chassis_move.chassis_speed[3]) * MOTOR_SPEED_TO_CHASSIS_SPEI
}

static void chassis_mode_set(void)
{
    if (switch_is_up(rc_ctrl.rc.s[CHASSIS_RC_MODE_S2]))
        chassis_mode = CHASSIS_RC_SLOW;
    if (switch_is_mid(rc_ctrl.rc.s[CHASSIS_RC_MODE_S2]))
        chassis_mode = CHASSIS_RC;
    if (switch_is_down(rc_ctrl.rc.s[CHASSIS_RC_MODE_S2]))
        chassis_mode = CHASSIS_KEYBOARD;
}
```

图 47 底盘控制代码

(2) **UI 控制**为工程在操作手端所画的为了方便机器人控制的代码。在此展示部分代码 如图：

```
char tmp1[30]={0},tmp2[30]={0},tmp3[30]={0};
memset(&strSPEED,0,sizeof(strSPEED));
for(int k=0;k<6;k++)
{
    memset(&Aim[k],0,sizeof(Aim[k]));
}
//清空图形数据
Line_Draw(&Aim[0],"AL1",UI_Graph_ADD,5,UI_Color_Green,3,960,300,960,540);
Line_Draw(&Aim[1],"AL2",UI_Graph_ADD,5,UI_Color_Green,2,940,520,980,520);
Line_Draw(&Aim[2],"AL3",UI_Graph_ADD,5,UI_Color_Black,3,930,478,990,478);
Line_Draw(&Aim[3],"AL4",UI_Graph_ADD,5,UI_Color_Yellow,2,920,380,1000,380);
//可通过宽度
Line_Draw(&Aim[4],"AL5",UI_Graph_ADD,6,UI_Color_Green,2,440,0,627,443);
Line_Draw(&Aim[5],"AL6",UI_Graph_ADD,6,UI_Color_Green,2,1480,0,1293,443);
Line_Draw(&Aim[6],"AL7",UI_Graph_ADD,5,UI_Color_Orange,3,920,400,1000,400);
Char_Draw(&strSPEED,"CAP",UI_Graph_ADD,8,UI_Color_Green,20,strlen(tmp1),2,860,100,tmp1);
```

图 48 UI 代码

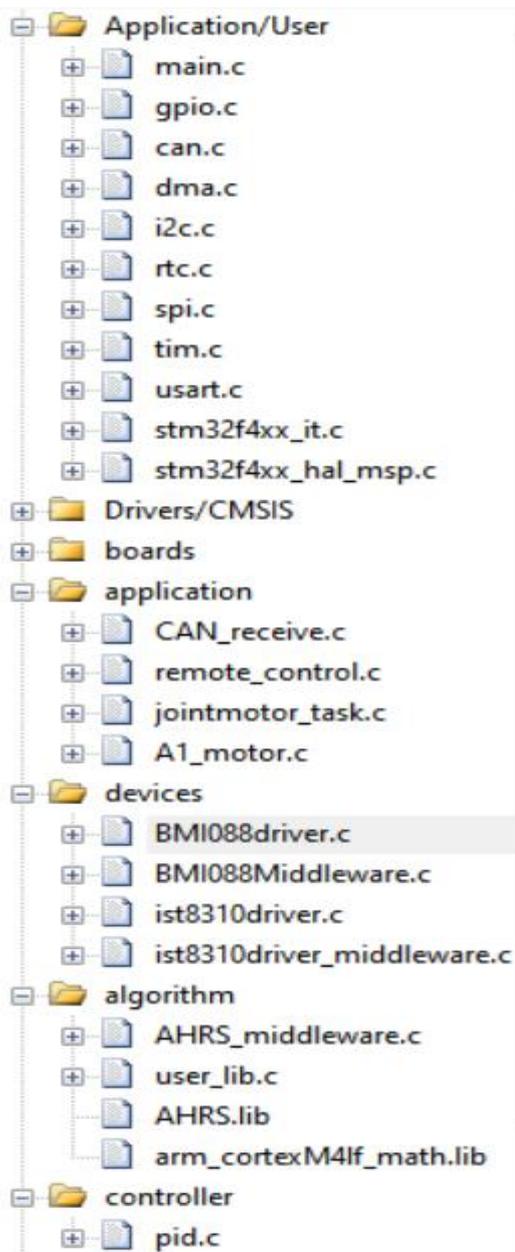
(3) 云台控制较为简单主要使用一个舵机控制图传的方向。

```

//判断舵机是否无力
//舵机标志位为0，舵机上电（初始模式舵机上电）
if(obstacle.obstacle_steer_flag == 0)
{
    obstacle_mode = OBSTACLE_MOVE;
    afdafawf = 1;
}
//舵机标志位为1，舵机无力模式
else if(obstacle.obstacle_steer_flag == 1)
{
    obstacle_mode = OBSTACLE_DOWN;
    afdafawf = 2;
}
    
```

图 49 云台控制代码

2.3.1.2. C 板结构框架



Application/Usar 主要为芯片的基础配置包括时钟配置，串口配置，引脚定义等。

Application 为主要任务，包括 CAN_receive.c(can 信号的接受与发送)remote_control.c(遥控器数据的接收处理)jointmotor_task.c(机械臂关节电机的运动学结算与路径规划)A1_motor.c(A1 电机的控制与使用)

Algorithm 主要为陀螺仪数据的使用

Controller 为使用的算法基础配置主要为 pid 算法

(1) can 信号的接受与发送主要为接收 A 板发送和接收的数据主要包括遥控器数据，自定义控制器数据等。

```

if(hcan->Instance == CAN1)
{
    HAL_CAN_GetRxMessage(hcan, CAN_RX_FIFO0, &RxHead, Rxdata);
    switch (RxHead.StdId)
    {
        case 0x205:
        case 0x206:
        {
            uint8_t i;
            i = RxHead.StdId - 0x205; //计算绝对偏移地址
            //get_total_angle(&moto_chassis[i]); //计算绝对角
            get_motor_measure(&motor_2006[i], Rxdata);
        }
        case 0x207:
        {
            rc_ctrl.rm.M6[0] = ((int16_t)(Rxdata[0] << 8 | Rxdata[1]));
            rc_ctrl.rm.M6[1] = ((int16_t)(Rxdata[2] << 8 | Rxdata[3]));
            rc_ctrl.rm.M6[2] = ((int16_t)(Rxdata[4] << 8 | Rxdata[5]));
            break;
        }
        default:break;
    }
}

else if(hcan->Instance == CAN2)
{
    HAL_CAN_GetRxMessage(hcan, CAN_RX_FIFO0, &RxHead, Rxdata);
    switch (RxHead.StdId)
    {
        case 0x209:
        {
            rc_ctrl.rm.M2006[0] = ((int16_t)(Rxdata[0] << 8 | Rxdata[1]));
            rc_ctrl.rm.M2006[1] = ((int16_t)(Rxdata[2] << 8 | Rxdata[3]));
            rc_ctrl.rc.s[0] = ((int16_t)(Rxdata[4] << 8 | Rxdata[5]));
            rc_ctrl.rc.s[1] = ((int16_t)(Rxdata[6] << 8 | Rxdata[7]));
            break;
        }
        case 0x208:
        {
            rc_ctrl.rc.ch[0] = ((int16_t)(Rxdata[0] << 8 | Rxdata[1]));
            rc_ctrl.rc.ch[1] = ((int16_t)(Rxdata[2] << 8 | Rxdata[3]));
            rc_ctrl.rc.ch[2] = ((int16_t)(Rxdata[4] << 8 | Rxdata[5]));
            rc_ctrl.rc.ch[3] = ((int16_t)(Rxdata[6] << 8 | Rxdata[7]));
            break;
        }
        default:break;
    }
}

```

图 50 通信代码

(2) 遥控器数据的接收处理为处理接收到遥控器数据和自定义控制器数据。主要流程为数据接收后先判断遥控器拨杆数据确定模式，如果为键盘控制则进入键盘控制代码，类似还有遥控器控制代码，自定义控制器代码。

```

static void chassis_control_set(void)
{
    uint8_t i = 0;

    if(chassis_mode == CHASSIS_RC)//遥控器控制
    {
        if (switch_is_up(rc_ctrl.rc.s[CHASSIS_RC_MODE_S1]))
            chassis_rc_controlup();
    }

    if(chassis_mode == CHASSIS_RC_slow)//自定义控制
    {
        if (switch_is_mid(rc_ctrl.rc.s[CHASSIS_RC_MODE_S1]))
            chassis_rc_controlup_slow();
    }

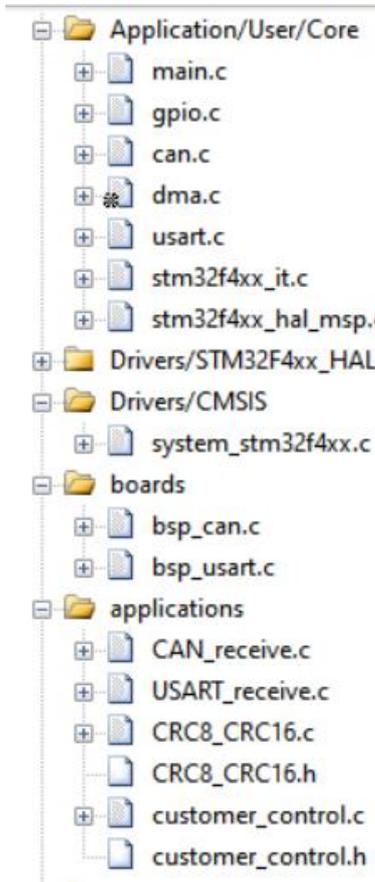
    else if(chassis_mode == CHASSIS_KEYBOARD)//键盘控制
        chassis_keyboard_control();
}

```

图 51 遥控器接收代码

2.3.2. 重点功能

2.3.2.1. 自定义控制器



Application/User/Core 依旧为芯片基础配置，由于自定义控制器代码较为简单所以以 main.c 为主代码区。

boards 为对 can 通讯和串口通讯的配置。

applications 为 can 通讯和串口通讯的使用和 crc 校验以及自定义控制器电机位置反馈代码

Customer_control 主要为接收自定义控制器每个电机数据计算出位姿，具体代码架构会在重点功能概述。

