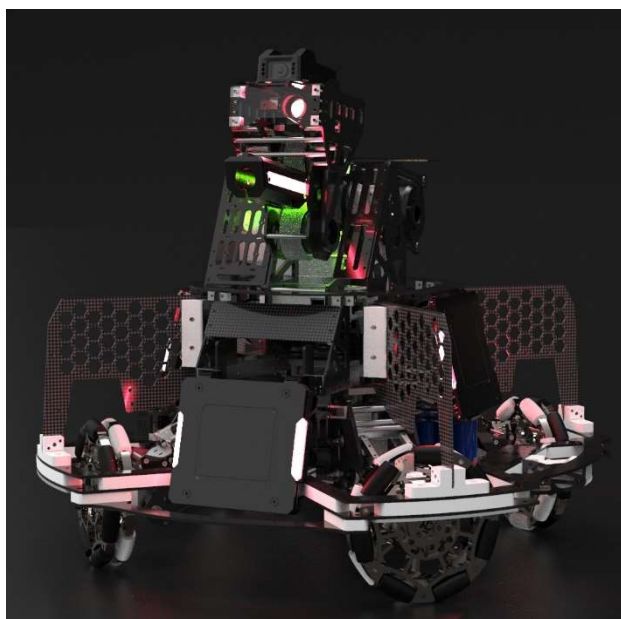

哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN



RoboMaster 2021 机甲大师超级对抗赛

步兵机器人技术说明文档



哈尔滨工业大学（深圳）南工骁鹰机器人队

2021 年 8 月

摘要

RoboMaster 作为一个以装甲板击打作为基础伤害原理的比赛，为了增加自身机器人的生存能力，唯一的办法就是尽可能降低对方的命中率。在 2018 年“小陀螺”由中国矿业大学首次带到了赛场上之后，在 2021 赛季已经变成了步兵的标配。但由于规则功率限制，和各大战队自瞄水平的提高，传统麦轮步兵的“小陀螺”时的速度和机动性已经不足以满足战场生存需要，故我们在 2021 赛季采用了 Omni 轮（全向轮）X 型布置设计，在未改变其直线加速能力的前提下，提高了其“小陀螺”速度，同时高低交错装甲板布置以干扰视觉识别，从而大大提高了生存能力。针对 2021 赛季赛场起伏路段和狭小的过道，我们首先创新的采用了圆形底盘，可顺畅的通过赛场的狭小空间，同时在被对面机器人卡位时，仍可以使用“小陀螺”手段防御。其次我们采用了四联动自适应底盘设计，保证了起伏路段和斜坡都有较好的通过性。

关键词：Omni 轮（全向轮），自适应，圆形底盘。

目录

机械部分	5
一、绪论	5
1.1 概述	5
1.2 机器人主要参数	5
二、云台设计	5
2.1 发射机构	6
2.2 pitch 轴	6
2.3 供弹	7
2.4 yaw 轴	8
2.5 弹舱	8
三、底盘设计	9
3.1 轮组	9
3.2 悬挂	10
3.3 交错式装甲板	10
3.4 保险杠	10
硬件部分	11
一、硬件框图	11
二、超级电容	11
电控部分	13
一、控制逻辑框图	13
二、卡尔曼滤波与自瞄预测	15
三、摩擦轮专用 PID 算法	17
四、UI 设计	18
视觉部分	20
一、系统组成	20
1.1 运算平台端	20
1.2 传感器端	20
二、视觉算法设计思路	20
2.1 思路概述	20

2.2 考量

2.3 代码框架

三、实际识别效果

20

21

24

机械部分

一、绪论

1.1 概述

我们战队本赛季步兵机器人底盘采用了 Omni 轮 X 型布置，环式自适应平行四边形悬挂，保证底盘的通过性和机动性。云台采用 yaw 轴供弹，纵置摩擦轮设计，降低了 pitch 的重量，提高了云台的响应速度。

