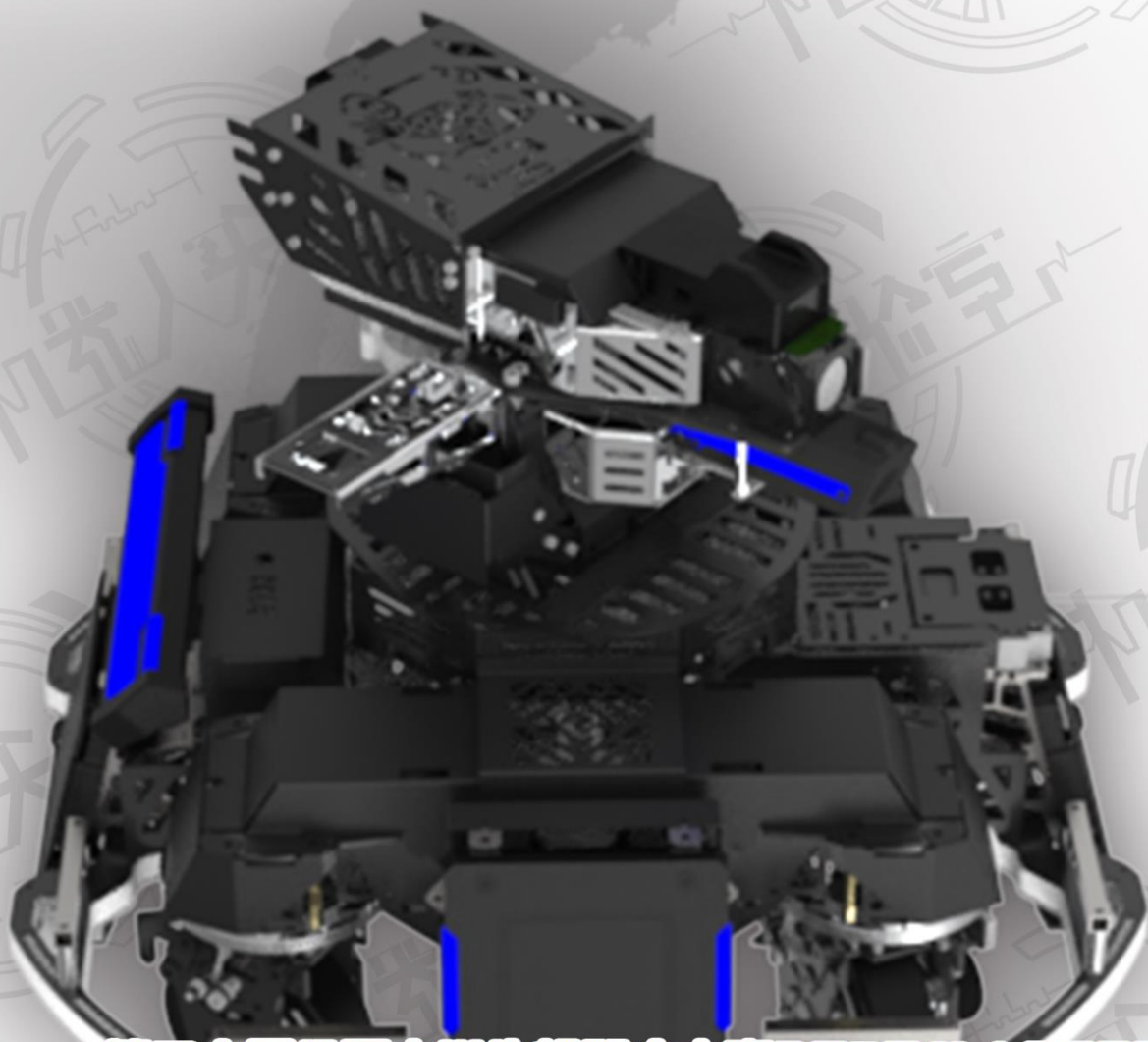




华南理工大学

South China University of Technology



第二届全国大学生机器人大赛ROBOMASTER2021

华南虎舵轮步兵机器人 设计报告

华南虎战队

— 华南理工大学机器人实验室 —

目录

1. 机器人概述	4
1.1 战术定位	4
1.2 主要思路	4
1.3 实现的技术方案	4
2. 机器人的主要技术参数	5
2.1 基本参数	5
2.2 执行器件	5
3. 机械设计分析	5
3.1 舵轮底盘设计方案	5
3.1.1 设计思路	5
3.1.2 救援机构设计	6
3.1.3 轮组设计	9
3.1.4 悬挂设计	10
3.2 云台设计方案	12
3.3 17mm 小弹丸发射机构设计方案	12
3.3.1 拨盘部分	12
3.3.2 发射部分	13
4. 机器人电路设计	14
4.1 底盘功率控制电路板设计	14
4.1.1 超级电容充电部分	14
4.1.2 采样部分	15
4.1.3 最小系统部分	15
4.1.4 逻辑部分供电	16
4.1.5 输出开关控制部分	17
4.1.6 超级电容充放电控制策略	17
4.2 超级电容模块	17
4.3 电池前端升压板	18

4.4 电调前端升压板.....	18
5. 机器人嵌入式控制.....	20
5.1 程序主要理论	20
5.2 嵌入式算法说明.....	21
5.3 代码设计介绍	21
6. 机器人视觉算法	22
6.1 代码框架	22
6.2 装甲检测模块	22
6.3 神符模块/干扰模块	24
7. 人机交互设计.....	24
7.1 自定义 UI 界面.....	24
7.2 状态指示灯.....	24
7.3 操作手便捷按键（半自动化设计）	25
8. 开发工具.....	25
9. 测试.....	25
9.1 小陀螺转速测试.....	25
9.2 下台阶、过盲道测试.....	26

1. 机器人概述

1.1 战术定位

舵轮底盘具有机动性强、弹丸躲避率高的优势，是地面对战强势单位。

通过击杀当下最首要的敌方目标，舵轮步兵能控制比赛节奏，比如击杀敌方英雄、工程、哨兵，干扰敌方打符等，并保护英雄推倒敌方基地。

1.2 主要思路

舵轮步兵机器人的设计重点为机动性和小陀螺性能的稳定。

1.3 实现的技术方案

表 1-1：舵轮步兵机器人技术指标

舵轮步兵机器人技术指标		
发射	射速	极差 1.0 以内，射速可控
	射频	最高射频：1 秒 15 发、单发可控无误射
	稳定性	1000 发以内弹丸射速均值掉速不超过 1.0
云台	pitch 轴	俯角 15°仰角 40°，满足击打地面机器人、哨兵和建筑的需求，30°的响应时间在 200ms 以内
	yaw 轴	可以 360°转动并且 30°的响应时间在 200ms 以内
底盘	防撞	保护舵轮车与步兵车、英雄车、工程车相向全速对撞、小陀螺对撞以及与水泥墙面全速冲撞后车辆无损坏
	悬挂	保证平面上四轮组同时着地且在行程内与地面的夹角基本维持在 90°，吸收机器人下台阶时的冲击
	运动性能	100w 时平移最大速度可以达到 3m/s 以上，小陀螺速度可以达到 6rad/s 以上，上坡时间可以稳定在 3s 以内，可以小陀螺上坡

2. 机器人的主要技术参数

2.1 基本参数

表 1-2: 舵轮步兵机器人基本参数

基本参数	
重量	24.7kg
长	595mm
宽	595mm
高	495mm
重心高度	180.7mm

2.2 执行器件

表 1-3: 舵轮步兵机器人执行器件

执行器件	数量	用途
M3508	4	底盘动力输出
GM6020	4	底盘转舵输出
M2006	1	拨盘动力输出
M3508 转子	2	摩擦轮动力输出
GM6020	2	云台 pitch 轴和 yaw 轴动力输出

3. 机械设计分析

3.1 舵轮底盘设计方案

3.1.1 设计思路

3.1.1.1 方案分析

目前步兵底盘形式大部分采用麦轮底盘，麦轮底盘能简单地实现全向移动，设计简单，死亡后可被拖动的优势，但也有缺点。麦轮底盘的自旋和横向平移需要轮组的速度分解与损耗来实现，能量的利用率低。

针对麦轮底盘的缺点，设计者选择舵轮底盘。舵轮底盘每个轮子可以转到任意角度，不需要速度的分解和抵消就能实现全向移动，响应更快，自旋和平移的能量利用率更高，控制更加灵活。但舵轮底盘本身有个致命缺点：无特殊装置时，舵轮底盘断电后不能像麦轮底盘一样随意拖动，工程机器人难以对其进行拖拽到基地进行复活。