```

/*
 *M6电机角度计算
 */
void m6_angle_calc(void)
{
    HAL_UART_Receive_IT(&huart6, Rx[0], 10);
    HAL_UART_Receive_IT(&huart7, Rx[1], 10);
    HAL_UART_Receive_IT(&huart8, Rx[2], 10);
    Control_Motor(&huart6, 0, 1, 0, 0, controller_m6_measure);
    Control_Motor(&huart7, 0, 2, 0, 0, controller_m6_measure+1);
    Control_Motor(&huart8, 0, 3, 0, 0, controller_m6_measure+2);

    for(int i=0; i<3; i++)
    {
        if((controller_m6_measure[i].ecd - controller_m6_measure[i].last_ecd) < -15000)
        {
            controller_m6_measure[i].ecd_count++;
        }
        else if((controller_m6_measure[i].ecd - controller_m6_measure[i].last_ecd) > 15000)
        {
            controller_m6_measure[i].ecd_count--;
        }

        controller_m6_measure[i].add_ecd = controller_m6_measure[i].ecd + (controller_m6_measure[i].ecd_count*32767);
        controller_m6_measure[i].angle = controller_m6_measure[i].add_ecd/32767.0*PI*2; //输出轴的机械刻度换算成角度

        controller_m6_measure[i].angle -= controller_m6_measure[i].init_angle; //初始角度清零

        controller_m6_measure[i].last_ecd=controller_m6_measure[i].ecd;
    }
}
    
```

图 52 自定义控制器部分代码

2.3.2.2. 模糊 PID

由于机械臂电机需要精密控制，为了使机械臂控制更加精确，我们采用了模糊 pid 算法来控制机械臂前端的 2006 电机和用于抬升的 3508 电机。

模糊 PID 控制是一种基于模糊逻辑的控制方法，它可以通过对 PID 控制器的参数进行实时调整，以克服传统 PID 参数无法实时调整的缺点。模糊 PID 控制涉及三个方面：模糊化、模糊推理确定模糊规则和清晰化。

其原理为把目标值 set 与输出值 out 的误差 e 与 e 的变化率 de/dt 作为模糊控制器的输入，通过模糊控制器先对输入值进行模糊化处理，接着进行模糊推理，最后把模糊推理的结果进行去模糊处理输出 PID 控制器的三个参数 k_p, k_i, k_d ，从而达到对 PID 控制器参数自适应整定的效果。

模糊控制原理分析：

(1) 模糊化

首先需要初始化论域：论域是一个根据控制模型所确定的范围，在工程机器人中由于不同结构参数不同，其论域也不同。以抬升为例由于输入 e, de/dt，输出 k_p, k_i, k_d 的范围各不相同，把输入映射到论域上更好统一处理。确定论域的范围后，再对论域进行划分。其速度环范围为 $[-3, 3]$ ，把论域平均分为 5 等份，即 $[-3, -2], [-2, -1], [-1, 0], [0, 1], [1, 2], [2, 3]$ 。接着把每个端点进行等级划分，依次为：NB(负大), NM(负中), NS(负小), ZO(零), PS(正小), PM(正中), PB(正大)。假设输入 e 的范围为 $[-10, 10]$ ，此刻 e 的值为 8，通过映射后的值为 2.4，2.4 在 $[2, 3]$ 区间，则该点在正中与正大之间。

(2) 隶属度函数的确定

工程中使用三角形隶属度函数，其值域为 $[0, 1]$ ，定义域为论域。若输入值 e 经过论域映射的值为 1.6，则 e 隶属于 PS 的百分比为 0.4，隶属于 PM 的百分比为 0.6。

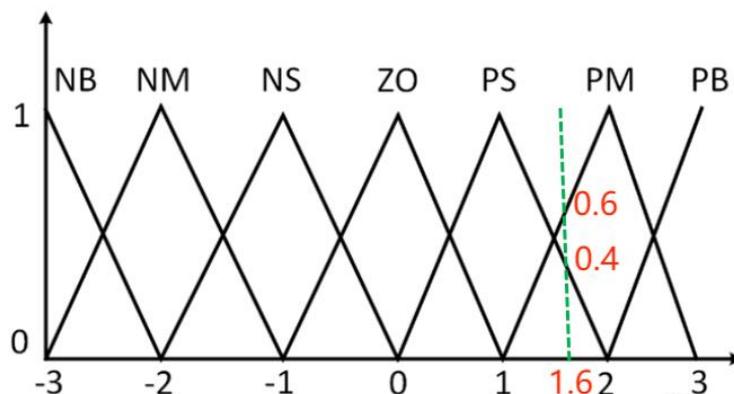


图 53 模糊化原理

之后确立模糊规则，输入量 e 与 de/dt 的模糊控制规则为：

U		de/dt						
		NB	NM	NS	ZO	PS	PM	PB
e	NB	PB	PB	PB	PB	PM	ZO	ZO
	NM	PB	PB	PB	PM	PM	ZO	ZO
	NS	PB	PM	PM	PS	ZO	NS	NM
	ZO	PM	PM	PS	ZO	NS	NM	NM
	PS	PS	PS	ZO	NM	NM	NM	NB
	PM	ZO	ZO	ZO	NM	NB	NB	NB
	PB	ZO	NS	NB	NB	NB	NB	NB

图 54 模糊控制规则

如若 e 的两个隶属度值为 PM、PB：

假设 e 属于 PM 的隶属度为 a ，则属于 PB 的隶属度为 $(1 - a)$ 。

再假设 de/dt 的两个隶属度值为 NB、NM。

假设 de/dt 属于 NM 的隶属度为 b ，则属于 NB 的隶属度为 $(1 - b)$ 。

根据模糊规则表,当 E 隶属于 PM 且 EC 隶属于 NB 时，可知输出值 U 隶属于 ZO，此时输出值 U 隶属于 ZO 的隶属度为 $a * (1 - b)$ ，其他同理。

根据以上求解规则，那么可以求得输出值 U 隶属于各个模糊子集的隶属度为：

隶属于 ZO 的隶属度为： $a * b + a * (1 - b) + (1 - a) * (1 - b)$ ；

隶属于 NS 的隶属度为： $(1 - a) * b$ ；

清晰化用隶属度值乘以隶属值求得输出值根据上一步结论求得输出值 $u = [a * b + a * (1 - b) + (1 - a) * (1 - b)] * ZO + (1 - a) * b * NS$ ；

根据模糊控制原理，先求出目标值 set 与输出值 out 的误差 e 与 e 的变化率 de/dt 的隶属度，再分别求出 K_p 、 K_i 、 K_d 的隶属度。 K_p 、 K_i 、 K_d 的模糊规则如下：

ΔK_p		de/dt隶属度						
		NB	NM	NS	ZO	PS	PM	PB
e 隶属度	NB	PB	PB	PM	PM	PS	ZO	ZO
	NM	PB	PB	PM	PS	PS	ZO	NS
	NS	PM	PM	PM	PS	ZO	NS	NS
	ZO	PM	PM	PS	ZO	NS	NM	NM
	PS	PS	PS	ZO	NS	NS	NM	NM
	PM	PS	ZO	NS	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB	

(1)

ΔK_i		de/dt隶属度						
		NB	NM	NS	ZO	PS	PM	PB
e 隶属度	NB	NB	NB	NM	NM	NS	ZO	ZO
	NM	NB	NB	NM	NS	NS	ZO	ZO
	NS	NB	NM	NS	NS	ZO	PS	PS
	ZO	NM	NM	NS	ZO	PS	PM	PM
	PS	NM	NS	ZO	PS	PS	PM	PB
	PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB	

(2)

ΔK_d		de/dt隶属度						
		NB	NM	NS	ZO	PS	PM	PB
e 隶属度	NB	PS	NS	NB	NB	NB	NM	PS
	NM	PS	NS	NB	NM	NM	NS	ZO
	NS	ZO	NS	NM	NM	NS	NS	ZO
	ZO	ZO	NS	NS	NS	NS	NS	ZO
	PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PM	PB	PS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB	

(3)

图 55 (1) (2) (3) 隶属度关系表

通过实时 e 隶属度与 de/dt 隶属度分别所对应的 K_p 、 K_i 、 K_d 的推理值计算出各自的隶属度，最后通过串级 pid 计算实现模糊 pid 的作用，达到双电机稳定协同控制。

2.3.2.3. 机械臂原理介绍

(1) 机械臂的知识架构

机械臂的运动需要运用机械臂运动学和机械臂动力学的知识。运动学 (kinematics) 是讨论运动状态本身位置 (X) 速度 (V) 加速度 (a) 时间 (t) 之间的关系, 包含移动副的位置、速度、加速度以及转动副的姿态、角速度、角加速度。

动力学 (Dynamics) 则是讨论力/力矩如何产生运动。需要根据牛顿第三定律、动能与势能的关系: $(T1+V1+U1-2=T2+V2)$ 及动能定理。

(2) 机械手臂的组成与描述

多个杆件 (link) 相串联, 具有复杂的几何外形杆件间可相对移动 (prismatic) 或转动 (revolute) 各个关节的运动由驱动器驱动来达到。而要描述手臂状态, 需要找出杆件间相对几何状态在各杆件上建立 frame, 以及 frame 状态代表杆件状态。并将机械臂拆解成关节 (Joint) 与杆 (Link) 来描述。每个转动或移动的 joint 具有一个自由度, 而每个 joint 对其特定的轴进行转动或移动。对于连接 joints 的杆件, 对其进行编号, 编号方式为:

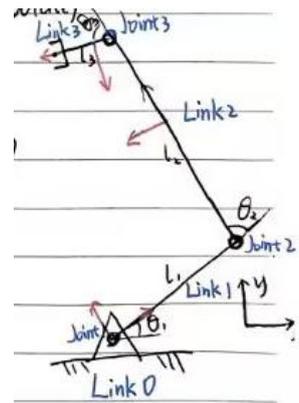


图 56 杆件转动图

Link0: 地杆, 不动的杆件; Link1: 和 Link0 相连, 第一个可动的杆件; Link2: 第二个可动的杆件...

(3) 机械臂各结构的几何表达

对空间中 2 个任意方向的轴, 两轴之间具有一段和此两个轴都互相垂直, 该线段长度记为 a_i , 两轴之间的夹角记为 α_i , 这样就建立起了两轴和其连接杆之间的几何关系。但若要多杆串联, 需要引入 (d_i, θ_i) 来描述 a_{i-1} 和 a_i 两相邻杆之间的关系。 d_i 表示 a_{i-1} 杆与 a_i 杆之间垂线的距离; θ_i 表示 a_{i-1} 杆与 a_i 杆之间的夹角。则串联杆件几何关系可表示为:

Revolute joint($\alpha_i, a_i, d_i, \theta_i$)

Prismatic joint($\alpha_i, a_i, d_i, \theta_i$)

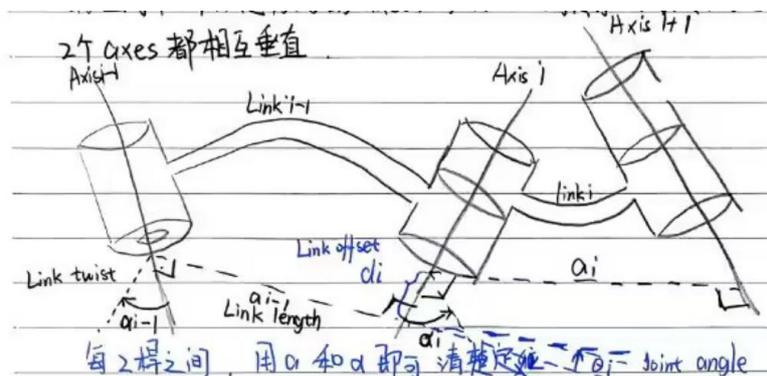


图 57 机械臂几何关系

为了更好的进行计算我们在杆件上建立 Frame (Craig DH 新版) :

向量 Z_i : 转动或移动 axis 的方向

向量 X_i : 沿着 a_i 方向 ($a_i \neq 0$) 和向量 Z_i 和向量 Z_{i+1} 两者垂直 ($a_i = 0$)

向量 Y_i : 与向量 X_i 和向量 Z_i 两者垂直, 依循右手定则。

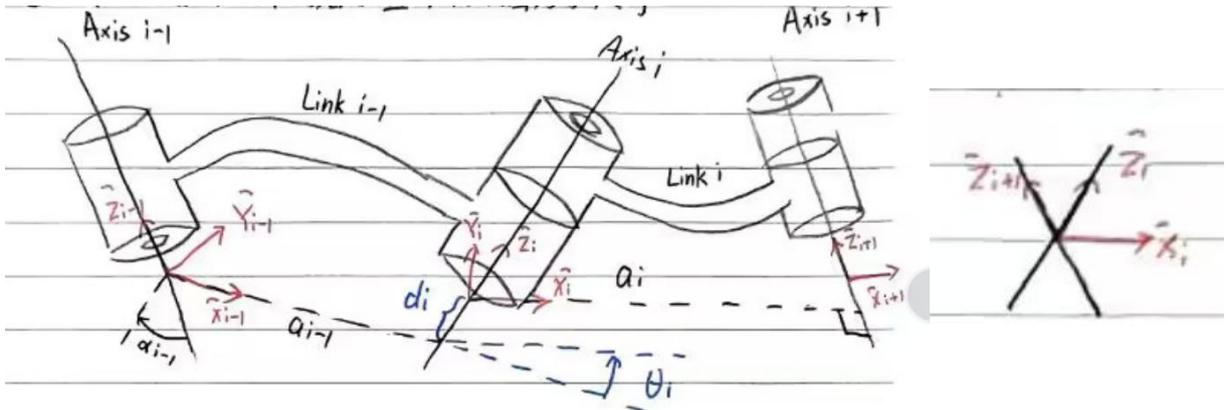


图 58 机械臂坐标系

(4) 特殊位置的处理

地杆 Link (0) 的坐标系与第一杆重合, 并使移动副关节角度 θ_i 设为 0, 转动副杆件距离 d_i 设为 0。Last link(n)的 X_n 取和向量 X_{n-1} 同方向以便更好的建模仿真。

2.3.2.4. 机械臂公式推导

(1) 机械臂正解

通过机械臂各个关节与杆件的几何关系所建立的坐标系通过 DH 参数表表示：

α_{i-1} :以向量 X_{i-1} 方向看，向量 Z_{i-1} 和 Z_i 之间的夹角

a_{i-1} :沿着向量 X_{i-1} 方向，向量 Z_{i-1} 和 Z_i 间的距离 ($a_i > 0$)

θ_i :以向量 Z_i 方向看，向量 X_{i-1} 和 X_i 间的夹角

d_i :沿着向量 Z_i 方向，向量 X_{i-1} 和 X_i 间的距离

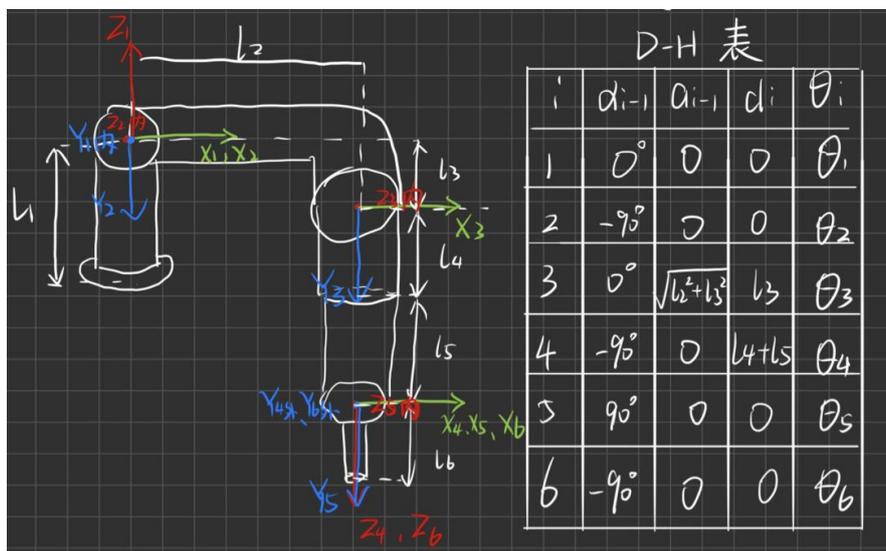


图 59 机械臂 DH 参数表

我们同时需要对机械臂初始位置进行设定，使上电后到达初始位置。

