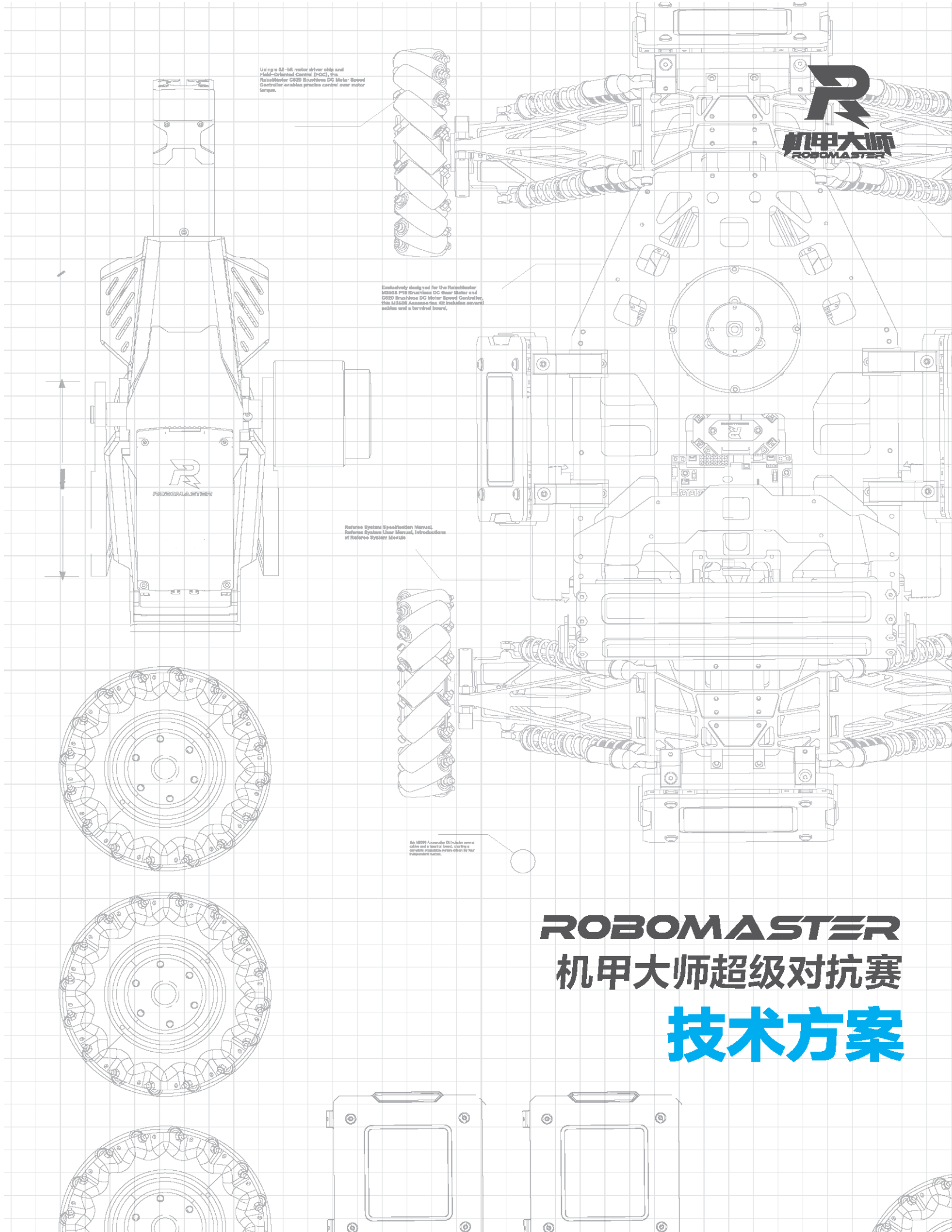

Using a 32-bit motor driver chip and Field-Oriented Control (FOC), the RoboMaster C620 Brushless DC Motor Speed Controller enables precise control over motor torque.
Exclusively designed for the RoboMaster M3508 P19 Brushless DC Gear Motor and C620 Brushless DC Motor Speed Controller, the M3508 Accessories Kit includes several cables and a terminal board.
Referee System Specification Manual, Referee System User Manual, Introductions of Referee System Module
ROBOMASTER
机甲大师超级对抗赛
技术方案

前言

本技术报告由辽宁科技大学 COD 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	王志强	
硬件	涂仁杰	
软件	王芃	
算法	王柏程	朱云飞
其他	赵欣	

目录

1. 概述

1.1 背景&目标

在新赛季等级体系下，步兵机器人等级上限提高到 10 级，且平衡步兵相较于普通步兵可额外获取 50%经验，这意味着平衡步兵在前中期将具有高等级带来的属性压制，此外平衡步兵具有特殊的装甲板排布，侧向对敌时可以更好的规避对方火力，是打破战场僵局，推进战术执行的中坚力量。随着场地拓扑结构的修改，具有跳跃能力的轮腿式平衡步兵将更具优势，为战场战术选择带来更多的可能性。

因此我队本赛季的研发重点转向具有多姿态自救能力、稳定飞坡、任意上下台阶和强自瞄能力的轮腿式平衡步兵。

1.2 其它学校机器人分析综述

表 1 轮腿式平衡步兵发展历程

年份	战队	发展历程
2021 年	哈尔滨理工大学(荣成)SPARK	首次开源轮腿式平衡步兵
2022 年	哈尔滨工程大学创梦之翼	侧向对敌、跳跃台阶、站立攻击
2023 年	大连交通大学 TOE 战队	稳定连续飞坡
2023 年	上海交通大学交龙战队	强自瞄能力
2024 年	西南石油大学铁人战队	特殊姿态对敌

图 1-1 哈理工（荣成）

图 1-2 哈工程

图 1-3 西南石油

根据上述优秀的案例来看：一台优秀的轮腿式平衡步兵机器人需要拥有高稳定性、高机动性、全地形适应性，并具有强自瞄能力等特点。

1.3 机器人功能定义

表 2 功能定义

功能	需求分析	设计思路
高机动性	底盘轻量化设计	仿真计算减少结构冗余，合理缩减尺寸
	可稳定上下 15cm 台阶	
精准打击	距离 5m 击打小装甲命中率 100%	射速、射频稳定维持在官方设定上限-5%以内
	射频 20Hz 以上，弹速稳定不卡弹	改善拨弹盘与枪管连接处的机械结构
功率控制	超级电容辅助控制功率	双向升降压超级电容做到实时功率充

		放
	针对不同功率上限的精准功率控制	推进功率控制板的研发及优化
	对电机建模精准控制功率	通过电机电流计算实时功率，根据当前功率限制计算电机所需功率的力矩
强自瞄能力	5 米范围稳定锁定敌方装甲板	利用特征判据识别灯条并匹配成对，经三层神经网络识别数字，参与评分机制锁定目标，经PNP 后转换坐标系进行瞄准
	弹道适配与参数解析	重新建立抛体运动模型，利用迭代法求解发射角度，以查表偏置调整具体弹道
	运动解析与位置预测	利用扩展卡尔曼滤波拟合观察数据，预测整体位置变化，解算未来时刻装甲位置
能量机关	高效识别能量机关	利用关键点检测算法得到部分标记点，使用坐标旋转算法补全关键点，经 PNP 测距和弹道解算存入向量

1.4 机器人核心参数

1.4.1 机械参数

表 3 机械基本参数

名称	参数
尺寸	590.40MM * 494.79MM * 515MM
重量	21.56KG

1.4.2 执行器件

表 4 执行器件

执行元件	用途	数量（个）
3508 电机	为弹丸提供初始动能	2
2006 电机	为拨弹盘提供动力源	1
6020 电机	驱使 Pitch 轴转动	1
达妙 4310 电机	驱使云台周转运动	1
达妙 8009 电机	关节驱动	4
领控 9025 电机	驱动轮	2

1.4.3 主要传感器参数

表 5 主要传感器参数

名称	型号	参数	数量（个）
海康威视图像摄像头	CA013	分辨率 1440*1080，帧率 120	1
IMU	BMI088	-	1

1.4.4 性能参数

表 6 性能参数

名称	参数
最大移动速度	2.5m/s
最大爬坡角度	25°
云台自由度	Yaw 轴 360°，Pitch 轴俯仰角-26°~25°

1.4.5 电路参数

表 7 电路参数

名称	参数
电路功耗	10W
超级电容总容量	1950J
工作电压范围	18V~27V

2. 设计方案

2.1 需求分析

在 RoboMaster 赛事中，平衡步兵具有特殊的装甲板排布，侧向对敌时可以更好的规避对方火力，而在新赛季等级体系下，平衡步兵在前中期更容易建立高等级带来的属性优势。且随着场地拓扑结构的修改，采用轮腿构型的平衡步兵机器人在具有轮驱的高能效优点的同时能收获腿带来的良好地形适应性，相比于传统轮式倒立摆平衡步兵机器人，腿的加入使机器人机构获得了更多的自由度，可以极大程度提升平衡步兵在赛场中的表现。综上所述，我们选择了轮腿式平衡步兵方案。

2.2 机械结构设计

2.2.1 轻量化处理

为了提高平衡机器人的稳定性和机动性，重要的是要尽可能减少机器人的整体重量。因此，在设计机器人时，轻量化处理显得尤为关键。良好的轻量化设计不仅有助于减轻重量，还可以使机器人更加美观。通常情况下，更轻的重量意味着更快的加速性能、更灵敏的动态响应以及更高的效率。在赛事中，由于存在功率限制，优秀的轻量化设计尤为关键。

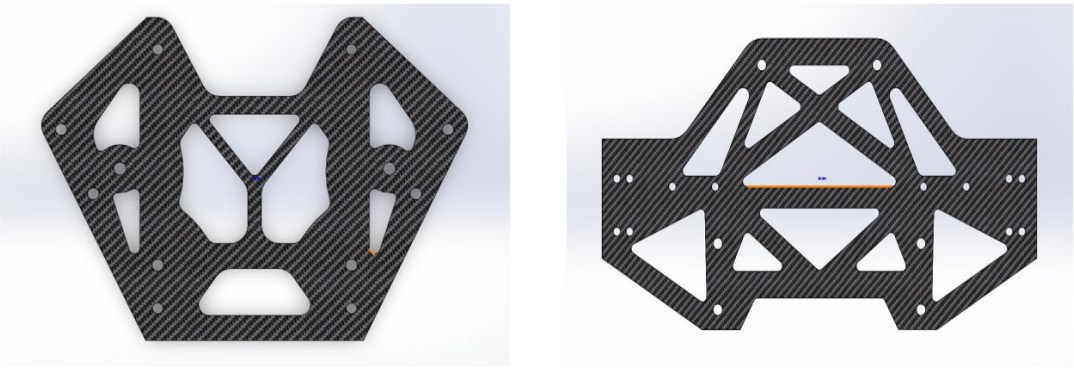


图 2-1 轻量化板材

2.2.2 防翻导轮设计

由于轮腿式平衡机器人本身结构限制以及场地复杂性影响，比赛时可能会出现翻车的情况，为了优化这一状况，我们采用了导轮来传导轮腿式机器人倒地时车架与地面的摩擦力。如下图所示，防翻导轮采用两块4MM玻纤板固定内6MM外30MM厚度10MM的包胶导轮，且考虑到硬接触可能会造成机器人零部件的损坏，所以我们选用材质为聚氨酯包胶的导轮，并在两块玻纤板中间加入了一块PLA材质的打印件来增加整体强度。