3.1.1.2 需求分析

表 3-1: 舵轮底盘需求分析表

需求分析	结构设计
舵轮底盘需要实现单个轮组的 360°无限旋转。	轮组的设计上沿用上一届队员的轮组设计。舵向电机使用 GM6020 电机，驱动电机选择 M3508 电机，GM6020 电机的空心轴穿过导电滑环，实现单

需求分析	结构设计
	个轮组的 360°无限旋转。
死亡时可被工程机器人拖拽。	设计底盘的救援机构，使得轮组能在舵轮模式与有偏心距的购物车模式间切换。
底盘防撞做成圆形获近似圆形，以此获得更好的通过性，避免接触场地或其他机器人不易被卡住的情况。	底盘做成圆形后，底盘空间减小，轮组本身占用空间大，导致整个底盘对空间较为敏感。
底盘的悬挂稳定性有较高的需求。	针对与起伏较大的赛场，悬挂的设计要有定量的分析。且由于底盘空间较小，选择了简单耐用的单摇臂独立悬挂。

3.1.2 救援机构设计

3.1.2.1 理论与计算分析

由于舵轮底盘的每个轮子是普通的胶轮，并不像麦轮和全向轮一样能直接进行拖拽，故需要设计救援机构实现工程机器人的对其救援。现实中胶轮能被拖动的案例，比如超市的购物车、四轮行李箱，它们被拖动时，各个轮子被迫转向被拖动的方向。

而购物车四个轮和舵轮底盘结构上不同的是：购物车的轮子是个万向轮，轮子的触地点不在轮组 yaw 轴轴线上，存在一个偏心距，使得我们在拖动购物车时，购物车车体给轮组一个转矩，使得轮组朝向拖动的方向。而舵轮底盘的轮子触地点与轮组 yaw 轴重合，而无法产生使舵转向的转矩。

设计者采取一种，在轮支架上加装一套连杆和一个舵机，使其能够控制轮支架的形状，在舵轮模式和有偏心距的购物车模式间切换。运用 motion 运动仿真，得出舵机所需的力矩小于 2000N·mm，故选用了 20kg·cm 的舵机。

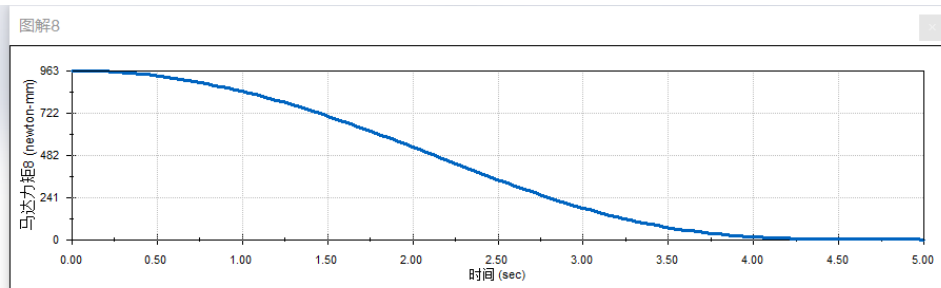


图 3-1：动力仿真下舵机施加力矩

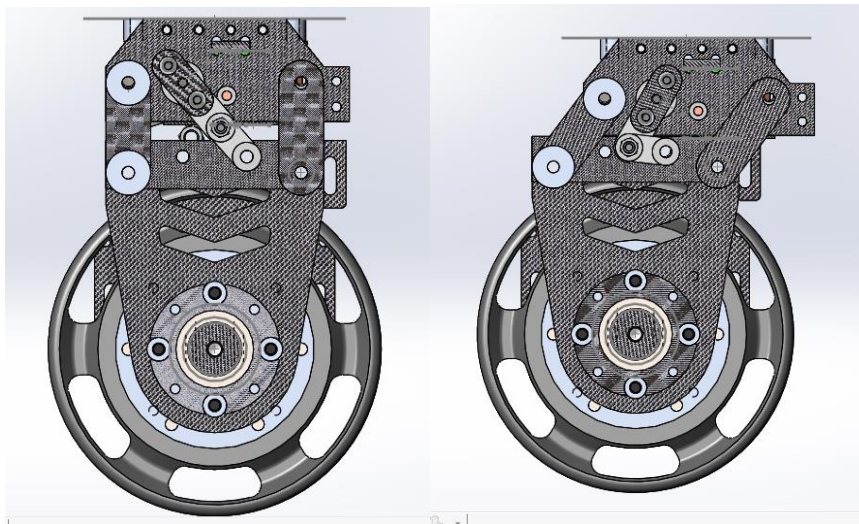


图 3-2：无偏心距模式与有偏心距模式

3.1.2.2 机构设计

该机构设计的一个难点是如何使用舵机给两侧的连杆同时施力以达到目的，起初使用单轴舵机，并用一根细长轴连接另一侧连杆进行传动。

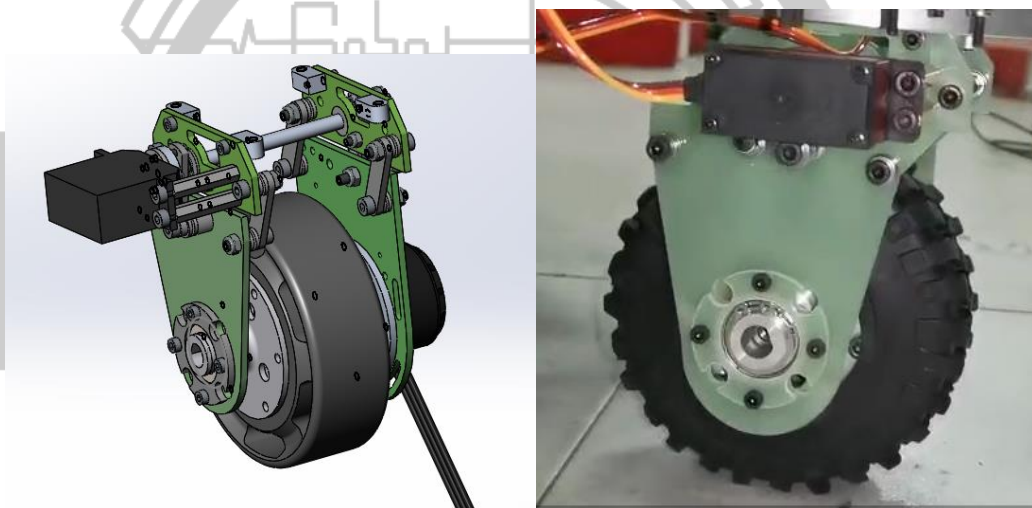


图 3-3：细长轴传动救援机构

搭出测试机构实物后进行了简单测试，实现了基本的功能：
负重后救援机构能在断电后变形为有偏心距模式，上电后变形为无偏心距模式；
负重后且断电之后，能由工程机器人进行拖拽救援。

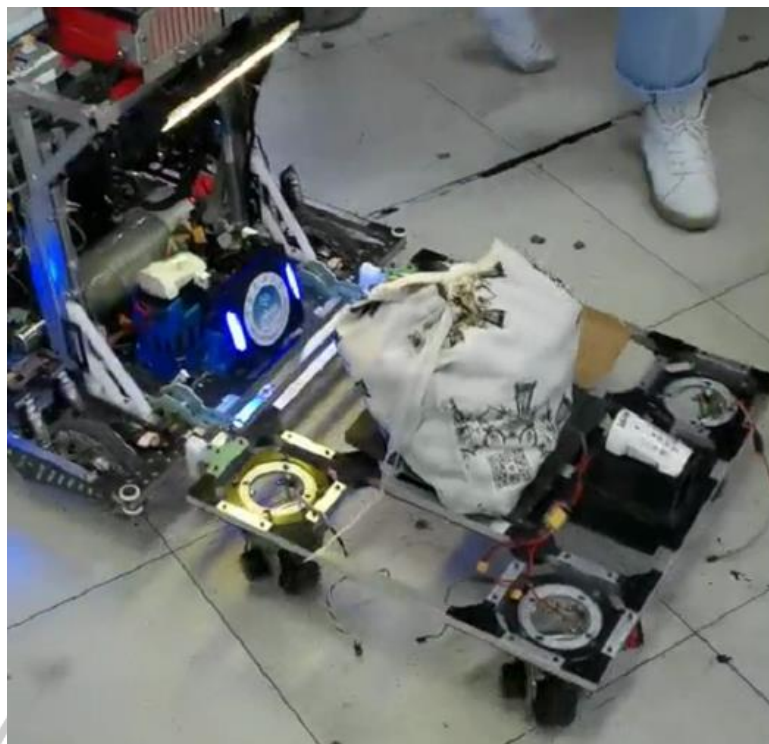


图 3-4：救援示意

这一版救援机构虽然基本满足了需求，但是缺点显而易见：

舵机外置，导致轮组所需回转半径极大。

细长轴传动件，细长轴受力时转角可能较大。

针对其缺点，设计者选择了一种双轴舵机，与单轴舵机的区别是：一端是输出轴，一端是转动副，可以使用 U 型传动件进行传动。设计者将舵机藏在两轮支架中间，传动件经过应力仿真符合要求，第二版救援机构成功解决了上述问题。