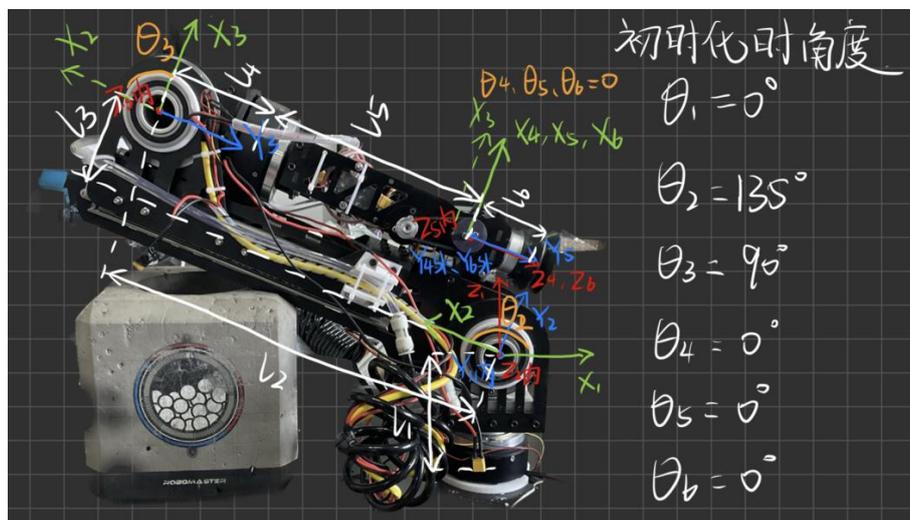


图 60 机械臂初始化参数

根据 DH 参数表构建变换矩阵:

$$\begin{aligned}
 {}^0_1 T &= \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^3_4 T &= \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 {}^1_2 T &= \begin{bmatrix} c\theta_2 & -s\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^4_5 T &= \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^2_3 T &= \begin{bmatrix} c\theta_2 & -s\theta_3 & 0 & a_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} & {}^5_6 T &= \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

根据矩阵乘法计算得出末端坐标在世界坐标系下的表达式:

$${}^0_6 T = {}^0_1 T {}^1_2 T {}^2_3 T {}^3_4 T {}^4_5 T {}^5_6 T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2) 机械臂逆解

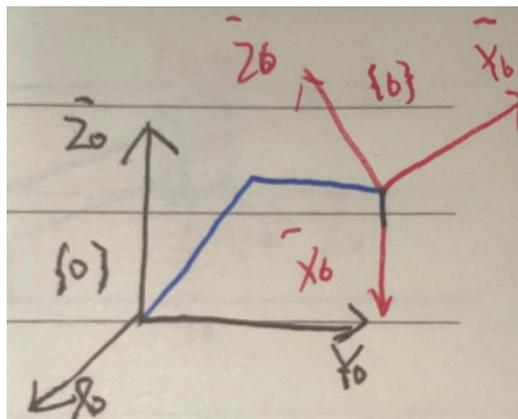


图 61 关节坐标关系图

逆向运动学为通过给定的末端关节坐标系解算各个轴转动角度。给予 $\{H\}$ 求得 θ_i 。该非线性超越方程组有六个未知数和六个限制条件。

$${}^0_6T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ & {}^0_6R_{3 \times 3} & & \\ & & 0 & 0 \\ & & & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ X_6 & Y_6 & Z_6 & R_{org} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

当我们求解时需要考虑末端关节是否可以达到的情况。有的位置手臂可以用一种以上的姿态到达，有的位置手臂可以用任何姿态到达，而所有这些可以到达的区间构成了机械手臂的运动范围（其中有些姿态会被遮挡或干涉）。

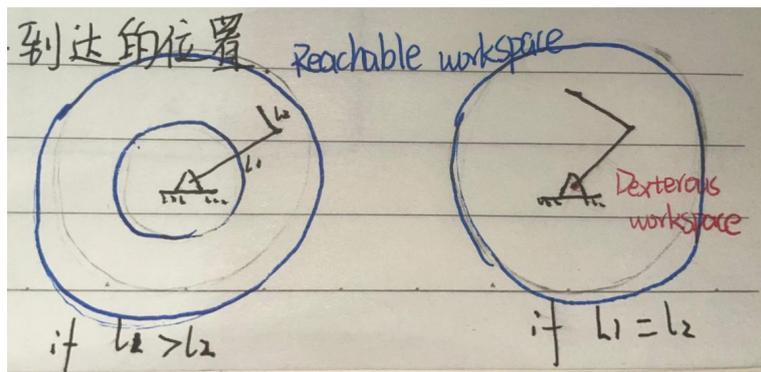


图 62 机械臂运动范围图

因此会出现多解的情况，解的数目由 joint 数目和 Link 参数决定。

对于六轴机械臂来说：

$$\begin{aligned} a_1 = a_3 = a_5 = 0 & \leq 4 \\ a_3 = a_5 = 0 & \leq 8 \\ a_3 = 0 & \leq 16 \\ \text{All } a_i \neq 0 & \leq 16 \end{aligned}$$

而对于本赛季工程机械臂结构来说，针对特定工作点由八组解每一个姿态中，具有两组手腕转动姿态。

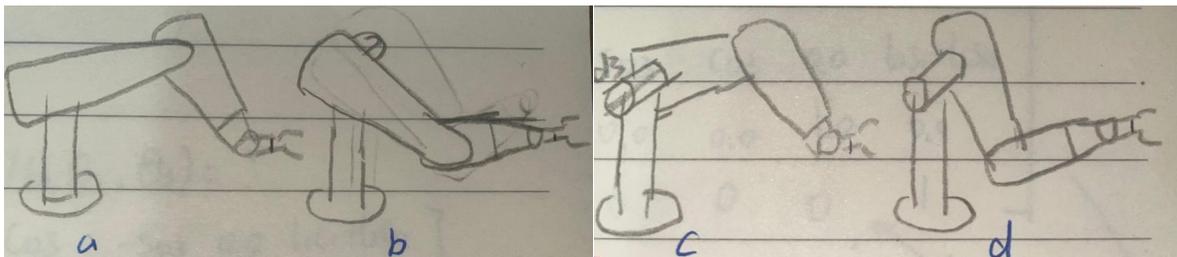


图 63 多解分析图

则：

$$\theta_4' = \theta_4 + 180^\circ$$

$$\theta_5' = -\theta_5$$

$$\theta_6' = \theta_6 + 180^\circ$$

若出现多重解，则确定解的方式原则为离目前状态最近的解且要避开障碍物。

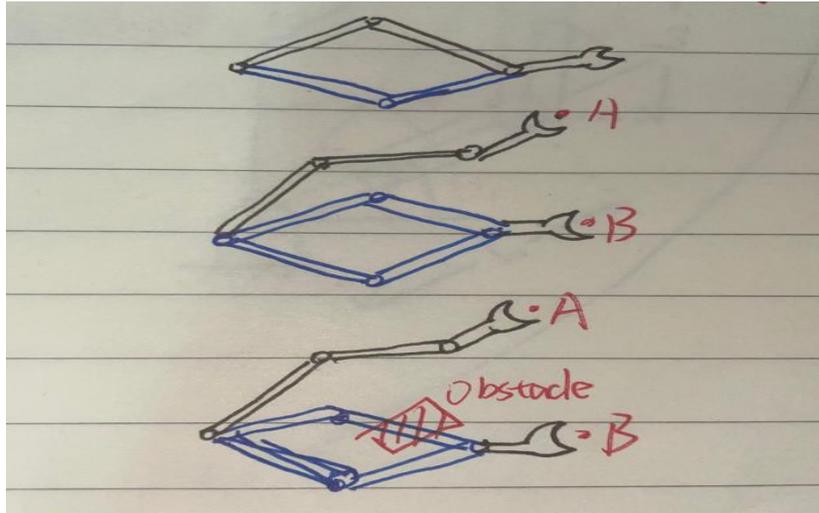


图 64 解的选择图

求解方式：

在设计初为了方便调试将后三个连续的轴交在同一点，对于前三轴空间上产生移动，后三轴空间上产生转动。

由于后三轴轴线交与一点，则

$${}^0R_6^{org} = {}^0R_4^{org}$$

$$\text{note: } {}_{i-1}^i T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a\alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = {}^0R_4^{org} = {}_1^0 T {}_2^1 T {}_3^2 T {}_4^3 T {}^0R_6^{org} = {}_1^0 T {}_2^1 T {}_3^2 T \begin{bmatrix} 0 & a_3 \\ -d_4 s\alpha_3 & \\ d_4 c\alpha_3 & \\ 1 & \end{bmatrix} = {}_1^0 T {}_2^1 T \begin{bmatrix} f_1(\theta_3) \\ f_2(\theta_3) \\ f_3(\theta_3) \\ 1 \end{bmatrix}$$

让 $\theta_1, \theta_2, \theta_3$ 层层分离

所以

$$\begin{bmatrix} f_1(\theta_3) \\ f_2(\theta_3) \\ f_3(\theta_3) \\ 1 \end{bmatrix} = {}_2^3 T \begin{bmatrix} a_3 \\ -d_4 s \alpha_3 \\ d_4 c \alpha_3 \\ 1 \end{bmatrix}$$

让 $\theta_1, \theta_2, \theta_3$ 层层分离, f 为 θ_1 的函数

$$f_1(\theta_3) = a_3 c_3 + d_4 s \alpha_3 s_3 + a_2$$

$$f_2(\theta_3) = a_3 c \alpha_2 s_3 - d_4 s \alpha_3 c \alpha_2 c_3 - d_4 s \alpha_2 c \alpha_3 - d_2 s \alpha_2$$

$$f_3(\theta_3) = a_3 c \alpha_2 s_3 - d_4 s \alpha_3 c \alpha_2 c_3 + d_4 s \alpha_2 c \alpha_3 + d_2 s \alpha_2$$

下一步

$${}^0 R_4^{org} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = {}_1^0 T {}_2^1 T \begin{bmatrix} f_1(\theta_3) \\ f_2(\theta_3) \\ f_3(\theta_3) \\ 1 \end{bmatrix} = {}_1^0 T \begin{bmatrix} g_1(\theta_2, \theta_3) \\ g_2(\theta_2, \theta_3) \\ g_3(\theta_2, \theta_3) \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 g_1 - s_1 g_2 \\ s_1 g_1 + c_1 g_2 \\ g_3 \\ 1 \end{bmatrix}$$

让 $\theta_1, \theta_2, \theta_3$ 层层分离, g 为 θ_2, θ_3 的函数

$$g_1(\theta_2, \theta_3) = c_2 f_1 - s_2 f_2 + a_1$$

$$g_2(\theta_2, \theta_3) = s_2 c \alpha_1 f_1 - c_2 c \alpha_1 f_2 - s \alpha_1 f_3 - d_2 s \alpha_1$$

$$g_3(\theta_2, \theta_3) = s_2 c \alpha_1 f_1 - c_2 c \alpha_1 f_2 + c \alpha_1 f_3 + d_2 c \alpha_1$$

令 r

$$= x^2 + y^2 + z^2 = g_1^2 + g_2^2 + g_3^2$$

$$= f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3 + 2a_1(c_2 f_1 - s_2 f_2)$$

$$= (k_1 c_2 + k_2 s_2) 2\alpha_1 + k_3$$

使用 k 函数让式子更加简洁

$$k_1(\theta_3) = f_1$$

$$k_2(\theta_3) = -f_2$$

$$k_3(\theta_3) = f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3$$

此外

$$z = g_3 = (k_1 s_2 - k_2 c_2) s \alpha_1 + k_4$$

$$k_1(\theta_3) = f_1$$

$$k_2(\theta_3) = -f_2$$

$$k_4(\theta_3) = f_3 c \alpha_1 + d_2 c \alpha_1$$

整合 r 和 z

$$\begin{cases} r = (k_1 c_2 + k_2 s_2) 2a_1 + k_3 \\ z = (k_1 s_2 - k_2 c_2) s \alpha_1 + k_4 \end{cases}$$

$$\text{if } a_1 = 0, r = k_3(\theta_3) = f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3$$

$$\text{if } s \alpha_1 = 0, z = k_4(\theta_3) = f_3 c \alpha_1 + d_2 c \alpha_1$$

$$\text{else } \frac{(r - k_3)^2}{4a_1^2} + \frac{(z - k_4)^2}{s^2 \alpha_1} = k_1^2 + k_2^2$$

则可解出 θ_3

最终，通过 $r = (k_1 c_2 + k_2 s_2) 2a_1 + k_3$ 解出 θ_2

通过 $r = c_1 g_1(\theta_2, \theta_3) - s_1 g_2(\theta_2, \theta_3)$ 解出 θ_1

前三轴转动角度解出来后则可得出 ${}^0_3R^{-1}$ 且已知 0_6R

$$\text{则 } {}^3_6R = {}^0_3R^{-1} {}^0_6R$$

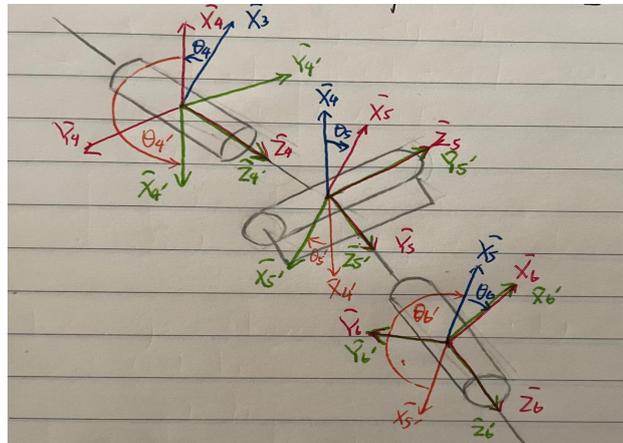


图 65 后三轴分解图

通过旋转矩阵求解方式

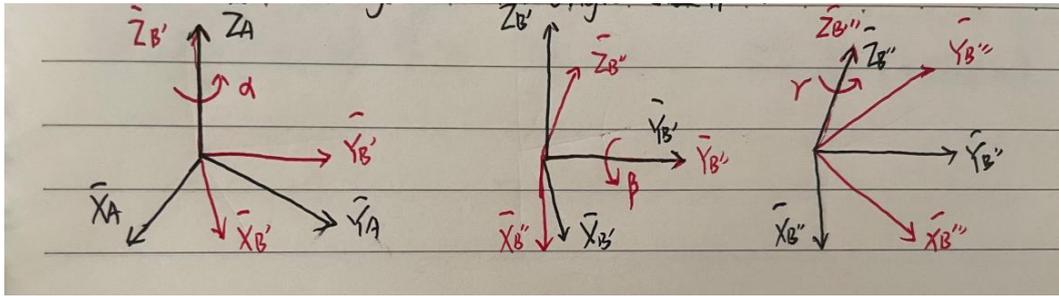


图 66 Z-Y-Z 分解图

$$\begin{aligned}
 {}^A_B R_{Z'Y'Z'}(\alpha, \beta, \gamma) &= R_{Z'}(\alpha)R_{Y'}(\beta)R_{Z'}(\gamma) \\
 &= \begin{bmatrix} cac\beta\gamma - sas\gamma & -cac\beta s\gamma - sac\gamma & cas\beta \\ sac\beta\gamma + cas\gamma & -sac\beta s\gamma + cac\gamma & sas\beta \\ -s\beta\gamma & s\beta s\gamma & c\beta \end{bmatrix} \\
 &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}
 \end{aligned}$$

if $\beta \neq 0^\circ$

$$\beta = \text{Atan2}(\sqrt{r_{31}^2 + r_{32}^2}, r_{33})$$

$$\alpha = \text{Atan2}(r_{23}/s\beta, r_{13}/s\beta)$$

$$\gamma = \text{Atan2}(r_{32}/s\beta, -r_{31}/s\beta)$$

if $\beta = 0^\circ$

$$\alpha = 0^\circ$$

$$r = \text{Atan2}(-r_{12}, r_{11})$$

if $\beta = 180^\circ$