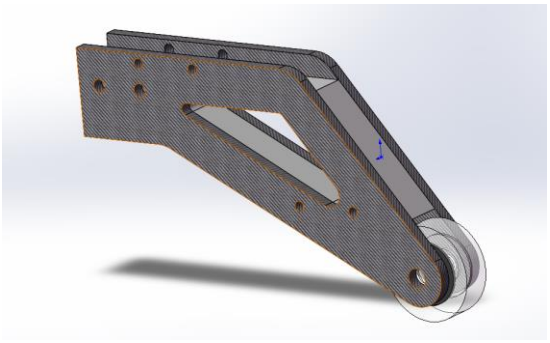


图 2-2 防翻导轮模块

2.2.3 快拆式底盘

为了保证关键线路以及控制板损坏时可以快速的维修，底盘采用了快拆式设计，

使用微型合页以及卡扣固定 C 型开发板以及场地交互模块的安装板，使其可以便捷的打开，同时降低机器人底盘电气布线的繁琐程度。

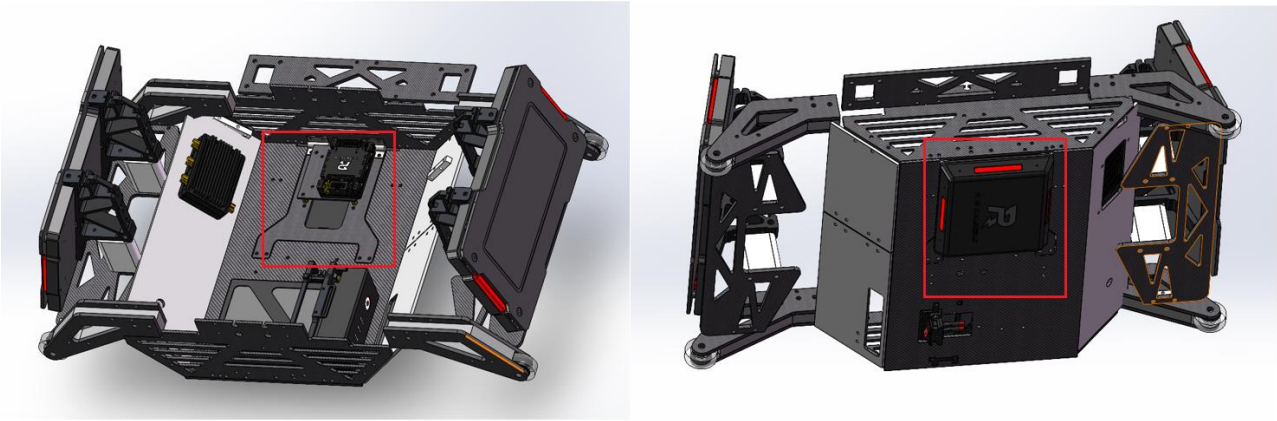


图 2-3 快拆式底盘

2.3 硬件设计

2.3.1 整机硬件方案框图

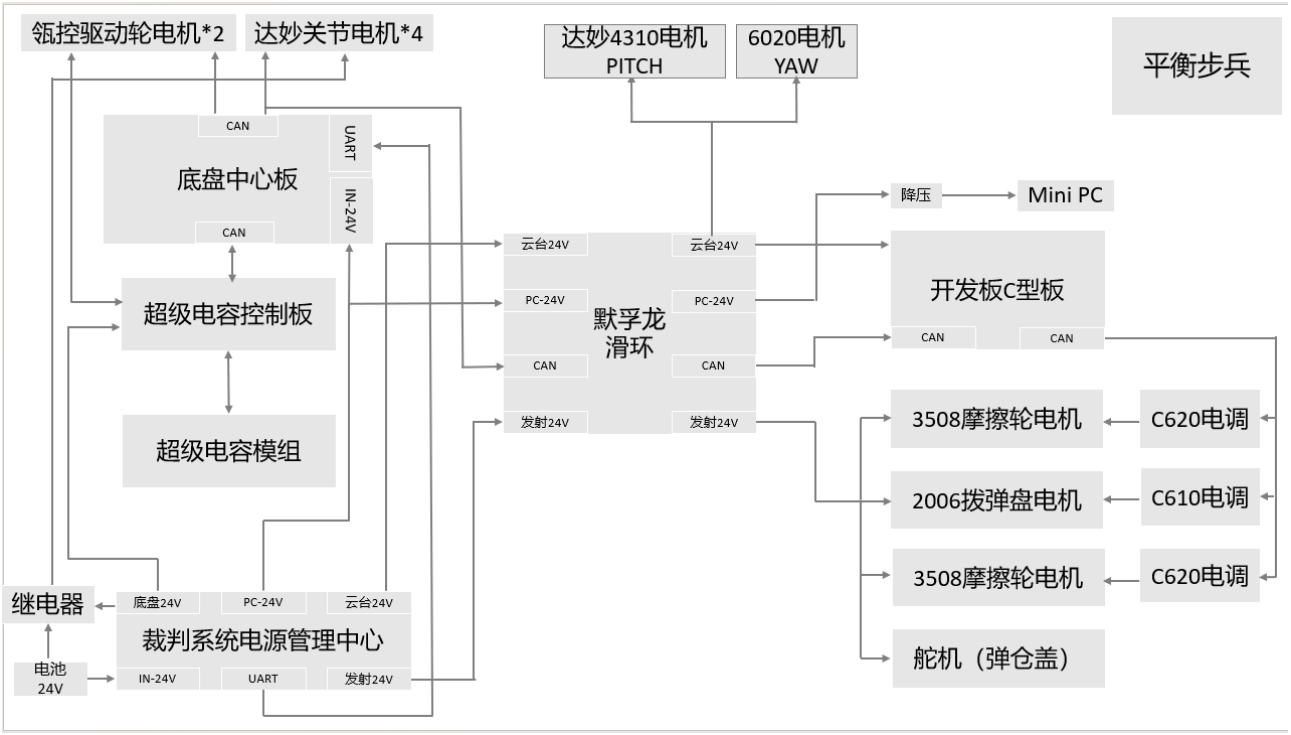


图 2-4 硬件连接框图

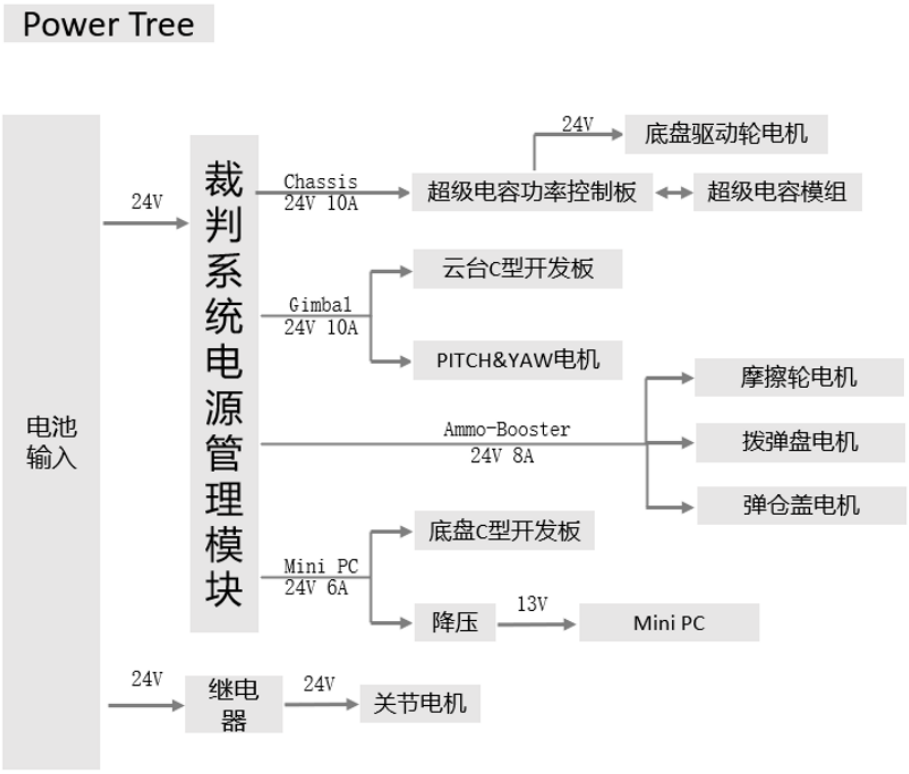


图 2-5 Power Tree

2.3.2 双向升降压超级电容

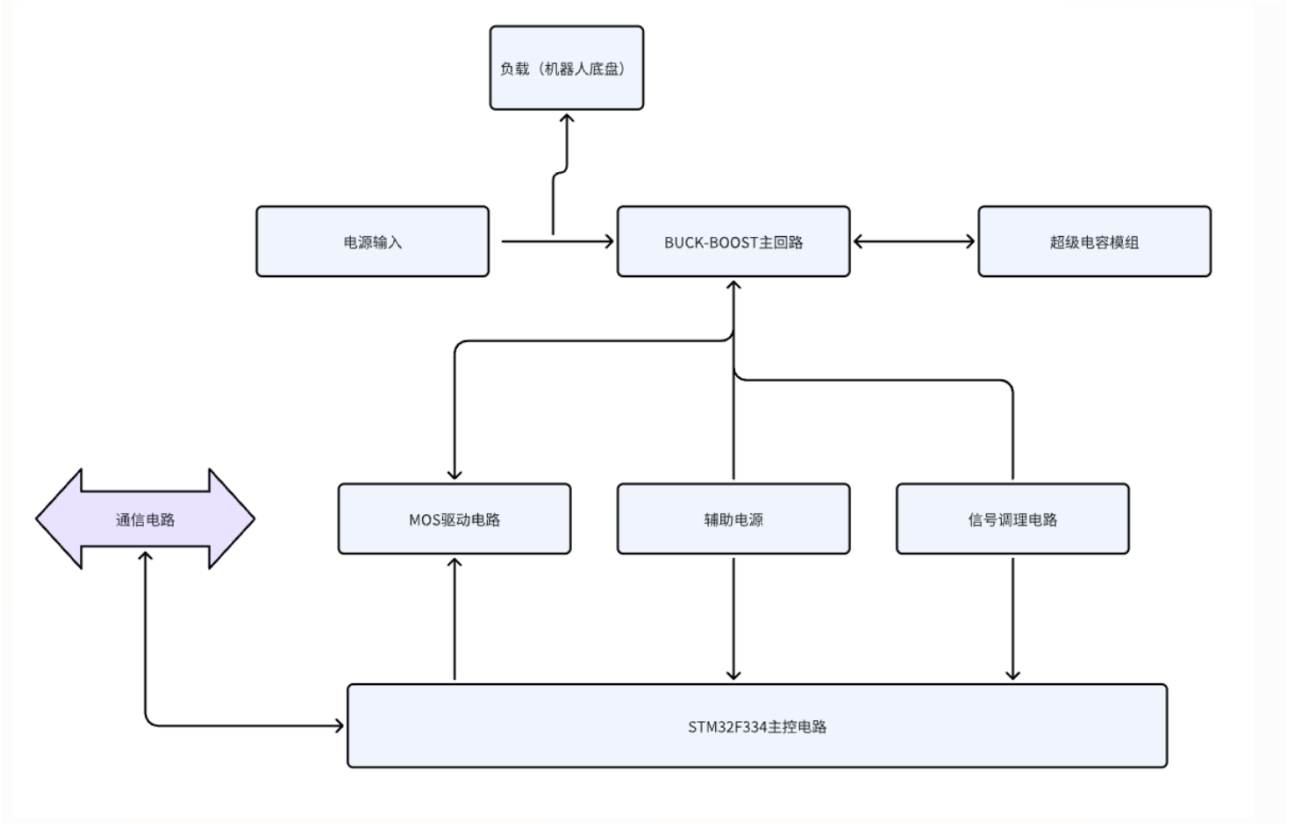


图 2-6 BUCK-BOOST 双向升降压超级电容控制器结构框图

在比赛过程中电源管理模块会对 Chassis(底盘)的输出功率进行限制，为了充分合理的利用底盘的功率，使用储能性能良好的法拉电容制作成电容模组对功率缺口进行补充，并制作超级电容功率控制模块对功率进行控制，以满足机器人在赛场上行动的各种功率需求。

底盘作为唯一负载连接在电池输入口以及 BUCK-BOOST 主回路正向输入口（也是反向输出口）。电源电流分两路进入底盘或流经 BUCK-BOOST 主回路给电容充电。电容充满后也可以反向提供电流经过 BUCK-BOOST 主回路给底盘进行供电。