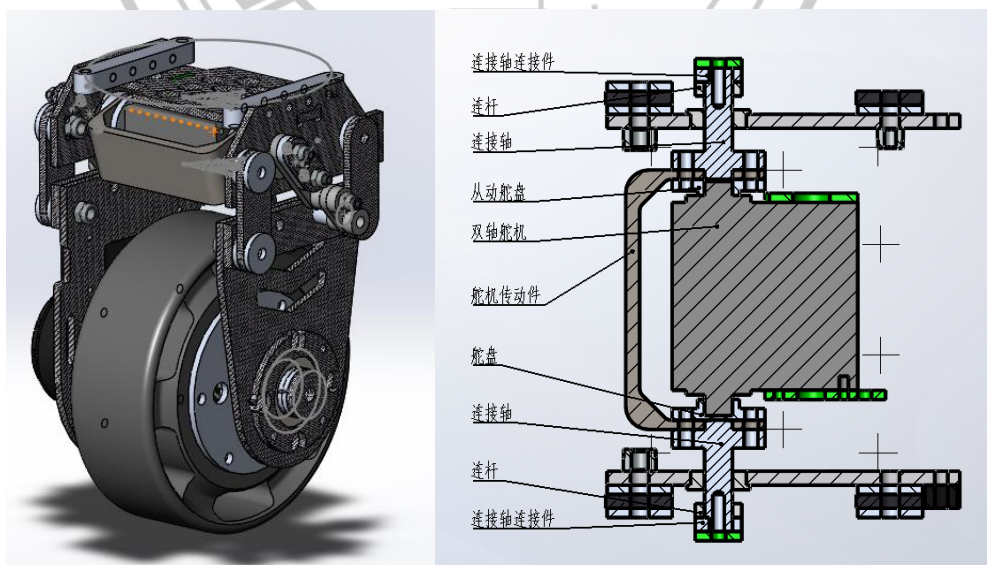


图 3-5：双轴舵机内置版救援机构

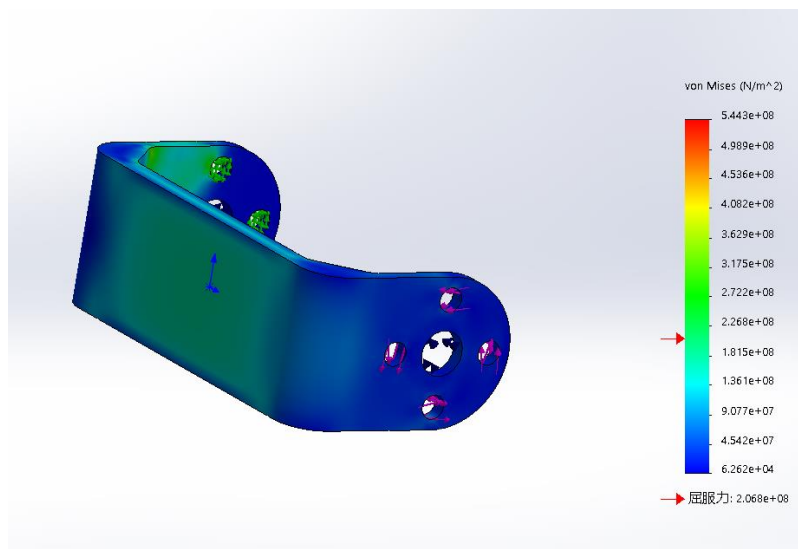


图 3-6：传动件应力仿真

总的来说，该机构满足了轮组变形以供救援的需求，但仍存在一定问题：救援机构增重较多。增加救援机构后，单个轮组相较于原来的轮组增重将近 1kg。

- 1) 工况恶劣，舵机容易烧坏。实际使用中，经过起伏路段及下台阶等情况下，舵机存在受冲击的情况，舵轮步兵测试和训练中存在舵机损坏的情况。
- 2) 维修性差。由于机构复杂，舵机的拆卸比较麻烦，更换烧坏的舵机需要较长的时间，更换一个舵机大概需要 15 分钟。

3.1.3 轮组设计

轮组设计基本沿用上一届队员的舵轮轮组设计。转向部分选用 GM6020 电机，6020 有着自带的绝对值编码器，且中间空心轴可以穿导电滑环，实现轮组 360°无限旋转，6020 电机与舵轮有非常高的适配率。驱动部分将 M3508 电机藏于轮毂之中，尽可能地缩小了回转半径并使得质心接近轴线，减小转动惯量，避免产生较大的机械振动不利于结构的寿命。

另外一点细节，但是会很大影响维护和检修的一点是四个电机固定螺钉，都在反向延伸的空间内做了镂空，保证螺钉可以直接取出和拧进，而不需要将轮组拆开来才能完成检修。

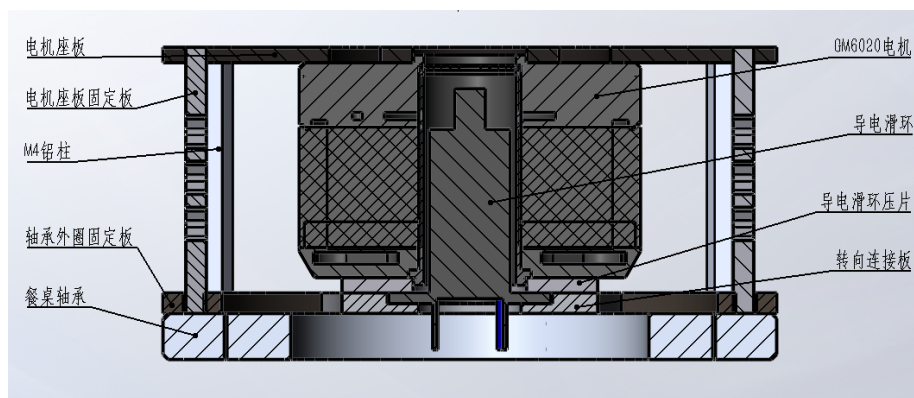


图 3-7：21RM 轮组设计（转向部分）

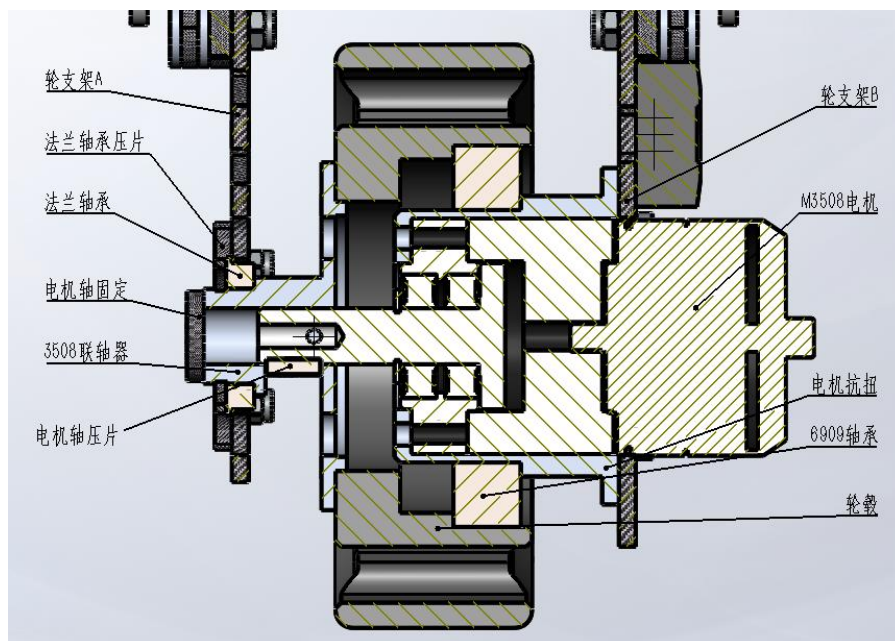


图 3-8: 21RM 轮组设计 (驱动部分)

3.1.4 悬挂设计

对于舵轮底盘，比较适合的悬挂需要保证的是，无论悬挂的弹簧如何受压变形，都要保证轮组相对于地面的直立（或者相对于斜坡要垂直于斜坡），上一年的悬挂设计采用一种平行四边形的连杆结构，利用平行四边形对边平行的特点，一端固定于车身，另一端固定于轮组，以此保持轮组和车身 yaw 轴的平行，也即和地面垂直。