$$\alpha = 0^\circ$$

$$r = \text{Atan2}(r_{12}, -r_{11})$$

所以 $\theta_4 = \alpha$; $\theta_5 = \beta$; $\theta_6 = \gamma$

(3) 速度规划

① 六轴机器人的速度雅可比：

在机器人学中，雅可比是一个把关节速度矢量 \mathbf{q} 变换成手抓相对基座标的广义速度矢量 \mathbf{v} 的变换矩阵，进行机器人速度分析时经常使用雅可比。根据之前课上所学的二轴机器人速度雅可比推算方法，对于六轴机器人来说，可以设其端点位置 X 、 Y 、 Z 、 ω_x 、 ω_y 、 ω_z 与关节 θ_1 、 θ_2 、 θ_3 、 θ_4 、 θ_5 、 θ_6 的关系为：

$$X = X(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$$

$$Y = Y(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$$

$$Z = Z(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$$

$$\omega_x = \omega_x(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$$

$$\omega_y = \omega_y(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$$

$$\omega_z = \omega_z(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$$

对其进行微分再写成矩阵的形式为（公式如下）：

$$\begin{bmatrix} dX \\ dY \\ dZ \\ d\omega_x \\ d\omega_y \\ d\omega_z \end{bmatrix} = \begin{bmatrix} \frac{\partial X}{\partial \theta_1} & \frac{\partial X}{\partial \theta_2} & \frac{\partial X}{\partial \theta_3} & \frac{\partial X}{\partial \theta_4} & \frac{\partial X}{\partial \theta_5} & \frac{\partial X}{\partial \theta_6} \\ \frac{\partial Y}{\partial \theta_1} & \frac{\partial Y}{\partial \theta_2} & \frac{\partial Y}{\partial \theta_3} & \frac{\partial Y}{\partial \theta_4} & \frac{\partial Y}{\partial \theta_5} & \frac{\partial Y}{\partial \theta_6} \\ \frac{\partial Z}{\partial \theta_1} & \frac{\partial Z}{\partial \theta_2} & \frac{\partial Z}{\partial \theta_3} & \frac{\partial Z}{\partial \theta_4} & \frac{\partial Z}{\partial \theta_5} & \frac{\partial Z}{\partial \theta_6} \\ \frac{\partial \omega_x}{\partial \theta_1} & \frac{\partial \omega_x}{\partial \theta_2} & \frac{\partial \omega_x}{\partial \theta_3} & \frac{\partial \omega_x}{\partial \theta_4} & \frac{\partial \omega_x}{\partial \theta_5} & \frac{\partial \omega_x}{\partial \theta_6} \\ \frac{\partial \omega_y}{\partial \theta_1} & \frac{\partial \omega_y}{\partial \theta_2} & \frac{\partial \omega_y}{\partial \theta_3} & \frac{\partial \omega_y}{\partial \theta_4} & \frac{\partial \omega_y}{\partial \theta_5} & \frac{\partial \omega_y}{\partial \theta_6} \\ \frac{\partial \omega_z}{\partial \theta_1} & \frac{\partial \omega_z}{\partial \theta_2} & \frac{\partial \omega_z}{\partial \theta_3} & \frac{\partial \omega_z}{\partial \theta_4} & \frac{\partial \omega_z}{\partial \theta_5} & \frac{\partial \omega_z}{\partial \theta_6} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}$$

图 67 雅可比矩阵

其中 J 就为所需要用到的六个自由度的速度雅可比矩阵，推而广之，对于 n 自由度机器人，关节变量可用广义关节变量 \mathbf{q} 表示，即 $\mathbf{q} = (q_1, q_2, q_3, \dots, q_n)$ 。

对于转动关节, $q_i = \theta_i$, 机器人末端在操作空间的位置和方位可用末端手抓的位姿 X 表示, 为一个 6 维变量, dX 可写成 $dX=J(q)dq$:

$$J(q) = \frac{\partial X}{\partial q^T} = \begin{bmatrix} \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \dots & \frac{\partial X}{\partial q_n} \\ \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \dots & \frac{\partial X}{\partial q_n} \\ \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \dots & \frac{\partial X}{\partial q_n} \\ \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \dots & \frac{\partial X}{\partial q_n} \\ \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \dots & \frac{\partial X}{\partial q_n} \\ \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \dots & \frac{\partial X}{\partial q_n} \\ \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \dots & \frac{\partial X}{\partial q_n} \\ \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \dots & \frac{\partial X}{\partial q_n} \end{bmatrix}$$

图 68 雅可比求解

② 如何使用速度雅可比求出手部速度:

在具体实际操作的课程设计中, 我们已知的是各关节速度 q , 在结合上述的六轴机器人速度雅可比矩阵, 根据课上所学的公式 $q=J^{-1}v$ (J^{-1} 为机器人的逆速度雅可比) 可以推导出 $v=qJ$, 就可以求出所需要知道的手部末端瞬时速度了。

若速度太快, 转动惯量太大, 容易损伤电机或者发生危险, 速度太慢, 控制运动效率太低, 影响进程。雅可比矩阵通过末端速度来实现各关节电机速度的计算, 通过矩阵计算计算出雅可比矩阵的逆, 再乘以末端速度后便可计算各关节的加速度, 由于路径规划将路程分解很多距离, 各关节速度也就近似于加速度的值。

(4) 六轴机器人的关节空间轨迹规划 (MATLAB 五次多项式插值法)

关节空间轨迹规划为把关节变量转换为与时间的函数, 再对角速度与加速度约束, 其优点为计算小, 易控制, 避免机构奇异性等。对于五次多项式有 6 个待定系数, 同时约束起点和终点的角度、角速度, 角加速度, 关节满足:

$$\begin{aligned} \theta(t) &= a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \\ \dot{\theta}(t) &= a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4 \\ \ddot{\theta}(t) &= 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3 \end{aligned}$$

将相邻两点作为一小段轨迹的起点和终间，用 θ_0 和 θ_f 表示，起始速度为 v_0 ，终止速度为 v_f ，有以下式子：

$$\begin{aligned} \theta(t_0) &= \theta_0 = a_0 \\ \theta(t_f) &= \theta_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ \dot{\theta}(t_0) &= \dot{\theta}_0 = a_1 \\ \dot{\theta}(t_f) &= \dot{\theta}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ \ddot{\theta}(t_0) &= \ddot{\theta}_0 = 2a_2 \\ \ddot{\theta}(t_f) &= \ddot{\theta}_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \end{aligned}$$

把约束条件代入对应公式解出系数，即可得到五次多项式插值法的轨迹：

$$\begin{aligned} a_0 &= \theta_0 \\ a_1 &= \dot{\theta}_0 \\ a_2 &= \ddot{\theta}_0 / 2 \\ a_3 &= \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3} \\ a_4 &= \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \\ a_5 &= \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5} \end{aligned}$$

现在设置一个一维轨迹做为例子，对其参数设置如图所示：

表 1 参数表

序号	位置 (m)	速度 (m/s)	加速度 (m/s ²)	时间 (s)
1	0	0	0	0
2	30	5	10	3
3	100	15	20	6
4	50	-10	-10	12
5	0	0	0	15

2.3.2.5. MATLAB 中的建模与仿真

Link 函数：用于定义六轴机器人的一个轴。包含了机器人的运动学参数、动力学参数、刚体惯性矩参数、电机和传动参数；

SerialLink 函数：用于构建机械臂。它的类函数比较多，包括显示机器人、动力学、逆动力学、雅可比等；

fkine 正解函数：用于求解出末端位姿 p 。

ikine 逆解函数：用于求解出关节角 q 。

jtraj 函数：已知初始和终止的关节角度，利用五次多项式来规划轨迹；

ctrjaj 函数：已知初始和终止的末端关节位姿，利用匀加速、匀减速运动来规划轨迹。

(1) MATLAB 建模