1.2 机器人主要参数

底盘最小直径：
595mm/630mm

轮分布圆直径：520mm

总高度：498mm

发射机构高度：354mm

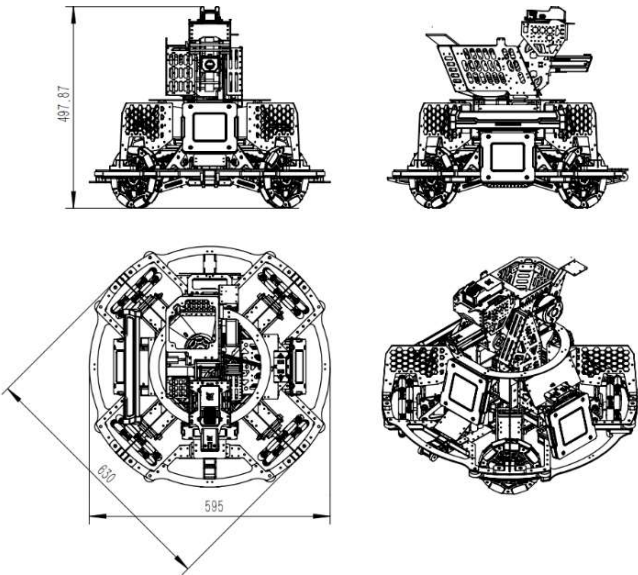
俯仰角：-25° ~+45°

总重：19.5kg

接近角：22° ~46° ※

通过角：22° ~35° ※

悬挂行程：80/110mm



二、云台设计

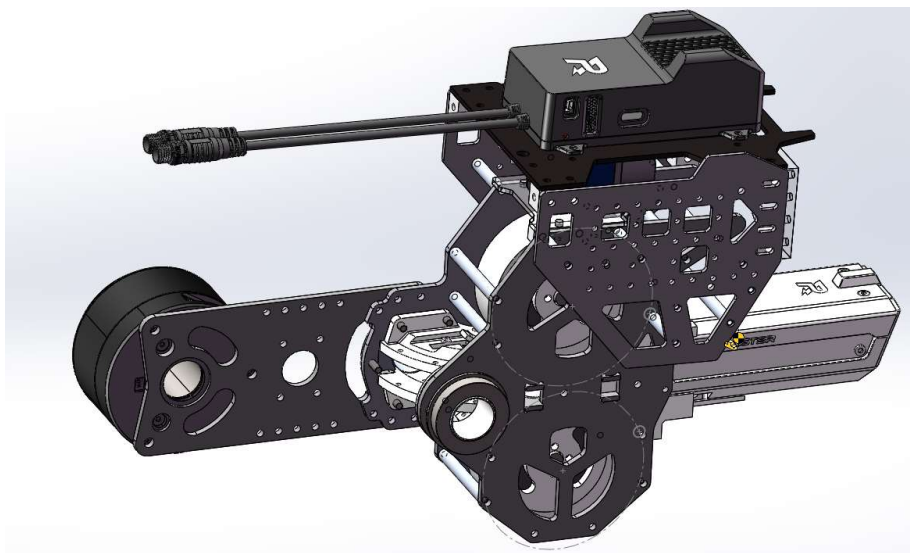
表 1 云台需求分析表

种类	指标
弹舱	300 颗小弹丸，补给站实际补单率达 90%以上，弹舱盖关闭下台阶不漏弹
射频	能够稳定实现单发、双发，最高射频可达25Hz且不卡弹
射速	射速稳定可调节，最高子弹射速不超过30m/s
射击准度	5米小装甲中速挡100%命中，8米小装甲高速挡100%命中
Yaw轴	采用6020电机直驱，保证结构强度，飞坡不断头，可穿过小滑环

Pitch轴	连杆，尽可能配平发射机构
云台支架	可容纳弹舱及小电脑
俯仰角	45° 仰角，-25° 俯角

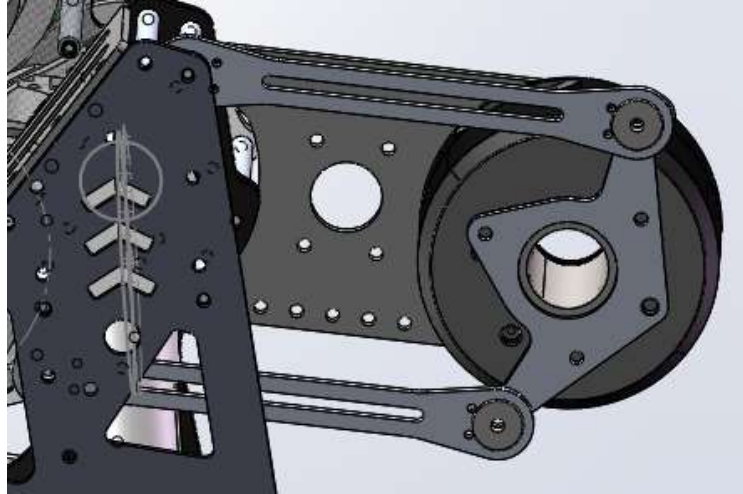
2.1 发射机构

根据空间布局需要，发射机构整体设计，采用纵置摩擦轮设计为纵置摩擦轮设计。电机采用 3508 电机去掉减速箱，摩擦轮采用溪地创新同款摩擦轮，直径 60mm。炮管方面采用板材和加工件结合方式的短枪管，单发限位采用上下 v 型轴承限位。荧光充能装置为官方提供的荧光充能板，置于炮管后尼龙材质的拐弯链路上。



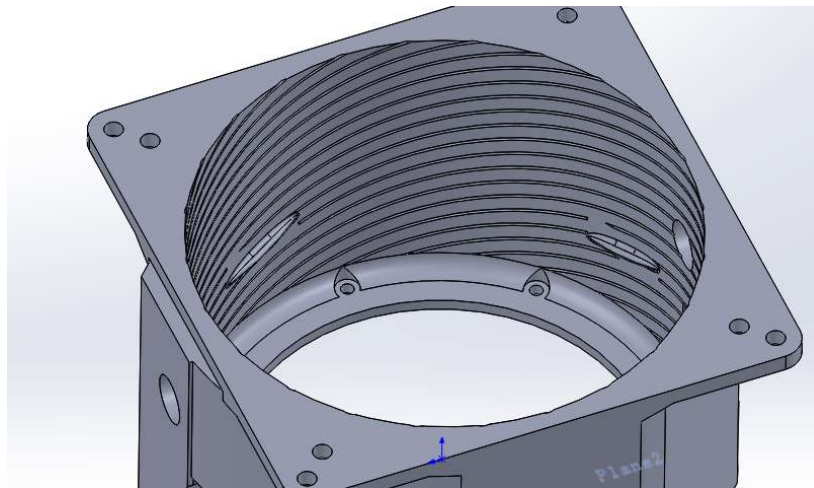
2.2 pitch 轴

Pitch 轴电机至于发射机构后端，距离 pitch 轴转动轴的轴线线 80mm，采用连杆控制俯仰。由于 yaw 轴供弹，摩擦轮布置较为靠前，导致发射机构重心较为靠前，此设计可一定程度配平发射机构的重量，降低发射机构水平时电机的负载，降低控制难度。



2.3 供弹

本云台设计采用 yaw 轴供弹的供弹方式，和传统弹舱不同，弹舱位于整个云台的中下部，不随发射机构俯仰进行运动。拨弹弹舱采用了螺旋线、扰动鳍破坏球的不稳定位置，强制下压。拨弹轮为八齿拨爪，同时采用横向轴承+球底环，消除拨弹中弹丸与底板、侧面的摩擦。供弹链路一共三个弯道，弯道处采用轴承润滑。



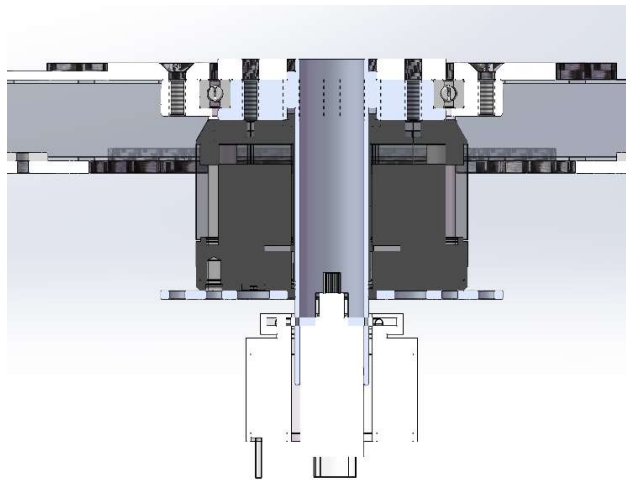
拨弹弹舱



拔弹盘

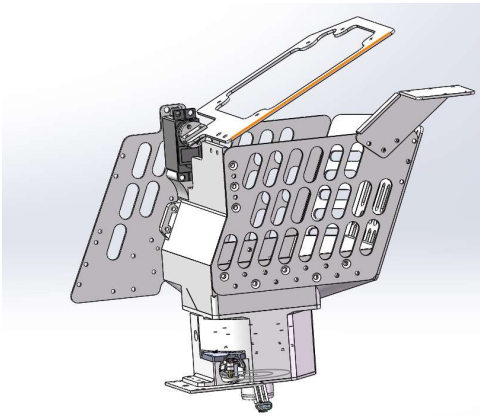
2.4 yaw 轴

Yaw 轴轴承采用 61810-2Z 深沟球轴承，由于交叉滚子轴承价格较高，重量较大，且深沟球配合加工件使用满足稳定性和结构强度需求，故采用更加便宜的深沟球轴承。