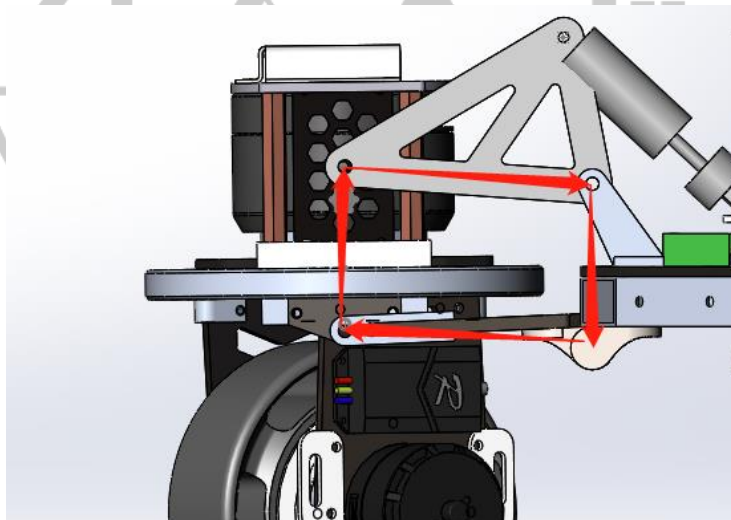


图 3-9: 20RM 悬挂设计

但由于今年的舵轮底盘新增一个救援机构，导致轮组整体拉长，轮组变形带来回转半径变大，而且还有把底盘防撞做圆的需求，导致使用平行四边形悬挂的难度增大。故今年决定采用最简单的独立悬挂形式，单摇臂独立悬挂，因其结构较为简单，占用空间较小，设计上难度较小，能满足舵轮底盘经过起伏路段和下台阶的需求。

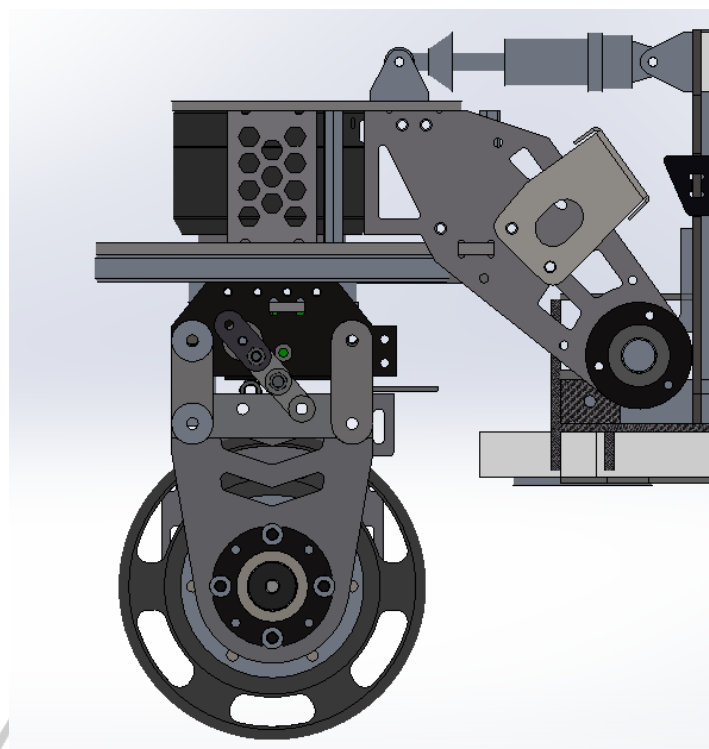


图 3-10: 21RM 悬挂设计 (1)

一方面，为了减小单摇臂独立悬挂对于舵轮底盘轮距改变的影响，设计者使避震器压缩量为 0 时保证每个轮组 yaw 轴与云台 yaw 轴垂直（即与地面垂直），保证舵轮底盘在平地上的性能最大化，悬挂只在经过起伏路段和下台阶时起到作用；另一方面，为了避免单摇臂独立悬挂在受力时出现 roll 轴上的偏转，两块转动板之间进行了固连，且连接在同一根转轴上，一定程度减弱了轮组 roll 轴上的偏转，保证了悬挂的稳定性。

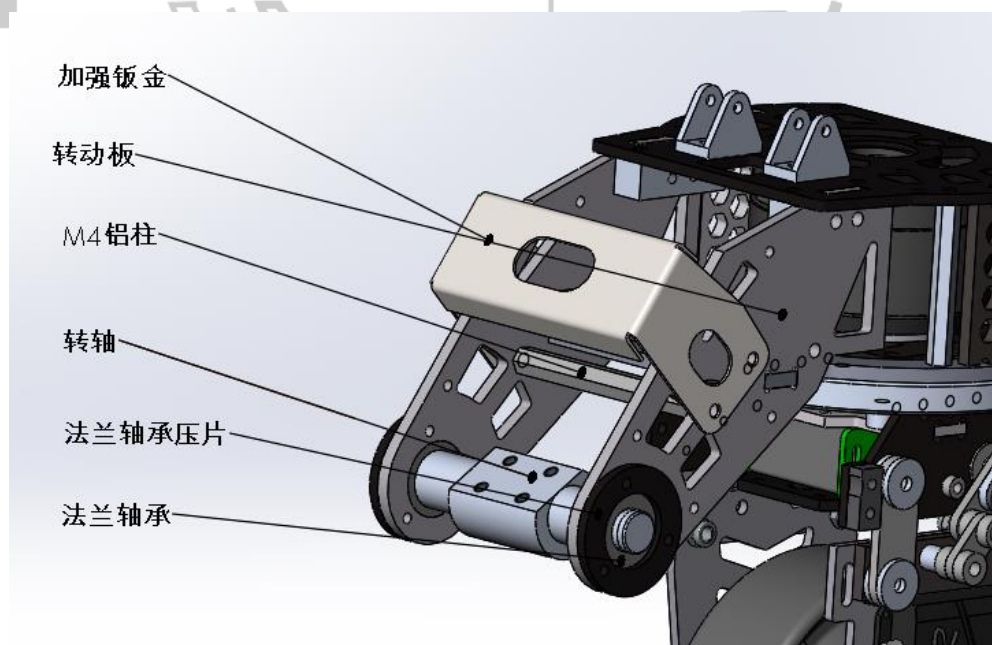


图 3-11: 21RM 悬挂设计 (2)

3.2 云台设计方案

舵轮步兵云台采用较为成熟的上供弹方案，管路短且发弹延迟低。Yaw 轴传动上采用深沟球 6817 轴承。用深沟球轴承传动的方式与轮组采用的餐桌轴承相比虚位更小，使用寿命更长。使用时通过自铣板件压紧分别压紧内外圈可以消除轴承内部游隙和给滚动体施加预载荷有提高轴承的转动性能、刚度和使用寿命，使云台运动响应更好，晃动更小。

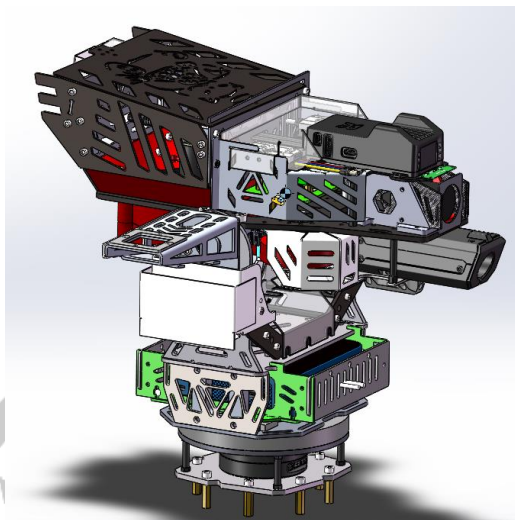


图 3-12：云台结构展示

3.3 17mm 小弹丸发射机构设计方案

3.3.1 拨盘部分

拨盘采用 2006 电机上置带动单层拨轮推弹，其结构如下：

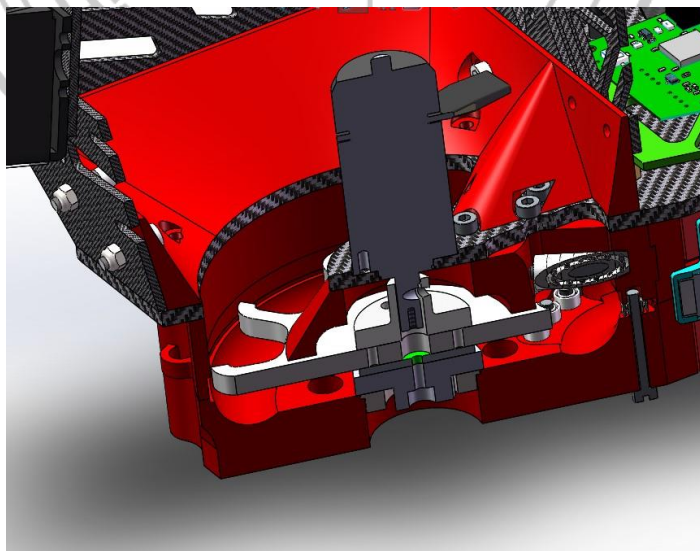


图 3-13：拨盘结构

结构说明：

在该结构中，一个拨轮有 10 个拨齿。在拨盘出口处有两种微型轴承来避免卡弹：一种通过尼龙螺钉和胶枪固定来修正出口处轨迹，另一种通过长螺钉攻进

打印件结合胶枪固定来防止出口处弹丸跳动带来卡弹。

存在的问题：

电机上置容易带来架桥卡弹的问题，且转动时容易不同轴。拨盘内径是 95，整个拨盘较大，给云台布线和运动带来不便。目前在射频 15 时最稳定，当射频增高时会有卡弹风险。卡在拨盘口最常见，同时也有前面发射顶针安装误差导致的卡弹。