```

theta(1) = 0; d(1) = 90; a(1) = 0; alp(1) = pi/2;
theta(2) = 0; d(2) = 0; a(2) = 355; alp(2) = 0;
theta(3) = 0; d(3) = 0; a(3) = 0; alp(3) = pi/2;
theta(4) = 0; d(4) = 396; a(4) = 0; alp(4) = -pi/2;
theta(5) = 0; d(5) = 0; a(5) = 0; alp(5) = pi/2;
theta(6) = 0; d(6) = 70; a(6) = 0; alp(6) = 0;
% DH parameters th d a alpha sigma
L1 = Link([theta(1), d(1), a(1), alp(1), 0]);
L2 = Link([theta(2), d(2), a(2), alp(2), 0]);
L3 = Link([theta(3), d(3), a(3), alp(3), 0]);
L4 = Link([theta(4), d(4), a(4), alp(4), 0]);
L5 = Link([theta(5), d(5), a(5), alp(5), 0]);
L6 = Link([theta(6), d(6), a(6), alp(6), 0]);
robot = SerialLink([L1, L2, L3, L4, L5, L6]);
robot.name='QI_qi';
robot.display()
theta = [0, 0, 0, 0, 0, 0]*pi/180;
robot.teach();
robot.plot(theta);
t = robot.fkine(theta); %末端执行器位姿

% joint_limit
L1.qlim=[-180,180]/180*pi;
L2.qlim=[-180,180]/180*pi;
L3.qlim=[-180,180]/180*pi;
L4.qlim=[-180,180]/180*pi;
L5.qlim=[-180,180]/180*pi;
L6.qlim=[-180,180]/180*pi;
%关节工作空间
  
```

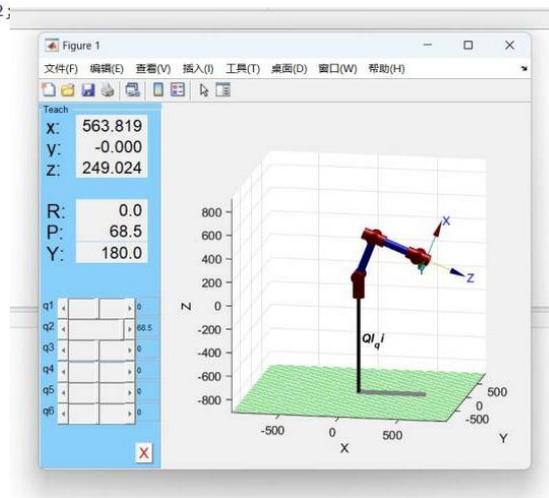


图 69 MATLAB 建模

(2) MATLAB 正解

```

theta1=20;
theta2=20;
theta3=20;
theta4=20;
theta5=20;
theta6=20;

d1=90;
a2=355;
d4=396;
d6=70;

%将角度转化为弧度
Q1=theta1/180*pi;
Q2=theta2/180*pi;
Q3=theta3/180*pi;
Q4=theta4/180*pi;
Q5=theta5/180*pi;
Q6=theta6/180*pi;

T_01 = [ cos(Q1), 0, sin(Q1), 0
         sin(Q1), 0, -cos(Q1), 0
         0, 1, 0, d1
         0, 0, 0, 1];

T_12 = [ cos(Q2), -sin(Q2), 0, a2*cos(Q2)
         sin(Q2), cos(Q2), 0, a2*sin(Q2)
         0, 0, 1, 0
         0, 0, 0, 1];

T_23 = [ cos(Q3), 0, sin(Q3), 0
         sin(Q3), 0, -cos(Q3), 0
         0, 1, 0, 0
         0, 0, 0, 1];

T_23 = [ cos(Q3), 0, sin(Q3), 0
         sin(Q3), 0, -cos(Q3), 0
         0, 1, 0, 0
         0, 0, 0, 1];

T_34 = [ cos(Q4), 0, -sin(Q4), 0
         sin(Q4), 0, cos(Q4), 0
         0, -1, 0, d4
         0, 0, 0, 1];