2.5 弹舱

弹舱在规则范围内尽可能做大做高，不仅仅保证了弹丸容量，实际可容纳 500 发弹丸；同时弹舱盖采用开合式，相对于旋转式弹舱盖，有效防止颠簸中弹丸洒落的情况。



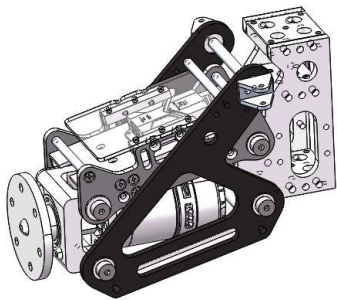
三、底盘设计

表 2 底盘需求分析表

种类	指标
机动性	采用全向轮，移动和麦轮相当，自旋速度更快
悬挂	自适应悬挂，有较强通过性
装甲板位置	实现装甲板的一个极致高低差
保险杠形状	圆形，自旋不会被墙体，敌方机器人卡住，方便更换及维修
底盘结构	整体结构简单，布局清晰，方便走线及安装
保护性	全方位保护，电路板，裁判系统的接口，线材不外露。电容、 电池等全方位保护。

3.1 轮组

轮组为平行四边形悬挂，并采用模块化设计，及保证了强度，也方便装配。

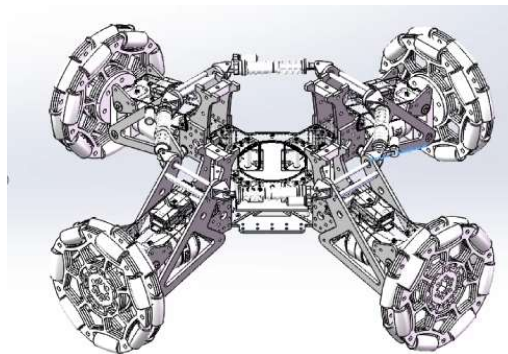


3.2 悬挂

悬挂采用环式自适应平行四边形悬挂，四个避震器水平放置介于两轮组之间。

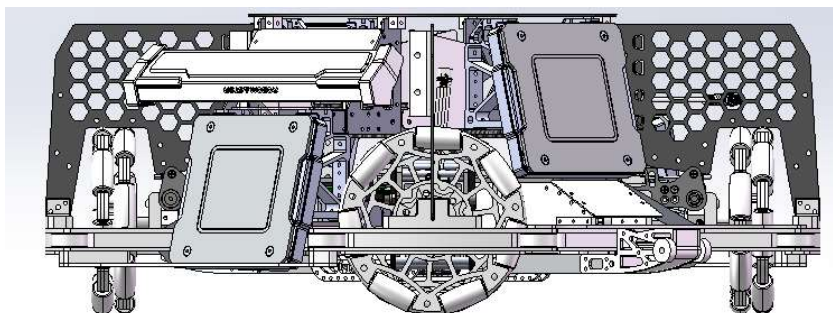
优点：保证轮组抓地、结构简单、可以空出中心云台位置。

缺点：空间占用较大、空间分割、轮组平行四边形悬挂设计难度较大。

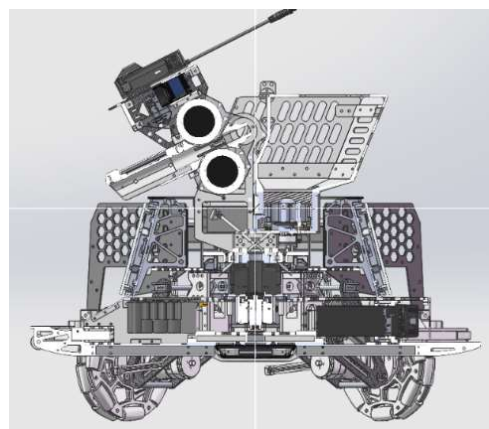


3.3 交错式装甲板

通过对底盘结构三层的划分，将装甲板在规则允许下尽可能的错层分布。低装甲 64mm，两装甲板间距 285mm，高装甲 139mm，两装甲板间距 412mm。



为了实现装甲板的一个极致高低差，不得不面临的问题是，装甲板的高度、车辆重心、发射机构的俯角将会冲突。于是在这里利用了一项规则，测速模块不计入装甲板的遮挡，将装甲板进行了缩入，这样在不断旋转的过程中，也能保持一个俯角，在近战中能取得优势。