3.3.2 发射部分

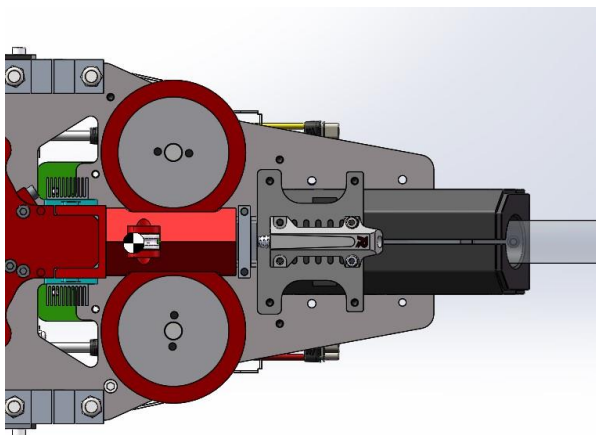


图 3-14：发射结构

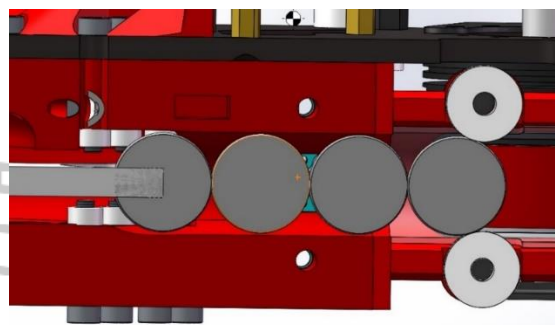


图 3-15：管路结构

1) 电机部分：

使用去掉减速箱的 3508 电机带动摩擦轮，包胶材料聚氨酯硬度 50，内径 48 外径 60，一般常温高速发射 2000 颗弹丸需要更换，内径 46 外径 60 时常温高速发射 1000 颗弹丸就需要更换，且与内径 46 相比高速时弹丸出膛速度下降。3508 比较大且沉，后续计划向轻便发展，测试过 snail，但目前 snail 高速时散布还比较大。目前结构是电机和枪管没有直接固定在一个板子上，容易有装配误差带来发射效果不好的问题，后面要把电机和枪管直接固定在一个板子上。

2) 顶针部分：

使用的是两颗 U 型槽轴承做顶针，发现在发射过程中会有弹丸在顶针处卡，且会出现弹丸在顶针和摩擦轮之间滚动的情况导致这颗弹丸掉速严重。两颗轴承效果比一颗好一些。

3) 枪管部分：

使用长枪管，内径 19 外径 21，目前测试效果是长枪管效果比无枪管效果好，内径 19 比内径 17.7 效果好，因此采用当前版本。

4) 管路部分：

拨轮末端到顶针和摩擦轮相切位置时距离有 3.38 颗弹，漏弹改率较小但仍存在。

4. 机器人电路设计

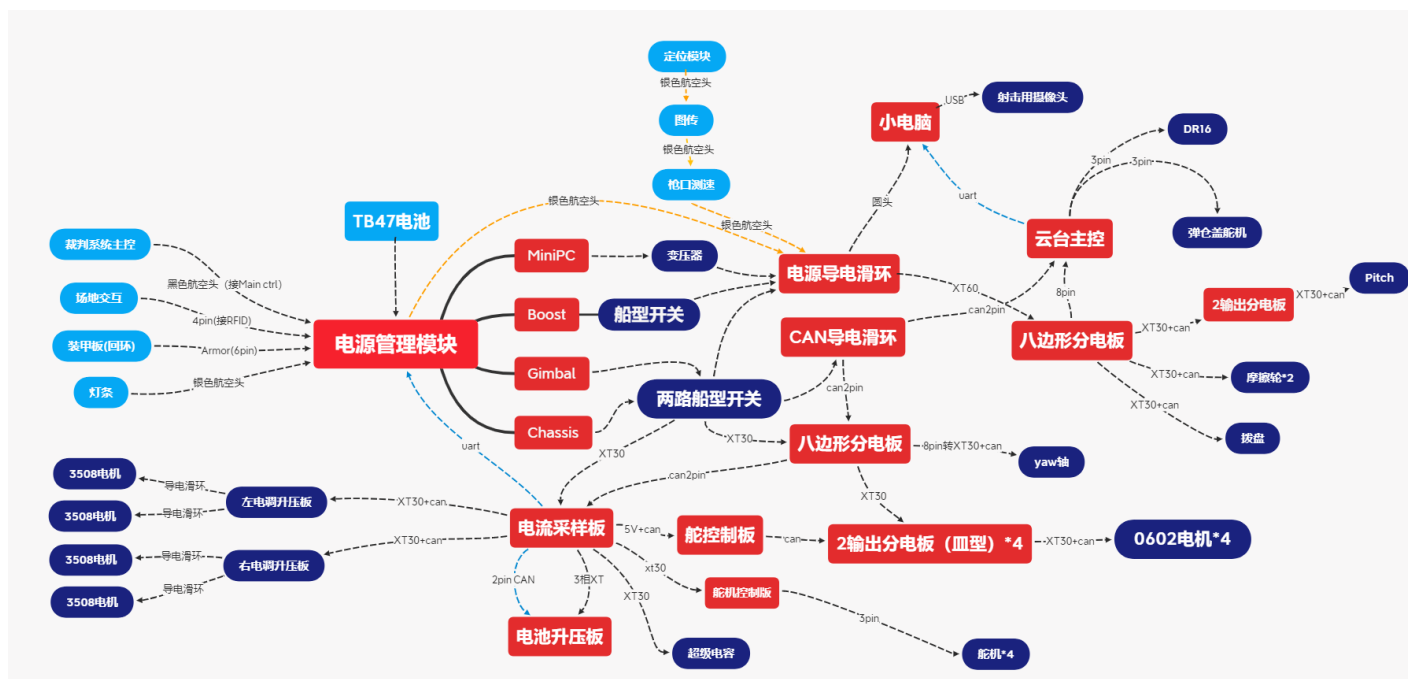


图 4-1：舵轮步兵机器人硬件拓扑图

机器人硬件拓扑图可简化分为底盘和云台两部分，其中的板间通信为 CAN 通信，通过导电滑环完成底盘与云台的通信连接。

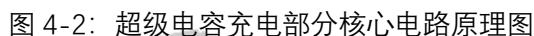
下面主要讲一下底盘部分硬件设计。

连接底盘硬件部分主要由底盘功率控制主控板和超级电容模块构成，主要功能为机器人底盘运动控制与裁判系统相关功率控制。底盘部分负载为四个 M3508 电机，是功率消耗的主要来源。

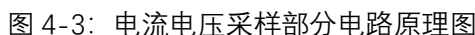
4.1 底盘功率控制电路板设计

4.1.1 超级电容充电部分

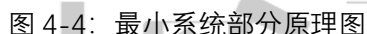
采用电池管理芯片 BQ24640 进行恒流充电控制，充电电流可以通过 ISET 脚进行设定。超级电容电压硬件部分通过 VFB 脚进行输出电压采样，当电容电压大于等于截止电压时硬件关闭输出，另外通过 10mR 的采样电阻与 SRP、SPN 脚进行电流监控，实现恒流充电的控制。考虑到底盘工作情况比较复杂，负载的变化通常很大，因此采用了较大的输入输出电容进行滤波。为了方便调试，增加指示灯 STAT 和 PG，显示充电或工作是否正常。软件上通过 CE 脚控制是否使能充电功能，其余功能均由硬件完成。



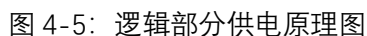
电流采样使用 2mR 的 2512 封装合金电阻，采样芯片使用 AD8628，对各部分电流放大 100 倍或 50 倍进行 ADC 采样；电压采样使用分压电阻分压，将目标电压幅值降为原来的 1/11，输入 MCU 进行采样。计算时直接使用采样的电压和电流相乘得到各部分的功率，进行闭环控制。