T_45 = [ cos(Q5), 0, sin(Q5), 0
         sin(Q5), 0, -cos(Q5), 0
         0, 1, 0, 0
         0, 0, 0, 1];

T_56 = [ cos(Q6), -sin(Q6), 0, 0
         sin(Q6), cos(Q6), 0, 0
         0, 0, 1, d6
         0, 0, 0, 1];

T_06=T_01*T_12*T_23*T_34*T_45*T_56;
nx=T_06(1,1);
ny=T_06(2,1);
nz=T_06(3,1);
ox=T_06(1,2);
oy=T_06(2,2);
oz=T_06(3,2);
ax=T_06(1,3);
ay=T_06(2,3);
az=T_06(3,3);
px=T_06(1,4);
py=T_06(2,4);
pz=T_06(3,4);

```

图 70 MATLAB 正解

(3) MATLAB 逆解

```

%rad2deg
%%设一个矩阵
TI_06 = [nx, ox, ax, px
         ny, oy, ay, py
         nz, oz, az, pz
         0, 0, 0, 1];

T_15=simplify(T_12*T_23*T_34*T_45); %中间四个矩阵乘起来
T1theta=simplify(inv(T_01)*TI_06*simplify(inv(T_56))); %与
%%解析法:通过正运动学得出齐次变换矩阵, 前三个列为第六个轴相对
T_06=[ 0.5322, -0.1137, 0.8390, 611.4
       -0.4697, -0.8641, 0.1809, 213.8
       0.7044, -0.4903, -0.5133, -127.9
       0, 0, 0, 1];

nx=T_06(1,1);
ny=T_06(2,1);
nz=T_06(3,1);
ox=T_06(1,2);
oy=T_06(2,2);
oz=T_06(3,2);
ax=T_06(1,3);
ay=T_06(2,3);
az=T_06(3,3);
px=T_06(1,4);
py=T_06(2,4);
pz=T_06(3,4);

%(px!=70*ax)反解
the_ik1=rad2deg(atan2(py-70*ay,px-70*ax)); %两组解 (y x) (-)
Q1=the_ik1/180*pi;

Q3=the_ik3/180*pi;
the_ik2=rad2deg(atan2(pz-70*az-90,sqrt((355+396*sin(Q3))^2...
+(396*cos(Q3))^2-(pz-70*az-90)^2))-atan2(-396*cos(Q3),355+396*sin(Q3)));
%theta2=rad2deg(2*atan((355+396*sin(Q3)-sqrt((355+396*sin(Q3))^2+(396*cos(Q3))^2))/
Q2=the_ik2/180*pi;

T_03=T_01*T_12*T_23;
T_03_inv =simplify(inv(T_03));

T_36=simplify(T_34*T_45*T_56);
T_36theta=simplify(T_03_inv*T_06);
%the_ik5=rad2deg(acos(ax*cos(Q2+Q3)*cos(Q1)-az*cos(Q2+Q3)+ay*sin(Q2...
the_ik5=rad2deg(atan2(ax*sin(Q2+Q3)*cos(Q1)-az*cos(Q2+Q3)...
+ay*sin(Q2+Q3)*sin(Q1),-sqrt(1-(ax*sin(Q2+Q3)*cos(Q1)-az*cos(Q2...
+Q3)+ay*sin(Q2+Q3)*sin(Q1))^2))-atan2(1,0));
Q5=the_ik5/180*pi;

%the_ik4=rad2deg(asin((px*sin(Q1)-py*cos(Q1))/70/sin(Q5)));
the_ik4=rad2deg(atan2(px*sin(Q1)-py*cos(Q1),sqrt((70*sin(Q5))^2-(px*sin(Q1)...
-py*cos(Q1))^2))-atan2(0,70*sin(Q5))); %+-sqrt
Q4=the_ik4/180*pi;
%70*sin(Q4)*sin(Q5)=px*sin(Q1)-py*cos(Q1);
%the_ik6=rad2deg(asin((oy*sin(Q2+Q3)*cos(Q1)-oz*cos(Q2+Q3)+ox*sin(Q2+...
the_ik6=rad2deg(atan2(oy*sin(Q2+Q3)*sin(Q1)-oz*cos(Q2+Q3)...
+ox*sin(Q2+Q3)*cos(Q1),sqrt((sin(Q5))^2-(oy*sin(Q2+Q3)*sin(Q1)...
-oz*cos(Q2+Q3)+ox*sin(Q2+Q3)*cos(Q1))^2))-atan2(0,oy*sin(Q2...
+Q3)*sin(Q1)-oz*cos(Q2+Q3)+ox*sin(Q2+Q3)*cos(Q1)));
Q6=the_ik6/180*pi;

Q_ALL=[Q1,Q2,Q3,Q4,Q5,Q6];
robot.plot(Q_ALL);
theta_IK_ALL=[the_ik1,the_ik2,the_ik3,the_ik4,the_ik5,the_ik6];

```

图 71 MATLAB 逆解

(4) MATLAB 轨迹规划

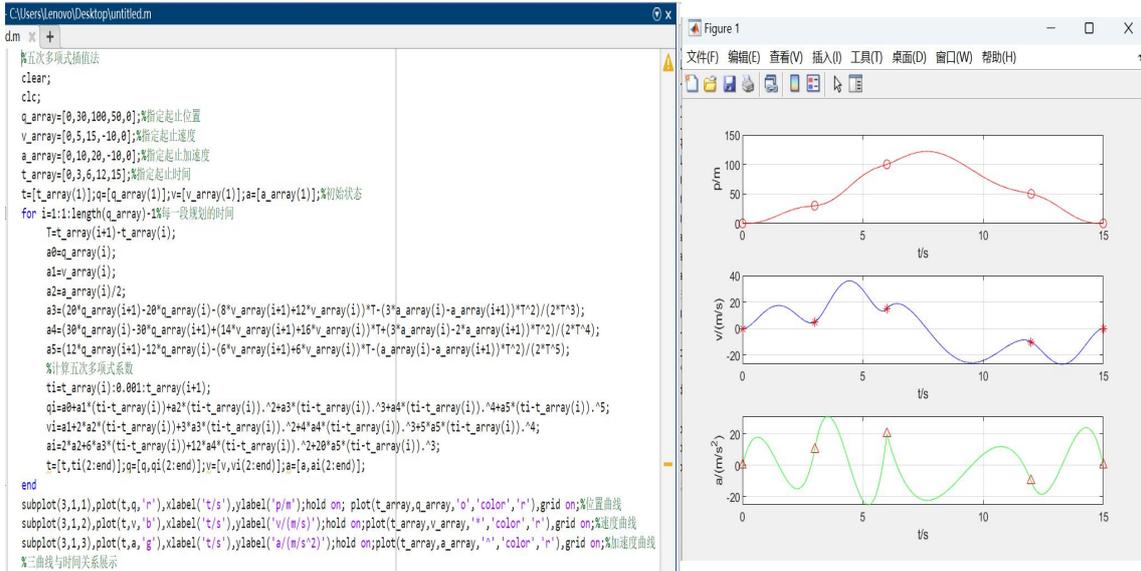


图 72 MATLAB 轨迹规划

经过上述代码的仿真运算，可得本次五次多项式插值法的轨迹规划曲线效果如图所示：

以上即为我们战队的机械臂结算与路径规划，由此得到的机械臂代码能使机械臂比较好控制与使用，成功达到完美兑换矿石的任务。

2.4. 导航算法设计

我们的工程机器人自动兑换功能尚不完善，其半自动模式不具备实战价值，但导航研发较为完全，作为技术储备写在下面。

2.4.1. 功能简介

采用激光雷达实现工程机器人全地图导航，使工程机器人可以到达大资源岛、小资源岛和兑换站，实现取矿和兑矿的功能。

2.4.2. 重要算法原理阐述、公式推导

2.4.2.1. 全局导航算法 A*

(1) 原理：

A*算法是一种在图形中查找最短路径的算法。它结合了最佳优先搜索的启发式方法和Dijkstra算法的节点遍历机制，通过为每个节点计算一个优先级 ($f(n) = g(n) + h(n)$)，并据此搜索下一个节点。其中， $g(n)$ 表示从起点到当前节点的实际代价， $h(n)$ 表示从当前节点到目标节点的估计代价（启发式函数）， $f(n)$ 则是两者的和，表示从起点经过当前节点到目标节点的估计总代价。

在搜索过程中，A*算法会维护两个列表：`open list`（开放列表）和 `close list`（关闭列表）。`open list` 存储待考察的节点，并按照 $f(n)$ 的值进行排序，优先考察 $f(n)$ 值最小的节点。`close list` 则存储已经考察过的节点，避免重复搜索。

(2) 公式推导：

A*算法的核心公式是 $f(n) = g(n) + h(n)$ 。其中， $g(n)$ 是实际代价，通常可以通过累加从起点到当前节点每一步的代价得到。 $h(n)$ 是估计代价，需要根据具体问题来定义。常见的启发式函数有欧几里得距离、曼哈顿距离等。

对于路径搜索问题，状态就是图中的节点，代价就是距离。因此， $g(n)$ 可以表示为从起点到当前节点的最短距离， $h(n)$ 可以表示为当前节点到目标节点的估计距离。 $f(n)$ 则是这两者的和，表示从起点经过当前节点到目标节点的估计总距离。

在搜索过程中，A*算法会不断地从 `open list` 中选择 $f(n)$ 最小的节点进行扩展，并更新其相邻节点的 $g(n)$ 和 $f(n)$ 值。如果某个节点的 $g(n)$ 值已经大于当前 `open list` 中最小的 $f(n)$ 值，

那么该节点就不会被扩展，因为它不可能构成更短的路径。当目标节点被加入 open list 时，搜索结束，此时就找到了从起点到目标点的最短路径。

需要注意的是，A 算法的正确性和效率与启发式函数 $h(n)$ 的选取密切相关。如果 $h(n)$ 始终小于等于实际从当前节点到目标节点的最短距离（即 $h(n) \leq h^*(n)$ ），那么 A* 算法一定能找到最短路径。同时， $h(n)$ 的选取也影响着搜索的效率和范围。因此，在实际应用中需要根据具体问题来选择合适的启发式函数。

2.4.2.2. 局部导航算法 TEB

原理：

TEB 算法是一种结合了全局路径规划和局部轨迹生成的导航算法。它首先通过全局路径规划器生成一条从起点到终点的粗略路径，然后利用 TEB 算法进行局部轨迹优化。在 TEB 算法中，路径被视为一条具有时间和位置信息的弹性带，通过调整弹性带的形状和位置来优化机器人的运动轨迹。

具体来说，TEB 算法通过在当前位置附近搜索可能的轨迹，并根据 TEB 约束对这些轨迹进行评估和选择。约束包括速度、加速度等限制，由机器人自身决定。同时，轨迹的选择也会考虑与障碍物的距离、与规划路径的跟随程度等因素。通过为每个影响因素设计奖励或惩罚函数，TEB 算法可以对可行的路线进行打分，并选择得分最高的局部路线作为机器人的运动轨迹。

在优化过程中，TEB 算法会考虑机器人的动力学约束和障碍物信息，以生成平滑且安全的运动轨迹。通过不断调整和优化弹性带的形状和位置，TEB 算法能够实现全局路径规划和局部轨迹生成的有效融合，使机器人在复杂环境中实现自主导航。

推导过程：

TEB 算法（Time Elastic Band）的公式推导涉及多个方面，包括轨迹表示、约束条件、优化目标等。由于 TEB 算法较为复杂，具体的公式推导过程可能较为繁琐，并且可能涉及较为深入的数学知识和优化技术。

在 TEB 算法中，轨迹通常通过一系列带有时间戳的控制点来表示。这些控制点不仅包含位置信息，还包含速度、加速度等动态信息。约束条件则包括机器人的速度、加速度限制，以及与障碍物的最小距离等。这些约束条件确保了生成的轨迹既符合机器人的动力学特性，又能避免与障碍物发生碰撞。

优化目标是 TEB 算法中的核心部分，它定义了如何评估和调整轨迹。常见的优化目标包括轨迹的平滑性、安全性、与规划路径的跟随程度等。这些目标通常通过定义相应的成本函数来实现，成本函数衡量了轨迹在不同方面的性能表现。

在优化过程中，TEB 算法采用一种优化算法（如梯度下降、非线性优化等）来求解最优轨迹。这涉及到对成本函数进行求导、构建雅可比矩阵和 Hessian 矩阵等数学操作，以找到使成本函数最小化的轨迹参数。

由于 TEB 算法的实现细节和具体应用场景可能有所不同，因此具体的公式推导过程也会有所差异。在实际应用中，需要根据具体需求和机器人特性来设计和调整 TEB 算法的相关参数和约束条件。

因此，由于 TEB 算法的复杂性，具体的公式推导需要深入理解算法的原理和数学基础，并结合具体的实现细节进行推导。如果您对 TEB 算法的公式推导有深入的需求，建议查阅相关的学术论文或实现案例，以获取更详细和具体的公式推导过程。

2.4.3. 算法库介绍

2.4.3.1. Navigation

(1) `navigation` 是 ROS 的二维导航功能包，是根据输入的里程计等传感器的信息流和机器人的全局位置，通过导航算法，计算得出安全可靠的机器人速度控制指令。

(2) `map_server`: 地图服务器，主要功能是保存地图和导入地图。

(3) `costmap_2d`: 可以生产代价地图，以及提供各种相关的函数。

(4) `robot_pose_ekf`: 扩展卡尔曼滤波器，输入是里程计、IMU、VO 中的任意两个或者三个，输出是一个融合之后的 `pose`。

(5) `localization`: 这里是两个定位用的 `package`。`fake_localization` 一般是仿真用的，`amcl` 才是实际定位用的 `package`。

(6) `nav_core`: 这里面只有三个文件，对应的是全局路径规划、局部路径规划、`recovery_action` 的通用接口定义，具体功能实现则是在各个对应的规划器 `package` 里。

(7) `move_base`: 这里实现的是整个导航的流程。什么时候调用全局路径规划、什么时候调用局部路径规划、什么时候调用 `recovery_action` 都是这个 `package` 管的。就是下图中间方框里做的事情，可以说是整个 `navigation stack` 的核心。

2.4.3.2. FAST-LIO2

一种快速、鲁棒、通用的激光雷达-惯性里程计框架。FAST-LIO2 基于高效的紧密耦合迭代卡尔曼滤波器，具有两个关键的新颖之处，可实现快速，稳健和准确的激光雷达导航(和测绘)。

第一种方法是直接将原始点注册到地图上然后更新地图，而不提取特征。这样可以利用环境中的细微特征，从而提高准确性。消除手工设计的特征提取模块，也使其能够适应不同扫描模式的新兴激光雷达。

第二个主要的新颖之处是通过增量 k-d 树数据结构(ikd-Tree)来维护映射，这使得增量更新(即激光点插入、删除)和动态重新平衡成为可能。与现有的动态数据结构(八叉树, R*-tree, nanoflann k-d tree)相比, ikd-Tree 进行下采样时可以获得更好的整体性能。

2.4.3.3. FAST_LIO_LOCALIZATION

一种快速的激光雷达定位技术，它利用激光雷达扫描周围环境，通过对激光点云数据的处理和分析，确定机器人在环境中的位置和姿态。

2.5. 其它

我们在工程的 UI 界面上显示底盘移动模式，矿石对位模式，工程吸盘模式，实时展示每一自由度状态，展示自由度是否到达对应位置。方便操作手及时了解，避免因紧张影响操作。此外，拥有程序中记录的抓取矿石数，也包括一键矿石对位是否触发和是否开启气泵及是否开启自瞄兑矿模式。

以下为代码：