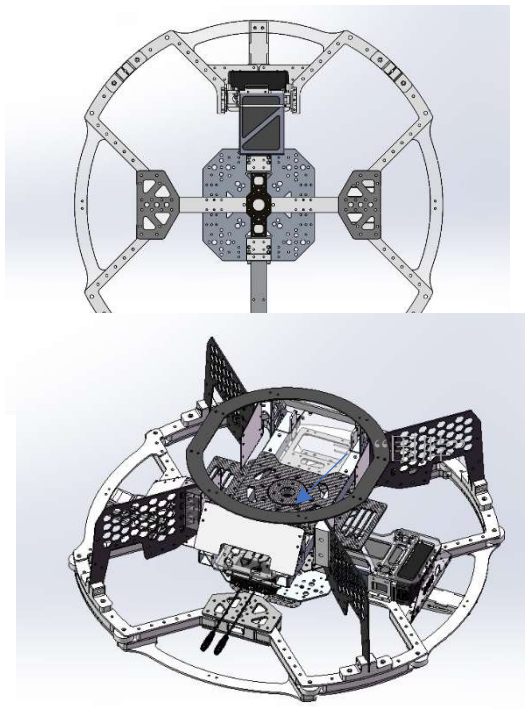


3.4 保险杠

结合底盘轮组布局，将保险杠设计为同心圆组成的类似圆形的结构，可有效避免场地元素，故我方机器人对步兵机器人的位移和小陀螺影响。

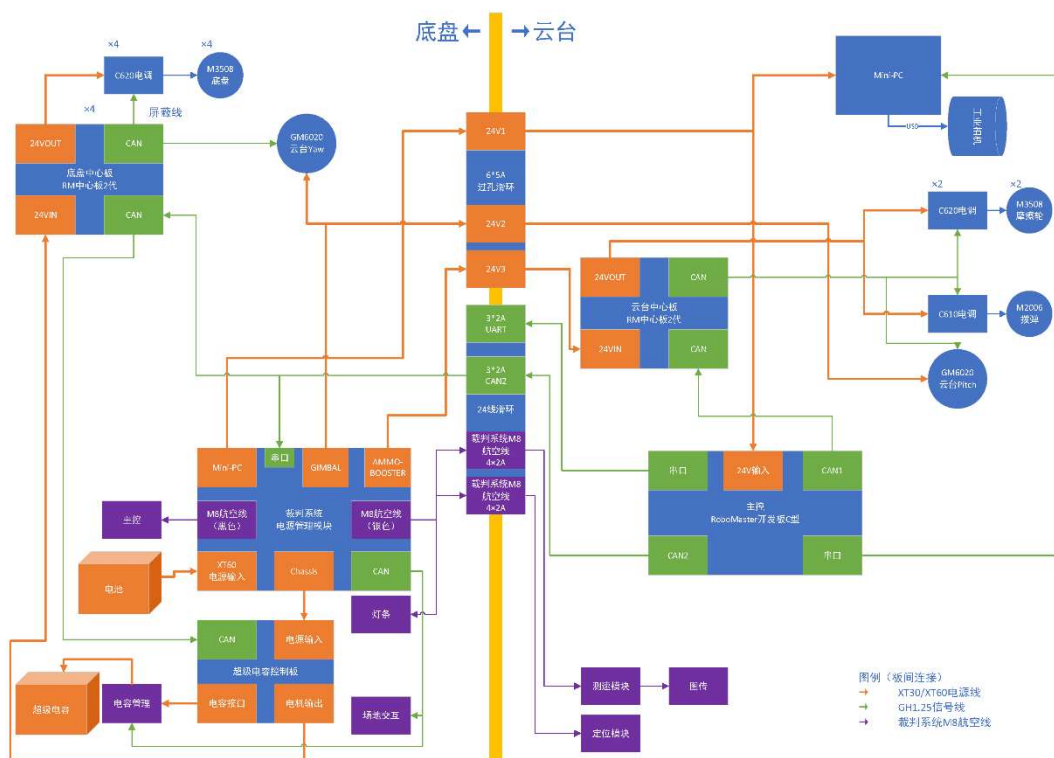
辅助设计“风扇板”结构，利用装甲板高低差和纵深的差距，分割装甲板区域，降低弹丸命中率（PS：理论上转速较快可以扇飞弹丸，不过效果有待商榷）。

上方“围脖”设计保证了装甲板安装和
风扇板的安装强度



硬件部分

一、硬件框图



二、超级电容

电容组：2.7V 8~10 串 54.7~63F 总容量 2000J

控制板设计：利用同步 BUCK 供电与负载反接变为同步 BOOST 的特性，理想二极管与 MOS 开关配合使用，充电降压低压侧电流 10A，放电升压低压侧电流 15A，电容最低极限工作电压 3V 提供 40W 放电功率，主控芯片：STM32G474CET6



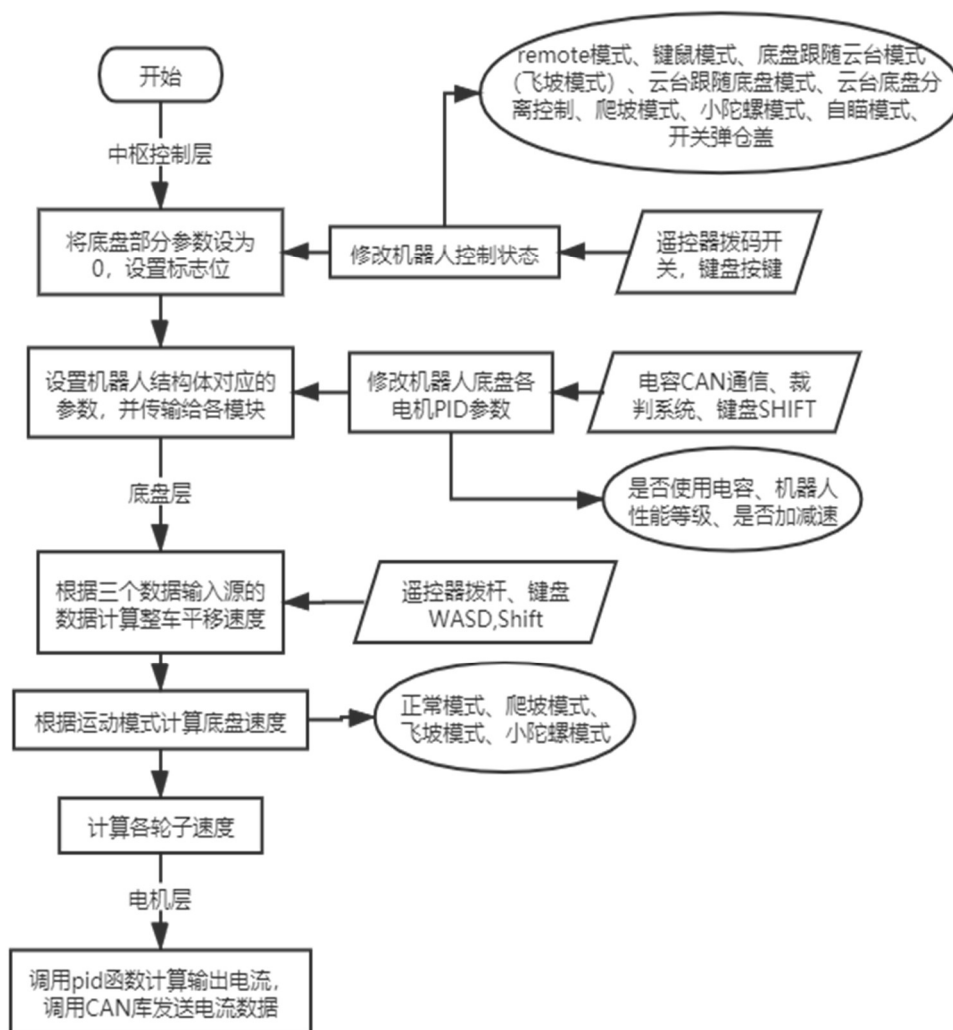
- 电流测量：INA240放大器
- 电压测量：0.1%精密电阻
- PWM频率：200kHz
- 主电感：10uH 16.5A
- MOS驱动：LM5106
- 电源：5V, 3.3V, 10V, 33V
- 模拟专用LDO：LP5907
- 2.5V基准电压
- 地平线分离，电感共地