超级电容模组作为储能池，负责将能量收集满，在需要的时候反向通过 BUCK-BOOST 主回路输出给底盘。

2.3.3 主要功能及指标

表 8 电气特性参数

电气特性	参数
输入电压	18~27V
电容电压	4~23V
输出电压	23~24V
DCDC 最大功率	150W
最大效率	92%
电源频率	280KHZ

2.3.4 主控电路

在主控的选型上, 主要考虑 STM32F3 系列以及 STM32G 系列有高级定时器可以输出高频率高精度的 PWM 波形，满足电源类控制实时性高的要求，其中 G 系列价格较高且开源资料中关于 STM32F3 系列的设计较多可以参考，基于采用 STM32F334C8T6 作为主控芯片，主控电路中除了主控芯片之外还需要必要的复位电路, boot 设置电路以及时钟电路, 上述三个部分是组成最小控制系统必不可少的。

2.3.5 信号调理电路

在运算放大器的选择上，电流检测采用 INA240 电流采样芯片，INA240 为双向检流芯片，支持检测双向的电流。我们选择 INA240A2，增益倍数为 50 倍，带宽和耐压也能够满足需求。

图 2-7 电流采样电路

电压检测采用 OPA2350UA/2K5 精密运放，其双通道单电源的特性大大节省了电压采样电路设计的成本和 PCB 布局空间。

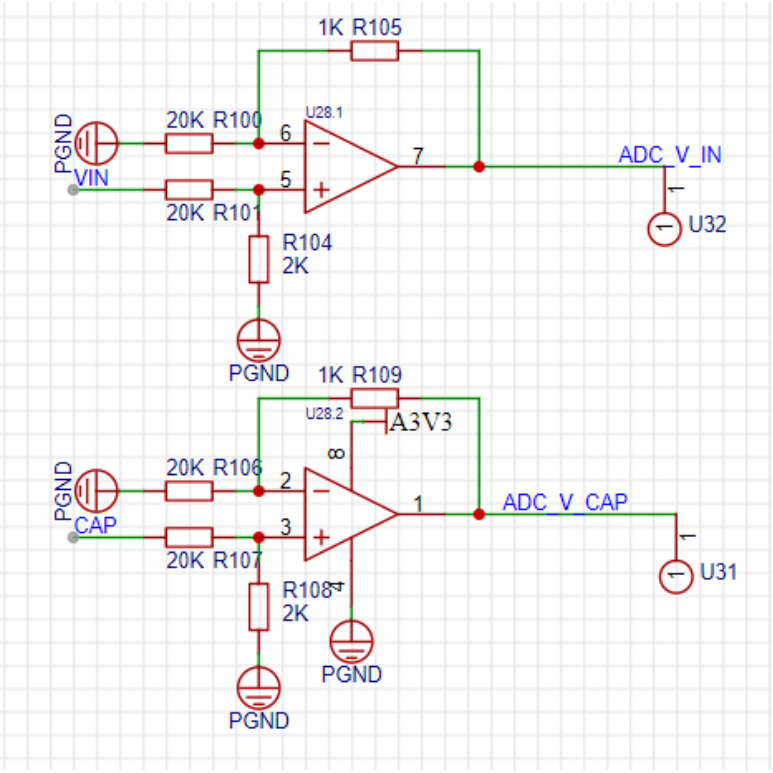
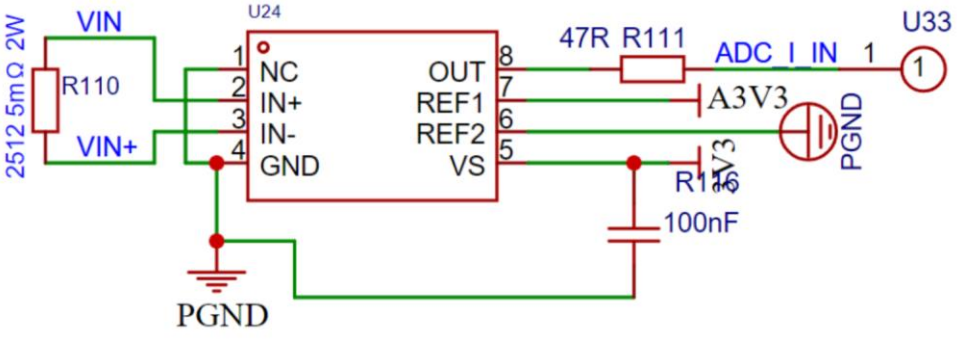


图 2-8 电压采样电路



2.3.6 驱动电路

MOS 驱动器的选型上，战队选择放弃使用 UCC27211，一方面 UCC27211 的价格常常差强人意，另一方面市面上购买到的芯片质量良莠不齐，综合考虑成本和稳定性我们选择使用具有独立的高侧和低侧驱动的半桥驱动芯片 SLM27211。

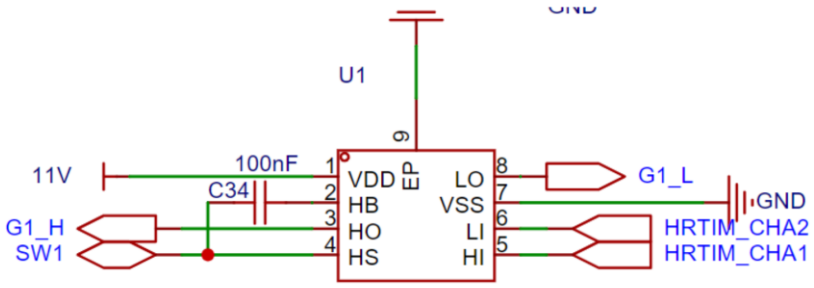


图 2-9 MOS 驱动电路

2.3.7 关键器件选型

表 9 关键器件选型一览

名称	选型
主控电路	采用 STM32F3 系列
电流电压检测芯片	INA240A2 芯片
MOS 驱动	采用具有独立的高侧和低侧驱动的半桥驱动芯片 SLM27211
MOS 管	采用英飞凌型号为 IRF3205ZSTRLPBF

主控选型：采用 STM32F3 系列。该系列单片机具有超快速比较器（25 ns）和可编程增益的运算放大器。2 个 ADC，3 个 DAC，3 个 comp、一个运放和 1 个 4 通道 HRTIM 基本满足超级电容控制器需要精准测量电压电流的需求。更高性能 STMG4

系列电源专用芯片成本较高且会有性能浪费。

电流电压检测芯片：采用运算放大器测量电流电压。电流检测电路由检流电阻、检流芯片以及外围电容电阻构成，被检测电流经过检流电阻后产生压降（电阻两端电压会产生微弱的差异），检流芯片可以以上百倍的倍率放大电阻两端的电压差，被放大后的电压差输出到主控芯片的 ADC 引脚，通过 AD 转换就可以得到电压差的数字量，通过电压差和电阻值即可计算出电阻上流过的电流 I=U/R（此式中 R 为检流电阻固定的电阻值，U 为电阻两端电压差）。pcb 板上有三路这样的检流模块，用来检测并计算出输入电流，输出电流以及电容输入输出口的电流值。

电流检测电路中的检流电阻的大小以及检流芯片的放大倍数共同决定检流范围，INA240 为双向检流芯片，可以检测双向的电流，但是主控芯片只能检测正 3.3V 范围的电压。因此检流芯片输出的电压以 1.65V 的中间电压为参考。即检测到电流为零时输出电压为 1.65V，1.65-3.3V 为正向电流检测范围值，0-1.65V 为反向电流检测范围值。我们使用 INA240A2 芯片，其放大倍数为 50 倍，检流电阻为 5 毫欧，则最大正向检测电流=1.65/50/0.005=6.6A，要选择合适的检流电阻值，电阻值过小会导致检测精度太低，电阻值过大会导致检流范围不足。

电压检测电路采用 OPA2350UA/2K5 精密运放，由于控制器的主控芯片 ADC 检测最大电压值为 3.3V，而我们需要检测 0-24V 范围的电压值，因此需要通过运算放大器按倍数缩小被检测电压的值，通过运算放大器后输出一个 0-3.3V 范围的电压，此电压与真实电压成比例关系，而比例值由运放所接电阻决定

ADC_V_IN=(R105/R100)*VIN=(1/20)*VIN;

MOS 驱动：MOS 管驱动器采用具有独立的高侧和低侧驱动的半桥驱动芯片 SLM27211，该芯片内部集成自举二极管，外部需要连接自举电容，采用自举升压的方式驱动高侧 MOS 管；自举电容选取 100nF,芯片驱动电流峰值高达 4.5A，最大引导电压直流 120V；驱动器输出的信号经过阻尼电阻（12R）之后输入 MOS 管栅极，并联于阻尼电阻上的二极管起期间保护作用，在电路出现异常时为倒灌电流提供泄流路径，防止反向电流倒灌进入驱动芯片。

MOS 管：本设计中采用较为廉价的英飞凌型号为 IRF3205ZSTRLPBF 的 MOS 管，耐压达 55V，最大可持续通过 75A 电流，最小导通电阻 6.5 毫欧；而本设计中最高电压为 30V 远低于 MOS 管耐压；最大峰值电流为 10A 远低于 MOS 管最大持续电流。在实际测试中性能稳定，散热能力出众，性价比较高。

2.3.8 PCB 设计

图 2-10 主控板与功率板

由于大多数开发板上不存在控制 24V 电压通断的电路，为了解决这个问题，战队设计了 24V 通断控制模块。根据开发板 IO 口的输出电平控制模块上 PMOS 的通

断从而达到控制 24V 电路通断的作用。在实际运用中与继电器搭配可以避免很多不必要的走线。

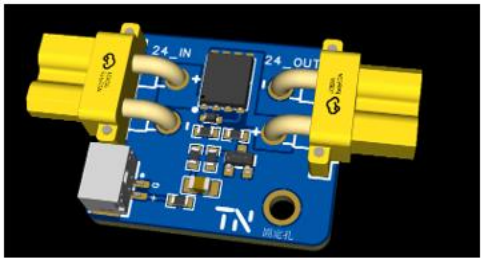


图 2-11 24V 通断控制模块

战队制作的通用 LED 驱动能够用于驱动各种 LED 灯，并且适配官方装甲板、基地引导灯等各种场景。为队内视觉组的调试提供了良好的硬件条件。使用 TP5410 芯片电池管理芯片对电池进行充电管理，根据芯片特性设计四颗 LED 反映当前电池剩余电量，清晰明了。

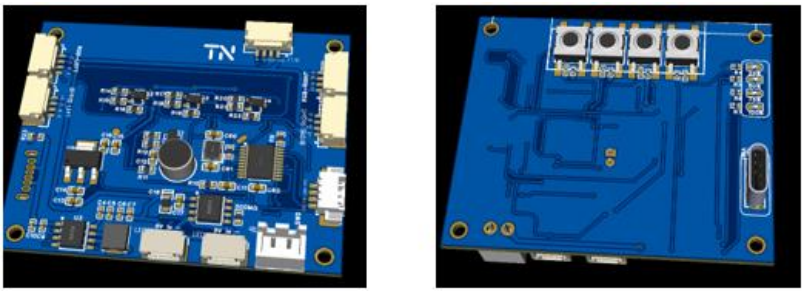


图 2-12 通用 LED 驱动

2.3.9 测试报告和记录说明

测试条件:

室温下使用电子负载仪和接入裁判系统（主控模块、电源管理模块、电容管理模块）。

测试方案:

校正功率控制板 ADC 采样后经过线性补偿后的电源管理模块 Chassis 端输出电压、电流；校正功率控制板 ADC 采样后经过简单线性补偿后的电容模组端口电压、电流。

输入两组真实值之后，读取实际采样值，将读取到的实际采样值进行线性校准补偿。

校准记录:

通过校正各端口电压电流，可以保证测量数据与真实值误差在 0.01 上下浮动。

表 10 校准记录

Board id = 0	真实值 1	测量值 1	真实值 2	测量值 2
inv	20.00f	20.00f	24.00f	24.78f
capv	17.00f	17.00f	22.00f	22.00f
inc	2.00f	2.00f	4.00f	4.00f
outc	2.00f	2.20f	4.00f	4.35f
capc	3.00f	3.00f	6.00f	6.00f
Board id = 1	真实值 1	测量值 1	真实值 2	测量值 2
inv	20.00f	21.48f	24.00f	25.33f
capv	17.00f	17.00f	22.00f	22.00f
inc	2.00f	2.00f	4.00f	4.00f

outc	2.00f	2.20f	4.00f	4.35f
capc	3.00f	3.00f	6.00f	6.00f
Board id = 2	真实值 1	测量值 1	真实值 2	测量值 2
inv	20.00f	21.43f	24.00f	25.62f
capv	17.00f	17.00f	22.00f	22.00f
inc	2.00f	2.00f	4.00f	4.00f
outc	2.00f	2.20f	4.00f	4.35f
capc	3.00f	3.00f	6.00f	6.00f
Board id = 3	真实值 1	测量值 1	真实值 2	测量值 2
inv	20.00f	21.41f	24.00f	25.51f
capv	17.00f	17.00f	22.00f	22.00f
inc	2.00f	2.00f	4.00f	4.00f
outc	2.00f	2.20f	4.00f	4.35f
capc	3.00f	3.00f	6.00f	6.00f
Board id = 4	真实值 1	测量值 1	真实值 2	测量值 2
inv	20.00f	21.39f	24.00f	25.22f
capv	17.00f	17.00f	22.00f	22.00f
inc	2.00f	2.00f	4.00f	4.00f
outc	2.00f	2.20f	4.00f	4.35f

capc	3.00f	3.00f	6.00f	6.00f

2.4 软件设计

软件整体基本架构由战队自主设计：COD_EC_Framework

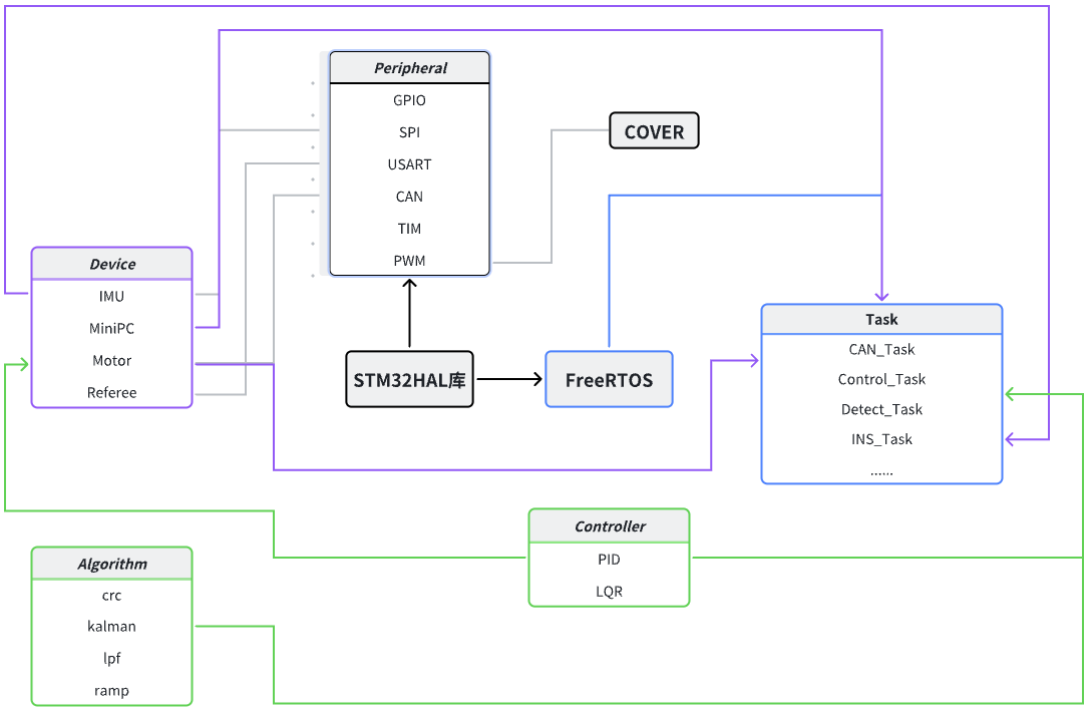


图 2-13 软件架构框图

2.4.1 开发环境

表 11 开发环境

名称	
开发工具	Keil MDK-ARM V5.39, Visual Studio Code
软件环境	Window11
硬件环境	DJI RoboMaster 开发板 C 型 (STM32F407IGHX)
编译工具	Arm Compiler v6.21

2.4.2 文件结构

```
COD_EC_Framework
    ├──────Algorithm ──────────算法层
    ├──────BSP ──────────BSP 基础层
    ├──────Controller ──────────控制器层
    ├──────Core ──────────内核层
    ├──────Drivers ──────────装置器件层
    ├──────MDK-ARM ──────────启动文件
    ├──────Middlewares ──────────中间件层。包含第三方，如 FreeRTOS、DSP 库等
    ├──────Modules ──────────姿态解算模型
    └──────Tasks ──────────任务层
```

图 2-14 文件结构

2.4.3 重点功能

自适应卡尔曼滤波

打滑是影响平衡步兵模型是否收敛的的一大原因，通常表现为平衡步兵碾压到小弹丸或突起的障碍物时，因速度突变导致平衡步兵控制系统发散，进而导致平衡

步兵失去稳定。为减小打滑的影响，可利用 IMU 获取机器人相对惯性空间的加速度，并通过当前速度与加速度从而估计下一时刻速度，设计适当的卡尔曼滤波器即可完成速度融合。

但在实际运用中，使用单一的线性卡尔曼滤波，最后的卡尔曼增益会逐渐收敛成一个定值。当卡尔曼增益较小时，滤波器更偏向实际轮毂反馈的速度，机器人依然存在打滑的情况；而卡尔曼增益较大的时候，滤波器更偏向加速度估计的速度，由于 IMU 加速度计存在误差，导致静止时解算出的加速度不为 0，机器人出现速度响应慢、静止时向前或向后移动等问题。因此我们采取自适应的卡尔曼滤波算法，当机器人不发生打滑时，滤波器更偏向实际轮毂反馈的速度，速度响应更快；而当机器人发生打滑时，滤波器更偏向惯性空间的估计速度，能有效防止出现机器人打滑的情况。

算法实际效果验证，我们选取 0.7 的滤波系数运行滤波器，让机器人通过铺满 17mm 小弹丸的地面：

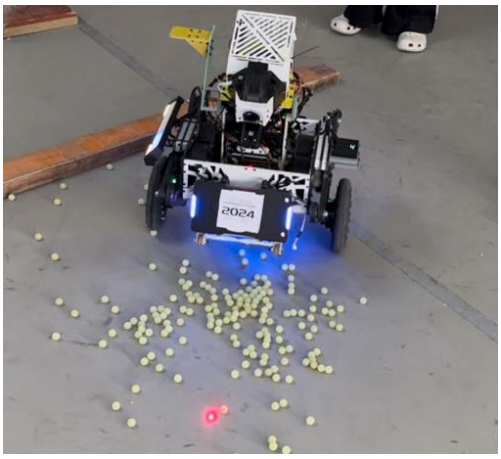


图 2-15 机器人通过小弹丸

通过串口打印未滤波前机体（Graph2）速度与滤波后机体（Graph2）速度（由于串口采样率的问题，串口绘图数据成波浪形，实际为正常曲线），在发生打滑时，可以看出，Graph2 可以很大程度降低轧过地形突起瞬间速度突变的程度，给予控制系统稳定可靠高质量的速度反馈值。

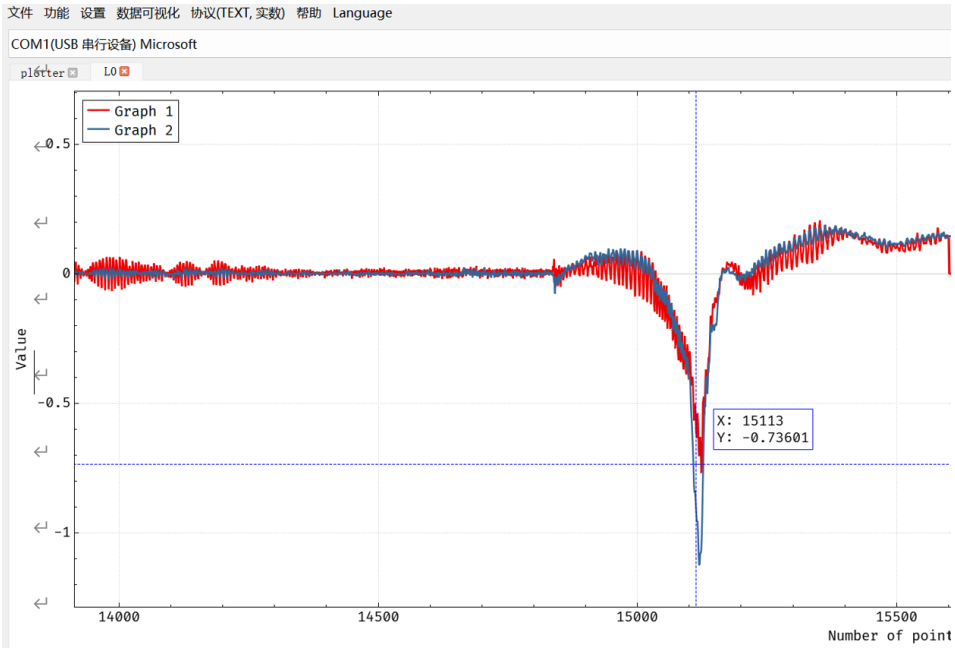


图 2-16 串口数据打印

功率控制

由平衡步兵 LQR 模型中可知，最后给驱动轮输出的是期望力矩。我队平衡步兵选用瓴控 MF9025 电机作为驱动轮电机，由电机参数可知额定扭矩T_R为 2.42 (N.m)，额定电流I_R为 7.45 (A)，扭矩常数k_{TC}为 0.32 (N.m/A)，电机的控制值Iqc为-2048-2048，对应电机的实际电流值Imc为-16.5~16.5A，根据公式：

$$k_{it} = \left(\frac{Imcmax}{Iqcmax}\right) / k_{TC}$$

可得出输出每 1N.M 对应的控制值系数k_{it}约为 387。

但是在实际上车之后发现，由于模型参数存在误差。如使用 387 作为k_{it}，会造成电机严重的超调与震荡。所以我们将k_{it}调整为 200，在该参数下驱动轮可兼顾力矩响应与稳定。

电机的功率主要由机械功率组成，查阅资料可得机械功率P_m：

$$P_m = (T * \omega)/k_c,$$

T 为电机输出力矩（N.M），ω为电机转速（rad/s）其中k_c为转化系数,通常在 9.0-10.0 之间。

由广东工业大学开源文档获知等式：

$$p_{in} = P_m + k_1\omega^2 + k_2T^2 + a$$

设指定功率为P_{Target}，取k_c为 9.5，代入上式，解得：

$$T = \frac{\frac{-\omega}{9.5} \pm \sqrt{\left(\frac{\omega}{9.5}\right)^2 - 4k_1\ (k_2\omega^2 + a - P_{Target})}}{2k_1}$$

此时采用功率再分配的控制策略，首先通过上述模型正向求得当前实时功率P_{cmd}，根据裁判系统反馈的最大功率限制P_{max}，可求得力矩放缩系数 K：

$$K = \frac{P_{max}}{P_{cmd}}$$

当底盘功率超过限制功率时，可通过 K 来得到有效限制。其中k_1、k_2、a三个参数，可以使用 Matlab 中的系统辨识工具箱收集电机实时电流、转速和功率，从

而进行更合理的参数整定。

自适应 PID

通过对平衡步兵系统建模，我们将跳跃分为四个阶段：第一阶段，机器人下蹲准备跳跃；第二阶段，机器人爆发起跳；第三阶段，机器人滞空收腿；第四阶段机器人伸腿缓冲落地。

在正常行驶状态，机器人关于腿长的 PID 的k_P系数较小，而跳跃需要关节电机瞬间的爆发力，正常行驶状态下的k_P系数响应不够，而在机器人落地缓冲时，又需要较小的k_P模拟弹簧减震。为实现轮腿的柔顺自适应与模仿弹簧减震效果，固需要机器人根据当前状态自适应 PID。