```
Graph_Data Aim[7];
String_Data strSPEED;
void UIRefreshTask_Loop(void)
{
    char tmp1[30]={0},tmp2[30]={0},tmp3[30]={0};
    memset(&strSPEED,0,sizeof(strSPEED));
    for(int k=0;k<6;k++)
    {
        memset(&Aim[k],0,sizeof(Aim[k]));
        //清空图形数据
        Line_Draw(&Aim[0],"AL1",UI_Graph_ADD,5,UI_Color_Green,3,960,300,960,540);
        Line_Draw(&Aim[1],"AL2",UI_Graph_ADD,5,UI_Color_Green,2,940,520,980,520);
        Line_Draw(&Aim[2],"AL3",UI_Graph_ADD,5,UI_Color_Black,3,930,478,990,478);
        Line_Draw(&Aim[3],"AL4",UI_Graph_ADD,5,UI_Color_Yellow,2,920,380,1000,380);
        //可通过宽度
        Line_Draw(&Aim[4],"AL5",UI_Graph_ADD,6,UI_Color_Green,2,440,0,627,443);
        Line_Draw(&Aim[5],"AL6",UI_Graph_ADD,6,UI_Color_Green,2,1480,0,1293,443);
        Line_Draw(&Aim[6],"AL7",UI_Graph_ADD,5,UI_Color_Orange,3,920,400,1000,400);
        Char_Draw(&strSPEED,"CAP",UI_Graph_ADD,8,UI_Color_Green,20,strlen(tmp1),2,860,100,tmp1);
        Char_ReFresh(strSPEED);
        vTaskDelay(50);
        UI_ReFresh(7,Aim[0],Aim[1],Aim[2],Aim[3],Aim[4],Aim[5],Aim[6]);
    }
}