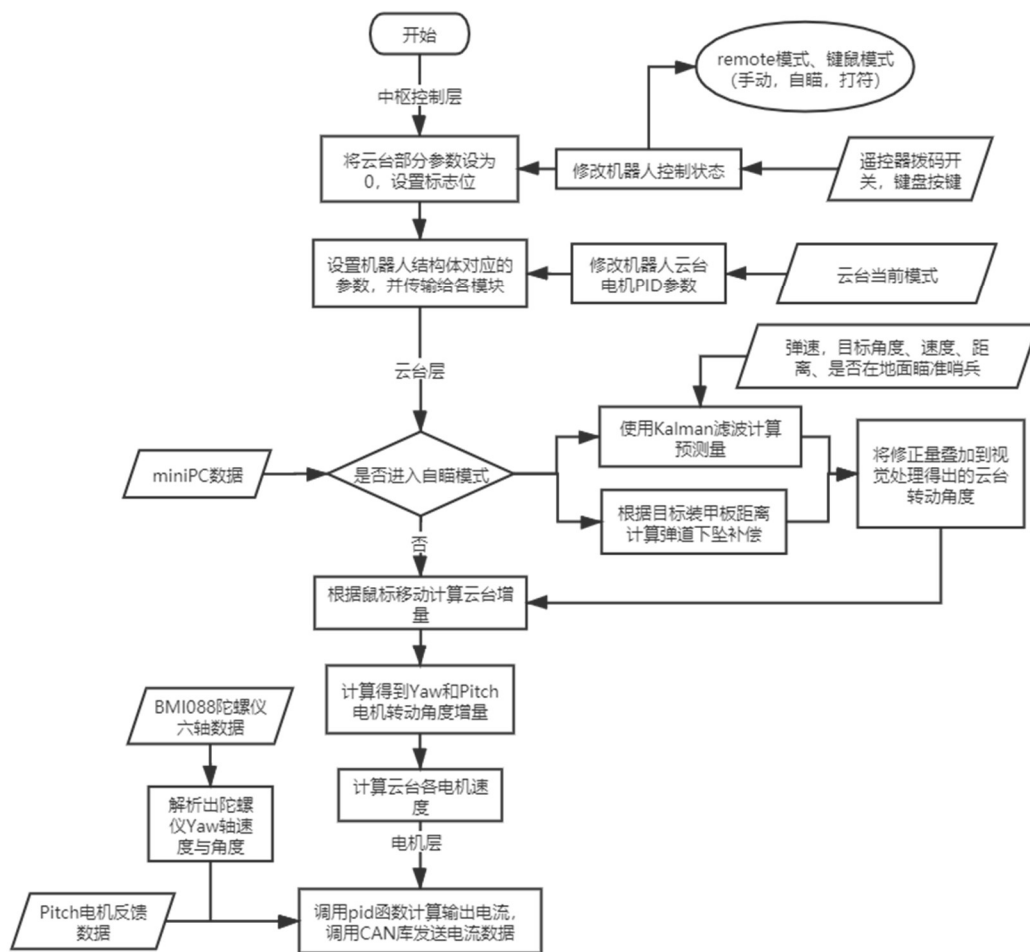
电控部分

一、控制逻辑框图

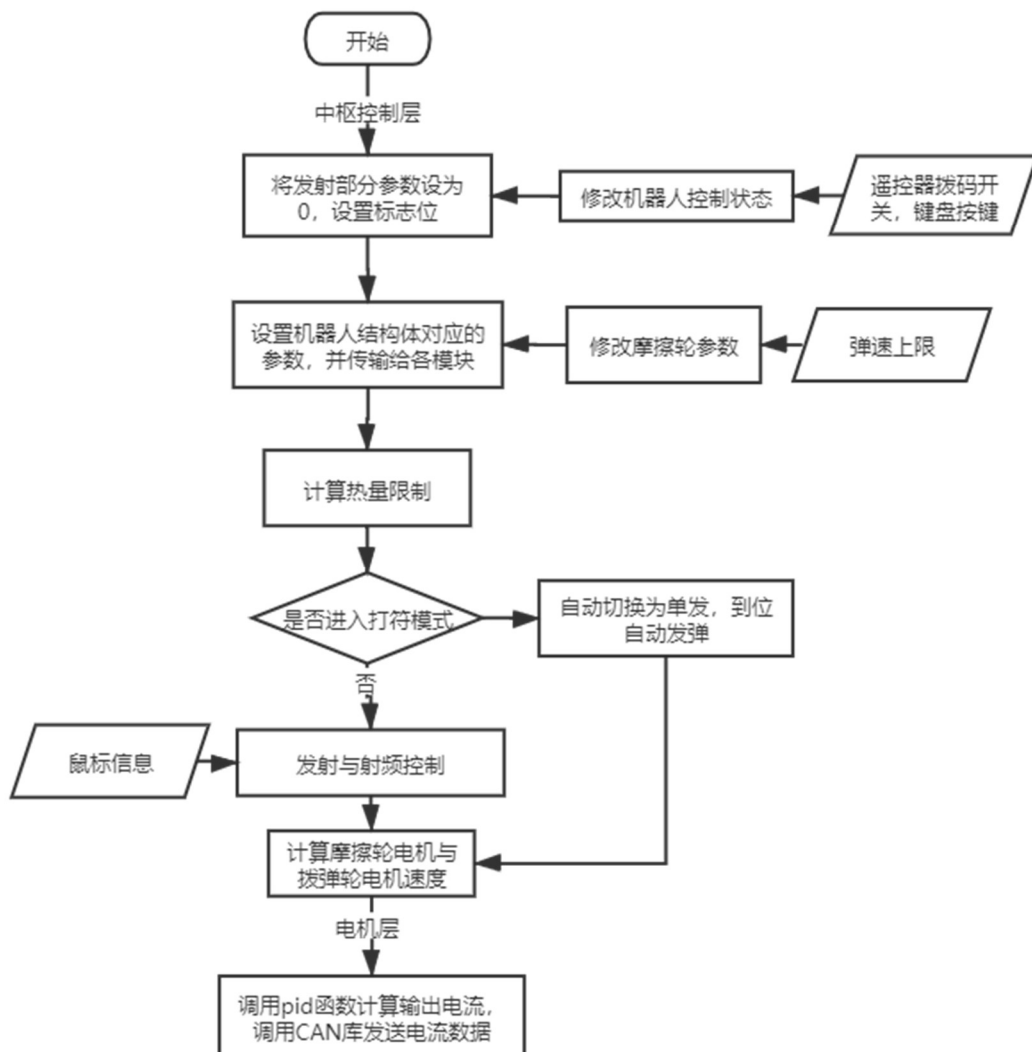
步兵机器人底盘控制逻辑



步兵机器人云台控制逻辑



步兵机器人发射控制逻辑



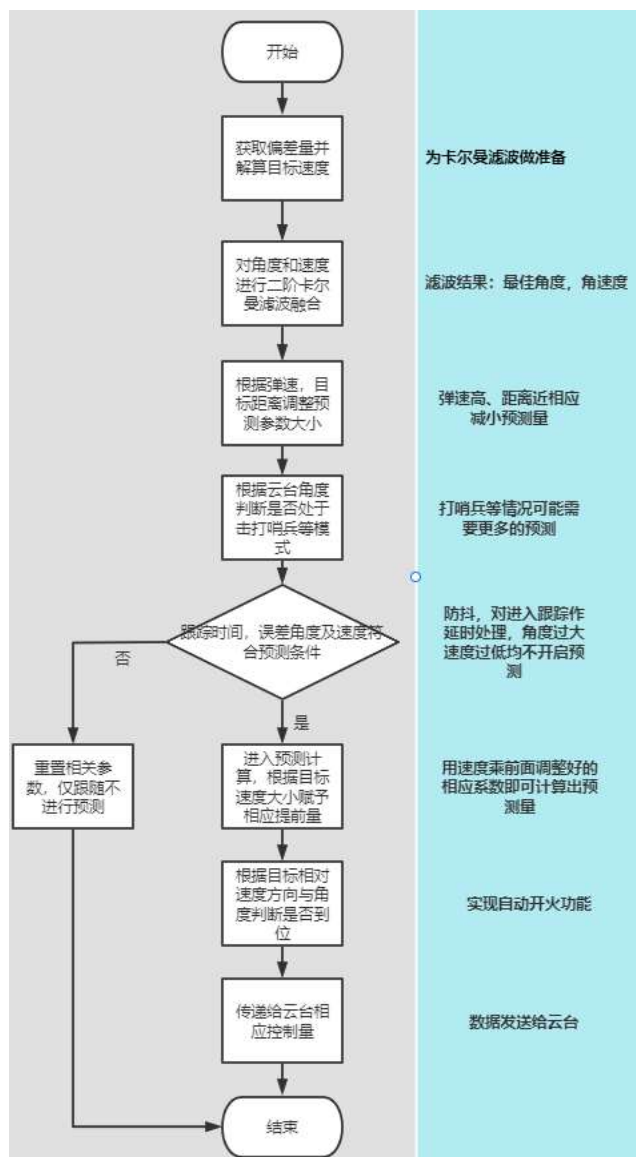
二、卡尔曼滤波与自瞄预测

由于队员水平和精力的限制，我们队伍的自瞄一直处于一个比较基础的阶段。在实际测试中，自瞄仅仅对于距离较近，移速较慢的目标有打击效果，而对于高速目标命中率低下。由于相机采集数据、视觉算法执行、数据传输、PID 控制，再到实际云台系统的固有延迟，加上子弹有飞行时间，延迟来自于各个环节，如

果进行预测，只能表现为每一发小弹打在目标的后面，因此卡尔曼滤波，这一带有预测功能的滤波算法在此就有必要。在学习这方面知识的过程中除了网上的一些教程，要特别感谢官方的开源，华南理工大学华南虎战队的教学视频以及太原理工大学，深圳大学的电控开源部分。

卡尔曼滤波的实现涉及大量矩阵运算，一些开源的电控代码使用了将矩阵运算拆开写的办法以提高程序执行速度，但是这样对于读者来说并不那么便于理解。为了使算法更高效的运行，需要启用主控的 FPU（浮点运算单元），因为我们所使用的 STM32-F407 属于 Cortex-M4F 构架，它与 M0、M3 的最大不同就是具有 FPU（浮点运算单元），支持浮点指令集，因此在处理数学运算时有更高的效率。为此我们使用了 **arm_math.h** 库，其中所含的矩阵处理函数能够便于我们实现卡尔曼滤波。

具体的预测逻辑如下图