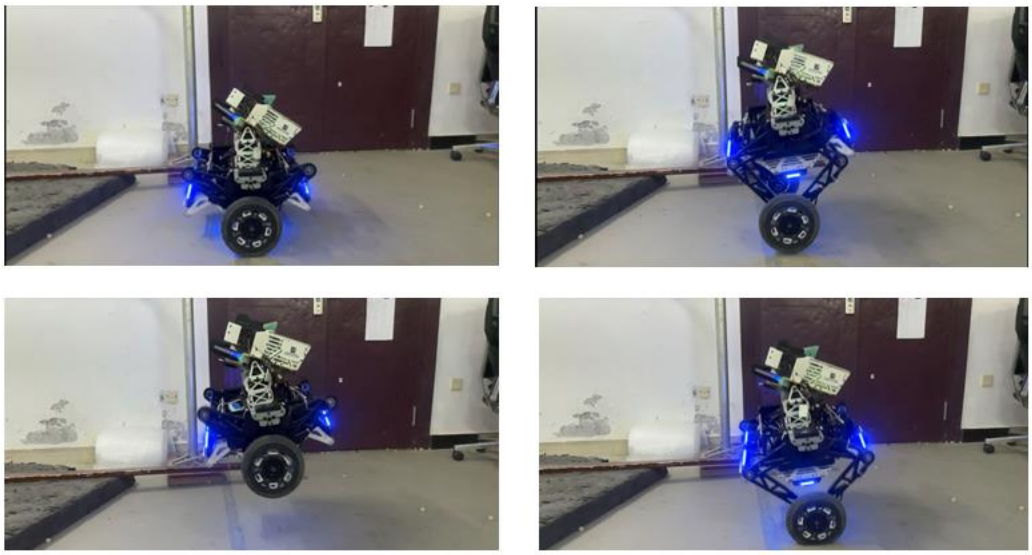


图 2-17 跳跃展示

2.5 算法设计

2.5.1 灯条识别

2024 赛季自动瞄准系统，在 2023 赛季的基础上改进了灯条过曝引起的白光扰乱问题。一般工业相机的动态范围有限，采用传统的通道相减法进行处理，易导致灯条中心过曝或者炫光，造成局部灯条空心。本赛季视觉组采取灯条灰度化方法进行二值化，寻找轮廓，获取最小外接矩形，根据像素面积、形状进行判断初筛，排除不符合条件的灯条，以进行后续装甲灯条的匹配处理。

在灯条颜色判断方面，通过统计指定范围内红色（R）与蓝色（B）通道像素，对比统计值多方面约束条件判断红蓝灯条。对过滤后灯条进行暴力匹配，结合长度、距离、等宽比等条件进行判断，从而得到符合装甲板特征的灯条组合并进行后续处理。

图 2-18 灰度图二值化处理

2.5.2 数字识别

经过灯条配对后组成的单装甲板，利用相似比信息，对感兴区域（ROI）进行透视变换，获取数字图像。由于图像中背景和数字背景图案黑白固定，采用大津法进行二值化处理，对数字的提取效果较好。值得一提的是，对数字的提取范围非常严

格，远离灯条近区，避免炫光影响处理效果。

数字识别部分采用深度学习算法，整体数字提取前处理具有旋转不变性，提取范围严格减少引入过多畸变，装甲数字特征明显，故利用多个神经元层组成多层感知机（MLP）进行分类，从线性过渡到非线性模型，通过损失梯度下降算法，使用 CIFAR-100 作为负样本，学习数字分类的权重偏置，实现对输入数字特征的非线性映射，从而解决更复杂的数字分类问题。

2.5.3 PNP 解算

同上赛季使用灯条两端像素角点，将装甲板作为像素平面，使用 Infinitesimal Plane-Based Pose Estimation (IPPE)方法，计算重投影误差最优解作为识别器输出，从而得到装甲板在相机坐标系下的平移矩阵与旋转向量，通过罗德里格（Rodrigues）旋转公式计算出欧拉角以便跟踪预测使用。

2.5.4 跟踪算法

使用多目标跟踪器 SORT 算法作为跟踪器的底层，关联装甲图像的前后帧处理，结合拓展卡尔曼滤波器成为观测跟踪器，整体观测敌方装甲与车体运动状态。在匀速空间中，通过预测后的速度以及解算的打击距离，对抛体弹道进行数学迭代以避免抛体运动无数值解问题。

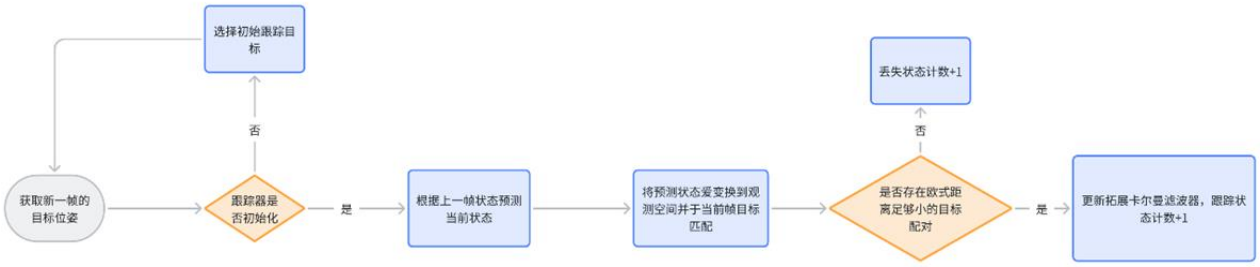


图 2-19 跟踪器工作流程

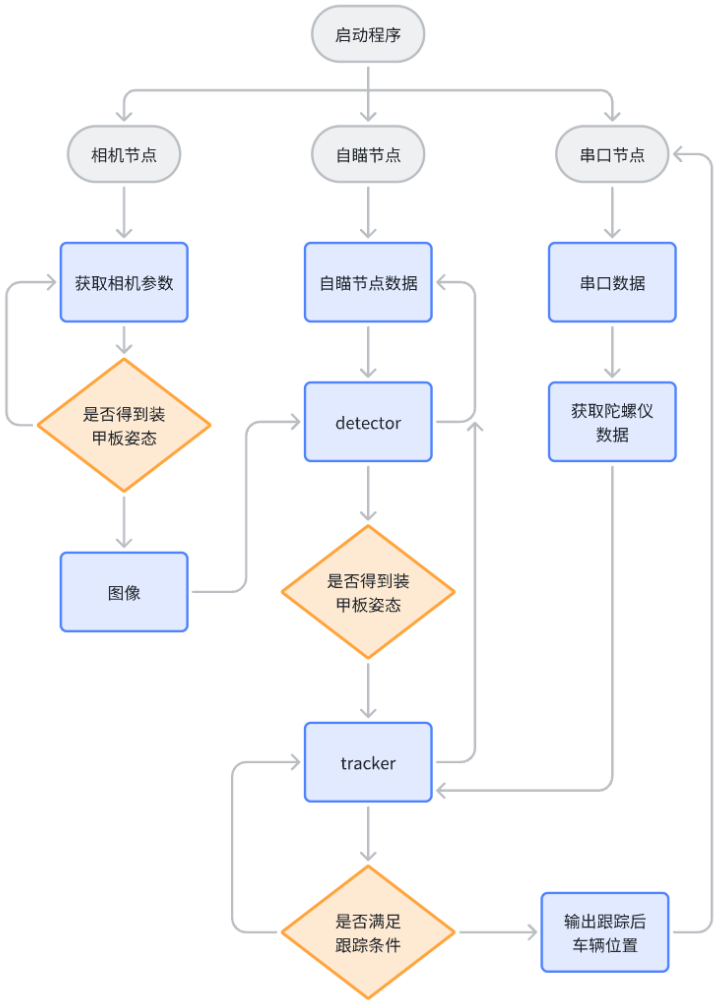


图 2-20 自瞄系统工作流程

2.5.5 灯条拟合装甲板

比赛中两条红/蓝灯条以及中间的数字组成装甲板视觉特征，处理过程中获取灰

度图后进行二值化处理并对其进行形态学处理，匹配完灯条时，在灯条数大于二的情况下，会进一步将两个灯条匹配为一个装甲板。

$$\begin{cases} \vec{v}_0 = l_{00} - l_{01} \\ \vec{v}_1 = l_{10} - l_{11} \\ \vec{v}_2 = P_{c0} - P_{c1} \\ l_0 = |l_{00} - l_{01}| \\ l_1 = |l_{10} - l_{11}| \\ l_2 = |P_{c0} - P_{c1}| \end{cases}$$

上式分别用限定两灯条的角度差、两灯条距离与其长度处于一定范围、两灯条的长度比值范围，通过两层循环进行匹配，当满足以上条件时即匹配成功。由于存在同一辆车的两块装甲板相邻灯条匹配到一起的情况，在匹配灯条后需要对相关区域做 ROI 处理，对获得的图像识别分类数字标签，进一步排除错误识别影响。

2.5.6 装甲板预测

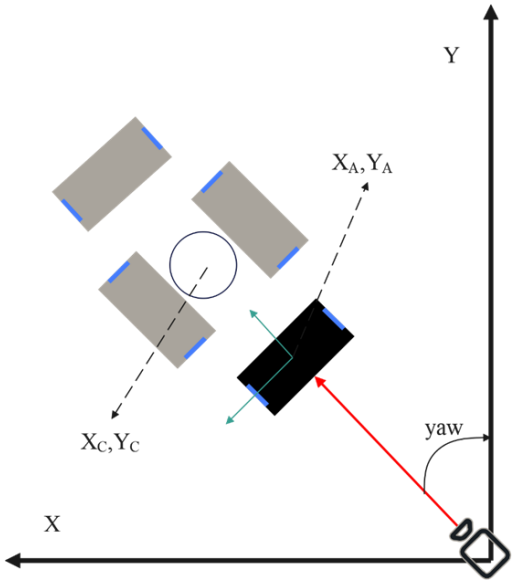


图 2-21 整体观测示意图

参考上交的多重卡尔曼观测结构与华师的整体观测方案，整理出基于拓展卡尔

曼的整车估计方案，由于目标在三维空间，摄像头采集信息为像素平面，是一个无深度信息的二维平面空间，此时处理装甲信息需要从多维度进行处理，采取的方法是先降维成二维，去除高度信息，或是在高度已知的情况下，将其转化为二维可处理的简单估计。

观测二维平面中装甲的位置、姿态，结合过去信息去滤波、预测车辆整体信息，反推出装甲模块未来位置。观测的主要变量为装甲的中心位置信息（xa，ya，za），装甲相对于本体惯性坐标系下的偏差值 yaw。

设本体状态 x 如下:

$$\vec{x} = [x_c, \dot{x}_c, y_c, \dot{y}_c, z_c, \dot{z}_c, yaw, omiga, r_1, r_2]^T$$

存在状态转移矩阵 F，构建运动模型如下:

$$\vec{x}_{i+1} = \vec{F}\vec{x}_i$$

$$\vec{F} = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

式中Δt 为采集图片时两帧图片的时间戳之差。同 ROI 处理相同，在切换目标时需重置相关参数，否则会因为角度的突变导致无法收敛，使车辆云台剧烈抖动以至于无法识别装甲板。

2.5.7 优缺点分析

使用 Docker 进行部署和版本控制具有显著的优势，可以快速进行部署，并且能够有效控制算力资源的使用。此外，Docker 还能够实现单台电脑多个程序的同时运行，并且彼此相互隔离，确保系统的稳定性和安全性。

在灯条识别方面，采用灰度图进行二值化的方法具有较高的准确度，但需要对相关参数进行调整。这一过程需要耗费大量时间进行调试，为"专车专调"，耗费较多时间。针对数字部分的识别，则采用多重感知机进行处理。在训练阶段，感知机相对简单且准确度高，能够有效识别数字部分。