最小系统 MCU 为 STM32F405RGT6, 包括复位电路、晶振电路和基本的额外指示灯, 引出串口、CAN、TF 卡等外设。



逻辑部分使用 K7805-500 模块降压到 5V 再通过 SPX3819 稳压到 3.3V 给逻辑部分设备供电，供电输入端有必要增加 TVS 以抑制尖峰，保护后面的器件。K7805 模块最大支持 5V/0.5A 的输出，由于底盘负载变化大，逻辑部分供电要求高，因此增大输入输出电容以提高电源稳定性。



4.1.5 输出开关控制部分

底盘功率控制板支持电池直接供电或电容供电，通过两组背对背 MOS 进行功率回路切换，控制芯片使用 BQ76200，控制信号由单片机给出。

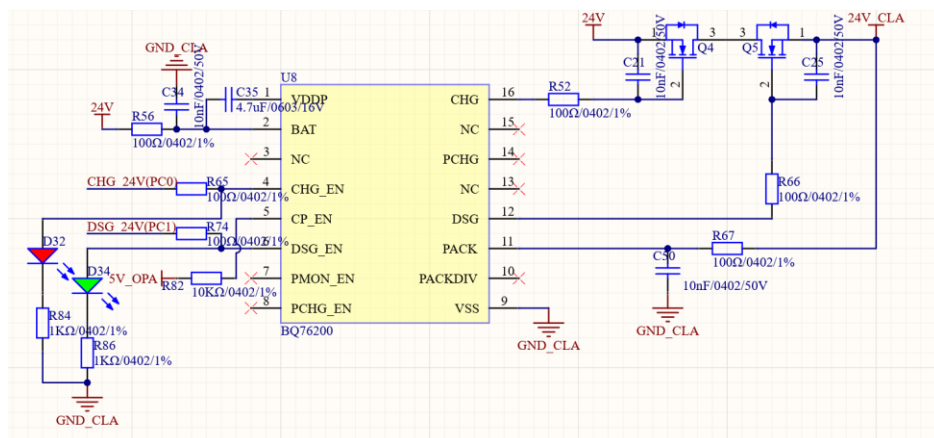


图 4-6：输出开关部分原理图

4.1.6 超级电容充放电控制策略

主控芯片通过电压采样方式获取超级电容实时电压（设计超级电容模块工作电压区间为 12V~24V），并以此为依据控制超级电容模块的充电电流，主控芯片通过 ISET 脚控制 BQ24640 芯片以控制超级电容模块充电电流，当充电电流为 0 时即为停止充电。机器人供电策略可简化为两种供电模式：电池供电模式和超级电容供电模式。两种模式之间的切换由主控芯片配合两块 BQ76200 芯片完成。

在电池供电模式下，主控芯片配合电池端 BQ76200 芯片控制电池高侧 MOS 管按序导通，超级电容端 BQ76200 芯片控制电容端高侧 MOS 管按序关断，形成电池供电回路，由电池对机器人进行供电，同时主控芯片通过信号使能控制 BQ24640 芯片对超级电容模块进行充电，直至电压达到 24V。

在超级电容供电模式下，主控芯片配合超级电容端 BQ76200 芯片控制电容高侧 MOS 管按序导通，电池端 BQ76200 芯片控制电池端高侧 MOS 管按序关断，形成超级电容供电回路，此时主控芯片通过 ISET 脚控制 BQ24640 芯片停止对超级电容模块充电，机器人由超级电容模块进行供电，同时，主控芯片通过电压采样实时检测超级电容模块电压，当电压降低到 12V 时停止使用，并切换为电池供电模式。

4.2 超级电容模块

超级电容模块采用九个 2.7V 50F 法拉电容串联，通过电容储能公式 $E=1/2CU^2$ 粗略计算电容储存总能量为 $9*(2.7*2.7*50/2)=1640.25J$ ，设计工作电压区间为 12V~24V。以模块工作电压区间为 12V~24V 粗略计算，可使用能量为 $9*(2.7*2.7*50/2)-9*(1.3*1.3*50/2)=1260J$ 。考虑总体效率约为 85%，实际使用能量为 $1260*0.85=1071J$ 。考虑电容充电功率恒定且由裁判系统功率决定，输出功率按照 200W，在 80W 限制下需要超级电容额外提供 120W 的功率，则可以输出的时间约为 9s。

4.3 电池前端升压板

为了解决电池低电量时电压较低导致性能下降的问题，同时需要解决超级电容充电电压比电池电压低的问题，研发了电池升压电路板。该电路板位于超级电容充电输出的前级，为超级电容充电的电源进行升压以保证超级电容能达到比较高的电压（如 24V）。使用 TI 的一款升压芯片 TPS43060 作为核心升压电路的控制芯片，同步整流拓扑以获得更高的效率。升压输出的电压根据 VFB 确定，设定的输出电压约为 24V，完全由硬件控制。输出端增加 TVS 以抑制尖峰。逻辑控制为 STM32F103C8T6 最小系统，引出 CAN、串口等外设进行通信和调试。

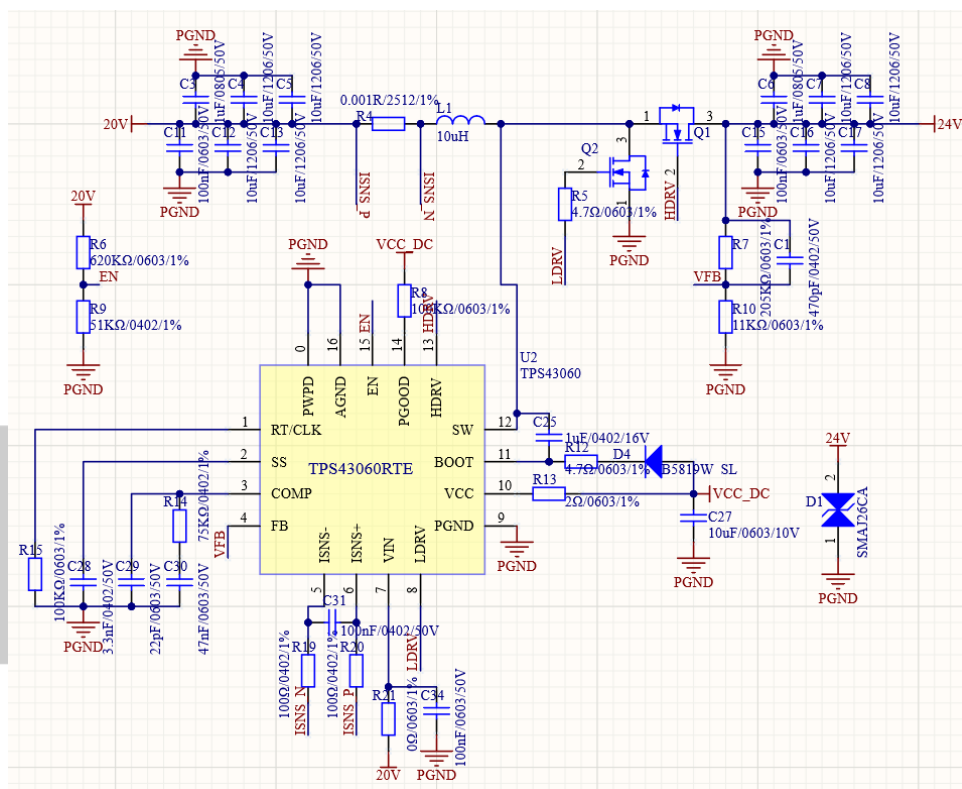


图 4-7：电池前端升压部分电路原理图

4.4 电调前端升压板

由于硬件方案通常为电池为电容充电，电容为电调供电，而电容的电压波动较大，需要电调前端的升压恒压输出，保证负载有稳定的电压。核心电路与电池升压板相同，也使用 TPS43060 作为控制芯片。由于单路输出电流有限，且四个电机实际负载不同，设计上将两路电调的升压电路集成在同一个板上，而车上需要有两个电调前端升压板。硬件上可以通过更改 VFB 的两个电阻 R61、R67 来调节升压的电压，通常根据各兵种不同设置几个默认电压档位，为了输出电压的可调，支持软件更改电压。

此外，升压板的逻辑控制为 STM32F103C8T6 最小系统，通过两个 IO 控制 PMOS 以对是否使能升压进行控制，MCU 外围引出 CAN、串口、散热风扇等外设。