这种预测方法能够让机器人更为准确的跟踪目标, 在测试中也有比较好的效果。不过场上由于环境光的干扰, 在攻击哨兵时常常出现误识别的情况, 这使得实际效果并没有完全达到预期。这套方法也存在着参数较多, 难以调整的问题。下个赛季将进一步优化预测逻辑, 同时, 在视觉抗干扰, 提升帧数, 云台控制等方面作更进一步的改进。

三、摩擦轮专用 PID 算法

在与机械组队员们一起优化步兵弹道的过程中, 观察摩擦轮的速度曲线, 我们发现在每次连发中会有所谓首发离群的问题, 即在第一发以后, 摩擦轮转速会出现一个稳态误差现象, 这导致单发与连发打出去的结果差异较大。同时, 摩

擦轮的速度恢复也还有提升空间。在重新调节 PID 参数的基础上，我们重新研发了一套摩擦轮 PID 算法。这套控制算法主要将控制分为粗，中，细调三个阶段。在误差较大的粗调阶段，使用 bangbang 算法，在中间段仅使用 PD 控制，不加入积分环节，而在细调阶段使用较小的比例参数，并加入积分环节以消除误差。经过测试，这套算法具有较好的效果，对弹道稳定性有明显提升。

四、UI 设计

为了让操作手更方便地了解机器人的控制状态，我们设计了步兵机器人专用 UI 系统。



位于左侧的是状态指示灯与状态栏，第一行是击打能量机关的指示灯，没进入打符状态前指示灯为白色，进入击打小能量机关模式时指示灯变为己方颜色，击打大能量机关模式时指示灯变为紫色；第二行是摩擦轮状态，为了节约电能，比赛前摩擦轮默认关闭，指示灯为橙色，当开始比赛后，开启摩擦轮，指示灯变绿；第三行是弹仓盖状态，打开时为橙色，关闭时为绿色，防止操作手忘关；第四行是云台控制状态，分为跟随云台陀螺仪模式和跟随底盘模式（用于飞坡），正常模式为跟随云台陀螺仪；第五行是底盘模式，分为正常模式、小陀螺模式、爬坡模式（底盘无加速度限制）、云台与底盘独立控制模式，不同状态对于不同颜色指示灯；比赛开始时，操作手确保下面四个指示灯都是绿色就可以放心出发