考虑在之后赛季使用自适应参数，设置状态之后自动进行调参，节约时间；并尝试使用神经网络对部分识别代码部分进行替换，尽可能提高识别速度和准确率。

2.5.8 算法库介绍

表 12 算法库介绍与接口说明

算法库	说明
OpenCV、Eigen、mutex	使用 ros2 封装代码，以节点方式存储不同模块，并通过话题的方式在节点中进行数据传输

2.5.9 算法结果

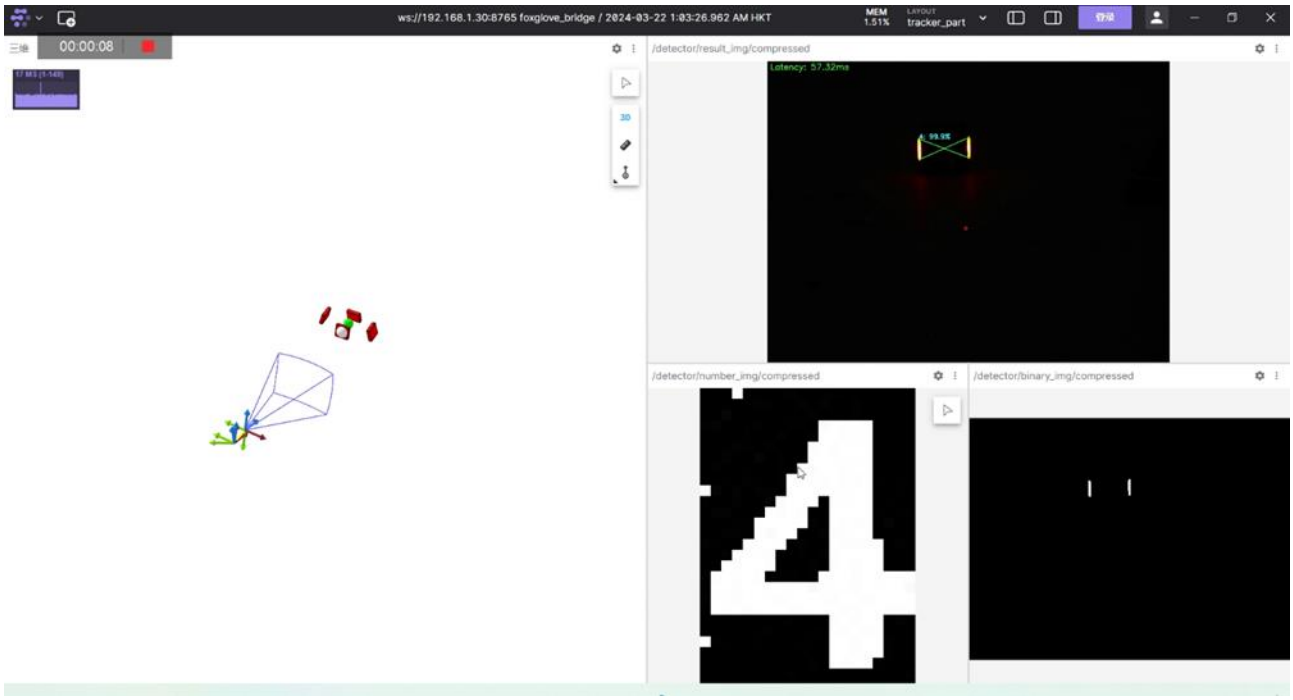


图 2-22 算法结果展示

2.6 UI 设计

2.6.1 UI 底层代码

本赛季我队机器人 UI 设计的代码部分采用的南京航空航天大学开源的 UI 设计器

(https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22955)

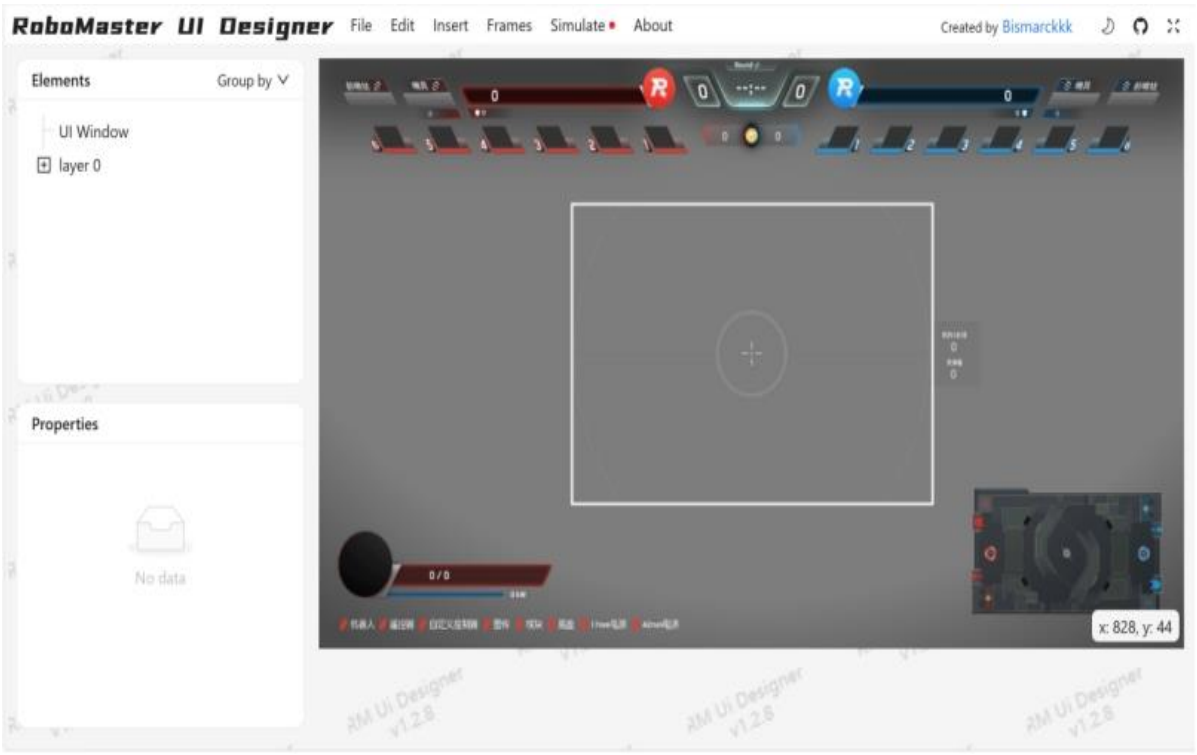


图 2-23 UI 显示界面

在其自动生成的 UI 代码基础上，适配了我队自主研发的电控框架

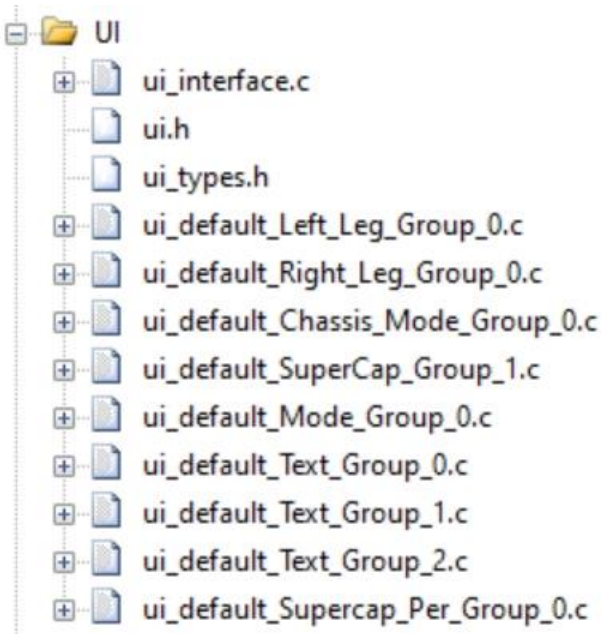


图 2-24 UI 代码目录结构

2.6.2 UI 设计理念

UI 动态刷新，我们直接修改每个包中图案对应的结构体指针的相应参数，通过调用南京航空航天大学开源的 UI 设计器生成的代码中的 Update 更新函数实现。

通过查询 RoboMaster 裁判系统串口协议附录 V1.6.3 可知，在 0x0301CMD_ID 下的上行频率最大为 10Hz，在我们实际测试中发现，两个 UI 包以 10ms 为间隔轮流发送给裁判系统，UI 将会卡死不能刷新。经过测试，将每个包之间发送频率修改为 30ms 发送，UI 可实现正常较高频率的更新。

```
  if(remote_ctrl.key.set.B == 1){
    Last_Systick1 = 0;
    if(systick % 50 == 0 && Last_Systick == 0){

      _ui_init_default_Right_Leg_Group_0();

      Last_Systick = systick;
    }else if(systick  == Last_Systick + 50){

    _ui_init_default_Left_Leg_Group_0();

    }else if(systick  == Last_Systick + 100){

      _ui_init_default_Chassis_Mode_Group_0();


    }else if(systick  == Last_Systick + 150){

      _ui_init_default_SuperCap_Group_1();


    }else if(systick  == Last_Systick + 200){

      _ui_init_default_Mode_Group_0();

    }else if(systick  == Last_Systick + 250){

       _ui_init_default_Text_Group_0();

    }else if(systick  == Last_Systick + 300){
```

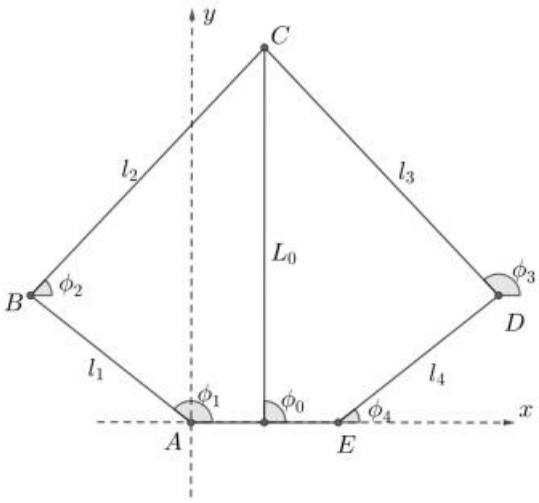
```
31
32          _ui_init_default_Text_Group_1();
33
34      }else if(systick  == Last_Systick + 350){
35
36          _ui_init_default_Text_Group_2();
37
38      }else if(systick  == Last_Systick + 400){
39
40        _ui_init_default_Supercap_Per_Group_0();
41        Last_Systick = 0;
42      }
43
44     }else{
45       Last_Systick = 0;
46       if(systick % 30 == 0 && Last_Systick1==0){
47
48       Right_Leg_Coordinate_Calucate();
49       _ui_update_default_Right_Leg_Group_0();
50         Last_Systick1 = systick;
51
52      }else if(systick  == Last_Systick1 + 30){
53
54           Left_Leg_Coordinate_Calucate();
55         _ui_update_default_Left_Leg_Group_0();
56
57      }else if(systick  == Last_Systick1 + 60){
58
59         Chassis_Mode_Calucate();
60        _ui_update_default_Chassis_Mode_Group_0();
61
62      }else if(systick  == Last_Systick1 + 90){
63
64        SuperCap_Calucate();
65       _ui_update_default_SuperCap_Group_1();
66
67      }else if(systick  == Last_Systick1 + 120){
68
69       ui_default_Supercap_Per_Group_Supercap->number = SuperCap_Info.Ca
   p_Percent;
70       _ui_update_default_Supercap_Per_Group_0();
71
72      }else if(systick  == Last_Systick1 + 150){
73
74       Mode_Calucate();
75        _ui_update_default_Mode_Group_0();
76         Last_Systick1 = 0;
77      }
```

```
78
79      }
```

因每个包之间发送频率修改为 30ms 发送，当发送包数量过多时（如上所示为 6 个时）一些需要高实时刷新的 UI 已存在明显延迟、卡顿的现象。后续将降低一些实时性较低的 UI 发送频率，实现刷新带宽的合理分配。