while(1)
{
    if(rc_ctrl.key.v&KEY_PRESSED_OFFSET_Z)
    {
        Line_Draw(&Aim[0],"AL1",UI_Graph_ADD,5,UI_Color_Green,3,960,300,960,540);
        Line_Draw(&Aim[1],"AL2",UI_Graph_ADD,5,UI_Color_Green,2,940,520,980,520);
        Line_Draw(&Aim[2],"AL3",UI_Graph_ADD,5,UI_Color_Black,3,930,478,990,478);
        Line_Draw(&Aim[3],"AL4",UI_Graph_ADD,5,UI_Color_Yellow,2,920,380,1000,380);
        //可通过宽度
        Line_Draw(&Aim[4],"AL5",UI_Graph_ADD,6,UI_Color_Green,2,440,0,627,443);
        Line_Draw(&Aim[5],"AL6",UI_Graph_ADD,6,UI_Color_Green,2,1480,0,1293,443);
        Line_Draw(&Aim[6],"AL7",UI_Graph_ADD,5,UI_Color_Orange,3,920,400,1000,400);
        Char_Draw(&strSPEED,"CAP",UI_Graph_ADD,8,UI_Color_Green,20,strlen(tmp1),2,860,100,tmp1);
        Char_ReFresh(strSPEED);
        vTaskDelay(50);
        UI_ReFresh(7,Aim[0],Aim[1],Aim[2],Aim[3],Aim[4],Aim[5],Aim[6]);
    }
}
```

3. 研发迭代过程

3.1. 测试记录

测试功能点	测试时间/环境	测试设备
吸盘系统	2023/11/15 地点：实验室 室温：20°C	铠盟 KMDP-C4 有刷真空泵 ZPT50HBS、ZPT63HBS 吸盘
机械臂基础功能	2024/1/20 地点：实验室 室温：17° C	机械臂 铠盟 KMDP-C4 有刷真空泵 ZPT63HBS 吸盘
底盘基础功能	2024/1/30 地点：实验室 室温：17° C	工程机器人底盘 自制盲道 15° 坡
银矿获取机构	2024/3/10 地点：创新创业楼大厅 室温：22° C	工程机器人底盘（安装银矿获取机构） 铠盟 KMDP-C4 有刷真空泵

3.2. 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	机械臂初步设计，制作。控制代码还没有完成，电机过载问题还未暴露。	2023.11.15
V1.0.1	代码初步成型，可以控制机械臂移动，但不够流畅。	2024.12.20
V1.1	底盘研发完毕，工程机器人初步成型，可以模拟兑换五级矿石，以及获取大小资源岛内矿石。但是电机过载问题开始显现	2024.1.20
V1.1.1	银矿获取机构成型，可以一次性获取多块银矿石。但仍未解决电机过载问题。	2024.2.27
V1.2	安装简易重力补偿机构，能够一定程度上缓解电机过热	2024.3.15

3.3. 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	吸盘吸紧矿石后在运动过程中如果底盘突然停止或者矿石碰到障碍物，矿石可能会掉落	所选用的吸盘过软，如果矿石受到径向力，可能会使矿石翘起，使吸盘与矿石之间产生缝隙，导致气泵旁通大气，丧失吸力	解决方案：给吸盘设计支撑架，在吸盘吸住矿石，吸盘本体开始压缩时，给矿石提供支持力，增大支撑面积，从而阻止漏气。 解决效果：可以吸紧矿石，在可能出现的碰撞情况下保证矿石不掉落	V1.0	机械工程师： 张昊阳
2	机械臂运动过程中会有抖动	机械臂在运动过程中每个关节电机的载荷都不是线性变化，导致原有参数不适用	解决方案：在代码中添加滤波函数 实际效果：抖动有所降低，运动速度稍慢	V1.0.1	硬件工程师： 薛福祥 嵌入式软件工程师： 李徽之、薛福祥
3	机械臂线路不稳定，容易出现断线的情况	机械臂所用关节电机的信号线与电源线插口都相对于安装面凸起，导致前期设计的垫块不能很好的给走线提供空间，且之前的线路在调试过程中损伤过大，出现破损。	解决方案：将机械臂关节位置拆散，线路全部更新，针对线路易损位置，更换走线方式，比如增加固定点等。 解决效果：机械臂线路在之后的没有出现过之前的问题，稳定性显著提高。	V1.1.1	机械工程师： 张昊阳，王浩 硬件工程师： 薛福祥 嵌入式软件工程师： 薛福祥
4	机械臂在伸长伸高后底部Pitch轴电机出现过热的问题，会导致触发电机保护，导致断电。	机械臂设计过程中并未很好的计算电机载荷，导致机械臂在兑换四级及以上矿石时会出现电机过热。	解决方案：在机械臂底座和第一级之间安装拉簧，补偿其重力。 解决效果：加装改进结构后，在兑换较为简单的四级姿态时，机械臂可以保证长时间不过热，不会断电导致兑换失败。	V1.2	机械工程师： 张昊阳、王浩 嵌入式软件工程师： 薛福祥

4. 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%)
张昊阳	机械设计制造及其自动化, 大二, 工程机器人负责人, 机械结构负责人	负责整个机器人的机械结构开发, 机器人与自定义控制器的维护	29%
薛福祥	机械设计制造及其自动化(卓越), 大二, 嵌入式软件与硬件负责人	负责整个机器人的嵌入式软件部分与硬件部分	29%
踪金坤	机械电子工程, 大三, 协助开发嵌入式软件部分	负责工程机器人机械臂解算部分	14%
李国瑞	材料工程, 研一, 项管, 协助开发机械结构	负责工程机器人自定义控制器的设计	14%
李徽之	机械设计制造及其自动化(中外), 大二, 负责图传链路开发	负责自定义控制器图传链路部分代码开发	14%

5. 参考文献

[1]皓月. 【RM2023-工程机器人开源】沈阳航空航天大学-TUP 战队.

<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22727&fromuid=75243>, 2023-8-19

[2]Splinting. 【RM2023-工程机器人机械结构开源】南方科技大学-ARTINX 战队.<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22759>, 2023-9-10

[3]金金鱼. 【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>, 2023-9-21

[4]Socratios. 【RM2023-工程开源】上海交通大学-云汉交龙战队.<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22758>, 2023-9-9

1.1

完整形态考核结束后，我们认为原有机机器人的各项性能不具备上场条件，故只保留了原有底盘结构，仍然采用了传统的“龙门架”抬升结构，重新设计了一款五轴机械臂负责获取矿石以及兑换矿石。

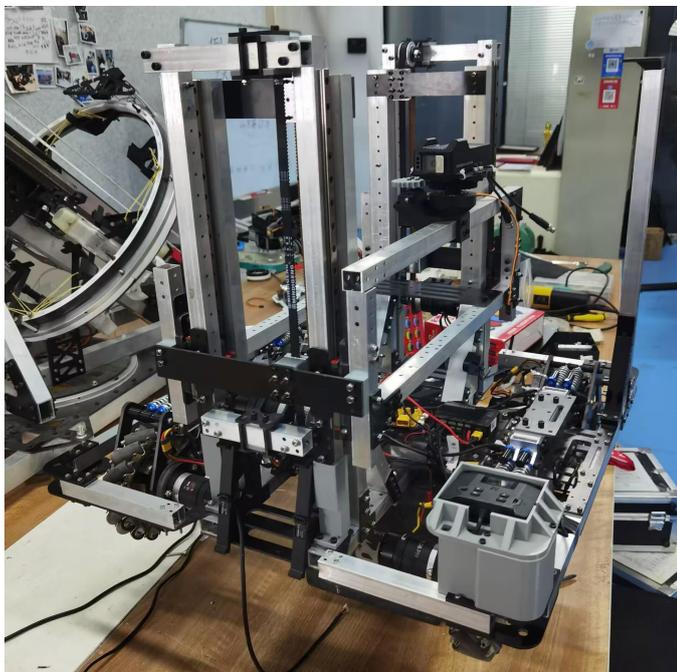


图 73 新版方案抬升结构实物图

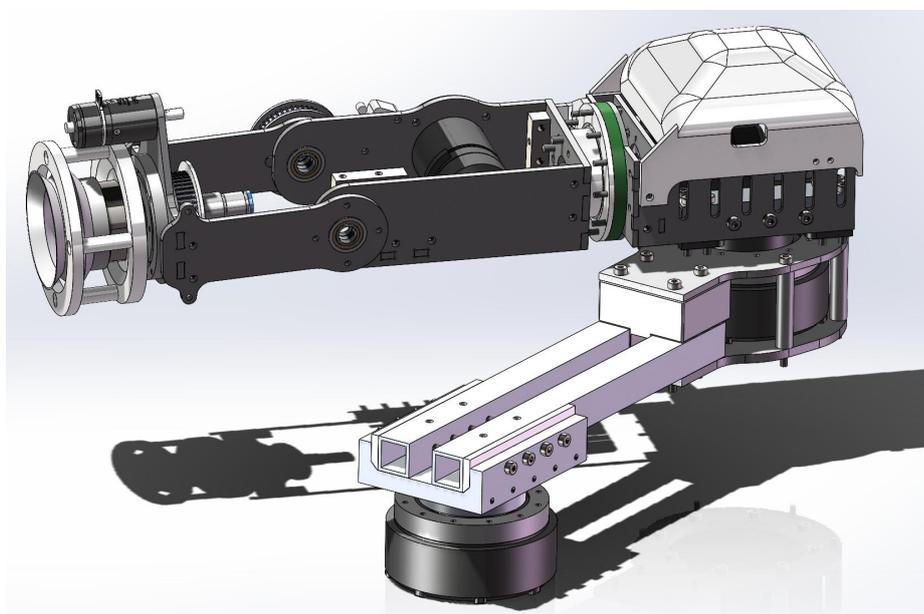


图 74 新版机械臂示意图

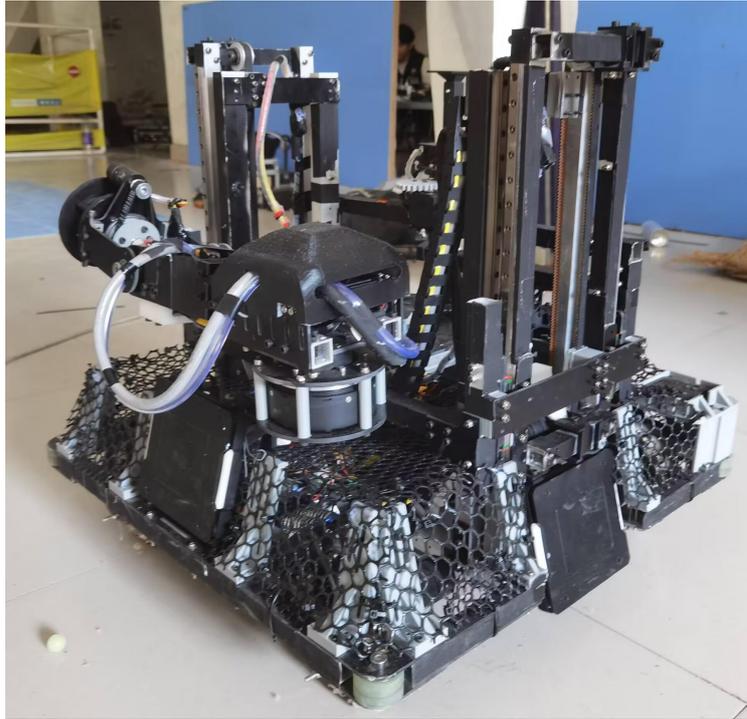


图 75 新版工程机器人实物图

由于技术方案描述的工程机器人并未上场，故以下内容将根据新版工程机器人的场上表现来进行撰写。

1.1.1 赛场性能表现情况分析

在 2024 赛季区域赛中，我队共参与了三场比赛，在这三场比赛中，工程机器人共计获取 8 块金矿石、0 块银矿石，共计兑换五块金矿石（4 块 3 级，1 块 4 级）共获得 1414 金币，三级矿石平均兑换时间 15 秒，四级矿石兑换时间 22 秒，并未在对局中尝试兑换五级矿石。整体来讲，本赛季的工程机器人基本符合心理预期，但与赛季初的规划有较大差距。

在赛场上我方为工程机器人制定的战术为：在前哨站停转之前，尽可能的兑换矿石，为英雄机器人购买大弹丸提供经济，前哨站停转之后，工程机器人要配合步兵机器人击杀对方英雄机器人。对于主要的获取经济的任务，工程机器人的矿石兑换能力尚可，能够基本保障前中期英雄机器人的经济，但是获取矿石的能力较差，获取矿石花费的时间往往接近 1 分钟，导致在前哨站停转之前只能完成一块矿石的获取与兑换。

1.1.2 赛场性能表现与规划对比分析

机器人最初设计的六轴机械臂虽制作完毕且进行了代码调试，但是并未达到上场标准，原因主要有 Pitch1 轴关节电机选型存在问题，扭矩不足，会导致电机过热，触发断电保护；并且机械臂运动学解算并未突破，自定义控制器重量较大，操作者无法很好的控制机械臂，故将工程机器人方案进行更改。

但是原定的银矿获取机构并未实施，导致在场上获取矿石效率低下，原因主要是改版后工程机器人没有空间容纳原有的银矿获取机构，且时间紧张，故将该机构从计划中移除。

在最初的规划中，我们计划一分钟之内获取三块银矿石。而新设计的抬升机构存在一定的机械旷量，导致机械臂伸平时并不与地面水平，而无论是银矿石还是金矿石都需要将矿石竖直上抬才可以取出，且小资源岛留给矿石的余量较小，场地材料摩擦力较大，故导致本方案获取矿石效率较低。

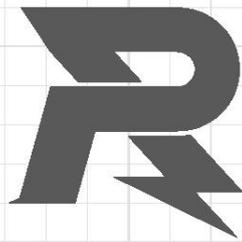
由于原定的自定义控制器方案为示教器结构，随着机械臂方案的改变，原有自定义控制器不再适用于新结构，新版自定义控制器因为时间原因只停留在图纸阶段。

1.1.3 经验总结

我队目前研发能力较弱，且经费不算充足，没有足够的试错成本，因此在开发机器人的过程中应该保留原有方案的优点、改进缺点，进行技术迭代。

机械组在进行机械结构设计的过程中要考虑到队伍电控的水平，切勿出现结构较好但代码无法实现的情况；再者，还要好好利用论坛里面的开源文档以及图纸，很多其他队伍的研究成果以及避开的设计误区都是十分宝贵的设计经验。

在前期的三维设计、加工零件、组装整车这三个环节中，机械组进度一定要压缩时间，一是要为电控组成员调试机器人留出足够的时间，因为代码的成型是很慢的，尤其是电机数量多、控制难度大的工程机器人；二是由于很多未知问题无法从图纸上得出，故要为后期修改方案留出时间。



邮箱: robomaster@dji.com

论坛: <http://bbs.robomaster.com>

官网: <http://www.robomaster.com>

电话: 0755-36383255 (周一至周五10:30-19:30)

地址: 广东省深圳市南山区西丽街道仙茶路与兴科路交叉口大疆天空之城T2 22F