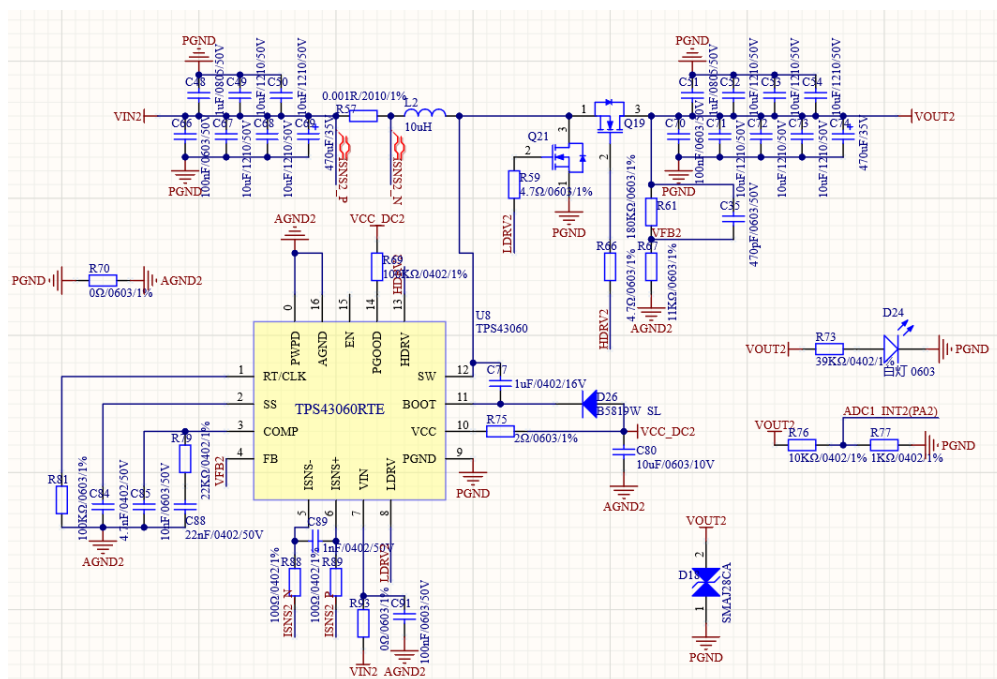


图 4-8：电调前端升压板电路原理图（1）

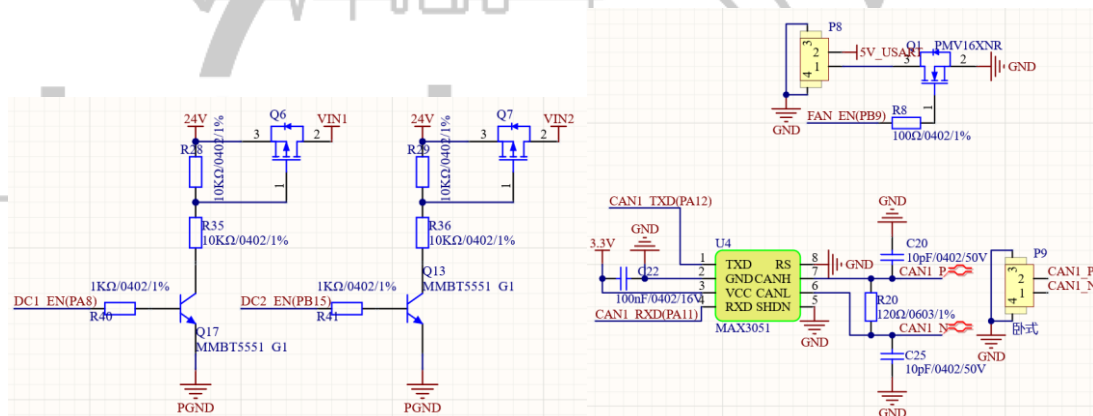


图 4-9：电调前端升压板电路原理图（2）

5. 机器人嵌入式控制

5.1 程序主要理论

舵轮底盘解算。底盘速度矢量的分解如下图所示：

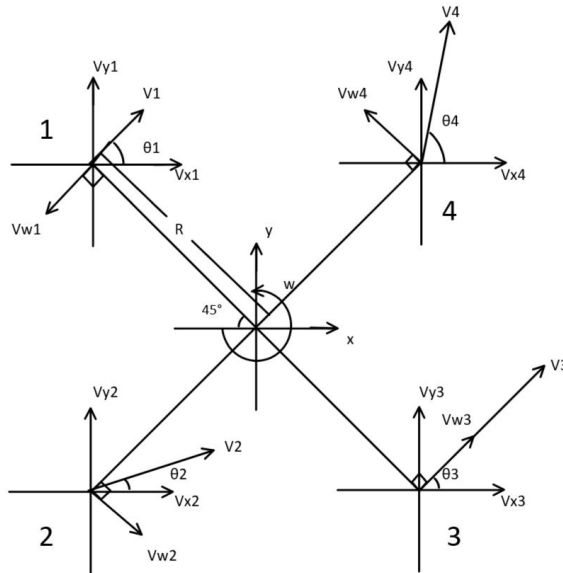


图 5-1：底盘速度矢量分解图

先假设车体运动的前进速度为 1，平移速度为 1，旋转速度为 1
此时可由车体 3 轴速度得到四个轮组的 3 轴速度：

$$V_{x1} = V_{x2} = V_{x3} = V_{x4} = 1$$

$$V_{y1} = V_{y2} = V_{y3} = V_{y4} = 1$$

$$V_{w1} = V_{w2} = V_{w3} = V_{w4} = 1 \times R = R$$

则由各轮组分速度合成后得到合速度的大小与方向：

$$V_1 = \sqrt{(V_{y1} - V_{w1} \cos 45^\circ)^2 + (V_{x1} - V_{w1} \sin 45^\circ)^2}$$

$$V_2 = \sqrt{(V_{y2} - V_{w2} \cos 45^\circ)^2 + (V_{x2} + V_{w2} \sin 45^\circ)^2}$$

$$V_3 = \sqrt{(V_{y3} + V_{w3} \cos 45^\circ)^2 + (V_{x3} + V_{w3} \sin 45^\circ)^2}$$

$$V_4 = \sqrt{(V_{y4} + V_{w4} \cos 45^\circ)^2 + (V_{x4} - V_{w4} \sin 45^\circ)^2}$$

$$\theta_1 = \tan^{-1} \frac{(V_{y1} - V_{w1} \cos 45^\circ)}{(V_{x1} - V_{w1} \sin 45^\circ)}$$

$$\theta_2 = \tan^{-1} \frac{(V_{y2} - V_{w2} \cos 45^\circ)}{(V_{x2} + V_{w2} \sin 45^\circ)}$$

$$\theta_3 = \tan^{-1} \frac{(V_{y3} + V_{w3} \cos 45^\circ)}{(V_{x3} + V_{w3} \sin 45^\circ)}$$

$$\theta_4 = \tan^{-1} \frac{(V_{y4} + V_{w4} \cos 45^\circ)}{(V_{x4} - V_{w4} \sin 45^\circ)}$$

5.2 嵌入式算法说明

底盘舵部分使用双环控制，外环为角度环，内环为速度环，其中角度目标值为上述解算出的值。

底盘轮部分使用单环控制，输入为轮组速度目标值，同时功率控制部分会给出一个倍率乘以 pid 输出作为轮输出。

云台部分 pitch、yaw 采用双环+前馈控制的方式，加快云台相应速度。其中前馈量是通过拟合后得到的曲线，在 pitch 轴上的可活动区间内每隔一定角度就测量一次输出值，之后将输出值进行拟合得到前馈函数。在操作手模式下，各轴外环目标值为操作手移动鼠标后得出的目标值，在自瞄模式下，各轴外环目标值为视觉传来的打击装甲板所需要的偏差值+当前值。

发射部分摩擦轮采用单环控制，输入为当前射速下摩擦轮目标转速，拨盘部分采用双环控制，每发射一颗弹拨盘角度目标值增加一个弹位的角度。

救援部分由四个舵机组成，在底盘断电前控制舵机的角度使轮子与地面接触点在轮组舵的旋转轴上，断电后使其偏心，以达到救援的目的。

5.3 代码设计介绍

使用 STM32CubeMX 新建 Keil 工程,之后加入实验室的库作为模板工程，其中实验室的库对于 CAN、Uart、I2C、SPI 等通信协议以及 Freertos、TIM 等需要的函数进行了模块化设计并统一了接口，以方便项目开发，减少开发时间。

代码中，各统一初始化放在了 System_Config.cpp 中，freertos 任务统一放在了 Service_Devices.cpp 中，底层的变量放在了 System_DataPool.cpp 中，与通信相关的任务及配置放在了 Service_Communication.cpp，云台、底盘不共有的部分各自放在了 pitch_yaw.cpp，auto_infantry.cpp 等文件中。