2.6.3 部分动态 UI 实现代码

2.6.3.1 腿部五连杆 UI



根据五连杆正运动学解算，以 A 点为坐标原点，根据两关节电机反馈的角度 Phi1、Phi4，即可解算出 B、C、D、E 点的相对坐标。以 AE 为作为基础线，根据解算出来的坐标即可算出 l1,l2,l3,l4,l5 的起点终点，将其代入到结构体指针中，即可实现动态刷新。

图 2-25 原理图

```
1   static void Left_Leg_Coordinate_Calucate(void){

2   ui_default_Left_Leg_Group_L1->end_x =  ui_default_Left_Leg_Group_L5->en
    d_x + 80 * (-arm_cos_f32(*Control_Info.L_Leg_Info.VMC.Phi1));
3   ui_default_Left_Leg_Group_L1->end_y =  ui_default_Left_Leg_Group_L5->en
    d_y - 80 * (arm_sin_f32(*Control_Info.L_Leg_Info.VMC.Phi1));

4   ui_default_Left_Leg_Group_L4->start_x =  ui_default_Left_Leg_Group_L5->
    start_x - 80 * (arm_cos_f32(*Control_Info.L_Leg_Info.VMC.Phi4));
5   ui_default_Left_Leg_Group_L4->start_y =  ui_default_Left_Leg_Group_L5->
    start_y - 80 * (arm_sin_f32(*Control_Info.L_Leg_Info.VMC.Phi4));

6   ui_default_Left_Leg_Group_L2->start_x =   ui_default_Left_Leg_Group_L5-
    >end_x  -
```

```
    ( 146 * arm_cos_f32(Control_Info.L_Leg_Info.VMC.Phi2) + 80 * arm_cos_f3
    2(*Control_Info.L_Leg_Info.VMC.Phi1)) ;
7   ui_default_Left_Leg_Group_L2->start_y =   ui_default_Left_Leg_Group_L5-
    >end_y  -
    ( 80 * arm_sin_f32(*Control_Info.L_Leg_Info.VMC.Phi1) + 146 * arm_sin_f
    32(Control_Info.L_Leg_Info.VMC.Phi2));

8   ui_default_Left_Leg_Group_L2->end_x =    ui_default_Left_Leg_Group_L1->
    end_x;
9   ui_default_Left_Leg_Group_L2->end_y =    ui_default_Left_Leg_Group_L1->
    end_y;


10  ui_default_Left_Leg_Group_L3->start_x =   ui_default_Left_Leg_Group_L4-
    >start_x;
11  ui_default_Left_Leg_Group_L3->start_y =   ui_default_Left_Leg_Group_L4-
    >start_y;

12  ui_default_Left_Leg_Group_L3->end_x =   ui_default_Left_Leg_Group_L2->s
    tart_x;
13  ui_default_Left_Leg_Group_L3->end_y =    ui_default_Left_Leg_Group_L2->
    start_y;

14  }
```

2.6.3.2 底盘状态以及云台朝向 UI

轮腿平衡步兵机器人与普通步兵机器人相比, 底盘状态多一个侧身状态, 跟随模式、侧身模式、小陀螺模式也有所不同。因轮腿机器人整体成矩形，固采用矩形来表示跟随模式。跟随模式即将矩形旋转 90 度。小陀螺模式用圆来表示。并在中间用一根较粗的线条表示云台朝向。根据 Yaw 轴电机误差以中心画圆即可实现云台朝向显示。

```
1   static void Chassis_Mode_Calucate(void){
2
3     if(Control_Info.Chassis_Mode == CHASSIS_FRONT){
4           ui_default_Chassis_Mode_Group_CHASSIS_FRONT->start_x = 881;
5       ui_default_Chassis_Mode_Group_CHASSIS_FRONT->start_y = 92;
6       ui_default_Chassis_Mode_Group_CHASSIS_FRONT->end_x = 1064;
7       ui_default_Chassis_Mode_Group_CHASSIS_FRONT->end_y = 211;
8
9       ui_default_Chassis_Mode_Group_CHASSIS_SIDE->start_y = 68 + 1080;
10      ui_default_Chassis_Mode_Group_CHASSIS_SIDE->end_y = 241 + 1080;
```

```
11
12          ui_default_Chassis_Mode_Group_CHASSIS_SPIN->start_x = 972;
13          ui_default_Chassis_Mode_Group_CHASSIS_SPIN->start_y = 1080+80;
14
15
16          }else if(Control_Info.Chassis_Mode == CHASSIS_SIDE) {
17
18              ui_default_Chassis_Mode_Group_CHASSIS_SPIN->start_x = 972;
19          ui_default_Chassis_Mode_Group_CHASSIS_SPIN->start_y = 1080+80;
20
21          ui_default_Chassis_Mode_Group_CHASSIS_FRONT->start_y = 92 + 1080;
22          ui_default_Chassis_Mode_Group_CHASSIS_FRONT->end_y = 211+1080;
23
24          ui_default_Chassis_Mode_Group_CHASSIS_SIDE->start_x = 912;
25          ui_default_Chassis_Mode_Group_CHASSIS_SIDE->start_y = 68;
26          ui_default_Chassis_Mode_Group_CHASSIS_SIDE->end_x = 1030;
27          ui_default_Chassis_Mode_Group_CHASSIS_SIDE->end_y = 241;
28
29          }else if(Control_Info.Chassis_Mode == CHASSIS_SPIN){
30
31              ui_default_Chassis_Mode_Group_CHASSIS_SPIN->start_x = 972;
32          ui_default_Chassis_Mode_Group_CHASSIS_SPIN->start_y = 151;
33
34            ui_default_Chassis_Mode_Group_CHASSIS_FRONT->start_y = 92 + 1080;
35          ui_default_Chassis_Mode_Group_CHASSIS_FRONT->end_y = 211+1080;
36
37              ui_default_Chassis_Mode_Group_CHASSIS_SIDE->start_y = 68 + 1080;

38          ui_default_Chassis_Mode_Group_CHASSIS_SIDE->end_y = 241 + 1080;
39
40          }
41
42          if(Control_Info.Yaw_Err > 1.f){
43
44              ui_default_Chassis_Mode_Group_Shoot->end_x = ui_default_Chassis_M
    ode_Group_Shoot->start_x +  arm_sin_f32((float)(Control_Info.Yaw_Err)*Ang
    le_to_rad)*93;
45          ui_default_Chassis_Mode_Group_Shoot->end_y = ui_default_Chassis_Mode_
    Group_Shoot->start_y +  arm_cos_f32((float)(Control_Info.Yaw_Err)*Angle_t
    o_rad)*93;
46        }else if(Control_Info.Yaw_Err  < -1.f){
47
48            ui_default_Chassis_Mode_Group_Shoot->end_x = ui_default_Chassis_Mod
    e_Group_Shoot->start_x + arm_sin_f32((float)(Control_Info.Yaw_Err)*Angle_
    to_rad)*93;
49          ui_default_Chassis_Mode_Group_Shoot->end_y = ui_default_Chassis_Mode_
    Group_Shoot->start_y + arm_cos_f32((float)(Control_Info.Yaw_Err)*Angle_to
    _rad)*93;
50
51          }else if(Control_Info.Yaw_Err  < 1.f && Yaw_Err >-1.f){
52
53            ui_default_Chassis_Mode_Group_Shoot->end_x = 972;
54          ui_default_Chassis_Mode_Group_Shoot->end_y = 244;
55      }
56
57
58      }
```

3. 研发迭代过程

3.1 测试记录

3.1.1 超级电容测试记录

负责人：涂仁杰

电容测试分为前期测试和上车测试，前期测试使用数控电源和负载仪对电容进行基础功能测试、疲劳测试和暴力测试。上车测试确保机器人连接完整裁判系统，进入比赛在不同功率限制之下进行基础功能测试和疲劳测试。对出现的问题进行记录并及时解决

一、前期测试

1. 电容重要参数

ID	in_v_1	in_v_2	cap_v_1	cap_v_2	in_c_1	in_c_2	out_c_1	out_c_2
0	20.58	24.58	17.00	22.00	2.13	4.13	2.20	4.35
1	21.48	25.78	17.00	22.00	2.00	4.00	2.20	4.35
2	21.43	25.62	17.00	22.00	2.00	4.00	2.20	4.35
3	21.41	25.51	17.00	22.00	2.00	4.00	2.20	4.35
4	20.72	24.88	17.00	22.00	2.00	4.00	2.20	4.35
5	21.12	25.35	17.00	22.00	2.08	4.11	2.20	4.35
6	21.15	25.38	17.00	22.00	2.08	4.11	2.20	4.35

```
const float calibration_parameters[ID][10] = {
{20.58f, 24.58f, 17.00f, 22.00f, 3.00f, 6.00f, 2.13f, 4.13f, 2.20f, 4.35f},
```

{21.48f, 25.78f, 17.00f, 22.00f, 3.00f, 6.00f, 2.00f, 4.00f, 2.20f, 4.35f},
{21.43f, 25.62f, 17.00f, 22.00f, 3.00f, 6.00f, 2.00f, 4.00f, 2.20f, 4.35f},
{21.41f, 25.51f, 17.00f, 22.00f, 3.00f, 6.00f, 2.00f, 4.00f, 2.20f, 4.35f},
{20.72f, 24.88f, 17.00f, 22.00f, 3.00f, 6.00f, 2.00f, 4.00f, 2.09f, 4.18f},
{21.12f, 25.35f, 17.00f, 22.00f, 3.00f, 6.00f, 2.08f, 4.11f, 2.19f, 4.35f},
{21.15f, 25.38f, 17.00f, 22.00f, 3.00f, 6.00f, 2.08f, 4.11f, 2.19f, 4.35f}

2. 电容基础功能测试

ID	ADC_1	ADC_2	ADC_3	actual_1	actual_2	actual_3	$\overline{err}$	充放电	保护
0	8.04	50.21	70.22	8.43	51.31	71.34	+0.8700	√	√
1	6.52	63.71	34.35	6.33	63.12	33.89	-0.2466	√	√
2	5.32	24.52	88.42	5.06	24.11	88.21	-0.2933	√	√
3	10.11	28.26	105.12	9.98	28.12	104.95	-0.1466	√	√
4	33.27	38.89	40.91	33.01	38.71	42.11	+0.2533	√	√
5	44.31	67.98	78.12	42.48	66.71	79.21	-0.6700	√	√
6	43.28	22.17	63.66	44.52	23.14	64.87	+1.1400	√	√

3. 电容疲劳测试

ID	底盘功率/ W	测试前温度/ ℃	测试后温度/ ℃	工作状态	测试时间/ min
0	120	28.1	31.8	正常	20
1	120	25.3	28.6	正常	22
2	120	30.4	33.1	正常	20

ID	底盘功率/ W	测试前温度/ ℃	测试后温度/ ℃	工作状态	测试时间/ min
3	120	30.2	33.7	正常	20
4	120	30.6	32.6	正常	19
5	120	29.7	32.8	正常	21
6	120	30.2	33.8	正常	20

ID	底盘功率上限/ W	测试前温度/ ℃	测试后温度/ ℃	工作状态	测试时间/ min	充放电次数
0	45	21.7	41.3	正常	20	20
	65	28.6	49.8	正常	20	23
	85	37.1	59.1	正常	20	30
	105	36.6	59.7	正常	20	39
	125	28.4	57.6	正常	20	42
1	45	31.7	51.4	正常	20	21
	65	34.4	56.5	正常	20	25
	85	27.3	58.1	正常	20	34

ID	底盘功率上限/W	测试前温度/℃	测试后温度/℃	工作状态	测试时间/min	充放电次数
0	45	21.7	41.3	正常	20	20
	105	31.4	61.6	正常	20	37
	125	38.4	64.6	正常	20	44
2	45	35.8	52.4	正常	20	21
	65	26.6	41.6	正常	20	25
	85	34.2	57.1	正常	20	31
	105	31.2	55.6	正常	20	39
	125	25.2	54.6	正常	20	43
3	45	24.6	41.4	正常	20	21
	65	23.4	42.7	正常	20	23
	85	33.3	51.1	正常	20	30
	105	32.6	55.6	正常	20	37
	125	33.4	67.6	正常	20	41
4	45	24.2	53.4	正常	20	20