了，代表一路绿灯~

位于右侧的是剩余热量对应发弹量，因为电控代码中有根据剩余热量限制发弹的代码，因此这个数字代表了操作手可以一次性打出去的弹丸数量。

视野中央的是自瞄提示框，进入自瞄模式后出现白框，锁上目标（离目标距离在阈值内）后变成绿框，提示操作手可以开火，避免尚未瞄准时不必要的弹丸浪费。

视野下方的是车体防撞框，如果建筑物在视野中进入了防撞框内，就说明有撞上的可能。

视野中央偏下的是，发射弹丸瞄准提示线，三根线分别代表了距离目标装甲板 3m, 4m, 5m 时所用的瞄准线，将瞄准线与目标装甲板下沿对齐，就对应了可以击中目标的仰角。

左上方的是超级电容电压指示条，用于指示当前超级电容剩余电量，当电量过低时，需要稍微减少加减速才能充电，电容电量达到一定值时才能保证步兵快速上坡，避免因为电容没电上不去被击杀；飞坡前也需要确保电容已充满。

视觉部分

一、系统组成

1.1 运算平台端

使用了自行组装的台式机

硬 件 设备	操 作系统	运行库	ToolChain
Nvidia Xavier NX/AGX	Ubu ntu18.04	Opencv4.5.1 Opencv_contrib 4.5.1 GLOG 大恒相机驱动 TensorRT	Ubuntu18.04 cmake 3.19 gcc 7 cuda 10.2 TensorRT 7

1.2 传感器端

硬件设备
大恒工业相机、陀螺仪

二、视觉算法设计思路

2.1 思路概述

视觉识别在 Robomaster 赛场上尤为重要，是提升己方作战效率，获得能量机关增益的必须手段。视觉识别经过多个赛季的发展，传统算法已经较为完善，但是对环境光、遮挡等不良条件非常敏感，而随着边缘计算设备和轻量级目标检测模型共同发展，在机器人等嵌入式平台上使用泛化能力更好的神经网络来完成视觉任务已经成为一种趋势，而我们也在本赛季采用了相关技术。

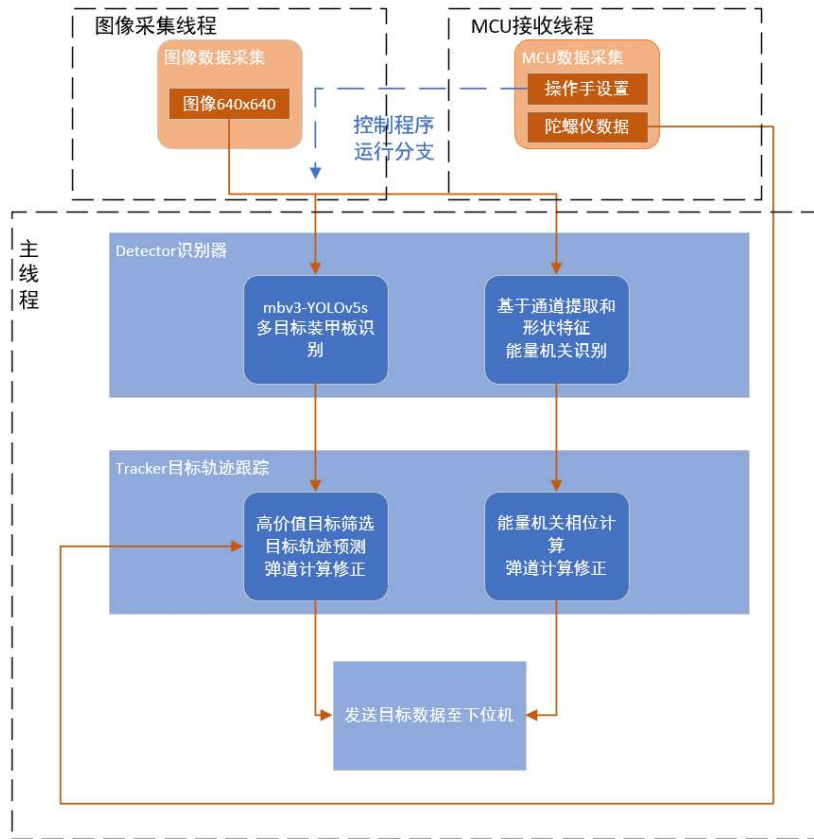
2.2 考量

视觉算法主要考量在极端环境下的识别/误识别概率以及检测速度，我们从

数据集和网络模型两方面入手，达到了可观的效果。

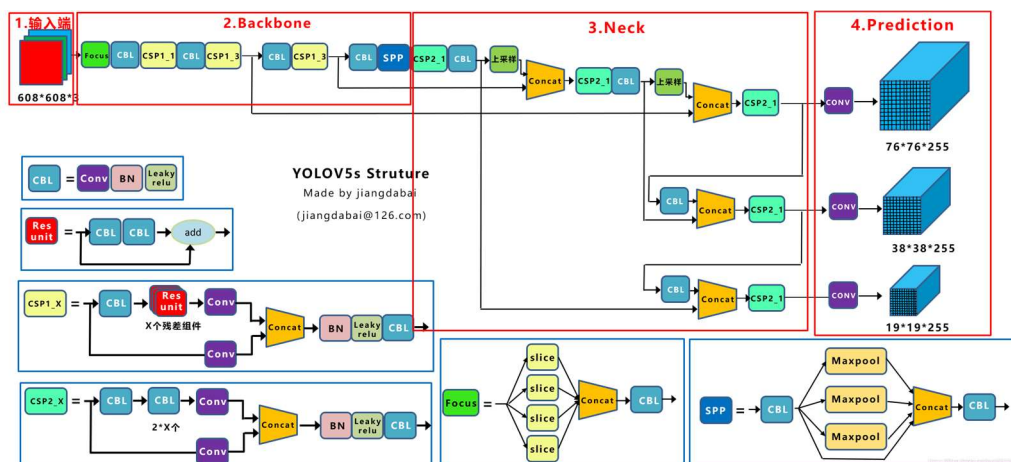
2.3 代码框架

2.3.1 以数据流为基础的程序框架图



2.3.2 重要识别算法 mbv3-yolov5s

2.3.2.1 原版 YOLOv5s

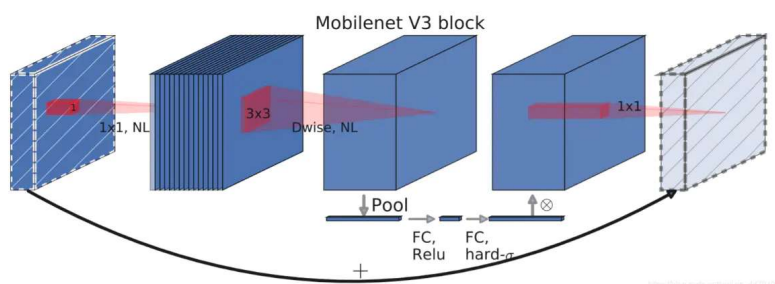


模型参数量仍然较大，在低内存带宽、较低 CPU 性能的环境下发挥不尽人意。在 Xavier NX 上使用 TensorRT 加速后仍然只能达到 50+ fps（640 X 640），不能满足要求。尝试缩小分辨率至 320 X 320，则小目标远距离识别不满足效果要求。