6. 机器人视觉算法

6.1 代码框架

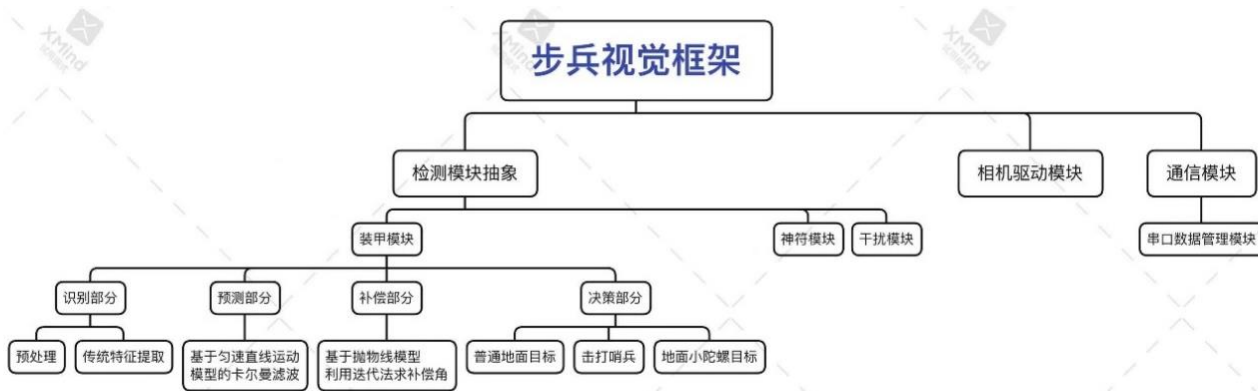


图 6-1：舵轮步兵视觉代码框架

分别有装甲模块（检测敌方装甲板）、神符模块（识别己方能量机关）、干扰模块（识别敌方已激活的能量机关并进行干扰）三个派生类继承于检测模块基类，并根据通信模块中视觉接收端数据进行判断是否切换检测模块。每个检测模块中均含有四个部分：识别、补偿、预测、决策。上图只在装甲模块中展出。

目前框架中各个模块之间耦合性还较大，在下一赛季中拟采用抽象工厂设计模式将每个模块的每个部分抽象出来。

6.2 装甲检测模块

- 1) 识别部分：采用的是传统特征提取的方式，通过对轮廓的长宽比、倾角、PnP 解算信息……进行筛选和匹配，匹配成功后将 `push_back` 进其专属的 `deque` 时间序列中，并用一个 `vector` 向量存储所有的 `deque` 时间序列。
- 2) 补偿部分：`vector<deque<>>` 时间序列将在此被分离成彼此独立的 `deque`，并根据通信模块中视觉接收端得到的 `pitch`、`yaw` 角度进行补偿角度的计算。

```

/**
 * @brief 获取补偿值以及飞行时间
 *
 * @param _gyroData 陀螺仪参数
 */
void ArmorTracker::calculateCompensate(GyroData &_gyroData)
{
    // 实例化补偿对象
    Compensate com;
    // 计算补偿值
    float angle = _gyroData.pitch - this->getGoalArmor()->relative_angles.y;
    float x_com = compensate_param.YAW_COMPENSATE;
    float y_com = com(this->pnnp_data.distance / 1000.f,
                     angle + compensate_param.PITCH_COMPENSATE,
                     (float)_gyroData.shootSpeed) -
                     angle;
    // 获取飞行时间
    this->flying_time = com.time;
    // 获取 pitch、yaw 补偿值
    this->xy_compensate = Point2f(x_com, y_com);
}

```

图 6-2：补偿值及飞行时间的函数实现

经过比对，发现 10m 距离 15m/s 的 17mm 小弹丸在考虑空气阻力和没有考虑空气阻力的两个模型下飞行时间相差在 1ms 左右，可忽略不计，因此我们采用了更简单的抛物线模型。

- 3) 预测部分：所有的 deque 时间序列将传入预测器单独考虑，每一个 deque 时间序列不得超过规定的最大长度，在长度满足要求后便可以进行预测。预测模型采用匀速直线模型，即“ $x=x_0+vt$ ”
 - [1] x_0 为当前相机捕获的位置信息
 - [2] v 为融合了相机中捕获的角速度和陀螺仪角速度后的真实角速度（此角速度为相对于底盘坐标系下的角速度）
 - [3] t 为补偿模型中得到的子弹飞行时间

此公式作为卡尔曼滤波的状态转移矩阵： $A = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 进行预测与平滑

处理。

- 4) 决策部分：此部分决定了云台响应以及发弹指令的给出。
 - [1] 普通模式下将按照正常的功能运行
 - [2] 此外在普通模式下还随时监视着敌方小陀螺的状态，如果连续多帧有高速运动的倾斜装甲板，则被判断为小陀螺模式。在小陀螺模式下有着不一样的响应方案。防止云台响应过大，避免给操作手不良的体验，因此在击打小陀螺模式中不进行预测，但在装甲板将要消失之时及时切换至已出现的另一块装甲板或灯条。但在发弹指令的决策中仍然需要考虑预测信息，为了能精确击打中目标，发弹延迟需要控制在 70ms 左右，云台响应要控制在 50ms 以下。按照此方法，击打 80W 舵轮步兵的小陀螺命中率在 60%以上。
 - [3] 哨兵模式下由于哨兵运动路线较为单一，因此给了较高的掉帧保留帧

数。为保证稳定性，滤波器的测量噪声协方差系数给的比普通模式下的稍大。

6.3 神符模块/干扰模块

- 1) 识别部分：逻辑与装甲模块一致，通过轮廓父子关系筛选，神符模块只寻找未激活的部分，而干扰模块则识别已激活的部分。
- 2) 补偿部分：与装甲模块一致
- 3) 预测部分：与装甲模块基本一致，公式改为角速度： $\theta = \theta_0 + \omega t$
- 4) 决策部分：为装甲模块的普通模式，发弹频率由电控端控制

7. 人机交互设计

7.1 自定义 UI 界面

自定义 UI 界面展示如下图所示：



图 7-1：操作手 UI 界面

各 UI 单元完成的功能如下：

- 1) 射击准星上下的横线为射击辅助瞄准线。
- 2) 左侧的 EVENT 为雷达站发送的发生事件，SUGGESTION 为雷达站发送的战术建议。
- 3) 左侧的 CAP_ENERGY 为超级电容的可用容量百分比，0-30%时为红色，30-70%为黄色，70-100%为绿色。
- 4) 下方中间位置的黄色矩形为车间线，用于给操作手判断车身宽度，减少撞击。

7.2 状态指示灯

在图传模块前方增设状态指示灯，可以实时向操作手反馈步兵机器人当前各功能模块的状态，以及所处模式；同时状态指示灯也方便电控调试、以及检修人员上场检修时快速定位问题。

状态指示灯各灯珠颜色代表意义如下表所示：

1	2	3	4	5	6
高速模式指示灯	超级电容状态指示灯	电容电压状态指示灯	弹仓盖状态指示灯	视觉目标状态指示灯	遥控状态指示灯
绿色：正常模式 白色：高速模式	绿色：正常模式 白色：超级电容模式 蓝色：上坡（加力）模式	白色：电容电压充足 绿色：电容电压较高 黄色：电容电压较低 红色：电容电压危险	绿色：弹仓关闭 红色：弹仓打开	绿色：打击地面目标 白色：打击哨兵 蓝色：打符 红色：与视觉断连	绿色：操作手模式 蓝色：遥控器模式 红色：遥控掉线

图 7-2：状态指示灯灯珠状态指示表

7.3 操作手便捷按键（半自动化设计）

增加一键掉头按键，底盘的前和后均能作为车体的前方。在前往兑换弹药区兑换完子弹后能迅速掉头，在上我方环形高地后也能迅速转过来来击打前哨站，增加了舵轮步兵机器人的机动性。

增加低速模式按键，操作手可在补弹时进入低速模式，进而对车体姿态进行微调，对准落弹口。

8. 开发工具

Keil、STM32CubeMX、Visual Studio

9. 测试

9.1 小陀螺转速测试

舵轮步兵开启小陀螺模式，对小陀螺转速进行测试，各功率挡位下的测试结果记录如下：

表 9-1：小陀螺转速测试表

功率	转子转速（转）	转速（圈/秒）	转速（m/s）
140W	8100	2.45	3.27
120W	7600	2.30	3.07
100W	6900	2.09	2.79
80W	5800	1.76	2.34
60W	4600	1.39	1.85
40W	3400	1.03	1.37

9.2 下台阶、过盲道测试

舵轮步兵跑场，进行下台阶和盲道地形测试，机构无损伤。部分测试视频截图如下图所示：



图 9-1：下台阶测试视频截图（1）



图 9-2：下台阶测试视频截图（2）