ID	底盘功率上限/W	测试前温度/℃	测试后温度/℃	工作状态	测试时间/min	充放电次数
0	45	21.7	41.3	正常	20	20
	65	27.1	46.6	正常	20	24
	85	36.4	53.1	正常	20	31
	105	38.6	59.6	正常	20	39
	125	38.4	67.3	正常	20	43
5	45	34.7	53.4	正常	20	21
	65	24.4	42.6	正常	20	25
	85	31.3	55.1	正常	20	31
	105	37.6	59.6	正常	20	39
	125	35.4	61.6	正常	20	43
6	45	34.7	53.4	正常	20	21
	65	24.4	42.6	正常	20	25
	85	31.3	55.1	正常	20	31
	105	37.6	59.6	正常	20	39

ID	底盘功率上限/ W	测试前温度/ ℃	测试后温度/ ℃	工作状态	测试时间/ min	充放电次数
0	45	21.7	41.3	正常	20	20
	125	35.4	61.6	正常	20	43

二、上车测试

上车测试确保机器人连接完整裁判系统，并记录测试过程中但不限于测试过程中电容在具体兵种测试过程出现的问题。并提供解决方案。

1. 基础功能测试

ID	底盘功率限制/ W	电容响应速度	电容工作	CAN通信
0	45	快	√	√
	65	快	√	√
	85	快	√	√
	105	快	√	√
	125	快	√	√
4	45	快	√	√
	65	快	√	√

ID	底盘功率限制 / W	电容响应速度	电容工作	CAN通信
	85	快	√	√
	105	快	√	√
	125	快	√	√

2. 疲劳测试

项目	ID	次数	底盘功率限制 / W	跑场时间 / min	电容工作
跑场	0	1	55	10	正常
		2	55	11	正常
		3	55	12	正常
		4	55	12	正常
		5	55	11	正常
		6	55	16	正常

项目	ID	次数	底盘功率限制 / W	测试前容量 / %	测试后容量 / %	飞坡是否成功	电容工作
飞坡	0	1	55	99	21	否	正常
		2	55	99	26	否	正常

项目	ID	次数	底盘功率限制 / W	测试前容量 / %	测试后容量 / %	飞坡是否成功	电容工作
		3	55	99	25	是	正常
		4	55	99	23	否	正常

三、常见问题及解决方案

常见问题	解决方案
当电容和缓存全部用完时，电容会抢占缓存上升所需的能量。导致缓存上升速度非常非常慢	拉低最小功率上限
电容在电池低电量时容易跑死，即不起作用	拉低低电量保护的阈值
轮腿在上电后两个驱动轮电机会进保护，导致无法正常工作	去除底盘与电源之间的防倒灌二极管。初步怀疑电机进保护的原因是电机无法自行消化电机反电动势能量，导致进保护，去除二极管后相当于将能力给电池反向充电
机器人裁判系统底盘断电后电容会供电导致无法完全断电。（检录无法通过）	电控方面做控制，读到底盘断电，失能
电容有时候突然跑死无法工作	初步怀疑是在突然加速或突然大功率导致电池电压突然下降，程序进低电压保护，当然不排除ADC采样到错误的数据的情况

3.2 版本迭代过程记录

表 13 版本迭代过程记录

版本号或阶段	功能或性能详细说明	完成时间
V1.0	第一版平衡步兵机器人底盘机械装配完成	2023.12.10
V1.1	第一版平衡步兵机器人底盘基本运动完成	2024.1.3
V1.2	第一版平衡步兵机器人整车装配完成	2024.2.24
V1.3	第一版平衡步兵机器人整车完整功能完成	2024.3.18
V2.0	1. 综合考量下，决定放弃下供弹云台方案， 上供弹云台 2. 第二版轮腿平衡步兵机器人完成加工 3. 重新对底盘电气结构进行布局，优化空间 4. 完成云台和底盘基本功能	2024.3.29
V2.1	调整 LQR 参数，完成第二代底盘算法部署	2024.4.8
V2.2	1. 完成底盘支持力解算 2. 部署底盘飞坡算法，较稳定飞坡 3. 部署底盘离地算法，将机器人置于空中可 不发散	2024.4.15
V2.3	1. 操作手训练 2. 综合测试	2024.4.27

3.3 重点问题解决记录

表 14 重点问题解决记录

序号	问题描述	问题产生原因	解决方案	机器人版本号	解决人员
1	平衡步兵机器人无法正常站立	模型部分参数极性错误	修改模型中部分参数正负极	V1.1	电控：王芃
2	平衡步兵机器人抗扰性差	LQR 权重系数错误	调整 LQR 权重系数	V1.1	电控：王芃
3	平衡步兵打滑容易摔倒	碾压到弹丸速度发生突变	融合机体加速度	V1.2	电控：王芃
4	超级电容使底盘电机进入过压保护	电容控制板二极管选型不合理	更换二极管	V1.3	电控：王芃 硬件：涂仁杰
5	1．底盘无法平衡或以奇形怪状的方 2．底盘无法正确 3．机器人控制系统非常容易发散，踩到弹丸或过障碍时非常容易摔倒。	1．对模型理解不够透彻，部分变量正负号不对 2．转向算法采用的角速度单控制环	1．将变量极性改为与模型相匹配的方向 2．采用角度角速度的串级 PID 算法 3．问题 3 未解决	V2.0	电控：王芃

6	1．超级电容在底盘无地方放置。 2．云台俯角过大，在较大俯角时发射小弹丸会击打底盘。 3．机器人控制系统非常容易发散，踩到小弹丸或过障碍时非常容易摔倒	1．机械考虑到超级容的安装。 2．机械将赛季普通步的云台直接来使用，为根轮腿底盘重设计	1．将超级电容绑在云台上部分，下一版本进行更改。 2．加大云台高度 3．问题未解决	V2.1	机械组：王志强
7	1．机器人控制系统非常容易发，踩到小弹丸或过障碍时非常容易摔倒。 2．机器人在摔后难以纠正，会出”滑跪”的情况。	1．LQR 模中一关键变量使用错误 2．机器人体速度、位数据计算错误 3．未对机人摔倒后做后续处理	．更换为正确的变量 ．使用自适应卡尔曼滤波对速度进行估计、得到可靠、稳定的速度 ．更换机器人倒地后的处理算法	V2.2	电控组：王芃

8	1．板间通信有时候会断连，下板收到的数据不会及时更新 2．Yaw 轴 6020 电机突然失能。无法控制。	1． CAN 通信出现问题。	1．未解决，后续将通过 CAN 分析仪查看	V2.3	电控组：王芃

4. 团队成员贡献

表 15 团队成员贡献

姓名	基本信息	主要负责工作内容描述	贡献度
王芃	22 级冶金工程 嵌入式软件开发及控制系统设计负责人	负责机器人整体嵌入式开发	35%
王禹茼	21 级机械制造及其自动化 机械结构设计者	负责机器人底盘机械结构设计	18%
王志强	21 级材料成型及控制工程 机械结构负责人	负责机器人整体机械设计与装配	17%
涂仁杰	22 级光电信息科学与工程 硬件设计负责人	负责机器人整体硬件维护和超级电容开发	18%
王柏程	21 级软件工程 视觉系统负责人	负责机器人整体视觉自瞄部署及调试	12%

5. 参考文献

[1] 陈阳,王洪玺,张兰勇.轮腿式平衡机器人控制[J].信息与控制,2023,52(05):648-659.DOI:10.13976/j.cnki.xk.2023.2533.

[2] 柏龙,葛文杰,陈晓红,等.星面探测仿生间歇式跳跃机器人设计及实现[J].机器人,2012,34(01):32-37+43.

[3] 于红英,唐德威,王建宇.平面五杆机构运动学和动力学特性分析[J].哈尔滨工业大学学报,2007(06):940-943.

[4] 谢惠祥. 四足机器人对角小跑步态虚拟模型直觉控制方法研究[D].国防科学技术大学,2015.

[5] 【RM2022-双枪平衡步兵机械开源】哈尔滨工程大学创梦之翼 https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22138&fromuid=67786

[6] 【RM2023-平衡步兵控制系统开源】上海交通大学-云汉交龙 https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22756&fromuid=67786

6. 技术方案复盘

6.1 赛场机器人性能表现情况分析

本赛季我队参加了区域赛的 6 场小组赛，轮腿平衡步兵机器人在其中 3 场准备阶段出现问题未能上场，1 场比赛中出现失控状况。其对于高强度的比赛稳定性和性能难以保证。在其正常上场的 2 场比赛中，性能较稳定，表现有可取之处，但远没有达到我们赛季初设定的目标和期望的性能。

- 全地形适应：在比赛中，轮腿平衡步兵机器人基本完成了除上台阶外的所有功能，能稳定爬上比赛场地中的所有斜坡、稳定下 15cm 台阶等功能。但在飞坡等进阶功能中表现不够稳定。
- 抗干扰能力：在比赛中，面对对方机器人的冲撞时能保证较好的鲁棒性，但在一场比赛中单边腿被障碍物绊倒时出现了翻车的情况。因未对机器人进行整体建模，当单边腿出现问题时容易使控制系统发散。
- 自动瞄准：相较于上赛季，我队步兵自瞄能力有较大提升，在场上没有存在频繁掉线情况，但依然存在瞄不准、易发散的情况。
- 功率控制：新赛季研发的功率控制算法表现良好，在场上未出现超功率的情况，但对功率的充分利用率还有待提高
- 准备阶段失控&场上失控：在比赛前一个月，机器人就出现了时常失控的情况，具体表现为 Yaw 轴电机突然失能，遥控器无法控制。或者板间通讯出现断连，

底盘无法正常控制的情况。排查情况为 CAN 信号受干扰导致。

6.2 赛场机器人性能表现与规划对比分析

表 16 对比分析表

机构	规划需求	实际性能表现
底盘	稳定上下环高、梯高、飞坡通道的斜坡。	较稳定上下环高、梯高、飞坡通道的斜坡
	稳定上下 15cm 台阶	稳定下 15cm 台阶，上 15cm 未能实现
	稳定通过铺满小弹丸的起伏路段不打滑	稳定通过铺满小弹丸的起伏路段不打滑
	稳定侧身防守对面	稳定侧身防守对面
	被包夹时开启小陀螺防守	被包夹时开启小陀螺防守
云台	双轴云台响应快	采用前馈+PID 的控制算法，云台响应快
	5 米小装甲板命中率 30%	近距离面对对方机器人，命中率高
	自瞄跟踪目标快速准确	40 发小弹丸面对敌方哨兵机器人，造成 150 滴伤害，

		自瞄跟踪较准确。
	装配精准，小陀螺运动云台 晃动幅度低，减小因重量问题 导致的电机负担	未能良好实现。Yaw 电机存在装配问题，运动时云台 晃动幅度低
发射机构	供弹系统稳定 保证不卡弹。	采用八尺拨弹盘不卡弹
	射频 20hz	未实现，未能对发射机构进行创新
	弹道稳定	更换摩擦轮，弹道精度有所提升

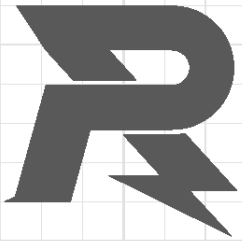
6.3 经验总结

1. 对于技术上出现的问题，需要记录并用严谨的方法排查、善用各种仪器、而不是靠玄学解释问题。
2. 关键部位的材料选择时，应适当地理论分析，避免增加后期的试错时间与经济成本。
3. 需理解我们所做东西的本质，究其原理，提高效率也需要摸清事物底层逻辑，

抓住根本原因吧 。

4. 好的机械设计能减少电控和视觉很多没必要的工作, 要在初期就把控好大方向, 确定设计方案

5. 好的管理是拿下好成绩的前提，赛前测试同等重要。管理能够节约时间成本、最大程度地利用已有资源；测试可以帮助发现并解决问题。二者须有机统一，任一过程的缺失都会造成不可预估的后果。



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