2.3.2.2mobilenetV3 + yolov5s

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

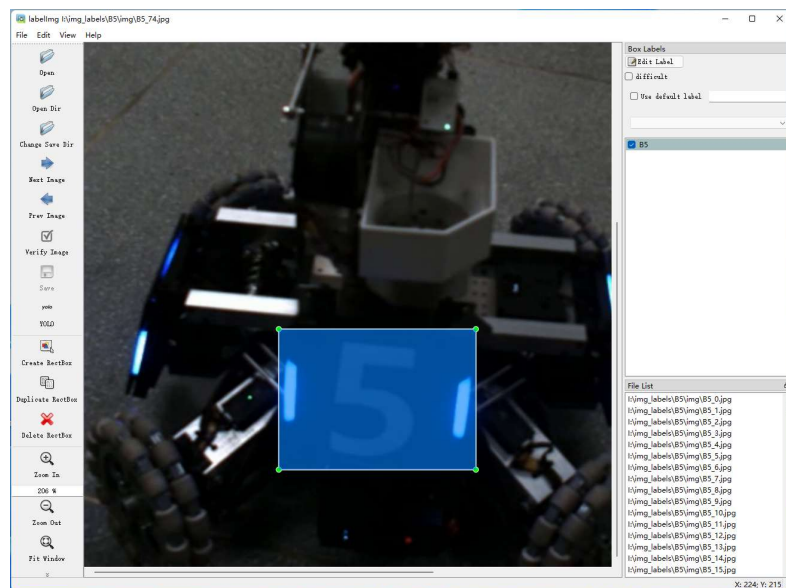
bneck结构如下图所示：



在替换骨干 backbone 网络为 mobilenetV3 之后，网络参数量减小，速度提升，在 NX 20W 模式下可达到 90~100 fps，基本满足了识别要求，经过测试对 7 米远的目标仍能较好的识别到。

2.3.3 数据集制作与模型训练

2.3.3.1 手动标注

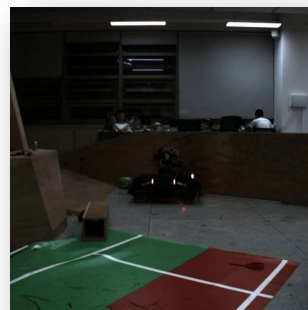
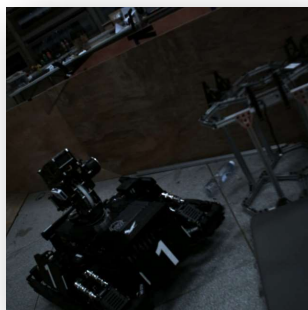


采用 labelImg 方式手动标注数据集。

2.3.3.2 自动标注

使用传统视觉获得边框，确保视野中只有一种类型的装甲板。后续通过程序自动处理生成标注数据，缺点是一张图片中只有一处正样本。

2.3.3.3 误识别处理



添加了不亮的装甲板和没有数字的装甲板等数据作为负样本，增强了算法对特殊光照等情况的处理能力。

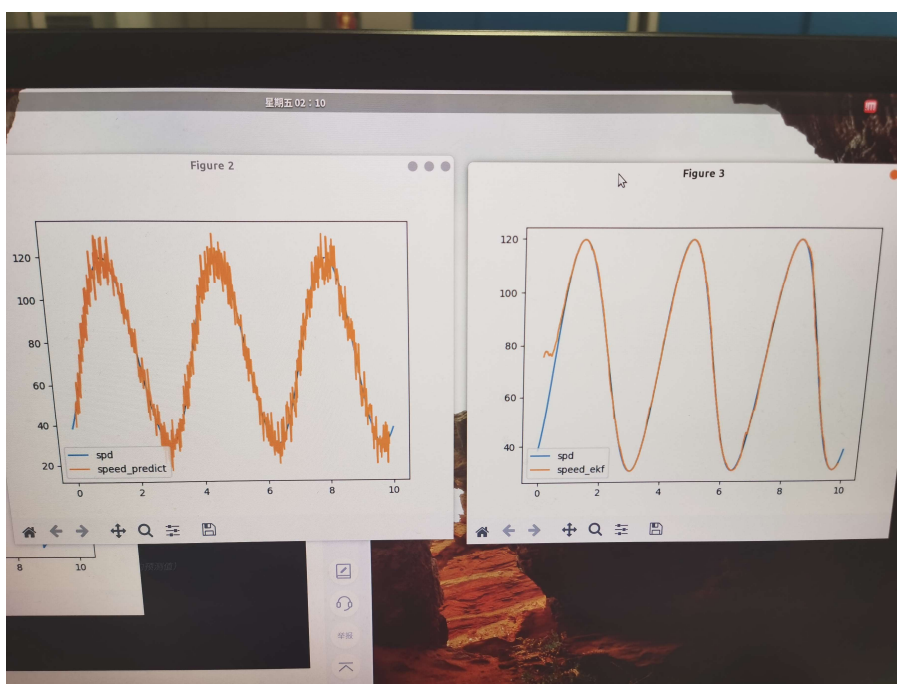
2.3.4 与下位机实时通讯

使用上层通用接口 RmSerial 类统一管理通信，调用时只需与 RmSerial 类对象进行交互。RmSerial 类的子类分别实现了 Can 通信和 Serial 串口通信。数据包格式如下。


```
typedef struct _SendData {  
    uint8_t start_flag;  
    float x;    // Yaw  
    float y;    // Pitch  
    float z;    // state  
    uint16_t u; // shoot_delay;  
    uint8_t end_flag;  
} SendData;
```

2.3.5 能量机关相位计算

采用 EKF 扩展卡尔曼滤波进行相位计算，使用时间和角度对初相位进行解算。使用 Python 生成噪音并跟踪效果图如下，左图为均值滤波之后的结果，右图为将中值滤波结果再进行 EKF 卡尔曼滤波的结果。



三、实际识别效果

